uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Frédéric Mustière

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical and Computer Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Speech Enhancement in Real-World Environments Using State-Space Based Algorithms

TITRE DE LA THÈSE / TITLE OF THESIS

Martin Bouchard

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Miodrag Bolic

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Hilmi Dajani

Rafik Goubran

Sri Krishnan (Ryerson University)

Yongyi Mao

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Speech enhancement in real-world environments using state-space based algorithms

by
Frédéric Mustière

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in Electrical and Computer Engineering

Ottawa-Carleton Institute for
Electrical and Computer Engineering

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

This work was conducted under the supervision of
Dr. Martin Bouchard and Dr. Miodrag Bolić

Your file Votre référence
ISBN: 978-0-494-69107-6
Our file Notre référence
ISBN: 978-0-494-69107-6

Acknowledgements

I would like to show my gratitude to my supervisors, Dr. Martin Bouchard and Dr. Miodrag Bolić, for all of their encouragement and expert guidance during the course of my research and studies.

I am also sincerely thankful to my family and friends for their support, and to the University of Ottawa as a whole for offering me the opportunity to perform this research.

Abstract

The family of state space-based speech enhancement algorithms is taken as the central building block for this thesis; such algorithms include for example Kalman Filters running on autoregressive models of speech. The goal of this thesis is to render this family of algorithms "real-world" ready, that is, upgrade the existing solutions to make them more flexible and more robust to real-world situations where different types of nonstationary, non-Gaussian noise are to be expected. The chosen context is that of sensitive applications such as hearing aids for which naturalness, intelligibility, and noise reduction are equally important.

Most of the state space-based speech enhancement algorithms are currently unable to operate robustly in nonstationary colored-noise environments, and those found in the literature usually assume AWGN conditions. Since in addition, they are in general more computationally demanding than other well-established algorithms, it is not surprising that they are often either overlooked or omitted in respected and recent speech enhancement publications, as the introduction in this thesis will underline. Nevertheless, for AWGN conditions they have still been reported to yield high quality and natural-sounding results, making them an appealing choice for the range of applications chosen.

To achieve the goal of this thesis, it is first necessary to categorize, implement and test many state-space based algorithms for speech enhancement; as it turns out, most of the existing algorithms revolve around the same principles and ideas. Then, in this work several tools and extensions of state spaced-based algorithms are developed in order to tackle the required handling of real-world noise. In parallel, since one of the main goals is to preserve naturalness and intelligibility, a variety of configurations based on these proposed tools and extensions are concocted and thoroughly tested so as to be able to determine the best solution both in

terms of output quality and complexity requirements. As an additional constraint, most of the tools devised are meant to be "compatible" with any state space-based algorithm, rendering the work in this thesis homogeneous and unifying.

To deal with real-world noise, novel fullband and subband solutions that take into account information from existing noise estimation algorithms are proposed. In the fullband domain, new all-pole modelling techniques are devised, and in the subband domain several alternatives to handle the noise are proposed as well. Throughout the thesis, several important accessory contributions are also developed, such as a new category of particle filtering algorithms for nonlinear speech models, low-cost post-processing techniques for further noise reduction, binaural extensions of monaural state-space models, and various improvement ideas specifically targeted to state-space algorithms running on autoregressive speech models.

As an example, a possible state-space based algorithm is proposed based on some of the conclusions of the several Chapters of this thesis, which is shown to outperform three well-established state-of-the-art algorithms in adverse conditions. Rather than merely justifying the work in this thesis by a final arm-wrestling contest, these latter results prove that state-space based algorithms deserve a place amongst highly regarded and viable algorithms for speech enhancement.

Table of contents

I. Introduction

1.1. Research motivation and goal of the thesis

The goal of speech enhancement is to reduce or remove the background noise contaminating recorded or transmitted speech signals. With more and more applications emerging (e.g. personal recording devices or cellular phones, or with the rising need for hearing aids in the general population), speech enhancement is a continuously evolving research field with a wide variety of existing techniques. Moreover, no current speech enhancement algorithm is universally recognized as the "best" solution in every aspect and every context [HU'06-2]. Practically speaking, each method has its strengths and weaknesses, and in general enhancement algorithms can be characterized by:

- The amount of noise reduction
- The amount of distortion (or "damage") inflicted to the speech at the output of the enhancer
- The effect of the algorithm on intelligibility
- The amount and the nature of artefacts introduced in the enhanced speech
- The flexibility of the method (i.e., for what range of noises or speakers will it perform as intended, and conversely, when will it fail?)
- The computational complexity of the enhancement system.

In general, one algorithm cannot excel simultaneously in all of the above categories; specifically it is well known that one of the central issues in speech enhancement consists of the trade-off between the amount of noise reduction and the output naturalness.

State space-based algorithms constitute a family of model-based speech enhancement and noise reduction methods, with many existing variations. In very generic terms, they operate on a time-domain, instant-by-instant production model for the speech and measurement signals – a *state-space* model – on which sequential algorithms such as the Kalman Filter [KAL'60] can be applied. Some of their advantages include the absence of musical artefacts (such as the ones observed in many frequency-domain enhancement algorithms), the ability to process data sample-by-sample, and a good quality and preserved naturalness in the enhanced speech [BEN'05]. On the downside, according to the same reference, they reportedly do not achieve as much noise reduction as other types of algorithms, they tend to be computationally more expensive, and importantly, currently in the literature they are for the most part not equipped to deal with real-world situations with common low to very low signal-to-noise ratios, where robustness is of utmost importance. The above combined downsides are generally enough for practitioners to resort to other better established solutions; for example, in the recent respected publications [LOI'07] and [HU'06-2] fully dedicated to speech enhancement in general, they are not mentioned.

In this thesis, the chosen applicatory context is that of sensitive applications such as hearing aids, for which naturalness is as essential – if not more – as the amount of noise reduction resulting from the enhancement scheme. The reported qualities of state-space based algorithms above make them suited to fulfil this criterion; however in the context of interest the common situation is of low SNR, with highly fluctuating environments and speakers – which is a problem that is not well tacked by these algorithms in the literature.

Motivated by their reported quality in terms of naturalness, this thesis aspires to create a framework for these model-based algorithms to handle real-world situations. To be more precise, "real-world situations" here refers to realistic noisy speech conditions, i.e., prone to one or more of the following:

- **Nonstationary, colored noise**, as typically heard in everyday situations (e.g. cafeteria, street, factory noise, etc),
- **Low signal-to-noise (SNR) ratio**; in other words, the noise level is at least as high as the speech level,
- **Reduced intelligibility**; that is, the nature and/or level of the speech or noise adversely affecting the decipherability of spoken words or sentences.

Since current solutions are for the most part limited to "academic situations" such as additive white Gaussian environment noise, one of the accompanying goals is to devise ways to augment their robustness to adverse environments, while retaining as much as possible their qualities in more basic situations.

In order to reach the above objective, a relatively large sample of algorithms from this family are first categorized, implemented and tested. We highlight that most of the existing algorithms are based on the same principles, and many components of several seemingly "mutually exclusive algorithms" are in fact interchangeable. In parallel to this first step, the existing state-space algorithms are adapted to accommodate several novel tools aimed at improving their overall output quality, complexity and flexibility. For example, a novel way of handling autoregressive colored noise in generic state-space equations is shown, with considerably lower computational requirements and equivalent output quality. Additionally, a

new branch in the family tree of these algorithms is also added and found to yield high-quality enhanced signals: it consists of particle filtering solutions applied to nonlinear speech models.

In order to accommodate real-world noise, it is chosen to make use of the relatively mature field of noise power spectrum density (PSD) estimation, as opposed to voice activity detection (VAD), in large part because of the greater robustness of noise PSD estimators [RAN'06]. Indeed, while VADs may perform well in low-complexity, stationary noises, their reliability quickly breaks down in realistic noisy environments. In fact, well-performing noise PSD estimators are technically able to track and update the noise PSD even during speech, which VADs are by nature unable to do. However, state-space algorithms operate in the time domain, and are not "ready" to directly include noise PSDs in their framework. Thus, some ways to incorporate the information returned by noise PSD estimators are then devised. Two distinct approaches are proposed: *fullband* and *subband* approaches, both investigated in details. In the fullband approach, new techniques pertaining to all-pole modelling of the noise PSD are shown; in the subband approach, which is presented as a compromise between time and frequency-domain enhancement, various solutions are given, including a full "discretization" of the noise PSD, thereby turning the global problem into small "white-noise-only" subproblems. In the subband domain, additional methods are given to improve the output quality, including the incorporation of psychoacoustic constraints, some post-processing aimed at reducing the amount of residual/artificial noise introduced in the enhanced speech, and the exploitation of the results in the field of Bandwidth Extension to assist the enhancement algorithm in the treatment of the high-frequency parts of corrupted speech.

Finally, some binaural extensions to the classical monaural cases are shown, along with other methods currently in development, categorized as future work projects. Then, a certain algorithmic setup, judged to be the most practicable in the selected context of the thesis, is described and some comparisons with state-of-the-art algorithms are made, showing that the state-space based method is a viable alternative, especially in low SNR conditions. The major algorithms devised are available online at http://www.site.uottawa.ca/~bouchard/papers/mustiere_thesis.zip , which also contains some audio demonstrations to illustrate the conclusions of the thesis.

1.2. Organization of the thesis

In this thesis, every chapter contains some relevant simulation results, with long speech segments and various noise conditions, and several objective quality measures. The speech and noise material, the detailed methodology, and the objective measures and performance metrics are described in **Chapter II**.

Having described the research objective in Section 1.1, the next chapter, **Chapter III**, is dedicated to a review of the central components that will be used and referred to throughout the thesis – that is, a review of existing state-space based algorithms. Special attention is paid to notation, with an effort to keep it consistent from the beginning to the end of the thesis. In Chapter III, first the speech, noise, and measurement processes are defined. Then, the underlying essential tools used in state-space based algorithms are recapitulated, namely several members of the Kalman Filtering family of methods. Finally, many existing state-space algorithms are presented, tested and compared in White Gaussian noise conditions, so as to get a view of their performance in basic situations. Finally, some generalities about

background noise estimation methods are given, and a brief overview of other speech enhancement techniques is presented.

Before moving on to the heart of the subject, in **Chapter IV** three miscellaneous "tricks" are introduced, which have been found to bring some overall improvements at low computational cost when applicable. The first two can be used in conjunction with every state-space based algorithm, and the last one is specific to particle filters.

Chapter V presents a simple alternative to the traditional handling of autoregressive colored observation noise processes in state-space based speech enhancement algorithms. The proposed approach decreases the dimension of the state vector and the amount of computations per iteration, and also naturally reduces to the white noise case when a zero-order autoregressive colored noise is chosen – all without losing any of the benefits in terms of enhancement (in fact, some cases are shown to yield even better results with the proposed method). In addition, any existing state-space based algorithm can be easily modified to accommodate this new approach.

In **Chapter VI**, a study of particle filtering solutions to the problem of speech enhancement in the context of nonlinear/neural-type speech models is conducted. Several variations of a global speech model are presented (single/multiple neurons; bias/no bias), and corresponding particle filtering solutions are thoroughly derived. The resulting algorithms resort to the colored noise handling shown in Chapter V.

In **Chapter VII**, some practical fullband solutions to incorporate the information returned by background noise estimation methods into state-space based algorithms are shown. This is

also done in a way that is compatible with the colored noise treatment of Chapter V, and the entire chapter is in fact closely related to all-pole modelling techniques. New modelling methods are tested against existing ones, and their effects and advantages are studied.

In **Chapter VIII**, as opposed to the previous chapter, the enhancement algorithms are assumed to treat the speech as a combination of decimated subband signals. Besides again working on the incorporation of noise information into the algorithms, we are here interested in taking advantage of a subband structure to reduce the overall workload, further reduce the residual noise (post-enhancement), and take into account perceptual criteria. In essence, this sort of setup constitutes a compromise between time and frequency domain solutions, since the state-space based methods still operate on a sample-by-sample basis on the subband signals. As a brand new approach, the concept of Bandwidth Extension is utilized as well in the context of speech enhancement, used here to reinforce the high-frequency estimates of the clean speech.

In **Chapter IX**, an example of state-space based algorithm is given based on the conclusions and recommendations from all previous chapters. The ultimate goal is to demonstrate that the family of state-space based algorithms can constitute a viable alternative for speech enhancement in realistic noise conditions; thus, some comparative simulation results are analyzed where three other state-of-the-art, well-reviewed algorithms are used.

Chapter X presents two other topics under investigation, with some work on the better modelling of the excitation signal in AR representations of speech, and on some binaural extensions of the monaural methods shown before.

Finally, **Chapter XI** closes the thesis with the main conclusions stemming from the theoretical developments and the experimental results.

1.3. Publications and other information

At the time that this thesis was written, the following articles, all related to the contents of this thesis, were either published or accepted for publication.

1.3.1. Journal papers

- F. Mustière, M. Bolic, and M. Bouchard, "Speech Enhancement based on Nonlinear Models using Particle Filters", IEEE Transactions on Neural Networks, vol. 20, no.12, pp.1923 – 1937, Dec. 2009.
- F. Mustière, M. Bouchard, and M. Bolic, "Low-cost modifications of Rao-Blackwellized particle filters for improved speech denoising", Signal Processing, vol. 88, no. 11, pp. 2678-2692, Nov. 2008.
- F. Mustière, M. Bouchard, and M. Bolic, "All-pole modeling of power spectral densities: a faster convergence and an alternative mean-squared-error cost function", submitted in July 2009 to IEEE Transactions on Audio, Speech, and Language Processing (based on Chapter VII)

1.3.2. Conference papers

- F. Mustière, M. Bouchard, and M. Bolic, "Efficient SNR-based subband post-processing for residual noise reduction in speech enhancement algorithms", Proceedings of 2010 European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark, Aug. 2010
- F. Mustière, M. Bouchard, and M. Bolic, "Real-world Particle Filtering-based Speech Enhancement", Proceedings of 2nd International Workshop on Cognitive Information Processing (CIP) 2010, Elba Island, Italy, June 2010
- F. Mustière, M. Bouchard, and M. Bolic, "Bandwidth Extension for Speech Enhancement", Proceedings of 23rd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2010, Calgary, Canada, May 2010
- F. Mustière, M. Bolic, and M. Bouchard, "Improved Colored Noise Handling in Kalman Filter-based Speech Enhancement Algorithms", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2008, Niagara Falls, Ontario, May 2008.
- F. Mustière, M. Bolic, and M. Bouchard, "Quality Assessment of Speech Enhanced using Particle Filters", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2007, Honolulu, Hawai'i, May 2007.
- F. Mustière, M. Bouchard, and M. Bolic, "A fast convergence two-step procedure for AR modeling of power spectral densities", submitted in April 2010 to the 2010 IEEE Workshop on Signal Processing Systems,

The following papers are in writing and/or being readied for publication:

- Combination of small-size Particle Filters and distributed resampling (Section 4.3)
- Model-based speech enhancement with improved excitation model (Section 10.1)
- Efficient binaural extension to subband state-space based speech enhancement algorithm (Section 10.2)

1.3.3. Accompanying MATLAB code

Some accompanying and illustrative MATLAB code for this thesis, as well as some sound demos of the results from this thesis, can be found at http://www.site.uottawa.ca/~bouchard/papers/mustiere_thesis.zip

II. Methodology and performance metrics

The methodology and evaluation criteria are of major importance in our context, since they are part of the problem definition, whereas the subsequent algorithms and proposed methods are only solutions or solvers of the defined problems. To answer this question in the context of speech enhancement, we propose to systematically employ multiple objective measures (presented below), to also systematically give our subjective impressions, and finally to use a fixed set of speech and noise data throughout the thesis. The experienced reader might object that objective quality measures sometimes give contradictive scores. Thus, a method of "algorithm ranking" based on multiple objective quality results is also defined below.

Besides output speech quality, it is clear that another important aspect that should not be overlooked as part of a hierarchy of algorithm is the computational complexity – this adds another criterion often completely independent of the output quality. In fact, the computational complexity may or may not be a crucial part of the problem, depending on the available resources. For these reasons, in this thesis, each time simulation results or new algorithms are described, it is chosen to rate first the quality of their output product, and next independently discuss their computational complexity.

2.1. Objective quality measures

In this section, some explanations and references regarding the choice of the objective quality measures used in the thesis are given.

The objective quality measures used are:

- The **SNR** and the the Average segmental SNR (**ASNR**) [HAN'98],
- The Coherence Speech Intelligibility Index (**CSII**) from [KAT'05, KAT'05-2],
- The wideband extension for the PESQ score [ITU'98] (**WPESQ**), and
- The three composite measures shown in [LOI'07], meant to reflect the level of speech distortion (**Csig**), the level of background noise intrusiveness (**Cbak**), and the overall quality (**Covl**).

More details regarding these measures now follow:

- The **SNR** is not usually considered an accurate predictor of speech quality, however for its simplicity it is very widely used. It is obtained by computing the ratio of the clean signal's power to the noise signal's power. The **ASNR**, which is simply obtained as the average **SNR**s computed across overlapping frames of speech and noise signals, has been found to be mostly correlated with the level of background noise intrusiveness [HU'06].
- The **CSII** was developed as an extension of the speech intelligibility index (SII) [KAT'05], and is meant to estimate the intelligibility of speech corrupted by various nonlinear distortions – including noise removal algorithms. It is obtained by first partitioning the clean speech into 3 regions (low/mid/high level regions), and then by computing for each of them an average signal-to-distortion ratio (SDR) based on the mean-squared coherence function (as opposed to a plain SNR for the SII), across overlapping frames. Finally, the single CSII measure is calculated by weighted sums of the SDRs across the frequencies, and across the 3 regions. The final score, a number between 0 and 1 (with 0 reflecting complete unintelligibility and 1 perfect intelligibility), is found in [KAT'05, KAT'05-2] to be an accurate predictor of speech intelligibility.

- To describe the **WPESQ**, let us briefly review the PESQ (Perceptual Evaluation of Speech Quality). The PESQ algorithm was designed as an objective method to predict the results of subjective MOS tests (in a P.800 listening setup), designed purposely for handset telephony speech codecs. This is the ITU-T recommendation P.862, approved in 2001. The PESQ algorithm compares the original, clean speech signal to the output of the enhancement algorithm, and penalizes the final score based on measures of the distortion. The PESQ is perceptual in the sense that the amount of distortion is calculated in the context of a model for the human auditory system, and the predicted MOS score is obtained from an estimate of the differences in perceived loudness between the clean and degraded signals. The main difference between the PESQ and its wideband extension (WPESQ) used in this thesis begins with the pre-filtering stage: in the PESQ, the signals are prefiltered so as to model the receive path of standard telephone handsets – grossly speaking a low-pass filter, constraining the analysis to 3.4 kHz narrowband signals. The WPESQ algorithm follows the same method as the PESQ but the prefiltering handles signals up to 8 kHz, and the same analysis takes place in a wider band. Although WPESQ scores were not designed for speech enhancement algorithms evaluation, they are still found to provide a meaningful indication of overall speech quality, and they are frequently used by researchers for this purpose.
- The three composite measures **Csig**, **Cbak** and **Covl** from [HU'06] were developed by nonlinear regression of several existing objective measures so as to achieve higher correlations with subjective ratings of speech quality. The objective measures used include the Weighted Spectral Slope (WSS), the Log-likelihood Ratio (LLR), the PESQ, and the ASNR (for a description of the WSS and LLR, please see [HAN'98]). Like the PESQ/WPESQ, they return numbers between 1 and 5 (where 5 is the best

possible result), which indicate the level of speech distortion or naturalness (**Csig**), the level of background noise intrusiveness – how noticeable the background noise is (**Cbak**), and the overall quality (**Covl**).

Note that to obtain meaningful results, before applying the above algorithms the signals are first downsampled to 16 kHz for the CSII and WPESQ measures, and to 8 kHz for the composite measures.

Another issue of important is that of the perceptual significance of the reported scores. First, each score reported is rounded to one decimal, as it is safe to assume that the second decimal is, perceivably speaking, irrelevant. In fact, except for the CSII measure, even the first decimal often has limited meaning. As a rough guideline based on our subjective impressions, the following Table shows what difference in objective score translates into perceivable differences.

Score	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Min. Difference	0.3	0.2	0.1	0.1	0.3	0.3	0.3

Table 1: Minimum score differences resulting in perceivable output quality, given as a rough guideline. The above is only based on our subjective impressions and in general terms.

2.2. Speech and noise data

One of the governing ideas behind the writing of this thesis is that "all results/scores reported are compatible" across all chapters and sections: for example, anywhere in the thesis, mentions of a "5 dB noisy sentence in car noise conditions" all refer to the exact same waveform.

The wideband audio material used in this thesis has a sampling frequency of 20 kHz. The clean speech material is obtained by concatenating multiple speakers (both male and female) from the TIMIT database [GAR'90] resampled at 20 kHz, and inserting silences in order to obtain a 60% activity rate (such lower activity rates are for example recommended for objective quality estimation in [ITU'98]). For simplicity, we use a single clean speech combination of sentences – however it is 30 seconds long. In details, the spoken sentences comprised in the clean speech material are, back to back:

- "She had your dark suit in greasy wash water all year" (Female)
- "To begin with, what is an interior designer" (Male)
- "The morning dew on the spider's web glistened in the sun" (Female)
- "Only lawyers love millionaires" (Male)
- "His eyes were dark, fluid, fearful, and he gave a sigh as my knife went in" (Female)
- "Medieval society was based on hierarchies" (Male)

Next, the noise data is obtained from: http://spib.rice.edu/spib/select_noise.html, containing examples from the NOISEX-92 database [VAR'92]. The babble (also referred to in the thesis as cafeteria or canteen), car production factory, leopard (tank) military vehicle, and Volvo 340 car interior noises are used – more accurately, their first 30 seconds. These 4 types of noise are respectively referred to as **CAF**, **FAC**, **MIL**, and **CAR**.

In addition, artificial (computer-generated) white Gaussian noise is also used, and referred to as **WGN**.

In each case, the obtained noisy signals were scaled with 4 different values so as to obtain various conditions, from very low, low, medium, to high input SNR (respectively referred to with the letters **VL, L, M, H**).

In tables and in some portions of the text, the experimental conditions are then described with the above codenames according to the following examples: **VL-MIL** for "very low SNR with military vehicle noise", or **H-WGN** for "high SNR with white Gaussian noise".

For clarity, and to avoid numerous repetitions, the objective quality scores (see the previous section) for each noisy speech file are given below:

- **WGN Conditions**

	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
VL-WGN	-4.14	-7.46	0.08	1.02	0.16	1.24	0.52
L-WGN	0.07	-5.58	0.21	1.03	0.55	1.51	0.80
M-WGN	6.09	-2.07	0.59	1.06	1.19	2.01	1.32
H-WGN	12.11	1.82	0.94	1.21	1.91	2.62	1.98

- **CAF Conditions**

	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
VL-CAF	-5.56	-7.51	0.03	1.03	0.27	0.83	0.43
L-CAF	0.46	-4.81	0.35	1.05	0.72	1.13	0.74
M-CAF	6.48	-1.23	0.72	1.17	1.28	1.58	1.15
H-CAF	10.91	1.65	0.91	1.37	1.73	1.97	1.53

- **FAC Conditions**

	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
VL-FAC	-8.61	-8.51	0.06	1.02	0.62	0.92	0.65
L-FAC	-0.65	-5.35	0.62	1.06	1.25	1.30	1.09
M-FAC	3.43	-3.14	0.96	1.16	1.65	1.60	1.42
H-FAC	7.86	-0.39	0.96	1.38	2.11	1.97	1.81

- **MIL Conditions**

	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
VL-MIL	-8.28	-8.38	0.03	1.02	0.83	0.88	0.75
L-MIL	-0.32	-5.16	0.04	1.04	1.39	1.23	1.12
M-MIL	5.69	-1.64	0.95	1.12	1.87	1.64	1.50
H-MIL	11.71	2.25	0.99	1.40	2.40	2.17	1.99

- **CAR Conditions**

	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
VL-CAR	-9.33	-8.09	0.88	1.05	1.93	1.26	1.49
L-CAR	-3.31	-5.91	0.97	1.15	2.43	1.62	1.93
M-CAR	4.64	-1.61	0.99	1.60	3.10	2.24	2.60
H-CAR	10.66	2.25	0.99	2.23	3.60	2.79	3.15

2.3. Algorithms ranking method

Due not only to the high degree of subjectivity related to rating algorithms, but also to the fact that objective quality measures sometimes give contradictive impressions, it is difficult to conclude each round of simulation with a clear-cut and undebatable hierarchy of algorithms.

In this thesis, in order to try and propose a reliable indication of which algorithms appear to perform better, the following procedure is chosen:

1. First, given a set of algorithms to compare and a certain type of noise (see Chapter II), four tables regrouping all 7 objective measures results are filled (i.e., corresponding to the four input SNR conditions) and placed for clarity in Appendix (specifically Appendix B).
2. Next, within the thesis – where the testing is announced and discussed – two tables of "results summaries" are filled, with each column (i.e. each objective measure) filled with the algorithms names specifically ordered from best to lowest score (the best score being listed at the top of the column). These two tables are compiled based on the average score obtained in **very low (VL)** and **low (L)** conditions on the one hand, and in **medium (M)** and **high (H)** conditions on the other hand. Therefore, in the case of white Gaussian noise testing, since only one noise corresponds to each of **VL**, **L**, **M**, and **H** conditions, each of the two tables is obtained by averaging two sets of results. Similarly, in the case of colored noise testing, each of the two tables is

obtained by averaging eight sets of results. In order to give a view of the range of results obtained, these two ables also contain the actual averages obtained for the best and worst algorithms (see point 3 below).

3. Finally, the *n* algorithms with the most entries in the top *m* lines are named the top *n* algorithms among those tested. Further discussion is then brought in order to complete the picture with our own subjective impressions, and also to point out the possible differerences between the top algorithms at **VL/L** conditions and at **M/H** conditions. In each case, *n* and *m* will be clearly indicated.

III. Background

3.1. Speech, noise, and measurement models used in the thesis

In this section, mathematical representations of discrete speech, noise, and measurement signals are presented. These representations are at the basis of the algorithms introduced in the thesis.

3.1.1. Autoregressive signals

In most cases, signals in this thesis will be assumed to be *autoregressive* (or *linear predictive*). With $x(k)$ representing the value of a discrete stationary signal x at time k, the autoregressive model for x is:

$$x(k) = \sum_{m=1}^{M} a_m x(k-m) + \sigma_g g(k) \qquad (1)$$

where M is the model order, $\{a_m\}_{m=1}^{M}$ and σ_g are the autoregressive (AR) parameters, and $g(k)$ is a process that can be viewed as an excitation to the all-pole filter corresponding to Equation (1), or as the prediction error in the AR model (or the *residual* of the AR model). In theory, the better the prediction, the whiter the error signal $g(k)$.

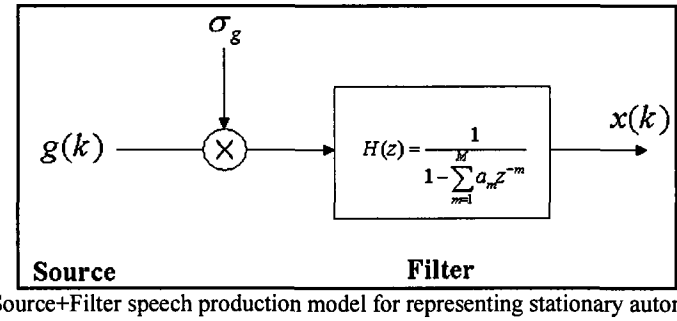

Figure 1: The Source+Filter speech production model for representing stationary autoregressive signals

When the signal is not stationary, the AR parameters are allowed to vary. In this thesis, the parameters are either constant over frames (the signal x is then assumed to be short-time stationary, over each frame), or they are set to vary at every time instant k. In either case, the AR parameters are then denoted by $\{a_m(k)\}_{m=1}^{M}$ and $\sigma_g(k)$. The type and rate of variation will be clearly specified when needed. Such autoregressive models with time-varying parameters are often termed "time-varying autoregressive", or TVAR.

While being extremely simple, AR modelling has proven to be both theoretically and practically founded when applied to speech signals. Theoretically, speech can be viewed as the product of a glottal excitation (originating from a flow of air in the lungs) of the vocal cords: $g(k)$ can then represent this glottal excitation, while the all-pole filter corresponding to Equation (1) mimics the configuration of the vocal cords. Practically, successful speech AR modelling has been achieved in many contexts, with a striking example being speech coding, where $g(k)$ usually takes different forms of train pulses (more regularly spaced during voiced segments), or is modelled by white Gaussian processes, etc[1]. Except in Chapter X, containing future work ideas, the signal $g(k)$ is in this work assimilated to a zero-mean unit variance white Gaussian noise, for convenience and simplicity.

3.1.2. Generic speech production model

In this thesis, the speech production phenomenon is represented according to the description contained in this section. Broadly speaking, first a speech model order M_s is chosen: denote

[1] Examples of viable excitation signals for speech coding/synthesis are investigated in [CHU'03, GOL'00].

by $\mathbf{x}(k) = [x(k)\ x(k-1) \cdots x(k-M_s+1)]^T$ the column vector containing a trail of M_s speech samples up to $x(k)$. The speech production is then assumed to follow:

$$\mathbf{x}(k) = \Phi_k(\mathbf{x}(k-1)) + \mathbf{G}(k)\mathbf{w}(k) \tag{2}$$

where $\Phi_k(.)$ is a possibly nonlinear, multidimensional function, $\mathbf{G}(k)$ is an M_s by M_s matrix with $\mathbf{G}(k)_{[1,1]} = \sigma_g(k)$ (as defined in Equation (1)) and all other components set to 0, and $\mathbf{w}(k)$ is an M_s-dimentional random vector whose only first entry matters (due to the special form of $\mathbf{G}(k)$), and is equal to $g(k)$ (which has been encountered in Equation (1)). Thus, Equation (2) is a generic predictive model, with prediction function $\Phi_k(.)$.

In this thesis, two cases are explored for the prediction function: the first one is the case where $\Phi_k(.)$ is linear, and the production model reduces to the AR model explained in section 3.1.1. Then, placing the AR vector $\{a_m(k)\}_{m=1}^M$ in a matrix $\mathbf{A}_k$ in canonical form, Equation (2) can be rewritten as:

$$\mathbf{x}(k) = \mathbf{A}_k\mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k) \tag{3}$$

In the second case, the prediction function mimics that of a feedforward Neural Network, as in Equation (4) below:

$$x(k) = \mathbf{c}(k)^T f(\mathbf{W}(k)\mathbf{x}(k-1) + \mathbf{d}(k)) + \sigma_g(k)g(k) \tag{4}$$

where:

- P is the number of neurons in the model,
- $\mathbf{W}(k)$ is a P by M_s matrix, representing the internal coefficients of the network,
- $\mathbf{c}(k)$ is a length P column vector denoting the output coefficients of the network,

- $\mathbf{d}(k)$ is a length P column vector denoting the bias inputs at each of the neurons in the network,
- $f(.)$ is the nonlinear activation function of the neurons

The reason for this second choice is the following: While the use of linear prediction models has been shown to produce satisfactory results in many applications, there is strong theoretical and experimental evidence that the phenomenon of speech production contains important nonlinearities ([FAU'02, THY'94, MA'98]). Concretely, a case for the use of nonlinear prediction models is presented in [CHE'05], where it is conjectured from theoretical considerations that the introduction of nonlinearities in the speech model should only be beneficial. Indeed, it has been reported that nonlinear models potentially achieve a higher prediction gain than traditional linear predictive models of the same order (see e.g. [CHE'05], [LIZ'94], and for a reference on linear prediction, see [MAK'75]).

3.1.3. Noise model

The noise signals are also modelled as autoregressive (see Section 3.1.1). Notationwise, let $n(k)$ denote the noise signal at time instant k and let M_n be the order of the noise AR model. The model for $n(k)$ is:

$$n(k) = \sum_{m=1}^{M_n} b_m(k) n(k-m) + \sigma_v(k) v(k) \tag{5}$$

where $\mathbf{b}(k) = \{b_m(k)\}_{m=1}^{M_n}$ and $\sigma_v(k)$ are the AR parameters of the noise, and $v(k)$ is a zero-mean unit variance white Gaussian noise (i.e., the excitation noise of the noise AR model).

3.1.4. Measurement model

The measurement at time instant k, denoted by $z(k)$, is simply the sum of $x(k)$ and $n(k)$, as in Equation (6) below:

$$z(k) = x(k) + n(k) \tag{6}$$

Therefore, this thesis is primarily concerned with the removal of additive noise. Naturally, one might object that in "real-world situations", it is quite common to encounter non-additive noises such as mitigating convolutive noise. As prime examples of non-additive noise, echo and reverberation are often present in real-world recordings and can reduce intelligibility just as much as additive noise.

As a first answer to the above concern, one could envision a two-stage enhancement system, in which the pre-processor could consist of one of the proposed methods in this thesis to remove the "additive" part of the noise. Then, a dedicated dereverberation and/or echo-removal algorithm could be use in cascade.

Secondly, as it will be noted later on in this thesis, the capacity of the algorithms proposed to track different types of noise is determined by that of dedicated noise power spectral density (PSD) estimators. Regarding the problem of reverberation, it is possible to specifically use *diffuse* noise PSD estimators (see for example the one proposed in the binaural domain in [KAM'09]), where much of the reverberation tail can actually be tagged as diffuse noise and thus removed in an additive noise model context.

3.2. State-space based algorithms

3.2.1. On the naturalness of state-space based algorithms

The claim that state-space algorithms better retain naturalness than frequency-domain methods has been documented in [BEN'05], containing subjective tests as well. One pertinent quote from the above reference would be the following: "*Kalman filter based algorithms are shown to maintain the natural speech quality. However, their noise reduction ability is limited.*"

In [GAN'98], this is also observed. The justifications for this claimed naturalness is based on the following facts:

- the absence of musical artefacts such as the ones observed in many frequency-domain enhancement algorithms (see Section 3.4). The musical artefacts can be associated with the flooring in the magnitude or power spectra that must occur to prevent negative values. The flooring results in remaining isolated patches of energy and "holes" in the spectrum. For the listener, the succession of seemingly randomly placed short-lived patches of energy sounds like random tones and is very unnatural. By nature, state-space based methods will not suffer from this effect since they operate in the time-domain on a sample-by-sample basis.
- With sample-by-sample treatment, the algorithms are able to continually adapt to the incoming signal, and thus are less prone to distorting and damaging onsets and offsets of speech. For example, in typical frequency domain methods, if a speech onset occurs at the end of a noisy frame, it might disappear from the reconstructed speech.

3.2.2. Kalman, Extended Kalman, Unscented Kalman, and Particle filters

The algorithms shown in this section are central tools for the enhancement methods of the thesis. More generally, they all provide solutions to the so-called sequential state estimation problem, with distinct advantages and drawbacks depending on the situation.

In this section, the state to estimate is denoted by $\mathbf{x}_k$ at time k, but it does not necessarily coincide with the clean speech vector $\mathbf{x}(k)$ defined in section 3.1.2 – depending on the case, $\mathbf{x}_k$ can actually contain other variables than just the speech itself. For simplicity, we keep $\mathbf{x}_k$ as the name of the generic state and if needed, we later on specify which algorithm is referred to and what exactly is contained in the state vector.

The generic sequential state estimation problem, given noisy measurements, can be summarized in state-space form as in the following coupled equations:

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{w}_k) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k) \end{aligned} \tag{7}$$

where again, $\mathbf{x}_k$ is the state to estimate, $\mathbf{z}_k$ is the measurement, $\mathbf{w}_k$ is the process noise, and $\mathbf{v}_k$ is the measurement noise. Functions f and h are assumed to be known, and may depend on time. At each time instant k, the goal is obtain an estimate $\mathbf{x}_k$ which minimizes the squared error (with respect to the true value) based on incoming and past measurements. For a continuous state vector, in general there is no closed-form (exact), optimal (in the mean-squared sense) solution to this problem. Several particular cases can however be handled with the algorithms below.

3.2.2.1. The Kalman Filter (KF)

When f and h are linear, $\mathbf{w}_k$ and $\mathbf{v}_k$ are independent and Gaussian, then there exists a closed form and optimal solution: the Kalman Filter (KF). In such cases, the equations shown in (7) can be rewritten as:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{H}_k\mathbf{v}_k \end{aligned} \tag{8}$$

where the matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$, and the vector $\mathbf{u}_k$ are known (arbitrary). In this case, all densities are Gaussian, and therefore it is only required to update the mean and covariance matrix of $\mathbf{x}_k$ as the measurements arrive – and this update is simple. It is given in algorithmic form in Algorithm 1 in Appendix A, which also contains most algorithms used in this thesis. Note that in Algorithm 1, the output "weight" is not used in regular KF applications, but it will be used later in Particle Filtering algorithms.

Again, in this linear and Gaussian case there is no reason to use any other algorithm if the mean-square error is the appropriate function to be minimized, since the KF is optimal in the mean-square sense.

3.2.2.2. The Extended Kalman Filter (EKF)

The EKF is an approximate solution, intended to be used when f and h are nonlinear in $\mathbf{x}_k$ only, and $\mathbf{w}_k$ and $\mathbf{v}_k$ are still independent and Gaussian. With an abuse of notation for f and h[2], we can write:

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{w}_k \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{D}_k\mathbf{u}_k + \mathbf{H}_k\mathbf{v}_k \end{aligned} \tag{9}$$

[2] The "abuse" stems from the fact that the functions were previously defined with multiple arguments

Basically, in the EKF algorithm f and h are linearly approximated about the current state estimate, effectively "replacing" when necessary the matrices $\mathbf{A}_k$ and $\mathbf{C}_k$ from Equation (8) by the Jacobian matrices of f and h at the most recent state estimate (i.e., the first order Taylor approximation of f and h, or the "slope" or "tangent" in one dimension). Therefore, the approximation can only be reasonable if f and h are not highly nonlinear.

Formally, the matrices $\mathbf{A}_k$ and $\mathbf{C}_k$ to be used are $\mathbf{A}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{k-1}}$ and $\mathbf{C}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=f(\mathbf{x}_{k-1})+\mathbf{B}_k\mathbf{u}_k}$, and the resulting algorithm is shown in Algorithm 2 found in Appendix A.

For the most part, the limitation of the EKF comes from the fact that it can diverge depending on the situation [DOU'01], however some good performance can in general be achieved provided a reliable initial estimate is given. In addition, because of the nonlinearities, all densities are not necessarily truly Gaussian, but the EKF treats them as such and only propagates a mean and a covariance matrix estimate.

3.2.2.3. The Unscented Kalman Filter (UKF)

The UKF applies to the same subcase as the EKF, although it is in fact able to handle non-Gaussian noise processes by nonlinear transformation (i.e., the function f can be nonlinear in both the state and the noise). This time however, instead of linearizing the system in order to propagate the mean and covariance estimates, a special deterministic sampling method is used to propagate a set of so-called *sigma-points*, whose mean and covariance matrix more accurately follow the true ones when passed through the nonlinearities. The central method that dictates the choice of the sigma-points is called the Unscented Transform [JUL'97].

An implementation is shown in Algorithm 3 in Appendix A (note: the subroutine "`sigma_points`" is not shown for clarity – its purpose is to compute the scaled symmetric sigma points and their weights (see [JUL'97] for details). It is easily implementable in less than 10 lines of MATLAB code).

While some significant improvements over EKFs have been reported, the UKF is still limited by the fact that all densities are assumed to be Gaussian – including the posterior density $p(\mathbf{x}_k \mid z_0, z_1, \ldots, z_k)$. This assumption does not hold for many complex real-world situations (with multi-modal distributions and non-Gaussian noises).

3.2.2.4. The Particle Filter (PF)

With particle filters, a discrete weighted approximation of the filtering (or smoothing) density is propagated at each iteration using Monte-Carlo simulation [DOU'01]. The assumptions are the broadest of all the filters above (f and h are nonlinear, $\mathbf{w}_k$ and $\mathbf{v}_k$ are independent but possibly non-Gaussian), and the weighted approximation of the densities is "asymptotically optimal" (i.e., as the number of points in the discrete approximation tends to infinity). Each discrete point in the representation of the posterior is called a *particle*, hence the name of the filter.

PFs are however more of a framework and some derivations must take place depending on the state dynamics and the noise assumptions to make them directly applicable. In addition, regardless of the situation, PF algorithms depend on the choice of an *importance function*,

which is a density from which the particles are drawn, and also directly impacts the algorithm derivation. The above is as much an advantage (in terms of flexibility for example) as it is a roadblock (as not every set of assumptions admit closed-form particle filtering equations). A generic PF algorithm is shown in Algorithm 4 in Appendix A, in pseudo-code format. The importance function is denoted by $q(.)$, and can be arbitrary in theory (although in practice, choices are often limited for the derivation to be possible).

While PFs have been reported to perform better than EKFs or UKFs in several situations, they are also the most complex in terms of the amount of computations required. Indeed, it is often the case that many particles are required to achieve a good performance and/or robustness.

3.2.2.5. The Rao-Blackwellized Particle Filter (RBPF)

The RBPF (a.k.a. the Kalman Particle Filter) is an "enhanced" particle filter, to be used when some components of the state-vector $\mathbf{x}_k$ evolve according to linear equations with only Gaussian noises ("linear Gaussian" equations such as (8)), conditioned upon the other state components [DOU'01]. More specifically, supposing that $\mathbf{x}_k = [\mathbf{x}_{1k}; \mathbf{x}_{2k}]$ is such that when $\mathbf{x}_{1k}$ is known, $\mathbf{x}_{2k}$ is found to follow linear-Gaussian equations and thus can be updated analytically via a simple Kalman Filter. As a result, in an RBPF, not only must $\{\mathbf{x}_{1k}^{(i)}\}_{i=1}^{N}$ and $\{\mathbf{x}_{2k}^{(i)}\}_{i=1}^{N}$ be updated (as in a regular PF), but also the covariance matrices of $\{\mathbf{x}_{2k}^{(i)}\}_{i=1}^{N}$ (as in a regular KF) which we denote by $\{\mathbf{P}_k^{(i)}\}_{i=1}^{N}$.

Formally, assuming that the knowledge of $\mathbf{x}_{1k}$ completely determines the matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$ and the vector $\mathbf{u}_k$, and that $\mathbf{x}_{2k}$ follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{C}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{u}_k + \mathbf{H}_k \mathbf{v}_k \end{aligned} \tag{10}$$

then the generic corresponding RBPF algorithm can be written as in Algorithm 5 (in Appendix A).

The RBPF procedure has been shown to sharply reduce the variance of the error estimates in theory, and this usually translates into better performance in practice. As a by-product, the amount of particles required for a given accuracy is in general lower than for a regular PF, to the point that overall, less computations are needed even though more computations are typically required per particle [KAR'05].

3.2.3. The central idea of state-space based algorithms for speech enhancement

Broadly speaking, all the state-space based algorithms used, devised, and referenced in this thesis revolve around the following central idea for speech enhancement:

1. Write a state-space model describing the speech, noise, and measurement, as done in Section 3.1.
2. Devise a method to estimate the speech (AR or NN) and noise AR parameters over time
3. Use one of Algorithm 1 to 5 to sequentially estimate the corresponding clean speech signal

As it will be seen in details in Section 3.2.4, the estimation of speech parameters in step 2 above can be carried out with separate iterative schemes (e.g. gradient-based or Expectation-Maximization based algorithms), with additional Kalman Filters (forming what is commonly called a "dual" method), or "internally" as in the case of PFs or RBPFs.

For the noise parameters estimation, in this thesis it is chosen to devise ways to exploit the information returned by completely independent background noise estimation algorithms as opposed to voice activity detectors (VADs). The main reason for this choice is that while VADs may perform well in low-complexity, stationary noises, their reliability quickly breaks down in realistic noisy environments [RAN'06]. In addition, background noise estimation is a fairly mature field, with various existing algorithms, and is currently evolving in the binaural domain (more details will be shown in Section 3.3).

3.2.4. Core algorithms used in the thesis

The following algorithms make use of the speech and noise models described in Section 3.1. In this section, it is assumed that the noise AR parameters (see Section 3.1.3) are known at any time instant. The type of categorizing used is the fruit of an extensive literature review.

As mentioned in the introduction, most state-space based algorithms found in the literature are not presented as being ready to deal with real-world noises. The noise types that are in general tackled are the lesser complex ones, such as stationary white Gaussian with known variance, stationary colored with either known statistics, or very good initialization and/or accurate voice-activity-detection (see for example Chapter 8 in [BEN'05]). Accordingly, realistic simulation results in such real-world noise conditions are scarce. Some methods to handle completely unknown and possibly highly nonstationary noises will be shown in subsequent chapters, but these methods still make use of the following core algorithms with a few modifications and/or context changes.

3.2.4.1. Optimization-based methods

In this category of algorithms, the clean speech parameters are estimated via optimization methods such as the Expectation-Maximization algorithm or with gradient descents. The speech models used are essentially linear.

First, several methods [GAN'98, PAR'06] assume that the speech parameters are constant over short frames, and follow the procedure outlined below:

1. Initialize a Kalman Filter and the speech parameters with any a priori knowledge available (e.g., knowledge from previous segments)
2. Given the current speech parameters estimates, run the KF on the noisy segment, obtaining a clean speech estimate along with its error covariance.
3. Given the current clean speech estimate, apply an optimization method to update the speech parameters
4. Go back to 2 until convergence criterion is met

In [GAN'98], basically the Yule-Walker equations are solved using the posterior correlation values returned by step 2. It is shown that doing so is equivalent to applying an Expectation-Maximization algorithm. Indeed, the E-step (Expectation step) is precisely equivalent to running the KF (since the KF yields the expected value of the speech given the measurement – and in this case conditioned upon the speech parameters). Next, with the information updated by the KF, it is shown that the modified YW equations maximize the likelihood of the speech AR parameters. This algorithm will be referred to as **KEM** in this thesis.

Essentially, [GIB'91] follows the same course of action except that the speech AR parameters are found using ordinary YW equations, which basically amounts to performing several "KF

passes", which [GAN'98] shows to be detrimental to the performance. Moreover, broadly speaking the more complex algorithm shown in [PAR'06] also follows the same outline, except that the excitation noise in the speech signal is Generalized Exponential and not Gaussian, so that a plain KF cannot be used anymore and a RBPF is used instead. Also, the development of the M-step optimization is further constrained with the positivity of the excitation noise variances. While partial results show some improvements, the complexity is also very much increased.

To the best of our knowledge, there appears to be an "untapped" line of speech AR parameters update following the procedure above: indeed, rather than a variant of the Yule-Walker method, one could use one of the many existing AR parameters estimation, such as the Burg method, the covariance method (and its modified counterpart), the unconstrained least squares method, etc. Moving away from the YW method has in fact been repeatedly encouraged in the literature [HOO'96, MAK'75]. Some configurations will be tested in the next section.

Next, in [WEI'00-2, WEI'94] an alternative is proposed which avoids the use of segments over which the speech parameters are assumed to be constant, and instead a sample-by-sample gradient descent of the Maximum-Likelihood function (the minimization of which being also the objective of the KEM algorithm in [GAN'98]) is derived. The resulting algorithm is more efficient computationally and is found to yield similar results at low SNR, but does not perform as well in high SNR conditions. It will be referred to as **KGD** hereafter.

3.2.4.2. Dual methods

In these methods, on top of the state space-based algorithms estimating the clean speech given some speech AR parameters estimates, another state space-based algorithm is run in parallel on a system whose goal is to obtain the speech parameters given the clean speech. In essence, this is very close to what is already presented in Section 3.2.4.1, except the algorithms used to update the speech parameters are of the same nature as those used to clean the noisy speech signal, and operate on the same equations, although from a different angle.

For the sake of example, suppose here that the environment noise is stationary white Gaussian, and the speech model is linear. Following the notation shown in Section 3.1, the state space-model that is used for the speech enhancement algorithm is:

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}_k \mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v v(k) \end{aligned} \tag{11}$$

where $\mathbf{C} = [1\,0 \ldots 0]$. If the AR parameters are assumed to be given (i.e., if $\mathbf{A}_k$ and $\mathbf{G}_k$ are known), and if the noise variance σ_v is known as well, then a simple KF can be used to estimate $\mathbf{x}(k)$. Next, another state-space model centered on the estimation of the AR vector $\mathbf{a}(k)$ is written as:

$$\begin{aligned} \mathbf{a}(k) &= \mathbf{a}(k-1) + \Gamma_a \alpha(k) \\ z(k) &= \mathbf{x}(k-1)^T \mathbf{a}(k) + \sigma_g(k) g(k) + \sigma_v v(k) \end{aligned} \tag{12}$$

where $\Gamma_a = \sigma_a \mathbf{I}_M$ and σ_a is a predetermined constant, and $\alpha(k)$ is a zero-mean unit variance Gaussian process. Thus, the additional assumption that is introduced via the first equation in (12) above is that the AR vector $\mathbf{a}(k)$ is "drifting" in time according to a Gaussian random walk with predetermined variance. Observe that in the second measurement equation, since $g(k)$ and $v(k)$ are independent, the observation noise is still white Gaussian with

variance $\left(\sigma_g(k)^2 + \sigma_v{}^2\right)$. Therefore, provided $\mathbf{x}(k-1)$ and $\sigma_g(k)$ are known, another KF can be used as well. While an estimate of $\mathbf{x}(k-1)$ can be returned by the first KF, $\sigma_g(k)$ must be determined by other means. Several methods can be found for this purpose in the literature [GIB'91, NEL'97, WAN'98, CHA'07]; each revolving around the same idea: For example, $\sigma_g(k)$ can be deduced from a Yule-Walker analysis on the most recently enhanced few samples of clean speech (i.e. calculating the MSE of a linear prediction on estimated clean data).

This type of method, employing two similar algorithms with intertwined state-space equations, is called "dual". In the example above, both KFs must continually exchange information. Following this idea, several dual configurations can be obtained (including with colored noise, for which the second state space algorithm can estimate both the speech and the noise parameters together), as recapitulated below. For a linear speech model, there is no reason to use different algorithms than plain KFs, obtaining the **DKF** as named hereafter. For nonlinear models however, several choices are possible, with for example the so-called Dual EKF (**DEKF**), used for example in [WAN'98] or the Dual UKF (**DUKF**), used for example in [MA'04] (albeit with a slight variation). One can also think of solutions with two concurrent PFs or RBPFs, which is the subject of Chapter VI.

One of the most important drawbacks affecting dual algorithms is that, in our experience, they are more vulnerable to instability, and more sensitive to initialization. This stems from the fact that if one of the two algorithms is suddenly "off" in its estimates, there can be some direct and damaging impact on the other one, which then worsens the situation as the information comes back, and so on until the overall error becomes unmanageable.

3.2.4.3. Holistic methods

In this category of algorithms, the approach consists in using a single state space-based algorithm to sequentially estimate not only the clean speech but also all required unknowns in the speech model. One of the theoretical advantages of this approach is that the convergence properties of the single filter as a whole remain applicable, while it is not the case when two filters are run in parallel.

To be able to write a complete single state-space model for the speech production, evolution models for the AR parameters (or for the NN weights in the nonlinear case) are necessary – including the excitation noise variance $\sigma_g^2(k)$. In Equation (12), a transition model for the AR vector $\mathbf{a}(k)$ was already introduced. The same type of Gaussian random walk cannot be applied to the variance of the excitation noise $\sigma_g^2(k)$, because it is required to stay positive. As a solution, a random walk on the log-variance of $\sigma_g(k)$, denoted by $\ell_g(k) = \log(\sigma_g(k))$, is proposed, and the complete set of equations becomes:

$$\begin{aligned} \mathbf{a}(k) &= \mathbf{a}(k-1) + \Gamma_a \alpha(k) \\ \ell_g(k) &= \ell_g(k-1) + \sigma_{\ell_g} \beta(k) \\ \mathbf{x}(k) &= \mathbf{A}_k \mathbf{x}(k-1) + \mathbf{G}_k \mathbf{w}(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v(k) v(k) \end{aligned} \tag{13}$$

In the UKF-based "joint" solution proposed by [GAN'03], while not clearly indicated in the paper, a slight change must be made to the system to reach a single state-space model operating on the joint state vector $\chi(k) = \begin{bmatrix} \mathbf{x}(k)^T & \mathbf{a}(k)^T \end{bmatrix}^T$: the transition matrix $\mathbf{A}_k$ must contain the past vector $\mathbf{a}(k-1)$ and not the current one. Moreover, it is still assumed that $\ell_g(k)$ is estimated separately. In that case, the global state-space model can be written as:

$$\begin{aligned}\chi(k) &= \begin{bmatrix} \mathbf{A}_k & 0_{M\times M} \\ 0_{M\times M} & I_{M\times M} \end{bmatrix}\chi(k-1) + \begin{bmatrix} \mathbf{G}_k & 0_{M\times M} \\ 0_{M\times M} & \Gamma_a \end{bmatrix}\begin{bmatrix} \mathbf{w}(k) \\ \alpha(k) \end{bmatrix} \\ z(k) &= \begin{bmatrix} \mathbf{C} & 0_{1\times M} \end{bmatrix}\chi(k) + \sigma_v(k)v(k)\end{aligned} \tag{14}$$

This system is however nonlinear because the matrix $\mathbf{A}_k$ contains part of the state $\chi(k-1)$. [GAN'03] therefore proposes an UKF to solve it, preferring it over the EKF. This algorithm is codenamed **JUKF**.

In another family of joint or holistic methods, PF-based methods have recently been receiving increased attention due to their reportedly strong performances in white environment noise. In this case, we first reassert that the transition matrix $\mathbf{A}_k$ contains the current vector $\mathbf{a}(k)$, once again supposing that the system is linear. Conditioned upon the knowledge of the speech parameters (both $\mathbf{A}_k$ and $\sigma_g(k)$), the state space model for the speech production is linear, with only Gaussian noises intervening. Therefore, a RBPF can be used (and should be used rather than a plain PF) following the equations:

$$\begin{aligned}\mathbf{a}(k) &= \mathbf{a}(k-1) + \Gamma_a\alpha(k) \\ \ell_g(k) &= \ell_g(k-1) + \sigma_{\ell_g}\beta(k) \\ \mathbf{x}(k) &= \mathbf{A}_k\mathbf{x}(k-1) + \mathbf{G}_k\mathbf{w}(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v(k)v(k)\end{aligned} \tag{15}$$

The two first lines in the equations above contain the parts in the state vector that are sampled from an importance distribution (matching $\mathbf{x}_{1k}$ in Algorithm 5), and then the corresponding speech signal (matching $\mathbf{x}_{2k}$ in Algorithm 5) is updated exactly by Rao-Blackwellization. Using the transitional prior distribution for $\mathbf{a}(k)$ and $\ell_g(k)$ (i.e., independent Gaussian random walks on each of them), we obtain an algorithm called $\mathbf{RBPF_{Lin}}$ below. Chapter V details the derivation of possible PF solutions in the context of nonlinear speech production models.

3.2.5. Performance and simulation results in white Gaussian noise environment

The goal of this section is to get a view of the strengths of each algorithm, using the objective quality assessment measures presented in Chapter II. These strengths will be based on the results obtained in zero-mean white Gaussian noises environments, and are still very pertinent to the subject and the rest of the thesis since several real-world noise cases can be seen as an extension of the white Gaussian noise (WGN) case. The detailed experimental conditions (corresponding to the WGN case) are given in Chapter II.

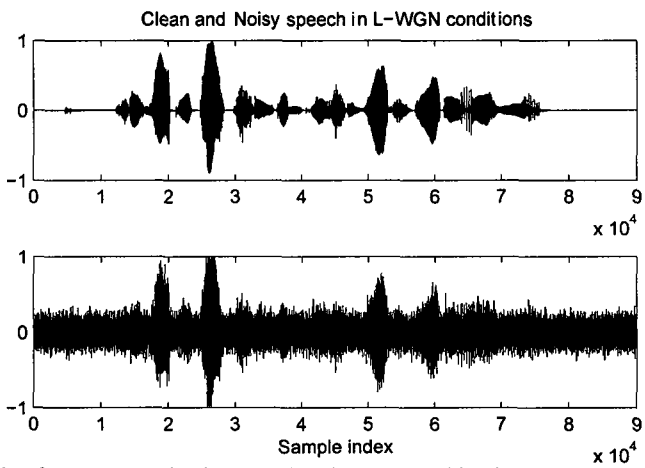

Figure 2: Example of a segment of noisy speech to be processed by the state space-based algorithms. Specifically, this segment was obtained from the L-WGN conditions defined in Chapter II.

Since there are many results to present from many algorithms, successive comparisons will be made between algorithms from each category, that is, optimization-based, dual, and finally holistic. The top algorithms will then be globally compared at the end of the section. Note that such global, cross-testing has not been reported in the literature, and thus this subsection can be viewed as a contribution in this thesis.

3.2.5.1. Comparisons between Dual algorithms

The DKF runs on a linear model of order 15, and the DEKF and DUKF algorithms run on a feedforward neural network of the same order (by "order", we mean "size of the input speech vector"). For the DEKF/DUKF, two cases are tested: a single-neuron case, and a multiple-neurons case. For the multiple neurons case, 4 hidden neurons are used. All neurons are activated using the "*tanh*" function. For all of these algorithms, two flavours are explored: a "sample-by-sample" parameter estimation, and a "frame-based" parameter estimation. In the sample-by-sample type, the speech parameters are updated once at each iteration. In the frame-based case, over a given frame the mean and covariance speech parameters are initialized with those from the previous frame, and the KF/EKF/UKF then performs several "passes" using the current data, assuming that the parameters are locally constant (this idea is mentioned in [NEL'97]). At the end of one pass (i.e., when all the noisy samples in the frame have been used), the algorithm goes through the same data again but replacing the mean and covariance of the speech parameters (but not of the estimated clean speech) by the ones most recently obtained. The frame-based dual algorithms will be denoted by $\mathbf{DKF_f}$, $\mathbf{DEKF_f}(P)$, and $\mathbf{DUKF_f}(P)$ (where P is the number of neurons used), and those without frames by **DKF**, **DEKF**(P), and **DUKF**(P). In our implementation, we find that in the frame-based cases 10 passes are sufficient for the speech parameters to converge.

For complete details regarding experimental conditions, objective measures and algorithms ranking procedures, please refer to Chapter II. The results are summarized in Tables 2 and 3 below, using the complete results given in Appendix B, specifically in B.1.1.

VL/L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**$DUKF_f(1)$**	DUKF(1)	**$DUKF_f(1)$**	$DEKF_f(4)$	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**
	DUKF(1)	**$DUKF_f(1)$**	$DEKF_f(4)$	**$DUKF_f(1)$**	DUKF(1)	DUKF(1)	DUKF(4)
	$DEKF_f(4)$	$DEKF_f(4)$	DUKF(4)	DUKF(4)	DUKF(4)	$DEKF_f(4)$	DUKF(1)
	DUKF(4)	DEKF(4)	$DUKF_f(4)$	DEKF(4)	$DUKF_f(4)$	DUKF(4)	DEKF(4)
	DEKF(4)	DUKF(4)	DUKF(1)	DUKF(1)	DEKF(4)	DEKF(4)	$DEKF_f(4)$
	$DEKF_f(1)$	DKF_f	DEKF(4)	$DUKF_f(4)$	$DEKF_f(4)$	$DUKF_f(4)$	$DUKF_f(4)$
	DKF_f	$DEKF_f(1)$	DKF_f	$DEKF_f(1)$	DKF_f	DKF	$DEKF_f(1)$
	DEKF(1)	DKF	DKF	DKF_f	$DEKF_f(1)$	DKF_f	DKF_f
	DKF	DEKF(1)	$DEKF_f(1)$	DEKF(1)	DKF	$DEKF_f(1)$	DKF
Worst	$DUKF_f(4)$	$DUKF_f(4)$	DEKF(1)	DKF	DEKF(1)	DEKF(1)	DEKF(1)
Noisy	-2.04	-6.53	0.15	1.03	0.36	1.37	0.66
$DUKF_f(1)$	7.68	-0.78	0.50	1.09	0.62	1.70	0.83
DEKF(1)	7.02	-0.95	0.32	1.08	0.40	1.62	0.68

Table 2: Dual algorithms ranking for VL/L WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top three lines) is shown in bold. The bottom two rows disclose the scores obtained by the best and worst ranked algorithms.

M/H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**	**$DUKF_f(1)$**
	DUKF(4)	DUKF(4)	$DUKF_f(4)$	$DUKF_f(4)$	$DUKF_f(4)$	DUKF(4)	DUKF(4)
	$DUKF_f(4)$	$DUKF_f(4)$	DUKF(4)	$DEKF_f(4)$	DUKF(4)	DUKF(1)	$DUKF_f(4)$
	$DEKF_f(4)$	$DEKF_f(4)$	DUKF(1)	DEKF(4)	DUKF(1)	$DUKF_f(4)$	DUKF(1)
	DUKF(1)	DUKF(1)	DEKF(4)	DUKF(4)	$DEKF_f(4)$	$DEKF_f(4)$	$DEKF_f(4)$
	DEKF(4)	DEKF(4)	$DEKF_f(4)$	$DEKF_f(1)$	DEKF(4)	DEKF(4)	DEKF(4)
	DKF_f	DKF_f	$DEKF_f(1)$	DUKF(1)	$DEKF_f(1)$	DEKF(1)	DEKF(1)
	$DEKF_f(1)$	DKF	DKF_f	DEKF(1)	DEKF(1)	$DEKF_f(1)$	$DEKF_f(1)$
	DKF	$DEKF_f(1)$	DKF	DKF_f	DKF_f	DKF_f	DKF_f
Worst	DEKF(1)	DEKF(1)	DEKF(1)	DKF	DKF	DKF	DKF
Noisy	9.10	-0.13	0.77	1.14	1.55	2.32	1.65
$DUKF_f(1)$	15.75	4.62	0.95	1.39	1.92	2.65	1.96
DKF	14.23	3.92	0.91	1.30	1.61	2.48	1.68

Table 3: Dual algorithms ranking for VL/L WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top three lines) is shown in bold. The bottom two rows disclose the scores obtained by the best and worst ranked algorithms.

In the **VL/L** conditions, the best rated 3 algorithms (with the most entries in the top 3 lines) are the $DUKF_f(1)$, the $DEKF_f(4)$, and the DUKF(1). For the **M/H** conditions, they are the $DUKF_f(1)$, the $DUKF_f(4)$, and the DUKF(4). Similarly, the worst 3 algorithms are the DEKF(1), the DKF, and the $DEKF_f(1)$ for the **VL/L** conditions, and the DKF, the DKF_f, and the DEKF(1) for the **M/H** conditions.

In Tables 2 and 3, note from the two bottom rows that the range of scores obtained is not very large. From the numbers only, a reasonable conjecture is that all of these dual algorithms are very close in terms of perceivable differences. As confirmed from the examination of the

complete Tables in Appendix B.1.1, the advantages of using frames are marginal as well – we conclude that they are not worth the extra significant amount of computations.

Indeed, from our subjective impressions of these results, all of these algorithms yield very comparable results and it is not obvious to tell them apart. Interestingly, the basic **DKF**, for which the two KFs themselves are very simple and which runs significantly faster than all of the other types of algorithms, yields very comparable results. Another interesting result is that in the nonlinear models, very little is to be expected from using multiple neurons as opposed to a single neuron, and in some cases the results are actually worse than when a simple **DKF** is used.

Only in the case of the dual UKF running on frames can we see some non-marginal benefits of employing nonlinearities, but only so at high SNR. However, the **DUKF** is also the most demanding of these dual algorithms, and therefore the computational overhead is difficult to justify. This is especially the case for the "top" algorithm here, which resorts to multiple passes per frames and therefore increases greatly the required amount of computations.

Our conclusion is that from a practical/application standpoint, the **DKF** turns out to be a very good option, since for a fraction of the computational cost, the enhanced product is very close (subjectively speaking) to the other algorithms tested. An example of the segment of speech enhanced by the DKF algorithm is shown in Figure 3.

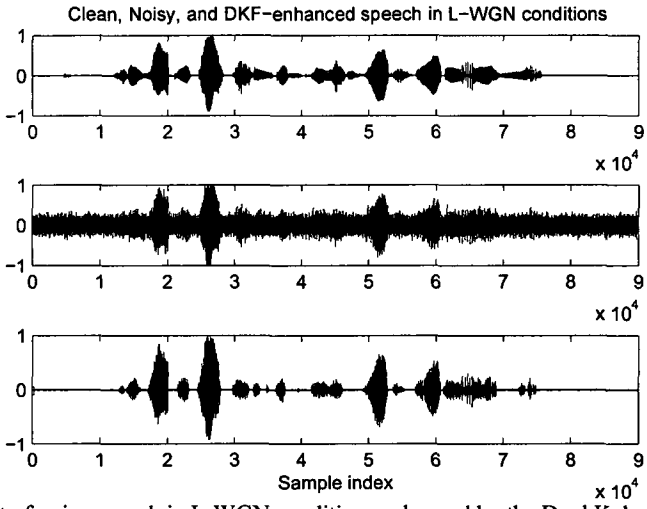

Figure 3: A segment of noisy speech in L-WGN conditions enhanced by the Dual Kalman Filter algorithm.

3.2.5.2. Comparisons between optimization-based algorithms

In these paragraphs, the following algorithms will be tested: the **KEM** and the **KGD** algorithms described in Section 3.2.4.1, and also modifications of the **KEM** algorithms where the **M**-step (i.e., the YW estimation of the AR parameters) is replaced by an update using the Burg, Covariance, and Modified Covariance methods to update the AR parameters of the speech. The last three methods will be denoted by $\mathbf{KEM}_{\mathbf{Burg}}$, $\mathbf{KEM}_{\mathbf{Cov}}$, and $\mathbf{KEM}_{\mathbf{Mcov}}$ below (note that these three methods are minor contributions, as we have not been able to find them in the literature).

Aside from the **KGD** algorithm, which only updates the speech parameters once at each time instant, the other algorithms are set to iterate 8 times per frame. For the **KGD** algorithm, we had initially noted poor convergence properties – especially at low SNR – also clearly reported by [BEN'05]. To circumvent these issues, smaller step sizes were used, and also regular "resets" of the speech excitation noise variance estimate were applied via the YW method on the clean speech estimated in the previous frame. Such implementation details can

be analyzed by the reader in our available MATLAB code at http://www.site.uottawa.ca/~bouchard/papers/mustiere_thesis.zip

The results are shown in Tables 4 and 5 below. Once again, for complete details regarding experimental conditions, objective measures and algorithms ranking procedures, please refer to Chapter II.

VL/L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	KEM_{Burg}	**KEM_{Mcov}**	**KEM_{Mcov}**	**KEM_{Mcov}**	KEM	**KEM_{Mcov}**	KEM
	KEM_{Mcov}	KEM_{Cov}	KEM_{Cov}	KEM_{Burg}	**KEM_{Mcov}**	KEM_{Cov}	KEM_{Burg}
	KEM_{Cov}	KEM_{Burg}	KEM_{Burg}	KEM_{Cov}	KEM_{Cov}	KEM_{Burg}	KEM_{Mcov}
	KEM	KEM	KGD	KEM	KEM_{Burg}	KEM	**KEM_{Cov}**
Worst	KGD	KGD	KEM	KGD	KGD	KGD	KGD
Noisy	-2.04	-6.53	0.15	1.03	0.36	1.37	0.66
KEM_{Mcov}	7.44	-1.21	0.57	1.09	0.71	1.75	0.93
KGD	4.14	-2.13	0.46	1.04	0.37	1.64	0.73

Table 4: Optimization-based algorithms ranking for VL/L WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top two lines) is shown in bold.

M/H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**KEM_{Burg}**	KEM_{Mcov}	**KEM_{Burg}**	KEM	KEM	**KEM_{Burg}**	KEM
	KEM_{Cov}	KEM_{Cov}	KEM_{Mcov}	**KEM_{Burg}**	**KEM_{Burg}**	KEM_{Cov}	**KEM_{Burg}**
	KEM_{Mcov}	**KEM_{Burg}**	KEM_{Cov}	KEM_{Mcov}	KEM_{Mcov}	KEM_{Mcov}	KEM_{Mcov}
	KEM	KEM	KEM	KEM_{Cov}	KEM_{Cov}	KEM	KEM_{Cov}
Worst	KGD	KGD	KGD	KGD	KGD	KGD	KGD
Noisy	9.10	-0.13	0.77	1.14	1.55	2.32	1.65
KEM_{Burg}	14.58	3.61	0.94	1.43	1.82	2.59	1.89
KGD	12.02	2.36	0.93	1.29	1.69	2.46	1.80

Table 5: Optimization-based algorithms ranking for M/H WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top two lines) is shown in bold.

In these two tables, the best rated algorithms are respectively the KEM_{Mcov} and the KEM_{Burg}, whereas the worst algorithm is for both the KGD.

This time, while the **KGD** algorithm is the fastest, its objective scores are non-negligibly lower than those obtained from the enhanced speech produced by the other optimization-based methods. From informal listening, the **KGD** algorithm is still able to remove a large amount of noise (accordingly, its ASNR score is relatively good), but the speech quality and intelligibility is noticeably degraded, even at high SNR. Next, we note that overall, using

Burg's method to update the speech parameters gives better results than the **KEM** algorithm, although these results are very similar to those obtained with the methods employing the Covariance and Modified Covariance methods. This is also noted when listening to test files, where it clearly sounds that there is less background residual noise at the output of the $\mathbf{KEM}_{\mathbf{Burg/Cov/Mcov}}$ methods than for the **KEM** one, but also that there is no clear perceivable difference between the $\mathbf{KEM}_{\mathbf{Burg/Cov/Mcov}}$ algorithms. The "top" optimization-based algorithm is therefore chosen to be the $\mathbf{KEM}_{\mathbf{Burg}}$ method.

A listening comparison of the $\mathbf{KEM}_{\mathbf{Burg}}$ output to that of the **DKF** shows that the results are fairly similar, but the $\mathbf{KEM}_{\mathbf{Burg}}$ is superior in terms of background residual noise.

3.2.5.3. Comparisons between Holistic methods

Four types of joint/holistic methods are tested below. The first three are UKF-based, and the last one is a RBPF. All these algorithms employ random-walk models on reflection coefficients rather than autoregressive ones (this has been found to be slightly advantageous in [FON'02, DEN'06]). The convention for the names of the algorithms is the same as for the dual ones. Summary of results are shown in Tables 6 and 7, where an average score over 10 runs is reported for the RBPF[3].

VL/L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**RBPF**	**RBPF**	**RBPF**	**RBPF**	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(1)$
	$JUKF_{Lin}$	$JUKF_{Lin}$	$JUKF_{Lin}$	$JUKF_{Lin}$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$
	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(1)$	$JUKF_{Lin}$	$JUKF_{Lin}$	$JUKF_{Lin}$
Worst	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$	**RBPF**	**RBPF**	**RBPF**
Noisy	-2.04	-6.53	0.15	1.03	0.36	1.37	0.66
RBPF	8.17	-0.50	0.55	1.09	0.59	1.69	0.79
$JUKF_{Nonlin}(4)$	7.34	-1.01	0.48	1.06	0.74	1.70	0.91

Table 6: Holistic algorithms ranking for VL/L WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top line) is shown in bold.

[3] For complete details regarding experimental conditions, objective measures and algorithms ranking procedures, please refer to Chapter II.

M/H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**RBPF**	**RBPF**	**RBPF**	**RBPF**	$JUKF_{Lin}$	**RBPF**	$JUKF_{Lin}$
	$JUKF_{Lin}$	$JUKF_{Lin}$	$JUKF_{Nonlin}(1)$	$JUKF_{Lin}$	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(1)$
	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(1)$	$JUKF_{Lin}$	$JUKF_{Nonlin}(1)$	$JUKF_{Nonlin}(4)$	$JUKF_{Lin}$	$JUKF_{Nonlin}(4)$
Worst	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(4)$	$JUKF_{Nonlin}(4)$	**RBPF**	$JUKF_{Nonlin}(1)$	**RBPF**
Noisy	9.10	-0.13	0.77	1.14	1.55	2.32	1.65
RBPF	16.08	4.97	0.96	1.59	1.83	2.61	1.86
$JUKF_{Nonlin}(4)$	14.66	3.92	0.94	1.35	1.93	2.60	1.96

Table 7: Holistic algorithms ranking for M/H WGN noise conditions. The "top" algorithm (i.e., with the most entries in the top line) is shown in bold.

From this table, it appears again that there is no clear advantage in introducing nonlinearities, except possibly at low SNR when considering the JUKF algorithms for the composite objective measures. The scores are relatively close, however the RBPF's ASNR and WPESQ stand out in most cases. From our informal listening tests, the RBPF offers fairly noticeably the most natural sounding output, but one must also take into account that they are also by far the most expensive of all methods. Because naturalness is one of the important criteria in the context of this thesis, in spite of its computational cost, overall the RBPF is chosen as the "top" holistic algorithm.

3.2.5.4. Comparison between top algorithms of each category

The following tables show comparisons between the DKF, the KEM_{Burg} and the RBPF algorithms, which were picked as good representants of each category of algorithms.

VL/L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-2.04	-6.53	0.15	1.03	0.36	1.37	0.66
RBPF	8.17	-0.50	0.55	1.09	0.59	1.69	0.79
KEM_{Burg}	7.45	-1.22	0.57	1.09	0.70	1.75	0.94
DKF	6.98	-0.95	0.35	1.08	0.41	1.63	0.69

Table 8: Comparative results between algorithms of each category in VL/L WGN conditions

M/H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	9.10	-0.13	0.77	1.14	1.55	2.32	1.65
RBPF	16.08	4.97	0.96	1.59	1.83	2.61	1.86
KEM_{Burg}	14.58	3.61	0.95	1.45	1.83	2.60	1.90
DKF	14.26	3.94	0.92	1.29	1.60	2.46	1.66

Table 9: Comparative results between algorithms of each category in M/H WGN conditions

According to the above tables, the top rated algorithm is mostly the RBPF, followed by the KEM_{Burg} and then the DKF. Nevertheless, note that the scores reported are still in the same range and that some objective measures offer contradictive impressions: for example, judging by the Covl measure, the KEM_{Burg} is the best algorithm. In our experience, such a situation (as opposed to a situation where all objective measures change in the same direction) indicates in general that the enhanced products are of similar quality, but have distinct "features". From informal listening, it is found that the RBPF offers a good compromise between naturalness and noise reduction, and stands out at high SNR for being able to remove the most background noise. Next, the DKF still offers very good noise reduction but yields a slightly less natural output. Finally, the KEM_{Burg} lies in between the two, with however a noticeable accent on naturalness at low SNR.

3.3. Monaural background noise estimation and types of noise covered

A high-level description of background noise estimation is presented in this section, with references for the algorithms used in this thesis.

As stated in Chapter I, it was chosen to resort to separate background noise estimation algorithms to support the enhancement schemes. The main reason for this choice is related to robustness constraints: while Voice Activity Detectors (VADs) offer a very simple solution, based on the detection of speech pauses over which the noise statistics are measured, they often yield unsatisfactory results in highly nonstationary environments [RAN'06]. This is principally due to two facts: first, the noise statistics may be rapidly changing during speech, and secondly the actual speech/pause detector is never fully reliable. As briefly discussed in Section 3.2., another motivating reason for choosing background noise PSD estimation rather

than VAD is the fact that it is currently evolving in the binaural domain, i.e., the field is currently moving towards estimation when two channels are available [KAM'09], with more accurate estimation then made possible.

What the choice of external PSD estimation implies is that the capacity of the techniques shown in the thesis to handle non-stationary conditions are determined by the capacity of the noise PSD estimators to track non-stationary noise. Currently PSD estimators are capable of quickly adapting to noise level changes, directional sources, and interfering speakers (see [KAM'09, KAM'10]). Moreover, as previously noted in Section 3.1.4, even with a "plain" additive noise model specific diffuse noise PSD estimators can in fact be used to remove reverberation tails, which can be tagged as diffuse noise.

Thus, not only do we have a wide range of complex noise types currently covered by PSD estimators, but also this range is in fact widening given the current evolution of the research field of noise PSD estimation. While noise types such as directional sources and interfering speakers do not appear in the results of this thesis, related experiments are mentioned in Chapter X, where binaural processing is introduced. In summary, provided a reliable noise PSD estimator is available to target an arbitrary form of complex noise, it can be naturally included in the frameworks developed throughout this thesis.

The research field of background noise estimation is fairly mature, and there exists many algorithms with good performance (see for example [MAR'01], [DOB'95], [COH'02], [HIR'95], [RAN'06]). In [MAR'01] and [DOB'95] for example, the main idea consists of tracking the spectral minima of the noisy speech, as assimilating it to the noise. In [HIR'95], the noise estimate is updated based on a simple level comparison between the measured noisy

speech power spectrum and the previous noise estimate, and again the values below a certain threshold are assimilated to the noise. In [RAN'06], the reader can find a good review of several other noise estimation algorithms, as well as an interesting comparison.

In this thesis, after experimenting with several algorithms, two methods were picked: the ones in [DOB'95] and [HIR'95] for two reasons: first, they are the simplest and most efficient (computationally speaking) of all the above-cited algorithms. Secondly, it was found that by averaging the two to obtain one output PSD estimate, similar results to any of the more complex algorithms above can be obtained.

Regardless, throughout this thesis it is assumed that every speech enhancement algorithm, whether existing in the literature or developed in these pages, has been adapted to use the exact same noise PSD estimator, in order to allow for fair comparisons.

3.4. Other classical speech enhancement algorithms

For completeness, three classical speech enhancement methods are overviewed in this section: the spectral subtraction, the signal subspace method, and the log-spectral amplitude estimator. Over the last few decades, these three algorithms have spanned many variations – they are considered in [LOI'07] to represent well the three major categories of well-recognized algorithms.

3.4.1. Spectral subtraction

The main idea behind spectral subtraction, originally described in [BOL'79], is very simple: given a signal corrupted by additive noise, estimate the spectrum of the noise and subtract it from the spectrum of the observed mixture, and then revert to time-domain to obtain the cleaned signal. Figure 4 shows the high-level steps of basic spectral subtraction.

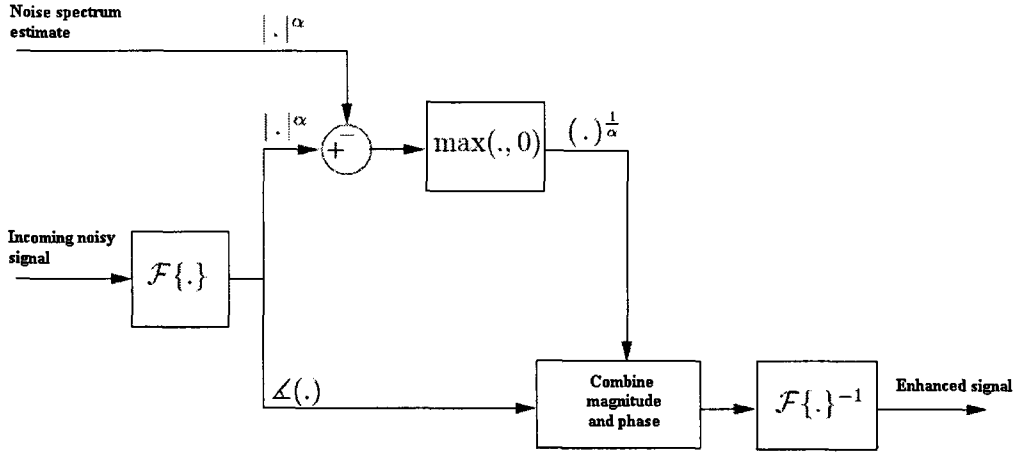

Figure 4: The basic spectral subtraction method. The parameter α can be fixed beforehand or adjusted online.

The technique of spectral subtraction is typically performed on frames of windowed data. The assumptions are the following: the noise and the signal are uncorrelated, and within a frame the noise is almost stationary. In addition, it is assumed that the effect of additive noise on the phase of the spectrum of the original signal is negligible to the human ear, so that the phase of the spectrum of the noisy signal is practically considered to be clean. Finally, an estimate for the magnitude of short-time noise spectrum is required as well.

Following Figure 4, the first step consists in converting the windowed measurements over a frame to the frequency domain. Denote by $Z(f)$ the obtained spectrum for the windowed noisy data and by $|V(f)|$ the magnitude of the corresponding noise spectrum. The magnitude $|Z(f)|$ and the phase $\measuredangle Z(f)$ are then separated.

The spectral subtraction can then take place as follows:

$$\left|\hat{X}(f)\right| = |Z(f)| - |V(f)| \tag{16}$$

It was found that it can be beneficial to perform the subtraction in a more general way, as follows:

$$\left|\hat{X}(f)\right|^{\alpha} = |Z(f)|^{\alpha} - \beta |V(f)|^{\alpha} \tag{17}$$

where α and β are positive numbers, either fixed or adapted online as in [VIR'99].

In practical implementations, it is wise to ensure that the right-hand side remains positive, and so we may rather write:

$$\left|\hat{X}(f)\right|^{\alpha} = \max\left\{|Z(f)|^{\alpha} - \beta |V(f)|^{\alpha}, 0\right\} \tag{18}$$

To complete the procedure, the assumption that the phase of the clean signal is close enough to that of the observed signal is used, so that the final expression for the estimated time-domain clean signal within the considered frame can be obtained by inverse Fourier transformation of $\max\left\{|Z(f)|^{\alpha} - \beta|V(f)|^{\alpha}, 0\right\} \times e^{j\angle Z(f)}$.

In general, subtractive-type algorithms suffer from an important drawback: the enhanced signal contains artefacts, including a sort of musical, robotic noise, sometimes described as a "synthetic bird-song" in the background. This phenomenon is due to the zero-flooring (the maximum operation in Equation (18)) that is applied to the short-time spectra of the frame under consideration, after subtraction. In the short-time spectrum of the actual true noise, some peaks can appear "randomly" in frequency. After subtraction and flooring, some of the narrower remaining peaks can be identified to time-varying tones, which produce the musical noise.

On the other hand, they are still widely used due to their simplicity, and also because a clever adaptation of α and β can reduce the artefacts in the enhanced product [VIR'99].

3.4.2. Signal subspace method

The signal subspace approach, introduced in [EPH'95], is another different class of methods used for speech enhancement. It is based on the following so-called linear model for speech; Suppose here that $\mathbf{x}$ represents a vector M clean speech signal samples. In the linear model, it is assumed that $\mathbf{x}$ can be written as[4]:

$$\mathbf{x} = V\mathbf{s} \tag{19}$$

[4] Such a model for speech signals has been validated by successful experimentations and practical implementations [EPH'95]

where V is an $M \times K$ full rank real matrix, $\mathbf{s}$ is a length K zero-mean complex random vector with (nonsingular) covariance matrix $R_{\mathbf{s}}$, and where we impose $K < M$.

First recall that for zero-mean processes, the covariance matrix is equal to the correlation matrix. In the following, we may use either terms intechangeably. This speech representation has multiple implications:

- First, $\mathbf{x}$ is zero mean since $\mathbf{s}$ is too,
- The covariance/correlation matrix of $\mathbf{x}$ is given by $R_{\mathbf{x}} = E(\mathbf{x}\mathbf{x}^T) = VR_{\mathbf{s}}V^T$,
- $R_{\mathbf{x}}$ has rank K only, and thus $R_{\mathbf{x}}$ has only K nonzero eigenvalues.

Suppose that a frame of white noise $\mathbf{v}$ is added to the frame $\mathbf{x}$. In contrast with the speech signal, since $\mathbf{v}$ is a white-noise random vector, its covariance matrix $R_{\mathbf{v}}$ is (diagonal and) non-singular, and all of its eigenvalues are nonzero — they are in fact all equal to the variance of the noise. Therefore, $\mathbf{v}$ lies in the entire M-dimensional space, but $\mathbf{x}$ lies only in the range of V.

The range of V, denoted by $R(V)$, it called the signal subspace. The orthogonal complement of $R(V)$ is called the noise subspace. Thus, in a frame of a noisy speech signal, the noise lies in both the signal and the noise subspaces, but the clean speech is only confined to the signal subspace. From these considerations, the signal subspace approach can be summarized in the following two steps:

- Given a frame of noisy speech, first try to determine the signal and the noise subspaces, and compute the orthogonal projection of the noisy speech onto the signal subspace,
- Secondly, from the resulting vector, obtain an estimate for the clean speech.

The details of the two steps above are not detailed here, but in summary the first point above is conducted conveniently via Karhunen-Loève transform, and the problems boils down to estimating the covariance matrices R_x, R_z (the covariance of the noisy frame), as well as the noise-only variance (in the white noise case). Once a projection of the noisy speech onto an estimate for the signal subspace is obtained, there remains to estimate the clean speech. Several estimators are presented in [EPH'95]; these estimators are principally concerned with minimizing the signal distortion while removing/reducing the remaining contribution from the noise present in the signal subspace. The reader is referred to these sources for more details on the subject. Note that while the above discussion assumes that the observation noise is white, several techniques exist to extend the method to colored noise (see [LOI'07]), for example by means of pre-whitening.

Signal subspace methods are recognized as being superior to classical spectral subtraction schemes [LOI'07], in the sense that they introduce less musical noise in the enhanced speech. However, the musical noise is not eliminated, and typically stems from rapid changes of model order, and mostly "subspace swapping" [KLE'02], where noise basis vectors are incorrectly used to span the speech subspace.

3.4.3. Log-spectral amplitude estimator

The Log-spectral amplitude estimator algorithm (LMMSE below), which can be found in the celebrated publication [EPH'85], operates in the same context as the spectral subtraction algorithm presented above. In the frequency domain, assuming that the phase of the clean speech is equal to that of the noisy speech, the goal is to estimate the magnitude (or spectral amplitude) of the clean speech. However, while spectral subtraction techniques make no assumption about the prior statistical distributions of speech and noise spectral amplitudes, the

LMMSE technique is based on such a priori knowledge. Specifically, in [EPH'85] it is proposed to derive the spectral amplitudes of the clean speech signal based on the assumption that the Fourier coefficients of the speech and noise signals are independent Gaussian random variables[5].

From this assumption, a closed form expression for the expected value of the log-spectral amplitude of the clean speech given the noisy observations (both viewed here as random variables), can be derived. This closed form solution can also be viewed as the one minimizing the mean square error of the clean log-power spectrum, hence the name of the algorithm. The reader can refer to [EPH'85] for the details regarding this derivation.

More importantly for our context, it is noted that this algorithm is very well-established and its performance is also very well rated in terms of output quality (see the reference [HU'06-2] where extensive subjective testing is performed on various algorithms). It introduces significantly less distortion than classical spectral subtraction algorithms, although it is still bound to introduce some by nature. It is also considered fairly lightweight, and is therefore commonly used (when considered together, the paper [EPH'85] and its non-log domain counterpart [EPH'84] have been cited by thousands of researchers).

[5] This assumption was in fact used in a previous effort by the same authors [EPH'84]. In [EPH'84], essentially the same work is done as in [EPH'85], but in the latter the solution is found in the log-domain, with reportedly better results.

3.5. Conclusion

In this Chapter, several important background elements were presented.

First, a description of the models for the speech and noise signals was given: in this thesis, they are seen as autoregressive processes, added together to form the noisy measurements. Next, a recapitulation and a categorization of many existing state-space based algorithms, along with a few novel variations for them, were presented. Three main categories were distinguished: Dual algorithms, Optimization-based algorithms, and Holistic algorithms, each having a distinct way of estimating the clean AR model's parameters. For each of these categories, based on output quality and computational complexity as determined via experimentation in WGN conditions, one algorithm was picked as the best representant: respectively, the Dual Kalman Filter (DKF), the optimization-based algorithm resorting to a repeated application of Burg's AR estimation method (KEM_{Burg}), and the Rao-Blackwellized Particle Filter (RBPF).

To complete the Chapter, a brief review of monaural noise PSD estimation was given, with references to the algorithms used in the algorithms of the thesis; and three classical speech enhancement methods were described – each representing one of the main branches of methods of enhancement as reported in [LOI'07].

IV. Low-cost improvement techniques used throughout the thesis

In this chapter, three independent and low-cost techniques aimed at improving the quality of the output speech in state-space based algorithms are described. The first two "tricks" can be used with any algorithm, while the last one is specifically targeted to particle filters.

In the first method, it is shown how the state-space model can be slightly modified/augmented to incorporate delayed estimates, which is found to be beneficial in terms of output quality. The second method explains how combining several algorithms of different nature can significantly improve the naturalness and overall quality of the enhanced speech. These first two techniques also appear in our journal article in [MUS'08-2]. Finally, the last section shows that using multiple small-size particle filters can yield even better results at a smaller computational cost than a single, large-size particle filter.

4.1. Delayed estimates

At every iteration, the state space-based speech enhancement algorithms return an estimate $\mathbf{x}(k) = [x(k) \quad x(k-1) \quad \cdots \quad x(k-M_s+1)]^T$ when the model order is M_s. One would naturally choose, at time k, the first element of $\mathbf{x}(k)$, denoted by $\mathbf{x}(k)_{[1]}$, as the estimate for the current speech sample. In [PAL'87] – one of the pioneering works on Kalman Filtering for speech enhancement – it was noted that in an AR speech production setting, a better estimate can be obtained by considering the delayed estimate given by $\mathbf{x}(k+M_s)_{[M_s]}$. As noted in [PAL'87], this phenomenon can be observed through the diagonal elements of the covariance matrix of $\mathbf{x}(k)$ which, as the KF unfolds, tend to an arrangement in descending order (according to the

notation of this thesis). Indeed, the steady-state marginal distribution of the last element of $\mathbf{x}(k)$ has a smaller covariance than that of the first element, and thus is more accurate.

In this thesis, it is proposed to go further while keeping an M_s th order autoregressive model: it suffices to choose $J \geq M_s$ and then define an "augmented" linear model (shown below in the white noise case for clarity):

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}_k \mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v v(k) \end{aligned} \tag{20}$$

where:

- M_s is still the autoregression order
- $\mathbf{x}(k) = \begin{bmatrix} x(k) & x(k-1) & \cdots & x(k-J+1) \end{bmatrix}^T$
- $\mathbf{a}_k = \begin{bmatrix} a_1(k) & a_2(k) & \cdots & a_{M_s}(k) \end{bmatrix}^T$
- $\mathbf{A}_k = \begin{bmatrix} \mathbf{a}_k^T & 0_{1\times J-M_s} \\ I_{J-1\times J-1} & 0_{J-1\times 1} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0_{1\times J-1} \end{bmatrix}$
- $\mathbf{G}_k = \mathrm{diag}\left\{\sigma_g(k); I_{J-1\times J-1}\right\}$

One can then consider instead the further delayed estimate $\mathbf{x}(k+J)_{[J]}$. By doing so, it may appear that a heavy increase of computations in each algorithm is imposed – it is in fact not the case, due to the nature of the matrix $\mathbf{A}_k$, and only a few extra operations are necessary per KF. For $J \geq M_s$, in PF algorithms the increase in memory requirements is also smaller compared to the so-called "fixed-lag smoothing" technique (explained in the PF context in [DOU'00, VER'02]).

4.2. Combination of estimates

A simple yet again very beneficial way of improving the estimates returned by enhancement algorithms is now presented. In a very generic setting, the idea is the following: given several independent estimates for an unknown variable, instead of selectively discarding the ones that are judged inadequate, it is proposed to try to take advantage of each of them by combining them, thereby creating a new hybrid estimate. This generic idea is in fact not new, and has been very successfully applied in other disciplines such as Medicine and Biology [MAS'00], Economics, Statistics, and more [CLE'89]. In [MAS'00], where data from different algorithms are combined to the mammographic skin–air interface, it is shown that the hybrid estimate is able to overcome many problems affecting each individual algorithm. Perhaps even more compelling is the following quote from [CLE'89], in which more than 200 academic studies in various fields were surveyed:

> *''Consider what we have learned about the combination of forecasts over the past twenty years. Models have been developed to find ''optimal'' combinations of forecasts. Both simulation and empirical studies have been done to test the models. Bayesian interpretations have been presented. The results have been virtually unanimous: combining multiple forecasts leads to increased forecast accuracy.''*

Based on the above considerations, it is therefore appealing to try to combine the estimates obtained from several speech enhancement algorithms. The combination scheme proposed here is based on Kalman Filtering as well. With the clean speech signal still denoted by $x(k)$, suppose that an estimate of this speech, denoted by $\tilde{x}(k)$, returned by another distinct algorithm is available. Now, denote by $\hat{x}(k)$ the estimate obtained by the primary enhancement algorithm. Now let $\theta(k) = [\lambda(k) \quad 1-\lambda(k)]^T$ be a vector of mixing proportions

$\lambda(k)$ and $1-\lambda(k)$ (restricted to sum to 1), let $\mathbf{y}(k)=[\hat{x}(k) \quad \tilde{x}(k)]^T$, and consider the following mixture model:

$$\begin{aligned} \lambda(k) &= \lambda(k-1)+\sigma_\lambda n_\lambda(k) \\ z(k) &= \mathbf{y}(k)^T\theta(k)+\sigma_v(k)v(k) \end{aligned} \tag{21}$$

where $n_\lambda(k)$ is a zero-mean unit-variance white Gaussian noise, σ_λ is a constant number, and $\sigma_v(k)v(k)$ is the same observation noise as the one used in the primary model used to obtain $\hat{x}(k)$ (it is white Gaussian for this example, but it could be colored as well).

Effectively, this model describes a possible way of viewing the combination proposed in state-space form, with the one-dimensional state defined by $\lambda(k)$. Using a simple KF, one could then determine the optimal $\theta(k)$ that must be used to combine the estimates from two algorithms so as to minimize the mean-square error. Used as such, this approach is not restricted in any way to the use of state space-based algorithms, however, an estimation of $\lambda(k)$ could very well be performed within the framework of Algorithms 1 to 5 by augmenting the state vector with $\lambda(k)$ and the observation equation with the other, external estimate $\tilde{x}(k)$. Note also that this method is obviously not restricted either to the use of only two distinct estimates. In fact, to go further, it is not even restricted to speech enhancement, nor to state-space formulated problems.

A priori, it is legitimate to expect that this scheme will benefit more from the use of algorithms with very distinct "enhancement strengths": for example, a spectral subtractive method enhances the speech very differently from the model-based methods described so far in this document. Our experience shows that a "winning combination" consists of a combined KF-based algorithm and a wavelet-packet based enhancement method. In lower

SNRs, we find that the combination with basic spectral subtraction estimation can also provide non-negligible improvements, at an even lower added computational cost.

In order to test the idea above, a large amount of simulations with algorithms of different kinds was considered. For the combination scheme, an external KF running on Equation (21) was used, with the primary algorithm being an RBPF enhancing speech corrupted with white Gaussian noise. Non-negligible improvements over the original individual estimates were consistently observed. However, the best improvements occurred when these individual estimates had very different "features" (e.g. background noise type, speech naturalness, etc.). To perform some comparisons, a few other algorithms were implemented and tested, with an emphasis on fast algorithms which can be easily implemented as "secondary" algorithms for the RBPF:

- A wavelet-packet thresholding scheme, based on the idea presented in [DON'94], was used with no entropy-based criterion for the tree decomposition, and where the noise variance was also known. A Daubechies-6 mother wavelet was used, and the decomposition was performed on blocks such that the resulting number of samples per band was 64. Then, a universal soft thresholding scheme was applied, and finally an inverse transformation was used to recover the speech. This first scheme is referred to hereafter as WPT.
- A frame-based amplitude spectral subtraction algorithm with fixed subtraction parameters, based on [BOL'79], in which the first few frames were used to determine the noise spectrum. This algorithm is codenamed SPECSUB.
- The simple Dual-KF (DKF) algorithm described in Section 3.2.4.2, operating on the same AR speech model as the RBPF.

The average results are summarized in Tables 10 and 11 below, following the methodology and objective measures described in Chapter II.

VL/L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-2.04	-6.53	0.15	1.03	0.36	1.37	0.66
(1) RBPF	8.17	-0.50	0.55	1.09	0.59	1.69	0.79
(2) WPT	7.40	-1.44	0.48	1.11	0.37	1.54	0.77
(3) SPECSUB	7.77	-0.95	0.70	1.09	-1.10	0.66	-0.34
(4) DKF	6.98	-0.95	0.35	1.08	0.41	1.63	0.69
{(1)+(2)}	**+0.82**	**+0.69**	**+0.11**	**+0.12**	**+0.34**	**+0.17**	**+0.31**
{(1)+(3)}	**+0.97**	**+0.57**	**+0.17**	**+0.10**	**+0.21**	**+0.17**	**+0.14**
{(1)+(4)}	-0.17	-0.04	0.00	**+0.01**	-0.03	0.00	0.00

Table 10: Simulations of several different combinations of algorithms, for VL/L WGN conditions. In the second part of the table, the notation "{(A)+(B)}" indicates the result of the amalgamation of the estimates from algorithm (A) and algorithm (B). In addition, for better readability only the actual difference in score from the regular RBPF in the first row is reported. The best improvement is highlighted in bold characters.

M/H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	9.10	-0.13	0.77	1.14	1.55	2.32	1.65
(1) RBPF	16.08	4.97	0.96	1.59	1.83	2.61	1.86
(2) WPT	13.78	3.19	0.94	1.68	1.69	2.50	1.94
(3) SPECSUB	11.64	2.37	0.95	1.34	0.09	1.50	1.59
(4) DKF	14.26	3.94	0.92	1.29	1.60	2.46	1.66
{(1)+(2)}	**+0.40**	**+0.56**	**+0.02**	**+0.30**	**+0.34**	**+0.18**	**+0.34**
{(1)+(3)}	-0.86	-0.34	**+0.03**	**+0.19**	-0.01	-0.09	-0.04
{(1)+(4)}	-0.46	-0.26	0.00	-0.10	-0.04	-0.04	-0.04

Table 11: Simulations of several different combinations of algorithms, for M/H WGN conditions. In the second part of the table, the notation "{(A)+(B)}" indicates the result of the amalgamation of the estimates from algorithm (A) and algorithm (B). In addition, for better readability only the actual difference in score from the regular RBPF in the first row is reported. The best improvement is highlighted in bold characters.

There are of course very many ways to apply such an algorithm; however the non-exhaustive list of results shown in Tables 10 and 11 still indicates a few trends. First, according to these results, improvements can be observed across several objective measures, especially at low SNR. A closer observation shows however that the degree of improvement depends on the accessory algorithm used. In fact, the algorithm from which the RBPF benefits the most is the WPT algorithm (at VL/L and M/H SNR conditions), whereas the one offering the least amount of changes is the Kalman-filter based algorithm (the DKF). For the SPECSUB algorithm, although some noticeable benefits are observed at VL/L SNR, only marginal benefits and even some mild overall quality degradation is observed at M/H SNRs. Neverthless, from informal listening the degradation is not striking, and does not outweight the improvements introduced at low SNR – where they are after all most needed.

One possible natural interpretation of the above observations is the following: if the mixture scheme is to gather the best features of each algorithm, then there is a larger expected benefit when the two algorithms are very different by nature.

Another interesting point is the following: in the cases presented, the second estimates were obtained by means of very fast algorithms, i.e., algorithms with very low computational complexity. Although these estimates are coarser, and include noticeably more residual noise, the benefits at low SNR are still clear at the output of the combination process.

From repeated informal and careful listening tests, it is found that the speech enhanced using combined algorithms retains the best aspect of each scheme. For example, although there is a large amount of residual noise in the wavelet packet and spectral subtractive schemes, the unvoiced sounds are still clearly reproduced at very low SNR. On the other hand, while the

residual noise in the RBPF-enhanced speech is almost inexistent between utterances, the RBPF often damages the unvoiced sounds, which in turn affects the speech intelligibility. In the combination of the two, there is still almost no interspeech residual noise, but the unvoiced sounds are more distinguishable, and therefore intelligibility is better preserved.

4.3. Combination of small-size particle filters

By running several small particle filters in parallel, our experiments have indicated that both execution time and especially accuracy in the estimation can be improved. The overall sample size was found to be greater than in the case of a single, large PF, resulting in richer particle representations of the clean speech. Theoretical analysis and experimental validation of this observation is left as a future project, but a few trends are already detectable, as explained below.

4.3.1. Presentation of the idea

Since particle filters are stochastic algorithms, every run yields a different output for a given amount N of particles. Thus, to get a proper view of their actual performance, several outputs must be observed and analyzed. In the context of this research, the type of score distribution obtained for the SNR, ASNR and WPESQ measures were analyzed for both the RBPF and the neural PF types of algorithms, and for various input SNR. It was observed that the type of distribution obtained consistently matches the following description: unimodal and slightly skewed towards the right for the SNR and ASNR – more so for the WPESQ. The variance of the results is mostly dependent on the number of particles used, as well as on the type of algorithm employed. It was found that for a sufficiently large number of particles, this variance is very small (in the same range for all three types of scores). It was also found that the distributions associated with each objective measure can be well approximated by

generalized logistic-type distributions, whose PDF is given by $f_X(x) = \frac{be^{-(x-\mu)/s}}{s(1+e^{-(x-\mu)/s})^{b+1}}$ (where μ is the location, s is the scale, and b is the skew).

Figure 5 illustrates these claims.

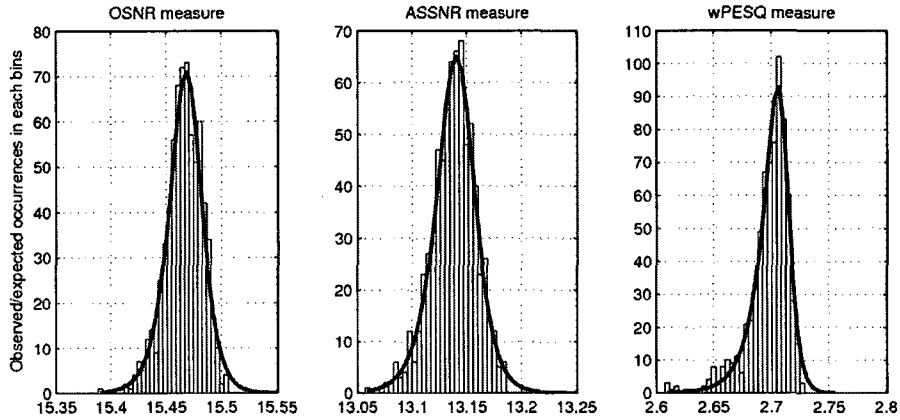

Figure 5: Generalized logistic distribution fit for an example of results obtained with a delayed RBPF using 1200 particles. In the above figures, the scores from 1000 runs were placed in 30 bins, and a minimum χ^2 fit was performed.

While analyzing such results, it can be found that there is a direct relationship between the so-called *effective sample size* of particle filters, and the actual performance at the output. The effective sample size of a particle cloud (or of a set of particles representing candidates for the clean speech), denoted by N_{eff} below, represents the fraction of particles having non-negligible weight. For all three measures, the lowest scores occur when, at some point(s) in the simulation run, a particle cloud is drawn with a low N_{eff} value, which in turns means that the average of the non-zero particles may in fact be a poor estimate for the clean speech. This also means that, during the resampling step, a very large amount of particles is discarded, and those candidates in the small set of viable particles are duplicated many times, resulting in a non-diverse set at the beginning of the next iteration.

Two interesting observations can be made in the context of speech enhancement. First, one finds that N_{eff} can actually drop to a *very* small fraction of N. For example, with N = 1000, N_{eff} can drop to around 10-15 occasionally, i.e., only a *percent* of the particle cloud is valid at some instant. The second observation that is of interest is that, for a fixed model, the actual instants at which the sample size drops repeat themselves from one run to another, with more or less intensity. Furthermore, these actual instants do not depend on the amount of particles, and most importantly and surprisingly, neither do the lowest values of N_{eff}. In other words, given a certain signal to enhance, there are particular areas for which it can be expected that the PF algorithm will struggle to maintain a diverse particle cloud, with little regard towards the initial amount of particles. In was observed that if, on average, at a certain time instant N_{eff} reaches a certain amount (say 20) with 1000 particles, then even if 500 particles are used, on average one will also observe an effective sample size of 20 at the problematic areas. One additional conclusion of these observations is that augmenting the amount of particles cannot eliminate certain local enhancement problems.

As an illustration, Figure 6 shows the N_{eff} obtained with different runs of a fixed algorithm -- for one configuration with 1000 particles. The purpose of Figure 6 is to show that, indeed, the different N_{eff}'s obtained do not differ very much from one run to the next. Next, Figure 7 shows, superimposed, the average (over 20 tests) of N_{eff} obtained for 1000 particles, and the average (again over 20 tests) of $2 \times N_{eff}$ obtained for 500 particles. Figure 7 is particularly interesting, as it supports the claims of the previous paragraph. Moreover, zooming in the minima of N_{eff} (Figure 8) shows that these minima are actually the points where there is an apparent benefit in using two separate, smaller PFs, rather than a larger one.

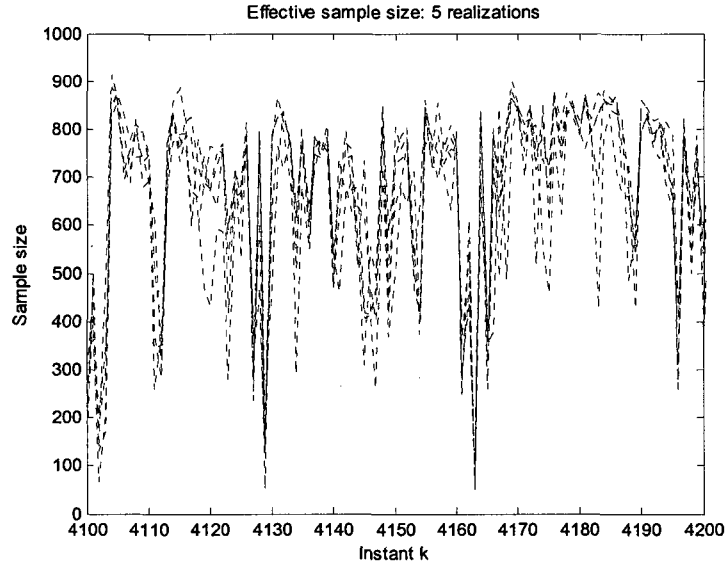

Figure 6: N_{eff} values obtained for 5 independent runs of an RBPF with a fixed amount of particles (here 1000). Note the very strong similarity between each curve.

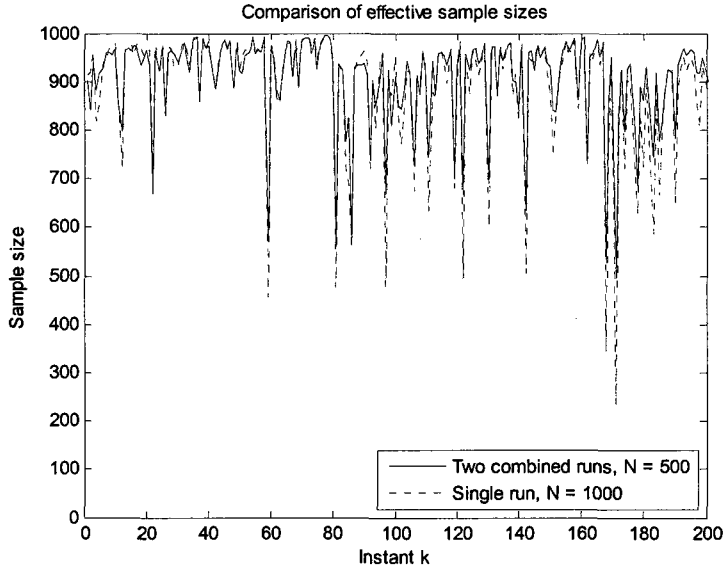

Figure 7: In dotted line, the average of N_{eff} obtained for 1000 particles is plotted, and in solid line, the average of $2 \times N_{eff}$ obtained for 500 particles is superimposed. The two curves are very similar, except for the lower minima, which are not as low for the blue curve (see next Figure for a zoomed graph).

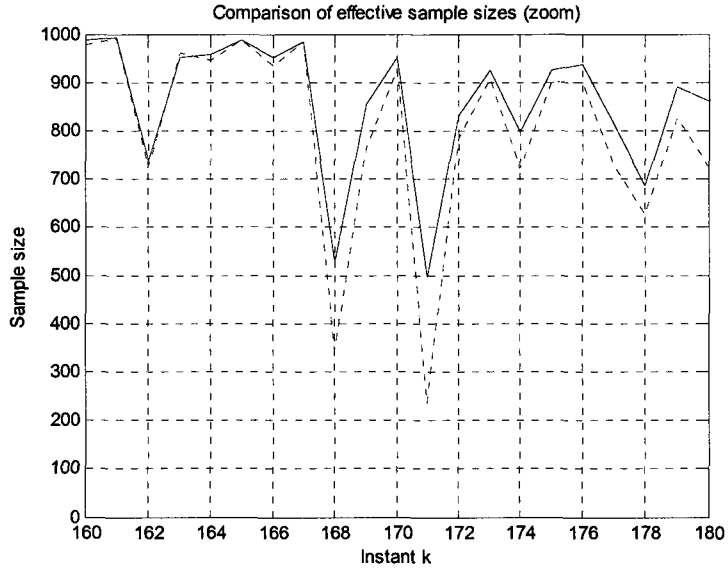

Figure 8: A zoomed version of the previous Figure, clearly showing the advantage of using two combined smaller PFs rather than one larger one. At the "problematic" instants (here $k = 168$ and $k = 171$), the two smaller PFs combine to offer a larger diversity of particles (by about 200 particles).

Based on these insightful simulation examples, some more rigorous theoretical work is currently being conducted to determine the true reason behind this phenomenon. It is believed that the reason comes down to a better effective resampling. In spite of the phenonema illustrated in Figure 7 and 8, at the problematic instants the smaller PFs must still come up with their own estimates based on two independent sets of particles of approximately the same sample size – but at this point it is still difficult to explain why combining the two, by simple averaging for example, will yield a better overall enhanced speech than using the single, larger PF. The next section shows some preliminary results regarding the latter claim.

4.3.2. Testing

In the batch of results below, a single RBPF is run in a stationary colored noise environment with $2N$ particles to form a first estimate, and two RBPFs with N particles each are combined

to form another one. The average over 10 results is reported in Table 12, for the VL-WGN experimental conditions defined in Chapter II.

Algorithm	Quality measure	*N*					
		100	200	400	600	1000	1500
One RBPF $2N$ particles	SNR	6.24	6.48	6.57	6.80	6.90	6.93
	ASNR	-1.57	-1.44	-1.34	-1.30	-1.28	-1.27
	CSII	0.32	0.35	0.37	0.38	0.40	0.41
	WPESQ	1.05	1.05	1.06	1.06	1.06	1.07
	Csig	0.34	0.36	0.38	0.39	0.40	0.41
	Cbak	1.53	1.54	1.54	1.55	1.55	1.56
	Covl	0.60	0.62	0.63	0.63	0.64	0.65
Two RBPFs N particles each	SNR	6.44	6.57	6.83	6.94	6.96	6.94
	ASNR	-1.45	-1.37	-1.32	-1.26	-1.26	-1.25
	CSII	0.35	0.37	0.39	0.40	0.40	0.41
	WPESQ	1.05	1.05	1.06	1.06	1.07	1.07
	Csig	0.37	0.38	0.40	0.40	0.41	0.41
	Cbak	1.55	1.54	1.55	1.55	1.56	1.56
	Covl	0.63	0.63	0.63	0.64	0.65	0.65

Table 12: Single, large RBPF ($2N$ particles) vs. two smaller RBPFs (N particles each).

Observing Table 12, there is a slight but consistent improvement across all input SNRs and all objective measures. It is worth noticing and mentioning that the improvement is more pronounced when N is "low" (below 600), and the differences become marginal as the number of particles grows very large.

Since there are no drawbacks (i.e., no increase in computational complexity – in fact a lower overall amount of particles is required for equivalent quality), this simple method is used as a standard in the rest of the thesis. One can in fact argue that the resampling process can be made less complex as well (since it occurs on two smaller sets of particles).

4.4. Conclusion

Three simple techniques were presented in this Chapter, each found to consistently benefit state-space based algorithms at either no cost or very small cost in terms of computations. The first two can be applied to any state-space based algorithm.

The first technique consists of a slight modification of the usual state-space model so as to incorporate a delay; the second one shows that some non-negligible improvements can be achieved by combining two estimates of different nature; finally in the third one it is observed that distributing N particles to several particle filters can yield better results than using a single PF to deal with the N particles.

V. Improved colored noise handling in state-space based algorithms

This chapter presents a simple alternative to the traditional handling of autoregressive colored observation noise processes in state-space based speech enhancement algorithms. The method is entirely centered on a rewriting of the state-space equations describing the problem, which in turns results in a different application of the central algorithms of Section 3.2.4.

The proposed approach decreases the dimension of the state vector and the amount of computations per iteration, and also naturally reduces to the white noise case when a zero-order autoregressive colored noise is chosen. This work was published in [MUS'08].

5.1. Problem statement

Let us first recall here the white-noise-only state-space model used for speech enhancement, in a more general nonlinear speech production setting (previously presented in Equation (2)):

$$\begin{aligned} \mathbf{x}(k) &= \Phi_k\left(\mathbf{x}(k-1)\right) + \mathbf{G}(k)\mathbf{w}(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v(k)v(k) \end{aligned} \tag{22}$$

The problem is now to incorporate an autoregressive colored observation noise, as described in Section 3.1.3, in the state space model. The traditional way of doing this consists in augmenting the speech state vector $\mathbf{x}(k)$ with a vector of lagged noise samples $\mathbf{n}(k)$ and modifying the equations as follows (see for example [GAN'98, GAB'05, GAB'02, DAV'87, KI'96, GIB'91, WAN'98, CHA'07, GRA'04, PAR'06, ZAV'06]).

Assuming a linear model for simplicity, if $\mathbf{b}(k)$ is the length M_n autoregressive vector governing the noise process, then the equations become:

$$\begin{aligned} \mathbf{s}(k) &= \mathbf{F}_k \mathbf{s}(k-1) + \mathbf{H}_k \eta(k) \\ z(k) &= \mathbf{D}\mathbf{s}(k) \end{aligned} \tag{23}$$

where:

- $\mathbf{s}(k) = [\mathbf{x}(k); \mathbf{n}(k)]$
- $\mathbf{F}_k = \text{blkdiag}(\mathbf{A}_k, \mathbf{B}_k)$
- $\mathbf{B}_k = \begin{bmatrix} \mathbf{b}_k^T \\ I_{M_n-1 \times M_n-1} \quad 0_{M_n-1 \times 1} \end{bmatrix}$
- $\mathbf{H}_k = \text{blkdiag}(\mathbf{G}_k, \sigma_v(k), 0_{M_n-1 \times M_n-1})$
- $\mathbf{D} = \begin{bmatrix} 1 & 0_{1 \times M_s-1} & 1 & 0_{1 \times M_n-1} \end{bmatrix}$

and where $\eta(k)$ is a zero-mean unit variance Gaussian random vector of appropriate dimension. The notation "blkdiag(X,Y)" above means "block diagonal concatenation of matrices X and Y". Note that Equation (23) is effectively noise-free. These equations are at the basis of all the above-cited sources.

In this thesis, a different way of incorporating the autoregressive noise process in the state-space equations is presented. The new form obtained yields multiple advantages, including a lesser dimension, significantly fewer computations and, according to the simulation results presented, a quality of enhancement that is at least as good. In addition, when $M_n = 0$, the system naturally reduces to white-noise-only equations, which is not the case for Equation (23).

5.2. Formal presentation

The main idea at the origin of the proposed equations stems from the fact that there is some redundancy in the state vector $\mathbf{s}(k)$ used in Equation (23) since the speech $x(k)$ and the noise $n(k)$ are deterministically related to each other by the measurement $z(k)$.

Compared to the white-noise-only equations (see Equation (22)), the modified equations only require the definition of one extra term, which is then added to the observation equation, as well as an appropriate modification of the matrix $\mathbf{C}$. With M_n still denoting the order of the colored observation noise, assume that $M_s > M_n$ (this assumption simply clarifies the notation – if this is not the case, one may either invert the roles of noise and speech, or simply zero-pad the speech AR vector, so the rest of the thesis assumes that $M_s > M_n$) and let $\mathbf{z}(k) = [z(k) \quad z(k-1) \quad \cdots \quad z(k-M_n+1)]^T$. Then, the proposed state-space equations are:

$$\begin{aligned} \mathbf{x}(k) &= \Phi_k(\mathbf{x}(k-1)) + \mathbf{G}(k)\mathbf{w}(k) \\ z(k) &= \tilde{\mathbf{C}}_k \mathbf{x}(k) + \mathbf{b}(k)^T \mathbf{z}(k-1) + \sigma_v(k)v(k) \end{aligned} \tag{24}$$

where $\tilde{\mathbf{C}}_k = \begin{bmatrix} 1 & -\mathbf{b}(k)^T & 0_{1\times M_s - M_n - 1} \end{bmatrix}$

Therefore, the transition equation remains unchanged, and the observation equation effectively reduces to that of the white Gaussian noise case when $M_n = 0$. In this setup, note that the state vector must only contain at most M_s lagged values of $x(k)$, instead of the $M_s + M_n$ values required by the traditional model defined in Equation (23). Moreover, in the eyes of the state space-based algorithms to be run, the observation equation still only contains white Gaussian noise, and thus overall very few changes need to be made to an existing algorithm to "upgrade it" to colored noise.

To gain some insight on the differences between the two methods, let us assume that the speech production is linear; we now write the details of a KF iteration using both the regular/traditional system and the modified one. The details are shown in Table 13 below. For simplicity and clarity, the time instant k is dropped, and the iteration is written in purely

algorithmic form. In addition, the letters $\mathbf{P}_s$ and $\mathbf{P}$ are used to denote the covariance matrices of the state vectors $\mathbf{s}$ and $\mathbf{x}$ (respectively) to be updated. The other letters used inside the iteration, but which have not been defined above are temporary variables introduced for clarity. For the proposed algorithm, the variable u is used to denote the term which at iteration k is equal to $\mathbf{b}(k)^T\mathbf{z}(k-1)$, and it is assumed to be computed at line 4. Based on this algorithm, more analysis is conducted in the next sections.

KF iteration for colored noise model, Traditional vs. proposed method

Left column: update of $[\mathbf{s}, \mathbf{P}_s]$; right column: update of $[\mathbf{x}, \mathbf{P}]$

Line	Traditional/regular method	Proposed method
1	$\mathbf{P}_{s_t} = \mathbf{F}\mathbf{P}_s\mathbf{F}^T + \mathbf{H}\mathbf{H}^T$	$\mathbf{P}_t = \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{G}\mathbf{G}^T$
2	$t_s = \mathbf{D}\mathbf{P}_{s_t}\mathbf{D}^T$	$t = \tilde{\mathbf{C}}\mathbf{P}_t\tilde{\mathbf{C}}^T + \sigma_v^2$
3	$\mathbf{s}_t = \mathbf{F}\mathbf{s}$	$\mathbf{x}_t = \mathbf{A}\mathbf{x}$
4	$y_s = \mathbf{D}\mathbf{s}_t$	$y = \tilde{\mathbf{C}}\mathbf{x} + u$
5	$\mathbf{J}_s = \mathbf{P}_{s_t}\mathbf{D}^T t_s^{-1}$	$\mathbf{J} = \mathbf{P}_t\tilde{\mathbf{C}}^T t^{-1}$
6	$\mathbf{s} = \mathbf{s}_t + \mathbf{J}_s(z - y_s)$	$\mathbf{x} = \mathbf{x}_t + \mathbf{J}(z - y)$
7	$\mathbf{P}_s = \mathbf{P}_{s_t} - \mathbf{J}_s\mathbf{D}\mathbf{P}_{s_t}$	$\mathbf{P} = \mathbf{P}_t - \mathbf{J}\tilde{\mathbf{C}}\mathbf{P}_t$

Table 13: Details of the Kalman Filter iteration in colored noise; Traditional vs. proposed method

5.3. Comparative computational and memory load

The goal of this section is to analyze the amount of computations and memory required to perform one iteration of each algorithm. It is important to first note that many of the algebraic operations mentioned above are simply assignment operations. For example, in line 3 of the proposed algorithm, only the first component of $\mathbf{x}_t$ must be computed, and the next $M_s - 1$ ones are the first $M_s - 1$ ones in $\mathbf{x}$. In addition, it is assumed that the matrices $\mathbf{P}_s$, $\mathbf{P}_{s_t}$, $\mathbf{P}$, and $\mathbf{P}_t$ are symmetric, so that not all entries must be computed. This in turn allows for the following: In both algorithms, in the computation of the variable t_s and t at line 2, it is

advantageous to temporarily store the value of the vector $\mathbf{DP}_{s_t}$ (resp. $\widetilde{\mathbf{C}}\mathbf{P}_t$) which is then used again at lines 5 (since $\mathbf{P}_{s_t}\mathbf{D}^T = \mathbf{DP}_{s_t}$ due to the symmetry of $\mathbf{P}_{s_t}$) and 7. Therefore, in the following analysis, it is assumed that after line 2, the value of $\mathbf{DP}_{s_t}$ (resp. $\widetilde{\mathbf{C}}\mathbf{P}_t$) is available in some temporary memory location.

Finally, the inversion of the variables t_s and t (5^{th} lines) is not counted – it has the same cost for both algorithms. A breakdown of the computational load in each line of both algorithms is shown in Tables 14 and 15. Line 1 of both algorithms contains the bulk of the number of multiplications – complete details can be found in [MUS'08].

Line	Multiplications	Additions/Subtractions
1	$M(M+1)+3$	$(M+1)(M-2)+3$
2	0	$M+1^{(*)}$
3	M	$M-2$
4	0	1
5	M	0
6	M	$M+1$
7	$\frac{M(M+1)}{2}$	$\frac{M(M+1)}{2}$
Total	$\frac{3}{2}(M+1)(M+2)$	$\frac{1}{2}(3M^2+5M+4)$

(*) This is necessary because $\mathbf{DP}_{s_t}$ is used at Lines 5 and 7.

Table 14: Computational load for the regular KF iteration. The letter M here represents $M_s + M_n$.

Line	Multiplications	Additions/Subtractions
1	$M_s(M_s+1)+1$	M_s^2
2	$M_n(M_s+1)+1$	$M_n(M_s+1)+1$
3	M_s	M_s-1
4	$2M_n$	$2M_n$
5	M_s	0
6	M_s	M_s+1
7	$\frac{M_s(M_s+1)}{2}$	$\frac{M_s(M_s+1)}{2}$
Total	$\frac{1}{2}(2M_sM+M_s^2+6M$ $+3M_s+4)$	$(M_s+1)(M+\frac{1}{2}Ms)$ $+M+M_n+1$

Table 15: Computational load for the proposed KF iteration.

From these tables, it is apparent that the decrease in computations is quite significant. For the regular KF step, the number of multiplications is $O(M^2)$, and for the proposed one, it is $O(M_sM)$. A few examples evaluated for concrete values of M_s and M_n are shown in Table 16.

Parameters		Regular			Proposed		
M_s	M_n	×	+	−	×	+	−
8	3	234	144	67	167	113	37
10	3	315	196	92	236	159	56
10	6	459	289	137	275	198	56
12	6	570	361	172	362	258	79
12	8	693	441	211	392	288	79

Table 16: Examples of computational load for both types of KFs.

In terms of required memory, the proposed iteration is also advantageous. Each must store the state vector and its covariance matrix (which is symmetric): for the regular iteration, respectively M_s+M_n and $\frac{(M_s+M_n)(M_s+M_n+1)}{2}$ memory locations are used. In the modified one, one must also keep a trail of M_n measurements: This amounts to M_s+M_n and $\frac{M_s(M_s+1)}{2}$ (if $M_s>M_n$ again) required operations. In algorithms such as Rao-

Blackwellized Particle Filters, where multiple KFs are used but share the same measurements, the gain in memory can thus be very significant.

5.4. Simulations

In order to test the above idea, several algorithms from Section 3.2.4, augmented to handle a colored autoregressive noise, are chosen to be used: the DKF, the KEM_{Burg}, and the RBPF (one algorithm from each of the "subgroups" defined in Section 3.2.4).

For each algorithm, we use online estimation of the noise AR coefficients, obtained via the solution of the Yule-Walker equations, using the noise estimation algorithm cited in Section 3.3.

Two sets of results are presented:

- First, for each algorithm the average differences in scores between the proposed handling of colored noise and the traditional one are reported for each individual algorithm. For example, "+1.96" under the SNR column for the RBPF algorithm indicates that the use of the proposed colored noise handling improves the SNR by 1.96 dBs.
- Secondly, comparative results between each algorithm (using the proposed colored noise handling scheme) are shown and discussed.

Once again, the methodology follows the one dictated in Chapter II – where the speech and noise material can also be found. The complete tables of results can be found in Appendix B.

5.4.1. Traditional vs. proposed colored noise handling: results for individual algorithms

Table 17 shows the effect of resorting to the proposed approach on each of the three tested algorithms.

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
DKF	+0.13	+0.09	+0.02	0.00	+0.02	+0.02	+0.01
KEM_{Burg}	+0.01	0.00	0.00	0.00	0.00	0.00	0.00
RBPF	+1.96	+0.70	+0.08	+0.02	+0.04	+0.10	+0.04
M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
DKF	+0.05	+0.05	+0.01	+0.01	+0.01	+0.01	+0.01
KEM_{Burg}	+0.01	0.00	0.00	0.00	0.00	0.00	0.00
RBPF	+1.05	+0.31	+0.02	+0.07	+0.04	+0.04	+0.03

Table 17: Average differences in scores obtained for individual algorithms in colored noise. The numbers have been obtained as an average across the 4 types of noise, separately for VL/L and M/H conditions. As an example explaining how to read the entries in this table: "+1.96" under the SNR column for the RBPF algorithm indicates that the use of the proposed colored noise handling improves the SNR by 1.96 dBs.

The initial intent was essentially to propose a less complex and redundant way of dealing with autoregressive colored noise, and the most important criterion for accepting the proposed handling was that the output quality was at least equivalent. Observing Table 17, one can conclude that this criterion was not only met, but in fact that the expectations were exceeded since the observed changes are only positive. For the DKF and KEM_{Burg} algorithm, the changes are marginal and practically unperceivable, however in the case of the RBPF, one can hear some improvements in terms of noise reduction, especially at low SNR.

The explanation for this observation is that the performance of a PF-based algorithm is known to be related to the dimension of the state vector: since the proposed method operates in a smaller dimension, such improvements in terms of objective scores are to be expected.

In our related publication [MUS'08], similar results are discussed and found for two extra cases: a "test" plain KF in which all parameters (speech and noise) are considered to be known in advance, and the KEM algorithm. Execution times are reduced and no apparent loss in quality is observed in the tests performed. Concerning the "plain KF" algorithm, which as

used here can be interpreted as representative of a scenario in which solid speech and noise parameter estimates are available, the use of the proposed model is particularly beneficial.

These observed properties therefore make the proposed method attractive for any cost-sensitive application, such as hearing aids or personal recording devices.

5.4.2. Comparative results between several algorithms for the proposed noise handling

From the set of simulations used above, one can also extract comparisons between the three algorithms in colored noise. The rankings (see Chapter II for complete details regarding experimental conditions, objective measures and algorithms ranking procedures) are shown in Table 18 below.

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**RBPF**	**RBPF**	**RBPF**	KEM_{Burg}	**RBPF**	**RBPF**	**RBPF**
	DKF	DKF	KEM_{Burg}	**RBPF**	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}
Worst	KEM_{Burg}	KEM_{Burg}	DKF	DKF	DKF	DKF	DKF
M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**RBPF**	**RBPF**	**RBPF**	**RBPF**	**RBPF**	**RBPF**	**RBPF**
	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}	KEM_{Burg}
Worst	DKF	DKF	DKF	DKF	DKF	DKF	DKF

Table 18: Algorithm rankings for the three tested algorithms in colored noise, separately for VL/L and M/H conditions. The RBPF (in bold characters) yields the most entries in the top line for all conditions, and is therefore considered the best algorithm.

Perhaps unsurprisingly, the rankings obtained for the WGN conditions (see Section 3.2.5) carry over to the colored noise situation. Indeed, the RBPF still leads for both the VL/L and M/H conditions, followed by the KEM_{Burg} and then by the DKF.

From informal listening tests however, as in the WGN case, it is still found that it is difficult to tell the three algorithms apart, except at higher SNR where the RBPF noticeably stands out. But it is clear that each algorithm belongs to the same family of methods, in that their features are very similar.

5.5. Conclusions

In this Chapter, a novel handling of colored noise cases in state-space based algorithms was proposed, with multiple advantages. As a first advantage, the resulting equations naturally reduce to the white noise case when a zero-order noise model is chosen. Next, the main attraction of the alternative shown is that it allows for a faster implementation. Finally, in addition no apparent loss in quality is observed in the tests performed – in fact, some improvement can be heard in several cases, most notably the RBPF algorithm. These properties make the proposed method attractive for any cost-sensitive application, such as hearing aids or personal recording devices. Accordingly, it is now assumed that the "improved" colored noise handling is used as a standard throughout the remainder of the thesis.

Additionally, based on multiple tests, we find that the performance hierarchy established in the white Gaussian noise case carries over to the colored noise case.

VI. A particle filtering framework for nonlinear speech models

The work presented in this chapter, which is a study of particle filtering solutions in the context of nonlinear, neural-type speech models, has been published in IEEE Transactions on Neural Networks in 2009 (see [MUS'09]).

6.1. Introduction

The recent successful applications of particle filtering algorithms to speech enhancement is now considered an important trend [EPH'06], as demonstrated by the growing literature on the subject (see for example [VER'02, FON'02, GOD'04, PAR'06, SHU'05, DEN'06, WIN'06]). They have recently been reported to challenge well-established algorithms such as the one shown in [EPH'85] and others under several conditions (see for example our work in [MUS'07]). Effectively, all the above-cited PF papers employ a linear speech production model, that is, identical or equivalent to a time-varying autoregressive model (see Section 3.1), at the basis of the PF algorithms (except for [SHU'05] discussed below). In spite of the solid performance reported under such linear models, there is strong theoretical and experimental evidence that the phenomenon of speech production contains important nonlinearities [FAU'02, THY'94, MA'98]. For example, a case for the use of nonlinear prediction models is presented in [CHE'05], where it is conjectured from theoretical considerations that the introduction of nonlinearities in the speech model should only be beneficial.

Coupled with the fact that particle filters are open to nonlinear models, the above is an invitation to experimentation, specifically to hopefully bring answers to the following

questions: From a practical standpoint, given the reportedly promising results of PFs on linear speech models, is it worth it to migrate/upgrade to PFs on nonlinear speech models? Recall some of the observations made in Section 3.2.5, where it was observed that in white Gaussian environment noises, there is no significant gain between several KF-based algorithms running on linear speech models and their EKF counterpart running on nonlinear models. However, some perceivable improvement was still observed when dual UKF-based algorithms were used on nonlinear models (noticeably at high SNR). A possible explanation is that no improvement was seen between the KF and EKF algorithms perhaps because of the underlying simplifications intrinsic to the EKF method (namely, Gaussian assumptions and especially approximated nonlinearities), whereas with the UKF-based technique – which does not linearize the system – we start to see some improvements. It is therefore appealing to evaluate what can be obtained in the particle filtering context, since there is then no simplifying assumption. Another important motivation is the following: from a theoretical standpoint, is there any way to build a generic framework, for non-white noise and nonlinear models, which would reduce to several existing PF algorithms, that is, in white-noise/linear-model cases? To answer these questions, in this chapter different PF algorithms applied to variations of a predictive feedforward neural network (NN) model for speech production are derived and tested.

There already exist in the literature several non-PF algorithms capable of running on such neural-type models. One of the most notable examples is the powerful family of dual Extended Kalman Filters (DEKFs), recapitulated in Section 3.2. Recall that in the DEKF algorithm, for the white noise case one EKF is given the task to estimate the clean speech signal given the NN weights, while another one must estimate the NN weights given the clean speech signal. In the colored-noise case [NEL'97, WAN'98], the state vectors of both EKFs

are augmented and include a trail of noise samples, assumed to follow an autoregression. Besides aiming at answering the two main questions in the last paragraph, the strong reported results of the dual EKFs can be viewed as an extra source of motivation in the following sense. First, in numerous situations, well designed PFs with sufficient computational resources have been shown to outperform EKFs (see for example a recent application in [MIH'07]). Secondly, this raises the question of the benefits of using duality in the context of a PF solution, since PFs are not theoretically bound to resort to it.

Interestingly, in a recent work presented in [SHU'05], titled "Neural dual Particle Filter and its application to speech enhancement", a partial investigation of the very problem posed here is conducted, in fact following closely the outline of the dual EKF paper [NEL'97]. However, in several aspects the amount of information and justification disclosed in [SHU'05] is not sufficient to attain stable conclusions. More specifically, the paper introduces an algorithm where two PFs are run in parallel for the separate estimation of the speech model parameters on the one hand, and the actual clean speech on the other hand. Yet there is no case for the duality employed, and the question eventually remains: why resort to duality when "nonduality" is not only directly possible, but by contrast also theoretically supported? Moreover, the dual algorithm presented is only done so in very generic terms, in the sense that there is limited information about the specifics of the model used, and the actual particle filters used for each sub-PF in the dual algorithm are not explicited. In addition, the accompanying tests results, which are performed on a few segments of 2-4 seconds with the average segmental SNR measure, can only lead to partial conclusions: since PFs are "stochastic" algorithms an average result over a large amount of tests is necessary. Moreover, comparisons to PFs running on linear models, as well as "non-dual" PFs running on nonlinear models, are also necessary to get a proper view of their respective benefits.

In this chapter, detailed particle filtering algorithms are derived according to the nonlinear speech production model shown in Equation (4) in Section 3.1.2 (this equation is repeated below). "Dual" counterparts of the algorithms, in which two particle filters run in parallel while exchanging information, are presented as well, along with their expected benefits and disadvantages. The duality follows the generic idea of [SHU'05], but the distribution of tasks and the detailed description of the nature of each PF are proper to this thesis. In essence, the idea is that one of the PFs tries to determine the model parameters while the other one estimates the clean speech signal, and each of the PFs rely on each other's current estimate. The main state-space model used in this chapter incorporates the colored noise in a similar manner as in Chapter V – reducing dimensionality, redundancy and complexity – compared with augmenting the state vector as in [SHU'05, NEL'97, WAN'98]. The methods shown are designed to readily handle white stationary/nonstationary noises and stationary colored noises, and in the next chapter (Chapter VI) some methods to handle nonstationary real-world noises are presented.

The goal of this chapter is thus to introduce a framework for the use of particle filters on a class of nonlinear speech models, and then assess whether or not the introduction of the nonlinearity is beneficial, and also whether it is useful to resort to duality. The dual and non-dual algorithms are both fully derived in Sections 6.3 and 6.4 respectively, and a discussion on their respective computational complexity is given in Section 6.5. Then, they are thoroughly tested in section 6.6. In contrast with [SHU'05], each configuration is individually tested in order to assess respective benefits, and the conclusions will be drawn based on simulation results obtained following the experimental procedure in Chapter II. Also, for comparison results obtained with the RBPF algorithms running on linear speech models are

included. In subsequent chapters, the performance of this class of PFs will be evaluated against other algorithms in real-world noise environments. In addition, because of the standard methodology employed, the reader can also easily judge the performance of the neural-based PFs with any other results found in this thesis.

6.2. The proposed speech and noise models

6.2.1. Speech production model

The speech production model is precisely the one that has previously been described in Section 3.1.2, in Equation (4), that is:

$$x(k) = \mathbf{c}(k)^T f(\mathbf{W}(k)\mathbf{x}(k-1) + \mathbf{d}(k)) + \sigma_g(k)g(k) \tag{25}$$

where:

- P is the number of neurons in the model,
- $\mathbf{W}(k)$ is a P by M_s matrix, representing the internal coefficients of the network,
- $\mathbf{c}(k)$ is a length P column vector denoting the output coefficients of the network,
- $\mathbf{d}(k)$ is a length P column vector denoting the bias inputs at each of the neurons in the network,
- $f(.)$ is the nonlinear activation function of the neurons

It is illustrated in Figure 9 below.

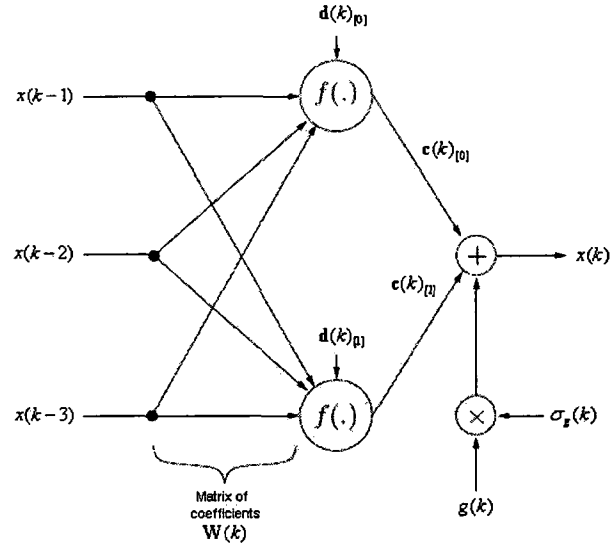

Figure 9: Nonlinear speech production model. The input data $x(k-1)$, $x(k-2)$, $x(k-3)$ make up the previous speech vector $\mathbf{x}(k-1)$, from which the next sample $x(k)$ must be predicted.

To complete the definition of the speech production model, a choice must be made regarding the evolution in time of the coefficients $\mathbf{W}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, and $\sigma_g(k)$. For lack of a more informed, theory-based evolution model, the choice made in this thesis is that of independent Gaussian random walk for these parameters, with:

- $\mathbf{W}(k) \sim N(\mathbf{W}(k-1); {\sigma_{\mathbf{W}}}^2)$
- $\mathbf{c}(k) \sim N(\mathbf{c}(k-1); {\sigma_{\mathbf{c}}}^2)$
- $\mathbf{d}(k) \sim N(\mathbf{d}(k-1); {\sigma_{\mathbf{d}}}^2)$
- $\log(\sigma_g(k)) \sim N(\log(\sigma_g(k-1)); {\sigma_{L_x}}^2)$

where $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{c}}$, $\sigma_{\mathbf{d}}$, and σ_{L_x} are constants that are known a priori. Thus, the only a priori knowledge about $\mathbf{W}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, and $\log(\sigma_g(k))$ is related to how much they are allowed to vary from one iteration to the next, but nothing is known about their actual values.

Even though random walks are very practical (they are used in the majority of PF-based speech enhancement papers), the equations above constitute the "weakest link" of the speech production model, because it assumes that each element of the state is then drawn independently of each other (that is, regardless of the current NN weights and configuration). Some ideas for improvement are discussed in the conclusion of this chapter. Regardless, as it will be noted later, the nondual algorithms derived in this chapter can readily handle many different evolution models for $\mathbf{W}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, and $\sigma_g(k)$.

For clarity, define $L_x(k) = \log(\sigma_g(k))$; we can then write from the above that $L_x(k) \sim N(L_x(k-1); {\sigma_{L_x}}^2)$. From this generic model, different variations are considered in this chapter:

- **Single-neuron** versus **multiple-neurons** models. In the single-neuron model, $P = 1$ and $\mathbf{c}(k) \equiv \mathbf{c}(k)_{[0]}$ is considered to be known and equal to 1. In the multiple-neurons model, obviously $P > 1$.
- **Biased** versus **non-biased** models. In the non-biased model, $\mathbf{d}(k)$ is set to 0.

6.2.2. Noise and measurement models

The noise and measurement models can be written together in the observation equation, using the results of Chapter V as follows:

$$z(k) = \widetilde{\mathbf{C}}_k \mathbf{x}(k) + \mathbf{b}(k)^T \mathbf{z}(k-1) + \sigma_v(k) v(k) \tag{26}$$

where $\widetilde{\mathbf{C}}_k = \begin{bmatrix} 1 & -\mathbf{b}(k)^T & 0_{1 \times M_s - M_n - 1} \end{bmatrix}$, and again $\mathbf{b}(k)$ is the length-M_n noise autoregression vector, and $\mathbf{z}(k) = \begin{bmatrix} z(k) & z(k-1) & \cdots & z(k-M_n+1) \end{bmatrix}^T$.

In the context of this chapter, as for the speech parameters an evolution model is assumed for the noise autoregressive parameters with:

- $\mathbf{b}(k) \sim N(\mathbf{b}(k-1); {\sigma_{\mathbf{b}}}^2)$
- $\log(\sigma_v(k)) \sim N(\log(\sigma_v(k-1)); {\sigma_{L_v}}^2)$, or $L_v(k) \sim N(L_v(k-1); {\sigma_{L_v}}^2)$ with $L_v(k) = \log(\sigma_v(k))$

These parameters are provided for completeness, however when $M_n > 0$ we also assume that some external background noise estimation algorithm (see Section 3.3) is running and regularly provides some information from which estimates of $\mathbf{b}(k)$ and $\sigma_v(k)$ can be extracted (e.g. using one of the methods given in the upcoming Chapter VI), effectively helping the enhancement algorithm to properly discriminate the speech and noise-like sources in the noisy mixture.

6.3. Joint (non-dual) algorithms

First, the actual state vector that is to be used in the PF algorithms, denoted here by $\chi(k)$, must this time contain more than the mere trail of speech samples. Specifically, we set: $\chi(k) = [\mathbf{x}(k) \quad \mathbf{W}(k) \quad \mathbf{c}(k) \quad \mathbf{d}(k) \quad L_x(k) \quad L_v(k)]$ (note that $L_v(k)$ may not be included, for example in the colored noise case, or simply in the case where the noise statistics are known. Either way, the impact on the resulting algorithm will be seen to be marginal).

Next, an importance function $q(.)$ must be chosen, and the weights computation equation in the PF algorithm must be derived accordingly. In many PF implementations, the so-called transitional prior is used for simplicity, i.e.:

$$q(\chi(k)\,|\,\chi(k-1),z(k)) = p(\chi(k)\,|\,\chi(k-1)) \tag{27}$$

Unfortunately, with such a density the particles are drawn regardless of the measurements, and this may result in poor overall performance [RIS'04]. The literature contains several efforts to devise better importance functions, depending on the application and model (see for example [ODO'06, MER'00]). In speech enhancement, the latest measurement has been included with success in a factor of the importance function in [GOD'04] and in Section 7.4.1 of [DOU'01], in the context of a linear predictive speech model. In a similar manner, in this Chapter the current measurement is incorporated as follows:

$$\begin{aligned} q(\chi(k)\,|\,\chi(k-1),z(k)) = {} & p(\mathbf{x}(k)\,|\,\mathbf{x}(k-1),\mathbf{W}(k),\mathbf{c}(k),\mathbf{d}(k),L_x(k),L_v(k),z(k)) \\ & \times p(\mathbf{W}(k)\,|\,\mathbf{W}(k-1))p(\mathbf{c}(k)\,|\,\mathbf{c}(k-1))p(\mathbf{d}(k)\,|\,\mathbf{d}(k-1)) \\ & \times p(L_x(k)\,|\,L_x(k-1))p(L_v(k)\,|\,L_v(k-1)) \end{aligned} \tag{28}$$

For readability, the following notation is introduced:

$$\begin{aligned} &\Xi(k) = [\mathbf{W}(k);\mathbf{c}(k);\mathbf{d}(k)] \\ &\Sigma(k) = [L_x(k);L_v(k)] \\ &r(k) = \mathbf{c}(k)^T f(\mathbf{W}(k)\mathbf{x}(k-1)+\mathbf{d}(k)) \\ &s(k) = \mathbf{b}(k)^T(\mathbf{z}(k-1)-\tilde{\mathbf{x}}(k-1)) \end{aligned} \tag{29}$$

$$\text{where } \tilde{\mathbf{x}}(k) = [x(k) \quad x(k-1) \quad \ldots \quad x(k-M_n+1)]^T$$

Only the first term in the product forming the right-hand side of Equation (28) must be derived – all the other terms directly follow from the speech and noise production models, with parameters' evolution defined as Gaussian random walks. In addition, since only the first component of $\mathbf{x}(k)$ is stochastic (given $\mathbf{x}(k-1)$), then it can be written that $p(\mathbf{x}(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k),z(k)) = p(x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k),z(k))$. Now, from elementary rules:

$$
\begin{aligned}
&p(x(k)\,|\,\mathbf{y}(k-1),\Xi(k),\Sigma(k),z(k)) \\
&\quad=\frac{p(x(k),z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))}{p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))} \\
&\quad=\frac{p(z(k)\,|\,x(k),\mathbf{x}(k-1),\Sigma(k))p(x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))}{p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))} \\
&\quad=\frac{N(z(k)\,|\,x(k)+s(k);\sigma_v(k)^2)N(y(k)\,|\,r(k);\sigma_g(k)^2)}{p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))}
\end{aligned}
\tag{30}
$$

Consider first the numerator of the last line in Equation (30). To simplify it, the following straightforward formula can be used (where each letter represents arbitrary numbers):

$$
N(x\,|\,a;\sigma_a{}^2)N(x\,|\,b;\sigma_b{}^2)=N(a\,|\,b;\sigma_a{}^2+\sigma_b{}^2)N(x\,|\,c;\sigma_c{}^2) \tag{31}
$$

where $\sigma_c{}^2=\left(\sigma_a{}^{-2}+\sigma_b{}^{-2}\right)^{-1}$ and $c=\sigma_c{}^2\left(a\sigma_a{}^{-2}+b\sigma_b{}^{-2}\right)$.

Noting that $N(z(k)\,|\,x(k)+s(k);\sigma_v(k)^2)=N(x(k)\,|\,z(k)-s(k);\sigma_v(k)^2)$, it can be seen that Equation (30) becomes:

$$
\begin{aligned}
&p(x(k)\,|\,\mathbf{y}(k-1),\Xi(k),\Sigma(k),z(k)) \\
&\quad=\frac{N(z(k)-s(k)\,|\,r(k);\sigma_v(k)^2+\sigma_g(k)^2)N(y(k)\,|\,\alpha(k);\theta(k)^2)}{p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))} \\
&\quad=\frac{N(z(k)\,|\,r(k)+s(k);\sigma_v(k)^2+\sigma_g(k)^2)N(y(k)\,|\,\alpha(k);\theta(k)^2)}{p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))}
\end{aligned}
\tag{32}
$$

where $\theta(k)^2=\left(\sigma_v(k)^{-2}+\sigma_g(k)^{-2}\right)^{-1}$ and $\alpha(k)=\theta(k)^2\left(\frac{z(k)-s(k)}{\sigma_v(k)^2}+\frac{r(k)}{\sigma_g(k)^2}\right)^{-1}$

Consider now the denominator of (30). We have:

$$
p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))=\int p(z(k),x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))dx(k) \tag{33}
$$

The integrand above is therefore no different from the numerator of Equation (32). After integration with respect to $x(k)$, only the term $N(z(k)\,|\,r(k)+s(k);\sigma_v(k)^2+\sigma_g(k)^2)$ remains in the denominator since it is independent of $x(k)$ and the other term $N(x(k)\,|\,\alpha(k);\theta(k)^2)$ sums to 1. Therefore, with $N(z(k)\,|\,r(k)+s(k);\sigma_v(k)^2+\sigma_g(k)^2)$ appearing both in the numerator and denominator of Equation (32), it is clear that Equation (30) can be simplified as:

$$p(x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k),z(k)) = N(x(k)\,|\,\alpha(k);\theta(k)^2) \tag{34}$$

with $\theta(k)$ and $\alpha(k)$ defined above.

Equation (34), along with Equation (28), explicitly defines the proposed importance function. It remains to establish the corresponding weight update equation. Referring to the weight update equation given in Algorithm 4 (in Appendix A) and dropping the superscript (i) for convenience, we have:

$$\begin{aligned}
\tilde{w}(k) &= w(k)\frac{p(z(k)\,|\,\chi(k))p(\chi(k)\,|\,\chi(k-1))}{q(\chi(k)\,|\,\chi(k),z(k))} \\
&\propto \frac{p(z(k)\,|\,x(k),\mathbf{x}(k-1),\Sigma(k))p(x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k))}{p(x(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k),z(k))} \\
&\propto p(z(k)\,|\,\mathbf{x}(k-1),\Xi(k),\Sigma(k)) \\
&\propto N(z(k)\,|\,r(k)+s(k);\sigma_v(k)^2+\sigma_g(k)^2)
\end{aligned} \tag{35}$$

As it turns out, the importance function calculation and the weight update are relatively simple and contain common terms. The resulting algorithm is quite compact and presented in Algorithm 6, in Appendix A. Note that the algorithm naturally reduces to the white noise case – by setting $M_n = 0$ (which is the noise autoregression order), then $s(k) = 0$ and the rest remains unchanged.

In addition, the single-neuron version of Algorithm 6 is a simplified version of the above, where $\mathbf{c}(k)$ is identical to 1 and $P = 1$ ($\mathbf{W}(k)$ is a vector), and thus there is no need to draw $\mathbf{c}(k)$ particles.

Observe that from Algorithm 6, one can obtain a linear, white noise speech enhancement algorithm, as presented in Section 7.4 of [DOU'01], also used in [GOD'04]. To do so, it suffices to set $\mathbf{d}(k) = 0$, $P = 1$ (and $\mathbf{c}(k) = 1$), $M_n = 0$, and choose $f(x) = x$. Although the weight update equation presented in [GOD'04] is formulated in a more complex manner, it can in fact be shown that it is identical to the simpler one presented above. Moreover, the experienced user intending to use a linear model ($f(x) = x$ and $P = 1$) will have foreseen that a fuller Rao-Blackwellization is then possible (including the actual speech samples). This full Rao-Blackwellization is described for the white noise case in [FON'02], and for the colored noise case it can be found in [DEN'06][6]. Although Algorithm 6 does not resort to Rao-Blackwellization, it is interesting to note that because of the approach taken, it has the advantage of dealing with a state vector of smaller dimension than that employed in the subband RBPFs for colored noise running on linear models described in [DEN'06], which also include a trail of noise samples in their state vectors. The literature contains other colored-noise solutions resorting to RBPFs, but they are not in direct relation to our work: For example, in [PAR'06] a form of Expectation-Maximization algorithm is used to update both speech and noise model parameters while an RBPF is used to update clean speech estimates, based on a linear-predictive model driven by a generalized exponential excitation noise. The results are encouraging but only reported for very small segments. Note again that, as for [DEN'06], the state vector used in [PAR'06] has a higher dimension than the one that would be required from using our equivalent model with $f(x) = x$ and $P = 1$.

[6] [DEN'06] in fact uses them as part of a subband solution, with one such colored noise RBPF for each subband, and employs a form of voice activity detection to update the noise autoregressive parameters

Finally, one can directly see how in Algorithm 6, different evolution models for all speech parameters could be used as well: as long as the new draws do not depend on $x(k)^{(i)}$, then in fact any transitional model can be used with no impact on the rest of the algorithm.

6.4. Dual algorithms

In this case, it is proposed to run two PFs concurrently, as stated in generic terms in [SHU'05]. By doing so, it is initially hoped that dividing the estimation process into two parts will achieve a better overall performance, principally based on the following conjectures: First, each particle filter will effectively have a lighter workload, and thus may perform better. Secondly, with a proper layout, it will be possible to have one PF using an optimal importance distribution, while the other lends itself naturally to Rao-Blackwellization, and thus again a higher accuracy may be expected. As it will be seen, the above conjectures will however be challenged when simulation results are unveiled. There are many ways of distributing parts of the state to estimate between the two PFs. Unfortunately, [SHU'05] does not discuss the actual procedure/algorithm used, and details about such task distribution problems are omitted as well. In this thesis, the following distribution is chosen: The first PF, labelled by the letter *A*, is used to estimate the clean speech $x(k)$, and the second PF, labelled by the letter *B*, bears the task of estimating all the speech parameters, i.e., the coefficients of the neural network model, the variance of the excitation noise, and if necessary, the noise parameters (autoregressive vector and variance of the observation noise). The rationale behind this choice will be apparent below. The first set of equations related to PF-*A* is:

$$\begin{aligned} x(k) &= \hat{\mathbf{c}}(k)^T f(\hat{\mathbf{W}}(k)\mathbf{x}(k-1) + \hat{\mathbf{d}}(k)) + \hat{\sigma}_g(k)g(k) \\ z(k) &= x(k) + \mathbf{b}(k)^T(\mathbf{z}(k-1) - \tilde{\mathbf{x}}(k-1)) + \hat{\sigma}_v(k)v(k) \end{aligned} \tag{36}$$

where all letters marked by a circumflex accent are estimates provided by PF-*B*. Next, for PF-*B* the state vector is composed of $\mathbf{W}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, $L_v(k)$, and $L_x(k)$ – all assumed to follow Gaussian random walks, and the measurement equation is written as:

$$z(k) = \mathbf{c}(k)^T f(\mathbf{W}(k)\hat{\mathbf{x}}(k-1) + \mathbf{d}(k)) + \mathbf{b}(k)^T(\mathbf{z}(k-1) - \hat{\tilde{\mathbf{x}}}(k-1)) + \sigma_g(k)g(k) + \sigma_v(k)v(k) \tag{37}$$

where $\hat{\mathbf{x}}(k)$ and $\hat{\tilde{\mathbf{x}}}(k)$ are returned by PF-*A*. The measurement Equation (37) is written as such in order to ensure that all drawn quantities are directly related to the measurement.

At the reception of a measurement $z(k)$, the above imposes the following order: First, PF-*B* estimates new speech parameters based on the previous speech estimate $\mathbf{x}(k-1)$, provided by PF-*A*, and on $z(k)$. Next, these parameters are passed to PF-*A* which uses them to estimate $x(k)$. This scheduling is described graphically in Figure 10.

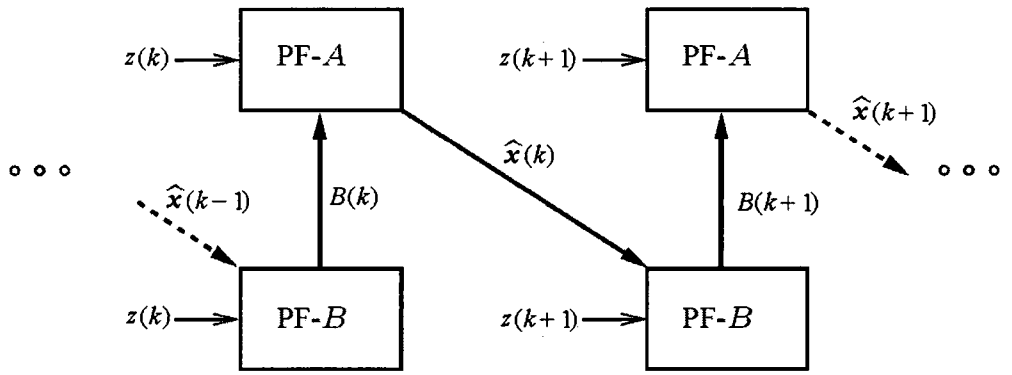

Figure 10: Scheduling scheme for the dual particle filter proposed. In this figure, $B(k) = \left[\hat{\mathbf{W}}(k); \hat{\mathbf{c}}(k); \hat{\mathbf{d}}(k); \hat{L}_x(k); \hat{L}_v(k)\right]$

The reason behind the particular distribution of quantities to estimate between PF-*A* and PF-*B* is twofold: First, PF-*A* can use an optimal importance density. Secondly, since the system on which PF-*B* operates is conditionally linear-Gaussian, then Rao-Blackwellization is possible

and convenient on $\mathbf{c}(k)$. In fact, if we let $\mathbf{x}_{2k} = \mathbf{c}(k)$, then the generic RBPF algorithm (Algorithm 5 from Appendix A) can be directly applied.

Let us now provide some implementation details about each of the PFs:

- For PF-*A*: The state vector is only composed of $\mathbf{x}(k)$. Following the steps shown in the previous section (for the joint case), it is easy to map the previously derived equations to this simplified context, in order to obtain the optimal importance function:

$$p(x(k) \mid \mathbf{x}(k-1), z(k)) = N(x(k) \mid \hat{\alpha}(k); \hat{\theta}(k)^2) \tag{38}$$

where:

$$\begin{aligned}
\hat{\theta}(k)^2 &= \left(\hat{\sigma}_v(k)^{-2} + \hat{\sigma}_g(k)^{-2}\right)^{-1} \\
\hat{\alpha}(k) &= \hat{\theta}(k)^2\left((z(k) - s(k))\hat{\sigma}_v(k)^{-2} + \hat{r}(k)\hat{\sigma}_g(k)^{-2}\right) \\
\hat{r}(k) &= \hat{\mathbf{c}}(k)^T f(\hat{\mathbf{W}}(k)\mathbf{x}(k-1) + \hat{\mathbf{d}}(k)) \\
s(k) &= \mathbf{b}(k)^T(\mathbf{z}(k-1) - \tilde{\mathbf{x}}(k-1))
\end{aligned}$$

Note that $\hat{\theta}(k)$ is common to all particles in this case. Similarly, the weight update equation can be directly written from the results of the previous section:

$$\tilde{w}_A(k) = N(z(k) \mid \hat{r}(k) + \hat{s}(k); \hat{\sigma}_v(k)^2 + \hat{\sigma}_g(k)^2) \tag{39}$$

- For PF-*B*: Using Algorithm 5 as a template, denote $\mathbf{x}_{1k} = \{\mathbf{W}(k); \mathbf{d}(k); L_x(k); L_v(k)\}$ and $\mathbf{x}_{2k} = \mathbf{c}(k)$. From this point, the derivation of the RBPF sub-algorithm is obvious.

The resulting algorithm is shown in Algorithm 7 in Appendix A, where $\mathbf{K}(k)$ represents the covariance matrix of the Gaussian distribution $p(\mathbf{c}(k) \mid \mathbf{W}(k), \mathbf{d}(k), L_x(k), L_v(k), \mathbf{z}(0:k))$ (the

variable $\mathbf{x}(k)$ is not conditioned upon, since it is considered to be known from PF-*A*). Note that in the case of a single-neuron configuration, since there is no need to update $\mathbf{c}(k)$ the Kalman Filter steps are non-existent and the algorithm is simplified. In this case, the weight update equation reduces to:

$$\tilde{w}_B(k) = N(z(k) \mid f(\mathbf{W}(k)\tilde{\mathbf{x}}(k-1) + \mathbf{d}(k)); \sigma_g(k)^2 + \sigma_v(k)^2) \tag{40}$$

Finally note that the above algorithm is based on one of the many possible ways to "conduct" the dual estimation – not only in terms of the distribution of the parts of the state to estimate, but also in terms of the actual information exchanged by the two PFs: Only the current estimated mean of each quantity is communicated between the two PFs, but one can envision other more advanced solutions where multiple parameters can be exchanged between two particle filters (e.g. variance, modes, etc.).

6.5. Differences in computational complexity

Observing Algorithms 6 and 7, the number of computations per particle depends on which neural network structure is used, and what type of noise is considered. In either case, when the observation noise is white, then $s(k)^{(i)} = 0$ for all k and i, that is, effectively the computation of $s(k)^{(i)}$ disappears from the main loop, saving $2M_n - 1$ additions/subtractions and M_n multiplications per particle.

Using complex colored noises (large M_n values) therefore impacts each algorithm relatively equally. However, the largest factor in terms of complexity is the number of neurons used (the size P of the vector $\mathbf{c}(k)$). Indeed, increasing P makes the computation of the variable $r(k)^{(i)}$

more expensive in both algorithms, and in the dual algorithm it affects all steps of the Kalman filter stage in the sub-RBPF, thus having an even larger impact on the dual algorithm. It is therefore important to determine whether or not using $P>1$ is worth these extra computations.

For both algorithms, the lighter schemes are obtained with a single-neuron configuration without bias, and when the observation noise is white. In this case, $r(k)$ is equal to $f(\mathbf{W}(k)\mathbf{y}(k-1))$, and the nonlinear function $f(.)$ is applied to a single number.

In summary, comparing Algorithms 6 and 7 for a given configuration, one finds that the dual version is always more demanding per particle, but only slightly if only a single neuron and white noise are used (in which case, no Rao-Blackwellization takes place). On the other hand, if a more complex model is used, then the dual algorithm quickly becomes significantly more complex than the standard one since all the steps in the KF update are directly affected.

Finally, given the amount of parameters and the range of possibilities for implementation, it is difficult to compare the complexity of Algorithms 6 and 7 with the regular RBPF such as the one presented in Section 3.2. It is however clear that for a single neuron configuration, and in white noise environments, both Algorithms 6 and 7 necessitate significantly less operations per particle than the regular RBPF scheme. However, as with any particle filter algorithm in general, one must differentiate "theoretical" complexity and "practical" complexity. The first type refers to the amount of computations per particle – in this context, Algorithms 6 and 7 are unequivocally less complex. The second one reflects the overall complexity of the algorithm operating with enough particles, such that adding more particles would only marginally improve the results. RBPF algorithms are typically more robust and require less

particles to achieve a given accuracy, and so it is difficult to make any claims about the "practical" complexity of Algorithms 6 and 7 before conducting enough experiments. Observations about the "practical" complexity of these algorithms will thus be presented in the section below.

6.6. Experiments

For readability, the variants of the particle filter algorithms given in this section are coded in the following manner: **NPF-XYZ**, where **X**, **Y**, and **Z** are boolean variables which can take the following values:

- **X** is either D or $\bar{D}$, and indicates whether the algorithm is dual (D) or not ($\bar{D}$).
- **Y** is either M or $\bar{M}$, and indicates whether the model uses multiple neurons (M), or a single one ($\bar{M}$).
- **Z** indicates whether the models uses biases (B) or not ($\bar{B}$).

For example, the code **NPF-**$\bar{D}MB$ is the code for the nondual algorithm with multiple neurons and with biases.

6.6.1. Fixed parameter search and initialization

At this point, a crucial distinction must be made between certain components of the algorithms which we have been so far calling jointly "parameters", so as to avoid possible confusion regarding both the validity of the experiments and the practicality of the algorithms presented. Namely, the following two sets of quantities belong to a different category of parameters:

- The time-varying, *a priori* unknown parameters $\mathbf{W}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, $L_x(k)$
- The fixed, *a priori* known parameters M, P, $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{c}}$, $\sigma_{\mathbf{d}}$, σ_{L_x}.

In the first category, the "parameters" are managed in the PF algorithm as a set of particles used to estimate unknown quantities. While both a set of initial time-varying parameters and the fixed parameters must be chosen before the algorithm begins, our experience is that the initialization of the time-varying parameters has a very limited influence on the outcome of each algorithm[7]. In our implementation, each particle of $\mathbf{W}(k)$, $\mathbf{c}(k)$, and $\mathbf{d}(k)$ is initially drawn randomly, without any *a priori* knowledge (we have in fact witnessed that setting them all to 0 yields equivalent results as well).

In many related publications dealing with KF/EKF/PF-based speech enhancement on both AR- or NN-type models of speech (e.g., [FON'02, GOD'04, HAY'01, SHU'052, VER'02]), preferred nominal values of what we call "fixed parameters" above are often briefly stated with few explanations. However, in our experience, the fixed parameters are of utmost importance and can have a large impact on performance, and cannot be chosen completely at random. Therefore, a discussion on their choice is given below. These fixed parameters are indeed meant to set a physically reasonable model for the evolving quantities at play; for example, it is clear that the performance will depend on the length of the input vector of noisy samples, given a certain sampling frequency. Note here that the distinction between fixed and time-varying parameters is important because it underlines the fundamental difference between a "classical" neural network training, in which usually the actual weights of network are iteratively adjusted using some example data by means of methods such as the backpropagation algorithm, and the PF fixed parameter search, which lies on a different level. In the PF fixed parameter search, specifically $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{c}}$, $\sigma_{\mathbf{d}}$, and σ_{L_x}, the goal is not to

[7] This finding is similar to the observation of the authors of [VER'02] while dealing with PFs for speech enhancement in white noise, as reported in their Section V-A. Note that this is not to be generalized for all contexts: initialization can be very significant in other PF applications such as target tracking.

determine some proposed values for the NN weights that would provide some good enhancement, but only to set *how much they are allowed to vary* from one iteration to the next, regardless of their values and of how they were initialized. This "variation rate" is quantified by the standard deviations (stds) of the random walks on each NN weight. In other words, in essence, we are neither fitting nor training the NN weights, but merely setting the allowed variation rate of these weights. Regarding our remark on the difference from common NN training, we concede that such types of parameters still appear in classical EKF training under the form of an artificial/virtual covariance matrix for the NN weights (sometimes called an artificial process noise), which is typically recommended to be set to a small value. In the context of particle filtering for online learning, however, the consequences of these variation rates are important, since not only are they part of the assumed physical boundaries of the process to be estimated (as opposed to being artificial/virtual) but they also define the "width" of the search net for plausible particles, algorithmically speaking. For example, too small of a value for σ_{L_x} will make the PF struggle to draw good state candidates when changes occur (e.g., at speech onsets). Similarly, a too large value will typically mean that many particles are "wasted," being drawn at each iteration too far from their current mean to relate to the process that they are trying to track, which can be a significant issue when only a few particles are available to begin with.

In summary, in this fixed model parameter search, the goal is not only to determine a range of values that can guarantee a good average performance (as measured by objective measures such as the segmental SNR and the WPESQ), but also a small variability in performance (i.e., a good robustness). Obviously, given the variety of implementations possible (for example, different random number generators, resampling schemes, etc., may be used), the wide range of possible experimental conditions (sampling rates, types of noise, and speakers), and the

large amount of computations necessary per simulation run, the determination of the fixed parameters for the speech model is a tedious task.

For a 20 kHz sampling rate, several thousand simulations were initially run on various speech and noise segments for this purpose. The speech segments and noise material (and levels) used were not restricted to those used in the actual simulations, although the types of noise were still chosen to be stationary. Such a parameter search revealed that the neural-based PF algorithms are more sensitive than the classical RBPF algorithms. Both for completeness and for the purpose of providing a good starting point for the reader wanting to experiment, the fixed parameters used in this thesis are reported in Table 19.

Symbol	Value
$f(.)$	$f(x) = \tanh(x)$
M	13
P	5
σ_w	5×10^{-3}
σ_c	1e-2
σ_d	5×10^{-4}
σ_{L_x}	0.4

Table 19: Default values for the fixed parameters used as part of the Neural-based PF solution.

For all Neural-based PF algorithms in this thesis, an improved resampling scheme is used in order to avoid situations in which the particle cloud entirely collapses to support points with negligible weight, possibly numerically zero, in which case the PF effectively fails. This additional step is embedded in the algorithm, and consists in first working in the logarithm domain, and also guaranteeing that at least a certain amount of particles have nonnegligible weights. It is presented in Appendix A.8.

Finally, the parameter search step led to some observations about the "practical" complexity of Algorithms 6 and 7, as compared with the regular RBPF algorithm. As expected, it was

found that RBPF algorithms require much less particles in order to attain a certain level of enhancement. As an illustration, Figure 11 shows the WPESQ value obtained after the enhancement of a section of speech degraded with white noise versus the number of particles N used, averaged over 20 runs for each value of N.

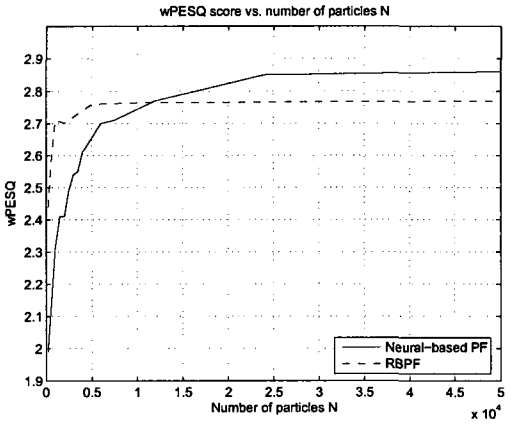

Figure 11: Average WPESQ obtained when a certain amount of particles N is used, for both the neural-based algorithm and the regular RBPF, in white Gaussian noise conditions. In this figure, the neural-based PF uses $M = 10$ and $P = 4$, and the RBPF uses $M = 12$.

On the other hand, the execution time and the amount of memory required for the two cases are not equal for a given amount of particles; in our implementation, it is found for example that a single-neuron network with $M = 12$ runs about six times faster than an RBPF of the same size in white noise, and about eight times faster if a colored noise of order 5 is considered. Instead of providing detailed execution times for all the considered cases, a qualitative summary of the findings related to "practical complexity" follows. Globally, it was noted that for a white-noise environment, the least "practically complex" algorithm was the RBPF, followed by the nondual Algorithms 6 and then the dual one. For our implementations and in the conditions of our experiments, the tendency was however different in the colored noise case, with the neural-based algorithms both achieving their best results with about the same amount of particles as the RBPF.

In this thesis, a fairly large amount of particles (a total of 10000 – 5000 for each sub-PFs for the dual case) are used for the neural-based algorithms, in an effort to ensure that the hierarchy obtained is "asymptotically stable". We find that this amount is roughly where, across all conditions, the enhanced signals begin to obtain their best scores (in the sense that adding more particles only marginally improves them). An additional advantage of using large amounts of particles is the fact that the variance of the resulting scores (SNR or WPESQ) is then very low, in 10^{-4} the range, ensuring that the differences in scores reported are statistically significant. It should be kept in mind, however, that statistical significance does not imply perceivable significance.

6.6.2. Analysis of simulation results

From the complete simulation results tables (20 of them are available in Appendix B.4) the following three summary tables were compiled, in an attempt to isolate the most pertinent findings.

First, Table 20 below shows the average differences in scores between some of the various configurations of neural networks.

Condition	Alg.	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
WGN	$\bar{B}-B$	0.45	0.29	0.04	0.04	0.12	0.10	0.07
	$\bar{D}-D$	1.37	0.78	0.06	0.18	0.28	0.16	0.23
	$M-\bar{M}$	0.26	0.10	0.02	0.03	0.11	0.05	0.08
Colored	$\bar{B}-B$	0.55	0.36	0.03	0.03	0.07	0.05	0.08
	$\bar{D}-D$	0.81	0.45	0.03	0.05	0.09	0.07	0.10
	$M-\bar{M}$	0.60	0.32	0.02	0.03	0.08	0.06	0.08

Table 20: Average score differences between several configurations of feedforward network. The rows named "$\bar{B}-B$" show the average score difference between unbiased and biased networks; similary "$\bar{D}-D$" indicates the average difference between nondual/joint and dual networks; finally, "$M-\bar{M}$" presents the average difference between multiple-neurons networks and single-neurons networks. For example, the top-left entry indicates that on average, using a bias reduces the SNR by 0.45 dBs.

This table is meant to reflect the following few points:

1. There is nothing to be gained from using a biased network , if the biases are to be governed by random walk models.
2. The global performance of the dual neural-based PF algorithm is markedly inferior to that of the nondual one. The average differences are most significant in WGN conditions.
3. It is advantageous to use multiple-neurons, as opposed to only one neuron.
4. The best results are obtained with a nondual multiple-neuron, unbiased network.

To complete the observations reflected by Table 20, the complete results in Appendix B.4 further suggest that the two best results are obtained via the **NPF-**$\bar{D}M\bar{B}$, followed by the **NPF-**$\bar{D}\bar{M}\bar{B}$. We note here that regardless of the configurations considered, virtually all dual networks perform worsely than the joint ones. For example, even an unbiased multiple-neurons dual algorithm yields inferior scores to biased, single-neurons joint algorithms.

Focusing on the **NPF-**$\bar{D}M\bar{B}$ and the **NPF-**$\bar{D}\bar{M}\bar{B}$ networks, Tables 21 and 22 show their average performance compared with that of the "regular" RBPF.

Condition	Algorithm	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
WGN VL/L	**NPF-**$\bar{D}\bar{M}\bar{B}$	8.25	-0.38	0.53	1.12	0.74	1.75	0.85
	NPF-$\bar{D}M\bar{B}$	8.49	-0.31	0.63	1.14	0.87	1.79	0.99
	RBPF	8.17	-0.50	0.55	1.09	0.59	1.69	0.79
Colored VL/L	**NPF-**$\bar{D}\bar{M}\bar{B}$	2.09	-3.84	0.62	1.11	1.34	1.33	1.14
	NPF-$\bar{D}M\bar{B}$	2.64	-3.15	0.64	1.12	1.39	1.39	1.25
	RBPF	1.70	-4.16	0.58	1.09	1.25	1.33	1.09

Table 21: Average scores in VL/L conditions for two NPF configurations and the RBPF.

Condition	Algorithm	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
WGN M/H	**NPF-** $\bar{D}\bar{M}\bar{B}$	15.37	4.53	0.96	1.44	1.59	2.51	1.68
	NPF- $\bar{D}M\bar{B}$	15.82	4.71	0.96	1.51	1.80	2.55	1.83
	RBPF	16.08	4.97	0.96	1.59	1.83	2.61	1.86
Colored M/H	**NPF-** $\bar{D}\bar{M}\bar{B}$	10.22	1.40	0.98	1.62	2.34	2.08	2.02
	NPF- $\bar{D}M\bar{B}$	10.72	1.49	0.98	1.66	2.43	2.14	2.10
	RBPF	11.31	1.72	0.97	1.68	2.47	2.23	2.11

Table 22: Average scores in M/H conditions for two NPF configurations and the RBPF.

From Tables 21 and 22, the following two further points are noted regarding the average comparative performance of each algorithms:

5. Overall, the "regular" RBPF offers a very comparable performance, although the observed trend is that the NPF algorithm offers a better rated enhancement at low SNR, and conversely the RBPF stands out at high SNR.
6. At very low and low SNR, in fact, even the neural-based PF with only one neuron can perform better than the RBPF algorithm.

Let us now comment on the 6 points listed above. First, it may seem surprising that the use of a bias would consistently penalize the algorithm (Observation 1). The bottom line is however that the addition of a bias on neurons increases the level of randomness entering each neuron because of the poor, uninformative evolution model for the biases (a Gaussian random walk).

Observation 2 can be anticipated from the fact that the theoretical properties of the nondual PF do not carry over to the dual algorithm. It also makes sense that, provided the number of particles is large enough, increasing the number of neurons to a reasonable extent should improve the results (Observation 3). In fact, it was also observed that the multiple-neurons algorithm is more robust than its single-neuron counterpart: in all the results observed (including those obtained during the parameter search), the single-neuron case has a tendency

to damage or omit certain small parts of the input speech much more often than the multiple neurons case.

In addition to Observation 5, the following is important to note. While the neural-model-based PF algorithm can be tuned to outperform the linear-model-based RBPF at low SNR, it must be underlined that the RBPF algorithm is easier to deal with, since in our experience, it is less sensitive to the user's initial choice of parameters. In that very sense, its main advantage over the multiple-neuron PF algorithm is its robustness.

In terms of subjectively perceived quality, from informal listening tests it was concluded that all PF algorithms have the same types of strengths and weaknesses. Namely, they yield low residual noise between utterances, but they still contain some within the spoken parts of the enhanced speech. Careful listening indicates, however, that the RBPF algorithm is more effective at reducing noise during pauses, while the neural PFs process actual speech utterances better (in the sense that the speech segments do not sound as distorted). This could explain the better NPF scores at low SNR.

6.7. Conclusions

Several particle filtering algorithms for speech enhancement, based on a nonlinear speech production model, were presented in this Chapter, offering new alternatives to the ones already available in the literature. One important result of this work is that the derived PF algorithms based on neural network speech model represent a framework that can be easily parameterized to operate on white-noise models previously reported in the literature. The series of tests performed has shown several important trends. In summary, first no clear advantage was found in using a dual algorithm in the case of the neural-based PFs. Second,

from the results shown, the use of a bias running on a Gaussian random walk is not recommended either. Third, it is beneficial to employ several neurons, even though it may drastically increase the amount of computations per iteration. The best possible configuration found was thus a nondual, multiple-neuron, and nonbias algorithm. The "classical" RBPF for speech enhancement is, however, very close in terms of performance, and as noted in this thesis, it is also easier to "configure"/set its parameters. It is also able to perform better at high SNRs, but struggles more in low SNR situations. The NPF algorithms are better suited for low and very low SNR situations.

As noted in Section 6.4, the dual algorithm described operates according to a particular form of information exchange. In fact, it is the simplest such form that one might envision: as only the estimated means of each quantity is exchanged between the two PFs. This is possibly one of the reasons why the dual PF does not perform as well as the standard one. We are currently exploring other possibilities in which multiple parameters will be exchanged between the two PFs, such as the variance, the modes, or other descriptive features of each estimated state distribution.

There is also still room for improvement for all the particle filtering algorithms presented above. Specifically, the evolution model (a Gaussian random walk) for the parameters such as $\mathbf{W}(k)$ is uninformative. Future research directions include the testing of more realistic/informed models for these parameters. An existing example of such an effort is reported in [WIN'06], where the parameters of a hypothetical evolution model for the line spectral frequency coefficients of clean speech are learnt prior to enhancement. This type of approach is a motivation to not completely abandon the inclusion of biases altogether and could also in fact prove another future advantage over the regular RBPF algorithms. Indeed, if

a more accurate evolution model can be found for each of the parameters $\mathbf{W}(k)$, $\mathbf{c}(k)$, and $\mathbf{d}(k)$ it can be hypothesized that a better enhancement accuracy can be obtained, due to the greater flexibility over the single AR vector in the RBPF case.

VII. Fullband processing and all-pole modelling of noise power spectral densities

7.1. Introduction

7.1.1. Context and goal

As explained in Section 3.3, this thesis proposes to employ background noise power spectral density (PSD) estimation, rather than the usually less robust voice-activity-detectors (VADs), as a module for the final enhancement algorithm. Thus, from the point of view of the state space-based algorithm, a discrete representation of the background noise PSD is assumed to be available. However, the noise model described in Section 3.1.3 is autoregressive or all-pole; therefore a conversion between the estimated discrete PSD and a set of AR parameters is necessary, and the spectrum induced by the converted AR parameters and the estimated noise spectrum must match as closely as possible, according to certain specified criteria. In addition, for the AR modelling (or all-pole modelling) of a real-world noise, the "conversion" is typically not one-to-one (both the induced and measured spectrum have identical AR representation, even though they are distinct) and some valuable information may be lost, possibly impeding the enhancement process to follow. This is especially the case if small model orders are required. Moreover, large AR model orders are impractical and are potentially computationally expensive.

Spectral matching and analysis is a mature research field, with many distinct autoregressive modeling techniques existing [BUR'87, MAR'87]. In the speech enhancement context described above however, the only data that is assumed to be available is the discrete noise PSD – not the actual time series of the noise. This implies that several well-known techniques

in spectral analysis, such as the popular Burg's method, cannot be readily used. The autocorrelation function is nevertheless available via inverse discrete Fourier transformation of the PSD, and thus speech enhancement algorithms typically use the classical Yule-Walker (YW) technique to obtain noise AR parameters (see for example [DEN'06]). Unfortunately, there is strong evidence in the literature that the YW technique can be inadequate, for example when modeling processes with poles located near the unit circle, or mainly with poorly conditioned covariance matrices [PRI'83, HOO'96]. Accordingly, in other applications such as formant tracking, YW has been shown to be insufficient when dealing with "harmonic" spectra (such as in voiced speech) [MAK'75, JAR'91]. Finally, the above references also agree that the mentioned problems are accentuated when the data size decreases. Thus, other solutions have been developed over the last few decades. Since the conversion of a discrete PSD to a set of AR parameters can be viewed as a minimization problem, these other solutions are essentially based on the introduction of new cost functions.

In this chapter, several cost functions are explored, a new one is introduced, and a new way to minimize them is proposed. The cost functions presented, besides the ratio of spectra (as in the YW method), are the Itakura-Saito distance, the log-spectral RMS, the COSH distance, and a cost function which minimizes the squared error between the measured and induced spectra, thereby naturally forcing a better match in the high-energy lobes (assumed to represent the most perceivable components of noise).

In the literature, the minimization of several of these cost functions can be found in the context of formant tracking (e.g. in [JAR'91, WEI'00, WEI'03]), where the spectra of high-pitched and/or harmonic signals must be analyzed, and the minimization is performed on a set of peaks in the spectra (chosen via peak-picking algorithms). The minimization algorithms

derived in the above sources are essentially all gradient-based, with quasi-Newton forms shown in [JAR'91, WEI'00, WEI'03] in the case of the IS and the COSH distance, and also with simple steepest-descent forms in [WEI'00, WEI'03]. In this chapter, these techniques are transposed to a novel two-step algorithm, operating sequentially on normalized AR coefficients and on the power of the matching error, with multiple advantages arising. Convenient ways to calculate exactly each gradient and Hessian are first derived, in each case with a unifying approach. A variety of appropriate descent techniques are shown, and the relationships with previous works are highlighted.

In the remainder of this chapter, let the letter E represent a generic cost function to be minimized.

7.1.2. Notations, definitions, and a few useful results

The true/measured PSD of the discrete-time signal $x(k)$ is denoted here by $P(f)$, where f is the normalized frequency ranging from 0 to 1 cycle/sample over one frequency period. The set of AR coefficients will be either considered to be "unnormalized" and denoted by $\tilde{\mathbf{a}} = \{\tilde{a}_m\}_{m=0}^{M}$ for an M^{th} order model, or "normalized" and then denoted by $\mathbf{a} = \{a_m\}_{m=1}^{M}$, such that the corresponding residual power is $\sigma_g^{\,2}$. With the two conventions, the induced power spectrum $\hat{P}(f)$ can be written as:

$$\hat{P}(f) = \frac{1}{\left(\sum_{m=0}^{M} \tilde{a}_m \cos(2\pi f m)\right)^2 + \left(\sum_{m=0}^{M} \tilde{a}_m \sin(2\pi f m)\right)^2} = \frac{\sigma_g^{\,2}}{\left(1 - \sum_{m=1}^{M} a_m \cos(2\pi f m)\right)^2 + \left(\sum_{m=1}^{M} a_m \sin(2\pi f m)\right)^2} \tag{41}$$

Note that $\tilde{a}_0 = \sigma_g$ with the unnormalized notation. In the rest of the chapter, we assume that $P(f)$ and its AR-induced counterpart are represented by their discretized version (i.e. DFT), leading to the notation $\{P(n)\}_{n=1}^{N}$ where N is the size of the frame considered (and of the DFT) and the discrete frequency points are at $\{f_n = (n-1)/N\}_{n=1}^{N}$.

Finally, we define the following quantities. For an arbitrary even sequence $P(n)$, define the corresponding sequence:

$$r_P(m) = \frac{2}{N}\sum_{n=1}^{N} P(n)\cos(2\pi f_n m) \tag{42}$$

Relatively to $P(n)$, define the following three matrices $\mathbf{R}_{P,T}$, $\mathbf{R}_{P,H}$, and $\mathbf{L}_{P,H}$, the first being Toeplitz and the last two having Hankel structures:

$$\mathbf{R}_{P,T} = \begin{bmatrix} r_P(0) & r_P(1) & \cdots & r_P(M-1) \\ r_P(-1) & r_P(0) & \cdots & r_P(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_P(-M+1) & r_P(-M+2) & \cdots & r_P(0) \end{bmatrix} \tag{43}$$

$$\mathbf{R}_{P,H} = \begin{bmatrix} r_P(2) & r_P(3) & \cdots & r_P(M+1) \\ r_P(3) & r_P(4) & \cdots & r_P(M+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_P(M+1) & r_P(M+2) & \cdots & r_P(2M) \end{bmatrix} \tag{44}$$

The $M \times M$ matrix $\mathbf{L}_{P,H}$ is then defined by its (m,k) entry as:

$$\mathbf{L}_{P,H}(m,k) = \sum_{j=1}^{M} a_j r_P(m+k-j) \tag{45}$$

Next, let:

$$c_n(\mathbf{a}) = 1 - \sum_{j=1}^{M} a_j \cos(2\pi f_n j) \tag{46}$$

$$s_n(\mathbf{a}) = \sum_{j=1}^{M} a_j \sin(2\pi f_n j) \tag{47}$$

$$\varphi(\mathbf{a},m,n) = s_n(\mathbf{a})\sin(2\pi f_n m) - c_n(\mathbf{a})\cos(2\pi f_n m) \tag{48}$$

Using the above definitions, one can derive the following few results, which will prove useful throughout this Chapter:

$$2\varphi(\mathbf{a},m,n)\cos(2\pi f_n i) = \varphi(\mathbf{a},m-i,n)+\varphi(\mathbf{a},m+i,n)$$

$$2\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n) = \frac{\sigma_g^{\,2}}{\hat{P}(n)}\left(\cos(2\pi f_n(m-k)-\cos(2\pi f_n(m+k)\right)-2c_n(\mathbf{a})\varphi(\mathbf{a},m+k,n)$$

$$\frac{\partial \hat{P}(n)}{\partial a_m} = -\frac{2}{\sigma_g^{\,2}}\hat{P}(n)^2\varphi(\mathbf{a},m,n)$$

$$\frac{\partial}{\partial a_m}\frac{1}{\hat{P}(n)} = \frac{2}{\sigma_g^{\,2}}\varphi(\mathbf{a},m,n)$$

$$\frac{\partial}{\partial a_m}\log\hat{P}(n) = -\frac{2}{\sigma_g^{\,2}}\hat{P}(n)\varphi(\mathbf{a},m,n)$$

$$\frac{2}{N}\sum_n P(n)\varphi(\mathbf{a},m,n) = \sum_{j=1}^{M} a_j r_P(m-j) - r_P(m)$$

Next, the following form of equation also arises often in this Chapter. Assume that $Q(n)$ is real and even (symmetric), with the same length as $P(n)$. Then, if $S(n)=\frac{Q(n)}{\hat{P}(n)}$ and $V(n)=\frac{2}{\sigma_g^{\,2}}c_n(\mathbf{a})Q(n)$, we have:

$$\frac{4}{N\sigma_g^{\,2}}\sum_n Q(n)\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n) = r_S(m-k)-r_S(m+k)+r_V(m+k)-\sum_{j=1}^{M} a_j(m+k-j)$$

In particular,

$$\frac{4}{N\sigma_g^{\,2}}\sum_n \hat{P}(n)^2\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n) \approx r_{\hat{P}}(m-k) \qquad (49)$$

with equality as $N\to\infty$; the reader can verify that the approximation is in fact very good even for relatively small N.

7.1.3. High-level description of the procedure

The problem at hand consists of an unconstrained optimization of various spectral distance functions, represented broadly by the letter E in the following. In other words, the goal consists of finding optimal parameters σ_g and $\{a_m\}_{m=1}^{M}$ for a given cost function E. For this purpose, it is proposed to use a type of descent method [BOY'04] (shown in its general form below), since setting the gradient of E to zero yields a set of equations that cannot be solved analytically (except for the ratio of spectra, as it will be seen).

General descent algorithm; dummy variable x

- Decide on a starting point x
- Then, until a stopping criterion is met, repeat the following:
    1) Determine a descent direction Δx
    2) Line search step: choose a step size α
    3) Update setp: replace x with $x+\alpha\Delta x$

In [JAR'91, WEI'00, WEI'03], the iterative minimization of the cost function E operates directly on $\tilde{\mathbf{a}}$, i.e., the descent algorithms consists of the update of $\tilde{\mathbf{a}}$ at every step. In contrast, our proposed approaches consist of the repeated application of the following two-step rule, to be repeated until convergence:

- Update $\mathbf{a}$ with a single iteration of an appropriate descent algorithm given the current estimate of σ_g
- Obtain the exact corresponding value of σ_g that minimizes E given $\mathbf{a}$

The second step above is possible for each of the cost functions due to the special form of

the all-pole PSD. As a result, the convergence is found to be much faster with the proposed method. In effect, not only is the descent performed in a smaller space, but one of the variables is constantly updated exactly conditioned upon the other ones, lowering each time the value of the cost function. A more formal and detailed minimization procedure will be shown in Section 7.3. In fact, for each cost function, we suggest in Section 7.3 appropriate optimization algorithms. Each of these algorithms require the calculation of several quantities. In [JAR'91, WEI'00, WEI'03], some derivations can be found pertaining to the gradient of a few cost functions operating on the unnormalized AR coefficients. In this chapter, using the normalized coefficients context, we derive convenient ways to compute the gradient vectors and the Hessian matrices, allowing for more flexibility in the choice of algorithm.

7.2. Cost functions, corresponding gradients and Hessians

In this section, the gradients ∇E, the exact corresponding values of σ_g, and the Hessian matrices $\nabla^2 E$ are derived in the context of normalized AR coefficients, since they will be subsequently referred to and required as part of the proposed minimization procedure. For the gradients in the unnormalized context, the reader can refer to the previously cited papers. While not containing any novelties, the Yule-Walker solution is first presented as well, not only for completeness but also to emphasize that it can be treated and viewed from the very same angle as the other cost functions.

7.2.1. Sum of spectral ratios and the Yule-Walker solution

This function is defined as:

$$E_{YW} = \frac{1}{N}\sum_{n}\frac{P(n)}{\hat{P}(n)} \tag{50}$$

It is the only function presented here for which the set of equations obtained from setting the gradient to zero admits a closed-form solution. Indeed, it yields a set of linear equations in terms of the autocorrelation function. For completeness, let us briefly outline a proof that the YW solution minimizes E_{YW}. For m, $1 \le m \le M$, we have:

$$\frac{\partial E_{YW}}{\partial a_m} = \sum_{n=1}^{N} \frac{P(n)}{N\sigma_g{}^2} \frac{\partial}{\partial a_m} \left(c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2 \right) \tag{51}$$

In this case, setting $\frac{\partial E_{YW}}{\partial a_m} = 0$ yields a set of M equations indexed by m, $1 \le m \le M$:

$$\sum_{j=1}^{M} a_j \left(\sum_{n=1}^{N} P(n) \cos(2\pi f_n (m-j)) \right) = \sum_{n=1}^{N} P(n) \cos(2\pi f_n m) \tag{52}$$

In the time domain (via IDFT, or using the previously introduced notation), the above equation becomes:

$$\sum_{j=1}^{M} a_j r_P(m-j) = r_P(m) \tag{53}$$

The M equations can be rewritten as:

$$\mathbf{R}_{P,T}\mathbf{a} = \mathbf{r}_P, \text{ yielding } \mathbf{a} = \mathbf{R}_{P,T}{}^{-1}\mathbf{r}_P \tag{54}$$

Since the matrix $\mathbf{R}_{P,T}$ is Toeplitz, the inversion can be conducted efficiently via the Levinson-Durbin algorithm.

A comprehensive coverage of the YW method and its effects is presented in [MAK'75], where it is stated that the spectral matching of $\hat{P}(f)$ is uniform over the frequency range – in other words, the small values of $P(f)$ are as important in terms of minimization/matching as

the large ones. Such a property can be either an advantage and a disadvantage, depending on the application. If $P(f)$ has some large-valued lobes (or "high energy frequency points" in [MAK'75]) but the majority of its values are small, $\hat{P}(f)$ may not sufficiently fit the large lobes of $P(f)$.

Moreover, it is shown in [PRI'83, HOO'96] that a poorly conditioned covariance matrix will result in a poor match when the Yule-Walker method is employed. [PRI'83, HOO'96] also note increased mismatches as the data size decreases.

7.2.2. Itakura-Saito distance

Another error criterion is given by the discrete version of the Itakura-Saito (IS) distance [AUL'84, ITA'70]:

$$E_{IS} = \frac{1}{N}\sum_{n}\left[\frac{P(n)}{\hat{P}(n)} - \ln\left(\frac{P(n)}{\hat{P}(n)}\right) - 1\right] \tag{55}$$

In the context of speech enhancement the use of the IS distance – often used as a measure of the perceptual difference between two processes represented by their spectra (e.g. HAN'98) – is appealing since the goal is to remove the most perceivable features of the background noise.

We have here the following results:

$$\frac{\partial E_{IS}}{\partial a_m} = \frac{2}{N\sigma_g^{\,2}}\sum_{n}\left(P(n) - \hat{P}(n)\right)\varphi(\mathbf{a}, m, n) \tag{56}$$

And thus:

$$\frac{\partial E_{IS}}{\partial a_m} = \frac{1}{\sigma_g^{\,2}}\left(\sum_{j=1}^{M} a_j\left(r_P(m-j) - r_{\hat{P}}(m-j)\right) - \left(r_P(m) - r_{\hat{P}}(m)\right)\right) \tag{57}$$

Therefore, in vector notation the $M \times 1$ gradient is:

$$\nabla E_{IS} = \frac{1}{\sigma_g^2}\left\{\left(\mathbf{R}_{P,T} - \mathbf{R}_{\hat{P},T}\right)\mathbf{a} - \left(\mathbf{r}_P - \mathbf{r}_{\hat{P}}\right)\right\} \tag{58}$$

Next:

$$\sigma_g^2 \frac{\partial^2 E_{IS}}{\partial a_m \partial a_k} = \frac{2}{N}\sum_n \left(P(n) - \hat{P}(n)\right)\cos\left(2\pi f_n (m-k)\right) - \frac{2}{N}\sum_n \varphi(\mathbf{a},m,n)\frac{\partial}{\partial a_k}\hat{P}(n)$$

$$= r_P(m-k) - r_{\hat{P}}(m-k) + \frac{4}{N\sigma_g^2}\sum_n \hat{P}(n)^2 \varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)$$

With $V(n) = \frac{2}{\sigma_g^2} c_n(\mathbf{a})\hat{P}(n)^2$, this gives:

$$\sigma_g^2 \frac{\partial^2 E_{IS}}{\partial a_m \partial a_k} = r_P(m-k) - r_{\hat{P}}(m+k) + r_V(m+k) - \sum_{j=1}^{M} a_j r_V(m+k-j) \tag{59}$$

which can be simplified in matrix notation as:

$$\nabla^2 E_{IS} = \frac{1}{\sigma_g^2}\left(\mathbf{R}_{P,T} - \mathbf{R}_{P,H} + \mathbf{R}_{V,H} - \mathbf{L}_{V,H}\right) \tag{60}$$

It appears that both the gradient and Hessian can be computed efficiently, many of their components being formed by autocorrelation functions (which can be obtained via inverse Fast Fourier Transforms). In addition, from the result of Equation (49), we can write:

$$\nabla^2 E_{IS} \approx \frac{1}{\sigma_g^2}\mathbf{R}_{P,T} \tag{61}$$

with equality as N grows. Interestingly, a similar approximate result for unnormalized AR coefficients was previously obtained in [JAR'91].

Finally, $\frac{\partial E_{IS}}{\partial \sigma_g^2} = 0$ yields the required corresponding value of σ_g^2, which can be obtained exactly as:

$$\sigma_g^2 = \sum_n P(n)(c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2) \tag{62}$$

7.2.3. RMS log-spectral ratio

The RMS log-spectral Ratio error function [GRA'76] is defined as:

$$E_{\log RMS} = \frac{1}{2N}\sum_n \left[\log\frac{P(n)}{\hat{P}(n)}\right]^2 \tag{63}$$

For this case, let $S(n) = \hat{P}(n)\log\frac{P(n)}{\hat{P}(n)}$. We then have the following:

$$\frac{\partial E_{\log RMS}}{\partial a_m} = \frac{2}{N{\sigma_g}^2}\sum_n S(n)\varphi(\mathbf{a},m,n) = \frac{1}{{\sigma_g}^2}\left(\sum_{j=1}^{M} a_j r_S(m-j) - r_S(m)\right) \tag{64}$$

Therefore:

$$\nabla E_{\log RMS} = \frac{1}{{\sigma_g}^2}\left(\mathbf{R}_{S,T}\mathbf{a} - \mathbf{r}_S\right) \tag{65}$$

Next:

$$\begin{aligned}{\sigma_g}^2\frac{\partial^2 E_{\log RMS}}{\partial a_m \partial a_k} &= \frac{2}{N}\sum_n S(n)\cos\left(2\pi f_n(m-k)\right) + \frac{2}{N}\sum_n \varphi(\mathbf{a},m,n)\frac{\partial}{\partial a_k}S(n)\\ &= r_S(m-k) + \frac{4}{N{\sigma_g}^2}\sum_n \hat{P}(n)^2\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)\\ &\quad - \frac{4}{N{\sigma_g}^2}\sum_n \hat{P}(n)S(n)\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)\end{aligned} \tag{66}$$

Using the same technique as for the E_{IS} cost function, the exact Hessian can therefore be obtained, but a very good approximation (which quickly becomes exact as N grows) is given by the following relation. With $V(n) = \frac{2}{{\sigma_g}^2}c_n(\mathbf{a})\hat{P}(n)S(n)$, we get:

$$\begin{aligned}{\sigma_g}^2\frac{\partial^2 E_{\log RMS}}{\partial a_m \partial a_k} &\approx r_S(m-k) + r_{\hat{P}}(m-k) - \frac{4}{N{\sigma_g}^2}\sum_n \hat{P}(n)S(n)\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)\\ &= r_{\hat{P}}(m-k) + r_S(m+k) - r_V(m+k) + \sum_{j=1}^{M} a_j r_V(m+k-j)\end{aligned} \tag{67}$$

Thus, the Hessian is closely approximated by:

$$\nabla^2 E_{\log RMS} \approx \frac{1}{\sigma_g^{\,2}}\left(\mathbf{R}_{\hat{P},T} + \mathbf{R}_{S,H} - \mathbf{R}_{V,H} + \mathbf{L}_{V,H}\right) \tag{68}$$

Note that if desired, the difference between the two matrices $\mathbf{R}_{S,H} - \mathbf{R}_{V,H}$ can be obtained directly, filling it appropriately with the inverse DFT of the sequence $S(n) - V(n)$.

Then, setting $\frac{\partial E_{\log RMS}}{\partial \sigma_g^{\,2}} = 0$ yields an exact equation for $\sigma_g^{\,2}$:

$$\sigma_g^{\,2} = \exp\left(\sum_n \log\left(P(n)(c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2)\right)\right) \tag{69}$$

7.2.4. COSH distance

The COSH error function – a symmetrized version of the Itakura-Saito distance [GRA'76] – is given by:

$$E_{COSH} = \frac{1}{N}\sum_n \left(\frac{P(n)}{\hat{P}(n)} + \frac{\hat{P}(n)}{P(n)}\right) \tag{70}$$

For the derivations below, we define $S(n) = \frac{\hat{P}(n)^2}{P(n)}$. With this error function the gradient is given by:

$$\frac{\partial E_{COSH}}{\partial a_m} = \frac{2}{N\sigma_g^{\,2}}\sum_n \left(P(n) - S(n)\right)\varphi(\mathbf{a}, m, n) \tag{71}$$

Thus,

$$\sigma_g^{\,2}\frac{\partial E_{COSH}}{\partial a_m} = \sum_{j=1}^{M} a_j\left(r_P(m-j) - r_S(m-j)\right) - \left(r_P(m) - r_S(m)\right) \tag{72}$$

In other words, we can use:

$$\nabla E_{COSH} = \frac{1}{\sigma_g^{\,2}}\left((\mathbf{R}_{P,T} - \mathbf{R}_{S,T})\mathbf{a} - (\mathbf{r}_P - \mathbf{r}_S)\right) \tag{73}$$

Next, we have:

$$\sigma_g^{\,2}\frac{\partial^2 E_{COSH}}{\partial a_m \partial a_k} = \frac{2}{N}\sum_n \big(P(n)-S(n)\big)\cos\big(2\pi f_n(m-k)\big) - \frac{2}{N}\sum_n W(n)\varphi(\mathbf{a},m,n)\frac{1}{P(n)}\frac{\partial}{\partial a_k}\hat{P}(n)^2$$
$$= r_P(m-k) - r_S(m-k) + \frac{8}{N\sigma^2}\sum_n \hat{P}(n)S(n)\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)$$

With $V(n) = \frac{2}{\sigma_g^{\,2}} c_n(\mathbf{a})\hat{P}(n)S(n)$, we get:

$$\sigma_g^{\,2}\frac{\partial^2 E_{COSH}}{\partial a_m \partial a_k} = r_P(m-k) + r_S(m-k) - 2r_S(m+k) + 2r_V(m+k) - 2\sum_{j=1}^{M} a_j r_V(m+k-j) \tag{74}$$

Hence, the Hessian is:

$$\nabla^2 E_{COSH} = \frac{1}{\sigma_g^{\,2}}\big(\mathbf{R}_{P,T} + \mathbf{R}_{S,T} - 2\big(\mathbf{R}_{S,H} - \mathbf{R}_{V,H} + \mathbf{L}_{V,H}\big)\big) \tag{75}$$

With $\frac{\partial E_{COSH}}{\partial \sigma_g^{\,2}} = 0$ the exact corresponding solution required is:

$$\sigma_g^{\,2} = \sqrt{\frac{\sum_n P(n)(c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2)}{\sum_n \big(P(n)(c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2)\big)^{-1}}} \tag{76}$$

7.2.5. MSE cost function

This cost function can be written as:

$$E_{MSE} = \frac{1}{2N}\sum_n \big[P(n) - \hat{P}(n)\big]^2 \tag{77}$$

If the PSD to be matched contains high energy lobes, in a typical YW-induced mismatch the lobes in the induced spectrum do not align well with those of the original PSD. With the E_{MSE} cost function, such a spectral mismatch of a high energy lobe has a large impact, therefore it is

anticipated that the spectral matching will naturally focus on these large lobes, without the need to introduce a potentially nontrivial weighting function $W(n)$. Note also that while in all the other cases, the error function had no unit, this time it has the dimension of the square of a PSD.

For the determination of the gradient and Hessian, let $S_1(n) = \left(P(n) - \hat{P}(n)\right)\hat{P}(n)^2$ and $S_2(n) = \hat{P}(n)^2 P(n) - 3S_1(n)$. Then, the gradient for the E_{MSE} cost function becomes:

$$\frac{\partial E_{MSE}}{\partial a_m} = \frac{2}{N\sigma^2}\sum_n S_1(n)\varphi(\mathbf{a},m,n) = \frac{1}{\sigma^2}\left(\sum_{j=1}^{M} a_j r_{S_1}(m-j) - r_{S_1}(m)\right) \tag{78}$$

Thus, we have:

$$\nabla E_{MSE} = \frac{1}{\sigma^2}\left(\mathbf{R}_{S_1,T}\mathbf{a} - \mathbf{r}_{S_1}\right) \tag{79}$$

Regarding the Hessian, we can write:

$$\begin{aligned}\frac{\partial^2 E_{MSE}}{\partial a_m \partial a_k} &= \frac{2}{N\sigma^2}\sum_n S_1(n)\cos\left(2\pi f_n(m-k)\right) + \frac{2}{N\sigma^2}\sum_n \varphi(\mathbf{a},m,n)\frac{\partial}{\partial a_k}S_1(n) \\ &= \frac{1}{\sigma^2}r_{S_1}(m-k) + \frac{4}{N\sigma^4}\sum_n \hat{P}(n)S_2(n)\varphi(\mathbf{a},m,n)\varphi(\mathbf{a},k,n)\end{aligned} \tag{80}$$

Therefore, with $V(n) = \frac{2}{\sigma^2}c_n(\mathbf{a})S_2(n)\hat{P}(n)$, we get:

$$\sigma^2\frac{\partial E_{MSE}}{\partial a_m \partial a_k} = r_{S_1}(m-k) + r_{S_2}(m-k) - r_{S_2}(m+k) + r_V(m+k) - \sum_{j=1}^{M} a_j r_V(m+k-j) \tag{81}$$

As a consequence, the Hessian matrix can be computed with:

$$\nabla^2 E_{MSE} = \frac{1}{\sigma^2}\left(\mathbf{R}_{S_1,T} + \mathbf{R}_{S_2,T} - \mathbf{R}_{S_2,H} + \mathbf{R}_{V,H} - \mathbf{L}_{V,H}\right) \tag{82}$$

Finally, $\frac{\partial E_{MSE}}{\partial \sigma_g^2} = 0$ yields:

$$\sigma_g^{\,2} = \frac{\sum_n \frac{P(n)}{c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2}}{\sum_n (c_n(\mathbf{a})^2 + s_n(\mathbf{a})^2)^{-2}} \tag{83}$$

7.3. Minimization procedure and convergence properties

As previously stated, one of the contributions of this chapter is to propose a two-step update, alternating between a decent on the normalized AR coefficients $\mathbf{a} = \{a_m\}_{m=1}^{M}$ and the residual variance $\sigma_g^{\,2}$. More specifically, the generic algorithm is given below.

Proposed two-step descent algorithm, cost function E

- Obtain an initial estimate for $\mathbf{a}$ and $\sigma_g^{\,2}$ by minimizing E_{YW} (i.e., solving the traditional YW equations)
- Then, repeat the following until a certain stopping criterion is reached:
  1) Fixing $\sigma_g^{\,2}$, determine a descent direction for $\mathbf{a}$, denoted by $\Delta\mathbf{a}$,
  2) Perform a line search: choose an appropriate step size α
  3) Replace $\mathbf{a}$ with $\mathbf{a} + \alpha\Delta\mathbf{a}$
  4) With the new estimate for $\mathbf{a}$, obtain the value of $\sigma_g^{\,2}$ that minimizes E

Simulation results will confirm that the above two-step algorithm achieves significantly faster convergence than any existing equivalent direct descent on the unnormalized AR coefficients.

7.3.1. Gradient Descent

The most straightforward solution consists of letting:

$$\Delta\mathbf{a} = -\nabla E \tag{84}$$

This can be done for each of the given cost functions, using Equations (58), (65), (73), and (79). In each case, the gradient vector can be obtained efficiently, in part from the possible use of IDFTs. Moreover, the computation of σ_g^2 can be carried out using several of the terms already needed in the update of $\mathbf{a}$.

7.3.2. Newton and Quasi-Newton methods

In the Newton descent method, we let:

$$\Delta\mathbf{a} = -\left(\nabla^2 E\right)^{-1} \nabla E \tag{85}$$

The above can be used for each of the cost functions as well. One of the important facts to consider is the special form of the matrices composing the Hessian, which is itself in each case shown to be Toeplitz-plus-Hankel – therefore, fast inversion algorithms exist (e.g. [GOH'89, HEI'88]).

As a particular case, a quasi-Newton algorithm can be obtained for the E_{IS} method by approximating the Hessian with Equation (61), that is, by zeroing all the Hankel terms in the true Hessian. The approximation is very good as N grows; a similar quasi-Newton descent was developed for the case of a direct unnormalized coefficients descent on E_{IS} in [JAR'91]. In terms of computational complexity, this is very beneficial since the approximated Hessian is constant and independent of $\mathbf{a}$, therefore, it can be inverted prior to the algorithm execution.

In [WEI'00, WEI'03], another quasi-Newton algorithm is proposed for the optimization of the E_{COSH} function, again operating with unnormalized coefficients; upon examination, we find that the approximate Hessian is however not as close to the true one as for the E_{IS} case, and

the two are not asymptotically equal. Transposing the rationale given in [WEI'00, WEI'03] (itself inspired by the work of [JAR'91]) to our two-step method, the resulting approximate Hessian becomes:

$$\hat{\nabla}^2 E_{COSH} = \frac{1}{\sigma_g^{\,2}} \mathbf{R}_{P,T} \tag{86}$$

As for the quasi-Newton descent on E_{IS}, it is also advantageous in terms of complexity, however the above proposed matrix is not asymptotically equal to the true Hessian.

7.3.3. Gauss-Newton and Levenberg-Marquardt methods

In the case of the minimization of $E_{\log RMS}$ and E_{MSE}, the problem can be directly formulated as a nonlinear Least-Squares optimization. Therefore, the Gauss-Newton and the Levenberg-Marquardt methods can be employed. They both require the computation of the $N \times M$ Jacobian matrix $\mathbf{J}$, whose (n,m) entry is defined as:

$$\mathbf{J}_{\log RMS}(n,m) = \frac{\partial \log \hat{P}(n)}{\partial a_m} = -\frac{2}{\sigma_g^{\,2}} \hat{P}(n)\varphi(\mathbf{a},m,n) \tag{87}$$

$$\mathbf{J}_{MSE}(n,m) = \frac{\partial \hat{P}(n)}{\partial a_m} = -\frac{2}{\sigma_g^{\,2}} \hat{P}^2(n)\varphi(\mathbf{a},m,n) \tag{88}$$

Then, with $\mathbf{P}$ corresponding to the length-N vectorized power spectrum $P(n)$, the Gauss-Newton step is here given by:

$$\Delta\mathbf{a} = -\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\left(\mathbf{P}-\hat{\mathbf{P}}\right) \tag{89}$$

For the Levenberg-Marquardt method, the step-dependent damping parameter λ is also included to yield:

$$\Delta\mathbf{a} = -\left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}_M\right)^{-1}\mathbf{J}^T\left(\mathbf{P}-\hat{\mathbf{P}}\right) \tag{90}$$

where $\mathbf{I}_M$ is the size-M identity matrix.

7.3.4. Line search

In the references [JAR'91, WEI'00, WEI'03], fixed-step descents are suggested (i.e., the parameter α is fixed). A more effective method consists of performing a backtracking line search [BOY'04] to obtain a step-adapted value for α; however, the method requires several evaluations of the cost function E and it is therefore more expensive computationally. In the context of this chapter, with $E(\mathbf{a})$ denoting the cost function evaluated at $\mathbf{a}$ and at the most recent value of σ_g^2, the method is shown below for completeness.

Backtracking line search, cost function E

- Choose positive numbers $\mu < 0.5$, $\beta < 1$.
- Set $\alpha = 1$.
- Then, while $E(\mathbf{a} + \alpha\Delta\mathbf{a}) > E(\mathbf{a}) + \mu\alpha(\nabla E(\mathbf{a}))^T \Delta\mathbf{a}$, replace α with $\beta\alpha$
- Return α

The reader is referred to [BOY'04] for a detailed analysis of this method, as well as the effect of each parameter.

7.3.5. Convergence results

A few examples of convergence curves are now shown. In the following examples, PSDs chosen at random are measured from various noise audio sources (cafeteria, car, and street noises), sampled at 20 kHz and estimated using Welch's averaged modified periodogram. An example of a frame is given in Figure 12. The minimizations are performed on a set of spectral points selected by "peak-picking" (i.e., on the peaks of the measured PSDs) as in

[JAR'91, WEI'00, WEI'03]. The choice of an appropriate peak-picking algorithm is simplified by the fact that we are working on smoothed PSDs (specifically, Welch estimates), and therefore a direct slope examination can yield accurate peak positions. The order chosen for the all-pole modelling of these PSDs is 10.

7.3.5.1. Gradient descent methods

In this subsection, we propose to compare the two-step minimization proposed in this Chapter with the existing gradient descent operating on unnormalized AR coefficients shown in [WEI'00, WEI'03]. For a more fair comparison with these methods, we use a fixed step size, although for each algorithm the step size is individually picked to bring out the best/fastest convergence behavior.

Three examples of typical results are shown in Figures 13, 14, and 15 in the context of the minimization of the E_{IS}, the $E_{\log RMS}$, and the E_{COSH} error functions. Observing these figures, the convergence is clearly much faster in the proposed two-step algorithm. This confirms the advantages of updating separately σ_g^2 and the normalized AR coefficients.

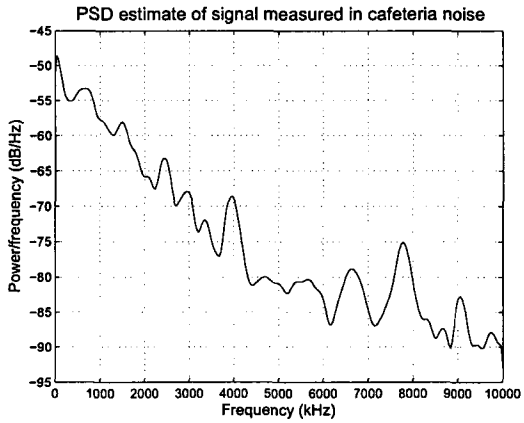

Figure 12: Cafeteria PSD estimated via Welch's averaged modified periodogram over a 512 points frame.

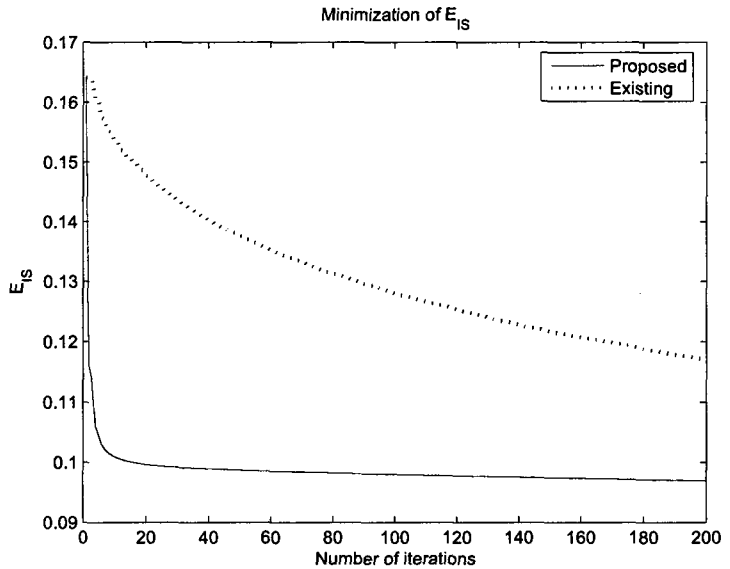

Figure 13: **Steepest descent** minimization of the **Itakura-Saito** distance: Convergence curves for the proposed two-step method vs. the existing method. The curves eventually converge to the same value.

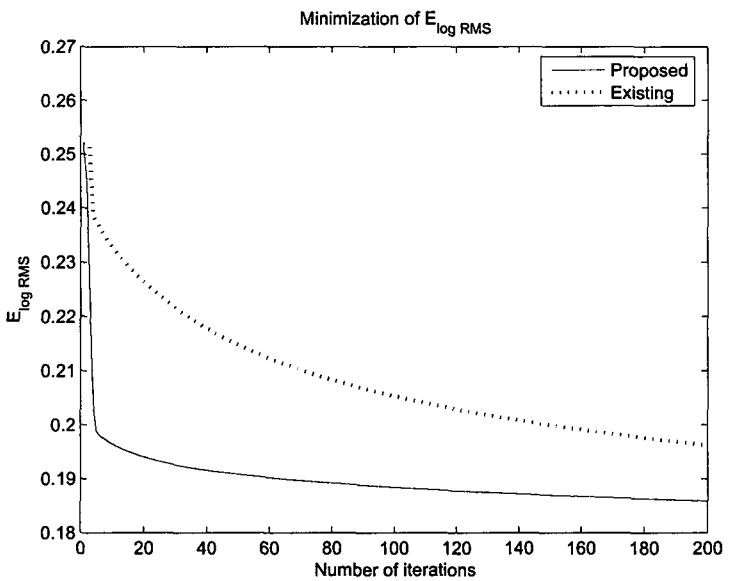

Figure 14: **Steepest descent** minimization of the **log-spectral RMS** distance: Convergence curves for the proposed two-step method vs. the existing method. The curves eventually converge to the same value.

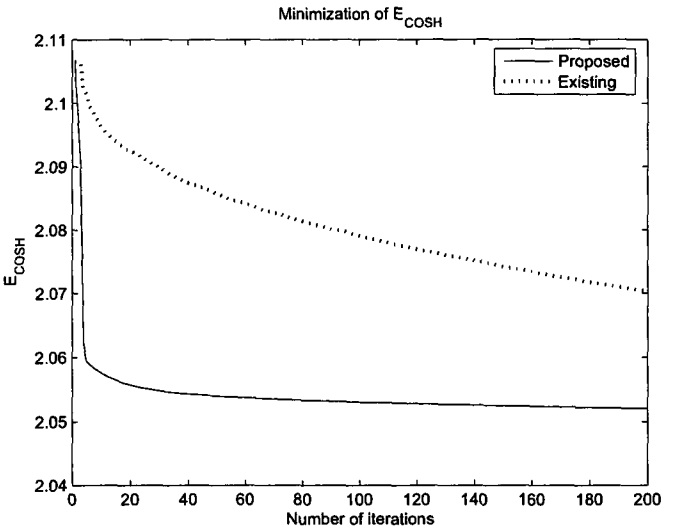

Figure 15: **Steepest descent** minimization of the **COSH** distance: Convergence curves for the proposed two-step method vs. the existing method. The curves eventually converge to the same value.

7.3.5.2. Newton method

Next, the minimization of E_{MSE} is conducted both with a gradient descent and with Newton's method. Both algorithms are here operating with normalized AR coefficients, and typical results are shown in Figure 16.

As expected, it is more advantageous to use the Newton method here, as it reaches a lower value faster than its steepest descent counterpart. The tradeoff is of course the complexity involved in obtaining and inverting the Hessian matrix..

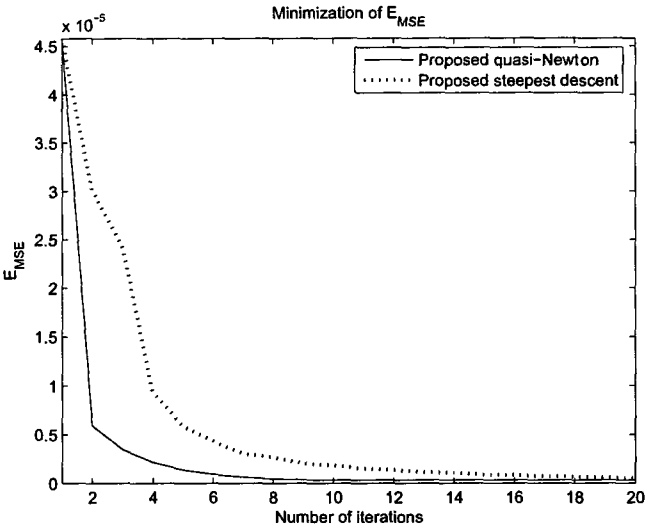

Figure 16: Minimization of the MSE : comparison between steepest descent and quasi-Newton versions.

7.4. Simulation results in speech enhancement

In this section, the above results are tested using three of the fullband algorithms presented in Section 3.2.4: the **DKF**, the $\mathbf{KEM_{BURG}}$, and the **DUKF(1)** algorithm. The three algorithms are tested on the speech and colored noise material given in Chapter II. For complete details regarding experimental conditions, objective measures and algorithms ranking procedures, please refer to Chapter II.

Before conducting actual speech enhancement experiments, several spectral matches were obtained using various noise segments and the different cost functions at hand, and the resulting induced PSDs were plotted against the regular, YW-induced PSD, in order to get a qualitative view of the effect of the cost functions at hand. As Figure 17 illustrates, it was noted that the largest visual differences occur when when larger spectral lobes are present in the PSD to be matched. In many cases though, the differences visually observed were relatively minimal.

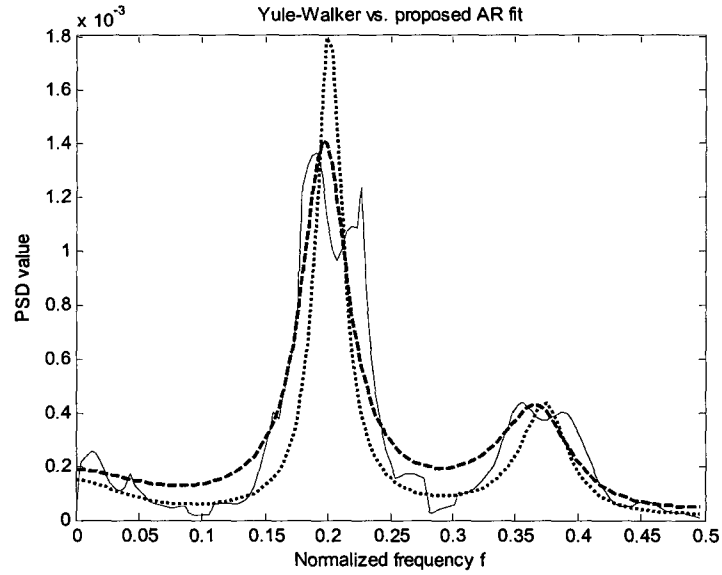

Figure 17: Illustration of the effect of the minimization of the proposed MSE cost function, versus the classical Yule-Walker fit. The solid line represents the PSD to be matched, the dotted line is the regular YW fit and the dashed line is the MSE-fit, both using an order 6 fit. The "test noise PSD" shown here has been artificially generated so as to include two large spectral lobes, and selected so as to showcase a clear difference between the spectral matches.

Next, in the speech enhancement context, no peak-picking algorithm is used. While it makes sense to use peak-picking for formant tracking applications, in the context of speech enhancement valleys are important too: For example, if only peaks are selected there is a risk of overestimation of the noise power in the vicinity of the spectral valleys, which can potentially result in unwanted speech distortion. In other words, in a way a "peak *and* valley" (or extrema) picking would appear more appropriate in this context. From experimentation with such extrema picking in speech enhancement, we find that while the execution time is significantly faster (since only a fraction of the PSD points are used), the changes in output quality can be both positive and negative – for example, larger mismatches in monotonic areas when only few extrema are present can lead to degraded quality. Therefore, we do not recommend the use of extrema picking in speech enhancement.

The complete simulation results are given in Appendix B.5 – some tables of summary now follow. We first show what improvements can be expected from using the proposed noise all-pole PSD modelling techniques in Section 7.4.1; Next, Section 7.4.2 uses the tables of Section B.5 to compare the several algorithms individually.

7.4.1. Average score differences obtained from using other cost functions

The goals of these simulations is to assess the benefits of using different cost functions for modelling the noise PSDs in fullband enhancement setups, as opposed to simply minimizing the Yule-Walker cost function E_{YW} in Equation (50). Therefore, in the first set of Tables 23 and 24 below, the average difference in scores between the enhanced speech obtained via minimization of E_{IS}, $E_{\log RMS}$, E_{COSH}, E_{MSE} (respectively denoted by IS, RMS, COSH, and MSE in the tables) and E_{YW} are reported for VL/L noise conditions on the one hand, and M/H conditions on the other hand.

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
IS	0.24	0.18	0.03	0.01	0.01	0.09	0.08
RMS	0.21	0.22	0.03	0.01	0.02	0.10	0.08
COSH	0.27	0.21	0.04	0.01	0.02	0.11	0.08
MSE	0.37	0.31	0.05	0.01	0.02	0.12	0.10

Table 23: Average differences in scores between the IS, logRMS, COSH, and MSE cost functions minimizations and the classic Yule-Walker minimization in the context of speech enhancement. The average is performed over the 3 tested algorithms and for all VL/L colored noise conditions. For example, in the top-left entry one can see that by minimizing the Itakura-Saito distance, on average the enhanced signals' SNR are higher by 0.24 dBs than that of the YW-based enhanced signal.

M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
IS	0.10	0.13	0.00	0.01	0.02	0.09	0.09
RMS	0.14	0.18	0.01	0.02	0.03	0.11	0.10
COSH	0.21	0.17	0.01	0.02	0.05	0.13	0.12
MSE	0.24	0.20	0.01	0.03	0.05	0.11	0.12

Table 24: Average differences in scores between the IS, logRMS, COSH, and MSE cost functions minimizations and the classic Yule-Walker minimization in the context of speech enhancement. The average is performed over the 3 tested algorithms and for all M/H colored noise conditions. For example, in the top-left entry one can see that by minimizing the Itakura-Saito distance, on average the enhanced signals' SNR are higher by 0.1 dBs than that of the YW-based enhanced signal.

The above tables indicate that there are consistent benefits on average, across all objective measures; in addition, the MSE cost function provides the best average improvements. However, looking at the above numbers only, the benefits appear to be quite marginal – which could indicate that the enhanced signals are almost indistinguishable from those obtained by using the classical YW minimization.

There are however more differences that are not reflected by the tables above, as seen when consulting the full tables in Appendix B.5:

- The improvement depends on the noise type. For example, the averaged overall SNR improvement across the algorithms tested in the L/M-MIL situations is +0.46 dB, and for the L/M-FAC situations it is +0.21 dB.
- The improvement also depends on the algorithm chosen. For example, again in the averaged L/M-MIL situations, the KEM_{Burg} scores obtained by miminizing the MSE cost function are improved by +0.56 dB from using the YW method, but the DKF scores by only +0.01 dB (that is, no improvement at all in realistic terms).

From informal listening tests, we find that the above remark on the indistinguishability of the results is partially true: for the most part, the signals obtained using the proposed cost functions are extremely similar to the original, YW-based ones, and their features are very comparable. But we also find that:

- There is overall better noise reduction *between* speech utterances, most noticeably for the MSE-based noise PSD matching.
- We also note that the differences heard depend on the type of noise. Among the four types of noise tested, the most noticeable differences can be heard in the military vehicle noise environment, with improvements "uniformly distributed" over time in the enhanced signal. In the cafeteria noise conditions, the use of the proposed cost function result in "bursts" of improvements scattered in time (especially when considering the MSE cost function) – for example, in some isolated segments the SNR improvement can reach +1.5dB. The analysis of such segments revealed that several of them correspond to sudden high-frequency noise events in the background cafeteria noise – however many of these segments do not appear to share any common traits.
- Moreover, the differences heard also depend on the type of speech enhancement algorithm. The KEM_{Burg} algorithm benefits the most from the proposed all-pole modelling techniques. On the other hand, our subjective impression is that the DKF is on average the least noticeably affected by the introduction of other PSD matching techniques, but this finding is not systematically verified over all SNR and noise conditions.

The above is difficult to explain in general, and we concede that at this point we are still unable to predict the expected type and amount of improvement that will be observed from using a given type of algorithm in a given type of noise – in the following conclusion paragraph some research ideas to study the above problem are proposed. Nevertheless, we rather consistently observe that when large spectral lobes of noise occur at certain

time/frequency indices (e.g. several of these events can be heard in the cafeteria noisy speech signal), a difference can be both measured and heard.

In conclusion, while in terms of convergence rate the methods devised are unequivocally superior, in the context of speech enhancement the benefits from using the proposed cost functions are overall not very significant. Nevertheless, they are at least consistent and therefore only recommended for applications capable of handling the extra computational load. In future works, in order to better study the effect of the changes in the choice of cost functions, it is proposed to:

- Try and isolate some particular forms of noise where the above methods would be more beneficial. For example, some tests with synthetic, large-lobes autoregressive noise could be performed,
- Perform more thorough analysis of time/spectral features of the noise types (or even only noise events) that are better handled by the proposed cost functions, and compare these features with those of the noise types that are not found to be handled any better by them,
- Conduct some experiments with different speech and noise model orders,
- Perform some tests with weighting functions $W(n)$; for example, it would be of interest to incorporate A-weighting in the optimization process (see for example [AAR'92]), in order to force better spectral matches in regions where human listeners will be more sensitive.

7.4.2. Comparative results between several algorithms for the proposed PSD modelling techniques

The tables of simulation results given in Section B.5 also allow us to compare the DKF, the KEM_{Burg}, and the DUKF(1) algorithms together. Recall from Section 5.4.2 (where simulation results pertaining to colored noise handling were discussed) that so far, the RBPF is the best fullband method, followed by the KEM_{Burg} and then the DKF.

In the following tables of results, we are interested in determining:

- where the DUKF(1) ranks when compared with the DKF and KEM_{Burg} in colored noise conditions and when YW-based noise PSD modelling is used.
- whether the above-determined rank still stands when using an MSE-based noise PSD modelling.

To answer these questions, Tables 25 and 26 are given below, where the rankings are determined following the procedure described in Chapter II.

VL/L, YW	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**DUKF(1)**	**DUKF(1)**	**DUKF(1)**	**DUKF(1)**	KEM_{Burg}	**DUKF(1)**	**DUKF(1)**
	DKF	DKF	KEM_{Burg}	KEM_{Burg}	**DUKF(1)**	KEM_{Burg}	KEM_{Burg}
Worst	KEM_{Burg}	KEM_{Burg}	DKF	DKF	DKF	DKF	DKF
VL/L, MSE	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**DUKF(1)**	**DUKF(1)**	KEM_{Burg}	**DUKF(1)**	KEM_{Burg}	**DUKF(1)**	**DUKF(1)**
	DKF	DKF	**DUKF(1)**	KEM_{Burg}	**DUKF(1)**	KEM_{Burg}	KEM_{Burg}
Worst	KEM_{Burg}	KEM_{Burg}	DKF	DKF	DKF	DKF	DKF

Table 25: Algorithms ranking in VL/L noise conditions between the DKF, KEM_{Burg}, and DUKF(1), for both YW-based and MSE-based noise PSD modelling cost functions. The best algorithm is the DUKF(1), in bold characters above.

M/H, YW	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Best	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**	DUKF(1)	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**
	DKF	DKF	DKF	**KEM_{Burg}**	DUKF(1)	DUKF(1)	DUKF(1)
Worst	DUKF(1)	DUKF(1)	DUKF(1)	DKF	DKF	DKF	DKF
M/H, MSE							
Best	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**	**KEM_{Burg}**
	DKF	DKF	DUKF(1)	DUKF(1)	DUKF(1)	DUKF(1)	DUKF(1)
Worst	DUKF(1)	DUKF(1)	DKF	DKF	DKF	DKF	DKF

Table 26: Algorithms ranking in M/H noise conditions between the DKF, KEM_{Burg}, and DUKF(1), for both YW-based and MSE-based noise PSD modelling cost functions. The best algorithm is the KEM_{Burg}, in bold characters above.

From Tables 25 and 26, the DUKF(1) stands out at VL/L SNR conditions – above the KEM_{Burg}. We verify however that it is still below the RBPF both in terms of objective measures, and in terms of our subjective impression. In contrast, at M/H conditions, the DUKF(1) algorithm becomes less capable than the KEM_{Burg}, and is then roughly equivalent in performance to the DKF. We also find that the use of the MSE cost function for noise PSD matching does not change the algorithm rankings.

7.5. Conclusions

This chapter introduced a two-step iterative procedure to be used as part of the minimization of various cost functions, in order to obtain autoregressive (all-pole) models for real PSDs. The main idea consists of separately managing a set of normalized AR coefficients and the model's corresponding residual power, rather than performing the required optimization on a set of unnormalized coefficients as previously done in the literature. This idea is shown to considerably improve convergence behavior without additional computational overhead. Methods for the calculation of the gradient and Hessians for the Itakura-Saito, Log-RMS, COSH, and MSE cost functions were all presented with a common approach, and several appropriate optimization algorithms were proposed and discussed.

Such convergence rate improvement makes the ideas in this chapter very appealing for other applications than speech enhancement, for example formant tracking applications as in [JAR'91, WEI'00, WEI'03].

In the context of fullband speech enhancement however, it is found that the use of different cost functions is only marginally beneficial, in that the objective quality measures are all

slightly improved on average, thereby offering only a moderate attractiveness for applications able to handle the extra computational cost.

It is conjectured that the improvements observed mostly correspond to larger spectral lobes in the noise PSD estimates, since they are the ones that are in general better matched by the proposed cost functions. Some additional research ideas are proposed so as to determine in which situation the proposed spectral matching methods would be most useful.

VIII. Subband approaches: a compromise between time and frequency domain processing

Rather than enhance the speech directly and across the entire spectrum (i.e., in the fullband domain) as done in Chapter VI, it is proposed here to approach the noise removal problem by first decomposing the incoming signal into several frequency bands. The obtained subband signals are then enhanced and finally the output signal is reconstituted. From the contents of this Chapter, at the submission of this thesis three conference papers have been published (see [MUS'10, MUS'10-1, MUS'10-2].

This chapter is organized as follows: first, in Section 8.1, a case for the use of subband decomposition is made, the generic subband configuration used throughout the chapter is described, and some notation is introduced. Then, in Section 8.2, different methods are given to take advantage of externally provided estimates for the noise PSD, as it was the case in previous chapters. Next, in Section 8.3 a particle filtering-based method is presented which can avoid altogether the use of external noise estimation. In Section 8.4, a few techniques taking advantage of subband processing are shown to further reduce the noise in the output signal, and finally Section 8.5 discusses some ways to make the overall subband enhancement scheme even less expensive while improving its output quality by incorporating a Bandwidth Extension step.

In Section 8.6, various simulation results covering the algorithms presented in this Chapter are analyzed and summarized, and global conclusions are given in Section 8.7.

8.1. Subband enhancement rationale, filterbank and notation used

The anticipated advantages from processing the noisy speech signal in the subband domain are multiple: first, a faster execution is expected, since in filterbank configurations each enhancement algorithm can run on decimated signals, and assume much smaller signal and noise model orders in each band. Such advantages have been noted in the literature, for example in [DEN'06, CHA'07]. In [CHA'07], small Kalman Filters in each band operate on speech and noise autoregressions of order 1 with promising results – however the architecture proposed is restricted to stationary colored noise. In [DEN'06], the same formula is applied (two order-1 autoregressions), although the results and tests are also restricted to stationary noise (the noise estimation proposed is VAD-based). Moreover, the tests shown are limited by the choice of 1 second-long speech material.

Nevertheless, inspired by these promising results, this chapter proposes to go further in the same direction so as to obtain a viable algorithm able to deal with wideband signals in real-world noise. Some of the solutions proposed are based on an even less complex subband model than [DEN'06, CHA'07] while achieving higher robustness in the following sense. In order to efficiently handle the crucial step of noise estimation, it is proposed here to let each subband state-space based algorithm assume that the noise can be viewed as an AWGN sequence with time-varying gain (effectively an order-0 autoregression). This not only enables the use of more elementary algorithm in each band, it also naturally allows Particle Filters to draw noise level (gain) candidates internally at each iteration, thereby introducing a novel noise estimation technique. In parallel, we still advocate the use of of dedicated external noise PSD estimation by simply "discretizing" the estimated fullband noise spectrum to a single point in each band for compatibility with our method. Next, such noise handling allows

for a simple bandwise analysis of the enhancement requirements and introduce psychoacoustic criteria and constraints.

In spite of all the anticipated benefits of subband treatments, there are potential downsides to keep in sight as well. First of all, depending on the order of the filters used and the amount of bands considered, some audible reconstruction artefacts can appear. This is due to the fact that any practical subband decomposition is not ideal, in the sense that there are frequency overlaps between bands (each filter's does not have a perfectly square magnitude response). In perfect reconstruction filterbanks for example, this means that a form of aliasing occurs at the synthesis stage to compensate for these overlaps; as a result, processing bands individually can sabotage the perfect reconstruction property and create some audible distortion. Again, the amount of artefacts introduced typically depends on the amount of bands employed, with more distortion likely to appear when more bands are used. Therefore, a compromise must be found. This is especially important to respect the objective of this thesis: Recall that in Chapter I, one of the reasons given for choosing state space-based algorithms is the reported naturalness of their enhanced products, due to the fact that they operate *in the time domain,* on a sample-by-sample basis. As the number of bands increase, our algorithms of choice expose themselves to potential artefacts and unnatural effects. The above qualitative discussion is in fact clearly audible through experimentation: for example, using some of the methods described below, it was found that using more than 32 bands may improve several objective measures, however more artefacts appear and the enhanced speech becomes clearly less natural. As a compromise, we found that a "good" naturalness is kept when a maximum of 32 bands is used. In other words, we find that convincing results are obtained when a good "time-frequency compromise" is reached.

Regarding the notation used, in this Chapter the global enhancement system is assumed to follow the procedure depicted in Figure 18. The incoming noisy signal $z(k)$ is first decomposed via a filterbank with M filters and then decimated by a factor M; the obtained signal in the m^{th} band is denoted by $z_m(l)$. The enhancement process then takes place M times on each of the subband noisy signals $\{z_m(l)\}_{m=1}^{M}$ to obtain the enhanced subband signals $\{y_m(l)\}_{m=1}^{M}$, which are upsampled and recombined to form the output fullband denoised signal $y(k)$.

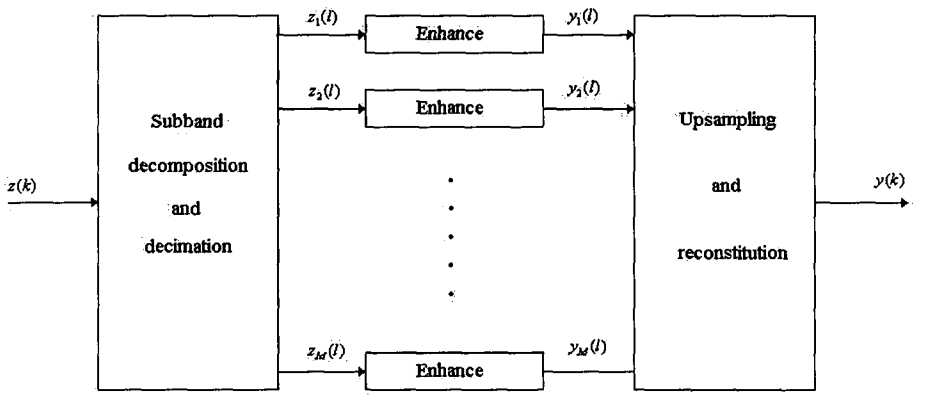

Figure 18: Generic subband enhancement system

To perform the subband decomposition, we consider maximally decimated modulated filter banks, initially with three options: a perfect reconstruction bank with modulation based on the discrete cosine transform (simply named "DCT" filterbank hereafter), a near-perfect reconstruction pseudo-QMF filterbank ("PQMF" below) as described in [NGU'94], and finally a "modified DCT" filterbank, obtained by cosine modulation of a Malvar sinusoidal window prototype ("MDCT" below) – see [SPA'07]. Each solution has its advantages: the DCT bank provides perfect reconstruction and linear phase filters, but is limited in terms of frequency selectivity since the filter orders must equal the number of bands. The PQMF

solution provides good selectivity (the filters' order can be arbitrarily increased) but the reconstruction is not perfect (although it is so in practical terms) and the filters are not linear-phase. Finally, the MDCT filterbank yields perfect reconstruction, a better selectivity than the DCT bank but is not linear-phase.

Our experience with using each of the above three solutions in the context of speech enhancement is that while there is no significant difference in output quality, the PQMF and the MDCT do yield moderately better results than the plain DCT bank of filters. Overall, we find that the PQMF itself yields slightly higher quality than the MDCT; therefore, **this chapter assumes that the subband decomposition is performed by means of PQMF filterbanks**[8].

8.2. Interpreting external noise PSD estimation in the subband context

If the model orders of the *subband* speech and the *subband* noise signals are M_s and M_n, the first question arising is the following: What values should be picked for M_s and M_n? Accordingly, how many bands should be used (i.e, what is the value of M?). In [CHA'07] previously cited in the introduction of this chapter, this problem is investigated in the context of narrowband speech (8 kHz sampling rate), and the authors reach the conclusion that $M = 16$, and M_s and M_n both set to 1 yield the best results in the stationary colored noise context of the thesis.

For broadband signals (more than 16 kHz sampling frequency), two configurations were found to yield interesting results:

[8] In future works, it is proposed to consider setups with critical band decompositions.

- Decomposition into few bands (we use $M = 4$ below) with speech and noise subband models of moderate order (we use M_s and M_n both equal to 4)
- Decomposition into "many" bands (we use $M = 32$ below) with AR models for the subband speech of order 1, and with 0-order for the noise (i.e., M_s and M_n set to (1,0))

As it will be seen, the second configuration enables the use of extremely simple, elementary subband algorithms. If we are to still employ noise PSD estimation algorithms, the two configurations above also imply different ways to incorporate them – the two following sections respectively address the two above cases.

8.2.1. Decomposition of the fullband noise into small-order AR noises

In this case, in each band the order of the noise model is $M_n > 1$, and we have at hand an estimate for the fullband background noise power spectrum. In order to enhance the subband signal $z_m(l)$, one could therefore simply transpose the results of Chapter VI by following the steps below:

- first truncate/window the noise PSD in the m^{th} band
- shift and stretch the truncated PSD such that it spans the entire frequency range (i.e. "downsample" in the frequency domain)

These steps are illustrated in Figures 19 to 22 below.

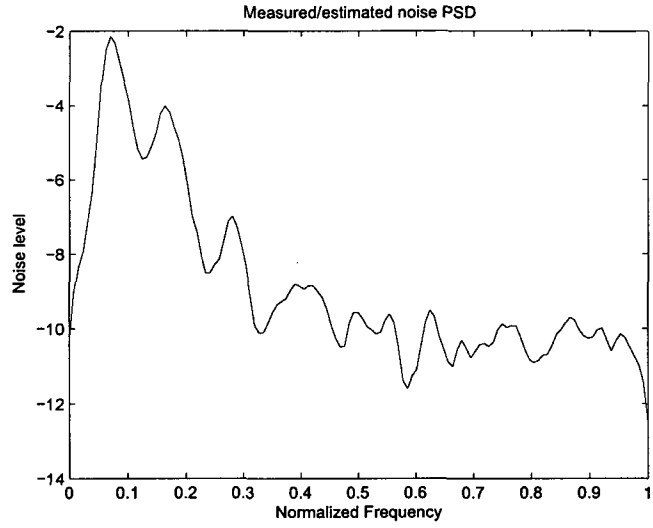

Figure 19: From Figures 19 to 22, an example is shown for the process of obtaining an autoregressive model for the noise in a given subband from the fullband noise PSD. In Figure 19, the estimated noise PSD returned by the external noise estimator is shown. In Figure 20, the portion of the PSD to be considered in the subband processing is highlighted (here, the system assumes 4 bands). In Figure 21, the result of the "decimation" is shown – the PSD has been appropriately truncated and stretched to span the entire frequency range. Finally, in Figure 22, the induced PSD obtained from applying simple Yule-Walker modeling is shown.

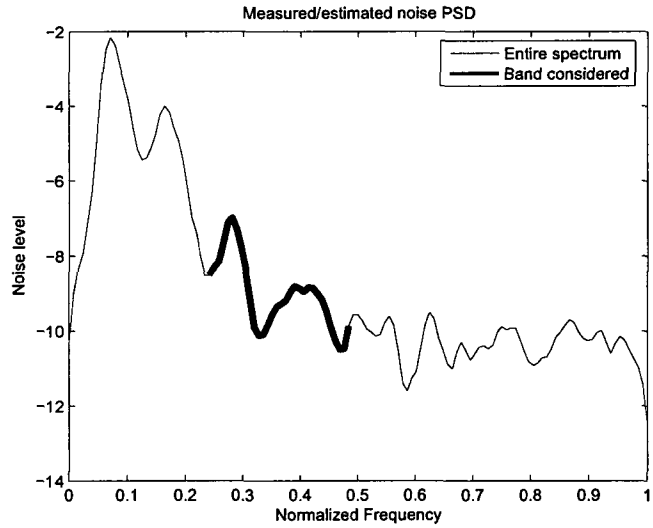

Figure 20: Portion of the PSD to be considered in the subband processing is highlighted. Please see the caption for Figure 19 for the full description of the context of this Figure.

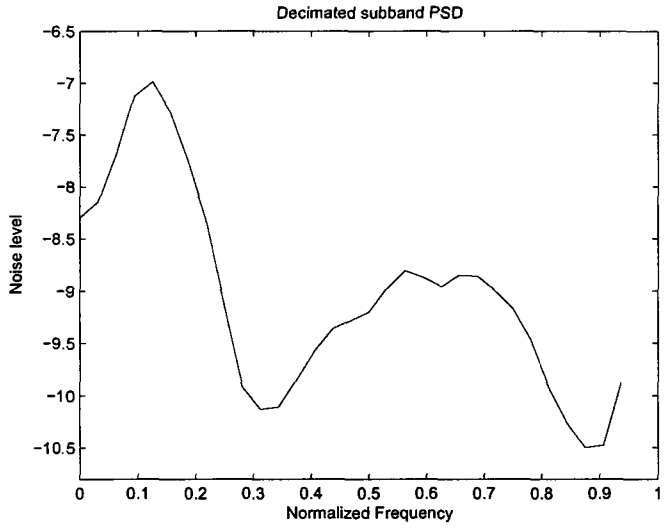

Figure 21: Truncated and stretched selected PSD. Please see the caption for Figure 19 for the full description of the context of this Figure.

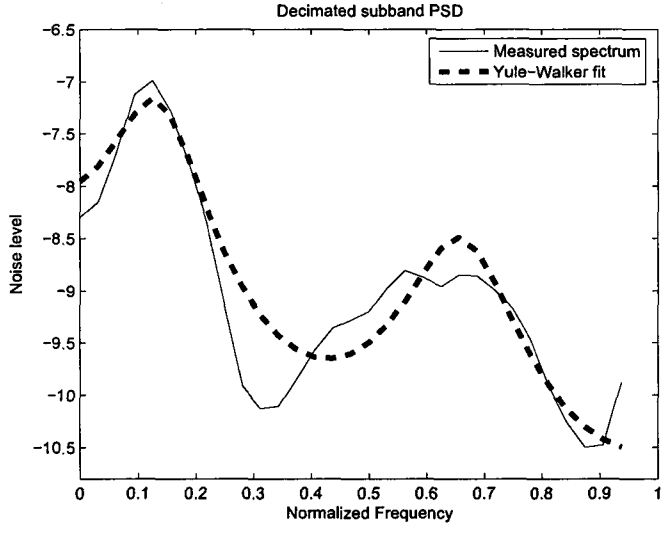

Figure 22: Subband induced PSD obtained. Please see the caption for Figure 19 for the full description of the context of this Figure.

It turns out that the above procedure allows for a closer match of the fullband PSD than what would be possible with an equivalent-order fullband modelling. In other words, using an order-4 AR model in each band can more accurately capture the PSD of a signal than a single order-16 AR model in the fullband. This is illustrated in Figure 23.

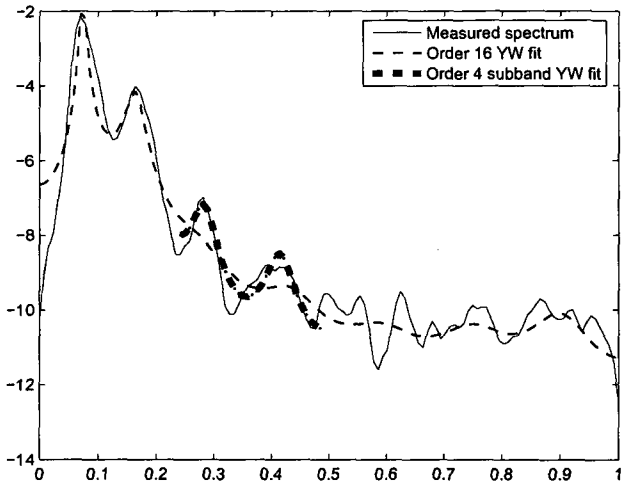

Figure 23: Comparison between an order-4 AR model in the second band and an order-16 AR model in the fullband. Note how much closer the subband fit is to the measured PSD than the fullband fit.

From this point, any of the algorithms previously presented can be applied to the subband signal; simulation results will be shown in Section 8.6.1.

8.2.2. Complete discretization of the noise PSD

In this case, the measured PSD is decomposed into a relatively larger number of bands, while the AR model orders of the subband speech is reduced to 1, while the noise is viewed as an AWGN sequence with time-varying gain (i.e., M_s and M_n set to 1 and 0). Practically speaking, first it implies the use of one-dimensional Kalman Filters (or EKF/UKF/RBPFs), which are considerably simplified building blocks. Next, this understates a full "discretization" of the incoming noise PSD: the average power (i.e., a single number) within a band for a given frame is returned to the state-space algorithm. This "discretization" is illustrated in Figure 24 below.

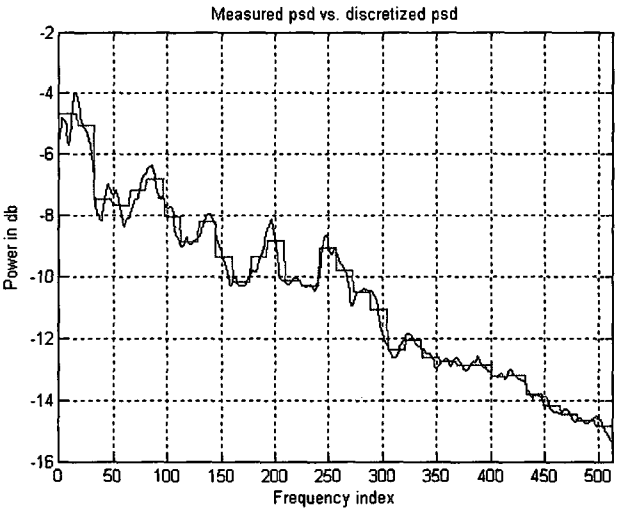

Figure 24: Noise PSD "discretization" procedure for a decomposition with 32 equally spaced bands. The discrete level is returned to the state-space algorithm.

In the above, note also that there is no need to perform any form of all-pole modelling either (since just an average is computed over a few points), this further reduces the overall complexity involved. The benefits of using the solution presented above will be evaluated in Section 8.6.1.

Viewing the noise as a modulated AWGN sequence in each band also allows the use of standalone particle filters to process the noisy speech. This is the subject of the next section.

8.3. Internal noise PSD estimation with subband RBPFs/PFs

The particle filtering algorithms given in Section 3.2.4 (for the linear model version) and in Algorithm 6/7 (Chapter VI, for the nonlinear model version) are naturally "good" at tracking nonstationary white noise, as shown in our previous work [MUS'08-2] (see below for more explanations). Accordingly, in the subband context presented in the preceeding sections, it is proposed here to let each subband PF algorithm assume that the noise can be viewed as an AWGN sequence with time-varying gain. In this essentially novel noise estimation scheme,

the time-varying noise variance (or gain) in each band is estimated on a sample-by-sample basis internally by the PF and by simply drawing for each particle a candidate value in the log-domain (more details are given below). In this option, the fullband colored noise is thus modelled as a combination of bandlimited white noises, in close relationship to the subband method shown in 8.2.2, and the enhancement performed by the RBPFs amounts to try to retain only the more correlated parts of the subband signal measured.

In order to track nonstationary white noise in a PF setting, the idea consists of defining a Gaussian random walk on the log-variance (or log-std) of the observation noise $\ell_v(k) = \log(\sigma_v(k))$, in the very same way that a Gaussian random walk is defined on the log-variance of the model/excitation noise as follows. Recall the set of Equations (15); we now rewrite these equations with an additional one as a complete set:

$$\begin{aligned}
\mathbf{a}(k) &= \mathbf{a}(k-1) + \Gamma_a \alpha(k) \\
\ell_g(k) &= \ell_g(k-1) + \sigma_{\ell_g} \beta(k) \\
\ell_v(k) &= \ell_v(k-1) + \sigma_{\ell_v} \gamma(k) \\
\mathbf{x}(k) &= \mathbf{A}_k \mathbf{x}(k-1) + \mathbf{G}_k \mathbf{w}(k) \\
z(k) &= \mathbf{C}\mathbf{x}(k) + \sigma_v(k) v(k)
\end{aligned} \tag{91}$$

The newly introduced equation (the third one in the set above) involves the zero mean unit-variance Gaussian process $\gamma(k)$, and requires the prior definition of the constant σ_{ℓ_v}. This constant can be chosen independently in each band, although in our implementations we have found that using a single global value yields very good performance. Computationally speaking, this approach is thus interesting, for it only requires the draw of an extra Gaussian pseudo-random number (a noise level candidate) per particle.

The following question now arises: what types of noise can this configuration handle? Obviously, when external noise estimation is present, then practically the algorithm can

handle any noise already handled by the noise estimator. In this case however, one can only expect that any form of noise that is "less correlated" than the speech be affected. For example, a tone in a certain band will almost surely not be removed by the above-described scheme. In practice, we find that this scheme is able to reduce many types of relatively stationary noises containing almost no "high energy lobes" (or harmonic content) in their spectra. In addition, the output signal is in general more natural-sounding than when PSD estimation is employed. One of the possible reasons for this naturalness comes from the fact that candidate "noise levels" are drawn and adjusted at each time instant, allowing for a smoother adaptation of the noise variance, due to the sample-by-sample treatment. In the future, we propose to simply combine both methods, by letting the external noise PSD propose a mean value for the noise around which the PF/RBPF can draw noise level candidates.

Some simulation results for this method can be found in Section 8.6.1.

8.4. Further noise reduction methods taking advantage of subband configurations

8.4.1. Residual noise reduction by low-cost post-processing

In this section, a subband-based post-processing technique is proposed with the following objectives in mind:

1. Remove additional background noise while retaining the positive features of (pre-) enhanced speech (i.e. intelligibility, low distortion, naturalness, etc)
2. As simple and efficient implementation as possible (i.e, aim for low computational complexity).

Both objectives are treated here with equal importance: indeed, if the second objective is not respected, one might as well rework and upgrade the pre-enhancement scheme. On the other hand, if the first objective can be attained with very small additions, then the appeal is more significant for real-world applications already employing certain well-established algorithms. Indeed, in many real-word applications, real time requirements are to begin with hardly met and hence we are interested in improving performance with adding very little computational requirements. Such a concern would for example be applicable to the post-processing method shown in [ZAV'07], in which the non-negligible additional workload consists of a harmonic analysis combined with pitch tracking on the pre-enhanced signal, followed by a (pre-trained) codebook mapping for the restoration of the parts of the signal that were damaged during the initial noise reduction algorithm. In addition, note that the primary goal of *restoring damaged speech components* is fundamentally different from our first objective of *removing excessive residual noise.*

In this section, the objective of the post-processor is not enhancement per se, but rather noticeable background noise removal. Other methods with similar objectives have appeared in the literature; for example the post-filtering method of [WAN'93], based on the detection of formant locations and spectral valleys, is found to perform well for narrowband speech in AWGN. In contrast, the proposed post-processor shown in this section is designed to be incorporated naturally as a module to the subband enhancement architectures of this Chapter, and is meant to operate in the same complex noise conditions.

In simple terms, the idea consists of scaling, on a frame-by-frame basis, the subband pre-enhanced signals depending on the respective estimated levels of speech and residual noise. Note that even in ideal conditions, it would not be desirable to apply such volume-scaling in a

fullband setup, as it would perceivably modulate the amplitude of the signal in a potentially disturbing manner. The generic structure is shown in Figure 25.

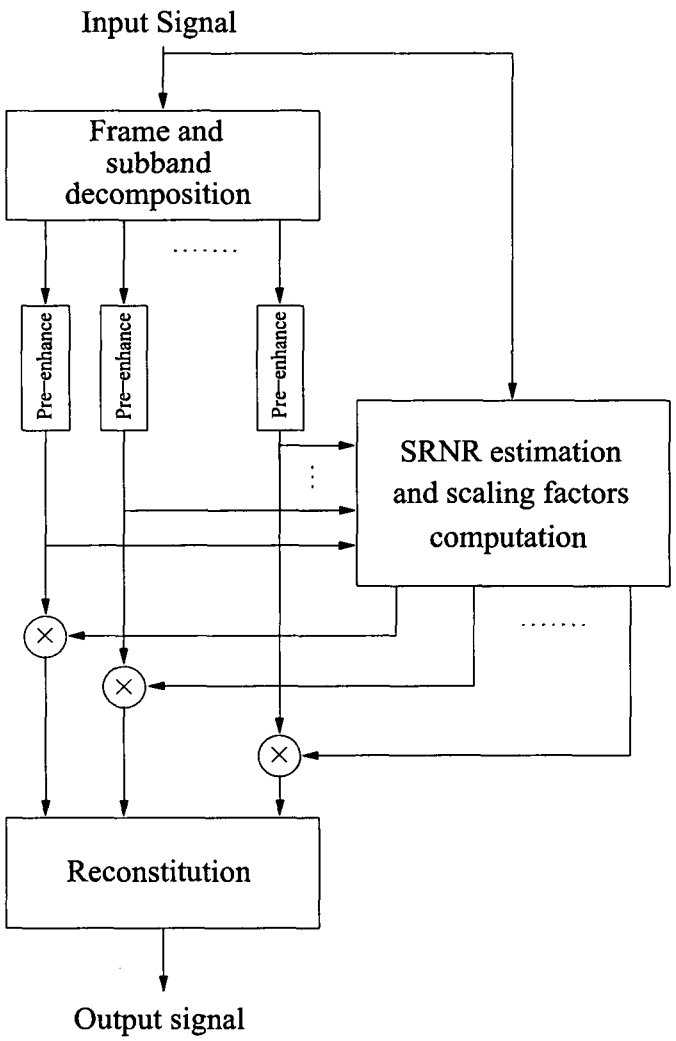

Figure 25 : The proposed post-processing scheme. The Signal to Residual Noise Ratio (SRNR) is estimated from both the noisy fullband and the enhanced subband signals to produce scaling factors which are then applied before reconstruction..

A similar form of subband-signal scaling structure has been successfully applied as the core of a "standalone" subband speech enhancement algorithm (as opposed to a mere "post-processor") in [DIE'04], where subband gains are directly applied to the incoming noisy speech, and are determined from a VAD-based estimation of the a posteriori Signal-to-Noise Ratio. In our context however, the goal is to determine scaling factors to be applied to pre-

enhanced subband speech signals, for which an estimate of the SNR has already been determined or is directly accessible. In addition, as a more important difference from [DIE'04], each subband domain signal (i.e., each of the decimated signals at the outputs of the filters of the filterbank) are here real-valued and can locally be viewed as time-domain signals.

To determine the scaling function in this context, we begin by assuming that the speech and noise statistics are fixed over small frames. Denote by $y_m(:,i)$ denoting the pre-enhanced decimated speech vector at subband m and at the i^{th} frame, assumed to contain the sum of the clean subband vector $x_m(:,i)$ and some residual noise $r_m(:,i)$. Next, suppose that over the i^{th} frame, as sums of random variables $x_m(:,i)$ and $r_m(:,i)$ are approximately i.i.d. with respective distributions $N(0;\sigma_x(i)^2)$ and $N(0;\sigma_r(i)^2)$ (the sequences can indeed by negative-valued, as opposed to spectral amplitudes in usual frequency-domain processing for example). With these assumptions, it is easy to show that, for all k indexing the subband frame:

$$p\big(x_m(k,i)\,|\,y_m(:,i)\big) = N\left(x_m(k,i)\,|\,y_m(k,i)\frac{\sigma_x(i)^2}{\sigma_x(i)^2+\sigma_r(i)^2};\frac{\sigma_x(i)^2\sigma_r(i)^2}{\sigma_x(i)^2+\sigma_r(i)^2}\right) \tag{92}$$

From the above, we can thus write the conditional expected value of $x_m(:,i)$ in terms of the Signal-to-Residual-Noise-Ratio, denoted here by $SRNR_m(i) = \frac{\sigma_x(i)^2}{\sigma_r(i)^2}$, to obtain the post-processed enhanced series $\hat{x}_m(:,i)$ as follows:

$$\hat{x}_m(:,i) = E\big(x_m(k,i)\,|\,y_m(:,i)\big) = \big(1+SRNR_m(i)^{-1}\big)^{-1} y_m(:,i) \tag{93}$$

The gain function is shown in Figure 26 below.

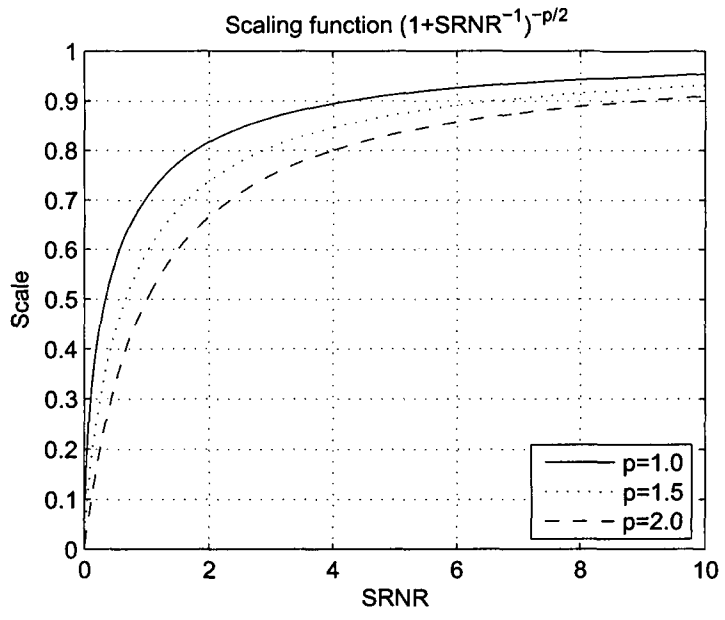

Figure 26: The proposed subband scaling function. When the subband Signal-to-Residual-Noise Ratio $SRNR$ is low, the subband frame is strongly scaled down. As it can also be seen, the parameter $p \geq 1$ is an aggressiveness factor.

As the reader will have noted, in its form the gain function given in Equation (93) is essentially superimposable to an SNR-based frequencydomain filtering formulation of a spectral subtractive gain. Besides the distinct decimated filterbank context and assumptions regarding the nature of the intervening signals, there are notable differences of practical nature: what is proposed here is to reduce the gain to a single number per band and per frame – i.e., to locally reduce it to a scaling factor. In other words, we take advantage of the time/frequency localization of each small frame of data at the output of the decimated filters to formulate some simplifying assumptions resulting in the application of a fixed gain within one subband over a few consecutive samples. To respect this criterion, a "medium" amount of subbands and a relatively small subband frame size is required. As an important practical advantage, our proposed method is both embeddable in existing enhancement algorithms already operating in real-valued filterbanks, and the resulting scaling is very efficient.

Obviously, the above requires the knowledge of $SRNR_m(i)$, which is difficult to accurately estimate as it strongly depends on the method/algorithm used and on the noise conditions. Nevertheless, a practical solution consists of estimating it from $SNR_m(i)$, the signal-to-noise ratio in the current subband frame (assumed to be known from the pre-enhancement stage) – the two are indeed strongly correlated. For this purpose, several methods can be envisioned: For example, using various training data obtained specifically using the chosen pre-enhancement algorithm, some mathematical relationship (e.g. linear regression) between the two sets of subband SNRs could be obtained. This is the object of current research. For the time being, heuristically it was however found that satisfactory preliminary results can be obtained by using the simple following rule:

$$SRNR_m(i) \approx \max\{SNR_m(i), SNR(i)\} \tag{94}$$

In the above rule, the practical value used to represent the residual noise ratio in each subband is simply taken as the maximum between the fullband estimated SNR and the current subband estimated SNR. The rationale for incorporating the fullband SNR was initially based on the observation that in many situations the "local" subband SNR is found to be in discordance with the fullband SNR and thus some low-amplitude speech components that are still important for intelligibility are more at risk of being filtered out. Note also that from Equation (94) we necessarily have $SRNR_m(i) \geq SNR_m(i)$, which is consistent with the expected effect of the pre-enhancement scheme. In practice, to further account for the effect of pre-enhancement, we found that the introduction of a constant $p \geq 1$ is also beneficial, so as to obtain the final rule:

$$\hat{x}_m(:,i) = \left(1 + SRNR_m(i)^{-1}\right)^{-p/2} y_m(:,i) \tag{95}$$

In our implementations, p is set to 1.15. The use of Equation (95) allows for a very low-cost post-processing (one of our primary goals), while the effectiveness of the above solution will

be confirmed in practical tests in Section 8.6.1. Figure 27 shows an example of a post-processed signal in time-alignment with the pre-processed and the noisy signals.

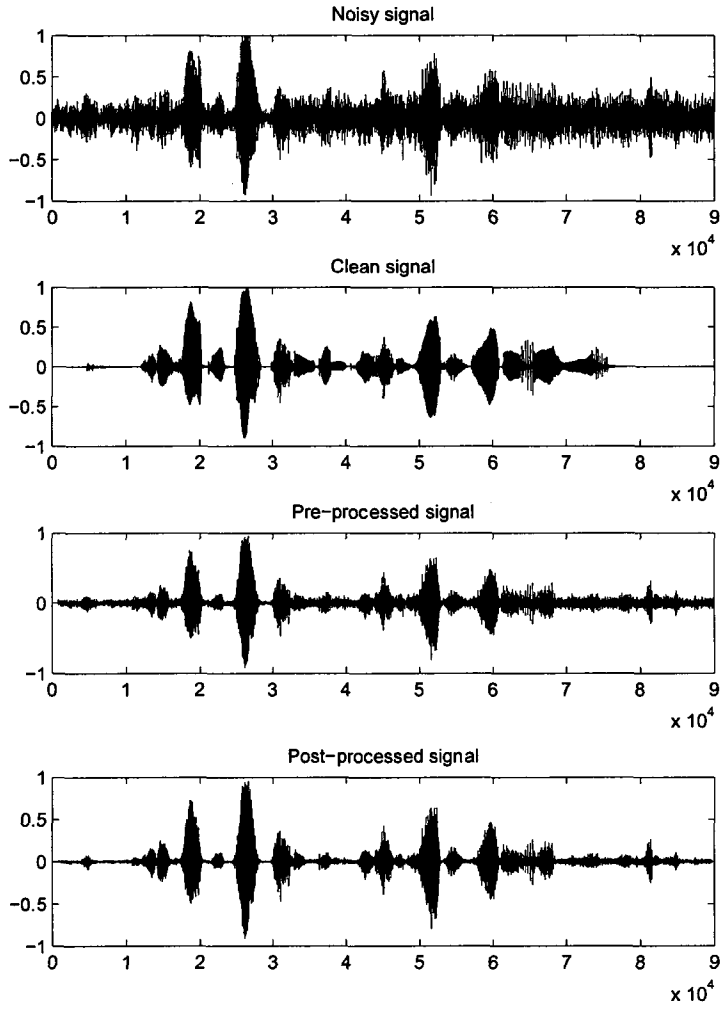

Figure 27: Example of waveforms obtained using the proposed post-processing

In terms of computational complexity, it can be readily seen that the proposed method is extremely low-cost, especially if the pre-enhancement scheme is already frame-based and employing subbands, in which case only one extra equation per band must be applied.

Again, in future research we propose to devise more solid methods to estimate the Signal-to-Residual Noise-Ratio.

8.4.2. Inclusion of perceptual constraints

The subband method resorting to "fully discretized" noise PSD in Section 8.2.2 lends itself very well to psychoacoustic treatment. In this section, a way to include perceptual constraints as part of this method is described. The idea is similar to that shown in the KF case of [CHA'07]; the differences are mainly related to the fact that we are only applying the constraints under certain risk-related conditions to avoid damage to the speech in complex noise conditions.

The central tool used here is the estimated *simultaneous masking threshold* of the clean speech. The simultaneous masking threshold of a signal represents, in the frequency domain, the level/curve below which nothing is audible in the presence of the particular signal being studied. In the ISO MPEG-1 layer 1 psychoacoustic model 1 [ISO'92], a technique to compute such an estimate of a signal's masking threshold is elaborated. In the context of MPEG coding, this is useful to determine how much quantization noise can be introduced while remaining unperceptible. In [SPA'07], a complete description of all the steps involved in the computation of the threshold can be found – it is an implementation of these steps that is used below.

Before moving on to explain how the masking threshold is used in the algorithm of Section 8.2.2, note that an estimated clean signal is to begin with required to compute the threshold; this could constitute a paradox, however in practice we find that a rough clean speech estimate (obtained via spectral subtraction for example) can provide results almost as good as

when the true clean speech is available. In addition, the distinct estimate used can be further used to improve the overall quality by combining it with the state-space algorithm's final estimate, as recommended in Section 4.2. In contrast, [CHA'07] computes the speech masking threshold based on a current estimate of speech signal (itself iteratively obtained via multiple KF passes).

In the algorithm, the masking threshold is used as follows. In a given frame, once the noise power P_z has been estimated and the speech power and the masking threshold P_x and T have been calculated (based on the prior spectral subtractive estimate), first in each band the average level of each of the above quantities is calculated (yielding $\overline{P}_z(m)$, $\overline{P}_x(m)$, and $\overline{T}(m)$), and the following two rules are applied:

1) If $\overline{P}_z(m) < \overline{T}(m) < \overline{P}_x(m)$, then the current data frame is left unprocessed
2) If $\overline{P}_x(m) < \overline{T}(m) < \overline{P}_z(m)$, then the enhancement is made more aggressive by purposely overestimating the corresponding observation variance in the state-space model.

The first rule is based on the assumption that if the noise component in band m is to begin with masked by the speech, then there is no need to perform any noise removal. Next, in the second rule, if the speech component is inaudible but some noticeable noise is present in band m, the enhancement takes place in a more aggressive manner.

Note that the above technique can naturally be followed by the one given in the previous section (Section 8.4.1), and it will in fact later be shown that this yields very good results. We have found that using such conservative rules allows for a less risky solution – and in turn for a more robust solution in nonstationary noise – than the one given in [CHA'07], where the scaling is done based only on the masking threshold level

Simulation results obtained from the application of the above method are given in Section 8.6.1.

As another final note, other perceptual constraints could be used in future works as well, such as temporal masking.

8.5. Using Bandwidth Extension for complexity reduction and quality improvement

In real-world situations, speech enhancement algorithms are typically faced with low or very low SNRs, highly nonstationary noise conditions, and limited computational resources. Regardless of the method used, the output naturalness and intelligibility depend on the adversity of the input conditions. At very low SNR, the enhancer runs the risk of producing a signal with even worse perceived quality than that of its input. Moreover, it is often the case that the SNR in higher bands is to begin with significantly lower than that observed in the lower bands: this is due to the long-term average spectral distribution of the speech energy, as noted for example in [BYR'94]. In practice, this means that while the estimated clean signal in the lower bands might be of acceptable quality, the overall wideband signal is not satisfying.

A set of techniques collectively referred to as Bandwidth Extension (BWE) or Spectral Band Replication (SBR) have recently reached mainstream use and recognition through implementations in high-end telephone systems and popular audio codecs such as MP3Pro™ and AAC+™. These methods use the dependencies between lower and higher bands to infer wideband speech/audio from their lower bands contents. In BWE techniques, it is generally

assumed that no information is available *a priori* about the higher bands, in contrast with SBR audio codec implementations where some coarse spectral envelope parameters are encoded in parallel. In this Section, we are interested in incorporating a simple model-based BWE/SBR method into the speech enhancement context as an assisting tool, with a double objective: that of improving the output speech quality at mid to low SNR, while either reducing or marginally increasing the overall computational requirements of the chosen enhancement scheme.

The solution proposed in this section does not require any *a priori* training or knowledge. It incorporates a model-based BWE solution, where the source/excitation signal is extended by replication, and a coarse wideband envelope is obtained directly obtain from an LPC analysis of the noisy incoming wideband stream. The enhancement algorithm focuses on the narrowband signal, allowing for processing at a lower rate. From its output, a wideband estimate is synthesized using the coarse wideband envelope and the extended source signal.

In the next section, generalities about model-based Bandwidth Extension are given, and in Section 8.5.2, the details of the proposed scheme are shown. Simulation results are reported in Section 8.6.2.

8.5.1. Bandwidth Extension background

In the field of speech coding applied in the context of today's telephone networks, the narrowband nature of the speech signals conveyed is a strong limitation. In order to accommodate wideband speech, a first solution would consist of upgrading all hardware components (terminals, network nodes) in the network, which is obviously bound to be a very costly approach. As an alternative and a projected catalyst, the recent idea of Bandwidth Extension (BWE) was proposed to process the narrowband speech at the receiving end of the

conversation, artificially introducing spectral components to make it perceivably wideband. Many BWE systems resort to autoregressive models of speech, effectively separating the problem into the extension of the source/excitation signal on the one hand, and of the spectral envelope on the other hand [QIA'03, LAR'04, QIA'05].

The basic idea behind such model-based BWE is the following, as illustrated in Figure 28 below. Some notation is necessary: let $x_w(k)$ represent the output wideband speech signal, and $x_n(k)$ represent the narrowband signal. Technically, $x_n(k)$ is in fact the interpolated (i.e. upsampled) narrowband signal so that the notation for the time index k is consistent with the one used for the wideband signal. In the BWE system above, speech signals are modelled as autoregressive signals (see Section 3.1.1), and can thus be generated by passing an excitation signal through an all-pole filter (called the synthesis filter in the BWE context). Let $e_w(k)$ and $e_n(k)$ represent the wide/narrow-band excitation signals. The corresponding autoregressive coefficients of the narrowband and wideband synthesis filter are denoted by $\mathbf{a}_w$ and $\mathbf{a}_n$ (for notational simplicity, assume that the excitation noise variances are included). The idea is the following: from a frame of the signal $x_n(k)$, obtain $\mathbf{a}_n$ (or some equivalent parameters) and from it infer $\mathbf{a}_w$, typically using prior knowledge. This first step is called the *envelope extension*. The next step consists in using $\mathbf{a}_w$ for the analysis filter (i.e., the inverse of the synthesis filter) in order to obtain $e_n(k)$, the bandlimited version of $e_w(k)$. Then, a good overall performance can be obtained by simple modulation of $e_n(k)$ to recover $e_w(k)$, a step called the *excitation extension*. Finally, the wideband frame is generated by passing $e_w(k)$ through the wideband synthesis filter with parameters $\mathbf{a}_w$.

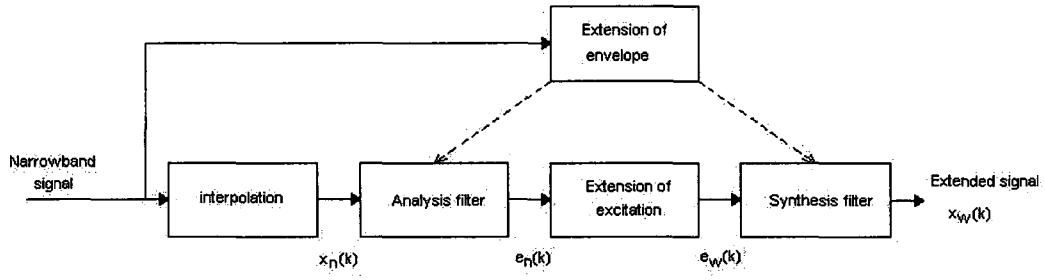

Figure 28: Generic and high-level representation of a bandwidth extension system.

8.5.2. BWE algorithm in the speech enhancement framework

In this section, we propose to use the technique of Bandwidth Extension as part of a speech enhancement algorithm. The idea is indeed appealing for several reasons: Assuming for the sake of presentation that a "perfect" BWE algorithm is available, the following main benefits are envisioned:

- In qualitative terms all of the "resources" of the speech enhancement algorithm can be spent in the narrowband. This can be very beneficial for solutions such as state-space based algorithms, which will typically require a smaller model order (and less particles for the RBPF for example) to achieve their best enhancement performance when the signal's bandwidth is reduced.
- In general, the speech energy in the lower bands is higher than in the higher bands [BYR'94], and accordingly the SNR in lower bands is typically higher (even though this of course depends on the noise spectrum). It is therefore more difficult to effectively remove the noise in higher bands without damaging their speech contents. If a reliable BWE algorithm can be used and a sufficient quality narrowband estimate is available, then the BWE method can be used to strengthen the wideband estimate. The reader might object that a non-ideal BWE itself is known to introduce some artefacts – the tradeoff will be discussed in Section 8.6.2.

The main idea is illustrated in Figure 29 below.

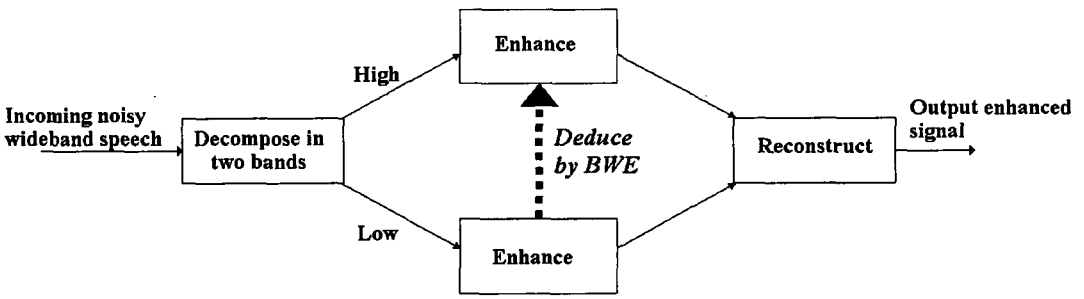

Figure 29: A proposed Speech enhancement scheme resorting to BWE.

For the above to be operational, first a core BWE algorithm is obviously required. In the literature and in the telephony context, the envelope extension algorithm requires prior knowledge and an appropriate training, which could be a strong limitation in speech enhancement. However, as opposed to typical telephony context, it is assumed here that we have access to the wideband noisy speech signal, $z_w(k) = z(k)$. Thus, instead of estimating the clean wideband spectral envelope both from its narrowband counterpart and from prior knowledge, it is proposed here to estimate it using $z(k)$, eliminating the need for prior training. In fact, we have found that a near-transparent base quality can be obtained by merely using an unmodified $z(k)$ at the basis of LPC analysis to obtain the analysis/synthesis filter coefficients – as in AAC+^{TM} codecs for example, it is recognized here that the most importance must be placed in the excitation signal. Specifically, given the true (clean) measured narrowband excitation signal, we compared the output extended speech obtained by using the wideband envelope of $z(k)$ itself at various signal-to noise ratios, and found a very limited impact on the perceived quality.

The proposed "**BWE algorithm for speech enhancement**" (BWE-SE) goes as follows:

- From $z(k)$, obtain $z_n(k)$ by simple low-pass filtering

- Enhance $z_n(k)$ (or practically speaking, a downsampled version of $z_n(k)$ which is the resampled after enhancement) with any algorithm to obtain $\hat{x}_n(k)$
- Obtain an estimate for the analysis/synthesis filter ($\mathbf{a}_w$) by performing an all-pole modelling of $z(k)$
- Filter $\hat{x}_n(k)$ with the estimated analysis filter to obtain the estimated narrowband excitation signal $\hat{e}_n(k)$
- Extend the narrowband excitation signal by simple modulation, as shown in Section 6.3.3 of [LAR'04]. In particular, the method of spectral folding described in [LAR'04] is found to be very well performing (similar techniques are used in base-band speech codecs, such as the GSM full-rate codec) and by far the most efficient excitation extension method.
- Filter the extended excitation with the synthesis filter to obtain the wideband clean speech estimate

In a similar fashion to SBR-based coding, in the excitation step above one can add white noise-like information with appropriate variance so as to reproduce unvoiced sounds when detected. In our experiments, little perceivable difference was found when doing so, and therefore it is left to the practitioner to decide whether this extra step is necessary.

Some simulation results are given in Section 8.6.2. In addition, in Section 8.6.2 we also show the results obtained by:

1. Enhancing $z(k)$ without resorting to BWE, obtaining $y_1(k)$
2. Performing BWE on a downsampled version of $y_1(k)$, obtaining $y_2(k)$
3. Averaging $y_1(k)$ and $y_2(k)$; that is, we verify the speech quality of $\frac{1}{2}(y_1(k)+y_2(k))$

Indeed, from the rationale given in Section 4.2 regarding the combination of estimates of different nature, it is hoped that the above averaging technique should provide some improvement at a very low cost as well. In the remainder of the thesis, this last technique is referred to as the method of "**averaged BWE algorithm for speech enhancement**" (aBWE-SE).

Finally note that, to the best of our knowledge, the computational complexity of common well-performing enhancement algorithms coupled with noise estimation methods, is such that the amount of computations involved for the proposed flavour of BWE is advantageous. Again, this is the case for state-space based solutions, for which a lower state dimension is sufficient when processing only the narrowband signal. Therefore, we claim that for both solutions, the amount of computations involved is in general either reduced (BWE-SE), or marginally increased (aBWE-SE).

8.6. Simulation results

In this section, the simulation results obtained from running the various algorithms and techniques developed in Chapter VII are analyzed and summarized.

8.6.1. Testing of various subband methods

This subsection is concerned with summarizing the simulation results given in details in Appendix B.6. More specifically, the following ideas are tested:

- The **decomposition into 4 bands** explained in **Section 8.2.1**. In the following tables and in the tables of Section B.6, an algorithm "X" employing such a decomposition are coded as "**4B-X**".
- The **decomposition into 32 bands** presented in **Section 8.2.2**. In the following and in Section B.6, algorithms "X" employing such a decomposition are coded as "**32B-X**".

- The **standalone (internal PSD estimation)** particle filtering solution shown in **Section 8.3** is tested on the RBPF algorithm, and denoted by **32B-RBPF$_{(Standalone)}$**.
- The **post-processing method** given in **Section 8.4.1**. Algorithm "X" used with this method is denoted by "**X-Post**".
- The **application of psychoacoustic constraints** as described in **Section 8.4.2**. For an algorithm "X" resorting to this technique, the code "**Ψ-32B-X**" is used.

The above list is summarized in Table 27 below for quick reference.

Technique	Section	Algorithm "X" code
4 bands decomposition	8.2.1	4B-X
32 bands decomposition	8.2.2	32B-X
Standalone RBPF solution	8.3	32B-RBPF$_{(Standalone)}$
Post-processing method	8.4.1	X-Post
Psychoacoustic constraining	8.4.2	Ψ-32B-X

Table 27: Summary and references of algorithms tested

In addition, a combination of psychoacoustic constraining *and* post-processing is tested as well, with code "Ψ-32B-X-Post". The algorithms used are the DKF, the KEM$_{Burg}$, the RBPF, the DEKF4, the DUKF1, the KEM and the NPF. For the first three, all the above flavours are tested (except the standalone solution for the non-RBPF ones), and for the last four only the "Ψ-32B-X-Post" results are published, for the reasons discussed below.

While again, all detailed results can be found in Appendix B.6, we choose to summarize these results into two sets of Tables:

- In Tables 28 and 29, we assess the average benefits of using each of the techniques in Table 27 across several algorithms. This is done by showing the average difference of scores obtained across the first three algorithms for all types of colored noise in VL/L and M/H conditions (respectively) are given. With reference to algorithm "X", the differences shown are those between each of the scores obtained by "4B-X", "32B-X",

"32B-X-Post", "Ψ-32B-X", and "Ψ-32B-X-Post" and the scores obtained from the fullband application of algorithm "X".

- In Tables 30 and 31, the 7 individual algorithms are compared in the context of a "Ψ-32B-X-Post" setup, by averaging the scores obtained for all types of colored noise in VL/L and M/H conditions (respectively).

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
4B-X	1.43	0.53	0.09	0.03	-0.04	0.03	0.01
32B-X	2.91	1.24	0.16	0.09	0.11	0.13	0.13
32B-X-Post	3.78	1.66	0.20	0.09	0.17	0.16	0.16
Ψ-32B-X	3.08	1.40	0.17	0.10	0.15	0.13	0.15
Ψ-32B-X-Post	3.65	1.82	0.23	0.10	0.18	0.18	0.19
$32B\text{-}RBPF_{(Standalone)}$	-1.50	-0.13	0.05	0.02	0.02	-0.03	0.04

Table 28: Estimation of the average benefits obtained by using the subband-based techniques presented in this chapter, in the context of VL/L colored noise conditions. "X" is a generic letter to designate an algorithm to which the techniques are applied – the averages were obtained with 3 algorithms.

M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
4B-X	-0.15	-0.16	-0.01	0.00	-0.14	-0.05	-0.08
32B-X	0.07	0.01	0.01	0.14	-0.03	0.02	0.03
32B-X-Post	1.06	0.63	0.02	0.17	0.08	0.10	0.13
Ψ-32B-X	0.30	0.21	0.02	0.19	0.08	0.05	0.13
Ψ-32B-X-Post	1.04	0.69	0.02	0.22	0.15	0.14	0.24
$32B\text{-}RBPF_{(Standalone)}$	-1.27	-0.55	0.00	-0.06	-0.14	-0.11	-0.08

Table 29: Estimation of the average benefits obtained by using the subband-based techniques presented in this chapter, in the context of M/H colored noise conditions. "X" is a generic letter to designate an algorithm to which the techniques are applied – the averages were obtained with 3 algorithms.

From the results shown in Tables 28 and 29, several conclusions can be made.

First, in VL/L conditions, there are rather clear advantages in using subband methods as opposed to fullband ones, especially so for the 32 bands case. This is all the more obvious when considering the fact that psychoacoustic constraining and post-processing can be readily applied and also provide non-negligible improvements. Next, it is interesting to note from the bottom rows of Tables 28 and 29 that, while the SNR and ASNR scores are lower when internal noise estimation is used, the rest of the measures are not far from those obtained with dedicated, external noise PSD estimation. From informal listening tests, we find that the subband methods are unambiguously better in terms of background noise reduction especially. In particular, it is also noticeable that the "Ψ-32B-X", and "Ψ-32B-X-Post" methods yield a higher signal quality and a better intelligibility. Moreover, we find that the "standalone" method performing internal noise estimation achieves less noise reduction but still preserves well the speech naturaless. Still, as noted in Section 8.3, this method remains interesting in terms of complexity since the internal noise estimation only adds a marginal amount of computations per particle.

Regarding medium to high SNR conditions, the results are relatively more contrasted, in the sense that the 4 bands solution actually yields slightly worse results, in that it marginally

penalizes each objective measures. However, recall that there are still advantages in terms of computational requirements, and thus the 4 bands treatment is still an appealing alternative when compared to fullband processing. On the other hand, the 32 bands case again provides significant advantages when coupled with psychoacoustic constraining and post-processing. In fact, even without any additional scheme, with 32 bands the WPESQ score is improved on average by 0.14 units. Careful listening of the enhanced signals yields observations that are in accordance with the above findings. For instance, it is difficult to differentiate the fullband and the 4 bands case, but improvements become more noticeable with 32 bands, especially with the reduction of background noise.

Finally, in Tables 30 and 31 the average scores obtained by each individual algorithm in a "Ψ-32B-X-Post" configuration are shown.

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
DKF	4.05	-2.59	0.82	1.20	1.45	1.48	1.27
DEKF	2.19	-3.61	0.71	1.14	1.40	1.36	1.19
DUKF	2.31	-3.65	0.67	1.11	1.38	1.36	1.18
KEM	2.17	-3.34	0.75	1.11	1.39	1.38	1.19
KEMburg	4.33	-2.69	0.81	1.18	1.43	1.46	1.26
RBPF	3.36	-2.62	0.79	1.18	1.42	1.46	1.26
NPF	3.38	-2.49	0.81	1.19	1.45	1.46	1.28

Table 30: Comparison between the average scores obtained from using the 7 listed algorithms in VL/L colored noise situations and a "Ψ-32B-X-Post" setup.

M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
DKF	12.22	2.48	0.98	1.80	2.43	2.26	2.11
DEKF	11.78	2.06	0.98	1.75	2.37	2.22	2.09
DUKF	11.31	1.78	0.98	1.72	2.37	2.19	2.06
KEM	10.58	1.17	0.98	1.66	2.31	1.92	1.99
KEMburg	11.20	1.81	0.99	1.75	2.42	2.22	2.10
RBPF	12.13	2.62	0.99	1.84	2.52	2.35	2.39
NPF	11.90	2.48	0.99	1.82	2.52	2.24	2.33

Table 31: Comparison between the average scores obtained from using the 7 listed algorithms in VL/L colored noise situations and a "Ψ-32B-X-Post" setup.

In the VL/L case, two "groups" of algorithms can be formed: First the DKF, NPF, KEM_{Burg}, and RBPF; and secondly the DEKF, DUKF, and KEM, all with markedly lower scores than the algorithms from the first group. Quite interestingly, in this setup it turns out that the very

simple DKF algorithm yields the best CSII, WPESQ, Csig (ex-aequo with the NPF), and Cbak scores – and second-best ASNR, Covl scores. The NPF, KEM_{Burg}, and RBPF still obtain very close (and some better) results (for example the Covl score for the NPF). Still, according to the objective scores the "Ψ-32B-DKF-Post" algorithm may very well be the best subband option in VL/L conditions.

Informal listening tests result in remarks that are in accordance with the above findings. However, while we are able to confirm that the first "group" of algorithms perform significantly better than the second group, we also find that the DKF, NPF, KEM_{Burg}, and RBPF are relatively difficult to tell apart. Nevertheless, while the DKF is able to remove a slightly larger amount of noise, the NPF overall tends to sound more natural.

In M/H conditions, the same algorithms can be separated into two groups. This time however, the RBPF and NPF both stand out – although the DKF is not far behind. Our subjective impressions, from listening to the enhanced speech files, agree with the above, but we also find that RBPF and NPF are this time more distinguishable from the rest, with crisper and higher quality speech.

8.6.2. Bandwidth extension

In this section, the simulation results obtained from the application of Bandwidth Extension for speech enhancement (which we call below "BWE-SE" – see Section 8.5, or more specifically Section 8.5.2) are summarized and analyzed.

The complete tables of simulation results can be found in Section B.7. The idea was applied to three algorithms which have already been well tested in the previous chapters, namely the DKF, the KEM_{Burg}, and the RBPF.

For these simulations, the following methodology is followed:

- The input noisy signals are downsampled by 2 before being sent for processing to the 3 algorithms.
- All the "resources" that were allocated for a certain algorithm "X" in previous chapters are allocated as well to them for the BWE-SE context. What this practically means is that for the KEM_{Burg}, the same number of iterations is used; for the RBPF, the same amount of particles is used, etc.

In Table 32, the average difference of scores obtained from using the BWE-SE technique (as opposed to not using it) across all types of algorithms and colored noise conditions are shown. These entries correspond to the row named "BWE". In addition, an extra row named "Average" shows the score obtained by the simple average of the signals enhanced by the "regular" algorithms (not using BWE-SE) and their bandwidth extended downsampled versions (see Section 8.5.2 for more details).

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
BWE	0.56	0.18	-0.05	0.02	0.19	0.01	0.08
Average	0.57	0.23	-0.02	0.02	0.23	0.02	0.12
M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
BWE	0.33	0.24	-0.01	0.06	0.22	0.02	0.09
Average	0.89	0.28	0.01	0.10	0.27	0.09	0.16

Table 32: Average differences in scores (across 3 algorithms and 4 colored noise types) from using the BWE technique applied to speech enhancement. For example, in the top-left entry it appears that using the BWE-SE technique improves on average the SNR by 0.56 dB in VL/L noise conditions.

Surprisingly, the results in Table 32 suggest that the use of BWE in the context of speech enhancement is quite beneficial. The "surprise" factor comes from the advantageous SNR and ASNR scores – before running the simulations, it had been anticipated that these scores would almost surely be lower than the non-BWE ones, since the upper-half spectral part of the wideband signal is artificial and its phase does not necessarily correspond to that of the true

signal's. What possibly "makes up for this" is the fact that in general, the SNR/ASNR in the higher frequency bands are to begin with significantly lower than in the lower bands. In other words, the above results suggest that a better estimate for the clean higher bands is obtained from extending the lower bands, rather than from enhancing the noisy higher bands. Moreover, looking at the "Average" entry in Table 32, a combination of the two actually yields even better results, which is intuitively sound.

Figure 30 below shows some example spectrograms of segments from the noisy, clean, enhanced without BWE and enhanced with BWE signals. In the spectrograms, it is interesting to note that while the BWE-based algorithm introduces artificial high-frequency components that may not exactly match those appearing in the original clean signal's, but that are still located in the correct "time-frequency" region (see for example the region around the coordinate [2000,0.6]). In contrast, the signal enhanced without BWE completely misses some of these features.

Next, from informal listening tests, we find that overall, the BWE-enhanced signals and the regular, fullband-enhanced signals are perceivably speaking very close, thereby justifying the lower required computational resources. As the main downside, occasionally the extended speech can still sound relatively bandlimited, in the sense that some vowels can sound more "muffled" for the BWE case. On the other hand, an improvement can be noticed in terms of noise reduction at lower SNR. At low to very low SNR, the artifacts introduced by BWE in the higher bands begin to appear, but they are no less natural than the artefacts present in higher bands produced by direct fullband processing. In other words, usual residual noise masking techniques such as the re-addition of a small fraction of noisy speech to the enhanced product are applicable as well. In summary, the perceived naturalness of the output signal is not adversely affected by the use of BWE, comparatively to regular fullband enhancement.

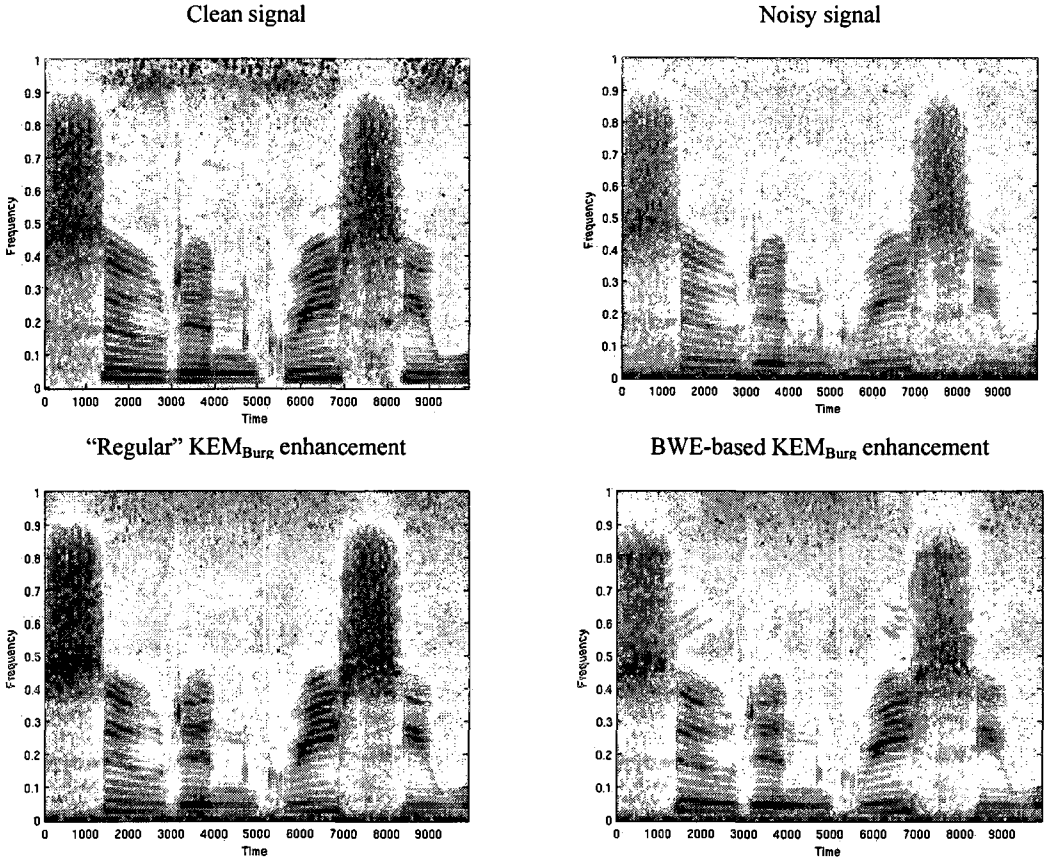

Figure 30: Examples of spectrograms showing the difference between BWE-based and regular speech enhancement. On the bottom-right figure, the artifical nature of the high-frequencies is apparent, for example around the [2000,0.6] coordinates.

In conclusion, the above method is recommended for speech enhancement not only because it is beneficial in terms of speech quality, but also because it only adds very few computations to the overall algorithm.

8.7. Conclusions

Many conclusions flow from the findings of this Chapter. For clarity, they are given in the form of a list below:

- From Section 8.2.1: Compared to fullband processing, using 4 bands processing does bring some noticeable improvement at very low and low SNR, especially in terms of

background noise reduction. But at medium and high SNR, the performance is globally slightly worse, both with objective scores and subjectively speaking (specifically, in terms of naturalness).

- From Section 8.2.2: Using 32 bands is significantly better than both fullband and 4 bands processing at very low and low SNR, and moderately better than fullband processing at medium/high SNR. Therefore, since there are no drawbacks we conclude that the 32 bands treatment is superior to the fullband treatment.
- From Section 8.3: The "standalone" RBPF clearly does not remove as much noise as the solutions resorting to noise PSD estimation. Nevertheless, the CSII and WPESQ scores are not far from the fullband solution, especially at very low and low SNR. While still viable and less computationally expensive than the other RBPF solution, the "standalone" RBPF is outperformed on too many levels by the other algorithms to be considered an appealing alternative.
- From Section 8.4.1 and Section 8.4.2: The post-processing algorithm is found to be clearly beneficial for all noise conditions, and consistently so when applied to each algorithm tested. The improvement in terms of background noise reduction and interspeech residual noise is noticeable. Using psychoacoustic constraints leads to non-negligible improvements as well, especially at high SNR. These two simple but well-performing methods further show the superiority of the 32 bands solution.
- Accessorily, it was found that when both psychoacoustic constraints and post-processing are used, the best two algorithms at very low and low SNR are the DKF and the NPF, followed by the RBPF and the KEM_{Burg}. At medium/high SNR, the two best ones are the RBPF and the NPF, however the DKF and KEM_{Burg} are not far behind – In fact, perceivably speaking there is only very small noticeable differences. In conclusion, due to both their simplicity and their very good performance, the DKF

and the KEM_{Burg} are our recommended choices in general. In addition, if naturalness is the most important aspect of the enhanced product, then if computational resources allow it the NPF may be preferred.

- From Section 8.5, it was found that the Bandwidth Extension method applied to speech enhancement yields surprisingly good results, in the sense that the averaging method is able to improve consistently all objective scores. Accordingly, it is found from informal listening that the improvement is non-negligible. These findings are all the more interesting that the method is even less computationally expensive.

IX. An example of state space-based algorithm for complex conditions

In this Chapter, we take advantage of the conclusions of several individual chapters and sections of this thesis and "piece together" an example algorithm capable of producing enhanced signals that are as close as possible to the objective of this thesis, as discussed in Chapter I. Note that this Chapter neither summarizes nor utilizes all the contributions of this thesis. Rather, we suggest a possible state space-based algorithm capable of handling real-world noises with an emphasis on naturalness and low-distortion. Additionally, we question where this proposed algorithm stands amongst state-of-the-art speech enhancement algorithms.

The first section refers to different sections from this thesis to explain how the proposed algorithm was devised. Next, it lists three well-established algorithms that have been shown to perform strongly in respected (and cited) publications.

The second subsection pertains to the discussion and analysis of simulation results performed on the four algorithms together.

9.1. Example of suggested algorithm and existing algorithms considered

9.1.1. An example of suggested state-space based algorithm

To begin with, it was chosen to place the treatment in the subband context, since Chapter VII indicates that the best results among the conditions tested can be achieved with 32 bands analysis.

Next, the choice for the core algorithm was based on the following constraints:

1. Good compromise between execution time and enhancement capabilities
2. Good "robustness" (low susceptibility to changes in noise conditions)
3. Emphasis in performance in VL/L conditions, rather than in M/H conditions
4. Accent on naturalness, at least as much as on noise reduction

Observing the various tables populating Chapter VII and judging from experience, it is found that if subband treatment is to be used, the KEM_{Burg} or the DKF algorithm can answer well the above criteria. If the strongest emphasis had been on M/H conditions, an RBPF could have been proposed. Regarding criteria 3 and 4 above however, we recall that a slight edge on naturalness in VL/L conditions was achieved with neural-based PFs. To respect criterion 1 and still include NPF processing, it was decided to include small NPFs in bands 7 to 14 (with band 1 referring to the lowest frequency band and 32 to the highest), that is, in the frequency region between 1875 to 4375 Hz. The reason for this choice is twofold: first, a large part of the speech energy lies in this frequency range. Secondly, due to its benefits the "averaged" Bandwidth Extension solution from Section 8.5.2 was chosen to be applied, and thus a reliable estimate for the clean speech in lower frequency bands was required.

Next, to maximize the enhancement performance, the following tools were implemented:

- All low-cost improvement techniques shown in Chapter I, including the small-size PFs (more exactly NPFs in our context) combination in Section 4.3.

- The combination of algorithms described in Section 4.2 was performed via a very simple implementation of the classical spectral subtraction algorithm (as described in [BOL'79]), adding minimal computational overhead.
- The joint, unbiased, multiple neurons NPF from Chapter V was used in 4 of the 32 bands, but since it is operating in small subbands the parameters used were $M_s = 3$ and $P = 2$.
- The noise PSD (estimated using the technique from Section 3.3), was incorporated as shown in Section 8.2.2, forcing each algorithm to treat the subband noise as an AWGN sequence with time-varying gain (and reducing the overall complexity)
- The psychoacoustic contraining of Section 8.4.2 was implemented
- The "averaged BWE algorithm for speech enhancement" from the end of Section 8.5.2 was also implemented.

Since subband processing is preferred, no methods from Chapters V and VI were used. In addition, the low-cost post-processing of Section 8.4.1 was not implemented – for a fair comparison it would otherwise be necessary to implement it as well for the other existing algorithm.

9.1.2. Existing state-of-the-art algorithms used for comparison

The algorithms used for comparison were chosen based on their level of recognition and their reported performance. In addition, they are fundamentally different from each other so as to avoid overlaps and redundant comparisons.

The three algorithms are the following:

- The statistical-based (**LMMSE**) algorithm previously mentioned in this thesis, shown in [EPH'85]. This algorithm is well-established and its performance is very well rated (see the reference [HU'06-2] where extensive subjective testing is performed on various algorithms).
- The **multi-band spectral subtractive** algorithm (**MSSUB**) shown in [KAM'02] (and referred to in [LOI'07]), which is shown to largely outperform the traditional spectral subtraction algorithm, and is targeted towards the handling of real-world noise as noted in the paper. As the **LMMSE** algorithm, it is shown in [HU'06-2] to perform very well: "*Overall, the statistical-model based methods performed the best across all conditions, followed by the multi-band spectral subtraction method*".
- The generalized subspace approach (**KLT**) given in [HU'03], found in [HU'06-2] to outperform the other recent and state-of-the-art subspace algorithms from [JAB'03], and considered to be one of the best subspace-based algorithms.

The MATLAB implementations for each of the above three algorithms have been obtained from the accompanying CD-Rom for the book [LOI'07], and have been slightly adapted to incorporate the very same noise PSD estimator as the one used in the proposed algorithm.

9.2. Results summaries and analysis

The following tables 33 and 34, containing average results, were compiled based on the full tables given in Section B.8, to which the reader is welcome to refer to.

VL/L	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-4.45	-6.71	0.37	1.05	1.18	1.15	1.03
MSSUB	2.34	-3.52	0.59	1.24	1.54	1.42	1.28
LMMSE	3.86	-2.30	0.72	1.20	1.54	1.45	1.28
KLT	3.08	-1.65	0.65	1.07	0.64	1.32	0.69
Proposed	**6.48**	**-0.88**	**0.82**	**1.41**	**1.67**	**1.61**	**1.71**

Table 33: Average comparative results in colored noise for VL/L conditions.

M/H	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.67	-0.23	0.93	1.43	2.22	2.00	1.90
MSSUB	8.36	0.99	0.94	1.84	2.56	2.26	2.17
LMMSE	13.09	2.98	0.97	1.68	2.39	2.28	2.03
KLT	12.03	3.13	0.97	1.42	2.10	2.22	1.81
Proposed	**12.36**	**2.49**	**0.99**	**1.92**	**2.60**	**2.35**	**2.27**

Table 34: Average comparative results in colored noise for M/H conditions.

From the average scores given in Table 33, it is apparent that the proposed method significantly outperforms the other three algorithms in very low and low conditions. In medium and high conditions (Table 34), the CSII, WPESQ, and composite scores are still non-negligibly above the other three algorithms, but the SNR and ASNR are not (although they are still in the same ballpark).

Subjectively speaking, from informal listening tests it is concluded that the proposed algorithm does overall sound more natural that the other three algorithms, and that it clearly performs better than them in the adverse conditions represented by Table 33. At higher SNR, the proposed method tends to yield less noise reduction, but the clean signal suffers less distortion, and thus is again preferred if naturalness is the main selection criterion.

X. Other topics and Future work

This chapter contains some research directions for future work, as well as some partial results from ideas under development.

10.1. Improvement of excitation noise model for state-space based algorithms

In the course of this research, it was decided to try and explore which areas in the generic models for state-space based algorithms needed the most improvement. Since it is the most autonomous of all algorithms presented in this work (and one of the most robust and performing), the following actions were taken in the context of a Rao-Blackwellized Particle Filter running on a linear speech production model under white Gaussian noise circumstances, using a large amount of particles:

1. Run the RBPF with "default" settings, i.e., assuming that the AR parameters and the excitation signals are completely unknown
2. Run the RBPF while feeding it AR parameters measured from the true clean speech, and still assuming that the excitation signal is unknown
3. Run the RBPF assuming that the AR parameters are unknown, except the exact variance of the excitation signal
4. Run the RBPF assuming that the AR parameters are unknown, but that the excitation signal inferred from the true clean speech is known

Our findings can be summarized as follows. First, solutions 1 and 2 yield almost equivalent results. Secondly, solution 3 produces noticeably better results than solutions 1 and 2, and finally solution 4 yields significantly higher-quality speech than all other solutions. In other

words, the prior knowledge of AR parameters only has little impact on the performance of the RBPF. However, as soon as some reliable information about the excitation signal is available, then a strong improvement is noted. Our conclusion is that one of the major performance limiters in the state-space based algorithms used in this thesis is the modelling and handling of the excitation noise.

The above considerations led to some efforts in trying to find better way of coming up with dependable estimates for the excitation signal while still using the algorithms of this thesis. As a solution, our current research focuses on the use of the mature field of Code-Excited Linear Prediction (CELP) specifically for enhancement purposes. Indeed, it appears that CELP coding would be naturally suited to fulfil the need for better excitation models in speech enhancement algorithms resorting to AR models of speech. In "regular" CELP coding performing Analysis by Synthesis (AbS), the spectral envelope of a clean speech signal is matched by some all-pole modelling technique (in fact, in recent works, some of the techniques shown in Chapter VI have been successfully used), and then from a fixed and/or adaptive codebook, an appropriate excitation signal is retrieved. It is the latter part that is of special interest to us.

In our current preliminary implementation, the main idea is the following, given a certain noisy speech frame:

- Obtain an estimate for the background noise power spectral density
- Estimate (by any means) the spectral envelope of the clean speech
- Perform a codebook search (along with long-term prediction) for an excitation signal that most closely matches the measured PSD of the noisy speech to the combined PSDs of the speech (obtained from its AR envelope) and of the estimated noise. Some perceptual weighting could also be considered.

At this stage, the above remains untested, although with primitive implementations and multiple simplifying assumptions we have observed interesting enough results for us to encourage and welcome any further work on the subject.

10.2. Binaural extensions

In many real-world applications, the noisy speech signal is picked up by more than one microphone. We are interested here in the binaural case, i.e., when two microphones on opposite sides of the head are used to listen to the noisy signal. Obviously, a possible way of enhancing the two channels consists in doing so independently. The question that is posed in this section is the following: In the context of state-space based algorithms, are there any benefits in coupling the enhancement between the two channels, both in terms of output quality but also of computational complexity? To answer this question, two possible configurations are currently being investigated.

10.2.1. "One source, 2 channels" model

In this first "natural" solution, the situation is modelled as in Figure 31.

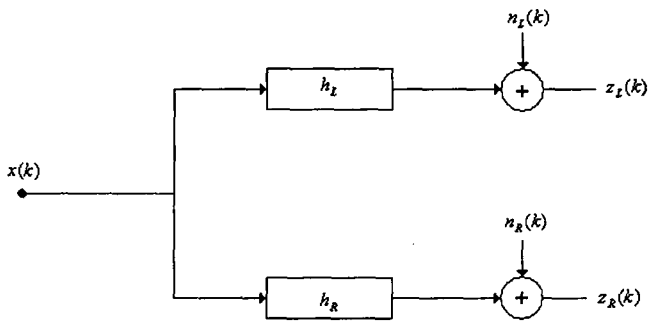

Figure 31: Binaural model, with one source and two channels

In this situation, the clean speech signal $x(k)$ travels to the left ear (respectively right) through the model filter (or HRTF) $\mathbf{h}_L$ (respectively $\mathbf{h}_R$), and is corrupted by an additive

noisy signal $n_L(k)$ (respectively $n_R(k)$); the signal obtained is named $z_L(k)$ (respectively $z_R(k)$).

Assuming that the HRTFs and noise statistics are known (or estimated via some external algorithm), then a (conditionally) linear and Gaussian state-space model can still be written, making all the algorithms in this thesis applicable. It is in fact also still possible to use the colored noise handling devised in Chapter V. Without getting into the cumbersome details, the equations can be globally rewritten as:

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}_k \mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k) \\ z_L(k) &= \mathbf{h}_L{}^T \mathbf{x}(k) + n_L(k) \\ z_R(k) &= \mathbf{h}_R{}^T \mathbf{x}(k) + n_R(k) \end{aligned} \tag{96}$$

, where the lengths of the vectors have been properly adjusted.

10.2.2. "Two sources, 1 channel" model

While complete and correct, the fact that the HRTFs are required in the measurement equations can make the previous model difficult to handle. The length of each HRTF is commonly fairly large compared to the speech model order, resulting in several relatively heavy algebraic manipulations in AR-modelled colored noise. This is especially true in the particle filtering context, which is in addition affected by potentially large memory requirements since each trajectory (corresponding to one particle) is different. Note also that the monaural source speech $x(k)$ is not what is ultimately desired, rather we are interested in the left and right signals represented by $\mathbf{h}_L{}^T \mathbf{x}(k)$ and $\mathbf{h}_R{}^T \mathbf{x}(k)$, with undamaged spatial and temporal cues. The wrong choice of HRTFs can therefore directly result in both estimation problems and "cues problems".

In an attempt to minimize and conceal the above problems, the following approach is proposed, based on the idea shown in Figure 32.

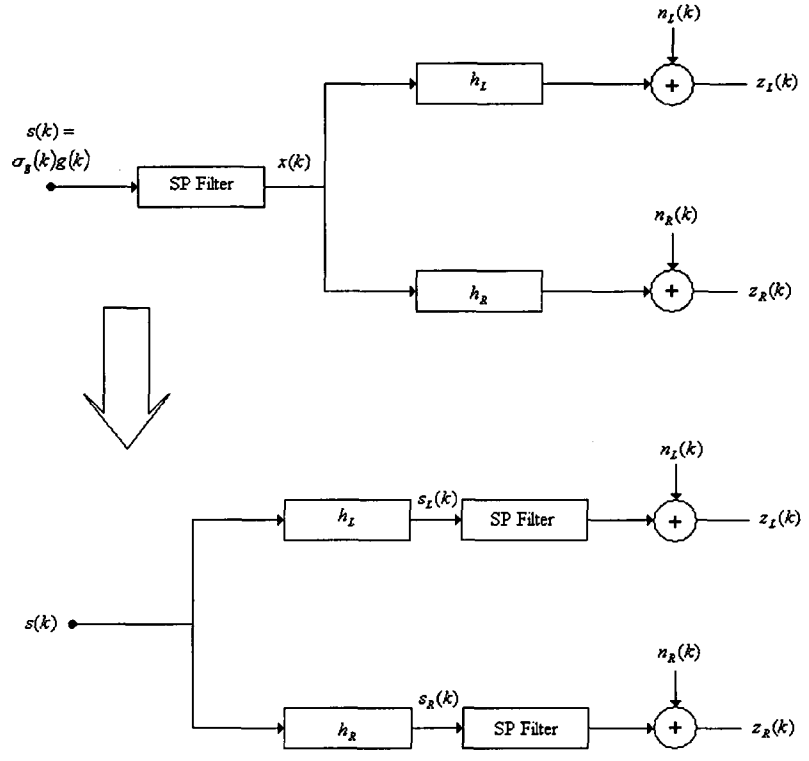

Figure 32: Alternate representation of the binaural model of Figure 31. The block with label "SP Filter" denotes the speech production filter (or vocal tract filter), determined by the autoregressive coefficients.

Recall that in the "excitation+filter" model for speech (Section 3.1.2), the speech signal can be broken down into an excitation signal (denoted by $\sigma_g(k)g(k)$ throughout this thesis) and an all-pole filter determined by a set of autoregressive parameters (denoted by "SP Filter" in Figure 32) – this decomposition is shown on the top part of the Figure. On the bottom part the order of the speech model's filter and the HRTFs was changed, and the signals $s_L(k)$ and $s_R(k)$ were introduced as the "filtered" excitation signals (In general, this interchange can only be done for LTI systems; it is applied here in the context of frames over which the signals are considered to be short-time stationary, which is the case for most state-space

algorithms in this thesis. Thus, the assumption is that the HRTFs and the speech signals will not drastically vary over a short period of time). With the assumption that the source signal $s(k)$ is a white and Gaussian process, one can deduce that:

1. $s_L(k)$ and $s_R(k)$ are jointly Gaussian
2. They are correlated
3. Individually, each of them is not iid (since they are obtained as the filtered version of an iid process)

With the above in mind, consider only the right-hand side of the bottom part of Figure 32, redrawn below:

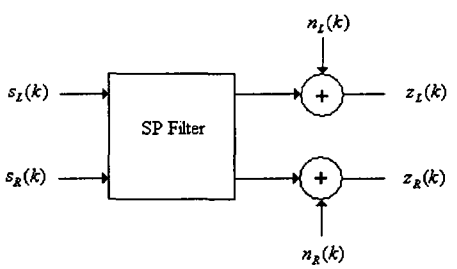

Figure 33: Rewritten binaural model. Again, the block with label "SP Filter" denotes the speech production filter determined by the autoregressive coefficients.

Letting $x_L(k) = \mathbf{h}_L{}^T\mathbf{x}(k)$ and $x_R(k) = \mathbf{h}_R{}^T\mathbf{x}(k)$, then the corresponding state-space model is:

$$\begin{bmatrix}\mathbf{x}_L(k)\\ \mathbf{x}_R(k)\end{bmatrix} = \begin{bmatrix}\mathbf{A}_k & 0\\ 0 & \mathbf{A}_k\end{bmatrix}\begin{bmatrix}\mathbf{x}_L(k-1)\\ \mathbf{x}_R(k-1)\end{bmatrix} + \mathbf{G}_s\begin{bmatrix}s_L(k)\\ s_R(k)\end{bmatrix}$$
$$\begin{bmatrix}z_L(k)\\ z_R(k)\end{bmatrix} = \begin{bmatrix}x_L(k)\\ x_R(k)\end{bmatrix} + \begin{bmatrix}n_L(k)\\ n_R(k)\end{bmatrix} \qquad (97)$$

where $\mathbf{G}_s = \begin{bmatrix}1 & 0_{1\times M_s-1} & 0 & 0_{1\times M_s-1}\\ 0 & 0_{1\times M_s-1} & 1 & 0_{1\times M_s-1}\end{bmatrix}^T$, a $2\times 2M_s$ matrix.

With the above system, we can retain two of the three properties listed above regarding $s_L(k)$ and $s_R(k)$, and still apply KF-based algorithms in a simplified and lighter set of equations. Assuming that the covariance matrix of the speech excitation vector $\mathbf{s}(k) = [s(k) \quad s(k-1) \quad \cdots \quad s(k-L+1)]$ is $\mathbf{Q}_s(k)$, with L still being the length of the HRTFs (namely, $\mathbf{s}(k) \sim N(0, \mathbf{Q}_s(k))$), then the covariance matrix of the process noise in the system above can be written as:

$$\mathbf{Q}_{LR} = \begin{bmatrix} \mathbf{h}_L^T \mathbf{Q}_s(k) \mathbf{h}_L & \mathbf{h}_R^T \mathbf{Q}_s(k) \mathbf{h}_L \\ \mathbf{h}_L^T \mathbf{Q}_s(k) \mathbf{h}_R & \mathbf{h}_R^T \mathbf{Q}_s(k) \mathbf{h}_R \end{bmatrix} \tag{98}$$

which, if $\mathbf{Q}_s(k) = \alpha_k I$, simplifies to:

$$\mathbf{Q}_{LR} = \alpha_k \begin{bmatrix} \|\mathbf{h}_L\|^2 & \mathbf{h}_R^T \mathbf{h}_L \\ \mathbf{h}_L^T \mathbf{h}_R & \|\mathbf{h}_R\|^2 \end{bmatrix} \tag{99}$$

Hence, all the properties induced by the model in Figure 32 are included, except that in the KF context, samples from the process noise are considered statistically independent, which is not the case in the model.

Regardless of the anticipated benefits in terms of output quality (as opposed to simply processing each side separately), there are obvious benefits in terms of computational complexity. For example, two completely independent PFs with N particles (overall $2N$ particles) will run slower than a joint PF with only N particles, while yielding inferior performance.

In order to assess the potential of the method, the following paragraphs contain simulation results in the context of uncorrelated additive White Gaussian noises on each side, and HRTF-

filtered binaural speech. For simplicity, the target is chosen to be frontal, and matrix $\mathbf{Q}_s(k) = \alpha_k I$ to be diagonal. To tackle this problem, the algorithms chosen is the RBPF, with an added dimension in the state used to estimate the constant α_k sequentially.

In Table 35 below, the left WGN data is the same one used throughout this thesis, and the right WGN data is simply a time-reversed version of the left one (creating an uncorrelated, same SNR stream of white Gaussian noise that is easy to obtain). The speech material is also the same, but again has been filtered with standard HRTFs (specifically, the ones from MIT Media Lab available at http://sound.media.mit.edu/resources/KEMAR.html). Each entry in the table has the form "X/Y" where "X" is the left channel's results and "Y" the right one.

VL-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.93/-4.02	-7.31/-7.40	0.07/0.07	1.03/1.03	0.14/1.18	1.25/1.25	0.50/0.53
RBPF	6.88/6.57	-1.26/-1.34	0.35/0.37	1.06/1.06	0.40/0.36	1.53/1.57	0.62/0.65
Bin. RBPF	8.15/7.75	-0.61/-0.81	0.47/0.40	1.29/1.17	0.69/0.66	2.21/1.98	1.33/1.16
L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.10/-0.17	-5.38/-5.79	0.20/0.21	1.03/1.03	0.52/0.54	1.50/1.51	0.80/0.80
RBPF	9.56/9.61	0.44/0.55	0.73/0.73	1.09/1.11	0.76/0.80	1.80/1.85	0.92/0.95
Bin. RBPF	10.78/11.21	1.29/0.95	0.81/0.79	1.40/1.41	1.02/1.35	2.24/2.30	1.16/1.52
M-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.89/5.36	-2.01/-2.25	0.55/0.49	1.05/1.05	1.19/1.16	1.95/1.86	1.30/1.23
RBPF	13.87/13.57	3.02/2.75	0.95/0.95	1.31/1.28	1.47/1.44	2.23/2.19	1.48/1.46
Bin. RBPF	15.03/14.95	4.01/3.47	0.97/0.97	1.56/1.45	1.81/1.73	2.85/2.81	2.16/1.94
H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	12.23/12.05	1.61/1.58	0.92/0.90	1.21/1.18	1.93/1.89	2.64/2.58	2.01/1.98
RBPF	18.66/18.61	6.95/6.56	0.98/0.98	1.88/1.84	2.20/2.19	2.94/2.93	2.23/2.21
Bin. RBPF	19.09/19.07	7.53/7.46	0.99/0.99	2.13/2.06	2.53/2.45	3.35/3.31	2.88/2.73

Table 35: Monaural vs. Binaural RBPF enhancement comparisons in WGN. The "RBPF" entries correspond to independent left/right treatment of the noisy binaural signal. The "Bin. RBPF" consists of the joint treatment presented in this section. The results are formatted as X/Y where X corresponds to the left-channel results and Y to the right-channel results.

The objective scores above are clearly in favour of the binaural setup, which is also confirmed by informal listening tests. As a conclusion, this method is promising, because it is possible to use directly in the context of real-world noise as in Section 8.2.2. In some preliminary tests using advanced binaural noise PSD estimators (e.g. [KAM'09, KAM'10]), we find that the resulting method is in fact able to reduce very complex noise such as interfering speakers,

clattering dishes, and diffuse noise altogether, while still retaining natural quality outputs. Future work is centered on further testing this solution.

XI. Conclusion

In respected and recent publications on the subject of speech enhancement (e.g. [HU'06-2], [LOI'07]), state-space based algorithms have been omitted, possibly due to their lack of robustness and their assumed complexity. This thesis attempts to show that this family of algorithms can be turned into a viable alternative, with in fact very good performance achievable at low SNRs, outperforming several well-established algorithms.

In order to explore and reinforce the capabilities of current state-space based algorithms when faced with real-world colored noise situations, the following was done. In Chapter II, a definition of the performance metrics and methodology was given. Then, as a first step, in Chapter III a literature review was conducted during which it was found that each enhancement algorithm resorting to state-space models essentially operates in the same manner, with the only major difference being the way that the clean speech parameters (i.e. autoregressive coefficients) are determined before running one of a KF/EKF/UKF/PF algorithm. Broadly speaking, the determination of the speech parameters fall in one of three categories: Dual methods, Optimization-based methods, and Holistic methods. Through experimentation, it was concluded in Chapter III that the Dual Kalman Filter (DKF), the Burg-based optimization method (KEM_{Burg}), and the RBPF were good ambassadors for each category. Globally, the RBPF was found to perform better than the KEM_{Burg}, with the DKF in last place – however, it was also noted that for the most part, the enhanced products from all the algorithms tested are difficult to distinguish, which is an observation in favour of less computationally expensive algorithms.

In Chapter IV, several techniques that are readily usable, very low cost, and which can improve the speech quality at the output of each state-space based algorithm are presented. They are based on delayed the estimates, combining various speech enhancement estimates to form a better one, and doing the same for noise PSD estimates. In addition, an improvement technique specifically targeted at PFs is shown, with the conclusion that two or more small-size PFs (i.e. handling few particles) produce better quality clean speech estimates than one large-size PF. These results are adopted as standard throughout the rest of the thesis, that is, it is always understated that they have been used when publishing simulation results.

The issue of colored noise speech enhancement is approached as central matter for the first time in Chapter V, where a novel way of handling autoregressive noise signals within the state-space based algorithms is devised. Based on a simple rewriting of the state-space equations representing the situation, it is found that the computational complexity can be significantly decreased, without penalizing the output quality. In fact, in several cases the enhancement performance is actually found to increase. The new method can also be used with any state-space based algorithm, and is systematically used throughout the rest of the thesis whenever a colored AR observation noise is required. While conducting simulations, it was accessorily found that the white-Gaussian noise hierarchy established for the RBPF, the KEM_{Burg}, and the DKF does carry over to the colored noise case.

Considering the existing algorithms reviewed in Chapter III and based on the range of application of Particle Filters, it was concluded that a PF framework for nonlinear speech production models (specifically, formed by feedforward neural-networks) was missing in the literature. In Chapter VI, such a framework is introduced, and many configurations are tested. Amongst them, the best results were obtained with a nondual, multiple-neuron, and nonbias

algorithm. Compared with the RBPF (which was until then the best rated algorithm), the neural-based PFs (or NPFs) are found to perform better at very low and low SNR conditions, whereas the RBPFs are better at higher SNRs. The "classical" RBPF for speech enhancement is, however, very close in terms of performance, and is also easier to "configure" (i.e., to set its parameters to ensure good performance and robustness).

While in Chapter V, simulation results and an efficient method of handling colored noise in state-space based algorithms were already shown, the question of how to properly represent the noise PSD as AR coefficients was not addressed. Chapter VII raises this question, and besides the simple Yule-Walker method (as used in Chapter VI), it is proposed to assess the benefits of using different PSD matching methods. For this, several cost functions are borrowed from the speech coding community, and in the process a significantly faster way to optimize them is shown. Closed and convenient forms for the computation of their gradient vectors and Hessian matrices are given. In the context of speech enhancement, it is eventually found that using other cost functions than the ratio of spectra (i.e. the Yule-Walker method) provides some consistent benefits in terms of output quality, but unfortunately relatively moderate.

In Chapter VIII, we turn to subband treatment and find that it is largely beneficial for several reasons: first, it is argued that the resulting subband algorithms are less computationally intensive than their equivalent fullband counterparts. Next, by stepping in the subband domain, the output quality is significantly improved; and finally several important frequency-domain methods can be used, such are psychoacoustic constraining. A new low-cost heuristic post-processing technique is also presented with consistent benefits. In addition, interesting results are unveiled where the technique of bandwith extension (BWE) is adapted to the

context of speech enhancement: it is principally found that extending the lower bands can provide at least as good of an estimate for the clean speech in high-bands as one would obtain by directly enhancing the noisy higher bands. Moreover, a combination of the two high-bands estimates can then yield a higher quality output when compared to solutions not resorting to BWE.

Gathering several ideas and recommendations from previous Chapters, Chapter IX builds an example of a viable subband-based algorithm with an emphasis on low SNR performance, while attempting to keep the overall computational requirements as low as possible. The designed algorithm comprises several small-order NPFs, a few KEM_{Burg} algorithms, psychoacoustic constraining, post-processing, and BWE-based enhancement. In simulation results, it is found to outperform three well-reviewed state-of-the-art algorithms, especially at very low and low SNRs. These strong results are a clear indication that state-space based algorithms are a viable alternative for speech enhancement.

Finally, two future work ideas are discussed in Chapter X. First, some ideas related to the possibilities for improvement of the speech production model are discussed. Secondly, some binaural extensions to the monaural state-space equations are derived, with some simulation results in WGN environments where non-negligible improvements can be noted. The extension to real-world colored noise has not been implemented, but as shown in Chapter VIII, in the subband domain and with complete discretization of the noise PSD, this extension should be a formality.

References

[AAR'92] R.M. Aarts, "A comparison of some loudness measures for loudspeaker listening tests". Journal of the Audio Engineering Society, vol. 40, no. 3, pp. 142–146, March 1992.

[AUL'84] R. McAulay, "Maximum Likelihood Spectral Estimation and its Application to Narrow-Band Speech Coding," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. 32, No. 2, pp. 243-251, 1984.

[BAH'02] M. Bahoura, J. Rouat, "Wavelet Speech Enhancement based on the Teager Energy Operator," IEEE Signal Processing Letters, vol. 8, no. 1, pp. 10–12, January 2002.

[BEN'05] J. Benesty, S. Makino, and J. Chen, Speech Enhancement, Springer, 2005.

[BOL'79] S. F. Boll, "Suppression of Acoustic Noise in speech using Spectral Subtraction", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27, no. 2, pp 113–120, 1979.

[BUR'87] J. P. Burg, "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, Enschede, The Netherlands, August 1968, reprinted in Modern Spectrum Analysis, D. G. Childers, Ed., IEEE Press, New York, 1987.

[BOY'04] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, 2004.

[BYR'94] D. Byrne, H. Dillon, and K. Tran, "An international comparison of long-term average speech spectra," The Journal of the Acoustical Society of America, vol. 96, pp. 2108–2120, 1994.

[CEX'06] J. C. Cexus and A. O. Boudraa, "Non-stationary signals analysis by Teager-Huang transform (THT)," 14th European Signal Processing Conference, Florence, Italy, September 2006.

[CHA'07] H. Y. Chang, N. K. Soo, and S. Rahardja, "Subband Kalman filtering incorporating masking properties for noisy speech signal," Speech Communication, vol. 49, no. 7, pp. 558–573, July 2007.

[CHE'05] M. Chetouani, A. Hussain, M. Faundez-Zanuy, and B. Gas, "Non-linear predictive models for speech processing," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3697, Warsaw, Poland, September 2005, pp. 779–784.

[CHU'03] Chu, Wai. Speech Coding Algorithms: Foundation and Evolution of Standardized Coders. Wiley-Interscience, 2003.

[CLE'89] R., Clemen, "Combining forecasts: a review and annotated bibliography", International Journal of Forecasting, vol. 5, pp. 559–583, 1989.

[COH'02] I. Cohen, "Noise estimation by minima controlled recursive averaging for robust speech enhancement", IEEE Signal Processing Letters, vol. 9, no. 1, pp. 12–15, 2002.

[DAV'01] C. E. Davila, "On the noise compensated Yule-Walker equations", IEEE Transactions on Signal Processing, vol. 49, no. 6, pp. 1119–1121, 2001.

[DAV'87] C. E. Davila, A. J. Welch, and H. G. III Rylander, "A Kalman Filter algorithm for estimating sinusoids in colored noise," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 1316–1319, Dallas, TX, USA, 1987,

[DEN'06] Y. Deng and J. Mathews, "Subband particle filtering for speech enhancement," in Fourteenth European Signal Processing Conference (EUSIPCO), Florence, Italy, September 2006

[DIE'04] E. J. Diethorn, "Subband Noise Reduction Methods for Speech Enhancement," Ch. 3 in Audio Signal Processing for Next-Generation Multimedia Communication Systems, Y. Huang and J. Benesty, eds., Kluwer Academic Publishers, 2004.

[DOB'95] G. Doblinger, "Computationally efficient speech enhancement by spectral minima tracking in subbands," Proceedings of EUROSPEECH, vol. 2, pp. 1513–1516, 1995.

[DON'94] D.L. Donoho and I.M., Johnstone, "Ideal spatial adaptation by wavelet shrinkage", Biometrika, vol. 81, no. 3, pp. 425–455, 1994.

[DOU'00] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo sampling methods for Bayesian filtering", Statistics and Computing, vol. 3, no. 10, pp 197–208, 2000.

[DOU'01] A. Doucet, J. Freitas, and N. Gordon, Eds., Sequential Monte Carlo methods in practice. New York: Springer-Verlag, 2001.

[EPH'06] Y. Ephraim and I. Cohen, "Recent advancements in speech enhancement," in The Electrical Engineering Handbook, vol. 35. CRC Press, 2006.

[EPH'84] Y. Ephraim and D. Malah. "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator". IEEE Transactions on Acoustics Speech and Signal Processing, ASSP-32(6):1109--1121, 1984.

[EPH'85] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 33, no. 2, pp. 443–445, April 1985.

[EPH'95] Y. Ephraim and H. L. Van Trees. "Signal subspace approach for speech enhancement", IEEE Trans. Speech Signal Processing, vol. 3, pp. 251–266, July 1995.

[FAU'02] M. Faundez-Zanuy, S. McLaughlin, A. Esposito, A. Hussain, J. Schoentgen, G. Kubin, W. B. Kleijn, and P. Maragos, "Nonlinear speech processing: overview and applications," Control and Intelligent Systems, vol. 30, no. 1, pp. 1–10, 2002.

[FON'02] W. Fong, S. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 438–449, February 2002.

[GAB'02] M. Gabrea, "Speech signal recovery in colored noise using an adaptive Kalman filtering," in Canadian Conference on Electrical and Computer Engineering, Winnipeg, Manitoba, vol. 2, pp. 974–979, May 2002.

[GAB'05] M. Gabrea, "An adaptive Kalman Filter for the enhancement of speech signals in colored noise," in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, United States, pp. 45–48, October 2005.

[GAN'03] S. Gannot and M. Moonen, "On the application of the unscented Kalman Filter to Speech processing", International Workshop on Acoustic Echo and Noise Control (IWAENC), pp. 27–30, Kyoto, Japan, September 2003.

[GAN'98] S. Gannot, D. Burshtein, and E. Weinstein, "Iterative and sequential Kalman Filter-based speech enhancement algorithms," IEEE Transactions on Speech and Audio Processing, vol. 6, no. 4, pp. 373–385, 1998.

[GAR'90] Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, NTIS speech disc," NTIS order number PB91-100354, 1990.

[GIB'91] J. D. Gibson, B. Koo, S. D. Gray, "Filtering of Colored Noise for Speech Enhancement and Coding", IEEE Transactions on Signal Processing, vol. 39, pp. 1732-1742, August 1991.

[GOD'04] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," Journal of the American Statistical Association, vol. 99, no. 465, pp. 156–168, March 2004.

[GOH'89] I. Gohberg and I. Koltracht, "Efficient algorithm for Toeplitz plus Hankel matrices," Integral Equations and Operator Theory, vol. 12, no. 1, 1989.

[GOL'00] Goldberg. A Practical Handbook of Speech Codecs. Boca Raton, CRC Press LLC, 2000.

[GRA'04] V. Grancharov, S. Srinivasan, J. Samuelsson, andW. B. Kleijn, "Robust spectrum quantization for LP parameter enhancement," in Proceedings of the 2004 European Signal Processing Conference (EUSIPCO), Vienna, Austria, 2004.

[GRA'76] A. Gray and J. Markel, "Distance measures for Speech Processing," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 24, No. 5, 1976.

[HAN'98] J. Hansen and B. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," In International Conference on Spoken Language Processing, vol. 7, pp. 2819–2822, Sydney, Australia, December 1998.

[HAY'01] S. Haykin, Kalman Filtering and Neural Networks. New York: Wiley, 2001.

[HEI'88] G. Heinig, P. Jankowski, and K. Rost, "Fast inversion algorithms of Toeplitz-plus-Hankel matrices," Numerische Mathematik, vol. 52, no. 6, 1988.

[HIR'95] H. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 153–156, 1995.

[HOO'96] M.J.L. de Hoon, T.H.J.J. van der Hagen, H. Schoonewelle, and H. Dam, "Why Yule-Walker should not be used for autoregressive modelling", Annals of Nuclear Energy, vol. 23, no. 15, pp. 1219–1228, October 1996.

[HU'03] Hu, Y. and Loizou, P. (2003). A generalized subspace approach for enhancing speech corrupted by colored noise. IEEE Trans. on Speech and Audio Processing, 11, 334-341, 2003.

[HU'06] Y. Hu and P. C. Loizou. "Evaluation of objective measures for speech enhancement", International Conference on Spoken Language Processing, Pittsburg, PA, USA, Pittsburg, PA, USA, September 2006.

[HU'06-2] Y. Hu and P. C. Loizou. "Subjective comparison of speech enhancement algorithms", International Conference on Spoken Language Processing, Pittsburg, PA, USA, Pittsburg, PA, USA, September 2006.

[ISO'92] ISO/IEC JTC1/SC29/WG11 MPEG, IS11172-3 "Information Technology – at up to About 1.5 Mbit/s, Part 3: Audio" 1992. ("MPEG-1"). Coding of Moving Pictures and Associated Audio for Digital Storage Media

[ITA'70] F. Itakura and S. Saito, "A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies," Electronics and Communications, Vol. 53, pp. 36-43, 1970.

[ITU'98] "Recommendation p. supplement 23: ITU-T coded-speech database," International Telecommunication Union, CH-Geneva, Tech. Rep., 1998.

[JAB'03] Jabloun, F. and Champagne, B. (2003). Incorporating the human hearing properties in the signal subspace approach for speech enhancement. IEEE Trans. on Speech and Audio Processing, 11(6), 700-708, 2003.

[JAR'91] A. El-Jaroudi and J. Makhoul, "Discrete all-pole modeling," IEEE Transactions on Signal Processing, vol. 39, no. 2, pp. 411–423, February 1991.

[JUL'97] S. Julier, J.K. Uhlmann, "A new extension of the Kalman Filter to nonlinear systems," in International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, Florida, 1997.

[JUN'03] Z. Junhui, K. Jingming, X. Xiang, and H. Shilei, "Noise suppression based on Teager Energy Operator for improving the robustness of ASR front-end," International Workshop on Acoustic Echo and Noise Control, Kyoto, Japan, September 2003.

[KAL'60] R.E. Kalman, "A new approach to linear filtering and prediction problems", Journal of Basic Engineering, vol. 82, Issue 1, pp. 35–45, 1960.

[KAM'02] S. Kamath and P. C. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," IEEE International Conference on Acoustics ,Speech, and Signal Processing, Orlando, 2002.

[KAM'09] A.H. Kamkar-Parsi and M. Bouchard, "Improved noise power spectrum density estimation for binaural hearing aids operating in a diffuse noise field environment", IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, Issue 4, pp. 521–533, May 2009.

[KAM'10] A.H. Kamkar-Parsi, and M. Bouchard, "Binaural Target PSD Estimation and Noise Reduction for Binaural Hearing Aids in Complex Acoustic Environments", submitted for publication in IEEE Transactions on Instrumentation & Measurement.

[KAR'05] R. Karlsson, T. Schon, and F. Gustafsson, "Complexity Analysis of the Marginalized Particle Filter", IEEE Transactions on Signal Processing, vol. 53, no. 11, pp. 4408–4411, 2005.

[KAT'05] J. Kates and K. Arehart, "Coherence and the Speech Intelligibility index," Journal of the Acoustical Society of America, vol. 117, no. 4, pp. 2224–2237, 2005

[KAT'05-2] J. Kates and K. Arehart, "A Model of Speech Intelligibility and Quality in Hearing Aids," IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, October 2005.

[KI'96] Y. L. Ki and S. Katsuhiko, "Efficient recursive estimation for speech enhancement in colored noise," IEEE Signal Processing Letters, vol. 3, no. 7, pp. 196–199, July 1996.

[KLE'02] M. Klein and P. Kabal, "Signal subspace speech enhancement with perceptual post-filtering," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 537–540, Orlando, May 2002.

[LAR'04] E. Larsen and R. M. Aarts. Audio Bandwidth Extension. John Wiley and Sons Ltd, 2004.

[LIZ'94] W. Lizhong and M. Niranjan, "On the design of nonlinear speech predictors with recurrent nets," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 2, Adelaide, Australia, pp. 529–532, April 1994.

[LOI'07] P. C. Loizou, Speech Enhancement, Theory and Practice, 1st ed. CRC Press, Boca Raton, 2007.

[MA'04] N. Ma, M. Bouchard, and R. A. Goubran, "Dual perceptually constrained Unscented Kalman Filter for enhancing speech degraded by colored noise", 7th International Conference on Signal Processing, Beijing, China, vol. 3, pp. 2522—2525, 2004.

[MA'98] N. Ma and G. Wei, "Speech coding with nonlinear local prediction model," in Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, pp. 1101–1104, 1998.

[MAK'75] J. Makhoul, "Linear prediction: A tutorial review," in Proceedings of the IEEE, vol. 63, pp. 561–580, April 1975,

[MAR'87] S. L. Marple, Digital Spectral Analysis with Applications, Englewood Cliffs, NJ, USA, Prentice-Hall, 1987

[MAR'01] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," IEEE Transactions on Speech and Audio Processing, vol. 9, no. 5, pp. 504–512, 2001.

[MAS'00] M. Masek, Y. Attikiouzel, and C.J. deSilva, "Combining data from different algorithms to segment the skin-air interface in mammograms", Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 2. pp. 1195-1198, 2000.

[MER'00] R. van der Merwe, A. Doucet, J. Freitas, and E. Wan, "The unscented particle filter," Cambridge University Engineering Department, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR380, August 2000.

[MIH'07] L. Mihaylova, D. Angelova, S. Honary, D. Bull, C. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," IEEE Transactions on Wireless Communications, vol. 6, no. 10, pp. 3589–3599, October 2007.

[MUS'07] F. Mustiere, M. Bolic, and M. Bouchard, "Quality assessment of speech enhanced using particle filters," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 3, Honolulu, USA, pp. 1197–1200, April 2007.

[MUS'08] F. Mustiere, M. Bolic, and M. Bouchard, "Improved Colored Noise Handling in Kalman Filter-based Speech Enhancement Algorithms", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2008, Niagara Falls, Ontario, May 2008.

[MUS'08-2] F. Mustiere, M. Bouchard, and M. Bolic, "Low-cost modifications of Rao-Blackwellized particle filters for improved speech denoising", Signal Processing, Volume 88, Issue 11, pp. 2678–2692, November 2008.

[MUS'09] F. Mustière, M. Bolic, and M. Bouchard, "Speech Enhancement based on Nonlinear Models using Particle Filters", IEEE Transactions on Neural Networks, vol. 20, no.12, pp.1923 – 1937, Dec. 2009.

[MUS'10] F. Mustière, M. Bouchard, and M. Bolic, "Efficient SNR-based subband post-processing for residual noise reduction in speech enhancement algorithms", Proceedings of 2010 European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark, Aug. 2010

[MUS'10-1] F. Mustière, M. Bouchard, and M. Bolic, "Real-world Particle Filtering-based Speech Enhancement", Proceedings of 2nd International Workshop on Cognitive Information Processing (CIP) 2010, Elba Island, Italy, June 2010

[MUS'10-2] F. Mustière, M. Bouchard, and M. Bolic, "Bandwidth Extension for Speech Enhancement", Proceedings of 23rd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2010, Calgary, Canada, May 2010

[NEL'97] A. T. Nelson and E. A. Wan, "Neural speech enhancement using dual extended Kalman filtering," in Proceedings of International Conference on Neural Networks (ICNN'97), Houston, TX, USA, pp. 2171–2175, 1997.

[NGU'94] T. Q. Nguyen, "Near-Perfect-Reconstruction Pseudo-QMF Banks," IEEE Transactions on Signal Processing, vol. 42, no. 1, Januray 1994.

[ODO'06] J.-M. Odobez, D. Gatica-Perez, and S. Ba, "Embedding motion in model-based stochastic tracking," IEEE Transactions on Image Processing, vol. 15, no. 11, pp. 3514–3530, 2006.

[PAL'87] K. K. Paliwal and A. Basu, "A speech enhancement method based on. Kalman filtering," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 177–180, 1987.

[PAR'06] S. Park and S. Choi, "Rao-Blackwellized particle filtering for sequential speech enhancement," in International Joint Conference on Neural Networks (IJCNN), pp. 1254–1259, 2006.

[POL'96] P. Pollak, "Robust LPC parametrization based on noise correlation compensation", Ecole Nationale des Telecommunications, Research Report of Sabbatical Stay, 1996.

[PRI'83] M. B. Priestley, Spectral Analysis and Time Series, Academic Press, London, 1983.

[QIA'03] Y. Qian and P. Kabal, "Dual-Mode Wideband Speech Recovery from Narrowband Speech", Proceedings of the 8th European Conference on Speech Communication and Technology, pp. 1433–1437, September 2003.

[QIA'05] Y. Qian and P. Kabal, "Classified Highband Excitation for Bandwidth Extension of Telephony Signals", Proceedings of the European Signal Processing Conference (Antalya, Turkey), September 2005.

[RAN'06] S. Rangachari and P.C. Loizou, "A noise-estimation algorithm for highly non-stationary environments", Speech Communication, vol. 48, no. 2, pp. 220–231, 2006.

[RIS'04] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman filter: Particle filters for tracking applications. London: Artech House, February 2004.

[SHU'05] W. Shu and Z. Zheng, "Neural dual particle filter and its application in speech enhancement," IEEE Workshop on Signal Processing Systems Design and Implementation, pp. 451–454, November 2005.

[SOR'05] K. Sorensen and S. Andersen, "Speech enhancement with natural sounding residual noise based on connected time-frequency speech presence regions", EURASIP Journal of Applied Signal Processing, vol. 18, pp. 2954–2964, 2005.

[SPA'07] A. Spanias. Audio Signal Processing and Coding. Wiley-Interscience, 2007.

[THY'94] J. Thyssen, H. Nielsen, and S. D. Hansen, "Non-linear short-term prediction in speech coding," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Adelaide, SA, Australia, pp. 185–188, 1994.

[VAR'92] A. Varga, H.J.M. Steeneken, M. Tomlinson, and D. Jones, "The NOISEX-92 study on the effect of additive noise on automatic speech recognition", Technical report, Defence Evaluation and Research Agency (DERA), Speech Research Unit, Malvern, United Kingdom, 1992.

[VER'02] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 3, pp. 173–185, March 2002.

[VIR'99] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system". IEEE Trans. Speech Audio Process. 7 (3), 126–137, 1999.

[WAN'98] E. A. Wan and A. T. Nelson, "Removal of noise from speech using the dual EKF algorithm," in International Conference on Acoustics, Speech and Signal Processing, vol. 1, Seattle, WA, USA, pp. 381–384, May 1998.

[WAN'93] F-M. Wang, P. Kabal, R. Ramachandran, D. O'Shaugnessy, "Frequency domain adaptive postfiltering for enhancement of noisy speech," Speech Communication, 12, no. 1, 41–56, February 1993.

[WEI'00] B. Wei and J. Gibson, "Comparison of distance measures in discrete spectral modeling", Ninth Digital Signal Processing Workshop, Hunt, Texas, October 2000.

[WEI'03] B. Wei and J. Gibson, "A new discrete spectral modeling method and an application to CELP coding", IEEE Signal processing letters, vol. 10, pp. 101–103, 2003.

[WEI'00-2] E. Weinstein, A.V. Oppenheim, and M. Feder, "Signal Enhancement using single and multi-sensor measurements," Technical Report no. 560, M.I.T., Cambridge, MA, November 1990.

[WEI'94] E. Weinstein, A.V. Oppenheim, M. Feder, and J.R. Bock, "Iterative and sequential algorithms for multisensor signal enhancement," IEEE Trans. Signal Processing, vol. 42, pp. 846–859, April 1994.

[WIN'06] S. Windmann and R. Haeb-Umbach, "Iterative speech enhancement using a non-linear dynamic state model of speech and its parameters," in 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, pp. 465–468, 2006.

[ZAV'06] E. Zavarehei, S. Vaseghi, and Q. Yan, "Inter-frame modeling of DFT trajectories of speech and noise for speech enhancement using Kalman Filters," Speech Communication, vol. 48, no. 11, pp. 1545–1555, November 2006.

[ZAV'07] E. Zavarehei, S. Vaseghi, and Q. Yan, "Noisy Speech Enhancement Using Harmonic-Noise Model and Codebook-Based Post-Processing," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 4, May 2007.

Appendix A: Some algorithms and code

Some of the algorithms repeatedly used and referred to in the thesis are listed below.

A.1. Algorithm 1: Kalman Filter iteration

Algorithm 1: Kalman Filter iteration

```
function [x,P,weight] = KF(x,P,z,A,B,C,D,G,H,u);
% function [x,P] = KF(x,P,z,A,B,C,D,G,H,u);
%
% One entire step for a Kalman filter with state-space model:
% x = Ax + Bu + Gw
% z = Cx + Du + Hv
% w,v are considered to be zero-mean unit covariance, P is the covariance matrix of x

Pt = A*P*A' + G*G';
T = C*Pt*C' + H*H'; Tinv = inv(T);
xt = A*x+B*u;
y = C*xt + D*u;
logweight = - 0.5*log(det(2*pi*T)) - 0.5*(z-y)'*Tinv*(z-y);
weight = exp(logweight);
J = Pt*C'*Tinv;
P = (eye(size(P,1)) - J*C)*Pt;
x = xt + J*(z-y);
```

A.2. Algorithm 2: Extended Kalman Filter iteration

Algorithm 2: Extended Kalman Filter iteration

```
function [x,P] = EKF(x,P,z,f,h,A,B,C,D,G,H,u);
% function [x,P] = EKF(x,P,z,f,h,A,B,C,D,G,H,u);
%
% One entire step for an EKF with state-space model:
% x = f(x) + Bu + Gw
% z = h(x) + Du + Hv
% w,v are considered to be zero-mean unit covariance, P is the covariance matrix of x
% A is the Jacobian matrix of f at the input x
% C is the Jacobian matrix of h at f(x)+B*u, where x is the input value

Pt = A*P*A' + G*G';
T = C*Pt*C' + H*H';
J = Pt*C'*inv(T);
P = (eye(size(P,1)) - J*C)*Pt;
xt = f(x) + B*u;
y = h(xt) + D*u;
x = xt + J*(z-y);
```

A.3. Algorithm 3: Unscented Kalman Filter iteration

Algorithm 3: Unscented Kalman Filter iteration

```
function [x,P] = ukf(x,P,z,f,h,B,D,G,H,u);
% function [x,P] = ukf(x,P,z,f,h,B,D,G,H,u);
%
% One entire step for a UKF with state-space model:
% x = f(x) + Bu + Gw
% z = h(x) + Du + Hv
% w,v are considered to be zero-mean unit covariance, P is the covariance matrix of x

% Dimensions
M = length(x); size_w = length(G(:,1)); size_v = length(H(:,1));

% Augment x and P into x_a and P_a
x_a = [x(:);zeros(size_w+size_v,1)]; P_a = blkdiag(P,G,H);

% Obtain N sigma points with weights W
[x_sp, W, N] = sigma_points(x_a, P_a);
       % i^th sigma point is x_sp(:,i), with weight W(i)
       % NOTE: First weight of covariance matrices is in W(N+1)

% Obtain predicted sigma points (state and measurement)
x_sp_pred = zeros(M,N); z_sp_pred = zeros(size_v,N);
for i = 1:N
       x_sp_pred(:,i) = f(x_sp(1:M,i)) + B*u + x_sp(M+1:M+size_w,i);
       z_sp_pred(:,i) = h(x_sp_pred(:,i)) + D*u + x_sp(M+size_w+1:length(x_a),i);
end

% Obtain their mean and covariance
% *     Means
x_pred = zeros(M,1); z_pred = zeros(size_v,1);
for i = 1:N
       x_pred = x_pred + W(i)*x_sp_pred(:,i);
       z_pred = z_pred + W(i)*z_sp_pred(:,i);
end
% *     Covariances, with first term weighted distinctly (its weight is in W(N+1))
z_diff = z_sp_pred(:,1)-z_pred; x_diff = x_sp_pred(:,1)-x_pred;
Pzz = W(N+1)*(z_diff*z_diff');
Pxx = W(N+1)*(x_diff*x_diff');
Pxz = W(N+1)*(x_diff*z_diff');
for i = 2:N
       z_diff = z_sp_pred(:,i)-z_pred;
       x_diff = x_sp_pred(:,i)-x_pred;
       Pzz = Pzz + W(i)*(z_diff*z_diff');
       Pxx = Pxx + W(i)*(x_diff*x_diff');
       Pxz = Pxz + W(i)*(x_diff*z_diff');
end

% Finally, update x and P
K  = Pxz*inv(Pzz);
x = x_pred + K*(z-z_pred);
P = Pxx - K*Pzz*K';
```

A.4. Algorithm 4: Particle Filter algorithm

Algorithm 4: Pseudo-code for a generic Particle Filter

Initialization:

- Draw an initial set of N particles $\left\{\mathbf{x}_{k=0}^{(i)}\right\}_{i=1}^{N}$ according to a priori belief.

Main iteration:

- For every incoming measurement $\mathbf{z}_k$, do the following:
  - For every particle, i.e., every $i, 1 \le i \le N$:
    - Draw $\mathbf{x}_k^{(i)} \sim q\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k\right)$
    - Compute the unnormalized weights:

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^{(i)}, \mathbf{z}_{k-1}) p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}$$

  - Normalize the weights and resample the particles according to the weights
  - Obtain any desired state estimate from $p(\mathbf{x}_k \mid \mathbf{z}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right)$

A.5. Algorithm 5: RBPF algorithm

Algorithm 5: RBPF algorithm for conditionally linear-Gaussian systems

Initialization:

- Draw an initial set of N particles $\left\{\mathbf{x}_{1k=0}^{(i)}; \mathbf{x}_{2k=0}^{(i)}; \mathbf{P}_{k=0}^{(i)}\right\}_{i=1}^{N}$ according to a priori belief.

Main iteration:

- For every incoming measurement $\mathbf{z}_k$, update the set of particles following:
  - For every particle, i.e., every $i, 1 \le i \le N$:
    - Draw $\mathbf{x}_{1k}^{(i)} \sim q\left(\mathbf{x}_{1k} \mid \mathbf{x}_{1k-1}^{(i)}, \mathbf{z}_k\right)$
    - From $\mathbf{x}_{1k}^{(i)}$, obtain $\mathbf{A}_k^{(i)}$, $\mathbf{B}_k^{(i)}$, $\mathbf{C}_k^{(i)}$, $\mathbf{D}_k^{(i)}$, $\mathbf{G}_k^{(i)}$, $\mathbf{H}_k^{(i)}$, and $\mathbf{u}_k^{(i)}$
    - Use the Kalman Filter (Algorithm 1 previously introduced) as:

$[\mathbf{x}_{2k}^{(i)}, \mathbf{P}_k^{(i)}, \text{tmp}] = \text{KF}(\mathbf{x}_{2k-1}^{(i)}, \mathbf{P}_{k-1}^{(i)}, \mathbf{z}_k, \mathbf{A}_k^{(i)}, \mathbf{B}_k^{(i)}, \mathbf{C}_k^{(i)}, \mathbf{D}_k^{(i)}, \mathbf{G}_k^{(i)}, \mathbf{H}_k^{(i)}, \mathbf{u}_k^{(i)})$;

    - Compute the unnormalized weights:

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{\text{tmp} \times p(\mathbf{x}_{1k}^{(i)} \mid \mathbf{x}_{1k-1}^{(i)})}{q(\mathbf{x}_{1k}^{(i)} \mid \mathbf{x}_{1k-1}^{(i)}, \mathbf{z}_{1k})}$$

  - Normalize the weights and resample the particles according to the weights
  - Obtain any desired state estimate from $p(\mathbf{x}_k \mid \mathbf{z}_{0k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right)$

A.6. Algorithm 6: Neural Particle Filter for speech enhancement

Algorithm 6: Standard neural PF for speech enhancement

Draw an initial set of N particles $\{\chi_{k=0}^{(i)}\}_{i=1}^{N}$ according to a priori belief.

For every incoming measurement $\mathbf{z}_k$, do the following:

- For every particle, i.e., every $i, 1 \le i \le N$:
  - Draw $\mathbf{W}(k)^{(i)}$, $\mathbf{c}(k)^{(i)}$, $\mathbf{d}(k)^{(i)}$, $L_x(k)^{(i)}$, and $L_v(k)^{(i)}$ (if necessary) according to a pre-defined Gaussian random walk.
  - Compute the following:

$$r(k) = \mathbf{c}(k)^T f(\mathbf{W}(k)\mathbf{x}(k-1) + \mathbf{d}(k))$$

$$s(k) = \mathbf{b}(k)^T (\mathbf{z}(k-1) - \tilde{\mathbf{x}}(k-1))$$

$$\theta(k)^{(i)} = \left[\left(\sigma_v(k)^{(i)}\right)^{-2} + \left(\sigma_g(k)^{(i)}\right)^{-2}\right]^{-1/2}$$

$$\alpha(k)^{(i)} = \theta(k)^{(i)2}\left[(z(k) - s(k)^{(i)})\left(\sigma_v(k)^{(i)}\right)^{-2} + r(k)^{(i)}\left(\sigma_g(k)^{(i)}\right)^{-2}\right]$$

  - Draw $x(k)^{(i)} \sim N(\alpha(k)^{(i)}; \theta(k)^{(i)2})$
- Compute the unnormalized weights:

$$\tilde{w}(k)^{(i)} = N(z(k) \mid r(k)^{(i)} + s(k)^{(i)}; \left(\sigma_v(k)^{(i)}\right)^2 + \left(\sigma_g(k)^{(i)}\right)^2)$$

- Normalize the weights and resample the particles

A.7. Algorithm 7: Dual Neural Particle Filter for speech enhancement

Algorithm 7: Dual neural PF for speech enhancement

Initialize PF-*A*: Draw N_A particles $\{\mathbf{x}(0)^{(i)}\}_{i=1}^{N_A}$.

Initialize PF-*B*: Draw N_B particles $\{\mathbf{W}(0)^{(i)};\mathbf{d}(0)^{(i)};L_x(0)^{(i)};L_v(0)^{(i)}\}_{i=1}^{N_B}$ as well as $\{\mathbf{c}(0)^{(i)};\mathbf{K}(0)^{(i)}\}_{i=1}^{N_B}$

For every incoming measurement $\mathbf{z}_k$, do the following:

Update parameters via PF-*B*:

- Define $s(k) = \mathbf{b}(k)^T(\mathbf{z}(k-1) - \hat{\tilde{\mathbf{x}}}(k-1))$
- For every $i, 1 \le i \le N_B$:
  - Draw $\mathbf{W}(k)^{(i)}$, $\mathbf{d}(k)^{(i)}$, $L_x(k)^{(i)}$, and $L_v(k)^{(i)}$ (if necessary) according to a pre-defined Gaussian random walk.
  - Let $\mathbf{C}(k)^{(i)} = f(\mathbf{W}(k)^{(i)}\hat{\mathbf{x}}(k-1) + \mathbf{d}(k)^{(i)})^T$, $\mathbf{H}(k)^{(i)} = \sqrt{(\sigma_v(k)^{(i)})^2 + (\sigma_g(k)^{(i)})^2}$
  - Use the Kalman Filter (Algorithm 1) as:

$[\mathbf{c}(k)^{(i)}, \mathbf{K}(k)^{(i)}, \tilde{w}_B(k)^{(i)}]$=KF($\mathbf{c}(k-1)^{(i)}, \mathbf{K}(k-1)^{(i)}, \mathbf{z}(k-1), I_P, 0, \mathbf{C}(k)^{(i)}, 0,$
$\sigma_c I_P, \mathbf{H}(k)^{(i)}, s(k)$)

- Normalize the weights, resample and obtain the estimates $\hat{\mathbf{W}}(k)$, $\hat{\mathbf{c}}(k)$, $\hat{\mathbf{d}}(k)$, $\hat{L}_x(k)$ and possibly $\hat{L}_v(k)$ to be made available to PF-*A*.

Update parameters via PF-*A*:

- Compute $\hat{\theta}(k) = (\hat{\sigma}_v(k)^{-2} + \hat{\sigma}_g(k)^{-2})^{-1/2}$
- For every particle, i.e., every $i, 1 \le i \le N_A$:
  - Compute the following:

$$\hat{r}(k)^{(i)} = \hat{\mathbf{c}}(k)^T f(\hat{\mathbf{W}}(k)\mathbf{x}(k-1)^{(i)} + \hat{\mathbf{d}}(k))$$
$$\hat{s}(k)^{(i)} = \mathbf{b}(k)^T(\mathbf{z}(k-1) - \tilde{\mathbf{x}}(k-1)^{(i)})$$
$$\hat{\alpha}(k)^{(i)} = \hat{\theta}(k)^2\left[(z(k) - \hat{s}(k)^{(i)})(\hat{\sigma}_v(k))^{-2} + \hat{r}(k)^{(i)}(\hat{\sigma}_g(k))^{-2}\right]$$

  - Draw $x(k)^{(i)} \sim N(\alpha(k)^{(i)}; \theta(k)^{(i)2})$
  - Compute the unnormalized weights:

$$\tilde{w}_A(k)^{(i)} = N(z(k) \mid \hat{r}(k)^{(i)} + \hat{s}(k)^{(i)}; \hat{\sigma}_v(k)^2 + \hat{\sigma}_g(k)^2)$$

- Normalize the weights, resample the particles, and obtain the estimate $\hat{\mathbf{x}}(k)$ to be made available to PF-*B*.

A.8. Improved resampling scheme

Besides the well-known problems of degeneracy and sample impoverishment, it may occur that PFs, at some iteration, entirely fail to "locate" good candidates for the state, resulting in a set of particles with only negligible weights. This problem is more likely to occur in highly complex situations and/or when the importance density is inappropriate. As an example, it is simple to envision this problem occurring in a tracking application where the tracked object suddenly moves in an unforeseen or unlikely manner. In the algorithms of this thesis, even though the weight computation is based on the evaluation of a Gaussian density function, which is never equal to 0, the actual set of weights may underflow and practically collapse to zero.

To circumvent this problem, first the logarithm of the weights can be used, ensuring a wider range of numerically representable weights. Before the resampling, the logweights can be either directly "normalized," or scaled to a range of values for which the application of the exponential function will not result in numerical underflows. The two equivalent methods are shown as follows.

- **Alternative 1: Normalization in the log domain**

  - For $i=1$ to N, during the PF iterations compute the unnormaliized log-weights $\tilde{l}_k^{(i)} = \log\left(\tilde{w}_k^{(i)}\right)$, and update $L_i = \log\left(\sum_{j=1}^{i} \tilde{w}_k^{(j)}\right)$ by using the formula $L_i = L_{i-1} + \log\left(1 + \exp\left[\tilde{l}_k^{(i)} - L_{i-1}\right]\right)$.
  - Then, for all i, do $l_k^{(i)} = \tilde{l}_k^{(i)} - L_N$, thereby normalizing the log-weights.

- The weights, now "brought back" to a numerically representable range of values, can be reobtained by applying the exponential function, and regular resampling can take place

- **Alternative 2: Prescaling and normalization**
  - During the update of each particle, keep track of the maximal unnormalized log-weight obtained (denote it by $\tilde{l}_k^{(j)}$, assuming it occurs at index j, $1 \leq j \leq N$).
  - Then, for all i, replace $\tilde{l}_k^{(i)}$ by $\tilde{l}_k^{(i)} - \tilde{l}_k^{(j)}$, which does *not* normalize the log-weights, but scales them to a range of values such that it is certain that at least one weight will be nonzero.
  - The unnormalized weights can be reobtained by applying the exponential function, and both regular normalizing and resampling can take place.

The above is a "numerical safety net," but does not prevent the particle cloud to become, at a certain point, unhealthy or ill-conditioned, with extremely bad state candidates being propagated iteration after iteration. The following simple trick was used in the neural-based PF algorithms of this thesis in order to prevent the above problem from occurring. In very simple terms, it consists of redrawing a given sample when its weight is found to be too small. The redraw, however, only takes place when deemed necessary, that is, when the current effective sample size is also too small. In addition, if after a certain amount of trials no viable sample is found, a radical resetting is performed and more trials are conducted.

In the large number of experiments conducted in this thesis, it was found that this method is helpful to maintain a high level of robustness, i.e., a certain subset of significant particles in

the following sense. Before its implementation, we observed that in a few dozen cases, among the thousands observed, at some isolated instants, the particle cloud would entirely disperse towards insignificant regions in the state-space. Typically, when this happened, the algorithm would go on and eventually recover, but not without missing a few milliseconds of speech, and possibly "replacing" it with very perceivable artefacts. After the implementation of the algorithm, for the majority of enhanced segments, the results are equivalent, except at the aforementioned isolated instants where they are significantly superior, thanks to the artificial persistence introduced by the method.

The "modified" resulting PF algorithm is shown in Algorithm 8, where Alternative 1 has been used. In the neural-based algorithms used in this paper, the thresholds used are $T_1 = 40$, $T_2 = 20$, $K_1 = -20$, and $K_2 = \log(N/100)$, and the unnormalized log-weights corresponding to a density $N(x|\mu,\sigma^2)$ are obtained as $\left(-\log\sigma - (x-\mu)^2/(2\sigma^2)\right)$. Roughly speaking, this choice of T_1 and T_2 means "try 40 times in total, but at the twentieth time reinitialize some elements." For the dual algorithms, Algorithm 6 is only embedded in PF-*B*, simply because the particle weights in PF-*A* do not depend on the actual particle drawn. As a final but important detail, instead of resetting the entire vector $\mathbf{x}_{k-1}^{(i)}$ according to $p(\mathbf{x}_0)$ (*a priori* belief in the generic PF Algorithm 5), we only reset the speech model parameters $\hat{\mathbf{W}}(k)$, $\hat{\mathbf{c}}(k)$, $\hat{\mathbf{d}}(k)$, $\hat{L}_x(k)$. If we were to reset the entire vector, some past speech samples would be affected and significant artefacts would be audible.

Algorithm 8: Generic PF with enhanced robustness

Define integer thresholds $T_2 < T_1$, real thresholds K_1, K_2, and initialize variable `count` **to 0 and declare variable L to any value.**

Initialization:

- Draw an initial set of N particles $\{\mathbf{x}_{k=0}^{(i)}\}_{i=1}^{N}$ according to a priori belief $p(\mathbf{x}_0)$.

Main iteration:

- For every incoming measurement $\mathbf{z}_k$, do the following:
  - For every particle, i.e., every $i, 1 \le i \le N$:
    - If $L < K_2$ or $i = 1$
      - Set `count` = 0
      - While `count` $< T_1$

        If `count` = T_2, draw $\mathbf{x}_{k-1}^{(i)} \sim p(\mathbf{x}_0)$ [9]

        Draw $\mathbf{x}_k^{(i)} \sim q\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k\right)$

        Compute $\tilde{l}_k^{(i)} = \log\left(\tilde{w}_k^{(i)}\right)$

        If $\tilde{l}_k^{(i)} > K_1$ or `count` = $T_1 - 1$

        If $i = 1$, let $L = \tilde{l}_k^{(i)}$

        If $i > 1$, replace L with $L + \log\left(1 + \exp\left[\tilde{l}_k^{(i)} - L\right]\right)$

        Set `count` = T_1

        Increment `count`
    - Else
      - Draw $\mathbf{x}_k^{(i)} \sim q\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k\right)$
      - Compute $\tilde{l}_k^{(i)} = \log\left(\tilde{w}_k^{(i)}\right)$
      - Replace L with $L + \log\left(1 + \exp\left[\tilde{l}_k^{(i)} - L\right]\right)$
  - Obtain normalized weights with $w_k^{(i)} = \exp(\tilde{l}_k^{(i)} - L)$
  - Resample the particles according to the normalized weights

[9] Or, only reinitialize some elements of $\mathbf{x}_{k-1}^{(i)}$ and not the whole vector

Appendix B: Full tables of simulation results

In this Appendix, the various detailed tables of results and their corresponding sections are shown. Due to the large amount of results, it was anticipated that leaving those tables along the text would have been a threat to clarity, while breaking the flow of the thesis. Instead, in the text only a summary of findings is given, and a reference to the appropriate table in this chapter.

B.1. Tables for Section 3.2.5 (Performance in WGN)

B.1.1. Dual Algorithms

VL-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-4.1	-7.5	0.1	1.0	0.2	1.2	0.5
DKF	5.7	-1.7	0.2	1.1	0.2	1.5	0.6
DEKF(1)	5.8	-1.7	0.2	1.1	0.2	1.5	0.5
DUKF(1)	6.3	-1.6	0.3	1.1	0.4	1.6	0.7
DEKF(4)	6.1	-1.6	0.3	1.1	0.3	1.6	0.6
DUKF(4)	6.0	-1.7	0.3	1.1	0.3	1.6	0.6
DKF_f	5.8	-1.7	0.2	1.1	0.2	1.5	0.6
$DEKF_f(1)$	5.8	-1.7	0.2	1.1	0.2	1.5	0.6
$DUKF_f(1)$	6.2	-1.6	0.3	1.1	0.4	1.6	0.7
$DEKF_f(4)$	6.1	-1.6	0.4	1.1	0.3	1.6	0.6
$DUKF_f(4)$	5.6	-1.9	0.3	1.1	0.3	1.5	0.6

Table 36: Performance of dual state-space based algorithms in VL-WGN conditions

L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.1	-5.6	0.2	1.0	0.6	1.5	0.8
DKF	8.2	-0.2	0.5	1.1	0.6	1.8	0.8
DEKF(1)	8.3	-0.2	0.5	1.1	0.6	1.7	0.8
DUKF(1)	9.0	0.0	0.6	1.1	0.8	1.8	1.0
DEKF(4)	8.6	-0.1	0.6	1.1	0.7	1.8	0.9
DUKF(4)	8.7	-0.1	0.6	1.1	0.8	1.8	1.0
DKF_f	8.3	-0.2	0.5	1.1	0.6	1.7	0.8
$DEKF_f(1)$	8.4	-0.2	0.5	1.1	0.6	1.7	0.8
$DUKF_f(1)$	9.1	0.1	0.7	1.1	0.9	1.8	1.0
$DEKF_f(4)$	8.8	0.0	0.6	1.1	0.7	1.8	0.9
$DUKF_f(4)$	8.2	-0.4	0.6	1.1	0.8	1.8	0.9

Table 37: Performance of dual state-space based algorithms in L-WGN conditions

M-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.1	-2.1	0.6	1.1	1.2	2.0	1.3
DKF	12.1	2.3	0.9	1.2	1.2	2.2	1.3
DEKF(1)	12.0	2.3	0.9	1.2	1.2	2.2	1.3
DUKF(1)	12.9	2.6	0.9	1.2	1.4	2.3	1.5
DEKF(4)	12.8	2.6	0.9	1.3	1.4	2.3	1.5
DUKF(4)	13.2	2.8	0.9	1.3	1.5	2.3	1.6
DKF_f	12.1	2.3	0.9	1.2	1.2	2.2	1.3
$DEKF_f(1)$	12.1	2.3	0.9	1.2	1.2	2.2	1.3
$DUKF_f(1)$	13.4	2.9	0.9	1.3	1.6	2.4	1.7
$DEKF_f(4)$	12.9	2.7	0.9	1.3	1.4	2.3	1.5
$DUKF_f(4)$	12.9	2.6	0.9	1.3	1.5	2.3	1.5

Table 38: Performance of dual state-space based algorithms in M-WGN conditions

H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	12.1	1.8	0.9	1.2	1.9	2.6	2.0
DKF	16.5	5.5	1.0	1.3	2.0	2.8	2.0
DEKF(1)	16.4	5.5	1.0	1.4	2.0	2.8	2.0
DUKF(1)	17.2	5.8	1.0	1.4	2.1	2.9	2.2
DEKF(4)	17.0	5.8	1.0	1.4	2.0	2.8	2.1
DUKF(4)	17.5	6.0	1.0	1.4	2.1	2.9	2.2
DKF_f	16.6	5.5	1.0	1.4	2.0	2.8	2.0
$DEKF_f(1)$	16.6	5.5	1.0	1.4	2.0	2.8	2.0
$DUKF_f(1)$	18.1	6.4	1.0	1.5	2.2	2.9	2.3
$DEKF_f(4)$	17.3	5.9	1.0	1.4	2.0	2.8	2.1
$DUKF_f(4)$	17.5	6.0	1.0	1.5	2.2	2.9	2.2

Table 39: Performance of dual state-space based algorithms in H-WGN conditions

B.1.2. Optimization-based Algorithms

VL-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-4.1	-7.5	0.1	1.0	0.2	1.2	0.5
KGD	2.9	-2.7	0.3	1.0	0.1	1.5	0.5
KEM	5.7	-2.4	0.3	1.1	0.6	1.6	0.8
KEM_{Burg}	6.2	-2.0	0.4	1.1	0.5	1.6	0.8
KEM_{Cov}	6.2	-2.0	0.4	1.1	0.5	1.6	0.8
KEM_{Mcov}	6.2	-2.0	0.4	1.1	0.5	1.6	0.8

Table 40: Performance of optimization based algorithms in VL-WGN conditions

L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.1	-5.6	0.2	1.0	0.6	1.5	0.8
KGD	5.4	-1.5	0.6	1.1	0.6	1.8	0.9
KEM	8.2	-1.0	0.6	1.1	0.9	1.9	1.1
KEM_{Burg}	8.7	-0.5	0.7	1.1	0.9	1.9	1.1
KEM_{Cov}	8.7	-0.5	0.7	1.1	0.9	1.9	1.1
KEM_{Mcov}	8.7	-0.5	0.7	1.1	0.9	1.9	1.1

Table 41: Performance of optimization based algorithms in L-WGN conditions

M-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.1	-2.1	0.6	1.1	1.2	2.0	1.3
KGD	9.6	0.7	0.9	1.2	1.3	2.2	1.5
KEM	12.1	1.7	0.9	1.3	1.5	2.3	1.6
KEM_{Burg}	12.4	2.1	0.9	1.3	1.5	2.3	1.6
KEM_{Cov}	12.4	2.1	0.9	1.3	1.5	2.3	1.6
KEM_{Mcov}	12.4	2.1	0.9	1.3	1.5	2.3	1.6

Table 42: Performance of optimization based algorithms in M-WGN conditions

H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	12.1	1.8	0.9	1.2	1.9	2.6	2.0
KGD	14.5	4.0	1.0	1.4	2.1	2.7	2.1
KEM	16.4	4.8	1.0	1.6	2.2	2.9	2.2
KEM_{Burg}	16.7	5.1	1.0	1.6	2.1	2.9	2.2
KEM_{Cov}	16.7	5.1	1.0	1.6	2.1	2.9	2.2
KEM_{Mcov}	16.7	5.2	1.0	1.6	2.1	2.8	2.2

Table 43: Performance of optimization based algorithms in H-WGN conditions

B.1.3. Holistic Algorithms

VL-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-4.1	-7.5	0.1	1.0	0.2	1.2	0.5
$JUKF_{Lin}$	6.7	-1.4	0.3	1.1	0.5	1.6	0.7
$JUKF_{Nonlin}(1)$	6.5	-1.5	0.3	1.1	0.5	1.6	0.7
$JUKF_{Nonlin}(4)$	6.3	-1.6	0.3	1.1	0.6	1.6	0.8
RBPF	6.8	-1.3	0.4	1.1	0.4	1.6	0.6

Table 44: Performance of holistic algorithms in VL-WGN conditions

L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.1	-5.6	0.2	1.0	0.6	1.5	0.8
$JUKF_{Lin}$	9.2	0.1	0.7	1.1	0.9	1.8	1.1
$JUKF_{Nonlin}(1)$	9.0	0.0	0.6	1.1	1.0	1.8	1.1
$JUKF_{Nonlin}(4)$	8.4	-0.4	0.6	1.1	0.9	1.8	1.0
RBPF	9.5	0.4	0.7	1.1	0.8	1.8	1.0

Table 45: Performance of holistic algorithms in L-WGN conditions

M-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.1	-2.1	0.6	1.1	1.2	2.0	1.3
$JUKF_{Lin}$	13.0	2.7	0.9	1.2	1.7	2.3	1.7
$JUKF_{Nonlin}(1)$	12.8	2.6	0.9	1.2	1.6	2.3	1.7
$JUKF_{Nonlin}(4)$	12.4	2.2	0.9	1.2	1.6	2.3	1.6
RBPF	13.7	3.2	0.9	1.3	1.5	2.3	1.5

Table 46: Performance of holistic algorithms in M-WGN conditions

H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	12.1	1.8	0.9	1.2	1.9	2.6	2.0
$JUKF_{Lin}$	17.2	5.8	1.0	1.5	2.4	2.9	2.4
$JUKF_{Nonlin}(1)$	17.0	5.7	1.0	1.5	2.4	2.9	2.3
$JUKF_{Nonlin}(4)$	17.0	5.6	1.0	1.5	2.3	2.9	2.3
RBPF	18.4	6.7	1.0	1.9	2.2	2.9	2.2

Table 47: Performance of holistic algorithms in H-WGN conditions

B.2. Tables for Section 4.2 (Combination of estimates)

Type of algorithm	*Quality measure*	*Noise*			
		VL-WGN	L-WGN	M-WGN	H-WGN
Noisy speech	SNR	-4.1	0.1	6.1	12.1
(1) RBPF	SNR	6.8	9.5	13.7	18.4
	ASNR	-1.3	0.4	3.2	6.7
	CSII	0.4	0.7	0.9	1.0
	WPESQ	1.1	1.1	1.3	1.9
	Csig	0.4	0.8	1.5	2.2
	Cbak	1.6	1.8	2.3	2.9
	Covl	0.6	1.0	1.5	2.2
(2) WPT	SNR	6.3	8.5	11.9	15.6
	ASNR	-2.2	-0.7	1.8	4.6
	CSII	0.3	0.6	0.9	1.0
	WPESQ	1.1	1.2	1.4	1.9
	Csig	0.1	0.6	1.3	2.1
	Cbak	1.4	1.7	2.2	2.8
	Covl	0.6	1.0	1.6	2.3
(3) SPECSUB	SNR	6.9	8.6	10.9	12.4
	ASNR	-1.6	-0.3	1.6	3.2
	CSII	0.6	0.8	0.9	1.0
	WPESQ	1.1	1.1	1.2	1.4
	Csig	-1.4	-0.9	-0.2	0.4
	Cbak	0.5	0.8	1.3	1.7
	Covl	-0.5	-0.2	0.4	0.8
(4) DKF	SNR	5.7	8.2	12.1	16.5
	ASNR	-1.7	-0.2	2.3	5.5
	CSII	0.2	0.5	0.9	1.0
	WPESQ	1.1	1.1	1.2	1.3
	Csig	0.2	0.6	1.2	2.0
	Cbak	1.5	1.8	2.2	2.8
	Covl	0.6	0.8	1.3	2.0
{(1)+(2)}	SNR	7.7	10.3	14.2	18.7
	ASNR	-0.6	1.0	3.8	7.2
	CSII	0.5	0.8	1.0	1.0
	WPESQ	1.1	1.3	1.6	2.2
	Csig	0.7	1.2	1.8	2.5
	Cbak	1.7	2.0	2.5	3.1
	Covl	0.9	1.3	1.9	2.5
{(1)+(3)}	SNR	8.0	10.3	13.7	16.7
	ASNR	-0.7	0.9	3.3	5.9
	CSII	0.6	0.9	1.0	1.0
	WPESQ	1.1	1.3	1.5	2.1
	Csig	0.5	1.1	1.3	2.3
	Cbak	1.9	1.9	2.1	2.9
	Covl	0.5	1.3	1.4	2.2
{(1)+(4)}	SNR	6.7	9.3	13.4	17.9
	ASNR	-1.4	0.3	3.0	6.4
	CSII	0.4	0.7	0.9	1.0
	WPESQ	1.1	1.1	1.3	1.6
	Csig	0.4	0.8	1.4	2.2
	Cbak	1.6	1.8	2.3	2.9
	Covl	0.6	0.9	1.5	2.2

Table 48: Combination of estimates obtained from different algorithms in WGN conditions.

B.3. Tables for Section 5.4 (Colored noise handling)

The algorithms with the "regular" or "traditional" handling of colored noise can be recognized from the appended letter "R" in the following tables of results.

B.3.1. Cafeteria Noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
DKF	-1.5	-5.7	0.2	1.0	0.3	0.9	0.5
DKF-R	-1.7	-5.8	0.1	1.0	0.3	0.9	0.5
KEM_{Burg}	-2.4	-6.1	0.2	1.0	0.3	0.9	0.5
KEM_{Burg}-R	-2.4	-6.1	0.2	1.0	0.3	0.9	0.5
RBPF	-1.0	-5.8	0.1	1.0	0.3	1.0	0.4
RBPF-R	-1.2	-6.0	0.1	1.0	0.3	0.9	0.4

Table 49: "Traditional" (-R) vs. proposed fullband colored noise handling – results in VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
DKF	4.3	-2.7	0.4	1.1	0.8	1.3	0.8
DKF-R	4.1	-2.8	0.4	1.1	0.8	1.3	0.8
KEM_{Burg}	3.9	-2.9	0.4	1.1	0.9	1.3	0.8
KEM_{Burg}-R	3.9	-2.8	0.4	1.1	0.9	1.3	0.9
RBPF	4.4	-2.9	0.4	1.1	0.8	1.3	0.8
RBPF-R	3.5	-3.1	0.4	1.1	0.7	1.2	0.7

Table 50: "Traditional" (-R) vs. proposed fullband colored noise handling – results in L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
DKF	9.4	0.4	0.8	1.2	1.3	1.7	1.2
DKF-R	9.3	0.4	0.8	1.2	1.3	1.7	1.2
KEM_{Burg}	9.4	0.6	0.9	1.3	1.4	1.7	1.3
KEM_{Burg}-R	9.4	0.6	0.9	1.3	1.4	1.7	1.3
RBPF	9.6	0.3	0.9	1.2	1.4	1.7	1.2
RBPF-R	9.3	0.2	0.8	1.2	1.3	1.7	1.2

Table 51: "Traditional" (-R) vs. proposed fullband colored noise handling – results in M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
DKF	12.7	2.6	1.0	1.4	1.7	2.0	1.5
DKF-R	12.7	2.6	1.0	1.4	1.7	2.0	1.5
KEM_{Burg}	12.5	2.8	1.0	1.5	1.9	2.1	1.7
KEM_{Burg}-R	12.5	2.8	1.0	1.5	1.9	2.1	1.7
RBPF	13.6	3.5	1.0	1.5	2.4	2.3	2.0
RBPF-R	13.4	3.5	1.0	1.5	2.4	2.3	2.0

Table 52: "Traditional" (-R) vs. proposed fullband colored noise handling – results in H-CAF conditions.

B.3.2. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
DKF	-1.4	-5.8	0.2	1.0	0.7	1.1	0.7
DKF-R	-1.5	-6.0	0.2	1.0	0.7	1.0	0.7
KEM_{Burg}	-2.8	-6.4	0.3	1.0	0.8	1.1	0.8
KEM_{Burg}-R	-2.8	-6.4	0.3	1.0	0.8	1.1	0.7
RBPF	-0.3	-5.7	0.3	1.0	0.7	1.1	0.7
RBPF-R	-1.3	-6.1	0.2	1.0	0.7	1.1	0.7

Table 53: "Traditional" (-R) vs. proposed fullband colored noise handling – results in VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
DKF	5.5	-2.1	0.8	1.1	1.3	1.5	1.2
DKF-R	5.4	-2.3	0.8	1.1	1.3	1.5	1.1
KEM_{Burg}	4.7	-2.6	0.9	1.2	1.5	1.5	1.3
KEM_{Burg}-R	4.7	-2.6	0.9	1.2	1.5	1.5	1.3
RBPF	5.7	-2.4	0.9	1.2	1.4	1.6	1.2
RBPF-R	4.1	-3.1	0.8	1.1	1.3	1.5	1.1

Table 54: "Traditional" (-R) vs. proposed fullband colored noise handling – results in L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
DKF	8.7	-0.2	1.0	1.2	1.7	1.8	1.4
DKF-R	8.6	-0.3	0.9	1.2	1.7	1.7	1.4
KEM_{Burg}	8.2	-0.5	1.0	1.3	1.9	1.8	1.6
KEM_{Burg}-R	8.2	-0.5	1.0	1.3	1.9	1.8	1.6
RBPF	8.7	-0.5	1.0	1.3	1.8	1.8	1.5
RBPF-R	8.0	-0.7	0.9	1.3	1.7	1.8	1.5

Table 55: "Traditional" (-R) vs. proposed fullband colored noise handling – results in M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
DKF	11.8	1.8	1.0	1.5	2.1	2.1	1.8
DKF-R	11.7	1.7	1.0	1.5	2.1	2.1	1.8
KEM_{Burg}	11.7	1.7	1.0	1.6	2.3	2.2	2.0
KEM_{Burg}-R	11.7	1.7	1.0	1.6	2.3	2.2	2.0
RBPF	11.8	1.6	1.0	1.6	2.2	2.1	1.9
RBPF-R	11.2	1.6	1.0	1.5	2.1	2.1	1.9

Table 56: "Traditional" (-R) vs. proposed fullband colored noise handling – results in H-FAC conditions.

B.3.3. Military vehicle noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
DKF	-3.9	-6.3	0.2	1.0	1.0	1.0	0.8
DKF-R	-4.0	-6.4	0.1	1.0	0.9	1.0	0.8
KEM_{Burg}	-2.9	-6.5	0.1	1.0	1.0	1.0	0.8
KEM_{Burg}-R	-2.9	-6.5	0.1	1.0	1.0	1.0	0.8
RBPF	0.2	-5.4	0.1	1.0	1.0	1.1	0.8
RBPF-R	-2.0	-6.2	0.0	1.0	0.8	1.0	0.8

Table 57: "Traditional" (-R) vs. proposed fullband colored noise handling – results in VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
DKF	3.9	-2.5	0.8	1.1	1.5	1.4	1.2
DKF-R	3.8	-2.6	0.8	1.1	1.4	1.4	1.1
KEM_{Burg}	4.6	-2.5	0.8	1.1	1.5	1.4	1.2
KEM_{Burg}-R	4.5	-2.5	0.8	1.1	1.5	1.4	1.2
RBPF	5.9	-2.1	1.0	1.1	1.5	1.4	1.2
RBPF-R	3.8	-2.5	0.5	1.1	1.5	1.3	1.2

Table 58: "Traditional" (-R) vs. proposed fullband colored noise handling – results in L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
DKF	9.0	0.4	1.0	1.2	1.9	1.7	1.5
DKF-R	8.9	0.3	1.0	1.2	1.9	1.7	1.5
KEM_{Burg}	10.4	1.3	1.0	1.2	1.7	1.8	1.4
KEM_{Burg}-R	10.4	1.3	1.0	1.2	1.7	1.8	1.4
RBPF	10.0	0.5	1.0	1.2	1.9	1.8	1.5
RBPF-R	7.7	-0.1	1.0	1.2	1.9	1.7	1.5

Table 59: "Traditional" (-R) vs. proposed fullband colored noise handling – results in M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
DKF	13.6	3.3	1.0	1.5	2.3	2.2	1.9
DKF-R	13.6	3.3	1.0	1.5	2.3	2.2	1.9
KEM_{Burg}	14.1	3.6	1.0	1.6	2.5	2.3	2.1
KEM_{Burg}-R	14.1	3.6	1.0	1.6	2.5	2.3	2.1
RBPF	13.5	3.6	1.0	2.4	3.5	2.8	3.1
RBPF-R	13.4	3.1	1.0	2.3	3.5	2.9	3.1

Table 60: "Traditional" (-R) vs. proposed fullband colored noise handling – results in H-MIL conditions.

B.3.4. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
DKF	-8.0	-6.7	1.0	1.1	1.9	1.3	1.5
DKF-R	-8.1	-6.8	1.0	1.1	1.9	1.3	1.5
KEM_{Burg}	-8.0	-7.1	1.0	1.1	1.9	1.3	1.5
KEM_{Burg}-R	-8.0	-7.1	1.0	1.1	1.9	1.3	1.5
RBPF	0.1	-4.8	1.0	1.1	2.0	1.5	1.6
RBPF-R	-3.8	-6.2	1.0	1.1	2.0	1.4	1.6

Table 61: "Traditional" (-R) vs. proposed fullband colored noise handling – results in VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
DKF	-1.5	-3.9	1.0	1.2	2.4	1.7	1.9
DKF-R	-1.6	-3.9	1.0	1.2	2.4	1.7	1.9
KEM_{Burg}	-1.4	-4.2	1.0	1.3	2.4	1.7	2.0
KEM_{Burg}-R	-1.4	-4.2	1.0	1.3	2.4	1.7	2.0
RBPF	4.8	-2.3	1.0	1.3	2.5	1.9	2.1
RBPF-R	1.1	-3.9	1.0	1.3	2.5	1.8	2.0

Table 62: "Traditional" (-R) vs. proposed fullband colored noise handling – results in L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
DKF	6.6	0.4	1.0	1.6	3.0	2.3	2.5
DKF-R	6.5	0.3	1.0	1.6	3.0	2.3	2.5
KEM_{Burg}	6.7	0.3	1.0	1.9	3.0	2.3	2.6
KEM_{Burg}-R	6.7	0.3	1.0	1.9	3.0	2.3	2.6
RBPF	10.0	1.1	1.0	1.9	3.1	2.4	2.6
RBPF-R	9.0	0.7	1.0	1.7	3.0	2.3	2.6

Table 63: "Traditional" (-R) vs. proposed fullband colored noise handling – results in M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
DKF	11.9	3.2	1.0	2.1	3.3	2.7	2.8
DKF-R	11.9	3.2	1.0	2.1	3.3	2.7	2.8
KEM_{Burg}	12.2	3.4	1.0	2.5	3.5	2.8	3.1
KEM_{Burg}-R	12.2	3.4	1.0	2.5	3.5	2.8	3.1
RBPF	13.5	3.6	1.0	2.4	3.6	2.9	3.1
RBPF-R	10.1	3.0	1.0	2.3	3.5	2.8	3.1

Table 64: "Traditional" (-R) vs. proposed fullband colored noise handling – results in H-CAR conditions.

B.4. Tables for Section 6.6 (Neural-based PFs)

B.4.1. White Gaussian noise

VL-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-4.1	-7.5	0.1	1.0	0.2	1.2	0.5
NPF- $\bar{D}\bar{M}\bar{B}$	6.9	-1.2	0.4	1.1	0.5	1.6	0.8
NPF- $\bar{D}\bar{M}B$	6.7	-1.3	0.3	1.1	0.4	1.5	0.7
NPF- $\bar{D}M\bar{B}$	7.1	-1.3	0.4	1.1	0.7	1.7	0.8
NPF- $\bar{D}MB$	7.0	-1.1	0.3	1.1	0.6	1.6	0.8
NPF- $D\bar{M}\bar{B}$	5.7	-1.7	0.3	1.1	0.3	1.5	0.6
NPF- $D\bar{M}B$	5.5	-2.0	0.2	1.0	0.2	1.4	0.5
NPF- $DM\bar{B}$	5.9	-1.8	0.2	1.1	0.3	1.5	0.6
NPF- DMB	6.9	-1.2	0.4	1.1	0.5	1.6	0.8
RBPF	6.8	-1.3	0.4	1.1	0.4	1.6	0.6

Table 65: Neural-based PFs vs. RBPF in VL-WGN conditions.

L-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.1	-5.6	0.2	1.0	0.6	1.5	0.8
NPF- $\bar{D}\bar{M}\bar{B}$	9.6	0.5	0.7	1.2	1.0	1.9	0.9
NPF- $\bar{D}\bar{M}B$	9.1	0.3	0.6	1.1	0.9	1.8	0.9
NPF- $\bar{D}M\bar{B}$	9.9	0.7	0.9	1.2	1.1	1.9	1.1
NPF- $\bar{D}MB$	9.4	0.2	0.6	1.2	1.0	1.8	1.1
NPF- $D\bar{M}\bar{B}$	8.4	-0.2	0.6	1.1	0.8	1.8	0.9
NPF- $D\bar{M}B$	7.6	-1.1	0.5	1.1	0.5	1.6	0.8
NPF- $DM\bar{B}$	8.2	-0.3	0.6	1.1	0.8	1.7	0.9
NPF- DMB	8.1	-0.4	0.6	1.1	0.6	1.7	0.9
RBPF	9.5	0.4	0.7	1.1	0.8	1.8	1.0

Table 66: Neural-based PFs vs. RBPF in L-WGN conditions.

M-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.1	-2.1	0.6	1.1	1.2	2.0	1.3
NPF- $\bar{D}\bar{M}\bar{B}$	12.9	2.7	0.9	1.2	1.2	2.2	1.3
NPF- $\bar{D}\bar{M}B$	13.0	2.6	0.9	1.1	1.1	2.0	1.2
NPF- $\bar{D}M\bar{B}$	13.4	2.9	0.9	1.2	1.5	2.3	1.5
NPF- $\bar{D}MB$	13.1	2.6	0.9	1.2	1.5	2.2	1.4
NPF- $D\bar{M}\bar{B}$	12.1	2.1	0.9	1.0	1.0	2.1	1.1
NPF- $D\bar{M}B$	11.0	1.7	0.9	0.9	0.9	1.9	1.0
NPF- $DM\bar{B}$	12.0	2.0	0.9	1.0	1.2	2.1	1.2
NPF- DMB	11.8	1.8	0.9	1.0	0.9	2.0	1.0
RBPF	13.7	3.2	0.9	1.3	1.5	2.3	1.5

Table 67: Neural-based PFs vs. RBPF in M-WGN conditions.

H-WGN	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	12.1	1.8	0.9	1.2	1.9	2.6	2.0
NPF- $\bar{D}\bar{M}\bar{B}$	17.8	6.4	1.0	1.7	2.0	2.9	2.0
NPF- $\bar{D}\bar{M}B$	17.5	5.8	1.0	1.7	1.9	2.8	2.0
NPF- $\bar{D}M\bar{B}$	18.3	6.5	1.0	1.8	2.1	2.8	2.1
NPF- $\bar{D}MB$	17.7	6.0	1.0	1.7	1.8	2.8	2.2
NPF- $D\bar{M}\bar{B}$	16.9	5.6	1.0	1.3	1.8	2.8	1.9
NPF- $D\bar{M}B$	15.5	5.1	1.0	1.2	1.7	2.5	1.8
NPF- $DM\bar{B}$	16.8	5.5	1.0	1.4	1.8	2.7	1.9
NPF- DMB	16.3	5.4	1.0	1.3	1.7	2.5	1.7
RBPF	18.4	6.7	1.0	1.9	2.2	2.9	2.2

Table 68: Neural-based PFs vs. RBPF in H-WGN conditions.

B.4.2. Cafeteria noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
DKF	-1.5	-5.7	0.2	1.0	0.3	0.9	0.5
KEM_{Burg}	-2.4	-6.1	0.2	1.0	0.3	0.9	0.5
RBPF	-1.0	-5.8	0.1	1.0	0.3	1.0	0.4
NPF- $\bar{D}\bar{M}\bar{B}$	0.5	-4.8	0.2	1.0	0.5	1.0	0.6
NPF- $\bar{D}\bar{M}B$	-1.8	-5.5	0.2	1.0	0.4	0.9	0.5
NPF- $\bar{D}M\bar{B}$	0.8	-4.5	0.2	1.1	0.5	1.0	0.6
NPF- $\bar{D}MB$	0.7	-5.0	0.2	1.0	0.4	1.0	0.5
NPF- $D\bar{M}\bar{B}$	0.0	-5.3	0.2	1.0	0.4	0.9	0.5
NPF- $D\bar{M}B$	-0.9	-6.1	0.2	1.0	0.3	0.9	0.4
NPF- $DM\bar{B}$	0.1	-4.9	0.2	1.0	0.5	0.9	0.5
NPF- DMB	-0.2	-5.3	0.2	1.0	0.4	0.9	0.5

Table 69: Neural-based PFs vs. RBPF in VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
DKF	4.3	-2.7	0.4	1.1	0.8	1.3	0.8
KEM_{Burg}	3.9	-2.9	0.4	1.1	0.9	1.3	0.8
RBPF	4.4	-2.9	0.4	1.1	0.8	1.3	0.8
NPF- $\bar{D}\bar{M}\bar{B}$	6.2	-1.7	0.5	1.1	1.0	1.4	0.9
NPF- $\bar{D}\bar{M}B$	3.2	-3.1	0.4	1.1	0.8	1.2	0.8
NPF- $\bar{D}M\bar{B}$	5.9	-1.9	0.5	1.1	1.0	1.4	1.0
NPF- $\bar{D}MB$	4.3	-2.7	0.4	1.1	1.0	1.3	0.9
NPF- $D\bar{M}\bar{B}$	4.2	-3.0	0.4	1.1	0.8	1.3	0.8
NPF- $D\bar{M}B$	4.0	-3.1	0.4	1.1	0.8	1.2	0.7
NPF- $DM\bar{B}$	4.3	-2.9	0.4	1.1	0.8	1.3	0.9
NPF- DMB	4.1	-3.0	0.4	1.1	0.8	1.3	0.8

Table 70: Neural-based PFs vs. RBPF in L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
DKF	9.4	0.4	0.8	1.2	1.3	1.7	1.2
KEM_{Burg}	9.4	0.6	0.9	1.3	1.4	1.7	1.3
RBPF	9.6	0.3	0.9	1.2	1.4	1.7	1.2
NPF- $\bar{D}\bar{M}\bar{B}$	9.7	0.8	0.9	1.2	1.3	1.7	1.3
NPF- $\bar{D}\bar{M}B$	8.1	-0.1	0.8	1.1	1.2	1.7	1.2
NPF- $\bar{D}M\bar{B}$	9.5	0.8	1.0	1.2	1.4	1.7	1.4
NPF- $\bar{D}MB$	9.1	0.3	0.8	1.2	1.3	1.7	1.2
NPF- $D\bar{M}\bar{B}$	8.1	0.1	0.8	1.2	1.3	1.6	1.2
NPF- $D\bar{M}B$	7.8	-0.3	0.8	1.2	1.2	1.6	1.1
NPF- $DM\bar{B}$	8.8	0.2	0.8	1.2	1.3	1.6	1.3
NPF- DMB	8.0	0.0	0.8	1.2	1.3	1.6	1.2

Table 71: Neural-based PFs vs. RBPF in M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
DKF	12.7	2.6	1.0	1.4	1.7	2.0	1.5
KEM_{Burg}	12.5	2.8	1.0	1.5	1.9	2.1	1.7
RBPF	13.6	3.5	1.0	1.5	2.4	2.3	2.0
NPF- $\bar{D}\bar{M}\bar{B}$	11.0	2.6	1.0	1.5	1.9	2.1	1.6
NPF- $\bar{D}\bar{M}B$	11.4	2.4	1.0	1.5	1.8	2.1	1.6
NPF- $\bar{D}M\bar{B}$	12.1	2.7	1.0	1.6	1.9	2.2	1.7
NPF- $\bar{D}MB$	12.0	2.5	1.0	1.5	1.8	2.1	1.7
NPF- $D\bar{M}\bar{B}$	10.1	2.4	1.0	1.4	1.8	2.0	1.6
NPF- $D\bar{M}B$	9.6	2.2	0.9	1.4	1.7	2.0	1.5
NPF- $DM\bar{B}$	10.9	2.4	1.0	1.5	1.8	2.0	1.6
NPF- DMB	10.1	2.3	1.0	1.4	1.8	2.0	1.6

Table 72: Neural-based PFs vs. RBPF in H-CAF conditions.

B.4.3. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
DKF	-1.4	-5.8	0.2	1.0	0.7	1.1	0.7
KEM_{Burg}	-2.8	-6.4	0.3	1.0	0.8	1.1	0.8
RBPF	-0.3	-5.7	0.3	1.0	0.7	1.1	0.7
NPF- $\bar{D}\bar{M}\bar{B}$	-0.3	-5.7	0.4	1.0	0.9	1.1	0.8
NPF- $\bar{D}\bar{M}B$	-1.3	-6.1	0.3	1.0	0.8	1.0	0.7
NPF- $\bar{D}M\bar{B}$	0.1	-5.3	0.4	1.1	0.9	1.1	0.9
NPF- $\bar{D}MB$	-0.4	-5.8	0.4	1.0	0.8	1.1	0.8
NPF- $D\bar{M}\bar{B}$	-1.1	-6.3	0.3	1.0	0.8	1.0	0.7
NPF- $D\bar{M}B$	-2.0	-6.8	0.3	1.0	0.7	0.9	0.7
NPF- $DM\bar{B}$	-0.2	-6.1	0.4	1.0	0.8	1.0	0.8
NPF- DMB	-1.0	-6.2	0.4	1.0	0.8	1.0	0.8

Table 73: Neural-based PFs vs. RBPF in VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
DKF	5.5	-2.1	0.8	1.1	1.3	1.5	1.2
KEM_{Burg}	4.7	-2.6	0.9	1.2	1.5	1.5	1.3
RBPF	5.7	-2.4	0.9	1.2	1.4	1.6	1.2
NPF- $\bar{D}\bar{M}\bar{B}$	5.5	-2.6	0.8	1.1	1.4	1.5	1.1
NPF- $\bar{D}\bar{M}B$	5.3	-2.8	0.8	1.1	1.4	1.5	1.1
NPF- $\bar{D}M\bar{B}$	6.0	-1.8	0.9	1.2	1.5	1.6	1.4
NPF- $\bar{D}MB$	5.7	-2.3	0.9	1.2	1.4	1.5	1.2
NPF- $D\bar{M}\bar{B}$	4.9	-3.0	0.8	1.1	1.4	1.5	1.1
NPF- $D\bar{M}B$	4.1	-4.0	0.8	1.1	1.3	1.5	1.1
NPF- $DM\bar{B}$	5.2	-2.7	0.8	1.1	1.4	1.5	1.1
NPF- DMB	4.9	-3.1	0.8	1.1	1.4	1.5	1.1

Table 74: Neural-based PFs vs. RBPF in L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
DKF	8.7	-0.2	1.0	1.2	1.7	1.8	1.4
KEM_{Burg}	8.2	-0.5	1.0	1.3	1.9	1.8	1.6
RBPF	8.7	-0.5	1.0	1.3	1.8	1.8	1.5
NPF- $\bar{D}\bar{M}\bar{B}$	8.5	-0.4	1.0	1.2	1.8	1.8	1.6
NPF- $\bar{D}\bar{M}B$	8.4	-0.6	1.0	1.2	1.8	1.8	1.5
NPF- $\bar{D}M\bar{B}$	8.9	-0.3	1.0	1.3	1.9	1.9	1.7
NPF- $\bar{D}MB$	8.5	-0.6	1.0	1.3	1.8	1.8	1.6
NPF- $D\bar{M}\bar{B}$	7.8	-0.9	1.0	1.2	1.8	1.7	1.5
NPF- $D\bar{M}B$	7.3	-1.2	1.0	1.2	1.7	1.6	1.4
NPF- $DM\bar{B}$	8.1	-0.5	1.0	1.2	1.8	1.7	1.5
NPF- DMB	8.0	-0.7	1.0	1.2	1.8	1.7	1.5

Table 75: Neural-based PFs vs. RBPF in M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
DKF	11.8	1.8	1.0	1.5	2.1	2.1	1.8
KEM_{Burg}	11.7	1.7	1.0	1.6	2.3	2.2	2.0
RBPF	11.8	1.6	1.0	1.6	2.2	2.1	1.9
NPF- $\bar{D}\bar{M}\bar{B}$	10.2	1.2	1.0	1.5	2.2	2.0	1.9
NPF- $\bar{D}\bar{M}B$	9.4	1.1	1.0	1.4	2.2	2.0	1.8
NPF- $\bar{D}M\bar{B}$	11.3	1.3	1.0	1.6	2.2	2.1	2.1
NPF- $\bar{D}MB$	10.3	1.2	1.0	1.5	2.2	2.0	1.8
NPF- $D\bar{M}\bar{B}$	9.7	1.1	1.0	1.4	2.1	2.0	1.8
NPF- $D\bar{M}B$	8.2	0.9	1.0	1.4	2.1	2.0	1.8
NPF- $DM\bar{B}$	10.0	1.1	1.0	1.4	2.2	2.0	1.9
NPF- DMB	9.6	1.1	1.0	1.4	2.2	2.0	1.8

Table 76: Neural-based PFs vs. RBPF in H-FAC conditions.

B.4.4. Military vehicle noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
DKF	-3.9	-6.3	0.2	1.0	1.0	1.0	0.8
KEM_{Burg}	-2.9	-6.5	0.1	1.0	1.0	1.0	0.8
RBPF	0.2	-5.4	0.1	1.0	1.0	1.1	0.8
NPF- $\bar{D}\bar{M}\bar{B}$	0.0	-5.4	0.1	1.0	1.0	1.0	0.9
NPF- $\bar{D}\bar{M}B$	-0.2	-5.5	0.1	1.0	1.0	0.9	0.8
NPF- $\bar{D}M\bar{B}$	0.8	-4.4	0.1	1.0	1.1	1.1	1.0
NPF- $\bar{D}MB$	0.2	-5.1	0.1	1.0	1.1	1.0	0.9
NPF- $D\bar{M}\bar{B}$	-1.2	-6.0	0.1	1.0	0.9	1.0	0.8
NPF- $D\bar{M}B$	-2.5	-6.5	0.1	1.0	0.8	0.9	0.7
NPF- $DM\bar{B}$	-0.5	-5.3	0.1	1.0	1.0	1.0	0.8
NPF- DMB	-0.9	-6.0	0.1	1.0	1.0	0.9	0.8

Table 77: Neural-based PFs vs. RBPF in VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
DKF	3.9	-2.5	0.8	1.1	1.5	1.4	1.2
KEM_{Burg}	4.6	-2.5	0.8	1.1	1.5	1.4	1.2
RBPF	5.9	-2.1	1.0	1.1	1.5	1.4	1.2
NPF- $\bar{D}\bar{M}\bar{B}$	6.2	-1.7	1.0	1.1	1.6	1.4	1.2
NPF- $\bar{D}\bar{M}B$	4.6	-2.4	0.8	1.1	1.5	1.4	1.2
NPF- $\bar{D}M\bar{B}$	6.1	-1.5	1.0	1.1	1.6	1.4	1.3
NPF- $\bar{D}MB$	5.2	-2.1	0.8	1.1	1.5	1.4	1.2
NPF- $D\bar{M}\bar{B}$	5.3	-2.3	0.9	1.1	1.4	1.4	1.2
NPF- $D\bar{M}B$	4.2	-3.1	0.7	1.1	1.4	1.3	1.2
NPF- $DM\bar{B}$	5.4	-2.1	0.9	1.1	1.5	1.4	1.2
NPF- DMB	5.0	-2.6	0.7	1.1	1.5	1.4	1.2

Table 78: Neural-based PFs vs. RBPF in L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
DKF	9.0	0.4	1.0	1.2	1.9	1.7	1.5
KEM_{Burg}	10.4	1.3	1.0	1.2	1.7	1.8	1.4
RBPF	10.0	0.5	1.0	1.2	1.9	1.8	1.5
NPF- $\bar{D}\bar{M}\bar{B}$	8.3	0.4	1.0	1.2	2.0	1.8	1.5
NPF- $\bar{D}\bar{M}B$	8.1	0.4	1.0	1.2	1.9	1.7	1.4
NPF- $\bar{D}M\bar{B}$	9.1	0.5	1.0	1.2	2.0	1.8	1.5
NPF- $\bar{D}MB$	8.6	0.4	1.0	1.2	2.0	1.7	1.5
NPF- $D\bar{M}\bar{B}$	8.1	0.4	1.0	1.2	1.9	1.8	1.5
NPF- $D\bar{M}B$	7.9	0.3	1.0	1.2	1.9	1.7	1.4
NPF- $DM\bar{B}$	8.8	0.4	1.0	1.2	2.0	1.8	1.5
NPF- DMB	8.3	0.4	1.0	1.2	1.9	1.8	1.5

Table 79: Neural-based PFs vs. RBPF in M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
DKF	13.6	3.3	1.0	1.5	2.3	2.2	1.9
KEM_{Burg}	14.1	3.6	1.0	1.6	2.5	2.3	2.1
RBPF	13.5	3.6	1.0	2.4	3.5	2.8	3.1
NPF- $\bar{D}\bar{M}\bar{B}$	13.0	3.5	1.0	2.2	3.3	2.1	2.8
NPF- $\bar{D}\bar{M}B$	12.9	3.4	1.0	2.2	3.1	2.1	2.8
NPF- $\bar{D}M\bar{B}$	13.1	3.5	1.0	2.3	3.4	2.2	2.9
NPF- $\bar{D}MB$	13.0	3.5	1.0	2.1	3.1	2.1	2.9
NPF- $D\bar{M}\bar{B}$	12.2	3.3	1.0	1.9	2.9	2.1	2.6
NPF- $D\bar{M}B$	12.0	3.1	1.0	1.6	2.7	2.0	2.3
NPF- $DM\bar{B}$	12.8	3.3	1.0	2.1	3.1	2.1	2.7
NPF- DMB	12.3	3.2	1.0	2.0	2.9	2.0	2.4

Table 80: Neural-based PFs vs. RBPF in H-MIL conditions.

B.4.5. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
DKF	-8.0	-6.7	1.0	1.1	1.9	1.3	1.5
KEM_{Burg}	-8.0	-7.1	1.0	1.1	1.9	1.3	1.5
RBPF	0.1	-4.8	1.0	1.1	2.0	1.5	1.6
NPF- $\bar{D}\bar{M}\bar{B}$	0.1	-4.8	1.0	1.1	2.0	1.5	1.6
NPF- $\bar{D}\bar{M}B$	-0.1	-4.9	1.0	1.1	1.9	1.4	1.6
NPF- $\bar{D}M\bar{B}$	0.3	-3.5	1.0	1.2	2.0	1.6	1.8
NPF- $\bar{D}MB$	0.1	-4.7	1.0	1.1	1.9	1.5	1.6
NPF- $D\bar{M}\bar{B}$	-0.2	-5.1	1.0	1.1	1.9	1.4	1.5
NPF- $D\bar{M}B$	-0.9	-5.2	1.0	1.1	1.9	1.3	1.4
NPF- $DM\bar{B}$	0.0	-4.8	1.0	1.1	2.0	1.4	1.6
NPF- DMB	-0.4	-4.9	1.0	1.1	1.9	1.4	1.6

Table 81: Neural-based PFs vs. RBPF in VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
DKF	-1.5	-3.9	1.0	1.2	2.4	1.7	1.9
KEM_{Burg}	-1.6	-3.9	1.0	1.2	2.4	1.7	1.9
DUKF(1)	-1.4	-4.2	1.0	1.3	2.4	1.7	2.0
NPF- $\bar{D}\bar{M}\bar{B}$	-1.4	-4.0	1.0	1.2	2.4	1.7	2.0
NPF- $\bar{D}\bar{M}B$	-1.5	-4.2	1.0	1.2	2.4	1.7	1.9
NPF- $\bar{D}M\bar{B}$	1.2	-2.4	1.0	1.3	2.5	1.8	2.0
NPF- $\bar{D}MB$	0.8	-3.7	1.0	1.2	2.5	1.8	1.9
NPF- $D\bar{M}\bar{B}$	-2.1	-4.7	1.0	1.2	2.4	1.7	1.9
NPF- $D\bar{M}B$	-2.2	-4.9	1.0	1.2	2.3	1.6	1.9
NPF- $DM\bar{B}$	-1.9	-4.2	1.0	1.2	2.5	1.8	2.0
NPF- DMB	-2.0	-4.4	1.0	1.2	2.4	1.7	1.9

Table 82: Neural-based PFs vs. RBPF in L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
DKF	6.6	0.4	1.0	1.6	3.0	2.3	2.5
KEM_{Burg}	6.7	0.3	1.0	1.9	3.0	2.3	2.6
RBPF	10.0	1.1	1.0	1.9	3.1	2.4	2.6
NPF- $\bar{D}\bar{M}\bar{B}$	8.8	0.3	1.0	1.7	2.9	2.3	2.4
NPF- $\bar{D}\bar{M}B$	8.9	0.1	1.0	1.6	2.8	2.3	2.2
NPF- $\bar{D}M\bar{B}$	9.1	0.4	1.0	1.7	3.0	2.4	2.4
NPF- $\bar{D}MB$	8.9	0.3	1.0	1.6	2.9	2.3	2.4
NPF- $D\bar{M}\bar{B}$	8.5	0.0	1.0	1.6	2.8	2.3	2.2
NPF- $D\bar{M}B$	8.2	-0.2	1.0	1.5	2.7	2.3	2.0
NPF- $DM\bar{B}$	9.0	0.1	1.0	1.7	2.8	2.4	2.4
NPF- DMB	8.4	0.0	1.0	1.7	2.8	2.3	2.2

Table 83: Neural-based PFs vs. RBPF in M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
DKF	11.9	3.2	1.0	2.1	3.3	2.7	2.8
KEM_{Burg}	12.2	3.4	1.0	2.5	3.5	2.8	3.1
RBPF	13.5	3.6	1.0	2.4	3.6	2.9	3.1
NPF- $\bar{D}\bar{M}\bar{B}$	12.3	3.0	1.0	2.4	3.3	2.8	3.1
NPF- $\bar{D}\bar{M}B$	12.0	2.7	1.0	2.4	3.3	2.7	3.0
NPF- $\bar{D}M\bar{B}$	12.7	3.1	1.0	2.4	3.6	2.9	3.2
NPF- $\bar{D}MB$	12.5	3.1	1.0	2.4	3.5	2.8	3.1
NPF- $D\bar{M}\bar{B}$	11.5	2.9	1.0	2.3	3.2	2.7	2.9
NPF- $D\bar{M}B$	11.1	2.7	1.0	2.3	3.2	2.7	2.9
NPF- $DM\bar{B}$	12.1	2.9	1.0	2.4	3.5	2.8	3.1
NPF- DMB	11.9	2.9	1.0	2.4	3.5	2.8	3.0

Table 84: Neural-based PFs vs. RBPF in H-CAR conditions.

B.5. Tables for Section 7.4 (All-pole modelling of noise)

B.5.1. Cafeteria noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
DKF+YW	-1.5	-5.7	0.2	1.0	0.3	0.9	0.5
DKF+IS	-1.1	-5.4	0.2	1.0	0.4	1.0	0.6
DKF+RMS	-1.2	-5.4	0.2	1.0	0.4	1.0	0.5
DKF+COSH	-1.1	-5.5	0.2	1.0	0.4	1.1	0.5
DKF+MSE	-0.9	-5.4	0.2	1.0	0.4	1.1	0.6
KEM_{Burg}+YW	-2.4	-6.1	0.2	1.0	0.3	0.9	0.5
KEM_{Burg}+IS	-1.8	-5.8	0.2	1.0	0.4	1.0	0.5
KEM_{Burg}+RMS	-1.9	-5.8	0.2	1.0	0.3	1.0	0.5
KEM_{Burg}+COSH	-1.9	-5.8	0.2	1.1	0.3	1.0	0.5
KEM_{Burg}+MSE	-1.5	-5.8	0.2	1.1	0.3	1.0	0.6
DUKF(1)+YW	-1.4	-5.9	0.2	1.1	0.4	0.9	0.5
DUKF(1)+IS	-1.4	-5.8	0.2	1.1	0.4	1.0	0.5
DUKF(1)+RMS	-1.4	-5.8	0.2	1.1	0.4	1.0	0.6
DUKF(1)+COSH	-1.3	-5.7	0.2	1.1	0.4	1.0	0.6
DUKF(1)+MSE	-1.3	-5.6	0.2	1.1	0.4	1.1	0.6

Table 85: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
DKF+YW	4.3	-2.7	0.4	1.1	0.8	1.3	0.8
DKF+IS	4.5	-2.5	0.4	1.1	0.8	1.4	0.9
DKF+RMS	4.5	-2.5	0.4	1.1	0.8	1.3	0.9
DKF+COSH	4.5	-2.5	0.4	1.1	0.8	1.4	0.9
DKF+MSE	4.5	-2.5	0.5	1.1	0.9	1.4	0.9
KEM_{Burg}+YW	3.9	-2.9	0.4	1.1	0.9	1.3	0.8
KEM_{Burg}+IS	4.3	-2.6	0.4	1.1	0.9	1.4	0.9
KEM_{Burg}+RMS	4.2	-2.6	0.4	1.1	0.9	1.4	0.9
KEM_{Burg}+COSH	4.4	-2.6	0.5	1.1	0.9	1.4	0.9
KEM_{Burg}+MSE	4.6	-2.5	0.5	1.1	0.9	1.4	1.0
DUKF(1)+YW	4.6	-2.7	0.4	1.1	0.8	1.3	0.8
DUKF(1)+IS	4.7	-2.5	0.4	1.1	0.8	1.4	0.9
DUKF(1)+RMS	4.6	-2.5	0.4	1.1	0.8	1.4	0.9
DUKF(1)+COSH	4.7	-2.4	0.4	1.1	0.8	1.4	0.9
DUKF(1)+MSE	4.9	-2.4	0.4	1.1	0.8	1.4	0.9

Table 86: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
DKF+YW	9.4	0.4	0.8	1.2	1.3	1.7	1.2
DKF+IS	9.4	0.6	0.9	1.2	1.3	1.8	1.2
DKF+RMS	9.5	0.6	0.9	1.2	1.3	1.8	1.3
DKF+COSH	9.5	0.6	0.9	1.2	1.3	1.8	1.3
DKF+MSE	9.5	0.6	0.9	1.2	1.3	1.8	1.3
KEM_{Burg}+YW	9.4	0.6	0.9	1.3	1.4	1.7	1.3
KEM_{Burg}+IS	9.6	0.7	0.9	1.3	1.4	1.8	1.4
KEM_{Burg}+RMS	9.5	0.7	0.9	1.3	1.4	1.9	1.4
KEM_{Burg}+COSH	9.6	0.7	0.9	1.3	1.5	1.9	1.4
KEM_{Burg}+MSE	9.7	0.7	0.9	1.3	1.5	1.9	1.4
DUKF(1)+YW	8.5	0.2	0.8	1.3	1.3	1.7	1.2
DUKF(1)+IS	8.5	0.3	0.8	1.3	1.3	1.8	1.3
DUKF(1)+RMS	8.8	0.3	0.9	1.3	1.3	1.8	1.3
DUKF(1)+COSH	8.5	0.3	0.8	1.3	1.3	1.8	1.3
DUKF(1)+MSE	8.6	0.3	0.8	1.3	1.3	1.8	1.3

Table 87: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
DKF+YW	12.7	2.6	1.0	1.4	1.7	2.0	1.5
DKF+IS	12.7	2.8	1.0	1.4	1.7	2.1	1.6
DKF+RMS	12.8	2.8	1.0	1.4	1.7	2.1	1.6
DKF+COSH	12.9	2.8	1.0	1.4	1.7	2.1	1.6
DKF+MSE	12.9	2.8	1.0	1.4	1.7	2.1	1.6
KEM_{Burg}+YW	12.5	2.8	1.0	1.5	1.9	2.1	1.7
KEM_{Burg}+IS	12.6	2.9	1.0	1.6	1.9	2.2	1.7
KEM_{Burg}+RMS	12.6	2.9	1.0	1.5	1.9	2.2	1.7
KEM_{Burg}+COSH	12.7	3.0	1.0	1.5	1.9	2.2	1.7
KEM_{Burg}+MSE	12.8	2.9	1.0	1.6	1.9	2.2	1.8
DUKF(1)+YW	9.9	1.8	1.0	1.5	1.7	2.0	1.5
DUKF(1)+IS	9.9	1.9	1.0	1.5	1.7	2.0	1.6
DUKF(1)+RMS	9.9	1.9	1.0	1.5	1.7	2.1	1.6
DUKF(1)+COSH	9.9	1.9	1.0	1.5	1.7	2.1	1.6
DUKF(1)+MSE	9.9	1.9	1.0	1.5	1.7	2.1	1.6

Table 88: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in H-CAF conditions.

B.5.2. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
DKF+YW	-1.4	-5.8	0.2	1.0	0.7	1.1	0.7
DKF+IS	-1.3	-5.7	0.2	1.0	0.7	1.2	0.8
DKF+RMS	-1.3	-5.7	0.2	1.0	0.7	1.1	0.8
DKF+COSH	-1.3	-5.7	0.2	1.0	0.7	1.2	0.8
DKF+MSE	-1.4	-5.7	0.2	1.0	0.7	1.2	0.8
KEM_{Burg}+YW	-2.8	-6.4	0.3	1.0	0.8	1.1	0.8
KEM_{Burg}+IS	-2.7	-6.2	0.7	1.0	0.8	1.2	0.8
KEM_{Burg}+RMS	-2.8	-6.2	0.7	1.0	0.8	1.2	0.8
KEM_{Burg}+COSH	-2.7	-6.2	0.6	1.0	0.8	1.2	0.8
KEM_{Burg}+MSE	-2.5	-6.1	0.6	1.0	0.8	1.2	0.8
DUKF(1)+YW	-0.2	-5.6	0.2	1.0	0.7	1.1	0.7
DUKF(1)+IS	-0.2	-5.5	0.2	1.1	0.7	1.2	0.8
DUKF(1)+RMS	-0.3	-5.5	0.2	1.1	0.7	1.2	0.8
DUKF(1)+COSH	-0.2	-5.5	0.2	1.1	0.7	1.2	0.8
DUKF(1)+MSE	-0.2	-5.4	0.2	1.0	0.7	1.2	0.8

Table 89: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
DKF+YW	5.5	-2.1	0.8	1.1	1.3	1.5	1.2
DKF+IS	5.6	-2.0	0.8	1.1	1.3	1.6	1.2
DKF+RMS	5.6	-2.0	0.9	1.1	1.4	1.6	1.2
DKF+COSH	5.8	-2.1	0.8	1.1	1.4	1.6	1.2
DKF+MSE	5.7	-2.0	0.8	1.1	1.4	1.6	1.2
KEM_{Burg}+YW	4.7	-2.6	0.9	1.2	1.5	1.5	1.3
KEM_{Burg}+IS	4.8	-2.5	0.9	1.2	1.5	1.7	1.4
KEM_{Burg}+RMS	4.8	-2.4	0.9	1.2	1.5	1.7	1.4
KEM_{Burg}+COSH	4.9	-2.4	0.9	1.2	1.5	1.7	1.4
KEM_{Burg}+MSE	5.0	-2.4	0.9	1.2	1.5	1.7	1.4
DUKF(1)+YW	4.1	-2.7	0.9	1.1	1.3	1.5	1.2
DUKF(1)+IS	4.3	-2.5	0.9	1.1	1.3	1.7	1.2
DUKF(1)+RMS	4.3	-2.5	0.9	1.1	1.3	1.7	1.3
DUKF(1)+COSH	4.6	-2.5	0.9	1.1	1.3	1.7	1.2
DUKF(1)+MSE	4.5	-2.2	1.0	1.1	1.3	1.7	1.3

Table 90: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
DKF+YW	8.7	-0.2	1.0	1.2	1.7	1.8	1.4
DKF+IS	8.7	0.0	1.0	1.3	1.7	1.9	1.5
DKF+RMS	8.8	0.0	1.0	1.3	1.7	1.8	1.5
DKF+COSH	8.8	-0.1	1.0	1.3	1.7	1.9	1.5
DKF+MSE	8.7	-0.1	1.0	1.3	1.7	1.9	1.5
KEM_{Burg}+YW	8.2	-0.5	1.0	1.3	1.9	1.8	1.6
KEM_{Burg}+IS	8.3	-0.4	1.0	1.3	1.9	1.9	1.7
KEM_{Burg}+RMS	8.3	-0.4	1.0	1.3	1.9	2.0	1.7
KEM_{Burg}+COSH	8.4	-0.4	1.0	1.3	1.9	1.9	1.7
KEM_{Burg}+MSE	8.5	-0.4	1.0	1.3	1.9	2.0	1.7
DUKF(1)+YW	8.3	-0.4	1.0	1.4	1.8	1.8	1.6
DUKF(1)+IS	8.5	-0.3	1.0	1.4	1.8	1.9	1.7
DUKF(1)+RMS	8.5	-0.3	1.0	1.4	1.8	2.0	1.7
DUKF(1)+COSH	8.5	-0.1	1.0	1.4	1.8	2.0	1.7
DUKF(1)+MSE	8.6	-0.3	1.0	1.4	1.8	2.0	1.7

Table 91: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
DKF+YW	11.8	1.8	1.0	1.5	2.1	2.1	1.8
DKF+IS	12.0	1.9	1.0	1.5	2.1	2.2	1.9
DKF+RMS	12.0	1.9	1.0	1.5	2.2	2.2	1.9
DKF+COSH	12.0	1.9	1.0	1.5	2.1	2.2	1.9
DKF+MSE	12.0	1.9	1.0	1.5	2.2	2.2	1.9
KEM_{Burg}+YW	11.7	1.7	1.0	1.6	2.3	2.2	2.0
KEM_{Burg}+IS	11.9	1.9	1.0	1.6	2.4	2.3	2.1
KEM_{Burg}+RMS	11.9	1.9	1.0	1.6	2.4	2.3	2.1
KEM_{Burg}+COSH	12.0	1.9	1.0	1.6	2.3	2.3	2.1
KEM_{Burg}+MSE	11.9	1.9	1.0	1.6	2.4	2.3	2.1
DUKF(1)+YW	10.5	1.3	1.0	1.6	2.3	2.2	2.0
DUKF(1)+IS	10.7	1.4	1.0	1.6	2.3	2.3	2.1
DUKF(1)+RMS	10.6	1.5	1.0	1.7	2.3	2.3	2.1
DUKF(1)+COSH	10.8	1.5	1.0	1.7	2.3	2.3	2.1
DUKF(1)+MSE	10.9	1.5	1.0	1.7	2.3	2.4	2.1

Table 92: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in H-FAC conditions.

B.5.3. Military vehicle noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
DKF+YW	-3.9	-6.3	0.2	1.0	1.0	1.0	0.8
DKF+IS	-3.4	-6.2	0.2	1.0	1.0	1.1	0.9
DKF+RMS	-3.4	-6.1	0.2	1.0	1.0	1.0	0.9
DKF+COSH	-3.4	-6.1	0.2	1.0	1.0	1.0	0.9
DKF+MSE	-3.3	-6.1	0.2	1.0	1.0	1.1	0.9
KEM_{Burg}+YW	-2.9	-6.5	0.1	1.0	1.0	1.0	0.8
KEM_{Burg}+IS	-2.5	-6.2	0.1	1.0	1.0	1.1	0.9
KEM_{Burg}+RMS	-2.6	-6.4	0.1	1.0	1.0	1.0	0.9
KEM_{Burg}+COSH	-2.5	-6.2	0.1	1.1	1.0	1.1	0.9
KEM_{Burg}+MSE	-2.4	-6.2	0.1	1.0	1.0	1.1	0.9
DUKF(1)+YW	-0.8	-5.7	0.1	1.0	1.0	1.0	0.9
DUKF(1)+IS	-0.4	-5.5	0.2	1.0	1.0	1.1	0.9
DUKF(1)+RMS	-0.6	-5.6	0.1	1.1	1.0	1.1	1.0
DUKF(1)+COSH	-0.5	-5.5	0.3	1.0	1.0	1.1	1.0
DUKF(1)+MSE	-0.4	-5.4	0.2	1.1	1.0	1.2	1.0

Table 93: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
DKF+YW	3.9	-2.5	0.8	1.1	1.5	1.4	1.2
DKF+IS	4.1	-2.3	0.8	1.1	1.5	1.4	1.2
DKF+RMS	4.0	-2.3	0.8	1.1	1.5	1.5	1.2
DKF+COSH	4.1	-2.3	0.8	1.1	1.5	1.5	1.2
DKF+MSE	4.3	-2.3	0.9	1.1	1.5	1.5	1.3
KEM_{Burg}+YW	4.6	-2.5	0.8	1.1	1.5	1.4	1.2
KEM_{Burg}+IS	4.9	-2.3	0.9	1.1	1.6	1.5	1.3
KEM_{Burg}+RMS	4.8	-2.4	0.8	1.1	1.6	1.5	1.3
KEM_{Burg}+COSH	4.7	-2.3	0.8	1.1	1.6	1.6	1.2
KEM_{Burg}+MSE	5.2	-2.1	0.9	1.1	1.6	1.6	1.3
DUKF(1)+YW	5.4	-1.9	1.0	1.1	1.5	1.5	1.2
DUKF(1)+IS	6.0	-1.8	1.0	1.1	1.5	1.6	1.3
DUKF(1)+RMS	6.1	-1.7	1.0	1.1	1.6	1.5	1.3
DUKF(1)+COSH	6.2	-1.7	1.0	1.1	1.6	1.6	1.3
DUKF(1)+MSE	6.2	-1.7	1.0	1.1	1.6	1.6	1.3

Table 94: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
DKF+YW	9.0	0.4	1.0	1.2	1.9	1.7	1.5
DKF+IS	9.0	0.5	1.0	1.2	1.9	1.9	1.6
DKF+RMS	9.1	0.6	1.0	1.2	1.9	1.9	1.6
DKF+COSH	9.3	0.6	1.0	1.2	1.9	1.9	1.6
DKF+MSE	9.3	0.7	1.0	1.2	1.9	1.8	1.6
KEM_{Burg}+YW	10.4	1.3	1.0	1.2	1.7	1.8	1.4
KEM_{Burg}+IS	10.5	1.4	1.0	1.3	2.0	2.0	1.7
KEM_{Burg}+RMS	10.4	1.4	1.0	1.3	2.0	2.0	1.7
KEM_{Burg}+COSH	10.7	1.4	1.0	1.2	2.0	2.0	1.8
KEM_{Burg}+MSE	10.9	1.5	1.0	1.3	2.0	2.0	1.7
DUKF(1)+YW	9.1	0.4	1.0	1.3	2.0	1.8	1.6
DUKF(1)+IS	9.3	0.6	1.0	1.3	2.0	1.9	1.7
DUKF(1)+RMS	9.2	0.6	1.0	1.3	2.0	2.0	1.7
DUKF(1)+COSH	9.2	0.5	1.0	1.3	2.0	2.0	1.8
DUKF(1)+MSE	9.2	0.6	1.0	1.3	2.0	2.0	1.8

Table 95: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
DKF+YW	13.6	3.3	1.0	1.5	2.3	2.2	1.9
DKF+IS	13.6	3.4	1.0	1.5	2.4	2.3	2.0
DKF+RMS	13.6	3.4	1.0	1.5	2.3	2.3	2.0
DKF+COSH	13.7	3.5	1.0	1.5	2.3	2.3	2.0
DKF+MSE	13.8	3.5	1.0	1.5	2.3	2.3	2.0
KEM_{Burg}+YW	14.1	3.6	1.0	1.6	2.5	2.3	2.1
KEM_{Burg}+IS	14.2	3.7	1.0	1.6	2.5	2.4	2.2
KEM_{Burg}+RMS	14.2	3.8	1.0	1.6	2.5	2.4	2.2
KEM_{Burg}+COSH	14.4	3.8	1.0	1.6	2.5	2.4	2.2
KEM_{Burg}+MSE	14.5	3.8	1.0	1.6	2.5	2.4	2.2
DUKF(1)+YW	12.5	3.4	1.0	1.6	2.3	2.2	1.9
DUKF(1)+IS	12.5	3.5	1.0	1.6	2.3	2.3	2.0
DUKF(1)+RMS	12.6	3.5	1.0	1.6	2.3	2.3	1.9
DUKF(1)+COSH	12.6	3.5	1.0	1.6	2.3	2.3	2.0
DUKF(1)+MSE	12.6	3.6	1.0	1.6	2.4	2.2	2.0

Table 96: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in H-MIL conditions.

B.5.4. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
DKF+YW	-8.0	-6.7	1.0	1.1	1.9	1.3	1.5
DKF+IS	-7.8	-6.5	1.0	1.1	1.9	1.4	1.6
DKF+RMS	-7.9	-6.5	1.0	1.1	1.9	1.4	1.6
DKF+COSH	-7.7	-6.4	1.0	1.1	1.9	1.4	1.5
DKF+MSE	-7.5	-6.3	1.0	1.1	1.9	1.4	1.6
KEM_{Burg}+YW	-8.0	-7.1	1.0	1.1	1.9	1.3	1.5
KEM_{Burg}+IS	-7.6	-6.9	1.0	1.1	1.9	1.4	1.6
KEM_{Burg}+RMS	-7.1	-6.1	1.0	1.1	1.9	1.4	1.6
KEM_{Burg}+COSH	-7.6	-6.9	1.0	1.1	1.9	1.4	1.6
KEM_{Burg}+MSE	-7.3	-6.7	1.0	1.1	1.9	1.4	1.6
DUKF(1)+YW	-5.1	-6.3	1.0	1.1	1.9	1.4	1.5
DUKF(1)+IS	-5.1	-6.1	1.0	1.1	1.9	1.4	1.6
DUKF(1)+RMS	-5.1	-6.2	1.0	1.1	1.9	1.5	1.6
DUKF(1)+COSH	-5.1	-6.1	1.0	1.1	1.9	1.5	1.6
DUKF(1)+MSE	-5.0	-6.1	1.0	1.1	1.9	1.5	1.6

Table 97: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
DKF+YW	-1.5	-3.9	1.0	1.2	2.4	1.7	1.9
DKF+IS	-1.4	-3.8	1.0	1.2	2.4	1.8	2.0
DKF+RMS	-1.5	-3.7	1.0	1.2	2.4	1.8	2.0
DKF+COSH	-1.3	-3.7	1.0	1.2	2.4	1.8	2.0
DKF+MSE	-1.4	-3.7	1.0	1.2	2.4	1.9	2.0
KEM_{Burg}+YW	-1.4	-4.2	1.0	1.3	2.4	1.7	2.0
KEM_{Burg}+IS	-1.2	-4.0	1.0	1.3	2.4	1.8	2.0
KEM_{Burg}+RMS	-1.3	-3.8	1.0	1.3	2.4	1.8	2.0
KEM_{Burg}+COSH	-1.3	-3.8	1.0	1.3	2.5	1.9	2.1
KEM_{Burg}+MSE	-1.2	-3.8	1.0	1.3	2.5	1.9	2.1
DUKF(1)+YW	1.4	-3.2	1.0	1.3	2.5	1.8	2.0
DUKF(1)+IS	1.4	-3.0	1.0	1.4	2.5	1.9	2.1
DUKF(1)+RMS	1.4	-3.0	1.0	1.4	2.5	1.9	2.1
DUKF(1)+COSH	1.4	-3.1	1.0	1.3	2.5	1.9	2.1
DUKF(1)+MSE	1.4	-2.1	1.0	1.4	2.5	1.9	2.1

Table 98: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
DKF+YW	6.6	0.4	1.0	1.6	3.0	2.3	2.5
DKF+IS	6.8	0.5	1.0	1.7	3.0	2.4	2.5
DKF+RMS	6.7	0.5	1.0	1.7	3.0	2.4	2.6
DKF+COSH	6.9	0.6	1.0	1.7	3.0	2.4	2.6
DKF+MSE	6.8	0.6	1.0	1.7	3.0	2.4	2.6
KEM_{Burg}+YW	6.7	0.3	1.0	1.9	3.0	2.3	2.6
KEM_{Burg}+IS	6.9	0.5	1.0	1.9	3.0	2.4	2.7
KEM_{Burg}+RMS	6.8	0.5	1.0	1.9	3.1	2.4	2.7
KEM_{Burg}+COSH	6.8	0.5	1.0	1.9	3.1	2.4	2.7
KEM_{Burg}+MSE	6.9	0.6	1.0	1.9	3.1	2.4	2.7
DUKF(1)+YW	8.0	0.4	1.0	2.0	3.1	2.4	2.7
DUKF(1)+IS	8.2	0.5	1.0	2.0	3.1	2.5	2.8
DUKF(1)+RMS	8.1	0.6	1.0	2.0	3.1	2.5	2.8
DUKF(1)+COSH	8.1	0.5	1.0	2.0	3.1	2.5	2.7
DUKF(1)+MSE	8.3	0.6	1.0	2.0	3.2	2.5	2.8

Table 99: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
DKF+YW	11.9	3.2	1.0	2.1	3.3	2.7	2.8
DKF+IS	11.9	3.3	1.0	2.1	3.3	2.7	2.9
DKF+RMS	12.2	3.6	1.0	2.2	3.4	2.8	2.9
DKF+COSH	12.1	3.5	1.0	2.2	3.4	2.8	2.9
DKF+MSE	12.1	3.6	1.0	2.2	3.4	2.7	2.9
KEM_{Burg}+YW	12.2	3.4	1.0	2.5	3.5	2.8	3.1
KEM_{Burg}+IS	12.3	3.5	1.0	2.5	3.6	2.9	3.2
KEM_{Burg}+RMS	12.6	3.9	1.0	2.5	3.6	2.9	3.2
KEM_{Burg}+COSH	12.7	4.0	1.0	2.5	3.6	3.0	3.2
KEM_{Burg}+MSE	12.6	3.7	1.0	2.6	3.6	3.0	3.2
DUKF(1)+YW	11.2	3.2	1.0	2.4	3.2	2.5	2.7
DUKF(1)+IS	11.3	3.3	1.0	2.4	3.2	2.7	2.7
DUKF(1)+RMS	11.7	3.3	1.0	2.4	3.3	2.7	2.8
DUKF(1)+COSH	12.0	3.3	1.0	2.6	3.6	2.9	3.2
DUKF(1)+MSE	11.9	3.4	1.0	2.5	3.3	2.7	2.8

Table 100: Fullband colored noise enhancement with different types of noise PSD autoregressive modelling, in H-CAR conditions.

B.6. Tables for Section 8.6.1 (Subband processing)

B.6.1. Cafeteria noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
1B-DKF	-1.5	-5.7	0.2	1.0	0.3	0.9	0.5
4B-DKF	0.2	-5.2	0.2	1.0	0.3	1.0	0.5
32B-DKF	0.7	-4.6	0.3	1.1	0.4	1.0	0.5
32B-DKF-Post	3.0	-3.6	0.3	1.1	0.5	1.1	0.6
Ψ-32B-DKF	3.3	-2.5	0.3	1.1	0.5	1.1	0.6
Ψ-32B-DKF-Post	3.4	-2.7	0.3	1.1	0.6	1.2	0.6
1B-KEM_{Burg}	-2.4	-6.1	0.2	1.0	0.3	0.9	0.5
4B-KEM_{Burg}	-2.3	-5.7	0.2	1.0	0.4	0.9	0.5
32B-KEM_{Burg}	0.8	-4.5	0.2	1.1	0.4	1.0	0.6
32B-KEM_{Burg}-Post	3.2	-3.4	0.3	1.1	0.5	1.1	0.6
Ψ-32B-KEM_{Burg}	2.5	-3.9	0.2	1.1	0.5	1.1	0.6
Ψ-32B-KEM_{Burg}-Post	3.1	-3.2	0.3	1.1	0.5	1.1	0.6
1B-RBPF	-1.0	-5.8	0.1	1.0	0.3	1.0	0.4
4B-RBPF	-0.8	-5.6	0.2	1.0	0.3	0.9	0.4
32B-RBPF	1.7	-4.2	0.2	1.1	0.4	1.0	0.5
32B-$RBPF_{(Standalone)}$	0.1	-4.9	0.2	1.1	0.2	1.0	0.4
32B-RBPF-Post	1.8	-4.0	0.2	1.1	0.4	1.0	0.5
Ψ-32B-RBPF	1.9	-4.2	0.2	1.1	0.4	1.1	0.5
Ψ-32B-RBPF-Post	3.1	-3.0	0.4	1.1	0.4	1.1	0.5
Ψ-32B-DEKF4-Post	-3.3	-6.2	0.1	1.0	0.4	0.9	0.5
Ψ-32B-DUKF1-Post	-0.3	-4.7	0.3	1.0	0.4	1.0	0.5
Ψ-32B-KEM-Post	1.6	-3.6	0.3	1.1	0.5	1.1	0.5
Ψ-32B-NPF-Post	3.2	-2.8	0.4	1.1	0.5	1.2	0.6

Table 101: Comparison between various subband methods introduced in Chapter VII, for VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
1B-DKF	4.3	-2.7	0.4	1.1	0.8	1.3	0.8
4B-DKF	4.7	-2.7	0.4	1.1	0.8	1.3	0.8
32B-DKF	6.2	-1.6	0.5	1.2	0.9	1.4	0.9
32B-DKF-Post	6.9	-1.4	0.6	1.2	1.1	1.4	1.0
Ψ-32B-DKF	6.1	-0.6	0.6	1.3	1.0	1.5	1.0
Ψ-32B-DKF-Post	6.6	-1.1	0.7	1.3	1.1	1.5	1.0
1B-KEM_{Burg}	3.9	-2.9	0.4	1.1	0.9	1.3	0.8
4B-KEM_{Burg}	4.1	-2.5	0.5	1.1	0.9	1.3	0.8
32B-KEM_{Burg}	6.1	-1.6	0.6	1.2	0.9	1.4	0.9
32B-KEM_{Burg}-Post	6.8	-1.5	0.6	1.1	1.0	1.4	1.0
Ψ-32B-KEM_{Burg}	6.6	-1.4	0.6	1.2	1.0	1.4	1.0
Ψ-32B-KEM_{Burg}-Post	6.6	-1.5	0.7	1.2	1.0	1.4	1.0
1B-RBPF	4.4	-2.9	0.4	1.1	0.8	1.3	0.8
4B-RBPF	4.8	-2.6	0.5	1.1	0.8	1.3	0.8
32B-RBPF	4.8	-1.5	0.6	1.2	0.9	1.4	0.9
32B-$RBPF_{(Standalone)}$	3.8	-2.5	0.5	1.1	0.8	1.3	0.8
32B-RBPF-Post	6.8	-1.1	0.6	1.2	1.0	1.4	1.0
Ψ-32B-RBPF	6.7	-1.4	0.7	1.2	0.9	1.4	0.9
Ψ-32B-RBPF-Post	6.9	-0.7	0.8	1.2	1.0	1.5	0.9
Ψ-32B-DEKF-Post	2.8	-3.4	0.4	1.1	0.9	1.3	0.9
Ψ-32B-DUKF-Post	4.2	-2.9	0.4	1.1	1.0	1.3	0.9
Ψ-32B-KEM-Post	2.4	-2.4	0.7	1.1	1.0	1.3	0.9
Ψ-32B-NPF-Post	7.1	-0.3	0.8	1.3	1.0	1.5	1.0

Table 102: Comparison between various subband methods introduced in Chapter VII, for L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
1B-DKF	9.4	0.4	0.8	1.2	1.3	1.7	1.2
4B-DKF	9.5	0.3	0.9	1.3	1.4	1.7	1.2
32B-DKF	10.4	1.2	0.9	1.4	1.4	1.8	1.4
32B-DKF-Post	11.0	1.3	0.9	1.4	1.6	1.8	1.4
Ψ-32B-DKF	9.4	1.4	0.9	1.4	1.6	1.8	1.4
Ψ-32B-DKF-Post	10.6	1.5	0.9	1.5	1.6	1.8	1.5
1B-KEM_{Burg}	9.4	0.6	0.9	1.3	1.4	1.7	1.3
4B-KEM_{Burg}	10.1	1.1	0.9	1.3	1.5	1.8	1.3
32B-KEM_{Burg}	10.7	1.3	0.9	1.4	1.5	1.8	1.3
32B-KEM_{Burg}-Post	10.8	1.3	1.0	1.4	1.5	1.8	1.3
Ψ-32B-KEM_{Burg}	10.2	1.0	1.0	1.4	1.5	1.8	1.4
Ψ-32B-KEM_{Burg}-Post	10.6	1.2	1.0	1.4	1.5	1.8	1.4
1B-RBPF	9.6	0.3	0.9	1.2	1.4	1.7	1.2
4B-RBPF	9.3	-0.1	0.9	1.3	1.3	1.7	1.2
32B-RBPF	9.9	1.0	0.9	1.3	1.4	1.8	1.3
32B-$RBPF_{(Standalone)}$	10.0	1.1	0.9	1.3	1.3	1.8	1.2
32B-RBPF-Post	10.9	1.6	0.9	1.4	1.5	1.8	1.4
Ψ-32B-RBPF	10.7	1.2	0.9	1.4	1.4	1.8	1.4
Ψ-32B-RBPF-Post	11.0	1.6	0.9	1.5	1.5	1.9	1.4
Ψ-32B-DEKF4-Post	10.8	1.0	0.9	1.4	1.6	1.8	1.4
Ψ-32B-DUKF-Post	10.6	1.0	0.9	1.4	1.6	1.8	1.4
Ψ-32B-KEM-Post	9.7	0.3	0.9	1.3	1.5	1.7	1.3
Ψ-32B-NPF-Post	11.0	1.8	0.9	1.5	1.6	1.8	1.4

Table 103: Comparison between various subband methods introduced in Chapter VII, for M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
1B-DKF	12.7	2.6	1.0	1.4	1.7	2.0	1.5
4B-DKF	12.3	2.2	0.9	1.5	1.7	2.0	1.5
32B-DKF	12.9	3.3	1.0	1.7	1.8	2.2	1.7
32B-DKF-Post	13.9	3.8	1.0	1.7	2.0	2.2	1.8
Ψ-32B-DKF	13.8	3.6	1.0	1.7	2.0	2.2	1.8
Ψ-32B-DKF-Post	14.7	4.1	1.0	1.8	2.0	2.2	1.8
1B-KEM_{Burg}	12.5	2.8	1.0	1.5	1.9	2.1	1.7
4B-KEM_{Burg}	13.7	3.5	1.0	1.7	2.0	2.2	1.8
32B-KEM_{Burg}	13.6	3.4	1.0	1.6	1.8	2.1	1.6
32B-KEM_{Burg}-Post	14.1	3.8	1.0	1.6	2.0	2.2	1.7
Ψ-32B-KEM_{Burg}	13.1	3.2	1.0	1.6	1.9	2.1	1.7
Ψ-32B-KEM_{Burg}-Post	14.2	3.5	1.0	1.6	2.0	2.2	1.8
1B-RBPF	13.6	3.5	1.0	1.5	2.4	2.3	2.0
4B-RBPF	12.2	2.5	1.0	1.5	1.9	2.0	1.7
32B-RBPF	13.0	2.9	1.0	1.7	2.0	2.2	2.0
32B-$RBPF_{(Standalone)}$	13.4	3.2	1.0	1.6	1.8	2.2	1.8
32B-RBPF-Post	13.3	3.1	1.0	1.7	2.0	2.2	1.9
Ψ-32B-RBPF	13.0	3.0	1.0	1.7	1.9	2.2	1.7
Ψ-32B-RBPF-Post	13.1	3.4	1.0	1.7	2.0	2.2	1.8
Ψ-32B-DEKF-Post	14.5	3.9	1.0	1.7	2.0	2.3	1.8
Ψ-32B-DUKF-Post	13.9	3.6	1.0	1.7	2.0	2.2	1.8
Ψ-32B-KEM-Post	12.4	2.3	1.0	1.5	1.9	2.0	1.7
Ψ-32B-NPF-Post	13.6	3.6	1.0	1.8	2.1	2.2	1.9

Table 104: Comparison between various subband methods introduced in Chapter VII, for H-CAF conditions.

B.6.2. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
1B-DKF	-1.4	-5.8	0.2	1.0	0.7	1.1	0.7
4B-DKF	-0.4	-5.6	0.6	1.0	0.6	1.1	0.7
32B-DKF	0.9	-5.1	0.8	1.0	0.8	1.2	0.8
32B-DKF-Post	1.5	-4.2	0.7	1.0	0.9	1.2	0.8
Ψ-32B-DKF	-0.5	-5.5	0.8	1.0	0.9	1.1	0.9
Ψ-32B-DKF-Post	1.1	-4.4	0.8	1.0	0.8	1.2	0.8
1B-KEM_{Burg}	-2.8	-6.4	0.3	1.0	0.8	1.1	0.8
4B-KEM_{Burg}	-0.6	-5.5	0.7	1.0	0.8	1.1	0.8
32B-KEM_{Burg}	0.8	-5.1	0.8	1.1	0.8	1.2	0.8
32B-KEM_{Burg}-Post	1.8	-4.1	0.8	1.1	0.9	1.2	0.8
Ψ-32B-KEM_{Burg}	0.7	-5.1	0.8	1.1	0.8	1.2	0.8
Ψ-32B-KEM_{Burg}-Post	1.6	-4.2	0.8	1.0	0.8	1.2	0.9
1B-RBPF	-0.3	-5.7	0.3	1.0	0.7	1.1	0.7
4B-RBPF	-2.7	-6.3	0.7	1.0	0.6	1.0	0.6
32B-RBPF	0.6	-5.2	0.8	1.0	0.8	1.2	0.8
32B-$RBPF_{(Standalone)}$	-2.7	-5.7	0.6	1.0	0.7	1.0	0.7
32B-RBPF-Post	0.7	-5.1	0.8	1.0	0.8	1.2	0.8
Ψ-32B-RBPF	0.8	-4.9	0.8	1.1	0.9	1.2	0.8
Ψ-32B-RBPF-Post	0.8	-4.6	0.8	1.0	0.9	1.2	0.9
Ψ-32B-DEKF-Post	1.8	-4.2	0.6	1.0	0.9	1.2	0.9
Ψ-32B-DUKF-Post	0.7	-4.8	0.4	1.0	0.8	1.1	0.8
Ψ-32B-KEM-Post	0.9	-4.5	0.5	1.0	0.8	1.2	0.8
Ψ-32B-NPF-Post	1.0	-4.3	0.9	1.0	0.9	1.1	0.9

Table 105: Comparison between various subband methods introduced in Chapter VII, for VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
1B-DKF	5.5	-2.1	0.8	1.1	1.3	1.5	1.2
4B-DKF	6.6	-1.8	0.8	1.2	1.3	1.5	1.1
32B-DKF	6.5	-1.7	0.9	1.2	1.5	1.6	1.3
32B-DKF-Post	6.6	-1.6	0.9	1.3	1.7	1.7	1.4
Ψ-32B-DKF	5.9	-1.9	0.9	1.3	1.6	1.7	1.4
Ψ-32B-DKF-Post	6.6	-1.5	1.0	1.3	1.7	1.7	1.5
1B-KEM_{Burg}	4.7	-2.6	0.9	1.2	1.5	1.5	1.3
4B-KEM_{Burg}	6.3	-1.7	0.9	1.2	1.5	1.6	1.3
32B-KEM_{Burg}	6.7	-1.6	0.9	1.3	1.5	1.6	1.4
32B-KEM_{Burg}-Post	6.7	-1.5	0.9	1.3	1.6	1.7	1.4
Ψ-32B-KEM_{Burg}	6.4	-1.7	0.9	1.3	1.6	1.6	1.4
Ψ-32B-KEM_{Burg}-Post	6.2	-1.7	0.9	1.3	1.6	1.6	1.4
1B-RBPF	5.7	-2.4	0.9	1.2	1.4	1.6	1.2
4B-RBPF	4.3	-3.0	0.8	1.1	1.3	1.4	1.1
32B-RBPF	6.4	-1.7	0.9	1.3	1.6	1.6	1.4
32B-$RBPF_{(Standalone)}$	4.2	-2.8	0.9	1.2	1.5	1.5	1.3
32B-RBPF-Post	6.6	-1.7	0.9	1.3	1.6	1.7	1.4
Ψ-32B-RBPF	6.4	-1.4	0.9	1.3	1.6	1.7	1.4
Ψ-32B-RBPF-Post	6.6	-1.1	0.9	1.3	1.6	1.7	1.5
Ψ-32B-DEKF-Post	6.1	-1.8	0.9	1.3	1.6	1.6	1.4
Ψ-32B-DUKF-Post	4.6	-2.4	0.9	1.2	1.6	1.6	1.4
Ψ-32B-KEM-Post	4.0	-2.9	0.9	1.1	1.5	1.5	1.4
Ψ-32B-NPF-Post	6.8	-0.9	1.0	1.3	1.6	1.7	1.4

Table 106: Comparison between various subband methods introduced in Chapter VII, for L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
1B-DKF	8.7	-0.2	1.0	1.2	1.7	1.8	1.4
4B-DKF	9.3	-0.2	0.9	1.3	1.6	1.7	1.4
32B-DKF	8.7	-0.1	1.0	1.5	1.9	1.9	1.7
32B-DKF-Post	9.2	0.2	1.0	1.5	2.1	2.0	1.8
Ψ-32B-DKF	8.7	-0.1	1.0	1.5	2.0	1.9	1.8
Ψ-32B-DKF-Post	9.1	0.1	1.0	1.5	2.1	2.0	1.8
1B-KEM_{Burg}	8.2	-0.5	1.0	1.3	1.9	1.8	1.6
4B-KEM_{Burg}	9.5	0.3	1.0	1.4	1.9	1.9	1.7
32B-KEM_{Burg}	9.0	0.0	1.0	1.5	1.9	1.9	1.7
32B-KEM_{Burg}-Post	9.3	0.1	1.0	1.5	2.1	1.9	1.8
Ψ-32B-KEM_{Burg}	8.5	-0.1	1.0	1.5	1.9	1.9	1.7
Ψ-32B-KEM_{Burg}-Post	8.8	0.0	1.0	1.5	2.0	2.0	1.8
1B-RBPF	8.7	-0.5	1.0	1.3	1.8	1.8	1.5
4B-RBPF	7.0	-1.2	0.9	1.3	1.7	1.7	1.5
32B-RBPF	8.8	-0.1	1.0	1.5	2.0	1.9	1.7
32B-$RBPF_{(Standalone)}$	7.0	-1.3	1.0	1.3	1.9	1.8	1.7
32B-RBPF-Post	9.1	0.3	1.0	1.5	2.0	1.9	1.7
Ψ-32B-RBPF	8.9	0.2	1.0	1.5	2.0	1.9	1.8
Ψ-32B-RBPF-Post	10.1	0.9	1.0	1.6	2.1	2.0	1.8
Ψ-32B-DEKF-Post	8.7	-0.3	1.0	1.5	2.0	1.9	1.8
Ψ-32B-DUKF-Post	8.8	-0.8	1.0	1.4	2.0	1.9	1.7
Ψ-32B-KEM-Post	7.6	-0.9	1.0	1.4	1.9	1.8	1.6
Ψ-32B-NPF-Post	9.7	0.7	1.0	1.6	2.1	2.0	1.8

Table 107: Comparison between various subband methods introduced in Chapter VII, for M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
1B-DKF	11.8	1.8	1.0	1.5	2.1	2.1	1.8
4B-DKF	11.7	1.7	1.0	1.5	1.9	2.0	1.7
32B-DKF	10.6	1.5	1.0	1.8	2.2	2.2	2.0
32B-DKF-Post	12.2	2.4	1.0	1.8	2.5	2.3	2.2
Ψ-32B-DKF	11.6	2.1	1.0	1.8	2.4	2.3	2.1
Ψ-32B-DKF-Post	12.8	2.8	1.0	1.8	2.5	2.3	2.2
1B-KEM_{Burg}	11.7	1.7	1.0	1.6	2.3	2.2	2.0
4B-KEM_{Burg}	12.6	2.4	1.0	1.8	2.3	2.3	2.1
32B-KEM_{Burg}	11.4	1.7	1.0	1.7	2.2	2.2	2.0
32B-KEM_{Burg}-Post	12.5	2.4	1.0	1.8	2.5	2.3	2.1
Ψ-32B-KEM_{Burg}	11.2	1.4	1.0	1.7	2.3	2.2	2.1
Ψ-32B-KEM_{Burg}-Post	11.3	1.9	1.0	1.8	2.4	2.3	2.1
1B-RBPF	11.8	1.6	1.0	1.6	2.2	2.1	1.9
4B-RBPF	8.6	0.3	1.0	1.5	2.1	2.0	1.8
32B-RBPF	11.0	1.5	1.0	1.8	2.4	2.2	2.1
32B-$RBPF_{(Standalone)}$	9.8	0.9	1.0	1.6	2.3	2.1	2.0
32B-RBPF-Post	12.6	2.8	1.0	1.8	2.4	2.3	2.1
Ψ-32B-RBPF	13.5	3.9	1.0	2.0	3.0	2.6	3.1
Ψ-32B-RBPF-Post	13.9	4.0	1.0	2.1	3.1	2.7	3.2
Ψ-32B-DEKF-Post	13.0	2.1	1.0	1.8	2.5	2.3	2.3
Ψ-32B-DUKF-Post	10.9	1.7	1.0	1.8	2.4	2.2	2.1
Ψ-32B-KEM-Post	11.7	0.9	1.0	1.7	2.4	1.2	2.2
Ψ-32B-NPF-Post	14.0	4.1	1.0	2.0	3.1	2.4	3.0

Table 108: Comparison between various subband methods introduced in Chapter VII, for H-FAC conditions.

B.6.3. Military noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
1B-DKF	-3.9	-6.3	0.2	1.0	1.0	1.0	0.8
4B-DKF	0.2	-5.1	0.4	1.0	0.9	1.1	0.8
32B-DKF	1.5	-4.6	0.5	1.0	1.0	1.1	0.9
32B-DKF-Post	2.6	-3.7	0.9	1.0	1.1	1.2	0.9
Ψ-32B-DKF	1.3	-4.1	0.5	1.1	1.1	1.1	0.9
Ψ-32B-DKF-Post	2.3	-3.8	0.9	1.0	1.1	1.2	0.9
1B-KEM$_{Burg}$	-2.9	-6.5	0.1	1.0	1.0	1.0	0.8
4B-KEM$_{Burg}$	-0.4	-5.3	0.3	1.0	1.0	1.1	0.9
32B-KEM$_{Burg}$	1.2	-4.6	0.5	1.1	1.0	1.1	0.9
32B-KEM$_{Burg}$-Post	2.7	-3.7	0.8	1.1	1.1	1.2	0.9
Ψ-32B-KEM$_{Burg}$	1.4	-4.5	0.5	1.1	1.0	1.1	0.9
Ψ-32B-KEM$_{Burg}$-Post	2.6	-3.8	0.9	1.1	1.1	1.2	1.0
1B-RBPF	0.2	-5.4	0.3	1.0	1.0	1.1	0.8
4B-RBPF	-1.3	-5.8	0.4	1.0	0.9	1.0	0.8
32B-RBPF	1.4	-4.6	0.5	1.0	1.1	1.1	0.9
32B-RBPF$_{(Standalone)}$	-2.3	-5.8	0.3	1.0	1.0	1.1	0.9
32B-RBPF-Post	1.4	-4.5	0.5	1.1	1.1	1.2	0.9
Ψ-32B-RBPF	1.4	-4.6	0.5	1.1	1.1	1.2	0.9
Ψ-32B-RBPF-Post	1.4	-4.4	0.5	1.1	1.1	1.2	0.9
Ψ-32B-DEKF-Post	2.5	-3.8	0.8	1.1	1.1	1.2	0.9
Ψ-32B-DUKF-Post	1.3	-4.3	0.4	1.0	1.0	1.1	0.8
Ψ-32B-KEM-Post	1.5	-4.1	0.6	1.1	1.1	1.1	0.9
Ψ-32B-NPF-Post	1.6	-4.0	0.5	1.1	1.1	1.2	1.0

Table 109: Comparison between various subband methods introduced in Chapter VII, for VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
1B-DKF	3.9	-2.5	0.8	1.1	1.5	1.4	1.2
4B-DKF	6.0	-1.9	0.9	1.1	1.4	1.4	1.2
32B-DKF	6.9	-1.2	1.0	1.2	1.6	1.6	1.3
32B-DKF-Post	7.5	-1.1	1.0	1.2	1.7	1.6	1.4
Ψ-32B-DKF	6.0	-1.8	0.9	1.2	1.6	1.5	1.3
Ψ-32B-DKF-Post	7.1	-1.3	1.0	1.2	1.7	1.6	1.4
1B-KEM$_{Burg}$	4.6	-2.5	0.8	1.1	1.5	1.4	1.2
4B-KEM$_{Burg}$	6.6	-1.4	1.0	1.1	1.6	1.5	1.3
32B-KEM$_{Burg}$	6.9	-1.2	1.0	1.2	1.6	1.6	1.3
32B-KEM$_{Burg}$-Post	7.6	-1.1	1.0	1.2	1.7	1.6	1.4
Ψ-32B-KEM$_{Burg}$	7.0	-1.2	1.0	1.2	1.6	1.6	1.3
Ψ-32B-KEM$_{Burg}$-Post	7.1	-1.1	1.0	1.2	1.7	1.6	1.4
1B-RBPF	5.9	-2.1	1.0	1.1	1.5	1.4	1.2
4B-RBPF	4.2	-2.8	0.9	1.1	1.4	1.3	1.2
32B-RBPF	6.8	-1.3	1.0	1.2	1.6	1.5	1.4
32B-RBPF$_{(Standalone)}$	4.0	-2.6	0.8	1.1	1.5	1.5	1.3
32B-RBPF-Post	7.7	-0.7	0.9	1.2	1.6	1.6	1.3
Ψ-32B-RBPF	6.9	-1.3	1.0	1.2	1.7	1.6	1.4
Ψ-32B-RBPF-Post	7.9	-0.6	1.0	1.2	1.6	1.6	1.4
Ψ-32B-DEKF-Post	7.1	-1.4	1.0	1.2	1.7	1.5	1.4
Ψ-32B-DUKF-Post	5.7	-2.1	0.9	1.1	1.6	1.5	1.3
Ψ-32B-KEM-Post	6.6	-1.3	1.0	1.1	1.6	1.5	1.3
Ψ-32B-NPF-Post	7.9	-0.4	1.0	1.2	1.7	1.5	1.4

Table 110: Comparison between various subband methods introduced in Chapter VII, for L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
1B-DKF	9.0	0.4	1.0	1.2	1.9	1.7	1.5
4B-DKF	9.8	0.5	1.0	1.2	1.8	1.8	1.4
32B-DKF	9.4	0.8	1.0	1.4	1.9	1.9	1.7
32B-DKF-Post	10.3	1.1	1.0	1.4	2.1	1.9	1.7
Ψ-32B-DKF	10.9	1.5	1.0	1.4	2.1	1.9	1.7
Ψ-32B-DKF-Post	11.3	1.6	1.0	1.4	2.1	2.0	1.7
1B-KEM_{Burg}	10.4	1.3	1.0	1.2	1.7	1.8	1.4
4B-KEM_{Burg}	11.0	1.4	1.0	1.3	2.0	1.9	1.7
32B-KEM_{Burg}	10.1	1.0	1.0	1.4	2.0	1.9	1.7
32B-KEM_{Burg}-Post	10.4	1.1	1.0	1.4	2.1	1.9	1.8
Ψ-32B-KEM_{Burg}	9.8	0.6	1.0	1.4	2.1	1.9	1.7
Ψ-32B-KEM_{Burg}-Post	10.0	0.9	1.0	1.4	2.1	1.9	1.8
1B-RBPF	10.0	0.5	1.0	1.2	1.9	1.8	1.5
4B-RBPF	7.4	-0.7	1.0	1.2	1.9	1.7	1.5
32B-RBPF	9.9	0.8	1.0	1.4	2.1	1.9	1.7
32B-$RBPF_{(Standalone)}$	8.3	0.2	1.0	1.3	2.0	1.8	1.6
32B-RBPF-Post	11.4	1.8	1.0	1.4	2.0	2.0	1.8
Ψ-32B-RBPF	9.9	0.9	1.0	1.4	2.1	1.9	1.7
Ψ-32B-RBPF-Post	11.4	1.9	1.0	1.4	2.0	2.0	1.9
Ψ-32B-DEKF-Post	10.0	0.9	1.0	1.4	2.0	1.9	1.7
Ψ-32B-DUKF-Post	10.2	0.8	1.0	1.3	2.1	1.9	1.7
Ψ-32B-KEM-Post	8.8	0.3	1.0	1.3	2.1	1.8	1.7
Ψ-32B-NPF-Post	11.2	1.9	1.0	1.4	2.0	2.0	1.9

Table 111: Comparison between various subband methods introduced in Chapter VII, for M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
1B-DKF	13.6	3.3	1.0	1.5	2.3	2.2	1.9
4B-DKF	12.9	2.6	1.0	1.5	2.1	2.1	1.7
32B-DKF	12.8	3.4	1.0	1.7	2.3	2.3	2.0
32B-DKF-Post	14.4	4.3	1.0	1.7	2.5	2.4	2.1
Ψ-32B-DKF	14.5	4.5	1.0	1.8	2.5	2.4	2.1
Ψ-32B-DKF-Post	15.7	5.0	1.0	1.8	2.6	2.4	2.2
1B-KEM_{Burg}	14.1	3.6	1.0	1.6	2.5	2.3	2.1
4B-KEM_{Burg}	14.1	4.0	1.0	1.8	2.5	2.4	2.1
32B-KEM_{Burg}	13.5	3.4	1.0	1.6	2.3	2.2	2.0
32B-KEM_{Burg}-Post	14.5	4.2	1.0	1.6	2.5	2.3	2.1
Ψ-32B-KEM_{Burg}	12.7	2.5	1.0	1.7	2.5	2.2	2.1
Ψ-32B-KEM_{Burg}-Post	12.7	3.4	1.0	1.7	2.5	2.3	2.1
1B-RBPF	13.5	3.6	1.0	2.4	3.5	2.8	3.1
4B-RBPF	12.8	2.5	1.0	1.5	2.7	2.2	2.5
32B-RBPF	12.8	2.9	1.0	1.7	2.7	2.3	2.1
32B-$RBPF_{(Standalone)}$	12.8	3.2	1.0	1.6	2.6	2.3	2.2
32B-RBPF-Post	13.9	3.8	1.0	1.7	2.4	2.4	2.1
Ψ-32B-RBPF	12.9	3.1	1.0	1.7	2.5	2.3	2.1
Ψ-32B-RBPF-Post	14.1	4.1	1.0	1.9	2.9	2.6	2.9
Ψ-32B-DEKF-Post	15.2	4.6	1.0	1.7	2.5	2.4	2.1
Ψ-32B-DUKF-Post	14.8	4.4	1.0	1.7	2.5	2.4	2.2
Ψ-32B-KEM-Post	13.1	3.2	1.0	1.7	2.4	2.3	2.1
Ψ-32B-NPF-Post	14.2	4.0	1.0	1.9	2.9	2.5	2.9

Table 112: Comparison between various subband methods introduced in Chapter VII, for H-MIL conditions.

B.6.4. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
1B-DKF	-8.0	-6.7	1.0	1.1	1.9	1.3	1.5
4B-DKF	-2.3	-5.1	0.9	1.1	1.9	1.5	1.5
32B-DKF	0.6	-4.5	1.0	1.2	2.1	1.6	1.7
32B-DKF-Post	2.0	-3.9	1.0	1.2	2.2	1.6	1.7
Ψ-32B-DKF	-1.1	-5.0	1.0	1.2	2.1	1.5	1.7
Ψ-32B-DKF-Post	-0.4	-4.5	1.0	1.2	2.2	1.6	1.8
1B-KEM$_{Burg}$	-8.0	-7.1	1.0	1.1	1.9	1.3	1.5
4B-KEM$_{Burg}$	-3.7	-5.5	1.0	1.2	2.1	1.5	1.7
32B-KEM$_{Burg}$	-0.6	-4.7	1.0	1.2	2.1	1.6	1.7
32B-KEM$_{Burg}$-Post	1.7	-4.0	1.0	1.2	2.2	1.6	1.8
Ψ-32B-KEM$_{Burg}$	0.5	-4.5	1.0	1.2	2.2	1.6	1.7
Ψ-32B-KEM$_{Burg}$-Post	1.9	-3.9	1.0	1.2	2.2	1.6	1.8
1B-RBPF	0.1	-4.8	1.0	1.1	2.0	1.5	1.6
4B-RBPF	-0.5	-5.1	1.0	1.1	1.8	1.4	1.5
32B-RBPF	-2.5	-5.3	1.0	1.2	2.1	1.5	1.7
32B-RBPF$_{(Standalone)}$	-5.3	-6.1	1.0	1.1	2.0	1.4	1.6
32B-RBPF-Post	-2.1	-5.2	1.0	1.2	2.1	1.5	1.8
Ψ-32B-RBPF	-2.7	-5.2	1.0	1.2	2.1	1.5	1.7
Ψ-32B-RBPF-Post	-2.2	-4.2	1.0	1.2	2.1	1.5	1.8
Ψ-32B-DEKF-Post	-1.1	-4.6	1.0	1.1	2.1	1.5	1.6
Ψ-32B-DUKF-Post	0.6	-4.4	1.0	1.1	2.1	1.6	1.7
Ψ-32B-KEM-Post	-0.8	-4.7	1.0	1.1	2.1	1.5	1.7
Ψ-32B-NPF-Post	-3.0	-4.8	1.0	1.2	2.2	1.5	1.8

Table 113: Comparison between various subband methods introduced in Chapter VII, for VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
1B-DKF	-1.5	-3.9	1.0	1.2	2.4	1.7	1.9
4B-DKF	2.5	-2.7	1.0	1.3	2.2	1.7	1.9
32B-DKF	5.4	-2.0	1.0	1.5	2.5	1.9	2.1
32B-DKF-Post	5.8	-1.9	1.0	1.5	2.6	2.0	2.2
Ψ-32B-DKF	5.6	-1.7	1.0	1.5	2.5	1.7	2.1
Ψ-32B-DKF-Post	5.7	-1.5	1.0	1.5	2.6	2.0	2.2
1B-KEM$_{Burg}$	-1.6	-3.9	1.0	1.2	2.4	1.7	1.9
4B-KEM$_{Burg}$	2.2	-2.5	1.0	1.5	2.6	1.9	2.1
32B-KEM$_{Burg}$	4.6	-2.2	1.0	1.5	2.5	1.9	2.1
32B-KEM$_{Burg}$-Post	5.5	-2.0	1.0	1.5	2.6	2.0	2.2
Ψ-32B-KEM$_{Burg}$	5.2	-2.1	1.0	1.5	2.6	1.9	2.1
Ψ-32B-KEM$_{Burg}$-Post	5.6	-2.0	1.0	1.5	2.6	2.0	2.2
1B-RBPF	-1.4	-4.2	1.0	1.3	2.4	1.7	2.0
4B-RBPF	3.2	-3.0	1.0	1.3	2.2	1.8	2.0
32B-RBPF	2.0	-2.9	1.0	1.4	2.6	1.9	2.1
32B-RBPF$_{(Standalone)}$	-0.1	-3.9	1.0	1.3	2.5	1.8	2.0
32B-RBPF-Post	2.4	-2.4	1.0	1.4	2.5	1.9	2.1
Ψ-32B-RBPF	1.9	-2.9	1.0	1.4	2.6	1.9	2.1
Ψ-32B-RBPF-Post	2.4	-2.4	1.0	1.4	2.6	2.0	2.2
Ψ-32B-DEKF-Post	1.5	-3.5	1.0	1.3	2.5	1.8	2.0
Ψ-32B-DUKF-Post	1.5	-3.6	1.0	1.3	2.5	1.8	2.0
Ψ-32B-KEM-Post	1.3	-3.3	1.0	1.4	2.6	1.8	2.1
Ψ-32B-NPF-Post	2.5	-2.5	1.0	1.4	2.6	2.0	2.2

Table 114: Comparison between various subband methods introduced in Chapter VII, for L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
1B-DKF	6.6	0.4	1.0	1.6	3.0	2.3	2.5
4B-DKF	8.5	0.5	1.0	1.6	2.5	2.1	2.2
32B-DKF	8.7	0.6	1.0	2.0	2.8	2.3	2.5
32B-DKF-Post	10.2	1.0	1.0	2.1	3.1	2.4	2.7
Ψ-32B-DKF	8.4	0.5	1.0	2.1	3.0	2.3	2.5
Ψ-32B-DKF-Post	9.8	0.9	1.0	2.1	3.1	2.5	2.7
1B-KEM$_{Burg}$	6.7	0.3	1.0	1.9	3.0	2.3	2.6
4B-KEM$_{Burg}$	9.3	1.4	1.0	2.1	3.1	2.5	2.7
32B-KEM$_{Burg}$	9.2	0.5	1.0	2.0	3.0	2.3	2.5
32B-KEM$_{Burg}$-Post	10.1	1.0	1.0	2.0	3.1	2.5	2.7
Ψ-32B-KEM$_{Burg}$	9.8	0.6	1.0	2.0	3.0	2.3	2.5
Ψ-32B-KEM$_{Burg}$-Post	9.7	0.7	1.0	2.1	3.1	2.5	2.7
1B-RBPF	10.0	1.1	1.0	1.9	3.1	2.4	2.6
4B-RBPF	8.0	0.2	1.0	1.8	2.9	2.3	2.5
32B-RBPF	7.9	0.1	1.0	2.0	3.1	2.4	2.7
32B-RBPF$_{(Standalone)}$	6.9	-0.2	1.0	1.8	3.1	2.3	2.7
32B-RBPF-Post	9.8	1.2	1.0	2.0	3.0	2.6	2.8
Ψ-32B-RBPF	8.1	0.2	1.0	2.0	3.1	2.5	2.7
Ψ-32B-RBPF-Post	9.8	1.3	1.0	2.1	3.2	2.5	2.9
Ψ-32B-DEKF-Post	9.0	0.5	1.0	2.0	3.0	2.4	2.6
Ψ-32B-DUKF-Post	8.5	0.2	1.0	2.0	3.0	2.3	2.6
Ψ-32B-KEM-Post	8.7	0.2	1.0	1.9	3.0	2.2	2.5
Ψ-32B-NPF-Post	8.4	0.8	1.0	2.0	3.0	2.4	2.7

Table 115: Comparison between various subband methods introduced in Chapter VII, for M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
1B-DKF	11.9	3.2	1.0	2.1	3.3	2.7	2.8
4B-DKF	12.4	2.9	1.0	1.9	2.6	2.3	2.1
32B-DKF	11.4	2.7	1.0	2.4	3.2	2.6	2.7
32B-DKF-Post	13.4	3.7	1.0	2.5	3.5	2.9	3.1
Ψ-32B-DKF	12.3	3.2	1.0	2.6	3.4	2.8	3.0
Ψ-32B-DKF-Post	13.7	4.0	1.0	2.6	3.5	2.9	3.1
1B-KEM$_{Burg}$	12.2	3.4	1.0	2.5	3.5	2.8	3.1
4B-KEM$_{Burg}$	13.5	4.0	1.0	2.7	3.6	3.0	3.3
32B-KEM$_{Burg}$	13.0	3.0	1.0	2.4	3.4	2.7	2.9
32B-KEM$_{Burg}$-Post	14.3	4.0	1.0	2.5	3.5	2.9	3.1
Ψ-32B-KEM$_{Burg}$	12.4	2.6	1.0	2.5	3.4	2.7	3.0
Ψ-32B-KEM$_{Burg}$-Post	12.4	3.1	1.0	2.5	3.6	2.9	3.2
1B-RBPF	13.5	3.6	1.0	2.4	3.6	2.9	3.1
4B-RBPF	10.3	2.9	1.0	2.3	3.3	2.7	2.8
32B-RBPF	12.5	2.3	1.0	2.5	3.5	2.7	3.1
32B-RBPF$_{(Standalone)}$	12.1	2.3	1.0	2.3	3.5	2.7	3.1
32B-RBPF-Post	13.3	3.5	1.0	2.5	3.5	3.0	3.2
Ψ-32B-RBPF	12.0	2.7	1.0	2.5	3.6	2.7	3.3
Ψ-32B-RBPF-Post	13.7	3.9	1.0	2.6	3.4	3.0	3.3
Ψ-32B-DEKF-Post	13.1	3.6	1.0	2.6	3.4	2.8	3.0
Ψ-32B-DUKF-Post	12.9	3.4	1.0	2.5	3.4	2.9	3.0
Ψ-32B-KEM-Post	12.6	3.0	1.0	2.4	3.4	2.5	2.9
Ψ-32B-NPF-Post	13.0	3.0	1.0	2.4	3.5	2.6	3.2

Table 116: Comparison between various subband methods introduced in Chapter VII, for H-CAR conditions.

B.7. Tables for Section 8.6.2 (Bandwidth extension)

B.7.1. Cafeteria noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
DKF	0.7	-4.6	0.3	1.1	0.4	1.0	0.5
DKF+BWE	0.6	-4.7	0.2	1.1	0.6	1.0	0.6
Average	0.9	-4.5	0.2	1.1	0.7	1.0	0.6
KEM_{Burg}	0.8	-4.5	0.2	1.1	0.4	1.0	0.6
KEM_{Burg}+BWE	0.5	-4.7	0.2	1.1	0.6	1.0	0.6
Average	0.9	-4.5	0.2	1.1	0.7	1.0	0.7
RBPF	1.7	-4.2	0.2	1.1	0.4	1.0	0.5
RBPF+BWE	0.9	-4.5	0.2	1.1	0.6	1.0	0.6
Average	1.7	-4.2	0.2	1.1	0.6	1.0	0.6

Table 117: Application of the method of Bandwidth Extension in the context of speech enhancement in VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
DKF	6.2	-1.6	0.5	1.2	0.9	1.4	0.9
DKF+BWE	5.8	-2.0	0.5	1.2	1.1	1.3	1.0
Average	6.3	-1.6	0.6	1.2	1.2	1.4	1.0
KEM_{Burg}	6.1	-1.6	0.6	1.2	0.9	1.4	0.9
KEM_{Burg}+BWE	5.8	-2.0	0.5	1.2	1.2	1.4	1.0
Average	6.3	-1.6	0.6	1.2	1.2	1.4	1.1
RBPF	4.8	-1.5	0.6	1.2	0.9	1.4	0.9
RBPF+BWE	6.1	-1.7	0.6	1.2	1.2	1.4	1.0
Average	6.7	-1.5	0.6	1.2	1.2	1.4	1.1

Table 118: Application of the method of Bandwidth Extension in the context of speech enhancement in L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
DKF	10.4	1.2	0.9	1.4	1.4	1.8	1.4
DKF+BWE	9.6	0.6	0.9	1.4	1.7	1.8	1.5
Average	10.6	1.1	0.9	1.5	1.8	1.8	1.5
KEM_{Burg}	10.7	1.3	0.9	1.4	1.5	1.8	1.3
KEM_{Burg}+BWE	10.0	0.7	0.9	1.4	1.8	1.8	1.5
Average	10.8	1.1	0.9	1.4	1.8	1.8	1.5
RBPF	9.9	1.0	0.9	1.3	1.4	1.8	1.3
RBPF+BWE	10.0	0.9	0.9	1.4	1.8	1.9	1.5
Average	10.7	1.2	0.9	1.5	1.8	1.9	1.6

Table 119: Application of the method of Bandwidth Extension in the context of speech enhancement in M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
DKF	12.9	3.3	1.0	1.7	1.8	2.2	1.7
DKF+BWE	11.5	2.3	1.0	1.7	2.1	1.1	1.8
Average	13.0	3.0	1.0	1.7	2.2	2.2	1.9
KEM_{Burg}	13.6	3.4	1.0	1.6	1.8	2.1	1.6
KEM_{Burg}+BWE	12.3	2.7	1.0	1.7	2.2	2.2	1.8
Average	13.6	3.2	1.0	1.7	2.2	2.2	1.8
RBPF	13.0	2.9	1.0	1.7	2.0	2.2	2.0
RBPF+BWE	12.2	2.8	1.0	1.8	2.3	2.3	2.0
Average	13.2	3.0	1.0	1.8	2.3	2.3	2.0

Table 120: Application of the method of Bandwidth Extension in the context of speech enhancement in H-CAF conditions.

B.7.2. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
DKF	0.9	-5.1	0.8	1.0	0.8	1.2	0.8
DKF+BWE	0.7	-5.1	0.6	1.1	1.0	1.1	0.9
Average	1.0	-5.0	0.7	1.0	1.0	1.2	0.9
KEM_{Burg}	0.8	-5.1	0.8	1.1	0.8	1.2	0.8
KEM_{Burg}+BWE	0.8	-5.0	0.7	1.1	1.0	1.2	0.9
Average	1.0	-4.9	0.7	1.1	1.0	1.2	0.9
RBPF	0.6	-5.2	0.8	1.0	0.8	1.2	0.8
RBPF+BWE	0.8	-5.0	0.6	1.1	1.0	1.2	0.9
Average	0.8	-5.0	0.7	1.1	1.0	1.2	0.9

Table 121: Application of the method of Bandwidth Extension in the context of speech enhancement in VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
DKF	6.5	-1.7	0.9	1.2	1.5	1.6	1.3
DKF+BWE	6.9	-1.6	0.9	1.3	1.8	1.6	1.5
Average	7.1	-1.5	0.9	1.3	1.8	1.7	1.5
KEM_{Burg}	6.7	-1.6	0.9	1.3	1.5	1.6	1.4
KEM_{Burg}+BWE	7.3	-1.4	0.9	1.3	1.7	1.7	1.5
Average	7.4	-1.3	0.9	1.3	1.8	1.7	1.5
RBPF	6.4	-1.7	0.9	1.3	1.6	1.6	1.4
RBPF+BWE	7.2	-1.4	0.9	1.3	1.7	1.7	1.5
Average	7.2	-1.4	0.9	1.3	1.8	1.7	1.5

Table 122: Application of the method of Bandwidth Extension in the context of speech enhancement in L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
DKF	8.7	-0.1	1.0	1.5	1.9	1.9	1.7
DKF+BWE	9.3	0.1	1.0	1.5	2.1	1.9	1.8
Average	9.6	0.2	1.0	1.5	2.2	2.0	1.9
KEM_{Burg}	9.0	0.0	1.0	1.5	1.9	1.9	1.7
KEM_{Burg}+BWE	9.9	0.4	1.0	1.5	2.1	1.9	1.7
Average	10.0	0.1	1.0	1.5	2.1	1.9	1.8
RBPF	8.8	-0.1	1.0	1.5	2.0	1.9	1.7
RBPF+BWE	9.7	0.3	1.0	1.5	2.1	2.0	1.8
Average	9.8	0.2	1.0	1.5	2.1	2.0	1.8

Table 123: Application of the method of Bandwidth Extension in the context of speech enhancement in M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
DKF	10.6	1.5	1.0	1.8	2.2	2.2	2.0
DKF+BWE	11.1	1.7	1.0	1.8	2.5	2.2	2.1
Average	11.7	1.9	1.0	1.8	2.5	2.3	2.1
KEM_{Burg}	11.4	1.7	1.0	1.7	2.2	2.2	2.0
KEM_{Burg}+BWE	12.0	2.1	1.0	1.8	2.5	2.3	2.1
Average	12.4	2.1	1.0	1.8	2.5	2.3	2.2
RBPF	11.0	1.5	1.0	1.8	2.4	2.2	2.1
RBPF+BWE	11.5	1.8	1.0	1.8	2.5	2.3	2.1
Average	11.8	1.8	1.0	1.8	2.5	2.3	2.2

Table 124: Application of the method of Bandwidth Extension in the context of speech enhancement in H-FAC conditions.

B.7.3. Military vehicle noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
DKF	1.5	-4.6	0.5	1.0	1.0	1.1	0.9
DKF+BWE	1.4	-4.6	0.3	1.1	1.2	1.1	1.0
Average	1.6	-4.4	0.4	1.1	1.3	1.1	1.0
KEM_{Burg}	1.2	-4.6	0.5	1.1	1.0	1.1	0.9
KEM_{Burg}+BWE	1.4	-4.6	0.4	1.1	1.2	1.1	1.0
Average	1.6	-4.5	0.4	1.1	1.2	1.1	1.0
RBPF	1.4	-4.6	0.5	1.0	1.1	1.1	0.9
RBPF+BWE	1.4	-4.6	0.5	1.0	1.2	1.1	1.0
Average	1.5	-4.5	0.5	1.0	1.2	1.1	1.0

Table 125: Application of the method of Bandwidth Extension in the context of speech enhancement in VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
DKF	6.9	-1.2	1.0	1.2	1.6	1.6	1.3
DKF+BWE	7.4	-1.1	0.9	1.2	1.8	1.6	1.4
Average	7.6	-1.0	1.0	1.2	1.8	1.6	1.5
KEM_{Burg}	6.9	-1.2	1.0	1.2	1.6	1.6	1.3
KEM_{Burg}+BWE	7.8	-0.9	1.0	1.2	1.8	1.6	1.4
Average	7.8	-0.9	1.0	1.2	1.8	1.6	1.5
RBPF	6.8	-1.3	1.0	1.2	1.6	1.5	1.4
RBPF+BWE	7.6	-1.0	1.0	1.2	1.8	1.6	1.4
Average	7.6	-1.1	1.0	1.2	1.8	1.6	1.5

Table 126: Application of the method of Bandwidth Extension in the context of speech enhancement in L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
DKF	9.4	0.8	1.0	1.4	1.9	1.9	1.7
DKF+BWE	10.5	1.2	1.0	1.5	2.2	2.0	1.8
Average	10.7	1.2	1.0	1.5	2.3	2.0	1.9
KEM_{Burg}	10.1	1.0	1.0	1.4	2.0	1.9	1.7
KEM_{Burg}+BWE	11.2	1.6	1.0	1.5	2.2	2.0	1.8
Average	11.4	1.4	1.0	1.5	2.2	2.0	1.8
RBPF	9.9	0.8	1.0	1.4	2.1	1.9	1.7
RBPF+BWE	10.8	1.3	1.0	1.4	2.2	2.0	1.8
Average	10.9	1.1	1.0	1.4	2.3	2.0	1.8

Table 127: Application of the method of Bandwidth Extension in the context of speech enhancement in M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
DKF	12.8	3.4	1.0	1.7	2.3	2.3	2.0
DKF+BWE	12.1	3.1	1.0	1.8	2.6	2.3	2.1
Average	13.5	3.4	1.0	1.8	2.7	2.4	2.2
KEM_{Burg}	13.5	3.4	1.0	1.6	2.3	2.2	2.0
KEM_{Burg}+BWE	13.1	3.5	1.0	1.8	2.6	2.4	2.2
Average	14.2	3.6	1.0	1.8	2.7	2.4	2.2
RBPF	12.8	2.9	1.0	1.7	2.7	2.3	2.1
RBPF+BWE	12.2	3.0	1.0	1.8	2.7	2.4	2.2
Average	13.2	3.0	1.0	1.8	2.7	2.4	2.3

Table 128: Application of the method of Bandwidth Extension in the context of speech enhancement in H-MIL conditions.

B.7.4. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
DKF	0.6	-4.5	1.0	1.2	2.1	1.6	1.7
DKF+BWE	0.8	-4.2	0.9	1.2	2.2	1.6	1.8
Average	0.8	-4.2	1.0	1.2	2.3	1.6	1.8
KEM_{Burg}	-0.6	-4.7	1.0	1.2	2.1	1.6	1.7
KEM_{Burg}+BWE	0.6	-4.2	1.0	1.2	2.3	1.6	1.8
Average	0.2	-4.3	1.0	1.2	2.3	1.6	1.8
RBPF	-2.5	-5.3	1.0	1.2	2.1	1.5	1.7
RBPF+BWE	0.4	-4.2	1.0	1.2	2.3	1.6	1.8
Average	-1.7	-4.7	1.0	1.2	2.3	1.6	1.8

Table 129: Application of the method of Bandwidth Extension in the context of speech enhancement in VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
DKF	5.4	-2.0	1.0	1.5	2.5	1.9	2.1
DKF+BWE	6.1	-1.4	1.0	1.5	2.6	2.0	2.1
Average	6.2	-1.5	1.0	1.6	2.7	2.0	2.2
KEM_{Burg}	4.6	-2.2	1.0	1.5	2.5	1.9	2.1
KEM_{Burg}+BWE	6.1	-1.3	1.0	1.6	2.7	2.0	2.2
Average	5.8	-1.6	1.0	1.6	2.7	2.0	2.3
RBPF	2.0	-2.9	1.0	1.4	2.6	1.9	2.1
RBPF+BWE	5.1	-1.8	1.0	1.5	2.7	2.0	2.2
Average	3.7	-2.3	1.0	1.5	2.7	2.0	2.3

Table 130: Application of the method of Bandwidth Extension in the context of speech enhancement in L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
DKF	8.7	0.6	1.0	2.0	2.8	2.3	2.5
DKF+BWE	10.9	1.7	1.0	2.0	3.0	2.4	2.5
Average	10.8	1.4	1.0	2.1	3.1	2.4	2.7
KEM_{Burg}	9.2	0.5	1.0	2.0	3.0	2.3	2.5
KEM_{Burg}+BWE	11.3	1.9	1.0	2.1	3.1	2.5	2.6
Average	11.0	1.4	1.0	2.1	3.2	2.5	2.7
RBPF	7.9	0.1	1.0	2.0	3.1	2.4	2.7
RBPF+BWE	10.5	1.4	1.0	2.1	3.1	2.5	2.7
Average	9.7	0.9	1.0	2.1	3.2	2.4	2.7

Table 131: Application of the method of Bandwidth Extension in the context of speech enhancement in M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
DKF	11.4	2.7	1.0	2.4	3.2	2.6	2.7
DKF+BWE	12.4	3.5	1.0	2.4	3.3	2.7	2.8
Average	13.3	3.3	1.0	2.6	3.5	2.8	3.0
KEM_{Burg}	13.0	3.0	1.0	2.4	3.4	2.7	2.9
KEM_{Burg}+BWE	13.1	3.6	1.0	2.5	3.5	2.8	3.0
Average	13.9	3.5	1.0	2.6	3.6	2.9	3.1
RBPF	12.5	2.3	1.0	2.5	3.5	2.7	3.1
RBPF+BWE	12.0	2.8	1.0	2.5	3.5	2.8	3.0
Average	12.7	2.8	1.0	2.7	3.7	2.9	3.2

Table 132: Application of the method of Bandwidth Extension in the context of speech enhancement in H-CAR conditions.

B.8. Tables for Chapter IX (Example of algorithm)

B.8.1. Cafeteria Noise

VL-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-5.6	-7.5	0.0	1.0	0.3	0.8	0.4
MSSUB	-3.4	-5.9	0.1	1.1	0.5	0.8	0.5
LMMSE	-1.1	-5.0	0.2	1.1	0.6	0.9	0.6
KLT	0.8	-3.3	0.2	1.0	-0.1	1.0	0.2
Proposed	2.9	-2.8	0.2	1.2	0.6	1.1	1.1

Table 133: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in VL-CAF conditions.

L-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	0.5	-4.8	0.4	1.1	0.7	1.1	0.7
MSSUB	2.4	-3.3	0.4	1.2	1.1	1.3	1.0
LMMSE	4.8	-2.1	0.5	1.1	1.1	1.3	0.9
KLT	5.6	-0.8	0.6	1.1	0.6	1.4	0.7
Proposed	7.4	-0.2	0.7	1.3	1.2	1.5	1.4

Table 134: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in L-CAF conditions.

M-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	6.5	-1.2	0.7	1.2	1.3	1.6	1.2
MSSUB	6.5	-0.8	0.8	1.4	1.8	1.7	1.5
LMMSE	10.4	1.2	0.9	1.3	1.6	1.7	1.4
KLT	9.9	2.1	0.9	1.3	1.4	1.8	1.3
Proposed	10.8	1.4	1.0	1.5	1.8	1.9	1.7

Table 135: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in M-CAF conditions.

H-CAF	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.9	1.7	0.9	1.4	1.7	2.0	1.5
MSSUB	8.1	0.7	0.9	1.6	2.2	2.1	1.9
LMMSE	13.8	3.7	1.0	1.5	2.0	2.1	1.7
KLT	12.5	3.9	1.0	1.5	1.9	2.1	1.7
Proposed	13.1	3.2	1.0	1.8	2.3	2.3	2.0

Table 136: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in H-CAF conditions.

B.8.2. Factory noise

VL-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.6	-8.5	0.1	1.0	0.6	0.9	0.7
MSSUB	0.8	-5.1	0.4	1.1	0.9	1.2	0.9
LMMSE	2.9	-3.5	0.6	1.1	1.2	1.2	1.0
KLT	0.9	-2.8	0.7	1.1	0.4	1.1	0.5
Proposed	4.8	-2.4	0.7	1.2	1.2	1.4	1.2

Table 137: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in VL-FAC conditions.

L-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.7	-5.4	0.6	1.1	1.3	1.3	1.1
MSSUB	6.0	-1.8	0.9	1.3	1.8	1.7	1.5
LMMSE	8.7	-0.5	0.9	1.4	1.8	1.7	1.5
KLT	7.2	0.3	0.9	1.1	1.1	1.6	1.0
Proposed	9.0	0.7	1.0	1.6	1.9	1.8	2.0

Table 138: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in L-FAC conditions.

M-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	3.4	-3.1	1.0	1.2	1.7	1.6	1.4
MSSUB	7.6	-0.3	1.0	1.5	2.2	2.0	1.9
LMMSE	11.8	2.4	1.0	1.6	2.1	2.0	1.8
KLT	10.2	2.3	1.0	1.2	1.6	1.9	1.4
Proposed	10.8	1.2	1.0	1.7	2.3	2.1	1.9

Table 139: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in M-FAC conditions.

H-FAC	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	7.9	-0.4	1.0	1.4	2.1	2.0	1.8
MSSUB	9.1	1.6	1.0	1.9	2.7	2.4	2.3
LMMSE	13.8	2.9	1.0	1.8	2.4	2.4	2.1
KLT	13.3	2.9	1.0	1.4	2.2	2.3	1.9
Proposed	13.0	2.6	1.0	2.0	2.8	2.4	2.4

Table 140: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in H-FACconditions.

B.8.3. Military vehicle noise

VL-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-8.3	-8.4	0.0	1.0	0.8	0.9	0.8
MSSUB	1.4	-4.6	0.2	1.1	1.2	1.2	1.0
LMMSE	3.0	-3.0	0.7	1.1	1.3	1.2	1.0
KLT	1.9	-2.1	0.2	1.0	0.3	1.1	0.4
Proposed	5.9	-1.4	0.9	1.2	1.4	1.4	1.5

Table 141: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in VL-MIL conditions.

L-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-0.3	-5.2	0.0	1.0	1.4	1.2	1.1
MSSUB	6.7	-1.3	0.9	1.3	1.8	1.6	1.5
LMMSE	9.0	0.4	1.0	1.2	1.7	1.6	1.4
KLT	7.9	0.9	0.9	1.1	0.9	1.5	0.8
Proposed	9.5	0.7	1.0	1.4	1.8	1.8	1.9

Table 142: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in L-MIL conditions.

M-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	5.7	-1.6	1.0	1.1	1.9	1.6	1.5
MSSUB	8.7	0.9	1.0	1.6	2.4	2.1	2.0
LMMSE	12.1	2.3	1.0	1.4	2.1	2.0	1.7
KLT	11.5	2.1	1.0	1.2	1.6	1.9	1.4
Proposed	11.5	2.0	1.0	1.6	2.4	2.1	2.0

Table 143: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in M-MIL conditions.

H-MIL	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	11.7	2.3	1.0	1.4	2.4	2.2	2.0
MSSUB	9.3	2.3	1.0	1.9	2.6	2.5	2.2
LMMSE	15.0	4.5	1.0	1.7	2.5	2.4	2.1
KLT	14.3	4.6	1.0	1.5	2.6	2.5	2.1
Proposed	14.1	3.6	1.0	1.9	2.7	2.4	2.3

Table 144: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in H-MIL conditions.

B.8.4. Car interior noise

VL-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-9.3	-8.1	0.9	1.1	1.9	1.3	1.5
MSSUB	-0.1	-4.5	0.9	1.3	2.3	1.6	1.8
LMMSE	-1.2	-3.9	1.0	1.2	2.2	1.6	1.7
KLT	-2.7	-3.8	0.9	1.0	0.8	1.3	0.8
Proposed	4.1	-2.1	1.0	1.5	2.4	1.8	2.2

Table 145: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in VL-CAR conditions.

L-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	-3.3	-5.9	1.0	1.2	2.4	1.6	1.9
MSSUB	4.9	-1.7	1.0	1.7	2.7	2.1	2.1
LMMSE	4.7	-0.8	1.0	1.5	2.6	2.0	2.1
KLT	3.1	-1.5	1.0	1.1	1.3	1.6	1.2
Proposed	8.4	0.3	1.0	1.9	2.8	2.1	2.5

Table 146: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in L-CAR conditions.

M-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	4.6	-1.6	1.0	1.6	3.1	2.2	2.6
MSSUB	8.4	1.1	1.0	2.2	3.0	2.4	2.5
LMMSE	12.1	2.6	1.0	2.0	3.0	2.6	2.6
KLT	10.3	2.7	1.0	1.4	2.2	2.3	2.0
Proposed	11.7	2.1	1.0	2.3	3.1	2.6	2.7

Table 147: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in M-CAR conditions.

H-CAR	SNR	ASNR	CSII	WPESQ	Csig	Cbak	Covl
Noisy	10.7	2.3	1.0	2.2	3.6	2.8	3.2
MSSUB	9.3	2.5	1.0	2.5	3.6	2.9	3.1
LMMSE	15.7	4.3	1.0	2.3	3.5	3.0	3.0
KLT	14.4	4.5	1.0	1.9	3.2	2.9	2.8
Proposed	13.9	3.8	1.0	2.6	3.6	3.0	3.2

Table 148: Comparative results between a proposed algorithm and three existing, well-reviewed algorithms. The above results were obtained from experiments in H-CAR conditions.