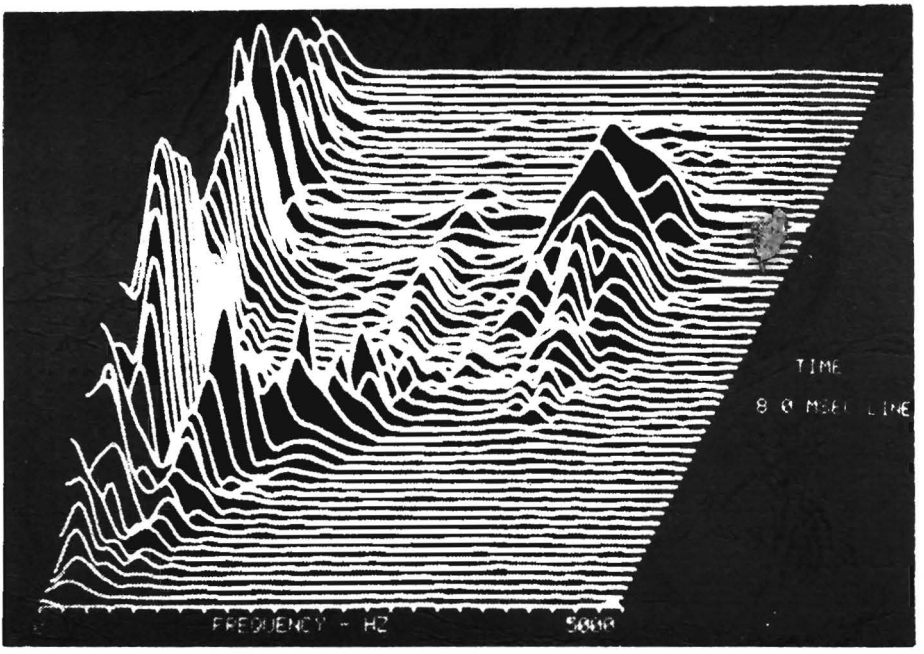
Linear and nonlinear adaptive filtering and their applications to speech intelligibility enhancement

LINEAR AND NONLINEAR ADAPTIVE FILTERING

AND

THEIR APPLICATION TO SPEECH INTELLIGIBILITY ENHANCEMENT



Y.H. GU

LINEAR AND NONLINEAR ADAPTIVE FILTERING AND THEIR APPLICATION TO SPEECH INTELLIGIBILITY ENHANCEMENT

LINEAR AND NONLINEAR ADAPTIVE FILTERING

AND

THEIR APPLICATION TO SPEECH INTELLIGIBILITY ENHANCEMENT

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus,
prof.dr. J. H. van Lint, voor een
commissie aangewezen door het College van
Dekanen in het openbaar te verdedigen op
dinsdag 15 september 1992 om 16.00 uur

door

GU, YU HUA

Geboren te Shanghai

druk: wibro dissertatiedrukkerij, helmond

to my parents

ACKNOWLEDGEMENT

I wish to express my gratitude to prof. Dr.ir. P. Eijkhof, Mr. Theo Ling, Mr. Tom C.T. Ho and his kind family, and many other friends, for their great efforts in helping me to get this research opportunity.

Many thanks are also due to prof.Dr.ir. W.M.G. van Bokhoven, my first promotor, who has kindly provided me the chance to do this research, who has given me many useful suggestions and critical remarks, and whom one could always rely on if difficulties appeared in the work.
I would also like to thank prof. Dr.-Ing. J.A.G. Jess, prof. Dr. ir. J.E.W. Beneken, prof. Dr.-Ing. O.E. Herrmann for their suggestions during their review of this thesis.

I wish to give special thanks to *all members* of the Circuit and System Design (EEB) group, who were always willing to provide help and who were always very kind to me. I enjoyed very much the four years I worked in this harmonic group. I wish to give many thanks to my colleague Jan Nijtmans for the discussions and remarks. I wish to thank to Speech Perception research group in IPO for kindly providing speech pitch data files.

Many thanks to Math Bollen for his useful suggestions.

I wish to thank my parents, who have always encouraged me to look at the positive side when I was in difficulties, and who have been firmly supporting me to pursue my own way of studies and research.

Finally, I would like to thank everybody else who has contributed to this thesis.

SUMMARY

In many practical applications, target-speech intelligibility enhancement from a contaminated signal is needed. For different kinds of statistic properties of the interference noise, the complexity of the processing is significantly different. In this dissertation, we will investigate the situation where the noise itself is interference-speech as well.

For this purpose, we developed new linear and nonlinear adaptive filtering techniques and robust pitch contour estimation algorithms, and applied them to the co-channel speech separation.
These new adaptive filtering algorithms and the pitch contour estimation algorithm can have many other applications apart from speech intelligibility enhancement.

A. Adaptive filtering techniques

In many situations signals being filtered are associated with time-varying *linear* and *nonlinear* systems, thus are nonstationary. In such cases, filtering signals only in the time-domain or in a transform-domain (rather than in a time-transform domain) seems not adequate. On the other hand, the signals to be filtered may need a large filter order in the time-domain. This can cause an undesired long time-delay in the filter output.

Motivated by this, we have investigated the LMS type filters in more detail. LMS filters are attractive due to their robustness and simplicity. New time-transform domain linear and nonlinear (second-order Volterra) LMS type adaptive filters have been developed under Gaussian (time-domain) data assumption.

In particular, we have considered the algorithms under a semi-ideal transform condition. A semi-ideal transform is defined as one-dimensional orthogonal transform which projects signals onto the orthogonal and non-overlapping sub-spaces called "bins". Under this assumption, the filter coefficients are decorrelated along the bin direction, i.e. linear filter coefficients are mutually independent over bins, and quadratic filter coefficients are mutually independent over bin-pairs.

A special selection of the window functions in the Discrete Short Time Fourier Transform (DSTFT) or in the Discrete Wavelet Transform (DWT) leads to a semi-ideal transform. Although most transforms are not semi-ideal, the time-transform bin domain LMS algorithm under a semi-ideal transform assumption can be used as a good approximation when the transform is nearly semi-ideal.

The formula relations and similarities among the linear and nonlinear algorithms in the time-transform bin domain and in the transform-domain are described. It is concluded that the T-TB domain NonLinear Normalized Least Mean Square ADaptive Filtering (NL NLMS ADF) algorithm is a generalized form, which involves all the other algorithms.

In addition, we have investigated the RLS type of *linear* and *nonlinear* algorithms. RLS filters in general have fast convergence, perform exact Least Square (LS) calculations, and are free from the Gaussian (time-domain) input data limitation.

Two new RLS adaptive filtering algorithms with an adaptive-sliding-window (one linear and one nonlinear time-domain filter), have been developed. They can provide versatile-tracking capabilities for adaptive filtering of nonstationary signals, especially those having non-constant changing speed of time-varying statistics.

B. Robust pitch contour estimation

We have built a general framework of pitch contour estimation from noise contaminated speech by a coarse-step of pitch **candidate** selection combined with a detailed-step of pitch **contour** estimation associated with stochastic models of pitch contours. This two-step algorithm is designed to use the existing pitch information both in the intra- and the inter-speech frames. It also make use of the *general* a-priori knowledge about speech pitch contours.

A new pseudo-perceptual pitch candidate estimation algorithm exploits pitch information from the local signal "carriers" and from the local signal "envelopes". The candidate estimation is then based on the coincidence of pitch correlated information over all frequency bins.

A new Hidden Markov Model (HMM)-based pitch contour estimation algorithm exploits the correlations of pitch periods in a number of successive frames (pitch contours). A stochastic model describes the pitch dynamics by using the autocorrelations of the pitch and its first and higher order derivatives. Due to the training process, the model contains some general a-priori knowledge of pitch contours, which can later serve for pitch contour estimation, where only extremely noisy speech is available.

C. Speech intelligibility enhancement by separation

The target-speech intelligibility enhancement from the co-channel speech has been investigated in this thesis. Co-channel speech signal is defined here as an additively combined signal from a target and an interference speech in a single channel.

New algorithms for a speech separation system have been developed for the co-channel speech signal over a range of Target-Interference Energy Ratio (TIR) between -12dB to +12dB. This system consists of a pitch estimation part and a speech separation part.

In the speech separation part, the above-mentioned T-TB domain linear and nonlinear adaptive filtering techniques are applied to the time-frequency bin domain as linear and nonlinear adaptive noise cancelers.

In the pitch estimation part, the above-mentioned two-step combined algorithm is applied for the simultaneous multi-pitch contour estimation.

The speech separation algorithms have been tested on summed stationary synthetic speech signals, summed nonstationary synthetic speech sentences of constant pitches and natural pitches at TIR between 0dB and -12dB. Good speech intelligibility enhancement is obtained by computer simulations. Compared with the linear version, the nonlinear one has brought further improvement in attenuating most of the remaining interference sound with slightly increased distortion.

CONTENTS

CHAPTER 1

INTRODUCTION

A human listener is generally only interested in a single source of sound (one speaker) at a time. All other air pressure variations reaching her/his ear are not of any importance in that case and can thus be considered as noise. The human listener turns out to be very capable of selecting the desired sound from the background (or foreground) noise. Those of us who visit cocktail parties or pop concerts know all about it.

Non-human listeners (i.e. machines) however, are not yet as good in this respect, despite of the amount of research invested to solve this problem. It is still more or less a mystery how the human brain processes the information picked up by the ear. What physically happens in the hearing organ is fairly well known. What happens after that the hearing nerve has picked up the information is more unclear.

The investigation, described in this dissertation, aims at the speech separation problem where one speaker has to be selected from a signal being the sum of two speakers. We follow a method which combines the advantages of the signal processing approach and the perceptual modeling approach. To develop a new speech separation method we have to develop new adaptive filtering and speech fundamental frequency (pitch) estimation techniques which are suitable for this specific application. *These described techniques are on their own applicable to more fields than just speech separation.*

Sound production model

Speech can be modeled by a *time-varying* filter representing the vocal-tract and an excitation representing the vocal-cord vibrations. The simplest model is to consider the voiced excitations as quasi-periodic impulses, the unvoiced excitations as white-noise sequences, and the vocal-tract function as an all pole filter containing at least six poles, as shown schematically in Fig.1.1.

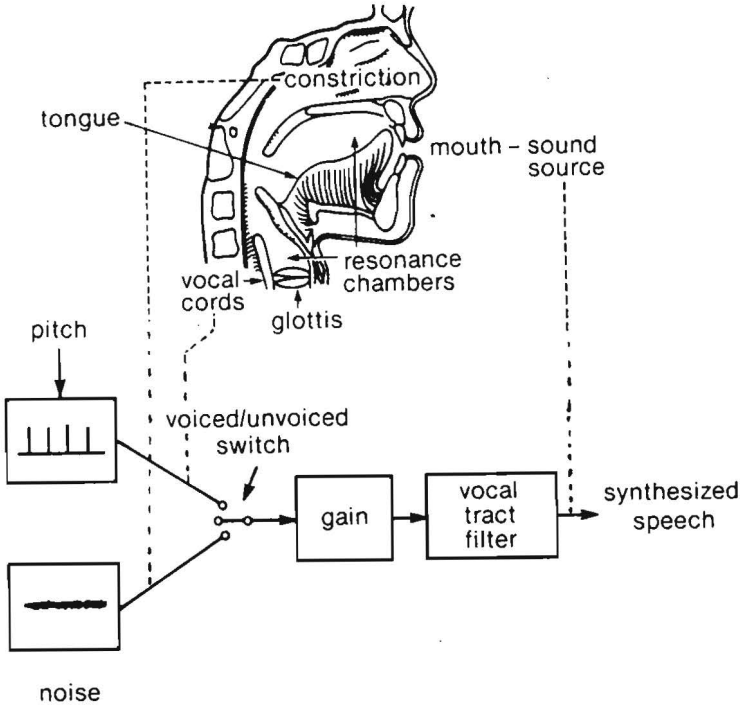


Fig.1.1 Speech production model

The reciprocal of the time interval between two vocal-cord impulses of voiced excitations is defined as the speech fundamental frequency, or the *pitch*. This speech fundamental frequency changes continuously and slowly with time. The time evolution curve of the speech fundamental frequency is called the *pitch contour*. The resonant peaks in the speech spectrum are called *formants*. They also change continuously with time, leading to the formant-trajectories.

For an intelligible voiced-speech, its fundamental frequency (pitch) and the first three formants are found to be the most important features which determine the speech with sufficient precision.

Co-channel speech separation

Because *the co-channel speech is defined as additively combined speech from multi-sources in a single channel*, it implies that a monaural speech signal is handled. The reason of selecting co-channel speech for the separation task is due to many existing problems. For example, speech from a telephone line, from a mobile telephone receiver, from a video conference, in a recorded tape, and the input speech signals for a computer automatic speech/ speaker recognition system, are almost always monaural. In such a case, our human auditory system is able to perform monaural sound analysis for tracing the target-speech. *Thus, monaural processing can be regarded as the basic processing performed by the human auditory system.* We believe that this can be an initial and fundamental investigation step towards a successful sound separation technique performed by a technical system (a machine).

For the above adaptive speech separation purpose, we will investigate new time-transform domain filtering techniques. The main reason is twofold. From the signal processing point of view, a nonstationary speech signal can be better processed in the (two-dimensional) time-transform domain because of the possibility of using temporal-localized signal-components, rather than in any one-dimensional transform. From the speech perception point of view, time-transform domain processing is closer to the human auditory global sound analysis and perception.
Due to the relatively slow time-varying characteristics of speech signals, LMS type of algorithms are good candidates because their convergence speed is expected to be suitable for such a signal. In the meantime, it is a simple algorithm with low computational costs.

In order to separate summed speech from a single receiver, it is important to estimate the pitch contour of each speaker so that the signal's quasi-periodicity can be exploited by the separation process. Because the summed speech is the only available information, we have to estimate the pitch contours from this speech signal. In order to determine the pitch periods of a sequence of speech frames, a robust multi-pitch contour estimation algorithm has been investigated. The investigation sets a general framework of pitch estimation and yields an algorithm based on a coarse candidate estimation step

followed by a detailed contour estimation step.

This dissertation simultaneously follows two outlines

1) *The first outline of the presented research is the development of new algorithms for an adaptive speech separation system.*

   This speech separation system consists of two parts: the actual speech separation part and a pitch estimation part. In the actual speech separation part, the pitch information is considered to be known. Thus the problem is limited to the adaptive summed-speech separation from a single receiver (or adaptive co-channel speech separation). The task of the pitch estimation part is to estimate pitch contours of all speakers from the summed-signal and provide it to the speech separation part.

As has been mentioned, the techniques of time-transform domain adaptive filtering and robust speech pitch estimation are applicable to more fields than just co-channel speech separation, hence lead to:

2) *A second outline, which can be followed along the investigation of adaptive filtering techniques, and the exploration of the stochastic model theory to pitch contour estimation, with a specific application to the adaptive co-channel speech separation.*

The remaining chapters of this thesis will be organized as follows:

In chapter 2, we will investigate linear and nonlinear adaptive filters for *nonstationary* signal processing. This includes the investigation of *new* time-transform domain LMS adaptive filters. It also includes *new* RLS types of algorithms with an adaptive size of the data window. Some *general* applications and simulations will also be described.

In chapter 3, we will concentrate on developing a new robust algorithm for simultaneous multi-pitch contour estimation. After reviewing the human auditory pitch perception models and the previous research on this field, we will propose a general structure by using a pitch candidate estimator plus a pitch contour estimator based on a stochastic model. Further details will be

described, including the development of a new pseudo perceptual pitch candidate estimation algorithm and exploitation of Hidden Markov Model techniques for the maximum likelihood pitch contour estimation. Simulations, results and further discussions will be included.

Chapter 4 is devoted to develop algorithms associated with a new adaptive speech separation system. We will first describe the basic ideas, the fundamentals and the basic system structure. We then describe how to apply the adaptive filtering techniques developed in chapter 2 to the adaptive speech separation. The simulations will be described in detail, including speech separation on summed stationary speech signals, summed speech sentences with constant and natural pitches. Some of the results are included. Some remarks and future work will also be given.

In chapter 5, some conclusions will be drawn, and future work will be discussed.

CHAPTER 2

ADAPTIVE FILTERING OF NONSTATIONARY SIGNALS

In this chapter, linear and nonlinear (Volterra) adaptive filters of LMS and RLS types are investigated.

For the linear LMS filters, we generalize the transform-domain **linear** *Normalized LMS (NLMS) adaptive filtering algorithm to the time-transform domain. One of the main reasons to introduce this new time-transform bin domain NLMS adaptive filtering algorithm is that it is a powerful tool for processing of nonstationary signals, for which separate time-domain or transform- (including frequency-) domain processing is not adequate anymore. This new algorithm can be used to dynamically filter nonstationary signals having large eigenvalue spread. In particular, we are interested in the algorithm under a semi-ideal transform assumption. In such a case, the filter coefficients become a set of independent sub-vectors. For other properly selected non semi-ideal orthogonal transforms, this algorithm is expected to produce a good approximation.*
The filter can also be used for reducing the filter input-output time-delay when (stationary) signals to be processed are associated with a long impulse response length.

For the nonlinear LMS filters, we generalize our transform-domain **nonlinear** *NLMS algorithm into the time-transform domain, leading to a new time-transform bin domain* **nonlinear** *NLMS adaptive filter. Again, the coefficients of the linear filter part are mutually decorrelated over bins, and the coefficients of the quadratic filter part are mutually decorrelated over bin-pairs under a semi-ideal transform assumption. Often, much reduction of the quadratic filter coefficient number can be obtained in relation to the base vector characteristics in each specific domain.*

The formula relations and the similarities among the algorithms of the transform-domain and the T-TB domain linear and NL filters are given. It can be concluded that a T-TB domain nonlinear normalized LMS adaptive filtering algorithm is a the generalized form.

Due to the Gaussian input restrictions and the relatively slow convergence of the LMS type of algorithms, the RLS type of algorithms are also investigated. Two new RLS algorithms associated with **linear** *and* **nonlinear** *adaptive filters having an adaptive sliding-window-length are derived. They are designed to provide versatile tracking capabilities to the nonstationary signals with non-constant changing speed in their time-varying statistics.*

These new filtering algorithms can have wide applications, not only in speech enhancement, but also in adaptive system identification, adaptive noise cancellation and adaptive filtering for various areas. Several examples are given.

2.1. INTRODUCTION

There has been increasing interest in *adaptive* filtering techniques in the recent decades[16,24,26,45,58,59,87]. Adaptive filters have wide application areas such as radar, sonar, underwater acoustics, seismic, audio and video signals, medical diagnoses, and many more, with various possible demands such as signal detection, estimation, filtering, system identification, noise reduction, echo cancellation, etc. Often, *nonstationary* signals (which have time-varying statistics) are handled, rather than stationary signals. Hence, this requires these filters be adaptive in order to search dynamically the time-varying optimal solution spaces.

the Least Mean Square (LMS) type of *linear* adaptive filtering algorithms are most popular, because of their simplicity and robustness. Although the time-domain filters are limited by relatively slow and non-uniform speed of convergence, improvement has been made by performing filters in other transform-domains, where signals can be decorrelated and whitened so that the convergence speed could possibly be improved[9,67,87].

However, when signals are nonstationary, a one-dimensional filter in the time-domain or in the transform-domain appears inadequate. Hence, it is necessary to investigate filters in a time-transform domain.

On the other hand, there has also been increasing interest in *nonlinear*

(NL) adaptive filtering techniques. Partially because for many NL problems, it is insufficient to use a linear approximation. Among various NL filters, much attention has been paid to the *Volterra* type of NL filters [13,16,44,45,54,59,92]. One of the attractive and particular important characteristics of Volterra filters is that *the filter output depends linearly on the filter transfer function H(z) (i.e. the Volterra kernel), despite the nonlinear relations between the filter input and output signals.* However, their complexity often prevents many practical application and consequently very limited investigation has been done up to now.

Another type of adaptive filtering technique, known as the Recursive Least Square (RLS) type has also drawn much attention [24,26,27,43,58,65,75,94]. These algorithms have faster convergence than those of the LMS type. They perform exact Least Square (LS) calculation at each time instant, and are not restricted to Gaussian input data as in the LMS type of nonlinear filter.
Sometimes, fast convergence is of paramount important during the real time processing of nonstationary signals. A relatively slow convergence filter could then always remain in the adaptation phase which is far from reaching the ideal solution. We therefore should also pay attention to RLS filters.
As the expense of convergence improvement, more calculations are usually needed for the RLS type than for the LMS type. The selection of these types of algorithm depends on the tradeoffs between the convergence speed and the computational cost.

Motivated by the above, we will first investigate in section 2.2 the *linear* LMS adaptive filtering algorithm in the time-transform domain. A new time-transform bin domain LMS adaptive filtering algorithm is developed which is suitable for processing nonstationary signals associated with a linear model (here a signal sub-space is called a transform bin).

Next, we investigate the *nonlinear* (Volterra type) LMS type of algorithm in section 2.3. A general form of LMS algorithm in the transform-domain is given. Following this line, a new *nonlinear LMS* algorithm in the time-transform domain is then derived.

In order to provide more versatile functions for the RLS type of algorithms, we will derive two *new* RLS algorithms (linear and NL,

respectively) with adaptive-sliding-window in order to cope with the filtering of nonstationary signals with non-constant changing statistics.

Finally, we will give several examples of possible applications such as adaptive noise cancellation, adaptive system identification and speech enhancement. Some simulation results from demonstrations are also included.

2.2. LINEAR LEAST MEAN SQUARE ADAPTIVE FILTERING

2.2.1. Review of the *linear* LMS adaptive filter

LMS adaptive filtering algorithms are widely used, because of the robustness and the simplicity.

Time-domain gradient LMS adaptive filtering

For a given input data sequence $\{x_n\}$, the time-domain gradient LMS adaptive filtering algorithm [103,104], as shown in Fig.2.0, can be expressed in vector forms as follows:

$$\hat{y}_n = \underline{x}_n^T \underline{h}_n \tag{2.2.1}$$

$$e_n = d_n - \hat{y}_n \tag{2.2.2}$$

$$\underline{h}_{n+1} = \underline{h}_n - \mu \frac{\partial (e_n^2)}{\partial \underline{h}_n} = \underline{h}_n + 2\mu e_n \underline{x}_n \tag{2.2.3}$$

where $\underline{x}_n=[x_n \quad x_{n-1}...x_{n-(N-1)}]^T$ is the data vector, $\underline{h}_n=[h_0(n) \quad h_1(n)...h_{N-1}(n)]^T$ is the time-domain filter weight vector, $\hat{y}_n$ is the filter output, d_n is the desired response signal which depends on the applications, and μ is the step-size controlling the convergence rate and the steady-state performance of the filter.

A sufficient condition for convergence is

$$0 < \mu < 1/\lambda_{max} \leq 1/\ tr(\mathbf{R}_{xx}) \tag{2.2.4}$$

where λ_{max} is the maximum eigenvalue of the data autocorrelation matrix

$\mathbf{R}_{xx}=E(\underset{-n}{x}\,\underset{-n}{x}^{T})$.

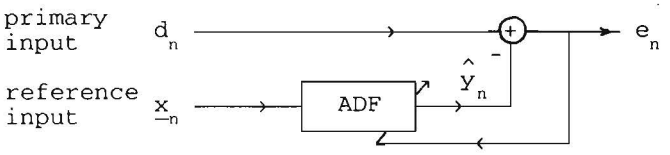


Fig.2.0 Block diagram of a LMS adaptive filter

The LMS algorithm generally suffers from a slow convergence speed

When the eigenvalue spread $\lambda_{max}/\lambda_{min}$ of the matrix R_{xx} is large, this time-domain LMS algorithm shows a slow convergence speed. This can be improved by using an adaptive filter in the transform-domain[67], provided that an orthogonal transform is properly chosen such that the spectrum of the transformed data is flattened (i.e. the eigenvalue spread is reduced).

Gradient Normalized LMS algorithm in the transform-domain[67]

After taking a block of N-data and performing an orthogonal transform **W**, a transform-domain adaptive filter can be applied. The filter coefficients are updated as soon as each new block of transformed data is available.

Suppose an orthogonal transform **W** is chosen. **W** is a unitary matrix with rank N such that $\mathbf{W}^T\mathbf{W}=\mathbf{W}\mathbf{W}^T=\mathbf{I}$. Thus, the data vectors in the time- and the transform-domain $\underset{-n}{x}$ and $\underset{-n}{Z}$, respectively, are related by the following formula

$$\underset{-n}{Z} = \mathbf{W}\,\underset{-n}{x} \qquad (2.2.5)$$

where $\underset{-n}{Z}=[z_{n,1}...z_{n,N}]^T$. Let Λ^2 be an N*N diagonal matrix with the $(i,i)^{th}$ element equal to the power estimate of $z_{n,i}$. The transform-domain gradient NLMS ADF algorithm can then be expressed as

$$\hat{y}_n = \underset{-n}{Z}^T\,\underset{-n}{H} \qquad (2.2.6)$$

$$e_n = d_n - \hat{y}_n \tag{2.2.7}$$

$$\underline{H}(n+1) = \underline{H}(n) + 2\mu_1 e_n \Lambda^{-2} \underline{Z}_n \tag{2.2.8}$$

where the matrix $\Lambda^2=\text{diag}[E|z_{n,1}|^2...E|z_{n,N}|^2]$, $\underline{H}$ is the filter weight vector in the transform-domain, μ_1 is a constant $0<\mu_1\leq 1$ associated with the filter step-size at bin i, i=1..N. The filter weight vectors between the time- and the transform-domain are related by $\underline{H}_{opt}=\mathbf{W}\underline{h}_{opt}$.

The convergence speed in the transform-domain depends on the ratio of the maximum and minimum eigenvalue ($\lambda_{max}/\lambda_{min}$) of the matrix ($\Lambda^{-2}\mathbf{R}_{zz}$). This eigenvalue spread is shown at least smaller than or equal to that in the time-domain[67]. Thus, the convergence speed in the transform-domain can be faster than that in the time-domain. A *properly* chosen **W**, like the Karhunen-Loeve transform discussed below, has the effect of pre-whitening the data, compressing the eigenvalue spread, and thus resulting in faster convergence of the filter weight vector.

*** The "ideal" transform-domain: The KLT domain**

An ideal transform **W** is the Karhunen-Loeve Transform (KLT) [1,67]. The orthogonal matrix **W** associated with KLT is formed by eigenvectors of $\mathbf{R}_{zz}$, which depends on the given data. In KLT, the autocorrelation matrix $\mathbf{R}_{zz}$ becomes diagonal, thus the eigenvalue spread in the matrix $(\Lambda^{-2}\mathbf{R}_{zz})=\mathbf{I}$ becomes one. Consequently, the fastest convergence speed can be obtained using the KLT. However it is computational costly because **W** is data-dependent.

As a result, due to the uncorrelated data in the transform-domain, solving a vector of coefficients $\underline{h}$ in the time-domain is simplified to the calculation of N scalar-coefficients of $\underline{H}$ in the transform-domain.

2.2.2. The time-transform bin domain *linear* NLMS ADF algorithm

- A new algorithm in the time-transform domain

A new generalized time-transform domain linear LMS filtering algorithm, called the Time-Transform Bin (T-TB) domain linear Normalized LMS (NLMS) ADF

algorithm, is developed in this section.

For stationary signals, a T-TB domain filter yields a mathematically equivalent solution to that of the transform-domain and of the time-domain in steady-state, but it can reduce the filter output time-delay for signals having a long impulse response length.

For nonstationary signals, especially signals having large eigenvalue spread, it provides an adequate and simple time-transform domain filtering approach.

2.2.2.1. The necessity of algorithm generalization

In some situations, a long filter tap-delay order is needed in the time-domain. Consequently, to obtain an equivalent filter order in the transform-domain, it is necessary to transform a long window of data before implementing a transform-domain filtering algorithm. This is often not suitable for a number of reasons:

* For a nonstationary signal, it is obvious that a long window is not suitable.
* If the signal is stationary, it can be associated with a long impulse response length. In such a case, the input-output of the ADF has long time-delay.

From the review in section 2.2.1, it is seen that the transform-/frequency-domain adaptive filtering is basically a *one-dimensional* filtering technique, even though the filter coefficients are adapted in each frame in order to follow the possible appearance of signal nonstationarity. A more suitable approach for filtering nonstationary signals is in the time-transform domain.

In the following, a new Time-Transform Bin (T-TB) domain *linear* NLMS ADF Algorithm is being derived *(here a signal sub-space is called a (transform) bin). In this algorithm, the length of the data sequence to be transformed can be selected shorter than the length of the system impulse response, possibly with overlap between the successive blocks where needed. By increasing the order of the adaptive filter at each bin, the previously transformed blocks of shifted data are used, so that the influence of the long length of the system*

impulse response can be taken into account.

2.2.2.2. Problem description in the T-TB domain

*** The problem in the time-domain**

Given a data sequence $\{x_n\}$, consider the following estimation problem

$$\hat{y}_n = \sum_{i=1}^{NM} h_i x_{n-i+1} \tag{2.2.9}$$

The Least Mean Square (LMS) estimation under consideration is to find, for each time instant n, an optimal solution of L=(MN) filter coefficients h_i such that the cost function J_n below is minimized

$$J_n = E[(e_n)^2] = E[(d_n - \hat{y}_n)^2] \tag{2.2.10}$$

Formula (2.2.9) can be re-written in a vector form

$$\hat{y}_n = \underline{x}_n^T \underline{h}_n \tag{2.2.11}$$

where vectors $\underline{x}_n$ and $\underline{h}_n$ are expressed by embedding the sub-vectors sequentially

$$\underline{x}_n = [x_n..x_{n-N+1} \mid x_{n-N}..x_{n-2N+1} \mid \cdots \mid x_{n-(M-1)N}..x_{n-MN+1}]^T = \begin{pmatrix} \underline{\alpha}_1(n) \\ \underline{\alpha}_2(n) \\ \vdots \\ \underline{\alpha}_M(n) \end{pmatrix}$$

$$\underline{h}_n = [h_1(n)..h_N(n) \mid h_{N+1}(n)..h_{2N}(n) \mid \cdots \mid h_{(M-1)N}(n)..h_{MN}(n)]^T = \begin{pmatrix} \underline{h}_1(n) \\ \underline{h}_2(n) \\ \vdots \\ \underline{h}_M(n) \end{pmatrix} \tag{2.2.12}$$

Formula (2.2.11) can be expressed equivalently in the trace of matrix-product

$$\hat{y}_n = tr(\mathbf{X}_n \mathbf{h}_n^T) \tag{2.2.13}$$

where the matrices $\mathbf{X}_n$ and $\mathbf{h}_n$ are arranged by dividing the vectors $\underline{x}_n$ and $\underline{h}_n$ from (2.2.12) into M frames, each frame having N-samples, and then successively embedding these frames into rows, respectively, as follows

$$\mathbf{X}_n = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix} = \begin{pmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_M^T \end{pmatrix}, \quad \mathbf{h}_n = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & \cdots & h_{2N} \\ \vdots & & & \vdots \\ h_{M1} & h_{M2} & \cdots & h_{MN} \end{pmatrix} = \begin{pmatrix} \underline{h}_1^T \\ \underline{h}_2^T \\ \vdots \\ \underline{h}_M^T \end{pmatrix} \qquad (2.2.14)$$

where the j^{th} row, (j=1..M), of $\mathbf{X}_n$ and $\mathbf{h}_n$ is associated with vectors $\underline{x}_j^T(n)$ and $\underline{h}_j^T(n)$ respectively in (2.2.12), $\underline{x}_j(n) = [x_{n-(j-1)N} \quad x_{n-(j-1)N-1} \cdots x_{n-jN+1}]^T$ and $\underline{h}_j(n) = [h_{(j-1)N+1} \quad h_{(j-1)N+2} \cdots h_{jN}]^T$. (2.2.14) can be expressed equivalently by column vectors as follows

$$\mathbf{X}_n = [\underline{x}_1 \; \underline{x}_2 \cdots \underline{x}_N], \quad \mathbf{h}_n = [\underline{h}_1 \; \underline{h}_2 \cdots \underline{h}_N] \qquad (2.2.14')$$

*** Solving the problem in the time-transform bin domain**

Suppose that the filter order needed in the time-domain is L=MN. The first step is to divide the data into M successive frames, each of length N (*overlap can be taken where necessity*). Then each frame of data is transformed by one-dimensional orthogonal transform **W** as follows:

$$\begin{pmatrix} z_{j1} \\ z_{j2} \\ \vdots \\ z_{jN} \end{pmatrix} = \mathbf{W} \begin{pmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jN} \end{pmatrix} \qquad j=1..M \qquad (2.2.15)$$

The relation between the M frames of data before and after the transform can be expressed in a matrix notation as follows

$$\mathbf{Z}_n = \mathbf{X}_n \, \mathbf{W}^T \qquad (2.2.16)$$

where the matrix $\mathbf{Z}_n$ is obtained by row-embedding the M frames of the transformed-data as follows,

$$\mathbf{Z}_n = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & & z_{2N} \\ \vdots & & & \vdots \\ z_{M1} & z_{M2} & \cdots & z_{MN} \end{pmatrix} = [\underline{Z}_1(n) \; \underline{Z}_2(n) \cdots \underline{Z}_N(n)] \qquad (2.2.17)$$

The filter coefficients in the T-TB domain are defined by the matrix $\mathbf{H}_n$,

$$\mathbf{H}_n = \begin{pmatrix} \underline{H}_{11} & \underline{H}_{12} & \cdots & \underline{H}_{1N} \\ \underline{H}_{21} & \underline{H}_{22} & & \underline{H}_{2N} \\ \vdots & \vdots & & \vdots \\ \underline{H}_{M1} & \underline{H}_{M2} & \cdots & \underline{H}_{MN} \end{pmatrix} = [\underline{H}_1(n)\ \underline{H}_2(n)\ \cdots\ \underline{H}_N(n)] \qquad (2.2.18)$$

where $\underline{Z}_i$ and $\underline{H}_i$, i=1..N, are the column vectors in matrices $\mathbf{Z}_n$ and $\mathbf{H}_n$, respectively. The following matrix relation between the time-domain and the T-TB domain filters holds

$$\mathbf{H}_n = \mathbf{h}_n \mathbf{W}^T \qquad (2.2.19)$$

For simplicity, $[\underline{Z}_1\ \underline{Z}_2 \cdots\ \underline{Z}_N]$ and $[\underline{H}_1\ \underline{H}_2\ \cdots\ \underline{H}_N]$ will be used to represent $[\underline{Z}_1(n)\ \underline{Z}_2(n)\ \cdots\ \underline{Z}_N(n)]$ and $[\underline{H}_1(n)\ \underline{H}_2(n)\ \cdots\ \underline{H}_N(n)]$, respectively, in the following.

By noticing that $\mathbf{W}^T\mathbf{W}=\mathbf{W}\mathbf{W}^T=\mathbf{I}$, the filter output in (2.2.13) can be expressed equivalently in the T-TB domain as follows

$$\hat{y}_n = \mathrm{tr}(\mathbf{Z}_n \mathbf{H}_n^T) \qquad (2.2.13')$$

(2.2.13) and (2.2.13') are equal due to the equality $\mathbf{Z}\mathbf{H}^T=(\mathbf{X}\mathbf{W}^T)(\mathbf{h}\mathbf{W}^T)^T=\mathbf{X}\mathbf{W}^T\mathbf{W}\mathbf{h}^T=\mathbf{X}\mathbf{h}^T$. Thus, an equivalent problem in the T-TB domain is to determine the matrix $\mathbf{H}$ such that the Least Mean Square (LMS) error in formula (2.2.10) is minimized.

Define the column-scanned vectors $\vec{Z}_n$, $\vec{X}_n$ and $\vec{H}_n$ of the matrices $\mathbf{Z}_n$, $\mathbf{X}_n$, $\mathbf{H}_n$ and $\mathbf{h}_n$, respectively as follows

$$\vec{Z}_n = \begin{pmatrix} \underline{Z}_1 \\ \underline{Z}_2 \\ \vdots \\ \underline{Z}_N \end{pmatrix}, \quad \vec{X}_n = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{pmatrix}, \quad \vec{H}_n = \begin{pmatrix} \underline{H}_1 \\ \underline{H}_2 \\ \vdots \\ \underline{H}_N \end{pmatrix}, \quad \vec{h}_n = \begin{pmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \vdots \\ \underline{h}_N \end{pmatrix} \qquad (2.2.20)$$

(2.2.13') can then be expressed equivalently

$$\hat{y}_n = \mathrm{tr}(\mathbf{Z}_n \mathbf{H}_n^T) = \vec{Z}_n^T \vec{H}_n = \sum_{j=1}^{N} \underline{Z}_j^T \underline{H}_j \qquad (2.2.21)$$

Define also a matrix $\mathbf{W2}=\mathbf{W}\otimes\mathbf{I}_M$, where $\otimes$ is the Kronecker product, then (2.2.16) and (2.2.19) can be expressed equivalently by the column-scanned vectors below

$$\vec{Z}_n = \mathbf{W2}\ \vec{X}_n \qquad (2.2.16')$$

$$\vec{H}_n = \mathbf{W2}\ \vec{h}_n \qquad (2.2.19')$$

In general, one has to solve the (MN) equations jointly to obtain $\mathbf{H}_n$. By taking the following partial derivatives and setting them to zero, the (MN) equations associated with $\mathbf{H}_n$ can be obtained as follows

$$\nabla_j(E(e_n^2)) = \frac{\partial E(d_n - \hat{y}_n)^2}{\partial \underset{-}{H}_j} = -2E[\underset{-j}{Z}(d_n - \sum_{i=1}^{N} \underset{-i}{Z}^T \underset{-i}{H})] = 0 \quad j=1...N \qquad (2.2.22)$$

By defining $\underset{-Z_j d}{P}(n)=E(\underset{-j}{Z} d_n)$ and $\mathbf{R}_{Z_j Z_i}(n)=E(\underset{-j}{Z}\underset{-i}{Z}^T)$, the above formula can be expressed equivalently as follows

$$\underset{-Z_j d}{P} - \sum_{i=1}^{N} \mathbf{R}_{Z_j Z_i} \underset{-i}{H} = 0 \qquad i,j=1...N \qquad (2.2.22')$$

for simplicity, $\underset{-Z_j d}{P}(n)$ and $\mathbf{R}_{Z_j Z_i}(n)$ are denoted by $\underset{-Z_j d}{P}$ and $\mathbf{R}_{Z_j Z_i}$, respectively, here and in the following. The above formula can be expressed equivalently in the column-scanned vector form as follows

$$\underset{-\vec{Z}d}{P}(n) - \mathbf{R}_{\vec{Z}\vec{Z}}(n)\ \vec{H}_n = 0 \qquad (2.2.23)$$

Thus, the optimal Wiener filter solution, which is independent of time instant n when the signal is stationary, is as follows

$$\vec{H}_{opt} = \mathbf{R}^{-1}_{\vec{Z}\vec{Z}}\ \underset{-\vec{Z}d}{P} \qquad (2.2.24)$$

provided that $\mathbf{R}^{-1}_{\vec{Z}\vec{Z}}$ is nonsingular, otherwise a pseudo inversion $\mathbf{R}^{+}_{\vec{Z}\vec{Z}}$ has to be used instead.

Remarks:

1) The column and the row vectors of **Z** and **H**

   Each row-vector in **Z** and **H** represents a specific frame of the N transformed-data and the associated filter coefficients respectively.

   While each column-vector in **Z** and **H** corresponds to the transformed data-components and the associated filter coefficients of a specific bin from M different frames, respectively.

   Because of the possible decorrelation along the bin direction, the characteristics of a T-TB domain filter can be better explained by the column-vectors of **H**, as will see later.

2) Selection of a specific time-transform bin domain

Before using a filter in the T-TB domain, we have to select a specific one-dimensional orthogonal transform **W**. Similar to the principle in the transform-domain, **W** has to be selected in such a way that the transformed-signals can be (nearly) decorrelated along the bin direction.

In the transform-domain, the KLT (which is data-dependent) is an ideal transform because of its complete decorrelation of the signals. In Time-KLT domain, this no longer holds. Because the base vectors change according to each frame of data, there is no guarantee of orthogonality among the base vectors in different frames. Consequently, the KLT is not a right choice for a T-TB domain, even if its calculation burden could be considered a negligible factor.

Because some kind of running time-transform processing will be performed, it is difficult to fully decompose the signal effectively in both time and transform directions. However, it might be possible to decompose signals in one direction. This leads to a so-called *"semi-ideal"* (one-dimensional) transform (defined later) in contrast to an "ideal" (two-dimensional) transform which fully decorrelates signals in both directions. For this purpose, an orthogonal one- dimensional transform, which can split signals into uncorrelated and orthogonal time-related components, is being searched. The base vectors of this transform must be orthogonal and data-independent. These vectors span a complete space of the signals under consideration.

In the following, the T-TB domain filtering algorithm under a "semi-ideal" transform assumption will be first considered. Then, the approximate solution under non semi-ideal transforms will be discussed. Finally, we will describe several transforms which, for special cases, are associated with a "semi-ideal" transform.

2.2.2.3. A T-TB domain linear LMS Adaptive filtering algorithm under a "semi-ideal" transform condition

- Definition of a "semi-ideal" transform

A "semi-ideal" transform is a one-dimensional orthogonal transform which

satisfies the following conditions:

* The transform matrix **W** is independent of the data;
* The space spanned by the *orthogonal* base vectors of **W** forms a complete signal space. Thus, each signal component is an orthogonal projection of the signal onto a specific base vector.

The name "semi-ideal" is used as opposed to an "ideal" (two-dimensional) transform, which fully decorrelates the signal both in time and in transform directions.

- **An optimum filter solution under a semi-ideal transform**

Consider the signals after a semi-ideal transform **W**. Under the semi-ideal definition, signals are split into non-overlapping and orthogonal bins, and the components at different transform bins become uncorrelated to each other, i.e.

$$\mathbf{R}_{Z_i Z_j} = 0 \qquad \text{if } i \neq j,\ i,j=1..N \tag{2.2.25}$$

Thus, matrix $\mathbf{R}_{\vec{Z}\vec{Z}}$ becomes block-diagonal

$$\mathbf{R}_{\vec{Z}\vec{Z}} = \text{diag}[\mathbf{R}_{Z_1 Z_1}\ \mathbf{R}_{Z_2 Z_2} \ldots \mathbf{R}_{Z_N Z_N}] \tag{2.2.26}$$

Consequently, (2.2.22') can be simplified to

$$\underline{P}_{Z_j d} - \mathbf{R}_{Z_j Z_j} \underline{H}_j = 0 \qquad j=1..N \tag{2.2.27}$$

This is equivalent to N-independent filters for N different bins, each having its own optimal coefficient vector of M-elements,

$$\underline{H}_j(\text{opt}) = \mathbf{R}^{-1}_{Z_j Z_j}\ \underline{P}_{Z_j d} \qquad j=1..N \tag{2.2.28}$$

Generally speaking, the M-coefficients within a vector are correlated due to signal correlations along the time direction in a "semi-ideal" transform associated T-TB domain.

- **Filter coefficient update formulas**

Before presenting the detailed update formulas of this algorithm, two

different approaches of updating the filter coefficients will be considered.

In the first approach, the filter output error is calculated in the time-domain, and the individual error associated with each bin is not available.

Another alternative approach is either the desired signal or the filter output error is transformed so that the the error associated with each individual bin can be obtained.

1)Updating filter coefficients using the filter output error in the time-domain

- *the limitation of using the gradient estimate*

One approach is to calculate the filter output error in the time-domain as follows

$$e_n = d_n - \hat{y}_n \qquad (2.2.29)$$

The advantage is that the desired response signals $\{d_n\}$ do not need to be transformed. Hence, the algorithm has less computational cost. In this case, the filter coefficients update formula becomes

$$\underline{H}_j(n+1) = \underline{H}_j(n) - \mu_j \nabla_j(E(e_n^2))$$

$$= \underline{H}_j(n) + 2\,\mu_j E(e_n \underline{Z}_j(n)) \qquad j=1..N \qquad (2.2.30)$$

Let e_{nj} represent the error resulted by the signals in the j^{th} bin, it is obvious that the equality $E(e_n \underline{Z}_j)=E(e_{nj} \underline{Z}_j)$ holds, because of the bin mutual independency property under a semi-ideal transform. Thus, the above formula is equal to the formula for each independent bin as follows

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\,\mu_j E(e_{nj} \underline{Z}_j(n)) \qquad j=1..N \qquad (2.2.31)$$

However, it should be mentioned that if the gradient-type estimate is used, extra errors can be introduced. This will be explained below.

In a Widrow-Hoff gradient LMS algorithm, the expectation E(.) in $\nabla_j E(e_n^2)$ is replaced by a single sample value as $\nabla_j(e_n^2)$ to simplify the calculation. Consequently, the filter coefficient update formula becomes

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\mu_j e_n \underline{Z}_j(n) \qquad j=1..N \qquad (2.2.32)$$

This formula implies that updating coefficients at bin j is influenced by the summed error value $e_n = \sum_{i=1}^{N} e_{n,i}$ rather than the error $e_{n,j}$ of the j^{th} bin. Under a semi-ideal transform assumption, all bins are linearly independent, hence it is obvious that $e_{n,j}$ rather than e_n should be used for updating $\underline{H}_j(n+1)$.

The error that may be introduced by replacing $e_{n,j}$ with e_n can be explained by a simple analysis as follows. Suppose that $e_{n,j}=0$ is reached at bin j, so that $\underline{H}_j(n+1)=\underline{H}_j(n)$ holds. This implies that the update will be stopped and an optimal solution is obtained. Unfortunately, according to (2.2.32) the filter coefficients may still need updating at (n+1) because $e_n=\sum_{i=1}^{N} e_{n,i}$ is probably not zero. Consequently, *the convergence speed is slowed down by this error bias.* Thus, there is a limitation using the Widrow-Hoff gradient LMS algorithm in a T-TB domain when the filter output error is calculated in the time-domain.

Actually, a similar situation also happens in the transform-domain gradient LMS algorithm when the error is calculated in the time-domain.

Summarizing, we apply the following iteration formulas in the corresponding algorithm

$$\hat{y}_n = \sum_{j=1}^{N} \underline{H}_j^T(n)\, \underline{Z}_j(n) \qquad (2.2.21)$$

$$e_n = d_n - \hat{y}_n \qquad \text{(time-domain error)} \qquad (2.2.29)$$

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\mu_j E(e_n \underline{Z}_j(n)) \qquad \text{(exact LMS)} \qquad (2.2.30)$$

$$\text{or: } \underline{H}_j(n+1) = \underline{H}_j(n) + 2\mu_j e_n \underline{Z}_j(n) \qquad \text{(gradient estimate)} \qquad (2.2.32)$$

2)Updating filter coefficients using the filter output error of individual bin

To overcome the extra errors introduced by the gradient estimate in the above approach, either the exact-LMS update should be used as in formula (2.2.30), or one of the following alternative approaches can be selected:

(a)Calculating the error of each bin using the transformed desired-response signals

In this method, the orthogonal transform **W** is also performed on the desired response signals $\{d_n\}$ to obtain $\{D_{nj}\}$, such that the filter output error can be calculated in the same processing domain. The disadvantage is more calculation due to this extra transform.

Now the error can be calculated in each bin individually

$$E_{nj} = D_{nj} - \underline{H}_j^T \underline{Z}_j(n) \qquad j=1..N \tag{2.2.33}$$

where E_{nj} and D_{nj} represent the output error and the desired response signal associated with the j^{th} bin in the T-TB domain, respectively. The filter weight vector of each bin can then be updated as follows

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\mu_j E_{nj} \underline{Z}_j(n) \quad j=1..N \quad \text{(Gradient estimate)} \tag{2.2.34}$$

when necessary, $\{E_{nj}\}$ can be inverse-transformed later into the time-domain error $\{e_n\}$.

(b)Calculating the error in the time-domain and followed by a transform

Another possibility is that the error is still calculated in the time-domain using $e_n = d_n - \hat{y}_n$, afterwards the error sequence $\{e_n\}$ is transformed by the same **W** to obtain the individual bin error $\{E_{nj}\}$. In this case, the filter coefficients can be updated as follows

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\mu_j E_{nj} \underline{Z}_j(n) \qquad \text{(Gradient estimate)} \tag{2.2.35}$$

Remarks

If one of the above approaches is used, one should be aware of the *wraparound-error* which might appear. This wraparound-error is caused by performing a circular convolution/correlation instead of the required linear one, when the data before the transform is not properly arranged. In order to obtain the correct results as in the linear convolution, we can use either the overlap-add or the overlap-save approaches[70,87], by adding zeros (or old-data of the previous frame) after (or in front of) the data sequence prior to the transform, and after taking the inverse-transform, discarding the incorrect part of the data.

2.2.2.4. A T-TB domain linear Normalized LMS adaptive filtering algorithm

In this section, the normalized algorithm corresponding to that in the latter section 2.2.3.3 is being derived.

Similar to the situation in the transform-domain, one can use a Normalized LMS (NLMS) algorithm in a T-TB domain in order to obtain fast convergence speed. This can be obtained by taking the filter step-size as follows,

$$\mu_j = \mu 0_j \,/\, ME(|z_{nj}|^2) \qquad 0 < \mu 0_j \le 1 \qquad j=1..N \tag{2.2.36}$$

where $\mu 0_j$ is a constant controlling the convergence speed and filter steady-state performance in bin j. The normalized update formula corresponding to (2.2.30) can be obtained as follows

$$\underline{H}_j(n+1) = \underline{H}_j(n) + 2\,\mu 0_j\,\Lambda_j^{-2} E(e_n\,\underline{Z}_j(n)) \qquad j=1..N \tag{2.2.37}$$

where Λ_j^2 is a M*M diagonal matrix defined as follows

$$\Lambda_j^2 = ME(|z_{nj}|^2)\mathbf{I}_M \tag{2.2.38}$$

Formula (2.2.37) can also be written in matrix form as follows

$$\vec{H}_{n+1} = \vec{H}_n + 2\,\mu\,\Lambda^{-2} E(e_n\,\vec{Z}_n) \tag{2.2.39}$$

where the matrices Λ^2 and μ are defined as follows

$$\Lambda^2 = diag[\Lambda_1^2 \ldots \Lambda_N^2] = diag[E(|z_{n1}|^2)..E(|z_{nN}|^2)]\otimes(M\mathbf{I}_M) \tag{2.2.40}$$

$$\mu = diag[\mu 0_1 \ldots \mu 0_N]\otimes\mathbf{I}_M \qquad 0 < \mu 0_j \le 1 \qquad j=1..N \tag{2.2.41}$$

where $\otimes$ is the Kronecker product, and the constant value may be selected differently $\mu 0_i \neq \mu 0_j$, $i \neq j$. This implies that the convergence speed of different bin can be controlled separately by a different step-size constant if necessary.

The above algorithm is summarized in Table 2.1.

Iteration at time instant n:

$$\hat{y}_n = \vec{H}_n^T \vec{Z}_n \qquad \text{(T2.1.1)}$$

$$e_n = d_n - \hat{y}_n \qquad \text{(T2.1.2)}$$

$$\Lambda^2 = \text{diag}[E(|z_{n1}|^2) \ldots E(|z_{nN}|^2)] \otimes (M\mathbf{I}_M) \qquad \text{(T2.1.3)}$$

$$\mu = \text{diag}[\mu 0_1 \ldots \mu 0_N] \otimes \mathbf{I}_M \qquad 0 < \mu 0_j \leq 1 \quad j=1..N \qquad \text{(T2.1.4)}$$

$$\vec{H}_{n+1} = \vec{H}_n + 2\,\mu\,\Lambda^{-2} E(e_n \vec{Z}_n) \quad \text{(exact LMS)} \qquad \text{(T2.1.5)}$$

$$\vec{H}_{n+1} = \vec{H}_n + 2\,\mu\,\Lambda^{-2} e_n \vec{Z}_n \quad \text{(gradient estimate)} \qquad \text{(T2.1.6)}$$

Table 2.1 A T-TB domain *linear* NLMS adaptive filtering algorithm

2.2.2.5. Further discussion

In the above, a T-TB domain *linear* NLMS ADF algorithm is developed under a semi-ideal transform assumption. The signal components are decorrelated along the transform-bin direction. Consequently, the algorithm reduces to a set of N-independent sub-algorithms associated with N-independent bins, each having its own adaptive step-size normalized by the signal energy in the associated bin.

As mentioned in the previous overview, in the transform-domain only the KLT can reach this aim. In a T-TB domain, however, the KLT is not a semi-ideal transform.

In the sequel, these import aspects will be discussed:

a) The existence of "semi-ideal" transforms, with some examples.

b) The use of the algorithm under a "semi-ideal" transform assumption as an approximation to other non-semi-ideal transform condition.

c) The necessity of signal windowing and overlapping.

d) The properties of a T-TB domain algorithm.

e) The advantages of a T-TB domain algorithm.

f) The degeneration of the algorithm under specific conditions.

a) The existence of a semi-ideal transform

-Examples: DSTFT-based/DWT-based time-frequency domain

Up to the latter section it is not yet clear whether a semi-ideal transform can actually be found. However in the following we may expect to find them for some restricted cases, which may yield a proper generalization later on.

In the following we will consider two different transforms, *the Discrete Short Time Fourier Transform (DSTFT) and the Discrete Wavelet Transform (DWT).* In fact, each of the transforms contains a set of transforms depending on the selection of a specific window/wavelet function. These two transforms represent two different types of signal frequency decompositions. Such transforms are especially suitable for nonstationary signal analysis in the time-frequency domain. *We then will notice that there is a special case in each of the transforms which is associated with a semi-ideal transform.*

*** DSTFT-based frequency decomposition**

DSTFT-type frequency decomposition[36,81] is associated with a base function made up by translating and modulating a single *window* function. Each specific frequency channel is related to the corresponding translated and modulated window function.

In particular, there is an *ideal* choice of frequency decompositions by the DSTFT which is associated with a "semi-ideal" orthogonal transform.
In this specific DSTFT, the signal space is decomposed into *non-overlapping* and *mutually orthogonal* frequency channels, with uniform frequency response and identical bandwidth. It is associated with the following *orthonormal* base functions

$$w_{j,k}(t) = w(t+k)\exp(i2\pi jt) \tag{2.2.42}$$

where w(t) is the sinc window function,

$$w(t) = \mathrm{sinc}(t) = \sin(\pi t)/(\pi t) \tag{2.2.43}$$

The signal f(t) can be expanded in this orthonormal basis as

$$f(t) = \sum_{j,k} a_{j,k} w_{j,k}(t) \qquad (2.2.44)$$

where the DSTFT coefficients $a_{j,k}$ are obtained by the following inner product

$$a_{j,k} = \langle f(t), w_{j,k}(t) \rangle = \sum_{t} f(t)\, w^{*}_{j,k}(t) \qquad (2.2.45)$$

where k is the time index at window center, j is the bin index, * is the complex conjugation, and the realm of time index t in the summation is determined by the length of the symmetric window function w(t) in (2.2.42). In this case, for each fixed j, $\{w_{j,k}\}$ spans the j^{th} channel.

However, there is no general technique to obtain such decompositions with windows that have both desirable localization properties and good numerical algorithms associated with them. Hence it represents an ideal case which is physically unrealizable.

In most cases, a desirable window w(t) is selected before a DSTFT. The corresponding base functions $w_{j,k}$ are then **not** orthonormal and have overlap between the neighboring channels. However, there usually exists a dual window $\tilde{w}(t)$ corresponding to the selected window w(t). Using the bi-orthogonal basis $w_{j,k}$ and $\tilde{w}_{j,k}$, a signal can be projected onto one set and later expanded and recovered in the other set. In order to use the algorithm under a semi-ideal transform assumption as an approximation, careful selection of this window function is needed.

*** Discrete wavelet transform-based frequency decompositions**

DWT-type frequency decompositions are associated with base functions made up by translation and dilation of a *wavelet* function[14,15,83].

There is an *ideal* case in which a DWT is associated with a so-called "semi-ideal" transform. In this case, an *ideal* wavelet basis function is selected which can decompose the signal into *non-overlapping orthonormal* frequency channels. The bandwidths of these channels are related to each other by a scaling factor, and the frequency response within each channel is uniform. All the channels are symmetric with respect to the frequency origin, so that all signal components in the frequency channels are *real-values*, provided that the time-domain signal is real. The wavelet

expansion of signal f(t) can be expressed as

$$f(t) = \sum_{j,k} a_{j,k}\psi_{j,k}(t) \tag{2.2.46}$$

where $\{\psi_{j,k}(t)\}$ is the set of wavelet base functions

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t-k) \tag{2.2.47}$$

$\psi(t)$ is the wavelet and ϕ is a sinc function, related as follows

$$\psi(t) = 2\phi(2t)-\phi(t) \tag{2.2.48}$$

$$\phi(t) = \text{sinc}(t) = \sin(\pi t)/(\pi t) \tag{2.2.49}$$

Using the orthonormal property, the DWT coefficients $a_{j,k}$ can be obtained by the following inner product

$$a_{j,k}=<f(t),\psi_{j,k}(t)> = \sum_{t} f(t)\psi^{*}_{j,k}(t) \tag{2.2.50}$$

where the indices have the same meaning as in formula (2.2.45).

Compared the DWT with the DSTFT, the base function $w_{j,k}$ of the DSTFT is a translated and modulated window w, while the wavelet base function $\psi_{j,k}$ of the DWT is a translated and dilated version of the wavelet ψ.

In the sequel, it describes how this DWT transform decomposes the signal into non-overlapping orthonormal channels.

Let V_j denote the signal space with its bandwidth limited to $[0,2^j\pi)$ and for a fixed j, let $\{\phi_{j,k}\}$ be the orthogonal (sinc) base functions for V_j. Let W_j denote the signal space with a bandwidth in the range $[2^j\pi, 2^{j+1}\pi)$, and let $\{\psi_{j,k}\}$ be the orthogonal (wavelet) base functions of W_j for a given j. Thus, W_j is the orthogonal complement of V_j in V_{j+1}, such that $V_{j+1}= V_j+W_j$, $(V_j \cup W_j)=V_{j+1}$ and $(V_j \cap W_j)=\{0\}$.

A signal in V_j can then be decomposed into a function in the *low frequency band part* V_{j-1} and a function in the *high frequency band part* W_{j-1}. Further, the signal in V_{j-1} can recursively be decomposed into functions in V_{j-2} and in W_{j-2}, and so on. Thus, a signal in space V_j can be described at arbitrary accuracy by its projections onto a group of non-overlap orthogonal filter bands $\{ W_{j-1} \mid l=1,2..N \}$, where N is chosen

such that the signal energy in V_{j-N} is small enough to be neglected.

It is obvious that this ideal DWT is not physically realizable because of the non-causal sinc function.

The advantages of DWT over DSTFT are that:

* DWT has log-scale uniform frequency bandwidths rather than the uniform frequency bandwidths of the DSTFT. This is closer to human sound and vision perception.
* DWT has different a window size at different channels: a wide time-window is used in a lower frequency band; and a narrow time-window in a higher frequency band. This is better than using a fixed-size time-window in the DSTFT.
* DWT is associated with real-valued signals, while DSTFT is associated with complex-valued signals, when the time-domain signals are real.

From the above, it is shown that by selecting an ideal window function in DSTFT, or by selecting an ideal wavelet function in DWT, The DSTFT or the DWT is associated with a semi-ideal transform for a specific Time-frequency bin domain.

b)An approximate solution by using the algorithm under a semi-ideal transform assumption for a non-semi-ideal transform case

We know that it is possible to select a *proper* orthogonal (or a nearly orthogonal) transform such that the transformed-signals are almost decorrelated. In such a situation, signal components in this T-TB domain may only be correlated among a few neighboring channels. It can then be expected that the T-TB domain NLMS ADF algorithm under a semi-ideal transform assumption can give a good approximate solution for a non-semi-ideal transformed data.

c) Necessity of overlapping and windowing

- *Generalization in data arrangement*

Previously, we have discussed the situation where the data samples under consideration are correlated over an MN time interval. We divided these data

into M frames of length N, each frame transformed by **W**, and then these frames of data were processed in a T-TB domain.

In general, we might want to have some overlap between the successive frames (e.g. to prevent wrap-around error), and to use some kind of window function for time-localization instead of using a rectangular window by simply cutting data into blocks (which introduces the Gibbs effect). In such a case, the data matrix $\mathbf{X}_n$ in formula (2.2.14) should be re-arranged as

$$\mathbf{X}_n = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{S1} & x_{S2} & \cdots & x_{SN} \end{pmatrix} = \begin{pmatrix} \underset{\sim}{x}_1^T \\ \underset{\sim}{x}_2^T \\ \vdots \\ \underset{\sim}{x}_S^T \end{pmatrix} \qquad (2.2.51)$$

where $\underset{\sim}{x}_j^T = [x_{j1} \quad x_{j2} \quad \cdots \quad x_{jN}] = [w_1 x_{n-(j-1)(N-L)} \quad w_2 x_{n-(j-1)(N-L)-1} \quad \cdots \quad w_N x_{n-(j-1)(N-L)-N+1}]$, j=1..S, S=MN/(N-L), L is the selected overlapping with $0 \le L \le (N-1)$, and w_i is a symmetric time-window of length N.

d)The properties of a T-TB domain *linear* NLMS adaptive filtering algorithm

*** Signal decorrelation in one direction**

Under a semi-ideal transform assumption, signals are fully decorrelated along the bin direction. Consequently, instead of finding an LMS solution $\underset{\sim}{h}_n$ in the time-domain which is a MN-element vector problem, finding an LMS solution $\mathbf{H}_n$ in a T-TB domain is associated with N-independent M-element sub-vector problems.

In a non semi-ideal transform case, properly selected orthogonal transform reduces the signal correlation degree in a T-TB domain as compared to the time-domain, In this case, an algorithm under a semi-ideal transform assumption can be used as a good approximate solution to the non-semi-ideal transformed data.

*** Convergence speed**

If the time-domain signal has large eigenvalue spread, the eigenvalue spread $\lambda_{max}/\lambda_{min}$ of the matrix $(\Lambda^{-2}\mathbf{R}_{\vec{z}\vec{z}})$ in a T-TB domain is compressed,

provided an orthogonal transform is properly selected. Thus, faster convergence speed can be expected.

e)Main advantages of a T-TB domain *linear* NLMS adaptive filtering algorithm

*** An adequate approach for filtering nonstationary-signal**

For nonstationary signals, separate time-domain or transform-domain filtering is inadequate. A T-TB domain algorithm using the time-transform domain filtering technique is thus more suitable for dynamically processing nonstationary signals having large eigenvalue spread.

*** Reduce the input-output time-delay**

The filter can also be used for reducing the input-output delay-time when (stationary/nonstationary) signals to be filtered are associated with a long impulse response length.

f)Algorithm degeneration to the transform-domain

As mentioned before, the algorithm is a generalization of the existing transform-domain algorithm.
If we choose the filter order along the time direction M=1, a T-TB NLMS ADF algorithm degenerates into the corresponding transform-domain algorithm.

2.2.2.6. Summary

In section 2.2, a new T-TB domain *linear* NLMS adaptive filtering algorithm has been developed. In Particular, an algorithm under a semi-ideal orthogonal transform assumption has been developed. The advantage under this assumption is that it yields N-independent sub-algorithms. Two transforms, DSTFT and DWT, are discussed as examples. Under an ideal base function, each of them is associated with a semi-ideal transform, although these are physically un-realizable. However by applying the algorithm under a semi-ideal transform assumption to a properly selected non-semi-ideal associated T-TB domain, a good approximate solution can be expected.

2.3. NONLINEAR LEAST MEAN SQUARE ADAPTIVE FILTERING

A new Time-Transform Bin (T-TB) domain Nonlinear (second-order Volterra type) NLMS Adaptive Filtering (ADF) algorithm is developed in this section under Gaussian (time-domain) data assumption.

The algorithm is particularly suitable for filtering nonstationary signals associated with time-varying NL models, and for filtering signals which are associated with a long impulse response length.
The relations and the similarities among the algorithms in the T-TB domain and in the transform-domain linear and nonlinear NLMS ADF filters are described. It is concluded that the T-TB domain NL NLMS ADF algorithm is a generalized form, which involves all the other algorithms.

Some advantages of this algorithm are discussed, such as the complete decorrelation for the linear filter coefficients among the bins and for the NL filter coefficients among the bin-pairs under the semi-ideal transform assumption, and the reduction of the quadratic filter coefficient number, etc.

2.3.1. Introduction

In this section, Volterra-type NonLinear (NL) adaptive filtering algorithms will be investigated. As mentioned before, a lot of practical problems are NL, for which a linear approximation is not sufficient. Hence, particular attention is paid to the investigation of a Volterra type of NL FIR filters in this section. This is because many NL systems can be well approximated by a Volterra series expansion of truncated order. Besides, Volterra filters have the attractive characteristic that *its output depends linearly on the filter transfer function*, despite of *its NL input-output relations*. Furthermore an FIR-type NL filter is numerically stable, in contrast to the IIR-type NL filter which usually suffers from numerical instability.

A parallel set of the **NL** LMS type of algorithms (parallel to the linear versions) in some new domains will be investigated in section 2.3. Signals under consideration now are associated with a NL model. We might have

nonstationary signals with large eigenvalue-spread, or (stationary/ nonstationary) signals associated with a long impulse response length. We then may expect to use a corresponding algorithm which can be performed in a transform-domain to improve the relatively slow and non-uniformly convergence speed in the time-domain. We may also need an adequate time-transform domain algorithm where the signal components evolve with time, in order to handle the nonstationary signals in a better way.

In many situations, we are dealing with signals having a Gaussian probability density function (pdf). This leads to the particularly attractive property that the linear and the quadratic filter coefficients are decoupled. *In the following we will restrict ourself to the Gaussian (time-domain) input data.*

The current section will be organized as follows. First the time-domain LMS NL Volterra filters will be reviewed in section 2.3.2. A *new* general form of a NL NLMS ADF filtering algorithm in the transform-domain will be given in section 2.3.3. Some properties are also investigated. Then, we will derive a *new* generalized *nonlinear* algorithm in the Time-Transform Bin (T-TB) domain, with further discussion on the properties of the NL filter part in section 2.3.4. The formula relations among the T-TB domain NL and linear, the transform-domain NL and linear ADF algorithms will be given in section 2.3.5. In section 2.3.6, an example is given to show how to use this algorithm. A short summary of the conclusion is given in section 2.3.7.

2.3.2. Review of the time-domain LMS nonlinear second-order Volterra filter

Consider the following NL estimation problem using the truncated second-order discrete *Volterra* kernel

$$\hat{y}_n = h_0 + \sum_{m_1=1}^{N} h^{(1)}_{m_1}(n)\, x_{n-m_1+1} + \sum_{m_1 . m_2=1}^{N} h^{(2)}_{m_1 . m_2}(n)\, x_{n-m_1+1} x_{n-m_2+1} \qquad (2.3.1)$$

or equivalently in vector and matrix notation

$$\hat{y}_n = h_0 + \underline{x}_n^T \underline{h}^{(1)}_n + \underline{x}_n^T \mathbf{h}^{(2)}_n \underline{x}_n \qquad (2.3.2)$$

where vectors $\underline{x}_n$, $\underline{h}^{(1)}_n$ and matrix $\mathbf{h}^{(2)}_n$ are defined as follows

$$\underset{\sim}{x}_n=[x_n\ x_{n-1}...x_{n-N+1}]^T, \qquad \underset{\sim}{h}_n^{(1)}=[h_1(n)\ h_2(n)...h_N(n)]^T \tag{2.3.3}$$

$$\mathbf{h}_n^{(2)}=\begin{pmatrix} h_{11} & \cdots & h_{1N} \\ h_{21} & & h_{2N} \\ \vdots & & \vdots \\ h_{N1} & \cdots & h_{NN} \end{pmatrix}_n \tag{2.3.4}$$

The Least Mean Square (LMS) estimation under consideration is to find, at each time instant n, an optimal solution of the filter coefficients, such that the following cost function J_n is minimized

$$J_n = E[(e_n)^2] = E[(d_n-\hat{y}_n)^2] \tag{2.3.5}$$

Where d_n is the desired response signal, and $\hat{y}_n$ is the filter output. Under zero-mean Gaussian input assumption, the time-domain NL LMS ADF algorithm[16,92] can be expressed as follows

$$\hat{y}_n^{(1)} = \sum_{i=1}^{N} h_i^{(1)} x_{n-i+1} \tag{2.3.6}$$

$$\hat{y}_n^{(2)} = \sum_{i=1}^{N}\sum_{j=1}^{N} h_{i,j}^{(2)} (x_{n-i+1}x_{n-j+1} - R_{xx}(i-j)) \tag{2.3.7}$$

$$e_n = d_n - \hat{y}_n = d_n - \hat{y}_n^{(1)} - \hat{y}_n^{(2)} \tag{2.3.8}$$

$$h_i^{(1)}(n+1) = h_i^{(1)}(n) + 2\mu_1 x_{n-i+1} \tag{2.3.9}$$

$$h_{i,j}^{(2)}(n+1) = h_{i,j}^{(2)}(n) + \mu_2 e_n x_{n-i+1}x_{n-j+1} \tag{2.3.10}$$

where μ_1 and μ_2 are chosen such that

$$0 < \mu_1 < 1/\lambda_{max}, \quad 0 < \mu_2 < 1/(2\lambda_{max}^2) \tag{2.3.11}$$

where λ_{max} is the maximum eigenvalue of $\mathbf{R}_{xx}$. In the steady-state, the linear and quadratic filter weights converge to the following optimal solution

$$\underset{\sim}{h}_{opt}^{(1)} = \mathbf{R}_{xx}^{-1}\ \underset{\sim}{P}_{dx} \tag{2.3.12}$$

$$\mathbf{h}_{opt}^{(2)} = 1/2\ \mathbf{R}_{xx}^{-1}\ \mathbf{R}_{dxx}\ \mathbf{R}_{xx}^{-1} \tag{2.3.13}$$

where $\mathbf{R}_{xx}=E(\underset{\sim}{x}_{n}\underset{\sim}{x}_{n}^{T})$, $\underset{\sim}{P}_{dx}=E(d_{n}\underset{\sim}{x}_{n})$ and $\mathbf{R}_{dxx}=E(d_{n}\underset{\sim}{x}_{n}\underset{\sim}{x}_{n}^{T})$. The convergence speed of the linear filter weights depends on the eigenvalue spread $(\lambda_{max}/\lambda_{min})$ of the data autocorrelation matrix $\mathbf{R}_{xx}$, while the convergence speed of the quadratic filter weights depends on the squared-ratio $(\lambda_{max}/\lambda_{min})^2$ of this eigenvalue spread.

2.3.3. The transform-domain *nonlinear* NLMS ADF algorithm

As mentioned previously, the convergence speed of this time-domain NL LMS ADF algorithm is relatively slow when the signal spectrum is not flat. In particular, the convergence speed of the NL filter part depends on the squared-ratio of the maximum and the minimum eigenvalues. Consequently, slow filter convergence speed is mainly caused by the NL part. Hence, it is important to improve the convergence speed of a NL filter.

A natural consideration is to introduce a general form of a transform-domain algorithm. In the following, a new general form of the transform-domain (second-order Volterra) NL NLMS ADF algorithm[33] will be given, which can be considered as an extension of the corresponding linear one[26,45,67]. This general form also involves other specific domains such as a frequency-domain NL algorithm[54].

Description of the problem by a transform-domain *nonlinear* filter

Suppose the filter in the time-domain is of order N. We can obtain the data in the transform-domain by using an orthogonal transform $\mathbf{W}$

$$\underset{\sim}{Z}_{n} = \mathbf{W}\,\underset{\sim}{x}_{n} \tag{2.3.14}$$

where $\mathbf{W}$ is an unitary matrix of rank N, $\mathbf{W}^T\mathbf{W}=\mathbf{W}\mathbf{W}^T=\mathbf{I}$, $\underset{\sim}{Z}_{n}=[z_{n,1}...z_{n,N}]^T$, and $\underset{\sim}{x}_{n}=[x_{n}...x_{n-(N-1)}]^T$ are the data vectors in the transform-domain and the time-domain, respectively. It will be proved below, that a NL LMS ADF algorithm in the transform-domain has the same steady-state function as that in the time-domain

$$\hat{y}_n = h_0 + \underline{x}_n^T \underline{h}_n^{(1)} + \underline{x}_n^T \mathbf{h}_n^{(2)} \underline{x}_n = h_0 + \underline{x}_n^T \mathbf{W}^T\mathbf{W}\, \underline{h}_n^{(1)} + \underline{x}_n^T \mathbf{W}^T\mathbf{W}\, \mathbf{h}_n^{(2)} \mathbf{W}^T\mathbf{W}\, \underline{x}_n$$

$$= h_0 + (\mathbf{W}\underline{x}_n)^T(\mathbf{W}\underline{h}_n^{(1)}) + (\mathbf{W}\underline{x}_n)^T(\mathbf{W}\mathbf{h}_n^{(2)}\mathbf{W}^T)(\mathbf{W}\underline{x}_n) = h_0 + \underline{Z}_n^T \underline{H}_n^{(1)} + \underline{Z}_n^T \mathbf{H}_n^{(2)} \underline{Z}_n \quad (2.3.15)$$

where vector $\underline{h}_n^{(1)}$ and matrix $\mathbf{h}_n^{(2)}$ are the linear and quadratic filter weights in the time-domain, $\underline{H}_n^{(1)}$ and $\mathbf{H}_n^{(2)}$ are their transform-domain counterparts. h_0 is needed for the unbiased filter output. In (2.3.15), the following relations between the filters in the time-domain and the transform-domain are used

$$\underline{H}_n^{(1)} = \mathbf{W}\underline{h}_n^{(1)}, \qquad \mathbf{H}_n^{(2)} = \mathbf{W}\mathbf{h}_n^{(2)}\mathbf{W}^T \quad (2.3.16)$$

By setting $E(\hat{y}_n)=0$ in (2.3.15), the constant term h_0 can be obtained as

$$h_0 = -\,\mathrm{tr}(\mathbf{H}_n^{(2)}\mathbf{R}_{zz}(n)) = -\,E(\underline{Z}_n^T \mathbf{H}_n^{(2)} \underline{Z}_n) \quad (2.3.17)$$

Substituting (2.3.17) into (2.3.15) yields

$$\hat{y}_n = \underline{Z}_n^T \underline{H}_n^{(1)} + \mathrm{tr}(\mathbf{H}_n^{(2)}(\underline{Z}_n \underline{Z}_n^T - \mathbf{R}_{zz}(n))^T) \quad (2.3.18)$$

where the vector $\underline{H}_n^{(1)}$ and the symmetric matrix $\mathbf{H}_n^{(2)}$ are associated with the linear and the quadratic filter weights in the transform-domain respectively.

An optimal *nonlinear* filter solution

It is important to notice that *if input variables* $\{x_n\}$ *are Gaussian i.i.d.'s (independent identical distributions), each variable in* $\{z_{n,j}\}$ *(being a linear combination of* $\{x_n\}$ *after an orthogonal transform* $\mathbf{W}$*) is also Gaussian.*

By taking the partial derivatives of $E(e_n^2)$ with respect to $\underline{H}^{(1)}$ and $\mathbf{H}^{(2)}$ and setting them to zero, the following is obtained

$$\nabla_{\underline{H}^{(1)}} E(e_n^2) = 2E(\underline{Z}_n (d_n - h_0 - \underline{Z}_n^T \underline{H}^{(1)} - \underline{Z}_n^T \mathbf{H}^{(2)} \underline{Z}_n)) = 0 \quad (2.3.19)$$

$$\nabla_{\mathbf{H}^{(2)}} E(e_n^2) = 2E(\underline{Z}_n (d_n - h_0 - \underline{Z}_n^T \underline{H}^{(1)} - \underline{Z}_n^T \mathbf{H}^{(2)} \underline{Z}_n) \underline{Z}_n^T) = 0 \quad (2.3.20)$$

Noticing that $h_0 = -E(\underline{Z}_n^T \mathbf{H}^{(2)} \underline{Z}_n)$, that all the odd-order moments of $z_{n,j}$ become zero under zero-mean Gaussian assumption, that the fourth-order moment of a Gaussian random variable z can be expressed as

$$E(z_i z_j z_k z_l) = E(z_i z_j)E(z_k z_l) + E(z_i z_k)E(z_j z_l) + E(z_i z_l)E(z_j z_k)$$
$$= R_z(i-j)R_z(k-l) + R_z(i-k)R_z(j-l) + R_z(i-l)R_z(j-k) \quad (2.3.21)$$

and that $\mathbf{H}_n^{(2)}$ is a symmetric matrix, the following relations can be obtained from (2.3.19) and (2.3.20) respectively

$$\underset{\sim}{P}_{zd} - \mathbf{R}_{zz}\,\underset{\sim}{H}_n^{(1)} = 0 \quad (2.3.22)$$

$$\mathbf{R}_{dzz} - 2\mathbf{R}_{zz}\mathbf{H}_n^{(2)}\mathbf{R}_{zz} = 0 \quad (2.3.23)$$

where $\mathbf{R}_{zz}(n)=E(\underset{\sim}{Z}_n \underset{\sim}{Z}_n^T)$, $\mathbf{R}_{dzz}(n)=E(d_n \underset{\sim}{Z}_n \underset{\sim}{Z}_n^T)$ and $\underset{\sim}{P}_{zd}(n)=E(\underset{\sim}{Z}_n d_n)$ are used, and for simplicity $\mathbf{R}_{zz}$, $\mathbf{R}_{dzz}$ and $\underset{\sim}{P}_{zd}$ (will) denote $\mathbf{R}_{zz}(n)$, $\mathbf{R}_{dzz}(n)$ and $\underset{\sim}{P}_{zd}(n)$ respectively here and elsewhere without mentioning.

It is important to notice from (2.3.22) and (2.3.23), that the linear and the quadratic filter parts are decoupled, because the random variables $z_{n,i}$ are Gaussian. Thus, we can expect that the linear part will behave exactly the same as in the Transform-domain linear LMS filter.

Supposing $\mathbf{R}_{zz}$ is non-singular, the following quadratic filter optimal solution can be obtained,

$$\mathbf{H}_{opt}^{(2)} = 1/2\ \mathbf{R}_{zz}^{-1}\,\mathbf{R}_{dzz}\mathbf{R}_{zz}^{-1} \quad (2.3.24)$$

where the matrices $\mathbf{R}_{zz}$ and $\mathbf{R}_{dzz}$ are independent of time. If $\mathbf{R}_{zz}$ is singular, a pseudo inversion $\mathbf{R}_{zz}^{+}$ is used in (2.3.24). The optimal solution of the linear filter part $\underset{\sim}{H}_{opt}^{(1)} = \mathbf{R}_{zz}^{-1}\,\underset{\sim}{P}_{zd}$ remains the same as that of the transform-domain linear filter in section 2.2.2.

Transform-domain *nonlinear* LMS ADF algorithm

By using the negative gradient for the filter coefficient update, the following filter weight update formulas can be obtained

$$H_i^{(1)}(n+1)=H_i^{(1)}(n)-\mu 1_i\,\nabla_{H_i^{(1)}}E(e_n^2)=H_i^{(1)}(n)+2\,\mu 1_i\,E(e_n z_{n,i}) \quad i=1..N \quad (2.3.25)$$

$$H_{i,j}^{(2)}(n+1)=H_{i,j}^{(2)}(n)-\mu 2_{i,j}\nabla_{H_{i,j}^{(2)}}E(e_n^2)= H_{i,j}^{(2)}(n) + 2\,\mu 2_{i,j}E(e_n\, z_{n,i}\, z_{n,j})$$

$$\mu 2_{i,j} = \mu 2_{j,i}, \quad H^{(2)}_{i,j} = H^{(2)}_{j,i}, \quad j=1..i,\ i=1..N \tag{2.3.26}$$

where $\mu 1_i$ and $\mu 2_{i,j}$ are the adaptive step-size associated with the linear and quadratic filters, respectively. They control the convergence speed and the steady-state performance of the linear and the quadratic filter part respectively.

The filter output error can be calculated by using the outputs of the linear filter part $\hat{y}^{(1)}_n$ and the quadratic filter part $\hat{y}^{(2)}_n$, separately

$$\hat{y}^{(1)}_n = \sum_{i=1}^{N} H^{(1)}_i(n)\, z_{n,i} = \underline{H}^{(1)T}_n \underline{Z}_n \tag{2.3.27}$$

$$\hat{y}^{(2)}_n = \sum_{\substack{i,\,j=1 \\ (i \circ j \le N)}}^{N} H^{(2)}_{i,j}(n)\, (z_{n,i} z_{n,j} - R_{zz}(i-j)) \tag{2.3.28}$$

$$e_n = d_n - \hat{y}_n = d_n - \hat{y}^{(1)}_n - \hat{y}^{(2)}_n \tag{2.3.29}$$

where the operator $\circ$ depends on the selected transform-domain, $R_{zz}(i-j)=E(z_{n,i} z_{n,j})$. The filter output in the m^{th} bin at time instant n can be expressed as

$$\hat{y}^{(1)}_{n,m} = H^{(1)}_m(n)\, z_{n,m} \tag{2.3.30}$$

$$\hat{y}^{(2)}_{n,m} = \sum_{i \circ j=m} H^{(2)}_{i,j}(n)\, (z_{n,i} z_{n,j} - R_{zz}(i-j)) \tag{2.3.31}$$

Transform-domain *nonlinear* LMS algorithm in the normalized form

It is necessary to normalize the algorithm. *In principle, a transform can decorrelate the filter coefficients, and the normalization can speed up the filter convergence.*

After transform, signals are decorrelated when their autocorrelation matrix becomes (nearly) diagonal. Normalization then plays a role of whitening signals such that, in an ideal case, all eigenvalues become equal. An uniform speed of convergence, thus, a fast speed of convergence can be obtained in this case.

The NLMS algorithm can be obtained by dividing the filter step-size by the power estimate of the relevant signal component as follows

$$\mu 1_i = \mu 10_i / (E|z_{n,i}|^2) \tag{2.3.32}$$

$$\mu 2_{i,j} = \mu 20_{i,j} / (2E(|z_{n,i}|^2)E(|z_{n,j}|^2)) \tag{2.3.33}$$

where $\mu 10_i$ and $\mu 20_{i,j}$ are constant,

$$0 < \mu 10_i \leq 1, \qquad 0 < \mu 20_{i,j} = \mu 20_{j,i} \leq 1/2 \quad i,j=1...N \tag{2.3.34}$$

The corresponding update formulas of the linear and quadratic coefficients can be expressed in vector and matrix form as follows

$$\underline{H}_n^{(1)} = \underline{H}_n^{(1)} + 2\,\mu 1\,\Lambda^{-2}E(e_n\,\underline{Z}_n) \tag{2.3.35}$$

$$\mathbf{H}_n^{(2)} = \mathbf{H}_n^{(2)} + \mu 2\,\Lambda^{-2}E(e_n\,\underline{Z}_n\,\underline{Z}_n^T)\Lambda^{-2} \tag{2.3.36}$$

where the matrices $\mu 1$, $\mu 2$ and Λ^2 are defined as follows

$$\Lambda^2 = \mathrm{diag}[E|z_{n,1}|^2) \ \ldots \ E|z_{n,N}|^2]$$

$$\mu 1 = \mathrm{diag}[\mu 10_1 \ \mu 10_2 \ \ldots \ \mu 10_N]$$

$$\mu 2 = \begin{pmatrix} \mu 20_{1,1} & \mu 20_{1,2} & \cdots & \mu 20_{1,N} \\ \mu 20_{2,1} & \mu 20_{2,2} & \cdots & \mu 20_{2,N} \\ \vdots & \vdots & & \vdots \\ \mu 20_{N,1} & \mu 20_{N,2} & \cdots & \mu 20_{N,N} \end{pmatrix} \quad \mu 20_{i,j} = \mu 20_{j,i} \tag{2.3.37}$$

In an ideal case (KLT-domain), the matrix Λ^2 becomes

$$\Lambda^2 = \mathrm{diag}[\lambda_1 \ \lambda_2 \ \ldots \ \lambda_N] \tag{2.3.38}$$

where λ_i, i=1..N, is the eigenvalue of the matrix $\mathbf{R}_{zz}$. (2.3.35) and (2.3.36) can be written in the scalar-form because both the coefficients in the vector $\underline{H}^{(1)}$ and in the symmetric matrix $\mathbf{H}^{(2)}$ become uncorrelated.

$$H_i^{(1)}(n+1) = H_i^{(1)}(n) + 2\,\mu 10_i\,E(e_n z_{n,i})/\lambda_i \quad i=1..N \tag{2.3.39}$$

$$H_{i,j}(n+1) = H_{i,j}(n) + \mu 20_{i,j}\,E(e_n z_{n,i} z_{n,j})/(\lambda_i \lambda_j) \quad j=1..i, i=1..N \tag{2.3.40}$$

In all these cases, the corresponding gradient estimate form can be obtained by replacing E(*) by a single sample variable (*).

The filter convergence speed in the transform-domain depends on the maximum- and minimum- eigenvalue ratio ($\lambda_{max}/\lambda_{min}$) and the squared-ratio of

the matrix $(\Lambda^{-2}\mathbf{R}_{zz})$ for the linear and the quadratic filter, respectively. For an ideal transform, $(\lambda_{max}/\lambda_{min})=1$, all the coefficients in the linear and the quadratic filter part converge with uniform speed, and this convergence speed is independent of the data.

In general, by properly selecting an unitary transform **W**, the eigenvalue spread in $(\Lambda^{-2}\mathbf{R}_{zz})$ may be reduced, and a faster convergence speed can be expected in the transform-domain.

Advantages of the transform-domain *nonlinear* NLMS ADF algorithm

A Karhunen-Loeve Transform (KLT) is an ideal orthogonal transform for the transform-domain filter. All the other orthogonal transforms are sub-optimum in the concept of decorrelating the signal. The degree of decorrelation depends on the specific transform and the specific signal under consideration. After a properly selected orthogonal transform, signals can be nearly decorrelated. In this case, the algorithm under an ideal transform assumption can be used as a good approximation to the non-ideal transformed data.

a) Fast convergence speed

* **Linear and quadratic weights converge with uniform speed because of the spectral whitening**

  In the KLT-domain, all signal components are *linearly* decorrelated (thus for Gaussian z_i *linearly* independent). Thus, the corresponding signal autocorrelation matrix $\mathbf{R}_{zz}$ becomes diagonal, and all the eigenvalues λ_i of the matrix $(\Lambda^{-2}\mathbf{R}_{zz})$ are equal. Consequently, the fastest convergence speed can be obtained, where the linear and the quadratic filter parts converge with the same speed.

  In general, when the signal eigenvalue spread is large in the time-domain, one can process the signal in the transform-domain with advantage. By properly selecting an orthogonal transform such that the transformed signals are nearly decorrelated, and the eigenvalue spread in matrix $(\Lambda^{-2}\mathbf{R}_{zz})$ approaches to one, a faster filter convergence speed than in the time-domain case can be expected.

b) Decorrelation of filter coefficients

Because of the decoupling, the linear and the quadratic filter part can be

considered separately.

* **Linear filter part**

The linear filter part has the same property as that in the transform-domain linear filter. $\underline{H}^{(1)}$ in the transform-domain is N scalars instead of a vector $\underline{h}^{(1)}$ in the time-domain.

* **Quadratic filter part**

For the quadratic filter part, the transform-domain matrix $\mathbf{H}^{(2)}$ is still symmetric as $\mathbf{h}^{(2)}$ in the time-domain. The difference is that, rather than jointly solving N(N+1)/2 coefficients in $\mathbf{h}^{(2)}$, coefficients in $\mathbf{H}^{(2)}$ can be solved independently because of the mutual (nonlinear) independence of all bin-pairs.

c) Possible reduction of the number of quadratic filter coefficients

Often, a better understanding of the physical meaning in a specific transform-domain can prevent the excessive use of the quadratic filter coefficients. Rather than blindly using all quadratic terms in the time-domain, it is often possible to select only a part of the quadratic coefficients $H^{(2)}_{i,j}$ in the transform-domain, depending on a selected domain. Several examples will be given below.

Quadratic coefficient constraint in a DFT-based frequency-domain

In the DFT-based frequency-domain, the base vectors $\{\exp(-j2\pi fn)\}$ satisfy the following frequency relation,

$$\exp(-j2\pi(f_1+f_2)n) = \exp(-j2\pi f_1 n)\exp(-j2\pi f_2 n) \qquad (2.3.41)$$

which implies that the product of two data components from the different bins i and j can contribute to the output estimate $\hat{y}^{(2)}_n(k)$ at bin k, if k=i+j is satisfied. Hence, in the NL filter part, only the quadratic terms $H^{(2)}_{i,j}$ associated with $z_{n,i}z_{n,j}$ from bin i and j, which satisfy the frequency constraint k=i+j, $1 \le K \le N$, i,j=1..N, need to be selected.

* **Quadratic coefficient constraint in a WHT-domain**

Walsh-Hadmamard Transform (WHT) is one of the most frequently used

non-sinusoidal type orthogonal transform[1].

In the WHT-domain, there exists the *sequency* relation on the basis function { wal(j,t) }

$$wal(i \oplus j,t) = wal(i,t)wal(j,t) \qquad (2.3.42)$$

Consequently, only those quadratic terms $H^{(2)}_{i,j}$ associated with $z_{n,i} z_{n,j}$ from bin i and j, which satisfy the sequency constraint $(i \oplus j)=k$, $1 \le k \le N$, i,j=1..N, can possibly be selected for estimating $\hat{y}^{(2)}_n$ (here $\oplus$ represents module 2 addition). Thus, much less quadratic terms are used in the WHT-domain than that in the time-domain.

Algorithms in several other transform-domain

*** DFT-type frequency-band domain: a complex-valued algorithm**

One can choose the orthogonal transform **W** to be the DFT (FFT). In this case it is associated with complex-valued data. The corresponding formulas can be obtained by minimizing the objective function $E(e_n e^*_n)$ with respect to the weight vector $\underline{H}^{(1)}$ and matrix $\mathbf{H}^{(2)}$. The weight update formulas in (2.3.35) and (2.3.36) become,

$$\underline{H}^{(1)}_n = \underline{H}^{(1)}_n + 2\,\mu 1\,\Lambda^{-2} E(e_n \underline{Z}^*_n) \qquad (2.3.43)$$

$$\mathbf{H}^{(2)}_n = \mathbf{H}^{(2)}_n + \mu 2\,\Lambda^{-2} E(e_n \underline{Z}^*_n \underline{Z}^{*T}_n)\,\Lambda^{-2} \qquad (2.3.44)$$

where * stands for the complex-conjugation, and $|z_{n,i}|^2$ in matrix Λ^2 in formula (2.3.37) represents $(z_{n,i} z^*_{n,i})$.

*** DCT-type frequency band-domain: a real-valued algorithm**

By choosing **W** to be the Discrete Cosine Transform (DCT) [1], a nearly diagonal $\mathbf{R}_{ZZ}$ can be obtained. The DCT is considered as an orthogonal transform especially suitable for speech and image signals. The base functions of DCT are the orthonormal Chebyshev polynomials $\{N^{-1/2},\ 2N^{-1/2}\cos\frac{(2m+1)k\pi}{2N}\}$. Signals in the DCT-domain can be considered as bandpass filtered and thus have *real values*. No complex arithmetic is involved.

2.3.4. The T-TB domain nonlinear NLMS ADF algorithm

- A new algorithm in the time-transform domain

A new Time-Transform Bin (T-TB) domain nonlinear (second-order Volterra) NLMS ADF algorithm is derived, which is a NonLinear (NL) extension of the previous T-TB domain Linear Normalized LMS ADF algorithm. In particular, we consider the algorithm under a semi-ideal transform assumption, where signal components are linearly decorrelated in the bin-direction. Due to the decoupling of the linear and the NL filter part, the linear part has the same properties as those in the corresponding linear algorithm in section 2.2.3. Meanwhile, in the NL part, quadratic coefficients associated with different bin-pairs are decorrelated. The necessary number of quadratic-terms can be much reduced depending on each specifically chosen T-TB domain and on the physical background of the problem. In a properly selected non-semi-ideal T-TB domain, the algorithm under a semi-ideal transform assumption can be used as a good approximation.

The T-TB domain NL filter is particularly suitable for nonstationary signals associated with a NL model. It can also be used to reduce the input-output time-delay needed for NL filtering of signals which are associated with long impulse response.

In this section, we will generalize the T-TB domain *linear* algorithm in Section 2.2.3 into a *nonlinear* one.

Like the T-TB domain linear filtering algorithm, a T-TB domain *NL* algorithm needs to be developed for coping with nonstationary signals and for reducing the long time-delay in the filter input-output.

First, an optimal solution of the T-TB domain NL filter in the steady-state condition will be given (section 2.3.4.2). Then the T-TB domain NL LMS and Normalized LMS algorithm under a semi-ideal transform assumption will be derived (section 2.3.4.3). Some properties of the algorithm are also discussed (section 2.3.4.4). Relations of the formulas among the linear and NL algorithms in the T-TB domain and in the transform-domain will be given

(section 2.3.4.5). A simple example is given to show how to use this algorithm in practical situation (section 2.3.4.6). A short summary will be given (section 2.3.4.7).

2.3.4.1. Nonlinear problem description in the T-TB domain

*** Nonlinear problem description in the time-domain**

Supposing the filter order needed in the time-domain is MN, the following estimate can be performed in the time-domain

$$\hat{y}_n = h_0 + \sum_{m_1=1}^{MN} h_m^{(1)}(n) x_{n-m_1+1} + \sum_{m_1,m_2=1}^{MN} h_{m_1,m_2}^{(2)}(n) x_{n-m_1+1} x_{n-m_2+1} \qquad (2.3.45)$$

Or, equivalently using the convolution expression form,

$$\hat{y}_n = h_0 + h_n^{(1)} * x_n + h_{n1,n2}^{(2)} * (x_{n1} \otimes x_{n2}) \qquad (2.3.45')$$

Where $*$ stands for linear convolution. Equivalently this can be expressed in the Z-domain,

$$\hat{Y}_n(z) = H_0 + H_1(z)X_n(z) + H_2(z)(X_n(z) \otimes X_n(z)) \qquad (2.3.45'')$$

Define the time-domain data matrix $\mathbf{X}_n$ and its column-scanned vector $\vec{\mathbf{X}}_n$, the autocorrelation matrix $\mathbf{R}_{\vec{x}\vec{x}}(n)$ of $\vec{\mathbf{X}}_n$, and the quadratic data matrix $\mathbf{X2}_n$ as follows

$$\mathbf{X}_n = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix} = [\underline{x}_1\ \underline{x}_2 \cdots \underline{x}_N], \quad \vec{\mathbf{X}}_n = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{pmatrix} \qquad (2.3.46)$$

$$\mathbf{R}_{\vec{x}\vec{x}}(n) = E[\vec{\mathbf{X}}_n \vec{\mathbf{X}}_n^T], \quad \mathbf{X2}_n = \vec{\mathbf{X}}_n \vec{\mathbf{X}}_n^T \qquad (2.3.47)$$

Define the matrix $\mathbf{h1}_n$ as the linear filter part, and the symmetric matrix $\mathbf{h2}_n$ as the quadratic filter part as follows

$$\mathbf{h1}_n = \begin{pmatrix} h1_{11} & \cdots & h1_{1N} \\ h1_{21} & & h1_{2N} \\ \vdots & & \vdots \\ h1_{M1} & \cdots & h1_{MN} \end{pmatrix} = [\underline{h}_1(n)\ \underline{h}_2(n) .. \underline{h}_N(n)], \quad \mathbf{h2}_n = \begin{pmatrix} \mathbf{h2}_{11} & \mathbf{h2}_{12} & \cdots & \mathbf{h2}_{1N} \\ \mathbf{h2}_{21} & \mathbf{h2}_{22} & & \mathbf{h2}_{2N} \\ \vdots & \vdots & & \vdots \\ \mathbf{h2}_{N1} & \mathbf{h2}_{N2} & \cdots & \mathbf{h2}_{NN} \end{pmatrix} \qquad (2.3.48)$$

with $\mathbf{h2}_{ij}(n)$, denoted by $\mathbf{h2}_{ij}$ for simplicity, represents a sub-matrix of order

M*M, i,j=1..N, given below,

$$\mathbf{h2}_{ij}=\begin{bmatrix} h2_{(i-1)M+1,(j-1)M+1} & h2_{(i-1)M+1,(j-1)M+2} & \cdots & h2_{(i-1)M+1,jM} \\ h2_{(i-1)M+2,(j-1)M+1} & h2_{(i-1)M+2,(j-1)M+2} & \cdots & h2_{(i-1)M+2,jM} \\ \vdots & & & \vdots \\ h2_{iM,(j-1)M+1} & h2_{iM,(j-1)M+2} & \cdots & h2_{iM,jM} \end{bmatrix} \qquad (2.3.49)$$

(2.3.45) can then be expressed in the matrix form as follows

$$\hat{y}_n = h_0 + tr(\mathbf{X}_n \mathbf{h1}_n^T) + \vec{X}_n^T \mathbf{h2}_n \vec{X}_n \qquad (2.3.50)$$

Here h_0 is needed for the unbiased filter output. By setting $E(\hat{y}_n)=0$, h_0 can be obtained as

$$h_0 = -E(\vec{X}_n^T \mathbf{h2}_n \vec{X}_n) = -tr(\mathbf{h2}_n \mathbf{R}^T_{\vec{x}\vec{x}}(n)) \qquad (2.3.51)$$

Substituting (2.3.51) into (2.3.50) yields

$$\hat{y}_n = tr(\mathbf{X}_n \mathbf{h1}_n^T) + tr\left[\mathbf{h2}_n (\mathbf{X2}_n - \mathbf{R}_{\vec{x}\vec{x}}(n))^T\right] = \vec{X}_n^T \vec{h1}_n + tr\left[\mathbf{h2}_n (\mathbf{X2}_n - \mathbf{R}_{\vec{x}\vec{x}}(n))^T\right] \qquad (2.3.52)$$

*** Nonlinear problem description in the T-TB domain**

In order to estimate $\hat{y}_n$ in the T-TB domain, similar as mentioned in section 2.2.3, the data sequence in the time-domain is first divided into M frames, each having length N, (if necessary overlap is allowed). Then, the orthogonal transform **W** is performed successively on M-frames of data as in (2.2.15). A matrix $\mathbf{Z}_n$, as defined in (2.2.17), is used to represent the row-embedded M-frames of transformed data. Thus, for the data matrices in the time-domain and in the T-TB domain, the same relation $\mathbf{Z}_n = \mathbf{X}_n \mathbf{W}^T$ as in (2.2.16) holds. Each row-vector in $\mathbf{X}_n$ and $\mathbf{Z}_n$ corresponds to the j^{th} frame of data before and after the transform.

Let us define the column-scanned vector $\vec{Z}_n$ of matrix $\mathbf{Z}_n$ like in (2.2.20). Define the autocorrelation matrix $\mathbf{R}_{\vec{z}\vec{z}}(n)$ of $\vec{Z}_n$, the quadratic data matrix $\mathbf{Z2}_n$, the T-TB domain linear filter part $\mathbf{H1}_n$, its column-scanned vector $\vec{H1}_n$, and the T-TB domain quadratic filter part $\mathbf{H2}_n$ respectively as follows

$$\mathbf{R}_{\vec{Z}\vec{Z}}(n) = E(\vec{Z}_n \vec{Z}_n^T), \quad \mathbf{Z2}_n = \vec{Z}_n \vec{Z}_n^T \tag{2.3.53}$$

$$\mathbf{H1}_n = \begin{pmatrix} H1_{11} \cdots H1_{1N} \\ H1_{21} \quad H1_{2N} \\ \vdots \quad \vdots \\ H1_{M1} \cdots H1_{MN} \end{pmatrix} = [\underline{H1}_1 \ \underline{H1}_2 \cdots \underline{H1}_N], \quad \vec{H1}_n = \begin{pmatrix} \underline{H1}_1 \\ \underline{H1}_2 \\ \vdots \\ \underline{H1}_N \end{pmatrix} \tag{2.3.54}$$

$$\mathbf{H2}_n = \begin{pmatrix} \mathbf{H2}_{11} & \mathbf{H2}_{12} & \cdots & \mathbf{H2}_{1N} \\ \mathbf{H2}_{21} & \mathbf{H2}_{22} & & \mathbf{H2}_{2N} \\ \vdots & \vdots & & \vdots \\ \mathbf{H2}_{N1} & \mathbf{H2}_{N2} & \cdots & \mathbf{H2}_{NN} \end{pmatrix} \tag{2.3.55}$$

where $\mathbf{H2}_{ij}(n)$, denoted by $\mathbf{H2}_{ij}$ for simplicity, is a sub-matrix of order M*M, i,j=1..N.

$$\mathbf{H2}_{ij} = \begin{pmatrix} H2_{(i-1)M+1,(j-1)M+1} & H2_{(i-1)M+1,(j-1)M+2} & \cdots & H2_{(i-1)M+1,jM} \\ H2_{(i-1)M+2,(j-1)M+1} & H2_{(i-1)M+2,(j-1)M+2} & \cdots & H2_{(i-1)M+2,jM} \\ \vdots & & & \vdots \\ H2_{iM,(j-1)M+1} & H2_{iM,(j-1)M+2} & \cdots & H2_{iM,jM} \end{pmatrix} \tag{2.3.56}$$

For the linear filter part, the following relation holds

$$\mathbf{H1}_n = \mathbf{h1}_n \mathbf{W}^T \tag{2.3.57}$$

or, equivalently, in column-scanned vector form

$$\vec{H1}_n = \mathbf{W2}\ \vec{h1}_n \tag{2.3.57'}$$

where $\mathbf{W2}=(\mathbf{W}\otimes\mathbf{I}_M)$ is defined. This is similar to formula (2.2.19) in section 2.2.3. The following relation of the quadratic filter parts in the time-domain and the T-TB domain holds,

$$\mathbf{H2}_n = \mathbf{W2}\ \mathbf{h2}_n \mathbf{W2}^T \tag{2.3.58}$$

An equivalent expression to (2.2.16) using column-scanned vector notation is

$$\vec{Z}_n = \mathbf{W2}\ \vec{X}_n \tag{2.3.59}$$

Hence, the filter output can be calculated by using a T-TB domain NL filter as follows

$$\hat{y}_n = h_0 + tr(\mathbf{Z}_n \mathbf{H1}_n^T) + \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n = h_0 + \vec{Z}_n^T \vec{H1}_n + \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n \qquad (2.3.60)$$

Using the relations $(\mathbf{W2})(\mathbf{W2})^T=(\mathbf{W2})^T(\mathbf{W2})=\mathbf{I}$, $\mathbf{W}\mathbf{W}^T=\mathbf{W}^T\mathbf{W}=\mathbf{I}$, and the filter relations in (2.3.57) and (2.3.58), it can be easily proved that a time-domain and a T-TB domain NL filter produce an equivalent solution in the steady-state, since

$$\begin{aligned}\hat{y}_n &= h_0 + \vec{Z}_n^T \vec{H1}_n + \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n \\ &= h_0 + (\mathbf{W2}\,\vec{X}_n)^T(\mathbf{W2}\,\vec{h1}_n) + (\mathbf{W2}\,\vec{X}_n)^T(\mathbf{W2}\,\mathbf{h2}_n\mathbf{W2}^T)(\mathbf{W2}\,\vec{X}_n) \\ &= h_0 + \vec{X}_n^T \mathbf{W2}^T\mathbf{W2}\,\vec{h1}_n + \vec{X}_n^T(\mathbf{W2}^T\mathbf{W2})\,\mathbf{h2}_n(\mathbf{W2}^T\mathbf{W2})\,\vec{X}_n \\ &= h_0 + \vec{X}_n^T \vec{h1}_n + \vec{X}_n^T \mathbf{h2}_n \vec{X}_n \end{aligned} \qquad (2.3.61)$$

By setting $E(\hat{y}_n)=0$, h_0 value can be obtained

$$h_0 = -E(\vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n) = -tr(\mathbf{H2}_n \mathbf{R}^T_{\vec{Z}\vec{Z}}(n)) \qquad (2.3.62)$$

Substituting h_0 in (2.3.60), $\hat{y}_n$ can be obtained as follows

$$\hat{y}_n = \vec{Z}_n^T \vec{H1}_n + tr\left[\mathbf{H2}_n(\mathbf{Z2}_n - \mathbf{R}_{\vec{Z}\vec{Z}}(n))^T\right] \qquad (2.3.63)$$

2.3.4.2. The optimal solution in a T-TB domain

If x_n in the time-domain is Gaussian, $z_{n,i}$ in a T-TB domain (which is a linear combination of x_n by an orthogonal transform) is also Gaussian. Thus, *variables remain Gaussian in a T-TB domain.* Consequently, the linear part and the quadratic part of the NL filter in a T-TB domain are decoupled.

In order to obtain the optimal filter solution in the T-TB domain , a similar method can be used as in the linear case, i.e., by taking partial derivatives of $E(e_n^2)=E[(d_n-\hat{y}_n)^2]$ with respect to **H1** and **H2**, and setting them to zero.

1) For an easy derivation, the column-scanned vector $\vec{Z}_n$ of $\mathbf{Z}_n$ is used for calculating $\nabla_{\mathbf{H1}_n} E(e_n^2)$. By setting $\nabla_{\vec{H1}_n} E(e_n^2)= 0$, we obtain

$$\nabla_{\vec{H}\vec{1}_n} E(e_n^2) = 2E(\vec{Z}_n(d_n - h_0 - \vec{Z}_n^T \vec{H1}_n - \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n)) = 0 \qquad (2.3.64)$$

Notice that z_i is a Gaussian zero-mean variable, so that all the odd-order moments become zero. This yields

$$E(\vec{Z}_n d_n) - E[\vec{Z}_n(\vec{Z}_n^T \vec{H1}_n)] = 0 \qquad (2.3.65)$$

defining $\underset{\sim}{P}_{\vec{Z}d}(n)=E(\vec{Z}_n d_n)$ and $\mathbf{R}_{\vec{Z}\vec{Z}}(n)=E(\vec{Z}_n \vec{Z}_n^T)$, (2.3.65) is equivalent to

$$\underset{\sim}{P}_{\vec{Z}d}(n) - \mathbf{R}_{\vec{Z}\vec{Z}}(n)\, \vec{H1}_n = 0 \qquad (2.3.65')$$

Suppose $\mathbf{R}_{\vec{Z}\vec{Z}}$ is non-singular, the optimal filter which is independent of n, can be obtained as below

$$\vec{H1}_{opt} = \mathbf{R}_{\vec{Z}\vec{Z}}^{-1}\, \underset{\sim}{P}_{\vec{Z}d} \qquad (2.3.66)$$

Otherwise, a pseudo-inversion $\mathbf{R}_{\vec{Z}\vec{Z}}^{+}$ is to be used instead of $\mathbf{R}_{\vec{Z}\vec{Z}}^{-1}$. This result is similar to (2.2.24) in section 2.2.3, as expected.

2) By setting $\nabla_{\mathbf{H2}} E(e_n^2) = 0$, the equations associated with **H2** can be obtained as follows

$$\nabla_{\mathbf{H2}} E(e_n^2) = 2E(\vec{Z}_n(d_n - h_0 - \vec{Z}_n^T \vec{H1}_n - \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n)\vec{Z}_n^T) = 0 \qquad (2.3.67)$$

Applying (2.3.21) for the fourth-order moment (**H2** is symmetric), and using the property that all the odd-order moments of $z_{n,i}$ are zero, then yields

$$\mathbf{P}_{d\vec{Z}\vec{Z}}(n) - 2\mathbf{R}_{\vec{Z}\vec{Z}}(n)\, \mathbf{H2}_n\, \mathbf{R}_{\vec{Z}\vec{Z}}(n) = 0 \qquad (2.3.68)$$

where $\mathbf{P}_{d\vec{Z}\vec{Z}}(n)=E(d_n \vec{Z}_n \vec{Z}_n^T)$. Supposing $\mathbf{R}_{\vec{Z}\vec{Z}}$ is non-singular, the optimal solution of the quadratic filter, which is independent of n, can be obtained as follows

$$\mathbf{H2}_{opt} = 1/2\, \mathbf{R}_{\vec{Z}\vec{Z}}^{-1}\, \mathbf{P}_{d\vec{Z}\vec{Z}}\, \mathbf{R}_{\vec{Z}\vec{Z}}^{-1} \qquad (2.3.69)$$

When $\mathbf{R}_{\vec{Z}\vec{Z}}$ is singular, a pseudo-inversion $\mathbf{R}^{+}_{\vec{Z}\vec{Z}}$ is to be used instead.

Under a *semi-ideal* transform assumption (as defined in section 2.2.3), calculations needed for (2.3.66) and (2.3.69) can be much reduced.
Consider the symmetric matrix $\mathbf{R}_{\vec{Z}\vec{Z}}$, containing N*N sub-matrices as below

$$\mathbf{R}_{\vec{Z}\vec{Z}} = E(\vec{Z}\vec{Z}^{T}) = E\begin{pmatrix} \underline{Z}_1 \\ \underline{Z}_2 \\ \vdots \\ \underline{Z}_N \end{pmatrix} [\underline{Z}_1^{T}\ \underline{Z}_2^{T} \ldots \underline{Z}_N^{T}] = \begin{pmatrix} E(\underline{Z}_1\underline{Z}_1^{T}) & E(\underline{Z}_1\underline{Z}_2^{T}) & \ldots E(\underline{Z}_1\underline{Z}_N^{T}) \\ E(\underline{Z}_2\underline{Z}_1^{T}) & E(\underline{Z}_2\underline{Z}_2^{T}) & \ldots E(\underline{Z}_2\underline{Z}_N^{T}) \\ \vdots & \vdots & \vdots \\ E(\underline{Z}_N\underline{Z}_1^{T}) & E(\underline{Z}_N\underline{Z}_2^{T}) & \ldots E(\underline{Z}_N\underline{Z}_N^{T}) \end{pmatrix} \quad (2.3.70)$$

After a semi-ideal transform, the signal components are fully linear decorrelated over transform-bins. Hence, $\mathbf{R}_{\vec{Z}\vec{Z}}$ becomes block-diagonal:

$$\mathbf{R}_{\vec{Z}\vec{Z}} = \begin{pmatrix} E(\underline{Z}_1\underline{Z}_1^{T}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & E(\underline{Z}_N\underline{Z}_N^{T}) \end{pmatrix} \quad (2.3.71)$$

Consequently, the matrix inversion $\mathbf{R}^{-1}_{\vec{Z}\vec{Z}}$ in (2.3.66) and (2.3.69) can be performed through N-independent inversions on sub-matrices. Each sub-matrix is of (maximum) rank M, which is much lower than the rank of $\mathbf{R}_{\vec{Z}\vec{Z}}$.

2.3.4.3. A T-TB domain nonlinear LMS/ NLMS ADF algorithm under a semi-ideal transform assumption

As mentioned before, under a semi-ideal transform assumption, all bins are linearly independent. This implies that the linear filter coefficients associated with different bins are mutually independent, the quadratic filter coefficients associated with different bin-pairs are also mutually independent. Thus, the following simplified algorithm is obtained under a semi-ideal transform assumption.

*** Nonlinear LMS adaptive filtering algorithm**

The filter coefficients update formulas can be obtained as follows

$$\underline{H1}_i(n+1) = \underline{H1}_i(n) - \mu 1_i \nabla_{\underline{H1}_i} E(e_n^2)$$

$$= \underline{H1}_i(n) + 2\,\mu 1_i E(e_n\, \underline{Z}_i(n)) \qquad i=1..N \tag{2.3.72}$$

$$\mathbf{H2}_{i,j}(n+1) = \mathbf{H2}_{i,j}(n) - \mu 2_{i,j}\, \nabla_{H2_{i,j}} E(e_n^2)$$

$$= \mathbf{H2}_{i,j}(n) + 2\,\mu 2_{i,j}\, E(e_n\, \vec{Z}_i(n)\, \vec{Z}_j^T(n))$$

$$\mathbf{H2}_{i,j} = \mathbf{H2}_{j,i}, \qquad j=1..i,\ i=1..N \tag{2.3.73}$$

where $\mu 1_i$ is the adaptive step-size for the linear filter part at the i^{th} bin , and $\mu 2_{i,j}$ is the adaptive step-size of the quadratic filter part associated with the bin-pair (i,j).

*** The Normalized LMS adaptive filtering Algorithm**

In order to obtain the normalized algorithm, the step-size associated with the linear filter part at the i^{th} bin can be set as follows,

$$\mu 1_i = \mu 10_i / (ME(|z_{n,i}|^2)) \qquad i=1..N \tag{2.3.74}$$

The step-size associated with the quadratic filter part at the bin-pair (i,j) can be set as

$$\mu 2_{i,j} = \mu 20_{i,j} / (2M^2 E(|z_{n,i}|^2) E(|z_{n,j}|^2)) \qquad j=1..i,\ i=1..N \tag{2.3.75}$$

where $\mu 10_i$ and $\mu 20_{i,j}$ are constants satisfying

$$0 < \mu 10_i \le 1, \qquad 0 < \mu 20_{i,j} = \mu 20_{j,i} \le 1/2 \tag{2.3.76}$$

They control the convergence speed and the steady-state performance of the linear filter part at the i^{th} bin and of the quadratic filter part at the bin-pair (i,j), respectively. The filter coefficient update formulas (2.3.72) and (2.3.73) can then be re-written as

$$\underline{H1}_i(n+1) = \underline{H1}_i(n) + 2\,\mu 10_i\, \Lambda_i^{-2}\, E(e_n\, \underline{Z}_i(n)) \qquad i=1..N \tag{2.3.77}$$

$$\mathbf{H2}_{i,j}(n+1) = \mathbf{H2}_{i,j}(n) + \mu 20_{i,j}\, \Lambda_i^{-2}\, E(e_n\, \underline{Z}_i(n)\, \underline{Z}_j^T(n))\, \Lambda_j^{-2}$$

$$\mathbf{H2}_{i,j} = \mathbf{H2}_{j,i}, \qquad j=1..i,\ i=1..N \tag{2.3.78}$$

with the matrix Λ_i^2 is defined as

$$\Lambda_i^2 = ME(|z_{n,i}|^2)\mathbf{I}_M \qquad i=1..N \tag{2.3.79}$$

(2.3.77) and (2.3.78) can be written in the matrix forms respectively as follows

$$\vec{H1}_{n+1} = \vec{H1}_n + 2\,\mu 1\,\Lambda^{-2}\,E(e_n\vec{Z}_n) \tag{2.3.80}$$

$$\mathbf{H2}_{n+1} = \mathbf{H2}_n + \mu 2\,\Lambda^{-2}E(e_n\vec{Z}_n\vec{Z}_n^T)\Lambda^{-2} \tag{2.3.81}$$

where the matrices Λ^2, $\mu 1$ and $\mu 1$ are defined by

$$\Lambda^2 = \text{diag}[E(|z_{n,1}|^2)...E(|z_{n,N}|^2)]\otimes(M\mathbf{I}_M) = \text{diag}[\Lambda_1^2\ \Lambda_2^2\ ...\Lambda_N^2] \tag{2.3.82}$$

$$\mu 1 = \text{diag}[\mu 10_1\ \mu 10_2\ ....\ \mu 10_N]\otimes\mathbf{I}_M \tag{2.3.83}$$

$$\mu 2 = \begin{pmatrix} \mu 20_{1,1} & \mu 20_{1,2} & ... & \mu 20_{1,N} \\ \mu 20_{2,1} & \mu 20_{2,2} & ... & \mu 20_{2,N} \\ \vdots & \vdots & & \vdots \\ \mu 20_{N,1} & \mu 20_{N,2} & ... & \mu 20_{N,N} \end{pmatrix}\otimes\mathbf{I}_M \tag{2.3.84}$$

and $\mu 2$ is a symmetric matrix $\mu 20_{i,j}=\mu 20_{j,i}$. In a simplest case, one can select

$$\mu 10=\mu 10_i,\quad \mu 20=\mu 20_{i,j} \qquad 1 \le i,j \le N \tag{2.3.85}$$

which means that all the bins in the linear filter part are governed by the same step-size constant, and that all the bin-pairs in the quadratic filter part are governed by another step-size constant.

The nonlinear normalized LMS ADF algorithm in the T-TB domain, under a semi-ideal transform assumption, is summarized in Table 2.2.

Iteration at time instant n:

$$\hat{y}_n^{(1)} = tr(\mathbf{Z}_n \mathbf{H1}_n^T) = \vec{Z}_n^T \vec{H1}_n \quad (T2.2.1)$$

$$\hat{y}_n^{(2)} = tr\left[\mathbf{H2}_n (\mathbf{Z2}_n - \mathbf{R}_{\vec{Z}\vec{Z}}(n))^T\right] = \vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n - E(\vec{Z}_n^T \mathbf{H2}_n \vec{Z}_n) \quad (T2.2.2)$$

$$e_n = d_n - \hat{y}_n^{(1)} - \hat{y}_n^{(2)} \quad (T2.2.3)$$

$$\vec{H1}_{n+1} = \vec{H1}_n + 2\,\mu 1\, \Lambda^{-2} E(e_n \vec{Z}_n) \quad (T2.2.4)$$

$$\mathbf{H2}_{n+1} = \mathbf{H2}_n + \mu 2\, \Lambda^{-2} E(e_n \vec{Z}_n \vec{Z}_n^T) \Lambda^{-2} \quad (T2.2.5)$$

where: (T2.2.6)

$$\Lambda^2 = diag[E(|z_{n,1}|^2) \ldots E(|z_{n,N}|^2)] \otimes (M\mathbf{I}_M) \quad (T2.2.7)$$

$$\mu 1 = diag[\mu 10_1\ \mu 10_2 \ldots \mu 10_N] \otimes \mathbf{I}_M \quad (T2.2.8)$$

$$\mu 2 = \begin{pmatrix} \mu 20_{1,1} & \mu 20_{1,2} & \ldots & \mu 20_{1,N} \\ \mu 20_{2,1} & \mu 20_{2,2} & \ldots & \mu 20_{2,N} \\ \vdots & \vdots & & \vdots \\ \mu 20_{N,1} & \mu 20_{N,2} & \ldots & \mu 20_{N,N} \end{pmatrix} \otimes \mathbf{I}_M \quad (T2.2.9)$$

$$0 < \mu 10_i \le 1 \text{ and } 0 < \mu 20_{i,j} \le 1/2 \quad \mu 20_{i,j} = \mu 20_{j,i} \quad (T2.2.10)$$

a simplest selection: $\mu 10 = \mu 10_i$, $\mu 20 = \mu 20_{i,j}$ i,j=1..N

Table 2.2 A *nonlinear* NLMS ADF algorithm in a T-TB domain under the semi-ideal transform assumption

2.3.4.4. Some properties of the T-TB domain nonlinear NLMS adaptive filtering algorithm

1) Decorrelation of the filter coefficients

* Linear filter part **H1**

Under a semi-ideal transform assumption, the linear filter part **H1** is decorrelated along the bin direction. Consequently, **H1** in the T-TB domain become N independent sub-vectors, each with M-elements.

* Quadratic filter part **H2**

Under a semi-ideal transform assumption, all the quadratic filter coefficients associated with different bin-pair are decorrelated. **H2** becomes N*N independent sub-matrices. Consequently, the quadratic filter part **H2** in the T-TB domain reduces to *N(N+1)/2 independent sub-matrices of size M*M*. Among them, N sub-matrices are symmetric thus have M(M+1)/2 unknown elements, the remaining ones M*M elements. Hence, the time-domain quadratic filter **h2** (equivalent to a vector of MN(MN+1)/2 coefficients) is partially decorrelated in the T-TB domain.

2) Possible reduction of the number of quadratic filter coefficients

Although the signal components are *linearly* decorrelated along the transform-bin direction, there still exist *NL* correlations between various bin-pairs. In general, all the different *bin-pair* combinations are possible.

Fortunately, in most cases there are some constraints on the NL filter coefficients according to the physical interpretations and the specific domain selected. Much less bin-pairs can then be used in the T-TB domain.

*** Quadratic filter coefficient constraint in the DSTFT-type T-FB domain**

Consider the DSTFT-type Time-frequency Bin (T-FB) domain. For a specific DSTFT transform satisfying the semi-ideal transform assumption, the base function is $\{w_{j,k}(t)=w(t+k)\exp(i2\pi jt)\}$, where $w(t)=\mathrm{sinc}(t)$, and j denotes the non-overlapping and orthogonal frequency bin, k is the time-index in the center of the window function w. From this base function, the following relation can be obtained

$$w_{j_1+j_2,k} = \left(\frac{1}{\sqrt{w(t+k)}}\, w_{j_1,k}\right)\left(\frac{1}{\sqrt{w(t+k)}}\, w_{j_2,k}\right) \qquad (2.3.86)$$

This implies that for an arbitrary bin j, only signal components from the two separate bins j_1 and j_2, $j=(j_1+j_2)$, may have (quadratic) NL correlation with the signal components of bin j. Hence, only the quadratic filter coefficients in the **block-matrices** $\mathbf{H2}_{j_1,j_2}$ which satisfy the frequency constraint $j=(j_1+j_2)$, $1\le j_1,j_2\le N$, can be selected. However, within each bin-pair (thus, in each block-matrix), signal components in the different bins are generally nonlinear-correlated along the time-direction.

3) Convergence speed

It can be expected, *for* **stationary** *signals, that the filter convergence speed in a T-TB domain is a median value between that in the time-domain and in the transform-domain.* This is likely because, in contrast to fully correlated data in the time-domain and fully decorrelated data in the transform-domain (KLT-domain), the T-TB domain data under a semi-ideal transform assumption is only decorrelated in one direction.

Analytically, the convergence speed of the NL NLMS ADF algorithm in a T-TB domain depends on the maximum- and minimum-eigenvalue ratio ($\lambda_{max}/\lambda_{min}$) and on the squared-ratio of $(\Lambda^{-2}\mathbf{R}_{\vec{z}\vec{z}})$ for linear and quadratic filter part, respectively. By selecting a semi-ideal transform $\mathbf{W}$, the signal components can be linearly decorrelated along the bin direction, resulting in a block diagonal matrix $\mathbf{R}_{\vec{z}\vec{z}}$ as in formula (2.3.71). After normalization, the eigenvalue spread of $(\Lambda^{-2}\mathbf{R}_{\vec{z}\vec{z}})$ is partially reduced. Hence, from this comes to the same conclusion as above.

4) Algorithm degeneration to the transform-domain

If M=1 is selected, which means only one frame of transformed data is considered, the algorithm degenerates to a corresponding transform-domain filter. This is similar to the T-TB domain Linear ADF case.

2.3.4.5. Relations and similarities among the linear and nonlinear filtering algorithms in the T-TB domain and in the transform-domain

In the following we will summarize the relations among the above four algorithms, i.e. the transform-domain linear NLMS ADF algorithm; the transform-domain nonlinear NLMS ADF algorithm; the time-transform bin domain linear NLMS ADF algorithm; and the time-transform bin domain nonlinear NLMS ADF algorithm.

1) Formula relations between the linear and NL filtering algorithms

The linear filter is equivalent to the linear filter part of the nonlinear second-order volterra filter under Gaussian assumption. This holds either for the transform-domain, or for the T-TB domain. This implies that the transform-domain (or the T-TB domain) linear filter is involved by the transform-domain (or the T-TB domain) nonlinear filter.

2) Formula similarities between the transform-domain and the T-TB domain algorithms

If the transform-domain vectors $\underset{\sim}{Z}_n$, $\underset{\sim}{H}_n^{(1)}$, and matrices $\mathbf{H}_n^{(2)}$, $\mathbf{R}_{zz}(n)$, $\mu 1$, $\mu 2$, Λ^2, $\mathbf{W}$ are replaced by the following T-TB domain vectors $\vec{Z}_n$, $\vec{H1}_n$, and matrices $\mathbf{H2}_n$, $\mathbf{R}_{\vec{z}\vec{z}}(n)$, $\mu 1 \otimes \mathbf{I}_M$, $\mu 2 \otimes \mathbf{I}_M$, $\Lambda^2 \otimes (M\mathbf{I}_M)$, $\mathbf{W2}=(\mathbf{W}\otimes \mathbf{I}_M)$, respectively, the formulas in the transform-domain algorithms directly correspond to the formulas in the T-TB domain algorithms. This is associated with a dimension-expansion in the transform-domain, along the time direction, to a T-TB domain.

In (chapter 6) Appendix, the formulas of these different algorithms and their corresponding relations will be synthesized.

2.3.4.6. An Example

A simple example is given below to show how to use the T-TB domain NL NLMS ADF algorithm, given sequence of data.

Given the observed signal sequence $\{y_k\}$ which is a desired signal corrupted by additive noise, i.e, $y_k=s_k+n_k$, as well as the noise correlated observation $\{x_k\}$. The signal s_k has zero-mean and is uncorrelated with the noise n_k and x_k

(n_k and x_k are correlated). The data sequence $\{...x_n \; x_{n-1}...x_{n-N+1}... \; x_{n-L}...\}$ is Gaussian zero-mean. An unknown NL system associated x_k and n_k as its input and output respectively. Thus, a NL NLMS ADF algorithm can be used as an NL Adaptive Noise Canceler (ANC). This NL ANC uses y_k and x_k as the primary and the reference input, respectively, and has $e_k=\hat{s}_k$ as its output, as shown in Fig.2.1.

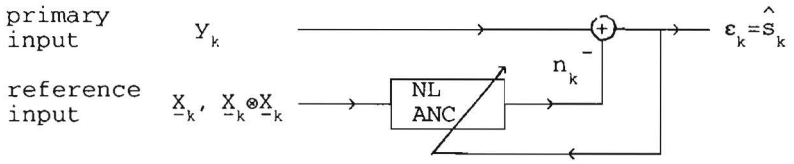


Fig.2.1 An example of a NL Adaptive Noise Canceler

The NL ADF minimizes the following mean square error

$$J1_k = E(e_k^2) = E[(y_k-\hat{n}_k)^2] \tag{2.3.87}$$

Because s_k is zero mean and uncorrelated with n_k and x_k, it equals to minimize $J2_k$ as below

$$J2_k = E(n_k-\hat{n}_k)^2 \tag{2.3.88}$$

Suppose that a T-TB domain gradient type NL NLMS adaptive filtering algorithm will be used. The noise n_k can be modeled by a second-order NL Volterra filter of size L=MN=6. The analysis frame length is chosen N=3, thus the time directional order needed to compensate the system impulse response is M=2. An orthogonal transform $\mathbf{W}_3$ is chosen as

$$\mathbf{W}_3 = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \tag{2.3.89}$$

The data matrices before and after the transform, $\mathbf{X}_n$ and $\mathbf{Z}_n$, are arranged respectively as follows

$$\mathbf{X}_n = \begin{pmatrix} x_{11}(n) & x_{12}(n) & x_{13}(n) \\ x_{21}(n) & x_{22}(n) & x_{23}(n) \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & x_{n-2} \\ x_{n-3} & x_{n-4} & x_{n-5} \end{pmatrix} \tag{2.3.90}$$

$$\mathbf{Z}_n = \begin{pmatrix} z_{11}(n) & z_{12}(n) & z_{13}(n) \\ z_{21}(n) & z_{22}(n) & z_{23}(n) \end{pmatrix} = \begin{pmatrix} z_n & z_{n-1} & z_{n-2} \\ z_{n-3} & z_{n-4} & z_{n-5} \end{pmatrix} \tag{2.3.91}$$

The relation $\mathbf{Z}_n = \mathbf{X}_n \mathbf{W}^T$ can be written as follows

$$\begin{pmatrix} z_n & z_{n-1} & z_{n-2} \\ z_{n-3} & z_{n-4} & z_{n-5} \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & x_{n-2} \\ x_{n-3} & x_{n-4} & x_{n-5} \end{pmatrix} \begin{pmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} \sum_{i=1}^{3} x_{n+1-i} w_{1i} & \sum_{i=1}^{3} x_{n+1-i} w_{2i} & \sum_{i=1}^{3} x_{n+1-i} w_{3i} \\ \sum_{i=1}^{3} x_{n-N+1-i} w_{1i} & \sum_{i=1}^{3} x_{n-N+1-i} w_{2i} & \sum_{i=1}^{3} x_{n-N+1-i} w_{3i} \end{pmatrix} \tag{2.3.92}$$

One iteration step of the gradient-type NL NLMS ADF algorithm now yields:

(1) Transform a new incoming data frame $\{x_{11}(n)\ x_{12}(n)\ x_{13}(n)\}$ as follows (overlap=0)

$$\begin{pmatrix} z_{11} \\ z_{12} \\ z_{13} \end{pmatrix}_n = \mathbf{W}_3 \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix}_n \tag{2.3.93}$$

(2) Arrange the new transformed-data matrix $\mathbf{Z}_n$ and the vector $\vec{Z}_n$

Delete the oldest data frame by shifting $\mathbf{Z}_{n-1}$ matrix (delete one bottom row, and add a new row on the top). Column-scan the new matrix $\mathbf{Z}_n$ to obtain $\vec{Z}_n$.

(3) Calculate $\Lambda^{-2}\ \vec{Z}_n$

$$\Lambda^{-2}\vec{Z}_n = \begin{pmatrix} \Lambda_1^2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \Lambda_2^2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \Lambda_3^2 \end{pmatrix}^{-1} \begin{pmatrix} z_{1,1} \\ z_{2,1} \\ z_{1,2} \\ z_{2,2} \\ z_{1,3} \\ z_{2,3} \end{pmatrix}_n \tag{2.3.94}$$

where $\Lambda_i^2 = \begin{pmatrix} 2E|z_{n,i}|^2 & 0 \\ 0 & 2E|z_{n,i}|^2 \end{pmatrix}$, i=1..3.

(4) Calculate the linear filter output $\hat{y}_n^{(1)}$

$$\hat{y}_n^{(1)} = \vec{H1}_n^T \vec{Z}_n \tag{2.3.95}$$

(5) Calculate the quadratic filter output $\hat{y}_n^{(2)}$

$$\hat{y}_n^{(2)} = \mathrm{tr}(\mathbf{H2}_n(\begin{pmatrix} Z_1Z_1^T & Z_1Z_2^T & \cdots & Z_1Z_N^T \\ Z_2Z_1^T & Z_2Z_2^T & \cdots & Z_2Z_N^T \\ \vdots & \vdots & & \vdots \\ Z_NZ_1^T & Z_NZ_2^T & \cdots & Z_NZ_N^T \end{pmatrix}_n - \begin{pmatrix} E(Z_1Z_1^T) & & & \mathbf{0} \\ & E(Z_2Z_2^T) & & \\ & & \ddots & \\ \mathbf{0} & & & E(Z_NZ_N^T) \end{pmatrix})^T) \tag{2.3.96}$$

(8) Calculate the filter output error e_n (the estimated signal $\hat{s}_n = e_n$)

$$e_n = y_n - \hat{y}_n^{(1)} - \hat{y}_n^{(2)} \tag{2.3.97}$$

(9) Update $\vec{H1}$

$$\begin{pmatrix} H1_{1,1} \\ H1_{2,1} \\ H1_{1,2} \\ H1_{2,2} \\ H1_{1,3} \\ H1_{2,3} \end{pmatrix}_{(n+1)} = \begin{pmatrix} H1_{1,1} \\ H1_{2,1} \\ H1_{1,2} \\ H1_{2,2} \\ H1_{1,3} \\ H1_{2,3} \end{pmatrix}_{(n)} + 2e_n \begin{pmatrix} \mu10_1 & & \mathbf{0} & & \mathbf{O} \\ & \mu10_1 & & & \\ \mathbf{0} & & \mu10_2 & \mathbf{0} & \\ & & & \mu10_2 & \\ \mathbf{0} & & \mathbf{0} & & \mu10_3 \\ & & & & \mu10_3 \end{pmatrix} \Lambda^{-2}\vec{Z}_n$$

$$0 < \mu10_j \le 1 \qquad j=1..3 \tag{2.3.98}$$

(10) Update **H2**

$$\mathbf{H2}_{n+1} = \mathbf{H2}_n + \mu 20\ e_n(\Lambda^{-2}\vec{Z}_n)(\Lambda^{-2}\vec{Z}_n)^T$$

where $\mathbf{H2}_n =$

$$\left(\begin{array}{cc|cc|cc} H2_{1,1} & H2_{1,2} & H2_{1,3} & H2_{1,4} & H2_{1,5} & H2_{1,6} \\ H2_{2,1} & H2_{2,2} & H2_{2,3} & H2_{2,4} & H2_{2,5} & H2_{2,6} \\ \hline H2_{3,1} & H2_{3,2} & H2_{3,3} & H2_{3,4} & H2_{3,5} & H2_{3,6} \\ H2_{4,1} & H2_{4,2} & H2_{4,3} & H2_{4,4} & H2_{4,5} & H2_{4,6} \\ \hline H2_{5,1} & H2_{5,2} & H2_{5,3} & H2_{5,4} & H2_{5,5} & H2_{5,6} \\ H2_{6,1} & H2_{6,2} & H2_{6,3} & H2_{6,4} & H2_{6,5} & H2_{6,6} \end{array}\right)_{(n)}$$

$$0 < \mu 20 \le 1/2 \qquad (\text{ here using } \mu 20=\mu 20_{i,j} \quad 1 \le i,j \le N) \qquad (2.3.99)$$

(11) Setting a new time instant
n:=n+N, repeat (1)-(10).

Further discussions:

(1)If the time-DCT bin domain is chosen, only the sub-matrices $\mathbf{H2}_{1,1}$ and $\mathbf{H2}_{1,2}$ ($\mathbf{H2}_{1,2}=\mathbf{H2}_{2,1}$), corresponding to the bin-pairs (1,1) and (1,2), can be selected, because of the frequency constraint $1\le(i+j)\le3$. Thus, step (10) can be simplified.

(2)If the time-WHT bin domain is chosen, only sub-matrices the $\mathbf{H2}_{1,2}$, $\mathbf{H2}_{1,3}$ and $\mathbf{H2}_{2,3}$, corresponding to bin-pair (1,2), (1,3) and (2,3), need to be selected as the quadratic terms due to the constraint $1\le(i\oplus j)=\le3$. Thus, step (10) is simplified because only three sub-matrices in **H2** are concerned.

2.3.4.7. Summary

In this section, we have derived a *new* T-TB domain *nonlinear* (Volterra type) NLMS ADF algorithm under a semi-ideal transform assumption. It actually is a time-transform domain NL NLMS FIR filtering algorithm. The complexity of this algorithm is between that of the corresponding algorithms in the time- and in the transform-domain. Many attractive properties holds in the T-TB

domain, such as:

* **The linear and quadratic filters are decoupled**, which is a consequence of the time-domain Gaussian input data assumption and the linear transform property of any orthogonal transform **W**.
  Consequently, the linear filter part of this NL filter has the same properties as those in the linear filter in section 2.2.3. In fact, the T-TB domain *linear* filtering algorithm is a subset (the linear part) of the corresponding *nonlinear* filtering algorithm.

* **The signal components are fully decorrelated along the bin-direction,** provided that a semi-ideal transform is selected. Consequently, The filter coefficients become **independent blocks of sub-vectors/ sub-matrices**, which are much easier to be solved.

* **Quadratic filter coefficient number can be greatly reduced** depending on a specific chosen transform **W**.

* The **convergence speed** is in between that of the time-domain and that of the transform-domain for **stationary** signals.

* The algorithm is suitable for filtering **nonstationary** signals and signals associated with a long impulse response length.

* Signal overlapping and windowing can be used when needed (similar to section 2.2.3).

* The T-TB domain algorithm can degenerate to a transform-domain one.

* **A T-TB domain nonlinear NLMS ADF algorithm is a generalized form**
- *It involves the T-TB domain linear NLMS adaptive filtering algorithm, and it can degenerate to the transform-domain linear and NL NLMS ADF algorithms.*

. By neglecting the quadratic filter part in the T-TB domain (or in the transform-domain) NL algorithm, the algorithm degenerates to the T-TB domain (or to the transform-domain) linear version.
. By finding the similarities between the corresponding variables (vectors, matrices) and substituting them into the corresponding formulas in the T-TB domain (or in the transform-domain) the transform-domain (or the T-TB

domain) algorithm is then obtained (as described in section 2.3.4.5). It is obvious that a T-TB domain nonlinear NLMS adaptive filtering algorithm is a general form of the transform-domain linear and nonlinear NLMS algorithms.

2.4. RECURSIVE LEAST SQUARE LINEAR ADAPTIVE FILTERING

A Recursive Least Squares (RLS) sliding-window covariance lattice filter has been extended with the **new** *function of* **adaptive window-length** *based on the vector space geometric projection. Such an added new functionality improves the filter performance of tracking nonstationary signals, especially when the signal statistics have non-constant variation speed.*

2.4.1. Introduction

In section 2.2 and 2.3, we have investigated LMS type adaptive filters. For the purpose of processing nonstationary signals and signals with a long impulse response length, we have concentrated mainly on developing new time-transform domain algorithms.

However, in some situations, nonstationary signals to be filtered have fast time-varying statistics. Thus, fast convergence is the main problem. A filter with relatively slow convergence (e.g. an LMS type filter) could always remain in the adaptation process, i.e. far from the ideal solution. In such a case, one should choose other alternatives, such as RLS type filters.
RLS filters have drawn much attention due to their fast convergence, the exact Least Square (LS) error calculations, and not being hampered by Gaussian data limitation (needed for the nonlinear LMS filters).

Selection of LMS or RLS type of filter: tradeoffs between convergence speed and filter complexity
As has been mentioned before, the LMS type of filter enjoys simplicity and robustness. The convergence speed of LMS filters can be improved by using a normalized version of some properly selected transform domain. On the other hand, RLS filters generally have faster convergence. They perform exact Least Square (LS) calculations at each time instant, and are free from the restriction to Gaussian input data. However more calculations are usually needed for the RLS type of filter.
Hence, the selection of an LMS type or an RLS type of adaptive filter depends on the application demands. Generally speaking, when a nonstationary signal has slowly changing time-varying statistics, one should choose an LMS type

filter; for fast changing nonstationary signals an RLS type filter should be selected. The main tradeoff for selection is the convergence speed and the algorithm's complexity.

2.4.2. RLS sliding-window covariance lattice filter with an adaptive window length

Among all kinds of *linear* RLS adaptive filters, the lattice filters are particularly attractive[24]. One of their main advantages is the mutual orthogonality of filter outputs at different orders. Thus, the filter is decoupled among different orders. This implies that a globally optimal filter can be implemented by choosing the local optimum for each order. A second main advantage is the numerical stability of this filter under finite length calculations, and the low sensitivity of the filter parameters to small disturbances such as caused by quantization. The difference among the various kinds of lattice filter results from the use of different windowed data for filter parameter estimation. The sliding-window covariance lattice filter[24,75] uses a constant-length data-window, which slides forward at each time instant. However, when *signal statistics are time-varying with non-constant changing speed,* it is desirable that the window size can be adjusted at the same time.

Motivated by this, we have developed a new adaptive sliding-window covariance lattice filter algorithm, which is an extension of the previously existing filter with constant size of the sliding window [24,75]. The basic idea is that the window length can be decreased recursively if the time-varying speed of the signal statistics is increased in a short time duration. While the data window length can be recursively increased when signal approaches stationarity, so that the error-variance can be reduced.

The key for the derivations is to find iterative relations among the variables corresponding to the different window w(t) and w(t+1) in the successive time instant, by using geometric projection update formulas. The corresponding derivations are not presented here -the interested reader is referred to [29] for further details.

2.5. RECURSIVE LEAST SQUARE NONLINEAR ADAPTIVE FILTERING

A new RLS Volterra type of **nonlinear** *adaptive filter with adaptive-sliding-window is developed in this section.*

By introducing a *finite data memory, and allowing recursive adaptation of* the *window-length* of *this memory, the new RLS NL filtering algorithm provides versatile capabilities for tracking nonstationary signals associated with a NL time-varying model having non-constant rate of changing speed.*

2.5.1. Introduction

There are many different types of NL filters. We will again concentrate on the NL Volterra filter due to the same reason mentioned before.

For RLS NL filters, very little investigation has been done on adaptively tracking the time-varying NL parameters. Recently, Mathews and Lee[58] presented a fast RLS adaptive Volterra filtering algorithm, and Giannakis and Dandawate proposed a RLS NL adaptive noise canceler[26,27]. In both algorithms, *prewindowed exponentially weighted data* is used.

Often, the signal to be filtered is nonstationary and its time-varying statistics has non-constant changing speed. Previous research on RLS NL filters was mainly associated with prewindowed data. It becomes unsuitable to memorize the infinite amount of past data in the nonstationary case. A corresponding recursive algorithm associated with an adaptive finite window length[32] is developed in this section.

The remaining part of section 2.5 will be organized as follows. First the adaptive NL RLS algorithm with a Sliding Window (SW) of constant length will be derived. Then the algorithm will be extended to an adaptive-window length. The simulation results will demonstrate the performance of the filter, with comparisons to that of the corresponding prewindowed one. Finally, some concluding remarks will be given.

2.5.2. RLS nonlinear ADF algorithm with an adaptive sliding-window

2.5.2.1. The algorithm for constant window length

Consider the general problem, where a NL filter needs to be explored. The nonlinearity can often be represented by a Volterra series expansion. Using a second-order Volterra kernel, the desired response d_n can be approximately represented by the output of a truncated N-order filter

$$\hat{y}_n = h_0 + \sum_{m_1=1}^{N} h^{(1)}_{m_1}(n) x_{n-m_1+1} + \sum_{m_1, m_2=1}^{N} h^{(2)}_{m_1,m_2}(n) x_{n-m_1+1} x_{n-m_2+1} \qquad (2.5.1)$$

or equivalently, in vector form

$$\hat{y}_n = \underline{h}_n^T \underline{Z}_n \qquad (2.5.1')$$

where $\underline{h}_n = [h^{(1)}_1(n) .. h^{(1)}_N(n) \; h^{(2)}_{1,1}(n) .. h^{(2)}_{N,N}(n)]^T = [\underline{h}^{(1)}_n \;\; \underline{h}^{(2)}_n]^T$, $\underline{Z}_n = [\underline{X}_n \;\; \underline{X}_n \otimes \underline{X}_n]^T$, $\otimes$ is the Kronecker product, and $\underline{X}_n = [x_n \; x_{n-1} \; ... x_{n-N+1}]^T$. The Least Square (LS) estimation under consideration is to find at each time instant n, **a sliding windowed**, RLS solution of the optimal coefficient vector $\underline{h}_n$, such that the following cost function is minimized for a fixed Window Length WL=(L+1)

$$J_n = \sum_{k=n-L}^{n} (d_k - \hat{y}_k)^2 \qquad (2.5.2)$$

For a time-varying NL system, the WL is chosen such that signals inside the window can be considered stationary. By taking partial derivative $\nabla_{\underline{h}}(J_n)$ with respect to $\underline{h}_n$ and setting it to zero, it can be proved that the optimal solution is

$$\underline{h}_{opt} = \mathbf{R}_n^{-1} \underline{P}_n \qquad (2.5.3)$$

where

$$\mathbf{R}_n = \sum_{k=n-L}^{n} \underline{Z}_k \underline{Z}_k^T = \sum_{k=n-L}^{n} \begin{pmatrix} \underline{X}_k \underline{X}_K^T & \underline{X}_k (\underline{X}_k \otimes \underline{X}_k)^T \\ (\underline{X}_k \otimes \underline{X}_k) \underline{X}_k^T & (\underline{X}_k \otimes \underline{X}_k)(\underline{X}_k \otimes \underline{X}_k)^T \end{pmatrix}$$

$$\underline{P}_n = \sum_{k=n-L}^{n} y_k \underline{Z}_k = \sum_{k=n-L}^{n} y_k \begin{pmatrix} \underline{X}_k \\ \underline{X}_k \otimes \underline{X}_k \end{pmatrix} \qquad (2.5.4)$$

If the signal is stationary, (2.5.3) is independent of time instant n, otherwise it is an optimal solution with respect to the given data inside the

window. A pseudo matrix inversion $\mathbf{R}_n^+$ is used in (2.5.3) when $\mathbf{R}_n^{-1}$ is singular or ill-conditioned. (2.5.4) can be expressed in the recursive form as

$$\mathbf{R}_{n+1} = \mathbf{R}_n + \underline{Z}_{n+1}\underline{Z}_{n+1}^T - \underline{Z}_{n-L}\underline{Z}_{n-L}^T \tag{2.5.5}$$

$$\underline{P}_{n+1} = \underline{P}_n + y_{n+1}\underline{Z}_{n+1} - y_{n-L}\underline{Z}_{n-L} \tag{2.5.6}$$

In order to derive an update formula for the $\underline{h}_{n+1}$ using present data x_{n+1} and the iteration results obtained at time instant n, the following auxiliary variables are introduced

$$\tilde{\mathbf{R}}_n = \mathbf{R}_n - \underline{Z}_{n-L}\underline{Z}_{n-L}^T \tag{2.5.7}$$

$$\tilde{\underline{P}}_n = \underline{P}_n - y_{n-L}\underline{Z}_{n-L} \tag{2.5.8}$$

with these, (2.5.5) and (2.5.6) can be expressed as follows

$$\mathbf{R}_{n+1} = \tilde{\mathbf{R}}_n + \underline{Z}_{n+1}\underline{Z}_{n+1}^T \tag{2.5.9}$$

$$\underline{P}_{n+1} = \tilde{\underline{P}}_n + y_{n+1}\underline{Z}_{n+1} \tag{2.5.10}$$

next, by using the following matrix inversion lemma

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{DA}^{-1} \tag{2.5.11}$$

and choosing the following associations to (2.5.11)

$$\mathbf{A} = \mathbf{R}_n,\ \mathbf{B} = \underline{Z}_{n-L},\ \mathbf{C} = -1,\ \mathbf{D} = \underline{Z}_{n-L}^T \tag{2.5.12}$$

we obtain

$$\tilde{\mathbf{R}}_n^{-1} = \mathbf{R}_n^{-1} + \frac{\mathbf{R}_n^{-1}\underline{Z}_{n-L}\underline{Z}_{n-L}^T\mathbf{R}_n^{-1}}{1 - \underline{Z}_{n-L}^T\mathbf{R}_n^{-1}\underline{Z}_{n-L}} \tag{2.5.13}$$

$\tilde{\underline{h}}_n = \tilde{\mathbf{R}}_n^{-1}\tilde{\underline{P}}_n$ can then be expressed as

$$\tilde{\underline{h}}_n = (\mathbf{R}_n^{-1} + \frac{\mathbf{R}_n^{-1}\underline{Z}_{n-L}\underline{Z}_{n-L}^T\mathbf{R}_n^{-1}}{1 - \underline{Z}_{n-L}^T\mathbf{R}_n^{-1}\underline{Z}_{n-L}})(\underline{P}_n - y_{n-L}\underline{Z}_{n-L}) \tag{2.5.14}$$

After simplification, this becomes

$$\tilde{\underline{h}}_n = \underline{h}_n - \underline{K}_n\eta_n^b \tag{2.5.15}$$

where the gain $\tilde{\underline{K}}_n$ and the backward filter residual η_n^b are given by

$$\tilde{\underline{K}}_n = \frac{\mathbf{R}_n^{-1}\,\underline{Z}_{n-L}}{1 - \underline{Z}_{n-L}^T\,\mathbf{R}_n^{-1}\,\underline{Z}_{n-L}} \tag{2.5.16}$$

$$\eta_n^b = y_{n-L} - \underline{Z}_{n-L}^T\,\underline{h}_n \tag{2.5.17}$$

Similarly, using (2.5.9) and making the following associations

$$\mathbf{A} = \tilde{\mathbf{R}}_n\ ,\ \mathbf{B} = \mathbf{Z}_{n+1},\ \mathbf{C} = 1,\ \mathbf{D} = \mathbf{Z}_{n+1}^T \tag{2.5.18}$$

to (2.5.11), the update formulas for K_{n+1}, R_{n+1}^{-1}, $\mathbf{h}_{n+1}$ and the forward prediction error e_{n+1}^f can be obtained. Table 2.3 summarizes this RLS SW NL filtering algorithm.

Algorithm 1: constant window length

Recursive step at time (n+1): (WL=L+1)

$$\tilde{\underline{K}}_n = \frac{\mathbf{R}_n^{-1}\,\underline{Z}_{n-L}}{1 - \underline{Z}_{n-L}^T\,\mathbf{R}_n^{-1}\,\underline{Z}_{n-L}} \qquad \text{(T2.3.1)}$$

$$\tilde{\mathbf{R}}_n^{-1} = (\mathbf{I} + \tilde{\underline{K}}_n\,\underline{Z}_{n-L}^T)\,\mathbf{R}_n^{-1} \qquad \text{(T2.3.2)}$$

$$\eta_n^b = y_{n-L} - \underline{Z}_{n-L}^T\,\underline{h}_n \qquad \text{(T2.3.3)}$$

$$\tilde{\underline{h}}_n = \underline{h}_n - \tilde{\underline{K}}_n\,\eta_n^b \qquad \text{(T2.3.4)}$$

$$\underline{K}_{n+1} = \frac{\tilde{\mathbf{R}}_n^{-1}\,\underline{Z}_{n+1}}{1 + \underline{Z}_{n+1}^T\,\tilde{\mathbf{R}}_n^{-1}\,\underline{Z}_{n+1}} \qquad \text{(T2.3.5)}$$

$$\mathbf{R}_{n+1}^{-1} = (\mathbf{I} - \underline{K}_{n+1}\,\underline{Z}_{n+1}^T)\,\tilde{\mathbf{R}}_n^{-1} \qquad \text{(T2.3.6)}$$

$$e_{n+1}^f = y_{n+1} - \underline{Z}_{n+1}^T\,\tilde{\underline{h}}_n \qquad \text{(T2.3.7)}$$

$$\underline{h}_{n+1} = \tilde{\underline{h}}_n + \underline{K}_{n+1}\,e_{n+1}^f \qquad \text{(T2.3.8)}$$

output: $\underline{h}_{n+1}$ (WL=L+1)

(where $\underline{Z}_{n+1} = [\underline{X}_{n+1}\ \underline{X}_{n+1} \otimes \underline{X}_{n+1}]^T$

and $\underline{X}_{n+1} = [x_{n+1}\ x_n\ \ldots x_{n-N+2}]^T$)

Table 2.3 RLS sliding-window nonlinear ADF algorithm

Remarks

R1. Generalize to the p^{th}-order NL filter

To generalize an NL filter algorithm to a p^{th} order Volterra kernel, one only needs to rewrite the vector $\underline{Z}$ as follows

$$\underline{Z}_n=[\underline{X}_n,\ \underline{X}_n\otimes\underline{X}_n,\ ...,\ \underbrace{\underline{X}_n\otimes\underline{X}_n\otimes...\otimes\underline{X}_n\otimes\underline{X}_n}_{p}]^T \qquad (2.5.19)$$

The other formulas in the algorithm will remain the same.

R2. Trade-off between tracking capabilities and calculation cost

The improved performance is obtained at the price of more calculations. The amount of calculations is almost doubled compared to the prewindowed method. However, it is still attractive for a NL system with low-order kernel, with partial NL coefficients, and with non-constant rate of changing time-varying speed.

R3. Initial value

The initial condition for the algorithm is $\underline{h}_0=\mathbf{0}$, $\mathbf{R}_0=\delta\ \mathbf{I}$. ($\delta$ is a small positive constant)

R4. The filtering domain and the related complexity

In the linear situation, because of the fast convergence speed of the RLS algorithm, it is usually not necessary to filter signals in the transform-domain. However, for a NL filter, the situation is slightly different. One might be interested in using a NL filter in other domains, such as in the frequency-domain[65]. Some benefit may then be obtained from performing filtering in the transform-domain. For each selected transform, as mentioned before, the number of NL coefficients may be significantly reduced, compared with the situation that all NL coefficients in the time-domain filter are chosen.

For example, in the frequency-domain only the p^{th}- and lower-order NL coefficients $H_{f_1 f_2 ... f_l}(n)$, l=2,3...p, satisfying the frequency constraint $0\le \sum_j f_j \le (N-1)$, will be contained in the filter (where N is the total frequency bin number, p is the order of Volterra kernel).

Discussion of the Gaussian input case

If the input variable x_n has Gaussian distribution, all the odd moments of $\underline{X}_k$ disappear. Due to the diagonal matrix $\mathbf{R}_n(2)$, the second-order Volterra filtering algorithm can be simplified. Below, the formulas associated with the Gaussian input case will be derived.

By taking partial derivatives to (2.5.2) with respect to $\underline{h}$ and setting them to zero, the detailed equations can be re-written as follows

$$\nabla_{\underline{h}}(J) = \sum_{k=n-L}^{n} \left\{ \begin{pmatrix} \underline{X}_k \\ \underline{X}_k \otimes \underline{X}_k \end{pmatrix} (y_k - [\underline{X}_k^T \; (\underline{X}_k \otimes \underline{X}_k)^T] \begin{pmatrix} \underline{h}_k(1) \\ \underline{h}_k(2) \end{pmatrix}) \right\}$$

$$= \sum_{k=n-L}^{n} \left\{ \begin{pmatrix} y_k \underline{X}_k \\ y_k (\underline{X}_k \otimes \underline{X}_k) \end{pmatrix} - \begin{pmatrix} \underline{X}_k \\ \underline{X}_k \otimes \underline{X}_k \end{pmatrix} [\underline{X}_k^T \; (\underline{X}_k \otimes \underline{X}_k)^T] \begin{pmatrix} \underline{h}_k(1) \\ \underline{h}_k(2) \end{pmatrix} \right\}$$

$$= \begin{pmatrix} \underline{P}_n(1) \\ \underline{P}_n(2) \end{pmatrix} - \begin{pmatrix} \mathbf{R}_n(2) & \mathbf{R}_n^T(3) \\ \mathbf{R}_n(3) & \mathbf{R}_n(4) \end{pmatrix} \begin{pmatrix} \underline{h}_n(1) \\ \underline{h}_n(2) \end{pmatrix} = 0 \qquad (2.5.20)$$

Where $\mathbf{R}_n(2) = \sum_{k=n-L}^{n} \underline{X}_k \underline{X}_k^T$, $\mathbf{R}_n(3) = \sum_{k=n-L}^{n} (\underline{X}_k \otimes \underline{X}_k) \underline{X}_k^T$, $\mathbf{R}_n(4) = \sum_{k=n-L}^{n} (\underline{X}_k \otimes \underline{X}_k)(\underline{X}_k \otimes \underline{X}_k)^T$, $\underline{P}_n(1) = \sum_{k=n-L}^{n} y_k \underline{X}_k$, $\underline{P}_n(2) = \sum_{k=n-L}^{n} y_k (\underline{X}_k \otimes \underline{X}_k)$.

Equation (2.5.20) is equivalent to the following two vector equations

$$\underline{P}_n(1) - \mathbf{R}_n(2)\underline{h}_n(1) - \mathbf{R}_n(3)\underline{h}_n(2) = 0 \qquad (2.5.21)$$

$$\underline{P}_n(2) - \mathbf{R}_n(3)\underline{h}_n(2) - \mathbf{R}_n(4)\underline{h}_n(2) = 0 \qquad (2.5.22)$$

Hence, general, $\underline{h}_n(1)$ and $\underline{h}_n(2)$ are coupled as indicated. However when $\underline{X}_n$ is zero-mean Gaussian, all the odd moments of $\underline{X}_n$ are zero, so the equations can be simplified to

$$\begin{pmatrix} \underline{P}_n(1) \\ \underline{P}_n(2) \end{pmatrix} - \begin{pmatrix} \mathbf{R}_n(2) & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_n(4) \end{pmatrix} \begin{pmatrix} \underline{h}_n(1) \\ \underline{h}_n(2) \end{pmatrix} = 0 \qquad (2.5.23)$$

i.e.

$$\underline{P}_n(1) - \mathbf{R}_n(2)\underline{h}_n(1) = 0 \qquad (2.5.24)$$

$$\underline{P}_n(2) - \mathbf{R}_n(4)\underline{h}_n(2) = 0 \tag{2.5.25}$$

which implies that $\underline{h}_n(1)$ and $\underline{h}_n(2)$ are now decoupled. The optimal solutions are then given by

$$\underline{h}_{opt}(1) = \mathbf{R}_n(2)^{-1}\underline{P}_n(1), \quad \underline{h}_{opt}(2) = \mathbf{R}_n(4)^{-1}\underline{P}_n(2) \tag{2.5.26}$$

Due to the decoupling, the $\mathbf{R}_n$ matrix becomes diagonal, and the third order terms of $\underline{X}_n$ can be neglected. Consequently, some calculations in formulas (T2.3.1) (T2.3.2) (T2.3.5) (T2.3.6) in Table 2.3 can be simplified respectively as follows

$$\begin{pmatrix} \underline{\tilde{K}}_n(1) \\ \underline{\tilde{K}}_n(2) \end{pmatrix} = \frac{\begin{pmatrix} \mathbf{R}_n^{-1}(2) & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_n^{-1}(4) \end{pmatrix} \begin{pmatrix} \underline{X}_{n-L} \\ (\underline{X}_{n-L} \otimes \underline{X}_{n-L}) \end{pmatrix}}{1 - \underline{X}_{n-L}^T \mathbf{R}_n^{-1}(2)\underline{X}_{n-L} - (\underline{X}_{n-L} \otimes \underline{X}_{n-L})^T \mathbf{R}_n^{-1}(4)(\underline{X}_{n-L} \otimes \underline{X}_{n-L})} \tag{2.5.27}$$

$$\begin{pmatrix} \tilde{\mathbf{R}}_n^{-1}(2) & \mathbf{O} \\ \mathbf{O} & \tilde{\mathbf{R}}_n^{-1}(4) \end{pmatrix} = \left(\mathbf{I}_2 + \begin{pmatrix} \underline{\tilde{K}}(1) \\ \underline{\tilde{K}}(2) \end{pmatrix} [\underline{X}_{n-L}^T (\underline{X}_{n-L} \otimes \underline{X}_{n-L})^T]\right) \begin{pmatrix} \mathbf{R}_n^{-1}(2) & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_n^{-1}(4) \end{pmatrix} \tag{2.5.28}$$

$$\begin{pmatrix} \underline{K}_{n+1}(1) \\ \underline{K}_{n+1}(2) \end{pmatrix} = \frac{\begin{pmatrix} \tilde{\mathbf{R}}_n^{-1}(2) & \mathbf{O} \\ \mathbf{O} & \tilde{\mathbf{R}}_n^{-1}(4) \end{pmatrix} \begin{pmatrix} \underline{X}_{n+1} \\ (\underline{X}_{n+1} \otimes \underline{X}_{n+1}) \end{pmatrix}}{1 + \underline{X}_{n+1}^T \tilde{\mathbf{R}}_n^{-1}(2)\underline{X}_{n+1} + (\underline{X}_{n+1} \otimes \underline{X}_{n+1})^T \tilde{\mathbf{R}}_n^{-1}(4)(\underline{X}_{n+1} \otimes \underline{X}_{n+1})} \tag{2.5.29}$$

$$\begin{pmatrix} \mathbf{R}_{n+1}^{-1}(2) & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{n+1}^{-1}(4) \end{pmatrix} = \left(\mathbf{I}_2 - \begin{pmatrix} \underline{K}(1) \\ \underline{K}(2) \end{pmatrix} [\underline{X}_{n+1}^T (\underline{X}_{n+1} \otimes \underline{X}_{n+1})^T]\right) \begin{pmatrix} \tilde{\mathbf{R}}_n^{-1}(2) & \\ & \tilde{\mathbf{R}}_n^{-1}(4) \end{pmatrix} \tag{2.5.30}$$

2.5.2.2. Window-length adaptation

In order to improve the tracking capability, the algorithm must be able to adapt its WL during iterations.

In this section, we will discuss algorithms for:

- Recursive window length decrement
- Recursive window length increment
  - a forward increment

- a backward increment

■ The decision on when to change the window length

(a) Recursive window-length decrement

When the time-varying speed of signal statistics speeds-up, the corresponding WL should be decreased such that less old data will become used.

In order to decide if the WL should be decreased, a detector will be applied which calculates the short-time average residual of the filter and decides if this WL decrement procedure will be called.

Description of the window-length decrement procedure

If WL decrement procedure is called at time-instant (n+1), it decreases the WL recursively by moving the window forward by one (from time instant n to n+1) and eliminating the oldest m (L>m≥1) data samples from the previous window. Noticing the following identities by their definition

$$\tilde{\mathbf{R}}_n = \mathbf{R}_{n,L-1}, \quad \tilde{\mathbf{P}}_n = \mathbf{P}_{n,L-1}, \quad \tilde{\mathbf{h}}_n = \mathbf{h}_{n,L-1}, \quad \tilde{\mathbf{K}}_n = \mathbf{K}_{n,L-1} \tag{2.5.31}$$

$$\mathbf{R}_{n+1} = \mathbf{R}_{n+1,L}, \quad \mathbf{P}_{n+1} = \mathbf{P}_{n+1,L}, \quad \mathbf{h}_{n+1} = \mathbf{h}_{n+1,L}, \quad \mathbf{K}_{n+1} = \mathbf{K}_{n+1,L} \tag{2.5.32}$$

and revising (2.5.5) and (2.5.6) to

$$\mathbf{R}_{n+1,L-m} = \mathbf{R}_{n,L} + \mathbf{Z}_{n+1}\,\mathbf{Z}_{n+1}^T - \sum_{i=0}^{m} \mathbf{Z}_{n-L+i}\,\mathbf{Z}_{n-L+i}^T \tag{2.5.33}$$

$$\mathbf{P}_{n+1,L-m} = \mathbf{P}_{n,L} + y_{n+1}\,\mathbf{Z}_{n+1} - \sum_{i=0}^{m} y_{n-L+i}\,\mathbf{Z}_{n-L+i} \tag{2.5.34}$$

the WL decrement procedure can be derived directly from the formulas (2.5.13-2.5.17). Table 2.4 summarizes this procedure.

Recursive step at time instant (n+1):
on entrance: WL=L+1
to decrease WL by m (m≤L)
1) Call algorithm 1 (constant WL:time n+1)
2) **For i:=1 to m do**

$$\tilde{\underline{K}}_{n+1} = \frac{\mathbf{R}_{n+1}^{-1}\,\underline{Z}_{n+1-L}}{1-\underline{Z}_{n+1-L}^{T}\,\mathbf{R}_{n+1}^{-1}\,\underline{Z}_{n+1-L}} \qquad \text{(T2.4.1)}$$

$$\tilde{\mathbf{R}}_{n+1}^{-1} = \mathbf{R}_{n+1}^{-1} + \tilde{\underline{K}}_{n+1}\,\underline{Z}_{n+1-L}^{T}\,\mathbf{R}_{n+1}^{-1} \qquad \text{(T2.4.2)}$$

$$\eta_{n+1}^{b} = y_{n+1-L} - \underline{Z}_{n+1-L}^{T}\,\underline{h}_{n+1} \qquad \text{(T2.4.3)}$$

$$\tilde{\underline{h}}_{n+1} = \underline{h}_{n+1} - \tilde{\underline{K}}_{n+1}\,\eta_{n+1}^{b} \qquad \text{(T2.4.4)}$$

$\underline{h}_{n+1} \Leftarrow \tilde{\underline{h}}_{n+1}$, $\mathbf{R}_{n+1}^{-1} \Leftarrow \tilde{\mathbf{R}}_{n+1}^{-1}$
L ⇐ L-1
end;{for}
on return: WL ⇐ WL-m
output: $\underline{h}_{n+1}$

Table 2.4 Recursive WL Decrement Procedure

(b) Recursive window-length increment

As the time-varying speed of the signal statistics slows down, taking more data for estimation can reduce the error-variance.

A detector will be used to check whether this window-length increment procedure will be called.

Description of the window-length increment procedure

If the WL increment procedure is called at time instant (n+1), it will increase the WL by m recursively as follows.

Two different approaches, forward and backward WL increment, are included if m>1 is selected. In principle, the WL will be increased by one at time instant (n+1), because the algorithm remembers only the data within the present WL. This is equivalent to one iteration step of the prewindowed algorithm with λ=1.0.

However, one may choose the WL increment m>1 at each time instant (n+1) by combining it with the backward WL increment, at the price that more than the

current WL of data should be stored in memory. Thus, the WL increment procedure is performed by adding one present new data sample and (m-1) successive old data samples preceding the previous window. Using the similar method as in the WL decrement procedure, the WL increment procedure can be easily derived from the formulas (2.5.13-17), by noticing the identity relations in (2.5.31) and (2.5.32) and revising (2.5.5) and (2.5.6) to

$$\mathbf{R}_{n+1,L+m} = \mathbf{R}_{n,L} + \underline{Z}_{n+1}\,\underline{Z}^{T}_{n+1} + \sum_{i=1}^{m-1} \underline{Z}_{n-L-i}\,\underline{Z}^{T}_{n-L-i} \tag{2.5.35}$$

$$\underline{P}_{n+1,L+m} = \underline{P}_{n,L} + y_{n+1}\,\underline{Z}_{n+1} + \sum_{i=1}^{m-1} y_{n-L-i}\,\underline{Z}_{n-L-i} \tag{2.5.36}$$

Table 2.5 summarizes such a combined procedure with arbitrary ($L \geq m \geq 1$) WL increment.

In Fig.2.2, a block diagram for the full RLS nonlinear adaptive filtering algorithm with an adaptive sliding-window is given.

recursive step at time instant (n+1):
on entrance: WL=L+1
to increase WL by m (m≥1)
(1) Forward WL increment:
(increase WL by 1)

$$k_{n+1} = \frac{\mathbf{R}_n^{-1} \underline{Z}_{n+1}}{1 + \underline{Z}_{n+1}^T \mathbf{R}_n^{-1} \underline{Z}_{n+1}} \qquad \text{(T2.5.1)}$$

$$\mathbf{R}_{n+1}^{-1} = \mathbf{R}_n^{-1} - \underline{K}_{n+1} \underline{Z}_{n+1}^T \mathbf{R}_n^{-1} \qquad \text{(T2.5.2)}$$

$$e_{n+1}^f = y_{n+1} - \underline{Z}_{n+1}^T \underline{h}_n \qquad \text{(T2.5.3)}$$

$$\underline{h}_{n+1} = \underline{h}_n + \underline{K}_{n+1} e_{n+1}^f \qquad \text{(T2.5.4)}$$

$L \Leftarrow L+1$

if (m-1)≥1 then do (2):
(2) Backward WL increment:
for i:=1 to (m-1) do

$$\bar{\underline{K}}_{n+1} = \frac{\mathbf{R}_{n+1}^{-1} \underline{Z}_{n-L}}{1 + \underline{Z}_{n-L}^T \mathbf{R}_{n+1}^{-1} \underline{Z}_{n-L}} \qquad \text{(T2.5.5)}$$

$$\bar{\mathbf{R}}_{n+1}^{-1} = \mathbf{R}_{n+1}^{-1} - \bar{\underline{K}}_{n+1} \underline{Z}_{n-L}^T \mathbf{R}_{n+1}^{-1} \qquad \text{(T2.5.6)}$$

$$e_{n+1}^b = y_{n-L} - \underline{Z}_{n-L}^T \underline{h}_{n+1} \qquad \text{(T2.5.7)}$$

$$\bar{\underline{h}}_{n+1} = \underline{h}_{n+1} + \bar{\underline{K}}_{n+1} e_{n+1}^b \qquad \text{(T2.5.8)}$$

$L \Leftarrow L+1$

$\mathbf{R}_{n+1}^{-1} \Leftarrow \bar{\mathbf{R}}_{n+1}^{-1}$, $\underline{h}_{n+1} \Leftarrow \bar{\underline{h}}_{n+1}$

end;{for}
on return: $WL \Leftarrow WL+m$
output: $\underline{h}_{n+1}$

Table 2.5 Recursive WL Increment Procedure

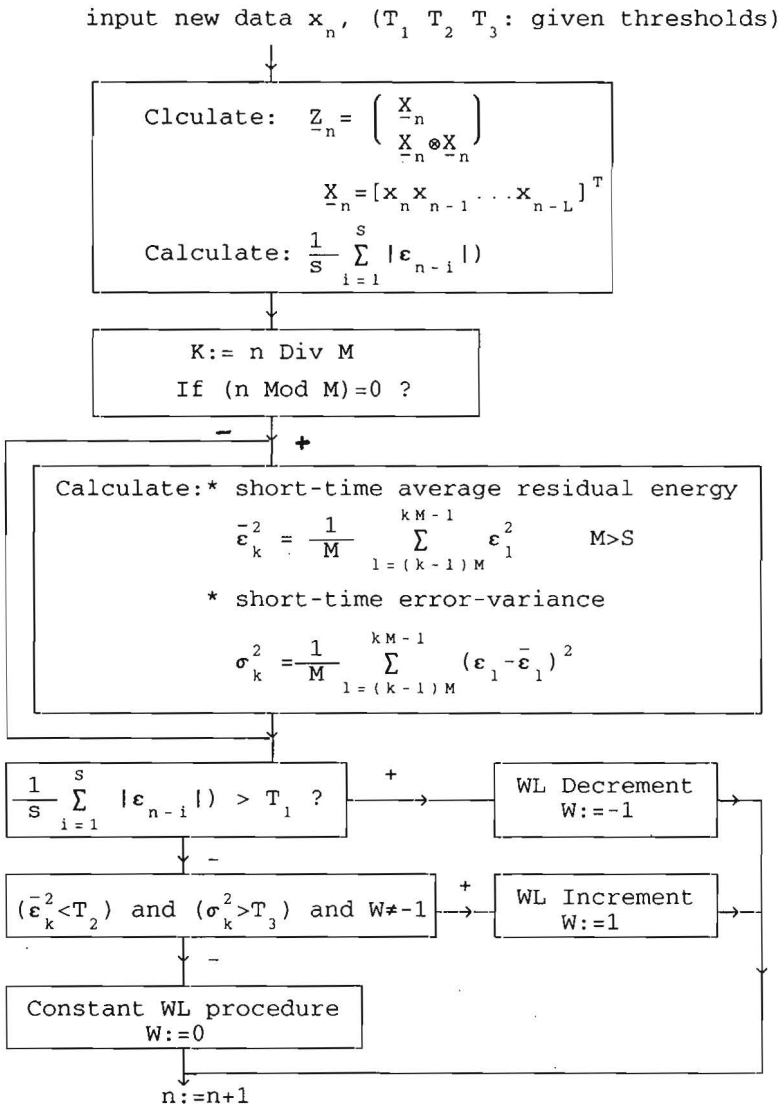


Fig. 2.2 Block diagram of a RLS nonlinear adaptive filter with an adaptive-sliding-window

2.5.3. Simulations and results

Simulations have demonstrated the tracking capability of the algorithm. In the simulations, the available observation signals are y_k and x_k, where $y_k=s_k+n_k$. The signal s_k is uncorrelated with the noise n_k and x_k, while n_k and x_k are mutually correlated. As an application of NL ADF, a NL Adaptive Noise Canceler (ANC) with an adaptive WL is then used to estimate $\hat{s}_k$.

The desired signal s_k is produced by passing zero-mean unit-variance Gaussian random noise through an AR(1) filter with pole at -0.5. The noise n_k is produced by passing a zero-mean unit-variance exponential random process x_k through a linear-quadratic filter. The output of the linear part is obtained by passing x_k through a MA(2) system with the time-varying coefficients $(h_1^{(1)},h_2^{(1)})$, while the output of the quadratic filter is obtained by squaring the linear filter output, as depicted in Fig.2.3.

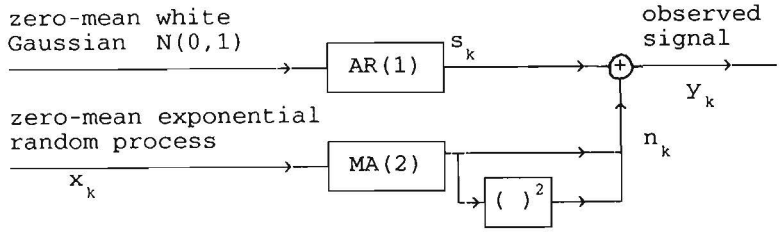


Fig. 2.3 The observed signal model used in simulations

The parameters of the time-varying NL filter, as listed in Table 2.6, are changed by a step function at t=3000. The SNR = $E(s_t^2)/ E(n_t^2)$ is set to -20dB. The NL ANC structure is the same as has been shown in Fig.2.1, with y_t and x_t as the primary and the reference input respectively.

The estimated filter parameters are compared with those obtained from the corresponding prewindowed filter (with λ=0.9975). Fig.2.4 shows the simulation results of this example.

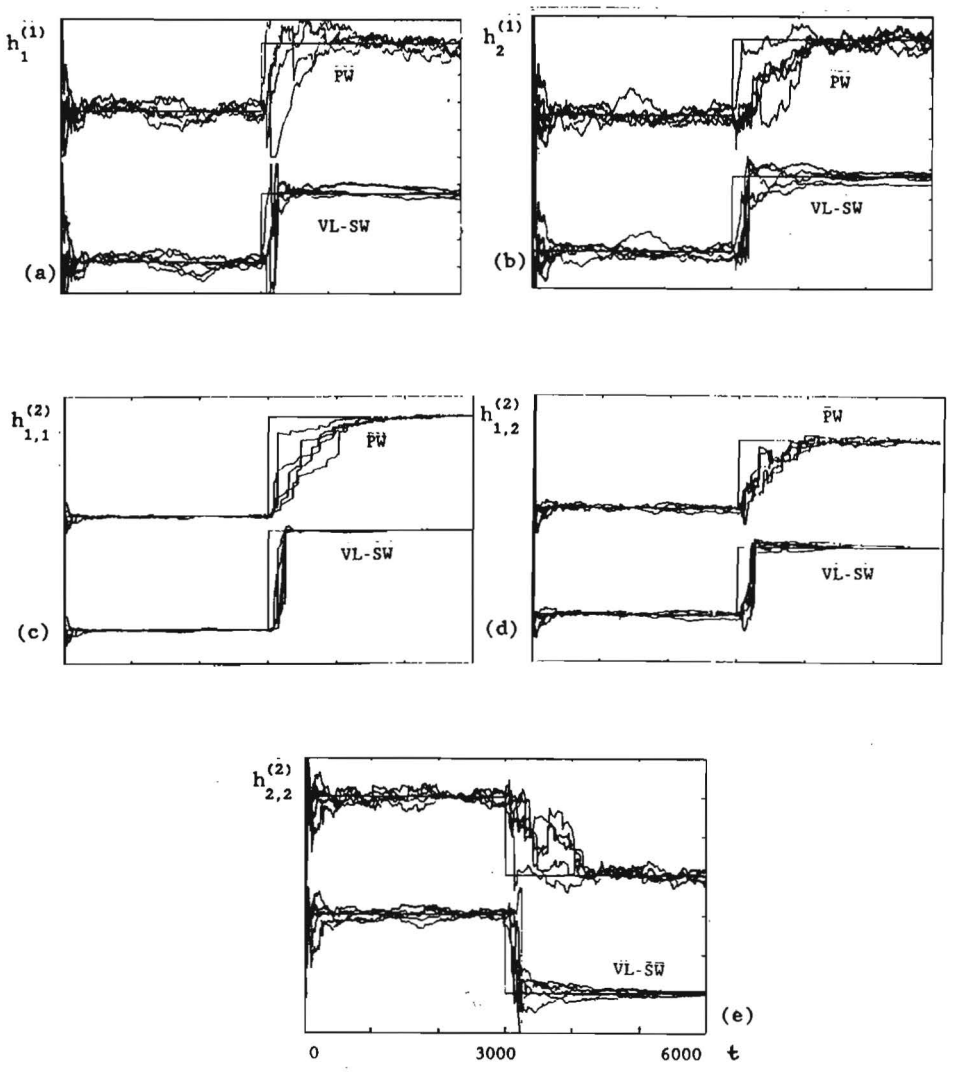


Fig. 2.4 Estimation of the parameters of a nonlinear time-varying system (5 runs, SNR=-20dB)

* Upper part: prewindowed (PW) method λ=0.9975
* Lower part: variable-window-length sliding window (VL-SW) method
* Straight line: ideal values (as listed in Table 2.6)

	$h_1^{(1)}$	$h_2^{(1)}$	$h_{1,1}^{(2)}$	$h_{1,2}^{(2)}$	$h_{2,2}^{(2)}$
$0 \le t \le 3000$	1.23	-0.45	1.5129	-0.5535	0.2025
$3000 < t \le 6000$	1.73	0.05	2.9929	0.0865	0.0025

Table 2.6 Time-varying nonlinear filter parameters

2.5.4. Concluding remarks

A new RLS NL ADF algorithm with an adaptive-sliding-window has been developed. It uses a finite but adaptive size of the data window. The adaptation of the WL is performed recursively in the algorithm.

Due to the adaptive window length, this algorithm can provide versatile functions and faster convergence speed for tracking nonstationary signals and NL systems. Especially, if the statistics of nonstationary signals or the parameters of NL systems are time-varying with non-constant changing speed, adjusting this window size during recursion when needed, is a very effective method. However, this improved performance is obtained at the price of more calculations.

2.6. APPLICATIONS

2.6.1. Nonlinear adaptive system identification

In many situations, we want to estimate an unknown system from the measurement of its input and output signals. This problem is associated with the system identification. When the linear/nonlinear system is time-varying, the task is associated with *adaptive* linear/nonlinear system identification from nonstationary measurements.

Adaptive system identification has wide applications. It is often combined with the controlling of an industrial process. Fig.2.5 shows a schematic block

diagram of a typical identification-control process using such a linear/nonlinear adaptive filter. Once the unknown system is estimated from the measurements, this system can then be used to control some (un)desirable behavior. For example, it can be used for the short-circuit protection of a high-voltage line. The system is time-varying and is changing slowly due to many factors, such as the number of users, the temperature, the time period (busy or idle), etc.

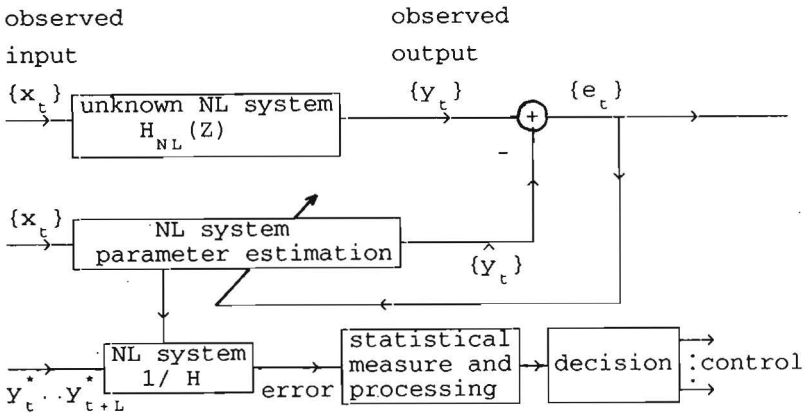


Fig.2.5 A schematic block diagram of
adaptive system identification and control

Previous work on adaptive system identification is mainly related to linear system model, represented commonly by a rational function such as an AR (AutoRegressive), MA (Moving Average) and ARMA process. Estimation of model parameters is the main task of modern linear system identification.
Unfortunately, many practical systems are NL, thus a linear system approximation often can not be used. The previously given NL adaptive Volterra filter with an adaptive-window can, in those cases, efficiently be applied to adaptive NL system identification.

For NL adaptive system identification, the estimation of model parameters

given measured input $\{x_n\}$ and output data $\{y_n\}$ is considered. The NL Volterra system which represents the input-output relations can then be estimated by

$$\hat{y}_n = h_0 + \sum_{m_1=1}^{N} h_{m_1}^{(1)}(n)x_{n-m_1+1} + \sum_{m_1,m_2=1}^{N} h_{m_1,m_2}^{(2)}(n)x_{n-m_1+1}x_{n-m_2+1} \qquad (2.6.1)$$

Under the Least Square (LS) criterion, the adaptive system identification is equivalent to finding an LS solution of the time-varying coefficient vector $\underline{h}_n$ of this system, which minimizes the cost function J_n

$$J_n = \sum_{k=n-L}^{n} (y_k - \hat{y}_k)^2 \qquad (2.6.2)$$

Remarks: From the characteristics of a specific system, one may choose only a part of the NL terms. As in most situations, selecting all the NL terms is neither economical nor necessary.

2.6.2. Nonlinear adaptive noise cancellation

The NL ADF algorithm developed above can be used as NL ANC. A similar example as in chapter 2.3.4.6. has been used. Suppose that the observed signal, $y_k=s_k+n_k$, is the desired signal s_k corrupted by noise n_k. Another observation x_k is given which is correlated with n_k. The signal s_k is zero-mean and uncorrelated with the noise n_k and x_k. The noise n_k and x_k are mutually correlated, which can be modeled by an unknown NL system with x_k and n_k as input and output respectively.

In this situation, we need to estimate signal $\{\hat{s}_k\}$ from the noisy observation $\{y_k\}$, given the noise correlated observation $\{x_k\}$ as a reference signal. Similar to linear ANC, a NL ANC can be used, with y_k and x_k as its primary and reference input, respectively. The NL ANC is designed to minimize the following LS objective function

$$J1_n = \sum_k (y_k - \underline{h}_k^T \underline{Z}_k)^2 \qquad (2.6.3)$$

where $\underline{Z}_k=[\underline{X}_k \ \underline{X}_k \otimes \underline{X}_k \ \ldots \ \underline{X}_k \otimes \underline{X}_k \otimes \ldots \otimes \underline{X}_k \otimes \underline{X}_k]^T$ and $\underline{X}_k=[x_k \ x_{k-1} \ \ldots x_{k-N+1}]^T$. Notice that s_k is zero-mean and uncorrelated with n_k and x_k. Hence it is equivalent to minimize J_2 below

$$J2_n = \sum_k (n_k - \underline{h}_k^T \underline{Z}_k)^2 \qquad (2.6.4)$$

where $\hat{y}_k = \hat{n}_k = \underline{h}_k^T \underline{Z}_k$ is the filter output. The above LMS or RLS NL ADF algorithm can be used directly for NL adaptive noise cancellation, as shown in Fig.2.6.

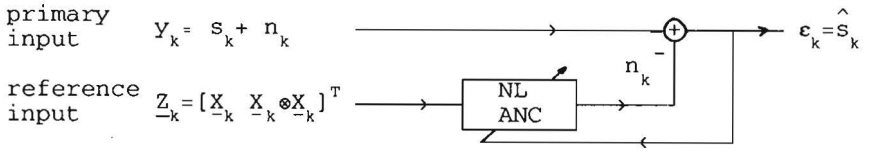


Fig.2.6 A block diagram of a nonlinear adaptive noise canceler

For the RLS algorithm above, an extra calculation should be added to calculate the forward filter residual ε^f_{n+1} (which is the estimated desired signal $\hat{s}_{n+1}$)

$$\varepsilon^f_{n+1} = y_{n+1} - \underline{Z}^T_{n+1} \underline{h}_{n+1} \qquad (2.6.5)$$

Remarks

(1) Relations between the adaptive system identification and the ANC

We can regard the above NL adaptive noise cancellation as the problem of identification of a NL system having x_k as input and n_k as output. Thus, the NL ANC is directly associated with an NL adaptive system identification.

2.6.3. Speech-like noise reduction

In chapter 4, we will show another application of the NL ADF algorithm to speech-noise reduction and speech intelligibility enhancement.

CHAPTER 3

ROBUST PITCH ESTIMATION

The speech fundamental frequency (pitch) can be accurately estimated by using pitch information existed within each speech frame and among the successive speech frames.

In order to do so, a new two-step algorithm will be built. First, a pseudo-perceptual pitch estimation algorithm has been developed as a coarse **pitch candidate** *estimator from each frame. It simplifies the perceptual method from the signal processing point of view, while maintaining robustness. Secondly, pitch contours are modeled by their stochastic characteristics using Hidden Markov Models. The parameters of HMMs can be trained by using the data from noise-free speech signals. A detailed-algorithm uses these models for* **pitch contour** *estimation from candidates under a Maximum Likelihood (ML) criterion. Some simulation results are included.*

3.1 Introduction

Speech fundamental frequency (or pitch), which is defined as the reciprocal of the interval between two vocal-cord impulses, is one of the most important features of voiced speech signals. Pitch estimation is one of the most important tasks in speech signal processing. In most cases, one is concerned with a pitch contour (we call the dynamic pitch as a function of time the "pitch contour") rather than a single pitch period. An accurate representation of voiced-information is often of paramount importance in many application aspects such as speech synthesis, coding, compression, enhancement, speaker identification, etc.

Pitch estimation from stationary frames of clean speech signals is a relatively easy task. Many algorithms[10,11,55,61,81] have been successfully developed to handle such a situation.

However, accurate pitch estimation remains a difficult problem in more

complicated situations. The difficulty arises when the speech signals are contaminated by various kinds of background noise such as white noise, interference speech and background music, etc, and during transition frames of vowel-vowel/vowel-consonant in clean speech signals. In previous studies, several robust pitch estimation algorithms have been proposed[39,89,101] for solving this problem.

In order to find a robust solution, we will first review some previous studies and search some common weak points which might be helpful for our consideration. Then a new pitch estimation algorithm will be formed, following the line of being consistent with the human auditory global processing, using as much as possible the source information, and using a-priori general knowledge of pitch contours.

This section will be organized as follows. First, we will describe the problem addressed in this research, and review some of the pitch estimation algorithms directed to robust pitch estimation. After analyzing the weak points of the existing algorithms and overviewing human pitch perception outlines, a basic skeleton of a new pitch estimation algorithm, a two-step algorithm consisting of coarse and detailed estimation, is then formed. Two parts of the algorithm will be described in detail separately, together with some brief introduction into the basic theory behind. The simulations are described in detail and some preliminary results are included. Finally discussion and conclusions will be given.

3.2 Pitch contour estimation from noisy speech signals

A large part of the speech signals is voiced, which is caused by the periodic excitations of the human vocal-cord. The frequencies of these excitations change slowly and continuously in speech sentences, and typically fall in the range between 50 and 400 Hz. Consequently, quasi-periodicity is an important characteristic for voiced-speech signal analysis.

In many practical applications, reliable and accurate estimation of the pitch periods from the continuous frames of a speech sentence (thus estimation of a pitch contour) is needed. Often, the speech signal is contaminated by

noise or is in some transition state. Hence a robust estimation algorithm is needed. The pitch estimation problem addressed here is to find such a robust algorithm. More specifically, it can be used to estimate one speaker's pitch contour from (white and colored) noise contaminated speech, or to simultaneously estimate multi-speakers' pitch contours from speech corrupted by another background interference speech.

As a direct link to our research in speech separation, we need an algorithm which can simultaneously and accurately estimate the pitch contours of both the target and the interference speech, from co-channel speech signals, with a wide range of Target-Interference Energy Ratio.

3.3. Review of the previous studies on robust pitch estimation

Robust pitch estimation is still an active field due to the increasing demands in speech processing.

Classical pitch estimation algorithms such as short-time AMDF (Average Magnitude Difference Function), short-time autocorrelation of linear prediction residuals with center clipping, etc[39,81], can only handle stationary and clean speech. Recently, a lot of algorithms with increased complexity have been developed to cope with these difficult situations. Some of these algorithms will be reviewed briefly. They are roughly categorized into two parts: the signal processing-based approaches and the pitch perception-based approaches. The reasons to review these algorithms are twofold. One is to find some common weak points in order to build an improved algorithm. The second is that some of these algorithms can be selected after proper modification to provide weighted pitch candidates, which can also be used in the first part, the coarse estimate part, of our new algorithm.

(1) Signal processing-based approaches

*** Histogram formed by spectral peak submultiples**

Parson[71] has proposed a simultaneous two-speakers' pitch estimation algorithm at TIR around 0dB. The algorithm is based on the principle that the spectrum of voiced-speech has peaks concentrated at the pitch harmonic

frequencies. In this algorithm, all *narrowband* filtered spectral peaks are assembled. A histogram collects all possible integer submultiples of these peaks. The maximum peak value in the histogram is selected to be the pitch of the first speaker. Then a second histogram is formed by using only those peaks which are not in the harmonic frequencies of the first pitch period. An improved approach can be obtained by using a sinusoidal model to represent pitch harmonics[60].

*** Cepstral-based pitch estimation (by linear and nonliner vocal-tract models)**

According to the simplified speech model, a speech signal is produced by the convolution of vocal-cord excitations (source) with the vocal-tract function (system). The cepstral-based method performs homomorphic deconvolution of these two elements. Because vocal-cord excitations change relatively faster than the vocal-tract function, it is then separable in the cepstral domain by using a lowpass and a highpass filter respectively. The output of the highpass filter can be used for pitch estimation. A multiplicative cepstral domain analysis for pitch estimation using a nonlinear vocal-tract model[41] is reported to give improved performance in pitch extraction from noisy speech.

*** Pitch predictor**

A first-order predictor, which is a function of both the predictor coefficient b and the pitch period delay M, can be used[48,82]. Using a Minimum Squared Error criterion $E(M,b)=\sum_{n=0}^{N-1}[x(n)-bx(n-M)]^2$, the iterations can be done by an Estimation-Maximization approach ($\forall M_i \in$ [possible pitch period realm], calculate the corresponding optimal b_i and $E(M_i,b_i)$. The pitch period is chosen by the argument maximum $M=\underset{M_i}{\operatorname{argmax}}\, E(M_i,b_i)$).

*** Super resolution pitch determination**

By defining two successive pitch periods of speech signals as the amplitude modulated version of each other using the similarity model $x_{T_0}(t,t_0) = a(t_0)y_{t_0}(t,t_0)+e(t,t_0)$, pitch period T_0 is selected from the argument t_0 which is associated with the minimum normalized squared error J [62]

$$T_0 = \underset{t_0}{\text{argmin}} \left\{ J = \frac{\int_{t_0}^{t_0+\tau} [x_\tau(t,t_0) - a(t_0) y_\tau(t,t_0)]^2 dt}{\int_{t_0}^{t_0+\tau} \left| x_\tau(t,t_0) \right|^2 dt} \right\} \qquad (3.3.1)$$

*** Maximum Likelihood (ML) pitch estimation**

Given noisy speech signal $r_k = s_k + n_k$, ML estimation of the pitch period [105] $\hat{P}$ is equivalent to the selection of a P such that the energy of the estimated signal $E_{\hat{S}}(P)$ is maximized. Or equivalently, it is equal to minimize the noise variance $\sigma_n^2(P)$. The assumptions that speech signals are periodic with the relation $s_k = q_{k \bmod P}$ and the noise n_k is white and Gaussian are needed.

(2) Pitch perception-based approaches

*** Perceptual pitch estimator**

The perceptual pitch estimator[89] combines a cochlear model with a bank of autocorrelators. In this algorithm, a group of bandpass filters is used to emulate the cochlear filters, each filter centered at its characteristic frequency (uniform in a Bark scale), with a specific *critical bandwidth*, and with the frequency response similar to the tuning curve of auditory nerve fibers. The filter output is then halfwave rectified, and passed through a multi-channel coupled-AGC to compress the dynamic range. The output of each cochlear filter is then subjected to short-time windowed autocorrelations, which can be expressed by a two-dimensional *correlogram* **at each time instant**. A pitch detector synthesizes the coincidence appearance of those correlation peaks across all channels.

A similar algorithm is proposed by Weintraub[101]. In the algorithm the outputs of each cochlear filter are explained as neural firing events, and the corresponding event function is then defined. The short-time correlations are then calculated over these event functions.

These algorithms are reported to have high performance under various kinds of noise and the ability to handle multi-pitch information.

*** Pitch estimation by interspike interval histograms**

Goldstein[28] has developed a method where the pitch frequency is chosen

from the harmonic values estimated from the Interspike Interval Histogram (IIH) in the Maximum Likelihood (ML) sense. In this algorithm, the output of each cochlear filter is regarded as neural firing. In each channel, the intervals between two successive firings are collected by a histogram, which can be depicted by a two-dimensional figure at each time, called *neurogram*. The peaks in the histograms are then synthesized over all channels for the pitch estimation.

In Allen's paper[1] synthesis of IIH is performed by taking the pointwise product of each channel with its neighbors, and summing up over all channels. The results are then used for extracting pure tones (a tone is a sinusoid with a single frequency) embedded in noise.

3.4. Overview of the pitch perception in human auditory models

Human auditory processing is still far more intelligent than any other artificial algorithm. Therefore, the more knowledge of human pitch perception we use, the more benefit we can obtain in developing a better algorithm.

Psycho-acoustic experiments indicate that the Basilar Membrane (BM) in the auditory system performs some sort of running short-time spectral analysis on the acoustic waveforms, by decomposing a signal into isolated frequency components, with further processing done essentially along the time axis.

There are several different models of auditory pitch perception. They all have the same processing in the first step, as shown in the following schematic figure in Fig. 3.1.

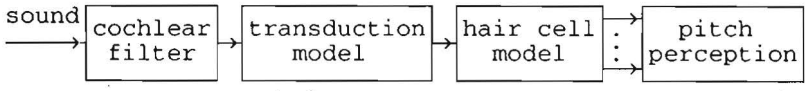


Fig.3.1 Auditory processing for pitch perception

Due to monotonous frequency band tuning along the length of the BM in the auditory system, the acoustic signal is split along the place dimension. To mimic this, the incoming speech signals are first processed through a group of specially designed bandpass filters, called "cochlear filters" (see section 1.2). The place abscissa in the BM can be regarded as frequency abscissa, scaled in the critical bands. The cochlear filter outputs are then fed to a transduction model emulated by multi-stage coupled Automatic Gain Control (AGC), followed by a hair cell model performed by a halfwave rectifier.

Afterwards there will be different processing depends on the pitch perception models. There are mainly two accepted theories.

In *Licklider's theory* of pitch perception[51], the output of each cochlear filter is passed through a neural autocorrelation mechanism which performs some kind of envelope autocorrelation of neural firings along the time direction. The analysis for pitch perception is performed in the temporal-place domain. In the latter section, the computational algorithms developed by Lyon and Weintraub are based on this model.

Another theory of pitch perception is based on *Goldstein's model*[28], which supposes that pitch perception is based on the ensemble of zero-crossing intervals of auditory neural firings. According to the theory, an array of fibers are firing synchronously with the stimulus: At moderate stimulus level, the neural firing-rate depends on the stimulus frequency. As the stimulus intensity increases, more fibers nearby will fire synchronously with this stimulus frequency. At high stimulus intensity, fibers are saturated so that the firing-rate will no longer increase. Thus, the *average firing-rate* is a function of the BM place, and the *firing-pattern* is a function of the stimulus intensity[25]. The computational algorithms of Goldstein[28] and Allen[1] in the above section use this model.

3.5. Skeleton of a robust pitch estimation algorithm

From the previous review, we see that most algorithms are only suitable for speech corrupted with white/colored noise. In speech separation there are two different situations, which lead to different demands for a pitch estimation

algorithm.

In the first situation, the sound of one speaker is always dominant. What we need to estimate is the pitch of the stronger speaker[37,66]. In such a case, a single speaker's pitch estimation algorithm from speech contaminated by colored noise can be selected. Another situation is somewhat more complicated. If speech of two speakers has nearly equal intensity (around 0 dB TIR), a multi-pitch estimation algorithm should be selected.

Some common weak points of the previous techniques

There are several weak points in the previous discussed algorithms.

(1)In the signal processing-based approaches, there is little consideration on how to use the human pitch perception advantages. A total neglection of this knowledge is not very wise, because no pitch estimation technique developed up to now can reach the level of human pitch perceptual robustness.

(2)Although the perceptual-based algorithms are reported with high robustness against various kinds of noise, their high computational burden often inhibits their applications.

(3)Most algorithms try to make a decision based on the estimation from each **isolated frame**. Some try to improve the results afterwards by using a simple smoothing algorithm or a Dynamic Programming (DP) method. However, the pitch information contained in the isolated frames is neither sufficient nor complete. Consequently, such an estimation can not be very reliable and accurate.

(4)Most algorithms use only one processing approach, thus are limited to specific cases. According to observation from psychophysics, human processing is flexible, running from simple to complex, depending on the complexity of input signal.

Possible improvement from the previous techniques

(1)There exists a gap between the signal processing-based and the perceptual-based pitch estimation approaches. We believe that it is necessary to take into consideration how to combine the human pitch perception advantages with the signal processing techniques.

We believe that mimicking the human auditory behavior for the purpose of pitch estimation is neither necessary nor possible. This is due to the fact that we are still in a very early stage of understanding the processing in the human auditory system, especially that in the human central nerve. Besides, from the signal processing point of view, we do not mind what this black-box system would look like. Rather, how the output pitch can be "best" estimated from the given input signals is concerned. As long as an algorithm provides good pitch estimation, it is acceptable.

The disadvantages of perceptual-based approach are that it must mimic all the micromechanisms in the human auditory system, and that it depends fully on the correctness of a given auditory model. This leads to high computational cost.

Our algorithm tries to bridge such a gap and combines the advantages of simplicity in signal processing approaches with the robustness of perceptual pitch algorithms.

(2)We believe that it is very important to use information from both the inter and intra analysis frames. Post-processing such as a smoothing or DP algorithm can not recover the lost pitch information caused by an unproper early decision based on each frame. Therefore, one should use such an isolated frame-based estimation algorithm as a step of obtaining an initial guess of pitch candidates. A decision should be postponed until all possible pieces of information are used.

(3)We believe that the complexity of the processing must be adaptable to the degree of input speech contamination. This fact, which is also indicated by psycho-acoustic experiments, is easy to be understood. In the most

complicated situations, the human auditory system even uses a-priori knowledge accumulated from the previous learning. In order to estimate pitch from extremely noisy speech, we should use more complex processing including a better initial guess of pitch candidates, and some previous learned knowledge.

Skeleton of the algorithm:

Following the above line, a skeleton of our new robust pitch candidate estimation algorithm can be formed.

First, the estimation can be performed in two-steps. A coarse pitch candidate estimation algorithm is used as an initial guess to provide only pitch candidates with probability/weight values. In general, this algorithm should be selected in relation with the complexity of noisy input speech. Hence, a group of different algorithms can be selected. In the second step, given the candidates a detailed algorithm uses dynamic pitch properties among the frames, to estimate a "best" pitch contour under a pre-selected criterion.

In particular, special emphasis is put on the consistency with the auditory system global processing throughout this whole processing. Such a processing shares partly the same computations in the speech separation part as will see in the next chapter. Consequently, as a by-product, this saves calculation cost for the whole system.

3.6. A Pseudo Perceptual Pitch Candidate Estimation Algorithm

- A coarse estimation as an initial guess of candidates

In this section, a new coarse pitch candidate estimation algorithm will be developed.

Analysis shows that not only the signal envelopes but also the signal "carriers" can be used independently for pitch estimation. Methods for calculating pitch candidates from the "carriers" and the envelopes are then both given, based on signal analysis. Simplifications in filter design and

signal processing are obtained over the perceptual type of algorithms. The same groups of bandpass filters are used for analyzing envelopes and "carriers". It is consistent with the auditory global processing. Robustness can be obtained by combining the two pieces of information to provide reliable estimation. Finally, simulations are described and some results are included.

From the previous analysis, it is seen that the perception type of pitch estimation algorithms has good robustness but high computational cost. From the signal processing point of view, we are only interested in the estimated output and not in the detailed micromechanisms of the human auditory processing. In order to estimate the "best" output, we are interested in finding what pieces of information can be related to pitch, and under which circumstances it can be extracted.

In the following, we will first analyze what information associated with pitch is obtainable. An algorithm to estimate pitch candidates will then be given[34].

3.6.1. Analysis of bandpass signals

Pitch estimation via signal envelopes and signal carriers

In order to be consistent with the auditory global processing, the speech signal is analyzed in the time-frequency domain.
Using a sinusoidal model, the observed speech signal s(t) can be represented as the components at pitch harmonic frequencies

$$s(t) = \sum_{k=1}^{K(\omega_0)} a_k(t)\cos(k\omega_0 t + \Phi_k) \qquad (3.6.1)$$

where $f_0=\omega_0/2\pi$ is the fundamental (pitch) frequency, $K(\omega_0)$ is the number of harmonics within the speech bandwidth, $a_k(t)$ and $\Phi_k(t)$ represent the k^{th} harmonic amplitude and the phase offset relative to the origin of a speech frame.

In practical situation, the signal component (harmonic) obtained is

multiplied by the frequency response of the bandpass filters as follows.

$$S(t) = \sum_{k=1}^{K(\omega_0)} b_k(t)\cos(k\omega_0 t + \Phi_k) \qquad (3.6.1')$$

where $b_k(t)=W(k\omega_0 t-\omega_l)a_k(t)$, and $W(\omega-\omega_l)$ is the frequency response of the l^{th} bandpass filter with center frequency $f_l=\omega_l/2\pi$, and it is supposed that the k^{th} harmonic is within the frequency band of the l^{th} bandpass filter.

In the following, we will show that both the signal envelope and the signal "carrier" of bandpass filtered signals contain pitch information. In order to simplify the analysis, we will set the phase value Φ_k to zero.

The information contained in the output of the k^{th} wide-bandpass filtered signal can be analyzed as follows:

(1)Only one harmonic is contained in a filter band

This situation can appear in both the *narrowband* and *wideband* filtered signal.

Supposing the m^{th} harmonic is contained in the k^{th} bandpass filter. In this case, the filter output is expressed as below

$$s_k(t) = b_m(t)\cos(m\omega_0 t) \qquad (3.6.2)$$

where the signal "carrier" contains pitch information, while no pitch information is contained in the envelope $b_m(t)$.

(2)Two harmonics are contained in a filter band

Suppose the m^{th} and $(m+1)^{th}$ harmonics are obtained in the bandwidth of the k^{th} bandpass filter, as follows

$$s_k(t) = b_m(t)\cos(m\omega_0 t) + b_{m+1}(t)\cos((m+1)\omega_0 t) \qquad (3.6.3)$$

After simple triangular transforms, we obtain

$$s_k(t)=b_m(t)\left(1+\frac{b_{m+1}(t)}{b_m(t)}\cos(\omega_0 t)\right)\cos(m\omega_0 t)-b_{m+1}(t)\sin(\omega_0 t)\sin(m\omega_0 t) \qquad (3.6.4)$$

where supposing $|b_m(t)| \geq |b_{m+1}(t)|$ does not present any loss of generality. When $m\omega_0$»(filter bandwidth), the first term in (3.6.4) is the amplitude modulated signal, having envelope frequency $f_0=\omega_0/2\pi$ and "carrier" frequency $f_{c1}=m\omega_0/2\pi$. The second term is the double-sideband (DSB) modulated signal. From the property of DSB, the envelope shows a periodicity at time instant $n/(2f_0)$, n=1,2... Because of the sudden 180^0 phase change in the "carrier" signal when the message signal undergoes zero values, the carrier signal is no longer periodic. By counting the short-time average local maximum numbers of the "carrier" signal for calculating "carrier" frequency, the average pseudo-carrier frequency will be $f_{c2}=(m+1)\omega_0/2\pi$.

From the above, we can conclude that:

(1)The envelope of $s_k(t)$ presents a correlation peak at time index $1/f_0$, if the $|b_m(t)| \neq |b_{m+1}(t)|$;

(2)One of the "carrier" period multiples of $s_k(t)$ is at the time index

$$1/f_0=m/f_{c1}=(m+1)/f_{c2} \tag{3.6.5}$$

Hence, either the signal envelope or the "carrier" from the *wide bandpass* filter can be used for pitch estimation.

(3)More than two harmonics are contained in a filter band

Suppose that the total number of harmonics within the filter bandwidth is n. In this case, the filter output can be expressed

$$s_k(t) = \sum_{i=-L}^{L1} b_{m+i}(t)\cos((m+i)\omega_0 t) \tag{3.6.6}$$

where L=(n div 2), L1=L if n is odd, otherwise L1=L-1. Similarly, after some triangular transformations we obtain

$$s_k(t) = b_m(t)\left(1 + \sum_i \frac{b_{m+i}(t)}{b_m(t)}\cos(i\omega_0 t)\right)\cos(m\omega_0 t) - \\ - \sum_i b_{m+i}(t)\sin(i\omega_0 t)\sin(m\omega_0 t) \tag{3.6.7}$$

which includes pitch information both in the signal envelope (if there exists $b_m(t) \neq b_n(t)$, $m \neq n$), and in the signal "carrier".

Remarks:

1) Pitch information is obtainable from the signal envelope of wideband filtered signals.
   Pitch information is extractable from the signal envelope, if there exists amplitude differences among the different $b_m(t)$. There is no special demand for the shape of the frequency response of the bandpass filters, because it is unlikely that all the frequency bands coincidently present no amplitude modulation. The amplitude modulated information can be enhanced afterwards if needed.

2) Pitch information is obtainable from the signal "carrier" of both the narrowband and the wideband filtered signals
   The pitch information is also extractable from the signal "carriers". This can be obtained from *either the narrowband or the wideband* filtered signals. This gives an analytical explaination to the perceptual pitch estimation algorithm of Goldstein[28] and Allen[1], and shows that the restriction that only a narrow-bandpass filtered signal carries such information[1], is not necessary.

3) If the amplitudes of all harmonics are in a same value within a band, it can be shown that the signal envelope contains no pitch information, while the signal "carrier" still contains pitch information.

4) In order to simplify the analysis, all phase values Φ_k, k=(m-L)...(m+L1), have been set to zero. In the case of more than one harmonics in a band, it represents a simple and a special case of zero phase-offset difference among these harmonics.

3.6.2. Algorithm descriptions

From the previous analysis we conclude that both the signal envelope and the signal "carrier" in each bin contain pitch information. Our algorithm is formed by extracting pitch information existed in these signals. Much attention will be paid to the consistency with the global approach of pitch perception in the auditory system by temporal-place domain time-directional

analysis. The analysis is performed along time-direction in the time-frequency band domain. Hence, this pitch estimation algorithm[34] can be regarded as a pseudo perceptual pitch estimator.

In the following, this algorithm will be described in detail, mainly concentrating on the following aspects:

* Splitting signals into frequency bins;
* Pitch estimation from the signal envelopes and from the signal "carriers";
* The enhancement of pitch candidates.

*** Splitting signals into frequency bins by a wide-bandpass filterbank**

In the pitch estimation algorithm, the speech signal is first filtered by a group of wide-bandpass filters with uniformly spaced center frequencies at $\omega_k=2\pi k/N$, k=1..N. The filter frequency response $H_k(\omega)$ is a symmetric function of ω_k. The output signal of the k^{th} bandpass filter is then in a real value and can be expressed as the following convolution

$$S_{\omega_k}(n) = s(n)^* h_k(n) \qquad (3.6.8)$$

where $h_k(n)=w(n)\cos(\omega_k n)$ is the impulse response of k^{th} bandpass filter, w(n) is a symmetric time-window of length L, (L is chosen comparable to the range of the pitch period), and ($N \geq L$).

In order to implement these filters we use the Short Time Fourier Transform (STFT), followed by shifting the output data into their bandpass version, and then taking the real part.
Because STFT performs FFT on the windowed data s(t)*w(t), its advantages of easy implementation, computational effectiveness in software, and fast speed by the dedicated Digital Signal Processor (DSP) hardware, made it an attractive tool for implementing the filterbank.

It should be mentioned that the same group of wide bandpass filters is shared by both signal envelope- and signal "carrier"-based analysis.

*** Estimate pitch from the signal envelopes**

1) Calculate envelope time-autocorrelations at each frequency bin.

First, the local maximum values and the associated time indices from the bandpass signals are picked up. A linear interpolation is performed between the successive local maxima. The d.c. element is then removed from the interpolated signals. The short-time envelope autocorrelations are calculated within the lags of the possible pitch period. Finally, the autocorrelation values are normalized by the signal energy at that bin.

2) Create a "correlogram"

The local autocorrelation peak values can be depicted in a figure of frequency-autocorrelation lag dimension using a so called "correlogram". The darkness (intensity) of each point reflects the relative correlation strength. The pitch period candidates are often associated with those lag indices which coincide most with the autocorrelation peaks over all bins, indicated by a dark line in a "correlogram". It should be mentioned that this "correlogram", (the name was originally used in perceptual pitch analysis[52]), represents only the signal envelope autocorrelations in this case.

(3)Synthesize the coincidence appearance information

A "correlogram" represents the envelope autocorrelations over a group of bins at only a specific time instant t_0. In order to select pitch candidates, or, to see the time evolution process, a "correlogram" is often synthesized into a one-dimensional expression. It can be created by accumulating at each lag the normalized autocorrelation values at those bins which present local peaks. The first several time indices associated with the prominent values are then selected as pitch candidates.

*** Estimate the pitch via the signal "carriers"**

1) Calculate short-time "carrier" period and its multiples

At each frequency bin, a short-time "carrier" period value and its multiples are calculated. The number of signal local maxima is counted over a short-time duration. The average interval time between two successive peaks is used as the "carrier" period T_c. The time indices of "carrier" period multiples nT_c, n=1,2... are also calculated.

2) Create "multiplogram"

A "multiplogram" depicts the appearance of the carrier period multiples of all frequency bins at a specific time instant t_0. Because of the coincidence appearance of the "carrier" period multiples at the pitch period over frequency bins, a vertical line is expected in the pitch period index along the time direction in this figure.

3) Synthesize the coincidence appearance information

A one-dimensional function is calculated from a multiplogram at each time instant t_0 by accumulating the (*weighted*) repetition number of the "carrier" period multiples over all bins.

Remarks:

Further improvement can be expected if the weighted repetition number is used, such that the total repetition number in these time-indices will count more heavily if the carrier period multiples appear successively in several neighboring frequency bins.

*** Enhance the pitch candidates**

The pitch candidates obtained from above are then enhanced, which results in a sharpened *candidate pitch contour figure.* This is done as follows: At each frame, the one dimensional accumulated normalized autocorrelation peak values from the envelopes are center clipped. Also, the one-dimensional accumulated repetition number of the signal "carrier" period multiples are center clipped and peak enhanced.

Remarks:

Some simplifications are obtained over the perceptual pitch estimator. For instance, the design of bandpass filters is simplified, which results in less calculations. Simple bandpass filters using Short Time Fourier Transform (STFT) can replace the special cochlear filters in the perceptual pitch estimator.

3.6.3. Simulations and results

In the following, the simulations with this pseudo-perceptual pitch candidate estimation algorithm will be described.

The simulations are performed on both *stationary* and *nonstationary* speech signals corrupted by white Gaussian noise and with a speech of a background interference speaker, respectively. We choose such cases because pitch estimation from noisy speech, and simultaneous multi-pitch estimation from co-channel signals often appear from many practical demands.

In the simulations, the contaminated speech signals first were subjected to a group of wide-bandpass filters, which are implemented by successively applying STFT on overlapped frames of hamming windowed-data, succeeded by frequency shifting from lowpass to bandpass, and then taking the real part of the signals. The Hamming window function is of length L as given below

$$W(n) = 0.54 - 0.46 \cos(2\pi n/(L-1)) \qquad 0 \le n \le (L-1) \qquad (3.6.9)$$

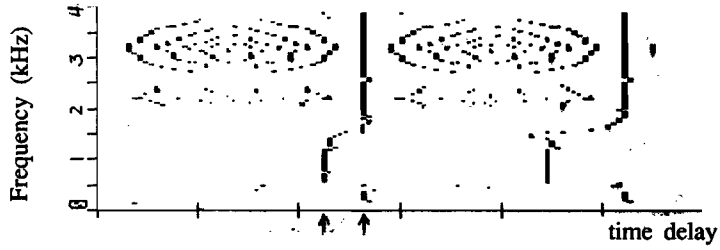
With the selected sample frequency f_s, the equivalent bandwidth of each bandpass filter is $2B = 4f_s / L$.

Pitch candidate estimation from stationary synthetic noisy-speech signals

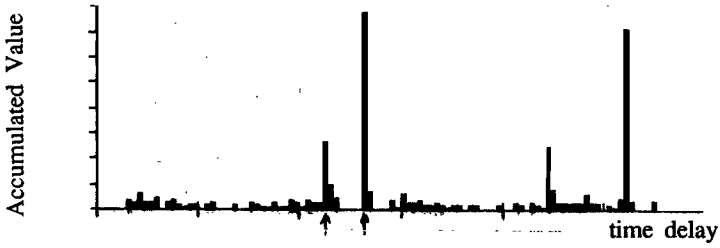
In the following examples, several simulation results on pitch estimation from **stationary speech signals** corrupted by an interference speech or by Gaussian white noise at different SNR are included. The speech signals are synthetic with three formants and given pitch periods.

Fig.3.2. shows two examples of pitch candidate estimation from speech signals corrupted by another interference speech in 0 dB SNR. Fig.3.3 shows the results of pitch candidate estimation from speech corrupted by white noise in SNR=0dB, Fig.3.4 includes two examples of pitch candidate estimation from speech signals corrupted by white noise in SNR=6 dB.

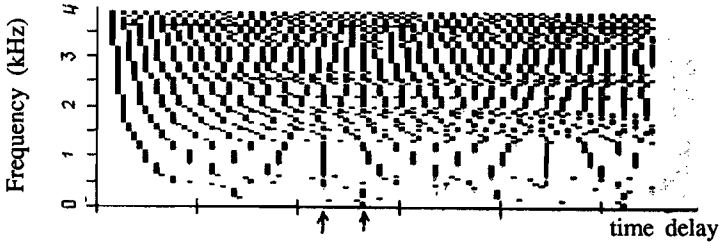
In these figures, the vertical lines along the frequency axis in a "correlogram" of (1) and a "multiplogram" of (2) indicate more coincident appearances of the possible pitch period values. In (3) and (4), the values in (1) and (2) are accumulated over frequency bins respectively to produce a one-dimensional view.



(a) Envelop correlations



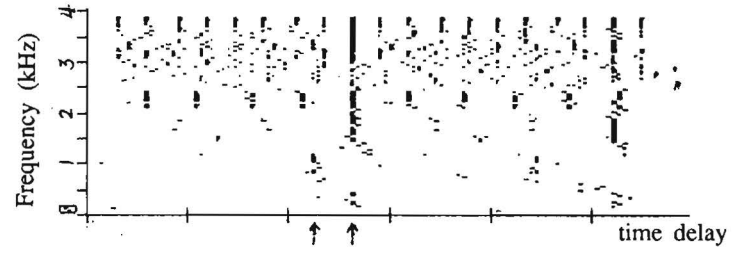
(b) Accumulated values of envelop peaks



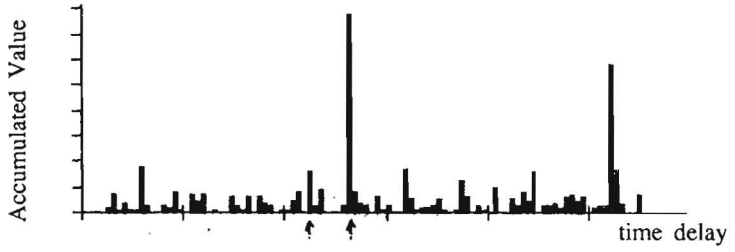
(c) Appearance of "carrier" period multiples

(d) Accumulated repetition number of the carrier perid multiples

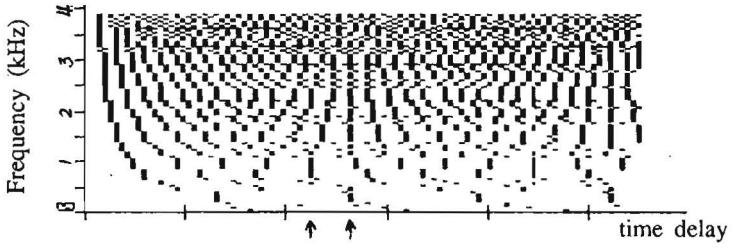
Fig. 3.2 (1) Estimation of pitch candidates from speech corrupted by another inteference speech (TIR=0 dB) (p_1=40, p_2=47(samples),f_S=8kHz)



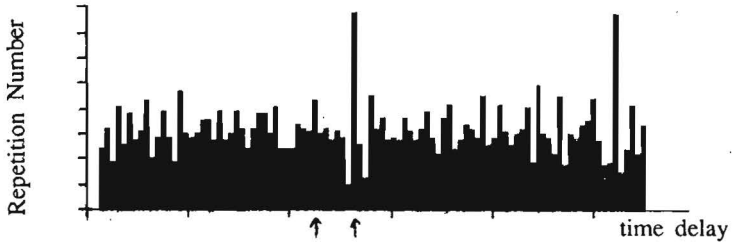
(a) Envelop correlations



(b) Accumulated values of envelop peaks

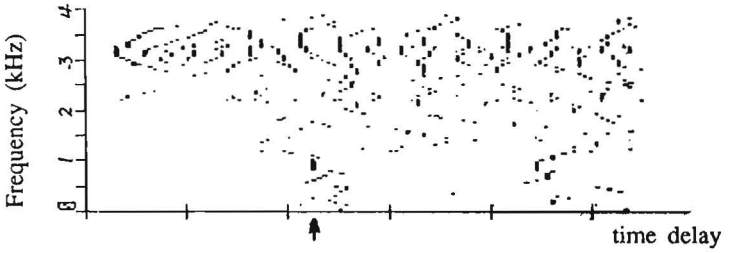


(c) Appearance of "carrier" period multiples

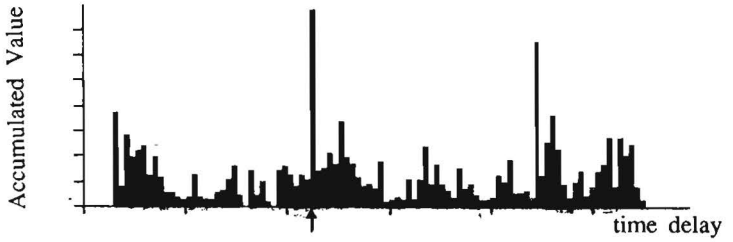


(d) Accumulated repetition number of the carrier perid multiples

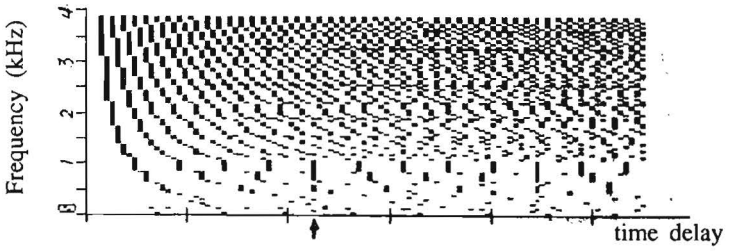
Fig. 3.2(2) Pitch candidate estimation from speech corrupted by another inteference speech (TIR=0 dB) (p_1=40, p_2=47(samples),f_s=8kHz)



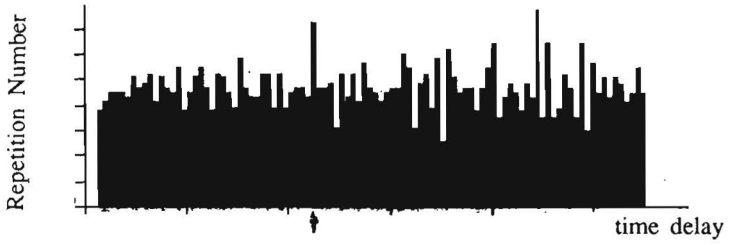
(a) Envelop correlations



(b) Accumulated values of envelop peaks

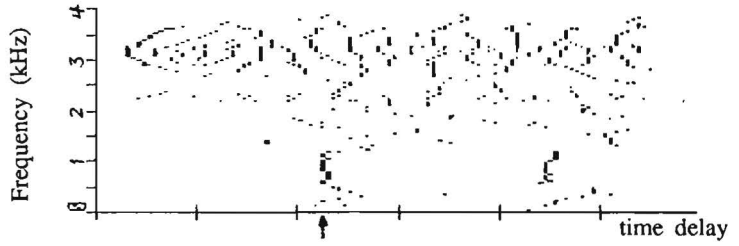


(c) Appearance of "carrier" period multiples

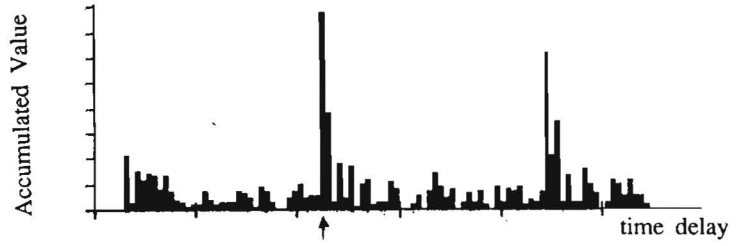


(d) Accumulated repetition number of the carrier perid multiples

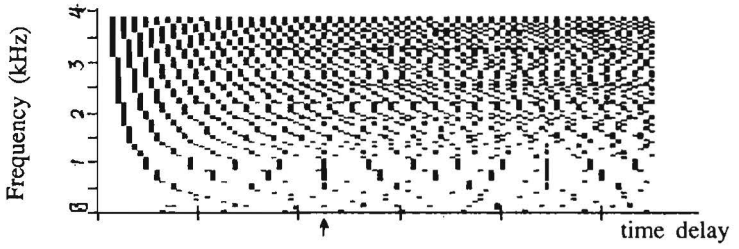
Fig. 3.3 Pitch candidate estimation from speech corrupted by white noise (SNR = 0 dB p=40(samples) f_s=8kHz)



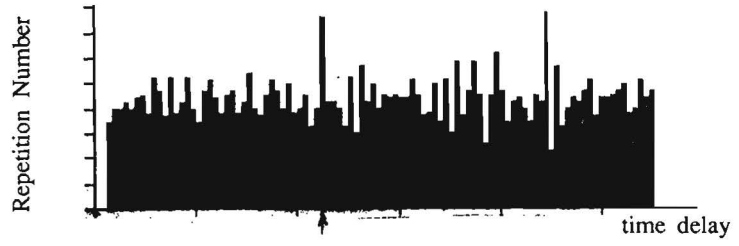
(a) Envelop correlations



(b) Accumulated values of envelop peaks

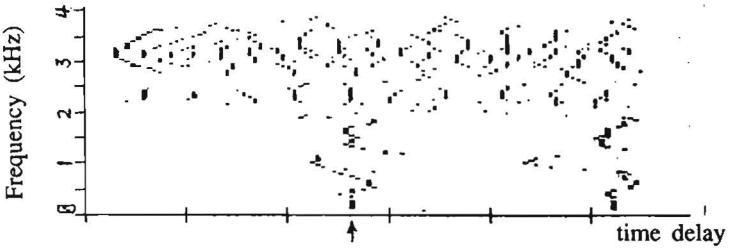


(c) Appearance of "carrier" period multiples

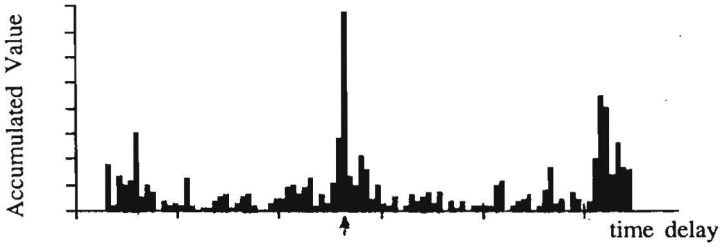


(d) Accumulated repetition number of the carrier perid multiples

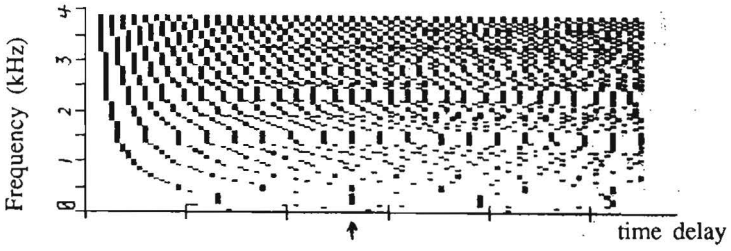
Fig.3.4(1) Pitch candidate estimation from speech corrupted by white noise (SNR = 6 dB p=40(samples) f_s=8kHz)



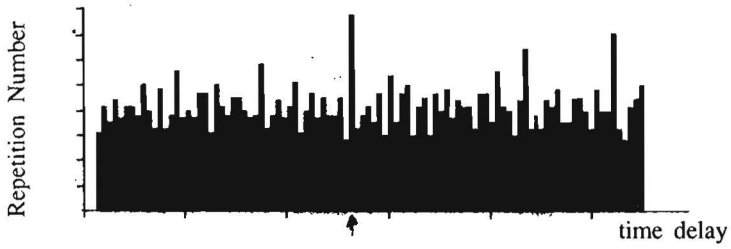
(a) Envelop correlations



(b) Accumulated values of envelop peaks



(c) Appearance of "carrier" period multiples



(d) Accumulated repetition number of the carrier perid multiples

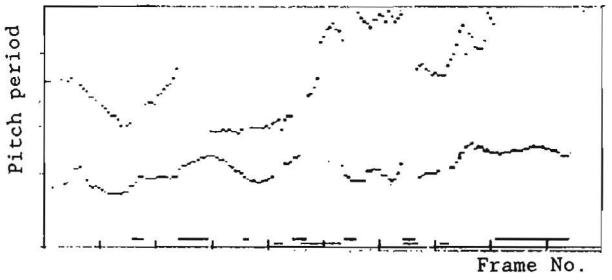
Fig.3.4(2) Pitch candidate estimation from speech corrupted by white noise (SNR = 6 dB p=47(samples) f_s=8kHz)

Pitch candidate estimation from summed synthetic speech sentences

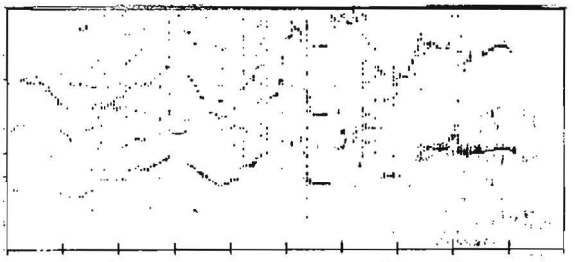
Another situation is to estimate simultaneously two-speakers' pitch candidates from summed **synthetic speech sentences** at low TIR. The pitch candidates are estimated from the signal "carriers" and the signal "envelopes", respectively. After that, the pitch candidates are peak enhanced, center clipped and thinned. Fig.3.5 shows the enhanced pitch candidate figures which are estimated from summed synthetic speech sentences of two speakers, i.e., female-male and female-female, in 0 dB TIR.

We noticed that the pitch information obtained from the signal envelope and from the signal "carrier" often complements each other. Consequently, better pitch candidates are provided.

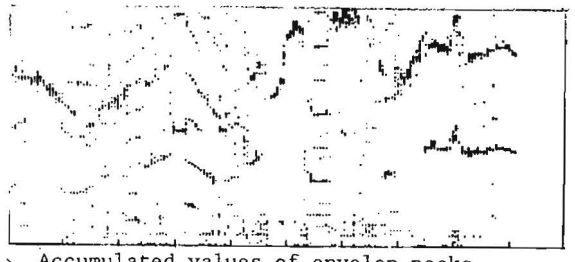
Also we noticed that when the ideal pitch contours of the two speakers undergo many times of crossing, and are too close to each other, it is still difficult to get sufficiently good candidate contours, as in the example of summed female-female case.



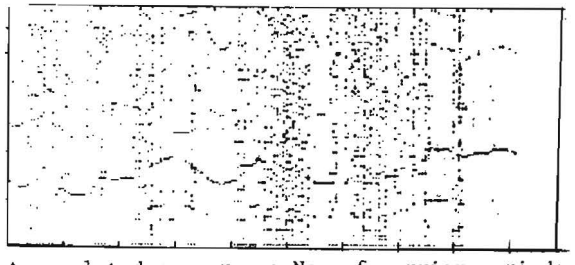
(a) Ideal pitch contours (female + male)



(b) Accumulated appearance No. of envelop peaks



(c) Accumulated values of envelop peaks



(d) Accumulated appearance No. of carrier periods

Fig.3.5(1) Pitch candidate estimation from a summed speech sentence (TIR=0dB)
S_1: "We do have a lot of good people in the office" by female
S_2: "You will now have fifteen seconds to do this" by male

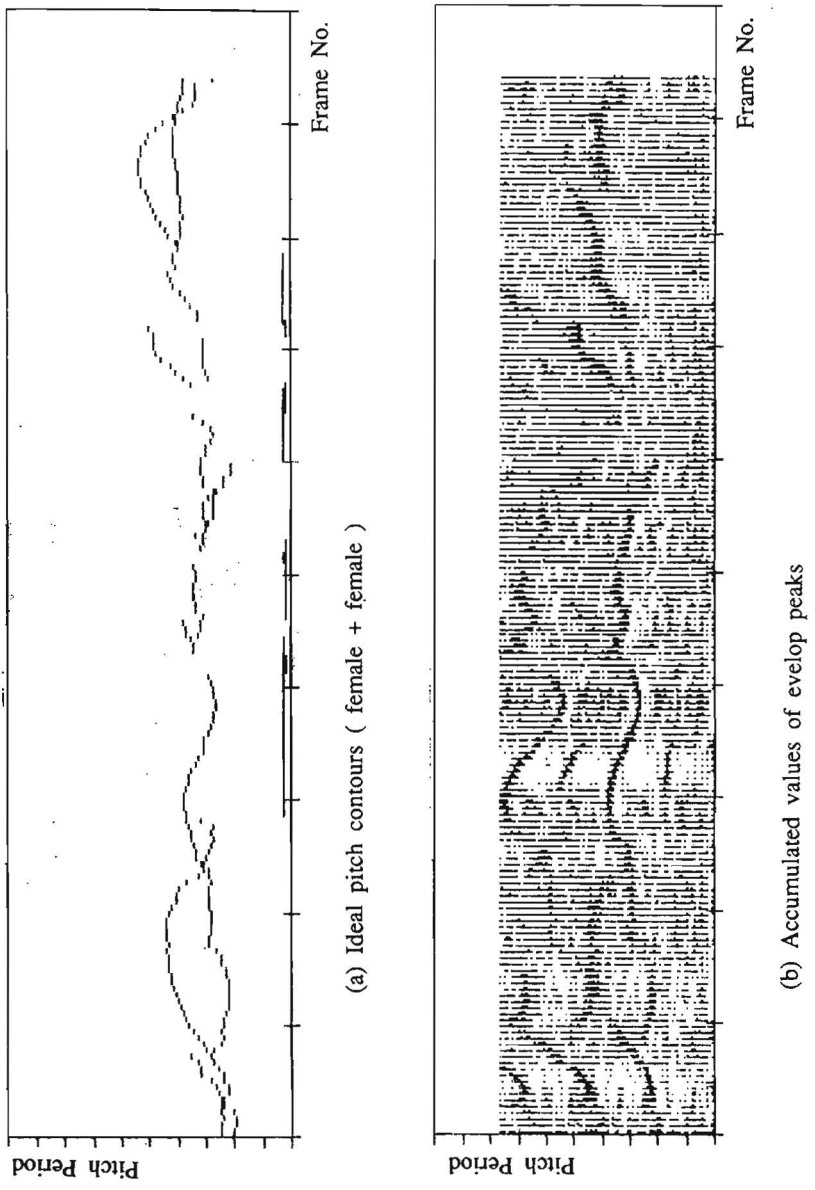


Fig.3.5(2) Pitch candidate estimation from a summed speech sentence (TIR=0dB)
S_1: "We do have a lot of good people in the office" by female
S_2: "Good morning, your passport please" by female

Remarks and discussion

The simulations proved that the pitch information is *independently* obtainable from either the signal "carriers" or from the signal envelopes. Thus, both of them can be used for pitch candidate estimation.

The simulations indicate that for extremely noisy speech, these two pieces of information can complement each other, and their combination produces better results. This phenomenon appeared in the simulations and can be explained as follows.
In principle, the envelope-based analysis has higher resolution when the bandwidth is wider, whereas the "carrier"-based analysis has higher resolution when the bandwidth becomes narrower. However, this bandwidth is a *relative* value associated with the speech fundamental-frequency (pitch). The fundamental-frequencies are time-varying within a sentence, and among different speakers. Thus, due to the wide dynamic range of fundamental-frequencies, a filter bandwidth may be considered as wideband during part of a sentence and as narrowband in another part of the same sentence. Consequently, it appears that in some parts the results from the envelope-based approach are weaker while those of "carrier"-based approach are sharper, or the vice versa, and that in some other part both the envelope-based and the "carrier"-based approach provide good resolution.

3.6.4. Concluding remarks

A new pitch candidate estimation algorithm has been proposed which is robust against various distortions on speech signals such as white noise and interference speech.

The following main features hold for this algorithm:

-The algorithm exploits the coincident appearance of pitch information contained either in the signal **envelopes** or in the signal "**carriers**". Their association with the speech fundamental frequency (pitch) is indicated by signal analysis.
This method implicates also the pitch perception theories of Licklider and Goldstein.

-Our simulations also proved that the "carriers" of either the narrow- or the

wide-bandpass filtered signals can be used independently for pitch estimation.

-For extremely "noisy" synthetic speech (the presence of background speech or other kinds of noise in high intensity), the pitch information obtained respectively from the signal "carriers" and the signal envelopes can complement each other. It is found that better estimation is obtained in such noisy circumstances by combining these two pieces of information.

-The algorithm is consistent with the auditory global processing without mimicking its behavior.

The preliminary simulations on noisy synthetic speech signals confirmed the robustness of the algorithm. More simulations are needed on the natural speech signals.

3.7. Hidden Markov Model-based maximum likelihood pitch contour estimation

- A detailed pitch contour estimation algorithm based on stochastic model

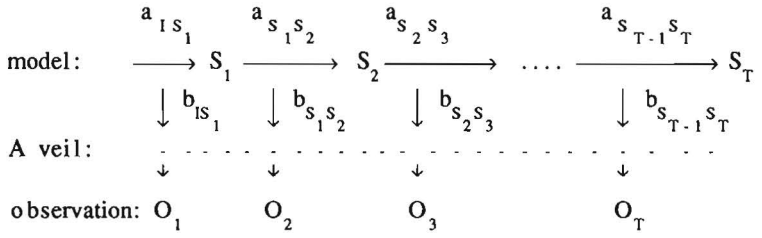
In this section, a new HMM-based algorithm is developed for the Maximum Likelihood (ML) estimation of pitch contours from candidates with a-priori probabilities obtained from a coarse algorithm. The theory of HMM modeling of pitch contours as well as a corresponding algorithm for training and ML pitch contour estimation are described. In order to limit the memory use and to solve the problem of missing candidates, we also describe a practical algorithm with candidate prediction, pruning, and beam search. The system is trained by a set of pitch contours from noise-free speech data.

3.7.1. Brief introduction of HMM theory

Hidden Markov Modeling (HMM) [40,49,73,76] is a probabilistic technique for the study of observed items arranged in a discrete-time series. The items can be countable or continuously distributed; they can be scalars or vectors. The technique uses stochastic methods; a time series is generated and analyzed by a parametric probability model.

An HMM has two components: a finite state Markov chain and a finite set of output probability distributions. The Markov chain synthesizes a sequence of states (a path) and the output distributions then turn this path into a time series. Thus, an observed time series gives evidence about the hidden path and the parameters of the generating model.

In an HMM, the output probabilities impose a "veil" between the state sequence and the observed time series, as shown schematically in Fig.3.6. In the effort to lift the veil, a substantial body of theory has been developed over the past decades. The initial work dealt with the probability spaces and addressed the problems of tractability of probability computation, the recovery of the hidden states, iterative maximum-likelihood estimation of model parameters from the observed time series and the proof of consistency of the estimates[6,7,8].



where: S_i: state i O_i: output observation

$a_{S_iS_j}$: transition probability from S_i to S_j

$b_{S_iS_j}$: output probability associated with arc between S_i and S_j

Figure 3.6 HMM: a veil between the observations and the model

An HMM is defined as a collection of states connected by transitions. Each transition carries two probabilities: a transition probability which provides the probability for taking this transition between states, and an output probability density function (pdf) which defines the conditional probability of emitting each output symbol from a finite alphabet when a transition is taken.

In a first-order HMM, there are two assumptions. The first is the Markov assumption:

$$p(X_{t+1}=x_{t+1} \mid X_1^t=x_1^t) = p(X_{t+1}=x_{t+1} \mid X_t=x_t) \qquad (3.7.1)$$

which states that the probability that the Markov chain is in a particular state at time instant (t+1) depends only on the state of the Markov chain at time instant t, and is conditional independent of the past. In (3.7.1), $X_t=x_t$ means that a state random variable X_t takes a specific value x_t at time instant t, and $X_1^t=x_t^t$ means that the random variable of the state sequence $X_1^t=(X_1...X_t)$ takes a specific value $x_1^t=(x_1...x_t)$.

The second assumption is output-independency:

$$p(Y_t=y_t \mid Y_1^{t-1}=y_1^{t-1}, X_1^{t+1}=x_1^{t+1}) = p(Y_t=y_t \mid X_t=x_t, X_{t+1}=x_{t+1}) \qquad (3.7.2)$$

which states that a particular symbol will be emitted at time instant t depending only on the transition taken at that time instant (from state x_t to x_{t+1}), and it is conditionally independent of the past. In (3.7.2), $Y_t=y_t$ represents that an output random variable Y_t takes a specific observation value y_t at time instant t, and $Y_1^{t-1}=y_1^{t-1}$ represents that an output random sequence $(Y_1...Y_{t-1})$ takes a specific observation sequence $(y_1...y_{t-1})$.

There are three typical problems of interest associated with HMMs:

*** The evaluation problem**

Given a model and a sequence of observations, what is the probability the model generating the observation sequence? Or more precisely, what is the probability $p(Y_1^T=y_1^T)$ for a given model M? Using the assumptions of HMM, we can manipulate it as

$$p(Y_1^T=y_1^T)= \sum_{X_1^{T+1}} \prod_{t=1}^{T} \left\{ p(X_{t+1}=x_{t+1} \mid X_t=x_t)\ p(Y_t=y_t \mid X_t=x_t, X_{t+1}=x_{t+1}) \right\} \qquad (3.7.3)$$

This probability can be calculated by using the Forward algorithm[40].

*** The decoding problem**

Given a model and a sequence of observations, what is the most likely state sequence in that model, producing the observations? The best we can do is to produce the state sequence that has the highest probability of being taken while generating the observation sequence i.e. choose a specific state sequence $X_1^{T+1}=x_1^{T+1}$ such that $p(X_1^{T+1}=x_1^{T+1} \mid y_1^T)$ is maximum. Using the HMM assumptions, this becomes

$$\max\left\{ p(X_1^{T+1}=x_1^{T+1} \mid y_1^T) \right\} = \max_{x_1^{T+1}} \left\{ p(Y_1^T=y_1^T \mid X_1^{T+1}=x_1^{T+1}) p(X_1^{T+1}=x_1^{T+1}) \right\} \qquad (3.7.4)$$

This can be calculated by using the Viterbi algorithm[40].

*** The learning problem**

Given a model and a set of observations, how to determine the parameters of the model such that it has a high probability of generating the observation? By defining the probability $\gamma_{ij}(t)= p(X_t=i,X_{t+1}=j \mid y_1^T)$ as the probability that the model is in state i at time t and in state j at time (t+1), given a specific observation sequence y_1^T, the transition probabilities can then be calculated as

$$a_{ij} = \sum_{t=1}^{T} \gamma_{ij}(t) \; / \; \sum_{t=1}^{T} \sum_{k} \gamma_{ik}(t) \qquad (3.7.5)$$

and the output (discrete) probabilities can be calculated as

$$b_{ij}(k) = \sum_{t : y_t = k} \gamma_{ij}(t) \; / \sum_{t=1}^{T} \gamma_{ij}(t) \qquad (3.7.6)$$

or, the parameters $\bar{f}$ of the output (continuous) pdf can be calculated as

$$\bar{f}_{ij} = \sum_{t=1}^{T} p(y_t \mid X_t=i,\; X_{t+1}=j) f(y_t) \; / \sum_{t=1}^{T} p(y_t \mid X_t=i,\; X_{t+1}=j) \qquad (3.7.7)$$

where a_{ij} is the transition probability from state i to j, $b_{ij}(k)$ is the discrete output probability that a symbol k is observed given the condition that the model take a transition form state i to j at that time, $f(y_t)$ is a function of y_t, and $\bar{f}_{ij}$ is the estimated parameter. (e.g.: In Gaussian output pdf case, two parameters need to be estimated: the mean value μ_{ij} and the variance σ^2_{ij}. The corresponding $f(y_t)$ are then y_t and $(y_t-\mu_{ij})^2$, respectively).
The γ_{ij} can be estimated by using the Baum-Welch (The Forward-Backward) algorithm[40].

Because of the veil between the output sequences and the state sequences produced by the *hidden* states in a given model, an HMM is a very powerful tool for many practical applications. For example it can be used to describe nested or implicative relations by stochastic models, provided that a large amount of data is available from measurements.

In the following, we will describe a new application of HMMs to the pitch

contour estimation. The remaining part of the section 3.7 is organized as follows. First, the theory of pitch contour estimation via HMMs is described after a short review of the basic theory. The algorithm will then be described in detail. After that, some simulations and results will be given. Finally a short discussion and concluding remarks are given.

3.7.2. Theory for HMM pitch contour estimation

In this section, we will concentrate on the following problem: given an array of weighted candidates obtained from the results of a coarse algorithm, how to estimate the "best" pitch contour.

A **new** approach of HMM pitch contour estimation[35] has been proposed. The reason that an HMM is chosen is that, some previous knowledge and a proper stochastic model should be used in order to handle complicated situations. This algorithm is designed to solve a general class of pitch contour estimation problems including speech signals corrupted by various kinds of noise.

HMM models are suitable to describe pitch contours

-First, a pitch contour changes continuously due to the real speech production model. A Markov process is suitable to describe highly correlated, continuously changing curves.

-With HMMs, each output sequence can be produced by many different (hidden) state sequences but with different output probabilities. Consequently, by avoiding to use a pitch sequence directly as a state sequence, it can provide more robustness against noise disturbances.

Pitch contour estimation via HMMs

In the Hidden Markov Modeling of pitch contours, pitch sequences are described by a family of models $M = \{m_i \mid i=1,2...\}$, each model is an HMM process of a 5-tuple, (S,Q,Π,A,B), representing respectively the sets of states, symbols of quantization, initial and transition probabilities, and the output parameter sets associated with Gaussian pdf's.

Three hidden states S = $\{s_1, s_2, s_3\}$, the constant pitch, the pitch increment and the pitch decrement, have been chosen in full inter connection, as shown in Fig.3.7.

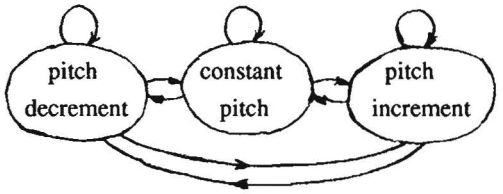


Fig 3.7 State diagram for HMM pitch estimation

The output pdf is represented by a joint multivariate distribution in order to describe the probabilities of the output pitch and pitch dynamics. The observed pitch Y_t, and its different-order derivatives $d_i Y_t$, i=1,2..., are chosen as the random variables. More random variables can be selected to introduce more constraints on pitch contours against noise distortions. In our tests, 3 random variables, the pitch value, and its first- and second-order derivatives, are used. The probability of a given model m_s (e.g. female/male/children) producing an output pitch sequence $(Y_1..Y_N)$ can be obtained by summing the probability of each specific state sequence producing the observations, over all the possible state sequences $(X_1..X_N)$, where $(X_1..X_N) \in \mathfrak{X}_N$ which is the set of state sequences, and $X_i \in S$.

$$p(Y_1..Y_N) = \sum_{(X_1..X_N) \in \mathfrak{X}_N} p(Y_1..Y_N;\ d_1Y_2..d_1Y_N;\ d_2Y_3..d_2Y_N;\ X_1..X_N) \qquad (3.7.8)$$

Supposing that the random variables Y_t, $d_1 Y_t$ and $d_2 Y_t$ are statistically independent, the joint output pdf can then be expressed as the products of the marginal pdf's. Using the output independence assumption, this can be expressed as follows

$$p(Y_1..Y_N) = \sum_{X_1^N} p_0(Y_1..Y_N|X_1^N)\ p_1(d_1Y_2..d_1Y_N|X_1^N) * p_2(d_2Y_3..d_2Y_N|X_1^N)\ p(X_1^N)$$

$$= \sum_{X_N^1 \in \mathfrak{X}_N} p_{BX_1} p_0(Y_1|X_1)p_0(Y_2|X_2)p_1(d_1Y_2|X_2)p(X_2|X_1)*$$

$$*\prod_{t=3}^{N}\left(p_0(Y_t|X_t)p_1(d_1Y_t|X_t)p_2(d_2Y_t|X_t)p(X_t|X_{t-1})\right)p_{X_N E} \qquad (3.7.9)$$

where $d_1Y_t=(Y_t-Y_{t-1})$, $d_2Y_t=(d_1Y_t-d_1Y_{t-1})$. $X_1^N=(X_1..X_N)$, B, E are the begin state and the end state respectively. The pitch contour is chosen among candidate contours associated with a maximum output probability.

3.7.3. HMM-based ML pitch contour estimation algorithm

Training-phase

The training process is performed in two steps: the supervised training followed by the unsupervised training.

The *supervised training* is used for the purpose of obtaining better initial parameter values.

The *unsupervised training* is then performed on a large set of pitch contours obtained from noise-free speech signals. In the unsupervised training, the parameters are trained using the corresponding Forward, Backward and Forward-backward algorithms with multivariate output. Since it is not difficult to derive these algorithms with multivariate output, and the results are clear and obvious, we will simply give those formulas.

In this training process, the forward probability $\alpha_i(t)=p(X_t=i,y_1^t)$, and the backward probability $\beta_i(t)=p(y_{t+1}^T|X_t=i)$ are calculated, recursively, as below (which correspond to the Forward algorithm and the Backward algorithm),

$$\alpha_i(t) = \sum_j \alpha_j(t-1)\, a_{ji}\, b0_{iy_t}\, b1_{id_1y_t}\, b2_{id_2y_t} \qquad (3.7.10)$$

$$\beta_i(t) = \sum_j \beta_j(t+1)\, a_{ij}\, b0_{jy_{t+1}}\, b1_{jd_1y_{t+1}}\, b2_{jd_2y_{t+1}} \qquad (3.7.11)$$

where a_{ij} is the transition probability from state i to j. The output probabilities $b0_{jy_t}$, $b1_{jd_1y_t}$ and $b2_{jd_2y_t}$ are defined as follows

$$b0_{jy_t} = p_0(y_t|X_t=j) \qquad (3.7.12)$$

$$b1_{jd_1y_t} = p_1(d_1y_t|X_t=j) \qquad (3.7.13)$$

$$b2_{jd_2y_t} = p_2(d_2y_t \mid X_t=j) \tag{3.7.14}$$

The forward-backward probability $\gamma_{ij}=p(X_t=i,X_{t+1}=j \mid y_1^T)$ can be calculated as follows (which corresponds to the Forward-Backward algorithm)

$$\gamma_{ij}(t)= \alpha_i(t)\ a_{ij}\ b0_{jy_{t+1}}\ b1_{jd_1y_{t+1}}\ b2_{jd_2y_{t+1}}\ \beta_j(t+1)/\ \alpha_{s_E}(T) \tag{3.7.15}$$

where $\alpha_{s_E}(T)= p(Y_1^T=y_1^T)= \sum_i \alpha_i(t)a_{is_E}$.

In order to prevent numerical underflow when the length of the output sequence is increased, the forward and backward probabilities can be scaled as follows,

$$\tilde{\alpha}_i(t) = \alpha_i(t)/ \sum_j \alpha_j(t), \quad \tilde{\beta}_i(t) = \beta_i(t)/ \sum_j \alpha_j(t) \tag{3.7.16}$$

The corresponding scaled algorithms are then as below

$$\tilde{\alpha}_i(t) = \frac{\sum_j \tilde{\alpha}_j(t-1)a_{ji}b0_{iy_t}\ b1_{id_1y_t}\ b2_{id_2y_t}}{\sum_{ij} \tilde{\alpha}_j(t-1)a_{ji}b0_{iy_t}\ b1_{id_1y_t}\ b2_{id_2y_t}} \tag{3.7.10'}$$

$$\tilde{\beta}_i(t) = \frac{\sum_j \tilde{\beta}_j(t+1)a_{ij}b0_{jy_{t+1}}\ b1_{jd_1y_{t+1}}\ b2_{jd_2y_{t+1}}}{\sum_{ij} \tilde{\beta}_j(t+1)a_{ij}b0_{jy_{t+1}}\ b1_{jd_1y_{t+1}}\ b2_{jd_2y_{t+1}}} \tag{3.7.11'}$$

$$\gamma_{ij}(t) = \tilde{\alpha}_i(t)\ a_{ij}\ b0_{jy_{t+1}}\ b1_{jd_1y_{t+1}}\ b2_{jd_2y_{t+1}}\ \tilde{\beta}_j(t+1)/\ \tilde{\alpha}_{s_E}(T) \tag{3.7.15'}$$

where $\tilde{\alpha}_{s_E}(T)=\sum_i \tilde{\alpha}_i(t)a_{is_E}$. They are used for estimating the Gaussian pdf parameters as follows

$$\mu_{k,j} = \sum_{i,t} \gamma_{ij}(t)\ f_k(y_{t+1})/ \sum_{i,t} \gamma_{ij}(t) \qquad k=0..2,\ j=1..3 \tag{3.7.17}$$

$$\sigma^2_{kj} = \left[\sum_{i,t} \gamma_{ij}(t)\ f_k^2(y_{t+1})/ \sum_{i,t} \gamma_{ij}(t) \right] - \mu^2_{k,j} \tag{3.7.18}$$

where $f_0(y_t)=y_t$, $f_1(y_t)=d_1y_t$, $f_2(y_t)=d_2y_t$. The different models are trained

separately by the *quantized* pitch sequences from each group of speakers.

To prevent excessive constraints, in each model (e.g.: male/female model) several states share the same output pdf. This is done by choosing the conditional output pdf's of the random variables Y and d_2Y independent of states, as follows

$$p_0(Y \mid X=j) = N(\mu_0,\sigma_0^2) \qquad j=1..3 \tag{3.7.19}$$

$$p_2(d_2Y \mid X=j) = N(\mu_2,\sigma_2^2) \qquad j=1..3 \tag{3.7.20}$$

which depend only on the different model, where $N(\mu,\sigma^2)$ denotes the Gaussian pdf with mean-value μ and varaince σ^2. It means that $p_0(Y \mid X)$ mainly plays the role of distinguishing among different models, and $p_2(d_2Y \mid X)$ mainly serves as the constraint on pitch acceleration. Separately training the pdf's indicates that such selections of mergence are reasonable, as shown in Fig. 3.8.

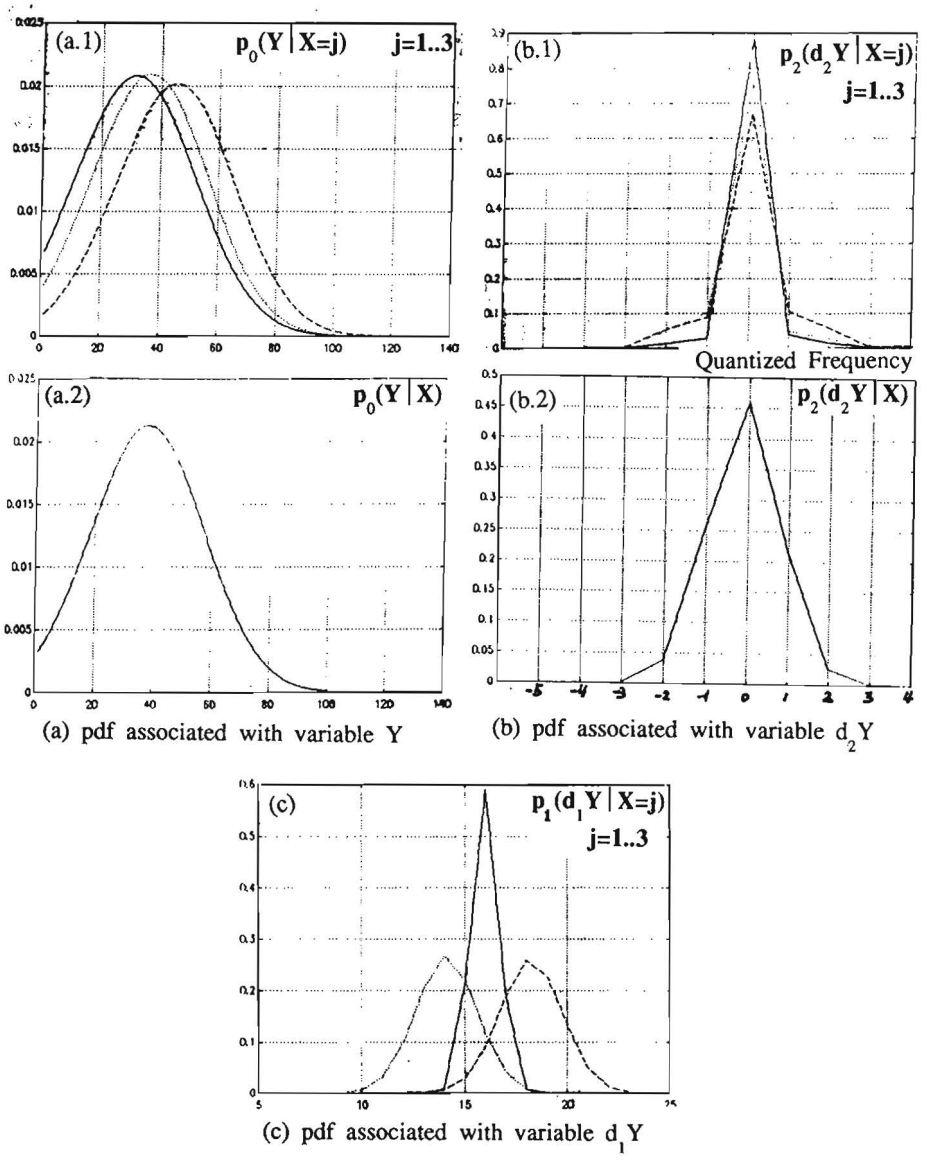


Fig.3.8 Output probability distributions in a HMM-based pitch contour model

(a.1) (b.1) (c) : Results from state-dependent training
(a.2) (b.2) : Results from state-independent training

The output pdf associated with random variable d_1Y is depending on the state,

$$p_1(d_1Y \mid X=j) = N(\mu_{1,j}, \sigma^2_{1,j}), \quad j=1..3. \tag{3.7.21}$$

Estimation-phase

Suppose that an array of pitch candidates, representing the pitch information from a speech sentence of L-frames, is obtained from the coarse algorithm. At an arbitrary (n^{th}) frame, a vector of weighted pitch candidates $Z_n=\{ Y_n^{k_i} \mid k_i = 1,2 .. L, p(Y_n^{k_i}) = \text{(weight value) and } \sum_{k_i=1}^{L} p(Y_n^{k_i})=1 \}$ is provided. A candidate contour ($Y_1^{k_i}\ Y_2^{k_1}\ ..\ Y_N^{k_n}$) is defined as a sequence formed by taking one element from each vector Z_i, i=1..N, sequentially. It is associated with the a-priori probability

$$p_{pri}(Y_1^{k_i}\ Y_2^{k_1}\ ...\ Y_N^{k_n}) = p(Y_1^{k_i})p(Y_2^{k_1})\ ...p(Y_N^{k_n}) \tag{3.7.22}$$

Given the a-priori probability, the estimation-phase then calculates the output probability of each candidate contour produced by a specific model m_s

$$p_F(Y_1^{k_i}..Y_n^{k_t} \mid m_s) = p_{pri}(Y_1^{k_i}..Y_n^{k_t})p(Y_1^{k_i}..Y_n^{k_t} \mid m_s) \tag{3.7.23}$$

and selects one "best" contour associated with the maximum probability.

In order to search for a global optimally sequence, the output probabilities of all possible pitch sequences are calculated in parallel by the Forward-algorithm. The probability at the n^{th} frame can be calculated recursively from the results obtained at the $(n-1)^{th}$ frame,

$$p_F(Y_1^{k_i}...Y_{n-1}^{k_s}Y_n^{k_t} \mid m_s) =$$

$$=p(Y_n^{k_t}) \sum_{i,j} \left(p_F(X_{n-1}=j, Y_1^{k_i}...Y_{n-1}^{k_s} \mid m_s) a_{ji} b0_{iy_n^{k_t}} b1_{id_1y_n^{k_t}} b2_{id_2y_n^{k_t}} \right) \tag{3.7.24}$$

A best-candidate contour is then selected as follows under Maximum Likelihood (ML) criterion

$$p_F(Y_1...Y_N \mid m_s) = \max_{m_s,(k_i(1)..k_n(N))} p_F(Y_1^{k_i}...Y_N^{k_n} \mid m_s) \tag{3.7.25}$$

The practical estimation algorithm

Practically when the frame number increases, the problems of excessive use of memory space and large calculations appear due to many candidate sequences appearing. Besides, the wanted pitch candidate can be missing from the coarse estimation. The following approaches are used to solve these problems, while keeping the error rate at a negligible low level.

* Beam search

In reality, pitch values change slowly between successive frames due to the continuity constraint. The probability that two successive candidates are far away is very small. Thus, beam search is applied instead of full search. i.e., for a given sequence $(Y_1^{k_i} \ldots Y_{n-1}^{k_s})$ of length (n-1), the search of its possible sequences $(Y_1^{k_i} \ldots Y_{n-1}^{k_s} \; Y_n^{k_t})$ of length n at the succeeding n^{th} frame is limited within the range of $Y_{n-1}^{k_s} \pm W_1$, where W_1 is a given threshold which is bigger than the maximum pitch change between successive frames.

* Pruning

Pruning is performed to limit the number of candidate contours in intermediate frames, by giving a scaled probability threshold. Pruning is also done when one contour splits into multi-contours and soon afterwards merge again.

* Prediction of missing candidates

Prediction over short and continuous frames is allowed. A very small a-priori probability value is given to the predicted candidate, such that a long continuous prediction will lead to small total probability score for the whole sequence. Thus the chance that a long predicted sequence will be selected as the estimated contour is very small.

* Length-weighted sequence a-priori probability

Due to the beam search and pruning, some of the candidates will be disconnected from the candidate pitch contours. These candidates are then formed as the roots of new contours. Consequently, candidate sequences may

have different lengths. Thus, it is necessary to weight the a-priori probability value of each sequence by its length.

Remarks and discussions:

1) Comparisons to the smoothing and DP algorithms

The advantages of HMMs over a pitch contour smoothing algorithm or a Dynamic Programming (DP) algorithm are obvious.

In a simple pitch contour smoothing algorithm, the wrongly estimated pitch value is smoothed out if there is a big jump among the successive pitch values, and then replaced with an interpolated pitch value such that the cumulate distance is minimized.
In a DP algorithm, one can use a cumulative probability measure, but the global optimum can only be obtained if it can be expressed by the sum of all the local maxima.

HMMs-based algorithms, however, estimate a pitch contour by "best fitting" to a given model. Because the model can be properly selected, and can be well trained by noise-free pitch contours prior to the estimation from noise contaminated signals, the approach is expected to have potentially more robustness for pitch estimation.

2) When to obtain the estimated pitch value

In the algorithm, estimation is off-line processing. A decision is postponed to the end of a sentence or a continuous piece of voiced-frames. When one desires to do real-time processing by making a decision after each analysis frame, degraded performance can be expected.

3) Obtaining candidates by other pitch estimation algorithms modified

It is also possible to estimate pitch candidates from one of the existing pitch estimation algorithms, according to the complexity of the input speech. As we have mentioned previously, in order to prevent errors by making unproper early decisions based on isolated frames, this selected algorithm must be modified to provide a set of weighted pitch candidates

rather than an estimated pitch value. These algorithms can then be served for coarse pitch estimation purposes.

4) Applications

The HMM-based pitch contour estimation algorithm can serve as a robust pitch contour estimator by post-processing the results of the conventional pitch estimation algorithms.

5) Limitations

In the presently developed algorithm, it is limited to the voiced-speech frames. Segmentation of voiced-unvoiced frames is not included, and thus must be judged elsewhere. To improve this, one can consider to add extra states to include the unvoiced case.

3.7.4. Simulations and results

In the program, a pointer and a binary tree structure are used, such that the branches of candidate contours can be added and pruned dynamically.

Some simulations have been done for speech signals corrupted by both white noise and background speech at low SNR around 0 dB.
It showed that candidate prediction produces reasonably good results in the case of missing candidates during continuous short frames. When candidates are missing, prediction plays an important role for the algorithm to yield a good estimation.

Two examples in which speech signals are corrupted by white noise at 0 dB SNR are included. In order to test the robustness of the algorithm for tracing pitch contours, a simple coarse pitch estimation algorithm of signal autocorrelation and peak picking is selected, so that the obtained candidates array is far from ideal. In many places, it is even difficult to figure out the options visually. Fig.3.9. and Fig.3.10. then shows the results of the estimated pitch contour via HMMs.

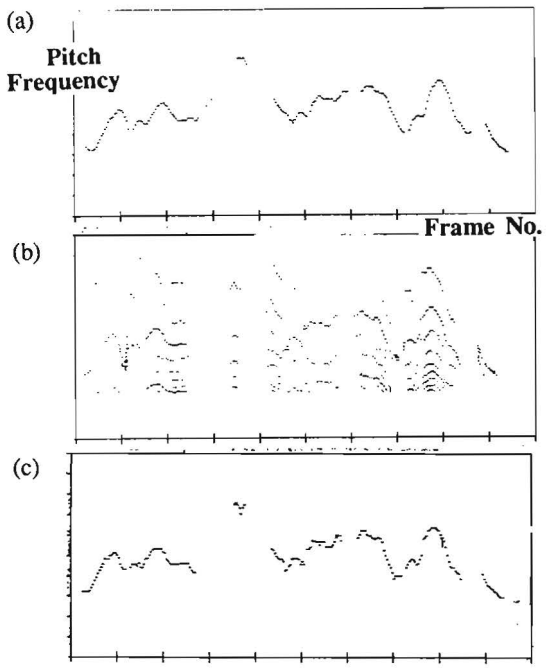


Fig.3.9 Pitch contour estimation via HMMs

S_1:"The engineer discovered an irregularity"
SNR = 0dB (S_1 + white noise)

(a) Ideal pitch contour
(b) Pitch candidates from a coarse estimation algorithm
(c) Estimated pitch contour from HMMs

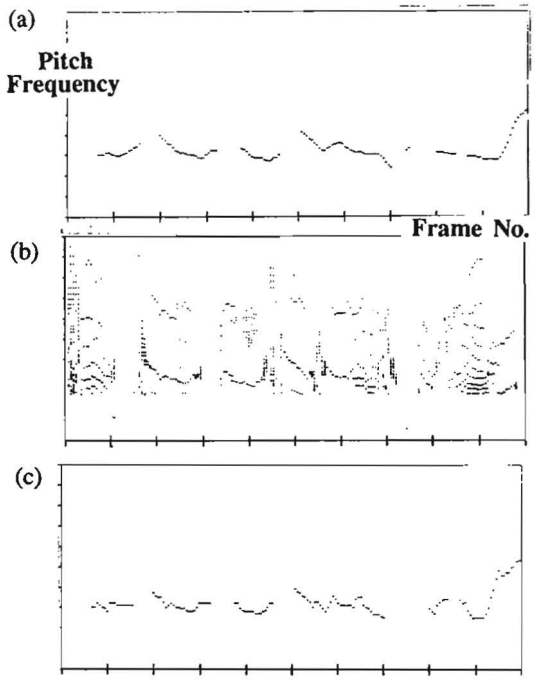
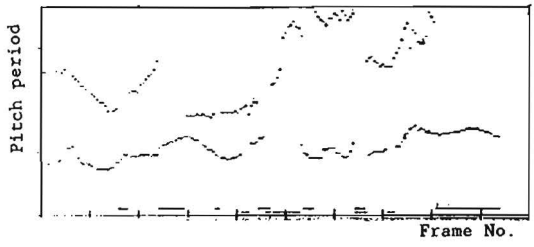


Fig.3.10 Pitch contour estimation via HMMs

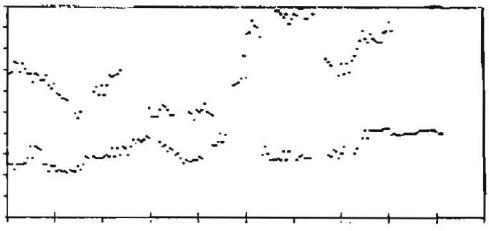
S_2:"So can you find somethingelse you would like?"
SNR = 0dB (S_2 + white noise)

(a) Ideal pitch contour
(b) Pitch candidates from a coarse estimation algorithm
(c) Estimated pitch contour from HMMs

Simulations have also been done on estimating pitch contours from summed speech signals of one female and one male speaker at 0dB. In the training phase, two models, the female and the male, are trained separately by a number of quantized pitch sequences from each group of speakers. Fig.3.11 shows the estimated pitch contours using the previously estimated pitch candidate results as given in Fig. 3.5.(1) (female-male case). Both the candidate files from the signal "carriers" and the signal envelopes are used. These two pieces of information are combined into one candidate file with a-priori probabilities.



(a) Ideal pitch contour (female + male)



(b) Estimated pitch contour (using candidate file in Fig.3.5(a))

Fig.3.11 HMM-based pitch contour estimation from summed speech

S_1: "We do have a lot of good people in the office" by female
S_2: "You will now have fifteen seconds to do this" by male
SNR = 0dB ($S_1 + S_2$)

Preliminary simulations indicate that in order to obtain reasonably good results, a relatively accurate coarse algorithm must be selected, due to the interaction of the pitch candidates from the two speakers (including their harmonics and sub-harmonics). If candidates are not good enough, for example, the pitch candidates for one speaker are missing during continuous frames, some difficulties may appear. It appears that the wanted pitch sequence may still be correctly predicted and estimated with relatively high probability score, but is not associated with the maximum one. This is because of too many prediction frames and the relatively small weight values from the coarse estimation algorithm.

3.7.5. Concluding remarks

By modeling the pitch contours, the HMM-based algorithm provides robustness in pitch estimation against noise. In addition, beam search, pruning, candidate prediction and candidate sequence a-priori probability are used to solve some practical problems. The algorithm showed to be promising in simulations. However, if the candidate array provided by a coarse algorithm is very poor in performance, the algorithm may fail to estimate correctly.

3.8. Robust pitch estimation via a combined-algorithm

In the above two sections, we developed individually a coarse pitch candidate estimation algorithm, and a detailed algorithm of HMM-based ML pitch contour estimation.

A combined pitch contour estimation algorithm can then be formed as the following schematic Fig.3.12.

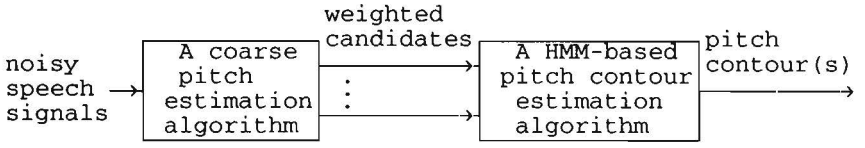


Fig 3.12 A proposed pitch contour estimation algorithm

The pitch candidates with their weighted values can be obtained by either a pseudo-perceptual pitch candidate estimation algorithm, or some other simpler estimation algorithms, depending on the complexity of the problem concerned. It is important not to ignore the weight values obtained from a coarse estimation algorithm, because it often provides good a-priori pitch information when the speech is stationary and not very noisy. An HMM-based ML pitch contour estimation algorithm is then used to search for the optimum contour.

3.9. Summary and conclusions

In this chapter, a new robust pitch estimation algorithm has been proposed which uses two processing steps. Key observations are:

The algorithm is based on the following point of views

* To combine the human perception advantages with the signal processing-based pitch estimation approach is absolutely necessary.

* To mimic the auditory behavior for the purpose of pitch estimation is neither necessary nor possible.

* To use all the existing information in both inter and intra analysis frames for pitch estimation is necessary.

* The complexity of the processing in the algorithm must be adaptive in accordance with the variable degree of contamination of the input speech.

The following *global* improvements are obtained by using this program

* To bridge the gap between the signal processing-based and perceptual-based algorithms. The algorithm is consistent with the auditory global processing.

* To postpone the decision of choosing a pitch value from each frame of speech. Rather, pitch candidates are used as an initial guess, in order to prevent unproper early decisions based on insufficient information from isolated frames.

* The complexity of the proposed algorithm can be adapted. In a simple case,

the coarse pitch estimation algorithm can be replaced by a simple one, which is revised to provide weighted pitch candidates.

* For extremely noisy speech input, more complex processing is used, including a better initial guess algorithm (e.g.: a pseudo-perceptual algorithm) and the use of a stochastic model and the previously trained knowledge.

Advantages of using a pseudo-perceptual pitch candidate estimator

* Pitch estimation can be *independently* performed by using the coincident appearance of the signal "carrier" multiples or the signal envelope autocorrelations. These two pieces of information often complement each other.

* Simplifications are obtained over the perceptual type of algorithms.

* Robustness is obtained in providing rich pitch information from pitch candidates.

Advantages of using an HMM-based pitch contour estimator

* Using a stochastic model to describe pitch contours rather than using the smoothing or DP algorithm based on minimum cumulative distance/probability measure.

* Because of the veil between the hidden states and the output in a HMM, it is potentially more robust against noise disturbance.

* A simple HMM can well describe the correlations of pitch contour and its continuity constraint.

* Because HMMs can be properly trained by a large set of noise-free data, the method is close to human pitch perception of using previously learned knowledge.

Conclusions and future work

The proposed pitch estimation algorithm showed its robustness and great potential in processing speech with variable contamination. The algorithm is

flexible in its design, with a replaceable coarse pitch candidate estimation part in accordance with the complexity of the input speech.

This algorithm could be refined, such as to include the unvoiced state in order to handle transitions between voiced and unvoiced frames, and to use, for example, forward combining backward search in order to handle better the pitch crossing points of two contours.

CHAPTER 4

SPEECH INTELLIGIBILITY ENHANCEMENT

A new adaptive speech separation system, designed for separating co-channel speech signals from a single-receiver with a range of Target-Interference Energy Ratio between -12 dB and +12 dB, is developed. With particular emphasis on the global consistency to the human auditory processing without emulating the detailed auditory behavior, this system bridges the gap between the methods by pure signal-processing and by pure mimicking of auditory speech perception.

The system mainly consists of two parts, the adaptive speech separation and the pitch contour estimation.
In the speech separation part, a new time-frequency bin domain adaptive speech separation approach is used, by separately exploiting the T-FB domain linear and nonlinear LMS adaptive filtering techniques as described in chapter 2.
In the pitch contour estimation part, a two-step estimation algorithm is applied, by exploiting a pitch candidate estimator plus an HMM-based pitch contour estimator as described in chapter 3.

Simulations on separating summed stationary speech signals with constant pitches, and on separating summed (nonstationary) speech sentences with constant and with natural pitch, will be described in detail. Some results will be included.

4.1. Description of the addressed speech intelligibility enhancement problem

*** Problem addressed**

The problem addressed by this chapter is the intelligibility enhancement of the target speech signal from a co-channel speech signal. A co-channel speech signal is defined as an additively combined signal from target speakers, from background interference speakers, and from various other kinds of noise in a single channel. Often we have to handle the situation

where there is no a-priori information about the target and the interference speakers and there is only a single-input from a single receiver rather than multi-inputs from an array of receivers (multi-receivers). The reason of selecting such a premise for this research is that in many practical situations the co-channel noisy-speech signal is the only signal available.

*** Fields of possible applications**

The addressed problem may appear in many different situations, such as in a microphone and a (mobile)telephone environment, from background competitive speakers, from crosstalk in neighboring communication channels or in frequency-reused channels (in the space domain), etc.
In many situations, we need to handle the problem of co-channel interference reduction. Some simple examples are given below.

- *Co-channel noise in a cellular mobile telecommunications system*
  In a cellular mobile telecommunications system, the reduction of co-channel interference becomes primarily important because of the introduction of frequency reuse[50]. In such a case, users in the different geographic locations may simultaneously use the same frequency channel from the different cells.

- *Co-channel noise in a car environment*
  The reduction of co-channel noise in a car environment is required for improving the speech quality in a mobile telephone receiver.

- *Presence of background-noise in an ASR system*
  Reduction of background noise in an Automatic Speech Recognition (ASR) system is of primary importance. It is well known that the speech recognition rate of ASR systems suffers significant degradation due to background interference noise.

- *Crosstalk and echo noise* in *a telephone system*
  The crosstalk from neighboring channels and echoes from the "hybrid" circuit (a transmission link) to a telephone receiver due to the impedance mismatch can degrade severely the speech quality in a telephone receiver.

- *Noisy-speech from competitive speakers in an environment with normal-hearing listeners and hearing-impaired*
  Difficulties arise for hearing-impaired people to understand the target-speech in noisy surroundings.
  Difficulties even arise for normal-hearing listeners when background speech from many competitive speakers is present at medium to high acoustic levels. In these situations, intelligibility improvement of the desired-speech is required. One of the situations is the well-known cocktail party, where it is often difficult to understand a target-speech from background competitive speech.

*** Difficulties in solving this problem**

After decades of investigations, co-channel speech enhancement remains a challenge. The key difficulty is that the target signal and the interference noise can be both speech signals, which implies that they can be both *nonstationary* and share *similar statistical characteristics*. In the conventional situations with white/colored noise, the spectra of signal and noise show a large difference. If both the signal and noise are speech, there is neither basic statistical difference in the time-domain nor in the frequency-domain.

We know that in principle whether a noise reduction technique can be effectively used mainly depends on two factors. First, there must exist some basic difference between signal and noise which can be exploited in a specific domain. The larger the differences, the easier the noise is being reduced. Secondly, it depends on whether suitable mathematical expressions and a corresponding algorithm can be developed to implement the idea.

However, there still exist *local signal differences* between the target and the interference speech which can be exploited to reduce the noise.

*** The human auditory processing**

If we could understand how the human auditory system processes noisy speech, and what characteristics it exploits, we might be able to develop an effective processing technique. Until now we are only starting to understand the highly intelligent mechanism of the human sound processing. However, the

current knowledge in this field will be helpful for developing our speech enhancement algorithm.

It should be emphasized that for the purpose of searching a proper signal processing technique of speech enhancement which can combine the human sound perception advantages, the auditory sound perception and the auditory speech processing are reviewed below. Thus, the detailed *micromechanism* of the auditory system is not essential for our algorithm, rather it serves for the purpose of a better understanding of human sound processing.

From the following review, it will also be clear that one can not expect to build a speech enhancement system by *emulating the auditory micromechanism* in order to fulfill such a task, since it is still unclear about the high level processing in the human brain although a relatively clear figure in the low-level auditory processing has been found.

4.2. Speech perception and auditory processing of noisy speech

In a noisy-room, most human listeners are able to perceive target speech, even though there is a lot of competitive speech at high acoustic levels.

The intelligibility of the human auditory system is at a far higher level than any existing signal processing technique. This observation motivated many researchers to investigate the human auditory system. It is obvious that a better understanding of the sound interpretation in the auditory system can help us to effectively improve the speech signal processing techniques.

Human auditory processing: an integration of multi-knowledge sources

From the observable facts, the following pieces of information are likely to be used by the human auditory system:

- *Visual information*

  People can understand better if they have face to face communication. During the communication, the observation of lip-movements can often help the speech understanding.

- *Binaural information*

Two ears can perceive sound better than only one ear. It is indicated that binaural information is used for distinguishing and perceiving a specific sound from a multi-source combined complex-sound. The binaural hearing system has the ability to infer the direction of different sound source by using the difference in sound intensity and arrival-time at the two ears.

- *Linguistic knowledge*

Human listeners can better perceive their mother language than foreign languages in noisy surroundings. Another fact from the experiments[99] showed that if one phoneme in a sentence is replaced by noise, the human auditory system can restore this. These indicate that sound perception needs some kind of high level processing associated with our previous knowledge.

- *Using multi-pieces of local information*

Listeners can perceive two or more simultaneous sounds. It has been noticed that the following pieces of information may play important roles in the auditory system speech perception.

**Pitch and pitch dynamics*

The more differences among the pitches in a mixed-sound, the easier for the listeners to follow a specific speech in that sound. It has also been found that big pitch changes are recognized by listeners as the presence of another sound source. These observations indicate that speech fundamental frequency (pitch) and its dynamics are very important for human speech perception.

**Formants and their dynamics*

The more differences among the speeches, the easier for listeners to distinguish the different speeches. As different speeches are characterized by their spectral peaks (formants) and the dynamic trajectories of those peaks, this indicated that the formant frequencies and their dynamics are used by the auditory system for sound analysis.

**Sound onset, offset and masking*

Some other pieces of information, such as sound onset and offset, sound mask and restoration, also appear to play important roles in sound perception.

**Dynamic complexity in sound processing*

In a complex situation more pieces of information are used rather than in a simple situation. A highly degraded sound needs more auditory processing. The complexity of auditory processing is thus adaptive.

- *The processing in the speech enhancement and in the speech recognition are mutually dependent*

From the above, we can also conclude that the processing of speech enhancement involves also the processing in the speech recognition. This is especially obvious from the auditory sound masking and restoration effect[99].

Analysis of the auditory processing

Much efforts have been done in understanding the structure and function of the peripheral portions of the auditory system. We have a relatively clear picture of the processing preceding the auditory nerve, however, we only know very little about the central processing at the higher level of the auditory system. The knowledge about the processing preceding to the auditory nerve can be summarized in Fig.4.1.

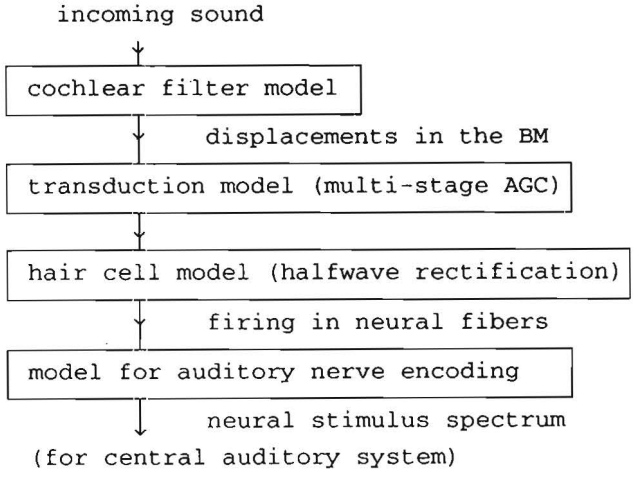


Fig.4.1 Schematic figure of auditory processing in low levels

The human ear can be partitioned into an outer, middle and inner part, as shown in Fig.4.2.

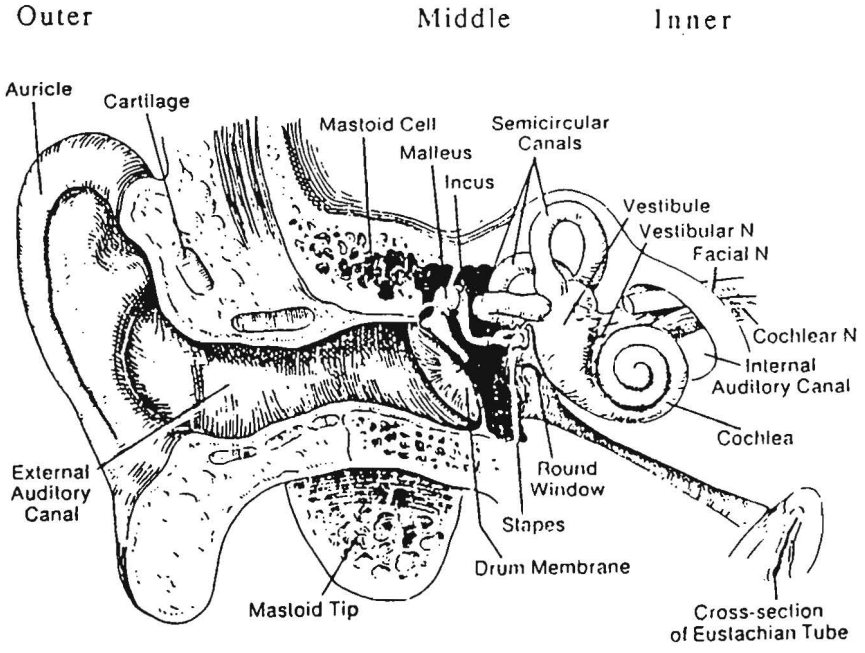


Fig. 4.2 Structure of the ear

outer ear

The transmission of a sound through the outer ear via the ear canal resonances introduces nonlinear effects which emphasize the frequency spectral components of the sound.

Middle ear

A complicated linkage of bones contained in the middle ear couples the movements of the eardrum to the oval window at one end of the cochlea in the inner ear. Thus, it transforms from air pressure variation to fluid pressure variation in the cochlea.

Inner ear

The Basilar Membrane (BM), which is a fibrous tissue extending through the middle of the cochlea, is a vital part for the hearing process. Each specific *place of the BM* reaches a maximum response on the stationary envelop of the travelling wave along the cochlea, as shown in the tuning curve of the BM in Fig.4.3. Hence, the frequency components of a sound are transformed to monotonous displacements along the BM places.

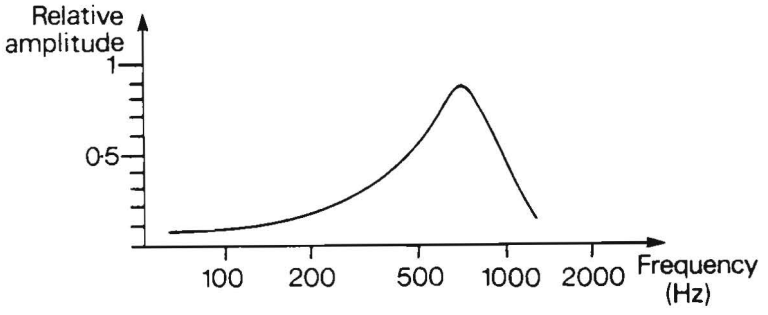


Fig.4.3 Response of the BM as a function of frequency: one tuning curve at a specific place of the BM

The Organ of Corti includes three rows of outer hair cells and one row of inner hair cells, which reside on the BM. The *transduction process* takes place in this organ. The movement of the BM causes the bending of the hair cells which stimulates the firing of the neurons of the auditory nervous. Thus, the frequency-selective displacements of the BM are changed into *neural response.*

The inner hair cells exhibit a kind of "stimulus selectivity". Each inner hair cell bends optimally to stimuli at a characteristic frequency and then causes the firing of neurons of the auditory nerve.

The outer hair cells are not sensory cells, rather they are the effectors of Automatic Gain Control (AGC) loop which modulates the mechanical motions of the BM. The outer hair cells, which are normally inhibited, act as "muscles" which can amplify the effect of low-level stimuli when the inhibition is reduced.

Auditory nerves communicate the response of hair cells to the nervous system. The representation of acoustic information in the nerve is important because it is the only source of information available to high levels of auditory processing.

The critical bandwidth and the masking effect in the auditory system

The masking effect plays a very important role in the auditory system speech analysis. *Critical bandwidth* is associated with the frequency resolution capabilities of the auditory system. It is around 100 Hz for center frequencies below 1 kHz, and is about 15% of the center frequencies above 1 KHz. Psycho-physical research indicated that tones within the critical bandwidth can not be perceived individually because of the frequency-resolution in the human auditory system.

Tones which are several dB lower than the noise within a same critical bandwidth can not be perceived because of the masking effect. Hence, The human auditory system is not sensitive to the detailed spectral structures of a sound within this bandwidth. Rather a weighted integration over all the tones above the perceivable level within the band is performed.

Computational model for auditory low level processing

Several existing models are based on the above knowledge.

- *Transmission line model of the cochlea*

  One of basic cochlear filter models is the transmission line model, or the one-dimensional model. This model describes the transformation of the travelling sound wave to the mechanical movement of the BM. Each small section along the cochlear spiral is modeled as a section of transmission line. A transmission line with a low transmission velocity is, however, difficult to realize physically. A more convenient approach is to use filters, each of which represents the filter characteristic at a single point of the BM. Thus, the incoming sound is processed through a group of cochlear filters. The different places along the BM are tuned monotonously to the specific frequency band.

  To mimic such functions, a group of cochlear filters can be used. The interval between the center frequencies of these filters is equal to the

critical bandwidth. The frequency response of each filter is designed to resemble the tuning curve of the auditory nerve fibers centered at its characteristic frequency. The frequency response is characterized by a sharp high frequency slope (at 100-400 dB/oct.), and a flat low frequency slope (at approximately 40 dB/oct) [1,40].

- *Transduction model to represent the saturation of* the *neural firing rate*

  A transduction model, the "hair cell model", is built to simulate the firing saturation and phase lock of the neural fibers. The hair cell model includes the saturation of the firing rate at high signal intensities and the phase lock at a particular point of a vibration cycle. This is simulated by the halfwave rectification and the compression via multi-stage AGC after the cochlear filter outputs.

- *Neural stimulus spectrum representation*

  There are mainly two different models to interpret the transformation of the fiber firings into the neural stimuli. They are based on the rate-place and the temporal-place representation, respectively.

  * *Rate-place representation*

    The model by Goldstein[28] is based on the interpretation of rate-place representation. The place abscissa can be regarded as the frequency abscissa scaled in the critical bands. Since each fiber innervates a single inner hair cell, and each hair cell is sensitive to a motion in a specific portion of the BM, the auditory system is considered to convey stimulus spectral content by the *average firing-rate* in each of the fibers of the auditory nerve.

    A *firing-pattern* should be included in the model[25], because the rate-place representation degrades when the intensity levels of the periodic stimuli increase. The firing pattern can be described as the width of the region in which fibers fire at the same stimulus period. Since more fibers fire synchronously as the stimulus intensity increase, this can be used as a measure of stimulus intensity.

    * *Temporal-place representation*

    Another model is based on the temporal-place representation [51]. It is noticed that neural fibers are capable of representing the temporal properties of the signal. The activities of the fibers are correlated with

the time-varying amplitude components of the signal. It is noticed from the observations that the fibers are phase-locked to the stimulus if the duration of a stimulus is longer than the duration of an action potential of 1 ms; otherwise the fibers are phase-locked to multiples of the stimulus-period.

The temporal-place representation is known to be capable of retaining detailed spectral information for large stimulus amplitudes of the periodic stimuli, however it may not be expected optimal to the unvoiced stimuli.

- *Parallel time-directional processing on neural-stimulus-spectrum*

  Further processing to perceive sound information is supposed to be performed by parallel time-directional processing on the neural-stimulus-spectrum obtained from the above models.

Auditory processing in the high level of human brain

It is still not clear what kind of detailed auditory processing is performed on the information perceived from the neural stimulus spectrum.

It is reasonable to assume that the central auditory processing is performed by **a network with highly recurrent hierarchy**. Thus, sound analysis might be performed in the acoustic, prosodic, phonetic, lexical, grammatical, semantic, linguistic layers with recurrence and constraints, meanwhile using a lot of other auxiliary information such as emotional attitudes and state, class, race, gender, etc. The layer structure of sound analysis is shown in Fig. 4.4.

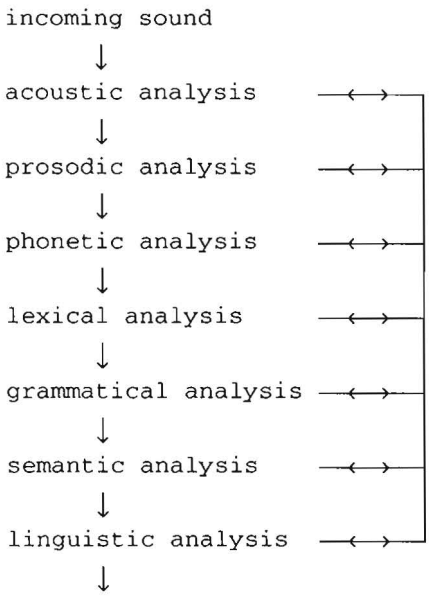


Fig.4.4 A recurrent hierarchy structure for auditory sound processing

Conclusion

In the above overview, it is indicated how the human auditory system collects many pieces of information, and then synthesizes and integrates them to interpret a sound. It is clearly an *adaptive* process: the kind and the amount of information used depends on the complexity of the incoming sound. *Speech enhancement and speech recognition for instance complement each other in a complex situation.*

4.3. Overview of the existing speech enhancement techniques

There are two main tendencies in speech enhancement processing. One concentrated on using different signal processing techniques based on mathematics, another mimicked the auditory micromechanism in order to approach

human processing results. In the following, an overview of the most important speech enhancement/separation approaches will be given.

- *Noise categories*

  Speech enhancement usually infers to target speech enhancement from noise contaminated signals. The noise can be nonstationary or stationary white or colored noise, or interference speech and interference audio sound, etc. Speech separation is one of the speech intelligibility enhancement techniques, which is used for separating the target from the competitive interference speech.

- *Basic difference in processing*

  For different kinds of noise, the speech enhancement techniques explored are usually different. White and colored noise (e.g.: car noise, airplane engine noise, machine gun noise, pop-music), have quite different statistical features and frequency spectra from that of the target speech signals. Thus, it is relatively easy to explore the differences between the signal and the noise. One can use spectral subtraction techniques in order to enhance the target speech.

  For the speech-like noise, there is no basic statistical difference between the target speech and the background speech "noise". Therefore, different approaches must be explored.

  Here we will concentrate on the latter case. For the first case, readers are referred to the references[22,46,69,97,98,102].

According to the basic principles, the existing techniques can be categorized as follows.

(1) Algorithms based on spectral subtraction by harmonic magnitude suppression/selection in the frequency-domain

(1a) Speech separation by exploring the frequency structure of the quasi-periodic voiced-speech signal.

It is noticed that the voiced speech energy concentrates around the frequencies of pitch harmonics. Speech enhancement can therefore be obtained by suppression of the interference speech only at the pitch harmonics when the SNR is negative, or by selection of the target speech

only at the pitch harmonics when the SNR is positive. In the frequency-domain, the magnitude spectrum of each speaker can be approximated by the spectrum of windowed sinusoids.

The idea, originally due to Parsons [71], has been used for selecting the harmonic magnitude spectrum of the target speech, when target and interference speech have about the same intensities.

Hanson and Wang[37] proposed a similar technique for suppressing interference speech when the SNR is negative. In this case, the interference parameters are in general easier to extract than those of the weaker desired-speech. The idea is that the target speech can be estimated by subtracting the estimated interference harmonic magnitude spectrum from the total spectrum. This method has been applied to SNR between -6 and -40 dB.

A post-processing of spectral tailor using the technique of Multi-signal Minimum-Cross-Entropy Spectral Analysis (M-MCESA) has been proposed [12], in order to improve the above results. Given the a-priori autocorrelations of signal R_s and of noise R_n estimated from the separated spectra, a M-MCESA approach estimates the a-posterior spectra of S(t) and n(t) by minimizing their cross-entropy under the constraint of the summed signal autocorrelations R_{s+n}.

The *adaptive comb filter*, which was used in the early 1970's for speech enhancement[23,86], has been found to give no improvement of intelligibility[72]. In a comb filter, only small bands of frequencies which are centered at pitch harmonics can be passed, while those portions of the competitive signal outside the passband of the filter are rejected. Thus, by adaptively controlling the type and size of window functions, the comb filter "enhances" the target speech signals. Perlmutter[72] has proved by experiments that such an adaptive comb filter provides no improvement of intelligibility for the desired speech, despite of using accurate pitch information.

(1b) A sinusodial speech analysis/synthesis model and Least Squares error approach

A sinusodial speech analysis/synthesis model [78,79,80,88], improves the harmonic magnitude suppression/selection approaches. The model can dissolve harmonics from the different speakers, thus can improve the distortion in (1a) when the harmonics of different speakers are too close to separate, or when the stronger harmonics mask the weaker harmonics.
The sinusodial model for N-speakers is defined as

$$s(n) = \sum_{i=1}^{N} \sum_{k=1}^{Mi} a_k^i \cos(\omega_{a,k} n + \phi_{a,k}) \qquad (4.3.1)$$

where a_k^i, $\omega_{a,k}=2\pi f_{a,k}$ and $\phi_{a,k}$ are the amplitude, the frequency and the phase of the k^{th} harmonic of the i^{th} speaker, respectively.
The windowed speech segment in the frequency-domain after Short Time Fourier Transform can be expressed as follows

$$S_w(\omega) = \sum_{i=1}^{N} \sum_{k_i=-M}^{Mi} A_k^i \exp(j\phi_{A,k}) \, W(\omega - k\omega_0^i) \qquad (4.3.2)$$

where A_k^i is amplitude of the k^{th} harmonic of the i^{th} signal, and $W(\omega)$ is the Fourier transform of the analysis window. If the frequencies in (4.3.2) are known a-priori, the Least Square (LS) parameter estimation becomes linear estimation.

(2) Statistical model-based speech enhancement

A Hidden Markov Model (HMM)-based speech enhancement technique for *white* noise has been proposed[19,20,21,22]. In this method, mixed-speech signals are modeled by an HMM which is associated with a random process $z_i=x_i+n_i$, where x_i and n_i are statistically independent and correspond to the speech and the noise respectively. Clean speech is modeled by the mixtures of Gaussian autoregressive (AR) output processes, and the (white) noise is modeled by a process of independent identically distributed (i.i.d.) Gaussian AR vectors. The parameters of speech and noise can be trained separately by the target-absence or the interference-absence segments of the mixed-speech signals. The speech enhancement is then performed under ML/MAP/MMSE criterion. It may be a promising approach to be generalized to the interference speech case.

Another HMM-based approach is proposed by Varga[98]. In this model, the output (observation) probability is the joint distribution from the two models associated with the target and the interference

$$p(\text{observation}) = p(\text{observation} \mid M1 \otimes M2) \tag{4.3.3}$$

where $\otimes$ is the combination operator, M1 and M2 are the random variables from the two models. The decomposition for two simultaneous components becomes

$$p_t(i,j) = \max_{u,v} p_{t-1}(u,v) a1_{u,i} a2_{v,j} b1_i \otimes b2_j(O_t) \tag{4.3.4}$$

where $a1_{u,i}$ (or $a2_{v,j}$) is the transition probability from state u to i (or v to j), $b1_i$ (or $b2_j$) is the output probability emitted from the state i (or j) in model M1 (or M2), $p_t(i,j)$ is a specific output observation sequence produced by the joint model under the constraints that model M1 is in state i and M2 is in state j at time instant t. This approach has only been tested for stationary pink noise and for machine gun noise.

(3) Time-domain LMS adaptive filtering

It has been proved[2] that the adaptive LMS weights of the summed periodic speech signals converge to the same weights of the dominant-only speech, *provided* that the power of the dominant speech is *much higher* than that of the weaker speech.

As it is mentioned in [2], "*The statistical and spectral similarity between the desired signal (main speaker) and the interfering signal (background speakers) often prohibits an improvement using only spectral filtering techniques*".

However, the summed periodic signals simulated are produced by a second-order model only. For speech signals, at least a six-order model (corresponding to three formants) is needed.

(4) Neural network based noise reduction

A noise reduction neural network has been proposed[95,96,97] and simulated for speech contaminated by computer room background noise.

(5) Methods based on monaural auditory sound separation

A computational algorithm for sound separation based on the auditory model

of Licklider[51] has been developed by Weintraub[100,101]. Sound from each cochlear filter output is represented by a group of neural events. An iterative dynamic programming pitch tracking algorithm is applied to determine the pitch period of each of the two sound sources. The number of speakers and the associated periodic/non-periodic characteristics are determined by a Markov model. The spectrum of each sound represented by the events is then iteratively estimated according to the amplitude ratio of the two sounds obtained from the histogram calculations from the trained database of each sound.

(6) *Array processing/beamformer based approach using multi-receivers*

The approach exploits the different time-arrivals and the different intensities of the multi-signals from an array of receivers [18,42,74,90]. By designing a proper array pattern tuning to a desired signal, the target speech signals can be enhanced. This is more close to the concept of binaural auditory sound perception.

Remarks:

(2) and (4) need prior-trained information. In the auditory system processing hierarchy, this processing corresponds to functions performed in the high auditory levels.

Moreover, (2) can also be regarded as a neural network, because an HMM is actually a recurrent neural network[68].

(1) (3) (5) and (6) correspond to the functions performed in the low auditory level.

4.4. Objective and subjective criteria for speech improvement

It is important to measure how much speech improvement can be achieved by a speech enhancement system, in order to compare several different methods and to make a good tradeoff between algorithm complexity and the obtainable subjective speech improvement. Unfortunately, *there is no such a universally applicable single measure available.* This is because the objective distortion measures reflect only partially, in a nonlinear way, our subjective acceptability.

Since speech perception is a highly complex process, it involves not only the

entire grammar and the resulted language structure, but also diverse factors such as semantic context, the speaker's emotional attitude and state, class, race and gender, and the divergence in human sound production organs. *The development of a universally applicable algorithm for prediction of user reactions to speech distortion is still somewhat elusive.* Instead of that, certain classes of objective distortion measures[17] can predict some aspects of speech distortion.

It is worth to mention that for our speech enhancement problem, a *SNR measure can not properly reflect the speech improvement levels to a human listener.* Often, after the processing the SNR is increased but the intelligibility of the speech can be decreased. Thus, what our processing needed is to enhance speech **intelligibly** in order to fit better to the subjective measure of the human auditory system.

4.5. Basis of this speech separation system

In previous sections, the auditory sound perception at different levels has been reviewed. An overview of the most important existing techniques for speech enhancement in the presence of interference speech has also been given.

In this section, we will first analyze some common weak points of these techniques, despite that they are very promising at providing speech enhancement within certain realms. We will then describe some basic thoughts of our speech separation system.

Some remarks on the common weak points of the reviewed techniques

(1) On harmonic-based approaches

It is basically a spectral filtering technique, which is not consistent with the auditory system global processing.

Each frame of transformed data is processed in isolation, without considering the correlation among frames.

It dissolves harmonics from the outputs of a narrowband filterbank. *From the speech intelligibility point of view, it is preferable to choose wideband rather than narrowband filters*[81]. In the narrowband filter case, the reverberation distortion is caused by lengthening the effective time-window duration of the filters. While in the wideband filter case,

the time-dimension aliasing distortion can be introduced by the down sampling in the channels. Although the narrowband and the wideband filters both introduce distortion, the perceptual degradation caused by reverberation-distortion in narrowband filtering is more severe than by the time-aliasing distortion from the wideband filtering, because of the severe damage of formant trajectories in the narrowband filtering.

(2) *On the statistical model-based approaches*

If one desires to generalize the statistical model-based approaches to enhance speech signals from the background speech noise, one probably needs to include the pitch information. Existing models only contain the vocal-tract parameters.

(3) *On the time-domain LMS algorithm based-speech enhancement*

This approach is limited to high SNR, which makes it less attractive. Moreover, because of the large dynamic range of speech spectra, the local SNR can not be always consistent with the global SNR constraint. Thus, the convergence property may not be consistent in different frequency areas. There may even be some divergence areas. Consequently, the algorithm may have difficulties for speech enhancement under high SNR constraint.

The structure of the speech intelligibility enhancement presented in this thesis

In order to form the structure of this speech separation system, we will exploit:

*The time-frequency bin domain filtering of speech signals;

*The information existing in the local speech signals;

*The inconsistency of global and local signal dominance;

*The wide-bandpass filtered speech having less subjective distortion.

Some global and basic properties of this system

1) The processing approach must exploit *local differences* in the target and the interference signals, including the local pitch and the local spectral differences, as much as possible.

2) The adaptive filtering must be explored in the time-transform domain rather than in a single dimension such as the time-domain or the transform-domain,

due to the speech signal nonstationarity.

3) The method has to be expressed mathematically and to be implemented in a simple algorithm, effective for speech enhancement.

4) The algorithm must be able to divide a complicated speech separation problem into a group of monotonic simple problems.

5) The algorithm must be consistent with the auditory system global processing but avoiding to mimic the auditory micromechanism.

6) The algorithm must be flexible in adaptation to the variable complexity of the noisy speech signals.

7) This algorithm should use no a-priori speech information of either a specific target or a specific interference.

4.6. Limitations of this speech separation system

As mentioned previously, the auditory system uses many pieces of information and contains many processing steps above the acoustic level. It must be pointed out that from the theoretical point of view, this speech separation system can not produce complete and perfect separation results, instead it can produce a separated target speech on which the ear can have a better intelligibility acceptance. This is because of our limitation to single-input co-channel signals (which corresponds to monaural sound), and because of the limitation caused by the acoustic level signal processing, without using linguistic information and combining the speech separation process with the speech recognition.

It should be mentioned that from information theory, it follows that information is lost in a co-channel. Thus one can not expect to restore and recover the target signal completely.

4.7. Fundamentals for single-input frequency-bin time-directional processing

The speech separation system described here is based on the parallel time-directional adaptive processing of the decomposed signals. Particular emphasis is put on the consistency to human auditory global processing without emulating the detailed auditory behavior.

Such a signal processing technique is applicable to speech separation due to the following fundamental understandings of the human auditory system and the following fundamental properties of the speech signals.

Consider the following characteristics of the auditory system:

1) The human auditory system performs some running short-time spectral analysis on the acoustic waveforms, by decomposing signals into isolated frequency components. Further processing is done essentially along the time axis[56,64].

2) For a human listener, a better subjective sound quality is obtained by a synthesized speech from a wideband filterbank rather than from a narrowband filterbank[81].

Consider further the following fundamental properties of the speech signals:

1) It has been noticed that the target and the interference speech signals can dominate differently in the various frequency bands. This is because the speech spectrum has a large dynamic range. It contains many peaks (formants), depending on the vocal-tract shape of a specific sound. The globally stronger signal can be the weaker one in some frequency regions. In general, the target and the interference signals can dominate differently in the different frequency bins, as shown schematically in Fig.4.5.

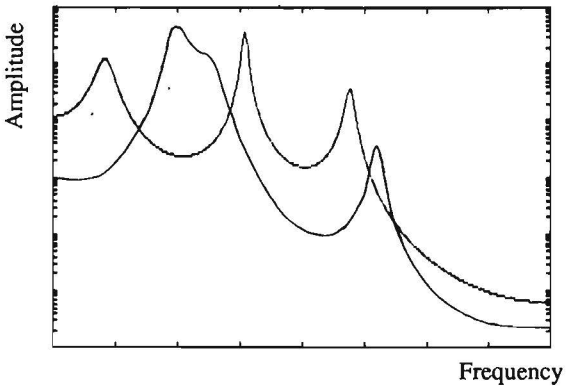


Fig.4.5 Two speech spectra dominating in different frequency regions

2) In each frequency-bin speech signals can preserve high time-resolution, if they are properly bandpass filtered. Consequently, voiced-speech signal components contain periodicities along the time-direction associated with both the target and the interference speaker.

The above mentioned properties enable a speech enhancement algorithm to perform frequency-bin time-directional adaptive processing. This implies that the speech separation problem can be divided into a group of monotonic sub-problems, by splitting speech signals into many nearly-independent bin-components evolving with time. Each bin only contains one monotonic dominant speaker.

4.8. General system description

As shown in Fig.4.6, the speech separation system consists mainly of five parts:

1) A wide bandpass filterbank splits the co-channel signal into frequency bins.

2) A robust pitch estimation algorithm is designed for simultaneously multi-pitch contour estimation.

3) The stronger/weaker speaker is identified for each bin by the estimated short-time local TIR using higher-order moments.

4) A time-frequency bin domain adaptive filtering algorithm is applied for speech separation.

5) The separated signals associated with each speaker are summed over all frequency bins.

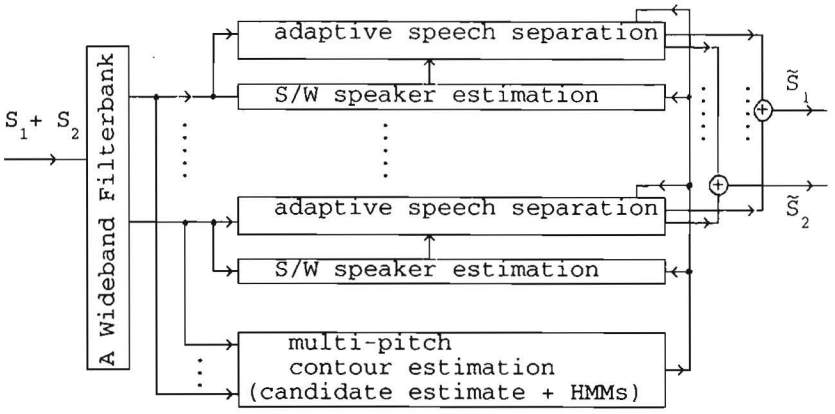


Fig.4.6 Block diagram of the speech separation system

In the following sections, the detailed system implementation will be described.

4.8.1. Signal decomposition

For speech separation purpose, the co-channel speech signal is split into frequency bands by a group of *wide-bandpass filters*. Each bandpass filter has a frequency response $H_k(\omega)$ symmetric to its center frequency $\omega_k=2\pi k/N$, k=1..N, with a bandwidth 2B. For the "noisy" co-channel speech signal, more frequency bands should be chosen in order to keep a certain redundancy.

In order to simplify calculation, we implement these bandpass filters by a Short-Time Fourier Transform (STFT), followed by shifting the output data into the corresponding frequency bands, and taking the real-part of the data.

STFT can be explained in terms of a bandpass filterbank with uniformly spaced center frequencies from 0 to π. Given a signal $\{x(t)\}$, the STFT is defined as

$$\tilde{X}(n,\omega_k) = e^{-j\omega_k n}\left(\sum_m x(n+m)w(-m)\, e^{-j\omega_k m} \right) \qquad (4.8.1)$$

where w(m) is a symmetric window of size L. The STFT output $\tilde{X}(n,\omega_k)$ is corresponding to the lowpass-shifted bandpass filter output.
The above formula can be re-written as follows

$$\tilde{X}(n,\omega_k) = e^{-j\omega_k n}\, X(n,\omega_k) \qquad (4.8.2)$$

where $X(n,\omega_k)$

$$X(n,\omega_k) = \sum_m x(n+m)w(-m)\, e^{-j\omega_k m} \qquad k=1..N \geq L \qquad (4.8.3)$$

is the output of a complex-valued bandpass filter which fits the above bandpass filterbank demand. Thus, the corresponding filter impulse response of the complex filter in the STFT is

$$h_k(n) = w(n)\, e^{j\omega_k n} \qquad (4.8.4)$$

*** Time-resolution and frequency-resolution are limited by the uncertainty principle**

It is well known, that the STFT suffers from the time-frequency resolution limitation. One can not make arbitrarily high time-resolution and high frequency-resolution simultaneously. The time-frequency resolution is governed by the *uncertainty principle*[14], i.e. $\Delta_t \Delta_f \geq 1/4\pi$, where Δ_t and Δ_f are the bandwidths of the time-window and the corresponding frequency window, respectively (the equality holds if and only if the time-window w is Gaussian).
However, by selecting the size of a time-window comparable to the average pitch period, speech spectrogram[77], which contains pitch periodicities and

formants trajectories, will satisfy the time-resolution demands for later processing.

One often likes to have different time and frequency resolution in different frequency bands by using a different analysis window size. This is referred to the "*zoom in and zoom out*" function. In high frequency bands, a higher time-resolution is needed than in the low frequency bands. Such a bandpass filterbank can be implement by using orthogonal **Wavelet Transform** (WT) [14,15,83]. Although it still suffers the time-frequency resolution limitation, the time-resolution and the frequency-resolution no longer have to be the same for every frequency bin. In WT, bandwidths of different bins are governed by a scale factor. The bandwidths can thus be chosen uniformly distributed on a logarithmic scale. From the physiology point of view, it is more suitable to decompose signal into frequency channels with a same logarithmically scaled bandwidth. This is easily understood that a large data window is needed in a low frequency band in order to have a relatively high frequency-resolution, and a small one in a high frequency band.

Although as mentioned above that filterbank obtained by WT is more attractive than that of the STFT, STFT type filterbank is still used in our simulations due to limitation of this research. We believe that the speech enhancement system depends mainly on the separation algorithm itself. However, one can always revise the filterbank implementation by WT in order to gain more benefit.

4.8.2. Estimation of pitch

For the pitch estimation part in this speech separation system, the algorithm described in chapter 3 can be applied directly for pitch contour estimation from co-channel speech signals.

4.8.3. Estimation of short-time local TIR

In each bin a short-time bin Target-Interference Energy Ratio (TIR) is roughly estimated in order to decide which voiced-speaker is dominant.

First, the bin signal X(t,k) is half-wave rectified to $\bar{X}(t,k)$ and the third order moment with the delay of each speaker's pitch period P_m is calculated as follows

$$Z_m(t,k) = \bar{X}(t,k)\bar{X}(t-P_m,k)\bar{X}(t-2P_m,k) \tag{4.8.5}$$

where P_m is provided by the pitch estimation algorithm. The short-time TIR in each frequency bin is then estimated by the energy ratio (in dB)

$$TIR(t_0,k) = 10/3 \log\left(\sum_t Z_T^2(t,k) \,/\, \sum_t Z_I^2(t,k) \right) \tag{4.8.6}$$

where $t_0 \in \{ t \}$.

4.8.4. Adaptive speech separation

After bandpass filtering, the noisy speech signal components are processed by using a time-frequency bin (T-TB) domain adaptive noise canceler.

Before describing the speech separation algorithm, we will first focus on the following two questions:

* Can a T-FB adaptive filter be used for speech separation?
* Does this algorithm converge to the target (interference) speech signal?

Consider the situation where the co-channel signal is composed of one target and one interference speaker in a voiced-voiced or voiced-unvoiced situation. The bandpass filtered signals at frequency bin k can be expressed as

$$X(t,k) = S_T(t,k) + S_I(t,k) \tag{4.8.7}$$

For a voiced-speech signal, it is highly correlated among the successive periods. Supposing the target speech signals S_T and the interference speech signals S_I are statistically uncorrelated, the autocorrelation function of the bandpass signal with its p_T delayed version (p_T is the target speech pitch period) at bin k can be expressed as below

$$E(X(t,k)\ X(t-p_T-j,k)) = R_{S_T}(j,k) + R_{S_I}(p_T+j,k) \tag{4.8.8}$$

where the relation $S_T(t-p_T-j) = S_T(t-j)$ has been used, i.e., it has been supposed that the target voiced-speech signal is stationary in the time interval of consideration. If the target speaker is dominant at this bin, the

first term in (4.8.8) becomes the main part. Here it is also supposed that the same speaker is dominant during the short-time interval. A similar analysis holds for the interference speech with pitch period P_I.

From the above, we would guess that a LMS filter weights in a single frequency bin would converge to the filter weights associated with a dominant speaker.

On the convergence property of a T-FB domain LMS algorithm

*** Convergence property for full-band signals**

For the summed (fullband) periodic speech signals under the constraint of a *dominant* speaker plus a weaker interference speaker, it has been proved[2] that the adaptive LMS weights initially converge to the same weights as would be produced by the dominant speaker-only case.

However, because of the large eigenvalue spread of the speech signals, it would be very difficult to keep this constraint consistently over the whole signal spectrum. Consequently, the algorithm is limited to have applications in most practical speech situations.

*** Convergence property in the T-FB domain**

However, we can extend the above convergence property to the T-FB domain LMS algorithm. This results in a part of the foundation for the T-FB domain speech separation.

The above convergence property implies that the adaptive filter of each frequency-bin will converge to the dominant speech signal components in a T-FB domain LMS algorithm.

Hence, we can group the fullband signal into the components in different regions belonging to these three categories: the bins dominated by the target-speaker; those dominated by the interference-speaker; and the bins where the two speakers have comparable signal energies.

Thus, for those bins dominated by the target (interference) speaker, the corresponding LMS weights fast converge to the target (interference) speaker. While in those bins where the energy difference is small, the LMS algorithm shows a poor convergence.

Possibility of speech separation using a T-FB domain LMS algorithm

Based on the above analysis, a speech separation algorithm can be formed by using a T-FB domain LMS algorithm.

A T-FB domain LMS algorithm, functioning as a T-FB domain Adaptive Noise Canceler (ANC), can be used to estimate the signal components of the stronger-speaker at different bins. The filter output residuals can be regarded as the estimated signal components of the weaker speaker.

In order to obtain the separated speech signals, we can simply sum, for each time instant, the separated signal components associated with a desired speaker over all bins.

It should be mentioned that applying an LMS algorithm in those bins having close local Target-Interference Energy can produce relatively large errors. Those bins are the main parts of distortion introduced to the separated signals.

In the following, two different speech separation approaches will be investigated. We will first describe the T-FB domain speech separation associated with a *linear* adaptive filtering approach in section 4.9, and then replace this linear adaptive filter with a *nonlinear* one, in section 4.10, in order to obtain further improvement.

4.9. Speech enhancement via the time-frequency bin domain *linear* NLMS adaptive filtering

We will first explore the possibility of speech separation via a T-FB domain linear LMS adaptive filtering algorithm[30].

In such a case, a T-FB domain linear LMS adaptive filter is used as a linear Adaptive Noise Canceler (L-ANC). This L-ANC extracts the stronger voiced-speech signals at *each bin*, by using the quasi-periodic correlations of the voiced-speech signals. In particular, we consider the algorithm under a semi-ideal transform assumption as defined in chapter 2, where all bins are linearly independent. Thus, the T-FB domain filter is simplified to a group of time-directional filters of independent bins.

This speech separation algorithm consists of the following three steps:

1) Determination of the dominant speaker at each bin

In each analysis frame, decisions on which speaker is locally dominant, and whether or not this speaker's signal is predominant are made at each bin, by using the estimated short-time local TIR.

2) Estimation of the stronger speaker's signal (or the voiced-speaker's signal in Voiced-UnVoiced case)

An LMS algorithm is applied to estimate the stronger speech signals in the V-V case (or the voiced-speech signals in a V-UV case) at each bin by using the periodic correlations. The output of the adaptive filter

$$y(t,k) = \sum_{j=0}^{M_i - 1} W_j^{(i)}(t,k) \quad X(t-P_s-j,\ k) \tag{4.9.1}$$

is assigned to the stronger speaker (or to the voiced-speaker in a V-UV case) as the separated signal component. The filter weights at each bin are updated at every time instant as follows

$$W_j^{(i)}(t+1,k) = W_j^{(i)}(t,k) + 2\ \mu_k e(t,k)\ X(t-P_s-j,\ k) \quad j=0..(M_i-1) \tag{4.9.2}$$

where $W_j^{(i)}(t+1,k)$ are the filter weights; M_i is the filter order associated with the speaker i, (i=1,2) at bin k; $\mu_k = \dfrac{\mu 0_k^{(i)}}{M_i E[X^2(t,k)]}$ is the normalized filter step-size at bin k; $\mu 0_k^{(i)}$ is a positive small constant controlling the convergence speed and the filter steady state performance.

3) Estimation of weaker speaker's signal

If the stronger speaker's signal is not predominant, the residual of the L-ANC

$$e(t,k) = X(t,k) - y(t,k) \tag{4.9.3}$$

is assigned to the weaker speaker as the separated signal, otherwise zero value is assigned.

As seen from the above, the voiced-speech signal is first extracted if the co-channel signal consists of one Voiced and one Unvoiced (V-UV) speech samples.

Fig.4.7 shows one of such L-ANCs, where the stronger voiced-speech signal in the k^{th} bin is supposed to be $S_1(t,k)$ with pitch period p_1

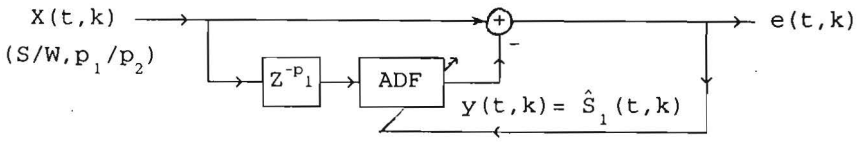


Fig.4.7 Linear Adaptive Noise Canceler at frequency-bin k

The program flowchart in Fig.4.8 summarizes this algorithm.

Remarks:

1) Distortion in the separated results

The difference of local signal energy and of the pitch periods between the two speakers are exploited by this algorithm. The larger these differences are, the better speech separation can be expected. The overall quality of the separated speech is ensured by the collective performance of all bins.

2) Selection of adaptive filter step-size constant $\mu 0_k^{(i)}$

In the separation algorithm, one should pay particular attention to the selection of $\mu 0_k^{(i)}$ to ensure that no divergence occurs at any bin. Otherwise, the signals in the diverged bin might cause a peak in the summed separated speech signals. The peak might be much higher than the normal recovered signal amplitude, thus would damage the overall results.

This algorithm has been applied for co-channel speech separation with promising results. The details of the simulations and some results will be described in section 4.11.

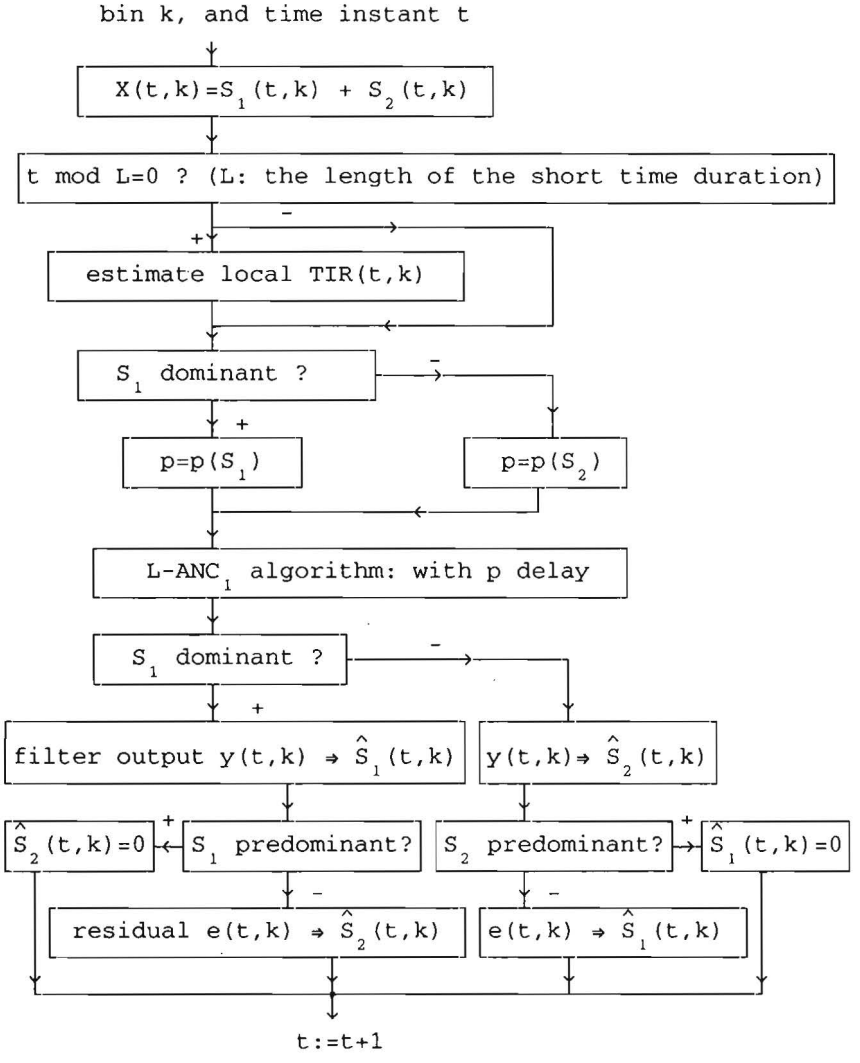


Fig.4.8 Program flowchart of speech separation using *linear*-ANC

4.10. Speech enhancement via the T-FB domain *nonlinear* NLMS ADF

Although the approach described in previous section is promising, some problems still remained. The simulations show that, after separation, some audible background interference sound still exists. The interference sound is however no longer understandable.

In order to further eliminate the background speech noise while maintaining reasonably good intelligibility and limited target speech distortion, speech separation via the T-FB domain Nonlinear Normalized LMS adaptive filtering has been investigated[31].

*** The basis for nonlinear processing**

1) Although linear predictions can be used for speech signal estimation from a single speaker based on an AR (AutoRegressive) model, this model is no longer suitable for the summed speech signals. The summed signals, which can be described by a system with parallel AR models, are in general nonlinear. The following formula describes an ARMA model produced by two parallel AR models.

$$\frac{u_0}{1+\sum_i a_i z^{-i}} + \frac{v_0}{1+\sum_j b_j z^{-j}} = \frac{d_0+\sum_l d_l z^{-l}}{1+\sum_k c_k z^{-k}} \qquad (4.10.1)$$

2) A more accurate speech terminal model is associated with a NL one. Although in the simplified case, an AR model is usually used for the ideal-lossless vocal-tract, evidence shows that for the nasal and the fricative sounds, a NL model should be considered[81]. Furthermore, by including the radiation at the lips, the actual vocal-cord excitations and the loss of the vocal-tract, the speech model is the series connection of the three parts R(z), G(z) and V(z) as shown in Fig.4.9, which is *nonlinear*.

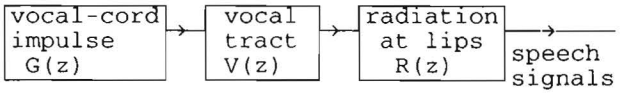


Fig. 4.9 Nonlinear terminal-model of speech signals

*** Possible benefit from the *nonlinear* processing**

As we will see, this nonlinear speech separation algorithm is associated with a T-FB domain NL-ANC, which is a direct application of the algorithm developed in section 2.3. This NL-ANC is associated with a second-order Volterra filter. In this case, the bandpass filtered signal and its delayed version can be used as the primary and the reference input. Again, the algorithm is considered under a semi-ideal transform assumption, where all the quadratic filter coefficients associated with different bin-pairs are mutually independent. Because the linear filter part is decoupled from the NL part as mentioned in section 2.3, we will therefore only discuss the new benefit introduced by the NL part processing.

By using the existing *nonlinear* correlations, some benefit can then be introduced. The following products of signal components (nonlinear) can be introduced for the purpose of speech enhancement:

1) Using signal components from other different bins

Because there exists nonlinear cross-correlations both along the bin and the time directions, signal components of bin i can be estimated from the weighted sum of signal component products from bin j and k. Suppose the local TIR at the i^{th} bin is not favorable to the concerned speaker (e.g.: the local TIR $\approx$ 0dB, or a local weaker speaker under another predominant one), using linear estimation can introduce relatively large distortion in this case. However, one can select some nonlinear terms from other bin-pairs where the local TIR values are favorable to the concerned speaker. The products of two signal components from a bin-pair (j,k), j+k=i, can be selected for the concerned-signal estimation at bin i. This means that part of the quadratic terms in the block-matrix $\mathbf{H2}_{j,k}$ in formula (2.3.55) can be selected for estimating the signal components of the concerned speaker, i.e., **part** of the following terms can be selected

$$y^{(2)}_{n,i} = \sum_{j,k,l,m} H2_{j,k}(l,m)\,(X_{n-l,j}X_{n-m,k} - R_{j,k}(l-m))$$

$$0 \le l,m \le (M-1),\quad 0 \le j,k,i \le (N-1) \text{ and } (j+k)=i \qquad (4.10.2)$$

where M is the time-directional quadratic filter order associated with the concerned speaker, N is the total bin number, $R_{j,k}(l-m) = E(X_{n-l,j}X_{n-m,k})$

is the estimated signal crosscorrelation function of bin j and bin k with lag (l-m).

2)Using the products of two signal components of a same bin but at different time instant.

In this case, the nonlinear signal time-autocorrelations and the periodicity of voiced-speech signals are exploited. For a co-channel signal $X_{n,i}=S^{(1)}_{n,i} + S^{(2)}_{n,i}$, with pitch periods p_1 and p_2 respectively, the p_i (i=1 or 2) delayed nonlinear products can be used to enhance the signal component estimation of a concerned speaker.

When the local TIR at the i^{th} bin is not favorable to the concerned speaker, one can profitably use the weighted sum of signal component products in another bin j (with favorable local TIR), having a time-delay equal to the concerned speaker's pitch-period p_i, i.e, by selecting **part** of the following terms

$$y^{(2)}_{n,i} = \sum_{\substack{0\le l,\, m\le(M-1) \\ |l-m|=p_i}} H2_{j,j}(l,m)(X_{n-l,j}X_{n-m,j} - R_{j,j}(l-m))$$

$$j=i/2,\ 0\le i,j\le(N-1) \qquad (4.10.3)$$

where $R_{j,j}(l-m)= E(X_{n-l,j}X_{n-m,j})$ is the estimated signal autocorrelation function of bin j with the lag (l-m).

When the local TIR of bin i is weak positive with respect to the concerned speaker, one can enhance the desired signal component estimation by selecting part of the following terms

$$y^{(2)}_{n,i} = \sum_{\substack{0\le l,\, m\le(M-1) \\ |l-m|=p_i}} H2_{i,i}(l,m)\sqrt{|X_{n-l,i}\, X_{n-m,i}|} \qquad (4.10.4)$$

*** Description of the algorithm**

The speech separation using the NL-LMS ADF algorithm consists of two steps. First the stronger speaker's signal will be estimated; next the weaker speaker's signals. Both steps will be described below.

1)Estimation of the stronger speaker's signal

The stronger speaker's signal is estimated at each bin by using a similar method as in the L-ANC case. The only difference is that some quadratic terms are introduced.

The quadratic terms { $\sqrt{|X_{n-m_1,i} X_{n-m_2,i}|}$ (when the local TIR(t,i) is weak positive), $X_{n-m_1,i/2} X_{n-m_2,i/2}$ (when the local TIR(t,i/2) is favorable) | $0 \le m_1, m_2 \le (M-1)$, $|m_1 - m_2| = p_s$, with p_s the pitch period of the stronger speaker of the i^{th} bin } can be selected. The quadratic terms $\{X_{n-l,j} X_{n-m,k}$ | $j+k=i$ and $0 \le j,k \le (N-1)$, $0 \le l,m \le (M-1)$ } from the bin-pair (j,k) can also be chosen to estimate the signal components of the stronger speaker at bin i, where the local TIR(t,j) and TIR(t,k) are favorable to the stronger speaker of the i^{th} bin.

This is expressed more precisely in formula (4.10.5) below. Suppose the stronger speaker at the i^{th} bin is speaker A. The signal components of this stronger speaker A are estimated by an NL-ANC$_1$ which may contain **part** of the following terms

$$\hat{y}_{n,i} = \sum_{m=0}^{M_A - 1} H1^{(A)}_{m,i} X_{n-m-p_s,i} + \sum_{0 \le m \le (M_A - 1)} H2^{(A)}_{i,i}(m, m-p_s) \sqrt{|X_{n-m,i} X_{n-m-p_s,i}|} +$$

$$+ \sum_{0 \le m \le (M_A - 1)} H2^{(A)}_{i/2,i/2}(m, m-p_s)(X_{n-m,i/2} X_{n-m-p_s,i/2} - R_{i/2,i/2}(p_s)) +$$

$$+ \sum_{\substack{0 \le l,m \le (M_A - 1) \\ j+k=i \ \& \ 0 \le j,k \le (N-1)}} H2^{(A)}_{j,k}(l,m)(X_{n-l,j} X_{n-m,k} - R_{j,k}(l-m)) \qquad (4.10.5)$$

where $X_{n,i} = S^{(A)}_{n,i} + S^{(B)}_{n,i}$ is the noisy signal at bin i, p_s is the pitch period of the stronger speaker at bin i in the present short-time duration, $H1^{(A)}_{m,i}$ and $H2^{(A)}_{j,k}(l,m)$ are the linear and quadratic filter coefficients of speaker A associated with NL-ANC$_1$, M_A is the filter order along the time direction for speaker A .

In the situation where one speaker's signal is predominant, the bin signal is assigned to the relevant speaker.

2)Estimation of the weaker speaker's signal

Suppose the weaker speaker at the i^{th} bin is B. Instead of assigning

residuals to the weaker speaker B in the L-ANC case, an NL-ANC$_2$ is used. In the NL-ANC$_2$, only the quadratic terms are chosen in order to reduce the disturbance caused by the residual of the stronger speaker signal (This residual sometimes can be relatively large). Thus an NL-ANC$_2$ may contain **part** of the following items

$$\hat{Z}_{n,i} = \sum_{0\le m\le (M_B-1)} H2^{(B)}_{i,i}(m,m-p_w)\{e1_{n-m,i}\,e1_{n-m-p_w,i}\} +$$

$$+ \sum_{0\le m\le (M_B-1)} H2^{(B)}_{i/2,i/2}(m,m-p_w)(X_{n-m,i/2}X_{n-m-p_w,i/2} - R_{i/2,i/2}(p_w)) +$$

$$+ \sum_{\substack{0\le l,m\le (M_B-1)\\ j+k=i \;\&\; 1\le j,k\le N}} H2^{(B)}_{j,k}(l,m)(X_{n-l,j}X_{n-m,k} - R_{j,k}(l-m)) \qquad (4.10.6)$$

where $e1_{n,i}$ are the residuals of NL-ANC$_1$, $H2^{(B)}_{j,k}$ are the quadratic filter coefficients of an NL-ANC$_2$ associated with weaker speaker B, M_B is the filter order along the time direction for speaker B, P_w is the pitch period of the weaker speaker.

The process of speech separation by the NL approach at arbitrary time instant t and bin k can be expressed by the flowchart in Fig.4.10.

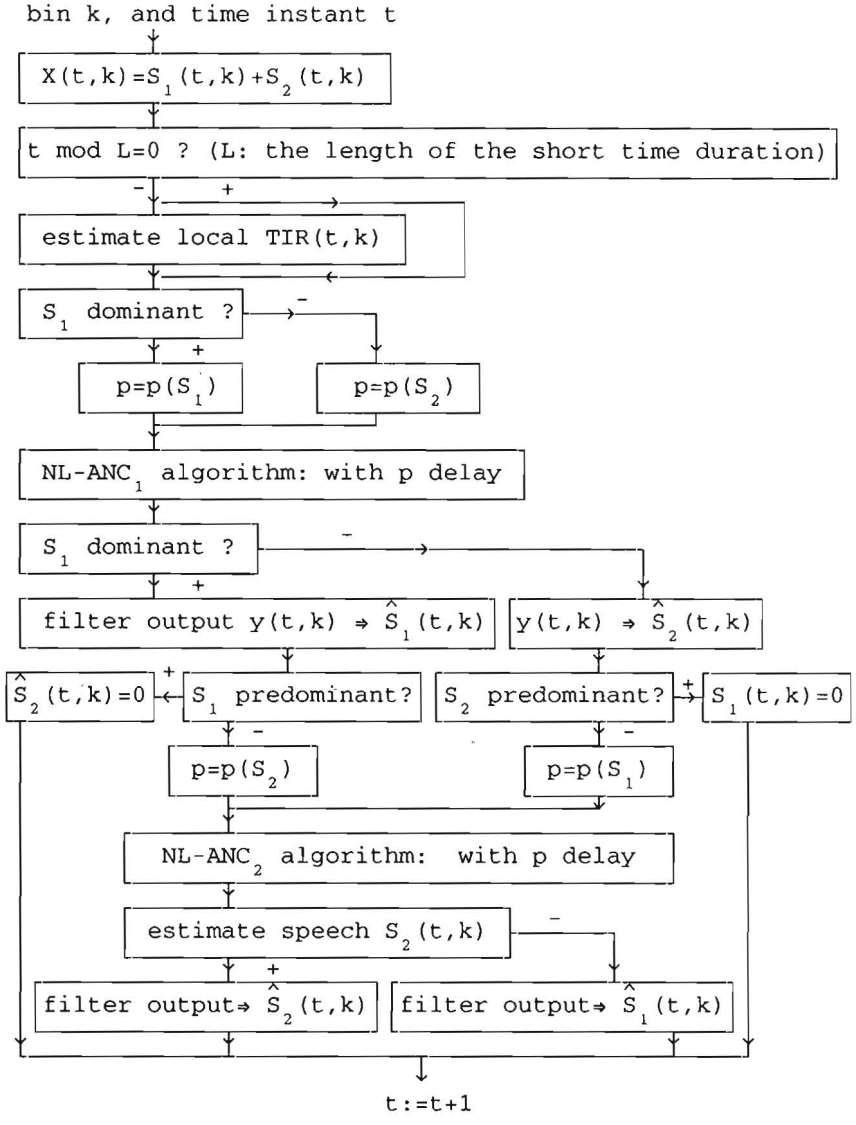


Fig.4.10 Speech separation via the T-FB domain NL-ANC

Simulations performed by using this NL-ANC show a further improvement over the linear approach. Details of the simulations are included in the following section.

4.11. Simulations and results

The simulations, including separation of stationary synthetic speech signals, nonstationary synthetic speech sentences with constant and natural pitch via linear and nonlinear approaches, will be described in detail. Some results are included.

Simulations are performed to test if the algorithms are practically applicable and effective for speech separation.

Available information for the speech separation

In all these simulations, only the summed-signals are available, this is equivalent to a *single receiver* case. The co-channel signals are created by adding two different speech signals with properly selected Target-Interference Energy Ratio (TIR). There is no a-priori information about the target and the interference signals.

Common processing part shared by the different kinds of simulations

The first step processing is same for all kinds of simulations, i.e., co-channel speech signals are split into frequency bins through a wide bandpass-filterbank. At each frequency bin, the local stronger/ weaker speaker is estimated in every frame.

Pitch estimation

For simultaneously estimating the two pitch values in summed stationary speech signals (or nonstationary speech sentences) with constant pitch values (or constant pitch contours), only the first part of the algorithm in chapter 3 is needed.

For estimating naturally changing pitch contours in summed-sentences, the V-UV frames need to be segmented first. This is because the HMM-based pitch contour estimation algorithm in chapter 3 does not yet include the unvoiced situation. The voiced and unvoiced segments can be decided by other V-UV

detection algorithms, such as the signal-energy and zero-crossings based V-UV detection approach[81]. After the segmentation, the algorithm described in chapter 3 can be used to estimate pitch contours of the two speakers.
However, in order to evaluate the adaptive speech separation part and the pitch estimation part separately, the pitch contours estimated from single speech sentences will be provided to the speech separation part temporally.

The three parts of the simulations
In the first part, the speech separation algorithm is applied to stationary synthetic speech with constant pitch period, in order to observe the separation results after a sufficient number of iterations.

In the second part, synthetic speech sentences with constant pitches are used for speech separation, by the *linear* and *nonlinear* approaches.
It is a relatively simple situation compared to the natural speech sentences having slowly changing pitch frequencies. The purpose of simulations in this part is three-fold. First, to test if the convergence of the algorithm is fast enough in adaptation to the time-varying vocal-tract parameters of speech sentences. Second, to find out if the quality of separated results is acceptable. Third, to see if the quality of the separated speech sentences has any apparent difference among the linear and nonlinear approaches.

The third part is associated with the preliminary tests on separation of synthetic sentences with naturally changing pitch frequencies. In these tests, the co-channel speech signals are obtained as the sum of the two synthetic speech sentences, which are produced by separately using an LPC synthesizer with the estimated single speaker's pitch contour as excitations. The pitch contours which are provided to the speech separation system are replaced by their estimated values from the single-speaker's signals, in these tests in order to distinguish the distortion introduced by the pitch estimation part and the speech separation part.

(a) Simulations on separating *stationary* synthetic voiced-speech signals

(a1)via T-FB domain linear LMS adaptive noise canceler

In these simulations, we will separate summed *stationary* synthetic speech

signals, in order to check if the algorithm works effectively.

Two stationary synthetic speech signals, each containing three formants with a specific pitch, are added with properly chosen TIR between 0 dB and ±12 dB. The only signal available to the separation algorithm is the summed-signal. The pitch information is obtained from the pitch estimation algorithm.

In the simulations, the synthetic speech signal is produced by passing periodic excitations to a filter with three given formant frequencies and bandwidths. The speech signals produced are then stored as a speech file. In each simulation, two synthetic speech signals are added by a pre-selected TIR value. Before taking the STFT, each frame of speech signals is Hamming windowed with the selected size L=80. Since the fullband speech signal bandwidth of the stationary signals is selected between 0-4 KHz, and the sampling frequency is f_s= 8 KHz, the equivalent bandwidth of each bandpass filter equals $2B=4f_s/L=400$ Hz. The total filter number is chosen N=100 to keep enough redundancy. The bandpass filters are uniformly distributed. The outputs of the bandpass filters are obtained by taking the real part of the data after the DSTFT and the bandpass shifting. The same step-size constant $\mu 0=\mu 0_i=0.1$ is chosen for all bins in the adaptive filter.

In the following, several separated results are included. Table 4.1 lists the parameters (the pitch values, the formant frequencies f_i and the associated bandwidth B_i, i=1..3) of the synthetic stationary speech signal of a single speaker. These parameters will be needed as references in the following figures. Fig.4.11. and Fig.4.12. show the speech waveforms and the LPC spectra from the original, the summed and the separated speech signals, respectively. The TIR is selected 0dB and -12dB, respectively.

synthetic	formant 1		formant 2		formant 3		pitch
speech	f_1(Hz)	B_1(Hz)	f_2	B_2	f_3	B_3	(samples)
S_1	730	50	1090	75	2440	100	40
S_2	270	50	2290	75	3010	100	47
S_3	420	50	1550	75	2400	100	47

Table 4.1 Parameters of the synthetic *stationary* voiced-speech signals

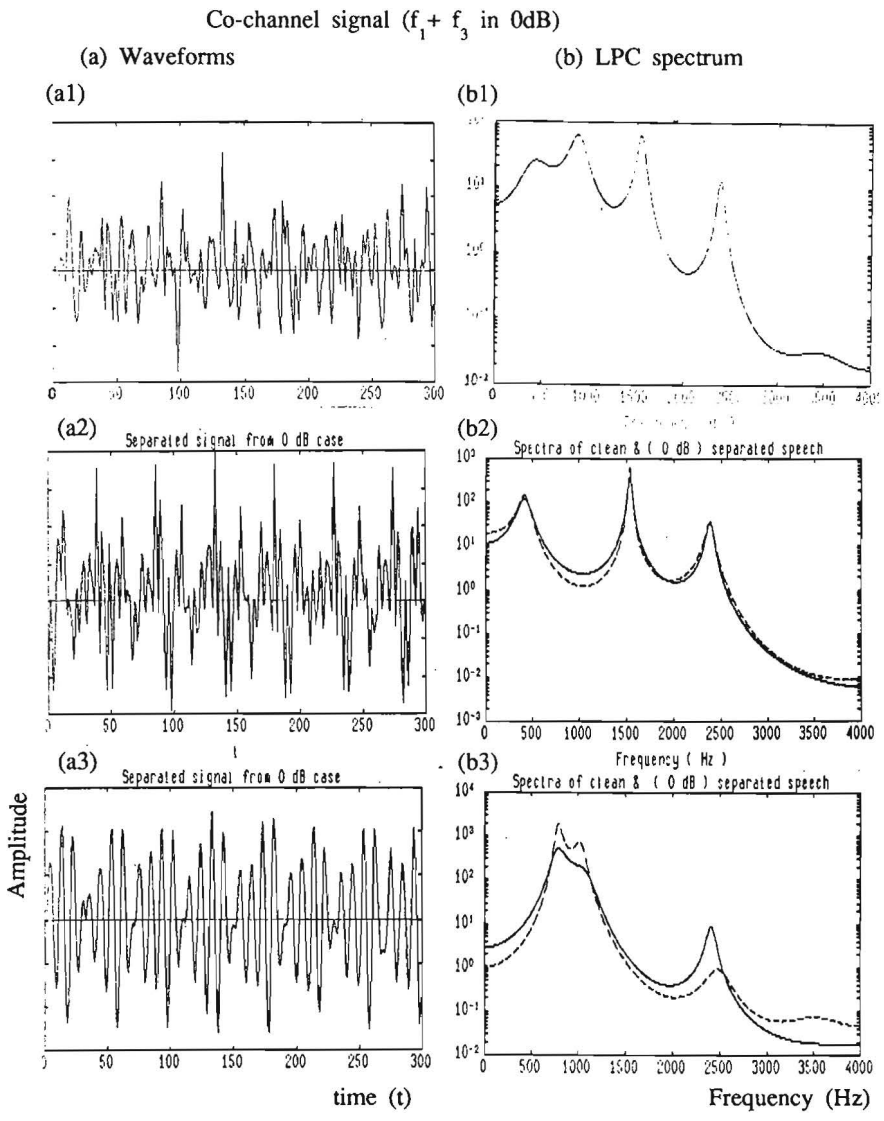


Fig.4.11(1) Separation of summed stationary speech signals

by the T-FB domain *linear*-ANC (TIR = 0dB)

(a1) (b1) co-channel speech signal

(a2) (b2) separated speech signal f_3

(a3) (b3) separated speech signal f_1

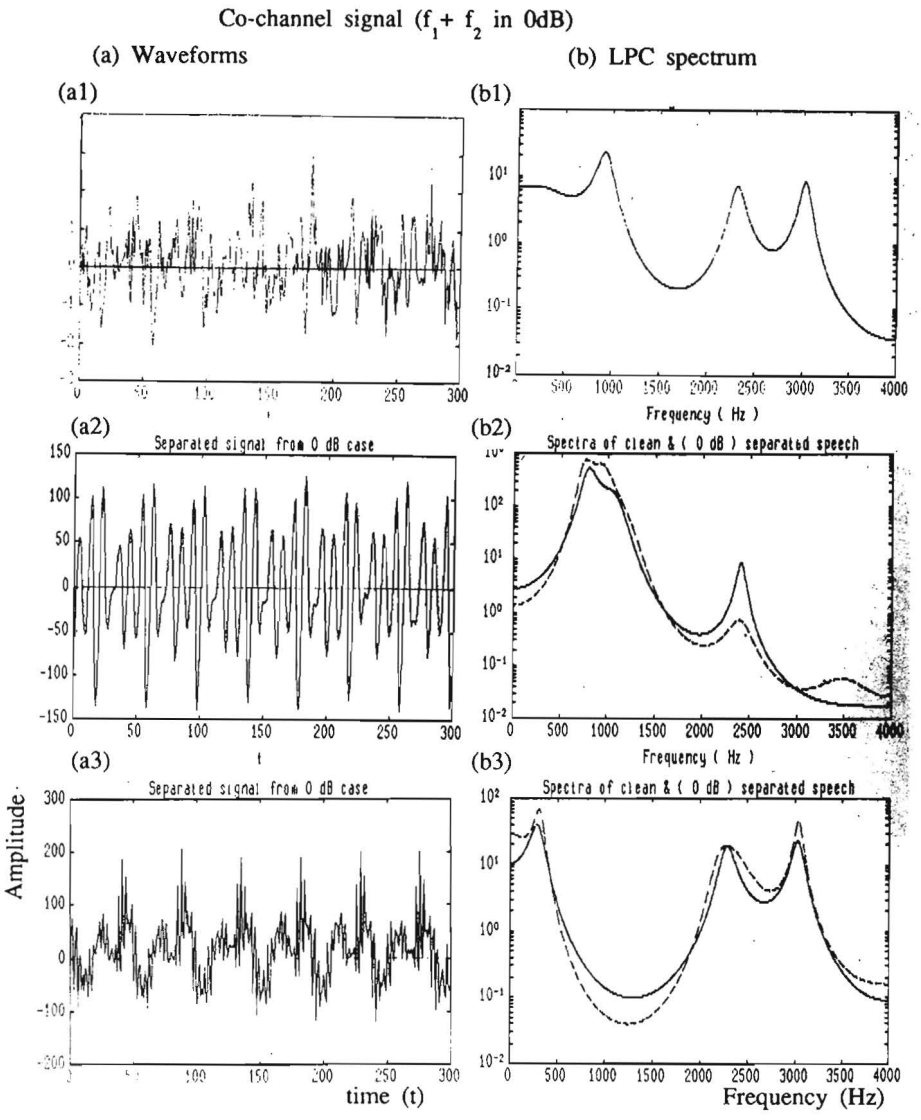


Fig.4.11(2) Separation of summed stationary speech signals

by the T-FB domain *linear*-ANC (TIR = 0dB)

(a1) (b1) co-channel speech signal

(a2) (b2) separated speech signal f_1

(a3) (b3) separated speech signal f_2

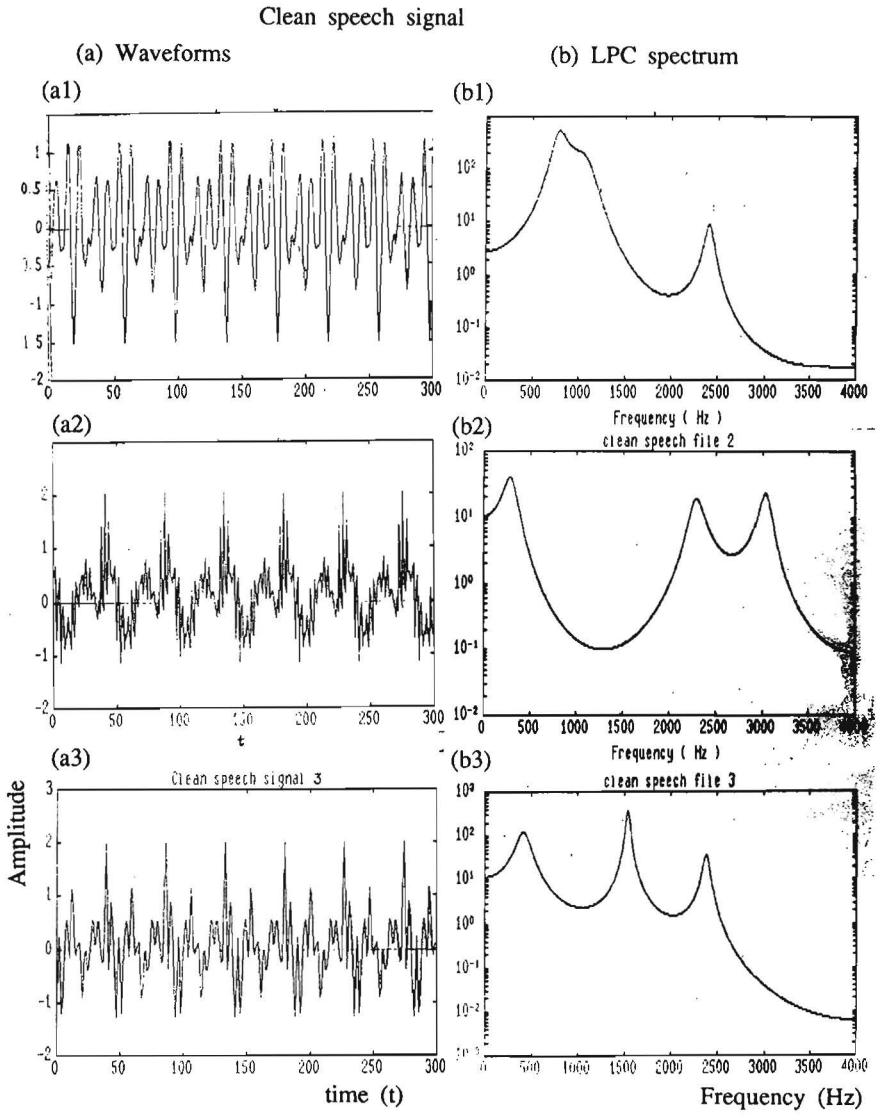


Fig.4.11(3) The clean signal waveforms and spectra (used in Fig.4.11(1), 4.11(2))

(a1) (b1) clean speech signal f_1
(a2) (b2) clean speech signal f_2
(a3) (b3) clean speech signal f_3

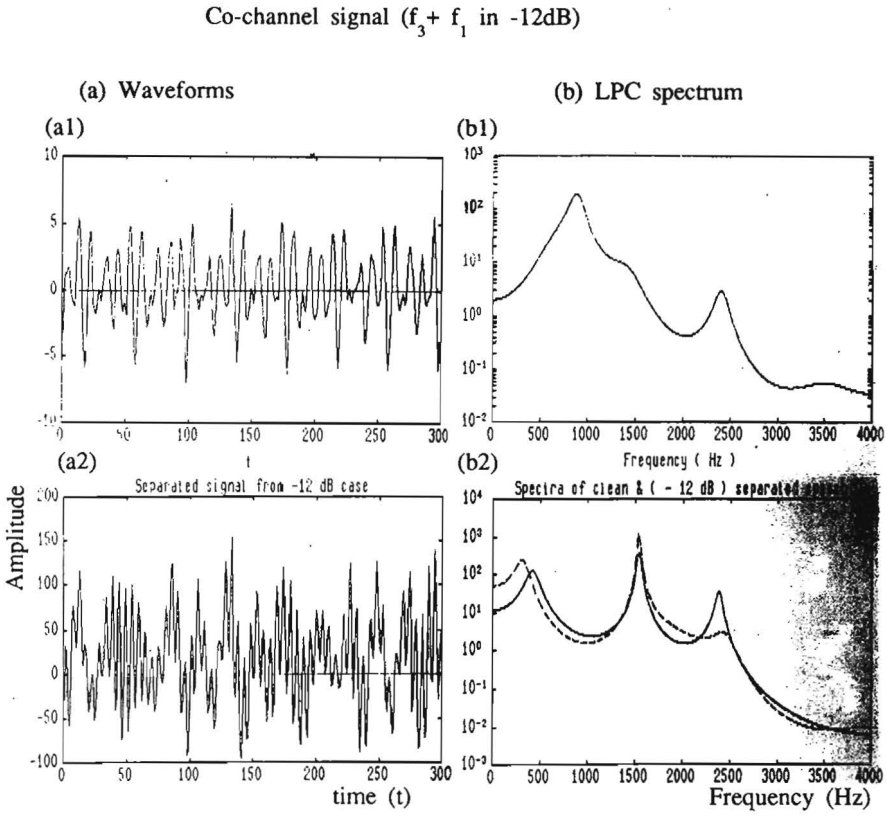


Fig.4.12(1) Separation of summed stationary speech signals by the T-FB domain *linear*-ANC (TIR = -12dB)

(a1) (b1) co-channel speech signal
(a2) (b2) separated weaker speech signal f_3

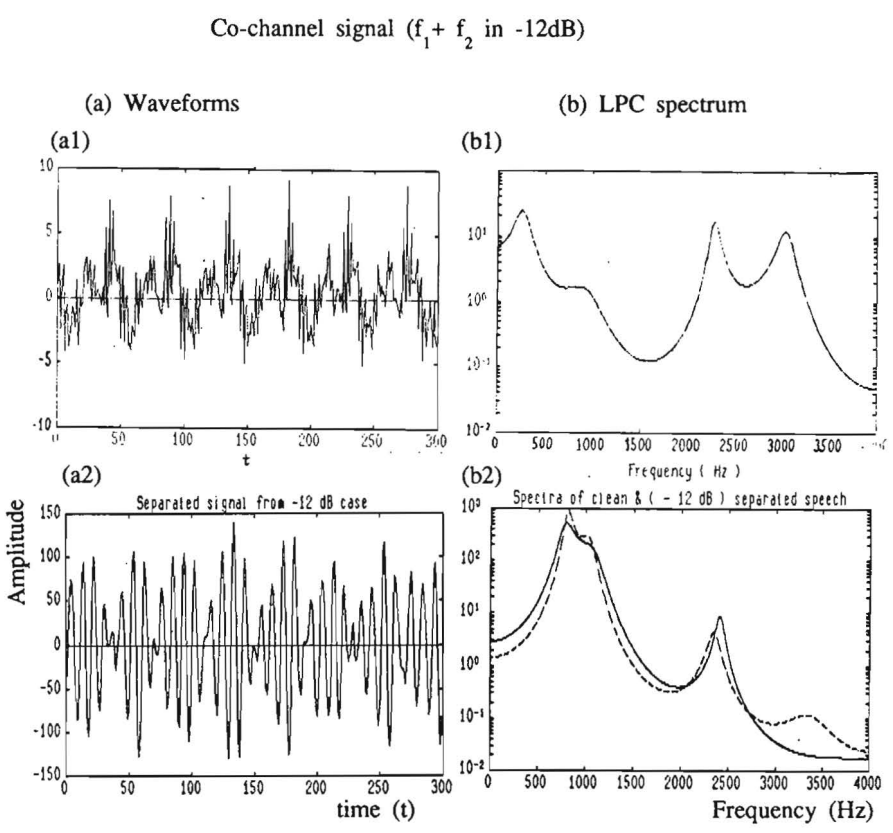


Fig.4.12(2) Separation of summed stationary speech signals by the T-FB domain *linear*-ANC (TIR = -12dB)

(a1) (b1) co-channel speech signal
(a2) (b2) separated weaker speech signal f_1

(a2)Some comparisons

Speech intelligibility enhancement via speech separation using a Harmonic Magnitude Suppression (HMS) approach[37] has proved its usefulness in the negative dB TIR situation. This method became one of the main trends for speech separation, as mentioned in section 4.3.

In order to compare the results obtained from these two methods, similar simulations as (a1) are performed using the HMS technique at TIR=-12dB. In the simulations, the size of each analysis frame is selected L=256 samples, with sample frequency f_s=8kHz. A hamming window is used before FFT transform. Thus, the equivalent bandpass filter bandwidth is $2B=4f_s/L=62.5$Hz (relatively to pitch frequencies 170.2 and 200 Hz, it is a narrowband filterbank). Fig.4.13 includes two simulation results obtained from using the HMS method (using the same summed speech signals as those in Fig.4.12). These simulation results indicate that stationary speech signal separation by the T-FB domain Linear-ANC algorithm in negative dB TIR are slightly better than that by the HMS method.

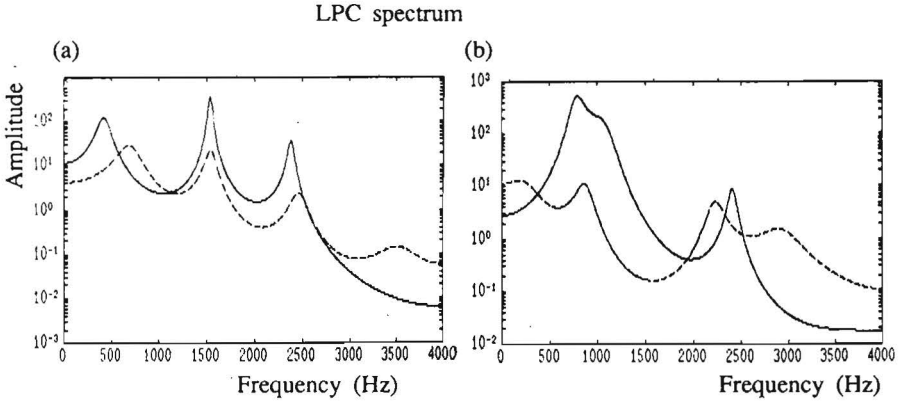


Fig.4.13 Separation of summed stationary speech signals by the *Harmonic Magnitude Suppression (HMS) technique* (TIR=-12dB)

(a) Separated weaker f_3 from summed speech (f_3+f_1) in -12dB
(b) Separated weaker f_1 from summed speech (f_1+f_2) in -12dB

Remarks:

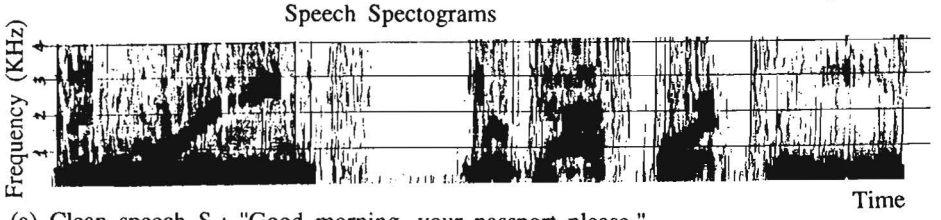
In the simulations of the HMS-based separation approach, there is a spectral peak around 3400 Hz, which is the least common multiple of the two pitch harmonics. Consequently, the HMS-based approach can not resolve these two harmonics according to its separation principle. While in the T-FB domain L-ANC approach, in some of the cases there is also a small peak around 3400 Hz. In principle, this distortion is introduced by poor local TIR values (close to 0 dB).

(b) Simulations on separating summed-synthetic speech sentences with constant pitches

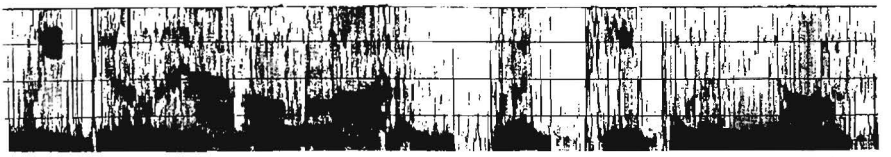
(b1)via the T-FB domain *linear* Adaptive Noise Canceler (L-ANC)

Two synthetic sentences with different but constant pitches are added with a selected global TIR between 0 dB and ±12 dB. The local TIR is calculated in each frame. The short-time signal energy at each bin is also calculated, which is then used for normalizing the filter step-size. The constant filter step-size is chosen $\mu 0=\mu_i=0.1$ for all bins. The hamming window size is selected as L=87. The bandwidth of the speech signals is 0-5KHz, with the sample frequency f_s=10KHz. Thus the equivalent bandpass filter bandwidth $2B=4f_s/L$ =460 HZ. The number of bandpass filters is chosen N=100. These filters are designed to cover only the signals within the frequency band 0-4kHz, in order to decrease the calculation burden. Two groups of weight vectors, each is associated with a specific speaker, are used. The filter weights are updated continuously across the successive frames, if there is no alternation of dominant speaker.

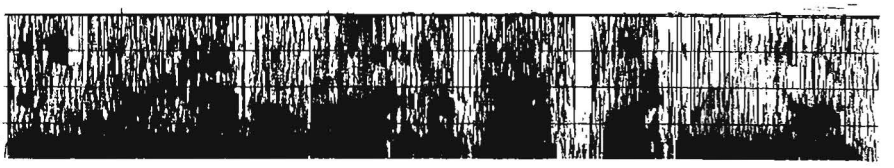
Simulation results showed that the T-FB domain LMS type of L-ANC can adapt quickly to speech sentences. As examples, Fig.4.14 shows the spectrograms of speech sentences before and after the separation, where the TIR =0dB. From the spectrograms, it follows that the intelligibility of the separated speech is well maintained. Informal listening tests also showed intelligibility enhancement of the target speech.



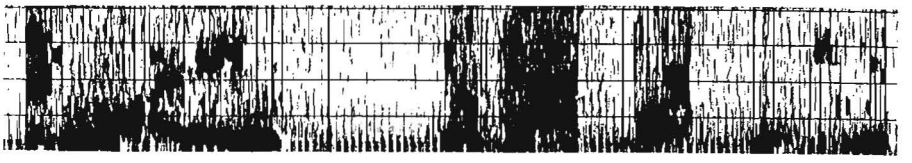
(a) Clean speech S_1: "Good morning, your passport please."



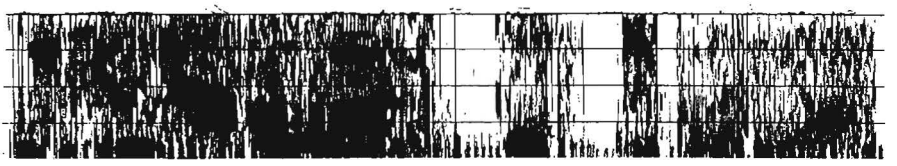
(b) Clean speech S_2: "We do have a lot of good people in the office."



(c) $S_1 + S_2$ in 0 dB



(d) Separated S_1 from (c)



(e) Separated S_2 from (c)

Fig.4.14 Separation of summed synthetic speech sentences with constant pitches by the T-FB domain *linear*-ANC

(TIR = 0 dB, p_1=40, p_2=47 samples, f_s=10kHz)

(b2)via the T-FB domain *NonLinear* Adaptive Noise Canceler (NL-ANC)

Similar to (b1), the co-channel speech signal is obtained by summed synthetic sentences with different constant pitches and a given TIR. For each speaker, there is a corresponding set of filter coefficients. The continuity of the filter weight coefficients associated with each speaker k is considered between the successive frames. The coefficients are updated continuously across the frames if there is no alternation of the dominant speaker in the concerned bin. However, if the dominant speaker changes between the successive frames, the filter coefficients are then initialized before starting a new update.

In the practical algorithm for the simulations, only signals in the same bin are considered as the quadratic terms in the NL-ANC.
At bin i, the linear and quadratic weights are updated separately as follows

$$H1^{(s)}_{j,i}(n+1) = H1^{(s)}_{j,i}(n) + 2\,\mu 1_i e1_{n,i} X_{n-j-p_s,i} \qquad j=0,1..M_s-1 \qquad (4.11.1)$$

$$H2^{(s)}_{i,i}(j,j-p_s,n+1) = H2^{(s)}_{i,i}(j,j-p_s,n) + 2\,\mu 1_i \mathrm{sign}(X_{n-j,i}X_{n-j-p_s,i}) e1_{n,i} \sqrt{|X_{n-j,i}X_{n-j-p_s,i}|} \qquad j=0,1..M_s-1 \qquad (4.11.2)$$

$$H2^{(w)}_{i,i}(j,j-p_w,n+1) = H2^{(w)}_{i,i}(j,j-p_w,n) + 2\,\mu 2_i \mathrm{sign}(e1_{n-j,i}e1_{n-j-p_w,i}) e2_{n,i} \sqrt{|e1_{n-j,i}e1_{n-j-p_w,i}|} \qquad j=0,1..M_w-1 \qquad (4.11.3)$$

where $\mu 1_i = \mu 10_i/(M_s E(X^2_{n,i}))$, $\mu 2_i = \mu 20_i/(M_w E(e1^2_{n,i}))$, $\mu 10_i = \mu 20_i = 0.1$, p_i and M_i are pitch period and the quadratic filter order along the time direction at bin i, respectively (i=S,W corresponds to the stronger and the weaker speaker of bin i), $X_{n,i}$, $e1_{n,i}$ and $e2_{n,i}$ are the bandpass filtered signal, the residuals of NL-ANC$_1$ and NL-ANC$_2$, of the i^{th} bin at time instant n, respectively.

The following output of the NL-ANC$_1$ is associated with the estimated signal component of the stronger-speaker at bin i,

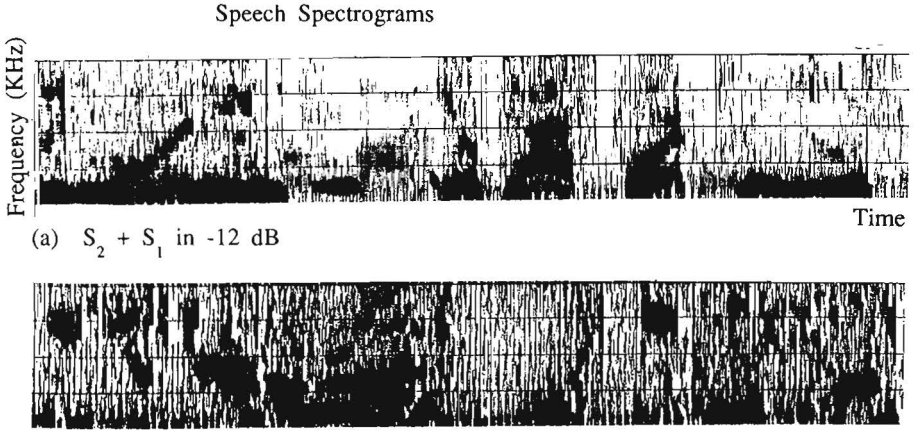
$$y_{n,i} = \sum_{j=0}^{Ms-1} H1_{j,i}^{(s)} X_{n-j-p_s,i} +$$

$$+ \sum_{j=0}^{Ms-1} H2_{i,i}^{(s)}(j,j-p_s)\ \mathrm{sign}(X_{n-j-p_s,i} X_{n-j,i}) \sqrt{|X_{n-j-p_s,i} X_{n-j,i}|} \qquad (4.11.4)$$

While the following output of the NL-ANC$_2$ is the estimated signal component of the weaker-speaker at bin i,

$$Z_{n,i} = \sum_{j=0}^{Mw-1} H2_{i,i}^{(w)}(j,j-p_s)\mathrm{sign}(e1_{n-j-p_w,i}\, e1_{n-j,i}) \sqrt{|e1_{n-j-p_w,i}\, e1_{n-j,i}|} \qquad (4.11.5)$$

Similar simulations are done by using the T-FB domain NL-ANC. Fig.4.15 shows the speech spectrograms of the clean speech sentences, the summed speech sentence at TIR=-12 dB, and the separated sentence of the weaker speaker, respectively.



(a) $S_2 + S_1$ in -12 dB

(b) Separated weaker speech signal S_2 from (a)

Fig.4.15 Separation of summed synthetic speech sentences with constant pitches by the T-FB domain *nonlinear*-ANC (TIR = -12dB, p_1=40, p_2=47 samples, f_S=10kHz)

(See Fig.4.14 for the clean speech S_1 and S_2)

Compared with the simulation results obtained from the T-FB domain L-ANC approach, it has been indicated that the NL-ANC approach further attenuates the interference sound at the expense of slightly more distortion on the target speech signals.

(c) Simulations on separating speech sentences with natural pitches

Preliminary simulations have also been done for separating speech sentences with natural pitches from two females at 0dB TIR. Informal listening tests indicate that there is some kind of reverberation distortion introduced to the separated sentences, although the intelligibility of the separated sentences is rather good.

This distortion may be due to inaccurate pitch values in those voiced-transition frames, where pitch values between the two successive frames change (relatively) quickly. Consequently, this equals to using *biased* pitch period delayed signals for the NL-ANC. Another reason might be the close pitch values of the two female pitches, which leads to more distortion in the separated results. Especially, there present many intercross frames (frames with two similar pitch values) in the testing sentence, which are non-separable by this algorithm.

Summary of the simulations

Simulation of separating the *stationary* speech signals between 0dB and ±12 dB showed excellent results. Compared with the results of the HMS-based approach at -12 dB, we obtained similar or even slightly better quality.

Simulations of separating synthetic speech sentences with constant pitches at TIR between 0dB and ±12dB using the T-FB domain *linear*-ANC and *nonlinear*-ANC have also been done.
The T-FB domain LMS ADF algorithms showed good tracking capability in adaptation to real speech sentences having time-varying vocal-tract parameters.
The linear approach provided good intelligibility of the separated sentences, although there is still some audible background interference sound.
The corresponding NL approach further improves the separated results by

attenuating most of the interference sound, at the price of a slightly increased distortion of the target speech and more calculation burden for the system.

Simulation of separating summed speech signals from two speakers with *natural* pitches did yield reasonable results with good intelligibility but with reverberation. We noticed that the two pitch contours may often intercross and occupy almost the same dynamic range.

4.12. Discussion on future work

The NL approach shows great potential for intelligibility separation of speech. As mentioned in section 4.10, properly selection of NL terms can improve the results. In our simulations, we use only the pitch delayed signal components from the same bin as quadratic terms because of the calculation burden and computer memory.
As mentioned in section 4.10, a more general method can be introduced by using (4.10.5) and (4.10.6). **The quadratic filter weights can be selected adaptively** according to the favorable local TIR values to the concerned speaker. **The linear filter weights can also be adaptively selected**. The constraints to the linear filter weights selection can be added. In the case of (relatively) poor local TIR, one can either select a very limited number of linear filter weights, or neglect the linear part totally. By doing this, one can expect to obtain more benefit and some improvement from the separated results.

For the natural speech sentence separation, a more accurate use of the estimated pitch values is needed, especially among the frames having fast pitch change. For the signals between the two successive frames, perhaps a medium pitch value should be considered. Meanwhile, one should avoid using a high-order filter along the time-direction, as this can cause reverberation distortion.

For improving the filterbank structure, a Discrete Wavelet Transform (DWT) type of filterbank can be a reasonable choice. By arranging frequency bandwidth in a logarithmic scale and using the different time-resolution at different bins, the DWT type of filterbank might produce better subjective results. Thus, further listening improvement on the separated speech may be

obtained.

4.13. Summary and conclusions

We have investigated the speech intelligibility enhancement problem via a new speech separation system. This system uses single-input (one receiver) co-channel speech signals without any a-priori knowledge about the target and the interference speech signals.

This speech separation system possesses the following main original features:

Speech separation is performed by the time-frequency bin domain adaptive filtering on the decomposed (nonstationary) signals, rather than the conventional one-dimensional processing in the frequency-domain or in the time-domain;
It is consistent with the human auditory global processing;
It concentrates on exploring local information for separating co-channel speech and for estimating pitch contours. For example, local signal component time-correlations, short-time local signal energies and TIRs are used for speech separation; the coincidence appearance of local information involved in the signal envelopes and the signal "carriers", the a-priori general knowledge of pitch contours and the stochastic models of pitch contours are used for pitch contour estimation;
It has a highly parallel structure, which might be attractive to fast hardware implementation.

The two speech separation algorithms described can be regarded as direct applications of the T-TB domain *linear* and *nonlinear* NLMS ADF algorithms in chapter 2.

We have described in detail the separation system and the algorithms via linear and nonlinear approaches. We have analyzed the benefit of using NL filter part. In those bins associated with poor TIRs, linear filter weights show a slow convergence. NL approach can be applied to help the speech separation in those bins.

Simulations of co-channel interference speech reduction by both the linear and NL based approaches over a range of TIR between -12 dB and +12 dB have

been done on the summed stationary speech signals with constant pitches, and summed speech sentences with constant pitches and natural pitches. Analysis of the separated results by spectra, spectrograms, and informal listening tests have shown that all these algorithms provide good intelligibility enhancement of the target speech signals. Compared with the results obtained by the linear and NL associated approaches, the NL one has brought further improvements on attenuating most background interference sound with slightly increase distortion of the target speech.

From the above simulations, we can conclude that the T-FB domain NL adaptive filtering is an effective approach for speech separation over a wide range of TIR.

However, the research on speech separation has been concentrated mainly on finding the proper techniques and methods, with off-line simulations and processing. The results are still limited to certain laboratory conditions, and the co-channel speech signal is synthetic and is constrained to the case of two-speakers. Therefore, further improvement is still needed before this work can be put into practical application.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The investigation contained in this dissertation consists of three parts:

- Linear and nonlinear (NL) adaptive filters for nonstationary signals;
- Robust pitch contour estimation from noise contaminated speech;
- Target-speech intelligibility enhancement from co-channel speech by adaptive separation.

Conclusions

- **On LMS adaptive filters**

Linear and nonlinear LMS filters in the time-transform domain have been explored. We have developed new time-transform bin domain linear and nonlinear Normalized LMS (NLMS) adaptive filtering algorithms.
For the NL filter version, a second-order Volterra kernel has been selected. A Gaussian restriction for the filter input (time-domain) data is needed. In particular, we have investigated the algorithms under a "semi-ideal" transform assumption.

The following conclusions can be drawn from this part of the research:

-If an ideal window function in the DSTFT or the DWT is selected, the transform is associated with a "semi-ideal" one.

-For a linear version, because of the signal partial decorrelation under the semi-ideal transform assumption, the filter becomes N-independent sub-filters of different bins, each sub-filter having order M along the time-direction.

-For the nonlinear filter version, the linear and the quadratic filter parts are decoupled in a T-TB domain under the Gaussian time-domain data restriction.

The linear filter part thus behaves the same as that of the linear version.
The quadratic filter part becomes a group of independent sub-filters at different bin-pairs, each having order M along the time-direction, if the semi-ideal transform assumption is satisfied.
Much reduction in the number of quadratic filter coefficients can be obtained, in relation to the base vector characteristics in each specific domain.

-From the relations and similarities among the linear and nonlinear normalized LMS adaptive filtering algorithms in the T-TB and in the transform-domain, it can be concluded that the T-TB domain nonlinear normalized LMS adaptive filtering algorithm is a generalized form. Each of the other three versions can be regarded as a specific degenerated form.

-Though most transforms are not semi-ideal, a properly selected orthogonal transform can decompose signals into nearly orthogonal and non-overlapping bins. Thus, the algorithms under the semi-ideal transform assumption can give good approximate solutions for those nearly semi-ideal cases.

-It is necessary to introduce such T-TB domain filters, because they are more adequate for processing nonstationary signals, and because they can be used also for reducing the filter output time-delay, when signals are associated with a long impulse response length.

■ **On the RLS adaptive filters**

Because of the relatively fast convergence speed, time-domain Linear/ nonlinear RLS filters have been explored. We have derived two new algorithms, an adaptive-sliding-window RLS (linear) covariance lattice filtering algorithm and an adaptive-sliding-window RLS nonlinear algorithm.

The following conclusions can be drawn from this part of the research:
-The RLS adaptive-sliding window covariance lattice (linear FIR) filtering algorithm is an extension of the existing constant-window-length version. Such an adaptive-window is more suitable than the existing algorithms for filtering nonstationary signals with non-constant changing time-varying

statistics.

-The RLS nonlinear (Volterra type FIR) filtering algorithm with an adaptive sliding window introduces finite data-memory. Consequently, this algorithm is more suitable for filtering nonstationary signals than the existing prewindowed NL filter version. Especially, when the nonstationary signals are associated with a time-varying NL model having non-constant changing speed, this algorithm provides better tracking capability.

- **On the robust pitch contour estimation**

Pitch contour estimation from noisy-speech signals has been investigated. A new robust pitch contour estimation algorithm has been developed. It is a combination of a coarse-step for candidate estimation and a detailed-step for HMM-based Maximum Likelihood pitch contour estimation.

The following conclusions can be drawn from this part of the research:

-Pitch information exists within each speech frame as well as among the successive speech frames. Hence, it is unproper to determine the pitch period based on each speech-frame in isolation. It is suggested that a group of *weighted* pitch candidates can be estimated from each frame.

-The new pseudo-perceptual pitch estimation algorithm, using local information from both the signal "envelopes" and from the signal "carriers", is robust for pitch candidate estimation from noisy speech. The information from the signal envelopes and the signal carriers is found to complement each other. The method is consistent with the auditory global speech analysis without mimicking its detailed behavior.

-It is unproper to use simple pitch contour smoothing algorithms without adding general a-priori knowledge about pitch contours, if the speech is "extremely" noisy.

-The Hidden Markov Model (HMM) is found to be a proper stochastic model for pitch contours. The fact that the parameters in each model can be trained from a large set of pitch contours from clean speech signals made it possible to use a-priori general knowledge. The veil between the output

probabilities and the states in the HMM makes the algorithm robust against noise disturbance.

-The HMM-based Maximum Likelihood estimation seems a proper approach for estimating pitch contour from the weighted candidates.

- **On target-speech intelligibility enhancement from co-channel speech by adaptive separation**

Target-speech intelligibility enhancement is investigated for the co-channel speech signal where the interference noise is from a competitive speaker. A new approach of the Time-Frequency Bin (T-FB) domain speech separation is developed, by applying the above T-TB domain linear/nonlinear NLMS adaptive filtering algorithms and the above pitch contour estimation algorithm.

Several conclusions can be drawn from this part of the research:

-Co-channel speech separation performed in the T-FB domain is more suitable than performed in the frequency-domain or the time-domain, from both the signal processing point of view and from the human speech perception point of view.

-It is an approach consistent globally to the human auditory *temporal-place* processing.

-In the T-FB domain, there are more possibilities for a signal processing algorithm to explore local differences between the signal and the interference, and to consider co-channel signal components which evolve with time.

-Theoretically, a time-domain LMS algorithm has been proved to converge to the weights of a dominant voiced-speaker in the summed signals. However, in practice it is generally not possible to have a consistent domination over the whole speech spectrum due to the large dynamic range.

-It can be deduced from [2] that a time-frequency bin domain LMS adaptive filtering algorithm converges at each bin to a *locally* dominant speaker. Thus, it can be applied to the separation of summed voiced speech.

-Co-channel speech separation via the T-FB domain linear/nonlinear adaptive noise canceler has been proved to be able to enhance effectively and intelligibly the target-speech over a range of TIR between -12dB and +12dB. Compared with the linear version, the NL one attenuates more interference sound with slightly more distortion on the target speech.

Future work

From the investigations described in this thesis, some possible directions for future work are found:

- **On the T-TB domain linear/nonlinear NLMS ADF algorithm**

- Further investigation on the T-TB domain linear/nonlinear NLMS ADF algorithms can be concentrated on the dynamic behavior of these filters and their comparisons.

- To investigate the error introduced by using the algorithms under the semi-ideal transform assumption to other non semi-ideal transform cases.

- To compare the difference between using the linear approximate solution and using the nonlinear solution for some nonlinear problems.

- **On the HMM-based pitch contour estimation algorithm**

- To include the unvoiced state in the HMM, in order to handle the transitions between voiced and unvoiced frames.

- To introduce the forward-combining-backward search in order to better handle the pitch crossing points of pitch contours.

- **On the co-channel speech separation algorithm**

- To study the *adaptive selection* of the quadratic filter weights: to select quadratic weights associated with these bin-pairs, which satisfy the frequency constraints and have favorable local TIR values.

- To study the *adaptive selection* of the linear filter weights: to select linear terms only associated with good local TIR condition.

- To test further speech separation on natural sentences. In order to obtain improved results, more accurate use of the estimated pitch value is needed, especially among those frames having quick pitch change. For signals between the two successive frames, a medium pitch value is perhaps a proper choice. Meanwhile, one should avoid using a large filter order along the time-direction to prevent reverberation distortion.

CHAPTER 6

APPENDIX

In the following, the formulas associated with the transform-domain and the time-transform bin domain linear and nonlinear LMS adaptive filtering algorithms obtained from sections 2.2 and 2.3 are listed in Table A.1. Table A.2 lists the corresponding relations between the variables in a transform-domain and a T-TB domain. By replacing the T-TB domain variables with the corresponding transform-domain ones, the T-TB domain algorithm then becomes the corresponding transform-domain one, and the vice versa. Because a T-TB domain algorithm can degenerate into a corresponding transform-domain one by setting M=1, and because a nonlinear filter can degenerate into a linear one by simply neglecting the nonlinear part, a T-TB domain nonlinear LMS adaptive filter is thus a generalized form. This relation is shown schematically in the following Fig. A.1.

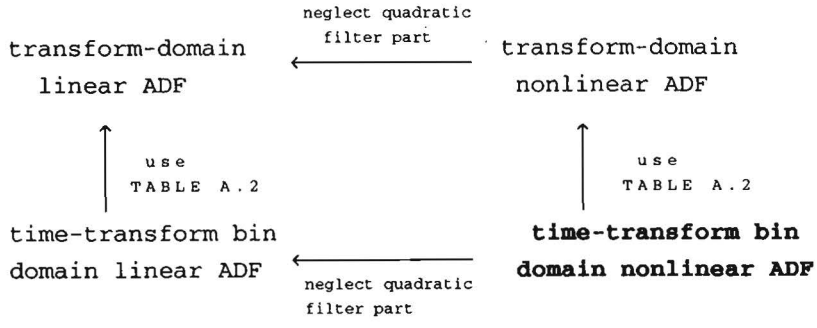


Fig.A.1 Relations among the different LMS adaptive filters

	transform-domain linear ADF	transform-domain nonlinear ADF
data relation	$\underline{Z}_n = \mathbf{W}\underline{x}_n$	
filter relation	$\underline{H}_n = \mathbf{W}\underline{h}_n$	$\underline{H}_n^{(1)} = \mathbf{W}\underline{h}_n^{(1)}$ $\quad \mathbf{H}_n^{(2)} = \mathbf{W}\mathbf{h}_n^{(2)}\mathbf{W}^T$
optimal filter	$\underline{H}_{opt} = \mathbf{R}_{zz}^{-1}\underline{P}_{zd}$	$\underline{H}_{opt}^{(1)} = \mathbf{R}_{zz}^{-1}\underline{P}_{zd}$ $\mathbf{H}_{opt}^{(2)} = 1/2\ \mathbf{R}_{zz}^{-1}\mathbf{R}_{dzz}\mathbf{R}_{zz}^{-1}$
filter update-formulas	$\underline{H}_{n+1} = \underline{H}_n + 2\mu e_n \Lambda^{-2}\underline{Z}_n$	$\underline{H}_{n+1}^{(1)} = \underline{H}_n^{(1)} + 2\ \mu 1\ e_n \Lambda^{-2}\underline{Z}_n$ $\mathbf{H}_{n+1}^{(2)} = \mathbf{H}_n^{(2)} + \mu 2\ e_n(\Lambda^{-2}\underline{Z}_n)(\Lambda^{-2}\underline{Z}_n)^T$
	$\Lambda^2 = \text{diag}[\lvert z_{n1}\rvert^2 \ldots \lvert z_{nN}\rvert^2]$	
	$\mu = \text{diag}[\mu 0_1\ \mu 0_2 \ldots \mu 0_N]$	$\mu 1 = \text{diag}[\mu 10_1\ \mu 10_2\ \ldots\ \mu 10_N]$
	$0 < \mu 0_j \le 1 \quad j = 1..N$	$0 < \mu 10_j \le 1 \qquad j = 1..N$
		$\mu 2 = \begin{pmatrix} \mu 20_{1,1} & \mu 20_{1,2} & \ldots \mu 20_{1,N} \\ \mu 20_{2,1} & \mu 20_{2,2} & \ldots \mu 20_{2,N} \\ \vdots & \vdots & \vdots \\ \mu 20_{N,1} & \mu 20_{N,2} & \ldots \mu 20_{N,N} \end{pmatrix}$
		$\mu 20_{i,j} = \mu 20_{j,i} \quad 0 < \mu 20_{i,j} \le 1/2$
	$\hat{y}_n = \underline{Z}_n^T \underline{H}_n$	$\hat{y}_n^{(1)} = \underline{Z}_n^T \underline{H}_n^{(1)}$
		$\hat{y}_n^{(2)} = \text{tr}(\mathbf{H}_n^{(2)}(\underline{Z}_n \underline{Z}_n^T - \mathbf{R}_{zz}(n))^T)$
		$\hat{y}_n = \hat{y}_n^{(1)} + \hat{y}_n^{(2)}$
	$e_n = d_n - \hat{y}_n$	

Table A.1 Formulas of LMS linear and nonlinear adaptive filters in a transform- and a T-TB domain

Appendix

T-TB domain linear ADF	T-TB domain nonlinear ADF
$\vec{Z}_n = \mathbf{W2}\ \vec{X}_n$	
$\vec{H}_n = \mathbf{W2}\ \vec{h}_n$	$\vec{HI}_n = \mathbf{W2}\ \vec{hI}_n \qquad \mathbf{H2} = \mathbf{W2}\ \mathbf{h2}_n \mathbf{W2}^T$
$\vec{H}_{opt} = \mathbf{R}^{-1}_{\vec{Z}\vec{Z}}\ \vec{P}_{\vec{Z}d}$	$\vec{HI}_{opt} = \mathbf{R}^{-1}_{\vec{Z}\vec{Z}}\ \vec{P}_{\vec{Z}d}$ $\mathbf{H2}_{opt} = 1/2\ \mathbf{R}^{-1}_{\vec{Z}\vec{Z}}\ \mathbf{R}_{d\vec{Z}\vec{Z}}\ \mathbf{R}^{-1}_{\vec{Z}\vec{Z}}$
$\vec{H}_{n+1} = \vec{H}_n + 2\ \mu\ e_n \Lambda^{-2} \vec{Z}_n$	$\vec{HI}_{n+1} = \vec{HI}_n + 2\ \mu 1\ e_n \Lambda^{-2} \vec{Z}_n$ $\mathbf{H2}_{n+1} = \mathbf{H2}_n + \mu 2\ e_n (\Lambda^{-2}\vec{Z}_n)(\Lambda^{-2}\vec{Z}_n)^T$
$\Lambda^2 = \mathrm{diag}[\lvert z_{n1}\rvert^2 \ldots \lvert z_{nN}\rvert^2] \otimes (\mathbf{MI}_M)$	
$\mu = \mathrm{diag}[\mu 0_1 \ldots \mu 0_N] \otimes \mathbf{I}_M$ $(0 < \mu 0_j \le 1) \quad j = 1 \ldots N$	$\mu 1 = \mathrm{diag}[\mu 10_1\ \mu 10_2 \ldots \mu 10_N] \otimes \mathbf{I}_M$ $(0 < \mu 10_j \le 1) \quad j = 1..N$ $\mu 2 = \begin{pmatrix} \mu 20_{1,1} & \mu 20_{1,2} & \ldots \mu 20_{1,N} \\ \mu 20_{2,1} & \mu 20_{2,2} & \ldots \mu 20_{2,N} \\ \vdots & \vdots & \vdots \\ \mu 20_{N,1} & \mu 20_{N,2} & \ldots \mu 20_{N,N} \end{pmatrix} \otimes \mathbf{I}_M$ $\mu 20_{i,j} = \mu 20_{j,i} \quad 0 < \mu 20_{i,j} \le 1/2$
$\hat{y}_n = \vec{Z}_n^T \vec{H}_n$	$\hat{y}_n^{(1)} = \vec{Z}_n^T \vec{HI}_n$ $\hat{y}_n^{(2)} = \mathrm{tr}(\mathbf{H2}_n (\vec{Z}_n \vec{Z}_n^T - \mathbf{R}_{\vec{Z}\vec{Z}}(n))^T)$ $\hat{y}_n = \hat{y}_n^{(1)} + \hat{y}_n^{(2)}$
$e_n = d_n - \hat{y}_n$	

variables in the transform-domain filters	variables in the T-TB domain filters
$\underline{Z}_n$	$\vec{Z}_n$
$\underline{H}_n^{(1)}$ (or $\underline{H}_n$)	$\vec{H1}_n$ (or $\vec{H}_n$)
$\mathbf{H}_n^{(2)}$	$\mathbf{H2}_n$
$\mathbf{R}_{zz}(n)$	$\mathbf{R}_{\vec{Z}\vec{Z}}(n)$
$\underline{P}_{zd}(n)$	$\underline{P}_{\vec{Z}d}(n)$
$\mathbf{R}_{dzz}$	$\mathbf{R}_{d\vec{Z}\vec{Z}}$
$\mu 1$	$\mu 1 \otimes \mathbf{I}_M$
$\mu 2$	$\mu 2 \otimes \mathbf{I}_M$
Λ^2	$\Lambda^2 \otimes (M\mathbf{I}_M)$
$\mathbf{W}$	$\mathbf{W2}=(\mathbf{W}\otimes\mathbf{I}_M)$
variables in the time-domain	
$\underline{x}_n$	$\vec{X}_n$
$\underline{h}_n^{(1)}$(or $\underline{h}_n$)	$\vec{h1}_n$ (or $\vec{h}_n$)
$\mathbf{h}_n^{(2)}$	$\mathbf{h2}_n$

Table A.2 Corresponding relations between the variables in a transform- and a T-TB domain

CHAPTER 7

ABBREVIATION LIST

ADF	ADaptive Filtering
AGC	Automatic Gain Control
AMDF	Average Magnitude Difference Function
ANC	Adaptive Noise Canceler
AR	Autoregressive
ARMA	Autoregressive Moving Average
ASR	Automatic Speech Recognitiom
BM	Basilar Membrane
BPE	Backward Prediction Error
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSB	Double SideBand
DSTFT	Discrete Short Time Fourier Transform
DSP	Digital Signal Processor
DP	Dynamic Programming
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FIR	Finite impulse response
FPE	Forward Prediction Error
HMM	Hidden Markov Model
HMS	Harmonic Magnitude Suppression
i.i.d.	independent identical distribution
IIH	Interspike Interval Histogram
IIR	Infinite Impulse Response
KLT	Karhunen-Loeve Transform
L-ANC	Linear Adaptive Noise Canceler
LMS	Least Mean Squares
LS	Least Squares
MA	Moving Average
MAP	Maximum A-Posterori

MCESA	Minimum-Cross Entropy Spectral Analysis
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
NL	NonLinear
NL-ANC	NonLinear Adaptive Noise Canceler
NLMS	Normalized Least Mean Squares
pdf	probability density function
RLS	Recursive Least Squares
SNR	Signal to Noise Ratio
SW	Sliding Window
TB	Transform Domain
T-FB	Time-Frequency Bin
TIR	Target to Interference Ratio
T-TB	Time-Transform Bin
WL	Window Length
WHT	Walsh-Hadamard Transform

REFERENCES

[1] N. Ahmed and K.R. Rao, "Orthogonal transforms for digital signal processing", U.S.A., 1975.

[2] S.T. Alexander, "Adaptive Reduction of Interfering Speaker Noise Using the Least Mean Squares Algorithm", PP.728-731, ICASSP 87.

[3] J.B. Allen, "Cochlear modeling", pp.3-29, IEEE ASSP magazine, 1985.

[4] M.S. Andrews, et. al., "Robust pitch determination via SVD based cepstral method", pp253-256, ICASSP 1990.

[5] M.R. Asharif, et. al., "Frequency domain noise canceler: Frequency bin adaptive filtering", pp. 2219-2222, ICASSP 1986.

[6] L.E. Baum & J.A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", pp. 360-363, Bull. Amer. Math. Soc. 73, 1967.

[7] L.E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process", Inequalities III, pp. 1-8, 1972.

[8] L.E. Baum, & T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains", pp. 1554-1563, Ann. Math. Stat. 37, 1966.

[9] N.J.Bershad & P.L.Feintuch, "A normalized frequency domain LMS adaptive algorithm", pp452-461, Vol-34, IEEE trans. on ASSP, 1986.

[10]D. Burshtein, "Joint ML estimation of pitch and AR parameters using the EM algorithm", pp. 797-800, ICASSP 1990.

[11]F.J. Charpentier, "Pitch detection using the short-term phase spectrum", pp.113-116, ICASSP,1986

[12]D.G. Childers & C.K. Lee, "Co-channel Speech Separation", PP181-184, ICASSP 1987.

[13]Y.S. Cho, S.B. Kim, E.L. Hixson, E.J. Powers, "Nonlinear distortion analysis using digital higher-order coherence spectra", pp.1165- 1168, ISCAS 1990.

[14]C.K. Chui, "An introduction to Wavelets", Academic Press, INC., U.S.A. 1992.

[15]C.K. Chui, "Wavelet: a tutorial in theory and application", Academic Press, INC. U.S.A., 1992.

[16]C.E. Davlia, et.al., "A second-order adaptive Volterra filter with rapid convergence", IEEE ASSP vol-35, pp1259-1263, 1987.

[17]S. Dimolisas, "Objective speech distortion measures and their relevance to speech quality assessments", pp317-324, vol 136, No.5, IEE Proc., 1989.

[18]D.E. Dudgeon and R.M. Mersereau, "Multidimensional digital signal processing", Prentice-Hall,Inc, 1984.

[19]Y. Ephraim & R.M. Ray, "A unified approach for encoding clean and noisy sources by means of waveform and AR model vector quantisation", IEEE trans Inform. Theory, Vol.IT-34, No.4, pp826-834, 1988.

[20]Y. Ephraim, D. Malah and B.H. Juang, "On the application of Hidden Markov models for enhancing niosy speech", IEEE trans. ASSP, Dec, 1989.

[21]Y. Ephraim, "Speech enhancement based upon HMM", pp 353-356, ICASSP 1989.

[22]Y. Ephraim, "A minimum mean Square Error approach for speech enhancement", pp 829-832, Vol 2, ICASSP 1990.

[23]R.H. Frazier, "An adaptive filtering approach towards speech enhancement", M.I.T. S.M. and E.E. thesis, June, 1975.

[24]B. Friedlander, "Lattice filters for adptive processing", Proc. IEEE, pp. 813-829, 1982.

[25]O. Ghitza, "Auditory nerve representation criteria for speech analysis/synthesis", pp736-740, No.6,Vol ASSP-35, IEEE Trans. ASSP, 1987.

[26]G.B. Giannakis & A.V.Dandawate, "Linear and NL adaptive noise cancelers", pp1373-1376, ICASSP 1990.

[27]G.B. Giannakis & et.al, "Higher-order statistics based input/output system identification and application to noise cancellation", to be published in Circuits systems and signal processing.

[28]J.L. Goldstein & P. Srulovicz, "auditory-nerve spike intervals as an adequate basis for aural spectrum analysis", in Psychophysics and Physiology of Hearing, E.F.Evans and J.P. Wilson, Eds., London, England:Academic, 1977.

[29]Y.H. Gu, "RLS lattice and circular lattice with real time variable sliding window", pp 916-919, vol. 2, proc. IEEE Inter. conf. on ASSP, ICASSP-1989.

[30]Y.H. Gu, & W.M.G. van Bokhoven, "A Frequency Bin Adaptive Separation Aproach for Co-channel Interference Speech Suppression", pp1099-1102, vol.2, Signal Processing V: theories & applications, Elsevier Science Publishers B.V., 1990.

[31]Y.H. Gu & W.M.G. van Bokhoven, "Co-channel speech separation using frequency bin nonlinear adaptive filtering", pp.949-952, vol.2, proc. IEEE Inter. Conf. ASSP, ICASSP-1991.

[32]Y.H. Gu & W.M.G. van Bokhoven, "A RLS Variable-length sliding-window nonlinear filtering algorithm for system identification and adaptive noise cancellation", pp. 2806-2809, vol.5, proc. IEEE inter. symp. CAS, ISCAS-1991.

[33]Y.H. Gu & W.M.G. van Bokhoven, "Frequency bin nonlinear LMS adaptive noise canceler and its application to co-channle speech noise reduction", pp. 2822-2825, vol 5, proc IEEE inter. symp. CAS, ISCAS-1991.

[34]Y.H. Gu, "A Robust Pseudo perceptual pitch estimator", pp. 453-456, vol. 2, proc. of EUROSPEECH 91, Italy, 1991.

[35]Y.H. Gu, "HMM-based noisy-speech pitch contour estimation" pp.II- 21-24, vol 2, proc. IEEE Inter. Conf. ASSP, ICASSP-1992.

[36]F. Halwatsch & G.F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representation", pp.21-67, IEEE SP magazine, April, 1992.

[37]B.A. Hanson,and D.Y. Wang, "The Harmonic Magni tude Suppression Technique for Intelligibility Enhancement in the Presence of Interfering Speech", PP.18.A.5.1-5.4, ICASSP 1984.

[38]D.J. Hermes, "Measurement of pitch by subharmonic summation", pp.257-264, J.Acoust. Soc. Am. 1988.

[39]W. Hessm, "Pitch determination of speech signals", New York: Springer, 1983.

[40]J.N. Holmes, "Speech synthesis and recognition", Van Nostrand Reinhold Co. Ltd., England, 1988.

[41]Hodgson, et. al. "Nonlinear multiplicative cepstral analysis for pitch extraction in speech", pp257-260, ICASSP 1990.

[42]J.E. Hudson, "Adptive array principles", Peter Peregrinus ltd., England, 1981.

[43]K.H. Kim, S.B. Kim, and E.J. Powers, "Fast RLS algorithms for general filters" pp181-184, proc. of the 2nd European Signal Processing Conference, Spain, 1990.

[44]T. Koh & E.J. Powers, "An adaptive nonlinear digital filter with lattice orthogonalization " pp. 37-40, ICASSP 1983.

[45]T. Koh & E.J. Powers, "Second-order Volterra filtering and its application to NL system identification", IEEE ASSP vol-33, pp.1445-1455,1985.

[46]B. Koo, J.D. Gibson & S.D. Gray, "Filtering of colored noise for speech enhancement and coding", pp 349-352, ICASSP 1989

[47]G.E. Kopec & M.A. Bush, "An LPC-based spectral similarity measure for speech recognition in the presence of co-channel speech interference", pp 270-273, ICASSP 1989.

[48]P. Kroon & B.S. Atal, "Pitch predictors with high temporal resolution", pp661-664, ICASSP 1990.

[49]k.F. Lee, "Automatic speech recognition", Kluwer Academic Publishers, The Netherlands, 1989.

[50]W.C.Y. Lee, "Mobile cellular-telecommunications system", McGraw- Hill Book Company, U.S.A., 1989.

[51]J.C.R. Licklider, "Three auditory theories", Psychology: A study of Science, S.Koch McGraw Hill, 1959.

[52]R.F.Lyon, "A computational model of binaural localization and separation", pp. 1148-1151, ICASSP 1983.

[53]R.F. Lyon, "Computational models of neural auditory processing", pp36.1.1-36.1.4, ICASSP 1984.

[54]D. Mansour and A.H. Gray, et.al., "Frequency domain nonlinear adaptive filter", pp. 550-553, ICASSP 1981.

[55]J.S. Marques, et.al., "Improved pitch prediction with fractional dalays in CELP coding", pp.665-668, ICASSP, 1990.

[56]J.P. Martens & L.V. Immerseel, "An auditory model based on the analysis of envelop patterns", pp401-404, ICASSP 1990.

[57]A.V. Mathews & J. Lee, "Second order Volterra filtering and its application to nonlinear system identification", pp. 1445-1455, IEEE trans. on ASSP, 1985.

[58]V.J. Mathews & J. Lee, "A fast RLS second order Volterra filter", pp.1383-1386, ICASSP 1988.

[59]V.J. Mathews, "Adaptive polynomial filters", pp. 10-26, IEEE SP magazine, July, 1991.

[60]R.J. McAulay & T.F. Quatieri, "Pitch estimation and voicing detection based on a sinusoidal speech model", pp. 249-252, ICASSP 1990

[61]Y. Medan & E. Yair, "Pitch synchronous spectral analysis scheme for voiced speech", vol.37, pp1321-1328, IEEE trans. on ASSP, 1989.

[62]Y.Medan, E.Yair, and D.Chazan, "Super resolution pitch determination of speech signals", pp.40-48, vol 39, IEEE trans on ASSP, 1991

[63]K. Min, D.Chien, et al., "Automated Two Speaker Separation System", pp. 537-540, ICASSP 1988.

[64]D.P. Morgan & C.L. Scofield, "Neural networks and speech processing", Kluwer Academic Publishers, Netherlands, 1991.

[65]S.W. Nam, S.B. Kim & E.J. Powers, "On the identification of a third-order Volterra nonlinear system using a frequency-domain block RLS adaptive algorithm", pp.2407-2410, ISCAS 1990.

[66]J.A. Naylor & S.F. Boll, "Techniques for suppression of interfering talker in co-channel speech" pp205-208, ICASSP 1987.

[67]S.S. Narayan, et al., "Transform domain LMS algorithm", pp. 609-615, No.3, Vol 31, IEEE ASSP, 1983.

[68]L.T. Nile and H.F. Silverman, "Combining HMM and Neural network classifiers", pp417-420, IEEE Inter. Conf. on ASSP, 1989.

[69]D. O'haughnessy, "Speech enhancement using vector quatization and a formant diatance measure", pp.549-552, ICASSP 1988.

[70]A.V. Oppenhaim, & R.W. Schafer, "Digital signal processing", Prentice-Hall, Inc, 1975.

[71]T.W. Parsons, "Separation of Speech from interfering speech by Means of Harmonic Selection", PP911-918, No.4, Vol.60, J.Acoust. Soc. Am., 1976.

[72]Y.M. Perlmutter, L.D. Braids, R.H. Frazier, and A.V.Oppenheim, "Evaluation of a speech enhancement system" pp212-215, ICASSP 1977.

[73]J. Picone,"Continuous speech recognition using HMMs", pp26-41, IEEE ASSP Magazine, July, 1990.

[74]S.U. Pillai, "Array signal processing", Springer-Verlag, 1989.

[75]B. Porat, B. Friedlander and M. Morf, "Square root covariance ladder algorithms", IEEE Trans. on Autom. Contr., pp.813-829, 1982.

[76]A.B. Poritz, "HMM: a guided tour", pp 7-13, ICASSP 1988.

[77]R.K. Potter at.el, "Visible Speech", Dover Publications, New York, 1966.

[78]T.F. Quatieri & R.G. Danisewicz, "An Approach to Co-channel Talker Interference Suppression using a Sinusoidal Model for Speech", PP. 565-568, ICASSP 1988.

[79]T.F. Quatieri & R.G.Danisewicz, "An approach to co-channel talker interference suppression using a sinusodial model for speech", pp56-69, vol.38,IEEE trans. on ASSP, 1990.

[80]T.F. Quatieri & R.J. McAulay, "Noise reduction using a soft-decision sine-wave vector quantizer", pp. 821-824, ICASSP, 1990.

[81]L.R. Rabiner & R.W. Schafer,"Digital processing of speech signals", Prentice-Hall Inc. 1978.

[82]R.P. Ramachandran & P. Kabal, "Pitch prediction filters in speech coding", pp. 467-478, IEEE Trans. on ASSP, 1989.

[83]O. Rioul & M. Vetterli, "Wavelets and signal processing", pp. 1538, IEEE ASSP magazine, Oct. 1991.

[84]C. Rogers, D. Chien, et al., "Neural Network Enhancement for a Two Speaker Separation System", PP 357-360, ICASSP 1989.

[85]S. Seneff, "Pitch and spectral estimation of speech based on auditory synchrony model", pp36.2.1-4, ICASSP 1984.

[86]V.C. Shields, "Separation of added speech signals by digital comb-filtering", M.I.T. S.M. Thesis, Sept, 1970.

[87]J.J. Shynk, "Frequency-domain and multirate adaptive filtering", pp15-37, IEEE SP Magazine, Jan. 1992.

[88]F.M. Silva & L.B. Almeida, "Speech separation by means of stationary least-squares harmonics estimation", pp809-812, ICASSP 1990.

[89]M. Slaney & R.F. Lyon, "A perceptual pitch detector", pp.357-360, ICASSP 1990.

[90]W. Soede, "Improvement of speech intelligibility in noise", Ph.D dissertation, Delft University of Technology, The Netherlands, 1990.

[91]T.V. Sreenivas, et.al.,"Spectral resolution and noise robustness in auditory modelling", pp 817-820, ICASSP 1990.

[92]J.C. Stapleton & S.C. Bass, "Adaptive noise cancellation for a class of NL, dynamic reference channels", IEEE CAS vol-32, pp143-150, 1985.

[93]R.J. Stubbs & Q. Summerfield, "Evaluation of two voice-separation algorithms using normal-hearing and heraing-impaired listerners", pp.1236-1248, vol.84. No.4, J.Acoust. Soc. Am. 1988.

[94]M.A. Syed & V.J. Mathews, "Lattice and QR decomposition-based algorithms for RLS adaptive nonlinear filters", pp262-265, ISCAS 1990.

[95]S. Tamura & A. Waibel, "Noise reduction using connectionist models", pp 553-556, ICASSP 1988.

[96]S. Tamura, "An analysis of a noise reduction neural network", pp. 2001-2004, ICASSP 1989.

[97]S. Tamura & M. Nakamura, "Improvement to the noise reduction neural network", pp825-828, ICASSP 1990.

[98]A.P. Varga & R.K. Moore, "HMM decomposition of speech and noise", pp 845-848, vol. 2, ICASSP 90.

[99]R.M. Warren and G.L. Sherman, "Phonemic restorations based on subsequent content", Perception and Psychophysics, 16(1), pp. 150-156, 1974.

[100]M.Weintraub, "A Computational Model for Separating Two Simultaneous Talkers", PP. 81-84, ICASSP 1984.

[101]M. Weintraub, "A Theory and computational model of auditory monaural sound separation", Ph.D. dissertation, Stanford University, 1985.

[102]C. Wheddon & R. Linggard, "Speech and language processing", Chapman and Hall, U.K. 1990.

[103]B. Widrow, et. al. ,"Adaptive antenna systems", pp2143-2159, vol 55, Proc. of the IEEE, 1967.

[104]B. Windrow, et al., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter ", Vol. 64, No.3, PP. 1151-1162, Proc. IEEE, 1976.

[105]J.D. Wise, et.al., "ML pitch estimation", pp.418-423, No.5, vol ASSP-24, IEEE trans. ASSP, 1976

SAMENVATTING

In vele praktijktoepassingen is het nodig de verstaanbaarheid te verbeteren van een spraaksignaal ingebed in ruis. De complexiteit van de vereiste signaalverwerking blijkt sterk afhankelijk te zijn van de statistische eigenschappen van de storende ruis. In dit proefschrift zullen we de situatie bestuderen waarin een tweede spraaksignaal de ruisbron vormt.

Daartoe ontwikkelden we nieuwe lineaire en niet-lineaire adaptieve-filteringstechnieken evenals krachtige algoritmen voor het schatten van het toonhoogteverloop (de "pitch"). Deze technieken en algoritmen worden toegepast voor het verbeteren van de verstaanbaarheid van spraaksignalen in een gemeenschappelijk kanaal. Ze zijn daarnaast bruikbaar voor een breed gebied van toepassingen.

A. Adaptieve-filteringstechnieken

In vele gevallen zijn de te filteren signalen niet-stationair, d.w.z. ze zijn geassocieerd met tijdsafhankelijke lineaire en niet-lineaire systeemmodellen. In ons geval blijkt het onvoldoende om de signalen slechts in het tijddomein of slechts in een transformdomein te filteren (in plaats van in een tijd-transformdomein). Daarnaast kan de lange impulsresponsie van een signaal een hoge filterorde in het tijddomein vereisen. Dit kan leiden tot grote tijdvertragingen in de filteruitvoer.

Hierdoor gemotiveerd hebben we ons gericht op LMS-type filters, bekend om hun eenvoud en robuustheid. Nieuwe LMS-type lineaire en niet-lineaire (2e orde Volterra) adaptieve filters voor een tijd-transformdomein zijn ontwikkeld onder de aanname van Gaussische data in het tijddomein.

We hebben de algoritmen voornamelijk beschouwd onder de semi-ideale-transformatievoorwaarde. Een semi-ideale transformatie definiëren we als een ééndimensionale orthogonale transformatie die de signalen projecteert op orthogonale niet-overlappende deelruimten (de zogenaamde "bins"). Onder deze aanname zijn de filtercoëfficiënten gedecorreleerd langs de "bin"-richting. Dat wil zeggen dat de lineaire filtercoëfficiënten voor elke "bin" onderling onafhankelijk zijn en de kwadratische filtercoëfficiënten voor elk "bin"-paar.

Het bestaan van een dergelijke semi-ideale transformatie wordt aangetoond. Een speciale keuze van de vensterfuncties in de "Discrete Short Time Fourier Transform" of in de "Discrete Wavelet Transform" leidt tot een semi-ideale transformatie. Bij de keuze van een bijna semi-ideale orthogonale transformatie, kan het tijd-transformdomeinalgoritme onder de semi-ideale aanname gebruikt worden als een goede benadering.

De diverse uitdrukkingen voor de lineaire en niet-lineaire algoritmen in het tijd-transformdomein en in het transformdomein worden met elkaar vergeleken. Hieruit blijkt dat het "time-transform bin domain nonlinear normalized Least Mean Square adaptive filtering" algoritme de generaliseerde vorm is die alle andere algoritmen bevat.

Naast het LMS-type hebben we ook RLS-type lineaire en niet-lineaire filters

onderzocht. RLS-type filters vertonen in het algemeen een snelle convergentie en voeren de kleinste-kwadratenberekeningen exact uit, zonder dat de aanname van Gaussische (tijddomein) invoerdata vereist is.

Twee nieuwe RLS-type adaptieve-filteringsalgoritmen, met een adaptief glijdend venster, voor filtering in het tijddomein zijn ontwikkeld: zowel een lineair als een niet-lineair filter. Deze algoritmen bieden flexibele-volg-mogelijkheden voor het adaptief filteren van niet-stationaire signalen. De RLS-type filters zijn vooral zinvol in het geval van signalen met een niet-constante snelheidsverandering van de tijdsafhankelijke statistische eigenschappen.

B. Robuste schatting van het toonhoogteverloop

We hebben een algemeen geraamte gebouwd voor het schatten van het toonhoogteverloop van een spraaksignaal ingebed in ruis. Een ruwe stap waarin een aantal kandidaten voor de toonhoogte worden bepaald gevolgd door een gedetailleerde stap waarin stochastische modellen worden gebruikt voor het kiezen van het meest-waarschijnlijke toonhoogteverloop. Dit twee-staps-algoritme is ontworpen om gebruik te kunnen maken van de informatie in de "intra and inter speech frames". Het algoritme maakt gebruik van de algemene kennis over optredende toonhoogteverlopen.

Een nieuw algoritme voor de schatting van toonhoogtekandidaten (dat enige overeenkomst vertoont met de menselijke waarneming) maakt gebruik van de plaatselijke signaaldraaggolven en van de plaatselijke signaalomhullenden. De kandidaatselectie is vervolgens gebaseerd op het gelijktijdig optreden van toonhoogte-gecorreleerde informatie over alle frequentie-"bins".

Een nieuw algoritme voor het schatten van het toonhoogteverloop, gebaseerd op een "Hidden Markov model", benut de correlatie van de toonhoogteperioden in een aantal opeenvolgende "frames" (toonhoogteverloop). Een stochastisch model beschrijft de toonhoogtedynamica aan de hand van de autocorrelaties van de toonhoogte en van de eerste en hogere orde afgeleiden hiervan. Als gevolg van het leerproces bevat het model enige a-priori kennis van toonhoogteverlopen. Deze kennis kan van nut zijn voor het schattingsproces in het geval het spraaksignaal slechts ingebed in zeer sterke ruis beschikbaar is .

C. Verstaanbaarheidsverbetering door middel van spraakscheiding

De verbetering van de verstaanbaarheid van één spraaksignaal afkomstig uit een gemeenschappelijk-kanaalsignaal, is in dit proefschrift onderzoekt. Het gemeenschappelijk-kanaalsignaal wordt gedefineerd als de som van twee spraaksignalen (het doelsignaal en het stoorsignaal) in een één kanaal.

Nieuwe algoritmen voor een spraakscheidingssysteem zijn ontwikkeld. Deze zijn geschikt voor het gemeenschappelijksignaalkanaal bij doel-storingsenergie-ratio's (TIR's) tussen -12 dB en +12 dB. Dit systeem bestaat uit een toonhoogteschattingsdeel en een spraakscheidingsdeel.

In het spraakscheidingsdeel worden de bovengenoemde tijd-transformdomein (lineaire én niet-lineaire) adaptieve-filteringstechnieken toegepast als

ruisonderdrukkers.

In het toonhoogteschattingsdeel wordt het bovengenoemde twee-stapsalgoritme toegepast voor het gelijktijdig schatten van het meervoudige toonhoogte-verloop.

De spraakscheidingsalgoritmen zijn getest aan de hand van gesommeerde stationaire synthetische spraaksignalen, gesommeerde synthetische uitgesproken zinnen met constante toonhoogte en natuurlijke toonhoogten met een TIR tussen 0 dB en -12 dB. Uit de computersimulaties blijkt dat een goede verstaanbaarheid van het spraaksignaal wordt verkregen. Het lineaire algoritme laat nog enige ongewenste spraak achter. Het niet-lineaire algoritme verwijdert ook deze, maar geeft iets meer vervorming van het doelsignaal.

CURRICULUM VITAE

GU, Yu Hua was born on May 19, 1953 in Shanghai, China.

She entered East China Normal University in February 1978, after the university system was interrupted for 10 years due to the "Cultural Revolution". She received the B.Sc. degree in January 1982, directed in electronics and the M.Sc. degree in December 1984, directed in digital signal processing. She stayed at East China Normal University as a scientific employee in the Department of Electronics till August 1985. From September 1985 till April 1988 she was engaged in graduate research in the Department of Electrical Engineering of Shanghai Jiao Tong University.

From September 1988 till September 1992 she worked as a research employee in the Circuit and System Design group (EEB), Faculty of Electrical Engineering, Eindhoven University of Technology, The Netherlands.

STATEMENTS

(1) Voiced speech separation by the harmonic magnitude suppression technique shows two fundamental disadvantages: it is basically inconsistent with human auditory global processing; and it dissolves speech harmonics from each *isolated* frame without considering the correlations among the frames.
(This thesis, chapter 4)

(2) It is generally not possible to have a consistent domination of one speaker over the whole speech spectrum. Consequently, the algorithm as proposed by Alexander will have difficulties when applied to voiced speech separation.
(This thesis, chapter 4; S.T.Alexander, proc. ICASSP 1985)

(3) A time-frequency domain LMS adaptive filtering algorithm converges at each bin to the *locally* dominant speaker. Thus, it can be applied to voiced speech separation.
(This thesis, chapter 4)

(4) It is worthwhile to use global processing of the human auditory system for machine speech intelligibility enhancement. However, contrary to Weintraub's opinion it is neither possible nor necessary to mimic all the micromechanisms of the human auditory system.
(This thesis, chapter 4; M.Weintraub, Ph.D. diss., Stanford Univ., 1985)

(5) A transform-domain LMS (or a block LMS) adaptive filter does not consider the time-evolving process of signal components, thus it is not adequate for nonstationary signal filtering. This disadvantage can be overcome by using a time-transform domain LMS adaptive filter, performing on the temporally localized signal components.
(This thesis, chapter 2)

(6) A successful Ph.D. research demonstrates one's ability of attacking difficult technical problems and doing research. It does not mean that one is only able to work in that small specialized field.

(7) An optimal filter without adaptation only remains optimal in a time-invariant system. A plan economy without adaptation by effective feedback from the dynamic market thus is not adequate.

(8) Understanding another culture is only possible through a noisy communication channel. To solve this estimation problem, in an acceptable way, one has to train the human neural network. Then cultural differences can be understood by the human brain.

(9) True religion and true science are always in agreement. True religion can never be opposed to scientific facts; true science which discovers the laws of the universe and supports our material and mental advancement can never be opposed to true religion which reveals spiritual truths.
(Gloria Faizi, "The Bahá'í faith: an introduction", p.73)

(10) When attending an international conference, the bi-directional exchange and stimulation of ideas are more important than the uni-directional presentation of a paper.

(11) The visa requirements for refugees from the former Jugoslavian republics, as imposed by several European governments, are a violation of the principle of "non-refoulement"[*] which is considered binding on all states.

[*] Article 33 of the 1951 United Nations Convention relating to the Status of Refugees.