Signal Subspace Methods for Speech Enhancement

Ph.D. Thesis

Peter S. K. Hansen

LYNGBY 1997

IMM-PHD-1997-42

IMM

IMM
DEPARTMENT OF MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Lyngby – Denmark

1997-09-30
pskh

Signal Subspace Methods for Speech Enhancement

Ph.D. Thesis

Peter S. K. Hansen

LYNGBY 1997

IMM-PHD-1997-42

IMM

ISSN 0909–3192

Printed by IMM, Technical University of Denmark

Preface

Purposes and Scope. This thesis has been prepared at the Department of Mathematical Modelling (IMM), Technical University of Denmark (DTU). It is a partial fulfillment of the requirements for the degree of Ph.D. in electrical engineering.

The subject is Signal Subspace Methods for Speech Enhancement where techniques from the areas of signal processing and numerical linear algebra are combined in order to obtain reliable and efficient algorithms.

Many of the techniques, described in this thesis, are on the forefront of technology and might lack sufficiently powerful hardware for practical applications today, but are merely natural parts of a theoretical framework. I hope the reader will have this in mind while reading the thesis.

Acknowledgments. During the course of this Ph.D. study several people have provided various degrees of support, and I would hereby like to record my thanks for their assistance.

First of all, I wish to acknowledge my advisors Associate Professor Steffen Duus Hansen, Associate Professor John Aasted Sørensen and Professor Per Christian Hansen at IMM, Technical University of Denmark for their encouragement and advice during my study.

Part of the work was made while I enjoyed the pleasant working conditions at the EE department, Princeton University, New Jersey, USA, and I would like to thank Professor S. Y. Kung and the members of his research group for many good discussions.

Thanks is also due to Søren Holdt Jensen, Ph.D., for making speech measurements available for me, and for introducing me to subspace based speech enhancement in the first place.

I also wish to acknowledge the staff at the Section for Digital Signal Processing, IMM, DTU for providing stimulating working conditions, especially my room mate Peter A. Toft, Ph.D., for his proofreading of the manuscript and good sense of humor.

Finally, I would like to thank my wife Jette and our son Martin (born on June 16, 1996) for their love and patience throughout this long project.

The financial support provided by DTU and the Danish Research Academy are gratefully acknowledged.

Papers. A number of publications have been produced during the Ph.D. period. The paper [53] has been presented at the *EUSIPCO Conference 96* in Trieste, Italy, and the paper [54] has been presented at the *IWAENC Workshop 97* in London, England. Two software packages have been developed and documentation for one of them is available in [52]. Finally, reports for teaching purposes have been made.

Lyngby, September 1997

Peter S. K. Hansen

ABSTRACT

This thesis focus on the theory, analysis and algorithm aspects of signal subspace methods used for speech enhancement in digital speech processing.

The problem is approached by initially performing an analysis of subspace principles applied to speech signals in order to characterize the usefulness of defining a signal subspace for this application. The theory is formulated by means of the singular value decomposition (or the eigendecomposition), and subspace methods are linked to filtering in the frequency domain.

Nonparametric speech enhancement using linear signal subspace based estimation of the clean signal from the noisy signal is reviewed, and connections between existing algorithms and litterature are explored. An analysis of the practical behavior of the estimators is given, and aspects regarding their performance in the case with prewhitening is covered. The relation to the popular spectral subtraction approach is discussed, and the origin of the musical noise is pointed out. A possible way to reduce the latter is devised.

In the noisy case, model based estimation is a nonlinear problem, which is normally solved by iterative techniques. However, a new idea based on multi-microphone inverse filtering is presented, where the solution is obtained by subspace methods.

The algorithm aspects of signal subspace methods are discussed in terms of the rank-revealing ULV/ULLV decompositions, which are numerically stable and can be cheaply updated, when a new data sample is present. The potential of the decompositions when applied to speech problems are analyzed, and different estimation strategies are suggested. Again, the practical behavior of the estimators are analyzed.

A recursive ULLV algorithm for a so called sliding window estimation is presented, which is new in its complete treatment and implementation. Many aspects of the algorithm are discussed in details, and important considerations are pointed out. Both the ULV/ULLV algorithms and the subspace based enhancement algorithms are implemented in a MATLAB toolbox.

Throughout the thesis, the speech enhancement application illustrates the power and robustness of the subspace approach, and a number of illustrative examples are given.

Resumé (In Danish)

Emnet for denne ph.d. afhandling er underrums (subspace) baserede metoder anvendt til støjreduktion af talesignaler inden for digital signalbehandling. Der er fokuseret på teori, analyse samt algoritme overvejelser.

Indledningsvis foretages en analyse af underrumsmetoder anvendt på talesignaler for at karakterisere muligheder og begrænsninger. Teorien er formuleret ved hjælp af singulær værdi dekomposition eller egenværdi dekomposition, og underrumsprincippet er relateret til filtrering i frekvensdomænet.

Støjreduktion baseret på ikke-parametrisk lineær estimering af talesignalet ud fra det støjfyldte signal er opsummeret, og sammenhængen mellem eksisterende metoder er belyst. Estimatorernes praktiske egenskaber er undersøgt, og aspekter vedrørende forhvidtning (prewhitening) er behandlet. Underrumsmetoders relation til spektral subtraktion samt den introducerede "musical" støj komponent er diskuteret, og en oplagt metode til at reducere den sidstnævnte er angivet.

Modelbaseret estimation af tale er i det støjfyldte tilfælde et ikke-lineært problem, som normalt løses ved hjælp af iterative teknikker. En alternativ ide baseret på invers filtrering er angivet, hvor løsningen opnås ved hjælp af underrumsmetoder.

Algoritme overvejelserne tager udgangspunkt i de rangbestemmende ULV/ULLV dekompositioner, som er numerisk stabile, og hvor beregningskompleksiteten med hensyn til opdatering er rimelig. Dekompositionerne er analyseret i forbindelse med støjreduktionsproblemet, og en formulering af forskellige estimationsstrategier er angivet. Igen er esimatorernes praktiske egenskaber analyseret.

Endelig er en rekursiv ULLV algoritme baseret på et såkaldt glidende vindue (sliding window) præsenteret. Algoritmen er en komplet implementering, og en række vigtige detaljer er fremhævet. Både ULV/ULLV algoritmerne og de underrums baserede støjreduktionsalgoritmer er implementeret i MATLAB programpakker.

Et gennemgående træk i afhandlingen er, at styrken og robustheden af underrums baserede metoder illustreres ved anvendelse inden for støjreduktion af talesignaler.

Contents

Chapter 1

Introduction

At a noisy site, speech communication is affected by the presence of acoustic noise, and reduction of noise in degraded speech is an important classical problem of signal processing. The objective of achieving higher quality and/or intelligibility of noisy speech may also contribute to improved performance in other speech applications, such as speech compression and speech recognition.

The scope of this thesis is to examine the potential of using signal subspace approaches for speech enhancement, when the noise is broad-banded. This includes development and analysis of new algorithms originating in the area of numerical linear algebra. Note that the study will focus on the theoretical aspects, so little attention is paid to issues concerning a practical implementation of a real-time noise reduction system.

This chapter provides a brief introduction to noise reduction methods, and an overview of the litterature on subspace based speech enhancement. Then, an application is proposed in order to motivate the improvement of existing algorithms. Finally, the outline of the work is described, followed by the notational conventions.

1.1 Basic Aspects of Speech Enhancement

Over the past two decades, the developments in digital signal processing have resulted in a wide variety of techniques for removal of noise in degraded speech, depending on the type of the application and the characteristics of the noise. A good source of references for enhancement of noisy speech can be found in [23].

The *adaptive noise cancelling* method [125] has often been proposed for enhancement of noisy speech [2, 22, 40, 69, 96]. The dual-microphone concept can be applied successfully if there is a high coherence of the noise in the two channels while the speech is only represented in one channel. However, in a large number of practical environments both requirements can not be obtained simultaneously. If also the speech signal is presented in both (all) microphones, *inverse filtering*[1] techniques can be used to separate the speech and noise signals [97]. A more general noise reduction approach based on several microphones is *beamforming*, where the array of microphones respond to the speech signal coming from a desired direction while discriminating against noises coming from other directions [38, 45, 13, 44, 14, 122, 104]. Again, a satisfactory performance of the last two methods is based on a high correlation between the noise signals in different channels.

The above mentioned *multi-microphone* noise reduction methods are characterized by being independent of the speech statistics, but rely on the spatial coherence of the noise field. In *single-microphone* techniques, the noise reduction is obtained by speech dependent filtering,

[1] Also referred to as the blind signal separation problem.

i.e., the remaining noise is nonstationary and has an annoying noticeable tonal characteristics referred to as *musical noise* [123]. Furthermore, the noise statistics must often be known or estimated during periods of silence between utterances. It follows that a stationary assumption of the noise is necessary. In spite of these drawbacks, speech enhancement methods based on a single microphone are very popular due to their simplicity and robustness, i.e., they can be used in most noise scenarios.

Many single-microphone speech enhancement techniques are accomplished in the frequency domain, such as *spectral subtraction* [11, 12, 70, 72, 20, 77, 29, 3, 105] to attenuate the noise outside the band of perceptual importance. These methods generally eliminate the noise in the frequency domain and then recover the speech in the time domain. However, the transformation of the signal from the frequency domain to the time domain usually causes additional degradation of the enhanced speech.

Another class of noise reduction techniques are based upon *modelling* of the speech by some parameters and using the estimated parameters to synthesize the speech [71, 30, 48]. It is well-known, from many practical applications, that the clean speech can be successfully represented by an autoregressive (AR) model [125]. Thus, if the parameters of the AR model for the speech can be estimated from the noisy signal, it is possible to generate the speech by using the estimated parameters. Even though these types of procedures remove the noise, the natural sound of speech is degraded significantly during the synthesis procedure.

The speech enhancement concepts introduced here are among the most popular methods today, and a more detailed discussion of some of them will be given in the next chapter, since they are either related to the signal subspace methods adressed in this thesis, or can be combined with them.

1.2 Subspace-based Speech Enhancement

Noise reduction of speech signals based on subspace decomposition has been proposed in the following references [25, 32, 62].

The noise reduction algorithm in [25] is based on the Singular Value Decomposition (SVD), which is a robust and widely used computational tool in noise suppression techniques. From the SVD of a Toeplitz structured data matrix, the Least Squares (LS) estimate of the signal-only matrix can be obtained by neglecting the smallest singular values and finally the Toeplitz structure of the estimate is restored to identify the time samples. The problem is that the method deals only with white noise and the LS estimate is sensitive to the number of retained singular values.

In [62], a noise reduction method based on the Quotient Singular Value Decomposition (QSVD) is presented, where a prewhitening is an integral part of the algorithm. Moreover, by using a Minimum Variance (MV) estimate [82] of the signal-only matrix, the algorithm is less sensitive to the choice of retained singular values.

In [32], the proposed estimators attempt to improve both the quality of the noisy signal while minimizing any loss in its intelligibility. The focus is on the Linear Minimum Mean Square Error (LMMSE) estimation criterion, which minimizes the error between the enhanced and the clean signal. The optimal estimator in this sense is the well-known Wiener filter. However, the error signal represents both signal distortion and residual noise, which can not be simultaneously minimized. Thus, the proposed estimators [32] control the level of the perceptually harmful residual noise (musical noise), while minimizing the signal distortion.

All the referred methods are used in a single microphone system and therefore introduce both audible musical noise and signal distortion in the enhanced signal. Furthermore, they are characterized by having a high computational complexity. Thus, it is necessary to improve such algorithms before use in any practical noise reduction application.

1.3 Motivation

One application for speech enhancement is digital mobile radio-communication systems, e.g., the Pan-European GSM system, which is a cellular system operating in the 900 MHz band.

Figure 1.1 shows the transmit side of the GSM system (see [33] for details). The input to the *speech encoder* is a 13-bit uniform pulse code modulation (PCM) signal from either the audio part of the mobile station or the network side. The sampling rate is 8 kHz, and the speech encoder operates on input frames of 160 samples (20 ms). The *voice activity detector* determines the presence of speech, and can also be used to distinguish between speech and noise frames in a speech enhancement method.

The noise reduction algorithm can be inserted right after sampling of the microphone signal (node 2), and more than one microphone can be used as long as the interface to the speech encoder is one signal satisfying the above mentioned requirements.

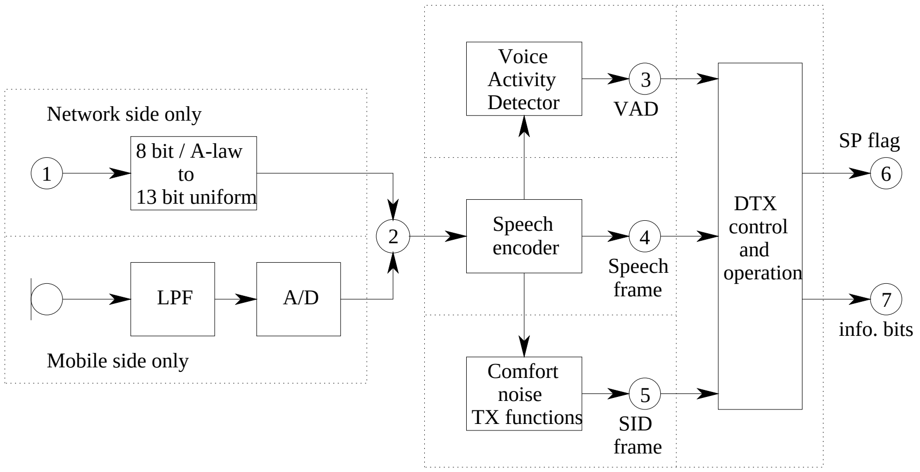


Figure 1.1 Transmit side of the GSM system.

The need for noise reduction in the GSM system is, e.g., encountered in hands-free operation[2] of mobile telephones in cars, where the microphone and loudspeaker are located remotely. The large distance between speaker and microphone implies a requirement for increased sensitivity of the microphone and the speech communication will be affected by the presence of acoustic noise in the car cabin.

The ambient noise is mainly due to the engine, traffic and wind and the Signal-to-Noise Ratio (SNR) is low. As discussed later, the broad-banded component of the noise is spatial uncorrelated, i.e., noise reduction methods based on a single microphone like the subspace approaches may be used successfully. However, note that this is only one possible application and the work presented is more general.

[2]Required by safety regulations in some country's.

1.4 Contents and Outline

The thesis is organized as follows. In Chapter 2, fundamental concepts in speech processing are introduced, and the most popular noise reduction methods are discussed in order to evaluate the subspace approaches addressed in this thesis.

Chapter 3 covers basic aspects of subspace principles with focus on speech enhancement applications, i.e., the theory is illustrated by speech related examples. The most important assumption in signal subspace methods is that the correlation matrix of the clean speech signal is rank deficient. However, even if this is not satisfied the analysis, performed here, of subspace principles applied to speech signals confirm that it is still meaningful to define a signal subspace for speech. The argument is that projection onto the signal subspace can be interpreted as filter oprations, removing the spectral components of the noisy spectrum having the lowest Signal-to-Noise Ratio (SNR).

In Chapter 4, nonparametric speech enhancement using signal subspace methods are considered. The focus is on linear estimation of the clean speech from the noisy signal under the assumption that the noise is additive and uncorrelated with the speech signal, i.e., the approach is based on some filtering techniques in the time domain.

The contribution here, is a unified presentation of the discussed estimation methods, formulated by means of both the eigendecomposition of correlation matrices and the SVD of data matrices. Relations between the different estimation methods are pointed out, and comparisons provide information on the improvement in the enhanced speech quality that can be gained with each estimator. Finally, the practical behavior of the estimators are analyzed and compared with the optimal ones.

As mentioned previously, the main problem in single-microphone techniques is the introduced musical noise. Thus, the origin of the musical noise is pointed out, and assuming an incoherent noise field, the introduction of noise reduction methods based on a single microphone in each channel of a simple delay-and-sum beamformer is shown here to be an efficient way to reduce (eliminate) the musical noise.

Model based estimation is discussed in Chapter 5, where the nonlinear problem is normally solved by iterative techniques. However, a new idea based on multi-microphone inverse filtering is presented, where the solution is obtained by subspace methods.

Subspace algorithms are normally implemented by means of the SVD/QSVD, which are computationally expensive and resists updating. In Chapter 6, subspace based algorithms are discussed in terms of the rank-revealing ULV/ULLV decompositions, which are numerically stable and can be updated cheaply. The potential of the decompositions when applied to speech problems are analyzed, and proposed formulations of different estimation strategies are given. Again, the practical behavior of the estimators are analyzed and compared with the optimal ones.

In Chapter 7, a recursive ULLV algorithm for a sliding window is presented, which is new in its complete treatment and implementation. Many aspects of the algorithm are discussed in details, and important considerations are pointed out.

Chapter 8 covers speech enhancement experiments, which illustrates the power and robustness of the subspace approach, and concluding remarks are given in Chapter 9. Finally, Appendices contains software packages and papers related to the Ph.D. study.

List of Figures and Tables, Bibliography and Index are included at the end of the thesis. All publications referred to in the text are linked to the Bibliography by a number n and will appear as $[n]$, sometimes after the name(s) of the author(s).

1.5 Notation and Terminology

Symbols and operators used in the text are (at least) introduced the first time they appear. However, the basic style conventions are only presented here.

The set of real and complex numbers will be designated $\mathbb{R}$ and $\mathbb{C}$, respectively. The space of all m-dimensional column vectors with elements in $\mathbb{R}$ will be denoted by $\mathbb{R}^m$, and the set of all $m \times n$ matrices with elements in $\mathbb{R}$ will be denoted by $\mathbb{R}^{m\times n}$. Lowercase and uppercase boldface letters are used to indicate vectors and matrices, respectively

$$\mathbf{x} \in \mathbb{R}^m \quad \text{and} \quad \mathbf{X} \in \mathbb{R}^{m\times n} \tag{1.1}$$

where x_i refers to the ith component of vector $\mathbf{x}$, and x_{ij} refers to the (i,j)th entry of matrix $\mathbf{X}$. Furthermore, the ith column of $\mathbf{X}$ will be denoted $\mathbf{x}_i$ and its jth component $x_{i,j}$.

The zero vector or matrix will be written $\mathbf{0}$. The identity matrix will be written $\mathbf{I}$ and its ith column $\mathbf{e}_i$. The transpose of a matrix $\mathbf{X}$ will be written $\mathbf{X}^T$ and its inverse $\mathbf{X}^{-1}$. In the complex case, conjugate transposition (Hermite transposition) will be written $\mathbf{X}^H$.

The following operators are assumed known: $|\cdot|$ denotes the magnitude of the scalar enclosed within, $\|\cdot\|_n$ denotes the n-norm of the vector or matrix enclosed within, $\kappa_n(\cdot)$ denotes the n-norm condition number of the matrix enclosed, the statistical expectation operator is designated by $E\{\cdot\}$, the span of vectors is denoted by either $\mathrm{span}\{\cdot\}$ or $\langle\cdot\rangle$, a diagonal matrix is denoted $\mathrm{diag}(\cdot)$, and the trace, rank, determinant, range and null space of a matrix are denoted by $\mathrm{tr}(\cdot)$, $\mathrm{rank}(\cdot)$, $\det(\cdot)$, $\mathrm{range}(\cdot)$ and $\mathrm{null}(\cdot)$, respectively.

The estimate of a scalar, vector, or matrix is designated by the use of a hat ($\hat{\ }$) placed over the pertinent symbol, and underscoring is used to distinguish a quantity as random[3], but only when it is not clear from the context wether it is random or not. Fourier transform pairs are marked as $(\cdot \leftrightarrow \cdot)$, and convolutions are denoted by the $(*)$ operator.

Finally, all numerical experiments have been performed in MATLAB, i.e., the machine precision denoted by μ is 2.22×10^{-16} .

[3]In this text, the words "random" and "stochastic" are synonyms.

Chapter 2

Speech Signals and Noise Reduction

In this chapter, some important concepts in digital speech processing are considered with focus on noise reduction methods. The review is intended to serve as a convenient reference for later chapters and to establish the notation that will be used throughout this thesis.

First, discrete time representation and sampling of speech signals is considered, followed by a summary of stochastic process theory, which is the basis for analytical techniques used in speech processing.

Then a presentation of the fundamentals of speech production is given in order to analyze and model speech, i.e., the anatomy of the speech production system can be used to explain time and frequency characteristics of short-time speech segments.

At a noisy site, speech communication is affected by the presence of acoustic noise, and speech enhancement methods attempt to improve the quality of the speech by removing the noise. It is often necessary to make assumptions about the noise mainly based on the application. The noise can, e.g., be a broadbanded stochastic signal or a harmonic component. The former is considered here, where a possible application is hands-free microphone systems inside cars.

At the end of the chapter, the performance and limitations of several noise reduction methods are discussed in order to evaluate the subspace methods addressed in this thesis.

2.1 Sampling

A *discrete time signal*, e.g., noisy speech, consisting of K data points will be indexed by integers as given by the vector

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \cdots & x_K \end{pmatrix}^T \tag{2.1}$$

and it represents samples of an analog waveform $x_a(t)$ at some sample period T_s, i.e.,

$$x_k = x_a(kT_s) = x_a(t)|_{t=kT_s}\,, \qquad k = 1, 2, \ldots, K \tag{2.2}$$

The reciprocal of T_s is referred to as the sampling frequency f_s, and the bandwidth of the signal should be below the *Nyquist frequency* $f_s/2$ in order to avoid aliasing. If this is not the case, prefiltering with an analog *anti-aliasing* filter is required.

Such a situation occurs in processing speech signals, where only the low-frequency band up to about 3–4 kHz is required for intelligibility, even though the speech signal may have significant frequency content in the 4–8 kHz range. Also even if the signal is naturally bandlimited, wideband additive noise may fill in the higher-frequency range, and as a result of sampling, these noise components will be aliased into the low-frequency band. In, e.g., telephone applications as considered here, the standard sampling frequency is 8 kHz and the speech bandwidth is usually 3.4 kHz.

2.2 Stochastic Processes

As explained in Section 2.3, there are two basic classes of speech sounds, *voiced* and *unvoiced*. The former is characterized by deterministic acoustic waveforms, while the latter corresponds to stochastic waveforms. Although random process theory will be necessary to analyze unvoiced signals only, it will in general be very useful to employ analytical techniques which are fundamentally motivated by stochastic process theory, e.g., the autocorrelation function.

A discrete *random*, or *stochastic*, process $\underline{x}$ (notice the underline notation) is defined as a collection of random variables, each indexed by a point in discrete time. For an observation interval of K samples, the following ordered random vector is obtained

$$\underline{\mathbf{x}} = \begin{pmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_K \end{pmatrix}^T \tag{2.3}$$

where each random variable $\underline{x}_k$ represents a model for the generation of values at its corresponding time k. The outcome from one experiment, i.e., a signal $\mathbf{x}$ as defined by (2.1), is called a *realization* of the random process, and the collection of all realizations is called an *ensemble*.

The *Probability Density Function* (PDF) associated with a random vector is simply the joint PDF among its component random variables

$$f_{\underline{\mathbf{x}}}(\mathbf{x}) = f_{\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_K}(x_1, x_2, \cdots, x_K) \tag{2.4}$$

and the associated stochastic process is *strictly stationary*, if the joint PDF is invariant to a shift in time. Two random vectors $\underline{\mathbf{x}} = (\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_K)^T$ and $\underline{\mathbf{y}} = (\underline{y}_1, \underline{y}_2, \cdots, \underline{y}_L)^T$ are *statistically independent* if, for any times $k \in [1; K]$ and $l \in [1; L]$, the random variabels $\underline{x}_k$ are independent of $\underline{y}_l$, i.e.,

$$f_{\underline{\mathbf{x}}\underline{\mathbf{y}}}(\mathbf{x}, \mathbf{y}) = f_{\underline{\mathbf{x}}}(\mathbf{x}) f_{\underline{\mathbf{y}}}(\mathbf{y}) \tag{2.5}$$

This is a very strong condition, where $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ may not be related in *any* functional way. When there is a linear dependency, the associated stochastic processes are always *correlated*. Thus, if they are *uncorrelated*, there is no *linear* dependence between them.

2.2.1 Expectation and Moments

Quite often, it is not possible to determine the joint PDF for a stochastic process, but it can be partially characterized by specifying the first and second moments. The *autocorrelation* function for the random process $\underline{x}$ is defined as

$$r_{\underline{x}}(k_1, k_2) = E\{\underline{x}_{k_1} \underline{x}_{k_2}\} \tag{2.6}$$

where $E\{\cdot\}$ denotes the *expectation* operator. Normally, it is assumed that the considered stochastic process $\underline{x}$ is *Wide Sense Stationary* (WSS) meaning that its autocorrelation is a function of time difference, or *lag*, η only, and its ensemble average, or mean, is constant, i.e., for any k

$$r_{\underline{x}}(\eta) = E\{\underline{x}_k \underline{x}_{k-\eta}\} \quad \text{and} \quad \mu_{\underline{x}} = E\{\underline{x}_k\} \tag{2.7}$$

For the case of two stochastic processes $\underline{x}$ and $\underline{y}$, the *cross-correlation* function is defined as

$$r_{\underline{x}\underline{y}}(k_1, k_2) = E\{\underline{x}_{k_1} \underline{y}_{k_2}\} \tag{2.8}$$

and for $\underline{x}$ and $\underline{y}$ to be *jointly* WSS, also the cross-correlation must be a function of time difference η only

$$r_{\underline{x}\underline{y}}(\eta) = E\{\underline{x}_k \underline{y}_{k-\eta}\} \tag{2.9}$$

That these statistical properties of the processes are invariant with time, is often sufficient to allow many useful analyses. Now consider an observation interval of K samples, i.e., the two random vectors $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$, then the expectation of the outer products

$$\mathbf{R}_{\underline{x}} = E\{\underline{\mathbf{x}}\,\underline{\mathbf{x}}^T\} \quad \text{and} \quad \mathbf{R}_{\underline{xy}} = E\{\underline{\mathbf{x}}\,\underline{\mathbf{y}}^T\} \tag{2.10}$$

are called the (auto)correlation matrix and cross-correlation matrix , respectively, and the elements are given by (2.7) and (2.9). The correlation matrix plays a key role in the statistical analysis of subspace methods, and it is therefore important to understand its various properties:

PROPERTIES. *The correlation matrix of a real-valued wide sense stationary stochastic process is*

- *Symmetric* $\mathbf{R}_{\underline{x}}^T = \mathbf{R}_{\underline{x}}$.
- *Toeplitz, i.e., all the elements on any (sub)diagonal are equal.*
- *Positive semidefinite, i.e.,* $\mathbf{q}^T\mathbf{R}_{\underline{x}}\mathbf{q} \geq 0$ *for all* $\mathbf{q} \neq \mathbf{0}$. *Positive definite if the matrix is nonsingular.*

Note, that the eigenanalysis in Section 3.2 is based on these facts.

In most practical experimental situations, only one realization x of the process is available. However, assuming the random process $\underline{x}$ to be wide sense stationary, it is possible to estimate $\mu_{\underline{x}}$ and $r_{\underline{x}}$ by computing *temporal averages* as

$$\hat{\mu}_x = \frac{1}{K}\sum_{k=1}^{K} x_k \quad \text{and} \quad \hat{r}_x(\eta) = \frac{1}{K}\sum_{k=1}^{K} x_k x_{k-\eta} \tag{2.11}$$

which are denoted the *sample mean* and *sample correlation*, respectively. The process is said to be *ergodic* if the estimates converge to the true values as the number of samples K approaches infinity. The last basic topic to summarize here is the *Power Density Spectrum* (PDS) of a WSS stochastic process $\underline{x}$ defined as the Discrete Fourier Transform (DFT) of its autocorrelation function

$$\Gamma_{\underline{x}}(\omega) = \sum_{\eta=-\infty}^{\infty} r_{\underline{x}}(\eta)e^{-j\omega\eta} \quad \leftrightarrow \quad r_{\underline{x}}(\eta) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Gamma_{\underline{x}}(\omega)e^{j\omega\eta}d\omega \tag{2.12}$$

Related to this definition the power in the process is given by

$$P_{\underline{x}} = E\{\underline{x}_k^2\} = r_{\underline{x}}(0) \tag{2.13}$$

Notice that since signals in speech processing are assumed to have zero mean, the correlation functions are identical with the *covariance* functions and both notations are often used.

2.3 Speech Signals

A speech signal is very often thought of as a realization of a stochastic process, where the underlying process must be assumed to have the appropriate stationarity and ergodicity properties to allow the computation of meaningful temporal statistics. However, speech is a very dynamic phenomenon and cannot possibly comprise a stationary random process, which mean that the assumption is valid only in short temporal regions. In the following, descriptions of locally stationary speech *segments* or *frames* (of period in the order of 20 ms) are denoted *short-time*, or *short-term* descriptions.

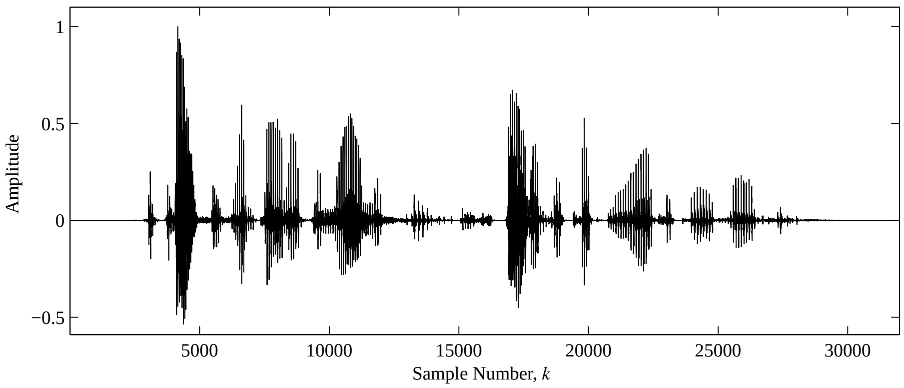


Figure 2.1 Amplitude waveform of speech sentence sampled at 8 kHz.

Figure 2.1 shows a 4 second speech waveform filtered by a 3.6 kHz analog low-pass filter and then sampled at 8 kHz. The speech signal is the result of the phonetically balanced sentence "The prices have gone up enormously in spite of the technological advances" spoken by a male speaker, and has also been used for simulations in [61]. It is obviously from the Figure that speech is a series of *steady-state* sounds with intermediate transitions, i.e., extremely nonstationary. This speech signal will be used in examples throughout this thesis and is denoted the *reference sentence*.

2.3.1 Fundamentals of Speech

In technical discussions, the entire combination of all speech production cavities is referred to as the *vocal tract* and comprises the main acoustic filter. The filter is excited by the organs below it (vocal cords, lungs, etc.) and loaded at its main output by a radiation impedance due to the lips (see Figure 2.2).

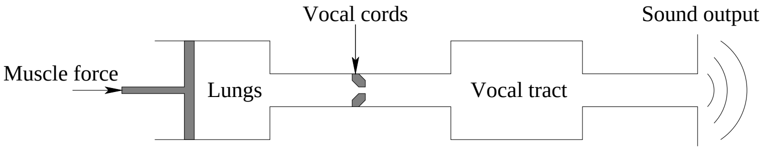
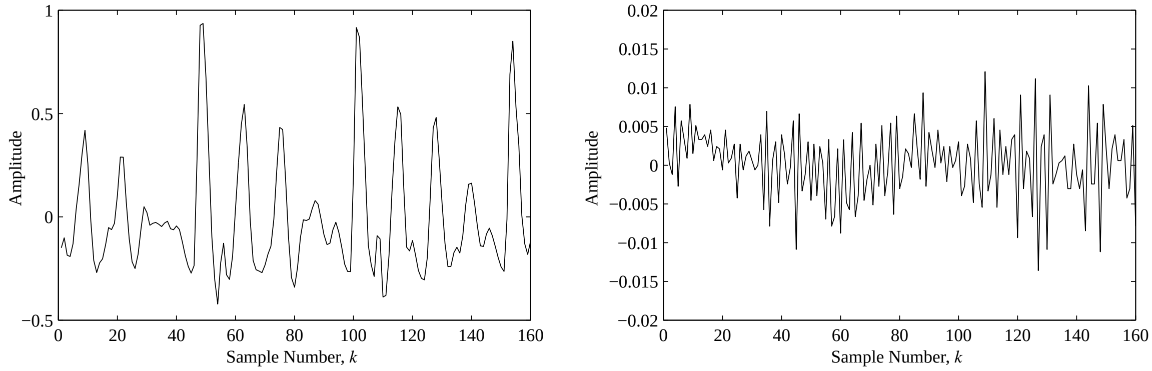


Figure 2.2 A simplified block diagram of human speech production.

Speech segments can be divided into two broad categories depending on the manner of excitation:

- A *voiced* speech sound is generated from a quasi-periodic vocal-cord sound with a *fundamental frequency* or *pitch* usually found to be below a few hundred Hertz.
- An *unvoiced* speech sound is generated from a random sound produced by turbulent airflow.

The latter involves a significant airflow restriction through the vocal tract and is therefore weaker in amplitude. A rough understanding of the two exitation types can be obtained by noting that many vowels are voiced sounds and many consonants are unvoiced sounds. However, this is not a general rule.



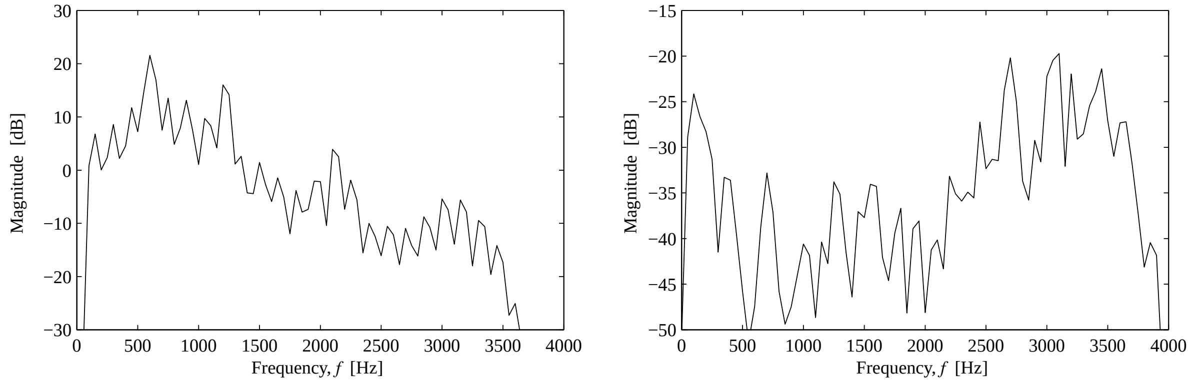
(a) (b)

Figure 2.3 (a) Amplitude waveform of voiced speech frame corresponding to the /Y/ sound in "prices". (b) Amplitude waveform of unvoiced speech frame corresponding to the /s/ sound in "spite".

Two speech frames of 20 ms (160 samples) representing each type of excitation are shown in Figure 2.3, and the above mentioned characteristics are clearly observed. The voiced frame is the /Y/ sound in "prices", and the unvoiced frame is the /s/ sound in "spite". The frames are taken from the reference sentence in Figure 2.1 (sample numbers 4161–4320 and 15701–15860, respectively), and will also be used in examples throughout the thesis.

The corresponding short-time magnitude spectra obtained by the discrete Fourier transform of the Hanning windowed frames are shown in Figure 2.4. The short length of the frames and the stochastic nature of the signals results in a large variance on the obtained spectra. However, it is still possible to observe a number of resonant frequencies, which depend upon the shape and physical dimensions of the vocal tract. These natural frequencies are referred to as *formants*, and there are typically between 3 and 5 formants in the Nyquist band after sampling.

Since speech signals are nonstationary, the spectral properties vary with time. It is possible to exhibit these changes using a three-dimensional plot of magnitude spectra over time as shown in Figure 2.5(a) for the first part of the reference sentence. The frame length is 160 samples (Hanning windowed) with an 80 sample overlap between adjacent frames. A blowup of the vocal



(a) (b)

Figure 2.4 Short-time magnitude spectra for (a) the voiced and (b) the unvoiced speech frame in Figure 2.3.

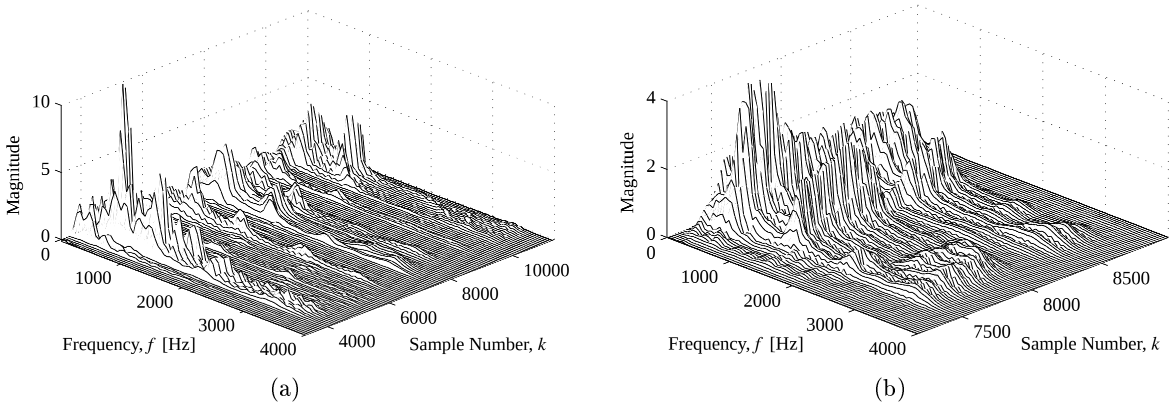


Figure 2.5 (a) Short-time magnitude spectra versus time for the first part of the speech sentence in Figure 2.1, i.e., "The prices have gone up enormously". (b) A blowup of the vocal system "gone up".

system "gone up" is shown in Figure 2.5(b), where the slowly variations of the formants can be observed. Notice that a majority of speech sounds are dominated by low frequencies.

2.3.2 Modeling Speech

The classical discrete-time model for the speech production process assumes that the sound-generating excitation is linearly separable from the intelligence-modulating vocal tract filter (see, e.g., Flanagan [37] or Deller et al. [23]). The vocal tract changes shape relatively slowly with time, and thus it can be modelled as a slowly time-varying filter which imposes its frequency-response properties on the spectrum of the excitation. Figure 2.6 shows a simplified block diagram of the model, where the wideband excitation depends on whether the speech sound is voiced or unvoiced:

- A *voiced* speech sound can be modelled by a sequence of impulses δ_k, which are spaced by a fundamental period equal to the pitch period. This signal then excites a linear filter whose impulse response equals the vocal-cord sound pulse. For simplicity, the linear filter is not shown explicitly on the figure, but is considered as a part of the vocal tract filter.
- An *unvoiced* speech sound is generated from an excitation which consists simply of a white noise source w_k. The probability distribution of the noise samples does not appear to be critical.

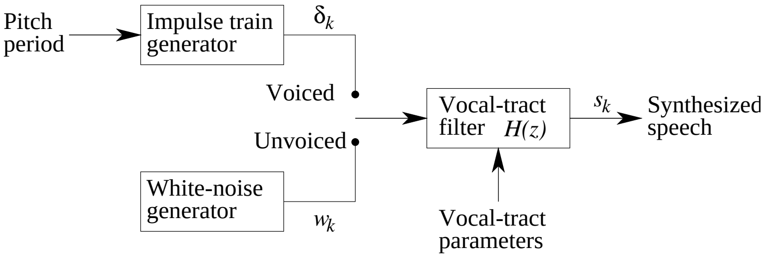


Figure 2.6 Simplified model for the speech production process.

The frequency response of the vocal tract filter $H(z)$ determines the short-time spectral envelope of the speech signal s_k and is characterized by its formants. For a majority of speech sounds, the vocal tract can be described by the transfer function of an autoregressive (AR) system

$$H(z) = \frac{b_0}{A(z)} = \frac{b_0}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{2.14}$$

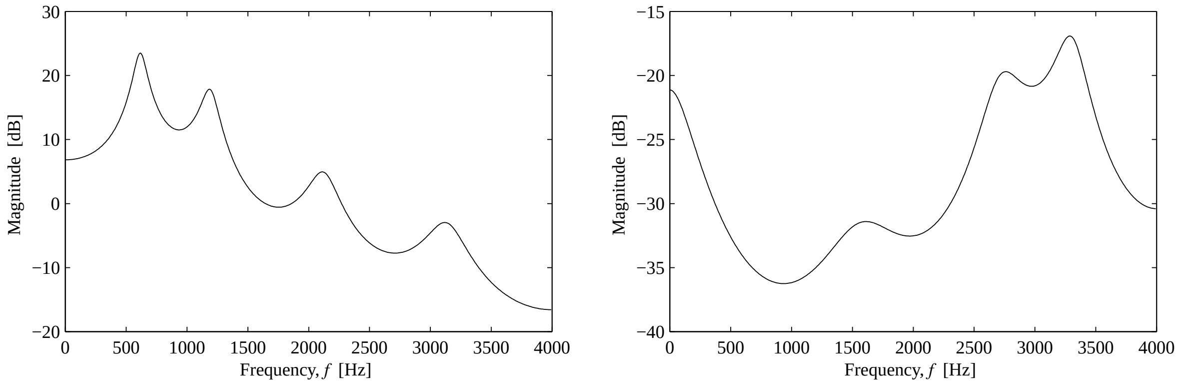
where the filter coefficients a_i are called the AR parameters or *Linear Predictive Coefficients* (LPC), p is the model order and b_0 is a gain parameter. The form of exitation applied to the AR-filter is either the sequence of delta pulses δ_k or white noise w_k corresponding to the voiced or unvoiced speech sounds, respectively

$$\text{Voiced:} \qquad s_k = \delta_k * h_k \tag{2.15}$$

$$\text{Unvoiced :} \qquad s_k = w_k * h_k \tag{2.16}$$

Thus in both cases, the excitation (input) has a flat spectral envelope. In this application the input data are real valued, hence the filter coefficients a_i are likewise real valued.

By estimating the AR parameters, i.e., the model, it is possible to exploit the short-time spectral properties of speech frames as shown in Figure 2.7, where a 10th order model is used. This model order is usually assumed to be satisfactory. Notice, that the formants are much better determined by the LPC model than by the DFT method (due to difference in degrees of freedom).



(a) (b)

Figure 2.7 10th order LPC-based magnitude spectra for (a) the voiced and (b) the unvoiced speech frame in Figure 2.3.

2.4 Noise Signals

In a noisy environment, the quality of speech is degraded, and in order to improve the clarity and intelligibility of the speech signal by a noise reduction method, the noise has to be characterized. Acoustic noise sources are normally distinguished as

- Narrow-banded, spatial localized source.
- Broad-banded, spatial localized source or diffuse[1] noise field.

[1]Spherically distributed set of uncorrelated wave fronts.

Only the broadbanded case is considered here, where the most important form of noise is *(discrete-time) white noise*, defined as a stationary stochastic process $\underline{n}$ with uniform power density spectrum $\Gamma_{\underline{n}}(f)$ over the Nyquist range

$$r_{\underline{n}}(\eta) = \nu_{noise}^2 \delta(\eta) \quad \text{and} \quad \Gamma_{\underline{n}}(f) = \nu_{noise}^2 \tag{2.17}$$

This is, e.g., obtained when the noise is drawn from a sequence of independent, identically distributed (i.i.d.) random variables. It is often assumed that the process is zero mean and Gaussian distributed due to the Central Limit Theorem[2], i.e., for K samples the joint PDF is

$$f_{\underline{n}}(\mathbf{n}) = (2\pi)^{-K/2} (\det \mathbf{R}_{\underline{n}})^{-1/2} \exp\left\{ -\frac{1}{2} \mathbf{n}^T \mathbf{R}_{\underline{n}}^{-1} \mathbf{n} \right\} \tag{2.18}$$

where $\mathbf{n} = (n_1, n_2, \cdots, n_K)^T$ is the vector of arguments.

Colored noise is a more realistic assumption in most applications and is obtained when white noise is filtered by an autoregressive moving average (ARMA) system. However, if the correlation function of the noise is assumed known, then the colored noise can always be whitened as discussed in Section 3.6.

2.4.1 Acoustic Noise in Cars

One application for speech enhancement is hands-free operation of mobile telephones inside cars as introduced in Section 1.3. In this section, the acoustic noise in a moving car is addressed, and the results will be used in examples throughout this thesis.

In [2, 22, 40, 61] a large number of measurements obtained in a car cabin under various driving conditions have been analyzed with identical conclusions. The examples given here is based on the measurements made by Jensen [61], where four omni-directional microphones were distributed in the car[3] at the locations shown in Figure 2.8. The distance between the mouth of the speaker and microphone 1 to 4 were 25 cm, 25 cm, 80 cm and 80 cm, respectively. The four channels were recorded simultaneously in order to know the spatial sound field, and sampled in accordance with the GSM specifications (8 kHz and 13 bit).

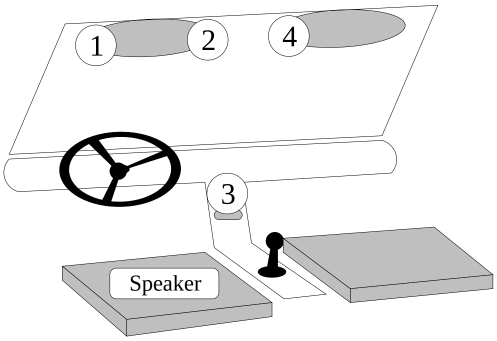


Figure 2.8 Locations of speaker and four microphones in the car cabin.

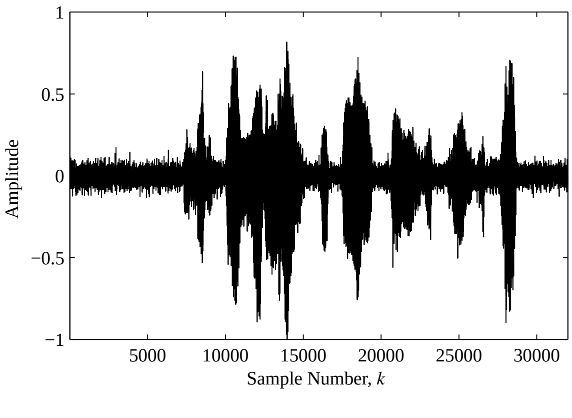


Figure 2.9 Noisy speech sentence recorded at 100 km/h by microphone 1.

[2]The Gaussian distribution is the limiting distribution for sums of random variables.
[3]Peugeot 309.

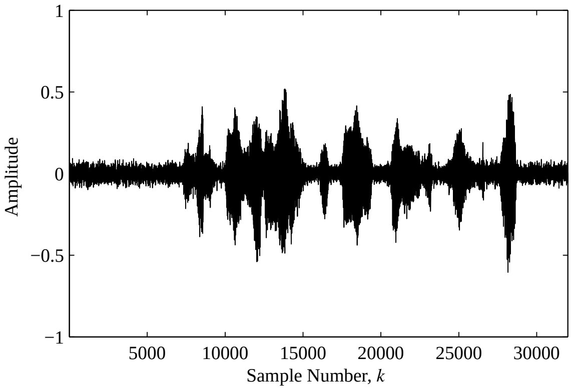


Figure 2.10 Noisy speech sentence recorded at 100 km/h by microphone 2.

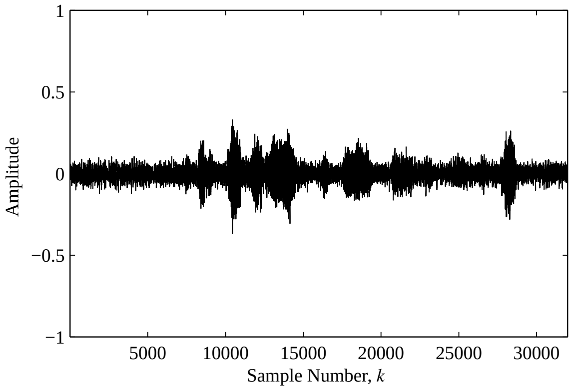


Figure 2.11 Noisy speech sentence recorded at 100 km/h by microphone 3.

Figure 2.9–2.11 shows a noisy speech sentence recorded at 100 km/h by microphone 1–3, respectively. Notice, that the speech amplitude depends on the distance between the mouth of the speaker and the microphone, while the noise level is independent of the location in the car. In the following, the noise will be described by its power spectrum whereas the noise field will be analysed by its coherence.

Power Spectral Density. Figure 2.12 shows the power spectral density of the acoustic noise picked up by microphone 3 when the car was driving 100 km/h. The PSD is calculated for a 4096 sample noise signal using Welch's method (512 sample segments, Hanning window and 256 sample overlap between adjacent segments). The plot shows a peak around 100 Hz due to the fundamental frequency of the engine, however, this noise component can easily be removed by an analog bandpass filter, since little speech information is represented at low frequencies. The broadbanded noise results mainly from the road and wind friction, and as indicated in the Figure, the noise can be modelled as a first order autoregressive process with transfer function defined as

$$\text{AR(1,-0.7):} \qquad H(z) = \frac{1}{1 + az^{-1}}\,, \qquad a = -0.7 \tag{2.19}$$

Hence, the acoustic noise in a car can be characterized as colored broadband noise.

Coherence. The *Magnitude Squared Coherence* (MSC) between two noise signals n_k and n'_k is defined by

$$\text{MSC}(\omega) = |\gamma_{nn'}(\omega)|^2 = \frac{|\Gamma_{nn'}(\omega)|^2}{\Gamma_n(\omega)\Gamma_{n'}(\omega)}\,, \qquad 0 \le \text{MSC}(\omega) \le 1 \tag{2.20}$$

where $\Gamma_{nn'}(\omega)$ is the cross spectral density, and $\Gamma_n(\omega)$ and $\Gamma_{n'}(\omega)$ are the power spectral densities. Thus, the MSC gives the percentage of signal energy coming from correlated sources for every frequency. Figure 2.13 shows the MSC[4] between the noise measured by microphone 1 and 2 (30 cm distance) at 100 km/h. There is only one peak close to one due to the engine frequency, i.e., the broadband noise is spatial uncorrelated.

[4]Calculated for noise signals consisting of 4096 samples using Welch's method (512 sample segments, Hanning window and 256 sample overlap between adjacent segments).

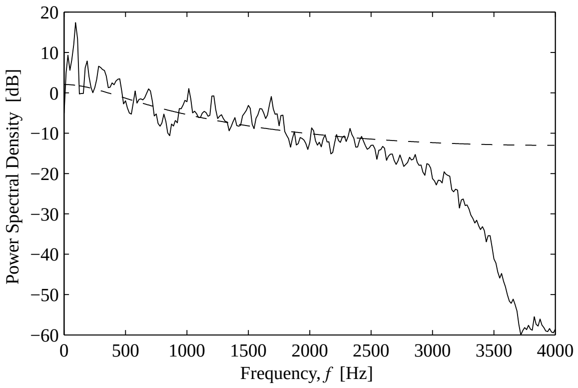


Figure 2.12 (solid) PSD of noise recorded at 100 km/h by microphone 3. (dashed) Magnitude spectrum of AR(1,-0.7) noise model.

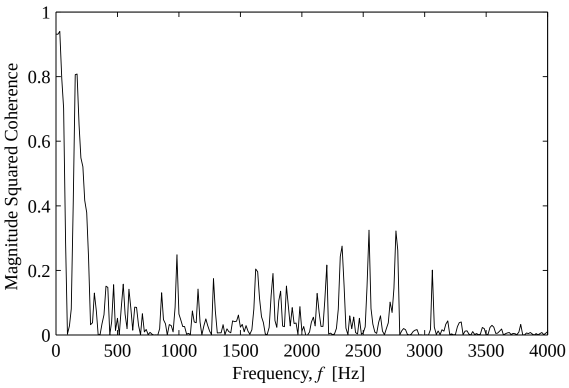


Figure 2.13 MSC between the noise measured by microphone 1 and 2 (30 cm distance) at 100 km/h.

In [22] is shown that the noise generation mechanism inside a car can be modelled as a diffuse sound field, i.e., the coherence is given by

$$\gamma_{nn'}(\omega) = \frac{\sin(2\pi f d/c)}{2\pi f d/c} \tag{2.21}$$

where d is the distance between the microphones and c is the sound velocity. For example, a $\mathrm{MSC}(\omega) > 0.7$ in the frequency band up to around 3 kHz requires a microphone distance less than 2 cm, so in practice, the acoustic noise in a car can be assumed spatial uncorrelated.

SNR. Finally, consider the Signal-to-Noise Ratio (SNR) in voiced and unvoiced speech frames. A reliable method is to add an AR(1,-0.7) noise signal to the clean speech sentence in Figure 2.1 in order to obtain amplitude waveforms similar to the noisy speech signals shown in Figure 2.9 and 2.11. The results for the voiced and unvoiced frames in Figure 2.3 are given in Table 2.1. Clearly, the voiced frame has a much better segmental SNR than the unvoiced, but notice that the chosen voiced frame belongs to the part of the sentence with highest signal power, so a voiced frame will be assumed to have a segmental SNR around 5–10 dB.

Noise level similar to	Global SNR of reference sentence	Segmental SNR of voiced frame	Segmental SNR of unvoiced frame
Figure 2.9	5 dB	18 dB	-17 dB
Figure 2.11	0 dB	13 dB	-22 dB

Table 2.1 SNR of speech sentence added an AR(1,-0.7) noise signal.

2.5 Speech Enhancement

The problem of enhancing speech degraded by additive *broadband* noise has received considerable attention in the past two decades. The objective is to achieve higher quality and/or intelligibility of noisy speech prior to processing by the auditory system. However, it has also been shown that front-end speech enhancement can be useful for other speech processing applications, such as processing before coding or recognition.

There are a number of ways in which speech enhancement systems can be classified, e.g., whether the speech is modelled based on stochastic processes or perceptual aspects. Only three broad classes are considered here, which cover the most popular methods today, and have all been applied to the problem of acoustic noise inside a car cabin, i.e.,

- Adaptive Noise Canceling
- Adaptive Beamforming
- Spectral Subtraction

In the following, the performance and limitations of each class are discussed in order to evaluate the subspace methods addressed in this thesis. Thus, adaptive noise canceling is used to demonstrate a technique that completely fails if the noise assumptions are not satisfied, while adaptive beamforming and spectral subtraction are related to theory considered in the next chapters.

2.5.1 Adaptive Noise Cancelling

Adaptive Noise Cancelling (ANC) using a linear adaptive filter has often been proposed as a method for the enhancement of noisy speech [125, 2, 22, 40, 69, 96]. The principle is illustrated in Figure 2.14, where the speech signal s_{pri} at the primary input is corrupted by some noise n_{pri}. If the noise at the reference input n_{ref} is correlated with n_{pri}, a properly designed adaptive filter is able to approximate the linear transfer function between the two microphones. Thus, the output of the adaptive filter is an estimate of the noise n_{pri}, which is then subtracted from the primary signal. A high correlation of the two noise signals will result in a low residual noise level at the output. However, the presence of speech in the reference input will cause a distortion of the speech signal at the output.

A measure of performance of ANC is the ratio between the output and primary input SNR as given by [40]

$$\eta(\omega) = \frac{\mathrm{SNR}_{out}(\omega)}{\mathrm{SNR}_{pri}(\omega)} = \frac{1}{1 - \mathrm{MSC}_{n_{pri}n_{ref}}(\omega)} \tag{2.22}$$

The problem with speech in the reference input has been analyzed by Widrow [125] (optimal filter coefficients) giving the following relation between SNRs

$$\mathrm{SNR}_{out}(\omega) = \frac{1}{\mathrm{SNR}_{ref}(\omega)} \tag{2.23}$$

Thus, to provide successful noise reduction, very high coherence with respect to the noise signals is required (see Figure 2.15) as well as sufficient decoupling with respect to the speech signal. An 5 dB noise reduction requires for example a MSC value near 0.7.

In the car application, this MSC value can only be obtained by a microphone distance of less than 2 cm, and the speech signal will therefore be represented in both inputs, i.e., the ANC scheme *can not* be applied successfully (see, e.g., the results in [2, 22, 40, 69, 96]).

2.5.2 Adaptive Beamforming

A more general noise reduction approach based on several microphones is *beamforming*, where the array of microphones respond to a signal coming from a desired direction while discriminating against noises coming from other directions [38, 45, 13, 44, 14, 122, 104].

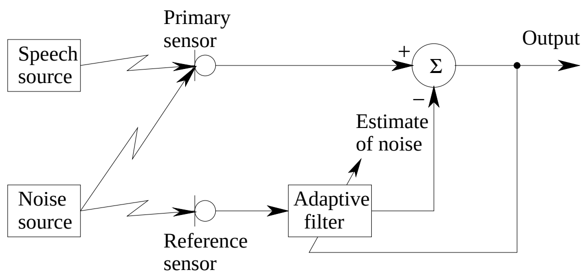


Figure 2.14 Block diagram of Adaptive Noise Canceller.

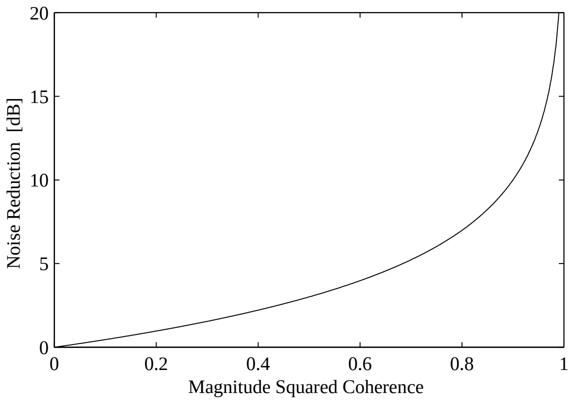


Figure 2.15 ANC based noise reduction $\eta(\omega)$ as function of MSC.

Delay-and-Sum beamforming is the simples method, where the output is just the sum of the delay-compensated multi-channel data (see Figure 2.16). Since the time-delay steering elements τ_i are adjusted so the desired speech signal will be coherently added, while the noise components will be incoherently added, the output has a higher SNR than the output from any of the individual microphones. The SNR improvement is theoretically limited to

$$\eta = 10 \log L \quad [\text{dB}] \tag{2.24}$$

with L the number of microphones. Hence, the SNR gain is not sufficient for the car application as only a small number of microphones may be used for practical reasons, e.g., four microphones correspond to 6 dB improvement.

Another issue is the determination of the time-delay steering elements τ_i. If the desired signal is a plane wave impinging on a linear array, delay-and-sum beamforming requires no prior knowledge of the signal or noise statistics. However, for unknown array geometries, alignment of the desired signal generally involves the use of numerical search algorithms, e.g., based on the cross-correlation functions.

To obtain better results, a *Linearly Constrained Minimum Variance* (LCMV) beamformer [38] can be used as illustrated in Figure 2.17 for the direct-form implementation. The constrained algorithm is used to iteratively adapt the filter weights $\mathbf{w}_i$ to minimize noise power in the array output, while the set of linear equality constraints maintains a chosen frequency characteristic for the array in the direction of interest. Note, that an equivalent implementation of the LCMV beamformer is the *Generalized Sidelobe Canceller* [45].

When the adaptive array operates in the presence of white noise only, the resulting beam pattern is referred to as the quiescent response. However, under conditions of correlated interference, the response changes so as to effectively steer nulls in the appropriate directions as indicated in Figure 2.17.

When used in a car cabin, the LCMV beamformer has two drawbacks. First, if signal components obtained outside the direction of interest are correlated with the desired signal, e.g., the reverberations in the car cabin, then the speech output will be distorted. Second, the diffuse noise field in the car cabin will result in a quiescent response similar to the Delay-and-Sum beamformer. These conclusions correspond to the results obtained in [86, 42].

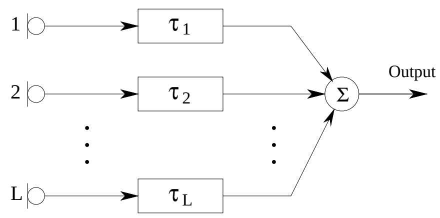


Figure 2.16 Block diagram of Delay-and-Sum beamformer.

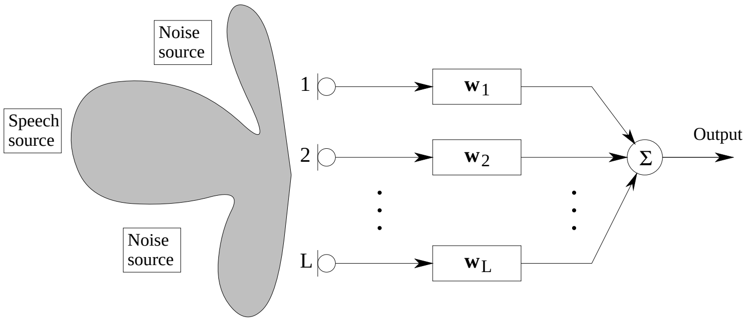


Figure 2.17 Direct-form structure for adaptive beamformer.

2.5.3 Spectral Subtraction

The spectral subtraction approach has become almost standard in speech enhancement [11, 12, 70, 72, 77, 29, 3, 105], and the name is originally motivated by the subtraction of the short-term magnitude spectrum of the noise from the short-term magnitude spectrum of the noisy signal [11]. Thus, it is a single microphone noise reduction technique based on the assumption that it is mainly the spectral magnitude rather than the phase that is important for speech intelligibility and quality. Fortunately, for all practical purposes, it is sufficient to use the noisy phase spectrum as an estimate of the clean speech phase spectrum [124].

In spectral subtraction signal estimation, the DFT is first applied to a segment of noisy speech. Spectral components whose estimated variance is smaller than or equal to the one obtained from the noise-only spectral component are nulled (half-wave rectification). The remaining spectral components are modified by a gain function and the inverse DFT is applied.

Let the matrix $\mathbf{D}^H \in \mathbb{C}^{K \times K}$ represent the DFT, then the spectral components of the noisy signal vector $\mathbf{x} \in \mathbb{R}^K$ are given by the vector $\mathbf{D}^H\mathbf{x}$. Assume that only $p < K$ components of $\mathbf{D}^H\mathbf{x}$ have variance greater than the noise level, and that those components are denoted $\mathbf{D}_1^H\mathbf{x}$, then the spectral subtraction estimator can be written as

$$\hat{\mathbf{s}}_{SPS} = \frac{1}{K} \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{pmatrix} \begin{pmatrix} \mathbf{G}_{SPS} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{D}_1^H \\ \mathbf{D}_2^H \end{pmatrix} \mathbf{x} \tag{2.25}$$

where $\mathbf{G}_{SPS}$ is an diagonal gain matrix, and $(\mathbf{D}_1 \; \mathbf{D}_2)$ is obtained from $\mathbf{D}$ by rearranging the columns.

Many approaches have been proposed for the evaluation of the short-time gain factor. This factor is adjusted individually on each frequency as a function of the local signal-to-noise estimation. Such methods include magnitude spectral subtraction [11], power spectral subtraction, Wiener filtering [72], soft-decision estimation [77] and Minimum Mean Square Error (MMSE) estimation [29].

When the Wiener gain function is used, the mth diagonal element of $\mathbf{G}_{SPS}$ is given by

$$g_{SPS}(m) = \frac{\hat{\Gamma}_x(m) - \hat{\Gamma}_n(m)}{\hat{\Gamma}_x(m)} \tag{2.26}$$

where $\hat{\Gamma}_x(m) = |\mathbf{D}^H\mathbf{x}|_m^2/K$ and $\hat{\Gamma}_n(m) = |\mathbf{D}^H\mathbf{n}|_m^2/K$ estimates the variance of the mth spectral component of the noisy signal and the noise process, respectively, i.e., they are estimates (periodograms) of the power spectral densities. Note that the noise PSD is found during nonspeech activity.

The gain function used in power spectral subtraction is obtained by taking the square root of Equation (2.26), while the MMSE estimator is based on soft-decisions aspects taking into account the uncertainty of speech presence in the noisy observations, i.e., speech distortion and musical noises should be reduced.

The optimal and practical behavior of the MMSE and Wiener filter is analyzed in [98], where experiments indicate[5] that the MMSE suppression rule always gives an enhanced speech signal which presents more distortion than the Wiener one. The main reason for this is due to the estimation of the filter parameters, where assumptions in the MMSE model is not really satisfied. The conclusion is that the Wiener filter can be used for most practical applications.

2.5.4 Speech Quality Assessment

All listeners have an intuitive understanding of speech quality, however, this is not easy to quantify in most speech enhancement applications, since it is based on subjective evaluation of the processed signal.

Noise reduction is normally discussed in terms of Signal-to-Noise Ratio (SNR), but it is important to note that this may not be the most appropriate performance criterion for speech enhancement. An overview of a large number of other speech quality measures can be found in [23, Page 595].

In the evaluation of speech enhancement methods, it is maybe even more important to understand the theoretical and practical properties of the different estimation principles. Thus, from such analyzes together with a basic knowledge of the human auditory system, it is possible to design noise reduction systems capable of improving both the quality and intelligibility of the noisy signal.

In this thesis, the analysis approach combined with SNR measures will be used as quality assessment tools. The main reason is that the SNR measure is well-known by most readers, while the quantities obtained by other speech quality measures are harder to evaluate without prior knowledge and experience.

In noise reduction systems, the original speech signal is needed to determine SNR improvements. Let x_k denote the noisy speech signal, s_k its noise-free equivalent, and $\hat{s}_k$ the corresponding enhanced signal, then the resulting SNR measure (in dB) is obtained as

$$\mathrm{SNR} = 10 \log \left(\frac{\sum_k s_k^2}{\sum_k (s_k - \hat{s}_k)^2} \right) \quad [\mathrm{dB}] \tag{2.27}$$

Several variations of SNRs will be used. The *global* SNR denotes measures calculated from longer signals like a sentence, while *segmental* SNR is obtained from a single frame. Finally, *spectral* SNR is obtained within a frequency band of a frame.

2.6 Summary

The non-stationarity of speech signals has been discussed, and short-time speech segments have been characterized in the time and frequency domain. The properties have been illustrated by means of a single phonetically balanced sentence, and by use of one voiced and one unvoiced segment. The same examples will be used in simulations throughout the thesis to illustrate and analyze the theory, and it should be noted that the conclusions drawn from these few examples are fairly general. However, other speech material has also been used to verify the results.

[5] Evaluated by cepstral and basilar distances.

Speech signals are characterized as broad-banded, so reduction of broad-banded, acoustic noise in speech signals is often a difficult problem to solve. One example, considered here, is the noise[6] inside a car cabin, which is colored, spatial uncorrelated and has distributed origin. Note, that the global SNR is often low (0–10 dB), and that this type of noise is also found in other environments.

Several speech enhancement principles have been summarized, and the conclusion is that noise reduction based on multi-microphone techniques typically requires spatial correlated noise. Thus, in the car application, the approximately best enhancement is obtained by the simple delay-and-sum beamformer, which is not sufficient.

In the single-microphone case, the popular spectral subtraction method has almost become the standard today. However, it is a frequency domain approach and suffers from the fact that speech signals are extremely non-stationary.

The subspace methods addressed in this thesis can be considered as a time domain version of the spectral subtraction approach. Thus, the obtained estimators can be expected to be closer to the optimal solutions.

[6] Resulting from road- and wind friction.

Chapter 3

Signal Subspace Methods

In the last two decades, signal subspace methods have been used frequently in digital signal processing in connection with, e.g., spectrum estimation [65, 116], system identification [66, 121] and direction of arrival problems [103, 5, 68]. However, the signal subspace approach has only recently been used in digital speech processing [25, 32, 61, 62].

The most important assumption in signal subspace methods for speech enhancement is that the correlation matrices of vectors of the clean speech signal need not be positive definite, i.e., they have some eigenvalues which are practically zero. This can be observed by either examining the empirical correlation matrices of the speech signal or by studying the commonly used linear model for that signal [94, 95]. This fact indicates that the energy of the clean signal vector is distributed among a subset of its coordinates, and the signal is confined to a subspace.

If the correlation matrix of an additive noise is assumed positive definite, i.e., all noise eigenvalues are strictly positive, then the noise vectors span the entire space. Thus, the noise components in the subspace complementary to the signal subspace can be removed without degrading the clean speech signal.

The decomposition of the vector space of the noisy signal can be performed by applying the eigendecomposition to the correlation matrix. However, the second-order statistics are estimated from a number of noisy vectors, so a better approach is to organize the vectors in a data matrix and use the singular value decomposition.

In this chapter, the basic signal subspace principles are introduced with focus on speech enhancement applications. Thus, whenever possible the theory will be illustrated by speech related examples, and general assumptions will be shown to be valid for this application.

3.1 Linear Model

A speech signal is nonstationary, but can in a short-time window (typically 10 to 30 ms long) be considered as a wide sense stationary stochastic process. The speech signal can therefore be represented by a *linear stochastic model* of the form

$$\mathbf{s} = \mathbf{H}\boldsymbol{\theta} = \sum_{i=1}^{p} \mathbf{h}_i \theta_i \tag{3.1}$$

where $\mathbf{s} = (s_1, s_2, \cdots, s_m)^T$ is a sequence of signal random samples, $\mathbf{H} \in \mathrm{I\!R}^{m \times p}$ is a model matrix, and $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_p)^T$ is a zero mean random coefficient vector drawn from a multivariate distribution. The linear model (3.1) is fairly general. It includes, for example, the *damped complex sinusoid* model, which has often been attributed to speech signals (see Section 5.2). However, in this chapter, the exact form of the linear model need not be specified.

In the linear model the columns $\mathbf{h}_i$, or modes, of $\mathbf{H}$

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_p \end{pmatrix} \tag{3.2}$$

span a *signal subspace* $\langle \mathbf{H} \rangle = \text{span}\{\mathbf{h}_1, \ldots, \mathbf{h}_p\}$. Assuming that the columns of $\mathbf{H}$ are linearly independent, i.e., $\mathbf{H}$ has full rank, then the dimension of the signal subspace is $p \leq m$. Thus, when $p < m$ the set of all possible signal vectors $\{\mathbf{s}\}$ is constrained to be in the signal subspace, which can be used for signal enhancement in the case of noise perturbations. As discussed above, such representation is always possible for speech signals.

As an illustrative example, consider the cosinusoid signal $s_k = A\cos(\omega k T_s - \phi)$, which can be expanded as $A\cos(\omega k T_s)\cos(\phi) + A\sin(\omega k T_s)\sin(\phi)$. Then a m-sample vector $\mathbf{s}$ may be written as

$$\mathbf{s} = \begin{pmatrix} 1 & 0 \\ \cos(\omega T_s) & \sin(\omega T_s) \\ \vdots & \vdots \\ \cos(\omega(m-1)T_s) & \sin(\omega(m-1)T_s) \end{pmatrix} \begin{pmatrix} A\cos(\phi) \\ A\sin(\phi) \end{pmatrix} = \mathbf{H}\boldsymbol{\theta} \tag{3.3}$$

Thus, for random phase ϕ, a harmonic component lies in a *real* linear vector space of dimension two. For a sum of $p/2$ cosinusoids with $\omega_i \neq \omega_j$, the matrix $\mathbf{H}$ consists of linearly independent sines and cosines which span a subspace of dimension p.

From the linearly independent modes that span the signal subspace $\langle \mathbf{H} \rangle$, an orthogonal complement subspace $\langle \tilde{\mathbf{H}} \rangle$ of dimension $m - p$ can be constructed

$$\tilde{\mathbf{H}} = \begin{pmatrix} \tilde{\mathbf{h}}_1 & \tilde{\mathbf{h}}_2 & \cdots & \tilde{\mathbf{h}}_{m-p} \end{pmatrix} \tag{3.4}$$

Taken together, the subspaces $\langle \mathbf{H} \rangle$ and $\langle \tilde{\mathbf{H}} \rangle$ span the Euclidean m-space $\mathbb{R}^m$. The subspace $\langle \tilde{\mathbf{H}} \rangle$ is called the *noise subspace*, but it should be emphasized that the noise typically fills in the entire Euclidean space. Thus, the signal subspace contains vectors of the pure speech signal as well as of the noise process. The orthogonal subspace contains vectors of the noise process only. In our context, it is assumed that the noise process satisfy the following conditions

ASSUMPTION 3.1 (*Basic Noise Assumptions*)

1. *The noise is zero mean, additive and broadbanded (positive definite correlation matrix).*

Consider now the noisy random vector $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T$ of dimension m, such that

$$\mathbf{x} = \mathbf{s} + \mathbf{n}\,, \qquad \mathbf{s} = \mathbf{H}\boldsymbol{\theta} \tag{3.5}$$

where $\mathbf{s}$ contains the pure speech component that lies in the subspace $\langle \mathbf{H} \rangle$ and $\mathbf{n}$ represents the noise that fills up the Euclidean space.

Thus, the speech signal is known to lie in a subspace of rank p, however the subspace is unknown. The noise reduction problem is to *estimate* the subspace $\langle \mathbf{H} \rangle$, i.e., its dimension and a suitable basis, determine how much of the measurements lies within it, and use this information in a signal processing procedure. Note, that it is in general not possible to find the *exact* signal subspace $\langle \mathbf{H} \rangle$.

3.1.1 Data Matrix

The space spanned by the model matrix $\mathbf{H}$ can be exploited by considering a sequence of n realizations of the random vector $\mathbf{x}$ as defined by (3.5). The set of noisy signal vectors is used to define the *data matrix* $\mathbf{X} \in \mathbb{R}^{m\times n}$

$$\mathbf{X} = \mathbf{S} + \mathbf{N}\,, \qquad \mathbf{S} = \mathbf{H}\boldsymbol{\Theta} \tag{3.6}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_n) \in \mathbb{R}^{p\times n}$ is the coefficient matrix and the signal matrices are given by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n)$ and $\mathbf{N} = (\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_n)$.

In Equation (3.6), the number of realizations should be much larger than the dimension of the vectors, i.e., $n \gg m$, in order to reveal the subspace information. However, numerical algorithms and notation normally assume $n \leq m$, so whenever this is done, the *rows* of the data matrix are assumed to be realizations of the random vector $\mathbf{x}$.

One way to construct the data matrix from a realization consisting of K samples

$$\mathbf{x}_K = \begin{pmatrix} x_1 & x_2 & \cdots & x_K \end{pmatrix} \tag{3.7}$$

is to organize a set of *time shifted* vectors in a $m \times n$ data matrix with Toeplitz structure

$$\mathbf{X} = \mathcal{T}_n(\mathbf{x}_K) = \begin{pmatrix} x_n & x_{n-1} & \cdots & x_1 \\ x_{n+1} & x_n & \cdots & x_2 \\ \vdots & \vdots & & \vdots \\ x_K & x_{K-1} & \cdots & x_{K-n+1} \end{pmatrix} = \mathbf{S} + \mathbf{N} \tag{3.8}$$

where the Toeplitz operator $\mathcal{T}_n$ has subscript n showing the number of columns and where the matrix dimensions are constrained by $K = m + n - 1$. In the case of speech signals, a good choice of (m, n) is $(141, 20)$ corresponding to $K = 160$ (see the discussion in Section 3.4.2). Note that one can also choose to work with Hankel matrices instead of Toeplitz matrices. There is no fundamental difference between these two approaches.

Let $n \leq m$, then $\text{rank}(\mathbf{N}) = n$ due to the broadband assumption and also $\text{rank}(\mathbf{X}) = n$. For the moment, however, no further statistical assumptions are made on $\mathbf{N}$, but it is simply treated algebraically as a matrix of full rank. Because the speech model is based upon p linearly independent modes with $p < n$, i.e., a low order model, the matrix $\mathbf{S}$ will be rank deficient

$$\text{rank}(\mathbf{S}) = p < n \leq m \tag{3.9}$$

and the matrix has $(n - p)$ zero singular values. As discussed in Section 3.4, this observation can be used to estimate the clean signal from the noisy data matrix using the singular value decomposition.

3.1.2 Correlation Matrix

The discussed signal vectors are realizations of stochastic processes, which mean that the analysis of subspace methods is based on correlation matrices.

First, consider the correlation matrix $\mathbf{R}_s \in \mathbb{R}^{m\times m}$ of the noise-free random vector $\mathbf{s}$ defined by the linear model (3.1), i.e.,

$$\mathbf{R}_s = E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{H}\mathbf{R}_\theta\mathbf{H}^T\,, \qquad \mathbf{R}_\theta = E\{\boldsymbol{\theta}\boldsymbol{\theta}^T\} \tag{3.10}$$

Hence, the rank of $\mathbf{R}_s$ is p, and this matrix has $(m - p)$ zero eigenvalues. Similarly, let the correlation matrix of the noise vector be denoted by $\mathbf{R}_n = E\{\mathbf{n}\mathbf{n}^T\}$, which has full rank. When considering second order statistics, it is useful to assume

ASSUMPTION 3.2 (*Correlation Based Noise Assumptions*)

1. *The elements of* $\mathbf{s}$ *and* $\mathbf{n}$ *are uncorrelated:* $\mathbf{R}_{sn} = \mathbf{R}_{ns} = \mathbf{0}$.
2. *The noise is white with variance* ν_{noise}^2*:* $\mathbf{R}_n = \nu_{noise}^2 \mathbf{I}_m$.

The last assumption is based on the fact that the correlation matrix of the noise vector is assumed known, i.e., colored noise can always be whitened as discussed in Section 3.6. Thus, the correlation matrix of the noisy vector (3.5) is given by

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{R}_s + \mathbf{R}_n = \mathbf{H}\mathbf{R}_\theta\mathbf{H}^T + \nu_{noise}^2\mathbf{I}_m \tag{3.11}$$

Obviously, the noise power is uniformly distributed in the entire Euclidean space, while the speech signal is constrained to p dimensions.

In practice, the exact knowledge of the second-order statistics is not available, but is estimated from the noisy signal. Assuming stationary and ergodic conditions, the estimate $\hat{\mathbf{R}}_x$ of the correlation matrix can be obtained from the Toeplitz data matrix $\mathbf{X}$ (3.8) as either

$$\hat{\mathbf{R}}_x = \frac{1}{n}\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{m\times m} \quad \text{or} \quad \hat{\mathbf{R}}_x = \frac{1}{m}\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{n\times n} \tag{3.12}$$

which converge to the true correlations

$$\mathbf{R}_x = \lim_{n\to\infty}\frac{1}{n}\mathbf{X}\mathbf{X}^T \quad \text{or} \quad \mathbf{R}_x = \lim_{m\to\infty}\frac{1}{m}\mathbf{X}^T\mathbf{X} \tag{3.13}$$

The estimate $\hat{\mathbf{R}}_x$ obtained from empirical data is also referred to as the *sample correlation matrix.*

3.2 The Symmetric Eigenvalue Problem

Decomposition of the noisy Euclidean space into the signal subspace and noise subspace can be performed by applying the eigendecomposition to the correlation matrix of the noisy signal.

Symmetry simplifies the real eigenvalue problem $\mathbf{R}\mathbf{q} = \lambda\mathbf{q}$ in two ways. It implies that all of $\mathbf{R}$'s eigenvalues λ_i are *real* and it implies that there is an *orthonormal* basis of eigenvectors $\mathbf{q}_i$. These properties are a consequence of [39, page 410]

THEOREM 3.1 (*Symmetric Real Schur Decomposition*) *If* $\mathbf{R} \in \mathbb{R}^{m\times m}$ *is a symmetric matrix, then there exists a real orthogonal matrix of eigenvectors*

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_m \end{pmatrix} \in \mathbb{R}^{m\times m} \tag{3.14}$$

such that

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \sum_{i=1}^{m}\lambda_i\mathbf{q}_i\mathbf{q}_i^T \tag{3.15}$$

where $\mathbf{\Lambda}$ *denotes a diagonal matrix of eigenvalues*

$$\mathbf{\Lambda} = \operatorname{diag}(\lambda_1,\ldots,\lambda_m) \in \mathbb{R}^{m\times m}, \qquad \lambda_1 \geq \ldots \geq \lambda_m \geq 0 \tag{3.16}$$

PROOF. For the proof, see [39, page 410]. □

Equation (3.15) is called an *Eigendecomposition* of $\mathbf{R}$ or that $\mathbf{R}$ is *orthogonal similar* to the diagonal matrix $\mathbf{\Lambda}$. The set of eigenvalues is called the *eigenvalue spectrum.*

Now, let the eigendecomposition of the rank-p correlation matrix $\mathbf{R}_s$ (3.10) be given by

$$\mathbf{R}_s = \mathbf{Q}\mathbf{\Lambda}_s\mathbf{Q}^T = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda}_{s1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{pmatrix} \tag{3.17}$$

where $\mathbf{Q}_1 \in \mathbb{R}^{m\times p}$ and $\mathbf{\Lambda}_{s1} \in \mathbb{R}^{p\times p}$. The correlation matrix $\mathbf{R}_n = \nu_{noise}^2\mathbf{I}_m$ of the additive white noise process has a single degenerate eigenvalue equal to the variance ν_{noise}^2 with multiplicity m, so any vector qualifies as the associated eigenvector, i.e., the eigenvectors $\mathbf{Q}$ are unaffected by the constant diagonal perturbation, and the eigendecomposition of $\mathbf{R}_x$ (3.11) is obtained as

$$\mathbf{R}_x = \mathbf{Q}\mathbf{\Lambda}_x\mathbf{Q}^T = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda}_{s1} + \nu_{noise}^2\mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \nu_{noise}^2\mathbf{I}_{m-p} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{pmatrix} \tag{3.18}$$

Here, the *principal* eigenvectors $\mathbf{Q}_1$ associated with the p largest eigenvalues span the signal subspace $\langle\mathbf{H}\rangle$ and $\mathbf{Q}_2$ span the noise subspace. Note, that the signal-only correlation matrix $\mathbf{R}_s$ can be recovered consistently from this equation, and the power in the process can be obtained from the eigenvalues as

$$r_x(0) = \frac{1}{m}\mathrm{tr}(\mathbf{R}_x) = \frac{1}{m}\mathrm{tr}(\mathbf{\Lambda}_x) \tag{3.19}$$

Thus, the projection of the vector $\mathbf{x}$ onto the signal subspace, i.e., $\mathbf{Q}_1\mathbf{Q}_1^T\mathbf{x}$, improves the SNR by a factor n/p, and the concept is called *Principal Component Analysis* (PCA).

This can be illustrated by a simple example, where a sinusoid ($p = 2$) with unit power is contaminated by white noise (SNR = 5 dB). The eigenvalue spectrum of $\mathbf{R}_x \in \mathbb{R}^{20\times 20}$ is shown in Figure 3.1, and the power spectral densities[1] of the noisy signal and the projection onto the signal subspace ($p = 2$) is shown in Figure 3.2. Clearly, the sinusoid is untouched, while 90 percent of the noise has been removed.

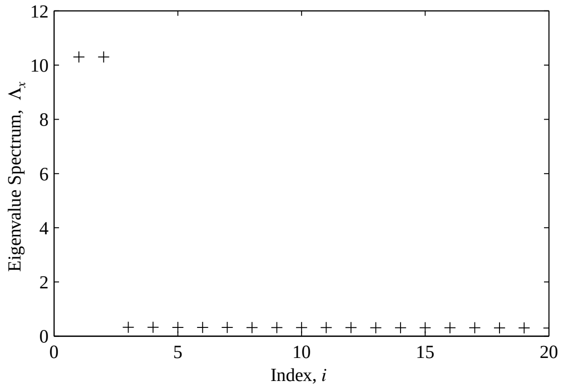


Figure 3.1 The eigenvalues of $\mathbf{R}_x \in \mathbb{R}^{20\times 20}$ representing a sinusoid with unit power, and added white noise (SNR=5dB).

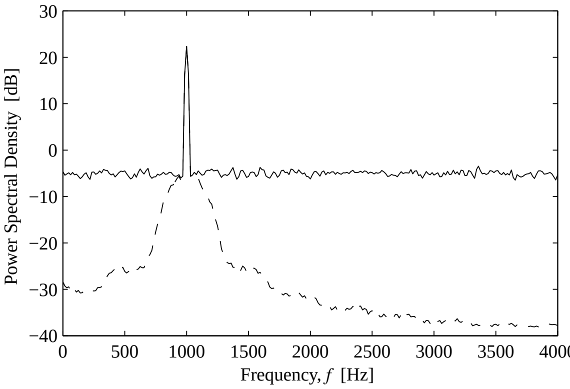


Figure 3.2 (solid) PSD of the noisy signal, and (dashed) of the projection onto the signal subspace ($p = 2$).

[1]Calculated for signals consisting of 16000 samples using Welch's method (512 sample segments, Hanning window and 256 sample overlap between adjacent segments).

3.3 Singular Value Decomposition

The eigenanalysis of the correlation matrix is closely related to the Singular Value Decomposition (SVD) of the data matrix, and in practice, the last approach will be used.

The SVD is a robust and widely used computational tool in noise suppression techniques and other signal processing applications. It is the most reliable tool for detecting numerical rank-deficiency in a matrix [39, page 246] and it also provides a basis for the fundamental subspaces associated with this matrix. The SVD of a matrix $\mathbf{X}$ is defined as follows (see, e.g., Golub and Van Loan [39, page 71])

THEOREM 3.2 (*Singular Value Decomposition*) *If* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *then there exist orthogonal matrices*

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{pmatrix} \in \mathbb{R}^{m\times m} \tag{3.20}$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{3.21}$$

such that

$$\mathbf{X} = \mathbf{U\Sigma V}^T = \sum_{i=1}^{q} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \ , \qquad q = \min\{m,n\} \tag{3.22}$$

where

$$\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_q) \in \mathbb{R}^{m\times n} \ , \qquad \sigma_1 \geq \ldots \geq \sigma_q \geq 0 \tag{3.23}$$

PROOF. For the proof, see [39, page 71]. □

The vector $\mathbf{u}_i$ is the ith *left singular vector* and the vector $\mathbf{v}_i$ is the ith *right singular vector*. The nonnegative diagonal elements of $\mathbf{\Sigma}$ are the *singular values* of $\mathbf{X}$ and their set is called the *singular spectrum* (see Stewart [109] for a historical survey of the SVD).

The SVD reveals much about the structure of a matrix. If the SVD of a rank deficient matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is given by Theorem 3.2, and p is defined by

$$\mathrm{rank}(\mathbf{X}) = p < q = \min\{m,n\} \tag{3.24}$$

then it is possible to partition the SVD of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U\Sigma V}^T = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{3.25}$$

where $\mathbf{U}_1 = (\mathbf{u}_1 \cdots \mathbf{u}_p) \in \mathbb{R}^{m\times p}$, $\mathbf{V}_1 = (\mathbf{v}_1 \cdots \mathbf{v}_p) \in \mathbb{R}^{n\times p}$ and $\mathbf{\Sigma}_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{p\times p}$. The submatrices span four fundamental spaces associated with $\mathbf{X}$

$$\langle \mathbf{U}_1 \rangle = \mathrm{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_p\} \quad = \mathrm{range}(\mathbf{X}) \ , \qquad \text{(Column space of } \mathbf{X}) \tag{3.26}$$

$$\langle \mathbf{U}_2 \rangle = \mathrm{span}\{\mathbf{u}_{p+1}, \ldots, \mathbf{u}_m\} = \mathrm{range}(\mathbf{X})^{\perp} \tag{3.27}$$

$$\langle \mathbf{V}_1 \rangle = \mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_p\} \quad = \mathrm{null}(\mathbf{X})^{\perp} \ , \qquad \text{(Row space of } \mathbf{X}) \tag{3.28}$$

$$\langle \mathbf{V}_2 \rangle = \mathrm{span}\{\mathbf{v}_{p+1}, \ldots, \mathbf{v}_n\} = \mathrm{null}(\mathbf{X}) \ , \qquad \text{(Null space of } \mathbf{X}) \tag{3.29}$$

which can be used to compute the orthogonal projections onto the column space and row space of $\mathbf{X}$, i.e.,

$$\mathbf{P}_{\mathbf{U}_1} = \mathbf{U}_1\mathbf{U}_1^T \quad \text{and} \quad \mathbf{P}_{\mathbf{V}_1} = \mathbf{V}_1\mathbf{V}_1^T \tag{3.30}$$

The decomposed matrix

$$\mathbf{X} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T = \sum_{i=1}^{p} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \tag{3.31}$$

can be used to define the *pseudoinverse* of $\mathbf{X}$ as

$$\mathbf{X}^+ = \mathbf{V}_1 \mathbf{\Sigma}_1^{-1} \mathbf{U}_1^T \in \mathbb{R}^{n \times m} \tag{3.32}$$

which is also given by $\mathbf{X}^+ = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ when $\mathbf{X}$ has full rank.

Finally, both the 2-norm (spectral norm) and the Frobenius norm (the sum of squares of all elements) are neatly characterized in terms of the SVD

$$\|\mathbf{X}\|_2 = \max_{\|\mathbf{z}\|_2=1} \|\mathbf{X}\mathbf{z}\|_2 = \sigma_1 \tag{3.33}$$

$$\|\mathbf{X}\|_F^2 = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X}\right) = \sigma_1^2 + \cdots + \sigma_p^2 \tag{3.34}$$

The SVD of the data matrix (3.8) may be used to represent the eigendecomposition of the sample correlation matrix $\hat{\mathbf{R}}_x$

$$\hat{\mathbf{R}}_x = \frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}_X \mathbf{\Sigma}_X^2 \mathbf{U}_X^T \tag{3.35}$$

or

$$\hat{\mathbf{R}}_x = \frac{1}{m}\mathbf{X}^T\mathbf{X} = \frac{1}{m}\mathbf{V}_X \mathbf{\Sigma}_X^2 \mathbf{V}_X^T \tag{3.36}$$

Thus, the SVD simultaneously diagonalizes the inner product $\mathbf{X}^T\mathbf{X}$ and the outer product $\mathbf{X}\mathbf{X}^T$. Obviously, the eigendecomposition of $\hat{\mathbf{R}}_x$ can mathematically reveal the same information as the SVD of the data matrix, but the SVD approach reduces the condition numbers to their square roots and increases numerical accuracy. Note that the singular values are unique, while the singular vectors will be unique only when σ_i^2 is a *simple* eigenvalue of $\mathbf{X}^T\mathbf{X}$.

Now, consider the singular values of the Toeplitz signal matrix $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a speech frame with 160 samples as shown in Figure 3.3 for the voiced and unvoiced case, respectively. Obviously, speech span the total space due to the stochastic nature of the signals and there is no well defined gap in the singular spectra. However, the quality of the speech is

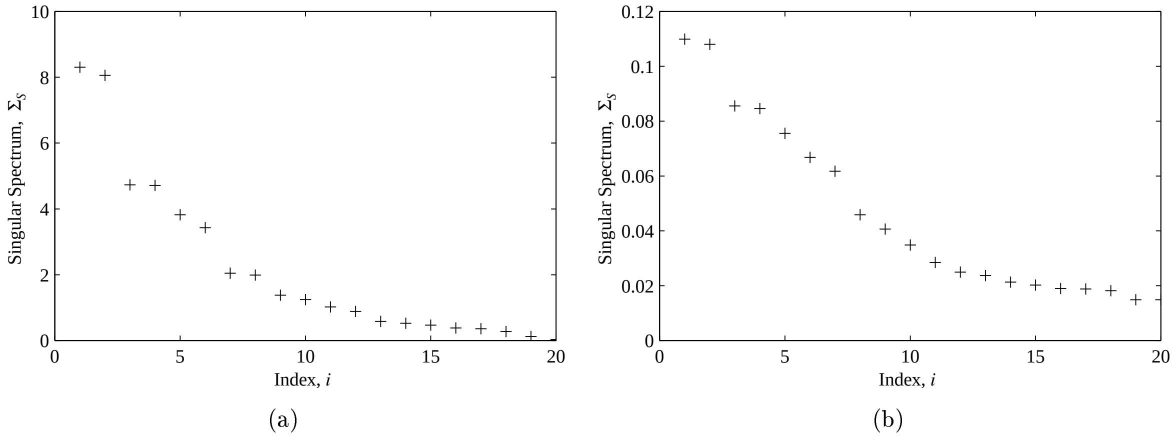


Figure 3.3 Singular spectrum of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced (a) and unvoiced (b) speech frame with 160 samples.

mainly connected to the formants, which is given by the pairs of singular values in the first part of the singular spectrum, i.e., projection onto a signal subspace can be done without noticeably degradation of the speech.

Another issue is the difference between voiced and unvoiced singular spectra. The latter has a smaller singular value spread and will require a larger signal subspace dimension. That this observation is fairly general can be seen in Figure 3.4, where amplitude waveform and singular spectra[2] versus time are shown for the word "enormously". Note that each singular spectrum is normalized with the largest singular value in order to compare frames with different amplitude levels. The only unvoiced sound /s/ around sample numbers 12000–13000 can easily be identified from the small singular value spreads.

A more detailed discussion of the rank decision problem for speech is postponed to Section 4.5, however, the result $p \approx 12$ will be used in examples from now on. In the following subsections, four SVD related topics important in subspace based noise reduction are considered.

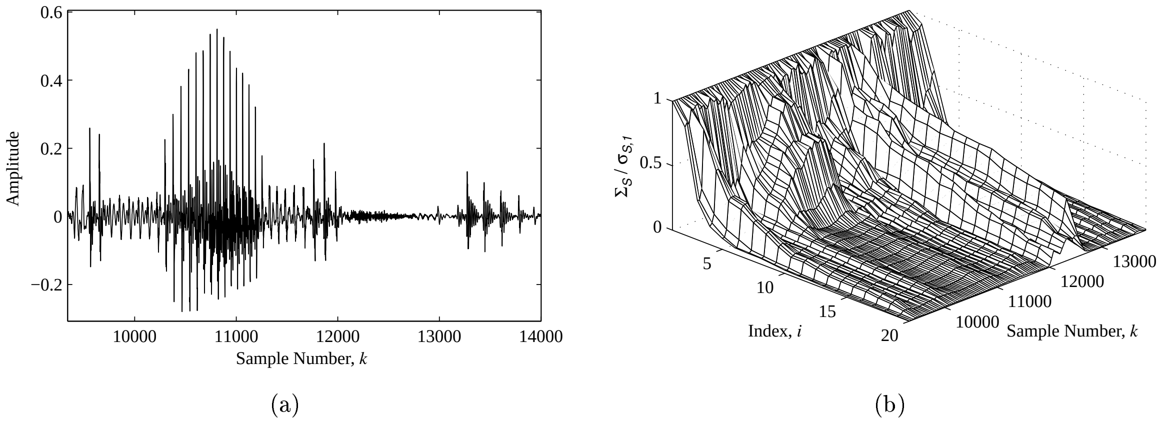


Figure 3.4 (a) Amplitude waveform of the word "enormously". (b) Normalized singular spectra $\mathbf{\Sigma}_S/\sigma_{S,1}$ versus time.

3.3.1 Matrix Approximations

The SVD plays an important role when a matrix $\mathbf{X}$ is approximated by a matrix of lower rank $p < n$ in the least squares sense as given by the following theorem (originally due to Eckart and Young [27])

THEOREM 3.3 (*Low Rank Matrix Approximation*) *Let the SVD of* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *be given by*

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{X1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \tag{3.37}$$

where $\mathbf{U}_{X1} \in \mathbb{R}^{m\times p}$, $\mathbf{\Sigma}_{X1} \in \mathbb{R}^{p\times p}$ *and* $\mathbf{V}_{X1} \in \mathbb{R}^{n\times p}$. *If*

$$\mathbf{X}_p = \mathbf{U}_{X1}\mathbf{\Sigma}_{X1}\mathbf{V}_{X1}^T \tag{3.38}$$

then

$$\min_{\mathrm{rank}(\mathbf{Y})=p} \|\mathbf{X}-\mathbf{Y}\|_F^2 = \|\mathbf{X}-\mathbf{X}_p\|_F^2 = \sigma_{X,p+1}^2 + \ldots + \sigma_{X,n}^2 \tag{3.39}$$

where the solution is unique.

[2]160 sample frames with 120 sample overlap between adjacent frames.

PROOF. Since $\mathbf{U}_X^T(\mathbf{X}-\mathbf{X}_p)\mathbf{V}_X = \mathbf{\Sigma}_{X2}$ it follows that $\|\mathbf{X}-\mathbf{X}_p\|_F^2 = \sigma_{X,p+1}^2 + \ldots + \sigma_{X,n}^2$. Now suppose $\mathrm{rank}(\mathbf{Y}) = p$ for some $\mathbf{Y} \in \mathbb{R}^{m\times n}$. Let a matrix $\mathbf{Z}$ with orthonormal columns span $\mathrm{null}(\mathbf{Y})$

$$\mathbf{YZ} = \mathbf{0} \quad \text{where} \quad \mathbf{Z}^T\mathbf{Z} = \mathbf{I} \tag{3.40}$$

then

$$\|\mathbf{X}-\mathbf{Y}\|_F^2 \geq \|(\mathbf{X}-\mathbf{Y})\mathbf{Z}\|_F^2 = \|\mathbf{XZ}\|_F^2 = \|\mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X^T\mathbf{Z}\|_F^2 \geq \sigma_{X,p+1}^2 + \ldots + \sigma_{X,n}^2 \tag{3.41}$$

□

The theorem is normally proved for the 2-norm (see, e.g., Golub and Van Loan [39, page 73]), where the minimum distance is

$$\min_{\mathrm{rank}(\mathbf{Y})=p} \|\mathbf{X}-\mathbf{Y}\|_2 = \|\mathbf{X}-\mathbf{X}_p\|_2 = \sigma_{X,p+1} \tag{3.42}$$

Hence, in words, $\mathbf{X}_p$ is the best least squares approximation of lower rank p to the given matrix $\mathbf{X}$, and $\mathbf{X}_p$ will be an estimate of the projection of data onto the signal subspace due to the relations (3.35) and (3.36). Note that when $\mathbf{X}$ has a specific structure, say, Toeplitz, the rank-p approximation generally spoil the Toeplitz structure (see Section 4.6 for a detailed discussion).

3.3.2 Numerical Rank

The rank-revealing property and the related subspace information are among the most valuable aspects of the SVD. The singular values can be considered as quantitative measures of the qualitative notation of rank, which algebraically is well-determined. However, in practice the effects of rounding errors and noisy data typically results in a full rank matrix making *numerical* rank determination of the underlying signal a nontrivial problem.

From the above considerations it follows that the numerical rank assigned to matrix $\mathbf{X}$ should depend on a tolerance reflecting the error level (see, e.g., Björck [9, page 99])

DEFINITION 3.1 (*Numerical Rank) A matrix* $\mathbf{X}$ *is said to have numerical* τ*-rank equal to* p *if*

$$p = \min\{\mathrm{rank}(\mathbf{Y}) \mid \|\mathbf{X}-\mathbf{Y}\|_2 \leq \tau\} \tag{3.43}$$

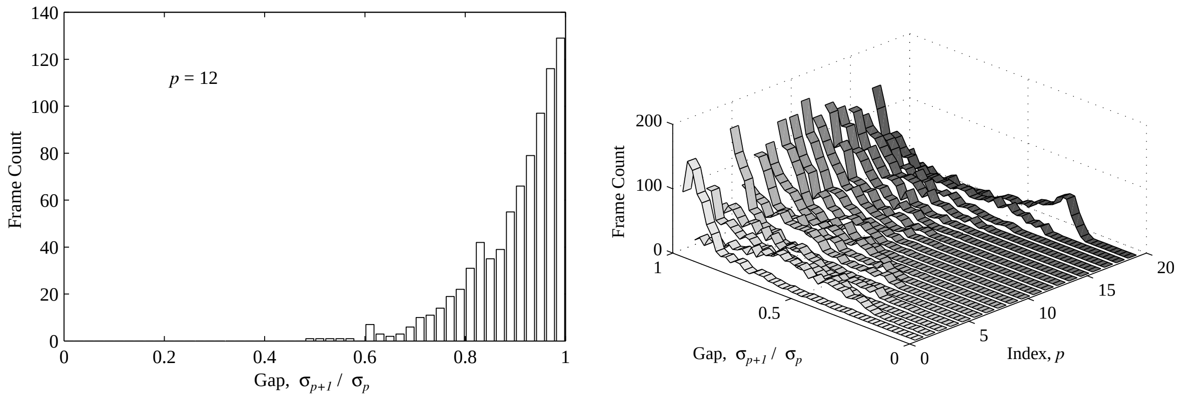
It follows from this definition, using Equation (3.42) that a matrix $\mathbf{X}$ has numerical τ-rank p if and only if its singular values satisfy

$$\sigma_1 \geq \cdots \geq \sigma_p > \tau \geq \sigma_{p+1} \geq \cdots \geq \sigma_n \tag{3.44}$$

The definition (3.44) is satisfactory only when there is a well-defined (relative) *gap* between σ_{p+1} and σ_p, i.e., if the ratio $\sigma_{p+1}/\sigma_p < 1$.

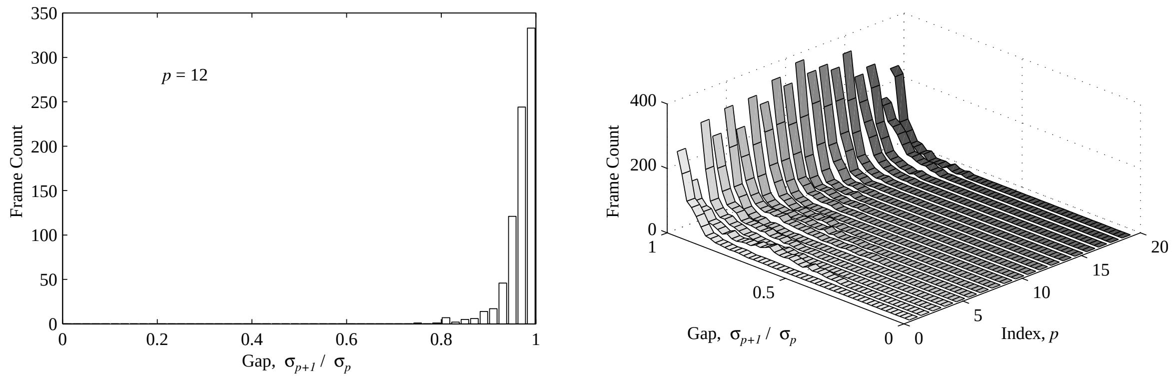
A gap at p may reflect a underlying rank degeneracy in a matrix of which $\mathbf{X}$ is a perturbation, or p may simply be a convenient point from which to reduce the dimensionality of a problem. Note that the numerical rank p is often chosen from the statement $\sigma_{p+1}/\sigma_p \ll 1$.

The gap as function of p has been computed for a large number of speech frames each consisting of 160 samples, i.e., the SVD was applied to the signal-only matrix $\mathbf{S} \in \mathbb{R}^{141\times 20}$. The frames were taken from the reference sentence with an 120 sample overlap between adjacent frames. In Figure 3.5, the distributions of the gap for each p are shown, and in general, they look like the example plot of $p = 12$, except for $p = 2$, 4 and perhaps 6 due to the separation of the dominant formants. The signal subspace dimension is normally chosen larger than 10, so the gap can be assumed to be negligible. When (white) noise is added, the gaps actually disappear as shown in Figure 3.6 for a global SNR = 10 dB.



(a) (b)

Figure 3.5 The gap σ_{p+1}/σ_p of 800 speech frames ($\mathbf{S} \in \mathbb{R}^{141\times 20}$) obtained from the reference sentence by shifting a 160 sample window by 40 samples. (a) $p = 12$. (b) $p = 1$ to 19.



(a) (b)

Figure 3.6 As Figure 3.5 with white noise added to the reference sentence (SNR=10dB).

3.3.3 Angles Between Subspaces

The angle between subspaces is the tool for validating subspace information, e.g., compare the signal subspace $\langle \mathbf{V}_{S1} \rangle$ given by the right singular vectors of the signal-only matrix $\mathbf{S}$ with the corresponding subspace $\langle \mathbf{V}_{X1} \rangle$ obtained from the noisy data matrix $\mathbf{X}$.

The relation between *canonical* angles/vectors and the SVD is given in the following theorem due to Björck and Golub [8]

THEOREM 3.4 (*Canonical Angles Between Subspaces*) *Assume that the columns of the matrices* $\mathbf{V}_{X1} \in \mathbb{R}^{n\times p_X}$ *and* $\mathbf{V}_{S1} \in \mathbb{R}^{n\times p_S}$ $(p_X \geq p_S)$ *form orthogonal bases for two subspaces, and let the SVD of* $\mathbf{V}_{X1}^T \mathbf{V}_{S1}$ *be*

$$\mathbf{V}_{X1}^T \mathbf{V}_{S1} = \mathbf{Y}_X \mathbf{C} \mathbf{Y}_S^T\,, \qquad \mathbf{C} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{p_S}) \tag{3.45}$$

where $1 \geq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{p_S}$, *then the canonical angles* $\theta_i \in [0; \pi/2]$ *between* $\mathrm{range}(\mathbf{V}_{X1})$ *and* $\mathrm{range}(\mathbf{V}_{S1})$ *and the associated canonical vectors* $\mathbf{w}_{X,i}$ *and* $\mathbf{w}_{S,i}$ *are given by*

$$\cos\theta_i = \sigma_i\,, \quad \mathbf{W}_X = \mathbf{V}_{X1}\mathbf{Y}_X\,, \quad \mathbf{W}_S = \mathbf{V}_{S1}\mathbf{Y}_S \tag{3.46}$$

PROOF. Follows from the minimax characterization of singular values, see [8]. □

The quantities $\cos\theta_i$ are also denoted the *canonical correlations*, and it is often convenient to introduce $\mathbf{\Theta}(\langle\mathbf{V}_{X1}\rangle, \langle\mathbf{V}_{S1}\rangle) \in \mathbb{R}^{ps\times ps}$ as the diagonal matrix of canonical angles between the two subspaces given by the arguments. The columns of $\mathbf{W}_X$ and $\mathbf{W}_S$ form orthogonal bases for $\mathbf{V}_{X1}$ and $\mathbf{V}_{S1}$, respectively, and it can be shown that

$$\mathbf{w}_{X,i}^T\mathbf{w}_{S,j} = 0\,, \qquad i \neq j \tag{3.47}$$

so Theorem 3.4 can be used to compute an orthonormal basis for the q-dimensional intersection of subspaces, i.e.,

$$\begin{gathered} 1 = \cos\theta_1 = \cdots = \cos\theta_q > \cos\theta_{q+1} \quad \Rightarrow \\ \mathrm{range}(\mathbf{V}_{X1}) \cap \mathrm{range}(\mathbf{V}_{S1}) = \mathrm{span}\{\mathbf{w}_{X,1},\ldots,\mathbf{w}_{X,q}\} = \mathrm{span}\{\mathbf{w}_{S,1},\ldots,\mathbf{w}_{S,q}\} \end{gathered} \tag{3.48}$$

which follows from the observation that if $\cos\theta_i = 1$, then $\mathbf{w}_{X,i} = \mathbf{w}_{S,i}$.

For small angles, θ_i is not well determined from $\cos\theta_i$, so a better choice is to find $\sin\theta_i$ from a complement based version of Equation (3.45) as shown in [8]

$$(\mathbf{I} - \mathbf{V}_{X1}\mathbf{V}_{X1}^T)\mathbf{V}_{S1} = \mathbf{Z}_X\mathbf{S}\mathbf{Y}_S^T\,, \qquad \mathbf{S} = \sin\mathbf{\Theta} \tag{3.49}$$

The largest canonical angle is also related to the notion of distance between equidimensional subspaces. The following definition from [39, page 77] quantifies this notation.

DEFINITION 3.2 (*Distance Between Equidimensional Subspaces) Let* $\mathbf{V}_X = (\mathbf{V}_{X1}\ \mathbf{V}_{X2})$ *and* $\mathbf{V}_S = (\mathbf{V}_{S1}\ \mathbf{V}_{S2})$ *be orthogonal matrices where* $\mathbf{V}_{X1}, \mathbf{V}_{S1} \in \mathbb{R}^{n\times p}$ *and* $\mathbf{V}_{X2}, \mathbf{V}_{S2} \in \mathbb{R}^{n\times(n-p)}$. *Then*

$$\begin{aligned} \mathrm{dist}(\langle\mathbf{V}_{X1}\rangle, \langle\mathbf{V}_{S1}\rangle) &= \|\mathbf{V}_{X1}\mathbf{V}_{X1}^T - \mathbf{V}_{S1}\mathbf{V}_{S1}^T\|_2 \\ &= \|\mathbf{V}_{X1}^T\mathbf{V}_{S2}\|_2 \\ &= \|\mathbf{V}_{X2}^T\mathbf{V}_{S1}\|_2 \\ &= \sqrt{1-\cos^2\theta_p} \\ &= \sin\theta_p \end{aligned} \tag{3.50}$$

where θ_p *is the largest canonical angle between* $\langle\mathbf{V}_{X1}\rangle$ *and* $\langle\mathbf{V}_{S1}\rangle$.

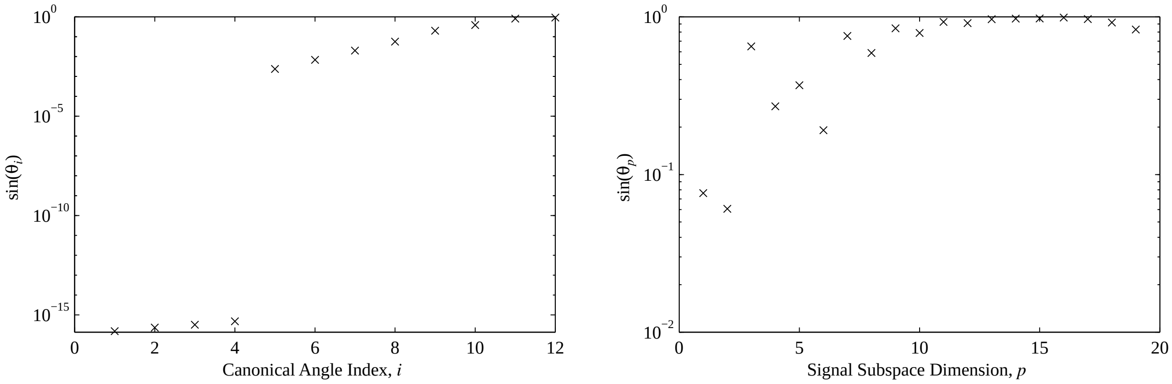
Note that the canonical vectors need not be uniquely defined, but the canonical angles always are.

In the speech enhancement application, the noise perturbed data matrix $\mathbf{X} = \mathbf{S} + \mathbf{N}$ defined by (3.8) is available, but what is interesting is the projection onto the signal subspace spanned by the pure signal matrix $\mathbf{S}$.

To illustrate this problem, consider the 12-dimensional signal subspace $\langle\mathbf{V}_{S1}\rangle$ given by the right singular vectors of the signal-only matrix $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing the voiced speech frame with 160 samples. The canonical angles between this signal subspace and the one obtained from the corresponding noisy data matrix using 100 independent white noise realizations and SNR = 5 dB are shown in Figure 3.7(a).

The distance is large ($\sin\theta_{max} = 0.91$), which can be explained by the missing gab in the singular spectrum of the clean signal, i.e., the signal and noise subspace are blurred together when noise is added (see Section 3.3.4). However, this does not mean that the noisy signal subspace is useless, since all directions $\mathbf{v}_{S,i}$ for i close to p contains approximately the same signal energi. Another observation is that four angles are close to the machine precision arising from the fact that two p-dimensional subspaces in a n-dimensional space must have a $2p - n$ dimensional intersection. Thus, a part of the noise-free space can be represented exactly.

In Figure 3.7(a), the distance $\sin\theta_p$ between the two subspaces is shown as function of the signal subspace dimension p. Clearly, the distance becomes close to one for $p > 8$ due to the fact that this speech frame has four formants, so there are (small) gaps in the signal-only singular spectrum for $p = 2$, 4, 6 and 8 (see Figure 3.3).



(a) (b)

Figure 3.7 (a) $\sin\mathbf{\Theta}(\langle\mathbf{V}_{S1}\rangle, \langle\mathbf{V}_{X1}\rangle)$, where the $p = 12$ dimensional right singular subspaces are obtained from $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples, and the corresponding noisy data matrix $\mathbf{X}$ (SNR=5dB). Note that $\sin\mathbf{\Theta}$ is averaged over 100 independent white noise realizations. (b) The distance $\sin\theta_p$ as function of the signal subspace dimension p.

3.3.4 Perturbation Theory

A possible mix of signal and noise subspace was observed in the last section when the signal-only matrix $\mathbf{S}$ was added noise, i.e., $\mathbf{X} = \mathbf{S} + \mathbf{N}$. This can be quantified by the perturbation theory for singular values/vectors.

First, consider the perturbation bounds of the singular values (see, e.g., Golub and Van Loan [39, page 428])

THEOREM 3.5 (*Perturbation Bounds for Singular Values*) *Let* $\mathbf{X} = \mathbf{S} + \mathbf{N} \in \mathbb{R}^{m\times n}$, $m \geq n$, *then*

$$|\sigma_{X,i} - \sigma_{S,i}| \leq \|\mathbf{N}\|_2 \tag{3.51}$$

$$\sum_{i=1}^{n} |\sigma_{X,i} - \sigma_{S,i}|^2 \leq \|\mathbf{N}\|_F^2 \tag{3.52}$$

The theorem shows that the singular values of a matrix are well-conditioned with respect to perturbations, i.e., perturbations of the elements of a matrix produce perturbations of the same, or smaller, magnitude in the singular values.

Perturbation results for singular vectors can be more or less general, i.e., smaller upper bounds is given by more complex expressions. The following theorem taken from Björck [9, page 14] is useful in our context

THEOREM 3.6 (*Perturbation Bounds for Singular Vectors*) *Let* $\mathbf{X} = \mathbf{S} + \mathbf{N} \in \mathbb{R}^{m\times n}$, $m \geq n$, *then*

$$\max\left(\sin\theta(\mathbf{u}_{X,i}, \mathbf{u}_{S,i}), \sin\theta(\mathbf{v}_{X,i}, \mathbf{v}_{S,i})\right) \leq \frac{\|\mathbf{N}\|_2}{\gamma_i - \|\mathbf{N}\|_2} = \beta_i \tag{3.53}$$

where γ_i *is the absolute gap between* $\sigma_{S,i}$ *and the other singular values*

$$\gamma_i = \min_{j\neq i} |\sigma_{S,i} - \sigma_{S,j}| \tag{3.54}$$

Thus, if the noise level $\|\mathbf{N}\|_2$ is equal to half the gap in the signal-only singular spectrum, then the upper bound β_i is $\frac{\pi}{2}$. For higher noise levels, the theory is no longer valid due to the possible *swap* of signal-only singular vectors when noise is added (see also the analysis in [112]).

3.4 Subspace Methods and the SVD

Bart De Moor has given a detailed derivation in [82] of the algebraic and geometric conditions that allow us to derive the signal-only model from the SVD of the noisy data matrix. This framework is important for understanding subspace methods and is summarized in this section.

Let the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ be defined by (3.8) and assume that the SVD of the rank deficient signal matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ is given by

$$\mathbf{S} = \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \tag{3.55}$$

where $\mathbf{U}_{S1} \in \mathbb{R}^{m\times p}$, $\mathbf{\Sigma}_{S1} \in \mathbb{R}^{p\times p}$ and $\mathbf{V}_{S1} \in \mathbb{R}^{n\times p}$. Then under the sufficient condition

$$\mathbf{V}_{S1}^T(\mathbf{S}^T + \mathbf{N}^T)\mathbf{N}\mathbf{V}_{S2} = \mathbf{0} \tag{3.56}$$

it is shown in [82], that from the SVD of the noisy data matrix $\mathbf{X}$, the rank p and the subspaces $\langle\mathbf{V}_{S1}\rangle$ and $\langle\mathbf{V}_{S2}\rangle$ can be obtained, while the subspaces $\langle\mathbf{U}_{S1}\rangle$ and $\langle\mathbf{U}_{S2}\rangle$ cannot be recovered, not even asymptotically as $m \to \infty$. Especially, one more restrictive splitting of (3.56) is useful in signal processing and can be formulated in the following three necessary conditions

ASSUMPTION 3.3 (*Algebraic and Geometric Conditions*)

1. *The signal is orthogonal to the noise in the sense:* $\mathbf{S}^T\mathbf{N} = \mathbf{0}$.
2. *The noise matrix* $\mathbf{N} = \sigma_{noise}\mathbf{Q}$, *where* $\mathbf{Q}$ *has orthonormal columns:* $\mathbf{N}^T\mathbf{N} = \sigma_{noise}^2\mathbf{I}_n$.
3. *There is a distinct gap in the singular values of the matrix* $\mathbf{X}$*:* $\sigma_{X,p} > \sigma_{X,p+1}$.

In practice, Assumption 3.3-1 and 3.3-2 are never satisfied exactly, but the SVD is robust with respect to mild violations of these conditions. Assumption 3.3-3 makes it possible to separate $\langle\mathbf{V}_{X1}\rangle$ from $\langle\mathbf{V}_{X2}\rangle$. The SVD of $\mathbf{X}$ can now be written in terms of the SVD of $\mathbf{S}$

$$\begin{aligned} \mathbf{X} &= \mathbf{S} + \mathbf{N} \\ &= \mathbf{U}_{S1}\mathbf{\Sigma}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S2}\mathbf{V}_{S2}^T \\ &= \begin{pmatrix} (\mathbf{U}_{S1}\mathbf{\Sigma}_{S1} + \mathbf{N}\mathbf{V}_{S1})(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1/2} & \mathbf{N}\mathbf{V}_{S2}\sigma_{noise}^{-1} \end{pmatrix} \\ &\qquad \times \begin{pmatrix} \sqrt{\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{X1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \end{aligned} \tag{3.57}$$

where the orthogonality of $\mathbf{U}_X$ depend on the above mentioned assumptions. This equation is the empirical counterpart to (3.18), and from (3.57), the following conclusions are drawn:

Singular values. There is a distinct gap in the singular spectrum of the matrix $\mathbf{X}$

$$\sigma_{X,p} = \sqrt{\sigma_{S,p}^2 + \sigma_{noise}^2} \;\; > \;\; \sigma_{X,p+1} = \sigma_{noise} \tag{3.58}$$

where $\sigma_{S,p}$ is the smallest value of $\mathbf{\Sigma}_{S1}$. The $n-p$ smallest singular values $\mathbf{\Sigma}_{X2}$ are all equal and can be interpreted as a *noise threshold*, which permits estimating the noise variance and the singular values of $\mathbf{S}$ from

$$\sigma_{noise}^2 = \frac{1}{n-p}\sum_{i=p+1}^{n}\sigma_{X,i}^2 \tag{3.59}$$

$$\mathbf{\Sigma}_{S1} = \sqrt{\mathbf{\Sigma}_{X1}^2 - \sigma_{noise}^2\mathbf{I}_p} \tag{3.60}$$

This gives one possible definition of the signal-to-noise ratio related to the total space

$$\mathrm{SNR} = \left(\frac{\sigma_{S,1}^2 + \cdots + \sigma_{S,p}^2}{n\sigma_{noise}^2}\right) = \frac{\|\mathbf{S}\|_F^2}{\|\mathbf{N}\|_F^2} \tag{3.61}$$

and the one related to the signal subspace

$$\mathrm{SNR}_1 = \left(\frac{\sigma_{S,1}^2 + \cdots + \sigma_{S,p}^2}{p\sigma_{noise}^2}\right) \tag{3.62}$$

giving the previously mentioned improvement factor n/p in SNR.

Right singular vectors. The row and null space of $\mathbf{S}$ can be recovered exactly from $\mathbf{V}_{S1} = \mathbf{V}_{X1}$ and $\mathbf{V}_{S2} = \mathbf{V}_{X2}$.

Left singular vectors. It is impossible to recover the original noise-free column space $\langle\mathbf{U}_{S1}\rangle$ of $\mathbf{S}$ since $\mathbf{U}_{X1} \neq \mathbf{U}_{S1}$. But the canonical angles $\{\phi\}_1^p$ between $\langle\mathbf{U}_{S1}\rangle$ and $\langle\mathbf{U}_{X1}\rangle$ that characterize the bias of the column space can be computed from the singular values $\mathbf{C}$ of the product $\mathbf{U}_{S1}^T\mathbf{U}_{X1}$ (see Section 3.3.3)

$$\begin{aligned}\mathbf{U}_{S1}^T\mathbf{U}_{X1} &= \mathbf{U}_{S1}^T(\mathbf{U}_{S1}\mathbf{\Sigma}_{S1} + \mathbf{N}\mathbf{V}_{S1})(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1/2} \\ &= \mathbf{\Sigma}_{S1}(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1/2} \\ &= (\mathbf{I}_p + \sigma_{noise}^2\mathbf{\Sigma}_{S1}^{-2})^{-1/2} \\ &= \mathbf{\Sigma}_{S1}/\mathbf{\Sigma}_{X1} = \mathbf{C}\end{aligned} \tag{3.63}$$

where the cosines of the canonical angles are given by the diagonal elements of the diagonal matrix $\mathbf{C}$

$$\mathbf{C} = \mathrm{diag}\,(\cos\phi_1 \cdots \cos\phi_p) \tag{3.64}$$

$$c_{ii} = \cos\phi_i = \frac{1}{\sqrt{1 + \frac{\sigma_{noise}^2}{\sigma_{S,i}^2}}}, \qquad i = 1, \ldots, p \tag{3.65}$$

Observe that the canonical angles depend on the gap in the singular spectrum (3.58) or equivalently on the signal-to-noise ratio. In the noise-free case all angles ϕ_i are zero.

The bias of $\mathbf{U}_{S1}$ is given by the second term in $(\mathbf{U}_{S1}\mathbf{\Sigma}_{S1} + \mathbf{N}\mathbf{V}_{S1})$ or equivalently by the second term in the column norms of $(\mathbf{U}_{S1}\mathbf{\Sigma}_{S1} + \mathbf{N}\mathbf{V}_{S1})$, which are the diagonal elements of

$$\sqrt{\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2 \mathbf{I}_p} \tag{3.66}$$

This equation is obtained from the orthogonality between the columns of $\mathbf{U}_{S1}\mathbf{\Sigma}_{S1}$ and $\mathbf{N}\mathbf{V}_{S1}$ and is a multivariate generalization of the Pythagorean lemma. This is also the reason why the original exact signal matrix $\mathbf{S}$ *can not* be recovered consistently when the exact data are perturbated by additive noise.

3.4.1 Stochastic Signals

The analysis in the last section was based on the algebraic and geometrical assumptions on $\mathbf{S}$ and $\mathbf{N}$ that are sufficient to recover the model information about $\mathbf{S}$ from the SVD of $\mathbf{X}$. Assumption 3.3 on Page 35 is never exactly satisfied, but it is shown in [82] that, under some mild conditions, the requirements are achieved asymptotically, as the row dimension $m \to \infty$.

Let the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ be defined by (3.8) and assume that $\mathbf{S}$ is a fixed deterministic matrix, then a statistical version of the algebraic and geometrical conditions are given by

Assumption 3.4 (*Statistical Conditions*)

1. *The elements of* $\mathbf{N}$ *and* $\mathbf{S}$ *are uncorrelated:* $E\{\mathbf{S}^T\mathbf{N}\} = \mathbf{0}$.

2. *The elements of* $\mathbf{N}$ *are independently and identically distributed with possibly unknown variance* ν_{noise}^2 *(white noise):* $E\{\mathbf{N}^T\mathbf{N}\} = m\nu_{noise}^2\mathbf{I}_n$.

Hence, in this case

$$E\{\mathbf{X}^T\mathbf{X}\} = \mathbf{S}^T\mathbf{S} + m\nu_{noise}^2\mathbf{I}_n \tag{3.67}$$

which satisfies Assumption 3.3 on Page 35. Typically, in practical experimental situations, one cannot average over several experiments with identical $\mathbf{S}$ matrix but with different realizations for $\mathbf{N}$ (ensemble average operator $E\{\cdot\}$). In one experiment, however, it is very well possible to take a large number of measurements, so instead consider the sample correlation matrix (see Section 3.1.2)

$$\frac{1}{m}\mathbf{X}^T\mathbf{X} = \frac{1}{m}\mathbf{S}^T\mathbf{S} + \frac{1}{m}\mathbf{N}^T\mathbf{N} + \frac{1}{m}(\mathbf{S}^T\mathbf{N} + \mathbf{N}^T\mathbf{S}) \tag{3.68}$$

and extend Assumption 3.4 with

Assumption 3.5 (*Additional Statistical Conditions*)

1. *The signals* $\mathbf{s}_i$ *are quasi-stationary:*

$$\lim_{m\to\infty}\left(\frac{1}{m}\mathbf{s}_i^T\mathbf{s}_i\right) = \text{finite} \quad \textit{and} \quad \lim_{m\to\infty}\left(\frac{1}{m^2}\mathbf{s}_i^T\mathbf{s}_i\right) = 0$$

2. *The fourth-order moments of the noise are bounded:* $E\{\mathbf{N}_{ii}^4\} = \text{finite}$.

then it is shown in [82] that the following conditions are satisfied

$$\operatorname*{plim}_{m\to\infty}\left(\frac{1}{m}\mathbf{S}^T\mathbf{N}\right) = \mathbf{0} \tag{3.69}$$

$$\operatorname*{plim}_{m\to\infty}\left(\frac{1}{m}\mathbf{N}^T\mathbf{N}\right) = \nu_{noise}^2\mathbf{I}_n \tag{3.70}$$

$$\operatorname*{plim}_{m\to\infty}\lambda_i\left(\frac{1}{m}\mathbf{X}^T\mathbf{X}\right) = \lambda_i\left(\frac{1}{m}\mathbf{S}^T\mathbf{S}\right) + \nu_{noise}^2 \tag{3.71}$$

where *plim* is the probability limit (i.e., mean values with zero variance), and $\lambda_i(\mathbf{M})$ is the ith eigenvalue of the matrix $\mathbf{M}$. This means that as $m \to \infty$ we gradually approach the geometric Assumption 3.3 on Page 35 implying that the SVD of $\mathbf{X}$ also gradually approaches the SVD in (3.57).

Since the singular values are perfectly conditioned (see Section 3.3.4), it follows from the perturbation theory for singular values that with this additive noise, the noise threshold $m\nu_{noise}$ will become more and more pronounced for increasing m. This allows us to estimate the singular values, the signal-to-noise ratio, and the canonical angles from Equations (3.59 – 3.65).

In the case of Gaussian noise with zero mean and variance ν_{noise}^2, the fourth-order moments are $3\nu_{noise}^4$ (bounded), which gives a more restrictive formulation of Assumption 3.5-2

ASSUMPTION 3.6 (*Gaussian Assumption 3.5-2)*

1. *The elements of* $\mathbf{N}$ *are Gaussian distributed.*

The conclusion is that the strong consistency of estimates of the singular values and the right singular vectors (short space) only depends on the convergence of the sample correlation matrix with probability 1 to its expected value, which is the case when the fourth-order moments are bounded and the exact signal is quasi-stationary.

The bias of the left singular vectors (long space) also depends on the orthogonality between the $\mathbf{S}$ and $\mathbf{N}$. However, since the noise is uniformly distributed in the entire Euclidean space, the norm of the projected noise matrix $\|\mathbf{S}^T\mathbf{N}/m\|_2$ goes to zero as $\nu_{noise}/\sqrt{m}$.

This is verified by the example shown in Figure 3.8 where $\mathbf{S} \in \mathbb{R}^{m\times 20}$ represents a voiced speech frame and $\mathbf{N}$ is obtained from a white noise realization (SNR = 5 dB).

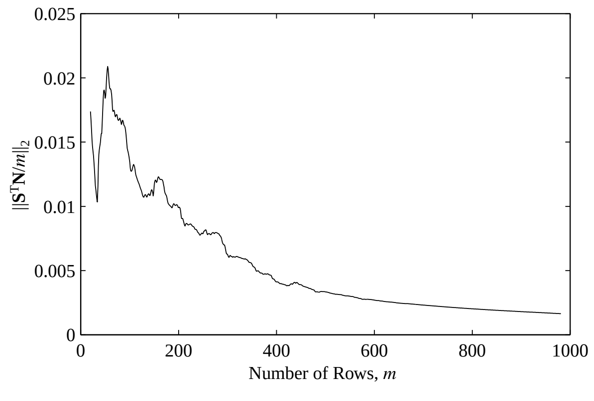


Figure 3.8 $\|\mathbf{S}^T\mathbf{N}/m\|_2$ as function of the row dimension m, where $\mathbf{S} \in \mathbb{R}^{m\times 20}$ represents a voiced speech frame with $m+19$ samples, and $\mathbf{N}$ is obtained from a white noise realization (SNR=5dB).

3.4.2 Data Matrix Dimensions

The speech signal can only be considered as a wide sense stationary stochastic process in a short window, which gives a limit on the number of data points K to construct the Toeplitz data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ (3.8). Typically, this number equals 200 – 300 samples at 8 kHz sampling rate (160 samples is used here due to the GSM specification).

Now the problem is, which choice of matrix dimension (m, n) is best subject to the constraints $K = m + n - 1$ and $m \geq n > p$.

- The statistical analysis in Section 3.4.1 suggests choosing m as large as possible to obtain an isotropic noise spectrum and to make the signal data orthogonal to the noise.
- Another issue is to have a large n, because the improvement in SNR obtained by the signal subspace approach is proportional to n/p, since the signal subspace dimension p is fixed, and the noise is evenly distributed in the entire n-dimensional space (see Section 3.3).
- Finally, n should be chosen small to reduce computational complexity in performing SVD of the data matrix.

Consider $K = m + n - 1$ as a constant, then the number of elements in $\mathbf{X}$ is given by the quadratic function

$$mn = -n^2 + (K+1)n \tag{3.72}$$

which has maximum $(K+1)^2/4$ for $n = (K+1)/2$ and decreases to K for $n = 1$. Thus, as $n \to p$, the signal energy

$$\|\mathbf{S}\|_F^2 = \sum_{i=1}^{p} \sigma_{S,i}^2 \approx mn\nu_s^2 \, , \qquad \nu_s^2 = E\{s_k^2\} \tag{3.73}$$

decreases quadratically, i.e., the signal-only singular values $\sigma_{S,i}$ decreases (see e.g., the related problem in [118]). However, as n decreases, m and thereby the noise singular values $\sigma_{noise}^2 \approx m\nu_{noise}^2$ increases. This implies that the gap in the singular values (3.58) narrows enlarging the bias (3.66) of the signal subspace estimate $\langle \mathbf{U}_{X1} \rangle$, so n should not be too small.

As a compromise the following dimensions are often chosen

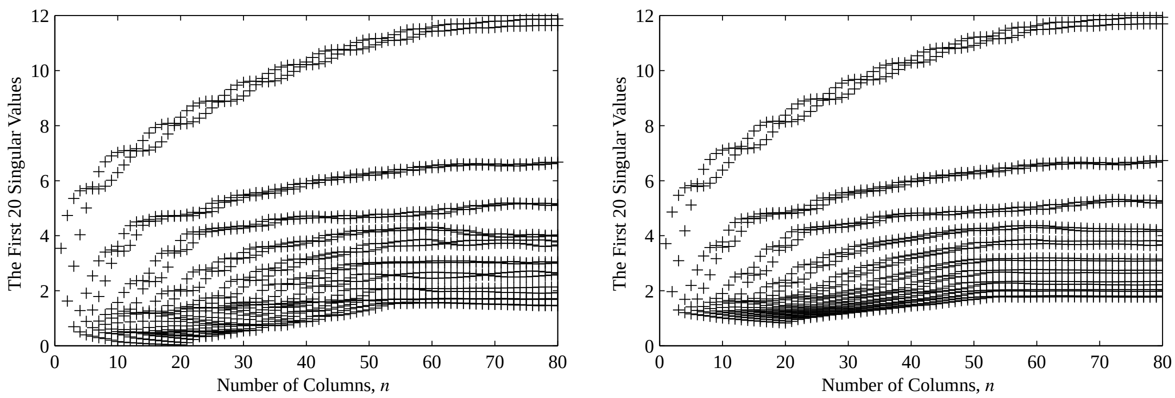
$$n \approx 2p \quad \Rightarrow \quad m \approx K - 2p + 1 \tag{3.74}$$

As discussed in Section 4.5, a common order of the speech model is around 10 giving the following matrix dimensions

$$n = 20 \quad \text{and} \quad m = 141 \tag{3.75}$$

In practice, the speech signal matrix $\mathbf{S}$ has full rank, but the above theory is still valid due to the low signal energy in the noise subspace. This can be observed in Figure 3.9(a), where the first 20 singular values of $\mathbf{S}$ representing a voiced speech frame with 160 samples are shown as function of the column dimension n. Figure 3.9(b) shows the average singular values of the corresponding noisy data matrix using 100 independent white noise realizations and SNR = 10 dB. Clearly, the smallest singular values are now dominated by noise for decreasing n, and the subspaces are more blurred together. Another interesting thing that can be seen from these figures is the pairing of the singular values, each representing a formant, i.e., for $n = 20$, the 3–5 most important formants will typically be in the signal subspace.

For the chosen column dimension, it is necessary to check whether the row dimension is large enough to satisfy the conditions (3.69 – 3.71). As shown in Figure 3.10, this can be done by

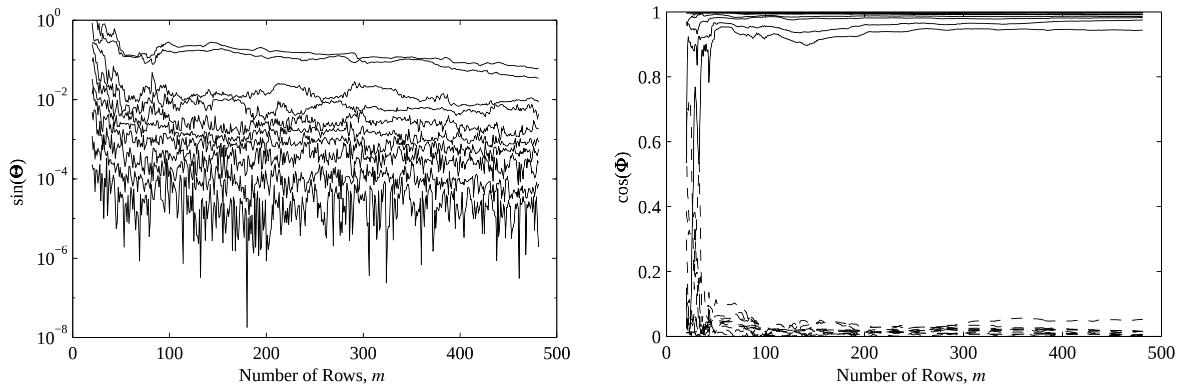


(a) (b)

Figure 3.9 (a) The first 20 singular values of $\mathbf{S}$ representing a voiced speech frame with 160 samples. (b) The average singular values of the corresponding noisy data matrix using 100 independent white noise realizations and SNR=10dB.

observing the canonical angles between the singular subspaces of the signal-only matrix $\mathbf{S}$ and the noisy data matrix $\mathbf{X}$ as function of the row dimension m. Here, the matrix $\mathbf{S} \in \mathbb{R}^{m\times 20}$ represents a voiced speech frame with $m + 19$ samples, and a white noise realization is added (SNR $= 5$ dB) to obtain the data matrix $\mathbf{X}$. The signal subspace dimension is $p = 10$.

If all assumptions are satisfied, the right singular subspace based values $\sin\mathbf{\Theta}(\langle\mathbf{V}_{S1}\rangle, \langle\mathbf{V}_{X1}\rangle)$ should converge to zero, and the left singular subspace based values $\cos\mathbf{\Phi}(\langle\mathbf{U}_{S1}\rangle, \langle\mathbf{U}_{X1}\rangle)$ should converge to the theoretical result (3.63). From Figure 3.10 is observed that both quantities have converged satisfactory for $m \approx 100$, and the error between the estimated and theoretical values even for large m rely on the fact that $\mathbf{S}$ is not rank deficient and that all the assumptions in Section 3.4.1 are not exactly satisfied.



(a) (b)

Figure 3.10 Canonical angles between signal subspaces (dimension $p = 10$) obtained from the signal-only matrix $\mathbf{S} \in \mathbb{R}^{m\times 20}$ representing the voiced speech frame with m+19 samples and the corresponding noisy data matrix $\mathbf{X}$ added white noise (SNR=5dB). (a) The angles between the right singular subspaces, i.e., $\sin\mathbf{\Theta}(\langle\mathbf{V}_{S1}\rangle, \langle\mathbf{V}_{X1}\rangle)$, which should converge to zero. (b) (solid) The angles between the left singular subspaces, i.e., $\cos\mathbf{\Phi}(\langle\mathbf{U}_{S1}\rangle, \langle\mathbf{U}_{X1}\rangle)$, which should converge to the theoretical result (3.63). (dashed) The error $|\mathbf{\Sigma}_{S1}/\mathbf{\Sigma}_{X1} - \cos\mathbf{\Phi}|$ between estimated and theoretical values.

3.5 Eigenfilters

The signal subspace approach can be interpreted as filter operations, i.e., in the frequency domain, as an alternative to the standard linear algebra formulation.

A *Finite Impulse Response* (FIR) filter whose impulse response has coefficients equal to the elements of an eigenvector of the correlation matrix of the signal under consideration is called an *eigenfilter* [74], and as discussed in [56, Page 145], the eigenvalue problem is linked to an optimum FIR filter, with the optimization criterion being that of maximizing the output signal-to-noise ratio (SNR).

Consider a linear FIR filter with coefficient vector $\mathbf{w} \in \mathbb{R}^n$, and assume that the input sequence $x_k = s_k + n_k$ is a wide-sense stationary stochastic process with correlation matrix $\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\} \in \mathbb{R}^{n\times n}$ as defined by (3.11). Then the average power of the filter ouput can be separated in a signal-only and noise part, respectively

$$P_s = \mathbf{w}^T\mathbf{R}_s\mathbf{w} \tag{3.76}$$

$$P_n = \nu_{noise}^2\mathbf{w}^T\mathbf{w} \tag{3.77}$$

giving the output signal-to-noise ratio

$$\mathrm{SNR}_{out} = \frac{P_s}{P_n} = \left(\frac{1}{\nu_{noise}^2}\right)\left(\frac{\mathbf{w}^T\mathbf{R}_s\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\right) \tag{3.78}$$

where the last factor is the *Rayleigh quotient* of the coefficient vector $\mathbf{w}$. Thus, the optimum filtering problem may be stated as follows

$$\max_{\mathbf{w}} \mathrm{SNR}_{out} \quad \text{subject to} \quad \mathbf{w}^T\mathbf{w} = 1 \tag{3.79}$$

and the solution follows directly from the minimax eigenvalue theorem [39, page 411], i.e.,

$$\mathbf{w}_o = \mathbf{q}_{max} \quad \text{and} \quad \mathrm{SNR}_{out,max} = \frac{\lambda_{s,max}}{\nu_{noise}^2} \tag{3.80}$$

where $\mathbf{q}_{max}$ is the eigenvector associated with the largest eigenvalue $\lambda_{s,max}$ of the correlation matrix $\mathbf{R}_s$. Such a procedure represents an example of *principal component analysis.*

Moreover, there is an interesting relation between the power spectral density $\Gamma_x(\omega)$ of the stochastic process x and the eigenvalues of $\mathbf{R}_x$. This can be seen by using the fact that the correlation function is the inverse DFT of the power spectral density (PSD), i.e,

$$r_x(l-k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_x(\omega)e^{j\omega(l-k)}d\omega \tag{3.81}$$

in the expanded form of the eigendecomposition

$$\begin{aligned}
\lambda_{x,i} &= \mathbf{q}_i^T\mathbf{R}_x\mathbf{q}_i \qquad (3.82)\\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} q_{i,k}\, r_x(l-k)\, q_{i,l}\\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{k=1}^{n} q_{i,k}e^{-j\omega k}\right)\left(\sum_{l=1}^{n} q_{i,l}e^{j\omega l}\right)\Gamma_x(\omega)d\omega\\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} |Q_i(\omega)|^2\Gamma_x(\omega)d\omega
\end{aligned}$$

where the discrete Fourier transform of $\mathbf{q}_i$ is denoted by

$$Q_i(\omega) = \sum_{k=1}^{n} q_{i,k} e^{-j\omega k} \tag{3.83}$$

Thus, the eigenvalue $\lambda_{x,i}$ is just the power of the output of the eigenfilter $Q_i(\omega)$, and (3.82) can be viewed as a frequency-domain expression of the eigendecomposition of $\mathbf{R}_x$, i.e., it is possible to illustrate the filtering aspects of subspace methods by plotting the magnitude of each eigenfilter. Note that since the n eigenfilters are mutually orthogonal, the n output signals will be uncorrelated.

As an illustrative example, consider the voiced speech frame with 160 samples organized in the Toeplitz matrix $\mathbf{S}^{141\times 20}$, then the eigenfilters can be estimated by the right singular vectors $\mathbf{V}_S$ of $\mathbf{S}$ (see Section 3.3). The first $p = 12$ eigenfilters corresponding to a chosen signal subspace dimension are given by $\mathbf{V}_{S1}$ and the magnitude spectra are shown in Figure 3.11, together with the one obtained from the noisy matrix $\mathbf{X}$ using a single white noise realization and SNR $= 5$ dB.

Clearly, the bandpass characteristic of the eigenfilters of the clean signal are concentrated near the first three formants (see Figure 2.7(a)), where especially the mainlobes of the two first filters are centered around the first formant (615 Hz), and the next two filters have mainlobes

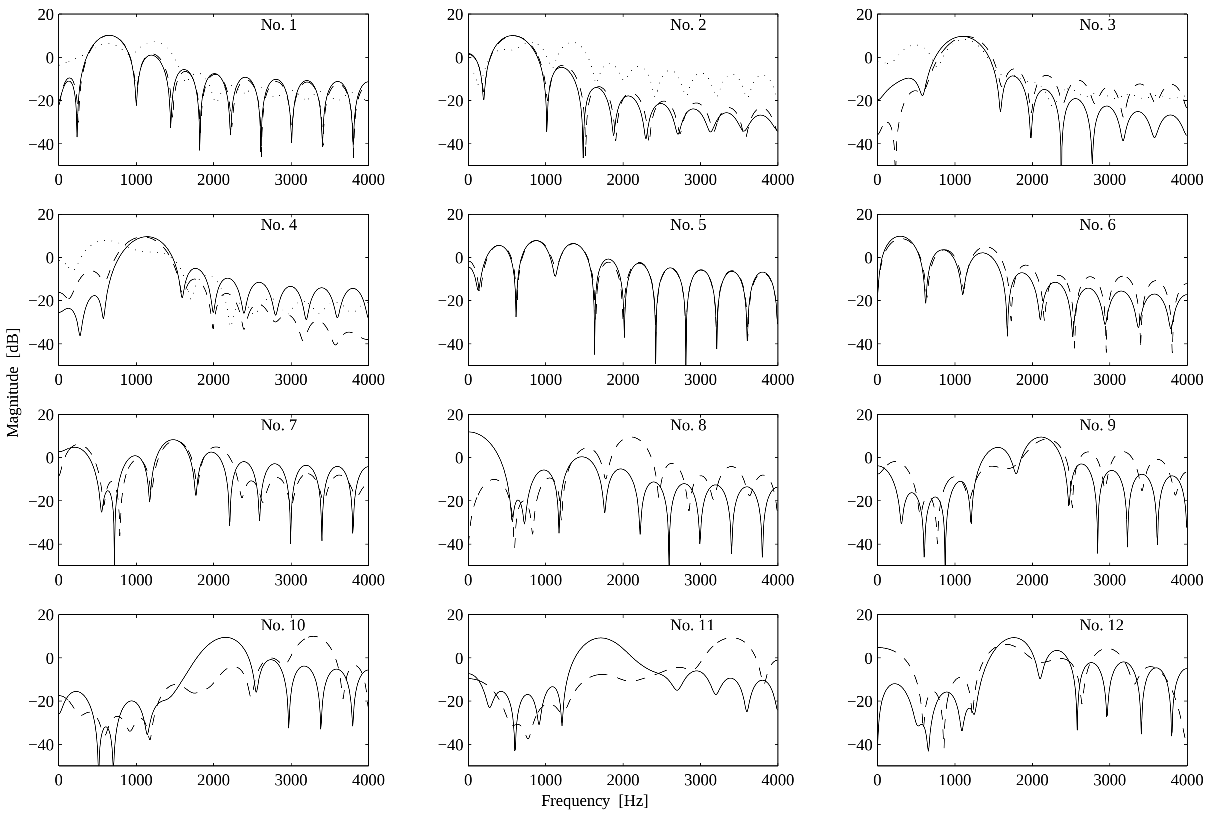


Figure 3.11 (solid) The first 12 eigenfilters $\mathbf{V}_{S1}$ of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples. (dashed) The corresponding eigenfilters obtained from the noisy data matrix $\mathbf{X}$ using a white noise realization and SNR=5dB. (dotted) The canonical vectors (filters) associated with the 4-dimensional intersection of the clean and noisy signal subspace.

centered around the second formant (1185 Hz). For the remaining eigenfilters, the magnitude spectra are more complex.

The related example in Section 3.3.3, showing the subspace angles between the clean and noisy signal subspace, indicates that in the noisy case, the eigenfilters corresponding to the dominant eigenvalues will be close to the true ones, and that the last few eigenfilters will be totally different. This is confirmed by Figure 3.11, where, e.g., filter number 10 and 11 have mainlobes near the fourth formant.

On the first four plots in the Figure, the canonical vectors (filters) associated with the 4-dimensional signal subspace intersection (see Figure 3.7(a)) are shown (without ordering), and obviously the combined filters will match the one obtained from the four dominant eigenfilters. *Thus, a subspace intersection means that the most important part of the noise-free space can be represented exactly.* Note that the canonical vectors form a basis for the intersection and will in general not match the individual eigenfilters.

If the noisy eigenfilters are averaged over 100 independent white noise realizations as shown in Figure 3.12, then all the mainlobes are close to the one obtained from the clean signal. However, since the eigenvectors have length one, the average of the ith eigenfilter $\hat{\mathbf{v}}_{X,i}$ will in general satisfy $E\{\|\hat{\mathbf{v}}_{X,i}\|_2\} \leq 1$. This can be observed in the figure as a lower magnitude, where the level indicates the variance on the eigenfilter estimate.

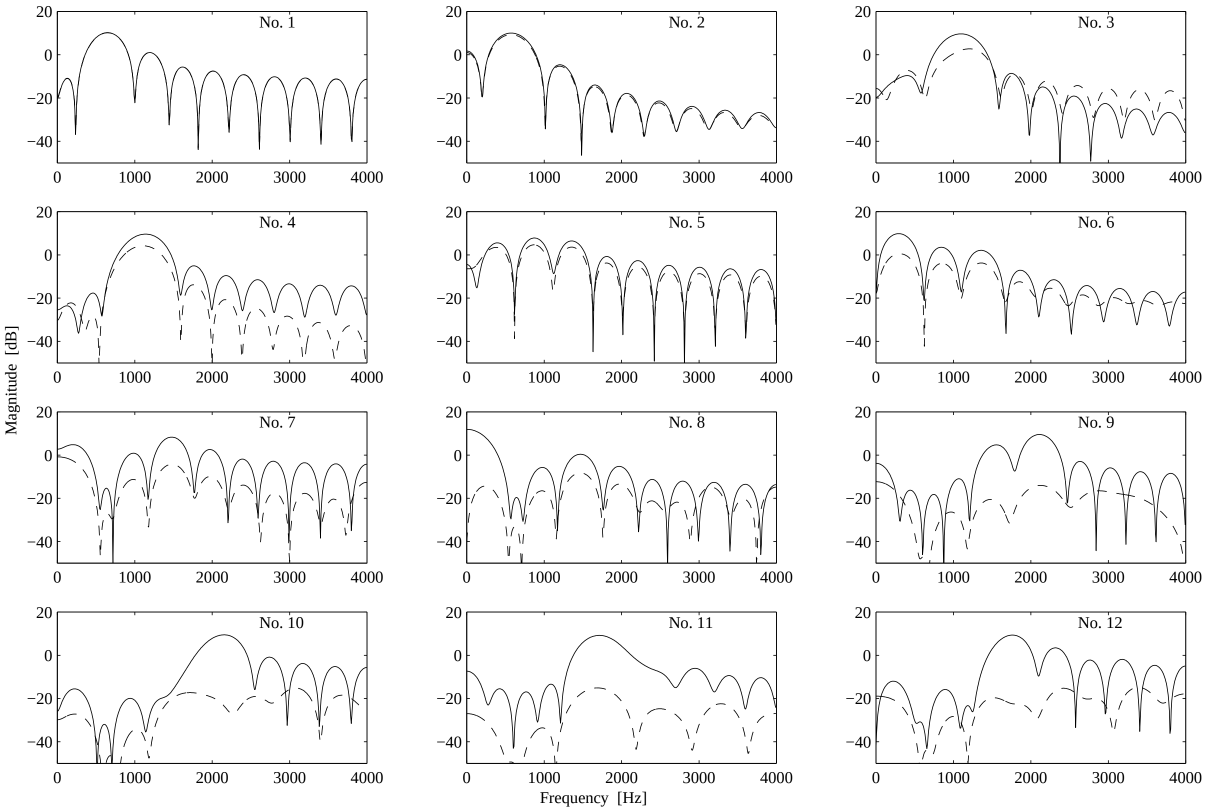


Figure 3.12 (solid) The first 12 eigenfilters $\mathbf{V}_{S1}$ of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples. (dashed) The corresponding averaged eigenfilters obtained from the noisy data matrix $\mathbf{X}$ using 100 independent white noise realizations and SNR=5dB.

3.6 Colored Noise

In the discussed subspace techniques, it has been assumed that the noise is white. If the additive noise $\mathbf{N} \in \mathbb{R}^{m\times n}$ is colored, $\mathbf{N}^T\mathbf{N} \neq \sigma^2_{noise}\mathbf{I}_n$, then a *prewhitening* transformation can be applied to the data matrix [99, page 137].

Assuming the sample correlation matrix $\hat{\mathbf{R}}_n$ of the noise is known, then the corresponding Cholesky factorization or QR decomposition is given by

$$m\hat{\mathbf{R}}_n = \mathbf{N}^T\mathbf{N} = \mathbf{R}^T\mathbf{R} \quad \text{or} \quad \mathbf{N} = \mathbf{QR} \tag{3.84}$$

where $\mathbf{R} \in \mathbb{R}^{n\times n}$ is the upper triangular Cholesky factor and $\mathbf{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_n$. Thus, assuming that $\mathbf{N}$ has full rank such that $\mathbf{R}$ is nonsingular, the signal and noise subspaces can still be recovered by considering the SVD of the *prewhitened* data matrix

$$\mathbf{XR}^{-1} = \mathbf{SR}^{-1} + \mathbf{NR}^{-1} = \mathbf{SR}^{-1} + \mathbf{Q} \tag{3.85}$$

which satisfies Assumption 3.3 in the previously analysis in Section 3.4. This transformation corresponds to a filtering operation and does not change the nature of the linear model for the speech signal (3.1) while it diagonalizes the correlation matrix of the noise ($\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_n$) such that $\sigma^2_{noise} = 1$. Note that in certain applications, the correlation matrix of the noise is rank deficient. However, as shown in [49], it is still possible to define a correct prewhitener in this case.

The consequence of prewhitening in signal subspace methods is illustrated with an example, where the reference sentence is contaminated by an AR(1,-0.7) noise process (SNR = 5 dB). The 10th order LPC-based magnitude spectra for clean speech frames consisting of 160 samples and the one obtained after prewhitening with the first noise frame are considered. Figure 3.13 shows an voiced and unvoiced frame together with the magnitude spectra of the noise process. Obviously, the magnitude of the prewhitened speech are rescaled in accordance with the noise spectrum, i.e., the lower part of the spectrum is attenuated compared to the upper part. This effect is even more clear in Figure 3.14, where magnitude spectra versus time for the first part of the reference sentence are shown before and after prewhitening.

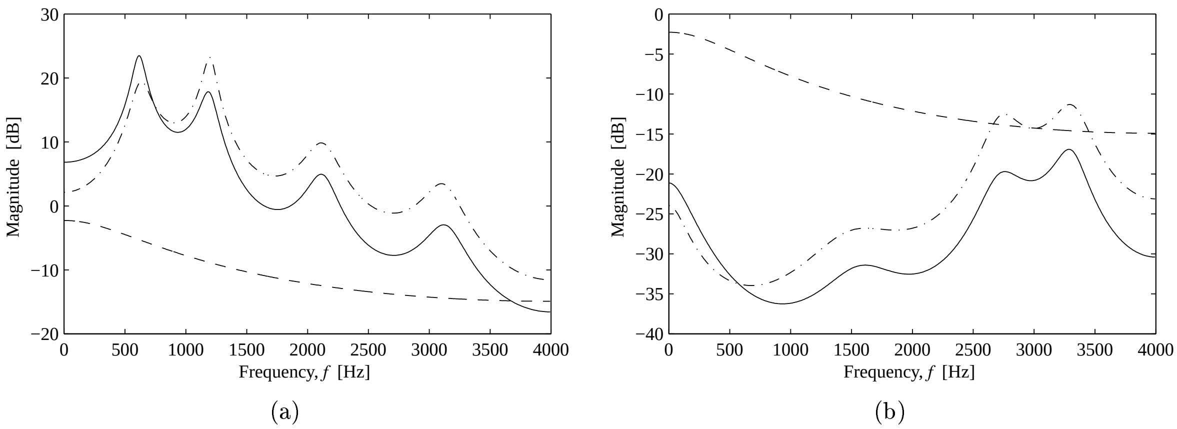


Figure 3.13 (a) LPC-based magnitude spectra for voiced speech frame (solid), AR(1,-0.7) noise process (dashed) and the speech prewhitened with the noise frame (dash-dot). (b) As (a) using unvoiced speech frame.

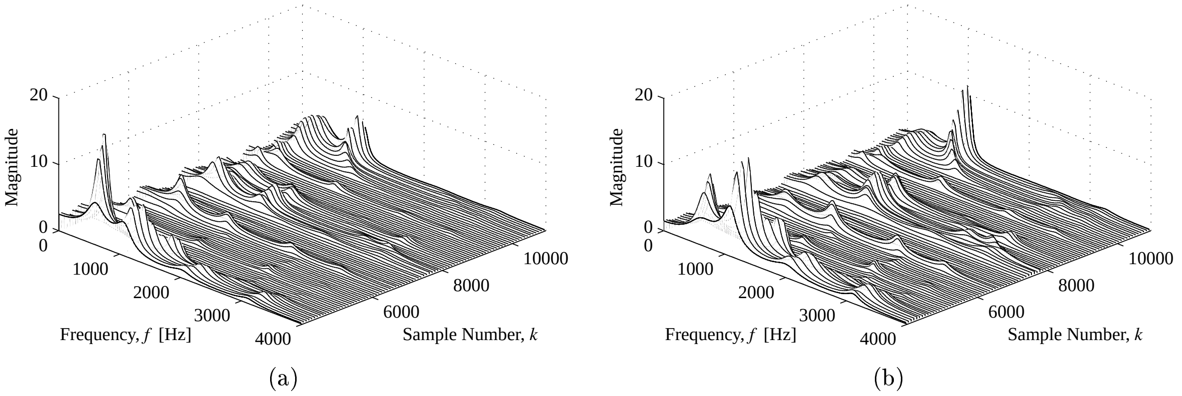


Figure 3.14 (a) LPC-based magnitude spectra versus time for the first part of the reference sentence. (b) As (a) after prewhitening with AR(1,-0.7) noise process (SNR=5dB).

In signal subspace methods, the dominant part of the magnitude spectrum is considered as the low-rank clean signal contaminated by noise, so in general, *the effect of prewhitening is a (maybe large) bias of the signal subspace.*

In the speech enhancement application, it is assumed that the noise matrix $\mathbf{N}$ can be estimated in periods without speech, so if the noise is stationary, then estimation can be performed from an initial segment of the noisy signal which was recorded before speech was present. When the noise is not stationary, a speech/noise detector must be used, and the correlation matrix of the noise is estimated and updated from nonspeech frames of the noisy signal. Thus, the noise reduction problem can always be converted to the case with uncorrelated or white noise.

The requirement of stationary noise for at least a sentence length (about 4 s) has been tested for noise (unit power) measured in a car cabin (see Figure 2.12). Segments of noise is organized in a data matrix $\mathbf{N} \in \mathbb{R}^{141\times 20}$, and the singular spectrum is found before and after prewhitening with the first noise frame as shown in Figure 3.15. All the singular values of the latter spectrum[3] is close to the theoretical value $\sigma_{noise} = 1$, and independent of the time, i.e., *prewhitening can be expected to work well in this environment*. Note, that before prewhitening, the condition numbers of $\mathbf{N}$ are less than 1000 with mean value around 500.

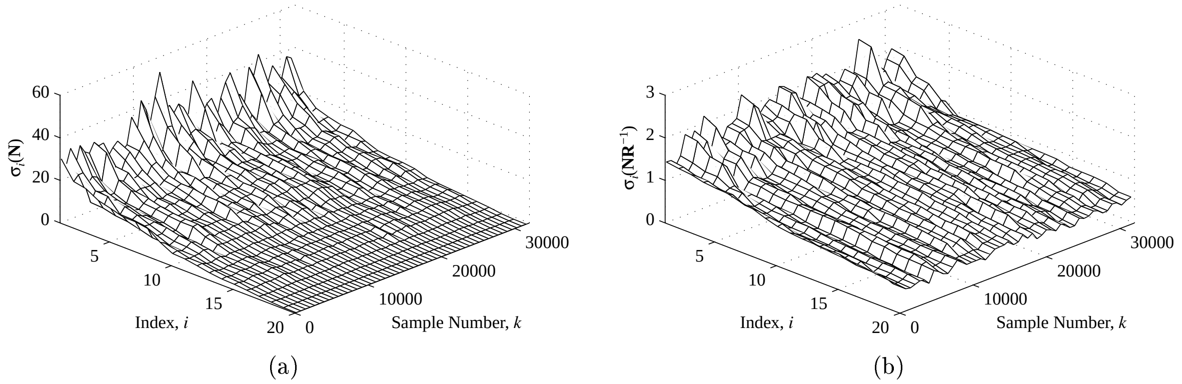


Figure 3.15 (a) Singular spectra versus time for noise measured in car cabin (see Figure 2.12) and normalized to have unit power. (b) The singular spectra after prewhitening with the first noise frame.

[3]This result closely match a similar experiment using the random generator in MATLAB.

3.6.1 Quotient Singular Value Decomposition

One problem concerning the prewhitening transformation, is the numerical computation of the SVD of $\mathbf{XR}^{-1}$. An explicit formation of the inverse $\mathbf{R}^{-1}$ followed by the explicit calculation of the product $\mathbf{XR}^{-1}$ may result in dramatic loss of numerical accuracy in the data. This can be avoided by using the *quotient* SVD, which delivers the required factorizations without forming the quotients and products. Another issue, which may be even more important in practice, is that it is very complicated to update the matrix $\mathbf{XR}^{-1}$ when $\mathbf{X}$ and $\mathbf{N}$ are updated, e.g., in a recursive application. Therefore, it is better to use a decomposition of the matrix pair $(\mathbf{X}, \mathbf{N})$ instead, which allows each matrix to be updated individually.

One possible generalization of the SVD to any two matrices $\mathbf{X} \in \mathbb{R}^{m_X \times n}$ and $\mathbf{N} \in \mathbb{R}^{m_N \times n}$ is the Quotient Singular Value Decomposition (QSVD), which is also called the generalized SVD [39, page 471]. While the ordinary SVD was discovered more than 100 years ago, the QSVD was first studied by Van Loan [119] and Paige and Saunders [90]. Thus, the QSVD is much more recent and is only one member of an infinite set of generalizations of the SVD for any number of matrices [81].

THEOREM 3.7 *Quotient Singular Value Decomposition. If* $\mathbf{X} \in \mathbb{R}^{m_X \times n}$ *with* $m_X \geq n$ *and* $\mathbf{N} \in \mathbb{R}^{m_N \times n}$*, then matrices* $\mathbf{U}_X \in \mathbb{R}^{m_X \times n}$ *and* $\mathbf{U}_N \in \mathbb{R}^{m_N \times n}$ *exist with orthonormal columns and an invertible* $\mathbf{Z} \in \mathbb{R}^{n \times n}$ *such that*

$$\mathbf{X} = \mathbf{U}_X \mathbf{C} \mathbf{Z}^{-1} \tag{3.86}$$

$$\mathbf{N} = \mathbf{U}_N \mathbf{S} \mathbf{Z}^{-1} \tag{3.87}$$

where

$$\mathbf{C} = \operatorname{diag}(c_1, \ldots, c_n) , \qquad 1 \geq c_1 \geq \ldots \geq c_n \geq 0 \tag{3.88}$$

$$\mathbf{S} = \operatorname{diag}(s_1, \ldots, s_q) , \qquad 1 \geq s_q \geq \ldots \geq s_1 \geq 0 , \quad q = \min\{m_N, n\} \tag{3.89}$$

and

$$\mathbf{C}^T\mathbf{C} + \mathbf{S}^T\mathbf{S} = \mathbf{I}_n \tag{3.90}$$

PROOF. For the proof, see [39, page 471]. □

The quotients $\delta_i = c_i/s_i$ are called the quotient singular values of the matrix pair $(\mathbf{X}, \mathbf{N})$

$$\boldsymbol{\Delta} = \mathbf{C}\mathbf{S}^{-1} = \operatorname{diag}\left(\frac{c_1}{s_1}, \ldots, \frac{c_q}{s_q}\right) , \qquad \frac{c_1}{s_1} \geq \ldots \geq \frac{c_q}{s_q} \geq 0 \tag{3.91}$$

and the matrix $\mathbf{Z}$ is called the quotient singular matrix of the matrix pair $(\mathbf{X}, \mathbf{N})$. It is often convenient to define $\boldsymbol{\Theta}^T = \mathbf{Z}^{-1}$, i.e.,

$$\boldsymbol{\Theta}^T \mathbf{Z} = \mathbf{I}_n \tag{3.92}$$

$$\mathbf{X}^T\mathbf{X}\mathbf{Z} = \boldsymbol{\Theta}\mathbf{C}^T\mathbf{C} \tag{3.93}$$

$$\mathbf{N}^T\mathbf{N}\mathbf{Z} = \boldsymbol{\Theta}\mathbf{S}^T\mathbf{S} \tag{3.94}$$

If the QSVD of a rank deficient matrix $\mathbf{X} \in \mathbb{R}^{m_X \times n}$ and a full rank matrix $\mathbf{N} \in \mathbb{R}^{m_N \times n}$ is given by Theorem 3.7, and p is defined by

$$\operatorname{rank}(\mathbf{X}) = p < n \tag{3.95}$$

then it is possible to partition the QSVD of $(\mathbf{X}, \mathbf{N})$ as follows

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix} \tag{3.96}$$
$$\mathbf{N} = \begin{pmatrix} \mathbf{U}_{N1} & \mathbf{U}_{N2} \end{pmatrix} \begin{pmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix}$$

where $\mathbf{U}_{X1} \in \mathbb{R}^{m_X \times p}$, $\mathbf{U}_{N1} \in \mathbb{R}^{m_N \times p}$, $\mathbf{C}_1 \in \mathbb{R}^{p \times p}$, $\mathbf{S}_1 \in \mathbb{R}^{p \times p}$ and $\mathbf{\Theta}_1 \in \mathbb{R}^{n \times p}$. Thus, like the SVD, the QSVD can reveal the rank but of the matrix $\mathbf{X}\mathbf{N}^+$, where $\mathbf{N}^+$ is the pseudoinverse of $\mathbf{N}$, i.e.,

$$\mathbf{X}\mathbf{N}^+ = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1\mathbf{S}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{N1}^T \\ \mathbf{U}_{N2}^T \end{pmatrix} \tag{3.97}$$

Another useful formulation of the QSVD is obtained from the QR-decomposition of $\mathbf{\Theta}$, i.e.,

$$\mathbf{\Theta}^T = \mathbf{L}\mathbf{V}^T \quad \text{and} \quad \begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{11}\mathbf{V}_1^T \\ \mathbf{L}_{21}\mathbf{V}_1^T + \mathbf{L}_{22}\mathbf{V}_2^T \end{pmatrix} \tag{3.98}$$

where $\mathbf{L} \in \mathbb{R}^{n \times n}$ is lower triangular and $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal. The partitioning of $\mathbf{L}$ and $\mathbf{V}$ are similar to (3.96).

3.6.2 Subspace Methods and the QSVD

Now, consider the situation where the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ defined by (3.8) consists of the low-rank signal matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ added *colored* noise, and assume that the noise-only matrix $\mathbf{N} \in \mathbb{R}^{m \times n}$ can be estimated in periods without speech.

Using the QR decomposition of $\mathbf{N}$ (3.84) and Theorem 3.7, it is observed that

$$\begin{aligned} \mathbf{X}\mathbf{R}^{-1} &= \mathbf{X}\mathbf{N}^+\mathbf{Q} \\ &= \mathbf{X}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{Q} \\ &= \mathbf{U}_X\mathbf{C}\mathbf{Z}^{-1}(\mathbf{Z}^{-T}\mathbf{S}^T\mathbf{U}_N^T\mathbf{U}_N\mathbf{S}\mathbf{Z}^{-1})^{-1}\mathbf{Z}^{-T}\mathbf{S}^T\mathbf{U}_N^T\mathbf{Q} \\ &= \mathbf{U}_X\mathbf{C}\mathbf{S}^{-1}\mathbf{U}_N^T\mathbf{Q} \\ &= \mathbf{U}_X\mathbf{\Delta}\bar{\mathbf{U}}_N^T \end{aligned} \tag{3.99}$$

where the matrix $\bar{\mathbf{U}}_N$ is orthogonal and can be rewritten by use of (3.84) and (3.87) as

$$\begin{aligned} \bar{\mathbf{U}}_N^T &= \mathbf{U}_N^T\mathbf{Q} \\ &= (\mathbf{S}^{-1}\mathbf{Z}^T\mathbf{N}^T)\mathbf{Q} \\ &= (\mathbf{R}\mathbf{Z}\mathbf{S}^{-1})^T \end{aligned} \tag{3.100}$$

Hence, working with the QSVD of $(\mathbf{X}, \mathbf{N})$ and the matrix $\mathbf{Q}$ is mathematically equivalent to working with the SVD of the prewhitened data matrix $\mathbf{X}\mathbf{R}^{-1}$. It is important to note that when the rank of $\mathbf{N}$ is less than n, then the SVD of $\mathbf{X}\mathbf{N}^+$ does not always correspond to the QSVD [9, page 158].

In the white-noise case with $\mathbf{R}^T\mathbf{R} = \sigma_{noise}^2\mathbf{I}_n$, the QSVD yields the ordinary SVD of $\mathbf{X}$ which can be seen from (3.99) with $\mathbf{R} = \sigma_{noise}\mathbf{I}_n$, i.e.,

$$\begin{aligned} \mathbf{X} &= \mathbf{U}_X\mathbf{\Delta}\bar{\mathbf{U}}_N^T\sigma_{noise} \\ &= \mathbf{U}_X(\mathbf{\Delta}\sigma_{noise})(\mathbf{S}^{-1}\sigma_{noise}\mathbf{Z}^T) \end{aligned} \tag{3.101}$$

Given the dimension $p < n \leq m$ of the signal subspace, a similar partitioning of the QSVD as defined by (3.96) is

$$\begin{aligned}\mathbf{X} &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix} \\ \mathbf{N} &= \begin{pmatrix} \mathbf{U}_{N1} & \mathbf{U}_{N2} \end{pmatrix} \begin{pmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix}\end{aligned} \tag{3.102}$$

and the prewhited data matrix $\mathbf{XR}^{-1}$ or equivalently the normalized data matrix $\mathbf{XN}^{+}$ is

$$\mathbf{XN}^{+} = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1\mathbf{S}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2\mathbf{S}_2^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{N1}^T \\ \mathbf{U}_{N2}^T \end{pmatrix} \tag{3.103}$$

Note, that in the noise-only case with $\mathbf{X} = \mathbf{N}$, all the quotient singular values are equal to one, i.e., $\delta_{noise}^2 = 1$, so in general the $n-p$ smallest quotient singular values $\mathbf{C}_2\mathbf{S}_2^{-1}$ in Equation (3.103) will be close to one depending on the matrix dimensions and the statistical structure of the pure signal matrix $\mathbf{S}$.

3.7 Summary

Basic subspace techniques have been introduced, and the theory is illustrated by speech related examples, which brings its own surprises and insights.

It is shown that no gab can be expected in the singular spectrum of noisy/clean speech frames, i.e., there is a high probability of a subspace swap when noise is added. However, this is not crucial for speech signals since the dimension of the signal subspace is chosen at a point with almost equal singular values.

The conditions that allow us to derive the speech signal from the SVD of the noisy data matrix has been discussed, and the conclusion is that both matrix dimensions should approach infinity in order to obtain the best performance. This is not possible due to the non-stationarity of speech signals, and considerations concerning a proper choice of dimensions are given.

The signal subspace approach has been interpreted as filter operations, i.e., in the frequency domain, as an alternative to the standard linear algebra formulation. This is very illustrative and shows that subspace methods can be viewed as a number of bandpass filters.

It has been shown that for a prober choice of signal subspace dimension, the signal subspaces obtained from the clean signal and the corresponding noisy signal will have an intersection, which means that the most important part of the noise-free space can be represented exactly.

The case with colored broad-band noise have been considered, and it is illustrated that the effect of prewhitening is a noise dependent bias of the signal subspace. Furthermore, prewhitening is based on the assumption of stationary noise, and it is shown that this is satisfied for noise measured in a car cabin. Finally, it is discussed how combined prewhitening and subspace decomposition can be formulated by means of the QSVD.

Chapter 4

Linear Signal Estimators

One approach for *nonparametric* speech enhancement is linear estimation of the clean signal from the noisy signal using signal subspace methods. Thus, the vector space of the noisy signal is decomposed into a signal subspace and a complementary orthogonal noise subspace, and estimation is performed from vectors in the signal subspace only, since the orthogonal noise subspace contains no (or in practice, little) signal information. Obviously, strategies for estimating the signal subspace dimension must be considered.

Since speech signals are nonstationary, a time varying estimator must be used. Such an estimator provides nonstationary residual noise with annoying noticeable tonal characteristics referred to as *musical noise* [123]. The most well known example for this situation is the musical noise obtained in the spectral subtraction approach. For this reason, both the level and structure of the residual noise must be taken into account, when designing a noise reduction system.

Several estimators from the litterature are presented in a unified notation, and practical implementations based on the data matrix are given. Comparisons between the different estimators provide information on the improvement in the enhanced speech quality that can be gained with each estimator, and the practical behavior of the estimators are compared with the optimal ones. Finally, a filter formulation and frame based implementation of the speech enhancement methods are discussed, where a number of practical aspects are considered.

4.1 Linear Signal Estimators

Different estimators are summarized here, and a detailed treatment of each one follows in the next subsections. Comparison and discussion of the estimators is postponed to the last two subsections.

A straightforward and simple solution to the estimation problem is obtained by use of the *Least Squares* (LS) criterion, which minimizes the squared fitting errors between the noisy measurements $\mathbf{x} \in \mathbb{R}^m$ defined by Equation (3.5) and the linear low order speech model $\mathbf{s} = \mathbf{H}\boldsymbol{\theta}$ defined by Equation (3.1)

$$\min_{\mathbf{H},\boldsymbol{\theta}} (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) \tag{4.1}$$

or the equivalent matrix formulation using the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ defined by Equation (3.8), and a low rank-p model $\mathbf{S}_p$ representing the speech, i.e.,

$$\min_{\mathrm{rank}(\mathbf{S}_p)=p} \mathrm{tr}\left((\mathbf{X} - \mathbf{S}_p)^T (\mathbf{X} - \mathbf{S}_p)\right) \tag{4.2}$$

The solution is easily obtained without any statistical knowledge about the signals.

Assume now that the estimator $\hat{\mathbf{s}} \in \mathbb{R}^m$ of the pure signal vector $\mathbf{s}$ is constrained to be a *linear function* of the measurement vector $\mathbf{x} \in \mathbb{R}^m$

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} \tag{4.3}$$

where $\mathbf{W} \in \mathbb{R}^{m\times m}$ is a filter matrix, then the *Linear Minimum Mean-Squared Error* (LMMSE) estimation problem is to find the matrix $\mathbf{W}$ that minimizes the mean-squared error between $\mathbf{s}$ and the linear estimator $\hat{\mathbf{s}}$

$$\min_{\mathbf{W}} \mathrm{tr} E\{(\mathbf{W}\mathbf{x} - \mathbf{s})(\mathbf{W}\mathbf{x} - \mathbf{s})^T\} \tag{4.4}$$

This theory produces the *Wiener-Hopf equations* as the fundamental design equations, i.e., the correlation (or second order) properties of the noisy signal and the noise process are required.

In practice, this information is not available and has to be estimated from the noisy data. Under stationary and ergodic conditions, the ensemble average operator $E\{\cdot\}$ can be implemented as the mean value of several time shifted vectors, i.e., by use of the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ and the signal-only matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$, c.f., Equation (3.8). This gives the *Minimum Variance* (MV) estimator as defined in [82]

$$\min_{\mathbf{W}} \mathrm{tr}\left((\mathbf{X}\mathbf{W} - \mathbf{S})^T(\mathbf{X}\mathbf{W} - \mathbf{S})\right) \tag{4.5}$$

which converges asymptotically to the LMMSE estimator as the number of vectors $m \to \infty$ (see Section 3.4.1). Note that $\hat{\mathbf{S}} = \mathbf{X}\mathbf{W}$ and $\mathbf{W} \in \mathbb{R}^{n\times n}$ in this case.

The residual signal $\mathbf{r} = \hat{\mathbf{s}} - \mathbf{s} \in \mathbb{R}^m$ minimized in the above methods represents signal distortion $\mathbf{r}_s$ and residual noise $\mathbf{r}_n$

$$\begin{aligned} \mathbf{r} &= \hat{\mathbf{s}} - \mathbf{s} \\ &= (\mathbf{W} - \mathbf{I}_m)\mathbf{s} + \mathbf{W}\mathbf{n} \\ &= \mathbf{r}_s + \mathbf{r}_n \end{aligned} \tag{4.6}$$

Since both terms can not be simultaneously minimized, a *Time Domain Constrained* (TDC) estimator is proposed by Y. Ephraim and H. L. Van Trees in [32], which keep the residual noise energy $\epsilon_n^2 = \mathrm{tr}E\{\mathbf{r}_n\mathbf{r}_n^T\}$ below some threshold while minimizing the signal distortion energy $\epsilon_s^2 = \mathrm{tr}E\{\mathbf{r}_s\mathbf{r}_s^T\}$

$$\min_{\mathbf{W}} \epsilon_s^2 \quad \text{subject to} \quad \epsilon_n^2 \le \alpha m \nu_{noise}^2 \tag{4.7}$$

where $0 \le \alpha \le 1$ and ν_{noise}^2 is the noise power. For speech signals, this estimation criterion will control the more perceptually harmful component, i.e., the nonstationary residual noise (musical noise) that is intolerable by the human auditory system, while minimizing the signal distortion. The optimal linear estimator in this sense is a Wiener filter with adjustable input noise level.

The *Spectral Domain Constrained* (SDC) estimator [32] is a generalization of the TDC estimator which minimizes the signal distortion while keeping the energy of the residual noise in each spectral component, defined by the eigenfilters $\mathbf{q}_i$, below some given threshold

$$\min_{\mathbf{W}} \epsilon_s^2 \quad \text{subject to} \quad \begin{cases} E\{|\mathbf{q}_i^T\mathbf{r}_n|^2\} \le \alpha_i \nu_{noise}^2\,, & i = 1,\ldots,p \\ E\{|\mathbf{q}_i^T\mathbf{r}_n|^2\} = 0\,, & i = p+1,\ldots,m \end{cases} \tag{4.8}$$

where p is the signal subspace dimension. This strategy allows shaping of the spectrum of the residual noise, e.g., making it similar to that of the speech signal. Thus, more noise is permitted to accompany high energy spectral components of the clean signal. The optimal filter in this sense has a structure that contains all the above methods.

The *empirical* versions of the TDC and SDC estimators are obtained from the matrix formulation of Equation (4.6), i.e.,

$$\begin{aligned}\mathbf{R} &= \hat{\mathbf{S}} - \mathbf{S} \\ &= \mathbf{S}(\mathbf{W} - \mathbf{I}_n) + \mathbf{N}\mathbf{W} \\ &= \mathbf{R}_S + \mathbf{R}_N\end{aligned} \tag{4.9}$$

so in the TDC case, the following definitions of the squared residual noise error $e_n^2 = \mathrm{tr}(\mathbf{R}_N^T \mathbf{R}_N)$ and signal distortion error $e_s^2 = \mathrm{tr}(\mathbf{R}_S^T \mathbf{R}_S)$ are used in the constrained minimization

$$\min_{\mathbf{W}} e_s^2 \quad \text{subject to} \quad e_n^2 \leq \alpha n \sigma_{noise}^2 \tag{4.10}$$

and similarly in the SDC case

$$\min_{\mathbf{W}} e_s^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{R}_N \mathbf{v}_{X,i}\|_2^2 \leq \alpha_i \sigma_{noise}^2\,, & i = 1, \ldots, p \\ \|\mathbf{R}_N \mathbf{v}_{X,i}\|_2^2 = 0\,, & i = p+1, \ldots, n \end{cases} \tag{4.11}$$

where $\{\mathbf{v}_{X,i}\}_1^n$ are the right singular vectors of $\mathbf{X}$.

Furthermore, if a known probability distribution of the involved signals is assumed, the theory of *Maximum Likelihood* (ML) can be used. The, in general, non-linear optimization problem of fitting a model to noisy measurements turns out to be linear in the case of Gaussian noise, i.e., in the observation model $x : \mathcal{N}[\mathbf{H}\boldsymbol{\theta}, \mathbf{R}]$. However, then the problem is equivalent to the LS problem in the case of white noise, or weighted LS in the case of colored noise.

4.1.1 Maximum Likelihood Estimator

A Maximum Likelihood (ML) principle to estimate the signal subspace and the clean signal from the noisy data matrix is given in [99, page 252], when the noise vectors (the columns of the noise-only matrix) are zero-mean, independently and identically Gaussian distributed. The independence is not satisfied for the Toeplitz data matrix (3.8) due to the structure of time shifted column vectors, but the method is summarized to show the connection with the methods in the next sections. However, in the array processing case where the columns in the data matrix correspond to different sensors, the independence is typically satisfied.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the noisy data matrix defined by Equation (3.6), where the columns are a sequence of n measurement vectors $\{\mathbf{x}_i\}_1^n$ of dimension m with observation noise vectors $\{\mathbf{n}_i\}_1^n$ drawn *independently* from an $\mathcal{N}[\mathbf{0}, \nu_{noise}^2 \mathbf{I}]$ distribution. Then, the probability density function $f_{\underline{\mathbf{x}}_i}(\mathbf{x}_i)$ of the ith vector $\mathbf{x}_i$ is $\mathcal{N}[\mathbf{s}_i, \nu_{noise}^2 \mathbf{I}]$, and the joint distribution of $\{\mathbf{x}_i\}_1^n$ has density $f_{\underline{\mathbf{x}}_1 \cdots \underline{\mathbf{x}}_n}(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ given by

$$\begin{aligned} f_{\underline{\mathbf{x}}_1 \cdots \underline{\mathbf{x}}_n}(\mathbf{x}_1, \cdots, \mathbf{x}_n) &= \prod_{i=1}^n f_{\underline{\mathbf{x}}_i}(\mathbf{x}_i) \\ &= (2\pi\nu_{noise}^2)^{-mn/2} \exp\left\{ -\frac{1}{2\nu_{noise}^2} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{s}_i)^T (\mathbf{x}_i - \mathbf{s}_i) \right\} \end{aligned} \tag{4.12}$$

The log-likelihood function for $\{\mathbf{x}_i\}_1^n$ is

$$L = \ln f_{\underline{\mathbf{x}}_1 \cdots \underline{\mathbf{x}}_n}(\mathbf{x}_1, \cdots, \mathbf{x}_n) = -\frac{mn}{2} \ln(2\pi\nu_{noise}^2) - \frac{1}{2\nu_{noise}^2} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{s}_i)^T (\mathbf{x}_i - \mathbf{s}_i) \tag{4.13}$$

The maximum likelihood method for identifying the signal first assumes that the p-dimensional signal subspace $\langle \mathbf{H} \rangle$ in the linear statistical model $\mathbf{s}_i = \mathbf{H}\boldsymbol{\theta}_i$ is *known*. Then the maximum likelihood estimate $\hat{\mathbf{S}}_{ML}$ of the clean signal matrix will maximize the log-likelihood under the constraint $\tilde{\mathbf{H}}^T\mathbf{S} = \mathbf{0}$, i.e.,

$$\max_{\mathbf{S}} L \quad \text{subject to} \quad \tilde{\mathbf{H}}^T\mathbf{S} = \mathbf{0} \tag{4.14}$$

or equivalently

$$\min_{\mathbf{S}}(-2\nu_{noise}^2 L) \quad \text{subject to} \quad 2\tilde{\mathbf{H}}^T\mathbf{S} = \mathbf{0} \tag{4.15}$$

The solution is found by forming the following Lagrangian

$$\mathcal{L} = \text{constant} + \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{s}_i)^T(\mathbf{x}_i - \mathbf{s}_i) + 2\sum_{i=1}^{n}\boldsymbol{\gamma}_i^T\tilde{\mathbf{H}}^T\mathbf{s}_i \tag{4.16}$$

and minimize $\mathcal{L}$ with respect to $\mathbf{s}_i$. The gradient equation is

$$\frac{\partial\mathcal{L}}{\partial\mathbf{s}_i} = -2(\mathbf{x}_i - \mathbf{s}_i) + 2\tilde{\mathbf{H}}\boldsymbol{\gamma}_i = \mathbf{0} \tag{4.17}$$

which gives

$$\mathbf{s}_i = \mathbf{x}_i - \tilde{\mathbf{H}}\boldsymbol{\gamma}_i \tag{4.18}$$

where $\boldsymbol{\gamma}_i = (\gamma_{i1}, \cdots, \gamma_{i(n-p)})$ contains the Lagrangians for the columns of $\tilde{\mathbf{H}}$. The constraints are enforced by substitute (4.18) into (4.14) and solving for $\boldsymbol{\gamma}_i$

$$\boldsymbol{\gamma}_i = (\tilde{\mathbf{H}}^T\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^T\mathbf{x}_i \tag{4.19}$$

Finally, the maximum likelihood estimate $\hat{\mathbf{s}}_i$ is given by

$$\hat{\mathbf{s}}_i = (\mathbf{I} - \tilde{\mathbf{H}}(\tilde{\mathbf{H}}^T\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^T)\mathbf{x}_i = (\mathbf{I} - \mathbf{P}_{\tilde{\mathbf{H}}})\mathbf{x}_i = \mathbf{P}_{\mathbf{H}}\mathbf{x}_i \tag{4.20}$$

where $\mathbf{P}_{\mathbf{H}}$ is the projection onto the signal subspace $\langle \mathbf{H} \rangle$.

Now, assume $\tilde{\mathbf{H}}$ *unknown*, then the likelihood must be further maximized with respect to $\tilde{\mathbf{H}}$, i.e., using the estimate $\hat{\mathbf{s}}_i$ in (4.13) and ignoring constants, the resulting log-likelihood function is

$$L = -\frac{1}{2\nu_{noise}^2}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{P}_{\tilde{\mathbf{H}}}\mathbf{x}_i = -\frac{n}{2\nu_{noise}^2}\text{tr}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^T\mathbf{P}_{\tilde{\mathbf{H}}}\right) = -\frac{n}{2\nu_{noise}^2}\text{tr}\left(\hat{\mathbf{R}}_x\mathbf{P}_{\tilde{\mathbf{H}}}\right) \tag{4.21}$$

If the eigendecomposition (3.35) of the sample correlation matrix $\hat{\mathbf{R}}_x \in \mathbb{R}^{m\times m}$ is partitioned as

$$\hat{\mathbf{R}}_x = \frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix}\begin{pmatrix} \boldsymbol{\Sigma}_{X1}^2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{X2}^2 \end{pmatrix}\begin{pmatrix} \mathbf{U}_{X1}^T \\ \mathbf{U}_{X2}^T \end{pmatrix} \tag{4.22}$$

where $\mathbf{U}_{X1} \in \mathbb{R}^{m\times p}$ and $\boldsymbol{\Sigma}_{X1} \in \mathbb{R}^{p\times p}$, then L is bounded as follows

$$L = -\frac{1}{2\nu_{noise}^2}\text{tr}\left(\mathbf{U}_X\boldsymbol{\Sigma}_X^2\mathbf{U}_X^T\mathbf{P}_{\tilde{\mathbf{H}}}\right) \leq -\frac{1}{2\nu_{noise}^2}\sum_{i=p+1}^{m}\sigma_{X,i}^2 \tag{4.23}$$

for any rank $(m-p)$ projector $\mathbf{P}_{\tilde{\mathbf{H}}}$. The maximum is achieved for a projector $\mathbf{P}_{\tilde{\mathbf{H}}}$ onto the subspace $\langle \mathbf{U}_{X2} \rangle$, so the resulting maximum likelihood approximation of the signal subspace is given by

$$\hat{\mathbf{H}}_{ML} = \mathbf{U}_{X1} \quad \Rightarrow \quad \hat{\mathbf{S}}_{ML} = \mathbf{U}_{X1}\mathbf{U}_{X1}^T\mathbf{X} \tag{4.24}$$

The maximum likelihood approximation of the signal subspace is based on statistical reasoning of the noise distribution, and the problem only has a simple solution in the case of (white) Gaussian noise. The obtained solution is identical to the least squares approximation discussed in the next section, indicating the intimate connection between the two principles when assuming Gaussian errors.

4.1.2 Least Squares Estimator

The idea behind Least Squares (LS) is to fit a model to measurements in such a way that the fitting errors between the measurements and the model are minimized. Least squares may be used in linear or nonlinear modelling, and unlike the maximum likelihood method *no statistical assumptions* are made of the additive noise.

Suppose $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the measurement matrix defined by (3.6), where the columns of $\mathbf{S}$ are known to lie in the rank-p subspace $\langle\mathbf{H}\rangle$, but the subspace is unknown. The problem is to estimate the signal subspace $\langle\mathbf{H}\rangle$ and use it to estimate the desired signal $\mathbf{S}$. The squared fitting error between $\mathbf{X}$ and $\mathbf{S} = \mathbf{H}\mathbf{\Theta}$ is defined as

$$\begin{aligned} e^2 &= \|\mathbf{X} - \mathbf{H}\mathbf{\Theta}\|_F^2 \\ &= \mathrm{tr}\left((\mathbf{X} - \mathbf{H}\mathbf{\Theta})^T(\mathbf{X} - \mathbf{H}\mathbf{\Theta})\right) \\ &= \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{H}\boldsymbol{\theta}_i)^T(\mathbf{x}_i - \mathbf{H}\boldsymbol{\theta}_i) \end{aligned} \tag{4.25}$$

which is minimized to obtain the least squares approximation. First assume that $\langle\mathbf{H}\rangle$ is *known*, then the gradient of e^2 with respect to $\mathbf{\Theta}$ is

$$\frac{\partial e^2}{\partial \mathbf{\Theta}} = \begin{pmatrix} \frac{\partial e^2}{\partial \boldsymbol{\theta}_1} & \cdots & \frac{\partial e^2}{\partial \boldsymbol{\theta}_n} \end{pmatrix} = \begin{pmatrix} -2\mathbf{H}^T(\mathbf{x}_1 - \mathbf{H}\boldsymbol{\theta}_1) & \cdots & -2\mathbf{H}^T(\mathbf{x}_n - \mathbf{H}\boldsymbol{\theta}_n) \end{pmatrix} \tag{4.26}$$

The least squares estimate equates the gradient to zero to produce the solutions

$$\hat{\mathbf{\Theta}}_{LS} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X} \tag{4.27}$$

$$\hat{\mathbf{S}}_{LS} = \mathbf{H}\hat{\mathbf{\Theta}}_{LS} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X} \tag{4.28}$$

which of course satisfies the *normal equations* $\mathbf{H}^T\mathbf{H}\mathbf{\Theta} = \mathbf{H}^T\mathbf{X}$. Therefore, if $\mathbf{H}$ has full rank as in our case, there is a unique LS solution. More generally, whatever the rank of $\mathbf{H}$, the matrix $\hat{\mathbf{\Theta}}_{LS} = \mathbf{H}^+\mathbf{X}$ is an LS solution, and it is the solution of minimal F-norm. Here, $\mathbf{H}^+$ is the pseudo-inverse of $\mathbf{H}$ (given by $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ when $\mathbf{H}$ has full rank).

Now assume $\mathbf{H}$ *unknown*, then the squared error e^2 must be further minimized with respect to $\mathbf{H}$. Using the estimate $\hat{\mathbf{\Theta}}_{LS}$ in (4.25) gives

$$\begin{aligned} e^2 &= \mathrm{tr}\left((\mathbf{X} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X})(\mathbf{X} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X})^T\right) \\ &= \mathrm{tr}\left(\mathbf{P}_{\tilde{\mathbf{H}}}\mathbf{X}\mathbf{X}^T\mathbf{P}_{\tilde{\mathbf{H}}}\right) \end{aligned} \tag{4.29}$$

where $\mathbf{P}_{\tilde{\mathbf{H}}}$ is the projection onto the noise subspace $\langle\tilde{\mathbf{H}}\rangle$. Let the eigendecomposition of the sample correlation matrix $\hat{\mathbf{R}}_x \in \mathbb{R}^{m\times m}$ be defined by Equation (4.22), then e^2 is bounded as follows

$$e^2 = \mathrm{tr}\left(\mathbf{P}_{\tilde{\mathbf{H}}}\mathbf{U}_X\mathbf{\Sigma}_X^2\mathbf{U}_X^T\mathbf{P}_{\tilde{\mathbf{H}}}\right) \geq \sum_{i=p+1}^{m}\sigma_{X,i}^2 \tag{4.30}$$

for any rank $(m-p)$ projector $\mathbf{P}_{\tilde{\mathbf{H}}}$. The minimum is achieved for a projector $\mathbf{P}_{\tilde{\mathbf{H}}}$ onto the subspace $\langle\mathbf{U}_{X2}\rangle$, so the resulting least squares approximation of the signal subspace is

$$\hat{\mathbf{H}}_{LS} = \mathbf{U}_{X1} \quad \Rightarrow \quad \hat{\mathbf{S}}_{LS} = \mathbf{U}_{X1}\mathbf{U}_{X1}^T\mathbf{X} \tag{4.31}$$

The relationship expressed in (4.31) shows that the parameters in the model matrix $\mathbf{H}$, e.g., frequencies and damping factors, need not be explicitly known in implementing the LS estimator. An equivalent formulation of (4.31) is obtained by using the SVD of the measurement matrix $\mathbf{X}$

$$\hat{\mathbf{S}}_{LS} = \mathbf{U}_{X1}\mathbf{\Sigma}_{X1}\mathbf{V}_{X1}^T = \mathbf{X}\mathbf{V}_{X1}\mathbf{V}_{X1}^T \tag{4.32}$$

This well-known result is also obtained without assuming a linear model for $\mathbf{S}$, when $\mathbf{X}$ is approximated by a matrix of lower rank $p < n$ in the least squares sense (see Section 3.3.1). The estimate is commonly referred as the *truncated* SVD approach and is, e.g., used in the SVD-based method for estimating the signal components of a noisy data vector proposed by Tufts, Kumarasan and Kirsteins in [116, 115, 65] and in the SVD-based method for speech enhancement proposed by Dendrinos, Bakamidis and Carayannis in [25].

4.1.3 Linear Minimum Mean-Squared Error Estimator

The Linear Minimum Mean-Squared Error (LMMSE) approach minimizes the mean-squared error between the clean signal and a linear function of the measurements.

Let $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} \in \mathbb{R}^m$ be a linear estimator of the pure signal vector $\mathbf{s}$, where $\mathbf{x} \in \mathbb{R}^m$ is defined by Equation (3.5) and $\mathbf{W} \in \mathbb{R}^{m\times m}$ is a filter matrix, and let

$$\epsilon_r^2 = \mathrm{tr}E\{\mathbf{r}\mathbf{r}^T\} \tag{4.33}$$

denote the energy of the residual signal $\mathbf{r} = \hat{\mathbf{s}} - \mathbf{s}$ (4.6). Then the LMMSE estimator is obtained from

$$\min_{\mathbf{W}} \epsilon_r^2 \tag{4.34}$$

which minimizes the mean-squared error between $\mathbf{s}$ and $\hat{\mathbf{s}}$. If Asumption 3.2 on Page 26 is satisfied, i.e., the noise process is uncorrelated with the clean signal, then the energy term is given by

$$\begin{aligned} \epsilon_r^2 &= \mathrm{tr}E\{(\mathbf{W}\mathbf{x} - \mathbf{s})(\mathbf{W}\mathbf{x} - \mathbf{s})^T\} \\ &= \mathrm{tr}\left(\mathbf{W}\mathbf{R}_x\mathbf{W}^T + \mathbf{R}_s - \mathbf{W}\mathbf{R}_s - \mathbf{R}_s\mathbf{W}^T\right) \end{aligned} \tag{4.35}$$

Thus, the filter matrix $\mathbf{W}$ is a stationary feasible point of ϵ_r^2 if it satisfies the gradient equation

$$\frac{\partial \epsilon_r^2}{\partial \mathbf{W}} = 2\mathbf{W}\mathbf{R}_x - 2\mathbf{R}_s = \mathbf{0} \tag{4.36}$$

which has the Wiener-Hopf equations as solution, i.e., the optimal filter is a *Wiener filter*

$$\begin{aligned} \mathbf{W}_{LMMSE} &= \mathbf{R}_s\mathbf{R}_x^{-1} \\ &= \mathbf{R}_s(\mathbf{R}_s + \nu_{noise}^2\mathbf{I}_m)^{-1} \\ &= (\mathbf{R}_x - \nu_{noise}^2\mathbf{I}_m)\mathbf{R}_x^{-1} \end{aligned} \tag{4.37}$$

where Equation (3.11) has been used. Applying the eigendecompositions (3.17) and (3.18) to (4.37), the optimal linear filter can be rewritten as

$$\begin{aligned} \mathbf{W}_{LMMSE} &= \mathbf{Q}_1\mathbf{\Lambda}_{s1}(\mathbf{\Lambda}_{s1} + \nu_{noise}^2\mathbf{I}_p)^{-1}\mathbf{Q}_1^T \\ &= \mathbf{Q}_1(\mathbf{I}_p - \nu_{noise}^2\mathbf{\Lambda}_{x1}^{-1})\mathbf{Q}_1^T \end{aligned} \tag{4.38}$$

Hence, the LMMSE estimator of $\mathbf{s}$ is given by $\hat{\mathbf{s}}_{LMMSE} = \mathbf{W}_{LMMSE}\mathbf{x}$. Note, that the method requires the correlation function of the noisy signal and the noise variance ν_{noise}^2.

4.1.4 Minimum Variance Estimator

One empirical version of the LMMSE approach is the Minimum Variance (MV) estimate as defined by Bart De Moor in [82]. It is the best estimate of the noise-free matrix that can be obtained by making linear combinations of the elements in the noisy data matrix, i.e., the MV estimate is the center of a *matrix ball* of equivalent signal reconstructions. This set of matrices represents the uncertainty in the estimation of the original signal due to the bias of the left singular matrix (see Section 3.4).

Suppose $\mathbf{X} \in \mathbb{R}^{m\times n}$ is a measurement matrix defined by (3.8), with $\mathbf{S}$ satisfying (3.9), then the minimum variance problem is to find the filter matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ that minimizes

$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{S}\|_F^2 \tag{4.39}$$

Assume that $\mathbf{S}$ is known, then the minimum is obtained by zero the derivatives of the squared error

$$\begin{aligned} e_r^2 &= \|\mathbf{XW} - \mathbf{S}\|_F^2 \\ &= \operatorname{tr}\Big((\mathbf{XW} - \mathbf{S})^T(\mathbf{XW} - \mathbf{S})\Big) \\ &= \operatorname{tr}\Big(\mathbf{W}^T\mathbf{X}^T\mathbf{XW} + \mathbf{S}^T\mathbf{S} - \mathbf{W}^T\mathbf{X}^T\mathbf{S} - \mathbf{S}^T\mathbf{XW}\Big) \end{aligned} \tag{4.40}$$

with respect to the elements of $\mathbf{W}$. The solution is given by

$$\mathbf{W}_{MV} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \tag{4.41}$$

Hence, the MV estimate of $\mathbf{S}$ is given by

$$\hat{\mathbf{S}}_{MV} = \mathbf{XW}_{MV} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \tag{4.42}$$

or by means of the SVD of $\mathbf{X} = \mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X^T$

$$\hat{\mathbf{S}}_{MV} = \mathbf{U}_X\mathbf{U}_X^T\mathbf{S} \tag{4.43}$$

which is the orthogonal projection of $\mathbf{S}$ onto the column space of the data matrix $\mathbf{X}$. Observe that $\operatorname{rank}(\hat{\mathbf{S}}_{MV}) = \operatorname{rank}(\mathbf{S}) = p$.

In spite of the fact that $\mathbf{S}$ is not known, it is possible to compute the MV estimate from the SVD of $\mathbf{X}$ if Assumption 3.3 on Page 35 is satisfied. Substituting (3.55) and (3.57) into (4.43) yields the desired MV estimate

$$\begin{aligned} \hat{\mathbf{S}}_{MV} &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{X1}^T \\ \mathbf{U}_{X2}^T \end{pmatrix} \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} (\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1/2}(\mathbf{\Sigma}_{S1}^T\mathbf{U}_{S1}^T + \mathbf{V}_{S1}^T\mathbf{N}^T) \\ \sigma_{noise}^{-1}\mathbf{V}_{S2}^T\mathbf{N}^T \end{pmatrix} \\ &\qquad\qquad \times \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \\ &= \mathbf{U}_{X1}\mathbf{\Sigma}_{S1}^2(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1/2}\mathbf{V}_{X1}^T \\ &= \mathbf{U}_{X1}\mathbf{\Sigma}_{X1}(\mathbf{I}_p - \sigma_{noise}^2\mathbf{\Sigma}_{X1}^{-2})\mathbf{V}_{X1}^T \end{aligned} \tag{4.44}$$

which is a singular value decomposition. This equation can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}}_{MV} = \mathbf{XV}_{X1}(\mathbf{I}_p - \sigma_{noise}^2\mathbf{\Sigma}_{X1}^{-2})\mathbf{V}_{X1}^T \tag{4.45}$$

The left and right singular vectors of this minimum variance estimate are the same as those of the least squares estimate (4.32) but the singular values are different

$$(\mathbf{\Sigma}_{X1}^2 - \sigma_{noise}^2 \mathbf{I}_p)\mathbf{\Sigma}_{X1}^{-1} = \mathbf{\Sigma}_{S1}^2 \mathbf{\Sigma}_{X1}^{-1} = \mathbf{C}\mathbf{\Sigma}_{S1} \tag{4.46}$$

where $\mathbf{C}$ are the cosines of the canonical angles as defined by (3.64).

Notice that the MV estimate requires the quantity σ_{noise}^2, which can be estimated as the average of the diagonal elements of $\mathbf{\Sigma}_{X2}$ (if no signal is present in the noise subspace)

$$\sigma_{noise}^2 = \frac{1}{n-p} \sum_{i=p+1}^{n} \sigma_{X,i}^2 \tag{4.47}$$

or it can be obtained from the noise variance ν_{noise}^2 estimated in periods without speech

$$\sigma_{noise}^2 = m\nu_{noise}^2 \tag{4.48}$$

Note also that under stationary and ergodic conditions, then the MV estimator converges with probability 1 to the LMMSE estimator as $m \to \infty$. The MV approach is used successfully for speech enhancement by Jensen in [62].

4.1.5 Time Domain Constrained Estimator

Based on the optimal LMMSE approach, Ephraim and Van Trees have proposed two perceptually more meaningful estimation criterias in [32], which are summarized in this and the next section, respectively.

Let $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} \in \mathbb{R}^m$ be the linear estimator of the pure signal vector $\mathbf{s}$, then the residual signal $\mathbf{r} = \hat{\mathbf{s}} - \mathbf{s} \in \mathbb{R}^m$ minimized in the above LMMSE method, represents signal distortion $\mathbf{r}_s$ and residual noise (musical noise) $\mathbf{r}_n$

$$\begin{aligned} \mathbf{r} &= \hat{\mathbf{s}} - \mathbf{s} \\ &= (\mathbf{W} - \mathbf{I}_m)\mathbf{s} + \mathbf{W}\mathbf{n} \\ &= \mathbf{r}_s + \mathbf{r}_n \end{aligned} \tag{4.49}$$

Since both terms can not be simultaneously minimized, a Time Domain Constrained (TDC) estimator is proposed in [32], which keeps the residual noise energy

$$\epsilon_n^2 = \mathrm{tr}E\{\mathbf{r}_n\mathbf{r}_n^T\} = \nu_{noise}^2 \mathrm{tr}\left(\mathbf{W}\mathbf{W}^T\right) \tag{4.50}$$

below some threshold while minimizing the signal distortion energy denoted by

$$\epsilon_s^2 = \mathrm{tr}E\{\mathbf{r}_s\mathbf{r}_s^T\} = \mathrm{tr}\left((\mathbf{W} - \mathbf{I}_m)\mathbf{R}_s(\mathbf{W} - \mathbf{I}_m)^T\right) \tag{4.51}$$

The linear estimator with TDC on the residual noise is obtained from

$$\min_{\mathbf{W}} \epsilon_s^2 \quad \text{subject to} \quad \epsilon_n^2 \leq \alpha m \nu_{noise}^2 \tag{4.52}$$

where $0 \leq \alpha \leq 1$ controls the permissible *segmental* noise level and ν_{noise}^2 is the noise power. The value of α is restricted to one, since for $\alpha \geq 1$, the optimal filter which satisfies the constraint and results in minimum (zero) signal distortion is $\mathbf{W} = \mathbf{I}_m$.

Given Assumption 3.2 on Page 26 is satisfied in the following, the optimal estimator in the sense of (4.52) can be found from the Lagrangian

$$\begin{aligned}\mathcal{L}(\mathbf{W},\gamma) &= \epsilon_s^2 + \gamma\left(\epsilon_n^2 - \alpha m \nu_{noise}^2\right) \\ &= \mathrm{tr}\left((\mathbf{W}-\mathbf{I}_m)\mathbf{R}_s(\mathbf{W}-\mathbf{I}_m)^T\right) + \gamma\left(\nu_{noise}^2 \mathrm{tr}(\mathbf{W}\mathbf{W}^T) - \alpha m \nu_{noise}^2\right)\end{aligned} \tag{4.53}$$

where $\gamma \geq 0$ is the Lagrange multiplier. Thus, the filter matrix $\mathbf{W}$ is a stationary feasible point if it satisfies the gradient equation

$$\frac{\partial \mathcal{L}(\mathbf{W},\gamma)}{\partial \mathbf{W}} = 2(\mathbf{W}-\mathbf{I}_m)\mathbf{R}_s + 2\gamma\nu_{noise}^2\mathbf{W} = \mathbf{0} \tag{4.54}$$

which has the solution

$$\begin{aligned}\mathbf{W}_{TDC} &= \mathbf{R}_s(\mathbf{R}_s + \gamma\nu_{noise}^2\mathbf{I}_m)^{-1} \\ &= (\mathbf{R}_x - \nu_{noise}^2\mathbf{I}_m)(\mathbf{R}_x - \nu_{noise}^2(1-\gamma)\mathbf{I}_m)^{-1}\end{aligned} \tag{4.55}$$

Hence, the optimal filter is a Wiener filter with adjustable input noise level $\gamma\nu_{noise}^2$. Applying the eigendecompositions (3.17) and (3.18) to Equation (4.55), the optimal linear filter can be rewritten as

$$\begin{aligned}\mathbf{W}_{TDC} &= \mathbf{Q}_1\mathbf{\Lambda}_{s1}(\mathbf{\Lambda}_{s1} + \gamma\nu_{noise}^2\mathbf{I}_p)^{-1}\mathbf{Q}_1^T \\ &= \mathbf{Q}_1(\mathbf{I}_p - \nu_{noise}^2\mathbf{\Lambda}_{x1}^{-1})(\mathbf{I}_p - \nu_{noise}^2(1-\gamma)\mathbf{\Lambda}_{x1}^{-1})^{-1}\mathbf{Q}_1^T\end{aligned} \tag{4.56}$$

The TDC estimator of $\mathbf{s}$ is then given by $\hat{\mathbf{s}}_{TDC} = \mathbf{W}_{TDC}\mathbf{x}$, and the Lagrange multiplier constraint is obtained by substituting $\mathbf{W}_{TDC}$ in (4.52), i.e.,

$$\epsilon_n^2 = \nu_{noise}^2 \mathrm{tr}\left(\mathbf{W}_{TDC}\mathbf{W}_{TDC}^T\right) = \alpha m \nu_{noise}^2 \tag{4.57}$$

Thus, it is found that γ is given by

$$\alpha = \frac{1}{m}\mathrm{tr}\left(\mathbf{\Lambda}_{s1}^2(\mathbf{\Lambda}_{s1} + \gamma\nu_{noise}^2\mathbf{I}_p)^{-2}\right) \tag{4.58}$$

Equation (4.58) has a single solution since α is a monotonically decreasing continuous function of γ, and in a practical implementation, γ is actually used as the parameter. Note, that the estimator provides zero signal distortion when

$$\alpha = \alpha_{max} = \frac{p}{m} \Leftrightarrow \gamma = 0 \tag{4.59}$$

and that

$$\alpha = \alpha_{min} = 0 \Leftrightarrow \gamma = \infty \tag{4.60}$$

For speech signals, the TDC estimation criterion will control the nonstationary residual noise referred to as *musical noise* , since this noise component decreases as $\gamma \rightarrow \infty$.

4.1.6 Spectral Domain Constrained Estimator

The Spectral Domain Constrained (SDC) estimator [32] is a generalization of the TDC estimator, which minimizes the signal distortion energy ϵ_s^2 (4.51), while keeping the energy of the residual noise $\mathbf{r}_n$ (4.49) in each spectral component, defined by the eigenfilters $\mathbf{q}_i$ of the correlation matrix $\mathbf{R}_x \in \mathbb{R}^{m\times m}$, below some given threshold

$$\min_{\mathbf{W}} \epsilon_s^2 \quad \text{subject to} \quad \begin{cases} E\{|\mathbf{q}_i^T\mathbf{r}_n|^2\} \le \alpha_i \nu_{noise}^2\,, & i=1,\dots,p \\ E\{|\mathbf{q}_i^T\mathbf{r}_n|^2\} = 0\,, & i=p+1,\dots,m \end{cases} \tag{4.61}$$

where $0 \le \alpha_i \le 1$ controls the permissible *spectral* noise levels in the signal subspace, and ν_{noise}^2 is the noise power. For $i = (p+1,\dots,m)$, $\alpha_i = 0$ is chosen since the signal energy in the noise subspace is zero.

This strategy allows shaping of the spectrum of the residual noise, i.e., it can be masked by the speech signal. Thus, more noise is permitted to accompany high energy spectral components of the clean signal.

Again, given Assumption 3.2 on Page 26 is satisfied in the following, the optimal estimator in the sense of (4.61) can be found from the Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{W},\mathbf{\Gamma}) &= \epsilon_s^2 + \sum_{i=1}^{m} \gamma_i \nu_{noise}^2 \left(\mathbf{q}_i^T\mathbf{W}\mathbf{W}^T\mathbf{q}_i - \alpha_i\right) \qquad (4.62) \\ &= \operatorname{tr}\left((\mathbf{W}-\mathbf{I}_m)\mathbf{R}_s(\mathbf{W}-\mathbf{I}_m)^T\right) + \nu_{noise}^2 \operatorname{tr}\left(\mathbf{\Gamma}\mathbf{Q}^T\mathbf{W}\mathbf{W}^T\mathbf{Q}\right) - \nu_{noise}^2 \boldsymbol{\gamma}^T\boldsymbol{\alpha} \end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha_1,\cdots,\alpha_m)^T$, $\mathbf{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma})$ and $\boldsymbol{\gamma} = (\gamma_1,\cdots,\gamma_m)$. The latter is the vector of Lagrange multipliers. Thus, the filter matrix $\mathbf{W}$ is a stationary feasible point if it satisfies the gradient equation

$$\frac{\partial \mathcal{L}(\mathbf{W},\mathbf{\Gamma})}{\partial \mathbf{W}} = 2(\mathbf{W}-\mathbf{I}_m)\mathbf{R}_s + 2\nu_{noise}^2\mathbf{Q}\mathbf{\Gamma}\mathbf{Q}^T\mathbf{W} = \mathbf{0} \tag{4.63}$$

Applying the eigendecomposition (3.17) of $\mathbf{R}_s$ to Equation (4.63) gives

$$\begin{aligned} 2(\mathbf{W}-\mathbf{I}_m)\mathbf{Q}\mathbf{\Lambda}_s\mathbf{Q}^T + 2\nu_{noise}^2\mathbf{Q}\mathbf{\Gamma}\mathbf{Q}^T\mathbf{W} = \mathbf{0} \quad &\Rightarrow \\ (\mathbf{Q}^T\mathbf{W}\mathbf{Q}-\mathbf{I}_m)\mathbf{\Lambda}_s + \nu_{noise}^2\mathbf{\Gamma}\mathbf{Q}^T\mathbf{W}\mathbf{Q} = \mathbf{0} \quad &\Rightarrow \\ (\mathbf{G}-\mathbf{I}_m)\mathbf{\Lambda}_s + \nu_{noise}^2\mathbf{\Gamma}\mathbf{G} = \mathbf{0} \qquad & \qquad (4.64) \end{aligned}$$

where $\mathbf{G} = \mathbf{Q}^T\mathbf{W}\mathbf{Q}$. Hence, using the solution of the gradient matrix equation (4.64), the optimal filter is

$$\mathbf{W}_{SDC} = \mathbf{Q}\mathbf{G}\mathbf{Q}^T \tag{4.65}$$

and the SDC estimator of $\mathbf{s}$ is then given by $\hat{\mathbf{s}}_{SDC} = \mathbf{W}_{SDC}\mathbf{x}$. The Lagrange multiplier constraint is obtained by substituting $\mathbf{W}_{SDC}$ in (4.61), i.e.,

$$\begin{aligned} E\{|\mathbf{q}_i^T\mathbf{r}_n|^2\} &= \nu_{noise}^2\mathbf{q}_i^T\mathbf{W}_{SDC}\mathbf{W}_{SDC}^T\mathbf{q}_i \qquad (4.66) \\ &= \nu_{noise}^2\mathbf{e}_i^T\mathbf{G}\mathbf{G}^T\mathbf{e}_i \\ &= \alpha_i\nu_{noise}^2\,, \qquad i=1,\dots,m \end{aligned}$$

A *possible* solution to (4.64) and (4.66) is obtained when $\mathbf{G}$ is diagonal

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4.67}$$

$$\mathbf{G}_1 = \operatorname{diag}(\sqrt{\alpha_1},\dots,\sqrt{\alpha_p}) = \mathbf{\Lambda}_{s1}(\mathbf{\Lambda}_{s1} + \nu_{noise}^2\mathbf{\Gamma}_1)^{-1} \tag{4.68}$$

$$\mathbf{W}_{SDC} = \mathbf{Q}_1\mathbf{G}_1\mathbf{Q}_1^T \tag{4.69}$$

where the necessary conditions for the constrained minimization, i.e., $\gamma_i \geq 0$, are satisfied

$$\sqrt{\alpha_i} = \frac{\lambda_{s,i}}{\lambda_{s,i} + \nu^2_{noise}\gamma_i} \Leftrightarrow \tag{4.70}$$

$$\gamma_i = \frac{\lambda_{s,i}}{\nu^2_{noise}}\left(\frac{1}{\sqrt{\alpha_i}} - 1\right) \geq 0\,, \qquad \alpha_i \in]0;1] \text{ and } i = 1,\ldots,p \tag{4.71}$$

Thus, in the diagonal case, the choice of $\{\alpha_i\}_1^p$ completely specifies the matrix $\mathbf{G}_1$, and it can be chosen independently of the signals. This fact can be explained by the linear estimation method and the known spectra ν^2_{noise} and $\alpha_i\nu^2_{noise}$ of the input and output residual noise, respectively. However, $\{\alpha_i\}_1^p$ are normally chosen as functions of the statistics of the signal and noise, and the optimal filter in this sense has a structure that contains all the above methods as discussed in Section 4.1.8.

4.1.7 Empirical TDC and SDC Estimators

First assume that Assumption 3.3 on Page 35 is satisfied in the following, then the *empirical* version of the TDC method (ETDC) is obtained from the matrix formulation of Equation (4.49), i.e.,

$$\begin{aligned}\mathbf{R} &= \mathbf{X}\mathbf{W} - \mathbf{S} \\ &= \mathbf{S}(\mathbf{W} - \mathbf{I}_n) + \mathbf{N}\mathbf{W} \\ &= \mathbf{R}_S + \mathbf{R}_N\end{aligned} \tag{4.72}$$

where $\mathbf{R}$, $\mathbf{R}_S$ and $\mathbf{R}_N \in \mathbb{R}^{m\times n}$ are denoted the residual matrix, the signal distortion matrix and the residual noise matrix, respectively. The squared signal distortion error e_s^2 and residual noise error e_n^2 can then be defined as

$$e_s^2 = \operatorname{tr}\left(\mathbf{R}_S^T\mathbf{R}_S\right) = \operatorname{tr}\left((\mathbf{W} - \mathbf{I}_n)^T\mathbf{S}^T\mathbf{S}(\mathbf{W} - \mathbf{I}_n)\right) \tag{4.73}$$

$$e_n^2 = \operatorname{tr}\left(\mathbf{R}_N^T\mathbf{R}_N\right) = \sigma^2_{noise}\operatorname{tr}\left(\mathbf{W}^T\mathbf{W}\right) \tag{4.74}$$

Using these equations in the constrained minimization

$$\min_{\mathbf{W}} e_s^2 \quad \text{subject to} \quad e_n^2 \leq \alpha n \sigma^2_{noise} \tag{4.75}$$

gives the following Lagrangian

$$\mathcal{L}(\mathbf{W},\gamma) = \operatorname{tr}\left((\mathbf{W} - \mathbf{I}_n)^T\mathbf{S}^T\mathbf{S}(\mathbf{W} - \mathbf{I}_n)\right) + \gamma\left(\sigma^2_{noise}\operatorname{tr}(\mathbf{W}^T\mathbf{W}) - \alpha n\sigma^2_{noise}\right) \tag{4.76}$$

and the corresponding gradient equation

$$\frac{\partial\mathcal{L}(\mathbf{W},\gamma)}{\partial\mathbf{W}} = 2\mathbf{S}^T\mathbf{S}(\mathbf{W} - \mathbf{I}_n) + 2\gamma\sigma^2_{noise}\mathbf{W} = \mathbf{0} \tag{4.77}$$

which has the solution

$$\mathbf{W}_{ETDC} = (\mathbf{S}^T\mathbf{S} + \gamma\sigma^2_{noise}\mathbf{I}_n)^{-1}\mathbf{S}^T\mathbf{S} \tag{4.78}$$

Applying the SVD (3.55) and (3.57) to (4.78), the optimal filter can be rewritten as

$$\begin{aligned}\mathbf{W}_{ETDC} &= \mathbf{V}_{S1}(\mathbf{\Sigma}^2_{S1} + \gamma\sigma^2_{noise}\mathbf{I}_p)^{-1}\mathbf{\Sigma}^2_{S1}\mathbf{V}^T_{S1} \\ &= \mathbf{V}_{X1}(\mathbf{I}_p - \sigma^2_{noise}(1-\gamma)\mathbf{\Sigma}^{-2}_{X1})^{-1}(\mathbf{I}_p - \sigma^2_{noise}\mathbf{\Sigma}^{-2}_{X1})\mathbf{V}^T_{X1}\end{aligned} \tag{4.79}$$

Similarly, the *empirical* version of the SDC method (ESDC) can be obtained by using the following equation based on the right singular vectors $\mathbf{v}_{X,i}$ of the data matrix $\mathbf{X}$

$$\|\mathbf{R}_N\mathbf{v}_{X,i}\|_2^2 = \mathbf{v}_{X,i}^T\mathbf{R}_N^T\mathbf{R}_N\mathbf{v}_{X,i} = \sigma_{noise}^2\mathbf{v}_{X,i}^T\mathbf{W}^T\mathbf{W}\mathbf{v}_{X,i} \tag{4.80}$$

together with Equation (4.73) in the constrained minimization

$$\min_{\mathbf{W}} e_s^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{R}_N\mathbf{v}_{X,i}\|_2^2 \le \alpha_i\sigma_{noise}^2\,, & i = 1,\dots,p \\ \|\mathbf{R}_N\mathbf{v}_{X,i}\|_2^2 = 0\,, & i = p+1,\dots,n \end{cases} \tag{4.81}$$

This gives the Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{W},\mathbf{\Gamma}) &= e_s^2 + \sum_{i=1}^{n}\gamma_i\sigma_{noise}^2\left(\mathbf{v}_{X,i}^T\mathbf{W}^T\mathbf{W}\mathbf{v}_{X,i} - \alpha_i\right) \\ &= \mathrm{tr}\left((\mathbf{W}-\mathbf{I}_n)^T\mathbf{S}^T\mathbf{S}(\mathbf{W}-\mathbf{I}_n)\right) + \sigma_{noise}^2\mathrm{tr}\left(\mathbf{\Gamma}\mathbf{V}_X^T\mathbf{W}^T\mathbf{W}\mathbf{V}_X\right) - \sigma_{noise}^2\boldsymbol{\gamma}^T\boldsymbol{\alpha} \end{aligned} \tag{4.82}$$

defined along the lines in (4.62), and the corresponding gradient equation

$$\frac{\partial\mathcal{L}(\mathbf{W},\mathbf{\Gamma})}{\partial\mathbf{W}} = 2\mathbf{S}^T\mathbf{S}(\mathbf{W}-\mathbf{I}_n) + 2\sigma_{noise}^2\mathbf{W}\mathbf{V}_X\mathbf{\Gamma}\mathbf{V}_X^T = \mathbf{0} \tag{4.83}$$

Applying the SVD of $\mathbf{S}$ (3.55) to Equation (4.83) and using the fact that $\mathbf{V}_X = \mathbf{V}_S$, c.f., Equation (3.57), results in

$$\begin{aligned} \mathbf{V}_X\mathbf{\Sigma}_S^2\mathbf{V}_X^T(\mathbf{W}-\mathbf{I}_n) + \sigma_{noise}^2\mathbf{W}\mathbf{V}_X\mathbf{\Gamma}\mathbf{V}_X^T = \mathbf{0} \quad &\Rightarrow \\ \mathbf{\Sigma}_S^2(\mathbf{V}_X^T\mathbf{W}\mathbf{V}_X - \mathbf{I}_n) + \sigma_{noise}^2\mathbf{V}_X^T\mathbf{W}\mathbf{V}_X\mathbf{\Gamma} = \mathbf{0} \quad &\Rightarrow \\ \mathbf{\Sigma}_S^2(\mathbf{G}-\mathbf{I}_n) + \sigma_{noise}^2\mathbf{G}\mathbf{\Gamma} = \mathbf{0} & \end{aligned} \tag{4.84}$$

where the optimal filter solution is determined by $\mathbf{G} \in \mathbb{R}^{n\times n}$

$$\mathbf{W}_{ESDC} = \mathbf{V}_X\mathbf{G}\mathbf{V}_X^T \tag{4.85}$$

The Lagrange multiplier constraint is equivalent to (4.66), so a possible solution for $\mathbf{G}$ is given by (4.67) with

$$\mathbf{G}_1 = \mathrm{diag}(\sqrt{\alpha_1},\dots,\sqrt{\alpha_p}) = \mathbf{\Sigma}_{S1}^2(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{\Gamma}_1)^{-1} \tag{4.86}$$

Obviously, under stationary and ergodic conditions, the empirical versions of the TDC and SDC estimators will converge to true ones as $m \to \infty$.

4.1.8 A Unified Notation

All the discussed linear signal subspace based estimates are formulated by means of two orthogonal transformations and a diagonal matrix containing weights, so it is appropriate to introduce a unified notation.

Let the eigendecomposition of the correlation matrix $\mathbf{R}_x$ be defined by $\mathbf{Q}\mathbf{\Lambda}_x\mathbf{Q}^T$, then the eigenvector based transformation $\mathbf{y} = \mathbf{Q}^T\mathbf{x}$ is called a *Karhunen-Loeve* transform (KLT) (see, e.g., [56, Page 144]) and it simply diagonalizes the correlation matrix $\mathbf{R}_y = \mathbf{Q}^T\mathbf{R}_x\mathbf{Q} = \mathbf{\Lambda}_x$, i.e., the components of $\mathbf{y}$ are independent. In the empirical case, the KLT is estimated by $\mathbf{y} = \mathbf{V}^T\mathbf{x}$, where $\mathbf{V}$ is the right singular vectors, and it operates on the data matrix $\mathbf{X}$ or the sample correlation matrix $\hat{\mathbf{R}}_x$ (3.36). Hence, all the linear signal estimates are obtained by the following steps (see Figure 4.1):

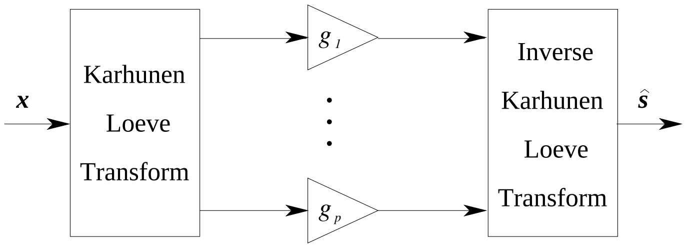


Figure 4.1 General model for signal subspace based linear signal estimator.

- KLT of the noisy signal onto the signal subspace.
- Modify the independent components of the KLT by a diagonal *gain function* $\mathbf{G}_1$.
- Inverse KLT of the modified components to reconstruct the signal in the signal subspace.

This scheme results in a generalized formulation of the optimal linear signal estimators

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} = \mathbf{Q}\mathbf{G}\mathbf{Q}^T\mathbf{x} = \mathbf{Q}_1\mathbf{G}_1\mathbf{Q}_1^T\mathbf{x}\,, \qquad \mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4.87}$$

or the empirical version based on the data matrix

$$\hat{\mathbf{S}} = \mathbf{X}\mathbf{W} = \mathbf{X}\mathbf{V}_X\mathbf{G}\mathbf{V}_X^T = \mathbf{X}\mathbf{V}_{X1}\mathbf{G}_1\mathbf{V}_{X1}^T = \mathbf{U}_{X1}\mathbf{\Sigma}_{X1}\mathbf{G}_1\mathbf{V}_{X1}^T \tag{4.88}$$

where the gain matrix $\mathbf{G}_1 = \mathrm{diag}(g_1 \ldots g_p) \in \mathbb{R}^{p\times p}$ depends on the estimation method (see Table 4.1).

Method	Gain matrix $\mathbf{G}_1$	
LS	$\mathbf{I}_p$	$\mathbf{I}_p$
LMMSE	$\mathbf{\Lambda}_{s1}(\mathbf{\Lambda}_{s1} + \nu_{noise}^2\mathbf{I}_p)^{-1}$	$(\mathbf{I}_p - \nu_{noise}^2\mathbf{\Lambda}_{x1}^{-1})$
MV	$\mathbf{\Sigma}_{S1}^2(\mathbf{\Sigma}_{S1}^2 + \sigma_{noise}^2\mathbf{I}_p)^{-1}$	$(\mathbf{I}_p - \sigma_{noise}^2\mathbf{\Sigma}_{X1}^{-2})$
TDC	$\mathbf{\Lambda}_{s1}(\mathbf{\Lambda}_{s1} + \gamma\nu_{noise}^2\mathbf{I}_p)^{-1}$	$(\mathbf{I}_p - \nu_{noise}^2\mathbf{\Lambda}_{x1}^{-1})(\mathbf{I}_p - \nu_{noise}^2(1-\gamma)\mathbf{\Lambda}_{x1}^{-1})^{-1}$
ETDC	$\mathbf{\Sigma}_{S1}^2(\mathbf{\Sigma}_{S1}^2 + \gamma\sigma_{noise}^2\mathbf{I}_p)^{-1}$	$(\mathbf{I}_p - \sigma_{noise}^2\mathbf{\Sigma}_{X1}^{-2})(\mathbf{I}_p - \sigma_{noise}^2(1-\gamma)\mathbf{\Sigma}_{X1}^{-2})^{-1}$
SDC/ESDC	$\mathrm{diag}(\sqrt{\alpha_1}, \ldots, \sqrt{\alpha_p})$	$\mathrm{diag}(\sqrt{\alpha_1}, \ldots, \sqrt{\alpha_p})$

Table 4.1 Diagonal gain matrix for different estimation methods formulated in terms of the clean signal (first column) or the noisy signal (second column).

Obviously, the relations between the TDC estimator and the LMMSE and LS estimators are

$$\mathbf{G}_{LS} = \mathbf{G}_{TDC}|_{\gamma=0} \tag{4.89}$$
$$\mathbf{G}_{LMMSE} = \mathbf{G}_{TDC}|_{\gamma=1} \tag{4.90}$$

Note, that the LS estimator only projects the noisy signal onto the signal subspace, i.e., this estimator results in the lowest possible (zero) signal distortion and in the highest possible residual noise level $(p/m)\nu_{noise}^2$, and that $\mathbf{G}_{LMMSE}$ is denoted the *Wiener gain function*.

In implementing the TDC estimator, the permissible noise level coefficient α should be signal dependent with value chosen according to the masking threshold of the auditory system, i.e., the residual noise becomes inaudible. Unfortunately, the masking threshold in the time domain is not fully understood. An alternative method is to specify γ, since increasing γ from zero to infinity, the level of the residual noise decreases while the level of signal distortion increases. A useful approach for choosing γ is to make it dependent on the SNR in each frame of the noisy signal [31]

$$\gamma = 1 + \frac{1}{1 + \mathrm{SNR}} \tag{4.91}$$

where SNR, e.g., could be given by (3.61). In this case, $1 \leq \gamma \leq 2$, and the TDC estimator acts similarly to the LMMSE estimator at high SNR, while at low SNR it further reduces the residual noise. A simpler approach is to set γ to a fixed value which is greater than 1, e.g., in the neighborhood of 2.

For the SDC estimator, two signal dependent choices for α_i are proposed in [32], which makes the spectrum of the residual noise look similar to that of the clean signal

$$1: \qquad \alpha_i = \left(\frac{\lambda_{s,i}}{\lambda_{s,i} + \nu_{noise}^2}\right)^{\beta_1} = \left(\frac{\lambda_{x,i} - \nu_{noise}^2}{\lambda_{x,i}}\right)^{\beta_1}, \qquad \beta_1 \geq 1 \tag{4.92}$$

$$2: \qquad \alpha_i = \exp\left(\frac{-\beta_2 \nu_{noise}^2}{\lambda_{s,i}}\right) = \exp\left(\frac{-\beta_2 \nu_{noise}^2}{\lambda_{x,i} - \nu_{noise}^2}\right), \qquad \beta_2 \geq 1 \tag{4.93}$$

where β is an experimentally determined constant controlling the suppression level of the noise as well as the resulting signal distortion, i.e., when β increases, the residual noise level decreases and the signal distortion level increases. As before, the following relations are obtained

$$\mathbf{G}_{LS} = \mathbf{G}_{SDC}|_{\alpha_i=1} \tag{4.94}$$

$$\mathbf{G}_{LMMSE} = \mathbf{G}_{SDC}|_{\beta_1=2} \tag{4.95}$$

Equation (4.93) is denoted the *generalized* Wiener gain function, because the first-order Taylor approximation of $\alpha_i^{-1/2}$ is the inverse of the Wiener gain function

$$\alpha_i^{-1/\beta_2} = \exp\left(\frac{\nu_{noise}^2}{\lambda_{s,i}}\right) \approx 1 + \frac{\nu_{noise}^2}{\lambda_{s,i}} = \frac{\lambda_{s,i} + \nu_{noise}^2}{\lambda_{s,i}} \tag{4.96}$$

The generalized Wiener gain function with $\beta_2 = 5$ is referred [32] particularly useful in speech enhancement.

Plot of the gain g_i as function of the spectral SNR, i.e., $\lambda_{s,i}/\nu_{noise}^2$, can be used to characterize the different estimation methods. In Figure 4.2(a) and 4.2(b), the Wiener gain function is compared with the TDC and SDC functions for different parameter choices. Note, that the Wiener function is symmetric around the point (0dB,$\frac{1}{2}$), and that the SDC approach results in a more aggressive transition between high and low gain levels, while the TDC method simply moves the Wiener transition to higher SNR due to the relation

$$g_{TDC,i}\left(\frac{\lambda_{s,i}}{\nu_{noise}^2}, \gamma\right) = g_{LMMSE,i}\left(\frac{\gamma \lambda_{s,i}}{\nu_{noise}^2}\right) \tag{4.97}$$

All the above gain functions have been obtained by introducing a signal subspace, i.e., the gains related to the noise subspace are explicitly set to zero. However, the Wiener based gain functions can do this implicitly, since spectral SNRs related to the noise subspace are assumed to be zero (in practice small). The difference between the two approaches become obvious, when the gains are estimated from empirical data, since small gains will be dominated by estimation errors. Thus, it is important to estimate the signal subspace dimension in advance to avoid flaws of the estimator. In the following, negative gain estimates are inverted (full-wave rectification).

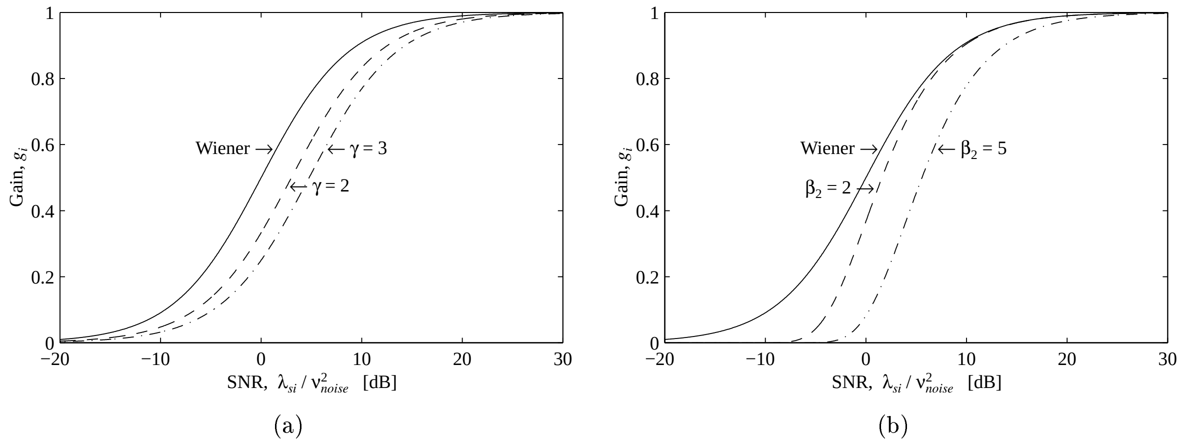


Figure 4.2 (a) Wiener and TDC gain functions for different choices of γ. (b) Wiener and generalized Wiener gain functions for different choices of β_2.

4.1.9 Analysis of Practical Gain Functions

In order to analyze the practical behaviour of the different gain functions, the speech and noise signals are used separately for the evaluation of the filter parameters.

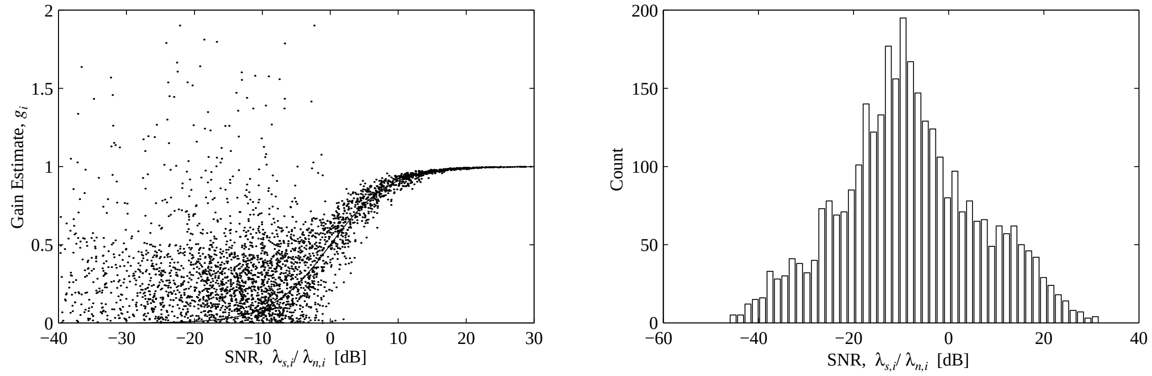
Figure 4.3(a) shows estimated Wiener gains $\{g_i\}_1^{20}$ of 165 speech frames ($\mathbf{X} \in \mathbb{R}^{141\times 20}$) obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample (white noise and SNR $= 10$dB). The estimated gains are calculated from (4.46) using the singular values of $\mathbf{X}$, and a σ^2_{noise} obtained from an initial noise matrix $\mathbf{N}$. Each gain g_i is related to a KLT vector, i.e., a right singular vector $\mathbf{v}_{X,i}$ of $\mathbf{X}$, which is used to plot g_i as function of the spectral SNR in that direction

$$\mathrm{SNR}_{spectral} = \frac{\|\mathbf{S}\mathbf{v}_{X,i}\|_2^2}{\|\mathbf{N}\mathbf{v}_{X,i}\|_2^2} \approx \frac{\lambda_{s,i}}{\lambda_{n,i}} \tag{4.98}$$

For spectral SNR less than about 0 dB, a large variance in the estimated gains is observed, which illustrates the importance of explicitly introducing a signal subspace. The estimation errors can be explained by the difference in the estimated eigenvalues of the noise given by $\sigma^2_{noise}\mathbf{I}$ and the true ones $\mathbf{\Sigma}^2_N$. From Figure 4.3(b), it can be seen that with this global noise level, the distribution of the spectral SNRs basicly covers the transition interval from -20 to 20 dB with the mean value around the lower part.

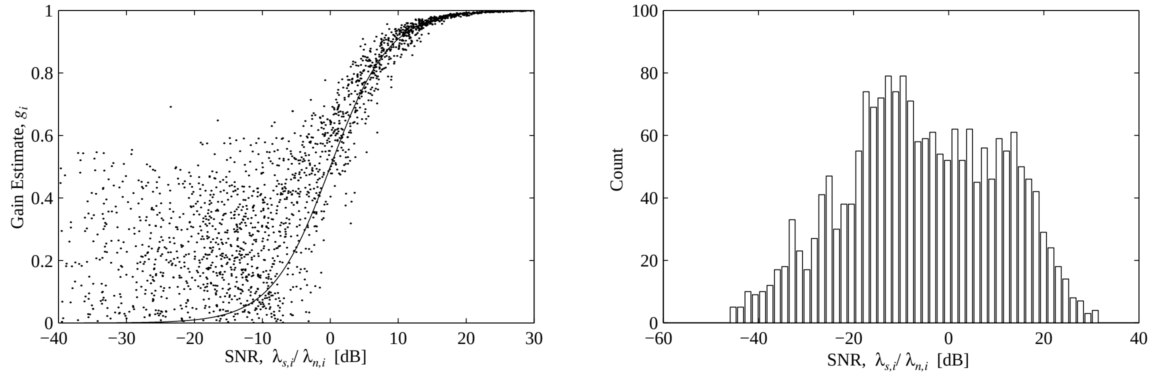
If only the gains in a 12-dimensional signal subspace are considered, the distribution still covers the same range (see Figure 4.4(b)), but the mean value is shifted to higher SNR. Thus, even in this case, a large number of badly estimated gains at low SNRs can be expected as shown in Figure 4.4(a).

The conclusion is that the TDC and SDC based gain functions can be expected to perform better, since they are less sensitive to errors in the noise estimation (for increasing parameter values). This is illustrated in Figure 4.5, where estimated gains are plotted for the TDC estimator ($\gamma = 3$) and the SDC estimator ($\beta_2 = 5$).



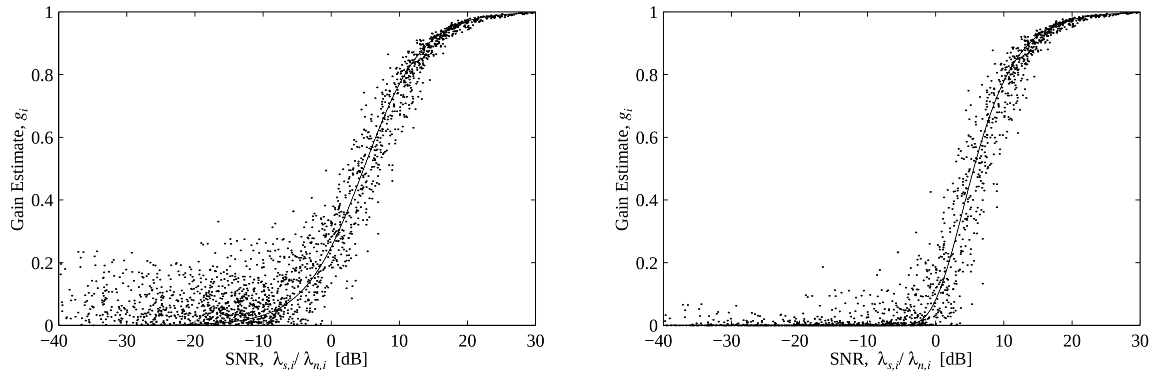
(a) (b)

Figure 4.3 (a) Estimated Wiener gains $\{g_i\}_1^{20}$ of 165 speech frames ($\mathbf{X} \in \mathbb{R}^{141\times 20}$) obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample (white noise and SNR=10dB). The estimated gains are plotted as function of the spectral SNR $\lambda_{s,i}/\lambda_{n,i}$. (b) Distribution of the gains.



(a) (b)

Figure 4.4 As Figure 4.3, but with gains $\{g_i\}_1^{12}$ belonging to a 12-dimensional signal subspace.



(a) (b)

Figure 4.5 As Figure 4.4, but with gains given by (a) the TDC estimator ($\gamma = 3$) and (b) the SDC estimator ($\beta_2 = 5$).

Now, consider the voiced speech frame of 160 samples added white noise (SNR = 5 dB), and organized in the data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$. The linear estimators as defined by Equation (4.88) with $p = 12$ are then characterized by the residual matrices $\mathbf{R} = \mathbf{R}_S + \mathbf{R}_N$, c.f., Equation (4.72), or the corresponding residual signals obtained by averaging along the diagonals (see Section 4.6). The latter can be used to compare the different estimators as illustrated in Figure 4.6, where the power of the residual signals are plotted as function of the TDC parameter γ and the SDC parameter β_2.

Clearly, for increasing parameter values, the level of the residual noise $\mathbf{r}_n$ decreases while the level of signal distortion $\mathbf{r}_s$ increases. The minimum residual power is obtained for the MV estimator ($\gamma = 1$ or approximated by $\beta_2 \approx 2$) and is dominated by the residual noise, however, by choosing the perceptually more meaningful parameters $\gamma = 3$ or $\beta_2 = 5$ as discussed in a moment, the signal distortion will become dominant for the price of a slightly increase in the level of the total residual signal.

The 10th order LPC-based magnitude spectra of the residual noise and signal distortion corresponding to the examples in Figure 4.6 are shown in Figure 4.7 and 4.8. For both the TDC and the SDC method, the spectrum of the residual noise looks like the one for the clean speech (see Figure 2.7(a)), and is referred to as musical noise (see Section 4.4). For increasing parameter values, less noise accompany the low energy spectral components in accordance to the masking threshold of the auditory system. However, then the spectra of the signal distortion moves towards the one for the clean speech.

Notice, that in the LS case ($\gamma = 0$), the residual noise actually has the highest level at the least dominant formant, i.e., the musical noise will be most audible for this estimator, and that the signal distortion corresponds to the part of clean signal laying in the noise subspace.

The examples demonstrate the importance of having other quality measures than SNR, taking into account the spectral density of different signal components.

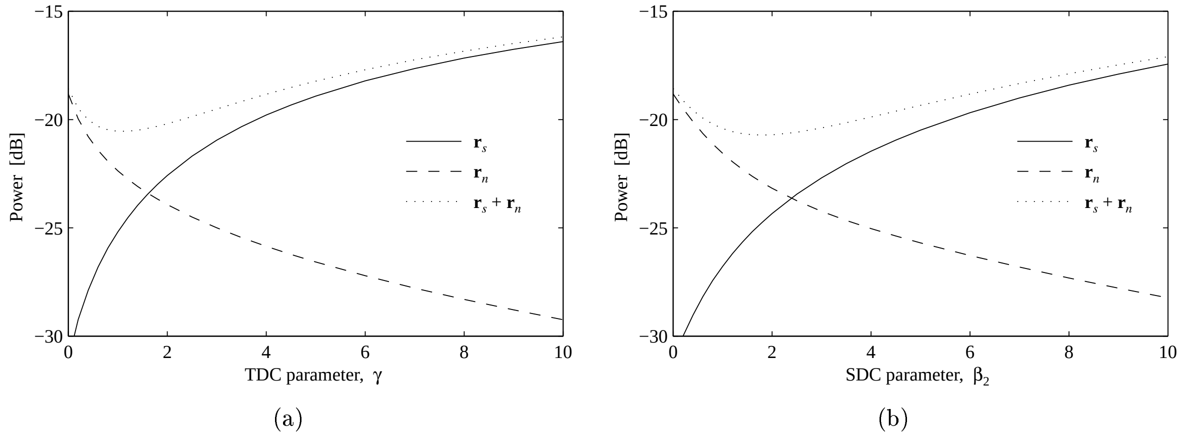
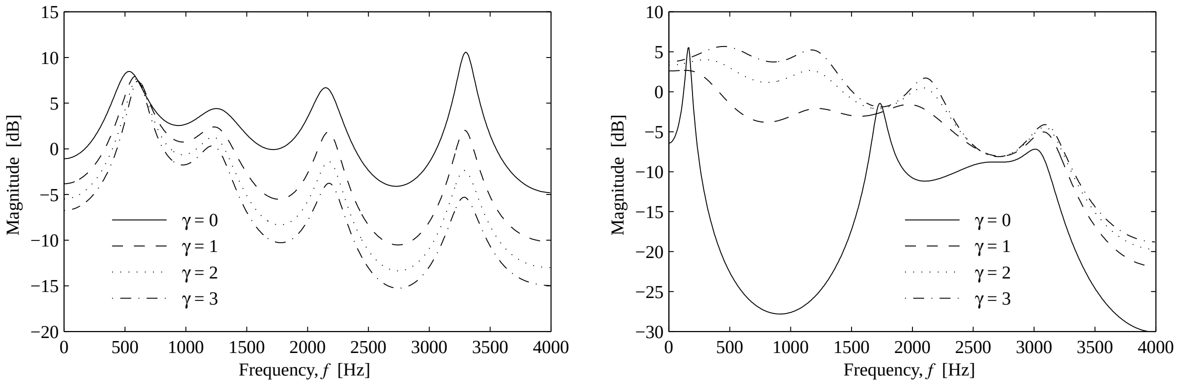
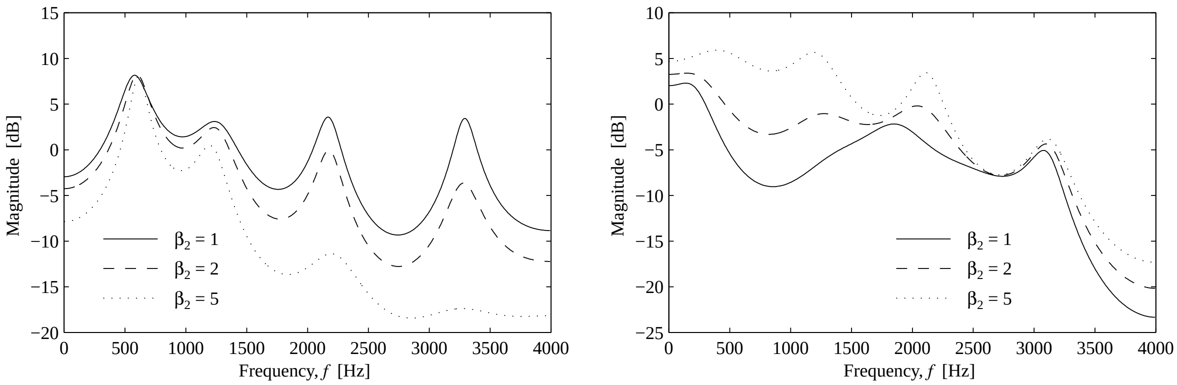


Figure 4.6 Power of the residual noise $\mathbf{r}_n$ and the signal distortion $\mathbf{r}_s$ for a linear estimator obtained from a 12-dimensional signal subspace. The data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ represents the voiced speech frame of 160 samples added white noise (SNR=5dB). The residual levels are plotted against the TDC parameter γ (a) and the SDC parameter β_2 (b).



(a) (b)

Figure 4.7 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.6(a).



(a) (b)

Figure 4.8 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.6(b).

4.2 Linear Signal Estimators by QSVD

In the colored noise case, all the SVD based linear signal estimators in Section 4.1 can easily be obtained from the QSVD of $(\mathbf{X}, \mathbf{N})$.

First, the desired rank-p estimate of the noise normalized signal $\mathbf{SN}^+$ is constructed from the normalized data matrix $\mathbf{XN}^+$ (3.103) along the lines in Sections 4.1.2, 4.1.4 and 4.1.7, i.e.,

$$\widehat{\mathbf{SN}^+} = \mathbf{U}_{X1}\mathbf{\Delta}_1\mathbf{G}_1\mathbf{U}_{N1}^T \tag{4.99}$$

where the gain matrix $\mathbf{G}_1 = \operatorname{diag}(g_1 \ldots g_p) \in \mathbb{R}^{p\times p}$ depends on the estimation method (see Table 4.2).

Thus, the weights g_i are computed by the same formulas as in the SVD based algorithms, but with the singular values σ_i replaced by the quotient singular values δ_i and by using the fact that the noise variance of the prewhitened signal is one.

Method	Gain matrix $\mathbf{G}_1$
LS	$\mathbf{I}_p$
MV	$(\mathbf{I}_p - \mathbf{\Delta}_1^{-2})$
ETDC	$(\mathbf{I}_p - (1-\gamma)\mathbf{\Delta}_1^{-2})(\mathbf{I}_p - \mathbf{\Delta}_1^{-2})$
ESDC	$\mathrm{diag}(\sqrt{\alpha_1}, \ldots, \sqrt{\alpha_p})$

Table 4.2 Diagonal gain matrix for different estimation methods as defined in Sections 4.1.2, 4.1.4 and 4.1.7.

To obtain the corresponding rank-p estimate of the signal $\mathbf{S}$, the estimate of $\mathbf{SN}^+$ must be denormalized by the noise-only matrix $\mathbf{N}$

$$\begin{aligned}\hat{\mathbf{S}} &= (\widehat{\mathbf{SN}^+})\mathbf{N} \\ &= \mathbf{U}_{X1}\mathbf{C}_1\mathbf{G}_1\mathbf{\Theta}_1^T\end{aligned} \tag{4.100}$$

which can be computed directly from (3.102), i.e., the prewhitening is now an integral part of the algorithm. By using the identity (3.92)

$$\begin{pmatrix} \mathbf{\Theta}_1^T\mathbf{Z}_1 & \mathbf{\Theta}_1^T\mathbf{Z}_2 \\ \mathbf{\Theta}_2^T\mathbf{Z}_1 & \mathbf{\Theta}_2^T\mathbf{Z}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-p} \end{pmatrix} \tag{4.101}$$

the matrix $\mathbf{U}_{X1}$ is given by

$$\mathbf{U}_{X1} = \mathbf{XZ}_1\mathbf{C}_1^{-1} \tag{4.102}$$

i.e., Equation (4.100) can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}} = \mathbf{XZ}_1\mathbf{G}_1\mathbf{\Theta}_1^T \tag{4.103}$$

The QSVD based LS/MV algorithms have been used succesfully for reduction of colored broad-banded noise in speech signals by Jensen [62].

4.2.1 Analysis of QSVD-Based Gain Functions

In this Section, the practical behaviour of the QSVD based estimators are analyzed and compared with the SVD examples in Section 4.1.9. Figure 4.9 and 4.10 shows estimated gains corresponding to the examples in Figure 4.4 and 4.5. However, the additive noise is a colored AR(1,-0.7) process, and each gain g_i is now related to a quotient singular vector $\mathbf{u}_{N,i}$ of $(\mathbf{X}, \mathbf{N})$, which is used to plot g_i as function of the spectral SNR in that direction

$$\mathrm{SNR}_{spectral} = \frac{\|\mathbf{S}\hat{\mathbf{N}}^+\mathbf{u}_{N,i}\|_2^2}{\|\mathbf{N}\hat{\mathbf{N}}^+\mathbf{u}_{N,i}\|_2^2} \tag{4.104}$$

where the noise matrix $\hat{\mathbf{N}}$ used in the prewhitening (normalization) is obtained from an initial noise segment. Compared to the white noise case, an increased variance in the estimated gains is observed due to the more complex noise dependency.

Now, consider the residual signals of the voiced speech frame, where the examples shown in Figure 4.11 – 4.13 are colored noise versions of the one in Figure 4.6 – 4.8. In Figure 4.11, the power of the residual signals are plotted as function of the TDC parameter γ and the SDC parameter β_2. As in the white noise case, increasing parameter values reduce the level of the residual noise $\mathbf{r}_n$ while the level of signal distortion $\mathbf{r}_s$ increases. Note that the minimum residual power has increased compared to Figure 4.6, and that it is no longer obtained for the MV estimator. The reason is the bias of the signal subspace due to the prewhitening, i.e., *the signal distortion has increased significantly.*

This can also be seen from the 10th order LPC-based magnitude spectra of the residual noise and signal distortion corresponding to the examples in Figure 4.11 as shown in Figure 4.12 and 4.13. Thus, prewhitening with a noise process dominated by low-frequency energy results in both signal distortion and residual noise, which also have high levels at low frequencies. Since, a majority of speech frames have this kind of spectral distribution, the effect of prewhitening will for this noise process be a more noticeably speech distortion and a musical noise characterized by low frequencies.

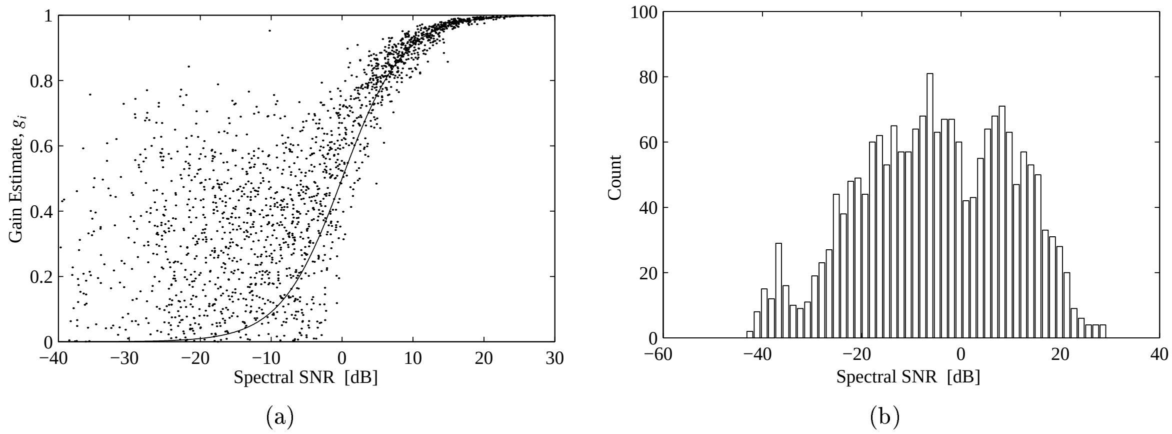


Figure 4.9 (a) Wiener gains $\{g_i\}_1^{12}$ estimated from a 12-dimensional signal subspace and obtained from 165 speech frames ($\mathbf{X} \in \mathbb{R}^{141\times 20}$), i.e., from the noisy reference sentence by shifting a 160 sample window by 160 sample (colored AR(1,-0.7) noise and SNR=10dB). The estimated gains are plotted as function of spectral SNR. (b) Distribution of the gains.

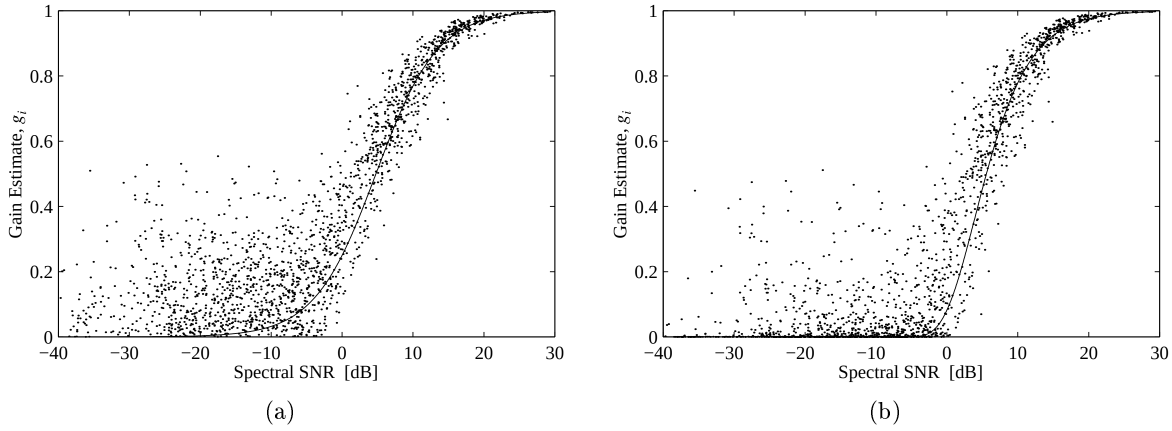
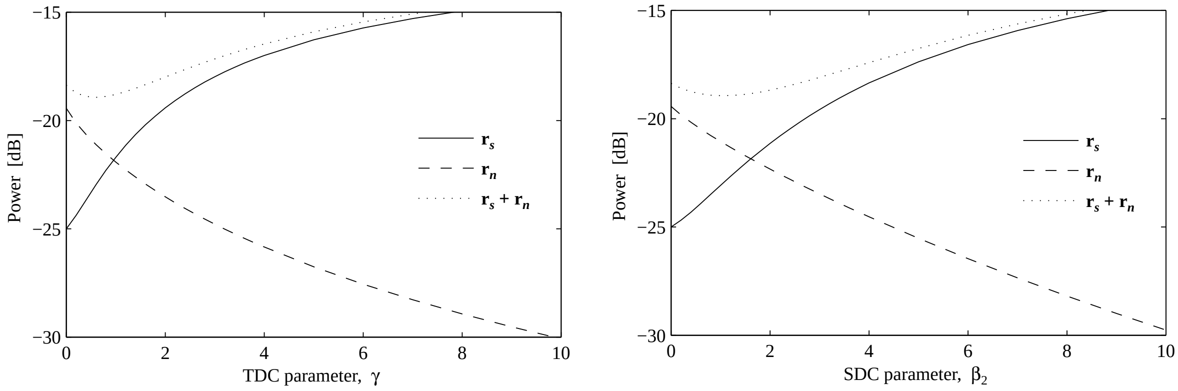
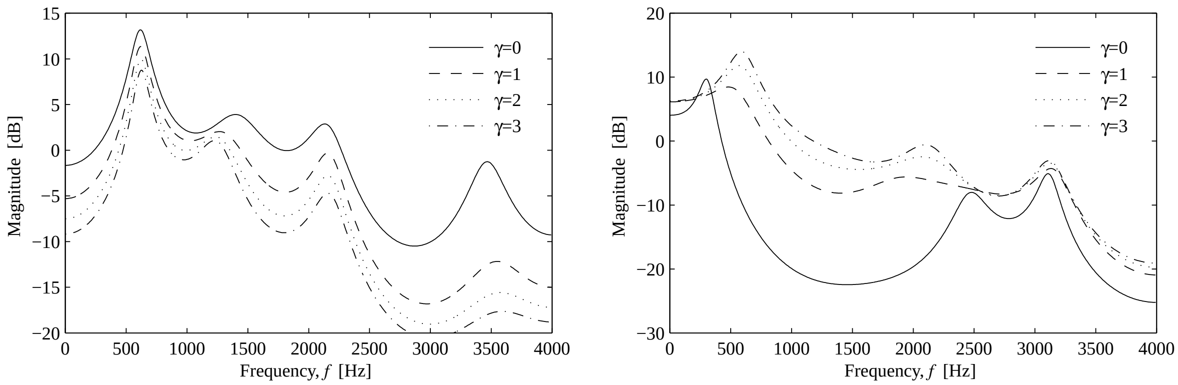


Figure 4.10 As Figure 4.9, but with gains given by (a) the TDC estimator ($\gamma = 3$) and (b) the SDC estimator ($\beta_2 = 5$).



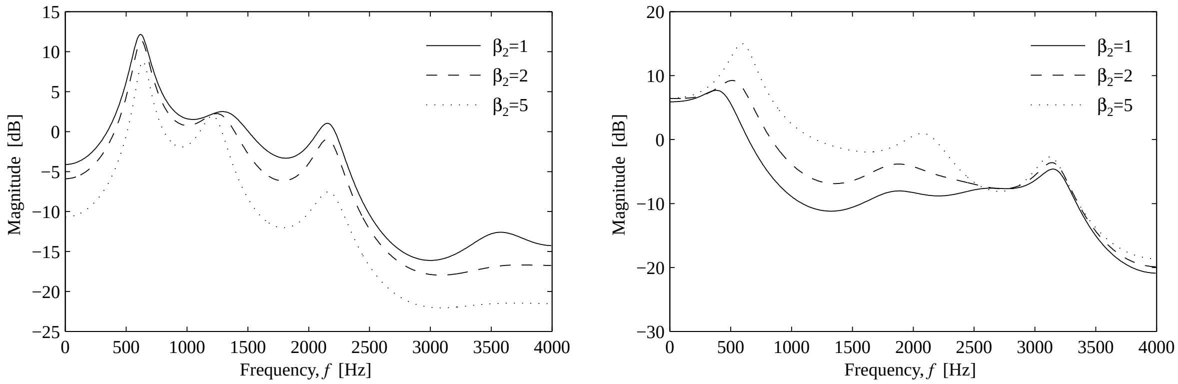
(a) (b)

Figure 4.11 Power of the residual noise $\mathbf{r}_n$ and signal distortion $\mathbf{r}_s$ for a linear estimator obtained from a 12-dimensional signal subspace. The data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ represents the voiced speech frame of 160 samples added colored noise (SNR=5dB). The residual levels are plotted against the TDC parameter γ (a) and the SDC parameter β_2 (b).



(a) (b)

Figure 4.12 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.11(a).



(a) (b)

Figure 4.13 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.11(b).

4.3 Relation to Spectral Subtraction

In the spectral subtraction approach, the DFT rather than the KLT is used (see Section 2.5.3), i.e., the second-order statistics of the signal and noise affect only the gain function of the estimator while the transform, i.e., the DFT, is signal independent.

Based on the fact that the KLT and the DFT are related[1] [41], the spectral subtraction method is by Ephraim and Van Trees [32] shown to be an approximate signal subspace approach, which is optimal in a sense determined by the specific gain function. Indeed, it is proven that if the Wiener gain function is used, then the spectral subtraction is optimal in an asymptotic LMMSE sense when the vector length m goes to infinity and the signal and noise are assumed stationary, i.e.,

$$\lim_{m\to\infty} \frac{1}{m} E\{\|\hat{\mathbf{s}}_{LMMSE} - \hat{\mathbf{s}}_{SPS}\|_2^2\} = 0 \tag{4.105}$$

Another issue is the existence of the signal subspace, i.e., part of the spectrum represents only noise components. Thus, the spectral subtraction gains are often found to be negative due to the nonperfect estimator. Normally, negative gains are set to zero (half-wave rectification), which is *equivalent* to nulling spectral components in the noise subspace.

Figure 4.14(a) shows estimated Wiener gains (half-wave rectification) of 165 speech frames obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample (white noise and SNR $= 10$ dB). The estimated gains are calculated from (2.26) using the periodograms $\hat{\Gamma}_x(m) = |\mathbf{D}^H\mathbf{x}|_m^2/K$ obtained from the Hanning windowed frames, and a $\hat{\Gamma}_n(m) = \nu_{noise}^2$ obtained from an initial noise frame. Each gain g_i is related to the spectral SNR given by $\hat{\Gamma}_s(m)/\hat{\Gamma}_n(m)$. Clearly, the variance in the estimated gains are larger than the one obtained by the subspace methods (see Figure 4.4). In Figure 4.14(b), the estimated Wiener gains are calculated after averaging four periodograms, and a result much closer to the subspace method is obtained. This is also what could be expected since the MV estimate converges to the LMMSE estimate when the frame length goes to infinity, while convergence of the spectral subtraction based estimator is only achieved when both the frame length and the number of periodograms goes to infinity.

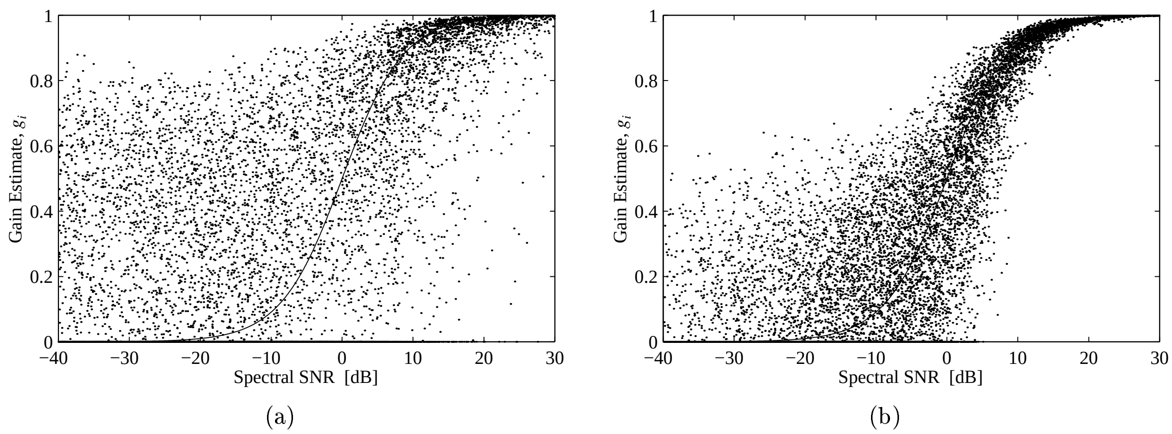


Figure 4.14 (a) Estimated Wiener gains (half-wave rectification) of 165 speech frames obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample (white noise and SNR=10dB). The estimated gains are plotted as function of the spectral SNR. (b) As (a) after averaging four adjacent periodograms (frames).

[1] The two transforms coincide if the covariance matrix of the signal is circulant.

4.4 Musical Noise

In spectral subtraction techniques, the musical noise introduced in the enhanced speech is often thought of as an estimation problem, i.e., negative spectral magnitude values are set to zero (half-wave rectification). However, this is not really true as discussed in the following, where arguments indicate that *the musical noise is linked to the speech dependent filtering.*

First, assume that the desired gain function can be obtained *perfectly*, then spectral components with low SNR are set to zero, while components with high SNR remain untouched. The latter will typically be the formants, i.e., the residual noise spectrum will also have formants similar to the one for the clean speech segment (see, e.g., Figure 4.7(a)). For a given noise level and gain function, one or more noise formants may dominate the corresponding speech formant(s), i.e., the spectral SNR is less than 0 dB. Thus, the tonal characteristics of the residual noise will be audible. This is, e.g., the case for the Wiener function, where the transition between high and low gains are centered around 0 dB, so components with spectral SNR less than 0 dB may be related to gains as high as 0.5.

One way to reduce the noise formants is by decreasing the related gain factors, e.g., by choosing the generalized Wiener function with $\beta_2 = 5$. However, then important components of the speech signal will be distorted. Thus, with this kind of filtering, there will always be an annoying noticeable residual signal.

If the gain function is *estimated*, then the speech-like noise spectrum will also be an estimate of the true one. Thus, it becomes important to introduce the signal subspace in order to avoid a flaw of the estimator at low spectral SNR. Obviously, the estimation problem will have influence on the musical noise, but it is not the source.

The musical noise can be reduced (eliminated), when a subspace based estimation method is combined with a multi-microphone enhancement technique like the Delay-and-Sum beamformer (see Figure 4.15). The necessary condition is a low correlation between the noise signals in the different channels. Then, the residual noise will be incoherently added, giving a whitened output due to the Central Limit Theorem, while the signal distortion will be untouched. This is illustrated in Figure 4.16 and 4.17 using 4 and 10 microphones, respectively, where both examples correspond to the ones in Figure 4.7. Thus, for the price of increased computational complexity, a significant improvement in performance is obtained with the proposed combination.

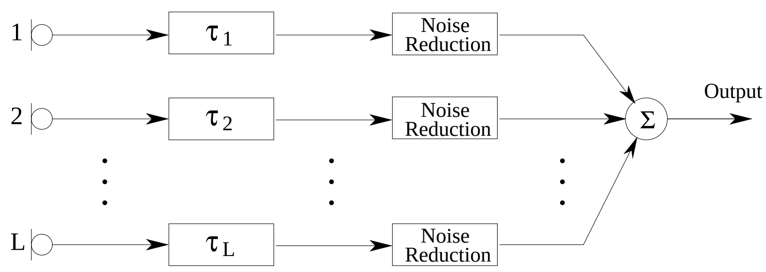


Figure 4.15 Delay-and-Sum beamformer combined with a single-microphone noise reduction technique, e.g., the signal subspace approach.

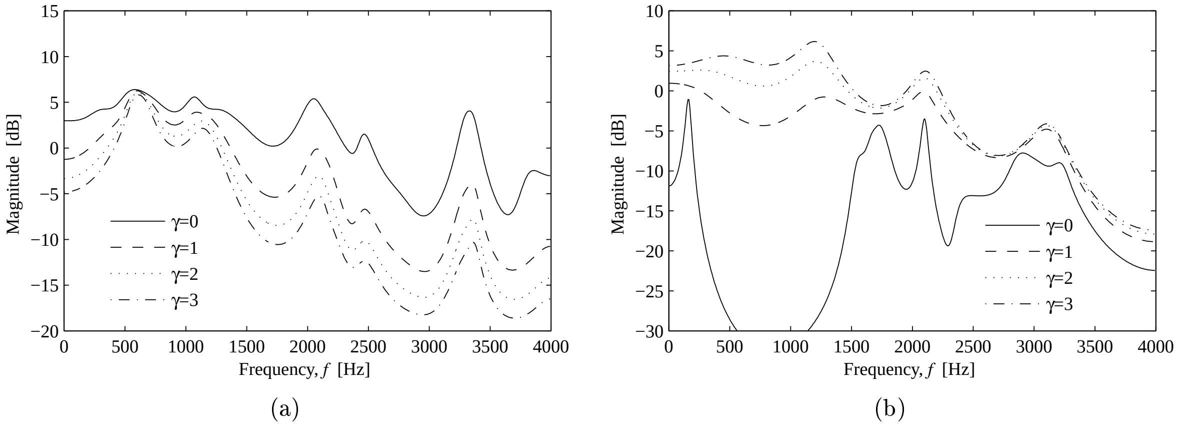


Figure 4.16 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.7, but averaged over 4 independent white noise realizations.

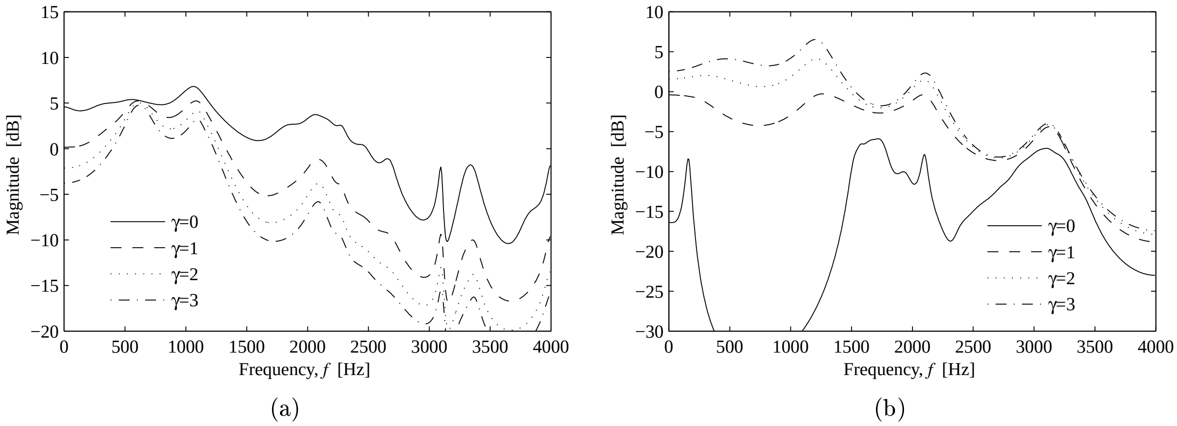


Figure 4.17 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 4.7, but averaged over 10 independent white noise realizations.

4.5 Rank Determination

As mentioned previously, the correlation matrix of speech signals has full rank, so signal subspace methods based on rank reduction must find the right tradeoff between model bias (signal distortion $\mathbf{r}_s$) and model variance (residual noise $\mathbf{r}_n$), when reconstructing signals from noisy data. The goal is to minimize the sum of the two terms, i.e., the residual signal $\mathbf{r} \in \mathbb{R}^m$. The general principle has been analyzed by Scharf et al. [102, 100, 101], and the results are illustrated here by applying it to the speech enhancement problem.

The residual signals as defined by Equation (4.49) are repeated here for convenience

$$\mathbf{r} = \hat{\mathbf{s}} - \mathbf{s} = (\mathbf{W} - \mathbf{I}_m)\mathbf{s} + \mathbf{W}\mathbf{n} = \mathbf{r}_s + \mathbf{r}_n \tag{4.106}$$

and the energy of the two terms $\mathbf{r}_s$ and $\mathbf{r}_n$ are given by (4.50) and (4.51), i.e., by using the gain matrix as defined by (4.87), the following equations are obtained

$$\epsilon_s^2 = \mathrm{tr}E\{\mathbf{r}_s\mathbf{r}_s^T\} = \mathrm{tr}\left((\mathbf{G}-\mathbf{I}_m)\mathbf{\Lambda}_s(\mathbf{G}-\mathbf{I}_m)^T\right) \tag{4.107}$$

$$\epsilon_n^2 = \mathrm{tr}E\{\mathbf{r}_n\mathbf{r}_n^T\} = \nu_{noise}^2\mathrm{tr}\left(\mathbf{G}\mathbf{G}^T\right) \tag{4.108}$$

For the LS estimator with $\mathbf{G}_1 = \mathbf{I}_p$, the bias ϵ_s^2 is simply the sum of signal-only eigenvalues that are discarded, and the variance ϵ_n^2 is a linear function of the signal subspace dimension p, i.e.,

$$\epsilon_s^2 = \sum_{i=p+1}^{m} \lambda_{s,i} \quad \text{and} \quad \epsilon_n^2 = p\nu_{noise}^2 \tag{4.109}$$

In this case, the low-rank projection improves the SNR whenever

$$\epsilon_s^2 + \epsilon_n^2 < m\nu_{noise}^2 \tag{4.110}$$

In general, the optimum choice of rank minimizes

$$p_{opt} = \arg\min_p(\epsilon_s^2 + \epsilon_n^2) \tag{4.111}$$

which occurs near the point where the rate of decrease in ϵ_n^2 approximately equals the rate of increase in ϵ_s^2. All of this is illustrated in Figure 4.18, where the average power of the residual noise $\mathbf{r}_n$ and the signal distortion $\mathbf{r}_s$ are plottet as a function of the signal subspace dimension p for the LS and MV estimators. The data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ represents the voiced speech frame of 160 samples added white noise (SNR = 0 dB), and the average is taken over 100 independent noise realizations.

Notice that in the LS case, the total residual power has a clear minimum which means that a proper choice of p is important for this estimate. However, the signal distortion and thereby the optimum rank depends on the eigenvalue spread of the speech signal, i.e., unvoiced frames with a smaller eigenvalue spread will for example require a higher signal subspace dimension for a given SNR. The problem could be solved by tracking the optimal rank with, e.g., the NEE-algorithm in [25], but such approaches are computationally expensive.

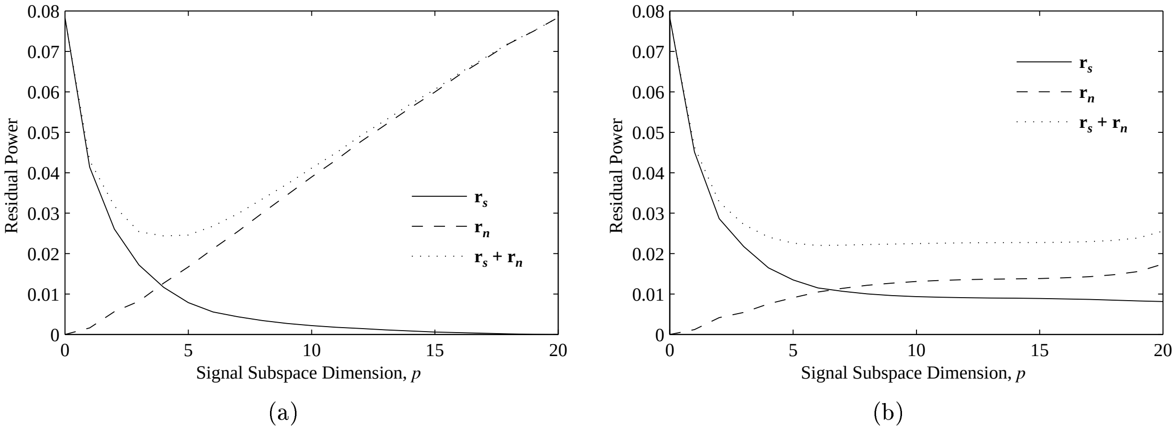
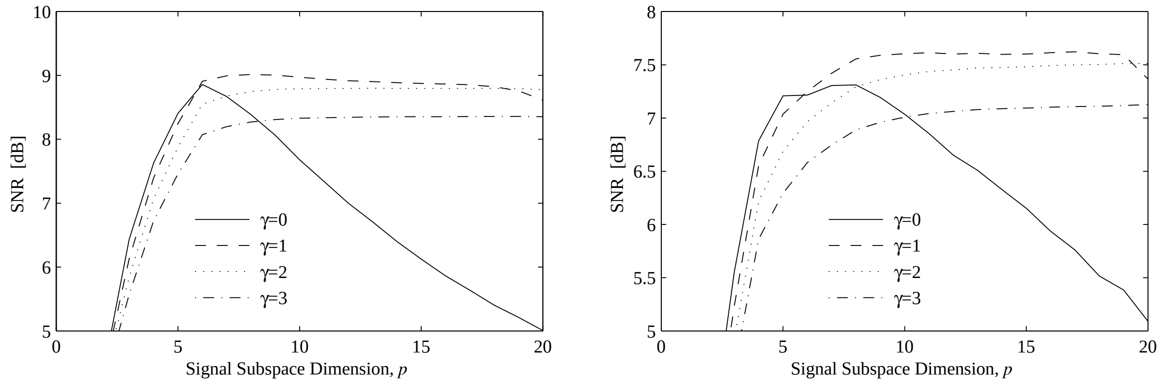


Figure 4.18 Average power of the residual noise $\mathbf{r}_n$ and the signal distortion $\mathbf{r}_s$ as function of the signal subspace dimension p for the LS estimator (a) and MV estimator (b). The data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ represents the voiced speech frame of 160 samples added white noise (SNR = 0dB), and the average is taken over 100 independent noise realizations.

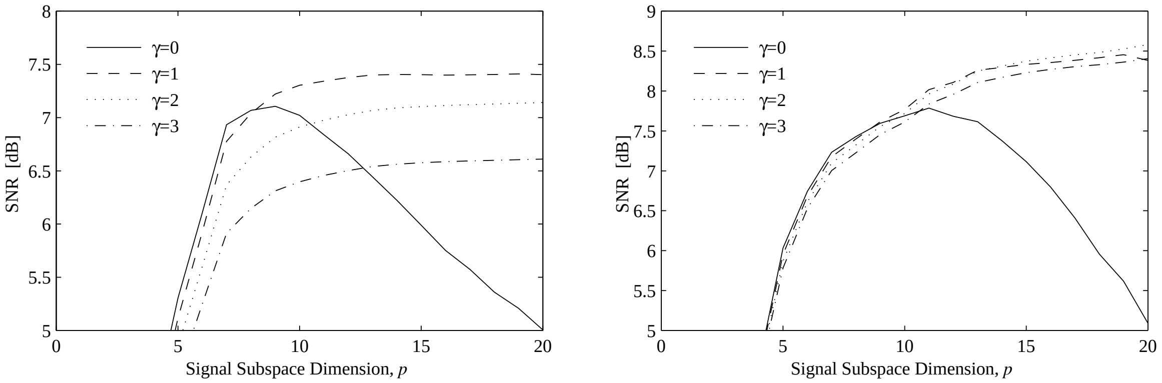
For the MV estimate, the choice of p is not so critical, since the Wiener gain function implicitly introduces a signal subspace. Thus, once p is large enough, a fixed value of p can be used, which is also desirable in real-time speech processing applications. This conclusion can be extended to the TDC and SDC estimates as illustrated in Figure 4.19 – 4.21, where the average SNR of the reconstructed speech segment (see Section 4.6) is plotted as a function of the signal subspace dimension p using 100 independent noise realizations (SNR = 5 dB and the noise variance is estimated from an initial segment). The examples in the Figures cover both the voiced and unvoiced frame, and the white and colored noise case. Notice that the unvoiced segment requires a larger value of p, and that colored noise can both degrade and improve the performance depending on the actual speech and noise spectra.

The fixed rank decision criterions are that p should be as small as possible to avoid flaws of the estimator, and that the degradation of reconstructed noise-free speech should be sufficiently small. Simulations and informal listening tests made by Jensen [61] have shown that for fixed values of $p \geq 12$, the quality of the reconstructed speech is satisfactory. Finally note that a fixed signal subspace dimension always introduces signal distortion, i.e., frames with high SNRs will be degraded.



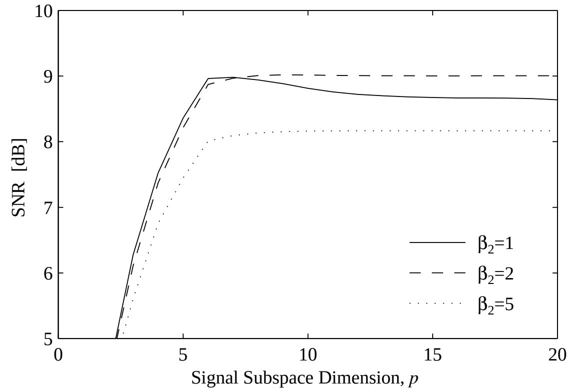
(a) (b)

Figure 4.19 Average SNR of a reconstructed noisy (voiced) speech segment using a TDC estimator with the listed γ values, and SNR=5dB. The average is taken over 100 independent white noise realizations (a) or AR(1,-0.7) noise realizations (b).

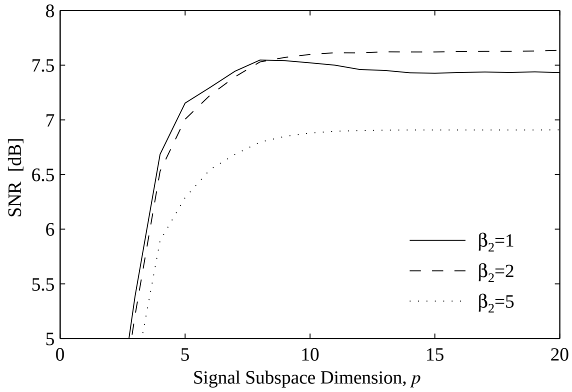


(a) (b)

Figure 4.20 As Figure 4.19 but for the unvoiced speech segment.



(a)



(b)

Figure 4.21 As Figure 4.19 but by using a SDC estimator with the listed β_2 values.

4.6 Signal Reconstruction by Toeplitz Matrix Approximation

This section discuss how to construct the enhanced output signal $\hat{\mathbf{s}}_K$ from the reduced-rank and generally unstructured matrix $\hat{\mathbf{S}}$ (4.88).

Consider the observed data vector $\mathbf{x}_K \in \mathbb{R}^K$ (3.7), which has a signal component $\mathbf{s}_K$ to be estimated, and an additive noise component $\mathbf{n}_K$

$$\mathbf{x}_K = \mathbf{s}_K + \mathbf{n}_K \tag{4.112}$$

where the index K is used for time sequences with K samples. The time signals are represented by the Toeplitz matrix formula as defined by (3.8)

$$\mathbf{X} = \mathbf{S} + \mathbf{N} \tag{4.113}$$

which is redundant, because the data samples are repeated in different locations within the matrix.

Using one of the signal subspace methods to estimate the rank-p signal matrix $\mathbf{S}$ from the Toeplitz data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ generally spoil the structure, i.e., results in a non-Toeplitz estimation matrix $\hat{\mathbf{S}}$. So, it is not possible to directly obtain a time signal vector

$$\hat{\mathbf{s}}_K = \begin{pmatrix} \hat{s}_1 & \hat{s}_2 & \cdots & \hat{s}_K \end{pmatrix}^T, \qquad K = m + n - 1 \tag{4.114}$$

corresponding to it. However, a *least squares* estimate of the time signal vector $\mathbf{s}_K$ can be obtained from the reduced-rank matrix $\hat{\mathbf{S}}$ as described by Tufts and Shah in [117], and discussed below.

The time signal estimation is based on the fact that the rank reduced matrix is made up of two parts, c.f., Equation (4.72)

$$\hat{\mathbf{S}} = \mathbf{S} + \mathbf{R} \tag{4.115}$$

where $\mathbf{S}$ is the signal-only matrix having a Toeplitz structure, and $\mathbf{R}$ is the residual matrix arising due to the noise matrix $\mathbf{N}$, i.e., the residual noise term, and due to any error in the estimation of the desired signal matrix $\mathbf{S}$, i.e., the signal distortion term. In general, the residual matrix $\mathbf{R}$ will not have a Toeplitz structure.

It is useful to rewrite matrix formula (4.115) as a vector formula by stacking the columns in each of the matrices $\hat{\mathbf{S}}$, $\mathbf{S}$ and $\mathbf{R}$

$$\hat{\mathbf{s}}' = \mathbf{s}' + \mathbf{r}' = \mathbf{M}\mathbf{s}_K + \mathbf{r}' \tag{4.116}$$

where $\hat{\mathbf{s}}', \mathbf{s}', \mathbf{r}' \in \mathbb{R}^{mn}$. The linear transformation represented by the matrix $\mathbf{M} \in \mathbb{R}^{mn \times K}$ on the time signal vector $\mathbf{s}_K \in \mathbb{R}^K$ can be visualized as a redundant rearrangement of the K elements in the redundant signal vector $\mathbf{s}'$. This matrix $\mathbf{M}$ is made up entirely of zeros and ones, and is obtained by exploiting the Toeplitz symmetry of the signal matrix $\mathbf{S}$

$$\mathbf{M}^T = \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_m^T \\ \mathbf{m}_{m+1}^T \\ \vdots \\ \mathbf{m}_{K-m+1}^T \\ \vdots \\ \mathbf{m}_K^T \end{pmatrix} = \left(\begin{array}{cccccccccccc} & & & & & & & & 1 & & & \\ & & & & 1 & & & & & 1 & & \\ & & & & & \ddots & & & & & \ddots & \\ & & & & & & 1 & & & & & 1 \\ & & & & & & & 1 & & & & \\ & & & \iddots & & & & & & & & \\ 1 & & & & & & & & & & & \\ & \ddots & & & & & & & & & & \\ & & 1 & & & & & & & & & \end{array}\right) \tag{4.117}$$

Now, let us consider the problem of estimating the signal vector $\mathbf{s}_K$ using the representation of the reduced rank matrix $\hat{\mathbf{S}}$ as the sum of a redundant Toeplitz structured signal component and the residual vector. The mn linear equations of formula (4.116) are approximately solved by the method of least squares, to obtain an estimate of $\mathbf{s}_K$

$$\min_{\mathbf{s}_K} \|\hat{\mathbf{s}}' - \mathbf{M}\mathbf{s}_K\|_2 \quad \Rightarrow \quad \hat{\mathbf{s}}_K = \mathbf{M}^+\hat{\mathbf{s}}' = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\hat{\mathbf{s}}' \tag{4.118}$$

Note, that the equivalent matrix formulation of Equation (4.118) is

$$\min_{\mathbf{s}_K} \|\hat{\mathbf{S}} - \mathcal{T}_n(\mathbf{s}_K)\|_F \tag{4.119}$$

where $\mathcal{T}_n$ is the Toeplitz operator (see definition in Section 3.1.1). It turns out that the operation of the linear transformation $\mathbf{M}^+$ on the vector $\hat{\mathbf{s}}'$ is nothing but an arithmetic averaging of the subdiagonals of $\hat{\mathbf{S}}$ as originally proposed in [116, 115, 65], i.e.,

$$\hat{\mathbf{s}}_K = \mathcal{A}(\hat{\mathbf{S}}) \tag{4.120}$$

where $\mathcal{A}$ is the diagonal averaging operator. The factor $(\mathbf{M}^T\mathbf{M})^{-1}$ is an $K \times K$ diagonal matrix given by

$$(\mathbf{M}^T\mathbf{M})^{-1} = \mathrm{diag}(1, 2^{-1}, \ldots, n^{-1}, n^{-1}, \ldots, n^{-1}, \ldots, 2^{-1}, 1) \tag{4.121}$$

which account for the fact that the first $n-1$ and last $n-1$ diagonals of $\hat{\mathbf{S}}$ do not have full length n.

The resulting Toeplitz matrix is associated with a unique vector $\hat{\mathbf{s}}_K$ of time samples, but then the exact rank is lost. Hence, it is commonly used to repeat the low-rank approximation and diagonal averaging a number of times to produce a low-rank Toeplitz matrix approximation [16, 17, 126]. It is proven that such iterative usage always converges [16]. However, the signal estimate to which this iterative algorithm converges need not be optimal in any signal estimation sense [83]. In speech processing, the improvement of the iteration is negligible [62].

4.6.1 Interpretation

The physical interpretation of signal reconstruction from reduced rank matrices by averaging along the diagonals is examined in [26, 50, 51] and discussed below.

First, insert the estimate $\hat{\mathbf{S}}$ (4.88) into (4.120) and use the fact that averaging and summation are interchangeable

$$\hat{\mathbf{s}}_K = \mathcal{A}\left(\sum_{i=1}^{p} g_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T\right) = \sum_{i=1}^{p} g_i \mathcal{A}(\sigma_i \mathbf{u}_i \mathbf{v}_i^T) \tag{4.122}$$

Now consider the rank-one matrix

$$\mathbf{u}_i \mathbf{v}_i^T = \begin{pmatrix} u_{1i}v_{1i} & u_{1i}v_{2i} & u_{1i}v_{3i} & \cdots & u_{1i}v_{ni} \\ u_{2i}v_{1i} & u_{2i}v_{2i} & u_{2i}v_{3i} & \cdots & u_{2i}v_{ni} \\ \vdots & \vdots & \vdots & & \vdots \\ u_{mi}v_{1i} & u_{mi}v_{2i} & u_{mi}v_{3i} & \cdots & u_{mi}v_{ni} \end{pmatrix} \tag{4.123}$$

for which the diagonal summation operation can be expressed as the linear convolution of the signals represented by $\mathbf{u}_i$ and $\mathbf{v}_i$ in reverse order. This is equivalent to a circular convolution of the sequences appended with zeros, until each has $K = m+n-1$ samples, i.e., the reconstructed signal becomes

$$\hat{s}_k = (\mathbf{m}_k^T \mathbf{m}_k)^{-1} \sum_{i=1}^{p} g_i \sigma_i \left(\sum_{l=1}^{K} u_{l,i} v_{l-k+n,i}\right) \quad \text{for} \quad k = 1, \ldots, K \tag{4.124}$$

where the linear transformation matrix $\mathbf{M}$ defined by (4.117) has been used. Equation (4.124) can be reformulated by means of the Toeplitz operator $\mathcal{T}_n$

$$\hat{\mathbf{s}}_K = (\mathbf{M}^T\mathbf{M})^{-1} \sum_{i=1}^{p} g_i \sigma_i \mathcal{T}_n \begin{pmatrix} \mathbf{0}_{n-1} \\ \mathbf{u}_i \\ \mathbf{0}_{n-1} \end{pmatrix} \mathbf{J}_n \mathbf{v}_i \tag{4.125}$$

where $\mathbf{J}_n$ is an $n \times n$ matrix with nonzero elements equal to one only on the anti-diagonal, i.e., it reverses the elements of $\mathbf{v}_i$. Thus, the output signal $\hat{\mathbf{s}}_K$ is the result of the *backward* filtering operation of the left singular vectors $\mathbf{u}_i$ through the eigenfilters $\mathbf{v}_i$.

On the other hand is $\mathbf{u}_i$ also the *forward* filtered version of the input signal $\mathbf{x}_K$ passed through the same eigenfilters $\mathbf{v}_i$

$$\mathbf{u}_i = \mathbf{X}\mathbf{v}_i \sigma_i^{-1} = \mathcal{T}_n(\mathbf{x}_K)\mathbf{v}_i \sigma_i^{-1} \tag{4.126}$$

Notice, that the first and last part of the linear convolution of the signals $\mathbf{x}_K$ and $\mathbf{v}_i$ are omitted in this case.

From (4.125) and (4.126) the precise relationship between the input vector $\mathbf{x}_K$ and the output vector $\hat{\mathbf{s}}_K$ can be expressed as follows

$$\hat{\mathbf{s}}_K = (\mathbf{M}^T\mathbf{M})^{-1} \sum_{i=1}^{p} g_i \mathcal{T}_n \begin{pmatrix} \mathbf{0}_{n-1} \\ \mathcal{T}_n(\mathbf{x}_K)\mathbf{v}_i \\ \mathbf{0}_{n-1} \end{pmatrix} \mathbf{J}_n \mathbf{v}_i \tag{4.127}$$

Basic signal processing theory shows that successive forward and backward filtering of a signal results in a zero phase filteret version of it (see e.g. [88]). Therefore, averaging along the diagonals of the matrix $\hat{\mathbf{S}}$ is equivalent to the weighted summation of p zero phase filtered versions of the

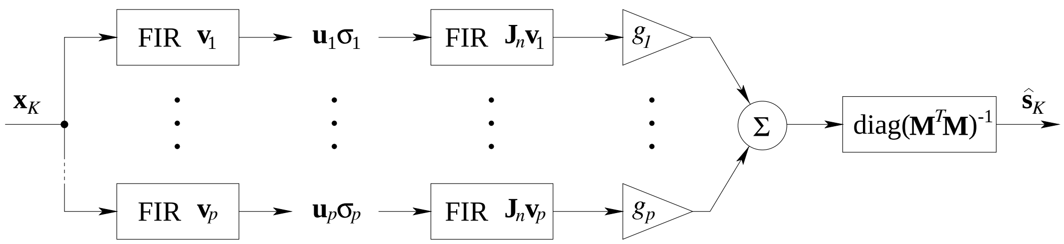


Figure 4.22 Zero phase filter model for rank-p signal subspace estimation followed by averaging along diagonals.

input signal $\mathbf{x}_K$ through the eigenfilters $\mathbf{v}_i$ and their reverse counterparts. The diagonal elements of the matrix $(\mathbf{M}^T\mathbf{M})^{-1}$ represents an K-point window.

The filter structure is illustrated in Figure 4.22, where the first and last part corresponds to (4.126) and (4.125), respectively. It is a simplified overview and does not explain the initial and last part of the filtering operation.

Another problem arises when there is a shift between two data matrices $\mathbf{X}'$ and $\mathbf{X}$, so averaging along diagonals include elements from both matrices as shown in Figure 4.23 for the time sample x_k.

If we first look at the ith forward eigenfilter, then all the filter coefficients $\mathbf{v}'_i$ are replaced with $\mathbf{v}_i$, when the row of input samples corresponding to the storage of the filter goes from $\mathbf{x}'^T_m$, the last row of $\mathbf{X}'$, to $\mathbf{x}_1^T$, the first row of $\mathbf{X}$. This results in a sequence of stacked $\mathbf{u}_i$ vectors as input to the ith backward eigenfilter. The coefficients of this filter must be changed one by one as the shift from $\mathbf{u}'_i$ to $\mathbf{u}_i$ pass through the filter storage and the two filter parts must be weighted separately as shown in Figure 4.23.

Note, that the above mentioned technique corresponds to interpolation of overlaping reconstructed frames, and that it can be further extended by overlaping adjacent data matrices and then reconstruct a subset of the rows. Note also that an equivalent formulation of the Toeplitz matrix approximation problem is to subtract from the signal x_k, zero phase filtered version of itself through the $n-p$ eigenfilters $\mathbf{v}_i$ corresponding to the $n-p$ smallest eigenvalues.

Thus, the filter interpretation of the signal subspace estimation methods both provide an appealing frequency-domain formulation and gives a possible implementation method.

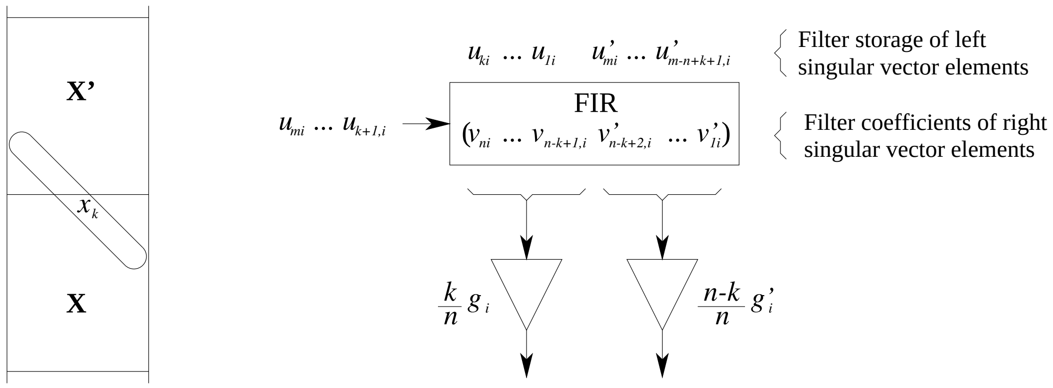


Figure 4.23 Filter model for averaging along diagonals in the case of shift between two datamatrices.

4.6.2 Signal Reconstruction and the QSVD

As in the SVD based algorithm, the output signal is computed by arithmetic averaging of the subdiagonals of $\hat{\mathbf{S}}$, i.e., $\hat{\mathbf{s}}_K = \mathcal{A}(\hat{\mathbf{S}})$.

Following the equations in Section 4.6.1, an expression for $\hat{\mathbf{s}}_K$ that leads to a filter interpretation is [51]

$$\hat{\mathbf{s}}_K = (\mathbf{M}^T\mathbf{M})^{-1} \sum_{i=1}^{p} g_i \mathcal{T}_n \begin{pmatrix} \mathbf{0}_{n-1} \\ \mathcal{T}_n(\mathbf{x}_K)\mathbf{z}_i \\ \mathbf{0}_{n-1} \end{pmatrix} \mathbf{J}_n \boldsymbol{\theta}_i \tag{4.128}$$

Thus, the filters that arise in the QSVD based algorithm have coefficients $\mathbf{z}_i$ and $\mathbf{J}_n\boldsymbol{\theta}_i$, and they are not zero-phase filters as in the white-noise case.

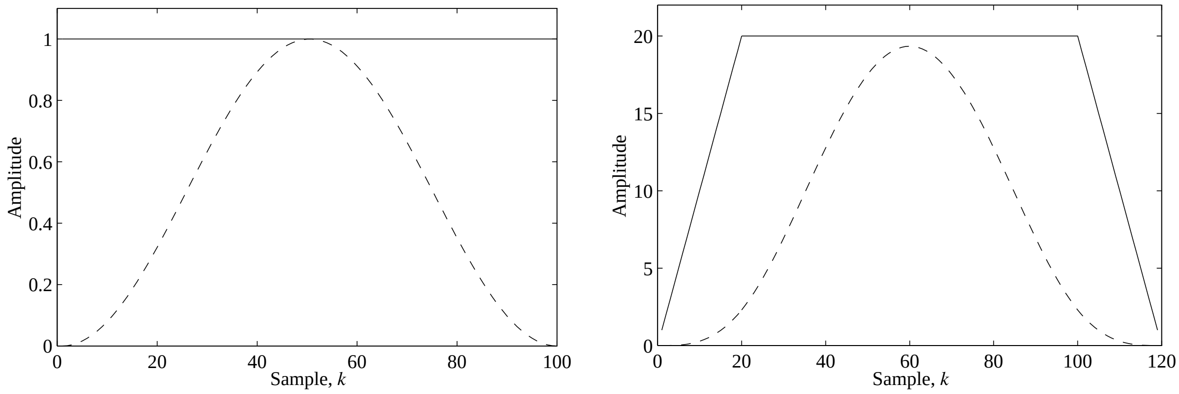
4.7 Frame Based Implementation

In most practical signal processing applications, it is necessary to work with *frames* of the signal. This is especially true if conventional analysis techniques are used on speech signals with nonstationary dynamics, i.e., it is necessary to select a portion of the signal that can reasonably be assumed to be stationary.

Thus, in implementing the linear signal estimators for time series, other parameters which must be specified are those characterizing the *analysis* and *synthesis* windows. The windows operates on the rows of the data matrix and is given by the diagonal matrices $\mathbf{W}_a$ and $\mathbf{W}_s$, respectively

$$\mathbf{W}_a = \mathrm{diag}(\mathbf{w}_a) \tag{4.129}$$
$$\mathbf{W}_s = \mathrm{diag}(\mathbf{w}_s) \tag{4.130}$$

where $\mathbf{w}_a \in \mathbb{R}^m$ and $\mathbf{w}_s \in \mathbb{R}^{m_s}$ is a real sequence used to select and weight the desired rows of $\mathbf{X}$ by a simple pre-multiplication process. Example plots of two commonly used window sequences are shown in Figure 4.24(a). Due to the Toeplitz structure of the data matrix (see Section 3.1.1), the corresponding windows related to the time sequences are the row windows convolved with a n-sample rectangular window (see Figure 4.24(b)).



(a) (b)

Figure 4.24 (a) Example time plots for the rectangular (solid) and Hanning (dashed) windows with length $m = 100$. (b) The windows after convolution with a rectangular window of length $n = 20$.

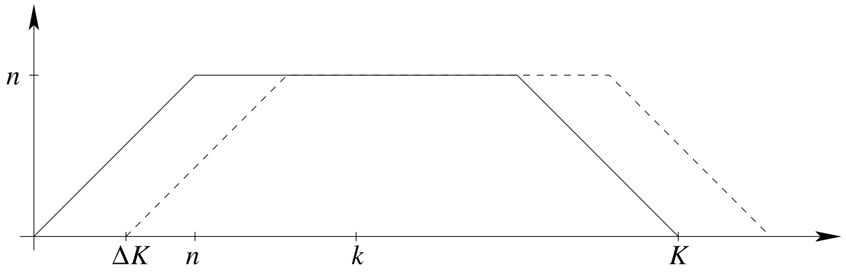


Figure 4.25 Time sequence plot of a rectangular analysis window of length K centered around time k. ΔK is the shift between adjacent frames.

First consider the analysis conditions, where it is important to have good estimates of the correlation matrix of the noisy signal and the noise process at the current time k, i.e., how to construct the matrices $\mathbf{X}$ and $\mathbf{N}$. This was discussed in Section 3.1.1 for Toeplitz matrices using a rectangular analysis window $\mathbf{W}_a = \mathbf{I}_m$ corresponding to the time sequence window shown in Figure 4.25.

In this case, the window length K is given by

$$K = m + n - 1 \tag{4.131}$$

and $m \geq n$ or equivalently $K \geq 2n - 1$. The windowed data segments are allowed to overlap. Thus, the data segments can be represented as

$$\mathbf{x}_{K+i\Delta K} = \begin{pmatrix} x_{1+i\Delta K} & \cdots & x_{K+i\Delta K} \end{pmatrix}, \qquad i = 0, 1, \ldots, F-1 \tag{4.132}$$

where ΔK is the shift between adjacent frames (see Figure 4.25), and F is the number of frames in the time sequence. Observe that if $\Delta K = K$, the segments do not overlap, and that a frame shift corresponds to a rank-ΔK update/downdate of the data matrix.

When applying a given analysis window $\mathbf{W}_a$ to $\mathbf{X}$, a constant weighting c_a^2 of the correlation matrix is obtained

$$E\{\mathbf{X}^T \mathbf{W}_a^2 \mathbf{X}\} = \mathbf{R}_x \sum_{i=1}^{m} w_{a,i}^2 = \mathbf{w}_a^T \mathbf{w}_a \mathbf{R}_x = c_a^2 \mathbf{R}_x \tag{4.133}$$

so the whiteness of the input noise is always preserved. Note, that in the prewhitening case, where different analysis windows $\mathbf{W}_{a,X}$ and $\mathbf{W}_{a,N}$ can be used on $\mathbf{X}$ and $\mathbf{N}$, the noise-only part $\mathbf{N}$ must be weighted by

$$\frac{c_{a,X}}{c_{a,N}} \tag{4.134}$$

When analyzing a nonstationary signal like speech, selection of a window length is an important consideration, because sliding it along in time, a longer window will require a longer period to cross transitional boundaries in the signal and events from different quasi-stationary regions will tend to be blurred together more frequently than if the window was shorter. This is also the argument for not using a rectangular analysis window and thereby omitting information in the correlation estimation. A Hamming window for example has smoother truncations with maximum weight at the current time k. This is equivalent to the *spectral-temporal resolution trade-off* [23, page 20].

Now consider the synthesis window, which is applied to $\mathbf{X}$ before the filter matrix $\mathbf{W}$ (4.88), i.e., the windowed and enhanced vectors are then given by

$$\hat{\mathbf{S}} = \begin{pmatrix} \mathbf{0} & \mathbf{W}_s & \mathbf{0} \end{pmatrix} \mathbf{X}\mathbf{W} = \begin{pmatrix} \mathbf{0} & \mathbf{W}_s & \mathbf{0} \end{pmatrix} \mathbf{W}_a^{-1} \mathbf{U}_{X1} \mathbf{\Sigma}_{X1} \mathbf{G}_1 \mathbf{V}_{X1}^T \tag{4.135}$$

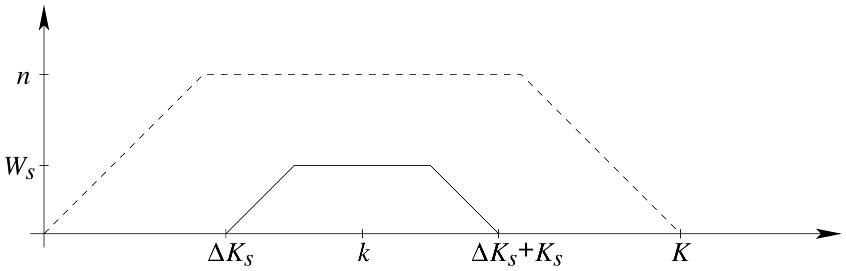


Figure 4.26 Time sequence plot of a rectangular synthesis window of length K_s centered around time k. W_s is the weighting factor and ΔK_s is the offset to the analysis window (dashed).

where $\hat{\mathbf{S}} \in \mathbb{R}^{m_s \times n}$, and $\mathbf{W}_s \in \mathbb{R}^{m_s \times m_s}$ is padded with zeros in order to operate on the middle rows of $\mathbf{X}$. Note, that the filter matrix $\mathbf{W}$ is obtained from the SVD of $\mathbf{W}_a\mathbf{X}$, so if the $\mathbf{U}_{X1}$ matrix is used to obtain the estimate $\hat{\mathbf{S}}$, it must be reweighted by $\mathbf{W}_a^{-1}$.

The time sequence window for a rectangular synthesis window is shown in Figure 4.26. The window length K_s and weighting factor W_s are given by

$$K_s = m_s + n - 1 \tag{4.136}$$

$$W_s = \begin{cases} m_s, & m_s < n \\ n, & m_s \geq n \end{cases} \tag{4.137}$$

where $m_s \geq 1$ or $K_s \geq n$. The window is centered around k and the offset between the analysis and the synthesis window is

$$\Delta K_s = \left\lfloor \frac{K - K_s}{2} \right\rfloor \tag{4.138}$$

where $\lfloor \cdot \rfloor$ is the truncation operator to nearest lower integer. Thus, the shift between adjacent synthesis windows is also ΔK and $K \geq K_s \geq \Delta K$.

The enhanced row vectors in $\hat{\mathbf{S}}$ are combined using the overlap and add synthesis approach, i.e., each diagonal in $\hat{\mathbf{S}}$ which relates to the same time sample is summed. The obtained K_s-sample time sequence $\hat{\mathbf{s}}_{K_s}$ must then be reweighted by the corresponding time sequence window which are the row window $\mathbf{w}_s$ convolved with a W_s-sample rectangular window due to the Toeplitz structure. Another issue is the overlap between adjacent frames, where the windows are also used in the interpolation. Thus, the overall filter structure can be illustrated by the block diagram in Figure 4.27.

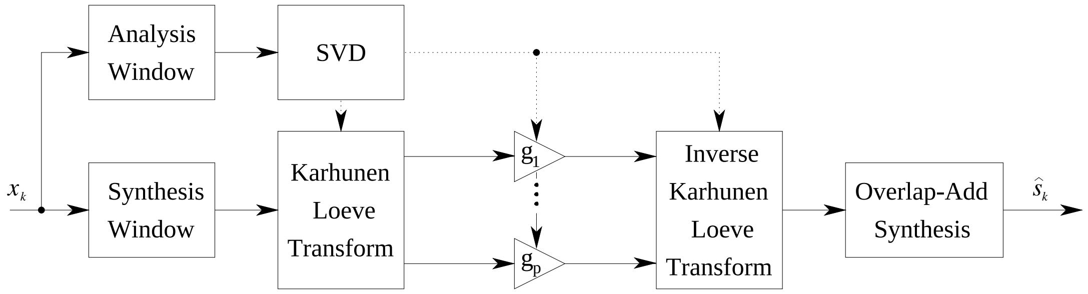


Figure 4.27 Filter structure.

4.8 Summary

It is shown that a number of nonparametric linear estimators based on second-order statistics can be defined in an empirical way based on the data matrix, where the latter expressions are based on the SVD instead of the eigendecomposition. The contribution here is mainly the unified framework, where relations between the different estimators are pointed out. However, the empirical TDC and SDC estimates are alternative formulations based on the data matrix.

The empirical estimators has been compared with the optimal ones, and it is demonstrated that the MV estimator introduces gains far away from the optimal Wiener filter. This is explained by the fact that the white noise assumption is not exactly satisfied for the considered frame length. The empirical TDC and SDC estimators are less sensitive to the estimate of the noise variance (for increasing parameter values), and they also introduce gains closer to the optimal filters.

However, the estimation errors illustrates the importance of explicitly introducing a signal subspace. Thus, the rank decision problem has been discussed, and the conclusion is that a fixed rank can be used for the Wiener based estimators.

The argument for introducing the TDC and SDC estimates is a reduced residual noise power with a spectrum that looks like the one for the clean speech, i.e., in accordance to the masking threshold of the auditory system. This has been verified by an example, which illustrates the tradeoff between residual noise and signal distortion. In the colored noise case, it is shown how prewhitening may increase the signal distortion significantly.

The relation between the signal subspace approach and spectral subtraction has been considered, and the origin of the musical noise has been pointed out. Combination of noise reduction methods based on a single microphone with the delay-and-sum beamformer is shown to be an efficient way to reduce (eliminate) the musical noise.

Finally, a filter formulation and frame based implementation of the speech enhancement methods are discussed, where a number of practical aspects are considered.

Chapter 5

Parametric Signal Estimators

None of the estimation methods in the last chapter assume a known speech model, so it is the hope that introducing a model can be used to improve the noise reduction capabilities.

Two well-known speech models are considered here, i.e., the output of an autoregressive (AR) system, and the sum of damped sinusoids. The former will be used throughout this chapter, while the latter is implicitly used in the last chapter.

In the noisy case, model based estimation is a nonlinear problem, which is normally solved by iterative techniques. Thus, a discussion about the different methods is given.

Finally, a new idea based on multi-microphone inverse filtering is presented, where the solution is obtained by subspace methods.

5.1 AR Difference Equation

As discussed in Section 2.3.2, a majority of speech sounds can be described by the transfer function of an autoregressive (AR) system (2.14), repeated here for convenience

$$H(z) = \frac{b_0}{A(z)} = \frac{b_0}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{5.1}$$

where a_i are the filter coefficients, p is the model order and b_0 is a gain parameter. The form of excitation applied to this filter is either a sequence of delta pulses δ_k or white noise w_k corresponding to the voiced or unvoiced speech sounds, respectively. Thus, the *synthesized* speech s_k is characterized by the following difference equation (voiced example)

$$s_k = -\sum_{i=1}^{p} a_i s_{k-i} + b_0 \delta_k \tag{5.2}$$

When written out for $k = 0, 1, \ldots, K-1$, these recursions produce the Toeplitz matrix equations

$$\begin{pmatrix} 1 & & & & & \\ a_1 & 1 & & & \mathbf{0} & \\ \vdots & \ddots & \ddots & & & \\ a_p & \cdots & a_1 & 1 & & \\ & \ddots & & \ddots & \ddots & \\ \mathbf{0} & & a_p & \cdots & a_1 & 1 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_p \\ \vdots \\ s_{K-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.3}$$

The last $(K-p)$ equations are written as

$$\mathbf{A}^T \mathbf{s} = \mathbf{0} \tag{5.4}$$

where the full rank matrix $\mathbf{A}^T \in \mathbb{R}^{(K-p)\times K}$ is denoted the *prediction error* matrix, i.e.,

$$\mathbf{A}^T = \begin{pmatrix} a_p & \cdots & a_1 & 1 & & \mathbf{0} \\ & \ddots & & \ddots & \ddots & \\ \mathbf{0} & & a_p & \cdots & a_1 & 1 \end{pmatrix} \tag{5.5}$$

The vector model (5.4) may be equivalently written as the covariance equations of linear prediction by organizing the coefficients a_i into a parameter vector $\mathbf{a}$, and the synthesized speech signal into a data matrix $\mathbf{S}$, i.e.,

$$\mathbf{A}^T\mathbf{s} = \mathbf{S}\mathbf{a} = \begin{pmatrix} s_0 & \cdots & s_p \\ s_1 & \cdots & s_{p+1} \\ \vdots & & \vdots \\ s_{K-1-p} & \cdots & s_{K-1} \end{pmatrix} \begin{pmatrix} a_p \\ \vdots \\ a_1 \\ 1 \end{pmatrix} = \mathbf{0} \tag{5.6}$$

The covariance equations show that the rank of the signal-only matrix $\mathbf{S}$ is, in fact, p and not $(p+1)$. It further shows that the parameter vector $\mathbf{a}$ lies in the null space of the matrix $\mathbf{S}$, i.e., the span of the prediction error matrix $\mathbf{A}$.

5.2 Modal Decomposition

Another commonly used way of representing the speech signal is in terms of a finite sum of p complex *modes* z_i [94, 95]

$$s_k = \sum_{i=1}^{p} c_i z_i^k \tag{5.7}$$

When written out for $k = 0, 1, \ldots, K-1$, the corresponding vector model is

$$\begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{K-1} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ z_1 & \cdots & z_p \\ \vdots & & \vdots \\ z_1^{K-1} & \cdots & z_p^{K-1} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{pmatrix} \tag{5.8}$$

or

$$\mathbf{s} = \mathbf{V}\mathbf{c} \tag{5.9}$$

where $\mathbf{V} \in \mathbb{C}^{K\times p}$ is a complex Vandemonde matrix and $\mathbf{c}$ is a vector of mode weights. The ith column of $\mathbf{V}$ is called the ith mode, and is linearly independent of the remaining columns of $\mathbf{V}$, provided $z_i \neq z_j$.

The modal representation (5.7–5.9) shows that the *synthesized* speech signal s_k obeys a linear model of the form (3.1), and that it is a damped complex sinusoid model whose ith mode is given by

$$\mathbf{v}_i = \begin{pmatrix} 1 & \rho_i e^{j\omega_i} & \cdots & \rho_i^{K-1} e^{j\omega_i(K-1)} \end{pmatrix}^T, \qquad i = 1, \ldots, p \tag{5.10}$$

where $0 \leq \omega_i \leq \pi$ and $\rho_i \leq 1$. However, as pointed out in Section 4.1.2, the parameters in the matrix $\mathbf{V}$ need not be explicitly known in implementing, e.g., the LS estimator. Thus, for this model, the estimator should be obtained from the eigendecomposition of the correlation matrix, i.e., there is no reason for the solution of a hard parameter estimation problem.

5.3 Noisy Signal

Now, the noise signal n_k is added to the speech signal s_k to produce the noisy measurements

$$x_k = s_k + n_k \quad \leftrightarrow \quad X(z) = S(z) + N(z) \tag{5.11}$$

The noise process is assumed locally stationary, white and uncorrelated with the speech signal. In the colored noise case, a prewhitening transformation can always be applied to the signal as discussed in section 3.6, and since the noise is assumed to be slightly time varying, the noise in each speechless section is modelled for the succeeding speech section.

Using (2.15) and (2.16) in (5.11), the following models are obtained for the voiced sounds

$$x_k * a_k = b_0\delta_k + n_k * a_k \quad \leftrightarrow \quad X(z) = \frac{b_0}{A(z)} + N(z) = \frac{b_0 + A(z)N(z)}{A(z)} \tag{5.12}$$

and for the unvoiced sounds

$$x_k * a_k = b_0 w_k + n_k * a_k \quad \leftrightarrow \quad X(z) = \frac{b_0 W(z)}{A(z)} + N(z) = \frac{b_0 W(z) + A(z)N(z)}{A(z)} \tag{5.13}$$

These equations can be equivalently represented by a common ARMA model as follows [63]

$$x_k * a_k = w'_k * c_k + n'_k * a_k \quad \leftrightarrow \quad X(z) = \frac{C(z)}{A(z)} W'(z) + N'(z) \tag{5.14}$$

where w'_k is another white noise sequence, c_k is a *finite* MA filter given by

$$C(z) = 1 + \sum_{i=1}^{p} c_i z^{-i} \tag{5.15}$$

and n'_k represent the error in modelling the nominator of (5.12) and (5.13) with a *finite* MA impulse response. Like the additive noise n_k, the model noise n'_k is assumed locally stationary, white and independent of s_k, but compared with n_k, the power spectral density of n'_k will normally be much smaller.

Thus, an AR model in additive white noise can be represented by an ARMA model and the MA part of this model is nothing but the additive noise in the signal.

5.4 Speech Modelling and Wiener Filtering

The Wiener filtering approach used in spectral subtraction (see Section 2.5.3) is reconsidered in this section. However, the gain function is formulated by means of the short-time Power Spectral Density (PSD) of the clean speech, i.e.,

$$g_{Wiener}(m) = \frac{\hat{\Gamma}_s(m)}{\hat{\Gamma}_s(m) + \hat{\Gamma}_n(m)} \tag{5.16}$$

The noise PSD, i.e., $\hat{\Gamma}_n(m)$, is estimated during periods of silence, while estimation of the speech PSD, i.e., $\hat{\Gamma}_s(m)$, is a more difficult problem, solved by estimation of speech parameters in an all-pole (AR) model.

The parameter estimation procedures that result in linear equations (5.6) without background noise become nonlinear when noise is introduced. However, by using iterative algorithms, the estimation procedure is linear at each iteration. In the next section, several methods for modelling speech in such an iterative framework are considered.

5.4.1 All-Pole Modelling

Suppose $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the measurement matrix defined by (3.6), where the columns of $\mathbf{S}$ are known to lie in the rank-p subspace $\langle \mathbf{H} \rangle$, but the subspace is unknown. The least squares approximation method in Section 4.1.2 considered the simultaneously estimation of $\mathbf{\Theta}$ and $\mathbf{H}$ in the linear model $\mathbf{S} = \mathbf{H\Theta}$, when the structure of $\mathbf{H}$ was not specified. We proceed by reconsider the minimization problem of the squared *fitting* error (4.29) with respect to $\mathbf{H}$ or equivalently $\tilde{\mathbf{H}}$

$$\begin{aligned} \min_{\tilde{\mathbf{H}}} e^2 &= \min_{\mathbf{H}} \|\mathbf{X} - \mathbf{H}\hat{\mathbf{\Theta}}_{LS}\|_F^2 \,, \qquad && \hat{\mathbf{\Theta}}_{LS} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X} \\ &= \min_{\tilde{\mathbf{H}}} \operatorname{tr}\left(\mathbf{P}_{\tilde{\mathbf{H}}}\mathbf{X}\mathbf{X}^T\mathbf{P}_{\tilde{\mathbf{H}}}\right), && \mathbf{P}_{\tilde{\mathbf{H}}} = \mathbf{I} - \mathbf{P}_{\mathbf{H}} \\ &= \min_{\tilde{\mathbf{H}}} \sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{P}_{\tilde{\mathbf{H}}} \mathbf{x}_i \end{aligned} \tag{5.17}$$

when $\mathbf{H}$ or $\tilde{\mathbf{H}}$ must be optimized within the AR parametric class . Using the prediction error matrix $\mathbf{A}^T$ (5.5), the problem is then to determine the least squares estimate of $\mathbf{A}$ having the specific structure in (5.5) from the nonlinear optimization

$$\hat{\mathbf{A}}_{LS} = \arg\min_{\mathbf{A}} \sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{P}_{\mathbf{A}} \mathbf{x}_i \,, \qquad \mathbf{P}_{\mathbf{A}} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \tag{5.18}$$

The *prediction errors* $\mathbf{A}^T\mathbf{x}$ may be rewritten with the signal values organized into a data matrix and the autoregressive parameters organized into a parameter vector

$$\mathbf{A}^T\mathbf{x} = \mathbf{X}\mathbf{a} = \begin{pmatrix} x_0 & \cdots & x_p \\ x_1 & \cdots & x_{p+1} \\ \vdots & & \vdots \\ x_{K-1-p} & \cdots & x_{K-1} \end{pmatrix} \begin{pmatrix} a_p \\ \vdots \\ a_1 \\ 1 \end{pmatrix} \tag{5.19}$$

which gives an alternative formulation of (5.18)

$$\hat{\mathbf{A}}_{LS} = \arg\min_{\mathbf{A}} \mathbf{a}^T\mathbf{X}^T(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{X}\mathbf{a} \tag{5.20}$$

There is proposed different ways to solve the minimization problem (5.18) or (5.20):

- *The modified least squares* problem approximates the projector $\mathbf{P}_A$ by $\mathbf{A}\mathbf{A}^T$, i.e., the minimization is *modified* in the sense that $(\mathbf{A}^T\mathbf{A})^{-1}$ is neglected

$$\begin{aligned} \hat{\mathbf{A}}_{MLS} &= \arg\min_{\mathbf{A}} \boldsymbol{\xi}^T\boldsymbol{\xi} \\ &= \arg\min_{\mathbf{A}} \mathbf{a}^T\mathbf{X}^T\mathbf{X}\mathbf{a} \end{aligned} \tag{5.21}$$

  where the prediction errors $\boldsymbol{\xi}$ is also called the equation errors, and the equations $\mathbf{A}^T\mathbf{x} = \boldsymbol{\xi}$ are simply the covariance equations of linear prediction.

- *The KiSS algorithm* minimizes the quadratic form using the iteration [64]

$$\min_{\mathbf{a}_i} \mathbf{a}_i^T\mathbf{X}^T\left(\mathbf{A}_{i-1}^T\mathbf{A}_{i-1}\right)^{-1}\mathbf{X}\mathbf{a}_i \tag{5.22}$$

  where $\mathbf{a}_i$ is the approximation of $\mathbf{a}$ at the ith iteration and $\mathbf{A}_{i-1}$ is built using the coefficients in $\mathbf{a}_{i-1}$. At each iteration, the new approximation is therefore obtained with a simple quadratic minimization. In [78] is shown that the KiSS algorithm is equivalent to the original Steiglitz-Mcbride algorithm .

Other iterative methods is the *maximum a posteriori* (MAP) estimation [71] and *estimate-maximize* (EM) algorithm [24]. However, in order to obtain reliable parameters, speech-specific constraints must be applied to the problem [48].

Common for the above methods is that errors in the parameter estimates introduce signal distortion, and the musical noise will also be presented in the enhanced signal due to the Wiener filter approch.

In the next section, a totally different approach is considered. Here, noise components are identified from modelling of the noisy speech signal, and then removed by inverse filtering in the time domain only. Thus, the enhanced signal should be free of musical noise.

5.5 Subspace Methods for Multichannel Inverse Filtering

In [128], a monochannel noise reduction procedure is presented, based on ARMA modelling of the noisy signal, followed by inverse filtering in the time domain. It is concluded that this method preserves the natural sound of speech, because it involves some filtering and inverse filtering procedures in the time domain only.

The idea presented here, is to combine the ARMA modelling approach with multi-microphone inverse filtering, and to obtain the solution by subspace methods.

First, assume that the speech signal in additive white noise is *ideally* modelled as an ARMA process (5.14), i.e., zero model noise n'_k, then inverse filtering by the MA part $C(z)$ yields a signal with the following power spectral density

$$\Gamma_s(\omega) = \left| \frac{G}{A(z)A(z^{-1})} \right|_{z=e^{j\omega}} \tag{5.23}$$

where G is a gain factor. Equation (5.23) represents the spectra for the clean speech. Thus, inverse filtering the measurement signal x_k by c_k in the time domain as

$$x_k = c_k * s_k \tag{5.24}$$

yields the clean speech

$$S(z) = \frac{X(z)}{C(z)} = \frac{W'(z)}{A(z)} \tag{5.25}$$

which is driven by the white noise input $W'(z)$. However, in practice the noisy speech signal cannot be ideally modelled by an ARMA model with finite MA part, so from (5.14) the signal x_k is modelled as

$$x_k = c_k * s_k + n'_k \tag{5.26}$$

where n'_k is the model noise. Figure 5.1(a) and (b) show the AR speech model with additive white noise and the corresponding ARMA model.

There are two problems concerning inverse filtering in the classical monochannel case: First, the identification of c_k is iterative and second, the inverse MA filter has infinite length, even in the absence of model noise. As discussed in the next sections, the identification procedure and finite inverse filtering is simplified in the multichannel case, when the additive noise $n_k^{(i)}$ in the ith channel is uncorrelated with the noise signals in the other channels. Since only the MA part of the ARMA model depends on $n_k^{(i)}$, the multichannel system can be considered as the clean speech signal s_k filtered with different FIR channels $C^{(i)}(z)$ and added independent model noises $n_k'^{(i)}$. This is shown in Figure 5.1(c) for the case of L channels.

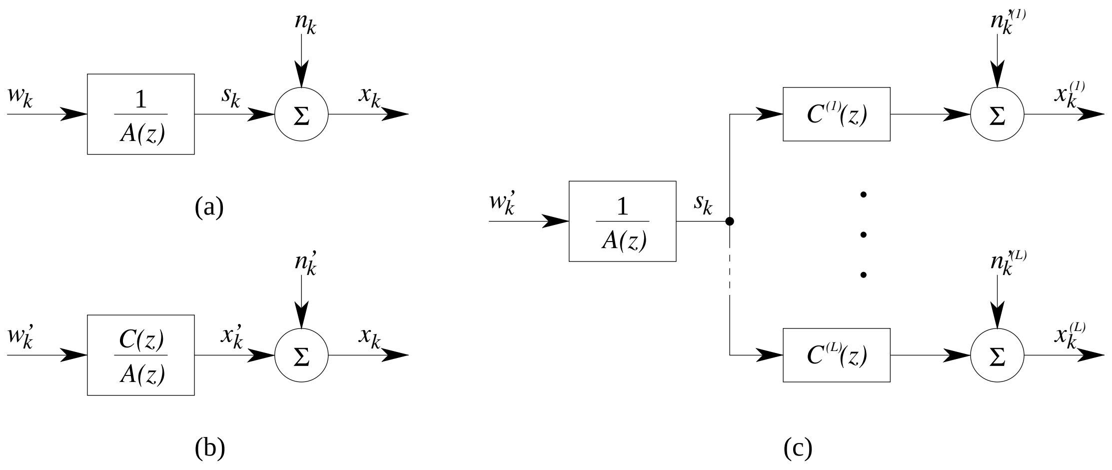


Figure 5.1 (a) AR speech model with additive white noise. (b) ARMA model for (a) with additive model noise. (c) Multichannel ARMA model for (a).

5.5.1 Subspace Based Identification of Multichannel FIR Filters

A simple blind identification method of multichannel FIR filters, based on the concept of *noise subspace*, is proposed in [84]. The input signal is allowed to be local stationary and the channel transfer functions may be nonminimum phase, hence the proposed algorithm is particularly suitable for the speech enhancement application in the previous section.

The block diagram of the unknown system model is shown in Figure 5.1(c). It is assumed that the number of channels $L \geq 2$. All channels are fed by the same unknown speech signal s_k which needs to be recovered from the observations $x_k^{(i)}$. From (5.26), the signal $x_k^{(i)}$ received from the ith channel is modelled as

$$x_k^{(i)} = c_k^{(i)} * s_k + n_k'^{(i)} \tag{5.27}$$

where each sequence $x_k^{(i)}$ depends on a different unknown impulse response $\mathbf{c}^{(i)}$ of order p characterizing the ith channel

$$C^{(i)}(z) = \sum_{k=0}^{p} c_k^{(i)} z^{-k} \quad \Rightarrow \quad \mathbf{c}^{(i)} = \begin{pmatrix} c_0^{(i)} & c_1^{(i)} & \cdots & c_p^{(i)} \end{pmatrix}^T \tag{5.28}$$

For a sequence of m successive samples $\mathbf{x}^{(i)} = (x_0^{(i)} x_1^{(i)} \cdots x_{m-1}^{(i)})^T$, Equation (5.27) is written as

$$\mathbf{x}^{(i)} = \mathbf{C}^{(i)}\mathbf{s} + \mathbf{n}'^{(i)} \tag{5.29}$$

where $\mathbf{s} = (s_0 s_1 \cdots s_{m+p-1})^T$, $\mathbf{n}'^{(i)} = (n_0'^{(i)} n_1'^{(i)} \cdots n_{m-1}'^{(i)})^T$ and $\mathbf{C}^{(i)} \in \mathbb{R}^{m\times(m+p)}$ is the *filtering matrix* associated with the linear filter $\mathbf{c}^{(i)}$, and defined as

$$\mathbf{C}^{(i)} = \begin{pmatrix} c_0^{(i)} & \cdots & c_p^{(i)} & & \mathbf{0} \\ & \ddots & & \ddots & \\ \mathbf{0} & & c_0^{(i)} & \cdots & c_p^{(i)} \end{pmatrix} \tag{5.30}$$

For L channels, the set of measurements depending on the same set of speech samples is given by a composite vector $\mathbf{x} \in \mathbb{R}^{Lm\times 1}$

$$\begin{pmatrix} \mathbf{x}^{(0)} \\ \vdots \\ \mathbf{x}^{(L-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{C}^{(0)} \\ \vdots \\ \mathbf{C}^{(L-1)} \end{pmatrix} \mathbf{s} + \begin{pmatrix} \mathbf{n}'^{(0)} \\ \vdots \\ \mathbf{n}'^{(L-1)} \end{pmatrix} \tag{5.31}$$

or

$$\mathbf{x} = \mathbf{C}\mathbf{s} + \mathbf{n}' \tag{5.32}$$

This linear system has dimension $Lm \times (m+p)$, and assuming $m \geq p$, then the system becomes more and more overdetermined as m increases.

A blind identification procedure consists in estimating the vector $\mathbf{c} \in \mathbb{R}^{L(p+1)\times 1}$ of channel coefficients

$$\mathbf{c} = \begin{pmatrix} \mathbf{c}^{(0)} \\ \vdots \\ \mathbf{c}^{(L-1)} \end{pmatrix} \tag{5.33}$$

from the composite measurement vector $\mathbf{x}$ or equivalently from its correlation matrix $\mathbf{R}_x \in \mathbb{R}^{Lm\times Lm}$

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\} \tag{5.34}$$

Since the additive model noise $\mathbf{n}'$ is assumed to be independent of the speech sequence, $\mathbf{R}_x$ can be expressed by use of (5.32) as

$$\mathbf{R}_x = \mathbf{C}\mathbf{R}_s\mathbf{C}^T + \mathbf{R}_{n'} \tag{5.35}$$

where $\mathbf{R}_{n'} = E\{\mathbf{n}'\mathbf{n}'^T\} = \sigma_{n'}^2\mathbf{I} \in \mathbb{R}^{Lm\times Lm}$ and $\mathbf{R}_s = E\{\mathbf{s}\mathbf{s}^T\} \in \mathbb{R}^{(m+p)\times(m+p)}$ is assumed to have full rank but otherwise unknown. The rank of the composite filter matrix $\mathbf{C}$ is given by the following theorem [113, 114, 84]

THEOREM 5.1 (*Rank of the Composite Filter Matrix) The composite filtering matrix denoted* $\mathbf{C} \in \mathbb{R}^{Lm\times(m+p)}$ *is full rank, i.e., rank* $(\mathbf{C}) = m + p$, *if*

1. *The channel transfer functions* $C^{(i)}(z) = \sum_{k=0}^{p} c_k^{(i)} z^{-k}$ *have no common zero.*
2. $m \geq p$.
3. *At least one polynomial* $C^{(i)}(z)$ *has degree* p.

Thus, since $\mathbf{R}_s$ has full rank, the signal part $\mathbf{C}\mathbf{R}_s\mathbf{C}^T$ of the correlation matrix $\mathbf{R}_x$ has rank $(m+p)$. The eigendecomposition of $\mathbf{R}_x$ can therefore be expressed as

$$\mathbf{R}_x = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_1^2 & \mathbf{0} \\ \mathbf{0} & \sigma_{n'}^2\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{pmatrix} \tag{5.36}$$

where $\mathbf{\Sigma}_1^2 \in \mathbb{R}^{(m+p)\times(m+p)}$. The columns of $\mathbf{U}_1 \in \mathbb{R}^{Lm\times(m+p)}$ span the *signal subspace*, while the columns of $\mathbf{U}_2 \in \mathbb{R}^{Lm\times(Lm-m-p)}$ span its orthogonal complement, the *noise subspace*. The signal subspace is also the linear space spanned by the columns of the composite filtering matrix $\mathbf{C}$. Hence, the columns of $\mathbf{C}$ are orthogonal to any vector in the noise subspace, i.e.,

$$\mathbf{u}_j^T\mathbf{C} = \sum_{i=0}^{L-1} \bar{\mathbf{u}}_j^{(i)T}\mathbf{C}^{(i)} = \mathbf{0}\,, \qquad j = m+p+1, \ldots, Lm \tag{5.37}$$

where the eigenvector $\mathbf{u}_j = (\bar{\mathbf{u}}_j^{(0)T} \cdots \bar{\mathbf{u}}_j^{(L-1)T})^T \in \mathbb{R}^{Lm}$ is partitioned into L blocks of length m as in (5.31). Equation (5.37) is the convolution sum of the vectors $\bar{\mathbf{u}}_j^{(i)}$ with the channel impulse responses $\mathbf{c}^{(i)}$. Since the convolution operation is commutative, (5.37) can also be expressed in terms of the vector $\mathbf{c}$ of channel coefficients and the filter matrices associated with the segments of the eigenvector $\mathbf{u}_j$ defined as

$$\bar{\mathbf{U}}_j = \begin{pmatrix} \bar{\mathbf{U}}_j^{(0)} \\ \vdots \\ \bar{\mathbf{U}}_j^{(L-1)} \end{pmatrix}, \qquad \bar{\mathbf{U}}_j^{(i)} = \begin{pmatrix} \bar{u}_{j0}^{(i)} & \cdots & \bar{u}_{j(m-1)}^{(i)} & & \mathbf{0} \\ & \ddots & & \ddots & \\ \mathbf{0} & & \bar{u}_{j0}^{(i)} & \cdots & \bar{u}_{j(m-1)}^{(i)} \end{pmatrix} \tag{5.38}$$

where $\bar{\mathbf{U}}_j \in \mathbb{R}^{L(p+1)\times(m+p)}$ and $\bar{\mathbf{U}}_j^{(i)} \in \mathbb{R}^{(p+1)\times(m+p)}$. Thus, an alternative formulation of Equation (5.37) is

$$\mathbf{c}^T\bar{\mathbf{U}}_j = \sum_{i=0}^{L-1} \mathbf{c}^{(i)T}\bar{\mathbf{U}}_j^{(i)} = \mathbf{0}\,, \qquad j = m+p+1, \ldots, Lm \tag{5.39}$$

In [84] is shown that the noise subspace $\langle \mathbf{U}_2 \rangle$ used in (5.39) *uniquely determines* the channel coefficients $\mathbf{c}$ up to a multiplicative constant, i.e.,

THEOREM 5.2 (*Uniqueness of the Composite Filter Matrix) Assume that*

1. *$m > p$.*
2. *The filter matrix corresponding to $m-1$ samples is full rank, i.e.,* $\mathrm{rank}(\mathbf{C}_{m-1}) = m+p-1$.

and let $\mathbf{c}' \in \mathbb{R}^{L(p+1)\times 1}$ be another nonzero filter vector with a corresponding filtering matrix $\mathbf{C}' \in \mathbb{R}^{Lp\times(m+p)}$. Then

$$\mathrm{range}\,(\mathbf{C}') \subset \mathrm{range}\,(\mathbf{C}) \tag{5.40}$$

iff

$$\mathbf{C} = \alpha\mathbf{C}'\,, \qquad \alpha \in \mathbb{R} \ \ \textit{and} \ \ \alpha \neq 0 \tag{5.41}$$

Hence, both matrices $\mathbf{C}$ and $\mathbf{C}'$ share the same column space if and only if they are proportional.

Theorem 5.2 is fundamental for the blind identification method, since it provides a mechanism for estimating the channel coefficients even in the cases where $\mathbf{R}_s$ is unknown, provided it has full rank. It relies on the specific structure of $\mathbf{C}$, which is by construction a filtering matrix.

The orthogonality condition (5.39) is *linear* in the unknown coefficients $\mathbf{c}$, which simplifies the identification procedure. In practice, only the sample correlation matrix $\hat{\mathbf{R}}_x$ is available

$$\hat{\mathbf{R}}_x = \frac{1}{n}\mathbf{X}\mathbf{X}^T \tag{5.42}$$

where the composite data matrix $\mathbf{X} \in \mathbb{R}^{Lm\times n}$ is given by

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}^{(0)} \\ \vdots \\ \mathbf{X}^{(L-1)} \end{pmatrix} \tag{5.43}$$

and the Toeplitz structured data matrix for the ith channel $\mathbf{X}^{(i)} \in \mathbb{R}^{m\times n}$ is defined by (3.8) with $n \gg m$. An estimate of the noise subspace $\hat{\mathbf{U}}_2$ are found from the SVD of $\mathbf{X}$

$$\frac{1}{\sqrt{n}}\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^T = \begin{pmatrix} \hat{\mathbf{U}}_1 & \hat{\mathbf{U}}_2 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{\Sigma}}_1^2 & \mathbf{0} \\ \mathbf{0} & \hat{\sigma}_{n'}^2 \mathbf{I} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{V}}_1^T \\ \hat{\mathbf{V}}_2^T \end{pmatrix} \tag{5.44}$$

Then, Equation (5.39) is solved in the least squares sense, which leads to the minimization of the following quadratic form

$$\begin{aligned} e(\mathbf{c}) &= \sum_{j=m+p+1}^{Lm} \|\mathbf{c}^T \hat{\hat{\mathbf{U}}}_j\|_2^2 \\ &= \sum_{j=m+p+1}^{Lm} \mathbf{c}^T \hat{\hat{\mathbf{U}}}_j \hat{\hat{\mathbf{U}}}_j^T \mathbf{c} \\ &= \mathbf{c}^T \left(\sum_{j=m+p+1}^{Lm} \hat{\hat{\mathbf{U}}}_j \hat{\hat{\mathbf{U}}}_j^T \right) \mathbf{c} \\ &= \mathbf{c}^T \mathbf{Q} \mathbf{c} \end{aligned} \tag{5.45}$$

By Theorem 5.2, if the quadratic form is constructed from the *true* correlation matrix $\mathbf{R}_x$, the true channel coefficients are the unique (up to a scalar factor) vector $\mathbf{c}$ such that $e(\mathbf{c}) = 0$, and the matrix $\mathbf{Q} \in \mathbb{R}^{L(p+1)\times L(p+1)}$ must therefore satisfy

$$\text{rank}\,(\mathbf{Q}) = L(p+1) - 1 \tag{5.46}$$

When only an *estimate* of the correlation matrix $\hat{\mathbf{R}}_x$ is available, the matrix $\mathbf{Q}$ has not exactly rank $L(p+1)-1$. Hence, estimates of $\mathbf{c}$ can be obtained by minimizing $e(\mathbf{c})$ subject to a properly chosen constraint avoiding the trivial solution $\mathbf{c} = \mathbf{0}$. A natural choice is minimization subject to quadratic constraint

$$\min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}} \mathbf{c}^T \mathbf{Q} \mathbf{c} \quad \text{subject to} \quad \|\mathbf{c}\|_2 = 1 \tag{5.47}$$

The *solution* $\hat{\mathbf{c}}$ is the unit-norm eigenvector associated to the smallest eigenvalue of the matrix $\mathbf{Q}$.

As pointed out in [84], the above algorithm is a reminiscent of the MUSIC method [103] for DOA estimation of narrowband sources. It exploits the orthogonality between the noise subspace and the column space of $\mathbf{C}$.

5.5.2 Multichannel Inverse Filtering

From the estimated channel coefficients $\mathbf{c}$, an inverse filter may be obtained. The method considered here, is the linear matrix equalizer $\mathbf{C}^+ \in \mathbb{R}^{(m+p)\times Lm}$ defined as the pseudoinverse of the composite filtering matrix $\mathbf{C}$

$$\mathbf{C}^+ = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{5.48}$$

The matrix $\mathbf{C}^+$ may be computed from the SVD of $\mathbf{C} = \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^T$

$$\mathbf{C}^+ = \mathbf{V}_c \mathbf{\Sigma}_c^{-1} \mathbf{U}_c^T \tag{5.49}$$

Each row of the matrix $\mathbf{C}^+$ is a linear scalar equalizer of length Lm, i.e., each channel is filteret by an m-tap filter and the L resulting outputs are added. The following remarks can be made to expression (5.48):

Equalizer. Each row of $\mathbf{C}^+$ provides an equalizer with various delays and noise enhancement properties.

Finite length equalization. In the absence of noise, (5.49) is a zero-forcing equalizer of *finite* length $m > p$. This property does not hold in the classical monochannel case, where the channel inverse have infinite length [4].

No reconstruction delay. Each row of the pseudoinverse is made from the coefficients of a multichannel equalizer able to reconstruct the data at time k from the previous observed samples only. The multichannel filter is *minimum phase*.

Applying the multichannel inverse filter $\mathbf{C}^+$ to the composite data vector $\mathbf{x}$ (5.32), gives the enhanced signal $\hat{\mathbf{x}}$

$$\hat{\mathbf{x}} = \mathbf{C}^+\mathbf{x} = \mathbf{s} + \mathbf{C}^+\mathbf{n}' \tag{5.50}$$

5.6 Summary

In this Chapter, model based speech enhancement has been considered. Most methods are based on Wiener filtering of the noisy signal, where the parameters of the speech model, used in the gain function, are obtained by iterative techniques. However, in order to obtain reliable parameters, speech-specific constraints must be applied to the problem. Another problem is the musical noise, which will also be presented in the enhanced signal due to the Wiener filter approch.

Another approach is ARMA modelling of the noisy signal, followed by inverse filtering in the time domain, to remove the noise represented by the MA part. Also the ARMA modelling is normally iterative, however, the approach presented here, use a multi-microphone technique to identify the MA parts. The concept has initially been applied to noisy AR-signals, however, the problem is to obtain a rank deficient $\mathbf{Q}$ matrix, i.e., further work has to be done in this area.

Chapter 6

Rank-Revealing Orthogonal Decompositions

The Singular Value Decomposition is the basic tool in the discussed noise reduction systems. One of its major merits is that it provides the rank of the matrix and a basis for four important spaces. Unfortunately, the SVD is computationally expensive and resist updating. The initial cost of computing a SVD would not be an objection, if the decomposition could be cheaply updated. However, all known updating schemes require $\mathcal{O}(n^3)$ flops for a matrix of order n even for a simple update such as adding a new row [15, 46, 80]. The problem with the SVD is that it is essentially unique. The requirement that $\mathbf{\Sigma}$ in the SVD be diagonal effectively determines $\mathbf{U}$ and $\mathbf{V}$. To circumvent these drawbacks any decomposition/algorithm that estimates the rank and the orthogonal spaces can be used in the place of the SVD. The following issues are critical for the application:

- Recomputing a factorization is too costly, and an update algorithm should be performed with as few operations and as little storage requirement as possible. This, e.g., important in real time applications.
- The outcome should be accurate up to the limitations of the data and conditioning of the problem, i.e., a stable numerical method must be used.
- It should be possible to use a computer with short word-length. This rules out the correlation based methods, which requires twice the word-length as methods based on the data matrix.

The Rank-Revealing QR (RRQR) decomposition could be one possible choice [18]. RRQR algorithms can track the numerical rank, and the computational complexity of this approach is $\mathcal{O}(n^2)$. However, it has been shown by Fierro and Bunch [35] that the quality of the approximation depends on the gap between the singular values, and approximations to the right singular subspaces are not directly available. Furthermore, the decomposition is not easy to update.

A better choice is the Rank-Revealing URV/ULV Decompositions introduced by Hanson and Lawson in [55] (URV version) and the related algorithms devised by Stewart [107, 110, 1, 43], which use two-sided orthogonal transformations to upper/lower triangularize the matrix. Thus, the URV/ULV decompositions are a compromise between the SVD and the RRQR decomposition with some of the virtues of both. Actually, these two decompositions are special cases of the triangular decompositions. The rank-revealing URV/ULV decompositions are effective in tracking the numerical rank and approximations to both the left and right subspaces are explicitly available. Furthermore, the quality of the subspace approximations are not sensitive to the magnitude of the gap in the singular values, and the decompositions can be updated in $\mathcal{O}(n^2)$ flops.

Another issue is the precision in the estimated row and column subspaces, which is not identical for the upper and lower triangular version. Therefore the ULV decomposition is chosen in our case, where only right subspaces are maintained. For a pair of matrices, the ULLV decomposition [73, 93, 67, 10] will be used, which is an approximation of the QSVD, and is also an extension of the ULV decomposition.

In this Chapter, the rank-revealing ULV/ULLV decompositions are presented. The quality of the subspaces are compared with the one produced by the SVD, and perturbation theory is considered. In both analyzes, the conclusions are drawn from speech related examples.

An ULV/ULLV formulation of the LS, MV and TDC estimation strategies is given, where the last two are suggested here. Thus, a *recursive updating* of the estimates are used instead of working in frames. Finally, implementation aspects are discussed.

6.1 The ULV Decomposition

The ULV decomposition of a matrix is a product of three matrices defined by the following theorem.

THEOREM 6.1 (*ULV Decomposition (ULVD)*) *Given a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, and assume $m \geq n$, then there exist a matrix $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ with orthogonal columns and a orthogonal matrix $\mathbf{V}_X \in \mathbb{R}^{n\times n}$ such that*

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{V}_X^T \tag{6.1}$$

where $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ is lower triangular.

A basic feature is that the ULV decomposition of a full rank matrix $\mathbf{X}$ can be made rank revealing, if there is a gap in the singular values, i.e., it can be used to break the space spanned by the matrix $\mathbf{X}$ into two subspaces, one corresponding to the cluster of largest singular values and the other corresponding to the group of smallest singular values. The rank-revealing version of the ULV decomposition was first introduced by Stewart [107] as

THEOREM 6.2 (*Rank-Revealing ULV Decomposition (RRULVD)*) *Assume that $m \geq n$ and that $\mathbf{X} \in \mathbb{R}^{m\times n}$ has numerical rank $p < n$ corresponding to a given tolerance τ, i.e., its singular values satisfy*

$$\sigma_1 \geq \cdots \geq \sigma_p \geq \tau \gg \sigma_{p+1} \geq \cdots \geq \sigma_n \tag{6.2}$$

Then there exist a matrix $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ with orthogonal columns and a orthogonal matrix $\mathbf{V}_X \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{V}_X^T = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \tag{6.3}$$

where $\mathbf{L}_X \in \mathbb{R}^{n\times n}$, $\mathbf{L}_{X1} \in \mathbb{R}^{p\times p}$ and $\mathbf{G}_X \in \mathbb{R}^{(n-p)\times(n-p)}$ are lower triangular, and

$$\sigma_{min}(\mathbf{L}_{X1}) \approx \sigma_p \tag{6.4}$$

$$\|\mathbf{F}_X\|_F^2 + \|\mathbf{G}_X\|_F^2 \approx \sigma_{p+1}^2 + \cdots + \sigma_n^2 \tag{6.5}$$

The tolerance τ is defined based on a detection threshold in the underlying signal processing problem. Thus, when $\mathbf{X}$ is the sum of a rank deficient signal matrix $\mathbf{S}$ and a full rank noise matrix $\mathbf{N}$, then the rank-revealing ULV decomposition can be used to estimate the signal and noise-only subspace.

For a rank deficient matrix, the ideal form of the rank-revealing ULV decomposition, i.e., $\mathbf{F} = \mathbf{G} = \mathbf{0}$, is called a *Complete Orthogonal Decomposition*

THEOREM 6.3 (*Complete Orthogonal Decomposition) Given a rank-p matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$, where $m \geq n > p$, then there exist orthogonal matrices $\mathbf{U}_S \in \mathbb{R}^{m\times m}$ and $\mathbf{V}_S \in \mathbb{R}^{n\times n}$ such that*

$$\mathbf{S} = \mathbf{U}_S \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}_S^T \tag{6.6}$$

where $\mathbf{L}_{S1} \in \mathbb{R}^{p\times p}$ is lower triangular.

The Complete Orthogonal Decomposition, like the SVD, also provides orthogonal bases for the row and null spaces of $\mathbf{S}$ and $\mathbf{S}^T$, but unlike the SVD neither the Complete Orthogonal Decomposition or the RRULVD of a given matrix is unique. Actually, the SVD is itself a Complete Orthogonal Decomposition/RRULVD.

6.1.1 Quality of Subspaces

In [75, 35], the approximate subspaces produced by the rank-revealing ULV decomposition as in (6.3) are compared with the subspaces produced by the SVD. How the singular values of $\mathbf{G}_X$ and $\mathbf{L}_{X1}$ are related to those of $\mathbf{X}$ is also considered. These results prove that the RRULVD not only reveal the numerical rank of the matrix but also provide good approximations to the singular subspaces as part of the factorization.

Assume that there is a sufficient gap in the singular values of the diagonal blocks of the matrix $\mathbf{L}_X$

$$\rho = \frac{\sigma_{max}(\mathbf{G}_X)}{\sigma_{min}(\mathbf{L}_{X1})} < 1 \tag{6.7}$$

then the relation in the singular values is bounded by [75]

THEOREM 6.4 (*Singular Value Bounds for the RRULVD) Let a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ have the RRULVD in (6.3) and let ρ be defined by (6.7), then*

$$1 \geq \frac{\sigma_{p+j}(\mathbf{L}_X)}{\sigma_j(\mathbf{G}_X)} \geq \sqrt{1 - \frac{\|\mathbf{F}_X\|_2^2}{(1-\rho^2)\sigma_{min}^2(\mathbf{L}_{X1})}}, \qquad j = 1, \ldots, n-p \tag{6.8}$$

$$1 \geq \frac{\sigma_j(\mathbf{L}_{X1})}{\sigma_j(\mathbf{L}_X)} \geq \sqrt{1 - \frac{\|\mathbf{F}_X\|_2^2}{(1-\rho^2)\sigma_{min}^2(\mathbf{L}_{X1})}}, \qquad j = 1, \ldots, p \tag{6.9}$$

Thus, an off-diagonal perturbation like neglecting $\mathbf{F}_X$ will only give a relative error $\mathcal{O}(\|\mathbf{F}_X\|_2)$ in the singular values, so even the smallest singular values retain their accuracy. It can also be seen from Theorem 6.4 that the singular values of $\mathbf{L}_{X1}$ will in general underestimate the first p values $\sigma_i(\mathbf{X})$ (this fact also follows from the minimax characterization of singular values). Similarily, the singular values of $\mathbf{G}_X$ will in general overestimate the corresponding last $n-p$ values $\sigma_i(\mathbf{X})$.

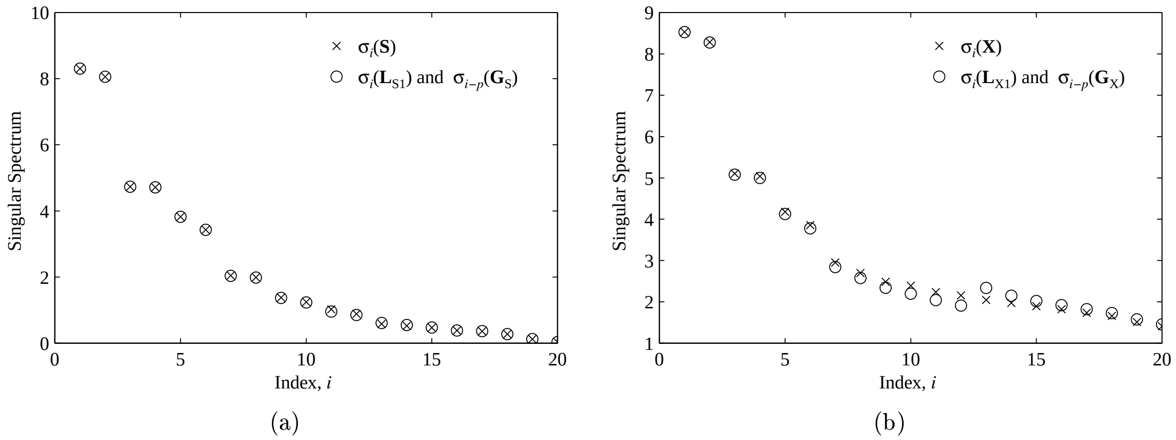


Figure 6.1 (a) The singular values of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing the voiced speech frame with 160 samples, and the corresponding values obtained from $\mathbf{L}_{S1}$ and $\mathbf{G}_S$ with $p = 12$ (Initial RRULV algorithm without refinement). (b) The average singular values of the corresponding noisy data matrix using 100 independent white noise realizations and SNR=5dB.

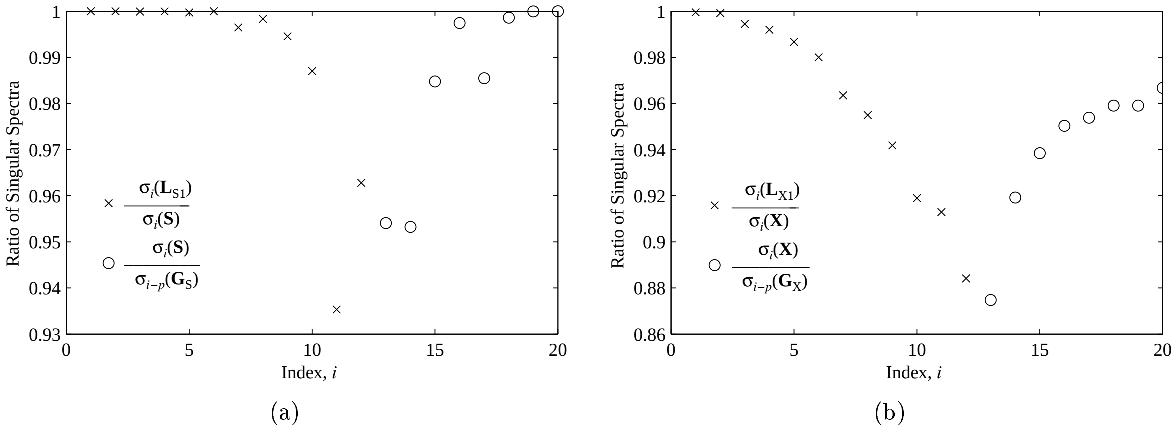


Figure 6.2 The ratios in Theorem 6.4, where (a) corresponds to Figure 6.1(a) and (b) to Figure 6.1(b).

The graphs in Figure 6.1(a) and 6.2(a) illustrates this problem. Here, the singular values of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing the voiced speech frame with 160 samples are compared with the one obtained from $\mathbf{L}_{S1}$ and $\mathbf{G}_S$ with $p = 12$. Note, that $\sigma_i(\mathbf{L}_{S1})$ are plotted against the first p indices, and $\sigma_i(\mathbf{G}_S)$ are plotted from index $p+1$ to n. It is seen that the largest and smallest singular values and thereby the dominant range and null space are well determined by the RRULVD, while the subspaces diverge a little near the rank-revealing point p due to the off-diagonal block $\mathbf{F}_X$. The gap ρ is 0.72 and the lower bounds in Theorem 6.4 are 0.76.

The under/overestimation becomes more pronounced when noise is added as shown in Figure 6.1(b) and 6.2(b), and Theorem 6.4 is obviously no longer valid since $\rho > 1$, i.e., the subspaces are now blurred together. The reduced performance in the noisy case can be explained by the smaller gap in the singular spectrum of $\mathbf{X}$, and the reason is the gap-sensitive condition estimator used in the rank-revealing deflation algorithm (see Section 7.5.2), i.e., $\|\mathbf{F}_X\|_2$ has increased. In the above example, a relative low segmental SNR (5 dB) is used and the RRULV algorithm is applied without refinement (see Chapter 7) to obtain significant errors.

The following theorem due to Mathias and Stewart [75] and Fierro and Bunch [35] shows that as the off-diagonal block $\mathbf{F}_X$ decreases, the RRULVD subspaces converge to their SVD counterparts, where the quality of the approximate subspaces is measured by the canonical angles.

THEOREM 6.5 (*Subspace Bounds for the RRULVD*) *Let* $\mathbf{\Theta}(\langle \mathbf{V}_{X2,SVD}\rangle, \langle \mathbf{V}_{X2}\rangle) \in \mathbb{R}^{(n-p)\times(n-p)}$ *be the diagonal matrix of canonical angles between the right singular subspace* $\langle \mathbf{V}_{X2,SVD}\rangle$ *corresponding to the smallest* $n-p$ *singular values of* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *and the approximate subspace* $\langle \mathbf{V}_{X2}\rangle$ *produced by the rank-revealing ULV decomposition of* $\mathbf{X}$ *(6.3), and let* $\mathbf{\Phi}(\langle \mathbf{U}_{X2,SVD}\rangle, \langle \mathbf{U}_{X2}\rangle) \in \mathbb{R}^{(n-p)\times(n-p)}$ *be the diagonal matrix of canonical angles for the case of left supspaces* $\langle \mathbf{U}_{X2,SVD}\rangle$ *and* $\langle \mathbf{U}_{X2}\rangle$*, then under Assumption (6.7)*

$$\|\sin \mathbf{\Theta}(\langle \mathbf{V}_{X2,SVD}\rangle, \langle \mathbf{V}_{X2}\rangle)\|_2 \le \frac{\rho\|\mathbf{F}_X\|_2}{(1-\rho^2)\sigma_{min}(\mathbf{L}_{X1})} \tag{6.10}$$

$$\frac{\|\mathbf{F}_X\|_2}{\|\mathbf{L}_{X1}\|_2 + \|\mathbf{G}_X\|_2} \le \|\sin \mathbf{\Phi}(\langle \mathbf{U}_{X2,SVD}\rangle, \langle \mathbf{U}_{X2}\rangle)\|_2 \le \frac{\|\mathbf{F}_X\|_2}{(1-\rho^2)\sigma_{min}(\mathbf{L}_{X1})} \tag{6.11}$$

Clearly, the quality of the approximate subspaces also depends on the off-diagonal block $\|\mathbf{F}_X\|_2$ and not really on the gap ρ in the singular spectrum, which implies that the matrix $\mathbf{F}_X$ should have small elements in absolute value.

The largest canonical angle is also related to the notion of distance between equidimensional subspaces. From Definition 3.2 on Page 33 follows

$$\begin{aligned} \mathrm{dist}(\langle \mathbf{V}_{X1,SVD}\rangle, \langle \mathbf{V}_{X1}\rangle) &= \mathrm{dist}(\langle \mathbf{V}_{X2,SVD}\rangle, \langle \mathbf{V}_{X2}\rangle) \\ &= \|\sin \mathbf{\Theta}(\langle \mathbf{V}_{X2,SVD}\rangle, \langle \mathbf{V}_{X2}\rangle)\|_2 \end{aligned} \tag{6.12}$$

and from that definition as well as [35, lemma 2.1]

$$\begin{aligned} \mathrm{dist}(\langle \mathbf{U}_{X1,SVD}\rangle, \langle \mathbf{U}_{X1}\rangle) &= \mathrm{dist}(\langle \mathbf{U}_{X2,SVD}\rangle, \langle \mathbf{U}_{X2}\rangle) \\ &= \|\sin \mathbf{\Phi}(\langle \mathbf{U}_{X2,SVD}\rangle, \langle \mathbf{U}_{X2}\rangle)\|_2 \end{aligned} \tag{6.13}$$

By comparing (6.10) and (6.11), it is seen that the approximation of the right singular subspaces are more exact than the approximation of the left singular subspaces. This result can be cast as more conventional bounds

$$\|\mathbf{X}\mathbf{V}_{X1}\|_2 = \left\| \begin{pmatrix} \mathbf{L}_{X1} \\ \mathbf{F}_X \end{pmatrix} \right\|_2 \quad \ge \quad \|\mathbf{U}_{X1}^T\mathbf{X}\|_2 = \|\mathbf{L}_{X1}\|_2 \tag{6.14}$$

$$\|\mathbf{X}\mathbf{V}_{X2}\|_2 = \|\mathbf{G}_X\|_2 \quad \le \quad \|\mathbf{U}_{X2}^T\mathbf{X}\|_2 = \left\| \begin{pmatrix} \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \right\|_2 \tag{6.15}$$

Therefore, the ULV decomposition is chosen for applications in which an approximate right singular subspace is needed or maintained, and the related URV decomposition is used in cases where the left singular subspaces are desired. Remember that the URV decomposition is an upper triangular version of the ULV decomposition.

In Figure 6.3, the canonical angles corresponding to the example in Figure 6.1 are plotted against their indices. For the clean speech frame, the signal subspace is well determined with only a few large angles ($\sin\theta_{max} = 0.28$). The gap ρ is again 0.72 and the upper bound in Theorem 6.5 is 0.67. In the noisy case, the reduced gap in the singular spectrum of $\mathbf{X}$ generally increases all the angles because of the increased norm of the off-diagonal block $\mathbf{F}_X$ as discussed

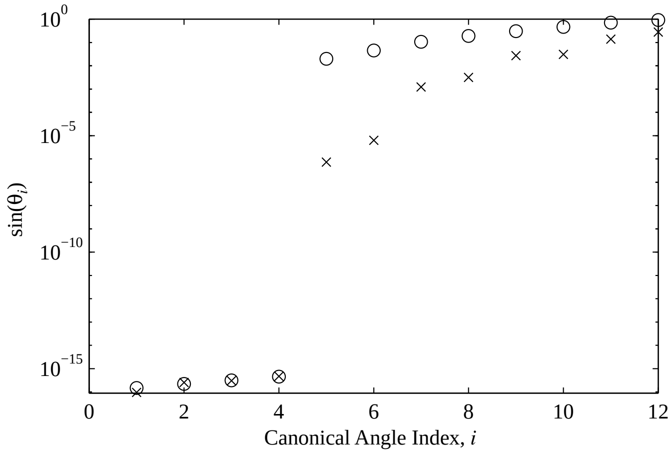


Figure 6.3 $(\times)$ $\sin\boldsymbol{\Theta}(\langle\mathbf{V}_{S1,SVD}\rangle, \langle\mathbf{V}_{S1,ULV}\rangle)$, where the $p = 12$ dimensional subspaces are obtained from $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples. $(\circ)$ The average $\sin\boldsymbol{\Theta}$ of the corresponding noisy data matrix using 100 independent white noise realizations and SNR=5dB.

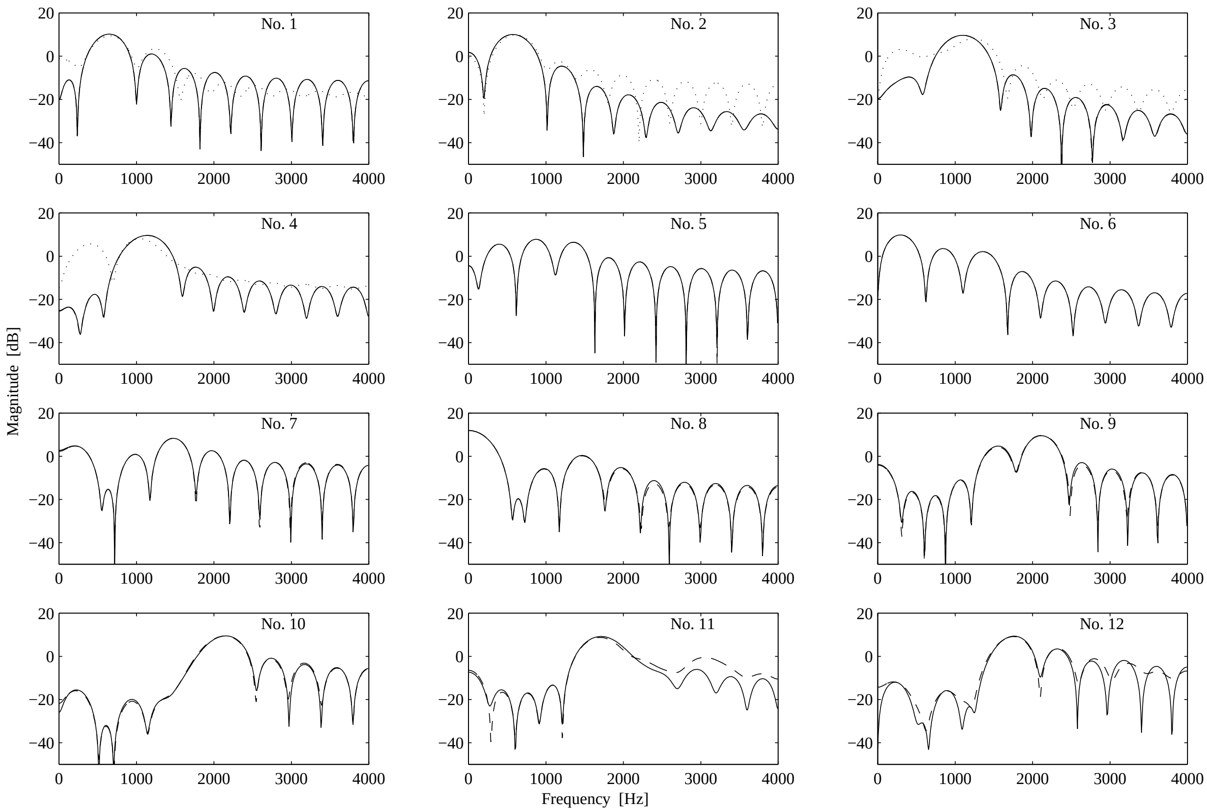


Figure 6.4 (Solid) The first 12 eigenfilters $\mathbf{V}_{S1,SVD}$ of $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples. (Dashed) The corresponding eigenfilters obtained from the RRULVD based rank-12 approximation. (Dotted) The canonical vectors (filters) associated with $\mathbf{V}_{S1,SVD}$.

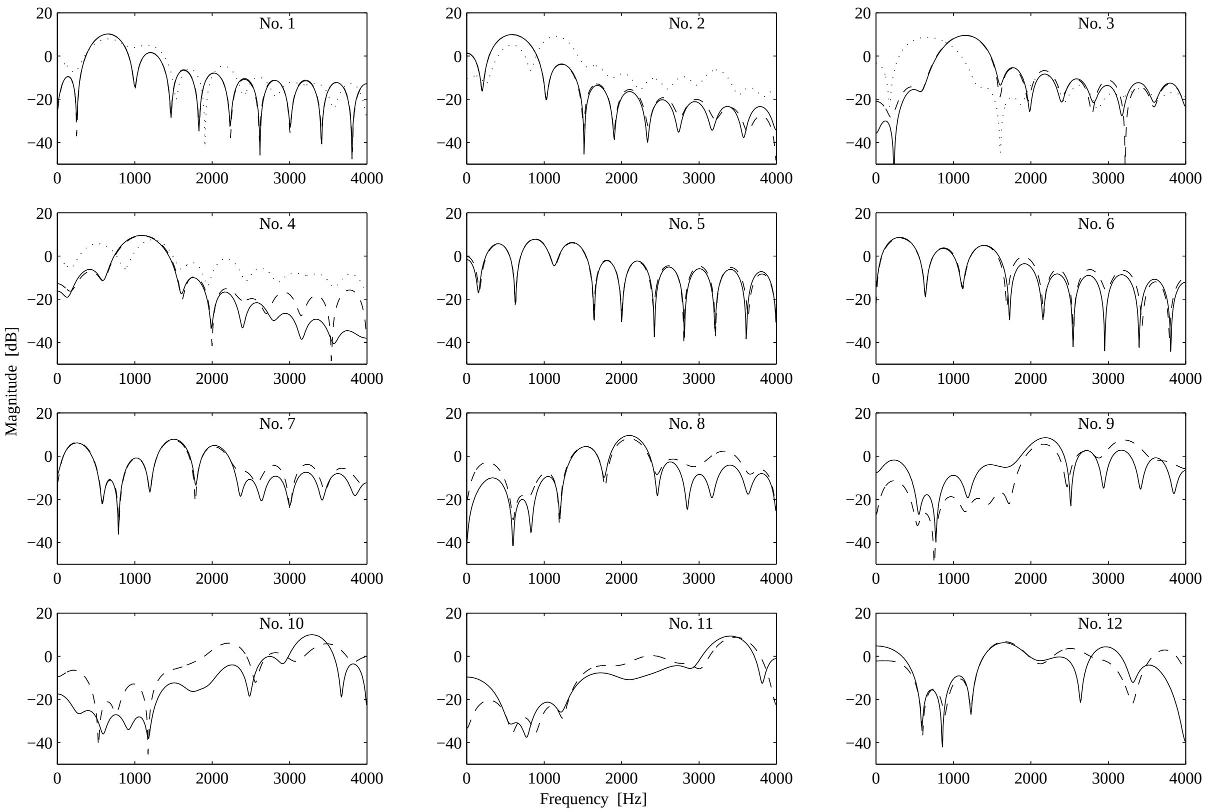


Figure 6.5 (Solid) The first 12 eigenfilters $\mathbf{V}_{X1,SVD}$ of $\mathbf{X} \in \mathbb{R}^{141\times 20}$ representing a noisy voiced speech frame with 160 samples and SNR=5dB. (Dashed) The corresponding eigenfilters obtained from the RRULVD based rank-12 approximation. (Dotted) The canonical vectors (filters) associated with $\mathbf{V}_{X1,SVD}$.

previously, but the possible mix of signal and noise subspace also results in angles close to one ($\sin\theta_{max} = 0.92$). The four angles close to the machine precision arise from the fact that two p-dimensional subspaces in a n-dimensional space must have a $2p - n$ dimensional intersection.

Now, compare the first $p = 12$ eigenfilters $\mathbf{V}_{S1,SVD}$ of the signal-only matrix $\mathbf{S}$ with the corresponding eigenfilters obtained from the RRULVD based rank-p approximation (see Section 6.2.1). Both sets of filters are illustrated in Figure 6.4 and, as the above discussion indicates, they are very similar. On the first four plots in the figure, the canonical vectors (filters) associated with the subspace intersection are shown (without ordering), and clearly they match the dominant eigenfilters. In the noisy case, the difference between the two sets of eigenfilters will become more clear, especially among the last few filters, as shown in Figure 6.5, where a single white noise realization has been added (SNR = 5 dB).

6.1.2 Perturbation Theory

In the noise reduction application, we will have access to the noise perturbed data matrix $\mathbf{X} = \mathbf{S} + \mathbf{N}$, but is interested in the projection onto the signal subspace spanned by the pure signal matrix $\mathbf{S}$. In [34], perturbation bounds for the RRULVD are given, i.e., bounds on the angle between the subspaces obtained by performing a RRULVD on the noisy data matrix $\mathbf{X}$ as in (6.3) and on the signal matrix $\mathbf{S}$ as

$$\mathbf{S} = \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{F}_S & \mathbf{G}_S \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \tag{6.16}$$

First define the signal subspace related noise level

$$\epsilon = \max \left\{ \|\mathbf{N}\mathbf{V}_{X1,SVD}\|_2, \|\mathbf{U}_{X1,SVD}^T \mathbf{N}\|_2 \right\} \leq \|\mathbf{N}\|_2 \tag{6.17}$$

and asumme that

$$\hat{\rho} = \frac{\sigma_{max}(\mathbf{G}_S)}{\sigma_{min}(\mathbf{L}_{X1})} < 1 \tag{6.18}$$

then the following theorem due to Fierro [34] bounds the sensitivity of the RRULVD subspaces.

THEOREM 6.6 (*Perturbation Bounds for the RRULVD) Let* $\mathbf{S}$ *and* $\mathbf{X}$ *have the RRULVD given by Equation (6.16) and (6.3), respectively, and let* ϵ *and* $\hat{\rho}$ *be defined by Equation (6.17) and (6.18), then*

$$\|\sin \mathbf{\Theta}(\langle \mathbf{V}_{X1} \rangle, \langle \mathbf{V}_{S1} \rangle)\|_2 \leq \frac{\hat{\rho}(\|\mathbf{F}_X\|_2 + \|\mathbf{F}_S\|_2 + \epsilon) + \epsilon}{(1-\hat{\rho}^2)\sigma_{min}(\mathbf{L}_{X1})} \tag{6.19}$$

$$\|\sin \mathbf{\Phi}(\langle \mathbf{U}_{X1} \rangle, \langle \mathbf{U}_{S1} \rangle)\|_2 \leq \frac{\|\mathbf{F}_X\|_2 + \|\mathbf{F}_S\|_2 + \epsilon}{(1-\hat{\rho})\sigma_{min}(\mathbf{L}_{X1})} \tag{6.20}$$

However, to measure the quality of the subspaces isolated using the RRULVD on the noise perturbed data matrix, they must be compared with those obtained using the SVD on the clean signal matrix. As pointed out in [60], such bounds is a special case of Theorem 6.6, when the RRULVD of $\mathbf{S}$ is reduced to the SVD, i.e., $\mathbf{F}_S = \mathbf{0}$, $\mathbf{L}_{S1} = \mathbf{\Sigma}_{S1}$, $\mathbf{G}_S = \mathbf{\Sigma}_{S2}$ and

$$\bar{\rho} = \frac{\sigma_{S,p+1}}{\sigma_{min}(\mathbf{L}_{X1})} < 1 \tag{6.21}$$

This results in the following theorem

THEOREM 6.7 (*True Perturbation Bounds for the RRULVD) Let* $\mathbf{X}$ *have the RRULVD given by Equation (6.3) and* $\mathbf{S}$ *the corresponding SVD, and let* ϵ *and* $\bar{\rho}$ *be defined by Equation (6.17) and (6.21), then*

$$\|\sin \mathbf{\Theta}(\langle \mathbf{V}_{X1} \rangle, \langle \mathbf{V}_{S1,SVD} \rangle)\|_2 \leq \frac{\bar{\rho}(\|\mathbf{F}_X\|_2 + \epsilon) + \epsilon}{(1-\bar{\rho}^2)\sigma_{min}(\mathbf{L}_{X1})} \tag{6.22}$$

$$\|\sin \mathbf{\Phi}(\langle \mathbf{U}_{X1} \rangle, \langle \mathbf{U}_{S1,SVD} \rangle)\|_2 \leq \frac{\|\mathbf{F}_X\|_2 + \epsilon}{(1-\bar{\rho})\sigma_{min}(\mathbf{L}_{X1})} \tag{6.23}$$

Theorem 6.7 indicates that as the norm of the off-diagonal block $\mathbf{F}_X$ decreases, the error between the ULV subspace and the corresponding true SVD subspace is dominated by the magnitude of the noise in the data matrix.

For $\mathbf{F}_X = \mathbf{0}$, the first cluster of singular values is decoupled from the second, and the theorem reduces to the perturbation bounds for singular subspaces by Wedin

$$\|\sin \mathbf{\Theta}(\langle \mathbf{V}_{X1,SVD} \rangle, \langle \mathbf{V}_{S1,SVD} \rangle)\|_2 \text{ and} \tag{6.24}$$

$$\begin{aligned} \|\sin \mathbf{\Phi}(\langle \mathbf{U}_{X1,SVD} \rangle, \langle \mathbf{U}_{S1,SVD} \rangle)\|_2 &\leq \frac{\epsilon}{\sigma_{X,p} - \sigma_{S,p+1}} \\ &\leq \frac{\epsilon}{\sigma_{S,p} - \sigma_{S,p+1} - \|\mathbf{N}\|_2} \end{aligned} \tag{6.25}$$

where perturbation theory of singular values, i.e., $|\sigma_{X,p} - \sigma_{S,p}| \leq \|\mathbf{N}\|_2$, has been used (see Section 3.3.4). Thus, if the noise level $\|\mathbf{N}\|_2$ is equal to half the gap in the signal-only singular spectrum, then the angle between the subspaces is $\frac{\pi}{2}$. For higher noise levels, the theory is no longer valid due to the possible mix of signal and noise subspace.

For the speech enhancement application, the gap is typically much smaller than the noise level, i.e., $\sigma_{noise} \gg \sigma_{S,p} - \sigma_{S,p+1}$, so Theorem 6.7 can not be used directly as a quality measure of the RRULVD. However, in practice $\|\mathbf{F}_X\|_2$ is also much smaller than σ_{noise}, which makes Equation (6.25) a close approximation to Theorem 6.7, and it is therefore reasonably to assume that the RRULVD and the SVD will deliver similar results in this application.

Note, that if the elements of $\mathbf{N}$ and $\mathbf{S}$ are uncorrelated, the noise is white $\mathbf{N} = \sigma_{noise}\mathbf{I}$, and the signal-only matrix $\mathbf{S}$ is rank deficient with rank p, then Equation (6.25) becomes

$$\cos(\phi_p) = \sqrt{1 - \sin^2(\phi_p)} = \sqrt{1 - \frac{\sigma^2_{noise}}{\sigma^2_{X,p}}} = \frac{\sigma_{S,p}}{\sigma_{X,p}} \tag{6.26}$$

which is identical to Equation (3.65).

The canonical angles between the signal subspace $\langle \mathbf{V}_{S1,SVD} \rangle$ obtained from the SVD of the clean signal matrix $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing the voiced speech frame with 160 samples and the subspaces $\langle \mathbf{V}_{X1,SVD} \rangle$ and $\langle \mathbf{V}_{X1,ULV} \rangle$ obtained in the noisy case using both the SVD and the RRULVD are shown in Figure 6.6. The angles are averaged over 100 independent white noise realizations (SNR = 5 dB). As expected, the SVD based subspace $\langle \mathbf{V}_{X1,SVD} \rangle$ is in the average closer to the true one.

Another issue is to compare the eigenfilters obtained from the SVD of the clean signal with the noisy case, where both the SVD and the RRULVD fail to match the last few eigenfilters as shown in Figure 6.7. Since the filters are averaged over 100 independent white noise realizations, the reduced magnitudes indicate the variance on the eigenfilter estimates (see Section 3.5). *Obviously, both methods deliver similar results, so the RRULVD will be comparable with the SVD when used in speech enhancement applications.*

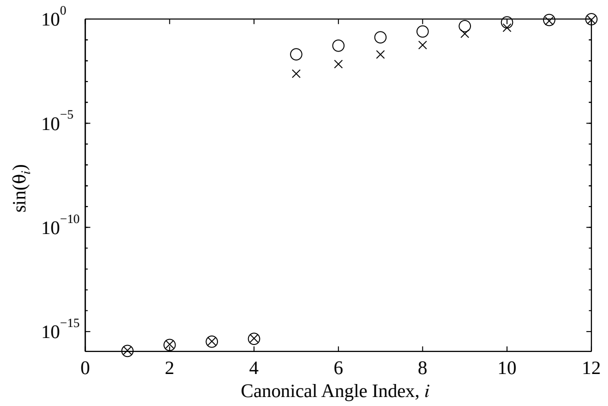


Figure 6.6 ($\times$) Average $\sin \mathbf{\Theta}(\langle \mathbf{V}_{S1,SVD} \rangle, \langle \mathbf{V}_{X1,SVD} \rangle)$, where the $p = 12$ dimensional subspace $\langle \mathbf{V}_{S1,SVD} \rangle$ is obtained from $\mathbf{S} \in \mathbb{R}^{141\times 20}$ representing a voiced speech frame with 160 samples, and $\langle \mathbf{V}_{X1,SVD} \rangle$ is obtained from the corresponding noisy data matrix using 100 independent white noise realizations and SNR=5dB. ($\circ$) The noisy subspace is now obtained from the RRULVD.

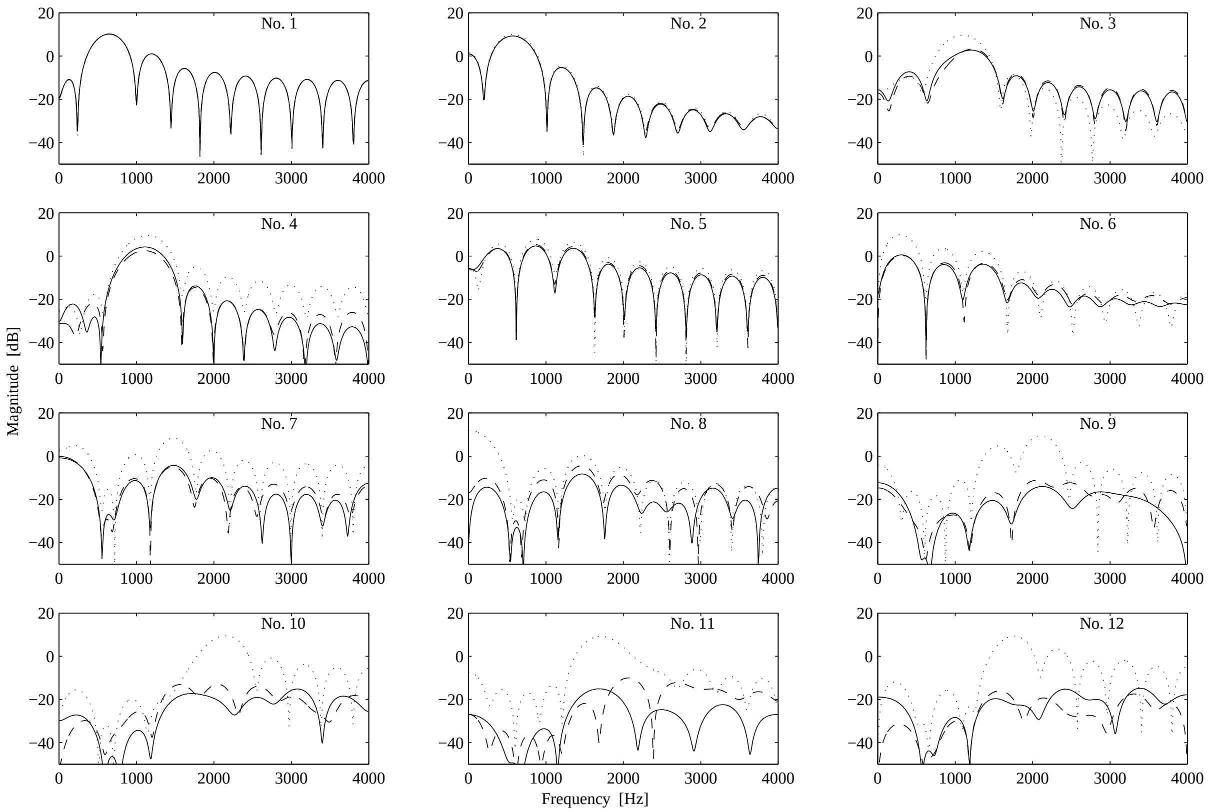


Figure 6.7 (Solid) The first 12 eigenfilters of $\mathbf{X} \in \mathbb{R}^{141\times 20}$ representing a noisy voiced speech frame with 160 samples and SNR=5dB. The filters are averaged over 100 independent white noise realizations. (Dashed) The corresponding eigenfilters obtained from the RRULVD based rank-12 approximation. (Dotted) The eigenfilters of the signal-only matrix.

6.2 Linear Signal Estimators by RRULVD

The Least Squares (LS), Minimum Variance (MV) and empirical TDC estimates of the clean signal from the noisy signal as described in Sections 4.1 can be *approximated* by using the rank-revealing ULV decomposition. The RRULVD based formulation of the last two estimates are proposals given here (see also [53, 54]).

In this section, $\mathbf{X} \in \mathbb{R}^{m\times n}$ is a measurement matrix defined by (3.8), with the signal matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ satisfying (3.9), thus the signal subspace is known to have rank $p < n \leq m$.

6.2.1 Least Squares Estimator

An *Approximate Least Squares* (ALS) estimate $\hat{\mathbf{S}}_{ALS}$ of the signal matrix $\mathbf{S}$ can be computed by essentially substituting the RRULV decomposition for the SVD in (4.32), thus replacing one problem with a similar, nearby problem that can be solved more efficiently.

Based on Theorem 6.2, a useful rank-p matrix approximation to $\mathbf{X} \in \mathbb{R}^{m\times n}$ is given by

$$\hat{\mathbf{S}}_{ALS} = \mathbf{X}_p = \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{V}_{X1}^T = \mathbf{U}_{X1}\mathbf{U}_{X1}^T\mathbf{X} \tag{6.27}$$

or

$$\hat{\mathbf{S}}_{ALS} = \mathbf{X}_p = \mathbf{X}\mathbf{V}_{X1}\mathbf{V}_{X1}^T \tag{6.28}$$

where $\mathbf{U}_{X1}$ and $\mathbf{V}_{X1}$ approximate the numerical column space and row space as defined via the SVD of $\mathbf{X}$. Note, that (6.27) and (6.28) are only equal when $\mathbf{F}_X = \mathbf{0}$, i.e., in general

$$\begin{aligned} \mathbf{X}\mathbf{V}_{X1}\mathbf{V}_{X1}^T &= \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{V}_{X1}^T + \mathbf{U}_{X2}\mathbf{F}_X\mathbf{V}_{X1}^T && (6.29) \\ &= \mathbf{U}_{X1}\mathbf{U}_{X1}^T\mathbf{X} + \mathbf{X}\mathbf{V}_{X2}\mathbf{G}_X^{-1}\mathbf{F}_X\mathbf{V}_{X1}^T && (6.30) \end{aligned}$$

and the relative difference between the two approximation methods is bounded by

$$\frac{\|\mathbf{X}\mathbf{V}_{X1}\mathbf{V}_{X1}^T - \mathbf{U}_{X1}\mathbf{U}_{X1}^T\mathbf{X}\|_2}{\|\mathbf{X}\|_2} \le \|\mathbf{G}_X^{-1}\|_2\|\mathbf{F}_X\|_2 \tag{6.31}$$

So in speech enhancement applications with relative high noise levels, i.e., $\|\mathbf{F}_X\|_2 \ll \sigma_{noise}$ and $\|\mathbf{G}_X^{-1}\|_2 \approx \sigma_{noise}^{-1}$, the bound is small and both estimates can be used.

For the low noise case, the accuracy of the rank-p matrix approximation is analysed in [36], where the subspace bounds play an important role. It is shown that when the subspace angles are sufficiently small then $\mathbf{X}_p$ in (6.28) closely approximates the SVD based least squares estimate $\mathbf{X}_{p,SVD}$. First, consider the perturbation

$$\begin{aligned} \|\mathbf{X}_{p,SVD} - \mathbf{X}_p\|_2 &= \|\mathbf{X}\ (\mathbf{V}_{X1,SVD}\mathbf{V}_{X1,SVD}^T - \mathbf{V}_{X1}\mathbf{V}_{X1}^T)\|_2 && (6.32) \\ &\le \|\mathbf{X}\|_2\ \|\mathbf{V}_{X1,SVD}\mathbf{V}_{X1,SVD}^T - \mathbf{V}_{X1}\mathbf{V}_{X1}^T\|_2 \\ &= \|\mathbf{X}_{p,SVD}\|_2\ \|\mathbf{V}_{X1,SVD}\mathbf{V}_{X1,SVD}^T - \mathbf{V}_{X1}\mathbf{V}_{X1}^T\|_2 \\ &= \|\mathbf{X}_{p,SVD}\|_2\ \|\sin\mathbf{\Theta}(\langle\mathbf{V}_{X1,SVD}\rangle, \langle\mathbf{V}_{X1}\rangle)\|_2 \end{aligned}$$

where Definition 3.2 on Page 33 has been used. Then from (6.10) it follows that

$$\frac{\|\mathbf{X}_{p,SVD} - \mathbf{X}_p\|_2}{\|\mathbf{X}_{p,SVD}\|_2} \le \|\sin\mathbf{\Theta}(\langle\mathbf{V}_{X1,SVD}\rangle, \langle\mathbf{V}_{X1}\rangle)\|_2 \le \frac{\rho\|\mathbf{F}_X\|_2}{(1-\rho^2)\sigma_{min}(\mathbf{L}_{X1})} \tag{6.33}$$

showing that the relative error in $\mathbf{X}_p$ is proportional to $\|\mathbf{F}_X\|_2$. An equivalent bound based on Equation (6.11) can be obtained for (6.27). It is interesting to notice that $\|\mathbf{F}_X\|_2$ can be made very small using refinement procedures as discussed in Section 7.5.3.

6.2.2 Minimum Variance Estimator

The Minimum Variance (MV) estimate $\hat{\mathbf{S}}_{MV}$ of the signal matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ as described in Section 4.1.4 can also be formulated by means of the RRULVD. The basic idea is to introduce an *idealized* rank-revealing ULV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$, i.e., with reference to (6.3), the necessary conditions are

ASSUMPTION 6.1 (*Algebraic and Geometric Conditions)*

1. *All three conditions of Assumption 3.3 in Section 3.4 are satisfied, i.e.,*
   - *(a) The signal is orthogonal to the noise in the sense:* $\mathbf{S}^T\mathbf{N} = \mathbf{0}$.
   - *(b) The matrix* $\mathbf{N} = \sigma_{noise}\mathbf{Q}$, *where* $\mathbf{Q}$ *has orthonormal columns:* $\mathbf{N}^T\mathbf{N} = \sigma_{noise}^2\mathbf{I}_n$.
   - *(c) There is a distinct gap in the singular values of the matrix* $\mathbf{X}$*:* $\sigma_{X,p} > \sigma_{X,p+1}$.
2. *The off-diagonal block* $\mathbf{F}_X$ *is zero.*
3. $\mathbf{G}_X$ *is a diagonal matrix containing the noise-only singular values* σ_{noise}.

Notice that the two additional assumptions are related to the RRULV algorithm and not (directly) to the signals. Thus, the idealized decomposition become

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \tag{6.34}$$

Similarly, let the idealized RRULVD (or complete orthogonal decomposition) of the low rank signal matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ be defined by

$$\mathbf{S} = \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \tag{6.35}$$

where $\mathbf{L}_{S1} \in \mathbb{R}^{p\times p}$, then Equation (6.34) can also be written in terms of (6.35), i.e.,

$$\begin{aligned} \mathbf{X} &= \mathbf{S} + \mathbf{N} \qquad (6.36) \\ &= \mathbf{U}_{S1}\mathbf{L}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S2}\mathbf{V}_{S2}^T \\ &= \begin{pmatrix} (\mathbf{U}_{S1}\mathbf{L}_{S1} + \mathbf{N}\mathbf{V}_{S1})\mathbf{L}_{X1}^{-1} & \mathbf{N}\mathbf{V}_{S2}\sigma_{noise}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \end{aligned}$$

The matrix $\mathbf{L}_{X1}^T\mathbf{L}_{X1}$ can be obtained by comparing the sample correlation matrix of $\mathbf{X}$ using the definitions of $\mathbf{S}$ and $\mathbf{N}$

$$\begin{aligned} \mathbf{X}^T\mathbf{X} &= \mathbf{S}^T\mathbf{S} + \mathbf{N}^T\mathbf{N} \qquad (6.37) \\ &= \mathbf{V}_{S1}\mathbf{L}_{S1}^T\mathbf{L}_{S1}\mathbf{V}_{S1}^T + \sigma_{noise}^2\mathbf{V}_{S1}\mathbf{V}_{S1}^T + \sigma_{noise}^2\mathbf{V}_{S2}\mathbf{V}_{S2}^T \\ &= \begin{pmatrix} \mathbf{V}_{S1} & \mathbf{V}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}^2\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \end{aligned}$$

with the one based on the idealized RRULV decomposition of $\mathbf{X}$ (6.34)

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} \mathbf{V}_{X1} & \mathbf{V}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1}^T\mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}^2\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \tag{6.38}$$

which gives the relation

$$\mathbf{L}_{X1}^T\mathbf{L}_{X1} = \mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p \tag{6.39}$$

In (6.37), Assumption 6.1-1 has been used, and this is also a necessary condition for (6.36) to be a ULV decomposition, i.e., $\mathbf{U}_X$ has orthonormal columns

$$\begin{aligned} \mathbf{U}_X^T\mathbf{U}_X &= \begin{pmatrix} \mathbf{L}_{X1}^{-T}(\mathbf{L}_{S1}^T\mathbf{U}_{S1}^T + \mathbf{V}_{S1}^T\mathbf{N}^T) \\ \sigma_{noise}^{-1}\mathbf{V}_{S2}^T\mathbf{N}^T \end{pmatrix} \qquad (6.40) \\ &\qquad \times \begin{pmatrix} (\mathbf{U}_{S1}\mathbf{L}_{S1} + \mathbf{N}\mathbf{V}_{S1})\mathbf{L}_{X1}^{-1} & \mathbf{N}\mathbf{V}_{S2}\sigma_{noise}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}_{X1}^{-T}(\mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p)\mathbf{L}_{X1}^{-1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}^{-1}\sigma_{noise}^2\mathbf{I}_{n-p}\sigma_{noise}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-p} \end{pmatrix} \end{aligned}$$

Using (6.35) and (6.36) in the MV definition (4.42) yields the desired MV estimate of $\mathbf{S}$

$$\begin{aligned}
\hat{\mathbf{S}}_{MV} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \qquad (6.41)\\
&= \mathbf{U}_X\mathbf{U}_X^T\mathbf{S}\\
&= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1}^{-T}(\mathbf{L}_{S1}^T\mathbf{U}_{S1}^T + \mathbf{V}_{S1}^T\mathbf{N}^T) \\ \sigma_{noise}^{-1}\mathbf{V}_{S2}^T\mathbf{N}^T \end{pmatrix}\\
&\qquad \times \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix}\\
&= \mathbf{U}_{X1}\mathbf{L}_{X1}^{-T}(\mathbf{L}_{S1}^T\mathbf{L}_{S1})\mathbf{V}_{X1}^T\\
&= \mathbf{U}_{X1}\mathbf{L}_{X1}^{-T}(\mathbf{L}_{X1}^T\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{I}_p)\mathbf{V}_{X1}^T\\
&= \mathbf{U}_{X1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T
\end{aligned}$$

where Assumption 6.1-1 and (6.39) have been used. This equation can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}}_{MV} = \mathbf{X}\mathbf{V}_{X1}\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \qquad (6.42)$$

In practice, Assumption 6.1-2 and 6.1-3 are nearly satisfied due to the way of implementing the RRULV algorithm, and this can further be improved by a number of refinement steps as discussed in Section 7.5.3. However, Assumption 6.1 are never satisfied exactly, but as the SVD, the rank-revealing ULV decomposition is robust with respect to mild violations of these conditions.

The *Approximate* MV (AMV) estimate $\hat{\mathbf{S}}_{AMV}$ can now be computed by substituting the rank-revealing ULV decomposition (6.3) for the idealized RRULV decomposition (6.34), and the quantity σ_{noise}^2 can, e.g., be obtained from (6.5)

$$\sigma_{noise}^2 = \frac{1}{n-p}\sum_{i=p+1}^{n}\sigma_{X,i}^2 \approx \frac{1}{n-p}\left(\|\mathbf{F}_X\|_F^2 + \|\mathbf{G}_X\|_F^2\right) \qquad (6.43)$$

Note, that the two estimation Equations (6.41) and (6.42) are no longer equal (see Section 6.2.1).

6.2.3 Empirical TDC Estimator

Assume that Assumption 6.1 in the last section is satisfied, then the empirical TDC estimate as defined in Section 4.1.7 can be obtained by applying the idealized RRULVD of $\mathbf{S}$ (6.35) to (4.78), i.e.,

$$\mathbf{W}_{ETDC} = \mathbf{V}_{S1}(\mathbf{L}_{S1}^T\mathbf{L}_{S1} + \gamma\sigma_{noise}^2\mathbf{I}_p)^{-1}\mathbf{L}_{S1}^T\mathbf{L}_{S1}\mathbf{V}_{S1}^T \qquad (6.44)$$

Using (6.36) and (6.39) in this equation gives

$$\begin{aligned}
\mathbf{W}_{ETDC} &= \mathbf{V}_{X1}(\mathbf{L}_{X1}^T\mathbf{L}_{X1} - \sigma_{noise}^2(1-\gamma)\mathbf{I}_p)^{-1}(\mathbf{L}_{X1}^T\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{I}_p)\mathbf{V}_{X1}^T \qquad (6.45)\\
&= \mathbf{V}_{X1}(\mathbf{L}_{X1} - \sigma_{noise}^2(1-\gamma)\mathbf{L}_{X1}^{-T})^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T
\end{aligned}$$

As in the MV method, the *Approximate* TDC (ATDC) estimate can now be computed by substituting the RRULVD for the idealized one, i.e.,

$$\hat{\mathbf{S}}_{ATDC} = \mathbf{X}\mathbf{W}_{ATDC} \qquad (6.46)$$

or

$$\hat{\mathbf{S}}_{ATDC} = \mathbf{U}_{X1}\mathbf{L}_{X1}(\mathbf{L}_{X1} - \sigma_{noise}^2(1-\gamma)\mathbf{L}_{X1}^{-T})^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \qquad (6.47)$$

Note, that the related SDC method (see Section 4.1.7) can not be implemented by the RRULVD, because the decomposition only tracks two subspaces, the signal- and noise subspace, i.e., information about the individual eigenfilters can not be obtained.

6.2.4 A Unified Notation

Also for the RRULVD based linear signal estimators, it is convenient to introduce a unified notation. First note that the transformation $\mathbf{y} = \mathbf{V}_X^T \mathbf{x}$ *approximates* the Karhunen-Loeve transform of $\mathbf{x}$, when $\mathbf{V}_X$ is generated by the RRULVD of the data matrix $\mathbf{X}$. Hence, all the above mentioned approximate linear signal estimates are obtained by the following steps (see Figure 6.8)

- AKLT of the noisy signal onto the signal subspace.
- Modify the components of the AKLT by a symmetric *gain filter matrix* $\mathbf{G}_1$.
- Inverse AKLT of the modified components to reconstruct the signal in the signal subspace.

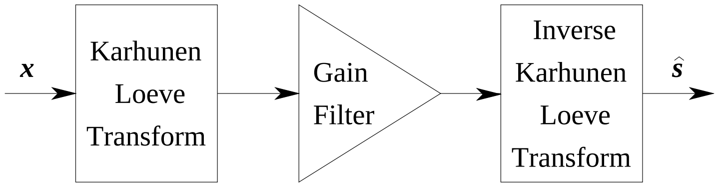


Figure 6.8 General model for approximate linear estimator.

This scheme results in a generalized formulation of the approximate linear estimators similar to Equation (4.88), i.e.,

$$\hat{\mathbf{S}} = \mathbf{X}\mathbf{W} = \mathbf{X}\mathbf{V}_X \mathbf{G} \mathbf{V}_X^T = \mathbf{X}\mathbf{V}_{X1}\mathbf{G}_1\mathbf{V}_{X1}^T\,, \qquad \mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{6.48}$$

where the gain filter matrix $\mathbf{G}_1 \in \mathbb{R}^{p\times p}$ depends on the estimation method (see Table 6.1).

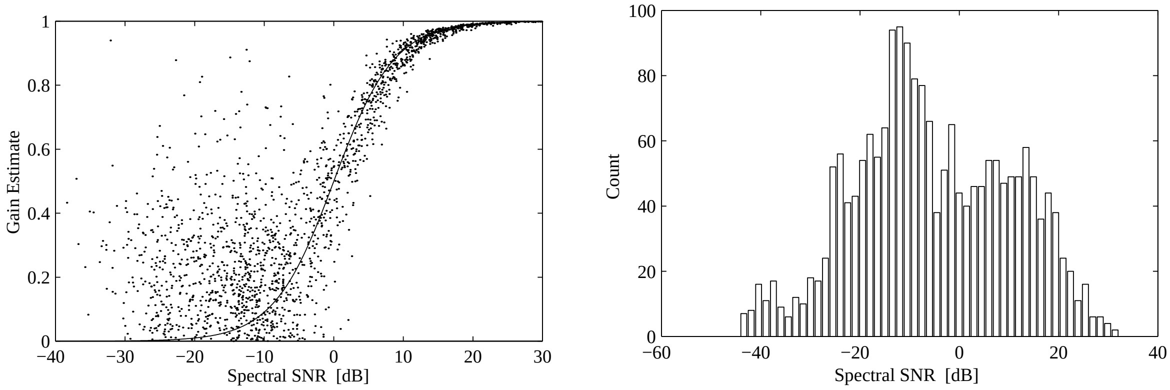
Method	Gain filter matrix $\mathbf{G}_1$	
LS	$\mathbf{I}_p$	$\mathbf{I}_p$
MV	$(\mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p)^{-1}\mathbf{L}_{S1}^T\mathbf{L}_{S1}$	$\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})$
ETDC	$(\mathbf{L}_{S1}^T\mathbf{L}_{S1} + \gamma\sigma_{noise}^2\mathbf{I}_p)^{-1}\mathbf{L}_{S1}^T\mathbf{L}_{S1}$	$(\mathbf{L}_{X1} - \sigma_{noise}^2(1-\gamma)\mathbf{L}_{X1}^{-T})^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})$

Table 6.1 Gain matrix for different estimation methods formulated in terms of the clean signal (first column) or the noisy signal (second column).

Plots of the estimated gains can be obtained from the eigendecomposition of $\mathbf{G}_1$ defined by $\mathbf{Q}_{G1}\mathbf{\Lambda}_{G1}\mathbf{Q}_{G1}^T$, i.e., the eigenvalue $\lambda_{G1,i}$ represents an estimated gain related to the modified transformation vector $\mathbf{V}_{X1}\mathbf{q}_{G1,i}$. The corresponding spectral SNR in that direction is given by

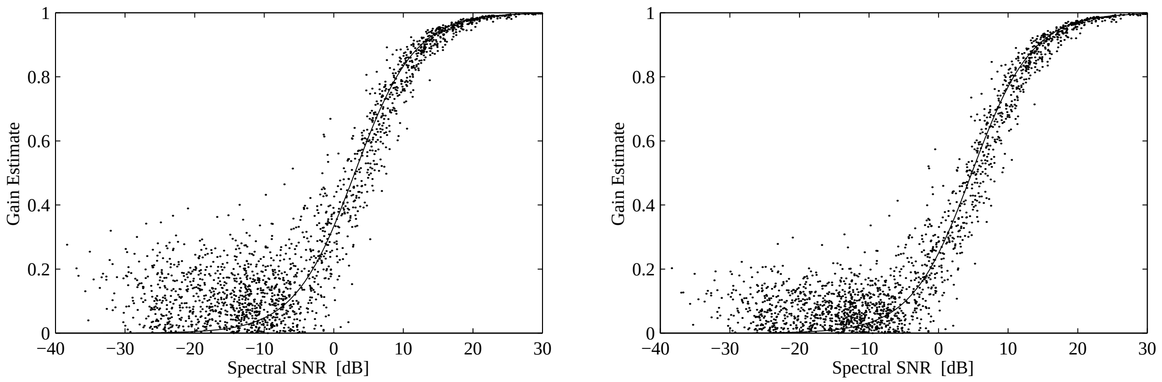
$$\mathrm{SNR}_{spectral} = \frac{\|\mathbf{S}\mathbf{V}_{X1}\mathbf{q}_{G1,i}\|_2^2}{\|\mathbf{N}\mathbf{V}_{X1}\mathbf{q}_{G1,i}\|_2^2} \tag{6.49}$$

Figure 6.9 and 6.10 shows estimated Wiener gains $\{\lambda_{G1,i}\}_1^{12}$ of 165 speech frames ($\mathbf{X} \in \mathbb{R}^{141\times 20}$) obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample



(a) (b)

Figure 6.9 (a) Estimated Wiener gains $\lambda_i(\mathbf{G}_1)$ of 165 speech frames ($\mathbf{X} \in \mathbb{R}^{141\times 20}$) obtained from the noisy reference sentence by shifting a 160 sample window by 160 sample (white noise and SNR=10dB). The estimated gains are plotted as function of the spectral SNR, and $\mathbf{G}_1$ is obtained from a 12-dimensional signal subspace using the RRULV algorithm without refinement. (b) Distribution of the gains.



(a) (b)

Figure 6.10 As Figure 6.9, but with gains given by the TDC estimator (a) $\gamma = 2$ and (b) $\gamma = 3$.

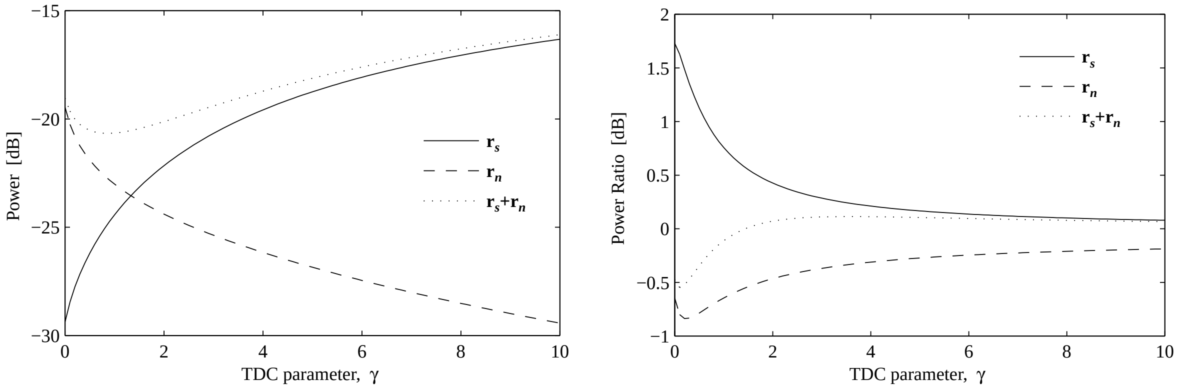
(white noise and SNR = 10 dB). The gain matrix $\mathbf{G}_1$ is calculated from (6.42) using the lower triangular matrix $\mathbf{L}_{X1}$ corresponding to a 12-dimensional signal subspace, and a σ^2_{noise} obtained from an initial noise matrix $\mathbf{N}$. The RRULV algorithm has been applied without refinement.

Like the related examples in Figure 4.4 and 4.5, a large variance in the estimated gains is observed for spectral SNR less than about 0 dB due to the difference in the estimated eigenvalues of the noise given by $\sigma^2_{noise}\mathbf{I}$ and the true ones $\mathbf{\Sigma}^2_N$. *However, the performance of the RRULVD based method is comparable with the SVD based method.*

Now, consider the voiced speech frame of 160 samples added white noise (SNR = 5 dB), and organized in the data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$. The linear estimators as defined by Equation (6.48) with $p = 12$ are then characterized by the residual matrices $\mathbf{R} = \mathbf{R}_S + \mathbf{R}_N$, c.f., Equation (4.72), or the corresponding residual signals obtained by averaging along the diagonals (see Section 4.6). Like the SVD case, the residual signals can be used to compare the different estimators as illustrated in Figure 6.11(a), where the power of the residual signals are plotted as function of the TDC parameter γ. Clearly, for increasing parameter values, the level of the residual noise $\mathbf{r}_n$ decreases while the level of signal distortion $\mathbf{r}_n$ increases. The minimum residual power is obtained for the MV estimator ($\gamma = 1$) and is dominated by the residual noise, however, by choosing $\gamma > 1.5$, the signal distortion will become dominant for the price of an increase in the level of the total residual signal.

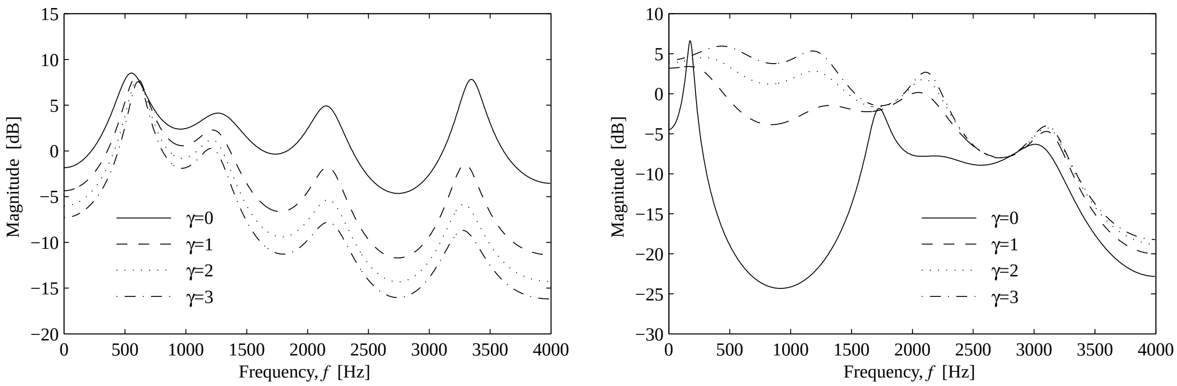
In Figure 6.11(a), the powers are compared with the corresponding SVD-based example in Figure 4.6(a), i.e., the ratios shown are $P_{\mathbf{r}}/P_{\mathbf{r},SVD}$. Thus, the RRULVD based method results in a larger signal distortion and smaller residual noise level, however, for $\gamma > 1$, the increase in the total residual signal is less than 0.1 dB. Similar results are obtained for the unvoiced frame.

The 10th order LPC-based magnitude spectra of the residual noise and signal distortion corresponding to the examples in Figure 6.11 are shown in Figure 6.12. As expected from the above discussion, there are only small changes compared with the SVD based method (see Figure 4.7), especially in the LS case.



(a) (b)

Figure 6.11 (a) Power of the residual noise $\mathbf{r}_n$ and the signal distortion $\mathbf{r}_s$ for the RRULVD based TDC estimator obtained from a 12-dimensional signal subspace. The data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ represents the voiced speech frame consisting of 160 samples and added white noise (SNR=5dB). The residual levels are plotted against the TDC parameter γ. (b) Ratios of the residual powers obtained from RRULVD and SVD based estimators.



(a) (b)

Figure 6.12 10th order LPC-based magnitude spectra of the residual noise (a) and signal distortion (b) corresponding to the examples in Figure 6.11(a).

6.3 The ULLV Decomposition

Like the QSVD, the ULLV decomposition operates on a pair of matrices. The definition given here, is the one used by Luk and Qiao [73] (a slightly different formulation has been given by Bojanczyk [10]).

THEOREM 6.8 (*ULLV Decomposition (ULLVD)) Given a matrix* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *and a matrix* $\mathbf{N} \in \mathbb{R}^{m\times n}$ *of full rank, where* $m \geq n$, *then there exist matrices* $\mathbf{U}_X \in \mathbb{R}^{m\times n}, \mathbf{U}_N \in \mathbb{R}^{m\times n}$ *with orthogonal columns and a orthogonal matrix* $\mathbf{V} \in \mathbb{R}^{n\times n}$ *such that*

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{L} \mathbf{V}^T \tag{6.50}$$

$$\mathbf{N} = \mathbf{U}_N \mathbf{L} \mathbf{V}^T \tag{6.51}$$

where $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ *and* $\mathbf{L} \in \mathbb{R}^{n\times n}$ *are lower triangular.*

As the ULV decomposition, so is the ULLV decomposition not unique. Also the ULLV decomposition can reveale the rank but of the matrix $\mathbf{X}\mathbf{N}^+$ assuming $\mathbf{N}$ has full rank.

THEOREM 6.9 (*Rank-Revealing ULLV Decomposition (RRULLVD)) Given two matrices* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{N} \in \mathbb{R}^{m\times n}$, *where* $m \geq n$, *and assume that* $\mathbf{X}\mathbf{N}^+$ *has numerical rank* $p < n$ *corresponding to a given tolerance* τ, *i.e., its quotient singular values satisfy*

$$\delta_1 \geq \cdots \geq \delta_p \geq \tau \gg \delta_{p+1} \geq \cdots \geq \delta_n \tag{6.52}$$

Then there exist matrices $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ *and* $\mathbf{U}_N \in \mathbb{R}^{m\times n}$ *with orthogonal columns and a orthogonal matrix* $\mathbf{V} \in \mathbb{R}^{n\times n}$ *such that*

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{L} \mathbf{V}^T = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \mathbf{L} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{6.53}$$

$$\mathbf{N} = \mathbf{U}_N \mathbf{L} \mathbf{V}^T \tag{6.54}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$, $\mathbf{L}_{X1} \in \mathbb{R}^{p\times p}$ *and* $\mathbf{G}_X \in \mathbb{R}^{(n-p)\times(n-p)}$ *are lower triangular, and*

$$\sigma_{min}(\mathbf{L}_{X1}) \approx \delta_p \tag{6.55}$$

$$\|\mathbf{F}_X\|_F^2 + \|\mathbf{G}_X\|_F^2 \approx \delta_{p+1}^2 + \cdots + \delta_n^2 \tag{6.56}$$

Thus, from the rank-revealing ULLV decomposition, the signal- and noise subspaces defined by the gap in the quotient singular values can be estimated. It is often convenient to define

$$\mathbf{L}\mathbf{V}^T = \mathbf{\Theta}^T = \mathbf{Z}^{-1} \tag{6.57}$$

$$\begin{pmatrix} \mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{11}\mathbf{V}_1^T \\ \mathbf{L}_{21}\mathbf{V}_1^T + \mathbf{L}_{22}\mathbf{V}_2^T \end{pmatrix} \tag{6.58}$$

$$\begin{pmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{V}_1\mathbf{L}_{11}^{-1} - \mathbf{V}_2\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{L}_{11}^{-1} & \mathbf{V}_2\mathbf{L}_{22}^{-1} \end{pmatrix} \tag{6.59}$$

(6.60)

where the partioning of $\mathbf{L}$ is similar to (6.53).

6.3.1 Subspace Methods and the RRULLVD

Now, consider the situation where the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ defined by (3.8) consists of the low-rank signal matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ added *colored* noise, and assume that the noise-only matrix $\mathbf{N} \in \mathbb{R}^{m \times n}$ can be estimated in periods without speech. Hence, by using the QR decomposition of $\mathbf{N}$ (3.84) and Theorem 6.9, the prewhitened data matrix is

$$\begin{aligned}
\mathbf{X}\mathbf{R}^{-1} &= \mathbf{X}\mathbf{N}^{+}\mathbf{Q} \\
&= \mathbf{X}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{Q} \\
&= \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{V}^T(\mathbf{V}\mathbf{L}^T\mathbf{U}_N^T\mathbf{U}_N\mathbf{L}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{L}^T\mathbf{U}_N^T\mathbf{Q} \\
&= \mathbf{U}_X\mathbf{L}_X\mathbf{U}_N^T\mathbf{Q} \\
&= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \begin{pmatrix} \mathbf{U}_{N1}^T \\ \mathbf{U}_{N2}^T \end{pmatrix} \mathbf{Q}
\end{aligned} \tag{6.61}$$

where the matrix $\bar{\mathbf{U}}_N^T = \mathbf{U}_N^T\mathbf{Q} = (\mathbf{R}\mathbf{V}\mathbf{L}^{-1})^T$ is orthogonal. Thus, the RRULLVD of $(\mathbf{X}, \mathbf{N})$ and the RRULVD of $\mathbf{X}\mathbf{R}^{-1}$ are equivalent, and in the white-noise case, the two decompositions yield the same information, which can be seen from (6.61) with $\mathbf{R} = \sigma_{noise}\mathbf{I}_n$, i.e.,

$$\begin{aligned}
\mathbf{X} &= \mathbf{U}_X\mathbf{L}_X\bar{\mathbf{U}}_N^T\sigma_{noise} \\
&= \mathbf{U}_X(\mathbf{L}_X\sigma_{noise})(\sigma_{noise}\mathbf{V}\mathbf{L}^{-1})^T
\end{aligned} \tag{6.62}$$

From (6.61) it is also clear that the perturbation theory in Section 6.1.1 and 6.1.2 can be used to compare the RRULLVD with the QSVD.

6.3.2 Linear Signal Estimators by RRULLVD

All the RRULVD based approximate linear signal estimators in Section 6.2 can in the colored noise case easily be obtained from the RRULLVD of $(\mathbf{X}, \mathbf{N})$.

First, the desired rank-p estimate of the noise normalized signal $\mathbf{S}\mathbf{N}^{+}$ is constructed from the normalized data matrix $\mathbf{X}\mathbf{N}^{+}$ (6.61) along the lines in Section 6.2.1, 6.2.2 and 6.2.3, i.e.,

$$\widehat{\mathbf{S}\mathbf{N}^{+}} = \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{G}_1\mathbf{U}_{N1}^T \tag{6.63}$$

where the gain matrix $\mathbf{G}_1 \in \mathbb{R}^{p \times p}$ depends on the estimation method (see Table 6.2).

Method	Gain filter matrix $\mathbf{G}_1$
LS	$\mathbf{I}_p$
MV	$\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \mathbf{L}_{X1}^{-T})$
ETDC	$(\mathbf{L}_{X1} - (1-\gamma)\mathbf{L}_{X1}^{-T})^{-1}(\mathbf{L}_{X1} - \mathbf{L}_{X1}^{-T})$

Table 6.2 Gain matrix for different estimation methods.

Thus, it is computed by the same formulas as in the RRULVD based algorithms, but by using the fact that the noise variance of the prewhitened signal is one.

To obtain the corresponding rank-p estimate of $\mathbf{S}$, the estimate of $\mathbf{S}\mathbf{N}^{+}$ must be denormalized by the noise-only matrix $\mathbf{N}$

$$\begin{aligned}
\hat{\mathbf{S}} &= (\widehat{\mathbf{S}\mathbf{N}^{+}})\mathbf{N} \\
&= \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{G}_1\mathbf{L}_{11}\mathbf{V}_1^T \\
&= \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{G}_1\mathbf{\Theta}_1^T
\end{aligned} \tag{6.64}$$

which can be computed directly from the RRULLVD, i.e., the prewhitening is now an integral part of the algorithm. In Equation (6.64), definition (6.57) has been used, and from the identity

$$\begin{pmatrix} \mathbf{\Theta}_1^T\mathbf{Z}_1 & \mathbf{\Theta}_1^T\mathbf{Z}_2 \\ \mathbf{\Theta}_2^T\mathbf{Z}_1 & \mathbf{\Theta}_2^T\mathbf{Z}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-p} \end{pmatrix} \tag{6.65}$$

the matrix $\mathbf{U}_{X1}$ is given by

$$\mathbf{U}_{X1} = \mathbf{X}\mathbf{Z}_1\mathbf{L}_{X1}^{-1} - \mathbf{U}_{X2}\mathbf{F}_X\mathbf{L}_{X1}^{-1} \approx \mathbf{X}\mathbf{Z}_1\mathbf{L}_{X1}^{-1} \tag{6.66}$$

i.e., Equation (6.64) can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\begin{aligned}\hat{\mathbf{S}} &= \mathbf{X}\mathbf{Z}_1\mathbf{G}_1\mathbf{\Theta}_1^T \\ &= \mathbf{X}\mathbf{V}_1\mathbf{L}_{11}^{-1}\mathbf{G}_1\mathbf{L}_{11}\mathbf{V}_1^T - \mathbf{X}\mathbf{V}_2\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{L}_{11}^{-1}\mathbf{G}_1\mathbf{L}_{11}\mathbf{V}_1^T\end{aligned} \tag{6.67}$$

As discussed in Section 6.2.1, the approximation in (6.66) gives no significant difference between the two estimates (6.64) and (6.67).

6.4 Recursive Implementation

In a recursive implementation, a rank-1 update/downdate of the data matrix is performed (see Chapter 7), i.e., the shift between adjacent frames is always one sample. With reference to Section 4.7, the synthesis window will therefore be of length one

$$W_s = 1 \tag{6.68}$$

The analysis window is determined by the ULV updating/downdating algorithm. At the moment, only a sliding (rectangular) and exponential window can be used. The case with a rectangular window follows from the frame based implementation, i.e., it is centered around the middle row of $\mathbf{X}$, which is selected by the synthesis window. The case with exponential window differ a little due to the non symmetric shape. Example plots of two windows with forgetting factor $\beta = 0.95$ and 0.99 are shown in Figure 6.13(a) and the corresponding time sequence

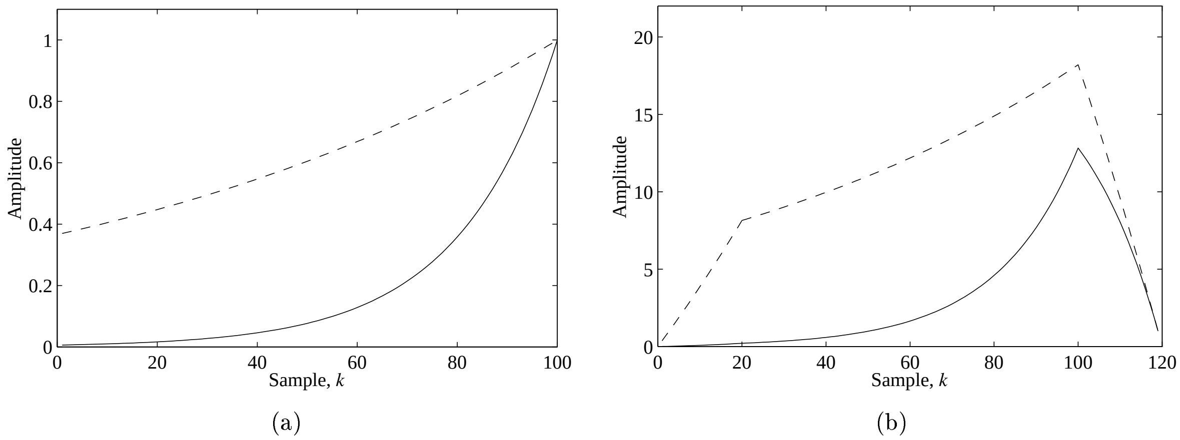


Figure 6.13 (a) Example time plots for the exponential windows with length $m = 100$ and forgetting factor $\beta = 0.95$ (solid) and 0.99 (dashed). (b) The windows after convolution with a rectangular window of length $n = 20$.

related windows in Figure 6.13(b). Thus, in order to operate on the row with maximum weight, the offset between the analysis and the synthesis window must in this case be

$$\Delta K_{s,exp} = m - 1 \tag{6.69}$$

The overall filter structure is shown in Figure 6.14.

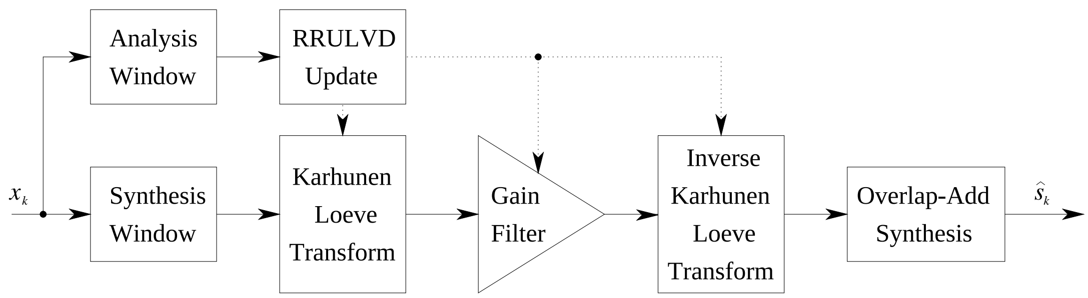


Figure 6.14 Filter structure.

6.5 Summary

Traditionally, the signal subspace approach for nonparametric speech enhancement is formulated by means of the SVD (or the eigendecomposition) using a frame-based implementation.

Here, the rank-revealing ULV decomposition is used instead of the SVD, and a recursive updating of the estimate is used instead of working in frames. An ULV formulation of three different estimation strategies is considered, where the last two are new proposals. In the colored noise case, the estmates are formulated by means of the rank-revealing ULLV decomposition, where the prewhitening operation becomes an integral part of the algorithm.

Experiments demonstrate that the ULV-based algorithms are able to achieve the same quality of the reconstructed speech signal as the SVD-based method. Thus, the results in Chapter 4 can also be obtained (with minor variations) by the ULV approach.

Chapter 7

The ULLV Algorithm

The rank-revealing ULV/ULLV decompositions can be used in many problems, e.g., speech enhancement as considered here, since not all of the information provided by the SVD is actually needed. However, the main advantage of the rank-revealing ULV/ULLV decompositions lies in the fact that they are numerically stable and easy to update/downdate.

In this chapter, both initial and recursive computations of the rank-revealing ULLV decomposition are considered. The computational complexity is also given, and error analysis is briefly discussed.

7.1 Algorithm Overview

The rank-revealing ULLV decomposition can be used in connection with up- and downdating problems arising from recursive algorithms in signal processing. Starting with the initial decompositions (6.53-6.54), repeated here for convenience

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{L} \mathbf{V}^T = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \mathbf{L} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{7.1}$$

$$\mathbf{N} = \mathbf{U}_N \mathbf{L} \mathbf{V}^T \tag{7.2}$$

the matrices are updated as the data $\mathbf{x}^T$ and $\mathbf{n}^T$ are brought in one row at a time. A new row is processed by some or all of the following steps:

- Updating: The new row of data is appended to $\mathbf{X}$ or $\mathbf{N}$ and incorporated into the decomposition.
- Downdating: The oldest row of data in $\mathbf{X}$ or $\mathbf{N}$ is isolated and removed in the decomposition.
- Deflation: Establish and maintain the rank-revealing nature of the decomposition.
- Refinement: The norm of the off-diagonal block $\mathbf{F}_X$ in the rank-revealing triangular matrix $\mathbf{L}_X$ is reduced to improve the subspace quality.

where special care is taken to keep small elements small in the matrices $\mathbf{F}_X$ and $\mathbf{G}_X$. In many applications, one would normally like to update only the $\mathbf{L}_X$, $\mathbf{L}$ and $\mathbf{V}$ factors without the extra cost of storing and updating the matrices $\mathbf{U}_X$ and $\mathbf{U}_N$. Such algorithms are less stable than methods that update all factors as discussed in Section 7.4.2. However, they can be stabilized by a refinement technique using the original data matrix $\mathbf{X}$.

The following discussion will assume that only one row $\mathbf{x}^T$ is being added/deleted (see [87] for rank-k problems), i.e., the matrix $\mathbf{X}$ is updated as

$$\hat{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}^T \end{pmatrix} \quad \Rightarrow \quad \hat{\mathbf{X}}^T\hat{\mathbf{X}} = \mathbf{X}^T\mathbf{X} + \mathbf{x}\mathbf{x}^T \tag{7.3}$$

and downdated as

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} \quad \Rightarrow \quad \hat{\mathbf{X}}^T\hat{\mathbf{X}} = \mathbf{X}^T\mathbf{X} - \mathbf{x}\mathbf{x}^T \tag{7.4}$$

Hence, the up/downdating of $\mathbf{X}$ is mathematically, but not numerically, equivalent to a rank one perturbation of the sample correlation matrix $\hat{\mathbf{R}}_x$.

From the relation $\sigma_i^2(\mathbf{X}) = \lambda_i(\mathbf{X}^T\mathbf{X})$ and classical perturbation theory for eigenvalues [39, page 412], it follows that the singular values $\hat{\sigma}_i = \sigma_i(\hat{\mathbf{X}})$ interleave with $\sigma_i = \sigma_i(\mathbf{X})$, where for updating

$$\hat{\sigma}_1 \geq \sigma_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_n \geq \sigma_n \geq 0 \tag{7.5}$$

and for downdating

$$\sigma_1 \geq \hat{\sigma}_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq \hat{\sigma}_n \geq 0 \tag{7.6}$$

Hence, in downdating, the smallest singular value may decrease to $\hat{\sigma}_n \approx 0$, i.e., the modified matrix $\hat{\mathbf{X}}$ may become rank deficient even when $\mathbf{X}$ has full rank. Moreover, any singular value may decrease by a considerable amount, which indicates that downdating can be a sensitive problem. On the other hand, updating $\mathbf{X}$ will increase all its singular values. Thus, accurate algorithms are needed when the downdating problem is ill-conditioned.

The rank-revealing ULLV algorithm presented in this chapter is new in its complete treatment and implementation. The update method was described by Luk and Qiao [73, 93], while the downdating scheme is new but inspired by the work by Bojanczyk and Lebak [10, 67]. Finally, the deflation and refinement steps are based on those defined by Stewart for the URV and ULV decompositions [107, 110]. Many aspects of the algorithm are discussed in details, and important considerations are pointed out.

7.2 Plane Rotations

The main computational tool in the ULLV algorithm is the plane rotation, which will be used to introduce zeros selectively into matrices. For this reason, the properties of plane rotations will be described first.

A plane rotation (or Givens rotation) [39, page 201] is an orthogonal transformation that combines two rows or two columns of a matrix. It can be constructed to introduce a zero at any point of the two rows/columns. The plane rotations are rank-two corrections to the identity and if only the real case is considered, they are of the form

$$\mathbf{P} = \begin{array}{cc} \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & \cdots & -s & & \\ & & \vdots & \ddots & \vdots & & \\ & & s & \cdots & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} & \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & i & & j & & \end{array} & \end{array} \tag{7.7}$$

where $c = \cos(\theta)$ and $s = \sin(\theta)$. The multiplication $\mathbf{P}^T\mathbf{x}$ rotates the vector $\mathbf{x}$ through θ radians clockwise in the (i, j) coordinate plane (see Figure 7.1).

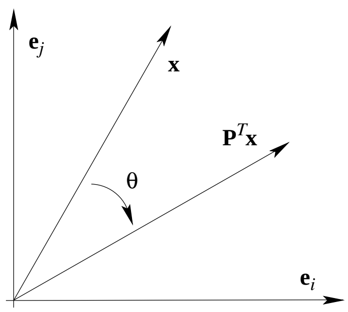


Figure 7.1 Plane rotation.

Let $\mathbf{X}$ denote a matrix, then premultiplication by *left rotations* $\mathbf{P}^T$ operates on the rows (i, j) which cannot change the norm of a column. Likewise, postmultiplication by *right rotations* $\mathbf{Q}$ operates on the columns (i, j) which again cannot change the norm of a row

$$\mathbf{P}^T\mathbf{X} \quad \Rightarrow \quad \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \mathbf{x}_i^T \\ \mathbf{x}_j^T \end{pmatrix} = \begin{pmatrix} c\mathbf{x}_i^T + s\mathbf{x}_j^T \\ -s\mathbf{x}_i^T + c\mathbf{x}_j^T \end{pmatrix} \tag{7.8}$$

$$\mathbf{X}\mathbf{Q} \quad \Rightarrow \quad \begin{pmatrix} \mathbf{x}_i & \mathbf{x}_j \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} c\mathbf{x}_i + s\mathbf{x}_j & -s\mathbf{x}_i + c\mathbf{x}_j \end{pmatrix} \tag{7.9}$$

Plane rotations are clearly orthogonal so multiplication by $\mathbf{P}\mathbf{P}^T = \mathbf{I}$ in between two elements in a matrix decomposition only change the individual matrices in the decomposition and not the outcome, i.e., when the right rotation $\mathbf{P}$ is applied to an element, then the left rotation $\mathbf{P}^T$ is applied to the following element. It is also easy to verify that the matrix 2-norm and the Frobenius norm are invariant with respect to orthogonal transformations

$$\|\mathbf{P}^T\mathbf{X}\mathbf{Q}\|_F = \|\mathbf{X}\|_F \quad \text{and} \quad \|\mathbf{P}^T\mathbf{X}\mathbf{Q}\|_2 = \|\mathbf{X}\|_2 \tag{7.10}$$

A left/right rotation can be constructed to introduce a zero at any point of the two rows/columns. However, the remaining elements are also affected. If X represents large elements, e small elements and 0 elements that are zero, then Equation (7.11) illustrates two rows of a matrix before and after the application of a left rotation, where $\bar{X}$ is the element to be put to zero.

$$\begin{array}{ccc} \text{Before:} & & \text{After:} \\ \begin{array}{ccccccc} \bar{X} & X & X & X & e & e & 0 \\ X & X & e & 0 & e & 0 & 0 \end{array} & \Rightarrow & \begin{array}{ccccccc} 0 & X & X & X & e & e & 0 \\ X & X & X & X & e & e & 0 \end{array} \end{array} \tag{7.11}$$

These facts, except for the first, are rather obvious. For example that a pair of small elements remains small follows from the fact that a left rotation is orthogonal and cannot change the norm of any column to which it is applied. This is essential to preserving the rank-revealing structure of $\mathbf{L}_X$ in the ULLV algorithm.

7.2.1 Computation of c and s

The rotation parameters c and s can be chosen to zero a specified entry in the jth row/column or in the ith row/column. The two cases are given by

$$\text{Case 1:} \quad \begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} a & b \end{pmatrix}\begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} r & 0 \end{pmatrix} \tag{7.12}$$

$$\text{Case 2:} \quad \begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ r \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} a & b \end{pmatrix}\begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} 0 & r \end{pmatrix} \tag{7.13}$$

First consider *case* 1, where the second equation has the solution

$$-sa + cb = 0 \quad \Leftrightarrow \quad c = s\frac{a}{b} \ \text{ and } \ s = c\frac{b}{a} \tag{7.14}$$

which gives two possibilities for substitution in the orthogonality condition $c^2 + s^2 = 1$. The following solution gards against overflow

$$\begin{array}{ll} |a| \geq |b| : & |a| < |b| : \\ c^2 + c^2\left(\frac{b}{a}\right)^2 = 1 & s^2 + s^2\left(\frac{a}{b}\right)^2 = 1 \\ \Downarrow & \Downarrow \\ c = \pm\frac{1}{\sqrt{1+\left(\frac{b}{a}\right)^2}} & s = \pm\frac{1}{\sqrt{1+\left(\frac{a}{b}\right)^2}} \\ s = c\frac{b}{a} & c = s\frac{a}{b} \end{array} \tag{7.15}$$

According to this, the element r in (7.12) also has to be calculated in two ways. Using (7.15), the following solutions are obtained

$$\begin{array}{ll} |a| \geq |b| : & |a| < |b| : \\ r = ca + sb & r = ca + sb \\ \quad = ca\left(1 + \left(\frac{b}{a}\right)^2\right) & \quad = sb\left(1 + \left(\frac{a}{b}\right)^2\right) \\ \quad = a\sqrt{1 + \left(\frac{b}{a}\right)^2} & \quad = b\sqrt{1 + \left(\frac{a}{b}\right)^2} \end{array} \tag{7.16}$$

Case 2 could be treated in the same manner, but by comparing (7.12) and (7.13), the parameters for case 2 can be obtained by using the algorithm for case 1

$$\left.\begin{array}{r} -s_1 a + c_1 b = 0 \\ c_2 a + s_2 b = 0 \end{array}\right\} \quad \Rightarrow \quad \begin{array}{l} c_2 = -s_1 \\ s_2 = c_1 \end{array} \tag{7.17}$$

and

$$r_2 = -s_2 a + c_2 b = -(c_1 a + s_1 b) = -r_1 \tag{7.18}$$

A more convenient way to get the parameters for case 2 is obtained by swapping the elements a and b

$$\text{Case 3:} \quad \begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} b & a \end{pmatrix}\begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} r & 0 \end{pmatrix} \tag{7.19}$$

and then use the algorithm for case 1. By comparing (7.12) and (7.19), the relations for c and s are

$$\left.\begin{aligned} -s_1 a + c_1 b = 0 \\ c_3 a - s_3 b = 0 \end{aligned}\right\} \quad \Rightarrow \quad \begin{aligned} c_3 &= -s_1 = c_2 \\ s_3 &= -c_1 = -s_2 \end{aligned} \tag{7.20}$$

and

$$r_3 = s_3 a + c_3 b = -(c_1 a + s_1 b) = -r_1 = r_2 \tag{7.21}$$

Thus, only a sign shift of the s parameter is necessary to give the desired quantities.

The calculation of c and s requires 2 multiplications, 2 divisions, 1 addition, and 1 squareroot, and the new element r only requires one more multiplication. Thus, the work involved is $\mathcal{O}(1)$. The application of the plane rotation to a row/column with n elements is more expensive. The work is here $4n$ multiplications and $2n$ additions, which is $\mathcal{O}(n)$.

7.3 Updating the ULLV Decomposition

In this section, it is examined how to update the ULLV decomposition (7.1-7.2) efficiently when new rows of data samples are added to $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$ [73, 93]. In the updating step a forgetting factor β is incorporated, where $\beta \in [0;1]$. This widely used method is called *exponential windowing* and the effect is to suppress the older data samples, so that they have less and less influence as the number of rows m increase

$$\mathbf{X}^{(m)} = \begin{pmatrix} \beta^{m-1}\mathbf{x}_1^T \\ \vdots \\ \beta\mathbf{x}_{m-1}^T \\ \mathbf{x}_m^T \end{pmatrix} \quad \text{and} \quad \mathbf{N}^{(m)} = \begin{pmatrix} \beta^{m-1}\mathbf{n}_1^T \\ \vdots \\ \beta\mathbf{n}_{m-1}^T \\ \mathbf{n}_m^T \end{pmatrix} \tag{7.22}$$

Thus, when new rows are added, the updated matrix is obtained as

$$\mathbf{X}^{(m+1)} = \begin{pmatrix} \beta\mathbf{X}^{(m)} \\ \mathbf{x}_{m+1}^T \end{pmatrix} \quad \text{and} \quad \mathbf{N}^{(m+1)} = \begin{pmatrix} \beta\mathbf{N}^{(m)} \\ \mathbf{n}_{m+1}^T \end{pmatrix} \tag{7.23}$$

In signal processing, $\mathbf{X}$ and $\mathbf{N}$ are often Toeplitz structured matrices where a new row, for the case of $\mathbf{X}$, is constructed from a new data sample x_k and $n-1$ previous samples. To keep the Toeplitz structure in the updated matrix, it is necessary to weight the new row

$$\mathbf{x}_{m+1}^T = \begin{pmatrix} x_k & x_{k-1}\beta & \cdots & x_{k-n+2}\beta^{n-2} & x_{k-n+1}\beta^{n-1} \end{pmatrix} \tag{7.24}$$

which results in the following matrix for dimensions $m = 6$ and $n = 5$

$$\begin{aligned} \mathbf{X}^{(m+1)} &= \begin{pmatrix} \beta\mathbf{X}^{(m)} \\ \mathbf{x}_{m+1}^T \end{pmatrix} \qquad\qquad (7.25) \\ &= \begin{pmatrix} x_{k-6}\beta^6 & x_{k-7}\beta^7 & x_{k-8}\beta^8 & x_{k-9}\beta^9 & x_{k-10}\beta^{10} \\ x_{k-5}\beta^5 & x_{k-6}\beta^6 & x_{k-7}\beta^7 & x_{k-8}\beta^8 & x_{k-9}\beta^9 \\ x_{k-4}\beta^4 & x_{k-5}\beta^5 & x_{k-6}\beta^6 & x_{k-7}\beta^7 & x_{k-8}\beta^8 \\ x_{k-3}\beta^3 & x_{k-4}\beta^4 & x_{k-5}\beta^5 & x_{k-6}\beta^6 & x_{k-7}\beta^7 \\ x_{k-2}\beta^2 & x_{k-3}\beta^3 & x_{k-4}\beta^4 & x_{k-5}\beta^5 & x_{k-6}\beta^6 \\ x_{k-1}\beta & x_{k-2}\beta^2 & x_{k-3}\beta^3 & x_{k-4}\beta^4 & x_{k-5}\beta^5 \\ x_k & x_{k-1}\beta & x_{k-2}\beta^2 & x_{k-3}\beta^3 & x_{k-4}\beta^4 \end{pmatrix} \end{aligned}$$

The exponential window is then connected with the time sequences and not the rows of the matrices. For β close to one, the exponential window length W is normally defined as

$$\beta^W = \frac{1}{e} \quad \Rightarrow \quad W = -\frac{1}{\ln(\beta)} \approx \frac{1}{1-\beta} \quad \text{for} \quad \beta \approx 1 \tag{7.26}$$

If the window length W is chosen smaller than the row dimension m, the elements in the top rows of $\mathbf{X}$ and $\mathbf{N}$ are insignificant and will not be represented in $\mathbf{L}_X$ and $\mathbf{L}$. Therefore, a top row of $\mathbf{U}_X$ and $\mathbf{U}_N$ can be removed after each update. This allows us to look at the local behavior of an arbitrarily long sequence of data.

First consider the update of $\mathbf{X}$ and afterwards the update of $\mathbf{N}$ using the notation

$$\text{Case X:} \qquad \mathbf{X} = \mathbf{X}^{(m)},\ \hat{\mathbf{X}} = \mathbf{X}^{(m+1)} \quad \text{and} \quad \mathbf{N} = \mathbf{N}^{(m)} \tag{7.27}$$

$$\text{Case N:} \qquad \mathbf{N} = \mathbf{N}^{(m)},\ \hat{\mathbf{N}} = \mathbf{N}^{(m+1)} \quad \text{and} \quad \mathbf{X} = \mathbf{X}^{(m)} \tag{7.28}$$

7.3.1 Adding a Row to X

Given the ULLV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$, the initial decomposition with a new row $\mathbf{x}^T$ added to $\mathbf{X}$ using a forgetting factor β are an augmented system of the form

$$\hat{\mathbf{X}} = \begin{pmatrix} \beta\mathbf{X} \\ \mathbf{x}^T \end{pmatrix} = \begin{pmatrix} \mathbf{U}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \beta\mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_X \boldsymbol{\mathcal{L}}_X \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.29}$$

$$\mathbf{N} = \mathbf{U}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_N \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.30}$$

Using only plane rotations, the updating procedure reconstruct the lower triangularity of $\mathbf{L}$ and update $\mathbf{L}_X$, so its rank revealing structure is retained. The rotations will be accumulated into three orthogonal transformations $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{J}$, and applied to the factors in (7.29) and (7.30) as follows

$$\hat{\mathbf{X}} = (\boldsymbol{\mathcal{U}}_X\mathbf{P})(\mathbf{P}^T\boldsymbol{\mathcal{L}}_X\mathbf{Q})(\mathbf{Q}^T\boldsymbol{\mathcal{L}}\mathbf{J})(\mathbf{J}^T\mathbf{V}^T) \tag{7.31}$$

$$\mathbf{N} = (\boldsymbol{\mathcal{U}}_N\mathbf{Q})(\mathbf{Q}^T\boldsymbol{\mathcal{L}}\mathbf{J})(\mathbf{J}^T\mathbf{V}^T) \tag{7.32}$$

where $\mathbf{P}^T\boldsymbol{\mathcal{L}}_X\mathbf{Q}$ reveals the numerical rank of $\hat{\mathbf{X}}\mathbf{N}^+$ and $\mathbf{Q}^T\boldsymbol{\mathcal{L}}\mathbf{J}$ is lower triangular. The update process is split up into three steps and illustrated by means of an example with matrix dimensions $m = 6$ and $n = 5$, which gives the initial matrices

$$\boldsymbol{\mathcal{L}}_X = \begin{pmatrix} \beta\mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ f & f & g & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\mathcal{L}} = \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ z & z & z & z & z \end{pmatrix}$$

$$\boldsymbol{\mathcal{U}}_X = \begin{pmatrix} \mathbf{U}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

Step X1. First eliminate all but the first element of the row $\mathbf{x}^T\mathbf{V}$, maintaining the rank revealing structure of $\mathbf{L}_X$ as much as possible. The following three substeps describes the elimination and are done in a loop for $i = n, n-1, \ldots, 2$, where rotations are operating on neighboring columns or rows $(i-1, i)$.

Step X1-1. Annihilate the ith element by postmultiply $\boldsymbol{\mathcal{L}}$ with a rotation $\mathbf{J}$ in the plane $(i-1, i)$, which introduces a nonzero $(i-1, i)$ entry z in $\mathbf{L}$

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \mathbf{J}\mathbf{J}^T\mathbf{V}^T = \begin{pmatrix} \mathbf{L}' \\ \xi\mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.33}$$
$$= \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & l & l & l \\ \xi & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

Step X1-2. Eliminate the superdiagonal of $\mathbf{L}'$ by applying from the left a rotation $\mathbf{Q}^T$ in the plane $(i-1, i)$, which introduces a nonzero $(i-1, i)$ entry in $\mathbf{L}_X$. This rotation must be propagated to the left in both decompositions (7.29) and (7.30)

$$\begin{pmatrix} \beta\mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L}' \\ \xi\mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi\mathbf{e}_1^T \end{pmatrix} \tag{7.34}$$
$$= \begin{pmatrix} l & z & & & & 0 \\ l & l & z & & & 0 \\ f & f & g & z & & 0 \\ f & f & g & g & z & 0 \\ f & f & g & g & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ \xi & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{U}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L}' \\ \xi\mathbf{e}_1^T \end{pmatrix} = \mathbf{U}_N\bar{\mathbf{Q}} \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \mathbf{Q}^T \begin{pmatrix} \mathbf{L}' \\ \xi\mathbf{e}_1^T \end{pmatrix} \tag{7.35}$$
$$= \bar{\mathbf{U}}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi\mathbf{e}_1^T \end{pmatrix}$$

where $\mathbf{Q}$ can be directly incorporated into $\mathbf{U}_N$ because only the $\mathbf{I}_n$ part of the second factor is affected.

Step X1-3. Eliminate the superdiagonal of $\mathbf{L}'_X$ by applying from the left a rotation $\mathbf{P}^T$ in the plane $(i-1,i)$

$$\begin{pmatrix} \mathbf{U}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{P}\mathbf{P}^T \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{U}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{7.36}$$

$$= \begin{pmatrix} u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

After the last step, the dimensions of $\mathbf{L}_{X1}$ in (6.53) will *increase by one* because of the linear combination of a row with large elements and a row with small elements. Thus, the numerical rank of $\bar{\mathbf{L}}_X$ may increase by one over that of $\mathbf{L}_X$. The decomposition is now

$$\hat{\mathbf{X}} = \begin{pmatrix} \bar{\mathbf{U}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.37}$$

$$\mathbf{N} = \bar{\mathbf{U}}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.38}$$

Step X2. In the second step eliminate the element ξ using a scaled rotation $\mathbf{Y}$ from the left in plane $(1,n+1)$. The matrix $\mathbf{Y}$ and its inverse are given by

$$\mathbf{Y} = \begin{pmatrix} c_2^2 & & & & & -c_2 s_2 \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ c_2 s_2 & & & & & c_2^2 \end{pmatrix}, \qquad \mathbf{Y}^{-1} = \begin{pmatrix} 1 & & & & & \frac{s_2}{c_2} \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ -\frac{s_2}{c_2} & & & & & 1 \end{pmatrix} \tag{7.39}$$

When the scaled rotation is applied from the left, the solution is chosen to be

$$\mathbf{Y} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \tag{7.40}$$

which explicitly determines the parameters s_2 and c_2

$$c_2 \begin{pmatrix} c_2 & -s_2 \\ s_2 & c_2 \end{pmatrix} \begin{pmatrix} \bar{l}_{11} \mathbf{e}_1^T \\ \xi \mathbf{e}_1^T \end{pmatrix} = c_2 \begin{pmatrix} (c_2 \bar{l}_{11} - s_2 \xi) \mathbf{e}_1^T \\ (s_2 \bar{l}_{11} + c_2 \xi) \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \bar{l}_{11} \mathbf{e}_1^T \\ \mathbf{0}^T \end{pmatrix} \Rightarrow \tag{7.41}$$

$$c_2 = \frac{\bar{l}_{11}}{\sqrt{|\xi|^2 + |\bar{l}_{11}|^2}} \quad \text{and} \quad s_2 = \frac{-\xi}{\sqrt{|\xi|^2 + |\bar{l}_{11}|^2}} \tag{7.42}$$

A scaled rotation is used, so that when the transformation $\mathbf{Y}^{-1}$ is propagated to the left, the leading $n \times n$ submatrix of the second factor in both decompositions (7.37) and (7.38) remains unchanged

$$\begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{Y}^{-1} = \begin{pmatrix} \bar{\mathbf{L}}_X & -\eta \bar{\mathbf{l}}_{X1} \\ \eta \mathbf{e}_1^T & 1 \end{pmatrix} = \begin{pmatrix} l_{11} & & & & & \frac{s_2}{c_2} l_{11} \\ l_{21} & l & & & & \frac{s_2}{c_2} l_{21} \\ l_{31} & l & l & & & \frac{s_2}{c_2} l_{31} \\ f_{41} & f & g & g & & \frac{s_2}{c_2} f_{41} \\ f_{51} & f & g & g & g & \frac{s_2}{c_2} f_{51} \\ -\frac{s_2}{c_2} & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7.43}$$

and

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \mathbf{Y}^{-1} = \begin{pmatrix} \mathbf{I}_n & -\eta \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \frac{s_2}{c_2} \\ & 1 & & & & 0 \\ & & 1 & & & 0 \\ & & & 1 & & 0 \\ & & & & 1 & 0 \end{pmatrix} \tag{7.44}$$

The transformation creates many nonzero elements in the last column of the second factor in both decompositions, but fortunately the last column of these two matrices can be dropped because the third factor now has a zero row at the bottom. Consequently the elimination of the element ξ introduces only one new element η given by

$$\eta = \frac{-s_2}{c_2} = \frac{\xi}{\bar{l}_{11}} \tag{7.45}$$

Hence, step X2 can be implemented using only one division, and the new decomposition is

$$\hat{\mathbf{X}} = \begin{pmatrix} \bar{\mathbf{U}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}}_X \\ \eta \mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{L}} \bar{\mathbf{V}}^T \tag{7.46}$$

$$\mathbf{N} = \bar{\mathbf{U}}_N \bar{\mathbf{L}} \bar{\mathbf{V}}^T \tag{7.47}$$

Step X3. In the third step eliminate the element η using a rotation $\mathbf{K}^T$ from the left in plane $(1, n+1)$

$$\begin{pmatrix} \bar{\mathbf{U}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{K}\mathbf{K}^T \begin{pmatrix} \bar{\mathbf{L}}_X \\ \eta \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{U}}_X & \mathbf{z} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}}_X \\ \mathbf{0}^T \end{pmatrix} \tag{7.48}$$

$$= \begin{pmatrix} c_3 u_{11} & u & u & u & u & -s_3 u_{11} \\ c_3 u_{21} & u & u & u & u & -s_3 u_{21} \\ c_3 u_{31} & u & u & u & u & -s_3 u_{31} \\ c_3 u_{41} & u & u & u & u & -s_3 u_{41} \\ c_3 u_{51} & u & u & u & u & -s_3 u_{51} \\ c_3 u_{61} & u & u & u & u & -s_3 u_{61} \\ s_3 & 0 & 0 & 0 & 0 & c_3 \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ f & f & g & g & \\ f & f & g & g & g \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Finally, the last column of the first factor can be dropped because the second factor now has a bottom row of zeros. The updated decomposition is

$$\hat{\mathbf{X}} = \hat{\mathbf{U}}_X \hat{\mathbf{L}}_X \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.49}$$

$$\mathbf{N} = \hat{\mathbf{U}}_N \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.50}$$

Note that the row dimension of $\mathbf{U}_X$ has increased by one, and that $\mathbf{U}_X$, $\mathbf{U}_N$ and $\mathbf{V}$ are not needed for the construction of plane rotations.

7.3.2 Adding a Row to N

When a new row $\mathbf{n}^T$ is added to $\mathbf{N}$, the forgetting factor β is associated with $\mathbf{L}$. Given the ULLV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$, the initial decomposition are an augmented system of the form

$$\mathbf{X} = \mathbf{U}_X \begin{pmatrix} \beta^{-1}\mathbf{L}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T = \mathbf{U}_X \boldsymbol{\mathcal{L}}_X \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.51}$$

$$\hat{\mathbf{N}} = \begin{pmatrix} \beta\mathbf{N} \\ \mathbf{n}^T \end{pmatrix} = \begin{pmatrix} \mathbf{U}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_N \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.52}$$

As in the updating of $\mathbf{X}$, the lower triangularity of $\mathbf{L}$ is reconstructed and $\mathbf{L}_X$ is updated, so its rank revealing structure is retained, where the accumulated plane rotations are applied to the factors as in (7.31) and (7.32).

In this case, the update process is split up into two steps and also here illustrated by means of an example with matrix dimensions $m = 6$ and $n = 5$, which gives the initial matrices

$$\boldsymbol{\mathcal{L}}_X = \begin{pmatrix} \frac{\mathbf{L}_X}{\beta} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ f & f & g & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \end{pmatrix}, \quad \boldsymbol{\mathcal{L}} = \begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ z & z & z & z & z \end{pmatrix}$$

$$\boldsymbol{\mathcal{U}}_N = \begin{pmatrix} \mathbf{U}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

Step N1. As in the updating of $\mathbf{X}$, first eliminate all but the first element of the row $\mathbf{n}^T\mathbf{V}$, maintaining the rank revealing structure of $\mathbf{L}_X$ as much as possible. The following three substeps describes the elimination and is done in a loop for $i = n, n-1, \ldots, 2$, where rotations are operating on neighboring columns or rows $(i-1, i)$.

Step N1-1. Annihilate the ith element by postmultiply $\boldsymbol{\mathcal{L}}$ with a rotation $\mathbf{J}$ in the plane $(i-1, i)$, which introduces a nonzero $(i-1, i)$ entry z in $\mathbf{L}$

$$\begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} \mathbf{J}\mathbf{J}^T\mathbf{V}^T = \begin{pmatrix} \mathbf{L}' \\ \xi\mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.53}$$

$$= \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & l & l & l \\ \xi & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

Step N1-2. Eliminate the superdiagonal of $\mathbf{L}'$ by applying from the left a rotation $\mathbf{Q}^T$ in the plane $(i-1,i)$, which introduces a nonzero $(i-1,i)$ entry in $\mathbf{L}_X$. This rotation must be propagated to the left in both decompositions (7.51) and (7.52)

$$\begin{pmatrix} \beta^{-1}\mathbf{L}_X & \mathbf{0} \end{pmatrix} \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L}' \\ \xi \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \tag{7.54}$$

$$= \begin{pmatrix} l & z & & & & 0 \\ l & l & z & & & 0 \\ f & f & g & z & & 0 \\ f & f & g & g & z & 0 \\ f & f & g & g & g & 0 \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ \xi & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{U}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L}' \\ \xi \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{U}}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \tag{7.55}$$

Step N1-3. Eliminate the superdiagonal of $\mathbf{L}'_X$ by applying from the left a rotation $\mathbf{P}^T$ in the plane $(i-1,i)$

$$\mathbf{U}_X \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \end{pmatrix} = \bar{\mathbf{U}}_X \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \tag{7.56}$$

$$= \begin{pmatrix} u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \end{pmatrix} \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \end{pmatrix}$$

After the last step, the dimensions of $\mathbf{L}_{X1}$ in (6.53) will *increase by one* because of the linear combination of a row with large elements and a row with small elements. Thus, the numerical rank of $\bar{\mathbf{L}}_X$ may increase by one over that of $\mathbf{L}_X$. The decomposition is now

$$\mathbf{X} = \bar{\mathbf{U}}_X \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.57}$$

$$\hat{\mathbf{N}} = \begin{pmatrix} \bar{\mathbf{U}}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.58}$$

Step N2. In the second step eliminate the element ξ using a rotation $\mathbf{K}^T$ from the left in plane $(1, n+1)$

$$\begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \mathbf{K}\mathbf{K}^T \begin{pmatrix} \bar{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{x} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \tag{7.59}$$

$$= \begin{pmatrix} c_2 l_{11} & & & & & -s_2 l_{11} \\ c_2 l_{21} & l & & & & -s_2 l_{21} \\ c_2 l_{31} & l & l & & & -s_2 l_{31} \\ c_2 f_{41} & f & g & g & & -s_2 f_{41} \\ c_2 f_{51} & f & g & g & g & -s_2 f_{51} \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \bar{\mathbf{U}}_N & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{K} = \begin{pmatrix} \hat{\mathbf{U}}_N & \mathbf{x} \end{pmatrix} = \begin{pmatrix} c_2 u_{11} & u & u & u & u & -s_2 u_{11} \\ c_2 u_{21} & u & u & u & u & -s_2 u_{21} \\ c_2 u_{31} & u & u & u & u & -s_2 u_{31} \\ c_2 u_{41} & u & u & u & u & -s_2 u_{41} \\ c_2 u_{51} & u & u & u & u & -s_2 u_{51} \\ c_2 u_{61} & u & u & u & u & -s_2 u_{61} \\ s_2 & 0 & 0 & 0 & 0 & c_2 \end{pmatrix} \tag{7.60}$$

Finally, the last column of the first factor can be dropped because the second factor now has a zero bottom row. The updated decomposition is

$$\mathbf{X} = \hat{\mathbf{U}}_X \hat{\mathbf{L}}_X \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.61}$$

$$\hat{\mathbf{N}} = \hat{\mathbf{U}}_N \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.62}$$

Similar to the updating of $\mathbf{X}$, the row dimension of $\mathbf{U}_N$ has increased by one, and again $\mathbf{U}_X$, $\mathbf{U}_N$ and $\mathbf{V}$ are not needed for the construction of plane rotations.

7.4 Downdating the ULLV Decomposition

An alternative to the exponential window is a *rectangular window (sliding window)* corresponding to the case of $\beta = 1$. Thus, when new rows of data samples are added to the bottom of the matrices $\mathbf{X}$ and $\mathbf{N}$, the signal energy is accumulated in $\mathbf{L}_X$ and $\mathbf{L}$ after the update step. To remove old samples in the top rows of $\mathbf{X}$ and $\mathbf{N}$, it is necessary to identify and remove the matching energi in the decomposition, which is done in a downdating step [10, 67, 85]. Consider the partitioning of $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{N} \in \mathbb{R}^{m \times n}$

$$\mathbf{X} = \mathbf{U}_X \mathbf{L}_X \mathbf{L} \mathbf{V}^T = \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} \quad \text{and} \quad \mathbf{N} = \mathbf{U}_N \mathbf{L} \mathbf{V}^T = \begin{pmatrix} \mathbf{n}^T \\ \hat{\mathbf{N}} \end{pmatrix} \tag{7.63}$$

then the problem of downdating the ULLV decomposition is to find the ULLV decomposition of $\hat{\mathbf{X}} \in \mathbb{R}^{(m-1) \times n}$ and $\hat{\mathbf{N}} \in \mathbb{R}^{(m-1) \times n}$ given the ULLV decomposition of $\mathbf{X}$ and $\mathbf{N}$.

Clearly, the sliding window method can track the change in the signal statistics more accurately than the exponential window method when there is an abrupt change in data.

Downdating can be viewed as the reverse process to updating the ULLV decomposition. In the updating process, the last row of $\mathbf{U}_X$ or $\mathbf{U}_N$ corresponding to the added row, which is initially external to the decomposition, is transformed to become an integral part of it. In the downdating process, therefore, the decomposition is transformed into a form where the first row of $\mathbf{U}_X$ or $\mathbf{U}_N$, which corresponding to the row to be deleted, is no longer an integral part of the decomposition. Thus, assume that a row is appended to the top of $\mathbf{X}$ or $\mathbf{N}$ – this is because rows are removed from the top – then the downdating follow the updating steps backwards.

There are two main steps in the ULLV downdating procedure, which will be called the *expansion* step and the *downdating* step, in accordance with the terms used in [67]. In the expansion step, the matrix $\mathbf{U}_X$ or $\mathbf{U}_N$ is extended with a new column, where the method depend on whether or not the matrix is maintained. In the downdating step, a sequence of plane rotations is used to produce the downdated decomposition.

7.4.1 Adding a Row to the Top of X and N

To follow the updating steps backwards, update of $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$ from the top is examined where it differs from the bottom case.

The initial decomposition with a new row $\mathbf{x}^T$ added to the top of $\mathbf{X}$ instead of the bottom is

$$\hat{\mathbf{X}} = \begin{pmatrix} \mathbf{x}^T \\ \beta\mathbf{X} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \mathbf{U}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \beta\mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T \tag{7.64}$$

$$\mathbf{N} = \mathbf{U}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T \tag{7.65}$$

which only differs in the last step X3 (7.48) compared to updating from bottom. The new step X3 becomes

$$\begin{pmatrix} \mathbf{0}^T & 1 \\ \bar{\mathbf{U}}_X & \mathbf{0} \end{pmatrix} \mathbf{K}\mathbf{K}^T \begin{pmatrix} \bar{\mathbf{L}}_X \\ \eta\mathbf{e}_1^T \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{U}}_X & \mathbf{z} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}}_X \\ \mathbf{0}^T \end{pmatrix} \tag{7.66}$$

$$= \begin{pmatrix} s_3 & 0 & 0 & 0 & 0 & c_3 \\ c_3u_{11} & u & u & u & u & -s_3u_{11} \\ c_3u_{21} & u & u & u & u & -s_3u_{21} \\ c_3u_{31} & u & u & u & u & -s_3u_{31} \\ c_3u_{41} & u & u & u & u & -s_3u_{41} \\ c_3u_{51} & u & u & u & u & -s_3u_{51} \\ c_3u_{61} & u & u & u & u & -s_3u_{61} \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ f & f & g & g & \\ f & f & g & g & g \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As before, the last column of the first factor can be dropped because the second factor now has a bottom row of zeros.

For the case of $\mathbf{N}$, the initial decomposition with a new row $\mathbf{n}^T$ added to the top instead of the bottom is

$$\mathbf{X} = \mathbf{U}_X \begin{pmatrix} \beta^{-1}\mathbf{L}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T \tag{7.67}$$

$$\hat{\mathbf{N}} = \begin{pmatrix} \mathbf{n}^T \\ \beta\mathbf{N} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \mathbf{U}_N & \mathbf{0} \end{pmatrix} \begin{pmatrix} \beta\mathbf{L} \\ \mathbf{n}^T\mathbf{V} \end{pmatrix} \mathbf{V}^T \tag{7.68}$$

which only differs in the last step N2 (7.60) compared to updating from bottom. The new step N2 become

$$\begin{pmatrix} \mathbf{0}^T & 1 \\ \bar{\mathbf{U}}_N & \mathbf{0} \end{pmatrix} \mathbf{K} = \begin{pmatrix} \hat{\mathbf{U}}_N & \mathbf{z} \end{pmatrix} = \begin{pmatrix} s_2 & 0 & 0 & 0 & 0 & c_2 \\ c_2u_{11} & u & u & u & u & -s_2u_{11} \\ c_2u_{21} & u & u & u & u & -s_2u_{21} \\ c_2u_{31} & u & u & u & u & -s_2u_{31} \\ c_2u_{41} & u & u & u & u & -s_2u_{41} \\ c_2u_{51} & u & u & u & u & -s_2u_{51} \\ c_2u_{61} & u & u & u & u & -s_2u_{61} \end{pmatrix} \tag{7.69}$$

Again, the last column of the first factor can be dropped because the second factor now has a zero bottom row.

To reverse this process, it is necessary to transform $\mathbf{U}_X$ in (7.63) to the form in (7.66) and $\mathbf{U}_N$ in (7.63) to the form in (7.69), which has *one column more* than the target matrices $\hat{\mathbf{U}}_X$ and $\hat{\mathbf{U}}_N$, the columns are *orthogonal* and the first row has *norm one* and is orthogonal to the other rows. This is done in the expansion step as explained in the next section for the general case of a matrix $\mathbf{U}$.

7.4.2 Expansion Step Methods

The objective of the expansion step is to obtain a new matrix $\mathbf{U}' \in \mathbb{R}^{m\times(n+1)}$ from $\mathbf{U} \in \mathbb{R}^{m\times n}$ in the form

$$\mathbf{U}' = \begin{pmatrix} \mathbf{U} & \mathbf{q} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^T & q_1 \\ \bar{\mathbf{U}} & \bar{\mathbf{q}} \end{pmatrix} \tag{7.70}$$

where $\mathbf{q} \in \mathbb{R}^{m\times 1}$ and $\mathbf{u}_1 \in \mathbb{R}^{n\times 1}$. The first row of $\mathbf{U}'$ should have norm one and be orthogonal to the other rows, and the matrix should still have orthogonal columns.

To obtain the added column $\mathbf{q}$, an initial vector $\mathbf{a} \in \mathbb{R}^{m\times 1}$ is orthogonalized to $\mathbf{U}$, where the vector $\mathbf{a}$ is constrained by the above requirements to $\mathbf{U}'$. First, calculate the projection $\mathbf{t} \in \mathbb{R}^{m\times 1}$ onto the subspace spanned by the orthogonal complement $\tilde{\mathbf{U}} \in \mathbb{R}^{m\times(m-n)}$

$$\mathbf{t} = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{a} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{a} \tag{7.71}$$

$$\|\mathbf{t}\|_2^2 = \mathbf{t}^T\mathbf{t} = \mathbf{a}^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{a} = \|\tilde{\mathbf{U}}^T\mathbf{a}\|_2^2 \tag{7.72}$$

After normalization, the new column $\mathbf{q}$ is

$$\mathbf{q} = \frac{\mathbf{t}}{\|\mathbf{t}\|_2} = \frac{\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{a}}{\|\tilde{\mathbf{U}}^T\mathbf{a}\|_2}\,, \qquad q_1 = \frac{\tilde{\mathbf{u}}_1^T\tilde{\mathbf{U}}^T\mathbf{a}}{\|\tilde{\mathbf{U}}^T\mathbf{a}\|_2} \tag{7.73}$$

The norm constraint on the first row of $\mathbf{U}'$ gives

$$\|\mathbf{u}_1^T\|_2^2 + q_1^2 = \mathbf{u}_1^T\mathbf{u}_1 + \frac{\mathbf{a}^T\tilde{\mathbf{U}}\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T\tilde{\mathbf{U}}^T\mathbf{a}}{\mathbf{a}^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{a}} = 1 \tag{7.74}$$

or

$$1 - \mathbf{u}_1^T\mathbf{u}_1 = \tilde{\mathbf{u}}_1^T\tilde{\mathbf{u}}_1 = \frac{\mathbf{y}^T(\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T)\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\,, \qquad \mathbf{y} = \tilde{\mathbf{U}}^T\mathbf{a} \tag{7.75}$$

This is the Rayleigh quotient of the matrix $\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T$, which has only one non-zero eigenvalue $\lambda_{max}(\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T) = \tilde{\mathbf{u}}_1^T\tilde{\mathbf{u}}_1$. The vector $\mathbf{y}$ is therefore the corresponding eigenvector

$$\mathbf{y} = \tilde{\mathbf{U}}^T\mathbf{a} = \frac{\tilde{\mathbf{u}}_1}{\|\tilde{\mathbf{u}}_1\|_2}\, c \tag{7.76}$$

where c is a constant. Because $m > n$, Equation (7.76) is an underdetermined system with an infinite number of solutions for the vector $\mathbf{a}$, but there is only one set of solutions independent of the unknown matrix $\tilde{\mathbf{U}}$

$$\mathbf{a} = c'\mathbf{e}_1\,, \qquad c' = \frac{c}{\|\tilde{\mathbf{u}}_1\|_2} \tag{7.77}$$

Normally, the vector $\mathbf{a}$ is chosen to be $\mathbf{e}_1$. If $\mathbf{e}_1$ is linearly dependent on the columns of $\mathbf{U}$ then mathematically $\mathbf{t} = \mathbf{0}$, leaving a handful of rounding errors in place of $\mathbf{t}$, and the orthogonalization will fail. In this case, $\|\mathbf{u}_1^T\|_2 = 1$ and q_1 must be zero. Thus, a random vector $\mathbf{d} \in \mathbb{R}^{(m-1)\times 1}$ can be orthogonalized to the matrix $\bar{\mathbf{U}}$ (see Daniel et al. [21]).

The corresponding numerical problem is more subtle. If $\|\mathbf{t}\|_2/\|\mathbf{e}_1\|_2$ is small, then numerical cancellation has occurred in forming $\mathbf{t}$, and $\mathbf{t}$, as well as $\mathbf{q}$, are likely to be inaccurate relative to their lengths. The vector $\mathbf{t}$ could now be corrected by an iterative reorthogonalization process [21], where the rounding errors would force some iterate $\mathbf{t}^{(k)}$ to have substantial components orthogonal to the range of $\mathbf{U}$, i.e., $\|\mathbf{t}^{(k)}\|_2/\|\mathbf{t}^{(k-1)}\|_2$ is not to small. However, the following algorithm relies on the fact that one reorthogonalization is always enough to compute a vector $\mathbf{q}$ which is orthogonal to $\mathbf{U}$ to working accuracy (see Parlett [92, page 107]).

ALGORITHM 7.1 (Expansion)

$\mathbf{t} = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{e}_1$
if $\|\mathbf{t}\|_2 \le \frac{\|\mathbf{e}_1\|_2}{\kappa}$ then
  $\mathbf{t}' = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{t}$
  if $\|\mathbf{t}'\|_2 \le \frac{\|\mathbf{t}\|_2}{\kappa}$ then
    $\mathbf{d} = \frac{1}{m-1}(0\,1\,1\cdots 1)^T$
    $\mathbf{t}'' = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{d}$
    $\mathbf{q} = \frac{\mathbf{t}''}{\|\mathbf{t}''\|_2}$
  else $\mathbf{q} = \frac{\mathbf{t}'}{\|\mathbf{t}'\|_2}$
else $\mathbf{q} = \frac{\mathbf{t}}{\|\mathbf{t}\|_2}$

The size of the quantity $\|\mathbf{U}^T\mathbf{q}\|_2$ depends on the parameter κ satisfying $\kappa \ge 1$, for instance $\kappa = \sqrt{2}$ is often used corresponding to a projection angle of $\pi/4$. Another issue is to identify when the orhogonalization will fail, i.e., $q_1 = 0$, as given by the following theorem

THEOREM 7.1 (*Expansion and Rank Decrease*) *In the downdating problem*

$$\mathbf{e}_1 \in \langle \mathbf{U} \rangle \Leftrightarrow \|\mathbf{u}_1^T\|_2 = 1 \tag{7.78}$$

if the rank of the downdated matrix decreases.

PROOF. [1] For the case of matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the ULLV decomposition is

$$\mathbf{X} = \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{V}^T = \begin{pmatrix} \mathbf{x}^T \\ \bar{\mathbf{X}} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{X1}^T \\ \bar{\mathbf{U}}_X \end{pmatrix} \mathbf{L}_X\mathbf{L}\mathbf{V}^T \tag{7.79}$$

Assume $\mathrm{rank}(\mathbf{X}) = p \le n$ and let the rank decrease, i.e.,

$$\mathrm{rank}(\bar{\mathbf{X}}) = p - 1 \tag{7.80}$$

$$\mathrm{rank}(\bar{\mathbf{U}}_X) \le n - 1 \tag{7.81}$$

Using the CS decomposition of the matrix pair $(\bar{\mathbf{U}}_X, \mathbf{u}_{X1}^T)$ gives

$$\begin{pmatrix} \bar{\mathbf{U}}_X \\ \mathbf{u}_{X1}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{U}}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{U}}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C} \\ \mathbf{S} \end{pmatrix} \boldsymbol{\mathcal{V}}^T \tag{7.82}$$

where $\boldsymbol{\mathcal{U}}_1 \in \mathbb{R}^{(m-1)\times(m-1)}$, $\boldsymbol{\mathcal{U}}_2 \in \mathbb{R}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n\times n}$ are orthogonal matrices and

$$\mathbf{C} = \mathrm{diag}(c, 1, \ldots, 1) \in \mathbb{R}^{(m-1)\times n} \tag{7.83}$$

$$\mathbf{S} = (s, 0, \ldots, 0) \in \mathbb{R}^{1\times n} \tag{7.84}$$

$$\mathbf{I}_n = \mathbf{C}^T\mathbf{C} + \mathbf{S}^T\mathbf{S} \tag{7.85}$$

From the rank deficiency of $\bar{\mathbf{U}}_X$, we conclude that $c = 0 \Leftrightarrow \mathrm{rank}(\bar{\mathbf{U}}_X) = n - 1$ giving $s = 1 \Leftrightarrow \|\mathbf{u}_{X1}^T\|_2 = 1$. Hence, we have proved the above relation. □

Downdating methods may be divided into two classes based on whether or not they maintain the matrix $\mathbf{U}$. If $\mathbf{U}$ is not maintained, then the row $\mathbf{u}_1^T$ and the extra element q_1 must be calculated in order to determine the inverse update rotations. Different expansion methods are used for the two classes. The first, *Modified Gram-Schmidt with Re-orthogonalization*, assumes the left side orthogonal matrices are maintained. The second, the *method of Corrected Semi-Normal Equations*, assumes that these matrices are not maintained.

[1] Due to Per Christian Hansen.

7.4.2.1 Modified Gram-Schmidt with Re-orthogonalization

The Modified Gram-Schmidt (MGS) process orthogonalizes the new initial column $\mathbf{e}_1$ against the columns of $\mathbf{U}$, but because of the orthogonality of $\mathbf{U}$, the computational procedure in Golub and Van Loan [39, p. 218] can be simplified.

The QR-factorization of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is given by

$$\mathbf{A} = \mathbf{QR} = \begin{pmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \vdots \\ \mathbf{r}_n^T \end{pmatrix} = \sum_{i=1}^{n} \mathbf{q}_i \mathbf{r}_i^T \tag{7.86}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times n}$ has orthogonal columns and $\mathbf{R} \in \mathbb{R}^{n\times n}$ is upper triangular. In the kth step of MGS, the kth column of $\mathbf{Q}$ (denoted by $\mathbf{q}_k$) and the kth row of $\mathbf{R}$ (denoted by $\mathbf{r}_k^T$) are determined. From the triangular structure of $\mathbf{R}$ and (7.86) is obtained

$$\mathbf{A} - \sum_{i=1}^{k-1} \mathbf{q}_i \mathbf{r}_i^T = \sum_{i=k}^{n} \mathbf{q}_i \mathbf{r}_i^T = \begin{pmatrix} \mathbf{0} & \mathbf{A}^{(k)} \end{pmatrix} \tag{7.87}$$

where

$$\mathbf{A}^{(k)} = \begin{pmatrix} r_{kk}\mathbf{q}_k & \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{z} & \mathbf{B} \end{pmatrix} \in \mathbb{R}^{m\times(n-k+1)} \tag{7.88}$$

From these two equations the MGS algorithm can be derived

ALGORITHM 7.2 (Modified Gram-Schmidt)
  Initialize $\mathbf{A}^{(1)} = \mathbf{A}$
  For $k = 1$ to n
    $(\mathbf{z} \quad \mathbf{B}) = \mathbf{A}^{(k)}$
    $r_{kk} = \|\mathbf{z}\|_2$
    $\mathbf{q}_k = \frac{\mathbf{z}}{r_{kk}}$
    $(r_{k,k+1} \quad \cdots \quad r_{k,n}) = \mathbf{q}_k^T \mathbf{B}$ // $\mathbf{q}_k^T \mathbf{q}_i = 0$ for $k \neq i$
    $\mathbf{A}^{(k+1)} = \mathbf{B} - \mathbf{q}_k (r_{k,k+1} \quad \cdots \quad r_{k,n})$
  End

The quality of the orthogonal basis computed by the MGS is [39]

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{I} + \mathbf{E}_{MGS}\,, \qquad \|\mathbf{E}_{MGS}\|_2 \approx \mu\kappa_2(\mathbf{A}) \tag{7.89}$$

where μ is the *machine unit roundoff* and $\kappa_2(\mathbf{A})$ is the 2-norm condition number of $\mathbf{A}$. Thus, in general the MGS should be used to compute orthogonal bases only when the vectors to be orthogonalized are fairly independent.

If the $n-1$ first columns of $\mathbf{A}$ are already orthogonal, then the last column of $\mathbf{Q}$ and $\mathbf{R}$ had to be found

$$\mathbf{A} = \begin{pmatrix} \mathbf{U} & \mathbf{a} \end{pmatrix} = \begin{pmatrix} \mathbf{U} & \mathbf{q} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{n-1} & \mathbf{r} \\ \mathbf{0} & \mathbf{r}_{nn} \end{pmatrix} \tag{7.90}$$

for $\mathbf{r} \in \mathbb{R}^{(n-1)\times 1}$ and scalar r_{nn}. First by multiplying both sides of (7.90) by $\mathbf{U}^T$, it is seen that

$$\mathbf{r} = \mathbf{U}^T\mathbf{a} \tag{7.91}$$

and from (7.90)

$$\mathbf{q}r_{nn} = \mathbf{a} - \mathbf{U}\mathbf{r}\,, \qquad r_{nn} = \|\mathbf{q}r_{nn}\|_2 \tag{7.92}$$

which is the desired projection $\mathbf{t} = \mathbf{q}r_{nn} = (\mathbf{I}-\mathbf{U}\mathbf{U}^T)\mathbf{a}$. For this case the algorithm only operates on the last column of $\mathbf{A}^{(k)}$, giving a simplified version of the MGS algorithm for obtaining $\mathbf{t}$

ALGORITHM 7.3 (MGS based Projection of Vector)
Initialize $\mathbf{a}^{(1)} = \mathbf{a}$ // $\mathbf{A} = (\ \mathbf{u}_1 \ \cdots \ \mathbf{u}_{n-1} \ \mathbf{a}\ )$
For $k = 1$ to $n-1$
$r_{kn} = \mathbf{u}_k^T \mathbf{a}^{(k)}$ // $r_{kk} = 1,\ (\ r_{k,k+1} \ \cdots \ r_{k,n-1}\ ) = \mathbf{0}$
$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} - \mathbf{u}_k r_{kn}$
End
$\mathbf{t} = \mathbf{a}^{(n)}$

When the last column $\mathbf{a}$ is initialized to $\mathbf{e}_1$, the algorithm can be compressed once more if the Classical Gram-Schmidt (CGS) is used, i.e.,

ALGORITHM 7.4 (CGS based Projection of $\mathbf{e}_1$)
Initialize $\mathbf{a}^{(1)} = \mathbf{e}_1$
For $k = 1$ to $n-1$
$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} - \mathbf{u}_k u_{1k}$
End
$\mathbf{t} = \mathbf{a}^{(n)}$

Thus, the expanded column $\mathbf{q}$ can be obtained by combining Algorithm 7.3 and 7.4 with Algorithm 7.1 on Page 127, where Algorithm 7.4 is used to orthogonalize $\mathbf{e}_1$ against the columns of $\mathbf{U}$, and Algorithm 7.3 is used if reorthogonalization of the obtained solution is required. Note, that only the $\mathbf{U}$ matrix is needed in the expansion.

7.4.2.2 Method of Corrected Semi-Normal Equations

The method of Corrected Semi-Normal Equations (CSNE) [6, 7, 91] obtain the row $\mathbf{u}_{X1}^T$ and the expansion element q_1 (7.70) from the matrix $\mathbf{X}$ and the ULLV decomposition elements $\mathbf{L}_X$, $\mathbf{L}$ and $\mathbf{V}$. As demonstrated by Björck et al. in [7], the explicit use of the original data $\mathbf{X}$ typically gives the same accuracy as the MGSR method based on the matrix $\mathbf{U}_X$.

Consider the QL-factorization

$$\mathbf{X}\mathbf{V} = \mathbf{U}_X \mathbf{L}_X \mathbf{L} \tag{7.93}$$

then the normal equations for solving the least-squares problem

$$\min_{\mathbf{z}} \|\mathbf{e}_1 - \mathbf{X}\mathbf{V}\mathbf{z}\| \tag{7.94}$$

can be rewritten as

$$(\mathbf{L}_X\mathbf{L})^T(\mathbf{L}_X\mathbf{L})\mathbf{z} = (\mathbf{X}\mathbf{V})^T\mathbf{e}_1 \tag{7.95}$$

which are called the *Semi-Normal Equations* (SNE). The desired quantities can now be found by two steps of triangular solves, where the first triangular system gives the first row of $\mathbf{U}_X$

$$\mathbf{u}_{X1}^T(\mathbf{L}_X\mathbf{L}) = \mathbf{e}_1^T(\mathbf{X}\mathbf{V}) = \mathbf{x}_1^T\mathbf{V} \tag{7.96}$$

and $\mathbf{z}$ is found from the second system

$$(\mathbf{L}_X\mathbf{L})\mathbf{z} = \mathbf{u}_{X1} \tag{7.97}$$

Then the residual vector in the least-squares problem is the desired projection $\mathbf{t}$

$$\begin{aligned}\mathbf{t} &= \mathbf{e}_1 - (\mathbf{XV})\mathbf{z} \\ &= \mathbf{e}_1 - \mathbf{XVL}^{-1}\mathbf{L}_X^{-1}\mathbf{u}_{X1} \\ &= \mathbf{e}_1 - \mathbf{XVL}^{-1}\mathbf{L}_X^{-1}(\mathbf{XVL}^{-1}\mathbf{L}_X^{-1})^T\mathbf{e}_1 \\ &= (\mathbf{I} - \mathbf{U}_X\mathbf{U}_X^T)\mathbf{e}_1\end{aligned} \tag{7.98}$$

which is orthogonal to the matrices $\mathbf{XV}$ and $\mathbf{U}_X$. As shown by Björck [6], the error in the SNE solution is of the same size as that for the method of normal equations.

In the CSNE method, the corrected solution is computed by performing one step of iterative refinement of the solution computed by SNE, i.e., to have a better estimate of $\mathbf{u}_{X1}$ and $\mathbf{t}$, corrections $\delta\mathbf{u}_{X1}$ and $\delta\mathbf{z}$ can be obtained by solving (7.95) with $\mathbf{e}_1 = \mathbf{t}$

$$\delta\mathbf{u}_{X1}^T(\mathbf{L}_X\mathbf{L}) = \mathbf{t}^T(\mathbf{XV}) \tag{7.99}$$

$$(\mathbf{L}_X\mathbf{L})\delta\mathbf{z} = \delta\mathbf{u}_{X1} \tag{7.100}$$

Then $\mathbf{u}_{X1}$ and $\mathbf{t}$ are corrected as follows

$$\mathbf{u}_{X1} = \mathbf{u}_{X1} + \delta\mathbf{u}_{X1} \tag{7.101}$$

$$\mathbf{t} = \mathbf{t} - \mathbf{XV}\delta\mathbf{z} \tag{7.102}$$

The CSNE method, although not backward stable, will usually give an acceptable-error stable result [6], i.e., in most applications similar to the MGS based method.

The semi-normal equations and the correction step can be used in an algorithm like Algorithm 7.1 on Page 127 to find the expanded column of $\mathbf{U}_X$, but unlike the MGSR, the correction step *must* always be performed. The case of $\mathbf{U}_N$ is treated in a similar manner.

In practice, the systems containing the product $\mathbf{L}_X\mathbf{L}$ is solved by a two step triangular process. Another issue is to handle the case with rank deficient $\mathbf{X}$ and thereby $\mathbf{L}_X$. Assume that the numerical rank of $\mathbf{X}$ is $p < n$, and that the decomposition of $\mathbf{X}$ is given by (6.53) with $\mathbf{F}_X \approx \mathbf{0}$ and $\mathbf{G}_X \approx \mathbf{0}$. Then $\langle\mathbf{V}_2\rangle \approx \text{null}(\mathbf{X})$, and the last $n-p$ elements of the right-hand side of (7.95) will be approximately zero. Thus, the solution to the first three triangular solves, i.e., with coefficient matrices $\mathbf{L}^T$, $\mathbf{L}_X^T$ and $\mathbf{L}_X$, will also have zero elements in the same places, and the systems can therefore be solved for the first p elements by using the upper-left $p \times p$ coefficient matrices. A possible tolerance for the numerical rank determination could be

$$\tau = 10n\mu \tag{7.103}$$

where μ is the machine unit roundoff.

7.4.3 Removing a Row from X

Given the ULLV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$, the starting point is the following initial downdate decomposition, where $\mathbf{U}_X$ is argumented with the column $\mathbf{e}_1$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_X & \mathbf{e}_1 \end{pmatrix} \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_X \boldsymbol{\mathcal{L}}_X \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.104}$$

$$\mathbf{N} = \mathbf{U}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_N \boldsymbol{\mathcal{L}} \mathbf{V}^T \tag{7.105}$$

The downdate process is split up into the expansion step and the downdating step consisting of three substeps. As the updating, the downdating process is illustrated by means of an example with matrix dimensions $m = 7$ and $n = 5$, which gives the initial matrices

$$\boldsymbol{\mathcal{L}}_X = \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\mathcal{L}} = \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{\mathcal{U}}_X = \begin{pmatrix} \mathbf{U}_X & \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} u & u & u & u & u & 1 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

7.4.3.1 Expansion step – DX1

Step DX1. First, use one of the expansion methods to orthogonalize $\mathbf{e}_1$ against the columns of $\mathbf{U}_X$

$$\begin{pmatrix} \mathbf{U}_X & \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{X1}^T & 1 \\ \bar{\mathbf{U}}_X & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{X1}^T & q_1 \\ \bar{\mathbf{U}}_X & \bar{\mathbf{q}} \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{u}_{X1} \\ \mathbf{0} & q_1 \end{pmatrix} = \mathbf{U'}_X \mathbf{R} \tag{7.106}$$

The orthogonal columns of $\mathbf{U}_X$ and the special form of the appended column determine the structure of $\mathbf{R}$, and substitution in (7.104) gives

$$\begin{aligned} \mathbf{U'}_X \mathbf{R} \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} &= \mathbf{U'}_X \begin{pmatrix} \mathbf{L}_X & \mathbf{u}_{X1} \\ \mathbf{0}^T & q_1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \\ &= \mathbf{U'}_X \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \end{aligned} \tag{7.107}$$

which mean that the factor $\mathbf{R}$ can be left out because the third factor has a zero row at the bottom.

7.4.3.2 Downdating step – DX2, DX3 and DX4

Step DX2. Second, eliminate all but the first and last element in the first row of $\mathbf{U'}_X$, maintaining the rank revealing structure of $\mathbf{L}_X$ as much as possible. The following three substeps describes the elimination and are done in a loop for $i = n, n-1, \ldots, 2$, where rotations are operating on neighboring columns or rows $(i-1, i)$.

Step DX2-1. Annihilate the ith element by postmultiply $\mathbf{U}'_X$ with a rotation $\mathbf{P}$ in the plane $(i-1,i)$, which introduces a nonzero $(i-1,i)$ entry z in $\mathbf{L}_X$

$$\mathbf{U}'_X\mathbf{P}\mathbf{P}^T\begin{pmatrix}\mathbf{L}_X & \mathbf{0}\\ \mathbf{0}^T & 1\end{pmatrix} = \bar{\mathbf{U}}_X\begin{pmatrix}\mathbf{L}'_X & \mathbf{0}\\ \mathbf{0}^T & 1\end{pmatrix} \tag{7.108}$$
$$= \begin{pmatrix} \bar{u}_{11} & 0 & 0 & 0 & 0 & \bar{u}_{16}\\ u & u & u & u & u & u\\ u & u & u & u & u & u\\ u & u & u & u & u & u\\ u & u & u & u & u & u\\ u & u & u & u & u & u\\ u & u & u & u & u & u \end{pmatrix}\begin{pmatrix} l & z & & & & 0\\ l & l & z & & & 0\\ l & l & l & z & & 0\\ l & l & l & g & z & 0\\ f & f & g & g & g & 0\\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Note, that postmultiplication by plane rotations cannot change the norm of a row, so the first row of $\mathbf{U}_X$ will still have norm one.

Step DX2-2. Eliminate the superdiagonal of $\mathbf{L}'_X$ by applying from the right a rotation $\mathbf{Q}$ in the plane $(i-1,i)$, which introduces a nonzero $(i-1,i)$ entry in $\mathbf{L}$. This rotation must be propagated to the left in both decompositions (7.104) and (7.105)

$$\begin{pmatrix}\mathbf{L}'_X & \mathbf{0}\\ \mathbf{0}^T & 1\end{pmatrix}\mathbf{Q}\mathbf{Q}^T\begin{pmatrix}\mathbf{L}\\ \mathbf{0}^T\end{pmatrix} = \begin{pmatrix}\bar{\mathbf{L}}_X & \mathbf{0}\\ \mathbf{0}^T & 1\end{pmatrix}\begin{pmatrix}\mathbf{L}'\\ \mathbf{0}^T\end{pmatrix} \tag{7.109}$$
$$= \begin{pmatrix} l & & & & & 0\\ l & l & & & & 0\\ l & l & l & & & 0\\ l & l & l & l & & 0\\ f & f & g & g & g & 0\\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z\\ l & l & l & l & l\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{U}_N\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\end{pmatrix}\mathbf{Q} = \mathbf{U}_N\bar{\mathbf{Q}}\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\end{pmatrix} = \bar{\mathbf{U}}_N\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\end{pmatrix} \tag{7.110}$$

where $\mathbf{Q}$ can be directly incorporated into $\mathbf{U}_N$ because only the $\mathbf{I}_n$ part of the second factor is affected.

Step DX2-3. Eliminate the superdiagonal of $\mathbf{L}'$ by applying from the right a rotation $\mathbf{J}$ in the plane $(i-1,i)$

$$\begin{pmatrix}\mathbf{L}'\\ \mathbf{0}^T\end{pmatrix}\mathbf{J}\mathbf{J}^T\mathbf{V}^T = \begin{pmatrix}\bar{\mathbf{L}}\\ \mathbf{0}^T\end{pmatrix}\bar{\mathbf{V}}^T \tag{7.111}$$
$$= \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} v & v & v & v & v\\ v & v & v & v & v\\ v & v & v & v & v\\ v & v & v & v & v\\ v & v & v & v & v \end{pmatrix}$$

After the first two substeps, the numerical rank of $\bar{\mathbf{L}}_X$ apparently *increase by one* over that of $\mathbf{L}_X$ because of the linear combination of a row (or column) with large elements and a row (or column) with small elements, but actually the rank cannot increase in the downdating step. The decomposition is now

$$\mathbf{X} = \bar{\mathbf{U}}_X \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.112}$$

$$\mathbf{N} = \bar{\mathbf{U}}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.113}$$

Step DX3 – Inverse step X3. The transformed matrices now have the form as after step X3 in the updating process. Thus, the first and last element of the first row of $\bar{\mathbf{U}}_X$ are the rotation parameters (c_3, s_3) in the update step X3

$$s_3 = \bar{u}_{11} \quad \text{and} \quad c_3 = \bar{u}_{16} \tag{7.114}$$

Then the inverse of the rotation (c_3, s_3) can be applied to the decomposition

$$\bar{\mathbf{U}}_X \mathbf{K}^T \mathbf{K} \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \hat{\mathbf{U}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}}_X & -s_3\mathbf{e}_1 \\ \eta\mathbf{e}_1^T & c_3 \end{pmatrix} \tag{7.115}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ c_3\bar{u}_{21} - s_3\bar{u}_{26} & u & u & u & u & 0 \\ c_3\bar{u}_{31} - s_3\bar{u}_{36} & u & u & u & u & 0 \\ c_3\bar{u}_{41} - s_3\bar{u}_{46} & u & u & u & u & 0 \\ c_3\bar{u}_{51} - s_3\bar{u}_{56} & u & u & u & u & 0 \\ c_3\bar{u}_{61} - s_3\bar{u}_{66} & u & u & u & u & 0 \\ c_3\bar{u}_{71} - s_3\bar{u}_{76} & u & u & u & u & 0 \end{pmatrix} \begin{pmatrix} c_3\bar{l}_{X11} & & & & & -s_3 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ l & l & l & l & & 0 \\ f & f & g & g & g & 0 \\ s_3\bar{l}_{X11} & 0 & 0 & 0 & 0 & c_3 \end{pmatrix}$$

where the last column of $\bar{\mathbf{U}}_X$, because of the orthogonality, is transformed to $\mathbf{e}_1$. Using (7.66) divide by c_3 also gives the first column of $\hat{\mathbf{U}}_X$ but this may be numerically inaccurate for c_3 close to zero.

Step DX4 – Inverse step X2. $\hat{\mathbf{L}}_X$ now has the form as after step X2 in the updating process. Thus the first element of the last row of $\hat{\mathbf{L}}_X$ is the value η in (7.45)

$$\eta = \frac{-s_2}{c_2} = s_3\bar{l}_{X11} \tag{7.116}$$

The inverse of the scaled rotation (c_2, s_2) in step X2 can then be applied to the decomposition. This single rotation had to zero the n first elements in the last column of the second factor in both decompositions. Because the third factor has a zero row at the bottom, then the last column of these two matrices can be changed to satisfy this requirement, which is the form in (7.43) and (7.44)

$$\begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{z} \\ \eta\mathbf{e}_1^T & z \end{pmatrix} \mathbf{Y}\mathbf{Y}^{-1} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{L}}_X & -\eta\hat{\mathbf{l}}_{X1} \\ \eta\mathbf{e}_1^T & 1 \end{pmatrix} \mathbf{Y}\mathbf{Y}^{-1} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \tag{7.117}$$
$$= \begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \xi\mathbf{e}_1^T \end{pmatrix}$$
$$= \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ l & l & l & l & & 0 \\ f & f & g & g & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ \frac{-s_2}{c_2}\bar{l}_{11} & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \mathbf{Y} = \begin{pmatrix} \mathbf{I}_n & -\eta\mathbf{e}_1 \end{pmatrix} \mathbf{Y} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \tag{7.118}$$

Consequently the step DX4 introduces only one new element ξ given by

$$\xi = \frac{-s_2}{c_2}\bar{l}_{11} = \eta\bar{l}_{11} \tag{7.119}$$

Hence, step DX4 can be implemented using only one multiplication, and the the new decomposition is

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \hat{\mathbf{U}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \xi\mathbf{e}_1^T \end{pmatrix} \hat{\mathbf{V}}^T \tag{7.120}$$
$$\mathbf{N} = \hat{\mathbf{U}}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \xi\mathbf{e}_1^T \end{pmatrix} \hat{\mathbf{V}}^T \tag{7.121}$$

Thus ξ is the norm of the row $\mathbf{x}^T$ being removed from $\mathbf{X}$

$$\|\mathbf{x}^T\| = \|\xi\mathbf{e}_1^T\mathbf{V}^T\| = \|\xi\mathbf{e}_1^T\| = |\xi| \tag{7.122}$$

Finally, the downdated decomposition is

$$\hat{\mathbf{X}} = \hat{\mathbf{U}}_X\hat{\mathbf{L}}_X\hat{\mathbf{L}}\hat{\mathbf{V}}^T \tag{7.123}$$
$$\mathbf{N} = \hat{\mathbf{U}}_N\hat{\mathbf{L}}\hat{\mathbf{V}}^T \tag{7.124}$$

Note, that the row dimension of $\mathbf{U}_X$ has decreased by one, and that the first row of $\mathbf{U}_X$ is needed to construct the plane rotations.

7.4.4 Removing a Row from N

Given the ULLV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$, the starting point is the following initial downdate decomposition, where $\mathbf{U}_N$ is argumented with the column $\mathbf{e}_1$

$$\mathbf{X} = \mathbf{U}_X \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \mathbf{U}_X\boldsymbol{\mathcal{L}}_X\boldsymbol{\mathcal{L}}\mathbf{V}^T \tag{7.125}$$
$$\mathbf{N} = \begin{pmatrix} \mathbf{n}^T \\ \hat{\mathbf{N}} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_N & \mathbf{e}_1 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \boldsymbol{\mathcal{U}}_N\boldsymbol{\mathcal{L}}\mathbf{V}^T \tag{7.126}$$

Again, the downdate process is split up into the expansion step and the downdating step consisting of two substeps, and the process is illustrated by means of an example with matrix dimensions $m = 7$ and $n = 5$, which gives the initial matrices

$$\boldsymbol{\mathcal{L}}_X = \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \end{pmatrix} = \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ f & f & g & g & & 0 \\ f & f & g & g & g & 0 \end{pmatrix}, \quad \boldsymbol{\mathcal{L}} = \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{\mathcal{U}}_N = \begin{pmatrix} \mathbf{U}_N & \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} u & u & u & u & u & 1 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \\ u & u & u & u & u & 0 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

7.4.4.1 Expansion step – DN1

Step DN1. First, use one of the expansion methods to orthogonalize $\mathbf{e}_1$ against the columns of $\mathbf{U}_N$

$$\begin{pmatrix} \mathbf{U}_N & \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{N1}^T & 1 \\ \bar{\mathbf{U}}_N & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{N1}^T & q_1 \\ \bar{\mathbf{U}}_N & \bar{\mathbf{q}} \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{u}_{N1} \\ \mathbf{0} & q_1 \end{pmatrix} = \mathbf{U}'_N \mathbf{R} \tag{7.127}$$

The orthogonal columns of $\mathbf{U}_N$ and the special form of the appended column determine the structure of $\mathbf{R}$, and substitution in (7.126) gives

$$\mathbf{U}'_N \mathbf{R} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} = \mathbf{U}'_N \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \tag{7.128}$$

which mean that the factor $\mathbf{R}$ can be left out because the second factor has a zero row at the bottom.

7.4.4.2 Downdating step – DN2 and DN3

Step DN2. Second, eliminate all but the first and last element in the first row of $\mathbf{U}'_N$, maintaining the rank revealing structure of $\mathbf{L}_X$ as much as possible. The following three substeps describe the elimination and are done in a loop for $i = n, n-1, \ldots, 2$, where rotations are operating on neighboring columns or rows $(i-1, i)$.

Step DN2-1. Annihilate the ith element by postmultiplying $\mathbf{U}'_N$ with a rotation $\mathbf{Q}$ in the plane $(i-1,i)$, which introduces a nonzero $(i-1,i)$ entry z in $\mathbf{L}$. This rotation must also be propagated to the left in (7.125), which introduces a nonzero $(i-1,i)$ entry z in $\mathbf{L}_X$

$$\mathbf{U}'_N \mathbf{Q}\mathbf{Q}^T \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} = \bar{\mathbf{U}}_N \begin{pmatrix} \mathbf{L}' \\ \mathbf{0}^T \end{pmatrix} \tag{7.129}$$
$$= \begin{pmatrix} \bar{u}_{11} & 0 & 0 & 0 & 0 & \bar{u}_{16} \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \end{pmatrix} \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & l & l & l \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{L}_X & \mathbf{0} \end{pmatrix} \mathbf{Q} = \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \end{pmatrix} = \begin{pmatrix} l & z & & & & 0 \\ l & l & z & & & 0 \\ l & l & l & z & & 0 \\ f & f & g & g & z & 0 \\ f & f & g & g & g & 0 \end{pmatrix} \tag{7.130}$$

Note, that postmultiplication by plane rotations cannot change the norm of a row, so the first row of $\mathbf{U}_N$ will still have norm one.

Step DN2-2. Eliminate the superdiagonal of $\mathbf{L}'_X$ by applying from the left a rotation $\mathbf{P}^T$ in the plane $(i-1,i)$

$$\mathbf{U}_X \mathbf{P}\mathbf{P}^T \begin{pmatrix} \mathbf{L}'_X & \mathbf{0} \end{pmatrix} = \bar{\mathbf{U}}_X \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \tag{7.131}$$
$$= \begin{pmatrix} u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \\ u & u & u & u & u & u \end{pmatrix} \begin{pmatrix} l & & & & & 0 \\ l & l & & & & 0 \\ l & l & l & & & 0 \\ l & l & l & l & & 0 \\ f & f & g & g & g & 0 \end{pmatrix}$$

Step DN2-3. Eliminate the superdiagonal of $\mathbf{L}'$ by applying from the right a rotation $\mathbf{J}$ in the plane $(i-1,i)$

$$\begin{pmatrix} \mathbf{L}' \\ \mathbf{0}^T \end{pmatrix} \mathbf{J}\mathbf{J}^T \mathbf{V}^T = \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.132}$$
$$= \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix}$$

After the first two substeps, the numerical rank of $\bar{\mathbf{L}}_X$ apparently *increase by one* over that of $\mathbf{L}_X$ because of the linear combination of a row (or column) with large elements and a row (or column) with small elements, but actually the rank cannot increase in the downdating step. The decomposition is now

$$\mathbf{X} = \bar{\mathbf{U}}_X \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.133}$$

$$\mathbf{N} = \bar{\mathbf{U}}_N \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \bar{\mathbf{V}}^T \tag{7.134}$$

Step DN3 – Inverse step N2. The transformed matrices now have the form as after step N2 in the updating process. Thus, the first and last element of the first row of $\bar{\mathbf{U}}_N$ are the rotation parameters (c_2, s_2) in the update step N2

$$s_2 = \bar{u}_{11} \quad \text{and} \quad c_2 = \bar{u}_{16} \tag{7.135}$$

Then the inverse of the rotation (c_2, s_2) can be applied to the decomposition. This single rotation had to transform the matrices to the form as before step N2 in the updating process. Because the factor to the right has a zero row at the bottom, then the last column of $\begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix}$ can be changed to satisfy this requirement, which is the form in (7.59)

$$\begin{aligned} \begin{pmatrix} \bar{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \mathbf{K}^T\mathbf{K} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} &= \begin{pmatrix} \bar{\mathbf{L}}_X & -s_2\hat{\mathbf{l}}_{X1} \end{pmatrix} \mathbf{K}^T\mathbf{K} \begin{pmatrix} \bar{\mathbf{L}} \\ \mathbf{0}^T \end{pmatrix} \\ &= \begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \\ &= \begin{pmatrix} \frac{\bar{l}_{X11}}{c_2} & & & & 0 \\ \frac{\bar{l}_{X21}}{c_2} & l & & & 0 \\ \frac{\bar{l}_{X31}}{c_2} & l & l & & 0 \\ \frac{\bar{l}_{X41}}{c_2} & l & l & l & 0 \\ \frac{\bar{f}_{X51}}{c_2} & f & g & g & g & 0 \end{pmatrix} \begin{pmatrix} c_2\bar{l}_{11} & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \\ s_2\bar{l}_{11} & 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{7.136}$$

and

$$\bar{\mathbf{U}}_N\mathbf{K}^T = \begin{pmatrix} \mathbf{0}^T & 1 \\ \hat{\mathbf{U}}_N & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ c_2\bar{u}_{21} - s_2\bar{u}_{26} & u & u & u & u & 0 \\ c_2\bar{u}_{31} - s_2\bar{u}_{36} & u & u & u & u & 0 \\ c_2\bar{u}_{41} - s_2\bar{u}_{46} & u & u & u & u & 0 \\ c_2\bar{u}_{51} - s_2\bar{u}_{56} & u & u & u & u & 0 \\ c_2\bar{u}_{61} - s_2\bar{u}_{66} & u & u & u & u & 0 \\ c_2\bar{u}_{71} - s_2\bar{u}_{76} & u & u & u & u & 0 \end{pmatrix} \tag{7.137}$$

where the last column of $\bar{\mathbf{U}}_N$, because of the orthogonality, is transformed to $\mathbf{e}_1$. Using (7.69) divide by c_2 also gives the first column of $\hat{\mathbf{U}}_N$ but this may be numerically inaccurate for c_2 close to zero. Unfortunately, the only way to find $\hat{\mathbf{L}}_X$ is to divide the first column by c_2, because the column $-s_2\hat{\mathbf{l}}_{X1}$ is unknown. The new decomposition is

$$\mathbf{X} = \hat{\mathbf{U}}_X \begin{pmatrix} \hat{\mathbf{L}}_X & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \hat{\mathbf{V}}^T \tag{7.138}$$

$$\mathbf{N} = \begin{pmatrix} \mathbf{n}^T \\ \hat{\mathbf{N}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \hat{\mathbf{U}}_N & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{L}} \\ \xi \mathbf{e}_1^T \end{pmatrix} \hat{\mathbf{V}}^T \tag{7.139}$$

Thus, the element ξ given by

$$\xi = s_2 \bar{l}_{11} \tag{7.140}$$

is the norm of the row $\mathbf{n}^T$ being removed from $\mathbf{N}$

$$\|\mathbf{n}^T\| = \|\xi \mathbf{e}_1^T \mathbf{V}^T\| = \|\xi \mathbf{e}_1^T\| = |\xi| \tag{7.141}$$

Finally, the downdated decomposition is

$$\mathbf{X} = \hat{\mathbf{U}}_X \hat{\mathbf{L}}_X \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.142}$$

$$\hat{\mathbf{N}} = \hat{\mathbf{U}}_N \hat{\mathbf{L}} \hat{\mathbf{V}}^T \tag{7.143}$$

Note again, that the row dimension of $\mathbf{U}_N$ has decreased by one, and that we need the first row of $\mathbf{U}_N$ to construct the plane rotations.

7.5 Making the ULLV Decomposition Rank-Revealing

After an update in the exponential window method, the numerical rank p of $\mathbf{XN}^+$ as defined in (6.52) can increase by one, decrease, or stay the same. In particular, since the effect of old data is only gradually phased out, it is likely that situations occur, where the numerical rank is not well determined, and the rank decisions are difficult.

An advantage of the sliding window method is that after adding a row, the rank can only stay the same or increase by one. Likewise after removing a row, the rank can only stay the same or decrease by one. Therefore, some a priori information on the numerical rank after the modification is always available.

In the ULLV decomposition an apparant increase in rank in the updating/downdating procedure can turn out to be spurious. We therefore need a means of detecting the numerical rank p of $\mathbf{XN}^+$ and computing a corresponding rank-reavealing ULLV decomposition for it. This is done in a deflation step [107, 110] to separate $\mathbf{L}_X$ in the matrices $\mathbf{L}_{X1}$, $\mathbf{F}_X$ and $\mathbf{G}_X$ (6.53), and possibly followed by a refinement step [75, 107] to bring $\mathbf{L}_X$ nearer to block diagonality.

7.5.1 Deflation

Step D1. The first step is to determine if $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ is defective in rank, that is, if the smallest singular value σ_n of $\mathbf{L}_X$ is less than a prescribed tolerance τ. This problem is solved by using a reliable condition estimator to find a vector $\hat{\mathbf{u}}_n$ of norm one, which estimate the left singular vector $\mathbf{u}_n$ associated with $\sigma_n(\mathbf{L}_X)$, i.e.,

$$\|\mathbf{L}_X^T \hat{\mathbf{u}}_n\|_2 \approx \sigma_n(\mathbf{L}_X) \tag{7.144}$$

If $\|\mathbf{L}_X^T \hat{\mathbf{u}}_n\|_2$ is greater than the tolerance τ, then there is nothing to be done. Otherwise, $\mathbf{L}_X$ must be modified by plane rotations so that the elements in its last row become small. This is described in step D2 to D5, which are done in a loop for $i = 2, 3, \ldots, n$, where rotations are operating on neighboring columns or rows $(i-1, i)$.

Step D2. Begin by apply a sequence of left rotations $\mathbf{P}^T$ in plane $(i-1, i)$ to transform $\hat{\mathbf{u}}_n$ into the nth unit vector $\pm\mathbf{e}_n$

$$\begin{array}{ccccccccc}
\to \bar{u} & & 0 & & 0 & & 0 & & 0 \\
\to u & & \to \bar{u} & & 0 & & 0 & & 0 \\
u & \Rightarrow & \to u & \Rightarrow & \to \bar{u} & \Rightarrow & 0 & \Rightarrow & 0 \\
u & & u & & \to u & & \to \bar{u} & & 0 \\
u & & u & & u & & \to u & & \pm 1
\end{array} \tag{7.145}$$

$$\mathbf{P}^T \hat{\mathbf{u}}_n = \pm \mathbf{e}_n \tag{7.146}$$

Step D3. Next apply the sequence of left rotations $\mathbf{P}^T$ to $\mathbf{L}_X$. Then the 2-norm of the last row of $\bar{\mathbf{L}}_X = \mathbf{P}^T \mathbf{L}_X$ approximates the smallest singular value of $\mathbf{L}_X$, since

$$\|\mathbf{e}_n^T \bar{\mathbf{L}}_X\|_2 = \|\hat{\mathbf{u}}_n^T \mathbf{P}\mathbf{P}^T \mathbf{L}_X\|_2 = \|\hat{\mathbf{u}}_n^T \mathbf{L}_X\|_2 \approx \sigma_n(\mathbf{L}_X) \tag{7.147}$$

The rotations $\mathbf{P}$ must also be propagated to the left in the ULLV decomposition

$$\mathbf{U}_X \mathbf{P}\mathbf{P}^T \mathbf{L}_X = \bar{\mathbf{U}}_X \bar{\mathbf{L}}_X = \begin{pmatrix} u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \end{pmatrix} \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ g & g & g & g & g \end{pmatrix} \tag{7.148}$$

where the last row of $\bar{\mathbf{L}}_X$ has been filled with g's to indicate that the elements are small.

Step D4. Eliminate the superdiagonal of $\bar{\mathbf{L}}_X$ by postmultiplying a plane rotation $\mathbf{Q}$ in the plane $(i-1, i)$. This rotation must be propagated to the left in both decompositions (6.53) and (6.54)

$$\bar{\mathbf{L}}_X \mathbf{Q}\mathbf{Q}^T \mathbf{L} = \hat{\mathbf{L}}_X \bar{\mathbf{L}} = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ g & g & g & g & g \end{pmatrix} \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & l & l & l \end{pmatrix} \tag{7.149}$$

and

$$\mathbf{U}_N \mathbf{Q} = \hat{\mathbf{U}}_N \tag{7.150}$$

The 2-norm of the last row of $\hat{\mathbf{L}}_X$ still approximates the smallest singular value of $\mathbf{L}_X$

$$\|\mathbf{e}_n^T \hat{\mathbf{L}}_X\|_2 = \|\hat{\mathbf{u}}_n^T \mathbf{L}_X \mathbf{Q}\|_2 = \|\mathbf{L}_X^T \hat{\mathbf{u}}_n\|_2 \approx \sigma_n(\mathbf{L}_X) \tag{7.151}$$

If $\hat{\mathbf{u}}_n$ is exactly the left singular vector associated with the smallest singular value of $\mathbf{L}_X$, then the last diagonal element of $\hat{\mathbf{L}}_X$ is $\sigma_n(\mathbf{L}_X)$ and the off diagonal elements in the last row of $\hat{\mathbf{L}}_X$ are zero [93]. Thus, an accurate condition estimator is important in the deflation step.

Step D5. Finally, triangularize $\bar{\mathbf{L}}$ by applying from the right a rotation $\mathbf{\Pi}$ in the plane $(i-1, i)$

$$\bar{\mathbf{L}} \mathbf{\Pi}\mathbf{\Pi}^T \mathbf{V}^T = \hat{\mathbf{L}} \hat{\mathbf{V}}^T = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix} \tag{7.152}$$

If the $(n-1) \times (n-1)$ leading principle submatrix of $\mathbf{L}_X$ is sufficiently well conditioned, then the rank-revealing ULLV decomposition is obtained. If not, the deflation steps can be repeated on the submatrix.

7.5.2 Condition Number Estimation

The deflation step was based on a *reliable* condition estimator, which was used to find an estimate of the left singular vector $\mathbf{u}_n$ associated with the smallest singular value of $\mathbf{L}_X \in \mathbb{R}^{n\times n}$.

The chosen condition estimator derived by Hager [47] use a convex optimisation approch to find the 1-norm of $\mathbf{L}_X^{-1}$. The 1-norm of $\mathbf{L}_X^{-1}$ is the maximal column sum and can be found as the maximal value of the convex function

$$f(\mathbf{d}) = \|\mathbf{L}_X^{-1}\mathbf{d}\|_1 \tag{7.153}$$

over the convex set $\mathcal{S} = \{\mathbf{d} : \|\mathbf{d}\|_1 \leq 1\}$. Obviously, the maximum is attained at one of the identity vectors $\mathbf{e}_j$. Based on these observations, the following gradient algorithm computes a lower bound $\gamma \leq \|\mathbf{L}_X^{-1}\|_1$ (see [47] or [59, page 291] for further details)

ALGORITHM 7.5 (Hager 1-norm Condition Estimator)
- Initialize $\mathbf{d} = \frac{1}{n}(1, 1, \cdots, 1)^T$
- Repeat
  - Solve $\mathbf{L}_X\mathbf{y} = \mathbf{d}$
  - Form $\boldsymbol{\xi}$ where $\xi_i = \begin{cases} 1, & y_i \geq 0 \\ -1, & y_i < 0 \end{cases}$
  - Solve $\mathbf{L}_X^T\mathbf{z} = \boldsymbol{\xi}$
  - If $\|\mathbf{z}\|_\infty \leq \mathbf{z}^T\mathbf{d}$ then
    - $\gamma = \|\mathbf{y}\|_1 = \|\mathbf{L}_X^{-1}\mathbf{d}\|_1 = f(\mathbf{d})$
    - Quit
  - Else $\mathbf{d} = \mathbf{e}_j$ where $|z_j| = \|\mathbf{z}\|_\infty$

The triangular systems are solved with forward/back-substitution and requires $2sn^2$ flops, where s is the number of iterations of the main loop for convergence. In the numerical experiments reported in [47] and [57], the algorithm almost always terminates on the second execution of the main loop, and it frequently yields $\|\mathbf{L}_X^{-1}\|_1$ exactly.

Finally, solve the triangular system

$$\mathbf{L}_X^T\mathbf{q} = \mathbf{y} \tag{7.154}$$

Then, the estimate of the left singular vector $\hat{\mathbf{u}}_n$ associated with the smallest singular value $\hat{\sigma}_n$ is

$$\hat{\mathbf{u}}_n = \frac{\mathbf{q}}{\|\mathbf{q}\|_2} \tag{7.155}$$

$$\hat{\sigma}_n = \|\mathbf{L}_X^T\hat{\mathbf{u}}_n\|_2 = \left\|\frac{\mathbf{L}_X^T\mathbf{q}}{\|\mathbf{q}\|_2}\right\|_2 = \frac{\|\mathbf{y}\|_2}{\|\mathbf{q}\|_2} \tag{7.156}$$

The Hager condition estimator is chosen due to the low computational cost and because Higham reports in [57] that this estimator provides accurate results.

Two other more reliable and more costly condition estimators are the 2-Norm Condition Estimator by Cline, Conn and Van Loan [19], which incorporates a look-behind technique in the ordinary LINPACK condition estimator, and the LAPACK 1-Norm Estimator by Higham [59, page 295], which is a modification of Hagers algorithm.

Numerical experiments with the singular value and singular vector estimators have been performed in MATLAB using three sets of random matrices (also used for test purposes in [57], see, e.g., The Test Matrix Toolbox for Matlab [58]):

Set 1 The elements of $\mathbf{X} \in \mathbb{R}^{n\times n}$ are chosen as random numbers from the uniform distribution on [0;1].

Set 2 Random matrices $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \in \mathbb{R}^{n\times n}$, where the random orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ are generated following Stewart [106] and the singular values have a preassigned exponential distribution $\sigma_i = \alpha_i^{i-1}$. The parameter α is determined by a chosen condition number $\kappa_2(\mathbf{X})$.

Set 3 Similar to the second set, but with the singular values having sharp-break distribution $1 = \sigma_i = \cdots = \sigma_{n-1} > \sigma_n = \frac{1}{\kappa_2(\mathbf{X})}$.

In each set, 1000 random matrices are formed for each of the sizes $n = 10$ and 20, and upper triangular matrices are generated by computing the QR decomposition of $\mathbf{X}$.

The estimated values $\hat{\sigma}_n$ and $\hat{\mathbf{u}}_n$ are compared with the one computed by the the MATLAB function SVD as $|\sigma_n - \hat{\sigma}_n|/\sigma_n$ and $\mathrm{dist}(\mathbf{u}_n, \hat{\mathbf{u}}_n)$. Averages over the 1000 matrices are given in Table 7.1 – 7.3 for the different cases, and selected plots showing the variation among 100 matrices are given in Figure 7.2.

In these tests, the quality of the estimates are quite sensitive to the gap σ_{n-1}/σ_n between the two smallest singular values. When the gap is small as in the exponential distribution (set 2), the error is large (see Table 7.2), and when the gap is large as in the sharp-break distribution (set 3), the error is small (see Table 7.3). Another observation is that the average estimation error is dominated by few poor estimates as shown by the example plot in Figure 7.2 and the bar graphs in Figure 7.3 – 7.4.

In the speech enhancement case, the singular spectrum is similar to one given by test set 1 and 2, i.e., from Table 7.1 – 7.2 and Figure 7.2 – 7.4 it can be concluded that *the Hager condition estimator is a good choice.*

Estimator	Size n	Average $\lvert\sigma_n - \hat{\sigma}_n\rvert/\sigma_n$	Average $\mathrm{dist}(\mathbf{u}_n, \hat{\mathbf{u}}_n)$
Cline et al.	10	0.198	0.200
	20	0.344	0.229
Hager	10	0.057	0.117
	20	0.068	0.127

Table 7.1 Average errors in set 1.

Estimator	κ_2	Size n	Average $\lvert\sigma_n - \hat{\sigma}_n\rvert/\sigma_n$	Average $\mathrm{dist}(\mathbf{u}_n, \hat{\mathbf{u}}_n)$
Cline et al.	10	10	0.290	0.620
		20	0.359	0.804
	100	10	0.321	0.436
		20	0.421	0.682
Hager	10	10	0.313	0.606
		20	0.402	0.799
	100	10	0.198	0.343
		20	0.297	0.611

Table 7.2 Average errors in set 2.

Estimator	κ_2	Size n	Average $\|\sigma_n - \hat{\sigma}_n\|/\sigma_n$	Average dist$(\mathbf{u}_n, \hat{\mathbf{u}}_n)$
Cline et al.	10	10	0.014	0.015
		20	0.090	0.035
	100	10	0.13e-03	0.14e-03
		20	1.46e-03	0.37e-03
Hager	10	10	0.103	0.025
		20	0.438	0.070
	100	10	0.09e-03	0.13e-03
		20	0.17e-03	0.18e-03

Table 7.3 Average errors in set 3.

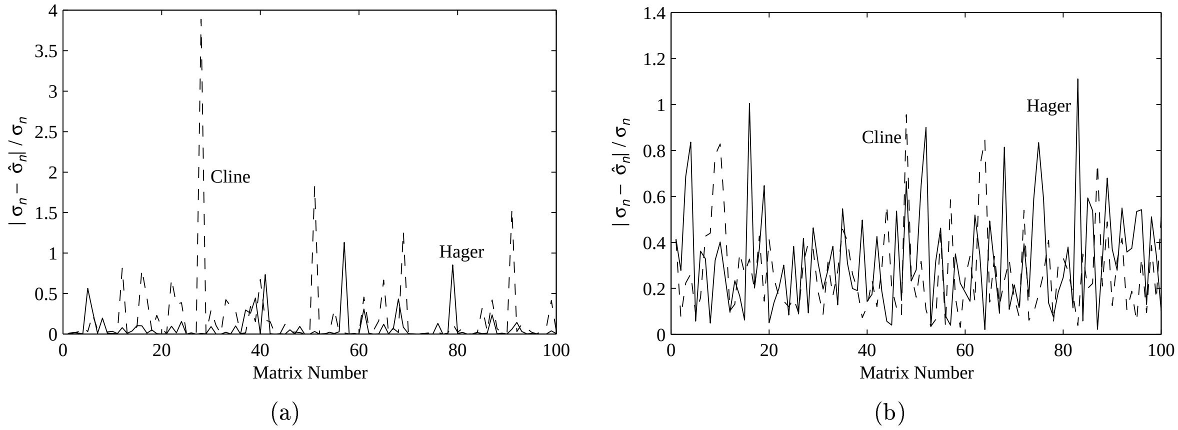


Figure 7.2 Relative error of the estimated smallest singular value of 100 triangular random matrices of dimension 10 using the Cline and Hager method. (a) Test set 1. (b) Test set 2 with condition number $\kappa_2 = 10$.

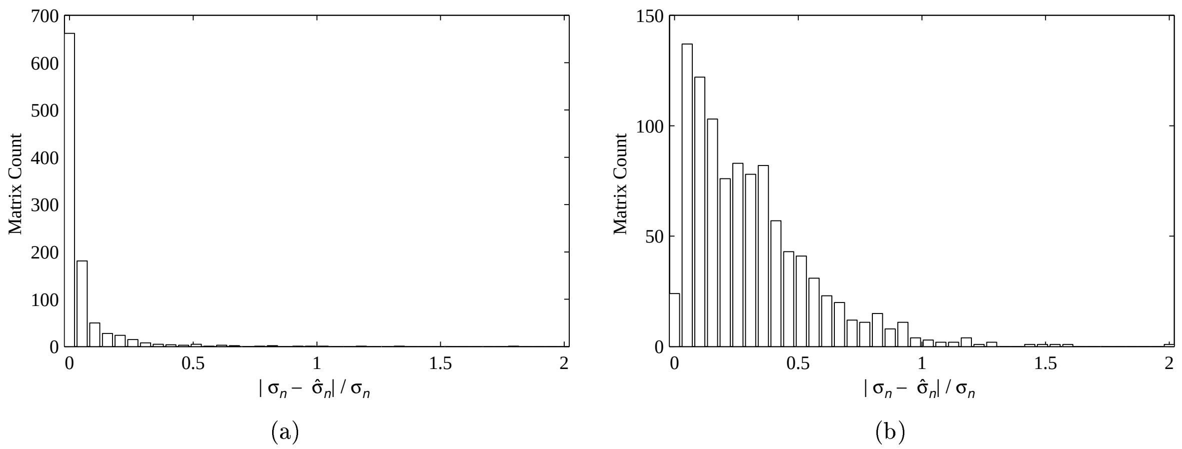
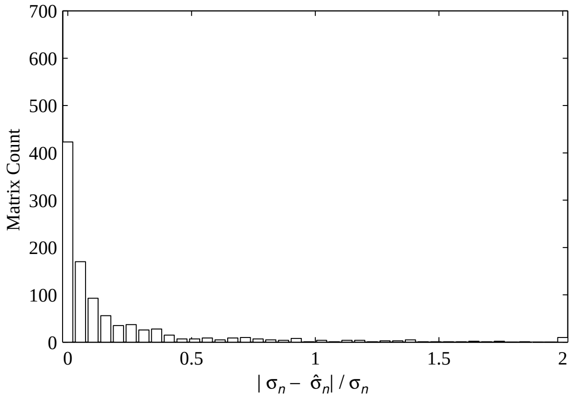
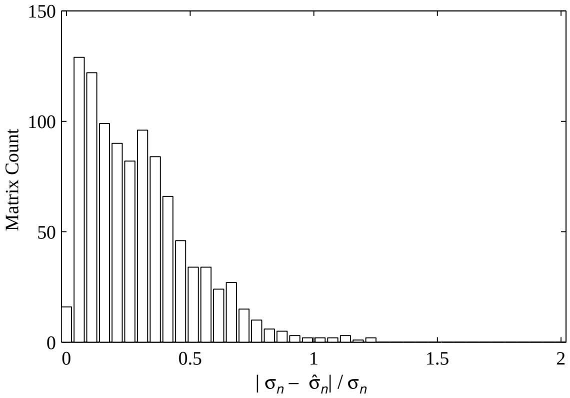


Figure 7.3 Histogram of relative error of the estimated smallest singular value of 1000 triangular random matrices of dimension 10 using Hager method. (a) Test set 1. (b) Test set 2 with condition number $\kappa_2 = 10$.



(a)



(b)

Figure 7.4 Histogram of relative error of the estimated smallest singular value of 1000 triangular random matrices of dimension 10 using Cline method. (a) Test set 1. (b) Test set 2 with condition number $\kappa_2 = 10$.

7.5.3 Refinement

Refinement is an iterative algorithm, which move a triangular matrix $\mathbf{L}$ toward diagonality. This improves the quality of the subspaces provided by the ULLV decomposition without changing the orthogonal or triangular properties of the factors.

Consider the block lower triangular matrix $\mathbf{L}_0 \in \mathbb{R}^{n \times n}$

$$\mathbf{L}_0 = \begin{pmatrix} \check{\mathbf{L}}_0 & \mathbf{0} \\ \mathbf{F}_0 & \mathbf{G}_0 \end{pmatrix} \tag{7.157}$$

where $\check{\mathbf{L}}_0 \in \mathbb{R}^{p \times p}$, $\mathbf{G}_0 \in \mathbb{R}^{n-p \times n-p}$ and $\mathbf{F}_0 \in \mathbb{R}^{n-p \times p}$. Flipping the matrix $\mathbf{L}_0$ to block upper triangular form decreases the norm of the off-diagonal block $\mathbf{F}$, and the same is valid when flipping down again

$$\mathbf{L}_1 = \mathbf{Q}_0^T \mathbf{L}_0 = \begin{pmatrix} \check{\mathbf{L}}_1 & \mathbf{F}_1 \\ \mathbf{0} & \mathbf{G}_1 \end{pmatrix} \tag{7.158}$$

$$\mathbf{L}_2 = \mathbf{L}_1 \mathbf{Q}_1 = \begin{pmatrix} \check{\mathbf{L}}_2 & \mathbf{0} \\ \mathbf{F}_2 & \mathbf{G}_2 \end{pmatrix} \tag{7.159}$$

If there is a sufficient gab in the singular values of the matrix

$$\rho = \frac{\sigma_{max}(\mathbf{G})}{\sigma_{min}(\check{\mathbf{L}})} < 1 \tag{7.160}$$

then the iteration will converge [75] in a rate

$$\|\mathbf{F}_i\| \leq \rho^i \|\mathbf{F}\| \tag{7.161}$$

$$\sigma_j(\check{\mathbf{L}}_{i+1}) \geq \sigma_j(\check{\mathbf{L}}_i)\,, \qquad j = 1, \ldots, p \tag{7.162}$$

$$\sigma_j(\mathbf{G}_{i+1}) \leq \sigma_j(\mathbf{G}_i)\,, \qquad j = 1, \ldots, n-p \tag{7.163}$$

which in the limits give

$$\lim_{i\to\infty} \|\mathbf{F}_i\| = 0 \tag{7.164}$$

$$\lim_{i\to\infty} \sigma_j(\check{\mathbf{L}}_i) = \sigma_j(\mathbf{L}) \,, \qquad j = 1, \ldots, p \tag{7.165}$$

$$\lim_{i\to\infty} \sigma_j(\mathbf{G}_i) = \sigma_{p+j}(\mathbf{L}) \,, \qquad j = 1, \ldots, n-p \tag{7.166}$$

However, the matrix $\mathbf{L}_i$ need not converge [75].

In the ULLV decomposition, refinement is used on the triangular matrix $\mathbf{L}_X$, where $\mathbf{F}_0$ and $\mathbf{G}_0$ is the last row. Thus, $\mathbf{F}_0$ is a vector and $\mathbf{G}_0$ is a scalar. The refinement is done in the following four steps.

Step R1. First flip the last row of $\mathbf{L}_X$ up by left rotations $\mathbf{P}^T$ in plane $(i-1, n)$ for $i = n, n-1, \ldots, 2$

$$\mathbf{U}_X \mathbf{P}\mathbf{P}^T \mathbf{L}_X = \hat{\mathbf{U}}_X \bar{\mathbf{L}}_X = \begin{pmatrix} u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \\ u & u & u & u & u \end{pmatrix} \begin{pmatrix} l & & & & \bar{g} \\ l & l & & & \bar{g} \\ f & f & g & & \bar{g} \\ f & f & g & g & \bar{g} \\ 0 & 0 & 0 & 0 & \bar{g} \end{pmatrix} \tag{7.167}$$

Step R2. If $\bar{\mathbf{L}}_X$ is triangularized using column rotations, then $\mathbf{L}$ will be filled up when the rotations are propagated to $\mathbf{L}$. To solve this problem, Qiao [93] has proposed to zero all but the first two elements of the last row of $\mathbf{L}$ before triangularization of $\bar{\mathbf{L}}_X$. This is done by right rotations $\mathbf{\Pi}$ in plane $(i-1, i)$ for $i = n, n-1, \ldots, 3$

$$\mathbf{L}\mathbf{\Pi}\mathbf{\Pi}^T \mathbf{V}^T = \mathbf{L}' \bar{\mathbf{V}}^T = \begin{pmatrix} l & & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix} \tag{7.168}$$

Step R3. Then flip the last column of $\bar{\mathbf{L}}_X$ down by right rotations $\mathbf{Q}$ in plane $(i-1, n)$ for $i = 2, 3, \ldots, n$. This rotation must be propagated to the left in both decompositions (6.53) and (6.54)

$$\bar{\mathbf{L}}_X \mathbf{Q}\mathbf{Q}^T \mathbf{L}' = \hat{\mathbf{L}}_X \bar{\mathbf{L}} = \begin{pmatrix} l & & & & \\ l & l & & & \\ f & f & g & & \\ f & f & g & g & \\ \hat{g} & \hat{g} & \hat{g} & \hat{g} & \hat{g} \end{pmatrix} \begin{pmatrix} l & z & & & \\ l & l & z & & \\ l & l & l & z & \\ l & l & l & l & z \\ l & l & l & l & l \end{pmatrix} \tag{7.169}$$

and

$$\mathbf{U}_N \mathbf{Q} = \hat{\mathbf{U}}_N \tag{7.170}$$

Step R4. Finally, for $i = 2, 3, \ldots, n$ apply right rotations $\mathbf{\Pi}$ in plane $(i-1, i)$ to triangularize $\bar{\mathbf{L}}$

$$\bar{\mathbf{L}}\mathbf{\Pi}\mathbf{\Pi}^T\bar{\mathbf{V}}^T = \hat{\mathbf{L}}\hat{\mathbf{V}}^T = \begin{pmatrix} l & & & & \\ l & l & & & \\ l & l & l & & \\ l & l & l & l & \\ l & l & l & l & l \end{pmatrix} \begin{pmatrix} v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \\ v & v & v & v & v \end{pmatrix} \tag{7.171}$$

The refinement step can be repeated to the size of the off-diagonal elements in the last row of $\mathbf{L}_X$ are insignificant compared to the diagonal element. As with the deflation step, the procedure can now be used on the $(n-1) \times (n-1)$ leading principle submatrix of $\mathbf{L}_X$.

7.5.4 The Complete Rank-Revealing Algorithm

Assume that $\mathbf{XN}^+$ has numerical rank p corresponding to a given tolerance τ, then a straightforward rank-revealing procedure requires $n-p$ deflation steps performed on the matrix $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ to the $(p+1) \times (p+1)$ leading principle submatrix and the same number of refinement steps, if the repetition rate is one.

However, when $\mathbf{L}_X$ is already separated in the matrices $\mathbf{L}_{X1}$, $\mathbf{F}_X$ and $\mathbf{G}_X$ before the up- and downdating, then the rank-revealing procedure can be simplified, because the dimension of $\mathbf{L}_{X1}$ only increase by one after the updating (step X1, N1) and also by one after the downdating (step DX2, DN2). Thus, in both cases the matrix $\mathbf{L}_{X1}$ is now defined as $\mathbf{L}_X(1\!:\!p+1, 1\!:\!p+1)$, which must be deflated to reflect the new rank of $\mathbf{XN}^+$. The required number of deflation steps is shown in Table 7.4 for the different possibilities of rank changes using either exponential or sliding window.

Rank-revealing after	Window	New rank of $\mathbf{XN}^+$	Required deflation steps	Required refinement steps
Updating	Both sliding and exponential	$p+1$	0	1
		p	1	1
	Exponential only	$p-1$	2	2
		$\vdots$	$\vdots$	$\vdots$
Downdating	Sliding only	p	1	1
		$p-1$	2	2

Table 7.4 Tolerance based rank-revealing.

The refinement step could still be applied to $\mathbf{L}_X$, corresponding to all the rows of the off-diagonal block $\mathbf{F}_X$, but fortunately the first row of $\mathbf{F}_X$ is the most important one to refine [67]. This results in one refinement step on the new row added to the top of $\mathbf{F}_X$ after each deflation step (see Table 7.4). For the case where the rank of $\mathbf{XN}^+$ increases, one refinement step will still be required.

Table 7.4 suggests the following general rank-revealing algorithm, where the tolerance τ and the rank p from the previous iteration is input and the new estimated rank is returned. The refinement can be repeated f times for each submatrix.

ALGORITHM 7.6 (Tolerance based Rank-Revealing)

```
If p < n                                         // Apparant increase in rank observed
  p = p + 1
End
Estimate σ_min(L_X(1 : p, 1 : p))
If σ_min < τ                                     // The rank stays the same or decrease
  While σ_min < τ and p > 1
    Deflate L_X(1 : p, 1 : p)
    For i = 1 to f
      Refine L_X(1 : p, 1 : p)
    End
    p = p − 1
    Estimate σ_min(L_X(1 : p, 1 : p))
  End
Else if p < n                                    // The rank increase by one
  For i = 1 to f
    Refine L_X(1 : p + 1, 1 : p + 1)
  End
End
```

Using the rank-revealing ULLV decomposition on speech signals, a fixed value of the numerical rank p can be used (see Section 4.5). For this situation, one deflation step and one refinement step are always required, so Algorithm 7.6 can be replaced with

ALGORITHM 7.7 (Rank based Rank-Revealing)

```
If p < n
  Estimate σ_min(L_X(1 : p + 1, 1 : p + 1))
  Deflate L_X(1 : p + 1, 1 : p + 1)
  For i = 1 to f
    Refine L_X(1 : p + 1, 1 : p + 1)
  End
  τ = σ_min
End
```

where the fixed rank p now is known and the tolerance level τ is returned. In many applications, e.g., speech enhancement, only a small (or no) improvement in performance is obtained by applying the refinement step, so typically it is left out.

7.6 Initialization of the ULLV Decomposition

To initialize the recursive ULLV algorithm, a method for computing the rank-revealing ULLV decomposition of two matrices $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$ is needed.

First compute the QL factorization of the matrix $\mathbf{N}$

$$\mathbf{N} = \mathbf{U}_N\mathbf{L} = \mathbf{U}_N\mathbf{L}\mathbf{V}^T\ , \qquad \mathbf{V} = \mathbf{I}_n \tag{7.172}$$

where $\mathbf{U}_N \in \mathbb{R}^{m\times n}$ has orthogonal columns and $\mathbf{L} \in \mathbb{R}^{n\times n}$ is lower triangular. Next, find the product $\mathbf{U}_X\mathbf{L}_X = \mathbf{Z}$ by solving a triangular system with multiple right-hand sides

$$\mathbf{X} = \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{V}^T = \mathbf{U}_X\mathbf{L}_X\mathbf{L} = \mathbf{Z}\mathbf{L}\,, \qquad \mathbf{V} = \mathbf{I}_n \tag{7.173}$$

The matrices $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ and $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ can now be found by another QL factorization of the solution matrix $\mathbf{Z}$

$$\mathbf{Z} = \mathbf{U}_X\mathbf{L}_X \tag{7.174}$$

At this point, the following decomposition is obtained

$$\mathbf{X} = \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{I}^T \tag{7.175}$$
$$\mathbf{N} = \mathbf{U}_N\mathbf{L}\mathbf{I}^T \tag{7.176}$$

Finally, find a rank-revealing ULLV decomposition by applying deflation and refinement steps on the matrix $\mathbf{L}_X$ to the $(p+1)\times(p+1)$ leading principle submatrix, where p is the numerical rank of $\mathbf{X}\mathbf{N}^+$.

7.7 The ULLV Algorithm Structure

Combining initialization, updating, downdating and rank-revealing the entirely ULLV algorithm for the case of matrix $\mathbf{X}$ becomes

Algorithm 7.8 (Rank-Revealing ULLV Algorithm)
  Initialize $\mathbf{U}_N, \mathbf{U}_X, \mathbf{L}_X, \mathbf{L}, \mathbf{V}$
  Repeat
    Update $\mathbf{X}$
    If (known tolerance τ)
      Rank-revealing Algorithm 7.6
    Else if (known rank p)
      Rank-revealing Algorithm 7.7
    End
    If (exponential window)
      Remove oldest row from $\mathbf{U}_X$
    Else if (sliding window)
      Downdate $\mathbf{X}$
      If (known tolerance τ)
        Rank-revealing Algorithm 7.6
      Else if (known rank p)
        Rank-revealing Algorithm 7.7
      End
    End

where p is the numerical rank of $\mathbf{X}\mathbf{N}^+$ corresponding to a given tolerance τ as defined in (6.52). Note, that in the sliding window case, the rank-revealing algorithm is applied twice, i.e., both after the updating and downdating step.

7.8 Computational Count

The arithmetic complexity of the rank-revealing ULLV algorithm of two matrices $\mathbf{X} \in \mathbb{R}^{m_X \times n}$ and $\mathbf{N} \in \mathbb{R}^{m_N \times n}$ is discussed in this section. One way to quantify the operation counts, is with the notion of a *flop*, which is a floating point operation, i.e., one multiplication, division, addition, subtraction or square root. Although flop counting is a crude approach to the measuring of algorithm efficiency [39], it gives us nevertheless a good idea of the possible gain in computational efficiency.

The amount of work involved in the initialization, updating, downdating, deflation and refinement steps are presented in Appendix A, where the parameters in the flop counts are given by

- m_X = row dimension of $\mathbf{U}_X$.
- m_N = row dimension of $\mathbf{U}_N$.
- n = column dimension.
- p = numerical rank of $\mathbf{X}\mathbf{N}^+$.
- r = dimension of the leading principle submatrix of $\mathbf{L}_X$ to be deflated and refined.

For large matrix dimensions (m, n), the work can be approximated to include only second-order terms as listed in Table 7.5. Thus, the recursive algorithm requires $\mathcal{O}(n^2)$ flops.

Algorithm	Flop count		
Updating $\mathbf{X}$			$12mn + 20.5n^2$
Updating $\mathbf{N}$			$12mn + 21n^2$
MGS or MGSR expansion	$2mn$	or	$6mn$
SNE or CSNE expansion	$2mn + 8n^2$	or	$6mn + 16n^2$
Downdating $\mathbf{X}$			$12mn + 18n^2$
Downdating $\mathbf{N}$			$12mn + 18n^2$
Deflation (1 step)			$12mp + 18np + 6p^2$
Refinement (1 step)			$12mp + 30np - 3p^2$

Table 7.5 Approximate flop count for substeps of the rank-revealing ULLV algorithm.

The operation counts for all substeps other than the CSNE-based expansion assume that the left-side orthogonal matrices $\mathbf{U}_X$ and $\mathbf{U}_N$ are maintained. To obtain the operation counts for these substeps, when the left-side orthogonal matrices are not maintained, simply eliminate all terms containing m from the given operation counts. Thus, in this case the CSNE-based expansion is used instead of the MGSR-based method. For our choice of parameters

- $m_X = m_N = 141$, $n = 20$ and $p = 12$.

the exact flop counts in Appendix A has to be used. This is shown in Table 7.6 for different portions of the ULLV algorithm. When calculating the number of operations for the deflation and refinement, a fixed numerical rank p is assumed (see Algorithm 7.7), so one step of each is required. Clearly, the expensive step is downdating, and the work increases a lot if $\mathbf{U}_X$ and $\mathbf{U}_N$ have to be calculated.

The total amount of work for the different cases is shown in Table 7.7, both for each sample and for one second of time with a sample rate $f_s = 8$ kHz. Note, that in the sliding window case,

Required	Algorithm	Flop count
$\mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Updating $\mathbf{X}$	8,465
	Downdating $\mathbf{X}$ (SNE or CSNE)	16,693 or 31,133
	Updating $\mathbf{N}$	8,694
	Downdating $\mathbf{N}$ (SNE or CSNE)	15,912 or 29,552
	Deflation (1 step)	5,783
	Refinement (1 step)	6,899
$\mathbf{U}_X, \mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Updating $\mathbf{X}$	24,680
	Downdating $\mathbf{X}$ (MGS or MGSR)	29,748 or 40,445
	Updating $\mathbf{N}$	24,768
	Downdating $\mathbf{N}$ (SNE or CSNE)	31,986 or 45,626
	Deflation (1 step)	15,935
	Refinement (1 step)	17,051
$\mathbf{U}_N, \mathbf{U}_X, \mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Updating $\mathbf{X}$	40,754
	Downdating $\mathbf{X}$ (MGS or MGSR)	45,822 or 56,519
	Updating $\mathbf{N}$	40,983
	Downdating $\mathbf{N}$ (MGS or MGSR)	45,841 or 56,538
	Deflation (1 step)	26,087
	Refinement (1 step)	27,203

Table 7.6 Flop count for substeps of the rank-revealing ULLV algorithm with $m_X = m_N = 141$, $n = 20$ and $p = 12$.

Required	ULLV algorithm	Expansion method	Number of ref.	Flop count	MFlop /second
$\mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Exponential window on $\mathbf{X}$	-	0	14,248	114
			1	21,147	169
	Sliding window on $\mathbf{X}$	SNE	0	36,724	294
			1	50,522	404
		CSNE	0	51,164	409
			1	64,962	520
$\mathbf{U}_X, \mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Exponential window on $\mathbf{X}$	-	0	40,615	325
			1	57,666	461
	Sliding window on $\mathbf{X}$	MGS	0	86,298	690
			1	120,400	963
		MGSR	0	96,995	776
			1	131,097	1,049
$\mathbf{U}_N, \mathbf{U}_X, \mathbf{L}_X, \mathbf{L}, \mathbf{V}$	Exponential window on $\mathbf{X}$	-	0	66,841	535
			1	94,044	752
	Sliding window on $\mathbf{X}$	MGS	0	138,750	1,110
			1	193,156	1,545
		MGSR	0	149,447	1,196
			1	203,853	1,631

Table 7.7 Flop count for the rank-revealing ULLV algorithm with $m_X = m_N = 141$, $n = 20$, $p = 12$ and $f_s = 8000$ sample/sek.

the deflation and refinement steps had to be used twice. Looking at the first part, where only $\mathbf{L}_X$, $\mathbf{L}$ and $\mathbf{V}$ are maintained, it is seen that computations can be saved by using the exponential window method, i.e., omitting the downdating step, and for a given choice of window, the refinement step can be left out. Thus, a given implementation of the rank-revealing ULLV algorithm is a tradeoff between complexity and accuracy. Finally, there is no numerical benefit in maintaining the left-side orthogonal matrices, so this can not be recommended considering the extra cost.

Another issue is the storage requirements which is roughly the decomposition elements, since $\mathbf{X}$ is needed when $\mathbf{U}_X$ is omitted. However, when $\mathbf{X}$ has a special structure, e.g., Toeplitz, the storage requirement for $\mathbf{X}$ can be much smaller than that for $\mathbf{U}_X$.

As comparison, the computational complexity of the QSVD algorithm is 1.8 Mflop, when implemented along the lines described in [120] and with the chosen parameters. If the QSVD is applied to non-overlapping frames of length 20 ms, the computational count is 90 Mflop per second ($f_s = 8$ kHz). Thus, the recursive RRULLV algorithm is more expensive, when updated for every new sample.

7.9 Error Analysis

In the following rounding error analysis, the indexed scalars ϵ_i will denote nonnegative numbers bounded above by the product of the machine unit roundoff μ and small constants dependent only on the number of orthogonal transformations and possibly the dimension of the problems. Calligraphic letters are used for the computed versions of their roman counterparts.

7.9.1 Plane Rotations

In the ULLV algorithm, the basic operations are plane rotations $\mathbf{P}$ and $\mathbf{Q}$ applied from the left and right to a given matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, so that ideally

$$\mathbf{Y} = \mathbf{P}^T \mathbf{X} \mathbf{Q} \tag{7.177}$$

The numerical properties of plane rotations follows from the work of Wilkinson [127], i.e., it can be shown that the computed rotations $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ satisfy

$$\|\mathbf{P} - \boldsymbol{\mathcal{P}}\|_2 = \epsilon_P \quad \text{and} \quad \|\mathbf{Q} - \boldsymbol{\mathcal{Q}}\|_2 = \epsilon_Q \tag{7.178}$$

and that the use of plane rotations are *backward stable*, so the computed matrix $\boldsymbol{\mathcal{Y}}$ is the exact update of a nearby matrix

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{P}}^T \mathbf{X} \boldsymbol{\mathcal{Q}} = \mathbf{P}^T (\mathbf{X} + \mathbf{E}) \mathbf{Q} \, , \qquad \|\mathbf{E}\|_2 = \epsilon_X \|\mathbf{X}\|_2 \tag{7.179}$$

7.9.2 Updating

Now, consider the case of updating $\mathbf{U}_X$, $\mathbf{L}_X$, $\mathbf{L}$, and $\mathbf{V}$ in the RRULLVD in an efficient and numerically stable way, when a row $\mathbf{x}^T$ is appended to $\mathbf{X}$. Thus, the problem is

$$\begin{pmatrix} \mathbf{L}_X \mathbf{L} \\ \mathbf{x}^T \mathbf{V} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_X^T & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{x}^T \end{pmatrix} \mathbf{V} \tag{7.180}$$

which determine the updated matrices

$$\hat{\mathbf{L}}_X \hat{\mathbf{L}} = \hat{\mathbf{U}}_X^T \hat{\mathbf{X}} \mathbf{V} \tag{7.181}$$

Since this equation is a continuation of (7.177), the computational results satisfy the equivalent of (7.178) and (7.179), i.e., the updating techniques are numerically *backward stable* (see Paige [89]). Note that (7.180) require $\mathbf{V}$ to carry out the update, and that $\mathbf{V}$ is unaltered, while $\mathbf{U}_X$ is trivially modified.

The analysis is independent of the form and rank of $\mathbf{X}$ and $\mathbf{L}_X\mathbf{L}$, but if $\mathbf{L}_X\mathbf{L}$ have a specific form (lower triangular), then the updated matrix $\hat{\mathbf{L}}_X\hat{\mathbf{L}}$ is expected to have the same form. In such cases the form can be efficiently regained by applying more numerically stable transformations from the left and right to the relevant parts of (7.180), which will just be further continuations of (7.177) and as such will make the computations backward stable.

7.9.3 Downdating

The downdate analysis is more complex. Assume that the expansion step has been applied to the matrix $\mathbf{U}_X$ in the RRULLVD, i.e.,

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_X & \mathbf{u}_{n+1} \end{pmatrix} \begin{pmatrix} \mathbf{L}_X\mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T \tag{7.182}$$

where the appended column $\mathbf{u}_{n+1}$ satisfy the requirements discussed in Section 7.4.2. Further assume that the transformations $\mathbf{U}_X$ and $\mathbf{V}$ are available whenever needed, so the less reliable methods that use $\mathbf{X}$ to find information on, e.g., $\mathbf{U}_X$ is not discussed (see Section 7.4.2). Thus, for the MGSR based expansion the analysis in Parlett [92, page 107] ensures that

$$\|\mathbf{u}_{n+1} - \boldsymbol{\mathcal{U}}_{n+1}\|_2 = \epsilon_u \tag{7.183}$$

The problem of removing a row $\mathbf{x}^T$ from $\mathbf{X}$ is now obtained by applying an orthogonal transformation $\mathbf{P}$ so

$$\begin{pmatrix} \mathbf{U}_X & \mathbf{u}_{n+1} \end{pmatrix} \mathbf{P} = \begin{pmatrix} \mathbf{0}^T & 1 \\ \hat{\mathbf{U}}_X & \mathbf{0} \end{pmatrix} \tag{7.184}$$

Clearly, $\hat{\mathbf{U}}_X$ has orthonormal columns and the downdated matrices are determined by

$$\begin{aligned} \mathbf{P}^T \begin{pmatrix} \mathbf{L}_X\mathbf{L} \\ \mathbf{0}^T \end{pmatrix} &= \mathbf{P}^T \begin{pmatrix} \mathbf{U}_X^T \\ \mathbf{u}_{n+1}^T \end{pmatrix} \mathbf{X}\mathbf{V} \\ &= \begin{pmatrix} \mathbf{0} & \hat{\mathbf{U}}_X^T \\ 1 & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} \mathbf{V} \\ &= \begin{pmatrix} \hat{\mathbf{U}}_X^T\hat{\mathbf{X}}\mathbf{V} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \\ &= \begin{pmatrix} \hat{\mathbf{L}}_X\hat{\mathbf{L}} \\ \mathbf{x}^T\mathbf{V} \end{pmatrix} \end{aligned} \tag{7.185}$$

The analysis by Paige in [89] shows that the computed version of $\hat{\mathbf{U}}_X$ is close to an orthogonal matrix

$$\|\hat{\mathbf{U}}_X - \hat{\boldsymbol{\mathcal{U}}}_X\|_2 = \epsilon_U \tag{7.186}$$

and that

$$\hat{\boldsymbol{\mathcal{L}}}_X\hat{\boldsymbol{\mathcal{L}}} = \hat{\boldsymbol{\mathcal{U}}}_X^T\hat{\mathbf{X}}\boldsymbol{\mathcal{V}} = \hat{\mathbf{U}}_X^T(\hat{\mathbf{X}} + \hat{\mathbf{E}})\mathbf{V}\ , \qquad \|\hat{\mathbf{E}}\|_2 = \epsilon_X\|\mathbf{X}\|_2 \tag{7.187}$$

Notice that (7.186) and (7.187) have the same form as (7.178) and (7.179). However, the bound in (7.187) depends on $\mathbf{X}$ rather than just $\hat{\mathbf{X}}$, so if $\|\mathbf{x}^T\|_2 \gg \|\hat{\mathbf{X}}\|_2$, the new decomposition might not be as accurate as if it has been computed directly from $\hat{\mathbf{X}}$, i.e., the downdating is as numerically stable as can be expected for the problem.

Again, $\mathbf{V}$ is unaltered, while the modification of $\mathbf{U}_X$ is nontrivial. Finally, numerically stable transformations can be applied to the left and right to give $\hat{\mathbf{L}}_X\hat{\mathbf{L}}$ any desired form (lower triangular).

7.9.4 Sequential Updates and Downdates

Stewart [111] has shown that *sequential* updates and downdates of two-sided orthogonal decompositions such as the ULV decomposition gives acceptable-error stable results, when the condition numbers of the triangular factors are substantially less than $1/\sqrt{\mu}$. The analysis is based on downdating algorithms like the LINPACK, i.e., the CSNE-algorithm discussed in Section 7.4.2.2 can reasonably be expected to perform better, and methods that maintain the $\mathbf{U}$-factors will have superior numerical properties.

In some applications, exponential windowing is an alternative to (sequential) downdating. Stewart [108] has shown that updating the ULV decomposition using exponential windowing is unconditionally stable in the presence of rounding errors. Thus, exponential windowing damps old rounding errors along with old data.

7.9.5 Maintaining Orthogonality

Unfortunately, it is usually required to accumulate the right-side transformation $\mathbf{V}$ in the ULV/ULLV algorithms, and here errors can accumulate very slowly. Algorithms for maintaining orthogonality in long products of orthogonal matrices, which arise, for example, in subspace tracking problems in signal processing, are analysed by Moonen et al. [79], Edelman and Stewart [28] and Mathias [76].

Let $\mathbf{V}_k \in \mathbb{R}^{n\times n}$ be iteratively updated according to a product of the form

$$\mathbf{V}_k = \mathbf{V}_{k-1}\mathbf{Q}_k = \mathbf{Q}_1 \cdots \mathbf{Q}_{k-1}\mathbf{Q}_k \tag{7.188}$$

where the matrices $\mathbf{Q}_i$ are orthogonal, then rounding errors will cause $\mathbf{V}_k$ to drift from orthogonality with increasing k. The deviation of $\mathbf{V}_k^T\mathbf{V}_k$ from the identity is bounded as [92]

$$\|\mathbf{I}_n - \mathbf{V}_k^T\mathbf{V}_k\|_F \leq (k+1)n^{1.5}\mu \tag{7.189}$$

where μ is the machine precision. In [79] is pointed out that simply *normalizing* the columns $\mathbf{v}_i$ of the product after each update preserve orthogonality, i.e.,

$$\mathbf{v}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2}, \qquad i = 1, \dots, n \tag{7.190}$$

The normalization technique is analyzed in [28], where it is shown that the method succeeds when the matrix $\mathbf{Q}_i$ manage to transfer off-diagonal error in the matrices $\mathbf{I}_n - \mathbf{V}_i^T\mathbf{V}_i$ to the diagonal, i.e., the method will fail only in certain unlikely circumstances.

From the point of view of accuracy, i.e., how close the computed product is to the true product, this algorithm is actually better than some of the more complicated algorithms (see [76] for details).

The normalization of $\mathbf{V}$ requires $3n^2$ flop, so it is too costly to apply it after every rotation on $\mathbf{V}$. The compromise proposed here, is to normalize $\mathbf{V}$ after a complete updating, downdating and rank-revealing of the RRULLV algorithm.

As an example, the RRULLV algorithm has been applied to a sinosoid in colored Gaussian noise (AR(1,-0.7) process and SNR = 10 dB). The RRULLVD of the data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ is up/downdated using a sliding window consisting of 160 samples, and a signal subspace dimension of two. The $\mathbf{U}$-matrices are not maintained and no refinement is used. When no intermediate normalizations are performed, orthogonality is gradually lost as shown in Figure 7.5. When the columns in $\mathbf{V}$ are normalized after each shift of the sliding window, the accumulation is apparently stable, where the level may depends on the number of rotations in between normalizations of $\mathbf{V}$. Thus, the conclusion is that the accumulated error is acceptable and there is no reason to use more complicated methods.

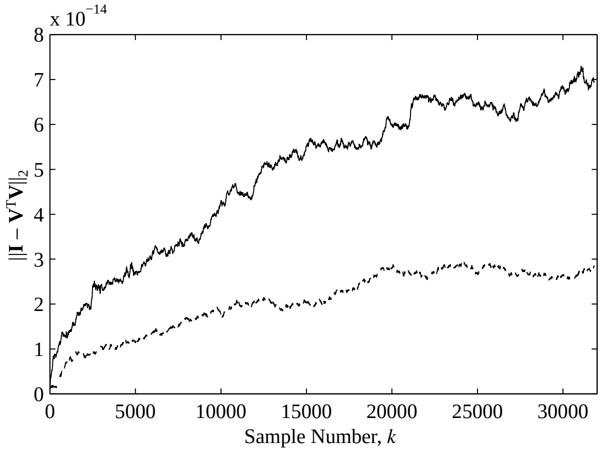


Figure 7.5 Orthogonality measure $\|\mathbf{I}_n - \mathbf{V}^T\mathbf{V}\|_2$, where $\mathbf{V}$ is obtained from the RRULLV algorithm applied to a sinosoid in colored Gaussian noise (AR(1,-0.7) process and SNR = 10dB). The RRULLVD of the data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ is up/downdated using a sliding window consisting of 160 samples. The solid graph is with no intermediate normalizations and the dashed is when the columns in $\mathbf{V}$ are normalized after each shift of the sliding window.

7.10 Numerical Results

In this section, numerical experiments using the rank-revealing ULLV algorithm are presented. The algorithm is implemented in MATLAB M-files, and a C-version with a MEX-file interface (see Appendix B), i.e., the machine precision is $\mu = 2.22 \times 10^{-16}$.

7.10.1 Test Matrix

The first group of experiments generates the data matrix $\mathbf{X}$ by using a sliding window method on a test matrix $\mathbf{W}$, i.e., in step k, the new row $\mathbf{w}_k^T$ is updated into the RRULLVD and the existing row $\mathbf{w}_{k-m}^T$ is downdated from the decomposition, where m is the window size. The following set of test matrices is also used for test purposes in [67].

Test matrix I Test matrix $\mathbf{W}_I$ is a 24×5 matrix composed of three 8×5 parts $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$. These matrices are constructed with elements chosen as random numbers from the uniform distribution on [0;1]. In order to vary the numerical rank of the data matrix, $\mathbf{W}_1$ and $\mathbf{W}_3$ is given 2 singular values of 10^{-12} while $\mathbf{W}_2$ is given 3 singular values of 10^{-12}. The window size m is 5 and as a tolerance τ for determining the numerical rank is used 10^{-9}.

Test matrix II Test matrix $\mathbf{W}_{II}$ is a 30×7 matrix with a window size m of 9. The first 9 rows are generated randomly. Each of the succeeding rows is a weighted combination of a fixed random row with a new random row. The weight assigned to the new component begins as 0.1 and is divided by 10 at each succeeding row. The numerical tolerance τ is again 10^{-9}.

The matrix $\mathbf{N}$ is made up of elements selected from a normal distribution and is not changed while matrix $\mathbf{X}$ is modified.

The first experiment test the algorithms ability to estimate the rank p of the data matrix based on the numerical tolerance τ as shown in Figure 7.6, where the QSVD of the data matrix is used as a reference in checking the accuracy. The MGSR based method gives the true rank estimates for both test matrices, while the CSNE based method sometimes fail to properly estimate the rank. Notice that this was also observed for the algorithm in [67]. The CSNE method can be improved by increasing the number of correction steps in the CSNE algorithm, however, the rank estimation problem is not important, when the RRULLV algorithm is used on signals with known rank.

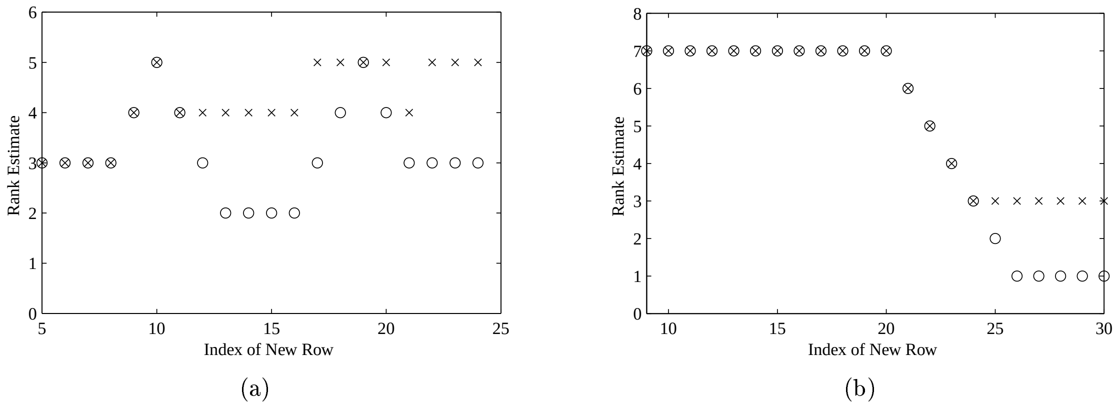


Figure 7.6 Rank estimates obtained with matrix $\mathbf{W}_I$ (a) and matrix $\mathbf{W}_{II}$ (b) using the MGSR method (○) and CSNE method (×). The estimates based on the MGSR method correspond to the true values as given by the QSVD.

7.10.2 Test Signals

The error in the decomposition is measured as $\|\mathbf{X} - \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{V}^T\|_2$, i.e., the accuracy of the decomposition is not tested for the CSNE-based downdating, as the full decomposition is not available. Figure 7.7 shows the decomposition error of the example in Figure 7.5, where the error can be explained by the accumulated errors in $\mathbf{V}$.

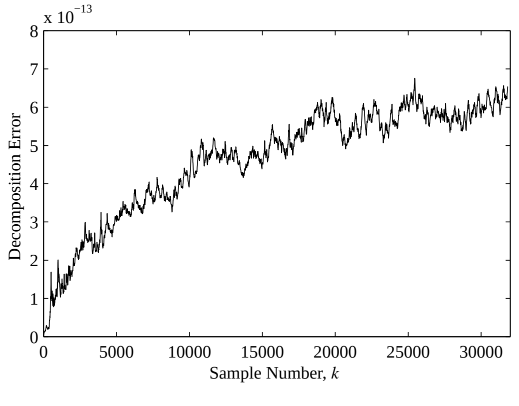


Figure 7.7 Decomposition error $\|\mathbf{X} - \mathbf{U}_X\mathbf{L}_X\mathbf{L}\mathbf{V}^T\|_2$, when the RRULLV algorithm (MGSR) is applied to a sinosoid in colored Gaussian noise (AR(1,-0.7) process and SNR = 10dB). The RRULLVD of the data matrix $\mathbf{X} \in \mathbb{R}^{141\times 20}$ is up/downdated using a sliding window consisting of 160 samples.

7.11 Summary

A complete RRULLV algorithm has been developed, analyzed and implemented. The updating step is straightforward, while the critical step is the downdating used in the sliding window method.

Two expansion methods have been considered, and it has been shown that the orthogonalization will fail if the rank of the downdated matrix decreases. The CSNE-based expansion is used when the left-side orthogonal matrices are not maintained, and here, the correction step must always be performed to stabilize the method. However, even in this case, it has been observed that the algorithm sometimes fail to properly estimate (track) the rank.

When the signal represents a stationary process, exponential windowing is to be preferred to downdating. It is simpler and has better numerical properties. However, in nonstationary situations the two techniques produce different results, and the decision between the two must depend on their behavior in the application in question.

Deflation is another important step, where in speech applications, the Hager condition estimator is shown to be a good choice. Finally, it is noted that the refinement step can be left out in this application.

The computational complexity of the RRULLV algorithm is high, and further work need to be done in order to reduce it. This could, e.g., be obtained by introducing rank-k up/downdating.

Chapter 8

Experimental Speech Enhancement

In this chapter, the discussed nonparametric signal subspace based estimators are used for speech enhancement, and comparisons are made on the basis of the improvement in quality (SNR), tracking capability, and informal listening tests.

The experiments will illustrate the differences in speech enhancement that may arise from the use of different estimation strategies, decomposition methods (SVD or RRULVD) and window types (sliding or exponential). Also the effect of prewhitening is evaluated.

8.1 Experimental Setup

All algorithms are programmed in Matlab, however, the rank-revealing ULV/ULLV algorithms are partly implemented in C to reduce execution time. The files are collected in two Matlab toolboxes provided for free (see Appendix B and C).

The practical implementation of the algorithms follows the lines in Section 4.7 and 6.4, and the experiments have been performed by using the following algorithm settings, if not otherwise stated:

- A rectangular analysis window consisting of 160 samples is used, and in the frame based case, without overlap between adjacent frames.
- The noise matrix $\mathbf{N}$ (or σ_{noise}) is only updated in periods without speech, i.e., from an initial noise-only segment.
- The estimators are construced from the subspace decomposition of Toeplitz data matrices having dimensions $(m, n) = (141 \times 20)$, and by using a fixed signal subspace dimension $p = 12$.
- The left-side orthogonal matrices $\mathbf{U}_i$ are not maintained in the decompositions, and therefore not used to construct the estimators.
- For the RRULVD based estimates, no refinement steps have been applied to improve the subspace quality.

As in the previous chapters, the speech material used in the examples, is the phonetically balanced reference sentence shown in Figure 2.1. However, the obtained results are fairly general, and from other projects, the sentence is known to be representative.

Finally, the experiments have been made with the clean speech signal added both artificial white noise and colored noise.

8.2 Speech Contaminated by White Noise

In this section, the different noise reduction methods are applied to speech signals contaminated by white noise (global SNR = 5 dB). First, the SVD-based algorithm is used to compare estimation strategies, and then the application of the ULV-based algorithm illustrates the benefits of the recursive approach.

The noisy speech sentence is shown in Figure 8.1, and Figure 8.2 – 8.4 show examples of enhanced speech signals obtained by the LS, MV and SDC estimator, respectively. During non-speech activity periods, it is observed that the Wiener-based gain functions achieve a better noise reduction than the LS method due to the nulling of components having low spectral SNR. Furthermore, the SDC estimator ($\beta_2 = 5$) seems to have the best performance, but speech distortion can hardly be seen from this type of plot.

The segmental SNRs of the noisy speech signal and the enhanced speech signal obtained by the SDC estimator ($\beta_2 = 5$) are shown in Figure 8.5. A shaded plot of the clean speech signal with arbitrary placement and amplitude is also given in order to evaluate the results (will in general be used on SNR plots).

From the figure, it is seen that the segmental SNRs have been improved in most cases.

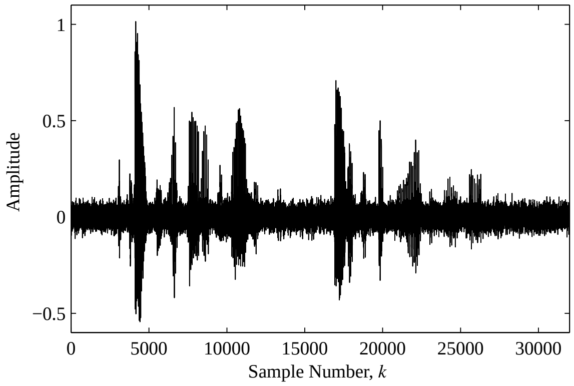


Figure 8.1 Noisy speech sentence contaminated by white noise (SNR=5dB).

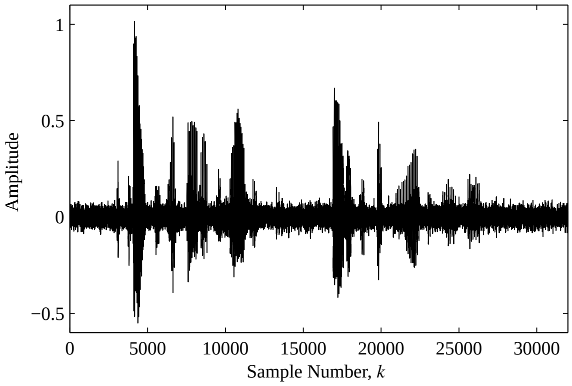


Figure 8.2 Enhanced speech signal obtained by the LS estimator.

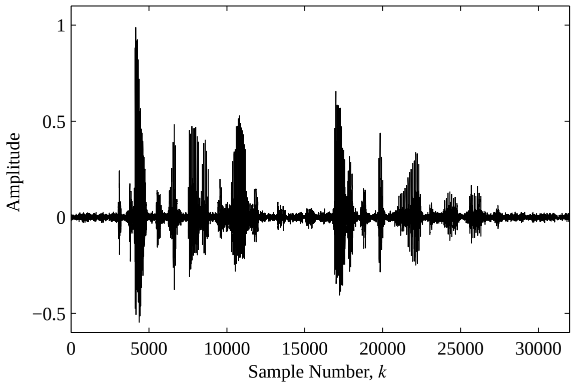


Figure 8.3 Enhanced speech signal obtained by the MV estimator.

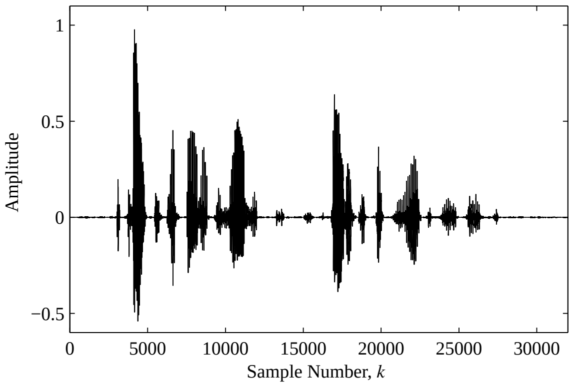


Figure 8.4 Enhanced speech signal obtained by the SDC estimator ($\beta_2 = 5$).

Only frames with high SNR (close to 20 dB) will not be enhanced due to the signal distortion obtained by introducing a signal subspace. Note also that the variations among the segmental SNRs are reduced, and that the SNRs of the enhanced signal are mainly above 0 dB. The latter observation rely on the actual gain function, which sets spectral components below 0 dB to zero (see Figure 4.2(b)).

Figure 8.6 illustrates the improvements in segmental SNRs for the enhanced waveforms shown in Figure 8.2 – 8.4. The LS estimator gives a nearly constant improvement n/p as expected from the p-dimensional signal subspace, while the two other methods perform considerably better. At low SNRs, the improvements obtained by the SDC estimator ($\beta_2 = 5$) are significantly larger than the ones obtained by the MV estimator, which can be explained by the practical behavior of the estimators (see Figure 4.4(a) and 4.5(b)). Thus, the SDC esimator is closer to the theoretical value. At high SNRs, both estimators approximately gives the theoretical values, and the MV estimator will therefore be slightly better, since it results in the minimum residual signal.

Figure 8.7 shows the input-output relations of segmental SNRs for the MV and SDC ($\beta_2 = 5$) estimators, i.e., the segmental SNRs of the enhanced speech signal as function of the segmental SNRs of the noisy signal. Clearly, the improvement in output SNR increases for decreasing input

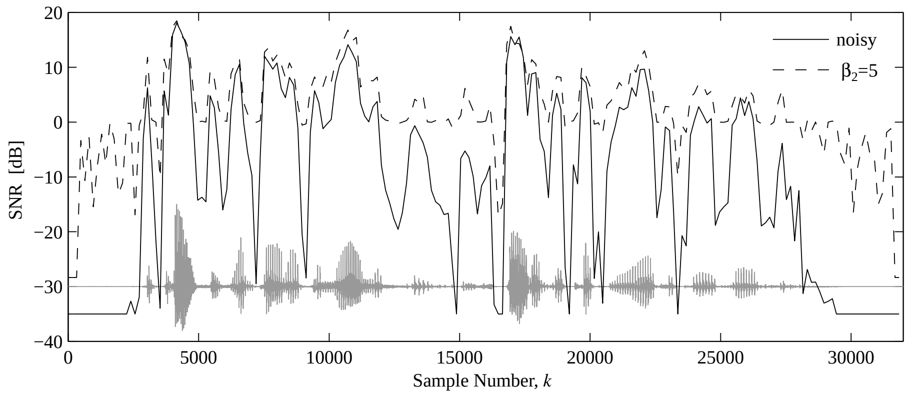


Figure 8.5 Segmental SNRs of the noisy signal and the enhanced waveforms shown in Figure 8.1 and 8.4.

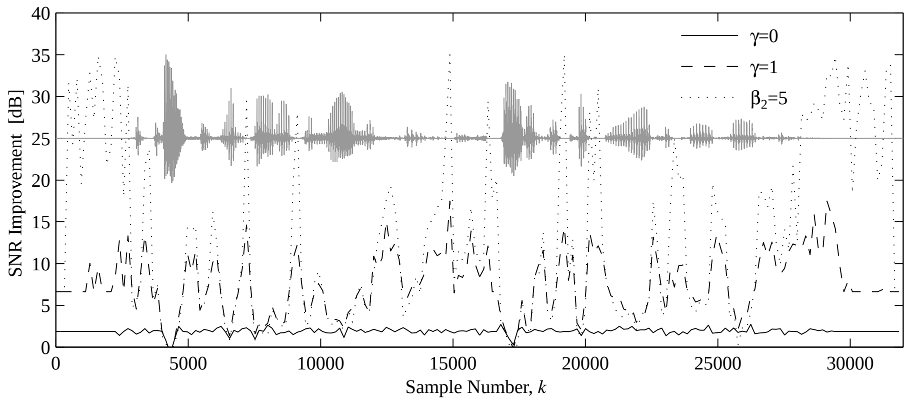


Figure 8.6 Improvement in segmental SNRs for the enhanced waveforms shown in Figure 8.2 – 8.4.

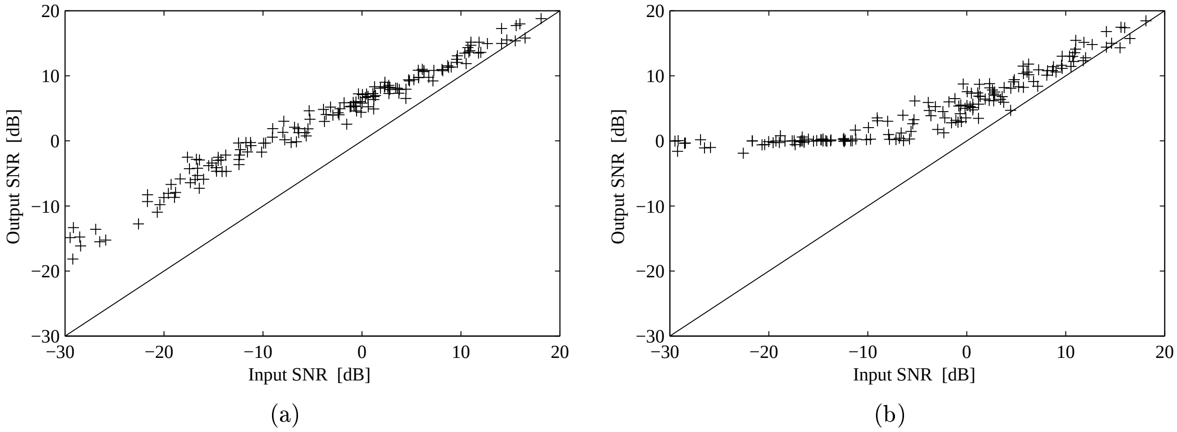


Figure 8.7 Segmental SNRs of the enhanced speech signal as function of the segmental SNRs of the noisy signal. (a) Using the MV estimator. (b) Using the SDC estimator ($\beta_2 = 5$).

SNR, and no improvement can be expected in frames with SNR close to 20 dB. Note again the 0 dB limit for the SDC estimator.

Figure 8.8 shows the improvements in segmental SNRs, when the RRULVD-based algorithm is used to enhance the noisy speech signal in Figure 8.1. The results are close to the one obtained by using the SVD, so the difference between the two approaches ($\text{SNR}_{RRULVD}/\text{SNR}_{SVD}$) is magnified in Figure 8.9 for the MV and TDC based estimation, respectively. Obviously, the recursive RRULVD method gives the best results, when there is a change in the dynamics of the signal, while the frame-based SVD approach is more accurate in stationary periods.

The same observation is made, when the RRULVD-based algorithm using a sliding window is compared with the one based on an exponential window (forgetting factor $\beta = 0.99$). The differences in segmental SNRs ($\text{SNR}_{sli}/\text{SNR}_{exp}$) are shown in Figure 8.10 for the two estimation strategies.

Thus, when signal subspace methods are applied to speech signals, there is an argument for both introducing a recursive approach like the RRULV algorithm, and for using a sliding window, which requires a numerically complex and computationally expensive downdating step.

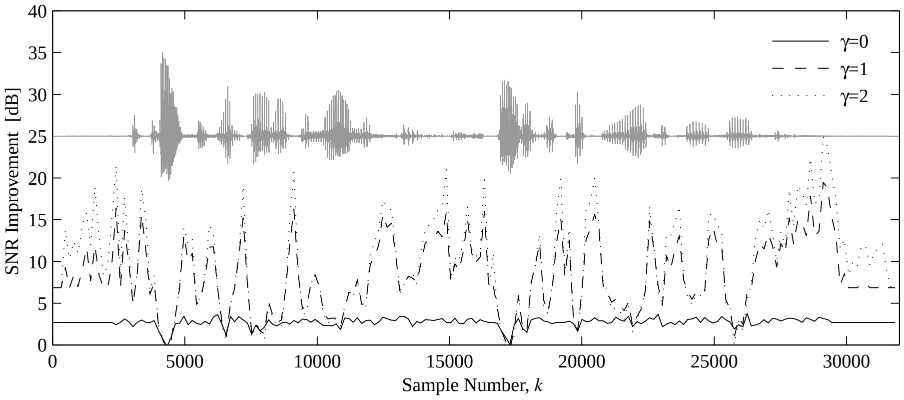
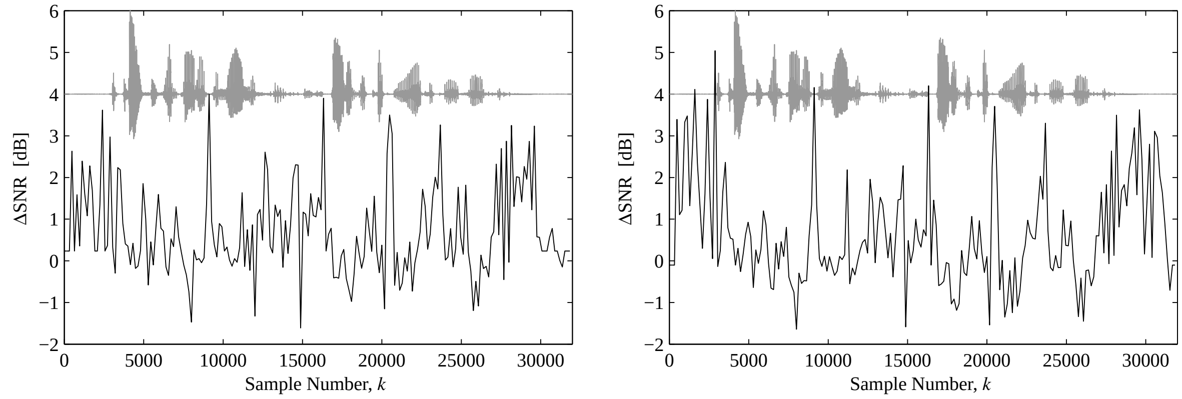
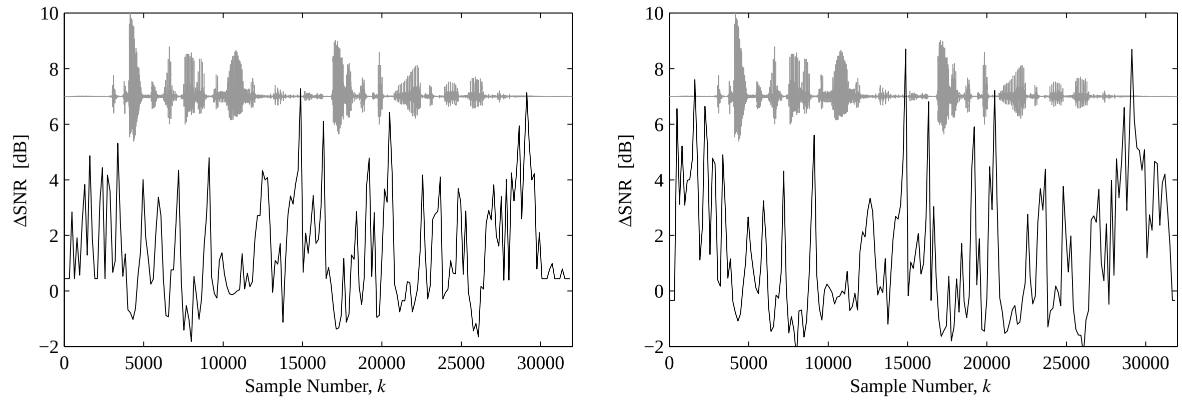


Figure 8.8 Improvement in segmental SNRs for the RRULVD based estimators.



(a) (b)

Figure 8.9 Difference between segmental SNRs of enhanced speech obtained by the RRULVD and the SVD, i.e., $\mathrm{SNR}_{RRULVD}/\mathrm{SNR}_{SVD}$. (a) Using the MV estimator. (b) Using the TDC estimator ($\gamma = 2$).



(a) (b)

Figure 8.10 Difference between segmental SNRs of enhanced speech obtained by the RRULVD using a sliding and exponential window ($\beta = 0.99$), respectively, i.e., $\mathrm{SNR}_{sli}/\mathrm{SNR}_{exp}$. (a) Using the MV estimator. (b) Using the TDC estimator ($\gamma = 2$).

8.3 Speech Contaminated by Colored Noise

When the speech signal is added the colored AR(1,-0.7) noise process (global SNR = 5 dB), the QSVD-based algorithm is used in order to include prewhitening. The method is compared with the SVD-based approach and the RRULLVD-based algorithm.

Figure 8.11 gives examples of enhanced speech signals obtained by using the SDC estimator ($\beta_2 = 5$), based on the SVD and the QSVD, respectively. If the enhanced waveforms are compared with the white noise case in Figure 8.4, it is observed that the results are more noisy. For the SVD case, the increased noise level rely on the fact that the eigenvectors of the correlation matrix of the clean signal and the one of the noisy signal are no longer equal, resulting in a bias of the estimator. This is indeed the argument for introducing prewhitening, however, then a bias of the signal subspace is obtained as discussed in Section 3.6, explaining the reduced performance of the estimator compared with the white noise case.

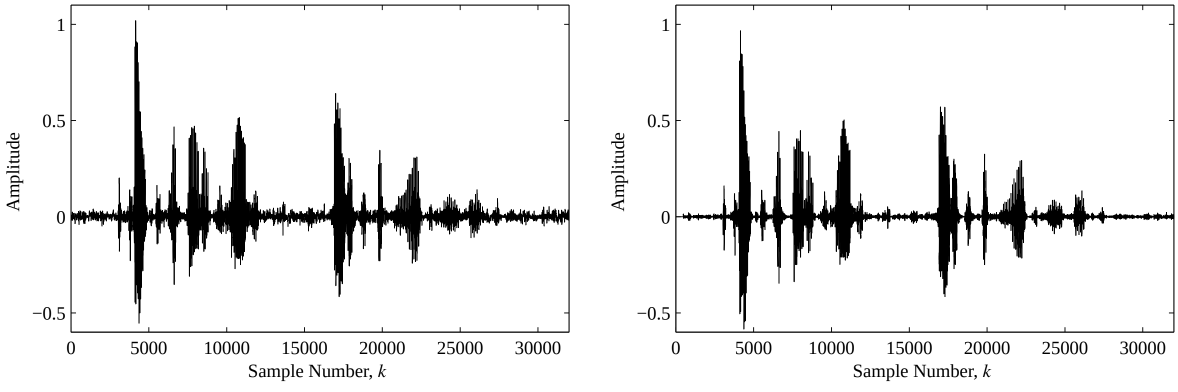


Figure 8.11 Enhanced speech signal obtained by the SDC estimator ($\beta_2 = 5$). (a) Using the SVD. (b) Using the QSVD.

Figure 8.12 illustrates the difference between SNR improvements obtained by estimators based on the QSVD and estimators based on the SVD, i.e., $\mathrm{SNR}_{QSVD}/\mathrm{SNR}_{SVD}$. Thus, for most frames, the QSVD approach with integrated prewhitening delivers the best result, so in spite of the discussed drawbacks, it is still better to use a signal subspace method with prewhitening, than without.

Another issue is to evaluate the amount of noise reduction, which can be obtained by the QSVD-based algorithm. The improvement plots in Figure 8.13 demonstrates a significant lower performance compared with the white noise case in Figure 8.6. Even larger degradations are observed for frames with high segmental SNR. The overall reduction in performance rely on the actual noise process, which is dominated by low frequencies, since a majority of speech frames also have this type of spectral distribution. However, the example is assumed to be representative, because many acoustical noise scenarios are of the type, considered here.

Figure 8.14 shows the input-output relations of segmental SNRs for the MV and SDC ($\beta_2 = 5$) estimators corresponding to the examples in Figure 8.7. Clearly, the improvements in output SNR have decreased, so the frames with high input SNRs are now degraded, for a broad range

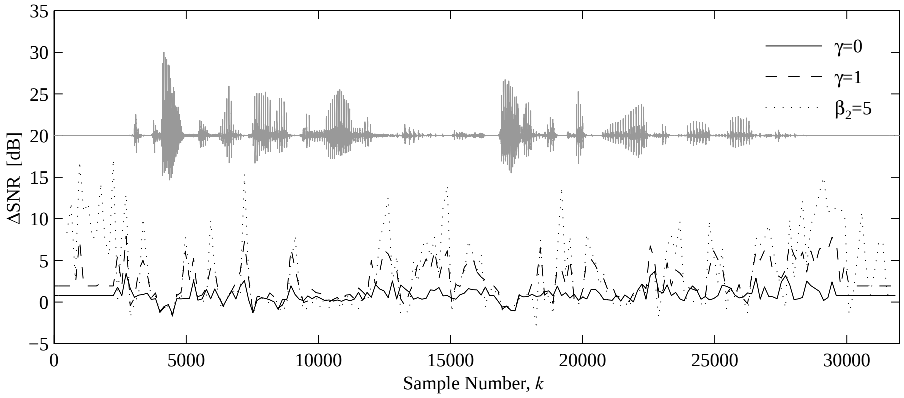


Figure 8.12 Difference between segmental SNRs of estimates obtained by using the QSVD and SVD algorithms, i.e., $\mathrm{SNR}_{QSVD}/\mathrm{SNR}_{SVD}$.

of input SNRs, small (or none) improvements are obtained, and only for low input SNRs, noise reduction comparable with the white noise case are obtained.

The difference between the RRULVD-based and QSVD-based algorithm, i.e., the ratio $\mathrm{SNR}_{RRULLVD}/\mathrm{SNR}_{QSVD}$, is magnified in Figure 8.9 for the MV and TDC based estimators, respectively. As in the white noise case, the tracking capabilities of the RRULLVD method are demonstrated to outperform the QSVD method, while the latter is a better approach in stationary periods. However, the variations between the two methods are larger in the colored noise case.

Finally, the difference between estimates obtained by using the RRULLVD without and with the left-side orthogonal matrices $\mathbf{U}$ is considered. This is demonstrated in the Figure 8.16, where the quantity $\mathrm{SNR}_{without\ U}/\mathrm{SNR}_{with\ U}$ is shown for the TDC ($\gamma = 2$) based enhanced speech. Obviously, there is no benefit in maintaining the $\mathbf{U}$ matrices.

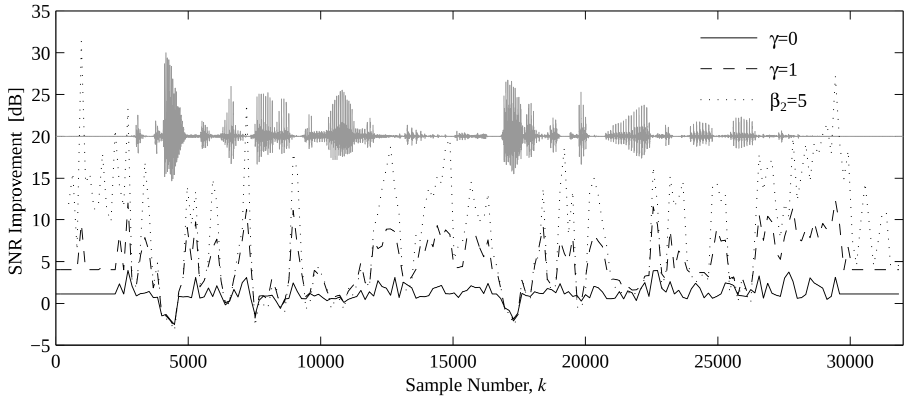


Figure 8.13 Improvement in segmental SNRs for the QSVD based estimators.

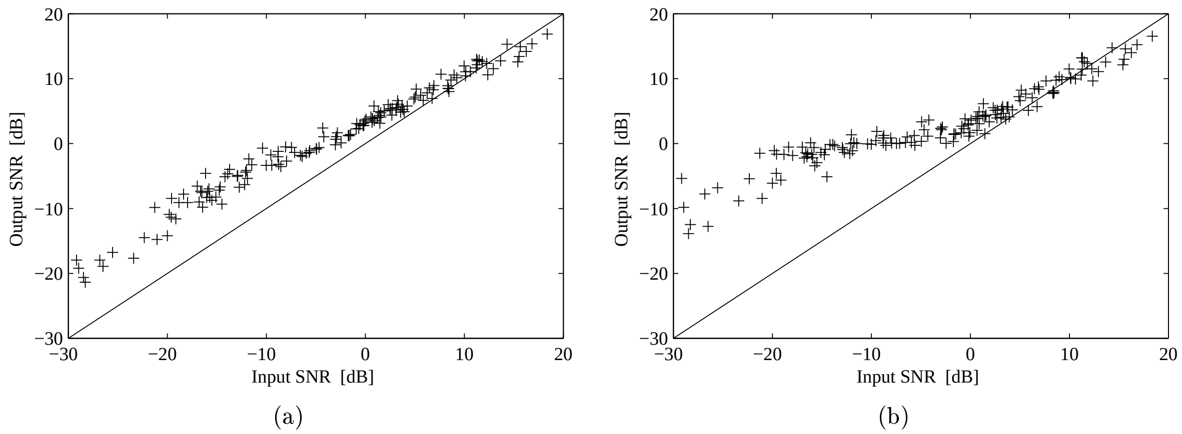
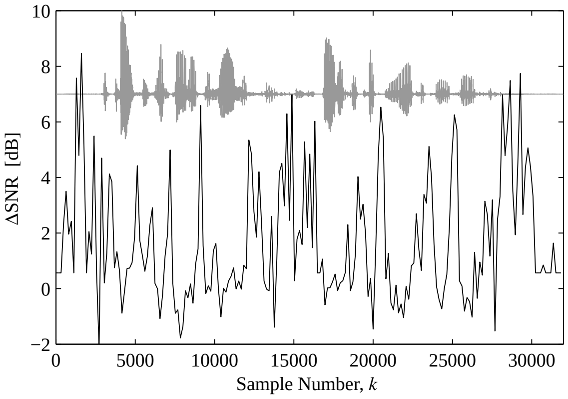
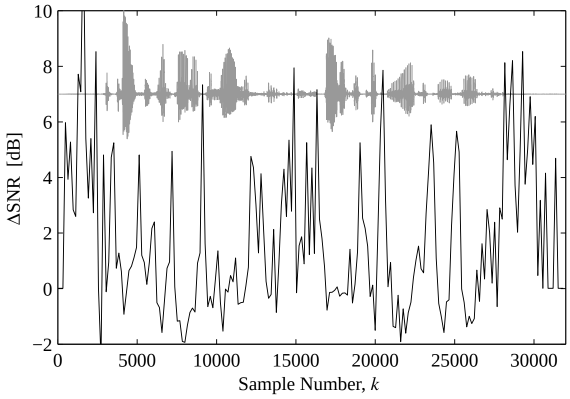


Figure 8.14 Segmental SNRs of the enhanced speech signal (QSVD based) as function of the segmental SNRs of the noisy signal. (a) Using the MV estimator. (b) Using the SDC estimator ($\beta_2 = 5$).



(a)



(b)

Figure 8.15 Difference between segmental SNRs of enhanced speech obtained by the RRULLVD and the QSVD, i.e., $\mathrm{SNR}_{RRULLVD}/\mathrm{SNR}_{QSVD}$. (a) Using the MV estimator. (b) Using the TDC estimator ($\gamma = 2$).

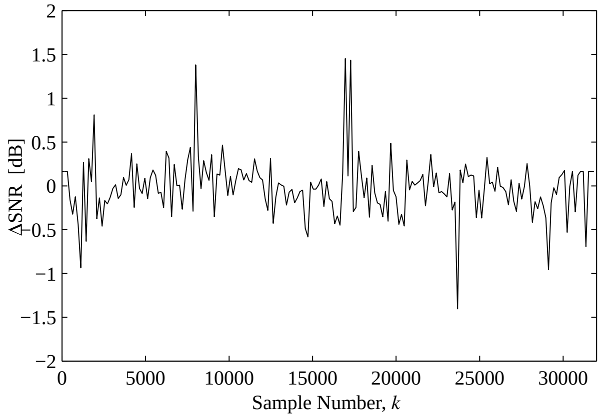


Figure 8.16 Difference beween segmental SNRs of TDC ($\gamma = 2$) based enhanced speech using the RRULLVD without and with the left-side orthogonal matrices $\mathbf{U}$, i.e., $\mathrm{SNR}_{without\ U}/\mathrm{SNR}_{with\ U}$.

8.4 Informal Listening Tests

Informal listening tests have been carried out for a number of speech sentences corrupted by white and colored noise. At higher noise levels (global SNR < 10 dB), the enhanced speech signals obtained by the LS and MV methods are seriously affected by the musical noise. For the TDC and SDC estimators, the informal listening tests confirm that musical noise and/or audible distortions are still present in the processed speech. For example, the SDC estimator with $\beta_2 = 5$ results in enhanced speech almost free of musical noise, but with a significant distortion of the speech.

In the case with colored noise, the audible speech distortion has increased and the musical noise is now dominated by low frequencies as discussed in Section 4.2.1.

If the computational complexity is of no importance, the musical noise can be reduced by increasing the frame length K and space dimension n, e.g., $K = 400$ and $n = 40$. Thus, the

SNR improvement n/p expected from the existence of the p-dimensional signal subspace is better utilized. However, since the complexity of the considered algorithms are at least $\mathcal{O}(n^2)$, better results can be obtained by using the proposed multi-microphone combination in Section 4.4, where the complexity is $\mathcal{O}(Ln^2)$. In this case, the number of microphones L increases, instead of n. Informal listening tests using four microphones confirm that the enhanced speech are almost free of both musical noise and distortions.

8.5 Summary

The subspace-based noise reduction algorithms have been applied successfully to continuous speech embedded in white noise as well as colored broad-band noise.

It has been demonstrated that the SVD-based signal subspace approach is able to achieve satisfactory improvements in the speech quality. Furthermore, arguments have been given for both introducing a recursive approach like the RRULV algorithm, and for using a sliding window, which requires a numerically complex and computationally expensive downdating step.

In the colored noise case, the performance is highly dependent on the noise statistics. Thus, a noise process dominated by the same frequencies as the speech, will results in a less reliable algorithm (typically with more speech distortion). Also here, a recursive approach like the RRULLV algorithm can be introduced successfully.

Chapter 9

Conclusion and Topics for Further Research

A study of subspace based speech enhancement techniques has been carried out having in mind applications in mobile telephone environments.

One purpose of the work has been to characterize subspace methods when applied to speech signals. The result of the preceding analyses is that no gab exists in the eigenspectrum of the short-term correlation matrix of the speech signal as could be expected from the fundamentals of the speech production system. However, the quality of the speech is related to the formants, represented by the dominating eigenvalues, so it still makes sense to define a signal subspace. It has been shown that projection onto the signal subspace can be interpreted as filter oprations, removing the spectral components of the noisy spectrum having the lowest Signal-to-Noise Ratio (SNR).

When the speech signal is corrupted by additive white noise, the signal and noise subspace are blurred together, but this is not crucial for speech signals as long as the formants are maintained in the signal subspace. However, in the case of colored noise, it has been demonstrated that this can not be guaranteed, since the effect of prewhitening is a noise dependent bias of the speech components. Thus, signal subspace methods are sensitive to the second order statistics of the noise.

The conditions that allows us to derive the speech signal from the SVD of the noisy data matrix have been discussed, and assuming stationarity, the conclusion is that both matrix dimensions should approach infinity in order to obtain the best performance. This is not possible due to the non-stationarity of speech signals, and considerations concerning a proper choice of dimensions have been given.

Projection onto a signal subspace is a widely used method to reduce the dimensionality of the considered problem and thereby removing the influence of noise components. However, from a noise reduction point of view, this is not sufficient. If information about the noise statistics can be obtained, combination of the signal subspace approach with classical estimation strategies like Wiener filtering will in general have better performance. This is indeed the case in nonparametric speech enhancement. Another issue is the annoying tonal characteristics of the resulting residual noise (musical noise). Thus, constrained estimation methods must be used in order to mask this component. The price paid is an increased speech distortion.

The contribution here, is a unified presentation of the discussed estimation methods, formulated by means of both the eigendecomposition of correlation matrices and the SVD of data matrices. Relations between the different estimation methods are pointed out, and comparisons provide information on the improvement in the enhanced speech quality that can be gained with each estimator.

The effects of short data observations, i.e., the practical behavior of the estimators, have been examined through simulations, where the introduction of the signal subspace is demonstrated to be important in order to avoid flaws of the estimators.

The signal subspace based speech enhancement method has also been compared with the classical spectral subtraction technique. The two methods are closely related, but they have different advantages. In the signal subspace approach, the noise subspace can easily be identified and its influence on the estimates removed, but the method is sensitive to prewhitening. The spectral subtraction approach can handle colored noise, but the method is influenced by estimation errors in the noise subspace, and it suffers from the fact that it is a frequency domain approach and speech signals are extremely non-stationary. However, frequency domain approaches typically offer significant reductions in computational requirements, which is also the case here.

The origin of the musical noise obtained in subspace related speech enhancement systems has also been pointed out, and a proposed combination of noise reduction methods based on a single microphone with the delay-and-sum beamformer is shown to be an efficient way to reduce (eliminate) the musical noise.

In the noisy case, model based estimation is a nonlinear problem, which is normally solved by iterative techniques. However, a new idea based on multi-microphone inverse filtering is presented, where the solution is obtained by subspace methods.

A recursive approach for nonparametric speech enhancement has been developed. Traditionally, the SVD (or the eigendecomposition) is performed on a frame-by-frame basis, but here the rank-revealing ULV decomposition is used instead of the SVD, and the decomposition is updated for each new sample instead of working in frames.

Furthermore, ULV formulations of Wiener based estimation strategies have been proposed. The necessary conditions are never exactly satisfied, however, it has been demonstrated that the rank-revealing ULV decomposition is robust with respect to mild violations of these conditions. Another point is that the rank-revealing ULV algorithm is implemented in a way that closely match the conditions.

A complete rank-revealing ULLV algorithm has been developed, analyzed and implemented. The critical step is the CSNE-based downdating used in the sliding window method when the left-side orthogonal matrices are not maintained.

When the signal represents a stationary process, exponential windowing is to be preferred to downdating. It is simpler and has better numerical properties. However, for nonstationary signals like speech, the techniques produce different results, and the downdating method can track the change in the signal statistics more accurately than the exponential window method.

Finally, experiments demonstrate that the ULV-based algorithm is able to achieve the same quality of the reconstructed speech signal as the SVD-based method. In the sense of signal dependent assumptions, the descriebed algorithms can be considered very robust.

For further research, structured estimation strategies are suggested, e.g., formulated by means of the Riemannian SVD. Another promising area is the multichannel blind identification approaches.

Appendix A

Computational Count for the RRULLV Algorithm

Table A.1–A.10 gives the amount of work involved in the initialization, updating, downdating, deflation and refinement step in the rank-revealing ULLV algorithm of the matrix pair $(\mathbf{X}, \mathbf{N})$.

The parameters in the flop counts are given by

- m_X = row dimension of $\mathbf{X}$ and $\mathbf{U}_X$.
- m_N = row dimension of $\mathbf{N}$ and $\mathbf{U}_N$.
- n = column dimension.
- p = numerical rank of $\mathbf{X}\mathbf{N}^+$.
- r = dimension of the leading principle submatrix of $\mathbf{L}_X$ to be deflated and refined.

and the floating point operation is either one multiplication, division, addition, subtraction or square root.

The operation counts for all substeps other than the CSNE-based expansion assume that the left-side orthogonal matrices $\mathbf{U}_X$ and $\mathbf{U}_N$ are maintained. To obtain the operation counts for these substeps, when the left-side orthogonal matrices are not maintained, simply eliminate all terms containing m_X and m_N from the given operation counts. Thus, in this case the CSNE-based expansion is used instead of the MGSR-based method.

Step	Formula	Item	Flop count
Init.	(7.29)	$\mathbf{x}^T\mathbf{V}$	$\sum_{i=1}^{n}(2n-1) \quad = 2n^2 - n$
	(7.30)	$\beta\mathbf{L}_X$	$\sum_{i=1}^{n}(n-i+1) = \frac{1}{2}n^2 + \frac{1}{2}n$
X1-1	(7.33)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\mathbf{L}'$	$\sum_{i=1}^{n-1} 6(i+1) \quad = 3n^2 + 3n - 6$
		$\bar{\mathbf{V}}$	$\sum_{i=1}^{n-1} 6n \quad = 6n^2 - 6n$
X1-2	(7.34)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2 - 3n$
		$\mathbf{L}'_X$	$\sum_{i=1}^{n-1} 6(i+1) \quad = 3n^2 + 3n - 6$
	(7.35)	$\bar{\mathbf{U}}_N$	$\sum_{i=1}^{n-1} 6m_N \quad = 6m_N(n-1)$
X1-3	(7.36)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2 - 3n$
		$\bar{\mathbf{U}}_X$	$\sum_{i=1}^{n-1} 6m_X \quad = 6m_X(n-1)$
X2	(7.45)	η	$= 1$
X3	(7.48)	(c,s,r)	$= 7$
		$\hat{\mathbf{U}}_X$	$= m_X$
$\sum_{flop} = 6m_X(n-1) + m_X + 6m_N(n-1) + 20.5n^2 + 14.5n - 25$			

Table A.1 Flop count for updating **X**.

Step	Formula	Item	Flop count
Init.	(7.51)	$\mathbf{n}^T\mathbf{V}$	$\sum_{i=1}^{n}(2n-1) \quad = 2n^2 - n$
	(7.52)	$\beta^{-1}\mathbf{L}_X$	$\sum_{i=1}^{n}(n-i+1) = \frac{1}{2}n^2 + \frac{1}{2}n$
		$\beta\mathbf{L}$	$\sum_{i=1}^{n}(n-i+1) = \frac{1}{2}n^2 + \frac{1}{2}n$
N1-1	(7.53)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\mathbf{L}'$	$\sum_{i=1}^{n-1} 6(i+1) \quad = 3n^2 + 3n - 6$
		$\bar{\mathbf{V}}$	$\sum_{i=1}^{n-1} 6n \quad = 6n^2 - 6n$
N1-2	(7.54)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2 - 3n$
		$\mathbf{L}'_X$	$\sum_{i=1}^{n-1} 6(i+1) \quad = 3n^2 + 3n - 6$
	(7.55)	$\bar{\mathbf{U}}_N$	$\sum_{i=1}^{n-1} 6m_N \quad = 6m_N(n-1)$
N1-3	(7.56)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n - 7$
		$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2 - 3n$
		$\bar{\mathbf{U}}_X$	$\sum_{i=1}^{n-1} 6m_X \quad = 6m_X(n-1)$
N2	(7.59)	(c,s,r)	$= 7$
		$\hat{\mathbf{L}}_X$	$= n$
	(7.60)	$\hat{\mathbf{U}}_N$	$= m_N$
$\sum_{flop} = 6m_X(n-1) + m_N + 6m_N(n-1) + 21n^2 + 16n - 26$			

Table A.2 Flop count for updating **N**.

Method	Formula	Item	Flop count
MGSR$^{(1)}$	Alg. 7.4	$\mathbf{t}$	$\sum_{i=1}^{n-1} 2m_X = 2m_X(n-1)$
MGSR$^{(2)}$	Alg. 7.3	$\mathbf{t}'$	$\sum_{i=1}^{n-1}(4m_X-1) = 4m_X(n-1)-n+1$
	Alg. 7.1	$\mathbf{q}$	$= 3m_X$
$\sum_{flop}^{(1)} = 2m_X n + m_X$			
$\sum_{flop}^{(2)} = 6m_X n - 3m_X - n + 1$			

Table A.3 Flop count for MGSR expansion step in downdating.

Method	Formula	Item	Flop count
CSNE$^{(1)}$	(7.96)	$\mathbf{u}_{X1}$	$n(2n-1)+2n^2 = 4n^2-n$
	(7.97)	$\mathbf{z}$	$= 2n^2$
	(7.98)	$\mathbf{t}$	$m_X(2n-1)+n(2n-1)+m_X = 2m_X n+2n^2-n$
CSNE$^{(2)}$	(7.99)	$\delta\mathbf{u}_{X1}$	$n(2m_X-1)+n(2n-1)+2n^2 = 2m_X n+4n^2-2n$
	(7.100)	$\delta\mathbf{z}$	$= 2n^2$
	(7.101)	$\mathbf{u'}_{X1}$	$= n$
	(7.102)	$\mathbf{t}'$	$m_X(2n-1)+n(2n-1)+m_X = 2m_X n+2n^2-n$
	Alg. 7.1	$\mathbf{q}$	$= 3m_X$
$\sum_{flop}^{(1)} = 2m_X n + 8n^2 + 3m_X - 2n$			
$\sum_{flop}^{(2)} = 6m_X n + 16n^2 + 3m_X - 4n$			

Table A.4 Flop count for CSNE expansion step in downdating.

Step	Formula	Item	Flop count
DX2-1	(7.108)	(c,s,r)	$\sum_{i=1}^{n-1} 7 = 7n-7$
		$\mathbf{L'}_X$	$\sum_{i=1}^{n-1} 6(i+1) = 3n^2+3n-6$
		$\bar{\mathbf{U}}_X$	$\sum_{i=1}^{n-1} 6m_X = 6m_X(n-1)$
DX2-2	(7.109)	(c,s,r)	$\sum_{i=1}^{n-1} 7 = 7n-7$
		$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{n-1} 6i = 3n^2-3n$
		$\mathbf{L}'$	$\sum_{i=1}^{n-1} 6(i+1) = 3n^2+3n-6$
	(7.110)	$\bar{\mathbf{U}}_N$	$\sum_{i=1}^{n-1} 6m_N = 6m_N(n-1)$
DX2-3	(7.111)	(c,s,r)	$\sum_{i=1}^{n-1} 7 = 7n-7$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{n-1} 6i = 3n^2-3n$
		$\bar{\mathbf{V}}$	$\sum_{i=1}^{n-1} 6n = 6n^2-6n$
DX3	(7.115)	$\hat{\mathbf{L}}_X$	$= 2$
		$\hat{\mathbf{U}}_X$	$= 3m_X$
DX4	(7.119)	ξ	$= 1$
$\sum_{flop} = 6m_X(n-1)+3m_X+6m_N(n-1)+18n^2+15n-30$			

Table A.5 Flop count for downdating $\mathbf{X}$.

Step	Formula	Item	Flop count
DN2-1	(7.129)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n-7$
		$\mathbf{L}'$	$\sum_{i=1}^{n-1} 6(i+1) = 3n^2+3n-6$
		$\bar{\mathbf{U}}_N$	$\sum_{i=1}^{n-1} 6m_N \quad = 6m_N(n-1)$
	(7.130)	$\mathbf{L}'_X$	$\sum_{i=1}^{n-1} 6(i+1) = 3n^2+3n-6$
DN2-2	(7.131)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n-7$
		$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2-3n$
		$\bar{\mathbf{U}}_X$	$\sum_{i=1}^{n-1} 6m_X \quad = 6m_X(n-1)$
DN2-3	(7.132)	(c,s,r)	$\sum_{i=1}^{n-1} 7 \quad = 7n-7$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{n-1} 6i \quad = 3n^2-3n$
		$\bar{\mathbf{V}}$	$\sum_{i=1}^{n-1} 6n \quad = 6n^2-6n$
DN3	(7.136)	$\hat{\mathbf{L}}_X$	$= n$
		$\hat{\mathbf{L}}$	$= 1$
	(7.137)	$\hat{\mathbf{U}}_N$	$= 3m_N$
	(7.140)	ξ	$= 1$
$\sum_{flop} =$ $6m_X(n-1)+3m_N+6m_N(n-1)+18n^2+16n-31$			

Table A.6 Flop count for downdating $\mathbf{N}$.

Step	Formula	Item	Flop count
D1	Alg. 7.5	$\hat{\mathbf{u}}$	$= 6r^2+4r+1$
D2	(7.146)	(c,s,r)	$\sum_{i=1}^{r-1} 7 \quad = 7r-7$
D3	(7.148)	$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{r-1} 6(i+1) = 3r^2+3r-6$
		$\bar{\mathbf{U}}_X$	$\sum_{i=1}^{r-1} 6m_X \quad = 6m_X(r-1)$
D4	(7.149)	(c,s,r)	$\sum_{i=1}^{r-1} 7 \quad = 7r-7$
		$\hat{\mathbf{L}}_X$	$\sum_{i=1}^{r-1} 6(n-i) = 6n(r-1)-3r^2+3r$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{r-1} 6(i+1) = 3r^2+3r-6$
	(7.150)	$\hat{\mathbf{U}}_N$	$\sum_{i=1}^{r-1} 6m_N \quad = 6m_N(r-1)$
D5	(7.152)	(c,s,r)	$\sum_{i=1}^{r-1} 7 \quad = 7r-7$
		$\hat{\mathbf{L}}$	$\sum_{i=1}^{r-1} 6(n-i) = 6n(r-1)-3r^2+3r$
		$\hat{\mathbf{V}}$	$\sum_{i=1}^{r-1} 6n \quad = 6n(r-1)$
$\sum_{flop} =$ $(6m_X+6m_N+18n)(r-1)+6r^2+37r-32$			

Table A.7 Flop count for one step of deflation on $\mathbf{L}_X(1:r,1:r)$.

Step	Formula	Item	Flop count
R1	(7.167)	(c,s,r)	$\sum_{i=1}^{r-1} 7 = 7r-7$
		$\bar{\mathbf{L}}_X$	$\sum_{i=1}^{r-1} 6i = 3r^2-3r$
		$\hat{\mathbf{U}}_X$	$\sum_{i=1}^{r-1} 6m_X = 6m_X(r-1)$
R2	(7.168)	(c,s,r)	$\sum_{i=1}^{r-2} 7 = 7r-14$
		$\mathbf{L}'$	$\sum_{i=1}^{r-2} 6(n-i-1) = 6n(r-2)-3r^2+3r+6$
		$\bar{\mathbf{V}}$	$\sum_{i=1}^{r-2} 6n = 6n(r-2)$
R3	(7.169)	(c,s,r)	$\sum_{i=1}^{r-1} 7 = 7r-7$
		$\hat{\mathbf{L}}_X$	$\sum_{i=1}^{r-1} 6(n-i) = 6n(r-1)-3r^2+3r$
		$\bar{\mathbf{L}}$	$\sum_{i=1}^{r-1} 6(i+1) = 3r^2+3r-6$
	(7.170)	$\hat{\mathbf{U}}_N$	$\sum_{i=1}^{r-1} 6m_N = 6m_N(r-1)$
R4	(7.171)	(c,s,r)	$\sum_{i=1}^{r-1} 7 = 7r-7$
		$\hat{\mathbf{L}}$	$\sum_{i=1}^{r-1} 6(n-i) = 6n(r-1)-3r^2+3r$
		$\hat{\mathbf{V}}$	$\sum_{i=1}^{r-1} 6n = 6n(r-1)$
$\sum_{flop}=$ $(6m_X+6m_N+18n)(r-1)+12n(r-2)-3r^2+37r-35$			

Table A.8 Flop count for one step of refinement on $\mathbf{L}_X(1:r,1:r)$.

Algorithm	Required steps	Flop count
Deflation	$n-p$	$(m_X+m_N)(3n^2-3n)+11n^3+12.5n^2-12.5n$ $-[(m_X+m_N+3n)(3p^2-3p)+2p^3+21.5p^2-12.5p]$
	1	$(m_X+m_N+3n)6p+6p^2+49p+11$
Refinement	$n-p$	$(m_X+m_N)(3n^2-3n)+14n^3-28.5n^2-0.5n$ $-[(m_X+m_N+3n)(3p^2-3p)+5p^3-19.5p^2-0.5p]$
	1	$(m_X+m_N+5n)6p-12n-3p^2+31p-1$

Table A.9 Flop count for rank-revealing to rank p.

Step	Formula	Item	Flop count
QL as Alg. 7.2	(7.172)	$\mathbf{U}_N, \mathbf{L}$	$3m_Nn^2+2m_Nn-\frac{1}{2}n^2+\frac{1}{2}$
Solve $\mathbf{X}=\mathbf{ZL}$	(7.173)	$\mathbf{Z}$	m_Xn^2
QL as Alg. 7.2	(7.174)	$\mathbf{U}_X, \mathbf{L}_X$	$3m_Xn^2+2m_Xn-\frac{1}{2}n^2+\frac{1}{2}$
Deflation ($n-p$ steps)			see Table A.9
Refinement ($n-p$ steps)			see Table A.9
$\sum_{flop}$ $= 10m_Xn^2+9m_Nn^2-4m_Xn-4m_Nn+25n^3-17n^2-12n$ $-[(m_X+m_N+3n)(6p^2-6p)+7p^3+2p^2-13p]$			

Table A.10 Flop count for initialization.

Appendix B

The ULV/ULLV Toolbox for Matlab

The ULV/ULLV Toolbox consists of a number of Matlab routines for calculation of rank-revealing ULV/ULLV decompositions. This includes initial calculation of the decompositions and up/downdating methods. The algorithms are based on Givens rotations, i.e., for-loops, resulting in a slow execution of Matlab m-files. Therefore, a c-version is also available with a mex-file interface. Totally, the package consist of 31 m-files and 53 c-files, and the present reference documentation [52] covers the m-version. One of the inspirations for the toolbox was a set of Matlab routines concerning rank-revealing ULLV updating kindly provided to us by S. Qiao. First, the installation of the ULV/ULLV Toolbox is described, followed by an overview of the files.

B.1 Installation

The ULV/ULLV Toolbox is distributed as a compressed Unix tar-file `UllvBox.tgz`, available by anonymous ftp from Internet address `ftp.imm.dtu.dk` in directory `out/pskh`. A short introduction and the manual pages is a compressed ps-file `UllvBox.ps.gz` [52] in the same location. To install the toolbox, download the tar-file into the directory, where you want the ULV/ULLV toolbox subdirectory to be, then type

```
tar xzvf UllvBox.tgz
```

A directory `UllvBox` will be created containing a number of subdirectories with m-files and c-files arranged as follows

```
UllvBox - MatFun
          Ulv
          Ullv
          Mex - include
                libAlloc
                libNum
                Ulv
                Ullv
```

To generate the mex-files, go to the `UllvBox` directory and type `make`. This will do all the job for you. Now you can either append the relevant directories to the Matlab path or copy the files to a known location.

B.2 Quick Reference Tables, M-files

This section contains quick reference tables to the ULV/ULLV Toolbox. All the M-files in the toolbox are listed by category in the three tables with a short description. Manual pages can be obtained through the on-line help facility.

Rank-Revealing ULV Decomposition	
rrulvd	Initial computation of the rank-revealing ULV decomposition.
ulv_x	Up/downdating the rank-revealing ULV decomposition.
ulv_up	Updating the ULV decomposition.
ulv_dw	Downdating the ULV decomposition.
ulv_rr	Rank-revealing algorithm for the ULV up/downdating.
ulv_def	Deflation on $\mathbf{L}$ in the ULV decomposition.
ulv_ref	One iteration of refinement in the ULV decomposition.

Rank-Revealing ULLV Decomposition	
rrullvd	Initial computation of the rank-revealing ULLV decomposition.
ullv_a	Up/downdating the $\mathbf{A}$-part of the rank-revealing ULLV decomp.
ullv_b	Up/downdating the $\mathbf{B}$-part of the rank-revealing ULLV decomp.
ullv_upa	Updating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_upb	Updating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_dwa	Downdating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_dwb	Downdating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_rr	Rank-revealing algorithm for the ULLV up/downdating.
ullv_def	Deflation on $\mathbf{L}_A$ in the ULLV decomposition.
ullv_ref	One iteration of refinement in the ULLV decomposition.

Matrix Tools	
pythag	Pythagoras equation.
back_sub	Column version of Back Substitution.
forward_sub	Column version of Forward Substitution.
given_cs	Givens rotation matrix.
given_rt	Applying Givens rotations (left/right).
exp_mgsr	Modified Gram-Schmidt expansion.
exp_csneL	Corrected Semi-Normal Equations expansion in an ULV-decomp.
exp_csneLL	Corrected Semi-Normal Equations expansion in an ULLV-decomp.
ql_house	QL-factorization using Householder transformations.
ql_mgs	QL-factorization using Modified Gram-Schmidt.
svd_t22	SVD of 2×2 upper triangular matrix (real).
vmin_ccvl	Generalized LINPACK estimator of the smallest singular value.
vmin_co	Convex optimization based estimator of the smallest singular value.
vmin_lb	Look-behind based estimator of the smallest singular value.

B.3 Quick Reference Tables, C-files

The MEX-file version of the toolbox is almost identical to the M-file version, but of course some of the build-in functionality of MATLAB, such as the norm functions, the QR-decomposition etc., is implemented.

Also the C-files in the toolbox are listed by category in the following four tables with a short description, and the manual pages are available from the file headers.

Rank-Revealing ULV Decomposition	
rrulvd_mex	MEX gateway function to be called from MATLAB.
rrulvd	Initial computation of the rank-revealing ULV decomposition.
ulv_x_mex	MEX gateway function to be called from MATLAB.
ulv_x	Up/downdating the rank-revealing ULV decomposition.
ulv_up	Updating the ULV decomposition.
ulv_dw	Downdating the ULV decomposition.
ulv_rr	Rank-revealing algorithm for the ULV up/downdating.
ulv_def	Deflation on **L** in the ULV decomposition.
ulv_ref	One iteration of refinement in the ULV decomposition.

Rank-Revealing ULLV Decomposition	
rrullvd_mex	MEX gateway function to be called from MATLAB.
rrullvd	Initial computation of the rank-revealing ULLV decomposition.
ullv_a_mex	MEX gateway function to be called from MATLAB.
ullv_a	Up/downdating the **A**-part of the rank-revealing ULLV decomp.
ullv_b_mex	MEX gateway function to be called from MATLAB.
ullv_b	Up/downdating the **B**-part of the rank-revealing ULLV decomp.
ullv_upa	Updating the **A**-part of the ULLV decomposition.
ullv_upb	Updating the **B**-part of the ULLV decomposition.
ullv_dwa	Downdating the **A**-part of the ULLV decomposition.
ullv_dwb	Downdating the **B**-part of the ULLV decomposition.
ullv_rr	Rank-revealing algorithm for the ULLV up/downdating.
ullv_def	Deflation on $\mathbf{L}_A$ in the ULLV decomposition.
ullv_ref	One iteration of refinement in the ULLV decomposition.

Matrix Tools	
vnorm2	Vector 2-norm.
mnorm1	Matrix 1-norm.
mnormf	Matrix Frobenius-norm.
mnorminf	Matrix inf-norm.
back_sub	Column version of Back Substitution.
forward_sub	Column version of Forward Substitution.
given_cs	Givens rotation matrix.
given_lt	Applying Givens Rotations (left).
given_rt	Applying Givens Rotations (right).
house_cv	Householder column vector.
house_rv	Householder row vector.
house_lt	Householder pre-multiplication.
house_rt	Householder post-multiplication.
house_p	Product of Householder matrices (right transformations).
house_q	Product of Householder matrices (left transformations).
solveXLY	Solve triangular system with multiple right-hand sides.
exp_mgsr	Modified Gram-Schmidt expansion.
exp_csneL	Corrected Semi-Normal Equations expansion in an ULV-decomp.
exp_csneLL	Corrected Semi-Normal Equations expansion in an ULLV-decomp.
qr_house	QR-factorization using Householder transformations.
qr_mgs	QR-factorization using Modified Gram-Schmidt.
ql_house	QL-factorization using Householder transformations.
ql_mgs	QL-factorization using Modified Gram-Schmidt.
vmin_ccvl	Generalized LINPACK estimator of the smallest singular value.
vmin_co	Convex optimization based estimator of the smallest singular value.

Memory Allocation Tools	
dvector	Allocates a double vector.
dmatrix	Allocates a double matrix.
matlab2pp	Change matrix pointer format to matrix pointer-pointer format.
free_dvec	Frees memory allocated by dvector.
free_dmat	Frees memory allocated by dmatrix.
free_mat2pp	Frees memory allocated by matlab2pp.

Appendix C

The Speech Enhancement Toolbox for Matlab

The Speech Enhancement Toolbox consists of a number of Matlab routines for signal subspace based noise reduction. This includes a GUI interface, signal generation, enhancement algorithms and analysis tools. First, the installation of the Toolbox is described, followed by an overview of the files.

C.1 Installation

The Speech Enhancement Toolbox is distributed as a compressed Unix tar-file `SubBox.tgz`, available by anonymous ftp from Internet address `ftp.imm.dtu.dk` in directory `out/pskh`. A short introduction and the manual pages is a compressed ps-file `SubBox.ps.gz` in the same location. To install the toolbox, download the tar-file into the directory, where you want the subdirectory of the speech enhancement toolbox to be, then type

```
tar xzvf SubBox.tgz
```

A directory `SubBox` will be created containing a number of subdirectories with m-files arranged as follows

```
SubBox - AnaFun
         GenFun
         GuiFun
         IoFun
         SigFun
         SubFun
```

Now you can either append the relevant directories to the Matlab path or copy the files to a known location.

C.2 Quick Reference Tables, m-files

This section contains quick reference tables to the Speech Enhancement Toolbox. The m-files in the toolbox are listed by category in the five tables with a short description. However, note that a large number of analysis functions are also available, but they are not yet included in the GUI structure, and will not be listed here. Manual pages can be obtained through the on-line help facility.

GUI Interface	
GuiMain	GUI controlled signal subspace based noise reduction demo.
GuiInfo	GUI information interface.
GuiSig	GUI input interface for signal generator parameters.
GuiSigExe	Generate noisy signals with parameters obtained from GUI.
GuiAlgo	GUI input interface for algorithm parameters.
GuiAlgoExe	Run noise reduction algorithm with parameters obtained from GUI.
GuiShift	Shifts secondary window.
GuiVis	Find indices of visible uicontrols.
GuiFrm	Generate frames for pairs of uicontrols.
GuiBtn	Generates a set of uicontrols.
GuiBtnPo	Generate position parameters for a matrix of uicontrols.

Signal Generator	
SigGen	Signal generator.
Noise	ARMA process generator.
Sinus	Sinus generator.
SynTalk	Syntetic talk generator.
VocalTr	Vocal tract filter.
LoadProc	Load mat-file from disk.
LoadMult	Load several mat-files from disk.
AddNoi	Add noise to signals.

Subspace Functions	
FiltFrm	Frame-based noise reduction algorithm.
FiltRec	Recursive noise reduction algorithm.
EstSVD	SVD-based estimation.
EstQSVD	QSVD-based estimation.
EstULV	ULV-based estimation.
EstULLV	ULLV-based estimation.
Toep	Generates a Toeplitz matrix.
DiagSum	Sum elements along the diagonals of a matrix.
DiagMean	Average elements along the diagonals of a matrix.
PowerRec	Short-term power computation using a sliding window.
Window	Generates a window function.
GenWin	Generate matrix window and the corresponding time window.
AppWin	Apply window to a matrix.
OlapAdd	Window-based overlap-add synthesis.

Signal Processing Functions	
BandPass	Butterworth bandpass filter.

I/O Functions	
RawRead	Read raw data from file.
RawWrite	Write raw data to file.
WavWrite	Write raw data to wav sound file.
adc	A/D converter.
dac	D/A converter.

Appendix D

EUSIPCO-96 Paper

This appendix contains the paper [53]: P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Noise Reduction of Speech Signals using the Rank-Revealing ULLV Decomposition. In *Signal Processing VIII Theories and Applications*, volume 2, pages 967–970. EUSIPCO-96, September 1996.

NOISE REDUCTION OF SPEECH SIGNALS USING THE RANK-REVEALING ULLV DECOMPOSITION

Peter S. K. Hansen, Per Christian Hansen†*, Steffen Duus Hansen and John Aasted Sørensen*

Department of Mathematical Modelling, Section for Digital Signal Processing
Technical University of Denmark, DK-2800 Lyngby, Denmark
E-mail: pskh@imm.dtu.dk, sdh@imm.dtu.dk and jaas@imm.dtu.dk

†UNI•C, Technical University of Denmark, DK-2800 Lyngby, Denmark
E-mail: Per.Christian.Hansen@uni-c.dk

ABSTRACT

A recursive approach for nonparametric speech enhancement is developed. The underlying principle is to decompose the vector space of the noisy signal into a signal subspace and a noise subspace. Enhancement is performed by removing the noise subspace and estimating the clean signal from the remaining signal subspace. The decomposition is performed by applying the rank-revealing ULLV algorithm to the noisy signal. With this formulation, a prewhitening operation becomes an integral part of the algorithm. Linear estimation is performed using a proposed minimum variance estimator. Experiments indicate that the approximative method is able to achieve a satisfactory quality of the reconstructed speech signal comparable with eigenfilter based methods.

1 INTRODUCTION

Recently, a new approach for noise reduction of speech signals based on subspace decomposition has been proposed [1, 2, 4]. The idea is to organize the noisy speech signal in a Toeplitz structured data matrix, and to decompose the span into two mutually orthogonal components.

The noise reduction algorithm in [1] is based on the Singular Value Decomposition (SVD), which is a robust and widely used computational tool in noise suppression techniques. From the SVD of the data matrix, the Least Squares (LS) estimate of the signal-only matrix can be obtained by neglecting the smallest singular values and finally the Toeplitz structure of the estimate is restored to identify the time samples. The problem is that the method deals only with white noise and the LS estimate is sensitive to the number of retained singular values.

In [4] is a noise reduction method based on the Quotient Singular Value Decomposition (QSVD) presented, where a prewhitening is an integral part of the algorithm. Moreover, by using a Minimum Variance (MV) estimate [7] of the signal-only matrix, the algorithm is less sensitive to the choice of retained singular values [4].

Unfortunately, the SVD/QSVD is computationally expensive and resists updating. This paper uses the rank-revealing ULV/ULLV decomposition [8, 6, 5] to estimate the rank and the orthogonal subspaces in the noise reduction algorithm. A recursive ULLV algorithm for a sliding window has been developed and an approximate MV estimate is proposed.

2 SIGNAL AND NOISE MODEL

Let $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T$ denote the noisy signal vector of m samples and assume that the noise is additive and uncorrelated with the speech signal, i.e.,

$$\mathbf{x} = \mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{s}$ contains the speech component and $\mathbf{n}$ represents the noise. A set of time shifted vectors can be organized in a data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ with Toeplitz structure

$$\mathbf{X} = \begin{pmatrix} x_n & x_{n-1} & \cdots & x_1 \\ x_{n+1} & x_n & \cdots & x_2 \\ \vdots & \vdots & & \vdots \\ x_{m+n-1} & x_{m+n-2} & \cdots & x_m \end{pmatrix} = \mathbf{S} + \mathbf{N} \tag{2}$$

where $m \geq n$. Moreover, assume that the noise is broad-banded so $\mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\mathbf{N}) = n$ and that the speech signal can be described by a low order model, giving a rank deficient matrix $\mathbf{S}$ with $\mathrm{rank}(\mathbf{S}) = p < n$. It includes, for example, the *damped complex sinusoid* model, which has often been attributed to speech signals.

Thus, the speech signal is known to lie in a subspace of order p, but the subspace is unknown. The noise reduction problem is to *estimate* the subspace, i.e., its dimension and a suitable basis, and use this information in a signal processing procedure. Note, that its not possible to find the *exact* subspace.

3 LINEAR SIGNAL ESTIMATORS

One approach for *nonparametric* speech enhancement is linear estimation of the clean signal from the noisy signal using signal subspace methods, which is based on the SVD of the data matrix $\mathbf{X}$ partitioned as follows

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{3}$$

where $\mathbf{U}_1 \in \mathbb{R}^{m\times p}$, $\mathbf{V}_1 \in \mathbb{R}^{n\times p}$ and $\mathbf{\Sigma}_1 \in \mathbb{R}^{p\times p}$.

A straightforward and simple solution to the estimation problem is obtained by use of the *Least Squares* (LS) criterion, which minimizes the squared fitting errors between the noisy measurements $\mathbf{X}$ and a low rank model $\mathbf{S}_p$, i.e.,

$$\min_{\mathrm{rank}(\mathbf{S}_p)=p} \mathrm{tr}\left((\mathbf{X} - \mathbf{S}_p)^T(\mathbf{X} - \mathbf{S}_p)\right) \quad \Rightarrow \tag{4}$$

$$\hat{\mathbf{S}}_{LS} = \mathbf{S}_p = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T \tag{5}$$

The estimate $\hat{\mathbf{S}}_{LS}$ is easily obtained without any statistical knowledge about the signals.

Assume now that the estimator $\hat{\mathbf{s}} \in \mathbb{R}^m$ of the pure signal vector $\mathbf{s}$ is constrained to be a *linear function* of the measurement vector $\mathbf{x}$, i.e., $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$ where $\mathbf{W} \in \mathbb{R}^{m\times m}$ is a filter matrix, then the *Linear Minimum Mean-Squared Error* (LMMSE) estimator problem is to find the matrix $\mathbf{W}$ that minimizes

$$\min_{\mathbf{W}} \mathrm{tr}E\{(\mathbf{W}\mathbf{x}-\mathbf{s})(\mathbf{W}\mathbf{x}-\mathbf{s})^T\} \ \Rightarrow \tag{6}$$

$$\mathbf{W}_{LMMSE} = \mathbf{R}_s\mathbf{R}_x^{-1} \tag{7}$$

This theory produces the *Wiener-Hopf equations* as the fundamental design equations, i.e., we require the covariance properties of the noisy signal and the noise process.

In practice, this information is not available and is estimated from the noisy data. Under stationary and ergodic conditions, the ensemble average operator $E\{\cdot\}$ can be implemented as the mean value of several time shifted vectors, i.e., by use of the data matrix $\mathbf{X}$ and the signal-only matrix $\mathbf{S}$ (2). This gives us the *Minimum Variance* (MV) estimator

$$\min_{\mathbf{W}} \mathrm{tr}\left((\mathbf{X}\mathbf{W}-\mathbf{S})^T(\mathbf{X}\mathbf{W}-\mathbf{S})\right) \ \Rightarrow \tag{8}$$

$$\mathbf{W}_{MV} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \tag{9}$$

which converges asymptotically to the LMMSE estimator as the number of rows $m \rightarrow \infty$ [7]. Note that $\mathbf{W} \in \mathbb{R}^{n\times n}$ in this case.

Since speech signals are nonstationary, a time varying estimator must be used. Such an estimator provides nonstationary residual noise with annoying noticeable tonal characteristics referred to as *musical noise*. This can be reduced [2] by maintaining the residual noise below some threshold either global or local in each eigenfilter. An ULV/ULLV treatment of these estimators is outside the scope of this paper.

4 ULV BASED SIGNAL ESTIMATION

The ULV decomposition was first introduced by Stewart [8]. A basic feature is that the ULV decomposition of a full rank matrix $\mathbf{X}$ can be made rank-revealing, if there is a gap in the singular values, e.g., when $\mathbf{X}$ is the sum of a rank deficient signal matrix $\mathbf{S}$ and a full rank noise matrix $\mathbf{N}$.

Assume that $\mathbf{X} \in \mathbb{R}^{m\times n}$ has numerical rank $p < n \leq m$ corresponding to a given tolerance τ, then its singular values satisfy

$$\sigma_1 \geq \cdots \geq \sigma_p \geq \tau \gg \sigma_{p+1} \geq \cdots \geq \sigma_n \tag{10}$$

and there exists a matrix $\mathbf{U} \in \mathbb{R}^{m\times n}$ with orthogonal columns and an orthogonal matrix $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{V}^T = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \mathbf{L}_1 & \mathbf{0} \\ \mathbf{F} & \mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{11}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$, $\mathbf{L}_1 \in \mathbb{R}^{p\times p}$ and $\mathbf{G} \in \mathbb{R}^{(n-p)\times(n-p)}$ are lower triangular, and

$$\sigma_{min}(\mathbf{L}_1) \ \approx \ \sigma_p \tag{12}$$

$$\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2 \ \approx \ \sigma_{p+1}^2 + \cdots + \sigma_n^2 \tag{13}$$

From the RRULVD we can estimate the signal- and noise subspaces defined by the gap in the singular values. The tolerance τ is defined based on a detection threshold in the underlying signal processing problem.

4.1 LS Estimate by RRULVD

An approximate least squares estimate $\hat{\mathbf{S}}_{ALS}$ of the signal matrix $\mathbf{S}$ can be computed by essentially substituting the ULV decomposition for the SVD based estimate [3], thus replacing one problem with a similar, nearby problem that can be solved more efficiently.

Based on (5) and (11), a useful rank-p matrix approximation to $\mathbf{X}$ is given by

$$\hat{\mathbf{S}}_{ALS} = \mathbf{U}_1\mathbf{L}_1\mathbf{V}_1^T = \mathbf{X}\mathbf{V}_1\mathbf{V}_1^T \tag{14}$$

where $\mathbf{U}_1$ and $\mathbf{V}_1$ approximate the numerical column space and row space as defined via the SVD of $\mathbf{X}$.

4.2 MV Estimate by RRULVD

The minimum variance estimate $\hat{\mathbf{S}}_{MV}$ of the signal matrix $\mathbf{S}$ can be obtained along the lines in [7] using an *idealized* rank-revealing ULV decomposition of $\mathbf{X} \in \mathbb{R}^{m\times n}$. With reference to (11), the necessary conditions are

1. The signal is orthogonal to the noise in the sense: $\mathbf{S}^T\mathbf{N} = \mathbf{0}$.
2. The matrix $\mathbf{N} = \sigma_{noise}\mathbf{Q}$, where $\mathbf{Q}$ has orthonormal columns: $\mathbf{N}^T\mathbf{N} = \sigma_{noise}^2\mathbf{I}_n$.
3. There is a distinct gap in the singular values of the matrix $\mathbf{X}$: $\sigma_p > \sigma_{p+1}$.
4. The off-diagonal matrix $\mathbf{F}$ is zero.
5. $\mathbf{G}$ is a diagonal matrix containing the noise-only singular values σ_{noise}.

Thus, we have

$$\mathbf{X} = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \tag{15}$$

Let the ULV decomposition of the matrix $\mathbf{S}$ be defined by

$$\mathbf{S} = \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \tag{16}$$

where $\mathbf{L}_{S1} \in \mathbb{R}^{p\times p}$, then we can write the idealized rank-revealing ULV decomposition of $\mathbf{X}$ in terms of the ULV decomposition of $\mathbf{S}$

$$\begin{aligned} \mathbf{X} &= \mathbf{U}_{S1}\mathbf{L}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S1}\mathbf{V}_{S1}^T + \mathbf{N}\mathbf{V}_{S2}\mathbf{V}_{S2}^T \\ &= \begin{pmatrix} (\mathbf{U}_{S1}\mathbf{L}_{S1} + \mathbf{N}\mathbf{V}_{S1})\mathbf{L}_{X1}^{-1} & \mathbf{N}\mathbf{V}_{S2}\sigma_{noise}^{-1} \end{pmatrix} \\ &\quad \times \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \end{aligned} \tag{17}$$

The matrix $\mathbf{L}_{X1}^T\mathbf{L}_{X1}$ can be obtained by comparing the matrix $\mathbf{X}^T\mathbf{X}$ using the definitions of $\mathbf{S}$ and $\mathbf{N}$ with the one based on the ULV decomposition of $\mathbf{X}$ (15), which gives

$$\mathbf{L}_{X1}^T\mathbf{L}_{X1} = \mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p \tag{18}$$

Using (16) and (17) in the MV definition (9) yields the desired MV estimate of $\mathbf{S}$

$$\begin{aligned} \hat{\mathbf{S}}_{MV} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \\ &= \mathbf{U}_X\mathbf{U}_X^T\mathbf{S} \\ &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1}^{-T}(\mathbf{L}_{S1}^T\mathbf{U}_{S1}^T + \mathbf{V}_{S1}^T\mathbf{N}^T) \\ \sigma_{noise}^{-1}\mathbf{V}_{S2}^T\mathbf{N}^T \end{pmatrix} \\ &\quad \times \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \\ &= \mathbf{U}_{X1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \end{aligned} \tag{19}$$

where (18) has been used. This equation can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}}_{MV} = \mathbf{X}\mathbf{V}_{X1}\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2 \mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \qquad (20)$$

The quantity σ_{noise}^2 can be obtained from (13)

$$\sigma_{noise}^2 = \frac{1}{n-p}\sum_{i=p+1}^{n} \sigma_i^2 \approx \frac{1}{n-p}\left(\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2\right) \qquad (21)$$

In practice, the above mentioned conditions are never satisfied exactly, but the rank-revealing ULV decomposition is robust with respect to mild violations of these conditions.

5 ULLV BASED SIGNAL ESTIMATION

If the additive noise $\mathbf{N}$ is colored, $\mathbf{N}^T\mathbf{N} \neq \sigma_n^2 \mathbf{I}_n$, then a prewhitening transformation can be applied to the data matrix using the QR decomposition of $\mathbf{N} = \mathbf{QR}$

$$\mathbf{X}\mathbf{R}^{-1} = \mathbf{S}\mathbf{R}^{-1} + \mathbf{N}\mathbf{R}^{-1} = \mathbf{S}\mathbf{R}^{-1} + \mathbf{Q} \qquad (22)$$

This transformation does not change the nature of the low order model of the speech signal while it diagonalizes the covariance matrix of the noise. In this application the noise matrix $\mathbf{N}$ can be estimated in periods without speech.

One problem concerning the prewhitening transformation is the complicated update of the matrix $\mathbf{X}\mathbf{R}^{-1}$ when $\mathbf{X}$ and $\mathbf{N}$ are updated, e.g., in a recursive application. This can be avoided by using the ULLV decomposition of the matrix pair $(\mathbf{X}, \mathbf{N})$, which allows each matrix to be updated individually and delivers the required factorizations without forming the quotients and products.

The definition given here for the rank-revealing ULLV decomposition (RRULLVD) of two matrices $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times n}$ is the one used by Luk and Qiao [6].

Assume that $\mathbf{X}\mathbf{N}^+$ ($\mathbf{N}^+$ is the pseudoinverse of $\mathbf{N}$) has numerical rank $p < n \leq m$ corresponding to a given tolerance τ, then its quotient singular values satisfy

$$\delta_1 \geq \cdots \geq \delta_p \geq \tau \gg \delta_{p+1} \geq \cdots \geq \delta_n \qquad (23)$$

and there exist matrices $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ and $\mathbf{U}_N \in \mathbb{R}^{m\times n}$ with orthogonal columns and a orthogonal matrix $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F} & \mathbf{G} \end{pmatrix} \mathbf{L} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} && (24) \\ \mathbf{N} &= \mathbf{U}_N \mathbf{L} \mathbf{V}^T && (25) \end{aligned}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$, $\mathbf{L}_{X1} \in \mathbb{R}^{p\times p}$ and $\mathbf{G} \in \mathbb{R}^{(n-p)\times(n-p)}$ are lower triangular, and

$$\begin{aligned} \sigma_{min}(\mathbf{L}_{X1}) &\approx \delta_p && (26) \\ \|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2 &\approx \delta_{p+1}^2 + \cdots + \delta_n^2 && (27) \end{aligned}$$

Thus, the ULLV decomposition reveals the rank of the matrix $\mathbf{X}\mathbf{N}^+$ assuming $\mathbf{N}$ has full rank

$$\mathbf{X}\mathbf{N}^+ = \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F} & \mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{N1}^T \\ \mathbf{U}_{N2}^T \end{pmatrix} \qquad (28)$$

Hence, working with the RRULLVD of $(\mathbf{X}, \mathbf{N})$ and the matrix $\mathbf{Q}$ is mathematically equivalent to working with the RRULVD of $\mathbf{X}\mathbf{R}^{-1}$.

5.1 LS Estimate by RRULLVD

An approximate LS estimate $\hat{\mathbf{S}}_{ALS}$ of the low-rank signal matrix $\mathbf{S}$ added colored noise can easily be obtained by first substituting the ULV decomposition of $\mathbf{X}\mathbf{N}^+$ for the SVD based estimate

$$(\mathbf{X}\mathbf{N}^+)_{ALS} = \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{U}_{N1}^T \qquad (29)$$

and then perform a denormalization of $(\mathbf{X}\mathbf{N}^+)_{ALS}$

$$\hat{\mathbf{S}}_{ALS} = (\mathbf{X}\mathbf{N}^+)_{ALS}\mathbf{N} = \mathbf{U}_{X1}\mathbf{L}_{X1}\mathbf{L}_1\mathbf{V}_1^T \qquad (30)$$

which can be computed directly from the ULLV decomposition, i.e., the prewhitening is now an integral part of the algorithm. As before, equation (30) can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}}_{ALS} = \mathbf{X}\mathbf{V}_1\mathbf{V}_1^T \qquad (31)$$

5.2 MV Estimate by RRULLVD

The approximate minimum variance estimate $\hat{\mathbf{S}}_{AMV}$ of the low-rank signal matrix in the colored noise case follows from the least squares analysis.

Using (19) with $\sigma_{noise}^2 = 1$, the approximate MV estimate of the normalized data matrix $\mathbf{X}\mathbf{N}^+$ defined by (28) is

$$(\mathbf{X}\mathbf{N}^+)_{AMV} = \mathbf{U}_{X1}(\mathbf{L}_{X1} - \mathbf{L}_{X1}^{-T})\mathbf{U}_{N1}^T \qquad (32)$$

To obtain the corresponding approximate minimum variance estimate of $\mathbf{S}$, we must denormalize $(\mathbf{X}\mathbf{N}^+)_{AMV}$

$$\hat{\mathbf{S}}_{AMV} = (\mathbf{X}\mathbf{N}^+)_{AMV}\mathbf{N} = \mathbf{U}_{X1}(\mathbf{L}_{X1} - \mathbf{L}_{X1}^{-T})\mathbf{L}_1\mathbf{V}_1^T \qquad (33)$$

where we have used (25). Again, this equation can be reformulated to avoid an explicit computation of $\mathbf{U}_X$

$$\hat{\mathbf{S}}_{AMV} = \mathbf{X}\mathbf{V}_1\mathbf{L}_1^{-1}\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \mathbf{L}_{X1}^{-T})\mathbf{L}_1\mathbf{V}_1^T \qquad (34)$$

6 EXPERIMENTS

A recursive RRULLV algorithm has been developed based on the methods given in [8, 6, 5]. Starting with initial matrices, the decomposition is updated as $\mathbf{X}$ and $\mathbf{N}$ are taken into account one row at a time. A new row is processed in the following four steps. Updating: The current row of $\mathbf{X}$ or $\mathbf{N}$ is incorporated into the decomposition. Downdating: The oldest row of $\mathbf{X}$ or $\mathbf{N}$ is isolated and removed in the decomposition. Deflation: Establishes and maintains the rank-revealing nature of the decomposition. Refinement: The norm of $\mathbf{F}$ is reduced to improve the subspace quality. By using an exponential window, the downdating step can be omitted, but clearly, the sliding window method can track the change in the signal statistics more accurately when there is an abrupt change in data.

The recursive RRULLV algorithm was applied to speech signals contaminated by an AR(1,-0.7) noise process and the noise matrix $\mathbf{N}$ was only updated in periods without speech. All the signals were sampled at 8 kHz and the matrix dimension was $m = 141$ and $n = 20$.

The typical average SNR of a reconstructed speech segment (voiced) using 100 noise realizations and SNR = 5 dB is illustrated in Fig. 1 as a function of the signal subspace dimension p. Clearly, the MV estimate is less sensitive to the choice of p compared with the LS estimate. Thus, using a fixed value of $p = 12$ as in the following results, we are able to achieve a satisfactory quality of the reconstructed speech. The behavior of the reconstructed segment in the frequency domain was also analyzed using a tenth order LPC model spectra of noise-free, noisy and reconstructed (MV estimate) speech segments, respectively. As shown in Fig. 2, the MV estimate improves the spectrum in the regions near the dominant formants. These results closely match the QSVD based method [4].

The RRULLV algorithm using a sliding window was applied to the speech signal in Fig. 3 added broad-band noise (global SNR of 5 dB). Observe from Fig. 4 that the global SNR improvement using the MV estimate is about twice the LS based improvement due to the fixed p. Moreover, the variations among the local SNRs of the various segments are reduced.

In the RRULLV algorithm computations can be saved by using the exponential window, but as demonstrated in Fig. 5, the sliding window method gives up to 6 dB better SNR, when there is a change in the dynamics of the signal. The same is true by comparing the SNRs obtained from the RRULLV sliding window method with the QSVD segment based approach also illustrated in Fig. 5.

7 SUMMARY

A recursive signal subspace approach for noise reduction of speech signals is presented. The algorithm is formulated by means of the RRULLVD using a proposed MV estimator. The method was demonstrated to be comparable with eigenfilter based methods. Integration of the RRULLVD with perceptually more meaningful estimation criterias is a topic of current research.

References

[1] M. Dendrinos, S. Bakamidis, and G. Carayannis. Speech Enhancement from Noise: A Regenerative Approach. *Speech Communication*, 10(1):45–57, February 1991.

[2] Yariv Ephraim and Harry L. Van Trees. A Signal Subspace Approach for Speech Enhancement. *IEEE Trans. on Speech and Audio Processing*, 3(4):251–266, July 1995.

[3] Ricardo D. Fierro and Per Christian Hansen. Accuracy of TSVD Solutions Computed from Rank-Revealing Decompositions. *Numerische Mathematik*, 70:453–471, 1995.

[4] S. H. Jensen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Reduction of Broad-Band Noise in Speech by Truncated QSVD. *IEEE Trans. on Speech and Audio Processing*, 3(6):439–448, November 1995.

[5] J. M. Lebak and A. W. Bojanczyk. Modifying a Rank-Revealing ULLV Decomposition. Technical report, Cornell Theory Center, June 1994.

[6] F. T. Luk and S. Qiao. A New Matrix Decomposition for Signal Processing. In M. S. Moonen et al., editor, *Linear Algebra for Large Scale and Real-Time Applications*, pages 241–247. Kluwer Academic Publishers, 1993.

[7] Bart De Moor. The Singular Value Decomposition and Long and Short Spaces of Noisy Matrices. *IEEE Trans. on Signal Processing*, 41(9):2826–2838, September 1993.

[8] G. W. Stewart. Updating a Rank-Revealing ULV Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, April 1993.

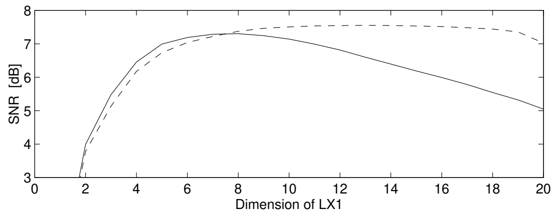


Figure 1 Average SNR of a reconstructed voiced speech segment, SNR=5dB, LS estimate (solid), MV estimate (dashed).

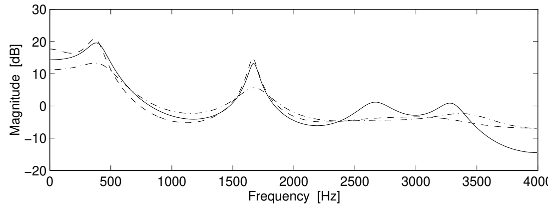


Figure 2 LPC model spectra of noise-free speech segment (solid), noisy speech segment, SNR=5dB (dash-dot) and MV estimate (dashed).

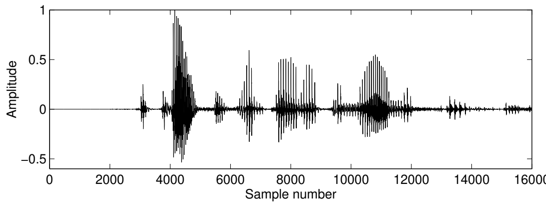


Figure 3 Noise-free speech signal.

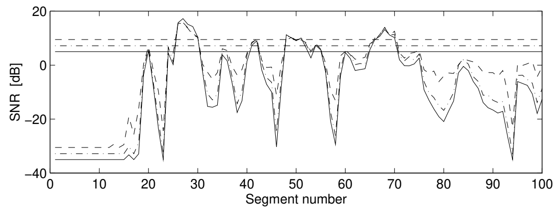


Figure 4 Local/global SNR of noisy speech signal (solid), LS estimate (dash-dot) and MV estimate (dashed).

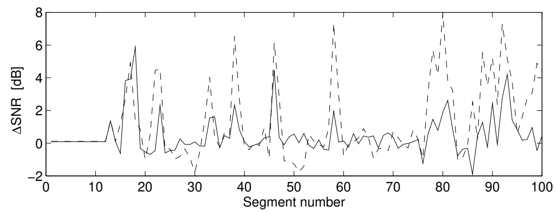


Figure 5 Difference in SNR between sliding and exponential window based MV estimate (solid) and between RRULLVD sliding window MV estimate and QSVD segment based MV estimate (dashed).

Appendix E

IWAENC-97 Paper

This appendix contains the paper [54]: P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. ULV-Based Signal Subspace Methods for Speech Enhancement. In *The International Workshop on Acoustic Echo and Noise Control, IWAENC'97*, pages 9–12, Imperial College London, September 1997.

ULV-BASED SIGNAL SUBSPACE METHODS FOR SPEECH ENHANCEMENT

Peter S. K. Hansen, Per Christian Hansen, Steffen Duus Hansen and John Aasted Sørensen

Department of Mathematical Modelling, Section for Digital Signal Processing
Technical University of Denmark, Building 321, DK-2800 Lyngby, Denmark
E-mail: pskh@imm.dtu.dk, pch@imm.dtu.dk, sdh@imm.dtu.dk and jaas@imm.dtu.dk

ABSTRACT

In this paper the signal subspace approach for nonparametric speech enhancement is considered. Traditionally, the SVD (or the eigendecomposition) is used in frame-based methods to decompose the vector space of the noisy signal into a signal- and noise subspace [1, 2, 5]. Linear estimation of the clean signal from the information in the signal subspace is then performed using a set of nonparametric estimation criteria. In this paper, the rank-revealing ULV decomposition is used instead of the SVD, and we use recursive updating of the estimate instead of working in frames. An ULV formulation of three different estimation strategies is considered: Least Squares, Minimum Variance and Time Domain Constrained. Experiments indicate that the ULV-based algorithm is able to achieve the same quality of the reconstructed speech signal as the SVD-based method.

1 SIGNAL AND NOISE MODEL

Let $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T$ denote the noisy signal vector of m samples and assume that the noise component $\mathbf{n}$ is additive and uncorrelated with the speech signal $\mathbf{s}$, i.e., $\mathbf{x} = \mathbf{s} + \mathbf{n}$.

A set of time shifted vectors can be organized in a data matrix $\mathbf{X} = \mathbf{S} + \mathbf{N} \in \mathbb{R}^{m\times n}$ with Toeplitz structure where $m \geq n$. We assume that the noise is broad-banded so $\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{N}) = n$, and that the speech signal can be described by a low order model, giving a rank deficient matrix $\mathbf{S}$ with $\text{rank}(\mathbf{S}) = p < n$. This formulation includes, for example, the *damped complex sinusoid* model, which has often been attributed to speech signals.

2 ULV BASED SIGNAL ESTIMATION

One approach for *nonparametric* speech enhancement is linear estimation of the clean signal from the noisy signal using signal subspace methods, which are based on the rank-revealing ULV decomposition (RRULVD) introduced by Stewart [8].

Assume that the singular values of $\mathbf{X}$ satisfy

$$\sigma_1 \geq \cdots \geq \sigma_p \geq \tau \gg \sigma_{p+1} \geq \cdots \geq \sigma_n \tag{1}$$

then there exists a matrix $\mathbf{U}_X \in \mathbb{R}^{m\times n}$ with orthogonal columns and an orthogonal matrix $\mathbf{V}_X \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} \mathbf{X} &= \mathbf{U}_X \mathbf{L}_X \mathbf{V}_X^T \qquad (2)\\ &= \begin{pmatrix} \mathbf{U}_{X1} & \mathbf{U}_{X2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{F}_X & \mathbf{G}_X \end{pmatrix} \begin{pmatrix} \mathbf{V}_{X1}^T \\ \mathbf{V}_{X2}^T \end{pmatrix} \end{aligned}$$

where $\mathbf{L}_{X1} \in \mathbb{R}^{p\times p}$, $\mathbf{G}_X \in \mathbb{R}^{(n-p)\times(n-p)}$ and $\mathbf{L}_X \in \mathbb{R}^{n\times n}$ are lower triangular, and

$$\sigma_{min}(\mathbf{L}_{X1}) \approx \sigma_p \tag{3}$$

$$\|\mathbf{F}_X\|_F^2 + \|\mathbf{G}_X\|_F^2 \approx \sigma_{p+1}^2 + \cdots + \sigma_n^2 \tag{4}$$

Thus, the signal- and noise subspaces defined by the gap in the singular values can be estimated using the RRULVD, where the quality depends on $\|\mathbf{F}_X\|_2$.

An *approximate* LS estimate $\hat{\mathbf{S}}_{ALS}$ of the signal matrix $\mathbf{S}$ can be computed by essentially substituting the RRULVD for the SVD based estimate [3], thus replacing one problem with a similar, nearby problem that can be solved more efficiently, i.e.,

$$\hat{\mathbf{S}}_{ALS} = \mathbf{X}\mathbf{V}_{X1}\mathbf{V}_{X1}^T \tag{5}$$

The estimate converges to the true LS solution, if the following condition is satisfied

- The off-diagonal matrix $\mathbf{F}_X$ is zero.

Assume now that the estimator $\hat{\mathbf{S}}$ of the pure signal matrix $\mathbf{S}$ is constrained to be a *linear function* of the data matrix $\mathbf{X}$, i.e., $\hat{\mathbf{S}} = \mathbf{X}\mathbf{W}$ where $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a filter matrix, then the *Minimum Variance* (MV) estimator problem [7] is to find the matrix $\mathbf{W}$ that minimizes

$$\min_{\mathbf{W}} \text{tr}\left((\mathbf{X}\mathbf{W} - \mathbf{S})^T(\mathbf{X}\mathbf{W} - \mathbf{S})\right) \Rightarrow \tag{6}$$

$$\mathbf{W}_{MV} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S} \tag{7}$$

Note, that under stationary and ergodic conditions, the MV estimator converges asymptotically to the *Linear Minimum Mean-Squared Error* (LMMSE) estimator as the number of rows $m \to \infty$ [7].

To obtain the RRULVD based MV estimate proposed in [4], i.e.,

$$\hat{\mathbf{S}}_{MV} = \mathbf{X}\mathbf{V}_{X1}\mathbf{L}_{X1}^{-1}(\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \quad (8)$$

we need the additional conditions

- The signal is orthogonal to the noise: $\mathbf{S}^T\mathbf{N} = \mathbf{0}$.
- The matrix $\mathbf{N}$ satisfy: $\mathbf{N}^T\mathbf{N} = \sigma_{noise}^2\mathbf{I}_n$.
- There is a distinct gap in the singular values of the matrix $\mathbf{X}$: $\sigma_p > \sigma_{p+1}$.
- $\mathbf{G}_X = \sigma_{noise}\mathbf{I}_{n-p}$ is a diagonal matrix.

The residual matrix $\mathbf{R} = \mathbf{S}(\mathbf{W} - \mathbf{I}_n) + \mathbf{N}\mathbf{W} = \mathbf{R}_S + \mathbf{R}_N$ minimized in the above method represents signal distortion $\mathbf{R}_S$ and residual noise $\mathbf{R}_N$. Since both terms can not be simultaneously minimized, a *Time Domain Constrained* (TDC) estimator is proposed in [2] which keep the residual noise energy $\epsilon_n^2 = \mathrm{tr}(\mathbf{R}_N^T\mathbf{R}_N)$ below some threshold while minimizing the signal distortion energy $\epsilon_s^2 = \mathrm{tr}(\mathbf{R}_S^T\mathbf{R}_S)$

$$\min_{\mathbf{W}} \epsilon_s^2 \quad \text{subject to} \quad \epsilon_n^2 \le \alpha n \sigma_{noise}^2 \quad \Rightarrow \quad (9)$$

$$\mathbf{W}_{TDC} = (\mathbf{S}^T\mathbf{S} + \gamma\sigma_{noise}^2\mathbf{I}_n)^{-1}\mathbf{S}^T\mathbf{S} \quad (10)$$

where α is a fixed or SNR-dependent parameter ($0 \le \alpha \le 1$), and γ is the Lagrange multiplier in (9). In a practical implementation, γ is actually used as the parameter.

Given the above conditions, we propose a RRULVD based TDC estimate, which can be obtained by using the following RRULVD formulations for $\mathbf{S}$ and $\mathbf{X}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{U}_{S1} & \mathbf{U}_{S2} \end{pmatrix} \begin{pmatrix} \mathbf{L}_{S1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \quad (11)$$

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} (\mathbf{U}_{S1}\mathbf{L}_{S1} + \mathbf{N}\mathbf{V}_{S1})\mathbf{L}_{X1}^{-1} & \mathbf{N}\mathbf{V}_{S2}\sigma_{noise}^{-1} \end{pmatrix} \\ &\quad \cdot \begin{pmatrix} \mathbf{L}_{X1} & \mathbf{0} \\ \mathbf{0} & \sigma_{noise}\mathbf{I}_{n-p} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{S1}^T \\ \mathbf{V}_{S2}^T \end{pmatrix} \end{aligned} \quad (12)$$

and the relation $\mathbf{L}_{X1}^T\mathbf{L}_{X1} = \mathbf{L}_{S1}^T\mathbf{L}_{S1} + \sigma_{noise}^2\mathbf{I}_p$, i.e.,

$$\begin{aligned} \hat{\mathbf{S}}_{TDC} &= \mathbf{X}\mathbf{V}_{X1}(\mathbf{L}_{X1} - \sigma_{noise}^2(1-\gamma)\mathbf{L}_{X1}^{-T})^{-1} \\ &\quad \cdot (\mathbf{L}_{X1} - \sigma_{noise}^2\mathbf{L}_{X1}^{-T})\mathbf{V}_{X1}^T \end{aligned} \quad (13)$$

Note that for $\gamma = 0$ we obtain (5) and for $\gamma = 1$ we obtain (8). For speech signals, the TDC estimation criterion will control the nonstationary residual noise with annoying noticeable tonal characteristics, referred to as *musical noise*, since this noise component decreases as $\gamma \to \infty$.

In practice, the above mentioned conditions are never satisfied exactly, but the RRULVD is robust with respect to mild violations of these conditions.

If the additive noise matrix $\mathbf{N}$ is colored, $\mathbf{N}^T\mathbf{N} \ne \sigma_{noise}^2\mathbf{I}_n$, then a prewhitening transformation can be applied to the data matrix using the QR decomposition of $\mathbf{N} = \mathbf{Q}\mathbf{R}$

$$\mathbf{X}\mathbf{R}^{-1} = \mathbf{S}\mathbf{R}^{-1} + \mathbf{N}\mathbf{R}^{-1} = \mathbf{S}\mathbf{R}^{-1} + \mathbf{Q} \quad (14)$$

One problem concerning the prewhitening transformation is the complicated update of the matrix $\mathbf{X}\mathbf{R}^{-1}$ when $\mathbf{X}$ and $\mathbf{N}$ are updated. This can be avoided by using the rank-revealing ULLV decomposition [6] of the matrix pair $(\mathbf{X}, \mathbf{N})$, which allows each matrix to be updated individually and delivers the required factorizations without forming the quotients and products.

3 IMPLEMENTATION

The transformation $\mathbf{y} = \mathbf{V}_X^T\mathbf{x}$ approximates the *Karhunen-Loeve* transform (KLT) of $\mathbf{x}$. Hence, all the above mentioned linear signal estimates are obtained by the following steps (see Figure 1)

- KLT of the noisy signal onto the signal subspace.
- Modify the components of the KLT by a gain filter matrix $\mathbf{G}_1$.
- Inverse KLT of the modified components to reconstruct the signal in the signal subspace.

This scheme results in a generalized formulation of the optimal linear estimator

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} = \mathbf{V}_{X1}\mathbf{G}_1\mathbf{V}_{X1}^T\mathbf{x} \quad (15)$$

where the matrix $\mathbf{G}_1 \in \mathbb{R}^{p \times p}$ depends on the estimation method as shown in the last section.

The two matrices $\mathbf{L}_X$ and $\mathbf{V}_X$ necessary for computing $\mathbf{W}$ are updated for each new sample x_k corresponding to a new row in the data matrix $\mathbf{X}$. A new row is processed in the following four steps.

- Updating: The new row of $\mathbf{X}$ is incorporated into the decomposition.
- Downdating: The oldest row of $\mathbf{X}$ is isolated and removed in the decomposition.
- Deflation: Establishes and maintains the rank-revealing nature of the decomposition.
- Refinement: The norm of $\mathbf{F}_X$ is reduced to improve the subspace quality.

Obviously, the filter matrix $\mathbf{W}$ is estimated in an analysis window of width $(m+n-1)$, centered around the middle row of $\mathbf{X}$. The linear estimator is applied to this row, giving a n-sample synthesis window. Finally, the enhanced vectors are combined using the overlap and add synthesis approach, which corresponds to the LS esimate of the noise-free signal s_k from the enhanced vectors [9].

4 EXPERIMENTS

A recursive RRULLV algorithm has been developed based on the methods given in [8, 6], and was applied to speech signals contaminated by an AR(1, -0.7) noise process. The noise matrix $\mathbf{N}$ was only updated in periods without speech, and the matrix dimension was $m = 141$ and $n = 20$. In all simulations, a TDC estimator is used.

The typical average SNR of a reconstructed speech segment (voiced) using 100 independent noise realizations and SNR = 5 dB is illustrated in Fig. 2 as a function of the signal subspace dimension p and the parameter γ. Clearly, the MV estimate ($\gamma = 1$) gives the best SNR improvement and is less sensitive to the choice of p compared with the other estimates. However, if γ is chosen in the neighbourhood of 1, the variations are minimal. Thus, using a fixed value of $p = 14$ as in the following results, we are able to achieve a satisfactory quality of the reconstructed speech. An informal listening test gave $\gamma \approx 2$ as the best fixed value, but a better choice is to make γ dependent on the local SNR.

The RRULLV algorithm using a sliding window was applied to the speech signal in Fig. 3 with added broad-band noise (global SNR of 5 dB). Observe from Fig. 4 that their is a SNR improvement using the TDC estimate and that the variations among the local SNRs of the various segments are reduced.

In the RRULLV algorithm computations can be saved by omitting the refinement step, i.e., accepting a larger $\|\mathbf{F}_X\|_2$, but then the singular values of $\mathbf{L}_{X1}$ will underestimate the first p values $\sigma_i(\mathbf{X}\mathbf{R}^{-1})$ by a larger factor. Similarily, the singular values of $\mathbf{G}_X$ will in general overestimate the corresponding last $n - p$ values $\sigma_i(\mathbf{X}\mathbf{R}^{-1})$.

The graphs in Fig. 5 and 6 illustrates this problem. Here, the average singular values of a prewhitened voiced speech frame are compared with the one obtained from $\mathbf{L}_{X1}$ and $\mathbf{G}_X$ with $p = 14$ and no refinement. Note, that $\sigma_i(\mathbf{L}_{X1})$ are plotted against the first p indices, and $\sigma_i(\mathbf{G}_X)$ are plotted from index $p+1$ to n. It is seen that the largest and smallest singular values and thereby the dominant range and null space are well determined by the RRULLVD, while the subspaces are blurred together near the rank-revealing point p due to the off-diagonal block $\mathbf{F}_X$ and the small gap in the singular spectrum. As shown in Fig. 6, the quality can be increased by applying a number of refinement steps.

In Fig. 7, the canonical angles between the QSVD and RRULLVD based signal subspaces are plotted against their indices, where the example corresponds to the one in Fig. 5. As expected, there is a group of large angles due to the mix of signal and noise subspace. However, since the singular spectrum of speech signals is relative constant at the rank-revealing point, this has no practical effect in a noise reduction algorithm as shown in Fig. 8. Here, four different speech segments all result in a reconstructed average SNR, which is nearly independent of the number of refinement steps. This is also why these results closely match the QSVD based method.

Another issue is that the conditions for the RRULLV based estimates are typically not satisfied. However, as demonstrated in Fig. 8, the method is very robust concerning this.

5 SUMMARY

A recursive signal subspace approach for noise reduction of speech signals is presented. The algorithm is formulated by means of the RRULVD using a proposed set of estimators. The method is demonstrated to be comparable with SVD-based methods.

References

[1] M. Dendrinos, S. Bakamidis, and G. Carayannis. Speech Enhancement from Noise: A Regenerative Approach. *Speech Communication*, 10(1):45–57, February 1991.

[2] Yariv Ephraim and Harry L. Van Trees. A Signal Subspace Approach for Speech Enhancement. *IEEE Trans. on Speech and Audio Processing*, 3(4):251–266, July 1995.

[3] Ricardo D. Fierro and Per Christian Hansen. Accuracy of TSVD Solutions Computed from Rank-Revealing Decompositions. *Numerische Mathematik*, 70:453–471, 1995.

[4] P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Noise Reduction of Speech Signals using the Rank-Revealing ULLV Decomposition. In *Signal Processing VIII Theories and Applications*, volume 2, pages 967–970. EUSIPCO-96, 1996.

[5] S. H. Jensen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Reduction of Broad-Band Noise in Speech by Truncated QSVD. *IEEE Trans. on Speech and Audio Processing*, 3(6):439–448, November 1995.

[6] F. T. Luk and S. Qiao. A New Matrix Decomposition for Signal Processing. In M. S. Moonen et al., editor, *Linear Algebra for Large Scale and Real-Time Applications*, pages 241–247. Kluwer Academic Publishers, 1993.

[7] Bart De Moor. The Singular Value Decomposition and Long and Short Spaces of Noisy Matrices. *IEEE Trans. on Signal Processing*, 41(9):2826–2838, September 1993.

[8] G. W. Stewart. Updating a Rank-Revealing ULV Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, April 1993.

[9] Donald W. Tufts and Abhijit A. Shah. Estimation of a Signal Waveform from Noisy Data Using Low-Rank Approximation to a Data Matrix. *IEEE Trans. on Signal Processing*, 41(4):1716–1721, April 1993.

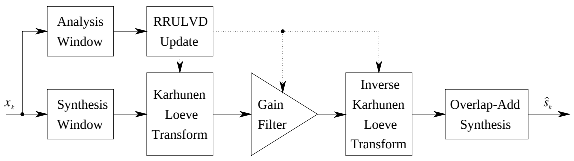


Figure 1 Filter structure.

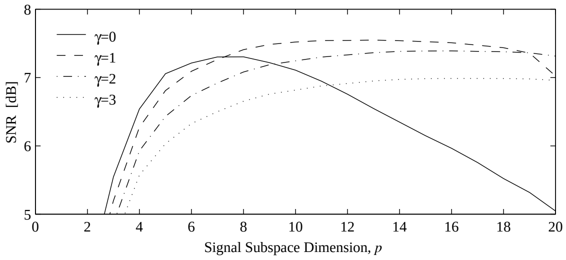


Figure 2 Average SNR of a reconstructed noisy (voiced) speech segment using a TDC estimator with the listed γ values, and SNR=5dB. The average is taken over 100 independent noise realizations.

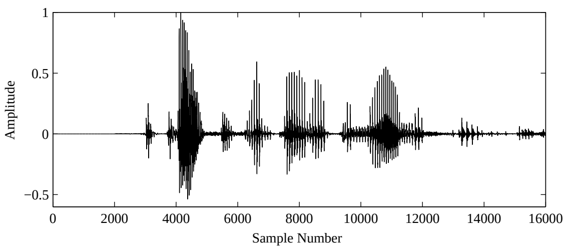


Figure 3 Noise-free speech signal.

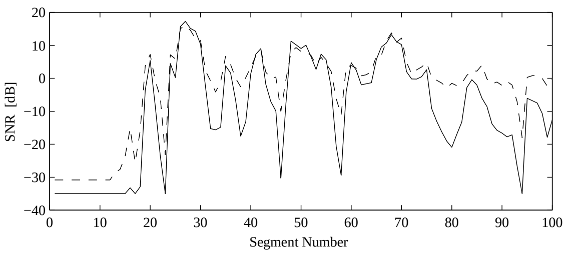


Figure 4 The local SNR of noisy speech signal (global SNR=5dB) and a TDC estimate with p=14 and γ=2.

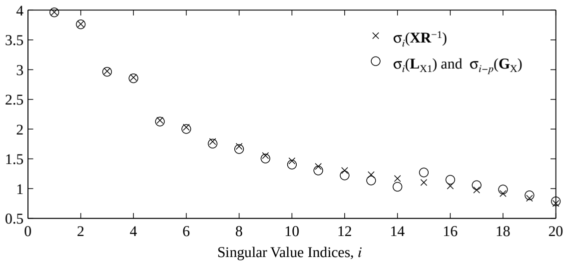


Figure 5 Average singular values of prewhitened (voiced) speech segment using 100 independent noise realizations and SNR=5dB. The rank revealed in $\mathbf{L}_X$ is $p = 14$ (without refinement).

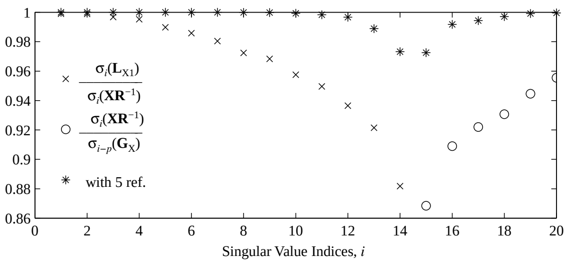


Figure 6 The ratios corresponds to the example in Fig. 5 without refinement, and a case with 5 refinement steps ($*$).

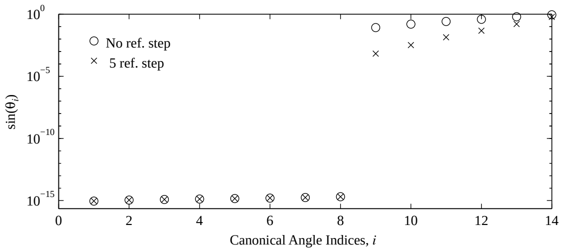


Figure 7 The average canonical angles $\sin(\theta_i)$ between the 14 dimensional signal subspaces obtained from the QSVD and the RRULLVD, respectively. The signals correspond to the example in Fig. 6 without refinement ($\circ$) and with 5 steps ($*$).

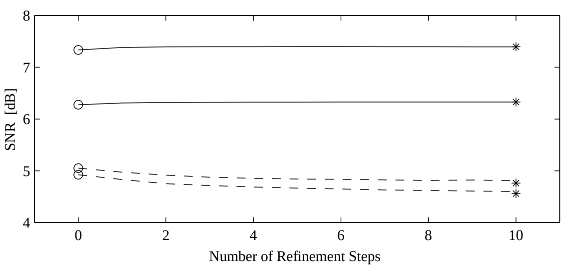


Figure 8 Average SNR corresponding to four different reconstructed noisy (voiced) speech segments using a TDC estimator with $\gamma = 2$, $p = 14$ and SNR=5dB. The ($*$) marks are the QSVD based estimates and voiced/unvoiced frames are given by solid/dashed lines. The average is taken over 100 independent noise realizations.

List of Figures

List of Tables

Bibliography

[1] G. Adams, M. F. Griffin, and G. W. Stewart. Direction-of-Arrival Estimation Using the Rank-Revealing URV Decomposition. In *ICASSP–91*, volume 2, pages 1385–1388. IEEE, 1991.

[2] W. Armbrüster, R. Czarnach, and P. Vary. Adaptive Noise Cancellation with Reference Input – Possible Applications and Theoretical Limits. In I.T. Young et al., editor, *Signal Processing III: Theories and Applications*, pages 391–394. EURASIP, Elsevier Science Publishers B.V., North-Holland, 1986.

[3] A. Akbari Azirani, R. Le Bouquin Jeannés, and G. Faucon. Speech Enhancement using a Wiener Filtering under Signal Presence Uncertainty. In *Signal Processing VIII Theories and Applications*, volume 2, pages 971–974. EUSIPCO-96, September 1996.

[4] Carlos A. Berenstein and E. Vincent Patrick. Exact Deconvolution for Multiple Convolution Operators – An Overview, Plus Performance Characterizations for Imaging Sensors. *Proceedings of the IEEE*, 78(4):723–734, April 1990.

[5] G. Bienvenu and L Kopp. Adaptivity to Background Noise Spatial Coherence for High Resolution Passive Methods. In *ICASSP–80*, pages 307–310. IEEE, 1980.

[6] Å. Björck. Stability Analysis of the Method of Seminormal Equations for Linear Least Squares Problems. *Linear Algebra and Its Applications*, 88/89:31–48, 1987.

[7] Å. Björck, L. Eldén, and H. Park. Accurate Downdating of Least Squares Solutions. *SIAM Journal on Matrix Analysis and Applications*, 15:549–568, 1994.

[8] Å. Björck and Gene H. Golub. Numerical Methods for Computing Angles Between Linear Subspaces. *Mathematics of Computation*, 27(123):579–594, July 1973.

[9] Åke Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.

[10] A. W. Bojanczyk. Modifications of a Generalized ULV Decomposition. School of Electrical Engineering, Cornell University, Ithaca, NY 14853-3801, 1993.

[11] S. F. Boll. Suppression of Acoustic Noise in Speech using Spectral Subtraction. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 27:113–120, April 1979.

[12] S. F. Boll and R. E. Wohlford. Event driven speech enhancement. In *ICASSP–83*, pages 1152–1155. IEEE, 1983.

[13] Kevin M. Buckley. Spatial/Spectral Filtering with Linearly Constrained Minimum Variance Beamformers. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(3):249–266, March 1987.

[14] Kevin M. Buckley and Lloyd J. Griffiths. Broad-Band Signal-Subspace Spatial-Spectrum (BASS-ALE) Estimation. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(7):953–964, July 1988.

[15] James R. Bunch and Christopher P. Nielsen. Updating the Singular Value Decomposition. *Numerische Mathematik*, 31:111–129, 1978.

[16] James A. Cadzow. Signal Enhancement – A Composite Property Mapping Algorithm. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(1):49–62, January 1988.

[17] James A. Cadzow and D. Mitchell Wilkes. Enhanced Sinusoidal and Exponential Data Modeling. In R. J. Vaccaro, editor, *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, pages 335–352, Amsterdam, 1991. Elsevier Science Publishers.

[18] T. F. Chan. Rank Revealing QR Factorizations. *Linear Algebra and Its Applications*, 88:67–82, 1987.

[19] A. K. Cline, A. R. Conn, and C. F. Van Loan. Generalizing the LINPACK condition estimator. In *Numerical Analysis*, volume 909 of *Lecture Notes in Mathematics*, pages 73–83. Springer Verlag, 1982.

[20] R. A. Curtis and R. J. Niederjohn. An Investigation of Several Frequency Domain Processing Methods for Enhancing the Intelligibility of Speech in Wideband Random Noise. In *ICASSP–78*, pages 602–605. IEEE, 1978.

[21] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization. *Mathematics of Computation*, 30(136):772–795, October 1976.

[22] N. Dal Degan and C. Prati. Acoustic Noise Analysis and Speech Enhancement Techniques for Mobile Radio Applications. *Signal Processing*, 15:43–56, July 1988.

[23] John R. Deller, John G. Proakis, and John H. L. Hansen. *Discrete-Time Processing of Speech Signals*. Macmillan, 113 Sylvan Avenue, Englewood Cliffs, N.J. 07632, 1993.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–88, 1977.

[25] M. Dendrinos, S. Bakamidis, and G. Carayannis. Speech Enhancement from Noise: A Regenerative Approach. *Speech Communication*, 10(1):45–57, February 1991.

[26] I. Dologlou and G. Carayannis. Physical Interpretation of Signal Reconstruction from Reduced Rank Matrices. *IEEE Trans. on Signal Processing*, 39(7):1681–1682, July 1991.

[27] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[28] A. Edelman and G. W. Stewart. Scaling for Orthogonality. *IEEE Trans. on Signal Processing*, 41(4):1676–1677, April 1993.

[29] Y. Ephraim and D. Malah. Speech Enhancement using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 32:1109–1121, December 1984.

[30] Yariv Ephraim, David Malah, and Biing-Hwang Juang. On the Application of Hidden Markov Models for Enhancing Noisy Speech. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37:1846–1856, December 1989.

[31] Yariv Ephraim and Harry L. Van Trees. A Signal Subspace Approach for Speech Enhancement. In *ICASSP–93*, volume 2, pages 355–358. IEEE, 1993.

[32] Yariv Ephraim and Harry L. Van Trees. A Signal Subspace Approach for Speech Enhancement. *IEEE Trans. on Speech and Audio Processing*, 3(4):251–266, July 1995.

[33] ETSI. GSM 06.01 – Speech Processing Functions: General Description. Valbonne Cedex, France, 1991.

[34] Ricardo D. Fierro. Perturbation Analysis for Two-Sided (or Complete) Orthogonal Decompositions. *SIAM Journal on Matrix Analysis and Applications*, 17(2):383–400, April 1996.

[35] Ricardo D. Fierro and James R. Bunch. Bounding the Subspaces from Rank Revealing Two-Sided Orthogonal Decompositions. *SIAM Journal on Matrix Analysis and Applications*, 16(3):743–759, July 1995.

[36] Ricardo D. Fierro and Per Christian Hansen. Accuracy of TSVD Solutions Computed from Rank-Revealing Decompositions. *Numerische Mathematik*, 70:453–471, 1995.

[37] J. L. Flanagan. *Speech Analysis, Synthesis and Perception*. Springer-Verlag, New York, 2 edition, 1972.

[38] O. L. Frost III. An Algorithm for Linearly Constrained Adaptive Array Processing. *Proc. of the IEEE*, 60(8):926–935, August 1972.

[39] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 2 edition, 1989.

[40] Martie M. Goulding and John S. Bird. Speech Enhancement for Mobile Telephony. *IEEE Trans. on Vehicular Technology*, 39:316–326, November 1990.

[41] Robert M. Gray. Toeplitz and Circulant Matrices: A review. Technical report, Information Systems Lab., Dep. of Electrical Engineering, Stanford University, 1997.

[42] Y. Grenier. A Microphone Array for Car Environments. *Speech Communication*, 12(1):25–39, March 1993.

[43] M. F. Griffin, E. C. Boman, and G. W. Stewart. Minimum-Norm Updating with the Rank-Revealing URV Decomposition. In *ICASSP–92*, volume 5, pages 293–296. IEEE, 1992.

[44] Lloyd J. Griffiths and Kevin M. Buckley. Quiescent Pattern Control in Linearly Constrained Adaptive Arrays. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(7):917–926, July 1987.

[45] Lloyd J. Griffiths and Charles W. Jim. An Alternative Approach to Linearly Constrained Adaptive Beamforming. *IEEE Trans. on Antennas and Propagation*, 30(1):27–34, January 1982.

[46] Ming Gu and Stanley C. Eisenstat. Downdating the Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 16(3):793–810, July 1995.

[47] William W. Hager. Condition estimates. *SIAM Journal on Sci. Statist. Comput.*, 5(2):311–316, 1984.

[48] John H. L. Hansen and Mark A. Clements. Constrained Iterative Speech Enhancement with Application to Speech Recognition. *IEEE Trans. on Signal Processing*, 39(4):795–805, April 1991.

[49] P. C. Hansen. Rank-Deficient Prewhitening with Quotient SVD and ULV Decompositions. To appear in BIT, 1997.

[50] P. C. Hansen and S. H. Jensen. Filter-Bank Interpretation of Reduced-Rank Noise Reduction. Scientific Computing Group Technical Report UNIC-95-12, UNI–C, Technical University of Denmark, DK-2800 Lyngby, Denmark, October 1995.

[51] P. C. Hansen and S. H. Jensen. FIR Filter Representations of Reduced-Rank Noise Reduction. To appear in IEEE Trans. on Signal Processing, 1997.

[52] P. S. K. Hansen. The ULV/ULLV Toolbox for Matlab. Department of Mathematical Modelling Technical Report IMM-REP-1997-06, Technical University of Denmark, DK-2800 Lyngby, Denmark, April 1997.

[53] P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Noise Reduction of Speech Signals using the Rank-Revealing ULLV Decomposition. In *Signal Processing VIII Theories and Applications*, volume 2, pages 967–970. EUSIPCO-96, September 1996.

[54] P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. ULV-Based Signal Subspace Methods for Speech Enhancement. In *The International Workshop on Acoustic Echo and Noise Control, IWAENC'97*, pages 9–12, Imperial College London, September 1997.

[55] Richard J. Hanson and Charles L. Lawson. Extensions and Applications of the Householder Algorithm for Solving Linear Least Squares Problems. *Mathematics of Computation*, 23:787–812, 1969.

[56] Simon Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, N.J. 07632, 1991.

[57] Nicholas J. Higham. A Survey of Condition Number Estimation for Triangular Matrices. *SIAM Review*, 29(4):575–596, December 1987.

[58] Nicholas J. Higham. The Test Matrix Toolbox for Matlab. Numerical Analysis Report No. 237, Departments of Mathematics, University of Manchester, Manchester M13 9PL, England, December 1993.

[59] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 1996.

[60] Srinath Hosur, Ahmed H. Tewfik, and Daniel Boley. ULV and Generalized ULV Tracking Adaptive Algorithms. Preprint, November 1995.

[61] S. H. Jensen. *Algorithms for Echo Cancellation and Noise Reduction in Digital Hands-Free Mobile Telephony.* PhD thesis, Technical University of Denmark, Electronics Institute, Building 349, DK-2800 Lyngby, Denmark, October 1994.

[62] S. H. Jensen, P. C. Hansen, S. D. Hansen, and J. Aa. Sørensen. Reduction of Broad-Band Noise in Speech by Truncated QSVD. *IEEE Trans. on Speech and Audio Processing*, 3(6):439–448, November 1995.

[63] Steven M. Kay. Noise Compensation for Autoregressive Spectral Estimates. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 28(3):292–303, June 1980.

[64] Ramdas Kumaresan, Louis L. Scharf, and Arnab K. Shaw. An Algorithm for Pole-Zero Modeling and Spectral Analysis. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(3):637–640, June 1986.

[65] Ramdas Kumaresan and Donald W. Tufts. Estimating the Parameters of Exponentially Damped Sinusoids and Pole-Zero Modeling in Noise. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 30(6):833–840, December 1982.

[66] S. Y. Kung. A New Identification and Model Reduction Algorithm via SVD. In *The 12th Asilomar Conference on Circuits, Syst., Comput.*, pages 705–714, Pacific Grove, CA, 1978.

[67] J. M. Lebak and A. W. Bojanczyk. Modifying a Rank-Revealing ULLV Decomposition. Technical report, Cornell Theory Center, June 1994.

[68] Fu Li and Richard J. Vaccaro. Unfied Analysis for DOA Estimation Algorithms in Array Signal Processing. *Signal Processing*, 25(2):147–169, November 1991.

[69] J. C. Liberti, T. S. Rappaport, and J. G. Proakis. Evaluation of Several Adaptive Algorithms for Canceling Acoustic Noise in Mobile Radio Environments. In *41th IEEE Vehicular Technology Conference*, pages 126–132, St. Louis, Mo., USA, May 1991.

[70] J. S. Lim. Evaluation of a Correlation Subtraction Method for Enhancing Speech Degraded by Additive White Noise. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26:471–472, October 1978.

[71] J. S. Lim and A. V. Oppenheim. All-Pole Modeling of Degraded Speech. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26:197–210, June 1978.

[72] J. S. Lim and A. V. Oppenheim. Enhancement and Bandwidth Compression of Noisy Speech. *Proceedings of the IEEE*, 67:1586–1604, December 1979.

[73] F. T. Luk and S. Qiao. A New Matrix Decomposition for Signal Processing. In M. S. Moonen et al., editor, *Linear Algebra for Large Scale and Real-Time Applications*, pages 241–247. Kluwer Academic Publishers, 1993.

[74] John Makhoul. On the Eigenvectors of Symmetric Toeplitz Matrices. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29(4):868–872, August 1981.

[75] R. Mathias and G. W. Stewart. A Block QR Algorithm and the Singular Value Decomposition. *Linear Algebra and Its Applications*, 182:91–100, 1993.

[76] Roy Mathias. Analysis of Algorithms for Orthogonalizing Products of Unitary Matrices. *Numerical Linear Algebra with Applications*, 3(2):125–145, 1996.

[77] R. J. McAulay and M. L. Malpass. Speech Enhancement using a Soft-Decision Noise Suppression Filter. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 28:137–145, April 1980.

[78] J. H. McClellan and D. Lee. Exact Equivalence of the Steiglitz-Mcbride Iteration and IQML. *IEEE Trans. on Signal Processing*, 39(2):509–512, February 1991.

[79] M. Moonen, P. Van Dooren, and Joos Vandewalle. A Note on Efficient Numerically Stabilized Rank-One Eigenstructure Updating. *IEEE Trans. on Signal Processing*, 39(8):1911–1913, August 1991.

[80] M. Moonen, P. Van Dooren, and Joos Vandewalle. Singular Value Decomposition Updating Algorithm for Subspace Tracking. *SIAM Journal on Matrix Analysis and Applications*, (4):1015–1038, October 1992.

[81] B. De Moor and H. Zha. A tree of generalizations of the ordinary singular value decomposition. *Linear Algebra and Its Applications*, 147:469–500, 1991.

[82] Bart De Moor. The Singular Value Decomposition and Long and Short Spaces of Noisy Matrices. *IEEE Trans. on Signal Processing*, 41(9):2826–2838, September 1993.

[83] Bart De Moor. Total Least Squares for Affinely Structured Matrices and the Noisy Realization Problem. *IEEE Trans. on Signal Processing*, 42(11):3104–3131, November 1994.

[84] Eric Moulines, Pierre Duhamel, Jean-François Cardoso, and Sylvie Mayrargue. Subspace Methods for the Blind Identification of Multichannel FIR Filters. *IEEE Trans. on Signal Processing*, 43(2):516–525, February 1995.

[85] Flemming Mühlbach and Michael B. Jørgensen. ULLV-dekomposition og støjreduktion af talesignaler i mobiltelefoni. Master's thesis, Technical University of Denmark, Electronics Institute, Building 349, DK-2800 Lyngby, Denmark, March 1994. (in Danish).

[86] Stephen Oh, Vishua Viswanathan, and Panos Papamichalis. Hands-Free Voice Communication in an Automobile with a Microphone Array. In *ICASSP–92*, volume 1, pages 281–284. IEEE, 1992.

[87] Serge J. Olszanskyj, James M. Lebak, and Adam W. Bojanczyk. Rank-k Modification Methods for Recursive Least Squares Problems. *Numerical Algorithms*, (7):325–354, 1994.

[88] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J. 07632, 1989.

[89] C. C. Paige. Error Analysis of Some Techniques for Updating Orthogonal Decompositions. *Mathematics of Computations*, 34(150):465–471, April 1980.

[90] C. C. Paige and M. A. Saunders. Toward a Generalized Singular Value Decomposition. *SIAM Journal on Numerical Analysis*, 18:398–405, 1981.

[91] Haesun Park and Lars Eldén. Downdating the Rank-Revealing URV Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 16:138–155, 1995.

[92] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1980.

[93] Sanzheng Qiao. Computing the ULLV Decomposition. CRL Technical Report 278, Communications Research Laboratory, McMaster University, Hamilton, Ontario L8S 4K1, Canada, January 1994.

[94] T. F. Quatieri and R. J. McAulay. Phase Coherence in Speech Reconstruction for Enhancement and Coding Applications. In *ICASSP–89*, pages 207–210. IEEE, May 1989.

[95] T. F. Quatieri and R. J. McAulay. Noise Reduction using a Soft-Decision Sine-Wave Vector Quantizer. In *ICASSP–90*, pages 821–824. IEEE, April 1990.

[96] Wallace R, B and R. A. Goubran. Noise Cancellation Using Parallel Adaptive Filters. *IEEE Trans. on Circuits and Systems*, 39(4):239–243, April 1992.

[97] Henrik Sahlin and Holger Broman. Signal Separation Applied to Real World Signals. In *The International Workshop on Acoustic Echo and Noise Control, IWAENC'97*, pages 73–76, Imperial College London, September 1997.

[98] Pascal Scalart, Jozue V. Filho, and José G. Chiquito. On Speech Enhancement Algorithms Based on MMSE Estimation. In *Signal Processing VIII Theories and Applications*, volume 1, pages 471–474. EUSIPCO-96, September 1996.

[99] Louis L. Scharf. *Statistical Signal Processing.* Addison-Wesley, 1991.

[100] Louis L. Scharf. The SVD and Reduced-Rank Signal Processing. In R. J. Vaccaro, editor, *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, pages 3–31, Amsterdam, 1991. Elsevier Science Publishers.

[101] Louis L. Scharf. The SVD and Reduced Rank Signal Processing. *Signal Processing*, 25(2):113–133, November 1991.

[102] Louis L. Scharf and Donald W. Tufts. Rank Reduction for Modeling Stationary Signals. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(3):350–355, March 1987.

[103] R. O. Schmidt. *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation.* PhD thesis, Stanford University, Standford, CA, 1981.

[104] Rajesh Sharma and Barry D. Van Veen. Large Modular Structures for Adaptive Beamforming and the Gram-Schmidt Preprocessor. *IEEE Trans. on Signal Processing*, 42(2):448–451, February 1994.

[105] Pavel Sovka, Petr Pollak, and Jan Kybic. Extended Spectral Subtraction. In *Signal Processing VIII Theories and Applications*, volume 2, pages 963–966. EUSIPCO-96, September 1996.

[106] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980.

[107] G. W. Stewart. An Updating Algorithm for Subspace Tracking. *IEEE Trans. on Signal Processing*, 40(6):1535–1541, June 1992.

[108] G. W. Stewart. Error Analysis of QR Updating with Exponential Windowing. *Mathematics of Computation*, 59(199):135–140, July 1992.

[109] G. W. Stewart. On the Early History of the Singular Value Decomposition. *SIAM Review*, 35(4):551–566, December 1993.

[110] G. W. Stewart. Updating a Rank-Revealing ULV Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, April 1993.

[111] G. W. Stewart. On the Stability of Sequential Updates and Downdates. *IEEE Trans. on Signal Processing*, 43(11):2642–2648, November 1995.

[112] John K. Thomas, Louis L. Scharf, and Donald W. Tufts. The Probability of a Subspace Swap in the SVD. *IEEE Trans. on Signal Processing*, 43(3):730–736, March 1995.

[113] Lang Tong, Guanghan Xu, and Thomas Kailath. Blind Identification and Equalization of Multipath Channels. In *International Conference on Communications, ICC '92*, volume 3, pages 1513–1517. IEEE, 1992.

[114] Lang Tong, Guanghan Xu, and Thomas Kailath. Fast Blind Equalization via Antenna Arrays. In *ICASSP–93*, volume 4, pages 272–275. IEEE, 1993.

[115] Donald W. Tufts and Ramdas Kumaresan. Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood. *Proceedings of the IEEE*, 70(9):975–989, September 1982.

[116] Donald W. Tufts, Ramdas Kumaresan, and Ivars Kirsteins. Data Adaptive Signal Estimation by Singular Value Decomposition of a Data Matrix. *Proceedings of the IEEE*, 70(6):684–685, June 1982.

[117] Donald W. Tufts and Abhijit A. Shah. Estimation of a Signal Waveform from Noisy Data Using Low-Rank Approximation to a Data Matrix. *IEEE Trans. on Signal Processing*, 41(4):1716–1721, April 1993.

[118] Sabine Van Huffel. Enhanced Resolution based on Minimum Variance Estimation and Exponential Data Modeling. *Signal Processing*, 33(3):333–355, September 1993.

[119] C. F. Van Loan. Generalizing the Singular Value Decomposition. *SIAM Journal on Numerical Analysis*, 13:76–83, 1976.

[120] C. F. Van Loan. Computing the CS and Generalized Singular Value Decomposition. *Numerische Mathematik*, 46:479–491, 1985.

[121] Peter van Overschee and Bart De Moor. *Subspace Identification for Linear Systems.* Kluwer Academic Publishers, 1996.

[122] Barry D. Van Veen. An Analysis of Several Partially Adaptive Beamformer Designs. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(2):192–203, February 1989.

[123] Peter Vary. Noise Suppression by Spectral Magnitude Estimation – Mechanism and Theoretical Limits. *Signal Processing*, 8(4):387–400, July 1985.

[124] D. L. Wang and J. S. Lim. The Unimportance of Phase in Speech Enhancement. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 30:679–681, August 1982.

[125] B. Widrow, J. R. Grover, J. M. McCool, J. Kaunitz, C.S. Williams, R. H. Hearn, J. S. Zeidler, E. Dong, and R. C. Goodlin. Adaptive Noise Canceling: Principles and Applications. *Proceedings of the IEEE*, 63:1692–1716, December 1975.

[126] D. M. Wilkes and M. H. Hayes. Block Toeplitz Approximation. *Signal Processing*, 15(3):303–313, October 1988.

[127] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.

[128] Fatoş T. Yarman-Vural. Enhancement of Speech in Additive, Locally Stationary and Colored Noise, using Linear Prediction. *Signal Processing*, 20(3):211–217, July 1990.

Index

I

K

L

M

N

O

P

Ph. D. Theses from IMM

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences.* *xiv* + 143 pp.
2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems.* *xiii* + 232 pp.
3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns.* *xv* + 136 pp.
4. **Melgaard, Henrik.** (1994). *Identification of physical models.* *xvii* + 246 pp.
5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system.* *xxii* + 153 pp.
6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data.* *xxiv* + 213 pp.
7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications.* *xviii* + 345 pp.
8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxxviii* + 282 pp., 283-569 pp.
9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer.* ATV Erhvervsforskerprojekt EF 435. *viii* + 174 pp.
10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* ATV Erhvervsforskerprojekt EF 385. Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).
11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions.* *iv* + 154 pp.
12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems.* *xiv* + 348 pp.
13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects.* *xiv* + 194 pp.
14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems.* *xiv* + 120 pp.
15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations.* *iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems. xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation. xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods. xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics. viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points. xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics. xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surgery simulation. xxvii* + 274 pp.

26. **Bendtsen, Claus.** (1996). *Parallel numerical algorithms for the solution of systems of ordinary differential equations. viii* + 79 pp.

27. **Lauritsen, Morten Bach.** (1997). *Delta-domain predictive control and identification for control. xxii* + 292 pp.

28. **Bischoff, Svend.** (1997). *Modelling Colliding-pulse mode-locked semiconductor lasers. xxii* + 217 pp.

29. **Arnbjerg-Nielsen, Karsten.** (1997). *Statistical analysis of urban hydrology with special emphasis on rainfall modelling.* Institut for Miljøteknik, DTU. *xiv* + 161 pp.

30. **Jacobsen, Judith L.** (1997). *Dynamic modelling of processes in rivers affected by precipitation runoff. xix* + 213 pp.

31. **Sommer, Helle Mølgaard.** (1997). *Variability in microbiological degradation experiments. - Analysis and case study. xiv* + 211 pp.

32. **Ma, Xin.** (1997). *Adaptive extremum control and wind turbine control. xix* + 293 pp.

33. **Rasmussen, Kim Ørskov.** (1997). *Nonlinear and stochastic dynamics of coherent structures. x* + 215 pp.

34. **Hansen, Lars Henrik.** (1997). *Stochastic modelling of central heating systems. xxii* + 301 pp.

35. **Jørgensen, Claus.** (1997). *Driftsoptimering på kraftvarmesystemer.* 290 pp.

36. **Stauning, Ole.** (1997.) *Automatic validation of numerical solutions.* *xiv* + 118 pp.

37. **Pedersen, Morten With.** (1997.) it Optimization of recurrent neural networks for time series modeling. X + 322 pp.

38. **Thorsen, Rune.** (1997.) *Restoration of hand function in tetraplegics using myoelectrically controlled functional electrical stimulation of the controlling muscle.* *viii* + 141 pp. + Appendix.

39. **Rosholm, Anders.** (1997.) *Statistical methods for segmentation and classification of images.* *xiv* + 181 pp.

40. **Petersen, Kim Tilgaard.** (1997.) *Estimation of speech quality in telecommunication systems.* *x* + 259 pp.

41. **Jensen, Carsten Nordstrøm.** (1997.) *Nonlinear systems with discrete and continuous elements.* 205 pp.

42. **Hansen, Peter S.K.** (1997.) *Signal subspace methods for speech enhancement.* *x* + 214 pp.