(12)

# SUBSPACE SIGNAL PROCESSING IN STRUCTURED NOISE

by

Richard Travis Behrens

B.S., Walla Walla College, 1985

M.S., University of Colorado, 1988

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

1990

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | December 1990 | Technical Report 10/88 - 12/90 |

**4. TITLE AND SUBTITLE**

Subspace Signal Processing in Structured Noise

**5. FUNDING NUMBERS**

N00014-89-J-1070

**6. AUTHOR(S)**

Richard T. Behrens

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Electrical & Computer Engineering
University of Colorado
Campus Box 425
Boulder, CO 80309-0425

**8. PERFORMING ORGANIZATION REPORT NUMBER**

DSP-502

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Office of Naval Research
Statistics & Probability Branch
Mathematics Division
800 North Quincy Avenue
Arlington, VA 22217

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Unclassified

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

Some common types of noise can be dealt with by applying a linear model to the noise as well as to the signal. ~~We describe~~ noise that obeys a low-rank linear model as structured noise and derive several signal processing methods based on a structured noise model. Whereas orthogonal projection operators play a key role in the solution of classical linear estimation and detection problems, the addition of a structured noise term to the model leads to oblique projection operators in the new solutions. ~~We~~ consider several subspace identification problems in the context of a structured noise model. ~~We~~ also consider parameter estimation with structured noise, where ~~we assume that~~ the signal and structured noise subspaces are known or have been identified from observed data. ~~We~~ apply these results to the decoding of complex number codes for detection and correction of impulse errors.

**14. SUBJECT TERMS**

**15. NUMBER OF PAGES**
145

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | None |

# SUBSPACE SIGNAL PROCESSING IN STRUCTURED NOISE

by

Richard Travis Behrens

B.S., Walla Walla College, 1985

M.S., University of Colorado, 1988





A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

1990

Behrens, Richard Travis (Ph.D., Electrical and Computer Engineering)

Subspace Signal Processing in Structured Noise

Thesis directed by Professor Louis L. Scharf

Linear signal models are commonly used in digital signal processing, leading naturally to the use of linear subspaces to separate signals from noise. A linear model is often a realistic model for a signal. In other cases a linear model represents a good approximation to the signal and is used because of its mathematical convenience.

Some common types of noise can also be dealt with by applying a linear model to the noise as well as to the signal. We describe noise that obeys a low rank linear model as structured noise, and derive several signal processing methods based on a structured noise model.

Whereas orthogonal projection operators play a key role in the solution of classical linear estimation and detection problems, the addition of a structured noise term to the model leads to oblique projection operators in the new solutions. Because of the importance of oblique projection operators, one chapter explores their properties.

Subspace identification is the determination of the modes of a linear signal or a structured noise source based on observed data. We consider the identification of signal subspaces with no prior knowledge about the signal except that it obeys a low rank linear model. We then consider signal subspace identification with the prior knowledge that the signal is a superposition of complex exponentials. We extend these identification techniques to a structured noise model by considering the identification of structured noise subspaces with varying degrees of prior knowledge about the signal and the structured noise. We propose a method of adaptively updating existing signal and noise subspace models based on new data. And we consider the issue of order selection when identifying subspaces.

Another category of contributions is parameter estimation with structured noise. Here we assume that the signal and structured noise subspaces are known or have been identified from observed data. We derive oblique projections for estimating signals with varying prior knowledge about the parameter distributions.

We apply these results to the decoding of complex number codes for detection and correction of impulse errors. In this example we apply both subspace identification and parameter estimation techniques.

To Debbie,

who thought I would finish school and get a job five and one half years before now.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Scharf for his help and guidance, both technical and personal, throughout my graduate studies. Without his support and encouragement this research would not have been possible. I am also deeply indebted to Professor Mullis, who taught me to appreciate and enjoy the intricacies of linear algebra. Professor Mathys has been a great help in defining the relationships between signal processing and coding theory, and it has been a pleasure working with him on this research. I would like to thank the other members of my thesis committee, Professors Etter, Avery and Holley, for their willingness to participate and for their time.

Financial support for this research came from Ball Aerospace Systems Division and from the Office of Naval Research. I would like to thank both sources for their specific contributions and for their favorable disposition towards basic research. I would also like to thank the MathWorks, Inc. for supplying MATLAB$^{TM}$, and acknowledge their trademark.

I would like to thank my wife Debbie for her love, encouragement and patience. I cannot imagine succeeding in this endeavor without the stability she provided at home.

I would like to thank my parents for everything that lead up to this. I hope they find this advanced degree a worthy addition to the excellent education they provided for me.

These acknowledgements cannot be complete without being a book unto themselves, but I must mention my gratitude to my big sister Janet for sparking my interest in mathematics before I was in the first grade, and to Julie Fredlund for a wide variety of advice and help in producing this document.

CONTENTS

# TABLES

Table

# FIGURES

Figure

# CHAPTER I

## Introduction

We present here a collection of advancements in the theory and practice of digital signal processing, based on the use of linear subspaces. All subspace signal processing techniques share the common goal of mitigating the effects of noise. Usually they are used in the context of some kind of an estimation or detection problem, as in the SVD based modification of linear prediction discovered by Tufts and Kumaresan [TuK82] where the coefficients of a whitening filter for a given signal are to be estimated. Their method takes advantage of the fact that linear combinations of complex exponential signals will lie in a subspace whose rank is equal to the number of different complex exponentials present. It follows that any components of the received data that lie outside this low rank subspace are due to noise. This is a typical example of subspace signal processing where the signal of interest is assumed to lie in a low rank linear subspace.

But noise comes in many varieties, from the background hiss of an analog magnetic audio tape to the sharp crackle of lightning striking a telephone wire. Previous techniques of noise suppression through subspace signal processing have been oriented toward the former variety of noise. They model the desired signal as a vector that lies in a low rank subspace, and the noise as a random vector that may fall anywhere in the observation space. The next level in noise modeling is to allow correlated noise by applying a shaped probability density to the noise vector. We go a step farther and allow total dependence of some noise samples by assuming that a significant component of the noise lies in some linear subspace. We call noise components that lie in a linear subspace "structured noise". Many of the new signal processing methods we propose deal with structured noise.

The application of a linear model to noise is arguably just as reasonable as the application of a linear model to signal. For what we consider as our desired signal in one problem may become interference in the next. Power transmission waves at 60 Hz may be signal to the power engineer. For most everyone else they represent a ubiquitous form of structured noise. Lightning may cause impulsive noise (large amplitude noise that affects only a few data samples), which is also a form of structured noise.

Subspace signal processing in the structured noise model may be divided into two main parts. First the signal and noise subspaces must be identified. Chapters III through V of this dissertation deal with the problem of subspace identification. Once identified, the subspaces must be used to some advantage in solving estimation or detection problems. Chapter VI applies the subspace models to several estimation problems.

## 1.1 Overview

This chapter contains a general introduction and outline of the dissertation, a summary of the research contributions of this work, and an introduction to linear modeling of signals and noise. We also begin to establish our mathematical notation. Several types of signals that are well represented by linear models are discussed, and it is shown how a model matrix for a linear model is formed for several cases.

Chapter II covers some of the specialized linear algebra necessary for the signal subspace techniques presented in later chapters. Particular emphasis is given to projection operators, their properties, and how to construct them for given subspaces. The distinction between orthogonal projections and oblique projections is emphasized, and a coordinate transformation is derived which relates the two.

In Chapter III we consider the problem of identifying linear subspaces from observed data. The chapter begins with a critical evaluation of the principle of Maximum Likelihood (ML), concluding that it is most appropriate for sets of parameters that are uniformly distributed, or at least not known to be highly nonuniform. We then present identification techniques for both signal subspaces and noise subspaces.

In Chapter IV we extend the subspace identification techniques of Chapter III to the case where we have a prior model for the signal which imposes structural constraints on the subspace estimates. Specifically we consider signals composed of complex exponential modes whose subspaces must therefore be spanned by Vandermonde type matrices (we follow Demeure [Dem89] in applying the term Vandermonde to non-square matrices whose columns are complex power series). We present improvements to two existing algorithms for identifying such subspaces. We then extend one of the algorithms to deal with the presence of structured noise.

Order selection, the process of choosing the appropriate rank of a signal subspace or structured noise subspace, is an important aspect of subspace identification. Chapter V addresses some problems in subspace order selection.

We consider a special set of estimation problems in Chapter VI. The distinguishing feature of these problems is the use of linear models for both signal and noise simultaneously. In other words, they are problems of signal (or parameter) estimation in structured noise. A common thread in most of the solutions is the appearance of oblique projection operators. The subspaces identified analytically or by the techniques of Chapters III through V are used to determine oblique projections to be used for signal processing.

Chapter VII concludes this dissertation with a summary of what has been accomplished, conclusions and limitations, and suggestions for extending the research.

## 1.2 Related Work

The problem of estimating the frequencies of multiple sinusoids is viewed here as a subspace identification problem. This problem has been addressed by many researchers, with some of the more notable papers being published by Prony [Pro1795], Rife and Boorstyn [RiB76], Tufts and Kumaresan [TuK82], Starer and Nehorai [StN88], and Kumaresan, Scharf and Shaw [KSS86]. The order selection aspect of the subspace identification problem has been addressed by Fuchs [Fuc88], Tuan [Tua88], Kumaresan, Tufts and Scharf [KTS84], Wax and Kailath [WaK85], and Akaike [Aka74].

Regarding the parameter estimation problems of Chapter VI, related work has been published by Marshall [Mar84], [Mar85], [Mar86], Wolf [Wol83], Kumaresan [Kum85], and Scharf, Mathys and Behrens [SMB87] in the context of error correction codes and burst errors. Our original presentation of the structured noise estimation problems addressed in Chapter VI is [BeS88].

For a treatment of the classical least squares problem without linearly modeled noise, see Golub and Van Loan [GVL89] or Lawson and Hanson [LaH74].

### 1.3 Contributions

We now summarize the original research contributions of this dissertation, indicating, where appropriate, the foundational work we have built upon. Specific contributions are

1) The emphasis on oblique projection operators as useful tools in signal processing.

2) The equations in Chapter II for construction of an oblique projection with a specified range and null space.

3) The representation in Chapter II of an oblique projection as coordinate transformation plus orthogonal projection.

4) The relationship in Chapter II between the singular values of an oblique projection and the principal angles between its range and null space.

5) The critical evaluation in Chapter III of the Maximum Likelihood principle and the example using a quadratic equation to illustrate the pitfalls of blind application of ML.

6) The extensions in Chapter III of the ML subspace identification technique of Scharf [Sch91]. One extension is to deal with complex data and unknown noise variance. Another is to deal with a constraint on the identified subspace. The last and most significant extension is to deal with the presence of structured noise.

7) The presentation in Chapter III of the heretofore unpublished method of Steve Voran [[]unpublished notes] for using Total Least Squares to update signal subspace models. and the extension of that method to allow simultaneous updates of signal subspaces and structured noise subspaces.

8) The incorporation in Chapter IV of the technique of Starer and Nehorai [StN88] for enforcing constraints on the AR parameters into the algorithm of Kumaresan, Scharf and Shaw [KSS86] (the KiSS algorithm, also called IQML [BrM86]) for finding the ML estimates of those parameters. Also a corrected and clarified presentation of the KiSS algorithm and a discussion of some implementation issues.

9) The extension in Chapter IV of the derivations by Starer and Nehorai [StN88] of the gradient and Hessian of the KiSS objective function to the case of complex data and parameters. Also the derivation of more elegant expressions for the gradient and Hessian leading to a filtering interpretation. The extension to complex data and parameters is more substantial than it may first appear, because of the complexity of the equations involved.

10) The extension in Chapter IV of the KiSS algorithm to deal with structured noise.

11) The derivation in Chapter V of a new order selection rule for rank reduction in the Linear Statistical Model. We first presented this result at the IEEE International Symposium on Information Theory in San Diego, January 1990, [BeS90].

12) The derivation in Chapter V of a Bayes hypothesis test for order selection in the identification of structured noise subspaces.

13) The oblique projection estimators in Chapter VI for signal estimation in the presence of structured noise. We first presented these results at the Asilomar Conference on Signals, Systems and Computers, November 1988, [BeS88].

14) The application in Chapter VI to decoding block codes over the real and complex number fields. We first presented this result at the Asilomar Conference on Signals, Systems and Computers, November 1987, [SMB87].

### 1.4 The Linear Model: Notation and Terminology

We represent a scalar by any symbol in an italic font, such as $n$. All vectors are column vectors and are represented by a symbol with an underbar, such as $\underline{x}$. Contexts requiring a row vector will be handled with the transpose of a column vector. All matrices are represented by symbols in a bold font, and are usually upper case, such as $\mathbf{H}$. The subspace spanned by the

columns of a matrix is represented with angle brackets around the symbol for the matrix, such as $\langle \mathbf{H} \rangle$.

A superscript $T$ is used to indicate the transpose of a matrix or vector, such as $\mathbf{H}^T$. For complex matrices we must distinguish between the plain transpose and the complex conjugate (Hermitian) transpose. We use superscript $H$ for Hermitian transpose and $T$ for plain transpose. The complex conjugate alone is represented by a superscript $*$. A circumflex over any variable generally represents an estimate of that variable, such as $\hat{\underline{x}}$ for an estimate of $\underline{x}$.

Except where noted all signals referred to in this dissertation are discrete time signals. In the linear model a signal is characterized as a weighted sum (linear combination) of certain modes. The set of weights determines a specific signal out of the class of signals which obey the model.

The convenience and power of a linear algebraic framework apply naturally to vector-valued signals. But even scalar-valued signals discrete-time can be placed in that framework by considering finite length (windowed) observations as vectors. Arrange the signal modes as columns of a matrix $\mathbf{H}$ and the mode weights as elements of a vector $\underline{\theta}$. The product of the mode matrix and the weight vector is the signal, $\underline{x} = \mathbf{H}\underline{\theta}$. This linear model for the signal places $\underline{x}$ in a finite dimensional linear subspace known as the signal subspace, and spanned by the columns of the mode matrix $\mathbf{H}$. The mode weights $\underline{\theta}$ of a specific signal parameterize its position within the signal subspace.

Let $\underline{x}$ be an $n$-vector representing the signal of interest. The linear model says that

$$\underline{x} = \mathbf{H}\underline{\theta},$$

$$\underset{n}{\begin{bmatrix} \\ \underline{x} \\ \\ \end{bmatrix}} = \underset{n \times m}{\begin{bmatrix} \\ \mathbf{H} \\ \\ \end{bmatrix}} \quad \underset{m}{\begin{bmatrix} \\ \underline{\theta} \\ \\ \end{bmatrix}} \tag{1.1}$$

Of primary interest to us is the so-called overdetermined case, where $n > m$ so that there are more observations than parameters. In this case, the space $\langle \mathbf{H} \rangle$ is the signal subspace.

## 1.5 The Linear Statistical Model

When additive random noise affects a linear signal, the resulting received data vector $\underline{y}$ obeys what has been called the linear statistical model:

$$\underline{y} = \underline{x} + \underline{\nu}$$

$$\underset{n}{\begin{bmatrix} \underline{y} \end{bmatrix}} = \overbrace{\underset{n \times m}{\begin{bmatrix} \mathbf{H} \end{bmatrix}} \underset{m}{\begin{bmatrix} \underline{\theta} \end{bmatrix}}} + \underset{m}{\begin{bmatrix} \underline{\nu} \end{bmatrix}}. \tag{1.2}$$

Here $\underline{\nu}$ is an $n$-vector of random noise. This model has found many applications in digital signal processing, such as those presented by Scharf [Sch91], Dunn [Dun86], and Buckley [Buc87].

## 1.6 The Structured Noise Model

A more appropriate model in many situations is the following generalization of the linear statistical model. This model for the received data is illustrated in Figure 1.1. The signal parameter $\underline{\theta}$ sets initial conditions, or excites, the linear system $\mathbf{H}$ to produce the signal $\underline{x}$. Noise added in the communication channel is modeled in two parts: the unstructured noise $\underline{\nu}$, and the structured noise $\underline{b}$ that results from an underlying process $\underline{\varphi}$ exciting the linear system $\mathbf{S}$.

The received data $\underline{y}$ is the sum

$$\underline{y} = \underline{x} + \underline{b} + \underline{\nu}$$

$$\underset{n}{\begin{bmatrix} \underline{y} \end{bmatrix}} = \overbrace{\underset{n \times m}{\begin{bmatrix} \mathbf{H} \end{bmatrix}} \underset{m}{\begin{bmatrix} \underline{\theta} \end{bmatrix}}} + \overbrace{\underset{n \times t}{\begin{bmatrix} \mathbf{S} \end{bmatrix}} \underset{t}{\begin{bmatrix} \underline{\varphi} \end{bmatrix}}} + \underset{n}{\begin{bmatrix} \underline{\nu} \end{bmatrix}}. \tag{1.3}$$

The additional term accounts for what we call structured, or low rank, noise that lies in the rank $t$ subspace $\langle \mathbf{S} \rangle$. It can be any signal which obeys a linear model but interferes with the signal of primary interest. In some cases its power may be comparable to or even greater than the power of the desired signal, resulting in a very poor signal-to-noise ratio when processed

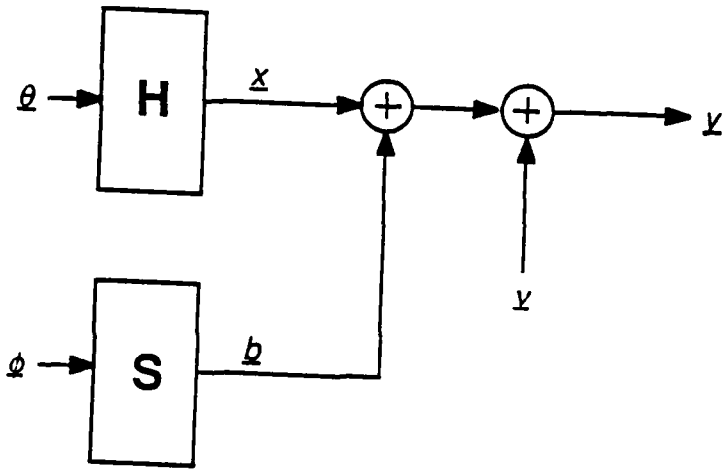


Figure 1.1 Block diagram of the structured noise model.

according to the ordinary linear statistical model. The additional term in the model allows knowledge about the structure of such noise to be used to its full advantage in processing the received data.

The matrices **H** and **S** are both assumed to have full column rank. In some of the developments which follow it is also necessary that they be linearly independent, so that the composite matrix [**H S**] also has full column rank. It is necessary, but not sufficient, that

$$m + t \leq n, \tag{1.4}$$

where $m$ and $t$ are the widths (and ranks) of **H** and **S**, and $n$ is the dimension of the measurement space (length of **H** and **S**). We **do not** require that **H** be orthogonal to **S**.

### 1.7 Motivation for the Model

The linear model is quite versatile in terms of the types of signals which obey it. The linear model includes the entire family of ARMA impulse responses such as complex exponentials, sinusoids, damped sinusoids, real exponentials, and sums of any of these. Impulse and burst signals also fit the framework of the linear model. Sometimes the linear model can

serve as an acceptable approximation to a nonlinear signal. One such example is when a nonlinear signal is band limited. We give some examples at the end of this chapter that show how several specific signals may be represented in the linear model.

In signal processing, noise is usually treated as a full rank process in the measurement space. However, in many situations it is more advantageous (or more realistic) to model part of the noise as a process occurring in a space of lower dimensionality, which is then mapped into the measurement space by some physical system. When the physical system is a linear map, the structured noise obeys a linear model. In the measurement space, the resulting noise is low rank and exhibits a structure dependent on the physical system. Thus we use the terms *low rank noise* and *structured noise* interchangeably.

The structured noise model of Equation 1.3 applies the modeling versatility of the linear model to both the signal and the noise simultaneously. It is especially appropriate in any environment containing several competing signals, each of which constitutes noise from the perspective of the others.

A classic example of competing signals occurs in any multiuser communication channel, such as the broadcast spectrum. Previous approaches to this problem have often centered around making the signals of each user orthogonal to all other users' signals. But there may be limits to the amount of control a designer has over the competing users. Orthogonality between the various competing signal subspaces is clearly not always attainable, and the structured noise model gives us a tool for dealing with competing signals without the orthogonality requirement.

## 1.8 Examples

Power line noise can appear in received data as a sinusoid of known frequency (e.g. 60 Hz or 50 Hz). It lies in a rank 2 linear subspace and the two unknown parameters *amplitude* and *phase* determine the position within that subspace. The Vandermonde matrix which spans

this structured noise subspace is

$$\mathbf{S} = \begin{bmatrix} (e^{j2\pi f/F})^0 & (e^{-j2\pi f/F})^0 \\ (e^{j2\pi f/F})^1 & (e^{-j2\pi f/F})^1 \\ (e^{j2\pi f/F})^2 & (e^{-j2\pi f/F})^2 \\ \vdots & \vdots \\ (e^{j2\pi f/F})^{n-1} & (e^{-j2\pi f/F})^{n-1} \end{bmatrix}, \tag{1.5}$$

where $f$ is the power line frequency and $F = 1/T$ is the sampling frequency. The two parameters which multiply the columns of $\mathbf{S}$ are not actually amplitude and phase, but are another parameterization of the amplitude and phase information.

The character of the problem changes somewhat when the frequency of the interfering sinusoid is unknown. Frequency enters the equation nonlinearly, through the Vandermonde matrix $\mathbf{S}$. Determination of frequency is thus equivalent to a subspace identification problem and is treated in Chapter IV.

For a band limited signal, a linear model does not necessarily apply, but an approximating linear model can be constructed using discrete prolate spheroidal wave functions (DPSWF's) [Sle78]. Begin by forming the autocorrelation matrix $\mathbf{R}$ of the band limited signal. For a signal with power spectrum

$$S(e^{j\omega}) = \begin{cases} 1 & \text{if } |\omega| \leq \Omega, \\ 0 & \text{if } \Omega < |\omega| \leq \pi, \end{cases} \tag{1.6}$$

the elements of the autocorrelation matrix $\mathbf{R}$ are given by

$$r_{ij} = \frac{\Omega}{\pi} \frac{\sin \Omega(i-j)}{Omega(i-j)}, \qquad i, j = 1 \ldots n. \tag{1.7}$$

The eigenvectors of $\mathbf{R}$ are index limited DPSWF's and are the vectors used to form the signal subspace. Choosing the eigenvectors corresponding to the $m$ largest eigenvalues results in the $m$-dimensional signal subspace containing the greatest possible portion of the signal energy.

We have given two examples of signals whose linear model may be constructed from theoretical considerations. With that, we would like to move into the estimation of signal subspaces in situations where theoretical models are insufficient to completely determine the subspace. But first we must lay some mathematical groundwork, and we turn to that in the next chapter.

# CHAPTER II

## Useful Mathematical Results

In this chapter we present some of the specialized mathematics used in the remainder of the dissertation. The most significant results presented in this chapter are those involving oblique projections. These include the oblique projection construction formulas, the coordinate transformation, and the connection between oblique projections and principal angles between subspaces.

### 2.1 Linear Subspaces and Spans

Our signal subspace processing algorithms work in the context of a vector space of $n$ complex elements. In most cases the same results apply to real $n$-dimensional space. A set of $m$ linearly independent vectors in such a vector space spans an $m$-dimensional linear subspace. The subspace is the collection of all vectors that can be expressed as a linear combination of the $m$ spanning vectors. We usually arrange the vectors of a span as columns of a matrix. If $\mathbf{H}$ is such a matrix, we designate the subspace spanned by the columns of $\mathbf{H}$ as $\langle \mathbf{H} \rangle$.

The orthogonal complement to a linear subspace $\langle \mathbf{H} \rangle$ is the linear subspace consisting of all vectors orthogonal to $\langle \mathbf{H} \rangle$, that is, all vectors orthogonal to every column of $\mathbf{H}$. We use the symbol $\langle \mathbf{H} \rangle^{\perp}$ to represent the orthogonal complement of $\langle \mathbf{H} \rangle$. In $n$-space, if $\langle \mathbf{H} \rangle$ is of dimension $m$, then $\langle \mathbf{H} \rangle^{\perp}$ is of dimension $n - m$. We also use the term *perp-space* to refer to the orthogonal complement of $\langle \mathbf{H} \rangle$.

The intersection of two linear subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ is the linear subspace consisting of all vectors that are contained in both $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$. The intersection may be trivial, containing only the zero-vector, in which case we say that the subspaces are *non-overlapping*. Non-overlapping does not imply orthogonality between subspaces. Orthogonality is a stronger

condition and it does imply a trivial intersection. A necessary and sufficient condition for subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ to be non-overlapping is that the composite matrix [H S] be of full column rank. This of course requires the sum of the subspace dimensionalities to be less than or equal to $n$.

## 2.2 Vandermonde and Toeplitz Spans

A rectangularly windowed ARMA impulse response with $m$ simple poles $z_1 \ldots z_m$ lies in an $m$-dimensional linear subspace spanned by a matrix of the form:

$$\mathbf{H} = \begin{bmatrix} z_1^0 & \cdots & z_m^0 \\ z_1^1 & \cdots & z_m^1 \\ \vdots & & \vdots \\ z_1^{n-1} & \cdots & z_m^{n-1} \end{bmatrix}. \tag{2.1}$$

Such a matrix is called a *Vandermonde matrix* when $m = n$ [GVL89]. We follow Demeure [Dem89] in using the term Vandermonde to apply also to the nonsquare matrix. Note that the subspace depends only on the AR parameters, since they alone determine the pole locations. The position of the ARMA impulse response within the subspace $\langle \mathbf{H} \rangle$ is determined by the MA parameters.

Let $a_0 \ldots a_m$ be the AR parameters, that is, the coefficients of the monic ($a_0 = 1$) polynomial whose roots are the pole locations $z_1 \ldots z_m$:

$$\sum_{j=0}^{m} a_j z_i^{-j} = 0, \qquad i = 1 \ldots m. \tag{2.2}$$

Then the Toeplitz matrix of these coefficients

$$\mathbf{A} = \begin{bmatrix} a_m^* & & 0 \\ \vdots & \ddots & \\ a_0^* & & \\ & \ddots & a_m^* \\ & & \vdots \\ 0 & & a_0^* \end{bmatrix} \in \mathbb{C}^{n \times (n-m)} \tag{2.3}$$

is orthogonal to the Vandermonde matrix H:

$$\mathbf{A}^H \mathbf{H} = 0. \tag{2.4}$$

This orthogonality is easily verified by application of Equation 2.2. Since the rank of $\mathbf{A}$ is guaranteed to be $n - m$ it follows that the orthogonal complement of the Vandermonde $\langle \mathbf{H} \rangle$ of Equation 2.1 is the Toeplitz $\langle \mathbf{A} \rangle$ of Equation 2.3. That is, $\langle \mathbf{H} \rangle^{\perp} = \langle \mathbf{A} \rangle$.

Vandermonde and Toeplitz matrices are not generally orthogonal spans for their subspaces. Where it is necessary to find an orthogonal span for a subspace defined by some non-orthogonal span, we use either a QR decomposition of the spanning matrix or its SVD.

## 2.3 Projection Operators

By the term *projection* we mean a matrix that is idempotent (equal to its own square):

$$\mathbf{E}^2 = \mathbf{E}. \tag{2.5}$$

The eigenvalues of a projection are equal to 0 or 1. However, a matrix whose eigenvalues are 0 or 1 is not necessarily a projection.

**Orthogonal projections.** Most mentions of projections in the literature refer only to orthogonal projections, the subset of idempotent matrices for which the null space is orthogonal to the range. In other words, an orthogonal projection whose range is $\langle \mathbf{H} \rangle$ has null space $\langle \mathbf{H} \rangle^{\perp}$. A necessary and sufficient condition for a projection to be orthogonal is Hermitian symmetry:

$$\mathbf{P}^H = \mathbf{P}. \tag{2.6}$$

For an orthogonal projection $\mathbf{P_H}$ whose range is $\langle \mathbf{H} \rangle$ and whose null space is $\langle \mathbf{A} \rangle = \langle \mathbf{H} \rangle^{\perp}$, we have

$$\begin{aligned} \mathbf{P_H H} &= \mathbf{H}, \\ \mathbf{P_H A} &= \mathbf{0}. \end{aligned} \tag{2.7}$$

**Oblique projections.** Projection matrices which are not orthogonal are referred to as *oblique projections*. Oblique projections are idempotent but not symmetric. This more general class of projections plays a key role in the structured noise problems of Chapters III through VI. Since an oblique projection lacks symmetry, its null space and range are not orthogonal. For an oblique projection $\mathbf{E_{H:S}}$ whose range is $\langle \mathbf{H} \rangle$ and whose null space is $\langle \mathbf{S} \rangle$, we have

$$\begin{aligned} \mathbf{E_{H:S} H} &= \mathbf{H}, \\ \mathbf{E_{H:S} S} &= \mathbf{0}. \end{aligned} \tag{2.8}$$

We use the following notation for projection operators. Orthogonal projections are represented as $\mathbf{P}$, usually with a subscript indicating the range. Oblique projections are represented as $\mathbf{E}$, usually with a double subscript referring first to the range and second to the null space.

**Construction of Projections.** We now give equations that will allow projection matrices to be built from subspace spans for desired ranges and null spaces. Other formulas, not equivalent to ours, for building oblique projections are given in [KaW89]. Assume that $\mathbf{H}$ is a complex matrix of size $n \times m$ having full column rank, and likewise that $\mathbf{S}$ is a complex matrix of size $n \times t$ having full column rank. Assume further that $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ are non-overlapping, which implies $m + t \leq n$.

The well known formula to build an orthogonal projection whose range is $\langle \mathbf{H} \rangle$ is

$$\mathbf{P}_{\mathbf{H}} = \mathbf{H}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H. \tag{2.9}$$

The orthogonal projection whose range is $\langle \mathbf{H} \rangle^\perp$ is given by

$$\mathbf{P}_{\mathbf{H}^\perp} = \mathbf{I} - \mathbf{P}_{\mathbf{H}}. \tag{2.10}$$

The last projection operator may be obtained in another way which is of some use in subsequent theoretical analyses

$$\mathbf{P}_{\mathbf{H}^\perp} = \lim_{r\to\infty} (\mathbf{I} + r\mathbf{H}\mathbf{H}^H)^{-1}. \tag{2.11}$$

The following proof uses the Sherman-Morrison-Woodbury matrix inversion formula [GVL89]:

$$\begin{aligned}
&\lim_{r\to\infty} (\mathbf{I} + r\mathbf{H}\mathbf{H}^H)^{-1} \\
&= \lim_{r\to\infty} (\mathbf{I} - r\mathbf{H}(\mathbf{I} + r\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H) \\
&= \lim_{r\to\infty} \left(\mathbf{I} - \mathbf{H}(\frac{1}{r}\mathbf{I} + \mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\right) \\
&= \mathbf{I} - \mathbf{H}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \\
&= \mathbf{P}_{\mathbf{H}^\perp}.
\end{aligned} \tag{2.12}$$

To build an oblique projection whose range is $\langle \mathbf{H} \rangle$ and whose null space contains $\langle \mathbf{S} \rangle$

take either of the expressions

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}), \\ \mathbf{E}_{\mathbf{H};\mathbf{S}} &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}. \end{aligned} \tag{2.13}$$

Any remainder of Euclidean space, orthogonal to both $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$, is also in the null space of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$. These expressions for oblique projections merit verification, since they are, as far as we know, new.

To verify that the first expression for $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ is idempotent, consider

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{E}_{\mathbf{H};\mathbf{S}} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp})\mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}) \\ &= (\mathbf{P}_{\mathbf{H}}\mathbf{P}_{\mathbf{H}} - \mathbf{P}_{\mathbf{H}}\mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{P}_{\mathbf{H}})(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}) \\ &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}) \\ &= \mathbf{E}_{\mathbf{H};\mathbf{S}}. \end{aligned} \tag{2.14}$$

In the preceding sequence of steps we use the fact that $\mathbf{P}_{\mathbf{H}}$ is itself idempotent and that $\mathbf{P}_{\mathbf{H}^\perp}\mathbf{P}_{\mathbf{H}} = \mathbf{0}$. Now we check the range and null space:

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{H} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp})\mathbf{H} \\ &= \mathbf{P}_{\mathbf{H}}\mathbf{H} - \mathbf{P}_{\mathbf{H}}\mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{H} \\ &= \mathbf{H}; \end{aligned} \tag{2.15}$$

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{S} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp})\mathbf{S} \\ &= \mathbf{P}_{\mathbf{H}}(\mathbf{S} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S}) \\ &= \mathbf{P}_{\mathbf{H}}(\mathbf{S} - \mathbf{S}) \\ &= \mathbf{0}. \end{aligned} \tag{2.16}$$

Thus $\langle \mathbf{H} \rangle$ is in the range of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ and $\langle \mathbf{S} \rangle$ is in the null space. Finally, if $\mathbf{A}$ spans the perp-space to $\langle \mathbf{H}, \mathbf{S} \rangle$ then $\mathbf{P}_{\mathbf{H}^\perp}\mathbf{A} = \mathbf{A}$, and $\mathbf{P}_{\mathbf{H}}\mathbf{A} = \mathbf{0}$, and $\mathbf{S}^H\mathbf{A} = \mathbf{0}$, so

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{A} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp})\mathbf{A} \\ &= \mathbf{P}_{\mathbf{H}}\mathbf{A} - \mathbf{P}_{\mathbf{H}}\mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{A} \\ &= \mathbf{0} - \mathbf{P}_{\mathbf{H}}\mathbf{S}(\mathbf{S}^H\mathbf{P}_{\mathbf{H}^\perp}\mathbf{S})^{-1}\mathbf{S}^H\mathbf{A} \\ &= \mathbf{0} \end{aligned} \tag{2.17}$$

and we see that $\langle \mathbf{A} \rangle$ is also in the null space of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$. Since we have accounted for all available dimensions we have determined that the range of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ is equal to $\langle \mathbf{H} \rangle$ and the null space is equal to $\langle \mathbf{S}, \mathbf{A} \rangle$.

To verify that the second expression for $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ in Equation 2.13 is idempotent, consider

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{E}_{\mathbf{H};\mathbf{S}} &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp} \\ &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp} \qquad (2.18) \\ &= \mathbf{E}_{\mathbf{H};\mathbf{S}}. \end{aligned}$$

Check its range and null space:

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{H} &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H} \\ &= \mathbf{H}; \qquad (2.19) \end{aligned}$$

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{S} &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{S} \\ &= \mathbf{0}; \qquad (2.20) \end{aligned}$$

$$\begin{aligned} \mathbf{E}_{\mathbf{H};\mathbf{S}}\mathbf{A} &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{A} \\ &= \mathbf{H}(\mathbf{H}^H\mathbf{P}_{\mathbf{S}^\perp}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{A} \qquad (2.21) \\ &= \mathbf{0}. \end{aligned}$$

Thus the second expression is a projection with the same range and null space as the first expression. Therefore they are equal.

Another useful pair of identities follows from the two expressions for $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ in Equation 2.13:

$$\mathbf{E}_{\mathbf{H};\mathbf{S}} = \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{E}_{\mathbf{S};\mathbf{H}}), \qquad (2.22)$$

$$\mathbf{E}_{\mathbf{S};\mathbf{H}} = \mathbf{P}_{\mathbf{S}}(\mathbf{I} - \mathbf{E}_{\mathbf{H};\mathbf{S}}). \qquad (2.23)$$

Where $\mathbf{E}_{\mathbf{S};\mathbf{H}}$ is the oblique projection with range $\langle \mathbf{S} \rangle$ and null space $\langle \mathbf{H}, \mathbf{A} \rangle$.

**Singular values of projections.** It is well known that the singular values of an orthogonal projection matrix are, like its eigenvalues, 0 or 1. This is true because for a symmetric matrix the singular values are equal to the absolute values of the eigenvalues. Since the 2-norm of a matrix is equal to its largest singular value, orthogonal projections have unit 2-norm and

will never make a vector longer by projection:

$$||\mathbf{P}\underline{x}||_2 \leq ||\underline{x}||_2. \tag{2.24}$$

For an oblique projection this is not the case. We will show in section 2.6 that the singular values of an oblique projection can be 0, 1 or any value greater than 1. It follows that oblique projections can have a 2-norm greater than unity and that $||\mathbf{E}\underline{x}||_2$ may be greater than $||\underline{x}||_2$.

## 2.4 A Three-Way Resolution of Euclidean Space

Given a subspace $\langle \mathbf{H} \rangle$ and a subspace $\langle \mathbf{S} \rangle$, define a new subspace $\langle \mathbf{A} \rangle$ as the portion of Euclidean space orthogonal to both $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$. That is, $\langle \mathbf{A} \rangle = \langle \mathbf{H}, \mathbf{S} \rangle^{\perp}$. Now any vector in Euclidean space can be expressed uniquely as the sum of three components, one each in $\langle \mathbf{H} \rangle$, $\langle \mathbf{S} \rangle$ and $\langle \mathbf{A} \rangle$. This resolves Euclidean space into three pieces as shown in Figure 2.1.

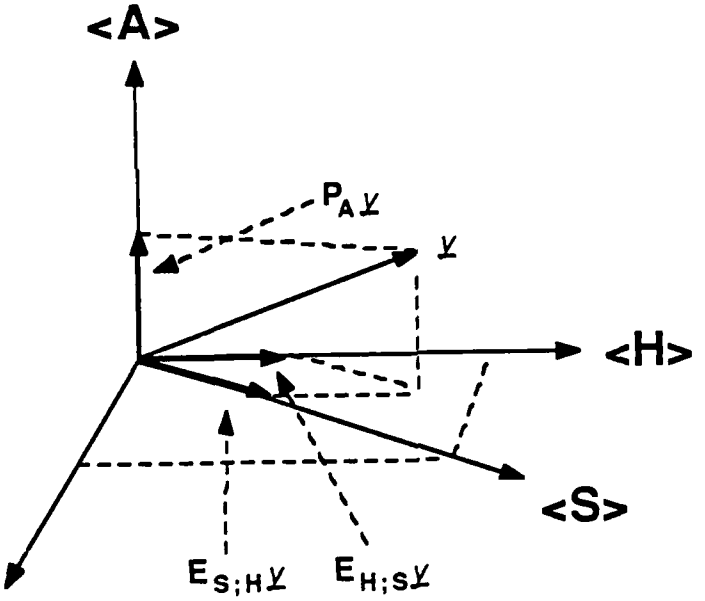


Figure 2.1 Three-way resolution of Euclidean space.

Corresponding to this geometric resolution is the algebraic identity

$$\mathbf{I} = \mathbf{E}_{\mathbf{H};\mathbf{S}} + \mathbf{E}_{\mathbf{S};\mathbf{H}} + \mathbf{P}_{\mathbf{A}}. \tag{2.25}$$

A corollary to Equation 2.25 is

$$\mathbf{P}_{\mathbf{HS}} = \mathbf{E}_{\mathbf{H};\mathbf{S}} + \mathbf{E}_{\mathbf{S};\mathbf{H}}, \tag{2.26}$$

where $\mathbf{P_{HS}}$ is the orthogonal projection whose range is $\langle \mathbf{H}, \mathbf{S} \rangle$.

Figure 2.1 is also useful to show how the oblique projection $\mathbf{E_{H;S}}$ works. We say that $\mathbf{E_{H;S}}$ projects $\underline{y}$ onto $\langle \mathbf{H} \rangle$ along $\langle \mathbf{S}, \mathbf{A} \rangle$. By this we mean that $\mathbf{E_{H;S}}\underline{y}$ lies in $\langle \mathbf{H} \rangle$, and that the difference $(\mathbf{I} - \mathbf{E_{H;S}})\underline{y}$ lies in $\langle \mathbf{S}, \mathbf{A} \rangle$.

## 2.5 A Coordinate Transformation for Oblique Projection

In this section, we characterize a general oblique projection operator as the composition of a coordinate transformation and an orthogonal projection, as shown in Figure 2.2. The required coordinate transformation $\mathbf{F}$ is derived to satisfy

$$\mathbf{E_{H;S}} = \mathbf{P_H F}. \tag{2.27}$$

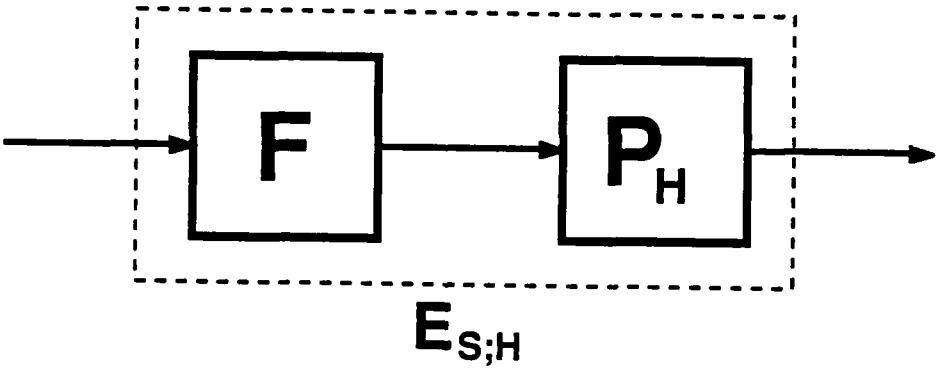


Figure 2.2 A characterization of oblique projections.

Assume we have an oblique projection $\mathbf{E_{H;S}}$ whose range is $\langle \mathbf{H} \rangle$ and whose null space is $\langle \mathbf{S}, \mathbf{A} \rangle$, where $\langle \mathbf{A} \rangle$ is defined as $\langle \mathbf{H}, \mathbf{S} \rangle^{\perp}$. The coordinate transformation $\mathbf{F}$ should rotate vectors in the subspace $\langle \mathbf{S} \rangle$ to a new subspace $\langle \mathbf{S}' \rangle$, while leaving vectors in $\langle \mathbf{H}, \mathbf{A} \rangle$ unaffected. The new subspace $\langle \mathbf{S}' \rangle$ must be orthogonal to $\langle \mathbf{H} \rangle$ to put it in the null space of $\mathbf{P_H}$. To complete

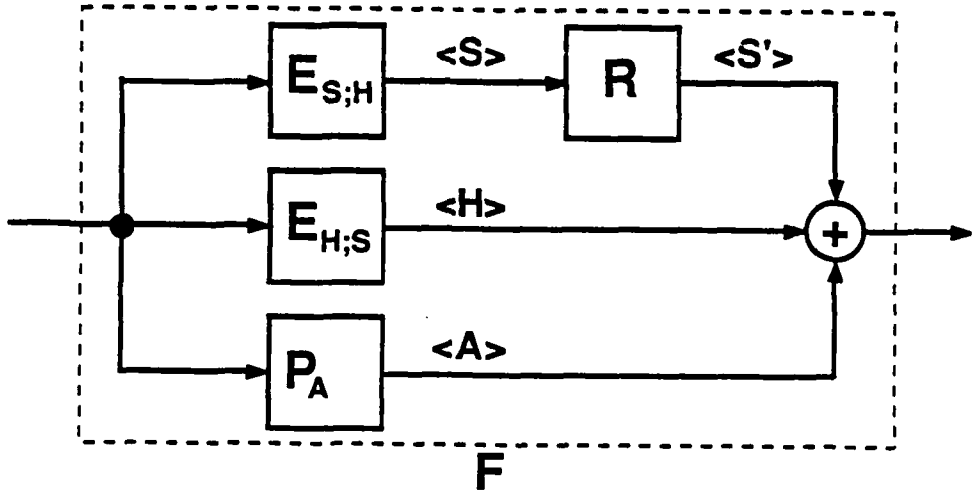


Figure 2.3 Building up the coordinate transformation.

the determination of $\langle \mathbf{S}' \rangle$, we also choose it to be orthogonal to $\langle \mathbf{A} \rangle$, resulting in the definition

$$\langle \mathbf{S}' \rangle = \langle \mathbf{H}, \mathbf{A} \rangle^{\perp}. \tag{2.28}$$

It is easily seen that $\langle \mathbf{S}' \rangle$ has the same dimensionality as $\langle \mathbf{S} \rangle$. This characterization of the desired coordinate transformation leads immediately to the representation shown in Figure 2.3, where $\mathbf{R}$ is the required rotation from $\langle \mathbf{S} \rangle$ to $\langle \mathbf{S}' \rangle$. The coordinate transformation is given by

$$\mathbf{F} = \mathbf{E}_{\mathbf{H};\mathbf{S}} + \mathbf{P}_{\mathbf{A}} + \mathbf{R}\mathbf{E}_{\mathbf{S};\mathbf{H}}. \tag{2.29}$$

The transformation $\mathbf{F}$ is not a rotation. A rotation which moves vectors in $\langle \mathbf{S} \rangle$ to $\langle \mathbf{S}' \rangle$ is given by

$$\mathbf{R} = \mathbf{Q}_{\mathbf{S}'}\mathbf{Q}_{\mathbf{S}}{}^{H}, \tag{2.30}$$

where $\mathbf{Q}_{\mathbf{S}}$ is any orthogonal span of $\langle \mathbf{S} \rangle$ and $\mathbf{Q}_{\mathbf{S}'}$ is any orthogonal span of $\langle \mathbf{S}' \rangle$. Note that $\mathbf{R}$ could be any mapping from $\langle \mathbf{S} \rangle$ to $\langle \mathbf{S}' \rangle$ and $\mathbf{F}$ would still satisfy Equation 2.27. The rotation was chosen to preserve the length of vectors in the subspace $\langle \mathbf{S} \rangle$.

## 2.6 Principal Angles

Principal angles between subspaces are a generalization of the geometrical concept of angles between lines and planes. Given two subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ of $n$-dimensional space there is a set of angles formed between them. The number of such angles is equal to the dimensionality of the lower rank subspace.

Golub and Van Loan [GVL89] give the following definition for principal angles:

$$\alpha_i = \arccos( \max_{\underline{u} \in \langle \mathbf{H} \rangle} \max_{\underline{v} \in \langle \mathbf{S} \rangle} \underline{u}^H \underline{v}) = \arccos(\underline{u}_i^H \underline{v}_i), \tag{2.31}$$

subject to

$$\begin{aligned} &\underline{u}^H \underline{u} = \underline{v}^H \underline{v} = 1, \\ &\underline{u}^H \underline{u}_j = 0 \qquad j = 1, \ldots, i-1 \\ &\underline{v}^H \underline{v}_j = 0 \qquad j = 1, \ldots, i-1. \end{aligned} \tag{2.32}$$

Note that the definition is recursive in that the vectors $\underline{u}$ and $\underline{v}$ for the $i^{th}$ principal angle are constrained to be orthogonal to all previous $\underline{u}_j$ and $\underline{v}_j$ respectively. Golub and Van Loan also show that the principal angles may be computed with the Singular Value Decomposition as follows. Let $\mathbf{U_H}$ be an orthogonal span for $\langle \mathbf{H} \rangle$, and $\mathbf{U_S}$ an orthogonal span for $\langle \mathbf{S} \rangle$. Then the principal angles between $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ are given by

$$\alpha_i = \arccos \lambda_i, \tag{2.33}$$

where $\lambda_i$ is a singular value of the product $\mathbf{U_H}^H \mathbf{U_S}$.

We extend these results as follows. For an oblique projection $\mathbf{E_{H;S}}$ formed from subspace spans $\mathbf{H}$ and $\mathbf{S}$ according to Equation 2.13, the singular values of the projection matrix are directly related to the principal angles between the two subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$. Let the singular values of $\mathbf{E_{H;S}}$ be denoted by $\sigma_i$ and the principal angles by $\alpha_i$. Then

$$\sigma_i = \frac{1}{\sin(\alpha_i)}. \tag{2.34}$$

To prove this, begin by noting that $\lambda_i$ being a singular value of $\mathbf{U_H}^H \mathbf{U_S}$ means that $\lambda_i^2$ is an eigenvalue of $\mathbf{U_H}^H \mathbf{U_S} \mathbf{U_S}^H \mathbf{U_H} = \mathbf{U_H}^H \mathbf{P_S} \mathbf{U_H}$. Substituting according to Equation 2.33

we have

$$\begin{array}{rccc} & \cos^2\alpha_i & \text{is an e.v. of} & \mathbf{U_H}^H\mathbf{P_S}\mathbf{U_H} \\ \Rightarrow & 1-\cos^2\alpha_i & \text{is an e.v. of} & \mathbf{I}-\mathbf{U_H}^H\mathbf{P_S}\mathbf{U_H} \\ \Rightarrow & \sin^2\alpha_i & \text{is an e.v. of} & \mathbf{U_H}^H(\mathbf{I}-\mathbf{P_S})\mathbf{U_H} \\ \Rightarrow & \frac{1}{\sin^2\alpha_i} & \text{is an e.v. of} & (\mathbf{U_H}^H\mathbf{P_{S^\perp}}\mathbf{U_H})^{-1}. \end{array} \tag{2.35}$$

Since eigenvalues are invariant to an orthogonal transformation, this also implies that

$$\frac{1}{\sin^2\alpha_i} \qquad \text{is an e.v. of} \qquad \mathbf{U_H}(\mathbf{U_H}^H\mathbf{P_{S^\perp}}\mathbf{U_H})^{-1}\mathbf{U_H}^H. \tag{2.36}$$

The matrix in Equation 2.36 is equal to $\mathbf{E_{H;S}}\mathbf{E_{H;S}}^H$, as can be easily verified by using the span $\mathbf{U_H}$ in the second form of Equation 2.13 for $\mathbf{E_{H;S}}$. It therefore follows that $(1/\sin\alpha_i)$ is a singular value of $\mathbf{E_{H;S}}$ and the proof is complete.

A corrolary to Equation 2.34 is that the singular values $\sigma_i$ of an oblique projection that correspond to principal angles $\alpha_i$ are in the interval $[1,\infty)$. Because an oblique projection is low rank, it also has singular values equal to zero that do not correspond to principal angles. Thus, the singular values of an oblique projection matrix may be 0, 1, and any value greater than 1.

# CHAPTER III

## Subspace Identification with No Prior Model

The first task in subspace based signal processing is to identify the signal subspace and, if appropriate, the structured noise subspace. Sometimes these subspaces can be identified from theoretical considerations, as for example when the structured noise is 60 Hz power line noise with unknown amplitude and phase. In other cases we must resort to observed data to identify the subspaces. Even then we may or may not have enough prior knowledge about the signal to impose constraints on the subspace estimate. In this chapter we consider the problem of estimating signal and noise subspaces without structural constraints.

We begin with a general discussion of the Maximum Likelihood (ML) principle in which we urge caution in the application of ML estimators, especially in the context of the ML invariance principle. We then present an algorithm for ML estimation of signal subspaces, and another algorithm for simultaneous ML estimation of signal and noise subspaces. The chapter ends with an application of Total Least Squares for updating signal and noise subspace estimates based on new data. All of the subspace identification algorithms in this chapter make use of the Singular Value Decomposition (SVD).

### 3.1 The Maximum Likelihood Principle

We use the principle of Maximum Likelihood (ML) to derive several of our subspace identification methods. In ML subspace identification a joint probability density is assumed for the observations $\underline{y}$. This density is a function of the signal subspace which is in turn a function of some set of parameters $\underline{a}$. The likelihood function for a given observation $\underline{y}_0$ is equal to the probability density for $\underline{y}$ evaluated at the observation $\underline{y}_0$ and considered a function of the parameters $\underline{a}$. It is customary to simplify the likelihood function by dropping any constants

that do not affect the location of the maximum in terms of the parameters $\underline{a}$. The ML estimate of $\underline{a}$ is the value of $\underline{a}$ that maximizes the likelihood function. Since the signal subspace is a function of $\underline{a}$, we can apply the invariance property of ML estimation [Sch91] to say that we have also found the ML estimate of the signal subspace.

I must digress to discuss the worthiness (or unworthiness) of the Maximum Likelihood principle. In some ways the ML principle is philosophically unattractive. Its basic assumption is that whatever observation you make must have been a relatively likely observation. But this need not be the case—unlikely realizations can and do occur, especially when the variance is large. A more attractive principle of estimation is Maximum A posteriori Probability (MAP). [1] In MAP estimation, one chooses the most likely parameter values given the observation and a prior density on the parameters. If we consider the parameters as random variables, the ML and MAP rules can be stated in a parallel fashion as

$$\begin{aligned} \text{ML}: &\quad \max_{\underline{a}} f_{\underline{y}|\underline{a}}(\underline{y}_0|\underline{a}); \\ \text{MAP}: &\quad \max_{\underline{a}} f_{\underline{a}|\underline{y}}(\underline{a}|\underline{y}_0). \end{aligned} \tag{3.1}$$

Thus while ML makes the observation likely, MAP makes the choice of parameters likely. Unfortunately MAP estimation requires the additional knowledge of the probability density of the parameters. Since this density is not always known, we cannot always use MAP.

In spite of the philosophical oddity behind ML estimation, ML estimators have some desirable properties that make them a good choice when the parameter density is unknown. First note that ML often corresponds to least squares estimation. More specifically, when the observations consist of signal plus zero-mean white Gaussian noise the ML estimator is the same as the least squares estimator wherein the parameters are chosen to minimize squared error between the observation vector and the mean vector (a function of the parameters). For colored noise, the ML estimator corresponds to a weighted least squares solution. The following quadratic form represents both the least squares objective function and the negative log of the

[1] A better name would be Maximum A posteriori Likelihood. [Sch91].

likelihood function:

$$(\underline{y}_0 - \underline{m})^H \mathbf{R}^{-1} (\underline{y}_0 - \underline{m}). \tag{3.2}$$

Here $\underline{m}$ is the mean of $\underline{y}$ as a function of the parameters $\underline{a}$, and $\mathbf{R}$ is the noise covariance matrix in the ML problem and $\mathbf{R}^{-1}$ is the weighting matrix in the least squares problem.

The case for the ML principle is further bolstered by the following argument. We might express the lack of prior knowledge about the distribution of the parameters by assigning a uniform prior probability density over the entire parameter domain $D$ (assume for the moment that the domain is finite in extent). If we do so, the philosophically attractive MAP estimator becomes identical to the ML estimator. This can be seen by application of Bayes' rule to invert the conditional densities:

$$\max_{\underline{a} \in D} f_{\underline{a}|\underline{y}}(\underline{a}|\underline{y}_0) = \max_{\underline{a} \in D} f_{\underline{y}|\underline{a}}(\underline{y}_0|\underline{a}) \left( \frac{f_{\underline{a}}(\underline{a})}{f_{\underline{y}}(\underline{y}_0)} \right). \tag{3.3}$$

ML and MAP are the same when $f_{\underline{a}}(\underline{a})$ is constant since the ratio on the right hand side of Equation 3.3 is then constant with respect to the maximization over $\underline{a}$.

What if the parameter domain $D$ is infinite in extent? Then a uniform density would have a value of zero everywhere, and the MAP estimator would be undefined. In this case we cannot make the preceding argument that ML corresponds to MAP, but we can argue that it doesn't matter because this case is physically unrealistic. To simplify the argument let us assume that the parameter is a real scalar $a$, and the domain $D$ is a subset of the real line with the Lebesgue measure of $D$ infinite. Then for any finite $M > 0$, the measure of the set $A = \{x : |x| \leq M\}$ is $2M$, which is finite. Thus the probability that $a \in A$ is 0 under a uniform density over an infinite $D$. It is difficult to imagine a parameter in the real world that has probability 1 of being larger than every finite number $M$. We conclude that in all physically meaningful cases, the ML estimator corresponds to the MAP estimator with a uniform prior density on the parameters.

This connection between ML and MAP serves as a justification of ML as long as we have no reason to reject a uniform prior density of the parameters. But we should be aware

that by using ML we are giving tacit approval to a uniform prior. This raises an interesting question with regard to the invariance principle of ML estimation. The invariance principle states that if $\underline{g}$ is a deterministic function of $\underline{a}$,

$$\underline{g} = G(\underline{a}), \tag{3.4}$$

and if $\hat{\underline{a}}$ is the ML estimator of $\underline{a}$, then $\hat{\underline{g}} = G(\hat{\underline{a}})$ is the ML estimator of $\underline{g}$ [Sch91]. The problem is that for most functions $G$, it is inconsistent to allow that both $\underline{a}$ and $\underline{g}$ are uniformly distributed. If one set of parameters is uniform we can calculate a specific nonuniform density for the other set. The question is whether or not the ML estimators $\hat{\underline{a}}$ and $\hat{\underline{g}}$ obtained by the invariance principle can be simultaneously appropriate. This question is addressed in the following example.

**Example: Complex Quadratic Roots.** Consider the real second order polynomial equation

$$z^2 + a_1 z + a_2 = 0, \tag{3.5}$$

with a complex conjugate pair of roots $z_1$ and $z_2$. Suppose we have made some observations that allow us to estimate the coefficients $a_1$ and $a_2$ and/or the roots $z_1$ and $z_2$. Label the root with positive imaginary part by $z_1$. The estimate of the roots might be expressed by $(\rho = |z_1|, \theta = \angle z_1)$, or by $(\alpha = \mathrm{Re}\, z_1, \beta = \mathrm{Im}\, z_1)$. The estimate of $z_2$ is determined by $z_1$. If we assume the roots lie inside the unit circle, corresponding to a stable causal system, we can specify the domain of each of the three sets of parameters:

$$\text{For } (\rho, \theta) : D_{\rho\theta} = \{(\rho, \theta) : 0 < \rho < 1, 0 < \theta < \pi\}. \tag{3.6}$$

$$\text{For } (\alpha, \beta) : D_{\alpha\beta} = \{(\alpha, \beta) : \alpha^2 + \beta^2 < 1, \beta > 0\}. \tag{3.7}$$

$$\text{For } (a_1, a_2) : D_{a_1 a_2} = \{(a_1, a_2) : \frac{a_1^2}{4} < a_2 < 1\}. \tag{3.8}$$

These parameter domains are shown in Figure 3.1.

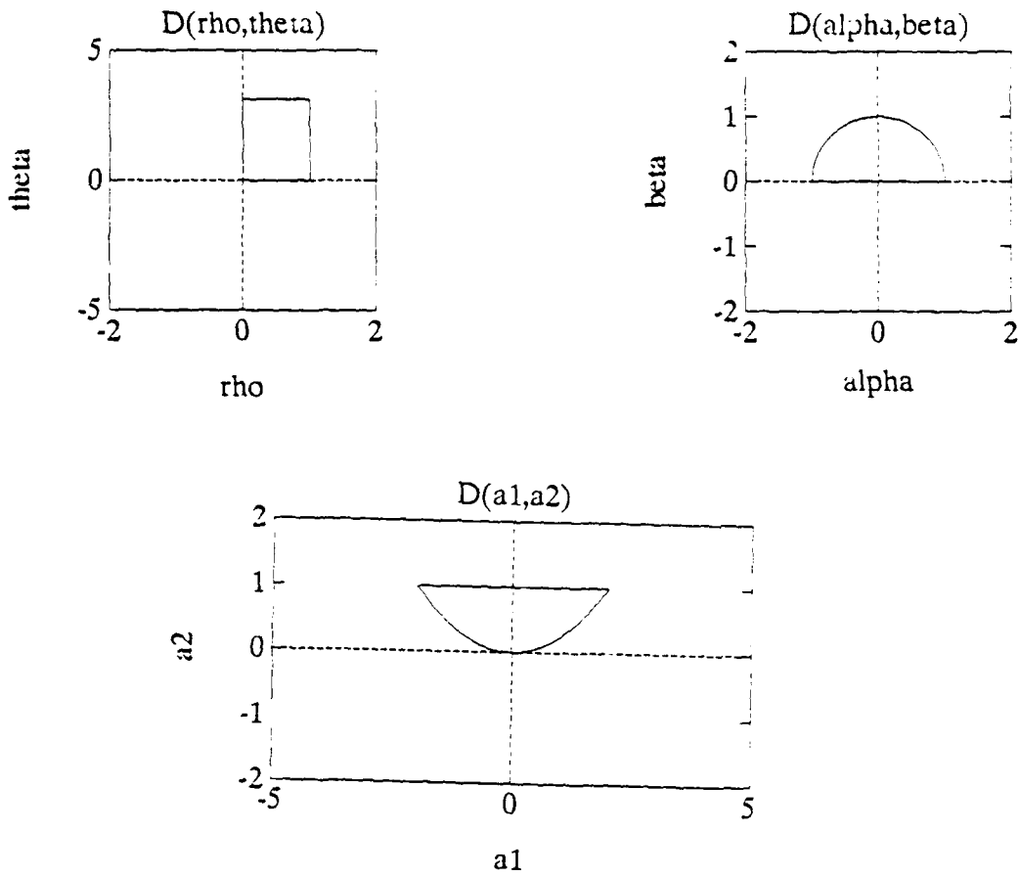


Figure 3.1 Parameter domains for complex quadratic roots.

If one of these sets of parameters is jointly uniformly distributed over its domain, we can compute the densities of the other parameterizations using a theorem in Papoulis [Pap84]. The densities in each row of Table 3.1 and Figure 3.2 are related by transformations of variables, each row corresponding to a uniform joint density in one of the sets of parameters. Note that when one set of parameters is assumed uniform, the densities of the other parameterizations are distinctly nonuniform and in some cases are actually unbounded.

As an example of the computations involved in creating Table 3.1, we present the derivation of the joint density of $(\rho, \theta)$ that corresponds to an assumption of uniformity of

Table 3.1 Density functions for quadratic root parameterizations.

| $(\rho,\theta)$ | $(\alpha,\beta)$ | $(a_1,a_2)$ |
|---|---|---|
| (uniform) | $f_{\alpha,\beta}(\alpha,\beta)=\frac{1}{\pi\sqrt{\alpha^2+\beta^2}}$ | $f_{a_1,a_2}(a_1,a_2)=\frac{1}{2\pi\sqrt{4-a_1^2}\sqrt{a_2}}$ |
| $f_{\rho,\theta}(\rho,\theta)=\frac{1}{\pi}$ | $f_\alpha(\alpha)=\frac{1}{\pi}\ln\left\{\frac{1+\sqrt{1-\alpha^2}}{\lvert\alpha\rvert}\right\}$ | $f_{a_1}(a_1)=\frac{1}{4\pi}\ln\left\{\frac{4\sqrt{4a_2-a_1^2}+8-a_1^2}{a_1^2}\right\}$ |
| $f_\rho(\rho)=1$ | $f_\beta(\beta)=\frac{1}{\pi}\ln\left\{\frac{1+\sqrt{1-\beta^2}}{1-\sqrt{1-\beta^2}}\right\}$ | $f_{a_2}(a_2)=\frac{1}{2\sqrt{a_2}}$ |
| $f_\theta(\theta)=\frac{1}{\pi}$ | | |
| $f_{\rho,\theta}(\rho,\theta)=\frac{2\rho}{\pi}$ | (uniform) | $f_{a_1,a_2}(a_1,a_2)=\frac{1}{\pi\sqrt{4a_2-a_1^2}}$ |
| | $f_{\alpha,\beta}(\alpha,\beta)=\frac{2}{\pi}$ | |
| $f_\rho(\rho)=2\rho$ | $f_\alpha(\alpha)=\frac{2\sqrt{1-\alpha^2}}{\pi}$ | $f_{a_1}(a_1)=\frac{\sqrt{4a_2-a_1^2}}{2\pi}$ |
| $f_\theta(\theta)=\frac{1}{\pi}$ | $f_\beta(\beta)=\frac{4\sqrt{1-\beta^2}}{\pi}$ | $f_{a_2}(a_2)=1$ |
| $f_{\rho,\theta}(\rho,\theta)=\frac{3}{2}\rho^2\sin\theta$ | $f_{\alpha,\beta}(\alpha,\beta)=\frac{3\beta}{2}$ | (uniform) |
| | | $f_{a_1,a_2}(a_1,a_2)=\frac{3}{8}$ |
| $f_\rho(\rho)=3\rho^2$ | $f_\alpha(\alpha)=\frac{3}{2}(1-\alpha^2)$ | $f_{a_1}(a_1)=\frac{3(4-a_1^2)}{32}$ |
| $f_\theta(\theta)=\frac{1}{2}\sin\theta$ | $f_\beta(\beta)=3\beta\sqrt{1-\beta^2}$ | $f_{a_2}(a_2)=\frac{3}{2}\sqrt{a_2}$ |

$(a_1,a_2)$. Assume that $(a_1,a_2)$ are uniformly distributed in $D_{a_1a_2}$:

$$f_{a_1,a_2}(a_1,a_2)=\begin{cases}\frac{3}{8} & \text{for } (a_1,a_2)\in D_{a_1a_2}\\ 0 & \text{otherwise.}\end{cases} \tag{3.9}$$

The functional relationship between $(a_1,a_2)$ and $(\rho,\theta)$ is one-to-one and onto:

$$\begin{array}{ll}\rho=\sqrt{a_2} & a_1=-2\rho\cos\theta\\ \theta=\cos^{-1}(\frac{a_1}{-2\sqrt{a_2}}) & a_2=\rho^2.\end{array} \tag{3.10}$$

The Jacobian matrix for the transformation is

$$\mathbf{J}=\begin{bmatrix}\frac{\partial\rho}{\partial a_1} & \frac{\partial\rho}{\partial a_2}\\ \frac{\partial\theta}{\partial a_1} & \frac{\partial\theta}{\partial a_2}\end{bmatrix}=\begin{bmatrix}0 & \frac{1}{2\sqrt{a_2}}\\ \frac{1}{\sqrt{4a_2-a_1^2}} & \frac{-a_1}{2a_2\sqrt{4a_2-a_1^2}}\end{bmatrix}. \tag{3.11}$$

And the Jacobian of the transformation is the absolute value of the determinant of the Jacobian matrix

$$J=\text{abs}|\mathbf{J}|=\frac{1}{2\sqrt{a_2}\sqrt{4a_2-a_1^2}}=\frac{1}{4\rho^2\sin\theta} \tag{3.12}$$

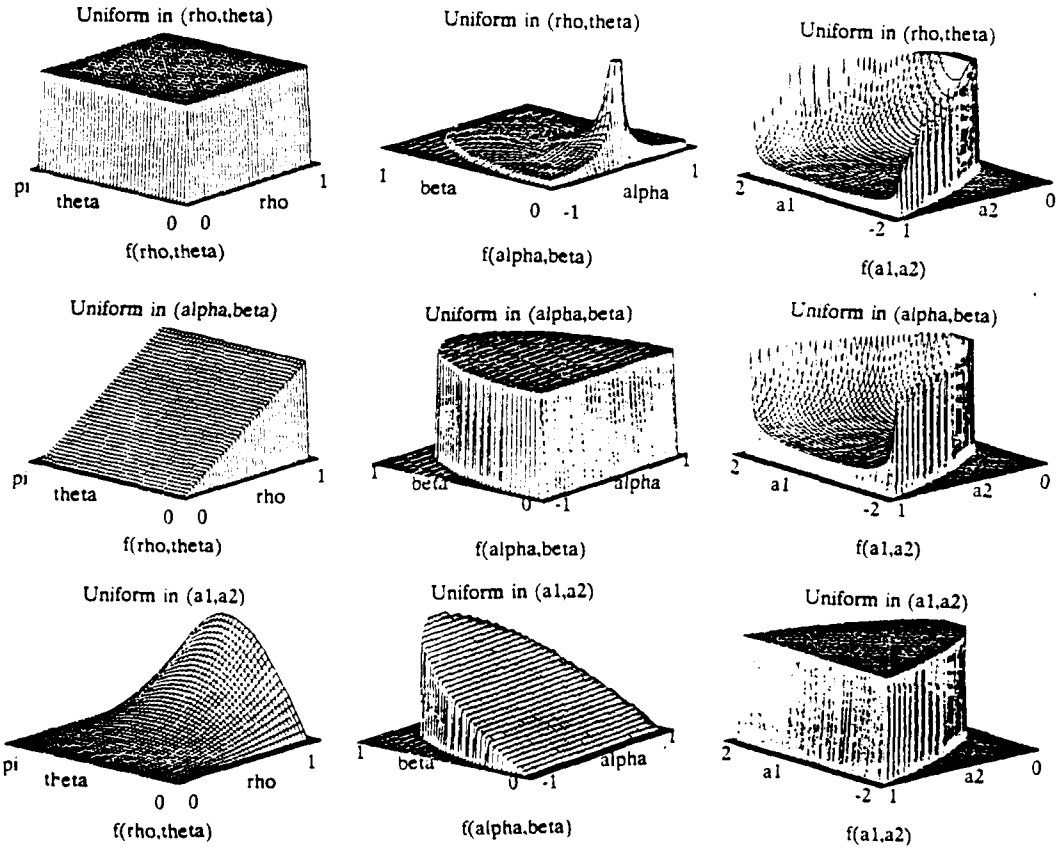


Figure 3.2 Density functions for quadratic root parameterizations.

Hence, the joint density of the radius and angle of $z_1$ is

$$f_{\rho,\theta}(\rho,\theta) = \begin{cases} \dfrac{f_{a_1,a_2}(-2\rho\cos\theta, \rho^2)}{J} = \dfrac{3}{2}\rho^2 \sin\theta & \text{for } (\rho,\theta) \in D_{\rho\theta} \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

Next we compute the marginal densities for $\rho$ and $\theta$ by integration of the joint density.

$$\begin{aligned} f_\rho(\rho) &= \int_0^\pi f_{\rho,\theta}(\rho,\theta)d\theta = 3\rho^2, \qquad \text{for } 0 \le \rho \le 1; \\ f_\theta(\theta) &= \int_0^1 f_{\rho,\theta}(\rho,\theta)d\rho = \frac{1}{2}\sin\theta, \qquad \text{for } 0 \le \theta \le \pi. \end{aligned} \tag{3.14}$$

Of the three choices in Table 3.1 for the density of the radius and angle of $z_1$, this is perhaps the most physically realistic. We say this because this density favors frequencies near the Nyquist rate and damping coefficients near 1.

What does this result imply about frequency estimation? If we know absolutely nothing about prior distributions we might as well use ML. But if we know enough to say that a uniform density is a better description of the polynomial coefficients than of the polar form of the root locations then we should probably use a MAP estimator based on the density just derived for $\rho$ and $\theta$. We will comment on how this might be applied in the section on the KiSS algorithm in Chapter IV.

In the following subsections we derive some ML estimators of subspaces. Their appropriateness to a given problem must be assessed according to the principles just discussed.

### 3.2 ML Signal Subspace ID with No Prior Model

The first ML subspace estimation problem we consider is the case where nothing is known a priori about the signal subspace except its rank $r$. The dimension of the observation space is $n$. Assume that $m$ observations of the signal vector are available ($m \ge r$) and that they obey the linear statistical model without structured noise:

$$\underline{y}_t = \mathbf{H}\underline{\theta}_t + \underline{\nu}_t \qquad (1 \le t \le m). \tag{3.15}$$

$$\text{where} \qquad \underline{y}_t, \underline{\nu}_t \in \mathbb{C}^n, \quad \underline{\theta}_t \in \mathbb{C}^r, \quad \mathbf{H} \in \mathbb{C}^{n\times r}$$

$$\mathrm{Re}(\underline{\nu}_t) : \mathrm{N}(\underline{0}, \sigma^2\mathbf{I}) \qquad \perp \qquad \mathrm{Im}(\underline{\nu}_t) : \mathrm{N}(\underline{0}, \sigma^2\mathbf{I}).$$

Assume further that $\underline{\nu}_i$ is independent of $\underline{\nu}_j$ for $i \neq j$. We desire an estimate of the span of H, the signal subspace. The following result is a slight extension of the work of Scharf [Sch91] in that here we take complex valued data and we assume the noise variance $\sigma^2$ is unknown and must be estimated from the data.

We can parameterize the signal subspace by a set of $n - r$ linearly independent unit vectors $\underline{a}_i$ that are orthogonal to the signal subspace. Define the signal $\underline{x}_t = \mathbf{H}\underline{\theta}_t$. Then

$$\underline{a}_i^H \underline{x}_t = 0; \qquad i = r+1, \ldots, n, \quad t = 1, \ldots, m. \tag{3.16}$$

It is convenient to arrange the vectors $\underline{a}_i$ into a matrix:

$$\mathbf{A} = [\underline{a}_{r+1} \quad \cdots \quad \underline{a}_n] \quad \in \mathbb{C}^{n \times n-r}. \tag{3.17}$$

Equation 3.16 then becomes

$$\mathbf{A}^H \underline{x}_t = \underline{0}, \qquad t = 1, \ldots, m. \tag{3.18}$$

The signal $\underline{x}_t$ is the mean of $\underline{y}_t$, and the density of $\underline{y}_t$ is joint normal:

$$f_{\underline{y}_t}(\underline{y}_t) = \frac{1}{(2\pi\sigma^2)^n} e^{-\frac{1}{2\sigma^2}(\underline{y}_t - \underline{x}_t)^H (\underline{y}_t - \underline{x}_t)}. \tag{3.19}$$

Note that the normalization reflects $2n$ dimensions because $\underline{y}_t$ is complex.

The log-likelihood function, given $\underline{y}_1, \ldots, \underline{y}_m$, is

$$L(\underline{x}_1, \ldots, \underline{x}_m, \sigma^2) = -mn \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{m} (\underline{y}_t - \underline{x}_t)^H (\underline{y}_t - \underline{x}_t). \tag{3.20}$$

We need to maximize the log-likelihood function under the orthogonality constraints of Equation 3.18. The Lagrangian to minimize the negative log-likelihood under these constraints is

$$\mathcal{L} = mn \ln(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{t=1}^{m} (\underline{y}_t - \underline{x}_t)^H (\underline{y}_t - \underline{x}_t) + \sum_{t=1}^{m} \underline{\lambda}_t^H \mathbf{A}^H \underline{x}_t. \tag{3.21}$$

where we have defined the Lagrange multipliers

$$\underline{\lambda}_t = \begin{bmatrix} \lambda_{t(r+1)} \\ \vdots \\ \lambda_{tn} \end{bmatrix} \qquad \in \mathbb{C}^{n-r}. \tag{3.22}$$

We would like to optimize $\mathcal{L}$ with respect to $\underline{x}_t$. The usual approach is to set the gradient of $\mathcal{L}$ with respect to $\underline{x}_t$ equal to zero. However, the gradient does not exist in this case because $\underline{x}_t$ is complex and the complex conjugate function is not differentiable in the sense of complex variables. All we really need, though, is that the gradient of $\mathcal{L}$ with respect to the real part of $\underline{x}_t$, and the gradient of $\mathcal{L}$ with respect to the imaginary part of $\underline{x}_T$ both be zero. We build a psuedo gradient by taking the gradient with respect to the real part plus $j$ times the gradient with respect to the imaginary part, and call it the gradient. If we assume for the moment that $\underline{a}_{r+1} \cdots \underline{a}_n$ and $\sigma^2$ are known, then the gradient (pseudo gradient) of $\mathcal{L}$ with respect to $\underline{x}_t$ is

$$\nabla_{\underline{x}_t} \mathcal{L} = -\frac{1}{\sigma^2}(\underline{y}_t - \underline{x}_t) + \mathbf{A}\underline{\lambda}_t. \tag{3.23}$$

To make the gradient equal to zero we choose

$$\underline{x}_t = \underline{y}_t - \sigma^2 \mathbf{A}\underline{\lambda}_t. \tag{3.24}$$

The constraints are imposed by writing

$$\mathbf{A}^H(\underline{y}_t - \sigma^2 \mathbf{A}\underline{\lambda}_t) = \underline{0} \tag{3.25}$$

and solving for $\underline{\lambda}_t$:

$$\underline{\lambda}_t = \frac{1}{\sigma^2}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\underline{y}_t. \tag{3.26}$$

The corresponding solution for $\underline{x}_t$ is the ML estimator

$$\begin{aligned} \hat{\underline{x}}_t &= (\mathbf{I} - \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H)\underline{y}_t \\ &= (\mathbf{I} - \mathbf{P}_\mathbf{A})\underline{y}_t. \end{aligned} \tag{3.27}$$

The resulting maximum value of the log-likelihood function is

$$\begin{aligned} L(\mathbf{A}, \sigma^2) &= -mn\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{m} \underline{y}_t^H \mathbf{P}_\mathbf{A} \underline{y}_t \\ &= -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\mathrm{Tr}(\mathbf{P}_\mathbf{A}\mathbf{R}), \end{aligned} \tag{3.28}$$

where $\mathbf{R}$ is the sample correlation matrix:

$$\mathbf{R} = \frac{1}{m}\sum_{t=1}^{m} \underline{y}_t \underline{y}_t^H \quad \in \mathbb{C}^{n\times n}. \tag{3.29}$$

We have compressed the likelihood by maximizing with respect to $\underline{x}_t$ so that it is no longer a function of $\underline{\theta}_t$. The next step is to maximize with respect to $\mathbf{A}$. Let the sample covariance matrix $\mathbf{R}$ have the orthogonal decomposition

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^H, \tag{3.30}$$

where $\mathbf{U}$ is unitary and $\mathbf{\Lambda}^2$ is diagonal and ordered:

$$\begin{aligned} &\mathbf{U}^H\mathbf{U} = \mathbf{I} \\ &\mathbf{U} = [\underline{u}_1 \cdots \underline{u}_{r+1} \cdots \underline{u}_n] \quad \in \mathbb{C}^{n\times n} \\ &\mathbf{\Lambda}^2 = \mathrm{diag}(\lambda_1^2 \ldots \lambda_{r+1}^2 \ldots \lambda_n^2) \quad \in \mathbb{R}^{n\times n} \\ &\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq \lambda_n^2. \end{aligned} \tag{3.31}$$

Then the log-likelihood is bounded by

$$L(\mathbf{A}, \sigma^2) = -\frac{mn}{2}\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\mathrm{Tr}(\mathbf{P}_\mathbf{A}\mathbf{R}) \leq -\frac{mn}{2}\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\sum_{i=r+1}^{n}\lambda_i^2 \tag{3.32}$$

for any projection $\mathbf{P}_\mathbf{A}$ of rank (n-r). This bound is achieved when $\mathbf{P}_\mathbf{A}$ is a projection onto the subspace $\langle \mathbf{U}_2 \rangle$ spanned by the $n-r$ least dominant eigenvectors of $\mathbf{R}$:

$$\begin{aligned} &\hat{\mathbf{A}} = \mathbf{U}_2 = [\underline{u}_{r+1} \quad \cdots \quad \underline{u}_n] \qquad \in \mathbb{C}^{n\times n-r}. \\ &\mathbf{P}_\mathbf{A} = \mathbf{U}_2\mathbf{U}_2^H = \sum_{i=r+1}^{n} \underline{u}_i\underline{u}_i^H, \end{aligned} \tag{3.33}$$

Now since $\langle \mathbf{H} \rangle$ is the space orthogonal to $\mathbf{A}$, we have the maximum likelihood estimate of the signal subspace:

$$\left\langle \hat{\mathbf{H}} \right\rangle = \langle \mathbf{U}_1 \rangle = \langle \underline{u}_1, \ldots, \underline{u}_r \rangle . \tag{3.34}$$

This ML estimator fits the conditions given earlier for equality with the least squares estimator. It was derived under the assumption of additive stationary zero-mean white Gaussian noise. The corresponding least squares problem is equivalent to finding the rank $r$ matrix closest in Frobenius norm to the given data matrix $\mathbf{Y} = [\underline{y}_1 \ldots \underline{y}_m] \in \mathbb{C}^{n\times m}$. This problem was solved by Eckart and Young [EcY36]. The solution is to form the signal subspace estimate by taking the $r$ dominant left singular vectors of $\mathbf{Y}$. This solution is identical to the ML estimator, which

takes the $r$ dominant eigenvectors of $\mathbf{R} = \mathbf{Y}\mathbf{Y}^H$, because the eigenvectors and eigenvalues of $\mathbf{R}$ are the same as the left singular vectors and squares of the singular values of $\mathbf{Y}$.

The last step is to maximize the log-likelihood with respect to $\sigma^2$ and obtain an estimate of the noise variance. The maximum value of log-likelihood from the previous step is

$$L(\sigma^2) = -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\sum_{i=r+1}^{n}\lambda_i^2. \tag{3.35}$$

Differentiating with respect to $\sigma^2$ gives

$$\frac{\partial L}{\partial \sigma^2} = -mn\frac{1}{2\pi\sigma^2}(2\pi) + \frac{m}{2\sigma^4}\sum_{i=r+1}^{n}\lambda_i^2. \tag{3.36}$$

Setting the derivative to zero and solving for $\sigma^2$ produces the ML estimator of variance

$$\hat{\sigma}^2 = \frac{1}{2n}\sum_{i=r+1}^{n}\lambda_i^2. \tag{3.37}$$

This is almost what one would expect for a variance estimator, except for the normalization by $n$. Since we are summing $n - r$ singular values it would seem natural to normalize by $n - r$. There is nothing to stop us from changing the normalization, and the resulting estimator might be better, but it would not be the ML estimator.

### 3.3 Constrained ML Signal Subspace ID

We consider now a variation of the Maximum Likelihood signal subspace identification problem treated in the preceding section. In this section we impose the constraint that the identified signal subspace be contained in a given higher rank subspace $\langle \mathbf{V} \rangle$. The given subspace $\langle \mathbf{V} \rangle$ could, for example, be a users allotted portion of a Code Division Multiple Access (CDMA) information channel. Any signal of significance to the user would lie in $\langle \mathbf{V} \rangle$, but the user may be interested in identifying a lower rank signal subspace.

While the other constrained identification problems are dealt with in Chapter IV, the simplicity of this constraint and the appearance of the SVD in the solution tie this problem to the unconstrained identification problems of the present chapter.

Assume as before that $m$ observations of the signal $n$-vector are available ($m \geq r$)

and that they obey the linear statistical model without structured noise as given in Equation 3.15. We desire an estimate of the signal subspace spanned by $\mathbf{H}$, under the constraint that $\langle \mathbf{H} \rangle \subset \langle \mathbf{V} \rangle$.

As before we can parameterize the signal subspace by a set of $n-r$ linearly independent unit vectors $\underline{a}_i$ that are orthogonal to the signal subspace, and collect the vectors $\underline{a}_i$ into a matrix $\mathbf{A}$:

$$\mathbf{A}^H \underline{x}_t = \underline{0}, \qquad t = 1, \ldots, m. \tag{3.38}$$

Let $\mathbf{B}$ be a matrix that spans the orthogonal complement of $\langle \mathbf{V} \rangle$. Now since $\langle \mathbf{H} \rangle \subset \langle \mathbf{V} \rangle$ it follows that $\langle \mathbf{B} \rangle \subset \langle \mathbf{A} \rangle$. Our constraint therefore serves to predetermine a portion of the subspace $\langle \mathbf{A} \rangle$, so we can represent $\mathbf{A}$ as the concatenation of a known portion $\mathbf{B}$ and an unknown portion $\tilde{\mathbf{A}}$ which we may take to be orthogonal to $\mathbf{B}$:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{B} & \tilde{\mathbf{A}} \end{bmatrix}, \\ \mathbf{P}_{\mathbf{A}} &= \mathbf{P}_{\mathbf{B}} + \mathbf{P}_{\tilde{\mathbf{A}}}. \end{aligned} \tag{3.39}$$

The Lagrangian for constrained minimization of the negative log-likelihood function is the same as in Equation 3.21,

$$\mathcal{L} = mn \ln(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{t=1}^{m} (\underline{y}_t - \underline{x}_t)^H (\underline{y}_t - \underline{x}_t) + \sum_{t=1}^{m} \underline{\lambda}_t^H \mathbf{A}^H \underline{x}_t, \tag{3.40}$$

and the development proceeds unchanged through Equation 3.28:

$$\begin{aligned} L(\mathbf{A}, \sigma^2) &= -mn \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{m} \underline{y}_t^H \mathbf{P}_{\mathbf{A}} \underline{y}_t \\ &= -mn \ln(2\pi\sigma^2) - \frac{m}{2\sigma^2} \mathrm{Tr}(\mathbf{P}_{\mathbf{A}} \mathbf{R}). \end{aligned} \tag{3.41}$$

The next step is to maximize with respect to the unknown portion of $\mathbf{A}$, and here we begin to differ from the development in the preceding section. The compressed negative log-likelihood function in Equation 3.41 may be resolved into a fixed portion for $\mathbf{B}$ and a variable portion for $\tilde{\mathbf{A}}$:

$$\begin{aligned} L(\tilde{\mathbf{A}}, \sigma^2) &= -mn \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{m} (\underline{y}_t^H \mathbf{P}_{\mathbf{B}} \underline{y}_t + \underline{y}_t^H \mathbf{P}_{\tilde{\mathbf{A}}} \underline{y}_t) \\ &= -mn \ln(2\pi\sigma^2) - \frac{m}{2\sigma^2} \mathrm{Tr}(\mathbf{P}_{\mathbf{B}} \mathbf{R} + \mathbf{P}_{\tilde{\mathbf{A}}} \mathbf{R}). \end{aligned} \tag{3.42}$$

So the problem reduces to finding $\tilde{\mathbf{A}}$ orthogonal to $\mathbf{B}$, or equivalently $\tilde{\mathbf{A}}$ in $\langle \mathbf{V} \rangle$, to minimize

$$\begin{aligned} L(\tilde{\mathbf{A}}, \sigma^2) &= -mn \ln(2\pi\sigma^2) - \frac{m}{2\sigma^2} \mathrm{Tr}(\mathbf{P}_{\tilde{\mathbf{A}}} \mathbf{R}) \\ &= -mn \ln(2\pi\sigma^2) - \frac{m}{2\sigma^2} \mathrm{Tr}(\mathbf{P}_{\tilde{\mathbf{A}}} \mathbf{R} \mathbf{P}_{\tilde{\mathbf{A}}}). \end{aligned} \tag{3.43}$$

But since $\left\langle \tilde{\mathbf{A}} \right\rangle \subset \langle \mathbf{V} \rangle$ we know that

$$\mathbf{P}_{\tilde{\mathbf{A}}} \mathbf{P}_{\mathbf{V}} = \mathbf{P}_{\tilde{\mathbf{A}}}, \tag{3.44}$$

so we can replace $\mathbf{P}_{\tilde{\mathbf{A}}}$ in Equation 3.43 by $\mathbf{P}_{\tilde{\mathbf{A}}} \mathbf{P}_{\mathbf{V}}$ where $\mathbf{P}_{\mathbf{V}}$ is the orthogonal projection onto $\langle \mathbf{V} \rangle$. Now let

$$\tilde{\mathbf{R}} = \mathbf{P}_{\mathbf{V}} \mathbf{R} \mathbf{P}_{\mathbf{V}}. \tag{3.45}$$

With this definition of $\tilde{\mathbf{R}}$ the log-likelihood function becomes

$$L(\tilde{\mathbf{A}}, \sigma^2) = -mn \ln(2\pi\sigma^2) - \frac{m}{2\sigma^2} \mathrm{Tr}(\mathbf{P}_{\tilde{\mathbf{A}}} \tilde{\mathbf{R}} \mathbf{P}_{\tilde{\mathbf{A}}}). \tag{3.46}$$

The projected sample correlation matrix $\tilde{\mathbf{R}}$ will now play the same role in this constrained subspace identification problem as $\mathbf{R}$ played in the unconstrained problem.

Let the projected sample covariance matrix $\tilde{\mathbf{R}}$ have the orthogonal decomposition

$$\tilde{\mathbf{R}} = \mathbf{U} \mathbf{\Lambda}^2 \mathbf{U}^H, \tag{3.47}$$

where $\mathbf{U}$ is unitary and $\mathbf{\Lambda}^2$ is diagonal and ordered:

$$\begin{aligned} &\mathbf{U}^H \mathbf{U} = \mathbf{I} \\ &\mathbf{U} = [\underline{u}_1 \cdots \underline{u}_{r+1} \cdots \underline{u}_n] \quad \in \mathbb{C}^{n \times n} \\ &\mathbf{\Lambda}^2 = \mathrm{diag}(\lambda_1^2 \ldots \lambda_{r+1}^2 \ldots \lambda_n^2) \quad \in \mathbf{R}^{n \times n} \\ &\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq \lambda_n^2. \end{aligned} \tag{3.48}$$

Since $\tilde{\mathbf{R}}$ has been projected onto $\langle \mathbf{V} \rangle$ it will have a set of zero eigenvalues $\lambda_i$ corresponding to eigenvectors $\underline{u}_i$ that span $\langle \mathbf{B} \rangle$, the orthogonal complement of $\langle \mathbf{V} \rangle$.

As in the unconstrained problem of the last section, the negative log-likelihood is bounded by an expression involving the sum of the largest eigenvalues of $\tilde{\mathbf{R}}$ and the bound is

achieved when $\mathbf{P_A}$ is a projection onto the subspace $\langle \mathbf{U}_2 \rangle$ spanned by the $n-r$ least dominant eigenvectors of $\tilde{\mathbf{R}}$. The projection $\mathbf{P_A}$ chosen in this way will automatically contain $\langle \mathbf{B} \rangle$ in its range because the eigenvalues corresponding to $\langle \mathbf{B} \rangle$ are zero.

In summary, the constrained ML estimate of the signal subspace $\mathbf{H}$ is the space orthogonal to $\mathbf{A}$, which is the space spanned by the $r$ most dominant eigenvectors of the projected sample correlation matrix $\tilde{\mathbf{R}} = \mathbf{P_V R P_V} = \frac{1}{m}\sum_{t=1}^{m}(\mathbf{P_V}\underline{y}_t)(\mathbf{P_V}\underline{y}_t)^H$. A simple way to think about the solution is to project all of the received data $\underline{y}_t$ onto the given constraint subspace $\langle \mathbf{V} \rangle$, and then proceed exactly as in the unconstrained problem by forming the sample correlation matrix, taking its SVD, and choosing the space spanned by the $r$ dominant eigenvectors as the signal subspace. We can go on to estimate $\sigma^2$ exactly as in the unconstrained problem once the signal subspace has been determined.

### 3.4 ML Signal and Noise Subspace ID with No Prior Signal Model

Consider the problem of simultaneously identifying the signal subspace and the structured noise subspace. As before we assume that nothing is known a priori about the signal subspace except its rank $r$. However, we cannot make a similar assumption about the structured noise subspace because there must be some distinguishing feature that allows us to discriminate between signal and structured noise. For the sake of this development, we assume that the structured noise is impulsive, having $q$ nonzero elements in unknown positions. It is not important that $q$ be known a priori.

As before the dimension of the observation space is $n$. Assume that $m$ observations of the data vector are available ($m \geq r+q$) and that they obey the structured noise model:

$$\underline{y}_t = \mathbf{H}\underline{\theta}_t + \mathbf{S}\underline{\alpha}_t + \underline{\nu}_t \qquad (1 \leq t \leq m), \tag{3.49}$$

$$\text{where} \quad \underline{y}_t, \underline{\nu}_t \in \mathbb{C}^n, \quad \underline{\theta}_t \in \mathbb{C}^r, \quad \mathbf{H} \in \mathbb{C}^{n\times r},$$

$$\underline{\alpha}_t \in \mathbb{C}^q, \quad \mathbf{S} \in \{0,1\}^{n\times q},$$

$$\mathrm{Re}(\underline{\nu}_t) : \mathrm{N}(\underline{0}, \sigma^2\mathbf{I}) \quad \perp\!\!\!\perp \quad \mathrm{Im}(\underline{\nu}_t) : \mathrm{N}(\underline{0}, \sigma^2\mathbf{I}).$$

Assume further that $\underline{\nu}_i$ is independent of $\underline{\nu}_j$ for $i \neq j$. We desire an estimate of the signal subspace $\langle \mathbf{H} \rangle$, and the structured noise subspace $\langle \mathbf{S} \rangle$. Note the limitation that $\mathbf{S}$ is the same for each of the $m$ observations of the data vector. This would be the situation, for example, in a sensor array in which some of the sensors were subject to high noise levels or had failed.

Since the number of selection matrices $\mathbf{S}$ of rank $q$ or less is finite, we can consider the merits of each one in turn and choose the best one. Assume for now that a candidate $\mathbf{S}$ has been chosen. Then the derivation of the maximum likelihood estimator of $\mathbf{H}$ is an extension of the preceding section. We proceed in a parallel fashion.

We can parameterize the signal subspace by a set of $n - r$ linearly independent unit vectors $\underline{a}_i$ that are orthogonal to the signal subspace. Define the signal $\underline{x}_t = \mathbf{H}\underline{\theta}_t$, and the structured noise $\underline{b}_t = \mathbf{S}\underline{\phi}_t$. Then

$$\underline{a}_i^H \underline{x}_t = 0; \qquad i = r+1, \ldots, n, \quad t = 1, \ldots, m. \tag{3.50}$$

It is convenient to arrange the vectors $\underline{a}_i$ into a matrix:

$$\mathbf{A} = [\underline{a}_{r+1} \quad \cdots \quad \underline{a}_n] \quad \in \mathbb{C}^{n \times n-r}. \tag{3.51}$$

Equation 3.50 then becomes

$$\mathbf{A}^H \underline{x}_t = \underline{0}, \qquad t = 1, \ldots, m. \tag{3.52}$$

The mean of $\underline{y}_t$ is the signal plus the structured noise, $\underline{x}_t + \underline{b}_t$, and the density of $\underline{y}_t$ is joint normal:

$$f_{\underline{y}_t}(\underline{y}_t) = \frac{1}{(2\pi\sigma^2)^n} e^{-\frac{1}{2\sigma^2}(\underline{y}_t - \underline{x}_t - \underline{b}_t)^H (\underline{y}_t - \underline{x}_t - \underline{b}_t)}. \tag{3.53}$$

The log-likelihood function, given $\underline{y}_1, \ldots, \underline{y}_m$, is

$$L(\underline{x}_1, \ldots, \underline{x}_m, \underline{b}_1, \ldots, \underline{b}_m, \sigma^2) = -mn \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{m} (\underline{y}_t - \underline{x}_t - \underline{b}_t)^H (\underline{y}_t - \underline{x}_t - \underline{b}_t). \tag{3.54}$$

We need to maximize the log-likelihood function under the orthogonality constraints of Equation 3.52. The Lagrangian to minimize the negative log-likelihood under these constraints

is

$$\mathcal{L} = mn\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{t=1}^{m}(\underline{y}_t - \underline{x}_t - \underline{b}_t)^H(\underline{y}_t - \underline{x}_t - \underline{b}_t) + \sum_{t=1}^{m}\underline{\lambda}_t^H\mathbf{A}^H\underline{x}_t. \tag{3.55}$$

where we have defined the Lagrange multipliers

$$\underline{\lambda}_t = \begin{bmatrix} \lambda_{t(r+1)} \\ \vdots \\ \lambda_{tn} \end{bmatrix} \quad \in \mathbb{C}^{n-r}. \tag{3.56}$$

If we assume for the moment that $\underline{a}_{r+1}\cdots\underline{a}_n$, $\underline{b}_1\ldots\underline{b}_m$ and $\sigma^2$ are known, then the gradient of $\mathcal{L}$ with respect to $\underline{x}_t$ is

$$\nabla_{\underline{x}_t}\mathcal{L} = -\frac{1}{\sigma^2}(\underline{y}_t - \underline{x}_t - \underline{b}_t) + \mathbf{A}\underline{\lambda}_t. \tag{3.57}$$

To make the gradient equal to zero we choose

$$\underline{x}_t = \underline{y}_t - \underline{b}_t - \sigma^2\mathbf{A}\underline{\lambda}_t. \tag{3.58}$$

The constraints are imposed by writing

$$\mathbf{A}^H(\underline{y}_t - \underline{b}_t - \sigma^2\mathbf{A}\underline{\lambda}_t) = \underline{0} \tag{3.59}$$

and solving for $\underline{\lambda}_t$:

$$\underline{\lambda}_t = \frac{1}{\sigma^2}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H(\underline{y}_t - \underline{b}_t). \tag{3.60}$$

The corresponding solution for $\underline{x}_t$ is the ML estimator

$$\begin{aligned} \hat{\underline{x}}_t &= (\mathbf{I} - \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H)(\underline{y}_t - \underline{b}_t) \\ &= (\mathbf{I} - \mathbf{P}_\mathbf{A})(\underline{y}_t - \underline{b}_t). \end{aligned} \tag{3.61}$$

The resulting maximum value of the log-likelihood function is

$$\begin{aligned} L(\mathbf{A}, \underline{\varrho}_1, \ldots, \underline{\varrho}_m, \sigma^2) &= -mn\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{m}(\underline{y}_t - \underline{b}_t)^H\mathbf{P}_\mathbf{A}(\underline{y}_t - \underline{b}_t) \\ &= -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\mathrm{Tr}(\mathbf{P}_\mathbf{A}\tilde{\mathbf{R}}) \end{aligned} \tag{3.62}$$

where $\tilde{\mathbf{R}}$ is the sample correlation matrix with the structured noise subtracted out:

$$\tilde{\mathbf{R}} = \frac{1}{m}\sum_{t=1}^{m}(\underline{y}_t - \underline{b}_t)(\underline{y}_t - \underline{b}_t)^H \quad \in \mathbb{C}^{n\times n}. \tag{3.63}$$

We have compressed the likelihood by maximizing with respect to $\underline{x}_t$ so that it is no longer a function of $\underline{\theta}_t$. The next step is to maximize with respect to $\mathbf{A}$. Let $\tilde{\mathbf{R}}$ have the orthogonal decomposition

$$\tilde{\mathbf{R}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}^2\tilde{\mathbf{U}}^H, \tag{3.64}$$

where $\tilde{\mathbf{U}}$ is unitary and $\tilde{\mathbf{\Lambda}}^2$ is diagonal and ordered:

$$\begin{aligned} &\tilde{\mathbf{U}}^H\tilde{\mathbf{U}} = \mathbf{I} \\ &\tilde{\mathbf{U}} = [\underline{\tilde{u}}_1 \cdots \underline{\tilde{u}}_{r+1} \cdots \underline{\tilde{u}}_n] \quad \in \mathbb{C}^{n\times n} \\ &\tilde{\mathbf{\Lambda}}^2 = \mathrm{diag}(\tilde{\lambda}_1^2 \ldots \tilde{\lambda}_{r+1}^2 \ldots \tilde{\lambda}_n^2) \quad \in \mathbf{R}^{n\times n} \\ &\tilde{\lambda}_1^2 \geq \tilde{\lambda}_2^2 \geq \cdots \geq \tilde{\lambda}_n^2. \end{aligned} \tag{3.65}$$

Then the log-likelihood is bounded by

$$\begin{aligned} L(\mathbf{A}, \underline{\phi}_1, \ldots, \underline{\phi}_m, \sigma^2) &= -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\mathrm{Tr}(\mathbf{P}_\mathbf{A}\tilde{\mathbf{R}}) \\ &\leq -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\sum_{i=r+1}^{n}\tilde{\lambda}_i^2 \end{aligned} \tag{3.66}$$

for any projection $\mathbf{P}_\mathbf{A}$ of rank (n-r). This bound is achieved when $\mathbf{P}_\mathbf{A}$ is a projection onto the subspace $\left\langle \tilde{\mathbf{U}}_2 \right\rangle$ spanned by the $n-r$ least dominant eigenvectors of $\tilde{\mathbf{R}}$:

$$\begin{aligned} &\hat{\mathbf{A}} = \tilde{\mathbf{U}}_2 = [\underline{\tilde{u}}_{r+1} \quad \cdots \quad \underline{\tilde{u}}_n] \qquad \in \mathbb{C}^{n\times n-r}, \\ &\mathbf{P}_\mathbf{A} = \tilde{\mathbf{U}}_2\tilde{\mathbf{U}}_2^H = \sum_{i=r+1}^{n}\underline{\tilde{u}}_i\underline{\tilde{u}}_i^H. \end{aligned} \tag{3.67}$$

Next we minimize the negative log-likelihood with respect to the structured noise $\underline{b}_t = \mathbf{S}\underline{\phi}_t$. This is most easily accomplished by decomposing the correlation matrix $\tilde{\mathbf{R}}$ onto the orthogonal complements $\langle \mathbf{S} \rangle$ and $\langle \mathbf{S} \rangle^\perp$:

$$\begin{aligned} \tilde{\mathbf{R}} &= \mathbf{P}_\mathbf{S}\tilde{\mathbf{R}}\mathbf{P}_\mathbf{S} + \mathbf{P}_{\mathbf{S}^\perp}\tilde{\mathbf{R}}\mathbf{P}_{\mathbf{S}^\perp} + \mathbf{P}_\mathbf{S}\tilde{\mathbf{R}}\mathbf{P}_{\mathbf{S}^\perp} + \mathbf{P}_{\mathbf{S}^\perp}\tilde{\mathbf{R}}\mathbf{P}_\mathbf{S} \\ &= \frac{1}{m}\sum_{t=1}^{m}\left[(\mathbf{P}_\mathbf{S}\underline{y}_t - \mathbf{S}\underline{\phi}_t)(\mathbf{P}_\mathbf{S}\underline{y}_t - \mathbf{S}\underline{\phi}_t)^H + \mathbf{P}_{\mathbf{S}^\perp}\underline{y}_t\underline{y}_t^H\mathbf{P}_{\mathbf{S}^\perp}\right] \\ &\quad + \mathbf{P}_\mathbf{S}\tilde{\mathbf{R}}\mathbf{P}_{\mathbf{S}^\perp} + \mathbf{P}_{\mathbf{S}^\perp}\tilde{\mathbf{R}}\mathbf{P}_\mathbf{S}. \end{aligned} \tag{3.68}$$

The first two terms of the decomposition are nonnegative definite, and the second term is constant with respect to $\underline{\phi}_t$. The last two terms, the cross terms, are nilpotent. The best choice of $\underline{\phi}_t$ makes all terms but the second term zero, which will minimize the sum of the trailing

singular values of $\tilde{\mathbf{R}}$ in Equation 3.66. To make the first term and the cross terms zero, we choose

$$\underline{\phi}_t = (\mathbf{S}^H\mathbf{S})^{-1}\mathbf{S}^H\underline{y}_t. \tag{3.69}$$

This implies that

$$\underline{b}_t = \mathbf{P}_\mathbf{S}\underline{y}_t, \tag{3.70}$$

and

$$\begin{aligned}\tilde{\mathbf{R}} &= (\mathbf{I}-\mathbf{P}_\mathbf{S})\left[\frac{1}{m}\sum_{t=1}^{m}\underline{y}_t\underline{y}_t^H\right](\mathbf{I}-\mathbf{P}_\mathbf{S}) \\ &= \mathbf{P}_{\mathbf{S}^\perp}\mathbf{R}\mathbf{P}_{\mathbf{S}^\perp}.\end{aligned} \tag{3.71}$$

The last step is to maximize the log-likelihood with respect to $\sigma^2$ and obtain an estimate of the noise variance. The maximum value of log-likelihood from the previous step is

$$\begin{aligned}L(\sigma^2) &= -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\sum_{i=r+1}^{n}\tilde{\lambda}_i^2 \\ &= -mn\ln(2\pi\sigma^2) - \frac{m}{2\sigma^2}\mathrm{Tr}\left(\mathbf{P}_\mathbf{A}\mathbf{P}_{\mathbf{S}^\perp}\mathbf{R}\mathbf{P}_{\mathbf{S}^\perp}\mathbf{P}_\mathbf{A}\right).\end{aligned} \tag{3.72}$$

Differentiating with respect to $\sigma^2$ gives

$$\frac{\partial L}{\partial\sigma^2} = -mn\frac{1}{2\pi\sigma^2}(2\pi) + \frac{m}{2\sigma^4}\sum_{i=r+1}^{n}\tilde{\lambda}_i^2. \tag{3.73}$$

Setting the derivative to zero and solving for $\sigma^2$ produces the ML estimator of variance

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{2n}\sum_{i=r+1}^{n}\tilde{\lambda}_i^2 \\ &= \frac{1}{2n}\mathrm{Tr}\left(\mathbf{P}_\mathbf{A}\mathbf{P}_{\mathbf{S}^\perp}\mathbf{R}\mathbf{P}_{\mathbf{S}^\perp}\mathbf{P}_\mathbf{A}\right) \\ &= \frac{1}{2n}\mathrm{Tr}\left(\mathbf{P}_{(\mathbf{H},\mathbf{S})^\perp}\mathbf{R}\mathbf{P}_{(\mathbf{H},\mathbf{S})^\perp}\right).\end{aligned} \tag{3.74}$$

The algorithm to estimate the signal and structured noise subspaces and the noise variance may be summarized as follows. First form the sample correlation matrix from the received data vectors. Then, for each possible structured noise matrix $\mathbf{S}$, perform the remaining steps and choose the $\mathbf{S}$ for which the final likelihood function is maximized. If the rank of $\mathbf{S}$ is not known, it will be necessary to include an order penalty term as an order selection rule. For each candidate $\mathbf{S}$, form the projected correlation matrix $\tilde{\mathbf{R}} = \mathbf{P}_{\mathbf{S}^\perp}\mathbf{R}\mathbf{P}_{\mathbf{S}^\perp}$ and find its orthogonal

decomposition. The $r$ dominant singular vectors of $\tilde{\mathbf{R}}$ span the candidate signal subspace. The sum of the trailing $n - r$ singular values is the negative log-likelihood corresponding to that choice of structured noise subspace **S**.

It is noteworthy that this algorithm will always produce an estimate of the signal subspace that is orthogonal to the estimate of the structured noise subspace. This is not ideal, but can be attributed to the low level of prior knowledge about the signal and structured noise.

### 3.5 Total Least Squares for Signal Subspace Updates

In this section we describe how Total Least Squares provides a natural way to update signal subspace estimates based on new data. The idea is due to Steve Voran [unpublished notes]. In the next section we extend the idea to the structured noise model.

In the theory of Total Least Squares, the prior model

$$\begin{aligned} \underline{y} &= \underline{x} + \underline{\nu} \\ \underline{x} &= \mathbf{H}\underline{\theta}, \qquad \mathbf{H} \in \mathbb{C}^{n \times m} \end{aligned} \tag{3.75}$$

is replaced by the posterior model

$$\begin{aligned} \underline{y} &= \mathbf{P}_1 \underline{y} + (\mathbf{I} - \mathbf{P}_1)\underline{y} \\ &= \hat{\underline{x}}_{TLS} + \hat{\underline{\nu}}_{TLS} \\ \hat{\underline{x}}_{TLS} &= \hat{\mathbf{H}}\hat{\underline{\theta}}_{TLS}. \end{aligned} \tag{3.76}$$

The projection $\mathbf{P}_1$ is chosen to minimize the *total* of the sum of the squares of the elements of $\hat{\underline{\nu}}_{TLS} = (\mathbf{I} - \mathbf{P}_1)\underline{y}$ *plus* the sum of the the squares of the elements of $\Delta_{\mathbf{H}} = (\mathbf{I} - \mathbf{P}_1)\mathbf{H}$ (thus the term Total Least Squares). In the posterior model, $\hat{\mathbf{H}}$ is the estimated (corrected or updated) signal model, $\hat{\underline{x}}_{TLS}$ is the estimated signal component and $\hat{\underline{\nu}}_{TLS}$ is the estimated noise component:

$$\begin{aligned} \hat{\mathbf{H}} &= \mathbf{P}_1 \mathbf{H} \\ \hat{\underline{x}}_{TLS} &= \mathbf{P}_1 \underline{y} = \mathbf{P}_1 \underline{x} + \mathbf{P}_1 \underline{\nu} \\ \hat{\underline{\nu}}_{TLS} &= (\mathbf{I} - \mathbf{P}_1)\underline{y} = (\mathbf{I} - \mathbf{P}_1)\underline{x} + (\mathbf{I} - \mathbf{P}_1)\underline{\nu}. \end{aligned} \tag{3.77}$$

The rightmost equalities differ from the LS case because the projection $\mathbf{P}_1$ is not matched to the subspace $\langle \mathbf{H} \rangle$, but is instead matched to $\left\langle \hat{\mathbf{H}} \right\rangle$, a kind of compromise between $\mathbf{H}$ and the

observation $\underline{y}$. From $\underline{x} = \mathbf{H}\underline{\theta}$ we deduce that the TLS estimate of $\underline{\theta}$ is

$$\underline{\hat{\theta}}_{TLS} = (\hat{\mathbf{H}}^H \hat{\mathbf{H}})^{-1} \hat{\mathbf{H}}^H \underline{\hat{x}}_{TLS} \tag{3.78}$$

The subspace $\left\langle \hat{\mathbf{H}} \right\rangle$ on which the TLS solution is based can be found from the Singular Value Decomposition (SVD) of the matrix formed by concatenating $\mathbf{H}$ and $\underline{y}$ as $[\mathbf{H}\ \underline{y}]$:

$$[\mathbf{H}\ \underline{y}] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H. \tag{3.79}$$

Observe that the TLS error is

$$\begin{aligned} e^2 &= ||\underline{\hat{\nu}}_{TLS}||_2^2 + ||\mathbf{\Delta}_{\mathbf{H}}||_F^2 \\ &= ||(\mathbf{I} - \mathbf{P}_1)[\mathbf{H}\ \underline{y}]||_F^2 \\ &= \mathrm{Tr}\left\{(\mathbf{I} - \mathbf{P}_1)[\mathbf{H}\ \underline{y}][\mathbf{H}\ \underline{y}]^H(\mathbf{I} - \mathbf{P}_1)\right\} \\ &= \mathrm{Tr}\left\{(\mathbf{I} - \mathbf{P}_1)\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H(\mathbf{I} - \mathbf{P}_1)\right\} \\ &= \mathrm{Tr}\left\{(\mathbf{I} - \mathbf{P}_1)\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^H(\mathbf{I} - \mathbf{P}_1)\right\} \end{aligned} \tag{3.80}$$

It is clear that to minimize the TLS error $e^2$ in the preceding expression, we should choose $\mathbf{P}_1$ to be aligned with the $m$ dominant left singular vectors in $\mathbf{U}$ which correspond to the $m$ largest singular values in $\mathbf{\Sigma}$, or equivalently choose the rank-one projection $(\mathbf{I} - \mathbf{P}_1)$ to be aligned with the one least dominant left singular vector. The solution is unique provided that the smallest singular value is not repeated. If the singular values are ordered from largest to smallest we can writ the solution for $\mathbf{P}_1$ as

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{U}_1\mathbf{U}_1^H, \\ &\text{where } \mathbf{U} = [\mathbf{U}_1\ \underline{u}]. \end{aligned} \tag{3.81}$$

Thus, Total Least Squares can be used to update a given signal subspace $\langle \mathbf{H} \rangle$ each time a new data vector is received. If we wish to change the rate at which the signal subspace adapts to new data, we can postmutiply $[\mathbf{H}\ \underline{y}]$ by a diagonal weighting matrix before taking its SVD.

## 3.6 Total Least Squares for Signal and Noise Subspace Updates

In this section we extend the TLS subspace updating technique of the preceding section to the structured noise model. Suppose we have a signal and structured noise model embodied in given matrices $\mathbf{H}$ and $\mathbf{S}$, but we know that the model matrices may be subject to error. This would be the case, for example, if the model were the result of a subspace identification process or if the actual subspaces were slowly varying in time. In this case, the technique of Total Least Squares (TLS) allows both signal and structured noise subspaces to be updated based on new data.

We wish to approximately solve the overdetermined linear system given by:

$$\underline{y} \approx \mathbf{H}\underline{\theta} + \mathbf{S}\underline{\phi}, \tag{3.82}$$

where we are given

$$\underline{y} \in \mathbb{C}^{n\times 1}$$

$$\mathbf{H} \in \mathbb{C}^{n\times m}$$

$$\mathbf{S} \in \mathbb{C}^{n\times t},$$

and

$$m + t \leq n.$$

That is, we wish to find $\hat{\underline{\theta}}$ and $\hat{\underline{\phi}}$ for which

$$(\underline{y} + \underline{\delta}) = (\mathbf{H} + \mathbf{\Delta_H})\hat{\underline{\theta}} + (\mathbf{S} + \mathbf{\Delta_S})\hat{\underline{\phi}}, \tag{3.83}$$

and the total sum of the squared perturbations

$$\varepsilon = \| [\mathbf{\Delta_H}\ \mathbf{\Delta_S}\ \underline{\delta}] \|_F^2 \tag{3.84}$$

is minimized.

Observe that the system can be rewritten in a form which makes it equivalent to the original Total Least Squares (TLS) problem solved by Golub and Van Loan. We write

$$\underline{y} \approx [\mathbf{H}\ \mathbf{S}] \begin{bmatrix} \hat{\underline{\theta}} \\ \hat{\underline{\phi}} \end{bmatrix}, \tag{3.85}$$

and apply their TLS solution, but with additional partitioning of the matrices involved. That

is, let the matrix $\mathbf{C} \in \mathbb{C}^{n\times(m+t+1)}$ be defined as the concatenation of $\mathbf{H}$, $\mathbf{S}$, and $\underline{y}$:

$$\mathbf{C} = [\mathbf{H}\ \mathbf{S}\ \underline{y}]. \tag{3.86}$$

Write the singular value decomposition of $\mathbf{C}$ as

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \tag{3.87}$$

where we partition the matrices as

$$\begin{aligned}
\mathbf{U} &= \underset{m+t}{[\,\mathbf{U}_1} \quad \underset{1}{\underline{u}_2\,]} \quad n \\
\mathbf{\Sigma} &= \underset{m+t}{\begin{bmatrix} \mathbf{\Sigma}_1 & \underline{0} \\ \underline{0}^T & \sigma_2 \end{bmatrix}} \quad \begin{matrix} m+t \\ 1 \end{matrix} \\
\mathbf{V} &= \underset{m+t}{\begin{bmatrix} \mathbf{V}_{11} & \underline{v}_{12} \\ \mathbf{V}_{21} & \underline{v}_{22} \\ \underline{v}_{31}^H & v_{32} \end{bmatrix}} \quad \begin{matrix} m \\ t \\ 1. \end{matrix}
\end{aligned} \tag{3.88}$$

The singular values in $\mathbf{\Sigma}$ are sorted with the largest in the upper left and the smallest in the lower right for the partitioning. Again the solution is unique only if there is a unique smallest singular value.

The TLS solution for the parameters is

$$\begin{aligned}
\hat{\underline{\theta}} &= -\frac{1}{v_{32}}\underline{v}_{12} \\
\hat{\underline{\varrho}} &= -\frac{1}{v_{32}}\underline{v}_{22}
\end{aligned} \tag{3.89}$$

with $v_{32}$ assumed nonzero.

The form of the TLS solution given by Zoltowski [Zol87] allows the perturbations to be easily characterized. Define the perturbed variables

$$\begin{aligned}
\hat{\mathbf{H}} &= (\mathbf{H} + \mathbf{\Delta}_\mathbf{H}) \\
\hat{\mathbf{S}} &= (\mathbf{S} + \mathbf{\Delta}_\mathbf{S}) \\
\hat{\underline{y}} &= (\underline{y} + \underline{\delta}).
\end{aligned} \tag{3.90}$$

and the orthogonal projection

$$\mathbf{P}_1 = \mathbf{U}_1\mathbf{U}_1^H. \tag{3.91}$$

Then by applying Zoltowski's work, we find that

$$\begin{aligned} \hat{\mathbf{H}} &= \mathbf{P}_1\mathbf{H} \\ \hat{\mathbf{S}} &= \mathbf{P}_1\mathbf{S} \\ \hat{\underline{y}} &= \mathbf{P}_1\underline{y}. \end{aligned} \tag{3.92}$$

This means that when the original overdetermined system of equations is projected onto $\langle \mathbf{U}_1 \rangle$, the exact solution to the projected system is the TLS solution to the original system. We have

$$\begin{aligned} \underline{y} &\approx \mathbf{H}\underline{\theta} + \mathbf{S}\underline{\phi} \\ \Rightarrow \quad \mathbf{P}_1\underline{y} &= \mathbf{P}_1\mathbf{H}\hat{\underline{\theta}} + \mathbf{P}_1\mathbf{S}\hat{\underline{\phi}} \\ \Rightarrow \quad \hat{\underline{y}} &= \hat{\mathbf{H}}\hat{\underline{\theta}} + \hat{\mathbf{S}}\hat{\underline{\phi}}. \end{aligned} \tag{3.93}$$

The estimated signal is

$$\hat{\underline{x}} = \hat{\mathbf{H}}\hat{\underline{\theta}}. \tag{3.94}$$

Let $\mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}$ be the oblique projection with range $\left\langle \hat{\mathbf{H}} \right\rangle$ and null space containing $\left\langle \hat{\mathbf{S}} \right\rangle$ as given in Chapter II. Application of $\mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}$ to both sides of Equation 3.93 gives another way to write the signal estimate:

$$\begin{aligned} \mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}\hat{\underline{y}} &= \mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}\hat{\mathbf{H}}\hat{\underline{\theta}} + \mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}\hat{\mathbf{S}}\hat{\underline{\phi}} \\ &= \hat{\mathbf{H}}\hat{\underline{\theta}} \\ &= \hat{\underline{x}}. \end{aligned} \tag{3.95}$$

So $\hat{\underline{x}}$ can be obtained with an oblique projection.

The complete system for realizing the above equations is shown in Figure 3.3. The $\hat{\mathbf{H}}$ and $\hat{\mathbf{S}}$ outputs of the system represent updated versions of the signal and noise subspaces, based on the new data $\underline{y}$. Recall that we began the TLS problem with the basic assumption that there may be errors in $\mathbf{H}$ and $\mathbf{S}$, so these outputs can be viewed as an attempt to correct those errors and bring the model into agreement with the observations.

It is clear that the updated signal subspace should depend on the current signal subspace and the new data. But the new signal subspace in the TLS update depends also on the structured noise subspace (through the SVD). Conversely, the signal subspace affects the update of the structured noise subspace. That the structured noise has any influence on the

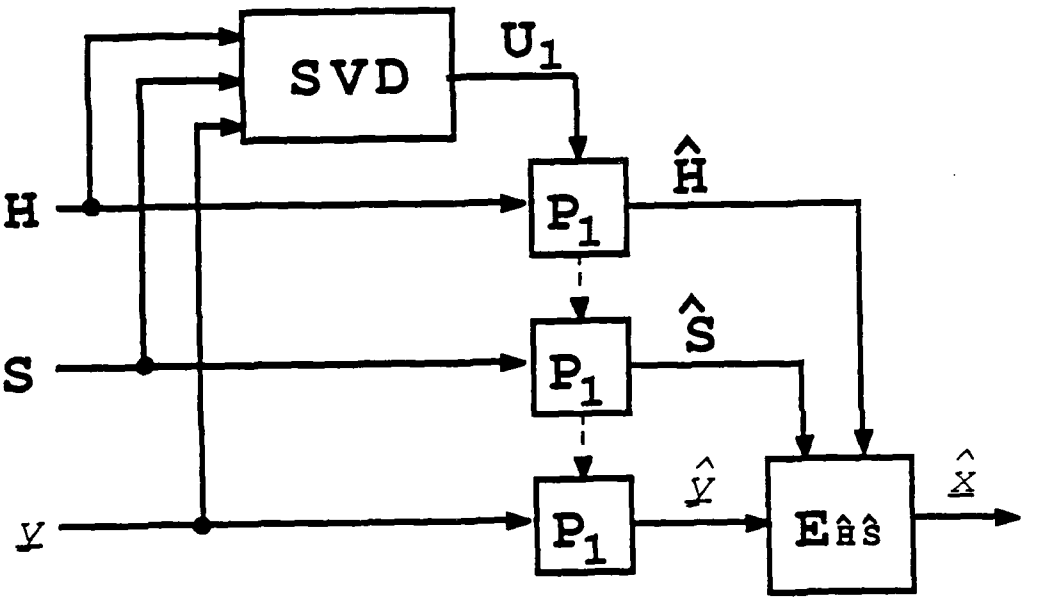


Figure 3.3 TLS with Structured Noise.

new signal subspace seems undesirable at first, since the goal of the structured noise model is to reduce the impact of the structured noise on the signal estimate. But we must consider that the new data has components of both signal and structured noise, and that the signal component cannot be isolated without using the structured noise subspace (e.g. through the oblique projection). This situation justifies some interaction between signal and structured noise subspaces in the update, although the actual interaction may not correspond exactly to the justified interaction.

**Partially fixed subspace models.** We now extend this technique to deal with the case where only part of the model is subject to error. As a basis for the development assume that the structured noise matrix **S** is still uncertain, but the signal matrix **H** is known to be without error. In this case, $\Delta_{\mathbf{H}}$ must be zero.

The key to solving the TLS problem for this case is the following observation: The perturbations $\underline{\delta}$ and the columns of $\Delta_{\mathbf{S}}$ must be orthogonal to $\langle \mathbf{H} \rangle$. This is so because if they

were not, each could be resolved into a component in $\langle \mathbf{H} \rangle$ and a component in $\langle \mathbf{H} \rangle^{\perp}$. The component in $\langle \mathbf{H} \rangle$ could be absorbed into the $\mathbf{H}\underline{\theta}$ term with a suitable change to $\underline{\theta}$ and the sum of squared perturbations would be reduced in so doing. Since the solution must give the minimum sum of squared perturbations, the $\langle \mathbf{H} \rangle$ components must be zero.

Assume the solution is

$$(\underline{y} + \underline{\delta}) = \mathbf{H}\hat{\underline{\theta}} + (\mathbf{S} + \mathbf{\Delta_S})\hat{\underline{\phi}}. \tag{3.96}$$

Now operate on the above equation with $\mathbf{P}_{\mathbf{H}^{\perp}}$, the orthogonal projection onto $\langle \mathbf{H} \rangle^{\perp}$:

$$\begin{aligned} & \mathbf{P}_{\mathbf{H}^{\perp}}(\underline{y} + \underline{\delta}) = \mathbf{P}_{\mathbf{H}^{\perp}}\mathbf{H}\hat{\underline{\theta}} + \mathbf{P}_{\mathbf{H}^{\perp}}(\mathbf{S} + \mathbf{\Delta_S})\hat{\underline{\phi}} \\ \Rightarrow \quad & (\mathbf{P}_{\mathbf{H}^{\perp}}\underline{y} + \underline{\delta}) = (\mathbf{P}_{\mathbf{H}^{\perp}}\mathbf{S} + \mathbf{\Delta_S})\hat{\underline{\phi}} \\ \Rightarrow \quad & (\tilde{\underline{y}} + \underline{\delta}) = (\tilde{\mathbf{S}} + \mathbf{\Delta_S})\hat{\underline{\phi}}, \end{aligned} \tag{3.97}$$

where we have defined

$$\begin{aligned} \tilde{\underline{y}} &= \mathbf{P}_{\mathbf{H}^{\perp}}\underline{y} \\ \tilde{\mathbf{S}} &= \mathbf{P}_{\mathbf{H}^{\perp}}\mathbf{S}. \end{aligned} \tag{3.98}$$

The last form of the projected equation above suggests an ordinary TLS problem of the Golub and Van Loan variety. Now we claim that when we solve the projected TLS problem above, we have also solved the unprojected TLS problem. Any set of parameters and perturbations which gives equality before projection must give equality after projection. Conversely, for every set of parameters and perturbations which gives equality in the projected equation, there exists a $\hat{\underline{\theta}}$ which will give equality in the unprojected equation (i.e., the component in $\langle \mathbf{H} \rangle^{\perp}$ is already equal and $\hat{\underline{\theta}}$ can be chosen so that the component in $\langle \mathbf{H} \rangle$ is also equal). For equality we need

$$\hat{\underline{\theta}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\underline{y}, \tag{3.99}$$

so $\hat{\underline{\theta}}$ is fixed during the optimization process. In both equations the optimization occurs over the same variables ($\hat{\underline{\phi}}$, $\mathbf{\Delta_S}$, and $\underline{\delta}$), with the same domain, and the same constraint (equality in both the projected and unprojected equations). Since the objective function is also the same (minimization of the sum of squared elements in $[\mathbf{\Delta_S}\ \underline{\delta}]$), the problems are equivalent.

Now we find $\underline{\delta}$ and $\Delta_{\mathbf{S}}$ by solving the TLS problem of the projected equation. Let $\tilde{\mathbf{P}}_1$ be the orthogonal projection whose range is spanned by the first $t$ left singular vectors of $[\tilde{\mathbf{S}}\ \tilde{\underline{y}}]$. Then we have

$$\begin{aligned} \underline{\delta} &= -(\mathbf{I} - \tilde{\mathbf{P}}_1)\tilde{\underline{y}} = (\tilde{\mathbf{P}}_1 - \mathbf{P}_{\mathbf{H}^\perp})\underline{y} \\ \Delta_{\mathbf{S}} &= -(\mathbf{I} - \tilde{\mathbf{P}}_1)\tilde{\mathbf{S}} = (\tilde{\mathbf{P}}_1 - \mathbf{P}_{\mathbf{H}^\perp})\mathbf{S}. \end{aligned} \tag{3.100}$$

If we define $\mathbf{P}_1$ as

$$\mathbf{P}_1 = (\tilde{\mathbf{P}}_1 + \mathbf{P}_{\mathbf{H}}), \tag{3.101}$$

then the following relationships hold:

$$\begin{aligned} \hat{\mathbf{H}} &= \mathbf{P}_1\mathbf{H} = \mathbf{H} \\ \hat{\mathbf{S}} &= \mathbf{P}_1\mathbf{S} \\ \hat{\underline{y}} &= \mathbf{P}_1\underline{y} \\ \hat{\underline{x}} &= \mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}\hat{\underline{y}} = \mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}\mathbf{P}_1\underline{y}. \end{aligned} \tag{3.102}$$

The oblique projection $\mathbf{E}_{\hat{\mathbf{H}};\hat{\mathbf{S}}}$ is defined in Chapter II. It is easy to see that the matrix $\mathbf{P}_1$ defined above is an orthogonal projection. The main difference between this problem and the problem where $\mathbf{H}$ was also subject to error is in how $\mathbf{P}_1$ is determined.

There is a more general problem wherein there may be some columns of $\mathbf{H}$ known exactly and some subject to error, and likewise for $\mathbf{S}$. This problem is not really much different from the last. The solution process is to project the equation onto the subspace perpendicular to the known parts of the model, solve the projected TLS problem for $\tilde{\mathbf{P}}_1$ and add back the projection onto the known parts to arrive at the projection $\mathbf{P}_1$ for which the above relations hold.

# CHAPTER IV

## Subspace Identification with a Prior Model

In this chapter we present algorithms for identification of signal and noise subspaces that are constrained to obey a prior structural model. We consider signals composed of linear combinations of complex exponentials, and we consider structured noises composed of linear combinations of impulses.

### 4.1 ML Signal Subspace ID with Complex Exponential Model

In this section we give a new presentation of the "KiSS" algorithm of Kumaresan, Scharf and Shaw [KSS86] (see also [EvF73], [BrM86], [McC89], and [StN88]). Our presentation includes a unified approach to implementing the most commonly used constraints on the parameters, and a discussion of when the circulant matrix approach of Kumaresan can and should be used for finding the necessary matrix inverses.

Let the signal be a sum of complex exponential modes, a scalar valued time series:

$$x(t) = \sum_{i=1}^{n_s} \theta_i z_i^t. \tag{4.1}$$

Suppose the observed data consists of signal plus noise for $n$ consecutive time indices:

$$y(t) = x(t) + \nu(t), \qquad t = 1, 2, \ldots, n. \tag{4.2}$$

Arrange these time series into vectors as

$$\underline{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(n) \end{bmatrix}, \qquad \underline{x} = \begin{bmatrix} x(1) \\ \vdots \\ x(n) \end{bmatrix}, \qquad \underline{\nu} = \begin{bmatrix} \nu(1) \\ \vdots \\ \nu(n) \end{bmatrix}; \qquad \underline{y}, \underline{x}, \underline{\nu} \in \mathbb{C}^n. \tag{4.3}$$

The signal model may be arranged in matrix form as

$$\underline{x} = \mathrm{H}\underline{\theta}, \tag{4.4}$$

where the signal subspace is determined by the (non-square) Vandermonde mode matrix

$$\mathbf{H} = \begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_1^2 & z_2^2 & \cdots & z_m^2 \\ \vdots & \vdots & & \vdots \\ z_1^n & z_2^n & \cdots & z_m^n \end{bmatrix} \in \mathbb{C}^{n \times m}, \tag{4.5}$$

and the position within the signal subspace is parameterized by

$$\underline{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix} \in \mathbb{C}^m. \tag{4.6}$$

Kumaresan, Scharf and Shaw [KSS86] have pointed out that the perp-space to the signal subspace is spanned by the Toeplitz matrix

$$\mathbf{A} = \begin{bmatrix} a_m^* & & & 0 \\ \vdots & \ddots & & \\ a_0^* & & \ddots & \\ & \ddots & & a_m^* \\ & & \ddots & \vdots \\ 0 & & & a_0^* \end{bmatrix} \in \mathbb{C}^{n \times n-m}, \tag{4.7}$$

where the elements $a_i$ are the coefficients of a polynomial whose roots are the $z_i$'s:

$$\sum_{i=0}^{m} a_i z_j^{-i} = 0, \qquad j = 1, \ldots, m. \tag{4.8}$$

We have

$$\mathbf{A}^H \mathbf{H} = \mathbf{0}. \tag{4.9}$$

We wish to estimate the parameters $a_0 \ldots a_m$ that determine the signal subspace through Equations 4.8 and 4.5. A least squares approach to estimating these parameters was proposed in [KSS86], equivalent to

$$\min_{\underline{a},\underline{\theta}} ||\underline{y} - \hat{\underline{x}}||^2, \tag{4.10}$$

where $\hat{\underline{x}} = \mathbf{H}(\underline{a})\underline{\theta}$. Bresler and Macovski [BrM86] then reported the same algorithm for minimization of the same objective function, calling it Maximum Likelihood estimation. It is, of course, both least squares and ML when the noise is additive, white, zero-mean, stationary and Gaussian.

It is convenient to arrange the parameters into a vector:

$$\underline{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix}. \tag{4.11}$$

After compression on $\underline{\theta}$, the negative log-likelihood function for parameters $a_0 \ldots a_m$, given observations $\underline{y}$, is

$$L(\underline{a}) = \underline{y}^H \mathbf{P}_\mathbf{A} \underline{y} = \underline{y}^H \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \underline{y}. \tag{4.12}$$

While $L$ is a simple quadratic form in the data $\underline{y}$, we need to minimize it with respect to the parameters $\underline{a}$. Since $L(\underline{a})$ is nonlinear in $\underline{a}$, iterative numerical approaches have been used to solve it. It is, in fact, a rational polynomial function in the parameters $a_0 \ldots a_m$, with numerator and denominator of order $2(n-m)$. This implies that we could upper bound the number of local minima of $L(\underline{a})$. One can envision such a bound being useful for locating the global minimum with certainty, although we have not proposed any such algorithms.

Let us digress a moment at this point to discuss how the principles we developed in Section 3.1 for application of ML estimators apply to the KiSS problem. The objective function of Equation 4.12 is appropriate if the parameters $\underline{a}$ are uniformly distributed, or at least if we have no reason to believe that they are highly nonuniform. But if, for example, uniformity is a better description of the polar coordinates of the root locations then the appropriate estimator is a MAP estimator based on the derived density of the parameters $\underline{a}$ as in the example of Section 3.1. The objective function for this case is obtained by adding to Equation 4.12 a term equal to the natural log of the density function of $\underline{a}$. This kind of a change in the objective function means that the KiSS algorithm we are about to describe would not apply. Depending on the nature of the density function for $\underline{a}$ it may be possible to use a KiSS-like principle to derive a minimization algorithm, namely to hold part of the expression in $\underline{a}$ constant for each iteration and optimize with respect to the remaining occurrences of $\underline{a}$. Even if this is not feasible it may still be possible to compute the derivatives needed to find the minimum by Newtons method as described later in this chapter. Finally, if even the derivatives are intractable, Newton's method can be implemented with finite difference approximations to the derivatives.

The key to the KiSS algorithm for minimizing the quadratic form of Equation 4.12 is to rewrite the product of the Toeplitz matrix $\mathbf{A}$ and the vector $\underline{y}$ as follows:

$$\mathbf{A}^H \underline{y} = \mathbf{Y}\underline{a}, \tag{4.13}$$

where $\mathbf{Y}$ is a Toeplitz data matrix defined as

$$\mathbf{Y} = \begin{bmatrix} y(m+1) & \cdots & y(1) \\ \vdots & \ddots & \vdots \\ \vdots & & y(m+1) \\ y(n-m) & & \vdots \\ \vdots & \ddots & \vdots \\ y(n) & \cdots & y(n-m) \end{bmatrix}. \tag{4.14}$$

Equation 4.13 can be viewed as an expression of the commutativity of convolution. With this identity, the objective function becomes

$$L(\underline{a}) = \underline{a}^H \mathbf{Y}^H (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{Y} \underline{a}. \tag{4.15}$$

Then at each iteration, the matrix $(\mathbf{A}^H\mathbf{A})^{-1}$ is held constant and the resulting quadratic form in $\underline{a}$ is minimized. At the zeroth iteration, $(\mathbf{A}^H\mathbf{A})^{-1}$ is set equal to the identity matrix. At each subsequent iteration, it is built from the solution to the previous iteration. It is interesting to note that the solution at the first iteration is equal to the Prony method.

**Constraints in the KiSS algorithm.** Prior knowledge about the signal may lead us to impose constraints on the locations of the roots $z_i$. The constraints of interest are spelled out clearly in [BrM86]. Starer and Nehorai [StN88] introduced a simple, yet powerful, model for implementing most of these constraints in the context of using a Newton method to solve the same minimization problem we are considering. Those constraints that cannot be imposed by the method of Starer and Nehorai have generally been considered too difficult to enforce at all (see [BrM86] and [KSS86]). Starer and Nehorai impose the tractable constraints by modeling the parameter vector $\underline{a}$ as an affine transformation of a minimally dimensioned new parameter vector $\underline{\alpha}$:

$$\underline{a} = \mathbf{T}\underline{\alpha} + \underline{c}, \tag{4.16}$$

where $\mathbf{T}$ is a constant rectangular matrix, and $\underline{c}$ is a constant vector. In our extension of this constraint to complex valued parameter vectors we take $\underline{\alpha}$ to be real valued, letting $\mathbf{T}$ map it into the appropriate complex values.

We now give explicit forms of $\mathbf{T}$ and $\underline{c}$ to implement the constraints of interest. The matrix $\mathbf{J}_n$ is the $n \times n$ exchange matrix (reverse permutation):

$$\mathbf{J}_n = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \in \mathbf{R}^{n \times n}, \tag{4.17}$$

and $\mathbf{I}_n$ is the $n \times n$ identity matrix. A column vector of $n$ zeros is represented by $\underline{0}_n$.

**Constraints for nontriviality.** The "nontriviality" constraint requires $a_0 = 1$ to ensure that the corresponding polynomial is monic and of degree $m$. To implement the nontriviality constraint and map real $\underline{\alpha}$ into complex $\underline{a}$ we take

$$\mathbf{T} = \left[\begin{array}{c|c} \underline{0}_{2m}^T & \\ \hline \mathbf{I}_m & j\mathbf{I}_m \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_m \end{bmatrix}, \tag{4.18}$$

so that the relationship between $\underline{a}$ and $\underline{\alpha}$ may be expressed as

$$\underline{a} = \begin{bmatrix} 1 \\ \operatorname{Re} a_1 + j \operatorname{Im} a_1 \\ \vdots \\ \operatorname{Re} a_m + j \operatorname{Im} a_m \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha_1 + j\alpha_{m+1} \\ \vdots \\ \alpha_m + j\alpha_{2m} \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} \operatorname{Re} a_1 \\ \vdots \\ \operatorname{Re} a_m \\ \operatorname{Im} a_1 \\ \vdots \\ \operatorname{Im} a_m \end{bmatrix}. \tag{4.19}$$

**Constraints for modeling undamped complex sinusoids.** When modeling undamped complex sinusiodal signals, we wish the roots of $\underline{a}$ to lie on the unit circle. A necessary, but not sufficient, condition for the roots of $\underline{a}$ to lie on the unit circle is complex conjugate symmetry of the coefficients

$$a_i = a_{m-i}^*. \tag{4.20}$$

The sufficient conditions for roots on the unit circle cannot be imposed with this kind of affine constraint. To require complex conjugate symmetry of the coefficients, along with the nontriviality constraint $\operatorname{Re}(a_0) = 1$, we consider two subcases. For $m$ odd we take $q = (m-1)/2$

and

$$\mathbf{T} = \left[\begin{array}{c|c} \underline{0}_q^T & \\ \hline \mathbf{I}_q & j\mathbf{I}_{q+1} \\ \hline \mathbf{J}_q & \\ \hline \underline{0}_q^T & -j\mathbf{J}_{q+1} \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_{2q} \\ 1 \end{bmatrix}, \tag{4.21}$$

so that

$$\underline{a} = \begin{bmatrix} 1 + j\alpha_{q+1} \\ \alpha_1 + j\alpha_{q+2} \\ \vdots \\ \alpha_q + j\alpha_m \\ \alpha_q - j\alpha_m \\ \vdots \\ \alpha_1 - j\alpha_{q+2} \\ 1 - j\alpha_{q+1} \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} \operatorname{Re} a_1 \\ \vdots \\ \operatorname{Re} a_q \\ \operatorname{Im} a_0 \\ \vdots \\ \operatorname{Im} a_q \end{bmatrix}. \tag{4.22}$$

For $m$ even we take $q = (m-2)/2$ and

$$\mathbf{T} = \left[\begin{array}{c|c} \underline{0}_{q+1}^T & \\ \hline \mathbf{I}_{q+1} & j\mathbf{I}_{q+1} \\ & \underline{0}_{q+1}^T \\ \hline \mathbf{J}_q \;\vert\; \underline{0}_q & -j\mathbf{J}_{q+1} \\ \hline \underline{0}_{q+1}^T & \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_{2q+1} \\ 1 \end{bmatrix}, \tag{4.23}$$

so that

$$\underline{a} = \begin{bmatrix} 1 + j\alpha_{q+2} \\ \alpha_1 + j\alpha_{q+3} \\ \vdots \\ \alpha_q + j\alpha_m \\ \alpha_{q+1} \\ \alpha_q - j\alpha_m \\ \vdots \\ \alpha_1 - j\alpha_{q+3} \\ 1 - j\alpha_{q+2} \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} \operatorname{Re} a_1 \\ \vdots \\ \operatorname{Re} a_{q+1} \\ \operatorname{Im} a_0 \\ \vdots \\ \operatorname{Im} a_{q+1} \end{bmatrix}. \tag{4.24}$$

**Constraints for real data.** If the data is real, then the coefficients $\underline{a}$ should be real.

For real data with only the nontriviality constraint we take

$$\mathbf{T} = \left[\begin{array}{c} \underline{0}_m^T \\ \hline \mathbf{I}_m \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_m \end{bmatrix}, \tag{4.25}$$

so that

$$\underline{a} = \begin{bmatrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}. \tag{4.26}$$

**Constraints for real undamped sinusoids.** For real data with constraints of nontriviality and symmetry,

$$a_i = a_{m-i}, \tag{4.27}$$

we again consider two subcases. For m odd we take $q = (m-1)/2$ and

$$\mathbf{T} = \left[\begin{array}{c} \underline{0}_q^T \\ \hline \mathbf{I}_q \\ \hline \mathbf{J}_q \\ \hline \underline{0}_q^T \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_{2q} \\ 1 \end{bmatrix}, \tag{4.28}$$

so that

$$\underline{a} = \begin{bmatrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_q \\ \alpha_q \\ \vdots \\ \alpha_1 \\ 1 \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} a_1 \\ \vdots \\ a_q \end{bmatrix}. \tag{4.29}$$

For m even we take $q = (m-2)/2$ and

$$\mathbf{T} = \left[\begin{array}{c|c} \underline{0}_{q+1}^T & \\ \hline \mathbf{I}_{q+1} & \\ \hline \mathbf{J}_q & \underline{0}_q \\ \hline \underline{0}_{q+1}^T & \end{array}\right], \qquad \underline{c} = \begin{bmatrix} 1 \\ \underline{0}_{2q+1} \\ 1 \end{bmatrix}, \tag{4.30}$$

so that

$$\underline{a} = \begin{bmatrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_q \\ \alpha_{q+1} \\ \alpha_q \\ \vdots \\ \alpha_1 \\ 1 \end{bmatrix}, \qquad \underline{\alpha} = \begin{bmatrix} a_1 \\ \vdots \\ a_{q+1} \end{bmatrix}. \tag{4.31}$$

**The KiSS algorithm.** With the affine constraint of Equation 4.16, the objective function of Equation 4.15 becomes

$$L(\underline{\alpha}) = (\underline{c}^H + \underline{\alpha}^H \mathbf{T}^H)\mathbf{Y}^H[\mathbf{A}(\underline{\alpha})^H \mathbf{A}(\underline{\alpha})]^{-1}\mathbf{Y}(\mathbf{T}\underline{\alpha} + \underline{c}) \tag{4.32}$$

Let

$$\mathbf{Q} = \mathbf{Y}^H[\mathbf{A}(\underline{\alpha})^H\mathbf{A}(\underline{\alpha})]^{-1}\mathbf{Y}, \tag{4.33}$$

and at each iteration fix $\mathbf{Q} = \mathbf{Q}_i$ and minimize

$$L_i(\underline{\alpha}) = (\underline{c}^H + \underline{a}^H\mathbf{T}^H)\mathbf{Q}_i(\mathbf{T}\underline{\alpha} + \underline{c}). \tag{4.34}$$

This is a quadratic optimization problem. Setting the derivative of $L_i(\underline{\alpha})$ equal to zero:

$$\nabla_{\underline{\alpha}} L_i(\underline{\alpha}) = 2\mathbf{T}^H\mathbf{Q}_i\mathbf{T}\underline{\alpha} + 2\mathbf{T}^H\mathbf{Q}_i\underline{c} = 0. \tag{4.35}$$

Since $\underline{\alpha}$ is real valued, $\nabla_{\underline{\alpha}} L(\underline{\alpha})$ must be real valued, and the new $\underline{\alpha}$ for iteration $i$ is

$$\underline{\alpha}_{i+1} = -[\mathrm{Re}(\mathbf{T}^H\mathbf{Q}_i\mathbf{T})]^{-1}\,\mathrm{Re}(\mathbf{T}^H\mathbf{Q}_i\underline{c}). \tag{4.36}$$

In summary, the constrained KiSS algorithm attempts to minimize $L(\underline{\alpha})$ by the following steps:

1. Build appropriate $\mathbf{Y}, \mathbf{T}, \underline{c}$.
2. Set $\mathbf{Q} = \mathbf{Y}^H\mathbf{Y}$.
3. Repeat until convergence:

   3a. Let $\underline{\alpha} = -[\mathrm{Re}(\mathbf{T}^H\mathbf{Q}\mathbf{T})]^{-1}\,\mathrm{Re}(\mathbf{T}^H\mathbf{Q}\underline{c})$.

   3b. Let $\underline{a} = \mathbf{T}\underline{\alpha} + \underline{c}$.

   3c. Build $\mathbf{A}$ from $\underline{a}$.

   3d. Let $\mathbf{Q} = \mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}$.
4. Stop.

Convergence is considered to be reached when no element of $\underline{\alpha}$ changes by more than some specified tolerance from one iteration to the next. We note that while the KiSS algorithm is known to be effective in practice, there is no guarantee that it will converge near the global minimum of $L(\underline{\alpha})$.

Our MATLAB™ code for the KiSS algorithm is given in Appendix A.

**Example: Second order KiSS objective function.** We may gain some insight

into the nature of the KiSS objective function of Equation 4.32 by examining it for a simple case. When $\underline{\alpha}$ consists of two real parameters we can generate a 3-dimensional plot of the objective function versus $\alpha_1$ and $\alpha_2$. Such a plot is shown in Figure 4.1, for a data set containing 25 samples of two complex exponentials and a second order model constrained to have complex conjugate symmetry (this signal is the one used in [TuK82]). You can see some tendencies toward symmetry about the global minimum, as well as the appearance of several local minima both on the "plateau" and in the "canyon". The "ridge" surrounding the global minimum would tend to foil descent based algorithms from most starting points. The KiSS algorithm, however, is not descent based, and usually converges near the global minimum except below the threshold SNR discussed later in the simulations section.

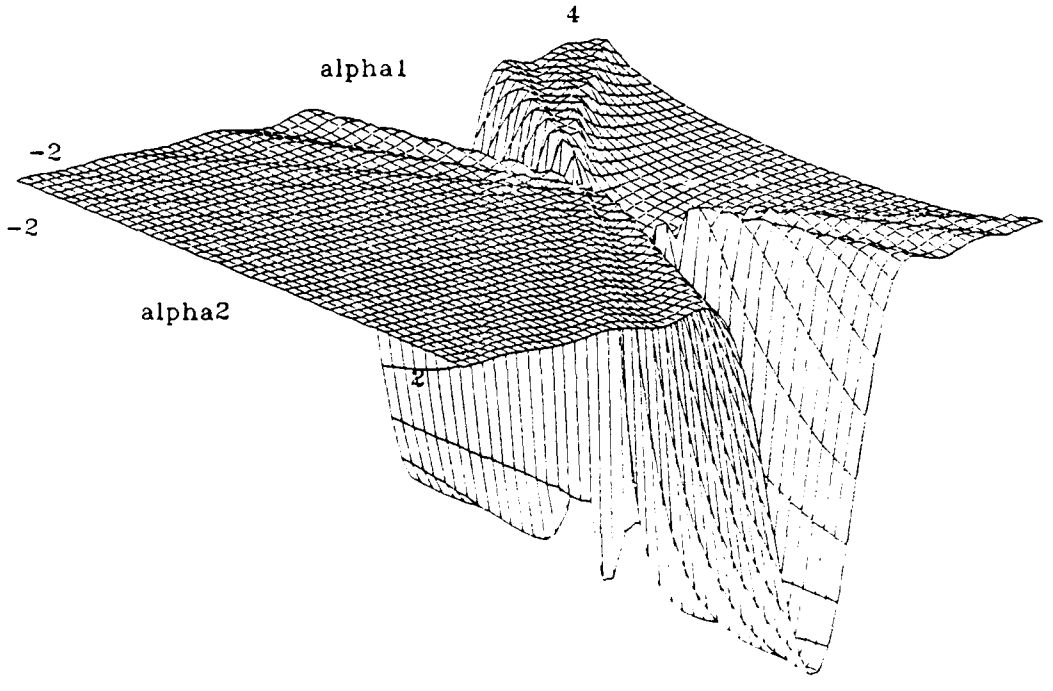


Figure 4.1 A Noiseless KiSS Objective Function.

Figure 4.2 shows that the addition of white Gaussian noise up to an input SNR of 10 dB (as defined in the following simulations section) does not change the basic characteristics of the KiSS objective function but does make the "plateau" rougher. It also perturbs the location of the global minimum.

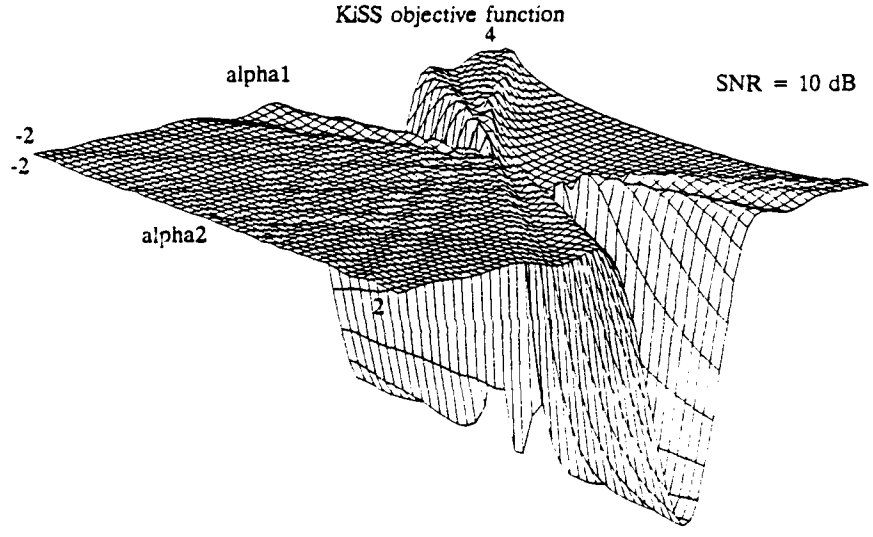


Figure 4.2 KiSS Objective Function at 10 dB SNR.

**KiSS implementation issues.** An issue involved in efficiently implementing the KiSS algorithm is inversion of the matrix

$$\mathbf{G} = (\mathbf{A}^H \mathbf{A}). \tag{4.37}$$

Kumaresan, Scharf and Shaw [KSS86] presented an efficient algorithm for this inversion based on properties of circulant matrices. While their algorithm is often a good idea, it has some lim-

itations. Before discussing those limitations we present the inversion algorithm here, correcting some errors in [KSS86]. First note that $\mathbf{G}$ is a banded MA Toeplitz matrix

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_m & 0 & \cdots & 0 \\ g_1^* & g_0 & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & 0 \\ g_m^* & & \ddots & \ddots & \ddots & & g_m \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & g_1 \\ 0 & \cdots & 0 & g_m^* & \cdots & g_1^* & g_0 \end{bmatrix}, \tag{4.38}$$

with elements

$$g_{j-i} = \begin{cases} \sum_{k=0}^{m-(j-i)} a_k a_{k+j-i}^* & \text{for } 0 \le j-i \le m \\ g_{i-j}^* & \text{for } -m \le j-i \le 0 \\ 0 & \text{for } |j-i| > m \end{cases} \tag{4.39}$$

If $n$ is at least as large as $3m$ we can define

$$\mathbf{U} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{n-3m\times m} & \mathbf{0}_{n-3m\times m} \\ \mathbf{0}_{m\times m} & \begin{bmatrix} 0 & & g_m \\ & \iddots & \vdots \\ g_m & \cdots & g_1 \end{bmatrix} \end{bmatrix} \in \mathbb{C}^{n-m\times 2m}, \tag{4.40}$$

and

$$\mathbf{V} = \mathbf{U}\mathbf{J}_{2m}. \tag{4.41}$$

Then we can make $\mathbf{G}$ into a circulant matrix $\mathbf{C}$ by adding to it the product $\mathbf{U}\mathbf{V}^H$:

$$\mathbf{C} = \mathbf{G} + \mathbf{U}\mathbf{V}^H = \begin{bmatrix} g_0 & g_1 & \cdots & g_m & 0 & \cdots & 0 & g_m^* & \cdots & g_1^* \\ g_1^* & g_0 & \ddots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & & \ddots & g_m^* \\ g_m^* & & \ddots & \ddots & \ddots & & \ddots & \ddots & & 0 \\ 0 & \ddots & & \ddots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \ddots & & \ddots & 0 \\ 0 & & \ddots & \ddots & & \ddots & \ddots & \ddots & & g_m \\ g_m & \ddots & & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & & \ddots & \ddots & g_1 \\ g_1 & \cdots & g_m & 0 & \cdots & 0 & g_m^* & \cdots & g_1^* & g_0 \end{bmatrix}. \tag{4.42}$$

A circulant matrix is computationally easy to invert using the discrete Fourier transform

[Dav79]. Let $c_i$ be the sequence defined by the first row of $\mathbf{C}$, indexed from $i = 0$ to $i = n-m-1$. Let $\lambda_l$ be the coefficients of the inverse discrete Fourier transform of the sequence $c_i$. Because of the symmetry of $c_i$ we can use the discrete cosine transform:

$$\lambda_l = \sum_{i=0}^{n-m-1} c_i \cos \frac{2\pi i l}{n-m} \qquad \text{for } 0 \le l \le n-m-1. \tag{4.43}$$

To compute the first row of $\mathbf{C}^{-1}$ (again indexed from zero) we take the forward discrete Fourier transform of the sequence defined by the reciprocals of $\lambda_l$:

$$(\mathbf{C}^{-1})_{0,k} = \frac{1}{n-m} \sum_{l=0}^{n-m-1} \lambda_l^{-1} \exp\left(-j \frac{2\pi l k}{n-m}\right) \qquad \text{for } 0 \le k \le n-m-1. \tag{4.44}$$

The rest of $\mathbf{C}^{-1}$ is obtained by circular shifts of the first row to create a circulant matrix. Once $\mathbf{C}^{-1}$ has been computed, we can use the Woodbury identity [GVL89] to obtain $\mathbf{G}^{-1}$ with only an inverse of size $2m$:

$$\mathbf{G}^{-1} = (\mathbf{C} - \mathbf{U}\mathbf{V}^H)^{-1} = \mathbf{C}^{-1} + \mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_{2m} - \mathbf{V}^H\mathbf{C}^{-1}\mathbf{U})^{-1}\mathbf{V}^H\mathbf{C}^{-1}. \tag{4.45}$$

The first question about the circulant matrix technique for inverting $\mathbf{G}$ is when it is more efficient than general inversion algorithms. General matrix inversion techniques require order $(n-m)^3$ operations, while the circulant technique reduces it to order $(n-m)\ln(n-m)$ (assuming an FFT algorithm) for inverting $\mathbf{C}$, plus order $(2m)^3$ for the smaller inverse. If $n$ is large with respect to $m$, this is an improvement in efficiency. Preliminary tests conducted with MATLAB$^{TM}$ suggest that the circulant technique pays off approximately when $n > 6m$. Inversion of $\mathbf{G}$ could instead be accomplished by a variation of the Levinson algorithm, taking advantage of the Toeplitz structure to invert $\mathbf{G}$ in order $(n-m)^2$ operations. Hence, the circulant technique is likely to be the most efficient in even fewer cases.

The other limitation of the circulant technique is that the circulant matrix $\mathbf{C}$ may be singular, even when $\mathbf{G}$ is nonsingular. In this case the algorithm fails. This problem is made worse by the fact that certain conditions guarantee singularity of $\mathbf{C}$. These conditions are that $\underline{a}$ be real and symmetric (as for real undamped sinusoids), and both $m$ and $n$ be odd. The

proof is to construct $\mathbf{C}$ for this case and show that

$$\mathbf{C}\underline{x} = \underline{0} \tag{4.46}$$

when $\underline{x}$ is given by

$$\underline{x} = \begin{bmatrix} 1 \\ -1 \\ \vdots \\ 1 \\ -1 \end{bmatrix}. \tag{4.47}$$

To prove Equation 4.46, first write the elements of $\mathbf{C}$ as linear combinations of the products $\alpha_i\alpha_j$, where $\underline{\alpha}$ is the parameter vector corresponding to the constraints of of Equation 4.28. A given row of Equation 4.46 can be demonstrated true by collecting coefficients of $\alpha_i\alpha_j$ into a matrix indexed by $i$ and $j$. We have not included the details of the proof, as they are not instructive.

While singularity of $\mathbf{C}$ remains a real danger, the impact of the guaranteed singularity case is less than it first seems. Real symmetry of $\underline{a}$ occurs by constraint when we are modeling real sinusoids, but in this case $m$ is unlikely to be odd since each real sinusoid is rank 2. Hence, it is less likely that the conditions for guaranteed singularity will all be met simultaneously.

## 4.2 A Newton Method for Complex Exponential Subspace ID

The KiSS algorithm was originally presented with a "phase 2" in which the iteration was modified to drive the gradient to zero in order to locate the minimum of the objective function exactly. Phase 2 was first described in [EvF73] for real data and without constraints. But [KSS86], while they state that they used phase 2, do not tell how phase 2 was extended to complex data with constraints. Bresler and Macovski [BrM86] simply dropped phase 2, claiming (perhaps rightfully) that it did not contribute significantly to performance.

A Newton method makes a good alternative to the phase 2 of Evans and Fishl because of its quadratic convergence behavior. Starer and Nehorai [StN88] derived expressions for the gradient and Hessian of the constrained KiSS objective function of Equation 4.32 for the case of real data and parameters. In the following we generalize their work by deriving similar

expressions for the case of complex data and parameters. For implementation of the Newton method, we use and recommend the modifications described by Dennis and Schnabel [DeS83] to improve convergence from poor starting points.

The following derivations of the gradient and Hessian of $L(\underline{\alpha})$ are rather involved, although the end results are pleasingly simple. The basic methods we have followed are those of Magnus and Neudecker in [MaN88]. Their formulas are intended to apply to real matrices, but many can be generalized to complex matrices. The derivatives we seek are of real valued likelihood $L(\underline{\alpha})$ with respect to real valued parameters $\underline{\alpha}$, so that only the intermediate results are complex valued. We use the notation

$$\begin{aligned}
\mathbf{X}^T &: \text{ Transpose of a matrix,} \\
\mathbf{X}^* &: \text{ Complex conjugate,} \\
\mathbf{X}^H &: \text{ Complex conjugate (Hermitian) transpose,} \\
\mathbf{X}^\# &: \text{ Moore-Penrose inverse, } = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H, \\
\mathbf{X} \otimes \mathbf{Y} &: \text{ Kronecker product of matrices,} \\
\mathrm{vec}(\mathbf{X}) &: \text{ Vectorization of a matrix,} \\
d\mathbf{X} &: \text{ Differential of a matrix.}
\end{aligned} \tag{4.48}$$

We also use three types of permutation matrices: the reverse permutation matrix $J_n$ defined in Equation 4.17, the circular down shift matrix $\mathbf{Z}_n$ defined as

$$\mathbf{Z}_n = \begin{bmatrix} \underline{0}_{n-1}^T & 1 \\ \mathbf{I}_{n-1} & \underline{0}_{n-1} \end{bmatrix}, \tag{4.49}$$

and the commutation matrix $\mathbf{K}_{m,n}$ defined to satisfy

$$\mathbf{K}_{m,n}\,\mathrm{vec}(\mathbf{A}) = \mathrm{vec}(\mathbf{A}^T), \qquad \text{for } \mathbf{A} \in \mathbb{C}^{m\times n}. \tag{4.50}$$

We have found the following identities from Magnus and Neudecker especially useful. They are given with their original equation numbers at left.

$$2.2.4 \qquad (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}, \qquad \text{if } \mathbf{AC} \text{ and } \mathbf{BD} \text{ exist.} \tag{4.51}$$

2.4.5 $$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\,\text{vec}\,\mathbf{B}, \tag{4.52}$$

3.7.4 $$\mathbf{K}_{p,m}(\mathbf{A} \otimes \mathbf{B}) = (\mathbf{B} \otimes \mathbf{A})\mathbf{K}_{q,n}, \qquad \text{for } \mathbf{A} \in \mathbb{C}^{m\times n},\ \mathbf{B} \in \mathbb{C}^{p\times q}, \tag{4.53}$$

9.13.17 $$d(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}(d\mathbf{X})\mathbf{X}^{-1}. \tag{4.54}$$

Magnus and Neudecker's identification theorems for first and second derivatives are also quite valuable.

**Gradient.** Begin by letting $q$ be the number of elements in $\underline{\alpha}$, and let $p = n - m$ be the dimension of the perp-space $\langle \mathbf{A} \rangle$:

$$\underline{\alpha} \in \mathbf{R}^q \qquad \text{and} \qquad \mathbf{A} \in \mathbb{C}^{n\times p}. \tag{4.55}$$

We may find an expression for $d\,\text{vec}(\mathbf{A})$, by first noting that

$$\underline{a}^* = (\mathbf{T}\underline{\alpha} + \underline{c})^* = \mathbf{T}^*\alpha + \underline{c}^*. \tag{4.56}$$

Then we can express the mapping from $\underline{a}$ (and $\underline{\alpha}$) to $\mathbf{A}$ as

$$\text{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \underline{a}^* = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T}^*\underline{\alpha} + \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \underline{c}^* \right\}. \tag{4.57}$$

The differential is

$$d\,\text{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T}^* d\underline{\alpha}. \tag{4.58}$$

Next, we need the differential of $\mathbf{A}^H\mathbf{A}$:

$$\begin{aligned} d\,\text{vec}(\mathbf{A}^H\mathbf{A}) &= \text{vec}\, d(\mathbf{A}^H\mathbf{A}) = \text{vec}[\mathbf{A}^H d(\mathbf{A}) + d(\mathbf{A}^H)\mathbf{A}] \\ &= \text{vec}(\mathbf{A}^H d\mathbf{A}) + \mathbf{K}_{p,p}\,\text{vec}(\mathbf{A}^H d\mathbf{A})^*, \end{aligned} \tag{4.59}$$

where $\mathbf{K}_{p,p}$ is a commutation matrix [MaN88]. We may apply Equation 4.52, and use the result of Equation 4.58 to obtain

$$\begin{aligned} d\,\text{vec}(\mathbf{A}^H\mathbf{A}) =& (\mathbf{I}_p \otimes \mathbf{A}^H)\,\text{vec}(d\mathbf{A}) + \mathbf{K}_{p,p}(\mathbf{I}_p \otimes \mathbf{A}^T)\,\text{vec}(d\mathbf{A})^* \\ =& (\mathbf{I}_p \otimes \mathbf{A}^H) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T}^* d\underline{\alpha} + \\ & \mathbf{K}_{p,p}(\mathbf{I}_p \otimes \mathbf{A}^T) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T} d\underline{\alpha}. \end{aligned} \tag{4.60}$$

Next, we need the differential of $\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}$. We use Equations 4.54 and 4.52 to obtain

$$\begin{aligned} d(\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}) &= -\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}d(\mathbf{A}^H\mathbf{A})(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y} \\ \operatorname{vec} d[\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}] &= -\left(\mathbf{Y}^T[(\mathbf{A}^H\mathbf{A})^{-1}]^* \otimes \mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\right) \operatorname{vec} d(\mathbf{A}^H\mathbf{A}). \end{aligned} \tag{4.61}$$

Substituting Equation 4.60 into Equation 4.61 and applying Equations 4.53 and 4.51 gives

$$\begin{aligned} \operatorname{vec} d[\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}] = -\Bigg\{ &\left(\mathbf{Y}^T[(\mathbf{A}^H\mathbf{A})^{-1}]^* \otimes \mathbf{Y}^H\mathbf{A}^{\#}\right) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T}^* + \\ &\mathbf{K}_{m+1,m+1}\left(\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1} \otimes \mathbf{Y}^T(\mathbf{A}^{\#})^*\right) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T} \Bigg\} d\underline{\alpha}. \end{aligned} \tag{4.62}$$

Finally, the differential of $L(\underline{\alpha})$ can be written down using the product rule

$$\begin{aligned} dL(\underline{\alpha}) &= d(\underline{a}^H\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\underline{a}) \\ &= d(\underline{a})^H\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\underline{a} + \underline{a}^H\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}d(\underline{a}) + \underline{a}^H d[\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}]\underline{a}. \end{aligned} \tag{4.63}$$

The first two terms are complex conjugates, so they may be combined as twice the real part of one of them. Also each term in this equation is a scalar, so we can vectorize the third term without changing anything. This allows us to apply Equation 4.52 and put the third term in a form where we can substitute our result from Equation 4.62. Noting also that $d\underline{a} = \mathbf{T}d\underline{\alpha}$, we obtain

$$\begin{aligned} dL(\underline{\alpha}) = 2\operatorname{Re}[&\underline{a}^H\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}]d\underline{\alpha} - \\ &(\underline{a}^T \otimes \underline{a}^H)\Bigg\{ \left(\mathbf{Y}^T[(\mathbf{A}^H\mathbf{A})^{-1}]^* \otimes \mathbf{Y}^H\mathbf{A}^{\#}\right) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T}^* + \\ &\mathbf{K}_{m+1,m+1}\left(\mathbf{Y}^H(\mathbf{A}^H\mathbf{A})^{-1} \otimes \mathbf{Y}^T(\mathbf{A}^{\#})^*\right) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix} \mathbf{T} \Bigg\} d\underline{\alpha}. \end{aligned} \tag{4.64}$$

Since $d\underline{\alpha}$ is entirely factored out to the right of this expression, we can identify the derivative of $L$ with respect to $\underline{\alpha}$ as the row vector that remains when $d\underline{\alpha}$ is dropped from Equation 4.64 (later we will switch to the gradient, which is the transpose of the derivative). In the next sequence of steps, we simplify the expression for the derivative. Applying Equations 4.51 and

4.53 gives us this expression for the derivative:

$$D_{\underline{\alpha}} L(\underline{\alpha}) = 2\,\mathrm{Re}[\underline{a}^H \mathbf{Y}^H (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{Y} \mathbf{T}] -$$
$$(\underline{a}^T \mathbf{Y}^T [(\mathbf{A}^H \mathbf{A})^{-1}]^* \otimes \underline{a}^H \mathbf{Y}^H \mathbf{A}^\#) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1 \times m+1} \end{bmatrix} \mathbf{T}^* - \tag{4.65}$$
$$(\underline{a}^H \mathbf{Y}^H (\mathbf{A}^H \mathbf{A})^{-1} \otimes \underline{a}^T \mathbf{Y}^T (\mathbf{A}^\#)^*) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1 \times m+1} \end{bmatrix} \mathbf{T}.$$

Now let $\underline{u}$ denote the following function of the measurement $\underline{y}$:

$$\underline{u} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{Y} \underline{a} = \mathbf{A}^\# \underline{y} \quad \in \mathbb{C}^p. \tag{4.66}$$

This definition of $\underline{u}$ is obviously convenient for simplifying the derivative expression, but it also has a physical interpretation. It contains just the information necessary to reproduce the *residual*, defined as the projection of the data $\underline{y}$ onto the orthogonal complement of the signal subspace:

$$\hat{\underline{\nu}} = \mathbf{P}_{\mathbf{A}^\perp} \underline{y}. \tag{4.67}$$

If the sequence $\underline{u}$ is presented at the input of an MA filter with coefficients $\underline{a}$ then the output is the residual sequence:

$$\underline{a} * \underline{u} = \mathbf{A} \underline{u} = \mathbf{P}_{\mathbf{A}^\perp} \underline{y} = \hat{\underline{\nu}}. \tag{4.68}$$

Here $*$ means convolution. Since $\underline{u}$ is a minimum dimensional coding of the residual, it corresponds to the *syndrome* in coding theory (see [SMB87]).

Proceeding with the simplification of Equation 4.65 we note that again two terms are complex conjugates and can be combined. Thus

$$D_{\underline{\alpha}} L(\underline{\alpha}) = 2\,\mathrm{Re} \left\{ \underline{u}^H \mathbf{Y} \mathbf{T} - (\underline{u}^H \otimes \underline{u}^T \mathbf{A}^T) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1 \times m+1} \end{bmatrix} \mathbf{T} \right\}. \tag{4.69}$$

The last remaining Kronecker product can be written out and simplified:

$$\begin{aligned} (\underline{u}^H \otimes \underline{u}^T \mathbf{A}^T) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} &= \begin{bmatrix} u_1^* \underline{u}^T \mathbf{A}^T & u_2^* \underline{u}^T \mathbf{A}^T & \cdots & u_p^* \underline{u}^T \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix} \\ &= (\underline{u}^T \mathbf{A}^T \mathbf{I}_n u_1^* + \underline{u}^T \mathbf{A}^T \mathbf{Z}_n u_2^* + \cdots + \underline{u}^T \mathbf{A}^T \mathbf{Z}_n^{p-1} u_p^*) \\ &= \underline{u}^T \mathbf{A}^T (\mathbf{I}_n u_1^* + \mathbf{Z}_n u_2^* + \cdots + \mathbf{Z}_n^{p-1} u_p^*). \end{aligned} \tag{4.70}$$

So the expression for the derivative becomes

$$D_{\underline{\alpha}} L(\underline{\alpha}) = 2\,\mathrm{Re}\left\{ \underline{u}^H \mathbf{Y}\mathbf{T} - \underline{u}^T \mathbf{A}^T \left( \mathbf{I}_n u_1^* + \mathbf{Z}_n u_2^* + \cdots + \mathbf{Z}_n^{p-1} u_p^* \right) \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1 \times m+1} \end{bmatrix} \mathbf{T} \right\}. \quad (4.71)$$

Now define the Toeplitz syndrome matrix $\mathbf{U}$ from the elements $u_i$ of the syndrome in the same manner as $\mathbf{A}$ is defined from $a_i$:

$$\mathbf{U} = \begin{bmatrix} u_p^* & & 0 \\ \vdots & \ddots & \\ u_1^* & & \ddots \\ & \ddots & u_p^* \\ & & \vdots \\ 0 & & u_1^* \end{bmatrix} \in \mathbb{C}^{n \times m+1}. \quad (4.72)$$

This syndrome matrix satisfies the identity

$$\left( \mathbf{I}_n u_1^* + \mathbf{Z}_n u_2^* + \cdots + \mathbf{Z}_n^{p-1} u_p^* \right) \begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1 \times m+1} \end{bmatrix} = \mathbf{J}_n \mathbf{U}, \quad (4.73)$$

allowing further simplification of the derivative:

$$D_{\underline{\alpha}} L(\underline{\alpha}) = 2\,\mathrm{Re}\left\{ \underline{u}^H \mathbf{Y}\mathbf{T} - \underline{u}^T \mathbf{A}^T \mathbf{J}_n \mathbf{U}\mathbf{T} \right\}. \quad (4.74)$$

It can be shown that

$$\underline{u}^T \mathbf{A}^T \mathbf{J}_n \mathbf{U} = \underline{u}^H \mathbf{J}_p \mathbf{A}^T \mathbf{U}^* \mathbf{J}_{m+1}, \quad (4.75)$$

so the derivative can be written

$$\begin{aligned} D_{\underline{\alpha}} L(\underline{\alpha}) &= 2\,\mathrm{Re}\left\{ \underline{u}^H \mathbf{Y}\mathbf{T} - \underline{u}^H \mathbf{J}_p \mathbf{A}^T \mathbf{U}^* \mathbf{J}_{m+1} \mathbf{T} \right\} \\ &= 2\,\mathrm{Re}\left\{ \underline{u}^H (\mathbf{Y} - \mathbf{J}_p \mathbf{A}^T \mathbf{U}^* \mathbf{J}_{m+1}) \mathbf{T} \right\}. \end{aligned} \quad (4.76)$$

The gradient is the transpose of the derivative (under the real-part operator we can take the Hermitian transpose):

$$g(\underline{\alpha}) = \nabla_{\underline{\alpha}} L(\underline{\alpha}) = 2\,\mathrm{Re}\left\{ \mathbf{T}^H (\mathbf{Y}^H - \mathbf{J}_{m+1} \mathbf{U}^T \mathbf{A}^* \mathbf{J}_p) \underline{u} \right\}. \quad (4.77)$$

This expression is a generalization of that of Starer and Nehorai [StN88]. The generalization is from real data and parameters to complex data and parameters. The real case of

our result agrees with their result, taking into account that we have made a slightly different definition of U. But we will now proceed to derive a more elegant expression for the gradient, taking advantage of some fascinating identities.

First note that $\mathbf{J}_p\mathbf{A}^T\mathbf{U}^*\mathbf{J}_{m+1}$ is a Toeplitz matrix of the same form as Y. In fact, its elements are the residuals $\hat{\underline{\nu}}$ defined in Equation 4.67! That is, the residual matrix

$$\hat{\mathbf{N}} = \mathbf{J}_p\mathbf{A}^T\mathbf{U}^*\mathbf{J}_{m+1} \tag{4.78}$$

is built from the elements $\hat{\underline{\nu}}$ in exactly the same way as Y is built from $\underline{y}$ in Equation 4.14. Corresponding to $\hat{\underline{\varepsilon}} = \underline{y} - \hat{\underline{\nu}}$, let us define an estimated signal matrix

$$\hat{\mathbf{X}} = \mathbf{Y} - \hat{\mathbf{N}}. \tag{4.79}$$

Now the expression for the gradient becomes

$$\begin{aligned} \underline{g}(\underline{\alpha}) = \nabla_{\underline{\alpha}} L(\underline{\alpha}) &= 2\,\mathrm{Re}\{\mathbf{T}^H(\mathbf{Y}^H - \hat{\mathbf{N}}^H)\underline{u}\} \\ &= 2\,\mathrm{Re}\{\mathbf{T}^H\hat{\mathbf{X}}^H\underline{u}\}. \end{aligned} \tag{4.80}$$

So the gradient is the convolution of the signal estimate $\hat{\underline{\varepsilon}}$ with the syndrome $\underline{u}$, transformed by the constraint matrix $\mathbf{T}^H$. This filtering interpretation is shown in Figure 4.3. At a maximum of $L(\underline{\alpha})$ the gradient is zero. For the gradient to be zero requires orthogonality between the signal estimate and the syndrome. This orthogonality can be enforced in the time domain or in the frequency domain, using the DTFT. This is another occurrence of the "orthogonality principle" that the error is orthogonal to the signal estimate in least squares problems.

**Hessian.** To implement a Newton algorithm for minimizing $L(\underline{\alpha})$ we also need the second derivative matrix, the Hessian. Our final expression for the gradient implies that the first differential is

$$dL(\underline{\alpha}) = d\underline{\alpha}^T 2\,\mathrm{Re}\{\mathbf{T}^H\hat{\mathbf{X}}^H\underline{u}\}. \tag{4.81}$$

Thus the second differential is

$$\begin{aligned} d^2L(\underline{\alpha}) &= d\underline{\alpha}^T 2\,\mathrm{Re}\{\mathbf{T}^H\hat{\mathbf{X}}^H d\underline{u} + \mathbf{T}^H d(\hat{\mathbf{X}})^H\underline{u}\} \\ &= d\underline{\alpha}^T 2\,\mathrm{Re}\{\mathbf{T}^H\hat{\mathbf{X}}^H d\underline{u} + (\underline{u}^T \otimes \mathbf{T}^H)\mathbf{K}_{p,m+1}\,\mathrm{vec}\,d(\hat{\mathbf{X}})^*\} \end{aligned} \tag{4.82}$$

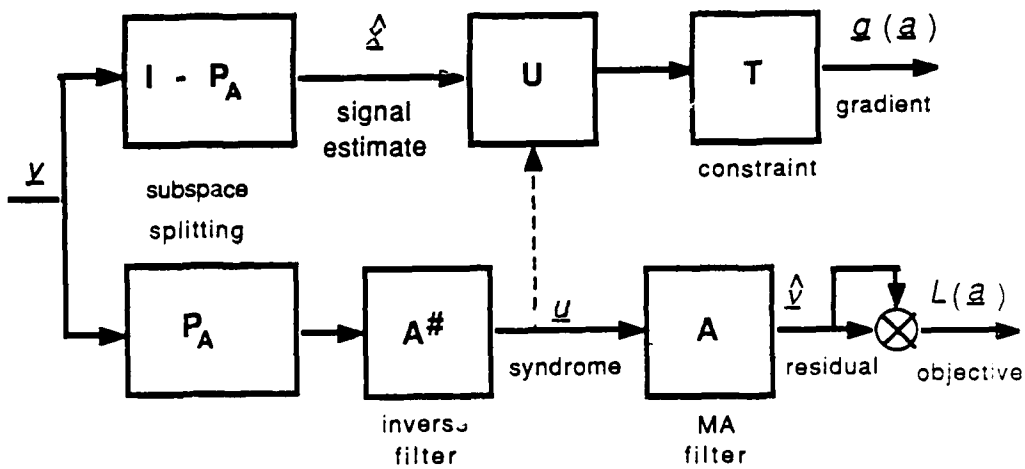


Figure 4.3 Filtering Interpretation of the KiSS Gradient.

So the challenges are to find the differentials of the syndrome and the estimated signal matrix.

Beginning with the syndrome we have

$$\underline{u} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\underline{a},$$

so the differential of the syndrome is

$$\begin{aligned} d\underline{u} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} + d[(\mathbf{A}^H\mathbf{A})^{-1}]\mathbf{Y}\underline{a} \\ &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - (\mathbf{A}^H\mathbf{A})^{-1}d(\mathbf{A}^H\mathbf{A})(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\underline{a} \qquad (4.83) \\ &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - (\underline{u}^T \otimes (\mathbf{A}^H\mathbf{A})^{-1})\,\mathrm{vec}\,d(\mathbf{A}^H\mathbf{A}). \end{aligned}$$

We can simplify this to a fairly nice expression by substituting for $\mathrm{vec}\,d(\mathbf{A}^H\mathbf{A})$ from Equation

4.60, applying Equations 4.51 through 4.53, and using the fact that $\mathbf{K}_{1,p} = \mathbf{I}_p$:

$$\begin{aligned}
d\underline{u} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - \left(\underline{u}^T \otimes (\mathbf{A}^H\mathbf{A})^{-1}\right)\left(\mathbf{I}_p \otimes \mathbf{A}^H\right)\mathrm{vec}(d\mathbf{A}) - \\
&\quad \left(\underline{u}^T \otimes (\mathbf{A}^H\mathbf{A})^{-1}\right)\mathbf{K}_{p,p}\left(\mathbf{I}_p \otimes \mathbf{A}^T\right)\mathrm{vec}(d\mathbf{A}) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - \left(\underline{u}^T \otimes \mathbf{A}^{\#}\right)\mathrm{vec}(d\mathbf{A}) - \mathbf{K}_{1,p}\left((\mathbf{A}^H\mathbf{A})^{-1} \otimes \underline{u}^T\right)\left(\mathbf{I}_p \otimes \mathbf{A}^T\right)\mathrm{vec}(d\mathbf{A}) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - \begin{bmatrix} u_1\mathbf{A}^{\#} & u_2\mathbf{A}^{\#} & \cdots & u_p\mathbf{A}^{\#} \end{bmatrix}\begin{bmatrix} \mathbf{I}_n \\ \mathbf{Z}_n \\ \vdots \\ \mathbf{Z}_n^{p-1} \end{bmatrix}\begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix}\mathbf{T}^* d\underline{\alpha} - \\
&\quad \left((\mathbf{A}^H\mathbf{A})^{-1} \otimes \underline{u}^T\mathbf{A}^T\right)\mathrm{vec}(d\mathbf{A}) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - \mathbf{A}^{\#}\left(u_1\mathbf{I}_n + u_2\mathbf{Z}_n + \cdots + u_p\mathbf{Z}_n^{p-1}\right)\begin{bmatrix} \mathbf{J}_{m+1} \\ \mathbf{0}_{p-1\times m+1} \end{bmatrix}\mathbf{T}^* d\underline{\alpha} - \\
&\quad \left((\mathbf{A}^H\mathbf{A})^{-1} \otimes \underline{\hat{\nu}}^T\right)\mathrm{vec}(d\mathbf{A}) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - \mathbf{A}^{\#}\mathbf{J}_n\mathbf{U}^*\mathbf{T}^* d\underline{\alpha} - \mathrm{vec}\left(\underline{\hat{\nu}}^T(d\mathbf{A}^*)[(\mathbf{A}^H\mathbf{A})^{-1}]^T\right) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - (\mathbf{A}^H\mathbf{A})^{-1}\tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^* d\underline{\alpha} - (\mathbf{A}^H\mathbf{A})^{-1}(d\mathbf{A}^H)\underline{\hat{\nu}},
\end{aligned} \tag{4.84}$$

where $\tilde{\mathbf{N}}$, introduced in the last line, is a Toeplitz matrix of backward residuals

$$\underline{\tilde{\nu}} = \mathbf{J}_n\mathbf{A}^*\mathbf{J}_p\underline{u}, \tag{4.85}$$

analogous to $\hat{\mathbf{N}}$ for forward residuals $\underline{\hat{\nu}}$. The third term in Equation 4.84 for $d\underline{u}$ may be simplified further by the fact that

$$(d\mathbf{A}^H)\underline{\hat{\nu}} = \hat{\mathbf{N}}d\underline{\alpha}. \tag{4.86}$$

This identity is analogous to the commutativity of convolution, and to the switch already used between $\mathbf{A}^H\underline{y}$ and $\mathbf{Y}\underline{a}$. Equation 4.84 then simplifies to

$$\begin{aligned}
d\underline{u} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Y}\mathbf{T}d\underline{\alpha} - (\mathbf{A}^H\mathbf{A})^{-1}\tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^* d\underline{\alpha} - (\mathbf{A}^H\mathbf{A})^{-1}\hat{\mathbf{N}}\mathbf{T}d\underline{\alpha} \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\left(\mathbf{Y}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^* - \hat{\mathbf{N}}\mathbf{T}\right)d\underline{\alpha} \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\left(\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^*\right)d\underline{\alpha}.
\end{aligned} \tag{4.87}$$

This is the differential of the syndrome. Now we can express the vectorization of the signal estimate matrix as

$$\mathrm{vec}\,\hat{\mathbf{X}} = \begin{bmatrix} \begin{bmatrix}\mathbf{0}_{p\times m} & \mathbf{I}_p\end{bmatrix}\mathbf{I}_n \\ \begin{bmatrix}\mathbf{0}_{p\times m} & \mathbf{I}_p\end{bmatrix}\mathbf{Z}_n^{-1} \\ \vdots \\ \begin{bmatrix}\mathbf{0}_{p\times m} & \mathbf{I}_p\end{bmatrix}\mathbf{Z}_n^{-m} \end{bmatrix}\underline{\hat{\epsilon}}. \tag{4.88}$$

Since $\hat{\underline{\varepsilon}} = \underline{y} - \mathbf{A}\underline{u}$, we have

$$\begin{aligned}
\operatorname{vec} d\hat{\mathbf{X}} &= -\begin{bmatrix} [0_{p\times m} \quad I_p]I_n \\ [0_{p\times m} \quad I_p]Z_n^{-1} \\ \vdots \\ [0_{p\times m} \quad I_p]Z_n^{-m} \end{bmatrix} [(d\mathbf{A})\underline{u} + \mathbf{A}d\underline{u} \\
&= -\begin{bmatrix} [0_{p\times m} \quad I_p]I_n \\ [0_{p\times m} \quad I_p]Z_n^{-1} \\ \vdots \\ [0_{p\times m} \quad I_p]Z_n^{-m} \end{bmatrix} [(\underline{u}^T \otimes I_n)\operatorname{vec}(d\mathbf{A}) + \mathbf{A}d\underline{u}] \\
&= -\begin{bmatrix} [0_{p\times m} \quad I_p]I_n \\ [0_{p\times m} \quad I_p]Z_n^{-1} \\ \vdots \\ [0_{p\times m} \quad I_p]Z_n^{-m} \end{bmatrix} \left\{ (\underline{u}^T \otimes I_n) \begin{bmatrix} I_n \\ Z_n \\ \vdots \\ Z_n^{p-1} \end{bmatrix} \begin{bmatrix} J_{m+1} \\ 0_{p-1\times m+1} \end{bmatrix} \mathbf{T}^* + \mathbf{A}^{\#H}[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}J_{m+1}\mathbf{T}^*] \right\} d\underline{\alpha} \\
&= -\begin{bmatrix} [0_{p\times m} \quad I_p]I_n \\ [0_{p\times m} \quad I_p]Z_n^{-1} \\ \vdots \\ [0_{p\times m} \quad I_p]Z_n^{-m} \end{bmatrix} \left\{ J_n\mathbf{U}^*\mathbf{T}^* + \mathbf{A}^{\#H}[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}J_{m+1}\mathbf{T}^*] \right\} d\underline{\alpha}.
\end{aligned} \tag{4.89}$$

Substituting these results into Equation 4.82 yields

$$\begin{aligned}
d^2L(\underline{\alpha}) = d\underline{\alpha}^T 2\,\mathrm{Re}\Bigg\{ &\mathbf{T}^H\hat{\mathbf{X}}^H(\mathbf{A}^H\mathbf{A})^{-1}[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}J_{m+1}\mathbf{T}^*] - \\
&(\underline{u}^T \otimes \mathbf{T}^H)K_{p,m+1} \begin{bmatrix} [0_{p\times m} \quad I_p]I_n \\ [0_{p\times m} \quad I_p]Z_n^{-1} \\ \vdots \\ [0_{p\times m} \quad I_p]Z_n^{-m} \end{bmatrix} \left( J_n\mathbf{U}\mathbf{T} + \mathbf{A}^{\#T}[\hat{\mathbf{X}}^*\mathbf{T}^* - \tilde{\mathbf{N}}^*J_{m+1}\mathbf{T}] \right) \Bigg\} d\underline{\alpha}.
\end{aligned} \tag{4.90}$$

The effect of premultiplication by $K_{p,m+1}$ is to reorder the rows of the multiplied matrix.

The result in this case can be expressed as

$$d^2L(\underline{\alpha}) = d\underline{\alpha}^T 2\,\mathrm{Re}\Bigg\{ \mathbf{T}^H\hat{\mathbf{X}}^H(\mathbf{A}^H\mathbf{A})^{-1}[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^*] - [\,u_1\mathbf{T}^H \quad u_2\mathbf{T}^H \quad \cdots \quad u_p\mathbf{T}^H\,] \cdot$$

$$\begin{bmatrix} [\mathbf{J}_{m+1} \quad \mathbf{0}_{m+1\times p-1}] \\ [\mathbf{0}_{m+1\times 1} \quad \mathbf{J}_{m+1} \quad \mathbf{0}_{m+1\times p-2}] \\ \vdots \\ [\mathbf{0}_{m+1\times p-1} \quad \mathbf{J}_{m+1}] \end{bmatrix} \Big(\mathbf{J}_n\mathbf{U}\mathbf{T} + \mathbf{A}^{\#T}[\hat{\mathbf{X}}^*\mathbf{T}^* - \tilde{\mathbf{N}}^*\mathbf{J}_{m+1}\mathbf{T}]\Big)\Bigg\} d\underline{\alpha}$$

$$= d\underline{\alpha}^T 2\,\mathrm{Re}\Bigg\{ \mathbf{T}^H\hat{\mathbf{X}}^H(\mathbf{A}^H\mathbf{A})^{-1}[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^*] -$$

$$\mathbf{T}^H\mathbf{U}^H\mathbf{J}_n\Big(\mathbf{J}_n\mathbf{U}\mathbf{T} + \mathbf{A}^*[(\mathbf{A}^H\mathbf{A})^{-1}]^*[\hat{\mathbf{X}}^*\mathbf{T}^* - \tilde{\mathbf{N}}^*\mathbf{J}_{m+1}\mathbf{T}]\Big)\Bigg\} d\underline{\alpha}$$

$$= d\underline{\alpha}^T 2\,\mathrm{Re}\Bigg\{ \mathbf{T}^H\hat{\mathbf{X}}^H(\mathbf{A}^H\mathbf{A})^{-1}\hat{\mathbf{X}}\mathbf{T} - \mathbf{T}^H\hat{\mathbf{X}}^H(\mathbf{A}^H\mathbf{A})^{-1}\tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^* -$$

$$\mathbf{T}^H\mathbf{U}^H\mathbf{U}\mathbf{T} - \mathbf{T}^H\mathbf{J}_{m+1}\tilde{\mathbf{N}}^T[(\mathbf{A}^H\mathbf{A})^{-1}]^*\hat{\mathbf{X}}^*\mathbf{T}^* + \mathbf{T}^H\mathbf{J}_{m+1}\tilde{\mathbf{N}}^T[(\mathbf{A}^H\mathbf{A})^{-1}]^*\tilde{\mathbf{N}}^*\mathbf{J}_{m+1}\mathbf{T}\Bigg\} d\underline{\alpha}. \tag{4.91}$$

Because of the real-part operator, we can take the conjugate of the last two terms and then factor the expression as

$$d^2L(\underline{\alpha}) = d\underline{\alpha}^T 2\,\mathrm{Re}\Big\{\Big[\mathbf{T}^H\hat{\mathbf{X}}^H - \mathbf{T}^T\mathbf{J}_{m+1}\tilde{\mathbf{N}}^H\Big](\mathbf{A}^H\mathbf{A})^{-1}\Big[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^*\Big] - \mathbf{T}^H\mathbf{U}^H\mathbf{U}\mathbf{T}\Big\} d\underline{\alpha}. \tag{4.92}$$

Let $\mathbf{S}$ denote the matrix

$$\mathbf{S} = \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\Big[\hat{\mathbf{X}}\mathbf{T} - \tilde{\mathbf{N}}\mathbf{J}_{m+1}\mathbf{T}^*\Big]. \tag{4.93}$$

Then the final expression for the second differential is

$$d^2L(\underline{\alpha}) = d\underline{\alpha}^T 2\,\mathrm{Re}\left\{\mathbf{S}^H\mathbf{S} - \mathbf{T}^H\mathbf{U}^H\mathbf{U}\mathbf{T}\right\} d\underline{\alpha}. \tag{4.94}$$

The "second identification theorem" of Magnus and Neudecker allows us to identify the Hessian from Equation 4.94 as

$$\mathbf{H}(\underline{\alpha}) = \nabla^2_{\underline{\alpha}}L(\underline{\alpha}) = 2\,\mathrm{Re}\left\{\mathbf{S}^H\mathbf{S} - \mathbf{T}^H\mathbf{U}^H\mathbf{U}\mathbf{T}\right\} \tag{4.95}$$

The filtering interpretation of the Hessian is shown in Figure 4.4. The data $\underline{x}$ excites the inverse filter $\mathbf{A}^{\#}$ to produce the syndrome $\underline{u}$. The syndrome is forward and backward

filtered by the MA filter **A** to produce the forward and backward residuals $\hat{\underline{\nu}}$ and *ntwiddle*. The constraint **T** is applied to the data $\underline{y}$ and to the residuals $\hat{\underline{\nu}}$ and $\tilde{\underline{\nu}}$, and the resuduals are subtracted from the data. The difference vector used to form a Toeplitz matrix and filtered again by the inverse filter $\mathbf{A}^{\#}$ to create the matrix **S**. In the other branch of the figure, the syndrome $\underline{u}$ is used to form the Toeplitz matrix **U** to which the constraint **T** is applied. The Hessian is then formed by summing the Grammians of **S** and **UT**.

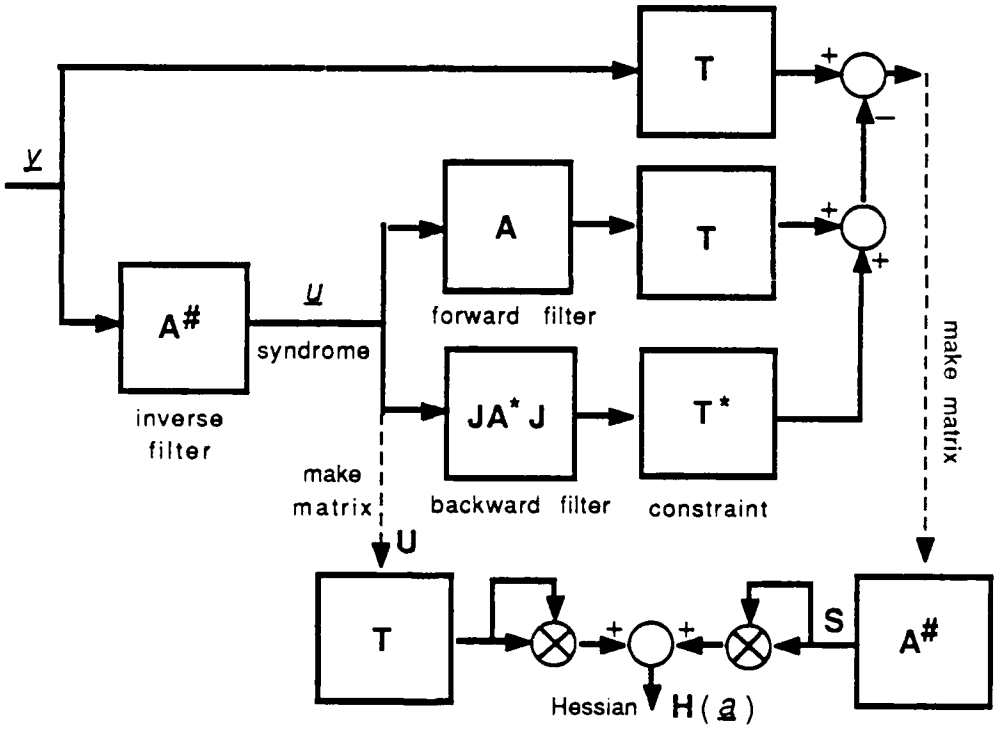


Figure 4.4 Filtering Interpretation of the KiSS Hessian.

**Newton step.** We can now implement a Newton method for the minimization of $L(\underline{\alpha})$. For any current iteration $\underline{\alpha}_i$, the Newton step is to subtract the inverse of the Hessian

times the gradient:

$$\underline{\alpha}_{i+1} = \underline{\alpha}_i - \mathrm{H}(\underline{\alpha}_i)^{-1} g(\underline{\alpha}_i). \tag{4.96}$$

**Performance of KiSS and Newton.** We have run simulations to test the performance of the KiSS algorithm with and without a Newton method as a second phase. We have used the same test signal for these tests as used in [KSS86] and others. The signal is $n = 25$ samples of

$$x(t) = e^{j\omega_1 t} + e^{j\frac{\pi}{4}} e^{j\omega_2 t}, \tag{4.97}$$

with

$$\begin{aligned} \omega_1 &= 2\pi(0.52), \\ \omega_2 &= 2\pi(0.50). \end{aligned} \tag{4.98}$$

Noise of variance $\sigma^2$ is added to each of the real and imaginary parts. The signal to noise ratio (SNR) is defined as

$$\mathrm{SNR} = 10\log_{10}(\frac{1}{2\sigma^2}). \tag{4.99}$$

At each of several signal to noise ratios, we ran 500 trials. For each trial the KiSS algorithm (phase 1) was run on the data with a new realization of the noise. The Newton-based phase 2 was started from where KiSS terminated. The results of the KiSS algorithm alone were compared with the results after phase 2. Figure 4.5 shows the performance as $-10\log_{10}(\mathrm{MSE})$ versus SNR. You can see that the Newton method is of limited value when it follows convergence of KiSS, improving performance above the threshold SNR of 9 dB only slightly and actually degrading the already bad performance below the threshold SNR. The Newton method may be more useful if used after fewer iterations of KiSS, or from some other starting point.

## 4.3 KiSS with Structured Noise

We now consider the problem of estimating structurally constrained signal subspaces in the presence of structured noise. We extend the KiSS algorithm to account for structured noise with a known subspace of dimension $t$. If the structured noise subspace is not known, but corresponds to impulse noise, then we can perform a search over the possible structured

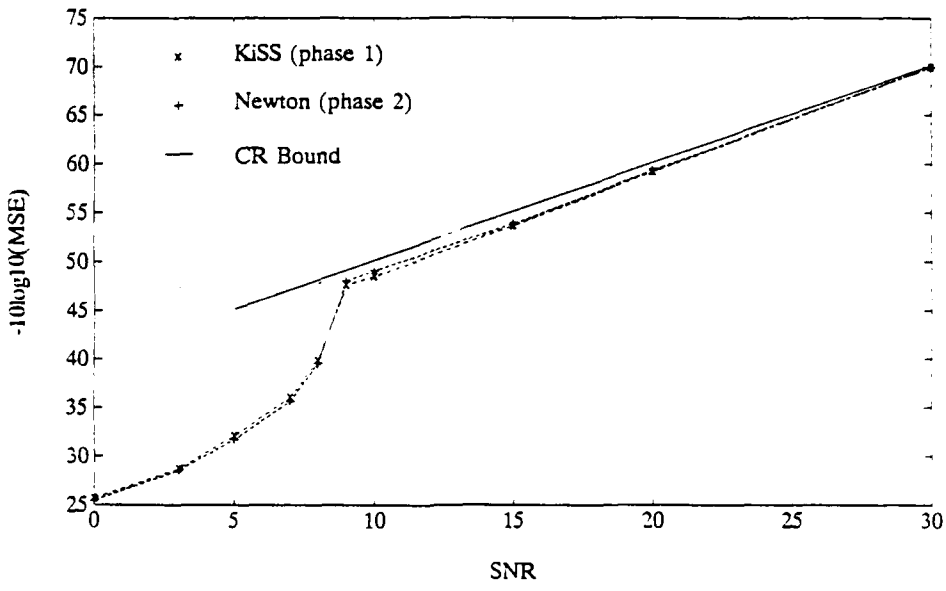


Figure 4.5 Performance of KiSS with and without a Newton phase 2.

noise subspaces, performing the extended KiSS algorithm for each one and choosing the one that minimizes the objective function. This outer optimization search corresponds exactly to the one described in Section 3.4 for the same problem without the complex exponential signal model. As in the earlier problem, we need to include an order penalty in the objective function if the rank $t$ is unknown.

**Signal model.** Let the signal be a sum of complex exponentials as in Section 4.1:

$$x(t) = \sum_{i=1}^{m} \theta_i z_i^t. \tag{4.100}$$

Suppose the observed data consists of signal $x(t)$, plus structured noise $b(t)$, plus background noise $\nu(t)$ for $n$ consecutive time indices:

$$y(t) = x(t) + b(t) + \nu(t), \qquad t = 1, 2, \ldots, n. \tag{4.101}$$

The matrix-vector formulation is

$$\underline{y} = \mathbf{H}\underline{\theta} + \mathbf{S}\underline{\phi} + \underline{\nu}, \tag{4.102}$$

where the signal is $\underline{x} = \mathbf{H}\underline{\theta}$ with

$$\underline{x} = \begin{bmatrix} x(1) \\ \vdots \\ x(n) \end{bmatrix} \in \mathbb{C}^n, \qquad \mathbf{H} = \begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_1^2 & z_2^2 & \cdots & z_m^2 \\ \vdots & \vdots & & \vdots \\ z_1^n & z_2^n & \cdots & z_m^n \end{bmatrix} \in \mathbb{C}^{n \times m}, \qquad \underline{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} \in \mathbb{C}^m. \tag{4.103}$$

The structured noise is $\underline{b} = \mathbf{S}\underline{\phi}$ with

$$\underline{b} = \begin{bmatrix} b(1) \\ \vdots \\ b(n) \end{bmatrix} \in \mathbb{C}^n, \qquad \mathbf{S} \in \mathbb{C}^{n \times t}, \qquad \underline{\phi} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_m \end{bmatrix} \in \mathbb{C}^t. \tag{4.104}$$

The background noise is $\underline{\nu} \in \mathbb{C}^n$.

**Objective function.** Given that the background noise is zero-mean white Gaussian noise we could derive the log-likelihood function and maximize it to obtain the ML signal subspace estimate. But this time we will take the equivalent least squares approach. The modeling error is the difference between the observed data and the two part model (signal plus structured noise):

$$\begin{aligned} \underline{e} &= \underline{y} - (\hat{\mathbf{H}}\hat{\underline{\theta}} + \hat{\mathbf{S}}\hat{\underline{\phi}}) \\ &= \underline{y} - \begin{bmatrix} \hat{\mathbf{H}} & \hat{\mathbf{S}} \end{bmatrix} \begin{bmatrix} \hat{\underline{\theta}} \\ \hat{\underline{\phi}} \end{bmatrix}. \end{aligned} \tag{4.105}$$

The least squares objective function is

$$e^2 = \underline{e}^H \underline{e}. \tag{4.106}$$

The last form of Equation 4.105 leads to the following form of the objective function when optimum values of $\hat{\underline{\theta}}$ and $\hat{\underline{\phi}}$ have been chosen (more detail is given in the chapter on parameter estimation):

$$e^2 = \underline{y}^H (\mathbf{I} - \mathbf{P}_{\mathbf{HS}}) \underline{y}, \tag{4.107}$$

where

$$\mathbf{P}_{\mathbf{HS}} = [\mathbf{H}\ \mathbf{S}] \left( [\mathbf{H}\ \mathbf{S}]^H [\mathbf{H}\ \mathbf{S}] \right)^{-1} [\mathbf{H}\ \mathbf{S}]^H \tag{4.108}$$

**Derivation of a KiSS type algorithm.** A key step in the derivation of the KiSS

algorithm was the switch from $\mathbf{H}$ to the Toeplitz matrix $\mathbf{A}$ spanning the orthogonal complement of the signal subspace. By rewriting the objective function of Equation 4.107 we will eventually be able to make the same switch.

$$\begin{aligned} e^2 &= \underline{y}^H\underline{y} - \underline{y}^H[\mathbf{H}\ \mathbf{S}]\left([\mathbf{H}\ \mathbf{S}]^H[\mathbf{H}\ \mathbf{S}]\right)^{-1}[\mathbf{H}\ \mathbf{S}]^H\underline{y} \\ &= \underline{y}^H\underline{y} - [\underline{y}^H\mathbf{H}\ \underline{y}^H\mathbf{S}]\begin{bmatrix}\mathbf{H}^H\mathbf{H} & \mathbf{H}^H\mathbf{S}\\ \mathbf{S}^H\mathbf{H} & \mathbf{S}^H\mathbf{S}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{H}^H\underline{y}\\ \mathbf{S}^H\underline{y}\end{bmatrix}. \end{aligned} \tag{4.109}$$

The inverse can be found by the block matrix inversion formula [Kai80]:

$$\begin{aligned} &\begin{bmatrix}\mathbf{H}^H\mathbf{H} & \mathbf{H}^H\mathbf{S}\\ \mathbf{S}^H\mathbf{H} & \mathbf{S}^H\mathbf{S}\end{bmatrix}^{-1} = \\ &\begin{bmatrix}(\mathbf{H}^H\mathbf{H})^{-1} + \mathbf{H}^{\#}\mathbf{S}\left[\mathbf{S}^H\mathbf{S} - \mathbf{S}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\right]^{-1}\mathbf{S}^H(\mathbf{H}^{\#})^H & -\mathbf{H}^{\#}\mathbf{S}\left[\mathbf{S}^H\mathbf{S} - \mathbf{S}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\right]^{-1}\\ -\left[\mathbf{S}^H\mathbf{S} - \mathbf{S}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\right]^{-1}\mathbf{S}^H(\mathbf{H}^{\#})^H & \left[\mathbf{S}^H\mathbf{S} - \mathbf{S}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\right]^{-1}\end{bmatrix}, \end{aligned} \tag{4.110}$$

so

$$\begin{aligned} e^2 = {} & \underline{y}^H\underline{y} - \underline{y}^H\mathbf{P}_{\mathbf{H}}\underline{y} - \underline{y}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\left[\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{H}})\mathbf{S}\right]^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}}\underline{y} + \underline{y}^H\mathbf{P}_{\mathbf{H}}\mathbf{S}\left[\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{H}})\mathbf{S}\right]^{-1}\mathbf{S}^H\underline{y} + \\ & \underline{y}^H\mathbf{S}\left[\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{H}})\mathbf{S}\right]^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{H}}\underline{y} - \underline{y}^H\mathbf{S}\left[\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{H}})\mathbf{S}\right]^{-1}\mathbf{S}^H\underline{y}. \end{aligned} \tag{4.111}$$

Now make the switch by replacing all occurrences of $\mathbf{P}_{\mathbf{H}}$ with $\mathbf{I} - \mathbf{P}_{\mathbf{A}}$:

$$\begin{aligned} e^2 = {} & \underline{y}^H\underline{y} - \underline{y}^H(\mathbf{I}-\mathbf{P}_{\mathbf{A}})\underline{y} - \underline{y}^H(\mathbf{I}-\mathbf{P}_{\mathbf{A}})\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{A}})\underline{y} + \\ & \underline{y}^H(\mathbf{I}-\mathbf{P}_{\mathbf{A}})\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H\underline{y} + \underline{y}^H\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H(\mathbf{I}-\mathbf{P}_{\mathbf{A}})\underline{y} - \underline{y}^H\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H\underline{y} \\ = {} & \underline{y}^H\mathbf{P}_{\mathbf{A}}\underline{y} - \underline{y}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\underline{y}. \end{aligned} \tag{4.112}$$

Then make the substitution $\mathbf{P}_{\mathbf{A}} = \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$ and factor as

$$e^2 = \underline{y}^H\mathbf{A}\left[(\mathbf{A}^H\mathbf{A})^{-1} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{S}\left[\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right]^{-1}\mathbf{S}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\right]\mathbf{A}^H\underline{y}. \tag{4.113}$$

Finally, taking advantage of the identity $\mathbf{A}^H\underline{y} = \mathbf{Y}\underline{a}$ we can write the objective function as

$$e^2 = \underline{a}^H\mathbf{Y}^H\left[(\mathbf{A}^H\mathbf{A})^{-1} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{S}\left(\mathbf{S}^H\mathbf{P}_{\mathbf{A}}\mathbf{S}\right)^{-1}\mathbf{S}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\right]\mathbf{Y}\underline{a}. \tag{4.114}$$

The form of Equation 4.114 is similar to the form of the KiSS objective function in Equation 4.15. The difference is the "correction term" subtracted from $(\mathbf{A}^H\mathbf{A})^{-1}$ which accounts for the structured noise portion of the model. The new objective function is, in fact,

invariant to all structured noise in the subspace $\langle \mathbf{S} \rangle$.

We assume that $\mathbf{S}$ is fixed (either known or selected as a candidate in an outer optimization) and try to minimize Equation 4.114 with respect to $\underline{a}$. We propose an algorithm to minimize $e^2$ that corresponds to the KiSS algorithm: Hold the expression in square brackets in Equation 4.114 fixed at each iteration and solve the resulting quadratic minimization problem for the new $\underline{a}$, then use that value of $\underline{a}$ to update the expression in square brackets for the next iteration. Before the first iteration the expression in square brackets is set to $\mathbf{I} - \mathbf{P_S}$. Constraints on $\underline{a}$ may be handled in the same way as for the KiSS algorithm.

**The KiSS algorithm with structured noise.** With the affine constraint of Equation 4.16, the objective function of Equation 4.114 becomes

$$e^2 = (\underline{c}^H + \underline{\alpha}^H \mathbf{T}^H)\mathbf{Y}^H \left[(\mathbf{A}^H\mathbf{A})^{-1} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{S}\left(\mathbf{S}^H\mathbf{P_A}\mathbf{S}\right)^{-1}\mathbf{S}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\right]\mathbf{Y}(\mathbf{T}\underline{\alpha} + \underline{c}). \tag{4.115}$$

The only difference from the KiSS algorithm without structured noise is the matrix $\mathbf{Q}_i$ that we fix for each iteration:

$$\mathbf{Q}_i = \mathbf{Y}^H \left[(\mathbf{A}^H\mathbf{A})^{-1} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{S}\left(\mathbf{S}^H\mathbf{P_A}\mathbf{S}\right)^{-1}\mathbf{S}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\right]\mathbf{Y}. \tag{4.116}$$

The explicit steps of the constrained KiSS algorithm with structured noise are

1. Build appropriate $\mathbf{Y}, \mathbf{T}, \underline{c}$.
2. Set $\mathbf{Q} = \mathbf{Y}^H(\mathbf{I} - \mathbf{P_S})\mathbf{Y}$.
3. Repeat until convergence:

3a. Let $\underline{\alpha} = -[\mathrm{Re}(\mathbf{T}^H\mathbf{Q}\mathbf{T})]^{-1}\,\mathrm{Re}(\mathbf{T}^H\mathbf{Q}\underline{c})$.

3b. Let $\underline{a} = \mathbf{T}\underline{\alpha} + \underline{c}$.

3c. Build $\mathbf{A}$ from $\underline{a}$.

3d. Let $\mathbf{Q} = \mathbf{Y}^H \left[(\mathbf{A}^H\mathbf{A})^{-1} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{S}\left(\mathbf{S}^H\mathbf{P_A}\mathbf{S}\right)^{-1}\mathbf{S}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\right]\mathbf{Y}$.

4. Stop.

Convergence is tested in the same way as the original KiSS algorithm.

**Example: Second order KiSS objective function with structured noise.** To

illustrate the difference between the original KiSS objective function and this one which accounts for structured noise, we return to the second order example of Figure 4.1 and Figure 4.2. When substantial structured noise is added to the signal, the surface is greatly distorted as shown in Figure 4.6. However, when the modified KiSS objective function is used, accounting for the structured noise, the surface returns to its original characteristic shape as shown in Figure 4.7.

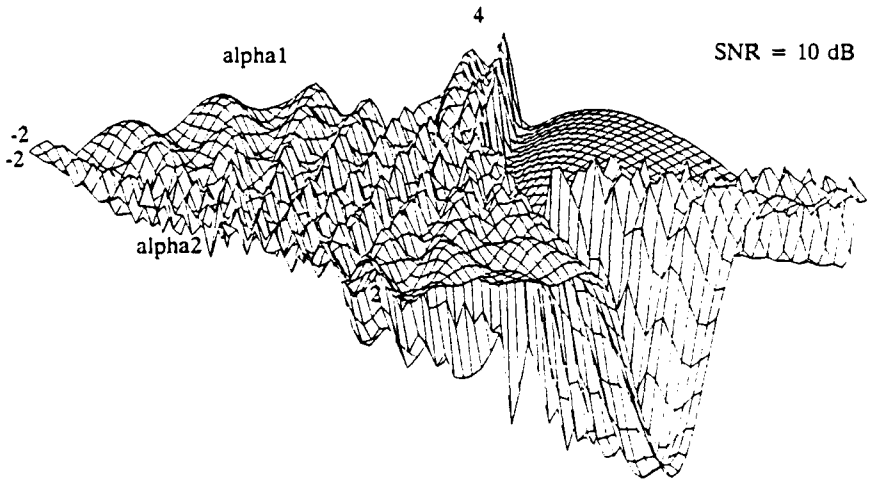


Figure 4.6 KiSS objective function corrupted by structured noise.

**Simulation Results.**

Simulations of the KiSS algorithm modified for structured noise have not been extensive because of the high computational cost involved. Enough simulations have been run, however, to make the following observations. The algorithm converged in all the trials (over 30,000) whether the structured noise matrix was correct or not. When given the correct struc-

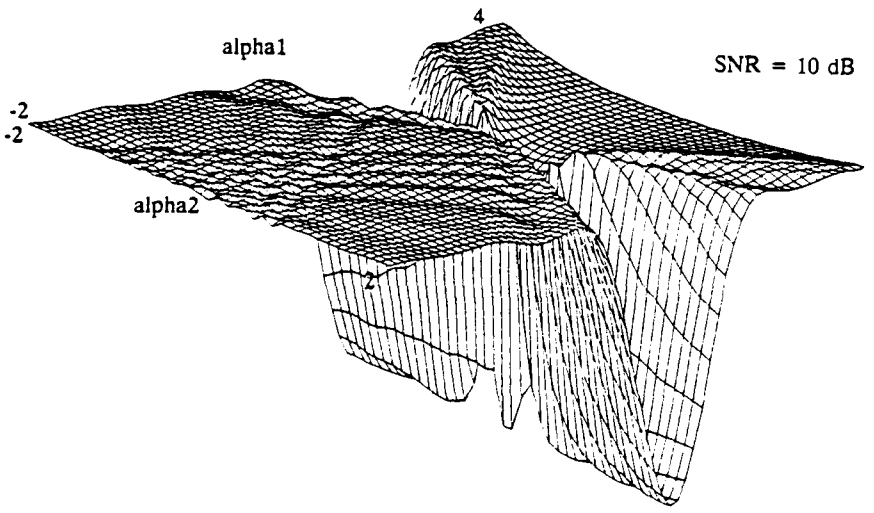


Figure 4.7 KiSS objective function with structured noise accounted for.

tured noise matrix, the frequency estimation performance (measured by mean squared error in $f_1$ or $f_2$) of the algorithm was essentially the same as the regular KiSS algorithm in the absence of structured noise. In contrast, the use of the regular KiSS algorithm when structured noise was present resulted in very poor estimator performance, as one would expect considering Figure 4.6.

The main set of tests was conducted as follows. The test signal was the same as the one we used for the KiSS algorithm, with 25 samples of two complex exponentials. The background noise was added in the same way as for the regular KiSS tests, and the SNR computed the same way. We added a fixed structured noise vector to the signal and background noise, with impulses in positions 2 and 23 whose values were $-5+7j$ and $4-3j$ respectively. Leaving out the

order selection aspect of the problem, we assumed knowledge of $t = 2$ impulses and conducted a combinatorial search over the $\binom{25}{2}$ possible locations for the pair of impulses. For each candidate $\mathbf{S}$ we applied the structured noise version of the KiSS algorithm until it converged. We then chose as our estimate of $\mathbf{S}$ the candidate for which the smallest value of the objective function had been obtained, and took the corresponding estimate of the AR parameters $\underline{a}$.

To be blunt, this method of identifying the structured noise subspace did not work. The correct subspace was chosen zero times out of 50 background noise realizations at 20 dB SNR, and zero times out of 50 even with very clean data at 50 dB SNR. The frequency estimation performance was also very bad, since the structured noise was not successfully removed from the data. We believe the algorithm was coded correctly, although a program bug is not impossible as the explanation. Our conclusion is therefore that this is not a good way to identify the structured noise subspace. We observe that a better way to identify the structured noise subspace might be found by comparing Figures 4.6 and 4.7. From these figures it appears that the error surface is "smoother" when the structured noise has been correctly accounted for. If this observation could be quantified it might lead to a better identification technique for simultaneously identifying complex exponentials and impulse noise. We leave this idea as a suggested extension of this work.

# CHAPTER V

## Order Selection for Subspace Identification

In this chapter we consider the problem of selecting the appropriate dimensionality of a subspace. We consider two kinds of subspace order selection problems. In the first, there is a known prior subspace model, but it may be beneficial to reduce the rank of that model because of the presence of noise. This is a typical rank reduction problem, where we trade off model bias and variance to minimize mean squared error. In the other order selection problem, we have no known prior subspace model. Instead we are trying to identify the subspace from the data as in Chapters III and IV. It is not clear that the bias versus variance tradeoff applies, since we have no model against which to measure bias.

### 5.1 Rank Reduction of a Prior Signal Subspace Model

The purpose of rank reduction is to reduce the Mean Squared Error (MSE) of an estimator. We measure the error in the observation space, noting that it could also be measured in the parameter space. Since the actual Squared Error (SE) is unknown, one class of order selection rules involves making an estimate of the SE at each rank, and choosing the rank for which this estimate is smallest. The order selection rule of [ScS86] is of this type, as is our new approach. Other methods of order selection in similar situations are given in [Aka74], [SMB87], [Mar87], [Kum85].

We begin with the linear statistical model of complex data $\underline{y}$,

$$\underline{y} = \underline{x} + \underline{\nu}$$

$$\underbrace{\begin{bmatrix} \\ \underline{y} \\ \\ \end{bmatrix}}_{n} = \overbrace{\underbrace{\begin{bmatrix} \\ \mathbf{H} \\ \\ \end{bmatrix}}_{n \times m} \underbrace{\begin{bmatrix} \\ \underline{\theta} \\ \\ \end{bmatrix}}_{m}}^{} + \underbrace{\begin{bmatrix} \\ \underline{\nu} \\ \\ \end{bmatrix}}_{n} \tag{5.1}$$

In this equation, H is a known system matrix which spans the $m$-dimensional signal subspace ($m < n$), $\underline{\theta}$ is the parameter vector and $\underline{x} = \mathbf{H}\underline{\theta}$ is the signal. The additive noise, $\underline{\nu}$, is zero mean, white, and Gaussian, with independent real and imaginary parts:

$$\begin{aligned} \mathrm{Re}(\underline{\nu}) &: \ \mathrm{N}(\underline{0}, \frac{1}{2}\sigma^2\mathbf{I}) \\ \mathrm{Im}(\underline{\nu}) &: \ \mathrm{N}(\underline{0}, \frac{1}{2}\sigma^2\mathbf{I}). \end{aligned} \tag{5.2}$$

At the receiver both the signal $\underline{x}$ and the parameter $\underline{\theta}$ are unknown, and a signal estimate is desired. The maximum likelihood estimate of $\underline{x}$ is the least squares solution obtained by projecting $\underline{y}$ onto the signal subspace. We shall denote this signal estimate by $\underline{x}_m$ since it lies in the $m$-dimensional signal subspace:

$$\underline{x}_m = \mathbf{P}_\mathbf{H}\underline{y}, \tag{5.3}$$

where

$$\mathbf{P}_\mathbf{H} = \mathbf{H}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H. \tag{5.4}$$

The estimator $\underline{x}_m$ is unbiased, and has variance $m\sigma^2$. A reduced rank estimator is formed when we replace $\mathbf{P}_\mathbf{H}$ in Equation 5.3 with a lower rank projection $\mathbf{P}_r$ ($r < m$), whose range is contained in the range of $\mathbf{P}_\mathbf{H}$:

$$\underline{x}_r = \mathbf{P}_r\underline{y}. \tag{5.5}$$

Using $\mathbf{P}_r$ reduces the variance of the estimate from $m\sigma^2$ to $r\sigma^2$, but introduces bias $\underline{b}_r$ given by

$$\underline{b}_r = (\mathbf{I} - \mathbf{P}_r)\underline{x} = (\mathbf{P}_\mathbf{H} - \mathbf{P}_r)\underline{x}. \tag{5.6}$$

Figure 5.1 shows how the error can be decomposed into two orthogonal components, one due to noise variance and the other due to the bias introduced by rank reduction. Because of orthogonality, the squared error can be estimated as the sum of the variance and the squared magnitude of the bias. Since the variance is known, we need an estimate of the squared magnitude of the bias, $b_r^2 = \underline{b}_r^H\underline{b}_r$. In the following pages we review several estimators of $b_r^2$ and introduce a new estimator based on two sequential applications of the principle of maximum likelihood.

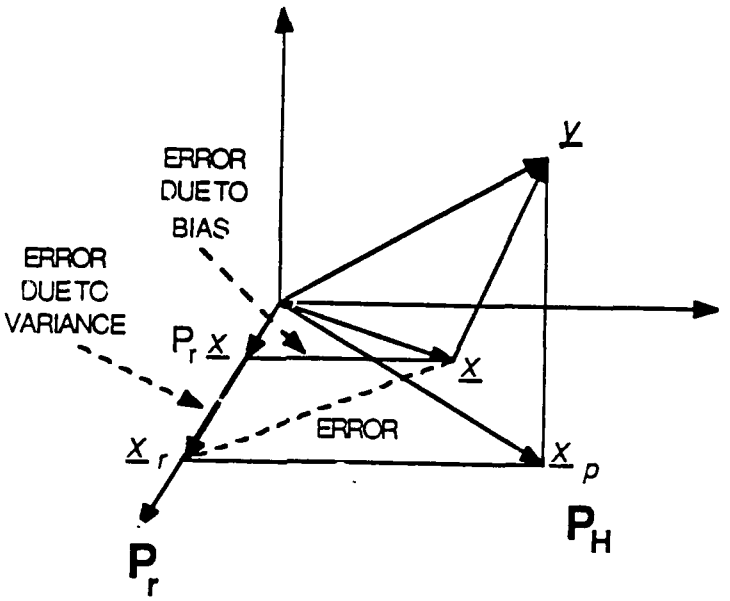


Figure 5.1 Orthogonal decomposition of error.

Once we have estimated the squared bias, we choose the rank for which our estimate of the squared bias plus the variance is minimum. But there is more to choosing the low rank projection $\mathbf{P}_r$ than the choice of rank. We must also choose the orientation of the subspace. For this we return to the model matrix, $\mathbf{H}$, and apply the singular value decomposition. The left singular vectors of $\mathbf{H}$ orthogonally span the signal subspace, and a subset of the singular vectors is chosen to form the range of $\mathbf{P}_r$. The subset may be chosen by taking the singular vectors corresponding to the largest singular values, or by a data dependent method such as that suggested in [KTS84], and [Sch91]. In most of our simulations we have used a data dependent selection method.

**Estimators of Squared Bias.** We now consider how the squared bias may be estimated from the data for a given low rank projection $\mathbf{P}_r$. Recall that the squared bias is defined as

$$b_r^2 = \underline{b}_r^H \underline{b}_r = \underline{x}^H (\mathbf{I} - \mathbf{P}_r) \underline{x} = \underline{x}^H (\mathbf{P}_H - \mathbf{P}_r) \underline{x} \qquad (5.7)$$

The maximum likelihood estimator for the bias vector itself is easily shown to be

$$\tilde{\underline{b}}_r = (\mathbf{P_H} - \mathbf{P}_r)\underline{y}. \tag{5.8}$$

The principle of invariance of maximum likelihood estimators [Sch91] implies that the maximum likelihood estimator of squared bias is

$$\begin{aligned} \tilde{b}^2_{r,\#1} &= \tilde{\underline{b}}_r^H \tilde{\underline{b}}_r \\ &= \underline{y}^H(\mathbf{P_H} - \mathbf{P}_r)\underline{y}. \end{aligned} \tag{5.9}$$

We shall call this estimator #1, and the order selection rule based on this estimator, rule #1.

But our earlier comments have called into question the blind application of ML, especially when the invariance principle is involved. Scharf and Storey [ScS86] observed that this ML estimator of squared bias is itself a biased estimator, with bias equal to $(m-r)\sigma^2$. This is true in spite of the fact that $\tilde{\underline{b}}_r$ is an unbiased estimator of $\underline{b}_r$. They proposed an unbiased estimator of the squared bias, obtained by subtracting the known bias from the maximum likelihood estimator of Equation 5.9.

$$\begin{aligned} \tilde{b}^2_{r,\#2} &= \tilde{\underline{b}}_r^H \tilde{\underline{b}}_r - (m-r)\sigma^2 \\ &= \underline{y}^H(\mathbf{P_H} - \mathbf{P}_r)\underline{y} - (m-r)\sigma^2. \end{aligned} \tag{5.10}$$

We shall call this estimator #2.

While unbiased, estimator #2 is not without disadvantage. It can give negative estimates of squared bias, which are clearly unrealistic. We propose as estimator #3 a modification of estimator #2 such that an estimate of zero is returned in place of any negative values.

$$\tilde{b}^2_{r,\#3} = \max(\tilde{b}^2_{r,\#2}, 0). \tag{5.11}$$

Of course, we have destroyed the unbiasedness in the modification. For a given rank, each of the estimators of bias squared may be plotted as a function of the maximum likelihood estimator as shown in Figure 5.2 for $(m-r) = 5$.

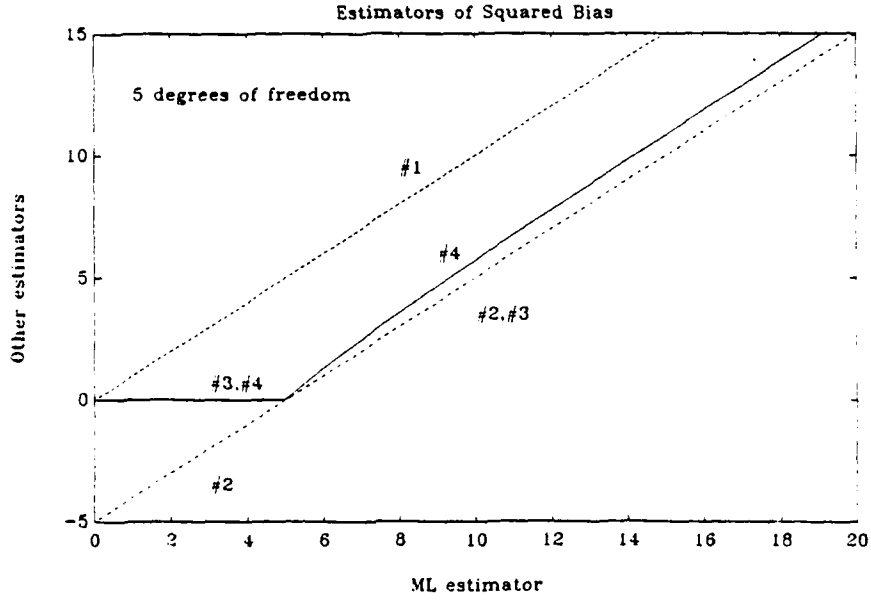


Figure 5.2 Four estimators of squared bias.

In the remainder of this section we develop our primary result, a new estimator of squared bias which will be called estimator #4. Consider the normalized maximum likelihood estimator of squared bias:

$$x^2 = \frac{\tilde{b}^2_{r,\#1}}{\sigma^2}. \tag{5.12}$$

The distribution of $x^2$ is noncentral chi-squared with $m-r$ degrees of freedom and noncentrality parameter $\lambda = \frac{1}{\sigma^2} b_r^2$. Thus, the estimation of $b_r^2$ may be posed as the fundamental problem of estimating the noncentrality parameter in a noncentral chi-squared distribution with known degrees of freedom, based on a single observation. A similar estimation problem was considered by Meyer [Mey67], but with only two degrees of freedom and with multiple observations.

We obtain the maximum likelihood estimator $\hat{b}_r^2$ for our case by maximizing the like-

lihood function:

$$\bar{b}^2_{r,\#4} = \arg\max_{\lambda \geq 0} \chi^2_{m-r}(x^2;\lambda). \tag{5.13}$$

In this Equation, $\chi^2_{m-r}(x^2;\lambda)$ is the noncentral chi-squared density function of $x^2$ with $m-r=d$ degrees of freedom and noncentrality parameter $\lambda$. It is given in [Lan69] as

$$\chi^2_d(x^2;\lambda) = \frac{e^{-\frac{1}{2}(x^2+\lambda)}(x^2)^{\frac{1}{2}(d-2)}}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})} \sum_{j=0}^{\infty} \frac{(\frac{x^2\lambda}{2})^j}{j! \prod_{k=0}^{j-1}(d+2k)}. \tag{5.14}$$

Setting the derivative of Equation 5.14 with respect to $\lambda$ equal to zero, we obtain the maximum likelihood equation

$$\frac{e^{-\frac{x^2}{2}}(-\frac{1}{2})e^{-\frac{\lambda}{2}}(x^2)^{\frac{1}{2}(d-2)}}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})} \sum_{j=0}^{\infty} \frac{(\frac{x^2\lambda}{2})^j}{j! \prod_{k=0}^{j-1}(d+2k)} + \frac{e^{-\frac{x^2}{2}}e^{-\frac{\lambda}{2}}(x^2)^{\frac{1}{2}(d-2)}}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})} \sum_{j=1}^{\infty} \frac{(\frac{x^2}{2})^j j\lambda^{j-1}}{j! \prod_{k=0}^{j-1}(d+2k)} = 0. \tag{5.15}$$

For $x^2 \neq 0$ we may multiply Equation 5.15 through by

$$\frac{2^{\frac{d}{2}}\Gamma(\frac{d}{2})}{e^{-\frac{x^2}{2}}e^{-\frac{\lambda}{2}}(x^2)^{\frac{1}{2}(d-2)}}$$

and simplify to

$$-\frac{1}{2} + \sum_{j=1}^{\infty} \frac{(\frac{x^2}{2})^j j\lambda^{j-1} - \frac{1}{2}(\frac{x^2}{2})^j \lambda^j}{j! \prod_{k=0}^{j-1}(d+2k)} = 0. \tag{5.16}$$

The result is a power series in $\lambda$, and may be approximately solved by any of several numerical approaches. The power series may be made more explicit by rewriting Equation 5.16 as

$$\sum_{j=0}^{\infty} a_j \lambda^j = 0, \tag{5.17}$$

where

$$a_j = \frac{(\frac{x^2}{2})^j(\frac{x^2-n-2j}{2})}{j! \prod_{k=0}^{j}(d+2k)}. \tag{5.18}$$

If Equation 5.17 has a root for positive real $\lambda$, that root is the ML estimator of the noncentrality parameter. If the largest real root is negative, the ML estimator of noncentrality is zero, since the maximization of Equation 5.13 is restricted to positive $\lambda$.

There are some theoretical curiosities about the new estimator of squared bias (#4). It was obtained by two sequential applications of the principle of maximum likelihood. First, through invariance, estimator #1 was found to be the ML estimator of squared bias. Then this first estimator was considered as a statistic and its distribution was known to be chi-squared with noncentrality parameter equal to the squared bias. So ML was applied again, based on the chi-squared statistic, to obtain our new estimator. One can envision this process being extended to further iterations of the ML principle, although the complexity of the likelihood functions increases rapidly, so that even one more iteration appears intractable.

Also of interest is the fact that the first ML estimator of squared bias is not a sufficient statistic for squared bias, as can be seen by application of the Fisher-Neyman factorization theorem [Sch91]. This fact tends to undermine the credibility of the new estimator. Even so, the simulations show that it outperforms the other three estimators at low signal-to-noise ratio. At high signal-to-noise ratio all four estimators perform about the same, and high rank models are generally preferred.

**Simulations.** We have performed simulations using MATLAB$^{TM}$ to test and compare estimators #1 through #4 and the corresponding order selection rules on real-valued data. Equation 5.17 is solved numerically by truncating the series and rooting the resulting polynomial. The typical pattern of root locations from this process is shown in Figure 5.3, where we are interested in the one positive real root. To save time during actual simulations, we precompute the solutions to Equation 5.17 for a range of the statistic $x^2$ from .1 to 50 in increments of .1, and for 1 to 10 degrees of freedom. The ML function is then evaluated by table look up with linear interpolation. For larger values of $x^2$, which occur at high signal-to-noise ratio, the ML function is nearly linear, and we use a linear approximation.

We have run four sets of tests at 5 values of SNR for each set. In each set 200 realizations of the noise vector have been simulated at each SNR. In the first set the parameter vector is also chosen randomly at each realization, while the parameter is fixed in the other three sets of tests. All four bias estimators are applied to each realization and the sample

mean squared error of the resulting signal estimators is observed along with the order selection behavior of each rule.

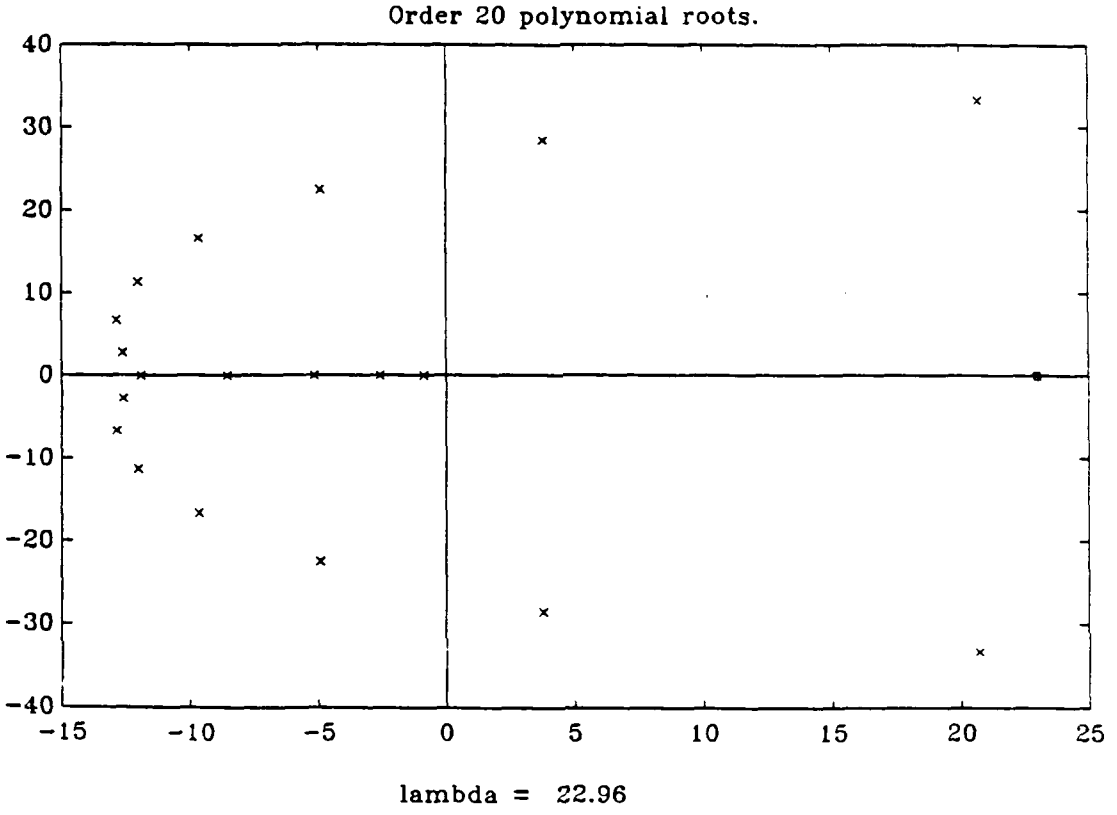


Figure 5.3 Roots of truncated power series.

For the first set of tests, we let $n = 10$ and $m = 6$. For each signal-to-noise ratio we fix a randomly generated system matrix $\mathbf{H}$ and run 200 trials. In each trial a new parameter $\underline{\theta}$ and a new noise vector $\underline{\nu}$ are generated according to

$$\begin{aligned} \underline{\theta} &: N(\underline{0}, \mathbf{I}), \\ \underline{\nu} &: N(\underline{0}, \sigma^2 \mathbf{I}). \end{aligned} \tag{5.19}$$

The signal-to-noise ratio used is the expected per-sample SNR, calculated as

$$\mathrm{SNR} = 10 \log_{10}\left(\frac{\mathrm{Tr}(\mathbf{H}^T\mathbf{H})}{n\sigma^2}\right). \tag{5.20}$$

The actual squared error (SE) for a typical realization at 0 dB is plotted versus rank in Figure 5.4. Also shown are estimators #2 (labeled CM) and #4 (labeled IML). In the realization shown, both rules selected rank 4 while the best choice was rank 6.

Figure 5.5 shows a plot versus SNR of the observed sample mean squared error (MSE) between the true signal $\underline{x}$ and the low rank estimator $\underline{x}_r$ chosen by each of the proposed order selection rules. The *Oracle* curve gives the sample MSE of a hypothetical order selection rule that always makes the best choice to minimize the true error and thus represents the best possible performance. Success as an order selection rule is plotted versus SNR in Figure 5.6, where the measure of success is the number of trials out of 200 where the rule made the right choice of rank, in agreement with the oracle. The histogram in Figure 5.7 gives a more complete characterization of the order selection behavior of rule #4 at 0 dB. For example, the bar above 1 indicates how often the rule overestimated the best rank by 1.

In the second set of tests, we have selected for $\mathbf{H}$ a $10 \times 3$ matrix representing three sinusoids with relative radian frequencies $\frac{2\pi}{10}$, $\frac{4\pi}{10}$, and $\frac{4.4\pi}{10}$. The singular values of this system matrix are approximately 2.9, 2.2 and 1.0. The parameter is fixed for all trials at $\underline{\theta} = [1\ 1\ 1]^T$. The results for this set of tests are given in Figures 5.8 and 5.9, where SNR is here defined as

$$\text{SNR} = 10 \log_{10}(\frac{\underline{x}^T \underline{x}}{n\sigma^2}). \tag{5.21}$$

The third set of tests differ from the second only in the new choice of $\underline{\theta} = [0\ 1\ 1]^T$. Thus for these tests the signal lies in a rank 2 subspace in terms of $\mathbf{H}$, but not in terms of the left singular vectors of $\mathbf{H}$. Results for this test are given in Figures 5.10 and 5.11.

For the fourth set of tests the same system matrix is used as for the second and third sets. This time the parameter vector has been chosen to result in a signal that is very nearly rank 2 in terms of the left singular vectors of $\mathbf{H}$, namely $\underline{\theta} = [0.3\ 1\ 1]^T$. Also different in this set of tests is the method used to choose the priority of the singular vectors. The first three tests use the data dependent method described in [Sch91] of choosing the singular vectors whose inner product with $\underline{y}$ is largest, indicating the strongest modes for each trial. In this test, the singular values of $\mathbf{H}$ are used to determine the dominant singular vectors. Figures 5.12 and 5.13 show the results for the fourth set of tests.

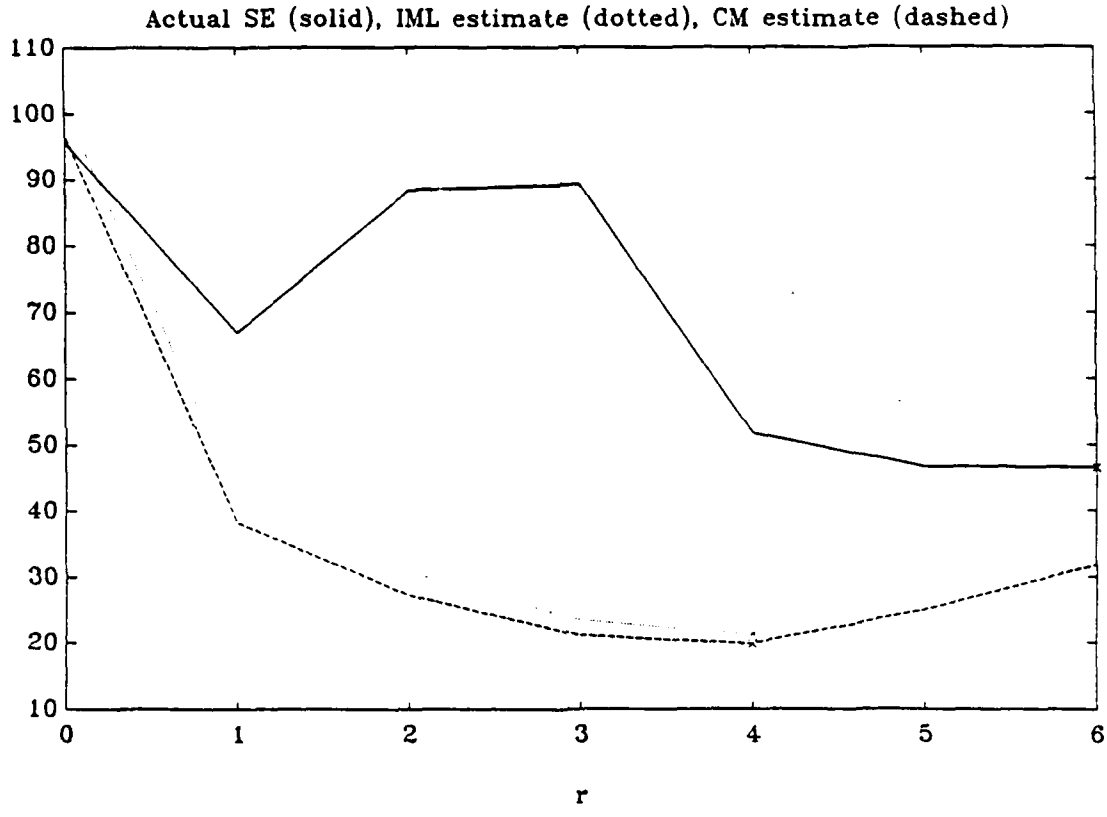


Figure 5.4 SE of a typical realization.

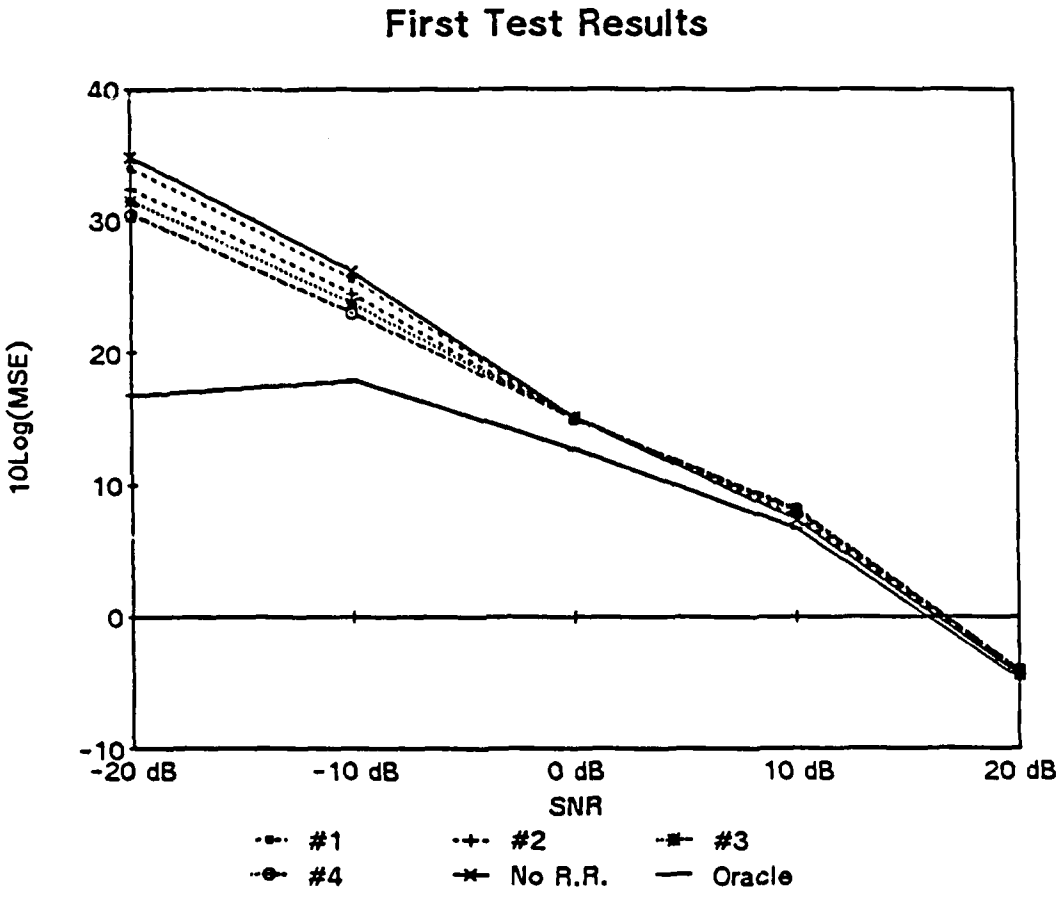


Figure 5.5 Sample MSE versus SNR.

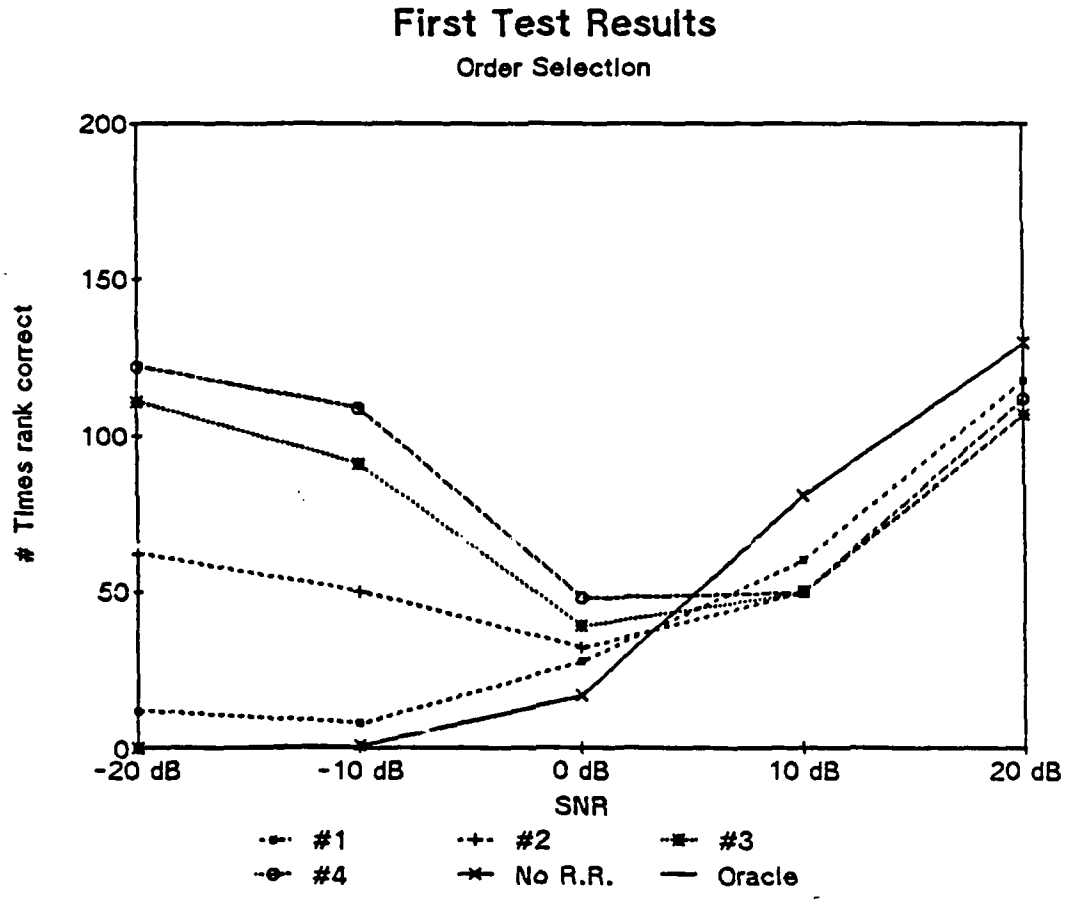


Figure 5.6 Order selection success versus SNR.

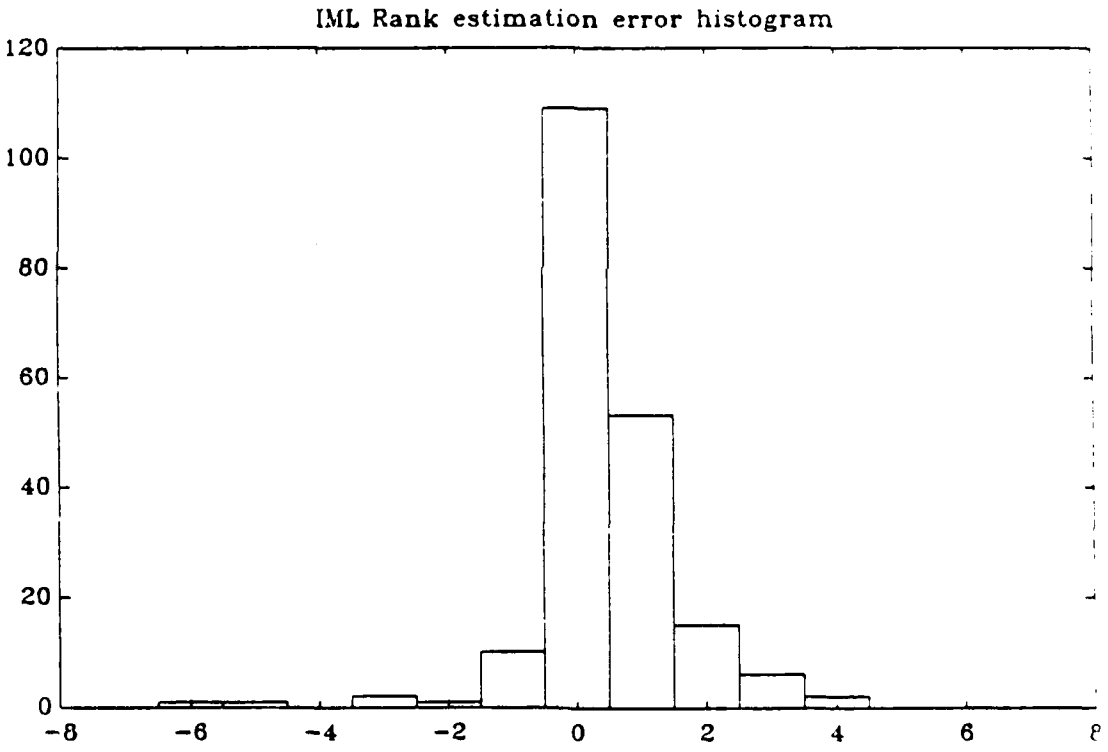


Figure 5.7 Order selection histogram.

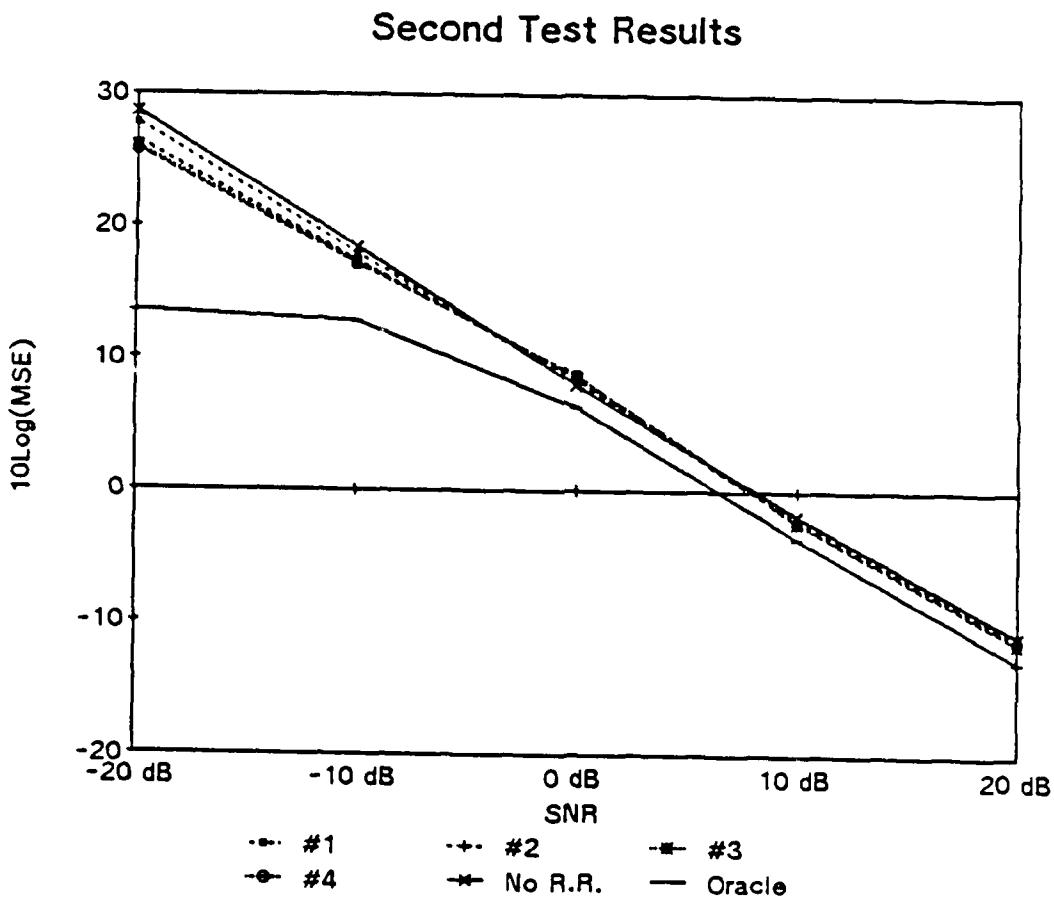


Figure 5.8 Sample MSE versus SNR.

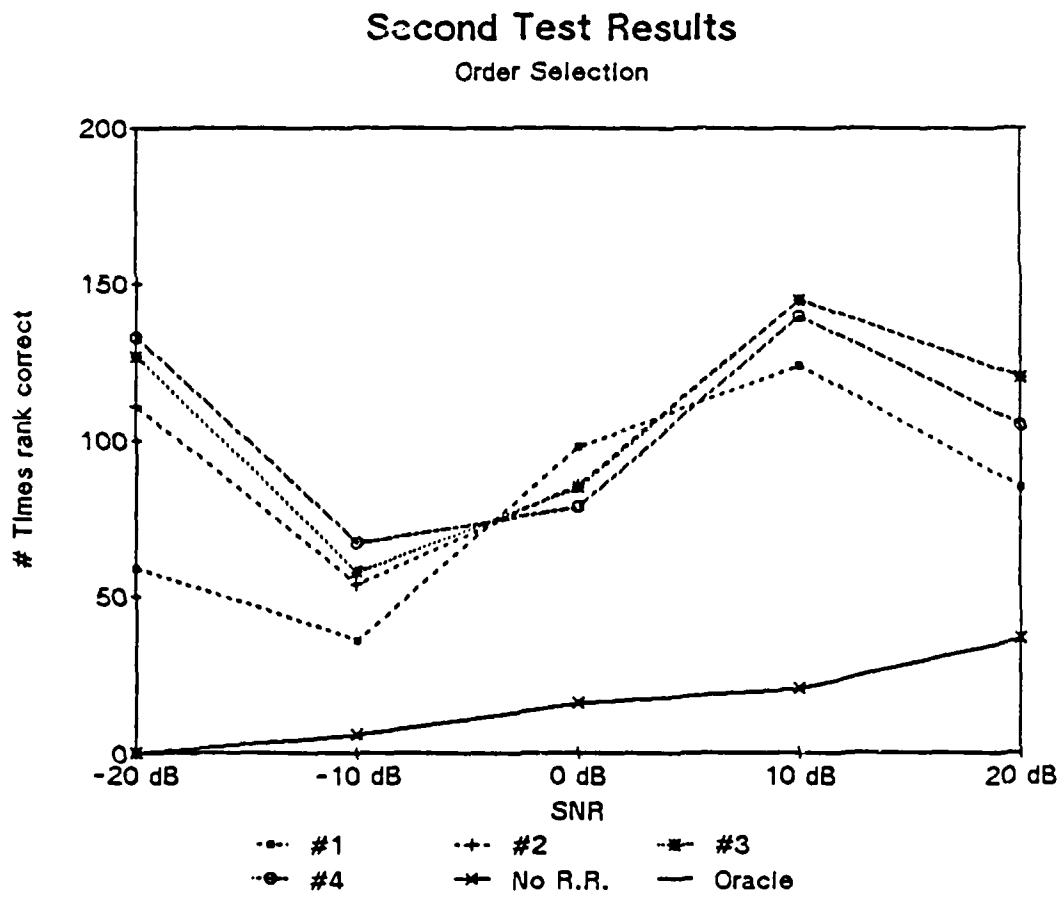


Figure 5.9 Order selection success.

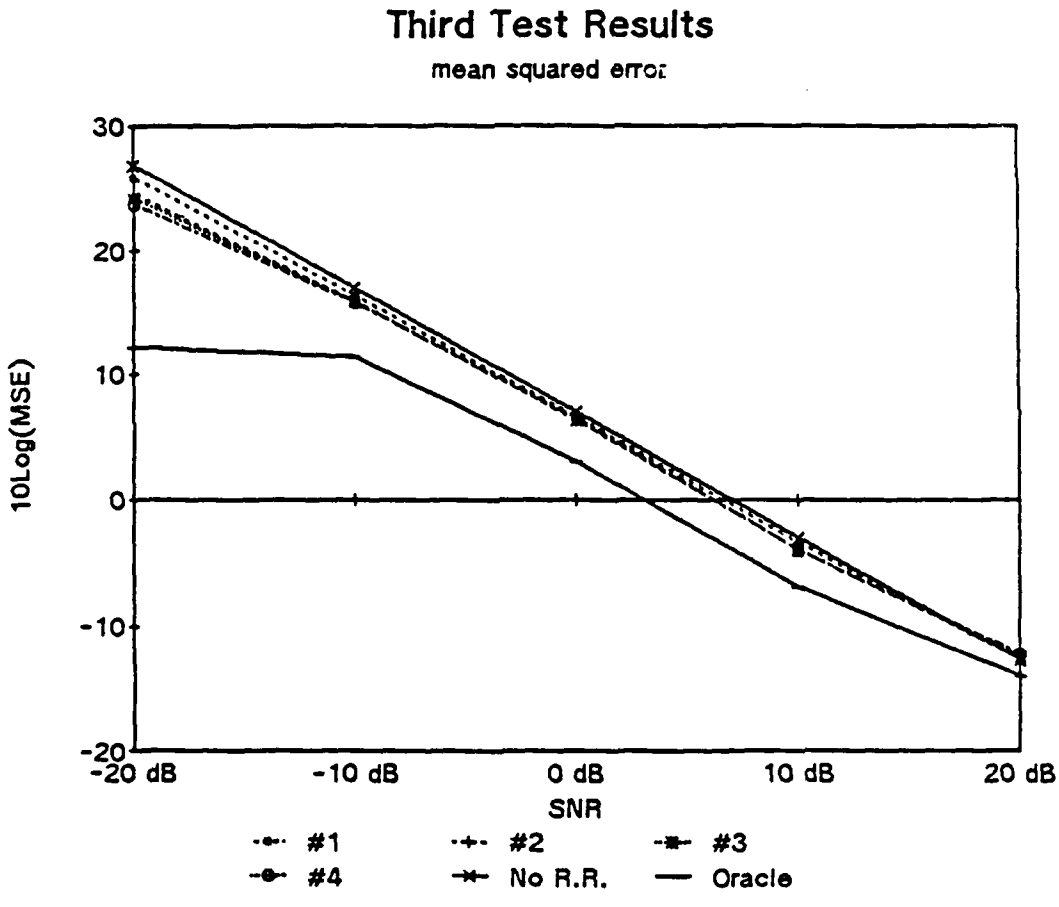


Figure 5.10 Sample MSE versus SNR.

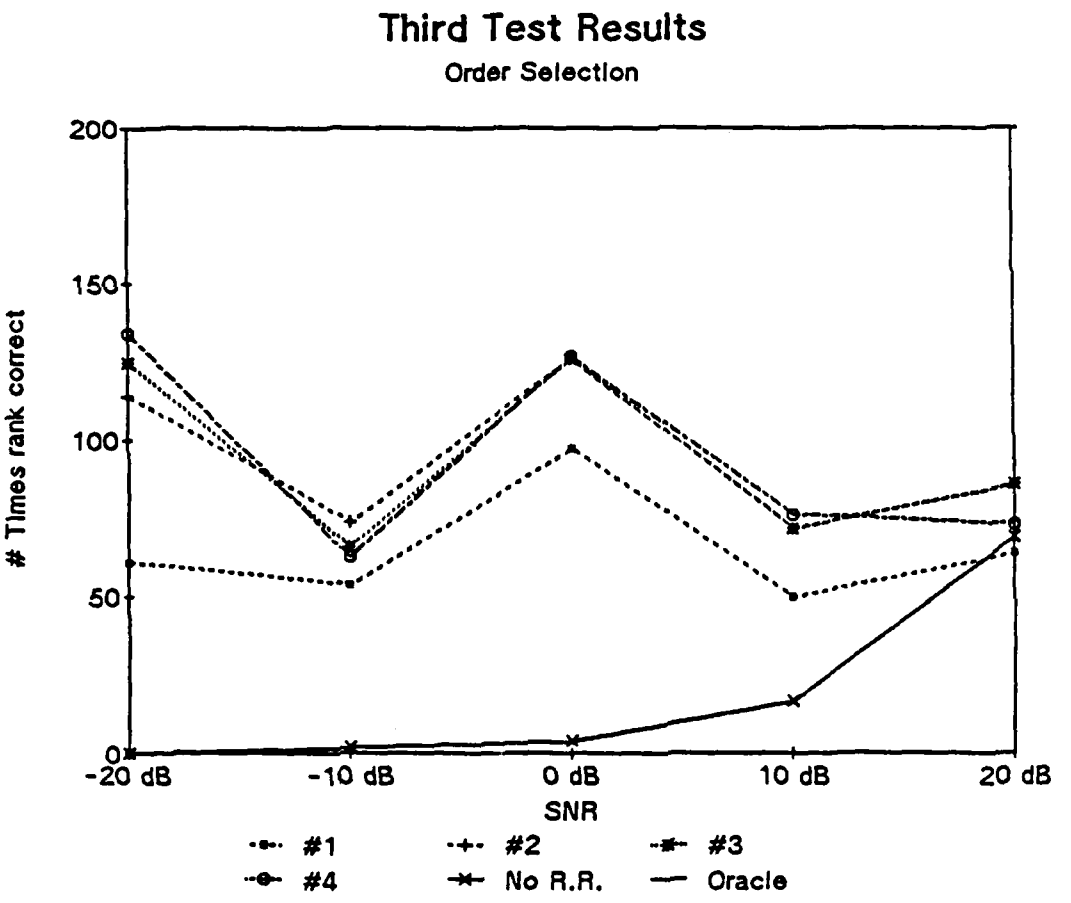


Figure 5.11 Order selection success.

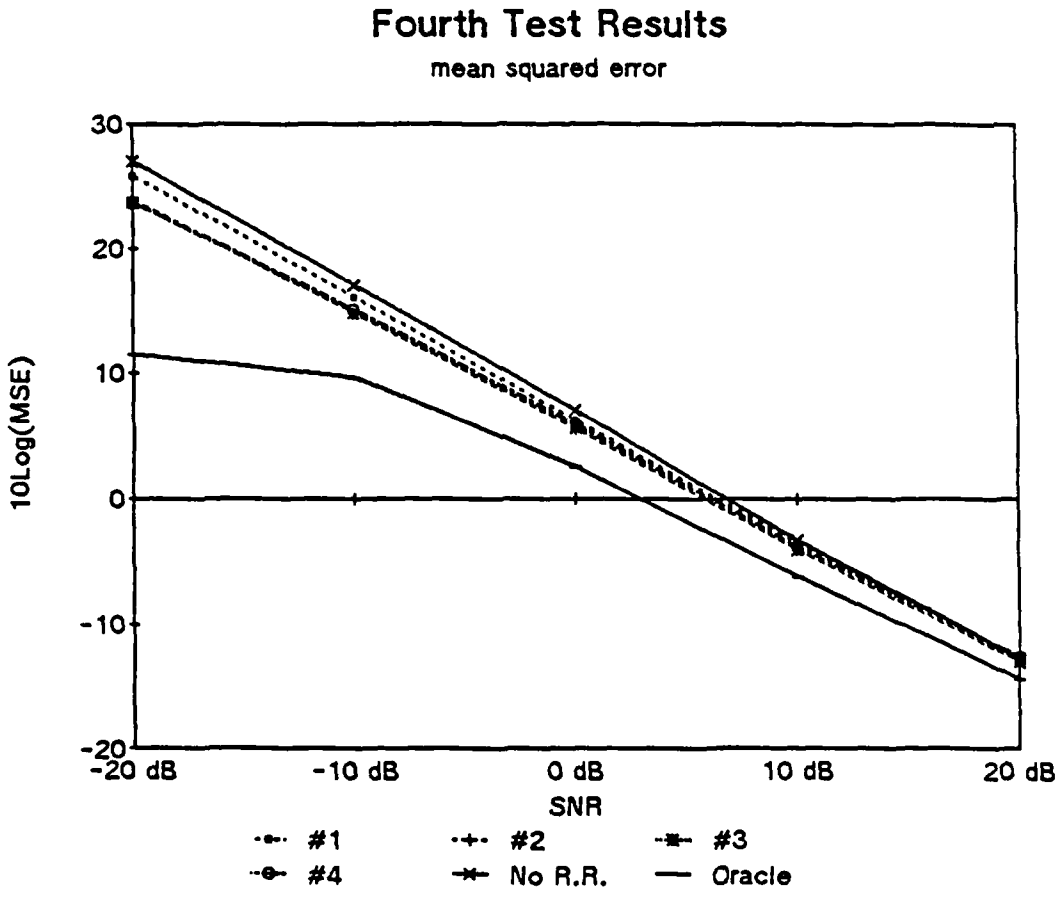


Figure 5.12 Sample MSE versus SNR.

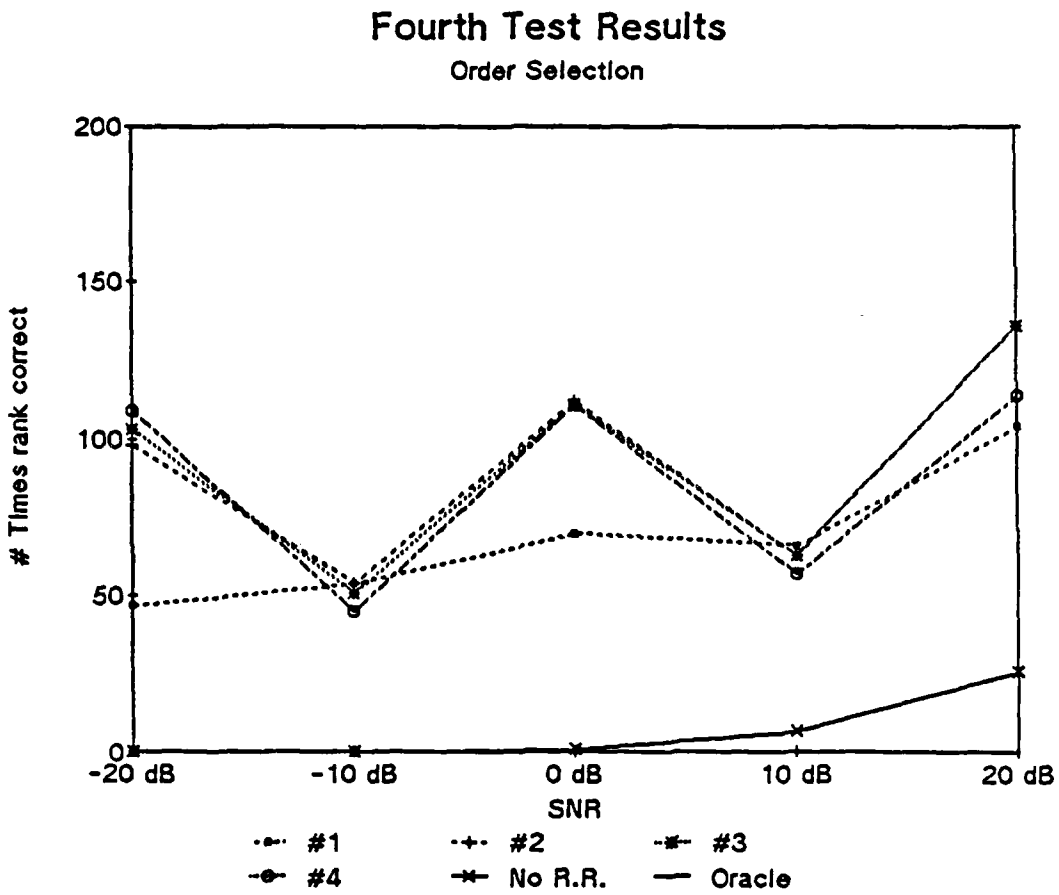


Figure 5.13 Order selection success.

**Conclusions.** As expected, relatively clean data is best modeled by the full rank system matrix. The tests generally show that rank reduction becomes useful only for relatively noisy data. On the other hand, if the data is too noisy the best choice of rank is always 0. For the range of SNR over which rank reduction is useful, the new order selection rule (#4) performs generally better than the others under the conditions of these tests.

The iterated ML approach to bias estimation leads to a nonlinear optimization problem. As such it requires much greater computational effort than the linear bias estimator suggested in [ScS86]. However, most of the extra computation can be done in advance and our experiments suggest that the extra computational effort required by the iterated ML estimator does improve performance. The concept of iterating the principle of maximum likelihood remains a curiosity.

## 5.2 Order selection in Subspace Identification

We now consider the order selection problem of identifying the number of impulses making up a structured noise subspace, assuming that the signal subspace (**H**) is known. In previous work [SMB87] we have successfully applied the order selection rule of Scharf and Storey [ScS86] to this problem. However, the derivation of that order selection rule is based on the bias versus variance tradeoff that occurs in the rank reduction problem. It is not clear that it should be applied to order selection for the subspace identification problem.

An alternative way to proceed is to use a Bayesian hypothesis test on $t$, the number of impulses presumed to exist in the observed data vector:

$$\begin{aligned} H_0 &: \quad t = 0 \\ H_1 &: \quad t = 1 \\ &\vdots \\ H_q &\quad t = q. \end{aligned} \tag{5.22}$$

The maximum number of impulses we can successfully correct is $q = n - m$ where $n$ is the size of the data vector and $m$ is the rank of the signal subspace.

The signal model and assumptions are as follows. The signal subspace **H** is known. The data consists of signal plus structured noise plus white Gaussian background noise:

$$\underline{y} = \mathbf{H}\underline{\theta} + \mathbf{S}\underline{\phi} + \underline{\nu}.$$

The structured noise is impulsive in nature, corresponding to a structured noise matrix **S** whose columns are an unknown subset of the columns of the identity matrix. The parameters $\underline{\theta}$ and $\underline{\phi}$ are unknown. We consider first the case where all variables are real valued, so the background noise is distributed as

$$\underline{\nu} : \mathcal{N}(\underline{0}, \sigma^2\mathbf{I}), \tag{5.23}$$

where we will consider both the case where $\sigma^2$ is known and the case where it is not known.

To derive the Bayes test on $t$ outlined earlier, we need to know the joint probability

density function of the data $\underline{y}$ and the number of impulses $t$. This can be expressed as

$$f_{\underline{y},t}(\underline{y},t) = f_{\underline{y}|\mathbf{S},t}(\underline{y}|\mathbf{S},t) f_{\mathbf{S}|t}(\mathbf{S}|t) f_t(t). \tag{5.24}$$

The conditional density of $\underline{y}$ given $\mathbf{S}$ and $t$ is a joint normal inherited from the noise density:

$$f_{\underline{y}|\mathbf{S},t}(\underline{y}|\mathbf{S},t) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}(\underline{y} - \mathbf{H}\underline{\theta} - \mathbf{S}\underline{\phi})^T(\underline{y} - \mathbf{H}\underline{\theta} - \mathbf{S}\underline{\phi})\right). \tag{5.25}$$

To get suitable density functions for $\mathbf{S}$ and $t$ let us assume that impulse errors affect the data samples independently, and each sample is affected with probability $p$. This implies that the number of impulses obeys a binomial distribution, while all combinations of $t$ impulses occur with the same probability. In this case the probability mass function for $t$ is

$$f_t(t) = \binom{n}{t} p^t (1-p)^{n-t}. \tag{5.26}$$

The conditional probability mass function for $\mathbf{S}$ given $t$ is uniform:

$$f_{\mathbf{S}|t}(\mathbf{S}|t) = \frac{1}{\binom{n}{t}}. \tag{5.27}$$

To implement the Bayes test for $t$ we must choose $t$ and the unknown parameters $\mathbf{S}, \underline{\theta}, \underline{\phi}$, and possibly $\sigma^2$ to maximize the joint density of $\underline{y}$ and $t$. This is of course equivalent to minimizing the negative log of the joint density function, so we now write our objective as

$$\min_{\mathbf{S},\underline{\theta},\underline{\phi},\sigma^2,t} \left[\frac{1}{2\sigma^2}(\underline{y} - \mathbf{H}\underline{\theta} - \mathbf{S}\underline{\phi})^T(\underline{y} - \mathbf{H}\underline{\theta} - \mathbf{S}\underline{\phi}) + \frac{n}{2}\ln(2\pi\sigma^2) - \ln\left(p^t(1-p)^{n-t}\right)\right]. \tag{5.28}$$

Minimization of this objective with respect to $\underline{\theta}$ and $\underline{\phi}$ may be accomplished by substituting the ML estimates of these parameters as derived in Chapter VI, resulting in the objective:

$$\min_{\mathbf{S},\sigma^2,t} \left[\frac{1}{2\sigma^2}\underline{y}^T(\mathbf{I} - \mathbf{P}_{\mathbf{HS}})\underline{y} + \frac{n}{2}\ln(2\pi\sigma^2) - \ln\left(p^t(1-p)^{n-t}\right)\right]. \tag{5.29}$$

We now proceed to evaluate this objective function for two cases depending on whether or not $\sigma^2$ is known.

**Known Noise Variance.** When $\sigma^2$ is known the second term in Equation 5.29 is

constant and may be dropped, resulting in the objective:

$$\min_{\mathbf{S},t} \left[ \frac{1}{2\sigma^2} \underline{y}^T (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} - \ln \left( p^t (1-p)^{n-t} \right) \right]. \tag{5.30}$$

Since $t$ is determined for each $\mathbf{S}$, this objective may be viewed as a criterion for choosing $\mathbf{S}$. For each of the finite number of possible selection matrices $\mathbf{S}$ (and its corresponding $t$), the function in Equation 5.30 is computed. The Bayes hypothesis test for order selection with known noise variance is to choose the $\mathbf{S}$ for which the computed value of Equation 5.30 is smallest.

This test may be compared to the order selection rule used in [SMB87] in which the objective function may be expressed as:

$$\min_{\mathbf{S},t} \left[ \frac{1}{\sigma^2} \underline{y}^T (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} + (2t - m) \right]. \tag{5.31}$$

In both rules the first term is data dependent while the second may be viewed as an order selection penalty function favoring some values of $t$ over others. Figure 5.14 shows a graph comparing the penalty functions for $n = 10$, with several values of the probability of impulse $p$. The rule of [SMB87] is linear in $t$. In the figure equal slopes imply equivalent order selection rules. You can see that for $n = 10$ the linear rule is most like a Bayes rule with $p$ about equal to 0.1. The figure also shows that the Bayes rule gives greatest favor (least penalty) to rank $t = np$, the expected value of the number of impulses present.

**Unknown Noise Variance.** When $\sigma^2$ is not known we substitute the ML estimate into Equation 5.29 as we did for the parameters $\underline{\theta}$ and $\underline{\phi}$. Minimizing Equation 5.29 with respect to $\sigma^2$ gives the following ML estimate of $\sigma^2$:

$$\hat{\sigma}^2 = \frac{1}{n} \underline{y}^T (\mathbf{I} - \mathbf{P_{HS}}) \underline{y}. \tag{5.32}$$

Substituting this into the objective function gives

$$\min_{\mathbf{S},t} \left[ \frac{n}{2} + \frac{n}{2} \ln \left( \frac{2\pi}{n} \underline{y}^T (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} \right) - \ln \left( p^t (1-p)^{n-t} \right) \right]. \tag{5.33}$$

The first term is constant and may be dropped from the minimization. We may obtain an

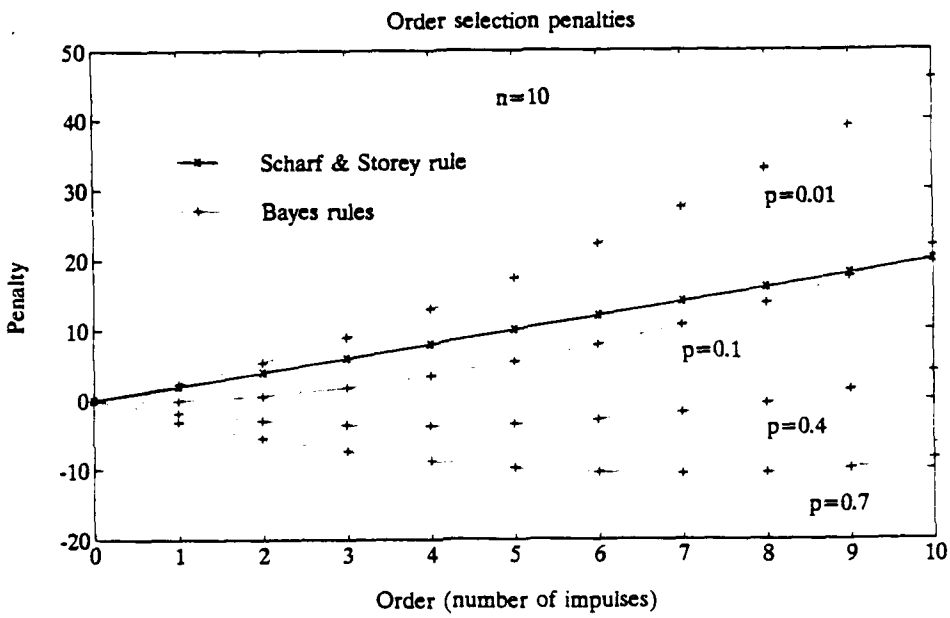


Figure 5.14 Two penalty functions for order selection.

alternate expression for this objective function by taking its natural exponential and dropping a constant multiplier:

$$\min_{\mathbf{S},t} \left[ \frac{(\underline{y}^T(\mathbf{I} - \mathbf{P}_{\mathbf{HS}})\underline{y})^{\frac{n}{2}}}{p^t(1-p)^{n-t}} \right]. \tag{5.34}$$

The Bayes hypothesis test for order selection with unknown noise variance is to choose the S for which the computed value of Equation 5.34 is smallest. This test cannot be compared directly to the tests in which $\sigma^2$ is known.

**Order Selection with Complex Data.** When the data $\underline{y}$ of Equation 5.22 is complex the appropriate distribution for the background noise becomes

$$\mathrm{Re}(\underline{\nu}) : \mathcal{N}(\underline{0}, \sigma^2\mathbf{I}) \quad \perp\!\!\!\perp \quad \mathrm{Im}(\underline{\nu}) : \mathcal{N}(\underline{0}, \sigma^2\mathbf{I}). \tag{5.35}$$

The matrix $\mathbf{H}$ and the parameters $\underline{\theta}$ and $\underline{\phi}$ may also be complex, but $\mathbf{S}$ is still a selection matrix. The only differences this introduces into the objective function for order selection is the replacement of the transpose by the Hermitian transpose and the replacement of $n/2$ by $n$ in the normalization of the Gaussian density.

For $\sigma^2$ known the objective function for complex data is

$$\min_{\mathbf{S},t} \left[ \frac{1}{2\sigma^2} \underline{y}^H (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} - \ln \left( p^t (1-p)^{n-t} \right) \right]. \tag{5.36}$$

The appropriate form of Equation 5.31 for complex data is the order selection rule from [SMB87]:

$$\min_{\mathbf{S},t} \left[ \frac{1}{2\sigma^2} \underline{y}^H (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} + (2t - m) \right]. \tag{5.37}$$

For $\sigma^2$ unknown the objective function for complex data is

$$\min_{\mathbf{S},t} \left[ n \ln \left( \frac{\pi}{n} \underline{y}^H (\mathbf{I} - \mathbf{P_{HS}}) \underline{y} \right) - \ln \left( p^t (1-p)^{n-t} \right) \right], \tag{5.38}$$

and the alternate expression for this objective function is

$$\min_{\mathbf{S},t} \left[ \frac{(\underline{y}^H (\mathbf{I} - \mathbf{P_{HS}}) \underline{y})^n}{p^t (1-p)^{n-t}} \right]. \tag{5.39}$$

**Simulation results.** The order selection rules presented above were tested on simulated complex data. There are three rules to be compared, corresponding to Equations 5.38, 5.37 and 5.36. The three were tested in parallel on the same data.

We chose $n = 10$ samples of the same complex exponential signal used to test the KiSS algorithm. We then added structured noise according to the model, with each vector element having probability $p = 0.1$ of an impulse. The impulse amplitudes were chosen with a fixed magnitude of 5 and uniform random phase in the complex plane. Background noise was added to the desired SNR (defined the same way as for KiSS tests in Chapter IV).

Table 5.1 shows order selection performance of the three rules at signal-to-noise ratios from 10 to 40 dB. The figures in the table are the number of times out of 50 trials that the correct $\mathbf{S}$ was chosen by the rule in question. The performance of the Bayes rule is the best in

the table, although you can see that the Bayes rule performs about the same as the Storey and Scharf rule. With unknown noise variance, the Bayes rule consistently chose a rank to high, and never got it right at any of the tested SNR's. Perhaps this could be improved by adjusting the normalization in the estimate of the noise variance $\sigma^2$, as discussed in Section 3.2.

Table 5.1 Order selection performance of three rules.

| SNR (dB) | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|
| Bayes Rule | 19 | 14 | 14 | 13 | 14 |
| Storey/Scharf Rule | 12 | 10 | 14 | 12 | 9 |
| Bayes Rule (unknown variance) | 0 | 0 | 0 | 0 | 0 |

# CHAPTER VI

## Parameter Estimation in the Structured Noise Model

Recall the structured noise model introduced in Chapter I:

$$\underline{y} = \underline{x} + \underline{b} + \underline{\nu}$$

$$\underset{n}{\begin{bmatrix} \underline{y} \end{bmatrix}} = \overbrace{\underset{n \times k}{\begin{bmatrix} \mathbf{H} \end{bmatrix}} \underset{k}{\begin{bmatrix} \underline{\theta} \end{bmatrix}}}^{} + \overbrace{\underset{n \times t}{\begin{bmatrix} \mathbf{S} \end{bmatrix}} \underset{t}{\begin{bmatrix} \underline{\varrho} \end{bmatrix}}}^{} + \underset{n}{\begin{bmatrix} \underline{\nu} \end{bmatrix}} \tag{6.1}$$

The goal in this chapter is to estimate the signal component $\underline{x}$ or the parameter $\underline{\theta}$ based on the received data $\underline{y}$. The model matrices $\mathbf{H}$ and $\mathbf{S}$ are assumed to be known, or previously estimated as in Chapters III and IV.

We consider three cases of this estimation problem: (1) We assume that $\underline{\theta}$ and $\underline{\varrho}$ are real or complex unknown parameters and the estimates $\widehat{\underline{\theta}}$ and $\widehat{\underline{\varrho}}$ are to be chosen to minimize the squared norm of the fitting error between $(\mathbf{H}\widehat{\underline{\theta}} + \mathbf{S}\widehat{\underline{\varrho}})$ and $\underline{y}$. In the real case, this is equivalent to placing a white Gaussian distribution on $\underline{\nu}$ and finding the ML estimators. (2) We assume that the noise vectors $\underline{\nu}$ and $\underline{\phi}$ are real random vectors drawn from Gaussian distributions with known autocorrelations. (3) In addition to the distributions of case 2, We assume that the parameter vector $\underline{\theta}$ is a real random vector drawn from a Gaussian distribution with known autocorrelation. These three cases may be summarized according to the density each one implies for to the data $\underline{y}$:

$$\begin{aligned} &(1) \qquad \underline{y} : \mathcal{N}(\mathbf{H}\underline{\theta} + \mathbf{S}\underline{\varrho}, \sigma_\nu^2 \mathbf{I}), \\ &(2) \qquad \underline{y} : \mathcal{N}(\mathbf{H}\underline{\theta}, \sigma_\nu^2 \mathbf{I} + \mathbf{S}\mathbf{R}_{\varrho\varrho}\mathbf{S}^T), \\ &(3) \qquad \underline{y} : \mathcal{N}(\underline{0}, \sigma_\nu^2 \mathbf{I} + \mathbf{S}\mathbf{R}_{\varrho\varrho}\mathbf{S}^T + \mathbf{H}\mathbf{R}_{\theta\theta}\mathbf{H}^T) \end{aligned} \tag{6.2}$$

In the least squares (LS) section (case 1), $\underline{\theta}$ and $\underline{\varphi}$ are assumed to be real or complex unknown parameters and the estimates $\hat{\underline{\theta}}$ and $\hat{\underline{\varphi}}$ are chosen to minimize the fitting error between $(\mathbf{H}\hat{\underline{\theta}} + \mathbf{S}\hat{\underline{\varphi}})$ and $\underline{y}$. In the minimum variance unbiased (MVUB) section (case 2) the noise processes $\underline{\nu}$ and $\underline{\varphi}$ are assumed to be real random vectors drawn from Gaussian distributions with known autocorrelations. The minimum mean squared error (MMSE) section deals with case 3. The last section shows a detailed application example for decoding linear block codes over the complex field.

## 6.1 Least Squares Estimation

In this section we consider $\underline{\theta}$ and $\underline{\varphi}$ in Equation 6.2 to be unknown real or complex parameters and we choose estimates of them which minimize the Euclidean norm of the residual, $\hat{\underline{\nu}}$, defined as

$$\hat{\underline{\nu}} = \underline{y} - (\mathbf{H}\hat{\underline{\theta}} + \mathbf{S}\hat{\underline{\varphi}}). \tag{6.3}$$

This is, of course, equivalent to an ML estimation problem when the vectors are real and $\underline{\nu}$ is white Gaussian noise. The following orthogonal projection operators will show up in the solution:

$$\begin{aligned} \mathbf{P}_{\mathbf{H}} &= \mathbf{H}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \\ \mathbf{P}_{\mathbf{H}^\perp} &= \mathbf{I} - \mathbf{P}_{\mathbf{H}} \\ \mathbf{P}_{\mathbf{S}} &= \mathbf{S}(\mathbf{S}^H\mathbf{S})^{-1}\mathbf{S}^H \\ \mathbf{P}_{\mathbf{S}^\perp} &= \mathbf{I} - \mathbf{P}_{\mathbf{S}}. \end{aligned} \tag{6.4}$$

There are several ways to proceed. One way is to estimate one of the parameters ($\underline{\theta}$ or $\underline{\varphi}$) as a function of the other, and then estimate the other parameter. Another way is to simultaneously estimate both of them. The same solution is obtained in either case, and we proceed here with the simultaneous estimation by rewriting the model equation as

$$\underline{y} = [\mathbf{H}\ \mathbf{S}] \begin{bmatrix} \underline{\theta} \\ \underline{\varphi} \end{bmatrix} + \underline{\nu} \tag{6.5}$$

This equation is in the form of an ordinary linear least squares problem, for which the solution,

involving the Moore-Penrose pseudoinverse, $[\mathbf{H}\ \mathbf{S}]^{\#}$, is easily shown [GVL89] to be

$$\begin{aligned}\begin{bmatrix}\hat{\underline{\theta}}\\ \hat{\underline{\phi}}\end{bmatrix} &= [\mathbf{H}\ \mathbf{S}]^{\#}\underline{y} = ([\mathbf{H}\ \mathbf{S}]^{H}[\mathbf{H}\ \mathbf{S}])^{-1}[\mathbf{H}\ \mathbf{S}]^{H}\underline{y}\\ &= \begin{bmatrix}\mathbf{H}^{H}\mathbf{H} & \mathbf{H}^{H}\mathbf{S}\\ \mathbf{S}^{H}\mathbf{H} & \mathbf{S}^{H}\mathbf{S}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{H}^{H}\\ \mathbf{S}^{H}\end{bmatrix}\underline{y}.\end{aligned} \tag{6.6}$$

Now apply the inversion formula for 2x2 block matrices [Kai80] to write the solution explicitly in terms of $\mathbf{H}$ and $\mathbf{S}$. The portion of the solution corresponding to $\hat{\underline{\theta}}$ is the parameter estimate

$$\begin{aligned}\hat{\underline{\theta}} &= (\mathbf{H}^{H}\mathbf{H})^{-1}\mathbf{H}^{H}[\mathbf{I}-\mathbf{S}(\mathbf{S}^{H}(\mathbf{I}-\mathbf{H}(\mathbf{H}^{H}\mathbf{H})^{-1}\mathbf{H}^{H})\mathbf{S})^{-1}\mathbf{S}^{H}\cdot\\ &\qquad (\mathbf{I}-\mathbf{H}(\mathbf{H}^{H}\mathbf{H})^{-1}\mathbf{H}^{H})]\underline{y}\\ &= (\mathbf{H}^{H}\mathbf{H})^{-1}\mathbf{H}^{H}(\mathbf{I}-\mathbf{S}(\mathbf{S}^{H}\mathbf{P}_{\mathbf{H}^{\perp}}\mathbf{S})^{-1}\mathbf{S}^{H}\mathbf{P}_{\mathbf{H}^{\perp}})\underline{y}.\end{aligned} \tag{6.7}$$

The least squares estimate of the signal $\underline{x}$ may be obtained by operating on $\hat{\underline{\theta}}$ with $\mathbf{H}$ resulting in

$$\begin{aligned}\hat{\underline{x}} &= \mathbf{H}\hat{\underline{\theta}}\\ &= \mathbf{E}_{\mathbf{H};\mathbf{S}}\underline{y},\end{aligned} \tag{6.8}$$

where $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ is the oblique projection defined in Chapter II as

$$\mathbf{E}_{\mathbf{H};\mathbf{S}} = \mathbf{H}(\mathbf{H}^{H}\mathbf{P}_{\mathbf{S}^{\perp}}\mathbf{H})^{-1}\mathbf{H}^{H}\mathbf{P}_{\mathbf{S}^{\perp}}. \tag{6.9}$$

So the operator which solves this least squares problem is an oblique projection.

Before going on to the other cases of the estimation problem, we give some interpretation and evaluation of the present result. Figure 2.1 in chapter II shows how Euclidean space can be resolved into three parts. It illustrates the action of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ on each component of received data. The range of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ is the signal subspace $\langle\mathbf{H}\rangle$, and the null space contains the structured noise subspace $\langle\mathbf{S}\rangle$. The remainder of Euclidean space, $\langle\mathbf{A}\rangle = \langle\mathbf{H},\mathbf{S}\rangle^{\perp}$, completes the null space. This leads to the interesting observation that the operator $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ entirely removes the structured noise, since according to the model, the structured noise lies in the null space of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$. At the same time, the signal, $\underline{x} = \mathbf{H}\underline{\theta}$, is undisturbed by $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ since it lies in the range.

While these properties are highly desirable, there is a tradeoff involved. The unstructured noise $\underline{\nu}$ is not dealt with as effectively by the oblique projection $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ as it would be by

the orthogonal projection $\mathbf{P_H}$. In fact, while some components of the full rank noise $\underline{\nu}$ will be reduced or removed, certain components may actually be amplified as shown in Figure 6.1. This possibility implies that the oblique projection estimator is best used when the structured noise dominates the full rank background noise. This claim is supported by an observation about the minimum variance unbiased estimate in Section 6.2.

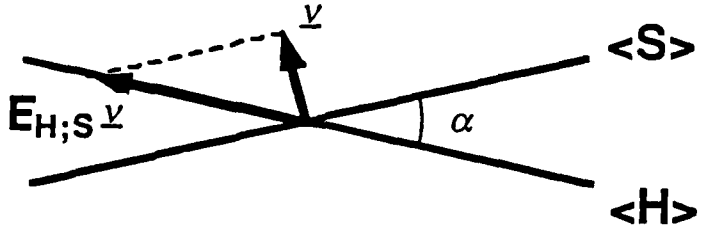


Figure 6.1 Possible effects of oblique projections

Just how bad is the effect of an oblique projection on the background noise? How can we evaluate a given oblique projection operator, $\mathbf{E}_{\mathbf{H};\mathbf{S}}$, in terms of its effect on the unstructured noise? The singular value decomposition (SVD) of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ plays an important role in this analysis. The SVD of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ is

$$\mathbf{E}_{\mathbf{H};\mathbf{S}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{6.10}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary and $\mathbf{\Sigma}$ is diagonal (all are $n \times n$):

$$\begin{aligned} \mathbf{U}\mathbf{U}^H &= \mathbf{U}^H\mathbf{U} = \mathbf{I} \\ \mathbf{V}\mathbf{V}^H &= \mathbf{V}^H\mathbf{V} = \mathbf{I} \\ \mathbf{\Sigma} &= \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & \mathbf{0} & \\ & & \sigma_k & & \\ & & & 0 & \\ & \mathbf{0} & & & \ddots \\ & & & & 0 \end{bmatrix}. \end{aligned} \tag{6.11}$$

The diagonal elements of $\mathbf{\Sigma}$ are the singular values of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$. Singular values are always non-negative reals, and as shown in Chapter II the singular values of an oblique projection may take on values of 0, 1 or any value greater than 1. This is an important distinction from orthogonal projections, whose singular values are all either 0 or 1.

The worst case noise power gain is, of course, given by the squared 2-norm of the operator, which is the square of the largest singular value of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$. If the noise vector $\underline{\nu}$ is resolved onto the basis $\mathbf{V}$ formed by the right singular vectors of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$, then each component will be multiplied by the corresponding singular value. If $\underline{\nu}$ has a spherically symmetrical distribution, it follows that the *average* noise power gain, $g$, is the average of all the squared singular values of $\mathbf{E}_{\mathbf{H};\mathbf{S}}$, of which only $k$ are nonzero:

$$\begin{aligned} g_{\text{ave}} &= \frac{1}{n}\sum_{i=1}^{k}\sigma_i^2, \\ g_{max} &= \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_k^2 \geq 0. \end{aligned} \tag{6.12}$$

We now give a geometric interpretation of this result. From Chapter II we know that the nonzero singular values of an oblique projection $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ are related to the principal angles between its range and null space. In the present setting this means that the background noise

gain is determined by the principal angles between the signal subspace $\langle \mathbf{H} \rangle$ and the structured noise subspace $\langle \mathbf{S} \rangle$, with the worst results when the angles are small. With $\alpha_i$ representing the principal angles we repeat the relation from Chapter II,

$$\sigma_i = \frac{1}{\sin(\alpha_i)}. \tag{6.13}$$

It should come as no surprise that the performance worsens when structured noise is close to signal by some measure, such as the sine of an angle between the subspaces.

## 6.2 Minimum Variance Unbiased Estimation with Real Data

We now return to the model Equation 6.2 and assume for case 2 that all vectors and matrices are real and the noise processes $\underline{\nu}$ and $\underline{\phi}$ are independent normally distributed random vectors with zero means and known correlations:

$$\begin{aligned} \underline{\nu} &: N(\underline{0}, \mathbf{R}_{\nu\nu}), \\ \underline{\phi} &: N(\underline{0}, \mathbf{R}_{\phi\phi}). \end{aligned} \tag{6.14}$$

We do not treat the complex valued vector case here. In earlier Chapters we have treated a complex vector with independent real and imaginary parts, each of which was distributed as $N(\underline{0}, \sigma^2\mathbf{I})$, as a complex normal random vector. But we know no definition for a normally distributed complex random vector with arbitrary correlation.

Let $\underline{w} = \mathbf{S}\underline{\phi} + \underline{\nu}$ denote the combined noise. Then $\underline{w}$ itself is normally distributed. The model takes on the form of the linear statistical model

$$\begin{aligned} \underline{y} &= \mathbf{H}\underline{\theta} + \underline{w}, \\ \underline{w} &: N(\underline{0}, \mathbf{R}_{ww}), \end{aligned} \tag{6.15}$$

where

$$\mathbf{R}_{ww} = \mathbf{R}_{\nu\nu} + \mathbf{S}^T\mathbf{R}_{\phi\phi}\mathbf{S}. \tag{6.16}$$

The unknown parameter $\underline{\theta}$ is to be estimated. The ML estimator of $\underline{\theta}$ [Sch91] is also the minimum variance unbiased (MVUB) estimator of $\underline{\theta}$. It is given by

$$\hat{\underline{\theta}} = (\mathbf{H}^T\mathbf{R}_{ww}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}_{ww}^{-1}\underline{y}. \tag{6.17}$$

The existence of $\mathbf{R}_{ww}^{-1}$ is guaranteed by the fact that $\mathbf{R}_{ww}$ is positive definite. To establish this, first note that $\mathbf{R}_{\nu\nu}$ is positive definite by the assumption that $\underline{\nu}$ is full rank noise. The matrix added to $\mathbf{R}_{\nu\nu}$ to form $\mathbf{R}_{ww}$ is non-negative definite, so the sum remains positive definite.

A signal estimate is obtained as in the least squares section by operating on $\hat{\underline{\theta}}$ with $\mathbf{H}$:

$$\begin{aligned} \hat{\underline{x}} &= \mathbf{H}\hat{\underline{\theta}} \\ &= \mathbf{E}\underline{y} \end{aligned} \tag{6.18}$$

where

$$\mathbf{E} = \mathbf{H}(\mathbf{H}^T\mathbf{R}_{ww}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}_{ww}^{-1}. \tag{6.19}$$

Once again, it involves an oblique projection, $\mathbf{E}$.

The range of $\mathbf{E}$ is the signal subspace $\mathbf{H}$ as in the least squares case. Its null space is not so easily characterized, although the following special case gives some insight and shows an additional connection to the least squares solution. If both noise processes are white ($\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ and $\mathbf{R}_{\nu\nu} = \sigma_\nu^2\mathbf{I}$), then the oblique projection $\mathbf{E}$ can be written as

$$\mathbf{E} = \mathbf{H}(\mathbf{H}^T(\mathbf{I} + r\mathbf{S}\mathbf{S}^T)^{-1}\mathbf{H})^{-1}\mathbf{H}^T(\mathbf{I} + r\mathbf{S}\mathbf{S}^T)^{-1}, \tag{6.20}$$

where

$$r = \sigma_\phi^2/\sigma_\nu^2. \tag{6.21}$$

When $r$ is zero there is no structured noise and $\mathbf{E}$ in Equation 6.20 simplifies to $\mathbf{P}_{\mathbf{H}}$, the orthogonal projection onto the signal subspace. On the other hand, when $r$ is large, the structured noise is the dominant interference. In Chapter II we found that in the limit as $r$ goes to infinity, the quantity $(\mathbf{I} + r\mathbf{S}\mathbf{S}^T)^{-1}$ converges to $\mathbf{P}_{\mathbf{S}^\perp}$. From this it immediately follows that

$$\lim_{r\to\infty} \mathbf{E} = \mathbf{E}_{\mathbf{H};\mathbf{S}}. \tag{6.22}$$

In other words, when $\underline{\phi}$ and $\underline{\nu}$ are white noise, the MVUB estimator converges to the least squares estimator as the structured noise becomes dominant.

In summary, the MVUB estimator is an oblique projection whose range is $\langle \mathbf{H} \rangle$ and

whose null space moves from $\langle \mathbf{H} \rangle^{\perp}$ toward $\langle \mathbf{S} \rangle$ as the structured noise power increases from zero toward infinity. This supports the earlier claim that the oblique projection obtained in the least squares problem, whose nullspace is $\langle \mathbf{S} \rangle$, is best suited to situations in which the structured noise dominates the unstructured noise.

## 6.3 Minimum Mean Squared Error Estimation with Real Data

For case 3 we assume Gaussian distributions on $\underline{\nu}$, $\underline{\phi}$, and $\underline{\theta}$:

$$\begin{aligned} \underline{\nu} &: N(\underline{0}, \mathbf{R}_{\nu\nu}), \\ \underline{\phi} &: N(\underline{0}, \mathbf{R}_{\phi\phi}), \\ \underline{\theta} &: N(\underline{0}, \mathbf{R}_{\theta\theta}). \end{aligned} \tag{6.23}$$

We choose an estimator $\hat{\underline{x}}$ to minimize the mean squared error between $\hat{\underline{x}}$ and $\underline{x}$. The solution is just a special case of the Gauss-Markov estimator, with the correlation matrix $\mathbf{R}_{yy} = \sigma_\nu^2 \mathbf{I} + \mathbf{S}\mathbf{R}_{\phi\phi}\mathbf{S}^T + \mathbf{H}\mathbf{R}_{\theta\theta}\mathbf{H}^T$ containing a term accounting for structured noise. The solution,

$$\hat{\underline{x}} = \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\underline{y} = \mathbf{H}\mathbf{R}_{\theta\theta}\mathbf{H}^T(\sigma_\nu^2\mathbf{I} + \mathbf{S}\mathbf{R}_{\phi\phi}\mathbf{S}^T + \mathbf{H}\mathbf{R}_{\theta\theta}\mathbf{H}^T)^{-1}\underline{y}, \tag{6.24}$$

may be found, for example, in [Poo88]. We note that $\hat{\underline{x}}$ is not an oblique projection of $\underline{y}$ in this case, nor in the Gauss-Markov estimator without structured noise. Also unlike the previous results, this solution is valid even if the subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ are overlapping. This can be interpreted to mean that even with overlapping subspaces, the information contained in the distribution functions allows some probabilistic separation of signal and structured noise.

## 6.4 Application to Decoding of Block Codes

In this section we present an example using the subspace identification and parameter estimation techniques we have developed. This work was originally published by Scharf, Mathys and Behrens [SMB87]. Since then, Mathys has submitted a paper [Mat90] for publication that continues and expands the work.

When the problem of decoding a linear block code is examined from the perspective of estimation theory it is seen to be equivalent to the problem of parameter estimation in the

Linear Statistical Model. In this section we exploit that equivalence to derive a procedure for decoding linear block codes over the real or complex field. We develop a decoding procedure based on a noise model that includes large impulsive errors in a few positions of the codeword as well as minor errors in all positions. The resulting decoder can be represented as an oblique projection operator determined by a finite search algorithm. Simulation results are given which show the decoder's performance in a specific situation.

Several authors have recently contributed to the extension of results from finite-field coding theory to the infinite field of the real numbers and its extension field, the complex numoers. Wolf [Wol83] has noted the effectiveness of error control coding against impulsive errors in the complex field, and has made the important observation that the error correction capacity of such codes is potentially nearly twice what finite-field coding theory would lead one to expect. Marshall [Mar85] has addressed the construction and implementation of complex number codes for impulse error correction, with attention given to the dynamic range of the elements in the codeword. Recent work in signal restoration has also been applied to complex number codes by Marshall [Mar86].

Practical decoding algorithms for impulse errors in complex number codes must consider that the codeword may also be subject to minor errors such as roundoff and/or background channel noise in every element. Wolf [Wol83] and Kumaresan [Kum86] have presented decoding strategies which account for such minor errors while protecting against major impulsive errors. The decoder we develop in this paper shows significant immunity to minor errors in correcting multiple impulse errors. In addition, some protection is provided against the minor errors themselves.

**Block Codes and the Linear Statistical Model.** We wish to recover the values of real or complex numbers that have been coded and transmitted over a channel subject to impulse noise and possibly also minor errors (background noise) in each value transmitted. By the use of appropriate linear block codes, the impulse errors can be detected and removed. In the absence of background noise, this removal can be entire for sufficiently few impulse errors.

One way to specify or represent any particular linear block code is by its encoder matrix, which we will call **H**. The information values to be sent are blocked into $m$-vectors $\underline{\theta}$, and the codeword vector $\underline{x}$ has length $n$ ($n > m$). Thus the dimensions of **H** are $n$ by $m$. The operation of encoding is expressed by the matrix-vector equation $\underline{x} = \mathbf{H}\underline{\theta}$ :

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & & \vdots \\ h_{n1} & \cdots & h_{nm} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}. \tag{6.25}$$

The elements $x_i$, $h_{ij}$ and $\theta_i$ may be real or complex, and **H** is full rank (rank $m$) for all uniquely decodable block codes. The received data $\underline{y}$ is the transmitted codeword $\underline{x}$ plus the channel noise $\underline{e}$.

$$\underline{y} = \mathbf{H}\underline{\theta} + \underline{e}. \tag{6.26}$$

Equation 6.26 has the form of the Linear Statistical Model. Results based on this connection will be used in the development of the decoder.

**Decoding for Impulse Noise Only.** By impulse noise we refer to an $n$-vector $\underline{b}$, which has zeros in all but $t$ positions ($t < n - m$). Such a vector has Hamming Weight equal to $t$, the number of non-zero entries. The nonzero elements may be large. In the next subsection we consider the combined effects of impulse noise and minor background noise, but at present we limit our consideration to impulse noise only. In this case the channel noise in Equation 6.26 is $\underline{e} = \underline{b}$.

We wish to determine the value of the impulse noise $\underline{b}$ by an examination of the received data $\underline{y}$, then to subtract it from $\underline{y}$ to obtain the transmitted codeword $\underline{x}$. To ensure that the estimate $\widehat{\underline{b}}$ is sparce according to the impulse noise model for $\underline{b}$, we represent $\underline{b}$ as the product of a selection matrix and a $t$-vector, as in the structured noise model described earlier:

$$\underline{b} = \mathbf{S}\underline{\phi}. \tag{6.27}$$

The vector $\underline{\phi}$ contains the (non-zero) amplitudes of the impulse errors. Each column of the

$n \times t$ selection matrix $\mathbf{S}$ corresponds to one impulse error and determines its position. In any column there is a 1 in the position of that impulse error and zeros in all other positions. The order of the columns is not significant; all selection matrices which are column permutations of one another represent the same set of impulse positions and are considered equivalent.

We separate the estimation problem into two parts: the estimation of $\mathbf{S}$, which is a subspace identification problem, and the estimation of the parameter $\underline{\phi}$. The subspace identification problem is equivalent to determining the number of impulse errors and their positions, and is accomplished by a search process through the fixed, finite space of possible selection matrices. The parameter estimation problem is equivalent to determining the amplitudes of the impulses, and is accomplished by an oblique projection.

We give here an alternate derivation of the oblique projection operator $\mathbf{E}_{\mathbf{H};\mathbf{S}}$ that solves the estimation problem, using the coding theory concepts of parity check matrices and syndromes. One way to split the observation space $\mathbf{R}^n$ (or $\mathbb{C}^n$) which contains the data $\underline{y}$ is into two linear subspaces: the $m$-dimensional code space spanned by $\mathbf{H}$, and its $(n-m)$-dimensional orthogonal complement $\langle \mathbf{H} \rangle^{\perp}$. We define $\mathbf{U}$ as an $n$ by $n-m$ matrix which spans the subspace $\mathbf{H}^{\perp}$. This gives us the property $\mathbf{U}^H\mathbf{H} = 0$. Coding theorists may recognize $\mathbf{U}$ as the parity check matrix for the code, which of course depends only on $\mathbf{H}$ and can be determined in advance. The parity check matrix $\mathbf{U}$ is not unique, and we find it advantageous to use an orthonormal span of the specified subspace, giving us the additional property $\mathbf{U}^H\mathbf{U} = \mathbf{I}$. Application of the parity check matrix $\mathbf{U}$ to the received data vector $\underline{y}$ produces the syndrome $\underline{z}$, a vector with length $(n-m)$:

$$\begin{aligned} \underline{z} &= \mathbf{U}^H\underline{y} = \mathbf{U}^H(\mathbf{H}\underline{\theta} + \underline{b}) \\ &= \mathbf{U}^H\underline{b} = \mathbf{U}^H\mathbf{S}\underline{\phi}. \end{aligned} \tag{6.28}$$

The syndrome is independent of the codeword because $\mathbf{U}^H\mathbf{H} = 0$. What remains is isomorphic to the projection of the impulse noise vector $\underline{b}$ onto $\langle \mathbf{U} \rangle$, the subspace orthogonal to the code space. The projection of $\underline{b}$ onto $\langle \mathbf{U} \rangle$ is, in fact, equal to $\mathbf{U}\underline{z} = \mathbf{U}\mathbf{U}^H\underline{b}$. We use the term *syndrome space* to refer to either the $(n-m)$-dimensional space in which the syndrome

$\underline{z}$ lies, or the isomorphic subspace of $\mathbf{R}^n$ (or $\mathbb{C}^n$) in which $\mathbf{U}\underline{z}$ lies. The impulse noise vector $\underline{b}$ may be resolved into a component in the code space and a component in the syndrome space. The component in the code space is indistinguishable from a legal codeword, so we turn to the syndrome as the basis of our estimation process.

If we assume $\mathbf{S}$ has already been estimated as $\widehat{\mathbf{S}}$ then the system of equations represented by $\underline{z} = \mathbf{U}^H \widehat{\mathbf{S}}\underline{\phi}$ is overdetermined (recall $t < n - m$) and can be solved in a least squares sense to obtain $\widehat{\underline{\phi}}$. Minimizing the usual least squares cost function $||\underline{z} - \mathbf{U}^H \widehat{\mathbf{S}}\widehat{\underline{\phi}}||^2$, the solution [GVL89] is

$$\widehat{\underline{\phi}} = \widehat{\mathbf{Q}}\underline{z}; \qquad \text{where} \qquad \widehat{\mathbf{Q}} = (\widehat{\mathbf{S}}^T \mathbf{U}\mathbf{U}^H \widehat{\mathbf{S}})^{-1} \widehat{\mathbf{S}}^T \mathbf{U}. \tag{6.29}$$

$\widehat{\mathbf{Q}}$ is the pseudoinverse of $\mathbf{U}^H \widehat{\mathbf{S}}$. Existence of the necessary inverse in Equation 6.29 is related to the relationship between the code space and the space spanned by $\widehat{\mathbf{S}}$, in which $\widehat{\underline{b}}$ lies. If the two share a common subspace of nonzero dimension, then that subspace of $\widehat{\mathbf{S}}$ is orthogonal to $\mathbf{U}$ and the product $\mathbf{U}^H \widehat{\mathbf{S}}$ will be rank deficient. The physical interpretation is that some impulse noise vectors in the $\widehat{\mathbf{S}}$ subspace are then legal codewords. This situation can be avoided, and the existence of the required inverse guaranteed, by choosing the code appropriately. For example, codes constructed by application of BCH or Reed-Solomon techniques to the field of reals and its extension field, the complex numbers, have codewords with Hamming Weight $> n - m$ (except the zero codeword). In this case no pattern of $n - m$ or fewer impulses has sufficient Hamming Weight to be a legal codeword.

We now observe that in the present case (with only impulse noise present) the overdetermined system of equations in Equation 6.28 is actually consistent if the estimate of $\mathbf{S}$ is correct. This is simply because, according to the model, the syndrome $\underline{z}$ originated from some actual $\underline{\phi}$ as $\mathbf{U}^H \mathbf{S}\underline{\phi}$. The search for $\widehat{\mathbf{S}}$, then, is a search for the smallest (minimum width $\widehat{t}$) selection matrix for which Equation 6.28 is consistent. Or, equivalently, a search for the smallest selection matrix for which the syndrome $\underline{z}$ lies in the subspace spanned by $\mathbf{U}^H \widehat{\mathbf{S}}$.

After determining $\widehat{\mathbf{S}}$, Equation 6.27 is used to synthesize $\widehat{\underline{b}}$, the estimate of the impulse

noise vector:

$$\begin{aligned}\hat{\underline{b}} &= \hat{\mathbf{S}}\hat{\underline{\phi}} = \hat{\mathbf{S}}\hat{\mathbf{Q}}\underline{z} \\ &= \hat{\mathbf{S}}(\hat{\mathbf{S}}^T\mathbf{U}\mathbf{U}^H\hat{\mathbf{S}})^{-1}\hat{\mathbf{S}}^T\mathbf{U}\mathbf{U}^H\underline{y} \\ &= \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}\underline{y}.\end{aligned} \tag{6.30}$$

Comparison with the oblique projection formulas in Chapter II shows that $\mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}$ is the oblique projection whose range is $\left\langle \hat{\mathbf{S}} \right\rangle$ and whose null space contains $\langle \mathbf{H} \rangle$ and the remaining subspace orthogonal to $\langle \mathbf{H} \rangle$ and $\left\langle \hat{\mathbf{S}} \right\rangle$. Subtracting $\hat{\underline{b}}$ from the data $\underline{y}$ gives the codeword estimate $\hat{\underline{x}}$:

$$\begin{aligned}\hat{\underline{x}} &= \underline{y} - \hat{\underline{b}} \\ &= (\mathbf{I} - \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}})\underline{y} \\ &= (\mathbf{P}_{(\mathbf{H};\hat{\mathbf{S}})^\perp} + \mathbf{E}_{\mathbf{H};\hat{\mathbf{S}}})\underline{y}.\end{aligned} \tag{6.31}$$

Note that the term $\mathbf{P}_{(\mathbf{H};\hat{\mathbf{S}})^\perp}\underline{y}$ will be zero in the absence of background noise, so $\hat{\underline{x}}$ lies in the code space $\langle \mathbf{H} \rangle$ and is an oblique projection of $\underline{y}$:

$$\hat{\underline{x}} = \mathbf{E}_{\mathbf{H};\hat{\mathbf{S}}}\underline{y}. \tag{6.32}$$

Figure 6.2 shows a block diagram of the communication system based on these principles.

After error correction, the decoding process to recover the original information vector $\underline{\theta}$ simply involves operating on the codeword estimate with $\mathbf{H}^{\#}$, the pseudoinverse of $\mathbf{H}$:

$$\hat{\underline{\theta}} = \mathbf{H}^{\#}\hat{\underline{x}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\hat{\underline{x}}. \tag{6.33}$$

Finally, note that under certain conditions the estimate $\hat{\underline{x}}$ is exactly equal to $\underline{x}$ in the absence of background noise. There are two conditions. First, the subspaces $\langle \mathbf{H} \rangle$ and $\langle \mathbf{S} \rangle$ must be disjoint so that the inverse in Equation 6.30 exists. Second, the smallest (least width) matrix $\hat{\mathbf{S}}$ for which Equation 6.29 is consistent must be unique. This uniqueness can be guaranteed only for appropriate codes (such as BCH and Reed-Solomon) and with $t < \frac{n-m}{2}$. However, if $\underline{e}$ arises from a continuous probability distribution, then the circumstances which lead to non-uniqueness occur with probability zero for $t$ up to $n - m - 1$. Uniqueness implies that the search will produce $\hat{\mathbf{S}} = \mathbf{S}$, from which it follows that

$$\hat{\underline{b}} = \mathbf{E}_{\mathbf{S};\mathbf{H}}\underline{y} = \underline{b}. \tag{6.34}$$

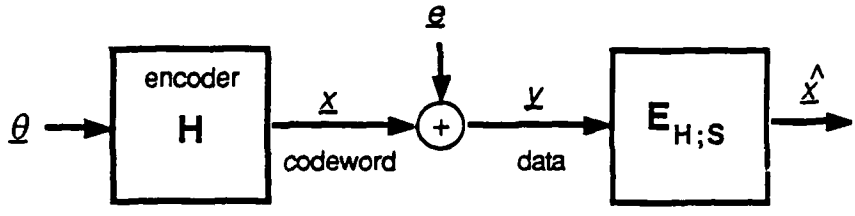


Figure 6.2 A Communication System

This implies that $\hat{\underline{x}} = \underline{x}$; so with probability one, the code can correct $t < n - m$ impulse errors.

**Decoding for Impulse and Background Noise.** Now we consider the case in which minor errors (background noise) may be present in all codeword positions in addition to the major impulse errors. For simplicity of analysis the background noise $\underline{\nu}$ is assumed to be white and Gaussian: $E\{\underline{\nu}\underline{\nu}^H\} = \sigma^2 \mathrm{I}$ ($E\{\cdot\}$ is the expected value operator). In the complex case we assume that the real and imaginary parts are independent white Gaussian random vectors with variance $\frac{\sigma^2}{2}$ in each part. The effect of the background noise on the decoding process is twofold. It corrupts the codeword and perturbs our estimate of the impulse errors.

We deal first with estimation of the impulse errors in the presence of the background noise. Then we will use whatever error correction capacity remains to reduce the background noise effects on the codeword itself. We use the same basic approach for estimating the impulse errors as was used without background noise, with the channel noise $\underline{e}$ now equal to $\underline{b}$ plus $\underline{\nu}$. The new version of Equation 6.28 for the syndrome is

$$\begin{aligned} \underline{z} &= \mathbf{U}^H \underline{y} = \mathbf{U}^H(\mathbf{H}\underline{\theta} + \mathbf{S}\underline{\phi} + \underline{\nu}) \\ &= (\mathbf{U}^H\mathbf{S})\underline{\phi} + \underline{w}; \end{aligned} \tag{6.35}$$

where $\underline{w} = \mathbf{U}^H \underline{\nu}$.

Observe now that Equation 6.35 is a Linear Statistical Model in the syndrome space, for the syndrome $\underline{z}$. Furthermore, with U being orthonormal ($\mathbf{U}^H\mathbf{U} = \mathbf{I}$), the new noise vector $\underline{w}$ retains the whiteness property of $\underline{\nu}$:

$$\begin{aligned} E\{\underline{w}\underline{w}^H\} &= E\{\mathbf{U}^H \underline{\nu}\underline{\nu}^H \mathbf{U}\} = \mathbf{U}^H E\{\underline{\nu}\underline{\nu}^H\}\mathbf{U} \\ &= \mathbf{U}^H(\sigma^2\mathbf{I})\mathbf{U} = \sigma^2\mathbf{U}^H\mathbf{U} = \sigma^2\mathbf{I}. \end{aligned} \tag{6.36}$$

As before we determine $\hat{\mathbf{S}}$ by a search of all selection matrices, but because of $\underline{w}$ we can no longer expect an exact solution. For each $\hat{\mathbf{S}}$ there is an associated $\hat{\underline{\phi}}$, given by Equation 6.29, which minimizes the least squares cost function $||\underline{z} - \mathbf{U}^H\hat{\mathbf{S}}\hat{\underline{\phi}}||^2$. For a given $\hat{t}$ (estimated number of impulses) we consider the best $\hat{\mathbf{S}}$ to be the one for which the minimized cost function is lowest. This gives us one $\hat{\mathbf{S}}$ for each candidate $\hat{t}$. For these, the least squares cost decreases as $\hat{t}$ increases, but the amount of decrease tends to become small for $\hat{t} > t$. This is because the $t$ large impulse errors make much greater contributions to the norm of the syndrome than the smaller background noise errors, and thus tend to be corrected first.

Choosing $\hat{t}$ too small results in uncorrected impulse errors, while choosing it too large inappropriately treats some background noise errors as impulses. The latter reduces our ability to deal appropriately with the white background noise after removal of the estimated impulse noise. We make the final choice of $\hat{t}$, and thus of $\hat{\mathbf{S}}$, by an order selection rule as discussed in Chapter V. Once $\hat{\mathbf{S}}$ and its associated $\hat{\underline{\phi}}$ are determined, the estimated impulse noise vector $\hat{\underline{b}}$ can then be synthesized according to Equation 6.27 and subtracted from the data:

$$\begin{aligned} \hat{\underline{v}} &= \underline{y} - \hat{\mathbf{S}}\hat{\underline{\phi}} \\ &= \mathbf{E}_{\hat{\mathbf{S}}:\mathbf{H}}\underline{y}. \end{aligned} \tag{6.37}$$

The communication system so far is almost identical to the one for impulse noise only, shown in Figure 6.2. The only difference the background noise brings is in the selection criteria

for $\hat{\mathbf{S}}$. Unlike the case without background noise however, the result $\hat{\underline{v}}$ is not yet the codeword estimate. It does not necessarily lie in the code space and we can still remove some of the effects of the background noise.

We now proceed to deal with the effects of the background noise on the codeword. Under the assumption that the impulses have been correctly located (i.e., $\hat{\mathbf{S}} = \mathbf{S}$), $\mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}$ has the properties $\mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}\underline{x} = \underline{x}$ and $\mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}\underline{b} = 0$. In accomplishing the elimination of the impulse noise, though, $\mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}}$ colors the background noise. The noise affecting $\hat{\underline{v}}$ is not white. In our earlier paper [SMB87], we derived the autocorrelation matrix for $\hat{\underline{v}}$, prewhitened $\hat{\underline{v}}$, and then applied a least squares estimator to determine $\hat{\underline{x}}$. This process lead to another oblique projection operator $\mathbf{E}_{\mathbf{R}}$, and the final codeword estimate was obtained by the sequence of two oblique projections

$$\hat{\underline{x}} = \mathbf{E}_{\mathbf{R}}(\mathbf{I} - \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}})\underline{y}. \tag{6.38}$$

As it turns out, further simplification is possible. The product $\mathbf{E}_{\mathbf{R}}(\mathbf{I} - \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}})$ is equal to $\mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}})$, and from the three-way resolution of identity in Chapter II we can write

$$\begin{aligned}
\hat{\underline{x}} &= \mathbf{P}_{\mathbf{H}}(\mathbf{I} - \mathbf{E}_{\hat{\mathbf{S}};\mathbf{H}})\underline{y} \\
&= \mathbf{P}_{\mathbf{H}}(\mathbf{P}_{(\mathbf{H};\mathbf{S})^{\perp}} + \mathbf{E}_{\mathbf{H};\hat{\mathbf{S}}})\underline{y} \\
&= \mathbf{P}_{\mathbf{H}}\mathbf{E}_{\mathbf{H};\hat{\mathbf{S}}}\underline{y} \\
&= \mathbf{E}_{\mathbf{H};\hat{\mathbf{S}}}\underline{y}.
\end{aligned} \tag{6.39}$$

Therefore, the error correction decoder involves one oblique projection and a search for the correct selection matrix. This is exactly the result one would expect from Section 6.1 on least squares parameter estimation in structured noise.

**Simulation Results.** We now present the results of a simulation of the system in Figure 6.2. For this example we have used an $n = 7$ BCH code wherein all codewords have zeros in positions 3, 4, 5 and 6 of their DFT, allowing $m = 3$ information values. The encoder matrix $\mathbf{H}$ for this case is

$$\mathbf{H} = \begin{bmatrix} w^0 & w^0 & w^0 \\ w^0 & w^1 & w^6 \\ w^0 & w^2 & w^5 \\ w^0 & w^3 & w^4 \\ w^0 & w^4 & w^3 \\ w^0 & w^5 & w^2 \\ w^0 & w^6 & w^1 \end{bmatrix}; \qquad w = e^{-j2\pi/7}. \tag{6.40}$$

We fixed $\underline{\theta}$ and $\underline{b}$ for 50 realizations of the background noise $\underline{\nu}$.

$$\underline{\theta} = \begin{bmatrix} 0.5 - j0.25 \\ -0.75 - j0.5 \\ 0.25 + j0.75 \end{bmatrix}, \qquad \underline{b} = \begin{bmatrix} 0 \\ 1.7 - j1.7 \\ 3.9 - j3.9 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.41}$$

The order selection rule was adapted to the complex case from [Sch87].

Signal to noise ratio (SNR) is defined as $\underline{x}^H \underline{x}/n\sigma^2$, where $\sigma^2$ is the variance of each element in $\underline{\nu}$ (real and imaginary parts have variance $\frac{1}{2}\sigma^2$ each). But SNR is irrelevant in the syndrome space where estimation of the impulse errors occurs. The relevant parameter there is the major to minor noise ratio (MMNR) defined as $\underline{b}^H \underline{b}/n\sigma^2$. We express both in dB by taking 10 times the base-10 log.

The plot in Figure 6.3 shows how close the final estimates of the information values were to their true values $\underline{\theta}$. The relative error shown is $\|\underline{\widehat{\theta}} - \underline{\theta}\|^2 / \|\underline{\theta}\|^2$. The MMNR used represents quite substantial background noise, and in most cases $\widehat{\mathbf{S}}$ was correctly determined.

**Conclusions.** We have derived an algorithm for decoding linear block codes over the complex field. First the number and locations of the impulses are determined, effectively identifying a structured noise subspace. Then the received data vector is operated on by the oblique projection operator whose range is the code space (signal subspace) and whose null space contains the identified structured noise subspace. Simulations verify that the decoder performs well.

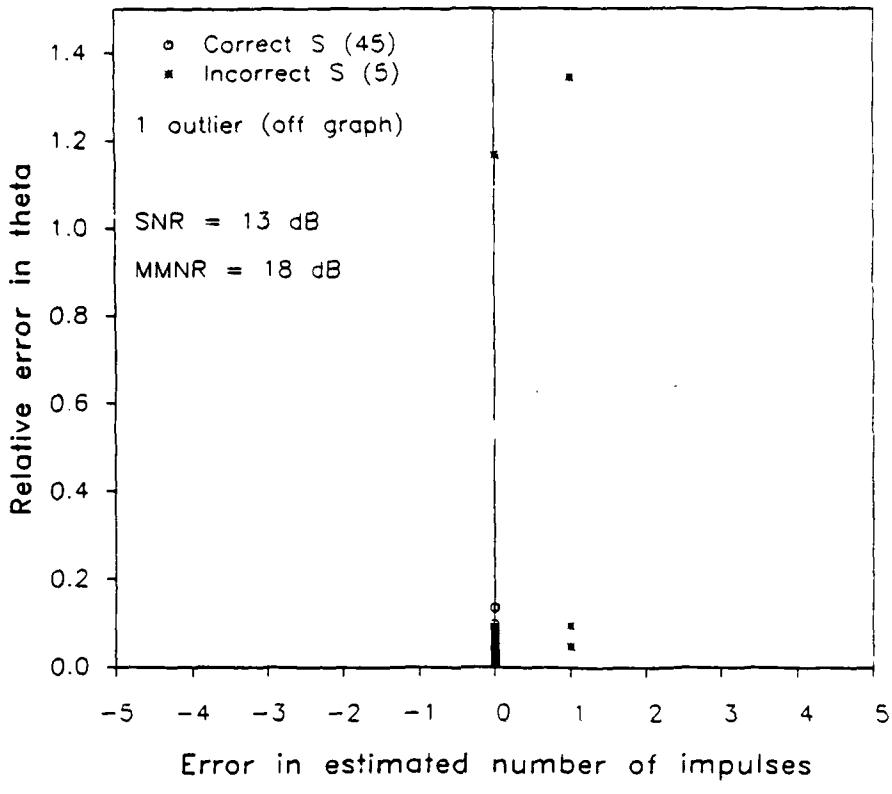


Figure 6.3 Final Estimates of Information Values.

# CHAPTER VII

## Conclusions

We have presented in this dissertation a collection of advancements in digital signal processing theory and practice, based on the use of linear subspaces. Most of the new algorithms are related to our proposed "structured noise model", wherein both the signal and a component of the noise are assumed to lie in low rank linear subspaces. We have argued that linear modeling of noise is often appropriate for the same reasons that linear modeling applies to signals.

Oblique projection operators occur naturally in the solution of estimation problems in structured noise. We consider the primary theme of this dissertation to be the emphasis on oblique projection operators as useful tools in signal processing. To support and develop that notion we have presented research that can be grouped into three areas: mathematical contributions, subspace identification techniques, and estimation problems in structured noise. In the following sections we discuss each of these areas in terms of implications, limitations, and possible extensions.

### 7.1 Mathematical Contributions

Most of the mathematics used here is not new. Since it is mostly of a linear algebraic nature, it can be found in such books as the classic *Matrix Computations* by Golub and Van Loan [GVL89]. But there are a few mathematical results we have not seen elsewhere.

The equations in Chapter II for construction of an oblique projection with a specified range and null space are our own, although they are of such a fundamental nature that we would not be surprised to discover them in earlier works. These equations give two different expressions for the same oblique projection matrix. These expressions are essential to the signal processing applications of oblique projections, both for analysis and implementation.

The natural geometrical interpretation of oblique projections built from our expressions is a three-way resolution of Euclidean space into the components of signal, structured noise, and the remainder of the space. This is expressed mathematically in our three-way resolution of the identity matrix as the sum of two oblique projection matrices and one orthogonal projection matrix.

We have demonstrated a link between oblique projections and orthogonal projections by deriving a coordinate transformation that makes an oblique projection problem into an orthogonal projection problem. This gives researchers the option of studying the properties of the coordinate transformation and combining them with the known properties of orthogonal projections as an indirect approach to the study of oblique projection operators.

The singular values of an oblique projection operator are critically important in ascertaining the effect of that operator on unstructured background noise. We have pointed out that, unlike an orthogonal projection operator, an oblique projection may have singular values larger than unity. If any of those singular values are too large, the benefit gained from elimination of the structured noise may be outweighed by amplification of the background noise. We have quantified that relationship in Chapter VI. We have also discovered a geometric interpretation of the singular values of an oblique projection, giving a relationship in Chapter II that connects them directly to the principal angles between the range and the null space.

The main limitation of the mathematical results on oblique projections is simply that the range (signal subspace) and null space (structured noise subspace plus remaining space) must be disjoint. The intersection between range and null space must include no vector other than the zero vector. In a signal processing context this means that we cannot build a projection operator to completely separate signal and noise when some structured noise vectors are identical to some signal vectors.

## 7.2 Subspace Identification Techniques

Before one can build an oblique projection to separate signals from structured noise, one must know the signal subspace and the structured noise subspace. While it may occasionally

be possible to determine both subspaces in advance through theoretical considerations, it is very important in many cases to be able to identify these subspaces from observed data and limited knowledge of the nature of the signals.

The principle of Maximum Likelihood is an old standby for estimation problems. We have adopted it in many of our subspace identification techniques. However, we recognize that it is not always an appropriate way to solve estimation problems. In Chapter III we give a critical evaluation of the Maximum Likelihood principle and an example using a quadratic equation to illustrate the pitfalls of blind application of ML. Our purposes in including this evaluation of ML are to warn the reader against blind application of our subspace identification techniques, to suggest the adaptation of our techniques to some kind of MAP estimation rule where appropriate, and to make a statement of our reservations about the careless use of ML.

The easiest class of subspace identification problems is signal subspace identification without structured noise. In this class we have presented improvements on existing algorithms for the nonparametric problem whose solution is obtained with the SVD, and for the parametric problem with complex exponential signal modes. The latter problem frequently needs to be solved under constraints regarding the location of the roots, and we have presented an extension of the "KiSS" algorithm that incorporates constraints. We have also found elegant expressions and interpretations for the gradient and Hessian of the KiSS objective function for complex data and parameters. A Newton algorithm can be implemented using these gradient and Hessian expressions.

The next most difficult subspace identification problem is that of identifying a structured noise subspace when you know your signal subspace. The coding example of Chapter VI falls into this category. The "KiSS with structured noise" algorithm of Chapter IV solves an equivalent problem, since it identifies a signal subspace when the structured noise subspace is known.

The most difficult problem is the simultaneous identification of both signal and structured noise subspaces. It is clear that this cannot be done without some prior knowledge that

will allow us to distinguish between the two. We have approached this problem by assuming the signal is a sum of complex exponentials and the structured noise is impulsive. Under these conditions the KiSS algorithm with structured noise can be used within a combinatorial search to locate the noise impulses. The results have been rather unsatisfactory.

A problem similar to subspace identification is that of updating existing signal and/or noise subspaces based on new data. For signal subspace updates without structured noise we have presented in Chapter III a technique using Total Least Squares. We have then derived an extension of that technique that allows simultaneous updates of both signal and noise subspaces. This technique is suitable for adaptively identifying slowly varying subspaces.

In Chapter V we have address the issue of order selection when identifying subspaces. We have proposed a new order selection rule for rank reduction in the Linear Statistical Model without structured noise based on two sequential applications of the principle of Maximum Likelihood. The new rule performs only slightly better than existing rules and is rather expensive computationally. The most intriguing aspect of this result is the concept of multiple applications of ML.

We have also derived in Chapter V a Bayes hypothesis test for order selection in the identification of structured noise subspaces. Even with its solid theoretical foundation, the Bayes rule performs only a little better than existing rules that are rather more ad hoc.

## 7.3 Estimation Problems in Structured Noise

We have derived signal estimators and parameter estimators for several structured noise problems, distinguished by the amount of statistical information available about the underlying processes. In the first case, without probability densities on the signal or noise, it was found that the least squares estimator of a signal in structured noise is the oblique projection of the received data vector onto the signal subspace with the structured noise subspace in the null space of the projection.

When a Gaussian density function is placed on the parameter vector of the structured noise term in the model the solution still involves an oblique projection whose range is the

signal subspace. The null space in this case is not perfectly aligned with the structured noise, but tends toward the structured noise subspace when the structured noise dominates the background noise. On the other hand, as the relative structured noise power decreases, the oblique projection converges to the orthogonal projection onto the signal subspace.

Finally we have given an example application in the decoding of linear block codes over the Real/Complex number field. This application illustrates both subspace identification and parameter estimation aspects of structured noise processing.

## 7.4 Extensions

When identifying impulse noise subspaces, the combinatorial search for the best selection matrix can become impractical for large data lengths. Further study is warranted on ways to reduce the search space. For example, with BCH codes the Prony type method of Kumaresan, Tufts, and Scharf [KTS84] may be used to obtain a starting point for the search.

The simultaneous identification of signal and structured noise subspaces has proven to be a difficult problem, and our current approaches are not adequate. We explain in Chapter IV that there appears to be a connection between the subjective smoothness of the KiSS objective function and the correctness of a candidate structured noise subspace. We suggest future research to quantify and exploit this connection and try to develop a better method of simultaneous subspace identification.

Signal detection problems in structured noise remain open as a possible extension of this research. The coordinate transformation relating oblique and orthogonal projections should be of use here in specifying invariance conditions for the detector. For example the group of transformations which characterize invariance to structured noise and to rotation of the signal within the signal subspace is given by

$$g(\underline{y}) = \mathbf{F}^{-1}\left[(\mathbf{U_H R_H U_H}^H + \mathbf{P}_{\mathbf{H}^\perp})\mathbf{F}\underline{y} + \mathbf{U}_{\mathbf{H}^\perp}\underline{v}\right], \tag{7.1}$$

where $\mathbf{F}$ is the coordinate transformation of Chapter II, $\mathbf{R_H}$ is any rotation in the signal subspace, $\mathbf{U_H}$ is an orthogonal span for the signal subspace (H), $\mathbf{U}_{\mathbf{H}^\perp}$ is an orthogonal span of

$(\mathbf{H})^{\perp}$, and $\underline{\psi}$ is any $(n-m)$-vector. The maximal invariant statistic for detecting an unknown signal in a known signal subspace, with known noise variance, under these invariance conditions is

$$\alpha = \frac{\underline{y}^H \mathbf{E}_{\mathbf{H};\mathbf{S}}{}^H \mathbf{E}_{\mathbf{H};\mathbf{S}} \underline{y}}{n\sigma^2}. \tag{7.2}$$

Detection statistics could be derived for other cases of known versus unknown signal, noise variance, and structured noise.

The ML esitmators of noise variance in Sections 3.2 and 3.4 provide an interesting area for further study. As mentioned at the end of Section 3.2, the normalization by $n$ would seem to be better replaced by a normalization by $n-r$, since we have noise power in only $n-r$ dimensions available for estimating $\sigma^2$. A study of the mean and variance of these ML estimators of $\sigma^2$ may reveal a correctable bias. The difficulty is in finding the expected value of a singular value of a matrix. The way around this difficulty may be in finding an alternate expression for the estimator that does not involve singular values.

The possible role of oblique projections in spectral estimation should be examined. This would include investigation of quadratic forms in oblique projections, and possible frequency domain analogs of oblique projections.

## BIBLIOGRAPHY


[Aka74] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. on AC*, vol. AC-19, pp. 716-723, 1974.

[BeS88] R. T. Behrens and L. L. Scharf, "Parameter Estimation in the Presence of Low Rank Noise," *Proceedings of the Twenty-Second Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1988.

[BeS90] R. T. Behrens and L. L. Scharf, "An Order Selection Rule for Rank Reduction in the Linear Statistical Model," *IEEE International Symposium on Information Theory*, San Diego, CA, Jan. 1990.

[BrM86] Y. Bresler and A. Macovski, "Exact Maximum Likelihood Parameter Estimation of Superimposed Exponential Signals in Noise," *IEEE Trans. on ASSP*, vol. ASSP-34, pp. 1081-1089, Oct. 1986.

[Buc87] K. M. Buckley, "Spatial/Spectral filtering with Linearly Constrained Minimum Variance Beamformers," *IEEE Trans. on ASSP*, vol. ASSP-35, pp. 249-266, Mar. 1987.

[Dav79] P. J. Davis, *Circulant Matrices*, Wiley, New York, NY, 1979.

[Dem89] C. J. Demeure, "Fast QR Factorization of Vandermonde Matrices," *Linear Algebra and Its Applications*, 1989.

[DeS83] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[DeS87] C. J. Demeure and L. L. Scharf, "Linear Statistical Models for Stationary Sequences and Related Algorithms for Cholesky Factorization of Toeplitz matrices," *IEEE Trans. on ASSP*, vol. ASSP-35, pp. 29-42, Jan. 1987.

[Dun86] M. J. Dunn, "Sufficiency and Invariance Principles Applied to Four Detection Problems," M.S. Thesis, University of Colorado, Boulder, 1986.

[EcY36] C. Echart and G. Young, "The Approximation of One Matrix by Another of Lower Rank," *Psychometrika*, vol. 1, pp. 211-218, Sept. 1936.

[Evf73] A. G. Evans and R. Rishl. "Optimal Least Squares Time-Domain Synthesis of Recursive Digital Filters," *IEEE Trans. on Audio and Electroacoustics*. vol. AU-21, pp. 61-65. Feb. 1973.

[Fuc88] J. Fuchs, "Estimating the Number of Sinusoids in Additive White Noise," *IEEE Trans. on ASSP*. vol. ASSP-36, pp. 1846-1853, Dec. 1988.

[GVL89] G. H. Golub and C. F. Van Loan, *Matrix Computations*. 2nd ed., Johns Hopkins University Press. Baltimore. MD. 1989.

[Kai80] T. Kailath, *Linear Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1980.

[KaW89] S. Kayalar and H. L. Weinert, "Oblique Projections: Formulas, Algorithms, and Error Bounds," *Mathematics of Control, Signals, and Systems,* vol. 2, no. 1, pp. 33-45, 1989.

[KSS86] R. Kumaresan, L. L. Scharf and A. K. Shaw, "An Algorithm for Pole-Zero Modeling and Spectral Analysis," *IEEE Trans. on ASSP,* vol. ASSP-34, pp. 637-640, June 1986.

[KTS84] R. Kumaresan, D. W. Tufts and L. L. Scharf, "A Prony Method for Noisy Data: Choosing the Signal Components and Selecting the Order in Exponential Signal Models," *Proceedings of the IEEE,* vol. 72, pp. 230-233, Feb. 1984.

[Kum85] R. Kumaresan, "Rank Reduction Techniques and Burst Error-Correction Decoding in Real/Complex Fields," *Proceedings of the Nineteenth Asilomar Conf. on Circuits, Systems and Computers,* Pacific Grove, CA, 1985.

[LaH74] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems,* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[Lan69] H. O. Lancaster, *The Chi-squared Distribution,* John Wiley and Sons, New York, p. 118, 1969.

[Lor39] E. R. Lorch, "On a Calculus of Operators in Reflexive Vector Spaces," *Trans. Amer. Math. Soc.,* vol. 45, pp. 217-234, 1939.

[MaN88] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics,* Wiley, Chinchester, 1988.

[Mar84] T. G. Marshall, Jr., "Coding of Real-number Sequences for Error Correction: A Digital Signal Processing Problem," *IEEE J. Sel. Areas of Communication,* vol. SAC-2, pp. 381-392, Mar. 1984.

[Mar85] T. G. Marshall, Jr., "Codes for Error Correction Based upon Interpolation of Real-number Sequences," *Proceedings of the Nineteenth Asilomar Conference on Circuits, Systems, and Computers,* Pacific Grove, CA, 1985.

[Mar86] T. G. Marshall, Jr., "Signal Restoration Viewpoints for Estimating Errors in Discrete-time Signals," *Procedings of the Twentieth Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, 1986.

[Mar87] T. G. Marshall, Jr., "Removing Noise Pulses from Frequency Constrained Signals," *IEEE International Conf. on Communications,* June 1987, pp. 997-1000.

[Mat90] P. Mathys, "A Channel Model and Corresponding MAP-Rule for Linear Block Codes over the Reals," submitted to *IEEE Trans. on Commun.,* June 1990.

[McC89] J. H. McClellan, "Exact Equivalence of the Steiglitz-McBride Iteration and IQML," submitted to *IEEE Trans. on ASSP,* 1989.

[Mey67] P. L. Meyer, "The Maximum Likelihood Estimate of the Non-Centrality Parameter of a Non-Central $X^2$ Variate." *Journal of the American Statistical Association,* vol. 62, pp. 1258-1264, 1967.

[Mur37] F. J. Murray, "On complementary manifolds and projections in $L_p$ and $l_p$," *Trans.*

*Amer. Math. Soc.*, vol. 41, pp. 138-152, 1937.

[Pap84] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed., McGraw-Hill, New York, NY, 1984.

[Poo88] H. V Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, New York, NY, 1988.

[Pro1795] R. Prony, "Essai Expérimental et Analytique Sur les lois de la Dilatabilité des fluides élastiques et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures," *L'ecole Polytechnique, Paris.* vol 1, no. 2, pp. 24-76, 1795.

[RiB76] D. C. Rife and R. R. Boorstyn, "Multiple Tone Parameter Estimation from Discrete Time Observations," *Bell Syst. Tech. J.*, vol. 55, pp. 1389-1410, Nov. 1976.

[Sch87] L. L. Scharf, ed. by J. Lacoume, T. Durrani and R. Stora, "Course 2: Topics in Statistical Signal Processing," *Les Houches, Session XLV, 1985*, Elsevier Science Publishers B. V., 1987.

[Sch91] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Addison-Wesley, Reading, MA, 1991.

[ScS86] L. L. Scharf and James Storey, "Rank Reduction and Order Selection for Parametric Spectrum Models," *Third ASSP Workshop on Spectrum Estimation and Modeling*, Boston, MA, Nov. 17-18, 1986.

[ScT87] L. L. Scharf and D. W. Tufts, "Rank Reduction for Modeling Stationary Signals," *IEEE Trans. on ASSP*, vol. ASSP-35, pp. 350-354, Mar. 1987.

[Sle78] D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - V: The Discrete Case," *Bell Syst. Tech. J.*, vol. 57, pp. 1371-1430, May 1978.

[SMB87] L. L. Scharf, P. Mathys and R. T. Behrens, "Rank Reduction for Decoding Linear Block Codes over the Complex Field," *Proceedings of the Twenty-First Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, 1987.

[StN88] D. Starer and A. Nehorai, "Maximum Likelihood Estimation of Exponential Signals in Noise Using a Newton Algorithm," *IEEE ASSP Workshop on Spectrum Estimation and Modeling*, Minneapolis, MN, pp. 240-245, Aug. 1988.

[Tua88] P. D. Tuan, "Estimation of Autoregressive Parameters and Order Selection for ARMA Models," *J. Time Series Analysis*, vol. 9, no. 3, pp. 265-279, 1988.

[TuK82] D. W. Tufts and R. Kumaresan, "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 975-989, Sept. 1982.

[WaK85] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," *IEEE Trans. on ASSP*, vol. ASSP-33, pp. 387-392, Apr. 1985.

[Wol83] J. K. Wolf, "Redundancy, the Discrete Fourier Transform, and Impulse Noise Cancellation," *IEEE Trans. Commun.*, vol. COM-31, pp. 458-461, Mar. 1983.

[Zol87] M. D. Zoltowski, "Signal Processing Applications of the Method of Total Least Squares,"

*Proceedings of the Twenty-First Asilomar Conference on Signals, Systems, and Computers,* Pacific Grove, CA, 1987.

# APPENDIX A

## MATLAB Code for the KiSS Algorithm

The following code implements the KiSS algorithm with constraints and optionally with structured noise. The main function KISS uses subfunctions AAINV, BUILDB, KSSDATA, MOD, and ORTHPROJ also listed here. For phase 2, the KISS function can use UMSOLVE, available through the MATLAB user group.

```
function [a,b,errnorm,thpath,T,c,iter1,iter2,termcode,ph1a,errnorm1] = ..
kiss(y,m,constr,ph2,tol1,tol2,usecirc,a0,S,useglob,fdebug)
% [a,b,errnorm,thpath,T,c,iter1,iter2,termcode] = ..
%         kiss(y,m,constr,ph2,tol1,tol2,usecirc,a0,S,useglob,fdebug)
%
%     This function implements the KiSS algorithm for estimation of
% parameters of superimposed sinusoids.    Inputs:
%   y - observed data vector (required).
%   m - model order (required).
%   constr - Constraint option:
%       [1] = Nontriviality constraint only.
%       2 = conjugate symmetry constraint.
%       3 = Real coefficients.
%       4 = Real and symmetric.
%       5 = conjugate symmetry plus "project to circle" contraints.
%       6 = real symmetry plus "project to circle" constraints.
%   ph2 - Phase 2 option:
%       [0] = No phase 2.
%       1 = Phase 2 by constrained Evans/Fishl method.  (is it right?)
%       2 = Phase 2 by Newton's method.
%   tol1 - Phase 1 convergence tolerance (default=eps^(1/3)).
%   tol2 - Phase 2 convergence tolerance (default=sqrt(eps)).
%   usecirc - permission to use circulant technique for inverse.
%   a0 - starting point for AR (denominator) parameters.
%   S - structured noise subspace.
%   useglob - flag indicating that the following global variables
%             have been declared before calling kiss:
%                kissglobY,kissgloby,kissglobT,kissglobc,kissglobYa,
%                kissglobQ,kissglobA,kissglobu,kissglobnhat
%             Doing this saves much computation for Newton phase 2.
%   fdebug - Debug mode flag.    Show progress when set.
%
% Outputs:
%   a - AR (denominator) parameters.
```

```
%   b - MA (numerator) parameters.
%   errnorm - fitting error norm.
%   thpath - sequence of iterates of theta.
%   T,c  - constraint model, a = T*theta+c.
%   iter1,iter2 - number of iterations in each phase.
%   termcode - from the Newton phase 2 (see umsolve).
%   ph1a - AR parameters after phase 1.
%   errnorm1 - fitting error norm after phase 1.
%
% Richard T. Behrens, May 1990, August 1990.
%
if (nargin < 11)
fdebug = 0;
end
if (nargin < 2)
clc
snr = input('Enter snr for kssdata:  ');
[y,sigma] = kssdata(snr);
n = 25;
m = input('Enter model order:  ');
fdebug = 2;
else
[n,k] = size(y);
if (k > 1)
if (n > 1)
error('First input argument must be a vector.')
end
y = y.';
n = k;
end
end
if (mod(m,2) == 1)
isodd = 1;
q = (m-1)/2;
else
isodd = 0;
q = (m-2)/2;
end
if (fdebug==2)
constr = input('Enter constraint option:  ');
ph2 = input('Enter phase 2 option:  ');
tol1 = input('Enter phase 1 tolerance:  ');
tol2 = input('Enter phase 2 tolerance:  ');
usecirc = input('Permission to use circulant inverse:  ');
S = input('Structured noise subspace:  ');
useglob = 0;
fdebug=1;
else
if (nargin<3), constr=[]; end
if (nargin<4), ph2 = []; end
if (nargin<5), tol1 = []; end
if (nargin<6), tol2 = []; end
if (nargin<7), usecirc = []; end
if (nargin<8), a0 = []; end
```

```
if (nargin<9), S = []; end
if (nargin<10), useglob = []; end
end
%
% Set up defaults.
%
if isempty(constr), constr=1; end
if isempty(ph2), ph2 = 0; end
if isempty(tol1), tol1 = eps^(1/3); end
if isempty(tol2), tol2 = sqrt(eps); end
if isempty(usecirc),
usecirc = ~(((constr==4)|(constr==6)) & (mod(n,2)==1) & isodd);
usecirc = usecirc & (n > 6*m); % An approximation of when it pays.
usecirc = 0; % Override and disable circulant inverses.
end
if isempty(useglob), useglob = 0; end
if ((~ isempty(S))&(ph2~=0))
disp('Cannot use phase 2 with structured noise.')
ph2 = 0;
end
Y = toeplitz(y((m+1):n),y((m+1):-1:1));
if (constr==1)
T = [zeros(1,2*m); eye(m) j*eye(m)];
c = [1; zeros(m,1)];
end
if ((constr==2)|(constr==5))
if isodd
T = [[zeros(1,q); eye(q); jay(q); zeros(1,q)] [j*eye(q+1); -j*jay(q+1)]];
c = [1; zeros(2*q,1); 1];
else
T = [[zeros(1,q+1); eye(q+1); jay(q) zeros(q,1); zeros(1,q+1)] ..
[j*eye(q+1); zeros(1,q+1); -j*jay(q+1)]];
c = [1; zeros(2*q+1,1); 1];
end
end
if (constr==3)
T = [zeros(1,m); eye(m)];
c = [1; zeros(m,1)];
end
if ((constr==4)|(constr==6))
if isodd
T = [zeros(1,q); eye(q); jay(q); zeros(1,q)];
c = [1; zeros(2*q,1); 1];
else
T = [zeros(1,q+1); eye(q+1); jay(q) zeros(q,1); zeros(1,q+1)];
c = [1; zeros(2*q+1,1); 1];
end
end
Ytilde = Y*T;
[k,q] = size(T);    % Hereafter, q is the length of theta.
if isempty(a0)      % start with Prony (or constrained Prony).
Q = Y'*Y;
theta = -real(T'*Q*T)real(T'*Q*c);
else
```

```
theta = pinv(T)*(a0-c);
if any(~ near(T*theta+c,a0))
errmsg('initial parameter a0 did not satisfy constraints')
end
end
if (constr>=5)
theta = projroot(theta,T,c);
end
thpath = theta;
iter1=0; iter2=0;
if fdebug
disp('Ready to start phase 1 iterations.')
keyboard
end
if (tol1>0)
old = zeros(q,1);
while ((~ all(near(theta,old,tol1)))&(iter1<100))  % phase 1 iteration
if fdebug
clc
disp(['Phase 1, iteration ' num2str(iter1) '.'])
disp('theta = ')
theta
pause(5)
clg
plot(thpath')
pause(5)
end
old = theta;
if isempty(S)
Q = Y'*aainv(theta,T,c,usecirc,n)*Y;
else
[aai,A] = aainv(theta,T,c,usecirc,n);
ap = aai*A'; aps = ap*S;
Q = Y'*(aai - aps*inv(S'*A*aps)*aps')*Y;
end
theta = -real(T'*Q*T)real(T'*Q*c);
thpath = [thpath theta];
iter1 = iter1 + 1;
end
end
ph1a = T*theta+c;
A = buildb(ph1a,n);
e = orthproj(A)*y;
errnorm1 = norm(e);
if (ph2==1)
old = zeros(q,1);
while (~ all(near(theta,old,tol2)))      % phase 2 iteration
if fdebug
clc
disp(['Phase 2, iteration ' num2str(iter2) '.'])
disp('theta = ')
theta
pause(5)
clg
```

```
plot(thpath')
pause(5)
end
old = theta;                          % (not sure if phase 2 is correct
[Q,A] = aainv(theta,T,c,usecirc,n);  % for T other than idendity).
W = A*Q;
d = A'*y;
L = zeros(n,1);
for k=1:m
dak = zeros(m+1,1); dak(k+1) = 1;
dAk = buildb(dak,n);
dWk = (dAk - W*(dAk'*A + A'*dAk))*Q;
L(:,k+1) = dWk*d;
end
U =(L' + Y'*W')*W;
Q = U*Y;
theta = -real(T'*Q*T)real(T'*Q*c);
thpath = [thpath theta];
iter2 = iter2 + 1;
end
end
if (ph2==2)       % Phase 2 by Newton's method.
fparam = [Y; T.'; c.'; usecirc n useglob zeros(1,m-2)];
details = zeros(17,1);
if fdebug, details(1) = 1; end   % (print trace)
details(2)=2;  % (use the hookstep)
details(3) = 1; % (don't use secant update)
details(4) = 1; % (use analytic gradient)
details(16) = 1; % (scale by starting point)
details(17) = 1; % (use analytic hessian)
[theta,termcode,path] = ..
umsolve('kissf',theta,details,fparam,'kissg','kissh');
iter2 = length(path)-2;
thpath = [thpath path(2:(length(path)-1),:)'];
if ((termcode==2)|(termcode==3))
% Because the desired minimum is rather deep and narrow we sometimes
% need to ease up on gradtol (so the gradient needn't get so small),
% and on steptol (so we can take very tiny steps).
%      disp('Tryinging harder to converge.')
details(4) = 0; % (don't use analytic gradient - for kicks)
details(8) = eps^(1/3); % (ease up on gradtol)
details(9) = 2*eps;     % (steptol--allow very tiny steps)
details(17) = 0; % (don't use analytic hessian - for kicks)
[theta,termcode,path] = ..
umsolve('kissf',theta,details,fparam,'kissg','kissh');
iter2 = iter2 + length(path)-2;
thpath = [thpath path(2:(length(path)-1),:)'];
end
end
if fdebug
clg
plot(thpath')
title('Final KiSS convergence path.')
pause
```

```
clg
unitcirc
hold on
plot(roots(T*theta+c),'x')
title('Final KiSS root positions')
pause(5)
hold off
axis('normal')
end
%
% Compute the coefficients and the final error
%
a = T*theta+c;
A = buildb(a,n);
e = orthproj(A)*y;
h = y - e;
H1 = toeplitz(h(1:m),[h(1); zeros(m,1)]);
b = H1*a;
errnorm = norm(e);
```

```
function [Q,A] = aainv(theta,T,c,usecirc,n)
% [Q,A] = aainv(theta,T,c,usecirc,n) forms the toeplitz coefficient matrix
% A from the prediction polynomial a=T*theta+c.    Then Q, the inverse of
% A'*A, is computed either directly (if usecirc=0) or by a fast algorithm
% using a circulant matrix (if usecirc=1).
%
%
% Richard T. Behrens, May, 1990.
%
[m,q] = size(T);
m = m - 1;
A = buildb(T*theta+c,n);
[n,k]=size(A);
if usecirc
v = A(:,1)'*A;
c = v + [0 conj(v(k:-1:2))];
Cinv = circinv(c);
qq = v((m+1):-1:2);
U = [[eye(m); zeros(k-m,m)] ..
[zeros(k-m,m); hankel([zeros(m-1,1); qq(1)], qq)]];
VC = [[zeros(m,n-2*m) toeplitz([conj(qq(1)); zeros(m-1,1)], ..
conj(qq))]; [rot90(eye(m)) zeros(m,k-m)]]*Cinv;
Q = Cinv + Cinv*U/(eye(2*m) - VC*U)*VC;
else
Q = inv(A'*A);
end
```

```
function B = buildb(b,n)
% B = buildb(b,n)
%
% This function builds the toeplitz matrix B from the coefficients b.
% The result B spans the perp-space to a signal built from powers of
% the roots of the polynomial b.     The size of B is n by n-m where m
% is the degree of polynomial b.
%
% This B is as defined by Bresler & Macovski, and is the hermetian
% transpose of the B defined by Kumaresan, Scharf and Shaw.
%
m =length(b)-1;
B =toeplitz([flipud(conj(b)); zeros(n-m-1,1)],[conj(b(m+1)); zeros(n-m-1,1)]);
```

```
function [y,nnorm] = kssdata(snr,fixed,n)
%
% [y,sigma] = KSSDATA(SNR,fixed,n) for fixed = 0, or
% [y,nnorm] = KSSDATA(SNR,fixed,n) for fixed = 1
% generates a signal x(i) according to the model given in the Kumaresan,
% Scharf and Shaw paper (same data as Tufts & Kumaresam).    It generates
% samples for i = 0:(n-1) of
%
%  x(i) = exp(j*omega1*i) + exp(j*pi/4)*exp(j*omega2*i)
%
% with omega1 = 2*pi*0.52
%      omega2 = 2*pi*0.50
%
% Noise is added to obtain y(i), with any desired input SNR.  If fixed=0
% the noise is gaussian with the right EXPECTED snr, if fixed=1 the noise
% is sperically symmetrical with the right EXACT snr.
%
if (nargin < 2)
fixed = 0;
end
if (nargin < 3)
n = 25;
end
omega1 = 2*pi*0.52;
omega2 = 2*pi*0.50;
x1 = exp(j*omega1*(0:(n-1))');
x2 = exp(j*pi/4)*exp(j*omega2*(0:(n-1))');
x = x1 + x2;
if (nargin < 1)
snr = input('Enter desired input signal to noise ratio:  ');
end
if fixed
nnorm = sqrt((x1'*x1)/(10^(snr/10)));
else
sigma = 1/sqrt(2*10^(snr/10));
end
rand('normal')
noise = rand(n,1) + j*rand(n,1);
if fixed
noise = (nnorm/norm(noise))*noise;
else
noise = sigma*noise;
end
y = x + noise;
if (~ fixed)
nnorm = sigma;   % return sigma instead.
end
```

```
theta=theta;function [m] = mod(x,n)
%Computes x modulo n.    If x or n is not an integer, it is first rounded.
%
%Neil H. Endsley 4/87.
%
y = round(x);
k = round(n);
m = y - k*fix(y/k);
```

```
function [p,k] = orthproj(h)
%
%   [P,K] = ORTHPROJ(H)
%   P = ORTHPROJ(H)
%
% Returns the orthogonal projection matrix P whose range is identical to
% the range of H. Optionally returns K, the rank of P.  A projection is
% eidempotent and hermetian symmetric (i.e.  P*P=P and P'=P).  If H is
% full rank the projection is computed directly (without factorization).
% The SVD is used if H is rank deficient.
%
% Richard T. Behrens      May 22, 1987.
%
k=rank(h);
[m,n]=size(h);
if k==n
p=(h/(h'*h))*h';
else
[u,s,v] = svd(h,0);
p = u(:,1:k) * u(:,1:k)';
end
```