DESIGN AND ANALYSIS OF ADAPTIVE NOISE SUBSPACE ESTIMATION ALGORITHMS

Yang LU

(*B. Eng. and B. A, Tianjin University, China*)

A THESIS SUBMITTED FOR THE DEGREE OF PH. D.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2009

to my parents

Acknowledgements

I would like to express my sincere thanks to my main supervisor Prof. Samir Attallah for his constant support, encouragement, and guidance, throughout my stay at NUS. I am thankful to my co-supervisor Dr. George Mathew for his support, advice and involvement in my research. I also thank Prof. Karim Abed Meriam for the many helpful discussions I had with him on my research work.

I would like to thank Mr. David Koh and Mr. Eric Siow for help and support. Thanks to all the members of the former Open Source Software Lab, Communications Lab and ECE-I^{2}R Lab for their invaluable friendship and time for discussion on research problems. I would like to thank my flatmates for their company and the joy they brought. I express my sincere gratitude to NUS, for giving me the opportunity to do research and for supporting me financially through NUS Research Scholarship.

Finally, I am forever grateful to my dear parents for their understanding and support during all these years. And I want to thank my boyfriend Zhang Ning, for sharing the joys as well as disappointments.

Contents

Summary

In this thesis, several adaptive noise subspace estimation algorithms are analyzed and tested. Adaptive subspace estimation algorithms are of importance because many techniques in communications are based on subspace approaches. To avoid the cubic-order computational complexity of the direct eigenvalue decomposition which makes real-time implementation impossible, many adaptive subspace algorithms which need much less computational effort have been proposed. Among them, there are only a few limited noise subspace estimation algorithms as compared with signal subspace estimation algorithms. Moreover, many of the existing noise subspace estimation algorithms are either unstable or nonrobust. Therefore, the aim of this thesis is to develop and analyze stable low cost noise subspace estimation algorithms.

To shed light on how to obtain stable results for noise subspace algorithms, the propagation of orthogonality error for FRANS (fast Rayleigh's quotient based adaptive noise subspace) algorithm is examined in the mean and in the mean-square sense. It is shown that FRANS suffers from numerical instability since its accumulated numerical errors grow geometrically. Then, an upper bound on the orthogonality error is derived for the Householder based FRANS (HFRANS) algorithm, which is numerically much more stable than FRANS algorithm.

To further improve the performance of HFRANS, a gradient adaptive step-size strategy is proposed. One drawback of such a strategy is the difficulty in choosing a proper initial value and convergence rate for the step-size update. Hence, we propose an optimal step-size strategy, which addresses the initialization issue. The proposed step-size strategies can also be applied on other noise and signal subspace estimation algorithms.

To speed up the convergence rate of adaptive subspace estimation algorithms, a diagonal matrix step-size strategy is proposed, which leads to a set of decoupled noise (or signal) subspace vectors that can be controlled individually. This results in better performance of the algorithms.

Finally, a hardware friendly approach, which is free from square root or division operations is proposed to stabilize FRANS while retaining its low computational complexity. This approach is suitable for VLSI (very large scale integration) implementation. An ordinary differential equation (ODE) based analysis is provided to examine the stability of the proposed algorithm. This analysis shows that the proposed algorithm is stable on the manifold and bounded at the equilibrium point.

List of Tables

List of Figures

List of Abbreviations

AMEX	adaptive minor component extraction
API	approximated power iterations
AWGN	additive white Gaussian noise
BER	bit error rate
CDMA	code division multiple access
DBPSK	differential binary phase shift keying
DFT	discrete Fourier transform
EVD	eigenvalue decomposition
FDMA	frequency division multiple access
FDPM	fast data projection method
FOOja	fast orthogonal Oja
FRANS	fast Rayleigh's quotient based adaptive noise subspace algorithm
GHFRANS	HFRANS with gradient adaptive step-size
GSM	global system for mobile communications
HFRANS	FRANS with Householder transformation
IDFT	inverse discrete Fourier transform
LS	least square
MALASE	maximum likelihood adaptive subspace estimation
MC	minor component

MCA	minor component analysis
MC-CDMA	multi-carrier code division multiple access
ML	maximum likelihood
MNS	minimum noise subspace
MOja	modified Oja's algorithm
MSA	minor subspace analysis
MSE	mean square error
NFQR	normalized fast Rayleigh's quotient algorithm
NIC	novel information criterion
NOOja	normalized orthogonal Oja
ODE	ordinary differential equation
OHFRANS	HFRANS with optimal step-size
PAST	projection approximation subspace tracking
PC	principal component
PCA	principal component analysis
PSA	principal subspace analysis
RLS	recursive least squares
SFRANS	stabilized FRANS
SMSR	self-stabilized minor subspace rule
SVD	singular value decomposition
TDMA	time division multiple access
VLSI	very large scale integration
YK	Yang and Kaveh's algorithm

List of Symbols and Notations

$\mathbf{x}$	denotes vector
$\mathbf{X}$	denotes matrix
$(\cdot)^*$	conjugate operator
$(\cdot)^H$	Hermitian transpose operator
$(\cdot)^T$	transpose operator
$(\cdot)_i$	ith element of a vector
$[\cdot]_{i,j}$	(i,j)th element of a matrix
$\lvert \cdot \rvert$	absolute value of a scalar
$\lVert \cdot \rVert$	Euclidean norm of a vector
$\lVert \cdot \rVert_F$	Frobenius norm of a matrix
$\mathcal{C}$	complex line
$\mathcal{C}^N$	$N\times$ 1 complex-valued vector
$\mathcal{C}^{N\times P}$	$N\times$ P complex-values matrix
angle$(\cdot)$	angle operator
diag$(\cdot)$	diagonal matrix operator
$E[\cdot]$	expectation operator
$\mathbf{I}_N$	$N \times N$ identity matrix
N	data length

$O(\cdot)$ multiplications	order of the number of multiplications required by each algorithm
P	dimension of noise subspace
$\mathcal{R}$	real line
$\mathrm{Re}(x)$	real part of the complex number x
$\mathrm{Tr}(\cdot\ )$	trace operator
$\mathrm{Tri}(\cdot)$	denotes that only the upper (or lower) triangular part is calculated and its Hermitian transposed version is copied to the another lower (or upper)triangular part
$\mathbf{W}(i)$	estimate of noise subspace at ith instant
π	permutation of $\{1, \cdots, N\}$
1_N	denote a permutation π if $\pi(i) = i$ for all $i = 1, \cdots, N$

Chapter 1

Introduction

In this opening chapter of the thesis, we briefly touch upon the field of wireless communications to establish a broader application context for subspace estimation. Following this, we present a brief review of the literature to motivate the research problem undertaken in the thesis. We conclude this chapter with a summary of the main contributions of this thesis and organization of the thesis.

1.1 Introduction

Wireless communication is a rapidly growing segment of the communications industry, with the potential to provide high-speed and high-quality information exchange between portable devices located anywhere in the world [48, Chap 1]. The main factor driving this tremendous growth in wireless coverage is that it does not need the setting up of expensive infrastructure such as copper or fiber lines and switching equipment.

The first generation of public cellular wireless communication systems was the analog mobile phone systems introduced in the early 1980s. They were followed by the second generation systems in the late 1980s, such as GSM (Global system for mobile communications). These systems are based on digital modulation techniques which provided better spectral efficiency. The first generation systems were mainly voice oriented, whereas the second generation systems can also provide low rate data transmission. Emerging requirements for higher data rates and better spectrum efficiency are the primary challenges faced by the third generation systems. Because the available frequency spectrum is limited, these requirements increase the demand for more band-width efficient multiple access schemes. FDMA (frequency division multiple access), TDMA (time division multiple access) and CDMA (code division multiple access) [15, 52, 85] are the most widely known multiple access techniques. Especially, CDMA is considered a promising solution for wireless communication, since it offers frequency diversity and interference diversity to enhance spectral efficiency and capacity [84][100, Chap 1].

The explosive growth in wireless communications has triggered the need for more efficient mobile radio systems. In order to accommodate the demand for wireless communications services, new techniques that allow for efficient use for limited available frequency spectrum, increased system capacity, high data rates and better accuracy are being developed. Several of the fundamental problems that must be solved to achieve these goals are from the area of signal processing for communications. Signal processing related research in the recent past has made significant progress in improving the quality and accuracy of communications systems in the areas of channel estimation [22, 43, 47, 60, 77, 96], spectral estimation [6, 45, 71], direction of arrival estimation [50, 70, 99, 110, 111], etc. In this thesis, we focus on a very specific signal processing problem known as ‘subspace estimation’. As briefly explained in the next section, subspace estimation is a key tool used in several applications of wireless communications to provide reliable and high quality communication systems. In fact, the application of subspace estimation goes beyond wireless communications to several day-to-day applications of signal processing as an effective tool for parameter estimation and signal separation.

1.2 Motivation

Most of the signal processing problems in communications can be formulated as parameter estimation problems. In the interest of optimality, the maximum likelihood (ML) approach is usually used to formulate parameter estimation problems [74]. However, the resulting signal processing algorithms are often not practically feasible due to their heavy computational requirements. Therefore, algorithms providing trade-off between performance and complexity are of primary interest.

Subspace based approaches to solving estimation problems in communications lead to potentially low cost algorithms and near-optimal performance. For example, in channel estimation [46], training based approaches can be used to obtain the channel information at the receiver, the price to be paid being reduced bandwidth efficiency. Furthermore, training approaches may lead to inaccurate channel estimates due to the presence of noise and the limited duration and number of training symbols. A good substitute is to use subspace based approaches. Subspace based approaches are not only important to communications problems, but could also be used in other areas such as signal separation problems in medical signal processing [95]. Signal separation refers to recovering underlying source signals from a set of observations obtained by an unknown linear mixture of the sources.

Subspace-based methods are usually preferred, because they yield high resolution results.

Subspace based methods are based on the concept that the observation space of the received signal can be partitioned into two orthogonal subspaces, known as signal and noise subspaces. Correspondingly, the eigenvectors of the covariance matrix of the observed data can be partitioned into two sets to form the bases of these subspaces. Thus, estimation of bases of signal and/or noise subspaces becomes the first step in subspace-based estimation approaches [6, 22, 43, 45, 50, 60, 70, 71, 77, 96, 110, 111].

Performance of subspace-based algorithms depends, to a large extent, on the speed and accuracy of the subspace estimation process, especially when the parameters (and hence the subspaces) are time-varying. One possible choice for subspace-based methods is to use the standard eigenvalue decomposition (EVD) of the data covariance matrix to compute the signal or noise subspace. Unfortunately, the EVD is computationally intensive and time consuming, especially when the dimension of the observed data vectors is large. Consequently, in practical applications [51] where the signal is time-varying, repeated EVD of a continuously updated covariance matrix makes the subspace-based method difficult to implement in real-time. Therefore, the scope of our research is to develop stable, robust and low cost adaptive subspace estimation algorithms for applications in signal processing problems related to wireless communications.

1.3 Brief Review of Literature

We now present a very brief review of the existing literature on subspace estimation to show where our work fits in the big picture. A detailed review of literature will be given in Chapter 2. The literature referring to the problem of adaptive subspace tracking is enormous. There was a review paper by Common and Golub [31] published in 1990, focusing on the problem of tracking the signal subspace. This article did an excellent survey of the literature up to that time. The adaptive methods described in [31] are grouped into two classes, according to their complexity. The first class requires $O(N^2(N-P))$ [1] operations and the second needs $O(N(N-P)^2)$, where N is the dimension of the data vector and $(N-P)$ is the dimension of the signal subspace. The first adaptive approach for estimating the signal eigenvectors was developed by Owsley [82]. Yang and Kaveh [110] reported an adaptive approach for estimation of the entire signal or noise subspace. Based on an algorithm for estimating a single eigenvector, they also proposed inflation and deflation techniques for estimating noise subspace and signal subspace, respectively.

Although the above mentioned algorithms have lower complexities than EVD, their complexities are still not low enough. In the recent past, a large number of low complexity algorithms were proposed. For signal subspace estimation, one of the most famous algorithms is PAST (projection approximation subspace tracking) developed by Yang [109] with complexity $O(N(N-P))$. Other signal subspace estimation algorithms with similar complexity are Oja's algorithm [78], orthogonal Oja algorithm [4], NFQR (normalized fast Rayleigh's quotient algorithm) [10] and MALASE (maximum likelihood adaptive subspace estimation) [28]. For noise

[1] $O(\cdot)$ denotes order of the number of multiplications required by each algorithm.

subspace estimation, the available algorithms are quite limited as compared with signal subspace algorithms. The $O(NP)$ complexity ones are modified Oja's algorithm [105], Chen *et al.*'s algorithm [23] and fast Rayleigh's quotient based adaptive noise subspace algorithm (FRANS) [9]. Unfortunately, these noise subspace algorithms lose their orthogonality gradually and no longer extract the true subspace. More recently, several stable algorithms with computational complexity $O(NP)$ were proposed, such as HFRANS (FRANS with Householder transformation) [11] and FDPM (fast data projection method) [41]. However, they converge slowly since they are gradient based and non-optimal step-sizes are used to update all the subspace vectors at the same speed. They also require division and square-root operations which make them difficult for real-time implementation [44]. Moreover, the complex forms of the equations specifying these algorithms make it extremely difficult to analyze their performance. Therefore, in this thesis, we propose novel approaches that result in stable and fast subspace estimation algorithms to enhance the performance of bandwidth-efficient high-speed communications systems.

1.4 Contributions of the Thesis

As briefly mentioned at the end of section 1.2, the main overall objective of the research undertaken during this thesis work is to develop stable and fast subspace estimation algorithms that are low in complexity and near-optimal in performance. Our main contributions in this thesis are as follows.

• FRANS algorithm [9] is known to be unstable [11]. In Appendix A of [40], a theoretical analysis of orthogonality error in FRANS was given. But that analysis was not evaluated through simulation studies. In this thesis, we provide a different approach with both mean and mean-square analysis. We provide simulation results to corroborate the mean-square analysis of FRANS. We also examine the propagation of orthogonality error in HFRANS [11], which is a stable implementation of FRANS with Householder transform. We show that the orthogonality error growth of HFRANS is bounded linearly, which implies that Householder transform is an effective tool for stabilization of noise subspace estimation algorithms. Hence, we recommend HFRANS for noise subspace estimation.

- Even though HFRANS is a more stable low cost noise subspace estimation algorithm, its convergence is slow because it is a stochastic gradient-based adaptive algorithm. To achieve a good trade-off between convergence speed and steady-state error for HFRANS, we propose a gradient step-size strategy and an optimal step-size strategy. The proposed strategies can also be used for other subspace algorithms.

- In the literature, some of the well-known noise subspace algorithms estimate a number of subspace vectors in parallel. Unfortunately, the step-sizes used in these algorithms are all constant scalars. To improve the performance, use of adaptive step-size is proposed. However, a single adaptive step-size parameter is used to update all the subspace vectors. In this thesis, we propose that every subspace vector has its own step-size parameter, and hence each one should be allowed to converge with different dynamics. We implement our proposed strategy on MOja (modified Oja's algorithm) [105] and YK (Yang and Kaveh's algorithm) [110] algorithms since they are well-known and their forms are simple for further cost reduction and orthonormalization operations. The original MOja and YK algorithms are either unstable or computationally costly. We propose several low cost and stable implementations for MOja and YK with a diagonal matrix step-size. The resulting algorithms outperform the original algorithms with smaller estimation error and/or faster convergence rate.

- Most of the existing noise subspace algorithms involve several square-root and/or division operations. Consequently, it becomes very costly to implement these algorithms using VLSI (very large scale integration) circuits [44]. In this thesis, we propose a square-root and division free stable noise subspace estimation algorithm, known as SFRANS, which is a stabilized version of FRANS. By first simplifying FRANS through first order approximation and then adding a stabilizing factor, the proposed algorithm avoids the need for conventional orthonormalization methods [55]. An ODE (ordinary differential equation) based analysis is also provided to prove the stability of the proposed algorithm. We show that the algorithm is numerically stable if it is initialized properly.

1.5 Publications Originating from the Thesis

The contributions in this thesis have been published or accepted for publication as listed below.

Journals

[J1] Y. Lu, S. Attallah, G. Mathew and K. Abed-Meraim, "Analysis of Orthogonality Error Propagation for FRANS and HFRANS Algorithms," *IEEE Trans. Signal Proc.*, vol. 56, no. 9, pp. 4515-4521, Sep. 2008.

[J2] Y. Lu and S. Attallah, "Adaptive Noise Subspace Estimation Algorithm Suitable for VLSI Implementation," *IEEE Signal Proc. Lett.*, accepted.

Conferences

[C1] Y. Lu and S. Attallah, “Speeding up noise subspace estimation algorithms using an optimal diagonal matrix step-size strategy for MC-CDMA application,” *in Proc. IEEE VTC 2008 spring*, May 2008, pp. 1335-1339.

[C2] Y. Lu and S. Attallah, “Adaptive noise subspace estimation algorithm with an optimal diagonal-matrix step-size,” *in Proc. IEEE SIPS 2007*, Oct. 2007, pp. 584-588.

[C3] Y. Lu, S. Attallah and G. Mathew, “Stable noise subspace estimation algorithm suitable for VLSI implementation,” *in Proc. IEEE SIPS 2007*, Oct. 2007, pp. 579-583.

[C4] Y. Lu, S. Attallah, G. Mathew and K. Abed-Meraim, “Propagation of orthogonality error for FRANS algorithm,” *in Proc. ISSPA 2007*, Feb. 2007, pp. 1-4.

[C5] Y. Lu, S. Attallah and G. Mathew, “Variable step-size base adaptive noise subspace estimation for blind channel estimation,” *in Proc. APCC 2006*, Aug. 2006, pp. 1-5.

1.6 Organization of the Thesis

This thesis is devoted to the design and analysis of noise subspace estimation techniques for wireless communications. The rest of the thesis is organized as follows.

Chapter 2 outlines the mathematical preliminaries, the standard eigenvalue decomposition, iterative subspace computation techniques, and a review of the literature on both signal and noise subspaces estimation algorithms. This chapter also describes and finally the data generation method and the performance measures used in the simulation studies reported in the thesis.

In Chapter 3, the propagation of orthogonality error in FRANS and HFRANS is analyzed. First, we examine the propagation of orthogonality error in the numerically unstable FRANS in the mean and in the mean-square sense. We show that FRANS accumulates rounding errors and its orthogonality error grows geometrically. We then demonstrate that the orthogonality error propagation of the numerically well-behaved HFRANS is bounded by an upper bound that only slowly grows with iterations. The theoretical analysis is verified through computer simulations.

In Chapter 4, we describe a gradient step-size strategy and an optimal step-size strategy to improve the convergence performance of HFRANS algorithm. We assess the performance of the proposed strategies under stationary and non-stationary (tracking) conditions. An application to multi-carrier CDMA (MC-CDMA) system is also presented.

In Chapter 5, we propose a diagonal matrix step-size strategy for MOja and YK algorithms where the decoupled subspace vectors are controlled individually. Several low cost implementations are developed for each algorithm. The proposed step-size strategy is optimized through the optimal step-size technique of Chapter 4. Finally, effectiveness of the proposed implementations is verified through computer simulations.

In Chapter 6, we propose a VLSI friendly noise subspace estimation algorithm called SFRANS. It is derived from FRANS, but with much better stability. An optimal step-size based on the method discussed in Chapter 4 is proposed. The stability of SFRANS on the manifold and at the equilibrium point is examined through a corresponding ODE method.

The thesis is concluded in Chapter 7 with some suggestions for further research directions.

Chapter 2

Background Information and Literature Review of Subspace Estimation Algorithms

In this chapter, some background information and a detailed literature review of subspace estimation algorithms are provided. Development and analysis of subspace estimation algorithms require knowledge of linear algebra and matrix computations. So we start with a short review of the essential mathematical preliminaries. This is followed by a review of existing subspace estimation algorithms, particularly for estimating noise subspace, since techniques that estimate noise subspace are very limited as compared to the wide variety available for signal subspace estimation. Our main focus during this review is algorithms with low computational complexity since complexity is the second most important factor in the selection of an algorithm for practical implementation, the first factor being performance.

2.1 Mathematical Preliminaries

2.1.1 Vector Space

Let $\mathcal{V}$ be a set of $N \times 1$ vectors with complex-valued elements and let $\mathcal{C}$ denote the set of all complex-valued scalars. Then, $\mathcal{V}$ is said to be a vector space over $\mathcal{C}$ if its elements satisfy the following properties [64, Chap 7]:

- Commutativity: $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$.
- Associativity of vector addition: $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$.
- Existence of additive identity: For all $\mathbf{x} \in \mathcal{V}$, there exists a zero vector, $\mathbf{0} \in \mathcal{V}$, such that $\mathbf{x} + \mathbf{0} = \mathbf{0} + \mathbf{x} = \mathbf{x}$.
- Existence of additive inverse: For all $\mathbf{x} \in \mathcal{V}$, there exists an additive inverse vector, $(-\mathbf{x}) \in \mathcal{V}$, such that $\mathbf{x} + (-\mathbf{x}) = (-\mathbf{x}) + \mathbf{x} = \mathbf{0}$.
- Associativity of scalar multiplication: For all $\mathbf{x} \in \mathcal{V}$ and $\alpha, \beta \in \mathcal{C}$, $(\alpha\beta)\mathbf{x} = \alpha(\beta\mathbf{x})$.
- Distributive in scalar addition: For all $\mathbf{x} \in \mathcal{V}$ and $\alpha, \beta \in \mathcal{C}$, $(\alpha + \beta)\mathbf{x} = \alpha\mathbf{x} + \beta\mathbf{x}$.
- Distributive in vector addition: For all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$ and $\alpha \in \mathcal{C}$, $\alpha(\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$.
- Existence of scalar multiplication identity: $\alpha\mathbf{x} = \mathbf{x}\alpha = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{V}$ and $\alpha = 1 \in \mathcal{C}$.

- Closed under scalar multiplication: For all $\alpha \in \mathcal{C}$ and $\mathbf{x} \in \mathcal{V}$, $\alpha\mathbf{x} \in \mathcal{V}$.
- Closed under vector addition: For all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, $\mathbf{x} + \mathbf{y} \in \mathcal{V}$.

2.1.2 Subspace, Dimension and Rank

- Subspace: Given a collection of vectors $\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots,\ \mathbf{x}_M \in \mathcal{C}^N$, the set of all possible linear combinations of these vectors is referred to as the span of $\{\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots,\ \mathbf{x}_M\}$:

$$\text{span}\{\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots,\ \mathbf{x}_M\} = \left\{ \sum_{i=1}^{M} \alpha_i \mathbf{x}_i :\ \alpha_i \in \mathcal{C} \right\}. \tag{2.1}$$

  Clearly, span$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ is also a vector space and is called a subspace of the parent vector space from which vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$ are taken.

- Basis of a vector space: The set of minimum number of vectors needed to span a vector space is called a basis of that vector space. Clearly, the vectors that form a basis of a vector space are linearly independent.
- Dimension of a vector space: The number of vectors in a basis of a vector space is called the dimension of that vector space.
- Rank of a matrix: Let $\mathbf{C}$ be a $N \times N$ matrix of complex-valued elements. Then, the number of linearly independent columns or rows of $\mathbf{C}$ is called the rank of $\mathbf{C}$.

2.1.3 Gram-Schmidt Orthogonalization of Vectors

Given a set of linearly independent $N \times 1$ vectors $\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots,\ \mathbf{x}_M$, the Gram-Schmidt orthogonalization procedure can be used to generate a set of orthonormal $N \times 1$ vectors $\mathbf{y}_1,\ \mathbf{y}_2,\ \cdots,\ \mathbf{y}_M$ such that $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M$ span the same vector space as $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$. We start by choosing $\mathbf{y}_1$ in the direction of $\mathbf{x}_1$ as

$$\mathbf{y}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \tag{2.2}$$

where $\|\mathbf{x}_1\|$ denotes the Euclidean norm of $\mathbf{x}_1$. To determine $\mathbf{y}_2$, we first get $\mathbf{z}_2$ by subtracting the component of $\mathbf{x}_2$ along $\mathbf{y}_1$ as

$$\mathbf{z}_2 = \mathbf{x}_2 - \left(\mathbf{y}_1^H \mathbf{x}_2\right) \mathbf{y}_1 \tag{2.3}$$

where superscript 'H' denotes Hermitian transpose. Then, $\mathbf{y}_2$ is obtained by normalizing $\mathbf{z}_2$ as

$$\mathbf{y}_2 = \frac{\mathbf{z}_2}{\|\mathbf{z}_2\|}. \tag{2.4}$$

Similarly, to determine $\mathbf{y}_3$, we get $\mathbf{z}_3$ by subtracting from $\mathbf{x}_3$ its components along $\mathbf{y}_1$ and $\mathbf{y}_2$ as

$$\mathbf{z}_3 = \mathbf{x}_3 - \left(\mathbf{y}_2^H \mathbf{x}_3\right) \mathbf{y}_2 - \left(\mathbf{y}_1^H \mathbf{x}_3\right) \mathbf{y}_1 \tag{2.5}$$

and then normalize $\mathbf{z}_3$ to get $\mathbf{y}_3$

$$\mathbf{y}_3 = \frac{\mathbf{z}_3}{\|\mathbf{z}_3\|}. \tag{2.6}$$

This approach leads to the following Gram-Schmidt algorithm:

- Compute $\mathbf{y}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}$.

- For $2 \leq i \leq M$, compute

$$\mathbf{z}_i = \mathbf{x}_i - \sum_{k=1}^{i-1} \left(\mathbf{y}_k^H \mathbf{x}_i\right) \mathbf{y}_k \tag{2.7a}$$

$$\mathbf{y}_i = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}. \tag{2.7b}$$

2.2 Eigenvalue Decomposition

Eigenvalue problems form an important class of problems in scientific computing. Let the Hermitian matrix $\mathbf{C}$, i.e. $\mathbf{C}^H = \mathbf{C}$, be the $N \times N$ covariance matrix of a complex-valued wide-sense stationary discrete-time stochastic process represented by the $N \times 1$ observation vector $\mathbf{r}(i)$. Then, there exists a $N \times N$ unitary matrix

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N], \tag{2.8}$$

i.e., $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_N = \mathbf{Q}\mathbf{Q}^H$ with $\mathbf{I}_N$ being the $N \times N$ identity matrix, and a real-valued diagonal matrix

$$\boldsymbol{\Lambda} = \operatorname{diag}\left[\lambda_1, \lambda_2, \cdots, \lambda_N\right], \tag{2.9}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, satisfying

$$\mathbf{C}\mathbf{q}_i = \lambda_i \mathbf{q}_i, \quad i = 1, 2, \cdots, N. \tag{2.10}$$

That is, $\mathbf{q}_i$ is an eigenvector of $\mathbf{C}$ with corresponding eigenvalue λ_i for $i = 1, 2, \cdots, N$. Then, the covariance matrix $\mathbf{C}$ can be expressed as

$$\mathbf{C} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H = \sum_{i=1}^{N} \lambda_i \mathbf{q}_i \mathbf{q}_i^H, \tag{2.11}$$

which is known as the eigenvalue decomposition (EVD) of $\mathbf{C}$.

We divide the eigenvalues into two groups:

1. λ_i for $i = 1, 2, \cdots, P$, and

2. λ_i for $i = P + 1, \cdots, N$.

Correspondingly, the space spanned by the eigenvectors of $\mathbf{C}$ can be divided into two subspaces:

1. Noise (minor) subspace: spanned by the eigenvectors associated with the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_P$, and

2. Signal (principal) subspace: spanned by the eigenvectors associated with the eigenvalues $\lambda_{P+1}, \cdots, \lambda_N$.

Here, we have assumed that the signal subspace is of dimension $(N - P)$ and noise subspace is of dimension P, where $N > P$. Noise subspace is the orthogonal complement of signal subspace. Denoting $[\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_P]$ by $\mathbf{U}_n \in \mathcal{C}^{N \times P}$ and $[\mathbf{q}_{P+1}, \cdots, \mathbf{q}_N]$ by $\mathbf{U}_s \in \mathcal{C}^{N \times (N-P)}$, we can further represent the covariance matrix

$\mathbf{C}$ as

$$\mathbf{C} = \begin{bmatrix} \mathbf{U}_n & \mathbf{U}_s \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} \mathbf{U}_n^H \\ \mathbf{U}_s^H \end{bmatrix} = \mathbf{U}_n \mathbf{\Lambda}_n \mathbf{U}_n^H + \mathbf{U}_s \mathbf{\Lambda}_s \mathbf{U}_s^H, \tag{2.12}$$

where $\mathbf{\Lambda}_n = \text{diag}[\lambda_1, \cdots, \lambda_P]$ and $\mathbf{\Lambda}_s = \text{diag}[\lambda_{P+1}, \cdots, \lambda_N]$. Let $\mathbf{W} = \mathbf{U}_n\mathbf{B}$, where $\mathbf{B}$ is an arbitrary $P \times P$ unitary matrix. Since the columns of $\mathbf{U}_n$ span the noise subspace, the columns of $\mathbf{W}$ also span the noise subspace. For the sake of convenience, we call matrix $\mathbf{U}_n$ or its rotation $\mathbf{W}$ as noise subspace, even though columns of $\mathbf{W}$ are not eigenvectors of $\mathbf{C}$. Similar statements can be made about signal subspace also.

2.3 Iterative Subspace Computation Techniques

In this section, two of the most well known iterative subspace computation techniques, power method and its variant known as orthogonal iteration method, are presented.

2.3.1 Power Iteration Method

The power iteration method produces a sequence of scalars $\varphi(i)$ and vectors $\mathbf{w}(i)$ that converge to the largest eigenvalue λ_N and corresponding eigenvector $\mathbf{q}_N$, respectively, of a $N \times N$ symmetric non-negative definite matrix $\mathbf{A}$. The iteration is summarized in Table 2.1. Its convergence rate is exponential and proportional to the ratio of the two largest eigenvalues $\left(\frac{\lambda_{N-1}}{\lambda_N}\right)^i$. More discussion on power iteration method can be found in [49, Chap. 8] and [54].

If $\mathbf{A}$ is replaced by $\mathbf{A}^{-1}$ in Table 2.1, the resulting algorithm is known as inverse

```
Initialization: Choose $\mathbf{w}(0)$ to be a unit norm vector
For $i = 1, 2, \cdots$
    1. $\mathbf{p}(i) = \mathbf{A}\mathbf{w}(i-1)$
    2. $\mathbf{w}(i) = \frac{\mathbf{p}(i)}{\|\mathbf{p}(i)\|}$
    3. $\varphi(i) = \mathbf{w}(i)^H \mathbf{A}\mathbf{w}(i)$
```

Table 2.1: Power iteration method.

power iteration. In this case, $\varphi(i)$ and $\mathbf{w}(i)$ converge to the smallest eigenvalue and corresponding eigenvector, respectively, of $\mathbf{A}$.

The power iteration method has computational complexity $O(N^2)$ [54].

2.3.2 Orthogonal Iteration

The orthogonal iteration [90, 91] is a generalization of the power iteration method for simultaneous extraction of P eigenvectors and corresponding eigenvalues of $\mathbf{A}$. If $P = 1$, orthogonal iteration will reduce to power method. We have the orthogonal iteration presented in Table 2.2. Here, $\mathbf{W}(i)$ is a $N \times P$ matrix and $\mathbf{\Lambda}(i)$ is a $P \times P$ diagonal matrix. The orthonormalization of $\mathbf{AW}(i-1)$ can be realized by QR decomposition [49, Chap. 5], which is an efficient implementation of Gram-Schmidt orthogonalization. As iteration proceeds, $\mathbf{W}(i)$ converges to the

```
Initialization: Choose $\mathbf{W}(0)$ to be a $N \times P$ matrix with orthonormal columns
For $i = 1, 2, \cdots$
    1. $\mathbf{W}(i) = \text{orthonomalize}\left(\mathbf{AW}(i-1)\right)$
    2. $\mathbf{\Lambda}(i) = \text{diag}\left(\mathbf{W}^H(i)\mathbf{AW}(i)\right)$
```

Table 2.2: Orthogonal iteration.

eigenvectors corresponding to P largest eigenvalues of $\mathbf{A}$ and $\mathbf{\Lambda}(i)$ converges to the corresponding eigenvalues.

The orthogonal iteration method has computational complexity $O(N^2P)$.

2.4 Literature Review

In many signal processing systems, the main objective is to extract a few features and/or minimize the noise inherent in the complex, high dimensional input data. Two of the general purpose extraction techniques are principal component analysis (PCA) and minor component analysis (MCA) [25] [29, Chap 3][59, Chap 1]. PCA was introduced by Pearson [83] in 1901 as a methodology for fitting planes in the least-square sense and next developed by Hotelling [53] in works done on psychometry. PCA was also developed by Karhunen [63] in the context of probability theory and was subsequently generalized by Loève [67]. Principal components (PCs) and minor components (MCs) can be obtained by solving an eigenvalue problem of the correlation matrix of the received data. PCs are the directions in which the input data has the largest strengths or variances. In contrast to PCs, MCs are the directions in which the input data has the smallest strengths or variances. So, MCA is the problem of finding a set of smallest eigenvalues and corresponding eigenvectors, and PCA is the problem of finding a set of largest eigenvalues and corresponding eigenvectors. Stating in the language of signal processing, the purpose of PCA is to derive a relatively small number of decorrelated random variables while retaining as much information as possible from the original variables. PCA has been widely studied and used in pattern recognition and signal processing. It is important and extensively used in mobile positioning [20], blind channel equalization [27, 36, 76, 97, 98, 104], etc.

When we are interested only in the subspace spanned by the largest or smallest eigenvectors, we do not necessarily need to solve for the eigenvectors, but a set of vectors that span the same subspace as the eigenvectors would suffice. Such problems are called principal subspace analysis (PSA) and minor subspace analysis (MSA) or signal subspace estimation and noise subspace estimation. In the following review, both PCA/MCA and PSA/MSA are considered.

The standard numerical method for subspace extraction is to first compute the correlation matrix, and then the subspace. However, if the dimension of the input data vector is very large, the size of the correlation matrix will be very large, and hence it will be computationally costly to compute the subspace. Adaptive algorithms enable us to estimate the subspace without computing or estimating large correlation matrices. Such approaches are very useful in tracking scenarios where we need to estimate a signal feature that varies with time.

From the point of view of practical implementation, we classify the existing adaptive algorithms according to their computational complexity. In this thesis, we use flops counts as a key measure of complexity for an algorithm, although many other parameters such as parallelism and memory requirement are also important in practice. For signal subspace estimation, schemes requiring $O(N^2(N-P))$ or $O(N^2)$ operations per iteration are classified as high complexity algorithms, schemes requiring $O(N(N-P)^2)$ complexity are referred to as medium complexity algorithms, and schemes with complexity $O(N(N-P))$ are considered as low complexity algorithms. For noise subspace estimation, the same classification criteria apply by changing the dimension $(N-P)$ to P. Note that methods in the high complexity class usually have faster convergence rates than algorithms in the other two classes. It is understandable since there is a tradeoff between complexity and performance.

2.4.1 Estimation of Signal Subspace

2.4.1.1 High Complexity Class

For the sake of completeness, we give a quick review of the algorithms with complexity $O(N^2(N-P))$ or $O(N^2)$. Owsley first introduced an adaptive procedure for the estimation of signal subspace with $O(N^2(N-P))$ complexity [82]. Sharman [88] applied the QR recursive algorithm to estimate the complete eigenstructure with $O(N^2)$ complexity. Other algorithms with similar complexity can be found in review papers by Comon and Golub [31] and Reddy *et al.* [86].

2.4.1.2 Medium Complexity Class

Yang and Kaveh [110] proposed a stochastic gradient based constrained adaptive algorithm for estimation of signal subspace. The algorithm is based on maximizing the cost function $E\left[\mathrm{Tr}\left(\mathbf{W}^H\mathbf{C}\mathbf{W}\right)\right]$ subject to the orthonormality constraint on $\mathbf{W}$, where $E[\cdot]$ denotes stochastic expectation operator, $\mathrm{Tr}(\cdot)$ denotes trace operator and $\mathbf{W}$ is a $N \times (N-P)$ matrix. The proposed algorithm is given in Table 2.3, where $\mathbf{W}(i)$ denotes the signal subspace estimate and $\mathbf{r}(i)$ denotes the observed data vector $(N \times 1)$ at ith instant, and β denotes the step-size parameter that controls the convergence speed. It has $O(N(N-P)^2)$ complexity, which arises from the Gram-Schmidt orthonormalization in Step 3.

In [62], Karasalo proposed a QR-based subspace algorithm, where the noise eigenvalues are replaced by their average value so that deflation could be used to

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{W}}(i) = \mathbf{W}(i-1) + \beta\mathbf{r}(i)\mathbf{y}^H(i)$
3. $\mathbf{W}(i) =$ Gram-Schmidt orthonormalization$(\bar{\mathbf{W}}(i))$
```

Table 2.3: Yang and Kaveh's algorithm [110] for signal subspace estimation.

reduce complexity. According to [31], it had the best performance to complexity ratio. The corresponding algorithm is presented in Table 2.4, with overall complexity $O(N(N-P)^2)$. Replacing the small dimensional singular value decomposition (SVD) used in [62] (complexity $O((N-P)^3)$) by a transposed QR iteration (complexity $O((N-P)^2)$), a TQR-SVD algorithm was proposed by Dowling *et al.* [42]. However, this resulted in a serious performance loss [91].

Strobach [89] draws two conclusions from the TQR-SVD [42] algorithm: (1) Karasalo's algorithm [62] is very sensitive to the accuracy of $(N-P+2)\times(N-P+1)$ SVD; and (2) more than a single iteration for this SVD in each step results in a computationally less attractive algorithm. To overcome this dilemma, the Bi-SVD algorithm proposed by Strobach [89] completely avoids the need for SVD. It is based on the bi-iteration method [30], and it involves only a single small $(N-P+1)\times(N-P)$ QR decomposition. This decomposition is much faster to compute than any approximation of the $(N-P+2)\times(N-P+1)$ SVD in Karasalo's algorithm. It claimed to outperform its related predecessors.

2.4.1.3 Low Complexity Class

Low complexity algorithms are the most important ones due to their suitability for real-time applications. In the early 1990's, the merger of signal processing and neural networks brought a lot of attention to Oja's subspace algorithm [78],

```
Initialization: Choose $\mathbf{W}(0) = \begin{bmatrix} \mathbf{I}_{N-P} \\ \mathbf{0}_{P\times(N-P)} \end{bmatrix}$, $\mathbf{S}_{N-P}(0) = \mathbf{I}_{N-P}$, $p(0) = 1$,
                and $0 \leq \lambda \leq 1$
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{r}_\perp(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
3. $z(i) = \mathbf{r}_\perp^H(i)\mathbf{r}_\perp(i)$
4. $\bar{\mathbf{r}}_\perp(i) = z^{-\frac{1}{2}}(i)\mathbf{r}_\perp(i)$
5. $\mathbf{K}(i) = \begin{bmatrix} \lambda^{\frac{1}{2}}\mathbf{S}_{N-P}(i-1) & \mathbf{0}_{(N-P)\times 1} \\ \mathbf{0}_{1\times(N-P)} & \lambda^{\frac{1}{2}}p(i-1) \\ (1-\lambda)^{\frac{1}{2}}\mathbf{y}(i) & (1-\lambda)^{\frac{1}{2}}z^{\frac{1}{2}}(i) \end{bmatrix}$
6. $\mathbf{K}(i) = \mathbf{U}(i)\mathbf{S}_{N-P+1}(i)\mathbf{V}^H(i)$: $(N-P+2)\times(N-P+1)$ SVD
7. $\mathbf{S}_{N-P+1}(i) = \begin{bmatrix} \mathbf{S}_{N-P}(i) & \mathbf{0}_{(N-P)\times 1} \\ \mathbf{0}_{1\times(N-P)} & s_{N-P+1}(i) \end{bmatrix}$
8. $\mathbf{V}(i) = \begin{bmatrix} \mathbf{\Theta}(i) & \vdots \\ \mathbf{f}^H(i) & * \end{bmatrix}$
9. $\mathbf{W}(i) = \mathbf{W}(i-1)\mathbf{\Theta} + \bar{\mathbf{r}}_\perp(i)\mathbf{f}^H(i)$
10. $p^2(i) = \frac{1}{P}\left(s_{N-P+1}(i) + \lambda(P-1)p^2(i-1)\right)$
```

Table 2.4: Karasalo's algorithm [62] for signal subspace estimation.

depicted in Table 2.5. This algorithm can be interpreted as a constrained or approximate gradient rule for maximizing the output variance or minimizing the mean square error (MSE). It has complexity $O(NP)$. Analyses in [24, 106] showed that it is globally convergent with an arbitrary small step-size. However, its accuracy is sensitive to the chosen step-size. If the step-size is not small enough, Oja's algorithm diverges.

The projection approximation subspace tracking algorithm (PAST) proposed by Yang [109] is based on a novel interpretation of the signal subspace as the solution of an unconstrained minimization task. A projection approximation is utilized

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{W}(i) = \mathbf{W}(i-1) + \beta\left[\mathbf{r}(i)\mathbf{y}^H(i) - \mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\right]$

Table 2.5: Oja's algorithm [78] for signal subspace estimation.

to reduce the minimization task to the well known exponentially weighted least square problem. Recursive least squares (RLS) methods are then used to track the signal subspace. The steps of PAST are presented in Table 2.6, where the notation Tri($\mathbf{X}$) denotes that only the upper (or lower) triangular part of $\mathbf{X}$ is calculated and its Hermitian transposed version is copied to another lower (or upper) triangular part. PAST has fast convergence, because it is a recursive least squares (RLS) type of implementation. However, it does not guarantee orthonormality of the estimated subspace matrix, which might be needed in some applications [43]. A variant of PAST, called PASTd (PAST with deflation), was also presented in [109], where the principal components are extracted sequentially. Using a deflation technique, the most dominant principal component is first extracted and is subtracted from the data vector, so that the second dominant principal component becomes the most dominant component in the modified data vector. This procedure is repeated until all the desired eigencomponents are estimated. An extension of PASTd to estimate the dimension of signal subspace and signal subspace was presented in [107]. And a theoretical convergence analysis for both PAST and PASTd was given in [108]. To ensure global convergence and to guarantee the orthonormality of the noise subspace matrix at each iteration, in [2], an explicit orthonormalization at each iteration of PAST was carried out, which resulted in the orthonormal PAST (OPAST) algorithm.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{h}(i) = \mathbf{P}(i-1)\mathbf{y}(i)$
3. $\mathbf{g}(i) = \mathbf{h}(i)/[\beta + \mathbf{y}^H(i)\mathbf{h}(i)]$
4. $\mathbf{P}(i) = \frac{1}{\beta}\mathrm{Tri}\{\mathbf{P}(i-1) - \mathbf{g}(i)\mathbf{h}^H(i)\}$
5. $\mathbf{e}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
6. $\mathbf{W}(i) = \mathbf{W}(i-1) + \mathbf{e}(i)\mathbf{g}^H(i)$
```

Table 2.6: PAST [109] for signal subspace estimation.

The NIC (novel information criterion) algorithm introduced in [75] is a generalization of the PAST algorithm. The NIC exhibits a single global maximum if and only if the estimated signal subspace spans the desired signal subspace. The performance of NIC is similar to that of PAST [54, 75]. A convergence analysis of NIC can be found in [66] and [75]. As shown in [54], the algorithms of Oja, PAST and NIC [75, 78, 109] are all variations of the power iteration method, whose key feature is that the key information for subspace update comes from multiplying the old subspace matrix by the new observed data vector.

In [54], a natural version of the power iteration method and its implementation with varied computational complexity are presented. Its linear complexity implementation, NP3, has faster convergence than other power method based algorithms, such as Oja [78], PAST [109] and NIC [75], and is globally convergent. More information on power iteration based methods can be found in [54] and [55].

In [13], Badeau *et al.* proposed a fast implementation of the power iteration method for subspace tracking, based on an approximation that is less restrictive than the well-known projection approximation. This algorithm, referred to as fast approximated power iteration (API), is faster than a previously proposed API method [12]. It guarantees orthonormality of the subspace estimate at each iteration. Moreover, it outperforms the other power iteration based methods, such as PAST [109], NIC [75], NP3 [54], and OPAST [2], while having the same computational complexity.

Non-power based methods include the maximum likelihood adaptive subspace estimation (MALASE) algorithm [28] and several quasi-Newton based algorithms proposed by Mathew *et al.* [72], Kang *et al.* [61] and Ouyang *et al.* [81]. MALASE [28] computes the singular vectors of the subspace of interest and the resulting estimates form an orthonormal matrix. However, it requires proper initialization or otherwise it does not converge. In [72], Mathew *et al.* presented a rapidly convergent quasi-Newton based algorithm derived from a cost function based on the penalty function method of optimization. This algorithm estimates the eigenvectors corresponding to the smallest eigenvalues up to the eigenvectors corresponding to the largest eigenvalues sequentially. However, in many applications we are only interested in getting the principal eigenvectors. Kang *et al.* [61] proposed another adaptive quasi-Newton algorithm by reversing the estimation order. They have demonstrated an improved convergence rate compared with the RLS type algorithms reported in[109] and [16] and the quasi-Newton algorithm reported in [72], by using a geometric progression approximation to the Hessian matrix. However, both [61] and [72] have computational complexity $O(N^2(N-P))$, and ad-hoc selection of the penalty factor may jeopardize performance under time-varying scenario. To alleviate these problems, a non-quadratic information criterion based quasi-Newton algorithm was proposed by Ouyang *et al.* in [81].

2.4.2 Estimation of Noise Subspace

In the literature, noise subspace estimation algorithms are very limited as compared with signal subspace estimation algorithms and many of the existing algorithms with low complexity are either unstable or lose orthogonality rapidly [69, 93]. One would expect that noise subspace algorithms can be derived from the same idea as the corresponding signal subspace algorithms, by simply changing the sign of the update term. However, this makes most of the noise subspace algorithms numerically unstable. For example, Xu *et al.* [105] proposed a straightforward modification of Oja's single unit rule [78] to obtain the minor component of the input sequence. It was shown in [26] and [35, Chap 4] that this simple change does not extract the minor component. More discussion on the convergence of PCA (or PSA) algorithms and the corresponding MCA (or MSA) algorithms are available in [26]. Exceptions where noise subspace algorithms derived from the same idea as the corresponding signal subspace algorithms are stable can be seen in [72] and [110], etc.

2.4.2.1 High Complexity Class

In [79], Ouyang *et al.* proposed an algorithm for adaptive extraction of minor components (AMEX) based on gradient search over a nonquadratic criterion function. Unlike other approaches, the AMEX algorithm is capable of obtaining the desired minor eigenvalues and the corresponding eigenvectors directly. The quasi-Newton algorithm proposed in [72] can also be used for noise subspace estimation.

2.4.2.2 Medium Complexity Class

Yang and Kaveh's algorithm (YK) [110] minimizes the cost function $E\left[\text{Tr}\left(\mathbf{W}^H\mathbf{C}\mathbf{W}\right)\right]$, subject to the orthonormality constraint $\left(\mathbf{W}^H\mathbf{W} = \mathbf{I}\right)$. This can be done by reversing the sign of the gradient term in Step 2 of Table 2.3. The resulting algorithm is shown in Table 2.7. Observe that it also has $O(NP^2)$ complexity.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{W}}(i) = \mathbf{W}(i-1) - \beta\mathbf{r}(i)\mathbf{y}^H(i)$
3. $\mathbf{W}(i) = \text{Gram-Schmidt orthonormalize}\left\{\bar{\mathbf{W}}(i)\right\}$
```

Table 2.7: Yang and Kaveh's algorithm [110] for noise subspace estimation.

2.4.2.3 Low Complexity Class

Most of the noise subspace algorithms of computational complexity $O(NP)$ are either unstable or non-robust. The modified Oja's single unit rule (MOja) proposed by Xu *et al.* [105], about which we mentioned at the beginning of Section 2.4.2, is shown in Table 2.8. However, Diamantaras and Kung [35] showed that this algorithm diverges quickly.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\left[\mathbf{r}(i)\mathbf{y}^H(i) - \mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\right]$
```

Table 2.8: Modified Oja's algorithm [105] for noise subspace estimation.

Table 2.9 shows another noise subspace estimation algorithm, proposed by Chen *et al.* [23]. In terms of orthonormality, it loses orthonormality slower than the modified Oja's algorithm of Xu *et al.* [105]. However, it eventually diverges.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\left[\mathbf{r}(i)\mathbf{y}^H(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1) - \mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\right]$
```

Table 2.9: Chen et al.*'s algorithm [23] for noise subspace estimation.*

Douglas *et al.* [37] proposed a self-stabilized minor subspace rule (SMSR) for estimation of noise subspace. The algorithm is shown in Table 2.10. Its self-stabilizing property makes it more stable than the above mentioned low complexity algorithms. Further, it contains no square-root or division operations, which makes it suitable for VLSI implementation. An ODE analysis is also given in [37] to prove its convergence. However, Abed-Meraim *et al.* [5] showed that this analysis is flawed when the data is time-variant. Based on a general method that transforms MSA algorithms into MCA algorithms [58], a modification of [37] was proposed by Jankovic and Reljin [57] for extraction of minor components.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\left[\mathbf{r}(i)\mathbf{y}^H(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\right.$
       $\left. - \mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\right]$
```

Table 2.10: Self-stabilized minor subspace rule [37] for noise subspace estimation.

More recently, four stable low cost noise subspace estimation algorithms, namely NOOja (normalized orthogonal Oja) [8], FOOja (fast orthogonal Oja) [21], HFRANS (FRANS with Householder transformation) [11] and FDPM (fast data projection method) [41] were proposed. If we categorize them by the algorithms from which they are derived, NOOja [8] and FOOja [21] are based on MOja [105], while HFRANS [11] and FDPM [41] are based on YK [110]. If we categorize them by the orthonormalization methods they use, NOOja [8] and HFRANS [11] directly orthonormalize the subspace estimation matrix, while FOOja [21] and FDPM [41] first orthogonalize and then normalize the subspace estimation matrix.

To introduce HFRANS algorithm [11], we have to start with its unstable implementation, known as FRANS algorithm [9]. Recall that Yang and Kaveh's algorithm [110] in Table 5.5 has $O(NP^2)$ complexity due to the Gram-Schmidt orthonormalization step. To reduce computational complexity, Attallah [9] proposed the FRANS algorithm. It is a low cost orthonormalized implementation of Yang and Kaveh's algorithm [110]. As shown in [110], Yang and Kaveh iteratively minimize the cost function

$$J_{\mathbf{W}} = E\left[\mathrm{Tr}(\mathbf{W}^H\mathbf{C}\mathbf{W})\right], \tag{2.13}$$

subject to the constraint

$$\mathbf{W}^H\mathbf{W} = \mathbf{I}_P, \tag{2.14}$$

where $\mathbf{W} \in \mathcal{C}^{N\times P}$, $\mathbf{I}_P$ is the $P \times P$ identity matrix and $\mathrm{Tr}(\cdot)$ denotes the trace operator. Using an instantaneous gradient-descent approach combined with Gram-Schmidt orthonormalization, the resulting adaptive noise subspace estimation algorithm is given by [110]

$$\begin{aligned}
\mathbf{y}(i) &= \mathbf{W}^H(i-1)\mathbf{r}(i)\\
\bar{\mathbf{W}}(i) &= \mathbf{W}(i-1) - 2\beta\mathbf{r}(i)\mathbf{y}^H(i)\\
\mathbf{W}(i) &= \text{Gram-Schmidt orthonormalization}(\bar{\mathbf{W}}(i)),
\end{aligned} \tag{2.15}$$

where β is a small positive step-size. To reduce complexity, Attallah [9] proposed to do orthonormalization using

$$\mathbf{W}(i) = \bar{\mathbf{W}}(i) \left(\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)\right)^{-\frac{1}{2}}, \tag{2.16}$$

instead of Gram-Schmidt orthonormalization. After some manipulations, the inverse square root of $\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)$ can be written as [9]

$$\left(\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)\right)^{-\frac{1}{2}} = \mathbf{I}_P + \tau(i)\mathbf{y}(i)\mathbf{y}^H(i), \tag{2.17}$$

where

$$\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}\left[\frac{1}{\sqrt{1-4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2}} - 1\right], \tag{2.18}$$

based on the assumption that $\mathbf{W}(i)$ is orthonormal, *i.e.* $\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{I}_P$, and the equality that

$$\left(\mathbf{I}_P - \mathbf{x}\mathbf{x}^H\right)^{-\frac{1}{2}} = \mathbf{I}_P + \left(\frac{1}{\sqrt{1-\|\mathbf{x}\|^2}} - 1\right)\frac{\mathbf{x}\mathbf{x}^H}{\|\mathbf{x}\|^2} \tag{2.19}$$

for any $N \times 1$ vector $\mathbf{x}$. Substituting (2.17) and (2.18) in (2.16), we get the FRANS algorithm [9]. Moreover, a normalized step-size was proposed in [7] as the instantaneous suboptimal of $\frac{\partial J_{\mathbf{W}(i)}}{\partial \beta(i)} = 0$, where $J_{\mathbf{W}}(i) = E\left[\mathrm{Tr}(\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i))\right]$, and $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\beta(i)\mathbf{r}(i)\mathbf{y}^H(i)$. The resulting step-size is $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$. The complete steps of FRANS with adaptive step-size is given in Table 2.11.

In a later paper [11], using simulations, Attallah showed that FRANS cannot preserve orthogonality. As mentioned in [9], a periodic re-orthonormalization of

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$
3. $\delta(i) = 4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2$
4. $\rho(i) = \sqrt{1-\delta(i)}$
5. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}(\frac{1}{\rho(i)} - 1)$
6. $\mathbf{p}(i) = -\tau(i)\mathbf{W}(i-1)\mathbf{y}(i)/\beta + 2\mathbf{r}(i)(1+\tau(i)\|\mathbf{y}(i)\|^2)$
7. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i)$
```

Table 2.11: FRANS algorithm [9] for noise subspace estimation.

the noise subspace estimate is necessary from time to time in order to keep the algorithm numerically stable. The Householder transformation was used successfully in [4] to stabilize Oja's algorithm [78] for noise subspace estimation. In [11], Attallah gave a mathematical proof for the existence of a Householder transformation-based implementation for FRANS algorithm. The proposed algorithm is named as HFRANS algorithm (FRANS with Householder transformation), where

$$\mathbf{W}(i) = \mathbf{H}(i)\mathbf{W}(i-1), \tag{2.20}$$

where $\mathbf{H}(i)$ is the Householder matrix given by

$$\mathbf{H}(i) = \mathbf{I}_N - 2\mathbf{u}(i)\mathbf{u}^H(i) \tag{2.21}$$

$$\mathbf{u}(i) = \frac{\mathbf{p}(i)}{\|\mathbf{p}(i)\|}. \tag{2.22}$$

Proposition 1 in [11] proved that Step 6 in Table 2.11 can be rewritten as (2.20). We summarize the HFRANS algorithm in Table 2.12, where a normalized step-size $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$ was used as the instantaneous suboptimal of $\frac{\partial J_{\mathbf{W}}(i)}{\partial \beta(i)} = 0$, with

$J_{\mathbf{W}}(i) = E\left[\mathrm{Tr}(\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i))\right]$, and $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\beta(i)\mathbf{r}(i)\mathbf{y}^H(i)$.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$
3. $\delta(i) = 4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2$
4. $\rho(i) = \sqrt{1-\delta(i)}$
5. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}(\frac{1}{\rho(i)} - 1)$
6. $\mathbf{p}(i) = -\tau(i)\mathbf{W}(i-1)\mathbf{y}(i)/\beta + 2\mathbf{r}(i)(1+\tau(i)\|\mathbf{y}(i)\|^2)$
7. $\mathbf{u}(i) = \mathbf{p}(i)/\|\mathbf{p}(i)\|$
8. $\mathbf{v}(i) = \mathbf{W}^H(i-1)\mathbf{u}(i)$
9. $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\mathbf{u}(i)\mathbf{v}^H(i)$
```

Table 2.12: HFRANS algorithm [11] for noise subspace estimation.

Using a similar approach, HFRANS's MOja based counterpart NOOja [8] can be obtained. NOOja is orthogonal MOja algorithm (OOja) [4] with a normalized step-size. By adding an orthonormalization step to MOja, OOja improves the stability of MOja. The orthonormalization is done as follows [8]

$$\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i) \tag{2.23}$$

$$\mathbf{z}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i) \tag{2.24}$$

$$\bar{\mathbf{W}}(i) = \mathbf{W}(i-1) - \beta\mathbf{z}(i)\mathbf{y}^H(i) \tag{2.25}$$

$$\mathbf{W}(i) = \bar{\mathbf{W}}(i)\left(\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)\right)^{-\frac{1}{2}}, \tag{2.26}$$

where

$$\left(\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)\right)^{-\frac{1}{2}} = \mathbf{I}_P + \tau(i)\mathbf{y}(i)\mathbf{y}^H(i) \tag{2.27}$$

$$\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}\left(\frac{1}{\sqrt{1+\beta^2\|\mathbf{z}(i)\|^2\|\mathbf{y}(i)\|^2}} - 1\right). \tag{2.28}$$

Note that equations (2.23)-(2.25) are the steps of MOja algorithm [105] (Table 2.8), and equation (2.26) is the same as the orthonormlization step (2.16) of FRANS [9]. However, (2.26) also suffers from the propagation of roundoff errors as FRANS. To alleviate the problem of roundoff error propagation, a Householder transformation implementation similar to the one in HFRANS was proposed as [8]

$$\mathbf{W}(i) = \mathbf{H}(i)\mathbf{W}(i-1), \tag{2.29}$$

where $\mathbf{H}(i)$ is the Householder matrix given by

$$\mathbf{H}(i) = \mathbf{I}_N - 2\mathbf{u}(i)\mathbf{u}^H(i) \tag{2.30}$$

$$\mathbf{u}(i) = \frac{\mathbf{p}(i)}{\|\mathbf{p}(i)\|}. \tag{2.31}$$

The Proposition in [4] proved that (2.26) can be rewritten as (2.29). The steps of OOja is summarized in Table 2.13.

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{z}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
3. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}\left(\frac{1}{\sqrt{1+\beta^2\|\mathbf{z}(i)\|^2\|\mathbf{y}(i)\|^2}} - 1\right)$
4. $\mathbf{p}(i) = -\frac{\tau(i)}{\beta}\mathbf{W}(i-1)\mathbf{y}(i) + (1+\tau(i)\|\mathbf{y}(i)\|^2)\mathbf{z}(i)$
5. $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\frac{\mathbf{p}(i)\mathbf{p}^H(i)\mathbf{W}(i-1)}{\|\mathbf{p}(i)\|^2}$

Table 2.13: OOja algorithm [4] for noise subspace estimation.

If the step-size β in OOja is too large, the algorithm will not track the true noise subspace. To avoid the uncertainty in choosing step-size, an optimal step-size was proposed in NOOja by maximizing the mean-square error (MSE) $J_{\mathbf{w}} = E\|\mathbf{r}(i) - \mathbf{W}\mathbf{W}^H\mathbf{r}(i)\|^2$. The NOOja algorithm is summarized in Table 2.14.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{z}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
3. $\beta(i) = \frac{\alpha}{\|\mathbf{r}(i)\|^2 - \|\mathbf{y}(i)\|^2 + \varepsilon}$
4. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}\left(\frac{1}{\sqrt{1+\beta^2(i)\|\mathbf{z}(i)\|^2\|\mathbf{y}(i)\|^2}} - 1\right)$
5. $\mathbf{p}(i) = -\frac{\tau(i)}{\beta(i)}\mathbf{W}(i-1)\mathbf{y}(i) + (1+\tau(i)\|\mathbf{y}(i)\|^2)\mathbf{z}(i)$
6. $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\frac{\mathbf{p}(i)\mathbf{p}^H(i)\mathbf{W}(i-1)}{\|\mathbf{p}(i)\|^2}$
where $\alpha$ and $\varepsilon$ are small positive constants.
```

Table 2.14: NOOja algorithm [8] for noise subspace estimation.

The FDPM approach proposed by Doukopoulos and Moustakides [41] is an alternative low cost implementation of Yang and Kaveh's algorithm [110]. Unlike HFRANS where orthonormalization is done directly by setting $\mathbf{W}(i) = \bar{\mathbf{W}}(i) \cdot \left(\bar{\mathbf{W}}^H(i)\bar{\mathbf{W}}(i)\right)^{-\frac{1}{2}}$, FDPM first orthogonalizes $\mathbf{W}(i)$ by left multiplication with a Householder matrix, which results in $\mathbf{Z}(i)$. Then $\mathbf{Z}(i)$ is normalized to produce orthonormal $\mathbf{W}(i)$. The details of FDPM are shown in Table 2.15. Comparing HFRANS and FDPM, FDPM requires $O(P)$ divisions and $O(P)$ square-root operations at each iteration, whereas HFRANS requires only 4 divisions and 1 square-root operation at each iteration. Further, this number does not grow with the dimension of the noise subspace. In very large scale integration (VLSI) technology, division and square-root operations are very costly to implement [44]. Therefore, HFRANS with significantly less division and square-root operations could be favored in practical applications.

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{T}_{\text{YK}}(i) = \mathbf{W}(i-1) - \frac{\beta}{\|\mathbf{r}(i)\|^2}\mathbf{r}(i)\mathbf{y}^H(i)$
3. $\mathbf{a}(i) = \mathbf{y}(i) - \|\mathbf{y}(i)\|\mathbf{e}_1(i)e^{jangle(\mathbf{e}_1^H\mathbf{y}(i))}$
4. $\mathbf{Z}(i) = \mathbf{T}_{\text{YK}}(i) - 2\frac{[\mathbf{T}_{\text{YK}}(i)\mathbf{a}(i)]\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}$
5. $\mathbf{D}(i) = \left[\text{diag}\left(\mathbf{Z}^H(i)\mathbf{Z}(i)\right)\right]^{-\frac{1}{2}}$
6. $\mathbf{W}(i) = \mathbf{Z}(i)\mathbf{D}(i)$

Table 2.15: FDPM algorithm [41] for noise subspace estimation.

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{z}(i) = \mathbf{W}(i-1)\mathbf{y}(i)$
3. $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{z}(i)$
4. $\mathbf{T}_{\text{Oja}}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i)$
5. $\mathbf{a}(i) = \mathbf{y}(i) - \|\mathbf{y}(i)\|\mathbf{e}_1e^{jangle(\mathbf{e}_1^H\mathbf{y}(i))}$
6. $\mathbf{Z}(i) = \mathbf{T}_{\text{Oja}}(i) - 2\frac{[\mathbf{T}_{\text{Oja}}(i)\mathbf{a}(i)]\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}$
7. $\mathbf{D}(i) = \left[\text{diag}\left(\mathbf{Z}^H(i)\mathbf{Z}(i)\right)\right]^{-\frac{1}{2}}$
8. $\mathbf{W}(i) = \mathbf{Z}(i)\mathbf{D}(i)$

Table 2.16: FOOja algorithm [21] for noise subspace estimation.

Bartelmaos and Abed-Meraim [21] applied the principle of FDPM [41] to MOja [105] and developed the FOOja algorithm [21] shown in Table 2.16. Comparing HFRANS [11], NOOja [8], FDPM [41] and FOOja [21], these four algorithms have similar performance in terms of estimation error. In terms of computational complexity, FDPM and FOOja require $O(P)$ divisions and $O(P)$ square-root operations at each iteration, whereas HFRANS and NOOja require only 4 or 5 divisions and 1 square-root operation at each iteration. In terms of orthogonality, the orthogonality errors in FDPM and FOOja do not grow as iteration grows. However, the orthogonality errors in HFRANS and NOOja grow slowly with iteration.

In [87], Regalia proposed a gradient-based algorithm based on minimization of Rayleigh quotient where the constraint is replaced by a Givens parametrization [49, Chap 5.]. Since there are no division or square-root operations in the update formula, it is suitable for VLSI implementation. Its convergence is proved in [34]. An extension of this algorithm to the complex-valued case is given by Delmas in [33].

In [14], YAST algorithm for noise subspace estimation was proposed by Badeau *et al.*. It reduces the computational cost by limiting the search to the range space of $\mathbf{W}(i)$ plus one or two additional search directions, which means the P dimensional range space of $\mathbf{W}(i)$ is to be found as a subspace of the $P+p$ dimensional space spanned by the $N \times (P+p)$ matrix $[\mathbf{W}(i-1), \mathbf{x}(i)]$, where $\mathbf{x}(i)$ contains $p = 1$ or 2 columns. Although it does not use the Householder matrix to restore orthonormality, it is numerically very stable. As claimed in [14], it outperforms NOOja algorithm [8]. It has $6NP$ complexity when $p = 1$, and $11NP$ complexity when $p = 2$, which is slightly higher than the $4NP$ complexity of NOOja [8].

2.5 Data Generation and Performance Measures

2.5.1 Data Generation

We now explain the data generation method in this section, since it will be used throughout the simulations presented in this thesis. Let $\mathbf{C}$ be the covariance matrix of the observed data sequence $\{\mathbf{r}(i)\}$. The EVD of $\mathbf{C}$ is given as

$$\mathbf{C} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H = \left(\mathbf{Q}\mathbf{\Lambda}^{\frac{1}{2}}\right)\left(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{Q}\right) = \mathbf{K}\mathbf{K}^H, \tag{2.32}$$

where $\mathbf{K} = \mathbf{Q}\mathbf{\Lambda}^{\frac{1}{2}}$. A sequence of independent jointly-Gaussian random vectors $\{\mathbf{n}(i)\} \in \mathcal{C}^{N\times 1}$ is generated, with

$$E\left[\mathbf{n}(i)\mathbf{n}^H(i)\right] = \mathbf{I}_N \tag{2.33}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. Then, the observed data is generated as

$$\mathbf{r}(i) = \mathbf{K}\mathbf{n}(i) \tag{2.34}$$

so that $E\left[\mathbf{r}(i)\mathbf{r}^H(i)\right] = \mathbf{C}$ is satisfied.

2.5.2 Performance Measures

In this thesis, as in [11], we use mainly two performance measures for assessing subspace estimation algorithms. These measures are given by

$$\sigma(i) = \frac{1}{r_0}\sum_{r=1}^{r_0} \frac{\text{Tr}(\mathbf{W}_r^H(i)\mathbf{U}_s\mathbf{U}_s^H\mathbf{W}_r(i))}{\text{Tr}(\mathbf{W}_r^H(i)\mathbf{U}_n\mathbf{U}_n^H\mathbf{W}_r(i))} \tag{2.35}$$

$$\eta(i) = \frac{1}{r_0}\sum_{r=1}^{r_0} \|\mathbf{W}_r^H(i)\mathbf{W}_r(i) - \mathbf{I}_P\|_F^2 \tag{2.36}$$

where r_0 denotes the number of Monte Carlo runs, $\mathbf{W}_r(i)$ denotes the estimated subspace matrix $\mathbf{W}(i)$ in the rth run, $\|.\|_F$ denotes the Frobenius norm, and $\mathbf{U}_s$ and $\mathbf{U}_n$ are the signal and noise subspaces, respectively, of the underlying covariance matrix $\mathbf{C}$. Here, $\sigma(i)$ measures the estimation error in $\mathbf{W}_r(i)$ with respect to the true noise subspace $\mathbf{U}_n$, and $\eta(i)$ measures the orthogonality error of the estimated noise subspace $\mathbf{W}_r(i)$. Another useful performance measure, that is used occasionally, is the cost function given by

$$J_{\mathbf{W}}(i) = \frac{1}{r_0} \sum_{r=1}^{r_0} \operatorname{Tr}\left(\mathbf{W}_r^H(i)\mathbf{C}\mathbf{W}_r(i)\right), \tag{2.37}$$

where $J_{\mathbf{W}}(i)$ measures the estimated sum of the P smallest eigenvalues of $\mathbf{C}$. In all our simulations, $\mathbf{W}(0)$ is initialized to be the first P columns of $\mathbf{I}_N$, and r_0 is set to 500.

Chapter 3

Analysis of Propagation of Orthogonality Error for FRANS and HFRANS Algorithms

Recall from Chapter 2 that the FRANS [9] and HFRANS [11] algorithms are computationally efficient implementations of Yang and Kaveh's adaptive algorithm [110] for estimation of noise subspace. It has been observed that FRANS becomes unstable due to accumulation of rounding errors. However, what is still not understood is how the rounding errors are accumulated in FRANS and which are the parameters that influence the error accumulation process. In this chapter, we first examine the propagation of orthogonality error for FRANS in the mean and in the mean-square sense. We also derive an upper bound for the orthogonality error of the HFRANS algorithm. Numerical examples and results are provided to corroborate the proposed error propagation models.

3.1 Introduction

FRANS and HFRANS algorithms of Attallah [9, 11] and FDPM (fast data projection method) algorithm of Doukopoulos *et al.* [39–41] are typical examples of adaptive noise subspace estimation algorithms with computational complexity $O(NP)$. Simulation studies have shown that FRANS algorithm loses orthogonality with time due to accumulation of round-off errors [11]. As a result, the subspace estimate diverges from the true noise subspace. The HFRANS algorithm is derived by stabilizing FRANS using the numerically well behaved Householder transformation. The HFRANS and FDPM algorithms have been compared in a recent paper [11], and are shown to have comparable convergence performance. According to a very recent paper [39], the orthogonality error of FDPM algorithm does not grow, whereas the orthogonality error of HFRANS algorithm grows at a very slow rate. Doukopoulos and Moustakides [41] presented some results on the stability of FRANS algorithm. According to [41], FRANS is non-robust, since a parameter that measures the orthogonality error of FRANS theoretically grows exponentially. However, there is no simulation result given in [41] to prove the theoretical result. To overcome this problem, a parameter that is easily measurable through simulation is defined in this chapter. Our main motivation is to study the propagation of orthogonality error for FRANS and to understand how the Householder transformation helps to improve the stability in HFRANS.

3.2 Propagation of Orthogonality Error in FRANS

Let us first recall the steps of FRANS algorithm in Table 3.1. For the sake of simplicity, instead of using the adaptive step-size $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$, a small positive

```
Initialization: Choose $\mathbf{W}(0)$ such that $\mathbf{W}^H(0)\mathbf{W}(0) = \mathbf{I}_P$
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\delta(i) = 4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2$
3. $\rho(i) = \sqrt{1-\delta(i)}$
4. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}(\frac{1}{\rho(i)} - 1)$
5. $\mathbf{p}(i) = -\tau(i)\mathbf{W}(i-1)\mathbf{y}(i)/\beta + 2\mathbf{r}(i)(1+\tau(i)\|\mathbf{y}(i)\|^2)$
6. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i)$
```

Table 3.1: FRANS algorithm [9] for noise subspace estimation.

constant step-size β is used here. We rewrite the main steps of FRANS algorithm as

$$\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}\left[\frac{1}{\sqrt{1-4\beta\left(1-\beta\|\mathbf{r}(i)\|^2\right)\|\mathbf{y}(i)\|^2}} - 1\right] \tag{3.1a}$$

$$\mathbf{A}(i) = \mathbf{I}_P + \tau(i)\mathbf{y}(i)\mathbf{y}^H(i) \tag{3.1b}$$

$$\mathbf{W}(i+1) = \left[\mathbf{I}_N - 2\beta\mathbf{r}(i)\mathbf{r}^H(i)\right]\mathbf{W}(i)\mathbf{A}(i). \tag{3.1c}$$

Let us define the orthogonality error matrix as

$$\mathbf{\Delta}(i) = \mathbf{W}^H(i)\mathbf{W}(i) - \mathbf{I}_P. \tag{3.2}$$

Note that a non-zero $\mathbf{\Delta}(i)$ may arise from either accumulation of round-off errors or non-orthonormal initialization of $\mathbf{W}(i)$. Since we always initialize $\mathbf{W}(0)$ orthonormally, the orthogonality error that arises here is due to round-off errors. In the following sub-sections, we examine the propagation of orthogonality error for FRANS in the sense of mean and mean-square.

3.2.1 Mean Analysis of Orthogonality Error

Using the definition of $\tau(i)$ in (3.1a), we can easily show that

$$\frac{1}{\|\mathbf{y}(i)\|^2}\left[1-\frac{1}{(1+\tau(i)\|\mathbf{y}(i)\|^2)^2}\right]=4\beta(1-\beta\|\mathbf{r}(i)\|^2) \tag{3.3}$$

Multiplying both sides of (3.3) by $(1+\tau(i)\|\mathbf{y}(i)\|^2)^2$ leads to

$$2\tau(i)+\tau^2(i)\|\mathbf{y}(i)\|^2=4\beta(1-\beta\|\mathbf{r}(i)\|^2)(1+\tau(i)\|\mathbf{y}(i)\|^2)^2. \tag{3.4}$$

Based on the definition of $\mathbf{A}(i)$, the product $\mathbf{A}^H(i)\mathbf{A}(i)$ can be written as

$$\mathbf{A}^H(i)\mathbf{A}(i)=\mathbf{I}_P+(2\tau(i)+\tau^2(i)\|\mathbf{y}(i)\|^2)\mathbf{y}(i)\mathbf{y}^H(i). \tag{3.5}$$

Using (3.4) in (3.5), we obtain

$$\mathbf{A}^H(i)\mathbf{A}(i)=\mathbf{I}_P+4\beta\left(1-\beta\|\mathbf{r}(i)\|^2\right)\left(1+\tau(i)\|\mathbf{y}(i)\|^2\right)^2\mathbf{y}(i)\mathbf{y}^H(i). \tag{3.6}$$

Using (3.1c) and (3.6), we can write a recursion for $\mathbf{\Delta}(i)$ as

$$\begin{aligned}\mathbf{\Delta}(i+1)&=\mathbf{A}^H(i)\mathbf{\Delta}(i)\mathbf{A}(i)\\&=\mathbf{\Delta}(i)+\tau(i)\mathbf{y}(i)\mathbf{y}^H(i)\mathbf{\Delta}(i)+\tau(i)\mathbf{\Delta}(i)\mathbf{y}(i)\mathbf{y}^H(i)\\&\quad+\tau^2(i)\mathbf{y}(i)\mathbf{y}^H(i)\mathbf{\Delta}(i)\mathbf{y}(i)\mathbf{y}^H(i).\end{aligned} \tag{3.7}$$

A first-order approximation is commonly used in stochastic approximation theory [65] to reach a mathematically tractable solution for recursive equations of the type we have in (3.7). Let us consider the general formulation of a stochastic recursive algorithm [29, Chap 5]

$$\mathbf{\Theta}(i+1) = \mathbf{\Theta}(i) + \beta(i) f(\mathbf{x}(i), \mathbf{\Theta}(i)), \quad i = 0, 1, 2, \cdots \tag{3.8}$$

where $\mathbf{x}(i) \in \mathcal{C}^N$ is a sequence of random vectors, $\beta(i)$ is a sequence of step-size parameters, f is a continuous and bounded matrix-valued function, and $\mathbf{\Theta}(i) \in \mathcal{C}^{N\times P}$ is a sequence of approximations of some desired parameter matrix $\mathbf{\Theta}^*$. Kushner and Clark [65] showed that under certain conditions, the sequence $\mathbf{\Theta}(i)$ of (3.8) converges with probability 1 to the solution of an associated deterministic ordinary differential equation (ODE). The assumptions made in this work are as follows.

A-1

$$\beta(i) \to 0 \ \text{ and } \ \sum_{i=0}^{\infty} \beta(i) = \infty \tag{3.9}$$

A-2. $f(\cdot,\cdot)$ is a bounded and measurable $\mathcal{C}^{N\times P}$-valued function.

A-3. The function $f(\mathbf{x},\cdot)$ is continuous and bounded (uniformly in $\mathbf{x}$).

A-4. There is a function $\bar{f}(\mathbf{\Theta})$ such that

$$\bar{f}(\mathbf{\Theta}) = \lim_{i\to\infty} \frac{\sum_{i=k}^{\infty} \beta(k) f(\mathbf{x}(k), \mathbf{\Theta})}{\sum_{i=k}^{\infty} \beta(k)} = \lim_{i\to\infty} E\{f(\mathbf{x}(i), \mathbf{\Theta})\} \tag{3.10}$$

for all fixed $\mathbf{\Theta}$. Here, $\mathbf{x}(i)$ does not necessarily have to be a stationary sequence, but it is required to be "stationary in the limit" as $i \to \infty$, so that its expectation is independent of i.

Theorem: Assume that A-1 through A-4 holds and let $\mathbf{\Theta}(i)$ be bounded with probability 1. Then the sequence $\mathbf{\Theta}(\cdot)$ is a convergent sequence whose limit satisfies the ordinary differential equation (ODE)

$$\frac{\mathrm{d}\boldsymbol{\Theta}}{\mathrm{d}t} = \bar{f}(\boldsymbol{\Theta}). \tag{3.11}$$

Based on this theorem, we now present a convergence analysis of the orthogonality error $\boldsymbol{\Delta}(i)$ in FRANS given in (3.7), assuming that the step-size is very small.

Assuming β to be small, we can approximate $\tau(i)$ in (3.1c) using first-order Taylor's series as

$$\tau(i) \approx 2\beta. \tag{3.12}$$

Substituting (3.12) into (3.7), and dropping all the terms that contain β in second-order or above, we can approximate the recursion for the orthogonality error matrix $\boldsymbol{\Delta}(i)$ as

$$\boldsymbol{\Delta}(i+1) \approx \boldsymbol{\Delta}(i) + 2\beta\mathbf{y}(i)\mathbf{y}^H(i)\boldsymbol{\Delta}(i) + 2\beta\boldsymbol{\Delta}(i)\mathbf{y}(i)\mathbf{y}^H(i). \tag{3.13}$$

Assume that $\boldsymbol{\Delta}(i)$ and $\mathbf{y}(i)$ are independent and $\mathbf{W}(i)$ is close to the optimal noise subspace matrix given by $\mathbf{W}_o = \mathbf{U}_n\mathbf{B}$, where $\mathbf{B}$ is a $P \times P$ unitary matrix. This last assumption is valid only until the algorithm has not diverged completely. In fact, $\mathbf{W}(i)$ will hover around $\mathbf{W}_o$ after initialization and before divergence, as long as we have $\mathbf{W}(i)^H\mathbf{W}(i) = \mathbf{I}_P$. This represents the region of interest to us as we would like to quantify the increase in orthogonality error before the algorithm has completely diverged and before our assumption fails.

To justify this assumption (i.e. $\mathbf{W}(i)$ is close to $\mathbf{W}_o$), firstly, for the sake of analysis, we initialize the algorithm with the best initial condition, *i.e.* $\mathbf{W}(0) = \mathbf{U}_n$.

In this case, $\mathbf{W}(i)$ is very close to $\mathbf{W}_o$ as long as the accumulation of round-off errors does not become too large through time. Secondly, using the stochastic approximation approach by Kushner and Clark [65] and replacing β by a sequence that satisfies $\beta(i) \rightarrow 0$ and $\sum_{i=0}^{\infty} \beta(i) = \infty$, we can associate an ODE to the FRANS algorithm given by

$$\frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t} = \mathbf{W}(t)\mathbf{W}^H(t)\mathbf{C}\mathbf{W}(t) - \mathbf{C}\mathbf{W}(t), \tag{3.14}$$

where $\mathbf{C}$ is the $N \times N$ covariance matrix of the data $\mathbf{r}(i)$. This ODE has been studied extensively in the literature [29, Chap 3] [26]. The solution of this ODE will be the noise subspace if $\mathbf{W}^H(t)\mathbf{W}(t) = \mathbf{I}_P$ for all t, *i.e.* $\mathbf{W}(t)$ should always belong to the Stiefel manifold $\mathbf{O}_{N,P}$ [29, Chap 5] [32]. However, if $\mathbf{W}(t)$ deviates to $\mathbf{W}(t) + \mathrm{d}\mathbf{W}$ outside $\mathbf{O}_{N,P}$ due to round-off errors, then the deviation will grow and $\mathbf{W}(t)$ will eventually diverge from $\mathbf{O}_{N,P}$. Since we choose $\mathbf{W}^H(0)\mathbf{W}(0) = \mathbf{I}_P$, the algorithm will try to converge to the noise subspace until the accumulated round-off errors become too large. So, $\mathbf{W}(i)$ will be quite close to $\mathbf{W}_o$ for some iterations before the divergence due to accumulation of round-off errors takes place. As we shall see later in Section 3.4 on simulation results, initializing $\mathbf{W}(0)$ by $\mathbf{U}_n$ or using the first P columns of $\mathbf{I}_N$ leads to the same orthogonality propagation error (see Figures 3.1 and 3.3), which further supports our assumption.

Under the assumptions mentioned above, the expectation of (3.13) can be written as

$$E\left[\boldsymbol{\Delta}(i+1)\right] \approx E\left[\boldsymbol{\Delta}(i)\right] + 2\beta\mathbf{B}^H\boldsymbol{\Lambda}_n\mathbf{B}E\left[\boldsymbol{\Delta}(i)\right] + 2\beta E\left[\boldsymbol{\Delta}(i)\right]\mathbf{B}^H\boldsymbol{\Lambda}_n\mathbf{B}, \tag{3.15}$$

where $\boldsymbol{\Lambda}_n \in \Re^{P\times P}$ is a diagonal matrix formed by the smallest P eigenvalues of $\mathbf{C}$.

With $\tilde{\mathbf{\Delta}}(i+1) = \mathbf{B}\mathbf{\Delta}(i+1)\mathbf{B}^H$, from (3.15), we obtain

$$E\left[\tilde{\mathbf{\Delta}}(i+1)\right] \approx E\left[\tilde{\mathbf{\Delta}}(i)\right] + 2\beta\mathbf{\Lambda}_n E\left[\tilde{\mathbf{\Delta}}(i)\right] + 2\beta E\left[\tilde{\mathbf{\Delta}}(i)\right]\mathbf{\Lambda}_n. \tag{3.16}$$

We can express (3.16) in component form as

$$E\left[\tilde{\mathbf{\Delta}}_{m,l}(i+1)\right] \approx \left[1 + 2\beta\left(\lambda_m + \lambda_l\right)\right] E\left[\tilde{\mathbf{\Delta}}_{m,l}(i)\right], \tag{3.17}$$

where $1 \le m, l \le P$ and $\lambda_1 \le \cdots \le \lambda_P$ are the noise eigenvalues of $\mathbf{C}$, and $\tilde{\mathbf{\Delta}}_{m,l}(i)$ is the (m, l)th element of $\tilde{\mathbf{\Delta}}(i)$. Observe from (3.17) that the sequences $\{E[\tilde{\mathbf{\Delta}}_{m,l}(i)]\}$ are geometric progressions with common ratios $1 + 2\beta\left(\lambda_m + \lambda_l\right)$. Therefore, the elements of orthogonality error matrix $\mathbf{\Delta}(i)$ increase geometrically since λ_m and λ_l are non-negative scalars.

3.2.2 Mean-square Analysis of Orthogonality Error

In the mean-square analysis, we examine the orthogonality error $\eta(i) = E\left[\|\mathbf{W}^H(i)\mathbf{W}(i) - \mathbf{I}_P\|_F^2\right]$. Using the aforedefined orthogonality error matrix $\mathbf{\Delta}(i)$, we can write

$$\eta(i) = E\left[\mathrm{Tr}\left(\mathbf{\Delta}^H(i)\mathbf{\Delta}(i)\right)\right] = E\left[\mathrm{Tr}\left(\tilde{\mathbf{\Delta}}^H(i)\tilde{\mathbf{\Delta}}(i)\right)\right]. \tag{3.18}$$

From (3.13), we have

$$\tilde{\mathbf{\Delta}}(i+1) \approx \tilde{\mathbf{\Delta}}(i) + 2\beta\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\mathbf{\Delta}}(i) + 2\beta\tilde{\mathbf{\Delta}}(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i), \tag{3.19}$$

where $\tilde{\mathbf{y}}(i) = \mathbf{B}\mathbf{y}(i)$. Using (3.19) and assuming that β is small, $\mathbf{W}(i)$ is near $\mathbf{W}_o$, and $\tilde{\boldsymbol{\Delta}}(i)$ and $\tilde{\mathbf{y}}(i)$ are independent, we obtain

$$\begin{aligned}
&E\left[\tilde{\boldsymbol{\Delta}}^H(i+1)\tilde{\boldsymbol{\Delta}}(i+1)\right] \\
&\approx E\left[\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right] + 4\beta E\left[\tilde{\boldsymbol{\Delta}}^H(i)\boldsymbol{\Lambda}_n\tilde{\boldsymbol{\Delta}}(i)\right] + 2\beta E\left[\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\boldsymbol{\Lambda}_n\right] \\
&\quad + 2\beta E\left[\boldsymbol{\Lambda}_n\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right] + 4\beta^2 E\left[\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right] \\
&\quad + 4\beta^2 E\left[\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\right] \\
&\quad + 4\beta^2 E\left[\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right] \\
&\quad + 4\beta^2 E\left[\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^H(i)\right]. \qquad (3.20)
\end{aligned}$$

Applying trace operator on the above equation, we obtain

$$\begin{aligned}
&E\left[\mathrm{Tr}\left(\tilde{\boldsymbol{\Delta}}^H(i+1)\tilde{\boldsymbol{\Delta}}(i+1)\right)\right] \\
&\approx E\left[\mathrm{Tr}\left(\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right)\right] + 4\beta\sum_{m=1}^{P}\sum_{l=1}^{P}E\left[|\tilde{\boldsymbol{\Delta}}_{m,l}(i)|^2\right]\lambda_m \\
&\quad + 4\beta\sum_{m=1}^{P}\sum_{l=1}^{P}E\left[|\tilde{\boldsymbol{\Delta}}_{m,l}(i)|^2\right]\lambda_l, \qquad (3.21)
\end{aligned}$$

where the terms that are second order in β are dropped. Note that

$$\eta(i) = E\left[\mathrm{Tr}\left(\tilde{\boldsymbol{\Delta}}^H(i)\tilde{\boldsymbol{\Delta}}(i)\right)\right] = \sum_{m=1}^{P}\sum_{l=1}^{P}E\left[|\tilde{\boldsymbol{\Delta}}_{m,l}(i)|^2\right]. \qquad (3.22)$$

When the noise eigenvalues are equal (*i.e.* $\lambda_1 = \lambda_2 = \cdots = \lambda_P$), we denote the resulting orthogonality error by $\eta_{eq}(i)$. From (3.21) and (3.22), we obtain a recursion for $\eta_{eq}(i)$ as

$$\eta_{eq}(i) \approx (1 + 8\beta\lambda_1)^{i-1}\, \eta_{eq}(1). \tag{3.23}$$

When the noise eigenvalues are unequal, we denote the resulting orthogonality error by $\eta_{neq}(i)$. Further, by assuming that the mean-square values of all the elements of $\tilde{\mathbf{\Delta}}(i)$ are equal, we obtain

$$\eta_{neq}(i) = P^2 E\left[|\tilde{\mathbf{\Delta}}_{m,l}(i)|^2\right] \text{ for all } m, l, \tag{3.24}$$

which results in the solution for $\eta_{neq}(i)$ as

$$\eta_{neq}(i) \approx \left(1 + \frac{8\beta \sum_{m=1}^{P} \lambda_m}{P}\right)^{i-1} \eta_{neq}(1). \tag{3.25}$$

From (3.23) and (3.25), we observe that the evolution of orthogonality error sequences $\{\eta_{eq}(i)\}$ and $\{\eta_{neq}(i)\}$ is described by geometric progressions with common ratios exceeding 1.0. Further, these common ratios depend on the step-size and the noise eigenvalues of the covariance matrix.

3.3 Propagation of Orthogonality Error in HFRANS

As shown in the previous section, FRANS loses orthogonality with iterations. To improve numerical stability while retaining its computational complexity at $O(NP)$, Attallah proposed the implementation of FRANS algorithm with Householder transformation (HFRANS) [11]. The steps of HFRANS algorithm are given in Table 3.2. Again, for the sake of simplicity and to be consistent with our analysis for FRANS algorithm, the step-size β is also considered as a small positive constant here.

```
Initialization: choose $\mathbf{W}(0)$ such that $\mathbf{W}^H(0)\mathbf{W}(0) = \mathbf{I}_P$
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\delta(i) = 4\beta(1 - \beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2$
3. $\rho(i) = \sqrt{1 - \delta(i)}$
4. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}(\frac{1}{\rho(i)} - 1)$
5. $\mathbf{p}(i) = -\tau(i)\mathbf{W}(i-1)\mathbf{y}(i)/\beta + 2\mathbf{r}(i)(1 + \tau(i)\|\mathbf{y}(i)\|^2)$
6. $\mathbf{u}(i) = \mathbf{p}(i)/\|\mathbf{p}(i)\|$
7. $\mathbf{H}(i) = \mathbf{I}_P - 2\mathbf{u}(i)\mathbf{u}^H(i)$
8. $\mathbf{W}(i) = \mathbf{H}(i)\mathbf{W}(i-1)$
```

Table 3.2: HFRANS algorithm [11] for noise subspace estimation.

If there are no round-off errors, the instantaneous orthogonality error $\hat{\eta}(i)$ of HFRANS is given by

$$\begin{aligned}\hat{\eta}(i) &= \|\mathbf{W}^H(i)\mathbf{W}(i) - \mathbf{I}_P\|_F^2 \\ &= \|\mathbf{W}^H(i-1)\mathbf{H}^H(i-1)\mathbf{H}(i-1)\mathbf{W}(i-1) - \mathbf{I}_P\|_F^2 \\ &= \|\mathbf{W}^H(i-1)\mathbf{W}(i-1) - \mathbf{I}_P\|_F^2 = \hat{\eta}(i-1),\end{aligned} \tag{3.26}$$

since the Householder matrix $\mathbf{H}(i-1)$ is a unitary matrix. Equation (3.26) demonstrates that if there is infinite precision in calculation, the orthogonality error $\hat{\eta}(i)$ should always be zero, if $\mathbf{W}(0)$ is orthonormal. However, due to the use of finite precision in simulations, round-off errors are inevitable.

In HFRANS, $\mathbf{W}(i)$ is essentially $\mathbf{W}(0)$ left multiplied by a sequence of Householder matrices

$$\mathbf{W}(i) = \mathbf{H}(i-1)\mathbf{H}(i-2)\cdots\mathbf{H}(0)\mathbf{W}(0). \tag{3.27}$$

Therefore, orthogonality error can be expressed as

$$\eta(i) = E\left[\|\mathbf{W}^H(0)\mathbf{H}(0)\cdots\mathbf{H}(i-1)\mathbf{H}(i-1)\cdots\mathbf{H}(0)\mathbf{W}(0) - \mathbf{I}_P\|_F^2\right]. \tag{3.28}$$

A detailed error analysis of the sequence of the product of unitary and Householder matrices can be found in [102] and [103]. To keep the analysis simple, we neglect the round-off errors before the application of Householder transformation (Steps 1-5 in Table 3.2). In other words, round-off errors are assumed to arise from the Householder transformation step (Steps 6-8 in Table 3.2) only, which also implies that the orthogonality error is independent of the step-size. The latter is consistent with our observation from Figures 3.5 and 3.6 that the orthogonality error is not strongly dependent on the step-size.

According to [102], if $\mathbf{H}(i)$ is computed exactly at each stage, the transformation is performed exactly, and the computed transform is rounded before proceeding, the following bound exists

$$\eta(i) \leq 2^{-2k}\left[1 + (1+2^{-k}) + \cdots + (1+2^{-k})^{i-1}\right]^2 N, \tag{3.29}$$

where k is the number of bits used for finite precision representation. Equation (3.29) shows that if k is large, the corresponding bound is dominated by $i^2 2^{-2k} N$, which is independent of the eigenvalues and the step-size parameter β.

3.4 Simulation Results and Discussion

In our simulations, we choose the covariance matrix with unequal noise eigenvalues as

$$\mathbf{C}_{neq} = \mathbf{Q}_{neq}\mathbf{\Lambda}_{neq}\mathbf{Q}_{neq}^H = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix}, \tag{3.30}$$

where $N = 4$, $P = 2$, $\mathbf{\Lambda}_{neq} = \text{diag}[\ 0.0157,\ 0.1690,\ 0.6058, 2.3096]$, and $\mathbf{Q}_{neq}$ is formed by all the eigenvectors of $\mathbf{C}_{neq}$. The covariance matrix with equal noise eigenvalues is generated as

$$\mathbf{C}_{eq} = \mathbf{Q}_{neq}\mathbf{\Lambda}_{eq}\mathbf{Q}_{neq}^H, \tag{3.31}$$

where $\mathbf{\Lambda}_{eq} = \text{diag}[\ 0.0157, 0.0157, 0.6058, 2.3096]$. Generation of the observed data $\mathbf{r}(i)$ is done as stated in Section 2.5.1.

3.4.1 Results and Discussion for FRANS Algorithm

3.4.1.1 Initializing $\mathbf{W}(0)$ by $\mathbf{U}n$

Simulation results for FRANS are shown in Figures 3.1 and 3.2. Figure 3.1 shows the theoretical prediction of the orthogonality error $\eta_{eq}(i)$ using (3.23) and the corresponding simulation result $\bar{\eta}(i)$ for FRANS, when the covariance matrix ($\mathbf{C}_{eq}$) has equal noise eigenvalues and $\mathbf{W}(0)$ is initialized using $\mathbf{U}_n$. Two different step-sizes are used: $\beta = 0.05$ and $\beta = 0.1$. As can be seen, the theoretical prediction closely approximates simulation results.

Figure 3.2 shows the theoretical prediction of the orthogonality error $\eta_{neq}(i)$ using (3.25) and the corresponding simulation result $\bar{\eta}(i)$ for FRANS, when the covariance matrix ($\mathbf{C}_{neq}$) has unequal noise eigenvalues and $\mathbf{W}(0)$ is initialized using $\mathbf{U}_n$. Two different step-sizes are used: $\beta = 0.005$ and $\beta = 0.01$. As can be seen, the difference between simulation results and theoretical predictions in Figure 3.2 is larger than that in Figure 3.1. This difference is mainly due to the assumption of equal mean-square values for the elements of $\tilde{\boldsymbol{\Delta}}(i)$. Nevertheless, the theoretical predictions do give the trends quite accurately.

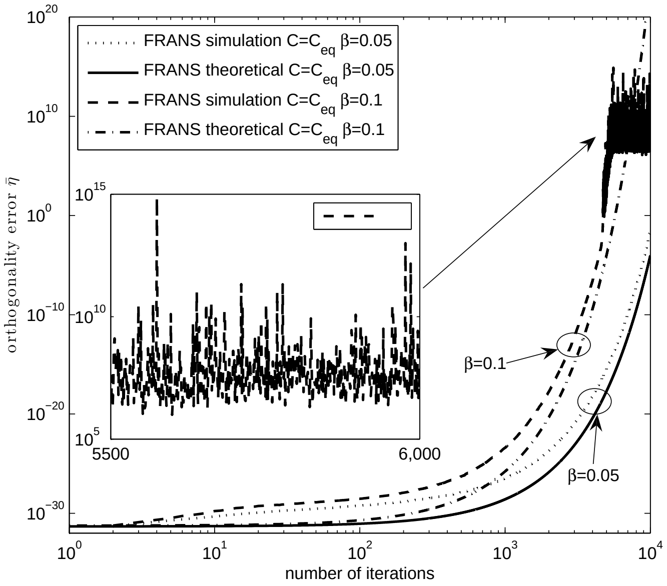

Fig. 3.1: Orthogonality error for FRANS with equal noise eigenvalues, initialized by $\mathbf{U}n$.

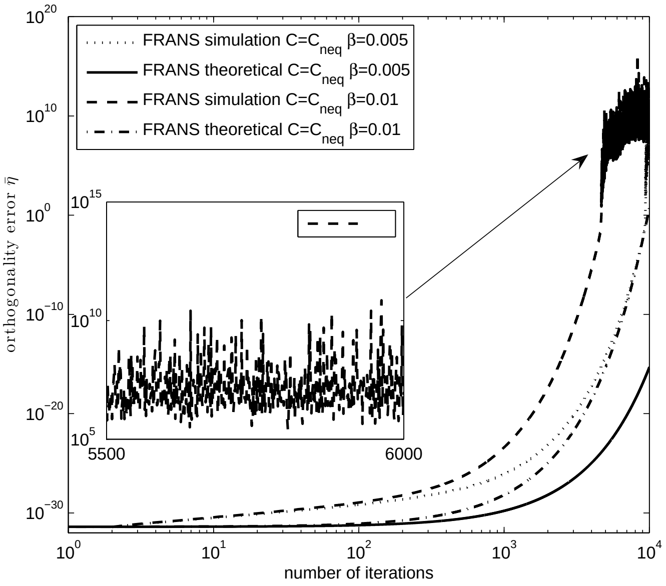

Fig. 3.2: Orthogonality error for FRANS with unequal noise eigenvalues, initialized by $\mathbf{U}n$.

3.4.1.2 Initializing $\mathbf{W}(0)$ by the First P Columns of $\mathbf{I}_N$

The orthogonality error results shown in Figure 3.3 are obtained under exactly the same conditions as that of Figure 3.1, except that $\mathbf{W}(0)$ is initialized using the first P columns of $\mathbf{I}_N$. Observe that Figure 3.3 is almost the same as Figure 3.1. The figure showing the simulation result of FRANS initialized by the first P columns of $\mathbf{I}_N$ with unequal noise eigenvalues is omitted since it is the same as Figure 3.2. Thus, we can see that the choice of initialization does not seem to influence the propagation of orthogonality error. Figure 3.4 shows the noise subspace estimation errors (see Section 2.5.2) of FRANS initialized by $\mathbf{U}_n$ and the first P columns of $\mathbf{I}_N$. Observe that irrespective of the choice of initialization, the algorithm hovers around $\mathbf{W}_o$ at least for the first few hundreds of iterations.

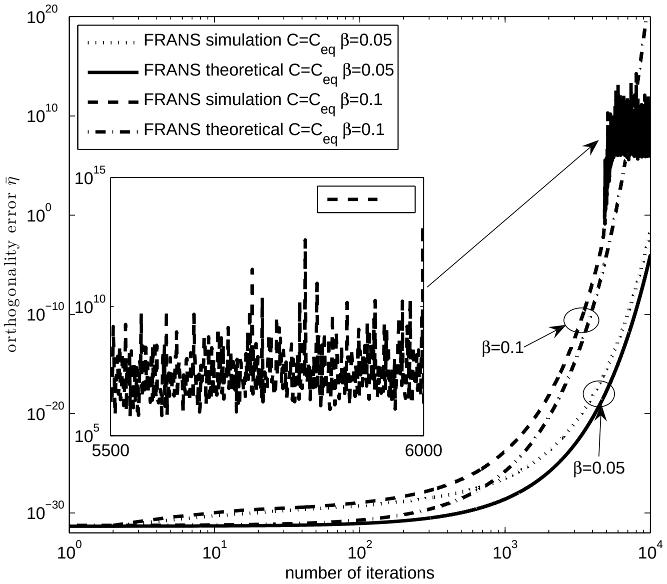

Fig. 3.3: Orthogonality error for FRANS with equal noise eigenvalues, initialized by the first P columns of $\mathbf{I}_N$.

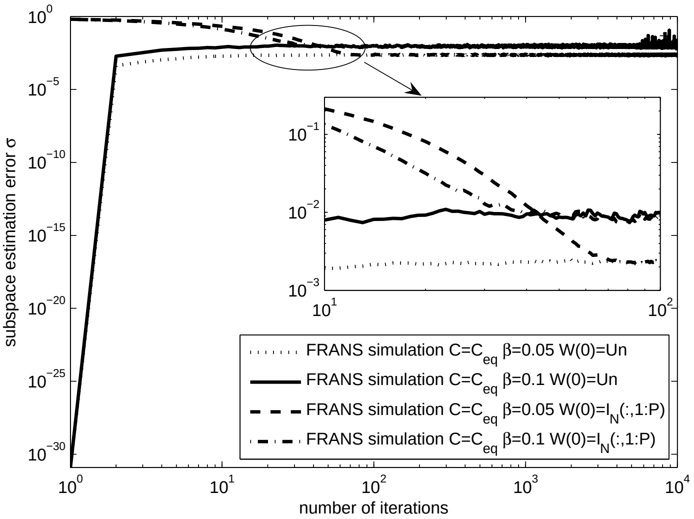

Fig. 3.4: Subspace estimation error for FRANS with equal noise eigenvalues.

3.4.1.3 Discussion on the Saturation of FRANS Algorithm

We can observe from Figures 3.1, 3.2 and 3.3 that the orthogonality error from simulations has a saturation floor for large iterations (see the zoom-in inside the figures). This is due to the divergence of the algorithm. When it loses orthogonality, the assumption that FRANS is based on ($\mathbf{W}^H(i-1)\mathbf{W}(i-1) = \mathbf{I}_P$) is no more valid. In other words, FRANS algorithm loses its theoretical support and becomes invalid once orthogonality is completely lost. Our main goal was to analyze the divergence rate of FRANS, when the algorithm is still valid. So the time slot we are interested in is the one before FRANS loses the orthogonality completely. The theoretical analysis does not exhibit this saturation behavior. This is because of the implicit assumption in FRANS that $\mathbf{W}(i)$ is orthonormal in each step.

3.4.2 Results and Discussion for HFRANS Algorithm

Figures 3.5 and 3.6 depict the orthogonality error for FRANS and HFRANS algorithms. In Figure 3.5, the covariance matrix used is $\mathbf{C}_{eq}$, and the step-sizes are $\beta = 0.05$ and $\beta = 0.1$. In Figure 3.6, the covariance matrix used is $\mathbf{C}_{neq}$, and the step-sizes are $\beta = 0.005$ and $\beta = 0.01$. Observe that the orthogonality error of HFRANS also grows with time, however, at a much slower rate than that of FRANS. Moreover, the orthogonality error $\bar{\eta}(i)$ seems to be independent of (at least not strongly dependent on) β, since the orthogonality error curves for HFRANS with different step-sizes are overlapping.

Figure 3.7 plots the orthogonality error for HFRANS and the theoretical bound given by (3.29). Since the simulation tool Matlab uses double precision, we have $k = 53$. In the simulations, the step-size used is $\beta = 0.01$ for both cases of the

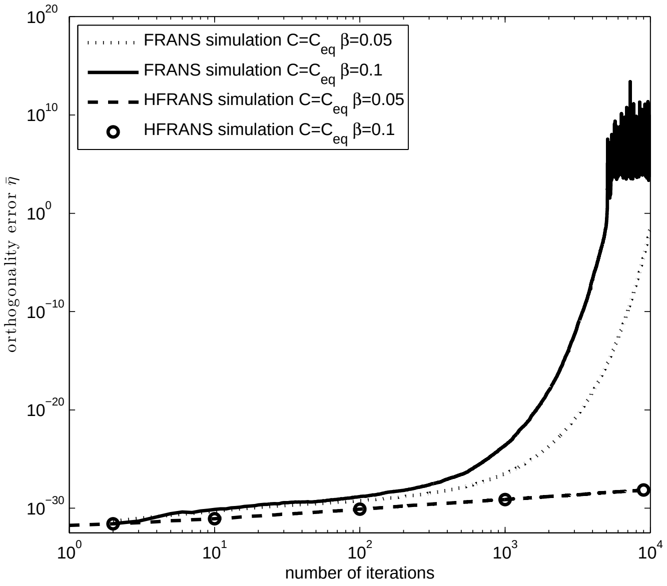

Fig. 3.5: Orthogonality error for FRANS and HFRANS with equal noise eigenvalues.

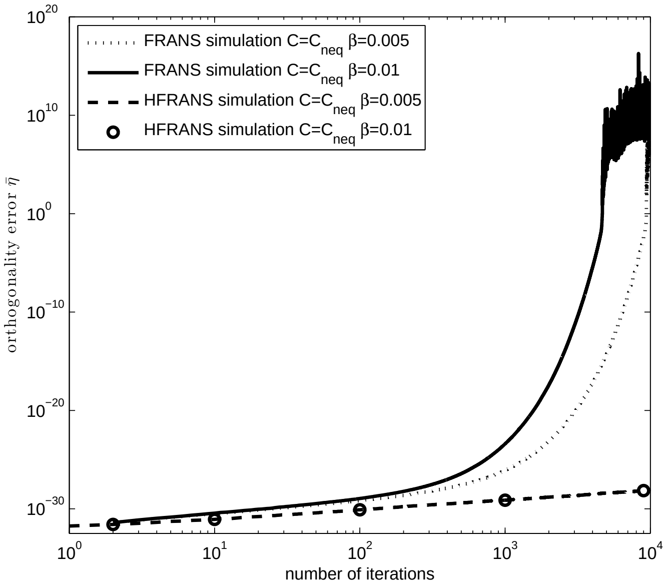

Fig. 3.6: Orthogonality error for FRANS and HFRANS with unequal noise eigenvalues.

covariance matrix: $\mathbf{C}_{eq}$ and $\mathbf{C}_{neq}$. Observe that the orthogonality error curves for the cases of equal and unequal eigenvalues are overlapping, implying that orthogonality error is independent of eigenvalues. The theoretical bound given by (3.29) is also independent of eigenvalues. As can be seen from Figure 3.7, although the theoretical upper bound (3.29) is a loose upper bound, it does give the trend in the propagation of orthogonality error for HFRANS. As can be seen, the orthogonality error curves of HFRANS and the theoretical bound grow extremely slowly. Therefore, HFRANS is numerically much more stable than FRANS.

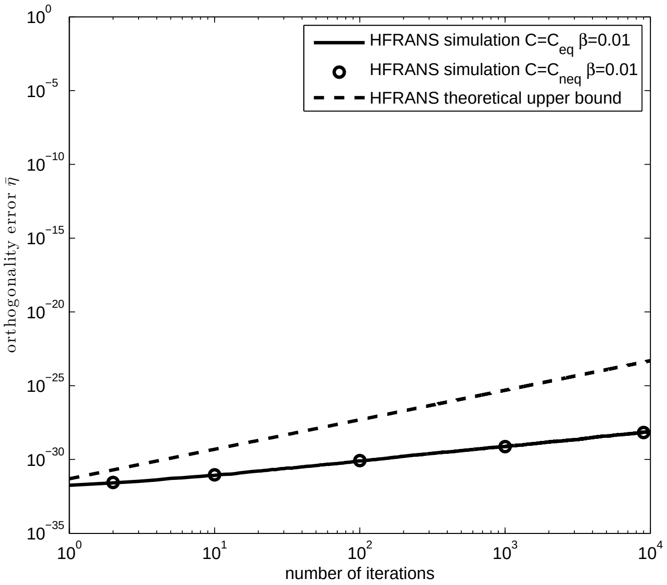

Fig. 3.7: Orthogonality error for HFRANS with equal and unequal noise eigenvalues and the theoretical upper bound (3.29).

3.5 Conclusion

In this chapter, we analyzed the propagation of orthogonality error for FRANS and HFRANS algorithms. It is shown that FRANS suffers from numerical instability since it accumulates numerical errors geometrically. However, HFRANS based on Householder transformation displays much better numerical behavior. Therefore, HFRANS algorithm is recommended for noise subspace estimation applications, especially since it has a low computational complexity which is composed of $O(NP)$ multiplications and only 1 square-root and 4 division operations per iteration.

Chapter 4

Variable Step-size Strategies for HFRANS

The HFRANS algorithm proposed by Attallah [11] is a stable adaptive noise subspace estimation algorithm with computational complexity $O(NP)$. In HFRANS, the adaptive step-size is chosen to be inversely proportional to the input signal power. To achieve better trade-off between convergence speed and steady-state error, we propose in this chapter a gradient step-size strategy and an optimal step-size strategy for HFRANS.

4.1 Introduction

Recall from the literature review in Chapter 2 that there are several linear complexity least-squares type signal subspace algorithms that converge rapidly [2, 109]. In contrast to this, existing low-complexity noise subspace algorithms are either unstable or gradient based that converge quite slowly. There is currently no linear complexity least-squares type noise subspace algorithm. Hence, our aim in this chapter is to develop novel strategies for improving the convergence performance of gradient type noise subspace algorithms. We choose HFRANS algorithm, which was shown in Chapter 3 as a stable linear complexity gradient based algorithm, for developing new approaches for improving convergence speed and reducing steady-state error. We may point out that the methods presented in this chapter are not limited to HFRANS, but could also be applied to other noise subspace algorithms and even signal subspace algorithms as well. First, in Section 4.2, we derive a gradient adaptive step-size strategy such that the step-size will gradually converge from a preset value to the optimal step-size. A main drawback of this strategy is the difficulty in choosing proper initial value and convergence rate for the step-size update. A bad choice of these parameters could lead to divergence. Therefore, in Section 4.3, we propose an optimal step-size strategy, which solves the initialization problem. Finally, in Section 4.4, using simulations, we assess the effect of the proposed step-size adaptation strategies under stationary and non-stationary (tracking) conditions.

4.2 Gradient Adaptive Step-size for HFRANS

The gradient adaptive step-size strategy applies the principle of stochastic gradient method to update the step-size parameter β in HFRANS. Recall from Section 2.4.2 that the cost function underlying HFRANS is $\mathrm{Tr}\left(\mathbf{W}^H\mathbf{C}\mathbf{W}\right)$, where $\mathbf{C}$ is the covariance matrix and $\mathbf{W}$ is the subspace matrix. Replacing $\mathbf{C}$ by its instantaneous value $\hat{\mathbf{C}}(i) = \mathbf{r}(i)\mathbf{r}^H(i)$ and $\mathbf{W}$ by its ith update $\mathbf{W}(i)$, we obtain the instantaneous value of the cost function as

$$\hat{J}_{\mathbf{W}}(i) = \mathrm{Tr}\left[\mathbf{W}^H(i)\hat{\mathbf{C}}(i)\mathbf{W}(i)\right], \tag{4.1a}$$

where

$$\mathbf{W}(i) = \mathbf{W}(i-1) - 2\beta(i-1)\hat{\mathbf{C}}(i-1)\mathbf{W}(i-1). \tag{4.1b}$$

Equation (4.1b) is same as equation (2.15) except for the orthogonalization step. For the sake of simplicity, we ignore the orthogonalization step while deriving the updating equation for β, since orthogonality is ensured by HFRANS at each iteration. Therefore, substituting (4.1b) in (4.1a), we can develop a gradient technique [73] to update the step-size as

$$\begin{aligned}\beta(i) = &\, \beta(i-1) - \mu\frac{\partial \hat{J}_{\mathbf{W}}(i)}{\partial \beta(i-1)} \\ = &\, \beta(i-1) - \mu\Big[-2\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)\mathbf{W}(i-1)\right) \\ &\quad -2\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i-1)\hat{\mathbf{C}}(i)\mathbf{W}(i-1)\right) \\ &\quad +8\beta(i-1)\mathrm{Tr}\left(\mathbf{W}^H(i-1)\mathbf{R}(i)\mathbf{W}(i-1)\right)\Big].\end{aligned} \tag{4.2}$$

where $\mathbf{R}(i) = \hat{\mathbf{C}}(i-1)\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)$.

Let us examine the convergence of $\beta(i)$ in the mean to derive the bounds on the step-size μ. Defining the error $v(i) = \beta(i) - \beta_o$, where β_o is the optimal step-size, we obtain from (4.2)

$$\begin{aligned}v(i) = &\, v(i-1)\left[1 - 8\mu\mathrm{Tr}\left(\mathbf{W}^H(i-1)\mathbf{R}(i)\mathbf{W}(i-1)\right)\right] \\ &\quad -8\mu\beta_o\mathrm{Tr}\left(\mathbf{W}^H(i-1)\mathbf{R}(i)\mathbf{W}(i-1)\right)\end{aligned}$$

$$+2\mu\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)\mathbf{W}(i-1)\right)$$
$$+2\mu\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i-1)\hat{\mathbf{C}}(i)\mathbf{W}(i-1)\right). \qquad (4.3)$$

Comparing equations (4.2) and (4.3), we see that the sum of the last three terms in (4.3) is equal to the negative gradient of the cost function with respect to the step-size parameter β when evaluated at the optimal step-size $\beta = \beta_o$. Therefore, the sum of the last three terms in (4.3) becomes zero for large enough i. Assume that $v(i)$, $\mathbf{W}(i-1)$ and $\hat{\mathbf{C}}(i)$ are mutually independent, and i is large enough such that $\mathbf{W}(i-1)$ is very close to $\mathbf{W}_o$, which is the optimal value of the subspace matrix given by $\mathbf{W}_o = \mathbf{U}_n\mathbf{B}$, where $\mathbf{U}_n$ is a $N \times P$ matrix formed by the eigenvectors of $\mathbf{C}$ corresponding to the smallest P eigenvalues, and $\mathbf{B}$ is a $P \times P$ unitary matrix. Hence, $\mathrm{span}(\mathbf{W}_o) = \mathrm{span}(\mathbf{U}_n)$. Therefore, we get

$$E[v(i)] = E[v(i-1)]\left[1 - 8\mu\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right)\right]. \qquad (4.4)$$

Thus, for $E[v(i)]$ to converge to zero, we require

$$\mid 1 - 8\mu\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right) \mid < 1,$$

which gives the bounds of μ as

$$0 < \mu < \frac{1}{4\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right)}. \qquad (4.5)$$

Evaluation of $\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right)$ is shown below. Let $\tilde{\mathbf{r}}(i) = \mathbf{Q}^H\mathbf{r}(i)$ and $\tilde{\mathbf{U}}_n = \mathbf{Q}^H\mathbf{U}_n$, where $\mathbf{Q}$ is a unitary matrix consisting of the orthonormal eigenvectors of $\mathbf{C}$. Using these, we get

$$\operatorname{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right) = \operatorname{Tr}\left(\tilde{\mathbf{U}}_n^H \mathbf{G} \tilde{\mathbf{U}}_n\right), \tag{4.6}$$

where $\mathbf{G} = E\left[\tilde{\mathbf{r}}(i-1)\tilde{\mathbf{r}}^H(i-1)\tilde{\mathbf{r}}(i)\tilde{\mathbf{r}}^H(i)\tilde{\mathbf{r}}(i-1)\tilde{\mathbf{r}}^H(i-1)\right]$. For the sake of simplifying the analysis, we assume that $\{\mathbf{r}(i)\}$ is a sequence of zero mean and independent Gaussian vectors. Then, we get the (k,l)th element of $\mathbf{G}$ as

$$g_{k,l} = \sum_{m=1}^{N} \mathrm{E}\left(\tilde{r}_k \tilde{r}_l^* |\tilde{r}_m|^2\right)\lambda_m, \tag{4.7}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ are the eigenvalues of $\mathbf{C}$, and $\tilde{r}_k$ denotes the kth element of $\tilde{\mathbf{r}}(i-1)$. Note that $\tilde{r}_k$, $\tilde{r}_l$ and $\tilde{r}_m$ are independent and zero mean complex Gaussian random variables, we can obtain

$$E\left(\tilde{r}_k \tilde{r}_l^* |\tilde{r}_m|^2\right) = \begin{cases} 0 & \text{for } k \neq l \\ \lambda_k \lambda_m \delta(k-l)\left[1-\delta(l-m)\right] & \text{for } k = l, l \neq m \\ 2\lambda_k^2 \delta(k-l)\delta(k-m) & \text{for } k = l = m. \end{cases} \tag{4.8}$$

Based on (4.8), the (k,k)th element of $\mathbf{G}$ is of the form

$$g_{k,k} = \lambda_k \sum_{m=1}^{N} \lambda_m^2 + \lambda_k^3, \tag{4.9}$$

and $g_{k,l} = 0$ if $k \neq l$. Using (4.9) in (4.6), we get

$$\begin{aligned}\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right) &= \mathrm{Tr}\left(\tilde{\mathbf{U}}_n^H \mathbf{G}\tilde{\mathbf{U}}_n\right)\\ &= \sum_{k=1}^{P}\sum_{m=1}^{N}\lambda_k\lambda_m^2 + \sum_{k=1}^{P}\lambda_k^3. \end{aligned} \tag{4.10}$$

Using (4.10) in (4.5), the bounds of μ are found to be

$$0 < \mu < \frac{1}{4\left(\sum_{k=1}^{P}\sum_{m=1}^{N}\lambda_k\lambda_m^2 + \sum_{k=1}^{P}\lambda_k^3\right)}. \tag{4.11}$$

Assuming that $\{\mathbf{r}(i)\}$ is a sequence of zero mean and independent Gaussian vectors, $\mathbf{W}(i-1)$ and $\hat{\mathbf{C}}(i)$ are mutually independent and $\mathbf{W}(i-1)$ is very close to $\mathbf{W}_o$, we have

$$\left.\begin{aligned}\textstyle\sum_{k=1}^{P}\lambda_k &= E\left[\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i)\mathbf{W}(i-1)\right)\right] = E[\|\mathbf{y}(i)\|^2]\\ \textstyle\sum_{k=1}^{P}\lambda_k^3 &= E\left[\mathrm{Tr}\left(\mathbf{W}^H(i-2)\hat{\mathbf{C}}(i-1)\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-2)\mathbf{W}(i-3)\right)\right]\\ &= E[\varsigma^*(i)\kappa(i)\mathbf{y}^H(i-2)\mathbf{y}(i-1)]\\ \textstyle\sum_{k=1}^{N}\lambda_k^2 &= E\left[\mathrm{Tr}\left(\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)\right)\right] = E[|\varsigma(i)|^2],\end{aligned}\right\} \tag{4.12}$$

where

$$\varsigma(i) = \mathbf{r}^H(i)\mathbf{r}(i-1) \qquad \text{and} \qquad \kappa(i) = \mathbf{r}^H(i)\mathbf{r}(i-2). \tag{4.13}$$

Therefore, estimates of $\sum_{k=1}^{P}\lambda_k$, $\sum_{k=1}^{P}\lambda_k^3$ and $\sum_{k=1}^{N}\lambda_k^2$ can be approximated by the following time averages, $\theta_1(i)$, $\theta_2(i)$ and $\theta_3(i)$, respectively, as

$$\left.\begin{aligned}
&\theta_1(i) = (1-\nu(i))\theta_1(i-1) + \nu(i)\|\mathbf{y}(i)\|^2\\
&\theta_2(i) = (1-\nu(i))\theta_3(i-1) + \nu(i)\mathrm{Re}[\varsigma^*(i)\kappa(i)\mathbf{y}^H(i-2)\mathbf{y}(i-1)]\\
&\theta_3(i) = (1-\nu(i))\theta_4(i-1) + \nu(i)|\varsigma(i)|^2,
\end{aligned}\right\} \tag{4.14}$$

where $\nu(i) = \frac{1}{i}$ for stationary signal and it is set to a small value $(0 < \nu(i) < 1)$ for non-stationary signal. Using (4.14), a practical bound of $\mu(i)$ can be obtained as

$$0 < \mu(i) < \frac{1}{4(\theta_1(i)\theta_3(i) + \theta_2(i))}. \tag{4.15}$$

The gradient adaptive step-size strategy for HFRANS given in (4.2) can be expressed as

$$\begin{aligned}
\mathbf{y}(i) &= \mathbf{W}^H(i-1)\mathbf{r}(i) \qquad (4.16)\\
\mathbf{z}(i) &= \mathbf{W}^H(i-1)\mathbf{r}(i-1)\\
\varsigma(i) &= \mathbf{r}^H(i)\mathbf{r}(i-1)\\
\beta(i) &= \beta(i-1) + 4\mu(i)\left(\mathrm{Re}\left[\varsigma(i)\mathbf{z}^H(i)\mathbf{y}(i)\right] - 2\beta(i-1)|\varsigma(i)|^2\|\mathbf{y}(i-1)\|^2\right), \qquad (4.17)
\end{aligned}$$

where $\mu(i)$ is bounded by (A.5) and $\mathrm{Re}[\upsilon]$ denotes the real part of υ.

The gradient adaptive step-size method developed above is for complex-valued data. The gradient adaptive step-size method developed for real-valued data is given in Appendix A.1.

4.3 Optimal Step-size for HFRANS

In the last section, we proposed a gradient step-size strategy for HFRANS algorithm. However, performance of the algorithm will rely on the choices for initial step-size and update rate of the step-size. For example, if the update rate is set to be too small, it takes too long before the step-size converges to the optimal value, and the algorithm will act just like using constant step-size; if the update rate is set to be too large, the step-size might fluctuate and not converge to the optimal value, etc. To avoid these problems, we propose a simple optimal step-size strategy, which does not have to deal with any special initialization or updating rate.

Let $\beta_o(i)$ be the instantaneous optimal of $\beta(i)$, *i.e.* the solution of $\frac{\partial \hat{J}_{\mathbf{W}}(i)}{\partial \beta(i-1)} = 0$. Therefore, setting $\frac{\partial \hat{J}_{\mathbf{W}}(i)}{\partial \beta(i-1)} = 0$ [see (4.2)], taking expectation of the result, and assuming that i is large enough such that $\mathbf{W}(i-1)$ is close to $\mathbf{W}_o$, we get

$$E[\beta_o(i-1)] = \frac{\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)\right]\mathbf{W}_o\right) + \mathrm{Tr}\left(\mathbf{W}_o^H E\left[\hat{\mathbf{C}}(i-1)\hat{\mathbf{C}}(i)\right]\mathbf{W}_o\right)}{4\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right)}. \tag{4.18}$$

Following in similar lines as in the last section, (4.18) can be simplified to

$$E[\beta_o(i-1)] = \frac{\sum_{k=1}^{P}\lambda_k^2}{2\sum_{k=1}^{P}\lambda_k\left(\lambda_k^2 + \sum_{m=1}^{N}\lambda_m^2\right)}, \tag{4.19}$$

where λ_k, $1 \leq k \leq P$, are the eigenvalues of the noise subspace and λ_k, $(P+1) \leq k \leq N$, are the eigenvalues of the signal subspace.

Estimates of $\sum_{k=1}^{P}\lambda_k$, $\sum_{k=1}^{P}\lambda_k^3$ and $\sum_{k=1}^{N}\lambda_k^2$ can be obtained as given in (4.14).

Assuming that $\{\mathbf{r}(i)\}$ is a sequence of zero mean and independent Gaussian vectors, $\mathbf{W}(i-1)$ and $\hat{\mathbf{C}}(i)$ are mutually independent and $\mathbf{W}(i-1)$ is very close to $\mathbf{W}_o$, we have

$$\sum_{k=1}^{P}\lambda_k^2 = E\left[\mathrm{Tr}\left(\mathbf{W}^H(i-1)\hat{\mathbf{C}}(i)\hat{\mathbf{C}}(i-1)\mathbf{W}(i-1)\right)\right] = E[\mathbf{z}^H(i)\mathbf{y}(i)\varsigma(i)]. \tag{4.20}$$

Therefore, the estimate of $\sum_{k=1}^{P}\lambda_k^2$, denoted by θ_4, can be obtained by

$$\theta_4(i) = (1-\nu(i))\theta_2(i-1) + \nu(i)\mathrm{Re}[\varsigma(i)\mathbf{z}^H(i)\mathbf{y}(i)], \tag{4.21}$$

where $\nu(i) = \frac{1}{i}$ for stationary signal and it is set to a small value $(0 < \nu(i) < 1)$ for non-stationary signal. Thus, the estimated optimal step-size for HFRANS at each instant is obtained as

$$\beta(i) = \frac{\theta_4(i)}{2\left(\theta_2(i) + \theta_1(i)\theta_3(i)\right)}. \tag{4.22}$$

To stabilize (4.22), we propose to add 2 positive constant α and γ into (4.22)

$$\beta(i) = \alpha\frac{\theta_4(i)}{2\left(\theta_2(i) + \theta_1(i)\theta_3(i)\right) + \gamma}. \tag{4.23}$$

The optimal step-size for HFRANS obtained above is for complex-valued data. The optimal step-size for HFRANS with real-valued data is given in Appendix A.2.

4.4 Simulation Results and Discussion

4.4.1 Performance under Stationary Conditions

In simulations, we set the covariance matrix to

$$\mathbf{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix},$$

where $N = 4$, and $P = 2$. Generation of the observed data $\mathbf{r}(i)$ is done as described in Section 2.5.1.

Figure 4.1 shows the evolution of the subspace estimation error $\sigma(i)$ [see (2.35) in Chapter 2] for HFRANS, HFRANS with gradient adaptive step-size (denoted by GHFRANS) and HFRANS with optimal step-size strategy (denoted by OHFRANS). In HFRANS, $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$, where α is chosen to be 0.06. Here, α is chosen such that HFRANS and GHFRANS have the similar convergence speed. In GHFRANS, we set $\beta(0) = 0.03$ and $\mu = 0.0003$. In OHFRANS, $\nu(i) = \frac{1}{i}$, $\alpha = 0.5$, and $\gamma = 0.001$. As can be seen, all the three algorithms have good numerical stability because of the use of Householder transformation in HFRANS. Further, GHFRANS and OHFRANS have smaller steady-state error than the original HFRANS algorithm.

Figure 4.2 presents the orthogonality error $\eta(i)$ [see (2.36) in Chapter 2] for each iteration. Clearly, all the three algorithms have similar orthogonality error, owing to the numerical stability of HFRANS due to the use of Householder transformation.

Figure 4.3 presents the step-size $\beta(i)$ for each iteration. To get a better of view of how step-size is updated, Figure 4.3 is zoomed to show only the first 1000 iterations. As can be seen, the step-size of HFRANS is fluctuated since an instantaneous step-size $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$ is used. Both the step-sizes of GHFRANS and OHFRANS start with a big value, and gradually go smaller, which ensures it to achieve a good trade-off between the convergence speed and the steady-state error.

We now present the computational complexities of the three algorithms. The original HFRANS requires $4NP$ multiplications, 1 square root and 5 divisions, GHFRANS requires $5NP$ multiplications, 1 square root and 4 divisions, and OHFRANS requires $4NP$ multiplications, 1 square root and 5 divisions (when $\nu(i) = \frac{1}{i}$). Therefore, HFRANS, GHFRANS and OHFRANS have similar computational complexity, and they are all linear complexity algorithms.

4.4.2 Performance under Non-stationary Conditions: Tracking

To examine the tracking ability of HFRANS algorithms with different step-size strategies, we generate the observed data vector as

$$\mathbf{r}(i) \begin{cases} \text{where } E[\mathbf{r}(i)\mathbf{r}^H(i)] = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H, \text{for } i \leq i_o \\ \text{where } E[\mathbf{r}(i)\mathbf{r}^H(i)] = \hat{\mathbf{Q}}\mathbf{\Lambda}\hat{\mathbf{Q}}^H, \text{for } i > i_o \end{cases} \tag{4.24}$$

where $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_P, \mathbf{q}_{P+1}, \cdots, \mathbf{q}_N]$ and $\mathbf{q}_i$ is the corresponding eigenvector of λ_i, $\mathbf{\Lambda} = \text{diag}[\lambda_1, \cdots, \lambda_N]$, and $\hat{\mathbf{Q}} = [\mathbf{q}_{P+1}, \cdots, \mathbf{q}_N, \mathbf{q}_1, \cdots, \mathbf{q}_P]$. Let $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H = \mathbf{C}$, where $\mathbf{C}$ is same as that given in Section 4.4.1. As in [92], we calculate a

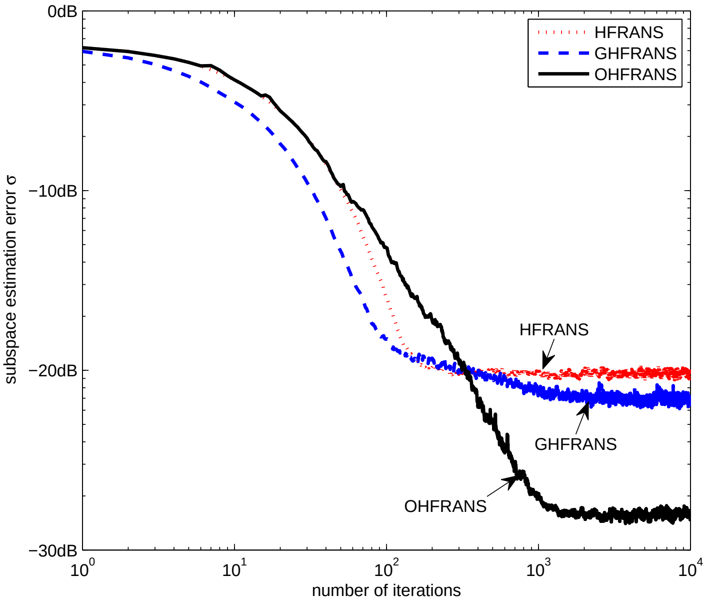

Fig. 4.1: Subspace estimation error for HFRANS, GHFRANS, and OHFRANS.

performance measure as

$$\phi(i) = \frac{1}{r_0} \sum_{r=1}^{r_0} \arccos\left(\sqrt{\varpi_r(i)}\right), \tag{4.25}$$

where $\varpi_r(i)$ denotes the smallest eigenvalue of $\mathbf{W}_r^H(i)\mathbf{U}_n\mathbf{U}_n^H\mathbf{W}_r(i)$ in the rth run, $r_0 = 1000$, and the unit of $\phi(i)$ is degree. The batch EVD (eigenvalue decomposition) is calculated as a benchmark for the tracking ability of HFRANS with different step-size strategies. The covariance matrix $\mathbf{T}(i)$ used for batch EVD is obtained as

$$\mathbf{T}(i) = \zeta\mathbf{T}(i-1) + (1-\zeta)\mathbf{r}(i)\mathbf{r}^H(i), \tag{4.26}$$

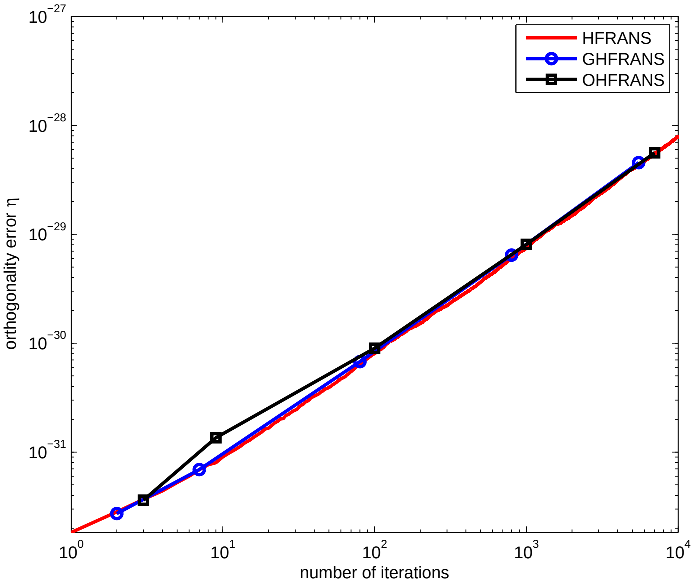

Fig. 4.2: Orthogonality error for HFRANS, GHFRANS, and OHFRANS.

where $\mathbf{T}(0)$ is initialized as $\mathbf{I}_N$, and $0 < \zeta < 1$. To be fair in comparison, the batch EVD is implemented at every iteration.

In Figure 4.4, we plot the dominant principal angle $\phi(i)$ for HFRANS, GHFRANS, OHFRANS and batch EVD. In HFRANS, $\alpha = 0.08$. In GHFRANS, we set $\beta(0) = 0.025$ and $\mu = 0.0000001$. In OHFRANS, $\nu(i)$ is set to be 0.005, $\alpha = 1$, and $\gamma = 0$. For batch EVD, $\nu(i)$ is set to be 0.005. Further, we set $i_o = 500$ in (4.24). Observe from Figure 4.4 that $\phi(i)$ has a 90° sudden change at 500th iteration because of the subspace rotation taking place in the observed data $\mathbf{r}(i)$. It is shown that all the three HFRANS algorithms have good tracking ability and are comparable with batch EVD. In particular, the OHFRANS algorithm performs closest to batch EVD.

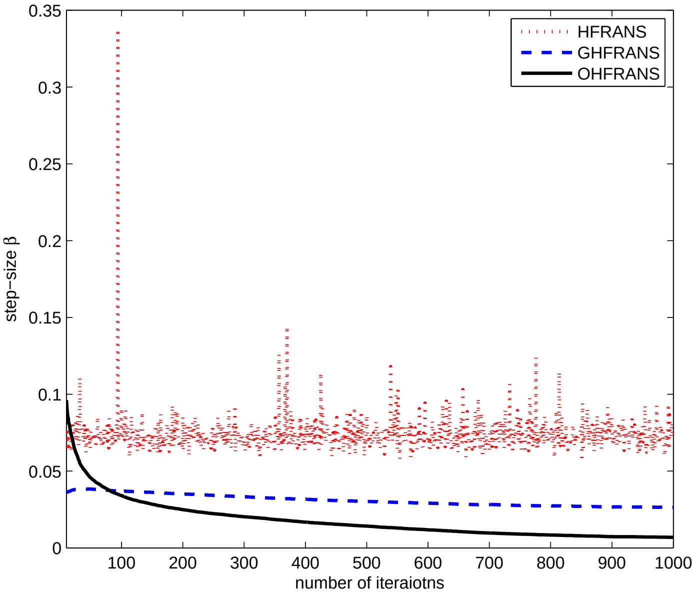

Fig. 4.3: Step-size adaptation for HFRANS, GHFRANS, and OHFRANS.

4.4.3 Application to MC-CDMA System with Blind Channel Estimation

As we know, the performance of subspace-based algorithms depends, to a large extent, on the speed and accuracy of the subspace estimation process. In this section, the proposed OHFRANS algorithm is applied to MC-CDMA (Multi-Carrier Code Division Multiple Access) system with blind channel estimation to show how the improvement in subspace estimation algorithms can influence the performance of subspace-based algorithms.

The received MC-CDMA signal vector is given by [43]

$$\mathbf{r}_n = \sum_{i=1}^{N-P} s_n^i \mathbf{C}_i \mathbf{F} \mathbf{h}_i + \mathbf{\Gamma}_n \tag{4.27}$$

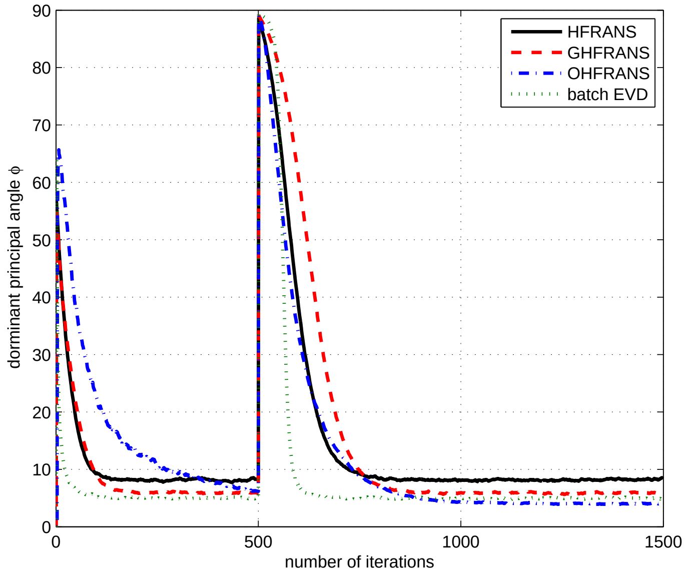

Fig. 4.4: Dominant principal angle, $\phi(i)$, for HFRANS, GHFRANS, OHFRANS and batch EVD.

where s_n^i denotes the nth transmitted symbol from the ith user, $(N-P)$ is the number of users, $\mathbf{C}_i$ is a $N \times N$ diagonal matrix whose elements are the N chips of the code corresponding to the ith user, $\mathbf{F}$ is a $N \times M$ DFT (discrete Fourier transform) matrix, $\mathbf{h}_i = [h_i(0), \cdots, h_i(M-1)]^T$ is the complex-valued channel impulse response vector and $\mathbf{\Gamma}_n = [\gamma_n(0), \cdots, \gamma_n(N-1)]^T$ is complex-valued additive white Gaussian noise. We use HFRANS and OHFRANS algorithms to do blind estimation of the impulse response of the channel. As shown in [43], the channel can be estimated blindly as

$$\hat{\mathbf{h}}_i = \arg \min_{\|\tilde{\mathbf{h}}_i\|^2=1} \tilde{\mathbf{h}}_i^H \left[\mathbf{F}^H \mathbf{C}_i^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{C}_i \mathbf{F}\right] \tilde{\mathbf{h}}_i. \tag{4.28}$$

Since $\mathbf{W}_o = \mathbf{U}_n\mathbf{B}$, where $\mathbf{B}$ is a $P \times P$ unitary matrix, it is obvious that

$$\mathbf{W}_o\mathbf{W}_o^H = \mathbf{U}_n\mathbf{U}_n^H. \tag{4.29}$$

Therefore, in the simulation, $\mathbf{W}(i)$ is obtained using HFRANS or OHFRANS algorithm. Then, $\mathbf{W}(i)\mathbf{W}^H(i)$ is calculated to estimate $\mathbf{U}_n\mathbf{U}_n^H$. Finally, the channel can be obtained from (4.28).

In Figures 4.5, the MSE (mean-square error) of the channel estimation is obtained for time-invariant Rayleigh fading channel. Without lost of generality, let the first user be the desired user. The MSE is measured by

$$MSE(i) = \frac{1}{r_0}\sum_{r=1}^{r_0} \|\mathbf{h}_1 - \hat{\mathbf{h}}_{1,r}^H(i)\mathbf{h}_1\hat{\mathbf{h}}_{1,r}(i)\|^2, \tag{4.30}$$

where $MSE(i)$ is the MSE of channel estimation at ith instant, r_0 denotes the number of Monte Carlo runs, and $\hat{\mathbf{h}}_{1,r}(i)$ denotes $\hat{\mathbf{h}}_1(i)$ in the rth run. The settings are: $N = 12$, $N - P = 8$ and $M = 4$, s_n^i is differential binary phase shift keying (DBPSK) modulated signal, SNR (signal to noise ratio)=10, diagonal elements of $\mathbf{C}_i \in \{+1, -1\}$ and $\mathbf{C}_i \neq \mathbf{C}_j$ when $i \neq j$. In HFRANS, α is set to be 0.025, nonstationary condition is considered with $\nu(i) = 0.005$. In OHFRANS, nonstationary condition is assumed, $\nu(i)$ is set to be 0.005, $\alpha = 1$, and $\gamma = 0$. As can be seen, the optimal step-size results in increased convergence rate for the MSE of the channel estimation.

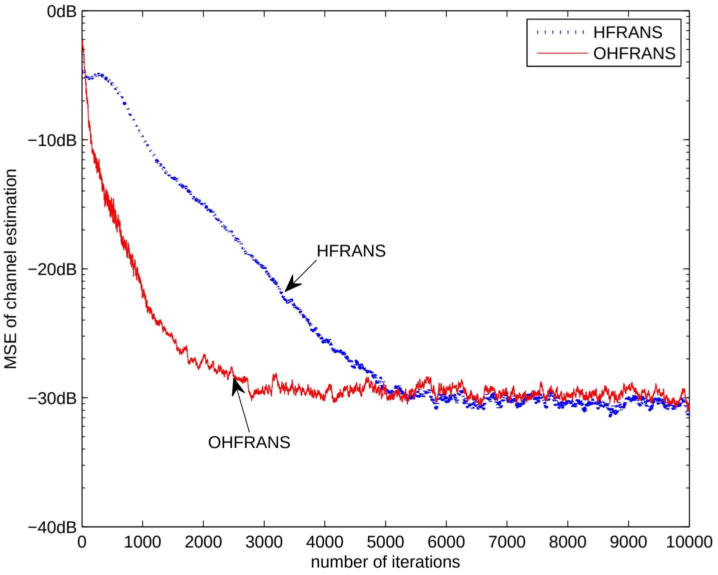

Fig. 4.5: MSE of channel estimation using HFRANS and OHFRANS algorithms.

4.5 Conclusion

In this chapter, we proposed a gradient adaptive step-size strategy and an optimal step-size strategy to enhance the performance of HFRANS algorithm. During simulation study, we have noticed that, under stationary conditions, GHFRANS has faster convergence and smaller steady-state error than the original HFRANS algorithm and OHFRANS has smaller steady-state error than HFRANS. Under nonstationary conditions (abrupt changes), HFRANS with proposed step-size strategies exhibits improvement in tracking ability over the original HFRANS algorithm.

Chapter 5

An Optimal Diagonal Matrix Step-size Strategy for Adaptive Noise Subspace Estimation Algorithms

In this chapter, we propose an optimal diagonal-matrix step-size strategy for two adaptive noise subspace estimation algorithms, modified Oja's algorithm (MOja) [105] and Yang and Kaveh's algorithm (YK) [110]. The proposed step-size strategy controls the updating of subspace vectors individually as compared to conventional methods where all the subspace vectors are updated using a single step-size parameter. Nevertheless, MOja and YK with the proposed strategy still have issues of instability or high computational complexity that these algorithms originally had. Several stable low cost implementations are then designed to restore stability while maintaining computational complexity at $O(NP)$. Optimal value of the diagonal step-size matrix is obtained using the method developed in Chapter 4 [see Section 4.3]. Simulation results show that this optimal diagonal-matrix step-size strategy outperforms the original algorithms.

5.1 Introduction

For real-time applications, it is desirable for the subspace algorithm to have a low computational complexity. In the literature, some of the well-known noise subspace algorithms with computational complexity $O(NP)$ are Chen's algorithm [23], SMSR [37] and MOja [105]. These algorithms can estimate in parallel a number of subspace vectors. However, the step-sizes used in these algorithms are all constant scalars. A bad choice (large value) of the step-size can lead to the algorithms' divergence. To ensure the algorithms' convergence, normalized step-sizes were proposed, such as the ones in NOOja [8] and HFRANS [11]. However, these proposals use a single normalized step-size parameter to update all the subspace vectors. As we know, the estimated subspace matrix is a rotation of the true eigen noise subspace, and each of the estimated subspace vector has its own dynamics since the associated eigenvalues are different. Hence, updating all the subspace vectors using the same step-size could slow down the overall convergence. To tackle this problem, we propose to use a different step-size for each of the subspace vectors. Therefore, instead of the original scalar step-size, a diagonal matrix $\mathbf{\Lambda}$ with the step-sizes of the corresponding vectors on its main diagonal is formed.

The algorithms considered in this chapter are modified Oja's algorithm (MOja) [105] and Yang and Kaveh's algorithm (YK) [110]. As we claimed in Chapter

2, if we categorize them by the cost functions used, NOOja [8] and FOOja [21] algorithms are derived from MOja algorithm; FDPM [41] and HFRANS [11] algorithms are low cost versions of YK algorithm. If we categorize them by the low cost orthonormaliztion methods used, NOOja and HFRANS use direct orthonormaliztion method; FOOja and FDPM use separate orthogonalization and normalization method. Therefore, NOOja is MOja algorithm with direct orthonormaliztion method; FOOja is MOja algorithm with separate orthogonalization and normalization method; HFRANS is YK algorithm with direct orthonormaliztion method; FDPM is YK algorithm with separate orthogonalization and normalization method. Hence, by considering MOja and YK algorithms, we have taken into account NOOja, FOOja, FDPM and HFRANS algorithms.

MOja is derived from its signal subspace counterpart, Oja's algorithm [78], by reversing the sign of the step-size. It has computational complexity $O(NP)$. However, it is not stable, as it loses orthogonality rapidly. Although YK is stable, it has computational complexity $O(NP^2)$, which is relatively high for real-time implementation. With the proposed diagonal matrix step-size strategy, the performance of MOja and YK are improved. Nevertheless, they still have the issues of instability and high computational complexity. To improve stability and reduce computational cost, different implementation methods are presented. In the following, we first design two methods to stabilize MOja with diagonal matrix step-size. Therefore, three techniques are presented to simplify the implementation of YK with diagonal matrix step-size.

5.2 Diagonal Matrix Step-size Strategy (DMSS) for MOja

The MOja algorithm is recalled in Table 5.1. Using a diagonal step-size matrix $\mathbf{\Lambda}(i) = \mathrm{diag}[\beta_1(i) \cdots \beta_P(i)]$, where $\beta_k(i)$ is the step-size for the kth column of $\mathbf{W}(i)$, MOja with diagonal matrix step-size can be expressed as in Table 5.2. If we compare

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
3. $\mathbf{T}_{\mathrm{MOja}}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i)$
4. $\mathbf{W}(i) = \mathbf{T}_{\mathrm{MOja}}(i)$
```

Table 5.1: MOja algorithm [105] for noise subspace tracking.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$
3. $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
4. $\bar{\mathbf{T}}_{\mathrm{MOja}}(i) = \mathbf{W}(i-1) - \mathbf{p}(i)\bar{\mathbf{y}}^H(i)$
5. $\mathbf{W}(i) = \bar{\mathbf{T}}_{\mathrm{MOja}}(i)$
```

Table 5.2: MOja with diagonal matrix step-size for noise subspace tracking.

MOja with diagonal matrix step-size suffers from instability as the original MOja. To improve its stability, we propose two different implementations, namely DMSS based on direct orthonormalization and DMSS based on separate orthogonalization and normalization.

5.2.1 MOja-DMSS by Direct Orthonormalization

The subspace matrix $\mathbf{W}(i)$ resulting from MOja with DMSS is not orthonormal. To orthonormalize it, we first have

$$\bar{\mathbf{T}}^H_{\text{MOja}}(i)\bar{\mathbf{T}}_{\text{MOja}}(i) = \mathbf{I}_P + \|\mathbf{p}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i), \tag{5.1}$$

since $\mathbf{p}(i)$ is orthogonal to $\mathbf{W}(i-1)$, *i.e.* $\mathbf{W}^H(i-1)\mathbf{p}(i) = \mathbf{0}$, when we assume that $\mathbf{W}(i-1)$ is orthonormal. Now, using [4]

$$\left(\mathbf{I} + \mathbf{x}\mathbf{x}^H\right)^{-\frac{1}{2}} = \mathbf{I} + \left(\frac{1}{\sqrt{1+\|\mathbf{x}\|^2}} - 1\right)\frac{\mathbf{x}\mathbf{x}^H}{\|\mathbf{x}\|^2}, \tag{5.2}$$

we obtain the inverse square root of $\bar{\mathbf{T}}^H_{\text{MOja}}(i)\bar{\mathbf{T}}_{\text{MOja}}(i)$ as

$$\left(\bar{\mathbf{T}}^H_{\text{MOja}}(i)\bar{\mathbf{T}}_{\text{MOja}}(i)\right)^{-\frac{1}{2}} = \mathbf{I}_P + \bar{\tau}(i)\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i), \tag{5.3}$$

where

$$\bar{\tau}(i) = \frac{1}{\|\bar{\mathbf{y}}(i)\|^2}\left(\frac{1}{\sqrt{1+\|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2}} - 1\right). \tag{5.4}$$

To make $\mathbf{W}(i)$ orthonormal, we set

$$\mathbf{W}(i) = \bar{\mathbf{T}}_{\text{MOja}}(i)\left(\bar{\mathbf{T}}^H_{\text{MOja}}(i)\bar{\mathbf{T}}_{\text{MOja}}(i)\right)^{-\frac{1}{2}} \tag{5.5a}$$

$$= \mathbf{W}(i-1) - \bar{\mathbf{p}}(i)\bar{\mathbf{y}}^H(i), \tag{5.5b}$$

where $\bar{\mathbf{p}}(i) = -\bar{\tau}(i)\mathbf{W}(i-1)\bar{\mathbf{y}}(i) + (1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)\mathbf{p}(i)$. Through simulations, we can see (5.5b) is not stable, as it accumulates rounding errors. On the other hand, direct implementation of (5.5a) is computationally very expensive though it is guaranteed to ensure stability.

To stabilize (5.5b), we propose an implementation of the algorithm based on Householder transform, which is well recognized as a stabilization tool in subspace estimation [11] [38]. In fact, the new implementation is a reformulation of (5.5b) in terms of Householder transform. As shown in Appendix B.1, we can prove that (5.5b) is mathematically equivalent to

$$\begin{aligned}\mathbf{u}(i) &= \frac{\bar{\mathbf{p}}(i)}{\|\bar{\mathbf{p}}(i)\|}\\ \mathbf{H}_N(i) &= \mathbf{I}_N - 2\mathbf{u}(i)\mathbf{u}^H(i)\\ \mathbf{W}(i) &= \mathbf{H}_N(i)\mathbf{W}(i-1).\end{aligned} \tag{5.6}$$

The steps of MOja with DMSS by direct orthogonalization are summarized in Table 5.3. If we compare Table 5.3 with Table 2.14, we can see their steps are exactly the same except for the step-size used. Since this implementation involves the use of numerically well-behaved Householder matrix $\mathbf{H}_N(i)$, the algorithm becomes very stable. The new implementation still preserves computational complexity at $O(NP)$.

5.2.2 MOja-DMSS by Separate Orthogonalization and Normalization

In DMSS by direct orthonormalization, orthonormality of $\mathbf{W}(i)$ is obtained by setting $\mathbf{W}(i) = \bar{\mathbf{T}}_{\text{MOja}}(i)\left(\bar{\mathbf{T}}^H_{\text{MOja}}(i)\bar{\mathbf{T}}_{\text{MOja}}(i)\right)^{-\frac{1}{2}}$, whose equivalent Householder implementation is $\mathbf{W}(i) = \mathbf{H}_N(i)\mathbf{W}(i-1)$. In this section, we use an alternative method by first orthogonalizing and then normalizing $\bar{\mathbf{T}}_{\text{MOja}}(i)$.

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$
3. $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
4. $\bar{\delta}(i) = \|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2$
5. $\rho(i) = \sqrt{1+\bar{\delta}(i)}$
6. $\bar{\tau}(i) = \frac{1}{\|\bar{\mathbf{y}}(i)\|^2}\left(\frac{1}{\bar{\rho}(i)} - 1\right)$
7. $\bar{\mathbf{p}}(i) = -\bar{\tau}(i)\mathbf{W}(i-1)\bar{\mathbf{y}}(i) + (1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)\mathbf{p}(i)$
8. $\mathbf{u}(i) = \frac{\bar{\mathbf{p}}(i)}{\|\bar{\mathbf{p}}(i)\|}$
9. $\mathbf{v}(i) = \mathbf{W}^H(i-1)\mathbf{u}(i)$
10. $\mathbf{W}(i) = \mathbf{W}(i-1) - 2\mathbf{u}(i)\mathbf{v}^H(i)$

Table 5.3: MOja with DMSS by direct orthonormalization.

Let us set $\mathbf{W}(i) = \bar{\mathbf{T}}_{\text{MOja}}(i)\mathbf{H}_P(i)\mathbf{D}(i)$, where $\mathbf{H}_P(i)$ is a $P \times P$ Householder matrix, and $\mathbf{D}(i)$ is a diagonal matrix. We first determine $\mathbf{H}_P(i)$ such that $\mathbf{Z}(i) = \bar{\mathbf{T}}_{\text{MOja}}(i)\mathbf{H}_P(i)$ is an orthogonal matrix, i.e. $\mathbf{X}(i) = \mathbf{Z}^H(i)\mathbf{Z}(i)$ is a diagonal matrix. To find $\mathbf{H}_P(i)$, we use Householder reflection properties [71, Chap 8] that can be assumed in the following lemma.

***Lemma* 5.1.** *Let $\mathbf{u}(i)$ and $\mathbf{v}(i)$ be $P \times 1$ non-zero complex-valued vectors having the same length. We define a vector in the direction of $\mathbf{u}(i) - \mathbf{v}(i)e^{j\alpha(i)}$ as*

$$\mathbf{a}(i) = \mathbf{u}(i) - \mathbf{v}(i)e^{j\alpha(i)}, \tag{5.7}$$

where $\alpha(i) = angle(\mathbf{v}^H\mathbf{u})$. A matrix $\mathbf{H}_P(i)$ of the form

$$\mathbf{H}_P(i) = \mathbf{I}_P - \frac{2\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2} \tag{5.8}$$

is called Householder reflector. Then, if $\mathbf{u}(i)$ *is pre-multiplied by* $\mathbf{H}_P(i)$*, we have*

$$\mathbf{v}(i)e^{j\alpha(i)} = \mathbf{H}_P(i)\mathbf{u}(i). \tag{5.9}$$

It is easy to verify that Householder reflector is unitary and Hermitian.

If $\mathbf{u}(i)$ *and* $\mathbf{v}(i)$ *are* $P \times 1$ *non-zero real-valued vectors having the same length, we define* $\mathbf{a}(i)$ *as*

$$\mathbf{a}(i) = \mathbf{u}(i) - \mathbf{v}(i). \tag{5.10}$$

A matrix $\mathbf{H}_P(i)$ *of the form*

$$\mathbf{H}_P(i) = \mathbf{I}_P - \frac{2\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2} \tag{5.11}$$

is called Householder reflector. Then, if $\mathbf{u}(i)$ *is pre-multiplied by* $\mathbf{H}_P(i)$*, we have*

$$\mathbf{v}(i) = \mathbf{H}_P(i)\mathbf{u}(i). \tag{5.12}$$

The proof of Lemma 5.1 for both complex-valued data and real-valued data is given in Appendix B.2. In this chapter, we are analyzing complex-valued data. So setting $\mathbf{a}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}$, where $\mathbf{e}_1 = [1, 0, \cdots, 0]^H$ is a $P \times 1$ vector, we have

$$\begin{aligned}
\mathbf{X}(i) &= \mathbf{Z}^H(i)\mathbf{Z}(i) \\
&= \mathbf{H}_P^H(i)\bar{\mathbf{T}}_{\mathrm{MOja}}^H(i)\bar{\mathbf{T}}_{\mathrm{MOja}}(i)\mathbf{H}_P(i) \\
&= \mathbf{H}_P^H(i)\left[\mathbf{I}_P + \|\mathbf{p}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)\right]\mathbf{H}_P(i)
\end{aligned}$$

$$= \mathbf{I}_P + \|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2\mathbf{e}_1\mathbf{e}_1^H. \tag{5.13}$$

As we can see, $\mathbf{X}(i)$ is a diagonal matrix. Finally, we normalize $\mathbf{Z}(i)$ in order to have an orthonormal matrix. The steps of MOja with DMSS by separate orthogonalization and normalization are summarized in Table 5.4. If we compare Table 5.4 with Table 2.16, we can see their steps are exactly the same except for the step-size used.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$
3. $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$
4. $\bar{\mathbf{T}}_{\mathrm{MOja}}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\bar{\mathbf{y}}^H(i)$
5. $\bar{\mathbf{a}}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}$
6. $\mathbf{Z}(i) = \bar{\mathbf{T}}_{\mathrm{MOja}}(i) - \frac{2}{\|\bar{\mathbf{a}}(i)\|^2}\left[\mathbf{T}_{\mathrm{MOja}}(i)\bar{\mathbf{a}}(i)\right]\bar{\mathbf{a}}^H(i)$
7. $\mathbf{D}(i) = \left[\mathrm{diag}\left(\mathbf{Z}^H(i)\mathbf{Z}(i)\right)\right]^{-\frac{1}{2}}$
8. $\mathbf{W}(i) = \mathbf{Z}(i)\mathbf{D}(i)$
```

Table 5.4: MOja with DMSS by separate orthogonalization and normalization.

5.3 Diagonal Matrix Step-size Strategy (DMSS) for Yang and Kaveh's Algorithm

We recall YK algorithm [110] in Table 5.5. Using a diagonal step-size matrix $\mathbf{\Lambda}(i) = \mathrm{diag}[\beta_1(i) \cdots \beta_P(i)]$, where $\beta_k(i)$ is the step-size for the kth column of $\mathbf{W}(i)$, YK with DMSS can be expressed as in Table 5.6.

The algorithm for YK with DMSS given in Table 5.6 has computational com-

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\mathbf{T}_{\mathrm{YK}}(i) = \mathbf{W}(i-1) - \beta\mathbf{r}(i)\mathbf{y}^H(i)$
3. $\mathbf{W}(i) = $ Gram-Schmidt orthonormalization$(\mathbf{T}_{\mathrm{YK}}(i))$
```

Table 5.5: YK algorithm [110] for noise subspace estimation.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$
3. $\bar{\mathbf{T}}_{\mathrm{YK}}(i) = \mathbf{W}(i-1) - \mathbf{r}(i)\bar{\mathbf{y}}^H(i)$
4. $\mathbf{W}(i) = $ Gram-Schmidt orthonormalization$(\bar{\mathbf{T}}_{\mathrm{YK}}(i))$
```

Table 5.6: YK with DMSS.

plexity $O(NP^2)$, which is not appropriate for real-time implementation. To reduce its complexity while maintaining stability, we propose three stable implementations of YK with DMSS. The implementation methods we use for YK with DMSS are different from the ones we used for MOja with DMSS. Recall that in MOja with DMSS, we had to orthonormalize $\bar{\mathbf{T}}_{\mathrm{MOja}}(i)$ where

$$\bar{\mathbf{T}}_{\mathrm{MOja}}^H(i)\bar{\mathbf{T}}_{\mathrm{MOja}}(i) = \mathbf{I}_P \underbrace{+\|\mathbf{p}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)}_{\text{rank 1}}, \tag{5.14}$$

which is an identity matrix plus a rank 1 matrix. Compared to this, in YK with DMSS, we need to orthonormalize $\bar{\mathbf{T}}_{\mathrm{YK}}(i)$ where

$$\bar{\mathbf{T}}_{\mathrm{YK}}^H(i)\bar{\mathbf{T}}_{\mathrm{YK}}(i) = \mathbf{I}_P \underbrace{-\mathbf{y}(i)\bar{\mathbf{y}}^H(i) - \bar{\mathbf{y}}(i)\mathbf{y}^H(i) + \|\mathbf{r}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)}_{\text{rank 2}}, \tag{5.15}$$

which is an identity matrix plus a rank 2 matrix. It is more difficult to orthonormalize using (5.15) than (5.14). We propose three different implementations for YK with DMSS using Givens rotation, by direct orthonormalization, and eigen decomposition.

5.3.1 YK-DMSS by Givens Rotation

Using Lemma 5.1 and setting $\mathbf{a}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}$, where $\mathbf{e}_1 = [1, 0, \cdots, 0]^H$ is a $P \times 1$ vector, we have

$$\begin{aligned}
\mathbf{H}_P(i) &= \mathbf{I}_P - 2\frac{\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2} \\
\mathbf{Z}(i) &= \bar{\mathbf{T}}_{\mathrm{YK}}(i)\mathbf{H}_P(i) \\
\mathbf{X}(i) &= \mathbf{Z}^H(i)\mathbf{Z}(i) \\
&= \mathbf{H}_P^H(i)\left[\mathbf{I}_P - \mathbf{y}(i)\bar{\mathbf{y}}^H(i) - \bar{\mathbf{y}}(i)\mathbf{y}^H(i) + \|\mathbf{r}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)\right]\mathbf{H}_P(i) \\
&= \mathbf{I}_P - \|\bar{\mathbf{y}}(i)\|\mathbf{d}(i)\mathbf{e}_1^H e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} - \|\bar{\mathbf{y}}(i)\|e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}\mathbf{e}_1\mathbf{d}^H(i) \\
&\quad + \|\mathbf{r}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2\mathbf{e}_1\mathbf{e}_1^H, \qquad (5.16)
\end{aligned}$$

where $\mathbf{d}(i) = \mathbf{H}_P(i)\mathbf{y}(i)$. Note that $\mathbf{X}(i)$ is a Hermitian matrix and all its elements are 0 except for those on the first column, first row and main diagonal.

To diagonalize $\mathbf{X}(i)$, we apply Jacobi methods [49, Chap 8]. The (l, k)th and (k, l)th elements of $\mathbf{X}(i)$ can be zeroed out by applying $\mathbf{G}_{k,l}(i)\mathbf{X}(i)\mathbf{G}_{l,k}(i)$, where

$\mathbf{G}_{k,l}(i)$ with $1 \leq l, k \leq P$ and $k \neq l$ is the Givens rotation matrix defined as

$$\mathbf{G}_{l,k}(i) = \begin{bmatrix} 1 \cdots & 0 & \cdots & 0 & \cdots 0 \\ \vdots \ddots & \vdots & & \vdots & \vdots \\ 0 \cdots & c_{l,k}(i) & \cdots & s_{l,k}(i) & \cdots 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 \cdots & -s_{l,k}(i) & \ldots & c_{l,k}(i) & \ldots 0 \\ \vdots & \vdots & & \vdots & \ddots \vdots \\ 0 \cdots & 0 & \cdots & 0 & \cdots 1 \end{bmatrix} \tag{5.17}$$

with $c^2_{l,k}(i) + s^2_{l,k}(i) = 1$, where

$$\begin{aligned} q_{l,k}(i) &= \frac{\mathbf{X}_{k,k}(i) - \mathbf{X}_{l,l}(i)}{2\mathbf{X}_{l,k}(i)} \\ l_{l,k}(i) &= \frac{\text{sign}(q_{l,k}(i))}{|q_{l,k}(i)| + \sqrt{q^2_{l,k}(i) + 1}} \\ c_{l,k}(i) &= \frac{1}{\sqrt{l^2_{l,k}(i) + 1}} \\ s_{l,k}(i) &= c_{l,k}(i) l_{l,k}(i). \end{aligned}$$

We can easily verify that $\mathbf{G}_{l,k}(i)$ is unitary and Hermitian. When $P > 2$, the computational complexity of Jacobi methods is high [49, Chap 8]. In this thesis, we will focus on the special case of $P = 2$. The study of diagonalization method with lower computational complexity for $P > 2$ will be our future focus.

When $P = 2$, the Givens rotation matrix can be written as

$$\mathbf{G}_{1,2}(i) = \begin{bmatrix} c_{1,2}(i) & s_{1,2}(i) \\ -s_{1,2}(i) & c_{1,2}(i) \end{bmatrix}. \tag{5.18}$$

Post-multiplying $\mathbf{Z}(i)$ with $\mathbf{G}_{1,2}(i)$, we obtain

$$\mathbf{R}(i) = \mathbf{Z}(i)\mathbf{G}_{1,2}(i), \tag{5.19}$$

where the product

$$\mathbf{R}^H(i)\mathbf{R}(i) = \mathbf{G}_{2,1}(i)\mathbf{X}(i)\mathbf{G}_{1,2}(i)$$

is a diagonal matrix which means that $\mathbf{R}(i)$ is orthogonal. Finally, $\mathbf{R}(i)$ is normalized, in order to have an orthonormal matrix $\mathbf{W}(i)$. The complete algorithm is presented in Table 5.7.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$
3. $\bar{\mathbf{T}}_{\mathrm{YK}}(i) = \mathbf{W}(i-1) - \mathbf{r}(i)\bar{\mathbf{y}}^H(i)$
4. $\mathbf{a}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}$
5. $\mathbf{Z}(i) = \bar{\mathbf{T}}_{\mathrm{YK}}(i) - 2\frac{\bar{\mathbf{T}}_{\mathrm{YK}}(i)\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}$
6. $\mathbf{X}(i) = \mathbf{Z}^H(i)\mathbf{Z}(i)$
7. $q_{1,2}(i) = \frac{\mathbf{X}_{2,2}(i)-\mathbf{X}_{1,1}(i)}{2\mathbf{X}_{1,2}(i)}$
   $l_{1,2}(i) = \frac{\mathrm{sign}(q_{1,2}(i))}{|q_{1,2}(i)|+\sqrt{q_{1,2}^2(i)+1}}$
   $c_{1,2}(i) = \frac{1}{\sqrt{l_{1,2}^2(i)+1}}$
   $s_{1,2}(i) = c_{1,2}(i)l_{1,2}(i)$
   where $c_{1,2}(i)$ and $s_{1,2}(i)$ generate $\mathbf{G}_{1,2}(i)$ as in (5.18).
8. $\mathbf{R}(i) = \mathbf{Z}(i)\mathbf{G}_{1,2}(i)$
9. $\mathbf{D}(i) = \left[\mathrm{diag}\left(\mathbf{R}^H(i)\mathbf{R}(i)\right)\right]^{-\frac{1}{2}}$
10. $\mathbf{W}(i) = \mathbf{R}(i)\mathbf{D}(i)$
```

Table 5.7: YK with DMSS by Givens rotation for the case $P = 2$.

5.3.2 YK-DMSS by Direct Orthonormalization

Using a similar method as MOja with DMSS by direct orthonormalization, we have

$$\mathbf{W}(i) = \bar{\mathbf{T}}_{\mathrm{YK}}(i)\left(\bar{\mathbf{T}}_{\mathrm{YK}}^{H}(i)\bar{\mathbf{T}}_{\mathrm{YK}}(i)\right)^{-\frac{1}{2}}, \tag{5.20}$$

where

$$\begin{aligned}\bar{\mathbf{T}}_{\mathrm{YK}}^{H}(i)\bar{\mathbf{T}}_{\mathrm{YK}}(i) &= \mathbf{I}_N + \mathbf{R}(i) && (5.21)\\ \mathbf{R}(i) &= -\mathbf{y}(i)\bar{\mathbf{y}}^{H}(i) - \bar{\mathbf{y}}(i)\mathbf{y}^{H}(i) + \|\mathbf{r}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^{H}(i). && (5.22)\end{aligned}$$

It is obvious that $\mathbf{R}(i)$ is a rank 2 Hermitian matrix. Fast computation of (5.20) is obtained using the following lemma [17, 19]. The proof of Lemma 5.2 can be found in [3].

***Lemma* 5.2.** *Let $\mathbf{R}$ be a d-rank Hermitian matrix with its range space spanned by column vectors $\mathbf{p}_1, \cdots, \mathbf{p}_d$. Then eigendecomposition of $\mathbf{R}$ is given by $\mathbf{R} = \bar{\mathbf{B}}\mathbf{D}\bar{\mathbf{B}}^H$, where $\mathbf{D} = diag(\kappa_1, \cdots, \kappa_d)$ and $\bar{\mathbf{B}}$ (orthonormal) are computed by*

$$\begin{aligned}\mathbf{P} &\triangleq [\mathbf{p}_1, \cdots, \mathbf{p}_d]\\ \mathbf{M} &= \left(\mathbf{P}^H\mathbf{P}\right)^{-1}\mathbf{P}^H\mathbf{R}\mathbf{P} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}\\ \mathbf{B} &= \mathbf{P}\mathbf{T} = [\mathbf{b}_1, \cdots, \mathbf{b}_d]\\ \mathbf{\Sigma} &= diag[\frac{1}{\|\mathbf{b}_1\|}, \cdots, \frac{1}{\|\mathbf{b}_d\|}]\\ \bar{\mathbf{B}} &= \mathbf{B}\mathbf{\Sigma} && (5.23)\end{aligned}$$

Let $\mathbf{N} = \mathbf{I} + \mathbf{R}$. *Then an inverse square root of* $\mathbf{N}$ *is given by*

$$\mathbf{N}^{-\frac{1}{2}} = \mathbf{I} + \bar{\mathbf{B}}\bar{\mathbf{D}}\bar{\mathbf{B}}^{H} \tag{5.24}$$

where

$$\bar{\mathbf{D}} = diag(\frac{1}{\sqrt{1+\kappa_1}} - 1, \cdots, \frac{1}{\sqrt{1+\kappa_d}} - 1). \tag{5.25}$$

Applying Lemma 5.2 on (5.22), we obtain the inverse square root of $\bar{\mathbf{T}}_{\mathrm{YK}}^{H}(i)\bar{\mathbf{T}}_{\mathrm{YK}}(i)$ through the following steps

$$\begin{aligned}
\mathbf{P}(i) &= [\mathbf{y}(i)\ \bar{\mathbf{y}}(i)] && (5.26)\\
\mathbf{M}(i) &= \left(\mathbf{P}^{H}(i)\mathbf{P}(i)\right)^{-1}\mathbf{P}^{H}(i)\mathbf{R}(i)\mathbf{P}(i) &&\\
\mathbf{M}(i) &= \mathbf{T}(i)\begin{bmatrix}\kappa_1(i) & 0\\ 0 & \kappa_2(i)\end{bmatrix}\mathbf{T}^{-1}(i) && (5.27)\\
\mathbf{B}(i) &= \mathbf{P}(i)\mathbf{T}(i) = [\mathbf{b}_1(i)\ \mathbf{b}_2(i)] &&\\
\mathbf{\Sigma}(i) &= \begin{bmatrix}1/\|\mathbf{b}_1(i)\| & 0\\ 0 & 1/\|\mathbf{b}_2(i)\|\end{bmatrix} &&\\
\bar{\mathbf{B}}(i) &= \mathbf{B}(i)\mathbf{\Sigma}(i) = [\bar{\mathbf{b}}_1(i)\ \bar{\mathbf{b}}_2(i)] &&\\
\left(\bar{\mathbf{T}}_{\mathrm{YK}}^{H}(i)\bar{\mathbf{T}}_{\mathrm{YK}}(i)\right)^{-\frac{1}{2}} &= \mathbf{I}_N + \bar{\mathbf{B}}(i)\bar{\mathbf{D}}(i)\bar{\mathbf{B}}^{H}(i), && (5.28)
\end{aligned}$$

where

$$\bar{\mathbf{D}}(i) = \begin{bmatrix}\frac{1}{\sqrt{1+\kappa_1(i)}} - 1 & 0\\ 0 & \frac{1}{\sqrt{1+\kappa_2(i)}} - 1\end{bmatrix}. \tag{5.29}$$

$\mathbf{P}(i)$ in (5.26) may not be full-rank always since it is possible thats $\bar{\mathbf{y}}(i)$ is linearly independent on $\mathbf{y}(i)$. To solve this problem, we need to check for every instant i that at least 2 elements on the main diagonal of $\mathbf{\Lambda}(i)$ are not equal or too close. For example, the two step-sizes with closest values are $\beta_m(i)$ and $\beta_n(i)$. If $|1 - \beta_m(i)/\beta_n(i)| < \varsigma$, the bigger one between $\beta_m(i)$ and $\beta_n(i)$ will be increased.

Since $\mathbf{M}(i)$ is a 2×2 Hermitian matrix (denote $\mathbf{M}(i)$ by $\begin{bmatrix} a(i) \, b(i) \\ c(i) \, a(i) \end{bmatrix}$), by solving

$$det(\kappa \mathbf{I}_2 - \mathbf{M}) = 0, \tag{5.30}$$

$\kappa_1(i)$ and $\kappa_2(i)$ in (5.27) can be easily obtained as

$$\kappa_1(i) = a(i) + \sqrt{b(i)c(i)} \text{ and } \kappa_2(i) = a(i) - \sqrt{b(i)c(i)}. \tag{5.31}$$

By solving

$$\begin{aligned} (\kappa_1(i)\mathbf{I}_2 - \mathbf{M}(i))\mathbf{t}_1(i) &= 0 \\ (\kappa_2(i)\mathbf{I}_2 - \mathbf{M}(i))\mathbf{t}_2(i) &= 0, \end{aligned} \tag{5.32}$$

$\mathbf{T}(i) = [\mathbf{t}_1(i) \ \mathbf{t}_2(i)]$ in (5.27) can be obtained as

$$\mathbf{T}(i) = \begin{bmatrix} 1 & 1 \\ \kappa_1(i) - a(i) & \kappa_2(i) - a(i) \end{bmatrix}. \tag{5.33}$$

By substituting (5.28) into (5.20), we obtain

$$\mathbf{W}(i) = \mathbf{W}(i-1) + \tau_1(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_1(i)\bar{\mathbf{b}}_1^H(i) + \tau_2(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_2(i)\bar{\mathbf{b}}_2^H(i) - \mathbf{r}(i)\bar{\mathbf{y}}^H(i) \\ -\tau_1(i)\mathbf{r}(i)\bar{\mathbf{y}}^H(i)\bar{\mathbf{b}}_1(i)\bar{\mathbf{b}}_1^H(i) - \tau_2(i)\mathbf{r}(i)\bar{\mathbf{y}}^H(i)\bar{\mathbf{b}}_2(i)\bar{\mathbf{b}}_2^H(i). \quad (5.34)$$

Since $\bar{\mathbf{B}}(i)$ is orthonormal as claimed by Lemma 5.2, we obtain

$$\bar{\mathbf{T}}(i) = \mathbf{T}(i)\mathbf{\Sigma}(i). \quad (5.35)$$

Hence, we can write

$$\bar{\mathbf{T}}^{-1}(i) = \bar{\mathbf{B}}^H(i)\mathbf{P}(i) = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}. \quad (5.36)$$

Using this in $\mathbf{P}(i) = \bar{\mathbf{B}}(i)\bar{\mathbf{T}}^{-1}(i) = [\mathbf{y}(i) \;\; \bar{\mathbf{y}}(i)]$, we get

$$\bar{\mathbf{y}}(i) = t_{12}\bar{\mathbf{b}}_1(i) + t_{22}\bar{\mathbf{b}}_2(i). \quad (5.37)$$

Finally, substituting (5.37) in (5.34), we obtain

$$\mathbf{W}(i) = \mathbf{W}(i-1) + \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{b}}_2^H(i) \quad (5.38)$$

where

$$\mathbf{d}(i) = \tau_1(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_1(i) - (\tau_1(i)+1)t_{12}\mathbf{r}(i) \\ \mathbf{h}(i) = \tau_2(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_2(i) - (\tau_2(i)+1)t_{22}\mathbf{r}(i). \quad (5.39)$$

Table 5.8 gives the steps of YK with DMSS by direct orthonormalization.

```
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\bar{\mathbf{y}}(i) = \boldsymbol{\Lambda}(i)\mathbf{y}(i)$
3. $\mathbf{R}(i) = -\mathbf{y}(i)\bar{\mathbf{y}}^H(i) - \bar{\mathbf{y}}(i)\mathbf{y}^H(i) + \|\mathbf{r}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)$
4. $\mathbf{P}(i) = [\mathbf{y}(i)\ \bar{\mathbf{y}}(i)]$
5. $\mathbf{M}(i) = \left(\mathbf{P}^H(i)\mathbf{P}(i)\right)^{-1}\mathbf{P}^H(i)\mathbf{R}(i)\mathbf{P}(i)$
6. Denote $\mathbf{M}(i)$ by $\begin{bmatrix} a(i) & b(i) \\ c(i) & a(i) \end{bmatrix}$
   $\kappa_1(i) = a(i) + \sqrt{b(i)c(i)}$    $\kappa_2(i) = a(i) - \sqrt{b(i)c(i)}$
   $\mathbf{T}(i) = \begin{bmatrix} 1 & 1 \\ \kappa_1(i) - a(i) & \kappa_2(i) - a(i) \end{bmatrix}$
7. $\mathbf{B}(i) = \mathbf{P}(i)\mathbf{T}(i) = [\mathbf{b}_1(i)\ \mathbf{b}_2(i)]$
8. $\boldsymbol{\Sigma}(i) = \begin{bmatrix} 1/\|\mathbf{b}_1(i)\| & 0 \\ 0 & 1/\|\mathbf{b}_2(i)\| \end{bmatrix}$
9. $\bar{\mathbf{B}}(i) = \mathbf{B}(i)\boldsymbol{\Sigma}(i) = [\bar{\mathbf{b}}_1(i)\ \bar{\mathbf{b}}_2(i)]$
10. $\bar{\mathbf{T}}(i) = \mathbf{T}(i)\boldsymbol{\Sigma}(i)$
11. $\tau_1(i) = \frac{1}{\sqrt{1+\kappa_1(i)}} - 1$    $\tau_2(i) = \frac{1}{\sqrt{1+\kappa_2(i)}} - 1$
12. $\bar{\mathbf{T}}^{-1}(i) = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$
13. $\mathbf{d}(i) = \tau_1(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_1(i) - (\tau_1(i)+1)t_{12}\mathbf{r}(i)$
14. $\mathbf{h}(i) = \tau_2(i)\mathbf{W}(i-1)\bar{\mathbf{b}}_2(i) - (\tau_2(i)+1)t_{22}\mathbf{r}(i)$
15. $\mathbf{W}(i) = \mathbf{W}(i-1) + \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{b}}_2^H(i)$
```

Table 5.8: YK with DMSS by direct orthonormalization.

Unfortunately, (5.38) still suffers from numerical instability. To limit the effect of rounding error, numerically well behaved Householder matrix is employed based on Lemma 5.3 [17, 19]. The proof of Lemma 5.3 can be found in [18].

***Lemma* 5.3.** *Step 15 in Table 5.8 can be reformulated as*

$$\mathbf{W}(i) = \mathbf{H}_1(i)\mathbf{H}_2(i)\mathbf{W}(i-1) \tag{5.40}$$

where $\mathbf{H}_1(i)$ *and* $\mathbf{H}_2(i)$ *are the Householder matrices given by*

$$\begin{aligned}\mathbf{H}_1(i) &= \mathbf{I}_N - 2\frac{\mathbf{a}_1(i)\mathbf{a}_1^H(i)}{\|\mathbf{a}_1(i)\|^2}\\ \mathbf{H}_2(i) &= \mathbf{I}_N - 2\frac{\mathbf{a}_2(i)\mathbf{a}_2^H(i)}{\|\mathbf{a}_2(i)\|^2}\end{aligned} \tag{5.41}$$

where $\mathbf{a}_1(i)$ *and* $\mathbf{a}_2(i)$ *are the eigenvectors of* $\mathbf{W}(i) - \mathbf{W}(i-1)$ *and* $\mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1)$*, respectively.*

Following the above lemma, $\mathbf{a}_1(i)$ is calculated as the principal left singular eigenvector of

$$\bar{\mathbf{Q}}(i) = \mathbf{W}(i) - \mathbf{W}(i-1) = \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{b}}_2^H(i) \tag{5.42}$$

Equivalently, $\mathbf{a}_1(i)$ can be seen as the eigenvector of the rank 2 Hermitian matrix

$$\mathbf{R}_Q(i) = \bar{\mathbf{Q}}(i)\bar{\mathbf{Q}}^H(i) = \mathbf{d}(i)\mathbf{d}(i)^H + \mathbf{h}(i)\mathbf{h}(i)^H. \tag{5.43}$$

According to Lemma 5.2, since $\mathbf{R}_Q(i)$ is a rank 2 matrix, the eigen matrix $\bar{\mathbf{B}}_Q(i)$ of $\mathbf{R}_Q(i)$ can be obtained as

$$\begin{aligned}\mathbf{Q}(i) &= [\mathbf{d}(i) \quad \mathbf{h}(i)]\\ \mathbf{\Pi}(i) &= \begin{bmatrix}\|\mathbf{d}(i)\|^2 & \mathbf{h}^H(i)\mathbf{d}(i)\\ \mathbf{d}^H(i)\mathbf{h}(i) & \|\mathbf{h}(i)\|^2\end{bmatrix} = \begin{bmatrix}\pi_{11}(i) & \pi_{12}(i)\\ \pi_{21}(i) & \pi_{22}(i)\end{bmatrix}\\ \mathbf{Q}_2(i) &= \mathbf{R}_Q(i)\mathbf{Q}(i)\\ &= [\pi_{11}(i)\mathbf{d}(i) + \pi_{12}(i)\mathbf{h}(i) \quad \pi_{21}(i)\mathbf{d}(i) + \pi_{22}(i)\mathbf{h}(i)]\\ &= \mathbf{\Pi}(i)\mathbf{Q}(i)\end{aligned}$$

$$\begin{aligned}
\mathbf{M}_Q(i) &= \left(\mathbf{Q}^H(i)\mathbf{Q}(i)\right)^{-1}\mathbf{Q}^H(i)\mathbf{R}_Q(i)\mathbf{Q}(i) \\
&= \left(\mathbf{Q}^H(i)\mathbf{Q}(i)\right)^{-1}\mathbf{Q}^H(i)\mathbf{Q}_2(i) \\
&= \mathbf{T}_Q(i)\begin{bmatrix}\kappa_{Q_1}(i) & 0 \\ 0 & \kappa_{Q_2}(i)\end{bmatrix}\mathbf{T}_Q^{-1}(i) \qquad (5.44) \\
\mathbf{B}_Q(i) &= \mathbf{Q}(i)\mathbf{T}_Q(i) = [\mathbf{b}_{Q1}(i) \;\; \mathbf{b}_{Q2}(i)] \\
\mathbf{\Sigma}_Q(i) &= \begin{bmatrix}\frac{1}{\|\mathbf{b}_{Q1}(i)\|} & 0 \\ 0 & \frac{1}{\|\mathbf{b}_{Q2}(i)\|}\end{bmatrix} \\
\bar{\mathbf{B}}_Q(i) &= \mathbf{B}_Q(i)\mathbf{\Sigma}_Q(i) = [\bar{\mathbf{b}}_{Q1}(i) \;\; \bar{\mathbf{b}}_{Q2}(i)], \qquad (5.45)
\end{aligned}$$

where step (5.44) can be implemented using method proposed in (5.33). Note that both the eigenvectors of $\mathbf{R}_Q(i)$ can do, so may we choose either $\mathbf{a}_1(i) = \bar{\mathbf{b}}_{Q1}(i)$ or $\mathbf{a}_1 = \bar{\mathbf{b}}_{Q2}(i)$.

Next, we observe from (5.40) and (5.41) that

$$\mathbf{Z}(i) = \mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1) = -2\mathbf{a}_2(i)\mathbf{a}_2^H(i)\mathbf{W}(i-1) \qquad (5.46)$$

is a rank 1 matrix. All column vectors of $\mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1)$ are equal to $\mathbf{a}_2(i)$ up to scalar constant. Hence, it is sufficient to set $\mathbf{a}_2(i)$ as the first column of $\mathbf{Z}(i)$. Using (5.38) in (5.46), $\mathbf{Z}(i)$ can be obtained as

$$\begin{aligned}
\mathbf{Z}(i) &= \mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1) \\
&= \left(\mathbf{I}_N - 2\mathbf{a}_1(i)\mathbf{a}_1^H(i)\right)\left(\mathbf{W}(i-1) + \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{b}}_2^H(i)\right) - \mathbf{W}(i-1) \\
&= \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{d}}_2^H(i) - 2\mathbf{a}_1(i)\mathbf{a}_1^H(i)\left(\mathbf{W}(i-1) + \mathbf{d}(i)\bar{\mathbf{b}}_1^H(i) + \mathbf{h}(i)\bar{\mathbf{d}}_2^H(i)\right)
\end{aligned} \qquad (5.47)$$

Therefore, $\mathbf{a}_2(i)$ can be obtained as

$$\mathbf{a}_2(i) = \bar{\mathbf{b}}^*_{1,1}(i)\mathbf{d}(i) + \bar{\mathbf{b}}^*_{2,1}(i)\mathbf{h}(i) - 2\mathbf{a}_1^H(i)[(\bar{\mathbf{b}}^*_{1,1}(i) + \bar{\mathbf{b}}^*_{2,1}(i))\mathbf{d}(i) + \mathbf{w}_1(i-1)]\mathbf{a}_1(i), \tag{5.48}$$

where $\bar{\mathbf{b}}_{1,1}(i)$ and $\bar{\mathbf{b}}_{2,1}(i)$ are the first elements of $\bar{\mathbf{b}}_1(i)$ and $\bar{\mathbf{b}}_2(i)$, respectively, and $\mathbf{w}_1(i-1)$ is the first column of $\mathbf{W}(i-1)$. The proposed method is called YK with DMSS by direct orthonormalization. We summarize the steps in Table 5.9.

```
Following Step 14 of Table 5.8:
1. $\mathbf{Q}(i) = [\mathbf{d}(i)\ \mathbf{h}(i)]$
2. $\mathbf{\Pi}(i) = \begin{bmatrix} \|\mathbf{d}(i)\|^2 & \mathbf{h}^H(i)\mathbf{d}(i) \\ \mathbf{d}^H(i)\mathbf{h}(i) & \|\mathbf{h}(i)\|^2 \end{bmatrix} = \begin{bmatrix} \pi_{11}(i) & \pi_{12}(i) \\ \pi_{21}(i) & \pi_{22}(i) \end{bmatrix}$
3. $\mathbf{Q}_2(i) = [\pi_{11}(i)\mathbf{d}(i) + \pi_{12}(i)\mathbf{h}(i) \quad \pi_{21}(i)\mathbf{d}(i) + \pi_{22}(i)\mathbf{h}(i)]$
4. $\mathbf{M}_Q(i) = \left(\mathbf{Q}^H(i)\mathbf{Q}(i)\right)^{-1}\mathbf{Q}^H(i)\mathbf{Q}_2(i)$
5. Denote $\mathbf{M}_Q(i)$ by $\begin{bmatrix} a(i) & b(i) \\ c(i) & a(i) \end{bmatrix}$
   $\kappa_{Q1}(i) = a(i) + \sqrt{b(i)c(i)} \qquad \kappa_{Q2}(i) = a(i) - \sqrt{b(i)c(i)}$
   $\mathbf{T}_Q(i) = \begin{bmatrix} 1 & 1 \\ \kappa_{Q1}(i) - a(i) & \kappa_{Q2}(i) - a(i) \end{bmatrix}$
6. $\mathbf{B}_Q(i) = \mathbf{Q}(i)\mathbf{T}_Q(i) = [\mathbf{b}_{Q1}(i)\ \mathbf{b}_{Q2}(i)]$
7. $\mathbf{a}_1(i) = \frac{\mathbf{b}_{Q1}(i)}{\|\mathbf{b}_{Q1}(i)\|}$
8. $\mathbf{a}_2(i) = \bar{\mathbf{b}}^*_{1,1}(i)\mathbf{d}(i) + \bar{\mathbf{b}}^*_{2,1}(i)\mathbf{h}(i) - 2\mathbf{a}_1^H(i)[(\bar{\mathbf{b}}^*_{1,1}(1) + \bar{\mathbf{b}}^*_{2,1}(1))\mathbf{d}(i)$
                $+ \mathbf{w}_1(i-1)]\mathbf{a}_1(i)$
9. $\mathbf{W}(i) = \left(\mathbf{I}_N - 2\mathbf{a}_1(i)\mathbf{a}_1^H(i)\right)\left(\mathbf{I}_N - 2\mathbf{a}_2(i)\mathbf{a}_2^H(i)\right)\mathbf{W}(i-1)$
```

Table 5.9: Stable YK with DMSS by direct orthonormalization with Householder implementation.

5.3.3 DMSS by Eigendecomposition

Applying Lemma 5.1, and setting $\mathbf{a}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}$, where $\mathbf{e}_1 = [1, 0, \cdots, 0]^H$ is a $P \times 1$ vector, we have

$$\begin{aligned}
\mathbf{H}_P(i) &= \mathbf{I}_P - 2\frac{\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2} \\
\mathbf{Z}(i) &= \bar{\mathbf{T}}_{\mathrm{YK}}(i)\mathbf{H}_P(i) \\
\mathbf{X}(i) &= \mathbf{Z}^H(i)\mathbf{Z}(i) \\
&= \mathbf{H}_P^H(i)\left[\mathbf{I}_P - \mathbf{y}(i)\bar{\mathbf{y}}^H(i) - \bar{\mathbf{y}}(i)\mathbf{y}^H(i) + \|\mathbf{r}(i)\|^2\bar{\mathbf{y}}(i)\bar{\mathbf{y}}^H(i)\right]\mathbf{H}_P(i) \\
&= \mathbf{I}_P - \|\bar{\mathbf{y}}(i)\|\mathbf{d}(i)\mathbf{e}_1^H e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}\mathbf{d}^H(i) \\
&\quad + \|\mathbf{r}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2\mathbf{e}_1\mathbf{e}_1^H, \qquad (5.49)
\end{aligned}$$

where $\mathbf{d}(i) = \mathbf{H}_P(i)\mathbf{y}(i)$. Note that $\mathbf{X}(i)$ is a Hermitian matrix and all its elements are 0 except for those on the first column, first row and main diagonal.

To diagonalize $\mathbf{X}(i)$, an unitary matrix $\mathbf{Q}(i)$ which can make $\mathbf{Q}^H(i)\mathbf{X}(i)\mathbf{Q}(i)$ diagonal is required. To obtain $\mathbf{Q}(i)$, we first solve for the eigenvalues of $\mathbf{X}(i)$ by

$$\det\left(\lambda\mathbf{I}_P - \mathbf{X}(i)\right) = 0. \qquad (5.50)$$

It is equivalent to solving for λ in

$$\left[(\lambda-1)^2 + b(i)(\lambda-1) + c(i)\right](\lambda-1)^{P-2} = 0, \qquad (5.51)$$

where

$$\begin{aligned} b(i) &= 2\|\bar{\mathbf{y}}(i)\|\mathrm{Re}(d_1(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}) - \|\mathbf{r}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2 \\ c(i) &= -\|\bar{\mathbf{y}}(i)\|^2\sum_{k=2}^{P}|d_k(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))}|^2, \end{aligned}$$

and $d_k(i)$ is the kth element of $\mathbf{d}(i)$. The eigenvalues are

$$\begin{aligned} \lambda_1(i) &= 1 + \frac{-b(i) + \sqrt{b^2(i) - 4c(i)}}{2} \\ \lambda_2(i) &= 1 + \frac{-b(i) - \sqrt{b^2(i) - 4c(i)}}{2} \\ \lambda_3(i) &= 1, \end{aligned}$$

where λ_3 has multiplicity $P-2$. Solving $(\lambda_i(i)\mathbf{I}_P - \mathbf{X}(i))\mathbf{q}_i(i) = \mathbf{0}$, where $i = 1,\ 2$, we obtain the eigenvectors of $\mathbf{X}(i)$ corresponding to λ_1 and λ_2 as

$$\begin{aligned} \mathbf{q}_1(i) &= \begin{bmatrix} \lambda_1(i) - 1 \\ -\|\bar{\mathbf{y}}(i)\|d_2(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} \\ \vdots \\ -\|\bar{\mathbf{y}}(i)\|d_P(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} \end{bmatrix} \\ \mathbf{q}_2(i) &= \begin{bmatrix} \lambda_2(i) - 1 \\ -\|\bar{\mathbf{y}}(i)\|d_2(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} \\ \vdots \\ -\|\bar{\mathbf{y}}(i)\|d_P(i)e^{-jangle(\mathbf{e}1^H\bar{\mathbf{y}}(i))} \end{bmatrix}, \end{aligned}$$

respectively. Normalizing $\mathbf{q}_1(i)$ and $\mathbf{q}_2(i)$, we obtain

$$\bar{\mathbf{q}}_1(i) = \frac{\mathbf{q}_1(i)}{\|\mathbf{q}_1(i)\|} \quad \text{and} \quad \bar{\mathbf{q}}_2(i) = \frac{\mathbf{q}_2(i)}{\|\mathbf{q}_2(i)\|}.$$

Solving $(\lambda_3(i)\mathbf{I}_P - \mathbf{X}(i))\mathbf{q}_3(i) = \mathbf{0}$, the eigenvectors of $\mathbf{X}(i)$ corresponding to λ_3 satisfies

$$q_{3,1}(i) = 0 \quad \text{and} \quad \sum_{k=2}^{P} q_{3,k}(i)d_k^*(i) = 0, \tag{5.52}$$

where $q_{3,k}(i)$ is the kth element of $\mathbf{q}_3(i)$. Define $\bar{\mathbf{q}}_3(i)$ as $\bar{\mathbf{q}}_3(i) = \frac{\mathbf{q}_3(i)}{\|\mathbf{q}_3(i)\|}$. Therefore, $\mathbf{Q}(i)$ can be obtained as

$$\mathbf{Q}(i) = \left[\bar{\mathbf{q}}_1(i), \bar{\mathbf{q}}_2(i), \bar{\mathbf{Q}}_3(i)\right], \tag{5.53}$$

where $\bar{\mathbf{Q}}_3(i)$ is $P \times (P-2)$ matrix spanned by $\bar{\mathbf{q}}_3(i)$. As we can see, the product

$$\mathbf{Q}^H(i)\mathbf{X}(i)\mathbf{Q}(i) = \begin{bmatrix} \lambda_1(i) & & & & \\ & \lambda_2(i) & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

is diagonal. Finally, $\mathbf{R}(i) = \mathbf{Z}(i)\mathbf{Q}(i)$ is normalized (since $\mathbf{R}^H(i)\mathbf{R}(i)$ is a diagonal rather than an identity matrix), in order to have an orthonormal matrix $\mathbf{W}(i)$. The complete algorithm is presented in Table 5.10.

1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$

2. $\bar{\mathbf{y}}(i) = \mathbf{\Lambda}(i)\mathbf{y}(i)$

3. $\bar{\mathbf{T}}_{\mathrm{YK}}(i) = \mathbf{W}(i-1) - \mathbf{r}(i)\bar{\mathbf{y}}^H(i)$

4. $\mathbf{a}(i) = \bar{\mathbf{y}}(i) - \|\bar{\mathbf{y}}(i)\|\mathbf{e}_1 e^{jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))}$

5. $\mathbf{H}_P(i) = \mathbf{I}_P - \frac{2\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}$

6. $\mathbf{Z}(i) = \bar{\mathbf{T}}_{\mathrm{YK}}(i)\mathbf{H}_P(i)$

7. $\mathbf{d}(i) = \mathbf{H}_P(i)\mathbf{y}(i)$

$b(i) = 2\|\bar{\mathbf{y}}(i)\|\mathrm{Re}(d_1(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))}) - \|\mathbf{r}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2$

$c(i) = -\|\bar{\mathbf{y}}(i)\|^2 \sum_{k=2}^{P} |d_k(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))}|^2$

$\lambda_1(i) = 1 + \frac{-b(i)+\sqrt{b^2(i)-4c(i)}}{2}$

$\lambda_2(i) = 1 + \frac{-b(i)-\sqrt{b^2(i)-4c(i)}}{2}$

$$\mathbf{q}_1(i) = \begin{bmatrix} \lambda_1(i) - 1 \\ -\|\bar{\mathbf{y}}(i)\|d_2(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))} \\ \vdots \\ -\|\bar{\mathbf{y}}(i)\|d_P(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))} \end{bmatrix}$$

$$\mathbf{q}_2(i) = \begin{bmatrix} \lambda_2(i) - 1 \\ -\|\bar{\mathbf{y}}(i)\|d_2(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))} \\ \vdots \\ -\|\bar{\mathbf{y}}(i)\|d_P(i)e^{-jangle(\mathbf{e1}^H\bar{\mathbf{y}}(i))} \end{bmatrix}$$

$\bar{\mathbf{q}}_1(i) = \frac{\mathbf{q}_1(i)}{\|\mathbf{q}_1(i)\|}$ and $\bar{\mathbf{q}}_2(i) = \frac{\mathbf{q}_2(i)}{\|\mathbf{q}_2(i)\|}$

$q_{3,1}(i) = 0$ and $\sum_{k=2}^{P} q_{3,k}(i)d_k^*(i) = 0$

$\bar{\mathbf{q}}_3(i) = \frac{\mathbf{q}_3(i)}{\|\mathbf{q}_3(i)\|}$ $\bar{\mathbf{Q}}_3(i)$ is spanned by $\bar{\mathbf{q}}_3(i)$

$\mathbf{Q}(i) = [\bar{\mathbf{q}}_1(i), \bar{\mathbf{q}}_2(i), \bar{\mathbf{Q}}_3(i)]$

8. $\mathbf{R}(i) = \mathbf{Z}(i)\mathbf{Q}(i)$

9. $\mathbf{D}(i) = \left[\mathrm{diag}\left(\mathbf{R}^H(i)\mathbf{R}(i)\right)\right]^{-\frac{1}{2}}$

10. $\mathbf{W}(i) = \mathbf{R}(i)\mathbf{D}(i)$

Table 5.10: YK with DMSS by eigendecomposition.

5.4 Estimated Optimal Diagonal-matrix Step-size

In Sections 5.2 and 5.3, we developed the DMSS for MOja and YK with different implementations. However, how to choose the individual step-size β_k was not addressed. In this section, we address by deriving derive an optimal value for the diagonal step-size matrix.

The cost function of YK is

$$J(i) = E\left[\mathrm{Tr}\left(\mathbf{W}(i)^H \mathbf{C}\mathbf{W}(i)\right)\right]. \tag{5.54}$$

Omitting the orthonormalization step, we can approximate the cost function by $E\left[\mathrm{Tr}\left(\bar{\mathbf{T}}_{\mathrm{YK}}(i)^H \mathbf{C}\bar{\mathbf{T}}_{\mathrm{YK}}(i)\right)\right]$, where $\bar{\mathbf{T}}_{\mathrm{YK}}(i) = \mathbf{W}(i-1) - \mathbf{r}(i)\mathbf{y}^H(i)\mathbf{\Lambda}(i)$. Plugging $\bar{\mathbf{T}}_{\mathrm{YK}}(i)$ into (5.54), we obtain

$$\begin{aligned} J(i) = {} & E\left[\mathrm{Tr}\left(\mathbf{W}^H(i-1)\mathbf{C}\mathbf{W}(i-1)\right)\right] - 2E\left[\mathrm{Re}\left(\sum_{k=1}^{P}\beta_k(i)y_k(i)\mathbf{r}^H(i)\mathbf{C}\mathbf{w}_k(i-1)\right)\right] \\ & + E\left[\sum_{k=1}^{P}\beta_k^2(i)|y_k(i)|^2\mathbf{r}^H(i)\mathbf{C}\mathbf{r}(i)\right], \end{aligned} \tag{5.55}$$

where $y_k(i) = \mathbf{w}_k^H(i-1)\mathbf{r}(i)$ is the kth element of $\mathbf{y}(i)$. To compute the optimal step-size β_k, we need to solve the following system of equations:

$$\frac{\partial J(i)}{\partial \beta_k(i)} = 0 \qquad \text{for } \; k = 1, \; \cdots, \; P. \tag{5.56}$$

The derivative of $J(i)$ with respect to $\beta_k(i)$ gives

$$\begin{aligned} \frac{\partial J(i)}{\partial \beta_k(i)} = {} & -2E\left[\mathrm{Re}\left(y_k(i)\mathbf{r}^H(i)\mathbf{C}\mathbf{w}_k(i-1)\right)\right] \\ & +2E\left[\beta_k(i)|y_k(i)|^2\mathbf{r}^H(i)\mathbf{C}\mathbf{r}(i)\right]. \end{aligned} \tag{5.57}$$

Setting the derivative in (5.57) to zero leads to

$$\beta_{o,k}(i) = \frac{E\left[\mathrm{Re}\left(y_k(i)\mathbf{r}^H(i)\mathbf{C}\mathbf{w}_k(i-1)\right)\right]}{E\left[|y_k(i)|^2\mathbf{r}^H(i)\mathbf{C}\mathbf{r}(i)\right]}. \tag{5.58}$$

Assuming $\mathbf{w}_k(i-1)$ and $\mathbf{r}(i)$ to be independent, we get the practical realization of the optimal step-size as

$$\hat{\beta}_{o,k}(i) = \gamma\frac{h_k(i)}{b_k(i)+\varepsilon}, \tag{5.59}$$

with

$$\begin{aligned} h_k(i) &= \alpha h_k(i-1) + (1-\alpha)|\mathrm{Re}(y_k(i)\theta(i)\phi_k(i))| \\ b_k(i) &= \alpha b_k(i-1) + (1-\alpha)|y_k(i)|^2|\theta(i)|^2 \\ \theta(i) &= \|\mathbf{r}^H(i)\|^2 \\ \phi_k(i) &= \mathbf{r}^H(i)\mathbf{w}_k(i-1), \end{aligned} \tag{5.60}$$

where α $(0 < \alpha < 1)$ is a forgetting factor, and γ and ϵ are two positive constants $(0 < \gamma < 1)$ which help improve the numerical stability of the algorithm.

The optimal step-size for MOja with DMSS is obtained by minimizing the mean square error (MSE)

$$\begin{aligned} J(i) &= E\|\mathbf{r}(i) - \mathbf{W}(i)\mathbf{W}(i)\mathbf{r}(i)\|^2 \\ &= \mathrm{Tr}(\mathbf{C}) - 2\mathrm{Tr}(\mathbf{W}(i)^H\mathbf{C}\mathbf{W}(i)) + \mathrm{Tr}(\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i)\mathbf{W}^H(i)\mathbf{W}(i)). \end{aligned} \tag{5.61}$$

When $\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{I}_P$, the MSE equation in (5.61) is equal to

$$J(i) = \mathrm{Tr}(\mathbf{C}) - \mathrm{Tr}(\mathbf{W}(i)^H \mathbf{C}\mathbf{W}(i)). \tag{5.62}$$

Omitting the orthonormalization step, we can approximate $\mathbf{W}(i)$ in (5.62) by $\bar{\mathbf{T}}_{\mathrm{MOja}} = \mathbf{W}(i-1) - \mathbf{p}(i)\bar{\mathbf{y}}^H(i)$, where $\mathbf{p}(i) = \mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)$. Following similar steps as the optimal step size of YK with DMSS, we get the optimal step-size for MOja with DMSS as

$$\hat{\beta}_{o,k}(i) = \gamma \frac{h_k(i)}{b_k(i) + \varepsilon}, \tag{5.63}$$

with

$$\begin{aligned} h_k(i) &= \alpha h_k(i-1) + (1-\alpha)|\mathrm{Re}(p_k(i)\theta(i)\phi_k(i))| \\ b_k(i) &= \alpha b_k(i-1) + (1-\alpha)|p_k(i)|^2|\theta(i)|^2 \\ \theta(i) &= \|\mathbf{r}^H(i)\|^2 \\ \phi_k(i) &= \mathbf{r}^H(i)\mathbf{w}_k(i-1), \end{aligned} \tag{5.64}$$

where α $(0 < \alpha < 1)$ is a forgetting factor, and γ and ϵ are two positive constants $(0 < \gamma < 1)$.

5.5 Simulation Results and Discussion

In our simulations, we set the covariance matrix to

$$\mathbf{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix},$$

where $N = 4$ and $P = 2$. The generation of the observed data $\mathbf{r}(i)$ is generated as mentioned in Section 2.5.1.

In this section, we perform several simulations to demonstrate the effect of the proposed DMSS with different implementations on performance. We start by comparing the performance of our proposed MOja with DMSS by direct orthonormalization and by separate orthogonalization and normalization with MOja [105], NOOja [8], and FOOja [21]. Then, we continue with the comparison of YK with DMSS by Givens rotation, by direct orthonormalization and by eigendecomposition with YK, HFRANS and FDPM.

5.5.1 Simulation Results and Discussion for MOja with DMSS

In Figures 5.1 and 5.2, we compare the performance of MOja with DMSS by direct orthonormalization and NOOja, since the former can be considered as NOOja with DMSS. In NOOja, $\beta(i) = \frac{0.017}{\|\mathbf{r}(i)\|^2 - \|\mathbf{y}(i)\|^2 + 0.04}$. In MOja with DMSS by direct orthonormalization, $\alpha = 0.998$, $\gamma = 0.01$, and $\varepsilon = 0.01$. As we can see MOja with DMSS by direct orthonormalization has faster convergence, smaller estimation error and similar orthogonality error compared to NOOja.

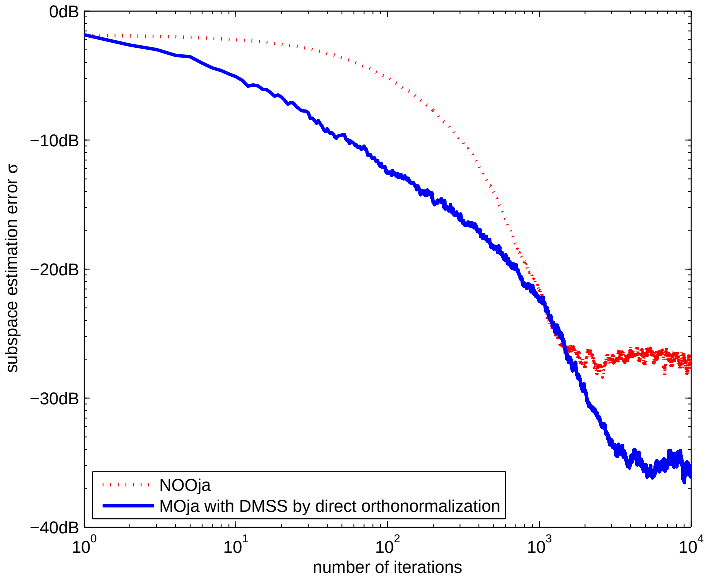

Fig. 5.1: Subspace estimation error for MOja with DMSS by direct orthonormalization and NOOja.

In Figures 5.3 and 5.4, we compare the performance of MOja with DMSS by separate orthogonalization and normalization and FOOja, since the former can be considered as FOOja with DMSS. In FOOja, $\beta(i) = 0.01$. In MOja with DMSS by separate orthogonalization and normalization, $\alpha = 0.998$, $\gamma = 0.005$, and $\varepsilon = 0$. As we can see MOja with DMSS by separate orthogonalization and normalization has faster convergence and similar orthogonality error compared to FOOja.

In Figures 5.5 and 5.6, the performance of MOja with DMSS by direct orthonormalization and MOja with DMSS by separate orthogonalization and normalization are compared. Their settings are the same as those of the earlier simulations in this section. As can been seen, they have similar estimation error performance. As far as orthogonality error is concerned, the orthogonality error

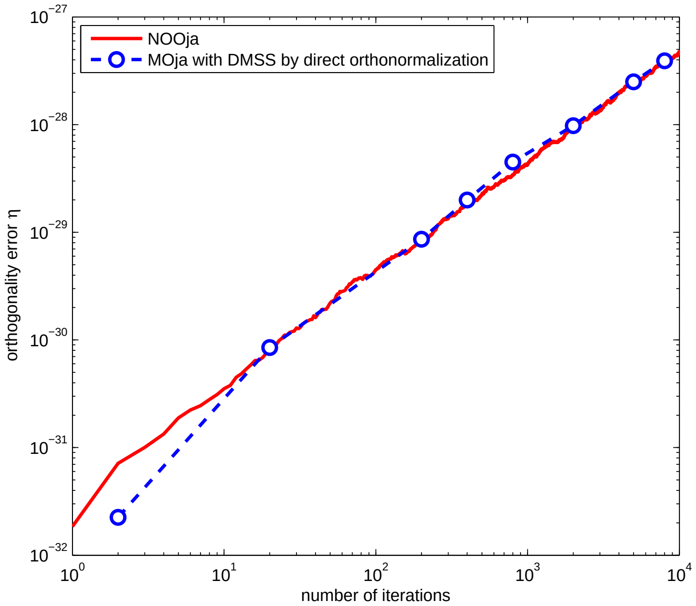

Fig. 5.2: Orthogonality error for MOja with DMSS by direct orthonormalization and NOOja.

of MOja with DMSS by direct orthonormalization slightly grows with iterations, whereas the orthogonality error of MOja with DMSS by separate orthogonalization and normalization remains stable.

5.5.2 Simulation Results and Discussion for YK with DMSS

In Figures 5.7 and 5.8, we compare performance of YK, YK with DMSS by Givens rotation, YK with DMSS by direct orthonormalization, and YK with DMSS by eigendecomposition. In YK, $\beta = 0.0045$. In YK with DMSS by Givens rotation, $\alpha = 0.998$, $\gamma = 0.15$, and $\varepsilon = 0$. In YK with DMSS by direct orthonormalization, $\alpha = 0.998$, $\gamma = 0.18$, and $\varepsilon = 0.001$. To make sure that $\mathbf{P}(i)$ is of full rank, we will check for every instant whether $|1 - \beta_1(i)/\beta_2(i)| < 0.15$. If $|1 - \beta_1(i)/\beta_2(i)| < 0.15$, the bigger one between $\beta_1(i)$ and $\beta_2(i)$ will be increased to 1.2 times its original

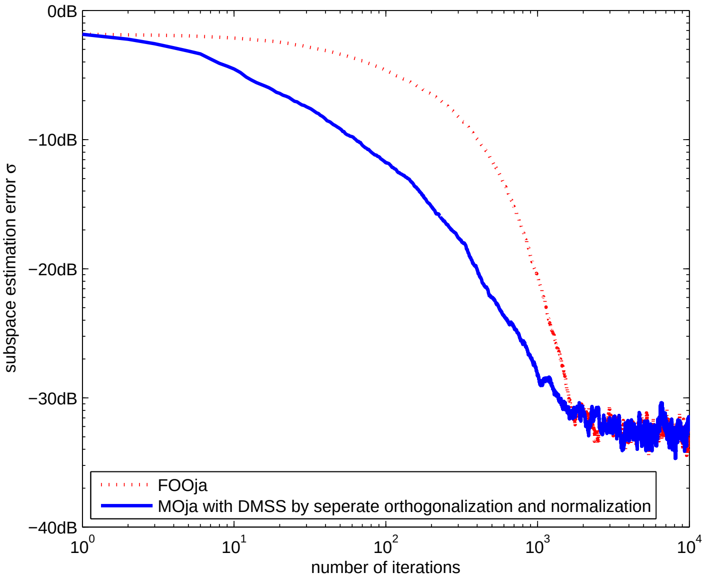

Fig. 5.3: Subspace estimation error for MOja with DMSS by separate orthogonalization and normalization and FOOja.

value. In YK with DMSS by eigendecomposition, $\alpha = 0.998$, $\gamma = 0.15$, and $\varepsilon = 0$. As we can see YK with the different proposed DMSS strategies have faster convergence and similar orthogonality error as the original YK algorithm.

5.6 Conclusion

In this chapter, we have proposed a new optimal diagonal-matrix step-size strategy for MOja and YK algorithms. The proposed optimal step-size matrix controls the updating rate of the subspace vectors individually. MOja and YK with this proposed step-size strategy still cannot give satisfactory results, since they suffer from either instability or computational complexity issues. Two implementations were proposed to stablize MOja with optimal step-size matrix. And

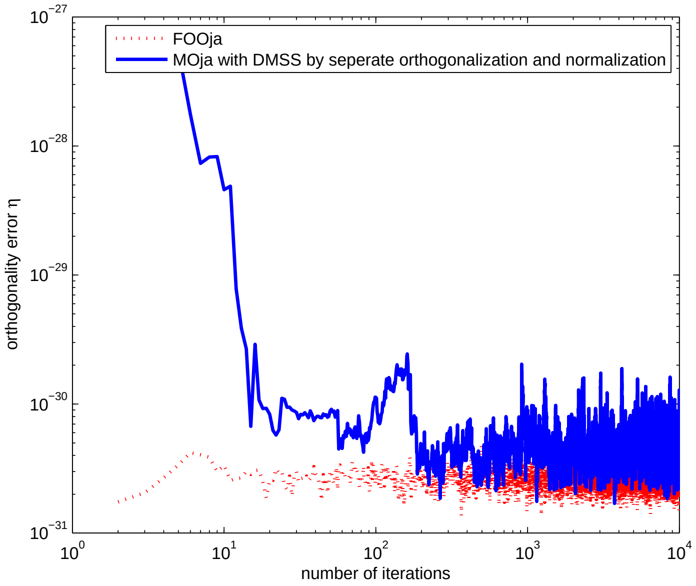

Fig. 5.4: Orthogonality error for MOja with DMSS by separate orthogonalization and normalization and FOOja.

three implementations were designed to reduce the computational complexity of YK with optimal step-size matrix. Simulation results show that the proposed step-size strategies result in faster convergence rate and/or smaller steady-state error than the original algorithms, while retaining $O(NP)$ computational complexity and keeping good stability. The proposed methods can also be extended to other algorithms.

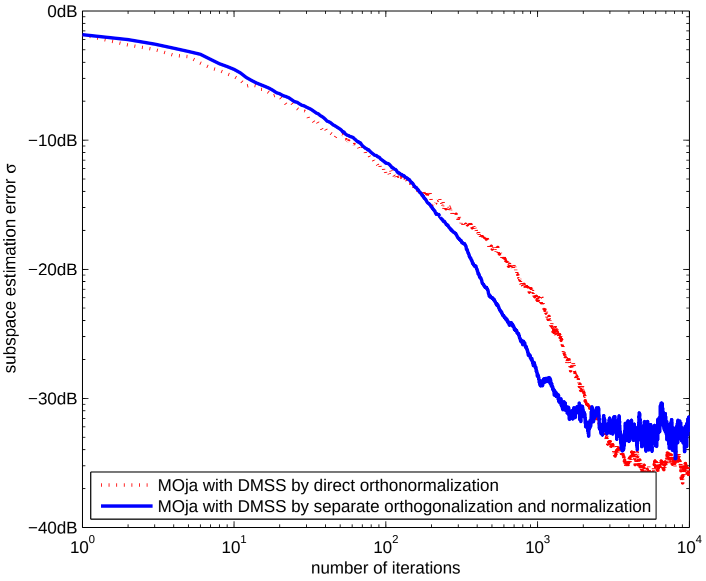

Fig. 5.5: Subspace estimation error for MOja with DMSS by direct orthonormalization and MOja with DMSS by separate orthogonalization and normalization.

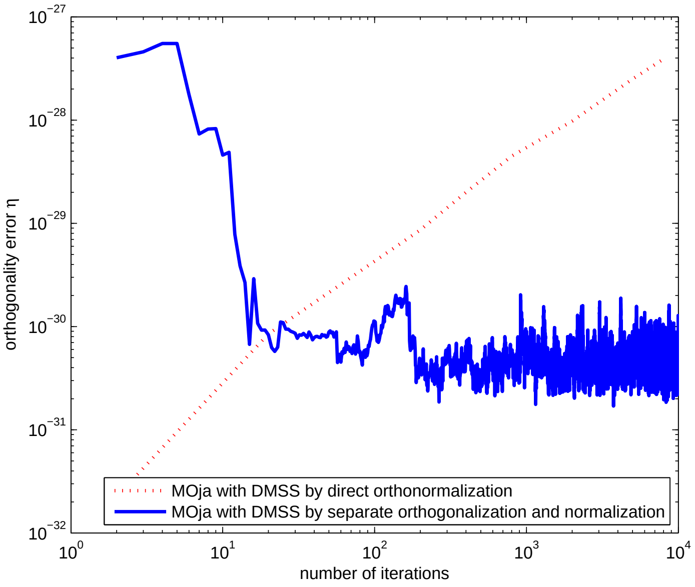

Fig. 5.6: Orthogonality error for MOja with DMSS by direct orthonormalization and MOja with DMSS by separate orthogonalization and normalization.

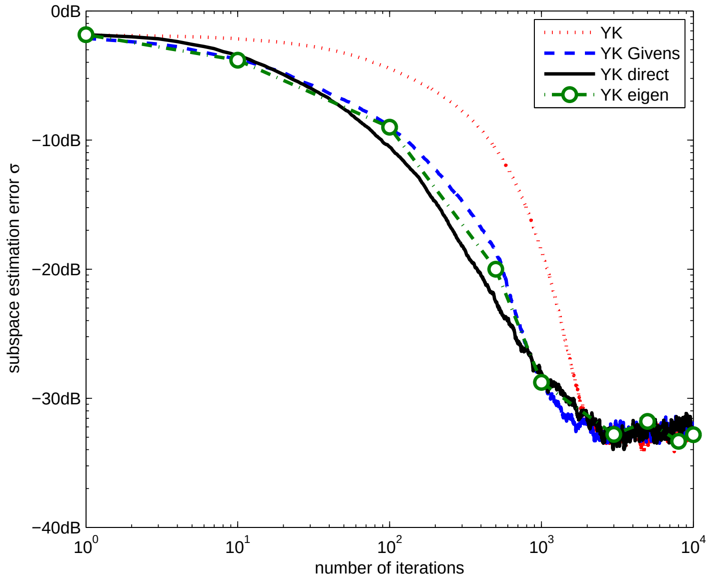

Fig. 5.7: Subspace estimation error for YK, YK with DMSS by Givens rotation, YK with DMSS by direction orthonormalization, and YK with DMSS by eigendecomposition.

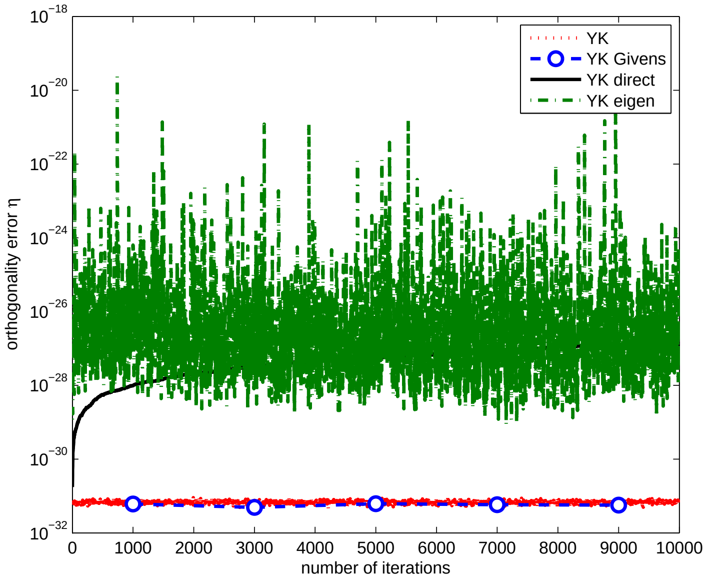

Fig. 5.8: Orthogonality error for YK, YK with DMSS by direction orthonormalization, and YK with DMSS by eigendecomposition

Chapter 6

Adaptive Noise Subspace Estimation Algorithm Suitable for VLSI Implementation

In this chapter, we propose a noise subspace estimation algorithm called SFRANS, which is suitable for VLSI implementation since SFRANS does not require square-root and division operations. The proposed algorithm is based on FRANS [9], but with much better stability. Through ODE analysis, we show that SFRANS is stable on the manifold and is bounded at the equilibrium point.

6.1 Introduction

As we have mentioned in Chapter 3, FRANS [9] is a low cost implementation of Yang and Kaveh's algorithm [110] with computational complexity $O(NP)$. However, it suffers from instability due to accumulation of roundoff errors [68]. To improve its numerical stability while retaining computational complexity at $O(NP)$, HFRANS is proposed using numerically well-behaved Householder transform [11]. Nevertheless, HFRANS requires 4 divisions and 1 square-root operation at each iteration which make it unfriendly for VLSI implementation [44], since square-roots or division operations require far more computational effort than multiplications. Therefore, it is highly desirable to minimize the number of square-roots and divisions required in algorithms.

In this chapter, a VLSI friendly approach (free from square-root or division operations) is proposed to stabilize FRANS, while retaining its low computational complexity. To achieve this goal, we first approximate FRANS using Taylor's series approximation. The resulting algorithm does not require square-root or division operations. However, it loses orthonormality rapidly as FRANS. The conventional orthonormalization techniques include matrix inverse square-root method or matrix inverse method, where square-roots and divisions are inevitable [55]. Instead of the conventional method, a simple stabilizing factor is added to the update equation. The obtained algorithm is called SFRANS (stabilized FRANS algorithm). An ordinary differential equation (ODE) analysis is provided to prove that if the received signal is stationary, the estimated noise subspace remains orthonormal. Performance and computational complexity of the proposed algorithm are compared with the self-stabilized minor subspace rule (SMSR) [37], which is also a stable noise subspace estimation algorithm suitable for VLSI implementation. Simulation results show when SMSR and SFRANS have similar estimation error performance, the proposed algorithm has smaller orthogonality error than SMSR with similar computational complexity.

In this chapter, we first propose SFRANS algorithm in Section 6.2. In Section 6.3 an ODE analysis is provided to analyze the orthonormality of SFRANS. Simulation results and discussions are given in Section 6.4. The chapter is concluded in Section 6.5.

6.2 Proposed SFRANS Algorithm

To facilitate our derivation, let us recall the complete steps of FRANS in Table 6.1. As we can see from Table 6.1, FRANS requires 2 divisions and 1 square-root operation at each iteration which arise from the calculation of $\tau(i)$. The division in Step 5 of Table 6.1 is not counted, because $\mathbf{p}(i)$ will be multiplied by β in Step 6. As stated in [44], square-root and division operations are not appropriate for VLSI implementation. Therefore, in the following, we first try to remove the square-root and divisions in the calculation of $\tau(i)$.

Initialization: choose $\mathbf{W}(0)$ such that $\mathbf{W}^H(0)\mathbf{W}(0) = \mathbf{I}_P$
1. $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{r}(i)$
2. $\beta(i) = \alpha/\|\mathbf{r}(i)\|^2$
3. $\delta(i) = 4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2$
4. $\rho(i) = \sqrt{1-\delta(i)}$
5. $\tau(i) = \frac{1}{\|\mathbf{y}(i)\|^2}(\frac{1}{\rho(i)} - 1)$
6. $\mathbf{p}(i) = -\tau(i)\mathbf{W}(i-1)\mathbf{y}(i)/\beta + 2\mathbf{r}(i)(1+\tau(i)\|\mathbf{y}(i)\|^2)$
7. $\mathbf{W}(i) = \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i)$

Table 6.1: FRANS for noise subspace estimation.

Using a truncated Taylor's series expansion and keeping only up to second-order terms in β, we can simplify $\rho(i)$ in Table 6.1 as

$$\sqrt{1-4\beta(1-\beta\|\mathbf{r}(i)\|^2)\|\mathbf{y}(i)\|^2} \approx 1-2\beta\|\mathbf{y}(i)\|^2+2\beta^2 e(i)\|\mathbf{y}(i)\|^2,$$

where $e(i)=\|\mathbf{r}(i)\|^2-\|\mathbf{y}(i)\|^2$. Similarly, we can simplify $\frac{1}{\rho(i)}$ as

$$\begin{aligned}&\frac{1}{1-2\beta\|\mathbf{y}(i)\|^2+2\beta^2 e(i)\|\mathbf{y}(i)\|^2}\\ &\approx 1+2\beta\|\mathbf{y}(i)\|^2-2\beta^2\|\mathbf{r}(i)\|^2\|\mathbf{y}(i)\|^2+6\beta^2\|\mathbf{y}(i)\|^4.\end{aligned}$$

Hence, we can approximate $\tau(i)$ as

$$\tau(i) \approx 2\beta(1-\beta\|\mathbf{r}(i)\|^2+3\beta\|\mathbf{y}(i)\|^2). \tag{6.1}$$

Injecting these approximations into Step 6 of Table 6.1 and keeping only up to second-order terms in β, the weight updating equation becomes

$$\begin{aligned}\mathbf{W}(i)=\mathbf{W}(i-1)&+2\beta(1-\beta\|\mathbf{r}(i)\|^2+3\beta\|\mathbf{y}(i)\|^2)\mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\\ &-2\beta(1+2\beta\|\mathbf{y}(i)\|^2)\mathbf{r}(i)\mathbf{y}^H(i).\end{aligned} \tag{6.2}$$

Unfortunately, (6.2) is as unstable as the original FRANS algorithm. In [37], Douglas proposes a self-stabilized noise subspace estimation algorithm

$$\begin{aligned}\mathbf{W}(i)=\mathbf{W}(i-1)&+\beta\Big[\mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)-\mathbf{r}(i)\mathbf{y}^H(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\\ &\cdot\mathbf{W}^H(i-1)\mathbf{W}(i-1)\Big].\end{aligned} \tag{6.3}$$

Using a similar technique as in [37], (6.2) can be stabilized by post-multiplying the third term on the RHS of (6.2) a stabilizing factor $\mathbf{W}^H(i-1)\mathbf{W}(i-1)\mathbf{W}^H(i-1)\mathbf{W}(i-1)$. The algorithm resulting from this stabilization step, which we call stabilized FRANS (denoted by SFRANS), is given as in Table 6.2. The complete steps of SFRANS are given in Table 6.2.

```
Initialization: choose $\mathbf{W}(0)$ such that $\mathbf{W}^H(0)\mathbf{W}(0) = \mathbf{I}_P$
1. $\mathbf{u}(i) = \mathbf{W}(i-1)\mathbf{y}(i)$
2. $\mathbf{x}(i) = \mathbf{W}^H(i-1)\mathbf{u}(i)$
3. $\mathbf{z}(i) = \mathbf{W}(i-1)\mathbf{x}(i)$
4. $\mathbf{d}(i) = \mathbf{W}^H(i-1)\mathbf{z}(i)$
5. $\mathbf{W}(i) = \mathbf{W}(i-1) + \beta \left[2(1-\beta\|\mathbf{r}(i)\|^2 + 3\beta\|\mathbf{y}(i)\|^2)\mathbf{u}(i)\mathbf{y}^H(i)\right.$
          $\left. -2(1+2\beta\|\mathbf{y}(i)\|^2)\mathbf{r}(i)\mathbf{d}^H(i)\right]$
```

Table 6.2: SFRANS for noise subspace estimation.

Comparing (6.3) and Table 6.2, we see that they are similar in form. In fact, if we eliminate all the update terms on the RHS of Step 5 that are second order in β, we will get (6.3). Note that SFRANS given in Table 6.2 has $O(NP)$ computational complexity and is completely free from division and square-root operations.

6.3 Convergence Analysis of SFRANS

Based on the stochastic approximation theorem [29, Chap 5] mentioned in Chapter 3 [65], we use first order approximation analysis assuming the step-size to be very small. In addition, an ODE analysis has also been developed for SFRANS.

6.3.1 Stability at the Equilibrium Points

Let us first introduce the following two lemmas [94] which will be used in our analysis. Their proof can be found in [94].Although we use these two lemmas, our ODE analysis is totally different from that in [94]. Because the analysis in [94] is for principal component analysis, while ours is for noise subspace estimation. Besides, the ODE equation under evaluation is different.

Denote a permutation of $\{1, \cdots, N\}$ by π, that is $\pi(i) \in \{1, 2, \cdots, N\}$ with $\pi(i) \neq \pi(j)$ for $i \neq j$ and $\{\pi(1), \cdots, \pi(N)\} = \{1, \cdots, N\}$. A permutation is called identity and is denoted by 1_N if $\pi(i) = i$ for all $i = 1, \cdots, N$.

Lemma 6.1: Let $\mathbf{\Gamma}$ be a diagonal matrix of size $N \times N$ and $\mathbf{G}$ be an arbitrary matrix of size $N \times N$. Let π be a permutation such that $\pi\pi = \pi^2 = 1_N$. Then $\mathbf{G\Gamma G}$ is diagonal if and only if the only non-zero elements of $\mathbf{G}$ are given by $(i, \pi(i))$ for $i = 1, \cdots, N$. This matrix $\mathbf{G}$ is called a nonunitary permutation matrix.

Lemma 6.2: Let $\mathbf{G}$ be a Hermitian nonunitary permutation matrix of size $N \times N$ and π be the corresponding permutation. Let $[\mathbf{G}]_{i,\pi(i)} = [\mathbf{G}]_{\pi(i),i} = g_i$. Then the eigenvalues of $\mathbf{G}$ are given by $\pm|g_i|$ for $i \neq \pi(i)$ and g_j for $j = \pi(j)$, for $i, j = 1, \cdots, N$.

We write the singular value decomposition (SVD) of $\mathbf{W}$ as $\mathbf{W} = \mathbf{U\Sigma V}^H$, where $\mathbf{U}$ is a $N \times P$ orthogonal matrix, $\mathbf{\Sigma}$ is a $P \times P$ diagonal matrix with positive entries $\{\sigma_1^2, \cdots, \sigma_P^2\}$, and $\mathbf{V}$ is a $P \times P$ orthogonal matrix. The ODE corresponding to SFRANS is

$$\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} = -\mathbf{CWW}^H\mathbf{WW}^H\mathbf{W} + \mathbf{WW}^H\mathbf{CW}. \tag{6.4}$$

Equation (6.4) is exactly the same as the ODE corresponding to (6.3). An ODE analysis of (6.4) is given in [37], however it uses a false assumption as shown in [5]. So an approach that is different from that in [37] will be used here.

At equilibrium point, (6.4) satisfies $\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} = \mathbf{0}$. Using the SVD of $\mathbf{W}$, we can write

$$\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} = -\mathbf{C}\mathbf{U}\mathbf{\Sigma}^5\mathbf{V}^H + \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^H\mathbf{C}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = \mathbf{0}. \tag{6.5}$$

Pre- and post-multiplying the above equation by $\mathbf{U}^H$ and $\mathbf{V}$, respectively, leads to

$$\mathbf{U}^H\mathbf{C}\mathbf{U}\mathbf{\Sigma}^5 = \mathbf{\Sigma}^2\mathbf{U}^H\mathbf{C}\mathbf{U}\mathbf{\Sigma}. \tag{6.6}$$

Assume that $\mathbf{C}$ has nonzero distinct eigenvalues $\lambda_1 < \cdots < \lambda_N$. Let $\mathbf{\Lambda}$ be a diagonal matrix given as $\mathbf{\Lambda} = \mathrm{diag}[\lambda_{\pi(1)}, \cdots, \lambda_{\pi(N)}]$, where π is a permutation of $1, \cdots, N$. Then the eigenvalue decomposition (EVD) of $\mathbf{C}$ can be given as $\mathbf{C} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^H$, where $\mathbf{P}$ is a unitary matrix containing the eigenvectors corresponding to $\mathbf{\Lambda}$. Using the EVD of $\mathbf{C}$ in (6.6), we have

$$\mathbf{X}^H\mathbf{\Lambda}\mathbf{X}\mathbf{\Sigma}^5 = \mathbf{\Sigma}^2\mathbf{X}^H\mathbf{\Lambda}\mathbf{X}\mathbf{\Sigma}, \tag{6.7}$$

where $\mathbf{X} = \mathbf{P}^H\mathbf{U}$. Let $\mathbf{A} = \mathbf{X}^H\mathbf{\Lambda}\mathbf{X}$. Then, we have

$$\mathbf{\Sigma}^4 = \mathbf{\Sigma}^{-1}\mathbf{A}^{-1}\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma} \tag{6.8}$$

which implies that $\mathbf{\Sigma}^{-1}\mathbf{A}^{-1}\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma}$ is symmetric, *i.e.*

$$\mathbf{\Sigma}^{-1}\mathbf{A}^{-1}\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma} = \mathbf{\Sigma}\mathbf{A}\mathbf{\Sigma}^2\mathbf{A}^{-1}\mathbf{\Sigma}^{-1}, \tag{6.9}$$

which yields

$$\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma}^2\mathbf{A} = \mathbf{A}\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma}^2. \tag{6.10}$$

Let $b_{ij} = [\mathbf{A}\mathbf{\Sigma}^2\mathbf{A}]_{ij}$. Then we have $\sigma_i^2 b_{ij} = \sigma_j^2 b_{ij}$, which implies that $b_{ij} = 0$ when $i \neq j$. This shows that $\mathbf{A}\mathbf{\Sigma}^2\mathbf{A}$ is a diagonal matrix. From Lemma 6.1, $\mathbf{A}$ is a Hermitian nonunitary permutation matrix. Since $\mathbf{A} = \mathbf{X}^H\mathbf{\Lambda}\mathbf{X}$ and $\mathbf{\Lambda}$ contains distinct positive entries, $\mathbf{A}$ is positive definite. From Lemma 6.2, $\mathbf{A}$ must be a diagonal matrix with distinct positive entries. Therefore, $\mathbf{X}$ must be a rectangular permutation matrix. Moreover, because $\mathbf{A}$ is diagonal, we get from (6.9)

$$\mathbf{\Sigma}^{-1}\mathbf{A}^{-1}\mathbf{\Sigma}^2\mathbf{A}\mathbf{\Sigma} = \mathbf{\Sigma}^2. \tag{6.11}$$

Comparing (6.8) and (6.11), we get

$$\mathbf{\Sigma}^4 = \mathbf{\Sigma}^2. \tag{6.12}$$

Since $\mathbf{\Sigma}$ is a diagonal matrix with all positive entries, $\mathbf{\Sigma} = \mathbf{I}_P$. Finally, we have the equilibrium points of SFRANS as

$$\dot{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = (\mathbf{P}\mathbf{X})\,\mathbf{V}^H, \tag{6.13}$$

where $\mathbf{PX}$ represents a $N \times P$ matrix whose columns correspond to eigenvectors of $\mathbf{C}$ in arbitrary order without duplication, and $\mathbf{V}$ is a $P \times P$ unitary matrix.

We now investigate the stability of SFRANS at the equilibrium point. Let

$\mathbf{Z} = \mathbf{P}^H\mathbf{W}\mathbf{V}$. Pre and post-multiplying (6.4) by $\mathbf{P}^H$ and $\mathbf{V}$, respectively, we have

$$\frac{\mathrm{d}\mathbf{Z}}{\mathrm{d}t} = -\mathbf{\Lambda}\mathbf{Z}\mathbf{Z}^H\mathbf{Z}\mathbf{Z}^H\mathbf{Z} + \mathbf{Z}\mathbf{Z}^H\mathbf{\Lambda}\mathbf{Z}. \tag{6.14}$$

Without lost of generality, let $\mathbf{PX}$ consist of the first P columns of $\mathbf{P}$ at the equilibrium point $\dot{\mathbf{W}} = \mathbf{P}\mathbf{X}\mathbf{V}^H$, since the first P columns of $\mathbf{P}$ are eigenvectors of $\mathbf{C}$ in arbitrary order. Then, it is clear that $\dot{\mathbf{Z}} = \mathbf{P}^H\dot{\mathbf{W}}\mathbf{V}$ is also an equilibrium point and it is clear that $\dot{\mathbf{Z}} = [\mathbf{I}_P, \mathbf{0}_{P,N-P}]^H$.

We consider the ODE of the perturbation $\mathbf{E}(t)$ at the equilibrium point

$$\begin{aligned}\frac{\mathrm{d}\mathbf{E}}{\mathrm{d}t} = & \left.\frac{\mathrm{d}\mathbf{Z}}{\mathrm{d}t}\right|_{\mathbf{Z}=\dot{\mathbf{Z}}+\mathbf{E}} \\ = & -\mathbf{\Lambda}\left(\dot{\mathbf{Z}}+\mathbf{E}\right)\left(\dot{\mathbf{Z}}^H+\mathbf{E}^H\right)\left(\dot{\mathbf{Z}}+\mathbf{E}\right)\left(\dot{\mathbf{Z}}^H+\mathbf{E}^H\right)\left(\dot{\mathbf{Z}}+\mathbf{E}\right) \\ & +\left(\dot{\mathbf{Z}}+\mathbf{E}\right)\left(\dot{\mathbf{Z}}^H+\mathbf{E}^H\right)\mathbf{\Lambda}\left(\dot{\mathbf{Z}}+\mathbf{E}\right) \\ = & -\mathbf{\Lambda}\Big(\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}} + \dot{\mathbf{Z}}\mathbf{E}^H\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}} + \mathbf{E}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}} \\ & +\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\mathbf{E}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}} + \dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}}\mathbf{E}^H\dot{\mathbf{Z}} + \dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\mathbf{E}\Big) \\ & +\left(\dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\mathbf{\Lambda}\dot{\mathbf{Z}} + \dot{\mathbf{Z}}\mathbf{E}^H\mathbf{\Lambda}\dot{\mathbf{Z}} + \mathbf{E}\dot{\mathbf{Z}}^H\mathbf{\Lambda}\dot{\mathbf{Z}} + \dot{\mathbf{Z}}\dot{\mathbf{Z}}^H\mathbf{\Lambda}\mathbf{E}\right) + \mathcal{O}\left(\|\mathbf{E}\|^2\right), \end{aligned} \tag{6.15}$$

where $\mathcal{O}(\|\mathbf{E}\|^2)$ is a matrix containing terms with $\mathbf{E}$ in second or larger order. We assume that $\mathcal{O}\left(\|\mathbf{E}\|^2\right)$ is negligibly small.

Let $[\mathbf{E}]_{ij} = e_{ij}$. Note that $\dot{\mathbf{Z}} = [\mathbf{I}_P, \mathbf{0}_{P,N-P}]^H$. Then, for $1 \leq j \leq P < i \leq N$, the ODE of the element e_{ij} can be expressed using (6.15) as

$$\frac{\mathrm{d}e_{ij}}{\mathrm{d}t} = -e_{ij}\left(\lambda_{\pi(i)} - \lambda_{\pi(j)}\right). \tag{6.16}$$

The solution to (6.16) is $e_{ij}(t) = Ce^{-(\lambda_{\pi(i)}-\lambda_{\pi(j)})}$, where $C = \pm e^{C_1}$ and C can be either positive or negative. It follows that if and only if $\lambda_{\pi(i)} > \lambda_{\pi(j)}$, $e_{ij}(t)$ will converge to 0. Thus, for $e_{ij}(\infty)$ to go to 0, we must have $\pi(j) \in \{1, 2, \cdots, P\}$ for $1 \leq j \leq P$. For $1 \leq i, j \leq P$, we obtain using (6.15)

$$\frac{\mathrm{d}e_{ij}}{\mathrm{d}t} = \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) e_{ij} + \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) e_{ji}^*, \tag{6.17}$$

where e_{ji}^* is the conjugate of e_{ji}. Similarly, switching the position of i and j in (6.17), we obtain

$$\frac{\mathrm{d}e_{ji}}{\mathrm{d}t} = \left(-2\lambda_{\pi(j)} + \lambda_{\pi(i)}\right) e_{ij}^* + \left(-2\lambda_{\pi(j)} + \lambda_{\pi(i)}\right) e_{ji}. \tag{6.18}$$

Rearranging (6.17), we have

$$e_{ji} = \frac{1}{-2\lambda_{\pi(i)} + \lambda_{\pi(j)}} \left[\frac{\mathrm{d}e_{ij}^*}{\mathrm{d}t} - \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) e_{ij}^*\right]. \tag{6.19}$$

Injecting (6.19) into (6.18), we obtain

$$\frac{\mathrm{d}e_{ji}}{\mathrm{d}t} = \left(-2\lambda_{\pi(j)} + \lambda_{\pi(i)}\right) e_{ij}^* + \frac{-2\lambda_{\pi(j)} + \lambda_{\pi(i)}}{-2\lambda_{\pi(i)} + \lambda_{\pi(j)}} \left[\frac{\mathrm{d}e_{ij}^*}{\mathrm{d}t} - \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) e_{ij}^*\right]. \tag{6.20}$$

From (6.17), we have

$$\frac{\mathrm{d}^2 e_{ij}^*}{\mathrm{d}t^2} = \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) \frac{\mathrm{d}e_{ij}^*}{\mathrm{d}t} + \left(-2\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) \frac{\mathrm{d}e_{ji}}{\mathrm{d}t}. \tag{6.21}$$

Replacing $\frac{\mathrm{d}e_{ji}}{\mathrm{d}t}$ in (6.21) by (6.20), we obtain the following second-order ODE of e_{ij} as

$$\frac{\mathrm{d}^2 e_{ij}}{\mathrm{d}t^2} + \left(\lambda_{\pi(i)} + \lambda_{\pi(j)}\right) \frac{\mathrm{d}e_{ij}}{\mathrm{d}t} = 0 \text{ for } 1 \leq i, j \leq P. \tag{6.22}$$

The solution of the above ODE is $e_{ij}(t) = C_1 + C_2 e^{-(\lambda_{\pi(i)}+\lambda_{\pi(j)})t}$, where C_1 and C_2 are real numbers depending on the initial conditions $e_{ij}(0)$ and $\frac{\mathrm{d}e_{ij}}{\mathrm{d}t}|_{t=0}$. The term $e^{-(\lambda_{\pi(i)}+\lambda_{\pi(j)})t}$ will go to 0 since $\lambda_{\pi(i)} > 0$ for all i. But the residue error C_1 will remain. As can be seen from simulation results in Section 6.4, the estimation error of $\mathbf{W}(i)$ with large iteration number i is stable and small, which means the residue error C_1 is negligibly small.

For the above analysis of the behavior of $e_{ij}(t)$ for $1 \leq i \leq P < i \leq N$ or $1 \leq i, j \leq P$, we can conclude that we must have $\pi(j) \in \{1, 2, \cdots, P\}$ for the perturbation error to be bounded. This implies that the eigenvectors in $\mathbf{PX}$ at the equilibrium point $\dot{\mathbf{W}} = \mathbf{PXV}^H$ must be the eigenvectors corresponding to the P smallest eigenvalues.

6.3.2 Stability on the Manifold

If the signal is stationary (i.e., the correlation matrix $\mathbf{C} = E[\mathbf{r}(i)\mathbf{r}^H(i)]$ is independent of iteration index i), we can prove that the proposed algorithm can estimate the noise subspace on the manifold $S = \left\{\mathbf{W}|\mathbf{W}^H\mathbf{W} = \mathbf{I}_P\right\}$. Stability on the manifold is a desirable property for adaptive algorithms, since orthogonality of $\mathbf{W}$ is guaranteed by the manifold.

Using (6.4), we get

$$\begin{aligned}\frac{\mathrm{d}\left(\mathbf{W}^H\mathbf{W}\right)}{\mathrm{d}t} &= \mathbf{W}^H\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{W}^H}{\mathrm{d}t}\mathbf{W} \\ &= -\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W} + \mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W} \\ &\quad -\mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W} + \mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W}. \qquad (6.23)\end{aligned}$$

We consider the variation $\delta\mathbf{W}(t)$ from a point $\hat{\mathbf{W}}$ on the manifold S, where $\delta\mathbf{W}(t)$

is assumed to be small. At the point $\mathbf{W}(t) = \hat{\mathbf{W}} + \delta\mathbf{W}(t)$, we have (dropping second-order variation terms)

$$\mathbf{W}^H(t)\mathbf{W}(t) \approx \mathbf{I} + \hat{\mathbf{W}}^H\delta\mathbf{W}(t) + \delta\mathbf{W}^H(t)\hat{\mathbf{W}}. \tag{6.24}$$

To analyze the stability of (6.4) on the manifold, we shall investigate the behavior of the variation $\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)$. After dropping second-order variation terms, we obtain

$$\delta\left(\mathbf{W}^H\mathbf{W}\right)(t) \approx \hat{\mathbf{W}}^H\delta\mathbf{W}(t) + \delta\mathbf{W}^H(t)\hat{\mathbf{W}}. \tag{6.25}$$

Comparing (6.24) and (6.25), we can obtain

$$\mathbf{W}^H(t)\mathbf{W}(t) \approx \mathbf{I} + \delta\left(\mathbf{W}^H\mathbf{W}\right)(t). \tag{6.26}$$

Therefore, we get

$$\frac{\mathrm{d}\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)}{\mathrm{d}t} \approx \left.\frac{\mathrm{d}\left(\mathbf{W}^H\mathbf{W}\right)}{\mathrm{d}t}\right|_{\mathbf{W}=\hat{\mathbf{W}}+\delta\mathbf{W}(t)}. \tag{6.27}$$

In view of (6.27), we can obtain the ODE of $\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)$ by injecting $\mathbf{W}(t) = \hat{\mathbf{W}} + \delta\mathbf{W}(t)$ into (6.23). Thus, we get

$$\begin{aligned}\frac{\mathrm{d}\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)}{\mathrm{d}t} &\approx \left(\mathbf{H} + \hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\delta\left(\mathbf{W}^H\mathbf{W}\right)\right) + \left(\mathbf{H} + \delta\left(\mathbf{W}^H\mathbf{W}\right)\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\right)\\ &\quad - \left(\mathbf{H} + 2\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\delta\left(\mathbf{W}^H\mathbf{W}\right)\right) - \left(\mathbf{H} + 2\delta\left(\mathbf{W}^H\mathbf{W}\right)\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\right)\\ &\approx -\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\delta\left(\mathbf{W}^H\mathbf{W}\right) - \delta\left(\mathbf{W}^H\mathbf{W}\right)\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}},\end{aligned} \tag{6.28}$$

where $\mathbf{H} = \mathbf{W}^H\mathbf{C}\mathbf{W}$. The detailed derivation of (6.28) is given in Appendix C.1. In terms of vector operator, the above equation can be rewritten as

$$\frac{\mathrm{d}vec\left[\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right]}{\mathrm{d}t} \approx -\left(\mathbf{I}\otimes\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}} + \hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\otimes\mathbf{I}\right)vec\left[\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right]. \tag{6.29}$$

Since $\mathbf{C}$ is positive definite, $\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}$ is also positive definite. Also, since the variation is assumed to be small, removal of second-order variation terms will not influence stability of the manifold. Therefore, $\left(\mathbf{I}\otimes\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\ +\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\otimes\mathbf{I}\right)$ is also positive definite based on Lemma 1 of Tanaka's paper [94]. Hence, as $vec\left[\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right] \rightarrow 0$, we have $\delta\left(\mathbf{W}^H\mathbf{W}\right)(t) \rightarrow 0$ when $t \rightarrow \infty$. This completes our proof that the manifold is stable under perturbation in the learning rule (6.4).

6.4 Simulation Results and Discussion

In this section, the performance of the proposed SFRANS algorithm is compared with FRANS [9] and the self-stabilized minor subspace rule (SMSR) proposed by Douglas *et al.* [37]. In our simulations, we set the covariance matrix to

$$\mathbf{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix},$$

where $N = 4$ and $P = 2$. The observed data $\mathbf{r}(i)$ is generated as stated in Section 2.5.1.

Figure 6.1 shows the average evolution of the estimation error $\sigma(i)$ [see (2.35) in Chapter 2] for FRANS, SMSR and SFRANS. The step-size β is chosen to be 0.002 for SMSR, 0.001 for FRANS and 0.001 for SFRANS. The step-sizes are chosen such that all the algorithms have no observable difference in convergence rate, so as to make the comparison fair. Note that the sudden drop in estimation error of FRANS indicates that the estimated subspace is divergent from the true one after around 6×10^4 iterations.

To show the divergence of FRANS after around 6×10^4 iterations, we introduce projection error $v(i)$ as

$$v(i) = \frac{1}{r_0} \sum_{r=1}^{r_0} \|\mathbf{W}_r(i)\mathbf{W}_r(i)^H - \mathbf{U}_n\mathbf{U}_n^H(i)\|_F^2, \tag{6.30}$$

where r_0 denotes the number of Monte Carlo runs, $\mathbf{W}_r(i)$ denotes the estimated subspace matrix $\mathbf{W}(i)$ in the rth run, $\|.\|_F$ denotes the Frobenius norm, and $\mathbf{U}_n$ is the noise subspace of the underlying covariance matrix $\mathbf{C}$. Here, $v(i)$ measures the projection error of $\mathbf{W}_r(i)$ with respect to the true noise subspace $\mathbf{U}_n$. As can be seen from Figure 6.2, there is a sudden increase of $v(i)$ for FRANS after around 6×10^4 iterations, indicating the divergence of FRANS.

Figure 6.3 shows the average evolution of the orthogonality error $\eta(i)$ [see (2.36) in Chapter 2] for the three algorithms. The proposed SFRANS has a significantly smaller orthogonality error than SMSR, whereas FRANS quickly loses its orthogonality due to numerical error buildup.

The computational complexities of the three algorithms are compared in Table

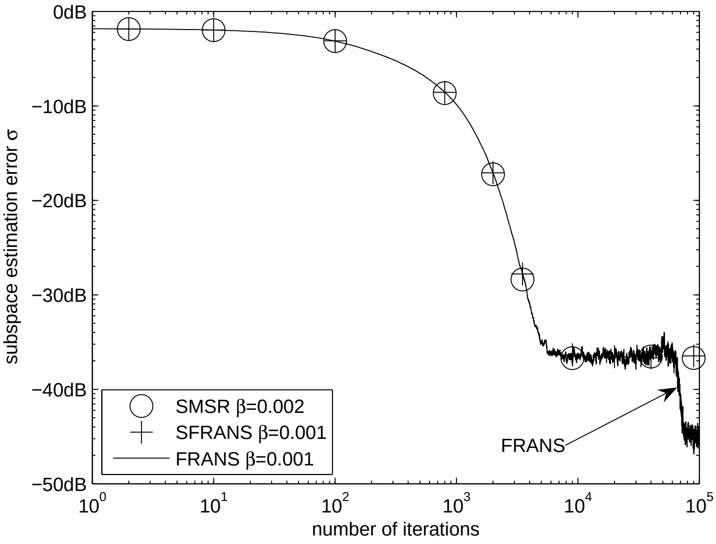

Fig. 6.1: Estimation error $\sigma(i)$ for SMSR, FRANS and SFRANS.

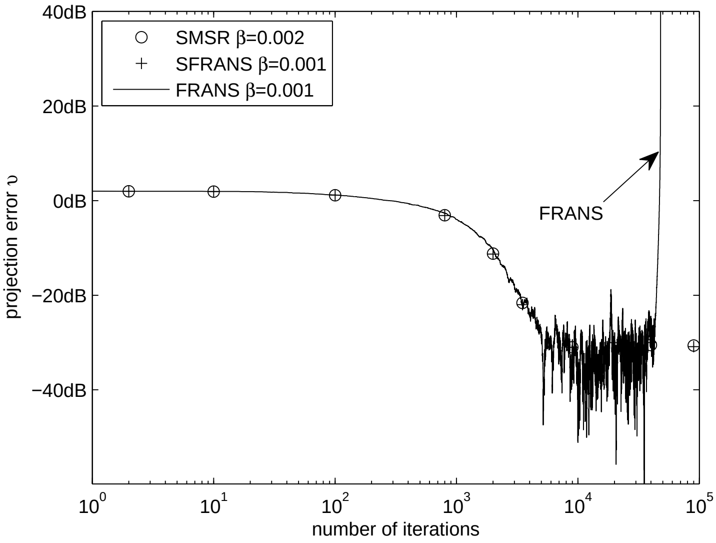

Fig. 6.2: Projection error $\upsilon(i)$ for SMSR, FRANS and SFRANS.

6.3. Note that they all have $O(NP)$ complexity, whereas SMSR and SFRANS involve no square-root or division. Therefore, we conclude that when they have

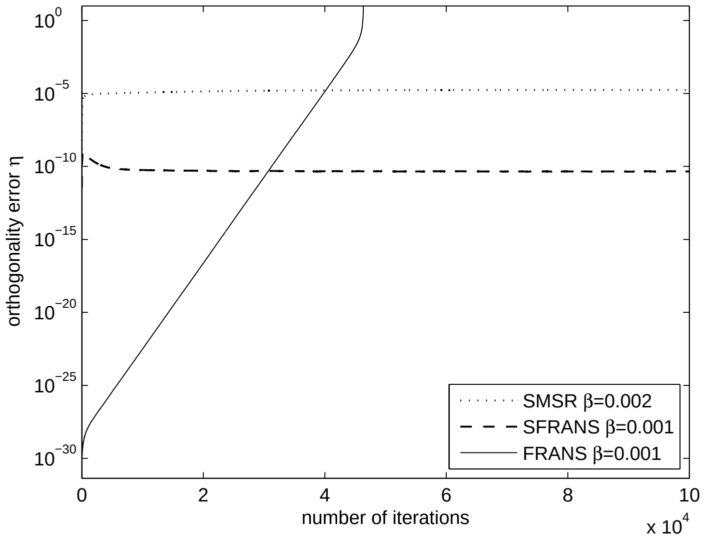

Fig. 6.3: Orthogonality error $\eta(i)$ for SMSR, FRANS and SFRANS.

similar convergence rate in estimation error, SFRANS has smaller orthogonality error than SMSR with similar computational complexity, and is superior to FRANS in terms of stability.

	multiplication	division	square-root
SMSR	$6NP$	0	0
FRANS	$4NP$	3	1
SFRANS	$6NP$	0	0

Table 6.3: Computational complexities for SMSR, FRANS and SFRANS.

6.5 Conclusion

In this chapter, we have considered the problem of developing a VLSI friendly noise subspace estimation algorithm. We have proposed a stable and low cost noise subspace estimation algorithm, SFRANS, which requires no square-root or division operation. The proposed algorithm is derived from FRANS through Taylor series approximation, but with similar computational complexity and much better stability. An ODE analysis is given to estabilish the stability of the proposed algorithm.

Chapter 7

Conclusion and Proposals for Future Work

7.1 Conclusion

In the literature, many of the existing noise subspace algorithms lose their orthogonality gradually and no longer extract the true subspace. Therefore, our study of this thesis starts from analyzing the stability of FRANS algorithm [9] to have a better understanding of how to make noise subspace estimation algorithms stable. More recently, several stable algorithms with computational complexity $O(NP)$ were proposed. However, they converge slowly since they are gradient based and non-optimal step-sizes are used to update all the subspace vectors at the same speed. They also require division and square-root operations which make them difficult for real-time implementation. Therefore, in this thesis, we propose novel approaches that result in stable fast subspace estimation algorithms, some of which are even VLSI friendly, to enhance the performance of bandwidth efficient high speed communications systems.

In Chapter 2, we started with a short review of the mathematical preliminaries, which will be needed in the subsequent chapters. It is followed by a review of the existing signal subspace algorithms and noise subspace algorithms. Data generation method and performance measures used for simulations in this thesis are also given in Chapter 2.

In Chapter 3, we showed that the instability of FRANS is due to the round-off error accumulation, given that the noise subspace estimate is orthonormally initialized. Therefore, with finite precision calculation, there is no way to stabilize FRANS. However, HFRANS based on Householder transformation displays a much better numerical behavior. Therefore, HFRANS algorithm is suitable for noise subspace estimation applications, especially since it has a low computational complexity of $O(NP)$ multiplications, 1 square-root and 4 division operations per iteration.

Chapter 4 presented two different step-size adaptation approaches for HFRANS. The first one is a stochastic gradient-descent based approach. It offers smaller steady-state estimation error, faster convergence speed, similar orthogonality and comparable computational complexity as compared with HFRANS at the same convergence speed. The second one is based on the estimation of optimal step-size at each instant, and it is less dependant on the initial conditions of the step-size adaptation. Both these step-size strategies can achieve smaller error in the steady-state. And they are not limited to HFRANS, but can be applied to other algorithms as well.

We proposed an optimal diagonal-matrix step-size strategy for MOja and Yang and Kaveh's algorithms [105, 110] in Chapter 5. The proposed step-size strategy controls the decoupled subspace vectors individually as compared to conventional methods where all the subspace vectors are updated using the same step-size value. Nevertheless, MOja and YK with the proposed strategy still have instability or high computational complexity issue as the original algorithms. Several stable low cost implementations were then developed to restore stability while maintaining computational complexity at $O(NP)$. An optimal value of the diagonal step-size matrix is also obtained using the method developed in Chapter 4.

Chapter 6 focused on developing a VLSI friendly algorithm. Because of the use of Householder matrix, it is difficult to simplify HFRANS while retaining its stability. So we proposed a stabilized FRANS algorithm, called SFRANS. It achieves a good tradeoff between FRANS and HFRANS algorithm, since it is much more stable than FRANS and requires no square-root or division operations.

7.2 Future Work

Extensions to this work seem to be possible in the following directions.

- By comparing HFRANS and SFRANS algorithms, we feel that there might be a general form for stable Householder based noise subspace estimation algorithms. If it is true, any algorithm in this general form can converge to the noise subspace. They should only differ in the speed and accuracy of convergence. Therefore, we need to investigate the existence of a general algorithm and the corresponding optimum algorithm.
- While developing SFRANS algorithm, we made the assumption that the step-size is small. However, the variable step-size strategy developed in Chapter 4 is based on a large step-size in the beginning. Hence, we need to examine if the assumptions

in SFRANS permit the development of a variable step-size strategy, such as that in Chapter 4, for SFRANS.

• The SFRANS algorithms developed in this thesis are for noise subspace estimation. We could explore the development of a unified SFRANS for estimation of signal and noise subspaces.

• Since part of the difficulty we face in doing an exact analysis of HFRANS is due to the constrained cost function, it is easier in terms of calculation if we can find an unconstrained cost function as in [79], and solve the unconstrained optimization problem. The convergence analysis can be possibly done through ODE analysis.

• HFRANS algorithm converges slow, since it is a gradient based algorithm. Newton type algorithms are known to converge faster than gradient based algorithms. Mathew *et al.* [72] estimated all or some of the orthogonal eigenvectors with a quasi-Newton method. However, it has $O(N^2)$ complexity. Therefore, we need to investigate the development of a low complexity quasi-Newton algorithm for noise subspace estimation.

• The recently proposed minimum noise subspace (MNS) method [1, 56] is an alternative to the conventional noise subspace estimation method. The MNS technique was first introduced as a computationally efficient subspace technique which exploits a minimum number of noise vectors for multichannel blind system identification. However, most of the MNS algorithms proposed so far are not adaptive and the noise subspace estimated is non-orthogonal, and therefore are not suitable for applying to wireless communications problems. Therefore, we may explore the development of adaptive MNS algorithms which can estimate noise subspace with orthogonal basis.

• In this thesis, what we dealt with is a special case of the problem of subspace estimation, where dimension of the noise subspace is known. We can also design subspace estimation algorithms, where subspace dimension is unknown. The motivation for developing estimation schemes for this case is that in CDMA systems the noise subspace dimension could be unknown and vary with time. This is because noise subspace depends on the number of users occupying the channel and this number is variable, since users keep entering or exiting the system.

Bibliography

[1] K. Abed-Meraim and Y. Hua, "Blind identification of multi-input multi-output system using minimum noise subspace," *IEEE Trans. Signal Proc.*, vol. 45, no. 1, pp. 254-258, Jan. 1997.

[2] K. Abed-Meraim, A. Chkeif and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Proc. Lett.*, vol. 7, no. 3, pp. 60-62, Mar. 2000.

[3] K. Adeb-Meraim, A. Chkeif, Y. Hua and S. Attallah, "On a class of orthonormal algorithms for principal and minor subspace tracking," *Journal of VLSI Signal Proc.*, vol. 31, pp. 57-70, 2002.

[4] K. Abed-Meraim, S. Attallah, A. Chkeif and Y. Hua, "Orthogonal Oja algorithm," *IEEE Signal Proc. Lett.*, vol. 7, no. 5, pp. 116-119, May 2000.

[5] K. Abed-Meraim, S. Attallah, A. Chkeif and Y. Hua, "Self-stabilized minor subspace extraction algorithm based on Householder transformation," in *Proc. IEEE Workshop on Stat. Signal and Array Proc. (SSAP)*, Pocono Manor, USA, Aug. 2000, pp. 90-93.

[6] A. N. Amini and T. T. Georgiou, "Tunable line spectral estimators based on state-covariance subspace analysis," *IEEE Trans. Signal Proc.*, vol. 54, no. 7, pp. 2662-2671, Jul. 2006.

[7] S. Attallah, "Adaptive algorithms for noise subspace estimation with application to MC-CDMA," in *Proc. Intl. Symp. Signal Proc. and its Appl. (ISSPA)*, Paris, France, Jul. 2003, vol. 1, pp. 501-504.

[8] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Proc. Lett.*, vol. 8, no. 7, pp. 203-206, Jul. 2001.

[9] S. Attallah and K. Abed-Maraim, "Low-cost adaptive algorithm for noise subspace estimation," *Electron. Lett.*, vol. 38, no. 12, pp. 609-611, Jun. 2002.

[10] S. Attallah, "Revisiting adaptive signal subspace estimation based on Rayleigh's quotient," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Salt Lake City, USA, May 2001, pp. 4009-4012.

[11] S. Attallah, "The generalized Rayleigh's quotient adaptive noise subspace algorithm: A Householder transformation-based implementation," *IEEE Trans. Ciruits and Systems*, vol. 53, no. 1, pp. 3-7, Jan. 2006.

[12] R. Badeau, G. Richard and B. David, "Approximated power iterations for fast subspace tracking," in *Proc. Intl. Symp. Signal Proc. and its Appl. (ISSPA)*, vol. 2, Paris, France, Jul. 2003, pp. 583-586.

[13] R. Badeau, B. David and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Proc.*, vol. 53, no. 8, pp. 2931-2941, Aug. 2005.

[14] R. Badeau, B. David and G. Richard, "YAST algorithm for minor subspace tracking," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Toulouse, France, May 2006, pp. 552-555.

[15] P. W. Baier, "A critical review of CDMA," in *Proc. IEEE Veh. Tech. Conf. (VTC)*, Atlanta, USA, May 1996, pp. 6-10.

[16] S. Bannour and M. R. Azimi-Sadjadi, "Principal component extraction using recursive least squares learning," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 457-469, Mar. 1995.

[17] S. Bartelmaos, K. Abed-Meraim and S. Attallah, "Efficient and fast tracking algorithm for minor component analysis," in *Proc. IEEE Personal, Indoor and Mobile Radio Comms. Symp. (PIMRC)*, Helsinki, Finland, Sep. 2006, pp. 1-5.

[18] S. Bartelmaos, *Estimation des sous espaces et localisation des mobiles en UMTS*, PhD Thesis, ENST Paris & Université de Pierre et Marrie Curie, Jan. 2008, pp. 62-64.

[19] S. Bartelmaos, K. Abed-Meraim and S. Attallah, "Fast algorithms for minor component analysis," in *Proc. IEEE Workshop on Stat. Signal Proc. (SSP)*, Bordeaux, France, Jul. 2005, pp. 239-244.

[20] S. Bartelmaos, K. Abed-Meraim and S. Attallah, "Mobile localization using subspace tracking," in *Proc. Asia-Pacific Conf. Comms. (APCC)*, Perth, Australia, Oct. 2005, pp. 1009-1013.

[21] S. Bartelmaos and K. Abed-Meraim, "Principal and minor subspace tracking: Algorithms and stability analysis," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Toulouse, France, May 2006, pp. 560-563.

[22] S. E. Bensley and B. Aazhang, "Subspace-based channel estimation for code division multiple access communication systems," *IEEE Trans. Comms.*, vol. 44, no. 8, pp. 1009-1020, Aug. 1996.

[23] T. Chen, S. -I. Amari and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, pp. 385-390, Apr. 1998.

[24] T. Chen, Y. Hua and W. Y. Yan, "Global convergence of Oja's subspace algorithm for principal component extraction," *IEEE Trans. Neural Networks*, vol. 9, no. 1, pp. 58-67, 1998.

[25] T. Chen, S. -I. Amari and B. Murata, "Sequential extraction of minor components," *Neural Proc. Lett.*, vol. 13, pp. 195-201, 2001.

[26] T. Chen and S.-I. Amari, "Unified stabilization approach to principal and minor components extraction algorithms," *Neural Networks*, vol. 14, no. 10, pp. 1377-1387, Dec. 2001.

[27] A. Chevreuil, E. Serpedin, P. Loubaton and G. B. Giannakis, "Blind channel identification and equalization using periodic modulation precoders: Performance analysis," *IEEE Trans. Signal Proc.*, vol. 48, no. 6, pp. 1570 - 1586, Jun. 2000.

[28] T. Chonavel, B. Champagne and C. Riou, "Fast adaptive eigenvalue decomposition: A maximum likelihood approach," *Signal Proc.*, vol. 83, no. 2, pp. 307-324, Feb. 2003.

[29] A. Cichocki and S.-I. Amari, *Adaptive Blind Signal and Image Processing: Learning Algortihms and Applications*, J. Wiley, 2003, Chap. 3, 5.

[30] M. Clint and A. Jennings, "A simultaneous iteration method for the unsymmetric eigenvalue problem," *J. Inst. Math. Appl.*, vol. 8, pp. 111-121, 1971.

[31] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327-1343, Aug. 1990.

[32] J. Dehaene, *Continuous-time Matrix Algorithms, Systolic Algorithms and Adaptive Neural Networks*, Ph.D. Thesis, Katholieke Univ. Leuven, Belgium, Oct. 1995.

[33] J.-P. Delmas, "A complex adaptive eigensubspace algorithm for DOA or frequency estimation and tracking," in *Proc. European Signal Proc. Conf. (EUSIPCO)*, Brussels, Belgium, Aug. 1992, pp. 657-660.

[34] J.-P. Delmas, “Performances analysis of a Givens parameterized adaptive eigenspace algorithm,” *Signal Proc.*, vol. 68, pp. 87-105, 1998.

[35] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*, New York: J. Wiley, 1996, Chap. 4.

[36] K. Dogancay and R. A. Kennedy, “Least squares approach to blind channel equalization,” *IEEE Trans. Comms.*, vol 47, no. 11, pp. 1678-1687, Nov. 1999.

[37] S. C. Douglas, S. -Y. Kung and S. -I. Amari, “A self-stabilized minor subspace rule,” *IEEE Signal Proc. Lett.*, vol. 5, no. 12, pp. 328-330, Dec. 1998.

[38] S. C. Douglas and X. Sun, “Designing orthonormal subspace tracking algorithms,” in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Pacific Grove, USA, Oct. 2000, pp. 1441-1445.

[39] X. G. Doukopoulos and G. V. Moustakides, “Fast and stable subspace tracking,” *IEEE Trans. Signal Proc.*, vol. 56, no. 4, pp. 1452-1465, Apr. 2008.

[40] X. G. Doukopoulos, *Power Techniques for Blind Channel Estimation in Wireless Communication Systems*, PhD Thesis, IRISA-INRIA, University of Rennes 1, France, 2004, Appendix A.

[41] X. G. Doukopoulos and G. V. Moustakides, “The fast data projection method for stable subspace tracking,” in *Proc. European Signal Proc. Conf. (EUSIPCO)*, Antalya, Turkey, Sep. 2005.

[42] E. M. Dowling, L. P. Ammann and R. D. DeGroat, “A TQR-iteration based adaptive SVD for real angle and frequency tracking,” *IEEE Trans. Signal Proc.*, vol. 42, no. 4, pp. 914-926, Apr. 1994.

[43] C. J. Escudero, D. I. Iglesia, M. F. Bugallo and L. Castedo, “Analysis of a subspace channel estimation technique for multicarrier CDMA systems,” in *Proc. IEEE Workshop on Stat. Signal and Array Proc. (SSAP)*, Pocono Maner, USA, Aug. 2000, pp. 10-14.

[44] E. N. Frantzeskakis and K. J. R. Liu, “A class of square root and division free algorithms and architecture for QRD-based adaptive signal processing,” *IEEE Trans. Signal Proc.*, vol. 42, no. 9, pp. 2455-2469, Sep. 1994.

[45] Z. Fu and E. M. Dowling, “Conjugate gradient eigenstructure tracking for adaptive spectral estimation,” *IEEE Trans. Signal Proc.*, vol. 43, no. 5, pp. 1151-1160, May 1995.

[46] F. Gao, T. Cui and A. Nallanathan, “On channel estimation and optimal training design for amplify and forward relay networks,” *IEEE Trans. Wireless Comms.*, vol. 7, no. 5, pp. 1907-1916, May 2008.

[47] F. Gao, Y. Zeng, A. Nallanathan and T.-S. Ng, “Robust subspace blind channel estimation for cyslic prefixed MIMO OFDM systems: Algorithm, identifiability and performance analysis,” *IEEE J. Select. Areas in Comms.*, vol. 26, no. 2, pp. 378-388, Feb. 2008.

[48] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005, Chap. 1.

[49] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed, The John Hopkins University Press, 1996, Chap. 5, 8.

[50] R. Grover, D. A. Pados and M. J. Medley, “Subspace direction finding with an auxiliary-vector basis,” *IEEE Trans. Signal Proc.*, vol. 55, no. 2, pp. 758-763, Feb. 2007.

[51] S. Haykin and A. Steinhardt, *Adaptive Radar Detection and Estimation*, J. Wiley, 1992, Chap. 5.

[52] S. Haykin and M. Moher, *Modern Wireless Communications*, Pearson Prentice Hall, 2005, Chap. 1.

[53] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *Journal of Educational Psychology*, vol. 24, pp. 417-441, pp. 498-520, 1933.

[54] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim and Y. Miao, “A new look at the power method for fast subspace tracking,” *Digital Signal Proc.*, vol. 9, no. 4, pp. 297-314, Oct. 1999.

[55] Y. Hua, “Asymptotic orthonormalization of subspace matrices without square root,” *IEEE Signal Proc. Mag.*, vol. 21, no. 4, pp. 56-61, Jul. 2004.

[56] Y. Hua, K. Abed-Meraim and M. Wax, “Blind system identification using minimum noise subspace.” *IEEE Trans. Signal Proc.* vol. 45, no. 3, pp. 770-773, Mar. 1997.

[57] M. V. Jankovic and B. Reljin, “A new minor component analysis method based on Douglas-Kung-Amari minor subspace analysis method,” *IEEE Signal Proc. Lett.*, vol. 12, no. 12. pp. 859-862, Dec. 2005.

[58] M. V. Jankovic and H. Ogawa, "Time-oriented hierarchical method for computation of minor components," in *Proc. Intl. Conf. Adaptive and Natural Computing Algorithms*, Coimbra, Portugal, vol. 13, Mar. 2005, pp. 38-41.

[59] I.T. Jolliffe, *Principal Component Analysis*, 2ed, Springer, 2002, Chap. 1.

[60] W. Kang and B. Champagne, "A low-complexity adaptive blind subspace channel estimation algorithm," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Hong Kong, Apr. 2003, vol. 4, pp.6-10.

[61] Z. Kang, C. Chatterjee and V. P. Roychowdhury, "An adaptive quasi-Newton algorithm for eigensubspace estimation," *IEEE Trans. Signal Proc.*, vol. 48, no. 12, pp. 3328-3333, Dec. 2000.

[62] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust. Speech Signal Proc.*, vol. 34, no. 1, pp. 8-12, Feb. 1986.

[63] K. Karhunen, "Über lineare methoden in der wahrscheinlichkeitsrechnung," *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, vol. 37, pp. 1-79, 1947.

[64] E. Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, 1999, Chap. 7.

[65] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*, Springer, Berlin, 1978.

[66] W. Liu, W. Y. Yan, V. Sreeram and K. L. Teo, "Global convergence analysis for the NIC flow," *IEEE Trans. Signal Proc.*, vol. 49, no. 10, pp. 2422-2430, Oct. 2001.

[67] M. Loève, "Fonctions alèatoires de second ordre," *Processus Stochastiques et Mouvement Brownien*, 1948.

[68] Y. Lu, S. Attallah, G. Mathew and K. Abed-Meraim, "Propagation of orthogonality error for FRANS algorithm," in *Proc. Intl. Symp. Signal Proc. and its Appl. (ISSPA)*, Sharjah, United Arab Emirates, Feb. 2007, pp. 1-4.

[69] F. Luo, R. Unbehauen and A. Cichocki, "A minor component analysis algorithm," *Neural Networks*, vol. 10, no. 2, pp. 291-297, 1997.

[70] M. L. McCloud and L. L. Scharf, "A new subspace identification algorithm for high-resolution DOA estimation," *IEEE Trans. Antennas and Prop.*, vol. 50, no. 10, pp. 1382-1390, Oct. 2002.

[71] D. G. Manolakis, V. K. Ingle and S. M. Kogon, *Statistical and Adaptive Signal Processing*, McGraw-Hill, Boston, 2000, Chap. 8.

[72] G. Mathew, V. U. Reddy and S. Dasgupta, "Adaptive estimation of eigensubspace," *IEEE Trans. Signal Proc.*, vol. 43, no. 2, pp. 401-411, Feb. 1995.

[73] V. J. Mathews and Z. Xie, "Stochastic gradient adaptive filters with gradient adaptive step sizes," *IEEE Trans. Acous. Speech Signal Proc.*, vol. 41, no. 6, pp. 2075-2087, Jun. 1993.

[74] X. Mestre and J. R. Fonollosa, "ML approaches to channel estimation for pilot-aided multirate DS/CDMA systems," *IEEE Trans. Signal Proc.*, vol. 50, no. 3, pp. 696-709, Mar. 2002.

[75] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Proc.*, vol. 46, no. 7, pp. 1967-1979, Jul. 1998.

[76] E. Moulines, P. Duhamel, J.-F. Cardoso and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. Signal Proc.*, vol. 43, pp. 516-525, Feb. 1995.

[77] B. Muquet, M. de Courville and P. Duhamel, "Subspace-based blind and semi-blind channel estimation for OFDM systems," *IEEE Trans. Signal Proc.*, vol. 50, no. 7, pp. 1699-1712, Jul. 2002.

[78] E. Oja, "Principal component, minor component and linear neural networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.

[79] S. Ouyang, Z. Bao, G.-S. Liao and P. C. Ching, "Adaptive minor component extraction with modular structure," *IEEE. Trans. Signal Proc.*, vol. 49, no. 9, pp. 2127-2137, Sep. 2001.

[80] S. Ouyang, and Y. Hua, "Bi-iterative least-square method for subspace tracking," *IEEE Trans. Signal Proc.*, vol. 53, no. 8, pp. 2984-2996, Aug. 2005.

[81] S. Ouyang, P. C. Ching and T. Lee, "Robust adaptive quasi-Newton algorithms for eigensubspace estimation," *IEE Proc. Vis. Image Signal Proc.*, vol. 150, no. 5, pp. 321-330, Oct. 2003.

[82] N. L. Owsley, "Adaptive data orthogonalization," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Tulsa, USA, Apr. 1978, vol. 3, pp. 109-112.

[83] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559-572, 1901.

[84] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "Theory of spread spectrum communications: A tutorial," *IEEE Trans. Comms.*, vol. 30, no. 5, pp. 855-884, May 1982.

[85] J. G. Proakis, *Digital Communications*, McGraw-Hill, 2001, Chap. 15.

[86] V. U. Reddy, G. Mathew and A. Paulraj, "Some Algorithms for Eigensubspace Estimation Digital Signal Processing," *Digital Signal Processing*, vol. 5, no. 2, pp. 97-115, Apr. 1995.

[87] P. A. Regalia, "An adaptive unit norm filter with applications to signal analysis and Karhunen-Loeve transformations," *IEEE Trans. Circuits and Systems*, vol. 37, no. 5, pp. 646-649, May 1990.

[88] K. C. Sharman, "Adaptive algorithms for estimating the complete covariance eigenstructure," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, Tokyo, Japan, Apr. 1986, vol. 11, pp. 1401-1404.

[89] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. Signal Proc.*, vol. 45, no. 5, pp. 1222-1240, May 1997.

[90] P. Strobach, "Low-rank adaptive filters," *IEEE Tran. Signal Proc.*, vol. 44, no. 12, pp. 2932-2947, Dec. 1996.

[91] P. Strobach, "Fast recursive orthogonal iteration subspace tracking algorithms and applications," *Signal Proc.*, vol. 59, no. 1, pp. 73-100, May 1997.

[92] P. Strobach, "Square-root QR inverse iteration for tracking the minor subspace," *IEEE Trans. Signal Proc.*, vol. 48, no. 11, pp. 2994-2999, Nov. 2000.

[93] A. Taleb and G. Cirrincione, "Against the convergence of the minor component analysis neurons," *IEEE Trans. Neural Networks*, vol. 10, no. 1, pp. 207-210, Jan. 1999.

[94] T. Tanaka, "Generalized weighted rules for principal components tracking," *IEEE Trans. Signal Proc.*, vol. 53, no. 4, pp. 1243-1253, Apr. 2005.

[95] F. Theis, "Blind signal separation into groups of dependent signals using joint block diagonalization ," in *Proc. IEEE Intl. Symp. Circuits and Systems (ISCAS)*, Kobe, Japan, May 2005, pp. 5878-5881.

[96] L. Tong and S. Perreau, "Multichannel blind identification: From subspace to maximum likelihood methods," *Proc. IEEE*, vol. 86, no. 10, pp. 1951-1968, Oct. 1998.

[97] L. Tong, G. Xu and T. Kailath, "A new approach to blind identification and equalization of multipath channels," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Pacific Grove, USA, Nov. 1991, pp. 856-860.

[98] L. Tong, G. Xu and T. Kailath, "Blind channel identification based on second-order statistics: A frequency-domain approach," *IEEE Trans. Inform. Theory*, vol. 41, pp. 329-334, Mar. 1995.

[99] S. Visuri, H. Oja and V. Koivunen, "Subspace-based direction-of-arrival estimation using nonparametric statistics ," *IEEE Trans. Signal Proc.*, vol. 49, no. 9, pp. 2060- 2073, Sep. 2001.

[100] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, 1995, Chap. 1.

[101] X. Wang and H. V. Poor, "Blind multiuser detection: A subspace approach," *IEEE Trans. Inform. Theory*, vol. 36, no. 2, pp. 677-689, Mar. 1998.

[102] J. H. Wilkinson, "Error analysis of transformations based on the use of matrices of the form $\mathbf{I}-2\mathbf{w}\mathbf{w}^H$," *Error in Digital Computation*, vol. 2, Wiley, New York, 1965, pp. 77-101.

[103] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford, 1965, Chap. 3.

[104] G. Xu, H. Liu, L. Tong and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Proc.*, vol. 43, pp. 2982-2993, Dec. 1995.

[105] L. Xu, E. Oja and C. Y. Suen, "Modified Hebbian learning for curve and surface fitting," *Neural Networks*, vol. 5, no. 3, pp. 441-457, 1992.

[106] W. Y. Yan, U. Helmke and J. B. Moore, "Global analysis of Oja's flow for neural networks," *IEEE Trans. Neural Networks*, vol. 5, no. 5, pp. 674-683, 1994.

[107] B. Yang, "An extension of the PASTd algorithm to both rank and subspace tracking," *IEEE Signal Proc. Lett.*, vol. 2, no. 9, pp. 179-182, Sep. 1995.

[108] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms," *Signal Proc.*, vol. 50, pp. 123-136, 1996.

[109] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Proc.*, vol. 43, no. 1, pp. 95-107, Jan. 1995.

[110] J. Yang and M. Kaveh, “Adaptive eigensubspace algorithms for direction or frequency estimation and tracking,” *IEEE Trans. Acoust. Speech Signal Proc.*, vol. 36, no. 2, pp. 241-251, Feb. 1988.

[111] Y. Yoon, L. M. Kaplan and J. H. McClellan, “TOPS: new DOA estimator for wideband signals,” *IEEE Trans. Signal Proc.*, vol. 54, no. 6, pp. 1977-1989, Jun. 2006.

Appendix A

Appendices to Chapter 4

A.1 Gradient Adaptive Step-size Method with Real-valued Data

If the received data source is real-valued, i.e. $\tilde{r}_k$, $\tilde{r}_l$ and $\tilde{r}_m$ are independent and zero mean real-valued Gaussian random variables, equation (4.8) now becomes

$$E\left(\tilde{r}_k \tilde{r}_l^* |\tilde{r}_m|^2\right) = \begin{cases} 0 & \text{for } k \neq l \\ \lambda_k \lambda_m \delta(k-l)\left[1 - \delta(l-m)\right] & \text{for } k = l, l \neq m \\ 3\lambda_k^2 \delta(k-l)\delta(k-m) & \text{for } k = l = m. \end{cases} \tag{A.1}$$

Based on (4.7) and (A.1), the (k,k)th element of $\mathbf{G}$ is of the form

$$g_{k,k} = \lambda_k \sum_{m=1}^{N} \lambda_m^2 + 2\lambda_k^3, \tag{A.2}$$

and $g_{k,l} = 0$ if $k \neq l$. Therefore, according to (4.6), we can get

$$\mathrm{Tr}\left(\mathbf{W}_o^H E\left[\mathbf{R}(i)\right]\mathbf{W}_o\right) = \sum_{k=1}^{P}\sum_{m=1}^{N} \lambda_k \lambda_m^2 + 2\sum_{k=1}^{P} \lambda_k^3. \tag{A.3}$$

The bounds of μ_R are found to be

$$0 < \mu_R < \frac{1}{4\left(\sum_{k=1}^{P}\sum_{m=1}^{N} \lambda_k \lambda_m^2 + 2\sum_{k=1}^{P} \lambda_k^3\right)}. \tag{A.4}$$

Using (4.14), a practical bound of $\mu_R(i)$ with real-valued received data can be obtained as

$$0 < \mu_R(i) < \frac{1}{4(\theta_1(i)\theta_3(i) + 2\theta_2(i))}. \tag{A.5}$$

The step-size adaptation steps with real-valued received data are the same as complex-valued received data as in (4.17).

A.2 Optimal Step-size with Real-valued Data

If the received data are real-valued, following (A.3) and (4.18), we have the theoretical optimal step-size as

$$E[\beta_{o,R}(i-1)] = \frac{\sum_{k=1}^{P} \lambda_k^2}{2\sum_{k=1}^{P} \lambda_k \left(2\lambda_k^2 + \sum_{m=1}^{N} \lambda_m^2\right)}, \tag{A.6}$$

Thus, the estimated optimal step-size for HFRANS with real-valued received data at each instant is obtained as

$$\beta(i)_R = \frac{\theta_4(i)}{2\left(2\theta_2(i) + \theta_1(i)\theta_3(i)\right)}. \tag{A.7}$$

Appendix B

Appendices to Chapter 5

B.1 Mathematical Equivalence of Eq. (5.5b) and Eq. (5.6)

To prove that (5.5b) can be rewritten as (5.6), we need to prove that $\bar{\mathbf{p}}(i)\bar{\mathbf{y}}^H(i) = 2\bar{\mathbf{p}}(i)\bar{\mathbf{p}}^H(i)\mathbf{W}(i)/\|\bar{\mathbf{p}}(i)\|^2$. Assuming $\mathbf{W}^H(i-1)\mathbf{W}(i-1) = \mathbf{I}_P$ and $\mathbf{p}^H(i)\mathbf{W}(i-1) = \mathbf{0}$, we can write $\bar{\mathbf{p}}(i)\bar{\mathbf{p}}^H(i)\mathbf{W}(i)$ as

$$\begin{aligned}\bar{\mathbf{p}}(i)\bar{\mathbf{p}}^H(i)\mathbf{W}(i) &= \bar{\mathbf{p}}(i)\left[-\bar{\tau}(i)\mathbf{W}(i-1)\bar{\mathbf{y}}(i) + (1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)\mathbf{p}(i)\right]^H \mathbf{W}(i)\\ &= -\bar{\tau}(i)\bar{\mathbf{p}}(i)\bar{\mathbf{y}}(i)\end{aligned} \tag{B.1}$$

Then we only need to prove $\|\bar{\mathbf{p}}(i)\|^2 = -2\bar{\tau}(i)$. We can write $\|\bar{\mathbf{p}}(i)\|^2$ as

$$\|\bar{\mathbf{p}}(i)\|^2 = \bar{\tau}^2(i)\|\bar{\mathbf{y}}(i)\|^2 + (1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)^2\|\mathbf{p}(i)\|^2,$$

and in view of (5.4) we can write

$$(1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)^2 = \frac{1}{1+\|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2}. \tag{B.2}$$

Therefore, $\|\bar{\mathbf{p}}(i)\|^2$ is equal to

$$\begin{aligned}\|\bar{\mathbf{p}}(i)\|^2 &= \bar{\tau}^2(i)\|\bar{\mathbf{y}}(i)\|^2 + \frac{\|\mathbf{p}(i)\|^2}{1+\|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2}\\ &= \bar{\tau}^2(i)\|\bar{\mathbf{y}}(i)\|^2 + \frac{1}{\|\bar{\mathbf{y}}(i)\|^2}\left(1-\frac{1}{1+\|\mathbf{p}(i)\|^2\|\bar{\mathbf{y}}(i)\|^2}\right)\end{aligned}$$

$$\begin{aligned}&=\frac{\bar{\tau}^2(i)\|\bar{\mathbf{y}}(i)\|^4+1-(1+\bar{\tau}(i)\|\bar{\mathbf{y}}(i)\|^2)^2}{\|\bar{\mathbf{y}}(i)\|^2}\\&=-2\bar{\tau}(i).\end{aligned}\tag{B.3}$$

B.2 Proof of Lemma 5.1

B.2.1 Proof of Lemma 5.1 with Complex-valued Data

Using (5.7), we can obtain

$$\left.\begin{array}{l}\mathbf{u}^H(i)\mathbf{v}(i)=|\mathbf{u}^H(i)\mathbf{v}(i)|e^{-j\alpha(i)}\\ \mathbf{v}^H(i)\mathbf{u}(i)=|\mathbf{u}^H(i)\mathbf{v}(i)|e^{j\alpha(i)}\\ \|\mathbf{a}(i)\|^2=2\|\mathbf{u}(i)\|^2-2|\mathbf{u}^H(i)\mathbf{v}(i)|.\end{array}\right\}\tag{B.4}$$

Using (5.7) and (5.8), we have the RHS of (5.9) as

$$\begin{aligned}\mathbf{H}_P(i)\mathbf{u}(i)&=\left[\mathbf{I}_P-2\frac{\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}\right]\mathbf{u}(i)\\&=\frac{\|\mathbf{a}(i)\|^2\mathbf{u}(i)-2\mathbf{a}(i)\mathbf{a}^H(i)\mathbf{u}(i)}{\|\mathbf{a}(i)\|^2}\\&=\frac{\|\mathbf{a}(i)\|^2\mathbf{u}(i)-2\left[\mathbf{u}(i)-\mathbf{v}(i)e^{j\alpha(i)}\right]\left[\mathbf{u}^H(i)-\mathbf{v}^H(i)e^{-j\alpha(i)}\right]\mathbf{u}(i)}{\|\mathbf{a}(i)\|^2}\\&=\frac{\|\mathbf{a}(i)\|^2-2\|\mathbf{u}(i)\|^2+2\mathbf{v}^H(i)\mathbf{u}(i)e^{-j\alpha(i)}}{\|\mathbf{a}(i)\|^2}\mathbf{u}(i)\\&\quad-2\frac{\mathbf{v}^H(i)\mathbf{u}(i)-\|\mathbf{u}(i)\|^2e^{j\alpha(i)}}{\|\mathbf{a}(i)\|^2}\mathbf{v}(i).\end{aligned}\tag{B.5}$$

By using (B.4) in (B.5), we can obtain

$$\mathbf{H}_P(i)\mathbf{u}(i)=\mathbf{v}(i)e^{j\alpha(i)},\tag{B.6}$$

which proves Lemma 5.1 with complex-valued data.

B.2.2 Proof of Lemma 5.1 with Real-valued Data

Using (5.10), we can obtain

$$\left.\begin{array}{l}\mathbf{u}^H(i)\mathbf{v}(i)=\mathbf{v}^H(i)\mathbf{u}(i)\\ \|\mathbf{a}(i)\|^2=2\|\mathbf{u}(i)\|^2-2\mathbf{u}^H(i)\mathbf{v}(i).\end{array}\right\}\tag{B.7}$$

Using (5.10) and (5.11), we have the RHS of (5.12) as

$$\begin{aligned}\mathbf{H}_P(i)\mathbf{u}(i) &= \left[\mathbf{I}_P - 2\frac{\mathbf{a}(i)\mathbf{a}^H(i)}{\|\mathbf{a}(i)\|^2}\right]\mathbf{u}(i)\\ &= \frac{\|\mathbf{a}(i)\|^2\mathbf{u}(i) - 2\mathbf{a}(i)\mathbf{a}^H(i)\mathbf{u}(i)}{\|\mathbf{a}(i)\|^2}\\ &= \frac{\|\mathbf{a}(i)\|^2\mathbf{u}(i) - 2\left[\mathbf{u}(i) - \mathbf{v}(i)\right]\left[\mathbf{u}^H(i) - \mathbf{v}^H(i)\right]\mathbf{u}(i)}{\|\mathbf{a}(i)\|^2}\\ &= \frac{\|\mathbf{a}(i)\|^2 - 2\|\mathbf{u}(i)\|^2 + 2\mathbf{v}^H(i)\mathbf{u}(i)}{\|\mathbf{a}(i)\|^2}\mathbf{u}(i) - 2\frac{\mathbf{v}^H(i)\mathbf{u}(i) - \|\mathbf{u}(i)\|^2}{\|\mathbf{a}(i)\|^2}\mathbf{v}(i).\end{aligned} \tag{B.8}$$

By using (B.7) in (B.8), we can obtain

$$\mathbf{H}_P(i)\mathbf{u}(i) = \mathbf{v}(i), \tag{B.9}$$

which proves Lemma 5.1 with real-valued data.

Appendix C

Appendices to Chapter 6

C.1 Derivation of (6.28)

Using (6.23) in (6.27), we obtain

$$\begin{aligned}
\left.\frac{\mathrm{d}\left(\mathbf{W}^H\mathbf{W}\right)}{\mathrm{d}t}\right|_{\mathbf{W}=\hat{\mathbf{W}}+\delta\mathbf{W}(t)} =& -\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W}+\mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}\\
&-\mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W}+\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W}\\
=&\,\mathbf{X}_1+\mathbf{X}_2+\mathbf{X}_3+\mathbf{X}_4,
\end{aligned} \tag{C.1}$$

where $\mathbf{W}^H\mathbf{W}=\mathbf{I}_P$ and

$$\mathbf{X}_1=-\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W},\quad \mathbf{X}_2=\mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W} \tag{C.2}$$

$$\mathbf{X}_3=-\mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{W},\quad \mathbf{X}_4=\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{C}\mathbf{W}. \tag{C.3}$$

Since $\mathbf{X}_1=\mathbf{X}_3^H$ and $\mathbf{X}_2=\mathbf{X}_4^H$, we only need to analyze $\mathbf{X}_1$ and $\mathbf{X}_2$.

Let $\mathbf{H}=\mathbf{W}^H\mathbf{C}\mathbf{W}$ and keeping only first-order variation terms, we have

$$\begin{aligned}
\mathbf{X}_1 =& -\left(\hat{\mathbf{W}}+\delta\mathbf{W}(t)\right)^H\left(\hat{\mathbf{W}}+\delta\mathbf{W}(t)\right)\left(\hat{\mathbf{W}}+\delta\mathbf{W}(t)\right)^H\left(\hat{\mathbf{W}}+\delta\mathbf{W}(t)\right)\mathbf{H}\\
\approx& -\left(\mathbf{I}_P+\hat{\mathbf{W}}^H\delta\mathbf{W}(t)+\delta\mathbf{W}^H(t)\hat{\mathbf{W}}\right)\left(\mathbf{I}_P+\hat{\mathbf{W}}^H\delta\mathbf{W}(t)+\delta\mathbf{W}^H(t)\hat{\mathbf{W}}\right)\mathbf{H}.
\end{aligned} \tag{C.4}$$

Since $\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\approx\hat{\mathbf{W}}^H\delta\mathbf{W}(t)+\delta\mathbf{W}^H(t)\hat{\mathbf{W}}$ because of (6.25), we obtain from (C.4)

$$\mathbf{X}_1\approx-\left(\mathbf{I}_P+\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right)\left(\mathbf{I}_P+\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right)\mathbf{H}$$

$$\begin{aligned}
&\approx -\left(\mathbf{I}_P + 2\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right)\mathbf{H} \\
&\approx -\mathbf{H} - 2\delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}. \qquad \text{(C.5)}
\end{aligned}$$

Similarly, we have $\mathbf{X}_2$ as

$$\begin{aligned}
\mathbf{X}_2 &= \mathbf{H}\left(\hat{\mathbf{W}} + \delta\mathbf{W}(t)\right)^H\left(\hat{\mathbf{W}} + \delta\mathbf{W}(t)\right) \\
&\approx \mathbf{H}\left(\mathbf{I}_P + \hat{\mathbf{W}}^H\delta\mathbf{W}(t) + \delta\mathbf{W}^H(t)\hat{\mathbf{W}}\right) \\
&\approx \mathbf{H}\left(\mathbf{I}_P + \delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\right) \\
&\approx \mathbf{H} + \hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\delta\left(\mathbf{W}^H\mathbf{W}\right)(t). \qquad \text{(C.6)}
\end{aligned}$$

Using $\mathbf{X}_1 = \mathbf{X}_3^H$ and $\mathbf{X}_2 = \mathbf{X}_4^H$, we have

$$\mathbf{X}_3 = -\mathbf{H} - 2\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}\delta\left(\mathbf{W}^H\mathbf{W}\right)(t) \qquad \text{(C.7)}$$

$$\mathbf{X}_4 = \mathbf{H} + \delta\left(\mathbf{W}^H\mathbf{W}\right)(t)\hat{\mathbf{W}}^H\mathbf{C}\hat{\mathbf{W}}. \qquad \text{(C.8)}$$

Using (C.5), (C.6), (C.7) and (C.8) in (C.1), we obtain (6.28).