ADAPTIVE ALGORITHMS FOR WIRELESS CHANNEL ESTIMATION: TRANSIENT ANALYSIS AND SEMI-BLIND DESIGN

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Tareq Y. Al-Naffouri
January 2005

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Arogyaswami Paulraj
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

George Papanicolaou

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ali H. Sayed

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ahmad Bahai

Approved for the University Committee on Graduate Studies.

Abstract

Accurate channel state information is important in communication systems. This is especially challenging in a wireless environment where the channel exhibits strong frequency and time selectivity. The literature is full with ingenious works devoted to the design and analysis of algorithms for channel estimation. In general, these works have approached the various algorithms distinctly obscuring commonalities that might exist among them. This dissertation presents two contributions related to the analysis and design of adaptive channel estimation algorithms.

The first part of the dissertation performs an analysis of a large class of adaptive algorithms for channel estimation. Adaptive filters are, by design, time-variant, nonlinear, and stochastic systems. For this reason, it is common to study different adaptive schemes separately due to the differences that exist in their update equations. The dissertation presents a unified approach to the analysis of adaptive filters that employ general data or error nonlinearities. In addition to deriving earlier results in a unified manner, the approach presented also leads to new stability and performance results without imposing restrictions on the color or statistics of the input sequence.

The second part of the dissertation presents an expectation-maximization (EM) based class of algorithms for joint channel and data recovery in OFDM (orthogonal frequency division multiplexing). The algorithms make use of the rich structure of the underlying communication problem– a structure induced by the data and channel constraints. These constraints include pilots, the cyclic prefix, the code, and the finite alphabet constraints on the data; sparsity, finite delay spread, and the statistical properties of the channel (time and frequency correlation). The algorithms become progressively more sophisticated as more

data and channel constraints are incorporated, with each new version of the algorithm subsuming the previous version as a special case, culminating in an EM-based forward backward Kalman filter. The dissertation finally scales up the algorithm design to support OFDM transmission over multiple-input multiple-output (MIMO) systems.

Acknowledgements

I reserve most thanks and appreciation to GOD the Almighty for His countless blessings. I pray that this work and the growth I had at Stanford will be used in His cause.

I would like to express my gratitude and appreciation to my Advisor, Prof. Arogyaswami Paulraj for his guidance and support throughout my Stanford experience. I thank him for giving me the freedom to explore various research paths, fruitful or not. I have learned immensely from his theoretical and practical knowledge.

It was a true honor to work with my co-advisor, Prof. Ali H. Sayed at UCLA. He held me to a very high standard, and pushed me very hard from time to time. In spite of the geographical distance, I have learned a lot from his deep knowledge, discipline, and intuition. I would also like to thank him for hosting me in his lab several times and for flying to Stanford to attend my thesis defense.

My thanks also go to Prof. A. Bahai for his supervision and guidance. I have learned a lot from his industrial experience during the adaptive wireless systems course and during the two internships I had at National Semiconductor and beyond.

My appreciation also goes to Prof. George Papanicolaou for agreeing to be on my oral and reading committee at a very short notice.

I would like to thank the members of the Smart Antenna Research Group at Stanford for their company and for the valuable discussions we had together.

The administrative assistance of Denise Murphy, Gina Masterantonio, Diane Shankle,

Kathleen Kombo, and Keith Gaul was very valuable. I take this opportunity to thank them and thank the other staff members in the Electrical Engineering Department.

I would like to express my deep gratitude and appreciation to my undergraduate school, King Fahd University of Petroleum and Minerals, Daharan, Saudi Arabia, for the solid education it gave me and for the generous financial support of my graduate education at Stanford. That was truly invaluable.

It is not an exaggeration to say that my time at Stanford has so far been the best experience in my life. This is in large part due to the great friends I met here: Ahmad Alyamani, Ali Abbas, David Walker, Dimitris Toumpakaris, Ghazi Alrawi, Husam Abu Haimed, Haitham Boshnak, Haitham Hindi, Ibrahim Almojel, Isabel Awad, Khaled Salama, Mohammad Alghahtani, Mohammad Alrawashdeh, Mohammad Sharawi, Musab Mutia-Al-Rahman, Naomi Brown, Sulaiman Albalawi, Thea Beradze, Wonil Roh, Yasser Djazaerly, and Yasser Taima. I thank all of them (and many more whom I forgot to mention) for their love and encouragement and for dragging me out of the academic work. All the good time we had together will remain a valuable part of my memory

Finally, my thanks go to my mother and father whose sacrifice, blessing, and good wishes have brought me to this stage. They worked hard to stay close when I moved far away. I am also deeply indebted to my sister Nowar and to my brother Bassel for their friendship, support, and encouragement. Their indirect contributions to my life at Stanford are too numerous to delineate here.

To my Mother and Father

Contents

List of Tables

List of Figures

Adaptive Algorithms for Wireless Channel Estimation: Transient Analysis and Semi-Blind Design

Tareq Y. Al-Naffouri

January 5, 2005

Chapter 1

Introduction

The radio revolution started with the invention of the telegraph by Guglielmo Marconi over 100 years ago. Today, and as we enter the new millennium, we notice a rapid growth in wireless network technology and a convergence of voice, data, and video technology. This is creating new services at lower costs and resulting in increased number of users, air-time usage and revenues, which are increasing at 40% per year.

The wireless channel, however, can significantly limit the performance of such a system. The transmitted signal suffers from propagation loss, macroscopic fading due to blocking, and macroscopic fading due to the movement of the transmitter, the receiver, or objects in the environment. This makes channel estimation a critical part of the wireless receiver.

This dissertation explores the analysis and design of adaptive and iterative algorithms for the estimation of linear and time-invariant wireless channels. The first –the analysis– part of the dissertation presents a unified analysis of a large class of adaptive algorithms that have been extensively used for channel estimation. The second –the design– part of the dissertation derives an iterative algorithm for channel estimation [1] that makes an intelligent use of the constraints that underly the communication problem.

This introductory chapter sets the stage for the dissertation. It starts by elaborating on the nature of the wireless channel. The chapter then describes the convenience of modelling the channel as a linear time-invariant system and the usefulness of employing adaptive and iterative algorithms for its estimation. The chapter then presents an overview of available adaptive filtering algorithms and previous work on channel estimation. This provides a

[1] We confine our attention to OFDM (orthogonal frequency division multiplexing) systems but the results apply in more general context.

context for the analysis and design parts of the dissertation. The chapter concludes by laying out the contributions of the subsequent chapters which also serve to outline the dissertation organization.

1.1 The Wireless Channel

A signal propagates in a wireless channel along a number of different paths. These paths arise from scattering, reflection, refraction, and drift by objects in the environment. As the signal propagates, its power drops due to three effects: propagation loss, macroscopic fading, and microscopic fading. We address each of these effects in the following. [73]

Path Loss

The power of the transmitted signal is inversely proportional to the square of the distance it travels. This reduction in power is caused by water absorption, foliage, and ground reflection. Path loss thus causes a steady drop in signal power. In contrast, macroscopic and microscopic fading (described next) cause the signal power to fluctuate.

Macroscopic Fading

Macroscopic fading results in long-term (or long-distance) variation in the signal level. It results from blockage effects caused by buildings and other natural features.

Microscopic Fading

Macroscopic fading causes short-term or rapid fluctuations of the received signal. Specifically, objects between the transmitter and receiver scatter the signal in different directions. Fading results from the superposition of a large number of these scattered components, and by the central limit theorem, the resulting signal can be assumed to be an independent Gaussian process. The signal will be be zero mean unless there is a direct line of sight (LOS) between the transmitter and receiver. Microscopic fading affects the signal in time, frequency, and space, as we explain next.

Doppler spread–time selective fading Time selective fading happens due to the movement of scatterers, the transmitter, or the receiver. In this type of fading, a tone at frequency ν_c spreads over a finite bandwidth ($\nu_c \pm \nu_{\max}$), creating a Doppler power

spectrum (the Fourier transform of the time correlation of the channel response to a CW wave). Time selective fading is measured by the coherence time-width which is the time lag at which the signal autocorrelation reduces to 0.7 of its peak value.

Delay spread–frequency selective fading As we mentioned earlier, the signal in a wireless channel travels along multiple paths. Scaled and time shifted versions of the signal arrive to the receiver along these paths. This delay spread results in frequency selective fading as the channel acts as a tapped-delay line. Frequency selective fading is characterized in terms of the coherence bandwidth, which is the frequency lag at which the channel autocorrelation drops to 0.7 of its peak value.

Angle spread–space selective fading Angle spread at the receiver refers to the spread of angles of arrivals of the multi-path components at the receiver (a similar definition can be made for the angle spread at the transmitter). Angle spread results in space selective fading. This type of fading is characterized by the coherence distance, which is the distance lag at which the channel autocorrelation function drops to 0.7 of its peak value.

1.2 Linear Time-Invariant Models

As noted above, we concentrate in this dissertation on linear and time-invariant (LTI) models for two reasons: [11]

Simplicity of description: LTI systems are easier to describe mathematically than time-variant systems. An LTI system is completely characterized by its (one-dimensional) impulse response. When the system becomes time-variant, however, the impulse response becomes 2-dimensional and hence more difficult to deal with.

Availability of powerful signal processing techniques: The eigenvalues of LTI systems are complex exponentials, and the corresponding eigenvalues are given by the frequency domain transfer function. This means that time-invariant systems always commute (while time-invariant systems do not). These and other properties of linear systems allow the use of powerful signal processing techniques such as the Fourier transform and its discrete implementation.

LTI systems also have significant signal processing applications. This includes linear prediction, acoustic and line echo cancellation, channel estimation, receiver design for CDMA

Table 1.1: *Common channel and data constraints used for channel estimation*

CONSTRAINTS	ASSUMPTIONS	REFERENCE
Data Constraints	**Finite alphabet constraint**	[88], [7]
	Code	[9], [80]
	Transmit precoding (e.g., cyclic prefix, silent gaurad band)	[20], [44], [40], [68], [52], [99], [9]
	Pilots	[21], [56], [71],[95],[72],[53]
Channel Constraints	**Finite delay spread**	[20], [9], [71]
	Sparsity: Channel has a few active taps	[102], [48], [97]
	Frequency correlation: Taps are Gaussian distributed	[54],[32], [9], [79]
	Time correlation	[94], [50], [45], [1], [58]
	Uncertainty information	[81], [54]

and OFDM transmission. Moreover, time-invariance is not as limiting as it sounds because we can still use time-invariant systems to model systems that vary with time on a block-by-block manner. For example, in a wireless mobile environment with reasonable vehicle speeds, we can assume that the channel response is approximately constant over a small time window.

1.3 Leverages for Channel Estimation

All algorithms for channel estimation rely on some inherent structure of the communication problem to perform channel (and data) recovery. This structure is created by constraints on the transmitted data or constraints on the channel itself, as described below (see also Table 1.1):

Data Constraints:

Finite alphabet constraint: Data is usually drawn from a finite alphabet [88], [7].

Code: Data usually exhibits some form of redundancy, such as a channel code that helps reduce the probability of error [9], [80].

Transmit precoding: The data might also contain some form of precoding (to facilitate equalization at the receiver) such as a cyclic prefix [44], silent guard bands [40], [68] and known symbol precoding [52].

Pilots: Pilots represent the most primitive form of redundancy and are usually inserted to perform channel estimation or simply to initialize the estimation process [21], [56], [71],[95],[72],[53].

Channel Constraints:

Finite delay spread: The channel impulse response has a finite duration [20], [9], [71]. [2]

Sparsity: The channel impulse response is usually sparse, dominated by a few strong paths at some delays with a near zero arrivals at other delays [102], [48], [97]. The number of paths and their delays are usually stationary. However, their amplitudes and relative phases usually vary much more rapidly with time. This essentially reduces the number of parameters to be estimated to that of the number of multipaths in the channel.

Frequency correlation: In addition to the information about which of the channel taps are inactive, we usually have additional statistical information about the active ones. This is usually captured by the frequency correlation of the frequency response [54],[32], [9], [79].

Time correlation: As channels vary with time, they exhibit some form of time correlation. The time-variant behavior could also be more structured, e.g., following a state-space model [94], [50], [45], [1], [58]. [3]

Uncertainty information: Channels also suffer from non-ideal effects such as nonlinearities and rapid time-variations that are difficult to model. The aggregate effect of this nonideal behavior could be represented as uncertainty information that can be used to build robust receivers (e.g., as in [81], [54]).

1.3.1 Channel Estimation Techniques

Several algorithms were suggested in literature for channel and data recovery for OFDM (orthogonal frequency division multiplexing). Each of these algorithms makes use of a subset of the constraints above. We summarize in what follows some of the main approaches for channel estimation (or generally, receiver design).

[2]This information is even assumed available at the transmitter as this knowledge is essential in designing the transmitted sequence, e.g., the length of the cyclic prefix in OFDM.

[3]By speaking of time correlation separately from frequency correlation, we are basically assuming that the channel autocorrelation function is separable (i.e., a product of time and frequency functions). While this might not be exact, it is much more convenient.

Training-Based Estimation

Assuming that the channel is time-invariant, pilots are transmitted along with data symbols and are used for channel estimation (see [71], [72], [56], [95], [53]). The number of pilots needed in the noiseless case is equal to the length of the impulse response (In the presence of noise, however, more pilots are needed to improve the estimation accuracy).

Blind Estimation

Blind algorithms rely completely on natural constraints underlying the communication problem to perform channel recovery. For example, [9] used frequency correlation, [9] and [79] used the outer code, [20] used the cyclostationarity induced by the cyclic prefix and transmitter precoding, [108] used a subspace constraint, [68] and [44] used the cyclic prefix.

Semi-Blind Estimation

Semi-blind techniques are a hybrid of blind and training based techniques, utilizing pilots and other natural constraints to perform channel estimation. Thus, in addition to pilots, [99], [9], and [8] used the cyclic prefix, [9] and [32] used frequency correlation, [94], [50], [45], [1], [58], [80] used frequency and time correlation, and [9], [10] , [65] and [80] used the outer code.

Data-Aided Channel Estimation

The main and perhaps the only reason to perform channel estimation is to use the estimate along with the channel output to recover the transmitted data. One can, in turn, use the detected data to enhance the channel estimate giving rise to an iterative technique for channel and data recovery. With this in mind, it is natural for the two operations, of channel and data recovery, to be considered jointly, especially since one operation can be used to enhance the performance of the other. This intuitive idea is the basis of joint channel estimation and data detection proposed in [50], [45], [57], [39]. Other works, like [2], [58], [24], [55], [101], and [7], arrived at iterative techniques more rigorously by employing the expectation-maximization (EM) algorithm. The data-aided approach seems the most sensible for channel estimation. However, just like many other works on channel estimation, the aforementioned works utilize only a subset of the constraints on the channel and the data.

The designer can implement most of the techniques above using batch processing or using adaptive/iterative techniques. The latter approach has clear advantages as we discuss below.

1.4 Why Make Channel Estimation Adaptive/Iterative?

There are several advantages to estimating a channel adaptively or iteratively. First, the optimum estimator might be too prohibitively complex to implement directly. The designer can alternatively implement the same algorithm adaptively or iteratively, which makes it possible to trade off complexity for speed. One such example is the recursive implementation of a least-squares problem, using the recursive least-squares (RLS) algorithm.

Secondly, complexity might be so much an issue that the designer would be willing to trade performance for lower complexity. The least-mean squares (LMS) algorithm and its family offer this flexibility. For example, some least-squares problems can be approximately solved using the LMS algorithm through some form of stochastic approximation.

Thirdly, it is easier to modify an adaptive algorithm to incorporate an additional constraint about a given problem than it is to modify the corresponding batch implementation of the same problem. For example, the RLS implementation of a least-squares channel estimation problem can be easily modified to deal with the case where the channel is varying slowly with time.

Finally, a closed form solution might not even be available and so an adaptive or iterative implementation might be the best solution that the designer can provide. A typical example is joint channel and data recovery from received data in a communication problem, where the only practical possibility is iterative recovery.

1.5 Overview of Available Adaptive Algorithms

Many adaptive algorithms have been proposed in literature. These algorithms can be classified into three main categories [82], [43].

The LMS algorithm and its family: The least-mean squares (LMS) algorithm and its family are one of the most important and ubiquitous classes of adaptive filters. They are usually built around transversal (taped-delay line) structures and are well known

for their simplicity, low complexity, and robustness. More details about this class of algorithms can be found in Chapters 2 and 3.

Recursive least squares (RLS) adaptive filter: As its name suggests, the RLS is a recursive implementation of the least-squares problem. This adaptive filter overcomes some limitations of the LMS family of filters as it is an order of magnitude faster and its behavior is insensitive to the color of the input. This is, however, achieved at a higher computational coast and an inherent computational instability.

The Kalamn filter: The Kalman filter is a generalization of the RLS filter and is of comparable complexity. It emphasizes the notion of a state and hence is more suited to time-variant environments.

These filters are usually built around transversal (taped-delay line) filters. Some adaptive filters are built around an infinite impulse response, but the choices here are much more limited and the filters involved are much less understood.

1.6 Motivation of this Work

The motivation of this thesis is to develop a unified framework to the problem of adaptive channel estimation. In particular, this thesis considers two aspects of this problem:

Analysis: There is extensive literature on the performance of adaptive filters with many ingenious results and approaches (see Chapters 2 and 3 and the references therein). However, most of these works study individual algorithms separately. Moreover, each study uses its own set of assumptions and approximations that fit the specific class of algorithms that it studies. This is because different adaptive schemes have different nonlinear update equations, and the particularities of each case tend to require different arguments and assumptions. The first objective of this thesis is thus to provide a unified analysis of adaptive algorithms with data and error nonlinearities.

Design: There are many algorithms proposed in the literature for channel estimation and equalization. Each of these algorithms rely on a particular set of assumption or a priori information on the channel or transmitted data. The second objective of the thesis is to provide a unified channel estimation method for the single and multiple antenna cases.

Although the thesis demonstrates the design for OFDM transmission, the design principles apply to a more general context.

1.7 Overview of Contributions

1.7.1 Chapter 2

In Chapter 2, we present a framework for the transient analysis of adaptive filters with general data nonlinearities (both scalar-valued and matrix-valued). The framework relies on energy conservation arguments and avoids the need for explicit recursions for the covariance matrix of the weight-error vector. Among other results, the derivation characterizes the transient behavior of such filters in terms of a linear time-invariant state-space model. In addition to deriving earlier results in a unified manner, the approach leads to stability and performance results without restricting the regression data to being Gaussian or white.

1.7.2 Chapter 3

In Chapter 3, we extend the same energy-based approach of Chapter 2 to the transient analysis of adaptive filters with error nonlinearities. This class of algorithms is among the most difficult to analyze, and it is not uncommon to resort to different methods and assumptions with the intent of performing tractable analyses for any member of this class. In addition to deriving earlier results in a unified manner, the approach also leads to new performance results without restricting the regression data to being Gaussian or white and without relying on any linearization arguments.

1.7.3 Chapter 4

In Chapter 4, we propose an expectation-maximization (EM) class of algorithms for joint channel and data recovery in OFDM. The algorithms make use of the rich structure of the underlying communication problem–a structure induced by the data and channel constraints. These constraints include pilots, the cyclic prefix, and the finite alphabet constraints on the data, and sparsity, finite delay spread, and the statistical properties of the channel (frequency and time correlation). The algorithms become progressively more sophisticated as more data and channel constraints are incorporated, with each new version of the algorithm subsuming the previous version as a special case, culminating in an EM-based forward-backward Kalman filter.

1.7.4 Chapter 5

In Chapter 5, we build on the previous chapter and scale up the receiver design to OFDM transmission over mutli-input multi-output (MIMO) systems. The receiver makes use of the rich structure of the underlying communication problem. Thus, in addition to aforementioned channel and data constraints, the receiver also makes use of the spatial correlation and of the space-time code used.

Finally, Chapter 6 overviews the thesis and point out some future directions of research.

Chapter 2

Transient Analysis of Data-Normalized Adaptive Filters

2.1 Introduction

[1] Adaptive filters are, by design, time-variant and nonlinear systems that adapt to variations in signal statistics and that learn from their interactions with the environment. The success of their learning mechanism can be measured in terms of how fast they adapt to changes in the signal characteristics, and how well they can learn given sufficient time (e.g., [100, 43, 60]). It is therefore typical to measure the performance of an adaptive filter in terms of both its transient performance and its steady-state performance. The former is concerned with the stability and convergence rate of an adaptive scheme, while the latter is concerned with the mean-square error that is left in steady-state.

There have been extensive works in the literature on the performance of adaptive filters with many ingenious results and approaches (e.g., [100, 43, 60, 46, 36, 63, 92, 34, 27, 76, 90]). However, it is generally observed that most works study individual algorithms separately. This is because different adaptive schemes have different nonlinear update equations, and the particularities of each case tend to require different arguments and assumptions.

This chapter and the next present a unified energy-based approach to the mean-square analysis of adaptive filters. The energy approach makes it possible not only to treat algorithms uniformly, but also to arrive at new performance results. This approach is based on

[1] A major part of this chapter is reproduced, with permission, from T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Transactions on Signal Processing,* vol. 51, No. 3, pp. 639-652, Mar. 2003.

studying the energy flow through each iteration of an adaptive filter, and it relies on an exact energy conservation relation that holds for a large class of adaptive filters. This relation has been originally developed in [84, 77, 85, 78] in the context of robustness analysis of adaptive filters within a deterministic framework. It has since then been used in [61, 105, 106, 103] as a convenient tool for studying the steady-state performance of adaptive filters within a stochastic framework as well. In this Chapter, we show how to extend the energy-based approach to the *transient* analysis (as opposed to the *steady-state* analysis) of adaptive filters that employ data-nonlinearities in their update equation.

2.1.1 Organization of the Chapter

This chapter is organized as follows. In the next section we introduce weighted estimation errors as well as weighted energy norms and relate these quantities through a fundamental energy relation. In Sections 2.3 and 2.4, we illustrate the mechanism of our approach for transient analysis by applying it to the LMS algorithm and its normalized version for Gaussian regressors. In Section 2.5, we study the general case of adaptive algorithms with data nonlinearities without imposing restrictions on the color of the regression data (i.e., without requiring the regression data to be Gaussian or white). The analysis leads to stability results and closed-form expressions for the mean-square error (MSE) and mean-square deviation (MSD). We further generalize our study in Section 2.6 to adaptive filters that employ matrix data nonlinearities. We again derive stability results and closed-form expressions for the MSE and MSD.

In the next chapter, we extend the energy-conservation approach to study the transient behavior of adaptive filters with error nonlinearities.

2.1.2 Notation

We focus on real-valued data, although the extension to complex-valued data is immediate. Small boldface letters are used to denote vectors, e.g., $\boldsymbol{w}$. Also, the symbol T denotes transposition. The notation $\|\boldsymbol{w}\|^2$ denotes the squared Euclidean norm of a vector, $\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T\boldsymbol{w}$, while $\|\boldsymbol{w}\|^2_{\boldsymbol{\Sigma}}$ denotes the weighted squared Euclidean norm, $\|\boldsymbol{w}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}$. All vectors are column vectors except for a single vector, namely the input data vector denoted by $\mathbf{u}_i$, which is taken to be a row vector. The time instant is placed as a subscript for vectors and between parentheses for scalars, e.g., $\boldsymbol{w}_i$ and $e(i)$.

2.1.3 Adaptive Filters with Data Nonlinearities

Consider noisy measurements $\{d(i)\}$ that arise from the model

$$d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$$

for some $M \times 1$ unknown vector $\boldsymbol{w}^o$ that we wish to estimate, and where $v(i)$ accounts for measurement noise and modelling errors, and $\mathbf{u}_i$ denotes a *row* regression vector. Both $\mathbf{u}_i$ and $v(i)$ are stochastic in nature. Many adaptive schemes have been developed in the literature for the estimation of $\boldsymbol{w}^o$ in different contexts. Most of these algorithms fit into the general description:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu f[e(i), \mathbf{u}_i]\mathbf{u}_i^T, \qquad i \geq 0 \tag{2.1}$$

where $\boldsymbol{w}_i$ is an estimate for $\boldsymbol{w}^o$ at iteration i, μ is the step-size,

$$e(i) = d(i) - \mathbf{u}_i \boldsymbol{w}_i \;=\; \mathbf{u}_i \boldsymbol{w}^o - \mathbf{u}_i \boldsymbol{w}_i + v(i) \tag{2.2}$$

is the estimation error, and $f[e(i), \mathbf{u}_i]$ denotes a generic function of $e(i)$ and the regression vector $\mathbf{u}_i$.

In terms of the weight-error vector $\tilde{\boldsymbol{w}}_i = \boldsymbol{w}^o - \boldsymbol{w}_i$, the adaptive filter equations (2.1) and (2.2) can be equivalently rewritten as

$$\boxed{\tilde{\boldsymbol{w}}_{i+1} = \tilde{\boldsymbol{w}}_i - \mu f[e(i), \mathbf{u}_i]\mathbf{u}_i^T} \tag{2.3}$$

and

$$\boxed{e(i) = \mathbf{u}_i \tilde{\boldsymbol{w}}_i + v(i)} \tag{2.4}$$

We restrict our attention in this chapter to nonlinearities $f[\cdot]$ that can be expressed in the *separable* form

$$\boxed{f[e(i), \mathbf{u}_i] = \tfrac{e(i)}{g[\mathbf{u}_i]}} \tag{2.5}$$

for some positive scalar-valued function $g[\mathbf{u}_i]$. In the latter part of this chapter (see Section 2.6), matrix nonlinearities $\mathbf{H}[\mathbf{u}_i]$ will also be considered, i.e., functions $f[\cdot]$ of the form

$$f[e(i), \mathbf{u}_i] = \mathbf{H}[\mathbf{u}_i]e(i)$$

Table 1 lists some examples of data nonlinearities $\{g[\cdot], \mathbf{H}[\cdot]\}$ that appear in the literature. In the table, the notation $\{u_{i_1}, u_{i_2}, \ldots, u_{i_M}\}$ refers to the entries of the regressor vector $\mathbf{u}_i$.

Table 2.1: *Examples of data nonlinearities* $g[\cdot]$ *or* $\mathbf{H}[\cdot]$

ALGORITHM	$g[\cdot]$ OR $\mathbf{H}[\cdot]$
LMS	1
NLMS	$\|\mathbf{u}_i\|^2$
ϵ-NLMS	$\epsilon + \|\mathbf{u}_i\|^2$
NLMS family	$\frac{1}{\|\mathbf{u}_i\|_q^q}\text{diag}\left(\|u_{i_1}\|^{q-1}\text{sgn}(u_{i_1}), \|u_{i_2}\|^{q-1}\text{sgn}(u_{i_2}), \ldots, \|u_{i_M}\|^{q-1}\text{sgn}(u_{i_M})\right)$
Power normalized LMS	$\text{diag}\left(p_1(i), p_2(i), \ldots, p_M(i)\right)$ $p_k(i+1) = \beta p_k(i) + (1-\beta)\|u_{i_k}\|^2, \quad 0 \ll \beta < 1$
Sign regressor	$\text{diag}\left(\frac{\text{sgn}(u_{i_1})}{u_{i_1}}, \frac{\text{sgn}(u_{i_2})}{u_{i_2}}, \ldots, \frac{\text{sgn}(u_{i_M})}{u_{i_M}}\right)$
Multiple step-sizes	$\text{diag}(\mu_1, \mu_2, \ldots, \mu_M)$

2.2 A Weighted Energy Relation

The adaptive filter analysis in future sections is based on an energy-conservation relation that relates the energies of several error quantities. To derive this relation, we first define some useful weighted errors. Thus, let $\mathbf{\Sigma}$ denote any symmetric $M \times M$ weighting matrix and define the weighted a-priori and a-posteriori error signals:

$$e_a^{\Sigma}(i) \stackrel{\Delta}{=} \mathbf{u}_i \mathbf{\Sigma} \tilde{\boldsymbol{w}}_i, \qquad e_p^{\Sigma}(i) \stackrel{\Delta}{=} \mathbf{u}_i \mathbf{\Sigma} \tilde{\boldsymbol{w}}_{i+1} \tag{2.6}$$

For $\mathbf{\Sigma} = \boldsymbol{I}$, we use the more standard notation

$$e_a(i) \stackrel{\Delta}{=} e_a^I(i) = \mathbf{u}_i \tilde{\boldsymbol{w}}_i, \quad e_p(i) \stackrel{\Delta}{=} e_p^I(i) = \mathbf{u}_i \tilde{\boldsymbol{w}}_{i+1}$$

The freedom in selecting $\mathbf{\Sigma}$ will enable us to perform different kinds of analyses. For now, $\mathbf{\Sigma}$ will simply denote an arbitrary weighting matrix.

2.2.1 Energy-Conservation Relation

The energy relation that we seek is one that relates the energies of the following error quantities:

$$\{\ \tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{w}}_{i+1}, e_a^{\Sigma}(i), e_p^{\Sigma}(i)\ \} \tag{2.7}$$

To arrive at the desired relation, we pre-multiply both sides of the adaptation equation (2.3) by $\mathbf{u}_i\boldsymbol{\Sigma}$ and incorporate the definitions (2.6). This results in an equality that relates the estimation errors $e_a^{\Sigma}(i)$, $e_p^{\Sigma}(i)$, and $e(i)$, namely,

$$e_p^{\Sigma}(i) = e_a^{\Sigma}(i) - \frac{\mu}{\overline{\mu}_{\Sigma}(i)} f[e(i), \mathbf{u}_i] \tag{2.8}$$

where we introduced, for compactness of notation, the scalar quantity

$$\overline{\mu}_{\Sigma}(i) \triangleq \begin{cases} 1/\mathbf{u}_i\boldsymbol{\Sigma}\mathbf{u}_i^T & \text{if } \mathbf{u}_i\boldsymbol{\Sigma}\mathbf{u}_i^T \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

Using (2.8), the nonlinearity $f[e(i), \mathbf{u}_i]$ can be eliminated from (2.3), yielding the following relation between the errors in (2.7):

$$\boxed{\tilde{\boldsymbol{w}}_{i+1} = \tilde{\boldsymbol{w}}_i - \overline{\mu}_{\Sigma}(i)\mathbf{u}_i^T[e_a^{\Sigma}(i) - e_p^{\Sigma}(i)]}$$

From this equation, it follows that the weighted energies of these errors are related by

$$\tilde{\boldsymbol{w}}_{i+1}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}_{i+1} = \left(\tilde{\boldsymbol{w}}_i - \overline{\mu}_{\Sigma}(i)\mathbf{u}_i^T[e_a^{\Sigma}(i) - e_p^{\Sigma}(i)]\right)^T \boldsymbol{\Sigma} \left(\tilde{\boldsymbol{w}}_i - \overline{\mu}_{\Sigma}(i)\mathbf{u}_i^T[e_a^{\Sigma}(i) - e_p^{\Sigma}(i)]\right)$$

or, more compactly, after expanding and grouping terms, by the following *energy-conservation* identity:

$$\boxed{\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2 + \overline{\mu}_{\Sigma}(i)\left|e_a^{\Sigma}(i)\right|^2 = \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 + \overline{\mu}_{\Sigma}(i)\left|e_p^{\Sigma}(i)\right|^2} \tag{2.10}$$

This result is *exact* for any adaptive algorithm described by (2.3), i.e., for any nonlinearity $f[\cdot,\cdot]$, and it has been derived without any approximations. Also, no restrictions have been imposed on the symmetric weighting matrix $\boldsymbol{\Sigma}$.

The result (2.10) with $\boldsymbol{\Sigma} = \boldsymbol{I}$ was developed in [84, 77, 85] in the context of robustness analysis of adaptive filters, and it was later used in [61, 105, 106, 103] in the context of steady-state and tracking analysis. The incorporation of a weighting matrix $\boldsymbol{\Sigma}$ allows us to perform transient analyses as well, as we shall discuss in future sections.

2.2.2 The Algebra of Weighted Norms

Before proceeding, it is convenient for the subsequent discussion to list some algebraic properties of weighted norms. So let a_1 and a_2 be scalars and let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be symmetric matrices of size M. Then the following properties hold:

1) Superposition.

$$a_1\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1} + a_2\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_2} = \|\tilde{\boldsymbol{w}}_i\|^2_{a_1\Sigma_1+a_2\Sigma_2} \tag{2.11}$$

2) Polarization.

$$\begin{aligned}(\mathbf{u}_i\mathbf{\Sigma}_1\tilde{\boldsymbol{w}}_i)\,(\mathbf{u}_i\mathbf{\Sigma}_2\tilde{\boldsymbol{w}}_i) &= \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1 u_i^T u_i \Sigma_2} \\ &= \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_2 u_i^T u_i \Sigma_1}\end{aligned} \tag{2.12}$$

3) Independence. If $\mathbf{u}_i$ and $\tilde{\boldsymbol{w}}_i$ are independent random vectors, then the polarization property gives

$$E\left[(\mathbf{u}_i\mathbf{\Sigma}_1\tilde{\boldsymbol{w}}_i)\,(\mathbf{u}_i\mathbf{\Sigma}_2\tilde{\boldsymbol{w}}_i)\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1 u_i^T u_i \Sigma_2}\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1 E[u_i^T u_i]\Sigma_2}\right]$$

where the last equality is true when $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are constant matrices.

4) Linear transformation. For any $N \times M$ matrix $\mathbf{A}$,

$$\|\mathbf{A}\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \|\tilde{\boldsymbol{w}}_i\|^2_{A^T\Sigma A}$$

5) Orthogonal transformation. If $\boldsymbol{Q}$ is orthogonal, it is easy to see that

$$\|\boldsymbol{Q}^T\tilde{\boldsymbol{w}}_i\|^2 = \|\tilde{\boldsymbol{w}}_i\|^2 \tag{2.13}$$

6) Blindness to asymmetry. The weighted sum of squares is blind to any asymmetry in the weight $\mathbf{A}$, i.e.,

$$\|\tilde{\boldsymbol{w}}_i\|^2_A = \|\tilde{\boldsymbol{w}}_i\|^2_{A^T} = \|\tilde{\boldsymbol{w}}_i\|^2_{\frac{1}{2}A+\frac{1}{2}A^T} \tag{2.14}$$

7) Notational convention. We shall often write

$$\|\tilde{\boldsymbol{w}}_i\|^2_{\mathrm{vec}(\Sigma_1)} \stackrel{\Delta}{=} \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1}$$

where vec (Σ_1) is obtained by stacking all the columns of $\mathbf{\Sigma}_1$ into a vector. For the special case when $\mathbf{\Sigma}_1$ is diagonal, it suffices to collect the diagonal entries of $\mathbf{\Sigma}_1$ into a vector and we thus write

$$\|\tilde{\boldsymbol{w}}_i\|^2_{\text{diag}(\Sigma_1)} \stackrel{\Delta}{=} \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_1}$$

2.2.3 Data-Normalized Filters

We now examine the simplifications that occur when $f[\cdot,\cdot]$ is restricted to the form (2.5). Upon replacing $e_p^{\Sigma}(i)$ in (2.10) by its equivalent expression (2.8) and expanding we get

$$\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma} = \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} - 2\mu\frac{e(i)e_a^{\Sigma}(i)}{g[\mathbf{u}_i]} + \frac{\mu^2}{\overline{\mu}_{\Sigma}(i)}\frac{e^2(i)}{g^2[\mathbf{u}_i]} \tag{2.15}$$

To proceed, we replace $e(i)$, as defined in (2.4), by

$$e(i) = e_a(i) + v(i)$$

Then (2.15) becomes

$$\begin{aligned}\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma} &= \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} - 2\mu\frac{e_a(i)e_a^{\Sigma}(i)}{g[\mathbf{u}_i]} + \frac{\mu^2}{\overline{\mu}_{\Sigma}(i)}\frac{e_a^2(i)}{g^2[\mathbf{u}_i]} \\ &\quad -2\mu\left(\frac{e_a^{\Sigma}(i)}{g[\mathbf{u}_i]} - \frac{\mu}{\overline{\mu}_{\Sigma}(i)}\frac{e_a(i)}{g^2[\mathbf{u}_i]}\right)v(i) + \frac{\mu^2}{\overline{\mu}_{\Sigma}(i)}\frac{v^2(i)}{g^2[\mathbf{u}_i]}\end{aligned} \tag{2.16}$$

Now note that $e_a^{\Sigma}(i)e_a(i)$ and $e_a^2(i)$ can be expressed as some weighted norms of $\tilde{\boldsymbol{w}}_i$. Indeed, from (2.12), we have

$$e_a(i)e_a^{\Sigma}(i) = (\mathbf{u}_i\tilde{\boldsymbol{w}}_i)\,(\mathbf{u}_i\mathbf{\Sigma}\tilde{\boldsymbol{w}}_i) = \|\tilde{\boldsymbol{w}}_i\|^2_{u_i^T u_i \Sigma} \tag{2.17}$$

and, subsequently,

$$e_a^2(i) = e_a(i)e_a^I(i) = \|\tilde{\boldsymbol{w}}_i\|^2_{u_i^T u_i} \tag{2.18}$$

Upon substituting (2.17) and (2.18) into (2.16), we get

$$\begin{aligned}\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma} &= \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} - 2\mu\frac{1}{g[\mathbf{u}_i]}\|\tilde{\boldsymbol{w}}_i\|^2_{u_i^T u_i \Sigma} + \mu^2\frac{\|\mathbf{u}_i\|^4_{\Sigma}}{g^2[\mathbf{u}_i]}\|\tilde{\boldsymbol{w}}_i\|^2_{u_i^T u_i} \\ &\quad -2\mu\left(\frac{e_a^{\Sigma}(i)}{g[\mathbf{u}_i]} - \frac{\mu}{\overline{\mu}_{\Sigma}(i)}\frac{e_a(i)}{g^2[\mathbf{u}_i]}\right)v(i) + \mu^2 v^2(i)\frac{\|\mathbf{u}_i\|^2_{\Sigma}}{g^2[\mathbf{u}_i]}\end{aligned}$$

This relation can be written more compactly by using the superposition property (2.11) to group the various weighted norms of $\tilde{\boldsymbol{w}}_i$ into one term, namely,

$$\boxed{\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma} = \|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma'} - 2\mu\left(\frac{e_a^{\Sigma}(i)}{g[\mathbf{u}_i]} - \frac{\mu}{\overline{\mu}_{\Sigma}(i)}\frac{e_a(i)}{g^2[\mathbf{u}_i]}\right)v(i) + \mu^2 v^2(i)\frac{\|\mathbf{u}_i\|^2_{\Sigma}}{g^2[\mathbf{u}_i]}} \tag{2.19}$$

where

$$\boldsymbol{\Sigma}' \triangleq \boldsymbol{\Sigma} - 2\mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\boldsymbol{\Sigma} + \mu^2\frac{\|\mathbf{u}_i\|^2_{\Sigma}}{g^2[\mathbf{u}_i]}\mathbf{u}_i^T\mathbf{u}_i \tag{2.20}$$

The only role that $\boldsymbol{\Sigma}'$ plays is a weight in the quadratic form $\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma'}$. Hence, and in view of (2.14), we can replace the defining expression (2.20) for $\boldsymbol{\Sigma}'$ by its symmetric part

$$\boxed{\boldsymbol{\Sigma}' \triangleq \boldsymbol{\Sigma} - \mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\boldsymbol{\Sigma} - \mu\boldsymbol{\Sigma}\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]} + \mu^2\frac{\|\mathbf{u}_i\|^2_{\Sigma}}{g^2[\mathbf{u}_i]}\mathbf{u}_i^T\mathbf{u}_i} \tag{2.21}$$

Finally, it is straightforward to conclude from the weight-error recursion

$$\tilde{\boldsymbol{w}}_{i+1} = \tilde{\boldsymbol{w}}_i - \mu\frac{\mathbf{u}_i^T}{g[\mathbf{u}_i]}[e_a(i) + v(i)]$$

and from $e_a(i) = \mathbf{u}_i\tilde{\boldsymbol{w}}_i$ that

$$\boxed{\tilde{\boldsymbol{w}}_{i+1} = \left(\boldsymbol{I} - \mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right)\tilde{\boldsymbol{w}}_i \;-\; \mu\frac{\mathbf{u}_i^T}{g[\mathbf{u}_i]}v(i)} \tag{2.22}$$

2.2.4 Weighted Variance Relation

A few comments are in place:

1. First, the pair (2.19) and (2.21) is equivalent to the energy relation (2.10) and hence is exact.

2. This pair represents the starting point for various types of analyses of adaptive filters

with data normalization.

3. As it stands, the energy relation (2.19)–(2.21) cannot be propagated in time since it requires a recursion describing the evolution of $e_a(i)$. However, this complication can be removed by introducing the following reasonable assumption on the noise sequence:

   AN. The noise sequence $v(i)$ is zero-mean, iid, and is independent of $\mathbf{u}_i$.

   This assumption renders the third term of (2.19) zero-mean and relation (2.19) simplifies under expectation to

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma'}^2\right] + \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|_{\Sigma}^2}{g^2(\mathbf{u}_i)}\right] \tag{2.23}$$

   Likewise, recursion (2.22) simplifies to

$$E\tilde{\boldsymbol{w}}_{i+1} = E\left[\left(\boldsymbol{I} - \mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right)\tilde{\boldsymbol{w}}_i\right] \tag{2.24}$$

   While the iterated relation (2.23) is compact, it is still hard to propagate since $\boldsymbol{\Sigma}'$ is dependent on the data $\mathbf{u}_i$, so that the evaluation of the expectation $E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma'}^2\right]$ is not trivial in general.

d) For this reason, we shall contend ourselves with the independence assumption:

   AI. The sequence of vectors $\mathbf{u}_i$ is independent and identically distributed.

   This condition enables us to the split the expectation in (2.23) as

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{E[\Sigma']}^2\right] + \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|_{\Sigma}^2}{g^2(\mathbf{u}_i)}\right] \tag{2.25}$$

   Observe that the weighting matrix for $\tilde{\boldsymbol{w}}_i$ is now given by the expectation $E[\boldsymbol{\Sigma}']$. As we shall see soon, the above equality renders the issue of transient and stability analyses of an adaptive filter equivalent to a multivariate computation of certain moments.

   In order to emphasize the fact that the weighting matrix changes from $\boldsymbol{\Sigma}$ to $E[\boldsymbol{\Sigma}']$ according to (2.21), we shall attach a time index to the weighting matrices and use

(2.21) and (2.25) to write more explicitly:

$$\boxed{E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma_{i+1}}\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_i}\right] + \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|^2_{\Sigma_{i+1}}}{g^2[\mathbf{u}_i]}\right]}$$

where we replaced $\boldsymbol{\Sigma}$ by $\boldsymbol{\Sigma}_{i+1}$ and $E[\boldsymbol{\Sigma}']$ by $\boldsymbol{\Sigma}_i$, which is now defined by

$$\boxed{\boldsymbol{\Sigma}_i \stackrel{\Delta}{=} \boldsymbol{\Sigma}_{i+1} - \mu E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right]\boldsymbol{\Sigma}_{i+1} - \mu\boldsymbol{\Sigma}_{i+1}E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right] \;+\; \mu^2 E\left[\frac{\|\mathbf{u}_i\|_{\Sigma_{i+1}}}{g^2[\mathbf{u}_i]}\mathbf{u}_i^T\mathbf{u}_i\right]}$$

Note that this recursion runs backwards in time, and its boundary condition will therefore be specified at ∞.

Likewise, applying the independence assumption A1 to the right-hand side of (2.24), we find that

$$\boxed{E\tilde{\boldsymbol{w}}_{i+1} = E\left(\boldsymbol{I} - \mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right)\cdot E\tilde{\boldsymbol{w}}_i}$$

with the expectation on the right-hand side of (2.24) split into the product of two expectations.

e) Inspection of recursions (2.19) and (2.23) reveals that the iid assumption (A5) on the noise sequence is critical. Indeed, while (2.23) can be propagated in time without the independence assumption A1, it is not possible to do the same for (2.19). Fortunately, assumption A5 is in general reasonable.

We summarize in the following statement the variance and mean recursions that will form the basis of our transient analysis.

Theorem 1 (Weighted-variance relation) *Consider an adaptive filter of the form*

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu\frac{\mathbf{u}_i^T}{g[\mathbf{u}_i]}e(i), \quad i \geq 0 \tag{2.26}$$

where $e(i) = d(i) - \mathbf{u}_i\boldsymbol{w}_i$ *and* $d(i) = \mathbf{u}_i\boldsymbol{w}^o + v(i)$. *Assume that the sequences* $\{v(i), \mathbf{u}_i\}$ *are* iid *and mutually independent. For any given* $\boldsymbol{\Sigma}_{i+1}$, *it holds that*

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma_{i+1}}\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_i}\right] + \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|^2_{\Sigma_{i+1}}}{g^2[\mathbf{u}_i]}\right] \tag{2.27}$$

where $\boldsymbol{\Sigma}_i$ *is constructed from* $\boldsymbol{\Sigma}_{i+1}$ *via*

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i+1} - \mu E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right]\boldsymbol{\Sigma}_{i+1} - \mu\boldsymbol{\Sigma}_{i+1}E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right] \;+\; \mu^2 E\left[\frac{\|\mathbf{u}_i\|_{\Sigma_{i+1}}}{g^2[\mathbf{u}_i]}\mathbf{u}_i^T\mathbf{u}_i\right] \tag{2.28}$$

It also holds that the mean weight-error vector satisfies

$$E\tilde{\boldsymbol{w}}_{i+1} = E\left(\boldsymbol{I} - \mu\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right)\cdot E\tilde{\boldsymbol{w}}_i \tag{2.29}$$

♢

The purpose of the sections that follow is to show how the above variance and mean recursions can be used to study the transient performance of adaptive schemes with data nonlinearities. In particular, we shall show how the freedom in selecting the weighting matrix $\boldsymbol{\Sigma}_{i+1}$ can be used advantageously to derive several performance measures.

First, however, we shall illustrate the mechanism of our analysis by considering two special cases for which results are already available in the literature. More specifically, we shall start with the transient analysis of LMS and normalized LMS algorithms for Gaussian regression data in Sections 2.3 and 2.4. Once the main ideas have been illustrated in this manner, we shall then describe our general procedure in Section 2.5, which applies to adaptive filters with more general data normalization and also to regression data that are not restricted to being Gaussian or white.

2.2.5 A Change of Variables

In the meantime, we remark that sometimes it is useful to employ a convenient change of coordinates, especially when dealing with Gaussian regressors. Thus let $\boldsymbol{R} = E\mathbf{u}_i^T\mathbf{u}_i$ denote the covariance matrix of $\mathbf{u}_i$ and introduce its eigen-decomposition,

$$\boldsymbol{R} = \boldsymbol{Q}^T\boldsymbol{\Lambda}\boldsymbol{Q}$$

where $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{\Lambda}$ is a positive diagonal matrix with entries $\{\lambda_k\}$. Define further

$$\overline{\boldsymbol{w}}_i \triangleq \boldsymbol{Q}\tilde{\boldsymbol{w}}_i, \qquad \overline{\mathbf{u}}_i \triangleq \mathbf{u}_i\boldsymbol{Q}^T, \qquad \overline{\boldsymbol{\Sigma}}_i \triangleq \boldsymbol{Q}\boldsymbol{\Sigma}_i\boldsymbol{Q}^T \tag{2.30}$$

In view of the orthogonal transformation property (2.13), we have

$$\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\Sigma}_i} = \|\overline{\boldsymbol{w}}_i\|^2_{\overline{\boldsymbol{\Sigma}}_i} \quad \text{and} \quad \|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}_i} = \|\overline{\mathbf{u}}_i\|^2_{\overline{\boldsymbol{\Sigma}}_i}$$

Moreover, assuming that the nonlinearity $g[\cdot]$ is invariant under orthogonal transformations, i.e., $g[\mathbf{u}_i] = g[\overline{\mathbf{u}}_i]$ (e.g., $g[\mathbf{u}_i] = 1$ or $g[\mathbf{u}_i] = \|\mathbf{u}_i\|^2$), we find that the variance relation (2.27) retains the same form, namely

$$\boxed{E\left[\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{\Sigma}_{i+1}}\right] = E\left[\|\overline{\boldsymbol{w}}_i\|^2_{\overline{\Sigma}_i}\right] + \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2_{\overline{\Sigma}_{i+1}}}{g^2[\overline{\mathbf{u}}_i]}\right]} \tag{2.31}$$

By pre-multiplying both sides of (2.28) by $\boldsymbol{Q}$ and post-multiplying by $\boldsymbol{Q}^T$, we similarly see that (2.28) also retains the same form:

$$\boxed{\overline{\boldsymbol{\Sigma}}_i = \overline{\boldsymbol{\Sigma}}_{i+1} - \mu E\left[\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{g[\overline{\mathbf{u}}_i]}\right]\overline{\boldsymbol{\Sigma}}_{i+1} - \mu\overline{\boldsymbol{\Sigma}}_{i+1}E\left[\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{g[\overline{\mathbf{u}}_i]}\right] + \mu^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|_{\overline{\Sigma}_{i+1}}}{g^2[\overline{\mathbf{u}}_i]}\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i\right]} \tag{2.32}$$

Likewise, (2.29) becomes

$$\boxed{E\overline{\boldsymbol{w}}_{i+1} = E\left(\boldsymbol{I} - \mu\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{g[\overline{\mathbf{u}}_i]}\right)\cdot E\overline{\boldsymbol{w}}_i} \tag{2.33}$$

2.3 LMS with Gaussian Regressors

Consider the LMS algorithm for which $g[\mathbf{u}_i] = 1$ and assume that:

AG. The regressors $\{\mathbf{u}_i\}$ arise from a Gaussian distribution with covariance matrix $\boldsymbol{R}$.

In this case, the data dependent moments that appear in (2.31)–(2.33) are given by

$$E\left[\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i\right] = \boldsymbol{\Lambda}, \quad E\left[\|\overline{\mathbf{u}}_i\|^2_{\overline{\Sigma}_i}\,\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i\right] = 2\overline{\boldsymbol{\Sigma}}_i\boldsymbol{\Lambda}^2 + \mathrm{Tr}\left(\overline{\boldsymbol{\Sigma}}_i\boldsymbol{\Lambda}\right)\boldsymbol{\Lambda}$$

Therefore, for LMS, recursions (2.31) and (2.32) simplify to

$$\boxed{E\left[\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{\Sigma}_{i+1}}\right] = E\left[\|\overline{\boldsymbol{w}}_i\|^2_{\overline{\Sigma}_i}\right] + \mu^2\sigma_v^2 E\left[\|\overline{\mathbf{u}}_i\|^2_{\overline{\Sigma}_{i+1}}\right]} \tag{2.34}$$

and

$$\overline{\boldsymbol{\Sigma}}_i = \overline{\boldsymbol{\Sigma}}_{i+1} - \mu\boldsymbol{\Lambda}\overline{\boldsymbol{\Sigma}}_{i+1} - \mu\overline{\boldsymbol{\Sigma}}_{i+1}\boldsymbol{\Lambda} + 2\mu^2\overline{\boldsymbol{\Sigma}}_{i+1}\boldsymbol{\Lambda}^2 + \mu^2\mathrm{Tr}\left(\overline{\boldsymbol{\Sigma}}_{i+1}\boldsymbol{\Lambda}\right)\boldsymbol{\Lambda} \tag{2.35}$$

while (2.33) becomes

$$\boxed{E\overline{\boldsymbol{w}}_{i+1} = E\left(\boldsymbol{I} - \mu\boldsymbol{\Lambda}\right)\cdot E\overline{\boldsymbol{w}}_i} \tag{2.36}$$

Now observe that in recursion (2.35), $\overline{\boldsymbol{\Sigma}}_i$ will be diagonal if $\overline{\boldsymbol{\Sigma}}_{i+1}$ is. Therefore, in order for all successive $\overline{\boldsymbol{\Sigma}}_i$'s to be diagonal it is sufficient to assume that the boundary condition for the recursion for $\boldsymbol{\Sigma}_i$ is taken as diagonal. In this way, the $\overline{\boldsymbol{\Sigma}}_i$'s will be completely characterized by their diagonal entries. This prompts us to define the column vectors

$$\overline{\boldsymbol{\sigma}}_i \triangleq \mathrm{diag}(\overline{\boldsymbol{\Sigma}}_i) \quad \text{and} \quad \boldsymbol{\lambda} \triangleq \mathrm{diag}(\boldsymbol{\Lambda})$$

In terms of these vectors, the matrix recursion (2.35) can be replaced by the more compact vector recursion

$$\overline{\boldsymbol{\sigma}}_i = \left(\boldsymbol{I} - 2\mu\boldsymbol{\Lambda} + 2\mu^2\boldsymbol{\Lambda}^2\right)\overline{\boldsymbol{\sigma}}_{i+1} + \mu^2\left(\boldsymbol{\lambda}^T\overline{\boldsymbol{\sigma}}_{i+1}\right)\boldsymbol{\lambda}$$

or

$$\boxed{\overline{\boldsymbol{\sigma}}_i = \overline{\boldsymbol{F}}\overline{\boldsymbol{\sigma}}_{i+1}} \tag{2.37}$$

where

$$\boxed{\overline{\boldsymbol{F}} \triangleq \left(\boldsymbol{I} - 2\mu\boldsymbol{\Lambda} + 2\mu^2\boldsymbol{\Lambda}^2\right) + \mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T}$$

The matrix $\overline{\boldsymbol{F}}$ describes the dynamics by which the weighting matrices $\overline{\boldsymbol{\Sigma}}_i$ evolve in time, and its eigen-structure turns out to be essential for filter stability. Using the fact that $\overline{\boldsymbol{\sigma}}_i = \overline{\boldsymbol{F}}\overline{\boldsymbol{\sigma}}_{i+1}$, we can rewrite (2.34) using a compact vector weighting notation:

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{\sigma}_{i+1}} = E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}\overline{\sigma}_{i+1}} + \mu^2\sigma_v^2 E\|\overline{\mathbf{u}}_i\|^2_{\overline{\sigma}_{i+1}} \tag{2.38}$$

Recursions (2.36), (2.37), and (2.38) describe the transient behavior of LMS, and conclusions about mean-square stability and mean-square performance are now possible.

In transient analysis we are interested in the time evolution of the expectations $\{E\tilde{\boldsymbol{w}}_i, E\|\tilde{\boldsymbol{w}}_i\|^2\}$ or, equivalently, $\{E\overline{\boldsymbol{w}}_i, E\|\overline{\boldsymbol{w}}_i\|^2\}$ since $\overline{\boldsymbol{w}}_i$ and $\tilde{\boldsymbol{w}}_i$ are related via the orthogonal matrix $\boldsymbol{Q}$. We start with the mean behavior.

2.3.1 Mean Behavior and Mean Stability

From (2.36) we find that the filter is convergent in the mean if, and only if, the step-size μ satisfies

$$\boxed{\mu < \frac{2}{\lambda_{\max}}} \tag{2.39}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{R}$.

2.3.2 Mean-Square Behavior

The evolution of $E\|\tilde{\boldsymbol{w}}_i\|^2 = E\|\overline{\boldsymbol{w}}_i\|^2$ can be deduced from the variance recursion (2.34) if $\overline{\boldsymbol{\Sigma}}_{i+1}$ is chosen as $\overline{\boldsymbol{\Sigma}}_{i+1} = \boldsymbol{I}$ (or, equivalently, $\boldsymbol{\Sigma}_{i+1} = \boldsymbol{I}$). This corresponds to choosing $\overline{\boldsymbol{\sigma}}_{i+1}$ in (2.38) as a column vector with unit entries, denoted by

$$\overline{\boldsymbol{\sigma}}_{i+1} \;=\; \mathbf{1} \;\stackrel{\Delta}{=}\; \mathrm{col}\{1, 1, \ldots, 1\}$$

Now we can see from (2.38) that

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2 = E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}1} \;+\; \mu^2\sigma_v^2\left(\sum_{k=1}^{M}\lambda_k\right) \tag{2.40}$$

which shows that in order to evaluate $E\|\overline{\boldsymbol{w}}_{i+1}\|^2$ we need $E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}1}$ with a weighting matrix equal to $\overline{\boldsymbol{F}}\mathbf{1}$. Now $E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}1}$ can be deduced from (2.38) by setting $\overline{\boldsymbol{\sigma}}_{i+1} = \overline{\boldsymbol{F}}\mathbf{1}$, i.e.,

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}1} = E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^2 1} \;+\; \mu^2\sigma_v^2\left(\boldsymbol{\lambda}^T\overline{F}1\right) \tag{2.41}$$

Again, in order to evaluate $E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}1}$ we need $E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^2 1}$, with weighting $\overline{\boldsymbol{F}}^2 1$. This term can be deduced from (2.38) by choosing $\overline{\boldsymbol{\sigma}}_{i+1} = \overline{\boldsymbol{F}}^2\mathbf{1}$:

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^2 1} = E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^3 1} \;+\; \mu^2\sigma_v^2\left(\boldsymbol{\lambda}^T\overline{F}^2\mathbf{1}\right) \tag{2.42}$$

and a new term with weighting matrix $\overline{\boldsymbol{F}}^3\mathbf{1}$ appears. Fortunately, this procedure terminates in view of the Cayley-Hamilton theorem. Thus let $p(x) = \det(x\boldsymbol{I} - \overline{\boldsymbol{F}})$ denote the characteristic polynomial of $\overline{\boldsymbol{F}}$; it is an $M-$th order polynomial in x,

$$p(x) = x^M + p_{M-1}x^{M-1} + p_{M-2}x^{M-2} + \ldots + p_1 x + p_0$$

with coefficients $\{p_k\}$. The Cayley-Hamilton theorem states that every matrix satisfies its characteristic equation, i.e., $p(\overline{\boldsymbol{F}}) = 0$, which allows us to conclude that

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^M 1} \;=\; \sum_{k=0}^{M-1} -p_k E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^k 1} \tag{2.43}$$

We can now collect the above results into a single recursion by writing (2.40)–(2.43) as:

$$\underbrace{\begin{bmatrix} E\|\overline{\boldsymbol{w}}_{i+1}\|^2_1 \\ E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}1} \\ E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^2 1} \\ \vdots \\ E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^{(M-2)} 1} \\ E\|\overline{\boldsymbol{w}}_{i+1}\|^2_{\overline{F}^{(M-1)} 1} \end{bmatrix}}_{=\mathcal{W}_{i+1}} = \underbrace{\begin{bmatrix} 0 & 1 & & & \\ 0 & 0 & 1 & & \\ 0 & 0 & 0 & 1 & \\ \vdots & & & & \\ 0 & 0 & 0 & & 1 \\ -p_0 & -p_1 & -p_2 & \cdots & -p_{M-1} \end{bmatrix}}_{=\mathcal{F}} \begin{bmatrix} E\|\overline{\boldsymbol{w}}_i\|^2_1 \\ E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}1} \\ E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^2 1} \\ \vdots \\ E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^{(M-2)} 1} \\ E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{F}^{(M-1)} 1} \end{bmatrix}$$

$$+ \mu^2\sigma_v^2 \underbrace{\begin{bmatrix} \boldsymbol{\lambda}^T\mathbf{1} \\ \boldsymbol{\lambda}^T\overline{\boldsymbol{F}}\mathbf{1} \\ \boldsymbol{\lambda}^T\overline{\boldsymbol{F}}^2\mathbf{1} \\ \vdots \\ \boldsymbol{\lambda}^T\overline{\boldsymbol{F}}^{M-1}\mathbf{1} \end{bmatrix}}_{\mathcal{Y}}$$

If we define the vector and matrix quantities $\{\mathcal{W}_i, \mathcal{F}, \mathcal{Y}\}$ as indicated above, then the recursion can be rewritten more compactly as

$$\boxed{\mathcal{W}_{i+1} = \mathcal{F}\mathcal{W}_i + \mu^2\sigma_v^2\mathcal{Y}} \tag{2.44}$$

We therefore find that the transient behavior of LMS is described by the $M-$dimensional state-space recursion (2.44) with coefficient matrix $\mathcal{F}$.[2] The evolution of the top entry of $\mathcal{W}_i$ corresponds to the mean-square deviation of the filter. Observe further that the eigenvalues of $\mathcal{F}$ coincide with those of $\overline{\boldsymbol{F}}$.

It is worth remarking that the same derivation that led to (2.44) with $\mathcal{W}_i$ defined in

[2]To be more precise, the transient behavior of LMS is described by the combination of both (2.44) and recursion (2.36).

terms of the unity vector $\mathbf{1}$, can be repeated for any other choice of $\overline{\boldsymbol{\sigma}}_{i+1}$, say $\overline{\boldsymbol{\sigma}}_{i+1} = \overline{\boldsymbol{\sigma}}$ for some $\overline{\boldsymbol{\sigma}}$, to conclude that the same recursion (2.44) still holds with $\mathbf{1}$ replaced by $\overline{\boldsymbol{\sigma}}$. For instance, if we choose $\overline{\boldsymbol{\sigma}} = \boldsymbol{\lambda}$, then the top entry of the resulting state vector $\mathcal{W}_i$ will correspond to the learning curve of the adaptive filter. In Subsection 2.5.2 we shall use this remark to describe more fully the learning behavior of adaptive filters with data normalization.

2.3.3 Mean-Square Stability

From the results in the above two subsections, we conclude that the LMS filter will be stable in the mean and mean-square senses if, and only if, μ satisfies (2.39) and guarantees the stability of the matrix $\overline{\boldsymbol{F}}$ (i.e., all the eigenvalues of $\overline{\boldsymbol{F}}$ should lie inside the unit circle). Since $\overline{\boldsymbol{F}}$ is easily seen to be nonnegative definite in this case, we only need to worry about guaranteeing that its eigenvalues be smaller than unity.

Let us write $\overline{\boldsymbol{F}}$ in the form

$$\overline{\boldsymbol{F}} = \boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{B}$$

where the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are both positive-definite and given by

$$\boldsymbol{A} \stackrel{\Delta}{=} 2\boldsymbol{\Lambda}, \qquad \boldsymbol{B} \stackrel{\Delta}{=} 2\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T \tag{2.45}$$

It follows from the argument in Appendix A that the eigenvalues of $\overline{\boldsymbol{F}}$ will be upper bounded by one if, and only if, the parameter μ satisfies

$$0 < \mu < \frac{1}{\lambda_{\max}\left(\boldsymbol{A}^{-1}\boldsymbol{B}\right)} \tag{2.46}$$

in terms of the maximum eigenvalue of $\boldsymbol{A}^{-1}\boldsymbol{B}$ (all eigenvalues of $\boldsymbol{A}^{-1}\boldsymbol{B}$ are real and positive). The above upper bound on μ can also be interpreted as the smallest positive scalar η that makes $(\boldsymbol{I} - \eta\boldsymbol{A}^{-1}\boldsymbol{B})$ singular. Let us denote this value of η by η^o. Combining (2.46) with (2.39) we find that μ should satisfy

$$0 < \mu < \min\{2/\lambda_{\max}(\boldsymbol{R}), \eta^o\}$$

We can be more specific about η^o and show that it is smaller than $1/\lambda_{\max}(\boldsymbol{R})$. Actually, we can characterize η^o in terms of the eigenvalues of $\boldsymbol{R}$ as follows. Using the definitions (2.45)

for $\boldsymbol{A}$ and $\boldsymbol{B}$, it can be verified that for all $\eta \in (0, 1/\lambda_{\max})$,

$$\det\left(\boldsymbol{I} - \eta \boldsymbol{A}^{-1}\boldsymbol{B}\right) = \left(1 - \frac{\boldsymbol{\lambda}^T}{2}\left[\eta^{-1}\boldsymbol{I} - \boldsymbol{\Lambda}\right]^{-1}\mathbf{1}\right) \cdot \det\left(\boldsymbol{I} - \eta\boldsymbol{\Lambda}\right)$$

The values of $\eta \in (0, 1/\lambda_{\max})$ that result in $\det\left(\boldsymbol{I} - \eta A^{-1}B\right) = 0$ should therefore satisfy

$$\frac{\boldsymbol{\lambda}^T}{2}\left(2\eta^{-1}\boldsymbol{I} - \boldsymbol{\Lambda}\right)^{-1}\mathbf{1} = 1$$

i.e.,

$$\frac{1}{2}\sum_{k=1}^{M}\frac{\lambda_k \eta}{1 - \eta\lambda_k} = 1$$

This equality has a unique solution η^o inside the interval $(0, 1/\lambda_{\max})$. This is because the function

$$f(\eta) \stackrel{\Delta}{=} \frac{1}{2}\sum_{k=1}^{M}\frac{\lambda_k \eta}{1 - \eta\lambda_k} = 1$$

is monotonically increasing in the interval $(0, 1/\lambda_{\max})$. Moreover, it evaluates to 0 at $\eta = 0$ and becomes unbounded as $\eta \to 1/\lambda_{\max}$. We therefore conclude that LMS is stable in the mean- and mean-square senses for all step-sizes μ satisfying

$$\boxed{\frac{1}{2}\sum_{k=1}^{M}\left(\frac{\lambda_k \mu}{1 - \mu\lambda_k}\right) < 1}$$

2.3.4 Steady-State Performance

Once filter stability has been guaranteed, we can proceed to derive expressions for the steady-state value of the mean-square error (MSE) and the mean-square deviation (MSD). To this end, note that in steady-state, we have that for any vector $\boldsymbol{\sigma}$

$$\lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_{i+1}\|_{\sigma}^2 = \lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_i\|_{\sigma}^2$$

Thus, in the limit, (2.38) leads to

$$\lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_i\|_{(I-F)\overline{\sigma}_\infty}^2 = \mu^2\sigma_v^2 E\|\overline{\mathbf{u}}_i\|_{\overline{\sigma}_\infty}^2 \tag{2.47}$$

Here, $\overline{\boldsymbol{\Sigma}}_\infty = \mathrm{diag}(\overline{\boldsymbol{\sigma}}_\infty)$ denotes the boundary condition of the recursion (2.32), which we are free to choose.

Now, in order to evaluate the MSE, we first recall that it is defined by

$$\mathsf{MSE} = \lim_{i\to\infty} Ee_a^2(i)$$

which, in view of the independence assumption AI, is also given by

$$\mathsf{MSE} = \lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_i\|_\lambda^2$$

This is because

$$Ee_a^2(i) = E\|\tilde{\boldsymbol{w}}_i\|_{u_i^T u_i} = E\|\tilde{\boldsymbol{w}}_i\|_R = E\|\overline{\boldsymbol{w}}_i\|_\lambda^2$$

Therefore, to obtain the MSE, we should choose $\overline{\boldsymbol{\sigma}}_\infty$ in (2.47) so that $\left(\boldsymbol{I} - \overline{\boldsymbol{F}}\right)\overline{\boldsymbol{\sigma}}_\infty = \boldsymbol{\lambda}$, in which case we get

$$\mathsf{MSE} = \mu^2\sigma_v^2 E\left[\|\overline{\mathbf{u}}_i\|^2_{(I-\overline{F})^{-1}\lambda}\right] \tag{2.48}$$

A more explicit expression for the MSE can be obtained by using the matrix inversion lemma to evaluate the matrix inverse that appears in (2.48). Doing so leads to the well-known result:

$$\boxed{\mathsf{MSE} = \frac{\sigma_v^2 \sum_{i=1}^M \frac{\mu\lambda_i}{2-2\mu\lambda_i}}{1 - \sum_{i=1}^M \frac{\mu\lambda_i}{2-2\mu\lambda_i}}}$$

The MSD can be calculated along the same lines by noting that

$$\mathsf{MSD} \;=\; \lim_{i\to\infty} E\|\tilde{\boldsymbol{w}}_i\|^2 = \lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_i\|_I^2 = \lim_{i\to\infty} E\|\overline{\boldsymbol{w}}_i\|_1^2$$

The above means that in order to obtain an expression for the MSD we should now choose $\overline{\boldsymbol{\sigma}}_\infty$ in (2.47) such that $\overline{\boldsymbol{\sigma}}_\infty = \left(\boldsymbol{I} - \overline{\boldsymbol{F}}\right)^{-1}\mathbf{1}$, which yields

$$\mathsf{MSD} = \mu^2\sigma_v^2 E\left[\|\overline{\mathbf{u}}_i\|^2_{(I-\overline{F})^{-1}1}\right]$$

Just like the expression for the MSE, we can use the matrix inversion lemma to get an explicit expression for $(\boldsymbol{I} - \boldsymbol{F})^{-1}\mathbf{1}$ and subsequently for the MSD,

$$\boxed{\mathsf{MSD} = \frac{\sigma_v^2 \sum_{i=1}^M \frac{\mu}{2-2\mu\lambda_i}}{1 - \sum_{i=1}^M \frac{\mu\lambda_i}{2-2\mu\lambda_i}}}$$

Both of these steady-state expressions were derived in [36]. Here we arrived at the expressions as a byproduct of a framework that can also handle a variety of data-normalized adaptive filters (see Section 2.5). In addition, observe how the expressions for MSE and MSD can be obtained simply by conveniently choosing different values for the boundary condition $\overline{\boldsymbol{\sigma}}_\infty$.

2.4 Normalized LMS with Gaussian Regressors

We now consider the normalized LMS algorithm, for which $g(\mathbf{u}_i) = \epsilon + \|\mathbf{u}_i\|^2$ with $\epsilon \geq 0$. For this choice of $g(\mathbf{u}_i)$, recursion (2.32) becomes

$$\begin{aligned}\overline{\boldsymbol{\Sigma}}_i &= \overline{\boldsymbol{\Sigma}}_{i+1} - \mu E\left[\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{\epsilon+\|\overline{\mathbf{u}}_i\|^2}\right]\overline{\boldsymbol{\Sigma}}_{i+1} \\ &\quad -\mu\overline{\boldsymbol{\Sigma}}_{i+1}E\left[\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{\epsilon+\|\overline{\mathbf{u}}_i\|^2}\right] + \mu^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2_{\overline{\Sigma}_{i+1}}}{(\epsilon+\|\overline{\mathbf{u}}_i\|^2)^2}\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i\right]\end{aligned} \tag{2.49}$$

Progress in the analysis is now pending on the evaluation of the moments

$$\mathbf{A} \triangleq 2E\left[\frac{\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{\epsilon+\|\overline{\mathbf{u}}_i\|^2}\right], \quad \mathbf{B}' \triangleq E\left[\frac{\|\overline{\mathbf{u}}_i\|^2_{\overline{\Sigma}_{i+1}}\overline{\mathbf{u}}_i^T\overline{\mathbf{u}}_i}{(\epsilon+\|\overline{\mathbf{u}}_i\|^2)^2}\right] \tag{2.50}$$

Although the individual elements of $\overline{\mathbf{u}}_i$ are independent, no closed form expressions for $\mathbf{A}$ and $\mathbf{B}'$ are available. However, we can carry out the analysis in terms of these matrices as follows. First, we argue in Appendix B that $\mathbf{A}$ is diagonal. We also show that if $\overline{\boldsymbol{\Sigma}}_{i+1}$ is diagonal, then so is $\mathbf{B}'$ and that

$$\operatorname{diag}\left(\mathbf{B}'\right) = \mathbf{B}\operatorname{diag}\left(\overline{\boldsymbol{\Sigma}}_{i+1}\right)$$

where $\mathbf{B}$ is the diagonal matrix

$$\mathbf{B} = E\left[\frac{\left(\overline{\mathbf{u}}_i\odot\overline{\mathbf{u}}_i\right)^T\left(\overline{\mathbf{u}}_i\odot\overline{\mathbf{u}}_i\right)}{\left(\epsilon+\|\overline{\mathbf{u}}_i\|^2\right)^2}\right]$$

Here, the notation $\odot$ denotes an element-by-element (Hadamard) product.[3] Thus, the successive $\overline{\boldsymbol{\Sigma}}_i$'s in recursion (2.49) will also be diagonal if the boundary condition is. Subsequently, as in the LMS case, we can again obtain a recursive relation for their diagonal entries of the form $\overline{\boldsymbol{\sigma}}_i = \overline{\boldsymbol{F}}\overline{\boldsymbol{\sigma}}_{i+1}$, where $\overline{\boldsymbol{F}}$ retains the same form, namely,

$$\overline{\boldsymbol{F}} = \boldsymbol{I} - \mu\mathbf{A} + \mu^2\mathbf{B}$$

Mean-square stability now requires that the step-size μ be chosen such that $\overline{\boldsymbol{F}}$ is a stable matrix (i.e., all its eigenvalues should be strictly inside the unit circle). For NLMS, it can be verified that $\mu < 2$ is a sufficient condition for this fact to hold, as can be seen from the following argument.

Choosing $\overline{\boldsymbol{\Sigma}}_{i+1} = \boldsymbol{I}$ we have

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2 = E\|\overline{\boldsymbol{w}}_i\|^2_{\overline{\Sigma}_i} \;+\; \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2}{(\epsilon + \|\overline{\mathbf{u}}_i\|^2)^2}\right]$$

and

$$\overline{\boldsymbol{\Sigma}}_i = \boldsymbol{I} - \mu\boldsymbol{A} \;+\; \mu^2\boldsymbol{B}'$$

Obviously, $\boldsymbol{B}' \leq \boldsymbol{A}/2$ so that

$$\boldsymbol{\Sigma}_i \;\leq\; \boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{A}/2$$

and, hence,

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2 \;\;\leq\;\; E\left[\overline{\boldsymbol{w}}_i^T(\boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{A}/2)\overline{\boldsymbol{w}}_i\right] \;+\; \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2}{(\epsilon + \|\overline{\mathbf{u}}_i\|^2)^2}\right]$$

Now it is clear that $0 < \lambda(\boldsymbol{A}/2) < 1$. Moreover, over the interval $0 < \mu < 2$, it holds that

$$\boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{A}/2 \;\leq\; \underbrace{\left(1 - 2\mu\lambda_{\min}(\boldsymbol{A}/2) + \mu^2\lambda_{\min}(\boldsymbol{A}/2)\right)}_{\alpha}\boldsymbol{I}$$

from which we conclude that

$$E\|\overline{\boldsymbol{w}}_{i+1}\|^2 \leq \alpha E\|\overline{\boldsymbol{w}}_i\|^2 \;+\; \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2}{(\epsilon + \|\overline{\mathbf{u}}_i\|^2)^2}\right]$$

[3]For two row vectors $\{\mathbf{x}, \mathbf{y}\}$, the quantity $\mathbf{x} \odot \mathbf{y}$ is a row vector with elementwise products — see [59].

where the scalar coefficient α is positive and strictly less than one for $0 < \mu < 2$. It follows that $E\|\overline{\boldsymbol{w}}_i\|^2$ remains bounded for all i, as desired. It is also straightforward to verify from

$$E\overline{\boldsymbol{w}}_{i+1} = \left[\boldsymbol{I} - \mu E\left(\frac{\overline{\mathbf{u}_i}^T\overline{\mathbf{u}_i}}{\epsilon + \|\overline{\mathbf{u}_i}\|^2}\right)\right] \cdot E\overline{\boldsymbol{w}}_i$$

that $\mu < 2$ guarantees filter stability in the mean as well (just note that $\overline{\mathbf{u}_i}^T\overline{\mathbf{u}_i}/(\epsilon + \|\overline{\mathbf{u}_i}\|^2)$ is a rank-one matrix whose largest eigenvalue is smaller than one).

Finally, repeating the discussion we had for the steady-state performance of LMS, we arrive at the following expressions for the MSE and MSD of normalized LMS:

$$\mathsf{MSE} = \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2_{(I-\overline{F})^{-1}\lambda}}{\epsilon + \|\overline{\mathbf{u}}_i\|^2}\right]$$

$$\mathsf{MSD} = \mu^2\sigma_v^2 E\left[\frac{\|\overline{\mathbf{u}}_i\|^2_{(I-\overline{F})^{-1}1}}{\epsilon + \|\overline{\mathbf{u}}_i\|^2}\right]$$

These expressions hold for arbitrary colored Gaussian regressors.

The presentation so far illustrates how the energy-conservation approach can be used to perform transient analysis of LMS and its normalized version. Our contribution lies in the ability to perform the analysis in a unified manner. This can be appreciated, for example, by comparing the analysis of the normalized LMS algorithm in [92, 76, 90, 15, 13] with the analysis in the previous section. A substantial part of prior studies is often devoted to studying the multivariate moments of (2.50), and as a result, eventually resort to some whiteness assumption on the data. Our derivation bypasses this requirement. Moreover, earlier approaches do not seem to handle non-Gaussian regression data, which is discussed later in Section 2.5.

2.5 Data-Normalized Filters

We now consider general data-normalized adaptive filters of the form (2.26) and drop the Gaussian assumption AG. The analysis that follows shows how to extend the discussions of the previous two sections to this general scenario.

Our starting point are the mean and variance relations (2.27), (2.28), and (2.29).

2.5.1 Mean-Square-Analysis

For arbitrary regression data, we can no longer guarantee that the data moments

$$E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right], \qquad E\left[\frac{\|\mathbf{u}_i\|_{\sum}^2\mathbf{u}_i^T\mathbf{u}_i}{g^2[\mathbf{u}_i]}\right]$$

are jointly diagonalizable (as we had, for example, in the case of LMS with Gaussian regressors). Consequently, $\boldsymbol{\Sigma}_i$ need not be diagonal even if $\boldsymbol{\Sigma}_{i+1}$ is, i.e., these matrices can no more be fully characterized by their diagonal elements alone. Still, we can perform mean-square analysis by replacing the diag operation with the vec operation, which transforms a matrix into a column vector by stacking all its columns on top of each other.

Let

$$\boldsymbol{\sigma}_{i+1} \triangleq \text{vec}\,(\boldsymbol{\Sigma}_{i+1})$$

Then using the Kronecker product notation (e.g., [59]) and the following property, for arbitrary matrices $\{\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{\Sigma}\}$,

$$\text{vec}(\mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}) = (\mathbf{Q}^T\otimes\mathbf{P})\text{vec}(\boldsymbol{\Sigma})$$

it is straightforward to verify that the recursion (2.28) for $\boldsymbol{\Sigma}_i$ transforms into the linear vector relation

$$\boxed{\boldsymbol{\sigma}_i = \boldsymbol{F}\boldsymbol{\sigma}_{i+1}}$$

where the coefficient matrix $\boldsymbol{F}$ is now $M^2 \times M^2$ and is given by

$$\boldsymbol{F} \;\triangleq\; \boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{B} \tag{2.51}$$

with the $M^2 \times M^2$ symmetric matrices $\{\boldsymbol{A}, \boldsymbol{B}\}$ defined by

$$\begin{aligned}
\boldsymbol{A} &= \left(E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right]\otimes\boldsymbol{I}_M\right) + \left(\boldsymbol{I}_M\otimes E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right]\right)\\
\boldsymbol{B} &= E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i\otimes\mathbf{u}_i^T\mathbf{u}_i}{g^2[\mathbf{u}_i]}\right]
\end{aligned}$$

In particular, $\boldsymbol{A}$ is positive-definite and $\boldsymbol{B}$ is nonnegative-definite. Introduce also the $M\times M$ matrix

$$\boldsymbol{P} \triangleq E\left[\frac{\mathbf{u}_i^T\mathbf{u}_i}{g[\mathbf{u}_i]}\right]$$

which appears in the mean weight-error recursion (2.29) and in the expression for $\boldsymbol{A}$.

It follows that, in terms of the vec notation, the variance relation (2.27) becomes

$$\boxed{E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\sigma_{i+1}}\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{F\sigma_{i+1}}\right] + \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|^2_{\sigma_{i+1}}}{g^2[\mathbf{u}_i]}\right]} \tag{2.52}$$

Now, contrary to the Gaussian LMS case, the matrix $\boldsymbol{F}$ is no longer guaranteed to be nonnegative-definite. It is shown in Appendix A that the condition $-1 < \lambda(F) < 1$ can be enforced for values of μ in the range:

$$0 < \mu < \min\left\{\frac{1}{\lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B})},\ \frac{1}{\max\left\{\lambda(\boldsymbol{L}) \in \mathbb{R}^+\right\}}\right\} \tag{2.53}$$

where the second condition is in terms of the largest positive real eigenvalue of the following block matrix,

$$\boxed{\boldsymbol{L} \triangleq \begin{bmatrix} \boldsymbol{A}/2 & -\boldsymbol{B}/2 \\ \boldsymbol{I}_M & 0 \end{bmatrix}}$$

when it exists. Since $\boldsymbol{L}$ is not symmetric, its eigenvalues may not be positive or even real. If $\boldsymbol{L}$ does not have any real positive eigenvalue, then the corresponding condition is removed from (2.53) and we only require $\mu < 1/\lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B})$. Condition (2.53) can be grouped together with the requirement $\mu < 2/\lambda_{\max}(\boldsymbol{P})$, which guarantees convergence in the mean, so that

$$\boxed{\mu\ < \min\left\{\frac{2}{\lambda_{\max}(\boldsymbol{P})},\ \frac{1}{\lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B})},\ \frac{1}{\max\left\{\lambda(\boldsymbol{L}) \in \mathbb{R}^+\right\}}\right\}} \tag{2.54}$$

Moreover, the same argument that we used in the LMS case in Section 2.3 would show that the transient behavior of data-normalized filters is characterized by the M^2-dimensional state-space model:[4]

$$\boxed{\mathcal{W}_{i+1} = \mathcal{F}\mathcal{W}_i + \mu^2\sigma_v^2\mathcal{Y}} \tag{2.55}$$

[4]Observe how the order of the model, in the general case, is M^2 and not M as was the case in the previous two sections with Gaussian regressors.

where

$$\mathcal{F} = \begin{bmatrix} 0 & 1 & & & \\ 0 & 0 & 1 & & \\ 0 & 0 & 0 & 1 & \\ \vdots & & & & \\ 0 & 0 & 0 & & 1 \\ -p_0 & -p_1 & -p_2 & \cdots & -p_{M^2-1} \end{bmatrix}$$

with

$$p(x) \triangleq \det(x\boldsymbol{I} - \boldsymbol{F}) = x^{M^2} + \sum_{k=0}^{M^2-1} p_k x^k$$

denoting the characteristic polynomial of $\boldsymbol{F}$. Also, $\mathcal{W}_i$ and $\mathcal{Y}$ are the $M^2 \times 1$ vectors

$$\mathcal{W}_i \triangleq \begin{bmatrix} E\|\tilde{\boldsymbol{w}}_i\|_{\sigma}^2 \\ E\|\tilde{\boldsymbol{w}}_i\|_{F\sigma}^2 \\ E\|\tilde{\boldsymbol{w}}_i\|_{F^2\sigma}^2 \\ \vdots \\ E\|\tilde{\boldsymbol{w}}_i\|_{F^{(M^2-1)}\sigma}^2 \end{bmatrix}, \qquad \mathcal{Y} = \begin{bmatrix} E\left(\|\mathbf{u}_i\|_{\sigma}^2/g^2[\mathbf{u}_i]\right) \\ E\left(\|\mathbf{u}_i\|_{F\sigma}^2/g^2[\mathbf{u}_i]\right) \\ E\left(\|\mathbf{u}_i\|_{F^2\sigma}^2/g^2[\mathbf{u}_i]\right) \\ \vdots \\ E\left(\|\mathbf{u}_i\|_{F^{(M^2-1)}\sigma}^2/g^2[\mathbf{u}_i]\right) \end{bmatrix}$$

for any $\boldsymbol{\sigma}$ of interest, e.g., more commonly, $\boldsymbol{\sigma} = \mathbf{1}$ or $\boldsymbol{\sigma} = \mathbf{r}$, where $\mathbf{r} = \mathrm{vec}(\boldsymbol{R})$.

Moreover, steady-state analysis can be carried out along the same lines of Subsection 2.3.4. Thus, assuming the filter reaches steady-state, recursion (2.52) becomes in the limit

$$\lim_{i\to\infty} E\|\tilde{\boldsymbol{w}}_i\|_{(I-F)\sigma_\infty}^2 = \mu^2\sigma_v^2 E\left[\frac{\|\mathbf{u}_i\|_{\sigma_\infty}^2}{g^2[\mathbf{u}_i]}\right]$$

in terms of the boundary condition $\boldsymbol{\sigma}_\infty$, which we are free to choose. This expression allows us to evaluate the steady-state value of $E\|\tilde{\boldsymbol{w}}_i\|_S^2$ for any symmetric weighting $\mathbf{S}$, by choosing $\boldsymbol{\sigma}_\infty$ such that $(\boldsymbol{I} - \boldsymbol{F})\boldsymbol{\sigma}_\infty = \mathrm{vec}(\mathbf{S})$. In particular, the EMSE corresponds to the choice $\mathbf{S} = \mathbf{R}$, i.e., $\boldsymbol{\sigma}_\infty = (\boldsymbol{I} - \boldsymbol{F})^{-1}\mathrm{vec}(\mathbf{R})$. Likewise, the MSD is obtained by choosing $\mathbf{S} = \boldsymbol{I}$, i.e., $\boldsymbol{\sigma}_\infty = (\boldsymbol{I} - \boldsymbol{F})^{-1}\mathrm{vec}(\boldsymbol{I})$. We summarize these results in the following statement, which holds for arbitrary input distributions and scalar data nonlinearities.

Theorem 2 (Scalar nonlinearities) *Consider an adaptive filter of the form*

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu\frac{\mathbf{u}_i^T}{g[\mathbf{u}_i]}e(i), \quad i \geq 0$$

where $e(i) = d(i) - \mathbf{u}_i \boldsymbol{w}_i$ *and* $d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$. *Assume that the sequences* $\{v(i), \mathbf{u}_i\}$ *are* iid *and mutually independent. Then the filter is stable in the mean and mean-square senses if the step-size* μ *satisfies (2.54). Moreover, the resulting* EMSE *and* MSD *are given by*

$$\begin{aligned} \mathsf{EMSE} &= \mu^2 \sigma_v^2 E \left[\frac{\|\mathbf{u}_i\|^2_{(I-F)^{-1}\mathrm{vec}(R)}}{g^2[\mathbf{u}_i]} \right] \\ \mathsf{MSD} &= \mu^2 \sigma_v^2 E \left[\frac{\|\mathbf{u}_i\|^2_{(I-F)^{-1}\mathrm{vec}(I)}}{g^2[\mathbf{u}_i]} \right] \end{aligned}$$

where $\boldsymbol{F}$ *is defined by (2.51).*

♢

2.5.2 Learning Curves

The learning curve of an adaptive filter refers to the time evolution of $Ee_a^2(i)$; its steady-state value is the MSE. Now since $Ee_a^2(i) = E\|\tilde{\boldsymbol{w}}_i\|_R^2$, the learning curve can be evaluated by computing $E\|\tilde{\boldsymbol{w}}_i\|_R^2$ for each i. This task can be accomplished recursively from relation (2.52) by choosing the boundary condition $\boldsymbol{\sigma}_{i+1}$ as $\mathbf{r} = \mathrm{vec}(\mathbf{R})$. Indeed, iterating (2.52) with this choice of $\boldsymbol{\sigma}_{i+1}$, and assuming $\boldsymbol{w}_o = 0$, we find that

$$E\|\tilde{\boldsymbol{w}}_{i+1}\|_r^2 = \|\boldsymbol{w}^o\|^2_{F^{i+1}r} + \mu^2 \sigma_v^2 E \left[\frac{\|\mathbf{u}_i\|^2_{(I+F+\cdots+F^i)r}}{g^2[\mathbf{u}_i]} \right]$$

that is,

$$E\|\tilde{\boldsymbol{w}}_{i+1}\|_r^2 = \|\boldsymbol{w}^o\|^2_{a_i} + \mu^2 \sigma_v^2 b_i$$

where the vector $\mathbf{a}_i$ and the scalar b_i satisfy the recursions

$$\begin{aligned} \mathbf{a}_i &= \boldsymbol{F}\mathbf{a}_{i-1}, \quad \mathbf{a}_{-1} = \mathbf{r} \\ b_i &= b_{i-1} + E \left[\frac{\|\mathbf{u}_i\|^2_{a_{i-1}}}{g^2[\mathbf{u}_i]} \right], \quad b_{-1} = 0 \end{aligned}$$

Using these definitions for $\{\mathbf{a}_i, b_i\}$, it is easy to verify that

$$\boxed{Ee_a^2(i) = Ee_a^2(i-1) \;+\; \|\boldsymbol{w}^o\|^2_{F^i(F-I)r} \;+\; \mu^2 \sigma_v^2 E \left[\frac{\|\mathbf{u}_i\|^2_{F^i r}}{g^2[\mathbf{u}_i]} \right]}$$

which describes the learning curve of a data-normalized adaptive filter.

2.6 Matrix Nonlinearities

In this section we extend the earlier results to the case in which the function $g[\mathbf{u}_i]$ is matrix-valued rather than scalar-valued. To motivate this extension, consider the sign-regressor algorithm (e.g., [34]):

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu \text{sgn}[\mathbf{u}_i]^T e(i)$$

where the sgn operates on the individual elements of $\mathbf{u}_i$. This is in contrast to the discussions in the previous sections where all the elements of $\mathbf{u}_i$ were normalized by the same data nonlinearity. Other examples of matrix nonlinearities can be found, e.g., in [14, 33, 42].

The above update is a special case of more general updates of the form:

$$\boxed{\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu \mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T e(i)} \tag{2.56}$$

where $\mathbf{H}[\mathbf{u}_i]$ denotes an $M \times M$ matrix nonlinearity.

2.6.1 Energy Relation

We first show how to extend the energy relation of Theorem 1 to the more general class of algorithms (2.56) with matrix data nonlinearities. Our starting point is the adaptation equation (2.56), which can be written in terms of the weight error vector $\tilde{\boldsymbol{w}}_i$ as

$$\tilde{\boldsymbol{w}}_{i+1} = \tilde{\boldsymbol{w}}_i - \mu \mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T e(i) \tag{2.57}$$

By pre-multiplying both sides of (2.57) by $\mathbf{u}_i\boldsymbol{\Sigma}$, we see that the estimation errors $e_a^{\Sigma}(i)$, $e_p^{\Sigma}(i)$, and $e(i)$ are related by

$$e_p^{\Sigma}(i) = e_a^{\Sigma}(i) - \mu \|\mathbf{u}_i\|_{\Sigma H}^2 e(i) \tag{2.58}$$

Moreover, the two sides of (2.57) should have the same weighted-energy, i.e.,

$$\tilde{\boldsymbol{w}}_{i+1}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{w}}_{i+1} = \left(\tilde{\boldsymbol{w}}_i - \mu \mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T e(i)\right)^T \boldsymbol{\Sigma} \left(\tilde{\boldsymbol{w}}_i - \mu \mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T e(i)\right)$$

so that

$$\begin{aligned}\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2 &= \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 - 2\mu e(i)\mathbf{u}_i\mathbf{H}[\mathbf{u}_i]\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}_i + \mu^2 e^2(i)\mathbf{u}_i\mathbf{H}[\mathbf{u}_i]\boldsymbol{\Sigma}\mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T \\ &= \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 - 2\mu e_a^{H\Sigma}(i)e(i) + \mu^2 e^2(i)\|\mathbf{u}_i\|_{H\Sigma H}^2 \end{aligned} \tag{2.59}$$

This form of the energy relation is analogous to (2.15). As it stands, (2.59) is just what we need for mean-square analysis. For completeness, though, we develop a cleaner form of (2.59) – a form similar to (2.10). To this end, notice that upon replacing $\boldsymbol{\Sigma}$ by $\mathbf{H}\boldsymbol{\Sigma}$ in (2.58), we get

$$\mu\|\mathbf{u}_i\|_{H\Sigma H}^2 e(i) = e_a^{H\Sigma}(i) - e_p^{H\Sigma}(i)$$

or, by incorporating the defining expression (2.9) of $\overline{\mu}_{(.)}(i)$,

$$\mu e(i) = \overline{\mu}_{H\Sigma H}(i)\left(e_a^{H\Sigma}(i) - e_p^{H\Sigma}(i)\right) \tag{2.60}$$

Substituting (2.60) into (2.59) produces the desired energy relation form

$$\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2 + \overline{\mu}_{H\Sigma H}(i)\left|e_a^{H\Sigma}(i)\right|^2 = \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 + \overline{\mu}_{H\Sigma H}(i)\left|e_p^{H\Sigma}(i)\right|^2$$

2.6.2 Mean-Square Analysis

To perform mean-square analysis, we start with (2.59). Bearing in mind the independence assumption on the noise AN and the fact that $e(i) = e_a(i) + v(i)$, (2.59) reads under expectation

$$\begin{aligned}E\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma_{i+1}}^2 = E\|\tilde{\boldsymbol{w}}_i\|_{\Sigma_{i+1}}^2 - 2\mu E\left[e_a^{H\Sigma_{i+1}}(i)e_a(i)\right] \;+ \\ \mu^2 E\left[e_a^2(i)\|\mathbf{u}_i\|_{H\Sigma_{i+1}H}^2\right] \;+\; \mu^2\sigma_v^2 E\left[\|\mathbf{u}_i\|_{H\Sigma_{i+1}H}^2\right]\end{aligned}$$

where the weight $\boldsymbol{\Sigma}$ was replaced by the time-indexed weight $\boldsymbol{\Sigma}_{i+1}$. If we further invoke the polarization identity (2.12), we get

$$e_a^{H\Sigma_{i+1}}(i)e_a(i) \;=\; \|\tilde{\boldsymbol{w}}_i\|_{\Sigma_{i+1}Hu_i^Tu_i}^2 = \|\tilde{\boldsymbol{w}}_i\|_{u_i^Tu_iH\Sigma_{i+1}}^2 \quad\text{and}\quad e_a^2(i) = \|\tilde{\boldsymbol{w}}_i\|_{u_i^Tu_i}^2$$

These equations, together with the linearity property (2.11) and the independence assumption AI, yield the following result.

Theorem 3 (Matrix nonlinearities) *Consider an adaptive filter of the form*

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu \mathbf{H}[\mathbf{u}_i]\mathbf{u}_i^T e(i), \;\; i \geq 0$$

where $e(i) = d(i) - \mathbf{u}_i \boldsymbol{w}_i$ *and* $d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$. *Assume that the sequences* $\{v(i), \mathbf{u}_i\}$ *are* iid *and mutually independent. Then it holds that*

$$E\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma_{i+1}} = E\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_i} + \mu^2\sigma_v^2 E\left[\|\mathbf{u}_i\|^2_{H\Sigma_{i+1}H}\right] \tag{2.61}$$

where

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i+1} - \mu\boldsymbol{\Sigma}_{i+1}E\left[\mathbf{H}\mathbf{u}_i^T\mathbf{u}_i\right] - \mu E\left[\mathbf{u}_i^T\mathbf{u}_i\mathbf{H}\right]\boldsymbol{\Sigma}_{i+1} + \mu^2 E\left[\|\mathbf{u}_i\|^2_{H\Sigma_{i+1}H}\mathbf{u}_i^T\mathbf{u}_i\right] \tag{2.62}$$

In addition, the stability condition and the MSE *and* MSD *expressions of Theorem 2 apply here as well with* $\{\boldsymbol{A}, \boldsymbol{B}\}$ *replaced by*

$$\begin{aligned} \boldsymbol{A} &= \left(E[\mathbf{u}_i^T\mathbf{u}_i\mathbf{H}]\otimes\boldsymbol{I}\right) + \left(\boldsymbol{I}\otimes E[\mathbf{u}_i^T\mathbf{u}_i\mathbf{H}]\right) \\ \boldsymbol{B} &= E[\mathbf{u}_i^T\mathbf{u}_i\mathbf{H}\otimes\mathbf{u}_i^T\mathbf{u}_i\mathbf{H}] \end{aligned}$$

Moreover, the construction of the learning curve in Subsection 2.5.2 also extends to this case.

♢

Compared with some earlier studies (e.g., [34, 27, 14]), the above results hold without restricting the regression data to being Gaussian or white.

2.6.3 The Sign-Regressor Algorithm

To illustrate the application of the above results, we return to the sign-regressor recursion

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu\text{sgn}[\mathbf{u}_i]^T e(i)$$

In this case, the matrix nonlinearity $\mathbf{H}[\mathbf{u}_i]$ is implicitly defined by the identity:

$$\mathbf{u}_i\mathbf{H}[\mathbf{u}_i] = \text{sgn}[\mathbf{u}_i]$$

which in turn means that relations (2.61) and (2.62) become

$$E\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{\Sigma_{i+1}} = E\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_i} + \mu^2\sigma_v^2 E\left[\ \|\mathrm{sgn}[\mathbf{u}_i]\|^2_{\Sigma_{i+1}}\ \right] \tag{2.63}$$

and

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i+1} - \mu\boldsymbol{\Sigma}_{i+1}E\left[\mathrm{sgn}[\mathbf{u}_i]^T\mathbf{u}_i\right] - \mu E\left[\mathbf{u}_i^T\mathrm{sgn}[\mathbf{u}_i]\right]\boldsymbol{\Sigma}_{i+1} + \mu^2 E\left[\|\mathrm{sgn}[\mathbf{u}_i]\|^2_{\Sigma_{i+1}}\mathbf{u}_i^T\mathbf{u}_i\right]$$

Assume that the individual entries of the regressor $\mathbf{u}_i$ have variance σ_u^2. Assume also that $\mathbf{u}_i$ has a Gaussian distribution. Then it follows from Price's theorem [74] that[5]

$$E\left[\mathrm{sgn}[\mathbf{u}_i]^T\mathbf{u}_i\right] = \sqrt{\frac{2}{\pi\sigma_u^2}}\mathbf{R}$$

which leads to

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i+1} - \mu\sqrt{\frac{2}{\pi\sigma_u^2}}\boldsymbol{\Sigma}_{i+1}\mathbf{R} - \mu\sqrt{\frac{2}{\pi\sigma_u^2}}\mathbf{R}\boldsymbol{\Sigma}_{i+1} + \mu^2 E\left[\|\mathrm{sgn}[\mathbf{u}_i]\|^2_{\Sigma_{i+1}}\mathbf{u}_i^T\mathbf{u}_i\right] \tag{2.64}$$

Now observe that $\|\mathrm{sgn}[\mathbf{u}_i]\|^2_{\Sigma_{i+1}} = \mathrm{Tr}(\boldsymbol{\Sigma}_{i+1})$ whenever $\boldsymbol{\Sigma}_{i+1}$ is diagonal. Thus assume we choose $\boldsymbol{\Sigma}_{i+1} = \boldsymbol{I}$. Then the expression for $\boldsymbol{\Sigma}_i$ becomes

$$\boldsymbol{\Sigma}_i = \boldsymbol{I} + \mu\left(\mu M - 2\sqrt{\frac{2}{\pi\sigma_u^2}}\right)\mathbf{R}$$

while (2.63) becomes

$$E\|\tilde{\boldsymbol{w}}_{i+1}\|^2 = E\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_i} + \mu^2\sigma_v^2 M \tag{2.65}$$

It is now easy to verify that $E\|\overline{\boldsymbol{w}}_{i+1}\|^2$ converges provided that $\lambda_{\max}(\boldsymbol{\Sigma}_i) < 1$ or, equivalently,

$$\mu < \sqrt{\frac{8}{\pi\sigma_u^2}}\frac{1}{M}$$

This is the same condition derived in [34].

[5]The theorem can be used to show that for two jointly zero-mean Gaussian real-valued random variables x and y, it holds that $E\left(x\mathrm{sgn}(y)\right) = \sqrt{\frac{2}{\pi}}\frac{1}{\sigma_y}E\left(xy\right)$.

To evaluate the MSE we observe from (2.65) that, in steady-state,

$$\mu\left(2\sqrt{\frac{2}{\pi\sigma_u^2}}\;-\;\mu M\right)\lim_{i\to\infty}E\|\boldsymbol{w}_i\|_R^2=\mu^2\sigma_v^2M$$

so that

$$\mathsf{MSE}\;\;=\;\;\frac{\mu\sigma_v^2M}{\sqrt{\frac{8}{\pi\sigma_u^2}}-\mu M}$$

which is again the same expression from [34].

2.7 Simulations

Throughout this section, the system to be identified is an FIR channel of length 4. The input $u(i)$ is generated by passing an iid uniform process $x(i)$ through a first-order model,

$$u(i)=au(i-1)+x(i) \tag{2.66}$$

By varying the value of a, we obtain processes $u(i)$ of different colors. We simulate the choices $a=0.2$ and $a=0.9$. The input sequence that is feeding the adaptive filter therefore has a correlated uniform distribution. The output of the channel is contaminated by an iid Gaussian additive noise at an SNR level of 30 dB.

Figures 2.1 and 2.2 show the resulting theoretical and simulated learning and MSD curves for both cases of $a=0.2$ and $a=0.9$. The simulated curves are obtained by averaging over 200 experiments, while the theoretical curves are obtained from the state-space model (2.55). It is seen that there is a good match between theory and practice.

Figure 2.3 examines the stability bound (2.54); it plots the filter EMSE as a function of the step-size using the theoretical expression from Theorem 2, in addition to a simulated EMSE. The bound on the step-size is also indicated.

2.8 Concluding Remarks

In this chapter, we developed a framework for the transient analysis of adaptive filters with general data nonlinearities (both scalar-valued and matrix-valued). The approach relies on energy conservation arguments. By suitably choosing the boundary condition of the

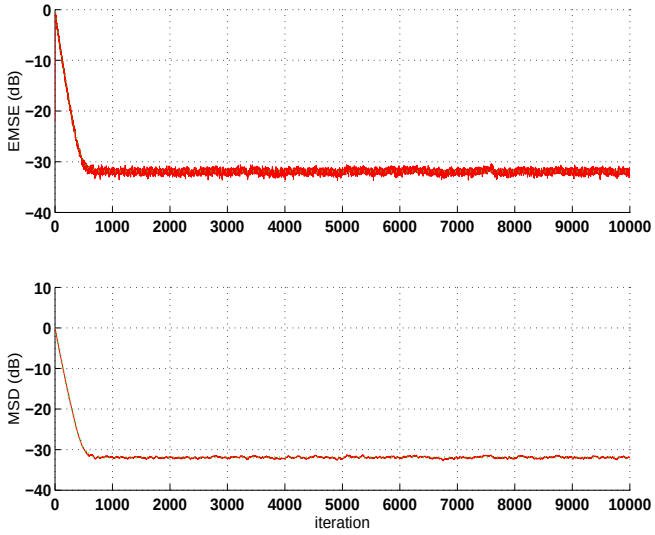

Figure 2.1: Theoretical and simulated learning and MSD curves for LMS using correlated uniform input data and $a = 0.2$

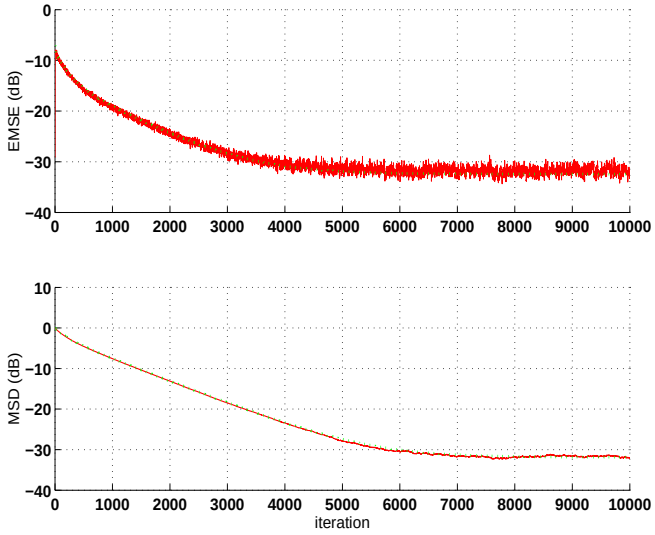

Figure 2.2: Theoretical and simulated learning curves for LMS using correlated uniform input data and $a = 0.9$

weighting matrix recursion, we can obtain MSE and MSD results, and also conditions for mean-square stability. We may add that extensions to leaky algorithms and to tracking

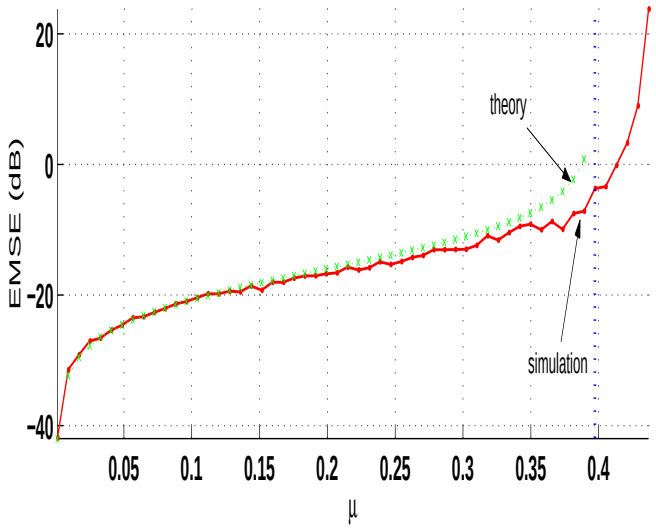

Figure 2.3: Theoretical and simulated EMSE vs. μ for LMS as a function of the step-size for correlated uniform input with $a = 0.2$

analysis are possible and are treated in, e.g., [4, 83].

2.9 APPENDIX A: Condition for Mean-Square Stability

Consider the matrix form $\boldsymbol{F} = \boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{B}$ with $\boldsymbol{A} > 0$, $\boldsymbol{B} \geq 0$, and $\mu > 0$. We would like to determine conditions on μ in order to guarantee that the eigenvalues of $\boldsymbol{F}$ satisfy $-1 < \lambda(\boldsymbol{F}) < 1$.

First, in order to guarantee $\lambda(\boldsymbol{F}) < 1$, the step-size μ should be such that $\boldsymbol{F} < \boldsymbol{I}$ or, equivalently, $\boldsymbol{A} - \mu\boldsymbol{B} > 0$. This condition is equivalent to requiring $\boldsymbol{I} - \mu\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-*/2} > 0$. But since the matrices $\boldsymbol{A}^{-1}\boldsymbol{B}$ and $\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-*/2}$ are similar, we conclude that μ should satisfy $\mu < 1/\lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B})$.

In order to enforce $\lambda(\boldsymbol{F}) > -1$, the step-size μ should be such that $\boldsymbol{G}(\mu) = 2\boldsymbol{I} - \mu\boldsymbol{A} + \mu^2\boldsymbol{B} > 0$. When $\mu = 0$, the eigenvalues of $\boldsymbol{G}$ are positive and equal to 2. As μ increases, the eigenvalues of $\boldsymbol{G}$ vary continuously with μ. Therefore, an upper bound on μ that guarantees $\boldsymbol{G}(\mu) > 0$ is determined by the smallest μ that makes $\boldsymbol{G}(\mu)$ singular.

Now the determinant of $\boldsymbol{G}(\mu)$ is equal to the determinant of the block matrix

$$\boldsymbol{K}(\mu) \triangleq \begin{bmatrix} 2\boldsymbol{I} - \mu\boldsymbol{A} & \mu\boldsymbol{B} \\ -\mu\boldsymbol{I} & \boldsymbol{I} \end{bmatrix}$$

Moreover, since

$$\boldsymbol{K}(\mu) \;=\; \begin{bmatrix} 2\boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} - \mu \begin{bmatrix} \boldsymbol{A}/2 & -\boldsymbol{B}/2 \\ \boldsymbol{I} & 0 \end{bmatrix} \right)$$

the condition $\det(\boldsymbol{K}(\mu)) = 0$ is equivalent to $\det(\boldsymbol{I} - \mu\boldsymbol{L}) = 0$, where

$$\boldsymbol{L} \triangleq \begin{bmatrix} \boldsymbol{A}/2 & -\boldsymbol{B}/2 \\ \boldsymbol{I} & 0 \end{bmatrix}$$

In this way, the smallest positive μ that results in $\det(\boldsymbol{K}(\mu)) = 0$ is given by

$$\mu < \frac{1}{\max\left\{\lambda(\boldsymbol{L}) \in \mathbb{R}^+\right\}}$$

This condition is in terms of the largest positive real eigenvalue of $\boldsymbol{L}$ when it exists. It follows that the following range of μ guarantees a stable $\boldsymbol{F}$,

$$0 \;<\; \mu \;< \min\left\{ \frac{1}{\lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B})},\; \frac{1}{\max\left\{\lambda(\boldsymbol{L}) \in \mathbb{R}^+\right\}} \right\}$$

2.10 APPENDIX B: A and B′ of (2.50) are Diagonal

An off-diagonal entry of $\boldsymbol{A}$ has the form

$$A_{jk} = E\left[\frac{2\overline{u}_{i_j}\overline{u}_{i_k}}{\epsilon + \|\overline{\mathbf{u}}_i\|^2}\right] \quad i \neq k$$

Now $\frac{2\overline{u}_{i_j}^T \overline{u}_{i_k}}{\epsilon+\|\overline{\mathbf{u}}_i\|^2}$ is an odd function of u_{i_k} which has an even (Gaussian) pdf and is independent of the other elements of $\overline{\mathbf{u}}_i$. Thus, $E\left[\frac{\overline{u}_{i_k}\overline{u}_{i_k}}{\epsilon+\|\overline{\mathbf{u}}_i\|^2} \mid \overline{u}_{i_j}\right] = 0$ and hence A_{jk} is zero too. So, $\mathbf{A}$ is diagonal. A similar argument can be used to prove that $\mathbf{B}'$ is diagonal. Now the kth

diagonal entry of $\mathbf{B}'$ can be written as

$$\begin{aligned} B'_{kk} &= E\left[\frac{\overline{u}_{i_k}^2 \|\overline{\mathbf{u}}_i\|_{\overline{\Sigma}_i}^2}{\left(\epsilon + \|\overline{\mathbf{u}}_i\|^2\right)^2}\right] \\ &= E\left[\frac{\overline{u}_{i_k}^2}{\left(\epsilon + \|\overline{\mathbf{u}}_i\|^2\right)^2}\overline{\mathbf{u}}_i \odot \overline{\mathbf{u}}_i\right] \operatorname{diag}\left(\overline{\mathbf{\Sigma}}_{i+1}\right) \end{aligned}$$

It follows that

$$\operatorname{diag}\left(\mathbf{B}'\right) = \mathbf{B} \operatorname{diag}\left(\overline{\mathbf{\Sigma}}_{i+1}\right)$$

where

$$\mathbf{B} = E\left[\frac{\left(\overline{\mathbf{u}}_i \odot \overline{\mathbf{u}}_i\right)^T \left(\overline{\mathbf{u}}_i \odot \overline{\mathbf{u}}_i\right)}{\left(\epsilon + \|\overline{\mathbf{u}}_i\|^2\right)^2}\right]$$

Chapter 3

Transient Analysis of Adaptive Filters with Error Nonlinearities

3.1 Introduction

[1]In this chapter, we show how to extend the same energy-based approach employed in Chapter 2 to the transient analysis of adaptive filters that involve error nonlinearities in their update equations (e.g., [43, 100, 60]). This class of algorithms is among the most difficult to analyze, and it is not uncommon to resort to different methods and assumptions with the intent of performing tractable analyses. Before discussing the features of the approach proposed herein and its contributions, we provide, as a motivation, a summary of selected techniques that have been employed earlier in the literature for the study of such algorithms.

a) Linearization (e.g., [31, 98, 41, 86]). In this method of analysis, the error nonlinearity is linearized around an operating point and higher-order terms are discarded. Analyses that are based on this technique fail to accurately describe the adaptive filter performance for large values of the error, e.g., at early stages of adaptation.

b) Restricted classes of nonlinearities (e.g., [63, 16, 17, 35, 23, 91, 22]). Here, the analysis is restricted to particular classes of algorithms such as the sign-LMS algorithm, the least-mean mixed-norm (LMMN) algorithm, the least-mean fourth (LMF) algorithm, and

[1]A major part of this chapter is reproduced, with permission, from T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of adaptive filters with error nonlinearities," *IEEE Transactions on Signal Processing,* vol. 51, No. 3, pp. 653-663, Mar. 2003.

error saturation nonlinearities. By limiting the study to a specific nonlinearity or to a class of nonlinearities, it is possible to avoid linearization and the analysis results become more accurate.

c) Assumptions on the statistics of the errors. While it is common to impose statistical assumptions on the regression and noise sequences, similar conditions can also be imposed on error quantities. For example, in studying the sign-LMS algorithm, it was assumed in [49] that the elements of the weight-error vector are jointly Gaussian. This assumption was shown in [87] to be valid asymptotically. More accurate is the assumption that the residual error is Gaussian [31, 17], or that its conditional value is [63, 16]. By central limit arguments, this assumption is justified for long adaptive filters [31, 17]. More importantly, this assumption is as valid in the early stages as in the final stages of adaptation. For shorter filters, exact expectation analysis can be employed as in [37, 30, 69].

d) A restricted class of inputs. It is common to assume that the input sequence is white and/or has a Gaussian distribution (e.g., [31, 41, 63, 16, 17, 35, 23, 38, 36, 76]).

e) Independence assumption. It is even more common to assume that the successive regressors are independent in what is widely known as the independence assumptions [43], [64]. Despite being unrealistic, the independence assumptions are among the most heavily used assumptions in adaptive filtering analysis.

f) Gaussian noise. Noise is sometimes restricted to be iid Gaussian as in [31], [63], [49], and [12], although Gaussianity is not as common as the previous assumptions. Surprisingly perhaps, the iid assumption on the noise is almost indispensable even for the analysis of the simplest of adaptive algorithms.

3.1.1 The Approach of this Chapter

In this chapter, we develop an approach that applies to arbitrary error nonlinearities irrespective of the input color and statistics. The arguments assume that the adaptive filter is long enough to justify the following approximations:

(i) The residual error $e_a(i)$, to be defined in (3.5) further ahead, can be assumed to be Gaussian.

(ii) The norm of the input regressor can be assumed to be uncorrelated with $f^2[e(i)]$, the square of the error nonlinearity to be defined in (3.1) further ahead.

Both of these assumptions are realistic for longer adaptive filters (see, e.g., the simulation results in Subsection 3.5.1). Fortunately, they are also realistic in all stages of adaptation (including the early stages).

3.1.2 Organization of the Chapter

The outline of this chapter is as follows. We set the stage in the next section by introducing adaptive filters that employ error nonlinearities. The energy relation is used in Section 3.3 to derive a general recursion that describes the mean-square evolution (i.e., learning curve) of an adaptive filter with error nonlinearity. To achieve this result, we rely on the long filter assumptions, which are formally introduced in this section. The independence assumption turns out to be useful in constructing the dynamical relation. In Section 3.4, we show that the excess mean-square error (EMSE) of an adaptive filter with error nonlinearity can be obtained as the fixed point of a nonlinear function. We present our simulations in Section 3.5 and conclude in Section 3.6.

3.2 Adaptive Algorithms with Error Nonlinearity

As described in Chapter 2, an adaptive filter attempts to identify a weight vector $\boldsymbol{w}^o$, of length M, by using a sequence of row regressors $\{\mathbf{u}_i\}$, also of length M, and output samples $\{d(i)\}$ that are related via

$$d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$$

where $v(i)$ accounts for measurement noise and modelling errors. In this chapter, we continue to consider the class of adaptive algorithms described by (2.1) and (2.2) and reproduced here for convenience

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu \mathbf{u}_i^T f[e(i)], \quad i \geq 0 \tag{3.1}$$

$$e(i) \stackrel{\Delta}{=} d(i) - \mathbf{u}_i \boldsymbol{w}_i = \mathbf{u}_i \boldsymbol{w}^o - \mathbf{u}_i \boldsymbol{w}_i + v(i) \tag{3.2}$$

where $\boldsymbol{w}_i$ is the estimate of $\boldsymbol{w}$ at time i, μ is the step size. However, we restrict our attention here to pure error nonlinearities, as one can infer from the adaptation equation. Table 3.1

Table 3.1: *Examples of error nonlinearities $f[e(i)]$*

ALGORITHM	ERROR NONLINEARITIES $f[e(i)]$
LMS	$e(i)$
LMF	$e^3(i)$
LMF family	$e^{2k+1}(i)$
LMMN	$ae(i) + be^3(i)$
Sign error	$\text{sgn}[e(i)]$
Sat. nonlin.	$\int_0^{e(i)} \exp\left(-\frac{z^2}{2\sigma_{\text{sat}}^2}\right) dz$

lists some common adaptive algorithms and their corresponding error nonlinearities.[2]

Given an adaptive filter of the family (3.1), we are interested in studying the time-evolution and the steady-state values of the variances

$$E|e(i)|^2 \qquad \text{and} \quad E\|\tilde{\boldsymbol{w}}_i\|^2 \tag{3.3}$$

where $\tilde{\boldsymbol{w}}_i$ stands for the weight-error vector

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{w}^o - \boldsymbol{w}_i$$

As explained in Subsection 2.2.1, we perform this study by relying on the energy conservation relationship (2.10) reproduced here for convenience

$$\boxed{\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2 + \overline{\mu}_{\Sigma}(i)\left|e_a^{\Sigma}(i)\right|^2 = \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 + \overline{\mu}_{\Sigma}(i)\left|e_p^{\Sigma}(i)\right|^2} \tag{3.4}$$

where the weighted errors $e_a^{\Sigma}(i)$ and $e_p^{\Sigma}(i)$ are the a priori and posteriori errors respectively and are defined by

$$e_a^{\Sigma}(i) \stackrel{\Delta}{=} \mathbf{u}_i\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}_i, \quad e_p^{\Sigma}(i) \stackrel{\Delta}{=} \mathbf{u}_i\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}_{i+1} \tag{3.5}$$

We can show that the estimation errors $e_a^{\Sigma}(i)$, $e_p^{\Sigma}(i)$, and $e(i)$ are related by

$$\boxed{e_p^{\Sigma}(i) = e_a^{\Sigma}(i) - \frac{\mu}{\bar{\mu}_{\Sigma^{(i)}}} f[e(i)]} \tag{3.6}$$

[2]In this table, LMF stands for the least-mean fourth algorithm [98] while LMMN stands for the least-mean mixed-norm algorithm [91], [22].

where

$$\bar{\mu}_{\Sigma}(i) \stackrel{\Delta}{=} \begin{cases} 1/\|\mathbf{u}_i\|_{\Sigma}^2 & \text{if } \|\mathbf{u}_i\|_{\Sigma}^2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

We can also show that

$$e(i) = e_a(i) + v(i) \tag{3.8}$$

3.3 Dynamical Behavior of the Weight-Error Vector

Our first step is to examine how the energy relation (3.4) can be used to characterize the time-evolution of the weighted variance $E\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2$, for any Σ. Thus consider (3.4) and replace the a-posteriori error $e_p^{\Sigma}(i)$ by its equivalent expression (3.6). This yields

$$\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2 = \|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 - 2\mu e_a^{\Sigma}(i) f[e(i)] + \mu^2 \|\mathbf{u}_i\|_{\Sigma}^2 f^2[e(i)]$$

or, upon taking the expectation of both sides,

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right] - 2\mu \overbrace{E\left[e_a^{\Sigma}(i) f[e(i)]\right]}^{①} + \mu^2 \overbrace{E\left[\|\mathbf{u}_i\|_{\Sigma}^2 f^2[e(i)]\right]}^{②} \tag{3.9}$$

Now, two expectations call for evaluation. This is facilitated by the following assumption on the noise sequence:

AN: The noise sequence $v(i)$ is iid and independent of $\mathbf{u}_i$.

3.3.1 Evaluating the Term ①

To evaluate the first expectation,

$$① \;=\; E\left[e_a^{\Sigma}(i) f[e(i)]\right]$$

we shall assume that the adaptive filter is long enough such that the random variables $e_a(i)$ and $e_a^{\Sigma}(i)$ are jointly Gaussian:

AG: For any constant matrix $\boldsymbol{\Sigma}$ and for all i, $e_a(i)$ and $e_a^{\Sigma}(i)$ are jointly Gaussian.

As mentioned in the introduction, this assumption is reasonable for longer filters by central limit arguments (see also the simulation results in Subsection 3.5.1). A similar assumption was adopted in [31, 16, 17] and its usefulness can be understood from the following result and from the subsequent discussion (see, e.g., [16, 17]).

Lemma 1 (Price's result) *Let x and y be jointly Gaussian random variables that are independent from a third random variable z. Then*

$$E\left[xf[y+z]\right] = \frac{E[xy]}{E[y^2]}E\left[yf[y+z]\right]$$

$\diamondsuit$

With Price's theorem at hand, we can use assumption AG together with the standing assumption on the noise AN and (3.8) to write ① as

$$E\left[e_a^{\Sigma}f[e(i)]\right] = E\left[e_a^{\Sigma}f[e_a(i)+v(i)]\right] = E\left[e_a^{\Sigma}(i)e_a(i)\right]\frac{E\left[e_a(i)f[e_a(i)+v(i)]\right]}{E[e_a^2(i)]} \quad (3.10)$$

At first glance, it would appear that we have replaced the expectation $E\left[e_a^{\Sigma}(i)f[e(i)]\right]$ with a similar one, $E[e_a(i)f(e(i))]$. However, this second form is more tractable. Indeed, the expectation $E[e_a(i)f(e(i))]$ depends on $e_a(i)$ through *the second moment* $E[e_a^2(i)]$ *only*.[3] This can be further seen by expanding it as (where we suppress the time index on the right hand-side):

$$E\left[e_a(i)f[e_a(i)+v(i)]\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e_a f[e_a+v]\frac{1}{\sqrt{2\pi E[e_a^2]}}e^{\frac{-e_a^2}{2E[e_a^2]}}p_v(v)de_a dv \quad (3.11)$$

where p_v is the pdf of the additive noise. The contribution of $e_a(i)$ to the result of the integration will depend solely on $E[e_a^2(i)]$. Therefore, the ratio $E\left[e_a(i)f[e(i)]\right]/E[e_a^2(i)]$, which appears in (3.10), is a function of $E[e_a^2(i)]$. This fact motivates the following definition:[4]

$$\boxed{h_{\mathsf{G}}\left[E[e_a^2(i)]\right] \triangleq \frac{E\left[e_a(i)f[e(i)]\right]}{E[e_a^2(i)]}} \quad (3.12)$$

[3]The expectation of any function of a Gaussian random variable will only depend on the variance of this variable and not on any higher-order moments of it.

[4]The Gaussianity assumption AG is the main assumption leading to the defining expression (3.12) for h_{G}, hence the subscript G. The subscript U for h_{U}, which is defined in (3.16) further ahead, is similarly motivated.

Table 3.2: $h_{\mathsf{G}}[\cdot]$ *for the error nonlinearities of Table 3.1* ($\sigma_{e_a}^2 \stackrel{\Delta}{=} E[e_a^2(i)]$)

ALGORITHM	$h_{\mathsf{G}}[\sigma_{e_a}^2]$ ($v(i)$ Gaussian)	$h_{\mathsf{G}}[\sigma_{e_a}^2]$ (general case)
LMS	1	1
LMF	$3(\sigma_{e_a}^2 + \sigma_v^2)$	$3(\sigma_{e_a}^2 + \sigma_v^2)$
LMF family	$\frac{(2k+2)!}{2^{k+1}(k+1)!}(\sigma_{e_a}^2 + \sigma_v^2)^k$	$\sum_{j=0}^{k} \begin{pmatrix} 2k+1 \\ j \end{pmatrix} \sigma_{e_a}^{2j} E\left[v^{2(k-j)}(i)\right]$
LMMN	$a + 3b(\sigma_v^2 + \sigma_{e_a}^2)$	$a + 3b(\sigma_v^2 + \sigma_{e_a}^2)$
Sign error	$\sqrt{\frac{2}{\pi}} \frac{1}{\sqrt{\sigma_{e_a}^2 + \sigma_v^2}}$	$\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_{e_a}} E\left[e^{-\frac{v^2(i)}{2\sigma_{e_a}^2}}\right]$
Sat. nonlin.	$\frac{\sigma_{\mathsf{sat}}}{\sqrt{\sigma_{e_a}^2 + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}}$	$\frac{\sigma_{\mathsf{sat}}}{\sqrt{\sigma_{e_a}^2 + \sigma_{\mathsf{sat}}^2}} E\left[e^{-\frac{v^2(i)}{2(\sigma_{e_a}^2 + \sigma_{\mathsf{sat}}^2)}}\right]$

For future reference, h_{G} is evaluated for the algorithms of Table 3.1 and the results are shown in Table 3.2 (for general noise distribution and for the Gaussian noise case as well). Combining (3.10) and (3.12) yields

$$E\left[e_a^{\Sigma}(i) f(e(i))\right] = E\left[e_a^{\Sigma}(i) e_a(i)\right] h_{\mathsf{G}}\left[E[e_a^2(i)]\right] \tag{3.13}$$

We finally use the polarization property (see (2.12) in Subsection 2.2.2) to write the first expectation in (3.13) as a weighted-norm of $\tilde{\boldsymbol{w}}_i$ yielding

$$\boxed{① \;=\; E\left[e_a^{\Sigma}(i) f(e(i))\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{\sum u_i^T u_i}\right] h_{\mathsf{G}}\left[E[e_a^2(i)]\right]} \tag{3.14}$$

3.3.2 Evaluating the Term ②

We turn our attention now to the second expectation in (3.9), ② $= E\left[\|\mathbf{u}_i\|^2_{\sum} f^2[e(i)]\right]$, which is easier to handle. The long filter assumption is also useful here:

AU: The adaptive filter is long enough such that $\mu\|\mathbf{u}_i\|^2_{\sum}$ and $f^2[e(i)]$ are uncorrelated.

The unweighted version of this assumption was used in [61], [105, 106]. It becomes more realistic as the filter gets longer. The assumption enables us to split the expectation ② as

$$E\left[\|\mathbf{u}_i\|^2_{\sum} f^2[e(i)]\right] = E\left[\|\mathbf{u}_i\|^2_{\sum}\right] E\left[f^2[e(i)]\right] \tag{3.15}$$

Moreover, since $e_a(i)$ is Gaussian and independent of the noise, we can show (as in (3.11)) that $E\left[f^2[e(i)]\right]$ depends on $e_a(i)$ through its *second moment only.* This prompts us to

Table 3.3: $h_{\mathsf{U}}[\cdot]$ *for the error nonlinearities of Table 3.1* ($\sigma_{e_a}^2 \stackrel{\Delta}{=} E[e_a^2(i)]$)

ALGORITHM	$h_{\mathsf{U}}[\sigma_{e_a}^2]$ ($v(i)$ Gaussian)	$h_{\mathsf{U}}[\sigma_{e_a}^2]$ (general case)
LMS	$\sigma_{e_a}^2 + \sigma_v^2$	$\sigma_{e_a}^2 + \sigma_v^2$
LMF	$15(\sigma_{e_a}^2 + \sigma_v^2)^3$	$15\sigma_{e_a}^6 + 45\sigma_{e_a}^4\sigma_v^2 +$ $15\sigma_{e_a}^2 E\left[v^4(i)\right] + E\left[v^6(i)\right]$
LMF family	$\frac{(4k+2)!}{2^{2k+1}(2k+1)!}(\sigma_{e_a}^2 + \sigma_v^2)^{2k+1}$	$\sum_{j=0}^{2k+1}\binom{4k+2}{2j}\frac{(2j)!}{2^j j!}\sigma_{e_a}^{2j}\times$ $E\left[v^{2(2k-j+1)}(i)\right]$
LMMN	$a^2(\sigma_{e_a}^2 + \sigma_v^2) + 6ab(\sigma_{e_a}^2 + \sigma_v^2)^2$ $+15b^2(\sigma_{e_a}^2 + \sigma_v^2)^3$	$15b^2\sigma_{e_a}^6 + (45b^2\sigma_v^2 + 6ab)\sigma_{e_a}^4$ $+(15b^2 E\left[v^4(i)\right] + 12ab\sigma_v^2 + a^2)\sigma_{e_a}^2$ $+E\left[(bv^2(i) + a)^2 v^2(i)\right]$
Sign error	1	1
Sat. nonlin.	$\sigma_{\mathsf{sat}}^2 \sin^{-1}\left(\frac{\sigma_{e_a}^2 + \sigma_v^2}{\sigma_{e_a}^2 + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}\right)$	$2\pi\sigma_{\mathsf{sat}}^2\left(\frac{1}{4} - \frac{1}{\pi}\int_{\pi/4}^{\pi/2}\sqrt{\frac{\sigma_{\mathsf{sat}}^2\sin^2(\theta)}{\sigma_{e_a}^2+\sigma_{\mathsf{sat}}^2\sin^2(\theta)}}\times\right.$ $\left. E[e^{-\frac{v^2(i)}{2(\sigma_{e_a}^2+\sigma_{\mathsf{sat}}^2\sin^2(\theta))}}]\right)$

define

$$h_{\mathsf{U}}\left[E[e_a^2(i)]\right] \stackrel{\Delta}{=} E\left[f^2[e(i)]\right] \tag{3.16}$$

which together with (3.15) yields

$$\boxed{E\left[\|\mathbf{u}_i\|_{\Sigma}^2 f^2[e(i)]\right] = E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] h_{\mathsf{U}}\left[E\left[e_a^2(i)\right]\right]} \tag{3.17}$$

The function h_{U} is evaluated for the algorithms of Table 3.1 and the results are shown in Table 3.3 for general noise and for the Gaussian noise special case (the last entry in the table is derived in the appendix).

3.3.3 Weight-Error Recursion

By substituting (3.14) and (3.17) into (3.9), we obtain

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right] - 2\mu h_{\mathsf{G}}\left[E[e_a^2(i)]\right] E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma u_i^T u_i}^2\right] + \mu^2 E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] h_{\mathsf{U}}\left[E[e_a^2(i)]\right]$$

Upon replacing the mean-square error $E\left[e_a^2(i)\right]$ with the equivalent expression $E\left[\|\tilde{\boldsymbol{w}}_i\|_{u_i^T u_i}^2\right]$, the recursion takes the more homogeneous form shown in the statement below.

Theorem 4 (Weighted-energy relation) *Consider an adaptive filter of the form*

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu \mathbf{u}_i^T f[e(i)], \quad i \geq 0$$

where $e(i) = d(i) - \mathbf{u}_i \boldsymbol{w}_i$ *and* $d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$. *Assume the noise sequence* $v(i)$ *is* iid *and independent of* $\mathbf{u}_i$, *and that the filter is long enough so that* $e_a(i)$ *and* $e_a^{\Sigma}(i)$ *are jointly Gaussian and that* $\mu \|\mathbf{u}_i\|_{\Sigma}^2$ *and* $f^2[e(i)]$ *are uncorrelated. Then the following recursion holds for the weighted weight-error variance,* $E\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2$*:*

$$\begin{aligned} E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right] - 2\mu h_{\mathsf{G}}\left[E\left[\|\tilde{\boldsymbol{w}}_i\|_{u_i^T u_i}^2\right]\right] E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma u_i^T u_i}^2\right] \\ + \mu^2 E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] h_{\mathsf{U}}\left[E\left[\|\tilde{\boldsymbol{w}}_i\|_{u_i^T u_i}^2\right]\right] \end{aligned} \tag{3.18}$$

where the functions $h_{\mathsf{G}}[\cdot]$ *and* $h_{\mathsf{U}}[\cdot]$ *are defined by*

$$h_{\mathsf{U}} = E\left[f^2[e(i)]\right], \quad h_{\mathsf{G}} = E[e_a(i) f[e(i)]] / E[e_a^2(i)]$$

$\diamondsuit$

Remarks

1. What we have achieved so far is to transform recursion (3.9) into (3.18), which depends on various weighted Euclidean norms of the weight-error vector – thanks to assumptions AG and AU.

2. Assumptions AG and AU eventually get translated into some mixing conditions on the signal statistics. In particular, the Gaussian assumption AG on $e_a(i) = \mathbf{u}_i \tilde{\boldsymbol{w}}_i$ requires that the process of individual summands $u_i(l) w_i(l)$ be mixing [18, Theorem 27.4]. Similarly, the AU assumption is justified by the law of large numbers which in turn requires that the input $\mathbf{u}_i$ be mixing [19].

3. The independence assumption on the noise AN is equally essential in developing (3.14), (3.17), and, hence, (3.18). It is a reasonable assumption that allows us to express the expectations in (3.9) in terms of the weight-error energy.

4. Recursion (3.18) as it stands is difficult to propagate in time. The reason is that the recursion is not self-contained as the right-hand side is dependent on $E\|\tilde{\boldsymbol{w}}_i\|_{\Sigma u_i^T u_i}^2$ and $E\|\tilde{\boldsymbol{w}}_i\|_{u_i^T u_i}^2$, in addition to $E\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2$.

5. Note that only a weak form of the independence assumption, namely AU, has been used so far. Contrast this with the standard (stronger)[5] independence assumption:

   AI: The sequence $\mathbf{u}_i$ is zero-mean, independent, and identically distributed, with autocorrelation matrix $\boldsymbol{R} = E\left[\mathbf{u}_i^T \mathbf{u}_i\right]$.

   In this case, recursion (3.18) reduces to the following.

   Corollary 1 (Energy recursion with independence) *Consider the same setting of Theorem 1. If, in addition, the sequence* $\mathbf{u}_i$ *is zero-mean,* iid*, and has covariance matrix* $\boldsymbol{R}$*, then (3.18) becomes*

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right] - 2\mu h_{\mathsf{G}}\left[E[\|\tilde{\boldsymbol{w}}_i\|_R^2]\right] E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma R}^2\right] + \mu^2 E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] h_{\mathsf{U}}\left[E[\|\tilde{\boldsymbol{w}}_i\|_R^2]\right] \tag{3.19}$$

   ♢

3.3.4 Constructing the Learning Curves

The learning curve of the filter refers to the time-evolution of the variance $Ee^2(i)$; its steady-state value is the mean-square error, MSE. Clearly, in view of (3.8), we have that

$$Ee^2(i) = Ee_a^2(i) \; + \; \sigma_v^2$$

so that studying the evolution of $Ee^2(i)$ is equivalent to studying the evolution of $Ee_a^2(i)$; the steady-state value of the latter is called the excess mean-square error, EMSE.

Now under the independence assumption we have

$$Ee_a^2(i) = E|\mathbf{u}_i \tilde{\boldsymbol{w}}_i|^2 \; = \; E\left[\|\tilde{\boldsymbol{w}}_i\|_R^2\right]$$

This suggests that the learning curve can be evaluated by computing $E[\|\tilde{\boldsymbol{w}}_i\|_R^2]$ for each i. This task can be accomplished recursively from (3.19) by essentially choosing $\boldsymbol{\Sigma} = \boldsymbol{R}$, as we now verify.

[5] For example, when the input is of constant modulus, assumption AU is true while AI is not.

The Case of White Regression Data

Consider first the case of white input data for which $\boldsymbol{R} = \sigma_u^2 \boldsymbol{I}$, so that $E[e_a^2(i)] = \sigma_u^2 E\|\tilde{\boldsymbol{w}}_i\|^2$. Restricting the input in this manner is a common practice in the literature (e.g., as in [31],[17], [23], [29], [5]).

Thus setting $\boldsymbol{\Sigma} = \boldsymbol{I}$ in (3.19) we get

$$\begin{aligned} E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|^2\right] &= E\left[\|\tilde{\boldsymbol{w}}_i\|^2\right] - 2\mu\sigma_u^2 h_{\mathsf{G}}\left[\sigma_u^2 E[\|\tilde{\boldsymbol{w}}_i\|^2]\right] E\left[\|\tilde{\boldsymbol{w}}_i\|^2\right] + \\ &\quad \mu^2\sigma_u^2 M h_{\mathsf{U}}\left[\sigma_u^2 E\left[\|\tilde{\boldsymbol{w}}_i\|^2\right]\right] \end{aligned} \tag{3.20}$$

Note that the right-hand side now depends on $E\|\tilde{\boldsymbol{w}}_i\|^2$ only and (3.20) can be propagated in time. We have thus obtained a recursion for the evolution of the variance $E\|\tilde{\boldsymbol{w}}_i\|^2$ for adaptive filters with error nonlinearities and white input regression data.

The Case of Correlated Regression Data

The result (3.19), however, allows us to evaluate the time-evolution of $E\|\tilde{\boldsymbol{w}}_i\|^2$ and $Ee_a^2(i)$ even without the whiteness assumption on the regression data (i.e., for general matrices $\boldsymbol{R}$). The key idea is to take advantage of the free parameter $\boldsymbol{\Sigma}$. Let us in particular write (3.19) for the choices $\boldsymbol{\Sigma} = \boldsymbol{I}, \boldsymbol{R}, \cdots, \boldsymbol{R}^{M-1}$ (the arguments of the functions h_{G} and h_{U} remain the same (i.e., $E\left[\|\tilde{\boldsymbol{w}}_i\|_R^2\right]$) regardless of the choice of $\boldsymbol{\Sigma}$ and are therefore suppressed for convenience of notation):

$$\left\{\begin{array}{lcl} E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_I^2\right] &=& E\left[\|\tilde{\boldsymbol{w}}_i\|_I^2\right] - 2\mu h_{\mathsf{G}} E\left[\|\tilde{\boldsymbol{w}}_i\|_R^2\right] + \mu^2 E\left[\|\mathbf{u}_i\|_I^2\right] h_{\mathsf{U}} \\ E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_R^2\right] &=& E\left[\|\tilde{\boldsymbol{w}}_i\|_R^2\right] - 2\mu h_{\mathsf{G}} E\left[\|\tilde{\boldsymbol{w}}_i\|_{R^2}^2\right] + \mu^2 E\left[\|\mathbf{u}_i\|_R^2\right] h_{\mathsf{U}} \\ & & \vdots \\ E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{R^{M-1}}^2\right] &=& E\left[\|\tilde{\boldsymbol{w}}_i\|_{R^{M-1}}^2\right] - 2\mu h_{\mathsf{G}} E\left[\|\tilde{\boldsymbol{w}}_i\|_{R^M}^2\right] + \mu^2 E\left[\|\mathbf{u}_i\|_{R^{M-1}}^2\right] h_{\mathsf{U}} \end{array}\right. \tag{3.21}$$

The problem now is that the left-hand side of (3.21) is always one variable short of the number of variables on the right-hand side. Fortunately, we do not have to continue in this manner indefinitely since the additional variable $E\left[\|\tilde{\boldsymbol{w}}_i\|_{R^M}^2\right]$ can be expressed in terms of the "lower-order" variables. Using the Cayley-Hamilton theorem, we have

$$\boldsymbol{R}^M = -p_0\boldsymbol{I} - p_1\boldsymbol{R} - \cdots - p_{M-1}\boldsymbol{R}^{M-1}$$

where

$$\begin{aligned} p(x) &\stackrel{\Delta}{=} \det\left(x\boldsymbol{I}-\boldsymbol{R}\right) \\ &= p_0+p_1x+\cdots+p_{M-1}x^{M-1}+x^M \end{aligned}$$

is the characteristic polynomial of $\boldsymbol{R}$. This induces the desired relation

$$\|\tilde{\boldsymbol{w}}_i\|^2_{R^M} = -p_0\|\tilde{\boldsymbol{w}}_i\|^2 - p_1\|\tilde{\boldsymbol{w}}_i\|^2_R - \cdots - p_{M-1}\|\tilde{\boldsymbol{w}}_i\|^2_{R^{M-1}}$$

and enables us to rewrite the last equation in (3.21) as

$$\begin{aligned} E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|^2_{R^{M-1}}\right] &= E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{R^{M-1}}\right] + 2\mu\left(p_0\|\tilde{\boldsymbol{w}}_i\|^2 + p_1\|\tilde{\boldsymbol{w}}_i\|^2_R + \cdots + p_{M-1}\|\tilde{\boldsymbol{w}}_i\|^2_{R^{M-1}}\right)h_{\mathsf{G}} \\ &\quad + \mu^2 E\left[\|\mathbf{u}_i\|^2_{R^{M-1}}\right]h_{\mathsf{U}} \end{aligned}$$

The system (3.21) now becomes truly self-contained and as such can be put into the state-space form shown in the statement below.

Theorem 5 (Transient behavior with independence) *Consider an adaptive filter of the form*

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \mu\mathbf{u}_i^T f[e(i)], \quad i \geq 0$$

where $e(i) = d(i) - \mathbf{u}_i w_i$ *and* $d(i) = \mathbf{u}_i\boldsymbol{w}^o + v(i)$. *Assume that* $\{v(i), \mathbf{u}_i\}$ *are* iid *and mutually independent, that the filter is long enough so that* $e_a(i)$ *and* $e_a^{\Sigma}(i)$ *are jointly Gaussian and that* $\mu\|\mathbf{u}_i\|^2_{\Sigma}$ *and* $f^2[e(i)]$ *are uncorrelated. Then, regardless of the statistics of the regression data, the transient behavior of the filter is characterized by the state-space recursion*

$$\boxed{\mathcal{W}_{i+1} = \mathcal{A}\mathcal{W}_i + \mu^2\mathcal{Y}} \tag{3.22}$$

where the state vector $\mathcal{W}_i$ *and the input vector* $\mathcal{Y}$ *are defined by*

$$\mathcal{W}_i = \begin{bmatrix} E\left[\|\tilde{\boldsymbol{w}}_i\|^2\right] \\ E\left[\|\tilde{\boldsymbol{w}}_i\|^2_R\right] \\ \vdots \\ E\left[\|\tilde{\boldsymbol{w}}_i\|^2_{R^{M-1}}\right] \end{bmatrix}, \qquad \mathcal{Y} = h_{\mathsf{U}} \cdot \begin{bmatrix} E\left[\|\mathbf{u}_i\|^2\right] \\ E\left[\|\mathbf{u}_i\|^2_R\right] \\ \vdots \\ E\left[\|\mathbf{u}_i\|^2_{R^{M-1}}\right] \end{bmatrix}$$

and the coefficient matrix $\mathcal{A}$ is given by

$$\mathcal{A} = \begin{bmatrix} 1 & -2\mu h_{\mathsf{G}} & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2\mu h_{\mathsf{G}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2\mu h_{\mathsf{G}} \\ 2\mu p_0 h_{\mathsf{G}} & 2\mu p_1 h_{\mathsf{G}} & 2\mu p_2 h_{\mathsf{G}} & \cdots & 2\mu p_{M-2} h_{\mathsf{G}} & 1 + 2\mu p_{M-1} h_{\mathsf{G}} \end{bmatrix}$$

in terms of $\{h_G, h_U\}$ and the $\{p_i\}$.

$\diamondsuit$

Remarks:

1. Since $\mathcal{A}$ and $\mathcal{Y}$ depend on $\{h_{\mathsf{U}}, h_{\mathsf{G}}\}$, they are also functions of $E\left[\|\tilde{\boldsymbol{w}}_i\|_R^2\right]$ and hence of the state vector $\mathcal{W}_i$. Thus, the state-space model (3.22) is generally nonlinear, yet time-invariant.

2. Stability and steady-state analysis of the adaptive filter can now be characterized by studying the properties of the state-space model (3.22).

3. The top entry of the state vector $\mathcal{W}_i$ characterizes the evolution of $E\|\tilde{\boldsymbol{w}}_i\|^2$ (mean-square deviation curve), while the second entry of $\mathcal{W}_i$ characterizes the evolution of $Ee_a^2(i)$ (learning curve).

3.4 Steady-State Analysis

Now that the transient behavior of adaptive filters of the class (3.1) has been characterized, we move on to show how the results so far can be used to evaluate the steady-state performance of this same class of filters. Actually, the discussion that follows does not require the independence assumption AI any longer.

We refer again to the averaged energy relation (3.18), which we rewrite using (3.5) as

$$E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right] - 2\mu h_{\mathsf{G}}\left[E[e_a^2(i)]\right] E\left[e_a^{\Sigma}(i) e_a(i)\right] + \mu^2 E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] h_{\mathsf{U}}\left[E[e_a^2(i)]\right] \tag{3.23}$$

Assuming that the weight-error vector reaches a steady-state mean-square value, i.e.,

$$\lim_{i\to\infty} E\left[\|\tilde{\boldsymbol{w}}_{i+1}\|_{\Sigma}^2\right] = \lim_{i\to\infty} E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2\right]$$

the energy relation (3.23) becomes in the limit:

$$\lim_{i\to\infty} h_{\mathsf{G}}\left[E[e_a^2(i)]\right] \lim_{i\to\infty} E\left[e_a^{\Sigma}(i)e_a(i)\right] = \frac{\mu}{2}E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] \lim_{i\to\infty} h_{\mathsf{U}}\left[E[e_a^2(i)]\right]$$

or

$$\lim_{i\to\infty} E[e_a^{\Sigma}(i)e_a(i)] = \frac{\mu}{2}E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] \frac{\lim_{i\to\infty} h_{\mathsf{U}}\left[E[e_a^2(i)]\right]}{\lim_{i\to\infty} h_{\mathsf{G}}[E[e_a^2(i)]]} \tag{3.24}$$

Now, let ζ denote the EMSE, i.e.,

$$\zeta = \lim_{i\to\infty} E[e_a^2(i)] \tag{3.25}$$

which, assuming the filter is mean-square stable, exists and is finite. Then,

$$\lim_{i\to\infty} h_{\mathsf{G}}[E[e_a^2(i)]] = h_{\mathsf{G}}[\zeta] \quad \text{and} \quad \lim_{i\to\infty} h_{\mathsf{U}}[E[e_a^2(i)]] = h_{\mathsf{U}}[\zeta]$$

and, accordingly, (3.24) can be written more compactly as shown below.

Theorem 6 (Steady-state performance) *Consider the same setting of Theorem 1. Then, assuming a mean-square stable filter with* EMSE *denoted by* ζ*, the following equality holds*

$$\boxed{\lim_{i\to\infty} E\left[e_a^{\Sigma}(i)e_a(i)\right] = \frac{\mu}{2}E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right] \frac{h_{\mathsf{U}}[\zeta]}{h_{\mathsf{G}}[\zeta]}} \tag{3.26}$$

♢

The above relation has been derived for general memoryless error nonlinearities. We now show how it can be used to evaluate various steady-state quantities such as the excess mean-square error and the mean-square deviation.

3.4.1 Excess Mean-Square Error

To calculate the excess mean-square error, we employ (3.26) with $\mathbf{\Sigma}$ set to the identity matrix:

$$\lim_{i\to\infty} E[e_a^2(i)] = \frac{\mu}{2}E\left[\|\mathbf{u}_i\|^2\right] \frac{h_{\mathsf{U}}[\zeta]}{h_{\mathsf{G}}[\zeta]} = \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{R}\right) \frac{h_{\mathsf{U}}[\zeta]}{h_{\mathsf{G}}[\zeta]}$$

Or, since $\zeta = \lim_{i\to\infty} E[e_a^2(i)]$, we arrive at the following statement.

Corollary 2 (EMSE) *Consider the same setting of Theorem 1. Then the* EMSE *is a positive solution of the equation*

$$\boxed{\zeta = \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{R}\right)\frac{h_{\mathsf{U}}[\zeta]}{h_{\mathsf{G}}[\zeta]}} \tag{3.27}$$

i.e., the EMSE *is a fixed point of the function* $(\mu/2)\mathrm{Tr}\left(\boldsymbol{R}\right)h_{\mathsf{U}}[\zeta]/h_{\mathsf{G}}[\zeta]$.

♢

In the following we show how (3.27) specializes for some nonlinearities.

The LMS Algorithm

In the LMS case, (3.27) reads

$$\zeta = \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{R}\right)\left(\zeta + \sigma_v^2\right)$$

or, upon solving for ζ, we obtain the well-known result [38]:

$$\zeta = \frac{\mu\sigma_v^2\mathrm{Tr}\left(\boldsymbol{R}\right)}{2 - \mu\mathrm{Tr}\left(\boldsymbol{R}\right)}$$

The Sign Algorithm

We start from (3.27) again. With the aid of Tables 3.2 and 3.3, we see that

$$\zeta = \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{R}\right)\frac{h_{\mathsf{U}}[\zeta]}{h_{\mathsf{G}}[\zeta]} = \mu\sqrt{\frac{\pi}{8}}\mathrm{Tr}\left(\boldsymbol{R}\right)\frac{\sqrt{\zeta}}{E\left[e^{\frac{-v^2(i)}{2\zeta}}\right]} \tag{3.28}$$

It is worth noting in the sign algorithm case, that assumption AU is not needed. In other words, we only need the Gaussian assumption AG to establish (3.28). This was the same conclusion arrived at in [105] but the study there was limited to the Gaussian noise case. Further progress is pending the evaluation of $E\left[e^{\frac{-v^2(i)}{2\zeta}}\right]$, which calls for specifying the noise statistics. Our findings are summarized in the Table 3.4. In particular, we arrive at the same EMSE expressions of [23] derived there under the independence assumption for iid input. In the second line of Table 3.4, the noise is assumed to be equal to $\pm\sigma_v$ with probability 1/2, while in the third line the noise is assumed to be uniformly distributed inside the interval $(-\sqrt{3}\sigma_v, \sqrt{3}\sigma_v)$. The erf function is defined by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$$

Table 3.4: EMSE *for the sign algorithm for various noise statistics*

Noise	EMSE
Gaussian [63, 105]	$\zeta = \alpha \dfrac{\alpha + \sqrt{\alpha^2 + 4\sigma_v^2}}{2}, \;\; \alpha = \mu\sqrt{\dfrac{\pi}{8}}\mathrm{Tr}\,(\boldsymbol{R})$
Binary [23]	$\zeta = \alpha^2 e^{\frac{\sigma_v^2}{\zeta}}, \;\; \alpha = \mu\sqrt{\dfrac{\pi}{8}}\mathrm{Tr}\,(\boldsymbol{R})$
Uniform [23]	$\zeta = \dfrac{\mu}{2}\dfrac{\sqrt{3\sigma_v^2}}{\mathrm{erf}\left(\sqrt{\dfrac{3\sigma_v^2}{2\zeta}}\right)}\mathrm{Tr}\,(\boldsymbol{R})$

Error-Saturation Algorithm

Consider the saturation nonlinearity in Table 3.1. The associated expectations h_{G} and h_{U} are relatively easy to establish in the Gaussian noise case (see Tables 3.2 and 3.3):

$$h_{\mathsf{G}}[\zeta] = \frac{\sigma_{\mathsf{sat}}}{\sqrt{\zeta + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}}, \qquad h_{\mathsf{U}}[\zeta] = \sigma_{\mathsf{sat}}^2 \sin^{-1}\left(\frac{\zeta + \sigma_v^2}{\zeta + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}\right)$$

which upon substitution in (3.27) yield the following relation for the EMSE

$$\frac{\zeta}{\sqrt{\zeta + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}} = \sigma_{\mathsf{sat}}\frac{\mu}{2}\mathrm{Tr}\,(\boldsymbol{R}) \sin^{-1}\left(\frac{\zeta + \sigma_v^2}{\zeta + \sigma_v^2 + \sigma_{\mathsf{sat}}^2}\right)$$

This is the same result arrived at in [17] under the independence assumption for iid input.

In the general noise case, we have

$$h_{\mathsf{G}}[\zeta] = \frac{\sigma_{\mathsf{sat}}}{\sqrt{\zeta + \sigma_{\mathsf{sat}}^2}} E\left[e^{-\frac{v^2(i)}{2(\zeta + \sigma_{\mathsf{sat}}^2)}}\right] \tag{3.29}$$

which encompasses the binary noise case considered in [17] as a special case. Evaluating h_{U} is more difficult; this was attempted in [17] and the argument led to a complicated expression involving double integrals and infinite limits. We arrive in the Appendix at the expression

$$h_{\mathsf{U}}[\zeta] = 2\pi\sigma_{\mathsf{sat}}^2\left(\frac{1}{4} - \frac{1}{\pi}\int_{\pi/4}^{\pi/2}\sqrt{\frac{\sigma_{\mathsf{sat}}^2\sin^2(\theta)}{\sigma_{e_a}^2 + \sigma_{\mathsf{sat}}^2\sin^2(\theta)}}E[e^{-\frac{v^2(i)}{2(\sigma_{e_a}^2 + \sigma_{\mathsf{sat}}^2\sin^2(\theta))}}]\right) \tag{3.30}$$

by relying on a convenient expression for the error function introduced [89]. Upon substituting (3.29) and (3.30) into (3.27), we obtain

$$\frac{\zeta}{\sqrt{\zeta+\sigma_{\mathsf{sat}}^2}}E\left[e^{-\frac{v^2(i)}{2(\zeta+\sigma_{\mathsf{sat}}^2)}}\right]=\mu\pi\sigma_{\mathsf{sat}}\mathrm{Tr}\left(\boldsymbol{R}\right)\left(\frac{1}{4}-\frac{1}{\pi}\int_{\pi/4}^{\pi/2}\sqrt{\frac{\sigma_{\mathsf{sat}}^2\sin^2(\theta)}{\sigma_{e_a}^2+\sigma_{\mathsf{sat}}^2\sin^2(\theta)}}E[e^{-\frac{v^2(i)}{2(\sigma_{e_a}^2+\sigma_{\mathsf{sat}}^2\sin^2(\theta))}}]\right)$$

which can be numerically solved for ζ, the EMSE.

The LMF Algorithm

For the LMF algorithm, and with the aid of Tables 3.2 and 3.3, (3.27) takes the form

$$\zeta=\frac{\mu}{6}\frac{15\zeta^3+45\sigma_v^2\zeta^2+15m_{v,4}\zeta+m_{v,6}}{\zeta+\sigma_v^2}\mathrm{Tr}\left(\boldsymbol{R}\right) \tag{3.31}$$

where $m_{v,4}$ and $m_{v,6}$ denote the fourth and sixth moments of $v(i)$. Finding the EMSE is thus equivalent to finding the roots of a 3rd-order equation, which can be done numerically. We can avoid this in the Gaussian case and obtain a *closed* formula for the EMSE.

Gaussian Noise. In the Gaussian noise case, (3.31) simplifies to

$$\zeta=\frac{5\mu}{2}\frac{\left(\zeta+\sigma_v^2\right)^3}{\zeta+\sigma_v^2}\mathrm{Tr}\left(\boldsymbol{R}\right)=\frac{\alpha}{2}\left(\zeta+\sigma_v^2\right)^2$$

where $\alpha=5\mu\mathrm{Tr}\left(\boldsymbol{R}\right)$. This is a quadratic equation in ζ with *two* positive roots

$$\zeta=\frac{(1-\alpha\sigma_v^2)\pm\sqrt{1-2\alpha\sigma_v^2}}{\alpha} \tag{3.32}$$

Simulations show that only the smaller root is meaningful.

It appears that calculating the steady-state error for super nonlinearities (e.g., the LMF algorithm, the LMF family, and the LMMN algorithm) has always involved some form of linearization (e.g., [98], [91], [105], [29], [5], [104]). The LMF derivation above demonstrates how the EMSE can be obtained for such algorithms without having to employ linearization arguments.

3.4.2 Mean-Square Deviation

The mean-square deviation (MSD), defined as

$$\mathsf{MSD} = \lim_{i\to\infty} E\|\tilde{\boldsymbol{w}}_i\|^2$$

can be related to the EMSE by invoking the independence assumption in the limit. More specifically, by combining (3.26) and (3.27), we obtain

$$\lim_{i\to\infty} E\left[e_a^{\Sigma}(i)e_a(i)\right] = \zeta \cdot \frac{E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right]}{E\left[\|\mathbf{u}_i\|^2\right]}$$

Assuming AI holds in the limit we have

$$\lim_{i\to\infty} E\left[e_a^{\Sigma}(i)e_a(i)\right] \;=\; \lim_{i\to\infty} E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma R}^2\right]$$

so that

$$\lim_{i\to\infty} E\left[\|\tilde{\boldsymbol{w}}_i\|_{\Sigma R}^2\right] = \zeta \cdot \frac{E\left[\|\mathbf{u}_i\|_{\Sigma}^2\right]}{E\left[\|\mathbf{u}_i\|^2\right]} \tag{3.33}$$

Since we are interested in $E\|\tilde{\boldsymbol{w}}_i\|^2$, we choose $\boldsymbol{\Sigma}$ in (3.33) as $\boldsymbol{R}^{-1}$, which leads us to the following conclusion

Corollary 3 (MSD) *Consider the same setting of Theorem 1 and assume, in addition, that the sequence* $\mathbf{u}_i$ *is zero-mean* iid. *Then the* MSD *is given by*

$$\boxed{\mathsf{MSD} = \frac{M\zeta}{\mathrm{Tr}(\boldsymbol{R})}}$$

where ζ *denotes the filter* EMSE.

♢

Other steady-state measures can be similarly evaluated. Thus, for any symmetric matrix $\mathbf{A}$, we have

$$\boxed{\lim_{i\to\infty} E\left[\|\tilde{\boldsymbol{w}}_i\|_A^2\right] = \frac{E\left[\|\mathbf{u}_i\|_{AR^{-1}}^2\right]}{E\left[\|\mathbf{u}_i\|^2\right]}\mathsf{EMSE}}$$

3.5 Simulations

Throughout this section, the system to be identified is an FIR channel of length 16. The input $u(i)$ is generated by passing an iid (uniform or Gaussian) process $x(i)$ through a first-order model,

$$u(i) = au(i-1) + x(i) \tag{3.34}$$

By varying the value of a, we obtain processes $u(i)$ of different colors. Here we set $a = 0.3$. The output is contaminated by an iid (uniform or Gaussian) additive noise at an SNR level of 10 dB.

3.5.1 Testing the Gaussianity of $e_a(i)$

We start by running a simulation to test the Gaussian assumption AG on $e_a(i)$ for the sign algorithm. We choose the sign algorithm because it was argued in [62] that $e_a(i)$ can never be Gaussian under the independence assumption. The signals involved are chosen to be non-Gaussian. Thus, the input is generated by (3.34) and the processes x and v are both taken to be iid uniform.

The Gaussian hypothesis is tested by running the adaptive algorithm 1000 times and plotting the histogram of $e_a(i)$ at the equi-spaced instants $i = 0, 200, \ldots, 1000$. The histograms, depicted in Figure 3.1, suggest that the Gaussian assumption on $e_a(i)$ is a reasonable approximation. The only exception is the histogram for $e_a(0)$, which is almost uniformly distributed (as it should be since $e_a(0)$ is generated by one data point for which the central limit theorem does not apply).

3.5.2 Learning Curves

Next, we study the match between the theoretical (Theorem 2) and simulated learning curves. We test the match for the sign and LMF algorithms. In both cases, the input is assumed to be a Gaussian correlated process with $a = 0.3$. As depicted in Figures 3.2 and 3.3, the experimental and theoretical learning curves agree very well. This agreement occurs despite the fact that large values of the step-size are used.

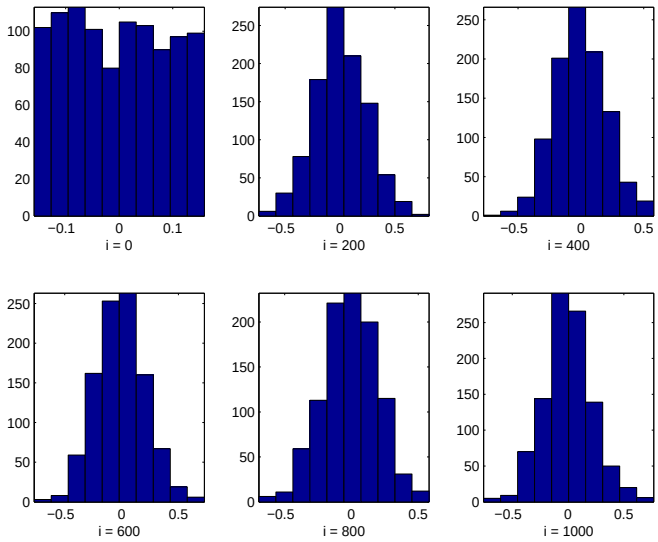

Figure 3.1: Histogram of $e_a(i)$ for the sign algorithm at different time instants (uniform noise, uniform input with $a = 0.3$, $\mu = 0.01$, $\mathsf{SNR} = 10\mathsf{dB}$)

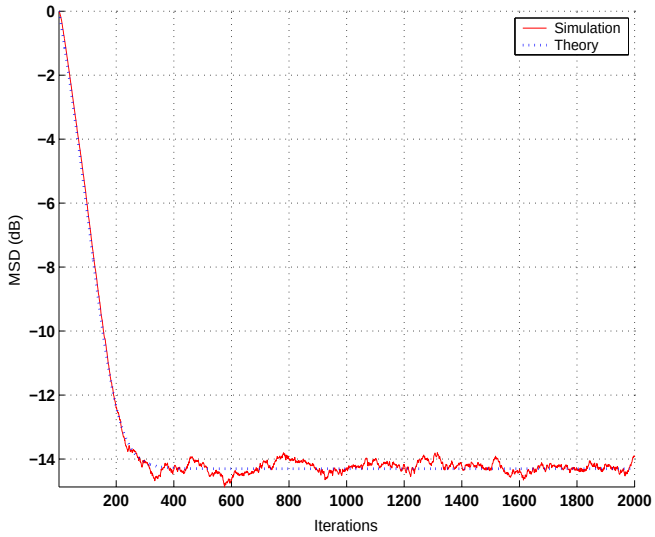

Figure 3.2: Theoretical and simulated learning curves for the sign algorithm (Gaussian noise, Gaussian input with $a = .1$, $\mu = .01$, $\mathsf{SNR} = 10\mathsf{dB}$)

3.5.3 Steady-State Behavior

Here, we simulate the steady-state behavior of the sign and LMF algorithms and compare the results to theory. We test the sign algorithm for correlated uniform input (with $a = 0.3$)

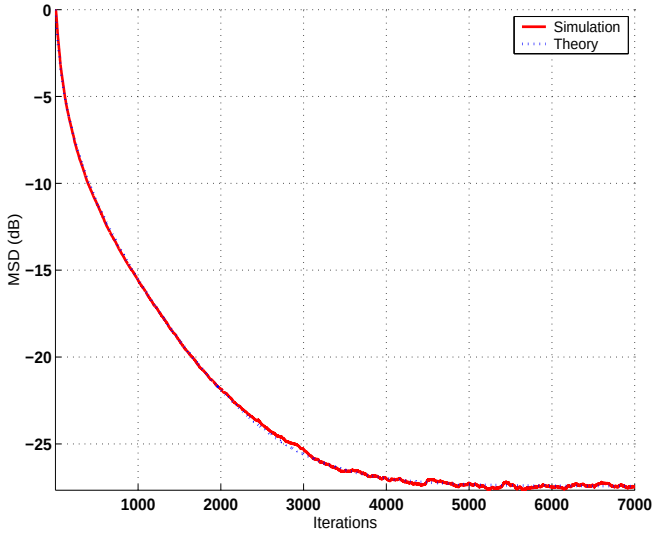

Figure 3.3: Theoretical and simulated learning curves for the LMF algorithm (Gaussian noise, Gaussian input with $a = 0.1$, $\mu = .0044$, SNR = 10dB)

and uniform noise. Figure 3.4 shows an excellent match between the EMSE generated by simulation and that predicted by theory (see Table 3.4).

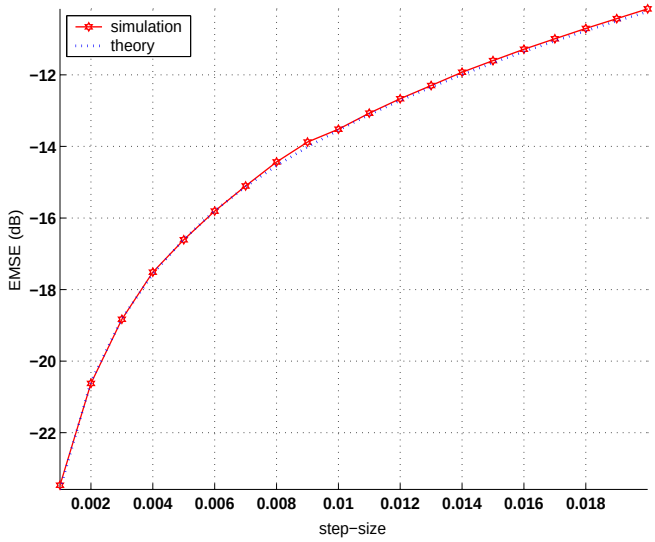

Figure 3.4: Theoretical and simulated EMSE vs. μ for the sign algorithm (uniform noise, Gaussian input with $a = 0.3$, SNR = 10dB)

The LMF is tested for correlated Gaussian input (with $a = 0.3$) and Gaussian noise. Figure 3.5 demonstrates the excellent match between simulation and theoretical values (predicted by (3.32)). In this figure, we also plot the value of the steady-state error as predicted by [105] which eventually employs some sort of linearization. The predictions of (3.32) are more accurate.

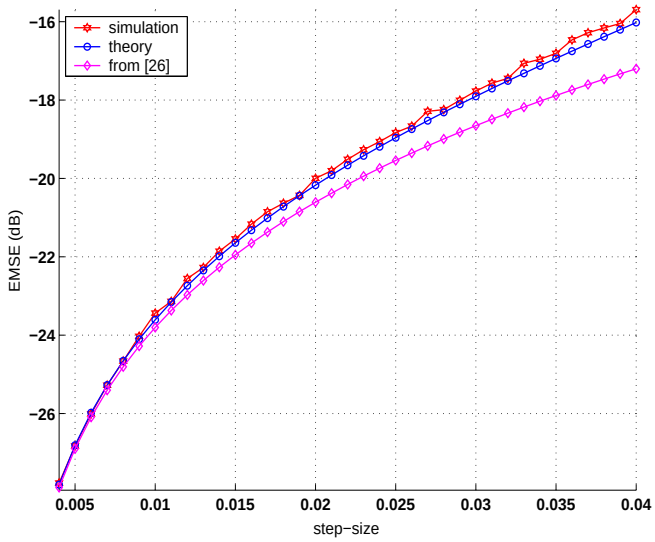

Figure 3.5: Theoretical and simulated EMSE vs. μ for the LMF algorithm (Gaussian noise, Gaussian input with $a = 0.3$, SNR $= 10$dB)

3.6 Concluding Remarks

In this chapter we employed energy-conservation arguments to study the transient performance of adaptive filters with error nonlinearities. The arguments of this chapter and the previous one demonstrate the convenience of working with the energy relation. In developing the energy relation, we basically push the algebraic operations to the limit before we undertake any averaging operation. We do so because our ability to maneuver algebraically under the expectation operator is usually limited. As a result, we are able to do away with assumptions that are otherwise necessary for algebraic operations only.

The main contributions of this chapter are Theorems 1, 2, and 3; the first relates to the energy conservation result, the second relates to the learning curve behavior, and the third relates to a nonlinear equation for EMSE calculation.

3.7 APPENDIX A: Evaluating h_{U} for the Error Saturation Nonlinearity (3.30)

To evaluate the expectation

$$h_{\mathsf{U}}\left[E\left[e_a^2(i)\right]\middle|\, v(i)\right] = E\left[f[e^2(i)]\middle|\, v(i)\right]$$

for the error saturation nonlinearity $f[e(i)] = \int_0^{e(i)} e^{-\frac{z^2}{2\sigma_{\mathsf{sat}}^2}} dz$, we rely on the equivalent representation

$$f[e(i)] = \sqrt{2\pi\sigma_{\mathsf{sat}}^2}\left(-\frac{1}{\pi}\int_0^{\frac{\pi}{2}} e^{-\frac{e^2(i)}{2\sigma_{\mathsf{sat}}^2\sin^2(\theta)}} d\theta\right)\mathrm{sign}(e(i)) \tag{3.35}$$

Powers of f are obtained by changing the integration limits in (3.35) (in addition to other minor changes [89]). Thus,

$$f^2[e(i)] = 2\pi\sigma_{\mathsf{sat}}^2\left(\frac{1}{4} - \frac{1}{\pi}\int_{\frac{\pi}{4}}^{\frac{\pi}{2}} e^{-\frac{e^2(i)}{2\sigma_{\mathsf{sat}}^2\sin^2(\theta)}} d\theta\right) \tag{3.36}$$

Thanks to (3.36), in evaluating $Ef^2[e(i)]$ given $v(i)$, the expectation operator can move inside the integral and operate on its integrand and we can show that

$$E\left[e^{-\frac{e^2(i)}{2\sigma_{\mathsf{sat}}^2\sin^2(\theta)}}\middle|\, v(i)\right] = \sqrt{\frac{\sigma_{\mathsf{sat}}^2\sin^2(\theta)}{\sigma_{e_a}^2 + \sigma_{\mathsf{sat}}^2\sin^2(\theta)}}\, e^{-\frac{v^2(i)}{2(\sigma_{e_a}^2+\sigma_{\mathsf{sat}}^2\sin^2(\theta))}} \tag{3.37}$$

where $\sigma_{e_a}^2 = E\left[e_a^2(i)\right]$. This yields the desired result.

Chapter 4

An EM-Based OFDM Receiver for Time-Variant Channels

4.1 Introduction

[1]Having performed mean-square analysis of well-known adaptive algorithms for channel estimation, we move our attention here into designing our own algorithm for channel estimation (and equalization). In a communication system, the sole purpose of channel estimation is to recover the transmitted data. As such, it is best to consider the two problems (of channel and data recovery) jointly. In other words, to come up with an optimum algorithm for channel estimation, one needs to consider the full problem of receiver design, which is the approach that this chapter adopts.

Moreover, we confine our attention to receiver design for orthogonal frequency division multiplexing (OFDM) although the basic structure of the receiver remains valid in the general case. We focus on OFDM systems because data recovery in OFDM is so simple that it makes it easier to understand the fundamentals of channel estimation (paving the way to eventually design a receiver for the more general case). Moreover, OFDM transmission is so widespread in communications applications that restricting the focus to it constitutes no limitation in practice.

[1]A major part of this chapter is reproduced, with permission, from T. Y. Al-Naffouri and A. Bahai "An EM-based OFDM receiver for time-variant channels," *submitted to IEEE Transactions on Signal Processing.*

Table 4.1: *Common channel and data constraints used for channel estimation*

CONSTRAINTS	ASSUMPTIONS	REFERENCE
Data Constraints	**Finite alphabet constraint**	[88], [7]
	Code	[9], [80]
	Transmit precoding (e.g., cyclic prefix, silent gaurad band)	[20], [44], [40], [68], [52], [99], [9]
	Pilots	[21], [56], [71],[95],[72],[53]
Channel Constraints	**Finite delay spread**	[20], [9], [71]
	Sparsity: Channel has a few active taps	[102], [48], [97]
	Frequency correlation: Taps are Gaussian distributed	[54],[32], [9], [79]
	Time correlation	[94], [50], [45], [1], [58]
	Uncertainty information	[81], [54]

4.1.1 What is OFDM?

Orthogonal frequency division multiplexing (OFDM) is an effective technique for high bit rate transmission. It has found widespread applications and is already part of many standards including digital audio and video broadcasting (DAB and DVB) in Europe and high speed transmission over digital subscriber line (DSL) in the United States. It has been proposed for local area mobile wireless broadband standards including IEEE 802.11a and HIPERLAN/2 [107] and for broadband wireless metropolitan access networks (WiMax).

OFDM avoids intersymbol interference by inserting a guard band (cyclic prefix) into the transmitted symbol. This effectively divides the channel into many narrowband channels over which parallel streams of data are transmitted. Frequency selectivity can now be mitigated using one tap equalizers. However, the receiver still needs accurate channel state information. Alternatively, the receiver can avoid the need for this information by employing differential modulation at the cost of 3 to 4 dB degradation in SNR. Otherwise, the receiver should jointly recover both the channel information and its input. For rapidly time-variant channels, the receiver faces the additional challenge of performing the recovery within the same symbol.

In performing these two operations, the receiver takes advantage of the rich structure of the underlying communication problem to enhance the performance of the receiver and to perform the recovery with zero latency (i.e. within one OFDM symbol). This structure can be either traced back to some inherent constraints on the data or on the channel itself, as explained in Chapter 1 and summarized in Table 1.1, which is reproduced here for convenience.

However, since it is too computationally complex to recover the channel and the data jointly, one can instead perform iterative recovery. Thus, the initial channel estimate is used to detect the data. The detected data is used to enhance the channel estimate which in turn is used to improve on the data estimate.

This intuitive idea is the basis of joint channel estimation and data detection proposed in [50], [45], [57], [39]. Other works, like [2], [58], [24], [55], [101], and [7], arrived at iterative techniques more rigorously by employing the expectation-maximization (EM) algorithm, which we explain in the following.

The EM Algorithm

The EM algorithm is used in estimating a desired parameter when some of the data required for the estimation is unobserved. The algorithm first performs an initial estimate of the unobserved data and uses the information to compute the maximum-likelihood (ML) estimate of the desired parameter. This is the maximization or M-step. The algorithm then uses the parameter estimate to update (compute the conditional expectation) of the unobserved data. This is the expectation or E-step. The process alternates between the M- and E-steps till a convergence criterion is satisfied [26] [66] .

As mentioned above, the EM-algorithm has been applied to receiver design in [58], [24], [55], [101]. These works consider the channel as the unobserved data and the transmitted signal as the desired parameter. The M-step is a maximum likelihood hard decision of the transmitted signal based on the previously calculated channel estimate and the E-step is based on an MMSE estimate of the channel.

4.1.2 The Approach of this Chapter

In contrast to the prior approach, we here take a channel estimation centric viewpoint and reverse the roles of the channel and the transmitted signal in employing the EM algorithm (as done in [2] and [7]). In addition, and in further contrast to other approaches, we make a collective use of the constraints induced by the data and channel that underly the communication problem. Specifically, we are able to make use of all the constraints in Table 4.1.

This chapter is organized as follows. After introducing our notation, we perform a careful study of the elements of OFDM transmission. In particular, we show how the OFDM channel can be decomposed into two channels, linear (represented in the time domain), and circular

(represented in the frequency domain). In this section, we also introduce the time-variant channel model. In Section 4.3, we use the expectation maximization algorithm for joint channel estimation and data detection. We show how the algorithm becomes progressively more sophisticated as more data and channel constraints are incorporated, with each new version of the algorithm subsuming the previous version as a special case, culminating in an EM-based Kalman filter. In Section 4.4, we extend the EM algorithm to the case where the data can be processed non-causally, i.e. when the receiver can wait for all OFDM symbols to arrive before performing the processing. This produces the most sophisticated version of the algorithm, an EM-based forward-backward Kalman filter. We present our simulations in Section 4.5 and our conclusions in Section 4.6.

4.1.3 Notation

We denote scalars with small-case letters, vectors with small-case boldface letters, and matrices with uppercase boldface letters. Calligraphic notation (e.g., $\boldsymbol{\mathcal{X}}$) is reserved for vectors in the frequency domain. The individual entries of a variable like $\boldsymbol{h}$ are denoted by $h(l)$. A hat over a variable indicates an estimate of the variable (e.g., $\hat{\boldsymbol{h}}$ is an estimate of $\boldsymbol{h}$). When any of these variables becomes a function of time, the time index i appears as a subscript (e.g., we write x_i, $h_i(l)$, $\boldsymbol{h}_i$, and $\boldsymbol{\mathcal{X}}_i$). When the various indexed vectors are concatenated, they induce a sequence of scalars (for example, the sequence of vectors $\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots$ induce the scalar sequence $\{h_k\}$).

Now consider a length-N vector $\boldsymbol{x}_i$. We deal with three derivatives associated with this vector. The first two are obtained by partitioning $\boldsymbol{x}_i$ into a lower (trailing) part $\underline{\boldsymbol{x}}_i$ (usually known as the cyclic prefix) and an upper (usually longer) vector $\tilde{\boldsymbol{x}}_i$. The third derivative, $\overline{\boldsymbol{x}}_i$, is created by concatenating $\boldsymbol{x}_i$ with a copy of its prefix $\underline{\boldsymbol{x}}_i$. The relationship among $\boldsymbol{x}_i$ and its derivatives is summarized by

$$\overline{\boldsymbol{x}}_i = \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \boldsymbol{x}_i \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix} \tag{4.1}$$

We can extend this convention to matrices and partition a matrix $\boldsymbol{Q}$ according to

$$\boldsymbol{Q} = \begin{bmatrix} \tilde{\boldsymbol{Q}}_{N-P} \\ \underline{\boldsymbol{Q}}_P \end{bmatrix} \tag{4.2}$$

The subscripts $N-P$ and P represent the number of rows in the partition and can be omitted when they are understood. When the submatrix is constructed from rows in $\boldsymbol{Q}$ that are not consecutive, we distinguish the submatrix by a subscript I corresponding to the rows index set, and we write $\boldsymbol{Q}_I$. Thus, with $\tilde{I} = \{1, \cdots, N-P\}$ and $\underline{I} = \{N-P+1, \cdots, N\}$, we can rewrite (4.2) in the equivalent form

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{\tilde{I}} \\ \boldsymbol{Q}_{\underline{I}} \end{bmatrix} \tag{4.3}$$

We could also introduce similar notation to denote a submatrix constructed by choosing a few columns (belonging to the index set I_{c}) from the mother matrix $\boldsymbol{Q}$. We avoid the need for another piece of notation and arrive at the same effect by forming the product $\boldsymbol{Q}\boldsymbol{I}^*_{I_{\mathsf{c}}}$– a product between $\boldsymbol{Q}$ and the submatrix $\boldsymbol{I}^*_{I_{\mathsf{c}}}$ of the identity matrix. This allows us to express column selection in terms of row selection.

As we will see, the notation adopted herein turns out to be very natural and will make it easy to write down various relationships almost by inspection.

4.2 Essential Elements of OFDM Transmission

Consider a sequence $\{\mathcal{X}_k\}$ that we wish to transmit. Data is collected and transmitted in symbols $\boldsymbol{\mathcal{X}}_i$ of length N. In an OFDM system, the symbol vector $\boldsymbol{\mathcal{X}}_i$ undergoes an IDFT operation to produce the transform vector $\boldsymbol{x}_i$. The two vectors are thus related by the unitary transformation

$$\boldsymbol{x}_i = \frac{1}{\sqrt{N}} \boldsymbol{Q}^* \boldsymbol{\mathcal{X}}_i \tag{4.4}$$

where $\boldsymbol{Q}$ is the $N \times N$ DFT matrix

$$\boldsymbol{Q} = \left[e^{-j\frac{2\pi}{N}(l-1)(m-1)} \right]$$

When juxtaposed, these symbols produce the sequence $\{x_k\}$. [2] If this sequence is transmitted through a channel $\underline{\boldsymbol{h}}_i$, which we take as FIR of maximum length $P+1$, it will be subject

[2] The time indices in the symbol sequence $\boldsymbol{x}_i$ and the underlying scalar sequence $\{x_k\}$ are dummy variables. Nevertheless, we chose to index the two sequences differently to avoid confusion as the two indices signify different sampling rates.

to intersymbol interference (ISI). To get around this, a guard band is inserted between any consecutive symbols, $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$. In particular, to each symbol, we append a cyclic prefix of length P as done in (4.1). Thus, instead of transmitting $\boldsymbol{x}_i$, we transmit the length $N+P$ supersymbol $\overline{\boldsymbol{x}}_i$ defined in (4.1). This eliminates intersymbol interference and leaves us with intrasymbol interference only (i.e. interference within the symbol $\boldsymbol{x}_i$).

The concatenation of the super-symbols $\overline{\boldsymbol{x}}_i$ induces the underlying scalar sequence $\{\overline{x}_k\}$. When it passes through the channel $\underline{\boldsymbol{h}}_i$, this sequence generates the sequence $\{\overline{y}_k\}$ at the channel output. Motivated by the symbol structure of the input, it is convenient to deal with the output in the form of supersymbols of length $M = N + P$, and further split each into a length-N symbol $\boldsymbol{y}_i$ and a prefix associated with it $\underline{\boldsymbol{y}}_i$, i.e.

$$\overline{\boldsymbol{y}}_i = \begin{bmatrix} \overline{\boldsymbol{y}}_i \\ \boldsymbol{y}_i \end{bmatrix} \tag{4.5}$$

This is a natural way to partition the output; the prefix $\underline{\boldsymbol{y}}_i$ actually absorbs all intersymbol interference between the adjacent symbols $\overline{\boldsymbol{x}}_{i-1}$ and $\overline{\boldsymbol{x}}_i$ while the remaining part, $\boldsymbol{y}_i$, depends on the ith input symbol $\boldsymbol{x}_i$ only, hence exhibiting intrasymbol interference only. These facts and more can be seen from the input/output relationship

$$\begin{bmatrix} \boldsymbol{y}_{i-1} \\ \underline{\boldsymbol{y}}_i \\ \boldsymbol{y}_i \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{H}}_i & \boldsymbol{O}_{N\times P} & \boldsymbol{O}_{N\times N} \\ \boldsymbol{O}_{P\times N} \quad \underline{\boldsymbol{H}}_{\mathsf{U}i} & \underline{\boldsymbol{H}}_{\mathsf{L}i} & \boldsymbol{O}_{P\times N} \\ \boldsymbol{O}_{N\times N} \quad \boldsymbol{O}_{N\times P} & \overline{\boldsymbol{H}}_i \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{x}}_{i-1} \\ \tilde{\boldsymbol{x}}_{i-1} \\ \underline{\boldsymbol{x}}_{i-1} \\ \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_{i-1} \\ \underline{\boldsymbol{n}}_i \\ \boldsymbol{n}_i \end{bmatrix} \tag{4.6}$$

where $\boldsymbol{n}_i$ is the output noise which we take to be white Gaussian. The matrices $\overline{\boldsymbol{H}}_i$, $\underline{\boldsymbol{H}}_{\mathsf{L}i}$, and $\underline{\boldsymbol{H}}_{\mathsf{U}i}$ are convolution (Toeplitz) matrices of proper sizes created from the vector $\underline{\boldsymbol{h}}_i$. The channel matrix in (4.6) could have been written as a single Toeplitz matrix. We chose instead to split it along the boarder lines defined by the input and output symbols and their prefixes. This partition coupled with the input redundancy allows us to decompose the OFDM channel into two distinct channels or constituent convolution operations. In what follows, we shall describe each of these channels separately.

4.2.1 Circular Convolution (Channel)

Starting from (4.6), we can parse the following input/output relationship

$$\boldsymbol{y}_i = \overline{\boldsymbol{H}}_i \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix} = \overline{\boldsymbol{H}}_i \, \overline{\boldsymbol{x}}_i + \boldsymbol{n}_i \tag{4.7}$$

Moreover, the existence of a cyclic prefix in $\overline{\boldsymbol{x}}_i$ allows us to rewrite (4.7) as

$$\boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{x}_i + \boldsymbol{n}_i \tag{4.8}$$

where $\boldsymbol{H}_i$ is a size-N circulant matrix

$$\boldsymbol{H}_i = \begin{bmatrix} \underline{h}_i(0) & 0 & \cdots & 0 & \underline{h}_i(P) & \cdots & \underline{h}_i(1) \\ \underline{h}_i(1) & \underline{h}_i(0) & \cdots & 0 & 0 & \cdots & \underline{h}_i(2) \\ \vdots & \vdots & \ddots & \vdots & \cdots & \ddots & \vdots \\ \underline{h}_i(P) & \underline{h}_i(P-1) & \cdots & \underline{h}_i(0) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \cdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \underline{h}_i(P) & \underline{h}_i(P-1) & \cdots & \underline{h}_i(0) \end{bmatrix} \tag{4.9}$$

In other words, the cyclic prefix of $\overline{\boldsymbol{x}}_i$ renders the convolution in (4.7) cyclic, and we can write

$$\boxed{\boldsymbol{y}_i = \boldsymbol{h}_i \circledast \boldsymbol{x}_i + \boldsymbol{n}_i} \tag{4.10}$$

where $\boldsymbol{h}_i$ is a length-N zero-padded version of $\underline{\boldsymbol{h}}_i$

$$\boldsymbol{h}_i = \begin{bmatrix} \underline{\boldsymbol{h}}_i \\ \boldsymbol{O}_{(N-P-1)\times 1} \end{bmatrix} \tag{4.11}$$

In the frequency domain, the cyclic convolution (4.10) reduces to the element-by-element operation

$$\boxed{\boldsymbol{\mathcal{Y}}_i = \boldsymbol{\mathcal{H}}_i \odot \boldsymbol{\mathcal{X}}_i + \boldsymbol{\mathcal{N}}_i} \tag{4.12}$$

where $\boldsymbol{\mathcal{H}}_i$, $\boldsymbol{\mathcal{X}}_i$, $\boldsymbol{\mathcal{N}}_i$, and $\boldsymbol{\mathcal{Y}}_i$, are the DFT's of $\boldsymbol{h}_i$, $\boldsymbol{x}_i$, $\boldsymbol{n}_i$, and $\boldsymbol{y}_i$, respectively, i.e.

$$\boldsymbol{\mathcal{H}}_i = \boldsymbol{Q}\boldsymbol{h}_i, \quad \boldsymbol{\mathcal{X}}_i = \frac{1}{\sqrt{N}}\boldsymbol{Q}\boldsymbol{x}_i, \quad \boldsymbol{\mathcal{N}}_i = \frac{1}{\sqrt{N}}\boldsymbol{Q}\boldsymbol{n}_i, \quad \text{and} \quad \boldsymbol{\mathcal{Y}}_i = \frac{1}{\sqrt{N}}\boldsymbol{Q}\boldsymbol{y}_i \tag{4.13}$$

Since $\underline{\boldsymbol{h}}_i$ corresponds to the first $P+1$ elements of $\boldsymbol{h}_i$, and in line with the notational convention (4.2), we can show that

$$\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i = \boldsymbol{\mathcal{H}}_i \tag{4.14}$$

This allows us to write (4.10) in terms of $\underline{\boldsymbol{h}}_i$ as

$$\boxed{\boldsymbol{\mathcal{Y}}_i = \operatorname{diag}(\boldsymbol{\mathcal{X}}_i)\,\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i + \boldsymbol{\mathcal{N}}_i} \tag{4.15}$$

4.2.2 Linear Convolution (Channel)

Starting from (4.6), we can also extract a constituent relationship between the input and output prefixes

$$\underline{\boldsymbol{y}}_i = \begin{bmatrix} \underline{\boldsymbol{H}}_{\mathsf{U}i} & \underline{\boldsymbol{H}}_{\mathsf{L}i} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{x}}_{i-1} \\ \underline{\boldsymbol{x}}_i \end{bmatrix} + \underline{\boldsymbol{n}}_i \tag{4.16}$$

This can be used to show that the prefix sequences $\{\underline{x}_i\}$ and $\{\underline{y}_i\}$ are related together with the channel $\underline{h}_i$ through a linear convolution, i.e.

$$\boxed{\underline{y}_i = \underline{h}_i * \underline{x}_i + \underline{n}_i} \tag{4.17}$$

The roles of $\underline{\boldsymbol{x}}_i$ and $\underline{\boldsymbol{h}}_i$ can also be interchanged in (4.16) and we can rewrite it as

$$\boxed{\underline{\boldsymbol{y}}_i = \underline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i + \underline{\boldsymbol{n}}_i} \tag{4.18}$$

where

$$\underline{\boldsymbol{X}}_i = \underline{\boldsymbol{X}}_{\mathsf{U}i} + \underline{\boldsymbol{X}}_{\mathsf{L}i} \tag{4.19}$$

$$\underline{\boldsymbol{X}}_{\mathsf{U}i} = \begin{bmatrix} 0 & \underline{x}_{i-1}(P-1) & \cdots & \underline{x}_{i-1}(0) \\ 0 & 0 & \cdots & \underline{x}_{i-1}(1) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \underline{x}_{i-1}(P-1) \end{bmatrix} \tag{4.20}$$

and

$$\underline{\boldsymbol{X}}_{\mathsf{L}i} = \begin{bmatrix} \underline{x}_i(0) & 0 & \cdots & 0 \\ \underline{x}_i(1) & \underline{x}_i(0) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \underline{x}_i(P-1) & \cdots & \underline{x}_i(0) & 0 \end{bmatrix} \tag{4.21}$$

4.2.3 Total Convolution (Channel)

The sequence $\{\overline{y}_i\}$ at the channel output is related naturally to the input sequence $\{\overline{x}_i\}$ through linear convolution with the channel

$$\overline{y}_i = \underline{h}_i * \overline{x}_i + \overline{n}_i \tag{4.22}$$

Now, in line with the notation adopted here, define $\overline{\boldsymbol{h}}_i$ to be another zero-padded version of $\boldsymbol{h}_i$ of length $N+P$ [3]

$$\overline{\boldsymbol{h}}_i \triangleq \begin{bmatrix} \underline{\boldsymbol{h}}_i \\ \boldsymbol{O}_{(N-1)\times 1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_i \\ \boldsymbol{O}_{P\times 1} \end{bmatrix}$$

Zero padding does not affect linear convolution, and the output sequence $\{\overline{y}_i\}$ is still related to $\{\overline{x}_i\}$ though linear convolution with $\overline{\boldsymbol{h}}_i$. Thus, the convolution (4.22) can be written in the (notationally more consistent) form

$$\boxed{\overline{y}_i = \overline{h}_i * \overline{x}_i + \overline{n}_i} \tag{4.23}$$

We can also express the input/output relationship in matrix form. In fact, we do that by concatenating the matrix output relationships (4.15) and (4.18) for the circular and convolutional channels

$$\boxed{\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i \underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i} \tag{4.24}$$

[3] Recall that $\boldsymbol{h}_i$, defined in (4.11), is too a zero-padded version of $\underline{\boldsymbol{h}}_i$, of length N.

where, in line with the notational convention (4.1),

$$\overline{\boldsymbol{X}}_i \triangleq \begin{bmatrix} \underline{\boldsymbol{X}}_i \\ \text{diag}(\boldsymbol{\mathcal{X}}_i)\tilde{\boldsymbol{Q}}_{P+1} \end{bmatrix}, \quad \overline{\boldsymbol{\mathcal{Y}}}_i = \begin{bmatrix} \underline{\boldsymbol{y}}_i \\ \boldsymbol{\mathcal{Y}}_i \end{bmatrix}, \quad \text{and} \quad \overline{\boldsymbol{\mathcal{N}}}_i = \begin{bmatrix} \underline{\boldsymbol{n}}_i \\ \boldsymbol{\mathcal{N}}_i \end{bmatrix} \tag{4.25}$$

Remark To maintain perspective, the various relationships are summarized in the Table 4.2 below. The last column of the table also summarizes the input/output relationship for sparse channels where I_{c} denotes the index set of the active taps in $\underline{\boldsymbol{h}}_i$.

Table 4.2: *Input/output relationships for the circular, linear, and total channels*

CHANNEL	SEQUENCE RELATIONSHIP	MATRIX RELATIONSHIP	MATRIX RELATIONSHIP Sparse channel
Linear	$\underline{y}_i = \underline{h}_i * \underline{x}_i + \underline{n}_i$	$\underline{\boldsymbol{y}}_i = \underline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i + \underline{\boldsymbol{n}}_i$	$\underline{\boldsymbol{y}}_i = \underline{\boldsymbol{X}}_i\boldsymbol{I}^*_{I_{\text{c}}}\underline{\boldsymbol{h}}_{i_{I_{\text{c}}}} + \underline{\boldsymbol{n}}_i$
Circular	$\boldsymbol{y}_i = \boldsymbol{h}_i \circledast \boldsymbol{x}_i + \boldsymbol{n}_i$ $\boldsymbol{\mathcal{Y}}_i = \boldsymbol{\mathcal{H}} \odot \boldsymbol{\mathcal{X}}_i + \boldsymbol{\mathcal{N}}_i$	$\boldsymbol{\mathcal{Y}}_i = \text{diag}\left(\boldsymbol{\mathcal{X}}_i\right)\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i + \boldsymbol{\mathcal{N}}_i$	$\boldsymbol{\mathcal{Y}}_i = \text{diag}(\boldsymbol{\mathcal{X}}_i)\tilde{\boldsymbol{Q}}_{P+1}\boldsymbol{I}_{I_{\text{c}}}\underline{\boldsymbol{h}}_{i_{I_{\text{c}}}} + \boldsymbol{\mathcal{N}}_i$
Total	$\overline{y}_i = \overline{h}_i * \overline{x}_i + \overline{n}_i$	$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i$	$\overline{\boldsymbol{y}}_i = \overline{\boldsymbol{X}}_i\boldsymbol{I}^*_{I_{\text{c}}}\underline{\boldsymbol{h}}_{i_{I_{\text{c}}}} + \overline{\boldsymbol{n}}_i$

From this table, we can also appreciate the notation adopted in this chapter. Note in particular the first column of the table where only similar variables (underline, calligraphic, overlined, or otherwise) are related.

Before we end this section, we briefly discuss how the input/output relationships change when the channel is sparse and in the presence of pilots.

4.2.4 Pilot/Output Relationships

Pilots are needed to kick-start the estimation process. Fortunately, pilots are sent as part of the OFDM symbol and are an integral part of several standards as they are needed for time and frequency recovery in addition to channel estimation. Within each symbol, pilots are maximally placed away from each other. Channel gains at frequency bins close to the pilots locations have a higher estimation accuracy than those further away. As such, pilot locations are rotated from one OFDM symbol to the next. From (4.15), the pilot/output equation is given

$$\boldsymbol{\mathcal{Y}}_{iI_{\text{p}}} = \text{diag}\left(\boldsymbol{\mathcal{X}}_i\right)_{I_{\text{p}}}\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i + \boldsymbol{\mathcal{N}}_{iI_{\text{p}}} \tag{4.26}$$

where $I_{\mathsf{p}} = \{i_1, i_2, \ldots, i_{L_{\mathsf{p}}}\}$ denotes the index set of the pilot bins.[4] Clearly, only the cyclic channel can make use of the pilot information. [5]

4.2.5 Channel Model

In this chapter, we assume a block fading model where the channel is assumed constant over any OFDM supersymbol (i.e. over the OFDM symbol and associated cyclic prefix). Apart from this constraint, we allow the channel to exhibit any behavior from one symbol to the next. [6] In addition to the finite delay spread property, we make use of any or all of the following constraints.

Frequency (tap) correlation The channel is assumed to be Gaussian distributed

$$\underline{\boldsymbol{h}}_i \sim \boldsymbol{\mathcal{N}}(\boldsymbol{m}, \boldsymbol{\Pi}) \tag{4.27}$$

as when the channel is Rayleigh or Rician fading.

Time correlation We assume that the time correlation between $\underline{\boldsymbol{h}}_i$ and $\underline{\boldsymbol{h}}_{i+1}$ (corresponding to the impulse response for two consecutive OFDM symbols) exhibits itself through a state-space model

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \tag{4.28}$$

where $\boldsymbol{u}_i$ is a white Gaussian noise with variance σ_u^2. The matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are a function of the Doppler spread, the power-delay profile (frequency correlation), and the transmit filter (Appendix A shows how this information can be used to construct $\boldsymbol{F}$ and $\boldsymbol{G}$). We assume that this information is known at the receiver. The model (4.28) thus captures both frequency and time correlation.

[4] The pilot index set might change from one OFDM symbol to the next as the number of pilots and/or their location might change.

[5] The effect of the pilot symbols is felt through the linear channel as well. However, this effect is contaminated by the effect of the other (unknown) symbols (through convolution) in a way that makes it difficult to use the output of that channel for initial channel estimation.

[6] This means that the channel changes at the symbol edges only. This might be unrealistic as we expect the channel to change in a graceful manner across the duration of the symbol. We choose this extreme block fading model because we would like to test our receiver against extreme time variations without dealing with intercarrier interference. This also allows us to use the finite delay spread property, reducing the number of parameters to be estimated from N to $P + 1$, and hence reducing the number of pilots needed. If the time-variant behavior is large enough to cause big intercarrier interference, then OFDM transmission might not be the best way to go as the CP overload would not be justified in this case.

Sparsity When the channel is sparse, we only need to estimate its active taps $\underline{\boldsymbol{h}}_{iI_c}$. Those active taps could also assume frequency correlation (similar to (4.27)) or could evolve in time following a state-space model similar to (4.28).

4.3 The EM Algorithm for Joint Channel and Data Estimation

4.3.1 The EM Algorithm

Ideally, we estimate $\underline{\boldsymbol{h}}_i$ using the total input/output relationship

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i \underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i \tag{4.29}$$

We achieve this by maximizing the log-likelihood function

$$\hat{\underline{\boldsymbol{h}}}_i^{\text{MAP}} = \arg\max_{\underline{h}_i} \ln p\left(\underline{\boldsymbol{h}}_i | \overline{\boldsymbol{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i\right)$$

When the noise is white Gaussian this reduces to the least-squares problem

$$\hat{\underline{\boldsymbol{h}}}_i = \arg\min_{\underline{h}_i} \|\overline{\boldsymbol{\mathcal{Y}}}_i - \overline{\boldsymbol{X}}_i \underline{\boldsymbol{h}}_i\|^2_{\sigma_n^{-2}}$$

Since the input $\overline{\boldsymbol{X}}_i$ is not observable, we maximize instead an averaged form of the log-likelihood function as mandated by the EM algorithm. In this case, the estimate of $\underline{\boldsymbol{h}}_i$ is calculated iteratively, with the estimate at the jth iteration given by

$$\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\max_{\underline{h}_i} E_{\overline{X}_i | \mathcal{Y}_i, \hat{\underline{h}}_i^{(j)}} \ln p\left(\underline{\boldsymbol{h}}_i | \overline{\boldsymbol{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i\right) \tag{4.30}$$

This involves two steps; an expectation step and a maximization step (hence the name expectation-maximization). In the following, we discuss these steps respectively.

4.3.2 The Expectation Step: Mean-Square Estimation of Data

The log-likelihood function $\ln p(\underline{\boldsymbol{h}}_i | \overline{\boldsymbol{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i)$ shows quadratic dependence on the input $\overline{\boldsymbol{X}}_i$. Hence, to perform the expectation step, we need to calculate the first two moments of $\overline{\boldsymbol{X}}_i$, or equivalently, the mean $E[\overline{\boldsymbol{X}}_i]$ and covariance $\text{Cov}[\overline{\boldsymbol{X}}_i^*]$ (see equation (4.39) further ahead). These two expectations are calculated given the output $\overline{\boldsymbol{\mathcal{Y}}}_i$ and the channel $\underline{\boldsymbol{h}}_i$ or

an estimate of it, i.e.

$$E[\overline{\boldsymbol{X}}_i|\overline{\boldsymbol{\mathcal{Y}}}_i, \hat{\underline{\boldsymbol{h}}}_i] \quad \text{and} \quad \text{Cov}[\overline{\boldsymbol{X}}_i^*]$$

Since the elements of $\overline{\boldsymbol{X}}_i$ are *linearly* dependent on the frequency domain symbol $\boldsymbol{\mathcal{X}}_i$, the elements of $E[\overline{\boldsymbol{X}}_i]$ will exhibit the same functional dependence on $E[\boldsymbol{\mathcal{X}}_i]$ and hence it is enough to calculate the expectation

$$E[\boldsymbol{\mathcal{X}}_i|\overline{\boldsymbol{\mathcal{Y}}}_i, \hat{\underline{\boldsymbol{h}}}_i]$$

Furthermore, we approximate this expectation with $E[\boldsymbol{\mathcal{X}}_i|\boldsymbol{\mathcal{Y}}_i, \hat{\underline{\boldsymbol{h}}}_i]$ or, equivalently, with $E[\boldsymbol{\mathcal{X}}_i|\boldsymbol{\mathcal{Y}}_i, \hat{\boldsymbol{\mathcal{H}}}_i]$. This approximation is convenient as the expectation $E[\boldsymbol{\mathcal{X}}_i|\boldsymbol{\mathcal{Y}}_i, \hat{\boldsymbol{\mathcal{H}}}_i]$ can be calculated on an element-by-element basis, i.e. by evaluating the scalar expectations

$$E[\mathcal{X}_i(l)|\mathcal{Y}_i(l), \hat{\mathcal{H}}_i(l)] \quad \text{for} \;\; l = 1, \ldots, N \tag{4.31}$$

Now, assuming that $\mathcal{X}_i(l)$ takes on its values from the alphabet

$$A = \{A_1, A_2, \ldots, A_{|A|}\} \tag{4.32}$$

with equal probability, we have that

$$f(\mathcal{X}_i(l)|\mathcal{Y}_i(l)) = \frac{e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}(l)\mathcal{X}_i(l)|^2}{\sigma_n^2}}}{\sum_{j=1}^{|A|} e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}(l)A_j|^2}{\sigma_n^2}}} \tag{4.33}$$

This can be used to show that

$$E[\mathcal{X}_i(l)|\mathcal{Y}_i(l)] = \frac{\sum_{j=1}^{j=|A|} A_j e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}(l)A_j|^2}{\sigma_n^2}}}{\sum_{j=1}^{|A|} e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}(l)A_j|^2}{\sigma_n^2}}} \tag{4.34}$$

This is nothing but the optimum MMSE estimate of $\mathcal{X}_i(l)$ given the output $\mathcal{Y}_i(l)$. [7]

As we did in the first moment case, we approximate the covariance calculation $\text{Cov}[\overline{\boldsymbol{X}}_i^*|\overline{\boldsymbol{\mathcal{Y}}}_i, \hat{\underline{\boldsymbol{h}}}_i]$

[7] Compare this estimate with the linear MMSE estimate

$$\hat{\mathcal{X}}_i^{\text{MMSE}}(l) = \frac{\sqrt{\mathcal{E}_X}\mathcal{H}^*(l)}{\mathcal{E}_X||\mathcal{H}(l)||^2 + \sigma_n^2}\mathcal{Y}_i(l)$$

This estimate makes use of the average energy of the input and not of the whole information provided by the finite alphabet constraint.

with that of $\mathrm{Cov}[\overline{\boldsymbol{X}}_i^*|\boldsymbol{\mathcal{Y}}_i, \hat{\underline{\boldsymbol{h}}}_i]$. This approximation allows us to use the circular channel to calculate the covariance of $\overline{\boldsymbol{X}}_i$ from the first moment of $\mathcal{X}_i(l)$, calculated in (4.34), and the second moment given by

$$E[|\mathcal{X}_i(l)|^2|\mathcal{Y}_i(l)] = \frac{\sum_{j=1}^{j=|A|} |A_j|^2 e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)A_j|^2}{\sigma_n^2}}}{\sum_{j=1}^{|A|} e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)A_j|^2}{\sigma_n^2}}} \tag{4.35}$$

Appendix B demonstrates how we can use the first two moments of $\boldsymbol{\mathcal{X}}_i$ to calculate the covariance of $\overline{\boldsymbol{X}}_i^*$.

4.3.3 The Maximization Step: Channel Estimation

With the two moments of the input at hand, we can proceed to perform the expectation and maximization steps and obtain the channel estimate. Recall that the EM iteration is expressed by

$$\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\max_{\underline{h}_i} \; E_{\mathcal{X}_i|\overline{\mathcal{Y}}_i, \hat{\underline{h}}_i^{(j)}} \left[\ln p(\underline{\boldsymbol{h}}_i|\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i)\right] \tag{4.36}$$

where averaging is performed conditioned on the output and the previous estimate of the channel $\hat{\underline{\boldsymbol{h}}}_i^{(j)}$. Now, using Bayes rule, we can write

$$\ln p(\underline{\boldsymbol{h}}_i|\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i) \;\; = \;\; \ln p(\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i|\underline{\boldsymbol{h}}_i) + \ln p(\underline{\boldsymbol{h}}_i) - \ln p(\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i)$$

So, the EM iteration can be expressed as

$$\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\max_{\underline{h}_i} E_{\mathcal{X}_i|\overline{\mathcal{Y}}_i, \hat{\underline{h}}_i^{(j)}} \ln p(\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i|\underline{\boldsymbol{h}}_i) + \ln p(\underline{\boldsymbol{h}}_i)$$

Employing Bayes rule again yields

$$\begin{aligned} \ln p(\boldsymbol{\mathcal{X}}_i, \overline{\boldsymbol{\mathcal{Y}}}_i|\underline{\boldsymbol{h}}_i) &= \ln p(\overline{\boldsymbol{\mathcal{Y}}}_i|\boldsymbol{\mathcal{X}}_i, \underline{\boldsymbol{h}}_i) + \ln p(\boldsymbol{\mathcal{X}}_i|\underline{\boldsymbol{h}}_i) \\ &= \ln p(\overline{\boldsymbol{\mathcal{Y}}}_i|\boldsymbol{\mathcal{X}}_i, \underline{\boldsymbol{h}}_i) + \ln p(\boldsymbol{\mathcal{X}}_i) \end{aligned}$$

where the second lines follows from the fact that $\boldsymbol{\mathcal{X}}_i$ and $\underline{\boldsymbol{h}}_i$ are independent. With this decomposition, we can finally express the EM iteration as

$$\boxed{\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\max_{\underline{h}_i} E_{\mathcal{X}_i|\overline{\mathcal{Y}}_i,\hat{\underline{h}}_i^{(j)}} \ln p(\overline{\boldsymbol{\mathcal{Y}}}_i|\boldsymbol{\mathcal{X}}_i,\underline{\boldsymbol{h}}_i) + \ln p(\underline{\boldsymbol{h}}_i)} \tag{4.37}$$

Now assuming that the channel values are equally probable and that the noise in the input/output equation (4.29) is white Gaussian, the update equation (4.37) reduces to

$$\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\min_{\underline{h}_i} E_{\mathcal{X}_i|\overline{\mathcal{Y}}_i,\hat{\underline{h}}_i^{(j)}} \|\overline{\boldsymbol{\mathcal{Y}}}_i - \overline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i\|^2_{\sigma_n^{-2}} \tag{4.38}$$

By expanding the norm, taking the expectation, and completing the squares, we can alternatively write (4.38) as

$$\boxed{\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\min_{\underline{h}_i} \|\overline{\boldsymbol{\mathcal{Y}}}_i - E[\overline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\|^2_{\sigma_n^{-2}} + \|\underline{\boldsymbol{h}}_i\|^2_{\mathrm{Cov}[\overline{X}_i^*]}} \tag{4.39}$$

4.3.4 The Maximization Step: Incorporating Frequency Correlation

In a wireless environment, the channel $\underline{\boldsymbol{h}}_i$ usually follows a Gaussian distribution, i.e. $\underline{\boldsymbol{h}}_i$ assumes the pdf

$$p(\underline{\boldsymbol{h}}_i) = \frac{1}{((2\pi)^{P+1}|\boldsymbol{\Pi}|)^{1/2}} e^{-\|\underline{\boldsymbol{h}}_i\|^2_{\Pi^{-1}}} \tag{4.40}$$

where $\boldsymbol{\Pi}$ is the channel covariance matrix. By incorporating this pdf into the EM iteration (4.37), we basically add the term $\|\underline{\boldsymbol{h}}_i\|^2_{\Pi^{-1}}$ to the right-hand side of (4.39). In other words, we have

$$\boxed{\hat{\underline{\boldsymbol{h}}}_i^{(j+1)} = \arg\min_{\underline{h}_i} \left\{ \left\|\overline{\boldsymbol{\mathcal{Y}}}_i + E[\overline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\right\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}\mathrm{Cov}[\overline{X}_i^*]} + \|\underline{\boldsymbol{h}}_i\|^2_{\Pi^{-1}} \right\}} \tag{4.41}$$

4.3.5 The Maximization Step: Incorporating Time Correlation

We now incorporate the additional information provided by the time-correlation– i.e., the correlation between $\underline{\boldsymbol{h}}_i$ and $\underline{\boldsymbol{h}}_{i+1}$. For simplicity, we assume that this correlation arises from

the state-space model [8]

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \tag{4.42}$$

where the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are assumed to be known at the receiver and where $\boldsymbol{u}_i$ is Gaussian noise with zero mean and covariance matrix $\sigma_u^2\boldsymbol{I}$. The initial state $\underline{\boldsymbol{h}}_0$ is also assumed to be Gaussian with zero-mean and covariance matrix $\boldsymbol{\Pi}_0$. The state-space model (4.42) thus captures both the frequency and time correlations. When the input is fully available, this model can be used together with the input/output equation

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i \tag{4.43}$$

to dynamically estimate the channel $\underline{\boldsymbol{h}}_i$ using the Kalman filter. Our problem, however, is that the input $\overline{\boldsymbol{X}}_i$ is not always available in full, and the Kalman filter has to be modified in the EM sense, as we did in the presence of frequency correlation information (See Appendix C).

4.3.6 The Kalman Filter

When the channel exhibits a dynamic behavior, as in (4.42), the optimum channel estimate is always obtained using a Kalman filter. This applies whether the input is perfectly known, the input is known in detected form only, only the pilot part of the input is known, or nothing is known about the input. Thus, when the input is perfectly known, we employ the Kalman filter to the state-space model

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \tag{4.44}$$

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i\underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i \tag{4.45}$$

[8] The channel could follow a more sophisticated model in which $\underline{\boldsymbol{h}}_{i+1}$ would depend not only on $\underline{\boldsymbol{h}}_i$ but also on past values of the impulse response. Extending our approach to this case is straightforward.

Table 4.3: *Modifications to the Kalman filter (4.46)–(4.50) under various input knowledge conditions*

SITUATION	SUBSTITUTION		
	$\overline{\boldsymbol{X}}_i$	$\overline{\boldsymbol{\mathcal{Y}}}_i$	$\boldsymbol{I}_{N+P}$
Input in detected form	$\begin{bmatrix} E\left[\overline{\boldsymbol{X}}_i\right] \\ \mathrm{Cov}\left[\overline{\boldsymbol{X}}_i^*\right]^{1/2} \end{bmatrix}$	$\begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_i \\ \boldsymbol{0}_{P\times 1} \end{bmatrix}$	$\boldsymbol{I}_{N+2P}$
Pilot info	$\mathrm{diag}\left(\boldsymbol{\mathcal{X}}_i\right)_{I_\mathsf{p}} \tilde{\boldsymbol{Q}}_{P+1}$	$\boldsymbol{\mathcal{Y}}_{iI_\mathsf{p}}$	$\boldsymbol{I}_{\lvert I_\mathsf{p}\rvert}$
No input info (Applies for $i \geq 1$)	$\boldsymbol{O}$	$\overline{\boldsymbol{\mathcal{Y}}}_i$	$\boldsymbol{I}_{N+P}$
Sparse channel	$\overline{\boldsymbol{X}}_i \underline{\boldsymbol{I}}^*_{I_\mathsf{c}}$	$\overline{\boldsymbol{\mathcal{Y}}}_i$	$\boldsymbol{I}_{N+P}$
No cyclic prefix info	$\mathrm{diag}\left(\boldsymbol{\mathcal{X}}_i\right) \tilde{\boldsymbol{Q}}_{P+1}$	$\boldsymbol{\mathcal{Y}}_i$	$\boldsymbol{I}_N$

In this case, the Kalman filter is given by [47]

$$\boldsymbol{R}_{e,i} = \sigma_n^2 \boldsymbol{I}_{N+P} + \overline{\boldsymbol{X}}_i \boldsymbol{P}_{i|i-1} \overline{\boldsymbol{X}}_i^* \quad \boldsymbol{P}_{0|-1} = \boldsymbol{\Pi}_0 \tag{4.46}$$

$$\boldsymbol{K}_{f,i} = \boldsymbol{P}_{i|i-1} \overline{\boldsymbol{X}}_i^* \boldsymbol{R}_{e,i}^{-1} \tag{4.47}$$

$$\underline{\hat{\boldsymbol{h}}}_{i|i} = \left(\boldsymbol{I}_{N+P} - \boldsymbol{K}_{f,i} \overline{\boldsymbol{X}}_i\right) \underline{\hat{\boldsymbol{h}}}_{i|i-1} + \boldsymbol{K}_{f,i} \overline{\boldsymbol{\mathcal{Y}}}_i \quad \underline{\boldsymbol{h}}_{0|-1} = \boldsymbol{0} \tag{4.48}$$

$$\underline{\hat{\boldsymbol{h}}}_{i+1|i} = \boldsymbol{F} \underline{\hat{\boldsymbol{h}}}_{i|i} \tag{4.49}$$

$$\boldsymbol{P}_{i+1|i} = \boldsymbol{F}_i \left(\boldsymbol{P}_{i|i-1} - \boldsymbol{K}_{f,i} \boldsymbol{R}_{e,i} \boldsymbol{K}_{f,i}^*\right) \boldsymbol{F}^* + \frac{1}{\sigma_u^2} \boldsymbol{G}\boldsymbol{G}^* \tag{4.50}$$

The desired estimate is $\underline{\hat{\boldsymbol{h}}}_{i|i}$.

When the input is not perfectly known, we perform the change of variables

$$\overline{\boldsymbol{X}}_i \longrightarrow \begin{bmatrix} E\left[\overline{\boldsymbol{X}}_i\right] \\ \mathrm{Cov}\left[\overline{\boldsymbol{X}}_i^*\right]^{1/2} \end{bmatrix} \quad \overline{\boldsymbol{\mathcal{Y}}}_i \longrightarrow \begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_i \\ \boldsymbol{0}_{P\times 1} \end{bmatrix} \tag{4.51}$$

Remarks

- Table 4.3 summarizes the substitutions to perform on the Kalman implementation (4.46)-(4.50) in various situations. The first line in the table applies when the input is known in detected form. The substitution is obtained by appealing to the EM algorithm as we show in Appendix C.

- The lower entries in the table are straightforward to obtain. For example, when only the pilot information is available, we replace the input/output equation (4.45) by

$$\boldsymbol{\mathcal{Y}}_{iI_\mathrm{p}} = \mathrm{diag}(\boldsymbol{\mathcal{X}}_i)_{I_\mathrm{p}} \tilde{\boldsymbol{Q}}_{P+1} \underline{\boldsymbol{h}}_i + \boldsymbol{\mathcal{N}}_i$$

  and the substitutions in the second line of the table follow accordingly.

- When compound situations happen, we employ the substitutions in tandem. Thus, when the channel is sparse and only the pilot part of the input is available, we perform the substitution

$$\begin{array}{ccccc} \overline{\boldsymbol{X}}_i & \overset{\text{sparsity}}{\longrightarrow} & \overline{\boldsymbol{X}}_i \boldsymbol{I}^*_{I_\mathrm{c}} & \overset{\text{pilots}}{\longrightarrow} & \mathrm{diag}(\boldsymbol{\mathcal{X}}_i)_{I_\mathrm{p}} \tilde{\boldsymbol{Q}}_{P+1} \boldsymbol{I}^*_{I_\mathrm{c}} \\ \overline{\boldsymbol{\mathcal{Y}}}_i & \overset{\text{sparsity}}{\longrightarrow} & \overline{\boldsymbol{\mathcal{Y}}}_i & \overset{\text{pilots}}{\longrightarrow} & \overline{\boldsymbol{\mathcal{Y}}}_{iI_\mathrm{p}} \\ \boldsymbol{I}_{N+P} & \overset{\text{sparsity}}{\longrightarrow} & \boldsymbol{I}_{N+P} & \overset{\text{pilots}}{\longrightarrow} & \boldsymbol{I}_{|I_\mathrm{p}|} \end{array}$$

4.4 Incorporating Forward and Backward Time Correlation

So far, channel estimation has been performed in a causal manner. In other words, the receiver estimates $\underline{\boldsymbol{h}}_i$ by using past and current data $\{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=0}^{i}$. In this way, channel (and data) recovery can be done with no latency. If we relax the latency constraint, we can use all data symbols $\{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=0}^{T}$ (past, current, and future) to estimate the IR $\underline{\boldsymbol{h}}_i$.

When the channel exhibits a dynamic behavior, the optimum channel estimate is obtained by employing a forward-backward Kalman filter. Interestingly, as in the forward Kalman filter case, once we describe the solution for the perfectly known input case, the solution for the other cases follow by some substitution (as described by Table 4.3). Thus, the same basic form of the filter can be used regardless of our degree of knowledge about the input.

4.4.1 The Forward-Backward Kalman

Consider the state-space model

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \tag{4.52}$$

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i \underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i \tag{4.53}$$

The MAP estimate of the sequence $\{\underline{\boldsymbol{h}}_k\}_{k=0}^{T}$ given the output sequence $\{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=0}^{T}$ is obtained by employing the forward-backward Kalman filter. As the name suggests, the filter consists of two runs, forward and backward:

Forward Run: The forward run coincides with the Kalman filter. Thus, the receiver runs recursions (4.46)-(4.50) which are used to obtain $\hat{\underline{\boldsymbol{h}}}_{i|i-1}, \hat{\underline{\boldsymbol{h}}}_{i|i}$, and $\boldsymbol{P}_{i|i-1}$ for $i = 0, 1, \ldots, N$.

Backward Run: For $i = N, N-1, \ldots, 0$, calculate

$$\boldsymbol{\lambda}_{i|N} = \left(\boldsymbol{I}_{P+N} - \overline{\boldsymbol{X}}_i^* \boldsymbol{K}_{f,i}^*\right) \boldsymbol{F}_i^* \boldsymbol{\lambda}_{i+1|N} + \overline{\boldsymbol{X}}_i \boldsymbol{R}_{e,i}^{-1} \left(\overline{\boldsymbol{\mathcal{Y}}}_i - \overline{\boldsymbol{X}}_i \hat{\underline{\boldsymbol{h}}}_{i|i-1}\right) \tag{4.54}$$

$$\hat{\underline{\boldsymbol{h}}}_{i|N} = \hat{\underline{\boldsymbol{h}}}_{i|i-1} + \boldsymbol{P}_{i|i-1} \boldsymbol{\lambda}_{i|N} \tag{4.55}$$

starting from $\boldsymbol{\lambda}_{N+1|N} = \boldsymbol{0}$.

The desired estimate is $\hat{\underline{\boldsymbol{h}}}_{i|N}$.

4.4.2 Summary of the Algorithm

At this point, we have all the elements necessary to implement the OFDM receiver. So consider a sequence of OFDM symbols passing through a block fading channel. The channel could exhibit no correlation, frequency correlation, or both frequency and time correlation.

Pilot Placement There are different requirements on the number of pilots needed depending on the a priori information available and the type of processing the receiver does (see Table 4.4). At one extreme, when there is no correlation information, the algorithm needs as many pilots in each symbol as the number of active channel taps. At the other extreme, when both time and frequency correlation information is available and when the receiver performs (forward-backward) smoothing, it is enough to have a few pilots in any one symbol.

Initial Channel Estimation Table 4.5 summarizes the different methods to perform initial channel estimation. The appropriate method depends on the available correlation information and the type of processing that is desired.

Table 4.4: *Conditions on pilot number and placement*

Pilot Conditions	Available Channel Info			
	NO Correlation Info	Frequency Correlation	Frequency and Time Correlation (filtering)	Frequency and Time Correlation (smoothing)
Presence of Pilots in symbols	Need pilots in each symbol	Need pilots in each symbol	Need pilots in first symbol	Need pilots in some symbol
Number of pilots per symbol	As many pilots as channel taps	Nonzero number of pilots	Nonzero number of pilots	Nonzero number of pilots

Table 4.5: *How to perform the initial channel estimation step under various conditions*

Channel Info	Initial Channel Estimate
NO correlation Info	$\min_{\underline{h}_i} \|\boldsymbol{\mathcal{Y}}_{iI_\mathrm{p}} - \mathrm{diag}(\boldsymbol{\mathcal{X}})_{I_\mathrm{p}} \boldsymbol{Q}_{P+1} \underline{\boldsymbol{h}}_i\|^2$
Frequency correlation	$\min_{\underline{h}_i} \|\boldsymbol{\mathcal{Y}}_{iI_\mathrm{p}} - \mathrm{diag}(\boldsymbol{\mathcal{X}})_{I_\mathrm{p}} \boldsymbol{Q}_{P+1} \underline{\boldsymbol{h}}_i\|^2 + \|\underline{\boldsymbol{h}}_i\|^2_{\Pi^{-1}}$
Frequency and Time Correlation (Filtering)	Employ the Kalman filter (4.46)-(4.50) with substitution from Table 4.3 depending on whether there are pilots or not
Frequency and Time Correlation (Smoothing)	Employ the foward-backward Kalman filter (4.46)-(4.50), (4.54),(4.55) with substitution from Table 4.3 depending on whether there are pilots or not

Data Detection Perform mean-square estimation of the first and second moments of the frequency domain symbol $\boldsymbol{\mathcal{X}}_i$ using (4.34) and (4.35). Use that to construct the first and second moments $E[\overline{\boldsymbol{X}}_i]$ and $\mathrm{Cov}[\overline{\boldsymbol{X}}_i^*]$, as described in Appendix B.

Channel Estimation Update Table 4.6 summarizes the various methods for updating the channel estimate. Again, the particular update used depends on the correlation information available and the type of processing desired (i.e. filtering or smoothing).

Iterate between Estimation and Detection Iterate between channel estimation and data detection. Simulations show that a few iterations are enough for convergence.

Table 4.6: *How to update the channel estimate under various conditions*

Channel Info	Channel Update
No correlation Info	$\min_{\underline{h}_i} \|\overline{\boldsymbol{\mathcal{Y}}}_i - E[\overline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\|^2 + \|\underline{\boldsymbol{h}}_i\|^2_{\mathrm{Cov}[\overline{X}_i^*]}$
Frequency correlation	$\min_{\underline{h}_i} \|\overline{\boldsymbol{\mathcal{Y}}}_i - E[\overline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\|^2 + \|\underline{\boldsymbol{h}}_i\|^2_{\mathrm{Cov}[\overline{X}_i^*]} + \|\underline{\boldsymbol{h}}_i\|^2_{\Pi^{-1}}$
Frequency and Time Correlation (Filtering)	Employ Kalman filter (4.46)-(4.50) with the substitution $\overline{\boldsymbol{X}}_i \longrightarrow \begin{bmatrix} E\left[\overline{\boldsymbol{X}}_i\right] \\ \mathrm{Cov}\left[\overline{\boldsymbol{X}}_i^*\right]^{1/2} \end{bmatrix}$, $\overline{\boldsymbol{\mathcal{Y}}}_i \longrightarrow \begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_i \\ \boldsymbol{0}_{P\times 1} \end{bmatrix}$, $\boldsymbol{I}_{N+P} \longrightarrow \boldsymbol{I}_{N+2P}$
Frequency and Time Correlation (Smoothing)	Employ the FB-Kalman filter (4.46)-(4.50), (4.54),(4.55) with substitution $\overline{\boldsymbol{X}}_i \longrightarrow \begin{bmatrix} E\left[\overline{\boldsymbol{X}}_i\right] \\ \mathrm{Cov}\left[\overline{\boldsymbol{X}}_i^*\right]^{1/2} \end{bmatrix}$, $\overline{\boldsymbol{\mathcal{Y}}}_i \longrightarrow \begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_i \\ \boldsymbol{0}_{P\times 1} \end{bmatrix}$, $\boldsymbol{I}_{N+P} \longrightarrow \boldsymbol{I}_{N+2P}$

4.5 Simulations

4.5.1 Simulation Setup

We consider an OFDM system that transmits a sequence of 5 symbols each with 64 carriers and a cyclic prefix of length $P = 15$. The input data is 16 QAM mapped from a binary bit stream through gray coding. We will use two pilot configurations. The first employs 16 pilots in the first symbol and x number of them in the subsequent four symbols with $1 \leq x \leq 16$. We denote this configuration by $16xxxx$. The second configuration, denoted $xx16xx$, is a cyclic rearrangement of the first with the 16 pilots placed in the middle (3rd) symbol and x pilots in the other symbols.

The channel IR consists of 16 complex taps (the maximum length possible that avoids intersymbol interference). The initial IR $\underline{\boldsymbol{h}}_0$ has an exponential delay profile $E[|\underline{h}_0(k)|^2] = e^{-0.2k}$. For $i \geq 0$, $\underline{\boldsymbol{h}}_i$ is generated according to the dynamical model (4.42)

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \tag{4.56}$$

Both $\boldsymbol{F}$ and $\boldsymbol{G}$ are diagonal matrices. Specifically, we set $\boldsymbol{F} = f\boldsymbol{I}$ with $0 < f < 1$ and set

the diagonal entries of $\boldsymbol{G}$ as $G(k,k) = \sqrt{(1-f^2)E[|\underline{h}_0(k)|^2]}$. The state noise $\boldsymbol{u}_i$ is iid with unit variance. This ensures that the channel maintains the same (exponential) delay profile at subsequent time instants. Throughout the simulations, we run the EM algorithm for 10 iterations.

4.5.2 Effect of Time Variation

Figure 4.1 demonstrates the effectiveness of the Kalman filter in dealing with different degrees of time variation ($f = .1, .3, .5, .7$). Specifically, we implement the $16xxxx$ pilot configuration for $x = 4, 8, 12, 16$. We observe that the BER curves will saturate when the time variation is excessive enough (i.e., f is small enough). The only exception is the 16-pilot case in which the BER decreases linearly with SNR regardless of how sever the time variation is. This is understood given the fact that 16 is actually the number of channel taps. Figure 4.2 demonstrates the same situation for the FB-Kalman receiver. The only difference is that this receiver is tested with the pilot configuration $xx16xx$. The observations above remain valid for this receiver too.

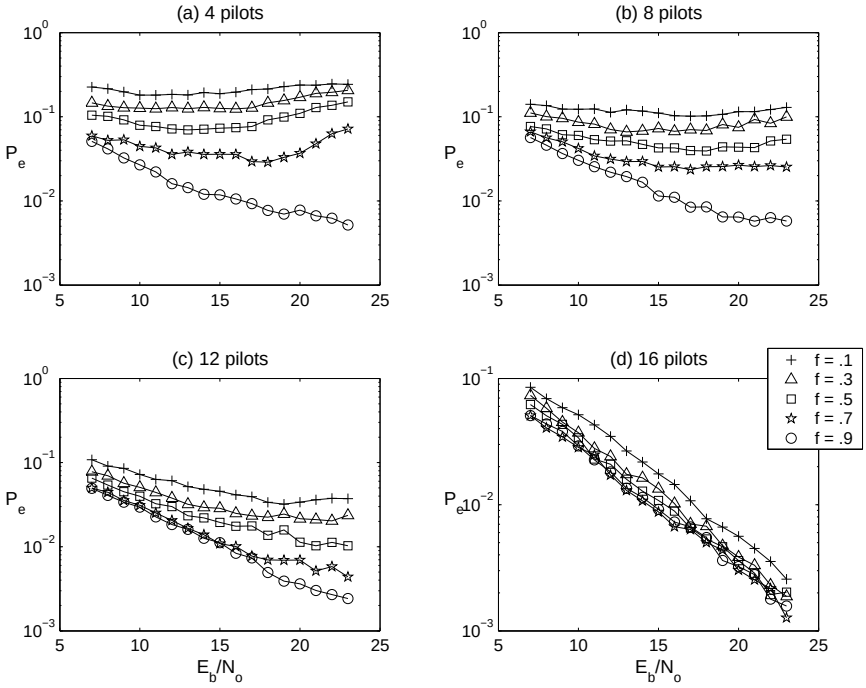

Figure 4.1: BER curves for the Kalman based receiver for different number of pilots and various degrees of time variation

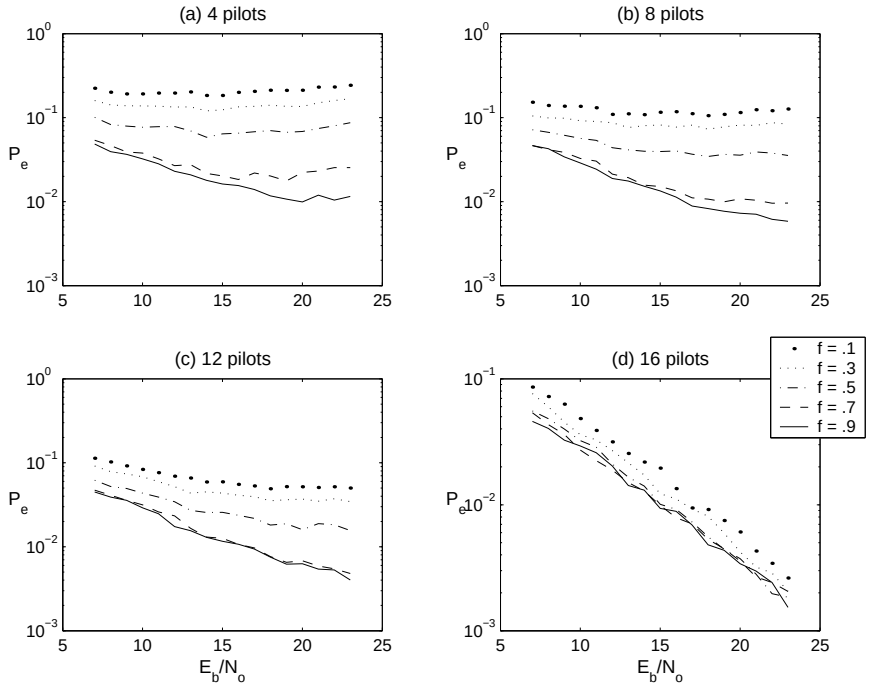

Figure 4.2: BER curves for the FB-Kalman based receiver for different number of pilots and various degrees of time variation

4.5.3 Comparing the Kalman and the Forward-Backward Kalman

In this subsection, we compare the effectiveness of the Kalman and the FB-Kalman when using the two configurations $16xxxx$ and $xx16xx$ for $x = 4, 8, 12, 16$ and for different degrees of time variation. Specifically, Figure 4.3 considers the $16xxxx$ pilot case. We note that almost consistently, the BER curve of the Kalman outperforms that of the FB-Kalman when the two receivers deal with the same degree of time variation (the only exception is the case $f = .7$ in which the FB-Kalman outperforms the Kalman filter).

Figure 4.4 considers the second pilot configuration, namely $xx16xx$. To avoid cluttering the figure, we draw the best BER curve for the Kalman filter case (corresponding to $f = .9$, the least degree of time variation). We note that this best case Kalman scenario is comparable to the worst FB-Kalman cases (those with high degrees of time variations). Thus, when the FB-Kalman receiver operates on channels with low degrees of time variation (e.g., for $f = .7$ or $f = .9$), it produces BER curves that are much better than that of the Kalman-based receiver.

We next compare the performance of the Kalman receiver employing the $16xxxx$ pilot

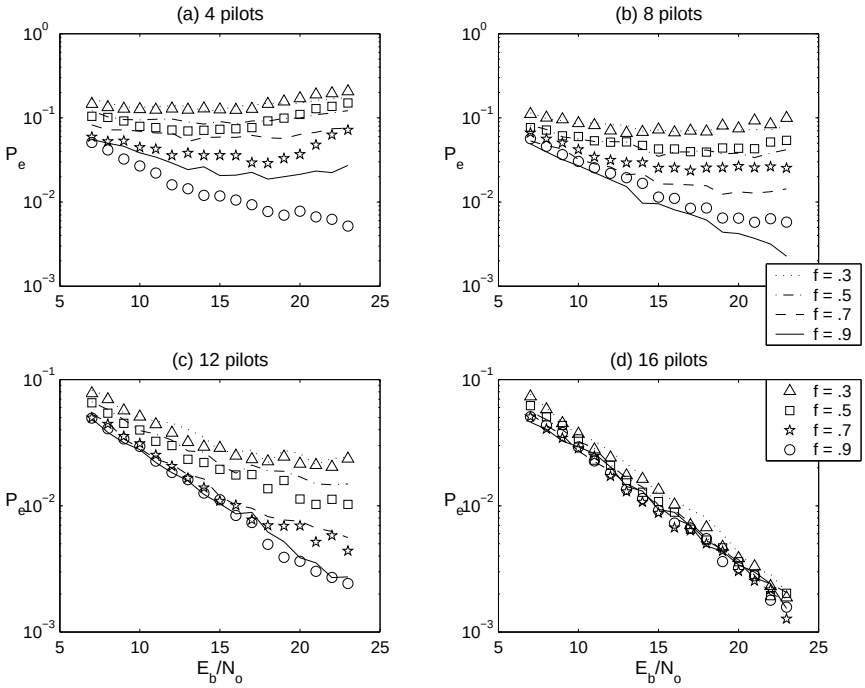

Figure 4.3: For the $16xxxx$ pilot configuration, the BER curves for the Kalman receiver outperform those of the FB-Kalman receiver

configuration with that of the FB-Kalman receiver employing the $xx16xx$ configuration. Thus, the two receivers employ the same number of pilots except that these pilots are distributed differently. We carry out this comparison for different levels of time variations. We note that the BER curves are quite comparable for the extreme cases of time variation (low and high values of f). However, for moderate levels of variation ($f = .7$), the FB-Kalman consistently outperforms the performance of the Kalman. This is not unexpected for when the variation is too slow, the two filters are equally able to track the channel with only a few pilots. When the time variation is too high, time correlation information becomes of little use. It is only at a moderate level of time variations that the additional signal processing mandated by the FB-Kalman becomes valuable.

4.5.4 Effect of Increased Signal Processing

We next consider the effect of increased signal processing on the BER curves for the Kalman and the FB-Kalman receivers. Specifically, we implement the two receivers

1. using the CP observation and the soft estimate of the input,

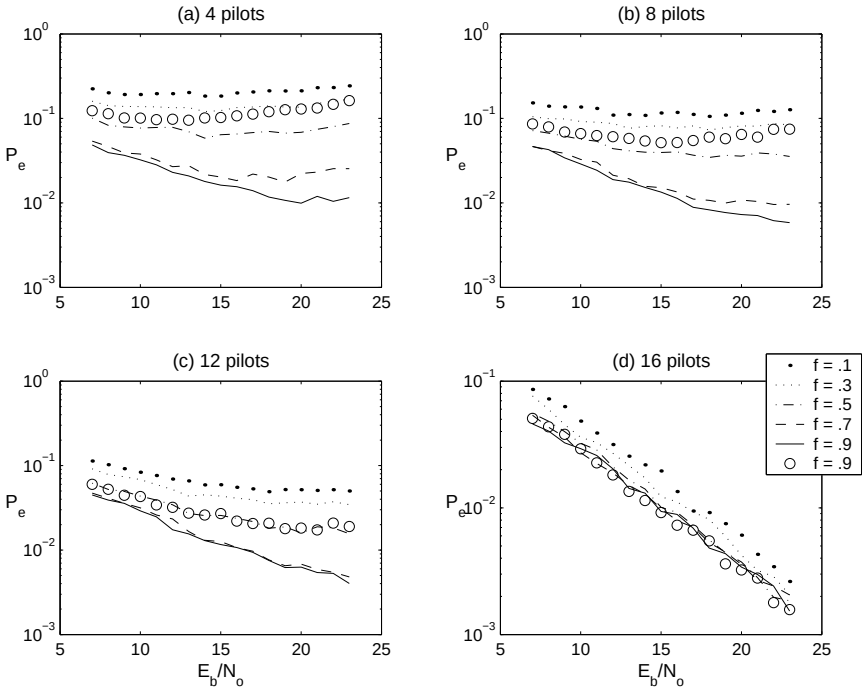

Figure 4.4: For the $xx16xx$ pilot configuration, the BER curves for the Kalman receiver outperform those of the FB-Kalman receiver

2. using the CP and the hard estimate of the input,

3. using no CP observation and using the hard estimate of the input.

Figures 4.6 and 4.7 show that increasing the level of signal processing pays off producing better BER performance. This applies for different number of pilots and different degrees of time variation.

4.5.5 Effect of Increasing the Number of Iterations

Figure 4.8 demonstrates the effect of increasing the number of EM iterations on the BER performance of the FB-Kalman receiver. We do that for different number of pilots and different degrees of time variation. As expected, the BER improves as we increase the number of iterations. The value of iterations eventually results in diminishing returns.

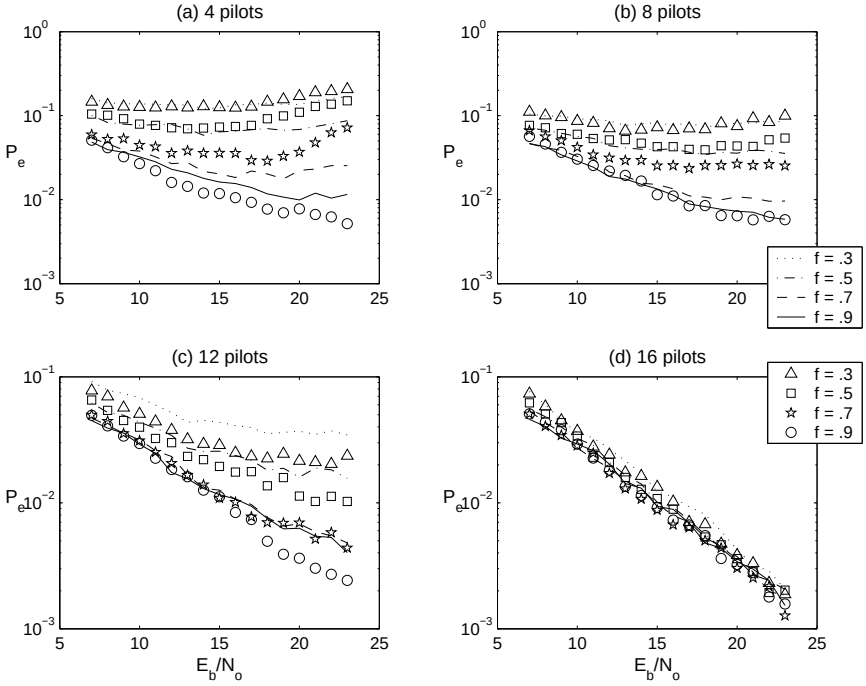

Figure 4.5: The Kalman and FB-Kalman show comparable performance at extreme levels of time variation. Only at moderate levels of variation does the FB-Kalman outperform the Kalman

4.5.6 Bench Marking

Finally, we bench mark the BER performance of the Kalman and FB-Kalman receivers against receivers that have been suggested in literature and also against the known-channel case. Specifically, Figure 4.9 compares the BER performance of the following five receivers:

1. EM-based least-squares (LS) receiver (i.e. a receiver employing frequency correlation only)
2. The EM-based receiver proposed by Lu, Wang, and Li in [58] [9].
3. The EM-based Kalman receiver
4. The EM-based FB-Kalman receiver
5. A receiver with perfect channel knowledge

[9] This receiver is similar to our Kalman-based receiver in that it makes use of the time and frequency correlation.

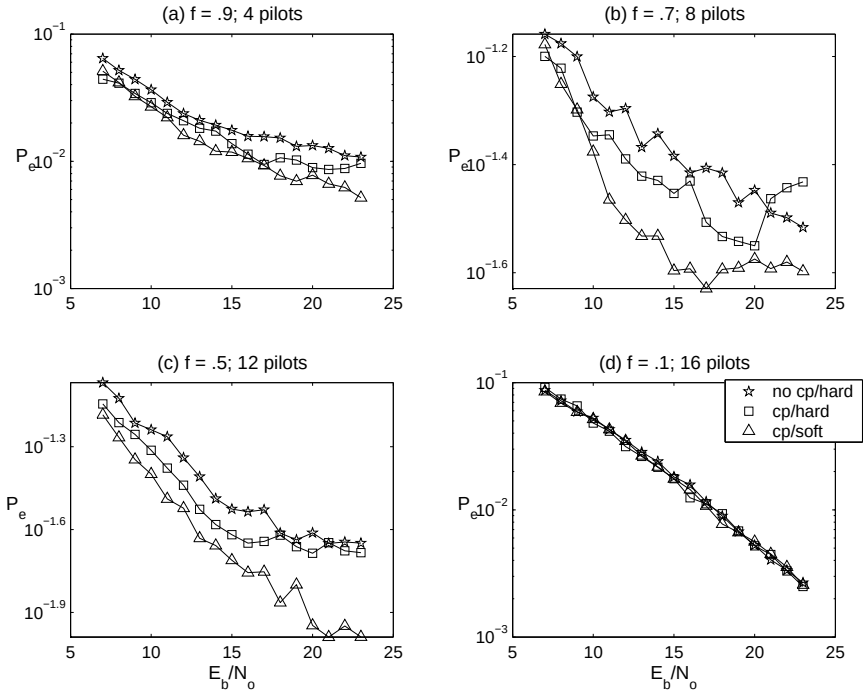

Figure 4.6: The Kalman-based receiver demonstrates improved BER with increasing levels of signal processing

All receivers implement the $16xxxx$ pilot configuration except the FB-Kalman which implements the $xx16xx$ configuration and the receiver with perfect channel knowledge which uses no pilots. We test these receivers against the dynamically variant channel (4.42) with $f = .7$.

Figure 4.9 demonstrates that the Kalman and FB-Kalman outperform the LS receiver and the receiver of [58]. This is especially the case for low number of pilots. Moreover, for this case of moderate time variation, the FB-Kalman consistently outperforms the Kalman receiver.

4.6 Conclusion

In this chapter, we considered the problem of semi-blind channel and data recovery in OFDM. The chapter first introduced some convenient notation and used that to perform a careful study of OFDM transmission. This study was subsequently used to design an OFDM receiver. The receiver makes use of the data and channel constraints to perform recovery

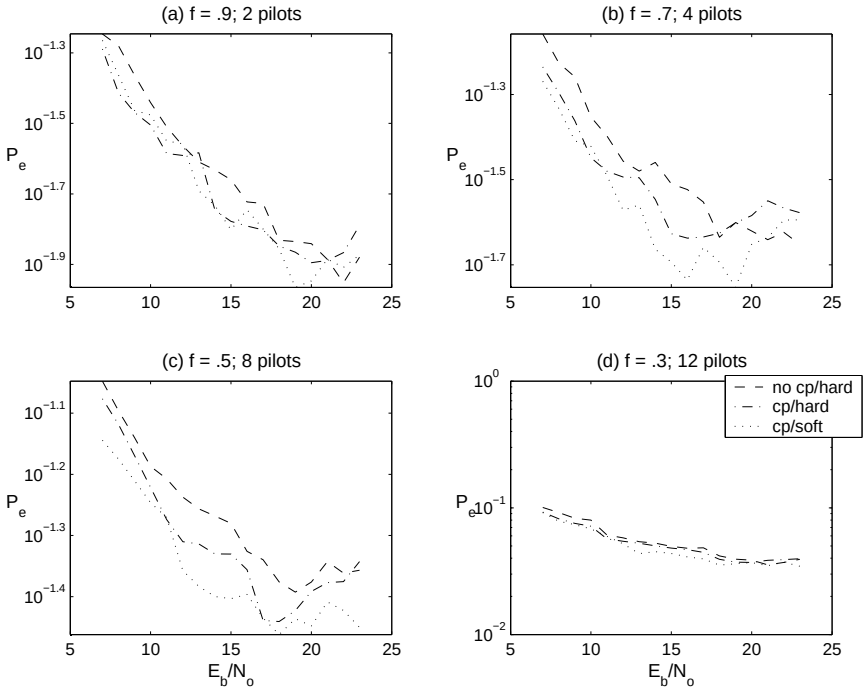

Figure 4.7: The FB-Kalman based receiver demonstrates improved BER with increasing levels of signal processing

with zero latency and minimal pilot overhead. Specifically, the receiver uses the pilots to kick start channel estimation and subsequently iterates between that and data recovery. In doing so, the receiver utilizes the data constraints (which include pilots, the cyclic prefix, and the finite alphabet nature of the data) and employs the data estimates in soft format. The receiver also makes use of the various constraints on the channel (which include sparsity and finite delay spread information as well as time and frequency correlation). Thanks to the presence of the cyclic prefix, optimal mean-square data recovery is done on an element by element basis while channel recovery always boils down to solving a regularized least-squares problem. Table 4.7 lists the various channel constraints and the associated LS norm that each contributes. Channel estimation using a subset of these constraints is performed by minimizing the sum of the corresponding squared norms.

The chapter culminates with the EM-based Kalman which performs data and channel recovery with zero latency and the EM-based forward-backward Kalman which performs batch processing to enhance the performance of the receiver (at the expense of increasing complexity, delay, and latency).

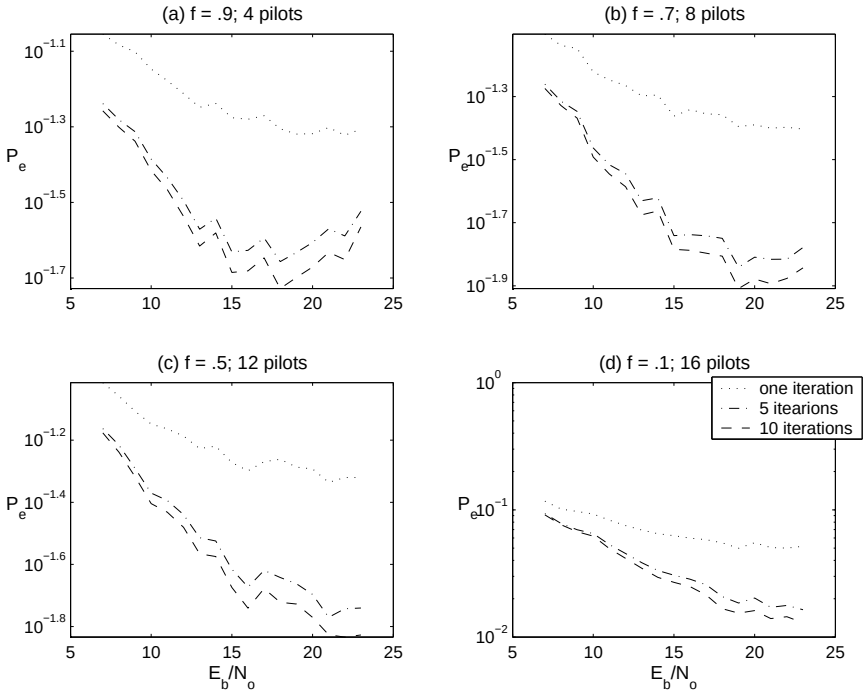

Figure 4.8: Increasing the number of EM iterations improves the BER of the FB-Kalman receiver, but the value of these iterations results in diminishing returns

The Kalman receivers derived here apply to nonstationary channels as well in which the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ of the state-space model (4.44) vary with time. It has already been extended to incorporate the coding constraint on the data [10]. The chapter assumes that these parameters are known perfectly at the receiver. However, the receiver can be generalized to estimate the state-space parameters and to be robust to uncertainties in these estimates (e.g., see [81]).

Table 4.7: *Weighted norm interpretation of various channel estimation methods*

CONSTRAINT	ASSOCIATED WEIGHTED NORM
Pilots	$\|\boldsymbol{\mathcal{Y}}_{i_{I_\mathsf{p}}} - \mathrm{diag}\,(\boldsymbol{\mathcal{X}}_i)_{I_\mathsf{p}}\,\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}}$
Frequency correlation	$\|\underline{\boldsymbol{h}}_i - \boldsymbol{m}\|^2_{\Pi^{-1}}$
Detected data	$\|\boldsymbol{\mathcal{Y}}_i - \mathrm{diag}(\hat{\boldsymbol{\mathcal{X}}}_i)\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}\tilde{Q}^*_{P+1}\mathrm{Cov}[\boldsymbol{\mathcal{X}}_i]\tilde{Q}_{P+1}}$
Cyclic prefix observation	$\|\underline{\boldsymbol{y}}_i - E[\underline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}\mathrm{Cov}[\underline{X}^*_i]}$
Time correlation	$\|\underline{\boldsymbol{h}}_i - \boldsymbol{F}\hat{\underline{\boldsymbol{h}}}_{i-1}\|^2_{P^{-1}_{i\|i-1}} + \|\overline{\boldsymbol{\mathcal{Y}}}_i - E[\overline{\boldsymbol{X}}_i]\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{\boldsymbol{h}}_i\|^2_{\frac{1}{\sigma_n^2}\mathrm{Cov}[\overline{X}_i]}$

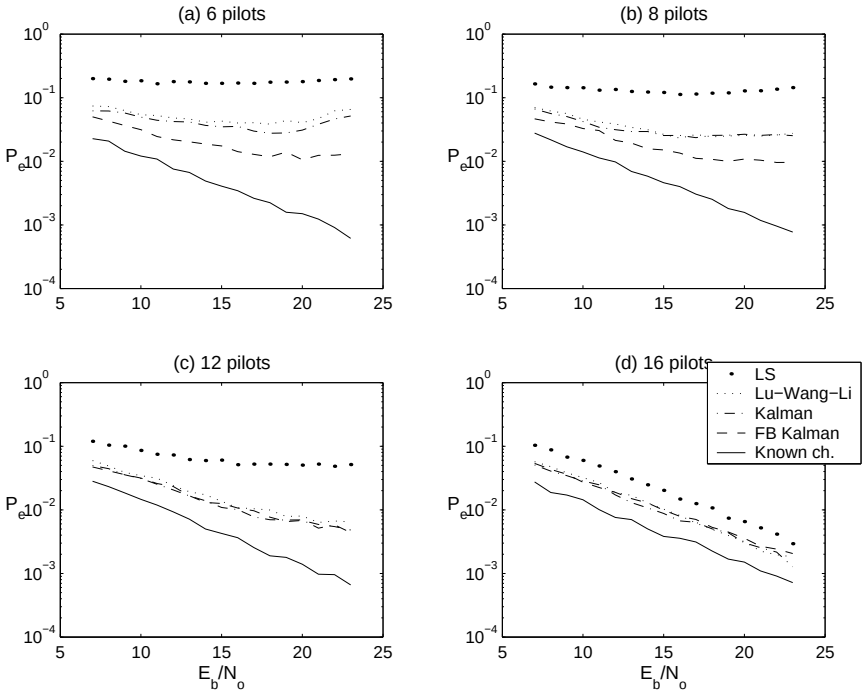

Figure 4.9: BER curves comparing a receiver that employs the Kalman filter with one employing the FB-Kalman and another with perfect channel knowledge. The two Kalman receivers employ the same number of pilots with optimum placement.

4.7 APPENDIX A: State-Space Channel Model

A typical OFDM symbol passes through two channels; the physical channel $\underline{\boldsymbol{c}}_i$ (which consists of $L+1$ paths arriving at instants $\tau_0, \tau_1 \ldots, \tau_L$) and the receive filter $\boldsymbol{r}$. The actual channel $\underline{\boldsymbol{h}}_i$, which is $P+1$ taps in length, is the convolution of these two channels. Thus, we can write

$$\begin{bmatrix} \underline{h}_i(1) \\ \underline{h}_i(2) \\ \vdots \\ \underline{h}_i(P+1) \end{bmatrix} = \begin{bmatrix} r(-\tau_0) & r(-\tau_1) & \cdots & r(-\tau_L) \\ r(T-\tau_0) & r(T-\tau_1) & \cdots & r(T-\tau_L) \\ \vdots & \vdots & \vdots & \vdots \\ r(PT-\tau_0) & r(PT-\tau_2) & \cdots & r(PT-\tau_L) \end{bmatrix} \begin{bmatrix} \underline{c}_i(1) \\ \underline{c}_i(2) \\ \vdots \\ \underline{c}_i(L+1) \end{bmatrix}$$

or more compactly

$$\underline{\boldsymbol{h}}_i = \boldsymbol{R}\underline{\boldsymbol{c}}_i \tag{4.57}$$

4.7.1 Modelling the Time-Variant Behavior

Due to the mobile nature of the channel, the physical channel taps $c_i(k)$ are time-variant. According to the WSSUS model [75], the process $c_i(k)$ is zero-mean wide-sense stationary complex Gaussian process with autocorrelation

$$E\left[c_i(k)c_{i'}(k')\right] = \mathcal{J}_0\left(2\pi f_c(k)(N+P)T|i-i'|\right)\delta_{kk'} \tag{4.58}$$

where T is the sampling (baud) rate, $f_c(k)$ is the Doppler frequency associated with the kth tap, and $\mathcal{J}_0$ denotes the zero-order Bessel function of the first kind.

We can approximate the time-variant behavior of the tap $c_i(k)$ arbitrarily closely by an autoregressive (AR) model with large enough order (see [50], [45]). As argued in [50], even a first-order AR model can capture most of the channel dynamics. We can show that the closest 1st-order AR fit of (4.58) is given by

$$\underline{c}_{i+1}(k) = \alpha(k)\underline{c}_i(k) + \sqrt{(1-\alpha^2(k))E[|\underline{c}_0(k)|^2]}u_i(k) \tag{4.59}$$

where

$$\alpha(k) \triangleq \mathcal{J}_0\left(2\pi f_c(k)(N+P)T\right)$$

The factor $\sqrt{(1-\alpha^2(k))E[|\underline{c}_0(k)|^2]}$ ensures that the tap $c_i(k)$ maintains the same power profile for all time. Collecting (4.59) for all taps yields

$$\boldsymbol{c}_{i+1} = \boldsymbol{F}_c\boldsymbol{c}_i + \boldsymbol{G}_c\boldsymbol{u}_i \tag{4.60}$$

where

$$\begin{aligned}\boldsymbol{F}_c &= \text{diag}\left(\ \alpha(1),\ \cdots,\ \alpha(L+1)\ \right)\\ \boldsymbol{G}_c &= \text{diag}\left[\ \sqrt{(1-\alpha^2(1))E[|\underline{c}_i(1)|^2]}\ \cdots\ \sqrt{(1-\alpha^2(L+1))E[|\underline{c}_i(L+1)|^2]}\ \right]\end{aligned}$$

We can use this dynamical relationship along with (4.57) to derive a dynamical relationship for the impulse response $\underline{\boldsymbol{h}}_i$. Specifically, multiplying both sides of (4.60) by $\boldsymbol{R}$ and noting that $\boldsymbol{R}^\dagger\boldsymbol{R} = \boldsymbol{I}$ [10], we obtain

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i$$

[10] For this to be true, the matrix $\boldsymbol{R}$ has to be tall which will be the case if the sampling rate is high enough so that the number of channel taps $P+1$ is larger then the number of physical paths.

where

$$\boxed{\boldsymbol{F} = \boldsymbol{R}\boldsymbol{F}_c\boldsymbol{R}^\dagger \quad \text{and} \quad \boldsymbol{G} = \boldsymbol{R}\boldsymbol{G}_c}$$

4.8 APPENDIX B: Evaluating Moments of the Input Matrix $\overline{\boldsymbol{X}}_i$

The expectation step boils down to calculating the first two moments $E[\overline{\boldsymbol{X}}_i]$ and $\mathrm{Cov}[\overline{\boldsymbol{X}}_i^*]$ (see (4.39) and (4.41) for example). The expressions (4.36) and (4.37) call for evaluating these two moments over $\boldsymbol{\mathcal{X}}_i$ only. However, since $\overline{\boldsymbol{X}_i}$ depends on $\boldsymbol{\mathcal{X}}_i$ as well as $\boldsymbol{\mathcal{X}}_{i-1}$, we will carry the expectation over these two consecutive symbols. This is needed for the EM-based forward-backward Kalman while the expectation over $\boldsymbol{\mathcal{X}}_i$ follows as a special case.

From (4.19) and (4.25), we can express the first moment of $\overline{\boldsymbol{X}}_i$ as

$$E_{\mathcal{X}_{i-1},\mathcal{X}_i}\left[\overline{\boldsymbol{X}}_i\right] \triangleq \begin{bmatrix} E_{\mathcal{X}_i}[\mathrm{diag}(\boldsymbol{\mathcal{X}}_i)]\tilde{\boldsymbol{Q}}_{P+1} \\ E_{\mathcal{X}_{i-1}}\left[\underline{\boldsymbol{X}}_{\mathsf{U}i}\right] + E_{\mathcal{X}_i}\left[\underline{\boldsymbol{X}}_{\mathsf{L}i}\right] \end{bmatrix} \tag{4.61}$$

$$= \begin{bmatrix} \mathrm{diag}(\hat{\boldsymbol{\mathcal{X}}}_i)\tilde{\boldsymbol{Q}}_{P+1} \\ \underline{\hat{\boldsymbol{X}}}_{\mathsf{U}i} + \underline{\hat{\boldsymbol{X}}}_{\mathsf{L}i} \end{bmatrix} \tag{4.62}$$

where the elements of $\hat{\boldsymbol{\mathcal{X}}}_i$ are evaluated in (4.34) and where the estimate $\underline{\hat{\boldsymbol{X}}}_{\mathsf{U}i}$ is obtained from $\underline{\hat{\boldsymbol{x}}}_{i-1}$ just as $\underline{\boldsymbol{X}}_{\mathsf{U}i}$ is obtained from $\underline{\boldsymbol{x}}_{i-1}$ (see (4.20)). The estimate $\underline{\hat{\boldsymbol{X}}}_{\mathsf{L}i}$ can be calculated similarly. Now, let's evaluate the covariance of $\overline{\boldsymbol{X}}_i^*$. Starting from the defining expression (4.25), it is easy to show that

$$\begin{aligned} \mathrm{Cov}_{\mathcal{X}_{i-1},\mathcal{X}_i}[\overline{\boldsymbol{X}}_i^*] &= \mathrm{Cov}_{\mathcal{X}_i}[\tilde{\boldsymbol{Q}}_{P+1}^*\mathrm{diag}(\boldsymbol{\mathcal{X}}_i)^*] + \mathrm{Cov}_{\mathcal{X}_{i-1},\mathcal{X}_i}[\underline{\boldsymbol{X}}_i^*] \\ &= \tilde{\boldsymbol{Q}}_{P+1}^*\mathrm{Cov}[\mathrm{diag}(\boldsymbol{\mathcal{X}}_i)^*]\tilde{\boldsymbol{Q}}_{P+1} + \mathrm{Cov}[\underline{\boldsymbol{X}}_{\mathsf{U}i}^*] + \mathrm{Cov}[\underline{\boldsymbol{X}}_{\mathsf{L}i}^*] \end{aligned}$$

The covariance $\mathrm{Cov}\left[\mathrm{diag}(\boldsymbol{\mathcal{X}}_i^*)\right]$ is a diagonal matrix whose diagonal elements are simply the variances

$$\mathrm{Cov}[\mathcal{X}_i(l)] = E[|\mathcal{X}_i(l)|^2] - |E[\mathcal{X}_i(l)]|^2 \tag{4.63}$$

and hence can be calculated from (4.34) and (4.35). The elements of the covariance matrix

$\boldsymbol{C}_{\mathsf{L}} \triangleq \mathrm{Cov}\,[\underline{\boldsymbol{X}}_{i\mathsf{L}}^{*}]$ are calculated recursively from

$$C_{\mathsf{L}}(j,k) = C_{\mathsf{L}}(j+1,k+1) + \underbrace{E[\underline{x}_i^*(P-j)\underline{x}_i(P-k)] - \underline{\hat{x}}_i^*(P-j)\underline{\hat{x}}_i(P-k)}_{\text{covariance evaluated in (4.67)}} \tag{4.64}$$

The recursion is run (backward) for $j,k = 1,2,\cdots,P$ starting from the boundary conditions

$$C_{\mathsf{L}}(P+1,l) = C_{\mathsf{L}}(l,P+1) = 0, \quad l = 1,2,\cdots,P \tag{4.65}$$

The covariances that appear in (4.64) are the entries of the covariance $\mathrm{Cov}\,[\underline{\boldsymbol{x}}_i]$ and are collectively calculated from

$$\begin{aligned} \mathrm{Cov}\,[\underline{\boldsymbol{x}}_i] &= \frac{1}{N}\underline{\boldsymbol{Q}}_P^{*}\mathrm{Cov}\,[\boldsymbol{\mathcal{X}}_i]\,\underline{\boldsymbol{Q}}_P & (4.66)\\ &= \frac{1}{N}\underline{\boldsymbol{Q}}_P^{*}\mathrm{Cov}\,[\mathrm{diag}(\boldsymbol{\mathcal{X}}_i)]\,\underline{\boldsymbol{Q}}_P & (4.67)\end{aligned}$$

where (4.66) follows from the partial IDFT relationship $\underline{\boldsymbol{x}}_i = (1/\sqrt{N})\underline{\boldsymbol{Q}}_P^{*}\boldsymbol{\mathcal{X}}_i$, and where the diagonal elements of the covariance $\mathrm{Cov}\,[\mathrm{diag}(\boldsymbol{\mathcal{X}}_i)]$ of (4.67) have already been calculated in (4.63). Similarly, we can show that the elements of the covariance $\boldsymbol{C}_{\mathsf{U}} \triangleq \mathrm{Cov}\,[\underline{\boldsymbol{X}}_{i\mathsf{U}}^{*}]$ satisfy the recursion

$$C_{\mathsf{U}}(j+1,k+1) = C_{\mathsf{U}}(j,k) + E[\underline{x}_{i-1}^*(P-j)\underline{x}_{i-1}(P-k)] - \underline{\hat{x}}_{i-1}^*(P-j)\underline{\hat{x}}_{i-1}(P-k)$$

The recursion is kick-started from the initial conditions

$$C_{\mathsf{U}}(1,l) = C_{\mathsf{U}}(l,1) = 0, \quad l = 1,2,\cdots,P \tag{4.68}$$

4.9 APPENDIX C: Derivation of the EM-Based Kalman Filters

In this appendix, we construct the EM-based Kalman filter. In particular, assume that the channel satisfies the state-space recursion

$$\underline{\boldsymbol{h}}_i = \boldsymbol{F}\underline{\boldsymbol{h}}_{i-1} + \boldsymbol{G}\boldsymbol{u}_i \tag{4.69}$$

with Gaussian distributed initial state $\underline{\boldsymbol{h}}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Pi}_0)$. The channel also satisfies the input/output relationship

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \overline{\boldsymbol{X}}_i \underline{\boldsymbol{h}}_i + \overline{\boldsymbol{\mathcal{N}}}_i \tag{4.70}$$

We obtain the MAP estimate of the IR sequence $\{\underline{\boldsymbol{h}}_k\}_{k=0}^T$ by maximizing the pdf of the sequence conditioned on the input and output sequences $\{\boldsymbol{\mathcal{X}}_k\}_{k=0}^T$ and $\{\boldsymbol{\mathcal{Y}}_k\}_{k=0}^T$. Alternatively, we obtain the MAP estimate by maximizing the full pdf $p\left(\{\underline{\boldsymbol{h}}_k\}_{k=0}^T, \{\boldsymbol{\mathcal{X}}_k\}_{k=0}^T, \{\boldsymbol{\mathcal{Y}}_k\}_{k=0}^T\right)$. It is straightforward to show that this pdf can be decomposed into

$$p\left(\{\underline{\boldsymbol{h}}_k\}_{k=0}^T, \{\overline{\boldsymbol{X}}_k\}_{k=0}^T, \{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=0}^T\right) = \prod_{k=0}^{T} p(\overline{\boldsymbol{\mathcal{Y}}}_k | \overline{\boldsymbol{X}}_k, \underline{\boldsymbol{h}}_k) \prod_{k=1}^{T} p(\underline{\boldsymbol{h}}_k | \underline{\boldsymbol{h}}_{k-1}) p(\underline{\boldsymbol{h}}_0)$$

Using the state-space equations, we can show that the log-likelihood function (excluding any terms that are independent of $\underline{\boldsymbol{h}}_k$) is give by

$$\begin{aligned} \ln p\left(\{\underline{\boldsymbol{h}}_k\}_{k=1}^T, \{\overline{\boldsymbol{X}}_k\}_{k=1}^T, \{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=1}^T\right) = -\|\underline{\boldsymbol{h}}_0\|^2_{\boldsymbol{\Pi}_0^{-1}} \quad &- \quad \sum_{k=1}^{T} \|\overline{\boldsymbol{\mathcal{Y}}}_k - \overline{\boldsymbol{X}}_k \underline{\boldsymbol{h}}_k\|^2_{\frac{1}{\sigma_n^2}} \\ &- \quad \sum_{k=1}^{T} \|\underline{\boldsymbol{h}}_k - \boldsymbol{F}\underline{\boldsymbol{h}}_{k-1}\|^2_{\frac{1}{\sigma_u^2} GG^*} \end{aligned} \tag{4.71}$$

Since the channel sequence $\{\underline{\boldsymbol{h}}_k\}_{k=0}^T$ is Gaussian distributed, the MAP estimate of the channel sequence given the output sequence $\{\overline{\boldsymbol{\mathcal{Y}}}_k\}_{k=0}^T$ is the same as the MMSE estimate given the same sequence.The MMSE estimate itself is obtained by the forward-backward filter described by (4.46)–(4.50), (4.54), and (4.55) (See problem 10.9 in [47]).

Now since the input symbols $\boldsymbol{\mathcal{X}}_0, \cdots, \boldsymbol{\mathcal{X}}_T$, are not available, we invoke the EM algorithm, maximizing the *averaged* likelihood instead. Specifically, given the initial estimates $\hat{\underline{\boldsymbol{h}}}_0, \cdots, \hat{\underline{\boldsymbol{h}}}_T$ together with the output symbols $\boldsymbol{\mathcal{Y}}_0, \cdots, \boldsymbol{\mathcal{Y}}_T$, we can estimate $\underline{\boldsymbol{h}}_0, \cdots, \underline{\boldsymbol{h}}_T$ iteratively using the expectation-maximization operation

$$\{\hat{\underline{\boldsymbol{h}}}_k^{(j+1)}\}_{k=0}^T = \arg \max_{\{\underline{h}_k\}_{k=0}^T} E_{\{\mathcal{X}_k\}_{k=0}^T | \{\hat{\underline{h}}_k^{(j)}\}_{k=0}^T, \{\mathcal{Y}_k\}_{k=0}^T} p(\{\underline{\boldsymbol{h}}_k\}_{k=0}^T, \{\boldsymbol{\mathcal{X}}_k\}_{k=0}^T, \{\boldsymbol{\mathcal{Y}}_k\}_{k=0}^T)$$

This, together with (4.71), yields

$$\begin{aligned}
\{\hat{\underline{\boldsymbol{h}}}_k^{(j+1)}\}_{k=0}^{T} &= \arg \max_{\{\underline{h}_k\}_{k=0}^{T}} -\|\underline{\boldsymbol{h}}_0\|^2_{\boldsymbol{\Pi}_0^{-1}} - \sum_{k=0}^{T} E\left\|\boldsymbol{\mathcal{Y}}_k - \overline{\boldsymbol{X}}_k \underline{\boldsymbol{h}}_k\right\|^2_{\frac{1}{2\sigma_n^2}} - \sum_{k=0}^{T} \|\boldsymbol{u}_k\|^2_{\frac{1}{2\sigma_u^2}} \\
= \arg \max_{\{\underline{h}_k\}_{k=0}^{T}} &-\|\underline{\boldsymbol{h}}_0\|^2_{\boldsymbol{\Pi}_0^{-1}} - \sum_{k=0}^{T} \left\|\boldsymbol{\mathcal{Y}}_k - E\left[\overline{\boldsymbol{X}}_k\right] \underline{\boldsymbol{h}}_k\right\|^2_{\frac{1}{2\sigma_n^2}} - \sum_{k=0}^{T} \|\underline{\boldsymbol{h}}_k\|^2_{\frac{1}{2\sigma_n^2}\mathrm{Cov}[\overline{X}_k^*]} - \sum_{k=0}^{T} \|\boldsymbol{u}_k\|^2_{\frac{1}{2\sigma_u^2}} \\
= &-\|\underline{\boldsymbol{h}}_0\|^2_{\boldsymbol{\Pi}_0^{-1}} - \sum_{k=0}^{T} \left\| \begin{bmatrix} \boldsymbol{\mathcal{Y}}_k \\ \mathbf{0}_{P\times 1} \end{bmatrix} - \begin{bmatrix} E[\overline{\boldsymbol{X}}_k] \\ \mathrm{Cov}[\overline{\boldsymbol{X}}_k^*]^{1/2} \end{bmatrix} \underline{\boldsymbol{h}}_k \right\|^2_{\frac{1}{2\sigma_n^2}} - \sum_{k=0}^{T} \|\boldsymbol{u}_k\|^2_{\frac{1}{2\sigma_u^2}} \qquad (4.72)
\end{aligned}$$

where the expectations are taken given the previous estimates $\hat{\underline{\boldsymbol{h}}}_k^{(j)}$ and the output of all data symbols $\boldsymbol{\mathcal{Y}}_0, \cdots, \boldsymbol{\mathcal{Y}}_T$. The averaged likelihood (4.72) function can be obtained from the original one (4.71) by performing the substitutions

$$\begin{aligned}
\overline{\boldsymbol{X}}_l &\longrightarrow \begin{bmatrix} E[\overline{\boldsymbol{X}}_l] \\ \mathrm{Cov}[\overline{\boldsymbol{X}}_l^*]^{1/2} \end{bmatrix} \\
\overline{\boldsymbol{\mathcal{Y}}}_l &\longrightarrow \begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_l \\ \mathbf{0}_{P\times 1} \end{bmatrix}
\end{aligned}$$

Thus, the maximizing impulse response values are also obtained by implementing the forward-backward Kalman filter taking into account the substitutions above. Alternatively, the impulse response values are obtained by implementing the forward-backward Kalman on the following state-space model

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i \qquad (4.73)$$

$$\begin{bmatrix} \overline{\boldsymbol{\mathcal{Y}}}_i \\ \mathbf{0}_{P\times 1} \end{bmatrix} = \begin{bmatrix} E[\overline{\boldsymbol{X}}_i] \\ \mathrm{Cov}[\overline{\boldsymbol{X}}_i^*]^{1/2} \end{bmatrix} \underline{\boldsymbol{h}}_i + \begin{bmatrix} \overline{\boldsymbol{\mathcal{N}}}_i \\ \underline{\boldsymbol{n}}_i \end{bmatrix} \qquad (4.74)$$

where $\boldsymbol{n}_i$ is virtual noise that is independent of the physical noise $\overline{\boldsymbol{\mathcal{N}}}_i$. This state-space model in turn motivates the EM-Kalman filter. In other words, the EM-Kalman based receiver applies the Kalman filter to the state-space model (4.73)–(4.74).

Chapter 5

Receiver Design for MIMO OFDM Transmission over Time-Variant Channels

5.1 Introduction

[1] This chapter builds on the previous chapter and scales up the receiver design to OFDM transmission over mutli-input multi-output (MIMO) transmission. OFDM is a technique that enables high speed transmission over frequency selective channels with simple equalizers by creating a set of parallel, frequency-flat channels. Moreover, for frequency flat fading channels, space-time codes provide diversity and coding gain benefits when compared with single-input single-output (SISO) systems, improving their BER performance [73]. When MIMO techniques are combined with OFDM, space-time codes for frequency flat channels can be used per tone, providing the benefit of multiple antennas with simple channel equalization. However, the spacial dimension in turn places a constraint in that the receiver has to recover the impulse response of the MIMO channel.

This chapter thus considers receiver design for MIMO OFDM transmission over frequency selective time-variant channels. While the presence of multi-antennas makes the problem

[1] A major part of this chapter is reproduced, with permission, from T. Y. Al-Naffouri O. Oteri, O. Awoniyi, and A. Paulraj, "Receiver design for MIMO-OFDM transmission over time-variant channels," Globecom 2004, Dallas, Texas, Nov. 2004.

more challenging, it also adds more (spatial) structure that the receiver can utilize. Specifically, in addition to the constraints we employed for SISO OFDM, the receiver can make use of the transmit and receive correlation as well as the space-time (ST) code. The chapter is organized along the same lines of Chapter 4. Specifically, following this introduction, we provide a system overview in Section 5.2 where we describe the transmitter, the receiver, and the channel model. In Section 5.3 we derive the I/O equations that are used for channel estimation and for space-time decoding. The channel estimator and data detector parts of the receiver are described in Section 5.4 and this is followed by some practical considerations and simulation results.

5.1.1 Notation

We continue to use the notation of Chapter 4. However, we need to adopt more notational elements to take care of the additional spatial and temporal dimensions. Thus, given a sequence of vectors $\underline{\boldsymbol{h}}_{r_x}^{t_x}$ for $r_x = 1 \cdots R_x$ and $t_x = 1 \cdots T_x$, we define the following stack variables

$$\underline{\boldsymbol{h}}_{r_x} = \begin{bmatrix} \underline{\boldsymbol{h}}_{r_x}^{1} \\ \vdots \\ \underline{\boldsymbol{h}}_{r_x}^{T_x} \end{bmatrix} \quad \text{and} \quad \underline{\boldsymbol{h}} = \begin{bmatrix} \underline{\boldsymbol{h}}_{1} \\ \vdots \\ \underline{\boldsymbol{h}}_{R_x} \end{bmatrix} \tag{5.1}$$

Moreover, given an $R_x \times T_x$ matrix $\boldsymbol{H}$, we denote its (r_x, t_x) entry by $h_{r_x}^{t_x}$, its r_x row by $\boldsymbol{h}_{r_x}$, and its t_x column by $\boldsymbol{h}^{t_x}$. Thus, we can write

$$\begin{aligned} \boldsymbol{H} &= [h_{r_x}^{t_x}] \\ &= \begin{bmatrix} \boldsymbol{h}^1 & \boldsymbol{h}^2 & \cdots & \boldsymbol{h}^{T_x} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_{R_x} \end{bmatrix} \end{aligned}$$

We also use the notation $\boldsymbol{h}^{(+)}$ ($\boldsymbol{h}^{(-)}$) to denote the value of $\boldsymbol{h}$ at the next (previous) time instant.

5.2 System Overview

In this section, we give an overview of the communications system: transmitter, channel, and receiver.

5.2.1 Transmitter

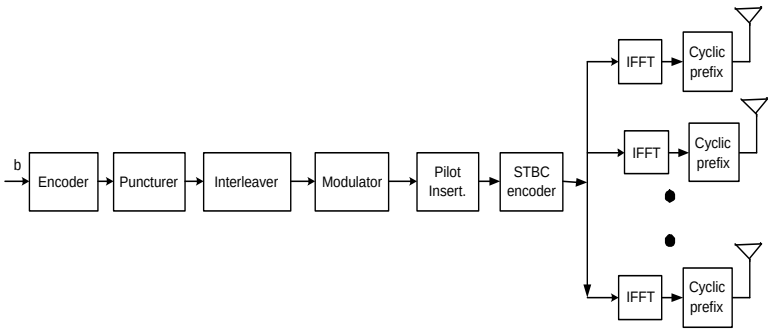

Figure 5.1: Transmitter

A block diagram of the transmitter is shown in Figure 5.1. The bit sequence to be transmitted passes through a convolutional encoder that serves as an outer code for the system. The coded output is then punctured to increase the code rate. The punctured sequence then passes through a random interleaver which rearranges the order of the bits according to a random permutation. The interleaved bit sequence is mapped to QAM symbols using gray coding and the QAM symbols are in turn mapped to the OFDM symbols with space reserved for the pilot symbols (as explained in Subsection 5.2.2). The STBC encoder uses the OFDM signals to construct the ST block by mapping the various OFDM symbols to a specific antenna and specific time slot depending on the ST code used. Each antenna performs an IFFT operation on the OFDM symbols to produce the time-domain OFDM symbols and adds a cyclic prefix to each prior to transmission.

5.2.2 Pilot Insertion

Pilots are employed to initialize channel estimation. The use of properly placed pilots and additional channel or data constraints can reduce the actual number of pilots needed. The first ST block in the transmitted packet uses N_{p} pilots. Subsequent ST blocks can use

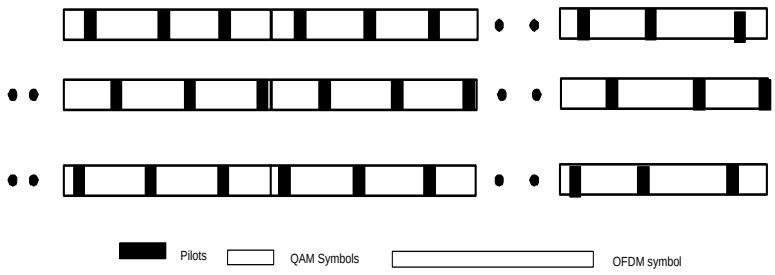

Figure 5.2: Pilot placement in OFDM symbols

a reduced number of pilots or no pilots at all (e.g., as in [24], [58]), relying instead on additional channel constraints (e.g., time correlation) to perform initial channel estimation.

The pilots are optimally placed at maximum distance away from each other [71]. Within the same STBC block, the pilots maintain the same position (across time and space) which prevents interference from non-pilot symbols. As such, no interference cancellation is needed during initial channel estimation. Since frequency bins close to the pilot positions have a higher estimation accuracy than those further away, the pilot positions are rotated from one ST block to the next. Figure 5.2 demonstrates these design guidelines.

5.2.3 Channel Model

In this chapter, we consider MIMO channels that are block fading and frequency selective. For proper channel modelling, consider the time domain I/O relationship

$$\boldsymbol{y}(i) = \sum_{p=0}^{P} \boldsymbol{H}(p)\boldsymbol{x}(i-p) + \boldsymbol{n}(i) \tag{5.2}$$

where $\boldsymbol{H}(p)$ is the $R_x \times T_x$ MIMO impulse response (IR) at tap p and where i is the sample time index. The taps $\boldsymbol{H}(p)$ usually incorporate the effect of the transmit filter and the effects of the transmit and receive spatial correlation making $\boldsymbol{H}(p)$ correlated across space and tap. We will assume $\boldsymbol{H}(p)$ here to be iid and relegate the spatially correlated case to Appendix A. Moreover, we will assume that $\boldsymbol{H}(p)$ remains constant over a single ST block (and hence over the constituent OFDM symbols).

To model the time variations, we scale up the SISO channel model of Chapter 4 to the MIMO case. Specifically, we assume that from one ST block to the next, the MIMO taps

change according to the dynamic equation

$$\boldsymbol{H}^{(+)}(p) = \alpha(p)\boldsymbol{H}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}\boldsymbol{U}(p) \tag{5.3}$$

where $\alpha(p)$ is related to the Doppler frequency $f_D(p)$ by $\alpha(p) = J_0(2\pi f_D(p)T)$ and T is the time duration of one ST block, and where $\boldsymbol{U}(p)$ is an iid matrix with entries that are $\mathcal{N}(0,1)$. The variable β in (5.3) corresponds to the exponent of the channel delay profile while the factor $\sqrt{(1-\alpha^2(p))e^{-\beta p}}$ ensures that each link maintains the exponential decay profile ($e^{-\beta p}$) for all time.

This channel model pushes the time variation to the limit while avoiding intercarrier interference and ensuring the proper operation of the space-time code. Using the dynamic equation in (5.3), we can obtain the state-space model for the impulse response $\underline{\boldsymbol{h}}_{r_x}^{t_x}$ acting between transmit antenna t_x and receive antenna r_x

$$\underline{h}_{r_x}^{t_x(+)}(p) = \alpha(p)\underline{h}_{r_x}^{t_x}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}u_{r_x}^{t_x}(p) \tag{5.4}$$

By stacking (5.4) for $p = 0, 1, \ldots, P$, we obtain the dynamic model

$$\underline{\boldsymbol{h}}_{r_x}^{t_x^{(+)}} = \boldsymbol{F}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \boldsymbol{G}\boldsymbol{u}_{r_x}^{t_x} \tag{5.5}$$

where

$$\boldsymbol{F} = \begin{bmatrix} \alpha(0) & & \\ & \ddots & \\ & & \alpha(P) \end{bmatrix} \text{ and } \boldsymbol{G} = \begin{bmatrix} \sqrt{1-\alpha^2(0)} & & \\ & \ddots & \\ & & \sqrt{(1-\alpha^2(P))e^{-\beta P}} \end{bmatrix}$$

By further stacking (5.5) over all transmit and receive antennas, as done in (5.1), we obtain

$$\underline{\boldsymbol{h}}^{(+)} = (\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{F})\,\underline{\boldsymbol{h}} + (\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{G})\,\boldsymbol{u} \tag{5.6}$$

where $\underline{\boldsymbol{h}}$, $\boldsymbol{u}$, and $\underline{\boldsymbol{h}}^{(+)}$ are vectors of size $T_xR_x(P+1) \times 1$. Note that while (5.3) and (5.6) are equivalent, the latter model is in vector form and hence lends itself more to the Kalman filter operations, which are essential for channel estimation.

Finally, for a complete characterization of this dynamic model, we need to specify the

covariance of $\boldsymbol{u}$. It is easy to show that

$$E\left[\boldsymbol{u}\boldsymbol{u}^*\right] = \boldsymbol{I}_{R_x} \otimes E\left[\boldsymbol{u}_{r_x}\boldsymbol{u}_{r_x}^*\right] \tag{5.7}$$

$$= \boldsymbol{I}_{R_x} \otimes \left(\boldsymbol{I}_{T_x} \otimes E\left[u_{r_x}^{t_x} u_{r_x}^{t_x *}\right]\right) \tag{5.8}$$

$$= \boldsymbol{I}_{R_x} \otimes \boldsymbol{I}_{T_x} \otimes \boldsymbol{I}_{P+1} = \boldsymbol{I}_{T_x R_x (P+1)} \tag{5.9}$$

We can similarly show that the channel covariance at the first time instant is given by

$$E\left[\underline{\boldsymbol{h}}\,\underline{\boldsymbol{h}}^*\right] = \boldsymbol{I}_{T_x R_x} \otimes \boldsymbol{G}\boldsymbol{G}^*$$

The covariance information is important for employing the Kalman filter to channel estimation. Appendix A considers the the more general spatially correlated case.

5.2.4 Receiver

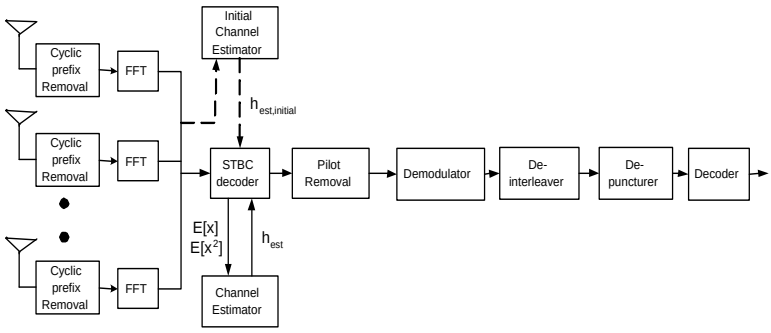

Figure 5.3: OSTBC OFDM receiver

A block diagram of the receiver is shown in Figure 5.3. The receiver uses the pilots to initialize its operation. The receiver's core operation is based on the expectation maximization (EM) algorithm which performs joint channel and data recovery. The iterative module is made up of the ST block decoder/data detector and the channel estimator.

STBC Decoder/Data Detector

The STBC decoder/data detector calculates the conditional first and second moments of the transmitted data (soft estimate) to be used by the channel estimator. This constitutes

the expectation step of the EM algorithm. At the last iteration of the algorithm, this block generates hard decisions of the STBC block.

Channel Estimator

The soft estimates of the STBC decoder are used by the channel estimator in the maximization step of the EM algorithm. It uses these estimates together with other data/channel constraints to produce an improved channel estimate.

These two processes (channel estimation and data detection) go on iteratively until a stopping criterion is satisfied. This could be that the iterative algorithm executes a maximum number of iterations or that the likelihood function does not change beyond a certain threshold.

The decoded OSTBC OFDM symbols are stripped of their pilot symbols, passing the remaining (data) symbols to the QAM demodulator. The demodulated bits subsequently pass through the de-puncturer, de-interleaver, and finally the Viterbi decoder.

5.3 Input/Output Equations for MIMO OFDM

To derive the I/O equations for a MIMO channel, we first derive the I/O equation for a SISO link between transmit antenna t_x and receive antenna r_x and subsequently use superposition to scale up the result to the MIMO case. The SISO I/O equations are described in Table 4.2 (Chapter 4), part of which is reproduced here for convenience.

Table 5.1: *Input/output relationships for the circular, linear, and total channels between transmit antenna t_x and receive antenna r_x*

CHANNEL	SEQUENCE RELATIONSHIP	MATRIX RELATIONSHIP
Linear	$\underline{y}_{r_x} = \underline{h}_{r_x}^{t_x} * \underline{x}_{t_x} + \underline{n}_{r_x}$	$\underline{\boldsymbol{y}}_{r_x} = \underline{\boldsymbol{X}}_{t_x}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x}$
Circular	$\boldsymbol{y}_{r_x} = \boldsymbol{h}_{r_x}^{t_x} \circledast \boldsymbol{x}_{t_x} + \boldsymbol{n}_{r_x}$ $\boldsymbol{\mathcal{Y}}_{r_x} = \boldsymbol{\mathcal{H}}_{r_x}^{t_x} \odot \boldsymbol{\mathcal{X}}_{t_x} + \boldsymbol{\mathcal{N}}_{r_x}$	$\boldsymbol{\mathcal{Y}}_{r_x} = \operatorname{diag}(\boldsymbol{\mathcal{X}}_{r_x})\,\tilde{\boldsymbol{Q}}_{P+1}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \boldsymbol{\mathcal{N}}_{r_x}$ $= \boldsymbol{X}_{r_x}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \boldsymbol{\mathcal{N}}_{r_x}$
Total	$\overline{y}_{r_x} = \overline{h}_{r_x}^{t_x} * \overline{x}_{t_x} + \overline{n}_{r_x}$	$\overline{\boldsymbol{\mathcal{Y}}}_{r_x} = \overline{\boldsymbol{X}}_{t_x}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \overline{\boldsymbol{\mathcal{N}}}_{r_x}$

5.3.1 Remarks

1. We can now use the I/O equations of Table 5.1, together with the superposition principle, to derive the corresponding I/O equations for a MIMO system. For example,

under superposition, the matrix relationship for the linear channel becomes

$$\underline{\boldsymbol{y}}_{r_x} = \sum_{t_x=1}^{T_x} \underline{\boldsymbol{X}}_{t_x} \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x} \tag{5.10}$$

2. Alternatively, if we adopt the Einstein notation for summation, [2] the following two expressions become equivalent

$$\underline{\boldsymbol{y}}_{r_x} = \underline{\boldsymbol{X}}_{t_x} \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x} \quad \text{and} \quad \underline{\boldsymbol{y}}_{r_x} = \sum_{t_x=1}^{T_x} \underline{\boldsymbol{X}}_{t_x} \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x} \tag{5.11}$$

This in effect means that Table 5.1, as it stands, completely characterizes the I/O equations for the MIMO channel.

3. The relationships of Table 5.1 exhibit no time dependence as they apply to any MIMO OFDM symbol. When time dependence needs to be emphasized, as when the OFDM symbol is part of a space-time block, we signify it by attaching (t_b) to the symbol. For example, we can rewrite (5.10) to stress the time-dependence as

$$\underline{\boldsymbol{y}}_{r_x}(t_b) = \sum_{t_x=1}^{T_x} \underline{\boldsymbol{X}}_{t_x}(t_b) \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x}(t_b) \tag{5.12}$$

For proper operation of the ST code, we assume that the channel $\underline{\boldsymbol{h}}_{r_x}^{t_x}$ remains constant over the duration of the block.

4. While Table 5.1 completely describes the behavior of a MIMO channel, we would like to write the I/O equation in the succinct form

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{A}\underline{\boldsymbol{h}} + \boldsymbol{\mathcal{N}}, \tag{5.13}$$

that incorporates the effect of the space-time code and the effect on all receive antennas– a form that lends itself to channel estimation. Before doing that, however, we digress to briefly introduce ST codes and input representation in their presence.

[2] The Einstein notation allows us to get rid of the summation symbol by invoking the understanding that summation runs over any subscript or superscript that is repeated in a given monomial. Thus, t_x appears twice in the monomial $\underline{h}_{r_x}^{t_x} * \underline{x}_{t_x}$ and so the monomial should be summed over the range of t_x making the two expressions in (5.11) equivalent.

5.3.2 Space-Time Coding for OFDM

For OFDM transmission, STBC can be implemented on the time or frequency symbol. In time-domain STBC, the code symbols of a given ST block occupy the same frequency bins of consecutive OFDM symbols. In contrast, the code symbols in frequency-domain STBC occupy consecutive frequency bins in the same OFDM symbol. While we assume time-domain STBC in this chapter, the design can be easily extended to frequency-domain STBC.

To derive the I/O equations for space-time coding, we adopt the approach of [51]. To this end, consider the set of N_s uncoded OFDM symbols $\{\boldsymbol{\mathcal{S}}(1), \ldots, \boldsymbol{\mathcal{S}}(N_s)\}$. Using ST coding, we wish to transmit these symbols in one OSTBC block using T_x antennas and T_b time slots. We achieve this using the set of $T_x \times T_b$ matrices $\{\boldsymbol{A}(1), \boldsymbol{B}(1), \ldots, \boldsymbol{A}(N_s), \boldsymbol{B}(N_s)\}$ which characterize the ST code used. For example, Alamouti's code with $T_x = 2, N_s = 2$ and $T_b = 2$ is characterized by the matrices

$$\boldsymbol{A}(1) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \boldsymbol{A}(2) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{B}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{B}(2) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Following the approach of [51], we can show that the OFDM symbol transmitted from antenna t_x at time slot t_b is given by

$$\boldsymbol{\mathcal{X}}_{t_x}(t_b) = \sum_{n_s=1}^{N_s} a_{t_x}^{t_b}(n_s)\mathcal{R}\boldsymbol{\mathcal{S}}(n_s) + j b_{t_x}^{t_b}(n_s)\mathcal{I}\boldsymbol{\mathcal{S}}(n_s) \tag{5.14}$$

where $a_{t_x}^{t_b}$ is the (t_x, t_b) element of $\boldsymbol{A}(n_s)$, $b_{t_x}^{t_b}$ is the (t_x, t_b) element of $\boldsymbol{B}(n_s)$, $j = \sqrt{-1}$, and $\mathcal{R}\boldsymbol{\mathcal{S}}(n_s)$ is the real part of $\boldsymbol{\mathcal{S}}(n_s)$ and $\mathcal{I}\boldsymbol{\mathcal{S}}(n_s)$ its imaginary part.

The coded symbol $\boldsymbol{\mathcal{X}}_{t_x}(t_b)$ is now ready for OFDM transmission. This is done at the transmitter by performing an IFFT on $\boldsymbol{\mathcal{X}}_{t_x}(t_b)$ to produce the time domain symbol

$$\boldsymbol{x}_{t_x}(t_b) = \frac{1}{\sqrt{N}}\boldsymbol{Q}^*\boldsymbol{\mathcal{X}}_{t_x}(t_b) \tag{5.15}$$

and by appending a cyclic prefix given by

$$\underline{\boldsymbol{x}}_{t_x}(t_b) = \frac{1}{\sqrt{N}}\underline{\boldsymbol{Q}}_P^*\boldsymbol{\mathcal{X}}_{t_x}(t_b) \tag{5.16}$$

The time and frequency domain symbols as well as the cyclic prefix can in turn be used to

generate the input matrices $\underline{\boldsymbol{X}}_{t_x}(t_b)$, $\boldsymbol{X}_{t_x}(t_b)$, and $\overline{\boldsymbol{X}}_{t_x}(t_b)$ that are used in the I/O equations as described in Chapter 4. With these equations at hand, we are now ready to write the I/O equations in the compact form (5.13).

5.3.3 Input/Output Equations with Space-Time Coding: Channel Estimation Version

Linear Channel

Consider the time-dependent I/O equation (5.12). Concatenating this equation for $t_b = 1, \ldots, T_b$ yields

$$\underbrace{\begin{bmatrix} \underline{\boldsymbol{y}}_{r_x}(1) \\ \underline{\boldsymbol{y}}_{r_x}(2) \\ \vdots \\ \underline{\boldsymbol{y}}_{r_x}(T_b) \end{bmatrix}}_{\underline{\boldsymbol{y}}_{r_x}} = \underbrace{\begin{bmatrix} \underline{\boldsymbol{X}}_{t_x}(1) \\ \underline{\boldsymbol{X}}_{t_x}(2) \\ \vdots \\ \underline{\boldsymbol{X}}_{t_x}(T_b) \end{bmatrix}}_{\underline{\boldsymbol{X}}_{t_x}} \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underbrace{\begin{bmatrix} \underline{\boldsymbol{n}}_{r_x}(1) \\ \underline{\boldsymbol{n}}_{r_x}(2) \\ \vdots \\ \underline{\boldsymbol{n}}_{r_x}(T_b) \end{bmatrix}}_{\underline{\boldsymbol{n}}_{r_x}}$$

We can write this more compactly as

$$\underline{\boldsymbol{y}}_{r_x} = \underline{\boldsymbol{X}}_{t_x} \underline{\boldsymbol{h}}_{r_x}^{t_x} + \underline{\boldsymbol{n}}_{r_x}$$

This represents the I/O relationship for the linear channel between transmit antenna t_x and receive antenna r_x. Using superposition, we can express the effect of T_x such antennas as

$$\begin{aligned} \underline{\boldsymbol{y}}_{r_x} &= \underbrace{\begin{bmatrix} \underline{\boldsymbol{X}}_1 & \cdots & \underline{\boldsymbol{X}}_{T_x} \end{bmatrix}}_{\underline{\boldsymbol{X}}} \underbrace{\begin{bmatrix} \underline{\boldsymbol{h}}_{r_x}^1 \\ \vdots \\ \underline{\boldsymbol{h}}_{r_x}^{T_x} \end{bmatrix}}_{\underline{\boldsymbol{h}}_{r_x}} + \boldsymbol{\mathcal{N}}_{r_x} && (5.17) \\ &= \underline{\boldsymbol{X}}\, \underline{\boldsymbol{h}}_{r_x} + \underline{\boldsymbol{n}}_{r_x} && (5.18) \end{aligned}$$

Finally, concatenating this relationship for all receive antennas yields

$$\begin{bmatrix} \underline{\boldsymbol{y}}_1 \\ \underline{\boldsymbol{y}}_2 \\ \vdots \\ \underline{\boldsymbol{y}}_{R_x} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{X}} & & & \\ & \underline{\boldsymbol{X}} & & \\ & & \ddots & \\ & & & \underline{\boldsymbol{X}} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{h}}_1 \\ \underline{\boldsymbol{h}}_2 \\ \vdots \\ \underline{\boldsymbol{h}}_{R_x} \end{bmatrix} + \begin{bmatrix} \underline{\boldsymbol{n}}_1 \\ \underline{\boldsymbol{n}}_2 \\ \vdots \\ \underline{\boldsymbol{n}}_{R_x} \end{bmatrix}$$

Using the stack notation (5.1), we can equivalently write this as

$$\boxed{\underline{\boldsymbol{y}} = (\boldsymbol{I} \otimes \underline{\boldsymbol{X}})\,\underline{\boldsymbol{h}} + \underline{\boldsymbol{n}}}$$

Circular Channel

Along the same lines, we derive the following I/O equation for the circular channel

$$\boxed{\boldsymbol{\mathcal{Y}} = (\boldsymbol{I} \otimes \boldsymbol{X})\,\underline{\boldsymbol{h}} + \boldsymbol{n}} \tag{5.19}$$

where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots \boldsymbol{X}_{T_x} \end{bmatrix}$$

with

$$\boldsymbol{X}_{t_x} = \begin{bmatrix} \boldsymbol{X}_{t_x}(1) \\ \boldsymbol{X}_{t_x}(2) \\ \vdots \\ \boldsymbol{X}_{t_x}(T_b) \end{bmatrix} = \begin{bmatrix} \mathrm{diag}(\boldsymbol{\mathcal{X}}_{t_x}(1))\tilde{\boldsymbol{Q}}_{P+1} \\ \mathrm{diag}(\boldsymbol{\mathcal{X}}_{t_x}(2))\tilde{\boldsymbol{Q}}_{P+1} \\ \vdots \\ \mathrm{diag}(\boldsymbol{\mathcal{X}}_{t_x}(T_b))\tilde{\boldsymbol{Q}}_{P+1} \end{bmatrix}$$

Total Channel

We can similarly show that the I/O equation governing the total channel is given by

$$\boxed{\overline{\boldsymbol{y}} = \left(\boldsymbol{I} \otimes \overline{\boldsymbol{X}}\right)\underline{\boldsymbol{h}} + \overline{\boldsymbol{n}}}$$

where

$$\overline{\boldsymbol{X}} = \begin{bmatrix} \overline{\boldsymbol{X}}_1 & \overline{\boldsymbol{X}}_2 & \cdots \overline{\boldsymbol{X}}_{T_x} \end{bmatrix}$$

with

$$\overline{\boldsymbol{X}}_{t_x} = \begin{bmatrix} \overline{\boldsymbol{X}}_{t_x}(1) \\ \overline{\boldsymbol{X}}_{t_x}(2) \\ \vdots \\ \overline{\boldsymbol{X}}_{t_x}(T_b) \end{bmatrix} \quad \text{and} \quad \overline{\boldsymbol{X}}_{t_x}(t_b) = \begin{bmatrix} \boldsymbol{X}_{t_x}(t_b) \\ \underline{\boldsymbol{X}}_{t_x}(t_b) \end{bmatrix} = \begin{bmatrix} \mathrm{diag}(\boldsymbol{\mathcal{X}}_{t_x}(t_b))\tilde{\boldsymbol{Q}}_{P+1} \\ \underline{\boldsymbol{X}}_{t_x}(t_b) \end{bmatrix}$$

5.3.4 Input/Output Equations with Space-Time Coding: Data Detection Version

Signal detection in ST-coded OFDM is done on a tone-by-tone basis, except that the tones are collected for the whole ST block (for R_x receive antennas and over T_b time slots). Consider the frequency domain I/O equation in Table 5.1. We can extract the following I/O equation for tone n belonging to the OFDM symbol t_b (equations (5.20)–(5.26) are valid at a given tone n but we don't show this dependence explicitly for brevity)

$$\mathcal{Y}_{r_x}(t_b) = \begin{bmatrix} \mathcal{H}_{r_x}^1 & \cdots & \mathcal{H}_{r_x}^{T_x} \end{bmatrix} \begin{bmatrix} \mathcal{X}_1(t_b) \\ \vdots \\ \mathcal{X}_{T_x}(t_b) \end{bmatrix} + \mathcal{N}_{r_x}(t_b) \tag{5.20}$$

Collecting this relationship for all receive antennas yields

$$\begin{bmatrix} \mathcal{Y}_1(t_b) \\ \vdots \\ \mathcal{Y}_{R_x}(t_b) \end{bmatrix} = \begin{bmatrix} \mathcal{H}_1^1 & \cdots & \mathcal{H}_1^{T_x} \\ \vdots & \cdots & \vdots \\ \mathcal{H}_{R_x}^1 & \cdots & \mathcal{H}_{R_x}^{T_x} \end{bmatrix} \begin{bmatrix} \mathcal{X}_1(t_b) \\ \vdots \\ \mathcal{X}_{T_x}(t_b) \end{bmatrix} + \begin{bmatrix} \mathcal{N}_1(t_b) \\ \vdots \\ \mathcal{N}_{R_x}(t_b) \end{bmatrix} \tag{5.21}$$

Or, more succinctly,

$$\boldsymbol{\mathcal{Y}}(t_b) = \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{X}}(t_b) + \boldsymbol{\mathcal{N}}(t_b) \tag{5.22}$$

By further concatenating this relationship for $t_b = 1, \cdots, T_b$, we can show that the following relationship holds (see [51] for more details)

$$\boxed{\boldsymbol{\mathcal{Y}} = \boldsymbol{C} \begin{bmatrix} \mathcal{RS} \\ \mathcal{IS} \end{bmatrix} + \boldsymbol{\mathcal{N}}} \tag{5.23}$$

where

$$\boldsymbol{\mathcal{Y}} = \begin{bmatrix} \boldsymbol{\mathcal{Y}}(1) \\ \vdots \\ \boldsymbol{\mathcal{Y}}(T_b) \end{bmatrix}, \quad \boldsymbol{\mathcal{S}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}(1) \\ \vdots \\ \boldsymbol{\mathcal{S}}(N_s) \end{bmatrix}, \quad \text{and} \quad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_a & \boldsymbol{C}_b \end{bmatrix}$$

with

$$\boldsymbol{C}_a = \left[\begin{array}{ccc} \text{vec}(\boldsymbol{\mathcal{H}}\boldsymbol{A}(1)) & \cdots & \text{vec}(\boldsymbol{\mathcal{H}}\boldsymbol{A}(N_s)) \end{array} \right] \tag{5.24}$$

$$\boldsymbol{C}_b = \left[\begin{array}{ccc} \text{vec}(\boldsymbol{\mathcal{H}}\boldsymbol{B}(1)) & \cdots & \text{vec}(\boldsymbol{\mathcal{H}}\boldsymbol{B}(N_s)) \end{array} \right] \tag{5.25}$$

We note that the STBC code is orthogonal if and only if the matrix $\boldsymbol{C}$ satisfies [51],

$$\mathcal{R}\left[\boldsymbol{C}^*\boldsymbol{C}\right] = ||\boldsymbol{\mathcal{H}}||^2\boldsymbol{I} \quad \forall \boldsymbol{\mathcal{H}} \tag{5.26}$$

This property is essential to perform data detection. On multiplying both sides of (5.23) by $\boldsymbol{C}^* = \left[\begin{array}{cc} \boldsymbol{C}_a^* & \boldsymbol{C}_b^* \end{array} \right]$, taking the real part, and rearranging terms, we can show that the following relationship holds

$$\boxed{\widetilde{\boldsymbol{\mathcal{Y}}}(n) = \|\boldsymbol{\mathcal{H}}(n)\|^2\boldsymbol{\mathcal{S}}(n) + \widetilde{\boldsymbol{\mathcal{N}}(n)}} \tag{5.27}$$

where

$$\widetilde{\boldsymbol{\mathcal{Y}}}(n) = \mathcal{R}\left[\boldsymbol{C}_a^*(n)\boldsymbol{C}(n)\boldsymbol{\mathcal{Y}}(n)\right] + j\mathcal{R}\left[\boldsymbol{C}_b^*(n)\boldsymbol{C}(n)\boldsymbol{\mathcal{Y}}(n)\right] \tag{5.28}$$

$$\widetilde{\boldsymbol{\mathcal{N}}}(n) = \mathcal{R}\left[\boldsymbol{C}_a^*(n)\boldsymbol{C}(n)\boldsymbol{\mathcal{N}}(n)\right] + j\mathcal{R}\left[\boldsymbol{C}_b^*(n)\boldsymbol{C}(n)\boldsymbol{\mathcal{N}}(n)\right] \tag{5.29}$$

We would like to remind the reader that the developments above apply at a given frequency tone n as explicitly indicated in (5.27)–(5.29). Since $\boldsymbol{C}$ is orthogonal, the noise $\widetilde{\boldsymbol{\mathcal{N}}}$ remains white and $\boldsymbol{\mathcal{S}}$ can be detected from (5.27) on an element-by-element basis.

5.4 Channel Estimation

Channel estimation is a critical part of the receiver proposed in this chapter. It is a challenging task because the receiver needs to estimate the channel by utilizing the underlying channel and data constraints when the input is not available at the receiver. The estimation makes use of the frequency and time correlation, finite delay spread, sparsity, ST code, cyclic prefix, and the finite alphabet constraints. We start this section by explaining how to estimate the channel when the data is known at the receiver. We use that as a spring board to treat the unobserved data case. Throughout this section, we will assume that the

channel $\underline{\boldsymbol{h}}_d$ satisfies the generic I/O equation

$$\boldsymbol{\mathcal{Y}}_d = \boldsymbol{X}_d \underline{\boldsymbol{h}}_d + \boldsymbol{\mathcal{N}}_d \tag{5.30}$$

and follows the dynamic model

$$\underline{\boldsymbol{h}}_d^{(+)} = \boldsymbol{F}_d \underline{\boldsymbol{h}}_d + \boldsymbol{G}_d \boldsymbol{u}_d \tag{5.31}$$

where the subscript d indicates dummy variables.[3]

5.4.1 Known Data Case

When the input $\boldsymbol{X}_d$ is available, we perform channel estimation by maximizing the log-likelihood function

$$\hat{\underline{\boldsymbol{h}}}_d = \arg\max_{\underline{\boldsymbol{h}}_d} p(\underline{\boldsymbol{h}}_d | \boldsymbol{\mathcal{Y}}_d, \boldsymbol{X}_d) \tag{5.32}$$

$$= \arg\max_{\underline{\boldsymbol{h}}_d} p(\underline{\boldsymbol{h}}_d) p(\boldsymbol{\mathcal{Y}}_d, \boldsymbol{X}_d | \underline{\boldsymbol{h}}_d) \tag{5.33}$$

Here $p(\underline{\boldsymbol{h}}_d | \boldsymbol{\mathcal{Y}}_d, \boldsymbol{X}_d)$ is the pdf of the channel given the input and output data. Assuming that the channel is Gaussian distributed ($\underline{\boldsymbol{h}}_d = \mathcal{N}(0, \boldsymbol{\Pi})$) and satisfies the I/O equation (5.30), we can show that the MAP estimate is given by

$$\hat{\underline{\boldsymbol{h}}}_d = \arg\min_{\underline{\boldsymbol{h}}_d} \|\boldsymbol{\mathcal{Y}}_d - \boldsymbol{X}_d \underline{\boldsymbol{h}}_d\|^2_{\sigma_n^{-2}} + \|\underline{\boldsymbol{h}}_d\|^2_{\boldsymbol{\Pi}^{-1}} \tag{5.34}$$

where σ_n^2 is the variance of the noise. The estimate makes use of the *frequency correlation* which manifests itself through the channel covariance matrix $\boldsymbol{\Pi}$.

If, in addition to the I/O equation, the channel satisfies the dynamic model (5.31), then we can use the previous channel estimate to improve on the current estimate. More precisely, the dynamic dependence between the present and the past expressed by (5.31) allows us to use all *past* input and output data in addition to the present ones. In this case, the log-likelihood function (5.32) is maximized given all the past and present data and is

[3] In this section, we describe channel estimation in terms of a generic state-space model and dummy variables. This allows us to describe channel estimation in general and succinct terms and without having to carry complicated expressions around (involving the kronecker product, for example).

achieved efficiently using the Kalman filter [47], described by the equations below

$$\boldsymbol{P}^{(+|-)} = \begin{cases} \boldsymbol{\Pi} & \text{for first time instant} \\ \boldsymbol{F}_d \boldsymbol{P}^{(-)} \boldsymbol{F}_d^* + \boldsymbol{G}_d \boldsymbol{G}_d^* & \text{otherwise} \end{cases} \tag{5.35}$$

$$\boldsymbol{R}_e = \sigma_n^2 \boldsymbol{I} + \boldsymbol{X}_d \boldsymbol{P}^{(+|-)} \boldsymbol{X}_d^* \tag{5.36}$$

$$\boldsymbol{K}_f = \boldsymbol{P}^{(+|-)} \boldsymbol{X}_d^* \boldsymbol{R}_e^{-1} \tag{5.37}$$

$$\hat{\underline{\boldsymbol{h}}}_d^{(+)} = \begin{cases} \boldsymbol{0} & \text{for the first time instant} \\ \left(\boldsymbol{I} - \boldsymbol{K}_f \boldsymbol{X}_d\right) \boldsymbol{F}_d \hat{\underline{\boldsymbol{h}}}_d + \boldsymbol{K}_f \boldsymbol{\mathcal{Y}}_d, & \text{otherwise} \end{cases} \tag{5.38}$$

$$\boldsymbol{P}^{(+)} = \boldsymbol{P}^{(+|-)} - \boldsymbol{K}_f \boldsymbol{R}_e \boldsymbol{K}_f^* \tag{5.39}$$

5.4.2 Unknown Data Case: The EM Algorithm

The challenge in our algorithm is that the input is not available. Hence, instead of maximizing the conditional distribution in (5.32), we maximize an *averaged* form of the distribution, i.e.

$$\hat{\underline{\boldsymbol{h}}}_d^{\text{new iter}} = \arg\max_{h_d} \; E_{X|\mathcal{Y}_d, \hat{h}_d^{\text{old iter}}} \left[\ln p(\underline{\boldsymbol{h}}_d | \boldsymbol{X}_d, \boldsymbol{\mathcal{Y}}_d)\right] \tag{5.40}$$

where averaging is performed over the unknown input given the output $\boldsymbol{\mathcal{Y}}_d$ and the channel estimate of the previous iteration. This represents the EM algorithm. Each iteration of the algorithm produces an estimate $\hat{\underline{\boldsymbol{h}}}_d$ that monotonically increases the likelihood of the channel $\underline{\boldsymbol{h}}_d$. This guarantees that the EM algorithm converges to a local maximum of the likelihood function [66]. Convergence to the global maximum depends on the initial condition from which the EM iterations are started.

From (5.40), we see that the EM algorithm consists of two steps repeated iteratively:

1. The expectation step where the log-likelihood is averaged over the unknown (input) variable given the most recent channel estimate. This corresponds to the data detection part of the receiver.

2. The maximization step where the averaged likelihood is maximized to update the channel estimate. This corresponds to the channel estimation part of the receiver.

These two steps are repeated until the algorithm converges.

For example, when the data $\boldsymbol{X}_d$ is unobserved, the MAP estimate of (5.34) is replaced

by the EM-based estimate

$$\hat{\underline{\boldsymbol{h}}}_d = \arg\max_{h_d} \; \|\boldsymbol{\mathcal{Y}}_d - E[\boldsymbol{X}_d]\underline{\boldsymbol{h}}_d\|^2_{\sigma_n^{-2}} + \|\underline{\boldsymbol{h}}_d\|^2_{\boldsymbol{\Pi}^{-1}} + \|\underline{\boldsymbol{h}}_d\|^2_{\mathrm{Cov}[X_d^*]} \tag{5.41}$$

where $E[\boldsymbol{X}_d]$ and $\mathrm{Cov}[\boldsymbol{X}_d]$ represent the expectation and covariance of $\boldsymbol{X}_d$, respectively.

Similarly, when the data $\boldsymbol{X}_d$ is unobserved, we can not employ the Kalman filter (5.35)–(5.39) to estimate the channel. Instead, the EM-based channel estimate is obtained by employing the Kalman filter (5.35)–(5.39) to the following state-space model [2]

$$\underline{\boldsymbol{h}}_d^{(+)} = \boldsymbol{F}_d\underline{\boldsymbol{h}}_d + \boldsymbol{G}_d\boldsymbol{u}_d \tag{5.42}$$

$$\begin{bmatrix} \boldsymbol{\mathcal{Y}}_d \\ \mathbf{0}_{P\times 1} \end{bmatrix} = \begin{bmatrix} E\left[\boldsymbol{X}_d\right] \\ \mathrm{Cov}\left[\boldsymbol{X}_d{}^*\right]^{1/2} \end{bmatrix} \underline{\boldsymbol{h}}_d + \begin{bmatrix} \boldsymbol{\mathcal{N}}_d \\ \underline{\boldsymbol{z}}_d \end{bmatrix} \tag{5.43}$$

where $\underline{\boldsymbol{z}}_d$ is Gaussian $\mathcal{N}(\mathbf{0}_{P\times 1}, \sigma_n^2\boldsymbol{I})$ and independent from $\boldsymbol{\mathcal{N}}_d$. In other words, we employ the Kalman filter (5.35)–(5.39) with the following change of variables

$$\boldsymbol{X}_d \longrightarrow \begin{bmatrix} E\left[\boldsymbol{X}_d\right] \\ \mathrm{Cov}\left[\boldsymbol{X}_d{}^*\right]^{1/2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{Y}}_d \longrightarrow \begin{bmatrix} \boldsymbol{\mathcal{Y}}_d \\ \mathbf{0}_{P\times 1} \end{bmatrix} \tag{5.44}$$

5.5 Algorithm Summary

In this section we summarize the steps taken in the algorithm

- **Initial channel estimation** The first step in the receiver operation is to obtain an initial estimate of the channel. We achieve this applying the Kalman filter to the dynamic channel model (5.6) together with the pilot/output equations (5.61), which are derived in the Appendix B

$$\underline{\boldsymbol{h}}^{(+)} = \left(\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{F}\right)\underline{\boldsymbol{h}} + \left(\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{G}\right)\boldsymbol{u} \tag{5.45}$$

$$\boldsymbol{\mathcal{Y}}_{I_\mathrm{p}} = (\boldsymbol{I} \otimes \boldsymbol{X}_{I_\mathrm{p}})\underline{\boldsymbol{h}} + \boldsymbol{\mathcal{N}}_{I_\mathrm{p}} \tag{5.46}$$

  The Kalman filter (5.35)–(5.39) thus provides the initial channel estimate by performing the substitution

$$\begin{aligned} \boldsymbol{F}_d &\longrightarrow (\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{F}) & \boldsymbol{G}_d &\longrightarrow (\boldsymbol{I}_{T_xR_x} \otimes \boldsymbol{G}) \\ \boldsymbol{\mathcal{Y}}_d &\longrightarrow \boldsymbol{\mathcal{Y}}_{I_\mathrm{p}} & \boldsymbol{X}_d &\longrightarrow \left(\boldsymbol{I}_{R_x} \otimes \boldsymbol{X}_{I_\mathrm{p}}\right) \end{aligned} \tag{5.47}$$

When time correlation information is not available to the receiver, the initial estimate can be obtained by solving the LS problem in (5.41) or by setting $\boldsymbol{F} = \mathbf{0}$ in (5.47). In subsequent ST blocks, the final estimate calculated in the previous block is used to calculate the predicted portion of the channel estimate.

- **Expectation step-data recovery** The receiver uses the latest channel estimate to perform the expectation step on the data. Let $A = \left\{A_1, \ldots, A_{|A|}\right\}$ where $|A|$ is the size of the set A, denote the alphabet set from which the elements of the (uncoded) OFDM symbol $\boldsymbol{\mathcal{S}}(n)$ take their values. Based on the data detection relationship in (5.27), we can show that the conditional pdf $f(\mathcal{S}(n_s)|\widetilde{\mathcal{Y}}(n_s))$ is given by ($\mathcal{S}(n_s)$ is the nth tone of the OFDM symbol $\boldsymbol{\mathcal{S}}(n_s)$ with n omitted for brevity)

$$f(S(n_s)|\widetilde{\mathcal{Y}}(n_s)) = \frac{e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 S(n_s)\right|^2}{2\sigma_n^2}}}{\sum_{i=1}^{|A|} e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 A_i\right|^2}{2\sigma_n^2}}} \tag{5.48}$$

where $\|\boldsymbol{\mathcal{H}}\|^2$ is based on the most recent channel estimate and is defined in (5.22). We can use this to calculate conditional expectation of $\mathcal{S}(n_s)$ and its second moment given the output $\widetilde{\mathcal{Y}}(n_s)$

$$E[\mathcal{S}(n_s)|\widetilde{\mathcal{Y}}(n_s)] = \frac{\sum_{i=1}^{|A|} A_i e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 A_i\right|^2}{2\sigma_n^2}}}{\sum_{i=1}^{|A|} e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 A_i\right|^2}{2\sigma_n^2}}} \tag{5.49}$$

$$E[|\mathcal{S}(n_s)|^2|\widetilde{\mathcal{Y}}(n_s)] = \frac{\sum_{i=1}^{|A|} |A_i|^2 e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 A_i\right|^2}{2\sigma_n^2}}}{\sum_{i=1}^{|A|} e^{-\frac{\left|\tilde{\mathcal{Y}}(n_s)-\left\|\boldsymbol{\mathcal{H}}\right\|^2 A_i\right|^2}{2\sigma_n^2}}} \tag{5.50}$$

- **Maximization step-channel estimation** The receiver now uses the first two moments of the data to perform the maximization step on the channel. As we argued in Subsection 5.4.2, the maximization step is carried out by running the Kalman filter

(5.35)–(5.39) with the following change of variables

$$\begin{aligned} \boldsymbol{F}_d &\longrightarrow (\boldsymbol{I}_{T_x R_x} \otimes \boldsymbol{F}) \qquad & \boldsymbol{G}_d &\longrightarrow (\boldsymbol{I}_{T_x R_x} \otimes \boldsymbol{G}_d) \\ \boldsymbol{\mathcal{Y}}_d &\longrightarrow \begin{bmatrix} \boldsymbol{\mathcal{Y}} \\ \mathbf{0}_{P\times 1} \end{bmatrix} & \boldsymbol{X} &\longrightarrow \begin{bmatrix} \boldsymbol{I}_{R_x} \otimes E\left[\boldsymbol{X}\right] \\ \boldsymbol{I}_{R_x} \otimes \mathrm{Cov}\left[\boldsymbol{X}^*\right]^{1/2} \otimes \boldsymbol{I}_{R_x} \end{bmatrix} \end{aligned} \tag{5.51}$$

When time correlation is unavailable, the channel estimate can be obtained by setting $\boldsymbol{F} = \mathbf{0}$ in (5.51).

- The expectation and maximization steps are alternated until a stopping criterion is satisfied, at which point the detected QAM-symbols are demodulated, de-punctured and de-interleaved. The resulting bits are then decoded by a Viterbi decoder.

5.6 Simulation Parameters

The transmitter and receiver illustrated in Figure 5.1 and Figure 5.3 were implemented. The outer encoder is a rate 1/2 convolutional encoder and the coded bits are mapped to 16-QAM symbols using gray coding. We use the OSTBC commonly known as the Alamouti code with $N_s = 2$ and $T_b = 2$ [6]. Our MIMO channel model is simulated using the state-space model with parameters, $\alpha = 0.985$, $\beta = 0.2$, $P = 7$ and $\boldsymbol{U}$ is $\mathcal{N}(0, I)$. The number of receive antennas, R_x, is set to 1 or 2.

Three thousand packets were simulated per SNR value. Each packet is comprised of 12 OFDM symbols transmitted over 6 ST blocks. Each OFDM symbol consists of 64 frequency tones and a cyclic prefix of length 16. 16 pilots are used in the OFDM symbols making up the first ST block, while the number of pilots in subsequent symbols vary between 2, 6, and 10.

5.7 Results and Discussion

In this section, we discuss the effect of various parameters on the BER performance of the receiver design.

5.7.1 Bench Marking

We compare our algorithm with an EM-based iterative MMSE receiver such as the one proposed in [58] and [24]. In contrast to our work, the authors in [58] and [24] take a

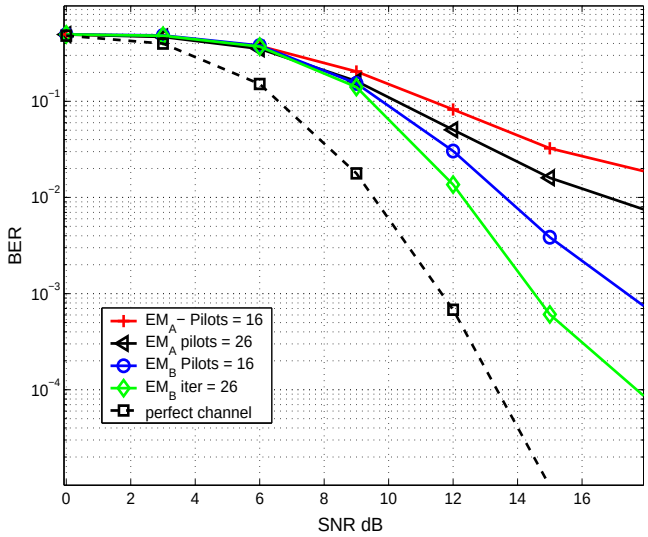

Figure 5.4: Receiver design comparison

data-centric approach, treating the transmitted signal as the desired parameter and the channel as the unobserved data. This algorithm further confines its pilots to the first ST block. The pilots are used to produce an initial channel estimate for the first ST block. This estimate is in turn used to predict the initial channel estimate for the subsequent ST blocks by employing a time correlation filter [58]. These initial estimates are used to kick-start the EM algorithm.

In this algorithm, the E-step is calculated by a conditional expectation of the channel given the received symbol and the current estimate of the transmitted data (i.e., through MMSE estimation). The maximization step is simply the hard decision, i.e. the ML estimate of the transmitted data.

In Figure 5.4, we compare both schemes with 16 pilots in the initial ST block and zero pilots in the subsequent blocks. EM_A refers to the iterative MMSE scheme while EM_B refers to the Kalman filter based scheme proposed in this chapter. We also implement both schemes with a total of 26 pilots as shown in Figure 5.4. The EM_A confines the pilots to the first ST block while in EM_B, we place 16 pilots in the first ST block and 2 pilots each in subsequent blocks. This ensures that both schemes incur the same pilot overhead.

Our algorithm (EM_B) outperforms EM_A of [58] in both pilot scenarios. One reason for this performance improvement is that our algorithm incorporates the time correlation information and the most recent channel estimate in every iteration of the EM algorithm.

5.7.2 Sensitivity to Number of Iterations and Spatial Diversity

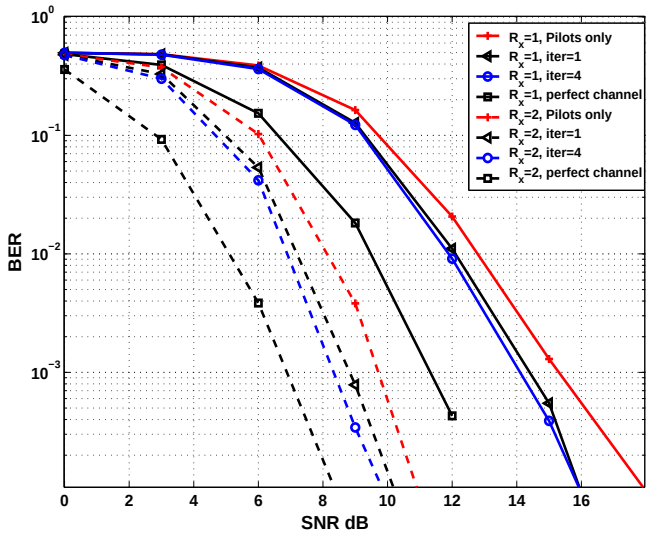

Figure 5.5: BER performance with iterations and spatial diversity

In this subsection, we test the sensitivity of our algorithm to the number of EM iterations used. We demonstrate this for one and two receive antennas. Here we employ 6 pilots per OFDM symbol (in addition to the 16 pilots per symbol employed in the first ST block). From Figure 5.5, we see that the first iteration yields substantial improvement over the pilot-based estimation. However, iterating beyond that yields diminishing returns.

Figure 5.5 also illustrates that by increasing the number of receive antennas (thereby increasing the spatial receive diversity), the BER performance gets closer to the perfectly known channel case, indicating that spatial receive diversity provides more tolerance to channel estimation errors.

5.7.3 Sensitivity to Number of Pilots

Here, we keep the number of pilots in the first ST block fixed at 16 per OFDM symbol and vary the number of pilot tones in the subsequent ST blocks (we use 2, 6 and 10 pilots per symbol). The results are shown in Figure 5.6.

We note that the BER performance improves with increasing number of pilots. We also note that additional EM iterations can have substantial improvement for the low number of pilots case (e.g. the BER curve for the 2 pilots case is almost similar to that of the 6 pilots

case).

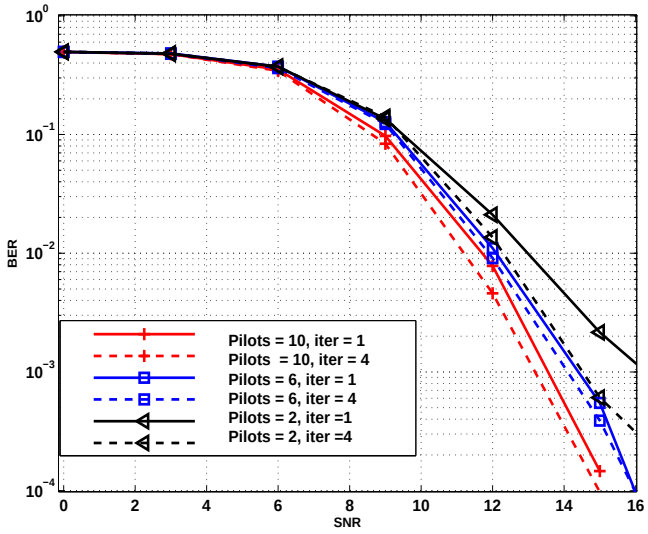

Figure 5.6: BER performance with varied number of pilots

5.7.4 Effect of Incorporating Frequency and Time Correlation in the Channel Estimation

The impact of using both frequency and time correlation in channel estimation is shown in Figure 5.7 for the 6-pilot scenario. In this figure, solid lines represent the one receive antenna case ($R_x = 1$) while the dashed lines stand for the two receive antenna case ($R_x = 2$). $P_e = 1$ refers to channel estimation using frequency correlation information only while $P_e = 2$ implies the use of both frequency and time correlation in channel estimation (see Section 5.5 for details).

We observe an error floor when only the frequency correlation information is used in channel estimation. This error floor remains regardless of the number of iterations. However, when we incorporate both frequency and time correlation information, we observe a significant improvement in BER (at a BER $= 10^{-2}$, the error floor drops by more than 10dB for $R_x = 1$ and $R_x = 2$). We also note that a single EM iteration provides substantial improvement when compared to the pilot-based estimation case.

We conclude that including time correlation in the channel estimation process (especially for channels with high time correlation) increases the amount of information that can be

harnessed by iterating.

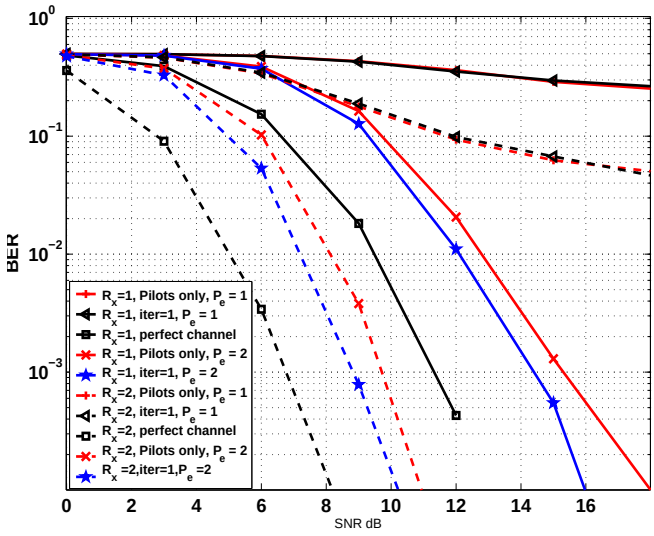

Figure 5.7: BER performance with frequency and time correlation

5.7.5 Sensitivity to Time Variation

In this subsection, we test the performance of our receiver against different degrees of time variation. This is parameterized by $\alpha\,(0 \leq \alpha \leq 1)$ with lower values of α indicating a more time-variant channel. In Figure 5.8 we show the BER curves for a system that employs 10-pilots per OFDM symbol.

From this figure, we observe that as α decreases (indicating more channel variation), the BER improves. This comes from increasing time diversity in the channel. Therefore, with enough number of pilots, we are able to track the channel and capture time diversity.

For comparison, in Figure 5.9, we show the BER curves for a system with fewer pilots (6-pilots per OFDM symbol) for $\alpha = 0.7, 0.8$ and 0.985. We observe an error-floor as the channel variation increases. So, in this case, we are unable to capture the time diversity. More pilots are thus needed to capture diversity and improve performance.

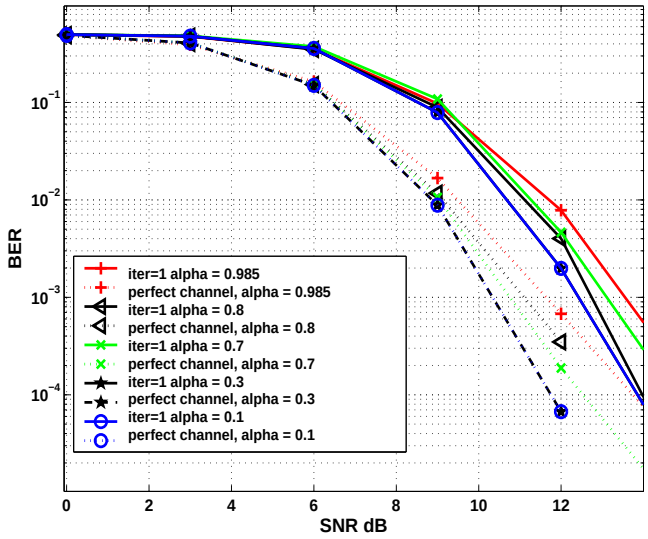

Figure 5.8: BER performance with varying channel correlation with 10 pilots

5.8 Practical Issues

5.8.1 Convergence and Stopping Criterion

For *deterministic* channels, i.e. in the absence of any correlation information, the minimum number of pilots needed for channel identifiability is equal to the number of taps of the MIMO impulse response [71]. However, when the channel becomes *random* as is the case in this chapter, we can decrease the number of pilots and compensate for the decrease with (frequency and time) correlation information. Quantifying the exact trade-off is beyond the scope of this chapter.

Each iteration of the EM algorithm produces an estimate of the channel $\hat{\underline{\boldsymbol{h}}}_d$ that monotonically increases the channel's likelihood function. This guarantees that the EM algorithm converges to a local maximum of the likelihood function [66]. Convergence to the global maximum depends on the initial condition from which the EM iterations are started.

Here, we use simulation results to investigate the effect of SNR and number of iterations on the convergence of the EM algorithm. We do this by plotting the mean-square error (MSE) between the channel estimate and the actual channel as a function of SNR. The MSE is plotted for pilot based estimation and for EM-based estimation with 1 and 4 iterations. We also plot the Cramer-Rao Bound (CRB) curve obtained when all the symbols are used as pilots [93]. We note that the pilot-based estimation curve in Figure 5.10 eventually reaches

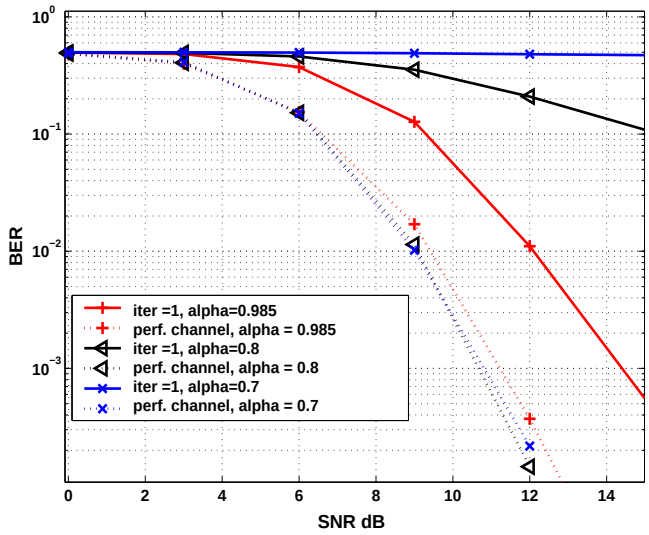

Figure 5.9: BER performance with varying channel correlation with 6 pilots

an error floor. From this figure, we also note that with increasing number of iterations and increasing SNR, the MSE of our estimate approaches that of the CRB.

Since the number of iterations used determines the computational complexity, we would like to devise a way to stop the iterations without compromising the performance of the EM algorithm. Thus, we will iterate for a maximum number of n_{iter} iterations as long as the percentage change in the two most recent channel estimates is greater than some constant η, i.e.

$$\frac{\|\hat{\underline{\boldsymbol{h}}}^{iter} - \hat{\underline{\boldsymbol{h}}}^{iter+1}\|^2}{\|\underline{\boldsymbol{h}}^{iter}\|^2} > \eta$$

We stop the algorithm if this condition is violated or if we reach the maximum number of iterations n_{iter}.

5.8.2 Robust Channel Estimation

For the proper operation of the receiver, the channel needs to follow the dynamical model and the receiver needs to have access to that model, i.e. to $\boldsymbol{F}$ and $\boldsymbol{G}$. When either of these assumptions is not true, the receiver assumes the dynamic model,

$$\underline{\boldsymbol{h}}_{r_x}^{t_x(+)} = \boldsymbol{F}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \boldsymbol{G}\boldsymbol{u} \tag{5.52}$$

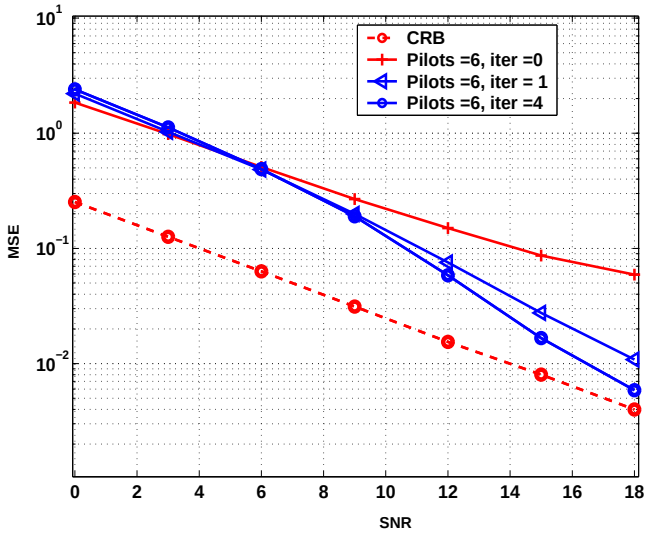

Figure 5.10: MSE vs. SNR

when the channel actually varies according to

$$\underline{\boldsymbol{h}}_{r_x}^{t_x(+)} = (\boldsymbol{F} + \Delta\boldsymbol{F})\underline{\boldsymbol{h}}_{r_x}^{t_x} + (\boldsymbol{G} + \Delta\boldsymbol{G})\boldsymbol{u}_{r_x}^{t_x} \tag{5.53}$$

where $\Delta\boldsymbol{F}$ and $\Delta\boldsymbol{G}$ characterize the uncertainties in the model. To minimize the effect of these uncertainties, one can implement robust versions of the Kalman filters, as done in [81].

5.8.3 Complexity

One of the main advantages of OFDM transmission is that it lends itself to simple equalization. [4] Thus, regardless of the OFDM receiver we employ, the major cost in complexity is incurred in channel estimation. So we will take the complexity of channel estimation as a bench mark to compare various receivers.

Roughly speaking, we can distinguish between three kinds of receivers: the non-iterative pilot based receiver, the iterative MMSE receiver (e.g. that of [58]), and the iterative Kalman filter based receiver we propose in this chapter. All of these methods employ some form of matrix inversion to solve for the MIMO impulse response, which consists of $T_x R_x (P+1)$

[4] In fact, we can show that the number of operations required to perform MMSE- or ML-based data detection of an OFDM symbol is always of the order $O(N)$.

taps. The cost of one such matrix inversion is thus $O(T_x^3 R_x^3 (P+1)^3)$ [47]. This inversion is done once for pilot-based estimation while its done n_{iter} number of times for the iterative receivers resulting in a complexity of $O(n_{iter} T_x^3 R_x^3 (P+1)^3)$. Thus, our Kalman based filter results in a linear increase in complexity as compared to the most basic (pilot-based) receiver and has comparable complexity to that of iterative MMSE receivers.

5.9 Conclusion

In this chapter, we have proposed a receiver for MIMO OFDM transmission over time-variant channels. The receiver makes full use of the data constraints (pilots, cyclic prefix, finite alphabet constraint and space-time code). It also exploits the channel constraints, particularly the time and frequency correlation. While we assumed the channel to be constant within the same space-time block, it is allowed to vary from one block to the next. This allows the receiver to operate in high speed environments. Apart from the outer code, the receiver also performs channel and data recovery within the same space-time block and hence avoids the need for data storage making the receiver suitable for real-time applications. When compared with other MIMO receivers, our receiver makes the most use of the underlying data and channel constraints.

The receiver employs the EM algorithm to achieve channel and data recovery. Specifically, the data recovery (or the expectation step) is as simple as decoding a space-time block code. Channel recovery (or the maximization step) is performed using a Kalman filter. Simulations demonstrated the favorable behavior of our receiver as compared to other receivers.

We can generalize the algorithm presented to include the effects of the transmit filter and the channel transmit and receive spatial correlation (see Appendix A). We can also modify the receiver to take care of (space-time) trellis as opposed to block codes. If storage and latency are not of concern, we can also modify the algorithm to perform estimation in the forward and backward direction resulting in better estimates, as done in Chapter 4.

5.10 APPENDIX A: Channel Model in the Presence of Spatial Correlation

In what follows, we present the transmit correlation case, and then generalize our results to deal with the general correlation case.

5.10.1 Transmit Correlation

In the transmit correlation case, $\boldsymbol{H}(\mathrm{p})$, the MIMO impulse response at tap p, is given by

$$\boldsymbol{H}(p) = \boldsymbol{W}(p)\boldsymbol{T}^{1/2}(p) \tag{5.54}$$

where $\boldsymbol{T}^{1/2}(p)$ is the transmit correlation matrix (of size T_x) at tap p and where $\boldsymbol{W}(p)$ consists of iid elements. The matrix $\boldsymbol{W}(p)$ remains constant over a single ST block and varies from one ST block to the next according to

$$\boldsymbol{W}^{(+)}(p) = \alpha(p)\boldsymbol{W}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}\boldsymbol{U}(p) \tag{5.55}$$

where $\alpha(p), \beta$, and $\boldsymbol{U}(p)$ are as defined in Subsection 5.2.3.

Just as we did in Subsection 5.2.3, we would like to construct a recursion for the tap $\underline{h}_{r_x}^{t_x}(p)$ and subsequently scale it up for the SISO and MIMO cases. Now since $\underline{h}_{r_x}^{t_x}(p)$ is the (r_x, t_x) element of $\boldsymbol{H}(p)$, we deduce from (5.54) that it is the inner product of the r_x row of $\boldsymbol{W}(p)$ and the t_x column of $\boldsymbol{T}^{1/2}$, i.e.

$$h_{r_x}^{t_x}(p) = \boldsymbol{w}_{r_x}(p)\boldsymbol{t}^{t_x}(p) \tag{5.56}$$

Moreover, from (5.55), we have the following recursion for $\boldsymbol{w}_{r_x}(p)$

$$\boldsymbol{w}_{r_x}^{(+)}(p) = \alpha(p)\boldsymbol{w}_{r_x}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}\boldsymbol{u}_{r_x}(p)$$

Post-multiplying both sides by $\boldsymbol{t}^{t_x}(p)$ yields

$$\boldsymbol{w}_{r_x}^{(+)}(p)\boldsymbol{t}^{t_x}(p) = \alpha(p)\boldsymbol{w}_{r_x}(p)\boldsymbol{t}^{t_x}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}\boldsymbol{u}_{r_x}(p)\boldsymbol{t}^{t_x}(p)$$

This means that $\underline{h}_{r_x}^{t_x}(p)$ satisfies the dynamical equation

$$\underline{h}_{r_x}^{t_x(+)}(p) = \alpha(p)\underline{h}_{r_x}^{t_x}(p) + \sqrt{(1-\alpha^2(p))e^{-\beta p}}ut_{r_x}^{t_x}(p) \tag{5.57}$$

where $ut_{r_x}^{t_x}$ is defined by

$$ut_{r_x}^{t_x}(p) = \boldsymbol{u}_{r_x}(p)\boldsymbol{t}^{t_x}(p)$$

Concatenating (5.57) for $p = 1, 2, \ldots, P$ yields a dynamic equation for the impulse response

$$\underline{\boldsymbol{h}}_{r_x}^{t_x} = \begin{bmatrix} h_{r_x}^{t_x}(0) \\ \vdots \\ h_{r_x}^{t_x}(P) \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_{r_x}(0)\boldsymbol{t}^{t_x}(0) \\ \vdots \\ \boldsymbol{w}_{r_x}(P)\boldsymbol{t}^{t_x}(P) \end{bmatrix}$$

which is the same as the dynamic equation (see (5.5)) for the spatially uncorrelated case

$$\underline{\boldsymbol{h}}_{r_x}^{t_x(+)} = \boldsymbol{F}\underline{\boldsymbol{h}}_{r_x}^{t_x} + \boldsymbol{G}\boldsymbol{u}\boldsymbol{t}_{r_x}^{t_x} \tag{5.58}$$

The only difference from the uncorrelated case is that $\boldsymbol{ut}_{r_x}^{t_x}$ is no more white. Rather, we have

$$E\left[\boldsymbol{ut}_{r_x}^{t_x}\boldsymbol{ut}_{r_x}^{t_x*}\right] \triangleq E\begin{bmatrix} \boldsymbol{u}_{r_x}(0)\boldsymbol{t}^{t_x}(0) \\ \boldsymbol{u}_{r_x}(1)\boldsymbol{t}^{t_x}(1) \\ \vdots \\ \boldsymbol{u}_{r_x}(P)\boldsymbol{t}^{t_x}(P) \end{bmatrix} \begin{bmatrix} \boldsymbol{t}^{t_x*}(0)\boldsymbol{u}_{r_x}^*(0) & \boldsymbol{t}^{t_x*}(1)\boldsymbol{u}_{r_x}^*(1) & \cdots & \boldsymbol{t}^{t_x*}(P)\boldsymbol{u}_{r_x*}(P) \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{tt}_{t_x}^{t_x}(0) & & & \\ & \boldsymbol{tt}_{t_x}^{t_x}(1) & & \\ & & \ddots & \\ & & & \boldsymbol{tt}_{t_x}^{t_x}(P) \end{bmatrix} \triangleq \mathrm{diag}(\boldsymbol{tt}_{t_x}^{t_x})$$

where

$$\boldsymbol{tt}_{r_x}^{t_x} = \begin{bmatrix} \boldsymbol{t}^{r_x*}(0)\boldsymbol{t}^{t_x}(0) \\ \boldsymbol{t}^{r_x*}(1)\boldsymbol{t}^{t_x}(1) \\ \vdots \\ \boldsymbol{t}^{r_x*}(P)\boldsymbol{t}^{t_x}(P) \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_{r_x}(0)\boldsymbol{t}^{t_x}(0) \\ \boldsymbol{t}_{r_x}(1)\boldsymbol{t}^{t_x}(1) \\ \vdots \\ \boldsymbol{t}_{r_x}(P)\boldsymbol{t}^{t_x}(P) \end{bmatrix}$$

and where the second line follows from the fact that $\boldsymbol{t}^{r_x*}(p) = \boldsymbol{t}_{t_x}(p)$ since $\boldsymbol{T}^{1/2}(p)$ is

conjugate symmetric. In general, we can show that

$$E[\boldsymbol{ut}_{r_x}\boldsymbol{ut}^*_{r'_x}] = \begin{bmatrix} \mathrm{diag}(\boldsymbol{tt}^1_1) & \mathrm{diag}(\boldsymbol{tt}^2_1) & \cdots & \mathrm{diag}(\boldsymbol{tt}^{T_x}_1) \\ \mathrm{diag}(\boldsymbol{tt}^1_2) & \mathrm{diag}(\boldsymbol{tt}^2_2) & \cdots & \mathrm{diag}(\boldsymbol{tt}^{T_x}_2) \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{diag}(\boldsymbol{tt}^1_{T_x}) & \mathrm{diag}(\boldsymbol{tt}^2_{T_x}) & \cdots & \mathrm{diag}(\boldsymbol{tt}^{T_x}_{T_x}) \end{bmatrix}$$

for $r_x = r'_x$ and is zero otherwise. Alternatively, we can write this as

$$E[\boldsymbol{ut}_{r_x}\boldsymbol{ut}^*_{r'_x}] = \begin{cases} \sum_{p=0}^{P} \boldsymbol{T}(p) \otimes \left(\underline{\boldsymbol{I}}^p \boldsymbol{B} \overline{\boldsymbol{I}}^p\right) & \text{for } r_x = r'_x \\ \boldsymbol{O} & \text{otherwise} \end{cases}$$

where

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \underline{\boldsymbol{I}} = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & 0 & \\ & & & 1 & 0 \end{bmatrix}, \quad \text{and} \quad \overline{\boldsymbol{I}} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & \ddots & & \\ & & \ddots & 1 & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}$$

Collecting (5.58) for all transmit and receive antennas yields

$$\underline{\boldsymbol{h}}^{(+)} = (\boldsymbol{I}_{T_x R_x} \otimes \boldsymbol{F})\, \underline{\boldsymbol{h}} + (\boldsymbol{I}_{T_x R_x} \otimes \boldsymbol{G})\, \boldsymbol{ut} \tag{5.59}$$

where

$$\begin{aligned} E[\boldsymbol{uu}^*] &= \boldsymbol{I}_{R_x} \otimes E[\boldsymbol{ut}_{r_x}\boldsymbol{ut}^*_{r_x}] \\ &= \sum_{p=0}^{P} \boldsymbol{I}_{R_x} \otimes \boldsymbol{T}(p) \otimes \left(\underline{\boldsymbol{I}}^p \boldsymbol{B} \overline{\boldsymbol{I}}^p\right) \end{aligned} \tag{5.60}$$

When the channel exhibits both transmit and receive correlation, the IR $\underline{\boldsymbol{h}}$ continues to satisfy the dynamical equation (5.59) except that the correlation of the innovation $\boldsymbol{u}$ is now given by

$$E[\boldsymbol{uu}^*] = \sum_{p=0}^{P} \boldsymbol{R}(p) \otimes \boldsymbol{T}(p) \otimes \left(\underline{\boldsymbol{I}}^p \boldsymbol{B} \overline{\boldsymbol{I}}^p\right)$$

5.11 APPENDIX B: Pilot/Output Equations for MIMO OFDM

Starting from the I/O equations constructed in (5.3), we derive the pilot/output equations that can be used for initial channel estimation. So let I_{p} denote the index set of pilot locations within the frequency bins. The pilot/output equations are simply those I/O equations whose numbers are elements of the index set I_{p}. Thus, from (5.19) and using the pruning notation of Chapter 4, we can write the pilot/output equations at antenna r_x and time instant t_b as

$$\boldsymbol{\mathcal{Y}}_{r_x\, I_{\mathrm{p}}}(t_b) = \sum_{n_s=1}^{N_s} \Big[\ \mathrm{diag}\Big(a_1^{t_b}(n_s)\mathcal{R}(\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)) + jb_1^{t_b}(n_s)\mathcal{I}\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)\Big) \quad \cdots \quad \mathrm{diag}\Big(a_{T_x}^{t_b}(n_s)\mathcal{R}(\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)) + jb_{T_x}^{t_b}(n_s)\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)\Big) \ \Big] \times \Big(\boldsymbol{I}\otimes\tilde{\boldsymbol{Q}}_{P+1}^{*}\Big)\,\underline{\boldsymbol{h}}_{r_x} + \boldsymbol{\mathcal{N}}_{r_x\, I_{\mathrm{p}}}(t_b)$$

We now proceed as we did in the full output case collecting the output $\boldsymbol{\mathcal{Y}}_{r_x I_{\mathrm{p}}}(t_b)$ over $t_b = 1, \ldots, T_b$ and over all receive antennas. This yields the pilot/output equation

$$\boxed{\boldsymbol{\mathcal{Y}}_{I_{\mathrm{p}}} = (\boldsymbol{I}\otimes\boldsymbol{X}_{I_{\mathrm{p}}})\underline{\boldsymbol{h}} + \boldsymbol{\mathcal{N}}_{I_{\mathrm{p}}}} \tag{5.61}$$

where

$$\boldsymbol{X}_{I_{\mathrm{p}}} = \sum_{n_s=1}^{N_s} \Big[\ \mathrm{diag}\Big(\boldsymbol{a}_1(n_s)\otimes\mathcal{R}\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s) + j\boldsymbol{b}_1(n_s)\otimes\mathcal{I}\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)\Big) \quad \cdots \quad \mathrm{diag}\Big(\boldsymbol{a}_{T_b}(n_s)\otimes\mathcal{R}\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s) + j\boldsymbol{b}_{T_b}(n_s)\otimes\mathcal{I}\boldsymbol{\mathcal{S}}_{I_{\mathrm{p}}}(n_s)\Big) \ \Big] \times \Big(\boldsymbol{I}\otimes\tilde{\boldsymbol{Q}}_{P+1}^{*}\Big)$$

Chapter 6

Conclusions and Future Work

6.1 Concluding Remarks

This dissertation has considered the analysis and design of adaptive algorithms for wireless channel estimation. The first part of the dissertation presented a framework for the transient analysis of adaptive filters with general data and error nonlinearities. The approach relies on energy conservation arguments. In addition to deriving earlier results in a unified manner, the approach leads to stability and performance results without restricting the regression data to being Gaussian or white. The framework also does not require an explicit recursion for the covariance matrix of the weight-error vector. We may add that extensions to leaky algorithms and to tracking analysis are possible and are treated in, e.g., [83].

The second part of the dissertation considered receiver design for SISO and MIMO OFDM transmission over frequency selective block-fading [1] channels. The receiver employs the expectation-maximization (EM) algorithm for joint channel and data recovery. It makes collective use of the data and channel constraints that characterize the communication problem. The data constraints include pilots, the cyclic prefix, the finite alphabet constraint, and space-time block coding. The channel constraints include the finite delay spread, sparsity, frequency and time correlation, and spacial correlation. The receiver algorithm becomes progressively more sophisticated as more data and channel constraints are incorporated, with each new version of the algorithm subsuming the previous version as a special case, culminating in the most general version, the EM-based forward-backward

[1] By block, we mean an OFDM symbol for SISO transmission and a space-time block for MIMO transmission.

Kalman filter.

The two parts of the dissertation share the following three themes:

Model-Based Filtering: Both the analysis and design of adaptive algorithms relied on some a priori models. For example, in performing adaptive filtering analysis, we adopted the system identification model, which relates the input and output according to (see Subsection 2.1.3 and Section 3.2)

$$d(i) = \mathbf{u}_i \boldsymbol{w}^o + v(i)$$

Similarly, in designing the OFDM receiver, we assumed that the channel evolves in time according to the dynamic model (see Subsections 4.2.5 and 5.2.3)

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i$$

Weighted-Norm Based Development: Weighted-norms were heavily used in the analysis and design of filters. Thus, mean-square analysis of adaptive filters was performed by carrying out the algebra on the Euclidean norm's weight. Similarly, the channel IR was estimated by minimizing a sum of weighted Euclidean norms (see the weighted-norm interpretation of Table 4.7). The norms involved and their weights depend on the a priori information about the channel and available information about the data.

State-Space Model: Both the analysis and design of adaptive algorithms are articulated in terms of state-space models. Thus, answering the questions of convergence and steady-state error for an adaptive algorithm with data (error) nonlinearity boils down to answering the same questions for a linear (nonlinear) time-invariant state-space model (see (2.55) and (3.22)). Similarly, channel estimation of the EM-based OFDM receiver boils down to applying a Kalman or FB-Kalman filter to some state-space model (see for example (4.73)).

The dissertation managed to answer several important questions related to the analysis and design of adaptive algorithms for channel estimation. Moreover, the transparency with which these questions were answered paves the way to ask other questions that can be pursued as future work, and which we elaborate on in the next section.

6.2 Future Work

6.2.1 Adaptive Filters with Optimized Behavior

Chapters 2 and 3 presented a unified performance analysis of adaptive filters employing general error or data nonlinearities. In particular, this general class of filters was characterized in terms of convergence speed and steady-state behavior. A natural question is, then, what is the optimum data or error nonlinearity that maximizes the convergence speed for a given steady-state performance, or, alternatively, reduces the steady-state error for a given convergence speed? Several papers have already attempted to answer this question and optimize the choice of data and error nonlinearities (e.g., as in [27, 3, 28, 14]). The optimum nonlinearities arrived at in these works were obtained by maximizing some performance measures which were themselves obtained under restrictive assumptions and approximations. The result is that the corresponding nonlinearities are in turn as good as or as valid as these assumptions and approximations.

In contrast, the performance analysis detailed in Chapters 2 and 3 is valid under much more general conditions. Starting from this general performance analysis, we can derive nonlinearities that are optimum under similarly general conditions.

6.2.2 Relaxing the Independence Assumption

The energy relationship described in Subsection 2.2.1 made it possible to relax many of the assumptions that are usually invoked in the study of adaptive filters. As a result, the independence assumption is the only assumption that Chapter 2 relies on for the mean-square analysis of adaptive filters with data nonlinearity. Moreover, by carefully examining the development of Chapter 2, we note that disposing of this assumption requires the manipulation of an infinite product of correlated random matrices [70]. There is significant literature on random matrices that has been successfully employed for recent advances in communication theory [96, 25, 67]. The same theory can be used to evaluate products of random matrices and hence study the performance of adaptive filters without relying on the independence assumption.

6.2.3 Reducing the Complexity of the OFDM Receiver

The presence of the cyclic prefix in OFDM transmission diagonalizes the communication channel. This simplifies the equalization part of the receiver because equalization can then

be carried out on a tone-by-tone basis. The presence of the cyclic prefix can be similarly helpful in reducing the computational complexity of the channel estimation part of the receiver. Specifically, from Subsection 4.2.1 we note that the input matrix $\boldsymbol{X}_i$ can be decomposed in the presence of a cyclic prefix as

$$\boldsymbol{X}_i = \text{diag}\left(\boldsymbol{\mathcal{X}}_i\right) \tilde{\boldsymbol{Q}}_{P+1}$$

i.e., as the product of a diagonal matrix and the partial FFT matrix $\tilde{\boldsymbol{Q}}_{P+1}$ (whose columns are orthogonal). These facts can be used to simplify the computational complexity of the Kalman and FB-Kalman filters.

6.2.4 Exchanging the Roles of the Channel and Data in Receiver Design

In receiver design, we need to recover two variables, the channel and the data. Recovering the two variables jointly is too computationally intensive. The EM algorithm employed in Chapters 4 and 5 performs the recovery iteratively by taking one unknown (the data) to be the hidden variable and the other unknown (the channel) to be the desired variable. We could also swap the roles the data and channel play in the EM algorithm, making the channel the hidden variable, and the data the desired variable. This approach will also result in a Kalman filter except that the state-variable will be twice as long. The advantage of this approach is that it generates the data estimate through maximum-likelihood estimation as opposed to mean-square estimation (as we do here).

6.2.5 OFDM Receiver Design Under Uncertainty

Chapters 4 and 5 considered receiver design for channels that remain constant during the transmission of an OFDM symbol and that can change arbitrarily at the OFDM symbol boundaries. Moreover, the chapters also assumed that the channel evolves according to a dynamical model

$$\underline{\boldsymbol{h}}_{i+1} = \boldsymbol{F}\underline{\boldsymbol{h}}_i + \boldsymbol{G}\boldsymbol{u}_i$$

and that the model is perfectly available at the receiver. All these assumption might not be valid in a real situation. Thus, the channel might change within the OFDM symbol itself (giving rise to intersymbol interference). Moreover, the state-space model may not accurately describe the evolution of the channel, and even when it does, the model is not usually available at the receiver.

These assumptions manifest themselves in the form of an uncertain state-space model

$$\underline{\boldsymbol{h}}_{i+1} = (\boldsymbol{F} + \boldsymbol{\Delta F})\underline{\boldsymbol{h}}_i + (\boldsymbol{G} + \boldsymbol{\Delta G})\boldsymbol{u}_i \quad (6.1)$$

$$\overline{\boldsymbol{\mathcal{Y}}}_i = \left(\overline{\boldsymbol{X}}_i + \boldsymbol{\Delta}\overline{\boldsymbol{X}}_i\right)\underline{\boldsymbol{h}}_i + \boldsymbol{n}_i \quad (6.2)$$

Deriving an EM-based receiver that is robust to these uncertainties would be a compelling future work.

6.2.6 Optimal Pilot Placement

The simulations of Chapter 4 demonstrate that pilot-placement heavily affects the performance of the Kalman and FB-Kalman receivers. The following questions are thus worthy of consideration:

1. How many pilots are needed in the first symbol? From a system identification perspective, we need as many pilots as channel taps. However, since we are employing an iterative procedure, how is the number of pilots affected the by number of the EM iterations and by the fact that the channel has some exponential decay profile (making the effect of some taps negligible)?

2. What is the best way to distribute a number of P pilots within an OFDM symbol? Is uniform distribution the best way to go? Does the knowledge of the frequency correlation among the taps help in placing the pilots optimally within the OFDM symbol?

3. Given a certain level of time variation (Doppler speed) and given a certain number of pilots to place in a number of OFDM symbols, what is the best way to distribute them, assuming causal processing at the receiver (Kalman filtering)? What is the best pilot distribution, assuming batch-processing (FB-Kalman filtering)?

Bibliography

[1] A. Aghamohammadi, H. Meyr, and G. Ascheid. Adaptive synchronization and channel parameter estimation using an extended kalman filter. *IEEE Trans. Commun.*, 37(11):1212–1219, Nov. 1989.

[2] T. Y. Al-Naffouri, A. Bahai, and A. Paulraj. Semi-blind channel identification and equalization in OFDM: an expectation-maximization approach. In *Proc. IEEE Vehicular Tech. Conf.*, volume 2, pages 13–17, Nov. 2002.

[3] T. Y. Al-Naffouri and A. H. Sayed. Optimum error nonlinearities for long adaptive filters. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 2, pages 1373–1376, 2002.

[4] T. Y. Al-Naffouri and A. H. Sayed. Transient analysis of data-normalized adaptive filters. *IEEE Trans. Signal Proc.*, 51(3), Mar. 2003.

[5] T. Y. Al-Naffouri, A. Zerguine, and M. Bettayeb. Convergence analysis of the LMS algorithm with a general error nonlinearity and an iid input. In *Proc. Asilomar Conf. on Signals, Syst., and Computers*, volume 1, pages 556–559, Nov. 1998.

[6] S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE J. Select. Areas Commun.*, 16:1451–1458, Oct. 1998.

[7] C. Aldana, E. de Carvalho, and J. M. Cioffi. Channel estimation for multicarrier multiple input single output systems using the EM algorithm. *IEEE Trans. Signal Proc.*, 51(12):3280–3292, Dec. 2003.

[8] G. Alrawi, T. Y. Al-Naffouri, A. Bahai, and J. Cioffi. Exploiting error-control coding and cyclic prefix in channel estimation for coded OFDM systems. In *Proc. IEEE Globecom*, pages 1152–1156, Nov. 2002.

[9] G. Alrawi, T. Y. Al-Naffouri, A. Bahai, and J. Cioffi. Exploiting error-control coding and cyclic prefix in channel estimation for coded OFDM systems. *IEEE Commun. Lett.*, 7(7):388–390, Jul. 2003.

[10] G. Alrawi, T. Y. Al-Naffouri, A. Bahai, and J. Cioffi. An iterative receiver for coded OFDM systems over time-varying wireless channels. In *Proc. IEEE Int. Conf. Commun.*, pages 3371–3376, May 2003.

[11] Harold Artés. *Algorithms for Time-Varying Channels: Scattering Function Estimation and Blind Equalization.* PhD thesis, Oct. 2003.

[12] J. Bermudez and N. J. Bershad. A nonlinear analytical model for the quantized LMS algorithm– the arbitrary step size case. *IEEE Trans. Signal Proc.*, 44(5):1175–1183, May 1996.

[13] N. J. Bershad. Analysis of the normalized LMS algorithm with gaussian inputs. *IEEE Trans. Acoust. Speech Signal Process.*, 34:793–806, 1986.

[14] N. J. Bershad. On the optimum data nonlinearity in LMS adaptation. *IEEE Trans. Acoust. Speech Signal Process.*, 34(1):69–76, Feb. 1986.

[15] N. J. Bershad. Behavior of the ϵ-normalized LMS algorithm with gaussian inputs. *IEEE Trans. Acoust. Speech Signal Process.*, 35:636–644, May 1987.

[16] N. J. Bershad. On error saturation nonlinearities in LMS adaptation. *IEEE Trans. Acoust. Speech Signal Process.*, 36(4):440–452, Apr. 1988.

[17] N. J. Bershad and M. Bonnet. Saturation effects in LMS adaptive echo cancellation for binary data. *IEEE Trans. Acoust. Speech Signal Process.*, 38(10):1687–1696, Oct. 1990.

[18] P. Billingsley. *Probability and Measure.* J. Wiley & Sons, 1995.

[19] T. Birkel. Laws of large numbers under dependence assumptions. *Statistics and Probability Lett.*, 14(4):355–362.

[20] H. Bölcskei, R. W. Heath, and A. J. Paulraj. Blind channel identification and equalization in OFDM-based multi-antenna systems. *IEEE Trans. Signal Proc.*, 50(1):96–109, Jan. 2002.

[21] J.K. Cavers. An analysis of pilot symbol assisted modulation for Rayleigh fading channels (mobile radio). *IEEE Trans. Vehicular Tech.*, 40(4):686–693, Nov. 1991.

[22] J. A. Chambers, O. Tanrikulu, and A. G. Constantindes. Least-mean mixed-norm adaptive filtering. *Electron. Lett.*, 30(19):1574–1575, Sep. 1994.

[23] T. Claasen and W. Mecklenbräuker. Comparison of the convergence of two algorithms for adaptive FIR digital filters. *IEEE Trans. Circuits Syst.*, 28(6):510–518, Jun. 1981.

[24] C. Cozzo and B. L. Hughes. Joint channel estimation and data detection in space-time communications. *IEEE Trans. Commun.*, 51(8):1266–1270, Aug. 2003.

[25] M. Debbah, P. Loubaton, and M. de Courville. Asymptotic performance of successive interference cancellation in the context of linear precoded OFDM systems. *IEEE Trans. Commun.*, 52(9):1444–1448, Sep. 2004.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc., Ser. B*, 39(1):1–38, 1977.

[27] S. C. Douglas and T. H. Y Meng. The optimum scalar data nonlinearity in LMS adaptation for arbitrary iid inputs. *IEEE Trans. Signal Proc.*, 40(6):1566–1570, Jun. 1992.

[28] S. C. Douglas and T. H. Y Meng. Normalized data nonlinearities for LMS adaptation. *Signal Proc.*, 42(6):1352–1365, Jun. 1994.

[29] S. C. Douglas and T. H. Y Meng. Stochastic gradient adaptation under general error criterion. *IEEE Trans. Signal Proc.*, 42(6):1335–1351, Jun. 1994.

[30] S. C. Douglas and W. Pan. Exact expectation analysis of the LMS adaptive filter. *IEEE Trans. Signal Proc.*, 43(12):2863–2871, Dec. 1995.

[31] D. L. Duttweiler. Adaptive filter performance with nonlinearities in the correlation multiplier. *IEEE Trans. Acoust. Speech Signal Process.*, 30(4):578–586, Aug. 1982.

[32] O. Edfors, M. Sandell, J. van de Beek, K. S. Wilson, and P. O. Brjesson. OFDM channel estimation by singular value decomposition. *IEEE Trans. Signal Proc.*, 46(7):931–939, Jul. 1998.

[33] W. B. Mikhael et al. Adaptive filters with individual adaptation of parameters. *IEEE Trans. Circuits Syst.*, 33(7), Jul. 1986.

[34] E. Eweda. Analysis and design of signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated gaussian data. *IEEE Trans. Circuits Syst.*, 37(11), Nov. 1990.

[35] E. Eweda. Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. *IEEE Trans. Signal Proc.*, 42(11):2937–2944, Nov. 1994.

[36] A. Feuer and E. Weinstein. Convergence analysis of LMS filters with uncorrelated gaussian data. *IEEE Trans. Acoust. Speech Signal Process.*, 33(1), 1985.

[37] S. Florian and A. Feuer. Performance analysis of the LMS algorithm with a tapped delay line (two-dimensional case). *IEEE Trans. Acoust. Speech Signal Process.*, 34(6):1542–1549, Dec. 1986.

[38] W. A. Gardner. Learning characteristic of stochastic-descent algorithms: A general study, analysis, and critique. *Signal Proc.*, 6(2):113–133, Apr. 1984.

[39] A. Ghorokhov and J. Linnartz. Robust OFDM receivers for dispersive time varying channels: equalization and channel acquisition. In *Proc. IEEE Int. Conf. Commun.*, pages 470–474, 2002.

[40] G.B. Giannakis. Filter banks for blind channel identification and equalization. *IEEE Signal Proc. Lett.*, 4(6):184–187, Jun. 1997.

[41] J. Gibson and S. Gray. MVSE adaptive filtering subject to a constraint on MSE. *IEEE Trans. Circuits Syst.*, 35(5):603–608, May 1988.

[42] R. W. Harris, D. M. Chabries, and F. A. Bishop. A variable step (VS) adaptive filter algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, 34(2):309–316, Apr. 1986.

[43] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 2004.

[44] R.W. Heath and G. B. Giannakis. Exploiting input cyclostationarity for blind channel identification in OFDM systems. *IEEE Trans. Signal Proc.*, 47(3):848–856, Mar. 1999.

[45] R.A. Iltis. Joint estimation of PN code delay and multipath using the extended kalman filter. *IEEE Trans. Commun.*, 38(10):1677–1685, Oct. 1990.

[46] S. K. Jones, R. K. Cavin, and W. M. Reed. Analysis of error-gradient adaptive linear estimators for a class of stationary dependent processes. *IEEE Trans. Inform. Theory*, 28(3):318–329, Mar. 1982.

[47] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear estimation*. Prentice Hall, 2000.

[48] I. Kang, M. P. Fitz, and S. B. Gelfand. Blind estimation of multipath channel parameters: a modal analysis approach. *IEEE Trans. Commun.*, 47(8):1140–1150, Aug. 1999.

[49] S. Koike. Convergence analysis of a data echo canceller with a stochastic gradient adaptive FIR filter using the sign algorithm. *IEEE Trans. Signal Proc.*, 43(12):2852–2861, Dec. 1995.

[50] C. Komninakis, C. Fragouli, A. H. Sayed, and R. Wesel. Multi-input multi-output fading channel tracking and equalization using kalman estimation. *IEEE Trans. Signal Proc.*, 50(5):1065–1076, May 2002.

[51] E. Larsson and P. Stoica. *Space-Time Block Coding for Wireless Communications*. Cambridge University Press, 2003.

[52] G. Leus and M. Moonen. Semi-blind channel estimation for block transmissions with non-zero padding. In *Proc. Asilomar Conf. on Signals, Syst., and Computers*, pages 762–766, Nov. 2001.

[53] Y. Li. Pilot-symbol-aided channel estimation for OFDM in wireless systems. In *Proc. IEEE Vehicular Tech. Conf.*, volume 2, pages 1131–1135, Jul. 1999.

[54] Y. Li, L. J. Cimini, and N. R. Sollenberger. Robust channel estimation for OFDM systems with rapid dispersive fading channels. *IEEE Trans. Commun.*, 46(7):902–915, Jul. 1998.

[55] Y. Li, C. N. Georghiades, and H. Garng. Iterative maximum-likelihood sequence estimation for space-time coded systems. *IEEE Trans. Commun.*, 49(6):948–951, Jun. 2001.

[56] Y. Li, N. Seshadri, and S. Ariyavisitakul. Channel estimation for OFDM systems with transmitter diversity in mobile wireless channels. *IEEE J. Select. Areas Commun.*, 17:461–471, Mar. 1999.

[57] J.-P.M.G. Linnartz and A. Gorokhov. New equalization approach for over dispersive and rapidly time varying channel. In *Symposium on Personal, Indoor, Mobile Radio Commun.*, pages 1375–1379, 2000.

[58] B. Lu, X. Wang, and Y. Li. Iterative receivers for space-time block-coded OFDM systems in dispersive fading channels. *IEEE Trans. Wireless Commun.*, 1(2):213–225, Apr. 2002.

[59] H. Lütkepohl. *Handbook of Matrices*. John Wiley & Sons, 1996.

[60] O. Macchi. *Adaptive Processing: The LMS Approach with Applications in Transmission*. Wiley, 1995.

[61] J. Mai and A. H. Sayed. A feedback approach to the steady-state performance of fractionally-spaced blind adaptive equalizers. *IEEE Trans. Signal Proc.*, 48(1):80–91, Jan. 2000.

[62] E. Masry and F. Bullo. Convergence analysis of the sign algorithm for adaptive filtering. *IEEE Trans. Inform. Theory*, 41(1):489–495, Mar. 1995.

[63] V. Mathews and S. Cho. Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, 35(4):450–454, Apr. 1987.

[64] J. E. Mazo. On the independence theory of equalizer convergence. *Bell Syst. Tech. J.*, 58(2):963–993, May/Jun. 1979.

[65] Xiao ming Chen and P. A. Hoeher. Blind equalization with iterative joint channel and data estimation for wireless dpsk systems. In *Proc. IEEE Globecom*, pages 274–279, 2001.

[66] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Proc. Mag.*, 13(6):47–60, Nov. 1996.

[67] R. R. Muller. On the asymptotic eigenvalue distribution of concatenated vector-valued fading channels. *IEEE Trans. Inform. Theory*, 48(7):2086–2091, Jul. 2002.

[68] B. Muquet, M. de Courville, G. B. Giannakis, Z. Wang, and P. Duhamel. Reduced complexity equalizers for zero-padded OFDM transmissions. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pages 2973–2976, Istanbul, Turkey, Jun. 2000.

[69] V. H. Nascimento and A. H. Sayed. Stability of the LMS adaptive filter by means of a state equation. In *Proc. Allerton Conf. Commun., Contr., Computing*, volume 36, pages 242–251, Oct. 1998.

[70] V. H. Nascimento and A. H. Sayed. On the learning mechanism of adaptive filters. *IEEE Trans. Signal Proc.*, 48(6), Jun. 2000.

[71] R. Negi and J. Cioffi. Pilot tone selection for channel estimation in a mobile OFDM system. *IEEE Trans. Consumer Electr.*, 44(3):1122–1128, Aug. 1998.

[72] S. Ohno and G. B. Giannakis. Optimal training and redundant precoding for block transmissions with application to wireless OFDM. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pages 2389–2392, 2001.

[73] A. Paulraj, R. Nabar, and D. Gore. *Introduction to Space Time Wireless Communications*. Cambridge University Press, 2003.

[74] R. Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Trans. Inform. Theory*, 4:69–72, Jun. 1958.

[75] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2001.

[76] M. Rupp. The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes. *IEEE Trans. Signal Proc.*, 41(3):1149–1160, Mar. 1993.

[77] M. Rupp and A. H. Sayed. A time-domain feedback analysis of filtered-error adaptive gradient algorithms. *IEEE Trans. Signal Proc.*, 44(6):1428–1439, Jun. 1996.

[78] M. Rupp and A. H. Sayed. On the convergence of blind adaptive equalizers for constant-modulus signals. *IEEE Trans. Commun.*, 48(5):795–803, May 2000.

[79] F. Sanzi, S. Jelting, and J. Speidel. A comparative study of iterative channel estimators for mobile OFDM systems. *IEEE Trans. Wireless Commun.*, 2(5):849–859, Sep. 2003.

[80] F. Sanzi and M. C. Necker. Totally blind APP channel estimation for mobile OFDM systems. *IEEE Commun. Lett.*, 7(11):517–519, Nov. 2003.

[81] A. H. Sayed. A framework for state-space estimation with uncertain models. *IEEE Trans. Auto. Control.*, 46(7):998–1013, Jul. 2001.

[82] A. H. Sayed. *Fundamentals of Adaptive Filtering.* Wiley, 2003.

[83] A. H. Sayed and T. Y. Al-Naffouri. Mean-square analysis of normalized leaky adaptive filters. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume VI, May 2001.

[84] A. H. Sayed and M. Rupp. A time-domain feedback analysis of adaptive algorithms via the small gain theorem. In *Proc. SPIE*, volume 2563, pages 458–469, Jul. 1995.

[85] A. H. Sayed and M. Rupp. *Robustness issues in adaptive filtering in* DSP Handbook. CRC Press, 1998.

[86] W. Sethares. Adaptive algorithms with nonlinear data and error functions. *IEEE Trans. Signal Proc.*, 40(9):2199–2206, Sep. 1992.

[87] R. Sharma, W. Sethares, and J. Bucklew. Asymptotic analysis of stochastic gradient-based adaptive filtering algorithms with general cost functions. *IEEE Trans. Signal Proc.*, 44(9):2186–2194, Sep. 1996.

[88] Z. Shengli and G. B. Giannakis. Finite-alphabet based channel estimation for OFDM and related multicarrier systems. *IEEE Trans. Commun.*, 49(8):1402–1414, Aug. 2001.

[89] M. K. Simon and M. S. Alouini. A unified approach to the probability of error for noncoherent and differentially coherent modulations over generalized fading channels. *IEEE Trans. Commun.*, 46(1):1625–1638, Dec. 1998.

[90] D. T. M. Slock. On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Trans. Signal Proc.*, 41(9):2811–2825, Sep. 1993.

[91] O. Tanrikulu and J. A. Chambers. Convergence and steady-state properties of the least-mean mixed-norm (LMMN) adaptive algorithm. *IEE Proc.– Vision, Image, and Signal Proc.*, 143(3):137–142, Jun. 1996.

[92] M. Tarrab and A. Feuer. Convergence and performance analysis of the normalized LMS algorithm with uncorrelated gaussian data. *IEEE Trans. Inform. Theory*, 34(4):680–690, Jul. 1988.

[93] H. L. Van Trees. *Detection, Estimation and Modulation Theory, Part I.* John Wiley and Sons, 1968.

[94] M. K. Tsatsanis, G. B. Giannakis, and G. Zhou. Estimation and equalization of fading channels with random coefficients. *Signal Proc.*, 53(2-3):211–229, Sept. 1996.

[95] F. Tufvesson and T. Maseng. Pilot assisted channel estimation for OFDM in mobile cellular systems. In *Proc. IEEE Vehicular Tech. Conf.*, volume 3, pages 1693–1643, May 1997.

[96] A. M. Tulino and S. Verdu. Asymptotic analysis of improved linear receivers for bpsk-cdma subject to fading. *IEEE J. Select. Areas Commun.*, 19(8):1544–1555, Aug. 2001.

[97] M.C. Vanderveen, A.-J. Van der Veen, and A. Paulraj. Estimation of multipath parameters in wireless communications. *IEEE Trans. Signal Proc.*, 46(3):682–690, Mar. 1998.

[98] E. Walach and B. Widrow. The least-mean fourth (LMF) adaptive algorithm and its family. *IEEE Trans. Inform. Theory*, 30(2):275–283, Aug. 1984.

[99] Xiaowen Wang and K. J. Ray Liu. Adaptive channel estimation using cyclic prefix in multicarrier modulation system. *IEEE Commun. Lett.*, 3(10):291–293, Oct. 1999.

[100] B. Widrow and S. D. Stearns. *Adaptive Signal Processing.* Prentice Hall, 1985.

[101] Y. Xie and C. N. Georghiades. Two EM-type channel estimation algorithms for OFDM with transmitter diversity. *IEEE Trans. Commun.*, 51(1):106–115, Jan. 2003.

[102] B. Yang, K. Ben Letaief, R. Cheng, and Z. Cao. Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modelling. *IEEE Trans. Commun.*, 49(3):467–479, Mar. 2001.

[103] N. R. Yousef and A. H. Sayed. A feedback analysis of the tracking performance of blind adaptive equalization algorithms. In *Proc. Conf. Decision and Contr.*, volume 1, pages 174–179, Dec. 1999.

[104] N. R. Yousef and A. H. Sayed. Tracking analysis of the LMF and LMMN adaptive algorithms. In *Proc. Asilomar Conf. on Signals, Syst., and Computers*, volume 1, pages 786–790, Nov. 1999.

[105] N. R. Yousef and A. H. Sayed. A unified approach to the steady-state and tracking analyses of adaptive filters. *IEEE Trans. Signal Proc.*, 49(2):314–324, Feb. 2001.

[106] N. R. Yousef and A. H. Sayed. Ability of adaptive filters to track carrier offsets and random channel nonstationarities. *IEEE Trans. Signal Proc.*, 50(7):1533–1544, Jul. 2002.

[107] W. Zhendao and G. B. Giannakis. Wireless multicarrier communications. *IEEE Signal Proc. Mag.*, 17(3):29–48, May 2000.

[108] S. Zhou, B. Muquet, and G. B. Giannakis. Subspace-based (semi-) blind channel estimation for block precoded space-time OFDM. *IEEE Trans. Signal Proc.*, 50(5):1215–1228, May 2002.