Approximate Inference for Wireless Communications

Hansen, Morten; Christensen, Lars P.B.; Winther, Ole

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Approximate Inference for Wireless Communications

Morten Hansen

Kongens Lyngby 2009
IMM-PHD-2009-227

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Abstract

This thesis investigates signal processing techniques for wireless communication receivers. The aim is to improve the performance or reduce the computationally complexity of these, where the primary focus area is cellular systems such as Global System for Mobile communications (GSM) (and extensions thereof), but also general Multiple-Input Multiple-Output (MIMO) systems are considered. The motivation for a performance improvement is that this is needed to achieve higher capacity in the systems, which can ensure increased bit-rates at the same or lower prices. A reduction in the computationally complexity can potentially lead to limited power consumption, which translates into longer battery life-time in the handsets.

The scope of the thesis is more specifically to investigate approximate (near-optimal) detection methods that can reduce the computationally complexity significantly compared to the optimal one, which usually requires an unacceptable high complexity. Some of the treated approximate methods are based on QL-factorization of the channel matrix. In the work presented in this thesis it is proven how the QL-factorization of frequency-selective channels asymptotically provides the minimum-phase and all-pass filters. This enables us to view Sphere Detection (SD) as an adaptive variant of minimum-phase pre-filtered reduced-state sequence estimation. Thus, a novel way of computing the minimum-phase filter and its associated all-pass filter using the numerically stable QL-factorization is suggested. Alternatively, fast QL-factorization methods can be applied which provides a computationally efficient way of obtaining these filers.

Additionally, Markov Chain Monte Carlo (MCMC) sampling has been investigated for near-optimal Maximum Likelihood Sequence Detection in MIMO

systems. The MCMC method considered in the thesis is the Gibbs sampler, which is proposed as an alternative to the SD in scenarios where the latter type of detector requires an unacceptable high complexity.

Resumé (Abstract in Danish)

Denne afhandling undersøger signalbehandlingsteknikker til trådløse kommunikationsmodtagere. Det overordnede mål er at forbedre ydeevnen eller reducere beregningskompleksiteten i disse, hvor det primære fokusområde er cellebaserede netværk såsom GSM (eller udvidelser deraf), men også generelle Multiple-Input Multiple-Output (MIMO) systemer vil blive betragtet. Motivationen for at forbedre ydeevnen i sådanne systemer er, at dette vil være nødvendigt hvis man vil forøge kapaciteten i disse, hvilket kan sikre en højere data-rate til samme eller lavere pris. En reduktion i beregningskompleksiteten vil potentielt set medføre et lavere strømforbrug, hvilket fører til længere batterilevetid for mobiltelefoner.

Formålet med afhandlingen er mere specifikt at undersøge approksimative (nær-optimale) detektionsmetoder, som kan reducere beregningskompleksiteten betydeligt sammenlignet med den optimale modtager, eftersom denne oftest vil kræve en uacceptabel høj kompleksitet. Nogle af de approksimative metoder som undersøges er baseret på QL-faktorisering af kanalmatricen. Det vises, hvorledes man ved en QL-faktorisering af frekvens-selektive kanaler kan opnå minimum-fase og all-pass filtrene. Herved er det muligt at betragte "Sphere Detection" (SD) som en adaptiv form for minimum-fase præ-filtreret "reduced-state sequence estimation". Der foreslås dermed en ny metode til at beregne minimum-fase filteret og det dertilhørende all-pass filter ved brug af den numerisk stabile QL-faktorisering. Alternativt kan "hurtige" QL-faktoriseringsmetoder benyttes, hvilket beregningsmæssigt set resulterer i effektive metoder til at udregne disse filtre.

Endvidere er "Markov Chain Monte Carlo" (MCMC) sampling blevet undersøgt til at opnå "Maximum Likelihood" sekvens detektion i MIMO-systemer. MCMC

metoden, som betragtes i afhandlingen er den såkaldte “Gibbs sampler”, der er blevet foreslået som et alternativ til SD, når denne type detektor vil kræve en uacceptabel høj beregningskompleksitet.

Preface

This thesis has been prepared at the department of Informatics Mathematical Modelling, Technical University of Denmark and Modem Algorithm Design, Nokia Denmark A/S in partial fulfillment of the requirements for acquiring the Ph.D. degree in electrical engineering.

The thesis deals with different aspects of signal processing techniques for detection in wireless communications. A main focus area has been the complexity of the detection methods, since this is a crucial factor in the design of "real-life" wireless communication systems.

The thesis consists of a summary report and a collection of four research papers written during the period 2006-2009.

Lyngby, December 2009

Morten Hansen

Papers included in the thesis

[A] Morten Hansen, Lars P. B. Christensen, and Ole Winther. On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation. *IEEE Global Telecommunications Conference (GLOBECOM)*. November 2007.

[B] Morten Hansen and Lars P. B. Christensen. Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. April 2009.

[C] Morten Hansen, Lars P. B. Christensen, and Ole Winther. Computing the Minimum-Phase Filter using the QL-Factorization. *IEEE Transactions on Signal Processing*. Submitted in June 2009, accepted, waiting for publication.

[D] Morten Hansen, Babak Hassibi, Alexandros G. Dimakis, and Weiyu Xu. Near-Optimal Detection in MIMO Systems using Gibbs Sampling. *IEEE Global Telecommunications Conference (GLOBECOM)*. November 2009.

Acknowledgements

I would like to thank the Technical University of Denmark (DTU) for allowing me the opportunity of doing this work. I would also like to thank Nokia Denmark A/S and the Modem Algorithm Design (MAD) group for partly supporting the Ph.D. study. A special thanks goes to Izydor Sokoler and in particular Niels Mørch for taking the effort of setting up the Ph.D. study in co-operation with DTU, and for giving me the opportunity to participate in activities in the MAD group.

I am very grateful to my supervisors Lars P. B. Christensen and Ole Winther, who have allowed me the freedom to pursue my own ideas while still ensuring that the research was going in a fruitful direction. I appreciate your mentorship and your willingness to come with helpful suggestions, in particular when I got stuck in my research.

I would also like to thank Lars Kai Hansen and Jan Larsen for interesting discussions in various topics in signal processing and machine learning. Furthermore, I want to thank the rest of my colleagues at both the department of Informatics Mathematical Modelling and Modem Algorithm Design for improving the working environment and for talks of both social and professional matter. A special thanks goes to Pedro Højen-Sørensen, Morten Hagdrup, and Søren S. Christensen for always being willing to share knowledge and ideas.

From California Institute of Technology I wish to thank Babak Hassibi for letting me visit his group in a period of five month and for letting me freely interact with the students in his group. In this context I would like to thank Alexandros G. Dimakis and Weiyu Xu for fruitful joint work. I would also like to express

my gratitude to Ravi Teja Sukhavasi, for making me feel welcome throughout the visit and for many hours of fun together.

I wish to thank Klaus S. Andersen and Carsten Stahlhut for proofreading this thesis.

Finally, I would like to thank my girlfriend Stine for her support, love, and encouragement over the years.

Nomenclature

a.k.a.	Also Known As
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BS	Base Station
CIR	Channel Impulse Response
DARE	Discrete-time Algebraic Riccati Equation
DFE	Decision-Feedback Estimation
EDGE	Enhanced Data rates for GSM Evolution (a.k.a. EGPRS)
EGPRS	Enhanced GPRS
FDMA	Frequency Division Multiple Access
FIR	Finite Impulse Response
GPRS	General Packet Radio Service
GS	Gibbs Sampling
GSM	Global System for Mobile communications
HT	Hilly Terrain
i.i.d.	Independent and Identically-Distributed
IIR	Infinite impulse response
ISI	Inter-Symbol Interference
LHS	Left-Hand-Side

LMMSE	Linear Minimum Mean-Square Error
LS	Least Squares
MAC	Multiply and Accumulate
MAP	Maximum A-Posteriori
MCMC	Markov Chain Monte Carlo
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detection
MS	Mobile Station
MUD	Multi User Detection
NP-hard	Non-deterministic Polynomial-time hard
PDF	Probability Density Function
RHS	Right-Hand-Side
RSSE	Reduced-State Sequence Estimation
SD	Sphere Detection
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
SUD	Single User Detection
TDMA	Time Division Multiple Access
TU	Typical Urban

Notation

General terms

x	Scalar
$\mathbf{x}$	Column vector
x_i	The ith element of $\mathbf{x}$
$\mathbf{x}_{\setminus i}$	Excluding the ith element in $\mathbf{x}$
$\mathbf{x}_i$	The ith "vector element" of $\mathbf{x}$
$\mathbf{x}_{1:N}$	The vector containing $\left[\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T\right]^T$
$\mathbf{X}$	Matrix
$(\mathbf{X})_{i,j}$	The (i,j)th element of matrix $\mathbf{X}$
$X_{i,j}$	The (i,j)th element of matrix $\mathbf{X}$
$\mathbf{I}_M$	Identity matrix of size $M \times M$
$\mathbf{0}_{M\times N}$	All-zero matrix of size $M \times N$
$\mathbf{1}_{M\times N}$	All-one matrix of size $M \times N$
i	$\sqrt{-1}$
$\lvert\cdot\rvert$	Absolute value of a complex number
$\angle$	Angular component of a complex number
$(\cdot)^*$	Complex conjugation
$\mathrm{P}(\cdot)$	Probability
E	Statistical expectation operator
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Complex-valued Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\boldsymbol{\chi}^2_N$	Chi-Square distribution with N complex-valued degrees-of-freedom
$\mathcal{O}(N)$	Of the order of N

Set operators

$\Re$	Real part
$\Im$	Imaginary part
$\mathbb{Z}$	The set of integer numbers
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\varnothing$	The empty set
$\lvert\cdot\rvert$	Cardinality of the set

Vector operators

$\text{diag}\,(\mathbf{x})$	Diagonal matrix with $\mathbf{x}$ in the diagonal

Matrix operators

$(\cdot)^{-1}$	Inverse matrix
$(\cdot)^{T}$	Transposed matrix
$(\cdot)^{H}$	Transposed and complex conjugated matrix
$\mathbf{X}^{+}$	Pseudo inverse matrix of the matrix $\mathbf{X}$
$\mathbf{X}^{n}$	The nth power of a square matrix $\mathbf{X}$
$\lVert\cdot\rVert_F$	Frobenius norm
$\text{diag}(\cdot)$	Vector given by diagonal of the matrix
$\det(\cdot)$	Determinant of matrix
$\text{tr}(\cdot)$	Trace operation
$\text{rank}(\cdot)$	Rank of matrix
$\otimes$	Kronecker product
$\lambda_i(\mathbf{X})$	The ith eigenvalue of $\mathbf{X}$
$\sigma_i(\mathbf{X})$	The ith singular value of $\mathbf{X}$

Contents

Chapter 1

Introduction and Motivation

Wireless communications is a very active research area due to the ever increasing demand for higher capacity and due to the huge amount of revenue there is in this field. As an example of the latter, we cite [1] where it is stated that "*The telecommunications industry is one of the largest industries worldwide, with more than* $1 *trillion in annual revenue for services and equipment*". Furthermore, according to the GSM Association (an association of mobile operators and related companies) the number of mobile connections[1] is now (in February 2009) above 4 billion and the association has predicted that the world will reach 6 billion connections by 2013.[2]
To keep up with the demand for ever higher bit-rates at the same or lower prices, several challenges must be met. Examples are:

- Higher-order modulations used for greater spectral-efficiency.
- Multipath propagation caused by reflections between transmitting and receiving antennas.
- Interference from other data streams and/or users in the system.
- Noise from analog processing, i.e. antenna and Radio Frequency (RF) front-end processing.

[1] The number of connections does not directly translate into the number of users, since a user may have multiple mobile phones.

[2] The press release by the GSM Association with the headline "Mobile World Celebrates Four Billion Connections" has been announced February 11 2009.

The optimal detector for the received signal is known (assuming that the received signal can be described by a linear model) but its complexity scales exponentially with the number of streams/users and the length of the channel.[3] This makes the optimal solution highly unrealistic for many real-life scenarios as the associated cost, power, and size would be unacceptable.

The focus of the Ph.D. thesis is to investigate methods for approximate inference of the transmitted information that preserves a near-optimal performance, but has a significantly reduced complexity compared to the optimal one. Having such detectors would enable far better use of resources leading to increased system capacity, coverage, and connection quality, all by upgrading the handsets used.
In the thesis, we only consider improvements related to the physical layer processing and, more specifically, processing of signals in the detector and its effects on objective performance measures such as the Bit Error Rate. Beside examining the performance improvements of detection methods, the computational complexity of the methods is also taken into account since this is often the limiting factor in an actual implementation. Thus, it might be that a method is abandoned due to its excessive complexity, even though it achieves huge performance gains.

1.1 Thesis Outline and Contributions

Chapter 2, Preliminaries, gives an introduction to the cellular system used in wireless communications and a general system model is presented. Next, the optimal sequence and symbol-by-symbol detectors are treated. Additionally, some near-optimal detection techniques are presented.

Chapter 3, Detection using QL-factorization, presents detection methods that are based on QL-factorization of the channel matrix. Firstly, the basic idea behind the Sphere Detector is described and it is shown how minimum-phase prefiltering can reduce the complexity of sphere detection in frequency-selective channels. Secondly, a proof that connects the minimum-phase and all-pass filters to the QL-factorization of the above-mentioned channel-type is presented. This leads to a novel approach to compute these two classical filters iteratively. The convergence rate for the iterative method is analyzed and a computationally efficient method for obtaining the filters is suggested.

Chapter 4, Sampling, describes a Markov Chain Monte Carlo detector, which uses Gibbs sampling to perform approximate (near-optimal) Maximum Likeli-

[3] In this thesis channel encoding is not considered, but if it should be taken into account the complexity of the optimal detector would increase even further, see e.g. [2].

hood Sequence Detection in Multiple-Input Multiple-Output systems having a huge number of receive and transmit dimensions. The novelty of the proposed Gibbs sampler is that it will, unlike simulated annealing techniques, use a fixed "temperature" parameter in all the iterations. This leads to the property that after the Markov chain has mixed, the probability of finding the optimal solution is polynomially rather than exponentially small.

Chapter 5, Conclusion, summarizes the thesis and presents some suggestions for interesting future research directions.

Contributions

Paper A, On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation, examines prefiltering techniques for sphere detection in frequency-selective channels. It is shown that it is possible to regard sphere detection as a generalization of traditional reduced-state sequence estimation. Further, simulations illustrate that minimum-phase prefiltering can reduce the complexity of sphere detectors significantly and still obtain near-optimal performance.

Paper B, Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization, contains a novel approach for computing both the minimum-phase filter and the associated all-pass filter in a computationally efficient way using fast QL-factorization. A desirable property of this approach is that the complexity is independent of the size of the matrix being QL-factorized. Instead, the complexity scales with the required precision of the filters. In order to evaluate the applicability of the method, simulations for communication channels used in the GSM system have been made, where the numerical effects of the method has been examined.

Paper C, Computing the Minimum-Phase Filter using the QL-Factorization, proves that the QL-factorization of a time-invariant multipath channel matrix gives the finite length equivalent to the minimum-phase and the all-pass filters and, thereby, it presents a novel method of computing these two classical filters in a numerically stable way. The convergence properties of this method is also analyzed such that the exact convergence rate has been computed for a simple SISO length $L = 2$ system and an upper bound has been derived, which is used for approximating the convergence in systems of arbitrary length.

Paper D, Near-Optimal Detection in MIMO Systems using Gibbs Sampling, describes a method for Maximum Likelihood Sequence Detection using a Markov Chain Monte Carlo Gibbs sampler. The method is novel in that

the "temperature" parameter is optimized so that in steady state, i.e. after the Markov chain has mixed, there is only polynomially (rather than exponentially) small probability of encountering the optimal solution.

Chapter 2

Preliminaries

This Chapter introduces some of the basic concepts which will be used throughout the rest of the thesis.

2.1 Cellular Systems

In order to derive efficient detection methods we first of all need a proper model, which can describe the received signal and, therefore, we first take a brief look at the cellular systems used for wireless communications. A general description of the wireless medium are treated in a vast number of books on wireless communications. Thus, for a more thorough treatment of this matter, the reader is referred to e.g. [1, 3–5] and the references therein.
A mobile network is divided into cells (hereby the name *Cellular System*) in order to provide coverage for the Mobile Station (MS). In the Global System for Mobile communications (GSM) both Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) are used, which enable network access to multiple subscribers at the same time. Each cell has a specific frequency for wireless communication but due to the limited resources in the frequency band, the available frequencies are being reused in other cells. However, the drawback of frequency reuse is occurrence of Co-Channel Interference (CCI).

When a signal is transmitted over a wireless medium, the effects of the surrounding environment will often lead to reflections of the signal, such that the MS will receive multiple copies of the same signal arriving at different time instances and having different attenuation levels. This leads to the concept of multipath channels. In cases where the delay time of the reception of the multiple copies of the signal is significant compared to the symbol interval, the transmitted symbols will affect each other, such that we get Inter-Symbol Interference (ISI). Due to the multiple paths of the signal, it can either add up constructively or destructively. This will lead to fading that depends on the signal wavelength (and thereby also the frequency) which is called frequency-selective fading. Likewise, in the case where the MS moves toward or away from the Base Station (BS), the fading will depend on the time and we then have time-selective fading [1].

2.2 System Model

In order to model the effects of the cellular systems, we introduce a general system model, which will be used throughout the thesis. A widely used channel model in wireless communication is the linear model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\upsilon} \ , \tag{2.1}$$

where $\mathbf{y} \in \mathbb{C}^M$ is the received signal, $\mathbf{H} \in \mathbb{C}^{M \times N}$ represents the channel matrix, and $\mathbf{x} \in \Omega^N$ it the transmitted symbols from alphabet Ω. The noise term $\boldsymbol{\upsilon} \in \mathbb{C}^M$ represents the thermal noise plus interference (from other users). In the case with no interference, we will assume that the noise term is simply Additive White Gaussian Noise (AWGN),i.e. $\boldsymbol{\upsilon} \sim \mathcal{CN}\left(\mathbf{0}_{M\times 1}, \sigma_\upsilon^2 \mathbf{I_M}\right)$. Both pulse-shaping and the receive filtering can be incorporated in the channel matrix since these are usually also linear operations.

In the case of a time-invariant Multiple-Input Multiple-Output (MIMO) system with a Finite Impulse Response (FIR) of length L, we can express the system model as

$$\mathbf{y}_j = \sum_{l=0}^{L-1} \mathbf{H}_l \mathbf{x}_{j-l} + \boldsymbol{\upsilon}_j \ , \tag{2.2}$$

where $\mathbf{y}_j \in \mathbb{C}^{N_R}$ is the received signal at time index j and $\mathbf{x}_j \in \Omega^{N_T}$ is the input signal at time $j = \{1, 2, \ldots, J\}$. J is the length of the transmitted sequence while N_R and N_T denote the receive and transmit dimensions, respectively. The matrix $\mathbf{H}_l \in \mathbb{C}^{N_R \times N_T}$ denotes the lth tap in the impulse response and $\boldsymbol{\upsilon}_j \in \mathbb{C}^{N_R}$ is the noise term, $\boldsymbol{\upsilon}_j \sim \mathcal{CN}\left(\mathbf{0}_{N_R\times 1}, \sigma_\upsilon^2 \mathbf{I_{N_R}}\right)$. The system model in

(2.1) is capable of modeling a multipath channel by letting the channel matrix $\mathbf{H}$ be a block-banded block Toeplitz matrix having the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_{L-1} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{H}_0 \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{L-1} \end{bmatrix} . \tag{2.3}$$

Here, each sub-matrix $\mathbf{H}_i$ has the size $N_R \times N_T$ and the matrix $\mathbf{H}$ will have the size $M = N_R(J + L - 1)$ and $N = J \cdot N_T$. If the channel is time-variant, the channel matrix will still be a block banded matrix, but it will no longer be block Toeplitz, since it will now change for each block-row of $\mathbf{H}$. In the case with no multipath effect, the channel will purely be a MIMO channel with $M = N_R$ and $N = N_T$.

The Signal-to-Noise Ratio (SNR) is defined as

$$\text{SNR} \triangleq \frac{\mathsf{E}\left\{\|\mathbf{Hx}\|_2^2\right\}}{\mathsf{E}\left\{\|\boldsymbol{v}\|_2^2\right\}} = \frac{\mathsf{E}\left\{\text{tr}\left(\mathbf{H}^H\mathbf{H}\mathbf{x}\mathbf{x}^H\right)\right\}}{\mathsf{E}\left\{\text{tr}\left(\boldsymbol{v}\boldsymbol{v}^H\right)\right\}} . \tag{2.4}$$

Given the situation where we transmit N independent symbols with the average symbol power σ_x^2 in a MIMO channel with no multipath and the noise term $\boldsymbol{v} \sim \mathcal{CN}\left(\mathbf{0}_{M\times 1}, \sigma_v^2\mathbf{I_M}\right)$, the SNR in (2.4) can be simplified to

$$\text{SNR} = \frac{\mathsf{E}\left\{\sigma_x^2\text{tr}\left(\mathbf{H}^H\mathbf{H}\right)\right\}}{\sigma_v^2 N_R} . \tag{2.5}$$

2.3 Optimal Detection

In this section we introduce the optimality criteria, which we often encounter in detection in wireless communications. Also, we briefly describe how optimal detection can be achieved. The equations below have been derived under the assumption that the noise is AWGN.

2.3.1 Optimal Sequence Detection

In the decoding part we are often interested in finding the most likely sequence,

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x}\in\Omega^N} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \ , \tag{2.6}$$

which is called Maximum Likelihood Sequence Detection (MLSD) and often abbreviated as Maximum Likelihood (ML) detection for simplicity. The optimization problem in (2.6) is considered to be NP-hard both in worst-case and in average sense for general $\mathbf{H}$ [6–8] when $\mathbf{x}$ belongs to a discrete symbol set. Thereby, the complexity seems at first glance to scale exponentially with the size of vector $\mathbf{x}$, i.e. Ω^N. However, in multipath channels clever dynamical programing can be applied (such as the Viterbi algorithm[1] [10, 11] a.k.a. the max-sum algorithm [12]), and the complexity will instead scale as $\Omega^{N_T \cdot L}$, which in communication systems usually is a *huge* reduction in complexity. As an example we can mention GSM where $N = 122$ (in the case of Single User Detection (SUD) where the training sequence and the tail bits has been excluded), which should be compared to $L = 7$ for the Typical Urban (TU) profile or $L = 10$ for the Hilly Terrain (HT) profile defined in [13]. Furthermore, in Multi User Detection (MUD) the complexity gets worse still, since the number of users will also influence the complexity exponentially.

MLSD using the Viterbi-algorithm

Since an extension to the above mentioned dynamical programing techniques is treated in Chapter 3, we briefly describe how the Viterbi-algorithm works and how it is capable of reducing the complexity in case of a multipath channel. For simplicity we only consider a Single-Input Single-Output (SISO) system, i.e. $N_T = N_R = 1$, and we assume we have a time-invariant channel.[2] Let us

[1] The forward-only max-log BCJR-algorithm [9] can also be used for optimal sequence detection.

[2] It is straightforward to extend the algorithm to the general time-variant MIMO system.

first rewrite the optimization problem given in (2.6) by using the signal model in (2.2)

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x}\in\Omega^N} \sum_{j=1}^{M} \left\| y_j - \sum_{l=0}^{L-1} h_l x_{j-l} \right\|_2^2 . \tag{2.7}$$

From this it can be seen that the cost function in (2.6) that we are minimizing can be computed recursively and, thereby, reducing the complexity.

2.3.2 Optimal Symbol-by-Symbol Detection

In case we are interested in optimal symbol-by-symbol detection, we instead maximize the a-posteriori probability of the symbols

$$\hat{x}_{k,\mathrm{MAP}} = \arg\max_{x_k\in\Omega} \mathrm{P}\left(x_k \,|\mathbf{y}, \mathbf{H}\right) , \tag{2.8}$$

which is also called Maximum A-Posteriori (MAP) detection and involves a marginalization over all possible symbol settings in $\mathbf{x}_{\backslash k}$. This will require evaluation of Ω^N probabilities but once more dynamical programming can be exploited for multipath channels by applying the forward-backward algorithm [14, 15] (a.k.a. the BCJR-algorithm [9] or the sum-product algorithm [12]). Thus, for SUD the complexity of this is again of order $\Omega^{N_T \cdot L}$. It should be mentioned that in communications systems where the detection stage is succeeded by a decoding stage, we will usually be more interested in the symbol probability $\mathrm{P}\left(x_k \,|\mathbf{y}\right)$ (a.k.a. the soft-symbol) instead of the "hard-symbol" given in (2.8).

2.4 Approximate Detection

For higher order modulation types, a complexity of order $\Omega^{N_T \cdot L}$ is still much too complex in the existing GSM system, e.g. for 16- and 32-QAM (Quadrature Amplitude Modulation), which has been standardized in EGPRS2 [16] to increase data rates. Instead approximate (sub-optimal) detection methods are applied in such systems and this has been the main motivation for considering approximate detection in this thesis. There exists a huge number of approximate methods, so an overall treatment of each of these is out of the scope.
In Chapter 3 Sphere Detection (SD) is described, which can either be used for approximate or exact detection. The kind of problems that we are considering will often (due to the complexity constraints in an implementation) imply that it is only feasible to perform approximate SD and, hence, we will focus on this. Another type of approximate detection is based on Markov Chain Monte

Carlo and we will treat that in Chapter 4. In the following we also describe some approximate methods which are widely used in systems with multipath channels.

2.4.1 Decision Feedback Techniques

Several approximate methods rely on some sort of feedback in the receiver. The most simple one is called Decision-Feedback Estimation (DFE) [17, 18] and it simply feeds back a weighted sum of past estimated symbols in order to deal with the ISI. To improve the performance of the DFE, the Delayed Decision-Feedback Sequence Estimation (DDFSE) [19] was introduced. This broadly speaking, consists of a parameter ξ that can be varied between 0 and the length of the Channel Impulse Response (CIR), L. If $\xi = 0$ the DDFSE reduces to the DFE detector, while $\xi = L$ corresponds to the Viterbi-algorithm, and the complexity is generally of order $\Omega^{N_T \cdot \xi}$. In cases where the cardinality of the alphabet is large, the Reduced-State Sequence Estimation (RSSE) [20] can be applied. The RSSE will, besides the DDFSE part, also employ set-partitioning [21] such that the constellation points are partitioned into subsets, and the complexity will now scale exponentially in the number of subsets instead of the size of the alphabet.
In order to obtain decent performance of the above mentioned feedback techniques for channels with large delay spread, prefiltering of the received signal is needed [19] such that the CIR is transformed into a minimum-phase filter [22–24]. Thus, the received signal will often be prefiltered with an all-pass filter as explained in Subsection 2.5 in order to obtain the desired minimum-phase characteristic for the CIR. This leads to the system illustrated in Figure 2.1.

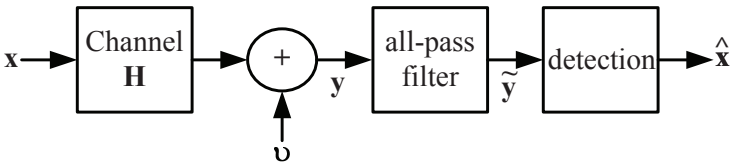


Figure 2.1: System model with prefilter and detection stage included.

2.5 The Minimum-Phase Filter

The spectral factorization theorem states that the spectrum of any linear time-invariant system can be factorized into minimum-phase components [25]. Furthermore, a generalization of spectral factorization states that any linear filter can be split into an all-pass filter and a minimum-phase filter found by spectral factorization [22]. The minimum-phase filter has the convenient property that it provides the highest possible energy concentration in the first filter taps (**??**). This filter is therefore crucial if the performance of the suboptimal trellis-based detectors (such as DDFSE and RSSE) should be close to the optimal one. Otherwise, if the channel impulse response is not minimum-phase, the suboptimal detectors are required to have a complexity close to the optimal one in e.g. the GSM system. This is due to the so-called C_0 pulse shape which is used in GSM [13,26] and shown in Figure 2.2. From Figure 2.2 it is clear that it is nec-

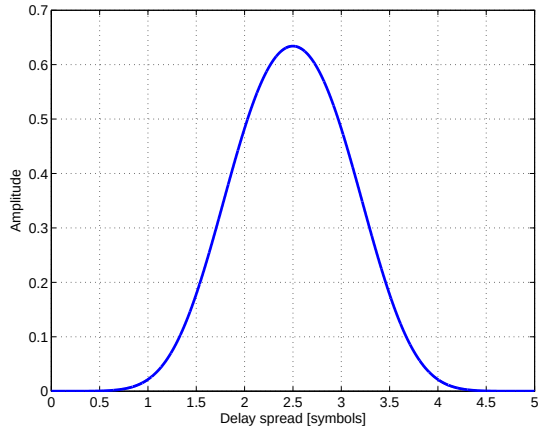


Figure 2.2: Pulse shape of C_0-pulse used in GSM.

essary to have a memory of length four in the trellis-based detectors in order to capture the most of the energy of the pulse, which for higher-order modulation types is a too high complexity. As an example we can mention EGPRS2 with $|\Omega| = 32$ giving approximately $3.3 \cdot 10^4$ states in the trellis diagram. Thus, the minimum phase filter is absolutely necessary in these types of applications.
Due to the broad applicability of the minimum-phase filter it has been studied intensively over the years and, thus, there exist various methods for computing the filter. In [25] and the references therein, a thorough treatment of several methods for spectral factorization can be found. One classical way of obtaining the minimum-phase filter is by using the root-method of spectral factorization,

in which the roots of

$$\mathbf{H}(z) = \sum_{l=0}^{L-1} \mathbf{H}_l z^{-l} \,, \tag{2.9}$$

located outside the unit circle are reflected inside to the conjugate reciprocal location [22, 23, 27]. Here, $\mathbf{H}(z)$ represents the z-transform of the equivalent infinite-length filter impulse response, which is connected with the finite-length system in (2.3) when we let the system size $J \to \infty$. The connection between the finite- vs. the infinite-length system is among others treated in [28]. The polynomial form described in (2.9) can be a useful representation in the analysis of the filter characteristics.
The simple root-method of spectral factorization has however its limitations. Particularly, in the case of MIMO systems since we besides the roots also need to know the direction of the "basis" vector associated with a root [24]. Some methods for solving the problem in this case have been described in among others [29–32], but these methods have the disadvantages of being mathematically rather complicated and, furthermore, they can suffer from numerical instabilities [24]. Thus, instead one might prefer to solve a Discrete-time Algebraic Riccati Equation (DARE), which is a numerical stable method. It has the particularly advantageous property that it easily can be extended to the vector case [25]. In the following, we briefly describe how the roots can be determined.

2.5.1 The root-method of spectral factorization

Let us for a moment assume that we are only interested in determining the roots of $\mathbf{H}(z)$ in (2.9). In a MIMO system where $N_T = N_R$, the roots can be obtained by finding the z-values where $\det(\mathbf{H}(z)) = 0$, [33], leading to a matrix polynomial in the scalar variable z. This type of matrix polynomial is normally called a lambda-matrix [34, 35] and the number of roots in such a polynomial is $\min(N_T, N_R) \cdot (L-1)$. In [34], it is shown that the roots can be obtained by determining the eigenvalues of the block-companion matrix, $\mathbf{C}$, of the associated *monic* polynomial. This can be obtained by $\tilde{\mathbf{H}}(z) \triangleq (\mathbf{H}_{L-1})^{-1}\mathbf{H}(z)$, assuming that $\mathbf{H}_{L-1}$ is invertible. Thus, we get the following block-companion matrix

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & -\tilde{\mathbf{H}}_0 \\ \mathbf{I} & \ddots & \vdots & -\tilde{\mathbf{H}}_1 \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{I} & -\tilde{\mathbf{H}}_{L-2} \end{bmatrix}, \tag{2.10}$$

where $\tilde{\mathbf{H}}_l \triangleq (\mathbf{H}_{L-1})^{-1}\mathbf{H}_l$. Since the method proposed in [34, 35] assumes that all $\mathbf{H}_l$ terms are square matrices, we cannot directly handle the case where $N_T \neq N_R$. As a consequence a modifcatoin of the problem is needed. If $N_R > N_T$,

we can instead introduce $\mathbf{S} = \mathbf{H}^H\mathbf{H}$ and find the roots of the lambda-matrix $\mathbf{S}(z)$,

$$\begin{aligned}\mathbf{S}(z) &= \mathbf{H}^H(z^{-*})\cdot\mathbf{H}(z) \\ &= \sum_{l=0}^{L-1}\mathbf{S}_l\cdot z^{-l} + \sum_{k=1}^{L-1}\mathbf{S}_k^H\cdot z^k \quad ,\end{aligned} \tag{2.11}$$

which gives the roots both inside and outside the unit circle (from the minimum- and maximum-phase filter, respectively). Likewise, we can construct $\tilde{\mathbf{S}} = \mathbf{H}\mathbf{H}^H$ when $N_R < N_T$ and once more select the roots inside the unit circle. However, this does not solve the problem of finding the zero directions and, therefore, we will also address an alternative way of computing the spectral factor.

2.5.2 The DARE Method

As mentioned in Subsection 2.5.1, the DARE method has the convenient property that it is straight forward to extend the method from the Single-Input Single-Output (SISO) case to the MIMO case. Furthermore, the method relates to results from Kalman filtering theory and, therefore, many of the properties of this method have been extensively studied, among others its convergence properties which are treated in [24].
The DARE method considered in this thesis, solves the Riccati equation

$$\mathcal{P} = \mathcal{F}\mathcal{P}\mathcal{F}^H - \left(\mathcal{F}\mathcal{P}\mathcal{H}^H + \mathcal{G}\right)\left(\mathcal{H}\mathcal{P}\mathcal{H}^H + \mathbf{S}_0\right)^{-1}\left(\mathcal{F}\mathcal{P}\mathcal{H}^H + \mathcal{G}\right)^H \quad , \tag{2.12}$$

where we have assumed that $N_R = N_T$. $\mathcal{F}$ represents a block-shifting matrix of dimension $(L-1)N_T \times (L-1)N_T$ having the form

$$\mathcal{F} = \begin{bmatrix} \mathbf{0}_{N_T\times N_T} & \cdots & \cdots & \mathbf{0}_{N_T\times N_T} \\ \mathbf{I}_{N_T} & \ddots & \ddots & \vdots \\ \mathbf{0}_{N_T\times N_T} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{I}_{N_T} & \mathbf{0}_{N_T\times N_T} \end{bmatrix} .$$

$\mathcal{G}$ is a matrix of size $(L-1)N_T \times N_T$ containing

$$\mathcal{G} = \begin{bmatrix} \mathbf{S}_L \\ \mathbf{S}_{L-1} \\ \vdots \\ \mathbf{S}_1 \end{bmatrix} ,$$

and $\mathcal{H} = \begin{bmatrix} \mathbf{0}_{N_T\times N_T}, & \mathbf{0}_{N_T\times N_T}, & \cdots, & \mathbf{1}_{N_T\times N_T} \end{bmatrix}$. If the iterative procedure described in [36] is used to solve the Riccati equation, we have

$$\mathcal{P}_{k+1} = \mathcal{F}\mathcal{P}_k\mathcal{F}^H - \left(\mathcal{F}\mathcal{P}_i\mathcal{H}^H + \mathcal{G}\right)\left(\mathcal{H}\mathcal{P}_k\mathcal{H}^H + \mathbf{S}_0\right)^{-1}\left(\mathcal{F}\mathcal{P}_k\mathcal{H}^H + \mathcal{G}\right)^H , \tag{2.13}$$

where $\mathcal{P} = \lim_{k\to\infty} \mathcal{P}_k$ and $\mathcal{P}_0 = \mathbf{0}_{N_T(L-1)\times N_T(L-1)}$. This results in the stability matrix (i.e. eigenvalues smaller than 1 in magnitude)

$$\mathcal{F} - \left(\mathcal{F}\mathcal{P}\mathcal{H}^H + \mathcal{G}\right)\left(\mathcal{H}\mathcal{P}\mathcal{H}^H + \mathbf{S}_0\right)^{-1} . \tag{2.14}$$

As described in [37], the minimum-phase filter coefficients can be computed based on the stabilizing solution as

$$\mathbf{H}_{mp,DARE}(z) = \begin{bmatrix} \left(\mathbf{S}_0 + (\mathcal{P})_{(N_1:N_2,N_1:N_2)}\right)^{\frac{1}{2}} \\ \left(\mathbf{S}_0 + (\mathcal{P})_{(N_1:N_2,N_1:N_2)}\right)^{-\frac{1}{2}} \left(\mathcal{F}\mathcal{P}\mathcal{H}^H + G\right)^H \mathbf{J} \end{bmatrix}^H \begin{bmatrix} \mathbf{1}_{N_T\times 1} \\ z^{-1}\mathbf{1}_{N_T\times 1} \\ \vdots \\ z^{-L+1}\mathbf{1}_{N_T\times 1} \end{bmatrix} \tag{2.15}$$

where $\mathbf{J} \in \mathbb{R}^{N_T(L-2)\times N_T(L-2)}$ is an anti-diagonal matrix with ones on the anti-diagonal. Furthermore, we have for simplicity introduced $N_1 \triangleq N_T(L-2)+1$ and $N_2 \triangleq N_T(L-1)$.
The complexity of computing the minimum-phase filter of a length L SISO system, using the DARE method described above, requires

$$\mathcal{O}_{min,DARE} = k\left(\frac{3}{2}L^2 - \frac{1}{2}L + 1\right) + 2L \tag{2.16}$$

operations, where k denotes the number of iterations used for computing the filter and where we define an operation as a complex Multiply-Accumulate (MAC) instruction.

2.6 Summary

This chapter has treated some of the effects which occur in cellular systems. Based on these, a general system model has been introduced which is capable of modeling multipath and multiuser MIMO systems. Next, optimal sequence detection and symbol-by-symbol detection has been described and some computationally efficient dynamical programming methods have briefly been mentioned. To reduce the complexity of the detection stage, some approximate feedback techniques have been proposed. It has been explained that to achieve decent performance of these techniques, both the minimum-phase filter and the associated all-pass filter (which is used for prefiltering the received signal) play a crucial role. Finally, some general properties of the minimum-phase filter has been described, among others how this filter can be computed and the complexity of this.

Chapter 3

Detection using QL-factorization

In this Section we treat algorithms which exploit the QR- or the QL-factorization in order to obtain efficient low complexity detection methods. First, the principle behind the Sphere Detection (SD) algorithm is described and next, a connection between the minimum-phase filter and the QL-factorization is proved. This connection is then used to obtain a novel approach to compute the minimum-phase filter and its associated prefilter using fast QL-factorization methods.
The Chapter is based on results which have originally been presented in the papers given in the Appendices A, B, and C.

3.1 Sphere Detection

The problem of performing MLSD using Sphere Detection has gained much attention over the years [6,38–44]. This Section briefly describes the SD algorithm and shows how the MLSD can be achieved computationally efficient. Furthermore, it is described how SD can be applied in frequency-selective channels and how the effect of imperfect channel estimation can be taken into account when the radius of the sphere is determined.

The Sphere Detection algorithm address the problem of finding the *integer least-squares solution* in a clever way. This corresponds to solving the least-squares problem when the unknown vector consists of integer elements. This is sometimes also referred to as the *closest lattice point problem* [40,41]. In general, the integer least-squares problem can be expressed as

$$\arg \min_{\mathbf{x} \in \mathbb{Z}^N} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \ , \tag{3.1}$$

where $\mathbb{Z}$ represents a N-dimensional integer vector. From (3.1), we see that it is equivalent with MLSD if we minimize over Ω instead of $\mathbb{Z}$. As mentioned in Section 2.3, the general integer least-squares problem is much more complicated to solve than the "standard" least squares problem and, thus, if the received signal, $\mathbf{y}$, is arbitrary, the expected complexity will be exponential [6]. However, when the received point is a lattice point distorted by additive Gaussian noise, it has been shown in [6] that the expected complexity tends to behave polynomially over a wide range of SNRs. The complexity will, however, still be exponential for cases where we have low SNR or where the number of receive- and transmit-dimensions are huge [45]. We will first consider the general case where there is no structure in $\mathbf{H}$, but since $\mathbf{H}$ often has a certain structure (e.g. Toeplitz) in many wireless systems, we also address this in the next Subsection.

The integer least-squares problem can geometrically be represented as finding the closest lattice point in a skewed lattice, $\mathbf{H}\mathbf{x}$, based on the received signal, $\mathbf{y}$ [6]. Thus, the basic idea behind the SD algorithm is simply to solve the MLSD problem by only searching over a limited number of lattice points inside a sphere around the received point as illustrated in Figure 3.1.

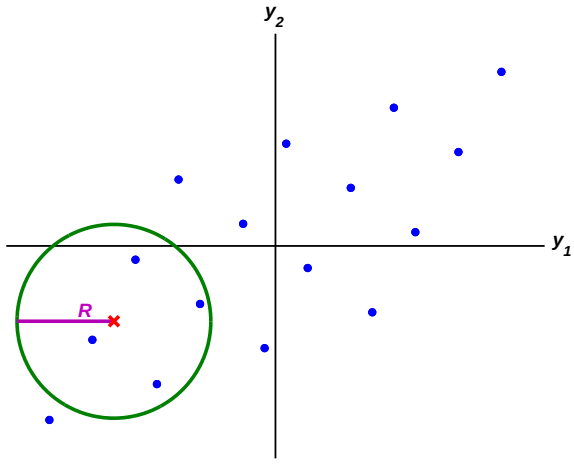


Figure 3.1: Principle of SD with radius, R, for the 2-dimensional case.

Even though the idea behind the SD is quite simple, we need to address two issues in order to design an efficient algorithm. Firstly, we need a method to determine which points are inside the sphere without computing distance from the received point to each of the lattice points (this will not get rid of the the exponential behavior). Secondly, we also need a scheme for choosing the radius, R, of the sphere. This might seem like a trivial thing, but it turns out that the size of the radius will have a significant impact on the complexity of the algorithm [46–48]. If the radius is too small there will be no lattice point inside the sphere, while a too large radius will include too many points and the complexity will still be exponential. A natural choice of the radius would be the *covering radius* of the lattice, which is defined as *"the smallest radius of spheres centered at the lattice points that cover the entire space"* [6]. However, the problem of finding the covering radius is also exponential in complexity [6] and, therefore, there have been several investigations on selecting the radius properly, in example see [46, 48, 49] and the references therein.

In order to determine if a point is inside the sphere efficiently, we perform a QL-factorization (or QR-factorization) of the channel matrix, $\mathbf{H} = \tilde{\mathbf{Q}}\tilde{\mathbf{L}}$. Based on the QL-factorization we can get a new equivalent system equation by multiplying the system equation in (2.1) by $\tilde{\mathbf{Q}}^H$ such that

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\upsilon} = \tilde{\mathbf{Q}}\tilde{\mathbf{L}}\mathbf{x} + \boldsymbol{\upsilon} \;\Leftrightarrow \tag{3.2a}$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{L}}\mathbf{x} + \tilde{\boldsymbol{\upsilon}} \;, \tag{3.2b}$$

where we have used the fact that $\tilde{\mathbf{Q}} \in \mathbb{C}^{M\times M}$ is a unitary matrix and defined $\tilde{\mathbf{y}} \triangleq \tilde{\mathbf{Q}}^H\mathbf{y}$ and $\tilde{\boldsymbol{\upsilon}} \triangleq \tilde{\mathbf{Q}}^H\boldsymbol{\upsilon}$. Importantly, it also follows from unitarity that the noise statistic is not changed (under the assumption that we have Gaussian noise). In order to ensure a unique factorization we have

$$\tilde{\mathbf{L}} \triangleq \begin{bmatrix} \mathbf{0}_{(M-N)\times N} \\ \mathbf{L} \end{bmatrix} , \tag{3.3}$$

where $M \geq N$ and where we require that the $N \times N$ lower triangular matrix, $\mathbf{L}$, corresponds to the Cholesky factor of $\mathbf{H}^H\mathbf{H}$. This implies that $\mathbf{L}$ is positive definite and thus contains real-valued positive diagonal elements (assuming that $\text{rank}(\mathbf{H}) = N$). Using (3.2) we can rewrite the optimization problem in (2.6) to

$$\hat{\mathbf{x}}_{\text{ML}} = \arg\min_{\mathbf{x}\in\Omega^N} \quad \|\tilde{\mathbf{y}} - \tilde{\mathbf{L}}\mathbf{x}\|_2^2 \tag{3.4a}$$

$$= \arg\min_{\mathbf{x}\in\Omega^N} \quad \|\mathbf{Q}^H\mathbf{y} - \mathbf{L}\mathbf{x}\|_2^2 + \|\mathbf{Q}_0^H\mathbf{y}\|_2^2 \;, \tag{3.4b}$$

where $\tilde{\mathbf{Q}} \triangleq [\mathbf{Q}_0, \mathbf{Q}]$ and $\mathbf{Q}_0$ contains the first $M - N$ columns of $\tilde{\mathbf{Q}}$, while $\mathbf{Q}$ represents the remaining N columns. Since we are optimizing over $\mathbf{x}$, we can disregard the latter term on the LHS in (3.4b) (or simply collect this term in

the radius constraint shown below). The lower triangular structure in $\mathbf{L}$ has the advantageous property that we can examine a single dimension at a time and, thereby, we can design an efficient detection algorithm. If a point lies inside (or on the boundary of) a sphere centered at the received point, $\acute{\mathbf{y}} \triangleq \mathbf{Q}^H\mathbf{y}$, with the radius, R, it has to satisfy

$$R^2 \geq \|\acute{\mathbf{y}} - \mathbf{L}\hat{\mathbf{x}}\|_2^2 \ . \tag{3.5}$$

This imply that it certainly has to fulfill the constraint

$$R^2 \geq |\acute{y}_1 - (\mathbf{L})_{1,1}\, x_1|^2 \ ,$$

and, thereby, we can construct boundaries for the symbol such that

$$\frac{-R + \acute{y}_1}{(\mathbf{L})_{1,1}} \leq \hat{x}_1 \leq \frac{R + \acute{y}_1}{(\mathbf{L})_{1,1}} \ . \tag{3.6}$$

We can then tighten the constraint in order to bound the next symbol based on our knowledge of $\hat{x}_1$:

$$\frac{-R_{2|1} + \acute{y}_{2|1}}{(\mathbf{L})_{2,2}} \leq \hat{x}_2 \leq \frac{R_{2|1} + \acute{y}_{2|1}}{(\mathbf{L})_{2,2}} \ ,$$

in which we have used $R_{2|1}^2 \triangleq R^2 - |\acute{y}_1 - (\mathbf{L})_{1,1}\,\hat{x}_1|^2$ and $\acute{y}_{2|1} \triangleq \acute{y}_2 - (\mathbf{L})_{2,1}\,\hat{x}_1$. This procedure can be repeated for each dimension until we have reached the Nth dimension.[1]

The radius of the sphere can be selected based on the noise statistics, such that

$$\|\tilde{\boldsymbol{\upsilon}} - \tilde{\mathbf{L}}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 \approx \|\tilde{\boldsymbol{\upsilon}}\|_2^2 \ , \tag{3.7}$$

where we have combined (3.5) and (3.2). The approximation in (3.7) is valid when the ML is in fact the transmitted sequence (i.e. when $\mathbf{x} = \hat{\mathbf{x}}$). Given that the noise is AWGN, the squared 2-norm of $\boldsymbol{\upsilon}$ is a Chi-Square distribution with M complex degrees for freedom, $\|\boldsymbol{\upsilon}\|^2 \in \boldsymbol{\chi}_M^2$. The probability of having a point inside the sphere (here denoted as $1-\varepsilon$) can, therefore, be computed by

$$\mathrm{P}\left(\boldsymbol{\chi}_M^2 \leq R^2\right) = 1 - \varepsilon \ .$$

This corresponds to evaluating the inverse Chi-Square distribution and can be implemented by look-up tables for the distribution [6, 48]. As indicated earlier, the SD algorithm can turn out to be very computationally complex at low SNR and for high dimensional lattice points, which is due to a very loose bounding caused by the radius. Instead, we can use increasing radii in the SD algorithm, which often gives a huge reduction in complexity but it comes with the cost of no longer guaranteeing the ML solution. However, in [46] a statistically sound method for computing the increasing radii, which can provide a performance that comes arbitrarily close to the ML solution, has been presented.

[1] It is assumed that the channel matrix has $\mathrm{rank}(\mathbf{H}) = N$. If the rank is lower than this, we can only apply SD until we have reached $\mathrm{rank}(\mathbf{H})$, and we would then have to perform exhaustive search in the remaining dimensions to obtain the ML solution.

3.1.1 SD in Multipath Channels

In situations where we have a multipath channel, we can apply the SD algorithm without performing a full QL-factorization of $\mathbf{H}$ due to the lower block triangular form of the channel matrix in (2.3). As it is shown in [2, 47, 48], we can achieve a lower triangular matrix $\tilde{\mathbf{H}}$ by the transformation

$$\tilde{\mathbf{H}} = \left(\mathbf{I} \otimes \mathbf{Q}_0^H\right) \mathbf{H} \ ,$$

where $\mathbf{I}$ is an identity matrix of size $(J + L - 1) \times (J + L - 1)$ and we have defined $\mathbf{Q}_0\mathbf{L}_0 \triangleq \mathbf{H}_0$. This simply corresponds to a factorization of each submatrix in $\mathbf{H}$ with $\mathbf{Q}_0$. Given a SISO system (i.e. a block size of one) we do not have to perform the QL-factorization at all since $\mathbf{H}$ already has a lower triangular structure.

The lower triangular structure in $\tilde{\mathbf{H}}$ makes it possible to combine the SD algorithm with the Viterbi algorithm as shown in [50], and the MLSD can be obtained by examining only the states in the trellis diagram, which lie inside the sphere, corresponding to a pruning of the trellis diagram. Alternatively, the SD algorithm can be combined with the MAP detector to obtain near-optimal symbol-by-symbol detection by forming approximate bit posteriors [38, 51].

To make the SD algorithm implementable in "real applications" it will sometimes be necessary to specify the maximum allowed complexity since it, in the worst case, is exponential. By using the Schnorr-Euchner search strategy [43], it is possible to set an upper limit on the number of states, which are allowed to be examined at each time index in the trellis diagram and, thereby, only search the most likely paths in the trellis diagram. Furthermore, it is possible to specify the maximum number of state transitions allowed from a given state. Both of these methods are, of course, suboptimal but can be a necessary compromise. The performance of these two suboptimal schemes will greatly depend on the CIR (among others, the numerical value of the diagonal elements in $\tilde{\mathbf{H}}$, since these appear in the denominator of (3.6), which are used for bounding of the symbols). The intuitive explanation for the connection between the bounding interval and the pulse shape of the CIR is that the decision of disregarding states in SD can first be made when a considerable extent of confidence has been obtained. In other words, this implies that the SD algorithm will first prune the trellis diagram when a certain amount of energy has been received from the transmitted symbol [47].

In Section 3.2 it will be proven that a QL-factorization of the multipath channel matrix will asymptotically correspond to prefilter the original CIR with an all-pass filter such that we obtain a minimum-phase filter on the detector side as illustrated in Figure 3.2. The Figure corresponds to the system model shown

in (3.2), which was obtained by the QL-factorization. It can be seen that it is possible to regard SD as a generalization of traditional reduced-state sequence estimation, providing a unifying framework for the two detection methods.

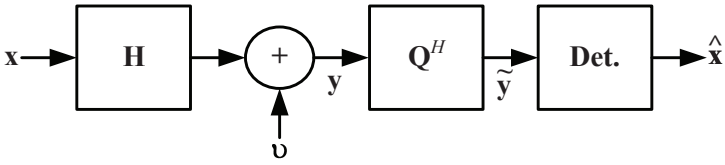


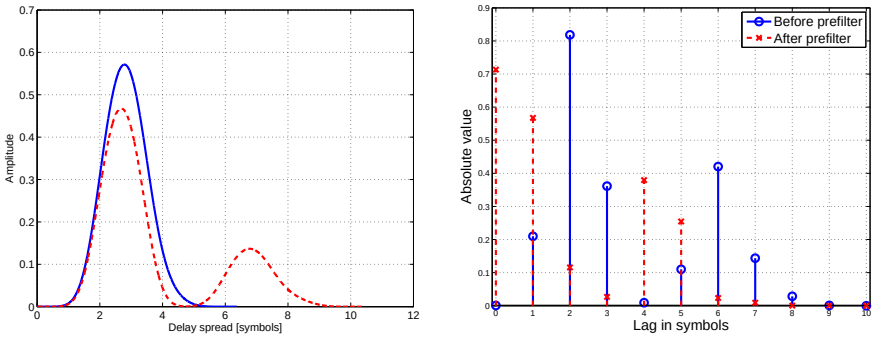
Figure 3.2: System model with prefilter and detection stage included.

As mentioned in Section 2.5, the minimum-phase filter has the convenient property of providing the highest possible energy concentration in the first taps. Thus, a full QL-factorization of the channel matrix can, potentially, reduce the computational complexity (compared to the decoding directly on $\tilde{\mathbf{H}}$) due to the earlier decision making in the trellis diagram. Furthermore, it will be shown in Section 3.3 that efficient methods for factorizing the channel matrix exist, which make it more likely that a complexity reduction can be achieved.

Simulations

In order to demonstrate the effect of the QL-factorization of the channel matrix, we present simulation results which originally were presented in [47]. The simulations are carried out for the EDGE system, having a frame format and modulation type identical to that specified in the EDGE standard [13], e.g. a $3\pi/8$ rotated 8-PSK signal is used. It is assumed that frequency hopping is made between each received burst and the CIR and noise variance are perfectly known. Only single user detection is considered in the simulations and only a single receive antenna is assumed to be available. Moreover, AWGN is added to account for any thermal noise. To exploit the diversity in the channel model, the oversampling factor in the channel is set to $N_{sps} = 2$ in respect to the symbol rate. Due to this oversampling, the received signal is jointly prefiltered before it is passed to the detector, leading to $N_R = 1$ in the detector. The channel models used in the simulations are the Typical Urban (TU0) and the Hilly Terrain (HT0) profiles defined in the GSM specifications [13].
The channel profiles of TU and HT, obtained by the convolution of the square root of the power delay profile with the transmit filter response (the so-called C_0-pulse in [26]), are shown in Figure 3.3a. 3.3b, an example of the channel coefficients of the HT profile is shown (N_{sps} is here set to 1). The coefficients obtained using a minimum-phase prefilter are also shown to illustrate the effect

of the filter. It is observed that the number of taps needed for modeling the channel properly is approximately $L = 7$ when there is no prefilter, while the channel length can be reduced to $L = 6$ using a minimum-phase prefilter. The optimal detector would in the latter case require a search in a trellis diagram of $8^{6-1} \approx 33 \cdot 10^3$ states per symbol, which is still an unacceptable high complexity and it will get worse still for the newly specified EGPRS2 standard, where modulation up to 32-QAM can be applied [16].

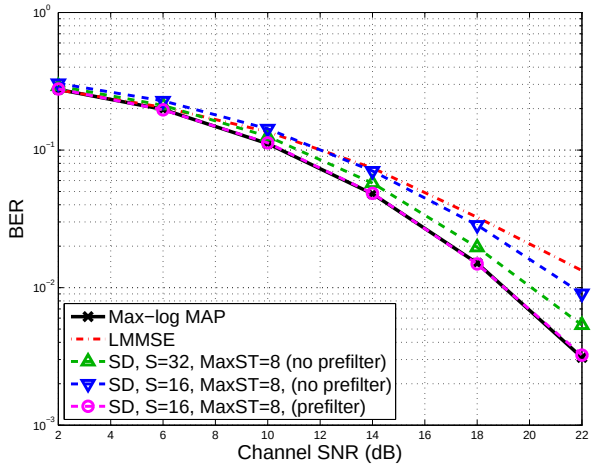


(a) The ensemble average of the pulse shape of the Typical Urban (TU) and the Hilly Terrain (HT) profiles (including the transmit pulse shaping).

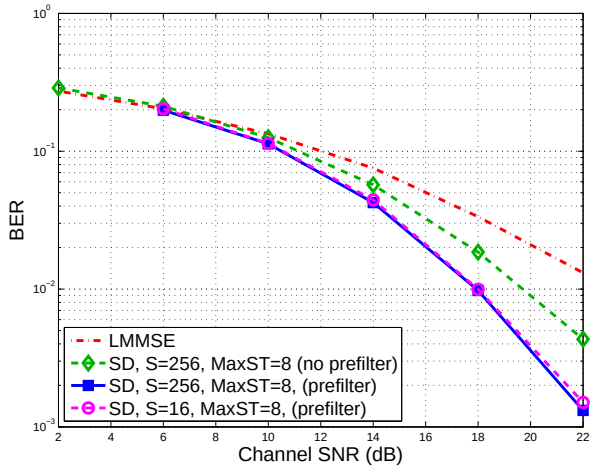
(b) The absolute value of filter coefficients for one realization of the HT profile with and without minimum-phase prefiltering.

Figure 3.3: Delay profile and single realization thereof for the TU and HT profiles in the EDGE system.

In the simulations SD has been combined with the max-log MAP receiver to obtain approximate bit posteriors and the increasing radii scheme has been used. The radii have been obtained from $\mathrm{P}\left(|\boldsymbol{v}_{1:n}|^2 > r_n^2\right) = \varepsilon^{2k}$, where k is the number of times the algorithm is restarted, which is done if no points are found inside the sphere. Furthermore, the approach of specifying the maximum number of allowed states in the trellis diagram has been used in all the simulations considering SD. The Bit Error Rates (BER)s for the two channel profiles have been plotted in Figure 3.4.
In Figure 3.4a, the BER performance of the proposed sphere detector is presented for the TU profile. To illustrate the effect of prefiltering, BER curves are given for the same simulation setup, but with and without minimum-phase prefiltering. In the labels, "S", denotes the maximum number of allowed *states* in the trellis diagram, while "MaxST" denotes the maximum number of allowed *state transitions* for a given state. In Figure 3.4a, the performance of the max-log MAP detector and the Linear Minimum Mean-Square Error (LMMSE) detector have also been included as references. The detector relying on minimum-phase prefiltered SD with at most 16 states in the trellis diagram is capable of obtaining a performance which is comparable with the max-log MAP. This is not the



(a) TU profile with and without prefiltering.



(b) HT profile with and without minimum-phase prefiltering.

Figure 3.4: BER performance for different channel profiles. S denotes the maximum number of states in the trellis diagram and MaxST is the maximum number of state transitions from a given state.

case when the prefilter is not used.
In Figure 3.4b, the performance of the detectors for the HT profile is shown. For this profile, it has not been possible to simulate the max-log MAP detector due to its huge complexity. From Figure 3.4b it is clear that prefiltering gives a significant improvement in BER performance. Furthermore, it is observed that the complexity can be reduced considerably without degrading the BER performance.

SD and Channel Uncertainty

When we do not have perfect knowledge of the CIR, which will be the case in wireless communication systems, this uncertainty of the channel estimation should be taken into account when we estimate the radius of the sphere. For simplicity we consider the case of SUD in a SISO system. We will assume AWGN and use the Least Squares (LS) estimate of the training sequence to determine the channel estimate $\hat{\mathbf{h}} \sim \mathcal{CN}\left(\mathbf{h}_{\mathrm{ML}}, \mathbf{\Sigma}_{\hat{\mathbf{h}}}\right)$ where we have

$$\mathbf{h}_{\mathrm{ML}} = \left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H\mathbf{y}_{TS} = \mathbf{X}^{+}\mathbf{y}_{TS} \tag{3.8a}$$

$$\mathbf{\Sigma}_{\hat{\mathbf{h}}} = \sigma_v^2\left(\mathbf{X}^H\mathbf{X}\right)^{-1} \ . \tag{3.8b}$$

In (3.8a), we have introduced the vector $\mathbf{y}_{TS} \in \mathbb{C}^{N_{TS}-L+1}$, which represents the received signal based on a transmission N_{TS} training symbols. Also, we have defined the matrix $\mathbf{X} \in \Omega^{(N_{TS}-L+1)\times L}$ containing the training symbols

$$\mathbf{X} \triangleq \begin{bmatrix} x_{N_L} & \cdots & x_1 \\ x_{N_{L+1}} & \cdots & x_2 \\ \vdots & \cdots & \vdots \\ x_{N_{TS}} & \cdots & x_{N_{TS}-L+1} \end{bmatrix} .$$

Assuming that the training set is designed such that all column vectors in $\mathbf{X}$ are orthonormal, we get from (3.8) that covariance matrix of size $L \times L$ will be

$$\mathbf{\Sigma}_{\hat{\mathbf{h}}} = \sigma_v^2\big(\left(N_{TS}-L+1\right)\mathbf{I}_L\big)^{-1} = \frac{\sigma_v^2}{N_{TS}-L+1}\mathbf{I}_L \ .$$

The estimated channel matrix can be expressed as $\hat{\mathbf{H}} \triangleq \mathbf{H} + \Delta\mathbf{H}$, where $\mathbf{H}$ represents the true channel matrix, and $\Delta\mathbf{H}$ denotes the estimation error which in the SISO case will have the form

$$\Delta\mathbf{H} = \begin{bmatrix} \Delta h_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \Delta h_{L-1} & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \Delta h_0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \Delta h_{L-1} \end{bmatrix} .$$

Here we have $\Delta h_l \sim \mathcal{CN}\left(0, \frac{\sigma_v^2}{N_{TS}-L+1}\right)$ leading to a detector that will be minimizing

$$\|\mathbf{y} - \hat{\mathbf{H}}\mathbf{x}\|_2^2 = \|\mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}) + \boldsymbol{v} - \Delta\mathbf{H}\hat{\mathbf{x}}\|_2^2 \approx \|\boldsymbol{v} - \Delta\mathbf{H}\mathbf{x}\|_2^2 \,, \tag{3.9}$$

where the approximation in (3.9) is valid when MLSD solution is in fact the transmitted symbol vector. If $\hat{\mathbf{x}}$ is independent of $\boldsymbol{v}$ and $\Delta\mathbf{H}$, the RHS of (3.9) can be described by a Chi-Square distribution having the variance $\tilde{\sigma_v}^2 = \sigma_v^2\left(1 + \frac{L}{N_{TS}-L+1}\right)$. Thus, the variance will no longer only depend on the noise variance, but also the number of training symbols and the channel length.

3.2 The QL-factorization and the Minimum-Phase Prefilter

In this Section we prove that the QL-factorization of a time-invariant channel matrix in a multipath environment will provide us with both the all-pass filter and the minimum-phase filter. We first present an intuitive argument for the connection and subsequently present a more formal proof of this. The proof has been given in [52] (see Appendix C), but in order to have coherent treatment of the QL-factorization of channel matrices in multipath channels it has been reproduced here.

When we QL-factorize the time-invariant block-banded block Toeplitz matrix, each block-row in $\mathbf{L}$ will be a shifted version of each other for $N \to \infty$, where each block-row is given by the spectral factorization, [53]. Likewise, the $M \times M$ unitary matrix, $\mathbf{Q}$, will be the matrix equivalent to the all-pass filter, where again each block-column of $\mathbf{Q}$ will be a shifted version of each other (for $N \to \infty$). Furthermore, it can be seen that each of these block-columns will correspond to the finite dimensional analog of the all-pass filter associated with the minimum-phase filter.
In the finite length case, each block-row of $\mathbf{L}$ (block-column of $\mathbf{Q}$) will not be exactly the same, but as we will show later in the paper, the values in each of these will converge toward the true minimum-phase filter as a function of the block-row number.[2] The block-columns of $\mathbf{Q}$ will similarly converge toward the associated all-pass filter.

[2]Strictly speaking the elements in the block-row of $\mathbf{L}$ converge toward the minimum-phase filter from the bottom up, since the Householder transformation computes the elements in the lower triangular matrix from the bottom (when we perform a QL-factorization instead of a QR-factorization).

Algorithm 1 Householder Transformation for QL-fact.

1: **Input:** Matrix $\mathbf{B}$
2: $\hat{\mathbf{B}} \Leftarrow \mathbf{B}$
3: **for** $k = 1$ to $\min\{M, N\}$ **do**
4: {Pick out the last column vector of $\hat{\mathbf{B}}$}
5: $\hat{\mathbf{b}} = \hat{\mathbf{B}}_{:,end}$
6: $\tilde{k} = M - k + 1$
7: {Do Householder reflection of $\hat{\mathbf{b}}$ (line 8 to 12)}
8: $\alpha = \left\|\hat{\mathbf{b}}\right\|$
9: $\tilde{\alpha} = e^{i \measuredangle \hat{b}_{\tilde{k}}} \alpha$
10: $\mathbf{v} = \hat{\mathbf{b}} + \tilde{\alpha}\mathbf{e}_{\tilde{k}}$
11: $\hat{\mathbf{U}}_k = e^{-i \measuredangle \hat{b}_{\tilde{k}}} \left(\frac{2\mathbf{v}\mathbf{v}^H}{\|\mathbf{v}\|^2} - \mathbf{I} \right)$
12: $\tilde{\mathbf{B}} = \hat{\mathbf{U}}_k \hat{\mathbf{B}}$
13: {Remove last row and last column of $\tilde{\mathbf{B}}$}
14: $\acute{\mathbf{B}} \Leftarrow \tilde{\mathbf{B}}_{1:(end-1),1:(end-1)}$
15: {Repeat for new $\hat{\mathbf{B}}$}
16: $\hat{\mathbf{B}} \Leftarrow \acute{\mathbf{B}}$
17: **end for**

After $K = \min\{M, N\}$ iterations we have;
$\mathbf{L} = \mathbf{U}_K \dots \mathbf{U}_2 \mathbf{U}_1 \mathbf{B}$
$\mathbf{Q} = \mathbf{U}_1^H \mathbf{U}_2^H \dots \mathbf{U}_K^H$

The Householder Transformation

In our analysis of the convergence toward the minimum-phase filter and the all-pass filter we use the Householder transformation to compute the QL-factorization. Therefore, we first briefly describe the steps of this transformation. The reason for choosing this transformation is its advantageous numerical stability to roundoff effects. For a more thorough treatment of the transformation and its numerical properties the reader is referred to [54]. In most textbooks, the Householder algorithm is only described for real numbers and since this transformation plays a crucial role in our treatment of the convergence rate, we here illustrate a complex version of the transformation. The Householder transformation (for QL-factorization) of a matrix $\mathbf{B} \in \mathbb{C}^{M \times N}$ works as illustrated in Algorithm 1, where $\mathbf{e}_k$ denotes the unit vector with 1 in the kth position, and where we have defined the unitary matrices $\mathbf{U}_k \in \mathbb{C}^{M \times M}$ and $\hat{\mathbf{U}}_k \in \mathbb{C}^{(M-k+1)\times(M-k+1)}$ as

$$\mathbf{U}_k \triangleq \begin{bmatrix} \hat{\mathbf{U}}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{k-1} \end{bmatrix}.$$

3.2.1 Convergence rate

In this Subsection we examine the convergence properties of the rows and columns the QL-factorization toward the minimum-phase and all-pass filters. In order to simplify this analysis, we first consider the simplest possible case, which is for the SISO case with a filter length of $L = 2$. We will then extend this result to the more general one.

SISO system with filter length $L = 2$

Any SISO filtering matrix of a length $L = 2$ system can be formulated as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a & 1 & \ddots & \vdots \\ 0 & a & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & a \end{bmatrix},$$

where we have normalized the impulse response such that $a \triangleq h_1/h_0 \neq 0$, leading to $H(z) = 1 + az^{-1}$. In this case it is trivial to compute the minimum-phase solution using the root-method

$$z_{mp} = \begin{cases} -a & \text{if } |a| \leq 1 \\ -1/a^* & \text{else} \end{cases}, \tag{3.10}$$

where z_{mp} represents the minimum-phase root. Since $H(z) = H_{ap}(z)H_{mp}(z)$ we have

$$H_{ap}(z) = \begin{cases} 1 & \text{if } |a| \leq 1 \\ \frac{z^{-1}+\frac{1}{a}}{1+\frac{1}{a^*}z^{-1}} & \text{else} \end{cases}, \tag{3.11}$$

where $H_{mp}(z)$ and $H_{ap}(z)$ represent the z-transformed minimum-phase and all-pass filters, respectively.[3]
By QL-factorizing the filtering matrix we get

$$\mathbf{L} = \begin{bmatrix} \alpha_N & 0 & \cdots & & 0 \\ \beta_{N-1} & \alpha_{N-1} & \ddots & & \vdots \\ 0 & \beta_{N-2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \alpha_2 & 0 \\ 0 & \cdots & 0 & \beta_1 & \alpha_1 \end{bmatrix}, \tag{3.12}$$

where we are interested in determining the α and β values. For notational brevity we introduce $\gamma_k \triangleq \hat{b}_{\tilde{k}}$, where $\hat{b}_{\tilde{k}}$ is defined in Algorithm 1 as the last

[3] In order to ensure that the magnitude response of the all-pass filter will always be one, we have normalized the minimum-phase filter such that $H_{mp}(z) = a(1 + 1/a^* z^{-1})$ whenever a root is reflected inside the unit circle (i.e. when $|a| > 1$).

element in vector $\hat{\mathbf{b}}$, which is being reflected in the kth iteration. Since $\hat{\mathbf{b}}_1 = [0, \ldots, 0, 1, a]^T$ we have $\gamma_1 = a$ for the first Householder reflection (also referred to as iteration $k = 1$). Based on the input vector we see from line 8 in Algorithm 1 that $\alpha_1 = \sqrt{1 + |\gamma_1|^2}$ and from lines 9-10 we get $\mathbf{v}_1 = [0, \ldots, 0, 1, \gamma_1 + \tilde{\alpha}_1]^T$. Lines 11 and 12 in Algorithm 1 lead to the following expression for the β,

$$\beta_1 = \frac{2a\, e^{-i\angle\gamma_1}\left(\gamma_1 + e^{i\angle\gamma_1}\alpha_1\right)}{1 + \left|\gamma_1 + e^{i\angle\gamma_1}\,\alpha_1\right|^2} \tag{3.13a}$$

$$= \frac{2a\,\left(|\gamma_1| + \alpha_1\right)}{1 + \left|\,|\gamma_1| + \alpha_1\right|^2}\,, \tag{3.13b}$$

where in (3.13b) we have used $\gamma_1 = e^{i\angle\gamma_1}|\gamma_1|$. After the first Householder reflection we have

$$\mathbf{U}_1\mathbf{H} = \begin{bmatrix} \ddots & 0 & \cdots & 0 \\ \ddots & 1 & \ddots & \vdots \\ \ddots & a & 1 & 0 \\ \vdots & 0 & \gamma_2 & 0 \\ \cdots & 0 & \beta_1 & \alpha_1 \end{bmatrix},$$

since there are only two non-zero elements in the columns of $\mathbf{H}$. In the next iteration we will have $\hat{\mathbf{b}}_1 = [0, \ldots, 0, 1, \gamma_2]^T$, and by examining the update steps in the Householder reflection carefully, it becomes clear that the value of γ_{k+1} can be expressed as a function of γ_k, leading to a recursive update given as

$$\gamma_{k+1} = a\left(1 - \frac{2}{1 + \left|\gamma_k + e^{i\angle\gamma_k}\alpha_k\right|^2}\right) \tag{3.14a}$$

$$= a\left(1 - \frac{2}{1 + \left|\,|\gamma_k| + \sqrt{1 + |\gamma_k|^2}\right|^2}\right) \tag{3.14b}$$

$$= \frac{a\,|\gamma_k|}{\sqrt{1 + |\gamma_k|^2}}\,. \tag{3.14c}$$

Likewise, the general expression for the α's and β's will be

$$\alpha_k = \sqrt{1 + |\gamma_k|^2} \tag{3.15}$$

$$\beta_k = \frac{2a\,\left(|\gamma_k| + \alpha_k\right)}{1 + \left|\,|\gamma_k| + \alpha_k\right|^2}\,. \tag{3.16}$$

From (3.15) we can verify that the α values will be positive and real-valued, which is exactly what is required from the QL-factorization. From (3.14) and (3.16) we also see the interesting property that all the values of the γ_k's and the β_k's will always have the same angle in the complex plane, determined by

$\angle \beta_k = \angle \gamma_k = \angle a$. This implies that the convergence of the β_k's to the true minimum-phase solution for each iteration takes place in the same direction in the complex plane.

Lemma 3.1 (Recursive computation of α_k and β_k) *In a time-invariant SISO system with $L = 2$, the coefficients in $\mathbf{L}$ obtained by the Householder transformation can be determined as*

$$\alpha_k = \sqrt{1 + |\gamma_k|^2}$$
$$\beta_k = \frac{2a\ (|\gamma_k| + \alpha_k)}{1 + |\,|\gamma_k| + \alpha_k|^2}$$

where

$$\gamma_{k+1} = \frac{a\,|\gamma_k|}{\sqrt{1 + |\gamma_k|^2}}\ .$$

Proof. Given above. As shown in [52, Appendix A] the recursive expression for γ_k given in (3.14), can be rewritten as

$$\gamma_k = e^{i\angle a}\sqrt{\frac{|a|^2 - 1}{1 - |a|^{-2k}}}\ . \tag{3.17}$$

Now in order to show that the values of α_k and β_k match the minimum-phase filter, we need to determine the fixed-point solutions for the parameter γ_k in (3.14), such that

$$\gamma_{fix} = f\left(\gamma_{fix}\right), \text{ where } f(x) = \frac{a\,|x|}{\sqrt{1 + |x|^2}}\ .$$

As shown in the lemma below, there are two fixed-points.

Lemma 3.2 (Fixed-points for γ) *In a time-invariant SISO system with $L = 2$, the fixed-point solutions for γ will be*

$$\gamma_{fix} = \begin{cases} 0 & \text{if } |a| \le 1 \\ e^{i\angle a}\sqrt{|a|^2 - 1} & \text{else} \end{cases}\ .$$

Proof. See [52, Appendix B] for a detailed proof.

Based on these fixed-points for γ we have

$$\alpha_{fix} = \begin{cases} 1 & \text{if } |a| \leq 1 \\ |a| & \text{else} \end{cases} \tag{3.18a}$$

$$\beta_{fix} = \begin{cases} a & \text{if } |a| \leq 1 \\ e^{i \measuredangle a} & \text{else} \end{cases} . \tag{3.18b}$$

Thus, the root of $\mathbf{L}$ obtained using the QL-factorization, will be $z_L = -\beta_{fix}/\alpha_{fix}$

$$\begin{aligned} z_L &= \begin{cases} -a & \text{if } |a| \leq 1 \\ -e^{i \measuredangle a}/|a| & \text{else} \end{cases} \\ &= \begin{cases} -a & \text{if } |a| \leq 1 \\ -1/a^* & \text{else} \end{cases} , \end{aligned} \tag{3.19}$$

which corresponds to the result given in (3.10), obtained by the traditional root-method of spectral factorization. Likewise, the unitary matrix will converge to the Infinite Impulse Response (IIR) all-pass filter given in (3.11). In order to ensure that we do in fact get the minimum-phase solution, we also need to prove that the recursive expression for γ_k converges to the fixed-points. In [52, Appendix C] it has been proved that this is indeed the case. Thus, it can be concluded that in the SISO case with a filter length of $L = 2$, the elements in the rows of $\mathbf{L}$ converge to the minimum-phase filter.[4]

In the following we examine the convergence rate to the fixed-point solutions, which can be determined based on the expression for γ_k given in (3.17). In order to compute the convergence rate we introduce

$$\gamma_k = \gamma_{fix} + \Delta\gamma_k , \tag{3.20}$$

where $\Delta\gamma_k$ represents the deviation of γ_k from the fixed-point solution. To upper bound the convergence we treat the cases of $|a| \leq 1$ and $|a| > 1$ separately.

The $|a| \leq 1$ case:
From (3.17) we get that

$$|\Delta\gamma_k| = |\gamma_k - \gamma_{fix}| = |a|^k \sqrt{\frac{|a|^2 - 1}{|a|^{2k} - 1}} \tag{3.21a}$$

$$\leq |a|^k \sqrt{\frac{|a|^2 - 1}{|a|^2 - 1}} = |a|^k \quad \text{for } \forall \ k \geq 1 . \tag{3.21b}$$

The $|a| > 1$ case:
When $|a| > 1$ the fixed-point is $|\gamma_{fix}| = \sqrt{|a|^2 - 1}$ and from [52, Lemma C.1] we

[4] This is no surprise, since it has already been shown in [25, 53] that the lower triangular matrix provides the spectral factor.

know that $|\gamma_k| \geq |\gamma_{fix}|$. As mentioned in [52, Appendix C] all of the terms which are compared have the same argument and, therefore, we can simply ignore the angle and just consider the case where the terms are real and positive. We then get

$$|\Delta\gamma_k| = |\gamma_k| - |\gamma_{fix}| \tag{3.22a}$$

$$= \sqrt{|a|^2 - 1}\left(\frac{1}{\sqrt{1 - |a|^{-2k}}} - 1\right) \tag{3.22b}$$

$$\leq \sqrt{|a|^2 - 1}\left(\frac{1}{1 - |a|^{-2k}} - 1\right) \tag{3.22c}$$

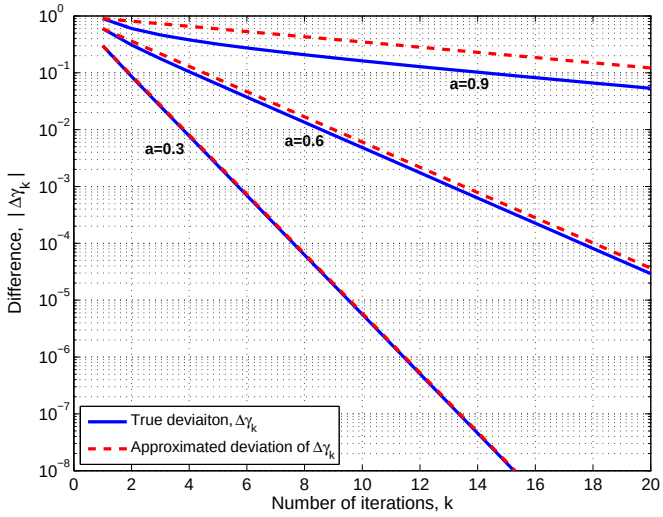
$$\leq |a|^{-2k}\,\frac{\sqrt{|a|^2 - 1}}{1 - |a|^{-2}} = |a|^{-2k}\,\frac{|a|^2}{\sqrt{|a|^2 - 1}}\,. \tag{3.22d}$$

Thus, we have the following lemma which upper bounds the convergence rate.

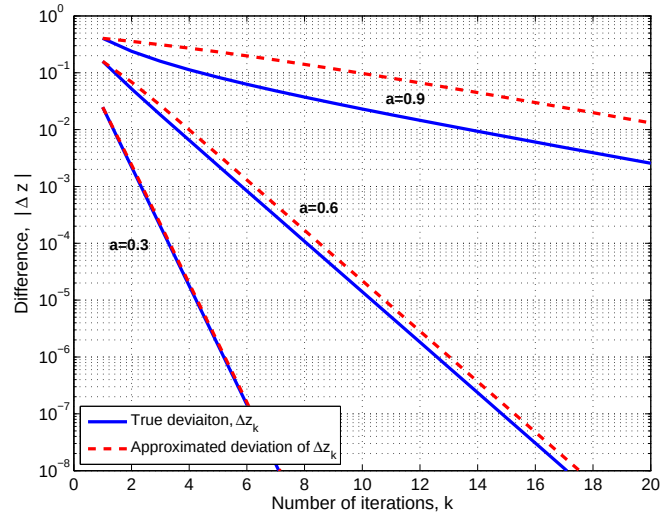
Lemma 3.3 (Upper bound on the convergence rate of γ) *In a time-invariant SISO system with $L = 2$, the convergence rate of γ_k can be upper bounded by*

$$|\Delta\gamma_k| \leq |\Delta\widehat{\gamma}_k| = \begin{cases} e^{k\,\ln(|a|)} & \text{if } |a| \leq 1 \\ \frac{|a|^2}{\sqrt{|a|^2-1}}\,e^{2k\,\ln(1/|a|)} & \text{else} \end{cases}\,.$$

From Lemma 3.3 we see the interesting property that the convergence rate is exponentially fast and is determined by $|a|$, in other words, the convergence rate to the fixed-point is governed by the localization of the root in the complex plane. In the case where we have a root which is close to the unit circle, we will have slow convergence to the minimum-phase solution. In Figure 3.5a the convergence of $\Delta\gamma_k$ and $\Delta\widehat{\gamma}_k$ have been shown as a function of the number of iterations for a $L = 2$ SISO system in the case where the root is $z = \{-0.3, -0.6, -0.9\}$, respectively. From the Figure it is clearly seen that the distance of the root from the unit circle has a huge influence on the convergence rate and, furthermore, we see that the upper bound becomes tighter as the distance between the root and the unit circle grows. It is also relevant to examine how the deviation $\Delta\gamma_k$ affects the value of the root and, therefore, we introduce $z_{L,k} \triangleq -\beta_k/\alpha_k$, which represents the root obtained from $\mathbf{L}$ in the kth iteration. Likewise, we have $\widehat{z}_{L,k} \triangleq -\widehat{\beta}_k/\widehat{\alpha}_k$, where the approximated values of α_k and β_k have been obtained using $\Delta\widehat{\gamma}_k$.
In Figure 3.5b the deviations from the true minimum-phase root has been plotted, where we have defined $\Delta z_k \triangleq z_{mp} - z_{L,k}$ and $\Delta\widehat{z}_k \triangleq z_{mp} - \widehat{z}_{L,k}$. From the Figure, we see that the deviation $\Delta\gamma_k$ is significantly larger than the deviation in the root value Δz_k.



(a) True and approximated value of $\Delta\gamma_k$.



(b) True and approximated value of root-deviation Δz_k.

Figure 3.5: Example of deviations of $\Delta\gamma_k$, $\Delta\widehat{\gamma}_k$, Δz_k, and $\Delta\widehat{z}_k$ in a SISO system, with length $L = 2$ and root at $a = \{0.3, 0.6, 0.9\}$, respectively

3.2.1.1 SISO system with filter length $L > 2$

In the case where we have a filter length of $L > 2$, the deviations of the recursions for the Householder transformation become much more complicated, since the vector $\mathbf{b}$ in Algorithm 1 will now have L non-zero elements. Thus, it will no longer be the simple scalar recursion for γ_k but instead a $(L-1)\times(L-1)$ matrix recursion and, furthermore, due to the multiple roots there will also be multiple fixed points. However, we can generalize the result obtained for the $L = 2$ SISO system by factorizing the filtering matrix into $(L-1)$ products of $L = 2$ filtering matrices,[5] such that

$$\mathbf{H} = \mathbf{H}_2^{(L-1)}\mathbf{H}_2^{(L-2)}\dots\mathbf{H}_2^{(1)} \ , \tag{3.23}$$

here $\mathbf{H}_2^{(l)}$ is the filtering matrix of the lth length two filter, where the z-transform of the equivalent infinite-length filter impulse response is given as $H_2^{(l)}(z) \triangleq 1 + a_l\, z^{-1}$. The factorization makes it possible to perform a QL-factorization on each of the $(L-1)$ terms in (3.23), which gives

$$\mathbf{H}_2^{(l)} = \mathbf{Q}_2^{(l)}\mathbf{L}_2^{(l)} \ . \tag{3.24}$$

where the convergence rate of each of the $(L-1)$ terms is given in Subsection concerning $L = 2$ systems. By inserting (3.24) into (3.23) we get

$$\mathbf{H} = \mathbf{QL} = \mathbf{Q}_2^{(L-1)}\mathbf{L}_2^{(L-1)}\mathbf{Q}_2^{(L-2)}\mathbf{L}_2^{(L-2)}\dots\mathbf{Q}_2^{(1)}\mathbf{L}_2^{(1)} \ . \tag{3.25}$$

We would like to reorder the terms on the RHS of (3.25) such that all $\mathbf{Q}_2^{(l)}$ terms are grouped together followed by all the $\mathbf{L}_2^{(l)}$ terms, i.e.

$$\mathbf{H} \cong \underbrace{\mathbf{Q}_2^{(L-1)}\mathbf{Q}_2^{(L-2)}\dots\mathbf{Q}_2^{(1)}}_{\mathbf{Q}}\ \underbrace{\mathbf{L}_2^{(L-1)}\mathbf{L}_2^{(L-2)}\dots\mathbf{L}_2^{(1)}}_{\mathbf{L}} \ , \tag{3.26}$$

where the equality holds when the system size $N \to \infty$. The reason that it is possible to rearrange the terms when the system size goes to infinity is due to fact that $\mathbf{L}_2^{(l)}$ and $\mathbf{Q}_2^{(l)}$ asymptotically become circulant matrices [55], and thereby, we can use the commutative property of circulant matrices [55]. Conceptually it is fairly easy to see why $\mathbf{L}_2^{(l)}$ asymptotically becomes circulant, since it is a banded matrix, but this might not be as obvious for the all-pass filtering matrix, which represents an IIR filter. However, it has been proved in [56] that the IIR filter has an exponential decay, which implies that, in the limit where the system size tends to infinity, the IIR filter becomes a Toeplitz matrix. In [55] it is proved that general Toeplitz matrices containing absolutely summable elements (also referred to as *Wiener Class Toeplitz Matrices*) asymptotically converge to circulant matrices too. Thus, in the limit $N \to \infty$ both matrices become

[5] It should be noted that the size of $\mathbf{H}_2^{(l)}$ decreases by one (both column- and row-wise) as l decreases by one, in order to enable the factorization.

circulant and, therefore, we know that the lower triangular matrix $\mathbf{L}$ converge to the minimum-phase filter for SISO systems of arbitrary length. Due to the unique factorization of $\mathbf{H} = \mathbf{QL}$ (where we require that the elements on the diagonal of $\mathbf{L}$ are real-valued and positive), $\mathbf{Q}$ *must* be the matrix version of the all-pass filter associated with the minimum-phase filter, since it is the only unitary matrix which links $\mathbf{L}$ with $\mathbf{H}$.
Based on the expression in (3.26) it is possible to approximate the convergence rate in a SISO system of arbitrary length, by examining the deviations in the approximated root values $\Delta\widehat{z}_k^{(l)} \triangleq z_{mp}^{(l)} - \widehat{z}_{L,k}^{(l)}$, where $z_{mp}^{(l)}$ represents the lth root of the true minimum-phase filter and $\widehat{z}_{L,k}^{(l)} \triangleq -\widehat{\beta}_k^{(l)}/\widehat{\alpha}_k^{(l)}$ is the approximated value of the lth root based on the upper bound given in Lemma 3.3. Thus, in the z-domain the difference between the true minimum-phase filter and the filter obtained based on $\widehat{z}_{L,k}^{(l)}$ becomes

$$\Delta H(z) \triangleq H_{mp}(z) - \widehat{L}_k(z) \tag{3.27a}$$

$$\approx z^{-(L-1)} \left[\prod_{l=1}^{L-1} \left(z - z_{mp}^{(l)} \right) - \prod_{l=1}^{L-1} \left(z - \widehat{z}_{L,k}^{(l)} \right) \right], \tag{3.27b}$$

where $\widehat{L}_k(z)$ represents the z-transform of the approximate value for the kth row in the lower triangular matrix, $\mathbf{L}$. In (3.27b) we have normalized the first coefficient and from the equation we can see that the main contribution to the difference between the true minimum-phase filter and the result obtained by the QL-factorization, will asymptotically come from the root which is closest to the unit circle. This observation fits well with what is described in [24, p. 508], where the convergence to the stabilizing solution of the DARE is exponential and determined by the spectral radius.

MIMO system

In the case of a MIMO system, we can first examine the length 2 system $\mathbf{H}(z) = \mathbf{I} + \mathbf{H}_1 z^{-1}$ where $N_T = N_R$. Compared to the SISO system of the same length, the only difference is that the operations now become $N_R \times N_T$ matrix operations instead of scalars. However, since we have $\eta = \min\{N_T, N_R\}(L-1)$ roots, we get 2^η fixed-points, thus it becomes more complicated to analyze even a simple $L = 2$ MIMO system. In the case of an arbitrary filter length, the argument presented in the previous Subsection, concerning the SISO system of length $L > 2$, can be repeated here.

It should be mentioned that during the process of writing the final version of the thesis it has been discovered, that the convergence analysis presented in this Subsection could also be done using classical fixed point theory. The contraction mapping theorem could e.g. be used for proving the existence and uniqueness of the fixed points [57]. The convergence rate and basins of attraction to a particular solution can then be determined and this framework might

be more suitable for analyzing the MIMO systems.

3.2.2 Simulation Results for Convergence

In this section simulation results for both SISO and MIMO systems are presented. For the SISO system we examine two channel scenarios; in the first one we have complex Gaussian distributed, $\mathcal{CN}(0,1)$, filter coefficients and in the second one we consider the Typical Urban (TU0) profile from the GSM specifications [13] shown in Figure 3.3.

In order to measure the convergence rate of the filter coefficients, we compute the relative difference between two overall filtering impulse response matrices, $\mathcal{H}_{a,k}$ and $\mathcal{H}_{b,k}$, at the iteration number k, as

$$d\left(\mathcal{H}_{a,k};\mathcal{H}_{b,k}\right) \triangleq \frac{\left\|\mathcal{H}_{a,k}-\mathcal{H}_{b,k}\right\|_2}{\left\|\mathcal{H}_{a,k}\right\|_2} \ . \tag{3.28}$$

We define $\mathcal{H}_{mp}$ as the impulse response of the true minimum-phase filter, and $\mathcal{H}_{L,k}$ represents the impulse response obtained from $\mathbf{L}$ (at iteration k). To measure how well the estimated all-pass filter, $\mathcal{H}_{Q,k}$, matches the estimated minimum-phase filter $\mathcal{H}_{L,k}$, we filter the original impulse response $\mathcal{H}$ with $\left(\mathcal{H}_{Q,k}\right)^H$, which gives us the output $\mathcal{H}_{\hat{L},k}$.

In all the simulations presented below, we have made 10,000 realizations of the examined channel profile, and computed the minimum-phase and the all-pass filter for each realization. The filter length of the all-pass filter is set to $L_{ap} = 64$ in the simulations. Based on the results obtained from the 10,000 filter realizations, we have computed the mean and median value of the relative errors, $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$ and $d(\mathcal{H}_{L,k};\mathcal{H}_{\hat{L},k})$.

The results for the Gaussian filter coefficients with uniform power in the delay domain are shown in Figure 3.6, where we see that the rows in $\mathbf{L}$ converge to the true minimum-phase filter as a function of the iteration number (i.e. the row number).[6] From the Figure we observe that the median value of $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$ converges exponentially to zero and that the median difference is about 10^{-8} after 140 iterations. The convergence of the average difference is considerably slower, due to the instances where a channel realization has zeros very close to the unit circle, which will lead to a slow convergence. Thus, these cases tend to bias the estimate of average convergence rate. This is indeed what can be observed from the estimated PDF of $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$. Likewise, the mean of

[6] Again, strictly speaking the convergence occurs from the last row and up, since it is the QL-factorization.

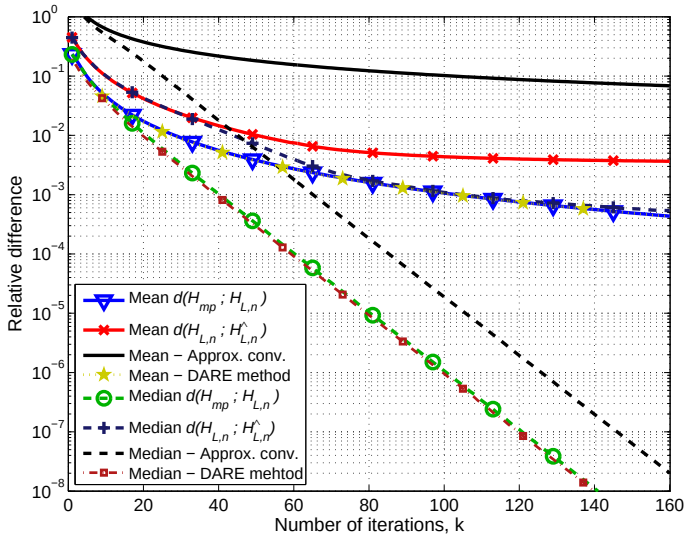


Figure 3.6: The relative deviations $d(\mathcal{H}_{mp};\mathcal{H}_{L,k})$ and $d(\mathcal{H}_{L,k};\mathcal{H}_{\hat{L},k})$ in a SISO channel with Gaussian coefficients having uniform power in the delay domain, $L = 6$.

Table 3.1: Complexity of computing the minimum-phase filter using the fast QL-factorization (QL) and the DARE method (DARE) using k iterations in a length L SISO system.

k	Method	$L = 5$	$L = 10$	$L = 15$	$L = 20$
10	QL	$3.18 \cdot 10^2$	$5.98 \cdot 10^2$	$9.03 \cdot 10^2$	$1.23 \cdot 10^3$
	DARE	$3.70 \cdot 10^2$	$1.48 \cdot 10^3$	$3.34 \cdot 10^3$	$5.95 \cdot 10^3$
20	QL	$6.38 \cdot 10^2$	$1.17 \cdot 10^3$	$1.72 \cdot 10^3$	$2.30 \cdot 10^3$
	DARE	$7.30 \cdot 10^2$	$2.94 \cdot 10^3$	$6.65 \cdot 10^3$	$1.19 \cdot 10^4$

$d(\mathcal{H}_{L,k};\mathcal{H}_{\hat{L},k})$ seems to be biased, which (besides the effect described above) is also due to the truncation of the IIR all-pass filter. Both the mean and median value of the approximated convergence of $d(\mathcal{H}_{mp};\mathcal{H}_{L,k})$ computed on the basis of (3.27) have also been plotted. From the Figure it can be seen that the trend of the true and approximated deviation behaves similarly. As a reference we have also included the relative deviation between the true minimum-phase filter and the one obtained using the DARE method, and from this it is possible to see that convergence of the two iterative methods is almost identical. In Table 3.1 the complexity of computing the minimum-phase filter using the two iterative methods has been compared (based on (2.16) and (3.40)), and from this it is seen that the fast QL-factorization method has a computational advantage.

In Figure 3.7 the result for the TU0 profile is shown and it is seen that the

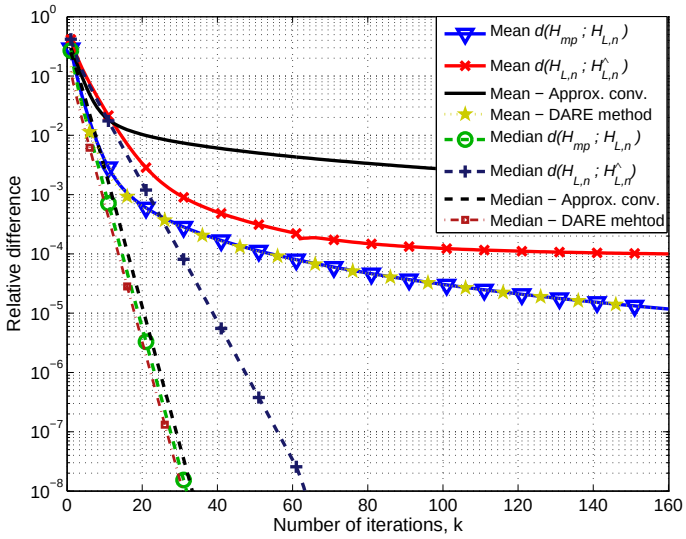


Figure 3.7: The relative deviations $d(\mathcal{H}_{mp}; \mathcal{H}_{L,k})$ and $d(\mathcal{H}_{L,k}; \mathcal{H}_{\hat{L},k})$ in the SISO channel TU0 with $L = 5$.

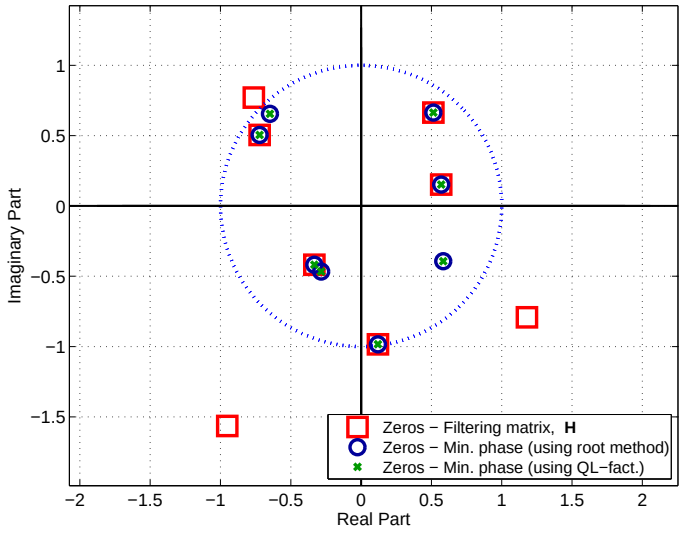


Figure 3.8: Locations of roots in a 2×2 MIMO system with Gaussian filter coefficients, $L = 5$. The number of iterations in the QL-factorization is $k = 200$.

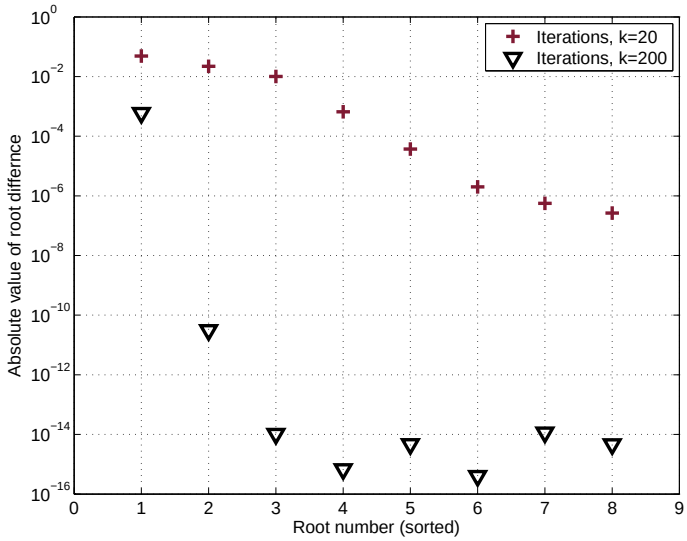


Figure 3.9: Deviation of the roots in a 2×2 MIMO system with Gaussian filter coefficients, $L = 5$ for iteration $k = 20$ and $k = 200$.

convergence rate is faster for this channel type compared with the Gaussian filter coefficients with uniform power in the delay domain. It is again observed that the median value of the difference decreases more rapidly than the mean value. We also see that the approximated convergence of $d(\mathcal{H}_{mp}; \mathcal{H}_{L,k})$ is even closer to the actual convergence for this channel profile and that the DARE method again has similar convergence.

In Figure 3.8 we have a plot of the location of the roots of a 2×2 MIMO system having Gaussian coefficients with filter length $L = 5$, leading to 8 roots. From the plot it is seen that the roots of $\mathbf{H}(z)$ (illustrated with squares) which lie outside the unit circle are reflected inside (the circles) using the root method. Furthermore, it is seen that these roots match the roots of $\mathbf{L}(z)$. In Figure 3.9 the root difference $\Delta z_k^{(l)} = z_{mp}^{(l)} - z_{L,k}^{(l)}$ has been plotted for each of the roots $l = \{1, \ldots, 8\}$ for iteration $k = 20$ and $k = 200$. The roots have been sorted according to their distance to the unit circle, such that the one closest to the unit circle is called root 1, etc. The Figure shows that the closer the root is to the unit circle, the slower the convergence is, which follows the convergence analysis given in Section 3.2.1. After $k = 200$ iterations, it is primarily the root closest to the unit circle which contributes to the difference between the filter obtained from $\mathbf{L}$ and the true minimum-phase filter. As a concluding remark to this subsection we can therefore say that it has hereby shown how the QL-factorization of the channel matrix gives the finite length equivalent to

the minimum-phase and the all-pass filters.

3.3 Efficient Minimum-Phase Prefilter Computation

In Section 3.2 it has been proved that the QL-factorization of the time-invariant multipath channel matrix provides the minimum-phase filter and the all-pass filter. This knowledge is used to present a novel method for computing these two classical filters in a computationally efficient way as shown in [58] (found in Appendix B). We illustrate here how the fast QL-factorization can be exploited for time-invariant channels.

When general methods are used to compute the QL-factorization, it requires $\mathcal{O}\left(N^3\right)$ operations [59]. But for Toeplitz matrices there exist methods with lower computational complexity. Different methods have been proposed for performing the fast QR-factorization (see e.g. [59–63] and the references therein), each of which has different numerical properties and slightly different complexity as well.[7] They do, however, all use the shift-invariance property of Toeplitz matrices to partition it in two ways and it is this partitioning that leads to the low complexity schemes. In [59], the QL-factorization can be performed using $13MN + 6N^2$ operations for general $M \times N$ Toeplitz matrices, while the method proposed in [61] requires $13MN + 6.5N^2$ operations. The methods described in the literature usually deal with real-valued matrices but the results can be extended to be valid over the complex field, [61]. To extend the method in [59] to complex numbers, will however require another type of rank-1 downdating, which is described in [64]. The methods can also be extended to handle block Toeplitz matrices for the general MIMO case as well, [65].

In the following, we illustrate how the fast QR-factorization methods utilize the structure in Toeplitz matrix, which has also been described in detail in [59]. To keep the description of the fast QR-factorization close to the treatment found in the literature, we first consider a general Toeplitz matrix, $\mathbf{T} \in \mathbb{R}^{M \times N}$. Afterwards, we will show the implications of having a more specific channel matrix, $\mathbf{H}$. Furthermore, we briefly describe how a generalization of the fast factorization algorithm to complex numbers can be achieved. The fact that the principal submatrices $\mathbf{T}_{-1} \in \mathbb{R}^{M-1 \times N-1}$ of the Toeplitz matrix $\mathbf{T}$ are identical [59]

[7] Methods for QR-factorization may easily be converted to QL-factorization.

is exploited in order to derive a fast factorization method, i.e.

$$\mathbf{T} = \begin{bmatrix} t_1 & t_{-2} & \cdots & t_{-n} \\ t_2 & t_1 & \cdots & t_{-n+1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & t_1 \\ \vdots & \cdots & \ddots & \vdots \\ t_m & t_{m-1} & \cdots & t_{m-n} \end{bmatrix} = \begin{bmatrix} t_1 & \mathbf{t}_{r(1)}^T \\ \mathbf{t}_{c(1)} & \mathbf{T}_{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{-1} & \mathbf{t}_{c(n)} \\ \mathbf{t}_{r(m)}^T & t_{m-n} \end{bmatrix} . \quad (3.29)$$

Here $\mathbf{t}_{r(1)}^T \triangleq \mathbf{T}_{1,2:N}$ denotes the row-vector containing the elements in the first row of $\mathbf{T}$ excluding the element t_1, $\mathbf{t}_{c(1)} \triangleq \mathbf{T}_{2:M,1}$, $\mathbf{t}_{r(m)}^T \triangleq \mathbf{T}_{M,1:N-1}$, and $\mathbf{t}_{c(n)}^T \triangleq \mathbf{T}_{N,1:M-1}$. Let $\mathbf{R} \in \mathbb{R}^{N\times N}$, which is similar to the definition in (3.3), be the upper triangular Cholesky factor of $\mathbf{T}^T\mathbf{T}$, which we can partition in two ways as well

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & \mathbf{r}_{r(1)}^T \\ \mathbf{0}_{1,N-1} & \mathbf{R}_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t & \mathbf{r}_{c(n)} \\ \mathbf{0}_{N-1,1} & r_{n,n} \end{bmatrix} , \quad (3.30)$$

where $\mathbf{r}_{r(1)}^T \triangleq \mathbf{R}_{1,2:N}$ and $\mathbf{r}_{c(n)}^T \triangleq \mathbf{R}_{N,2:N}$. By substituting the two expressions of $\mathbf{T}$ in (3.29) and the two expressions of $\mathbf{R}$ in (3.30) into $\mathbf{R}^T\mathbf{R} = \mathbf{T}^T\mathbf{T}$ we obtain the following two equations:

$$\left[\begin{array}{c|c} r_{1,1}^2 & r_{1,1}\mathbf{r}_{r(1)}^T \\ \hline r_{1,1}\mathbf{r}_{r(1)} & \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T + \mathbf{R}_b^T\mathbf{R}_b \end{array}\right] = \left[\begin{array}{c|c} t_1^2 + \mathbf{t}_{c(1)}^T\mathbf{t}_{c(1)} & t_1\mathbf{t}_{r(1)}^T + \mathbf{t}_{c(1)}^T\mathbf{T}_{-1} \\ \hline t_1\mathbf{t}_{r(1)} + \mathbf{T}_{-1}^T\mathbf{t}_{c(1)} & \mathbf{t}_{r(1)}\mathbf{t}_{r(1)}^T + \mathbf{T}_{-1}^T\mathbf{T}_{-1} \end{array}\right] \quad (3.31a)$$

$$\left[\begin{array}{c|c} \mathbf{R}_t^T\mathbf{R}_t & \mathbf{R}_t^T\mathbf{r}_{c(n)} \\ \hline \mathbf{r}_{c(n)}^T\mathbf{R}_t & \mathbf{r}_{c(n)}^T\mathbf{r}_{c(n)} + r_{n,n}^2 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{T}_{-1}^T\mathbf{T}_{-1} + \mathbf{t}_{r(m)}\mathbf{t}_{r(m)}^T & \mathbf{T}_{-1}^T\mathbf{t}_{c(n)} + \mathbf{t}_{r(m)}t_1 \\ \hline \mathbf{t}_{c(n)}^T\mathbf{T}_{-1} + \mathbf{t}_{r(m)}^T t_{m-n} & \mathbf{t}_{c(n)}^T\mathbf{t}_{c(n)} + t_{m-n}^2 \end{array}\right] \quad (3.31b)$$

From the upper left submatrix of (3.31b) we get

$$\mathbf{R}_t^T\mathbf{R}_t = \mathbf{T}_{-1}^T\mathbf{T}_{-1} + \mathbf{t}_{r(m)}\mathbf{t}_{r(m)}^T , \quad (3.32)$$

and from lower right submatrix of (3.31a) we have

$$\mathbf{R}_b^T\mathbf{R}_b + \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T = \mathbf{T}_{-1}^T\mathbf{T}_{-1} + \mathbf{t}_{r(1)}\mathbf{t}_{r(1)}^T . \quad (3.33)$$

By combining (3.32) and (3.33) a relation between $\mathbf{R}_b$ and $\mathbf{R}_t$ is achieved:

$$\mathbf{R}_b^T\mathbf{R}_b = \mathbf{R}_t^T\mathbf{R}_t + \mathbf{t}_{r(1)}\mathbf{t}_{r(1)}^T - \mathbf{t}_{r(m)}\mathbf{t}_{r(m)}^T - \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T . \quad (3.34)$$

This implies that we can compute $\mathbf{R}_b^T\mathbf{R}_b$ based on $\mathbf{R}_t^T\mathbf{R}_t$ using one rank-1 updating and two rank-1 downdating modifications [59].

We will here briefly describe how a single downdate is executed in [59], and in order to do this we introduce

$$\mathbf{Z}^T\mathbf{Z} \triangleq \mathbf{R}_t^T\mathbf{R}_t + \mathbf{t}_{r(1)}\mathbf{t}_{r(1)}^T - \mathbf{t}_{r(m)}\mathbf{t}_{r(m)}^T \ ,$$

and

$$\mathbf{A}^T\mathbf{A} = \mathbf{Z}^T\mathbf{Z} - \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T \ ,$$

where we have defined

$$\mathbf{A} \triangleq \begin{bmatrix} i\cdot\mathbf{r}_{r(1)}^T \\ \mathbf{Z} \end{bmatrix} .$$

Using the expression for $\mathbf{Z} \in \mathbb{R}^{N\times N}$ and $\mathbf{A} \in \mathbb{C}^{(N+1)\times N}$ we can formulate (3.34) as

$$\mathbf{R}_b^T\mathbf{R}_b = \mathbf{Z}^T\mathbf{Z} - \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T = \mathbf{A}^T\mathbf{A} \ , \tag{3.35}$$

where the task is to compute the Cholesky factor $\mathbf{R}_b$ based on $\mathbf{Z}$ and $\mathbf{r}_{r(1)}$ without forming the Cholesky factorization of $\mathbf{R}_b^T\mathbf{R}_b$.
In order to achieve this we introduce the matrix $\mathbf{V}$ such that

$$\mathbf{V}\mathbf{A} = \mathbf{V}\begin{bmatrix} i\cdot\mathbf{r}_{r(1)}^T \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b \\ \mathbf{0}_{1\times N} \end{bmatrix} , \tag{3.36}$$

where the matrix $\mathbf{V} \in \mathbb{C}^{(N+1)\times(N+1)}$ consists of a product of N plane rotation matrices, i.e. $\mathbf{V} \triangleq \tilde{\mathbf{V}}_{N,(N+1)}\tilde{\mathbf{V}}_{(N-1),N}\cdots\tilde{\mathbf{V}}_{1,2}$. Each of the plane rotations is defined as

$$\tilde{\mathbf{V}}_{k,(k+1)} = \begin{bmatrix} 1 & 0 & \cdots & & & \cdots & 0 \\ 0 & \ddots & & & & & \vdots \\ \vdots & & i\cdot\rho_k & \varsigma_k & & & \\ & & \varsigma_k & i\cdot\rho_k & & & \vdots \\ \vdots & & & & & \ddots & 0 \\ 0 & \cdots & & & \cdots & 0 & 1 \end{bmatrix} \begin{matrix} \\ \\ \longleftarrow \\ \longleftarrow \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ k\text{ th row} \\ (k+1)\text{ th row} \\ \\ \\ \end{matrix} \tag{3.37}$$

In (3.37), the elements $\rho_k \in \mathbb{R}$ and $\varsigma_k \in \mathbb{R}$ are obtained in a similar fashion as one would compute the elements in a Givens rotation [54,66]. Thus, having the $(N+1)$-dimensional vector with the elements $i\cdot a_k$ and a_{k+1} at position k and $(k+1)$, respectively, we can choose ρ_k and ς_k such that

$$\begin{bmatrix} i\cdot\rho_k & \varsigma_k \\ -\varsigma_k & i\cdot\rho_k \end{bmatrix}\begin{bmatrix} i\cdot a_k \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} \sqrt{a_k^2 - a_{k+1}^2} \\ 0 \end{bmatrix} ,$$

where we have assumed that both a_k and a_{k+1} are real numbers and $a_k^2 > a_{k+1}^2$. Furthermore, it should be noted that $\tilde{\mathbf{V}}_{k,(k+1)}^T\tilde{\mathbf{V}}_{k,(k+1)} = \mathbf{I}_{N+1}$, and that $\tilde{\mathbf{V}}_{k,(k+1)}$ is simply a plane rotation operating in the plane $(k, k+1)$ (just like the ordinary Givens rotation), which imply that a premultiplication with $\tilde{\mathbf{V}}_{k,(k+1)}$

only changes the elements in the kth and $(k+1)$th rows. In other words, we can compute one row at a time. Thus, if we know the first row of $\mathbf{Z}$ (along with $\mathbf{r}_{r(1)}^T$), we can compute first row of $\mathbf{R}_b$.

From the treatment above, it is seen that we can compute the first row in $\mathbf{R}_b$ if we know the first row in $\mathbf{R}_t$ (based on (3.34)). Since the first row in $\mathbf{R}_b$ is equivalent to the second row in $\mathbf{R}_t$ (except of a column shift), we have, thereby, indirectly computed the second row of $\mathbf{R}_t$. This procedure can be repeated for the second row, and so forth and, therefore, we obtain a recursive scheme for computing each row of $\mathbf{R}$. Likewise, the unitary matrix $\mathbf{Q}$ can be computed recursively using plane rotation matrices. This is done using, among others things, the (update) plane rotation matrix $\acute{\mathbf{V}}_{k,k+1}$ associated with (downdate) plane rotation matrix $\mathbf{V}_{k,k+1}$, i.e.

$$\acute{\mathbf{V}} \begin{bmatrix} \mathbf{r}_{r(1)}^T \\ \mathbf{R}_b^T \end{bmatrix} = \begin{bmatrix} \mathbf{Z} \\ \mathbf{0}_{1,N} \end{bmatrix},$$

where $\mathbf{Z}^T\mathbf{Z} = \mathbf{R}_b^T\mathbf{R}_b + \mathbf{r}_{r(1)}\mathbf{r}_{r(1)}^T$. See [59] for a more detailed treatment of each of the rank-1 updating and downdating steps, and for a more comprehensive description of the fast QR-factorization algorithm.

Generalization to Complex Numbers

At first glance one might think that it is a bit exaggerated to address the issue of generalizing the fast QL-factorization method to complex numbers, since it from a mathematical point of view is considered to be trivial to extend methods from the real to the complex domain. However, this is not always the case on the implementation level, and as an example we can refer to the quotation in [54, p. 233] where it is stated; "*Most of the algorithms that we present in this book have complex versions that are fairly straight forward to derive from their real counterparts. (This is NOT to say that everything is easy and obvious at the implementation level).*" Thus, we will shortly describe this extension. The downdating procedure given in (3.37) is not applicable when the Toeplitz matrix $\mathbf{T}$ contains complex numbers, and instead a procedure for downdating without the imaginary multiplier is needed. To mention one, in [64] a treatment of downdating the Cholesky factorization without imaginary multiplier is found, which rely on $\mathbf{\Sigma}$-unitary transformations with

$$\mathbf{\Sigma} = \begin{bmatrix} -1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

applied in a similar way as the Givens rotations are used in the rank-1 updating method [64,67].[8] When we compute the rank-1 downdate we should be particu-

[8] A matrix $\mathbf{A}$ is called $\mathbf{\Sigma}$-unitary if it fulfill $\mathbf{A}^H\mathbf{\Sigma}\mathbf{A} = \mathbf{\Sigma}$.

larly careful with the phases of complex numbers since these have to be tracked from transformation to transformation.

Fast QR-Factorization for Channel Matrices

When Toeplitz matrix is a banded channel matrix with L non-zero (complex) elements in each column (here, we still only consider a SISO system), the expression in (3.34) simplifies to

$$\mathbf{R}_b^H \mathbf{R}_b = \mathbf{R}_t^H \mathbf{R}_t - \mathbf{r}_{r(1)} \mathbf{r}_{r(1)}^H \ , \tag{3.38}$$

since $\mathbf{t}_{r(1)} = \mathbf{0}_{N-1,1}$ and $\mathbf{t}_{r(m)} = \mathbf{0}_{N-1,1}$. Thus, we only need one rank-1 downdating modification when we want to compute $\mathbf{R}$. The first row in $\mathbf{R}$ (excluding the first element in $\mathbf{R}$, which is simply the Euclidean norm of the first column of $\mathbf{T}$, i.e. $R_{1,1} = \|\mathbf{T}_{1:M,1}\|_2$) can be computed as

$$\mathbf{r}_{r(1)}^H = \mathbf{t}_{c(1)}^H \mathbf{T}_{-1} / R_{1,1} \ . \tag{3.39}$$

Here $\mathbf{t}_{c(1)}$ is a vector containing $L-1$ channel coefficients and having zeros in the remaining elements. The complexity of computing $\mathbf{t}_{c(1)}^H \mathbf{T}_{-1}$ in (3.39) is $(L-1) + \sum_{l=1}^{L-1} l = 1/2(L^2+L) - 1 \approx 1/2(L^2+L)$ and since the vector has at most $L-1$ non-zero elements, it requires $\left(1/2L^2 + 3/2L\right)$ complex operations to compute $\mathbf{r}_{r(1)}^H$. On top of this, we also need L complex operations to compute $R_{1,1}$ plus one square root operation.[9]

From (3.36) it is seen that we need to perform N plane (downdating) rotations $\tilde{\mathbf{V}}_{k,k+1}$ to obtain $\mathbf{R}$. Besides that, we also need the associated updating matrix $\acute{\mathbf{V}}_{k,k+1}$ in order to compute the unitary matrix. The rotation matrices $\tilde{\mathbf{V}}_{k,k+1}$ and $\acute{\mathbf{V}}_{k,k+1}$ can be computed using seven complex operations plus one square root operation. As already mentioned, the rank-1 downdating procedure in [64] also requires a rotation of the input signal, $\mathbf{r}_{r(1)}$, when it is used for complex numbers. Since the number of non-zero elements in $\mathbf{r}_{r(1)}$ is L, we need L complex operations to obtain the rotated signal. Furthermore, for each plane rotation we also need to multiply $\tilde{\mathbf{V}}_{k,k+1}$ with $\tilde{\mathbf{V}}_{k-1,k} \ldots \tilde{\mathbf{V}}_{1,2}\mathbf{A}$. Complexity-wise, this corresponds to a multiplication of a 2×2 plane rotation matrix with a $2 \times L$ matrix (the latter represents the recursively computed input signal), and the complexity of this is therefore $4L$. Thus, we can compute each row in $\mathbf{R}$ using $5L+7$ complex operations and one square root computation leading to a total complexity of $(5L+7)N + \left(1/2L^2 + 5/2L\right)$ complex operations and $N+1$ square root operations for calculating $\mathbf{R}$.

[9] Like in the rest of this thesis, we define an operation as a complex Multiply and Accumulate (MAC) instruction.

Recall that the fast QL-factorization computes a single row of $\mathbf{L}$ (or column of $\mathbf{Q}$) at a time, which is a great advantage when the QL-factorization is used for prefilter computation. This is due to the fact that each row of $\mathbf{L}$ converges to the true minimum-phase filter as shown in Section 3.2. This implies that we can stop the computation of the rows in $\mathbf{L}$ once we have obtained the required precision of the filter coefficients. Likewise, we only need to compute a certain fraction of the columns in $\mathbf{Q}$ to obtain the required precision of the all-pass filter. By using the fast QL-factorization to compute the filters, the complexity no longer scales with the size of the channel matrix $\mathbf{H}$, but depends on the required precision. The number of rows in $\mathbf{L}$ (and thus columns in $\mathbf{Q}$), which is used to obtain the estimated minimum-phase and all-pass filters, is referred to as the number of *iterations*, k. Thus, if the required precision of the minimum-phase estimate can be obtained using k iterations, the computational complexity will be

$$\mathcal{O}_{min} = (k-1)\cdot(5L+7) + \left(1/2L^2 + 5/2L\right) \tag{3.40}$$

complex operations plus $k+1$ square root operations. Each of the last $L_{ap}-L$ columns of $\mathbf{Q}$ require $(L+\tilde{j})(\tilde{j}+1)$ operations with $\tilde{j} = \{0,\ldots,L_{ap}-L-1\}$ and L_{ap} denotes the all-pass filter length. The complexity of computing each of the $j+1$ last columns of $\mathbf{Q}$ is $L_{ap}(j+1)$ for $j = \{L_{ap}-L,\ldots,L_{ap}\}$. If the number of required iterations is higher than the length of the prefilter, we also need $L_{ap}(L_{ap}+1)$ complex operations to calculate each of the remaining columns (i.e. the columns from $L_{ap}+1$ to k counted from right to left). Thus, the overall complexity of computing the prefilter, is

$$\begin{aligned}\mathcal{O}_{ap} = \quad & \textstyle\sum_{l=0}^{\min\{(L_{ap}-1);(k-1)\}} \min\{(L+l);L_{ap}\}\cdot(l+1) \\ & + \max\{0;(k-L_{ap})\}\cdot L_{ap}(L_{ap}+1)\ ,\end{aligned} \tag{3.41}$$

assuming that $k \geq L_{ap}-L+1$. Note that the last term in (3.41) vanishes when $k \leq L_{ap}$ and that we can obtain the first L_{ap} filter coefficients after $(L_{ap}-L+1)$ iterations. This means that whenever L is close to L_{ap} we only need a few iterations if we are willing to sacrifice precision in favor of complexity.

From (3.40) and (3.41) it is seen that the all-pass filter is the "bottleneck" complexity-wise if the all-pass filter is long compared to the channel length. Thus, we can often achieve a further complexity reduction by only computing the minimum-phase filter using the fast QL-factorization and then estimate the all-pass filter based on a polynomial division (or deconvolution operation) since we have

$$H_{ap}(z) = \frac{H(z)}{H_{mp}(z)}. \tag{3.42}$$

This polynomial division can be achieved using

$$\mathcal{O}_{ap,deconv} = L_{ap}L$$

operations, since each all-pass filter coefficient can be computed using L operations.

The approximate low complexity method proposed in [68], which uses Linear Prediction (LP) to obtain an estimate of the all-pass and minimum-phase filters, will approximately require $1/2 \cdot (L+1)(L+2)+L_p^2+2L_p+(L+1)(L_p+1)$ operations (complex multiplications). Here, L_p denotes the order of the prediction-error filter. Figure 3.10 shows a plot of the complexity of the LP method and the fast QL-factorization method (when the all-pass filter has been obtained by polynomial division). In the Figure the number of iterations in the fast QL-factorization has been adjusted such that the two methods gives similar accuracy in the filter coefficients of the minimum-phase filter in order to make a fair comparison. Using this setup, the all-pass filter obtained based on the fast QL-factorization method seems to give a slightly better accuracy. From Figure 3.10 it is seen that the fast QL-factorization method has a complexity which is comparable with the LP method. Additionally, the former method has the advantage that an arbitrary number of iterations can be made without affecting the all-pass filter length, which is not the case for the LP method.

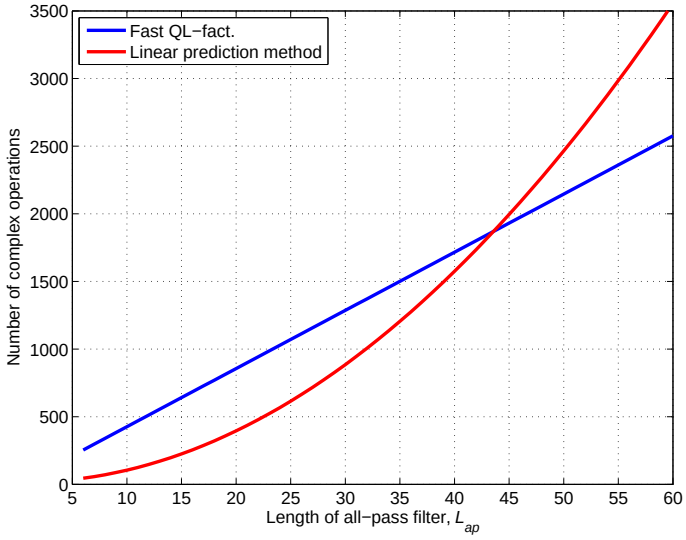


Figure 3.10: The complexity of computing the minimum-phase and all-pass filters for a fixed channel length $L = 6$ using the fast QL-factorization and the linear prediction method shown in [68], respectively.

In Figure 3.11 the complexity of computing the minimum-phase filter using the fast QL-factorization and the DARE method for a fixed channel length $L = 10$ has been plotted (the prefilter computation has not been included here). Also, the complexity of the two methods has been plotted as a function of the channel length as shown in Figure 3.12

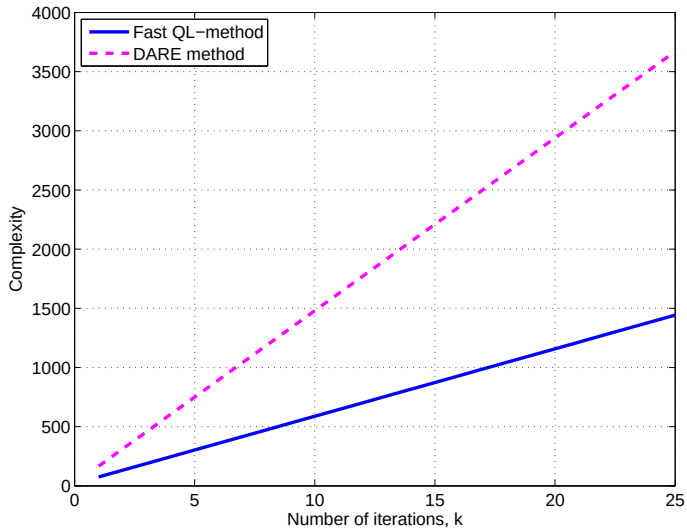


Figure 3.11: The complexity of computing the minimum-phase for a fixed channel length $L = 10$ using the fast QL-factorization and the DARE method.

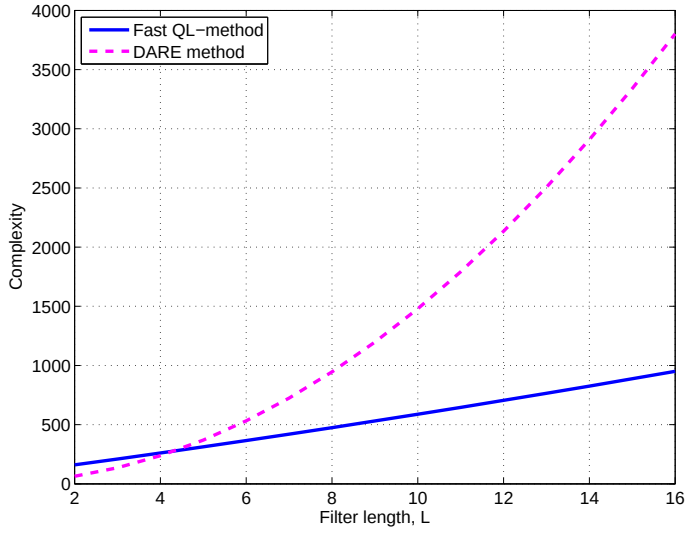


Figure 3.12: The complexity of computing the minimum-phase for a fixed number of iterations $k = 10$ using the fast QL-factorization and the DARE method.

Simulations of Filter Computation using Fast QL-factorization

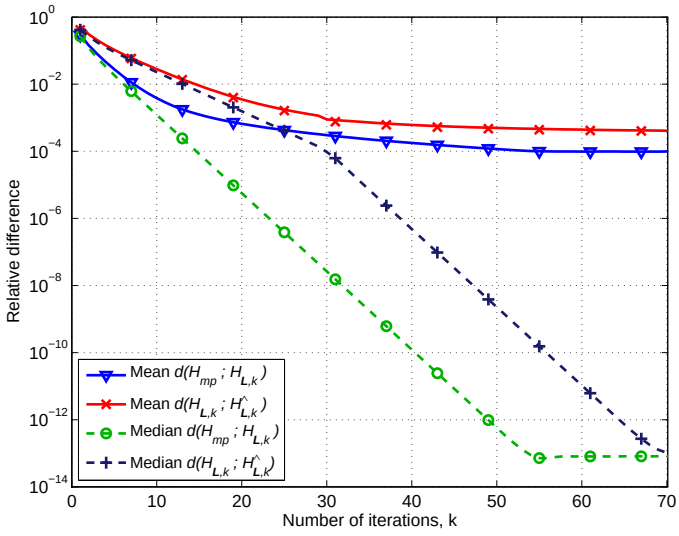


Figure 3.13: TU0 profile, $L = 6$. Mean and median value of the relative deviations, $d(\mathcal{H}_{mp}; \mathcal{H}_{L,k})$ and $d(\mathcal{H}_{L,k}; \mathcal{H}_{\hat{L},k})$ when $L_{ap} = 32$.

Here we present simulation results for the same two channels as shown in Figure 3.3, namely the Typical Urban (TU0) and the Hilly Terrain (HT0) profiles. In this simulation setup we do, however, only consider the SISO case, i.e. the oversampling is $N_{sps} = 1$. The simulation setup is similar to the simulations presented in Section 3.2, which imply that we have made 10,000 realizations of the examined channel profile and computed the minimum-phase and the all-pass filter for each realization. Again, we measure the relative difference between the two filters (as shown in (3.28)) as a function of the iteration number, k. The filter length of the all-pass filter is in all simulations $L_{ap} = 32$.

To sum up on the notation, we have $\mathcal{H}_{mp}$, which represents the impulse response of the true minimum-phase filter, and $\mathcal{H}_{L,k}$, which is the impulse response obtained from $\mathbf{L}$ (at iteration k). To measure how well the estimated all-pass filter, $\mathcal{H}_{Q,k}$, match the estimated minimum-phase filter $\mathcal{H}_{L,k}$, we filter the original impulse response $\mathcal{H}$ with $(\mathcal{H}_{Q,k})^H$, which gives us the output $\mathcal{H}_{\hat{L},k}$. Once again, we have computed the mean and median value of the relative errors, $d(\mathcal{H}_{mp}; \mathcal{H}_{L,k})$ and $d(\mathcal{H}_{L,k}; \mathcal{H}_{\hat{L},k})$.

In Figure 3.13, the results for the TU0 profile are shown and, here, we can see that the average relative deviation between the true minimum-phase filter and

estimated solution is approximately 10^{-2} after 7-8 iterations. To obtain the same relative deviation between the estimated minimum-phase filter and the estimated all-pass filter we need approximately 14-15 iterations. We can see from Figure 3.13 that the median value of the relative error converges faster than the mean value, which indicates that some of the realizations will bias the estimate of the mean value due to "outliers" in the distribution of the relative error. By inspecting the approximated Probability Density Function (PDF) for different iterations, it is observed that a few realizations converge slower than the majority and they will, therefore, in some sense bias the estimate as we also mentioned in Section 3.2.

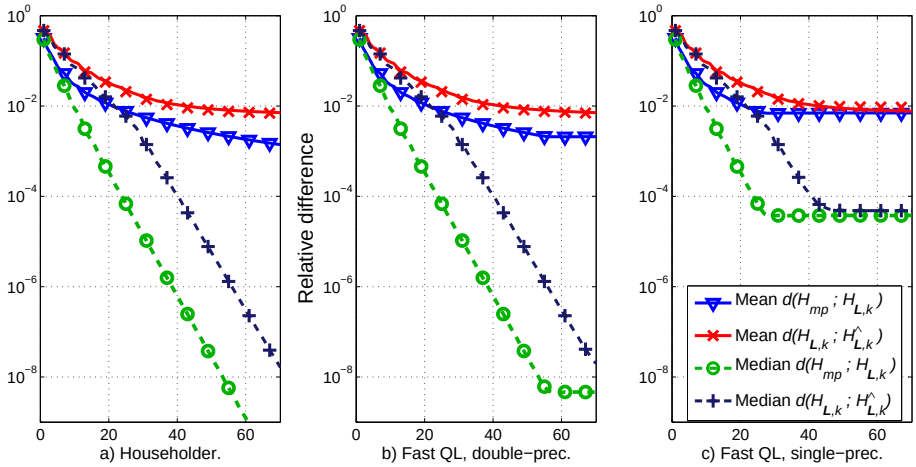


Figure 3.14: HT0 profile, $L = 10$. Mean and median value of the relative deviations, $d(\mathcal{H}_{mp};\mathcal{H}_{L,k})$ and $d(\mathcal{H}_{L,k};\mathcal{H}_{\hat{L},k})$ when $L_{ap} = 32$. Result given for a) the Householder transformation and for b) and c) the fast QL-factorization using floating-point double- and single-precision, respectively.

Figure 3.14(b) shows the result for the HT0 profile and in this case the convergence is slower than the TU0 profile. This is not surprising since the channel impulse response is longer, which makes it more likely that there are roots close to the unit circle. For this profile, we need 21 iterations to obtain an average precision of 10^{-2} between the true and estimated minimum-phase filter. In Figure 3.13 and Figure 3.14(b), we see that the relative difference $d(\mathcal{H}_{L,k};\mathcal{H}_{\hat{L},k})$ tends to be biased due to the usage of a finite length all-pass filter. This bias term can be reduced by increasing the length of the all-pass filter, L_{ap}.

To examine the numerical stability of the fast QL-factorization, the Householder

transformation has been used as a reference, since this algorithms is considered to be numerically stable [54]. In Figure 3.14(a), the minimum-phase filter for the HT0 profile has been computed using Householder transformation and the result is compared to the ones obtained by the fast QL-factorization using either double- or single-precision floating-point operations. We see that the numerical stability of the fast QL-factorization algorithm is not as good as the Householder transformation. This is basically due to a numerical instability of the rank-1 downdating procedure [64]. However, the numerical instability do not seem to be of such a significant scale that it prevents the fast method from being applicable for practical purposes.

3.4 Summary

It has been proven how the QL-factorization of the channel matrix (of a multi-path system) gives the finite length equivalent to the minimum-phase and the all-pass filters. Thereby, a novel method for computing these two classical filters in a numerically stable way has been presented. Alternatively, this new method can be used for determining the filters in a computationally efficient manner using the fast QL-factorization with the complexity depending on the required precision. Based on the link between minimum-phase prefiltering and the QL-factorization of frequency-selective channels, it is possible to relate SD with minimum-phase prefiltered RSSE. As a result, it is possible to regard SD as a generalization of the traditional RSSE, providing a unifying framework for the two detection methods.

Chapter 4

Sampling

Over a wide range of SNRs, the average complexity of SD is significantly smaller than exhaustive search detectors. But in worst case, the complexity is still exponential [45]. Thus, in scenarios with poor SNR or in MIMO systems with huge transmit and receive dimensions, even SD can be infeasible. A way to overcome this problem is to use approximate Markov Chain Monte Carlo (MCMC) detectors instead. In this Chapter, we will therefore treat a detector relying on Gibbs Sampling (GS). The treatment given in this Chapter is based on the paper in Appendix D.

4.1 Gibbs Sampling

It is well known that Markov Chain Monte Carlo detectors asymptotically can provide the optimal solution [69–71]. The MCMC method called Gibbs sampling (also known as Glauber dynamics) is a special case of the Metropolis-Hastings algorithm [12], which can be used for sampling from distributions of multiple dimensions and it has among others been proposed for detection purposes in wireless communication in [1, 72–74] (see also the references therein). In contrast to previously proposed MCMC methods for such problems, we here suggest an approach in which we optimize the "temperature" parameter so that in

steady state, i.e. after the Markov chain has mixed, there is only polynomially (rather than exponentially) small probability of encountering the optimal solution. More precisely, we obtain the largest value of the temperature parameter for this to occur since the higher the temperature, the faster the mixing. This is in contrast to simulated annealing techniques where, rather than being held fixed, the temperature parameter is tended to zero in its search for a global minimum.

In the treatment below, we have a minor change in the system model given in (2.1) such that we instead use the equivalent model

$$\mathbf{y} = \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{x} + \acute{\boldsymbol{\upsilon}} \ . \tag{4.1}$$

The model in (4.1) has the advantage that both the channel matrix, $\acute{\mathbf{H}}$, and the noise vector, $\acute{\boldsymbol{\upsilon}}$, will have i.i.d. $\mathcal{N}(0,1)$ entries. This simplifies the derivations, and for the same reason, we will only consider the real-valued block-fading MIMO system with N transmit and receive dimensions. Furthermore, we assume that $\Omega = \{\pm 1\}$ and that the channel coefficients are known. The normalization in (4.1) guarantees that SNR represents the signal-to-noise ratio per receive dimension (which we define as the ratio of the total transmit energy per channel divided by the per-component noise variance as described in among others [41]). It should be noted that the real-valued system considered here can easily be extended to complex-numbers by performing a so-called *IQ-splitting* (a.k.a. *composite real representation*) where $\mathbf{x} = \left[\Re(\mathbf{x}_c)^T, \Im(\mathbf{x}_c)^T\right]^T$, $\acute{\boldsymbol{\upsilon}} = \left[\Re(\boldsymbol{\upsilon})^T, \Im(\boldsymbol{\upsilon})^T\right]^T$, and

$$\acute{\mathbf{H}} = \begin{bmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{bmatrix} .$$

For further details on this see e.g. [48, 75].

The equivalent system model (4.1) leads to the following ML optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\Omega^N} \left\| \mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{x} \right\|_2^2 . \tag{4.2}$$

As explained further below, for analysis purposes we will focus on the regime where $\mathrm{SNR} > (2+\epsilon)\ln(N)$ (where $\epsilon > 0$) to get the probability of error of the ML detector to go to zero. Further, in our analysis, without loss of generality, we will assume that the all minus one vector was transmitted, $\mathbf{x} = -\mathbf{1}_{N\times 1}$, and we therefore get

$$\mathbf{y} = \acute{\boldsymbol{\upsilon}} - \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{1}_{N\times 1} \ . \tag{4.3}$$

One way of solving the optimization problem given in (4.2) is by using Markov Chain Monte Carlo simulations, which as mentioned in the introduction of this Section asymptotically converge to the optimal solution. More specifically, the GS computes the conditional probability of each symbol in the constellation set at the jth index in the estimated symbol vector. This conditional probability is obtained by keeping the $(j-1)$ other values in the estimated symbol vector fixed. Thus, in the kth iteration the probability of the jth symbol adopts the value $\omega \in \Omega$, is given as

$$p\left(\hat{\mathbf{x}}_j^{(k)} = \omega \,|\theta\right) = \frac{e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}-\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\tilde{\mathbf{x}}_{j|\omega}\right\|_2^2}}{\sum\limits_{\tilde{\mathbf{x}}_{j|\tilde{\omega}}\in\Omega} e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}-\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\tilde{\mathbf{x}}_{j|\tilde{\omega}}\right\|_2^2}}, \tag{4.4}$$

where $\tilde{\mathbf{x}}_{j|\omega}^T \triangleq \left[\hat{\mathbf{x}}_{1:j-1}^{(k)}, \omega, \hat{\mathbf{x}}_{j+1:N}^{(k-1)}\right]^T$ and where we for simplicity have introduced $\theta = \left\{\hat{\mathbf{x}}^{(k-1)}, \mathbf{y}, \acute{\mathbf{H}}\right\}$.[1] The parameter α represents a tunable positive number, which controls the mixing time of the Markov chain. This parameter is also known as the "temperature". The larger α is, the faster the mixing time of the Markov chain will be. But as we will show in this Chapter, there is an upper limit on α, in order to ensure that the probability of finding the optimal solution in steady state is not exponentially small. The Gibbs Sampler will with probability $p\left(\hat{\mathbf{x}}_j^{(k)} = \omega \,|\theta\right)$ keep ω at the j'th index in the estimated symbol vector and compute conditional probability of the $(j+1)$th index in a similar fashion. We define one iteration of the Gibbs sampler as a randomly-ordered update of all the $j = \{1, \ldots, N\}$ indices in the estimated symbol vector $\hat{\mathbf{x}}$.[2] The initialization of the symbol vector $\hat{\mathbf{x}}^{(0)}$ can either be chosen randomly or, alternatively, e.g. the zero-forcing solution can be used.

[1] When we compute the probability of symbol ω at the j'th position, we more precisely condition on the symbols $\hat{\mathbf{x}}_{1:j-1}^{(k)}$ and $\hat{\mathbf{x}}_{j+1:N}^{(k-1)}$, but to keep the notation simple, we do not explicitly state that in the equations above.

[2] We need a randomly-ordered update for the Markov chain to be reversible and for our subsequent analysis to go through. It is also possible to just randomly select a symbol j to update, without insisting that a full sequence should be done. This also makes the Markov chain reversible and has the same steady state distribution. In practice a fixed, say sequential, order can be employed, although the Markov chain is no longer reversible. Note that our theoretical analysis is assuming randomly selected symbol updates for analytical convenience. In our experimental section we used a sequential updating order which empirically yields a slight convergence acceleration.

4.1.1 Complexity of the Gibbs sampler

The conditional probability for the j'th symbol in (4.4) can be computed efficiently by reusing the result obtained for the $j-1$'th symbol when we evaluate $\left\|\mathbf{y}-\sqrt{\mathrm{SNR}/N}\acute{\mathbf{H}}\tilde{\mathbf{x}}_{j|\omega}\right\|_2^2$. Since we are only changing the j'th symbol in the symbol vector, the difference $\mathbf{d}_j \triangleq \mathbf{y}-\sqrt{\mathrm{SNR}/N}\acute{\mathbf{H}}\tilde{\mathbf{x}}_{j|\omega}$ can be expressed as

$$\mathbf{d}_j = \mathbf{d}_{j-1} - \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}_{1:N,j}\Delta x_{j|\omega} \ , \tag{4.5}$$

where $\Delta x_{j|\omega} \triangleq x_{j|\omega}^{(k)} - x_{j|\tilde{\omega}}^{(k-1)}$. Thus, the computation of conditional probability of a certain symbol in the j'th position costs $2N$ operations since we need to compute both the inner product $\mathbf{d}_j^T\mathbf{d}_j$ and the product $\acute{\mathbf{H}}_{1:N,j}\Delta x_{j|\omega}$.[3] We need to update $|\Omega|-1$ symbols per index j, which leads to a complexity of $\mathcal{O}\left(2N^2[|\Omega|-1]\right)$ operations per iteration. For further details on the implementation of the Gibbs sampler see [76].

GS using QL-factorization

In the case were the number of iterations in the Gibbs sampler is sufficiently larger than the system size, the complexity of GS can be reduced using a QL-factorization (or QR-factorization) of the channel matrix, $\acute{\mathbf{H}} = \mathbf{QL}$, such that the optimization problem in 4.2 becomes

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\Omega^N} \quad \|\tilde{\mathbf{y}}-\mathbf{L}\mathbf{x}\|_2^2 \ ,$$

with $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^H\mathbf{y}$. Due to lower triangular structure in $\mathbf{L}$, the product $\mathbf{Lx}$ requires less computations compared to a full channel matrix. Thus, for a square channel matrix of size N, the complexity per iteration will be reduced to $\mathcal{O}\left(N(N+1)[|\Omega|-1]\right)$ and we will, therefore, roughly save $(N^2-N)[|\Omega|-1] \approx N^2[|\Omega|-1]$ operations per iteration. This computation saving should be compared with the complexity of performing the QL-factorization, which requires $\mathcal{O}\left(N^3\right)$. Therefore, in cases where the number of iterations is $k > \frac{N}{|\Omega|-1}$, we can achieve a reduction in complexity.

4.1.2 Probability of Error

We first examine the probability of error for the ML detector, which will be used in order to evaluate the performance of the Gibbs sampler. To ease our

[3] Like in Chapter 3, we define an operation as a MAC instruction.

analysis, we will assume that the ML detector finds the correct transmitted vector. Before we derive the probability of error for the ML detector, we will state a lemma which we will make repeated use of.

Lemma 4.1 (Gaussian Integral) *Let* $\mathbf{v}$ *and* $\mathbf{w}$ *be independent Gaussian random vectors, each with distribution* $\mathcal{N}(\mathbf{0}_{N\times 1}, \mathbf{I}_N)$. *Then, if* $1-2a^2\eta(1+2\eta) > 0$,

$$\mathsf{E}\left\{e^{\eta(\|\mathbf{v}+a\mathbf{w}\|_2^2-\|\mathbf{v}\|_2^2)}\right\} = \left(\frac{1}{1-2a^2\eta(1+2\eta)}\right)^{N/2}. \tag{4.6}$$

Proof.

$$\begin{aligned}
&\mathsf{E}\left\{e^{\eta(\|\mathbf{v}+a\mathbf{w}\|_2^2-\|\mathbf{v}\|_2^2)}\right\} \\
&= \int \frac{d\mathbf{w}d\mathbf{v}}{(2\pi)^N} e^{-\frac{1}{2}\left[\begin{array}{cc}\mathbf{v}^T, & \mathbf{w}^T\end{array}\right]\left[\begin{array}{cc}\mathbf{I}_N & -2a\eta\mathbf{I}_N \\ -2a\eta\mathbf{I}_N & (1-2a^2\eta)\mathbf{I}_N\end{array}\right]\left[\begin{array}{c}\mathbf{v}\\ \mathbf{w}\end{array}\right]} \\
&= \frac{1}{\det^{1/2}\left[\begin{array}{cc}\mathbf{I}_N & -2a\eta\mathbf{I}_N \\ -2a\eta\mathbf{I}_N & (1-2a^2\eta)\mathbf{I}_N\end{array}\right]} \\
&= \frac{1}{\det^{N/2}\left[\begin{array}{cc}1 & -2a\eta \\ -2a\eta & 1-2a^2\eta\end{array}\right]} = \left(\frac{1}{1-2a^2\eta(1+2\eta)}\right)^{N/2}.
\end{aligned}$$

This completes the proof of Lemma 4.1. □

Assuming that the vector $\mathbf{x} = -\mathbf{1}_{N\times 1}$ was transmitted, the ML detector will make an error if there exists a vector $\mathbf{x} \neq -\mathbf{1}_{N\times 1}$ such that

$$\left\|\mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{x}\right\|_2^2 \leq \left\|\mathbf{y} + \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{1}_{N\times 1}\right\|_2^2 = \|\acute{\boldsymbol{\upsilon}}\|_2^2 .$$

In other words, the probability of an error P_e is

$$\begin{aligned}
\mathrm{P}_e &= \mathrm{P}\left(\left\|\mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{x}\right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2\right) \\
&= \mathrm{P}\left(\left\|\acute{\boldsymbol{\upsilon}} + \sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}(-\mathbf{1}_{N\times 1} - \mathbf{x})\right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2\right),
\end{aligned}$$

for some $\mathbf{x} \neq -\mathbf{1}_{N\times 1}$, which can be formulated as

$$\mathrm{P}_e \quad = \quad \mathrm{P}\left(\left\| \acute{\boldsymbol{\upsilon}} + 2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2 \right), \tag{4.7}$$

for some $\boldsymbol{\delta} \neq 0$, where we in (4.7) have defined $\boldsymbol{\delta} \triangleq \frac{1}{2}(-\mathbf{1}_{N\times 1} - \mathbf{x})$. Now, using the union bound

$$\mathrm{P}_e \leq \sum_{\boldsymbol{\delta}\neq 0} \mathrm{P}\left(\left\| \acute{\boldsymbol{\upsilon}} + 2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2 \right). \tag{4.8}$$

We will use the Chernoff bound [77] to bound the quantity inside the summation. Thus,

$$\mathrm{P}\left(\left\| \acute{\boldsymbol{\upsilon}} + 2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2 \right) \leq \mathsf{E}\left\{ e^{-\beta\left(\left\| \acute{\boldsymbol{\upsilon}} + 2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right\|_2^2 - \|\acute{\boldsymbol{\upsilon}}\|_2^2 \right)} \right\} \tag{4.9a}$$

$$= \left(\frac{1}{1 + 8\frac{\mathrm{SNR}\|\boldsymbol{\delta}\|_2^2}{N}\beta(1-2\beta)} \right)^{N/2}, \tag{4.9b}$$

where $\beta \geq 0$ is the Chernoff parameter [77], and where we have used Lemma 4.1 with $\eta = -\beta$ and $a = 2\sqrt{\frac{\mathrm{SNR}\|\boldsymbol{\delta}\|_2^2}{N}}$, since

$$\mathsf{E}\left\{ \left(2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right) \left(2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right)^T \right\} = 4\frac{\mathrm{SNR}\|\boldsymbol{\delta}\|_2^2}{N} \mathbf{I}_N.$$

The optimal value for β is $\frac{1}{4}$, which yields the tightest bound:

$$\mathrm{P}\left(\left\| \acute{\boldsymbol{\upsilon}} + 2\sqrt{\frac{\mathrm{SNR}}{N}} \acute{\mathbf{H}} \boldsymbol{\delta} \right\|_2^2 \leq \|\acute{\boldsymbol{\upsilon}}\|_2^2 \right) \leq \left(\frac{1}{1 + \frac{\mathrm{SNR}\|\boldsymbol{\delta}\|_2^2}{N}} \right)^{N/2}. \tag{4.10}$$

Note that this depends only on $\|\boldsymbol{\delta}\|_2^2$, the number of nonzero entries in $\boldsymbol{\delta}$. Plugging this into the union bound yields

$$\mathrm{P}_e \leq \sum_{i=1}^{N} \binom{N}{i} \left(\frac{1}{1 + \frac{\mathrm{SNR}i}{N}} \right)^{N/2}. \tag{4.11}$$

Let us first look at the linear (i.e., i proportional to N) terms in the above sum. Thus,

$$\binom{N}{i} \left(\frac{1}{1 + \frac{\mathrm{SNR}i}{N}} \right)^{N/2} \approx e^{NH(\frac{i}{N}) - \frac{N}{2}\ln\left(1 + \frac{\mathrm{SNR}i}{N}\right)},$$

where $H(\cdot)$ is entropy in "nats". Clearly, if $\lim_{N\to\infty} \mathrm{SNR} = \infty$, then the linear terms go to zero (superexponentially fast).

Let us now look at the sublinear terms. In particular, let us examine $i = 1$ and substitute the SNR with a function $g(N)$:

$$\xi_N \triangleq N\left(\frac{1}{1+\frac{\mathrm{SNR}}{N}}\right)^{N/2} \Rightarrow \tag{4.12a}$$

$$\ln(\xi_N) = \ln(N) + \frac{N}{2}\ln\left(1+\frac{g(N)}{N}\right)^{-1} = \frac{N}{2}\left\{\frac{2}{N}\ln(N) - \ln\left(1+\frac{g(N)}{N}\right)\right\} \tag{4.12b}$$

$$= \frac{1}{2}\frac{\frac{2}{N}\ln(N) - \ln\left(1+\frac{g(N)}{N}\right)}{\frac{1}{N}} \tag{4.12c}$$

Since all terms in (4.12c) tends to zero as $N \to \infty$, we can use l'Hôpital's rule:

$$\lim_{N\to\infty} \ln(\xi_N) = \frac{1}{2}\frac{2\left(-N^{-2}\ln(N)+1\right) - \left(1+\frac{g(N)}{N}\right)^{-1}\left\{-N^{-2}g(N) + \frac{1}{N}g'(N)\right\}}{-N^{-2}} \tag{4.13a}$$

$$= \ln(N) - 1 + \frac{1}{2}\left(Ng'(N) - g(N)\right) \tag{4.13b}$$

We need to let term ξ_N go to zero in order to get P_e in (4.11) to go to zero. By choosing $g(N) \geq (2+\epsilon)\ln(N)$ where $\epsilon > 0$, we get from (4.13b) that

$$\lim_{N\to\infty} \ln(\xi_N) = \frac{\epsilon}{2}\left(1 - \ln(N)\right) = -\infty \ ,$$

leading to $\xi_N \to 0$ for $N \to \infty$. Therefore, we require that $\mathrm{SNR} \geq (2+\epsilon)\ln(N)$ [4]. A similar argument shows that all other sublinear terms also go to zero, and so, which gives us Lemma 4.2. [5]

Lemma 4.2 (SNR scaling) *If* $SNR > (2+\epsilon)\ln(N)$, *where* $\epsilon > 0$ *then* $\mathrm{P}_e \to 0$ *as* $N \to \infty$.

[4] We could also use a slower growing function $g(N) = 2\ln(N) + \tilde{g}(N)$, where $\tilde{g}(N)$ is any function that goes to infinity as N goes to infinity, such as $\tilde{g}(N) = \epsilon\ln(\ln(N))$ and so forth. But for simplicity we have just chosen $\tilde{g}(N) = \epsilon\ln(N)$.

[5] A rigorous proof can be given using the saddle point method, similarly to the proof in the Subsection 4.1.3.

4.1.3 Computing the optimal α

Assuming that the vector $\mathbf{x} = -\mathbf{1}_{N\times 1}$ has been transmitted, the probability of finding this solution *after the Markov chain has mixed* is simply $\pi_{-\mathbf{1}}$, the steady-state probability of being in the all -1 state. Clearly, if this probability is *exponentially small*, it will take exponentially long for the Gibbs sampler to find it. We will, therefore, insist that the mean of $\pi_{-\mathbf{1}}$ must only be *polynomially small*.

Mean of $\pi_{-\mathbf{1}}$

This calculation has a lot in common with the one given in Section 4.1.2. Note that the steady state value of $\pi_{-\mathbf{1}}$ is simply

$$\pi_{-\mathbf{1}} = \frac{e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}+\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{1}_{N\times 1}\right\|_2^2}}{\sum_{\mathbf{x}} e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}+\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\mathbf{x}\right\|_2^2}} = \frac{e^{-\frac{1}{2\alpha^2}\|\acute{\boldsymbol{\upsilon}}\|_2^2}}{\sum_{\mathbf{x}} e^{-\frac{1}{2\alpha^2}\left\|\acute{\boldsymbol{\upsilon}}+\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}(\mathbf{x}-\mathbf{1}_{N\times 1})\right\|_2^2}} \tag{4.14a}$$

$$= \frac{e^{-\frac{1}{2\alpha^2}\|\acute{\boldsymbol{\upsilon}}\|_2^2}}{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left\|\acute{\boldsymbol{\upsilon}}+2\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\boldsymbol{\delta}\right\|_2^2}} = \frac{1}{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left(\left\|\acute{\boldsymbol{\upsilon}}+2\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\boldsymbol{\delta}\right\|_2^2-\|\acute{\boldsymbol{\upsilon}}\|_2^2\right)}}, \tag{4.14b}$$

where the summations (over $\mathbf{x}$ and $\boldsymbol{\delta}$) are over 2^N terms. By Jensen's inequality on (4.14), we get

$$\mathsf{E}\{\pi_{-\mathbf{1}}\} \geq \frac{1}{\mathsf{E}\left\{\frac{1}{\pi_{-\mathbf{1}}}\right\}} = \frac{1}{\mathsf{E}\left\{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left(\left\|\acute{\boldsymbol{\upsilon}}+2\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\boldsymbol{\delta}\right\|_2^2-\|\acute{\boldsymbol{\upsilon}}\|_2^2\right)}\right\}} \tag{4.15a}$$

$$= \frac{1}{\sum_{\boldsymbol{\delta}} \mathsf{E}\left\{e^{-\frac{1}{2\alpha^2}\left(\left\|\acute{\boldsymbol{\upsilon}}+2\sqrt{\frac{\mathrm{SNR}}{N}}\acute{\mathbf{H}}\boldsymbol{\delta}\right\|_2^2-\|\acute{\boldsymbol{\upsilon}}\|_2^2\right)}\right\}} \tag{4.15b}$$

$$= \frac{1}{1+\sum_{\boldsymbol{\delta}\neq 0}\left(\frac{1}{1+4\frac{\mathrm{SNR}\|\boldsymbol{\delta}\|_2^2}{N}\frac{1}{\alpha^2}(1-\frac{1}{\alpha^2})}\right)^{N/2}} \tag{4.15c}$$

$$= \frac{1}{1+\sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2}}. \tag{4.15d}$$

In (4.15c), we have used Lemma 4.1 and in (4.15d) we have defined $\beta \triangleq 4\mathrm{SNR}\frac{1}{\alpha^2}(1-\frac{1}{\alpha^2})$. While it is possible to focus on the linear and sublinear terms in the above summation separately, to give conditions for $\mathsf{E}\{\pi_{-\mathbf{1}}\}$ to have the form of $1/\mathrm{poly}(N)$, we will be interested in the exact exponent and, thus, we will need a more accurate estimate. To do this, we shall use saddle point integration. Note that

$$\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx e^{NH(\frac{i}{N})-\frac{N}{2}\ln\left(1+\frac{\beta i}{N}\right)} ,$$

where again $H(\cdot)$ represents the entropy in "nats". Therefore, the summation in the denominator of (4.15d) can be approximated as a Stieltjes integral[6] [79]:

$$\sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx N\sum_{i=1}^{N} e^{NH(\frac{i}{N})-\frac{N}{2}\ln\left(1+\frac{\beta i}{N}\right)}\frac{1}{N} \tag{4.16a}$$

$$\approx N\int_0^1 e^{NH(x)-\frac{N}{2}\ln(1+\beta x)}dx . \tag{4.16b}$$

For large N, this is a saddle point integral and can be approximated by the formula

$$\int_0^1 e^{Nf(x)}dx \approx \sqrt{\frac{2\pi}{N|f''(x_0)|}}e^{Nf(x_0)} , \tag{4.17}$$

where x_0 is the saddle point of $f(\cdot)$, i.e.,$f'(x_0)=0$. In our case,

$$f(x) = -x\ln x-(1-x)\ln(1-x)-\frac{1}{2}\ln(1+\beta x) ,$$

and thus,

$$f'(x) = \ln\frac{1-x}{x}-\frac{1}{2}\frac{\beta}{1+\beta x} .$$

In general, it is not possible to solve for $f'(x_0)=0$ in closed form. However, in our case, if we assume that $\beta = 4\mathrm{SNR}\frac{1}{\alpha^2}(1-\frac{1}{\alpha^2}) \gg 1$ (which is true since the SNR grows at least logarithmically), then it is not too hard to verify that the saddle point is given by

$$x_0 = e^{-\frac{\beta}{2}} . \tag{4.18}$$

Hence $f(x_0) =$

$$-e^{-\frac{\beta}{2}}\ln e^{-\frac{\beta}{2}}-(1-e^{-\frac{\beta}{2}})\ln(1-e^{-\frac{\beta}{2}})-\frac{1}{2}\ln(1+\beta e^{-\frac{\beta}{2}})$$

$$\approx \frac{\beta}{2}e^{-\frac{\beta}{2}}+e^{-\frac{\beta}{2}}-\frac{1}{2}\beta e^{-\frac{\beta}{2}} = e^{-\frac{\beta}{2}} .$$

[6]The Stieltjes integral are also sometimes referred to as the Riemann-Stieltjes integral, [78].

Furthermore, by inserting x_0 into $f''(x) = -\frac{1}{x} - \frac{1}{1-x} - \frac{1}{2}\frac{\beta^2}{(1+\beta x)^2}$, yields

$$f''(x_0) \approx -e^{\frac{\beta}{2}} - 1 + \frac{1}{2}\beta^2 \approx -e^{\frac{\beta}{2}} \ . \tag{4.19}$$

Replacing these into the saddle point expression in (4.17) show that

$$\sum_{i=1}^{N} \binom{N}{i} \left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx \sqrt{2\pi/N} \exp\left(Ne^{-\frac{\beta}{2}} - \frac{\beta}{4}\right) . \tag{4.20}$$

We want $\mathsf{E}\{\pi_{-\mathbf{1}}\}$ to behave as $\frac{1}{N^\zeta}$ and according to (4.15), this means that we want the expression in (4.20) to behave as N^ζ. Let us take

$$e^{Ne^{-\frac{\beta}{2}}} = N^\zeta \ .$$

Solving for β yields

$$\beta = 4\mathrm{SNR}\frac{1}{\alpha^2}\left(1 - \frac{1}{\alpha^2}\right) = 2\left(\ln N - \ln\left(\ln N\right) - \ln\zeta\right). \tag{4.21}$$

Incidentally, this choice of β yields $e^{-\frac{\beta}{4}} \approx \frac{1}{\sqrt{N}}$, and so we have the following result.

Lemma 4.3 (Mean of $\pi_{-\mathbf{1}}$) *If α is chosen such that*

$$\frac{\alpha^2}{1-\frac{1}{\alpha^2}} = \frac{2SNR}{\ln N - \ln\left(\ln N\right) - \ln\zeta} \ , \tag{4.22}$$

then

$$\mathsf{E}\{\pi_{-\mathbf{1}}\} \geq N^{-\zeta} \ . \tag{4.23}$$

Value of α

Note that from (4.15d) it is clear that the larger β is, the larger $\pi_{-\mathbf{1}}$ is. Therefore, the range of α that gives a polynomially small probability to $\pi_{-\mathbf{1}}$ is

$$\frac{\alpha^2}{1-\frac{1}{\alpha^2}} \leq \frac{2\mathrm{SNR}}{\ln N - \ln\left(\ln N\right) - \ln\zeta}. \tag{4.24}$$

It can be shown that in the regime, $\mathrm{SNR} > 2\ln N$, the above quadratic inequality in α has two positive real solutions, $\alpha_+ \geq \alpha_-$, and that the inequality holds for all $\alpha \in [\alpha_-, \alpha_+]$.

We know that the larger α is, the faster the Markov chain mixes.[7] Therefore, it is reasonable that we choose the largest permissible value for α, i.e., α_+.

Figure 4.1 and Figure 4.2 show the values of α_+ and α_- as a function of SNR for systems with $N = 10$ and $N = 50$ when we have $\zeta = 1/\ln(N)$. The values of α_+ and α_- have also been plotted as a function of the system size N which is shown in Figure 4.3.

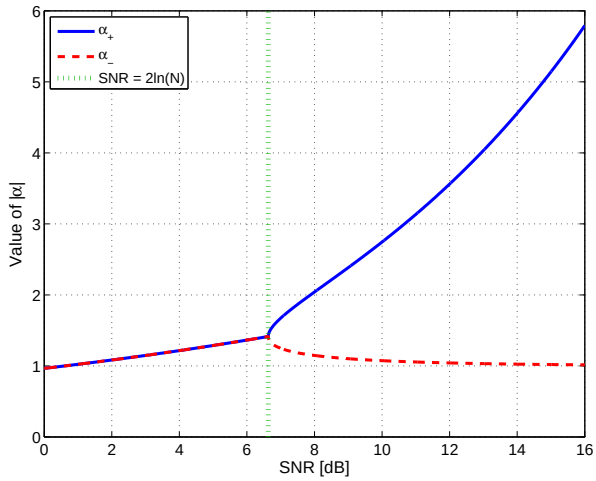


Figure 4.1: Value of α vs. SNR for system size $N = 10$.

Mixing time of Markov Chain

One open question is whether the Markov chain is rapidly mixing when using the strategy above for choosing α, however, the simulations presented in Section 4.1.4 seem to indicate this is the case. Furthermore, the simulations also suggest that the computed value of α is very close to the optimal choice, even in the case where the condition $\text{SNR} > 2\ln(N)$ is not satisfied.

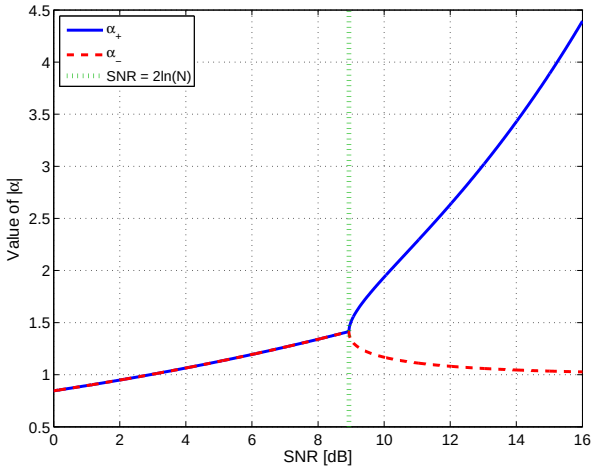


Figure 4.2: Value of α vs. SNR for system size $N = 50$.

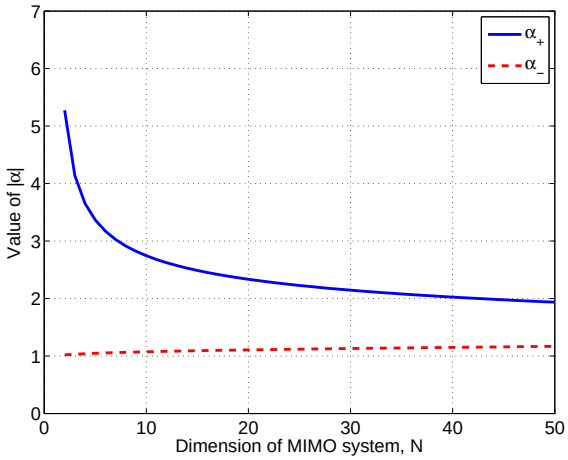


Figure 4.3: Value of α vs. system size N for SNR $= 10dB$.

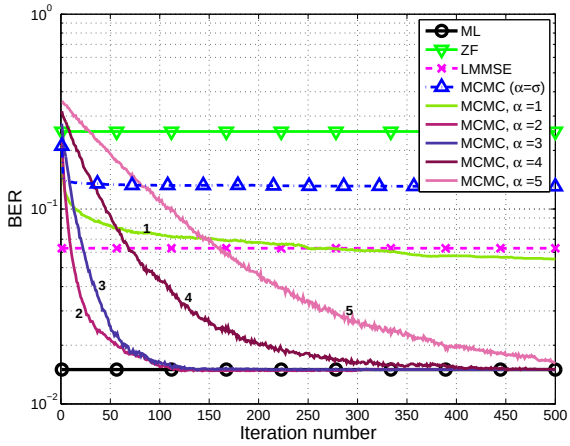


Figure 4.4: BER vs. iterations, 10×10 system. SNR = 10 dB. Theoretical value of $\alpha_+ = 2.7$.

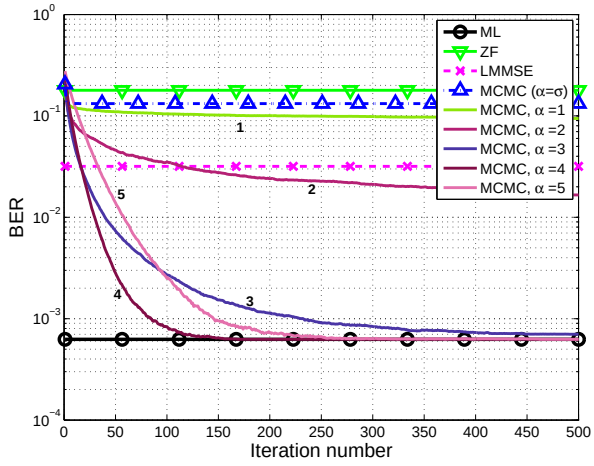


Figure 4.5: BER vs. iterations, 10×10 system. SNR = 14 dB. Theoretical value of $\alpha_+ = 4.6$.

4.1.4 Simulation Results

In this Section, we present simulation results for a MIMO $N \times N$ system with a full square channel matrix containing i.i.d. Gaussian entries. In Figure 4.4 and Figure 4.5, the BER of the Gibbs sampler, initialized with a random $\mathbf{x}$, has been evaluated as a function of the number of iterations in a 10×10 system using a variety of α values. Thereby, we can inspect how the parameter α affects the convergence rate of the Gibbs sampler. The performance of the ML, the Zero-Forcing (ZF), and the LMMSE detector has also been plotted to ease the comparison of the Gibbs sampler with these. It is seen that the Gibbs sampler outperforms both the ZF and the LMMSE detector after only a few iterations in all the presented simulations when the tuning parameter α is chosen properly. Furthermore, it is observed that the parameter α has a huge influence on the convergence rate and that the Gibbs sampler converges toward the ML solution as a function of the number of iterations. Here, it should be noted that the way we decode the symbol vector to a given iteration, is to select the symbol vector, which has the lowest cost function in all the iterations up to that point in time. The optimal value of α (in terms of convergence rate) is quite close to the theoretical values from Figure 4.1 of $\alpha_+ = 2.7$ and $\alpha_+ = 4.6$ at SNR's at 10 and 14 dB, respectively. It is also observed that the performance of the Gibbs sampler is significantly deteriorated if the temperature parameter is chosen based on the SNR (and, thereby, on the noise variance) such that $\alpha = \sigma \triangleq 1/\text{SNR}$. Thus, the latter strategy is clearly not a wise choice.
Figure 4.6 shows the BER performance for the MCMC detector for fixed number of iterations, $k = 100$. From Figure 4.6 we see that the SNR has a significant influence on the optimal choice of α given a fixed number of iterations.
The performance of the Gibbs sampler is also shown for a 20×20 and a 50×50 system, which represents a ML decoding problem of huge complexity where an exhaustive search would require $2^{20} \approx 10^6$ and $2^{50} \approx 10^{15}$ evaluations, respectively.
In Figure 4.7 the BER performance is plotted as a function of the number of iterations for the 20×20 system. Again, it is observed that the parameter α has a huge influence on the convergence rate. The BER as a function of the SNR has been plotted in Figure 4.8 for $k = 250$ iterations.

For the 50×50, system even the sphere decoder has an enormous complexity under moderate SNR.[8] Thus, it has not been possible to simulate the performance of this decoder within a reasonable time and we have, therefore, for the 50×50

[7] In general, there is a trade-off between faster mixing time of the Markov chain (due to an increase of α) versus slower encountering the optimal solution in steady-state. In fact, at infinite temperature, our algorithm reduces to a random walk in a hypercube, which mixes in $O(N \ln N)$ time [80].

[8] In fact, it can be shown that for $\text{SNR} = \mathcal{O}(\ln N)$, the lower bound on the complexity of the sphere decoder obtained in [45] is exponential.

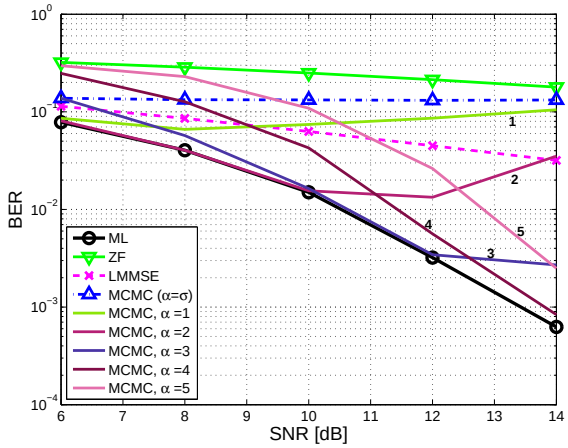


Figure 4.6: BER vs. SNR, 10×10 system. Number of iterations, $k = 100$.

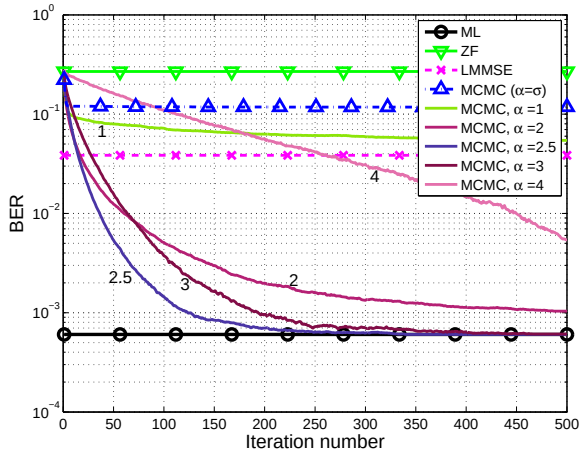


Figure 4.7: BER vs. iterations, 20×20 system. SNR = 12 dB. Theoretical value of $\alpha_+ = 3.1$

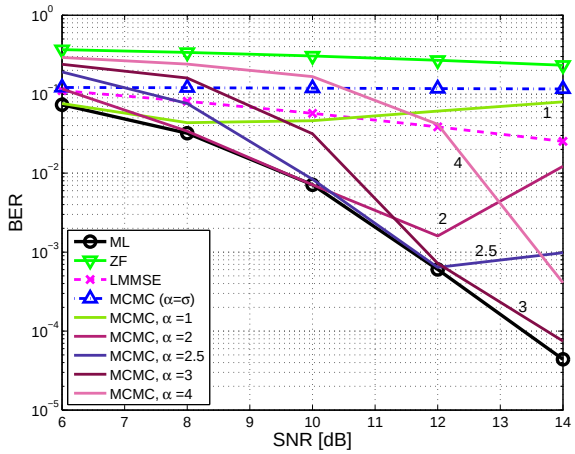


Figure 4.8: BER vs. SNR, 20×20 system. Num. of iter., $k = 250$.

Table 4.1: Complexity (MAC operations) of Sphere Detection (SD) and Gibbs Sampling (GS).

N	Method	SNR 6 dB	10 dB	14 dB
10	GS	$9.8 \cdot 10^3$	$10.9 \cdot 10^3$	$16.4 \cdot 10^3$
	SD	$10.0 \cdot 10^3$	$1.7 \cdot 10^3$	$1.5 \cdot 10^3$
20	GS	$4.6 \cdot 10^4$	$6.1 \cdot 10^4$	$15.5 \cdot 10^4$
	SD	$2.5 \cdot 10^7$	$8.3 \cdot 10^4$	$9.8 \cdot 10^3$
50	GS	$7.6 \cdot 10^5$	$9.5 \cdot 10^5$	$10.6 \cdot 10^5$
	SD	$\gg 1.9 \cdot 10^9$	$\gg 1.9 \cdot 10^9$	$37.7 \cdot 10^5$

system "cheated" a little by initializing the radius of the sphere to the minimum of either the norm of the transmitted symbol vector or the solution found by the Gibbs sampler. This has been done to evaluate the BER performance of the optimal detector. Figure 4.9 shows the BER curve as a function of the iteration number, while Figure 4.10 illustrates the BER curve vs. the SNR. From Figure 4.9, we see that there is a good correspondence between the simulated α and the theoretical value $\alpha_+ = 2.6$ obtained from Figure 4.2.

The average complexity (MAC pr. symbol vector) of the Gibbs sampler having a BER performance comparable with the ML detector is shown in Table 4.1. The SD has been included as a reference.[9] It is observed that the complexity of the Gibbs sampler is not affected by the SNR as much as the SD, which can also be seen from Figure 4.11 where the number of operations has been plotted

[9] It has not been possible to simulate the SD for a 50×50 system when $SNR \leq 10dB$ and, therefore, the complexity of $SNR = 12dB$ has been used as a lower bound.

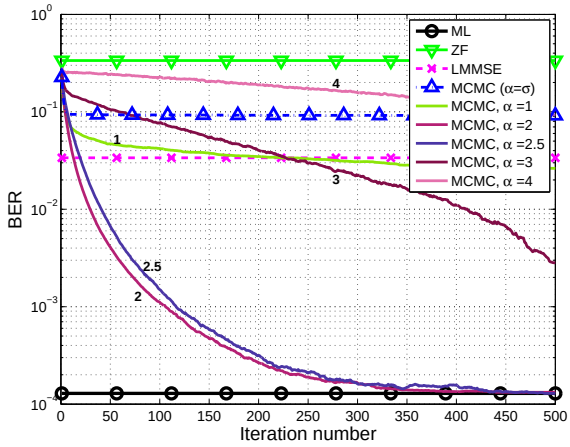


Figure 4.9: BER vs. iterations, 50×50 system. SNR = 12 dB. Theoretical value of $\alpha_+ = 2.6$

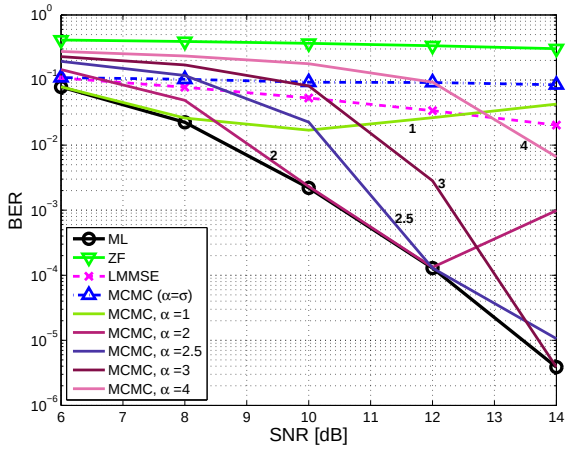


Figure 4.10: BER vs. SNR, 50×50 system. Num. of iter., $k = 500$.

as a function of the SNR for a 20×20 system.

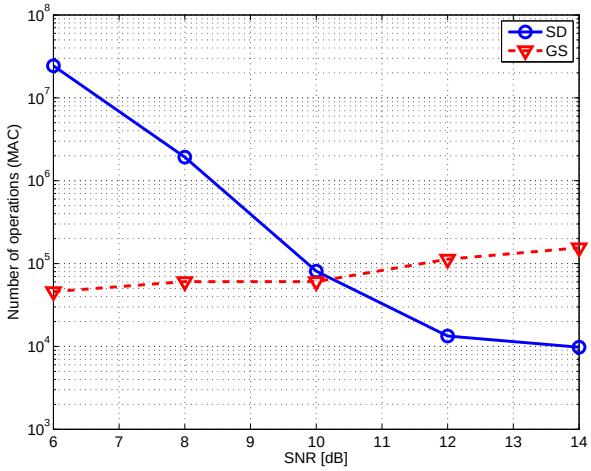


Figure 4.11: Complexity of SD and GS vs. SNR for a 20×20 system.

4.2 Summary

We have described and analyzed a way to compute the ML solution using Gibbs sampling. It has been shown that the method can be used for achieving a near-optimal and computationally efficient solution of the problem, even for systems having a huge dimension. The proposed MCMC method will, unlike simulated annealing techniques, have a fixed "temperature" parameter in all the iterations, with the property that after the Markov chain has mixed, the probability of encountering the optimal solution is only polynomial small (i.e. not exponentially small). We have computed the optimal (here largest) value of the temperature parameter that guarantees this. Simulation results reveal that the choice of the temperature parameter has an impact on how fast the GS finds the ML solution and show that our computed value gives a very good approximation to the optimal value of the GS. Furthermore, the simulation results suggest that the Markov chain is rapidly mixing. Thus, it has been observed that even in cases were ML detection using e.g. sphere decoding becomes infeasible, the Gibbs sampler can still offer a near-optimal solution using much less computations.

Chapter 5

Conclusion

In this thesis different approximate methods for detection in wireless communications have been treated, where one focus area has been the computationally complexity of the investigated methods. This is motivated by the fact that if a certain method should have any chance of being applied in an actual implementation the complexity has to be reasonable.
The main application field has been the existing 2G cellular system (such as GSM, EDGE, and EGPRS2), but also methods for general MIMO systems have been investigated.

It is proved that the QL-factorization of frequency-selective channels asymptotically provides the minimum-phase and all-pass filters. The exact convergence rate for the minimum-phase filter has been computed for a simple SISO length $L = 2$ system and an upper bound has been derived. This is used to approximate the convergence in systems of arbitrary length and, asymptotically, these results also generalize to MIMO systems. This makes it possible to view sphere detection as an adaptive variant of traditional reduced-state sequence estimation and, in that way, it provides a unifying framework for the two detection methods. Simulations have indicated that a significant reduction in complexity is obtained using a minimum-phase prefilter in front of a sphere detector. Thus, the proposed sphere detector is capable of obtaining near-optimal BER performance, even though it has a very limited complexity.
Moreover, a novel method for computing the minimum-phase filter and its

associated all-pass filter in a computationally efficient manner using fast QL-factorization has been presented. The method convergences asymptotically toward the true filters and has the convenient property of having a complexity depending on the required precision.

Markov Chain Monte Carlo Gibbs sampling has been proposed for MLSD in MIMO systems with a large number of transmit and receive dimensions. The proposed Gibbs sampler is novel as it uses a fixed "temperature" parameter in all the iterations, unlike simulated annealing techniques. Hereby, the detector will have the property that after the Markov chain has mixed, the probability of encountering the optimal solution is only polynomial small (i.e. not exponentially small). Further, the approximate optimal value of the temperature parameter that guarantees this convergence has been computed. Simulation results reveal that the choice of the temperature parameter has an impact on the performance of the Gibbs sampler, just as it was expected. Also, the simulations show that the computed value gives a quite good approximation to the optimal value of the Gibbs sampler.

Suggestions for Further Work

During the Ph.D. study several interesting ideas, which has not been treated in this thesis, have become apparent. Some interesting directions for future research would be the ones listed below.

Detection using QL-factorization

- Analyze more thoroughly under which conditions minimum-phase prefiltering provides a complexity reduction for sphere detection in time-invariant multipath channels. More specifically, it would be relevant to investigate what effect the distribution of the channel coefficients, the channel length, and the system size have on the complexity. Thereby, e.g. a system designer will have the possibility of evaluating whether or not prefiltering reduces the complexity.

- Investigate if fast QL-factorization methods can be extended to handle a slowly time-varying multipath channel. For this type of channel it will be unrealistic to perform an exact QL-factorization using fast methods. But it might be that under certain behavior in the time-variant filter coefficients the fast method can provide an approximate solution which is close enough to the actual factorization to be of interest for practical usage.

- Find an upper bound (instead of the approximate expression) for the convergence rate of SISO and MIMO systems of arbitrary length for the minimum-phase filter computation using the QL-factorization method. Such an upper bound will significantly strengthen the convergence analysis of this method.

Gibbs Sampling

- Generalize the results for the proposed Gibbs sampler such that they also cover higher order modulation formats and banded matrices. Such a generalization will be of interest for many communication systems.
- Investigate whether the underlying Markov chain in the Gibbs sampler mixes in polynomial time for an appropriate choice of temperature parameter. Theoretically, this will be a really important result, since it is then proved that the Gibbs sampler can solve a NP-hard problem in polynomial time with high probability over the channel matrices for a sufficiently large SNR.
- Examine how well the suggested Gibbs sampler estimates the soft-information. In many communication systems the detector should be capable of providing soft-information as an output and, therefore, it is relevant to examine how good estimates the Gibbs sampler can provide.
- Investigate how imperfect SNR and channel estimation affect the performance of the Gibbs sampler. Before the Gibbs sampler can be used in an actual communication system it should be designed such that it is robust to these kinds of effects.

Appendix A

On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation

---

Morten Hansen, Lars P. B. Christensen, and Ole Winther. On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation. *IEEE Global Telecommunications Conference (GLOBECOM)*. November 2007.

On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation

Morten Hansen and Ole Winther
Informatics and Mathematical Modelling,
Technical University of Denmark,
Building 321, DK-2800 Lyngby, Denmark,
E-mail: {mha, owi}@imm.dtu.dk

Lars P. B. Christensen
Nokia Denmark
Frederikskaj 5, DK-1790 Copenhagen V,
Denmark,
E-mail: lars.christensen@nokia.com

***Abstract*—In this paper, prefiltering techniques for Sphere Detection (SD) in frequency-selective channels are examined. It is shown that a link between QL-factorization of the channel matrix and minimum-phase prefiltering exists. As a result, it is possible to regard SD as a generalization of traditional reduced-state sequence estimation, providing a unifying framework for the two detection methods. It is illustrated how minimum-phase prefiltering or the Linear Minimum Mean-Square Error Decision Feedback Equalization (LMMSE-DFE) forward filter is capable of reducing the complexity of sphere detectors significantly, while still obtaining near-optimal performance. The significant reduction in complexity is obtained as prefiltering enables earlier decision making in SD. Simulations carried out in an EDGE system confirm that prefiltering leads to a considerable complexity reduction for sphere detectors.**

I. Introduction

Throughout the recent years, Sphere Detection (SD) has gained much attention in the literature. Wireless communication is one of the fields where SD has been proposed. This paper deals with SD in frequency-selective (multipath) channels, where the complexity of the optimal detectors is enormous when higher order modulation techniques are used. SD addresses the issue of performing the Closest Lattice Point Search (CLPS) in a computationally efficient way. CLPS is equivalent to Maximum Likelihood Sequence Detection (MLSD), which normally leads to exhaustive search, having a complexity that is exponential with respect to both the length of the channel and number of transmit dimensions. It has been proved that the average complexity of SD is significantly smaller than exhaustive search detectors, over a wide range of Signal-to-Noise Ratios (SNR)s [1]. However, the complexity is in the worst case still exponential [2].

In a real implementation of a detector (e.g. in a mobile phone), it is necessary to upper-bound the complexity. The exponential complexity of SD is in conflict with this implementation issue, and therefore it is relevant to study the performance of sphere detectors with upper bounded complexity.

Concentrating the energy of the channel impulse response in the first taps enables early decision making in the trellis search. Thus, when a prefiltering stage is placed in front of the sphere detector, the complexity is likely to be reduced. Using a minimum-phase filter, the energy of the channel impulse response will obtain this desired property.

The rest of the paper is organized as follows. Section II presents the signal model, while the basic concepts of SD and additional pruning techniques are treated in Section III. Section IV establish a link between minimum-phase prefiltered Reduced-State Sequence Estimation (RSSE) and SD. Furthermore, prefiltering using the Linear Minimum Mean-Square Error Decision Feedback Equalization (LMMSE-DFE) forward filter is considered. The simulation results of the examined detectors are presented in Section V and a conclusion can be found in Section VI.

Throughout the paper bold lowercase letters (e.g. $\mathbf{x}$) denote column vectors, while bold uppercase letters denote matrices (e.g. $\mathbf{H}$). The matrix transpose is denoted $(\cdot)^T$, while $(\cdot)^H$ is the Hermitian matrix transpose.

II. Signal Model

Consider an uncoded frequency-selective Multiple-Input Multiple-Output (MIMO) channel with channel length L. The channel can be modeled as

$$\mathbf{x}_n = \sum_{l=0}^{L-1} \mathbf{H}_l \mathbf{s}_{n-l} + \mathbf{z}_n \ , \tag{1}$$

where $\mathbf{x}_n \in \mathbb{C}^{N_R}$ is the received signal at time index $n = \{1, 2, \ldots, K+L-1\}$. K is the length of the transmit sequence and N_R and N_T (assuming $N_R \geq N_T$) denote the number of receive and transmit dimensions, respectively. The matrix $\mathbf{H}_l \in \mathbb{C}^{N_R \times N_T}$ contains the coefficients of the channel impulse response, while $\mathbf{s}_n \in \Omega^{N_T}$ is the vector of transmitted symbols at time n, each symbol belonging to the complex-valued alphabet Ω. $\mathbf{z}_n$ represents the Additive White Gaussian Noise (AWGN) term with the variance σ^2, i.e. $\mathbf{z}_n \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$.

Using matrix notation, the signal model in (1) can be expressed as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{z} \ , \tag{2}$$

where $\mathbf{x} = \left[\ \mathbf{x}_1^T, \ \mathbf{x}_2^T, \ \ldots, \ \mathbf{x}_{K+L-1}^T \ \right]^T$. Let $M \triangleq K \cdot N_T$, then $\mathbf{s}$ is the M-dimensional vector containing the complex symbols, i.e. $\mathbf{s} = \left[\ \mathbf{s}_1^T, \ \mathbf{s}_2^T, \ \ldots, \ \mathbf{s}_K^T \ \right]^T$. $\mathbf{H}$ is a block banded block Toeplitz channel convolution matrix

having the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{H}_{L-1} & \ddots & \ddots & \mathbf{H}_0 \\ \mathbf{0} & \mathbf{H}_{L-1} & \ddots & \mathbf{H}_1 \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{L-1} \end{bmatrix} ,$$

leading to a block-fading channel model. To ease the notation let $N \triangleq N_R(K+L-1)$, leading to $\mathbf{H} \in \mathbb{C}^{N \times M}$. MLSD can now be expressed as

$$\hat{\mathbf{s}} = \min_{\mathbf{s} \in \Omega^M} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|^2 . \tag{3}$$

This detection problem is considered in the rest of the paper.

III. SD For Frequency-Selective Channels

In SD the search problem given in (3) is transformed to an equivalent problem, which often requires less computations [3]. This is done by performing either a QR- or QL-factorization of the channel matrix, $\mathbf{H} = \mathbf{Q}\mathbf{L}$, where $\mathbf{Q}$ is a $N \times N$ unitary matrix and $\mathbf{L}$ (or $\mathbf{R}$) is an $N \times M$ lower (or upper) triangular matrix with positive real diagonal elements. Thus, the only difference between $\mathbf{R}$ and $\mathbf{L}$ is the location of the non-zero elements. The QL-factorization is preferred here because it is more intuitive to use from a filtration point of view since the lower triangular structure corresponds to a causal system. It should be noted though that both a QR- and a QL-factorization leads to the same result for SD.

When $\mathbf{H}$ is QL-factorized, (3) can be rewritten as

$$\hat{\mathbf{s}} = \min_{\mathbf{s} \in \Omega^M} \|\tilde{\mathbf{x}} - \mathbf{L}\mathbf{s}\|^2 , \tag{4}$$

due to $\mathbf{Q}$ being unitary with $\tilde{\mathbf{x}} \triangleq \mathbf{Q}^H\mathbf{x}$. Furthermore, the QL-factorization is directly linked to Cholesky factorization as we require that $\mathbf{L}$ is the Cholesky factor of $\mathbf{H}^H\mathbf{H}$, i.e.

$$\mathbf{H}^H\mathbf{H} = \mathbf{L}^H\mathbf{L} , \tag{5}$$

where $\mathbf{L}$ is positive definite. The triangular structure makes it feasible to examine a single dimension at a time, and find lattice points located inside a hyper-sphere with a specified radius. Actually, several radii, which increase as a function of the examined dimension, can be applied in SD. Thus, for each dimension a radius is specified, leading to a complexity reduction compared to the case of a single radius. Especially when the dimension of $\mathbf{L}\mathbf{s}$ is large, a significant complexity reduction is obtained. However, using increasing radii, we are no longer guaranteed to find the closest lattice point. A statistically sound method for computing the radii is presented in [4].

When the transmitted (complex) symbols lie on a lattice structure, which e.g. is the case for quadrature amplitude modulated signals, the real and imaginary parts of the symbols are normally bounded one at a time. In the case of real-valued symbols (and assuming $N_R = N_T = 1$), the bounding of the first symbol is given as

$$\frac{-r + \tilde{x}_1}{l_{11}} \leq s_1 \leq \frac{r + \tilde{x}_1}{l_{11}} , \tag{6}$$

where r denotes the radius of the sphere for the current dimension and l_{ij} denotes the (i,j)-th entry of $\mathbf{L}$.

In the case of e.g. Phase Shift Keying (PSK), the bounding is carried out in the complex plane, where the intersection of two circles (i.e. the ones that represent the symbols and the sphere, respectively) is determined.

Symbols which lie inside a given sphere are assumed to be potential solutions to (4), and the most likely symbols (being the ones that are closest to the received point) are traditionally examined first, using the Schnorr-Euchner enumeration [5].

Since the channel convolution matrix has a lower block triangular form, it is possible to perform SD without having to do a full QR- or QL-factorization. When the block size of $\mathbf{H}_l$ is one (i.e. $N_R = N_T = 1$), it is actually not required to do a QL-factorization, since $\mathbf{H}$ already has a lower triangular structure. The bounding in SD is then applied directly on the frequency-selective channel matrix. When the block size is larger than one, a lower triangular structure can be obtained conveniently by

$$\tilde{\mathbf{H}} = \left(\mathbf{I} \otimes \mathbf{Q}_0^H\right)\mathbf{H} , \tag{7}$$

where $\mathbf{H}_0 = \mathbf{Q}_0\mathbf{L}_0$ and $\otimes$ denotes the Kronecker product. Thus, it is possible to obtain the lower triangular structure of $\tilde{\mathbf{H}}$, by multiplication of $\mathbf{Q}_0^H$ with each sub-matrix in $\mathbf{H}$. However, as will be illustrated in Section IV and Section V, it is often preferable to perform the full QL-factorization of $\mathbf{H}$ anyway, since the computational complexity is likely to be reduced.

In [6] it is shown that it is possible to apply SD on top of the Viterbi algorithm. Thus, the MLSD can be obtained by examining only the states in the trellis diagram, which lie inside the sphere. Alternatively, the SD algorithm can be combined with the Maximum A Posteriori (MAP) detector to obtain near-optimal symbol-by-symbol detection by forming approximate bit posteriors.

In an implementation of the SD algorithm, it is necessary to be able to specify the maximum allowed complexity since in the worst case it is exponential. By using the Schnorr-Euchner search strategy, it is possible to specify the maximum number of states, which are allowed to be examined in the trellis diagram. Thus, only the most likely paths through the trellis diagram are considered, leading to further reduction in complexity. Likewise, it is possible to specify the maximum number of state transitions allowed from a given state. However, both of these methods are of course suboptimal, but can be a necessary compromise.

IV. Minimum-Phase Prefiltered RSSE

The spectral factorization theorem states that the spectrum of any Linear Time-Invariant (LTI) system can be factorized

into minimum-phase components [7]. Furthermore, a generalization of spectral factorization states that any linear filter can be split into an all-pass filter and the minimum-phase filter found by spectral factorization [8]. As shown in [9], the minimum-phase filter has the convenient property that it provides the highest possible energy concentration in the first taps and thus, enabling earlier decision making in the trellis diagram [1]. For that reason a minimum-phase preprocessing stage is typically also found in front of a detector relying on RSSE. This stage would filter the received signal with the conjugate of the all-pass filter related to the channel, leading to a minimum-phase channel impulse response.

The finite-length equivalent of the spectral factorization is given by the Cholesky decomposition of covariance matrices [10]. However, a generalization of splitting a LTI system into an all-pass and a minimum-phase filter to systems of finite-length appears to be unknown. It is explained in the next subsection that such an extension exists and is given by the QL-factorization.

A. Minimum-Phase Prefiltering Implements QL-Factorization

Due to the Toeplitz structure of $\mathbf{H}$ it is ensured that $\mathbf{H}^H\mathbf{H}$ has a Toeplitz form with bandwidth $(2L-1)N_T$ (the bandwidth represents the number of non-zero entries in a row of the Toeplitz matrix). As a result of (5), this will also be the case for $\mathbf{L}^H\mathbf{L}$. Now assume that $M \to \infty$, while the channel length is kept fixed. This will lead to a certain structure in $\mathbf{L}$, where each row will be a shifted version of each other, which is precisely given by the spectral factorization [10].

For a moment regard the unitary matrix as a filter. Since this is the only filter which does not change the spectrum (or the covariance), the filter must correspond to an all-pass filter. This is substantiated by recalling that a unitary matrix leaves the statistics of a vector unchanged. Thus, a unitary matrix must be the matrix equivalent to the all-pass filter. Since $\mathbf{Q}$ is the only unitary matrix that links $\mathbf{H}$ with $\mathbf{L}$, $\mathbf{Q}$ is the all-pass filter associated with the spectral factorization. Let again $M \to \infty$, leading to vanishing boundary effects. The rows of $\mathbf{Q}$ will be shifted versions of each other and will correspond to the all-pass filter associated with the minimum-phase filter.

In the case of finite-length systems, the Toeplitz structures of $\mathbf{Q}$ and $\mathbf{L}$ are no longer ensured due to boundary conditions. However, for large systems these conditions have a less significant role, and therefore, the results above will converge towards their asymptotic values in the middle of the matrices, where the influence of the boundary conditions is less. Thus, there is a direct link between minimum-phase prefiltering and QL-factorization of the channel matrix, $\mathbf{H}$.

Based on the link between minimum-phase prefiltering and the QL-factorization of frequency-selective channels, it is possible to relate SD with minimum-phase prefiltered RSSE. One difference between SD and RSSE is though that the decision of disregarding states in SD is not made until a considerable extent of confidence has been obtained. This is because traditional SD does not have a limit on the maximum number of examined states, but only prunes away states when they are outside the sphere. Through the connection with the QL-factorization, a reasonable method of using minimum-phase prefiltering in sphere detection can be constructed, which leads to tighter bounds in SD.

[1] An earlier decision making in the trellis diagram is possible since the diagonal elements (and the ones that are close to it) of the lower triangular matrix, $\mathbf{L}$, will have larger numerical values after prefiltering. Thus, from (6) it is clear that tighter bounds are obtained.

B. LMMSE-DFE prefiltering

Another method for obtaining tighter bounds in SD is proposed in [11]. Here, a LMMSE-DFE prefilter is obtained by reformulating the detection problem in (3), which is done by augmenting the channel matrix

$$\hat{\mathbf{H}} \triangleq \begin{bmatrix} \mathbf{H} \\ \sigma\mathbf{I} \end{bmatrix} = \hat{\mathbf{Q}}\hat{\mathbf{L}}_1 = \begin{bmatrix} \hat{\mathbf{Q}}_1 \\ \hat{\mathbf{Q}}_2 \end{bmatrix} \hat{\mathbf{L}}_1 \ , \tag{8}$$

where the unitary matrix and the lower triangular matrix are defined in a slightly different way, since $\hat{\mathbf{Q}} \in \mathbb{C}^{(N+M)\times M}$ and $\hat{\mathbf{L}}_1 \in \mathbb{C}^{M\times M}$. The columns of $\hat{\mathbf{Q}}$ will be unitary and the upper $N \times M$ part of $\hat{\mathbf{Q}}$ is denoted $\hat{\mathbf{Q}}_1$. In this reformulation of the detection problem, $\hat{\mathbf{Q}}_1$ represents the forward LMMSE-DFE filter, while $\hat{\mathbf{L}}_1$ is the backward filter [11]. The detection problem in (3) can be altered to

$$\hat{\mathbf{s}} = \min_{\mathbf{s}\in\Omega^M} \|\mathbf{y} - \hat{\mathbf{L}}_1\mathbf{s}\|^2 \ , \tag{9}$$

where $\mathbf{y} \triangleq \hat{\mathbf{Q}}_1^H\mathbf{x}$. It should be emphasized that the detection problem in (3) and (9) are not equivalent, since the columns of $\hat{\mathbf{Q}}_1$ are not unitary.

An advantage of the change of detection problem is that $\hat{\mathbf{L}}_1^H\hat{\mathbf{L}}_1 = \mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}$. Thus, the change of the detection problem ensures that the channel matrix will be better conditioned and that $\mathrm{rank}(\hat{\mathbf{H}}) = M$. However, by altering the detection problem the noise statistics is changed, such that the modified noise is non-Gaussian and data-dependent. The noise will still be white with variance σ^2 and (9) is, therefore, only to some extent assumed to be suboptimal [11].

V. Simulation Results

The simulation results presented in this section are carried out in an EDGE system. The frame format and modulation type used are identical to that specified in the EDGE standard, e.g. a $3\pi/8$ rotated 8-PSK signal is used in the modulation. It is assumed that frequency hopping is made between each received burst, and that the channel impulse response and noise variance are perfectly known. AWGN is added to account for any thermal noise. Only single user detection is considered in the simulations and only a single receive antenna is assumed to be available. The SNR is defined as the average received signal power from the user, divided by the noise power. To exploit the diversity in the channel model, the oversampling factor in the channel is set to $N_{sps} = 2$ in respect to the symbol rate. Due to this oversampling, the received signal is jointly prefiltered, before it is passed on to the detector, leading to $N_R = 1$ in the detector. The channel models used in the

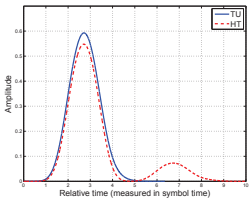


Fig. 1. Channel profiles of Typical Urban and Hilly Terrain (including the transmit pulse shaping).

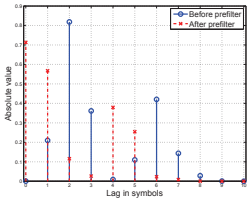


Fig. 2. The absolute value of channel coefficients with and without minimum-phase prefilter for the Hilly Terrain profile.

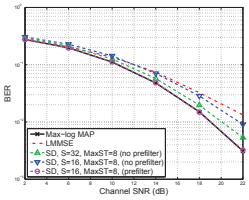


Fig. 3. BER performance in TU with and without prefiltering. S is the maximum number of states in the trellis diagram and MaxST is the maximum number of state transitions from a given state.

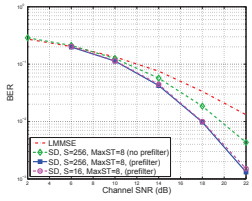


Fig. 4. BER performance in HT with and without minimum-phase prefiltering.

simulations are the Typical Urban (TU0) and the Hilly Terrain (HT0) profiles defined in the GSM specifications.

In Fig. 1, the channel profiles of TU and HT including the transmit pulse shaping are shown. In Fig. 2, an example of the channel coefficients of the HT profile is shown (N_{sps} is here set to 1). The coefficients obtained using a minimum-phase prefilter are also shown to illustrate the effect of the filter. It is observed that the number of taps needed for modeling the channel properly is approximately $L = 7$, when there is no prefilter. Using a the minimum-phase prefilter the channel length can be reduced to $L = 6$. The optimal detector would in the latter case require a search in a trellis diagram of $8^{6-1} \approx 33 \cdot 10^3$ states per symbol, which is an unacceptable high complexity.

In the simulations presented, SD is combined with the max-log MAP receiver to obtain approximate bit posteriors. Furthermore, the approach of specifying the maximum number of allowed states in the trellis diagram (described in Section III) has been used in all the simulations considering SD.

The increasing radii are found from $\mathrm{P}\left(|\mathbf{z}_{1:n}|^2 > r_n^2\right) = \varepsilon^{2i}$, where i is the number of times the algorithm is restarted (which is done if no points are found inside the sphere), and $\mathbf{z}_{1:n} = \left[\ \mathbf{z}_1^T, \quad \ldots \quad \mathbf{z}_n^T\ \right]^T$. This leads to a probability $1-\varepsilon^{2i}$ of finding the transmitted point inside the n-dimensional hyper-sphere, and due to the AWGN assumption, the radii can be determined using the Chi-square distribution.

The filter coefficients of the prefilter and the associated channel impulse response, have been calculated by extracting a part of one of the rows in the middle of the unitary matrix and the lower triangular matrix, respectively. In some of the figures the performance of the LMMSE detector is also given, which serves as a reference.

In Fig. 3, the Bit Error Rate (BER) performance of the proposed sphere detector is presented for the TU profile. To illustrate the effect of prefiltering, BER curves are given for the same simulation setup, but with and without minimum-phase prefiltering. In the labels, "S", denotes the maximum number of allowed *states* in the trellis diagram (e.g. S = 16, indicates that 16 states are allowed). Furthermore, "MaxST" denotes the maximum number of allowed *state transitions* for a given state (e.g. MaxST = 8, indicates that 8 state transitions from each state are allowed). In Fig. 3, the performance of the max-log MAP detector is also included. The detector relying on minimum-phase prefiltered SD with at most 16 states in the trellis diagram, is capable of obtaining a performance which is comparable with the max-log MAP. This is not the case when the prefilter is not used.

In Fig. 4, the performance of the detectors for the HT profile is shown. From the figure it is clear that prefiltering gives a significant improvement in BER performance. Furthermore, it is observed that the complexity can be reduced considerably without degrading the BER performance.

Fig. 5 presents a comparison between the sphere detector using either the minimum-phase or the LMMSE-DFE prefilter. From the figure it is observed that the BER performance is almost identical for the two prefilters. However, when S = 16 and

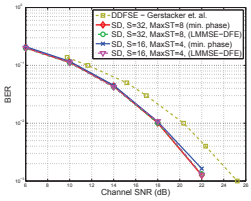


Fig. 5. BER performance for SD using different prefilters in HT.

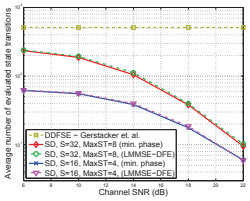


Fig. 6. The average number of evaluated state transitions per symbol.

MaxST = 4 the LMMSE-DFE prefilter gives a slightly better performance at low SNR (approximately 0.15 dB). This is probably due to the reformulation of detection problem, which has a regularizing effect on the channel matrix.

To be able to evaluate the performance of the proposed detector it has been compared with the best detector (with respect to BER performance) presented in [12], which relies on Delayed Decision Feedback Sequence Estimation (DDFSE). The DDFSE detector uses a trellis diagram of 64 states and minimum-phase prefiltering is also applied. In [12], a similar simulation setup is found, which makes a comparison possible. The performance curve of the detector has been directly reproduced from [12]. Thus, there may be minor differences in the simulation setup, even though the general setup is the same. The SD using prefilter seems to outperform the detector in [12] significantly, both with respect to BER performance and complexity (i.e. the number of examined states).

The complexity of the detectors used to obtain the simulation results presented in Fig. 5 is illustrated in Fig. 6. The complexity is expressed as the average number of *state transitions* in the trellis diagram. Recall, that the channel length needed to model the HT channel properly (using prefilter) is $L = 6$, leading to $8^6 \approx 2.6 \cdot 10^5$ evaluated state transitions per symbol for the max-log MAP detector. Thus, compared to the max-log MAP detector, the SD detector is capable of pruning the trellis diagram efficiently, and thereby obtaining a huge reduction in complexity. In Fig. 6, it is observed that the complexity is upper bounded by the product of the specified number of allowed states and state transitions (denoted S and MaxST). From the figure, it is seen that SD using minimum-phase or LMMSE-DFE prefiltering leads to almost the same complexity.

Based on the presented simulations, it is observed that near-optimal performance is obtained, while still having a reasonable complexity. Thus, in contrast to other sphere detectors (e.g. [1] and [6]) the worst-case complexity for the proposed method will no longer be exponential. The price paid for upper bounding the complexity is that the proposed method might not find the ML solution (especially at low SNR). This is because the ML path in the trellis diagram might be pruned away since there are many candidates inside the sphere at low SNR, but only a limited number of these is used in the search.

VI. Conclusion

A link between minimum-phase prefiltering and the QL-factorization of frequency-selective channels has been established. This makes it possible to relate SD with minimum-phase prefiltered RSSE. Simulations have indicated that a significant reduction in complexity is obtained using either a minimum-phase prefilter or a LMMSE-DFE forward filter in front of a sphere detector. Thus, the proposed sphere detector is capable of obtaining near-optimal BER performance, even though it has a very limited complexity. The performance of SD using either the minimum-phase or the LMMSE-DFE prefilter has been evaluated. The two prefilters lead to almost the same performance, although the LMMSE-DFE forward filter seems to give a slightly better performance at low SNR. The complexity of SD using the two prefilters is likewise similar.

References

[1] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm. I. Expected Complexity," *IEEE Trans. on Signal Processing*, vol. 53, pp. 2806–2818, Aug. 2005.

[2] J. Jaldén and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Trans. on Signal Processing*, vol. 53, pp. 1474–1484, Apr. 2005.

[3] M. O. Damen, H. E. Gamal, and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," *IEEE Trans. on Info. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[4] R. Gowaikar and B. Hassibi, "Statistical Pruning for Near-Maximum Likelihood Decoding," *IEEE Trans. on Signal Processing*, vol. 55, pp. 2661–2675, Jun. 2007.

[5] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," *Mathematical Programming*, vol. 66, pp. 181–199, 1994.

[6] B. Hassibi, H. Vikalo, and U. Mitra, "Sphere-Constrained ML Detection for Frequency-Selective Channels," in *ICASSP'03*, vol. 4, 2003, pp. 1–4.

[7] A. H. Sayed and T. Kailath, "A Survey of Spectral Factorization Methods," *Numerical Linear Algebra with Applications*, vol. 8, pp. 467–496, Jul. 2001.

[8] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice Hall, 1995.

[9] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback," *IEEE Trans. on Communications*, vol. 36, pp. 13–20, Jan. 1988.

[10] N. Al-Dhahir and J. M. Cioffi, "MMSE Decision-Feedback Equalizers: Finite-Length Results," *IEEE Trans. on Info. Theory*, vol. 41, pp. 961–975, Jul. 1995.

[11] A. Murugan, H. Gamal, M. Damen, and G. Caire, "A Unified Framework for Tree Search Decoding: Rediscovering the Sequential Decoder," *IEEE Trans. on Info. Theory*, vol. 52, pp. 933–953, Mar. 2006.

[12] H. Gerstacker and R. Schober, "Equalization Concepts for EDGE," *IEEE Trans. on Wireless Communications*, vol. 1, pp. 190–199, Jan. 2002.

Appendix B

Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization

---

Morten Hansen and Lars P. B. Christensen. Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. April 2009.

EFFICIENT MINIMUM-PHASE PREFILTER COMPUTATION USING FAST QL-FACTORIZATION

Morten Hansen

Technical University of Denmark,
Informatics and Mathematical Modelling,
Build. 321, DK-2800 Lyngby, Denmark,
E-mail: mha@imm.dtu.dk

Lars P. B. Christensen

Nokia Denmark
Frederikskaj 5,
DK-1790 Copenhagen V, Denmark,
E-mail: lars.christensen@nokia.com

ABSTRACT

This paper presents a novel approach for computing both the minimum-phase filter and the associated all-pass filter in a computationally efficient way using fast QL-factorization. A desirable property of this approach is that the complexity is independent of the size of the matrix being QL-factorized. Instead, the complexity scales with the required precision of the filters as well as the filter length.

Index Terms— Communications, prefiltering, minimum-phase systems, fast QL-factorization.

1. INTRODUCTION

The minimum-phase filter has an important role in general signal processing theory, see e.g. [1], and one application thereof is communication systems when higher-order modulation schemes over multipath channels are used. In such systems, optimal sequence detection can be obtained using Maximum-Likelihood Sequence Estimation (MLSE), but MLSE typically require an unacceptable high complexity for channels with large delay spread (i.e. long impulse responses). Therefore, other suboptimal techniques such as delayed decision feedback, or reduced-state sequence estimation, will often be used in such systems [2]. To obtain reliable detection using these techniques, both the minimum-phase and the associated all-pass filter are used.
In this paper we describe a new approach for efficiently computing the minimum-phase filter and the all-pass filter by performing a fast QL-factorization of the channel matrix. The paper is organized as follows; In Section 2 we present the signal model and Section 3 describes the connection between the minimum-phase filter and the QL-factorization. In Section 4 we illustrate how the fast QL-factoization can be utilized for time-invariant channels, while the simulation results are found in 5. Finally, Section 6 contains some concluding remarks.

2. SYSTEM MODEL

Consider a time-invariant Single-Input Single-Output (SISO) system[1], which can be described by the Finite Impulse Response (FIR) filter, $\mathcal{H}$, having the length L. The output signal $y_k \in \mathbb{C}$ at time index k can be expressed as

$$y_k = \sum_{l=0}^{L-1} h_l x_{k-l} \,, \tag{1}$$

where $x_k \in \mathbb{C}$ is the input signal at time index $k = \{1, 2, \ldots, N+L-1\}$, N is the length of the input sequence, and $h_l \in \mathbb{C}$ denotes the l'th tap in the impulse response. Using matrix notation, the system model in (1) can be formulated as

$$\mathbf{y} = \mathbf{H}\mathbf{x} \,, \tag{2}$$

where $\mathbf{y} = \left[\begin{array}{cccc} y_1, & y_2, & \ldots, & y_{N+L-1} \end{array} \right]^T$ and $\mathbf{x} = \left[\begin{array}{cccc} x_1, & x_2, & \ldots, & x_N \end{array} \right]^T$. To ease the notation let $M \triangleq (N+L-1)$, leading to $\mathbf{y} \in \mathbb{C}^M$. Due to the time-invariant property of the filter, $\mathbf{H} \in \mathbb{C}^{M \times N}$ will be a banded Toeplitz convolution matrix having the form

$$\mathbf{H} \triangleq \left[\begin{array}{cccc} h_0 & 0 & \cdots & 0 \\ \vdots & h_0 & \ddots & \vdots \\ h_{L-1} & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & h_0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{L-1} \end{array} \right] .$$

In the analysis of the filter characteristic, it is often useful to z-transform the channel impulse response [1], which leads to

$$H(z) = \sum_{l=0}^{L-1} h_l z^{-l} \,. \tag{3}$$

[1] Results presented may be directly extended to Multiple-Input Multiple-Output (MIMO) systems, but this is outside the scope of this paper.

A classical way of obtaining the minimum-phase filter, H_{min}, is by using the root method of spectral factorization, where we first find roots in the polynomial given in (3), and reflect the roots located outside the unit circle, into the circle, [1], [3]. Based on the roots inside and on the unit circle, a new polynomial can be computed in the z-domain, which represents the minimum-phase filter. There exists however several other spectral factorization methods which among others is described in [4]. In many applications (e.g. in communications) we also need the associated all-pass filter, which is used to prefilter the output signal, $\mathbf{y}$, such that the modified output signal matches the minimum-phase filter. As finding the minimum-phase and all-pass filters can be computationally expensive, approximative methods having lower complexity may be of practical interest [2].

3. QL-FACTORIZATION AND THE MINIMUM-PHASE FILTER

It is well-known that the minimum-phase filter can be obtained in various ways, [4] and recently, it has been discovered that the minimum-phase filter and its associated all-pass filter can be obtained by performing a QL-factorization of the channel matrix, $\mathbf{H}$, [5], [6]. When we perform the factorization,

$$\mathbf{H} = \mathbf{Q}\tilde{\mathbf{L}} = \mathbf{Q}\begin{bmatrix} \mathbf{0}_{(M-N)\times N} \\ \mathbf{L} \end{bmatrix}, \qquad (4)$$

we require that the $N \times N$ lower triangular matrix, $\mathbf{L}$, corresponds to the Cholesky factor of $\mathbf{H}^H\mathbf{H}$, meaning that $\mathbf{L}$ is positive definite and contains real-valued positive diagonal elements (assuming that $rank(\mathbf{H}) = N$). Since we perform a factorization of a banded Toeplitz matrix, each row in $\mathbf{L}$ will be a shifted version of each other as $\{M, N\} \to \infty$, and each row is precisely given by the spectral factorization, [7]. Likewise, the $M \times M$ unitary matrix $\mathbf{Q}$ will be the matrix equivalent of the all-pass filter and again each column of $\mathbf{Q}$ will be a shifted version of each other. Furthermore, it can be seen that each of these columns will correspond to the all-pass filter associated with the minimum-phase filter. For a detailed description of this, see [5], [6].

In the finite length case, each row of $\mathbf{L}$ (column of $\mathbf{Q}$) will not be exactly the same, but as can be seen in [6], the values in each row of $\mathbf{L}$ will converge toward the true minimum-phase filter as a function of the row number[2], likewise the columns of $\mathbf{Q}$ will converge toward the associated all-pass filter. Thus, the accuracy of the estimated filter coefficients (compared to the true filters) depends on where in $\mathbf{L}$ and $\mathbf{Q}$ we take out the filter coefficients.

4. FAST QL-FACTORIZATION

When general methods are used to compute the QL-factorization it requires $\mathcal{O}\left(N^3\right)$ operations, [8], but for Toeplitz matrices there exist methods with lower computational complexity. Different methods have been proposed for performing the fast QL-factorization[3] [8], [9], [10], each of which has different numerical properties and slightly different complexity as well. They do however all use the shift-invariance property of Toeplitz matrices to partition it in two ways, and it is this partitioning that leads to the low complexity schemes. In [8], the QL-factorization can be performed using $13MN + 6N^2$ operations for general $M \times N$ Toeplitz matrices, while the method proposed in [10] require $13MN + 6.5N^2$. The methods described in [8], [9], and [10] all deal with real-valued matrices, but the results can be extended to be valid over the complex field, [10]. To extend the method in [8] to handle complex numbers, will however require another type of rank-1 downdating, which is described in [11]. The methods can also be extended to handle block Toeplitz matrices for the general MIMO case as well, [12].

The fast QL-factorization computes a single row of $\mathbf{L}$ (or column of $\mathbf{Q}$) at a time, which is a great advantage when the QL-factorization is used for prefilter computation. This is due to the fact that each row of $\mathbf{L}$ converges toward the true minimum-phase filter, which implies that we can stop the computation of the rows in $\mathbf{L}$ once we have obtained the required precision of the filter coefficients. Likewise, we only need to compute a certain fraction of the columns in $\mathbf{Q}$ to obtain the required precision of the all-pass filter. Thus, by using the fast QL-factorization to compute the filters, the complexity no longer scales with the size of the matrix, $\mathbf{H}$, but depends on the required precision. The number of rows in $\mathbf{L}$ (and thus columns in $\mathbf{Q}$) which is used to obtain the estimated minimum-phase and all-pass filters, is referred to as the number of iterations, n.

The complexity of the fast QL-factorization can be reduced even further, using the fact that the Toeplitz channel matrix, $\mathbf{H}$, contains at most L non-zero elements in each row. Thus, using the method described in [8] and the rank-1 downdate given in [11], we can compute of each row in $\mathbf{L}$ using $5L + 7$ complex operations and two square root computations. On top of that we also need to take into account the initialization step, which among others determines the bottom row of $\mathbf{L}$ [4], requiring $(L-1)L/2 + 4L$ complex operations and two square root computations. Thus, if the required precision of the minimum-phase estimate can be obtained using n iterations, the computational complexity will be

$$\mathcal{O}_{min} = (n-1)\cdot(5L+7) + (L-1)L/2 + 4L\,, \qquad (5)$$

complex operations plus $n + 1$ square root operations. Each of the last $L_{ap} - L$ columns of $\mathbf{Q}$ require $(L + i)(i + 1)$ operations where $i = \{0, \ldots, L_{ap} - L - 1\}$ and L_{ap} denotes the all-pass filter length. The complexity of computing each of the $j + 1$ last columns of $\mathbf{Q}$ is $L_{ap}(j + 1)$ for

[2] Using the Householder method, elements of rows in $\mathbf{L}$ converge toward the minimum-phase filter from the bottom and up due to elements in the lower triangular matrix being computed from the bottom and up.

[3] Methods for QR-factorization may easily be converted to QL.

[4] The QL-factorization starts from the bottom row and works its way up to the top, while the QR-factorization uses a top down approach.

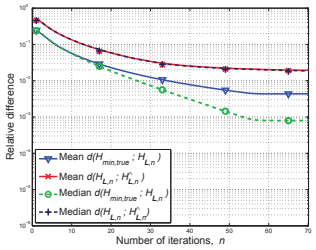


Fig. 1: Gaussian filter coefficients, $L = 7$. Mean and median value of the relative deviations, $d\left(\mathcal{H}_{min,true};\mathcal{H}_{L,n}\right)$ and $d(\mathcal{H}_{L,n};\widehat{\mathcal{H}}_{L,n})$ when $L_{ap} = 32$.

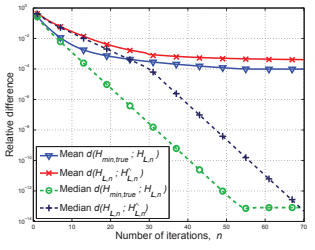


Fig. 2: TU0 profile, $L = 6$. Mean and median value of the relative deviations, $d\left(\mathcal{H}_{min,true};\mathcal{H}_{L,n}\right)$ and $d(\mathcal{H}_{L,n};\widehat{\mathcal{H}}_{L,n})$ when $L_{ap} = 32$.

$j = \{L_{ap} - L, \ldots, L_{ap}\}$. If the number of required iterations is higher than the length of the prefilter, we also need $L_{ap}(L_{ap}+1)$ complex operations to calculate each of the remaining columns (i.e. the columns from $L_{ap}+1$ to n counted from right to left). Thus, the overall complexity of computing the prefilter, is

$$\mathcal{O}_{ap} = \sum_{k=0}^{\min\{(L_{ap}-1);(n-1)\}} \min\{(L+k);L_{ap}\}\cdot(k+1) + \max\{0;(n-L_{ap})\}\cdot L_{ap}(L_{ap}+1) \tag{6}$$

assuming that $n \geq L_{ap} - L + 1$. Note that the last term in (6) vanishes when $n \leq L_{ap}$ and that we will obtain the first L_{ap} filter coefficients after $(L_{ap} - L + 1)$ iterations. Thus, in cases where L is close to L_{ap} we only need a few iterations if we are willing to sacrifice precision in favor of complexity. Thus, for the Hilly Terrain (HT0) profile specified in [13], the minimum-phase filter and the all-pass filters can be obtained using 503 operations (where $L = 10$ and using $L_{ap} = 14$, $n = 5$).

The approximate low complexity method proposed in [2], which uses Linear Prediction (LP) to obtain an estimate of the all-pass and minimum-phase filters, will approximately require $1/2\cdot(L+1)(L+2)+L_p^2+2L_p+(L+1)(L_p+1)$ operations (complex multiplications). Here L_p denotes the order of the prediction-error filter. When $L_p = 14$ this method requires 455 operations for the HT0 profile. Thus, the method proposed here will for some practical channel profiles have comparable complexity to that of the LP-method, but in other cases, the price paid for the better prefilter is a somewhat higher complexity.

5. SIMULATION RESULTS

In this section, we present simulation results for 3 different types of SISO channels. First we assume that we have complex Gaussian distributed, $\mathcal{CN}(0,1)$, channel coefficients. Secondly, we consider two types of channels defined in the GSM specifications [13], namely the Typical Urban (TU0) and the Hilly Terrain (HT0) profiles, see e.g. [5].

We compute the relative difference between the two filters, $\mathcal{H}_a$ and $\mathcal{H}_b$, as a function of the iteration number, n, as

$$d\left(\mathcal{H}_{a,n};\mathcal{H}_{b,n}\right) \triangleq \frac{\|\mathcal{H}_{a,n} - \mathcal{H}_{b,n}\|_2}{\|\mathcal{H}_{a,n}\|_2}, \tag{7}$$

which is done in order to measure the convergence rate of the filter coefficients. In the simulations $\mathcal{H}_{min,true}$ denotes the impulse response of the true minimum-phase filter, and $\mathcal{H}_{L,n}$ is the impulse response obtained from $\mathbf{L}$ (at iteration n). To measure how well the estimated all-pass filter, $\mathcal{H}_{Q,n}$, match the estimated minimum-phase filter $\mathcal{H}_{L,n}$, we filter the original impulse response $\mathcal{H}$ with $\mathcal{H}^o_{Q,n}$, which gives us the output $\widehat{\mathcal{H}}_{L,n}$. In all the simulations presented below, we have made 10000 realizations of the examined channel profile, and computed the minimum-phase and the all-pass filter for each realization. The filter length of the all-pass filter is in all simulations $L_{ap} = 32$. Based on the result of the 10000 filter realizations, we have computed the mean and median value of the relative errors, $d\left(\mathcal{H}_{min,true};\mathcal{H}_{L,n}\right)$ and $d(\mathcal{H}_{L,n};\widehat{\mathcal{H}}_{L,n})$. The result for the Gaussian channel coefficients is shown in Fig. 1, where we see that there is a convergence toward the true minimum-phase filter as a function of the iteration number. In Fig. 2 the result for the TU0 profile is shown, and here we can see that the average relative deviation between the true minimum-phase filter and estimated solution is approximately 10^{-2} after 7-8 iterations. To obtain the same relative deviation between the estimated minimum-phase filter and the estimated all-pass filter we need approximately 14-15 iterations. We can see from the figure that the median value of the relative error converges faster than the mean value, which indicates that some of the realizations will bias the estimate of the mean value due to "outliers" in the distribution of the relative error. By inspecting the approximated PDF for different iterations, it is observed that a few realizations converge slower than the majority, and they will therefore in some sense

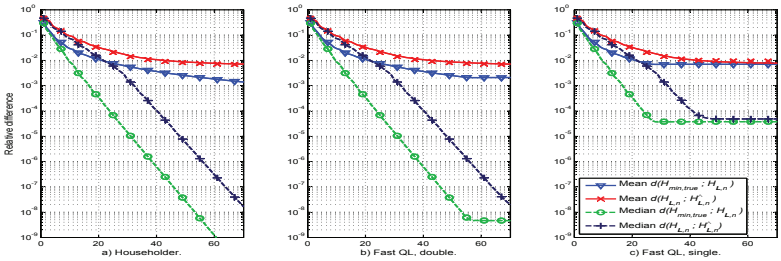


Fig. 3: HT0 profile, $L = 10$. Mean and median value of the relative deviations, $d\left(\mathcal{H}_{min,true}; \mathcal{H}_{L,n}\right)$ and $d(\mathcal{H}_{L,n}; \widehat{\mathcal{H}}_{L,n})$ when $L_{ap} = 32$. Result given for a) the Householder transformation and for b) and c) the fast QL-factorization using floating-point double- and single-precision, respectively.

bias the estimate. The realizations which converge slowest are the ones which contain roots located close to the unit circle. From Fig. 2 it can also be observed that the convergence rate of the median value is exponential. Fig. 3b show the result for the HT0 profile, and in this case the convergence is slower than the TU0 profile. This is not surprising, since the channel impulse response is longer, which makes it more likely that there are roots close to the unit circle. For this profile we need 21 iterations to obtain an average precision of 10^{-2} between the true and estimated minimum-phase filter. In Fig. 1, Fig. 2, and Fig. 3b we see that the relative difference $d(\mathcal{H}_{L,n}; \widehat{\mathcal{H}}_{L,n})$ tends to be biased due to the usage of a finite length all-pass filter. This bias term can be decreased by increasing the length of the all-pass filter, L_{ap}. In the figures we also see that the difference between the true and the estimated minimum phase filter $d(\mathcal{H}_{min,true}; \mathcal{H}_{L,n})$ is biased, which is due to the numerical instability of the rank-1 down-dating procedure, [11]. This effect can be observed by inspecting the median value of the difference between the two filters. To examine the numerical stability of the fast QL-factorization, the Householder transformation has been used as a reference. In Fig. 3a the minimum-phase filter for the HT0 profile has been computed using Householder transformation, and the result is compared to the ones obtained by the fast QL-factorization using either double- or single-precision floating-point operations.

6. CONCLUSION

In this paper we introduced a new approach for computing the minimum-phase filter and its associated all-pass filter in a computationally efficient manner using fast QL-factorization. The proposed method convergences asymptotically toward the true filters with the complexity depending on the required precision.

7. ACKNOWLEDGEMENT

We thank Adam W. Bojanczyk for advice on the extension of fast QR-factorization from real to complex numbers.

8. REFERENCES

[1] J. G. Proakis and D. G Manolakis, *Digital Signal Processing*, Prentice Hall, 3 edition, 1995.

[2] W. H. Gerstacker, F. Obernosterer, Meyer R., and J. B. Huber, "On Prefilter Computation for Reduced-State Equalization," *IEEE Trans. on Wireless Commun.*, vol. 1, pp. 793–800, Oct. 2002.

[3] J. F. Claerbout, *Fundamentals of Geophysical Data Processing*, Blackwell Scientific Publications, 1985.

[4] A. H. Sayed and T. Kailath, "A Survey of Spectral Factorization Methods," *Numerical Linear Algebra with Applications*, vol. 8, pp. 467–496, Jul. 2001.

[5] M. Hansen, L. P. B. Christensen, and O. Winther, "On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation," in *GLOBECOM'07*, 2007, pp. 4237–4241.

[6] M. Hansen, L. P. B. Christensen, and O. Winther, "Computing the Minimum-Phase Filter using QL-Factorization," *Unpublished, under preparation*.

[7] N. Al-Dhahir and J. M. Cioffi, "MMSE Decision-Feedback Equalizers: Finite-Length Results," *IEEE Trans. on Info. Theory*, vol. 41, pp. 961–975, Jul. 1995.

[8] A. W. Bojanczyk, R. P. Brent, and F.R. de Hoog, "QR Factorization of Toeplitz Matrices," *Numer. Math.*, vol. 49, pp. 81–94, Jul. 1986.

[9] D. R. Sweet, "Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 43, pp. 1–21, Feb. 1984.

[10] S. Qiao, "Hybrid Algorithm for Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 53, pp. 351–366, May 1988.

[11] A. W. Bojanczyk, Brent, R. P., Dooren, P. van, and F.R. de Hoog, "A Note on Downdating the Cholesky Factorization," *SIAM J. Sci. Stat. Comput.*, vol. 8, pp. 210–221, May 1987.

[12] D. R. Sweet, "Fast Block Toeplitz Orthogonalization," *Numer. Math.*, vol. 58, pp. 613–629, 1991.

[13] 3GPP TSG GERAN 3GPP TS 45.005, *Radio Transmission and Reception (Release 5)*.

Appendix C

Computing the Minimum-Phase Filter using the QL-Factorization

---

Morten Hansen, Lars P. B. Christensen, and Ole Winther. Computing the Minimum-Phase Filter using the QL-Factorization. *IEEE Transactions on Signal Processing*. Submitted in June 2009, accepted.

Computing the Minimum-Phase Filter using the QL-Factorization

Morten Hansen, Lars P. B. Christensen, and Ole Winther,

***Abstract*—We investigate the QL-factorization of a time-invariant convolutive filtering matrix and show that this factorization not only provides the finite length equivalent to the minimum-phase filter, but also gives the associated all-pass filter. The convergence properties are analyzed and we derive the exact convergence rate and an upper bound for a simple Single-Input Single-Output system with filter length $L = 2$. Finally, this upper bound is used to derive an approximation of the convergence rate for systems of arbitrary length. Implementation-wise, the method has the advantage of being numerically stable and straight forward to extend to the Multiple-Input Multiple-Output case. Furthermore, due to the existence of fast QL-factorization methods, it is possible to compute the filters efficiently.**

***Index Terms*—Minimum-phase filtering, QL-factorization, sphere detection, spectral factorization, wireless communications.**

I. Introduction

THE minimum-phase and the all-pass filters have over the years attracted much attention due to their broad applicability in signal processing. For this reason these types of filters are generally covered in various classical books on signal processing, cf. e.g. [1]–[3]. One area where the minimum-phase filter is widely used is in digital communications over multipath channels where higher-order modulation schemes are employed. In such scenarios the optimal symbol-by-symbol or sequence detector will often require a very high complexity, due to its exponential growth in complexity as a function of the filter length. Furthermore, in multi user detection the complexity grows further, since the number of users will also influence the complexity exponentially. Thus, suboptimal schemes, such as delayed decision feedback, [4], or reduced-state sequence estimation, [5], will often be applied in such systems instead [6]. However, in order to ensure acceptable performance of these schemes, both the minimum-phase filter and the associated all-pass filter are usually needed [4], since the minimum-phase filter provides the highest possible energy concentration in the beginning of the filter impulse response [2].

The scope of this paper is to show a new method of computing both the minimum-phase filter and the associated all-pass filter using the QL-factorization. This insight provides an alternative approach for computing the minimum-phase filter in a numerically stable way, due to the Householder transformation [7], [8]. Furthermore, as shown in [9], fast low-complexity algorithms can be exploited when computing the QL-factorization of the filtering matrix [10]–[13], if one is willing to sacrifice numerical precision in favor of reduced complexity. Thus, the minimum-phase and all-pass filters can be computed efficiently using this method (see [9] for a more detailed description of this).

The rest of the paper is organized as follows; In Section II we present the general system setup, including the system model, while Section III describes some of the existing methods for computing the minimum-phase filter, and Section IV presents an overall treatment of why the QL-factorization provides the minimum-phase filter. Section V contains an elaborate proof of this as well as an analysis of the convergence rate. In Section VI some simulation results are shown, and some concluding remarks are given in Section VII.

Throughout the paper bold lowercase letters (e.g. $\mathbf{x}$) denote column vectors, while bold uppercase letters denote matrices (e.g. $\mathbf{H}$). The matrix transpose is denoted $(\cdot)^T$, while $(\cdot)^H$ is the Hermitian matrix transpose, and the complex conjugate of a complex number is represented by $(\cdot)^*$.

II. System Model

We consider a time-invariant Multiple-Input Multiple-Output (MIMO) system with a Finite Impulse Response (FIR) length L. The output signal $\mathbf{y}_j \in \mathbb{C}^{N_R}$ at time index j can be expressed as

$$\mathbf{y}_j = \sum_{l=0}^{L-1} \mathbf{H}_l \mathbf{x}_{j-l} + \boldsymbol{\upsilon}_j \ , \tag{1}$$

where $\mathbf{x}_j \in \mathbb{C}^{N_T}$ is the input signal at time $j = \{1, 2, \ldots, J\}$, and $\boldsymbol{\upsilon}_j \in \mathbb{C}^{N_R}$ represents the noise term, $\boldsymbol{\upsilon}_j \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$. J is the length of the input sequence and N_R and N_T denote the size of input and output vectors, respectively (in communications also called receive and transmit dimensions). We assume that $N_R \geq N_T$ which implies that the matrix $\mathbf{H}_l \in \mathbb{C}^{N_R \times N_T}$ denoting the lth tap in the impulse response, will be either a "tall-thin" or a square matrix. Using matrix notation, the system model in (1) can be formulated as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\upsilon} \ , \tag{2}$$

where $\mathbf{y} = \left[\ \mathbf{y}_1^T,\ \mathbf{y}_2^T,\ \cdots,\ \mathbf{y}_{J+L-1}^T\ \right]^T$ and $\mathbf{x} = \left[\ \mathbf{x}_1^T,\ \mathbf{x}_2^T,\ \ldots,\ \mathbf{x}_J^T\ \right]^T$. To ease the notation let $M \triangleq N_R(J + L - 1)$ and $N \triangleq JN_T$, leading to $\mathbf{y} \in \mathbb{C}^M$ and $\mathbf{x} \in \mathbb{C}^N$. Due to the time-invariant property of the filter, $\mathbf{H} \in \mathbb{C}^{M \times N}$ will be a block-banded block Toeplitz

M. Hansen and O. Winther are with DTU Informatics, Technical University of Denmark, DK-2800 Lyngby, e-mail: {mha,owi}@imm.dtu.dk.

L. P. B. Christensen is with Modem Algorithm Design, Nokia Denmark, Frederikskaj, DK-1790 Copenhagen V., e-mail: lars.christensen@nokia.com.

Manuscript submitted, June 22, 2009; revised February 22, 2010.

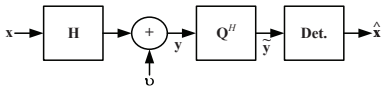


Figure 1. System model with prefilter and detection stage included.

convolution matrix (also referred to as the filtering matrix), having the form

$$\mathbf{H} \triangleq \begin{bmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_{L-1} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{H}_0 \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{L-1} \end{bmatrix} .$$

The finite-length system above can be described in polynomial form when we let the system size $J \to \infty$. The connection between the finite- vs. the infinite-length system is, among others, treated in [14], and the z-transform of the equivalent infinite-length filter impulse response, is given as

$$\mathbf{H}(z) = \sum_{l=0}^{L-1} \mathbf{H}_l z^{-l} , \tag{3}$$

which is a useful representation in the analysis of the filter characteristics.

By QL-factorizing the filtering matrix, $\mathbf{H} = \mathbf{QL}$ in (2) and multiplying by $\mathbf{Q}^H$ with (2), we get a new equivalent system equation

$$\tilde{\mathbf{y}} \triangleq \mathbf{Q}^H \mathbf{y} = \mathbf{L}\mathbf{x} + \tilde{\boldsymbol{\upsilon}} , \tag{4}$$

where we have used the fact that $\mathbf{Q}$ is a unitary matrix. Importantly, it also follows from unitarity that the noise statistic is unchanged (under the assumption of Gaussian noise). Fig. 1 illustrates the system model given in (4).

III. Theory on the Minimum-Phase Filter

As mentioned in the introduction, the minimum-phase filter has been studied intensively over the years due to its broad applicability, and there are various methods for computing the filter. In [15], [16] and the references therein a thorough treatment of several methods for spectral factorization can be found. One classical way of obtaining the minimum-phase filter is by using the root-method of spectral factorization, in which the roots of (3) - which for $N_R = N_T$ satisfy $\det(\mathbf{H}(z)) = 0$ - located outside the unit circle are reflected inside to the conjugate reciprocal location [1], [2], [17]. This simple method however has its limitations, particularly in the case of vector observations (i.e. MIMO systems), since besides the roots we also need to know the direction of the vector associated with that root [3]. Some methods for solving the problem in this case have been described in, among others [18]–[21], but these methods have the disadvantage of being mathematically rather complicated and, furthermore, can suffer from numerical instabilities [3, p. 206]. Thus, one might prefer to solve a Discrete-time Algebraic Riccati Equation (DARE) instead, which is a numerical stable method, having the particularly advantageous property that it can easily be extended to the vector case [15]. In the following we briefly describe how the roots can be determined, which we will be using in the analysis of the convergence rate.

A. The root-method of spectral factorization

Let us for a moment assume that we are only interested in determining the roots of $\mathbf{H}(z)$ in (3). In a MIMO system where $N_T = N_R$, the roots can be obtained by finding the z-values where $\det(\mathbf{H}(z)) = 0$, [22], leading to a matrix polynomial in the scalar variable z. This type of matrix polynomial is normally called a lambda-matrix [23], [24] and the number of roots in such a polynomial is $\min\{N_T, N_R\}(L-1)$. In [23], it is shown that the roots can be obtained by determining the eigenvalues of the block-companion matrix, $\mathbf{C}$, of the associated *monic* polynomial, which can be obtained by $\tilde{\mathbf{H}}(z) \triangleq (\mathbf{H}_{L-1})^{-1}\mathbf{H}(z)$, where we have assumed that $\mathbf{H}_{L-1}$ is invertible. Thus, we get the following block-companion matrix

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & -\tilde{\mathbf{H}}_0 \\ \mathbf{I} & \ddots & \vdots & -\tilde{\mathbf{H}}_1 \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{I} & -\tilde{\mathbf{H}}_{L-2} \end{bmatrix} , \tag{5}$$

where $\tilde{\mathbf{H}}_k \triangleq (\mathbf{H}_{L-1})^{-1}\mathbf{H}_k$. Since the method proposed in [23], [24] assumes that all $\mathbf{H}_l$ terms are square matrices, we cannot directly handle the case where $N_T \neq N_R$ and, therefore, we need to modify the problem. If $N_R > N_T$ we can instead introduce $\mathbf{S} = \mathbf{H}^H\mathbf{H}$ and find the roots of the lambda-matrix based on $\mathbf{S}(z)$,

$$\begin{aligned} \mathbf{S}(z) &= \mathbf{H}^H(z^{-*})\,\mathbf{H}(z) \\ &= \sum_{l=0}^{L-1} \mathbf{S}_l z^{-l} + \sum_{k=1}^{L-1} \mathbf{S}_k^H z^k , \end{aligned} \tag{6}$$

giving the roots both inside and outside the unit circle (from the minimum- and maximum-phase filter, respectively). This does not however solve the problem of finding the zero directions in the lambda-matrix, and we will therefore also address an alternative way of computing the spectral factor.

B. The DARE Method

As mentioned in the previous subsection, the DARE method has the convenient property that it is straight forward to extend from the Single-Input Single-Output (SISO) case to the MIMO case. Furthermore, the method relates to results from Kalman filtering theory and, therefore, many of the properties of this method have been extensively studied, among others its convergence properties, [3].

The DARE method considered in this paper, solves the Riccati equation using the iterative procedure described in [16],[1] and once the stabilizing solution has been obtained, the filter

[1] In [16] the procedure is referred to as *the method of doubling*.

coefficients are computed as described in [25]. The complexity of computing the minimum-phase filter of a length L SISO system using the DARE method is

$$\mathcal{O}_{min,DARE} = k\left(\frac{3}{2}L^2 - \frac{1}{2}L + 2\right) + 2L \qquad (7)$$

operations, where k denotes the number of iterations used for computing the filter.[2]

IV. Connection between the Minimum-Phase Filter and the QL-factorization

As mentioned earlier, it is well-known that the minimum-phase filter can be computed in several ways, and recently it has been realized that the minimum-phase filter and the associated all-pass filter can be obtained by QL-factorizing the filtering matrix $\mathbf{H}$, [26], [27], such that

$$\mathbf{H} = \mathbf{Q}\bar{\mathbf{L}} = \mathbf{Q}\begin{bmatrix} \mathbf{0}_{(M-N)\times N} \\ \mathbf{L} \end{bmatrix}, \qquad (8)$$

where $M \geq N$, and we require that the $N\times N$ lower triangular matrix, $\mathbf{L}$, corresponds to the Cholesky factor of $\mathbf{H}^H\mathbf{H}$, implying that $\mathbf{L}$ is positive definite and thus contains real-valued positive diagonal elements (assuming that $\text{rank}(\mathbf{H}) = N$). In Section V we prove the connection between the minimum-phase filter and the QL-factorization in a more formal way compared to the argument presented in [27], but we would first like to repeat the intuitive argument in order to clarify why we can obtain minimum-phase and all-pass filters using the QL-factorization.

When we QL-factorize the time-invariant block-banded block Toeplitz matrix, each block-row in $\mathbf{L}$ will be a shifted version of each other for $N \to \infty$, where each block-row is given by the spectral factorization, [28]. Likewise, the $M \times M$ unitary matrix, $\mathbf{Q}$, will be the matrix equivalent to the all-pass filter, where again each block-column of $\mathbf{Q}$ will be a shifted version of each other (for $N \to \infty$). Furthermore, it can be seen that each of these block-columns will correspond to the finite dimensional analog of the all-pass filter associated with the minimum-phase filter. In the finite length case, each block-row of $\mathbf{L}$ (block-column of $\mathbf{Q}$) will not be exactly the same, but as we will show later in the paper, the values in each of these will converge toward the true minimum-phase filter as a function of the block-row number.[3] The block-columns of $\mathbf{Q}$ will similarly converge toward the associated all-pass filter.

As shown in [9] the complexity of obtaining the minimum-phase filter of a SISO system using fast QL-factorization is

$$\mathcal{O}_{min,QL} = (k-1)(5L+7) + (L-1)L/2 + 4L \qquad (9)$$

operations and $k+1$ square root operations, while the all-pass filter requires

$$\mathcal{O}_{ap,QL} = \sum_{n=0}^{\min\{(L_{ap}-1);(k-1)\}} \min\{(L+n); L_{ap}\}(n+1) + \max\{0;(k-L_{ap})\}L_{ap}(L_{ap}+1) \qquad (10)$$

operations, assuming that $k \geq L_{ap}$. Here L_{ap} denotes the length of the all-pass filter. From (7) and (9) we see that the fast QL-factorization has a computational advantage over the DARE method described in Section III-B, which can also be seen from Table I in Section VI where a comparison of the complexity of the methods is given.

It should be noted that in the case where we have a time-variant filter, the block-rows in the lower triangular matrix will in some sense represent an analog to the "instantaneous" minimum-phase filter and, likewise, the block-columns of unitary matrix will represent the associated "instantaneous" all-pass filter. It should also be noted that if we perform a QR-factorization of the filtering matrix instead of a QL-factorization, we will get the *maximum-phase* filter.

A. The Householder Transformation

In our analysis of the convergence toward the minimum-phase filter and the all-pass filter we use the Householder transformation to compute the QL-factorization. Therefore, we first briefly describe the steps of this transformation. The reason for choosing this transformation is its advantageous numerical stability to roundoff effects. For a more thorough treatment of the transformation and its numerical properties the reader is referred to [7]. In most textbooks, the Householder algorithm is only described for real numbers and since this transformation plays a crucial role in our treatment of the convergence rate in Section V, we here illustrate a complex version of the transformation. The Householder transformation (for QL-factorization) of a matrix $\mathbf{B} \in \mathbb{C}^{M\times N}$ works as illustrated in Algorithm 1, where $\mathbf{e}_k$ denotes the unit vector with 1 in the kth position, and where we have defined the unitary matrices $\mathbf{U}_k \in \mathbb{C}^{M\times M}$ and $\check{\mathbf{U}}_k \in \mathbb{C}^{(M-k+1)\times(M-k+1)}$ as

$$\mathbf{U}_k \triangleq \begin{bmatrix} \check{\mathbf{U}}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{k-1} \end{bmatrix}.$$

V. Convergence Rate

In this section we examine the convergence properties of the rows and columns the QL-factorization toward the minimum-phase and all-pass filters. In order to simplify this analysis, we first consider the simplest possible case, which is for the SISO case with a filter length of $L = 2$. We will then extend this result to the more general one.

A. SISO system with filter length $L = 2$

Any SISO filtering matrix of a length $L = 2$ system can be formulated as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a & 1 & \ddots & \vdots \\ 0 & a & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & a \end{bmatrix},$$

where we have normalized the impulse response such that $a \triangleq h_1/h_0 \neq 0$, leading to $H(z) = 1 + az^{-1}$ for the equivalent

[2] We here define an operation as a complex Multiply-Accumulate (MAC) instruction.

[3] Strictly speaking the elements in the block-row of $\mathbf{L}$ converge toward the minimum-phase filter from the bottom up, since the Householder transformation computes the elements in the lower triangular matrix from the bottom (when we perform a QL-factorization instead of a QR-factorization).

4

Algorithm 1 Householder Transformation for QL-fact.

1: **Input:** Matrix $\bar{\mathbf{B}}$
2: $\hat{\mathbf{B}} \Leftarrow \mathbf{B}$
3: **for** $k = 1$ to $\min\{M, N\}$ **do**
4: {Pick out the last column vector of $\hat{\mathbf{B}}$}
5: $\hat{\mathbf{b}} = \hat{\mathbf{B}}_{:,end}$
6: $\tilde{k} = M - k + 1$
7: {Do Householder reflection of $\hat{\mathbf{b}}$ (line 8 to 12)}
8: $\alpha = \left\|\hat{\mathbf{b}}\right\|$
9: $\tilde{\alpha} = e^{i\angle \hat{b}_{\tilde{k}}}\, \alpha$
10: $\mathbf{v} = \hat{\mathbf{b}} + \tilde{\alpha}\mathbf{e}_{\tilde{k}}$
11: $\hat{\mathbf{U}}_k = e^{-i\angle \hat{b}_{\tilde{k}}} \left(\frac{2\mathbf{v}\mathbf{v}^H}{\|\mathbf{v}\|^2} - \mathbf{I}\right)$
12: $\tilde{\mathbf{B}} = \hat{\mathbf{U}}_k \hat{\mathbf{B}}$
13: {Remove last row and last column of $\tilde{\mathbf{B}}$}
14: $\acute{\mathbf{B}} \Leftarrow \tilde{\mathbf{B}}_{1:(end-1),1:(end-1)}$
15: {Repeat for new $\acute{\mathbf{B}}$}
16: $\hat{\mathbf{B}} \Leftarrow \acute{\mathbf{B}}$
17: **end for**

After $K = \min\{M, N\}$ iterations we have;
$\mathbf{L} = \mathbf{U}_K \ldots \mathbf{U}_2 \mathbf{U}_1 \mathbf{B}$
$\mathbf{Q} = \mathbf{U}_1^H \mathbf{U}_2^H \ldots \mathbf{U}_K^H$

infinite-length filter impulse response. In this case it is trivial to compute the minimum-phase solution using the root-method

$$z_{mp} = \begin{cases} -a & \text{if } |a| \leq 1 \\ -1/a^* & \text{else} \end{cases}, \tag{11}$$

where z_{mp} represents the minimum-phase root. Since $H(z) = H_{ap}(z)H_{mp}(z)$ we have

$$H_{ap}(z) = \begin{cases} 1 & \text{if } |a| \leq 1 \\ \frac{z^{-1}+\frac{1}{a}}{1+\frac{1}{a^*}z^{-1}} & \text{else} \end{cases}, \tag{12}$$

where $H_{mp}(z)$ and $H_{ap}(z)$ represent the z-transformed minimum-phase and all-pass filters, respectively.[4]
By QL-factorizing the filtering matrix we get

$$\mathbf{L} = \begin{bmatrix} \alpha_N & 0 & \cdots & & 0 \\ \beta_{N-1} & \alpha_{N-1} & \ddots & & \vdots \\ 0 & \beta_{N-2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \alpha_2 & 0 \\ 0 & \cdots & 0 & \beta_1 & \alpha_1 \end{bmatrix}, \tag{13}$$

where we are interested in determining the α and β values. For notational brevity we introduce $\gamma_k \triangleq \hat{b}_{\tilde{k}}$, where $\hat{b}_{\tilde{k}}$ is defined in Algorithm 1 as the last element in vector $\hat{\mathbf{b}}$, which is being reflected in the kth iteration. Since $\hat{\mathbf{b}}_1 = [0, \ldots, 0, 1, a]^T$ we have $\gamma_1 = a$ for the first Householder reflection (also referred to as iteration $k = 1$). Based on the input vector we see from line 8 in Algorithm 1 that $\alpha_1 = \sqrt{1 + |\gamma_1|^2}$ and from lines 9-10 we get $\mathbf{v}_1 = [0, \ldots, 0, 1, \gamma_1 + \tilde{\alpha}_1]^T$. Lines 11 and 12 in Algorithm 1 lead to the following expression for the β,

$$\beta_1 = \frac{2a\, e^{-i\angle\gamma_1} \left(\gamma_1 + e^{i\angle\gamma_1}\alpha_1\right)}{1 + |\gamma_1 + e^{i\angle\gamma_1}\, \alpha_1|^2} \tag{14a}$$

$$= \frac{2a\, (|\gamma_1| + \alpha_1)}{1 + |\, |\gamma_1| + \alpha_1|^2}, \tag{14b}$$

where in (14b) we have used $\gamma_1 = e^{i\angle\gamma_1}|\gamma_1|$. After the first Householder reflection we have

$$\mathbf{U}_1\mathbf{H} = \begin{bmatrix} \ddots & 0 & \cdots & 0 \\ \ddots & 1 & \ddots & \vdots \\ \ddots & a & 1 & 0 \\ \vdots & 0 & \gamma_2 & 0 \\ \cdots & 0 & \beta_1 & \alpha_1 \end{bmatrix},$$

since there are only two non-zero elements in the columns of $\mathbf{H}$. In the next iteration we will have $\hat{\mathbf{b}}_1 = [0, \ldots, 0, 1, \gamma_2]^T$, and by examining the update steps in the Householder reflection carefully, it becomes clear that the value of γ_{k+1} can be expressed as a function of γ_k, leading to a recursive update given as

$$\gamma_{k+1} = a\left(1 - \frac{2}{1 + |\gamma_k + e^{i\angle\gamma_k}\alpha_k|^2}\right) \tag{15a}$$

$$= a\left(1 - \frac{2}{1 + \left|\, |\gamma_k| + \sqrt{1 + |\gamma_k|^2}\right|^2}\right) \tag{15b}$$

$$= \frac{a\,|\gamma_k|}{\sqrt{1 + |\gamma_k|^2}}. \tag{15c}$$

Likewise, the general expression for the α's and β's will be

$$\alpha_k = \sqrt{1 + |\gamma_k|^2} \tag{16}$$

$$\beta_k = \frac{2a\, (|\gamma_k| + \alpha_k)}{1 + |\, |\gamma_k| + \alpha_k|^2}. \tag{17}$$

From (16) we can verify that the α values will be positive and real-valued, which is exactly what is required from the QL-factorization. From (15) and (17) we also see the interesting property that all the values of the γ_k's and the β_k's will always have the same angle in the complex plane, determined by $\angle\beta_k = \angle\gamma_k = \angle a$. This implies that the convergence of the β_k's to the true minimum-phase solution for each iteration takes place in the same direction in the complex plane.

Lemma V.1 (Recursive computation of α_k and β_k)**.** *In a time-invariant SISO system with $L = 2$, the coefficients in $\mathbf{L}$ obtained by the Householder transformation can be determined as*

$$\alpha_k = \sqrt{1 + |\gamma_k|^2}$$
$$\beta_k = \frac{2a\, (|\gamma_k| + \alpha_k)}{1 + |\, |\gamma_k| + \alpha_k|^2}$$

where

$$\gamma_{k+1} = \frac{a\,|\gamma_k|}{\sqrt{1 + |\gamma_k|^2}}.$$

[4] In order to ensure that the magnitude response of the all-pass filter will always be one, we have normalized the minimum-phase filter such that $H_{mp}(z) = a(1 + 1/a^* z^{-1})$ whenever a root is reflected inside the unit circle (i.e. when $|a| > 1$).

Proof: Given above. ■

As shown in Appendix A the recursive expression for γ_k given in (15), can be rewritten as

$$\gamma_k = e^{i\angle a}\sqrt{\frac{|a|^2-1}{1-|a|^{-2k}}}\ . \tag{18}$$

Now in order to show that the values of α_k and β_k match the minimum-phase filter, we need to determine the fixed-point solutions for the parameter γ_k in (15), such that

$$\gamma_{fix} = f\left(\gamma_{fix}\right),\ \text{where}\ f(x) = \frac{a\,|x|}{\sqrt{1+|x|^2}}\ .$$

As shown in the lemma below, there are two fixed-points.

Lemma V.2 (Fixed-points for γ)**.** *In a time-invariant SISO system with $L = 2$, the fixed-point solutions for γ will be*

$$\gamma_{fix} = \begin{cases} 0 & \textit{if } |a| \le 1 \\ e^{i\angle a}\sqrt{|a|^2-1} & \textit{else} \end{cases}\ .$$

Proof: See Appendix B for a detailed proof. ■

Based on these fixed-points for γ we have

$$\alpha_{fix} = \begin{cases} 1 & \text{if } |a| \le 1 \\ |a| & \text{else} \end{cases} \tag{19a}$$

$$\beta_{fix} = \begin{cases} a & \text{if } |a| \le 1 \\ e^{i\angle a} & \text{else} \end{cases}\ . \tag{19b}$$

Thus, the root of $\mathbf{L}$ obtained using the QL-factorization, will be $z_L = -\beta_{fix}/\alpha_{fix}$

$$\begin{aligned} z_L &= \begin{cases} -a & \text{if } |a| \le 1 \\ -e^{i\angle a}/|a| & \text{else} \end{cases} \\ &= \begin{cases} -a & \text{if } |a| \le 1 \\ -1/a^* & \text{else} \end{cases}\ , \end{aligned} \tag{20}$$

which corresponds to the result given in (11), obtained by the traditional root-method of spectral factorization. Likewise, the unitary matrix will converge to the Infinite Impulse Response (IIR) all-pass filter given in (12). In order to ensure that we do in fact get the minimum-phase solution, we also need to prove that the recursive expression for γ_k converges to the fixed-points. In Appendix C it has been proved that this is indeed the case. Thus, it can be concluded that in the SISO case with a filter length of $L = 2$, the elements in the rows of $\mathbf{L}$ converge to the minimum-phase filter.[5]

In the following we examine the convergence rate to the fixed-point solutions, which can be determined based on the expression for γ_k given in (18). In order to compute the convergence rate we introduce

$$\gamma_k = \gamma_{fix} + \Delta\gamma_k\ , \tag{21}$$

where $\Delta\gamma_k$ represents the deviation of γ_k from the fixed-point solution. To upper bound the convergence we treat the cases of $|a| \le 1$ and $|a| > 1$ separately.

[5]This is no surprise, since it has already been shown in [15], [28] that the lower triangular matrix provides the spectral factor.

1) The $|a| \le 1$ case: From (18) we get that

$$|\Delta\gamma_k| = |\gamma_k - \gamma_{fix}| = |a|^k\sqrt{\frac{|a|^2-1}{|a|^{2k}-1}} \tag{22a}$$

$$\le |a|^k\sqrt{\frac{|a|^2-1}{|a|^2-1}} = |a|^k \quad \text{for } \forall\ k \ge 1\ . \tag{22b}$$

2) The $|a| > 1$ case: When $|a| > 1$ the fixed-point is $|\gamma_{fix}| = \sqrt{|a|^2-1}$ and from Lemma C.1 we know that $|\gamma_k| \ge |\gamma_{fix}|$. As mentioned in Appendix C all of the terms which are compared have the same argument and, therefore, we can simply ignore the angle and only consider the case where the terms are real and positive. We then get

$$|\Delta\gamma_k| = |\gamma_k| - |\gamma_{fix}| \tag{23a}$$

$$= \sqrt{|a|^2-1}\left(\frac{1}{\sqrt{1-|a|^{-2k}}} - 1\right) \tag{23b}$$

$$\le \sqrt{|a|^2-1}\left(\frac{1}{1-|a|^{-2k}} - 1\right) \tag{23c}$$

$$\le |a|^{-2k}\ \frac{\sqrt{|a|^2-1}}{1-|a|^{-2}} = |a|^{-2k}\ \frac{|a|^2}{\sqrt{|a|^2-1}}\ . \tag{23d}$$

Thus, we have the following lemma which upper bounds the convergence rate.

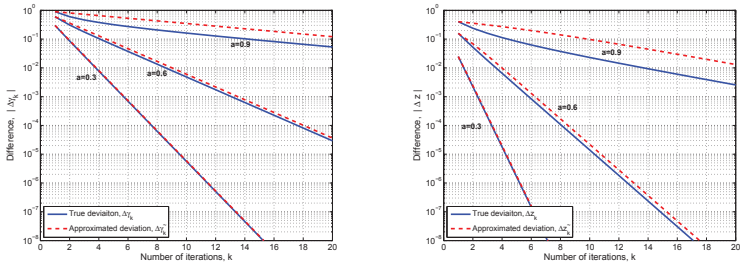
Lemma V.3 (Upper bound on the convergence rate of γ)**.** *In a time-invariant SISO system with $L = 2$, the convergence rate of γ_k can be upper bounded by*

$$|\Delta\gamma_k| \le |\Delta\tilde{\gamma}_k| = \begin{cases} e^{k\,\ln(|a|)} & \textit{if } |a| \le 1 \\ \frac{|a|^2}{\sqrt{|a|^2-1}}\,e^{2k\,\ln(1/|a|)} & \textit{else} \end{cases}\ .$$

From Lemma V.3 we see the interesting property that the convergence rate is exponential and is determined by $|a|$. In other words, the convergence rate to the fixed-point solution is governed by the localization of the root in the complex plane. In the case where we have a root which is close to the unit circle, we will have slow convergence to the minimum-phase solution. In Fig. 2(a) the convergences of $\Delta\gamma_k$ and $\Delta\tilde{\gamma}_k$ have been shown as a function of the number of iterations for a $L = 2$ SISO system in the case where the root is $z = \{-0.3, -0.6, -0.9\}$, respectively. From the figure it is clearly seen that the distance between the root and the unit circle has a significant influence on the convergence rate and, furthermore, we see that the upper bound becomes tighter as the distance grows. It is also relevant to examine how the deviation $\Delta\gamma_k$ affects the value of the root. Therefore, we introduce $z_{L,k} \triangleq -\beta_k/\alpha_k$, which represents the root obtained from $\mathbf{L}$ in the kth iteration. Likewise, we have $\tilde{z}_{L,k} \triangleq -\tilde{\beta}_k/\tilde{\alpha}_k$, where the approximated values of α_k and β_k have been obtained using $\Delta\tilde{\gamma}_k$.

In Fig. 2(b) the deviations from the true minimum-phase root have been plotted, where we have defined $\Delta z_k \triangleq z_{mp} - z_{L,k}$ and $\Delta\tilde{z}_k \triangleq z_{mp} - \tilde{z}_{L,k}$. From the figure, we see that the deviation $\Delta\gamma_k$ is significantly larger than the deviation in the root value Δz_k.

6



(a) True and approximated value of $\Delta\gamma_k$.

(b) True and approximated value of root-deviation Δz_k.

Figure 2. Example of deviations of $\Delta\gamma_k$ and Δz_k in a SISO system, with length $L = 2$ and root at $a = \{0.3, 0.6, 0.9\}$, respectively

B. SISO system with filter length $L > 2$

In the case where we have a filter length of $L > 2$, the deviations of the recursions for the Householder transformation become much more complicated, since the vector $\mathbf{b}$ in Algorithm 1 will now have L non-zero elements. Thus, it will no longer be the simple scalar recursion for γ_k but instead a $(L-1) \times (L-1)$ matrix recursion and, furthermore, due to the multiple roots there will also be multiple fixed points. However, we can generalize the result obtained for the $L = 2$ SISO system by factorizing the filtering matrix into $(L-1)$ products of $L = 2$ filtering matrices,[6] such that

$$\mathbf{H} = \mathbf{H}_2^{(L-1)}\mathbf{H}_2^{(L-2)} \dots \mathbf{H}_2^{(1)} , \tag{24}$$

here $\mathbf{H}_2^{(l)}$ is the filtering matrix of the lth length two filter, where the z-transform of the equivalent infinite-length filter impulse response is given as $H_2^{(l)}(z) \triangleq 1 + a_l\, z^{-1}$. The factorization makes it possible to perform a QL-factorization on each of the $(L-1)$ terms in (24), which gives

$$\mathbf{H}_2^{(l)} = \mathbf{Q}_2^{(l)}\mathbf{L}_2^{(l)} . \tag{25}$$

where the convergence rate of each of the $(L-1)$ terms is given in Subsection V-A. By inserting (25) into (24) we get

$$\mathbf{H} = \mathbf{QL} = \mathbf{Q}_2^{(L-1)}\mathbf{L}_2^{(L-1)}\mathbf{Q}_2^{(L-2)}\mathbf{L}_2^{(L-2)} \dots \mathbf{Q}_2^{(1)}\mathbf{L}_2^{(1)} . \tag{26}$$

We would like to reorder the terms on the RHS of (26) such that all $\mathbf{Q}_2^{(l)}$ terms are grouped together followed by all the $\mathbf{L}_2^{(l)}$ terms, i.e.

$$\mathbf{H} \cong \underbrace{\mathbf{Q}_2^{(L-1)}\mathbf{Q}_2^{(L-2)} \dots \mathbf{Q}_2^{(1)}}_{\mathbf{Q}} \underbrace{\mathbf{L}_2^{(L-1)}\mathbf{L}_2^{(L-2)} \dots \mathbf{L}_2^{(1)}}_{\mathbf{L}} , \tag{27}$$

where the equality holds when the system size $N \to \infty$. The reason that it is possible to rearrange the terms when the system size goes to infinity is due to fact that $\mathbf{L}_2^{(l)}$ and $\mathbf{Q}_2^{(l)}$ asymptotically become circulant matrices [29], and thereby, we can use the commutative property of circulant matrices [29]. Conceptually it is fairly easy to see why $\mathbf{L}_2^{(l)}$ asymptotically becomes circulant, since it is a banded matrix, but this might not be as obvious for the all-pass filtering matrix, which represents an IIR filter. However, it has been proved in [30] that the IIR filter has an exponential decay, which implies that, in the limit where the system size tends to infinity, the IIR filter becomes a Toeplitz matrix. In [29] it is proved that general Toeplitz matrices containing absolutely summable elements (also referred to as *Wiener Class Toeplitz Matrices*) asymptotically converge to circulant matrices too. Thus, in the limit $N \to \infty$ both matrices become circulant and, therefore, we know that the lower triangular matrix $\mathbf{L}$ converge to the minimum-phase filter for SISO systems of arbitrary length. Due to the unique factorization of $\mathbf{H} = \mathbf{QL}$ (where we require that the elements on the diagonal of $\mathbf{L}$ are real-valued and positive), $\mathbf{Q}$ *must* be the matrix version of the all-pass filter associated with the minimum-phase filter, since it is the only unitary matrix which links $\mathbf{L}$ with $\mathbf{H}$.

Based on the expression in (27) it is possible to approximate the convergence rate in a SISO system of arbitrary length, by examining the deviations in the approximated root values $\Delta\tilde{z}_k^{(l)} \triangleq z_{mp}^{(l)} - \tilde{z}_{L,k}^{(l)}$, where $z_{mp}^{(l)}$ represents the lth root of the true minimum-phase filter and $\tilde{z}_{L,k}^{(l)} \triangleq -\tilde{\beta}_k^{(l)}/\tilde{\alpha}_k^{(l)}$ is the approximated value of the lth root based on the upper bound given in Lemma V.3. Thus, in the z-domain the difference between the true minimum-phase filter and the filter obtained based on $\tilde{z}_{L,k}^{(l)}$ becomes

$$\Delta H(z) \triangleq H_{mp}(z) - \tilde{L}_k(z) \tag{28a}$$

$$\approx z^{-(L-1)} \left[\prod_{l=1}^{L-1} \left(z - z_{mp}^{(l)} \right) - \prod_{l=1}^{L-1} \left(z - \tilde{z}_{L,k}^{(l)} \right) \right] , \tag{28b}$$

where $\tilde{L}_k(z)$ represents the z-transform of the approximate value for the kth row in the lower triangular matrix, $\mathbf{L}$. In (28) it is only the effect of each deviation in the approximated root value that has been taken into account, but the approximation

[6] It should be noted that the size of $\mathbf{H}_2^{(l)}$ decreases by one (both column- and row-wise) as l decreases by one, in order to enable the factorization.

made in (27) will to some extent affect the convergence rate due to roots only being asymptotically independent. The tightness of the approximate expression for convergence when $L > 2$ has been evaluated empirically in Figs. 4 and 5 given in Section VI. In (28b) we have normalized the first coefficient and from the equation we can see that the main contribution to the difference between the true minimum-phase filter and the result obtained by the QL-factorization, will asymptotically come from the root which is closest to the unit circle. This observation fits well with what is described in [3, p. 508], where the convergence to the stabilizing solution of the DARE is exponential and determined by the spectral radius.

C. MIMO system

In the case where we have a MIMO system, we can first examine the length 2 system $\mathbf{H}(z) = \mathbf{I} + \mathbf{H}_1 z^{-1}$ where $N_T = N_R$. Compared to the SISO system of the same length, the only difference is that the operations now become $N_R \times N_T$ matrix operations instead of scalars. However, since we have $\eta = \min\{N_T, N_R\}(L-1)$ roots, we get 2^η fixed-points, thus it becomes more complicated to analyze even a simple $L = 2$ MIMO system. In the case where we have an arbitrary filter length, the argument presented in Subsection V-B, concerning the SISO system of length $L > 2$, can be repeated here.

VI. Simulation Results

In this section simulation results for both SISO and MIMO systems are presented. For the SISO system we examine two channel scenarios. In the first scenario we have complex Gaussian distributed, $\mathcal{CN}(0,1)$, filter coefficients. In the second scenario we consider a channel defined in the GSM specifications [31], namely the Typical Urban (TU0) profile. The channel profile, obtained by the convolution of the square root of the power delay profile with the transmit filter response (the so-called C_0-pulse in [32]), is plotted in Fig. 3, and from the figure it is seen that roughly speaking we need about four symbols to capture the energy of the pulse. If the cardinality of the alphabet is denoted by $|\Omega|$, this implies that a Delayed Decision Feedback Equalizer Sequence Estimation (DDFSE) type of equalizer would require about $|\Omega|^{(4-1)}$ states in the state-space model in order to achieve a performance close to the optimal Maximum Likelihood Sequence Detector (MLSD). The DDFSE equalizer will then have a complexity similar to the MLSD, if no prefiltering is made. In e.g. EGPRS2 [33], this would lead to an unacceptably high decoding complexity, since we can have cardinalities up to $|\Omega| = 32$, giving approximately $3.3 \cdot 10^4$ states. Thus, in such applications it is a great advantage to prefilter the pulse with the all-pass filter to obtain the minimum-phase filter. In [27] we have shown a practical application of this, where we have evaluated the effect of prefiltering in terms of performance of equalizers employing reduced-state sequence estimation techniques.

In order to measure the convergence rate of the filter coefficients, we compute the relative difference between two overall filtering impulse response matrices, $\mathcal{H}_{a,k}$ and $\mathcal{H}_{b,k}$, at the iteration number k, as

$$d\left(\mathcal{H}_{a,k}; \mathcal{H}_{b,k}\right) \triangleq \frac{\|\mathcal{H}_{a,k} - \mathcal{H}_{b,k}\|_2}{\|\mathcal{H}_{a,k}\|_2} \,. \tag{29}$$

Table I
Complexity of computing the minimum-phase filter using the fast QL-factorization (QL) and the DARE method (DARE) using k iterations in a length L SISO system.

k	Method	$L=5$	$L=10$	$L=15$	$L=20$
10	QL	$3.18 \cdot 10^2$	$5.98 \cdot 10^2$	$9.03 \cdot 10^2$	$1.23 \cdot 10^3$
	DARE	$3.70 \cdot 10^2$	$1.48 \cdot 10^3$	$3.34 \cdot 10^3$	$5.95 \cdot 10^3$
20	QL	$6.38 \cdot 10^2$	$1.17 \cdot 10^3$	$1.72 \cdot 10^3$	$2.30 \cdot 10^3$
	DARE	$7.30 \cdot 10^2$	$2.94 \cdot 10^3$	$6.65 \cdot 10^3$	$1.19 \cdot 10^4$

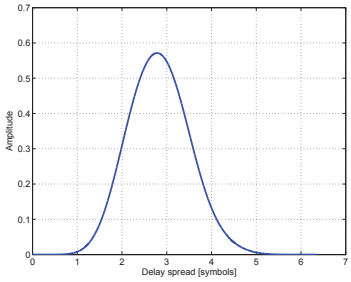
We define $\mathcal{H}_{mp}$ as the impulse response of the true minimum-phase filter, and $\mathcal{H}_{L,k}$ represents the impulse response obtained from $\mathbf{L}$ (at iteration k). To measure how well the estimated all-pass filter, $\mathcal{H}_{Q,k}$, matches the estimated minimum-phase filter $\mathcal{H}_{\tilde{L},k}$, we filter the original impulse response $\mathcal{H}$ with $(\mathcal{H}_{Q,k})^H$, which gives us the output $\mathcal{H}_{\tilde{L},k}$.

In all the simulations presented below, we have made 10,000 realizations of the examined channel profile, and computed the minimum-phase and the all-pass filter for each realization. The filter length of the all-pass filter is set to $L_{ap} = 64$ in the simulations. Based on the results obtained from the 10,000 filter realizations, we have computed the mean and median value of the relative errors, $d\left(\mathcal{H}_{mp}; \mathcal{H}_{L,k}\right)$ and $d(\mathcal{H}_{L,k}; \mathcal{H}_{\tilde{L},k})$. The results for the Gaussian filter coefficients with uniform power in the delay domain are shown in Fig. 4, where we see that the rows in $\mathbf{L}$ converge to the true minimum-phase filter as a function of the iteration number (i.e. the row number).[7] From the figure we observe that the median value of $d\left(\mathcal{H}_{mp}; \mathcal{H}_{L,k}\right)$ converges exponentially to zero and that the median difference is about 10^{-8} after 140 iterations. The convergence of the average difference is considerably slower, due to the instances where a channel realization has zeros very close to the unit circle, which will lead to a slow convergence. Thus, these cases tend to bias the estimate of average convergence rate. This is indeed what can be observed from the estimated probability density function (pdf) of $d\left(\mathcal{H}_{mp}; \mathcal{H}_{L,k}\right)$. Likewise, the mean of $d(\mathcal{H}_{L,k}; \mathcal{H}_{\tilde{L},k})$ seems to be biased, which (besides the effect described above) is also due to the truncation of the IIR all-pass filter. Both the mean and median value of the approximated convergence $d(\mathcal{H}_{mp}; \mathcal{H}_{\tilde{L},k})$ have also been plotted, and from (29) it is seen that this term represents the energy of the approximated deviation obtained in (28) normalized with respect to the energy of the minimum-phase filter. From the figure it can be seen that the trend of the true and approximated deviation behaves similarly. As a reference we have also included the relative deviation between the true minimum-phase filter and the one obtained using the DARE method, and from this it is possible to see that convergence of the two iterative methods is almost identical. In Table I the complexity of computing the minimum-phase filter using the two iterative methods has been compared (based on (7) and (9)), and from this it is seen that the fast QL-factorization method has a computational advantage.

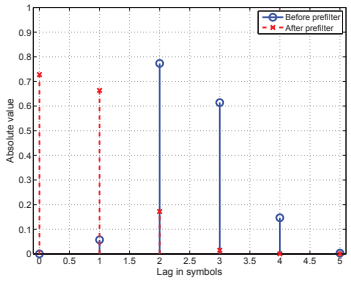
In Fig. 5 the result for the TU0 profile is shown and it is seen that the convergence rate is faster for this channel type compared with the Gaussian filter coefficients with uniform power in the delay domain. It is again observed that the

[7] Again, strictly speaking the convergence occurs from the last row and up, since it is the QL-factorization.

8



(a) The ensemble average of the pulse shape of the TU profile (including the transmit pulse shaping).



(b) The absolute value of filter coefficients for one realization of the TU profile with and without minimum-phase prefiltering.

Figure 3. Delay profile and single realization thereof for the Typical Urban (TU) channel.

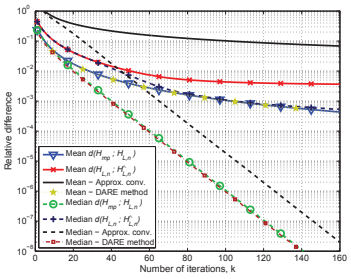


Figure 4. The relative deviations $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$ and $d(\mathcal{H}_{L,k};\mathcal{H}_{L,k})$ in a SISO channel with Gaussian coefficients having uniform power in the delay domain, $L = 6$.

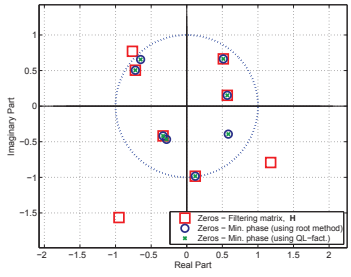


Figure 6. Locations of roots in a 2×2 MIMO system with Gaussian filter coefficients, $L = 5$. The number of iterations in the QL-factorization is $k = 200$.

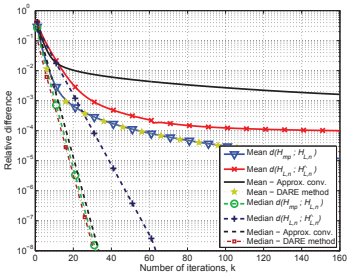


Figure 5. The relative deviations $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$ and $d(\mathcal{H}_{L,k};\mathcal{H}_{L,k})$ in the SISO channel TU0 with $L = 5$.

median value of the difference decreases more rapidly than the mean value. We also see that the approximated convergence of $d\left(\mathcal{H}_{mp};\mathcal{H}_{L,k}\right)$ is even closer to the actual convergence for this channel profile and that the DARE method again has similar convergence.

In Fig. 6 we have a plot of the location of the roots of a 2×2 MIMO system having Gaussian coefficients with filter length $L = 5$, leading to 8 roots. From the plot it is seen that the roots of $\mathbf{H}(z)$ (illustrated with squares) which lie outside the unit circle are reflected inside (the circles) using the root method. Furthermore, it is seen that these roots match the roots of $\mathbf{L}(z)$. In Fig. 7 the root difference $\Delta z_k^{(l)} = z_{mp}^{(l)} - z_{L,k}^{(l)}$ has been plotted for each of the roots $l = \{1, \ldots, 8\}$ for iteration $k = 20$ and $k = 200$. The roots have been sorted according to their distance to the unit circle, such that the one closest to the unit circle is called root 1, etc. The figure shows that the closer the root is to the unit circle, the slower the convergence it will have, which follows the convergence analysis given in Section

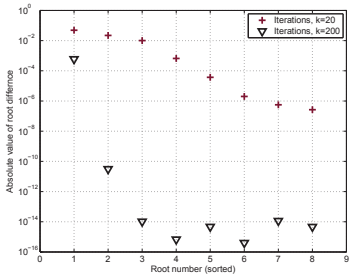


Figure 7. Deviation of the roots in a 2×2 MIMO system with Gaussian filter coefficients, $L = 5$ for iteration $k = 20$ and $k = 200$.

V. After $k = 200$ iterations, it is primarily the root closest to the unit circle which contributes to the difference between the filter obtained from $\mathbf{L}$ and the true minimum-phase filter.

VII. Conclusion

It has been shown how the QL-factorization of the filtering matrix gives the finite length equivalent to the minimum-phase and the all-pass filters and thereby presents a new way of computing these two classical filters in a numerically stable way. The exact convergence rate has been computed for a simple SISO length $L = 2$ system and an upper bound has been derived, which is used for approximating the convergence in systems of arbitrary length. Asymptotically these results also generalize to MIMO systems. The derived convergence results correspond well with what is observed in simulations and, furthermore, they are in agreement with existing results for the DARE method.

Acknowledgment

The authors would like to thank Prof. Babak Hassibi, for the useful discussions on spectral factorization and the Discrete-time Algebraic Riccati Equation. We also thank the reviewers for useful comments and suggestions.

Appendix A
Functional Form of Recursive Parameter γ_k

In this appendix we rewrite the recursive expression for the parameter γ_k to a functional form. The recursion given in (15) is

$$\gamma_{k+1} = \frac{a\,|\gamma_k|}{\sqrt{1+|\gamma_k|^2}}\ ,$$

and by inserting $\gamma_1 = a$ into the expression above we get $\gamma_2 = e^{i\angle a}\ |a|^2 / \sqrt{1+|a|^2}$ and we then see that

$$\gamma_3 = \frac{e^{i\angle a}\ |a|\ \frac{|a|^2}{\sqrt{1+|a|^2}}}{\sqrt{1+\frac{|a|^4}{1+|a|^2}}} = \frac{e^{i\angle a}\ |a|^3}{\sqrt{1+|a|^2+|a|^4}}\ . \tag{30}$$

The reduction made in (30) can be repeated in every step of the recursion and, therefore, in general we get

$$\gamma_k = \frac{e^{i\angle a}\ |a|^k}{\sqrt{\sum_{k'=0}^{k-1}|a|^{2k'}}} = \frac{e^{i\angle a}\ |a|^k}{\sqrt{\frac{1-|a|^{2(k-1)+2}}{1-|a|^2}}} \tag{31a}$$

$$= e^{i\angle a}\ |a|^k \sqrt{\frac{|a|^2-1}{|a|^{2k}-1}}\ . \tag{31b}$$

Appendix B
Fixed-Point Solutions

This appendix proves Lemma V.2. From Section V-A we have seen that the parameter γ_k, which determines the coefficients in $\mathbf{L}$ of the Householder transformation, can be computed recursively. In order to find the fixed-point solution

$$\gamma_{fix} = f(\gamma_{fix})\ , \quad \text{where } f(x) = \frac{a\,|x|}{\sqrt{1+|x|^2}}\ , \tag{32}$$

we will first assume that $\gamma_{fix} \neq 0$. We then have

$$\gamma_{fix} = \frac{a\,|\gamma_{fix}|}{\sqrt{1+|\gamma_{fix}|^2}} \quad \Leftrightarrow \tag{33a}$$

$$a = \sqrt{1+|\gamma_{fix}|^2}\, e^{i\angle\gamma_{fix}} \quad \Leftrightarrow \tag{33b}$$

$$|a|^2 = 1+|\gamma_{fix}|^2 \quad \Rightarrow \tag{33c}$$

$$\gamma_{fix} = e^{i\angle a}\sqrt{|a|^2-1}\ , \quad \text{where } |a|>1\ . \tag{33d}$$

In (33b) and (33d) we have used the fact that $\angle\gamma_{fix}$ will always be the same as $\angle a$. By inserting (33d) in (32) it is easily verified that this is indeed a fixed-point solution for $|a| > 1$.

Let us next consider the case where $\gamma_{fix} = 0$. By inserting this value of γ_{fix} into (32), it is easily seen that this is actually a fixed-point. Thus, Lemma V.2 has hereby been proved.

Appendix C
Convergence to Fixed-Point Solutions

In order to prove that the recursive update of γ_k converges to the fixed-points we are interested in showing that

$$|\gamma_{k+1}-\gamma_{fix}| < |\gamma_k-\gamma_{fix}|\ , \quad \forall\ k \geq 1\ . \tag{34}$$

The proof has been split up into the case for $|a| \leq 1$ and $|a| > 1$, but we will first prove Lemma C.1, which turns out to be a useful lemma when proving the convergence to the fixed-point for $|a| > 1$.

Lemma C.1. *Let γ_{fix} be the fixed-point solution given in Lemma V.2 and let ϵ be a complex valued constant with $\angle(\epsilon) = \angle\gamma_{fix} = \angle a$, we then have*

$$|f(\epsilon+\gamma_{fix})| > |\gamma_{fix}|\ , \quad \textit{where } f(x) = \frac{a\,|x|}{\sqrt{1+|x|^2}}\ . \tag{35}$$

Proof: It is trivial to show that Lemma C.1 is valid for $|a| \leq 1$, since $\gamma_{fix} = 0$, so we will focus on $\gamma_{fix} = e^{i\angle a}\sqrt{|a|^2-1}$. Since the arguments for all the terms in (35)

are the same, we can simply ignore the angle and only consider the simple case where all values are real and positive, i.e.

$$f(\epsilon + \gamma_{fix}) > \gamma_{fix} \qquad \Rightarrow \qquad (36a)$$

$$\frac{a\left(\epsilon + \sqrt{a^2-1}\right)}{\sqrt{1+\left(\epsilon+\sqrt{a^2-1}\right)^2}} > \sqrt{a^2-1}\ , \qquad (36b)$$

which can be rewritten as

$$\frac{1}{(\epsilon^2 + 2\epsilon\sqrt{a^2-1})/a^2 + 1} < 1\ . \qquad (37)$$

Since all the terms on the LHS in (37) are positive, the inequality is true for all $|a| > 1$. This completes the proof. ■

We now turn our attention to proving (34).

A. Case where $|a| \leq 1$

It is fairly straight forward to prove the convergence in this case, since $\gamma_{fix} = 0$ and, thus, (34) reduces to showing that $\lim_{k\to\infty} \gamma_k = 0$, or equivalently $|\gamma_k| > |\gamma_{k+1}|$,

$$|\gamma_k| > |\gamma_{k+1}| = \frac{|a|\,|\gamma_k|}{\sqrt{1+|\gamma_k|^2}} \qquad \Leftrightarrow \qquad (38a)$$

$$\sqrt{1+|\gamma_k|^2} > |a|\ . \qquad (38b)$$

Since we assumed that $|a| \leq 1$, (38b) is satisfied for all $\gamma_{fix} \neq 0$. In the case where $\gamma = 0$ we have the fixed-point, and therefore, we have proved the convergence to the fixed-point for $|a| \leq 1$.

B. Case where $|a| > 1$

Since the initial input to the recursion in (15) is a, which numerically is greater than γ_{fix}, we can use Lemma C.1 to rewrite (34) as

$$|\gamma_{k+1}| - |\gamma_{fix}| < |\gamma_k| - |\gamma_{fix}| \qquad \Leftrightarrow \qquad (39a)$$

$$|\gamma_{k+1}| = \frac{|a|\,|\gamma_k|}{\sqrt{1+|\gamma_k|^2}} < |\gamma_k| \qquad \Leftrightarrow \qquad (39b)$$

$$\frac{|a|}{\sqrt{1+|\gamma_k|^2}} < 1\ . \qquad (39c)$$

By recalling from Lemma C.1 that $|\gamma_{fix}| < |\gamma_k|$ (when $\gamma_k \neq \gamma_{fix}$), we can upper bound LHS in (39c) as

$$\frac{|a|}{\sqrt{1+|\gamma_k|^2}} < \frac{|a|}{\sqrt{1+|\gamma_{fix}|^2}} \qquad (40a)$$

$$= \frac{|a|}{\sqrt{1+|a|^2-1}} = 1\ . \qquad (40b)$$

Now since the RHS in (40a) is equal to one, (39) must be true, which completes the proof of convergence for $|a| > 1$, and provides us with the following lemma.

Lemma C.2 (Convergence to fixed-point)**.** *Let γ_{fix} be the fixed-point solution given in Lemma V.2 and let $\gamma_k \neq 0$ be a complex valued number given by the recursion in* (15)*. For $\gamma_1 = a$ the value of γ_k satisfy*

$$|\gamma_{k+1} - \gamma_{fix}| < |\gamma_k - \gamma_{fix}|\ , \quad \forall\ k \geq 1\ .$$

References

[1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice Hall, 1995.

[2] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[3] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[4] A. Duel-Hallen, , and C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. on Commun.*, vol. 37, pp. 428–436, May 1989.

[5] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback," *IEEE Trans. on Commun.*, vol. 36, pp. 13–20, Jan. 1988.

[6] W. H. Gerstacker, F. Obernosterer, M. R., and J. B. Huber, "On Prefilter Computation for Reduced-State Equalization," *IEEE Trans. on Wireless Commun.*, vol. 1, pp. 793–800, Oct. 2002.

[7] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

[8] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[9] M. Hansen and L. P. B. Christensen, "Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization," in *ICASSP'09*, 2009.

[10] A. W. Bojanczyk, R. P. Brent, and F. d. Hoog, "QR Factorization of Toeplitz Matrices," *Numer. Math.*, vol. 49, pp. 81–94, Jul. 1986.

[11] D. R. Sweet, "Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 43, pp. 1–21, Feb. 1984.

[12] S. Qiao, "Hybrid Algorithm for Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 53, pp. 351–366, May 1988.

[13] D. R. Sweet, "Fast Block Toeplitz Orthogonalization," *Numer. Math.*, vol. 58, pp. 613–629, 1991.

[14] N. Al-Dhahir and J. Cioffi, "Finite-length vs. infinite-length MMSE-DFE: The Connection," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, 1993, pp. 677–681.

[15] A. H. Sayed and T. Kailath, "A Survey of Spectral Factorization Methods," *Numerical Linear Algebra with Applications*, vol. 8, pp. 467–496, Jul. 2001.

[16] V. Kucera, "Factorization of Rational Spectral Matrices: A Survey of Methods," in *International Conference on Control 1991.*, 1991, pp. 1074–1078.

[17] J. F. Claerbout, *Fundamentals of Geophysical Data Processing*. Blackwell Scientific Publications, 1985.

[18] D. Youla, "On the Factorization of Rational Matrices," *Information Theory, IRE Transactions on*, vol. 7, no. 3, pp. 172–189, 1961.

[19] D. Youla and N. Kazanjian, "Bauer-Type Factorization of Positive Matrices and the Theory of Matrix Polynomials Orthogonal on the Unit Circle," *IEEE Trans. on Circuits and Systems*, vol. 25, no. 2, pp. 57–69, 1978.

[20] Y. Rozanov, "Spectral Properties of Multivariate Stationary Processes and Boundary Properties of Analytic Matrices," *Theory of Probability and its Applications*, vol. 5, p. 362, 1960.

[21] A. Yaglom, "Effective Solutions of Linear Approximation Problems for Multivariate Stationary Processes with a Rational Spectrum," *Theory of Probability and its Applications*, vol. 5, pp. 239–264, 1960.

[22] R. F. H. Fischer, "Sorted Spectral Factorization of Matrix Polynomials in MIMO Communications," *IEEE Trans. on Info. Theory*, vol. 53, pp. 945–951, Jun. 2005.

[23] J. E. Dennis Jr., J. F. Traub, and R. P. Weber, "The Algebraic Theory of Matrix Polynomials," *SIAM Journal on Numerical Analysis*, vol. 13, pp. 813–845, Dec. 1976.

[24] ——, "Algorithms for Solvents of Matrix Polynomials," *SIAM Journal on Numerical Analysis*, pp. 523–533, 1978.

[25] B. Anderson, K. Hitz, and N. Diem, "Recursive Algorithm for Spectral Factorization," *IEEE Transactions on Circuits and Systems*, vol. 21, no. 6, pp. 742–750, 1974.

[26] L. P. B. Christensen, "Signal Processing for Improved Wireless Receiver Performance," Ph.D. dissertation, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2007.

[27] M. Hansen, L. P. B. Christensen, and O. Winther, "On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation," in *GLOBECOM'07*, 2007, pp. 4237–4241.

[28] N. Al-Dhahir and J. M. Cioffi, "MMSE Decision-Feedback Equalizers: Finite-Length Results," *IEEE Trans. on Info. Theory*, vol. 41, pp. 961–975, Jul. 1995.

[29] R. Gray, *Toeplitz And Circulant Matrices: A Review (Foundations and Trends in Communications and Information Theory)*. Now Publishers Inc. Hanover, MA, USA, 2006.
[30] T. I. Laakso and V. Välimäki, "Energy-Based Effective Length of the Impulse Response of a Recursive Filter," in *ICASSP'98*, vol. 3, 1998, pp. pp. 1253–1256.
[31] 3GPP TS 45.005, *3GPP TSG GERAN*. Radio Transmission and Reception (Release 5).
[32] P. Laurent, "Exact and Approximate Construction of Digital Phase Modulations by Superposition of Amplitude Modulated Pulses (AMP)," *IEEE Trans. on Commun.*, vol. 34, no. 2, pp. 150–160, 1986.
[33] 3GPP TS 45.004, *3GPP TS GERAN*. Radio Access Network; Modulation (Release 8), 2008.

Morten Hansen (S'07, M'09) was born in Copenhagen, Denmark, in 1979. He received his B.E. and M.Sc. degrees in electrical engineering from the Technical University of Denmark (DTU), in 2004 and 2006, respectively. He is currently pursuing his Ph.D. degree in signal processing for wireless communications at DTU. During his Ph.D. studies he has been a graduate student visitor at California Institute of Technology (CalTech) in Prof. Babak Hassibi's group.

Lars P. B. Christensen (S'05, M'08) received his M.Sc. and Ph.D. degrees in 2003 and 2007 respectively, both in electrical engineering from the Technical University of Denmark. Since graduating he has been with the Modem Algorithm Design group at Nokia working on research and design of signal processing algorithms for wireless communications. His interests are within estimation, detection and decoding related to communication systems as well as the general areas of signal processing and information theory.

Ole Winther (OW) works in machine learning with applications in bioinformatics, data mining, 3G+ wireless communication and collaborative filtering. Ole Winther (M.Sc. physics and computer science '94 and Ph.D. physics '98, both at University of Copenhagen (KU)) has published 50+ scientific papers and has previously held positions at Lund University and Center for Biological Sequence Analysis, Technical University of Denmark (DTU). He currently holds joint positions as associate professor at Intelligent Signal Processing (ISP), IMM, DTU and group leader at Bioinformatics, KU.

Appendix D

Near-Optimal Detection in MIMO Systems using Gibbs Sampling

Morten Hansen, Babak Hassibi, Alexandros G. Dimakis, and Weiyu Xu. Near-Optimal Detection in MIMO Systems using Gibbs Sampling *IEEE Global Telecommunications Conference (GLOBECOM)*. November 2009.

Near-Optimal Detection in MIMO Systems using Gibbs Sampling

Morten Hansen*, Babak Hassibi†, Alexandros G. Dimakis†, and Weiyu Xu†

*Technical University of Denmark, Informatics and Mathematical Modelling,
Build. 321, DK-2800 Lyngby, Denmark,
E-mail: mha@imm.dtu.dk

†California Institute of Technology, Department of Electrical Engineering,
Pasadena, CA 91125, USA
Email: hassibi@systems.caltech.edu, adim@eecs.berkeley.edu, and weiyu@caltech.edu

***Abstract*—In this paper we study a Markov Chain Monte Carlo (MCMC) Gibbs sampler for solving the integer least-squares problem. In digital communication the problem is equivalent to performing Maximum Likelihood (ML) detection in Multiple-Input Multiple-Output (MIMO) systems. While the use of MCMC methods for such problems has already been proposed, our method is novel in that we optimize the "temperature" parameter so that in steady state, i.e. after the Markov chain has mixed, there is only polynomially (rather than exponentially) small probability of encountering the optimal solution. More precisely, we obtain the largest value of the temperature parameter for this to occur, since the higher the temperature, the faster the mixing. This is in contrast to simulated annealing techniques where, rather than being held fixed, the temperature parameter is tended to zero. Simulations suggest that the resulting Gibbs sampler provides a computationally efficient way of achieving approximative ML detection in MIMO systems having a huge number of transmit and receive dimensions. In fact, they further suggest that the Markov chain is rapidly mixing. Thus, it has been observed that even in cases were ML detection using, e.g. sphere decoding becomes infeasible, the Gibbs sampler can still offer a near-optimal solution using much less computations.**

I. Introduction

The problem of performing Maximum Likelihood (ML) decoding in digital communication has gained much attention over the years. One method to obtain the ML solution is Sphere Decoding (SD) [1]–[5]. Over a wide range of Signal-to-Noise Ratios (SNR)s the average complexity of SD is significantly smaller than exhaustive search detectors, but in worst case the complexity is still exponential [6]. Thus, in scenarios with poor SNR or in Multiple-Input Multiple-Output (MIMO) systems with huge transmit and receive dimensions, even SD can be infeasible. A way to overcome this problem is to use approximate Markov Chain Monte Carlo (MCMC) detectors instead, which asymptotically can provide the optimal solution, [7], [8]. Gibbs sampling (also known as Glauber dynamics) is one MCMC method, which is used for sampling from distributions of multiple dimensions. The Gibbs sampler has among others been proposed for detection purposes in wireless communication in [9]–[12] (see also the references therein). The scope of this paper is to describe and analyse a new way of solving the integer least-squares problem using MCMC. It will be shown that the method can be used for achieving a near-optimal and computationally efficient solution of the problem, even for systems having a huge dimension.

The paper is organized as follows; In Section II we present the system model that will be used throughout the paper. The MCMC method is described in Section III and in Section IV we analyse the probability of error for the ML detector. Section V treats the optimal selection of the temperature parameter α, while the simulation results are given in Section VI and some concluding remarks are found in Section VII.

II. System Model

We consider a real-valued block-fading MIMO antenna system, with N transmit and N receive dimensions, with know channel coefficients.[1] The received signal $\mathbf{y} \in \mathbb{R}^N$ can be expressed as

$$\mathbf{y} = \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{s} + \boldsymbol{\upsilon}\ , \tag{1}$$

where $\mathbf{s} \in \Omega^N$ is the transmitted signal, and Ω denotes the constellation set. To simplify the derivations in the paper we will assume that $\Omega = \{\pm 1\}$. $\boldsymbol{\upsilon} \in \mathbb{R}^N$ is the noise vector where each entry is Gaussian $\mathcal{N}(0,1)$ and independent identically distributed (i.i.d.), and $\mathbf{H} \in \mathbb{R}^{N\times N}$ denotes the channel matrix with i.i.d. $\mathcal{N}(0,1)$ entries. The normalization in (1) guarantees that SNR represents the signal-to-noise ratio per receive dimension (which we define as the ratio of the total transmit energy per channel use divided by the per-component noise variance as described in among others [5]). As explained further below, for analysis purposes we will focus on the regime where $\mathrm{SNR} > 2\ln(N)$, in order to get the probability of error of the ML detector to go to zero. Further, in our analysis, without loss of generality, we will assume that the

[1]For simplicity we have assumed that the receive and the transmit dimensions are the same, but the results presented in the paper can be generalized to cover different dimensions.

all minus one vector was transmitted, $\mathbf{s} = -\mathbf{1}$. Therefore

$$\mathbf{y} = \boldsymbol{v} - \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{1} \ . \tag{2}$$

We are considering a minimization of the average error probability $P(\mathbf{e}) \triangleq P(\hat{\mathbf{s}} \neq \mathbf{s})$, which is obtained by performing Maximum Likelihood Sequence Detection (here simply referred to as ML detection) given by

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\Omega^N} \left\| \mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{s} \right\|^2 \ . \tag{3}$$

III. Gibbs Sampling

One way of solving the optimization problem given in (3) is by using Markov Chain Monte Carlo (MCMC) simulations, which asymptotically converge to the optimal solution [13]. More specifically, the MCMC detector we investigate here is the Gibbs sampler, which computes the conditional probability of each symbol in the constellation set at the jth index in the estimated symbol vector. This conditional probability is obtained by keeping the $j-1$ other values in the estimated symbol vector fixed. Thus, in kth iteration the probability of the jth symbol adopts the value ω, is given as

$$p\left(\hat{\mathbf{s}}_j^{(k)} = \omega \,|\theta\right) = \frac{e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}-\sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\tilde{\mathbf{s}}_{j|\omega}\right\|^2}}{\sum_{\tilde{\mathbf{s}}_{j|\omega}\in\Omega} e^{-\frac{1}{2\alpha^2}\left\|\mathbf{y}-\sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\tilde{\mathbf{s}}_{j|\omega}\right\|^2}} \ , \tag{4}$$

where $\tilde{\mathbf{s}}_{j|\omega}^T \triangleq \left[\hat{\mathbf{s}}_{1:j-1}^{(k)}, \omega, \hat{\mathbf{s}}_{j+1:N_T}^{(k-1)}\right]^T$ and where we for simplicity have introduced $\theta = \left\{\hat{\mathbf{s}}^{(k-1)}, \mathbf{y}, \mathbf{H}\right\}$.[2] α represents a tunable parameter which controls the mixing time of the Markov chain, this parameter is also sometimes called the "temperature". The larger α is the faster the mixing time of the Markov chain will be, but as we will show in the paper, there is an upper limit on α, in order to ensure that the probability of finding the optimal solution in steady state is not exponentially small. The MCMC method will with probability $p\left(\hat{\mathbf{s}}_j^{(k)} = \omega\,|\theta\right)$ keep ω at the j'th index in estimated symbol vector, and compute conditional probability the $(j+1)$th index in a similar fashion. We define one iteration of the Gibbs sampler as a randomly-ordered update of all the $j = \{1, \ldots, N_T\}$ indices in the estimated symbol vector $\hat{\mathbf{s}}$.[3] The initialization of the symbol vector $\hat{\mathbf{s}}^{(0)}$ can either be chosen randomly or, alternatively, e.g. the zero-forcing solution can be used.

[2]When we compute the probability of symbol ω at the j'th position, we more precisely condition on the symbols $\hat{\mathbf{s}}_{1:j-1}^{(k)}$ and $\hat{\mathbf{s}}_{j+1:N_T}^{(k-1)}$, but to keep the notation simple, we do not explicitly state that in the equations above.

[3]We need a randomly-ordered update for the Markov chain to be reversible and for our subsequent analysis to go through. It is also possible to just randomly select a symbol j to update, without insisting that a full sequence be done. This also makes the Markov chain reversible and has the same steady state distribution. In practice a fixed, say sequential, order can be employed, although the Markov chain is no longer reversible. Note that our theoretical analysis is assuming randomly selected symbol updates for analytical convenience. In our experimental section we used a sequential updating order which empirically yields a slight convergence acceleration.

A. Complexity of the Gibbs sampler

The conditional probability for the j'th symbol in (4) can be computed efficiently by reusing the result obtained for the $j-1$'th symbol, when we evaluate $\left\|\mathbf{y} - \sqrt{\mathrm{SNR}/N}\mathbf{H}\tilde{\mathbf{s}}_{j|\omega}\right\|^2$. Since we are only changing the j'th symbol in the symbol vector, the difference $\mathbf{d}_j \triangleq \mathbf{y} - \sqrt{\mathrm{SNR}/N}\mathbf{H}\tilde{\mathbf{s}}_{j|\omega}$ can be expressed as

$$\mathbf{d}_j = \mathbf{d}_{j-1} - \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}_{1:N,j}\Delta s_{j|\omega} \ , \tag{5}$$

where $\Delta s_{j|\omega} \triangleq s_{j|\omega}^{(k)} - s_{j|\bar{\omega}}^{(k-1)}$. Thus, the computation of conditional probability of certain symbol in the j'th position costs $2N$ operations, where we define an operation as a Multiply and Accumulate (MAC) instruction.[4] This leads to a complexity of $\mathcal{O}\left(2N^2[|\Omega|-1]\right)$ operations per iteration. For further details on the implementation of the Gibbs sampler see [14].

IV. Probability of Error

In this paper, we are interested in evaluating the performance of the aforementioned Gibbs sampler, compared to the ML solution. To ease our analysis, we will assume that the ML detector finds the correct transmitted vector. Before we derive the probability of error for the ML detector, we will state a lemma which we will make repeated use of.

Lemma IV.1 (Gaussian Integral)**.** *Let* $\mathbf{v}$ *and* $\mathbf{x}$ *be independent Gaussian random vectors with distribution* $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ *each. Then, if* $1 - 2a^2\eta(1+2\eta) > 0$,

$$E\left\{e^{\eta(\|\mathbf{v}+a\mathbf{x}\|^2 - \|\mathbf{v}\|^2)}\right\} = \left(\frac{1}{1-2a^2\eta(1+2\eta)}\right)^{N/2} \ . \tag{6}$$

Proof: See Appendix VIII-A for a detailed proof. ■

Assuming that the vector $\mathbf{s} = -\mathbf{1}$ was transmitted, the ML detector will make an error if there exists a vector $\mathbf{s} \neq -\mathbf{1}$ such that

$$\left\|\mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{s}\right\|^2 \leq \left\|\mathbf{y} + \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{1}\right\|^2 = \|\boldsymbol{v}\|^2 \ .$$

In other words,

$$\begin{aligned} P_e &= \mathrm{Prob}\left(\left\|\mathbf{y} - \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\mathbf{s}\right\|^2 \leq \|\boldsymbol{v}\|^2\right) \\ &= \mathrm{Prob}\left(\left\|\boldsymbol{v} + \sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}(-\mathbf{1}-\mathbf{s})\right\|^2 \leq \|\boldsymbol{v}\|^2\right) \ , \end{aligned}$$

for some $\mathbf{s} \neq -\mathbf{1}$, which can be formulated as

$$P_e = \mathrm{Prob}\left(\left\|\boldsymbol{v} + 2\sqrt{\frac{\mathrm{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 \leq \|\boldsymbol{v}\|^2\right) \ ,$$

[4]We need to compute both the inner product $\mathbf{d}_j^T\mathbf{d}_j$ and the product $\mathbf{H}_{1:N,j}\Delta s_{j|\omega}$.

for some $\boldsymbol{\delta} \neq 0$. Note that in the above equation $\boldsymbol{\delta}$ is a vector of zeros and -1's. Now using the union bound

$$P_e \leq \sum_{\boldsymbol{\delta}\neq 0} \text{Prob}\left(\left\|\boldsymbol{v} + 2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 \leq \|\boldsymbol{v}\|^2\right). \tag{7}$$

We will use the Chernoff bound to bound the quantity inside the summation. Thus,

$$\text{Prob}\left(\left\|\boldsymbol{v} + 2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 \leq \|\boldsymbol{v}\|^2\right) \tag{8a}$$

$$\leq E\left\{e^{-\beta\left(\left\|\boldsymbol{v}+2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 - \|\boldsymbol{v}\|^2\right)}\right\} \tag{8b}$$

$$= \left(\frac{1}{1+8\frac{\text{SNR}\|\boldsymbol{\delta}\|^2}{N}\beta(1-2\beta)}\right)^{N/2}, \tag{8c}$$

where $\beta \geq 0$ is the Chernoff parameter, and where we have used Lemma IV.1 with $\eta = -\beta$ and $a = 2\sqrt{\frac{\text{SNR}\|\boldsymbol{\delta}\|^2}{N}}$, since

$$E\left\{\left(2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right)\left(2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right)^*\right\} = 4\frac{\text{SNR}\|\boldsymbol{\delta}\|^2}{N}\mathbf{I}_N.$$

The optimal value for β is $\frac{1}{4}$, which yields the tightest bound

$$\text{Prob}\left(\left\|\boldsymbol{v} + 2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 \leq \|\boldsymbol{v}\|^2\right) \leq \left(\frac{1}{1+\frac{\text{SNR}\|\boldsymbol{\delta}\|^2}{N}}\right)^{N/2}. \tag{9}$$

Note that this depends only on $\|\boldsymbol{\delta}\|^2$, the number of nonzero entries in $\boldsymbol{\delta}$. Plugging this into the union bound yields

$$P_e \leq \sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{\text{SNR}i}{N}}\right)^{N/2}. \tag{10}$$

Let us first look at the linear (i.e., i proportional to N) terms in the above sum. Thus,

$$\binom{N}{i}\left(\frac{1}{1+\frac{\text{SNR}i}{N}}\right)^{N/2} \approx e^{NH\left(\frac{i}{N}\right)-\frac{N}{2}\ln\left(1+\frac{\text{SNR}i}{N}\right)},$$

where $H(\cdot)$ is entropy in "nats". Clearly, if $\lim_{N\to\infty}\text{SNR} = \infty$, then the linear terms go to zero (superexponentially fast).

Let us now look at the sublinear terms. In particular, let is look at $i = 1$:

$$N\left(\frac{1}{1+\frac{\text{SNR}}{N}}\right)^{N/2} \approx Ne^{-\text{SNR}/2}.$$

Clearly, to have this term go to zero, we require that $\text{SNR} > 2\ln N$. A similar argument shows that all other sublinear terms also go to zero, and so.[5]

Lemma IV.2 (SNR scaling)**.** *If* $\text{SNR} > 2\ln N$*, then* $P_e \to 0$ *as* $N \to \infty$.

[5]Due to space constraints we only present a sketch of this bound. A rigorous proof can be given using the saddle point method, similarly to the proof in the next section.

V. Computing the optimal α

Assuming that the vector $\mathbf{s} = -\mathbf{1}$ has been transmitted, the probability of finding this solution *after the Markov chain has mixed* is simply $\pi_{-\mathbf{1}}$, the steady-state probability of being in the all -1 state. Clearly, if this probability is exponentially small, it will take exponentially long for the Gibbs sampler to find it. We will therefore insist that the mean of $\pi_{-\mathbf{1}}$ be only polynomially small.

A. Mean of $\pi_{-\mathbf{1}}$

This calculation has a lot in common with the one given in Section IV. Note that the steady state value of $\pi_{-\mathbf{1}}$ is simply

$$\pi_{-\mathbf{1}} = \frac{e^{-\frac{1}{2\alpha^2}\left\|\boldsymbol{y}+\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\mathbf{1}\right\|^2}}{\sum_{\mathbf{s}} e^{-\frac{1}{2\alpha^2}\left\|\boldsymbol{y}+\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\mathbf{s}\right\|^2}} \tag{11a}$$

$$= \frac{e^{-\frac{1}{2\alpha^2}\|\boldsymbol{v}\|^2}}{\sum_{\mathbf{s}} e^{-\frac{1}{2\alpha^2}\left\|\boldsymbol{v}+\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}(\mathbf{s}-\mathbf{1})\right\|^2}} \tag{11b}$$

$$= \frac{e^{-\frac{1}{2\alpha^2}\|\boldsymbol{v}\|^2}}{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left\|\boldsymbol{v}+2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2}} \tag{11c}$$

$$= \frac{1}{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left(\left\|\boldsymbol{v}+2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 - \|\boldsymbol{v}\|^2\right)}}, \tag{11d}$$

where $\boldsymbol{\delta}$ is a vector of zeros and ones and the summations (over $\mathbf{s}$ and $\boldsymbol{\delta}$) are over 2^n terms.

Now, by Jensen's inequality

$$E\{\pi_{-\mathbf{1}}\} \geq \frac{1}{E\left\{\frac{1}{\pi_{-\mathbf{1}}}\right\}} \tag{12a}$$

$$= \frac{1}{E\left\{\sum_{\boldsymbol{\delta}} e^{-\frac{1}{2\alpha^2}\left(\left\|\boldsymbol{v}+2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 - \|\boldsymbol{v}\|^2\right)}\right\}} \tag{12b}$$

$$= \frac{1}{\sum_{\boldsymbol{\delta}} E\left\{e^{-\frac{1}{2\alpha^2}\left(\left\|\boldsymbol{v}+2\sqrt{\frac{\text{SNR}}{N}}\mathbf{H}\boldsymbol{\delta}\right\|^2 - \|\boldsymbol{v}\|^2\right)}\right\}} \tag{12c}$$

$$= \frac{1}{1+\sum_{\boldsymbol{\delta}\neq 0}\left(\frac{1}{1+4\frac{\text{SNR}\|\boldsymbol{\delta}\|^2}{N}\frac{1}{\alpha^2}\left(1-\frac{1}{\alpha^2}\right)}\right)^{N/2}} \tag{12d}$$

$$= \frac{1}{1+\sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{2i}{N}}\right)^{N/2}}. \tag{12e}$$

In (12d) we have used Lemma IV.1 and in (12e) we have defined $\beta \triangleq 4\text{SNR}\frac{1}{\alpha^2}\left(1-\frac{1}{\alpha^2}\right)$. While it is possible to focus on the linear and sublinear terms in the above summation separately, to give conditions for $E\{\pi_{-\mathbf{1}}\}$ to have the form of $1/\text{poly}(N)$, we will be interested in the exact exponent

and so will need a more accurate estimate. To do this we shall use saddle point integration. Note that

$$\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx e^{NH(\frac{i}{N})-\frac{N}{2}\ln\left(1+\frac{\beta i}{N}\right)} ,$$

where again $H(\cdot)$ represents the entropy in "nats". And so the summation in the denominator of (12e) can be approximated as a Stieltjes integral:

$$\sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx N\sum_{i=1}^{N} e^{NH(\frac{i}{N})-\frac{N}{2}\ln\left(1+\frac{\beta i}{N}\right)}\frac{1}{N} \tag{13a}$$

$$\approx N\int_0^1 e^{NH(x)-\frac{N}{2}\ln(1+\beta x)}dx . \tag{13b}$$

For large N, this is a saddle point integral and can be approximated by the formula

$$\int_0^1 e^{Nf(x)}dx \approx \sqrt{\frac{2\pi}{N|f''(x_0)|}}e^{Nf(x_0)} , \tag{14}$$

where x_0, is the saddle point of $f(\cdot)$, i.e.,$f'(x_0)=0$. In our case,

$$f(x) = -x\ln x-(1-x)\ln(1-x)-\frac{1}{2}\ln(1+\beta x) ,$$

and so

$$f'(x) = \ln\frac{1-x}{x}-\frac{1}{2}\frac{\beta}{1+\beta x} .$$

In general, it is not possible to solve for $f'(x_0)=0$ in closed form. However, in our case, if we assume that $\beta = 4\text{SNR}\frac{1}{\alpha^2}(1-\frac{1}{\alpha^2}) \gg 1$ (which is true since the SNR grows at least logarithmically), then it is not too hard to verify that the saddle point is given by

$$x_0 = e^{-\frac{\beta}{2}} . \tag{15}$$

And hence $f(x_0)=$

$$-e^{-\frac{\beta}{2}}\ln e^{-\frac{\beta}{2}}-(1-e^{-\frac{\beta}{2}})\ln(1-e^{-\frac{\beta}{2}})-\frac{1}{2}\ln(1+\beta e^{-\frac{\beta}{2}})$$

$$\approx \frac{\beta}{2}e^{-\frac{\beta}{2}}+e^{-\frac{\beta}{2}}-\frac{1}{2}\beta e^{-\frac{\beta}{2}} = e^{-\frac{\beta}{2}} ,$$

and further plugging x_0 into $f''(x) = -\frac{1}{x}-\frac{1}{1-x}-\frac{1}{2}\frac{\beta^2}{(1+\beta x)^2}$, yields

$$f''(x_0) \approx -e^{\frac{\beta}{2}}-1+\frac{1}{2}\beta^2 \approx -e^{\frac{\beta}{2}} . \tag{16}$$

Replacing these into the saddle point expression in (14) show that

$$\sum_{i=1}^{N}\binom{N}{i}\left(\frac{1}{1+\frac{\beta i}{N}}\right)^{N/2} \approx \sqrt{2\pi/N}\exp\left(Ne^{-\frac{\beta}{2}}-\frac{\beta}{4}\right) . \tag{17}$$

We want $E\{\pi_{-\mathbf{1}}\}$ to behave as $\frac{1}{N^\zeta}$ and according to (12) this means that we want the expression in (17) to behave as N^ζ. Let us take

$$e^{Ne^{-\frac{\beta}{2}}} = N^\zeta .$$

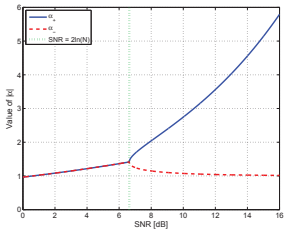


Figure 1: Value of α vs. SNR for system size $N=10$.

Solving for β yields

$$\beta = 4\text{SNR}\frac{1}{\alpha^2}\left(1-\frac{1}{\alpha^2}\right) = 2\left(\ln N-\ln\ln N-\ln\zeta\right) . \tag{18}$$

Incidentally, this choice of β yields $e^{-\frac{\beta}{4}} \approx \frac{1}{\sqrt{N}}$, and so we have the following result.

Lemma V.1 (Mean of $\pi_{-\mathbf{1}}$)**.** *If α is chosen such that*

$$\frac{\alpha^2}{1-\frac{1}{\alpha^2}} = \frac{2SNR}{\ln N-\ln\ln N-\ln\zeta} , \tag{19}$$

then

$$E\{\pi_{-\mathbf{1}}\} \geq N^{-\zeta} . \tag{20}$$

B. Value of α

Note that from (12e) it is clear that the larger β is, the larger $\pi_{-\mathbf{1}}$ is. Therefore, the range of α that gives a polynomially small probability to $\pi_{-\mathbf{1}}$ is

$$\frac{\alpha^2}{1-\frac{1}{\alpha^2}} \geq \frac{2\text{SNR}}{\ln N-\ln\ln N-\ln\zeta} . \tag{21}$$

It can be shown that in the regime, SNR $> 2\ln N$, the above quadratic inequality in α has two positive real solutions, $\alpha_+ \geq \alpha_-$, and that the inequality holds for all $\alpha \in [\alpha_-,\alpha_+]$.

We know that, the larger α is, the faster the Markov chain mixes.[6] Therefore it is reasonable that we choose the largest permissible value for α, i.e., α_+.

Figures 1 and 2 show the values of α_+ and α_- as a function of SNR for systems with $N=10$ and $N=50$, when we have $\zeta = 1/\ln(N)$.

C. Mixing time of Markov Chain

One open question is whether the Markov chain is rapidly mixing when using the strategy above for choosing α. This is something we are currently investigating, and the simulations presented in Section VI seem to indicate that this is the case. Furthermore, the simulations also suggest that the computed value of α is very close to the optimal choice, even in the case where the condition SNR $> 2\ln(N)$ is not satisfied.

[6]In general, there is a trade-off between faster mixing time of the Markov chain (due to an increase of α) versus slower encountering the optimal solution in steady-state. In fact, at infinite temperature our algorithm reduces to a random walk in a hypercube which mixes in $O(N\ln N)$ time.

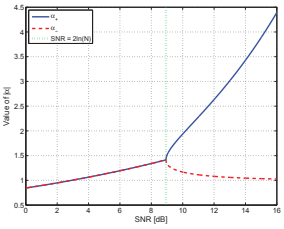


Figure 2: Value of α vs. SNR for system size $N = 50$.

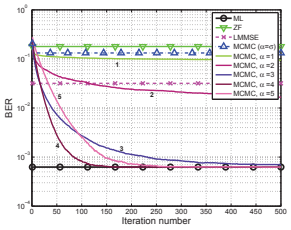


Figure 4: BER vs. iterations, 10×10 system. SNR = 14 dB.

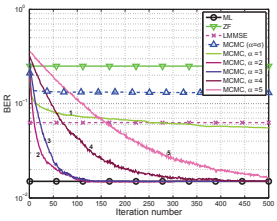


Figure 3: BER vs. iterations, 10×10. SNR = 10 dB.

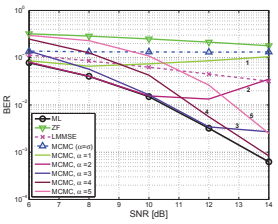


Figure 5: BER vs. SNR, 10×10. Number of iterations, $k = 100$.

VI. Simulation Results

In this section we present simulation results for a MIMO $N \times N$ system with a full square channel matrix containing i.i.d. Gaussian entries. In Fig. 3 and Fig. 4 the Bit Error Rate (BER) of the Gibbs sampler, initialized with a random $\mathbf{s}$, has been evaluated as a function of the number of iterations in a 10×10 system using a variety of α values. Thereby, we can inspect how the parameter α affects the convergence rate of the Gibbs sampler. The performance of the Maximum Likelihood (ML), the Zero-Forcing (ZF), and the Linear Minimum Mean Square Error (LMMSE) detector has also been plotted, to ease the comparison of the Gibbs sampler with these. It is seen that the Gibbs sampler outperforms both the ZF and the LMMSE detector after only a few iterations in all the presented simulations, when the tuning parameter α is chosen properly. Furthermore, it is observed that the parameter α has a huge influence on the convergence rate and that the Gibbs sampler converges toward the ML solution as a function of the iterations.[7] The optimal value of α (in terms of convergence rate) is quite close to the theoretical values from Fig. 1 of $\alpha_+ = 2.7$ and $\alpha_+ = 4.6$ at SNR's at 10 and $14dB$, respectively. It is also observed that the performance of the Gibbs sampler is significantly deteriorated if the temperature parameter is chosen based on the SNR (and thereby on the noise variance), such that $\alpha = \sigma \triangleq 1/\text{SNR}$. Thus, the latter strategy is clearly not a wise choice.

Figure 5 shows the BER performance for the MCMC detector for fixed number of iterations, $k = 100$. From the figure we see that the SNR has a significant influence on the optimal choice of α given a fixed number of iterations.

The performance of the Gibbs sampler is also shown for a 50×50 system, which represents a ML decoding problem of huge complexity where an exhaustive search would require $2^{50} \approx 10^{15}$ evaluations. For this problem even the sphere decoder has an enormous complexity under moderate SNR.[8] Therefore, it has not been possible to simulate the performance of this decoder within a reasonable time and we have therefore "cheated" a little by initializing the radius of the sphere to the minimum of either the norm of the transmitted symbol vector or the solution found by the Gibbs sampler. This has been done in order to evaluate the BER performance of the optimal detector. Figure 6 shows the BER curve as a function of the iteration number, while Figure 7 illustrates the BER curve vs. the SNR. From Figure 6 we see that there is a quite good correspondence between the simulated α and the theoretical value $\alpha_+ = 2.6$ obtained from Figure 2. The average complexity (MAC pr. symbol vector) of the Gibbs

[7] It should be noted that the way we decode the symbol vector to a given iteration, is to select the symbol vector which has the lowest cost function in all the iterations up to that point in time.

[8] In fact, it can be shown that, for SNR = $\mathcal{O}(\ln N)$, the lower bound on the complexity of the sphere decoder obtained in [6] is exponential.

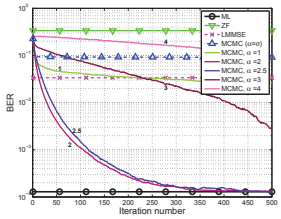


Figure 6: BER vs. iterations, 50×50 system. SNR = 12 dB.

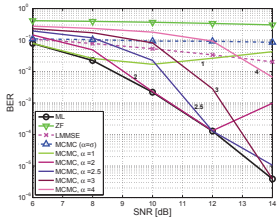


Figure 7: BER vs. SNR, 50×50 system. Num. of iter., $k = 500$.

Table I: Complexity of SD and Gibbs Sampler (GS).

N	Method	SNR 6 dB	10 dB	14 dB
10	GS	$9.8 \cdot 10^3$	$10.9 \cdot 10^3$	$16.4 \cdot 10^3$
	SD	$10.0 \cdot 10^3$	$1.7 \cdot 10^3$	$1.5 \cdot 10^3$
50	GS	$7.6 \cdot 10^5$	$9.5 \cdot 10^5$	$10.6 \cdot 10^5$
	SD	$\gg 1.9 \cdot 10^9$	$\gg 1.9 \cdot 10^9$	$37.7 \cdot 10^5$

sampler having a BER performance comparable with the ML detector is shown in Table I. The SD has been included as a reference.[9] It is observed that the complexity of the Gibbs sampler is not affected by the SNR as much as the SD.

VII. Conclusion

In this paper we considered solving the integer least-squares problem using Monte Carlo Markov Chain Gibbs sampling. The novelty of the proposed MCMC method is that, unlike simulated annealing techniques, we have a fixed temperature parameter in all the iterations, with the property that after the Markov chain has mixed, the probability of encountering the optimal solution is only polynomial small (i.e. not exponentially small). We further compute the optimal (here largest) value of the temperature parameter that guarantees this. Simulation results indicate the sensitivity of the method to the choice of the temperature parameter and show that our computed value gives a very good approximation to its optimal value. Investigating whether the Markov chain mixes in polynomial time for this choice of temperature parameter is currently under investigation.

[9] It has not been possible to simulate the SD for a 50×50 system when $SNR \leq 10dB$ and, therefore, the complexity of $SNR = 12dB$ has been used a lower bound.

VIII. Appendix

A. Proving Lemma IV.1

Lemma IV.1 (Gaussian Integral) *Let* $\mathbf{v}$ *and* $\mathbf{x}$ *be independent Gaussian random vectors with distribution* $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ *each. Then*

$$E\left\{e^{\eta\left(\|\mathbf{v}+a\mathbf{x}\|^2-\|\mathbf{v}\|^2\right)}\right\} = \left(\frac{1}{1-2a^2\eta(1+2\eta)}\right)^{N/2}. \quad (22)$$

Proof: In order to determine the expected value we compute the multivariate integral

$$E\left\{e^{\eta\left(\|\mathbf{v}+a\mathbf{x}\|^2-\|\mathbf{v}\|^2\right)}\right\}$$

$$= \int \frac{d\mathbf{x}d\mathbf{v}}{(2\pi)^N} e^{-\frac{1}{2}\left[\begin{array}{cc}\mathbf{v}^T, & \mathbf{x}^T\end{array}\right]\left[\begin{array}{cc}\mathbf{I}_N & -2a\eta\mathbf{I}_N \\ -2a\eta\mathbf{I}_N & (1-2a^2\eta)\mathbf{I}_N\end{array}\right]\left[\begin{array}{c}\mathbf{v} \\ \mathbf{x}\end{array}\right]}$$

$$= \frac{1}{\det^{N/2}\left[\begin{array}{cc}1 & -2a\eta \\ -2a\eta & 1-2a^2\eta\end{array}\right]} = \left(\frac{1}{1-2a^2\eta(1+2\eta)}\right)^{N/2}.$$

Thus, Lemma IV.1 has hereby been proved. ∎

References

[1] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.*, vol. 51, no. 3, pp. 389–399, 2003.

[2] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm. I. Expected Complexity," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 2806–2818, Aug. 2005.

[3] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm. II. Generalizations, Second-Order Statistics, and Applications to Communications," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 2819–2834, Aug. 2005.

[4] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[5] M. O. Damen, H. E. Gamal, and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," *IEEE Trans. on Info. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[6] J. Jaldén and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 1474–1484, Apr. 2005.

[7] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2 edition, 2004.

[8] O. Häggström, *Finite Markov chains and algorithmic applications*, Cambridge University Press, 2002.

[9] H. Zhu, B. Farhang-Boroujeny, and R.R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods," *IEEE Sig. Proc. Letters*, vol. 12, pp. 669–672, 2005.

[10] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 5, pp. 1896–1909, 2006.

[11] X. Wang and V. H. Poor, *Wireless Communications Systems: Advanced Techniques for Signal Reception*, Prentice Hall, 2003.

[12] R. Chen, JS Liu, and X. Wang, "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 255–270, 2002.

[13] D.J.C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

[14] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov Chain Monte Carlo Based Multiuser/MIMO Detector," *IEEE Trans. on Circuits and Sys. - I: Regular Papers*, vol. 56, pp. 246–255, 2009.

Bibliography

[1] X. Wang and V. H. Poor, *Wireless Communications Systems: Advanced Techniques for Signal Reception.* Prentice Hall, 2003.

[2] L. P. B. Christensen, "Signal Processing for Improved Wireless Receiver Performance," Ph.D. dissertation, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2007.

[3] J. G. Proakis, *Digital Communications.* McGraw-Hill, 4th ed., 2001.

[4] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge Univ Press, 2005.

[5] T. S. Rappaport, *Wireless Communications, Principles and Practice.* Prentice Hall, 2nd ed., 2002.

[6] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm. I. Expected Complexity," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 2806–2818, Aug. 2005.

[7] M. Ajtai, "The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing.* ACM New York, NY, USA, 1998, pp. 10–19.

[8] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, 2nd ed., 1993.

[9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Info. Theory*, vol. 20, pp. 284–287, 1974.

[10] A. D. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithms," *IEEE Trans. on Info. Theory*, vol. 13, pp. 260–269, Apr. 1967.

[11] G. D. Forney Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Info. Theory*, vol. 18, no. 3, pp. 363–378, 1972.

[12] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] 3GPP TS 45.005, *3GPP TSG GERAN*. Radio Transmission and Reception (Release 5).

[14] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[15] J. R. Barry, "The BCJR Algorithm for Optimal Equalization," *School of Electrical and Computer Engineering, Georgia institute of Technology, March*, 2000.

[16] 3GPP TS 45.004, *3GPP TS GERAN*. Radio Access Network; Modulation (Release 8), 2008.

[17] P. Monsen, "Feedback Equalization for Fading Dispersive Channels," *IEEE Trans. on Info. Theory*, vol. 17, no. 1, pp. 56–64, 1971.

[18] C. A. Belfiore and J. H. Park Jr, "Decision Feedback Equalization," *Proceedings of the IEEE*, vol. 67, no. 8, pp. 1143–1156, 1979.

[19] A. Duel-Hallen, , and C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. on Commun.*, vol. 37, pp. 428–436, May 1989.

[20] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback," *IEEE Trans. on Commun.*, vol. 36, pp. 13–20, Jan. 1988.

[21] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Info. Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[22] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice Hall, 1995.

[23] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[24] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[25] A. H. Sayed and T. Kailath, "A Survey of Spectral Factorization Methods," *Numerical Linear Algebra with Applications*, vol. 8, pp. 467–496, Jul. 2001.

[26] P. Laurent, "Exact and Approximate Construction of Digital Phase Modulations by Superposition of Amplitude Modulated Pulses (AMP)," *IEEE Trans. on Commun.*, vol. 34, no. 2, pp. 150–160, 1986.

[27] J. F. Claerbout, *Fundamentals of Geophysical Data Processing.* Blackwell Scientific Publications, 1985.

[28] N. Al-Dhahir and J. Cioffi, "Finite-length vs. infinite-length MMSE-DFE: The Connection," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, 1993, pp. 677–681.

[29] D. Youla, "On the Factorization of Rational Matrices," *Information Theory, IRE Transactions on*, vol. 7, no. 3, pp. 172–189, 1961.

[30] D. Youla and N. Kazanjian, "Bauer-Type Factorization of Positive Matrices and the Theory of Matrix Polynomials Orthogonal on the Unit Circle," *IEEE Trans. on Circuits and Systems*, vol. 25, no. 2, pp. 57–69, 1978.

[31] Y. Rozanov, "Spectral Properties of Multivariate Stationary Processes and Boundary Properties of Analytic Matrices," *Theory of Probability and its Applications*, vol. 5, p. 362, 1960.

[32] A. Yaglom, "Effective Solutions of Linear Approximation Problems for Multivariate Stationary Processes with a Rational Spectrum," *Theory of Probability and its Applications*, vol. 5, pp. 239–264, 1960.

[33] R. F. H. Fischer, "Sorted Spectral Factorization of Matrix Polynomials in MIMO Communications," *IEEE Trans. on Info. Theory*, vol. 53, pp. 945–951, Jun. 2005.

[34] J. E. Dennis Jr., J. F. Traub, and R. P. Weber, "The Algebraic Theory of Matrix Polynomials," *SIAM Journal on Numerical Analysis*, vol. 13, pp. 813–845, Dec. 1976.

[35] ——, "Algorithms for Solvents of Matrix Polynomials," *SIAM Journal on Numerical Analysis*, pp. 523–533, 1978.

[36] V. Kucera, "Factorization of Rational Spectral Matrices: A Survey of Methods," in *International Conference on Control 1991.*, 1991, pp. 1074–1078.

[37] B. Anderson, K. Hitz, and N. Diem, "Recursive Algorithm for Spectral Factorization," *IEEE Transactions on Circuits and Systems*, vol. 21, no. 6, pp. 742–750, 1974.

[38] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.*, vol. 51, no. 3, pp. 389–399, 2003.

[39] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm. II. Generalizations, Second-Order Statistics, and Applications to Communications," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 2819–2834, Aug. 2005.

[40] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Info. Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[41] M. O. Damen, H. E. Gamal, and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," *IEEE Trans. on Info. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[42] U. Fincke and M. Pohst, "Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis," *Mathematics of Computation*, pp. 463–471, 1985.

[43] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," *Mathematical Programming*, vol. 66, pp. 181–199, 1994.

[44] E. Viterbo and J. Boutros, "A Universal Lattice Code Decoder for Fading Channels," *IEEE Trans. on Info. Theory*, vol. 45, no. 5, pp. 1639–1642, 1999.

[45] J. Jaldén and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 1474–1484, Apr. 2005.

[46] R. Gowaikar and B. Hassibi, "Statistical Pruning for Near-Maximum Likelihood Decoding," *IEEE Trans. on Sig. Proc.*, vol. 55, pp. 2661–2675, Jun. 2007.

[47] M. Hansen, L. P. B. Christensen, and O. Winther, "On Sphere Detection and Minimum-Phase Prefiltered Reduced-State Sequence Estimation," in *GLOBECOM'07*, 2007, pp. 4237–4241.

[48] M. Hansen, "Comparison of Optimal and Near-Optimal Detection in GSM/EDGE," Master's thesis, Technical University of Denmark, Informatics and Mathematical Modelling, 2006.

[49] G. B. Giannakis and W. Zhao, "Sphere Decoding Algorithms with Improved Radius Search," in *Wireless Communications and Networking Conference, (WCNC)*, 2004, pp. 2290–2294.

[50] B. Hassibi, H. Vikalo, and U. Mitra, "Sphere-Constrained ML Detection for Frequency-Selective Channels," in *ICASSP'03*, vol. 4, 2003, pp. 1–4.

[51] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative Decoding for MIMO Channels via Modified Sphere Decoding," *IEEE Trans. on Wireless Commun.*, vol. 3, no. 6, pp. 2299–2311, 2004.

[52] M. Hansen, L. P. B. Christensen, and O. Winther, "Computing the Minimum-Phase Filter using QL-Factorization," *Submitted June 2009, unpublished.*

[53] N. Al-Dhahir and J. M. Cioffi, "MMSE Decision-Feedback Equalizers: Finite-Length Results," *IEEE Trans. on Info. Theory*, vol. 41, pp. 961–975, Jul. 1995.

[54] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

[55] R. Gray, *Toeplitz And Circulant Matrices: A Review (Foundations and Trends in Communications and Information Theory)*. Now Publishers Inc. Hanover, MA, USA, 2006.

[56] T. I. Laakso and V. Välimäki, "Energy-Based Effective Length of the Impulse Response of a Recursive Filter," in *ICASSP'98*, vol. 3, 1998, pp. pp. 1253–1256.

[57] D. P. Mandic and I. Yamada, "Machine Learning and Signal Processing Applications of Fixed Point Theory," in *Tutorial in IEEE ICASSP'07.*, 2007.

[58] M. Hansen and L. P. B. Christensen, "Efficient Minimum-Phase Prefilter Computation Using Fast QL-Factorization," in *ICASSP'09*, 2009.

[59] A. W. Bojanczyk, R. P. Brent, and F. d. Hoog, "QR Factorization of Toeplitz Matrices," *Numer. Math.*, vol. 49, pp. 81–94, Jul. 1986.

[60] D. R. Sweet, "Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 43, pp. 1–21, Feb. 1984.

[61] S. Qiao, "Hybrid Algorithm for Fast Toeplitz Orthogonalization," *Numer. Math.*, vol. 53, pp. 351–366, May 1988.

[62] J. Chun, T. Kailath, and H. Lev-Ari, "Fast Parallel Algorithms for QR and Triangular Factorization," *SIAM Journal on Scientific and Statistical Computing*, vol. 8, pp. 899–913, 1987.

[63] H. Park and L. Elden, "Fast and Accurate Toeplitz Matrix Triangulation for Linear Prediction," in *Workshop on VLSI Signal Processing, VI.*, 1993, pp. 343–351.

[64] A. W. Bojanczyk, Brent, R. P., Dooren, P. van, and F. d. Hoog, "A Note on Downdating the Cholesky Factorization," *SIAM J. Sci. Stat. Comput.*, vol. 8, pp. 210–221, May 1987.

[65] D. R. Sweet, "Fast Block Toeplitz Orthogonalization," *Numer. Math.*, vol. 58, pp. 613–629, 1991.

[66] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, 4th ed., 2002.

[67] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for Modifying Matrix Factorizations," *Mathematics of Computation*, pp. 505–535, 1974.

[68] W. H. Gerstacker, F. Obernosterer, M. R., and J. B. Huber, "On Prefilter Computation for Reduced-State Equalization," *IEEE Trans. on Wireless Commun.*, vol. 1, pp. 793–800, Oct. 2002.

[69] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer, 2004.

[70] O. Häggström, *Finite Markov Chains and Algorithmic Applications.* Cambridge University Press, 2002.

[71] D. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[72] H. Zhu, B. Farhang-Boroujeny, and R. Chen, "On Performance of Sphere Decoding and Markov Chain Monte Carlo Detection Methods," *IEEE Sig. Proc. Letters*, vol. 12, pp. 669–672, 2005.

[73] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov Chain Monte Carlo Algorithms for CDMA and MIMO Communication Systems," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 5, pp. 1896–1909, 2006.

[74] R. Chen, J. S. Liu, and X. Wang, "Convergence Analyses and Comparisons of Markov Chain Monte Carlo Algorithms in Digital Communications," *IEEE Trans. on Sig. Proc.*, vol. 50, no. 2, pp. 255–270, 2002.

[75] H. Vikalo and B. Hassibi, "Maximum-Likelihood Sequence Detection of Multiple Antenna Systems over Dispersive Channels via Sphere Decoding," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 5, pp. 525–531, 2002.

[76] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov Chain Monte Carlo Based Multiuser/MIMO Detector," *IEEE Trans. on Circuits and Sys. - I: Regular Papers*, vol. 56, pp. 246–255, 2009.

[77] T. Hagerup and C. Rüb, "A Guided Tour of Chernoff Bounds," *Information processing letters*, vol. 33, no. 6, pp. 305–308, 1990.

[78] E. W. Weisstein, "Stieltjes Integral," *MathWorld - A Wolfram Web Resource*, http://mathworld.wolfram.com/StieltjesIntegral.html.

[79] D. W. Stroock, *A Concise Introduction to the Theory of Integration*. Birkhauser, 3rd ed., 1999.

[80] A. Sinclair, *Markov Chain Monte Carlo: Foundations and Applications*. Lecture note from Berkeley course CS294: Lecture 5: September 17, 2009.