Reduced Complexity Sequential Monte Carlo Algorithms for Blind Receivers

A Thesis
Presented to
The Academic Faculty

by

Soner Özgür

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Reduced Complexity Sequential Monte Carlo Algorithms for Blind Receivers

Approved by:

Dr. Douglas B. Williams, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. James H. McClellan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. G. Tong Zhou
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gordon Stüber
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Brani Vidakovic
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: April 2006

This dissertation is dedicated to my parents and my brother.

Mukaddes, Sebahattin, and İlker:

Thank you for your love, encouragement, and support.

ACKNOWLEDGEMENTS

The work on this thesis has been challenging and inspiring. The thesis would not have been possible without the tremendous support and encouragement of a number of people.

I would like to thank my advisor Dr. Douglas B. Williams for his guidance and support. He has given me several new ideas and infused me with confidence and enthusiasm through his fruitful discussions. He has been the ideal advisor in every aspect.

I also thank the members of my thesis committee, Dr. James H. McClellan, Dr. G. Tong Zhou, Dr. Gordon L. Stuber, and Dr. Brani Vidakovic for their comments and suggestions. Their feedback has greatly improved this thesis my research.

I would also like to thank many former students at Georgia Tech. I thank Fatih Demirkol, Volkan Cevher, Ali Erdem Ertan, Cagatay Candan and Bahadir Güntürk for their technical discussions and for inspiring me with their own approaches to my research problems.

I would like to thank Toygar Akgün for being a great roommate during our internship in San Diego. I also thank all the students in CSIP for their continuous support and high morale.

I thank Guclu Onaran, Baris Bicen, and Andrew Yanovski for their friendship and helping me go through my dissertation.

Finally I would like to acknowledge my parents and my brother for their support and encouragement. Their love and support helped much in coping with stress and pressure.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Monte Carlo algorithms can be used to estimate the state of a system given relative observations. In this dissertation, these algorithms are applied to physical layer communications system models to estimate channel state information, to obtain soft information about transmitted symbols or multiple access interference, or to obtain estimates of all of these by joint (or dual) estimation.

Initially, we develop and analyze a multiple access technique utilizing mutually orthogonal complementary sets (MOCS) of sequences. These codes deliberately introduce interchip interference, which is naturally eliminated during processing at the receiver. However, channel impairments can destroy their orthogonality properties and additional processing becomes necessary.

We utilize Monte Carlo algorithms to perform joint channel and symbol estimation for systems utilizing MOCS sequences as spreading codes. Linearity and Gaussianity in the system model simplifies the problem significantly. Using Rao-Blackwellization, we obtain and track only the first two moments of the channel, which is modeled as a random vector. However, dense signaling constellations, multiuser environments, and the interchannel interference introduced by the spreading codes all increase the dimensionality of the symbol state space significantly. A full maximum likelihood solution is computationally expensive and generally not practical. However, obtaining the optimum solution is critical and looking at only a part of the symbol space is generally not a good solution. Unlike estimation in continuous spaces, where a small error is forgivable, incorrect symbol detection, even if it is the neighboring one in the constellation, causes bit errors. We have sought algorithms that would guarantee that the correct transmitted symbol is considered, while only sampling a portion of the full symbol space.

We have developed a method that partitions the symbol state-space based on the recursive structure of the posteriors. The Markovian structure enables us to sample a significantly smaller subspace and obtain an intermediate likelihood function. We then proceed by updating the likelihood based on the next observation, which adds additional information about the transmitted symbol. The performance of the proposed method is comparable to the Maximum Likelihood (ML) algorithm. While the computational complexity of ML increases exponentially with the dimensionality of the problem, the complexity of our approach increases only quadratically.

Markovian structures such as the one imposed by MOCS spreading sequences can be seen in other physical layer structures as well. We have applied this partitioning approach with some modification to blind equalization of frequency selective fading channel and to multiple-input multiple output (MIMO) receivers that track channel changes.

The performance of the regular and simplified Monte Carlo algorithms are analyzed. The effects of channel order, number of antennas, cardinality of the signaling constellation, receiver noise, interference power, and spatial correlation are demonstrated through simulations. The performance is compared to the performance of previous algorithms that tackle the same problems.

Additionally, we develop a method that obtains a metric for quantifying the convergence rate of Monte Carlo algorithms. Our approach yields an eigenvalue based method that is useful in identifying sources of slow convergence and estimation inaccuracy.

CHAPTER I

INTRODUCTION

In wireless communications, transmitting users must use a common channel, whose parameters are unknown to them, to transmit information. The receiver also typically starts with no information about the channel and must determine the transmitted information. Additionally, the receiver is often interested in only one particular message and discards the information originating from other users. All of this communication must be coordinated somehow, so that all the users in the communications system are able to deliver their messages.

Since the transmitting medium is a single common channel, users must share it intelligently. Various access methods have been developed to enable the sharing of channels, such as time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA).

FDMA and TDMA assign orthogonal channels to users. The users are allocated slices of the frequency spectrum or the transmission time for their exclusive use. These channels are orthogonal because ideally no interference exists among them, but in practice distortions introduce interference. CDMA techniques, on the other hand, allow the users to use all of the time and frequency resources simultaneously by assigning a code to each user. The structure of the codes determines how these resources are used. In the direct sequence CDMA (DS-CDMA), these codes are called pseudo noise (PN) codes. The inherent non-perfect orthogonality of these codes and various channel effects introduce interference issues.

Under Rayleigh fading, multiple transmit and receive antennas canb increase the capacity of a communications channel [96, 97]. Various codes have been developed that increase the diversity of the system in such communication channels [74–76]. With all of these improvements over the single user channel, similar improvements can be expected in the respective multiuser communication channels. The interference issues, however, become

more complicated as the number of antennas per user increases. Clearly the implementation of the optimum multiuser detector is computationally prohibitive [83]. To combat the complexity issue, various suboptimal multiuser detectors have been developed [62, 83, 95].

In this dissertation, we concentrate on mutually orthogonal complementary sets (MOCS) of sequences instead of PN sequences. The former have properties that make them suitable for scenarios where uncoupled parallel channels exist. We construct a low complexity minimum mean square error (MMSE) multiuser detector [58]. We evaluate its performance in a serially concatenated turbo scheme with a channel code. We study its performance through simulations and analytically provide upper bounds on the diversity order of the system.

In the above scenarios, the receiver was assumed to have complete channel state information (CSI), which can be achieved by periodical transmission of training symbols known by the receiver [62]. Blind techniques based on knowledge of user codes or statistics of transmitted symbols also exist [37, 95]. Particularly interesting are the Monte Carlo Markov chain techniques, because they yield soft information outputs and can be utilized in iterative multiuser receivers.

We propose a method to decrease the computational demand of the standard SMC algorithm when used as a multiuser detector. Monte Carlo algorithms are known for their high computational complexity. However, they are very useful when the analytical solution of a problem is intractable. Maximum likelihood (ML) sequence estimation/detection (MLSE or MLSD) solutions are known to be optimal, but their complexity increases exponentially with the dimension of the parameter to be estimated [62]. SMC algorithms provide an approximation to the solution and, yet, approach the accuracy of ML detectors.

To analyze the improvements, we utilize this approach in single user scenarios. We propose an SMC based equalizer design for frequency selective fading channel communications. We compare the performance to other type equalizers, and analyze how the performance degrades as the number of used particles decreases. We also construct a receiver for a multi-antenna communications system. We analyze the improvements and the effects of the utilized blind algorithms on diversity.

We propose a method to analyze and quantify the convergence rate of SMC algorithms.

This approach enables one to quantify the effects of changes made to the algorithm, such as choosing different proposal functions, using different resampling methods, or simply changing the number of particles.

The remainder of this chapter is organized as follows. We briefly describe some basic multiuser detector techniques. Then, we introduce iterative (turbo) detection methods and discuss the internal mechanics of such receivers. Next, we introduce Monte Carlo methods and give a brief description of the most commonly used kinds. We discuss the state of the art of Monte Carlo methods in telecommunications literature. We conclude the chapter with a summary of the contributions of this thesis.

1.1 Multiuser Detectors

Multiuser detection has been studied extensively. In this section we describe some basic multiuser detector techniques for DS-CDMA.

The baseband received vector after chip-matched filtering can be expressed with the following equation.

$$\mathbf{r}[n] = \sum_{k=1}^{K} A_k b_k[n]\mathbf{s}_k + \eta[n], \tag{1}$$

where A_k is kth user's channel gain, $b_k[n]$ is kth user's data symbol, $\mathbf{s}_k$ is the kth user's PN sequence vector, and $\eta[n]$ is white receiver noise with a complex Gaussian pdf [83, 95] and variance σ^2. The above equation can be written in a more compact form by constructing $\mathbf{A} \triangleq \text{diag}[A_1, \ldots, A_K]$, $\mathbf{S} \triangleq [\mathbf{s}_1, \ldots, \mathbf{s}_K]$, and $\mathbf{b}[n] \triangleq [b_1[n], \ldots, b_K[n]]^T$. The vector $\mathbf{b}[n]$ is also known as the multiuser symbol vector.

$$\mathbf{r}[n] = \mathbf{SAb}[n] + \eta[n] \tag{2}$$

The optimum receiver would choose the symbol $\hat{\mathbf{b}}[n]$ such that

$$\hat{\mathbf{b}}[n] = \arg \max_{\hat{\mathbf{b}}[n] \in \mathcal{A}^K} \exp\{-\frac{|\mathbf{r}[n] - \mathbf{SAb}[n]|^2}{\sigma^2}\} \tag{3}$$

Note that the complexity of this method increases exponentially with the number of users K. Lower-complexity sub-optimal algorithms are discussed below.

Linear detectors apply linear processing to the received vector $\mathbf{r}[n]$.

$$z = \mathbf{w}^H \mathbf{r}[n] \tag{4}$$

Then, z is compared to some threshold and a decision about the symbol is made. With these methods, the estimate for the whole multiuser symbol need not be computed [62, 83, 95].

1.1.1 Decorrelating Detector

The decorrelating detector is a zero-forcing solution. When applied to the received vector, it results in zero multiple-access interference (MAI). Let $\mathbf{R} \triangleq \mathbf{S}^H\mathbf{S}$. The decorrelating detector for user 1 is

$$\mathbf{w}_1 = \mathbf{S}\mathbf{R}^{-1}\mathbf{e}_1 \tag{5}$$

The major drawback of this detector, like similar zero-forcing solutions, is noise enhancement [83].

1.1.2 MMSE Detector

The MMSE detector minimizes the total effect of the MAI and the noise.

$$\mathbf{w}_1 = \arg \max_{\mathbf{w}_1 \in \mathcal{C}^N} E\{\|\mathbf{w}^H\mathbf{r}[n] - A_1 b_1[n]\|^2\} \tag{6}$$

For user 1 the detector is

$$\mathbf{w}_1 = \mathbf{S}(\mathbf{R} + \sigma^2|\mathbf{A}|^{-2})^{-1}\mathbf{e}_1 \tag{7}$$

Note that the receiver must know the received power of each user's signal [62, 83].

1.2 Nonlinear Detectors

Decision-driven detectors are nonlinear detectors. They detect other interfering users' symbols, recreate their contributions to the received signal, and then subtract them from the received signal, yielding a better estimate for the desired user's transmitted symbol. Typically this processing is performed in multiple stages. Based on how the interferers are subtracted, there are a few subcategories of detectors [83].

1.2.1 Successive Interference Cancelation (SIC)

This method is also called stripping or successive decoding. The algorithm detects the most powerful interferer and subtracts it from the received signal. Then, based on the received signal with one less interferer, the next strongest interferer is detected and subtracted out.

The process continues until all the interferers are canceled. Finally, the desired user's data symbol is estimated [83].

1.2.2 Parallel Interference Cancelation (PIC)

This algorithm is similar to SIC, except that all the interferers' signals are detected simultaneously, based on the received signal. Their contributions are subtracted, and the desired user's data symbol is estimated. This algorithm can be implemented in parallel. Additionally, it performs better than SIC in cases with equal power interferers such as systems with strict power control, because incorrect interferer estimates in SIC may lead to incorrect symbol estimates of the other interferers. In the PIC method, interferer estimates are independent of each other and depend on the received vector only [60, 83].

1.2.3 Multistage Detection

These types of detectors are an improved version of PIC detectors. They perform matched filter detection for each user in the system in parallel. Then, based on the cross-correlation properties of the codes, their estimates are corrected. The algorithm at the end of the second stage produces K soft estimates with the interferer effects subtracted. Additional stages that refine the estimates even further may be added to improve the performance [62, 83].

1.2.4 Decision-Feedback (DF) Detection

DF detectors are based on the Cholesky decomposition of the cross-correlation matrix $\mathbf{R} = \mathbf{F}^T\mathbf{F}$. The matrix $\mathbf{F}^{-T}$ is used to "prewhiten" the received signal. Note that the first element in $\mathbf{F}^{-T}\mathbf{r}$ is the estimate of the first user data symbol with all the interference subtracted. The second user's symbol estimate contains interference from user 1, and so on. To cancel this interference, the effect of the first user is subtracted from the rest of the estimates, yielding an interference-free estimate of the second user's signal. We continue the process until the last user.

$$\hat{\mathbf{b}} = sgn(\mathbf{F}^{-T}\mathbf{r} - \text{diag}\{\mathbf{F}\}\mathbf{A}\hat{\mathbf{b}}) \tag{8}$$

Note that this method ignores the effect of noise and, like the decorrelating detector, suffers from noise enhancement. An MMSE version can be formulated by the Cholesky factorization

of $\mathbf{R} + \sigma^2\mathbf{A}^{-2} = \mathbf{F}^T\mathbf{F}$ and application of the above steps [83].

1.3 Iterative (Turbo) Receiver Architectures

Recently, turbo processing techniques have been developed quickly since the initial discovery of turbo codes [7, 8]. The turbo principle can be applied not only to channel decoders, but also to a wide variety of combinations of detectors, decoders, equalizers, multiuser detectors, coded modulators, joint source/channel coders, etc. (Figure 1) [95].

A simple communications system consists of a collection of cascaded system blocks. For example, let's consider a receiver consistsing of a symbol detector and a decoder (Figure 2). In a conventional system, the detector makes a hard decision about the symbol based on the received signal. Then,its decision is passed to the decoder that decides what the transmitted data bits were. This solution works, but there is significant information loss when the information about a symbol is truncated to a hard decision. If the confidence level of the detector is passed along with its symbol decision, approximately 2dB of performance is gained [60,62,95]. However this performance is still far from optimal, because earlier stages are not getting any of the information gleaned by the later stages in the chain (Figure 2) [95].

The optimal solution is a maximum likelihood sequence estimation technique, requiring the construction and evaluation of a super-trellis that includes the channel and code effects. This way, the estimation process considers the joint effects of both the channel and the coder. Although this approach is optimal, it is computationally prohibitive.

Iterative processing methods provide an alternative to pass the information from later stages to earlier stages. For iterative processing to work, the individual sub-blocks must produce maximum a posteriori (MAP) estimates of the quantities that they estimate. Namely, both the detector and the decoder must produce MAP estimates of the transmitted bits.

1.3.1 Iterative Detector and Decoder Components

Let us take a closer look at the detector. At each iteration of the loop in Figure 2, the detector makes a decision about the coded bits $c[n]$ by considering the received signal $r[n]$, a priori information about the coded bits from previous iteration, and using knowledge of

the system structure, which includes channel structure, modulation type, noise statistics, etc. The superscript ^{j} indicates the jth iteration of the turbo processing algorithm. The system blocks indicated by π and π^-1 are called interleaver and deinterleaver respectively. Their function purpose and function will be described later. By applying Bayes's rule we get the following factorization:

$$p^j(c[n] = 1|r[n], \text{system structure}) = \frac{p(r[n]|c[n] = 1)p^{j-1}(c[n] = 1)}{p(r[n])} \tag{9}$$

$$p^j(c[n] = 0|r[n], \text{system structure}) = \frac{p(r[n]|c[n] = 0)p^{j-1}(c[n] = 0)}{p(r[n])} \tag{10}$$

Instead of posterior distributions, usually log-likelihood ratios (LLRs) are used.

$$\Lambda_1^j(c[n]) = \log \frac{p^j(c[n] = 1|r[n])}{p^j(c[n] = 0|r[n])} = \underbrace{\log \frac{p(r[n]|c[n] = 1)}{p(r[n]|c[n] = 0)}}_{\lambda_1^j(c[n])} + \underbrace{\log \frac{p^{j-1}(c[n] = 1)}{p^{j-1}(c[n] = 0)}}_{\lambda_2^{j-1}(c[n])} \tag{11}$$

The quantity $\lambda_1^j(c[n])$ is called the extrinsic information produced by the detector, which is the information about the coded bit extracted from the received signal, and the a priori information about the other coded bits, but not from the a priori probability of $c[n]$. The quantity $\lambda_2^{j-1}(c[n])$ is the a priori LLR of $c[n]$. The extrinsic information $\lambda_1^j(c[n])$ is sent to the channel decoder, which uses it as a priori information.

The channel decoder uses the information passed by the detector and information about the channel structure to calculate an a posteriori LLR.

$$\begin{aligned}\Lambda_2^j(c[n]) &= \log \frac{p^j(c[n] = 1|\lambda_1^j(c[n]), \text{code structure})}{p^j(c[n] = 0|\lambda_1^j(c[n]), \text{code structure})} \\ &= \underbrace{\log \frac{p(\lambda_1^j(c[n])|c[n] = 1, \text{code structure})}{p(\lambda_1^j(c[n])|c[n] = 0, \text{code structure})}}_{\lambda_2^j(c[n])} + \underbrace{\log \frac{p^j(c[n] = 1)}{p^j(c[n] = 0)}}_{\lambda_1^j(c[n])}\end{aligned} \tag{12}$$

The likelihood here again is expressed as a sum of the extrinsic likelihood $\lambda_2^j(c[n])$ gleaned from the input, the code structure and all coded bits except $c[n]$, and the a priori

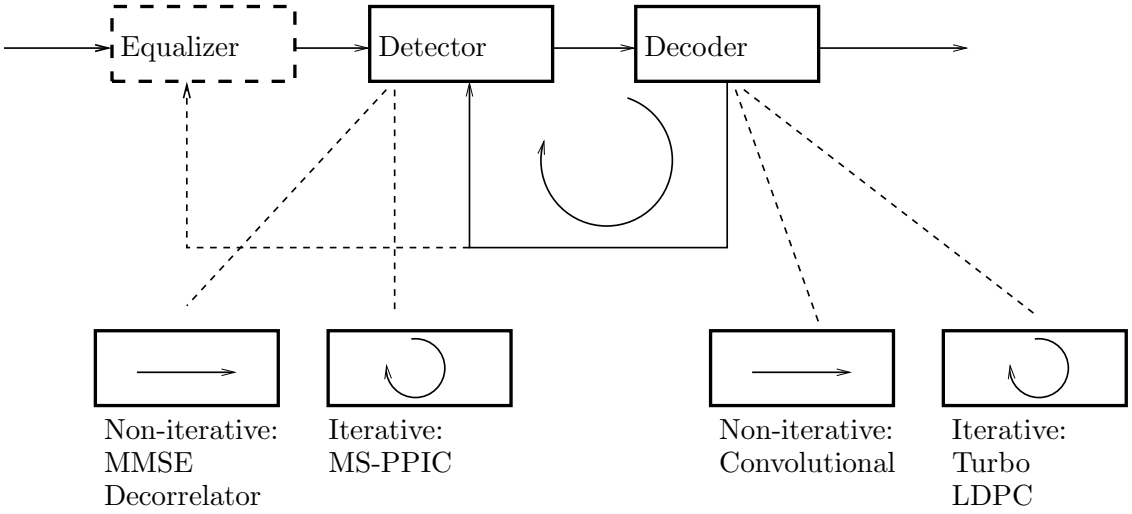


Figure 1: A typical turbo processing receiver architecture

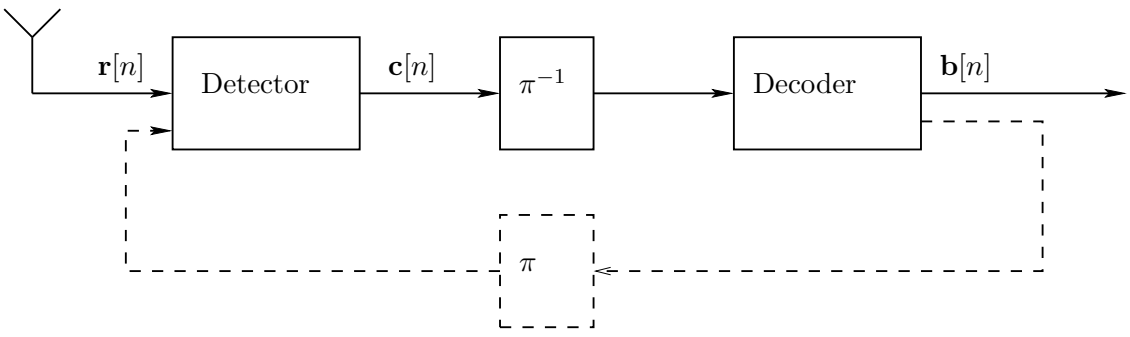


Figure 2: A simple receiver that allows turbo processing.

likelihood $\lambda_1^j(c[n])$. The extrinsic information $\lambda_2^j(c[n])$ is fed back to the first block as a priori information about the coded bits.

It is important to note that the above equations hold only if the inputs to the individual sub-blocks are independent. Obviously, the sequence of coded bits is not independent, because the parity bits are generated from the data bits and, hence, there is some correlation among them. To remedy this, a device that shuffles the bits is inserted to make the bit sequence appear random. The block on which it operates must be large (approximately on the order of 1000 bits or more.) This device is called an interleaver. For simplicity here we employ a random interleaver. Another advantage of the interleaver is that it disperses burst errors evenly throughout the frame. Also, bits that happen to be transmitted during a long lasting fade of the channel, get shuffled and more confident decisions about the new neighbors helps reconstruct the original.

Remarkably, after a few iterations, the decisions become refined, and the estimates become significantly confident. The overall performance of the receiver approaches the performance of the full complexity MLSE receiver.

The individual components in a turbo receiver can be of any type as long as they can produce soft information (Figure 1), and the structure of the detector can vary significantly depending on the type of the communications system. It can be a multiuser-detector, MIMO BLAST type detector, single user detector/equalizer, etc. Poor and Wang, for example, develop an iterative MMSE receiver, that after surprisingly few iterations, approaches the single-user receiver performance [89, 93]. Their detector utilizes a priori symbol estimates from the outer decoder and subtracts them from the received signal. An MMSE processing is applied to the residual signal to further improve the estimate.

The decoder structure can also vary significantly depending on the channel code used. In fact, any kind of code can be used as is seen from Figure 1. For example, the code can be a convolutional code, LDPC code, or even a cascade of a turbo decoder itself. In this work we utilize a simple convolutional coder, but a different code can also be chosen. In the following section the structure of a MAP decoder is described.

1.3.2 MAP Channel Decoder

The MAP decoder takes an a priori LLR as an input and produces an a posteriori LLR based on knowledge of the structure of the code and the code constraints. The algorithm decodes the coded bit sequence by constructing a trellis and by evaluating the probabilities of state transitions in the trellis.

Consider a binary convolutional code of rate k_0/n_0, with constraint length of $k_0 L$. At each time epoch n, assume that the input to the encoder is the information bit sequence

$$\mathbf{b}_n = [b^1[n], \ldots, b^{k_0}[n]]^T, \tag{13}$$

and the output of n_0 coded bit sequence is

$$\mathbf{c}_n = [c^1[n], \ldots, c^{n_0}[n]]^T. \tag{14}$$

At each time epoch n the state of the trellis can be represented by the memory of the code.

$$S_n = [s_n^1, \ldots, s_n^{k_0(L-1)}]^T = [\mathbf{b}_{n-1}^T, \ldots, \mathbf{b}_{n-L+1}^T]^T \tag{15}$$

The dynamics of the convolutional code are fully represented by a trellis describing the state transitions in two consecutive time epochs. Let us denote the input information bit sequence that causes a transition from state $S_{n-1} = s'$, to state $S_n = s$ by $\mathbf{b}(s', s)$. Similarly, let us denote the output coded bit sequence that results from the same transition with $\mathbf{c}(s', s)$. Let us also denote a path of the trellis between time n_1 and $n_2 > n_1$ by the states it traverses as

$$\mathcal{L}_{n_1}^{n_2} = [S_{n_1}^T, \ldots, S_{n_2}^T]^T \tag{16}$$

If a systematic code is used, the pair $(s', \mathbf{c}_n)$ fully describes the state transition (s', s).

Let us assume that the encoding starts from a state of all zeros at $n = 0$ and ends with a state of all zeros at $n = N + L$, where N is the length of the frame. The sequence of information bits is denoted by $\{\mathbf{b}_n\}_{n=0}^N$.

We compute the probability of the state transition $s' \to s$, given the a priori probability of the code bits $\mathbf{c}_n$ and the code structure. It is the sum of the probabilities of all paths from beginning to end, that have the state transition $(s' \to s)$ at time n.

$$P(S_{n-1}=s', S_n=s) = \sum_{\mathcal{L}_0^{N+L}:S_{n-1}=s',S_n=s} P(\mathcal{L}_0^{N+L})$$
$$= \underbrace{\sum_{\mathcal{L}_0^{n}:S_{n-1}=s'} P(\mathcal{L}_0^{n})}_{\alpha_{n-1}(s')} P(\mathbf{c}_n=\mathbf{c}(s',s)) \underbrace{\sum_{\mathcal{L}_{n+1}^{N+L}:S_n=s} P(\mathcal{L}_{n+1}^{N+L})}_{\beta_n(s)} \tag{17}$$

In equation (17), $\alpha_{n-1}(s')$ is the total probability of all path segments from the beginning to $n-1$ that end at state s', and $\beta_n(s)$ is the total probability of all path segments from n to the end that start with state s. These quantities are called forward and backward recursions, respectively, and can be computed recursively as follows:

$$\alpha_n(s) = \sum_{s'} \alpha_{n-1}(s') P(\mathbf{c}_n=\mathbf{c}(s',s)) \qquad , n=1,2,\ldots,N+L \tag{18}$$

$$\beta_n(s) = \sum_{s'} \beta_{n+1}(s') P(\mathbf{c}_n=\mathbf{c}(s,s')) \qquad , n=N+L-1,N+L-2,\ldots,0 \tag{19}$$

The starting and ending boundary conditions are $\alpha_0(\mathbf{0})=1, \alpha_0(s)=0$ for $s\neq\mathbf{0}$ and $\beta_{N+L}(\mathbf{0})=1, \beta_{N+L}(s)=0$ for $s\neq\mathbf{0}$, because the trellis starts and terminates at state $\mathbf{0}$.

Knowing the probability of the state transitions, we can calculate the a posteriori LLRs for the coded bits. Assuming ideal interleaving, the joint distribution of coded bits factorizes as

$$P(\mathbf{c}_n=\mathbf{c}(s',s)) = \prod_{i=1}^{n_0} P(c^i[n]=c^i(s',s)). \tag{20}$$

The a posteriori LLR also factorizes as

$$\begin{aligned}
\Lambda_2(c^j[n]) &= \log \frac{p(c^j[n]=1|P(\mathbf{c}_n),\ \text{code structure})}{p(c^j[n]=0|P(\mathbf{c}_n),\ \text{code structure})} \\
&= \log \frac{\sum_{(s',s)\in S_j^+} \alpha_{n-1}(s')\beta_n(s) \prod_{i=1}^{n_0} P(c^i[n]=c^i(s',s))}{\sum_{(s',s)\in S_j^-} \alpha_{n-1}(s')\beta_n(s) \prod_{i=1}^{n_0} P(c^i[n]=c^i(s',s))} \\
&= \log \frac{\sum_{(s',s)\in S_j^+} \alpha_{n-1}(s')\beta_n(s) \prod_{i\neq j}^{n_0} P(c^i[n]=c^i(s',s))}{\sum_{(s',s)\in S_j^-} \alpha_{n-1}(s')\beta_n(s) \prod_{i\neq j}^{n_0} P(c^i[n]=c^i(s',s))} + \log \frac{P(c^j[n]=+1)}{P(c^j[n]=-1)},
\end{aligned} \tag{21}$$

where the first part of the sum is the extrinsic LLR, and the second part is the a priori LLR.

1.4 Blind Multiuser Detectors

Multiuser detection techniques can substantially increase the capacity of CDMA systems. Considerable recent attention has been focused on the problem of adaptive multiuser detection [38, 39]. Woodward investigated a range of adaptive multiuser detectors such as the adaptive decorrelating detector, the zero-forcing detector, the linear MMSE detector and the MMSE-Block decision Feedback (MMSE-BDF) detector [99]. However, all these adaptive detectors considered require training sequences, which significantly reduces the achievable effective throughput and the resulting spectrum efficiency. To circumvent this problem, Honig proposed the minimum-output-energy (MOE) based blind adaptive detector, which only requires the prior knowledge of the signature waveform and the timing of the reference user, but does not invoke any training sequences. Schodorf et. al proposed a method that reduces the residual power in MOE methods by applying array processing techniques [65].

Subspace methods were first introduced by Wang and Poor for single-carrier DS-CDMA systems communication over synchronous AWGN channels [91]. Later they extended their approach to dispersive asynchronous CDMA environments, where a subspace based channel estimation approach was proposed [90]. Similar approaches were applied to uplink DS-CDMA systems, which exploited the prior knowledge of all known signature waveforms, rather than the one of the desired user [40, 88].

A thorough discussion on existing blind multiuser detection approaches can be found in [95]. Bayesian multiuser detection is a recent research area. The contributions of this thesis are in this area and will be discussed thoroughly in the subsequent sections.

1.5 Blind Equalization

Many practical channels in communications present intersymbol interference (ISI). Blind equalization algorithms attempt to determine the transmitted symbol sequence in the presence of ISI without prior knowledge of the channel state information. The earliest blind equalization algorithm has been proposed by Sato [63]. The algorithm was generalized by Benveniste et al. [5] and Godard [30]. They defined classes of algorithms called BGR and Godard, respectively. The constant Modulus Algorithm (CMA), which is one member of the Godard family, has been extensively implemented because of its simplicity [80]. On the other hand CMA algorithms may converge to undesired local minima [19]. Furthermore the convergence is extremely slow compared to training based equalizer algorithms. These algorithms are based on steepest gradient approaches. The algorithms differ mainly in the cross-correlations which are computed between the equalizer signal output and some non-linear transformation of it. Bellini generalized these algorithms to the Bussgang algorithm [4]. These algorithms were further generalized into the Multimodulus algorithms [100, 101]. Some more recent work exists that is also based on second order statistics, but involves subspace methods as well [78, 79].

Second-order statistics of the received signal are known to provide information on the magnitude of the channel characteristics, but not the phase. Using cyclostationarity property of the received signal allows recovery of the phase information [79].

It is also possible to obtain an estimate of the channel impulse response by using higher order statistical methods. In particular, discrete linear time invariant systems can be obtained explicitly from cumulants of the received signal [10, 28]. Such methods were also proposed by [26, 36, 82]. Analysis and simulations have shown that convergence time for these high-order-statistics-based algorithms is much too long for mobile communication channels. The methods that are based on cyclostationarity properties of the received signal exhibit faster convergence [79].

Another type of blind equalizers is the maximum likelihood (ML) type equalizers. These equalizers can be derived in a systematic way. In addition, their performance is optimal because they approximate the minimum variance unbiased estimators. Asymptotically, ML

estimators approach the Cramér–Rao bound (CRB), which is the lower bound on variance for all unbiased estimators. Although very complex, these methods can be made very effective by subspace methods or other suboptimal approaches as initialization procedures [61].

The ML approaches can be classified into two groups: deterministic ML (DML) and statistical ML (SML). The DML approach turns the estimation problem into a least-squares problem. The formulation of the problem enables separation of the joint symbol and channel estimation. There are a number of methods that tackle this problem such as IQML [9] and TSML [41, 42]. Slock and Papadias pointed out the connection of these algorithms with DML [69, 70]

Seshadri proposed a ML based equalization method that utilized the Viterbi algorithm (VA) to obtain the optimal estimate [66]. A similar algorithm was also proposed by Ghosh et. al. [27]. In their approaches, the channel estimation problem was turned into an LS problem, whereas the symbol estimation problem remained ML, and it was solved using the VA. In another approach Seshadri used generalized VA (GVA) [67]. He assigned a set of channel estimates to each symbol state and expanded the trellis. The correct channel was estimated by eliminating the sequences and keeping the survivors.

Another statistical method is proposed by Iltis et. al. [44, 45]. Their algorithm turns the problem into a Kalman filtering problem using the EKF. These are main approaches that perform blind symbol estimation. There are many publications in this area and one may find many variations of an algorithm. A very thorough literature survey can be found in [20, 29].

Bayesian approaches for the blind deconvolution problem and possible solutions based on Monte Carlo methods are thoroughly discussed in the subsequent sections of this thesis.

1.6 Blind MC Receivers

A sequential Monte Carlo (SMC) framework has been developed and a wide range of applications have been discussed in [50]. This Bayesian technique is primarily utilized in target tracking algorithms [12]; however, other applications also exist. Blind receivers are one

application of this technique in the telecommunications field, and an adaptive scheme has been proposed in [17].

With SMC methods one tries to estimate a set of parameters based on an observation that is a function of these parameters. In the communications area, the parameters are the channel information $h(\cdot)$ and the transmitted symbols $\mathbf{s}$. The observation is the received signal $\mathbf{r}$.

$$\mathbf{r} = h(\mathbf{s}) + \mathbf{n} \tag{22}$$

Simply, the goal is to obtain an estimate for the parameters given the observations. One approach is to process the received signals with a function that reverses the effect of $h(\cdot)$. Depending on how the noise is treated, the estimator takes the form of a zero forcing (ZF) or a minimum mean squared error (MMSE) estimator.

The Bayesian framework treats the problem differently. It considers the parameters as random quantities that are statistically related to the observation. Instead of estimating the parameters directly, the probability density function of the parameters is obtained:

$$p(\mathbf{s}, h(\cdot)|\mathbf{r}) : \text{ joint pdf of } \mathbf{s}, \text{and } h(\cdot) \text{ given } \mathbf{r}. \tag{23}$$

Once the pdf of a random quantity is evaluated, various (such as mean (MMSE), mode (MAP), median, and higher order) estimates of the parameter could be obtained by evaluating Monte Carlo integrals.

Evaluating the posterior density may be impossible. Therefore we resort to distribution functions that are proportional to the posterior by a scalar scaling factor. The Bayes' rule can be applied to the above distribution:

$$p(\mathbf{s}, h(\cdot)|\mathbf{r}) \propto p(\mathbf{r}|\mathbf{s}, h(\cdot))p(\mathbf{s})p(h(\cdot)), \tag{24}$$

where $p(\mathbf{s}, h(\cdot)|\mathbf{r})$ is called the posterior, $p(\mathbf{r}|\mathbf{s}, h(\cdot))$ is called the likelihood, $p(\mathbf{s})$ the prior distribution of the data symbols, and $p(h(\cdot))$ the prior distribution of the channel. The transmitted symbols and the communications channel are physically independent entities; therefore, their joint distribution factorizes naturally.

From the above equations, it is essential that the likelihood distribution be known. If it can not be expressed analytically, a numerical representation can be evaluated. Namely, the continuous parameter space is discretized and the likelihood function is evaluated at each sampling point. In communications theory problems, the channel is a continuous random process, and it must be discretized. However, the symbol space is usually discrete already, and either all points can be evaluated or some other smart sampling method can be utilized to evaluate the likelihood function efficiently. In the next section we describe how Monte Carlo methods are used to obtain a sampled representation of the density of interest.

1.7 Monte Carlo Methods

Monte Carlo methods provide a general statistical approach for simulating multivariate distributions by generating an efficient discretized representation of the needed posterior density.

Monte Carlo Markov chain (MCMC) methods construct Markov chains whose stationary probability approaches the posterior distribution that we want to simulate. Generally, sampling sampling from the distribution of interest is difficult. As an alternative, these algorithms update the Markov chain, based on the outcome of a sample from a separate distribution called the proposal transition and the likelihood evaluated at that particular point. The stationary distribution of the drawn samples converges to the desired distribution. Dynamic processes can not be simulated with these methods. Only stationary channels can be simulated.

Sequential Monte Carlo (SMC) methods, on the other hand, provide a Bayesian framework that models time varying channels, and produces a numerical representation of the posterior distribution. This is done by sampling from a proposal distribution and weighting the drawn samples with proper weights. In the literature, this method is also called sequential importance sampling.

In the next subsections, we will describe several MCMC methods (including the acceptance-rejection algorithm, Metropolis-Hastings algorithm, and the Gibbs sampler), and the SMC method.

1.7.1 Acceptance-Rejection Algorithm

Let us assume that we have a distribution $\pi(x) \propto \alpha(x)$ that is easy to evaluate at discrete points, but from which samples are difficult to draw, and our goal is to obtain realizations of $\pi(x)$. In such cases we use the acceptance-rejection algorithm. According to the algorithm, a sample is drawn from another distribution $\beta(x)$ and, based on the outcome, a decision is made on whether to keep the sample or reject it. This algorithm is summarized in Table 1.

The distribution $\beta(x)$ must be chosen such that $\alpha(x) \leq c\beta(x)$ for some constant c. The convergence rate of the algorithm depends on the choice of the proposal distribution $\beta(x)$. The convergence rate is slow, because the proposed transition to the next state x' could be anywhere in the parameter space.

Table 1: Acceptance-Rejection Sampling

i. Generate a sample x' from $\beta(x')$.
ii. Generate a sample $u \sim \mathcal{U}(0,1)$, the uniform distribution on $(0,1)$.
iii. if $u \leq \frac{\alpha(x')}{c\beta(x')}$, then $x = x'$(accept). Otherwise go to step i.

1.7.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings (M-H) algorithm is similar to the Acceptance-Rejection (A-R) algorithm described above. It differs in the way the Markov chain is evolved. In the A-R algorithm when a new state is proposed, it can be anything no matter what the previous state is. With the M-H algorithm, on the other hand, there is a proposal transition density $q(x \rightarrow x')$, which proposes a new sample conditioned on the previous sample. The algorithm is summarized in Table 2.

The perturbation function $q(x \rightarrow x')$ is nearly arbitrary and is completely specified by the user. Metropolis et al. restricted their choice to symmetric functions only, i.e. $q(x \rightarrow x') = q(x' \rightarrow x)$, which means that there is no trend bias in the choices of next states [52, 53].

Table 2: Metropolis-Hastings Algorithm

---

i. Generate a sample x' from the perturbation function $q(x \to x')$.

ii. Compute the Metropolis ratio

$$r(x, x') = \min\left\{1, \frac{\pi(y)q(x', x)}{\pi(x)q(x, x')}\right\}.$$

iii. Draw a sample $u \sim \mathcal{U}(0, 1)$, from the uniform distribution on $(0, 1)$.

iv. if $u \leq r(x, x')$, then $x = x'$ (accept), otherwise keep the old state x as the new one

---

Hastings generalized the choice of $q(x \to x')$ to those that satisfy the property $q(x \to x') > 0$, if and only if $q(x' \to x) > 0$ [35]. Such a transition function satisfies the balance condition. Namely, when both x and x' are from the distribution $\pi(\cdot)$ the following equation is specified:

$$\pi(x)q(x \to x') = \pi(x')q(x' \to x) \tag{25}$$

Equation (25) implies that the Markov chain is reversible, and its invariant distribution is $\pi(\cdot)$.

Let's assume that the proposed transition x' violates the balance equation:

$$\pi(x)q(x \to x') > \pi(x')q(x' \to x) \tag{26}$$

We need to balance the equation with the metropolis ratio $r(x, x')$:

$$\pi(x)q(x \to x')r(x, x') = \pi(x')q(x' \to x), \tag{27}$$

which means that the moves from x to x' are accepted with probability

$$r(x, x') = \min\left\{1, \frac{\pi(x')q(x' \to x)}{\pi(x)q(x \to x')}\right\}. \tag{28}$$

Note that the algorithm keeps the old state as the new state, if the the proposed state is rejected. If the transition distribution is symmetric, the chain evolves with updates that increase the likelihood of the state.

1.7.3 Gibbs Sampler

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_d]$, where $\mathbf{x}_i$ could be a scalar or a vector. Suppose the goal is to obtain samples from a joint distribution $p(\mathbf{X})$. Let's define $\mathbf{X}_{[-i]} \triangleq \mathbf{X} \backslash \mathbf{x}_i$, which is the vector $\mathbf{X}$ with the ith element deleted. The Gibbs sampler, chooses a coordinate i and draws a new sample $\mathbf{x}'_i$ from $p(\mathbf{x}_i|\mathbf{X}_{[-i]})$. There are a few variations of the algorithm, which are summarized in Table 3 and Table 4.

Table 3: Random-Scan Gibbs Sampler

Suppose the current sample is $\mathbf{X}^{(n)} = [\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_d^{(n)}]$. Then:

- Select i randomly from the index set $\{1, \ldots, d\}$.
- Draw $\mathbf{x}'^{(n+1)}_i$ from the conditional distribution $p(\mathbf{x}_i|\mathbf{X}^{(n)}_{[-i]})$, and let $\mathbf{X}^{(n+1)}_{[-i]} = \mathbf{X}^{(n)}_{[-i]}$ (accept the sample, and keep the rest the same)

Note that each conditional update does not modify the joint distribution $p(\cdot)$. Suppose that currently, $\mathbf{X}^{(\mathbf{n})} \sim p(\cdot)$. Then,

$$p(\mathbf{x}^{(n+1)}|\mathbf{X}^{(n)}_{[-i]}).p(\mathbf{X}^{(n)}_{[-i]}) = p(\mathbf{x}^{(n+1)}, \mathbf{X}^{(n)}_{[-i]}), \tag{29}$$

which means that the joint distribution is not changed after an update.

Table 4: Systematic-Scan Gibbs Sampler

Suppose the current sample is $\mathbf{X}^{(n)} = [\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_d^{(n)}]$.

- For $i = \{1, \ldots, d\}$

  Draw $\mathbf{x}'^{(n+1)}_i$ from the conditional distribution

$$p(\mathbf{x}_i|\mathbf{x}_1^{(n+1)}, \mathbf{x}_2^{(n+1)}, \ldots, \mathbf{x}_{i-1}^{(n+1)}, \mathbf{x}_{i+1}^{(n)}, \ldots, \mathbf{x}_d^{(n)}) \tag{30}$$

The convergence of the Gibbs sampler has been investigated by several authors [13, 47, 64, 73, 98]. It have proved that

- The distribution of $\mathbf{X}^{(n)}$ converges to $p(\mathbf{X})$, as $n \to \infty$
- For any integrable function $f(\cdot)$,

$$\frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{X}^{(n)}\right) \xrightarrow{a.s} \int f(\mathbf{X})p(\mathbf{X})(d)\mathbf{X}, \text{ as } N \to \infty \tag{31}$$

Although the algorithm has been proven to converge, it converges slowly. An initial period of n_0 burn-in samples are required to let the algorithm reach equilibrium. These initial samples must always be discarded. The convergence is detected with heuristic methods. For example one can look at the discrepancy between the sampled and desired distributions and stop the algorithm when the difference is small enough to satisfy the user [73].

Also, note that because the draws are from conditional distributions, the samples are not independent. One solution for reducing the correlation between samples is to consider every rth sample. When r is large enough, the samples are almost independent [47]. Another way to reduce correlation is to group variables together and draw them jointly. Thus, reducing correlation and increasing convergence rate [47].

1.7.4 Sequential Monte Carlo

Monte Carlo Markov chain (MCMC) methods cannot be used to track parameters of a dynamic system. Every time the system parameters evolve, their distribution also changes. After such a change, the algorithm must be allowed a burn-in period to let it converge, and a number of samples in the burn-in must be discarded. Moreover, previous state estimates cannot be used in the estimate of the current state and must also be discarded, because the algorithm does not allow for dynamic updates. All of these result in wasted computational resources.

In a dynamical system, the parameters are expected to change, but not too quickly. Hence, when the parameter distribution is being calculated, it can be assumed that the new state of the system will be correlated with the previous state (using the state update model). Hence, results of previous computations should not be discarded, but kept to improve the overall estimate.

Consider the following state-space model:

$$\begin{aligned} x_n &\sim q(x_n|x_{n-1}) \\ y_n &\sim f_n(y_n|x_n) \end{aligned} \tag{32}$$

The function $q(\cdot)$ is called the state density function and $f_n(\cdot)$ is the observation density function. Let us denote with boldface $\mathbf{x}_{0:n} = \{x_0, \ldots, x_n\}$ the sequence of data samples. Then, the target posterior can be factorized as

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) &= \frac{p(y_n|\mathbf{x}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1})}{p(y_n|\mathbf{y}_{0:n-1})} \\ &= \frac{p(y_n|\mathbf{x}_{0:n}, \mathbf{y}_{0:n-1})p(x_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})}{p(y_n|\mathbf{y}_{0:n-1})} p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \end{aligned} \tag{33}$$

Using the state-space model in (32) and Markov assumptions, we get

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = f(y_n|x_n)q(x_n|x_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \tag{34}$$

Given the approximation for the posterior distribution at time $n-1$ and the most recent observations, we want to obtain the posterior distribution at time n. The SMC algorithm, as we had mentioned before, generates an approximation to the posterior density by sampling the continuous space at discrete points. The distribution is sampled at these points, chosen by a proposal function $g(\cdot)$, and weighted so as to appear as if they are samples of the desired posterior distribution.

Let us introduce some terminology. A random variable X drawn from distribution $g(\cdot)$ is said to be properly weighted by a weighing function $w(X)$ with respect to the distribution $\pi(\cdot)$ if for any integrable function $h(\cdot)$

$$E_g\{h(X)w(X)\} = E_\pi\{h(X)\}. \tag{35}$$

Since our goal is approximate the posterior distribution, we have a number of discrete draws of samples. Now follows the discretized version of the above definition. A set of random draws and their weights $\{x^{(j)}, w^{(j)}\}, \quad j = 1, 2, \ldots, N_s$ is called properly weighted with respect to π if

$$\lim_{N_s \to \infty} \frac{\sum_j h(x^{(j)})w^{(j)}}{\sum_j w^{(j)}} = E_\pi\{h(X)\} \tag{36}$$

for any integrable function $h(\cdot)$. Clearly the weights must satisfy

$$w_n^{(j)} \propto \frac{p(\mathbf{x}_{0:n}^{(j)}|\mathbf{y}_{0:n})}{g(\mathbf{x}_{0:n}^{(j)}|\mathbf{y}_{0:n})}. \tag{37}$$

If the expression above is factorized, then, using the state-space formulation along with Markov chain assumptions, the following expression is derived:

$$w_n^{(j)} \propto w_{n-1}^{(j)} \frac{f_n(y_n|x_n)q_n(x_n|x_{n-1})}{g(x_n|x_{n-1}, y_n)} \tag{38}$$

The proposal function must have the same support as the posterior. If this is not the case, some parameter values will not be realized at all, and the resulting approximation will deviate significantly. Alternatively some other values will be realized more often than they should, and will be assigned low weights. A large number of weights close to zero will cause a wasting of computational resources. The proposal function must be as close as possible to the posterior. Significant mismatch between the two will deteriorate the estimation performance, both in terms of accuracy and computational efficiency.

The degeneracy problem, characterized by the weights gradually assuming values close to zero is a common problem with all variations of SMC methods [50, 73]. An additional step called resampling is required to keep the efficiency at normal levels [48]. With this step, particles with low weights are discarded and the rest are replicated so that the distribution is efficiently represented. If resampling is done too often, the resulting estimate gets biased. The choice for the number of particles and the resampling frequency form a tradeoff between estimate variance, bias and the computational power requirement [11]. Clearly, a poor choice of a proposal distribution will lead to a faster decay of the weights, which will require more frequent resampling and cause large variance of the estimate.

Let us illustrate the effect of proposal function choice on estimation performance with the following two examples.

In the first example we represent the distribution with its realizations at discrete samples in the state-space domain. Let us assume that the distribution that we would like to represent is the well known Gaussian density with mean μ and covariance matrix $\mathbf{\Sigma}$:

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi|\mathbf{\Sigma}|}} \exp\left\{-\frac{(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}{2}\right\}, \tag{39}$$

where

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{\Sigma} = \begin{bmatrix} 1 & -\sqrt{2} \\ -\sqrt{2} & 2 \end{bmatrix}. \tag{40}$$

We draw $N = 1000$ samples from the posterior distribution by using a square root of the covariance matrix to introduce the desired correlation. The dots in Figure 3 are the sampling points and the ellipse

$$(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) = 4 \tag{41}$$

is displayed to illustrate the correlation property of the drawn samples.

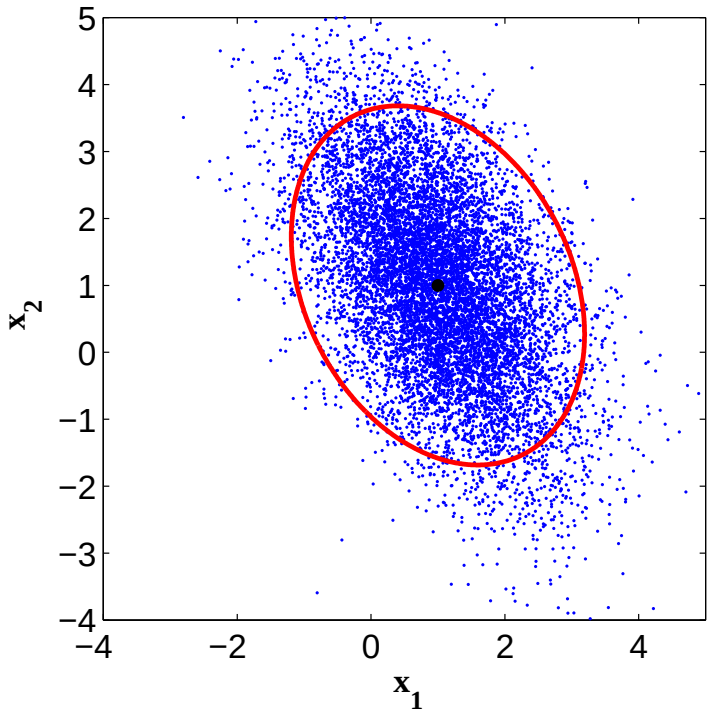


Figure 3: Samples from the posterior distribution when chosen as the proposal distribution

The sample mean and sample covariance of this distribution can be easily calculated:

$$E\{\mathbf{x}\} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}^{(j)} = \begin{bmatrix} 1.0005 \\ 1.001 \end{bmatrix} \tag{42}$$

and

$$E\left\{(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\right\} = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{x}^{(j)} - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\right)\left(\mathbf{x}^{(j)} - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\right)^T$$
$$= \begin{bmatrix} 1.014 & -0.7303 \\ -0.7033 & 1.989 \end{bmatrix} \tag{43}$$

These estimates improve as the sample size N is increased and converge to the ensemble expectations asymptotically.

The next example illustrates how we obtain the same mean and covariance estimates when we have to sample from a different distribution called a proposal distribution. In this particular case, a different distribution might be used because of a lack of computational resources to generate correlated samples. We obtain the estimates by calculating a weighted sum of the particles:

$$E\left\{h(\mathbf{x})\right\} \approx \frac{\sum_{j=1}^{N} h(\mathbf{x}^{(j)})w^{(j)}}{\sum_{j=1}^{N} w^{(j)}} \tag{44}$$

This computation requires that the support of the proposal function must cover the support of the distribution of interest [22]. The weights simply are a ratio of the distributions evaluated at the sampling points (37).

Let us choose a proposal function $g(\mathbf{x}) \sim \mathcal{N}(\mu_g, \mathbf{\Sigma}_g)$. In this example we choose a slightly different mean, to illustrate that the importance sampling method works as long as the support regions match.

$$\mu_g = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \qquad \mathbf{\Sigma}_g = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \tag{45}$$

The black dashed ellipse represents the posterior distribution, and the red ellipse represents the proposal distribution (Figure 4).

$$E\left\{\mathbf{x}\right\} = \frac{1}{N}\sum_{j=1}^{N} w_*^{(j)}\mathbf{x}^{(j)} = \begin{bmatrix} 0.9845 \\ 1.004 \end{bmatrix} \tag{46}$$

and

$$E\left\{(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\right\} = \frac{1}{N}\sum_{j=1}^{N} w_*^{(j)} \left(\mathbf{x}^{(j)} - \frac{1}{N}\sum_{i=1}^{N} w_*^{(j)}\mathbf{x}^{(i)}\right)\left(\mathbf{x}^{(j)} - \frac{1}{N}\sum_{i=1}^{N} w_*^{(j)}\mathbf{x}^{(i)}\right)^T$$
$$= \begin{bmatrix} 0.9536 & -0.6855 \\ -0.6855 & 1.942 \end{bmatrix}, \tag{47}$$

where

$$w_*^{(j)} = \frac{w^{(j)}}{\sum_j w^{(j)}}. \tag{48}$$

Note that although the distribution parameters are mismatched, the estimated mean and covariance are very accurate. Also it has been proven that the estimation variance improves as the particle number increases [11, 22, 50]

1.8 State-of-the-Art

All blind receivers have ambiguity problems [95]. Phase ambiguity arises from the fact that the receiver cannot tell whether the phase changes originate from the transmitted symbol or from the phase of the unknown channel coefficients. Differential coding is one solution to the problem, where information is transmitted on the phase differences of the signal rather than on the absolute phase. In some works, authors use symbol constellations that are not symmetric about the origin, which is another way to avoid ambiguity [14, 49, 50]. Others assume Gaussian channels, with unknown, but real positive only, channel coefficients [93]. This case constrains the source of phase changes to the symbol constellation.

Antenna ambiguity is another type of ambiguity, in which the receiver cannot determine from which antenna the signals are transmitted. Space-time coding solves this problem by introducing orthogonality among the space-time symbols [75, 76]. Some authors spread the signals with distinct spreading codes, a scheme called space-time spreading, which is just another form of orthogonalization [60].

The other type of ambiguity is user ambiguity which is resolved by using distinct spreading or scrambling codes which can be processed by multiuser detectors at the receiver end.

In all of the applications below, some method of eliminating these ambiguities is used. If the ambiguities were not eliminated, the received signal distributions would be multi-modal. Consequently, any of the utilized MC algorithms would exhibit extremely slow convergence. Rather than refining the estimate at each iteration, the algorithms would keep dwelling among the proximities of the ambiguous states and never converge on one of them [11].

One of the early applications of MCMC processing in communications is done by Chen and Li. They propose an application of the Gibbs sampler for blind deconvolution [14]. Liu and Chen also propose a SMC algorithm for the same problem [49]. This applications is very similar to the problem of symbol estimation for frequency selective channels that introduce ISI. In [50] the authors investigated the effects of different resampling techniques and concluded that residual resampling with dynamic decisions on when to resample results in smaller estimate variances.

Wang and Poor designed numerous symbol detectors based on MCMC methods [95]. Initially, they concentrated on multi-user detectors for DS-CDMA systems that transmit on AWGN channels. The transmit power and noise variance are assumed unknown, and Monte Carlo integration is performed to account for their effects. The symbols are drawn and detected with a Gibbs sampler technique using batch processing.

The Gibbs sampler is capable of accounting for any kind of noise, since it can generate samples from any distribution. Using this property, the detection technique can be generalized to non-Gaussian channels, impulsive noise channels [84, 89], and correlated unknown channels [92]. Some recent Markov chain Monte Carlo receiver applications for wireless communicatoins are discussed in [87].

The use of Markov chain Monte Carlo methods is not limited to these special cases only. They can be used to estimate any stationary system parameters. The MCMC method has been adapted to be used in time-invariant ISI channels [85] where the channel coefficients are assumed stationary, and they, along with the noise variance, are estimated using MCMC methods. In [94, 102], this processing has been extended to multicarrier CDMA systems with space-time coding. In this case, the receiver has knowledge of the spreading codes of each, user and it estimates the channel coefficients and noise variance jointly using a Gibbs

sampler receiver. In [103] the work has been generalized to asynchronous CDMA with quasistatic multipath fading, where the asynchronous delay parameter is another variable to be estimated. In all of these applications, the unknown system parameters are stationary and are estimated via MCMC methods.

MCMC methods have a wide range of applications in telecommunications as can be seen in the literature. However, they cannot be applied to dynamic systems. Time varying channels are a good example of a dynamically evolving system. The receivers which are designed for such channels must adapt to the new system state quickly enough to be able to estimate the transmitted data accurately. SMC methods are designed for this purpose specifically. The algorithms track system changes and estimate the system parameters in a recursive manner. Namely, they use the previous estimate, process the observation data, and infer the new estimate.

In telecommunications, an example by Liu and Chen of using SMC methods was given in [49, 50]. They illustrate a blind deconvolution problem, in which the channel parameters are known, and the symbol constellation is asymmetric around the origin of the complex plane to avoid phase ambiguity. SMC methods have been applied to various channels and modulation schemes such as OFDM [104]. They have also been used, along with wavelet characterization of the communications channel, in [33]. MIMO systems have been considered in [32]. Many of these algorithms utilize mixture Kalman filters.

Mixture Kalman filters relieve the computational complexity and improve the convergence rate of SMC algorithms significantly. Mixture Kalman filters were introduced by Chen and Liu in [15]. Many systems can be described by distinct sets of linear equations for the state update and observation. The perturbations in the state and the observation noise are modeled as Gaussian random processes. At a given time, one particular set of equations represents the current observation and the state update. The choice for the right set of state-space equations that describe the observation depends on the realization of some auxiliary random variable. Such systems are called conditional dynamical linear models (CDML), which are conditioned on the realization of the random indicator variable.

The authors tackle the equalization problem by solving the linear part analytically by Rao-Blackwellization (i.e. a Kalman filter), and the indicator random variable part by Monte Carlo processing. This idea not only speeds up the processing by significantly by reducing the number of required particles, but also improves the accuracy via the Rao-Blackwell theorem.

Rao-Blackwellized SMC have been applied to telecommunications by Wang, Liu and Chen in [86]. They introduce an SMC based MIMO multiuser detector. Another application to MIMO has been elaborately discussed in [21, 32]. The authors using QR factorization and convert the observation equation to an upper triangular matrix form. Then, using back-substitution processing, all the symbols are detected. Because of the random nature of the problem, and the artificially introduced sequential dependence, SMC processing is applied. Simple nulling and cancelation is used to detect the symbols and calculate their weights. However, to detect each symbol they use only a part of the observation, which is sub-optimal.

In their alternative algorithm, which Dong et al. call deterministic SMC, a trajectory search is performed [21, 32]. Imagine a ML sequence detector (MLSD), that constructs a trellis with a certain memory length and keeps track of the best paths and their metrics to make decisions about the system state. Their algorithm does similar processing, with the exception that the system memory is truncated and the number of system states is significantly smaller. The amount of truncation is a design parameter and zero truncation corresponds to full MLSD processing. Indeed they report better performance for the deterministic SMC.

Djuric et al. apply SMC processing to frequency selective channels [54]. They improve the accuracy by introducing delay in the channel estimate [54]. The performance is also analyzed when the true channel impulse response length does not match the model. To combat this problem they design an SMC method that has the length of the channel impulse response as a parameter to be estimated. They report significantly better performance than other equalization methods such as linear MMSE and per-survivor processing. They claim that SMC processing approaches the performance of MLSE equalizer.

Huang et al. also design a SMC based MIMO detector [43]. A Cholesky factorization was performed on the observation equation after matched filtering, which not only results in a triangular matrix structure, but also whitens the noise after matched filtering. They exploited this structure and by back-substitution for nulling and canceling, and recovered the transmitted symbols. Instead of considering one conditional posterior at a time, which is sufficient for simple nulling and canceling, they considered M branches with the largest posterior densities, which is the key principle of the M-algorithm [1, 68]. Analogously to Wang et al. they also proposed a stochastic M-algorithm which used SMC processing on the posteriors.

In the early literature, only basic applications of Monte Carlo processing have been applied to the telecommunications field. The methods have been adapted to different system architectures, and various assumptions have been made about the channel models. In more recent publications, authors are more creative [43]. They take advantage of the data structure and channel structure to construct more sophisticated algorithms that achieve higher detection accuracy and lower computational complexity. Instead of blindly sampling the entire state-space, the support region is reduced intelligently. In addition, rather than approximating a single easy to compute posterior, more complex posteriors, based on additional observations, are also considered when making symbol decisions.

1.9 Contributions of the Thesis

The contributions of this thesis can be summarized as follows. In Chapter 2 a multiuser receiver based on MOCS is constructed. MOCS are used because they introduce significantly lower bandwidth expansion compared to standard pseudo-noise codes. The constructed receiver also exploits space-time codes to increase spatial diversity. The properties of the MOCS are used to distinguish between the desired user and the interferers, all of which are using multiple transmit antennas. A high order of diversity is achieved in flat Rayleigh fading channels and the ability to cancel multi-user interference. The performance of the system in the presence of multiple users is shown, both analytically and by simulation, to be comparable to the single user case. We also derive an upper bound for the bit error rate,

which guarantees that the system maintains its diversity order with the MOCSs.

In the next chapter we propose a state-space partitioning method on the symbol space of a blind SMC based equalizer. The receiver is used in a single user communications system, however it can be generalized to multiuser scenarios easily. The main contribution of this work is in the reduction of the complexity of the SMC algorithm. With the proposed temporal partitioning of the symbol space, we draw sub-symbols from each subspace and then we merge the results. We show that the proposed partitioning reduces the computational complexity without any performance loss or additional error propagation effects.

In chapter 4, we apply a similar partitioning approach to MIMO systems. We propose a blind MIMO detector based on SMC methods. The partitioning in this case is spatial, and analagous to vertically layered BLAST systems. Similar techniques could be applied to diagonally and horizontally layered systems, depending on the structure of the space-time code. We compare the performance to a standard SMC algorithm that does not have any embedded partitoning scheme.

In chapter 5, we design a blind multiuser receiver that also incorporates symbol space partitioning. In this case the transmitter is exactly the same as the transmitter used in chapter 2. The symbol space is partitioned based on the conditional independence of the likelihood distributions for each received signal. With this approach the computational complexity of the algorithm is decreased, from being exponential in the constraint length of the MOCS to only quadratic in the same parameter. We analyze the ability of the algorithm to track channel changes and compare the algorithm to the regular SMC algorithm without partitioning.

In chapter 6, we propose a novel method that quantifies the convergence rate of SMC algorithms. With the proposed method we describe how various parameters, such as noise power, channel fading rate, or simply the utilized number of particles, affect the convergence rate. We also show how the convergence of our partitioned algorithms differs from the unpartitioned SMC algorithms.

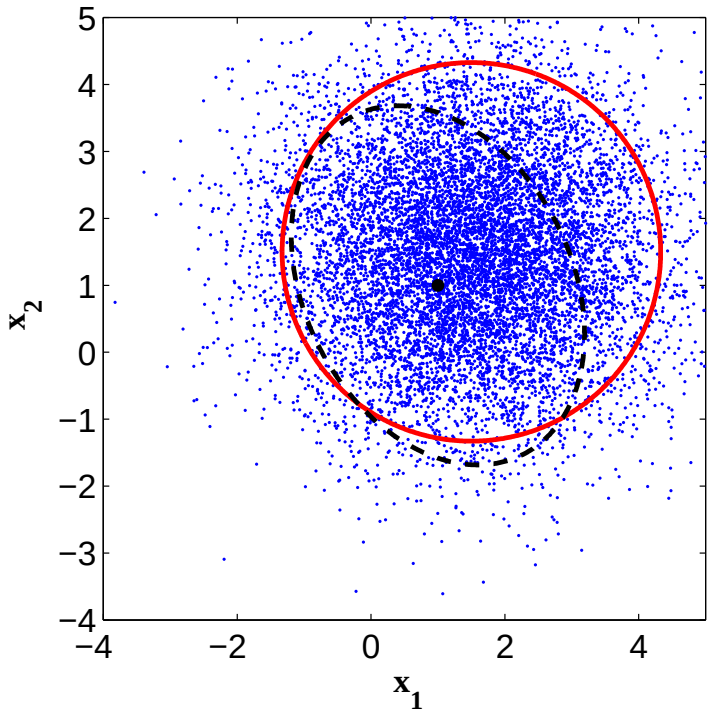


Figure 4: Samples drawn samples from a proposal function requiring less computation

CHAPTER II

AN ITERATIVE MMSE MULTIUSER RECEIVER FOR MUTUALLY ORTHOGONAL COMPLEMENTARY SETS

We propose an iterative multi-user detectoion scheme that exploits the properties of mutually orthogonal (MO) complementary sets and space-time codes for multi-user communications with spatial diversity and minimal bandwidth expansion. The properties of the MO sequences are used to distinguish between the desired user and the interferers, all of which are using multiple transmit antennas. We achieve a high order of diversity in flat Rayleigh fading channels and the ability to cancel multi-user interference. The performance of the system in the presence of multiple users is shown, both analytically and by simulation, to be comparable to the single user case.

2.1 Introduction

Recently, much research has been done on signal processing techniques for transmit diversity over fading channels to take advantage of the large capacity of multi-antenna systems [24]. Space-time (ST) codes that maximize transmit diversity have been discovered and systematic constructions have been found [75]. In particular, the orthogonal structure of ST block codes makes them attractive because maximum likelihood (ML) decoding can be implemented with a linear decoder.

In wireless communication systems, many users share the same communications medium. Direct sequence code division multiple access (DS-CDMA) is one of the techniques that enables sharing the common medium. The maximum number of DS-CDMA users depends on the properties of the pseudo-noise (PN) codes assigned to them. Among the many multiuser detection (MUD) and multiuser access interference (MAI) cancellation algorithms that have been found is the minimum mean square error (MMSE) interference cancellation algorithm. At the receiver end, after matched filtering, MMSE techniques can be applied

to the signal to eliminate interference from other users [55]. Poor et al. proposed a turbo technique to improve the performance of the system [93].

The multi-user detection method that we propose is also an iterative approach. Unlike the single antenna technique discussed in [93], the transmitters in our case have multiple transmit (Tx) antennas. We also utilize mutually orthogonal complementary sets (MOCS) rather than DS-CDMA PN codes, allowing effective multi-user communication and a natural spatial diversity implementation with almost no bandwidth expansion.

2.2 System Description

We consider a convolutionally coded system with K users. Each user k employs a distinct mutually orthogonal complementary mate set of sequences $(A_{km}[n], \quad 1 \leq m \leq M)$ of length p as a means to distinguish itself from the other users. Complementary series were first investigated in [31] and, consequently, their autocorrelation properties have been found to be suitable for the field of communications. They can be used to send data in uncoupled parallel channels and still allow interference cancelation [57]. A more detailed discussion on the construction and properties of MOCS can be found in [81].

Let $\psi_{A_{pm}A_{pm}}$ be the autocorrelation function of the sequence $A_{pm}[n]$, and let $\psi_{A_{pm}A_{pm}}[k]$ denote the kth element of this sequence. The set of sequences $(A_{pm}[n], \quad 1 \leq m \leq M)$ is a complementary set of sequences if and only if:

$$\sum_{m=1}^{M} \psi_{A_{pm}A_{pm}}[k] = 0, \qquad \forall k \neq 0. \tag{49}$$

Let $\psi_{A_{pm}A_{qm}}$ be the cross correlation function of the sequences A_{pm} and A_{qm}. A set of sequences $(A_{qm}[n], \quad 1 \leq m \leq M)$ is a mate of the set $A_{pm}[n]$ if:

- the length of the sequences A_{pm} and A_{qm} are the same for $1 \leq m \leq M$
- the set $(A_{qm}[n], \quad 1 \leq m \leq M)$ is a complementary set
- $\sum_{m=1}^{M} \psi_{A_{pm}A_{qm}}[k] = 0, \qquad \forall k$

Notice that, MO sequences, unlike PN sequences in DS-CDMA, are not orthogonal to each other. Instead, the sum of cross correlations between sequences from two mate sets is

zero. We use these properties as a means to distinguish between the desired user and the interferers. The convolutionally encoded data $c_k[n]$ is interleaved and then encoded with these sequences to produce the signals $s_{km}[n]$. These signals are encoded with a space-time block code and are transmitted from M antennas (Figure 5).

In space-time block code (STBC) systems, the elements of the symbol vector of user k, $\underline{s}_k[n] = [s_{k1}, s_{k2}, \ldots, s_{kM}]^T$, are mapped into a ST code matrix $\mathbf{S}_k$ of size $L \times M$ and transmitted through a flat, block static, Rayleigh fading channel from the M transmit (Tx) antennas. The variable $h_{km}[n]$ denotes the channel coefficient for the kth user's mth Tx antenna at time epoch n. The received vector due to user k, $\underline{r}_k[n]$, is then decoded using the STBC decoder matrix for user k $\mathbf{H}_k$ [74, 75].

$$\mathbf{S}_k = \begin{bmatrix} s_{k1} & s_{k2} & s_{k3} & s_{k4} \\ -s_{k2} & s_{k1} & -s_{k4} & s_{k3} \\ -s_{k3} & s_{k4} & s_{k1} & -s_{k2} \\ -s_{k4} & -s_{k3} & s_{k2} & s_{k1} \\ s_{k1}^* & s_{k2}^* & s_{k3}^* & s_{k4}^* \\ -s_{k2}^* & s_{k1}^* & -s_{k4}^* & s_{k3}^* \\ -s_{k3}^* & s_{k4}^* & s_{k1}^* & -s_{k2}^* \\ -s_{k4}^* & -s_{k3}^* & s_{k2}^* & s_{k1}^* \end{bmatrix}, \underline{h}_k = \begin{bmatrix} h_{k1}[n] \\ h_{k2}[n] \\ \vdots \\ h_{kM}[n] \end{bmatrix} \tag{50}$$

The received signal equation can be written compactly as

$$\underline{r}_k[n] = \mathbf{S}_k \underline{h}_k + \underline{\nu}[n]. \tag{51}$$

The next step is to take the complex conjugate of the second half of the received vector. The resulting vector can be expressed in terms of the decoder matrix $\mathbf{H}_k$.

$$\underline{\tilde{r}}_k[n] \triangleq \begin{bmatrix} r[n] \\ \vdots \\ r[n+M/2-1] \\ r^*[n+M/2] \\ \vdots \\ r^*[n+M-1] \end{bmatrix} = \mathbf{H}_k \underline{s}_k + \begin{bmatrix} \nu[n] \\ \vdots \\ \nu[n+M/2-1] \\ \nu^*[n+M/2] \\ \vdots \\ \nu^*[n+M-1] \end{bmatrix}, \tag{52}$$

where the superscripts * and ^{H} denote complex conjugation and Hermitian of a matrix, respectively, and the STBC decoder matrix has the following form:

$$\mathbf{H}_k = \begin{bmatrix} h_{k1} & h_{k2} & h_{k3} & h_{k4} \\ h_{k2} & -h_{k1} & h_{k4} & -h_{k3} \\ h_{k3} & -h_{k4} & -h_{k1} & h_{k2} \\ h_{k4} & h_{k3} & -h_{k2} & -h_{k1} \\ h_{k1}^* & h_{k2}^* & h_{k3}^* & h_{k4}^* \\ h_{k2}^* & -h_{k1}^* & h_{k4}^* & -h_{k3}^* \\ h_{k3}^* & -h_{k4}^* & -h_{k1}^* & h_{k2}^* \\ h_{k4}^* & h_{k3}^* & -h_{k2}^* & -h_{k1}^* \end{bmatrix}_{L \times M} . \tag{53}$$

The matrices in (50) and (53) are an example for a 1/2 rate STBC code that utilizes $M = 4$ Tx antennas for $L = 8$ successive times [74, 75]. Obtaining the symbol estimate vector is simple because of the orthogonal structure of the code.

$$\hat{s}_k[n] = \mathbf{H}_k^H[n]\underline{\tilde{r}}_k[n] = C\mathbf{s}_k[n] + \eta[n], \tag{54}$$

where $C = \sum |h_k|^2$ is a positive scalar and η is, again, a white Gaussian noise because of the orthogonal structure of the code.

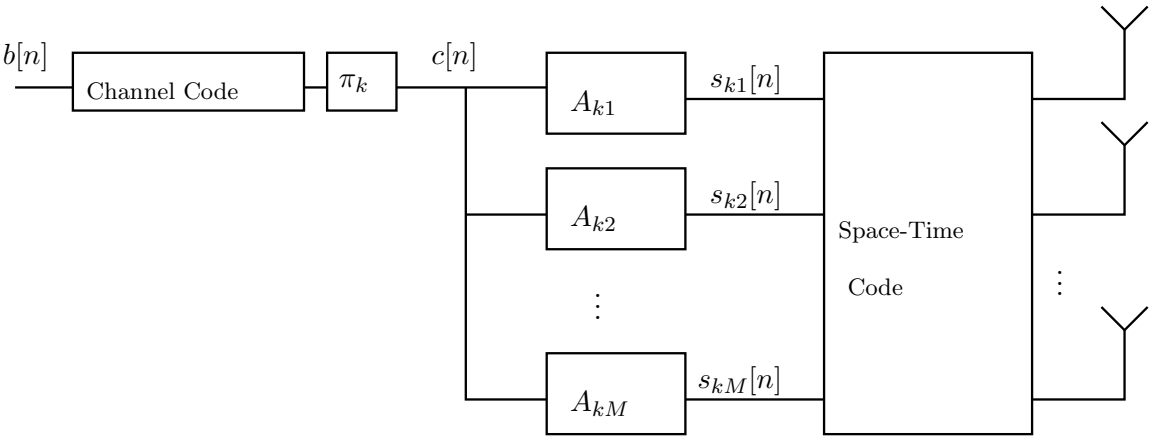


Figure 5: Transmitter model for a MOCS multiuser system

Here, the receiver utilizes one receive antenna (Figure 6). The performance of the system increases as more Rx antennas are used, but here we investigate transmit diversity

specifically. The receiver is assumed to have complete channel state information (CSI). The signal from the receive antenna is processed with a linear minimum mean-square error (MMSE) filter that also utilizes a priori information about the coding of the interferers. The result is the extrinsic log-likelihood ratio (LLR) of the desired user's coded data. This extrinsic information is processed with a soft-input soft-output (SISO) channel decoder utilizing the BCJR algorithm [2], producing an extrinsic LLR $\lambda_2[c_k[n]]$ of the coded symbols and LLR $\Lambda_2[d_k[n]]$ of the desired user's information symbols. The extrinsic LLR $\lambda_2[c_k[n]]$ is fed back to the MMSE filter to improve the estimate during the next iteration.

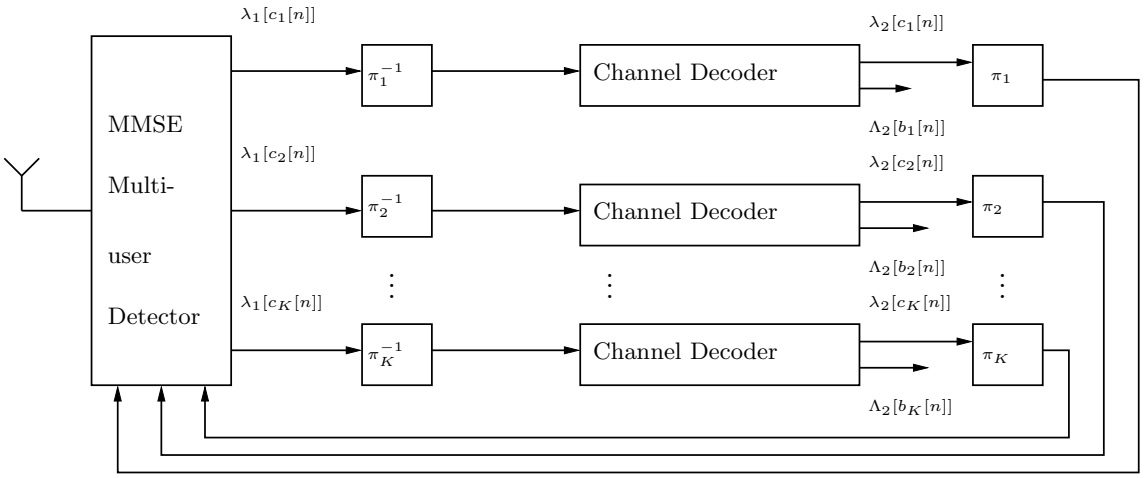


Figure 6: Turbo Receiver Model

Let us elaborate on the signaling at the transmitter end:

$$s_{km}[n] = c[n] * A_{km}[n], \tag{55}$$

where $(*)$ denotes convolution. We stack $s_{km}[n]$

$$\underline{s}_k[n] \triangleq [s_{k1}[n] \ \ldots \ s_{kM}[n]]^T. \tag{56}$$

We define the following quantities by stacking each user's successive samples of the coded data vectors. The goal here is to obtain a transmitter model that will account for all K users. $\mathbf{c}[n]$ is the channel coded symbols vector, $\mathbf{s}[n]$ is the vector of symbols filtered with

the MOCS. $\mathbf{r}[n]$ is the received signal, and $\eta[n]$ is the noise in the received signal:

$$\mathbf{c}[n] \triangleq \begin{bmatrix} \underline{c}[n-(p-1)] \\ \vdots \\ \underline{c}[n+(p-1)] \end{bmatrix}_{K(2p-1)\times 1} \tag{57}$$

$$\mathbf{s}[n] \triangleq \begin{bmatrix} \underline{s}_1[n] \\ \vdots \\ \underline{s}_K[n] \\ \vdots \\ \underline{s}_1[n+(p-1)] \\ \vdots \\ \underline{s}_K[n+(p-1)] \end{bmatrix}_{KMp\times 1} \tag{58}$$

$$\mathbf{r}[n] \triangleq [\underline{r}[n] \ \ldots \ \underline{r}[n+(p-1)]]^T_{Lp\times 1}, \tag{59}$$

$$\eta[n] \triangleq [\underline{\eta}[n] \ \ldots \ \underline{\eta}[n+(p-1)]]^T_{Lp\times 1} \tag{60}$$

where $\underline{c}[n] \triangleq [c_1[n], \ldots, c_K[n]]^T$.

Let us represent the convolution in Equation (55) with a convolution matrix, where

$$\mathbf{A}_k[i] \triangleq [A_{k1}[i] \ \ldots \ A_{kM}[i]]^T_{M\times 1}, \tag{61}$$

$$\mathbf{A}[i] \triangleq \begin{bmatrix} \mathbf{A}_1[i] & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{A}_K[i] \end{bmatrix}_{KM\times K} \text{, and} \tag{62}$$

$$\mathcal{A} \triangleq \begin{bmatrix} \mathbf{A}[\mathbf{p-1}] & \ldots & \mathbf{A}[0] & \ldots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{A}[p-1] & \ldots & \mathbf{A}[0] \end{bmatrix}_{KMp\times K(2p-1)}$$

$$\mathbf{s}[n] = \mathcal{A}\mathbf{c}[n] \tag{63}$$

Now, let us define $\mathbf{H}[n]$ so we can account for all K users.

$$\mathbf{H}[n] = \Big[\mathbf{H}_1^H[n] \ \dots \ \mathbf{H}_K^H[n] \Big]_{L \times KM}, \tag{64}$$

where $\mathbf{H}_k^H[n]$ is a block ST decoding matrix constructed as in [74, 75]. $\mathcal{H}[n]$is defined to be a block diagonal matrix with an STBC matrix in the main diagonal:

$$\mathcal{H} \triangleq \begin{bmatrix} \mathbf{H}[n] & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}[n+1] & \mathbf{0} & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{H}[n+p-1] \end{bmatrix}_{Lp \times KMp}. \tag{65}$$

Above, $\mathbf{H}[n]$ is an $L \times KM$ STBC matrix associated with time epoch n.

Let the matrix $\mathcal{P}$ be a diagonal matrix containing the amplitudes $\sqrt{P_k}$ of all K users' signals.

$$\mathcal{P} = \mathbf{I}_{2p-1} \otimes \mathrm{diag}[\sqrt{P_1} \ \dots \ \sqrt{P_K}]. \tag{66}$$

The received signal vector $\mathbf{r}[\mathbf{n}]$ can be described as

$$\mathbf{r}[\mathbf{n}] = \mathcal{H}[n]\mathcal{A}\mathcal{P}\mathbf{c}[n] + \eta[n] \tag{67}$$

Now, we have simplified our model to a single line of notation, and we have a sufficient vector $\mathbf{r}[n]$ that can be passed to the MMSE filter.

2.3 Performance Analysis

Equation (67) can be written as

$$\tilde{\mathbf{r}}[n] = \sum_{k=1}^{K} \sqrt{P_k}\mathbf{S}_k[n]\mathbf{h}_k[n] + \tilde{\eta}[n], \tag{68}$$

by taking the complex conjugate of the second half of $\mathbf{r}[n]$ [75]. Here, $\mathbf{h}_k^T[n]$ is a row vector containing the complex amplitudes of each channel. $\sqrt{P_k}$ is the transmit power amplitude of each user. $\mathbf{S}_k[n]$ is a ST code matrix constructed from the $s_{km}[n]$s [74, 75]. Notice that the statistics of the noise η are not affected. The new noise vector $\tilde{\eta}[n]$ is also a complex

Gaussian vector with zero mean and the same variance. Therefore, the pairwise error probability (PEP) of transmitting $\mathbf{S}[n]$ and receiving $\hat{\mathbf{S}}[n]$ is

$$\begin{aligned} P(\mathbf{S}[n] \leftrightarrow \hat{\mathbf{S}}[n]|\mathbf{h}[n]) &= Q\left(-\frac{\mathbf{h}[n]^T\mathbf{P}(\mathbf{S}[n]-\hat{\mathbf{S}}[n])(\mathbf{S}[n]-\hat{\mathbf{S}}[n])^T\mathbf{P}^T\mathbf{h}[n]}{4N_0}\right) \\ &\leq \exp\left\{-\frac{\mathbf{h}[n]^T(\mathbf{S}[n]-\hat{\mathbf{S}}[n])(\mathbf{S}[n]-\hat{\mathbf{S}}[n])^T\mathbf{h}[n]}{4N_0}\right\}. \end{aligned} \tag{69}$$

We assume that the channel is a Rayleigh fading channel. Hence, the channel coefficients are i.i.d. complex Gaussian random variables with zero mean and unit variance. We average the above expression to get rid of the conditioning on $\mathbf{h}[n]$. The resulting expression depends on the eigenvalues of the square of the symbol difference matrix, i.e. $(\mathbf{S}[n]-\hat{\mathbf{S}}[n])(\mathbf{S}[n]-\hat{\mathbf{S}}[n])^T$.

$$P(\mathbf{S}[n] \leftrightarrow \hat{\mathbf{S}}[n]) \leq \sum_{k=1}^{K}\prod_{i=1}^{M}\frac{1}{1+\frac{E_k}{4N_0}\lambda_{ik}}, \tag{70}$$

where the λ_{ik}s are the eigenvalues of the square of the symbol difference matrix. At high SNR

$$P(\mathbf{S}[n] \leftrightarrow \hat{\mathbf{S}}[n]) \leq \sum_{k=1}^{K}(\prod_{i=1}^{r}\lambda_{ik})^{-1}(\frac{E_k}{4N_0})^{-r},\ r \leq M. \tag{71}$$

We find that the rank of the above matrix does not change as the number of users increases, and the distribution of eigenvalues is not adversely affected by introducing additional users to the system. Thus, the system should perform well regardless of the number of users in it, as long as each user is assigned a distinct code set.

2.4 Low Complexity MMSE Multi-user Detector

The soft-input soft-output (SISO) MMSE filter takes the sufficient vector $\mathbf{r}[n]$ and the a priori LLR of the code bits of all users $\lambda_2[c_k[n]]$, and produces the extrinsic LLR of the code bits $\lambda_1[c_k[n]]$ by accounting for the interference to the desired user's signal.

First, soft estimates of the code bits are formed:

$$\tilde{\mathbf{c}}[n] = \tanh(0.5\lambda_2[\tilde{\mathbf{c}}[n]]) \tag{72}$$

Then $\tilde{\mathbf{c}}_k[n]$ is defined by setting the $K(p-1)+k$th element to zero:

$$\tilde{\mathbf{c}}_k[n] = \tilde{\mathbf{c}}[n] - \tilde{\mathbf{c}}[n]_{K(p-1)+k}\mathbf{e}_{K(p-1)+k}, \tag{73}$$

where $\mathbf{e}_{K(p-1)+k}$ is a column vector of zeros with a one at the $K(p-1)+k$ position. Soft interference cancellation is performed for each user k by subtracting the effect of the interferers

$$\mathbf{r}_k[n] = \mathbf{r}[n] - \mathcal{H}[n]\mathcal{A}\mathcal{P}\tilde{\mathbf{c}}_k[n] = \mathcal{H}[n]\mathcal{A}\mathcal{P}(\mathbf{c}_k[n] - \tilde{\mathbf{c}}_k[n]) + \eta[n]. \tag{74}$$

Then, a MMSE filter is applied to suppress the interference even more

$$z_k[n] = \mathbf{w}^H[n]\mathbf{r}_k[n], \tag{75}$$

where

$$\mathbf{w}[n] = \arg\min_{\mathbf{w}[n]} \mathrm{E}\{\|c_k[n] - \mathbf{w}^H[n]\mathbf{r}_k[n]\|^2\}. \tag{76}$$

Let us define $\mathcal{B} \triangleq \mathcal{H}[n]\mathcal{A}\mathcal{P}$ and $\mathcal{R}[n]$ as the covariance matrix $cov\{\mathbf{c}[n] - \tilde{\mathbf{c}}_k[n]\}$. Because of the interleaver effect and because of independent users, $\mathcal{R}[n]$ will be diagonal, and

$$\mathbf{w}[n] = [\mathcal{B}\mathcal{R}[n]\mathcal{B}^H + \sigma^2\mathbf{I}]^{-1}\mathcal{B}\mathbf{e}_{K(p-1)+k}. \tag{77}$$

The signal $z_k[n]$ is our estimate for the kth user's nth coded bit. However, we need to obtain the LLR $\lambda_1[c_k[n]]$

$$\lambda_1[c_k[n]] \triangleq \log \frac{p[z_k[n]|c_k[n]=1]}{p[z_k[n]|c_k[n]=-1]}. \tag{78}$$

Referring to [46, 93], we find that we can compute it as follows:

$$\lambda_1[c_k[n]] \triangleq \frac{4\mathbb{R}\{z_k[n]\}}{1 - \mathbf{e}^H_{K(p-1)+k}\mathcal{B}^H[\mathcal{B}\mathcal{R}[n]\mathcal{B}^H + \sigma^2\mathbf{I}]^{-1}\mathcal{B}\mathbf{e}_{K(p-1)+k}}, \tag{79}$$

where $\mathbb{R}\{\}$ denotes the real part of its argument. Notice that the matrix inversion above can be implemented efficiently using the well-known matrix inversion lemma:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H \tag{80}$$

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{\mathbf{H}}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \tag{81}$$

2.5 Soft Input Soft Output Channel Decoder

In our system, there are K channel decoders − one for each user. The input to the kth channel decoder is the extrinsic LLR $\lambda_1[c_k[n]]$ computed by the MMSE multiuser detector.

The channel decoder computes the extrinsic LLR $\lambda_2[c_k[n]]$ and the information bit LLR $\Lambda_2[b_k[n]]$ based on the constraints of the convolutional code.

The algorithm implemented here is the log-MAP algorithm [18, 93], which is a slight modification of the well-known BCJR algorithm. For a more elaborate discussion the reader may refer to [2].

Note that our system is a system of serially concatenated blocks. We must choose an appropriate outer code so that our system performs well. In such systems there is a constraint on the properties of the inner code - it must be recursive, because it must generate long error events [93]. However, there is no such constraint on the properties of the outer code. Thus, we choose a nonsystematic convolutional code.

2.6 Simulation Results

For our simulations, the channel encoder uses a rate 1/2 convolutional code with generators $23, 35$ in octal form. The channel was a flat, block static, Rayleigh fading channel. The receiver had complete CSI, and, hence, the time selectivity of the channel did not affect the performance of our system. We utilized $M = 4$ transmit antennas per user. The mutually orthogonal sequences were of length $p = 4$. We had four sequences per mate set and we had $K = 4$ mate sets − one for each user. All interferers had equal transmit power and we repeated the simulation for various signal-to-interference ratios (SIR). In our simulations we utilized a STBC for 4 Tx antennas [75].

In Figures 7 and 8, it is clear that the system performs very close to the single-user scenario. Although there is a loss of roughly $0.5 - 0.7$ dB, the diversity gain of the system is retained. The performance loss can be attributed to incorrect estimation of the interferers' signals, which causes incorrect detection of the desired user's signal. Note that at low signal-to-interference ratios (SIR) (Figure 9), the estimate of the desired user's signal improves significantly at each iteration. In this case, the interferers are stronger, their signals are detected more reliably, and their effect on the received signal is accounted for more accurately. At high SIR, in contrast, there is no significant improvement as we iterate the information back and forth between the decoders. In these cases, the interferers are too

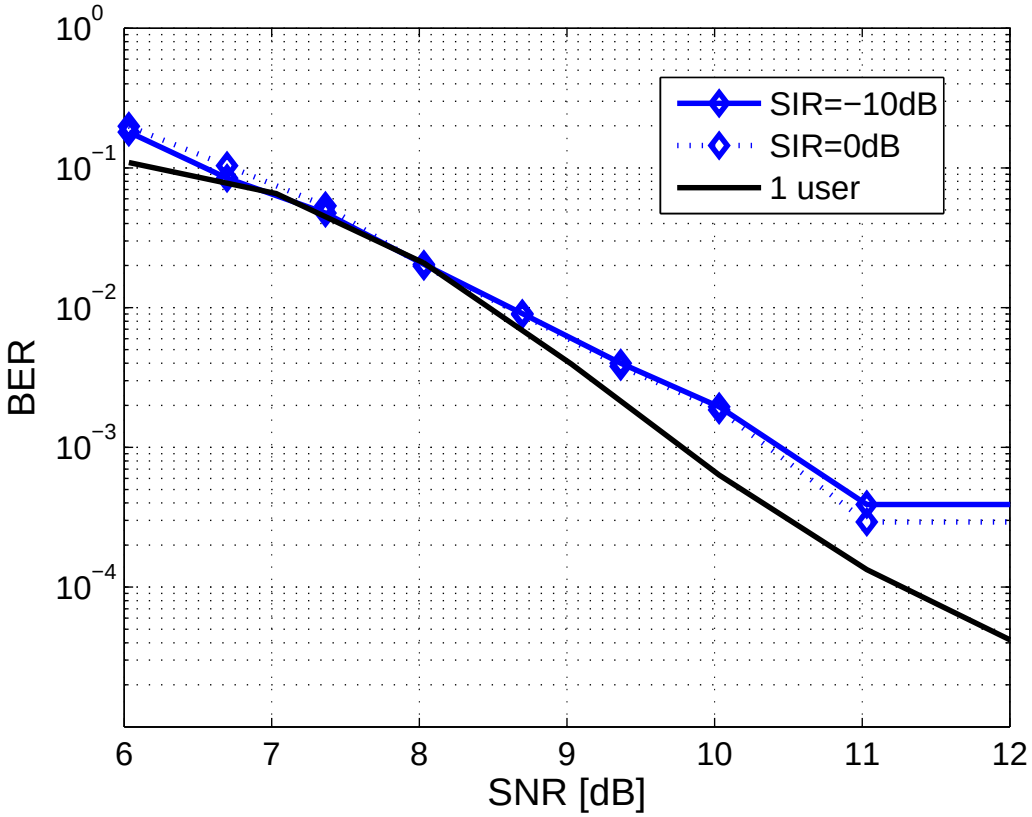


Figure 7: Performance of system with $K = 2$ users, 4 Tx antennas per user, after 5 iterations, compared to the single user case.

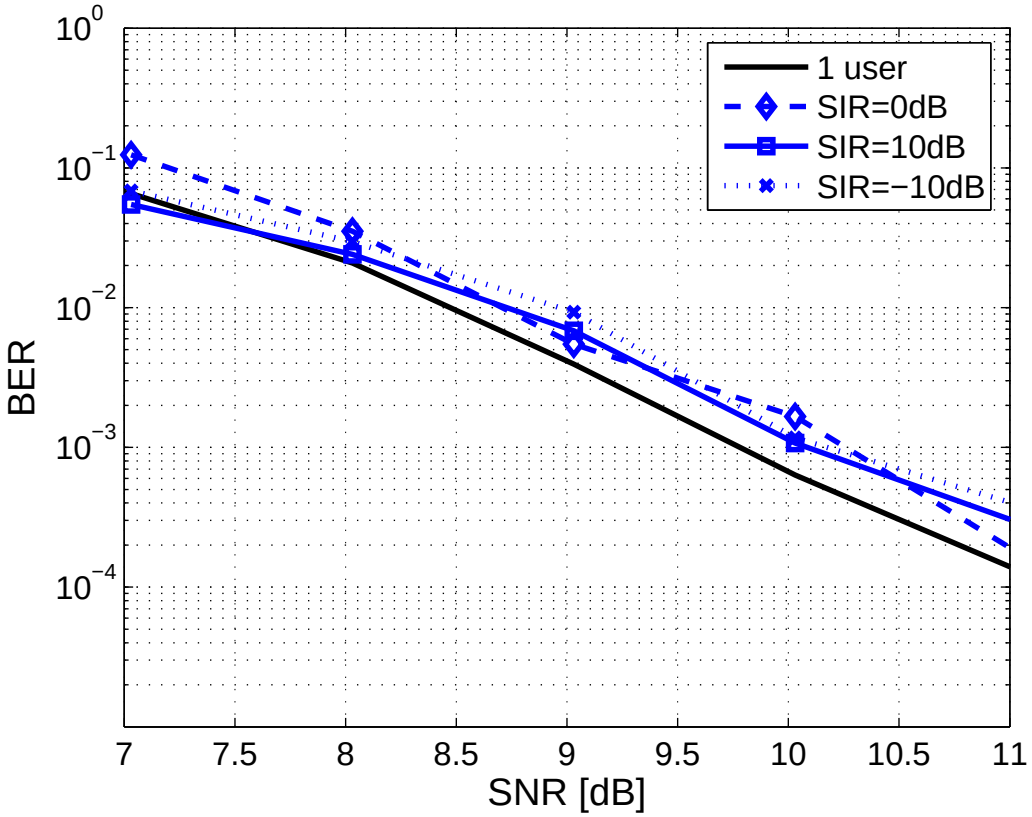


Figure 8: Performance of system with $K = 4$ users, 4 Tx antennas per user, after 5 iterations, compared to the single user case.

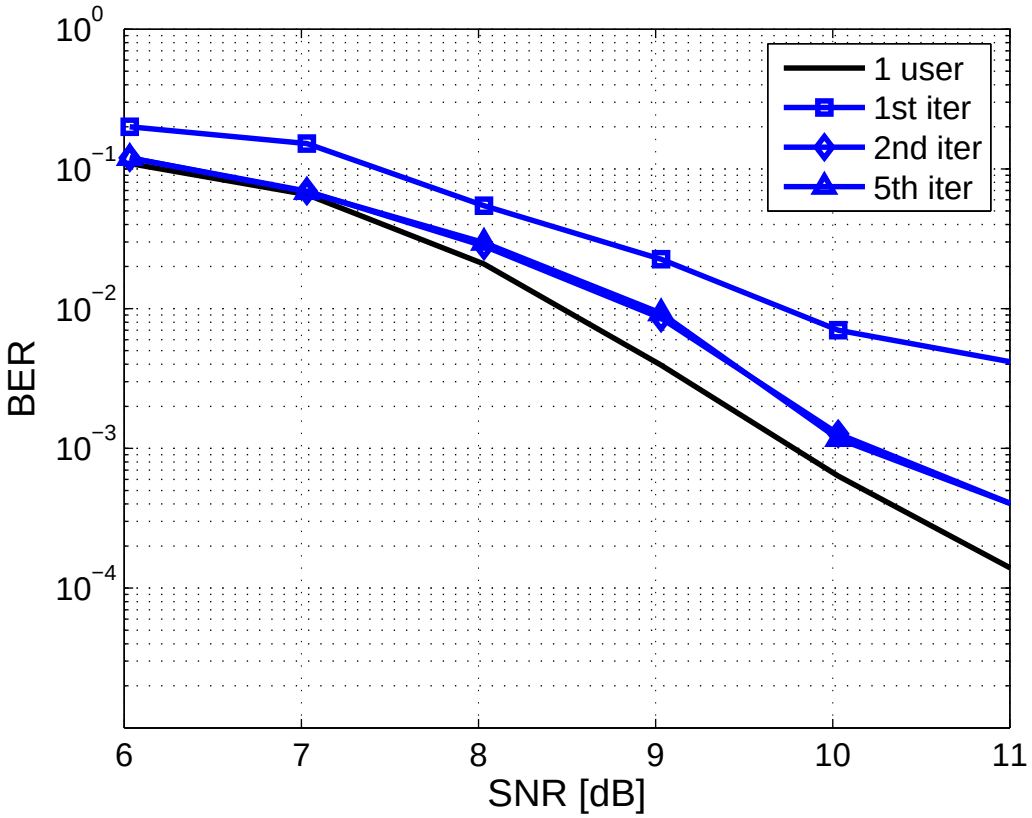


Figure 9: Convergence of system with $K = 4$ users, 4 Tx antennas per user, SIR=-10dB after 1,2, and 5 iterations, compared to the single user case.

weak to be estimated reliably and additional iterations are only moderately useful.

2.7 Conclusions

With this receiver we have utilized the properties of MOCS to achieve near single user performance for an iterative MMSE multiuser detector concatenated with a channel decoder. Additionally, the diversity order of the system is preserved as is seen from the results. At each iteration, the MUD improves the estimate of the desired user's signal.

The complexity of the proposed system increases only linearly with the number of users in the system. The performance results indicate that there is no need to decode all users jointly in a large trellis. Maximum likelihood sequence detection would be prohibitively expensive to implement.

CHAPTER III

A SEQUENTIAL MONTE CARLO ALGORITHM FOR FREQUENCY SELECTIVE FADING CHANNELS

3.1 Introduction

Detection of symbols transmitted through a time-varying frequency-selective channel is an important problem in telecommunications. Sophisticated algorithms have been developed to allow a performance close to theoretical capacity limits. When the channel parameters are known, optimal detection can be implemented based on maximum likelihood sequence detection methods such as the Viterbi algorithm [62]. When the channel coefficients are unknown, a training symbol sequence, which is known by the receiver in advance, is transmitted. This technique is effective, but results in efficiency loss, because a fraction of the bandwidth is wasted on sending training symbols. Therefore, much research has focused on blind methods. These methods detect the transmitted symbols, without knowledge of channel state information or training data. Such algorithms usually do joint channel estimation and symbol detection, there are numerous ways to do this, including the expectation maximization algorithm, or utilizing the Viterbi algorithm and implementing per-survivor-processing (PSP) [3, 62].

Until recently, Monte Carlo methods have been ignored because of their computational requirements. With the advances in computation technology, these methods have gained popularity again. These methods are called Monte Carlo methods because they form an estimate for a parameter from a random measure that approximates the true distribution. Monte Carlo Markov chain (MCMC) methods and sequential Monte Carlo methods particularly have attracted significant attention [95].

The Gibbs sampler, which is an MCMC method, has been widely used to estimate stationary features. The method requires the assumption that the parameter to be estimated

does not vary in time. In addition, this method is a batch processing method. It requires that the entire batch of observations be received before processing begins [95]. This method cannot be applied without modification to estimate a time-varying channel, because every time the channel state changes significantly, a long burn-in period must be performed. Therefore, a mechanism that would sense the channel changes must be incorporated in the algorithm.

The sequential Monte Carlo method, as its name suggests, does not require that all observations be available. It can process the data on the fly, as data is received. This method calculates samples (called particles) of the desired posterior distribution and assigns weights to them. With these pairs, any estimator can be tightly approximated. Another name for this method is sequential importance sampling (SIS). It has been reported that BER bounds are lower for this method compared to the Gibbs sampler method [54]. Numerous articles on SIS applications to communications have been written by Poor and Wang and then collected in their book [95].

In this work we tackle the blind equalization issue for single user communication systems. We define the signal model and a Bayesian formulation of the problem. We also use Rao-Blackwellization to simplify the channel update part of the problem to reduce the necessary number of particles. The main contribution of this work is in the reducing of the complexity of the SMC algorithm. We propose temporal partitioning of the symbol space. We draw sub-symbols from each subspace and then we merge the results. The proposed partitioning reduces the computational complexity without any performance loss or additional error propagation effects.

The organization of the work is as follows. Section 3.2 describes the signal model. In section 3.4 the recursive density updates that are necessary for the SMC algorithm are formulated. The incremental weight updates and the choice of a proposal function are described in section 3.5. In section 3.6 Rao-Blackwellization is described. Channel modeling and Doppler effects are modeled in section 3.3. Channel resampling methods and an extra channel particle update after resampling are described in sections 3.7 and 3.8. The algorithm is summarized in section 3.9. Symbol space partitioning and the modified

algorithm are described in section 3.10 and 3.11. Finally, results and conclusions are in sections 3.12 and 3.13.

3.2 The Signal Model

Consider a digital communications system where BPSK symbols $s_n \in \pm 1, n = 0, 1, \ldots$ are transmitted through a frequency selective multipath fading channel. Assuming a receiver front-end consisting of a matched filter and a symbol sampler, the discrete time received signal can be written as

$$r_n = \sum_{l=0}^{L-1} s_{n-l} h_n[l] + v_n, \qquad n = 0, 1, 2, \ldots, \tag{82}$$

where $v_n = \mathcal{N}(0, \sigma^2)$ is a complex additive white Gaussian noise with zero mean and variance σ^2, and $h_l[n]$ is the discrete-time channel impulse response. Assuming the channel impulse response is finite, it can be represented by the $L \times 1$ vector

$$\mathbf{h}_n \triangleq [h_n[d-1], h_n[d-2], \ldots, h_n[0]]^T, \tag{83}$$

where the superscript ^{T} denotes transposition, and d is the number of resolvable propagation paths. The channel order d determines the dimension of the vectors and the span of the intersymbol interference (ISI). Every symbol s_n is dispersed in d consecutive observations of the received signal r_n.

The symbols can also be represented in a vector form:

$$\mathbf{s}_n \triangleq [s_{n-L+1}, s_{n-L+2}, \ldots, s_n]^T. \tag{84}$$

The received signal equation (82) can be written more compactly as:

$$r_n = \mathbf{s}_n^T \mathbf{h}_n + v_n, \qquad n = 0, 1, 2, \ldots \tag{85}$$

The notation $s_{0:n}$ represents the set $s_0, s_1, \ldots, s_n$. Similar notation will be used in describing the density functions below.

3.3 Channel Modeling

The communications channel is a slow Rayleigh fading channel. Its temporal autocorrelation can be characterized by a zeroth order Bessel function of the first kind [72]

$$\phi_{h[l]h[l]}(\tau) = \frac{1}{2} J_0(2\pi f_m \tau). \tag{86}$$

The symbol τ, above, is the time delay, and f_m is the Doppler frequency shift.

The power spectral density of this process is a non-rational function (equation (87)) and must be approximated by an autoregressive moving average process of high order. However, the impulse response of such a process is long and the increases computational complexity significantly [72]

$$S_{h[l]h[l]}(f) = \begin{cases} \frac{1}{4\pi f_m} \frac{1}{\sqrt{1-(f_m/f)^2}}, & |f| \leq f_m \\ 0, & \text{otherwise} \end{cases}. \tag{87}$$

Instead, our algorithm uses a first order autoregressive model (equation (88)). Essentially, the Bessel function is approximated by an exponential function. The discrepancy between the true channel and the model used by the algorithm is compensated by increasing the number of particles.

$$\mathbf{h}_n = \zeta \mathbf{h}_{n-1} + (1-\zeta)\eta_{n-1}, \tag{88}$$

where η_n is a complex Gaussian random process with zero mean and 1 variance and ζ is defined as

$$\zeta = 2 - cos(2\pi f_m T) - \sqrt{(2 - cos(2\pi f_m T))^2 - 1} \tag{89}$$

3.4 Recursive Definition of the Posterior

The maximum a posteriori estimate of the transmitted symbol sequence is

$$s_{0:N} = \arg\max_{s_{0:N}} p(s_{0:N} | r_{0:N+L-1}). \tag{90}$$

The optimum solution is given by the maximum likelihood sequence detector, whose computational complexity grows exponentially with the number of symbols N and the length of

the channel impulse response L. This solution is out of the question because of its enormous complexity. A lower complexity, recursive update of the posterior density can be considered as an alternative.

The posterior density is difficult to calculate, but it is proportional to this likelihood.

$$\begin{aligned} p(s_{0:n}|r_{0:n+L-1}) &= \int p(s_{0:n}|r_{0:n+L-1}, \mathbf{h}_{0:n}) p(\mathbf{h}_{0:n}) \mathrm{d}\mathbf{h}_{0:n} \\ &\propto \sum_{s_{n+1:n+L-1} \in \{\pm 1\}^{L-1}} \int_{\mathbf{h}_{0:n}} p(r_{0:n+L-1}, s_{n+1:n+L-1}|s_{0:n}, \mathbf{h}_{0:n}) \\ &\qquad p(\mathbf{h}_{0:n}) \mathrm{d}\mathbf{h}_{0:n} \end{aligned} \tag{91}$$

The summation comes from the assumption of iid symbols. Alternatively, the posterior density for the symbol s_n rather than the whole sequence is

$$\begin{aligned} p(s_n|r_{n:n+L-1}) &= \int p(s_n|r_{0:n+L-1}, \mathbf{h}_{n:n+L-1}) p(\mathbf{h}_{n:n+L-1}) \mathrm{d}\mathbf{h}_{n:n+L-1} \\ &\propto \sum_{s_{n+1:n+L-1} \in \{\pm 1\}^{L-1}} \int_{\mathbf{h}_{n:n+L-1}} p(r_{0:n+L-1}, s_{n+1:n+L-1}|s_{0:n}, \mathbf{h}_{n:n+L-1}) \\ &\qquad p(\mathbf{h}_{n:n+L-1}) \mathrm{d}\mathbf{h}_{n:n+L-1} \end{aligned} \tag{92}$$

We further factorize the above likelihood in the integrand trying to obtain a recursive expression.

$$\begin{aligned} p(r_{0:n+L-1}|s_{0:n}, \mathbf{h}_{0:n}) &= \\ &= \sum_{s_{n+1:n+L-1}} \int p(r_{n:n+L-1}, s_{n+1:n+L-1}, h_n|r_{0:n-1}, s_{0:n-1}, \mathbf{h}_{0:n-1}) \\ &\qquad p(r_{0:n-1}|s_{0:n-1}, \mathbf{h}_{0:n-1}) p(h_n|\mathbf{h}_{0:n-1}) \mathrm{d}h_n \end{aligned} \tag{93}$$

A delay of $L-1$ samples is introduced because the effect of the nth transmitted symbol is dispersed in the next $L-1$ observations.

3.5 Incremental Weight Update

The complicated high dimensional posterior density can be represented by multiple Monte Carlo samples drawn from it. We obtain a set of samples and assign weights to it; $w_n^{(i)}, i = 1, \ldots, N_s$; so it is properly weighted with respect to the posterior that it represents. Note

that $\{s_n^{(i)}, w_n^{(i)}\}_{i=1}^{N_s}$ denotes a random measure that characterizes the a posteriori distribution $p(s_{0:n}|r_{0:n+L-1})$.

The symbol $s_n^{(i)}$ is sampled from the proposal density $q(s_n^{(i)}|r_{n:n+L-1})$ and the weights are set to

$$w_n^{(i)} = \frac{p(s_{0:n}^{(i)}|r_{0:n+L-1})}{q(s_{0:n}^{(i)}|r_{0:n+L-1})}. \tag{94}$$

After applying the recursion above we get

$$w_n^{(i)} = w_{n-1}^{(i)} \frac{p(s_{0:n}^{(i)}|r_{0:n+L-1})}{p(s_{0:n-1}^{(i)}|r_{0:n+L-2}) q(s_n^{(i)}|s_{0:n-1}^{(i)}, r_{0:n+L-1})}. \tag{95}$$

If the proposal density $q(\cdot)$ is chosen as

$$q(s_n|s_{0:n-1}, r_{0:n+L-1}) \triangleq p(s_n|s_{0:n-1}, r_{0:n+L-1}), \tag{96}$$

Then, the weight update simplifies even further to

$$\begin{aligned} w_n^{(i)} &= w_{n-1}^{(i)} \frac{p(s_{0:n-1}^{(i)}|r_{0:n+L-1})}{p(s_{0:n-1}^{(i)}|r_{0:n+L-2})} \\ &\propto w_{n-1}^{(i)} \sum_{s_{n+1:n+L-1}^{(i)} \in \{\pm 1\}^{L-1}} \int p(r_{n:n+L-1}, s_{n+1:n+L-1}^{(i)}, \mathbf{h}_{n:n+L-1}|r_{0:n-1}, s_{0:n-1}^{(i)}, \mathbf{h}_{0:n-1}) \\ &\qquad p(\mathbf{h}_{n:n+L-1}|\mathbf{h}_{0:n-1}) \mathrm{d}\mathbf{h}_{n:n+L-1} \\ w_n^{(i)} &\propto w_{n-1}^{(i)} u_n^{(i)}. \end{aligned} \tag{97}$$

3.6 Rao-Blackwellization of the Channel Coefficient Update

We need to evaluate the above incremental weight update. The distribution actually is very easy to evaluate. The likelihood $p(r, \cdot|\cdot, \cdot)$ is Gaussian because of equation (85). The channel is a Rayleigh channel, which implies that the second distribution is also Gaussian. Hence, the weight update is also Gaussian. Because of Gaussianity and linearity in the observation equation (85), we can Rao-Blackwellize the channel coefficient updates [48]. Namely, instead of representing the posterior distribution with a discretized approximation, we can represent it with the analytically computed first and second moments of the Gaussian distribution .

The channel coefficients are Gaussian random processes with mean $\mathbf{h}_n^{(i)}$ and covariance $\mathbf{\Sigma}_n^{(i)}$. The weight update function is also Gaussian with mean $\mu_n^{(i)}$ and covariance $P_n^{(i)}$. Instead of sampling from $p(\mathbf{h})$, our algorithm updates the mean and covariance of the channel estimates

$$u_n^{(i)} \sim \mathcal{N}(\mu_n^{(i)}, P_n^{(i)}), \tag{98}$$

where

$$\mu_n^{(i)} = {\mathbf{s}_n^{(i)}}^T \mathbf{h}_n^{(i)} \tag{99}$$

and

$$P_n^{(i)} = \sigma^2 + {\mathbf{s}_n^{(i)}}^T \mathbf{\Sigma}_n^{(i)} \mathbf{s}_n^{(i)}. \tag{100}$$

The channel state can be updated with Kalman filter equations.

$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(r_n - \mu_n^{(i)}) \tag{101}$$

$$\mathbf{\Sigma}_n^{(i)} = (1 - K_n^{(i)}){\mathbf{s}_n^{(i)}}^T \mathbf{\Sigma}_{n-1}^{(i)}, \tag{102}$$

where the Kalman gain is

$$\mathbf{K}_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)} \mathbf{s}_n^{(i)} / \mathbf{P}_n^{(i)}. \tag{103}$$

3.7 Resampling

Sequential Monte Carlo algorithms suffer from a degeneracy problem. After a number of iterations, most of the particles diverge from the state they are expected to estimate, and are accordingly, assigned a low weight. Only a few of the particles concentrate on the mode of the distribution. Eventually, only one particle remains at the mode of the distribution, and the rest get scattered in the tails, resulting in low efficiency. Namely, computational resources are spent on calculating particle likelihoods, even though their weight is zero and they don't contribute to the estimates [50].

A solution called resampling replicates the particles with high weight and deletes the ones with low weight. If done often enough (but not too often), the particles are kept from diverging and algorithm efficiency is maintained [50].

A method called simple random sampling, replicates the particles with a probability proportional to their weights. At the end, all resulting particles are assigned equal weights

[50]. Typically the Metropolis-Hastings algorithm or the acceptance-rejection method is used to draw the particles.

Another method called residual sampling, is reported to be sightly better [50] and is summarized in Table 5. With this method, fewer particles must be drawn at random, which

Table 5: Residual Resampling Pseudo-code

- Let $w^{(i)}$ be the normalized weights of the particles. Retain $k_i = \lfloor N_s w^{(i)} \rfloor$ copies of $h_n^{(i)}$ for each i.
- Let $N_{sr} = N_s - \sum_{i=1}^{N_s} k_i$ be the number of non-replicated particles.
- Obtain N_{sr} iid draws from the original particles set with probabilites proportional to $N_s w^{(i)} - k_i, \;\; i = 1, \ldots, N_s$.
- Assign equal weights.

relieves the computational complexity. Also the efficiency of the random draw algorithm is increased, because the distribution from which the particles are drawn is more uniform and, hence, suitable for A-R.

3.8 Extra Reshuffle

This step is optional (Table 7). If utilized, an extra reshuffle improves the channel coefficient estimate variance. The idea comes from the fact that, after resampling, many particles are identical. Also, some particles, even though their likelihood is high, may not be significantly close to the mode of the distribution.

We apply this step on the channel coefficient particles. It does not improve the current symbol decision, because the likelihoods are evaluated based on already drawn symbols. However, this step improves the channel tracking ability of the algorithm. The channel coefficients for the next time step are estimated with smaller variance.

Table 6: Additional Perturbation

1. Particle likelihoods are calculated.
2. A very small random perturbation is added to each particle.
3. The resulting likelihoods are calculated.
4. For each particle, acceptance-rejection method is used to decide if the resulting particle should be substituted for the old one or not.
5. go to step 1 and repeat 5-10 times

3.9 The SMC algorithm

We summarize the standard sequential Monte Carlo algorithm for blind equalization in Table 7. The process of obtaining symbol samples is described in Table 8. Each channel sample is evolved according to the channel model. Then, the likelihoods $\beta(\cdot)$ of all possible transmitted symbols are computed. The transmitted symbol is chosen based on the realization of a random variable drawn from a distribution that is proportional to the computed likelihoods. The symbol's weight, which is also a measure of certainty, is a function of the same likelihoods. Based on the outcome of the random draw, the channel sample is updated according to the Rao-Blackwellized equations. This process is repeated for each channel sample. After the iteration is completed, we have obtained a collection of symbol samples and their weights. The estimate of the transmitted symbol is just a weighted sum of these particles.

One problem with this algorithm is that it must account for all possible transmitted symbols. Likelihoods for all possible transmitted symbols from an L dimensional constellation space must be calculated and stored. The symbol estimate is drawn from a distribution that is proportional to the computed likelihoods. The symbol can be drawn using an A-R method, M-H method or a Gibbs sampler method as described earlier. For that particular received signal, the symbol distribution conditioned on the channel information is static, rather than dynamic. Therefore MCMC methods are acceptable. To obtain a reliable

Table 7: Blind equalizer SMC pseudo-code

- Initialization
- for $i = 1, \ldots, N_s$ /*for each particle*/
  - Calculate $\beta(\cdot)$ as described in Table 8
  - Draw $\tilde{s}_{n+k}^{(i)} \sim \beta(\cdot)$ as described in Table 8
  - Update

$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(\mathbf{r}_n - \mu_n^{(i)})$$

$$\mathbf{\Sigma}_n^{(i)} = (1 - K_n^{(i)})\mathbf{s}_n^{(i)^T}\mathbf{\Sigma}_{n-1}^{(i)},$$

    where the Kalman gain is:

$$K_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)}\mathbf{s}_n^{(i)}(P_n^{(i)})^{-1}$$

  - Update weight $w_n^{(i)} \propto w_{n-1}^{(i)}\beta_{n:n+L-1}^{(i)}(\cdot)$
- Obtain symbol estimate as

$$\hat{s}_n = \arg \max_{s_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)}\delta(s_n - s_n^{(i)}),$$

- Resample if needed
- Perturb channel particles and update based on likelihoods.

estimate, again a burn-in period is required, which is an additional computational burden.

In the next section we propose a method that uses the Markovian structure of the observations and reduces the overall complexity by partitioning the symbol space and calculating the likelihoods recursively.

3.10 Temporal Partitioning

The standard Monte Carlo method for blind equaliziation is of high complexity. When the channel has a long impulse response, the complexity of the algorithm increases significantly. The problematic areas of the conventional MC based blind equalizer can be itemized as follows.

- A likelihood function, which is also the incremental weight update, must be evaluated 2^L times for each channel particle.
- An L dimensional symbol must be drawn from an L dimensional distribution. Typically, a variation of Metropolis-Hastings algorithm, or an accept-rejection method is used for symbol draws. To draw a proper symbol of this dimensionality, a significant number of trials must be made, which also increases the complexity.
- Vector dimensions grow linearly with L, which also increases the computational complexity of the channel update equations.

The biggest problem here is the high dimensionality of the likelihood function from which the symbols are drawn. Upon careful analysis, we notice the conditional independence of the received signals, which makes factorization of the likelihood function possible

$$\begin{aligned} p(r_{n:n+L-1}, s^{(i)}_{n+1:n+L-1}, \mathbf{h}_{n:n+L-1} | r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}) = \\ = \prod_{l=0}^{L-1} p(r_{n+l}, s_{n:n+l}, \mathbf{h}_{n+l} | r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}). \end{aligned} \tag{104}$$

As a result of this factorization we get L distributions $\beta_{l,n}(\cdot), \quad l = 0, \ldots, L-1$,

$$\beta_{l,n}(r_{n+l}, s_{n:n+l}, \mathbf{h}_{n+l} | r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}). \tag{105}$$

In the standard algorithm, to estimate s_n, L consecutive received signals $r_{n:n+L-1}$ must be considered, because of ISI. Evaluation of the likelihood function and a draw of a multi-dimensional symbol $s_{n:n+L-1}$ is necessary. However, it is interesting to see that the symbol s_{n+L-1} has effect only on r_{n+L-1}. Respectively, the symbol $s_{n+L-2:n+L-1}$ has effect on $r_{n+L-2:n+L-1}$ only. Following the pattern we can draw the "sub-symbols" sequentially. Namely,

- Evaluate

$$\sum_{s^{(i)}_{n:n+L-2}\in\{\pm 1\}^{L-2}} \beta_{L-1,n}(r_{n+L-1}, s_{n:n+L-1}, \mathbf{h}_{n+L-1}|r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}) \tag{106}$$

  for $s_{n+L-1} \in \{\pm 1\}$, and draw $\tilde{s}_{n+L-1}$.

- Substituting the drawn symbol $\tilde{s}_{n+L-1}$ in $\beta_{L-1,n}(\cdot)$, evaluate

$$\sum_{s^{(i)}_{n:n+L-3}\in\{\pm 1\}^{L-3}} \beta_{L-1,n}(r_{n+L-1}, s_{n:n+L-2}, \tilde{s}_{n+L-1}, \mathbf{h}_{n+L-1}|r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}) \tag{107}$$

  and

$$\sum_{s^{(i)}_{n:n+L-3}\in\{\pm 1\}^{L-3}} \beta_{L-2,n}(r_{n+L-2}, s_{n:n+L-2}, \mathbf{h}_{n+L-2}|r_{0:n-1}, s^{(i)}_{0:n-1}, \mathbf{h}_{0:n-1}) \tag{108}$$

  for $s_{n+L-2} \in \{\pm 1\}$. Take their product, and draw $\tilde{s}_{n+L-2}$.

- Repeat the recursion until $\tilde{s}_n$ is drawn, which is the desired draw.

The SMC algorithm for blind equalization with temporal partitioning is summarized in Table 9. Note how the likelihood calculation and symbol draws are interleaved. This also results in reduced storage requirements: once a 'sub-symbol' is drawn, there is no need to store the likelihoods of the unlikely 'sub-symbols'. In the table only the modifications in computing the likelihoods are shown. The symbol draws are again done by an MCMC method such as the A-R algorithm or the M-H algorithm.

Table 8: Symbol draw pseudo-code

- for $l = 0, ..., L-1$
  - Compute

$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1}$$
$$\mu_{n+l}^{(i)} = {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{h}_{n+l}^{(i)}$$
$$P_{n+l}^{(i)} = \sigma^2 + {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{\Sigma}_{n+l}^{(i)} \mathbf{s}_{n+l}^{(i)}$$

    , for each possible $\tilde{s}_{n:n+L-1}^{(i)} \in \{\pm 1\}^L$
  - Compute $\beta_{n+l}^{(i)}(\cdot) = p(r_{n+l}|\mathbf{h}_{n+l}^{(i)}, \tilde{s}_{n:n+l}^{(i)}, s_{0:n-1})$
- Draw $\tilde{s}_{n:n+L-1}^{(i)} \sim \beta_{n:n+L-1}^{(i)}(\cdot)$

Table 9: Likelihood calculation and sub-symbol draws based on temporal partitioning of symbol space

- for k=m-1,...,0 /*start with last sub-symbol*/
  - for l=k,...,m-1 /*iterate*/
    - Compute

$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1} \quad (109)$$
$$\mu_{n+l}^{(i)} = {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{h}_{n+l}^{(i)}$$
$$P_{n+l}^{(i)} = \sigma^2 + {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{\Sigma}_{n+l}^{(i)} \mathbf{s}_{n+l}^{(i)}$$

      ,for each possible $s_{n+k}^{(i)} \in \{\pm 1\}$
    - Compute $\beta_{l,n}^{(i)}(\cdot) = p(r_{n+l}|\mathbf{h}_{n+l}^{(i)}, s_{n+k}^{(i)}, \tilde{s}_{n+k+1:n+L-1}^{(i)}, s_{0:n-1})$
  - Draw $\tilde{s}_{n+k}^{(i)} \sim \prod_{l=k}^{m-1} \beta_{l,n}^{(i)}$

3.11 Nested SMC approach

The above algorithm is based on back-substitution. Namely, a symbol is obtained from observations that form sufficient statistics, and in the next recursion step the symbol is used to form the distribution for the next symbol.

The drawn sub-symbol is the best estimate one can obtain, because it is based on sufficient statistics. However, if an incorrect decision is made, the effect propagates and affects the decisions about the rest of the sub-symbols. One may argue that an incorrect decision made by all particles happens extremely rarely. In other words, if the number of particles is set to a sufficiently large number, the majority of the particles correct the overall decision. This is not quite the case though. Because of spectral nulls, incorrect decisions can be very frequent become the majority.

Let us illustrate this phenomenon with an example. Suppose the impulse response of our communications channel is time-invariant and is set to $\mathbf{h} = [1, 1, 1]^T$. Let the transmitted symbols be BPSK symbols that are set to $s_{n-2:n+2} = \{1, -1, -1, 1, 1\}$. The receiver noise is AWGN with variance σ^2. Let us also assume that the all symbols in the past have been detected correctly. The received signals at times $n, n+1$, and $n+2$ are:

$$\begin{aligned} r_n &= -1 + \eta_n \\ r_{n+1} &= -1 + \eta_{n+1} \\ r_{n+2} &= 1 + \eta_{n+2} \end{aligned} \tag{110}$$

Performing the recursion, we take r_{n+2} and evaluate the log-likelihood $\beta_{n+2}(\tilde{s}_{n:n+2}|\cdot) = -(r_{n+2} - \tilde{s}^T_{n:n+2}\mathbf{h})^2$.

$$
\begin{aligned}
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{-1,-1,-1\}|\cdot) \sim -16 - 32\eta_{n+2} \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{-1,-1,1\}|\cdot) \sim -4 - 8\eta_{n+2} \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{-1,1,-1\}|\cdot) \sim -4 - 8\eta_{n+2} \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{-1,1,1\}|\cdot) = -\eta_{n+2}^2 \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{1,-1,-1\}|\cdot) \sim -4 - 8\eta_{n+2} \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{1,-1,1\}|\cdot) = -\eta_{n+2}^2 \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{1,1,-1\}|\cdot) = -\eta_{n+2}^2 \\
&\beta_{n+2}(\tilde{s}_{n:n+2} = \{1,1,1\}|\cdot) \sim -16 - 32\eta_{n+2}
\end{aligned}
\tag{111}
$$

After marginalization, almost equal likelihoods are assigned to $\tilde{s}_{n+2} = 1$ and $\tilde{s}_{n+2} = -1$. An incorrect choice here propagates and eventually affects the outcome of $\tilde{s}_n$. This is precisely the reason why a high order of diversity can not be obtained with back-substitution based methods.

The channel in the above scenario has zeroes on the unit circle in the complex $z-$ plane and, therefore, has adverse effect on the transmitted signal which has uniform spectrum. This particular scenario is not a rare and isolated event. In fact the roots of random polynomials tend to group around the unit circle in the complex plane [71]. Namely, a fading channel has deep nulls in its spectrum with very high probability. Conversely, from a time-domain perspective, the energy of a symbol is spread to neighboring symbols creating ISI. The goal of wide-band communications is to take advantage of spectral diversity and improve the symbol detection rate at high SNRs.

To eliminate the decision error propagation effect, we propose a nested particle filtering approach to the equalization algorithm similar to the algorithms used in sphere decoding [34]. Instead of drawing sub-samples and keeping them fixed, we sample the sub-symbol space with particles and retain a portion of them for further processing. The weights of the particles are updated as we proceed to the next sub-symbol. Because of the Markovian structure, we can take advantage of the recursion again. This approach retains choices with nearly equal likelihoods (like in the above example) and passes them on to the next stage, reducing decision error propagation.

Effectively we only change the portion of the algorithm that calculates the likelihoods and draws the symbol s_n. The algorithm, as summarized in Table 10, is a sequential Monte Carlo approach by itself. The Markovian structure of the likelihood functions is used to derive an artificial sequential dependence of the observations. The structure is further utilized to generate a list of trajectories which expand at each step.

The special property of this SMC application is that the parameter state-space is discrete. A conventional resampling stage after each trajectory expansion would yield replicating trajectories. In a continuous state-space scenario resampling is used to combat degeneracy. However in this case, it results in wasted computational resources, because of replicated trajectories.

A different resampling strategy that does not result in replicated trajectories must be utilized. Initaially there is one null trajectory. At each subsequent stage, the number of trajectories grows progressively. When the number of trajectories does not exceed the maximum allowed by the system, which is a design parameter chosen by the user, there is nothing to do. All possible states are fully represented. However, when there are more trajectories than this maximum, a subset of them must be chosen and retained according to some criteria.

A residual resampling approach is utilized here. Those trajectories with normalized likelihoods greater than or equal to $1/N_{particles}$ are retained. The remaining spots of the total of $N_{particles}\_max$ trajectories are taken by trajectories randomly chosen among the rest. Those trajectories are selected based on their likelihoods.

The overall algorithm is summarized in Table 11.

3.12 Simulation Results

We simulate the performance of our algorithm in a Rayleigh frequency selective slow fading channel. The simulated channel has delay spread of $m = 2, 3$, and 4 symbols. The paths are set to coincide with the delays of consecutive symbols. Namely, the delay for the path 0 is $\tau_0 = 0$. The delay of path 1 is $\tau = T$, where T is a symbol period. Let us also denote the variance of the complex gain of path 0 with σ_0^2. The variance of path 1 would be represented

Table 10: Symbol draw using nested SMC

- Set $N_{partilces} = 1$
- for k=0,...,m-1 /*start with first sub-symbol*/
  - Compute

$$\mathbf{h}_{n+k} = \zeta \mathbf{h}_{n-1+k} + (1-\zeta)\eta_{n-1} \tag{112}$$

$$\mu_{n+k}^{(i)} = {\mathbf{s}_{n+k}^{(i)}}^T \mathbf{h}_{n+k}^{(i)}$$

$$P_{n+k}^{(i)} = \sigma^2 + {\mathbf{s}_{n+k}^{(i)}}^T \mathbf{\Sigma}_{n+k}^{(i)} \mathbf{s}_{n+k}^{(i)}$$

    ,for each possible $s_{n+k}^{(i)} \in \{\pm 1\}$
  - Compute $p(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = -(r_{n+k} - \mu_{n+k}^{(i)})^2 / P_{n+k}^{(i)}$
  - For $j = 1, \ldots, N_{particles}$
    - Expand each trajectory to $|\{\pm 1\}| = 2$ (alphabet_size) distinct paths

$$\beta^{(2*j)}(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = \beta^{(j)}(r_{n+k-1}|\mathbf{h}_{n+k-1}^{(i)}, \mathbf{s}_{n+k-1})$$
$$p(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k-1}, s_{n+k} = -1)$$

$$\beta^{(2*j+1)}(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = \beta^{(j)}(r_{n+k-1}|\mathbf{h}_{n+k-1}^{(i)}, \mathbf{s}_{n+k-1})$$
$$p(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k-1}, s_{n+k} = +1)$$

  - Update $N_{particles} = N_{particles} * alphabet\_size$
  - if $N_{particles} > N_{partilces}\_max$ Truncate the trajectory list to $N_{partilces}\_max$ and set $N_{particles} = N_{partilces}\_max$. Use residual resampling when choosing the subset to be retained.
- Choose $\tilde{s}_n^{(i)}$ based on the trajectory with highest likelihood.

Table 11: Blind equalizer using nested SMC

- Initialization
- for $i = 1, \ldots, N_s$ /*for each particle*/
  - Draw $\tilde{s}_{n+k}^{(i)} \sim \beta(\cdot)$ as described in Table 10
  - Update

$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(\mathbf{r}_n - \mu_n^{(i)})$$

$$\mathbf{\Sigma}_n^{(i)} = (1 - K_n^{(i)})\mathbf{s}_n^{(i)^T}\mathbf{\Sigma}_{n-1}^{(i)},$$

    where the Kalman gain is:

$$K_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)}\mathbf{s}_n^{(i)}(P_n^{(i)})^{-1}$$

  - Update weight $w_n^{(i)} \propto w_{n-1}^{(i)}\beta_{n:n+L-1}^{(i)}(\cdot)$
- Obtain symbol estimate as

$$\hat{s}_n = \arg \max_{s_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)}\delta(s_n - s_n^{(i)}),$$

- Resample if needed
- Perturb channel particles and update based on likelihoods.

with σ_1^2 and so on.

For the $m = 2$ case we use equal power profile for each delay coefficient, i.e. $\sigma_1^2/\sigma_0^2 = 1 = 0dB$. We used $N = 300$ particles to estimate the channel state, and maintained $N\_max = 2$ trajectories for symbol estimates. Setting $N\_max = 4$ would be the full MLE solution. The proposed SMC algorithms outperform the Gibbs based algorithm. While the Gibbs equalizer exhibits an error floor, the proposed algorithms do not, because the SMC algorithm can track the state of dynamic systems. The nested SMC algorithm outperforms the partitioning approach because it keeps track of a few trajectories, instead of only one. Note that spectral diversity is preserved in this case as the slope of the BER plot is retained.

The genie aided SMC in Figure 10 is the nested SMC algorithm with perfect channel information. Its performance can easily be seen to converge to the MLE equalizer ($N\_max = 4$) as SNR is increased.

Using more particles to estimate the channel state does not justify the increase in computational cost. As it is seen in Figure 11, using 100 particles results in approximately only a $1dB$ loss.

The power profile of the $m = 3$ tap channel is chosen to represent a realistic channel, i.e. $\sigma_1^2/\sigma_0^2 = 0dB$, $\sigma_2^2/\sigma_0^2 = -6dB$. Comments similar to the above can be made about these plots as well (Figure 12). The nested SMC approach outperforms the other blind algorithms and diversity is preserved again.

In the case of $m = 4$ multipath delays, the power profile is $\sigma_1^2/\sigma_0^2 = 0dB$, $\sigma_2^2/\sigma_0^2 = -6dB$, $\sigma_3^2/\sigma_0^2 = -9dB$. The average power in the last path is very little, therefore significant spectral diversity gain is not expected (Figure 13).

To further analyze the ability of the SMC algorithm to take advantage of spectral diversity in severe ISI conditions, we set the communications channel to have equal average energies on each multipath: $\sigma_1^2/\sigma_0^2 = 0dB$, $\sigma_2^2/\sigma_0^2 = 0dB$, $\sigma_3^2/\sigma_0^2 = 0dB$. The nested SMC approach is observed to outperform the one based on temporal partitioning. However, for the rich spectral diversity case, the blind algorithms tend to lose their effectiveness in capturing all energy from the multiple delays, because the equalizer is unable to track all of the channel coefficients reliably well. The genie aided equalizer preserves the diversity

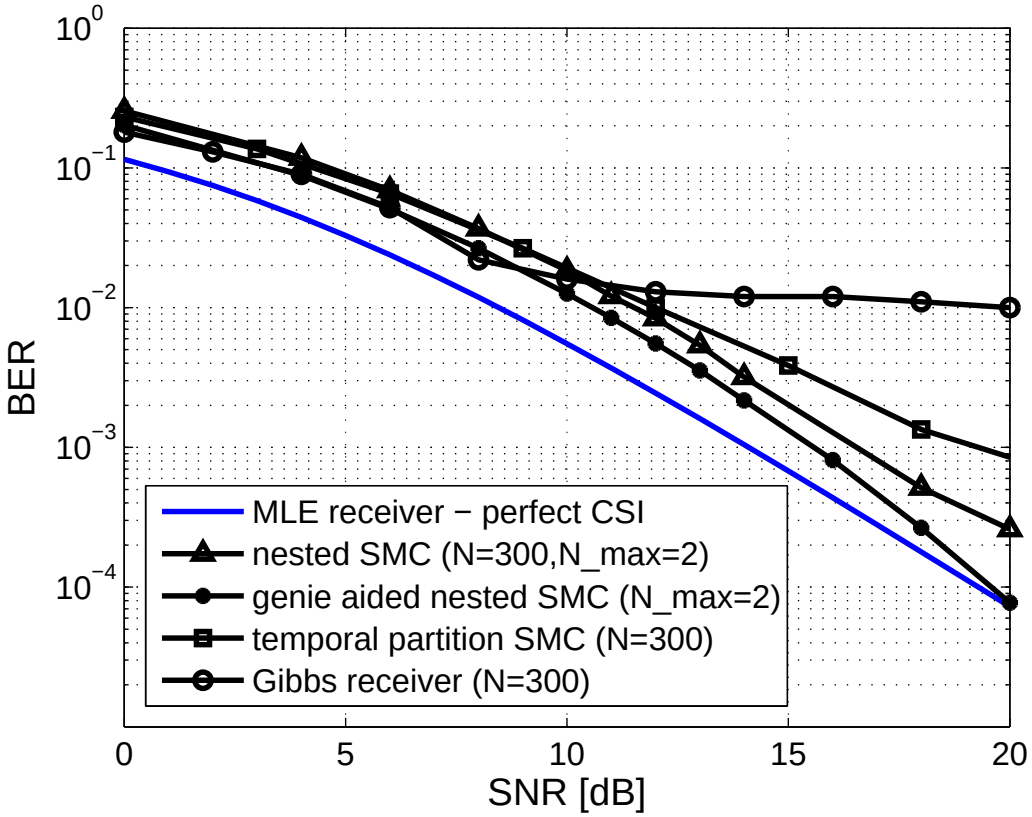


Figure 10: Performance of the SMC blind equalizer with comparison to the Gibbs equalizer, and to ML equalization.

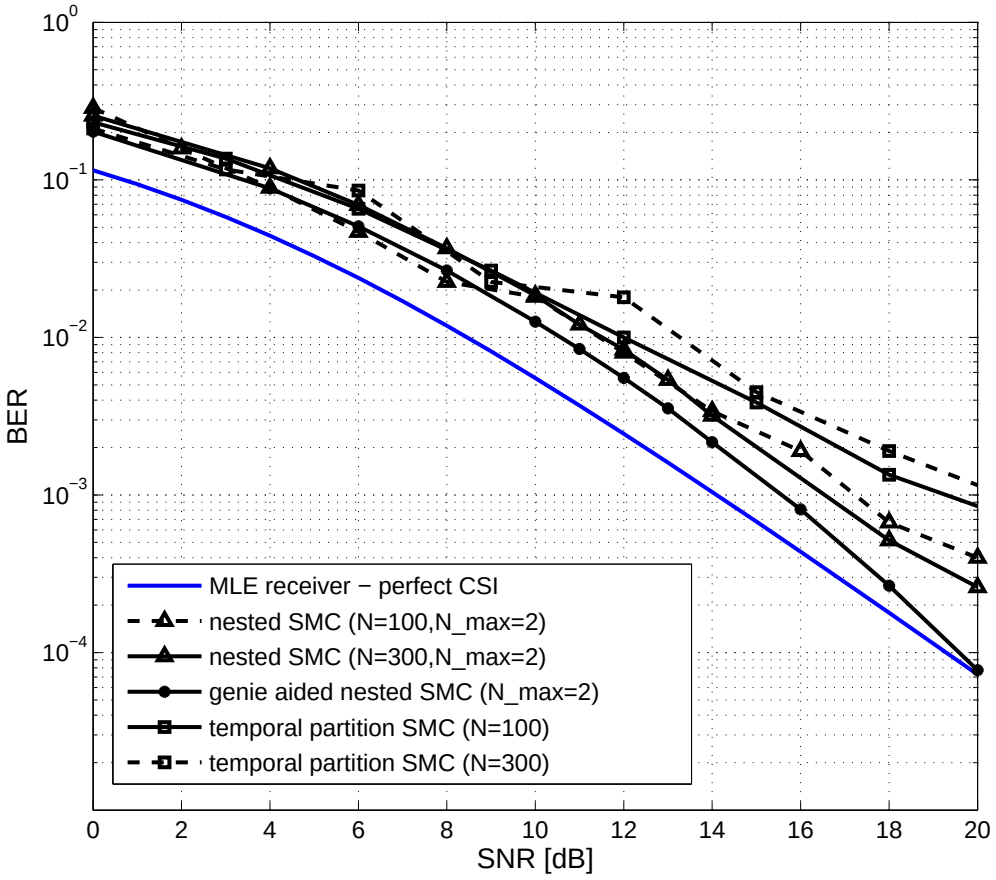
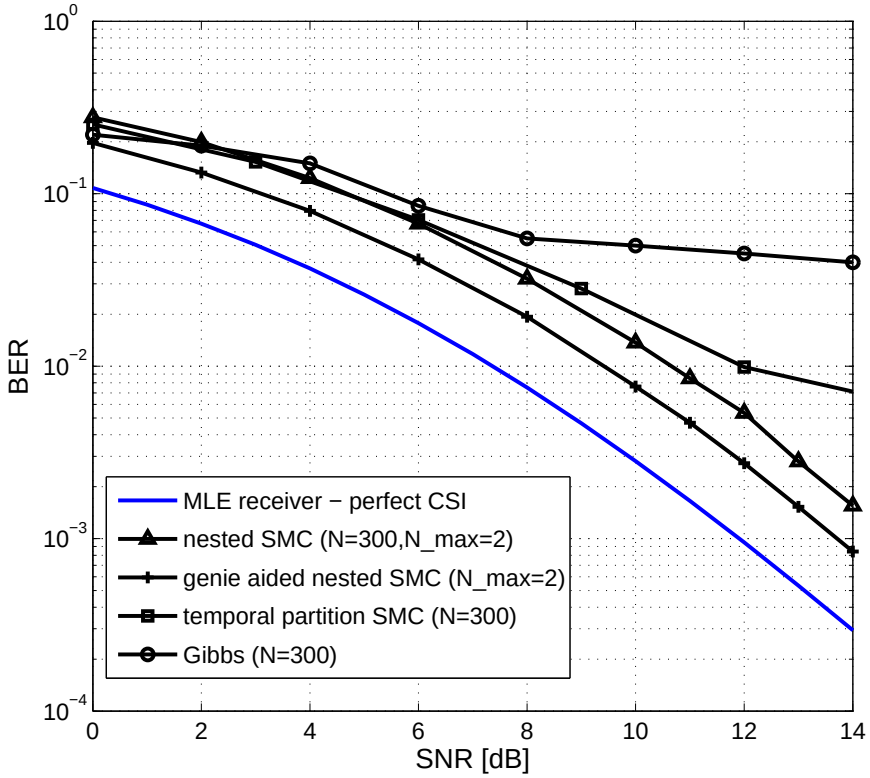
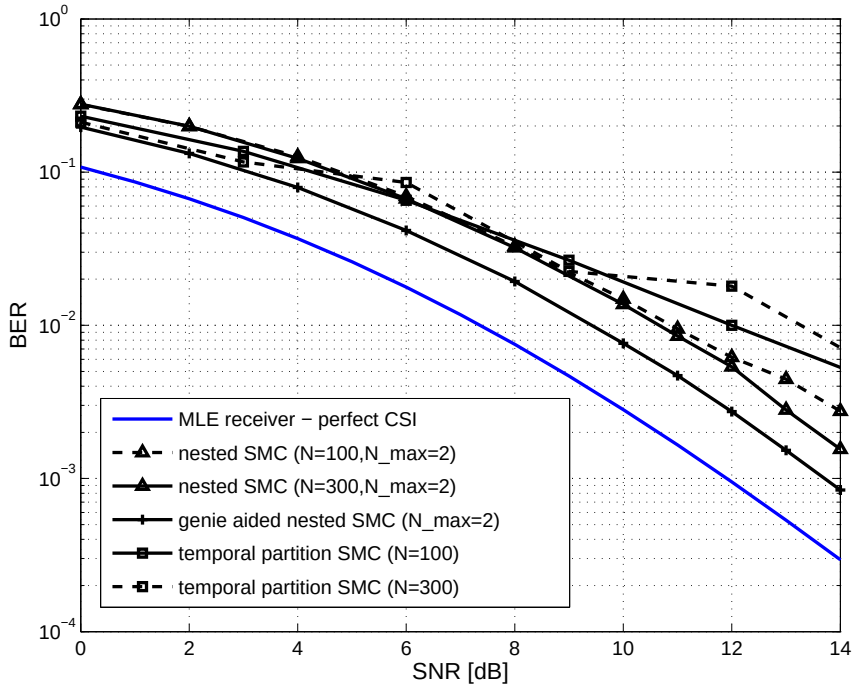


Figure 11: Effect of number of used particles on SMC algorithm performance.

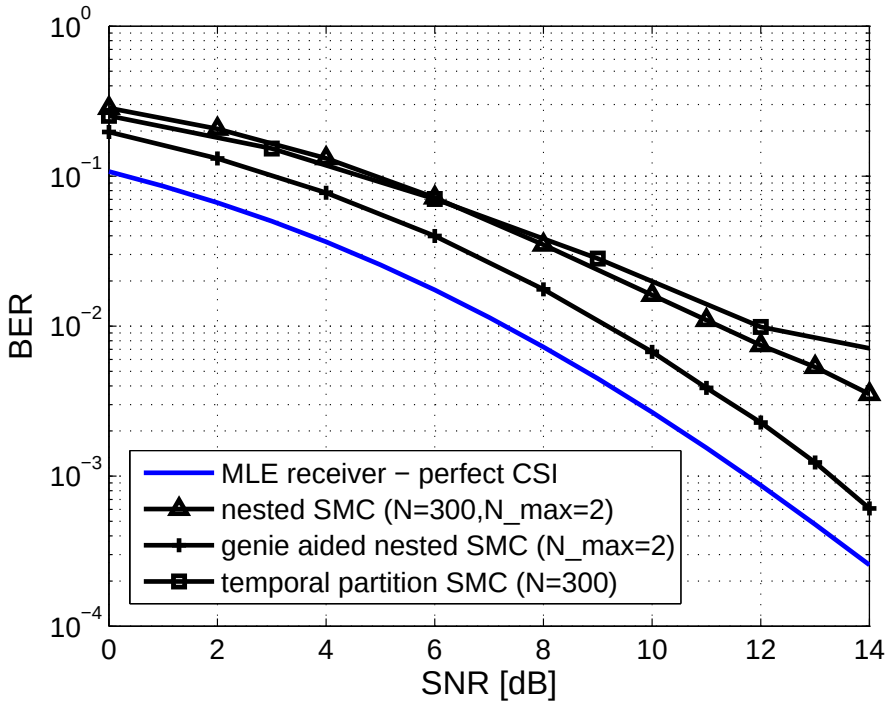


(a) Performance of SMC blind equalizer.

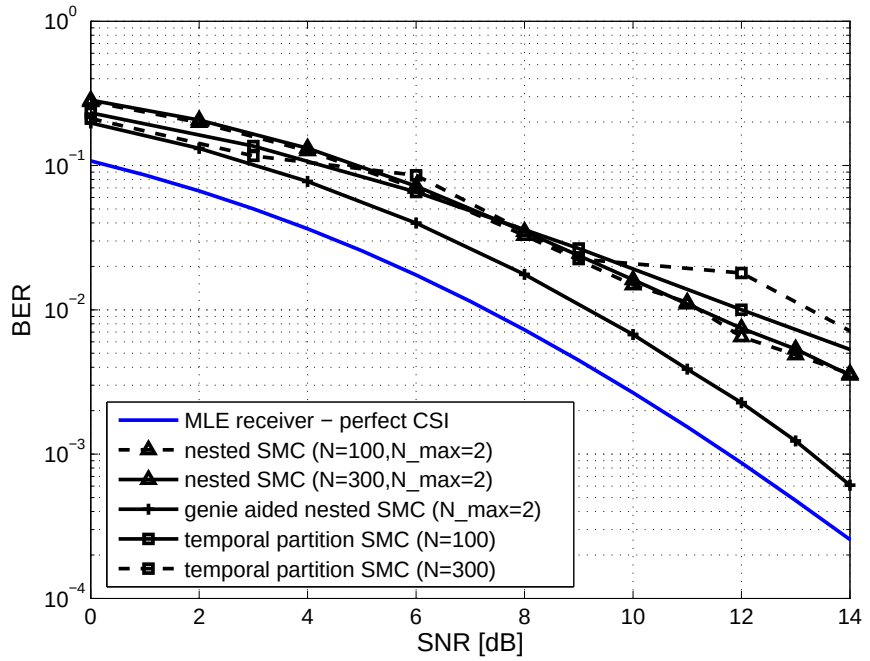


(b) Effect of number of used particles on performance.

Figure 12: Performance of SMC blind equalizer, $m = 3$ multipath channel



(a) Performance of SMC blind equalizer.



(b) Effect of number of used particles on performance.

Figure 13: Performance of SMC blind equalizer, $m = 4$ multipath channel

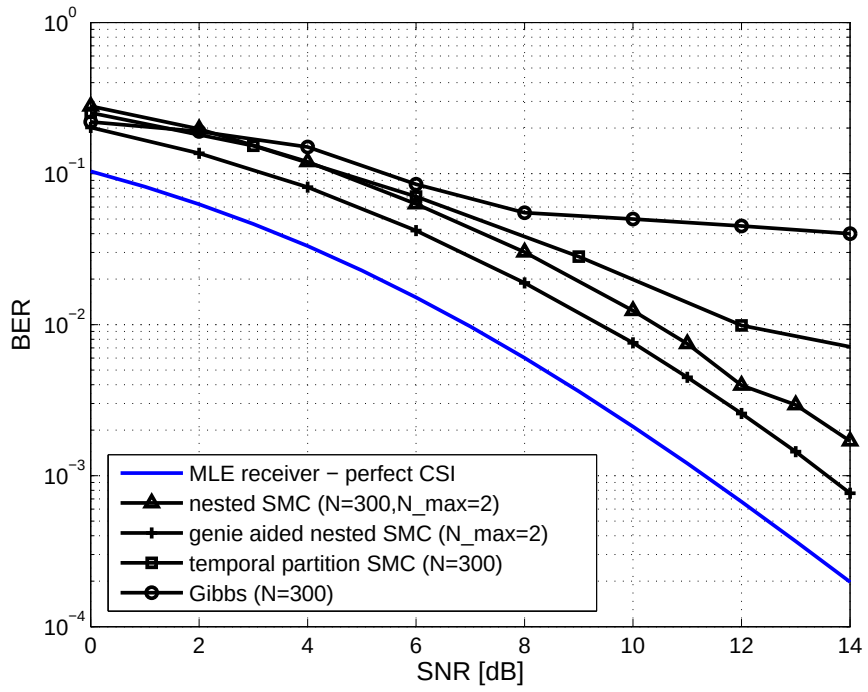
order firmly by retaining only 2 symbol trajectories. However, the blind equalizers cannot obtain reliable estimates for the channel state.

3.13 Conclusion

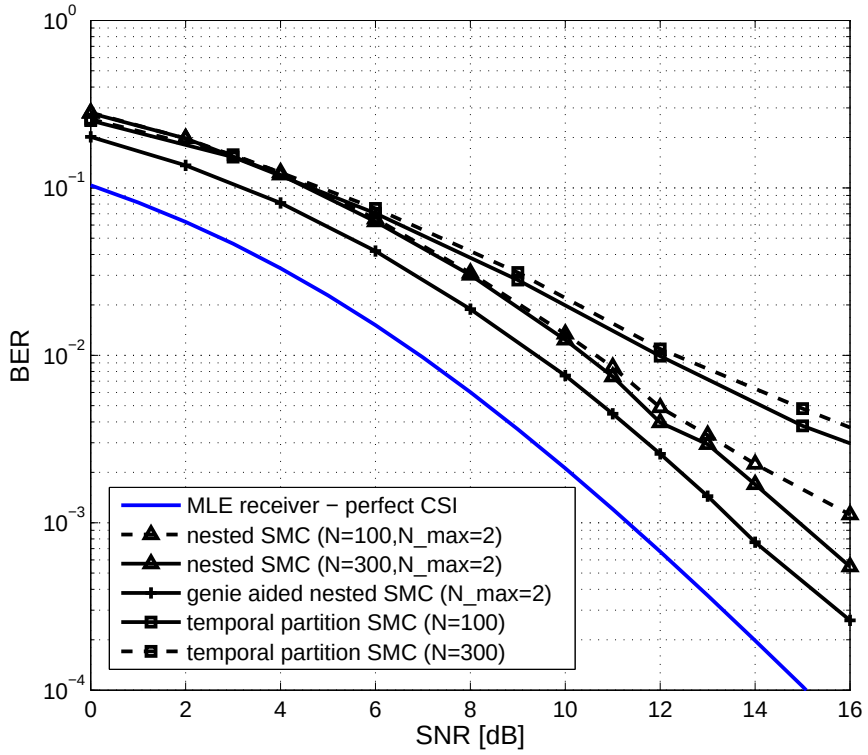
We studied Bayesian methods for blind equalization. We constructed a signal model and communications channel model. As in all equalizers, the goal is to obtain reliable estimates for the transmitted symbols, eliminate ISI and possibly achieve some spectral diversity gain. We defined the Bayesian problem and formulated the posterior distribution definitions. We noted the recursive Markovian structure and also defined the recursive incremental weight update for the SMC algorithm.

We outlined the standard SMC algorithm that solves the equalization problem. Then we realized that because of the Markovian structure of the underlying densities, the whole symbol space need not be sampled each time. Instead, we partitioned the symbol space recursively. In the following section we introduced an alternative way to obtain symbol samples. By taking advantage of the recursive posteriors, we applied a SMC algorithm to draw the symbol samples. The particles of the this SMC stored information about the evolution of symbol trajectories.

In the results section we displayed the simulation results and concluded that the SMC algorithm is a very effective algorithm when applied to blind equalization. It outperforms the other blind MCMC algorithms significantly. Even with the simplifications to the symbol space sampling approaches, it outperforms MCMC algorithms and is very close to the regular SMC solution approach, but with significantly reduced computational complexity. The complexity is reduced from being exponential in the path delay spread, to only quadratic.

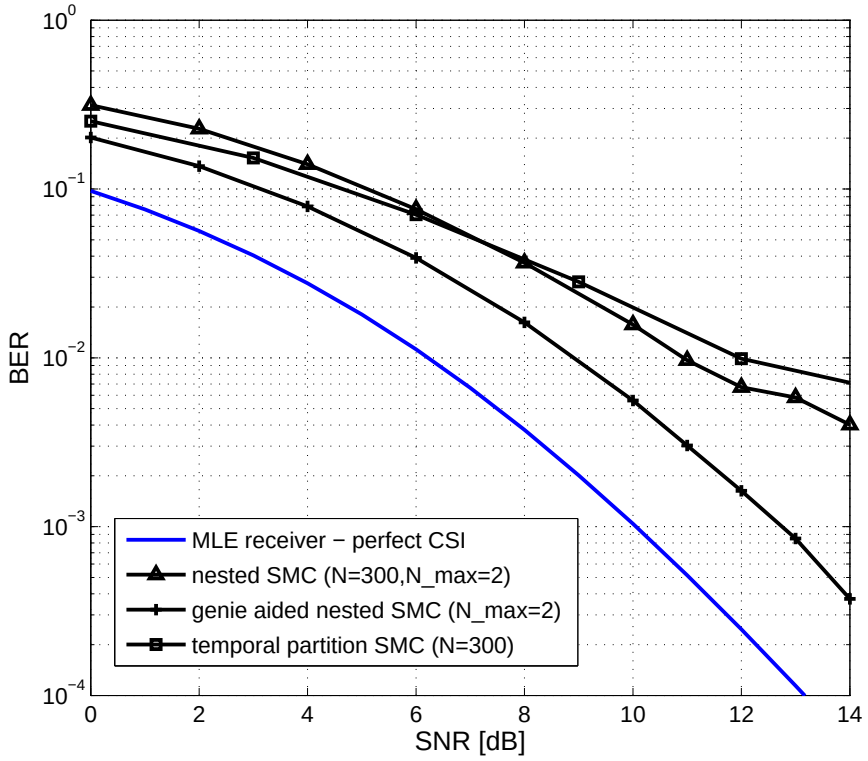


(a) Performance of SMC blind equalizer.

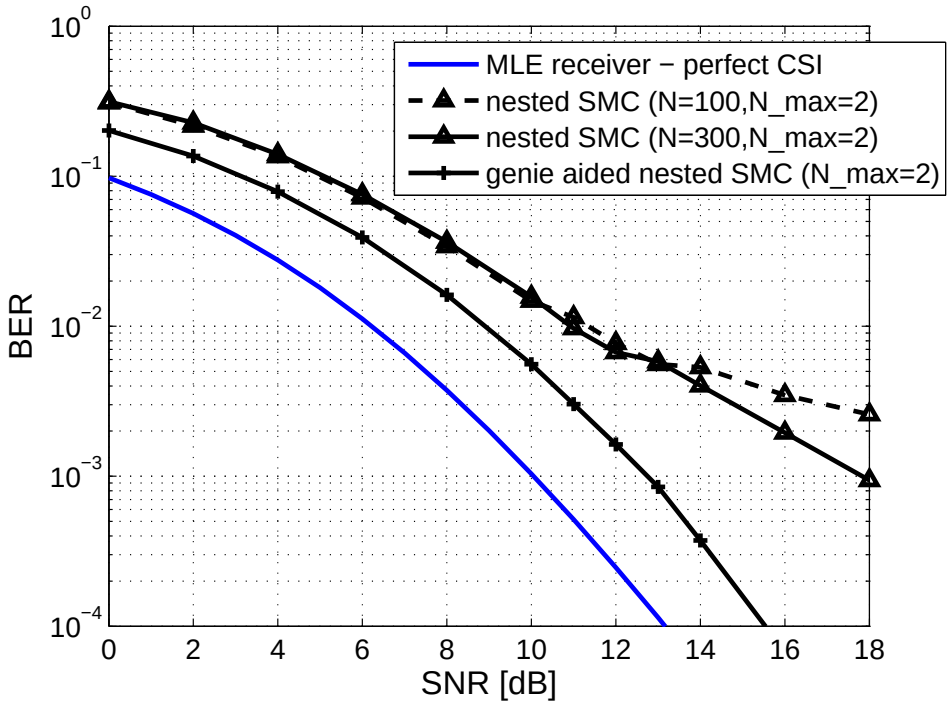


(b) Effect of number of used particles on performance.

Figure 14: Performance of SMC blind equalizer, $m = 3$ multipath channel



(a) Performance of SMC blind equalizer.



(b) Effect of number of used particles on performance.

Figure 15: Performance of SMC blind equalizer, $m = 4$ multipath channel

CHAPTER IV

A SEQUENTIAL MONTE CARLO ALGORITHM FOR MIMO SYSTEMS

In this work we propose an SMC algorithm for Multiple-Input Multiple-Output system receivers. The algorithm estimates the transmitted MIMO symbol vectors without knowledge of the channel state information. In addition, temporal partitioning of the symbol space is applied to decrease the computational complexity of the algorithm. The performance of the algorithm is compared to the performance of a receiver with perfect channel state information. Spatial diversity is shown to be preserved.

4.1 Introduction

Symbol detection in multiple input multiple output (MIMO) systems has been researched significantly [23, 24]. Maximum likelihood sequence estimator receivers perform optimally; however, their computational complexity is very high. The parameter space grows exponentially with the number of transmit antennas and the symbol constellation size. Interference cancelation and nulling techniques have been applied to decrease the dimensionality of the search space [23, 24]. Namely, the symbol from the antenna with the strongest signal is detected, and then its effect is subtracted from the received signal. The technique is similar to SIC described in the introduction chapter in this dissertation. Foschini and Gans showed that capacity increases linearly with the lesser of N_t, the number of transmit antennas, and N_r, the number of receive antennas number [24].

SMC algorithms have been applied in the design of receivers for MIMO systems. Many proposed designs assume that channel state information is present at the receiver or obtain some estimate by using a training symbol sequence. The SMC architecture in these cases is utilized to take advantage of the artificially introduced sequential processing of serial interference cancelation and nulling. This technique is applied at each layer of symbol

detection, and symbol estimates are refined as the algorithm proceeds from one layer to the next [21, 43].

In this work, we apply an SMC algorithm in the design of a blind receiver for a MIMO system. The receiver obtains and refines its knowledge of the channel state as more symbols are received. The nature of SMC allows for the receiver to track changes in the channel coefficients. We use spatial partitioning of the symbol space to decrease the sampling space and, consequently, decrease the computational complexity of the algorithm.

4.2 System Model

We consider a flat Rayleigh fading MIMO system. The transmitter has N_t transmit antennas, and, accordingly, its bit stream is split into N_t substreams. After encoding and mapping, the transmit symbols are fed into N_t antennas.

The receiver has N_r antennas. For the symbols to be properly detected, we require that $N_r \geq N_t$. The received signal sampled at the symbol rate can be represented by the standard MIMO received signal equation:

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \eta, \tag{113}$$

where $\mathbf{s} = [s_1, \ldots, s_{N_t}]^T$ is the transmitted symbol vector. The matrix $\mathbf{H}$ is a $N_r \times N_t$ matrix that contains the complex gains of the channels from the N_t transmit antennas at the transmitter to the N_r receive antennas at the receiver. We assume a flat Rayleigh fading channel. Thus, the elements of $\mathbf{H}$ are i.i.d. zero-mean, complex Gaussian random variables. The noise term η is a zero mean complex additive white Gaussian noise vector with covariance $\sigma^2\mathbf{I}_{N_r}$.

4.3 Channel Model

With this representation we assume that the channel coefficients represent the state of a continuous Markov process. Therefore, the channel can be modeled as a Markov chain with transition kernel $q_1(\cdot|\cdot)$. In reality this is acceptable because the communications channel is a slow-fading, flat Rayleigh channel. The autocorrelation function of the coefficients of

such a channel can be described by a zeroth order Bessel function of the first kind [72]:

$$\phi_{h[l]h[l]}(\tau) = \frac{1}{2} J_0(2\pi f_m \tau) \tag{114}$$

The symbol τ, above, is the time delay, and f_m is the Doppler frequency shift.

The power spectral density (psd) of this process is a non-rational function

$$S_{h[l]h[l]}(f) = \begin{cases} \frac{1}{4\pi f_m} \frac{1}{\sqrt{1-(f_m/f)^2}}, & |f| \leq f_m \\ 0, & \text{otherwise} \end{cases} \tag{115}$$

and can be estimated by an autoregressive moving average process of high order. However, the impulse response of such a process is long and increases computational complexity significantly [72].

Each element of the channel matrix $\mathbf{H}$ is a random process that satisfies the above autocorrelation and psd equations. Each element can be approximated by a Markov process, or equivalently, as a first order autoregressive model (equation (116)), or a Markov process.

$$\mathbf{H}_n = \zeta \mathbf{H}_{n-1} + (1-\zeta)\mathbf{W}_{n-1}, \tag{116}$$

where $\mathbf{W}_n$ is a matrix with the same dimension as $\mathbf{H}$ and each of its element is an iid complex Gaussian random process with zero mean and variance 1, and ζ is defined as

$$\zeta = 2 - cos(2\pi f_m T) - \sqrt{(2 - cos(2\pi f_m T))^2 - 1}. \tag{117}$$

Essentially, the Bessel function is approximated by an exponential function.

4.4 Symbol Prediction

No correlation among transmitted matrix symbols is assumed. Therefore, we model the symbols as an uncorrelated uniform distributed discrete random process

$$p(\mathbf{s}_n | \mathbf{s}_{n-1}) \propto 1, \qquad \forall \mathbf{s}_n, \mathbf{s}_{n-1}. \tag{118}$$

4.5 Sequential Monte Carlo

The system can be represented with the following state-space representation:

$$\begin{aligned} \mathbf{h}_n &\sim q_1(\mathbf{H}_n|\mathbf{H}_{n-1}) \\ \mathbf{S}_n &\sim q_2(\mathbf{s}_n|\mathbf{s}_{n-1}) \\ \mathbf{r}_n &\sim f(\mathbf{r}_n|\mathbf{s}_n, \mathbf{H}_n) \end{aligned} \tag{119}$$

4.6 Brute force detection algorithm

The SMC algorithm approximates the posterior distribution by sampling and evaluating at discrete points in the state space. Each sample is called a particle and, along with the assigned weights, the set $\{s_n^{(i)}, w_n^{(i)}\}_{i=1}^{N_s}$ is called a random measure that characterizes the a posteriori distribution $p(s_{0:n}|r_{0:n})$.

The particles are assigned weights in order to make them appear as if they are drawn from the desired posterior distribution. If the posterior density is chosen for the symbol draws, the incremental weight update simplifies:

$$\begin{aligned} w_n^{(i)} &= w_{n-1}^{(i)} \frac{p(\mathbf{s}_{0:n-1}^{(i)}|r_{0:n})}{p(\mathbf{s}_{0:n-1}^{(i)}|r_{0:n})} \\ &\propto w_{n-1}^{(i)} \int p(r_n, \mathbf{H}_n|r_{0:n-1}, \mathbf{s}_{0:n-1}^{(i)}, \mathbf{H}_{0:n-1}) p(\mathbf{H}_n|\mathbf{H}_{0:n-1}) \mathrm{d}\mathbf{H}_n \\ w_n^{(i)} &\propto w_{n-1}^{(i)} u_n^{(i)} \end{aligned} \tag{120}$$

The algorithm first updates the channel state according to the Markov transition probability for all particles. Next, a symbol is drawn for each particle from the posterior distribution, conditioned on the predicted channel coefficients. Then, based on the symbol outcome and the received signal, the channel state is updated and a weight for each particle is assigned.

Rao-Blackwellization is used for the channel coefficient estimation part of the problem. Because of the Gaussian noise and linearity of the state update and observation equations, in the observation, the estimates for the channel coefficients can also be viewed as Gaussian random variables. Gaussian random variables are fully characterized by their means and

covariances. Therefore, instead of obtaining Monte Carlo estimates of the channel coefficients, we approach the problem differently. We obtain analytic expressions for the mean and the covariance of the channel estimates. Furthermore, we obtain expressions that perform incremental updates from the previous state. These equations, in fact, are the Kalman filter equations.

The algorithm is summarized in Table 12. In the table N_s denotes the number of particles, and σ^2 is the observation noise variance. The symbol Σ_n indicates the covariance matrix of the channel coefficient estimates and is also an indicator of the uncertainty of the estimate. Therefore, at initialization, Σ_n is assigned to be a diagonal matrix with very large numbers on the diagonal.

This algorithm requires the computation of symbol likelihoods from a multidimensional space. For example in the case of signaling in $16 - QAM$ space and utilizing 8 transmit antennas, 2^{32} possible transmitted symbols must be considered. In addition, obtaining uncorrelated sample realizations from such a multidimensional distribution is extremely computationally intensive. A Gibbs sampler with a long burn-in period must be utilized to obtain such samples. In the results section, we do not provide results for the full complexity algorithm, because it is computationally intractable to simulate.

4.7 An SMC approach for sampling the symbol space

The BLAST techniques utilize a back-substitution approach for obtaining reliable estimates of the transmitted symbols. They, however, suffer from error propagation problems and low spatial diversity. The reason for the error propagation problems is that those algorithms do not provide methods for altering a previously made decision. Low spatial diversity results from the inability to use sufficient statistics in the decision for a given symbol. The latter will become clearer in the following sections.

Let us multiply the received signal vector with the Hermitian of the channel matrix.

$$\tilde{\mathbf{r}} = \mathbf{H}^H \mathbf{H}\mathbf{s} + \tilde{\eta}, \tag{122}$$

where $\tilde{\eta}$ is Gaussian noise with covariance matrix $E\{\mathbf{H}\mathbf{H}^H\}\sigma^2$.

Table 12: Temporally Partitioned SMC for Frequency Selective Fading Channels

---

- Initialization
- for $i = 1, \ldots, N_s$
  - Compute
$$\mathbf{H}_n^{(i)} = \zeta \mathbf{H}_{n-1}^{(i)} + (1-\zeta)\mathbf{W}_n$$
$$\mu_n^{(i)} = \mathbf{H}_n^{(i)} \mathbf{s}_n^{(i)}$$
  - Let $\mathbf{h}_n^{(i)} = \mathrm{vect}(\mathbf{H}_n^{(i)})$, and construct $\mathbf{S}_n^{(i)}$ so that $\mathbf{S}_n^{(i)} \mathbf{h}_n^{(i)} = \mathbf{H}_n^{(i)} \mathbf{s}_n^{(i)}$
$$P_n^{(i)} = \sigma^2 \mathbf{I} + \mathbf{S}_n^{(i)} \mathbf{\Sigma}_n^{(i)} {\mathbf{S}_n^{(i)}}^H$$
    ,for each possible $\mathbf{s}_n^{(i)} \in \{\pm 1\}^{N_t}$
  - Compute $\beta_n^{(i)}(\cdot) = p(r_n | \mathbf{H}_n^{(i)}, \mathbf{s}_n^{(i)})$
  - Draw $\hat{\mathbf{s}}_n^{(i)} \sim \beta_n^{(i)}(\cdot)$
  - Construct $\hat{\mathbf{S}}_n^{(i)}$
  - Update
$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)} (\mathbf{r}_n - \mu_n^{(i)})$$
$$\mathbf{\Sigma}_n^{(i)} = (\mathbf{I} - K_n^{(i)} \hat{\mathbf{S}}_n^{(i)}) \mathbf{\Sigma}_{n-1}^{(i)},$$
    where the Kalman gain is:
$$K_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)} \hat{\mathbf{S}}_n^{(i)H} (P_n^{(i)})^{-1}$$
  - Update weight $w_n^{(i)} \propto w_{n-1}^{(i)} \beta_{n:n+L-1}^{(i)}(\cdot)$
- Obtain symbol estimate as
$$\hat{\mathbf{s}}_n = \arg\max_{\mathbf{s}_n} \sum_{i=1}^{N_s} w^{(i)} \delta(\mathbf{s}_n - \hat{\mathbf{s}}_n^{(i)}), \tag{121}$$
- Resample if needed
- Perturb channel particles and update based on likelihoods.

4.7.1 Zero Forcing

The product $\mathbf{H}^H\mathbf{H}$ is a positive definite Hermitian matrix. Therefore, it has a unique Cholesky factorization $\mathbf{U}_{ZF}^H\mathbf{U}_{\mathbf{ZF}} = \mathbf{H}^H\mathbf{H}$, where $\mathbf{U}_{ZF}$ is an upper triangular matrix. If we multiply the equation with U_{ZF}^{-H} we find:

$$\mathbf{y} = \mathbf{U}_{ZF}\mathbf{s} + \nu, \tag{123}$$

where ν is white noise with covariance matrix $\sigma^2\mathbf{I}_{N_t}$. The inverse of a triangular matrix can be easily computed using back-substitution and its computational burden is negligible.

To make the equations more readable and understandable, we flip the matrices upside-down so that the upper-triangular matrix becomes lower triangular. The flipping corresponds to a permutation of the antennas and does not change the problem. The subscripts in the equations below denote the index of a vector or a matrix. The colon (:) operator in $m : n$, for example, refers to integers in the $[m, n]$ range, including the bounds.

The above equation can be solved for $\mathbf{s}$ easily by back-substitution, which is a very commonly seen decision feedback scheme. Namely,

$$\hat{s}_1 = \frac{y_1}{U_{1,1}} \tag{124}$$

$$\hat{s}_2 = \frac{y_2 - U_{2,1}Q(\hat{s}_1)}{U_{2,2}} \tag{125}$$

$$\hat{s}_3 = \frac{y_3 - U_{3,1}Q(\hat{s}_1) - U_{3,2}Q(\hat{s}_2)}{U_{3,3}} \tag{126}$$

and so on and so forth. The $Q(\cdot)$ function above is a quantization function that maps the soft symbol $\hat{s}$ to the nearest symbol in the constellation. This particular approach is equivalent to the zero forcing (ZF) V-BLAST approach [23–25].

4.7.2 MMSE

The MMSE approach can be obtained by applying the MMSE solution to the received signal,

$$\hat{\mathbf{s}} = \mathbf{R}_{MMSE}^{-1}\mathbf{H}^H(\mathbf{H}\mathbf{s} + \eta) \tag{127}$$

with $\mathbf{R}_{MMSE} = \mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}$.

The above equation is a linear MIMO detector. The symbol estimate is soft and needs to be quantized to the nearest symbol in the constellation. This equation cannot be solved efficiently like the ZF approach by triangularizing the equation. However, it has been reported that it achieves better performance than the ZF V-BLAST, which is also a linear approach [23–25].

4.7.3 SMC

A drawback to the ZF approach is error propagation. There is no mechanism in the algorithm that can correct a previous decision. In addition, the information passed during the back-substitution process is hard information. Any information about the certainty of a particular decision is lost.

Moreover, sufficient statistic are not utilized fully when making a decision about a symbol. For example, the sufficient statistics for the symbol s_{N_t} is the whole vector $\mathbf{y}$. However, only the last element is used in making a decision about s_{N_t}, resulting in spatial diversity loss.

These two drawbacks could be dealt with by using a sequential Monte Carlo algorithm. Using the structure in equation (123), instead of setting symbols fixed at each stage (i.e. layer) and letting wrong decisions propagate, a number of nearly equiprobable choices could be stored for processing in later stages. Beginning at the first layer, there would be 16 choices for the first symbol (assuming 16-QAM signaling.) In the next stage, the number of possible choices for the combination of the first and second symbol would be 256. Keeping track of all possible choices is equivalent to the ML solution. An alternative is to limit the number of stored symbol states to a fixed number.

This procedure can be illustrated as an expanding trellis of states (Figure 16). Each state represents the detected symbol state. Because of the tree structure, the path to the root is unique. Each branch is labeled with the conditional posterior symbol density. To obtain the probability of a symbol, all probabilities along the branches leading back to the root must be multiplied. In order to avoid storage and computational issues, some paths with low probabilities are terminated and discarded.

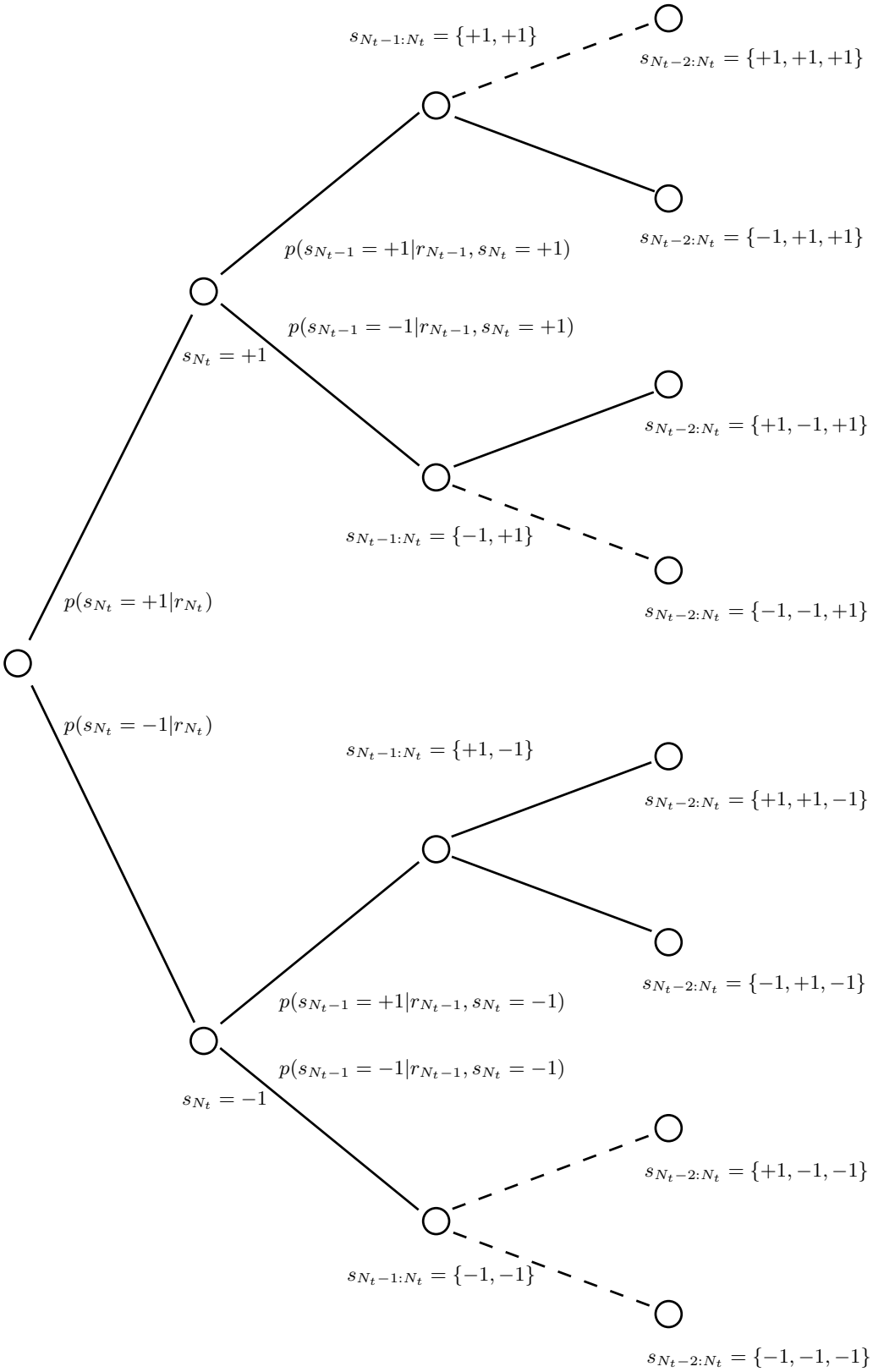


Figure 16: Symbol state trajectory.

Such a recursive formulation is essential in SMC algorithms, as it simplifies the calculations significantly. Let us formulate the SMC algorithm that would obtain a symbol estimate $\hat{\mathbf{s}}$ from the observation $\mathbf{r}$. The structure of the algorithm is presented in Table 13. The symbol state space is expanded. At each recursion the the number of particles increases, as is seen in Figure 16. Then, a weight for each particle is computed and resampling is performed when necessary. The process is repeated until the whole symbol space is covered, and at the end a soft estimate about the transmitted symbol is obtained.

Table 13: SMC algorithm for symbol draw

- for $i = N_t : -1 : 1$
  - Calculate likelihoods for each possible particle trajectory
  - Obtain samples from the space based on the computed likelihoods
  - Update weights
  - Resample if necessary
- Obtain a symbol estimate according to the most likely particle trajectory

Because of the Markovian structure of the densities we write the following factorization:

$$p(s_{1:n}|r_{1:n}) = p(s_n|r_{1:n}, s_{1:n-1})p(s_{1:n-1}|r_{1:n}) \tag{128}$$

The proposal density that is used to obtain samples from the symbol space is denoted by $q(\cdot)$. The weights of the drawn samples are computed by the familiar equation

$$w_n^{(i)} = \frac{p(s_{1:n}^{(i)}|r_{1:n})}{q(s_{1:n}^{(i)}|r_{1:n})}. \tag{129}$$

After applying the recursion above we get

$$w_n^{(i)} = w_{n-1}^{(i)} \frac{p(s_{1:n}^{(i)}|r_{1:n})}{q(s_{1:n-1}^{(i)}|r_{1:n-1})q(s_n^{(i)}|s_{1:n-1}^{(i)}, r_{1:n})}. \tag{130}$$

If the posterior density is used to sample from the symbol space, the incremental weight update simplifies even further. Since the symbol space is a discrete space, exact evaluation of the probability mass functions is possible. Approximations to this density are needed

only in cases where the symbol constellation size is very large and computation of entire distribution is not practical.

We set $q(\cdot) = p(\cdot)$ and find

$$w_n^{(i)} = w_{n-1}^{(i)} p(r_n | s_{1:n-1}^{(i)}, r_{1:n-1}). \tag{131}$$

The weights are sufficient to characterize the underlying densities up to a normalizing constant. Therefore, normalization on every recursion of the algorithm is necessary.

We improve the SMC algorithm described in Table 13 by taking advantage of the discrete state space. We calculate the exact posterior distribution and obtain samples from the entire symbol space at each processing layer. In order to keep the number of particles low, we retain only a subset with high likelihoods. Those particles are chosen based on a modified version of the residual resampling method.

The number of particles to be kept is set to N_max. From the expanded list of particles we choose the particles with normalized weight greater than $1/N\_max$. These particles are not replicated as in the regular residual resampling method. Replication would result in wasted computation, as the state-space is discrete. The rest of the spots to fill the necessary N_max space are filled by particles chosen randomly from the rest of the particles. Here care must be taken not to pick the same particle more than once, for the reason pointed to above.

We summarize the modified SMC algorithm based on the discrete space improvements in Table 14.

4.8 Blind MIMO detection with SMC

In this section we incorporate the symbol draw algorithm within the channel coefficient estimation algorithm. The structure of the algorithm is similar to the full complexity SMC version. The difference is in the triangularization of the received signal equations.

It is significantly easier for the particle filtering algorithm to obtain symbol estimates from a triangular[1] equation system than to obtain symbol estimates from a general matrix

[1]Please remember that we permute the rows so that the matrix is lower triangular.

Table 14: SMC algorithm for symbol draw

- Set $N\_cur\_particles = 1$
- for $k = 1 : N_t$
  - Expand symbol space, i.e. Set $N\_cur\_particles = N\_cur\_particles \times alphabet\_size$
  - Calculate likelihoods for each possible particle trajectory branch

    $$p(y_k | s^{(i)}_{1:k-1}, r_{1:k-1}),$$

    by using the equation from the kth row of

    $$\mathbf{y} = \mathbf{U}\mathbf{s} + \nu.$$

    The estimate for the lower triangular matrix $\mathbf{U}$ is obtained elsewhere.
  - Extend trajectories for all possible transmitted symbols
  - Update weights for each trajectory
  - If $N\_cur\_particles > N\_max$
    - Select particles with normalized weight greater than $1/N\_max$
    - Draw particles with probability proportional to their weights from the rest of the unselected choices
    - Set $N\_cur\_particles = N\_max$
- Obtain a symbol estimate corresponding to the most likely particle trajectory

equation. By taking this approach we avoid sampling from the entire symbol space and process the data layer by layer.

4.9 Simulation Results

We simulate this algorithm in systems with 4×4 and 8×8 channel matrices. To illustrate the performance in large symbol spaces we utilize both QPSK and 16-QAM constellations. The channel is a flat Rayleigh fading channel with a Doppler coefficient of $f_m T = 1e-3$. Such a channel is usually described as a slow fading flat channel.

In Figure 17 we compare the performance of the SMC receiver to other receivers. It is clear that the proposed receiver outperforms the standard ZF-VBLAST and MMSE-VBLAST techniques with full CSI. V-BLAST algorithms degrade due to their inherent error propagation effect. The Gibbs sampler based receiver does not perform well because the receiver cannot account for dynamically changing channel states. The used algorithm utilizes the Markovian structure of posterior densities and dramatically reduces the size of the symbol space to be sampled. This reduction results in approximately $1dB$ of performance degradation. If we add the effect of having an unknown channel state, the performance degradation grows another $\sim 3dB$, however, which is still significantly better than the VBLAST techniques. The performance difference of utilizing 100 vs 300 particles for the channel estimates is minimal - only $0.5dB$. Similar results are observed in the 16 - QAM scenario (Figure 18). We were unable to perform a full ML-simulation, as the size of the symbol space is 16^4. However, having full CSI and increasing the number of traced symbol trajectories to 16 makes a big difference. In addition to the coding gain, there is also a spatial diversity gain realized. The latter is a result of the ability to resolve spatial diversity by keeping track of more symbol trajectories.

In Figure 19, the case of spatial correlation among the antennas is illustrated. The cross-correlation coefficient is set to $\rho = 0.9$, and the effect of spatial diversity is reduced. At low SNRs, the receiver is unable to track the channels during deep fades. Therefore, the performance is very degraded. At high SNR, the performance approaches the known channel scenario.

Table 15: Blind MIMO detector using SMC

- Initialization
- for $i = 1, \ldots, N_s$
  - Compute
$$\mathbf{H}_n^{(i)} = \zeta \mathbf{H}_{n-1}^{(i)} + (1-\zeta)\mathbf{W}_n$$
  - Obtain the Cholesky factorization $\mathbf{U}^H\mathbf{U} = \mathbf{H}^H\mathbf{H}$.
  - Obtain $\mathbf{y} = \mathbf{U}^{-H}\mathbf{H}^H\mathbf{r}$
  - Use the modified SMC algorithm in Table 14 to obtain an estimate $\hat{\mathbf{s}}_n^{(i)}$.
  - Let $\mathbf{h}_n^{(i)} = \mathrm{vect}(\mathbf{H}_n^{(i)})$, and construct $\hat{\mathbf{S}}_n^{(i)}$ so that $\hat{\mathbf{S}}_n^{(i)}\mathbf{h}_n^{(i)} = \mathbf{H}_n^{(i)}\hat{\mathbf{s}}_n^{(i)}$
  - Compute
$$\mu_n^{(i)} = \mathbf{H}_n^{(i)}\hat{\mathbf{s}}_n^{(i)}$$
$$P_n^{(i)} = \sigma^2\mathbf{I} + \hat{\mathbf{S}}_n^{(i)}\mathbf{\Sigma}_n^{(i)}\hat{\mathbf{S}}_n^{(i)H}$$
  - Update
$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(\mathbf{r}_n - \mu_n^{(i)})$$
$$\mathbf{\Sigma}_n^{(i)} = (\mathbf{I} - K_n^{(i)}\hat{\mathbf{S}}_n^{(i)})\mathbf{\Sigma}_{n-1}^{(i)},$$
    where the Kalman gain is:
$$K_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)}\hat{\mathbf{S}}_n^{(i)H}(P_n^{(i)})^{-1}$$
  - Update weight $w_n^{(i)} \propto w_{n-1}^{(i)}\beta_n^{(i)}(\cdot)$, where $\beta_n^{(i)}(\cdot)$ is the likelihood of the symbol trajectory computed in Table 14.
- Obtain symbol estimate as
$$\hat{\mathbf{s}}_n = \arg\max_{\mathbf{s}_n} \sum_{i=1}^{N_s} w^{(i)}\delta(\mathbf{s}_n - \hat{\mathbf{s}}_n^{(i)}), \tag{132}$$
- Resample if needed

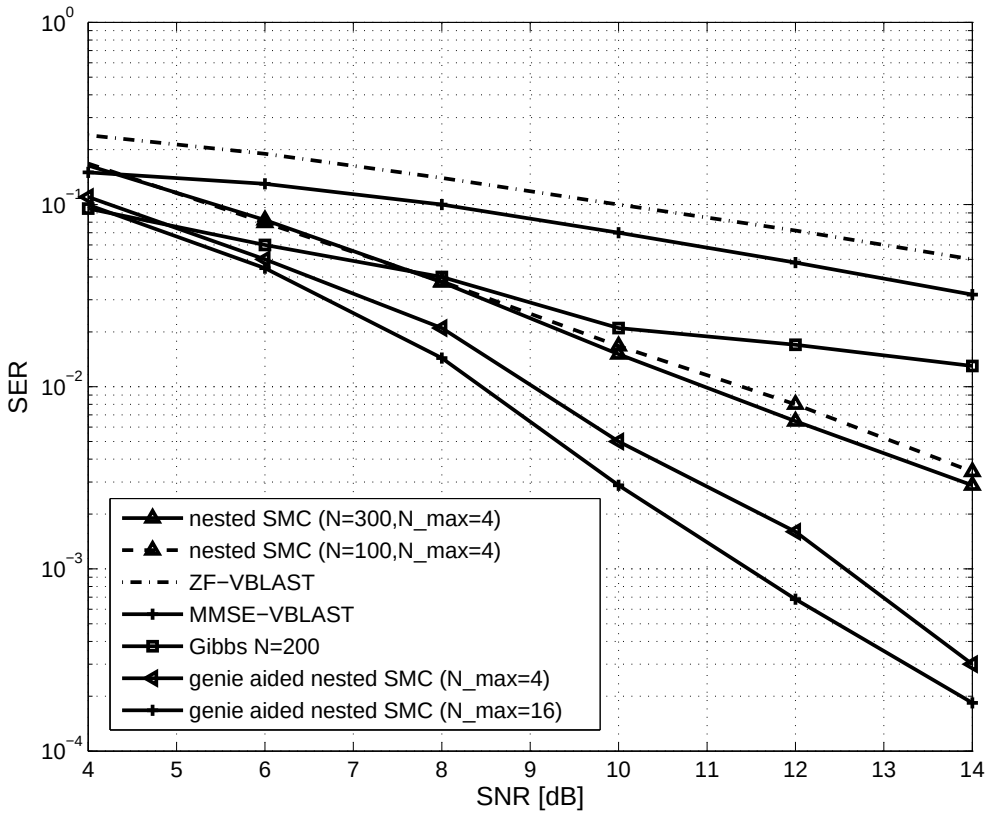


Figure 17: Performance of blind SMC MIMO equalizer with 4×4 antennas and QPSK signaling

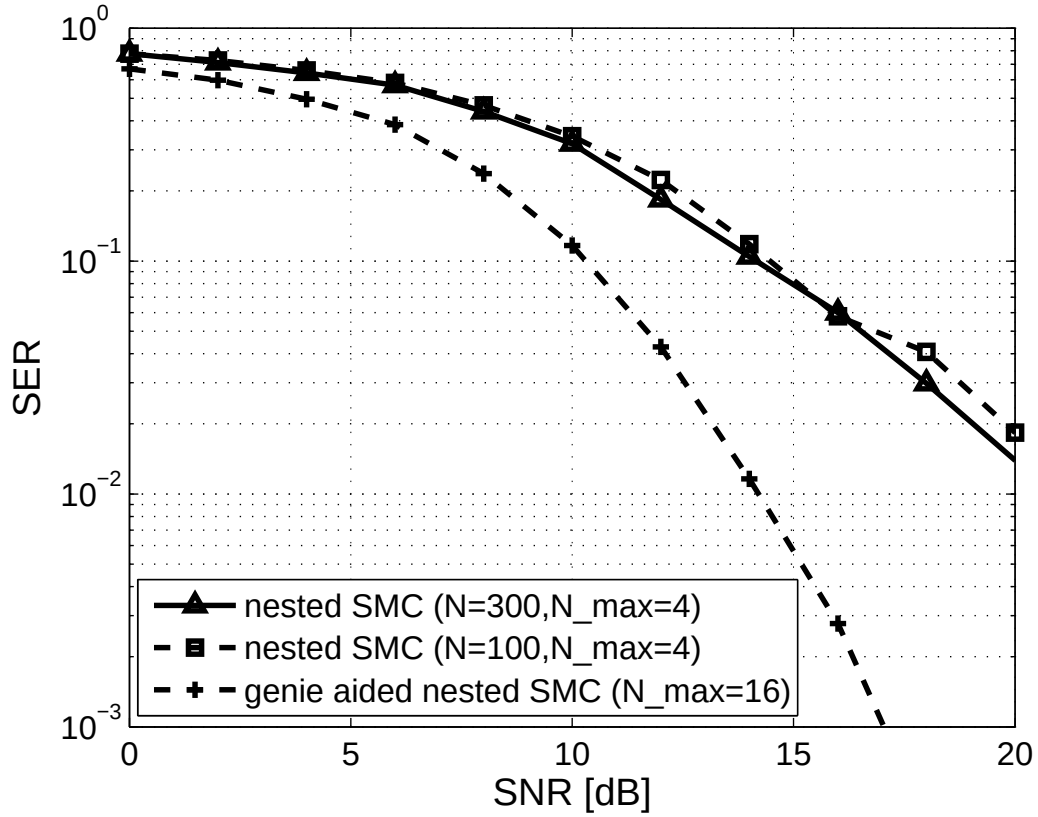


Figure 18: Performance of the blind SMC MIMO equalizer with 4×4 antennas and 16-QAM signaling

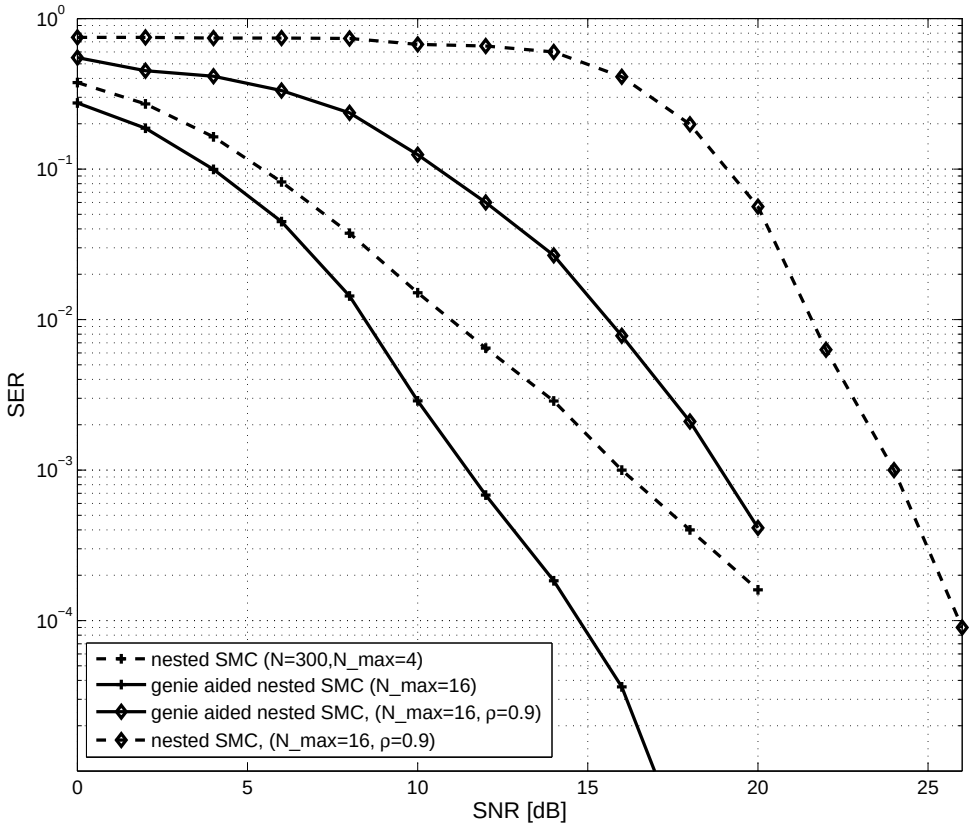


Figure 19: Performance of blind SMC MIMO equalizer for the spatially correlated case, $\rho = 0.9$, with 4×4 antennas and QPSK signaling

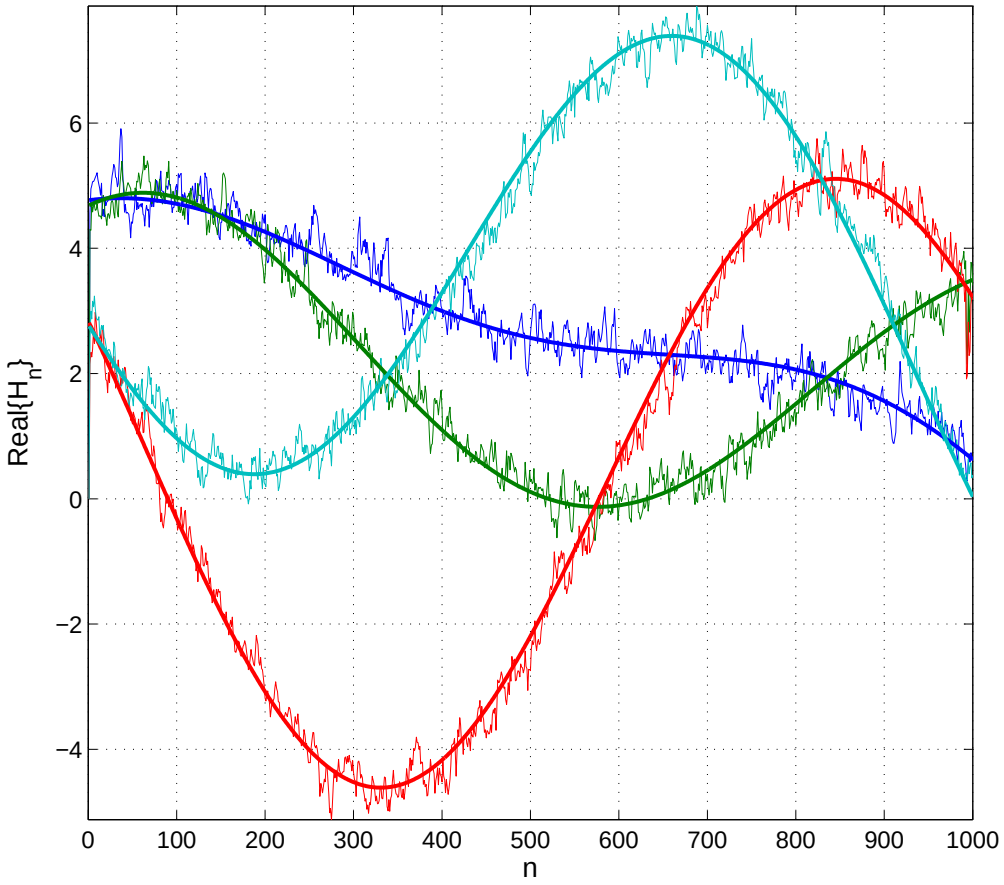


Figure 20: Channel estimates obtained by nested SMC with 4×4 antennas, QPSK signaling, 20dB SNR, and 300 particles. Solid thick lines are the true channel coefficients, and the thin jagged lines are the estimated values.

In Figure 20, the channel state tracking capability is displayed. For high SNRs, the channel coefficients are estimated well. For low SNRs (Figure 21), the thermal noise, symbol phase, symbol magnitude, and antenna ambiguities introduce larger errors. The particles begin tracking a rotated replica of the multidimensional channel state-space.

The problem is more complex in the 8×8 case. The symbol space has 4^8 and 16^8 elements, respectively, for the the QPSK and 16-QAM cases. We observe again that the proposed method significantly outperforms V-BLAST techniques, which are based on decision feedback (Figure 22). The gain in our receiver comes from the stored and traced multiple symbol trajectories.

As is evident from Figure 24, the larger we keep the maximum number of stored trajectories N_max, the better the results are. The computational cost grows linearly with the number of trajectories. Even in the case of genie aided receivers, where the true channel state information is obtained by our "genie", there is approximately $2dB$ of performance difference in the cases with $N\_max = 8$ and $N\_max = 16$. The performance degrades significantly when the receiver has to obtain the values of the 8×8 channel matrix and obtain symbol estimates. The SER performance is still better than DF-based V-BLAST algorithms.

In case of spatial correlation among the channels, the performance degrades significantly. In Figure 23, the case of spatial correlation among the antennas is illustrated. We observe similar effects as in the 4×4 case.

4.10 Conclusion

We constructed signal and communications channel models for MIMO systems. We then formulated a few approaches for obtaining symbol estimates. We defined the Bayesian problem, formulated the posterior distribution definitions, and obtained recursive formulations for the distributions and weight updates.

Given the problem, we outlined the standard SMC algorithm, which requires sampling from the entire symbol space. Based on the recursive formulation, we sample at each layer and retain symbol trajectories in order to avoid error propagation. With this approach the

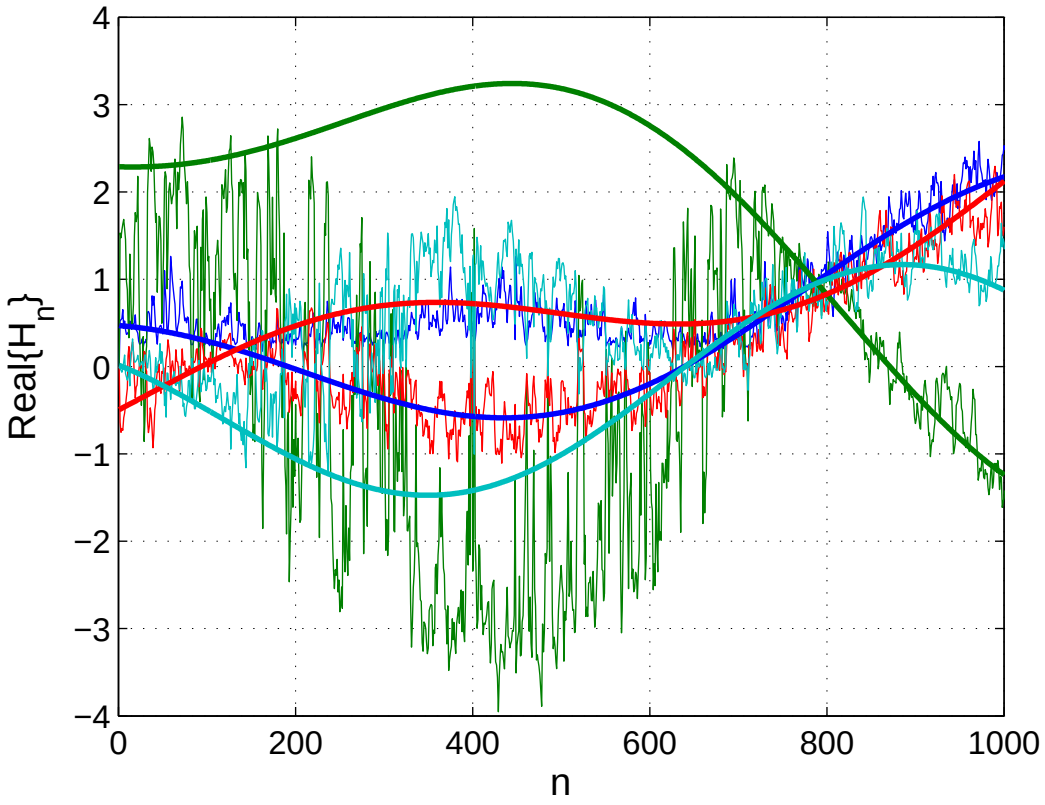


Figure 21: Channel estimates obtained by nested SMC with 4×4 antennas, QPSK signaling, 10dB SNR, and 300 particles. Solid thick lines are the true channel coefficients, and the thin jagged lines are the estimated values.

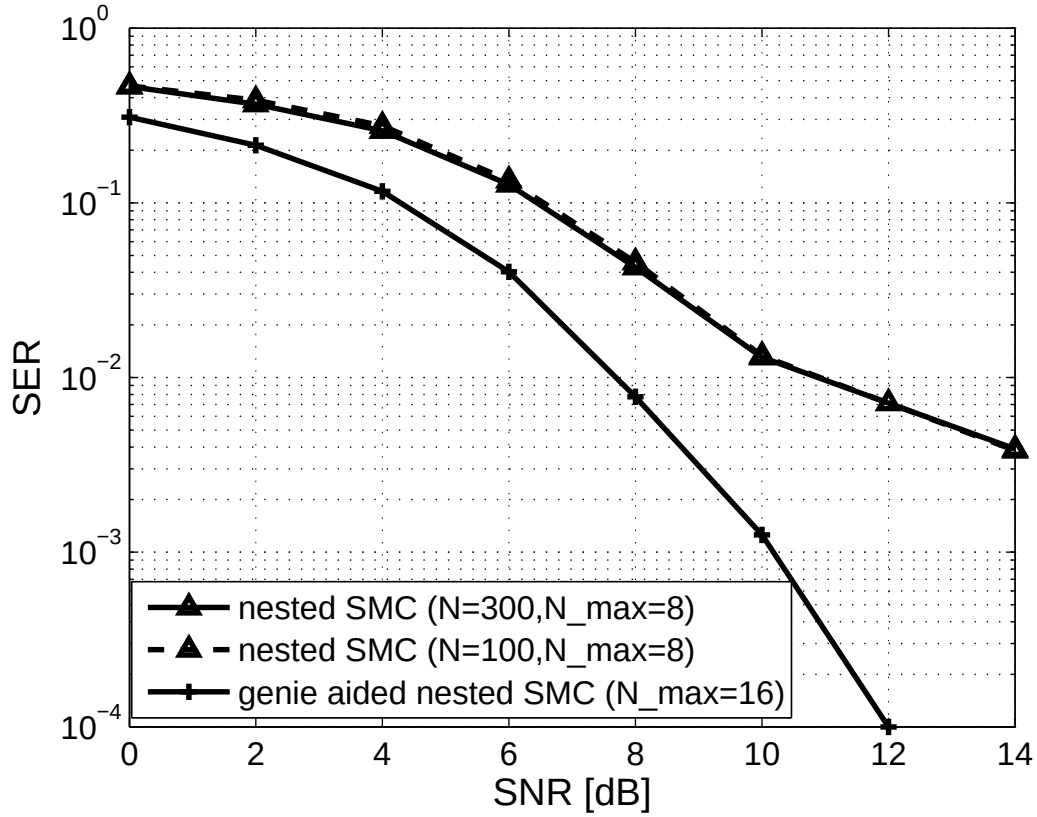


Figure 22: Performance of the blind SMC MIMO equalizer with 8×8 antennas and QPSK signaling

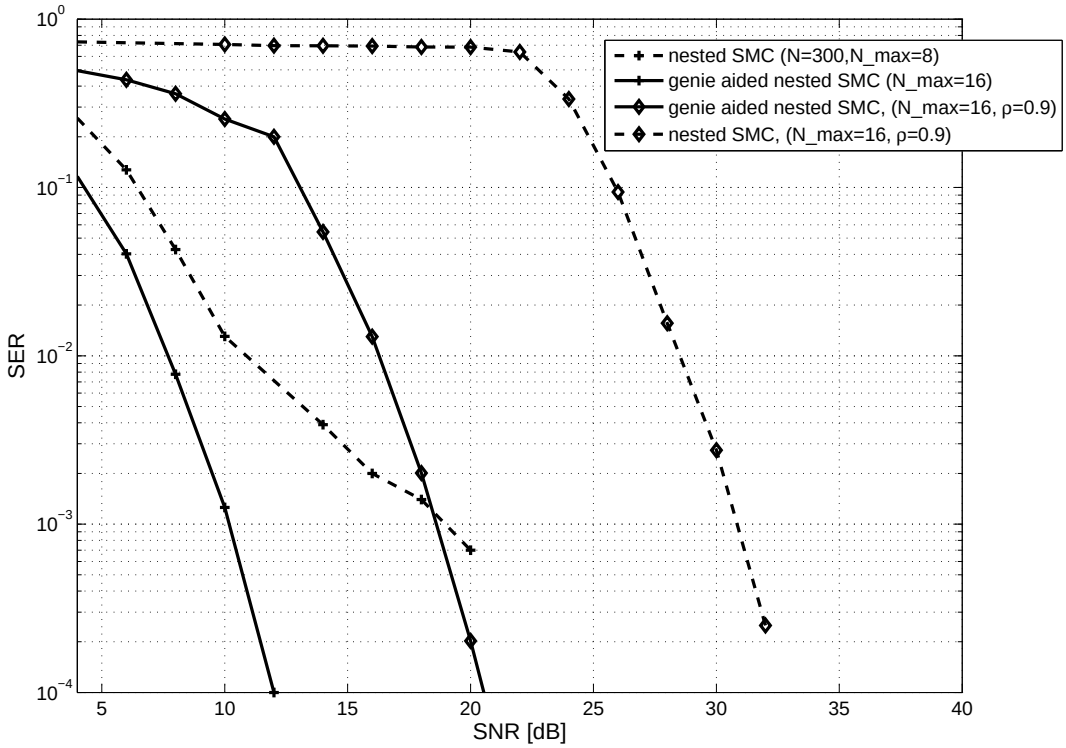


Figure 23: Performance of the blind SMC MIMO equalizer, for the spatially correlated case, $\rho = 0.9$, with 8×8 antennas and QPSK signaling

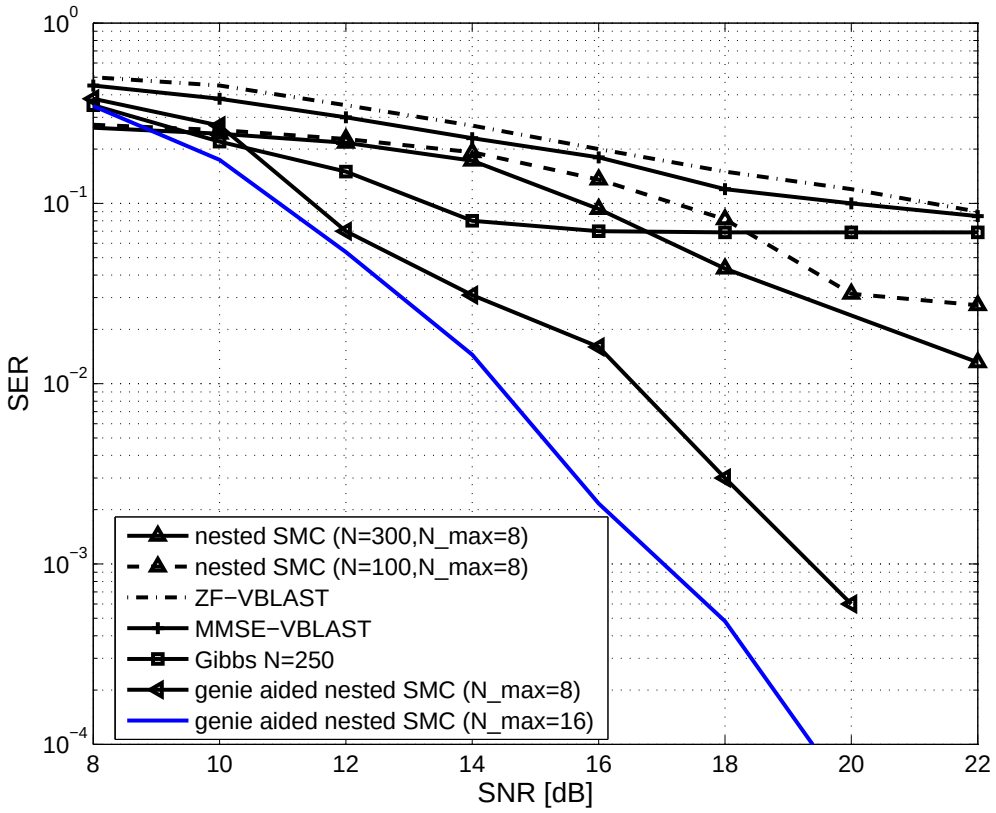


Figure 24: Performance of blind SMC MIMO equalizer with 8×8 antennas and 16-QAM signaling

complexity is reduced from being exponential in the number of transmit antennas, to only quadratic.

We tested the algorithm and compared it to the performance of other blind and non-blind algorithms. We found that the performance degradation because of layered sampling is minimal justifying our approach. The blind nested SMC algorithm outperforms standard DF based V-BLAST approaches with full CSI.

As the dimensionality of the problem decreases with both signalling constellation size and antenna numbers, the ambiguities also decrease. The performance of our blind algorithm is very close to that of the full complexity ML solution.

CHAPTER V

A BLIND RECEIVER BASED ON A SEQUENTIAL MONTE CARLO ALGORITHM FOR SPACE-TIME CODES AND MOCS

We propose a blind multiuser detector based on Monte Carlo Markov chain (MCMC) techniques. The detector exploits mutually orthogonal complementary sequences to distinguish between transmitting users and space-time codes to take advantage of the available spatial diversity. We propose a partitioning scheme for the symbol draws in the MCMC algorithm that reduces the complexity without any degradation in performance. We also propose a sequential Monte Carlo method to obtain the symbol estimates. The detector's performance is simulated in an iterative receiver that utilizes an outer coder. The simulations display some loss in coding gain because of the blind nature of the system; however, diversity gain is preserved.

5.1 Introduction

A sequential Monte Carlo (SMC) framework has been developed and a wide range of applications have been discussed in [50]. This Bayesian technique is primarily utilized in target tracking algorithms [12]; however, other applications also exist. Blind receivers are one application of this technique in the telecommunications field, and an adaptive scheme has been proposed in [17]. Here we extend this SMC technique for efficient blind detection of multiple users in multi-antenna scenarios.

Our approach utilizes mutually orthogonal (MO) sets rather than pseudo-noise (PN) sequences. MO sets introduce minimal bandwidth expansion since the chip-rate can be set to be the bit-rate. MO sets are also suitable for multi-user applications where uncoupled

parallel communication channels exist. A soft multi-user detector was introduced and utilized in an iterative receiver in [58], where the channel was assumed known at the receiver. In this work the channel is unknown at the receiver; however, a slow fading rate is assumed.

The contribution in this work is in the partitioning of the symbol space. Partitioning has been applied previously to visual and acoustic tracking problems with particle filters [51,56]. Here, though, we take advantage of the conditional independence of the received signal vectors. Instead of drawing symbols from the entire symbol space, we partition it, draw "sub-symbols" from the subspaces, and then merge the results. This technique reduces the complexity of the algorithm significantly without any performance loss or additional error propagation. We display the performance of the detector in an iterative receiver utilizing an outer channel code.

5.2 System Description

A detailed discussion on the construction and properties of MO complementary sets can be found in [81]. Use of these sets in multi-user communications has been discussed in [58]. Let us elaborate on the signaling at the transmitter in Fig. 25. The channel coded symbols c_k are differentially encoded into d_k and then convolved with the mutually orthogonal set A_{km} assigned to the kth user:

$$d_k[n] = d_k[n-1]c_k[n] \tag{133}$$

$$s_{km}[n] = d_k[n] * A_{km}[n], \tag{134}$$

where $(*)$ denotes convolution. We stack $s_{km}[n]$

$$\mathbf{s}_k[n] \triangleq [s_{k1}[n] \ \ldots \ s_{kM}[n]]^T. \tag{135}$$

We define the following quantities by stacking each user's successive samples of the coded data vectors. The goal here is to obtain a transmitter model that will account for all K users. Then, the symbols in the vector $\mathbf{s}_k$ are mapped to the space-time block code matrix

$\mathbf{S}_k[n]$. Here a code matrix designed by Tarokh is chosen [75]

$$\mathbf{S}_k = \begin{bmatrix} s_{k1} & s_{k2} & s_{k3} & s_{k4} \\ -s_{k2} & s_{k1} & -s_{k4} & s_{k3} \\ -s_{k3} & s_{k4} & s_{k1} & -s_{k2} \\ -s_{k4} & -s_{k3} & s_{k2} & s_{k1} \\ s_{k1}^* & s_{k2}^* & s_{k3}^* & s_{k4}^* \\ -s_{k2}^* & s_{k1}^* & -s_{k4}^* & s_{k3}^* \\ -s_{k3}^* & s_{k4}^* & s_{k1}^* & -s_{k2}^* \\ -s_{k4}^* & -s_{k3}^* & s_{k2}^* & s_{k1}^* \end{bmatrix}. \tag{136}$$

5.3 A Blind Multiuser Detector

To simplify notation we stack $\mathbf{S}_k$s, $\mathbf{h}_k$s, c_ks and d_ks as follows: $\mathbf{c}[n] = [c_1[n] \dots c_K[n]]^T$, $\mathbf{d}[n] = [d_1[n] \dots d_K[n]]^T$, $\mathbf{S}[n] = [\mathbf{S}_1[n] \dots \mathbf{S}_K[n]]$, $\mathbf{h}[n] = [\mathbf{h}_1[n]^T \dots \mathbf{h}_K[n]^T]^T$. Here, $\mathbf{h}_k[n]$ is a column vector containing the complex amplitudes of each channel. Subscripts for the remainder of the paper will indicate time epoch rather than user number or a sequence number within a set.

$$\mathbf{d}_{n_1:n_2} \triangleq [\mathbf{d}[n_1]^T, \mathbf{d}[n_1+1]^T, ..., \mathbf{d}[n_2]^T]^T, \quad n_2 > n_1 \tag{137}$$

and similarly $\mathbf{r}_n \triangleq \mathbf{r}[n]$.

The received signal vector can be described as:

$$\mathbf{r}_n = \mathbf{S}_n \mathbf{h}_n + \eta_n, \tag{138}$$

where η_n is receiver noise, which is additive white Gaussian noise with zero mean and σ^2 variance. In this paper we assume that the noise power is known and in practice it can be easily estimated.

Our goal is to calculate the a posteriori distribution $p(\mathbf{c}_n|\mathbf{r}_{0:n+L-1})$ of all the users' coded symbols $\mathbf{c}$ without prior knowledge of the channel state information $\mathbf{h}$. The architecture of the receiver is illustrated in Figure 26, which consists of serially cascaded MUD and channel decoders that pass extrinsic soft information to each other. Our contribution is in the design of the MC-MUD.

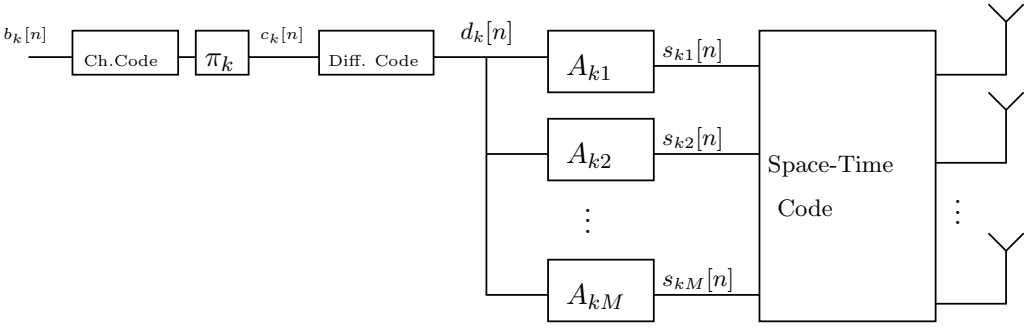


Figure 25: Transmitter Model

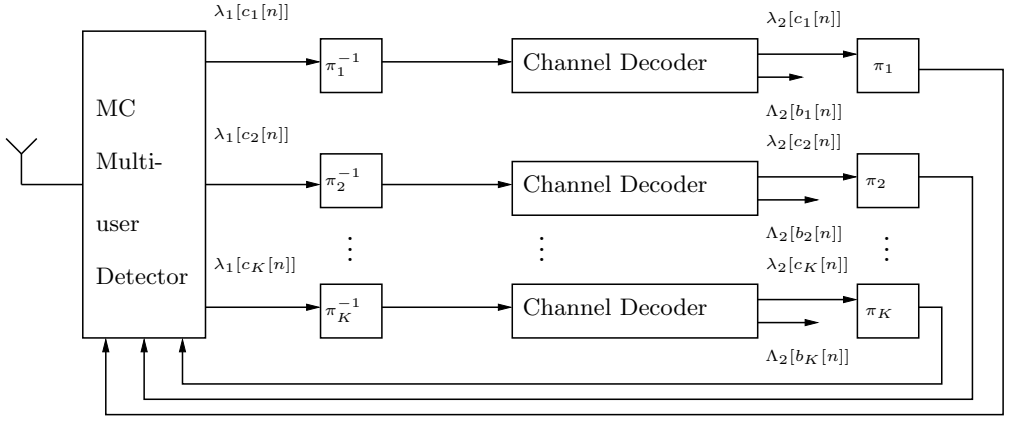


Figure 26: Iterative Receiver Model

5.4 The SMC Detector

The MOCS codes disperse the symbol $\mathbf{d}_n$ in the transmitted signal throughout L consecutive time epochs, where L is the length of the MO sequences. Namely, the sufficient statistics for $\mathbf{d}_n$ are the received signals $\mathbf{r}_{n:n+L-1}$. The joint posterior distribution of the symbol state-space and the channel state-space can be represented as the following factorization

$$p(\mathbf{d}_n, \mathbf{h}_n | \mathbf{r}_{n:n+L-1}) = p(\mathbf{d}_n | \mathbf{r}_{n:n+L-1}, \mathbf{h}_n) p(\mathbf{h}_n | \mathbf{r}_{n:n+L-1}). \tag{139}$$

We need to come up with a good representation of the above density. To simplify the equations, let us assume that we have a good estimate for the channel $\mathbf{h}_n$. Later we will include the effect of the uncertainty on $\mathbf{h}_n$.

5.4.1 Symbol detection

Let $\{\mathbf{d}_n^{(i)}, w_n^{(i)}\}_{i=1}^{N_s}$ denote a random measure that characterizes the a posteriori distribution $p(\mathbf{d}_n | \mathbf{d}_{0:n-1} \mathbf{r}_{n:n+L-1})$. Conditioning on $\mathbf{d}_{0:n-1}$ implies that those symbols have been detected previously. Let $\mathbf{d}_n^{(i)}, i = 1, \ldots, N_s$ be a sample draw. Our goal is to obtain samples of the transmitted symbols $\{\mathbf{d}_n^{(i)}, w_n^{(i)}\}$, properly weighted with respect to the distribution $p(\mathbf{d}_n | \mathbf{d}_{0:n-1}, \mathbf{r}_{n:n+L-1})$.

The weights are defined as:

$$w_n^{(i)} \propto \frac{p(\mathbf{d}_{0:n}^{(i)} | \mathbf{r}_{1:n+L-1})}{q(\mathbf{d}_{0:n}^{(i)} | \mathbf{r}_{1:n+L-1})}. \tag{140}$$

We choose a proposal density:

$$q(\mathbf{d}_n | \mathbf{d}_{0:n-1}, \mathbf{r}_{1:n+L-1}) \triangleq p(\mathbf{d}_n | \mathbf{d}_{0:n-1}, \mathbf{r}_{1:n+L-1}). \tag{141}$$

We choose to use the full posterior density as the proposal density, requiring heavy calculations in continuous state-space parameters. However, the symbols are from a discrete state-space and the posterior density is actually a probability mass function.

As a result of the proposal choice, the weight update simplifies significantly:

$$w_n^{(i)} \propto w_{n-1}^{(i)} p(\mathbf{r}_{n+L-1} | \mathbf{r}_{1:n+L-2}, \mathbf{d}_{0:n-1}^{(i)}). \tag{142}$$

The goal is to draw a sample from the posterior distribution. Alternatively, the draw can be done from its normalized likelihood probability mass function, requiring the evaluation of the likelihood at the entire symbol state-space. A MCMC method such as A-R, M-H, or the Gibbs sampler can be utilized to draw a sample from this distribution. The algorithm is outlined in Table table:MOCSblind:SMC pseudocode1. The process of likelihood evaluation and symbol draws is outlined in Table 17.

5.4.2 Channel Estimation and Rao-Blackwellization

In this section we elaborate on how the channel estimate is obtained.

The symbols $\mathbf{d}_{n-(L-1):n}$ contribute to the received signal $\mathbf{r}_n$. Therefore, the probability distribution of the received signal conditioned on the channel coefficients and the symbols $\mathbf{d}_{n-(L-1):n}$ is Gaussian:

$$p(\mathbf{r}_n|\mathbf{d}_{n-L+1:n}, \mathbf{h}_n) \sim \mathcal{N}(\mathbf{S}_n\mathbf{h}_n, \sigma^2\mathbf{I}) \tag{144}$$

We are interested in computing the multidimensional likelihood that appears in the factorization

$$\begin{aligned} p(\mathbf{d}_n|\mathbf{r}_{n:n+L-1}) \propto \int p(\mathbf{r}_{n:n+L-1}|\mathbf{d}_{n-(L-1):n+L-1}, \mathbf{h}_{n:n+L-1}) \\ p(\mathbf{h}_{n:n+L-1})p(\mathbf{d}_{n-(L-1):n+L-1})d\mathbf{h}_{n:n+L-1} \end{aligned} \tag{145}$$

The first distribution in the integrand is Gaussian. The second distribution describes the channel. The channel is a flat Rayleigh fading channel, and its distribution is that of a complex Gaussian random process. Hence, the above integral is also a complex Gaussian pdf

$$\begin{aligned} &\beta^{(i)}_{n:n+L-1}(\mathbf{r}_{n:n+L-1}|\mathbf{d}_{n-(L-1):n+L-1}, \mathbf{h}_{n:n+L-1}) = \\ &p(\mathbf{r}_{n:n+L-1}|\mathbf{d}_{n-(L-1):n+L-1}, \mathbf{h}_{n:n+L-1}) \sim \mathcal{N}(\mu^{(i)}_{n,j}, \mathbf{P}^{(i)}_{n,j}) \end{aligned} \tag{146}$$

$$\mu^{(i)}_{n,j} = \begin{bmatrix} S^{(i)}_{n,j} & 0 & \dots & 0 \\ 0 & S^{(i)}_{n+1,j} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & S^{(i)}_{n+L-1,j} \end{bmatrix} \begin{bmatrix} h^{(i)}_n \\ h^{(i)}_{n+1} \\ \vdots \\ h^{(i)}_{n+L-1} \end{bmatrix}. \tag{147}$$

Table 16: Blind MUD SMC pseudo-code

0. Initialization

1. for $i = 1, \ldots, N_s$ /*for each particle*/

   - Update channel coefficients using state-update equation
   - Calculate $\beta(\cdot)$ as described in Table 17
   - Draw $\tilde{\mathbf{d}}_n^{(i)} \sim \beta(\cdot)$ as described in Table 17
   - Update channel state based on outcome of the draw
   - Differentially decode

     $$\mathbf{c}_n^{(i)} = \tilde{\mathbf{d}}_n^{(i)} \odot \mathbf{c}_{n-1}^{(i)}, \tag{143}$$

     where $\odot$ denotes element-wise multiplication.

2. Update weight $w_n^{(i)} \propto w_{n-1}^{(i)} \beta_{n:n+L-1}^{(i)}(\cdot) p(\mathbf{c}_n^{(i)})$, where the last term is apriori probability, supplied by a previous iteration of the turbo receiver. Initially it is uniform.

3. Obtain symbol estimate

   $$\hat{\mathbf{c}}_n = \arg \max_{\mathbf{c}_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)} \delta(\mathbf{c}_n - \mathbf{c}_n^{(i)}),$$

4. Resample if needed

5. Increment time epoch and go to 1.

The elements in the main diagonal are space-time code matrices associated with the ith particle and jth symbol in the alphabet $\mathcal{A}^{KL}$. For shorter notation:

$$\mu_{n,j}^{(i)} = \mathbf{\Xi}_{n,j}^{(i)} \mathbf{h}_{n:n+L-1}^{(i)}. \tag{148}$$

Each particle $\mathbf{h}_{n:n+L-1}^{(i)}$ for the channel is also associated with a covariance matrix $\mathbf{\Sigma}_{n-1,j}^{(i)}$, which is defined recursively in equation (151). The covariance of the distribution in (146) is

$$\mathbf{P}_{n,j}^{(i)} = \sigma^2 \mathbf{I} + \mathbf{\Xi}_{n,j}^{(i)} \mathbf{\Sigma}_{n-1,j}^{(i)} {\mathbf{\Xi}_{n,j}^{(i)}}^H. \tag{149}$$

The channel estimate can be modeled as a Gaussian random process. Such random processes are fully characterized by their first and second moments. Dynamic updates of their means and covariances will increase the accuracy of the channel estimates. It is interesting to note that the updates turn out to be the Kalman filtering equations. With $\mu_{n,j}^{(i)}$ and $\mathbf{P}_{n,j}^{(i)}$ known, the channel updates are formulated easily.

$$\mathbf{h}_{n:n+L-1}^{(i)} = \mathbf{h}_{n-1:n+L-2}^{(i)} + \mathbf{K}_n^{(i)} (\mathbf{r}_{n:n+L-1} - \mu_{n,j}^{(i)}) \tag{150}$$

$$\mathbf{\Sigma}_{n,j}^{(i)} = (\mathbf{I} - \mathbf{K}_n^{(i)}) \mathbf{\Xi}_{n,j}^{(i)} \mathbf{\Sigma}_{n-1,j}^{(i)}, \tag{151}$$

where the Kalman gain is:

$$\mathbf{K}_n^{(i)} = \mathbf{\Sigma}_{n-1,j}^{(i)} {\mathbf{\Xi}_{n,j}^{(i)}}^H (\mathbf{P}_{n,j}^{(i)})^{-1}. \tag{152}$$

The algorithm is described in a more detailed fashion in Table 18.

5.5 Temporal Partition Particle Filter (TPPF)

The algorithm described in the previous section requires the computation of a weight update that involves length-$2ML$ vectors, increasing the complexity of the algorithm even further in equation (149). 2^{KL} matrices must be computed, each of which requires multiplication of $2ML \times LMK$ sized matrices. Moreover, in equation (152) the inverse of a $2ML \times 2ML$ sized matrix must be calculated. The primary contribution of this work is in the simplification of this process. We propose a partitioning scheme for the symbol draws that reduces the complexity of the algorithm and preserves the performance.

Our goal is to draw samples from the following distribution:

$$p(\mathbf{d}_n, \mathbf{h}_n|\mathbf{r}_{n:n+L-1}) \propto p(\mathbf{r}_{n:n+L-1}|\mathbf{d}_{n-(L-1):n+L-1}, \mathbf{h}_{n:n+L-1}) \\ p(\mathbf{h}_{n:n+L-1})p(\mathbf{d}_{n-(L-1):n+L-1}). \tag{159}$$

However, the received vectors are conditionally independent:

$$p(\mathbf{d}_n, \mathbf{h}_n|\mathbf{r}_{n:n+L-1}) \propto \prod_{l=0}^{L-1}[p(\mathbf{r}_{n+l}|\mathbf{d}_{n-(L-1)+l:n+l}, \mathbf{h}_{n+l}) \\ p(\mathbf{h}_{n+l})]p(\mathbf{d}_{n-(L-1):n+L-1}). \tag{160}$$

As a result of this partitioning, we get L distributions $\beta_{l,n,j}^{(i)}(\cdot), \quad l = 0, \ldots, L-1$, whose product is proportional to the incremental weight update.

$$\beta_{0,n,j}^{(i)}(\mathbf{r}_n|\mathbf{h}_n) = |\mathbf{P}_{n,j}^{(i)}|^{-1} \exp\Big\{-(\mathbf{r}_n - \mu_{n,j}^{(i)})^H \\ (\mathbf{P}_{n,j}^{(i)})^{-1}(\mathbf{r}_n - \mu_{n,j}^{(i)})\Big\} p(\mathbf{c}_n = \mathbf{d}_{n-1}^{(i)} \odot \mathbf{d}_n) \tag{161}$$

$$\beta_{1,n,j}^{(i)}(\mathbf{r}_{n+1}|\mathbf{h}_{n+1}, \mathbf{d}_{n+1}) = |\mathbf{P}_{n+1,j}^{(i)}|^{-1} \exp\Big\{-(\mathbf{r}_{n+1} - \mu_{n+1,j}^{(i)})^H \\ (\mathbf{P}_{n+1,j}^{(i)})^{-1}(\mathbf{r}_{n+1} - \mu_{n+1,j}^{(i)})\Big\} p(\mathbf{c}_n = \mathbf{d}_{n-1}^{(i)} \odot \mathbf{d}_n) \tag{162}$$

$$\beta_{L-1,n,j}^{(i)}(\mathbf{r}_{n+L-1}|\mathbf{h}_{n+L-1}, \mathbf{d}_{n+1:n+L-1}) = |\mathbf{P}_{n+L-1,j}^{(i)}|^{-1} \exp\Big\{-(\mathbf{r}_n - \mu_{n,j}^{(i)})^H \\ (\mathbf{P}_{n+L-1,j}^{(i)})^{-1}(\mathbf{r}_n - \mu_{n,j}^{(i)})\Big\} p(\mathbf{c}_n = \mathbf{d}_{n-1}^{(i)} \odot \mathbf{d}_n) \tag{163}$$

We may appear to be making the process more complex, but actually the computational complexity is relieved significantly because the matrix $\mathbf{P}_{n,j}^{(i)}$ has to be evaluated N_{symb}^L times less. The matrix itself is L times smaller in each dimension, hence computation of its inverse is also easier. N_{symb} above is the cardinality of the symbol constellation.

Let us summarize our TPPF blind detector:

0. Initialization

1. For each particle i

   - Compute

$$\mu_{n,j}^{(i)} = \tilde{\mathbf{S}}_{n,j}^{(i)} \mathbf{h}_n^{(i)} \tag{164}$$

$$\mathbf{P}_{n,j}^{(i)} = \sigma^2 \mathbf{I} + \tilde{\mathbf{S}}_{n,j}^{(i)} \mathbf{\Sigma}_{n-1,j}^{(i)} \tilde{\mathbf{S}}_{n,j}^{(i)H} \tag{165}$$

$$\beta_{0,n,j}^{(i)}, \beta_{1,n,j}^{(i)}, \ldots, \beta_{L-1,n,j}^{(i)} \tag{166}$$

for each possible $\tilde{\mathbf{S}}_{n:n+L-1} \in \mathcal{A}^{KL}$

- Draw a sample $\tilde{\mathbf{d}}_{n+L-1}^{(i)} \in \mathcal{A}^K$ from the distribution:

$$\begin{aligned} p(\mathbf{d}_{n+L-1}^{(i)} | \mathbf{d}_{0:n-1}, \mathbf{r}_{n+L-1}) \propto \\ \sum_{\mathbf{d}_{n:n+L-2}} p(\mathbf{r}_{n+L-1} | \mathbf{d}_{n:n+L-1}, \mathbf{h}_{n+L-1}) p(\mathbf{h}_{n+L-1}) p(\mathbf{d}_{n:n+L-1}) \propto \\ \sum_{\mathbf{d}_{n:n+L-2}} \beta_{L-1,n,j}^{(i)} \end{aligned} \tag{167}$$

- For $p = L-2, L-3, \ldots, 2$ draw a sample $\tilde{\mathbf{d}}_{n+p}^{(i)} \in \mathcal{A}^K$ from the distribution:

$$\begin{aligned} p(\mathbf{d}_{n+p}^{(i)} | \mathbf{d}_{0:n-1}, \mathbf{r}_{n+p:n+L-1}, \mathbf{sym}_{n+p:n+L-1}) \propto \\ \sum_{\mathbf{d}_{n:n+p-1}} \prod_{l=p}^{L-1} p(\mathbf{r}_{n+l} | \mathbf{d}_{n:n+p-1}, \mathbf{h}_{n+p}, \mathbf{sym}_{n+p:n+L-1}) p(\mathbf{h}_{n+p}) p(\mathbf{d}_{n:n+p-1}) \propto \\ \sum_{\mathbf{d}_{n:n+p-1}} \prod_{l=p}^{L-1} \beta_{l,n,j}^{(i)} \end{aligned} \tag{168}$$

- Draw a sample $\tilde{\mathbf{d}}_{n}^{(i)} \in \mathcal{A}^K$ from the distribution:

$$\begin{aligned} p(\mathbf{d}_{n}^{(i)} | \mathbf{d}_{0:n-1}, \mathbf{r}_{n:n+L-1}) \propto \\ \prod_{l=0}^{L-1} p(\mathbf{r}_{n+l} | \mathbf{d}_{n}, \mathbf{h}_{n+l}, \mathbf{sym}_{n+1:n+L-1}) p(\mathbf{h}_{n+l}) p(\mathbf{d}_{n}) \propto \\ \prod_{l=0}^{L-1} \beta_{l,n,j}^{(i)} \end{aligned} \tag{169}$$

- Differentially decode

$$\mathbf{c}_{n}^{(i)} = \tilde{\mathbf{d}}_{n}^{(i)} \odot \mathbf{c}_{n-1}^{(i)}, \tag{170}$$

where $\odot$ denotes element-wise multiplication.

- Update the a posteriori mean and covariance of the channel: $\mathbf{h}_n^{(i)}$ and $\mathbf{\Sigma}_{n,j}^{(i)}$ for the particular drawn symbol. Update the mean and covariance of the received signal vector, $\mu_{n:n+L-1,j}^{(i)}$ and $\mathbf{P}_{n:n+L-1,j}^{(i)}$, using equations (164-165)

$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(\mathbf{r}_n - \mu_{n,j}^{(i)}) \tag{171}$$

$$\mathbf{\Sigma}_{n,j}^{(i)} = (\mathbf{I} - \mathbf{K}_n^{(i)})\mathbf{S}_{n,j}^{(i)}\mathbf{\Sigma}_{n-1,j}^{(i)}, \tag{172}$$

where the Kalman gain is:

$$\mathbf{K}_n^{(i)} = \mathbf{\Sigma}_{n-1,j}^{(i)}{\mathbf{S}_{n,j}^{(i)}}^H(\mathbf{P}_{n,j}^{(i)})^{-1}. \tag{173}$$

- Update the weights:

$$w_n^{(i)} \propto w_{n-1}^{(i)} \prod_{l=0}^{L-1} \beta_{l,n,j}^{(i)} p(\mathbf{c}_n^{(i)}), \tag{174}$$

where the last term is the a priori probability, supplied by a previous iteration of the turbo receiver.

- Normalize the weights such that $\sum_{i=1}^{N_s} w_n^{(i)} = 1$.

2. Obtain symbol estimate $\hat{\mathbf{c}}_n = \arg \max_{\mathbf{c}_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)} \delta(\mathbf{c}_n - \mathbf{c}_n^{(i)})$

3. Resample if needed

4. Increment time epoch and go to 1.

Notice that the new algorithm uses sufficient statistics to estimate the current symbol. The space from which the symbols are drawn, however, has L times fewer dimensions. If the symbol constellation has cardinality N_{symb}, the complexity is reduced from $N_{symb}\exp(KL)$ to $N_{symb}L^2\exp(K)$ without any sacrifice in performance.

One may argue that any error in the "sub-symbol" draws may propagate and lead to incorrect symbol estimates and, consequently, an incorrect channel estimate. Such is not the case, because the received signal vectors are conditionally independent. Also note that, $p(\mathbf{r}_{n:n+L-1}|\mathbf{d}_{n+L-1}) = p(\mathbf{r}_{n+L-1}|\mathbf{d}_{n+L-1})$. Similar expressions could be written for the rest of the "sub-symbols." Therefore, the probability of incorrectly drawing a "sub-symbol" in the temporally partitioned case is the same as if we had considered the entire received signal

vector and the entire symbol space. Our temporal partitioning of the space does not affect the performance of the algorithm other than reducing the computation significantly.

5.6 A SMC approach to the symbol draw

We utilize the nested SMC approach to obtain a more reliable estimate for the transmitted symbols. The recursive structure of the probability densities is utilized to obtain a tree of symbol states similar to the one obtained in the previous chapter. The tree is traversed from the root toward the leaves and the less likely branches are discarded. The retained branches are expanded as we proceed to the next symbol. If the total number of the trajectories exceeds the chosen limit, a subset of them is chosen according to a stochastic criterion that was defined in the previous chapter.

This approach retains multiple trajectories corresponding to distinct samples from the symbol space. It is a more accurate approach because of decreases in the error propagation effect due to the decision feedback in the previous approach. Hence, the SMC approach results in fewer required channel particles to solve the problem.

Because of the similarity of the approach to the one in the previous chapter we only give basic pseudocode in Table 19 and Table 20.

5.7 Simulation Results

For our simulations, the channel encoder uses a rate 1/2 convolutional code with generators $23, 35$ in octal form. The channel was a flat, block static, slow ($f_m T = 1e-3$) Rayleigh fading channel. We utilized $M = 4$ transmit antennas per user. The mutually orthogonal sequences were of length $L = 4$. We had 4 sequences per mate set and $K = 2$ and $K = 4$ mate sets − one for each user. All interferers had equal transmit power. In our simulations we utilized a space-time block code for 4 Tx antennas [75].

The architecture of the transmitter causes a low transmission rate. A data symbol is transmitted in eight consecutive time slots. The transmitted power is normalized for comparison with other approaches. Obtaining reliable estimate for the channel state information

Table 17: Symbol draw pseudo-code

- for $l = 0, ..., L-1$
  - Compute
$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1}$$
  - For each possible $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \in \{\pm 1\}^L$ construct the space-time code matrix $\tilde{\mathbf{S}}_n$ using equations (134), (135) and the structure of a STBC.
$$\mu^{(i)}_{n+l} = {\mathbf{S}^{(i)}_{n+l}}^H \mathbf{h}^{(i)}_{n+l}$$
$$P^{(i)}_{n+l} = \sigma^2 \mathbf{I} + {\mathbf{S}^{(i)}_{n+l}}^H \mathbf{\Sigma}^{(i)}_{n+l} \mathbf{H}^{(i)}_{n+l}$$
  - For each possible $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \in \{\pm 1\}^L$ compute $\beta^{(i)}_{n+l}(\cdot) = p(r_{n+l}|\mathbf{h}^{(i)}_{n+l}, \tilde{\mathbf{S}}^{(i)}_{n:n+L-1}, s_{0:n-1})$
- Draw $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \sim \beta^{(i)}_{n:n+L-1}(\cdot)$

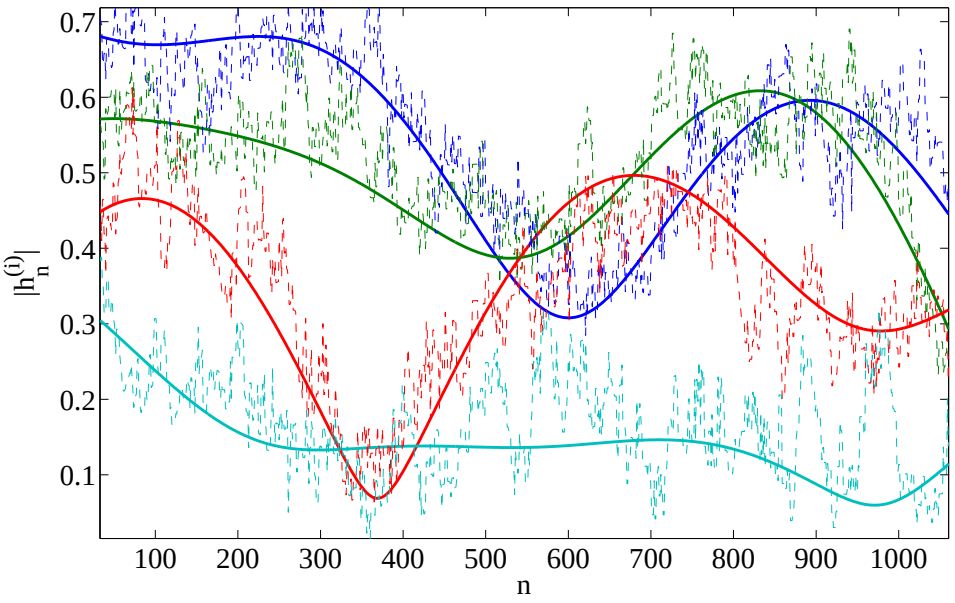


Figure 27: Channel as estimated by the temporally partitioned method with $K = 2$ users, 4 Tx antennas per user, and SNR=8dB. One user's channel shown.

Table 18: SMC algorithm for blind MOCS MUD

0. Initialization

1. for $i = 1, \ldots, N_s$ /*for each particle*/

   - Update channel coefficients using state-update equation, for $l = 0 : L-1$

$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1}$$

   - For each possible $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \in \{\pm 1\}^{KL}$ construct the space-time code matrix $\tilde{\mathbf{\Xi}}^{(i)}_{n,j}$ using equations (134), (135), the structure of a STBC, and equation (147).
   - Calculate

$$\mu^{(i)}_{n,j} = \mathbf{\Xi}^{(i)}_{n,j} \mathbf{h}^{(i)}_{n:n+L-1}. \tag{153}$$

$$\mathbf{P}^{(i)}_{n,j} = \sigma^2 \mathbf{I} + \mathbf{\Xi}^{(i)}_{n,j} \mathbf{\Sigma}^{(i)}_{n-1,j} \mathbf{\Xi}^{(i)\,H}_{n,j}. \tag{154}$$

   - For each possible $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \in \{\pm 1\}^{KL}$ compute $\beta^{(i)}_{n:n+L-1}(\mathbf{r}_{n:n+L-1} | \tilde{\mathbf{S}}^{(i)}_{n:n+L-1}, \mathbf{h}_{n:n+L-1})$.
   - Draw $\tilde{\mathbf{S}}^{(i)}_{n:n+L-1} \sim \beta^{(i)}_{n:n+L-1}(\cdot)$ using M-H sampler, or Gibbs sampler.
   - Update channel state based on outcome of the draw

$$\mathbf{h}^{(i)}_{n:n+L-1} = \mathbf{h}^{(i)}_{n-1:n+L-2} + \mathbf{K}^{(i)}_n (\mathbf{r}_{n:n+L-1} - \mu^{(i)}_{n,j}) \tag{155}$$

$$\mathbf{\Sigma}^{(i)}_{n,j} = (\mathbf{I} - \mathbf{K}^{(i)}_n) \mathbf{\Xi}^{(i)}_{n,j} \mathbf{\Sigma}^{(i)}_{n-1,j}, \tag{156}$$

     where the Kalman gain is:

$$\mathbf{K}^{(i)}_n = \mathbf{\Sigma}^{(i)}_{n-1,j} \mathbf{\Xi}^{(i)\,H}_{n,j} (\mathbf{P}^{(i)}_{n,j})^{-1} \tag{157}$$

   - Differentially decode

$$\mathbf{c}^{(i)}_n = \tilde{\mathbf{d}}^{(i)}_n \odot \mathbf{c}^{(i)}_{n-1}, \tag{158}$$

     where $\odot$ denotes element-wise multiplication.
   - Update weight $w^{(i)}_n \propto w^{(i)}_{n-1} \beta^{(i)}_{n:n+L-1}(\cdot) p(\mathbf{c}^{(i)}_n)$, where the last term is the a priori probability, supplied by a previous iteration of the turbo receiver. Initially it is uniform.

2. Obtain symbol estimate $\hat{\mathbf{c}}_n = \arg \max_{\mathbf{c}_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)} \delta(\mathbf{c}_n - \mathbf{c}^{(i)}_n)$

3. Resample if needed

4. Increment time epoch and go to 1.

Table 19: Nested SMC algorithm for blind MOCS MUD

0. Initialization

1. for $i = 1, \ldots, N_s$ /*for each particle*/

   - Update channel coefficients using state-update equation, for $l = 0 : L-1$

$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1}$$

   - Obtain a symbol sample according to Table 20.
   - Update channel state based on outcome of the draw

$$\mathbf{h}^{(i)}_{n:n+L-1} = \mathbf{h}^{(i)}_{n-1:n+L-2} + \mathbf{K}^{(i)}_n(\mathbf{r}_{n:n+L-1} - \mu^{(i)}_{n,j}) \tag{175}$$

$$\mathbf{\Sigma}^{(i)}_{n,j} = (\mathbf{I} - \mathbf{K}^{(i)}_n)\mathbf{\Xi}^{(i)}_{n,j}\mathbf{\Sigma}^{(i)}_{n-1,j}, \tag{176}$$

     where the Kalman gain is:

$$\mathbf{K}^{(i)}_n = \mathbf{\Sigma}^{(i)}_{n-1,j}{\mathbf{\Xi}^{(i)}_{n,j}}^H(\mathbf{P}^{(i)}_{n,j})^{-1}. \tag{177}$$

   - Differentially decode

$$\mathbf{c}^{(i)}_n = \tilde{\mathbf{d}}^{(i)}_n \odot \mathbf{c}^{(i)}_{n-1}, \tag{178}$$

     where $\odot$ denotes element-wise multiplication.
   - Update weight $w^{(i)}_n \propto w^{(i)}_{n-1}\beta^{(i)}_{n:n+L-1}(\cdot)p(\mathbf{c}^{(i)}_n)$, where the last term is apriori probability, supplied by a previous iteration of the turbo receiver. Initially it is uniform.

2. Obtain symbol estimate $\hat{\mathbf{c}}_n = \arg \max_{\mathbf{c}_n \in \{\pm 1\}} \sum_{i=1}^{N_s} w^{(i)}\delta(\mathbf{c}_n - \mathbf{c}^{(i)}_n)$

3. Resample if needed

4. Increment time epoch and go to 1.

Table 20: Symbol draw using nested SMC

- Set $N_{partilces} = 1$
- for k=0,...,L-1 /*start with first sub-symbol*/
  - Compute
  - Calculate

$$\mu_{n+k,j}^{(i)} = \mathbf{S}_{n+k,j}^{(i)} \mathbf{h}_{n+k}^{(i)}. \tag{179}$$

$$\mathbf{P}_{n+k,j}^{(i)} = \sigma^2 \mathbf{I} + \mathbf{S}_{n+k,j}^{(i)} \mathbf{\Sigma}_{n-1,j}^{(i)} {\mathbf{S}_{n+k,j}^{(i)}}^H. \tag{180}$$

    ,for each possible $\mathbf{S}_{n+k}^{(i)} \in \{\pm 1\}^K$. The matrices $\mathbf{S}_{n+k}$ are STBC matrices constructed fromthe elements of $\mathbf{s}_{n+k}$.
  - Compute $p(\mathbf{r}_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = -(\mathbf{r}_{n+k} - \mu_{n+k}^{(i)})^H {\mathbf{P}_{n+k}^{(i)}}^{-1} (\mathbf{r}_{n+k} - \mu_{n+k}^{(i)})$
  - For $j = 1, \ldots, N_{particles}$
    - Expand each trajectory to $|\{\pm 1\}^K| = 2^K$ (alphabet size of a multiuser symbol) distinct paths

$$\beta^{(2^K * j)}(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = \beta^{(j)}(r_{n+k-1}|\mathbf{h}_{n+k-1}^{(i)}, \mathbf{s}_{n+k-1})$$
$$p(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k-1}, s_{n+k} = \{-1, \ldots, -1\})$$

$$\beta^{(2^K * j+1)}(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k}) = \beta^{(j)}(r_{n+k-1}|\mathbf{h}_{n+k-1}^{(i)}, \mathbf{s}_{n+k-1})$$
$$p(r_{n+k}|\mathbf{h}_{n+k}^{(i)}, \mathbf{s}_{n+k-1}, s_{n+k} = \{-1, \ldots, -1, +1\})$$

  - Update $N_{particles} = N_{particles} * alphabet\_size$
  - if $N_{particles} > N_{partilces}\_max$ Truncate the trajectory list to $N_{partilces}\_max$ and set $N_{particles} = N_{partilces}\_max$. Use residual resampling when choosing the subset to be retained.
- Choose $\tilde{s}_n^{(i)}$ based on the trajectory with highest likelihood.

becomes problematic. At low SNRs, the algorithm performs poorly, mainly because no reliable channel estimate can be obtained. As the SNR is increased, at a certain threshold, the performance improves rapidly (Figure 28). The channel estimates become significantly more reliable, and the turbo decoder helps to drastically improve the bit estimates. Consequently, the BER plot exhibits a steep slope.

This phenomenon is better observed in the 4 interferers scenario (Figure 29). The gap between the blind and the receivers with full CSI has increased because the uncertainty in the channel estimates for all 4 transmitters, each of which has 4 antennas, increases. We have also found that increasing the particle number for the channel estimates beyond 300 does not improve the performance significantly and does not justify the additional computational complexity (Figure 30).

5.8 Conclusion

In this chapter, we have utilized the properties of MOCS codes to construct a blind multiuser detector. The full complexity SMC detector has been illustrated in detail to point out the modifications made to the algorithm. This detector's implementation is nearly impossible because its complexity grows exponentially with the number of users and the length of the MOCSs. The complexity is significantly reduced by implementing a smart partitioning of the symbol space without any reduction in performance. Through simulations, the performance of these algorithms has been found to be acceptable and the solution of the problem tractable.

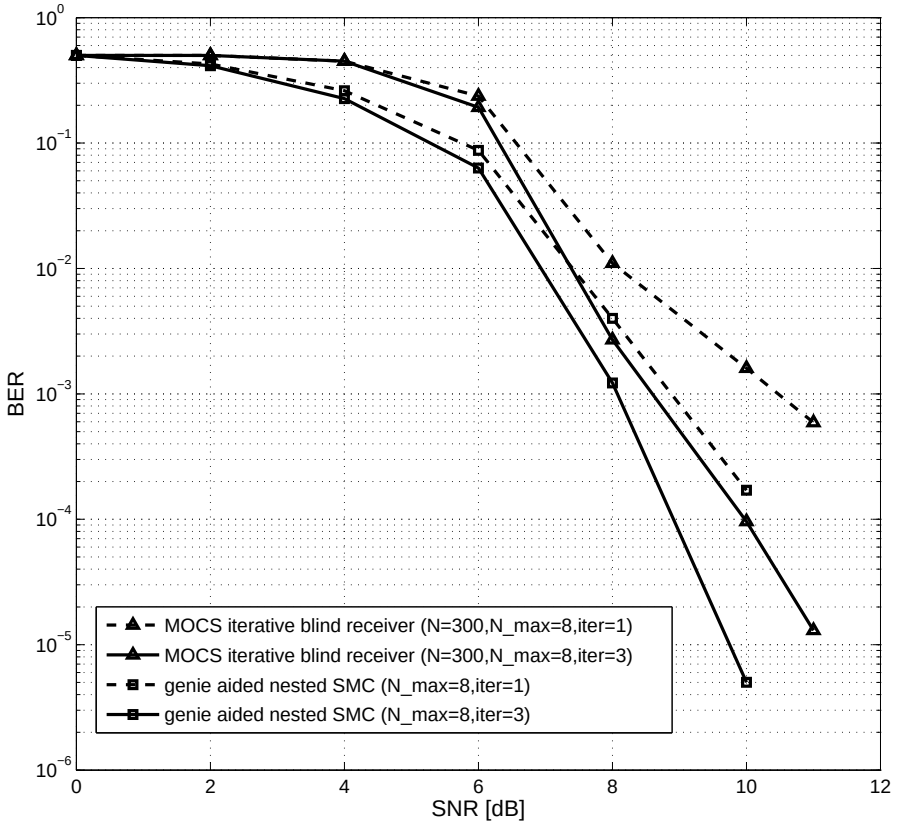


Figure 28: Performance of the nested SMC approach compared to the case of full channel state information (CSI) at receiver with $K = 2$ equal power users and 4 Tx antennas per user.

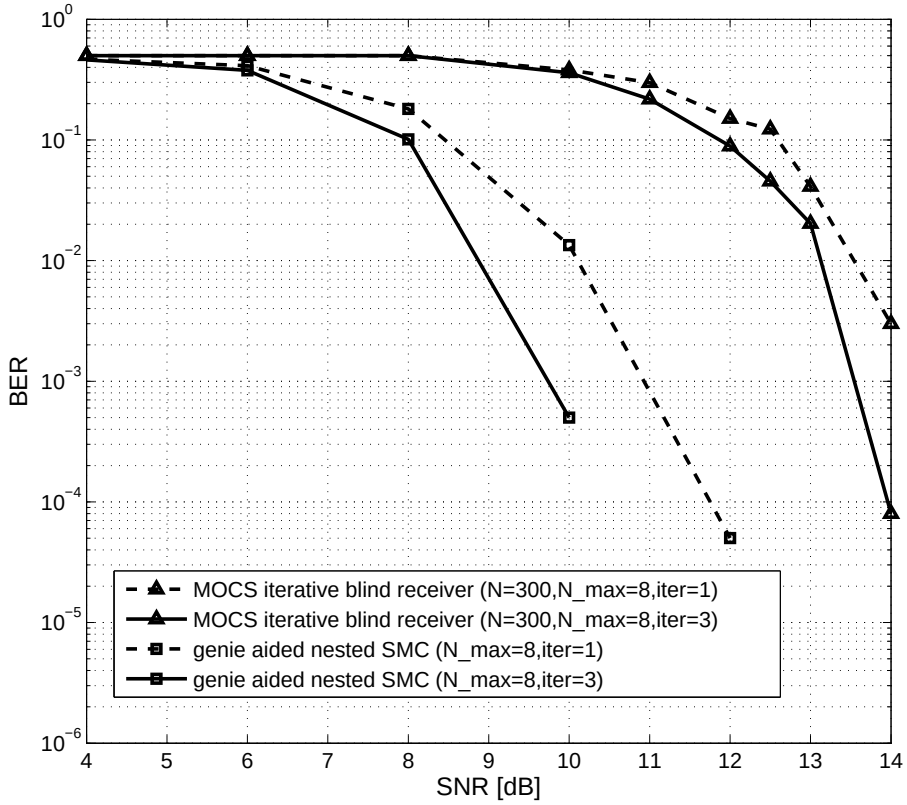


Figure 29: Performance of the nested SMC approach compared to the case of full channel state information (CSI) at receiver with $K = 4$ equal power users and 4 Tx antennas per user.

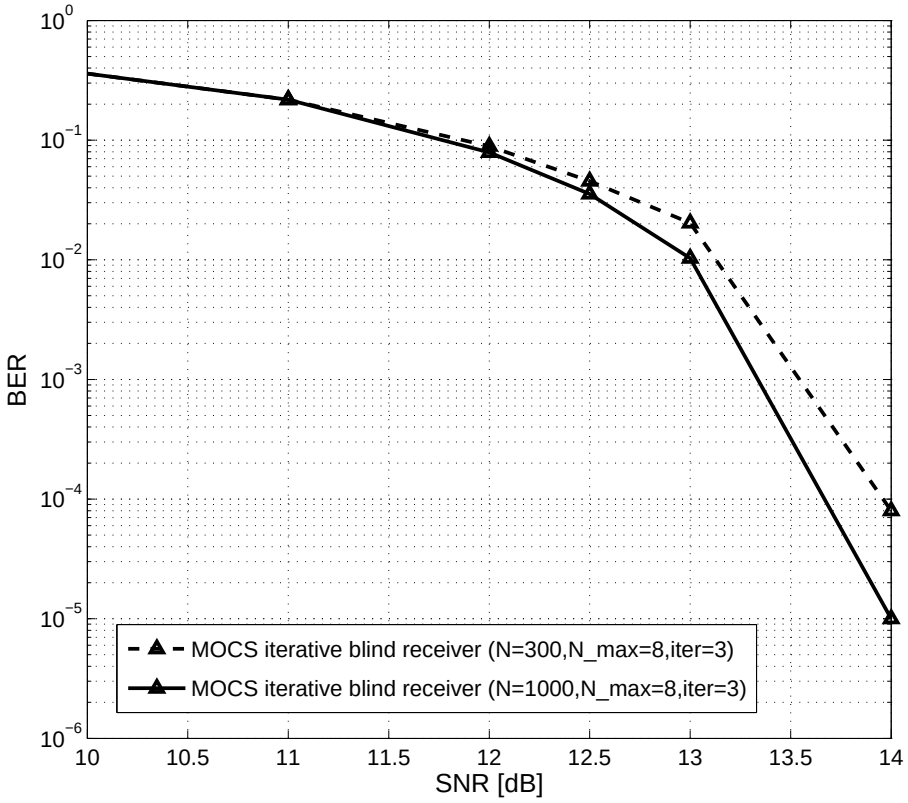


Figure 30: Performance of the nested SMC approach with $N\_particles = 300$ and $N\_particles = 1000$ with $K = 4$ equal power users and 4 Tx antennas per user.

CHAPTER VI

CONVERGENCE ANALYSIS AND COMPARISONS OF SEQUENTIAL MONTE CARLO ALGORITHMS

Recently, sequential Monte Carlo methods have been applied in the telecommunications field and found application in receiver design. The properties of these receivers make these design approaches very attractive. These receivers do not require channel state information or training. Therefore, they are bandwidth efficient and no communication bandwidth needs to be wasted on training. The receivers are optimal in the sense that they achieve minimum symbol error rate regardless of the noise distribution, nonlinearities in the system, and distribution of the transmitted symbol. Moreover, these receivers are capable of producing soft-information outputs, which enables the designer to utilize iterative receiver architectures for near-optimal performance. In this work we investigate the convergence properties of these algorithms when utilized in various types of receivers. We quantify the convergence rate. We describe how various parameters (e.g. noise power, channel fading rate, etc) and factors (e.g. state-space model mismatch) affect the convergence rate and point out the factors that should be improved first to gain speed and accuracy in the convergence.

6.1 Introduction

Monte Carlo methods allow sampling from complex, intractable distributions by simulating a Markov chain, whose invariant distribution approaches the desired *stationary* distribution almost surely [50]. In Monte Carlo terms, distributions are represented by discrete state realizations, called particles, that are distributed according to the underlying distribution after convergence. On the other hand, sequential Monte Carlo methods, also known as particle filters, employ a different approach to recursively sample from dynamically varying distributions: they reuse the current particle support to help reconstruct the new particles needed to represent the evolving system. All these Monte Carlo methods rely on the proposal

function, whose choice depends on the practitioner [73].

SMC methods use a weighting mechanism to account for the discrepancies caused by mismatch between the true distribution and the discrete support generated by the proposal distribution. Hence, the performance of the SMC methods is mainly determined by the proposal function given a constant number of particles. In theory, directly sampling the posterior would be optimal but is generally impractical. Therefore, algorithm designers resort to distributions that are very easy to sample from and in some manner resemble the posterior in terms of shape, support coverage, or some other metric [11]. One possible metric is the Kullback-Leibler divergence. However, it does not satisfy the triangle inequality and, therefore, cannot be used to compare the quality of the proposal functions [11].

For an SMC algorithm to work effectively, the proposal distribution must move the particles to their new support, mimicking the true posterior. Specifically, the particles must appear to represent the posterior distribution as closely as possible. Proposals functions that disperse the particles to the tail regions of the posterior instead of concentrating them in the mode regions are not efficient. In these cases, we spent computational power on the tail particles that do not affect the estimation because they have very low weights. The converse scenario is also bad. If the proposal distribution concentrates and restricts the particles to a small range within the mode of the posterior, the posterior is under represented and the estimates for higher moments for this distribution become inaccurate. Hence, a good proposal function should correctly cover the support of the posterior, while preserving computational power [22].

We propose a method to evaluate the quality of the proposal functions in terms of the particle filter convergence rate. The method is similar to the convergence analysis methods applied to the discrete Markov chains [16]. We relate the convergence speed of the chain to the magnitude of the second largest eigenvalue of the Markov chain transition probability matrix. We also compare the effectiveness of different proposal functions by monitoring temporal evolution of this eigenvalue.

The organization of this section is as follows. We introduce the convergence conditions

for the SMC proposal functions in section. Next, we describe a way to evaluate the convergence rate of such chains and SMC algorithms. We propose a method to measure the convergence of continuous Markov chains. In the last section we provide discrete state-space SMC examples from the telecommunications field.

6.2 Convergence of Sequential Monte Carlo Algorithm

The trajectory of each particle in SMC algorithms can be viewed as a realization of a Markov chain. At every iteration of the SMC algorithm, the particles are moved to a new position by the proposal distribution. The proposal distribution constitutes a Markov transition kernel. As the algorithm iterates, the trajectory of each particle reflects the state changes of the system. However, observation noise, model mismatches, and other factors cause deviations from the ideal trajectory [50]. For the algorithm to continue tracking a dynamic state space, the Markov chain should quickly converge around the mode of the posterior. When this convergence rate is slow, the resulting estimates are inaccurate.

6.2.1 Basics of Markov Chains

Let us denote a trajectory with $\{\mathbf{x}_0^{(i)}, \mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_N^{(i)}\}$, which is also a realization of a Markov chain defined on a state space E [16, 48]. Let $\mathcal{E}$ be a countably generated σ-algebra on E. Let $P(A|\mathbf{x}) = P(\mathbf{x}_{n+1}^{(i)} \in A|\mathbf{x}_n^{(i)} = \mathbf{x})$ for all $A \in \mathcal{E}$ and $\mathbf{x} \in E$, which is the probability of transition from state $\mathbf{x}$ to state A ($\mathbf{x}$ is in the state space and A is in the σ-algebra). Let also $P^n(A|\mathbf{x})$ denote a n-step transition function $P^n(A|\mathbf{x}) = P(\mathbf{x}_n^{(i)} \in A|\mathbf{x}_0^{(i)} = \mathbf{x})$.

The property of irreducibility is a first measure of the sensitivity of the Markov chain to the initial conditions. Irreducibility is crucial because it leads to a guarantee of convergence, without detailed study of the transition function.

Basically, a chain is irreducible if all of its states communicate [11]. The formal definition is a little more involved. Given a measure ϕ, the Markov chain $\mathbf{x}_n$ with transition kernel $P(A|\mathbf{x})$ is *ϕ-irreducible* if, for every $A \in \mathcal{E}$ with $\phi(A) > 0$ there exists n such that $P^n(A|\mathbf{x}) > 0$ for all $\mathbf{x} \in E$.

Another property of the Markov chain that must be specified is its aperiodicity. A

period of a state $\mathbf{x} \in E$ is defined as

$$d(\mathbf{x}) = \text{g.c.d}\{n \geq 1; P^n(\mathbf{x}, \mathbf{x}) > 0\} \tag{181}$$

The value of the period for all states that communicate with $\mathbf{x}$ is the same. If the chain is irreducible, the period is the same for all states. If the period is one for the state $\mathbf{x}$ and the chain is irreducible, then the chain is called aperiodic.

Here, we introduce the L^1-distance between two measures P and Q, defined as

$$\begin{aligned} \|P - Q\|_{\text{var}} &= \sup_{A \in \mathcal{E}} |P(A) - Q(A)| \\ &= \frac{1}{2} \sum_{\mathbf{x} \in E} |P(\mathbf{x}) - Q(\mathbf{x})| \equiv \frac{1}{2} \|P - Q\|_{L^1} \end{aligned} \tag{182}$$

When the two measures have density functions $p(\mathbf{x})$ and $q(\mathbf{x})$, the distance can be written as

$$\|P - Q\|_{\text{var}} = \frac{1}{2} \int |P(\mathbf{x}) - Q(\mathbf{x})| \text{d}\mathbf{x} \tag{183}$$

Theorem 6.2.1. *[77]: Suppose the state space* E *of a Markov chain is finite. The transition function of this chain is irreducible and aperiodic; then,* $P^n(A|\mathbf{x}) = P^n(\mathbf{x}_n^{(i)} \in A | \mathbf{x}_0^{(i)} = \mathbf{x})$ *as a probability measure on* A *converges to its invariant distribution* $\pi(\cdot)$ *geometrically in variation distance; namely, there exists a* $0 < r < 1$ *and* $c > 0$ *such that*

$$\|P^n(A|\mathbf{x}) - \pi(A)\|_{\text{var}} \leq cr^n \tag{184}$$

Theorem 6.2.2. *(Tierney [77]) : Suppose* A *is* π*-irreducible and* $\pi A = \pi$*. Then* A *is positive recurrent and* π *is the unique invariant distribution of* A*. If* A *is also aperiodic, then, for almost all* $\mathbf{x}$ *(for all* $\mathbf{x}$ *except a subset whose measure under* π *is zero),*

$$\|P^n(A|\cdot) - \pi(\cdot)\|_{\text{var}} \to 0, \tag{185}$$

where $\|\cdot\|_{\text{var}}$ *denote the total variation distance.*

The above two theorems prove that all Markov chains that are irreducible and aperiodic converge geometrically to their invariant distribution. This convergence proof is achieved without the explicit analysis of the transition kernel.

6.2.2 Convergence Rate

Consider a Markov chain on a finite discrete state space with κ states. We define a state transition probability matrix $\mathbf{P}$ whose entry p_{ij} indicates the transition probability from state i to state j. By definition, $\sum_j p_{ij} = 1$ for all i. Hence, $\mathbf{P}$ has an eigenvalue equal to 1.

Consider an arbitrary function g on the state-space E. Because the state-space is discrete, the function can be represented by a vector $\mathbf{g}_0$. The vector after the state transition is $\mathbf{g}_1 = \mathbf{P}\mathbf{g}_0$, which is simply the conditional expectation function:

$$g_1(\mathbf{x}) = E\left\{g(\mathbf{x}_1)|\mathbf{x}_0 = \mathbf{x}\right\} \tag{186}$$

We also note that

$$\text{var}\left\{E\{g_1(\mathbf{x})\}\right\} \leq \text{var}\left\{g(\mathbf{x}_1)\right\}. \tag{187}$$

Hence, the eigenvalues of the matrix $\mathbf{P}$ must be less than or equal to 1 in absolute value. Moreover, Berrman and Plemmons show that all of $\mathbf{P}$'s eigenvalues are real and can be diagonalized [6]

$$\mathbf{P} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^{-1}, \tag{188}$$

where $\mathbf{\Lambda} = diag([1, \lambda_2, \ldots, \lambda_\kappa])$, with the eigenvalues sorted in descending order of their absolute value, $1 \geq |\lambda_2| \geq \ldots |\lambda_\kappa|$. Therefore, as $n \to \infty$,

$$\mathbf{P}^n = \mathbf{B}\begin{bmatrix} 1 & 0 & & 0 \\ 0 & \lambda_2^n & & 0 \\ & & \ddots & \\ 0 & & & \lambda_\kappa^n \end{bmatrix}\mathbf{B}^{-1} \overset{n\to\infty}{\longrightarrow} \mathbf{B}\begin{bmatrix} 1 & 0 & & 0 \\ 0 & 0 & & 0 \\ & & \ddots & \\ 0 & & & 0 \end{bmatrix}\mathbf{B}^{-1} \tag{189}$$

The above is true if and only if $|\lambda_2| < 1$. Because $\pi\mathbf{P}^n = \pi$ and the limit of $\mathbf{P}^n$ is of rank 1, every row of $\mathbf{P}^\infty$ must be the same as π.

6.3 Convergence Analysis of an SMC Blind Equalizer for Frequency Selective Fading Channels

Details of receiver design and computational complexity improvements are thoroughly discussed in Chapter 3 and in [59]. Here we briefly outline the algorithm.

6.3.1 System Model

Consider a communications system that transmits a sequence of symbols $\mathbf{s}_{0:n} = \{s_0, s_1, \ldots, s_n\}$ randomly chosen from some constellation $\mathcal{A}$. At the receiver end, the receiver observes the signal $\mathbf{r}_{0:n}$ which is a random process that depends on the symbols $\mathbf{s}_{0:n}$ and channel parameters $\mathbf{h}_{0:n}$. Typically the channel state information $\mathbf{h}_{0:n}$ is a multidimensional continuous-valued quantity. The system can be described with the following state-space representation:

$$\begin{aligned} \mathbf{h}_n &\sim q_1(\mathbf{h}_n|\mathbf{h}_{0:n-1}) \\ s_n &\sim q_2(s_n|s_{n-1}) \\ \mathbf{r}_n &\sim f(\mathbf{r}_n|\mathbf{s}_{0:n}, \mathbf{h}_{0:n}). \end{aligned} \tag{190}$$

Based on the state-space representation, we can formulate two inference problems:

- Symbol Sequence Detection: Assuming the channel $\mathbf{h}_{0:n}$ is known, find

$$\hat{\mathbf{s}}_{0:n} = \arg \max_{\mathbf{s}_{0:n} \in \mathcal{A}^{n+1}} p(\mathbf{s}_{0:n}|\mathbf{r}_{0:n}, \mathbf{h}_{0:n}) \tag{191}$$

- Channel Estimation: Assuming the transmitted symbols $\mathbf{s}_{0:n}$ are known, find

$$\hat{\mathbf{h}}_{0:n} = \arg \max_{\mathbf{h}_{0:n}} p(\mathbf{h}_{0:n}|\mathbf{r}_{0:n}, \mathbf{s}_{0:n}) \tag{192}$$

The channel is usually not known and needs to be estimated. For this purpose either a training sequence of symbols is required or a blind approach must be utilized. Using Monte Carlo processing, the channel parameters can be marginalized out

$$\hat{\mathbf{s}}_{0:n} = \arg \max_{\mathbf{s}_{0:n} \in \mathcal{A}^{n+1}} E\{p(\mathbf{s}_{0:n}|\mathbf{r}_{0:n}, \mathbf{h}_{0:n})|\mathbf{h}_{0:n}\}, \tag{193}$$

where the expectation is taken with respect to the distribution of $\mathbf{h}_{0:n}$.

We represent the posterior with a discretized approximation. We obtain a set of samples and assign weights to it. The notation $\{\mathbf{s}_n^{(i)}, w^{(i)}\}_{i=0}^{N_s}$ is called a random measure that characterizes the posterior distribution $p(\mathbf{s}_{0:n}|\mathbf{r}_{0:n})$. A maximum a posteriori estimate (MAP) is calculated as follows:

$$\hat{s}_n = \arg \max_{s_n \in \{\mathcal{A}\}} \sum_{i=1}^{N_s} w^{(i)} \delta(s_n - s_n^{(i)}), \tag{194}$$

A detailed description of the equalizer algorithm is given in Table 21.

Table 21: Temporally Partitioned SMC for Frequency Selective Fading Channels

---

- Initialization
- for $i = 1, \ldots, N_s$ /*for each particle*/
  - for k=m-1,...,0 /*start with last sub-symbol*/
    - for l=k,...,m-1 /*iterate*/
      - Compute

$$\mathbf{h}_{n+l} = \zeta \mathbf{h}_{n-1+l} + (1-\zeta)\eta_{n-1} \tag{195}$$

$$\mu_{n+l}^{(i)} = {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{h}_{n+l}^{(i)}$$

$$P_{n+l}^{(i)} = \sigma^2 + {\mathbf{s}_{n+l}^{(i)}}^T \mathbf{\Sigma}_{n+l}^{(i)} \mathbf{s}_{n+l}^{(i)}$$

        ,for each possible $s_{n+k}^{(i)} \in \{\pm 1\}$
      - Compute $\beta_{l_n}^{(i)}(\cdot) = p(r_{n+l}|\mathbf{h}_{n+l}^{(i)}, s_{n+k}^{(i)}, \tilde{s}_{n+k+1:n+L-1}^{(i)}, s_{0:n-1})$
    - Draw $\tilde{s}_{n+k}^{(i)} \sim \prod_{l=k}^{m-1} \beta_{l_n}^{(i)}$
  - Update

$$\mathbf{h}_n^{(i)} = \mathbf{h}_{n-1}^{(i)} + \mathbf{K}_n^{(i)}(\mathbf{r}_n - \mu_n^{(i)})$$

$$\mathbf{\Sigma}_n^{(i)} = (1 - K_n^{(i)}){\mathbf{s}_n^{(i)}}^T \mathbf{\Sigma}_{n-1}^{(i)},$$

    where the Kalman gain is:

$$K_n^{(i)} = \mathbf{\Sigma}_{n-1}^{(i)} \mathbf{s}_n^{(i)} (P_n^{(i)})^{-1}$$

  - Update weight $w_n^{(i)} \propto w_{n-1}^{(i)} \beta_{n:n+L-1}^{(i)}(\cdot)$
- Resample if needed
- Perturb channel particles and update based on likelihoods.

6.3.2 Convergence

We utilize Rao-Blackwellization in the algorithm, which means that we proceed with analytical solutions rather than resorting to Monte Carlo whenever possible and feasible. This results in improved efficiency and convergence. In this case we Rao-Blackwellize the channel coefficient part of the problem. As it is easily seen the results resemble Kalman filtering equations. The symbol estimation part, however, is a non-Gaussian problem. The symbol distribution is non-Gaussian. Therefore, we retain Monte Carlo methodology. The symbols are drawn from the posterior distribution that can be analytically computed, which also results in increased convergence and efficiency.

The properties of the transition distribution $p(\mathbf{s}_{0:n}|\mathbf{s}_{0:n-1}, \mathbf{r}_{0:n})$ determine the convergence rate of the algorithm. Let us factorize the posterior distribution:

$$p(\mathbf{s}_{0:n}|\mathbf{r}_{0:n}) = \frac{p(\mathbf{r}_n|\mathbf{s}_{0:n}, \mathbf{r}_{0:n-1})p(\mathbf{s}_n|\mathbf{s}_{0:n-1}, \mathbf{r}_{0:n-1})}{p(\mathbf{r}_n)} p(\mathbf{s}_{0:n-1}|\mathbf{r}_{0:n-1}) \tag{196}$$

The transition kernel turns out to be

$$\begin{aligned} p(\mathbf{s}_{0:n}|\mathbf{s}_{0:n-1}, \mathbf{r}_{0:n}) &= \frac{p(\mathbf{s}_{0:n}|\mathbf{r}_{0:n})}{p(\mathbf{s}_{0:n-1}|\mathbf{r}_{0:n-1})} \\ &\propto p(\mathbf{r}_n|\mathbf{s}_{0:n}, \mathbf{r}_{0:n-1}). \end{aligned} \tag{197}$$

The second term in the numerator in (196) gets absorbed in the proportionality because the transmitted symbols are independent and uniformly distributed. The denominator in (196) is just a proportionality constant.

The transition kernel turns out to be the incremental weight update factor, when the posterior distribution is used for the symbol draws (Table 21) .

To analyze the convergence we construct a transition probability matrix for each particle and compute an average matrix with the particle weights (198). Since our transmitted symbols are from the set $\{\pm 1\}$, we have a 2×2 probability transition matrix. Each element contains the following transition probability:

$$p(\mathbf{s}_n|\mathbf{s}_{n-1}, \mathbf{r}_n) = \sum_{i=1}^{N_s} p(\mathbf{s}_n^{(i)}|\mathbf{s}_{n-1}^{(i)}, \mathbf{r}_n) w_n^{(i)} \tag{198}$$

As was described in the previous section, the second eigenvalue of this matrix determines the rate of convergence.

6.4 Numerical Results

In our simulations we use frequency selective channels with $d = 2$ and 4 channel coefficients. We implement the SMC algorithm with temporal partitioning of the symbol space [59]. The transmitted symbols are BPSK. The receiver noise is additive white Gaussian noise.

We display the distribution of the second largest eigenvalue with a boxplot and with a plot of the cumulative distribution function. In the boxplot, the horizontal axis indicates the SNR, and the vertical axis is the magnitude of the second largest eigenvalue. The eigenvalue is obtained from the simulations and depends on the particular realizations of the channel, received signal and transmitted data. The line in the middle of the box indicates the median value for the collected samples of λ_2. The upper and the lower edge of the box indicate the upper and the lower quartile of the data. The vertical lines, extending from each end of the boxes, show the extent of the rest of the data. The horizontal lines at the ends of the extending vertical lines indicate an interval of 1.5 times the interquartile range. The red + markers indicate outliers.

As expected, the median of λ_2 decreases as we increase the SNR as seen in the box plots in Figures 31 ,33, and 35. Similar observations can be made about the cdf plots as well (Figures 32 ,34, and 36).

Comparing Figure 32 to Figure 36, one may conclude more particles are needed to track a 4 dimensional channel state vector relative to the $d = 2$ case.

There is another interesting observation that can only be seen in the cdf plot, but not in the boxplot (Figures 37 and 38). For a fixed SNR, as the particle count for the channel state is increased, the eigenvalue distribution tends to get bi-modal. At a low number of channel particles, there tends to be a peak close to $|\lambda_2| = 1$ (look at the slope). This is an indication of an estimation bias. When the number of particles is insufficient, the wrong transmitted symbol is selected.

To have an idea, of how the algorithm tracks the channel coefficients, we display a

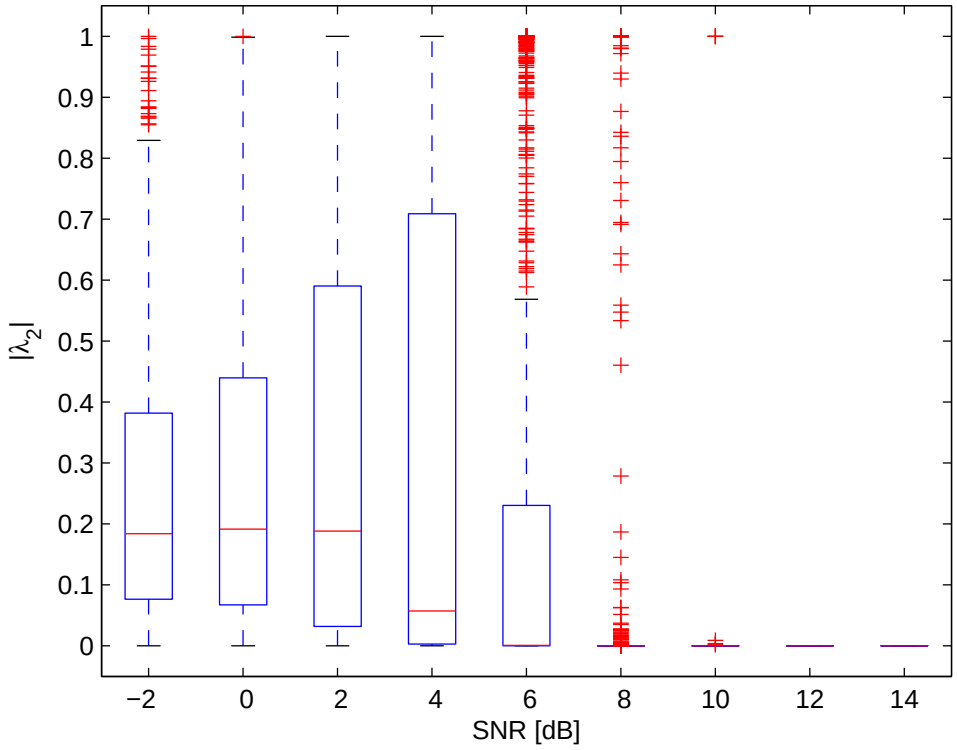


Figure 31: Boxplot of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 2$ taps and 300 channel particles.

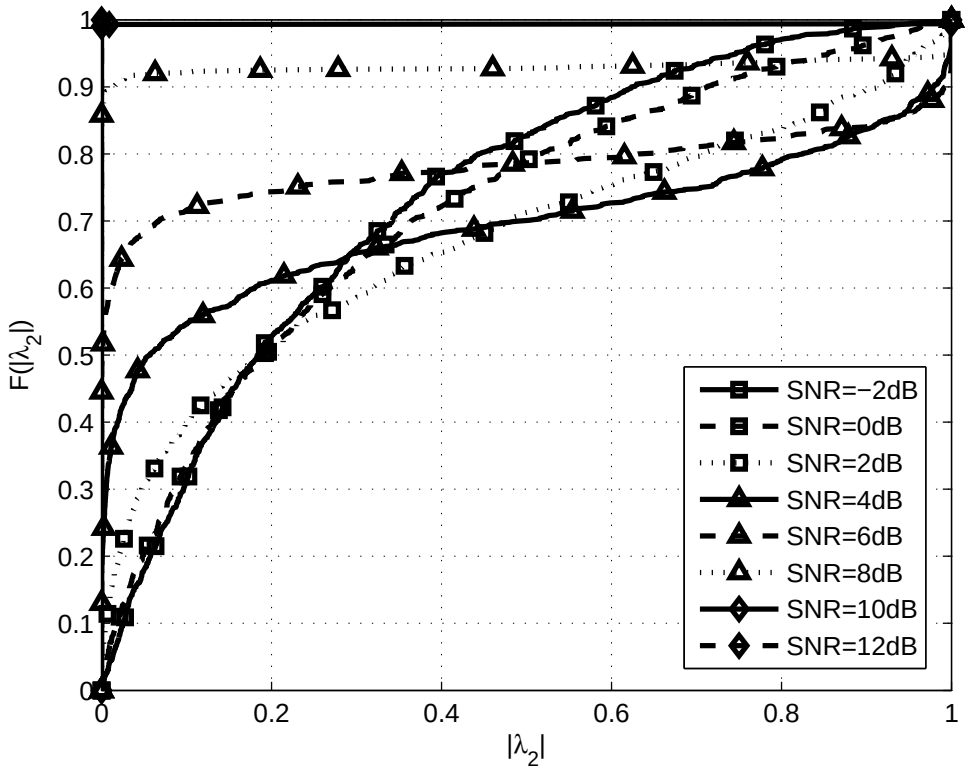


Figure 32: CDF of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 2$ taps and 300 channel particles.

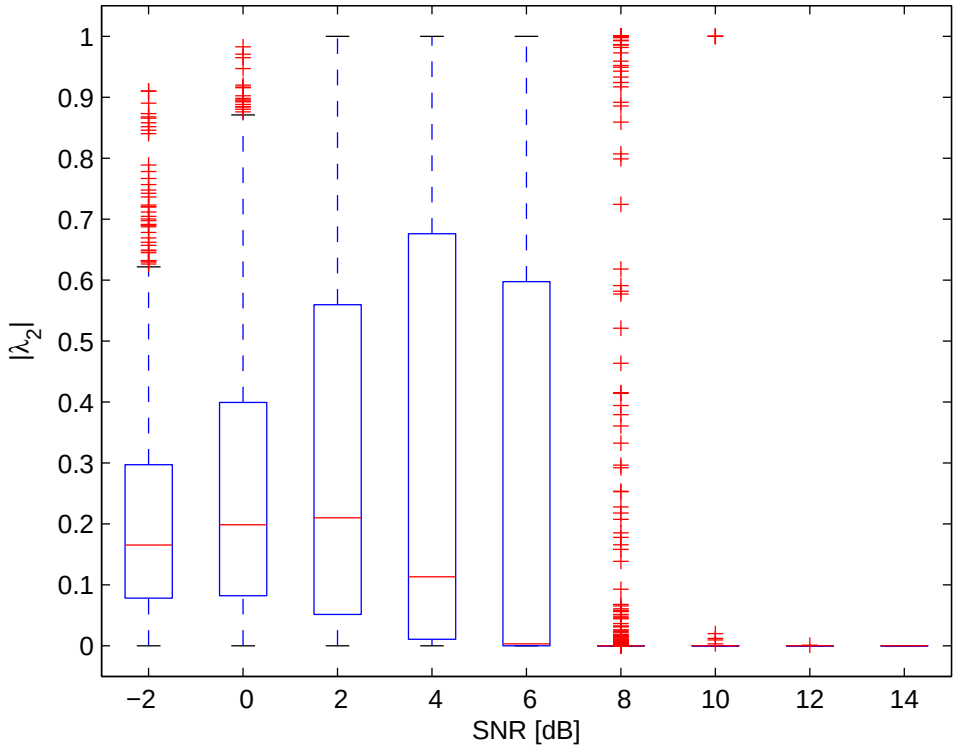


Figure 33: Boxplot of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 3$ taps and 300 channel particles.

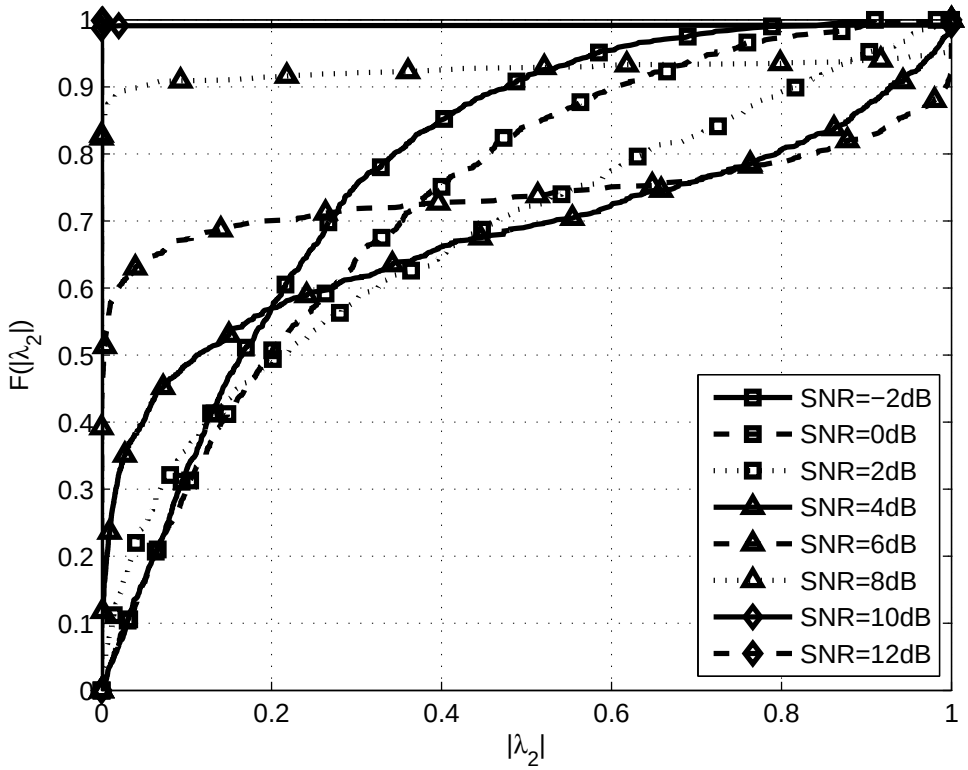


Figure 34: CDF of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 3$ taps and 300 channel particles.

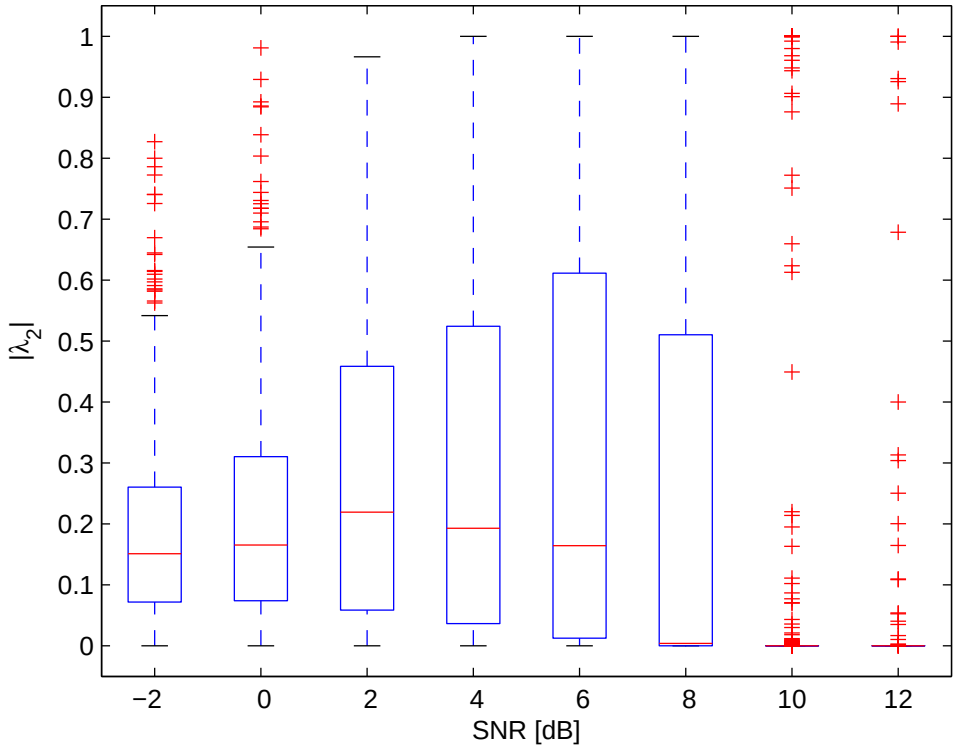


Figure 35: Boxplot of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 4$ taps and 300 channel particles.

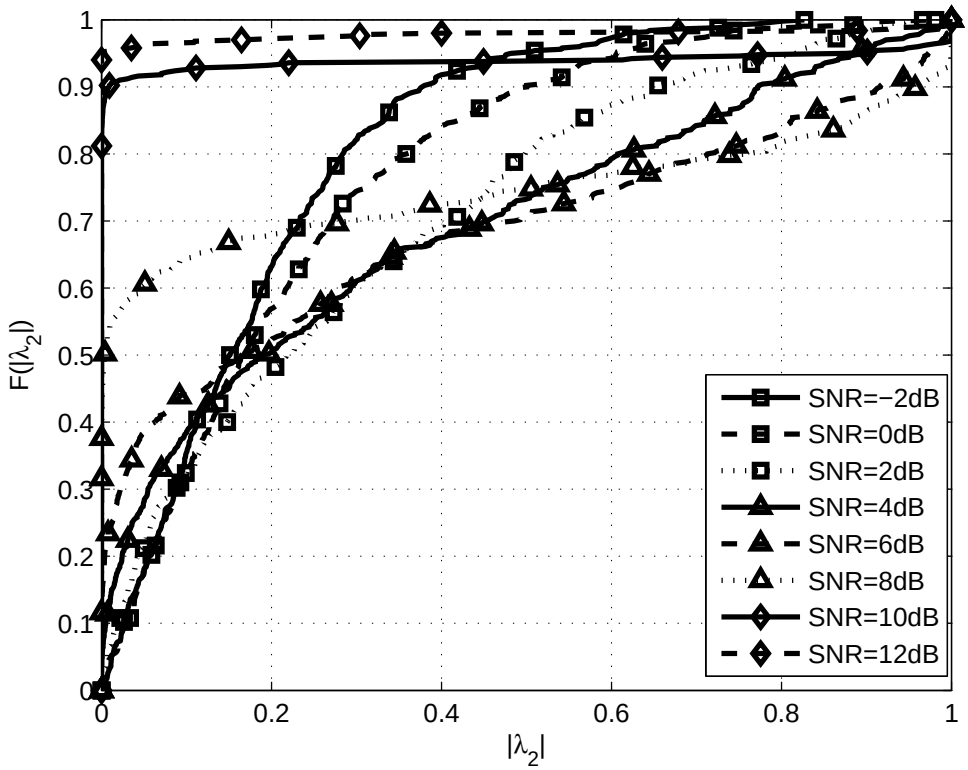


Figure 36: CDF of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 4$ taps and 300 channel particles.

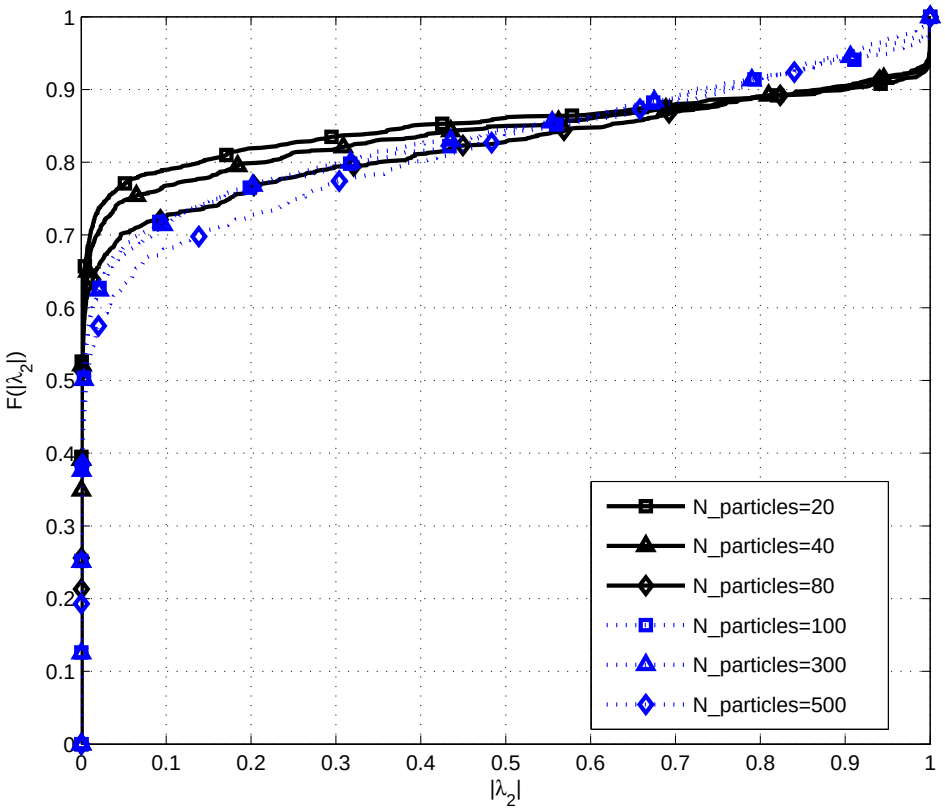


Figure 37: CDF of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 3$ taps, $SNR = 6dB$ and varying number of channel particles.

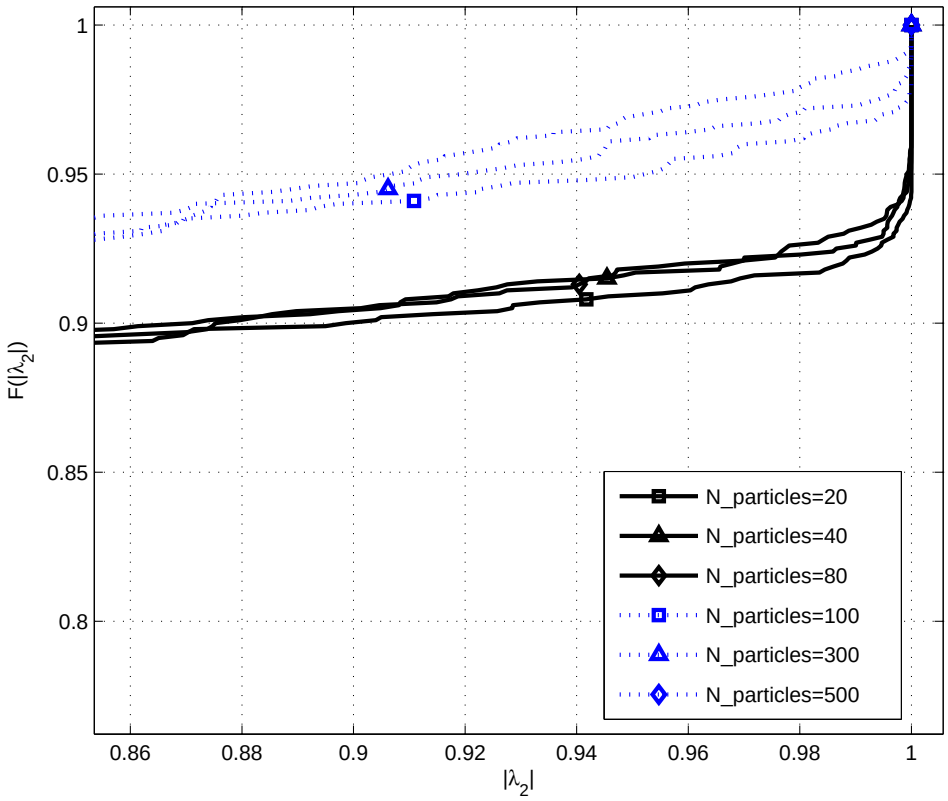


Figure 38: CDF of convergence rate for temporally partitioned SMC on a time-invariant frequency selective channel with $d = 3$ taps, $SNR = 6dB$ and varying number of channel particles. The display is zoomed at the second mode.

sample realization in Figure 39.

Similar conclusions can be drawn from the results of the fading channel case.The ability of the SMC algorithm to track time-varying channels is illustrated in Figure 44. The channel is a slowly fading Rayleigh channel with Doppler coefficient $f_m T = 10^{-3}$. The channel coefficients are symbol spaced and mutually independent. In all simulations there were used 300 particles.

Figures 40 and 41 display the boxplot and the cdf of $|\lambda_2|$ for the 3-tap fading channel. As SNR increases, the eigenvalue gets smaller, which indicates that the number of used particles to estimate the channel may be reduced at high SNR. Similar behavior is observed at the channel with delay spread of 4 symbol intervals (Figures 42 and 43).

The dependence between the second largest eigenvalue and used particles is illustrated in Figures 45 and 46. The bias as the particle number decreases is evident.

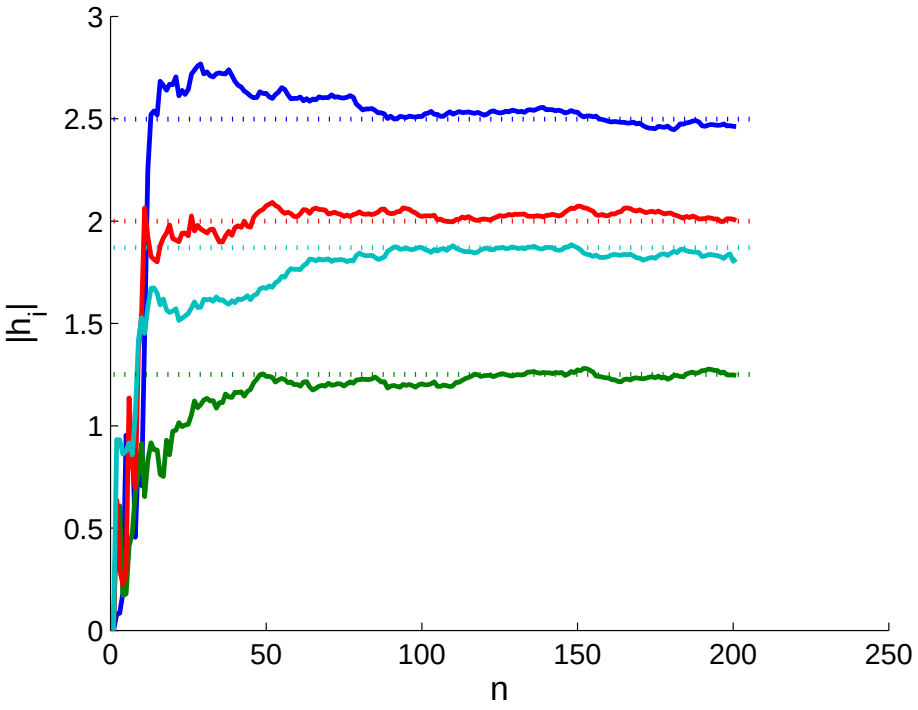


Figure 39: Channel coefficient magnitude estimates. Random initialization. Time invariant frequency selective channel with $d = 4$ taps, SNR=$10dB$.

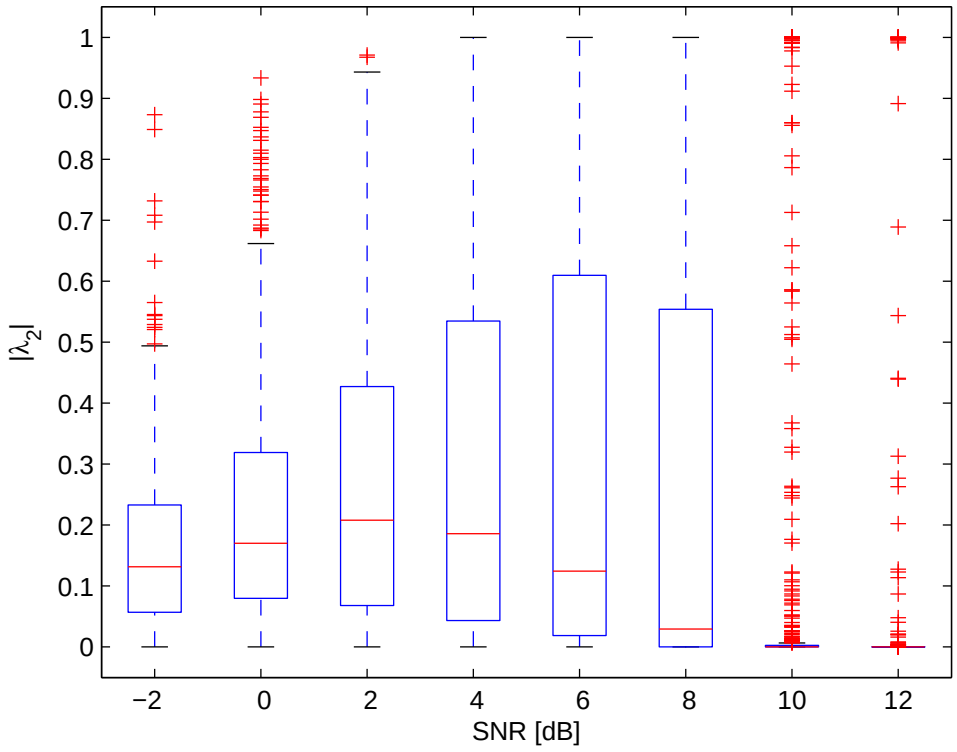


Figure 40: Boxplot of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 3$ taps, 300 channel particles and a Doppler shift of $f_m T = 10^{-3}$.

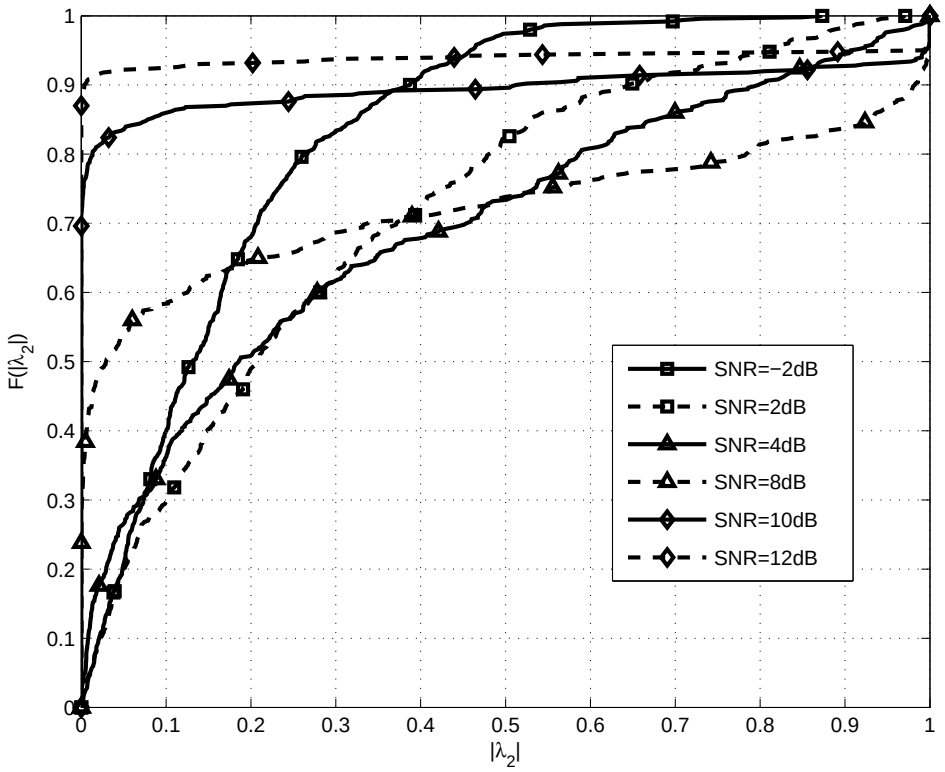


Figure 41: CDF of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 3$ taps, 300 channel particles and a Doppler shift of $f_m T = 10^{-3}$.

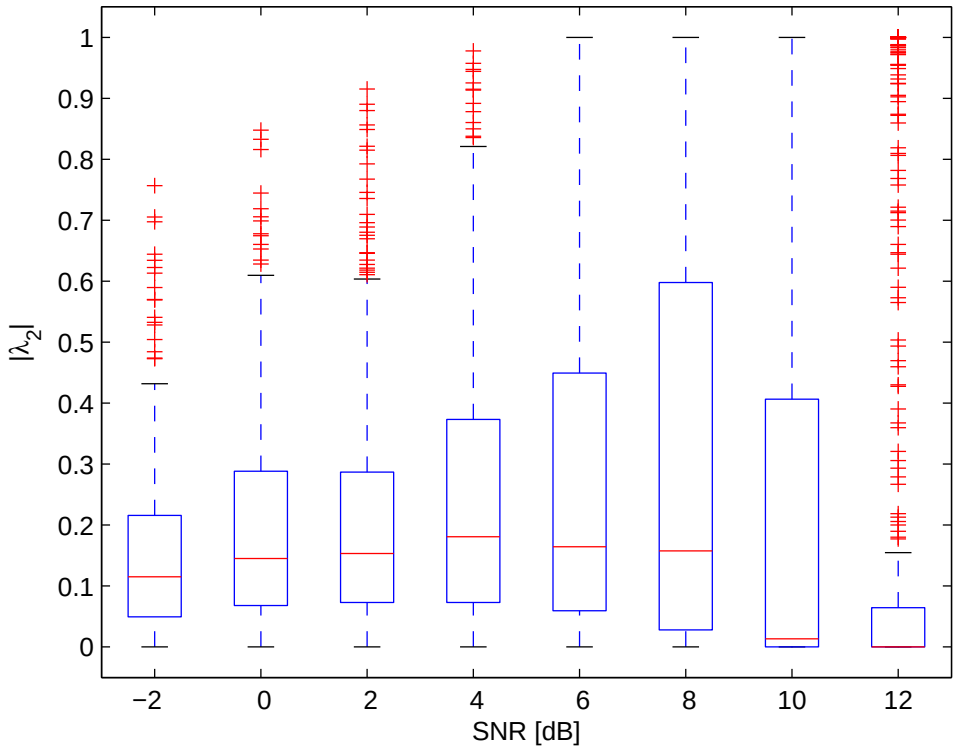


Figure 42: Boxplot of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 4$ taps, 300 channel particles and a Doppler shift of $f_m T = 10^{-3}$.

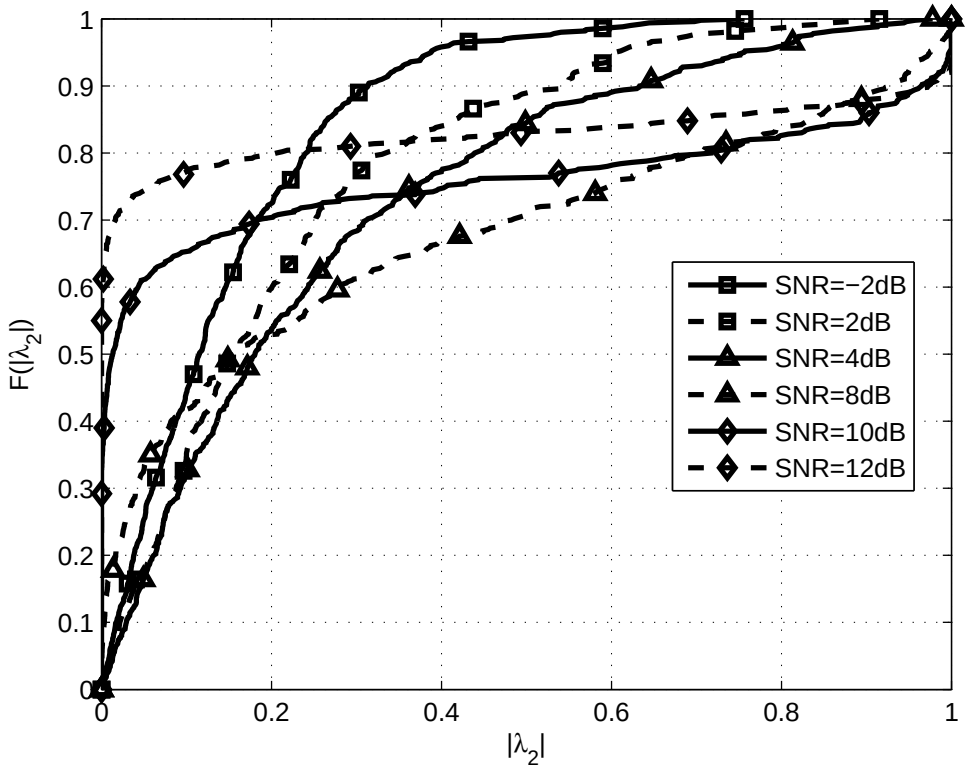


Figure 43: CDF of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 4$ taps, 300 channel particles and a Doppler shift of $f_m T = 10^{-3}$.

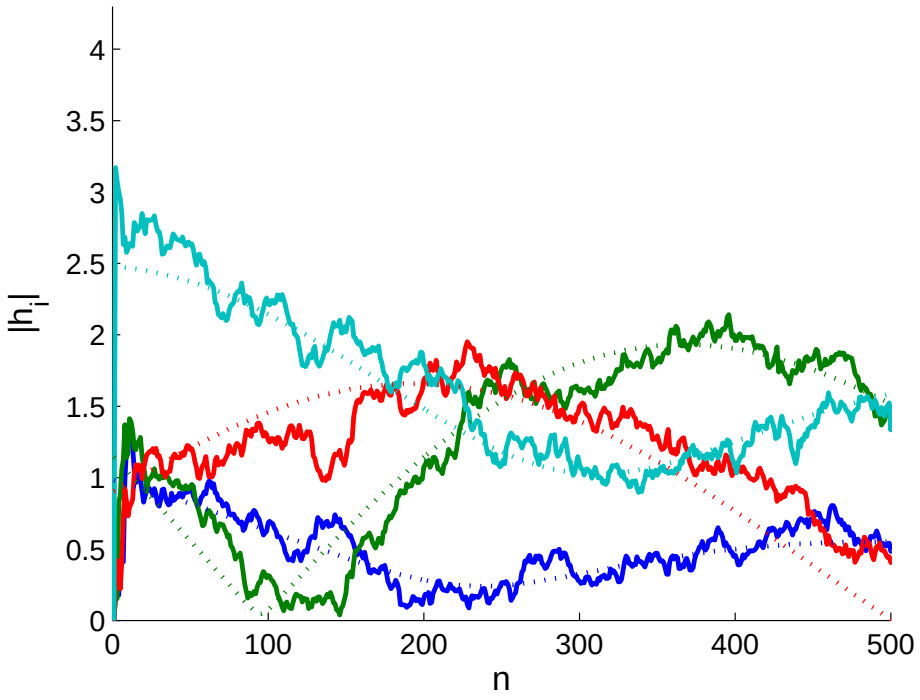


Figure 44: Channel coefficient magnitude estimates. Random initialization. Rayleigh fading frequency selective channel with $d = 4$ taps, SNR=$10dB$, $fmT = 1e-3$.

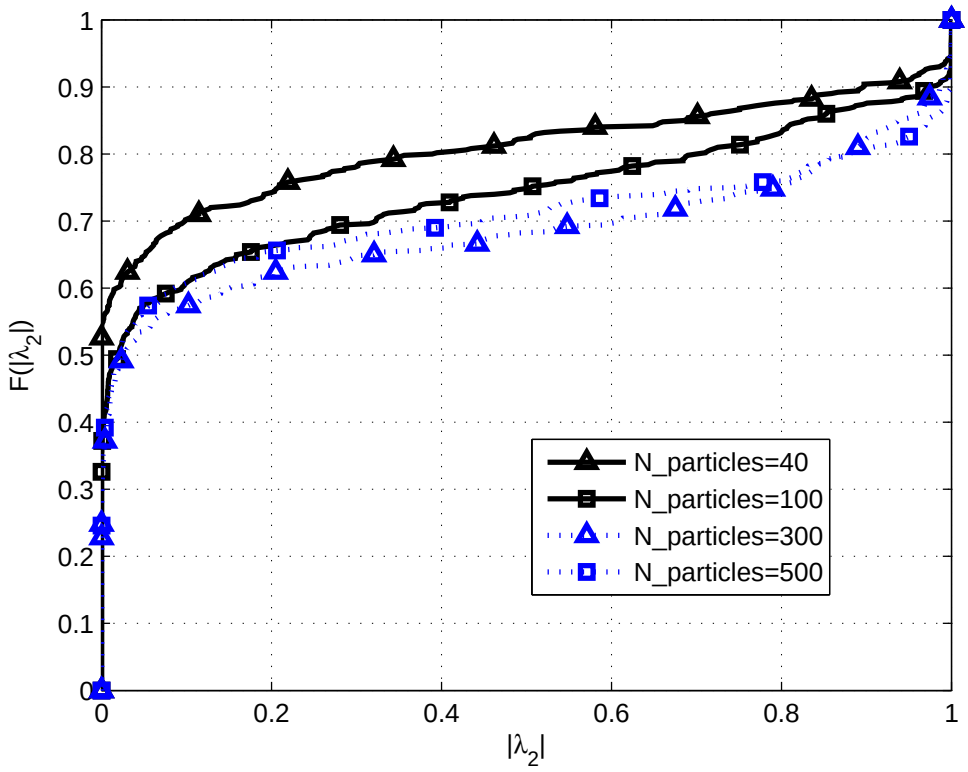


Figure 45: CDF of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 3$ taps, $SNR = 8dB$, a Doppler shift of $f_m T = 10^{-3}$ and varying number of channel particles.

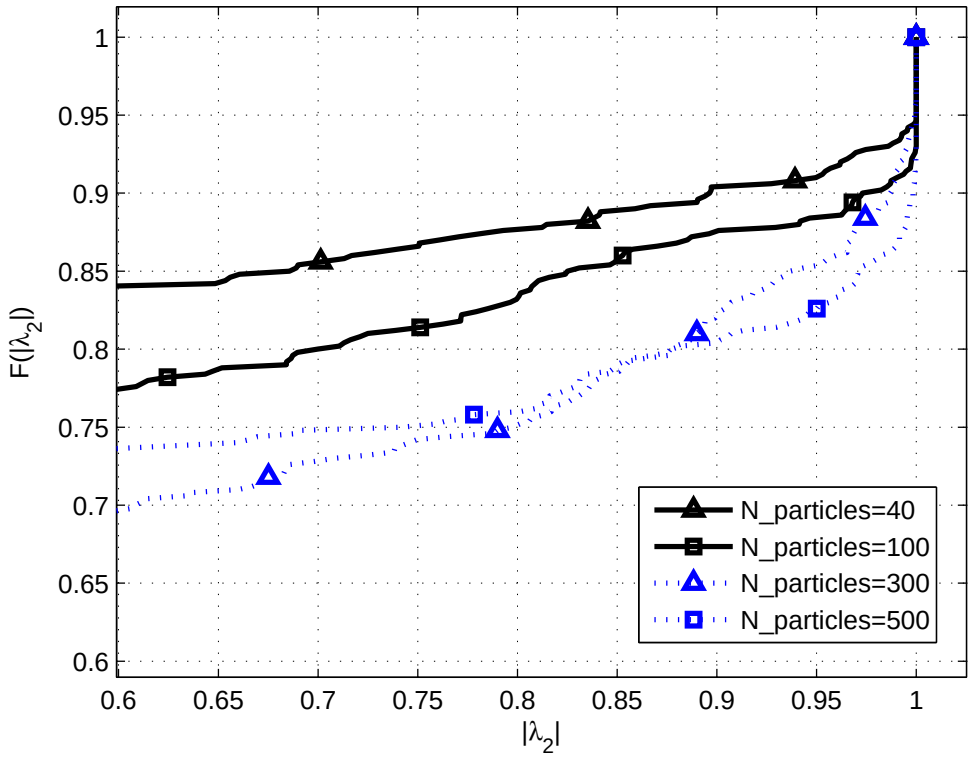


Figure 46: CDF of convergence rate for temporally partitioned SMC on a frequency selective fading channel with $d = 3$ taps, $SNR = 8dB$, a Doppler shift of $f_m T = 10^{-3}$ and varying number of channel particles. The display is zoomed at the second mode.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

In this chapter, we summarize the research results and contributions of this dissertaion, and suggest directions for the future work.

7.1 Contributions and Conclusions

The research reported in this dissertation has given us an in-depth understanding of various detection and estimation algorithms and the challenges they bring for an effective utilization of resources in a telecommunications system.

We have developed a multi-antenna spread spectrum communications system model that utilizes MOCS codes as spreading codes. The MOCS's variable spreading gain and their ability to naturally suppress multiple-access interference (MAI) in multi-antenna transmission cases makes them a superb candidate in multi-antenna communications systems. We specified a coded space-time spreading transmitter. We designed an iterative linear receiver that suppresses MAI and preserves the diversity rate of the system. We analyzed the performance of the receiver analytically and through simulations, and concluded that the system is near-far resistant and the diversity rate is preserved.

Then, we examined blind multiuser detection algorithms. Monte Carlo algorithms were particularly attracting, because of their ability to estimate parameters in non-linear and even non-Gaussian systems. They can be used in dual estimation scenarios, accept a priori information and output soft information about the estimate. These properties make them suitable for utilization in blind and iterative detection scenarios.

We applied the sequential Monte Carlo algorithm to equalize blindly a frequency selective fading channel. Then we found a method to take advantage of the recursive structure of the posterior densities. We simplified the process of evaluating symbol likelihoods and obtaining symbol draws from the posterior distribution. We partitioned the symbol space temporally.

This process significantly accelerates the algorithm by reducing the complexity from being an exponential function of the channel order to only a polynomial function. We merged the symbol estimation and the channel estimation components and obtained a blind receiver. We demonstrated that the performance of this equalizer approaches the performance of the maximum likelihood solution.

Similar recursive structures in posterior density definitions can be found in MIMO systems. After a linear processing stage, analogous techniques for spatially partitioned symbol spaces were used. This algorithm resembles the tree searching algorithm used in sphere decoding algorithms. However, the technique used here is stochastic rather than deterministic. We analyzed the performance of this receiver in MIMO systems with various antenna numbers and spectral utilization factors. We concluded that the designed blind receiver is able to perform significantly better than existing DF receivers with full CSI.

We returned to the MOCS spread spectrum system and applied the simplified SMC algorithm in the receiver design. We constructed a system utilizing an outer convolutional code. We adapted the blind SMC algorithm to accept a priori information from previous iterations and simulated the system. We concluded that the system preserves its spatial diversity order. The BER performance is degraded mainly because of the uncertain channel estimates. We also demonstrated the degraded performance in the case of spatial correlation among the communications channels. We show that the simplifications made to symbol likelihood evaluation and symbol draws reduces the complexity significantly, without a significant degradation compared to the full complexity blind receiver.

Finally, in this work, we investigate the convergence properties of SMC algorithms when utilized in various types of receivers. We propose a metric that is based on the likelihood functions of the symbol samples. Analogous to the MCMC convergence evaluation techniques, we construct a transition probability matrix and extract the second largest eigenvalue in magnitude . Using this metric, we quantify the convergence rate of the SMC algorithms utilized in a particular system. We describe how various parameters, such as noise power and channel fading rate, and system modeling factors such as state space model mismatch, affect the convergence rate. Using this metric, it is possible to point out the factor

that should be improved first to gain convergence speed and estimation accuracy.

7.2 Suggestions for Future Work

Some directions for research that follow the currently designed simplifications to MC algorithms are:

- **Preprocessing:** MC algorithms can be significantly accelerated if the support region of the distribution to be approximated is reduced. Initial processing that localizes the solution to a small region in the parameter space could prove to be useful. The accuracy of such processing need not be precise. Even coarse solutions could be beneficial.

- **Multiuser MIMO receivers:** In this dissertation, we only analyzed a single user, SMC-based MIMO receiver. The scenario with multiple transmitters and interfering links could also be considered.

- **Application in systems with pilot or training sequences:** We did not consider the performance of the algorithm in systems with training sequences. The algorithm could be used to refine the channel estimates. Alternatively, the algorithm could be used to aid in the estimation process when other algorithms fail. A system with a blind SMC algorithm that is turned on only during severe channel conditions could be analyzed and the additional complexity could be quantified.

REFERENCES

[1] Anderson, J. B. and Mohan, S., "Sequential coding algorithms: A survey and cost analysis," *IEEE Transactions on Communications*, vol. COM-32, pp. 169–177, Feb. 1984.

[2] Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[3] Barry, J. R., Lee, E. A., and Messerschmitt, D. G., *Digital Communication*. Norwell: Kluwer, 2004.

[4] Bellini, S., "Bussgang techniques for blind equalization," *Proc. Global Telecommunication Conf.*, pp. 1634–1640, Dec. 1986.

[5] Benveniste, A., Goursat, M., and Ruget, G., "Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications," *IEEE Trans. Autom. Control*, vol. AC-25, pp. 385–399, Jun. 1980.

[6] Berrmon, A. and Plemmons, R. J., "Non-negative matrices in the mathematical sciences," *Classics in applied mathematics 9, Society for industrial and applied mathematics*, 1994.

[7] Berrou, C. and Glavieux, A., "Near optimum error-correcting coding and decoding: Turbo codes," *IEEE Transaction on Communications*, vol. 44, pp. 1261–1271, Oct. 1996.

[8] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near shannon limit error-correction coding and decoding: Turbo codes," *Proceedings of 1993 IEEE Int. Conf. Commun.*, pp. 1064–1070, 1993.

[9] Bresler, Y. and Macovski, A., "Exact maximum likelihood parameter estimation of supperimposed exponential signals in noise," *IEEE Trans. Audio, Speech, Signal Processing*, vol. ASSP-34, pp. 1081–1089, Oct. 1986.

[10] Brillinger, D., "The identification of polynomial systems by means of higher-order spectra," *J. Sound Vib.*, vol. 20, pp. 301–313, 1970.

[11] C. P. Robert, G. C., *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2004.

[12] Cevher, V. and McClellan, J. H., "General direction-of-arrival tracking with acoustic nodes." To appear in *IEEE Trans. on Signal Processing*.

[13] Chan, K. S., "Asymptotic behaviour of the gibbs sampler," *Journal of the American Statistical Association*, vol. 88, pp. 320–326, 1993.

[14] Chen, R. and Li, T., "Blind restoration of linearly degraded discrete signals by gibbs sampler," *IEEE Transactions on SIgnal Processing*, vol. 43, pp. 2410–2413, 1995.

[15] Chen, R. and Liu, J. S., "Mixture kalman filters," *Journal of the Royal Statistical Society*, vol. 62, no. 3, pp. 493–509, 2000.

[16] Chen, R., Liu, J. S., and Wang, X., "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, February 2002.

[17] Chen, R., Wang, X., and Liu, J. S., "Adaptive joint detection and decoding in flat-fading channels via kalman filtering," *IEEE Transsactions on Information Theory*, vol. 46, Sep. 2000.

[18] Cheng, J. and Ottosson, T., "Linearly approximated log-map algorithms for turbo coding," *Proc., IEEE Veh. Tech. Conf., (Houtson, TX)*, May 2000.

[19] Ding, Z., Kennedy, R. A., Anderson, B. D. ., and C. R. Johnson, J., "Ill-convergence of godard blind equalizers in data communication systems," *IEEE Trans. Commun.*, vol. 39, pp. 1313–1327, Sep. 1991.

[20] Ding, Z. and Li, Y., *Blind Equalization and Identification*. New York: Marcel Dekker, 2001.

[21] Dong, B., Wang, X., and Doucet, A., "A new class of soft MIMO demodulation algortihms," *IEEE Trans. on Signal Processing*, vol. 51, no. 11, November 2003.

[22] Doucet, A., Freitas, N., and Gordon, N., eds., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[23] Foschini, G. J., "Layered space-time architecture for wireless communication in a fading environment when using multi-antenna elements," *Bell Labs Technical Journal*, 1996.

[24] Foschini, G. J. and Gans, M. J., "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[25] Foschini, G. J. and Miljanic, Z., "A simple distributed autonomous power control algorithm and its convergence," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 641–646, Nov. 1993.

[26] Friedlander, B. and Porat, B., "Asymptotically optimal estimation of ma and arma parameters of non-gaussian processes from higher-order moments," *IEEE Trans. Automat. Control*, vol. 35, pp. 27–35, Jan 1990.

[27] Ghosh, M. and Weber, C. L., "Maximum-likelihood blind equalization," *Opt. Eng.*, vol. 31, pp. 1224–1228, Jun. 1992.

[28] Giannakis, G. and Mendel, J., "Identification of nonminimum phase systems using higher order statistics," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 360–377, Mar. 1989.

[29] Giannakis, G. B., Hua, Y., Stoica, P., and Tong, L., *Signal Processing Advances in Wireless and Mobile Communications*, vol. 1. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[30] Godard, D. N., "Self-recovering equalization and carrier tracking in twodimensional data communication systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867–1875, Nov. 1980.

[31] Golay, M. J. E., "Complementary series," *IRE Transactions on Information Theory*, vol. IT-7, pp. 82–87, Apr. 1961.

[32] Guo, D. and Wang, X., "Blind detection in mimo systems via sequential Monte Carlo," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, April 2003.

[33] Guo, D. and Wang, X., "Wavelet-based sequential Monte Carlo blind receivers in fading channels with unknown channel statistics," *IEEE Trans. on Signal Processing*, vol. 52, no. 1, January 2004.

[34] Hasibi, B. and Vikalo, H., "On the sphere decoding algorithm: Part i, the expected complexity," *To appear in IEEE Transactions on Signal Processing*, 2004.

[35] Hastings, W., "Monte carlo sampling methods using markov chains and their application," *Biometrica*, vol. 57, pp. 97–109, 1970.

[36] Hatzinakos, D. and Nikias, C., "Estimation of multipath channel response in frequency selective channels," *IEEE J. Select. Areas Commun*, vol. 7, pp. 12–19, Jan. 1989.

[37] Haykin, S., *Adaptive Filter Theory*. Upper Saddle River: Prentice Hall, 3 ed., 1996.

[38] Honig, M. L. and Poor, H. V., *Wireless communications: A signal processing perspective*. Upper Saddle River, NJ: Prentice Hall, 1998.

[39] Honig, M. L. and Tsatsanis, M. K., "Adaptivr techniques for multiuser cdma receivers," *IEEE Trans. Signal Process.*, vol. 17, pp. 49–61, May 2000.

[40] Host-Madsen, A. and Wang, X., "Performance of blind and group blind multiuser detection," *IEEE Transactions on Information Theory*, vol. 48, pp. 1849–1872, Jul. 2002.

[41] Hua, Y., "Fast maximum likelihood for blind identification of multiple fir channels," *Proc. 28th Asilomar Conf. Signals, Systems, and Computers*, Nov. 1994.

[42] Hua, Y., "Fast maximum likelihood for blind identification of multiple fir channels," *IEEE Trans. Signal Processing*, vol. 44, pp. 661–672, Mar. 1996.

[43] Huang, Y., Zhang, J. M., and Djuric, P. M., "Bayesian detection for BLAST," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1086–1096, Mar. 2005.

[44] Iltis, R. A., "A bayesian maximum-likelihood sequence estimation algorithm for a priori unknown channels and symbol timing," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 579–588, Apr. 1992.

[45] Iltis, R. A., Shynk, J. J., and Giridhar, K., "Bayesian algorithms for blind equalization using parallel adaptive filtering," *IEEE Trans. Commun.*, vol. 42, pp. 1017–1032, Feb.–Apr. 1994.

[46] Jayaweera, S. K. and Poor, H. V., "Iterative multiuser detection for space-time coded synchronous CDMA," *Vehicular Technology Conference, 2001*, pp. 2736–2739, Oct. 2001.

[47] Liu, J. S., "The collapsed gibbs sampler with applications to a gene regualtion problem," *Journal of the American Statistical Association*, vol. 89, pp. 958–966, Sep. 1994.

[48] Liu, J. S., *Monte Carlo Strategies in Scientific Computing*. New York: Springer-Verlag, 2001.

[49] Liu, J. S. and Chen, R., "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, vol. 90, pp. 567–576, 1995.

[50] Liu, J. S. and Chen, R., "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, Sep. 1998.

[51] MacCormick, J. and Isard, M., "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proceedings of the Europen Conference on Computer Vision*, 2000.

[52] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equations of state calculations by fast computing machines," *J. Chem. Phys*, vol. 21, pp. 1087–1092, 1953.

[53] Metropolis, N. and Ulam, S., "The monte carlo method," *J. American Statist. Assoc.*, vol. 44, pp. 335–441, 1953.

[54] Miguez, J. and Djuric, P. M., "Blind equalization of frequency selective channels by sequential importance sampling," *IEEE Trans. on Signal Processing*, vol. 52, no. 10, October 2004.

[55] Naguib, A. F. and Seshadri, N., "Combined interference cancellation and ML decoding of space-time block coding," *Proc. of 7th Communication Theory Mini Conference, held in conjunction with Globecom98*, Nov. 1999.

[56] Orton, M. and Fitzgerald, W., "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 216–223, February 2002.

[57] Ozgur, S. and Williams, D. B., "Multi-user detection for mutually orthogonal sequences with space-time coding," *Proc. of the IEEE Vehicular Technology Conference*, Oct. 2003.

[58] Ozgur, S. and Williams, D. B., "An iterative multi-user detector for coded mutually orthogonal complementary sets with space-time coding," *Proceedings of DSP Workshop*, Aug. 2004.

[59] Ozgur, S. and Williams, D. B., "Temporal partition particle filtering for multiuser detectors with mutually orthogonal sequences," *ICASSP 2005*, Mar. 2005.

[60] Poor, H. V. and Wornell, G. W., *Wireless Cmmunications*. Upper Saddle River: Prentice Hall, 1998.

[61] Porat, B., *Digital Processing of Random Signals*. Prentice-Hall,Englewood Cliffs, NJ, 1993.

[62] Proakis, J. G., *Digital Communications*. New York: McGraw-Hill, 1995.

[63] Sato, Y., "A method of self-recovering equalization for multi-level amplitude modulation," *IEEE Trans. Communications*, vol. COM-23, pp. 679–682, Jun. 1975.

[64] Schervish, M. J. and Carlin, B. P., "On the convergence of successive substitution sampling," *Journal of Computational and Graphical Statistics*, no. 1, pp. 111–127, 1992.

[65] Schodorf, J. B. and Williams, D. B., "A constrained optimization approach to multiuser detection," *Signal Processing, IEEE Transactions on*, vol. 45, pp. 258–262, Jan. 1997.

[66] Seshadri, N., "Joint data and channel estimation using fast blind trellis search techniques," *Proc. Globecom'90*, pp. 1659–1663, 1991.

[67] Seshadri, N., "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Commun.*, vol. 42, pp. 1000–1011, Feb.–Apr. 1994.

[68] Simmons, A. J., "Breath-first trellis decoding with adaptive effort," *IEEE Transactions on Communications*, vol. 38, pp. 3–12, Jan. 1990.

[69] Slock, D., "Blind fractionally-spaced equalization, perfect reconstruction filterbanks, and multilinear prediction," *ICASSP*, 1994.

[70] Slock, D. and Papadias, C. B., "Further results on blind identification and equalization of multiple fir channels," *IEEE Proc. Intl. Conf. Acoustics, Speech, Signal Processing*, pp. 1964–1967, Apr. 1995.

[71] Sparo, D. I. and Sur, M. G., "On the distribution of roots of random polynomials," *Vestnik Moskov. Univ. Ser. I Mat. Meh.*, no. 3, pp. 40–43, 1962.

[72] Stüber, G. L., *Principles of Mobile Communication*. Boston: Kluwer Academic Publishers, 2nd ed., 2001.

[73] Tanner, M. A., *Tools for Statistics Inference*. New York: Springer-Verlag, 1991.

[74] Tarokh, V., Jafarkhani, H., and Calderbank, A. R., "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, pp. 1456–67, Jul. 1999.

[75] Tarokh, V., Jafarkhani, H., and Calderbank, A. R., "Space-time block coding for wireless communications: performance results," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 451–460, Mar. 1999.

[76] Tarokh, V., Seshadri, N., and Calderbank, A. R., "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, Mar. 1998.

[77] TIERNEY, L., "Markov chains for exploring posterior distributions," *Annals of Statistics*, vol. 22, pp. 1701–1728, 1994.

[78] TONG, L. and PERREAU, S., "Multichannel blind identification: From subspace to maximum likelihood methods," *Proc. IEEE,*, vol. 86, pp. 1951–1968, Oct. 1998.

[79] TONG, L., XU, G., and KAILATH, T., "Blind identification and equalization based on second-order statistics: A time domain approach," *IEEE Trans. Inf. Theory*, vol. 40, pp. 340–349, Jun. 1994.

[80] TREICHLER, J. R. and AGEE, B. G., "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Audio, Speech, Signal Processing*, vol. ASSP-31, pp. 459–472, Apr. 1983.

[81] TSENG, C. C., "Complementary sets of sequences," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 644–652, Sep. 1972.

[82] TUGNAIT, J. K., "Blind equalization and estimation of digital communication fir channels using cumulant matching," *IEEE Trans. Commun*, vol. 43, pp. 1240–1245, Feb.-Apr. 1995.

[83] VERDU, S., *Multiuser detection*. New York: Cambrige University Press, 1998.

[84] WANG, X. and CHEN, R., "Adaptive bayesian multiuser detection for synchronous cdma in gaussian and impulsive noise," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2013–2028, Jul. 2000.

[85] WANG, X. and CHEN, R., "Blind turbo equalization in gaussian and impulsive noise," *IEEE Transactions on Vehicular Technology*, vol. 50, pp. 1092–1105, Jul. 2001.

[86] WANG, X., CHEN, R., and LIU, J. S., "Blind adaptive multiuser detection in mimo systems via monte carlo," *Proc. 2000 Conf. Inorm. Sci. Syst.*, Mar. 2000.

[87] WANG, X., CHEN, R., and LIU, J. S., "Monte carlo signal processing for wireless communications," *Journal on VLSI Signal Processing*, vol. 30, pp. 89–105, Jan./ Mar. 2002.

[88] WANG, X. and HOST-MADSEN, A., "Group-blind multiuser detection for uplink cdma," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 2082–2089, Dec. 1999.

[89] WANG, X. and POOR, H. V., "Joint channel estimation and symbol detection in rayleigh flat fading channels with impulsive noise," *IEEE Communications Letters*, vol. 1, pp. 19–21, Jan. 1997.

[90] WANG, X. and POOR, H. V., "Blind equalization and multiuser detection in dispervie CDMA channels," *IEEE Transactions on Communications*, vol. 6, pp. 91–103, Jan. 1998.

[91] WANG, X. and POOR, H. V., "Blind multiuser detection: a subspace approach," *IEEE Transactions on Information Theory*, vol. 44, pp. 677–690, March 1998.

[92] Wang, X. and Poor, H. V., "Blind joint equalization and multiuser detection for DS-CDMA in unknown correlated noise," *IEEE Transcations on Circuit Systems II: Analog Digital Signal Processing*, vol. 46, pp. 886–895, July 1999.

[93] Wang, X. and Poor, H. V., "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions On Communications*, vol. 47, pp. 1046–1061, July 1999.

[94] Wang, X. and Poor, H. V., "Space-time multiuser detection in multipath CDMA channels," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2356–2374, Sep. 1999.

[95] Wang, X. and Poor, H. V., *Wireless communications systems: Advanced techniques for signal reception*. Upper Saddle River: Prentice Hall, 2004.

[96] Winters, J. H., "Smart antennas for wireless systems," *IEEE Personal Communications*, pp. 23–27, Feb. 1998.

[97] Winters, J. H. and Gans, M. J., "The range increase of adaptive versus phased arrays in mobile radio systems," *Proc. of IEEE Asilomar Conference on Signals, Systems, and Computers*, pp. 109–115, Oct. 1994.

[98] Wong, M. A. T. W. H., "The calculation of posterior distribution by data augmentation," *Journal of the American Statistical Association*, vol. 82, pp. 528–550, 1987.

[99] Woodward, G. K., *Adaptive detection for DS-CDMA*. PhD thesis, University of Sydney, 1999.

[100] Yang, J., Werner, J. J., and Dumont, G., "The multimodulus blind equalization algorithm," *Proc. Thirteenth Int. Conf. Digital Signal Process.*, Jul. 1997.

[101] Yang, J., Werner, J. J., and Dumont, G., "The multimodulus blind equalization and its generalized algorithms," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 997–1015, Jun. 2002.

[102] Yang, Z. and Wang, X., "Bayesian monte carlo multiuser receivr for space-time coded multi-carrier cdma systems," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1625–1763, Aug. 2001.

[103] Yang, Z. and Wang, X., "Blind turbo multiuser detection for long-code multipath CDMA," *IEEE Transactions on Communications*, vol. 50, pp. 112–125, Jan. 2002.

[104] Yang, Z. and Wang, X., "A sequential Monte Carlo blind receiver for OFDM systems in frequency selective fading channels," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, February 2002.

VITA

Soner Ozgur received his B.S Degree in Electrical Engineering from University of Southern California, Los Angeles, CA in 1998. During the summer of 2005 he was employed by Qualcomm Inc. He will receive his PhD Degree in Electrical Engineering from Georgia Institute of Technology in 2006. His research interests include multiuser detection, equalization, channel estimation, blind methods, particularly those based on Monte-Carlo methods, adaptive filtering, and array signal processing.