ITERATIVE RECEIVER DESIGN FOR MIMO-OFDM SYSTEMS VIA SEQUENTIAL MONTE CARLO (SMC) TECHNIQUES

BAY LAY KHIM

NATIONAL UNIVERSITY OF SINGAPORE

2007

ITERATIVE RECEIVER DESIGN FOR MIMO-OFDM SYSTEMS VIA SEQUENTIAL MONTE CARLO (SMC) TECHNIQUES

BAY LAY KHIM
(B.Eng.(Hons.), NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACKNOWLEDGEMENTS

Two years have passed, seemingly as fast as the blink of an eye. Throughout these two years, I have learnt a lot and this is all thanks to my supervisors Dr. Nallanathan Arumugam and Prof Hari Krishna Garg. The guidance offered by Dr. Nallanathan has instilled in me an even stronger inclination towards research. The advices and tips learnt are life long.

I would also like to thank my family who have always been and will always be there for me. With their support, I was able to make it through the periods of stress where everything seems to occur at the same time.

For my lunchmate, Elisa, thanks for having lunch with me almost everyday. It's great to have someone as great as you to talk to! To all my lab mates, all of you are so inspiring!

Bay Lay Khim

June 2007

TABLE OF CONTENTS

SUMMARY

From a Bayesian viewpoint, the hidden state variables of a dynamic system can be estimated by reconstructing the posterior probability density function of those variables, using information from the measurements available. Kalman filters are typically being employed if the systems involved are linear. However if non-linear systems or non-linear noise are involved, Sequential Monte Carlo (SMC) techniques will have to be used.

SMC performs online estimations via Monte Carlo techniques. Conventionally, SMC techniques utilize sequential importance sampling and resampling. Through recursive sampling and updating, the desired probability density function is represented as a set of random particles with associated weights. It is common that after a few iterations, only one particle with significant weight is left. This leads to a wastage of computational resources as significant efforts are used to update particles that have negligible contribution to the desired function. This phenomenon, also know as degeneracy, is inevitable as the variance of the importance weights of the particles increases with time. Degeneracy can be curbed by performing resampling, which duplicates particles with large weights and removes particles with negligible weights. However resampling is computationally intensive and causes problems such as impoverishment of diverse trajectories and difficulty in implementing the SMC algorithm in parallel. In this work, an algorithm that circumvents resampling and hence avoiding the associated problems is proposed.

In the proposed algorithm, SMC technique is used at the first stage of an iterative receiver to address the issue of symbol detection in a differentially encoded MIMO-OFDM system over multipath frequency selective channels. Both rate ½ convolutional coded and LDPC coded MIMO-OFDM systems are considered. After MAP decoding, the symbol probabilities are computed from the bit probabilities and are sent back to the SMC detector to serve as the a priori symbol probabilities. Periodic termination of the differential phase trellis is employed and the promising simulation results justify the elimination of the resampling step.

The effect of different antenna arrangements, different termination periods and various power delay profile channels are also investigated. It is seen that with the same total number of transmit and receive antennas, the system with the most number of receive antennas performs the best. It is also observed that with a smaller termination period, the performance is the best but this is at the expense of a higher overhead. The proposed algorithm performs better under a uniform than an exponential power delay profile channel. It is also compared to a system with SMC detection and with resampling performed. It is seen that the proposed system is able to achieve similar performance.

Using the periodically terminated symbols as pilot symbols, channel estimation is performed. Through the simulations, it is seen that the performance of the various systems are close to their respective lower channel bounds that are obtained by assuming that the receiver has perfect knowledge of the channel state information (CSI).

The proposed algorithm enables the computationally intensive resampling step to be avoided and the promising results of the proposed algorithm show that it is a viable alternative to be considered for MIMO-OFDM systems with differential QPSK. Another contribution of this work is that the termination states used can serve as pilot symbols for channel estimation.

This work has been submitted to the International Conference on Communications, 2008.

LIST OF TABLES

LIST OF FIGURES

LIST OF COMMONLY USED SYMBOLS

f_C	Carrier frequency
T_m	Delay spread
B_C	Coherence bandwidth
B_D	Doppler spread
T_C	Coherence time
$f_{\max}$	Maximum Doppler shift
v	Speed of vehicle
c	Speed of light
N_C	Number of subcarriers
T_{sym}	Duration of an OFDM or a MIMO-OFDM symbol
T_S	Sampling duration
h	Discrete time channel response
L	Length of channel response
N_T	Number of transmit antennas
N_R	Number of receive antennas
N_{Pf}	Separation between the pilot symbols along the frequency axis
N_{Pt}	Separation between the pilot symbols along the time axis
Ω	Number of Monte Carlo particles
K	Termination period

LIST OF COMMONLY USED ABBREVIATIONS

CSI	Channel State Information
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
LDPC	Low Density Parity Check
LLR	Log Likelihood Ratio
LS	Least Squares
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
PACE	Pilot-symbol Aided Channel Estimation
PF	Particle Filtering
SC	Subcarrier
SIS	Sequential Importance Sampling
SMC	Sequential Monte Carlo

CHAPTER 1

INTRODUCTION

1.1 Background

Orthogonal Frequency Division Multiplexing (OFDM) is gaining popularity in many areas as it is able to support high data rates and is robust towards multipath fading effects. The idea of using parallel data streams and FDM started off in the mid 60s [1-2]. To ensure efficient usage of the spectrum, the subcarriers (SCs) are overlapped and the orthogonality of the SCs aids in combating multipath delays and amplitude distortion. This idea was further extended to incorporate Discrete Fourier Transform (DFT) into the modulation and demodulation processes [3] where it helps to eliminate the need for a bank of oscillators and coherent demodulators. The beauty of using DFT lies in the completely digital implementation that results. The concept was further improved by the use of FFT [4], which allows high speed processing. With the recent advances in VLSI technology, chips that perform high speed and large size FFT are readily available at a low cost. This helps to elevate the status of OFDM to become a very promising technology for high speed data transmission over wireless mobile channels. In fact OFDM is being widely used and has been adopted in high speed wireless applications such as IEEE 802.11a LAN and IEEE 802.16a LAN/MAN [5-7].

OFDM can be employed in a multiple transmit and multiple receive antenna scheme to increase capacity or to enhance the diversity gain [8]. It has been shown that in a multiple-input and multiple-output (MIMO) system, the system capacity can be improved by a factor of the minimum of the number of transmit and the number of receive antennas [9-10]. Space Division Multiplexing (SDM) is a technique that achieves high capacity by transmitting different data symbols simultaneously on the different transmit antennas [11], in so doing, it creates spatial diversity and helps to combat multipath fading [12].

In transmitting a signal from a location to another, the environment that exists between these two locations determines the quality of the received signal. There are generally two types of channel models to characterize the mobile radio channel. First is the large-scale channel model, which takes into account the path loss and shadowing effects while the other is the small-scale channel model, which considers the signal variations in a small local area [13]. In this work, only small scale effects, also known as multipath fading, is considered. Fading is caused by multiple copies of the same signal that arrives at the receiver with different amplitudes, phases and time delays. The three most important effects of fading are, rapid variations in the strength of the signal over a short duration of time, time dispersion due to the propagation delays of the different paths and if the various multipaths have different Doppler spreads, this will also lead to different frequency modulations of the signal [14].

From a Bayesian viewpoint, estimation for the hidden states of dynamic systems can be performed through the reconstruction of the posterior density function of those states by taking into account all the available measurements [15]. Sequential Monte Carlo (SMC) methods [16-21] have been used to perform blind equalization [18], detection and decoding in fading environments [22-28] and multiuser detection in CDMA systems [29]. SMC performs online estimations via techniques such as sequential importance sampling (SIS) and resampling. The desired probability density function is represented by a set of random particles and associated weights. Regions of high probabilities will be represented by particles with larger weights while regions of low probabilities will be represented by particles with smaller weights. After a few iterations of sampling and updating, it is common to find that only one particle of a significant weight is left. This phenomenon is known as degeneracy and it is inevitable whenever SIS is involved. However it can be curbed by performing resampling, which removes particles with negligible weights and replicates particles with large weights. On the other hand, resampling is computationally intensive and introduces problems such as impoverishment of diverse trajectories and difficulty in implementing the SMC algorithm in parallel [30].

With the advent of Turbo codes [31-32], iterative (turbo) receivers have been receiving lots of attention because of their ability to handle soft inputs and outputs [33-34] and hence leading to better performances over systems using hard decisions. Iterative receivers have been employed for various roles such serial

concatenation decoding, multiuser detection and joint source and channel decoding [35-36].

Transmitting a radio signal over a multipath fading channel will result in the signal being received with an unknown phase and amplitude. Channel estimation is essential to ensure that the signal is detected and demodulated correctly. Channel estimation can be performed with the aid of pilot symbols, also known as pilot-symbol aided channel estimation (PACE). Pilot-symbol assisted modulation (PSAM) for a single carrier under flat fading environment was first analyzed in [37] while PACE for OFDM was first demonstrated in [38] and subsequently [39-49]. The pilot symbols can be scattered across the 2-dimensional (2-D) time-frequency lattice, i.e. across different OFDM symbols and different tones. Estimation is first performed at the locations of the pilot tones and these estimates are interpolated across the different tones to obtain the channel estimates at the data SCs. Subsequently, these estimated parameters are further interpolated across the different OFDM symbols. The estimation can be performed using the Minimum Mean Square Error (MMSE) method or Least Squares (LS) method. MMSE estimation performs better than Least Squares (LS) method as the latter suffers from high mean square errors [50].

Channel estimation is especially challenging in the case of MIMO-OFDM system, where different signals are transmitted from each transmit antenna, causing the received signal to be a superposition of the different transmitted signals. However,

it shall seen in Chapter 5 that channel estimation for MIMO-OFDM systems can be extended from the available techniques for single-input single-output OFDM channel estimations.

1.2 Contribution of Thesis

In this piece of work, resampling which is normally present in the SMC methods is circumvented so as to avoid the problems associated with it. To do this, the proposal is to periodically terminate the stream of the differentially encoded symbols at desired states by inserting certain symbols into the stream. It is well known that the variance of the importance weights of the particles can only increase with time [30]. With periodic termination, the variance of the weights is prevented from increasing by huge amounts as imputations are only carried over a short period, as such degeneracy is curbed and therefore resampling is no longer necessary.

Though periodic termination results in overheads, these overheads can be put to good use by serving as pilot symbols to aid in the channel estimation process. The amount of overheads can be lowered with a larger termination period. However the performance of the system degrades with increase in termination period. The effect of the termination period on the performance of the system is investigated and the simulation results are shown in this thesis.

The proposed algorithm is also compared with a system that employs resampling. Through simulations, it is found that resampling only adds a slight improvement to the performance as compared to the proposed algorithm. Therefore, considering the added complexity and the problems associated with resampling, one might prefer to skip resampling at the expense of a very slight tradeoff in performance.

PACE is employed for the MIMO-OFDM system by multiplexing known pilot symbols into the data stream to be transmitted. Therefore the receiver is able to estimate the channel at any instance given the observations provided by the pilot symbols. As the pilot symbols are inserted during periodic termination, only 1-dimensional (1-D) channel estimation needs to be employed.

Interpolation is carried out in the frequency domain by exploiting the correlation of the channel transfer function (CTF) between the different SCs. To address the issue of different transmit antennas transmitting different symbols at the same time, pilot symbols are inserted into the same SCs across all the antennas. This is similar to the joint pilot grid (JPG) stated in [50]. The performances of the proposed algorithm with PACE under different scenarios have been simulated and found to be comparable with the respective lower bounds with perfect channel state information (CSI).

In this work, an algorithm that avoids the computationally intensive resampling step and its associated problems has been proposed and successfully demonstrated.

The performance tradeoff is slight and the overheads can be utilized as pilot symbols to aid in the channel estimation process.

1.3 Organization of Thesis

The remainder of this thesis is structured as follows: Chapter 2 introduces the MIMO-OFDM system including FFT implementation. The wireless channel model is also covered and the system equations are given. The forward error correction codes used in the system, namely convolutional codes and LDPC codes are also briefly mentioned. Finally, iterative receivers and channel estimation based on pilot symbols are also documented.

Chapter 3 provides the theoretical background of the SMC methods and the steps involved. The proposed algorithm, the system model and the simulation results are presented in Chapter 4. In Chapter 4, it is assumed that the receiver has perfect CSI and hence no channel estimation is performed. In Chapter 5, changes are introduced into the system model to incorporate the task of channel estimation. Likewise, the simulation results for different cases are presented.

Lastly, the results of this piece of work are summarized, followed by the list of references consulted.

CHAPTER 2

MIMO-OFDM COMMUNICATION SYSTEMS

2.1 Characterization of the Wireless Channel Model

In wireless communications channel, transmitting a signal will generally result in the signal being received with attenuation and distorted phase. Moreover there may be no direct line of sight (LOS) component and the signal may be reflected by a number of scatterers, resulting in the receiver receiving multiple attenuated and delayed copies of the same signal. On top of these, in a mobile system either or both the transmitter and receiver may be in motion. This is depicted in Fig. 1.

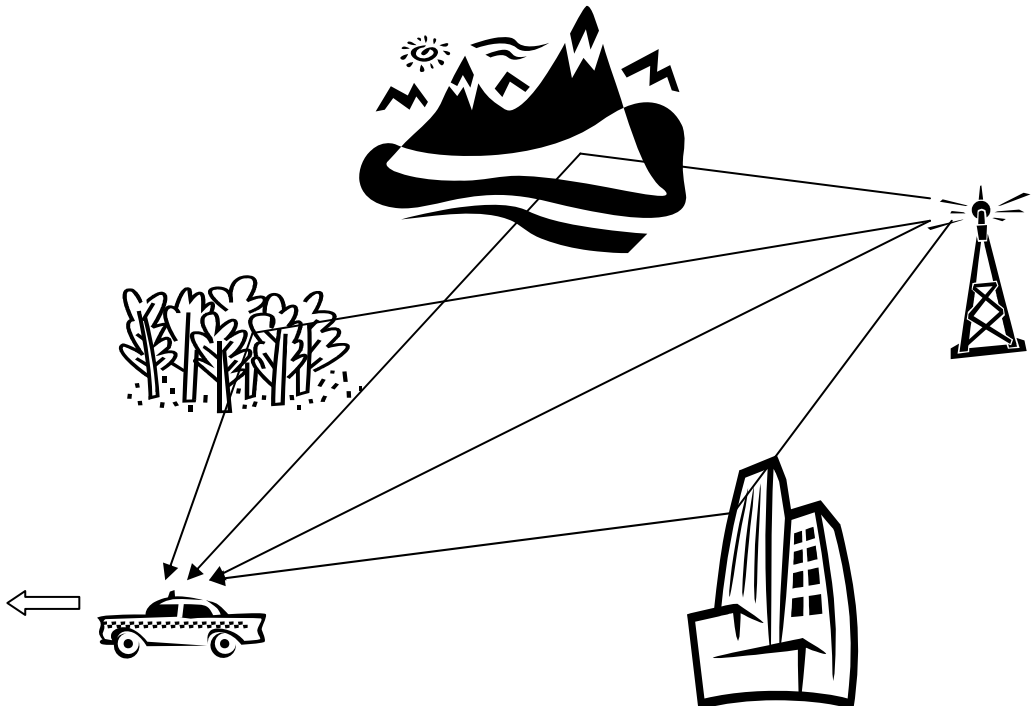


Fig. 1: An illustration of a typical wireless mobile channel

All these channel conditions impose limitations on the performance of the system. In order to understand the effects that the channel has on the transmitted signal, it is necessary to model the channel correctly.

2.1.1 Channel Models

There are generally two types of channel models, namely, large scale and small scale channel models. The large scale channel model models the signal attenuation with distance by considering the effects of path loss and shadowing. Path loss dictates the attenuation in signal strength as a function of the distance between the transmitter and the receiver while shadowing models the effects due to blockage of the line of sight component (LOS) at a fixed distance. On the other hand, the small scale channel model considers the effects due to the multipath components in small areas where the large scale effects can be ignored. Small scale effects are caused by the interference from multiple copies of the same signal arriving at the receiver with different magnitudes and phases and at different times. Therefore small scale effects are also appropriately known as multipath fading.

Several factors affect the degree of small scale fading, for instance, multipath propagation, speed of the mobile, speed of the surrounding objects and the bandwidth of the transmitted signal [14].

The equivalent low pass multipath channel model can be represented as the time variant impulse response

$$h(\tau;t)=\sum_{l=0}^{L-1}\gamma_l(t)e^{j2\pi f_C\tau_l(t)}\delta(\tau-\tau_l(t)) \qquad (2.1)$$

where $\gamma_l(t)$ is the attenuation factor and $\tau_l(t)$ is the propagation delay of the l^{th} path at time t [51]. When $h(\tau;t)$ is modeled as a zero-mean complex-valued Gaussian random process, the resultant channel is a Rayleigh fading channel.

The multipath intensity profile or the power delay profile (PDP) of the channel models the average received power of the signal from the different paths. It is determined by taking the average of $|h(\tau;t)|^2$ over a small area. The delay over which the received power is non-zero is known as the delay spread, T_m of the channel as shown in Fig. 2.

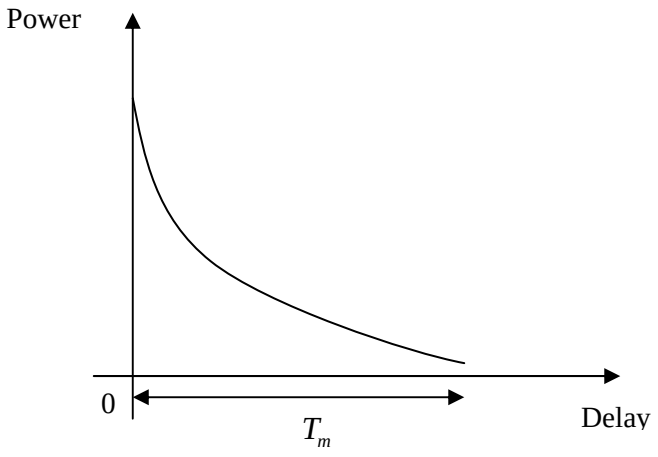


Fig. 2: Example of a multipath intensity profile

The reciprocal of the delay spread of the channel is the coherence bandwidth, B_C of the channel, which is given by,

$$B_C \approx \frac{1}{T_m}. \qquad (2.2)$$

While coherence bandwidth and delay spread describe the time dispersive nature of the channel, coherence time T_C and Doppler spread B_D describe the time varying nature of the channel due to movement of the transmitter or receiver or the surrounding objects. Doppler spread gives an indication of the expansion of the spectrum due to the relative motion between the transmitter and the receiver. It is taken to be the range of frequencies where the Doppler spectrum is non-zero. An example of a Doppler spectrum is shown in Fig. 3.

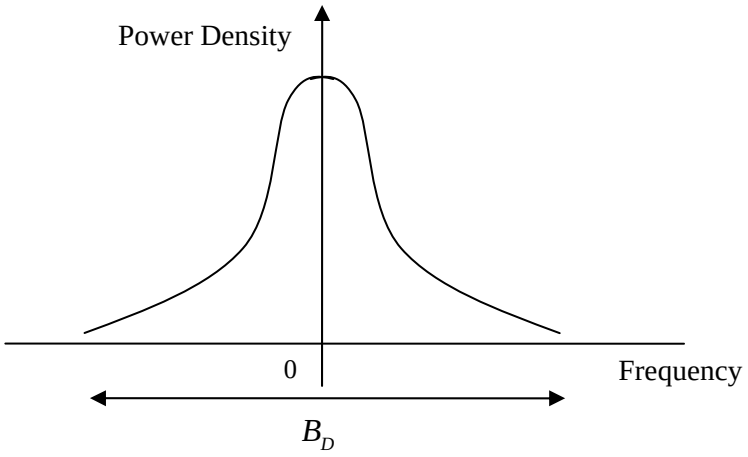


Fig. 3: Example of a Doppler spectrum

Coherence time is the reciprocal of the Doppler spread, given by,

$$T_C \approx \frac{1}{B_D}. \qquad (2.3)$$

2.1.2 Types of Small Scale Fading

Small scale fading can be classified as flat or frequency selective and slow or fast fading. Multipath delay spread gives rise to time dispersion and frequency selective fading while Doppler spread gives rise to frequency dispersion and time selective fading. The effects from both are independent.

Multipath delay spread causes either flat fading or frequency selective fading. In flat fading, the signal has a bandwidth that is smaller than the coherence bandwidth of the channel. This also means that the symbol period is larger than the delay spread of the channel. Therefore the channel appears to be of constant gain and linear phase to all the spectral components of the signal. The received signal only suffers from amplitude variations as a result of the changes in the channel gain over time due to the multipath effects but the spectral characteristics of the signal are preserved. In this case, the signal undergoes flat fading. On the other hand, if the bandwidth of the signal is larger than the coherence bandwidth, the different spectral components of the signal will be affected differently. In the time domain, this means that the symbol period is smaller than the delay spread. In this case, the received signal is distorted and dispersed as it comprises multiple copies of the transmitted signal, attenuated and delayed. This results in time dispersion and leads to intersymbol interference (ISI). In such scenarios, the signal undergoes frequency selective fading.

When there is relative motion between the transmitter and the receiver, frequency dispersion results as the Doppler spectrum widens. Doppler spreading gives rise to either fast or slow fading. In fast fading, the Doppler spread is large implying that the channel changes faster than the signal, i.e. the coherence time of the channel is smaller than the symbol period. This leads to frequency dispersion and causes distortion to the signal. On the other hand, in slow fading, the channel remains constant over a longer period of time and the coherence time of the channel is longer than the symbol period, as such the channel can be considered as constant over a few symbol durations. Translating to the frequency domain, this means that the Doppler spread is small.

2.1.3 Rayleigh Fading

Therefore there are 4 possible types of fading a signal can experience, namely, flat-slow, flat-fast, frequency selective-slow and frequency selective-fast fading. In flat fading channels, the variations of the magnitude of the received signal can be modeled by Rayleigh distribution. Rayleigh fading is an appropriate model where there are many objects in the environment that scatter the signal, constituting different paths, and there is no direct LOS component. As the number of paths increases, by the Central Limit Theorem, the inphase (real) and quadraturephase (imaginary) components of the envelope of the channel impulse response (CIR) will be Gaussian. Therefore, Rayleigh fading is modeled by representing the real and imaginary parts of the CIR by independent and identically distributed (iid) zero-mean Gaussian process so that the envelop of the response is the sum of these

two processes. The Rayleigh distribution has a probability density function (pdf) given by [51]

$$p(r)=\begin{cases}\dfrac{r}{\sigma^2}\exp\left(-\dfrac{r^2}{2\sigma^2}\right), & r\geq 0.\\ 0, & r<0.\end{cases} \tag{2.4}$$

where σ^2 is the average power of the received signal before envelope detection.

The Doppler spectrum is given by

$$S(f)=\begin{cases}\dfrac{1}{\pi f_{\max}}\dfrac{1}{\sqrt{1-(f/f_{\max})^2}}, & |f|\leq f_{\max}.\\ 0, & |f|>f_{\max}.\end{cases} \tag{2.5}$$

where $f_{\max}=\dfrac{f_C v}{c}$. The Jakes model [52] is a deterministic way to model the time-correlated Rayleigh fading waveforms. It assumes M rays of equal strength, arriving at the moving receiver, each at angle θ_m. Each ray experiences a Doppler shift of $\omega_m=\omega_{\max}\cos\theta_m$ where $\omega_{\max}=2\pi f_{\max}$. An illustration of the Doppler power spectrum is shown in Fig. 4.

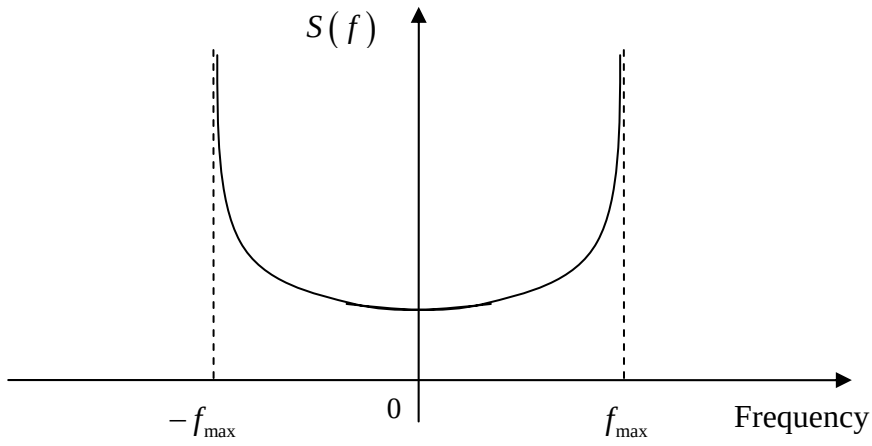


Fig. 4: An illustration of Doppler spectrum for a mobile radio channel

2.2 Background to MIMO-OFDM

OFDM started in the mid ’60s when Chang published a method of transmitting signals simultaneously through a bandlimited channel without intersymbol (ISI) and interchannel interference (ICI) [1] and subsequently the performance was analysed by Saltzberg in [2]. A great milestone was achieved in 1971 when Weinstein and Ebert employed DFT to perform baseband modulation and demodulation [3]. On top of this, to eliminate ISI and ICI, Weinstein and Ebert pre-pended a guard interval between the symbols and used raised cosine filtering in the time domain. However, orthogonality was not achieved until Peled and Ruiz [54] employed a cyclic prefix (CP), a cyclic extension of the symbol, as the guard interval. This is equivalent to the channel performing cyclic convolution and ensures orthogonality when the CP is longer than the delay spread of the channel.

Receiver design for OFDM is rather simple as the available channel bandwidth is divided into N_C number of SCs. As the bandwidth of each SC is smaller than the coherence bandwidth of the channel, this leads to the conversion of the original frequency selective fading channel into N_C number of flat fading channels thus allowing equalization to be performed easily. Moreover due to its immunity to multipath fading and impulse noise and its high spectral efficiency, OFDM has been widely used in digital audio broadcasting, digital video broadcasting and wireless LANs.

Multiple transmit and receive antennas can be used to enhance capacity or to increase diversity gain [8]. If spatial multiplexing is used, the transmission rate is increased as different signals are sent simultaneously from each of the transmit antennas [11]. In fact, it has been shown that in the MIMO system where there are N_T transmit antennas and N_R receive antennas, the capacity of the system under Rayleigh fading can be improved by a factor of $\min\left(N_T, N_R\right)$ given that all the paths between the N_T transmit antennas and N_R receive antennas are statistically independent [9-10].

With multiple receive antennas, several copies of the same signal will be obtained. This helps to mitigate the effects of fading as each copy of the signal has propagated through a different path. Therefore this provides diversity gain and hence better performance.

2.2.1 OFDM System Model

The essence of OFDM is to divide the available spectrum of W_T Hz into a number of SCs as shown in Fig. 5.

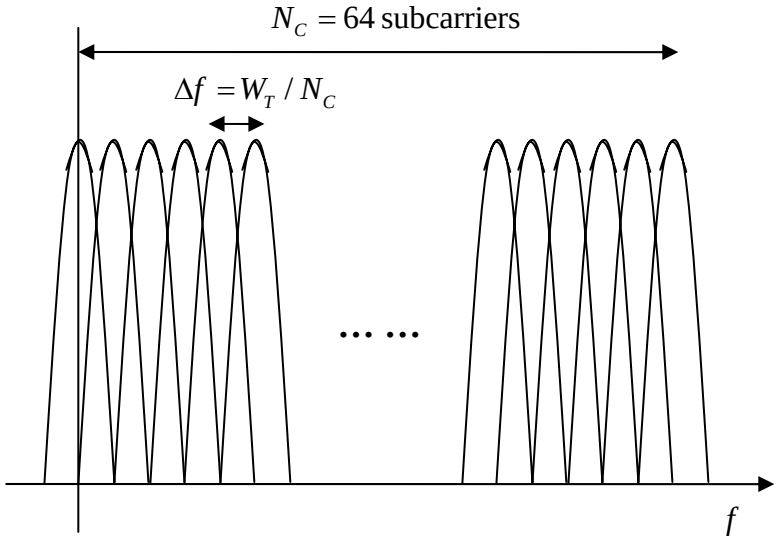


Fig. 5: An illustration of the individual SCs for an OFDM system with 64 tones

By ensuring that all SCs are narrowband, they will experience almost flat fading, which allows equalization to be performed easily.

An OFDM symbol can be expressed as

$$\vartheta(t) = \sum_{k=0}^{N_C-1} \theta_k e^{j2\pi f_k t} = \sum_{k=0}^{N_C-1} \theta_k \psi_k(t), \qquad 0 \le t \le T_{sym}. \tag{2.6}$$

where $\{\theta_k\}$ represents the sequence of complex data symbols to be transmitted, N_C is the number of FFT points, f_k is the frequency of the k^{th} SC and $\{\psi_k(t)\}$ is a set of orthogonal functions. In the absence of noise, demodulation of OFDM signal can be carried out as follows

$$\frac{1}{T_{sym}}\int_0^{T_{sym}} \vartheta(t) e^{-j2\pi f_k t} dt = \frac{1}{T_{sym}}\int_0^{T_{sym}} \left(\sum_{i=0}^{N_C-1} \theta_i \psi_i(t) \right) \psi_k^*(t)\, dt. \qquad (2.7)$$
$$= \theta_k$$

2.2.1.1 Implementation using FFT and IFFT

It can be observed that with a sampling duration of $T_S = \frac{T_{sym}}{N_C}$ and $f_k T_{sym} = k$, modulation and demodulation can be performed using a N_C point IFFT and FFT operations respectively. In the case where N_C is not a power of 2, IDFT and DFT will be used instead. Fig. 6 shows a block diagram of the OFDM transceiver.

As shown in Fig. 6, the IFFT and FFT operations simplify the transceiver structure considerably as they replace the banks of oscillators that would otherwise be needed. At the input of the IFFT is a set of N_C complex data symbols taken from an appropriate signal constellation, such as phase shift keying (PSK) or quadrature amplitude modulation (QAM).

Performing IFFT on the symbols yields

$$\vartheta_n = \sum_{k=0}^{N_C-1} \theta_k e^{j\frac{2\pi nk}{N_C}} \qquad (2.8)$$

where the sequence $\{\vartheta_n\}$ constitutes the time domain sequence. At the receiver, FFT is performed to obtain the sequence $\{\alpha_k\}$. In the absence of noise, $\{\theta_k\} = \{\alpha_k\}$.

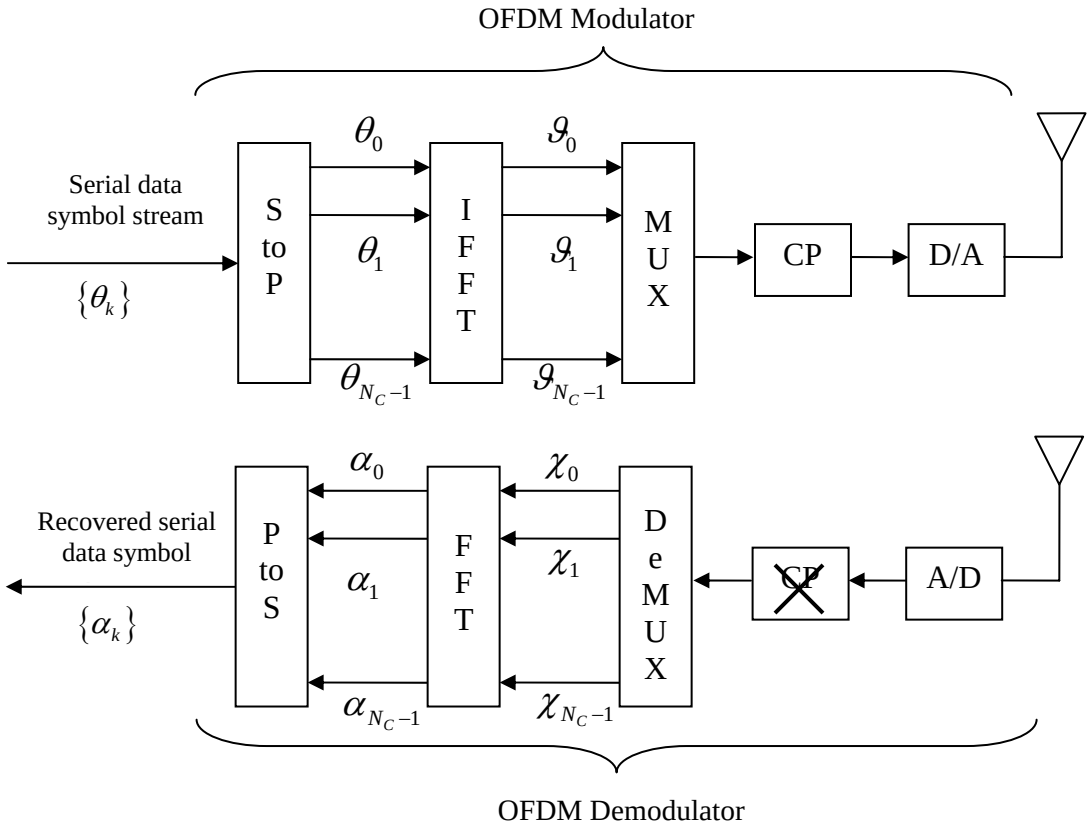


Fig. 6: Baseband model of an OFDM system

2.2.1.2 Cyclic Prefix

The time domain OFDM signal is cyclically extended to mitigate the effects of time dispersion. The last N_G samples of the OFDM symbol, of duration T_G is copied and inserted to the front of the OFDM symbol as shown in Fig. 7.

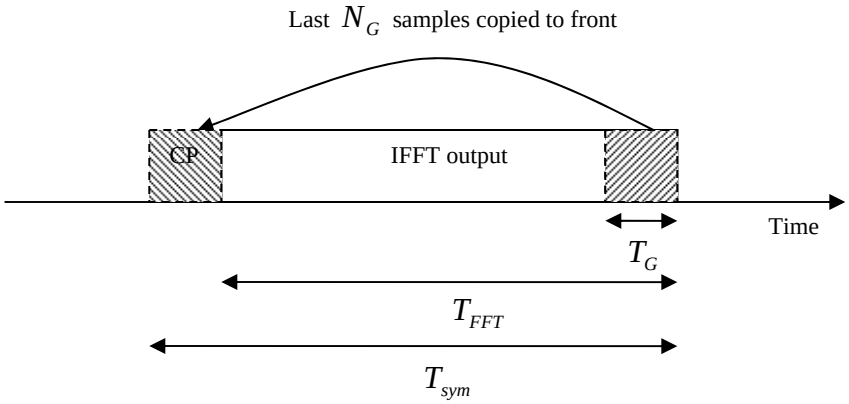


Fig. 7: Cyclic extension of an OFDM symbol

Thus each OFDM symbol consists of $N_C + N_G$ samples, which corresponds to a duration of $T_{sym} = T_{FFT} + T_G$. The length of the CP should be longer than the delay spread of the channel in order to avoid ISI. Due to the insertion of the CP, the linear convolution of the transmitted signal with the discrete time channel becomes a cyclic convolution. From the properties of cyclic convolution, it can be easily seen that the effect of the multipath channel becomes a point-wise multiplication of the transmitted data by the transfer function of the channel. Therefore, with CP, both the effects of ISI and ICI are removed [52]. Though overheads are incurred with the insertion of CP, the CP can also be used in timing and frequency synchronization.

At the receiver, the CP is removed and each block of N_C received samples is converted back to the frequency domain via the FFT operation.

2.2.1.3 Transmission Model

In the absence of ISI and ICI and assuming the channel is slowly fading such that the channel remains constant over one OFDM symbol, it is possible to model the OFDM system as a set of N_C parallel, independent Gaussian channels [39-41] shown in Fig. 8.

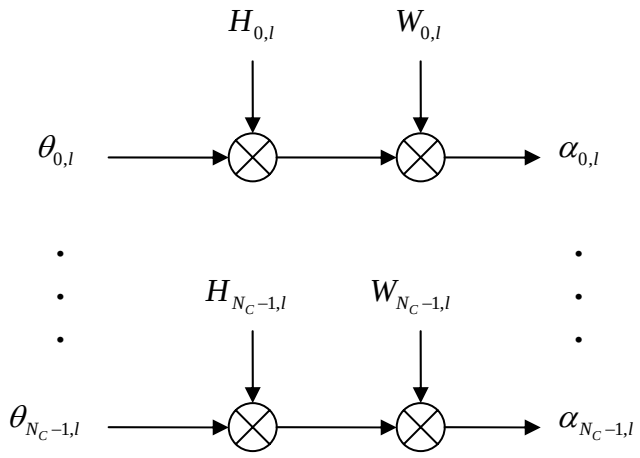


Fig.8 : OFDM system model in the absence of ISI and ICI

where l denotes the l^{th} OFDM symbol. Considering over one OFDM symbol, the system can be re-written as

$$\alpha_k = H_k\theta_k + W_k, \qquad k = 0,...,N_C - 1. \tag{2.9}$$

Alternatively, in matrix notation,

$$\bar{\boldsymbol{\alpha}} = \mathbf{H}\bar{\boldsymbol{\theta}} + \bar{\mathbf{W}} \tag{2.10}$$

where $\bar{\boldsymbol{\alpha}} = \left[\alpha_0\ \alpha_1 \cdots \alpha_{N_C-1}\right]^{\mathrm{T}}$ is the received vector of length N_C and α_k represents the received symbol at the k^{th} SC after FFT has been performed. $\mathbf{H} = diag\left(\bar{\mathbf{H}}\right)$ is a $\left(N_C \times N_C\right)$ diagonal matrix where $\bar{\mathbf{H}}$ denotes the channel

transfer function (CTF) given by $\bar{\mathbf{H}} = \left[H_0 \; H_1 \cdots H_{N_C-1} \right]^{\mathrm{T}} = \mathrm{FFT}(h)$. Each H_k denotes the complex channel attenuation at the k^{th} SC and $h = \left[h_0 \; h_1 \cdots h_{L-1} \right]^{\mathrm{T}}$. $\bar{\boldsymbol{\theta}}$ is the vector of transmitted symbols of length N_C while $\bar{\mathbf{W}} = \left[W_0 \; W_1 \cdots W_{N_C-1} \right]^{\mathrm{T}}$ is a vector of i.i.d. complex zero-mean Gaussian noise with variance σ_w^2. It is further assumed that there is no correlations between $\mathbf{H}$ and $\bar{\mathbf{W}}$. It should be noted that the H_ks are correlated with one another. For instance, if the transfer function of a particular channel is bad at a certain time, it is very likely that the H_ks adjacent to it will be bad too.

2.2.2 MIMO-OFDM System Model

The single-input single-output OFDM case can be easily extended to the MIMO case by performing the IFFT/FFT and insertion/removal of CP operations at each transmit and receive antennas respectively. Fig. 9 shows the setup of a MIMO-OFDM system. The data bitstream is encoded by the channel encoder and fed into a MIMO encoder where it is mapped to a constellation and demultiplexed into N_T symbol streams each of length N_C. These N_T symbol streams are sent to an OFDM modulator each. Insertion of the CP and modulation by IFFT are performed within the OFDM modulator as shown in Fig. 6. At the receiver, the received symbol streams from different antennas are synchronized and the CPs are removed. Demodulation is performed by FFT within the OFDM demodulator. The

symbol streams are combined within the MIMO decoder and demapped into the coded bitstream before it is decoded by the channel decoder.

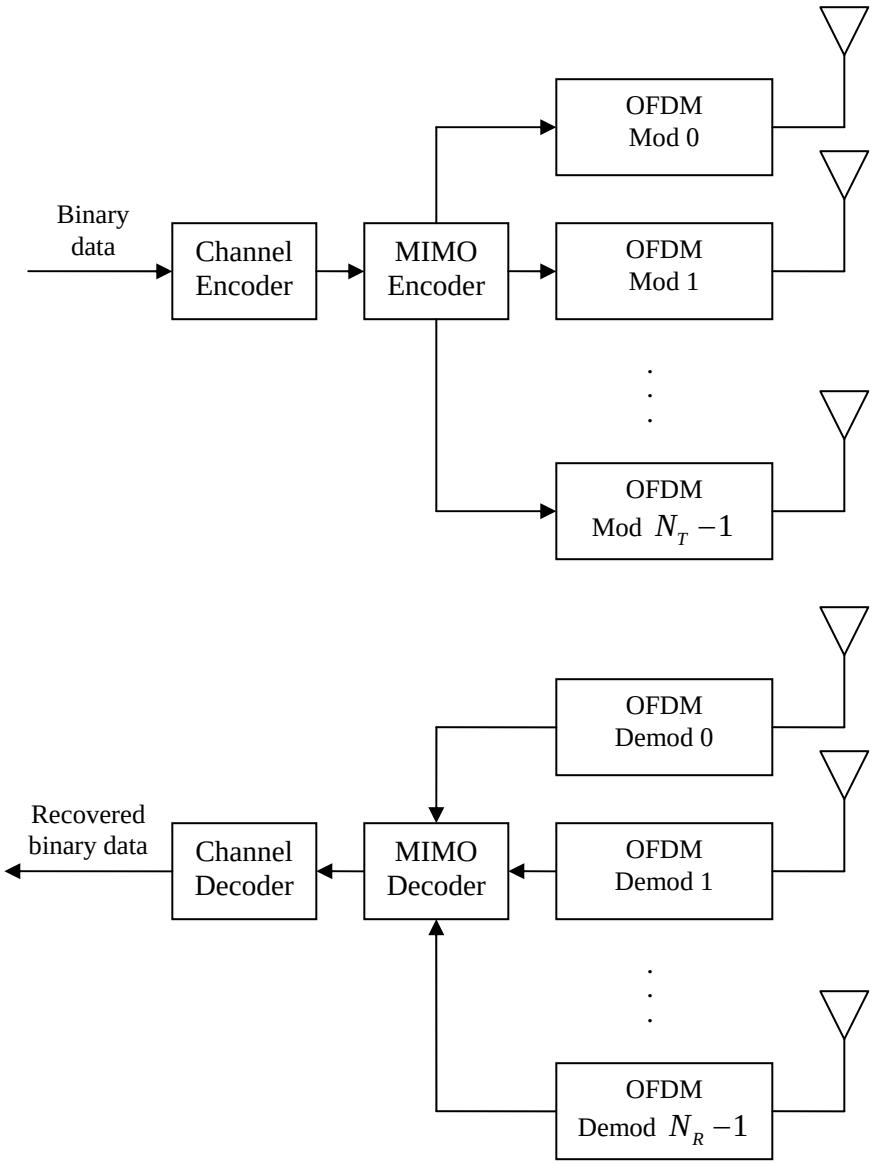


Fig. 9: MIMO-OFDM system

The MIMO-OFDM system equations can be easily formed from (2.10). The following equations are for the l^{th} MIMO-OFDM symbol but the denotation has

been dropped for clarity. Let $h^{(i,j)} = \left[h_0^{(i,j)}\ h_1^{(i,j)} \cdots h_{L-1}^{(i,j)} \right]^{\mathrm{T}}$ denote the discrete-time channel response between the i^{th} transmit antenna and the j^{th} receive antenna and $\bar{\mathbf{H}}^{(i,j)} = \left[H_0^{(i,j)}\ H_1^{(i,j)} \cdots H_{N_C-1}^{(i,j)} \right]^{\mathrm{T}} = \mathrm{FFT}\left(h^{(i,j)} \right)$. The system equation for MIMO-OFDM system is given by

$$\tilde{\boldsymbol{\alpha}} = \tilde{\mathbf{H}}\tilde{\boldsymbol{\theta}} + \tilde{\mathbf{W}} \tag{2.11}$$

where $\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^{(0,0)} & \cdots & \mathbf{H}^{(N_T-1,0)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}^{(0,N_R-1)} & \cdots & \mathbf{H}^{(N_T-1,N_R-1)} \end{bmatrix}$ and each $\mathbf{H}^{(i,j)} = diag\left(\bar{\mathbf{H}}^{(i,j)} \right)$. $\tilde{\boldsymbol{\alpha}}$ represents the received vector across all the receive antennas, of length $N_R N_C$. It is defined as $\tilde{\boldsymbol{\alpha}} = \left[\bar{\boldsymbol{\alpha}}^{(0)}\ \bar{\boldsymbol{\alpha}}^{(1)} \cdots \bar{\boldsymbol{\alpha}}^{(N_R-1)} \right]^{\mathrm{T}}$ where each $\bar{\boldsymbol{\alpha}}^{(j)}$ is the received symbols at the j^{th} receive antenna after FFT, given by $\bar{\boldsymbol{\alpha}}^{(j)} = \left[\alpha_0^{(j)}\ \alpha_1^{(j)} \cdots \alpha_{N_C-1}^{(j)} \right]^{\mathrm{T}}$. Similarly, $\tilde{\boldsymbol{\theta}}$ is the transmitted symbols across all the transmit antennas, of length $N_T N_C$ and is defined as $\tilde{\boldsymbol{\theta}} = \left[\bar{\boldsymbol{\theta}}^{(0)}\ \bar{\boldsymbol{\theta}}^{(1)} \cdots \bar{\boldsymbol{\theta}}^{(N_T-1)} \right]^{\mathrm{T}}$ where each $\bar{\boldsymbol{\theta}}^{(i)}$ denotes the transmitted symbols at the i^{th} transmit antenna given by $\bar{\boldsymbol{\theta}}^{(i)} = \left[\theta_0^{(i)}\ \theta_1^{(i)} \cdots \theta_{N_C-1}^{(i)} \right]^{\mathrm{T}}$. In a similar manner, the noise vector of length $N_R N_C$ is given as $\tilde{\mathbf{W}} = \left[\bar{\mathbf{W}}^{(0)}\ \bar{\mathbf{W}}^{(1)} \cdots \bar{\mathbf{W}}^{(N_R-1)} \right]^{\mathrm{T}}$.

2.3 Forward Error Correction in MIMO-OFDM

OFDM avoids the issue of ISI by transmitting N_C data symbols on N_C orthogonal SCs, however this leads to the following problem. In transmission through the

multipath channel, some SCs will be received with very low amplitudes when the channel is undergoing deep fades. This leads to the corresponding symbols being lost. Thus even though most of the data may be detected correctly, the bit error rate (BER) is largely dominated by the few SCs that have very low amplitudes. To circumvent this problem, forward error correction (FEC) coding is necessary. By employing FEC across all the SCs, the errors due to the weak SCs can be rectified up to a certain extent depending on the power of the FEC code used. FEC is accomplished by adding redundancy to the transmitted symbols using a pre-determined algorithm such that the receiver is able to detect and correct errors. In so doing, retransmission is avoided but at the expense of extra bandwidth incurred. In this section, convolutional codes and LDPC codes are discussed.

2.3.1 Convolutional Codes

Convolutional codes were introduced by Elias in 1955 [34] and a major milestone was achieved when Viterbi proposed a maximum likelihood decoding algorithm (MLD) [56] that was easy to implement for soft-decision decoding of convolutional codes with short constraint lengths. A few years later, maximum a posteriori probability (MAP) decoding algorithm was introduced by Bahl, Cocke, Jelinek and Raviv, the algorithm was appropriately known as the BCJR algorithm [57]. Unlike the ML algorithm where the a priori probabilities must be equal, MAP decoding caters for situations where the information bits have different a priori probabilities. Since then convolutional codes have been one of the most widely used channel codes in communication systems where low latency is

required. A brief description of the encoding and decoding process of convolutional codes is provided in the following subsections.

2.3.1.1 Encoding Convolutional Codes

A convolutional code is generated by passing the data through a linear finite state shift register [58] as shown in Fig. 10.

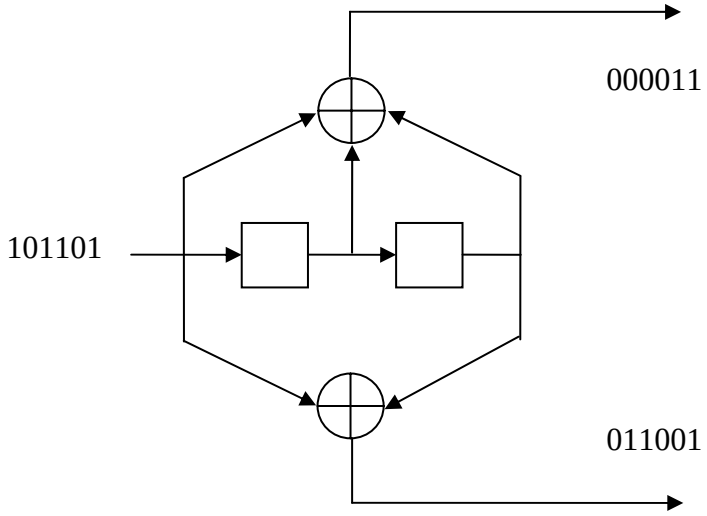


Fig. 10: Example of a binary convolutional encoder

The information bits are input into the shift registers and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. Therefore the output of the encoder does not only depend on the input bit but also on the past m bits where m denotes the memory order of the encoder. The code rate r of a convolutional code is defined as

$$r = \frac{k}{n} \tag{2.12}$$

where k is the number of parallel input bits and n is the number of parallel output bits at one clock period. The constraint length K is defined as $K = m + 1$. The convolutional encoder illustrated in Fig. 10 has a code rate, $r = \frac{1}{2}$, memory order, $m = 2$ and constraint length of $K = 3$.

Another way to describe the convolutional encoder in Fig. 10 is to define its generator polynomial. A generator polynomial describes the location of the taps in forming an output bit, a '1' denotes a connection and a '0' denotes no connection. For example, an alternative way to describe Fig. 10, would be $g_0 = [111] = 7_{\mathbf{o}}$ and $g_1 = [101] = 5_{\mathbf{o}}$. In general, convolutional codes suffer from burst errors and therefore, an interleaver needs to be used to scatter the errors.

2.3.1.2 Decoding Convolutional Codes

In decoding convolutional codes, one can choose to do hard or soft decision decoding. The former performs a 1-bit quantization on the received signal while the latter uses multi-bits quantization in order to get a better resolution. This is shown in Fig. 11. In the hard decision case, by performing a 1-bit quantization on the received values, information is lost and performance is degraded. In general, the performance of hard decision MLD is 3dB poorer than soft decision MLD.

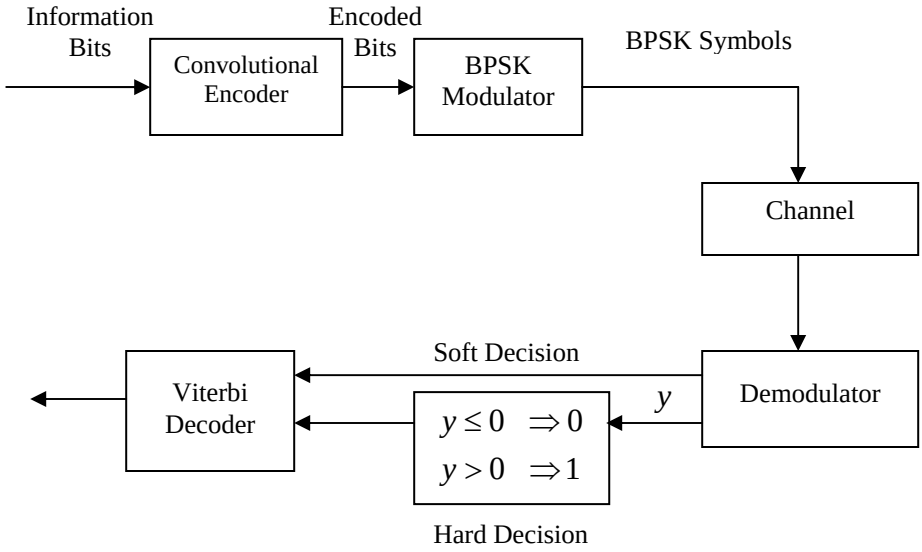


Fig. 11: Soft and hard decision decoding

Viterbi algorithm [56] achieves optimality by minimizing the error probability of the codeword while BCJR algorithm [57] minimizes the error probability of the information bit. When the information bits have equal a priori probabilities, Viterbi algorithm results in near-optimum BER performance. However when the a priori probabilities are different, BCJR algorithm performs better. On top of this, if iterative decoding is used, BCJR algorithm has to be employed instead of Viterbi algorithm, as the a priori probabilities change with every iteration.

2.3.2 LDPC Codes

Besides Turbo codes, LDPC codes is another class of Shannon-limit approaching codes and was first introduced by Gallager in 1962 [59]. However because of the decoding complexity involved and a lack of systematic ways to construct good LDPC codes, they were ignored for about 30 years. It was only until the late 1990s

when researchers began to improve on Gallager's initial construction by making use of irregular graphs and fields beyond GF(2) [60] that LDPC codes garnered more attention. Recently Davey and MacKay [61] constructed irregular rate ¼ LDPC codes over GF(8) that performs better than Turbo Codes at very low SNRs. In general, LDPC codes perform better than Turbo codes at substantial code length, for instance where the code length is more than 1000, as the minimum distance of an LDPC code increases proportionately with the length of the code [62]. The encoding and decoding processes of LDPC codes are briefly introduced in the following subsections.

2.3.2.1 Encoding LDPC Codes

An LDPC code is defined as the null space of a sparse parity check matrix H [34, 58]. Let $(n-k, n)$ be the size of H. For regular LDPC codes, H contains w_C number of 1's per column and $w_R = w_C\left(\frac{n}{n-k}\right)$ number of 1's per row where $w_C << n$ and $w_R << n-k$. On the other hand, if the number of 1's per column is not uniform, irregular LDPC codes are formed.

As LDPC code is a linear block code, encoding can be performed in a same way as encoding other block codes. An (n, k) block code C is a mapping between the k-bits message, m and the n-bits codeword, c. In a matrix form,

$$c = mG \tag{2.13}$$

where G is the generator matrix of size $k \times n$. The rows of G form a basis of the code subspace while the dual space, $C^{\perp}$ contains all the vectors that are orthogonal to C [58]. Therefore for all $c \in C$ and all $d \in C^{\perp}$, $c \cdot d = 0$. The rows of H form the basis for $C^{\perp}$, thus for all $c \in C$, $cH^{\mathrm{T}} = 0$.

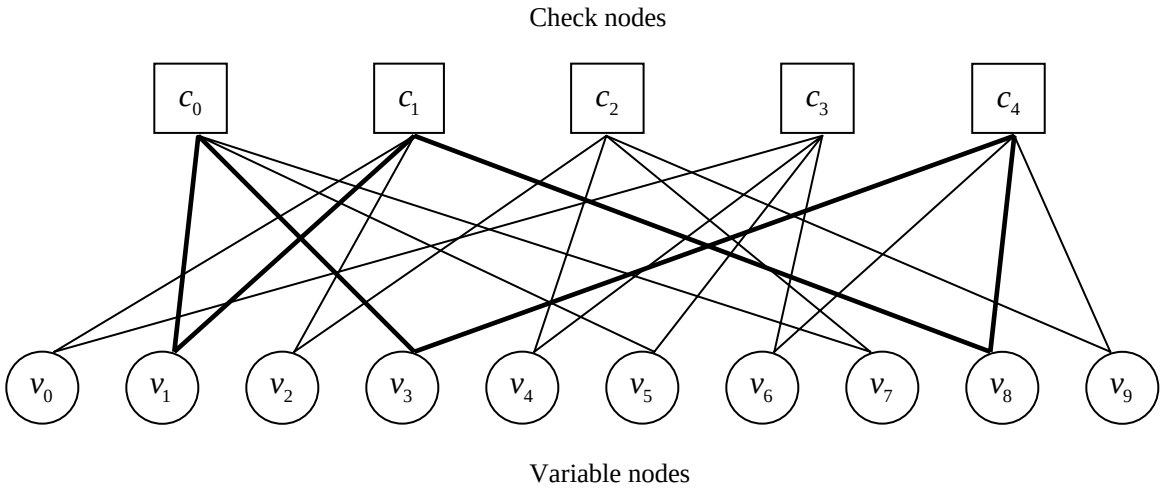
Therefore, encoding can be simply done in a straightforward manner by reducing H to a systematic form, $H_{sys} = \left[I_{n-k} \mid P \right]$ where I_{n-k} is an identity matrix of size $\left(n-k\right) \times \left(n-k\right)$ via Gaussian elimination. H_{sys} will have $n-k$ rows if H is full rank. After H_{sys} is obtained, $G_{sys} = \left[P^{\mathrm{T}} \mid I_k \right]$ can be formed and encoding can be easily performed.

2.3.2.2 Decoding LDPC Codes

An LDPC code can be decoded in a number of ways, namely, majority-logic decoding, bit-flipping (BF) decoding, weighed BF decoding, a posteriori probability (APP) decoding and iterative decoding based on belief propagation, also known as sum product algorithm (SPA). APP and SPA offer the best performance at the cost of a high complexity. However APP is computationally intractable.

Similar to the MAP, SPA is a symbol by symbol soft-in soft-out (SISO) decoding algorithm. It works by iteratively passing messages from the variable nodes, also known as code nodes, to the check nodes and vice versa. The Tanner graph

representation of the LDPC code showing the check nodes and the variable nodes is illustrated in Fig. 12.



$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Fig. 12: Tanner graph of a (10, 5) LDPC code with $w_C = 2$

The messages are actually APPs or log likelihood ratios (LLRs) computed based on the received values of the variable nodes and satisfying the constraints imposed by the check sums. In addition, the message that the variable node sends to a check node should not include the message that that check node has sent to that variable node in the previous iteration. The same applies for messages passed from the check nodes to the variable nodes [34]. At the end of each iteration, the APPs or

LLRs computed will serve as the inputs for the next iteration. The iterations continue until a stopping criterion is met. The 6 segments shown in bold in Fig. 12 define the girth (shortest cycle) of the Tanner graph. As the decoder computes the APPs or LLRs in an iterative manner, a small girth of 4 will cause the computed values to be highly correlated as the same information has been reused many times. This will result in a poor performance compared to a code with a large girth.

2.3.3 Concatenated Codes

Concatenated codes were first introduced by Forney in the 1960s. Concatenation of codes presents a simple yet powerful way to approach Shannon capacity with polynomial decoding complexity. Berrou et al [31] first showed via simulations that with iterative decoding, the performance of the parallel concatenation of 2 convolutional codes, also known as turbo codes, can approach the Shannon capacity very closely. This rekindled the interest in concatenated codes and researchers have also shown that serially concatenated convolutional codes yield performance that is comparable to turbo codes [63]. Fig. 13 shows the general structure of a serial concatenated code. The interleaver, π is optional.

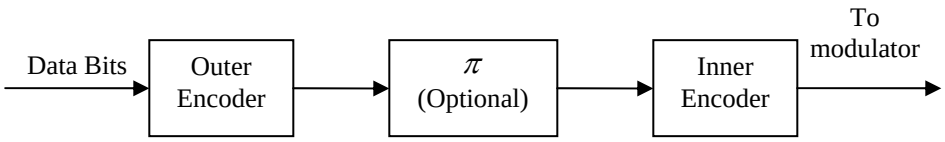


Fig. 13: Block diagram of a serial concatenated code

The inner code corrects most of the errors while the outer code helps to eliminate the remaining errors. An interleaver can be used to offer protection against burst

errors. In fact, a greater part of the interest in serial concatenated configuration stems from the fact that the outer and inner encoders can take on slightly different interpretations, for instance, the inner code can be treated to be the spreading code in CDMA systems [36] or it can also be treated as a modulator [24-25] or even a combination of a modulator and a differential encoder [22-23].

2.4 Iterative Receiver

In coded communication systems, the receiver often employs the Turbo principle to reduce loss of information. In order to do this, soft information is passed from the outer to inner decoder and vice versa until a certain stopping criterion is met. It has been shown that by iterating a number of times, the performance will approach that obtained from the optimum decoding of the composite code. The structure of an iterative receiver is shown in Fig. 14.

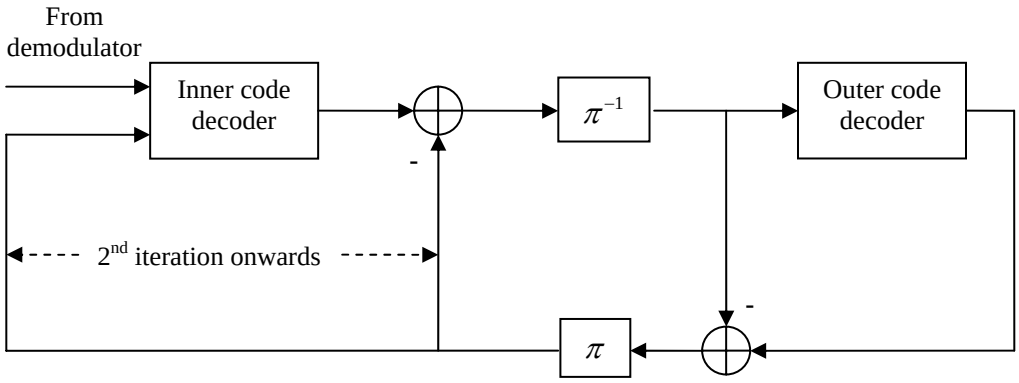


Fig. 14: Structure of iterative receiver

In an iterative receiver, both the inner code and outer code decoders employ MAP algorithm and exchange the soft decision outputs with one another. Only the

extrinsic portion of the information is passed on in an attempt to avoid reusing the same information over and over again. In cases where the alternative definition of the inner code is used [22-25], the inner code decoder shown in Fig. 14 will be a soft detector and its inputs will be the quantized A/D received signal. The soft detector uses the a priori symbol probabilities to compute the a posteriori symbol probabilities, which are passed to the SISO MAP channel decoder for it to compute the LLRs of the data bits and that of the code bits. The LLRs of the code bits are then passed to the detector as inputs. The detector and decoder exchange information with each other to improve the receiver performance.

2.5 Channel Estimation in OFDM

Accurate CSI at the receiver is essential in order to exploit the full potential of the OFDM system. Channel estimation in OFDM systems has been widely studied [23, 38-50]. In general, they can be divided into 2 main categories, blind and training (pilot) based. In a blind channel estimation method, the receiver has to determine the CSI without the help of known symbols. Even though higher bandwidth efficiency can be achieved due to the absence of training overheads, the estimation accuracy is compromised and higher complexities are involved. Therefore, they are less widely used than training based estimations.

In training based estimation techniques, known pilot symbols or training sequences are transmitted to help in the receiver's channel estimation algorithm.

The channel estimation method presented in the following sections will be extended to MIMO-OFDM systems in Chapter 5.

2.5.1 PACE

In an OFDM system, the original wideband channel that experiences frequency selective fading is converted into a number of flat fading narrowband channels, hence the task of channel estimation reduces to a trivial task of estimating the channel transfer function at each SC. As OFDM systems are generally used in frequency selective fading and time varying environments, therefore the channel transfer function changes across both the SCs and the OFDM symbols. Hence pilots are periodically inserted into both the frequency and the time domains as shown in Fig. 15.

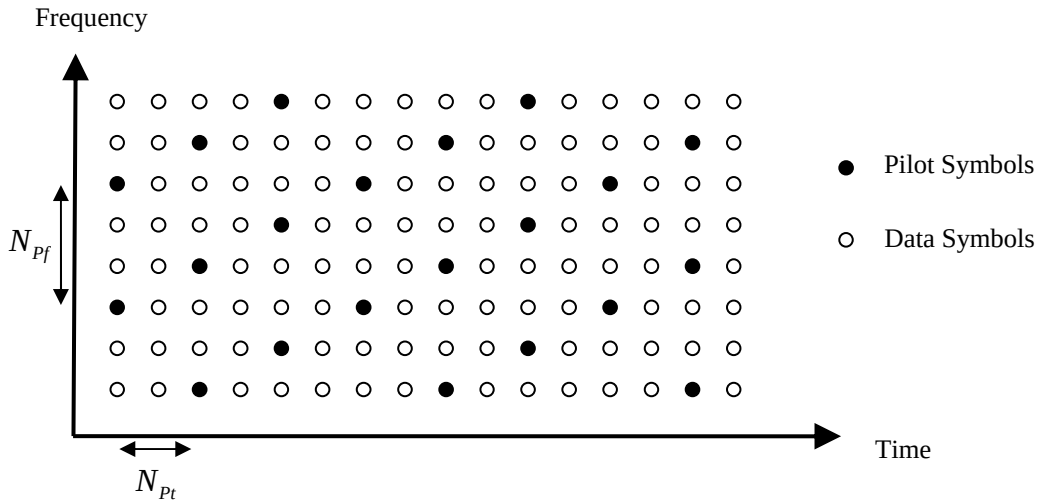


Fig. 15: Scattered pilot symbols over the 2-D frequency-time grid

There are many ways to arrange the pilot symbols in the 2-D frequency-time grid but the arrangement has to satisfy the sampling theorem [42-43] such that channel estimation and interpolation over the entire 2-D grid are possible. The sampling theorem states that

$$f_{\max} T_{sym} N_{Pt} \leq \frac{1}{2} \tag{2.14}$$

and

$$T_m \Delta f N_{Pf} \leq \frac{1}{2}. \tag{2.15}$$

For PACE method, the channel coefficients at the pilot locations are firstly estimated using LS methods. These estimates are then interpolated across the entire frequency-time grid via 2-dimensional (2-D) Wiener filtering [42]. However, 2-D Wiener filtering is very complex. To decrease complexity, two cascaded 1-D estimators in time and frequency can be used instead (2×1-D) [43].

If the sampling theorem is satisfied in both directions, the order in which filtering is performed is arbitrary. This is a result of linearity. It was found [42] that filtering in two dimensions performs better than 1-D filtering in terms of mean square error (MSE) performance. However, it was also found that for the same amount of computational effort, 2×1-D filtering offers a similar performance to 2-D filtering. Therefore in this work, only 1-D filtering is considered.

2.5.2 1-D Channel Estimators

With reference to (2.10), LS estimators minimize $\left|\bar{\alpha} - \hat{\bar{\mathbf{H}}}\bar{\theta}\right|^2$ for all possible $\hat{\bar{\mathbf{H}}}$. LS estimators are the simplest to implement but they suffer from a high MSE. On the other hand, MMSE estimators minimize $\mathrm{E}\left(\left|\hat{\bar{\mathbf{H}}} - \bar{\mathbf{H}}\right|^2\right)$ for all $\hat{\bar{\mathbf{H}}}$, offer good performance at the expense of a slightly higher level of complexity.

As the channel vector $\bar{\mathbf{H}}$ is Gaussian and uncorrelated with the noise vector $\bar{\mathbf{W}}$, the MMSE estimate of $\bar{\mathbf{H}}$ is given as [64]

$$\hat{\bar{\mathbf{H}}}_{MMSE} = \mathrm{R}_{\bar{\mathbf{H}}\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}} \mathrm{R}^{-1}_{\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}} \hat{\bar{\mathbf{H}}}_{\mathrm{LS}} \tag{2.16}$$

where $\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}$ is the LS estimate of the channel at the pilot locations, given by

$$\hat{\bar{\mathbf{H}}}_{\mathrm{LS}} = \frac{\alpha_P}{\theta_P} \tag{2.17}$$

where α_P and θ_P are the received and transmitted symbol vector at the pilot locations respectively. $\mathrm{R}_{\bar{\mathbf{H}}\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}}$ is the cross covariance matrix between $\bar{\mathbf{H}}$ and $\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}$ and $\mathrm{R}_{\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}}$ is the auto correlation of $\hat{\bar{\mathbf{H}}}_{\mathrm{LS}}$.

2.5.3 MIMO-OFDM Channel Estimation

Channel estimation for MIMO-OFDM is significantly more challenging than in the single-input single-output OFDM system as the former has a larger number of channel responses to be characterized. PACE for MIMO-OFDM has been

performed via 1-D estimation in the frequency domain [44-46] mostly for only 2 transmit antennas. In this work, besides 2 transmit antennas, the system of 4 transmit antennas is also taken into account.

Auer introduced two types of arrangements for the pilot symbols, which he named as disjoint pilot grid (DPG) and joint pilot grid (JPG) [49-50]. In the DPG, a non-overlapping, equally-spaced unique set of SCs are used to transmit the pilot symbols for each transmit antenna such that when a transmit antenna is transmitting a pilot symbol, the rest of the transmit antennas are silent [65-66]. On the other hand, for the JPG case, all antennas will be transmitting equally-spaced pilot symbols at the same time but the pilot symbols must be orthogonal. These two different types of grids are shown in Fig. 16, N_P denotes the number of pilot symbols transmitted from each antenna within a MIMO-OFDM symbol.

In this work, for the case of $N_T \times N_R$ antenna arrangement, the total number of pilots used per MIMO-OFDM symbol is $N_T N_P$. More details will be given in Chapter 5.

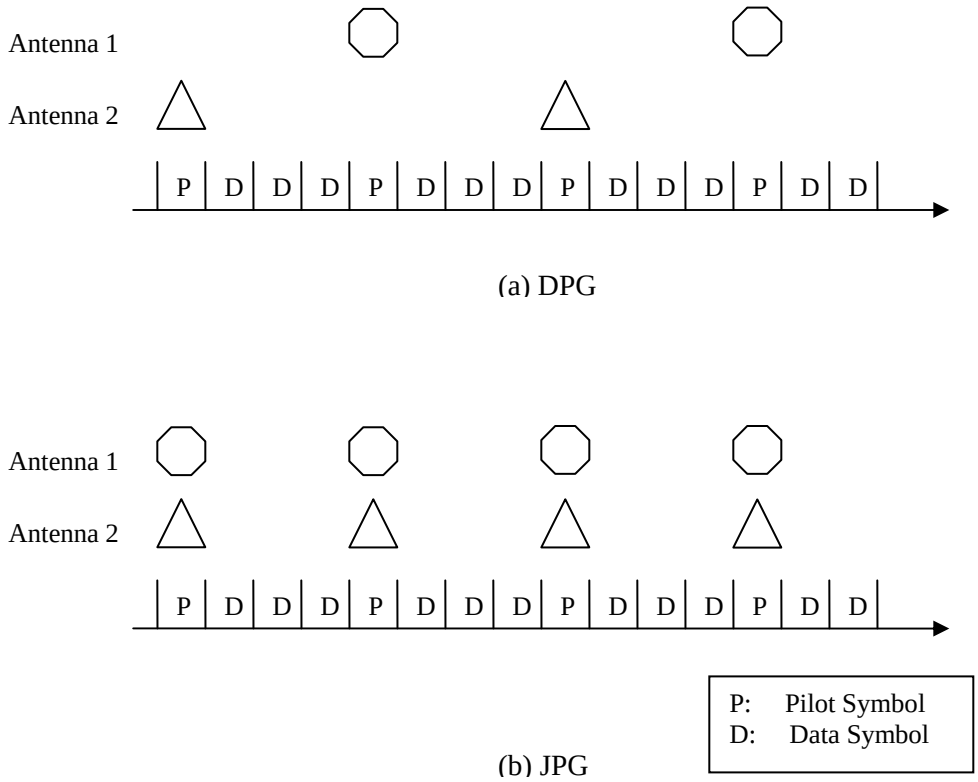


Fig. 16: Arrangements of pilot symbols for $N_T = 2$ with $N_P = 4$

CHAPTER 3

SEQUENTIAL MONTE CARLO METHODS

3.1 Background

From a Bayesian viewpoint, estimation of the hidden states of dynamic systems can be performed by reconstructing the posterior density function of those states by taking into account all the information that is available [15].

In the case of a linear system with Gaussian noise, Kalman filter [67] can be easily applied. This is because at each iteration, the posterior density function to be tracked is Gaussian. Therefore the Kalman filter is able to predict and update its mean and covariance. The extended Kalman filter [68] can be used if the system is nonlinear by approximating the target posterior density function to be Gaussian. However if the system is not Gaussian, the estimation will be erroneous. In the case of nonlinear and non-Gaussian systems, there are no closed forms for the target posterior density functions and hence, no analytical methodologies are able to deal with them. In such cases, SMC or Particle filtering (PF) methods [15-21] emerge as a class of powerful solutions.

The idea behind SMC methods is to impute multiple samples of the hidden state based on all the available observations and associate a weight to each of these imputations according to how well they can predict the next observation. At

regions where the density is higher, the weights of the particles will be larger and at regions of low density, the weights will be smaller, as shown in Fig. 17.

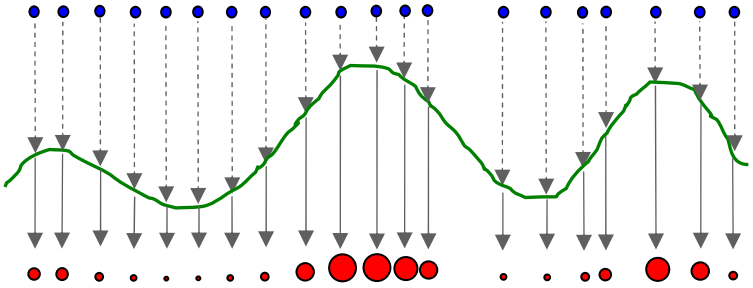


Fig. 17: Discrete representation of density using 20 weighted particles

SIS is generally involved as it is usually difficult to draw particles from the target distribution. In so doing, SMC methods allow one to estimate iteratively the APPs of the hidden states via discrete modeling of the evolution of the state variables.

The basic idea of SIS first emerged in the 1950s [69] but due to a lack of reasonable computational power at that time, it was ignored for a long period. In the recent years, with the advent of fast computing powers, SMC methods are garnering popularity. Moreover, with the resampling step that Gordon [21] introduced in 1993, it has helped to avoid the degeneracy issue that SMC methods generally face and lifted the use of SMC methods to another level. Since then SMC methods have been applied to various fields and have demonstrated to hold great promise in solving different types of non-linear estimation problems. For instance, in the area of communications, SMC methods have been successfully applied to perform blind equalization [18], detection and decoding in fading environments [22-28] and multiuser detection in CDMA systems [29].

In general, SMC methods involve state space representation, Bayesian filtering and resampling. The following sections will elaborate these steps in greater details.

3.2 State Space Representation

In general, the evolution of states in a dynamic system can be described by the state space and observation equations as follows

$$\begin{aligned} \theta_k &= \xi\left(\theta_{k-1}, v_k\right) \\ \alpha_k &= \zeta\left(\theta_k, w_k\right) \end{aligned} \tag{3.1}$$

where k is the step index, θ_k is the k^{th} element of the state vector, α_k is the k^{th} element of the observations vector while v_k and w_k are the corresponding elements of the process noise and observation noise vectors respectively and both are independent of each other. $\xi(\square)$ is the state transition function and $\zeta(\square)$ is the observation function, both of these functions can be nonlinear. It is assumed that the analytical forms of $\xi(\square)$ and $\zeta(\square)$, the distributions of v_k and w_k and the pdf of the initial state, $p(\theta_0)$ are known. The aim is to estimate θ_k recursively from the observations, α_k made. To do this, Bayesian filtering is employed.

3.3 Bayesian Filtering

The objective of Bayesian filtering is to construct the posterior density function, $p(\theta_{0:k} \mid \alpha_{1:k})$ [1] from all the observations available up to step k. As $p(\theta_{0:k} \mid \alpha_{1:k})$ contains all the information that the observations have about the hidden state, the mean, variance and other functions of the hidden state can be easily computed.

The posterior density function, $p(\theta_{0:k} \mid \alpha_{1:k})$ can be obtained in two steps, namely prediction and update. Prediction involves obtaining the predictive density $p(\theta_k \mid \alpha_{1:k-1})$ from $p(\theta_{k-1} \mid \alpha_{1:k-1})$ assuming the latter is available and that the process is Markov [30],

$$p(\theta_k \mid \alpha_{1:k-1}) = \int p(\theta_k \mid \theta_{k-1})\, p(\theta_{k-1} \mid \alpha_{1:k-1})\, d\theta_{k-1}. \tag{3.2}$$

Update involves computing the filtering density from the predictive density by using Bayes' Theorem

$$p(\theta_k \mid \alpha_{1:k}) = \frac{p(\alpha_k \mid \theta_k)\, p(\theta_k \mid \alpha_{1:k-1})}{p(\alpha_k \mid \alpha_{1:k-1})} \tag{3.3}$$

where $p(\alpha_k \mid \alpha_{1:k-1}) = \int p(\alpha_k \mid \theta_k)\, p(\theta_k \mid \alpha_{1:k-1}) d\theta_k$.

[1] There are no observations at $k = 0$.

3.4 Importance Sampling

Conventionally, if Θ is a random variable, taking on values $\theta_{1:k}$ and χ is another random variable taking on values $\alpha_{1:k}$, the expected value of a function of Θ with the aid of some observations $\alpha_{1:k}$, $\Gamma(\Theta \mid \chi)$ is given by [52]

$$\mathrm{E}\left(\Gamma(\Theta) \mid \chi\right) = \int \Gamma\left(\theta_{0:k}\right) p\left(\theta_{0:k} \mid \alpha_{1:k}\right) d\theta_{0:k}. \tag{3.4}$$

However, not all integrals are tractable and close forms are not usually possible. Moreover, the computation of (3.4) is usually prohibitive due to the large dimensional integral. Therefore one can resort to using Monte Carlo method. Monte Carlo method approximates the posterior density function $p\left(\theta_{0:k} \mid \alpha_{1:k}\right)$ with a set of weighted particles,

$$p\left(\theta_{0:k} \mid \alpha_{1:k}\right) \approx \frac{1}{\Omega} \sum_{p=1}^{\Omega} \delta\left(\theta_{0:k} - \theta_{(p,0:k)}\right) \tilde{p}\left(\theta_{0:k} \mid \alpha_{1:k}\right) \tag{3.5}$$

where $p = 1, \ldots, \Omega$ is the particle index When Ω is sufficiently big, $\tilde{p}\left(\theta_{0:k} \mid \alpha_{1:k}\right) \approx p\left(\theta_{0:k} \mid \alpha_{1:k}\right)$. Using Monte Carlo, the expression for (3.4) can be computed as

$$\mathrm{E}\left(\Gamma(\Theta) \mid \chi\right) = \frac{1}{\Omega} \sum_{p=1}^{\Omega} \Gamma\left(\theta_{(p,0:k)}\right). \tag{3.6}$$

However to do this, it must be possible to draw particles from $p\left(\theta_{0:k} \mid \alpha_{1:k}\right)$. Nevertheless, it is generally impossible to do so. Therefore importance sampling (IS) is used instead [15]. In IS, one draws particles from another density, also

known as importance density, $\varphi(\theta_{0:k} \mid \alpha_{1:k})$. The importance density should be easy to sample from, resemble the original target density and have a support that encompasses the original support. The degree of resemblance between the new and target densities is represented in the form of weights given as

$$w_{(p,k)} \propto \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{\varphi\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)} \tag{3.7}$$

and (3.4) can be expressed as

$$\mathrm{E}\left(\Gamma(\Theta) \mid \chi\right) = \sum_{p=1}^{\Omega} \tilde{w}_{(p,0:k)} \Gamma\left(\theta_{(p,0:k)}\right) \tag{3.8}$$

where

$$\tilde{w}_{(p,k)} = \frac{w_{(p,k)}}{\sum_{p=1}^{\Omega} w_{(p,k)}} \qquad \forall\, k \tag{3.9}$$

As the observations are made sequentially and a prediction of the next step index is performed whenever a new observation is available, it is not necessary to store all the past history of the observations. Therefore, a sequential form of IS (SIS) is preferred. In order to perform SIS, the importance density should satisfy the following requirement [15,17]

$$\varphi(\theta_{0:k} \mid \alpha_{1:k}) = \varphi(\theta_k \mid \theta_{0:k-1}, \alpha_{1:k}) \varphi(\theta_{0:k-1} \mid \alpha_{1:k-1}). \tag{3.10}$$

It can be seen that (3.10) has a recursive structure, for instance at a step index of k,

$$\varphi(\theta_{0:k} \mid \alpha_{1:k}) = \varphi(\theta_0) \prod_{m=1}^{k} \varphi(\theta_m \mid \theta_{0:m-1}, \alpha_{1:m}). \tag{3.11}$$

This essentially means that the set of particles from step 0 to k can be obtained by concatenating the set of particles from step 0 to $k-1$ with the current set of particles at step k. In so doing, one is able to make use of the set of particles at the previous step $k-1$ and the current observations to impute the estimate of the current state. The current set of particles, $\left\{\theta_{(p,k)}\right\}_{p=1}^{\Omega}$ is drawn from the importance density, $\varphi\left(\theta_k \mid \theta_{0:k-1}, \alpha_{1:k}\right)$ and the importance weights are given as

$$
\begin{aligned}
w_{(p,k)} &= \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{\varphi\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)} \\
&= \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{\varphi\left(\theta_0\right)\prod_{m=1}^{k}\varphi\left(\theta_{(p,m)} \mid \theta_{(p,0:m-1)}, \alpha_{1:m}\right)} \\
&= \frac{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right)}{\varphi\left(\theta_0\right)\prod_{m=1}^{k-1}\varphi\left(\theta_{(p,m)} \mid \theta_{(p,0:m-1)}, \alpha_{1:m}\right)} \frac{1}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right)} \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{\varphi\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)} \\
&= w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right)\varphi\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)}.
\end{aligned}
\tag{3.12}
$$

For the state space model given in (3.1), if $\varphi\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right) = p\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)$, the update of importance weights proceeds as

$$
\begin{aligned}
w_{(p,k)} &= w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right) p\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)} \\
&= w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k}\right) p\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right) p\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)} \\
&= w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k}\right)}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right)} \\
&= w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k-1)}, \alpha_{1:k-1}, \alpha_k\right) p\left(\alpha_{1:k-1}\right)}{p\left(\theta_{(p,0:k-1)}, \alpha_{1:k-1}\right) p\left(\alpha_{1:k}\right)} \\
&\propto w_{(p,k-1)} p\left(\alpha_k \mid \theta_{(p,0:k-1)}, \alpha_{1:k-1}\right).
\end{aligned}
\tag{3.13}
$$

The last line of the derivation is due to the fact that the term $\frac{p\left(\alpha_{1:k-1}\right)}{p\left(\alpha_{1:k}\right)}$ bears no relation with the drawn particles, hence only a proportional value of the importance weights is necessary instead of their exact values. In this case, the Monte Carlo variance is minimized. More on this will be further discussed in the next section. On the whole, the SIS algorithm can be summarized as

Table 1: SIS algorithm for the k^{th} step

1. Draw Ω particles from the importance density, $\left\{\theta_{(p,k)}\right\}_{p=1}^{\Omega} \sim \varphi\left(\theta_k \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)$.
2. Assign a weight to each particle, $\left\{w_{(p,k)}\right\}_{p=1}^{\Omega} = w_{(p,k-1)} \frac{p\left(\theta_{(p,0:k)} \mid \alpha_{1:k}\right)}{p\left(\theta_{(p,0:k-1)} \mid \alpha_{1:k-1}\right) \varphi\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)}$.

3.5 Resampling

A problem faced when using the SIS algorithm is that the distribution of particles becomes more slewed as the number of iterations increases. Usually after a

number of iterations, there will only be one particle with a large weight left while the rest have negligible weights. This results in worthless Monte Carlo estimates as the variance involved is large. In other words, a lot of computational efforts will be wasted as there are a lot of particles with negligible weights that do not contribute to the estimation of the posterior density, $p(\theta_{0:k} \mid \alpha_{1:k})$. In addition, the estimate obtained would be poor. This phenomenon is known degeneracy and it cannot be avoided as it has been shown [15] that the variance of the importance weights can only increase with time hence leading to degeneracy. Kong et al [17] and Liu and Chen [18] suggested using the following measure, the effective sample size Ω_{eff}, to indicate the severity of the degeneracy problem

$$\Omega_{eff} = \frac{\Omega}{1+Var\left(\left\{w_{(p,k)}\right\}_{p=1}^{\Omega}\right)}. \tag{3.14}$$

However the variance of the importance weights cannot be easily computed, hence, (3.14) can be approximated by

$$\tilde{\Omega}_{eff} = \frac{1}{\sum_{p=1}^{\Omega}\left(w_{(p,k)}\right)^2} \tag{3.15}$$

instead. The effective sample size ranges from 1 to Ω. When the effective sample size is close to Ω, i.e. the weights are almost uniform, it implies that the sampling is done from a distribution that closely resembles $p(\theta_{0:k} \mid \alpha_{1:k})$ and there is little problems with degeneracy. However if the effective sample size is below a certain threshold, degeneracy effect may be severe and efforts must be invested to curb it.

There are two ways to reduce the onset of degeneracy, namely using an optimal importance density and applying resampling. The optimal importance density is the true posterior density, $p(\theta_{0:k} \mid \alpha_{1:k})$. It has been proven in [15] that with the optimal importance density, the variance of the importance weights is zero and the effective sample size achieves its maximum value. However, sampling from the optimal importance density is not usually possible. Therefore, resampling presents itself as another alternative.

With resampling, trajectories with negligible weights are discarded while particles with large weights are replicated. This helps to rejuvenate the process. Basically, resampling involves sampling from the discrete approximation of $\tilde{p}(\theta_l \mid \alpha_{1:l}) = \sum_{p=1}^{\Omega} w_{(p,l)} \delta(\theta_l - \theta_{(p,l)})$ such that $\tilde{p}(\theta_l \mid \alpha_{1:l}) \approx \sum_{p=1}^{\Omega} \frac{\Omega_{(p,l)}}{\Omega} \delta(\theta_l - \theta_{(p,l)})$ at the l^{th} step where $\Omega_{(p,l)}$ is the number of replicates of $\theta_{(p,l)}$. After resampling, all particles will have the same weight of $\frac{1}{\Omega}$ [19-20] and the new particles will be concentrated in regions of higher posterior probability, which leads to better estimates. A pictorial view of resampling is shown in Fig. 18.

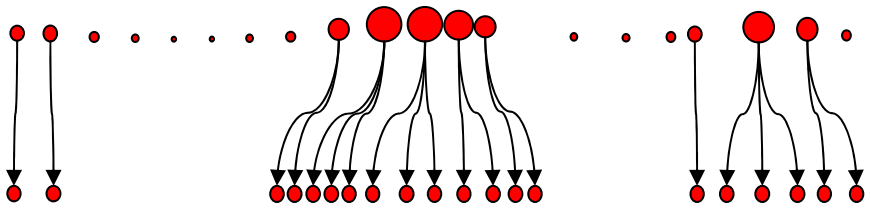


Fig. 18: A pictorial view of resampling

Resampling can be done deterministically at every l^{th} step or dynamically when the effective sample size falls below a certain threshold, for example below $\frac{\Omega}{2}$. It should be noted that resampling should not be performed if the effective sample size is close to Ω as resampling will decrease the number of distinct trajectories in such cases and lead to a poorer estimate of the true posterior density.

There are many different ways to perform resampling of which systematic resampling [30] is one of them. Systematic resampling is easy to implement and has an order of complexity of $\mathrm{O}(\Omega)$ and it keeps the Monte Carlo variation to a minimum. It involves generating Ω ordered numbers

$$m_p = \frac{(p-1)+\tilde{m}}{\Omega} \tag{3.16}$$

where $\tilde{m} \sim \mathrm{U}[0,1)$ and use them to determine to replicate or to eliminate the particles as shown in Table 2.

Table 2: Resampling algorithm for the k^{th} step

1. For $l = 2:\Omega$,
   $d_l = d_{l-1} + w_{(p,k)}$, where $d_1 = 0$
   End For.
2. Start with $l = 1$,
3. For $p = 1:\Omega$,
   $m_p = \frac{(p-1)+\tilde{m}}{\Omega}$.
   While $m_p > d_l$, increment l, end while.
   Update particles, $\theta_{(p,k)} = \theta_{(l,k)}$.
   Update weights, $w_{(p,k)} = \frac{1}{\Omega}$.
   End For.

Even though resampling helps to retard the effects of degeneracy, it comes with a price. To perform resampling, full data dependency is necessary as all particles have to be combined, hence making parallel processing impossible [15]. Moreover as particles with large weights are chosen repeatedly, there is a loss of diversity. On top of this, though resampling is a vital step, it is also computationally expensive [70]. There have been attempts to decrease the complexity of resampling [71-72] and to explore efficient means of performing resampling [70]. Though the new resampling algorithms reduce the number of operations and memory access, the complexity involved is still a burden.

3.6 Sequential Monte Carlo Methods

On the whole, SMC methods are a class of simulation methods that involve SIS and resampling. The SMC algorithm is summarized in Table 3 and a pictorial view of SMC in action is shown in Fig. 19.

Table 3: SMC algorithm for the k^{th} step

1. Draw Ω particles from the importance density, $\left\{\theta_{(p,k)}\right\}_{p=1}^{\Omega} \sim \varphi\left(\theta_k \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)$.
2. Assign a weight to each particle, $\left\{w_{(p,k)}\right\}_{p=1}^{\Omega} = w_{(p,k-1)} \frac{p\left(\alpha_k \mid \theta_{(p,k)}\right) p\left(\theta_{(p,k)} \mid \theta_{(p,k-1)}\right)}{\varphi\left(\theta_{(p,k)} \mid \theta_{(p,0:k-1)}, \alpha_{1:k}\right)}$.
3. Normalize the importance weights.
4. Perform resampling.

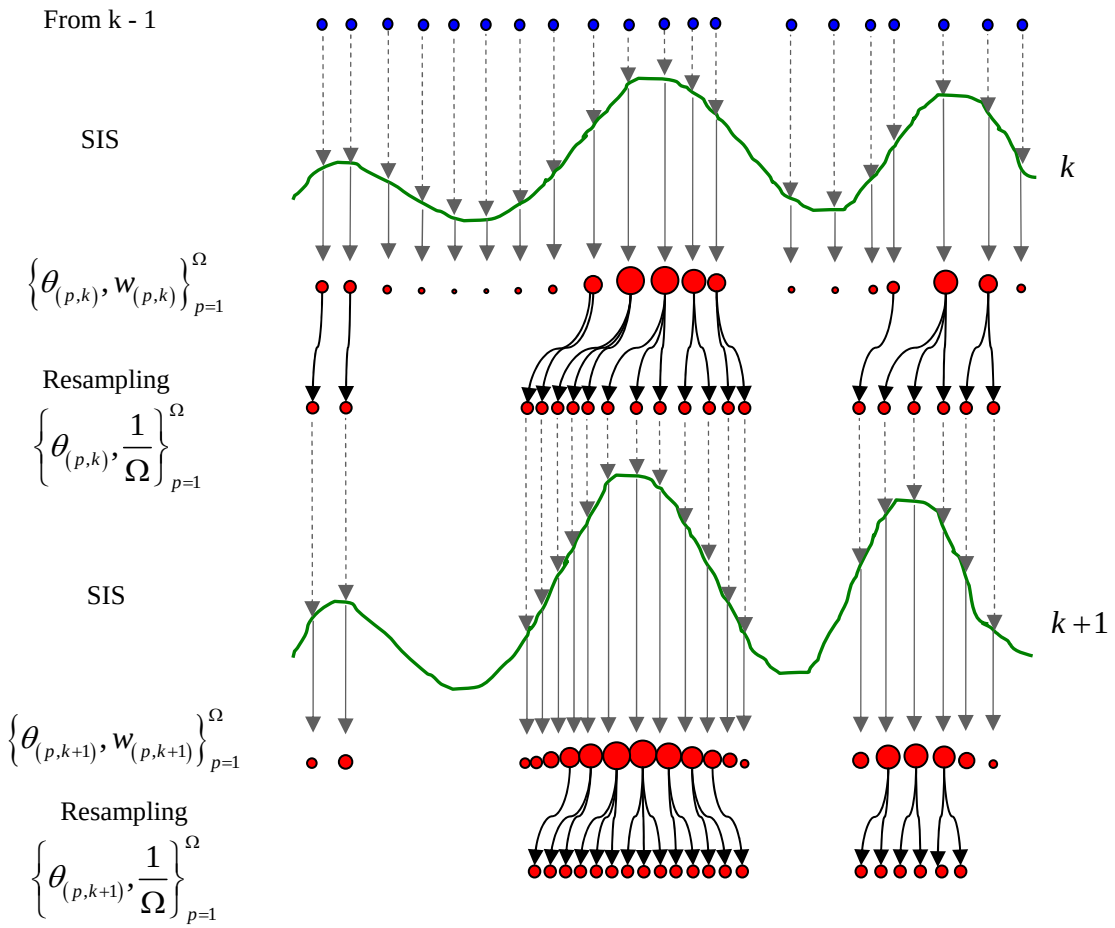


Fig. 19: A pictorial view of SMC in action

In this work, the fact that the SMC detector is able to use the a priori data symbol probabilities to produce the a posteriori data symbol probabilities such that it is well-suited to serve as the SISO demodulator in an iterative receiver is exploited. The iterative receiver iterates between the Bayesian demodulation stage and the SISO MAP channel decoding stage such that extrinsic information regarding the data symbols can be passed between these two stages to continually improve the performance of the receiver. More details on the system models will be given in the next two chapters.

CHAPTER 4

ITERATIVE RECEIVER DESIGN FOR MIMO-OFDM SYSTEMS VIA SEQUENTIAL MONTE CARLO (SMC) TECHNIQUES

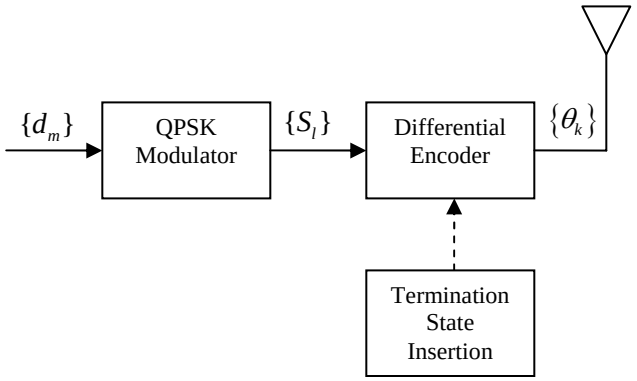
4.1 Background

In this work, SMC methods are used in the design of the first stage of the iterative receiver, which is the SISO detector. Differential encoding is employed. The novelty in this algorithm is that the computationally intensive and sequential-processing resampling step is avoided with little performance degradation, simply by periodically terminating the stream of differentially encoded symbols. Both convolutional coded and LDPC coded MIMO-OFDM systems over Rayleigh fading channels are considered. In this part of the work, the receiver is assumed to have perfect CSI and the carrier frequency and phase offset are assumed to have been compensated.

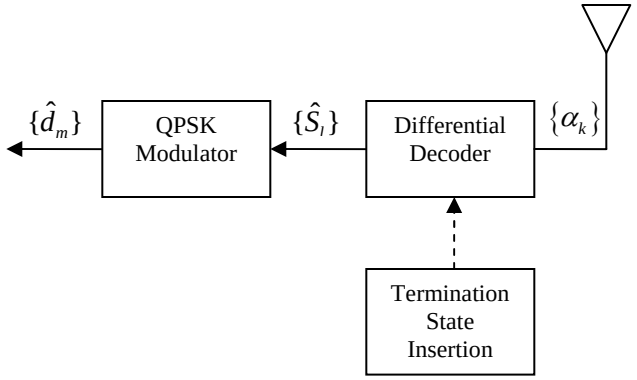
The organization of this chapter is as follows, a discussion of periodic termination of the trellis is first presented followed by the system model for the coded MIMO-OFDM system. The system model of the non-resampling SMC detector is presented together with a brief computation of the complexity involved. This is followed by the simulation results and the conclusions.

4.2 Periodic Termination

In this section, the principle that enables resampling to be avoided is presented. Consider the baseband model of differentially encoded QPSK symbols transmitted over AWGN channel as shown in Fig. 20.



(a) Transmitter



(b) Receiver

Fig. 20: Baseband model of differentially encoded QPSK symbols transmitted over AWGN channel

The data bits $\{d_m\}$ are mapped into QPSK symbols $\{S_l\}$. Termination states are inserted and differential encoding is performed before the symbols $\{\theta_k\}$ are

transmitted over the AWGN channel. The differential encoding and decoding processes are given in Table 4.

Table 4a: Differential Encoding

Previous Output	Current Input	Current Output
00	00	00
01	00	01
10	00	10
11	00	11
00	01	01
01	01	10
10	01	11
11	01	00
00	10	10
01	10	11
10	10	00
11	10	01
00	11	11
01	11	00
10	11	01
11	11	10

Table 4b: Differential Decoding

Previous Output	Current Input	Current Output
00	00	00
01	01	00
10	10	00
11	11	00
00	01	01
01	10	01
10	11	01
11	00	01
00	10	10
01	11	10
10	00	10
11	01	10
00	11	11
01	00	11
10	01	11
11	10	11

With differential encoding, the stream of differentially encoded $\{S_l\}$ will take on different phases. Fig. 21 shows the phase trellis of differentially encoded symbols $\{\theta_k\}$ before periodic termination. θ_0 is initialized to zero phase, which also corresponds to bits {00}. States $\{0, 1, 2, 3\}$ refer to $\{00, 01, 10, 11\}$ respectively. S_l on each segment refers to the symbol that caused the transition in phase.

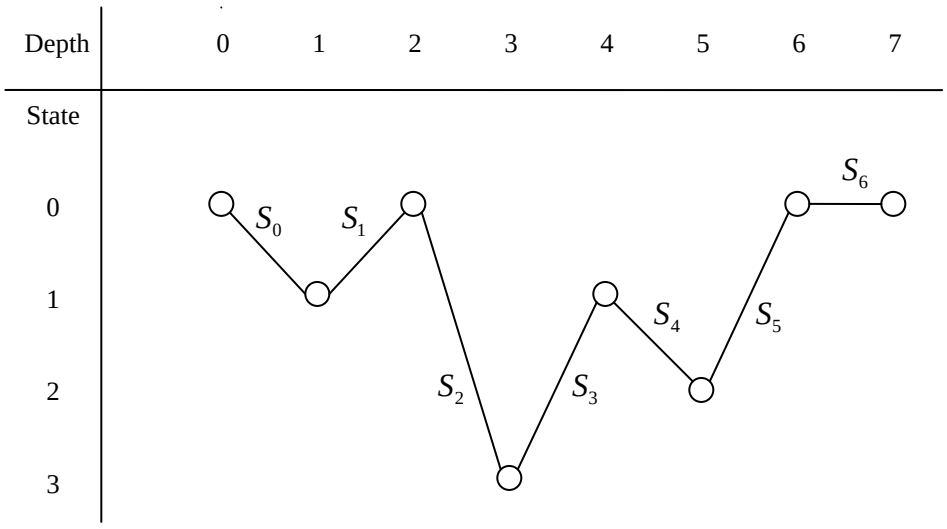


Fig. 21: Phase trellis of differentially encoded symbols before periodic termination

With periodic termination, extra symbols $\{T_j\}$ are inserted into the sequence of $\{S_l\}$ periodically as shown in Fig. 22. The extra symbols can be chosen such that the differentially encoded termination states, shown as black circles in Fig. 22, will cycle through the different states as shown in the figure or they can be chosen such that the termination states are all the same, for example, state 00. Both options are viable as these extra symbols will be discarded after demodulation. The beauty of

these two options is that with only the knowledge of the initial differentially encoded symbol, θ_0 and the scheme that is used, the receiver is able to know the rest of the differentially encoded termination states. This knowledge is necessary for demodulation.

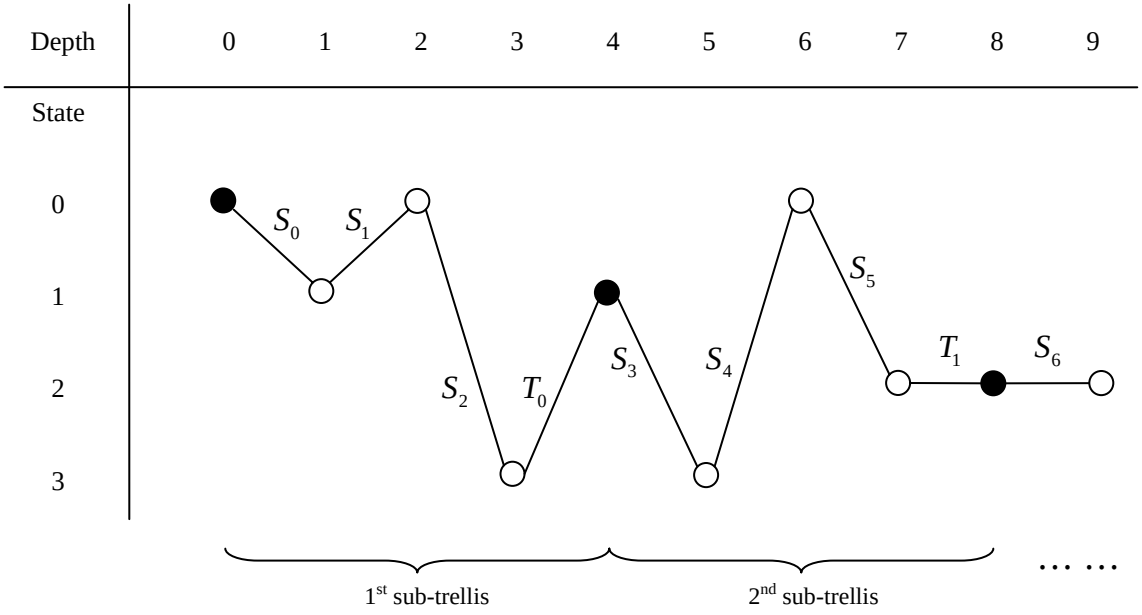


Fig. 22: Phase trellis of differentially encoded symbols, $\{\theta_k\}$ with periodic termination of period $K = 4$

There are generally two groups of possible values for K, namely when K is a factor of N_C and when it is not a factor. It is seen that when K is a factor of N_C, there will be a set of unique SCs that are solely dedicated to the transmission of the termination states or pilot symbols. On the other hand, if K is not a factor of N_C, the pilot symbols will be scattered throughout the entire grid. An illustration of this can be found in Fig. 25. The exact pattern of the distribution will be

dependent on the values of K and N_C. Apparently, the smaller the value of K, the larger the amount of overheads incurred.

4.2.1 Effects of Periodic Termination

Consider the system shown in Fig. 20 without termination states inserted. The state space equations can be written similar to (3.1)

$$\begin{aligned} \theta_k &= \xi(\theta_{k-1}, S_k) \\ \alpha_k &= \zeta(\theta_k, W_k) \end{aligned} \tag{4.1}$$

where k is the step index, θ_k is the differentially encoded symbol, S_k is the data symbol that causes the transition from θ_{k-1} to θ_k, α_k is the received symbol while W_k is the AWGN noise with variance σ_w^2. $\xi(\square)$ is the differential encoding function and $\zeta(\square)$ is the observation function. The data bits can be easily recovered once the data symbol stream $\{\hat{S}_k\}$ is recovered. To recover $\{\hat{S}_k\}$, the receiver has to trace the trellis shown in Fig. 21. To do this, the receiver can, with the knowledge of θ_0, impute Ω weighted particles, $\{\hat{\theta}_{(p,k)}, w_{(p,k)}\}_{p=1}^{\Omega}$ at each step k for each white circle in Fig. 21. Apparently, these Ω trajectories can consist of any possible path through the trellis out of a total of 4^K possible paths for the case of a block of K QPSK symbols. As the variance of the importance weights of the particles can only increase over time [30], this causes degeneracy and leads to poor estimation. Therefore resampling is necessary if K is large. However resampling is computationally intensive. Moreover, ideally the Ω trajectories should be as diverse and independent as possible to ensure the correct trajectory is

among one of the imputed trajectories but resampling duplicates multiple copies of a heavily weighted particle and causes a loss of independence.

Therefore it is proposed that known termination states are inserted into the phase trellis as shown in Fig. 22, such that the original long block of symbols can be segmented into shorter blocks, sub-trellises, hence keeping degeneracy in check. Moreover with a smaller K, there is a higher chance that the correct path is present in the Ω imputed trajectories. Large burst lengths are also avoided as each sub-trellis is only of a short length of K. As each sub-trellis starts with a known termination state, all sub-trellises are independent given knowledge of the termination states. Therefore parallel processing is possible, ensuring a shorter processing time. As observed from Fig. 22, the last symbol of each sub-trellis that leads to the termination state is a dummy symbol $\{T_j\}$ which is not used after demodulation therefore no imputation is necessary. Therefore, the total number of imputations for the system with periodic termination is the same as that without periodic termination.

In the MIMO-OFDM system with perfect CSI at the receiver, periodic termination helps to reduce the multi-dimensional pdf into a mere product of N_R 1-D pdf which can be easily computed.

Periodic termination causes overheads and a loss of bandwidth efficiency. However these overheads can be employed as pilot symbols to aid in the channel estimation process as will be shown in the next chapter.

4.3 Coded MIMO-OFDM System Model

A MIMO-OFDM system with N_T transmit and N_R receive antennas and N_C SCs as shown in Fig. 23 is considered.

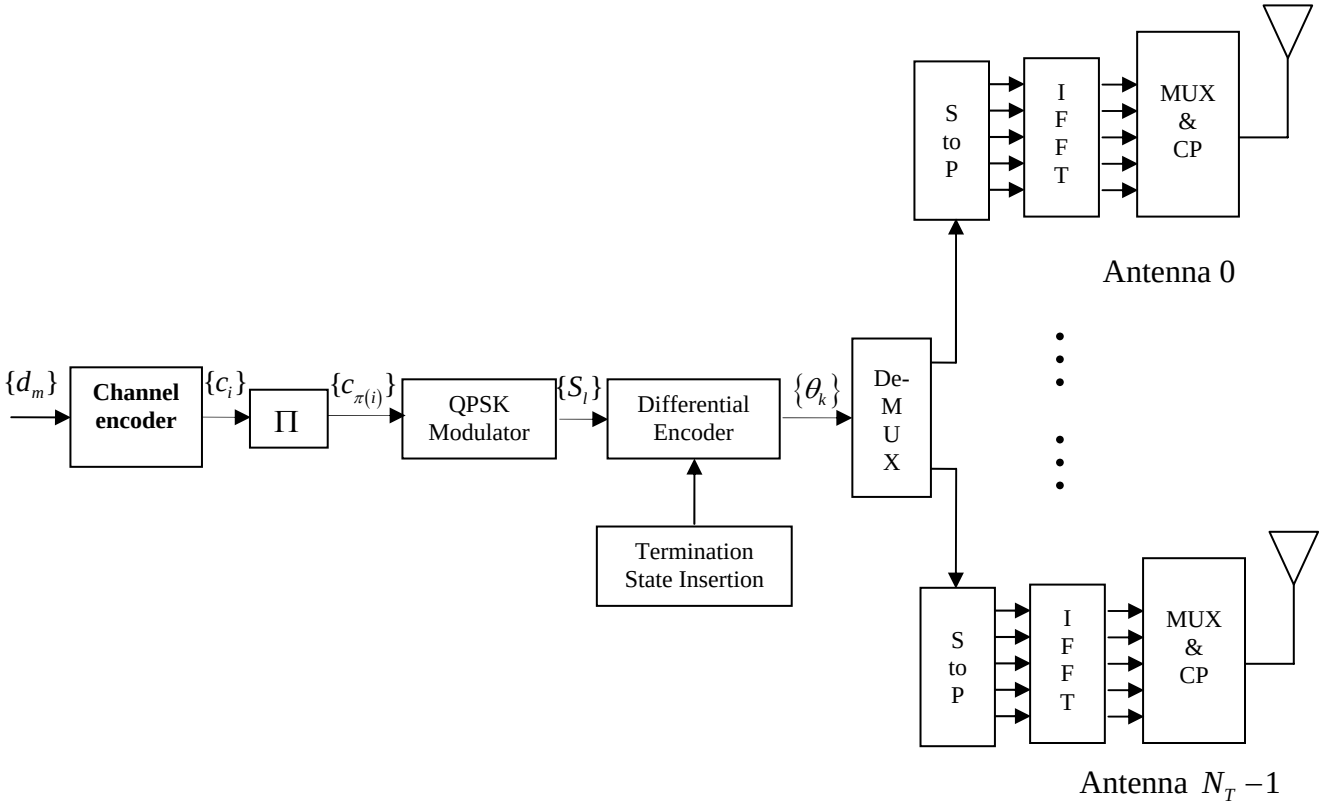


Fig. 23: Structure of proposed transmitter

The data bits $\{d_m\}$ are first encoded by the channel encoder to form code bits $\{c_i\}$ which are then interleaved and mapped into QPSK symbols $\{S_l\}$. These QPSK

symbols are differentially encoded and termination states are inserted. The differentially encoded symbols $\{\theta_k\}$ are de-multiplexed N_C symbols at a time to each transmit antenna to be processed into an OFDM signal.

The transmitter can be viewed to have a serial concatenated structure as the channel encoder serves as the outer encoder while the combination of the QPSK modulator and the differential encoder forms the inner encoder. As such, an iterative structure can be used at the receiver. It consists of two SISO stages, namely, the SMC detector and the MAP channel decoder. The proposed receiver structure is shown in Fig. 24.

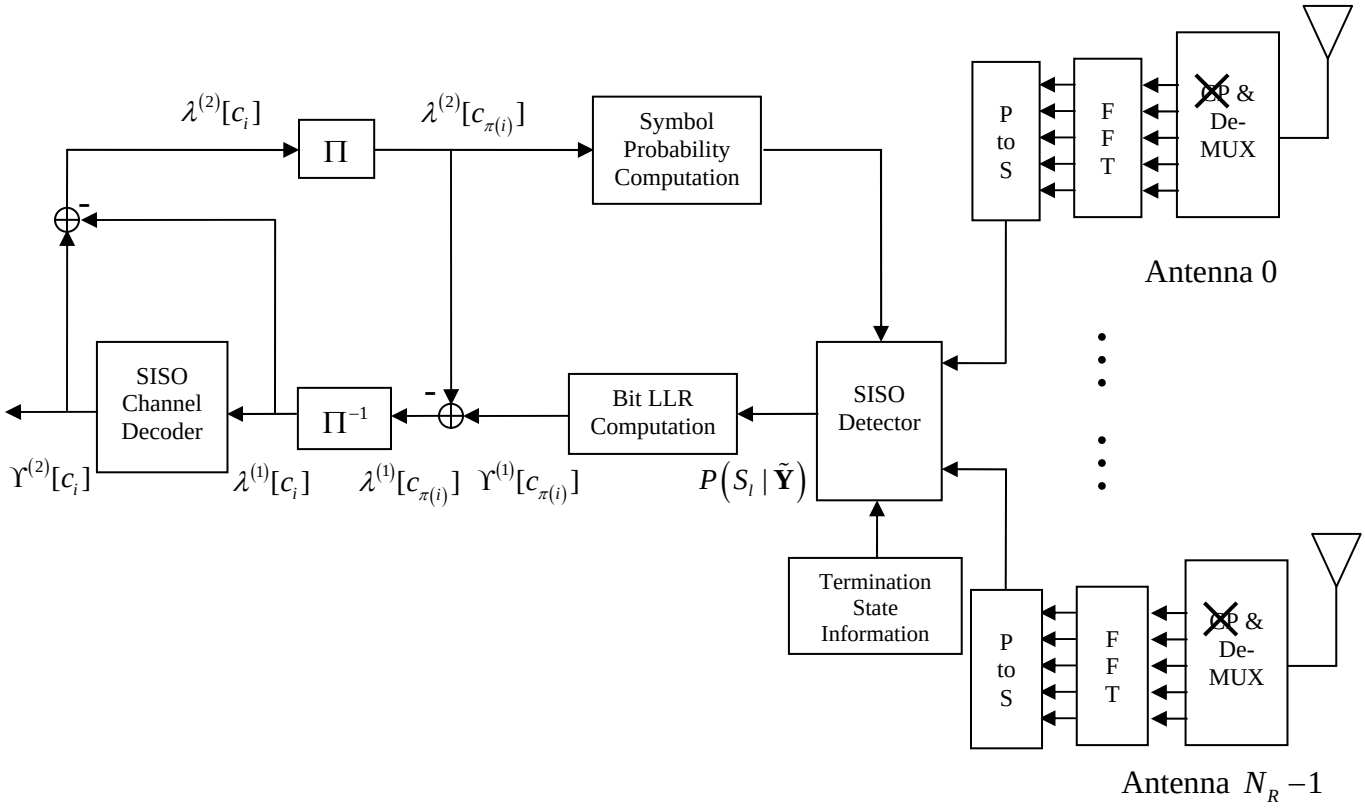


Fig. 24: Structure of proposed receiver

After the OFDM demodulators at each receive antenna, the received symbols are multiplexed into $\{\alpha_k\}$ and passed into the SMC detector. The SMC detector utilizes the a priori symbol probabilities obtained from the previous iteration to compute the a posteriori symbol probabilities. The difference between the a priori and the a posteriori information of the bits, also known as the extrinsic information, is passed to the channel decoder. The channel decoder computes both the log likelihood ratios (LLRs) of the code bits and the information bits. As both the SMC detector and the channel decoder are SISO in nature, they exchange extrinsic information iteratively to improve the performance of the system. More details of the iterative receiver would be presented in the next section.

4.4 Iterative Receiver Design for Coded MIMO-OFDM Systems with Non-Resampling SMC Detection

In this section, the derivation of the iterative receiver with non-resampling SMC detector is presented.

4.4.1 Transmission Model

Assuming that the CP operation ensures the orthogonality of the SCs and eliminates ISI between adjacent OFDM symbols, the system can be modeled as a set of parallel Gaussian channels [39-41], as described in sections 2.2.1.3. In a similar manner, the MIMO-OFDM system can be expressed as

$$\tilde{\alpha} = \tilde{\mathbf{H}}\tilde{\theta} + \tilde{\mathbf{W}} \tag{4.2}$$

where $\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^{(0,0)} & \cdots & \mathbf{H}^{(N_T-1,0)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}^{(0,N_R-1)} & \cdots & \mathbf{H}^{(N_T-1,N_R-1)} \end{bmatrix}$ and each $\mathbf{H}^{(i,j)} = diag\left(\bar{\mathbf{H}}^{(i,j)}\right)$. $\bar{\mathbf{H}}^{(i,j)}$ denotes the FFT of the discrete-time channel response between the i^{th} transmit and the j^{th} receive antenna. $\tilde{\alpha}$ represents the received vector across all receive antennas, $\tilde{\alpha} = \left[\bar{\alpha}^{(0)}\ \bar{\alpha}^{(1)} \cdots \bar{\alpha}^{(N_R-1)}\right]^{\mathrm{T}}$ and $\tilde{\theta}$ represents the transmitted symbols across all transmit antennas, $\tilde{\theta} = \left[\bar{\theta}^{(0)}\ \bar{\theta}^{(1)} \cdots \bar{\theta}^{(N_T-1)}\right]^{\mathrm{T}}$. In a similar manner, the noise vector is given as $\tilde{\mathbf{W}} = \left[\bar{\mathbf{W}}^{(0)}\ \bar{\mathbf{W}}^{(1)} \cdots \bar{\mathbf{W}}^{(N_R-1)}\right]^{\mathrm{T}}$ where each element of $\bar{\mathbf{W}}^{(j)}$ is zero-mean circularly symmetric complex Gaussian noise with variance σ_w^2. More details can be found in section 2.2.2.

4.4.2 Channel Model

For the MIMO system, the mobile wireless channel between each pair of transmit and receive antennas, is taken to be independent and uncorrelated but follow the same statistics as described in the following. A multipath fading frequency-selective channel model consisting of L impulses is assumed. Considering the channel is sample-spaced, such that all the energies are mapped to the taps, the discrete time channel model between the i^{th} transmit and the j^{th} receive antenna can be expressed as $h^{(i,j)} = \left[h_0^{(i,j)}\ h_1^{(i,j)} \cdots h_{L-1}^{(i,j)}\right]^{\mathrm{T}}$. Due to the motion of the vehicle,

the $\{h_l\}_{l=0}^{L-1}$ s are i.i.d. zero mean, circularly symmetric, complex Gaussian random variables with variance κ_l^2.

The generation of the random variables $\{h_l\}_{l=0}^{L-1}$ depends on the PDP and the channel dispersion. Two PDPs are considered, uniform (UNI) and exponential (EXP). In UNI PDP, all impulses have the same average power κ_l^2 while in EXP PDP, the average power of each path takes on values Ce^{-l} where C is a normalizing constant. For a more accurate comparison of the MIMO-OFDM systems under different PDPs, the delay spreads of these PDPs are maintained to be about the same. This delay spread is defined as

$$T_d = \sqrt{\sum_{l=0}^{L-1} \frac{\kappa_l^2}{\sum_{j=0}^{L-1} \kappa_j^2} T_l^2 - \left(\sum_{l=0}^{L-1} \frac{\kappa_l^2}{\sum_{j=0}^{L-1} \kappa_j^2} T_l \right)^2} \qquad (4.3)$$

where $\sum_{j=0}^{L-1} \kappa_j^2$ denotes the total average power of the channel's impulse response and the quantity is usually taken to be unity.

To simulate the effect of Doppler spreading caused by relative motion, an improved Jakes Model [73] based on the sum-of-sinusoids method is incorporated. It is assumed that the channel fades slowly such that the channel coefficients are constant over the duration of one MIMO-OFDM symbol.

4.4.3 System Model

At the receiver, after the removal of the CP and the FFT operations, the received symbols are passed into the SMC detector. With knowledge of the CSI and the termination states, the SMC detector proceeds as follows:

When $N_T = N_R$, the system can be modeled as

$$\breve{\alpha}_k = \breve{H}_k \breve{\theta}_k + \breve{W}_{,k} \qquad k = 0,..., N_C - 1 \tag{4.4}$$

where $\breve{\alpha}_k$ represents the received symbols across all the N_R receive antennas at the k^{th} SC, i.e. $\breve{\alpha}_k = \left[\alpha_k^{(0)}, \alpha_k^{(1)}, ..., \alpha_k^{(N_R-1)}\right]^{\mathrm{T}}$, similarly $\breve{\theta}_k = \left[\theta_k^{(0)}, \theta_k^{(1)}, ..., \theta_k^{(N_T-1)}\right]^{\mathrm{T}}$ and $\breve{W}_k = \left[W_k^{(0)}, W_k^{(1)}, ..., W_k^{(N_R-1)}\right]^{\mathrm{T}}$ where $W_k^{(j)} \sim N\left(0, \sigma_w^2\right)$. Lastly $\breve{H}_k$ is a complex matrix representing the FFT of the channel coefficients at the k^{th} SC.

For clarity, denote $\tilde{\boldsymbol{\theta}}_k = \left\{\breve{\theta}_i\right\}_{i=0}^{k}$ and $\tilde{\boldsymbol{\alpha}}_k = \left\{\breve{\alpha}_j\right\}_{j=0}^{k}$ where the symbols are taken over all transmit or receive antennas and $\tilde{\mathbf{H}}_k = \left\{\breve{H}_j\right\}_{j=0}^{k}$. The aim of the SMC detector is to perform the dynamic estimation of the a posteriori symbol probability, given the received symbols up to step k and the knowledge of the CSI

$$P\left(S_k^{(i)} = b_l \mid \tilde{\boldsymbol{\alpha}}_k, \tilde{\mathbf{H}}_k\right), \qquad b_l \in \mathrm{B}, \quad i = 0,..., N_T - 1, \quad k = 0,..., N_C - 1 \tag{4.5}$$

where $\mathrm{B} = \left\{b_0, b_1, b_2, b_3\right\}$ contains all the possible symbols from the QPSK alphabet.

In order to perform SMC, a set of weighted particles, $\left\{\hat{\theta}_{(p,k)}, w_{(p,k)}\right\}_{p=1}^{\Omega}$ properly weighted w.r.t. the distribution $p\left(\tilde{\boldsymbol{\theta}}_k \mid \tilde{\boldsymbol{\alpha}}_k, \tilde{\mathbf{H}}_k\right)$ has to be generated for each transmitted data symbol under consideration. Fig. 25 shows the symbol grid of the MIMO-OFDM transmission symbol. The number in each box refers to the sub-trellis index. The SMC detector traces the trajectories in increasing SC index, k. As shown, processing by the SMC detector can be performed independently as the sub-trellises are independent.

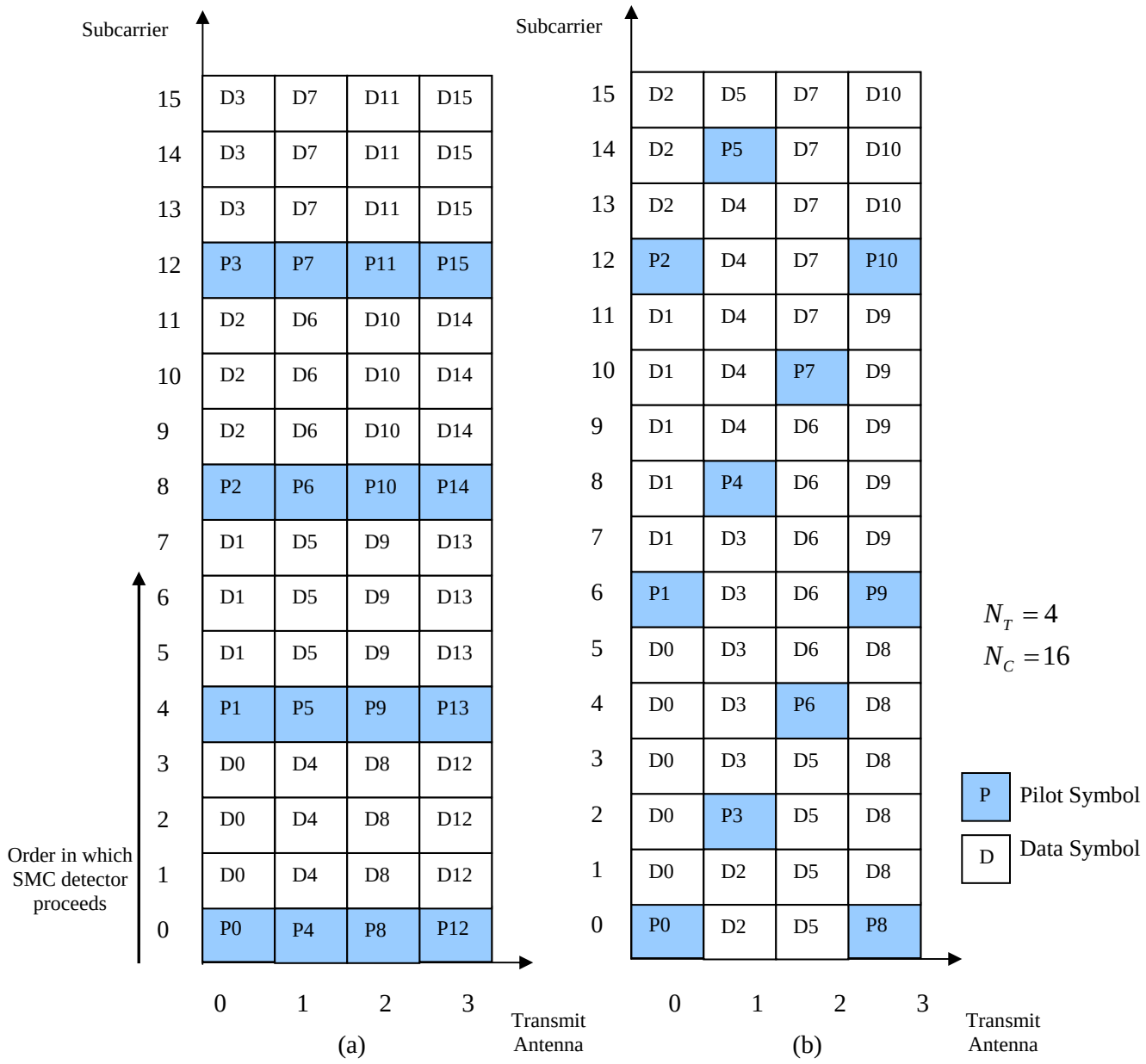


Fig. 25(a): Symbol grid for $K = 4$ and (b) $K = 6$ with $N_T = 4$ and $N_C = 16$

In general, each $\breve{\theta}_k$ comprises the target data transmitted from the i^{th} antenna, θ_k^i, the known pilots, $\breve{\theta}_{(pilot,k)}$ and the rest of the data $\breve{\theta}_{(rdata,k)}$ as shown

$$\breve{\theta}_k = \theta_k^i \cup \breve{\theta}_{(pilot,k)} \cup \breve{\theta}_{(rdata,k)} \tag{4.6}$$

where

$$\breve{\theta}_{(pilot,k)} = \left[\theta_{(pilot_0,k)}, \theta_{(pilot_1,k)}, ..., \theta_{\left(pilot_{P_k-1},k\right)}\right]^{\mathrm{T}} \tag{4.7}$$

$$\breve{\theta}_{(rdata,k)} = \left[\theta_{(rdata_0,k)}, \theta_{(rdata_1,k)}, ..., \theta_{\left(rdata_{N_T-P_k-2},k\right)}\right]^{\mathrm{T}} \tag{4.8}$$

where P_k denotes the number of pilots at the k^{th} SC and $P_k \leq N_T$ depending on the value of K used. As the pilots are known, each $\breve{\theta}_k$ can take on $|\mathrm{B}|^{N_T-P_k}$ possible permutations. For instance, with reference to Fig. 25(b), permutation of the symbols for the 3rd SC is defined as

$$\theta_{(perm,2)} = \left[\theta_2^0, \theta_{(pilot,2)}, \theta_2^2, \theta_2^3\right]^{\mathrm{T}} \tag{4.9}$$

where θ_2^0, θ_2^2 and θ_2^3 can each take on $|\mathrm{B}|$ possible values and $\theta_{(pilot,2)}$ is known. Therefore, while imputing for θ_2^0 and disregarding this target symbol and only considering the rest of the data symbols, there will be a total of $|\mathrm{B}|^2$ permutations to consider. For the rest of the discussions, the values taken on by $\breve{\theta}_{(pilot,k)}$ and $\breve{\theta}_{(rdata,k)}$ are denoted as

$$\breve{b}_{(pilot,k)} = \left[b_{(pilot_0,k)}, b_{(pilot_1,k)}, ..., b_{\left(pilot_{P_k-1},k\right)}\right]^{\mathrm{T}} \tag{4.10}$$

$$\breve{b}_{(rdata,k)} = \left[b_{(rdata_0,k)}, b_{(rdata_1,k)}, ..., b_{\left(rdata_{N_T-P_k-2},k\right)}\right]^{\mathrm{T}} \tag{4.11}$$

respectively.

To minimize the Monte Carlo variance, the optimal importance density is used as the predictive state distribution

$$p\left(\theta_k^i \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_k, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)}\right) \tag{4.12}$$

where θ_k^i is the transmitted symbol from the i^{th} transmit antenna at the k^{th} SC, $\theta_{(p,1:k-1)}^i$ is the imputed p^{th} particle for the transmitted symbol from the i^{th} transmit antenna up to the $k-1^{th}$ SC and $\breve{\theta}_{(pilot,k)}$ is the pilot knowledge at the k^{th} SC.

It is observed that (4.12) can be re-written as

$$\begin{aligned}
&P\left(\theta_k^i = b_l \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_k, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)}\right) \\
&= P\left(\theta_k^i = b_l \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_k, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right) \\
&= \frac{p\left(\breve{\alpha}_k \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta_k^i = b_l\right) P\left(\theta_k^i = b_l \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right)}{p\left(\breve{\alpha}_k \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right)} \\
&\propto p\left(\breve{\alpha}_k \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta_k^i = b_l\right) P\left(\theta_k^i = b_l \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right) \\
&= p\left(\breve{\alpha}_k \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta_k^i = b_l\right) P\left(\theta_k^i = b_l \mid \theta_{(p,k-1)}^i\right) \\
&= \beta_{(p,k,l)}^i.
\end{aligned} \tag{4.13}$$

The 3rd equality arises because $\left\{\theta_k^i\right\}$ is a first-order Markov chain and θ_k^i is independent of $\tilde{\boldsymbol{\alpha}}_{k-1}$. The term $P\left(\theta_k^i = b_l \mid \theta_{(p,k-1)}^i\right)$ refers to the a priori probability of the data symbol S_{iN_C+k-1}. As some of the symbols are known pilots, only the data symbols need to be imputed. The last equality is for clarity purposes.

To compute the probability term, $p\left(\breve{\alpha}_k \mid \theta_{(p,1:k-1)}^i, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta_k^i = b_l\right)$ given in (4.13), it is necessary to first consider the joint distribution

$p\left(\breve{\alpha}_k, \breve{\theta}_{(rdata,k)} = \breve{b}_{(rdata,k)} \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilots,k)}, \theta^i_k = b_l\right)$ and summing the resultant over all possible permutations of the rest of the data symbols except the target data, as follows

$$p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilots,k)}, \theta^i_k = b_l\right)$$
$$= \sum_{\forall b_{(rdata,k)} \in \mathrm{B}^{N_T - P_k - 1}} p\left(\breve{\alpha}_k, \breve{\theta}_{(rdata,k)} = \breve{b}_{(rdata,k)} \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilots,k)}, \theta^i_k = b_l\right). \tag{4.14}$$

Denoting the combination of the terms $\breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta^i_k = b_l, \breve{\theta}_{(rdata,k)} = \breve{b}_{(rdata,k)}$ as $\breve{\theta}_{(perm,k,l)}$, the term $p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right)$ is a multivariate Gaussian distribution with mean vector and covariance matrix as shown

$$\begin{aligned} \boldsymbol{\mu}^i_{(p,k,l)} &= \mathrm{E}\left\{\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right\} \\ &= \breve{H}_k \breve{\theta}_{(perm,k,l)} \\ &= \left[\mu^0_k, \mu^1_k, ..., \mu^{N_R-1}_k\right]^{\mathrm{T}} \end{aligned} \tag{4.15}$$

$$\boldsymbol{\Xi}^i_{(p,k,l)} = diag\left[\upsilon^0_k, \upsilon^1_k, ..., \upsilon^{N_R-1}_k\right]^{\mathrm{T}} \tag{4.16}$$

where υ^j_k is the variance of α^j_k and each $\upsilon^j_k = \sigma^2_w$. Therefore, it is quite apparent that $p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right)$ is the product of N_R 1-D Gaussian likelihood functions, each with mean and covariance μ^j_k and υ^j_k respectively.

With these, (4.14) can be easily computed as follows

$$
\begin{aligned}
& p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta^i_k = b_l\right) \\
&= \sum_{\breve{b}_{(rdata,k)} \in \mathrm{B}^{N_T - P_k - 1}} p\left(\breve{\alpha}_k, \breve{\theta}_{(rdata,k)} = \breve{b}_{(rdata,k)} \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}, \theta^i_k = b_l\right) \\
&= \sum_{\breve{b}_{(rdata,k)} \in \mathrm{B}^{N_T - P_k - 1}} p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right) P\left(\breve{\theta}_{(rdata,k)} = \breve{b}_{(rdata,k)} \mid \breve{\theta}_{(p,k-1)}\right) \\
&= \sum_{\breve{b}_{(rdata,k)} \in \mathrm{B}^{N_T - P_k - 1}} p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right) \prod_{r=0}^{N_T - P_k - 2} P\left(\theta_{(rdata_r,k)} = b_{(rdata_r,k)} \mid \theta_{(rdata_r,p,k-1)}\right) \\
&= \sum_{\breve{b}_{(rdata,k)} \in \mathrm{B}^{N_T - P_k - 1}} p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(perm,k,l)}\right) \prod_{r=0}^{N_T - P_k - 2} P\left(\theta_{(rdata_r,k)} = \theta_{(rdata_r,p,k-1)} b_{(rdata_r,k)}\right).
\end{aligned}
\tag{4.17}
$$

The product in the last equality is due to the independence of sub-trellises, and as shown in the last equality, the term $P\left(\theta_{(rdata_r,k)} = b_{(rdata_r,k)} \mid \theta_{(rdata_r,p,k-1)}\right)$ represents the a priori probabilities of the rest of the data symbols, besides the target data symbol, that are obtained from the SISO decoder in the previous iteration.

From (3.13), the update of the importance weights can be done recursively by

$$
w^i_{(p,k)} \propto w^i_{(p,k-1)} p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right) \tag{4.18}
$$

where

$$
p\left(\breve{\alpha}_k \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_{k-1}, \tilde{\mathbf{H}}_k, \breve{\theta}_{(pilot,k)} = \breve{b}_{(pilot,k)}\right) = \sum_{b_l \in \mathrm{B}} \beta^i_{(p,k,l)}. \tag{4.19}
$$

Therefore,

$$
w^i_{(p,k)} \propto w^i_{(p,k-1)} \sum_{b_l \in \mathrm{B}} \beta^i_{(p,k,l)}. \tag{4.20}
$$

A summary of the algorithm used by the SMC detector is presented in Table 5.

Table 5: Algorithm of SMC Detector in MIMO-OFDM Systems

1. Initialize all importance weights $w^i_{(p,0)} = 1$.
2. For each target data symbol at the k^{th} SC and the i^{th} transmit antenna and for $p = 1, 2, ..., \Omega$,

   For each $b_l \in \mathrm{B}$, compute $\beta^i_{(p,k,l)}$.

   Impute $\theta^i_{(p,k)}$ with probability $\beta^i_{(p,k,l)}$.

   Compute the importance weights, $w^i_{(p,k)} \propto w^i_{(p,k-1)} \sum_{\forall l} \beta^i_{(p,k,l)}$.

With this, the a posteriori symbol probability of the data symbol S_{iN_C+k} can be computed as

$$\begin{aligned} P\left(S_{iN_C+k} = b_l \mid \tilde{\boldsymbol{\alpha}}_k\right) &= P\left(\theta^i_{k-1}\theta^i_k = b_l \mid \tilde{\boldsymbol{\alpha}}_k\right) \\ &= \mathrm{E}\left\{\mathbf{I}\left(\theta^i_{k-1}\theta^i_k = b_l\right) \mid \tilde{\boldsymbol{\alpha}}_k\right\} \\ &\approx \frac{1}{W^i_k}\sum_{p=1}^{\Omega}\mathbf{I}\left(\theta^i_{(p,k-1)}\theta^i_{(p,k)} = b_l\right)w^i_{(p,k)}, \qquad b_l \in \mathrm{B} \end{aligned} \tag{4.21}$$

where $W^i_k = \sum_{p=1}^{\Omega} w^i_{(p,k)}$ and $\mathbf{I}(\bullet)$ is an indicator function where it is unity if the argument is true and zero if otherwise.

From the a posteriori symbol probability, the a posteriori LLRs of the interleaved code bits, $c_{\pi(i)}$ are computed

$$\begin{aligned} \Upsilon^{(1)}\left[c_{\pi(i)}\right] &= \log\frac{P\left(c_{\pi(i)} = 1 \mid \tilde{\boldsymbol{\alpha}}\right)}{P\left(c_{\pi(i)} = 0 \mid \tilde{\boldsymbol{\alpha}}\right)} \\ &= \log\frac{\sum_{b_l \in \mathrm{B}, S_j = b_l, c_{\pi(i)}=1} P\left(S_j = b_l \mid \tilde{\boldsymbol{\alpha}}\right)}{\sum_{b_l \in \mathrm{B}, S_j = b_l, c_{\pi(i)}=0} P\left(S_j = b_l \mid \tilde{\boldsymbol{\alpha}}\right)}. \end{aligned} \tag{4.22}$$

In (4.22), it is assumed that $c_{\pi(i)}$ is mapped to data symbol S_j, and applying Bayes' Rule, (4.22) can be decomposed into two components,

$$\begin{aligned}\Upsilon^{(1)}\left[c_{\pi(i)}\right] &= \log\frac{P\left(c_{\pi(i)}=1\right)}{P\left(c_{\pi(i)}=0\right)} + \log\frac{P\left(\tilde{\boldsymbol{\alpha}}\,|\,c_{\pi(i)}=1\right)}{P\left(\tilde{\boldsymbol{\alpha}}\,|\,c_{\pi(i)}=0\right)} \\ &= \lambda^{(2)}\left[c_{\pi(i)}\right] + \lambda^{(1)}\left[c_{\pi(i)}\right]\end{aligned} \tag{4.23}$$

where $\lambda^{(2)}\left[c_{\pi(i)}\right]$ is the interleaved version of the a priori LLR of the code bit $c_{\pi(i)}$ that is obtained from the SISO channel decoder during the previous iteration while $\lambda^{(1)}\left[c_{\pi(i)}\right]$ represents the extrinsic information about the code bit $c_{\pi(i)}$, delivered by the SMC detector based on the received signals $\tilde{\boldsymbol{\alpha}}$, the signal model and the a priori probabilities of all other code bits. The extrinsic information is deinterleaved and fed into the SISO channel decoder as a priori LLR of the code bits. With this, the SISO channel decoder, using the MAP channel decoding algorithm [34,57], will compute the a posteriori LLR of the code bit

$$\begin{aligned}\Upsilon^{(2)}\left[c_i\right] &= \frac{P\left(c_i=1\,|\,\text{code constraints}\right)}{P\left(c_i=0\,|\,\text{code constraints}\right)} \\ &= \lambda^{(1)}\left[c_i\right] + \lambda^{(2)}\left[c_i\right]\end{aligned} \tag{4.24}$$

subject to the code constraints. Removing the a priori LLR from the a posteriori LLR, the extrinsic information of the code bit $\lambda^{(2)}\left[c_{\pi(i)}\right]$ delivered by the channel decoder, is interleaved and the a priori symbol probability is computed from it.

Assuming that the QPSK symbol b_l corresponds to the bit pair $\left(\delta_{l_0}, \delta_{l_1}\right)$ and that the transmitted symbol S_i corresponds to the code bit pair $\left(c_{i_0}, c_{i_1}\right)$, the a priori symbol probability is then calculated as

$$P\left(S_i = b_l\right) = P\left(c_{i_0} = \delta_{l_0}\right) P\left(c_{i_1} = \delta_{l_1}\right) \tag{4.25}$$

where the code bit probability is given as

$$\begin{aligned} P\left(c_i = 1\right) &= \frac{\exp\left(\lambda^{(2)}\left[c_i\right]\right)}{1+\exp\left(\lambda^{(2)}\left[c_i\right]\right)} \\ &= \frac{\exp\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)}{\exp\left(-\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)+\exp\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)} \\ &= \frac{\cosh\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)\left[1+\tanh\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)\right]}{2\cosh\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)} \\ &= \frac{1+\tanh\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)}{2}. \end{aligned} \tag{4.26a}$$

Similarly, $P\left(c_i = 0\right)$ can be derived to be

$$\begin{aligned} P\left(c_i = 0\right) &= \frac{1}{1+\exp\left(\lambda^{(2)}\left[c_i\right]\right)} \\ &= \frac{1-\tanh\left(\frac{\lambda^{(2)}\left[c_i\right]}{2}\right)}{2}. \end{aligned} \tag{4.26b}$$

The computed symbol probabilities are fed back to the SMC detector as prior information for the next iteration. During the first iteration, all symbols are equi-probable and $\lambda^{(2)}\left[c_{\pi(i)}\right]$ is zero. The SISO SMC detector exchanges the extrinsic information with the SISO channel decoder iteratively to improve the performance of the receiver. However though the extrinsic information is uncorrelated during the first iteration, it gets progressively correlated as the number of iterations increase. Therefore, the gain obtained diminishes with increasing number of iterations.

4.4.4 Computational Complexity

In this section, a brief comparison of the computational complexity of the proposed non-resampling SMC detector with that of a resampling SMC detector for the same system, based on the same bandwidth, same number of subcarriers and pilot symbols and the same pilot locations is presented. In essence, resampling is being introduced into the proposed algorithm. Specifically, the number of multiplications and imputations necessary for each triplet (i, p, k), for the proposed algorithm, where $k = 0,1,..., N_C - 1$, $i = 0,1,..., N_T - 1$ and $p = 1,..., \Omega$ are calculated. The result of this investigation is summarized in Table 6 and it follows that the total number of multiplications and imputations needed for the proposed algorithm for each MIMO-OFDM symbol are $N_C\Omega\left(3N_R\left|\mathrm{B}\right|^{N_T-P_k} + N_T - P_k\right)$ and $N_C\Omega\left(N_T - P_k\right)$ respectively. For the system employing resampling, the number of times resampling is performed for each MIMO-OFDM symbol is $\left\lfloor \frac{N_C N_T - 1}{K} \right\rfloor$.

The number of multiplications necessary for each resampling step is Ω. Therefore,

the additional multiplications incurred by resampling for each MIMO-OFDM symbol is given by $\Omega\left\lfloor \frac{N_C N_T - 1}{K} \right\rfloor$ while there is no change to the number of imputations.

Table 6: Computational Complexity of Non-Resampling SMC Detector for a given triplet (i, p, k)

	Proposed Algorithm
Number of multiplications to compute each $\theta^i_{(p,k,l)}$	$3N_R \left\|\mathrm{B}\right\|^{N_T - P_k}$
Number of multiplications to update importance weights	1
Number of imputations	1

4.5 Simulation Results

In this section, the simulation results of the performance of the non-resampling SMC iterative receiver under various scenarios are presented. In this section, it is assumed that the carrier frequency and phase offset have already been compensated.

A $N_C = 64$ SC MIMO-OFDM system, each with a SC bandwidth of 12.5*kHz* transmitting over a frequency-selective multipath-fading environment is considered. QPSK modulation is employed and the termination period, K is set to 12. Various transmitter and receiver antenna arrangements are considered. Channels with UNI and EXP PDPs with a Doppler frequency of 200 Hz are used to represent different mobile environments. Each of these channels has length

$L = 5$ while the delay spreads for the UNI and EXP profiles are $T_d = 1.27\mu s$ and $T_d = 1.07\mu s$ respectively.

It is assumed that the coefficients of the channel's tap delay line are wide sense stationary narrowband complex Gaussian random processes with Rayleigh distribution with the same variance of $\frac{1}{N_R L}$, and are of band-limited Doppler power spectral density following Jakes' model [53]. The Jakes' model used is generated using the sum-of-sinusoids method proposed in [73]. The channels are taken to be block-fading where the channels remain static over the duration of one MIMO-OFDM symbol.

The total received energy of each transmitted symbol is normalized to unity [22] and it is assumed that the receiver has perfect knowledge of the CSI. In the simulations, the number of Monte Carlo particles used is 50 and $\frac{E_b}{N_0}$ is defined to be the ratio of the instantaneous bit power per transmit antenna to the average noise power [24].

Investigation of effects of various antenna arrangements

Firstly, the effect of various antenna arrangements on the performance of the MIMO-OFDM system, where the total number of antennas is fixed to 8, i.e. $N_T + N_R = 8$ over a UNI channel is considered. Three systems are simulated, namely, 2×6, 3×5 and 4×4 MIMO-OFDM. A rate ½, constraint length 5 convolutional code with generator polynomials $[23\ 35]_{\mathbf{o}}$ is employed, and the BCJR [57] MAP decoding algorithm is used by the SISO channel decoder. The number of iterations between the SMC detector and the channel decoder is set to 4 and the BCJR algorithm makes hard decisions on the a posteriori LLRs of the data bits $\{d_m\}$ at the end of the 4^{th} turbo iteration. The interleaver used is randomly generated and fixed throughout the simulation. The BER performances of these systems are shown in Fig. 26.

It can be seen that the 2×6 MIMO-OFDM system offers the best BER performance, giving an improvement of 1.3dB when compared to the 4×4 MIMO-OFDM system. This result implies that for a fixed number of total antennas under the same channel conditions, the MIMO system with the biggest number of receive antennas will have the best BER performance. This is explained by the decrease in ICI and the increase in receiver diversity when N_T is decreased and N_R is increased.

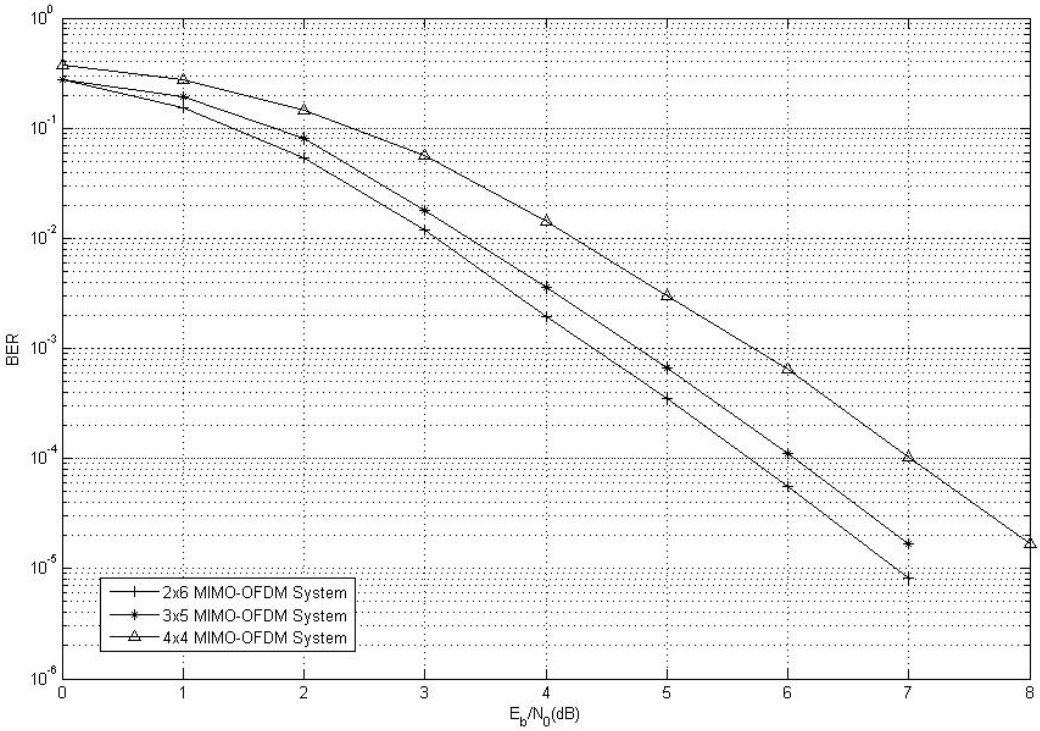


Fig. 26: Comparisons of various antenna arrangements for $N_T + N_R = 8$ Convolutional coded MIMO-OFDM system for data transmitted over a UNI channel with $T_d = 1.27\mu s$ and $K = 12$

Investigation of effects of different termination periods

The effect of different termination periods on the BER performance is also investigated. A 4×4 MIMO-OFDM system with signals transmitting over a UNI channel is considered. As seen in Fig. 27, as the termination period K, decreases from 12 to 3, there is a 1.3dB improvement in performance. This suggests that with a frequent termination, better performance can be achieved at the expense of increased overheads. This is to be expected as in the absence of resampling, the Monte Carlo variance of the importance weights increases with each step of the SIS. Therefore the estimation becomes more inferior as time passes. With a more

frequent termination, the increase in Monte Carlo variance is kept in check hence resulting in better performance. For comparison purposes, the performance of MMSE detection is also simulated. As degeneracy is not inherent in MMSE detection, its performance therefore does not depend on K and so only one case of K is considered. As observed, the performance of MMSE detection with $K = 12$ is poor, compared to the proposed SMC algorithm. This is because MMSE detection only offers a diversity order of 1 for the case of a 4x4 MIMO–OFDM system [75].

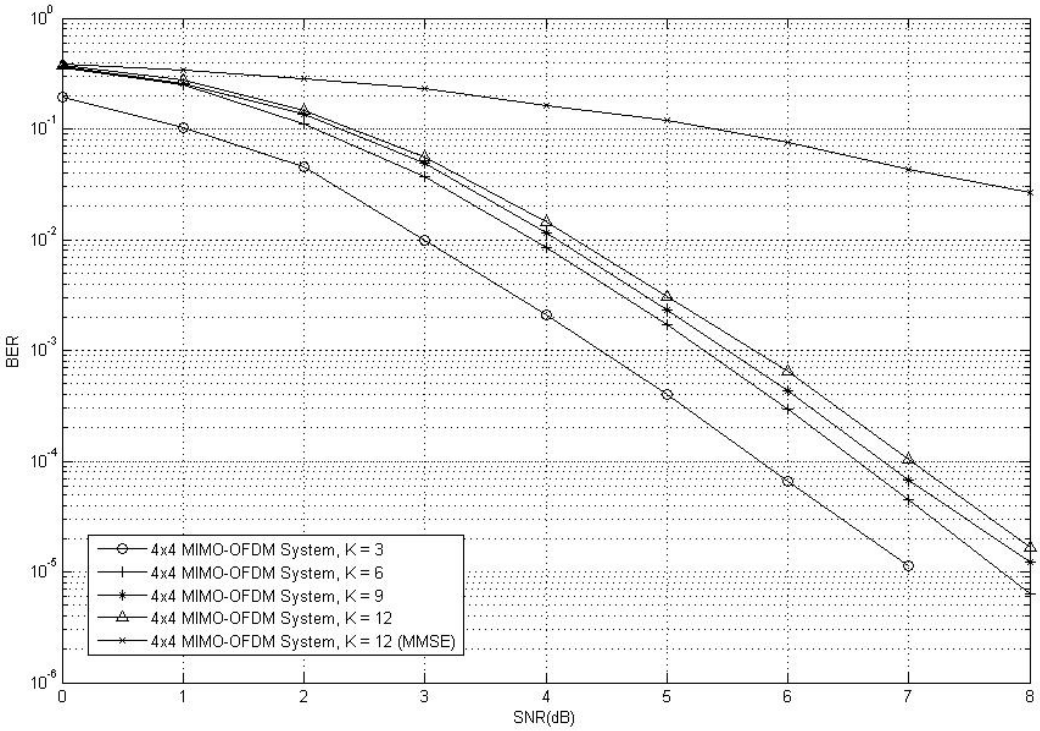


Fig. 27: Effect of different termination periods on performance of a 4×4 Convolutional coded MIMO-OFDM system for data transmitted over a UNI channel with $T_d = 1.27 \mu s$

Investigation of effects of different PDPs

The performances of the 4×4 MIMO-OFDM system under different environments are also simulated. Two types of PDPs are considered, UNI with $T_d = 1.27\mu s$ and EXP with $T_d = 1.07\mu s$. The value of K is maintained at 12. The result of this comparison is shown in Fig. 28. It can be seen that the best performance is achieved in a UNI channel with a gain of 0.3dB over the EXP channel.

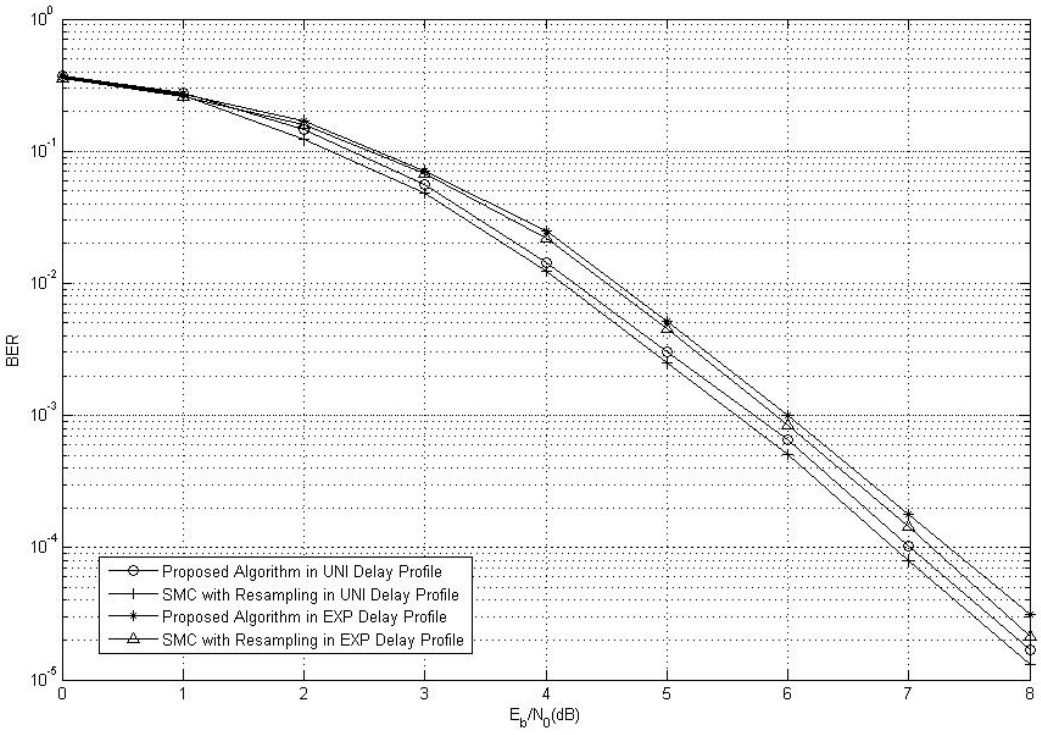


Fig. 28: Performance of a 4×4 Convolutional coded MIMO-OFDM system for data transmitted over a UNI channel with $T_d = 1.27\mu s$ and EXP channel with $T_d = 1.07\mu s$, and $K = 12$

The performance of the proposed detector is also compared with the performance of a SMC detector with resampling employed. For the latter, deterministic

resampling is used and it is performed once in between every two pilot symbols. Therefore, this serves as a lower bound for the performance of the proposed detector. It can be seen that in both channel environments, the proposed detector performs close to the respective lower bounds. For instance, in the UNI channel, the performance is only 0.13 dB away from its lower bound while in the EXP delay profile the performance is only 0.2dB away.

Investigation of effects of various antenna arrangements for LDPC coded system

Besides convolutional coded system, LDPC coded MIMO-OFDM system over the UNI channel is also investigated. Three different antenna arrangements are considered, namely, 2×2, 3×3 and 4×4. The value of K is kept at 12. A rate ½ LDPC code of length 1404 is used. The row weights of the parity check matrix is maintained as uniform as possible while the column weight is fixed at 3. It is also ensured that the parity check matrix does not have cycles of 4. The SPA [34] that is used within the SISO decoder iterates a maximum of 100 cycles while the number of iterations between the SMC detector and the SISO decoder is kept at 5. No interleaver is used in this case. The simulation results are shown in Fig. 29.

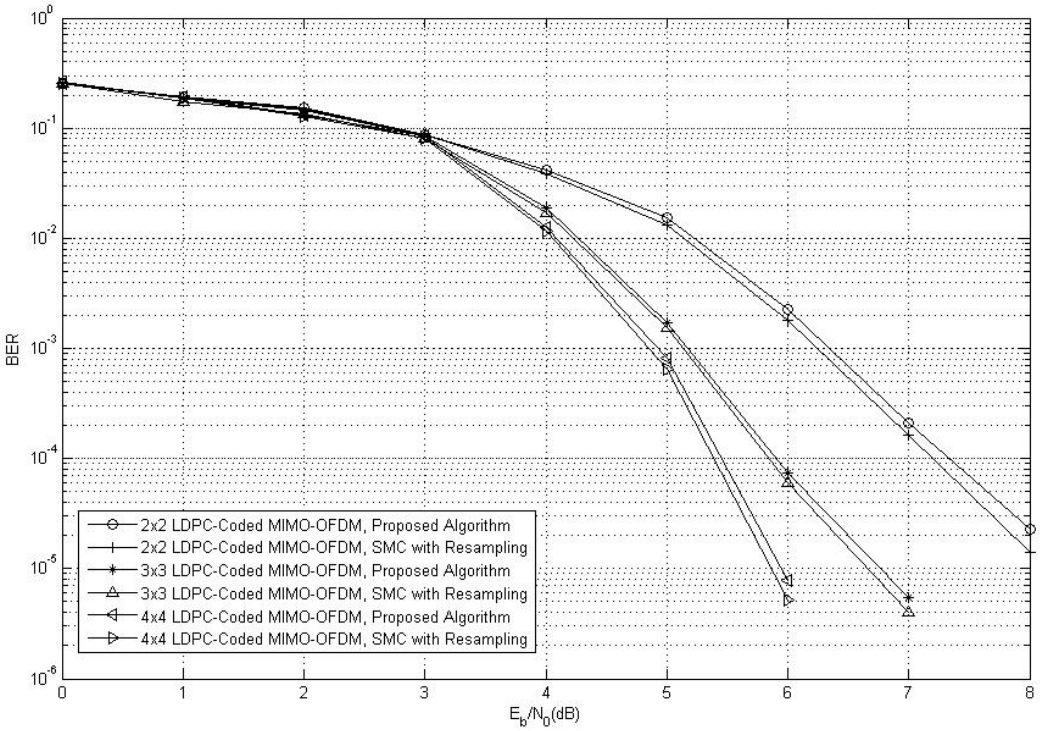


Fig. 29: Comparisons of various antenna arrangements for LDPC coded MIMO-OFDM system for data transmitted over a UNI channel with $T_d = 1.27\mu s$, $K = 12$, and 5 turbo iterations

From Fig. 29, it can be seen that the BER performance of the proposed detector is very similar to the performance of its corresponding lower bound for all the three cases. For instance, for the 2×2 MIMO-OFDM system, the performance degradation of the proposed detector is 0.2dB, for the 3×3 system the degradation is 0.1dB while for the 4×4 system the degradation is only 0.07dB. It can also be observed that with the same antenna arrangement, the LDPC coded MIMO-OFDM system performs better than the convolutional coded system.

4.6 Conclusions

This section investigates the design and performance of a non-resampling SMC detector as the first-stage of an iterative receiver for coded MIMO-OFDM systems under different environments. The proposed receiver iterates between the SMC detector and the MAP channel decoding stage. The extrinsic a posteriori LLRs of the code bits and the a posteriori symbol probabilities are exchanged between these two stages iteratively to improve the performance of the receiver.

The proposed detector employs periodic termination on the differentially encoded symbols in an attempt to keep degeneracy in check. Therefore, with this proposed receiver, the problems associated with resampling such as the high computational complexity involved, the infeasibility of parallel processing and the issue of sample impoverishment have all been eliminated.

It has been shown from the simulations that receiver diversity plays a large part in the performance of the receiver. A system with a larger number of receive antennas will have a better performance. It is also observed that in all cases, the proposed receiver consistently performs closely to its lower bound, which is taken to be the performance of the SMC detector with resampling employed under the same conditions.

It is also seen that with a more frequent periodic termination, better performance can be achieved at the expense of higher overheads. Even though overheads are

incurred, they can be used as pilot symbols for channel estimation purposes as will be discussed in the next chapter.

CHAPTER 5

ITERATIVE RECEIVER DESIGN FOR MIMO-OFDM SYSTEMS VIA SMC TECHNIQUES WITH PILOT AIDED CHANNEL ESTIMATION (PACE)

5.1 Background

There are generally two main types of channel estimation methods for OFDM systems, namely, blind and training (pilot) based. In the blind channel estimation method, the receiver has to estimate the CSI without the aid of known symbols. Even though they can achieve higher bandwidth efficiency due to the absence of training overheads, the estimation accuracy is compromised and usually the complexity involved is also higher. For these reasons, they are less widely used than training based estimations. In training based estimation methods, known pilot symbols or training sequences are transmitted to help the receiver to estimate the CSI.

This chapter starts with the extension of PACE for single-input single-output OFDM system to the case of MIMO-OFDM systems, followed by the simulation results of the coded MIMO-OFDM systems utilizing PACE [74] and the conclusions.

5.2 System Model of Coded MIMO-OFDM System with Channel Estimation

For the channel estimation purposes, it is assumed that the carrier frequency and phase offset have already been compensated. The structure of the transmitter remains the same as Fig. 23. It is replicated here for ease of reference.

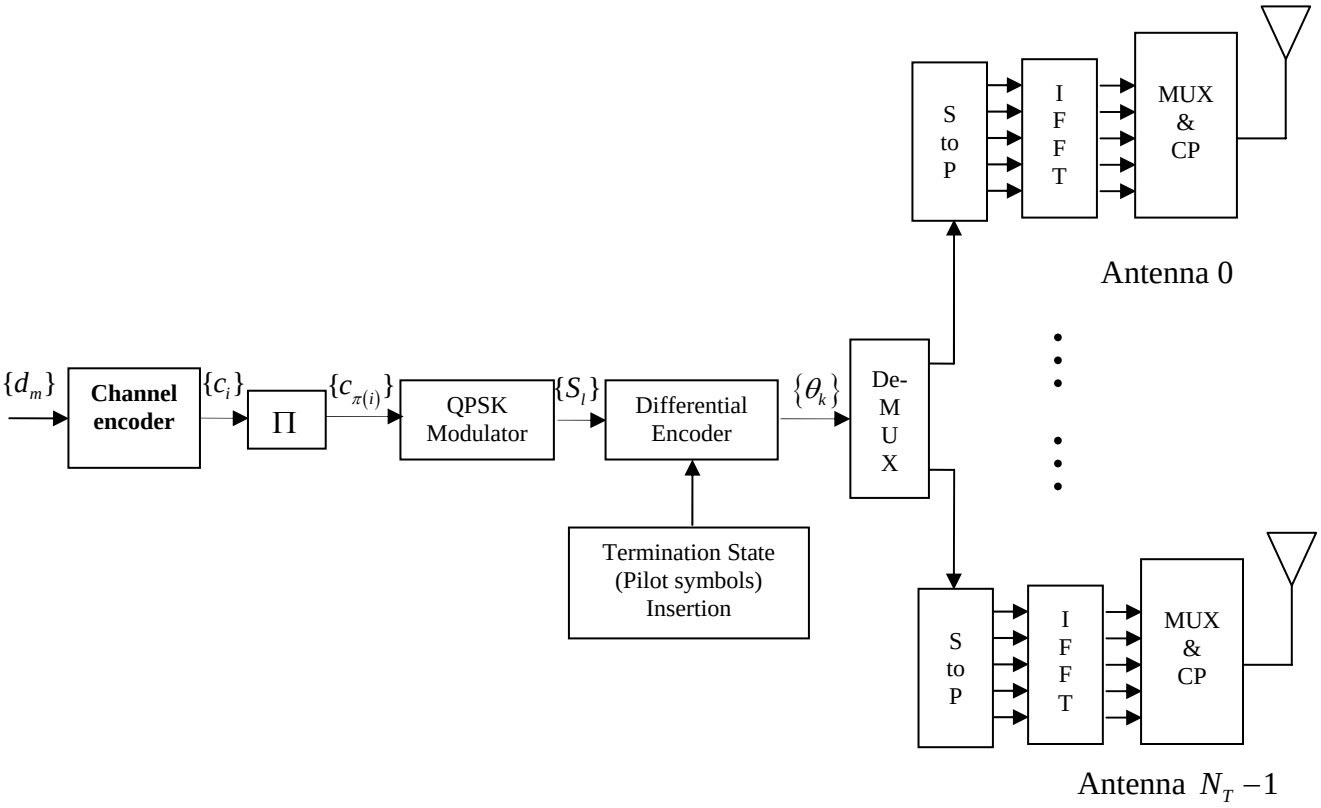


Fig. 30: Structure of proposed transmitter

While there is a slight addition to the structure of the receiver as shown in Fig. 31.

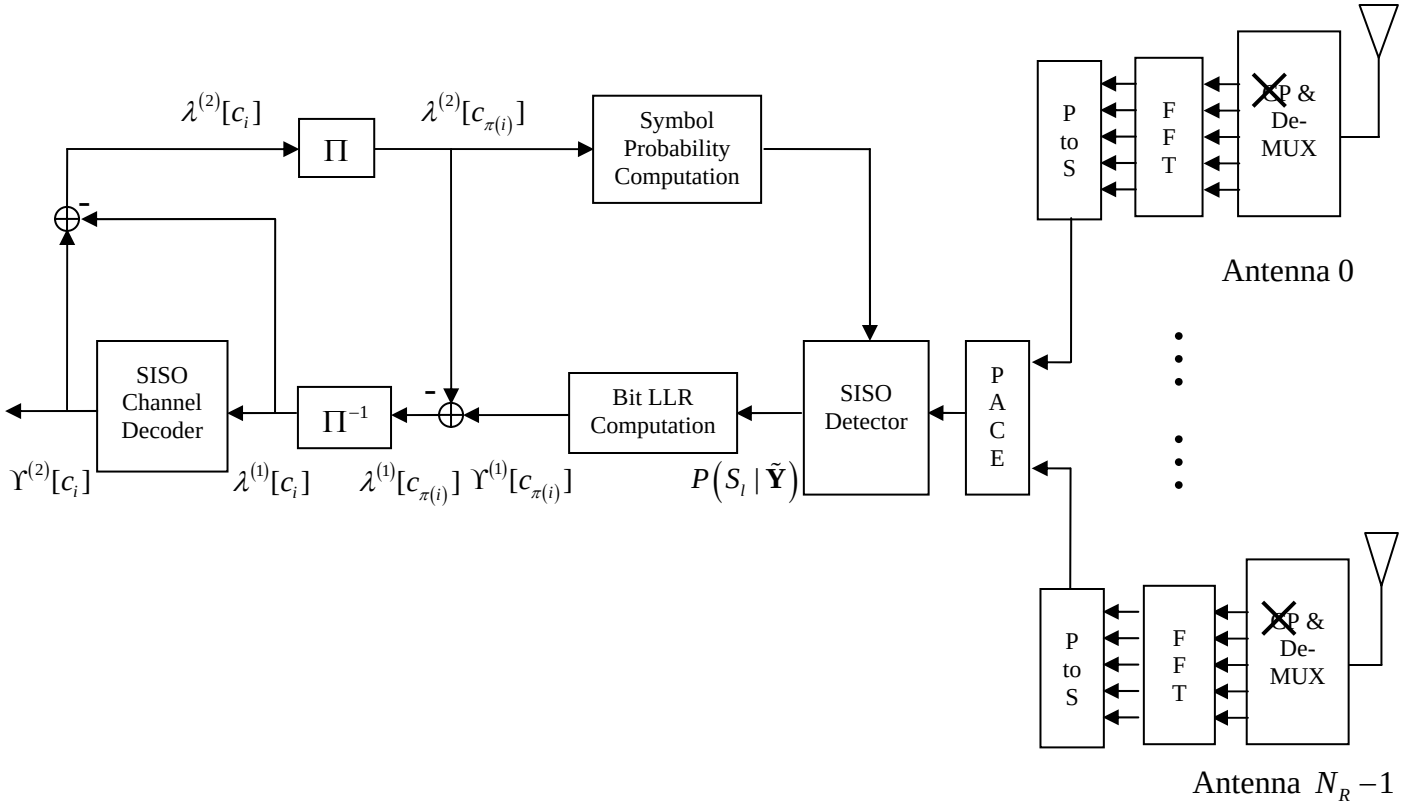


Fig. 31: Structure of proposed receiver

PACE is performed after the FFT operation, before data detection by the SMC detector. Since each SC is flat fading, estimation of the CSI ought to be rather trivial if not for the interference from other transmit antennas which are transmitting symbols at the same time coupled with the larger number of channel responses to be estimated.

The proposed PACE method adopts the joint pilot grid (JPG) method that Auer employed in [49-50]. In this case, all the transmit antennas will be transmitting pilot symbols simultaneously on a certain set of SCs and the termination period, K is a factor of N_C. In this work orthogonal sequences are used. The total

number of pilots per MIMO-OFDM symbol is $N_T N_P$ where N_P denotes the number of pilot symbols per MIMO-OFDM symbol per transmit antenna. It is assumed that the channel is static for the duration of N_T MIMO-OFDM symbols.

To employ PACE, it is necessary to fulfill the sampling theorem (2.14) and (2.15). In the proposed system, (2.14) is automatically satisfied as each MIMO-OFDM symbol has its own set of pilot symbols due to periodic termination. The number of pilots per transmit antenna per MIMO-OFDM symbol along the frequency axis should also follow the constraint imposed by (2.15) as shown in the following section.

Recall the MIMO-OFDM model is given by

$$\tilde{\boldsymbol{\alpha}} = \tilde{\mathbf{H}}\tilde{\boldsymbol{\theta}} + \tilde{\mathbf{W}} \tag{5.1}$$

where $\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^{(0,0)} & \cdots & \mathbf{H}^{(N_T-1,0)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}^{(0,N_R-1)} & \cdots & \mathbf{H}^{(N_T-1,N_R-1)} \end{bmatrix}$ and each $\mathbf{H}^{(i,j)} = diag\left(\bar{\mathbf{H}}^{(i,j)}\right)$. Each $\bar{\mathbf{H}}^{(i,j)} = \left[H_0^{(i,j)}, H_1^{(i,j)}, \cdots, H_{N_C-1}^{(i,j)}\right]^{\mathrm{T}} = \mathrm{FFT}\left(h^{(i,j)}\right)$.

If there is no aliasing, $\bar{\mathbf{H}}^{(i,j)}$ can be perfectly reconstructed by applying a low pass filter. For this to happen, the OFDM system has to fulfill the sampling theorem (2.15) [42-43], stated as

$$T_m^{(i,j)} \Delta f N_{Pf} \leq \frac{1}{2} \tag{5.2}$$

where Δf denotes the bandwidth of each SC, N_{Pf} denotes the spacing between the pilots in the frequency domain. In the case of MIMO, it is assumed that the maximum delay spreads of the channel between the i^{th} transmit antenna and the j^{th} receive antenna are the same for all, i.e. $T_m^{(i,j)} = T_m$. As the no-aliasing duration of $\frac{1}{\Delta f N_{Pf}}$ remains constant while the number of transmit antennas is now N_T [50], the sampling theorem has to be restated as

$$\begin{aligned} N_{Pf} N_T \frac{T_m}{T_{FFT}} &\le \frac{1}{2} \\ N_T \frac{T_m}{N_{FFT} T_S} \frac{N_C}{N_P} &\le \frac{1}{2} \\ N_P &\ge 2 N_T \frac{T_m N_C}{T_S N_{FFT}} \\ N_P &\ge 2 N_T \frac{T_m}{T_S} \end{aligned} \tag{5.3}$$

where the simplification in the last equation is due to $N_C = N_{FFT}$ in the system.

The PACE for the MIMO-OFDM system is based on the LS method, which will be derived in the following section with the aid of Fig. 32. Fig. 32 considers a simple setup of 2 transmit antennas and 2 receive antennas and 8 SCs. The termination period, K used is 4 and $N_P = 2$. Before proceeding on, some notations are necessary. Let $\theta_{(p,t)}^{(i)}$ denote the p^{th} pilot symbol transmitted from the i^{th} antenna during the t^{th} MIMO-OFDM symbol while $\alpha_{(p,t)}^{(j)}$ denotes the p^{th} pilot symbol received at the j^{th} antenna during the t^{th} MIMO-OFDM symbol.

Let $\widehat{\mathbf{H}}^{(i,j)} = \left[\widehat{H}_0^{(i,j)}, \widehat{H}_1^{(i,j)} ..., \widehat{H}_{N_p-1}^{(i,j)} \right]^{\mathrm{T}}$ [2] denote a subset of the N_{FFT}-point FFT of the CIR from the i^{th} transmit antenna to the j^{th} receive antenna. Note that $\widehat{\mathbf{H}}^{(i,j)}$ only contains a subset of the original CTF, $\bar{\mathbf{H}}^{(i,j)} = \left[H_0^{(i,j)}, H_1^{(i,j)}, \cdots, H_{N_C-1}^{(i,j)} \right]^{\mathrm{T}}$. It is apparent that $\widehat{\mathbf{H}}^{(i,j)} = \left[H_0^{(i,j)}, H_K^{(i,j)} ..., H_{(N_p-1)K}^{(i,j)} \right]^{\mathrm{T}}$, where each element is spaced K symbols away from the other. In Fig. 32, $\widehat{H}_{(p,t)}^{(i,j)}$ denotes the FFT of the CIR between the i^{th} transmit and the j^{th} receive antenna of the p^{th} SC during the t^{th} MIMO-OFDM symbol.

The channel is assumed to be noiseless and static for the duration of two MIMO-OFDM symbols. The pilot sequence $\left[\theta_{(0,0)}^{(0)}, \theta_{(0,1)}^{(0)} \right]^{\mathrm{T}}$ is orthogonal to $\left[\theta_{(0,0)}^{(1)}, \theta_{(0,1)}^{(1)} \right]^{\mathrm{T}}$ while $\left[\theta_{(1,0)}^{(0)}, \theta_{(1,1)}^{(0)} \right]^{\mathrm{T}}$ is orthogonal to $\left[\theta_{(1,0)}^{(1)}, \theta_{(1,1)}^{(1)} \right]^{\mathrm{T}}$.

[2] As a convention, variables pertaining to pilot symbols will be marked with a $\widehat{\bullet}$ in the following sections.

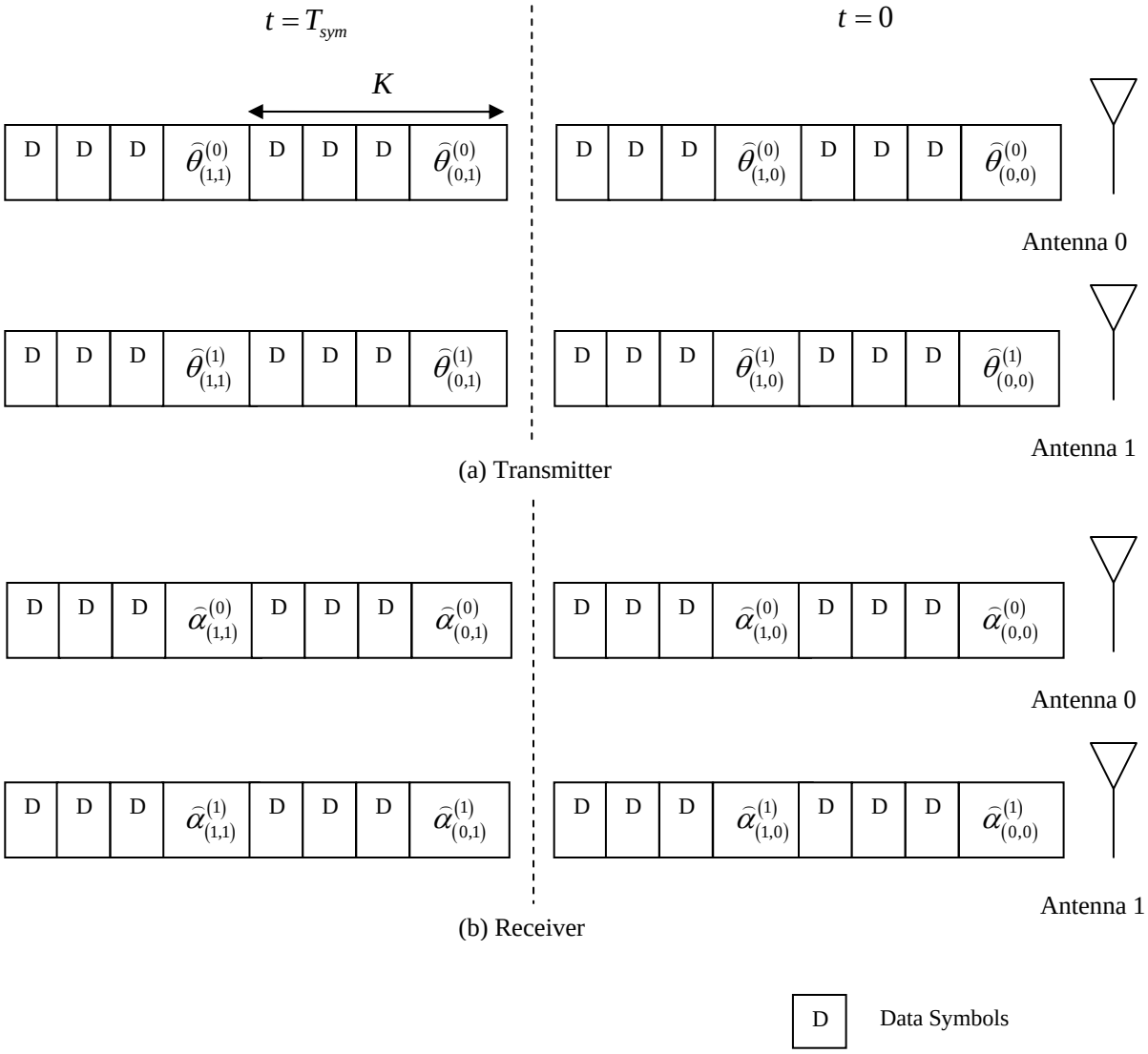


Fig. 32: Pilot arrangement for 2×2 MIMO-OFDM system

At $t = 0$,

$$\begin{bmatrix} \hat{\alpha}^{(0)}_{(0,0)} \\ \hat{\alpha}^{(0)}_{(1,0)} \\ \hat{\alpha}^{(1)}_{(0,0)} \\ \hat{\alpha}^{(1)}_{(1,0)} \end{bmatrix} = \begin{bmatrix} \hat{H}^{(0,0)}_{(0,0)} & 0 & \hat{H}^{(1,0)}_{(0,0)} & 0 \\ 0 & \hat{H}^{(0,0)}_{(1,0)} & 0 & \hat{H}^{(1,0)}_{(1,0)} \\ \hat{H}^{(0,1)}_{(0,0)} & 0 & \hat{H}^{(1,1)}_{(0,0)} & 0 \\ 0 & \hat{H}^{(0,1)}_{(1,0)} & 0 & \hat{H}^{(1,1)}_{(1,0)} \end{bmatrix} \begin{bmatrix} \hat{\theta}^{(0)}_{(0,0)} \\ \hat{\theta}^{(0)}_{(1,0)} \\ \hat{\theta}^{(1)}_{(0,0)} \\ \hat{\theta}^{(1)}_{(1,0)} \end{bmatrix} \quad (5.4a)$$

Simplify (5.4a) as

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_0 &= \begin{bmatrix} diag\left(\widehat{\mathbf{H}}_0^{(0,0)}\right) & diag\left(\widehat{\mathbf{H}}_0^{(1,0)}\right) \\ diag\left(\widehat{\mathbf{H}}_0^{(0,1)}\right) & diag\left(\widehat{\mathbf{H}}_0^{(1,1)}\right) \end{bmatrix} \widehat{\boldsymbol{\theta}}_0 \\ &= \widehat{\mathbf{H}}_0 \widehat{\boldsymbol{\theta}}_0 \end{aligned} \tag{5.4b}$$

and the system at $t = T_{sym}$ can similarly be expressed as

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_1 &= \widehat{\mathbf{H}}_1 \widehat{\boldsymbol{\theta}}_1 \\ &= \widehat{\mathbf{H}}_0 \widehat{\boldsymbol{\theta}}_1 \end{aligned} \tag{5.5}$$

where $\widehat{\boldsymbol{\theta}}_1 = \left[\theta_{(0,1)}^{(0)}, \theta_{(1,1)}^{(0)}, \theta_{(0,1)}^{(1)}, \theta_{(1,1)}^{(1)}\right]^{\mathrm{T}}$ and $\widehat{\boldsymbol{\alpha}}_1 = \left[\alpha_{(0,1)}^{(0)}, \alpha_{(1,1)}^{(0)}, \alpha_{(0,1)}^{(1)}, \alpha_{(1,1)}^{(1)}\right]^{\mathrm{T}}$ as the channel remains unchanged for the 2nd MIMO-OFDM symbol.

Recall that the pilot sequence $\left[\theta_{(0,0)}^{(0)}, \theta_{(0,1)}^{(0)}\right]^{\mathrm{T}}$ is orthogonal to $\left[\theta_{(0,0)}^{(1)}, \theta_{(0,1)}^{(1)}\right]^{\mathrm{T}}$ while $\left[\theta_{(1,0)}^{(0)}, \theta_{(1,1)}^{(0)}\right]^{\mathrm{T}}$ is orthogonal to $\left[\theta_{(1,0)}^{(1)}, \theta_{(1,1)}^{(1)}\right]^{\mathrm{T}}$, $\widehat{\mathbf{H}}_1$ can be obtained via

$$\widehat{\mathbf{H}}_1 = \frac{1}{N_T} conj\left[\widehat{\boldsymbol{\theta}}_0 \quad \widehat{\boldsymbol{\theta}}_1\right] \begin{bmatrix} \widehat{\boldsymbol{\alpha}}_0{}^{\mathrm{T}} \\ \widehat{\boldsymbol{\alpha}}_1{}^{\mathrm{T}} \end{bmatrix}. \tag{5.6}$$

In general, for N_T transmit antennas and while transmitting the t^{th} MIMO-OFDM symbol, $\widehat{\mathbf{H}}_t = \begin{bmatrix} diag\left(\widehat{\mathbf{H}}_t^{(0,0)}\right) & \cdots & diag\left(\widehat{\mathbf{H}}_t^{(N_T-1,0)}\right) \\ \vdots & \ddots & \vdots \\ diag\left(\widehat{\mathbf{H}}_t^{(0,N_R-1)}\right) & \cdots & diag\left(\widehat{\mathbf{H}}_t^{(N_T-1,N_R-1)}\right) \end{bmatrix}$ can be easily obtained by

$$\widehat{\mathbf{H}}_t = \frac{1}{N_T} conj\left(\underline{\widehat{\boldsymbol{\theta}}}_t\right) \underline{\widehat{\boldsymbol{\alpha}}}_t, \tag{5.7}$$

where $\underline{\hat{\theta}}_t = \left[\hat{\theta}_{t-N_T+1} \quad \cdots \quad \hat{\theta}_t\right]$ and $\underline{\hat{\alpha}}_t = \begin{bmatrix} \hat{\alpha}_{t-N_T+1}^{\mathrm{T}} \\ \vdots \\ \hat{\alpha}_t^{\mathrm{T}} \end{bmatrix}$.

As all $\hat{\mathbf{H}}^{(i,j)}$ have been estimated, it is now necessary to interpolate the estimated channel values across all the data SCs. With the help of periodic termination, the highly complex, computationally intensive 2-D Wiener filtering is avoided as each MIMO-OFDM symbol has its own set of pilot symbols, hence no interpolation across time is required. Therefore, only a simple 1-D interpolation across frequency is necessary. This is illustrated in Fig. 33 where the termination period used is 4.

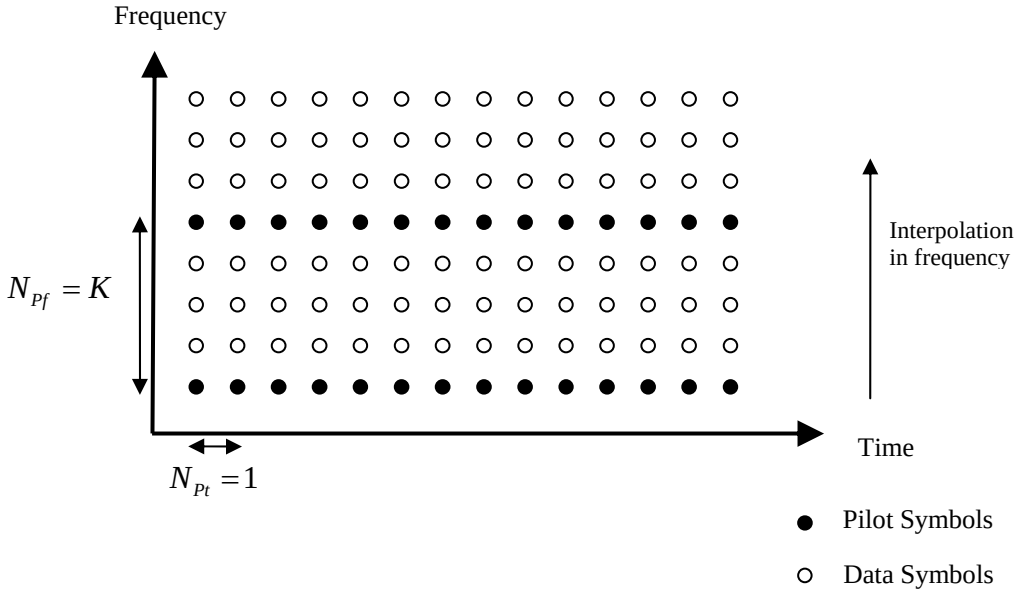


Fig. 33: Scattered pilot symbols over the 2-D frequency-time grid with $K = 4$

By considering each $\widehat{\mathbf{H}}^{(i,j)}$ independently, it is possible to interpolate each N_P-long $\widehat{\mathbf{H}}^{(i,j)}$ vector to the original N_C-long $\bar{\mathbf{H}}^{(i,j)}$ vector by using the interpolation method described in section 2.5.2. As each $\widehat{\mathbf{H}}^{(i,j)}$ is considered independently, the subscripts (i,j) are dropped for clarity. The MMSE estimate of $\bar{\mathbf{H}}$ is given as [64],

$$\hat{\bar{\mathbf{H}}}_{MMSE} = \mathrm{R}_{\bar{\mathbf{H}}\widehat{\mathbf{H}}} \mathrm{R}_{\widehat{\mathbf{H}}\widehat{\mathbf{H}}}^{-1} \widehat{\mathbf{H}} \tag{5.8}$$

where $\mathrm{R}_{\bar{\mathbf{H}}\widehat{\mathbf{H}}}$ is the cross covariance matrix between $\bar{\mathbf{H}}$ and $\widehat{\mathbf{H}}$ and $\mathrm{R}_{\widehat{\mathbf{H}}\widehat{\mathbf{H}}}$ is the auto correlation of $\widehat{\mathbf{H}}$. With the knowledge of $\widehat{\mathbf{H}}$, (5.8) minimizes the MSE between the actual CTF, $\bar{\mathbf{H}}$ and the estimates, $\hat{\bar{\mathbf{H}}}_{MMSE}$. The MIMO channel matrix can be re-written as $\hat{\mathbf{H}}_{MMSE} = \begin{bmatrix} \hat{\mathbf{H}}_{MMSE}^{(0,0)} & \cdots & \hat{\mathbf{H}}_{MMSE}^{(N_T-1,0)} \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{H}}_{MMSE}^{(0,N_R-1)} & \cdots & \hat{\mathbf{H}}_{MMSE}^{(N_T-1,N_R-1)} \end{bmatrix}$ where $\hat{\mathbf{H}}_{MMSE}^{(i,j)} = diag\left(\hat{\bar{\mathbf{H}}}_{MMSE}^{(i,j)}\right)$.

The SMC receiver algorithm proceeds as presented in section 4.4.3 with these estimates instead of the perfect channel knowledge assumed earlier in chapter 4. This is summarized in Table 7.

Table 7: Algorithm of SMC Detector with Channel Estimation in MIMO-OFDM Systems

1. Perform channel estimation for each $\bar{\mathbf{H}}^{(i,j)}$.
   Perform LS estimation to obtain $\widehat{\mathbf{H}}^{(i,j)}$.
   Interpolate $\widehat{\mathbf{H}}^{(i,j)}$ to obtain $\hat{\bar{\mathbf{H}}}_{MMSE}^{(i,j)}$.
2. Initialize all importance weights $w_{(p,0)}^{i} = 1$.
3. For each target data symbol at the k^{th} SC and the i^{th} transmit antenna and for $p = 1, 2, ..., \Omega$,

For each $b_l \in \mathrm{B}$, compute $\hat{\beta}^i_{(p,k,l)} = P\left(\theta^i_k = b_l \mid \theta^i_{(p,1:k-1)}, \tilde{\boldsymbol{\alpha}}_k, \hat{\mathbf{H}}_{MMSE_k}, \breve{\theta}_{(pilot,k)}\right)$.

Impute $\theta^i_{(p,k)}$ with probability $\hat{\beta}^i_{(p,k,l)}$.

Compute the importance weights, $w^i_{(p,k)} \propto w^i_{(p,k-1)} \sum_{\forall l} \hat{\beta}^i_{(p,k,l)}$.

5.3 Simulation Results

In this section, the simulation results of the performance of the non-resampling SMC iterative receiver with channel estimation incorporated under various scenarios are presented. It is assumed that the carrier frequency and phase offset have already been compensated.

Similar to previous simulations, a $N_C = 64$ SC MIMO-OFDM system, each with a SC bandwidth of $12.5kHz$ transmitting over a frequency-selective multipath-fading environment is considered. QPSK modulation is employed and in order to satisfy (5.3), the termination period, K is set to 4 for 4×4 system while it can be 4 or 8 for the 2×2 system. Various transmitter and receiver antenna arrangements are considered. Channels with UNI and EXP PDPs with a Doppler frequency of 40 Hz are used to model different mobile environments. Each of these channels has length $L = 3$ while the delay spreads for the UNI and EXP profiles are $T_d = 1.02\mu s$ and $T_d = 0.814\mu s$ respectively.

It is assumed that the coefficients of the channel's tap delay line are wide sense stationary narrowband complex Gaussian random processes with Rayleigh distribution with the same variance of $\frac{1}{N_R L}$, and are of band-limited Doppler

power spectral density following Jakes' model [53], generated using the sum-of-sinusoids method proposed in [73]. The channels are taken to be block-fading where the channels remain static over the duration of N_T MIMO-OFDM symbols. As before, the total received energy of each transmitted symbol is normalized to unity [22], the number of Monte Carlo particles used is 50 and $\frac{E_b}{N_0}$ is defined to be the ratio of the instantaneous bit power per transmit antenna to the average noise power [24].

Investigation of effects of different termination periods

The effect of different termination periods on the BER performance is investigated. A 2×2 MIMO-OFDM system with signals transmitting over a UNI channel is considered. As before, a rate ½, constraint length 5 convolutional code with generator polynomials $[23\ 35]_{\mathbf{o}}$ is employed, and the BCJR [57] MAP decoding algorithm is adopted by the SISO channel decoder. The number of iterations between the SMC detector and the channel decoder is kept at 4 and the BCJR algorithm makes hard decisions on the a posteriori LLRs of the data bits $\{d_m\}$ at the end of the 4^{th} iteration. Again, the interleaver used is randomly generated and fixed throughout the simulation.

As seen in Fig. 34, as the termination period K, decreases from 8 to 4, the degradation incurred by the incorporation of channel estimation is successively smaller, from 0.79dB to 0.65dB when comparing with the corresponding system

with perfect knowledge of CSI. This is to be expected as with a longer termination period, the errors incurred in the channel estimation are larger as the effects of degeneracy have kicked in. This led to the imputations not being as accurate as the case with knowledge of CSI and has slightly affected the assumption that the multivariant Gaussian distribution can be approximated by the product of N_R 1-D Gaussian likelihood functions. In fact, comparing among the systems with channel estimation, when K is increased from 4 to 8 the degradation is 0.50dB while for the systems with perfect knowledge of CSI, i.e. where no channel estimation is performed, the degradation is only 0.34dB.

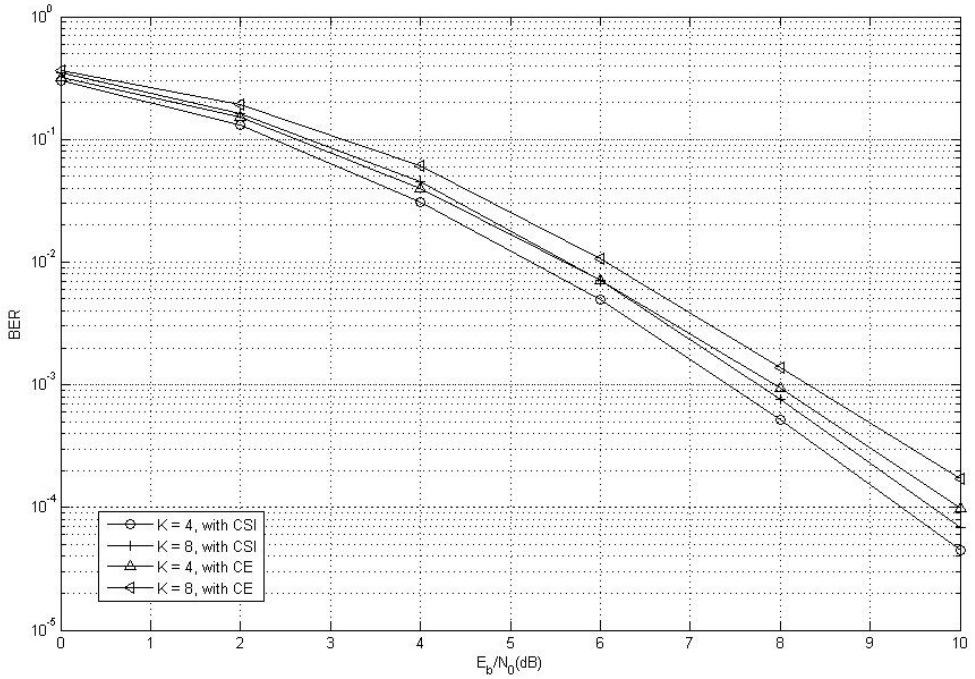


Fig. 34: Effect of different termination periods on performance of a 2×2 Convolutional coded MIMO-OFDM system with PACE for data transmitted over a UNI channel with $T_d = 1.02\mu s$

Investigation of effects of various antenna arrangements

The effect of various antenna arrangements on the performance of the MIMO-OFDM system over a UNI channel is also simulated. Both 2×2 and 4×4 MIMO-OFDM systems are considered. The BER performance of this system is shown in Fig. 35. In the case of 2×2 system, the degradation introduced by channel estimation is 0.65dB while that for the 4×4 system is 0.60dB. Moreover the performance of the 4×4 system is better than the 2×2 system both with and without channel estimation. As N_R gets larger, receiver diversity gain compensates for the performance degradation caused by channel estimation error and makes the system less vulnerable to the time-varying channel.

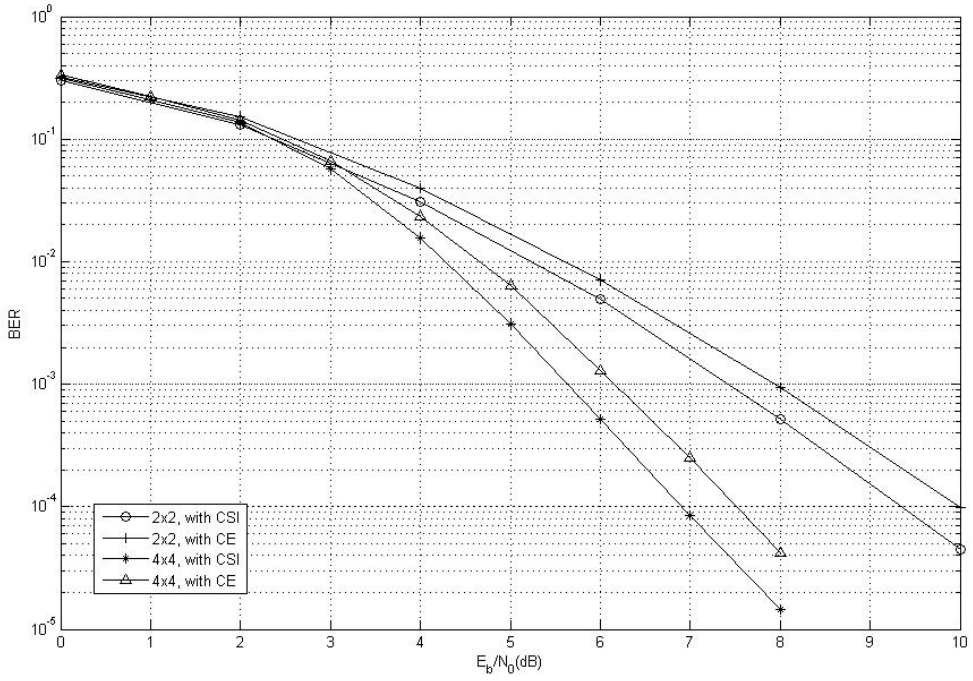


Fig. 35: Comparisons of 2×2 and 4×4 Convolutional coded MIMO-OFDM systems with PACE for data transmitted over a UNI channel with $T_d = 1.02\mu s$, $K = 4$

Investigation of effects of different PDPs

The effects of the different environments on the performance of the 4×4 MIMO-OFDM system are also simulated. Two types of PDPs are considered, UNI and EXP. The value of K is maintained at 4. The result of this comparison is shown in Fig. 36. Firstly, it can be seen that in the UNI channel, the degradation incurred by the channel estimation method is 0.60dB while it is 0.70dB in the EXP channel. Secondly, it can be seen that the performance of the system with resampling is only very slightly better than the proposed system. This is because the value of K used is already very small therefore the benefits of performing the additional resampling step in between every two pilot symbols are very minimal.

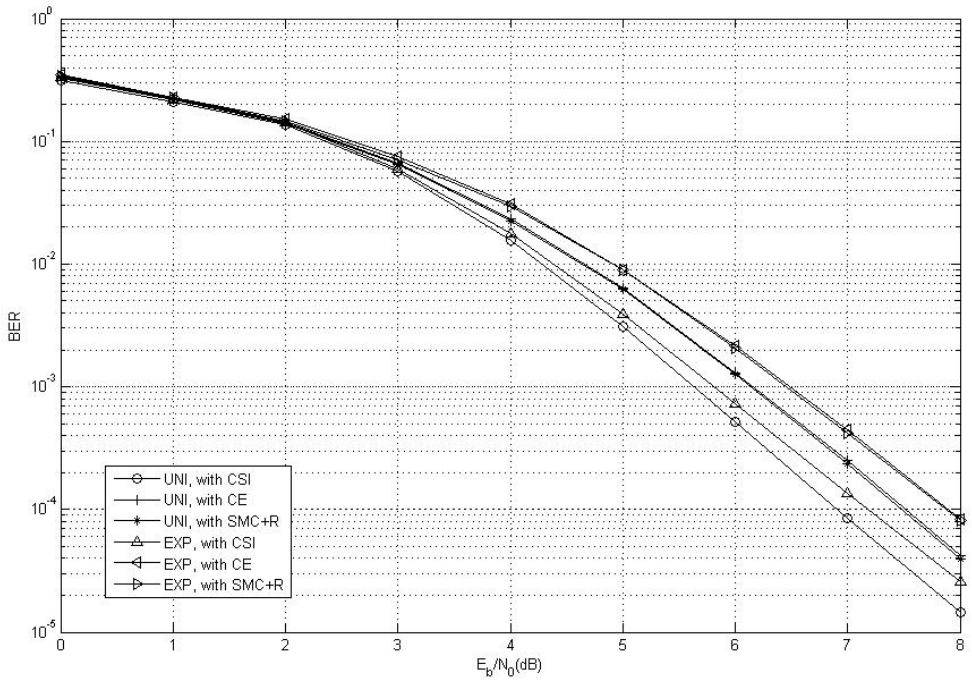


Fig. 36: Performance of a 4×4 Convolutional coded MIMO-OFDM system with PACE for data transmitted over a UNI channel with $T_d = 1.02\mu s$ and EXP channel with $T_d = 0.814\mu s$, and $K = 4$

Investigation of effects of various antenna arrangements for LDPC coded system

Lastly, LDPC coded MIMO-OFDM system over the UNI channel is investigated. Two different antenna arrangements are considered, namely, 2×2 and 4×4. The value of K is maintained at 4. A rate ½ LDPC code of length 1404 is employed. The row weights of the parity check matrix is kept as uniform as possible while the column weight is fixed at 3. It is ensured that the parity check matrix does not have cycles of 4. The SPA [34] that is used within the SISO decoder iterates a maximum of 100 cycles while the number of iterations between the SMC detector and the SISO decoder is set at 5. No interleaver is used. The simulation results are shown in Fig. 37. It is observed that similar to the convolutional coded case, significant improvement in BER is achieved with the increased number of receive antennas. In addition, the degradation caused by channel estimation is 0.60dB for the 2×2 system while it is 0.42dB for the 4×4 system. Moreover, it is also seen that the proposed algorithm performs very closely to the system with resampling.

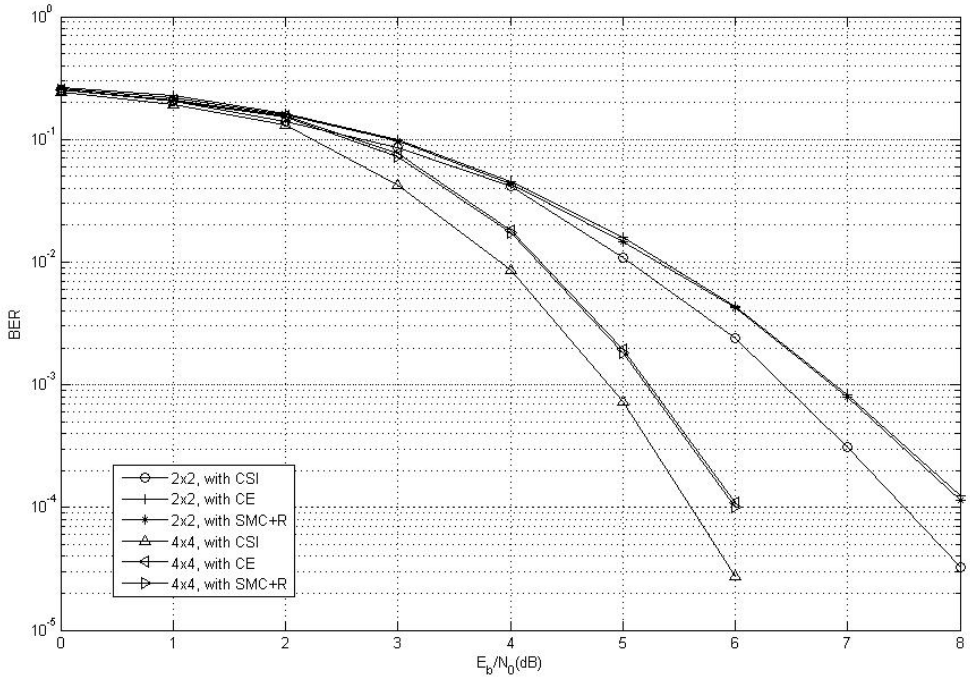


Fig. 37: Comparisons of various antenna arrangements for LDPC coded MIMO-OFDM system with PACE for data transmitted over a UNI channel with $T_d = 1.02\mu s$, $K = 4$, and 5 turbo iterations

5.4 Conclusions

Channel estimation is a vital task in MIMO-OFDM systems. In this section, the performances of the non-resampling SMC iterative receiver with channel estimation under different scenarios are investigated. Instead of employing blind estimation, better estimation is obtained by making use of the termination states as pilot symbols in the PACE method employed here.

It is seen that the performance of the channel estimation algorithm is related to the value of the termination period used where better estimation is possible when the termination period is smaller. Moreover, when $N_T = N_R$, a larger N_R would give

a much better performance as the gains brought about by the increased in receiver diversity compensate for the degradation caused by the increased in ICI due to the increase in the number of transmit antennas. The effect of the channel estimation error incurred is also very slightly milder for the case of 4×4 system than the 2×2 system. It is also observed that the channel estimation algorithm performs better in the UNI channel than EXP channel and there are negligible differences whether resampling is employed. The same trend is seen when the performance of the LDPC coded MIMO-OFDM system with channel estimation is simulated.

Therefore, the overheads caused by periodic termination can be put to good use for good channel estimation performance while at the same time, the proposed algorithm has avoided the computationally intensive resampling step and making parallel processing possible.

CHAPTER 6

CONCLUSIONS

Resampling is an inevitable step when performing SIS while employing SMC methods. Unfortunately, resampling is computationally intensive and a lot of work has been done to address this issue [70-72]. On top of this, resampling causes problems such as making parallel processing impossible and the act of selecting the same large weight particle many times leads to a loss of diversity of the resulting trajectories [30]. In order to circumvent all these issues, an algorithm that skips the resampling step is introduced and its performance simulated. The essence of the algorithm is to periodically insert known termination states into the differential phase trellis such that the termination period is kept short and hence the effects of degeneracy can be kept in check, even though no resampling is performed and at the same time, the diversity of the trajectories is ensured.

As SMC detector is able to take in the soft a priori symbol probabilities from the output of the SISO MAP channel decoder and it is able to generate the a posteriori symbol probabilities, it is very suited to be the first stage of an iterative receiver. An iterative receiver is preferred because it is able to approach the optimum performance with increasing number of iterations.

In chapter 4, the proposed non-resampling SMC iterative receiver is assumed to have perfect knowledge of the CSI and while computing the predictive state

distribution, the multivariant Gaussian distribution involved can be reduced to the product of N_R 1-D Gaussian likelihood functions thus reducing the complexity. Importantly, it has been shown through simulations that the proposed receiver performs very closely to the system with resampling employed, and with a lesser complexity. The results also show that for a system with $N_T = N_R$ antennas, a larger number of receive antennas will lead to the benefits of receiver diversity overweighting the degradation caused by the increased in transmit antennas. The effects of different termination periods, PDPs and channel codes have also been investigated and the performance of the system has been shown to be promising in all cases.

Even though periodic termination results in additional overheads, these overheads can be put to good use by employing them as pilot symbols in PACE. It is well known that PACE gives better performance than blind estimation. In chapter 5, the system with PACE is simulated for different scenarios. The JPG introduced by Auer [49-50] is used and it is shown that the degradation caused by the channel estimation algorithm is only a mere 0.60dB in a UNI channel for a 4×4 system. Similarly, a larger number of receive antennas leads to significant BER improvement. Simulations performed to investigate the effects of different termination periods, PDPs and channel codes have all shown to be promising. This suggests that the proposed non-resampling SMC iterative receiver with channel estimation is a possible way to skip resampling and enabling parallel processing at

a negligible performance tradeoff. Further reduction in complexity can be addressed in future work.

BIBLIOGRAPHY

I received my Bachelor of Engineering degree from the National University of Singapore in 2002. After working for three years as a research engineer, I decided to pursue a Master of Engineering degree. My research interests include OFDM, SMC and coding.

REFERENCES

[1] R. W. Chang, "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission", *Bell System Technical Journal*, vol. 45, pp. 1775-1796, Dec 1966.

[2] B. R. Saltzberg, "Performance of an Efficient Parallel Data Transmission System," *IEEE Trans. on Commun.*, vol. 15, no. 6, pp. 805-811, Dec 1967.

[3] S. B. Weinstein and P. M. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE. Trans. on Commun.*, vol. 19, no. 5, pp. 628-634, Oct 1971.

[4] L. J. Climini, Jr., "Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing," *IEEE Trans. on Commun.*, vol. COM 33, no. 7, pp. 665-675, Jul 1985.

[5] R. N., G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, "New high-rate wireless LAN standards," *IEEE Commun. Mag.*, pp. 82-88, Dec 1999.

[6] Local and Metropolitan Area Networks Part 16, Air Interface for Fixed Broadband Wireless Access Systems, *IEEE Standard IEEE 802.16a*.

[7] Marc Engels (Ed), Wireless OFDM Systems. Boston: Kluwer Academic Publishers, 2002.

[8] G. L. Stuber, J. R. Barry, S. W. McLaughin, Y. Li, M. A. Ingram and T. G. Pratt, "Broadband MIMO-OFDM Wireless Communications," *Proc. IEEE*, vol. 92, pp. 271-294, Feb 2004.

[9] E. Telatar, “Capacity of Multi-antenna Gaussian Channels,” *AT&T-Bell Labs Internal Tech. Memo.*, Jun 1995.

[10] G.J. Foschini and M. J. Gans, “On limits of Wireless Communications in a Fading Environment when Using Multiple Antennas,” *Wireless Personal Commun.*, vol. 6, no. 3, pp. 311-335, 1998.

[11] A. V. Zelst, R. V. Nee and G. A. Awater, “Space Division Multiplexing (SDM) for OFDM systems,” *Proc. IEEE Vehicular Technology Conf.*, vol. 2, pp.1070- 1074, May 2000.

[12] B. A. Bjerke and J. G. Proakis, “Multiple-Antenna Diversity Techniques for Transmission over Fading Channels,” *Wireless Commun. and Networking Conf.*, vol. 3, pp. 1038-1042, Sep 1999.

[13] R. Prasad, OFDM for Wireless Communications Systems, Boston: Artech House, 2004.

[14] T. S. Rappaport, Wireless Communications, Upper Saddle River, N.J. : Prentice Hall PTR, 1996.

[15] A. Doucet, “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering,” *Statist. and Comput.*, vol. 10, pp. 197-208, Jul 2000.

[16] R. Chen and J. Liu, “Mixture Kalman filters,” *J. R. Statist. Soc. B*, vol. 62, no. 3, pp. 493–508, 2000.

[17] A. Kong, J. Liu, and W. Wong, “Sequential imputations and Bayesian missing data problems,” *J. Amer. Statist. Assoc.*, vol. 89, no. 425, pp. 278–288, Mar 1994.

[18] J. Liu and R. Chen, “Blind Deconvolution via Sequential Imputations,” *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 567-576, Jun 1995.

[19] J. Liu and R. Chen, “Sequential Monte carlo methods for Dynamic Systems,” *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032-1044, Sep 1998.

[20] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo and J. Miguez, “Particle filtering,” *IEEE Signal Processing Magazine*, vol. 20, pp. 19-38, Sep 2003.

[21] N. J. Gordon, D. J. Salmond and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proc F Radar & Signal Proc*, vol. 140, pp. 107-113, Apr 1993.

[22] D. Guo and X. Wang, “Blind Detection in MIMO Systems via Sequential Monte Carlo,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 464–473, Apr 2003.

[23] Z. Yang and X. Wang, “A Sequential Monte Carlo Blind Receiver for OFDM Systems in Frequency-Selective Fading Channels,” *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 271-280, Feb 2002.

[24] Y. Su, X. Zhang and X. Zhu, “A Low Complexity Sequential Monte Carlo Algorithm for Blind Detection in MIMO Systems,” *IEEE Trans. Sig. Proc.*, vol. 54, no. 7, pp. 2485-2496, Jul 2006.

[25] B. Dong and X. Wang, “A New Class of Soft MIMO Demodulation Algorithms,” *IEEE Trans. Sig. Proc.*, vol. 51, no. 11, pp.2752–2763, Nov 2003.

[26] C. J. Bordin Jr. and L. A. Baccala, "Joint Blind Equalization and Decoding," *Proc. 4th IEEE Int. Symp. on Sig. Proc. and Info. Tech 2004*, pp. 1-4, Dec 2004.

[27] C. J. Bordin Jr. and L. A. Baccala, "Particle filter algorithms for joint blind equalization/decoding of convolutionally coded signals," *ICASSP 2005*, vol. 3, pp. 497-500, Mar 2005.

[28] C. J. Bordin Jr. and L. A. Baccala, "Deterministic Particle Filters for Joint Blind Equalization and Decoding on Frequency Selective Channels," *IEEE/SP 13th Workshop on Statist. Sig. Proc. 2005*, pp. 375-380, Jul 2005.

[29] E. Punskaya, C. Andrieu, A. Doucet and W. J. Fitzgerald, "Particle Filtering for Multiuser Detection in Fading CDMA Channels," *Proc. IEEE Signal Proc. Workshop*, pp. 38-41, Aug 2001.

[30] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online/Non-Gaussian Bayesian Tracking," *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp.174-188, Feb 2002.

[31] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit Error-Correction Coding and Decoding: Turbo Codes," *IEEE Int. Conf. on Commun. (ICC 93)*, vol. 2, pp. 1064-1070, May 1993.

[32] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo Codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261-1271, Oct 1996.

[33] X. Wang and H. V. Poor, “Iterative (turbo) soft interference cancellation and decoding for coded CDMA,” *IEEE Trans. Commun.*, vol. 47, pp. 1046–1061, Jul 1999.

[34] S. Lin and D. J. Costello, Error Control Coding, Upper Saddle River, N.J. : Pearson-Prentice Hall, 2ed, 2004.

[35] H. V. Poor, “Iterative Multiuser Detection,” *IEEE Sig.Proc. Mag.*, vol. 21, pp. 81-88, Jan 2004.

[36] J. Hagenauer, “The Turbo Principle,” *Proc. Int. Symp. on Turbo-Codes*, Brest, France, pp. 1-11, Sep. 1997.

[37] J. K. Cavers, “An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels,” *IEEE Trans. Veh. Technol.*, vol. 40, pp. 686-693, Nov 1991.

[38] P. Hoeher, “TCM on Frequency Selective Land-Mobile Radio Channels,” *Proc. 5th Tirrenia Int. Workshop on Dig. Commun.*, Tirrenia, Italy, pp. 317-328, Sep 1991.

[39] J. Beek, O. Edfors, M. Sandell, S. K. Wilson and P. O. Borjesson, “On Channel Estimation in OFDM Systems,” *IEEE 45th VTC*, vol. 2, pp. 815-819, Jul 1995.

[40] R. Nilsson, O. Edfors, M. Sandell and P. Borjesson, “An Analysis of Two-Dimensional Pilot-Symbol Assisted Modulation for OFDM,” *Proc. IEEE Int. Conf. Personal Wireless Commun. (ICPWC ’97)*, Mumbai, India, pp. 71-74, 1997.

[41] O. Edfors, M. Sandell, J. J. Beek, S. K. Wilson and P. O. Borjesson, "OFDM Channel Estimation by Singular Value Decomposition," *IEEE Trans. Commun.*, vol. 46, pp. 931-939, Jul 1998.

[42] P. Hoeher, S. Kaiser and P. Robertson, "Two-Dimensional Pilot-Symbol-Aided Channel Estimation by Wiener Filtering," *Proc. 1997 IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, Munich, Germany, pp. 1845-4848, Apr 1997.

[43] P. Hoeher, S. Kaiser and P. Robertson, "Pilot-Symbol-Aided Channel Estimation in Time and Frequency," *Proc. Commun. Theory Mini-Conf. (CTMC) within IEEE Global Telecommun. Conf. (GLOBECOM'97)*, Phoenix, USA, pp. 90-96, 1997.

[44] Y. Li, N. Seshadri and S. Ariyavisitakul, "Channel Estimation for OFDM Systems with Transmitter Diversity in Mobile Wireless Channels," *IEEE J. Sel. Areas Commun.*, vol. 17, pp. 461-470, Mar 1999.

[45] Y. Li, "Pilot-Symbol-Aided Channel Estimation for OFDM in Wireless Systems," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1207-1215, Jul 2000.

[46] Y. Li, "Simplified Channel Estimation for OFDM Systems with Multiple Transmit Antennas," *IEEE Trans. Wireless Commun.*, vol. 1, pp. 67-75, Jan 2002.

[47] I. Tolochko and M. Faulkner, "Real Time LMMSE Channel Estimation for Wireless OFDM Systems with Transmitter Diversity," *IEEE 56th VTC*, vol. 3, pp. 1555-1559, Sep 2002.

[48] G. Auer, “Channel Estimation in Two Dimensions for OFDM Systems with Multiple Transmit Antennas,” *GlobeCom ’03*, vol. 1, pp. 322-326, Dec 2003.

[49] G. Auer, “Analysis of Pilot-Symbol Aided Channel Estimation for OFDM Systems with Multiple Transmit Antennas,” *ICC 2004*, vol. 6, pp. 3221-3225, Jun 2004.

[50] G. Auer, “Pilot-Symbol Aided Channel Estimation by Wiener Filtering for OFDM Systems with Multiple Transmit Antennas,” *3G Mobile Commun. Technol. 2004, 5th IEEE Int. Conf.*, pp. 6-10, 2004.

[51] J. G. Proakis, Digital Communications, 4th Ed., McGraw-Hill 2001.

[52] A. Papoulis and S. U. Pillai, Probability, Random Variables and Stochastic Processes, 4th Ed., McGraw-Hill, 2002.

[53] W. C. Jakes, Microwave Mobile Communications, New York, Wiley 1974.

[54] A. Peled and A. Ruiz, “Frequency domain data transmission using reduced computational complexity algorithms,” *Proc. IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, ICASSP ’80, vol. 5, pp. 964-967, Apr 1980.

[55] Z. Wang and G. B. Giannakis, “Wireless Multicarrier Communications,” *IEEE Sig. Proc. Mag.*, vol. 17, pp. 29-48, May 2000.

[56] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,” *IEEE Trans. Inform. Theory*, vol. 13, pp. 260-269, Apr 1967.

[57] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate (Corresp.),” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, Mar 1974.

[58] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Englewood Cliffs, N.J.: Prentice Hall, 1995.

[59] R. G. Gallager, “Low-density Parity-check Codes,” *IEEE Trans. Inform. Theory*, vol.8, pp. 21-28, Jan 1962.

[60] M. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, “Improved Low Density Parity Check Codes using Irregular Graphs and Belief Propagation,” *Proc. IEEE Int. Symp. Inform. Theory*, pp. 117, Aug 1998.

[61] M. C. Davey and D. J. C. MacKay, “Low Density Parity Check Codes over GF(q),” *Inform. Theory Workshop*, pp. 70-71, Jun 1998.

[62] T. Richardson, A. Shokrollahi and R. Urbanke, “Design of a Provably Good Low-Density Parity-Check Codes,” *Proc. IEEE Int. Symp., Inform. Theory*, pp. 199, Jun 2000.

[63] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Serial Concatenation of Interleaved Codes: Performance Analysis, Design and Iterative Decoding,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 909-926, May 1998.

[64] L. L. Scharf, Statistical Signal Processing, Addison-Wesley, 1991.

[65] J. Moon, H. Jin, T. Jeon and S.K. Lee, “Channel Estimation for MIMO-OFDM Systems Employing Spatial Multiplexing,” *IEEE Veh. Tech. Conf. 2004, VTC2004*, vol. 5, pp. 3649-3654, Sep 2004.

[66] S. Kang and J. S. Lehnert, “Channel Estimation in MIMO OFDM Systems with Sparse Pilot Tones,” IEEE *Veh. Tech. Conf. 2005, VTC2005*, vol. 1, pp. 221-225, May 2005.

[67] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *J. of Basic Engineering*, vol. 82, pp. 35-45, 1960.

[68] A. Gelb, Applied Optimal Estimation, Boston: MIT Press, 1974.

[69] J. M. Hammersley and K. W. Morton, “Poor Man’s Monte Carlo,” *J. of Royal Statistical Society B (Methodological)*, vol. 16, pp. 23-28, 1954.

[70] S. Hong, S. Chin and S. Magesh, “A flexible resampling mechanism for parallel particle filters,” Int. *Symp. VLSI Tech. Systems and Applications*, pp. 288-291, 2003.

[71] M. Bolic, P. M. Djuric and S. Hong, “New resampling algorithms for particle filters,” *IEEE Int. Conf. Acoustics, Speech and Signal Proc. (ICASSP ’03)*, vol. 2, pp. 589-592, Apr 2003.

[72] M. Bolic, P. M. Djuric and S. Hong, “Resampling algorithms for particle filters: a computational complexity perspective,” *EURASIP Journal on Applied Sig. Proc.*, vol. 15, pp. 2267-2277, 2004.

[73] Y. R. Zheng and C. Xiao, “Simulation models with correct statistical properties for Rayleigh fading channels,” *IEEE Trans. Commun.*, vol. 51, pp. 920-928, Jun 2003.

[74] L. K. Bay, N. Arumugam and H. K. Garg, “Iterative Receiver Design for MIMO-OFDM Systems via Sequential Monte Carlo (SMC) Techniques,” submitted to International Conference on Communications, 2008.

[75] R. Nee, A. Zelst and G. Awater, “Maximum Likelihood Decoding in a Space Division Multiplexing System,” *IEEE VTC2000*, Japan, vol. 1, pp.6-10, May 2000.