BLIND CHANNEL ESTIMATION BASED ON THE LLOYD-MAX ALGORITHM IN NARROWBAND FADING CHANNELS AND JAMMING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR DİZDAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2011

Approval of the thesis:

BLIND CHANNEL ESTIMATION BASED ON THE LLOYD-MAX ALGORITHM IN NARROWBAND FADING CHANNELS AND JAMMING

submitted by **ONUR DİZDAR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ________________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Ali Özgür Yılmaz ________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Yalçın Tanık
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Mete Severcan
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Dept., METU ________________

M.Sc. Ertuğrul Kolağasıoğlu
Senior Design Leader, ASELSAN ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ONUR DİZDAR

Signature :

ABSTRACT

BLIND CHANNEL ESTIMATION BASED ON THE LLOYD-MAX ALGORITHM IN NARROWBAND FADING CHANNELS AND JAMMING

Dizdar, Onur
M.Sc., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Ali Özgür Yılmaz

June 2011, 65 pages

In wireless communications, knowledge of the channel coefficients is required for coherent demodulation. In this thesis, a blind channel estimation method based on the Lloyd-Max algorithm is proposed for single-tap fading channels. The algorithm estimates the constellation points for the received signal using an iterative least squares approach. The algorithm is investigated for fast-frequency hopping systems with small block lengths and operating under partial-band and partial-time jamming for both detecting the jammer and estimating the channel. The performance of the Lloyd-Max channel estimation algorithm is compared to the performance of pilot-based channel estimation algorithms which also use the least squares approach and non-coherent demodulation and decoding.

Keywords: Lloyd-Max, blind channel estimation, jamming detection, differential encoding, serially concatenated convolutional code

ÖZ

DARBANT SÖNÜMLEMELİ KANAL VE KARIŞTICI ALTINDA LLOYD-MAX ALGORİTMASINA DAYALI KÖR KANAL KESTİRİMİ

Dizdar, Onur
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Ali Özgür Yılmaz

Haziran 2011, 65 sayfa

Kablosuz haberleşmede, faz uyumlu çözme işlemi için kanal katsayılarının bilgisi gereklidir. Bu tezde, tek tapalı sönümlemeli kanallar için Lloyd-Max algoritmasına dayalı kör bir kanal kestirim metodu önerilmiştir. Algoritma, tekrarlı bir en düşük kareler yaklaşımı kullanarak alınan sinyal için işaret kümesi noktalarını bulur. Algoritma, hızlı frekans atlayan küçük blok uzunluğuna sahip ve parçalı-bantlı ve parçalı-zamanlı karıştırıcılar altında çalışan sistemler işin hem karıştırıcıyı tespit etmek hem de kanalı kestirmek amacıyla incelenmiştir. Lloyd-Max kanal kestirim algoritmasının performansı pilot-temelli ve aynı zamanda en düşük kareler yaklaşımını kullanan kanal kestirim algoritmaları ve faz uyumlu olmayan demodülasyon ve kod çözme performanslarıyla karşılaştırılmıştır.

Anahtar Kelimeler: Lloyd-Max, kör kanal kestirimi, karıştırıcı tesbiti, farksal kodlama, seri bağlanmış kodlar

To my family and fiancee

ACKNOWLEDGEMENTS

First, I would like to thank my thesis advisor and mentor Assoc. Prof. Dr. Ali Özgür Yılmaz, whom I have been working with since my undergraduate years. His patience, support and encouragement motivated me towards my master's degree. His knowledge and dedication guided me during my studies. I am truly grateful to have had the chance of working with him throughout my education career.

I would like to thank my lovely fiancee Seçil for all her support and encouragement. She always believed in me and was always there for me in my times of need. Her support made it possible for me to overcome the stressful days and complete this thesis.

This thesis would not have been possible without my family. I owe my deepest thanks to them for bringing me to these days, giving me all the love and support for my entire life. It was their sacrifice and devotion that made me what I am today. I would like to thank my parents Güler and Yüksel, my aunts Nurten and Gülten, and my uncle Saim for all they have done for me. I owe my special thanks to my cousin Hüseyin and his wife Banu for their mentorship, encouragement and for helping me to make my career choice in the first place.

I offer my deepest gratitude to my colleague Ertuğrul Kolagasıoğlu. His support, attitude, and teachings were beyond what I could expect or imagine and it was a privilege working with him. I thank deeply my colleague Güven, whom I started my career with and shared much throughout my journey to the end. I enjoyed working with him, and I hope to get the same chance again for the rest of my career. I also offer my deepest thanks to my colleagues Sinan, Tolga, Barış, Füruzan and Özlem for all their support in my working career and thesis, and for being such wonderful persons.

I would like to show my gratitude to ASELSAN for the support in my graduate education. I would also like to acknowledge TÜBİTAK BİDEB for providing me the 2210 National Scholarship Programme for M.Sc. Students scholarship for two years.

TABLE OF CONTENTS

CHAPTERS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In wireless communications, an information signal is affected by different types of disturbances from the moment it is transmitted into the air. These disturbances are both environmentally induced and human-made. Generally, there are two types of disturbances on a wireless signal. Many types of enviromental effects, such as reflections of the waves, shadowing due to objects in the environment, attenuation and delays combine to cause distortions on the transmitted electromagnetic waves due to propagation. On top of these distortions, random fluctuations caused by imperfections in electrical circuits of communicating devices are added to the signal, which is called the *noise*. The sum of these environmental distortions is referred to as *channel* effects. A communication system has to provide reliable communication under the effect of these disturbances.

A widely used approach to mitigate the distortion on the transmitted signal is to detect the characteristics of the main sources of distortions. This is accomplished by estimating the noise statistics and, more importantly, channel parameters at the receiver. Once a receiver obtains an estimate of the channel, it uses this estimate for extracting the transmitted information from the received signal. The quality of the channel estimate has a tremendous effect on the quality of communication services. In order to obtain low data error rates, the channel estimate used to resolve the transmitted data should contain small errors.

Channel estimation methods and the effects of the channel estimates on the system performance are widely investigated topics in literature. The channel estimator is a major component in receivers. The investigated estimation methods can be classified into two groups: the methods using no previous knowledge of the received information symbols, and the methods using the received symbols called *pilot* symbols that are known both at the transmitter and receiver. The methods in the first group are referred to as *blind* methods, whereas the second group is referred to as *pilot-based* or *data-aided* methods.

The main disadvantage of pilot-based estimators is the reduced power and bandwidth efficiency. The loss of power and bandwidth may be critical for systems with limited power or using small blocks of symbols for communication. Moreover, systems using pilot-based estimators are vulnerable to hostile systems, such as jammers, since large distortions caused by jammers on pilot symbols can disturb the communication severely. Pilots may not be preferred also in order to avoid the detection of transmitted signals by algorithms tracking periodicity.

In [4], a pilot-based channel estimator was derived based on the least-squares solution, which minimizes the sum of errors between the pilots and received distorted versions of pilots. The least-squares approach is the simplified form of a *maximum-likelihood* (ML) estimation [1] derived for estimating the channel using known symbols. The LS algorithm is widely used in wireless communications because of its performance and simplicity.

Beyond the environmental effects, wireless systems may be the target of jammers having the intention of disturbing the communication. Detection of jamming is important for a system since the negative effects of the jammer has to be taken into account during the processing of the received signal. Various methods for detecting different types of jamming schemes have been proposed in [9], [10], [11], [13], and [23]. In [9], soft decoder outputs are used to estimate interference variance with a suboptimal approach to ML estimation. Pilots are used to enhance the estimation. In [10], SCCC with differential modulation is used. The channel is AWGN channel with random phase. 8 different trellis are calculated at the decoder, each using a different quantized value for the channel phase. A threshold test using forward metrics is employed to detect partial-band jamming. No pilot symbols except one reference symbol are used since there is no need for channel estimation. In [11], scaling parameters are calculated for jammed dwells depending on the demodulator outputs. In [13], an iterative EM algorithm is derived for the estimation of channel and jamming parameters. The estimation algorithm have both pilot-based and blind versions. The phase ambiguity problem in blind version is solved for BPSK modulation. Jammed symbols are detected using BCJR algorithm. It is shown that pilot-based and blind algorithms perform close to each other in AWGN channel. In [23], binary pilot symbols and their hard decisions are used for jamming detection. The algorithms in [10], [11], and [23] propose scaling for the jammed dwells before they are decoded.

In this thesis, a blind channel estimation algorithm is developed for tactical radios with fast-frequency hopping and the capability of operating under jamming. The estimation is based on the Lloyd-Max algorithm, which is an iterative quantization algorithm originally developed for quantizing analog voltage levels of PCM signals [5]. The algorithm uses the LS criterion in its attempt to determine the best quantization points that yield the minimum squared quantization errors. The Lloyd-Max algorithm does not use any previously known information about the signal for the quantization procedure. The channel gain and noise variance are estimated by this algorithm to be used for processing the received signal. In this thesis, the performance of Lloyd-Max based estimation is compared with the performance of the pilot-based LS algorithm for different types of receivers and under the effect of jamming. The outline of the thesis is as follows:

In Chapter 2, the system model is introduced. Mathematical derivations for the Lloyd-Max algorithm, LS channel estimator, an extension of LS estimator are presented. The receiver structures that will be used with the estimators are described. In Chapter 3, the channel estimator based on the Lloyd-Max algorithm is explained and its performance is investigated with the other two LS-based estimators. In Chapter 4, jamming detection algorithms and the performances of the estimators are given under partial-time and partial-band jammers. The conclusion is given in Chapter 5.

CHAPTER 2

BACKGROUND

2.1 Coherent and Non-Coherent Demodulation in Communication Systems

In wireless communications, communication is established between a transmitter and a receiver through the air. The information to be sent is generated after various operations (encoding, modulation, pulse shaping etc.) and given to the air by the transmitter. The transmitted signal is affected by many factors as it travels through the air and arrives at the receiver, such as obstacles between communicating devices, reflections from obstacles, thermal noise on electrical components, and so on. These effects distort the received signal, causing difficulties in obtaining the transmitted information. In order to extract the transmitted information from the received signal, the signal and the distortions are represented using mathematical models. In digital communications, the received signal can be modeled using discrete-time signals and systems [2]. This is enabled by proper filtering and sampling operations at the receiver on the signals induced by the received electromagnetic waves. The received signal in discrete time domain is represented by

$$y_k^i = \sum_{i=0}^{L-1} h_i a_{k-i} + n_k \tag{2.1}$$

where y_k is the sample taken from the filtered received signal at the time instance $k{\cdot}T_s$, T_s is the duration of the information symbol and also the sampling period, and a_k are the information symbols. The information symbols take values from finite symbol sets depending on the type of modulation. The symbols a_k can have real or complex values, e.g., $a_k \in \left\{+\sqrt{E_s},\ -\sqrt{E_s}\right\}$ for Binary Phase-Shift Keying (BPSK) modulation or $a_k \in \left\{+\sqrt{E_s},\ +j\sqrt{E_s},\ -\sqrt{E_s},\ -j\sqrt{E_s}\right\}$ for Quadrature Phase-Shift Keying (QPSK) modulation, where E_s represents the energy used by the transmitter for one symbol. Each symbol represents $log_2 M$ bits, where M is the number of elements in the symbol sets.

The channel gains at different delays are denoted by h_i which reflects the effects of attenuation, obstacles between the communicating antennas, propagation delays, and reflections of the signal with delays small enough with respect to symbol duration for a delay of $k{\cdot}T_s$ seconds. If $L > 1$, the channel is called a *multipath channel*. The impulse response of a multipath channel is long enough so that when echoes of the transmitted signal are summed up at the receiver antenna, different symbols are added on top of each other, causing intersymbol interference. It should be noted that in (2.1), the channel taps are time-invariant, i.e., they are constant for all received samples. In this thesis, only the channels with $L = 1$, which are also called *narrowband* channels, are taken into consideration. Using this assumption, (2.1) becomes

$$y_k = ha_k + n_k \tag{2.2}$$

where h is modeled as a zero-mean circularly symmetric complex Gaussian (ZMCSCG) random variable with variance 1, n_k represents the additive thermal noise at the receiver with a flat frequency spectrum (white). The thermal noise is also a ZMCSCG random variable with variance N_0. In order to quantify the strength of distortions on the signal, parameters called *instantaneous signal-to-noise power ratio* and *average signal-to-noise power ratio* are defined. When there is a fading channel, $\frac{|h|^2 E_s}{N_0}$ is the instantaneous SNR and the average SNR is defined as

$$\gamma_s = \frac{E\left\{|h|^2\right\} E_s}{N_0} = \frac{E_s}{N_0}. \tag{2.3}$$

The demodulation of the signal generally requires operations which are the opposite of those utilized for modulation. The demodulation operation is divided into two classes: coherent and non-coherent demodulation. The difference of coherent and non-coherent demodulation is the use of channel phase in the demodulation procedure. In coherent demodulation, the channel phase has to be known and used for correcting the phases of the received samples. This correction is important when the information is carried in the phase of the modulation symbols, e.g., M-PSK. M-PSK modulation uses constant amplitude symbols with different phases to transmit the information. In the demodulation procedure, errors occur when the distortions change the phase of the received sample so that the sample appears to have the phase of a different symbol. Therefore, the distortion inflicted by the channel to the phase of the symbol is corrected or taken into account in coherent demodulation. On the other hand, in non-coherent demodulation, the phase of the channel is not necessary for obtaining the transmitted information from received samples. It is used for modulation techniques where the information is not carried in the phase of the symbols, but rather the amplitude, frequency or the phase difference between the symbols.

2.2 Partial-Time and Partial-Band Jamming

Besides the additive thermal noise and the channel, a wireless signal may also be corrupted on purpose by hostile systems, namely jammers. A jammer is a device that corrupts a part of the frequency spectrum in different ways in order to distrupt communication. In practice, there are many types of jammers trying to disturb the communication system using different methods. Partial-time and partial-band jammers are two types of jammers that are widely used in literature, because of their wide use in practice and easily handled mathematical models.

Against a possible threat of jammers, a system may take some precautions in order to protect its communication. Frequency hopping is one of these precautions and widely utilized especially in military communications. In frequency hopping, the carrier frequency used for transmitting the signal is changed according to a predetermined hopping sequence. This reduces the probability of a jammer to detect and jam the signal, and forces it to blind jamming. It also has benefits against channel fading, since fading is a function of the carrier frequency. A block that contains symbols that are sent with the same carrier frequency is called a *dwell*. The length of a dwell, i.e., the number of symbols in a dwell, is a design parameter depending on the requirements of the system. When frequency hopping is used, the signal model defined by (2.2) becomes

$$y_k^i = h_i^i a_k^i + n_k^i \tag{2.4}$$

where i is the index showing the dwell number.

In partial-band jamming [9], the jammer corrupts a part of the frequency spectum continuously. The bandwidth of the corrupted part is expressed as ρ times the whole operational spectrum of the jammer. Therefore, for a random frequency hopping system, the probability that the system hops into a jammed band is ρ. When the system hops into the jammed part of the spectrum, it continues its operation in that part for the duration of a frequency dwell.

Therefore, an entire dwell is jammed with a probability of ρ. The power of the jammer over the whole spectrum is denoted as N_j and the total power in the part of the spectrum where the jammer dwells is N_j/ρ [9], [10], [11], [12]. When there is a partial-band jammer affecting the system, the received signal model defined by (2.4) is modified as

$$y_k^i = h^i a_k^i + n_k^i + v^i w_k^i \tag{2.5}$$

where w_k^i is the jammer signal which is considered to be ZMCSCG with variance $\frac{N_j}{\rho}$, and v^i is the jamming indicator function; it is equal to one if the the dwell is jammed and zero otherwise.

Partial-time jammers use frequency hopping to change the spectrum they aim to disturb. A frequency hopping system can be jammed by a partial-time jammer at any part of a dwell in time domain. Unlike in partial-band jammers, when a frequency dwell is jammed, only some and generally consecutive symbols of the dwell are jammed.

A partial-time jammer is very often modeled by a 2-state Markov Chain [13], as used for Gilbert-Elliot Channels in [14], [15]. The jammer is characterized by state transition probabilities which are shown in Figure 2.1.

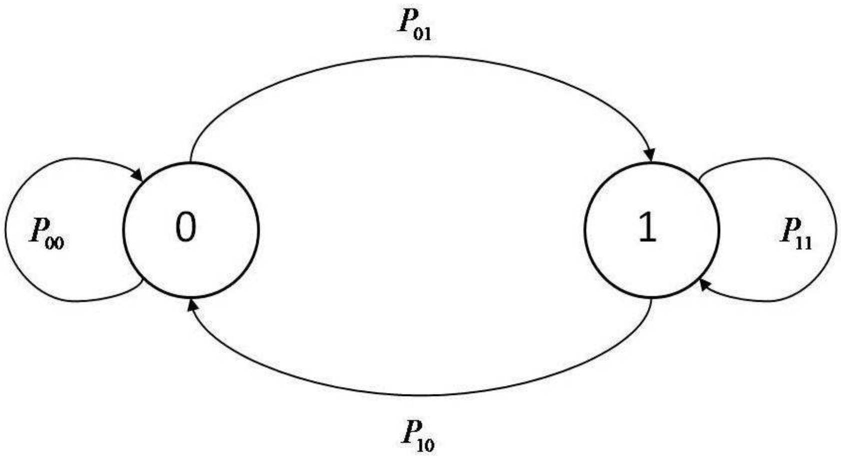

Figure 2.1: Markov model for partial-time jammers

State-0 represents the no-jamming and State-1 represents the jamming states of the Markov chain. The probability of passing from one state to another is denoted by P_{ij}, where i represents the previous state and j represents the current state. The transition matrix is defined by

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}, \tag{2.6}$$

where the discrete-time index is denoted by n, the states of the chain are positive recurrent, i.e.,

$$\sum_{n=0}^{\infty} P_{00}^{n} = \infty \qquad and \qquad \sum_{n=0}^{\infty} P_{11}^{n} = \infty \tag{2.7}$$

and the expected time the process returns to State-k starting from State-k is finite [3]. States are also aperiodic, meaning that given $P_{ii}^{n} = 0$ for n values which are not divisible by period d, $d = 1$ for State-0 and State-1. Therefore, the State-0 and State-1 are said to be ergodic. For irreducible ergodic Markov chains, π_j are defined as

$$\pi_j = lim_{n\to\infty} P_{ij}^{n} \tag{2.8}$$

$$= \sum_{i=0}^{\infty} \pi_i P_{ij} \tag{2.9}$$

where π_j denotes the probability of being at State-j for a long observation period, i.e., $\pi_j^n = P(state\ at\ time\ n\ =\ j)$. The vector $\overline{\pi}^n$ is the vector of the probabilities for each state at time n, which is calculated by

$$\overline{\pi}^n = \overline{\pi}^0 P^n \tag{2.10}$$

If initial probabilities $\overline{\pi}^0$ are chosen such that they satisfy (2.9), then $\overline{\pi}^n = \overline{\pi}^0$, and π_j are called stationary probabilities.

Using the basic information about Markov chains above, a partial-time jammer can be modeled as follows: Define ρ and $E\{T_1\}$ as the probability of a symbol to be jammed and the expected value of the number of consecutive symbols that are jammed in a dwell, respectively [13]. In terms of Markov chain parameters, $\rho = \pi_1$ and

$$E\{T_1\} = \sum_{n=1}^{\infty} n P_{11}^{n-1}(1 - P_{11}) = \frac{1}{1 - P_{11}} \tag{2.11}$$

The signal model under partial-time jamming is defined as

$$y_k^i = h^i a_k^i + n_k^i + v_k^i w_k^i \tag{2.12}$$

v_k^i is the jamming indicator function similar to the one defined in (2.12), with one difference that it changes for every symbol instead of every dwell.

2.3 Channel Estimation Based on the Least Squares Method

The Least-Squares (LS) method is used for solving a set of equations to find the unknown parameters in an overdetermined system, where the number of equations is larger than the number of unknowns. The LS solution for the unknown parameter is the value which yields the least sum of squared errors. A general usage for LS is data fitting, in which for a given number of observations, a curve minimizing the sum of errors between the points on the curve and the observations is found.

In communication systems, the LS algorithm is widely used to estimate the channel parameters affecting the transmitted signals. In LS channel estimation, the channel estimate which minimizes the sum of squared errors is calculated using a number of known symbols (pilot symbols) and the received samples corresponding to these symbols [4].

Considering the system model defined in Section 2.1, (2.1) can also be written in the vector notation as

$$\mathbf{y} = \mathbf{A}\mathbf{h} + \mathbf{n} \tag{2.13}$$

where

$$\mathbf{h} = [h_0\, h_1\ \ldots\ h_{L-1}]^T \tag{2.14}$$

is the channel coefficients vector,

$$\mathbf{y} = [y_0\, y_1\ \ldots\ y_{N-1}]^T\,, \qquad \mathbf{n} = [n_0\, n_1\ \ldots\ n_{N-1}]^T\,, \tag{2.15}$$

are the vector of received symbols and noise samples, respectively,

$$\mathbf{A} = \begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 \\ a_1 & a_0 & 0 & \cdots & 0 \\ a_2 & a_1 & a_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ a_{N-1} & a_{N-2} & & \cdots & a_{N-L} \end{bmatrix} \tag{2.16}$$

is the matrix of known symbols, N is the number of known symbols and $(\cdot)^T$ denotes the transpose operation.

Minimizing $e(\hat{\mathbf{h}}) = ||\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}^T||$ with respect to $\hat{\mathbf{h}}$ yields

$$\hat{\mathbf{h}} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{y} \tag{2.17}$$

where $||\cdot||$ denotes the *norm* operation and $(\cdot)^H$ denotes the Hermitian operation. For single-tap channels, (2.17) becomes

$$\hat{h} = \frac{\mathbf{a}^H\mathbf{y}}{NE_s} \tag{2.18}$$

with $\mathbf{a} = [a_0\ a_1\ \ldots\ a_{N-1}]^T$, which is basicly the average of projections of the known symbols on the received samples.

2.4 The Lloyd-Max Algorithm

The Lloyd-Max algorithm is a quantization algorithm using Least Squares approximation [5]. The algorithm was originally developed to be used for quantization of PCM signals. A quantization process matches given input values to predetermined representation points. It can be considered as a transformation from an infinite alphabet to a finite alphabet. When a signal from an infinite alphabet is represented with a finite alphabet, the signal is distorted. This distortion is called the quantization noise. The algorithm is developed as a solution to the problem of minimizing the quantization errors when an analog signal is pulse-code modulated. For the continous voltage values of the analog signal, the Lloyd-Max algorithm finds the close-to-optimum quantization values to be used in order to minimize the difference between the original and quantized signals.

The algorithm divides the domain of the original signal Q into quantization subsets, and finds a single representation point for each subset. Assume that $\{Q_1, Q_2, \ldots, Q_v\}$ are disjoint subsets of Q, and $\{q_1, q_2, \ldots, q_v\}$ are the representation points, namely *quanta*, for these subsets respectively. Define the function $\gamma(x)$ by

$$\gamma(x) = \begin{cases} 1, & x \in Q_1 \\ 2, & x \in Q_2 \\ & \vdots \\ v, & x \in Q_v \end{cases} \tag{2.19}$$

After the quantization procedure, x is represented by

$$x = \gamma(x) + z(x) \tag{2.20}$$

where $z(x)$ is the quantization noise.

Assuming that $s(t)$ is a stationary random process, the cumulative probability distribution function of $s(t)$ is

$$F_s(x) = P\{s(t) \leq x\}, \quad -\infty < x < \infty \tag{2.21}$$

which is independent of time t since $s(t)$ is a stationary process. By using the probability distribution function, the quantization noise power can be written as

$$N = \int_{-\infty}^{\infty} z^2(x)dF(x) = \sum_{m=1}^{\nu} \int_{Q_m} (q_m - x)^2 dF(x), \quad -\infty < t < \infty. \tag{2.22}$$

The MMSE criterion leads to the fact that the best quanta q_m are found by minimizing N with respect to fixed Q_m, which suggests that

$$q_m = \frac{\int_{Q_m} xdF(x)}{\int_{Q_m} dF(x)}, \quad m = 1, 2, \ldots, \nu. \tag{2.23}$$

One may note that the quantum q_m is the center of mass for Q_m.

The best sets for fixed quanta are found independently from the best quanta calculations. It is assumed that $q_a \neq q_b$ for $a \neq b$ since the sets Q_a and Q_b are disjoint sets for an optimum partitioning. From (2.22), it is observed that the noise magnitude is directly proportional to $(q_m - x)^2$. This implies that, for the minimization of quantization noise, the quantum minimizing $(q_m - x)^2$ should be chosen as the quantized value of x so that

$$Q_a = \left\{x : (q_a - x)^2 < (q_b - x)^2, \ \forall b \neq a\right\}, \quad a = 1, \ldots, \nu \tag{2.24}$$

or equivalently,

$$Q_a = \left\{x : (q_b - q_a)(x - \frac{q_a + q_b}{2}) < 0, \ \forall b \neq a\right\}, \quad a = 1, \ldots, \nu. \tag{2.25}$$

From (2.25), it is observed that the end point of a best set is the arithmetic mean of the quantum of that set and quanta of the neighbouring sets. In other words,

$$
\begin{aligned}
Q_1 &= \{x : -\infty < x \le x_1\} \\
Q_2 &= \{x : x_1 < x \le x_2\} \\
&\vdots \\
Q_{\nu-1} &= \{x : x_{\nu-2} < x \le x_{\nu-1}\} \\
Q_\nu &= \{x : x_{\nu-1} < x \le \infty\}, \qquad (2.26)
\end{aligned}
$$

where

$$
\begin{aligned}
x_1 &= \frac{q_1 + q_2}{2} \\
x_2 &= \frac{q_2 + q_3}{2} \\
&\vdots \\
x_{\nu-1} &= \frac{q_{\nu-1} + q_\nu}{2} \qquad (2.27)
\end{aligned}
$$

and $q_1 < q_2 < \ldots < q_\nu$.

With these results, the algorithm suggests a trial-and-error method to find the optimum quantization scheme:

```
Start with arbitrary initial endpoints $\left\{x_1^i, x_2^i, \ldots, x_\nu^i\right\}$ for the sets $\left\{Q_1^i, Q_2^i, \ldots, Q_\nu^i\right\}$
Calculate the quanta $\{q_1, q_2, \ldots, q_\nu\}$ according to $\left\{x_1^i, x_2^i, \ldots, x_\nu^i\right\}$
REPEAT
    Calculate the endpoints $\{x_1, x_2, \ldots, x_\nu\}$ according to $\{q_1, q_2, \ldots, q_\nu\}$
    Calculate the quanta $\{q_1, q_2, \ldots, q_\nu\}$ according to $\{x_1, x_2, \ldots, x_\nu\}$
UNTIL maximum number of iterations is reached
```

The details of the derivations can be found in [5].

2.5 Decision-Directed Least Squares Channel Estimation

Decision-Directed Least Squares Channel Estimation (DDLS) is an extension to the classical LS channel estimation method. In DDLS, data symbols are also used in channel estimation in addition to pilot symbols. In DDLS, the data symbols are not used to track the changes in the channel. Rather, they are used to enhance the LS estimatiom. Assuming the system is modeled by (2.2), i.e., the channel is single-tap, the channel estimation process is completed in two steps. In the first step, the channel is estimated over the pilots using (2.18). After $\hat{h}_{LS}$ is calculated, hard decisions for received data symbols $\tilde{d}_k$ are made, where d_k are the received data symbols. The hard decisions are obtained by

$$\tilde{d}_k = \underset{\alpha \in A}{\operatorname{argmin}} |d_k - \hat{h}_{LS}\,\alpha|^2, \tag{2.28}$$

where A is the set of constellation points used for M-PSK modulation. For multipath channels, hard decisions can be obtained by equalization or other means of ISI cancellation.

Once hard decisions are available, the second step of the estimation procedure starts. The hard decisions can be utilized as pilot symbols for estimation purposes. Using the hard decisions, DDLS channel estimate is calculated using (2.18) again, this time using both pilots and hard decisions as

$$\hat{h}_{DDLS} = \frac{\mathbf{d}^H \mathbf{y}}{(N+K)E_s} \tag{2.29}$$

where $\mathbf{d} = \begin{bmatrix} a_0 & \ldots & a_{N-1} & \tilde{d}_0 & \ldots & \tilde{d}_{K-1} \end{bmatrix}^T$ and K is the length of data symbols.

2.6 Serially Concatenated Convolutional Coding

The class of serially concatenated convolutional codes (SCCC) is a family of iteratively decoded channel codes. An SCCC encoder consists of two convolutional encoders connected serially with an interleaver between them. Figures 2.2 and 2.3 show the encoder and decoder structures for SCCC, respectively.

The blocks denoted by $\prod$ and $\prod^{-1}$ are the interleaver and deinterleaver, respectively. The subscript π means that the symbols are interleaved and the subscript π^{-1} means that the symbols are deinterleaved. Neglecting the possible termination bits for the component encoders, the

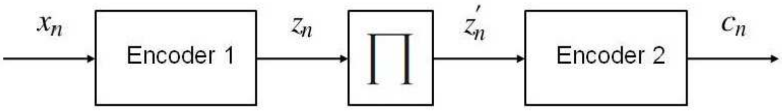

Figure 2.2: SCCC Encoder

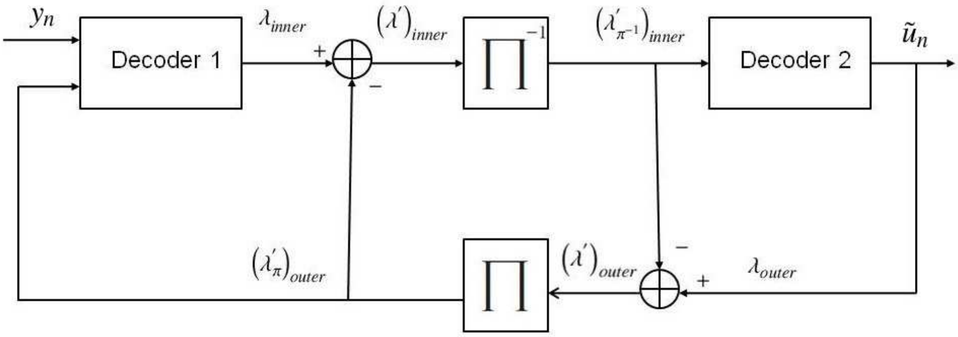

Figure 2.3: SCCC Decoder

rate of the SCCC encoder is the product of the rates of the convolutional encoders $R_{outer} = n/p$ and $R_{inner} = p/k$. The interleaver length N is important for the performance of the code. As the length of the interleaver increases, the bit error probability decreases with the inverse power of it under the assumption of uniform interleaving [7]. Also, in order to obtain a good performance from an SCCC, the inner code should be a recursive code and the outer code should have the largest free distance possible [8]. The details of the encoder structure for recursive codes are explained in [17].

The decoders are soft-output decoders whose outputs are in the form of *log-likelihood ratio*s (LLR). LLR is a probabilistic measure of a bit u_n to be 1 or 0, and formulated by

$$\lambda = ln\frac{P(u_n = 1|\mathbf{y})}{P(u_n = 0|\mathbf{y})}. \tag{2.30}$$

with $\mathbf{y}$ denoting the vector of received samples. The output of the inner and outer decoder are denoted by λ^i_{inner} and λ^i_{outer} respectively, where the superscripts show the iteration number.

The extrinsic information $\left(\lambda'\right)_{inner}^{i}$ and $\left(\lambda'\right)_{outer}^{i}$ are calculated by subtracting the extrinsic information given by the other decoder in the previous iteration from the current output

$$\left(\lambda'\right)_{inner}^{i} = \lambda_{inner}^{i} - \left(\lambda'_{\pi}\right)_{outer}^{i-1} \; \left(\lambda'\right)_{outer}^{i} = \lambda_{outer}^{i} - \left(\lambda'_{\pi^{-1}}\right)_{inner}^{i-1}$$

In the next step, $\left(\lambda'_{\pi^{-1}}\right)_{inner}^{i}$ is fed to the outer decoder as the input and $\left(\lambda'_{\pi}\right)_{outer}^{i}$ is fed to the inner decoder as the *a priori* information. Performance curves of SCCC are given in [8]. Compared to parallel concatenated convolutional codes (PCCC), SCCC's yield slightly worse performance at low to moderate SNR values yet perform better at high SNR's since their error floors are much lower than those of PCCC's.

CHAPTER 3

CHANNEL ESTIMATION BASED ON THE LLOYD-MAX ALGORITHM

In this chapter, the application of the Lloyd-Max algorithm to the channel estimation problem in wireless communications systems will be investigated. Using this channel estimation method, the performance of different receiver systems are explored by simulations.

3.1 Channel Estimation with the Lloyd-Max Algorithm

As explained in Chapter 2, the Lloyd-Max algorithm is a quantization algorithm based on finding the best quantization points that minimize the total squared error. The Lloyd-Max Channel Estimator (LMCE) uses this method for the channel estimation problem. For the received channel samples, the algorithm finds the optimum quanta. Each of the quanta are the constellation points observed after the scaling and rotation of the channel.

LMCE starts the channel estimation process with the unit energy M-PSK symbols as the initial quanta

$$\begin{aligned} q_1 &= \alpha_1 = exp(j\frac{2\pi \cdot 0}{M}) \\ q_2 &= \alpha_2 = exp(j\frac{2\pi}{M} \cdot 1) \\ &\vdots \\ q_M &= \alpha_m = exp(j\frac{2\pi \cdot (M-1)}{M}). \end{aligned} \tag{3.1}$$

Different from the algorithm explained in Chapter 2, the Lloyd-Max algorithm in channel estimation does not determine region boundaries for the received samples. The Lloyd-Max channel estimator uses the received symbols as training symbols to determine the quantization points. Denote the set of indices of the received samples falling into the region Q_m by

$$S_m = \left\{ i : |y_i - q_m| \leq \left| y_i - q_p \right|, \ \ \forall p \neq m \right\} \tag{3.2}$$

with $|S_m|$ being the cardinality of S_m. In Step 1, the center of mass of the points in S_m are calculated by

$$q_m = \frac{1}{|S_m|} \sum_{i \in S_m} y_i, \qquad m = 1, \ldots, M \tag{3.3}$$

which corresponds to an arithmetic mean operation.

By evaluating (3.3) for each m, a set of new quanta is found. In Step 2, the index sets are updated with the new quanta using (3.2) again. The algorithm continues by repeating Steps 1 and 2 one after another either until a stopping criterion is met or for a desired number iterations.

In general, the Lloyd-Max algorithm does not impose a structure on the set of quanta. However, the channel estimation problem should exert a relation between quanta since the received constellation is just a scaled and rotated version of the original constellation. Thus, we constrain the quanta obtained using LMCE by

$$q_m = \hat{h}\alpha_m \tag{3.4}$$

where $\hat{h}$ corresponds to the channel estimate, though in a phase ambiguous manner as to be explained later.

The constraint formulated in (3.4) forces the channel estimation procedure to use all of the observations to estimate the channel instead of using the observations in each Q_m seperately. This can be achieved by using the decision-directed LS structure explained in Chapter 2. Using (2.29) for a narrowband channel, the channel estimation equation becomes

$$\hat{h} = \frac{1}{E_s N} \sum_{m=1}^{M} \sum_{k \in S_m} y_k (\alpha_m)^* \tag{3.5}$$

where $N = \sum_{m=1}^{M} |S_m|$. As it is seen from (3.5), the hard decisions for all indices $k \in S_m$ are taken to be α_m, which are the original constellation points.

When LMCE calculates the estimated constellation, the MSE of the received constellation can be calculated. By using the index sets defined above, the MSE can be calculated as

$$\hat{\sigma_n}^2 = \frac{1}{N} \sum_{m=1}^{M} \sum_{k \in S_m} |y_k - q_m|^2 \,. \tag{3.6}$$

It should be noted that there is a phase ambiguity in the channel estimate $\hat{h}$ and the estimated constellation points q_m. This is the result of the unknown phase of the channel, $\Omega\{h\}$. If $|\Omega\{h\}| > \frac{2\pi}{M}$, then at the received constellation, the phase of the channel appears to be $mod_{\frac{2\pi}{M}}\{|\Omega\{h\}|\}$ since the constellation points α_m are points chosen on a circle such that the distances between the phases of neighbouring ones are constant and equal to $\frac{2\pi}{M}$. For this reason, $\Omega\left\{\hat{h}\right\}$ may be the phase of the actual channel estimate or $\frac{2\pi k}{M}$ rotated versions of it for $k = 1, \ldots, M - 1$.

The estimation procedure can be visualized by examining the constellation points of the received samples and the channel estimation output. Assuming QPSK modulation, the received scatterplot for 50 symbols with instantaneous SNR $E_b/N_0 = 10$dB is seen as in Figure 3.1. The Lloyd-Max algorithm makes 10 iterations to estimate the constellation points. The original, rotated and estimated constellations are given in Figure 3.2.

To have a better understanding of the performance of the algorithm, the mean-squared error (MSE) value of the estimation error is given in Figure 3.3. The channel estimate of the Lloyd-Max algorithm is obtained in a genie-aided manner. This means that the point that has the minimum distance to the perfectly known channel among the points of the estimated constellation is chosen to be the channel estimate output the Lloyd-Max algorithm and the MSE is calculated using the chosen point to avoid gross errors. The x-axis denoted by E_b/N_0 shows the instantaneous SNR values.

A lower bound for the channel estimation MSE for narrowband channels can be derived using the LS algorithm. The noise process is white, i.e., the cross-correlations of different samples are zero. Data symbols are independent and identically distributed. Since the noise and data

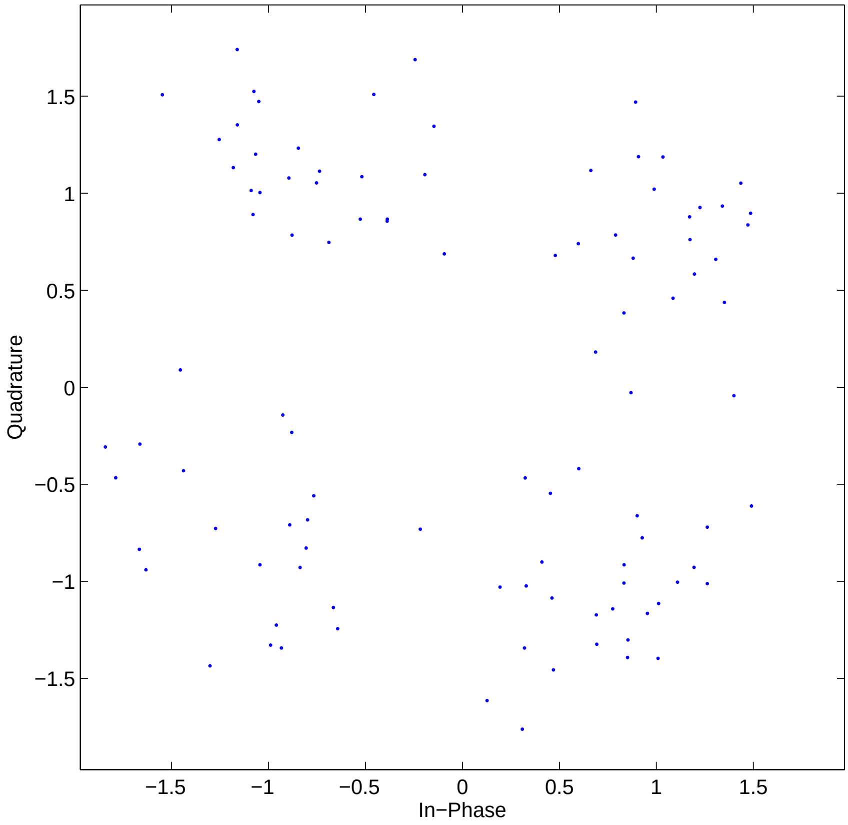

Figure 3.1: Received channel samples

symbols are uncorrelated,

$$
\begin{aligned}
MSE_{LS} &= E\left\{\left|\frac{1}{NE_s}\sum_{k=1}^{N} y_k a_k^* - h\right|^2\right\} \\
&= E\left\{\left|\frac{1}{NE_s}\sum_{k=1}^{N} (ha_k + n_k)a_k^* - h\right|^2\right\} \\
&= E\left\{\left|\frac{1}{NE_s}\sum_{k=1}^{N} h|a_k|^2 + \frac{1}{NE_s}\sum_{k=1}^{N} n_k a_k^* - h\right|^2\right\} \\
&= E\left\{\left|\frac{1}{NE_s}\sum_{k=1}^{N} n_k a_k^*\right|^2\right\} = \frac{1}{N^2E_s^2}\sum_{k=1}^{N} E\left\{|n_k|^2\right\} E\left\{|a_k|^2\right\} \\
&= \frac{1}{N^2E_s^2}\cdot N_0 N E_s = \frac{N_0}{NE_s}
\end{aligned}
\qquad (3.7)
$$

where N is the number of pilot symbols, and expectation is taken over the ensembles of data and noise.

In Figure 3.3, MSE of the LMCE algorithm is depicted with respect to different number of received samples used for estimation. The bounds for each length are calculated as if all samples that are used were pilot symbols and LS estimation was employed. As seen from the figure, the LS performance is achieved around an instantaneous SNR value of 8dB. Since LMCE is a blind algorithm, its estimations may carry high errors if the samples used in estimation are not reliable, i.e., the noise level affecting the samples is high. For LMCE, the samples are considered as reliable starting from 8dB SNR. The performance of LMCE changes with the number of LMCE iterations. The effect of the number of iterations on the performance of the estimator is given in Figure 3.4 for 50 symbols in a dwell. It is observed in Figure 3.4 that the LMCE reaches its capability with nearly 10 iterations. Therefore, for the rest of the thesis, the Lloyd-Max algorithm makes 10 iterations unless stated otherwise.

Another important issue to examine is the comparison of MSE of LMCE with other methods defined in Sections 2.3 and 2.5. Figure 3.5 shows this comparison. The estimation is cal-

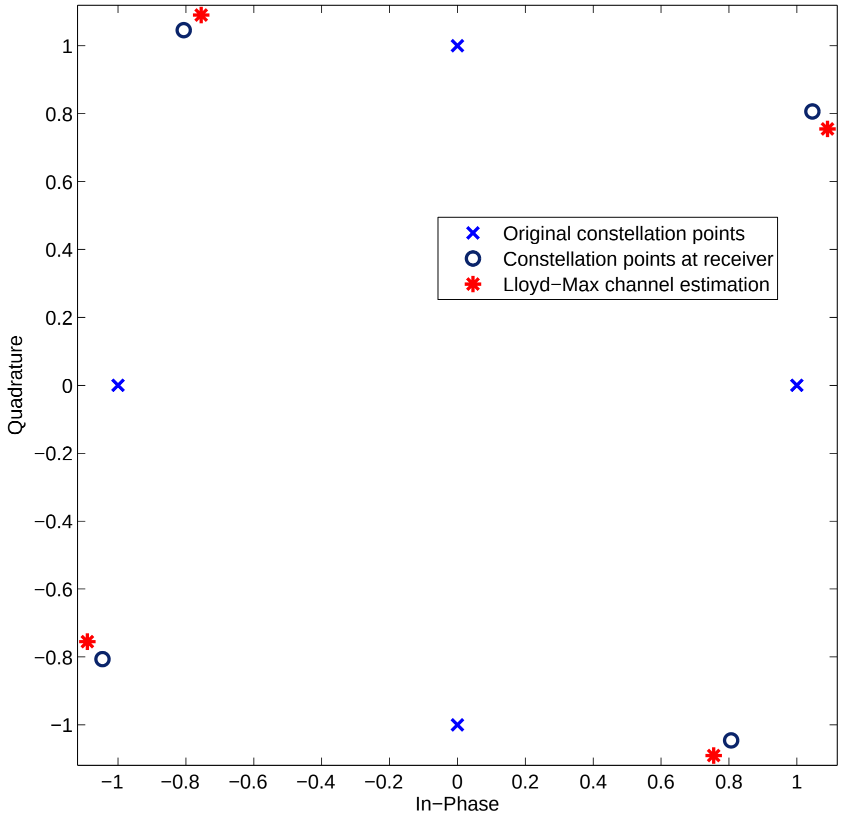

Figure 3.2: Original, channel output and estimated constellations

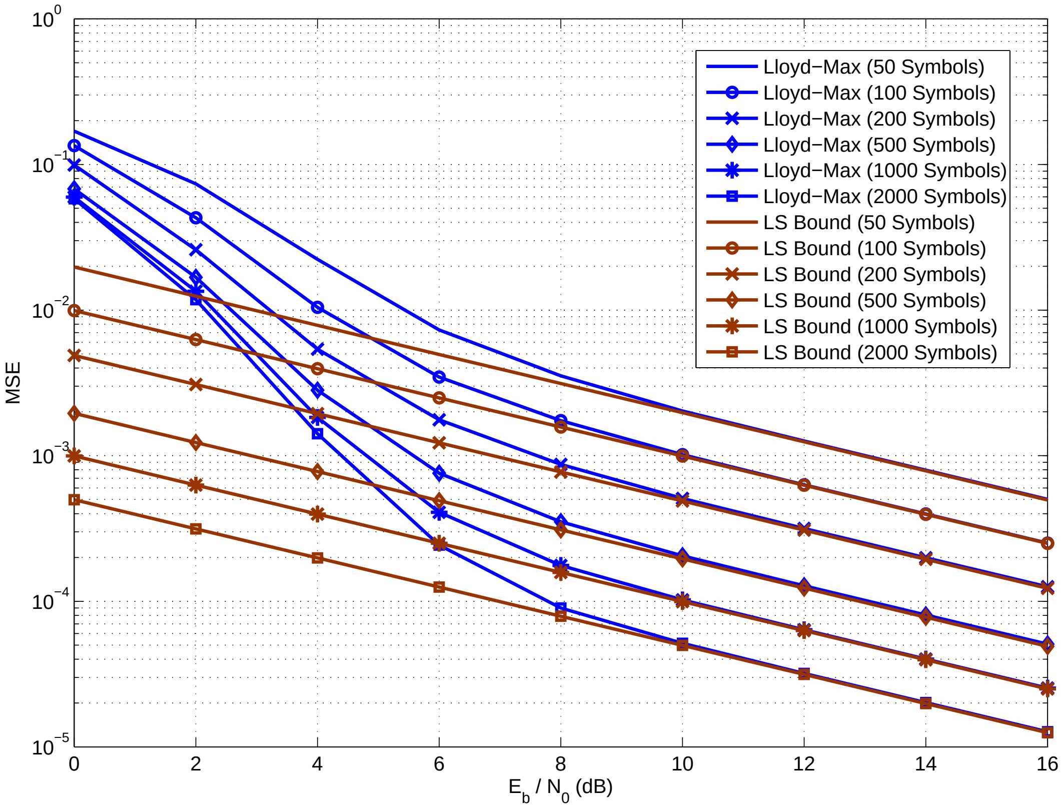

Figure 3.3: MSE of Lloyd-Max channel estimate for different block lengths

culated over different number of pilot symbols for LS and DDLS, and 50 data symbols for DDLS and LMCE. The pilot symbols of LS and DDLS are chosen from the optimal pilot sequences obtained for LS channel estimation in [4]. It is seen from Figure 3.5 that LMCE performs very close to LS bound for SNR values greater than 6dB. DDLS performs close to LMCE for pilot lengths of 5 and 10. It should be noted that the gain obtained from DDLS when pilot number is increased to 10 from 5 is not significant. LS estimation performs close to LMCE and DDLS for lower SNR values but LMCE and DDLS outperforms LS as SNR increases.

As SNR exceeds 10dB, LMCE and DDLS have exactly the same estimation MSE and they satisfy the LS bound. This is the result of the similarity of LMCE and DDLS in using hard-decision symbols for channel estimation. LMCE makes hard-decisions at every iteration with respect to the latest channel estimate, and DDLS makes hard-decisions with respect to the channel estimate obtained from pilot symbols only. At high SNR values, the channel samples

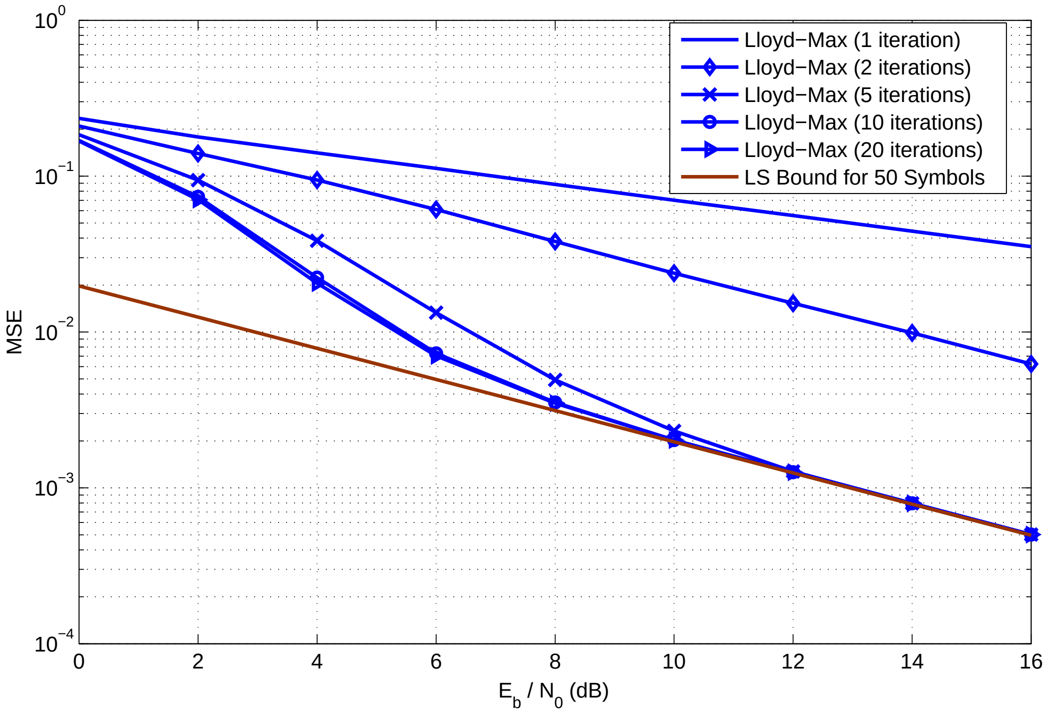

Figure 3.4: MSE of Lloyd-Max channel estimate for different number of iterations (50 symbols in a dwell)

used as hard-decisions are so reliable that there is no difference between using hard-decisions and known symbols for channel estimation.

Choosing the correct channel estimate from the output quanta of LMCE is a major problem that has to be dealt with. In Sections 3.2 and 3.3, two different approaches for this problem will be discussed.

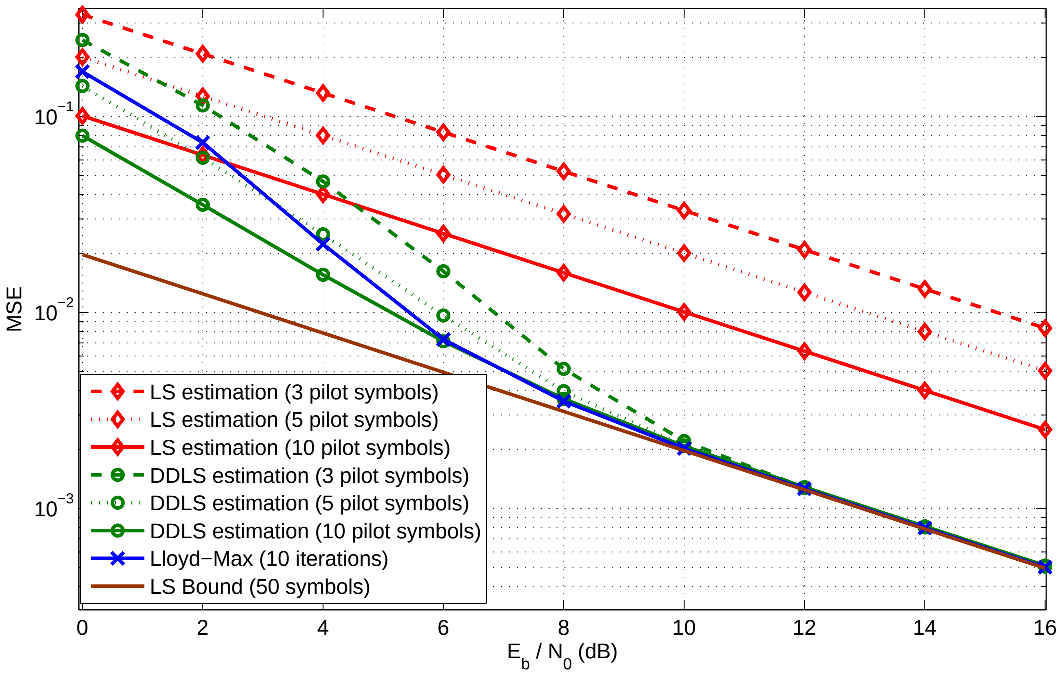

Figure 3.5: MSE of Lloyd-Max, LS, and DDLS channel estimates

3.2 Pilot-Aided LMCE

Among the output quanta q_m calculated by LMCE, one quantum has to be chosen to be the channel estimate. This choice can be made by using a reference point. By comparing the reference to the quanta, the quantum that has the minimum distance to the reference can be chosen as the channel estimation. Defining this method, a question arises: how does one find a reference point? It is clear that the best reference point is the channel itself, which was defined as the genie-aided method in Section 3.1. Since the perfect channel state information is obviously not available, an estimate of the channel can be used. This estimate can be obtained from LS or DDLS estimation methods using a small number of pilot symbols.

The performance of LMCE aided by LS or DDLS can be examined by checking the error rates. The channel model to be used in the simulations is the narrowband Rayleigh fading channel model[16]. The channel estimates of LS and DDLS are calculated as explained in Chapter 2. The channel estimates of LMCE aided by LS or DDLS are calculated by

$$\hat{h}_{LMCE} = \arg \min_{m \in 1,\ldots,M} \left| \hat{h}_{LS/DDLS} - q_m \right|^2 \tag{3.8}$$

where q_m are the constellation points calculated by LMCE. The noise variance estimate of LMCE is obtained by (3.9). The noise variance estimates of LS and DDLS can be obtained in a similar way by

$$\hat{\sigma_n}^2 = \frac{1}{N} \sum_{k=1}^{N} \left| y_k - \hat{h}_{LS/DDLS} \right|^2 . \tag{3.9}$$

The transmitter uses convolutional channel coding with a bit interleaver following the encoder. The coded and interleaved bits are passed through the modulator. For the simulations, channel coding is performed by a rate $R = \frac{1}{2}$ convolutional code with generator polynomials $g[1] = (33)_8$ and $g[0] = (23)_8$ in octal form [17]. The channel interleaver is a uniform random interleaver for which the probabilities of each bit to be sent in a specific location is equal.

The convolutional decoder at the receiver is a soft-output decoder using the BCJR [18] algorithm. A decoder with the BCJR algorithm calculates the output LLR's of the information bits as

$$\lambda(u_m) = ln \frac{P(u_m = 1|\mathbf{y})}{P(u_m = 0|\mathbf{y})} \tag{3.10}$$

where u_m is the m^{th} information bit and $\overline{y}$ is the vector of received samples. A BCJR (MAP) decoder uses three different probabilities in its calculations, which are given by

$$\alpha_k(p) = P\left(\Psi_k = p, \mathbf{r}_{<k}\right) \tag{3.11}$$

$$\beta k(q) = P\left(\mathbf{r}_{>k} | \Psi_{k+1} = q\right) \tag{3.12}$$

$$\gamma_k(p,q) = P\left(\Psi_{k+1} = q, r_k | \Psi_k = p\right) \tag{3.13}$$

$\alpha_k(p)$, $\beta_k(q)$, and $\gamma_k(p,q)$ are the forward, backward, and transition probabilities, respectively. The variable Ψ_k is the state variable of the trellis at time k and p and q denote different states. Using the Max-Log-MAP approximation to Log-MAP algorithm [19], the forward and backward probabilities of the decoder are given by

$$\begin{aligned} A_{k+1}(q) &= ln\left(\alpha_{k+1}(q)\right) \\ &= ln\left(\sum_{p=1}^{M-1} \alpha_k(p)\gamma_k(p,q)\right) \\ &= ln\left(\sum_{p=1}^{M-1} exp(A_k(p) + \Gamma_k(p,q))\right) \\ &\approx max_{p\in\{0,1,\dots,M-1\}}\left(A_k(p) + \Gamma_k(p,q)\right) \end{aligned} \tag{3.14}$$

and

$$\begin{aligned} B_k(p) &= ln\left(\beta_k(p)\right) \\ &= ln\left(\sum_{q=1}^{M-1} \beta_{k+1}(q)\gamma_k(p,q)\right) \\ &= ln\left(\sum_{q=1}^{M-1} exp(B_{k+1}(q) + \Gamma_k(p,q))\right) \\ &\approx max_{q\in\{0,1,\dots,M-1\}}\left(B_{k+1}(q) + \Gamma_k(p,q)\right). \end{aligned} \tag{3.15}$$

The state transition probability gamma is

$$\begin{aligned} \Gamma_k(p,q) &= ln(\gamma_k(p,q)) \\ &= ln\left(\frac{1}{\sqrt{2\pi\hat{\sigma_n}^2}} exp(-\frac{\left|y_k - \hat{h}\acute{a}(p,q)\right|^2}{2\hat{\sigma_n}^2}) \prod_{n=1}^{K} P(c_{n,k} = \acute{c}_n(p,q))\right) \\ &= ln\left(\frac{1}{\sqrt{2\pi\hat{\sigma_n}^2}}\right) - \frac{\left|y_k - \hat{h}\acute{a}(p,q)\right|^2}{2\hat{\sigma_n}^2} + ln\left(\prod_{n=1}^{K}\left[\frac{e^{-\lambda^{'}(c_{n,k})/2}}{1+e^{-\lambda^{'}(c_{n,k})}}\right] e^{\acute{c}_n(p,q)\lambda^{'}(c_{n,k})/2}\right) \\ &= k_1(\hat{\sigma_n}^2) + k_1(\lambda^{'}(c_{m,k})) - \frac{\left|y_k - \hat{h}\acute{a}(p,q)\right|^2}{2\hat{\sigma_n}^2} + \sum_{n=1}^{K} \acute{c}_n(p,q)\frac{\lambda^{'}(c_{n,k})}{2}, \end{aligned} \tag{3.16}$$

where $u_{n,k}$ is the n^{th} information bit that is encoded and carried at time k in the symbol a_k, $\acute{u}_n(p,q)$ is the n^{th} input bit for encoder to change its state from p to q, $\acute{a}(p,q)$ is the the output symbol when the encoder changes its state from p to q, and K is the number of input bits that are encoded together at a single time instance. The constant values $k_1(\hat{\sigma_n}^2)$, $k_2(\lambda^{'}(u_{n,k}))$, $\frac{|y_k|^2}{2\hat{\sigma_n}^2}$, and $\frac{\left|\hat{h}\acute{a}(p,q)\right|^2}{2\hat{\sigma_n}^2}$ can be ignored, since they are eliminated when the likelihood-ratios are calculated. Therefore, the branch metric used in the decoder is

$$\Gamma_k(p,q) = \frac{\Re\left\{y_k^*\hat{h}\acute{a}(p,q)\right\}}{\hat{\sigma_n}^2} + \sum_{n=1}^{K} \acute{u}_n(p,q)\frac{\lambda^{'}(u_{n,k})}{2}. \tag{3.17}$$

Instead of using received symbols y_k in (3.17), the LLR's of coded bits calculated from y_k can be used. Calculation of coded bit LLR's from y_k [24] is done by

$$z_{n,k} \triangleq \frac{P\left(u_{n,k} = 1|y_k\right)}{P\left(u_{n,k} = 0|y_k\right)} \tag{3.18}$$

Eqn. (3.18) can be approximated as

$$z_{n,k} \approx min_{\mathbf{u}_1}\left|y_k - \hat{h}\alpha\left(\mathbf{u}_1\right)\right|^2 - min_{\mathbf{u}_0}\left|y_k - \hat{h}\alpha\left(\mathbf{u}_0\right)\right|^2 \tag{3.19}$$

where $\mathbf{u}_j$ is the set of groups of log_2M bits having the n^{th} element as j, and $\alpha\left(\mathbf{u}_j\right)$ are the modulation symbols used to represent the bit groups in $\mathbf{u}_j$. Using (3.19), (3.17) is modified to

$$\Gamma_k(p,q) = \frac{\Re\left\{\sum_{n=1}^{log_2M} z_{n,k}^*\acute{z}_n(p,q)\right\}}{2} + \sum_{n=1}^{K} \acute{u}_n(p,q)\frac{\lambda^{'}(u_{n,k})}{2}. \tag{3.20}$$

with $\acute{z}_n(p,q)$ as n^{th} the output symbol when the encoder changes its state from p to q. When the decoder uses (3.20) instead of (3.17), the decoder inputs are soft input bits instead of modulation symbols. This enables the transmitter to use bit interleaving between the encoding and modulation operations when modulations with $M > 2$ are used. The estimates $\hat{\sigma_n}^2$ and $\hat{h}$ are the calculated by LS, DDLS, or the Lloyd-Max algorithm.

For the simulations, a frame length of 5 dwells, and a dwell length of 50 data symbols with QPSK modulation are used. Channels and noise at each dwell are independent. The noise power N_0 is assumed to be unknown, therefore, it has to be estimated if needed. Figure 3.6 shows the bit error rate performance and Figure 3.7 shows the frame error rate performance of LMCE when the channel estimate is obtained by the reference of LS or DDLS channel estimates. The performances are compared with decoding with perfect *channel state information*

(CSI), which means that decoding is done with perfectly known channel and noise parameters. The x-axes of the graphs show the average SNR values for the uncoded information bits in dB scale. The extra energy needed for the pilots are taken into account by distributing the energy of pilots over data bits. Since rate of the channel code $R = \frac{1}{2}$ and the modulation is QPSK,

$$\gamma_b = \gamma_s \cdot \frac{N+L}{N} \cdot \frac{1}{Rlog_2 M} \tag{3.21}$$

where L is the number of data symbols and N is the number of pilot symbols in a dwell. The pilot symbols of LS and DDLS are chosen from the optimal pilot sequences as stated in Section 3.1. The simulations are run until 50 frame errors occur for each SNR value. It should be noted that the simulation parameters given here are valid for all simulations throughout this thesis.

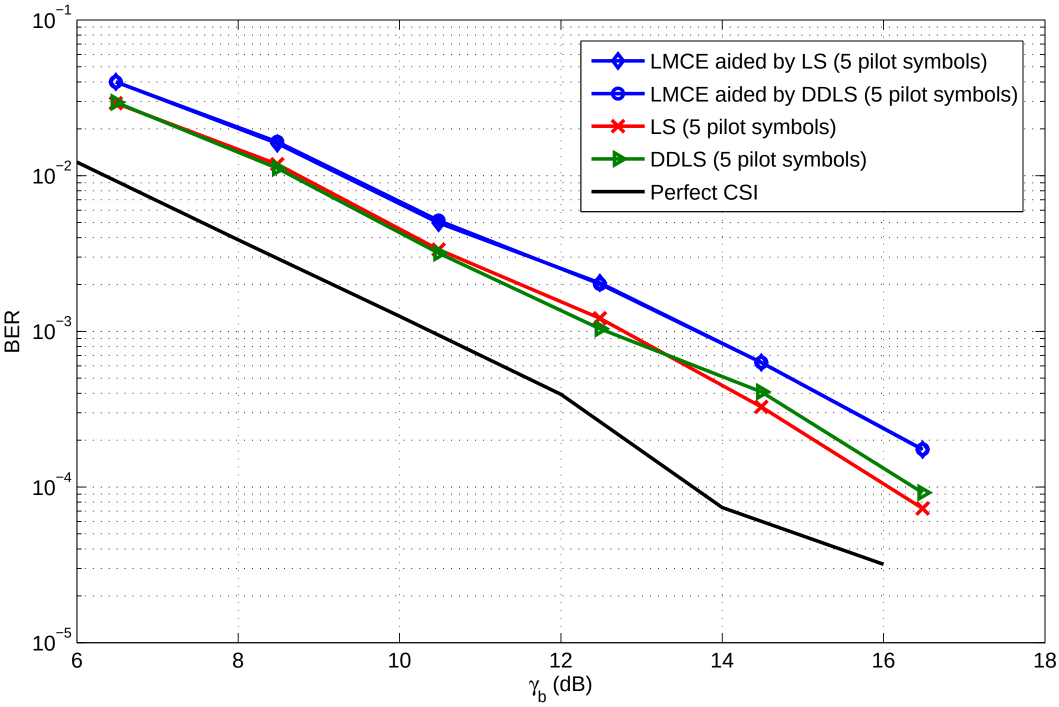

Figure 3.6: BER performance with convolutional code

Using Max-Log-MAP algorithm with the channel estimates may yield different results than other realization methods of BCJR algorithm. Max-Log-MAP is an approximation to the Log-MAP realization of the BCJR algorithm, and the robustness of these algorithms to estimation errors may differ. Figure 3.7 shows the frame error rate performances of the estimation algo-

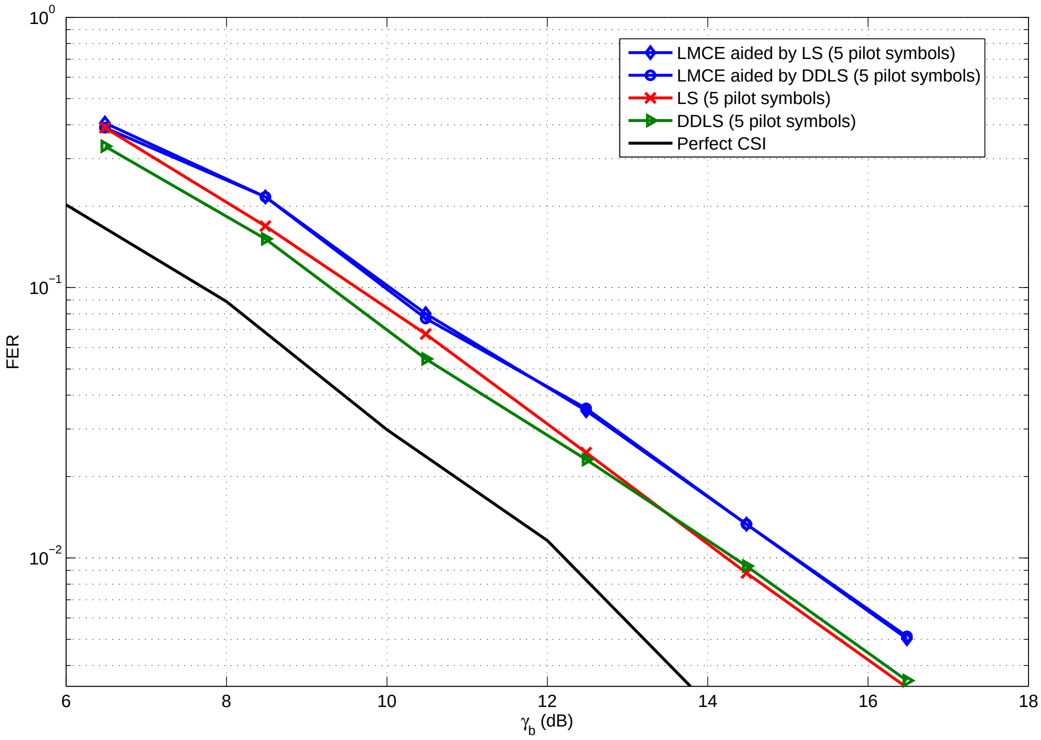

Figure 3.7: FER performance with convolutional code

rithms with a soft-output decoder employing Log-MAP algorithm. It is shown in Figure 3.8 that the channel estimate errors have similar effects on both algorithms, therefore Max-Log-MAP algorithm is used for the rest of the simulations in this thesis.

It is seen that the error performance of DDLS and LS are slightly better than LMCE. From Figure 3.5, it is known that MSE's of DDLS and LS are larger than or equal to MSE of LMCE, so it is expected that their error performances also show the same characteristics. The reason DDLS and LS perform better than LMCE is the wrong reference problem. LMCE chooses its channel estimate among all quanta q_m using the reference that DDLS or LS calculates. If there is excessive noise on the DDLS or LS channel estimate, the reference point may be closer to a neighbouring quantum instead of the actual channel estimate quantum. In such a situation, the LMCE channel estimate becomes a phase shifted version of the actual channel estimate, and the phase shift is on the order of $\frac{2\pi}{M}$. This phase shift causes major decreases in the error performance. On the other hand, DDLS and LS performances may not be affected as much as LMCE, since the phase shifts of DDLS or LS do not have to be on the order of $\frac{2\pi}{M}$

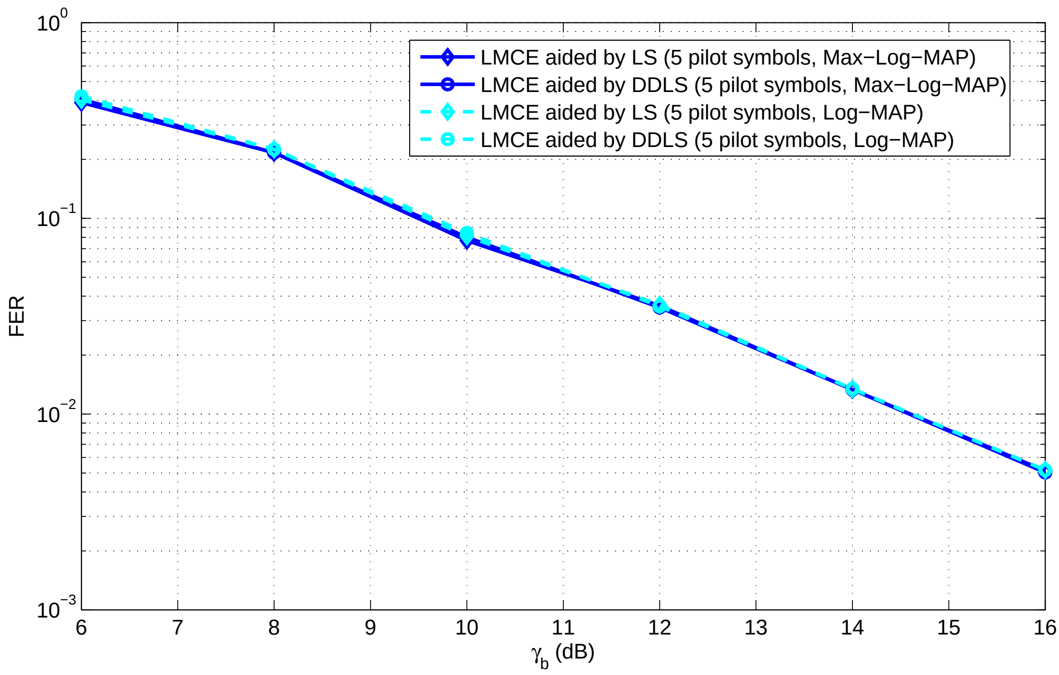

Figure 3.8: FER performance with convolutional code (Log-MAP algorithm)

for causing erronuous decisions for LMCE. To validate this hypothesis, the performance of genie-aided LMCE can be compared with the performances of DDLS and LS.

Figure 3.9 and Figure 3.10 suggest that the performance decrease in LMCE is caused by the noisy reference points. From Figures 3.9 and 3.10, it is seen that there is about a 1dB loss between decoding with perfect channel state information and genie-aided decoding even at SNR values greater than 10dB. Figure 3.3 shows that the MSE of LMCE channel estimation is around $2x10^{-3}$ at this SNR level. Since this error is negligible, it is expected that the genie-aided LMCE would perform almost identical to the perfect CSI case. However, under Rayleigh fading channels, the instantaneous SNR is generally not equal to the average SNR. Therefore, the MSE of the estimations may be much larger than $2x10^{-3}$ even if the average SNR is 10dB.

It is important for LMCE to have good reference points. Noisy reference points may cause phase-shifts on the order of $\frac{2\pi}{M}$ and affect the error performance severely. It can be concluded that obtaining a single channel estimate from LMCE using more noisy reference points performs poorer than using the more noisy reference point directly. We propose a solution to this

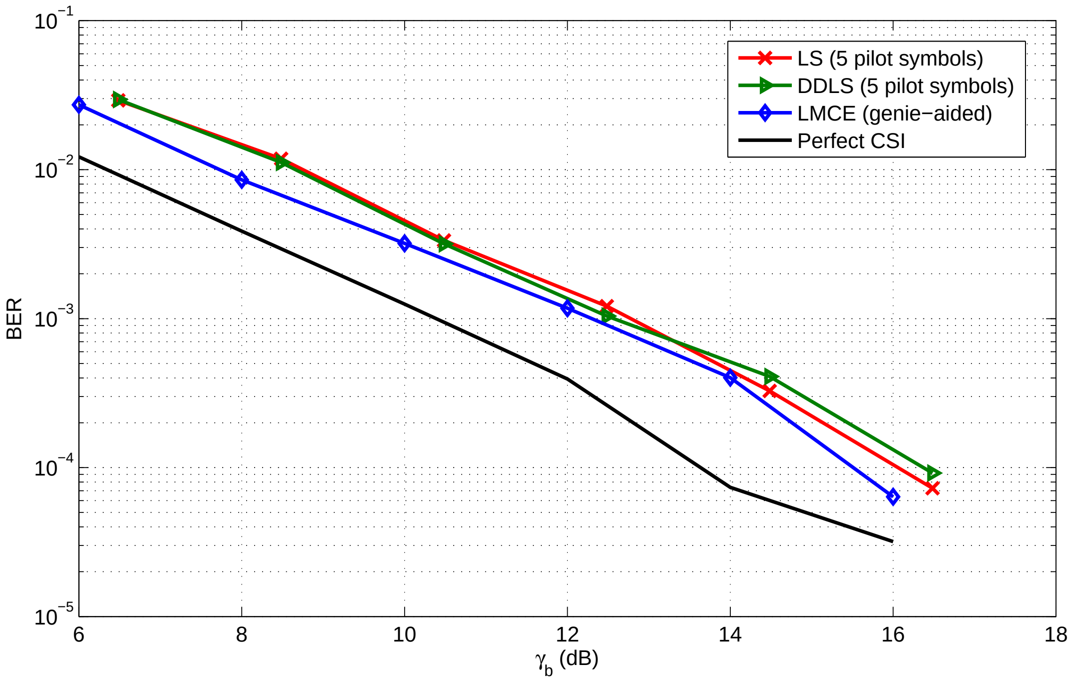

Figure 3.9: BER performance of genie-aided LMCE

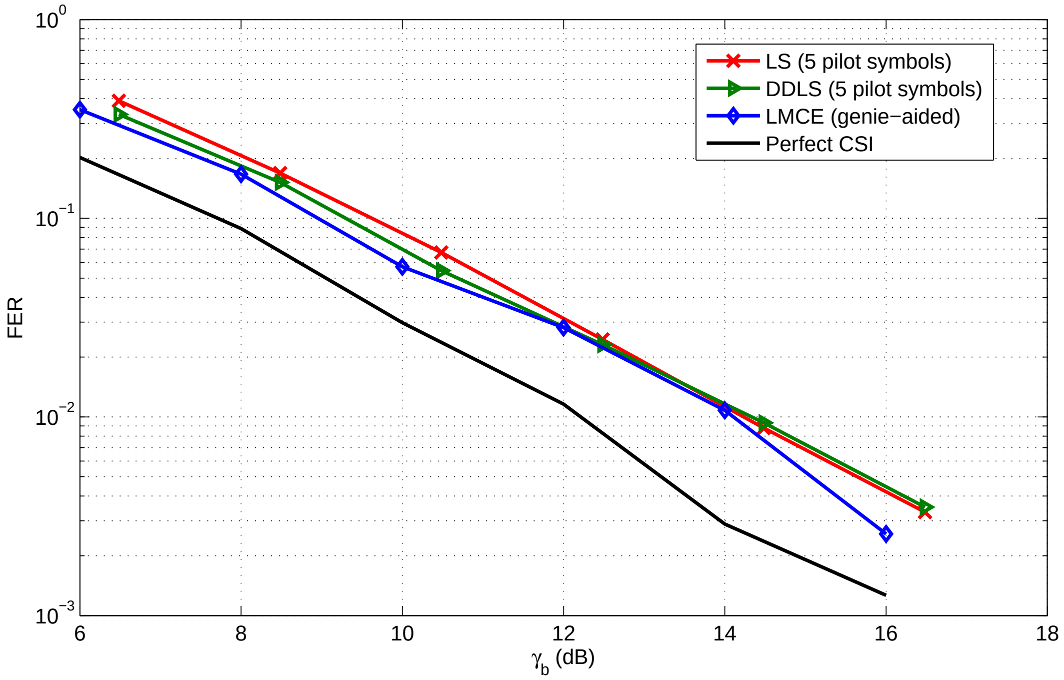

Figure 3.10: FER performance of genie-aided LMCE

problem in the following section.

3.3 Differential Encoding with LMCE

3.3.1 Encoding

The output quanta of the LMCE is usually quite close to the actual channel estimate or its phase-shifted replicas. In Section 3.2, it was shown that trying to find the actual channel estimate by using more noisy references is not an effective approach. It also require the use of pilot symbols for calculating references for LMCE. Instead, other approaches such as differential demodulation or decoding that work well with phase ambiguity may produce better results.

Differential Phase Shift Keying (DPSK) modulation is a modulation technique in which modulation is performed based on the phase differences of consecutive symbols [2]. In DPSK, instead of transmitting the M-PSK symbols directly, the symbol-by-symbol summation of the phases of the M-PSK symbols are transmitted, such that

$$d_k = d_{k-1}a_k \tag{3.22}$$

where a_k is an M-PSK symbol, d_k and d_{k-1} are DPSK symbols which take values from the alphabet $\left\{e^{j\frac{2\pi\cdot 0}{M}}, e^{j\frac{2\pi\cdot 1}{M}}, \ldots, e^{j\frac{2\pi\cdot(M-1)}{M}}\right\}$. Another representation for DPSK modulation is in the form of $modM$ summation.

$$\acute{d}_k = (\acute{d}_{k-1} + \acute{a}_k)_{modM}. \tag{3.23}$$

In this representation $\acute{a}_k$ and $\acute{d}_k$ take values from the alphabet $\{0, 1, \ldots, M-1\}$, where each element represents $k = log_2 M$ bits. Therefore, a differential modulator can also be considered as a $rate = \frac{k}{k}$ recursive encoder, which is given in Figure 3.11. The mapping to the constellation can be done using natural or Gray coding.

Since differential modulation can be expressed as convolutional encoding, a differential modulated signal can be decoded coherently as a convolutional code. The differential decoding is a coherent procedure, since the soft-output decoders require channel information. However, coherent differential decoding is robust against the phase ambiguity in the channel estimate

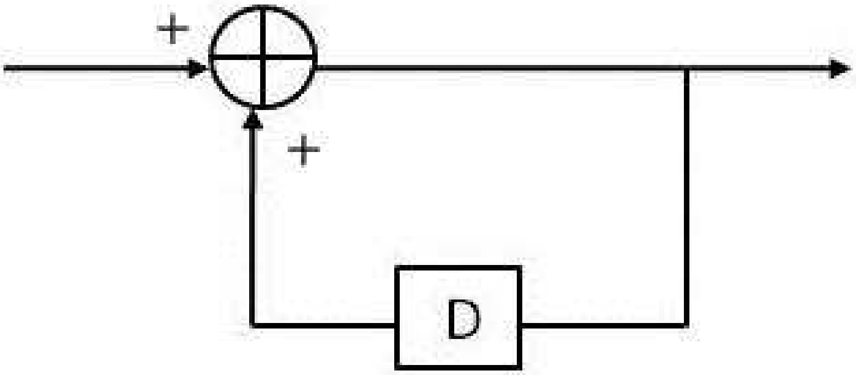

Figure 3.11: Differential encoder

when the ambiguity is on the order of symbol phase differences. This situation can be understood more clearly with the help of a differential decoder trellis given in Figure 3.12. For simplicity, only the transitions from two states are shown.

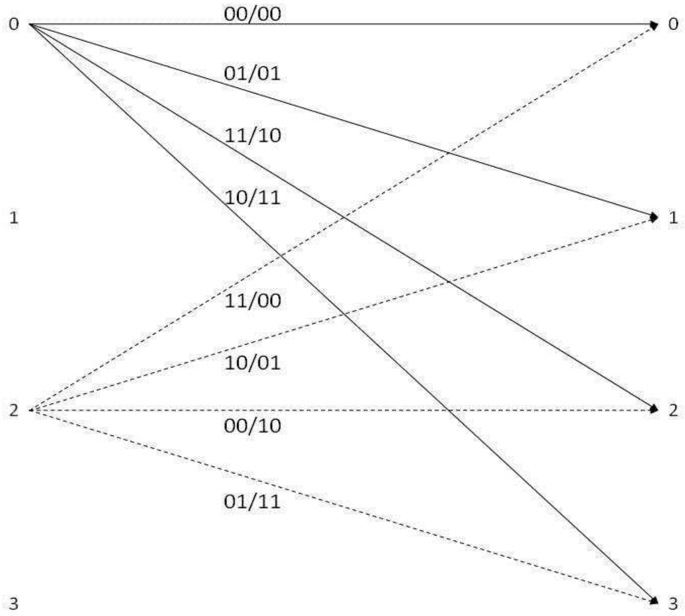

Figure 3.12: Differential trellis

Figure 3.12 shows the trellis of a Gray-coded QPSK differential encoder. The states of the trellis are actually the outputs of the trellis branches coming into that state. Therefore, the output QPSK symbols are matched to the states of the trellis. The inputs causing the state transitions are parallel, e.g., transition from state 0 to state 1, state 1 to state 2, state 2 to state 3, and state 3 to state 0 are all the results of the input [01]. When the most probable states are decided, it is not important that the channel or channel with a phase shift $\frac{2\pi k}{4}$ is used, since the same amount of shift in all phases of the trellis path yield the same output as the correct trellis path. An advantage of choosing coherent differential decoding over noncoherent differential modulation is that an SCCC code forms by using a convolutional encoder as the outer code and the differential encoder as the inner code [6]. By this method, coding gain can also be obtained without increasing the overall coding rate.

In a frequency hopping system, the transmitter and receiver structures employing an SCCC code are given in Figure 3.13 and Figure 3.14 where the inner encoder is a differential encoder and the outer a convolutional one.

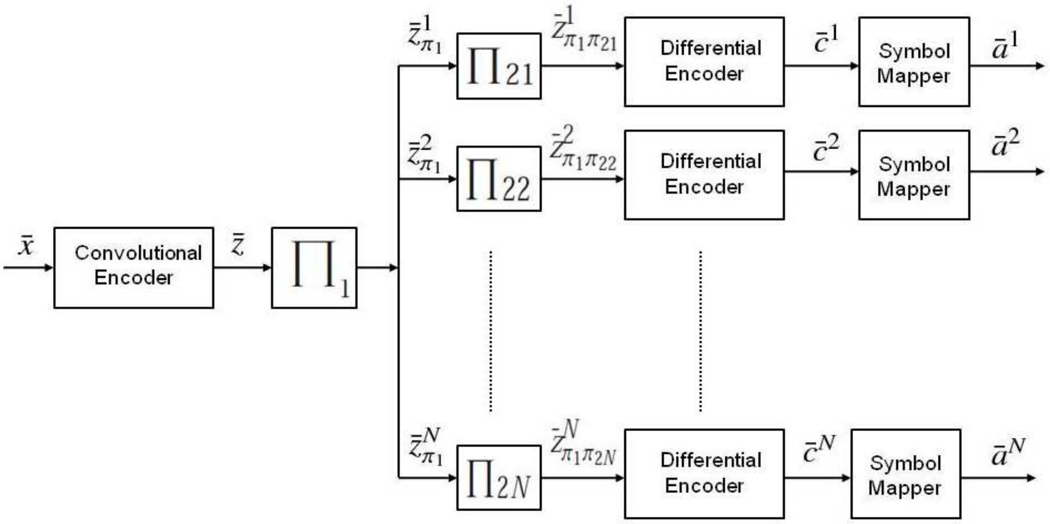

Figure 3.13: Differential SCCC encoder

Interleaving is performed in two stages. In the first stage, the bits are interleaved using a block interleaver [19]. A block interleaver writes the input bits to the rows of the interleaver matrix and reads the columns of the matrix to form the interleaved output. This multiplexes consecutive bits into different dwells. In the second stage, bitwise random interleaving is

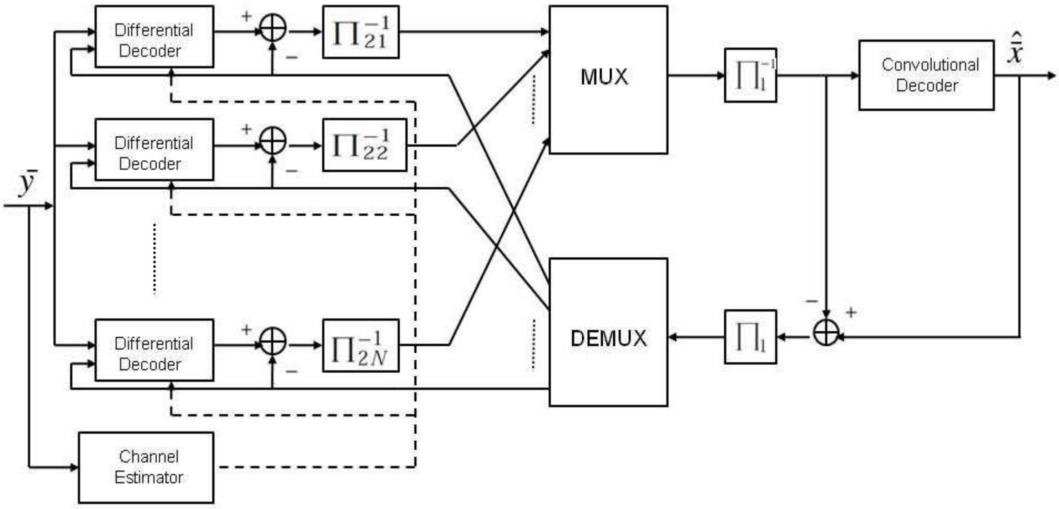

Figure 3.14: Differential SCCC decoder

done in each dwell. Using block interleaving followed by random interleaving increases the distance between the consecutive bits which increases the diversity gain [20]. Each path in Figure 3.14 with an interleaver and an encoder corresponds to a single frequency dwell sent in a carrier frequency different than others.

3.3.2 Decoding

The soft-output decoders for both differential and convolutional codes use the Max-Log-MAP algorithm. From (2.31), the extrinsic output of the differential and convolutional decoders is given by

$$\left(\lambda'\right)^i_{diff} = \lambda^i_{diff} - \left(\lambda'_\pi\right)^{i-1}_{conv},\ \left(\lambda'\right)^i_{conv} = \lambda^i_{conv} - \left(\lambda'_{\pi^{-1}}\right)^{i-1}_{diff}.$$

For simplicity, the π subscripts that denote the interleaving and deinterleaving will be ignored. λ^i are the outputs of the Max-Log-MAP algorithm explained in Section 3.2. The extrinsic differential decoder output can be written as

$$\left(\lambda'(u^m_{n,k})\right)^i_{diff} = ln\frac{P(u^m_{n,k} = 1|\bar{y_k}^m, \left(\Lambda'\right)^{i-1}_{conv} / \left(\lambda'(u^m_{n,k})\right)^{i-1}_{conv})}{P(u^m_{n,k} = 0|\bar{y_k}^m, (\Lambda')^{i-1}_{conv} / \left(\lambda'(u^m_{n,k})\right)^{i-1}_{conv})} \tag{3.24}$$

where superscript m denotes the number of the dwell, and $\left(\Lambda'\right)^{i-1}_{conv} / \left(\lambda'(z^l_{n,k})\right)^{i-1}_{conv}$ denotes the vector of all LLR outputs of convolutional decoder in the previous iteration except the LLR of the bit $u_{n,k}^{m}$. The differential decoder calculates the output LLR's using (3.14) and (3.15) for the forward and backward metrics, respectively. The state transition metrics are calculated by (3.17) since there is no bit interleaving between the differential decoder and modulator.

The vector of extrinsic differential decoder outputs obtained from each frequency dwell $\left(\Lambda'\right)_{conv}^{i-1}$ are then passed through the first and second deinterleavers to form the input to the convolutional decoder. Like the differential decoder, the convolutional decoder also uses the Max-Log-MAP algorithm. However, it uses (3.20) to calculate the transition metrics, since the inputs to the convolutional decoder are not channel observations but bit LLR values. The convolutional decoder also cannot use *a priori* information since there is no other information source in the system [8].

The convolutional encoder uses trellis termination after the encoding of data bits is completed [17]. Trellis termination enables the receiver to have the knowledge of the last state of the encoder. Therefore, the backward metric of the decoder is initialized to be 0 ($ln\,(1)$ for Max-Log-MAP decoder) for state-0 and a large negative number ($ln\,(0)$ for Max-Log-MAP decoder) for the other states. Termination can also be done in the differential encoder in the same manner. However, differential encoder termination does not yield a significant gain in the performance of systems using coherent decoding [21]. Moreover, since the system using LMCE does not have the exact information about the phase of the channel, and the states of the differential decoder are also the symbols transmitted through the channel, the initializations of the last backward metric and the first forward metric are not possible. The first forward and last backward metrics of differential decoder are given equal probabilities ($ln\left(\frac{1}{M}\right)$).

The disadvantage of using equal-probability initialization of differential decoder is that in the first iteration, the backward metrics do not make any contribution to the output LLR calculations. To explain this, first it is assumed that at a time instance k,$B_{k+1}(s_2) + \Gamma_k(s_1, s_2)$ has the largest value among all $B_{k+1}(q) + \Gamma_k(p, q)$, where s_n are the differential encoder states and $p, q \in \{s_1, s_2, \ldots, s_M\}$. Using (3.15), all $B_k(p)$ values yield the same result $B_{k+1}(s_2) + \Gamma_k(s_1, s_2)$, since each state p has a branch to the state s_2. Having no a priori information in the first iteration, the backward metrics $B_k(p)$ are the same for each k. If the forward metrics cannot be initialized either, then $\lambda(z_{l,1}^{n})$ for $l = 1, \ldots, log_2 M$ yield no information in the first iteration. Therefore, there occurs an information loss of $log_2 M$ bits for the outer decoder. To overcome this problem, $log_2 M$ pilot bits can be added in front of the input bits of the differential encoder, which causes the encoder to make 1 state transition before encoding the input bits. Then, the first $log_2 M$ LLR outputs of the differential decoder, which carry no information, are the output LLR's of these pilots. Since, the pilots are not related to the convolutional encoder, their probabilities are not used at the convolutional decoder. Therefore, the information loss is prevented. One may note that this transmission of pilot symbols corresponds to the reference symbol used in differential encoding.

3.3.3 Numerical Results

The error curves with LMCE, LS, and DDLS channel estimator with SCCC are given in the following figures. The simulation parameters in Section 3.2 are used, except that no pilots are used in the system with LMCE but a single arbitrary symbol is transmitted at the beginning of the dwells for differential encoding with LMCE. In Figures 3.15 and 3.16, the bit error and frame error performances of LS channel estimation with different training lengths are given. As stated in Section 3.2, the SNR values on the x-axes of the figures denotes the average SNR per data bit.

It is seen from Figures 3.15 and 3.16 that increasing the number of pilots to more than 5 symbols does not make a significant improvement in the error performance although it decreases the MSE of the channel estimate as shown in Figures 3.5. In Figures 3.17 and 3.18, the bit error and frame error performances of DDLS channel estimation with different training lengths are given.

Figures 3.17 and 3.18 show that the error performance of DDLS does not improve after pilot lengths of 3 symbols. Finally, the performances of LMCE, LS, and DDLS with SCCC are compared in Figure 3.19 and 3.20. It is also possible to use non-coherent demodulation with soft-output decoders and a channel estimator. Therefore, the frame and bit error performances of differential demodulation are also given in the figures.

As seen from Figures 3.19 and 3.20, the performances of all three channel estimation methods

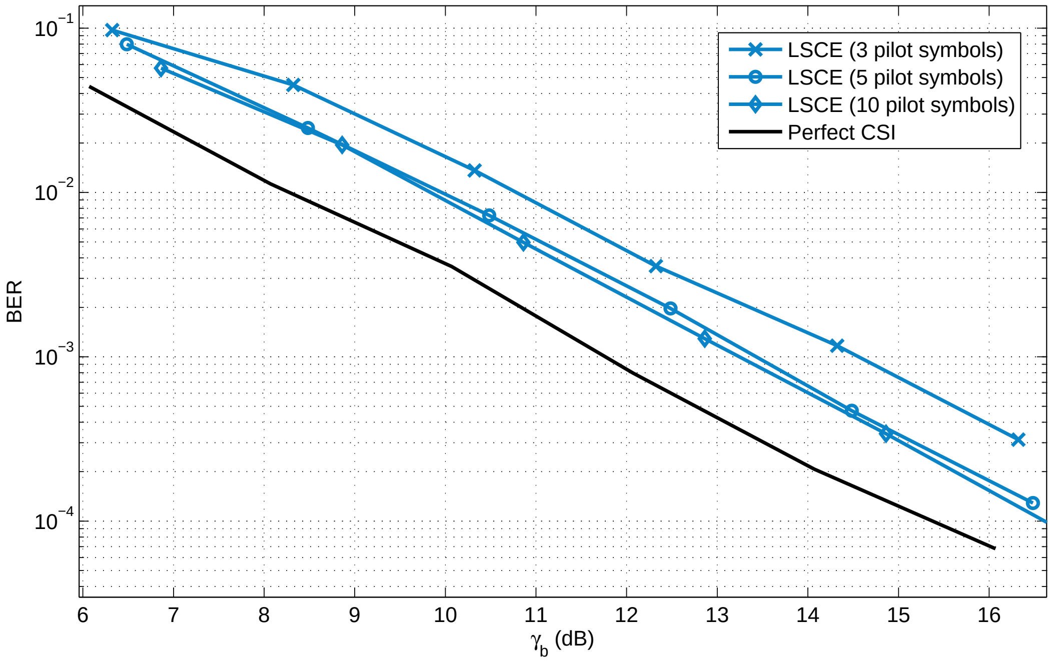

Figure 3.15: BER performance of LSCE with SCCC

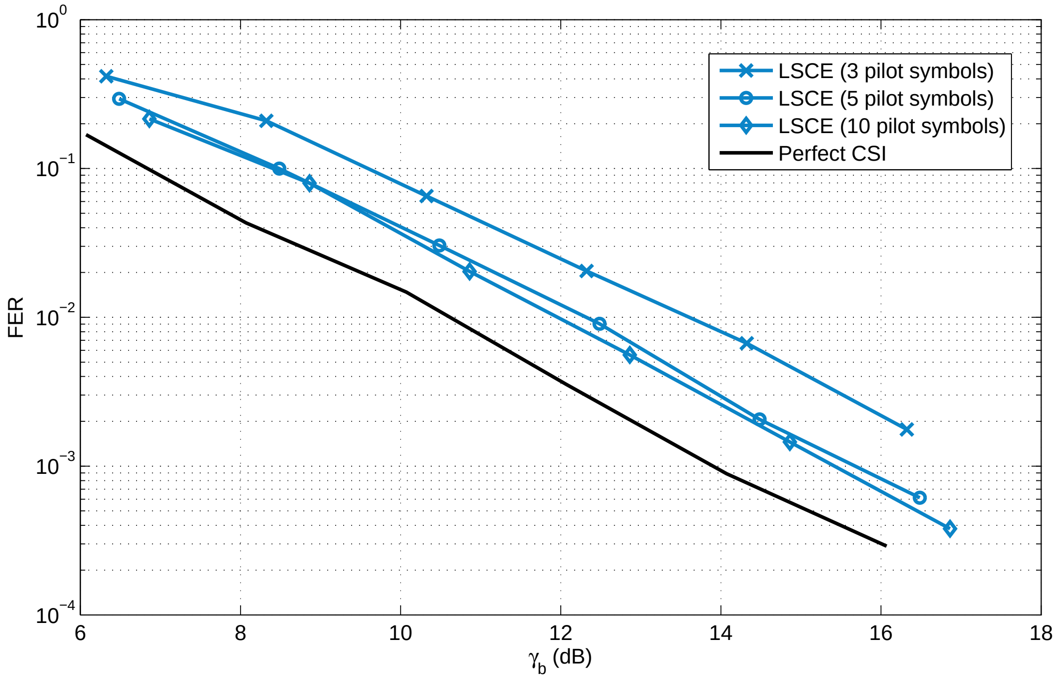

Figure 3.16: FER performance of LSCE with SCCC

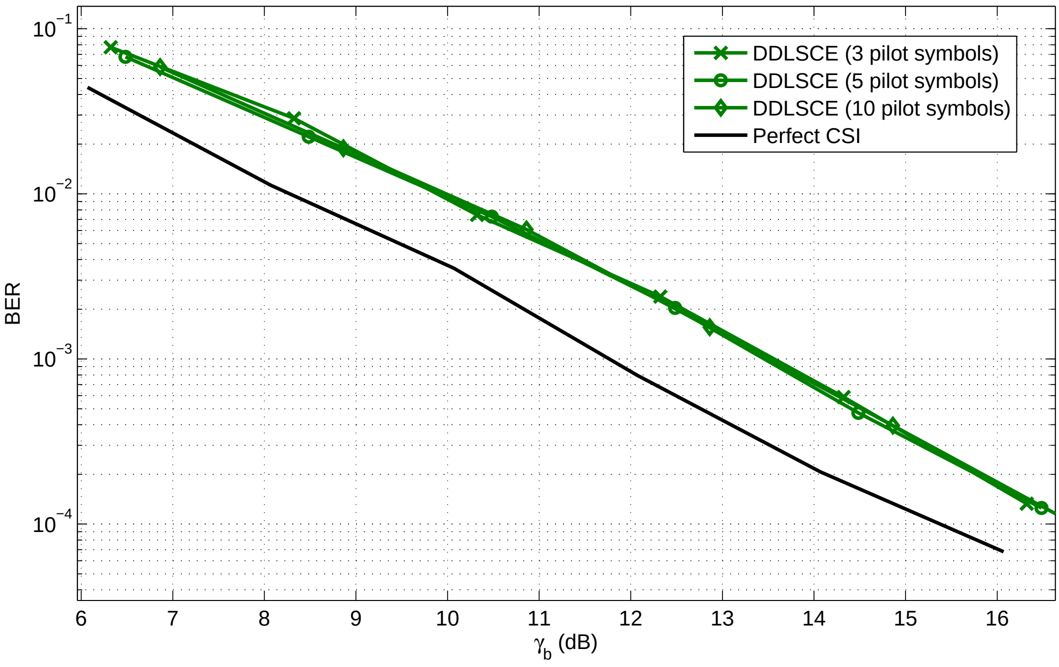

Figure 3.17: BER performance of DDLSCE with SCCC

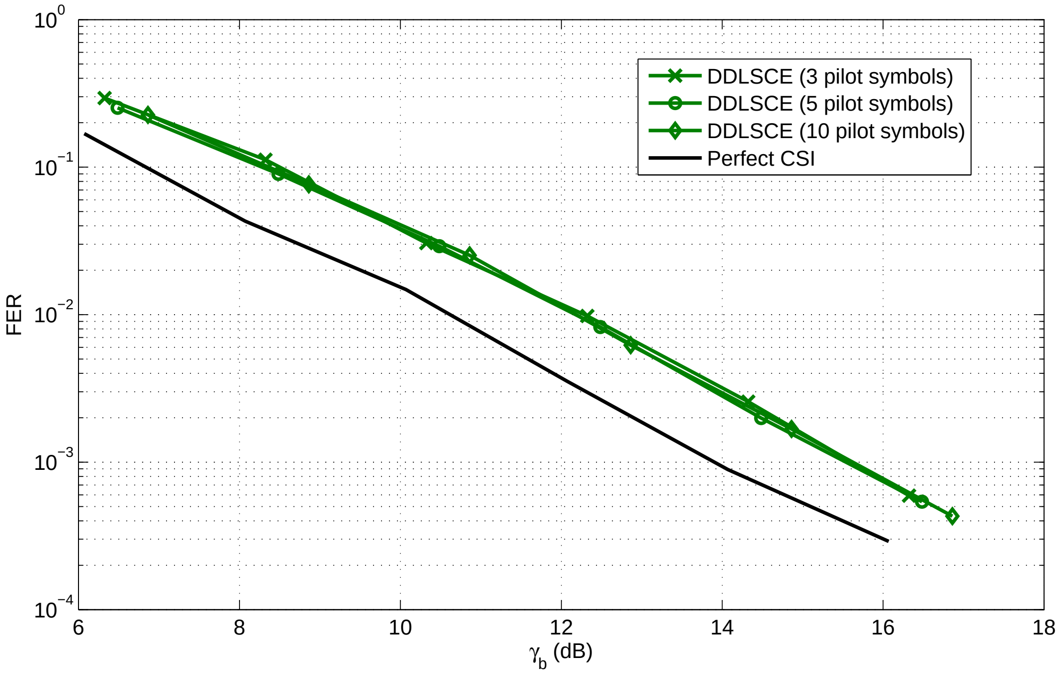

Figure 3.18: FER performance of DDLSCE with SCCC

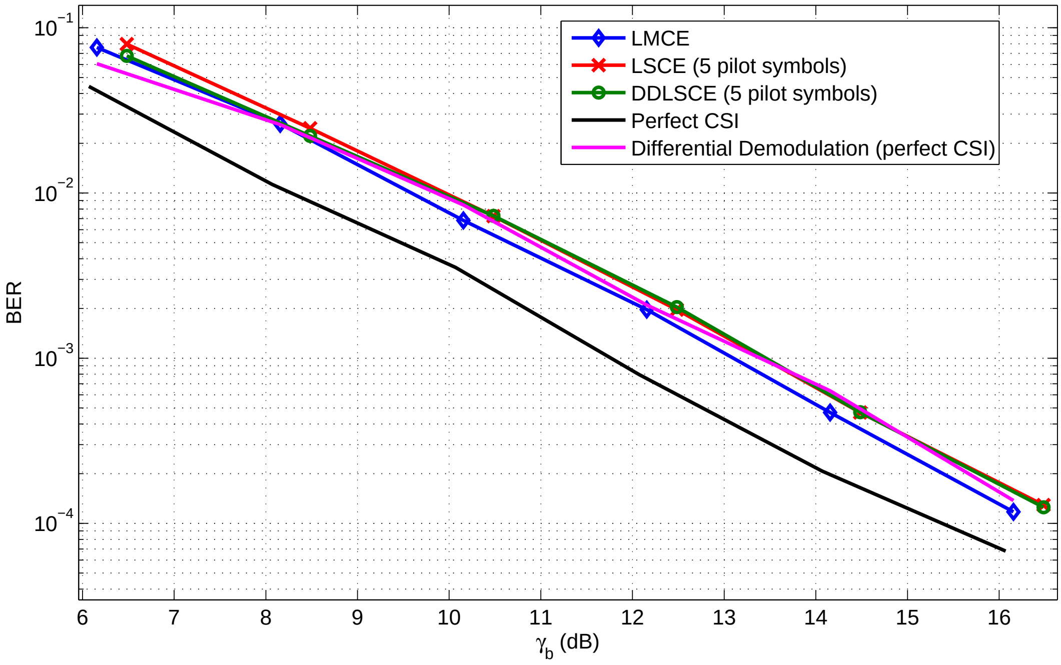

Figure 3.19: BER performance with differential SCCC

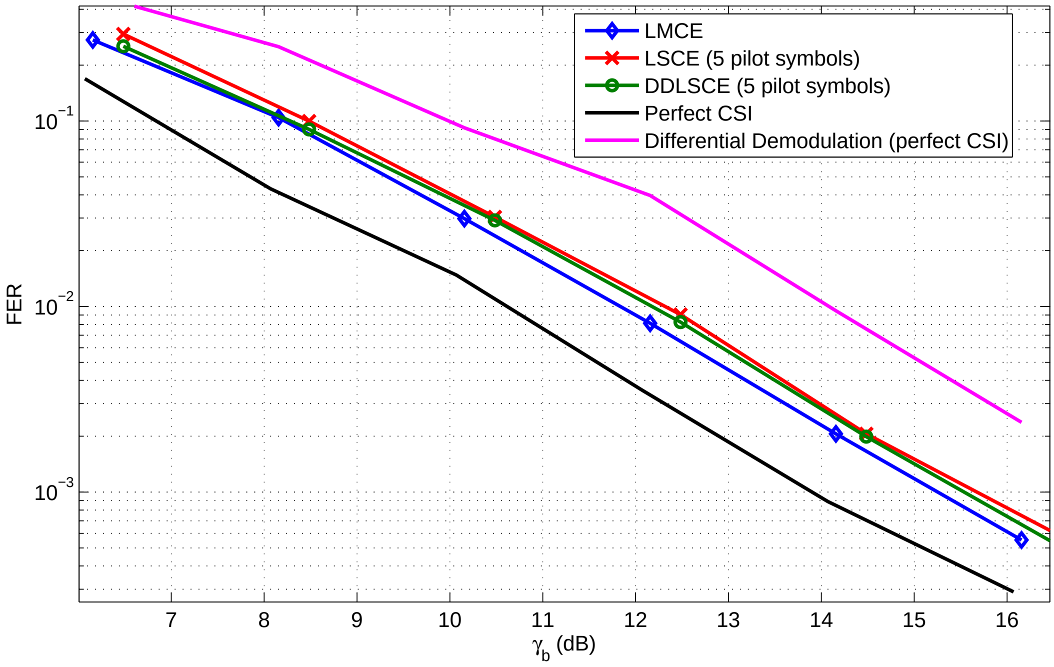

Figure 3.20: FER performance with differential SCCC

are very close to each other. LMCE, being a blind algorithm, obtains a performance as good as pilot-aided algorithms with 5 pilot symbols. It should be noted that the rate of the system using LMCE is almost 10% higher than the rate of the systems using LS or DDLS channel estimator. The algorithms are about 1dB away from perfect CSI decoding, which uses the perfect channel coefficients and noise power of each dwell for decoding. The performance of non-coherent demodulation with convolutional decoding and perfect CSI is also given in the figures. The convolutional decoder is provided with $|h|^2$ as the channel estimate since the output of the differential demodulator is given by

$$
\begin{aligned}
r_k &= y_k y_{k-1}^* && (3.25)\\
&= |h|^2 E_s a_k + h d_k n_{k-1}^* + h^* d_{k-1}^* n_k + n_k n_{k-1}^*. && (3.26)
\end{aligned}
$$

The sum of the terms $h d_k n_{k-1}^* + h^* d_{k-1}^* n_k + n_k n_{k-1}^*$ form the noise on the demodulator output. The decoder was provided with different noise estimate values to observe the effect of the noise variance knowledge on the performance, and no significant difference was observed between different noise variance values and N_0 or N_0^2. We note that the decoder does not take the correlation in noise into account.

Differential demodulation with perfect CSI decoding falls 3dB away in frame error performance from the performance of differential decoder with perfect CSI. However, the bit error performance of differential demodulation is close to the performances of other estimators. The reason of this is the difference between the error correction capabilities of the two systems. The distribution for the number of bit errors in a block solved by the two systems is given in Figures 3.21 and 3.22. An SCCC decoder either corrects the whole received block or makes a large number of bit errors. The system with a differential demodulator and a convolutional decoder generally generates a smaller number of bit errors than SCCC but more blocks contain errors.

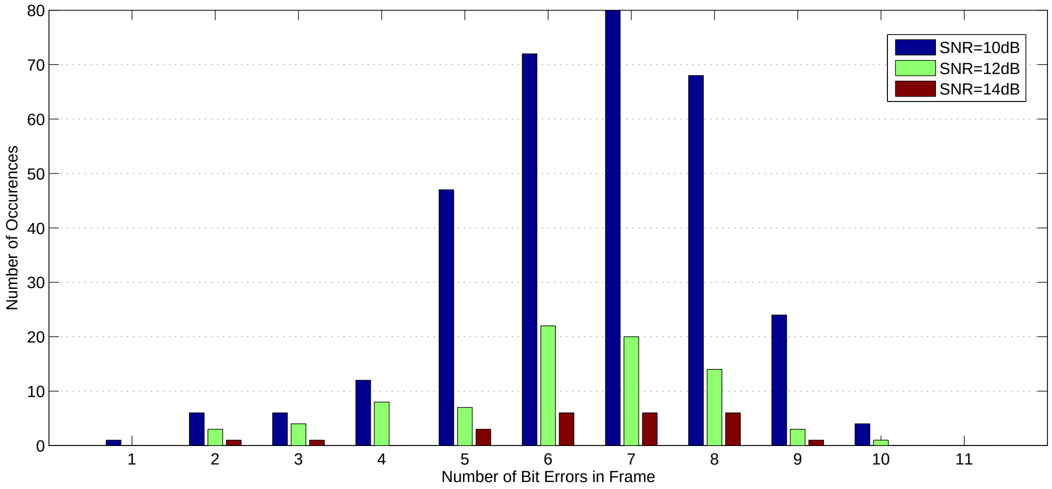

Figure 3.21: Bit error distribution for the SCCC decoder

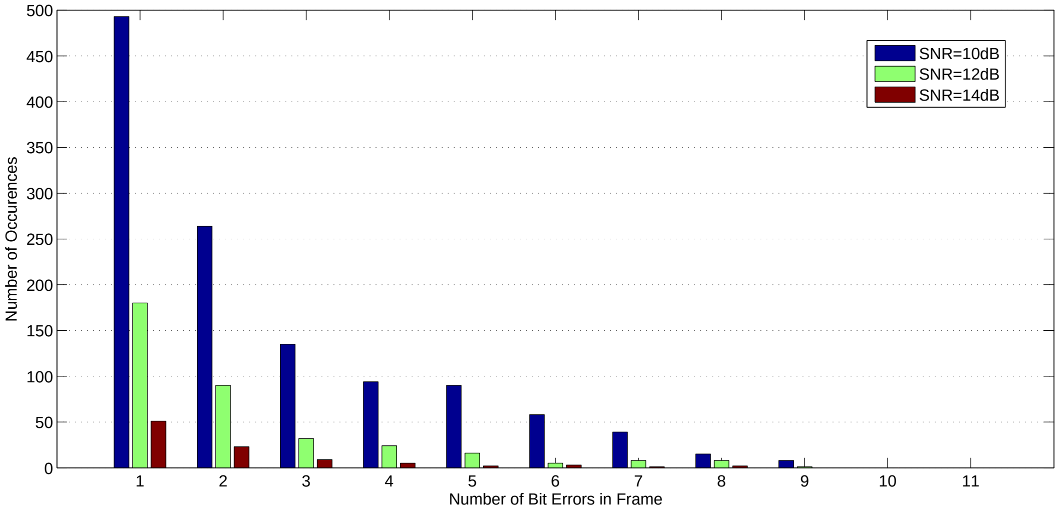

Figure 3.22: Bit error distribution for the differential demodulator and convolutional decoder pair

CHAPTER 4

PERFORMANCE UNDER JAMMING

4.1 Partial-Band Jamming Detection by LMCE

4.1.1 Detection Methods

When a communication system is jammed by a partial-band jammer, the noise power in the affected dwells increases. These dwells should be detected and treated different than the others in order to minimize the damage on the system. In literature, the output of the soft-output decoders or the channel estimation algorithms are used to detect the jammer ([9], [10], [13]). Using a similar approach, we propose to utilize LMCE for jammer detection.

As explained in Chapter 3, LMCE also finds the MSE value of the received samples with respect to its channel estimate. This MSE value is considered as the total interference power and can be used for jamming detection. However, in Figure 3.3, it was shown that the performance of LMCE is unreliable below some SNR values depending on the length of the dwell. Since LMCE finds estimates which minimize the overall error, the estimations are not the ones closest to the actual parameters for low SNR values but the ones that minimize the overall error. Therefore, the MSE calculated using the channel estimate may not represent the actual interference power at low SNR values.

Similar to the assumptions in Chapter 3, the thermal noise power, the jammer power and the channel gain are assumed to be unknown at the receiver. Consider a case when there is only noise in the received signal, and the total average power of the received signal is normalized to a constant level at the receiver for each dwell. Even if there exists no detectable signal in the environment, in other words the jammer is much stronger than the received signal power, LMCE still finds a channel estimate and an MSE for the received samples. Figure 4.1 shows the values and the distribution of the MSE values calculated by LMCE when there is only additive white Gaussian noise as the input. The average input power is normalized to 1 to ensure that MSE values are independent of the noise power.

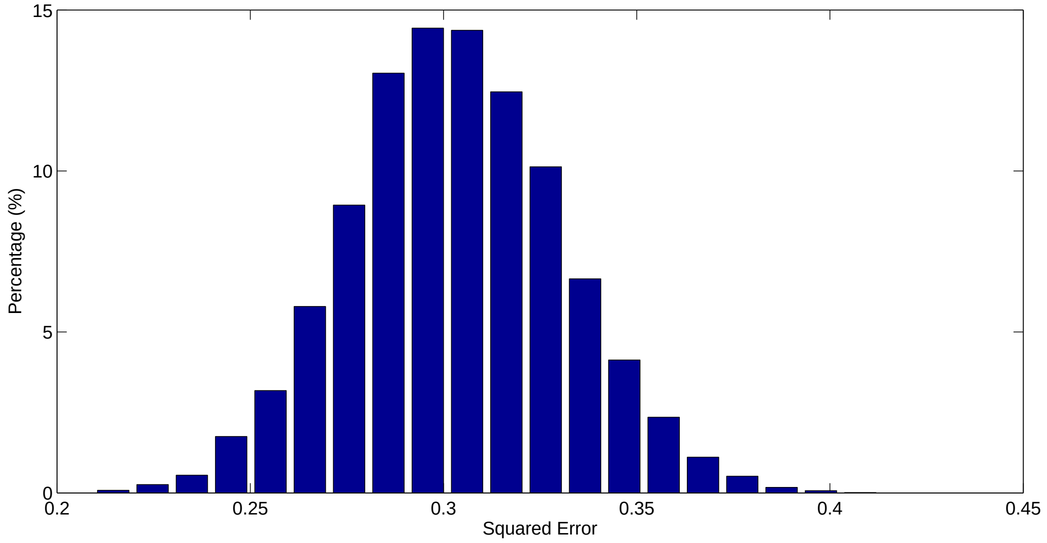

Figure 4.1: MSE calculated by LMCE for noise only

To detect partial-band jamming, the MSE of each dwell can be compared with a threshold value, and the dwells failing the test are then considered as jammed. The threshold value should be chosen according to both signal and noise powers. Since neither of these parameters are known, the signal power is normalized to a constant at the receiver input. When this normalization is applied, there is no need to have a priori knowledge of the noise and channel powers, since the dwells with high SNR will always yield small MSE values compared to the ones with low SNR, even if they have the same thermal noise power. The dwells yielding MSE values larger than the threshold are either jammed or affected by fading so that the channel estimate error is large. Therefore, not only the jammed dwells but also the dwells with low SNR's are detected by the threshold test.

Once the dwells with large MSE's are detected, the decoder has to take some precautions in order to improve the performance. The information extracted from jammed dwells have less reliability than information from clean dwells. Therefore, the decoder has to take this reliability difference into consideration during the decoding procedure. [23] and [11] propose a decoding strategy for the jammed dwells. In [23], hard decisions are calculated for the received pilot symbols in a dwell and they are compared to the known values of the pilot symbols. If an error occurs, the SNR level is declared as insufficient (or the dwell may be jammed by a jammer), and the rest of the dwell is multiplied by a constant $0 \leq \lambda \leq 1$. In [11], λ is calculated using demodulator outputs. When $\lambda = 0$, the dwell is completely erased and when $\lambda = 1$, the dwell is passed to the decoder with the same reliability as other dwells. The parameter λ can be chosen as a design parameter or a function of the demodulator and soft decoder outputs. The results in [23] show that using a constant $\lambda = 0.5$ provide a performance very close to using adaptive λ.

The method in [23] can be used with LMCE in a similar way. Jamming is detected in a completely blind way using the Lloyd-Max algorithm, unlike the pilot-aided approach in [23]. When jamming is detected, the jammed dwell can be weighted by λ to reduce the reliability at the input of the soft-input soft-output convolutional decoder, which means that the output of the differential decoder for the jammed dwells are multiplied by λ.

4.1.2 Numerical Results

Figure 4.2 and Figure 4.3 show the frame error performances of LMCE for different threshold values under partial-band jammers with $\rho = 0.1$ and $\rho = 0.3$, respectively. The power of the input signal is normalized to 1 in each dwell. λ is chosen as constant and equal to 0.5. The average *signal-to-jammer power* ratio, SJR, is defined as

$$\gamma_j = \frac{E\left\{|h|^2\right\} E_s}{N_j} \cdot \frac{N+L}{N} \cdot \frac{1}{Rlog_2 M} \tag{4.1}$$

and average SNR $\gamma_s = 20$dB in the figure.

The results in Figures 4.2 and 4.3 suggest that the best performance is obtained for the threshold value of 0.28. Even at very low γ_j levels, the systems do not see errors in every frame (FER$<$ 1) since the partial-band jammer does not jam all dwells in the frames. It is also observed that the performance of the decoder with perfect CSI also increases when a threshold value of 2dB is used, which yields the best result among other threshold values. This effect of scaling the extrinsic information on the performance of iterative decoders has been investigated in [22]. Therefore, scaling the extrinsic information from dwells with low SNR values

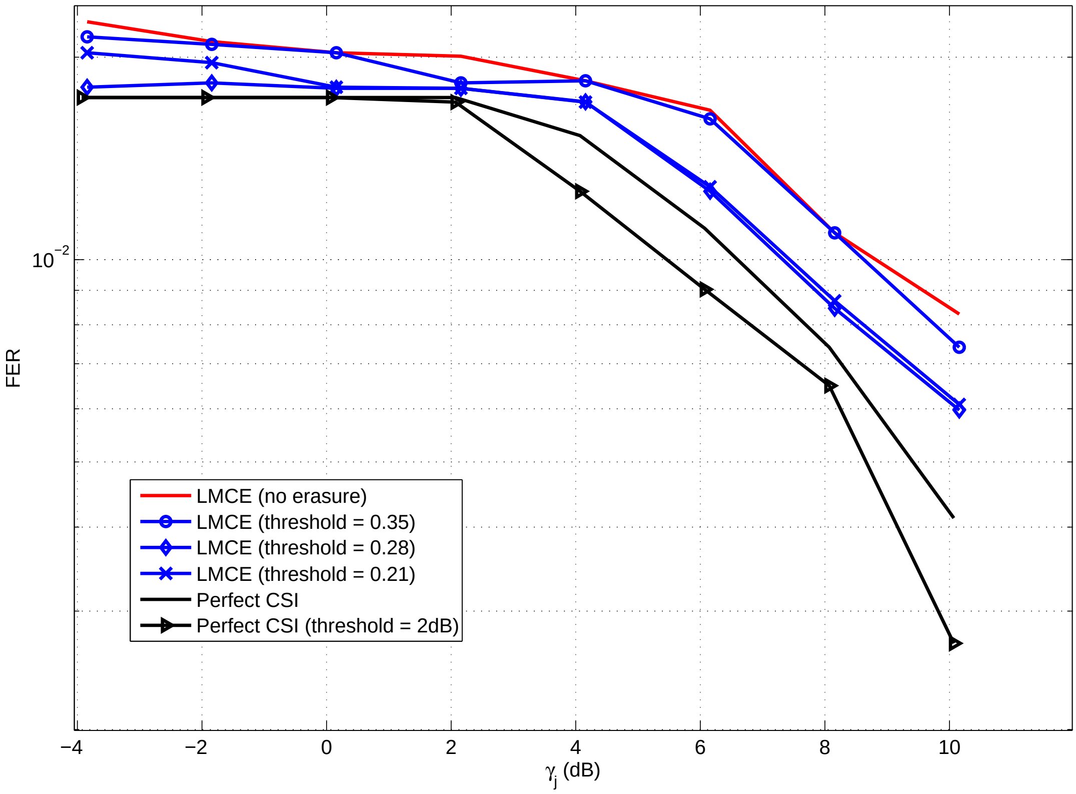

Figure 4.2: LMCE performance under partial-band jamming with ρ=0.1

improve the performance even for the perfect CSI case. One may note here that using an MSE threshold may cause the dwells with small instantaneous SNR values to be considered as jammed even if they are not. Since these dwells have small instantaneous SNR values, the estimates obtained for them are also not as reliable as the estimates of the other dwells. Therefore, a smaller weighting given to such dwells increases the performance.

The approach explained above can also be used with LS and DDLS channel estimation methods. The thresholds for these methods are obtained through exhaustive search by simulations. The frame and bit error performances of Lloyd-Max, LS and DDLS channel estimators for $\rho = 0.1$ and $\rho = 0.3$ are given in Figures 4.4 - 4.7. The channel characteristics and frame structures are the same as the ones used for LMCE in Figures 4.2 and 4.3, except that the pilot length is 5 symbols.

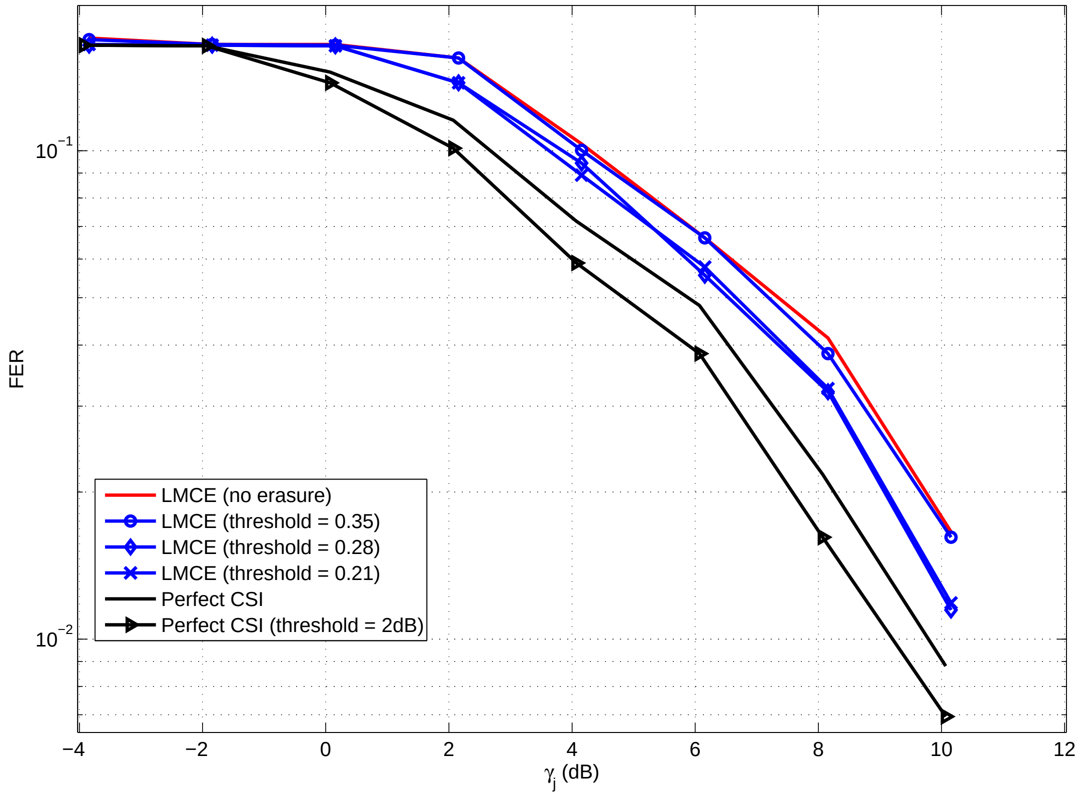

Figure 4.3: LMCE performance under partial-band jamming with ρ=0.3

As seen from the Figures 4.4 - 4.7, performances of different estimation methods are close to each other, similar to the results obtained in Chapter 3. There is about 2dB performance loss in comparison to the perfect CSI decoding, where all channel and jamming parameters are perfectly known. It should be noted that, the approach of scaling the jammed symbols at the output of the differential decoder is also used in perfect CSI decoding. All three estimation methods are affected in the same way from the partial-band jamming, since the jammer jams pilots and data symbols together, which simply reduces the effective SNR of the jammed dwell for each estimation method.

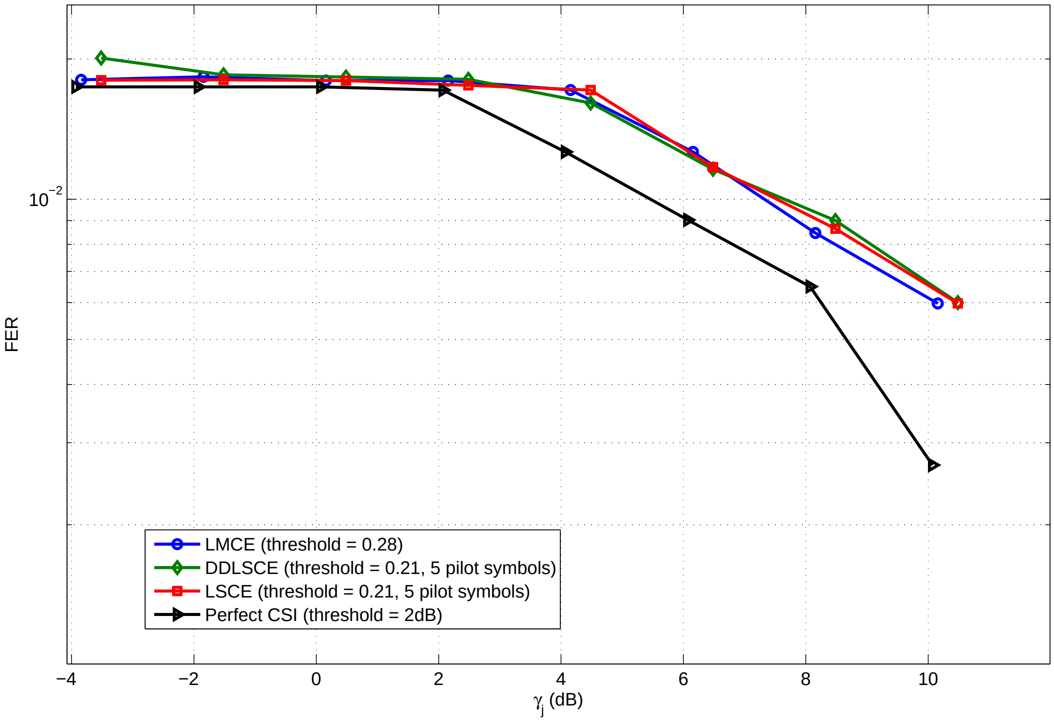

Figure 4.4: FER performance of estimators under partial-band jamming with ρ=0.1

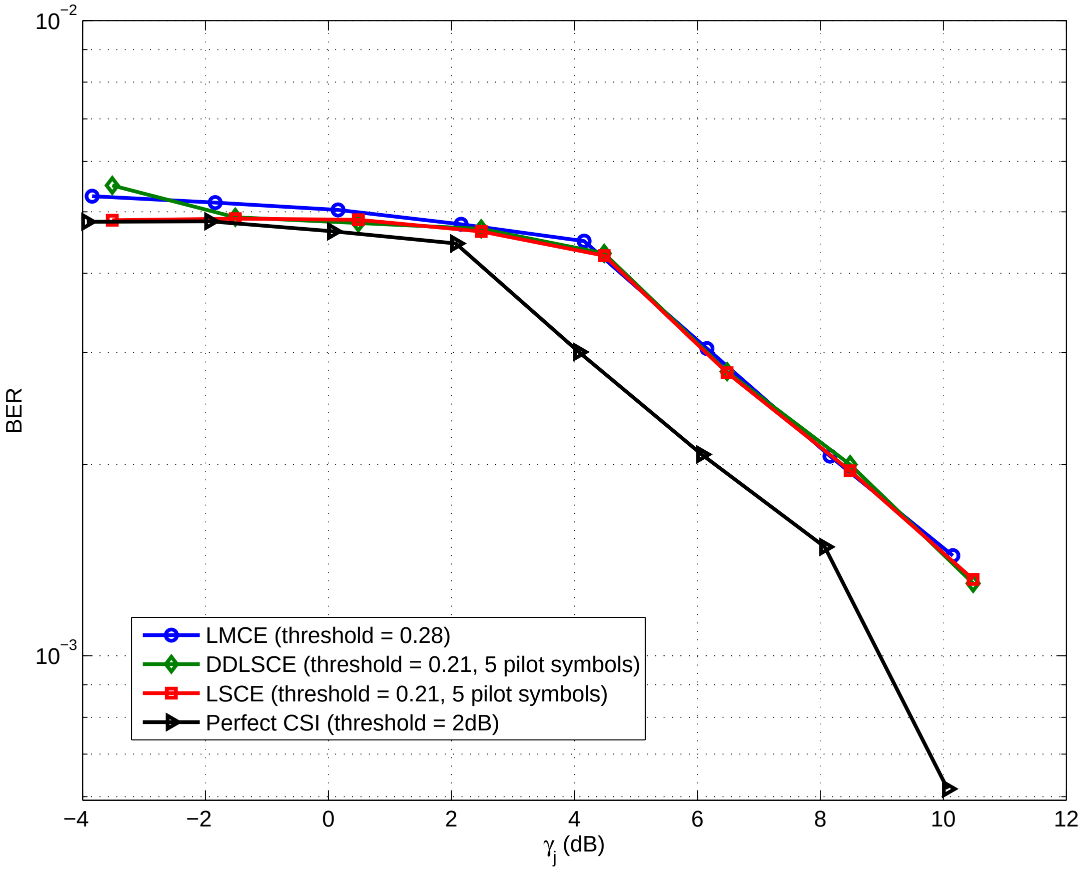

Figure 4.5: BER performance of estimators under partial-band jamming with ρ=0.1

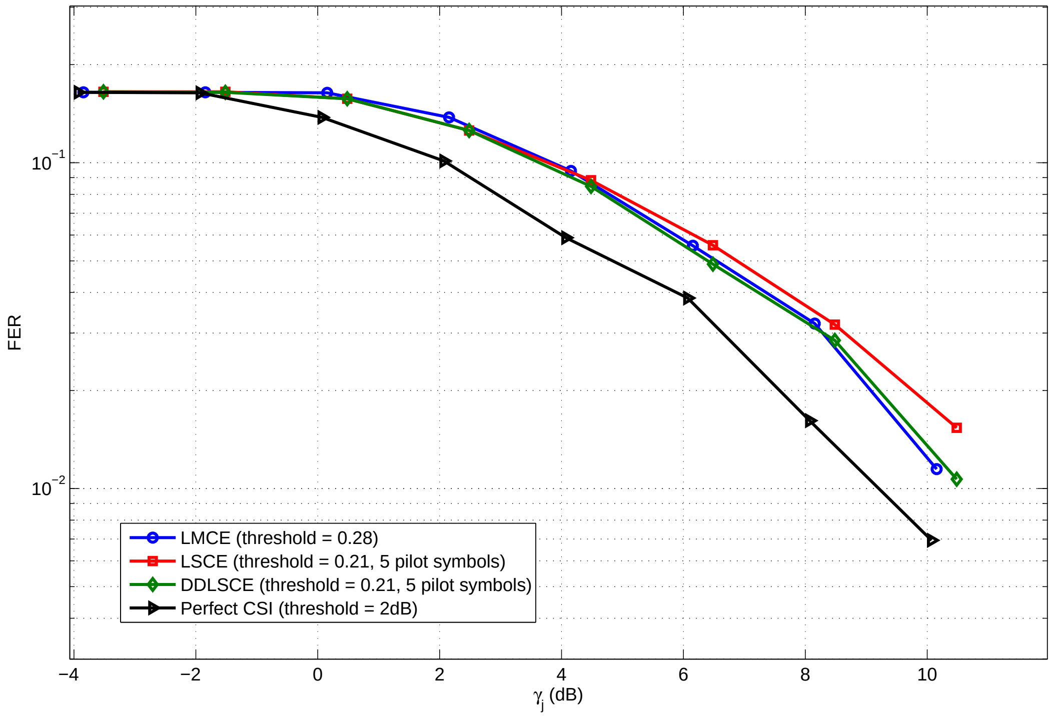

Figure 4.6: FER performance of estimators under partial-band jamming with ρ=0.3

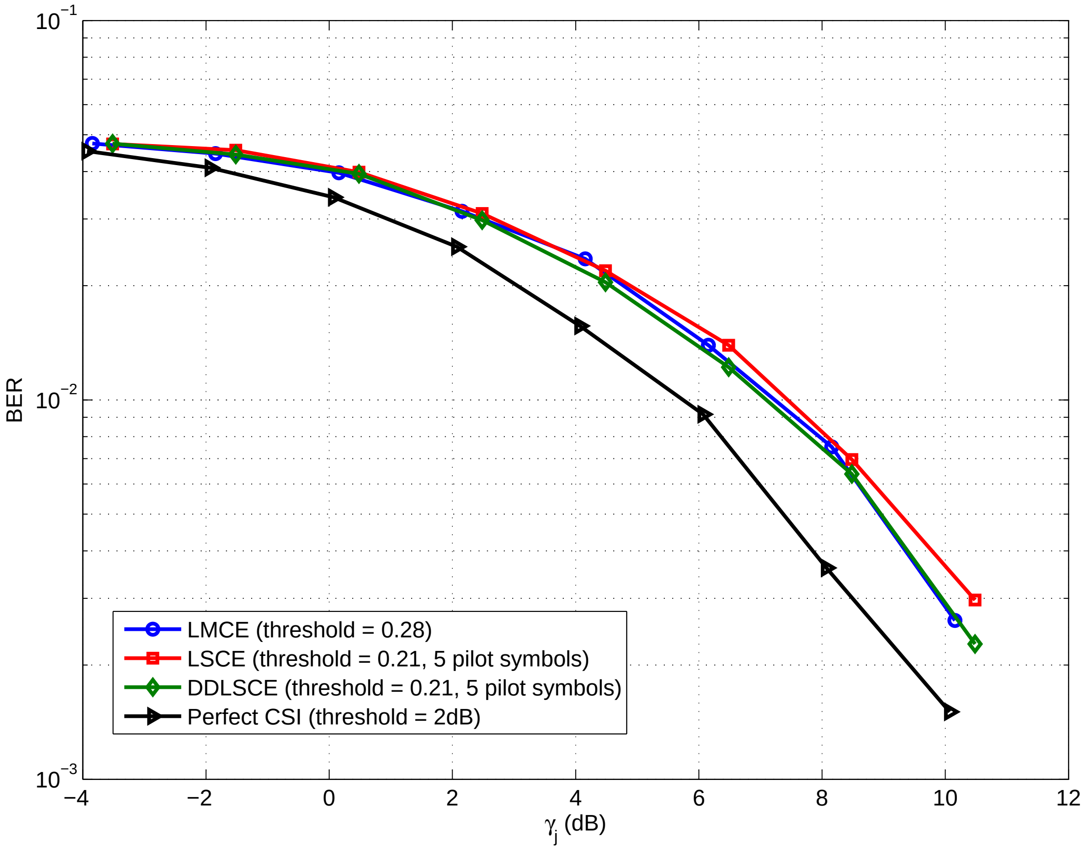

Figure 4.7: BER performance of estimators under partial-band jamming with ρ=0.3

4.2 Partial-Band Jamming Detection by LMCE Under No Jamming

It is stated in Section 4.1 that the threshold test used to detect the dwells jammed by the partial-band jammer also causes the dwells that are not jammed but have small instantaneous SNR's to be considered as jammed. It is shown that the threshold test improve the error rate performance, since the estimates obtained for the dwells failing the test are less reliable than the ones obtained from other dwells, whether they are actually jammed or not. Therefore, the threshold test designed for partial-band detection can also be used to eliminate the dwells with small instantaneous SNR's when there is no jamming. In this section, the effects of the threshold test on the performance of the system under no jamming will be investigated. As the performance bound, the perfect CSI case is used, as in Chapter 3. A weighting operation should also be considered for the perfect CSI case for low SNR values, since the outputs of the differential decoder are less reliable when the instantaneous SNR is low even if the channel parameters are perfectly known.

When there is no partial-band jamming, the frame error rate performance of LMCE for different threshold values is given in Figures 4.8.

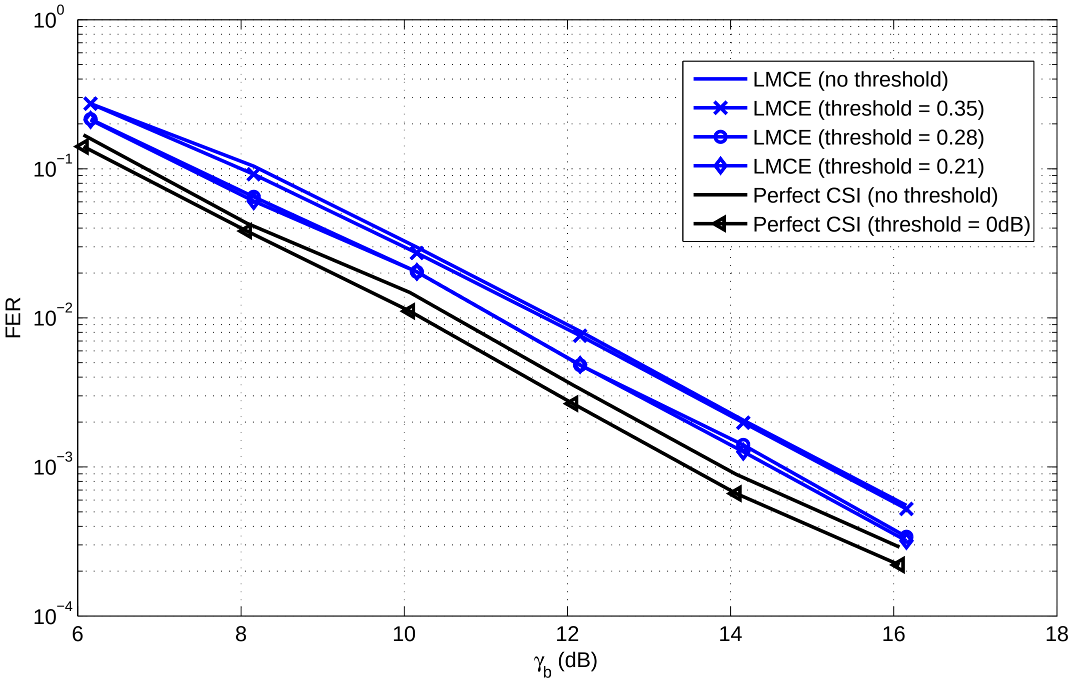

Figure 4.8: FER performance of LMCE with weighting for dwells

As it is seen from the Figure 4.8, the MSE threshold test improves the performance of the system even when there is no jamming. A similar performance enhancement is expected for DDLS and LS estimators. Figures 4.9 and 4.10 show the frame error performance of DDLS and LS estimators, respectively.

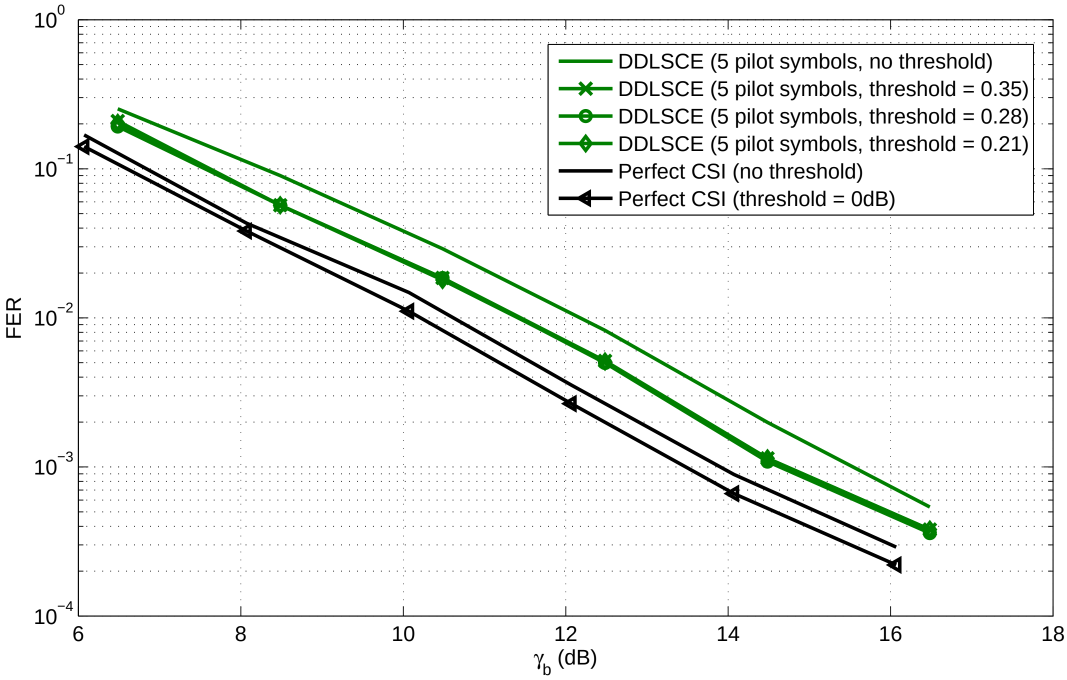

Figure 4.9: FER performance of DDLS with weighting for dwells

Finally, Figures 4.11 and 4.12 show the performances of the three estimators together. As it was seen in Chapter 3, the performances are very close to each other and about 1dB worse than the perfect CSI case.

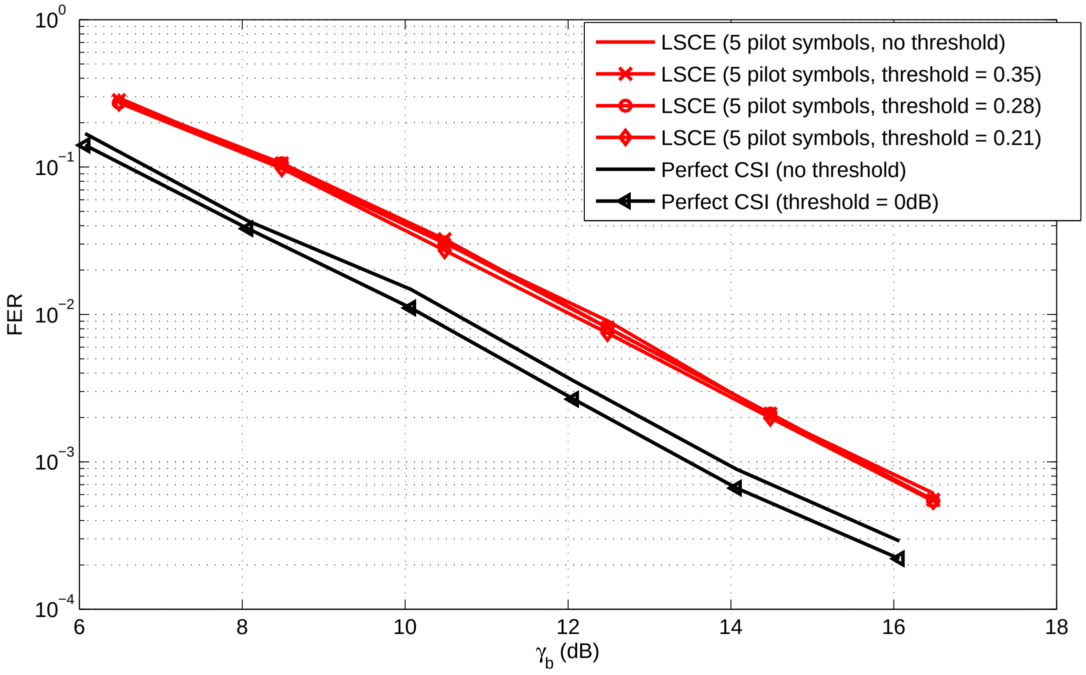

Figure 4.10: FER performance of LS with weighting for dwells

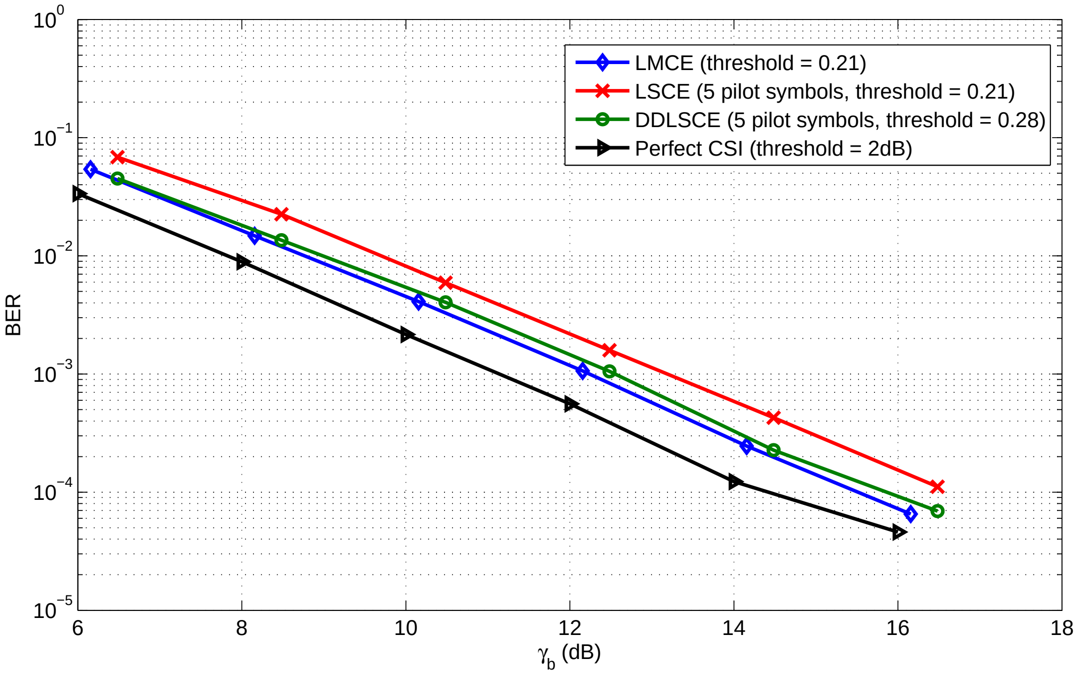

Figure 4.11: BER performance under no-jamming

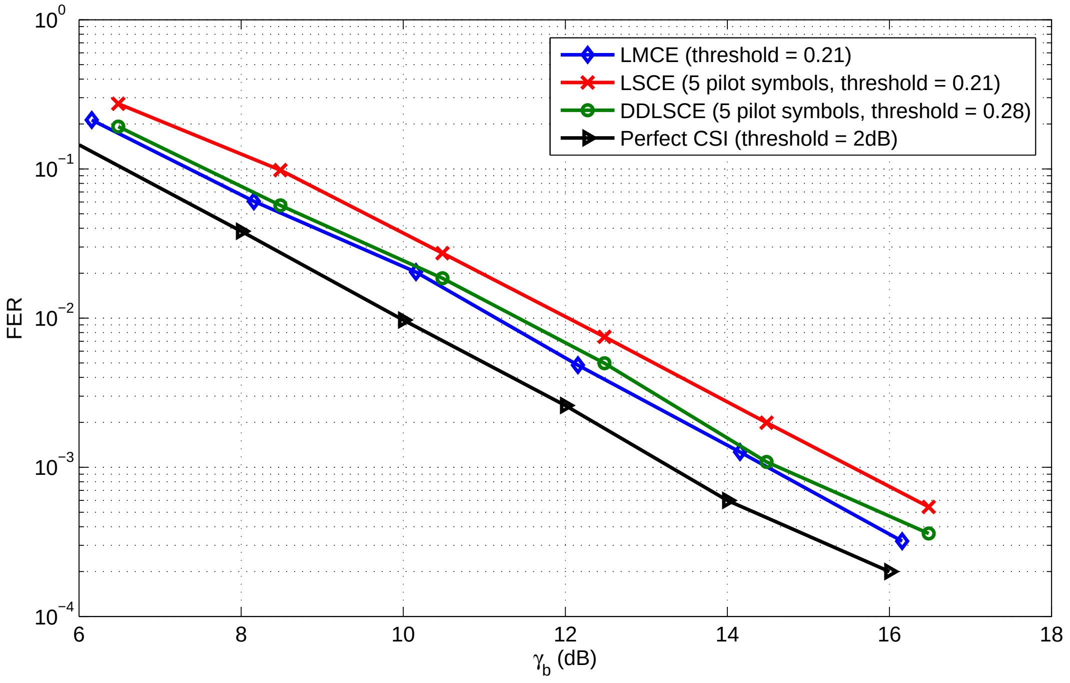

Figure 4.12: FER performance under no-jamming

4.3 Partial-Time Jamming Detection by LMCE

4.3.1 Detection Methods

As in partial-band jamming, partial-time jamming should be detected in order to improve the error performance. In partial-time jamming, the received symbols are seperated into two groups depending on whether they are jammed or not. The purpose of this seperation is to eliminate the jammed symbols and use the rest of the symbols to obtain a good estimate of the channel. The main challenge here is to decide on which symbols are jammed. Since the noise level and the transition probabilities of the jammer are not known to the receiver, determining the jammed symbols is not a straightforward process. Using LMCE, two methods are proposed for partial-time jammer detection. Method 1 uses only LMCE outputs and Method 2 uses LMCE outputs and soft-outputs of the differential decoder to seperate the jammed symbols from others.

In Method 1, detection is based on the errors of the received symbols with respect to the estimated constellation points. The detection procedure starts with operating LMCE on the whole dwell to obtain an estimate of the received constellation. Using this estimate and the distances of each received sample to the nearest constellation point, an instantaneous SNR is calculated for each sample. The samples are then sorted according to their instantaneous SNR values. The κ_0 samples with the smallest SNR are considered as jammed by the jammer, where κ_0 is an arbitrarily chosen number which is small with respect to the dwell length. An average SNR value γ_s^0 is calculated over the samples excluding κ_0 samples with the smallest SNR values. Using the constellation estimate once again, the instantaneous SNR values of all samples are calculated and the samples having an SNR smaller than $\frac{\gamma_s^0}{2}$ are also considered as jammed. LMCE is operated once again to obtain a second constellation estimate, this time using only the samples that are not jammed. Using this estimate, the process to determine the jammed samples is repeated and the final decision about the locations of the jammed samples is given. The differential decoder is provided with the second estimate for decoding. The noise estimates for jammed and clean symbol groups are calculated by the MSE of the related groups with respect to the second constellation estimate. It should be noted that the number of symbols that are marked as jammed may be more than κ_0 but cannot be less than it when the algorithm performs these operations. This will not affect the performance of the system severely if there are no jammed symbols in the dwell. The MSE value of these symbols will be close to MSE of clean symbols as they are given to the differential decoder and scaling extrinsic information of κ_0 symbols at the differential decoder output do not affect the convolutional decoder performance since κ_0 is small compared to the dwell length. This can be shown using the partial-band jammer case as an example. Consider a system operating under partial-band jamming with the detection algorithm described in Section 4.1. In addition to this detection algorithm, 10 symbols with the smallest SNR's are considered as jammed in each dwell and the noise variance estimates given to the differential decoder and differential decoder outputs for these symbols are treated as described by Method 1, whether the dwell is jammed or not. The frame error performance of this system employing LMCE is compared with the system described in Section 4.1 in Figure 4.13. Figure 4.13 proves that considering a small number (compared to the dwell length) of symbols with small SNR's as jammed in clean dwells does not affect the performance of decoding severely.

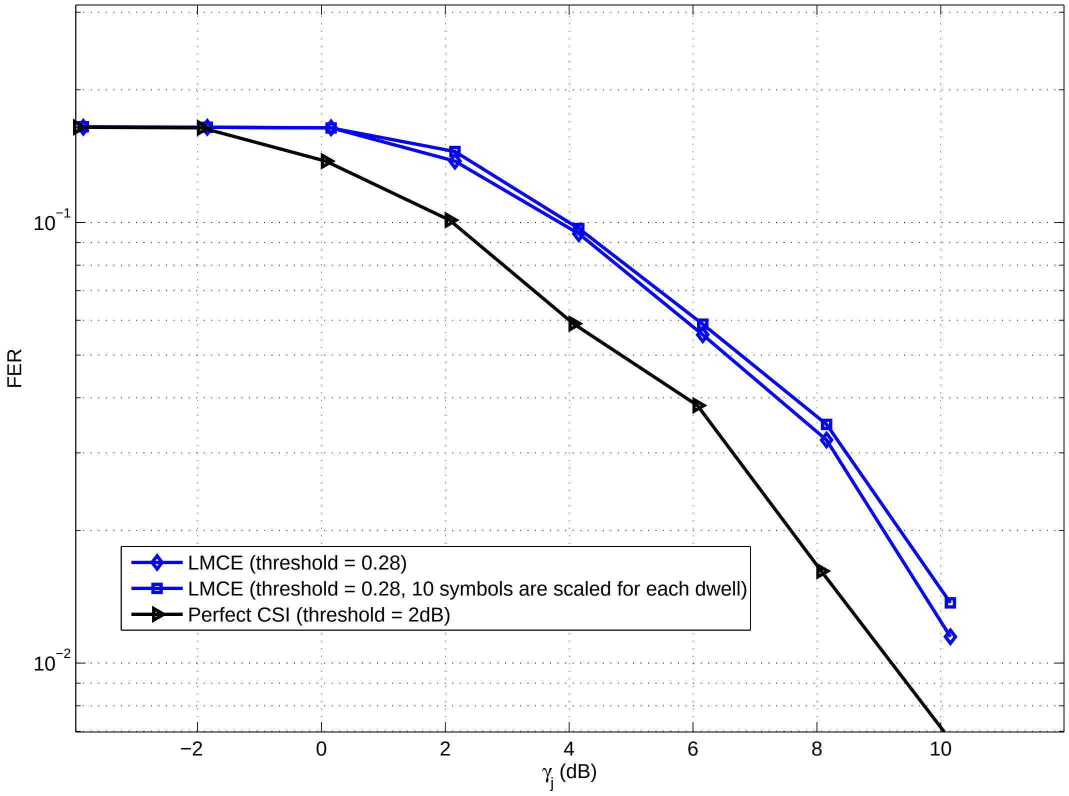

Figure 4.13: FER performance of the system scaling constant number of symbols in jammed and clean dwells

Method 2 operates in a similar way to Method 1. The constellation estimates are calculated as explained for Method 1. Using the second estimate and the noise estimates, the differential decoder decodes the dwell and calculates the bit LLR's as the output. The forward, backward and transition metrics of the decoder can be used to obtain soft estimates for the received samples. In [24], the differential decoder outputs are used to calculate soft estimates of the symbols in the codeword. The calculation is done by finding the expected value of the symbols using the probabilities at the output of the decoder. The soft estimates of the received symbols can be used to interpret symbol reliabilities by observing their Euclidien distances to the constellation points.

The probabilities to be used in the calculation of the soft estimates are obtained from the differential decoder, since the decoder calculates the probability of being in each state for each symbol. Using Max-Log-MAP decoder metrics [19], the probability of being at state m at time k, or the probability of a_k to be the constellation point α_q is

$$P(a_k = \alpha_q) = max_{p \in \{0,1,\ldots,M-1\}} \{A_{k-1}(p) + \Gamma_k(p,q) + B_k(q)\} \tag{4.2}$$

The soft estimates of the symbols can be calculated as

$$s_k = \sum_{i=0}^{M-1} \alpha_i P(a_k = \alpha_i) \tag{4.3}$$

The Euclidien distances between the expected values and the constellation points closest to them gives a metric about the symbol reliabilities:

$$r_k = min_{\alpha \in A} |s_k - \alpha|^2 \tag{4.4}$$

The amplitude of the metric is inversely proportional to the reliability of the decoder outputs for a symbol. If $r_k = 0$ for the received symbol y_k, it means that the transmitted bit u_k is perfectly extracted by decoding. The mean of the reliabilities $\mu_{r,j}$ of the symbols in the jammed group is calculated and the symbols in the clean group that have $r_k \geq \tau_1 \mu_{r,j}$ are also placed among the jammed symbols, where $0 < \tau_1 < 1$. Then, the received symbols are given to the differential decoder with the channel and proper noise variance estimates for the jammed and clean symbols.

4.3.2 Numerical Results

The parameters used to define the characteristics of a partial-time jammer are defined in Chapter 2. $E\{T_1\}$, which is the expected number of consecutive jammed symbols, is taken to be 15. Solving (2.11) for P_{11} yields $P_{11} = \frac{14}{15}$.

$$\overline{\pi}^{n+1} = \overline{\pi}^n P$$

$$[0.7\ 0.3] = [0.7\ 0.3]\begin{bmatrix} P_{00} & 1 - P_{00} \\ \frac{1}{15} & \frac{14}{15} \end{bmatrix}$$

Solving (4.5) for P_{00} yields $P_{00} = 0.97143$. The frame error performance of the LMCE decoder under partial-time jamming with these parameters is given in Figures 4.14. γ_s is chosen as 15dB and τ_1 for Method 2 is 0.8.

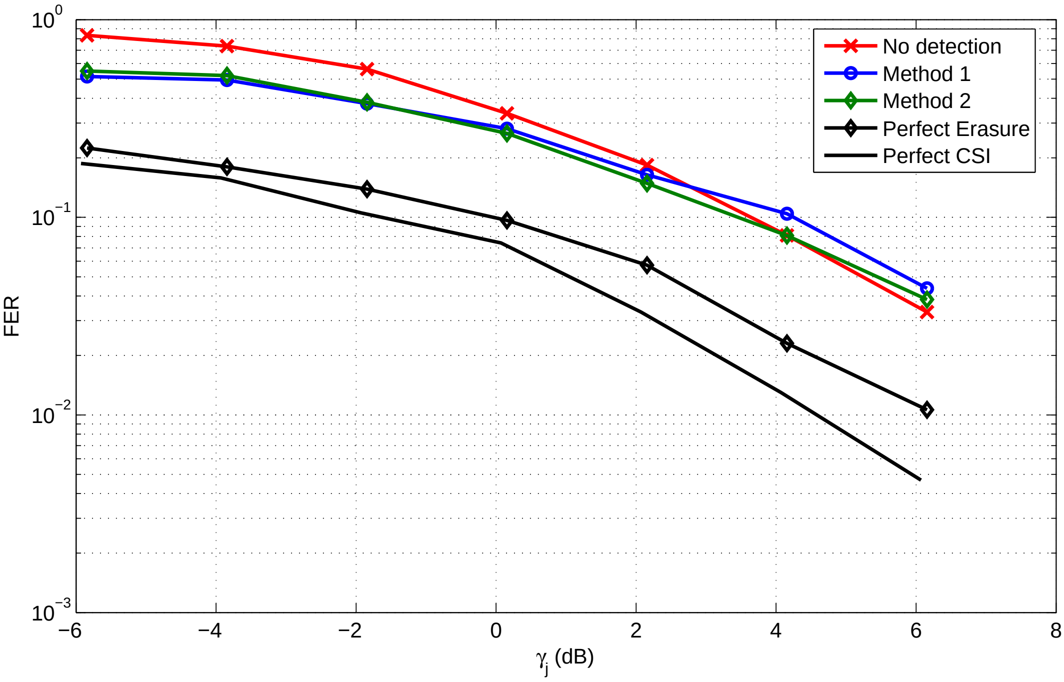

Figure 4.14: FER performance of LMCE under partial-time jamming

As seen from Figure 4.14, there is a 5dB performance difference between Methods 1 and 2 and perfect CSI decoding. The performance curve labeled as “Perfect Erasure” is the performance that is obtained by the decoder with LMCE if the locations of jammed symbols are perfectly known. The performance of Perfect Erasure is 4dB better than Methods 1 and 2. To understand the reason of this performance difference, the decoders using Method 1 and 2 are provided with the perfect knowledge of the locations of the jammed symbols. However, they only use this information to use the calculated large MSE value as the noise variance for jammed symbols. This does not affect the constellation or noise variance estimates but makes the differential decoder aware of the non-reliabilities of the jammed symbols. The frame error performance of this case is given in Figure 4.15.

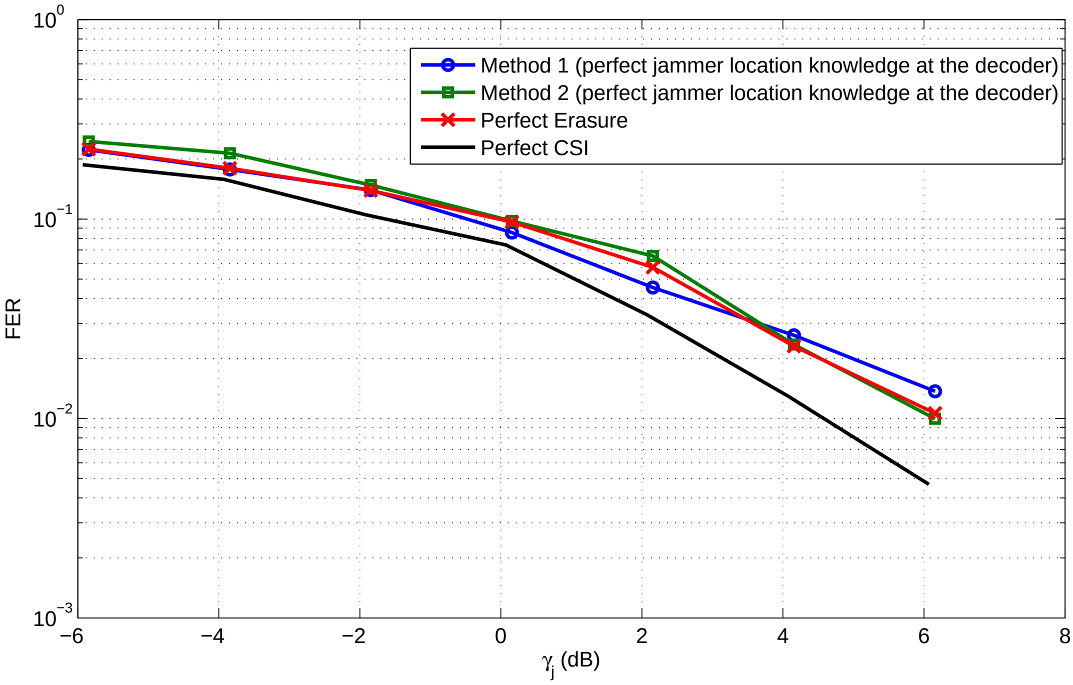

Figure 4.15: FER performance of LMCE with perfect symbol jam knowledge under partial-time jamming

The performance of Method 1 and Method 2 are almost identical to perfect erasure case if the differential decoder is given the correct noise variances. Therefore, the main reason of performance loss when these methods are used is the failure of the differential decoder when jammed symbols are given to the decoder with the same reliability as other symbols. Method 1 and Method 2 cannot provide the decoder with the perfect locations of the jammed symbols. One major reason for the miss of a jammed symbol (a jammed symbol considered as clean) is the location of the received sample on the constellation diagram. If a jammed symbol appears close to a constellation point different from its original one, this symbol is not considered as jammed in either of these methods. It is intuitively expected that a decoder can determine a non-reliable symbol with such condition from the previous and next state informations. However, a differential decoder cannot detect this since each state of the differential encoder trellis is accessible from each state. This property of the trellis is the reason for the failure of the decoder. When a symbol jammed by the jammer falls closer to a constellation point different from the true constellation point, the decoder calculates the state corresponding to the closer point as the most probable state. If the noise variance of that symbol is given equal to the symbols that are not jammed, the decoder output for that symbol is calculated as reliable as other symbols. Even if the next symbol is not jammed by the jammer and very reliable, the decoder cannot correct the erroneous decision it makes for the current jammed symbol. The forward, backward and transition metrics calculated for the jammed symbols also cause the error to propagate in the trellis. Therefore, a stronger code which can correct the erronous decisions of the differential decoder is required as the inner code, such as the turbo code[19]. Turbo coding is a class of concatenated codes, which employs two convolutional encoders concatenated in parallel. When the inner convolutional code is replaced by a turbo code, which uses the same convolutional encoders that is used in SCCC, the frame error performance is given in Figure 4.16.

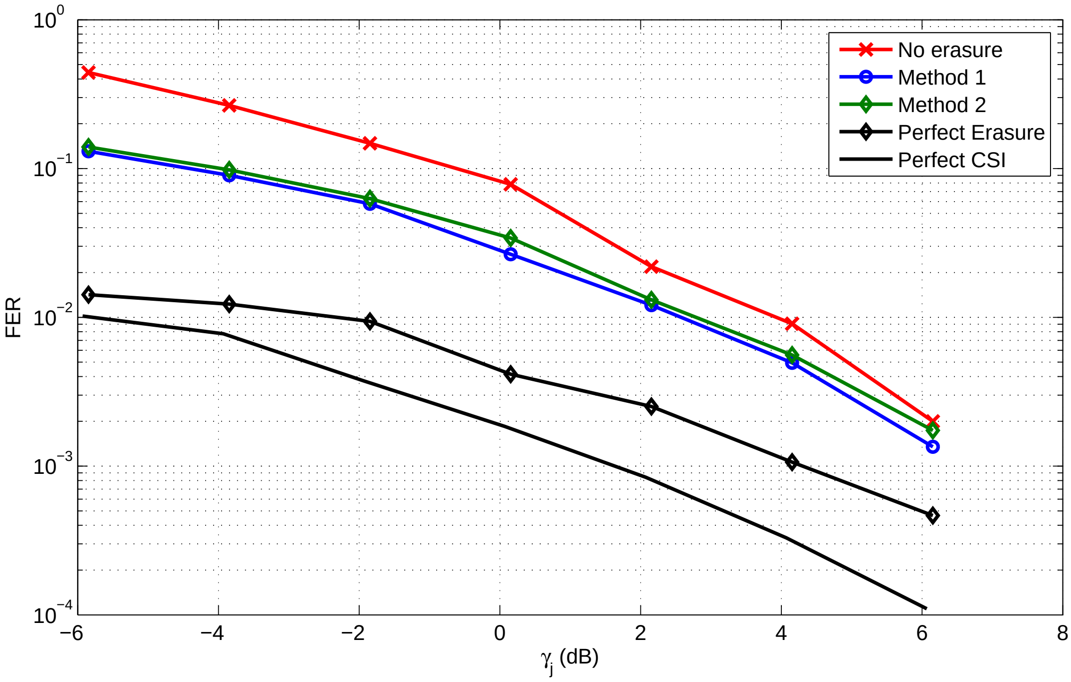

Figure 4.16: FER performance of LMCE with Turbo coding under partial-time jamming

From Figure 4.16, it is seen that also turbo decoder cannot correct the errors made by the differential decoder. However, Method 1 and Method 2 still provide a performance gain for high jammer power values. Moreover, since LMCE is a blind algorithm, it is not affected by partial-time jammers as much as pilot-based systems, since they fail to operate when the pilot symbols are jammed. This can be observed from Figure 4.17, which shows the performances of Lloyd-Max, LS, and DDLS channel estimators when they apply no detection and have perfect knowledge of the locations jammed symbols.

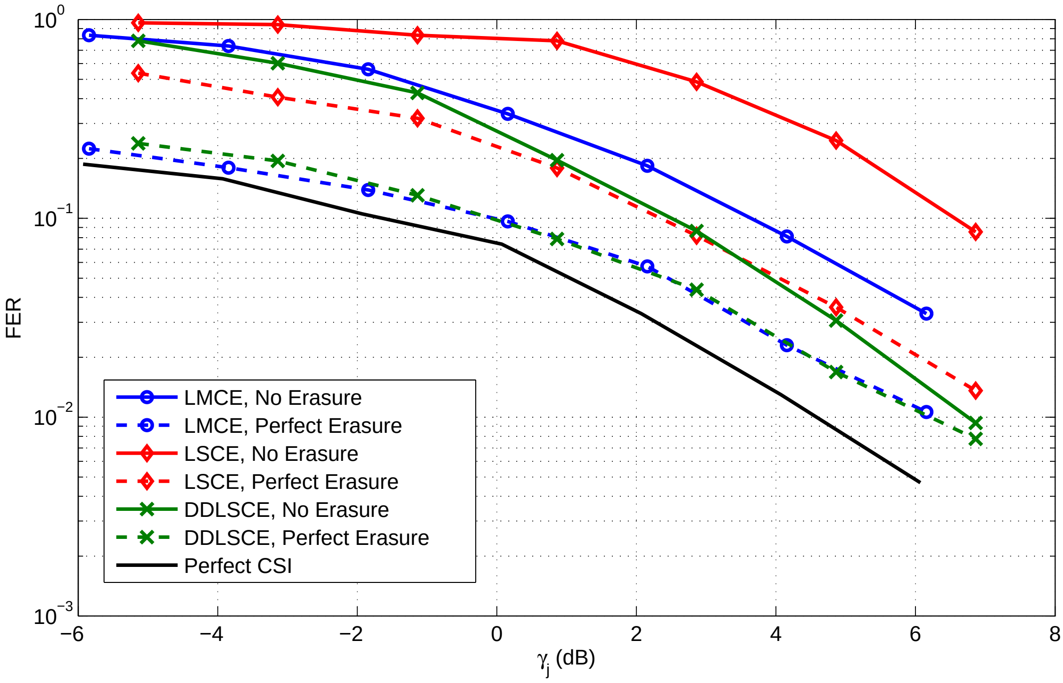

Figure 4.17: FER performance of the estimators with no erasure and perfect erasure

As it is seen from Figure 4.17, the LS estimator yields the worst performance among the three esimators. This is because of the vulnerability of LS estimation to jamming when the pilot symbols are jammed. LMCE and DDLSCE perform better than LSCE since their estimations are calculated from all of the received symbols. Therefore, even if better algorithms than Method 1 and Method 2 are developed for partial-time jamming, the performance of the LS estimator will never reach the performance of the Lloyd-Max or DDLS estimator.

CHAPTER 5

CONCLUSION

In this work, a blind channel estimation method, Lloyd-Max channel estimation (LMCE), is proposed for frequency hopping systems under narrowband channels. The performance of LMCE is compared with the pilot-based LS algorithm and the decision directed LS. The effect of channel coding has also been investigated by the use of convolutional codes and serially concatenated convolutional codes. The SCCC employs a differential code as the inner code to avoid the phase ambiguity in the channel estimations and a convolutional code as the outer code. When used with SCCC, the results show that LMCE performs slightly better than pilot-based estimators with a number of pilots that corresponds to approximately 10% of dwell length. The estimator performances are compared to the performance with the perfect channel-state information and it is observed that the estimator performances are about 1dB worse than perfect CSI case when SCCC code is used. Moreover, the performance of the channel estimators with SCCC are also compared with non-coherent differential demodulation and convolutional decoding with perfect CSI which are outperformed by a 3dB margin with regard to the frame error rate.

In addition to channel estimation, LMCE is used for partial-band and partial-time jamming detection with SCCC. Under partial-band jamming, a threshold test is performed on the noise variance estimates for the dwells and the dwells failing the test are considered as jammed. To avoid the bad effects of the non-reliabilities of the symbols in the jammed dwells on decoding, the differential decoder outputs of these dwells are scaled by a constant before they are used in the convolutional decoder. This method improves the performance of the systems and the estimator performances are similar to the performance results with no jamming situation; LMCE performs slightly better than the other two estimators using pilots. It is also observed for the case of no jamming that using the threshold test and the scaling method improves the performances of all estimators. This is due to the fact that low quality channel estimates and the non-reliable symbol LLR's at low SNR's affect the performance of iterative decoders using feedback. In partial-time jamming, two methods are proposed to detect the locations of the jammed symbols. One method uses the LMCE outputs only and the other uses both LMCE and decoder outputs. Even a small number of errors made in the detection of the jammed symbols affect the performance of the differential decoder severely and the error rate performances fall 4dB away from the perfect CSI case which includes the perfect knowledge of the jammed symbol locations. However, with the perfect knowledge of the jammed symbols, LMCE and the DDLS channel estimator performs better than the LS channel estimator, since the LS estimator fails to operate when the pilots symbols are jammed.

With the results obtained under no jamming and jamming situations, it can be concluded that the Lloyd-Max channel estimation algorithm performs well compared to the pilot-based LS and DDLS algorithms. LMCE removes the dependency of a system on pilots, which is especially needed for systems requiring high data rates and systems operating under jamming. LMCE can be used in systems with high frequency hopping rates and small dwell lengths, with intolerance to the rate loss resulting from the pilot symbols in a dwell, or systems which have other reasons to avoid the use of pilot symbols. In addition to the studies in this thesis, the following is a list of possible research directions with respect to the topic of this thesis:

- LMCE used under partial-time jamming with a better jammer detection algorithm.
- LMCE performance and modifications to LMCE for systems with carrier frequency offset errors.

REFERENCES

[1] M. Morelli, U. Mengali, “Carrier-frequency estimation for transmissions over selective channels,” *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1580-1589, September 2000.

[2] A. Goldsmith, *Wireless Communications*. New York: Cambridge Univ. Press, 2005.

[3] S.M. Ross, *Introduction to Probability Models*. San Diego:Academic Press, 2003.

[4] S.N. Crozier, D.D. Falconer and S.A. Mahmoud, “Least sum of squared errors (LSSE) channel estimation,” *IEE Proceedings-F*, vol. 138, no. 4, pp. 371-378, August 1991.

[5] S.P. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129-137, March 1982.

[6] K.R. Narayanan, “A serial concatenation approach to iterative demodulation and decoding,” *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 956-961, July 1999.

[7] S. Benedetto , G. Montorsi, ”Unveiling turbo codes: some results on parallel concatenated coding schemes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409-428, March 1996.

[8] S. Benedetto , D. Divsalar, G. Montorsi and F. Pollara, “Serial concatenation of interleaved codes: performance analysis, design and iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909-926, May 1998.

[9] H.E. Gamal, E. Geraniotis, “Iterative channel estimation and decoding for convolutionally coded anti-jam FH signals”, *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 321-331, February 2002.

[10] W.G. Phoel, “Iterative demodulation and decoding of frequency-hopped PSK in partial-band jamming”, *IEEE Journal on Selec. Areas Commun.*, vol. 23, no. 5, pp. 1026-1033, May 2005.

[11] M.B. Pursley, J.S. Skinner, “Adaptive coding for frequency-hop transmission in mobile ad hoc networks with partial-band interference,” *IEEE Trans. Commun.*, vol. 27, no. 3, pp. 801-811, March 2009.

[12] D. Torrieri, S. Cheng, M.C. Valenti, “Robust frequency hopping for interference and fading channels,” *IEEE Trans. Commun.*, vol. 56, no. 8, pp. 1343-1351, August 2008.

[13] J.W. Moon, T.F. Wong, J.S. Shea, “Pilot-assisted and blind joint data detection and channel estimation in partial-time jamming,” *IEEE Trans. Commun.*, vol. 54, no. 11, pp. 2092-2102, November 2006.

[14] J.G Frias, J.D. Villasenor, “Turbo decoding of Gilbert-Elliot channels,”’ *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 357-363, March 2002.

[15] M. Mushkin, I.B. David, “Capacity and coding for the Gilbert-Elliot channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277-1290, November 1989.

[16] J.G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[17] D. Divsalar, F. Pollara, “On the design of turbo codes,” *TDA Progress Report 42-123*, pp. 99-121, November 1995

[18] L.R. Bahl, J. Cocke, F. Jelinek, J.Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284-287, March 1974.

[19] T.K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. New Jersey: Wiley-Interscience, 2005.

[20] E. Malkamäki, H. Leib “Coded diversity on block-fading channels,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 771-781, March 1999

[21] P. Hoeher, J. Lodge, “Turbo DPSK: iterative differential PSK demodulation and channel decoding,” *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 837-843, June 1999

[22] J. Vogt, A. Finger, “Improving the max-log-MAP turbo decoder,” *Electronics Letters*, vol. 36, no. 23, pp. 1937-1939, November 2000.

[23] M.B. Pursley, J.S. Skinner, “Decoding strategies for turbo product codes in frequency hop wireless communications,” *Proceedings of the 2003 IEEE International Conference on Communications*, vol. 3, pp. 2963-2968, May 2003.

[24] C. Laot, A. Glavieux, J. Labat, “Turbo equalization, adaptive equalization and channel decoding jointly optimized,” *IEEE Journal on Selec. Areas Commun.*, vol. 19, no. 9, pp. 1744-1752, September 2001.