LDPC-OFDM: Channel Estimation and Power Considerations

by

Basel Nabulsi

A thesis submitted to the graduate program in
Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
April 2013

Copyright © Basel Nabulsi, 2013
Abstract

Small cells are low-powered radio access nodes that operate in licensed and unlicensed spectrum that have a range of 10 meters to 200 meters, compared to a mobile macrocell which might have a range of a few kilometres. This dissertation proposes algorithms for the enhancement of small cells installed in high speed rails. The thesis addresses two main points: the link between the small cell and the base station, and the link between the end-users and the small cell. The channel between the small cell and the base station is a fast fading channel due to the mobility of the high speed rail. The first part of the thesis proposes methods to enhance the link between the small cell and the base station using Low-Density Parity-Check codes (LDPC) for fast fading channels. The proposed uses nonuniform reconstruction methods based on the soft output log-likelihood ratio (LLR) provided by the LDPC decoder. The LLRs provide information about the location of the symbols with high probability of being correct. The grid formed under the assumption of a correlated Rayleigh channel affecting the transmitted data is highly nonuniform. Reconstruction of the channel under such assumptions is highly unstable. A signal-to-noise-ratio dependent regularization method is implemented to enhance the performance under imperfect Doppler spread estimation. The second part of the thesis proposes algorithms for the link between the end-user and the small cell. Since power efficiency is a major factor for end-users employing battery powered devices, we propose a Linear Programming (LP) algorithm for signal shaping to minimize the average transmitted power. The
other problem the thesis addresses is the minimization of Peak-to-Average Power-Ratio (PAPR) of Orthogonal Frequency Division Multiplexing (OFDM) signals. The PAPR is minimized using a set of phase shifts for the constituting subcarriers of the OFDM signal. The set of phase shifts is determined using a LP approach that minimizes the complexity when the block length is high. A real-time implementation of some of the algorithms is carried out using the TMS320C6713 Texas Instruments board. The results for fixed-point versus floating-point implementation is shown for a different number of precision bits.
Dedication

To

All free thinkers in the world
Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Ibnkahla for his guidance, patience, encouragement, and strong support during my PhD program. Dr. Ibnkahla directed me to enter the realm of information theory and signal processing. Over the past years, I have benefited and learned a lot from his creative ideas, insightful advices, and exceptional foresight. This thesis would have never been accomplished without his invaluable insights. To him also belongs the credit for proposing the research projects and attracting several sources of funding. I also would like also to thank Prof. Kim for the precious methods of reasoning I acquired in his MIMO course and comprehensive exams discussion. I owe Dr. Alajaji for his help on LDPC codes. I am also very grateful for all my colleagues at the Wireless Communications and Signal Processing Laboratory research group lead by Dr. Ibnkahla, especially Ala’a Abualkheir and Abdallah Alma’aaitah for their compassion, generosity and wisdom. Also I would like to acknowledge my dear friend Dr. Ali Massoud for his sincere suggestions and continuous support.

The scholarship and the research allowance provided by Queens university are gratefully acknowledged.

My truest and most deep thanks go to the love of my life, Veronica Poitras, who was there for me and kept me on my feet. She is a true light when all others cease to exist. Her smile made me forget all the rough days I had. I hope the angels of God will keep her safe and surround her with the love that she gave me. I would like to thank my parents for their continuous and unlimited moral support. They have been
a constant source of love, encouragement and inspiration in my entire life including this thesis work.
Statement of Originality

I herewith declare that I have produced this thesis without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such.

The thesis work was conducted from 2009 to 2012 under the supervision of Professor M. Ibnkahla at Queen’s University.

Kingston,
# Contents

Abstract i

Dedication iii

Acknowledgments iv

Statement of Originality vi

Contents vii

List of Figures x

Abbreviations xiv

Chapter 1: Introduction 1

1.1 Motivation ......................................................... 1
1.2 Organization of the thesis ......................................... 4

Chapter 2: LDPC codes 6

2.1 Introduction ....................................................... 6
2.2 Bipartite Graphs .................................................... 7
2.3 Irregular LDPC Codes ............................................... 9
2.4 Decoding Algorithms .............................................. 11
  2.4.1 The Gallager C algorithm ..................................... 12
  2.4.2 The Min-Sum Algorithm .................................... 18
2.5 Conclusion .......................................................... 20

Chapter 3: Nonuniform Sampling in Channel Estimation 21

3.1 Pilot Symbol Assisted (PSA) Channel Estimation ................. 21
3.2 Contributions ...................................................... 23
3.3 System Structure .................................................. 24
3.4 Estimation Using Nonuniform Samples ............................ 25
Chapter 4: Signal Shaping Using LDPC Codes

4.1 Introduction .................................................. 69
4.2 Notion of Shaping ............................................. 71
4.3 Measures of Performance ................................. 72
4.3.1 Ultimate Shaping Gain ................................ 73
4.3.2 Shaping Methods .......................................... 73
4.4 System Architecture ......................................... 81
4.5 Shaping using Linear Programming ..................... 84
4.6 Adaptive Linear Programming ............................ 89
4.7 Numerical Simulations ...................................... 94
4.8 Conclusion ..................................................... 97

Chapter 5: PAPR Reduction Via Linear Programming

5.1 Introduction .................................................. 98
5.2 OFDM System Model and PAPR Reduction Methods .... 101
5.3 PAPR Reduction Using Linear Programming ............ 102
5.3.1 Partial Transmit Sequences (PTS) .................... 102
5.3.2 Real Baseband OFDM PAPR reduction via LP methods ... 103
5.3.3 Simplified Reformulation ............................... 113
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4 Numerical Simulations</td>
<td>115</td>
</tr>
<tr>
<td>5.5 Conclusion</td>
<td>118</td>
</tr>
<tr>
<td>Chapter 6: Hardware Implementation</td>
<td>119</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>119</td>
</tr>
<tr>
<td>6.2 Hardware Implementation of a Transceiver System</td>
<td>120</td>
</tr>
<tr>
<td>6.3 Comparison Between Fixed-Point and Floating-Point Implementation</td>
<td>121</td>
</tr>
<tr>
<td>6.4 Conclusion</td>
<td>126</td>
</tr>
<tr>
<td>Chapter 7: Conclusion and Future Work</td>
<td>127</td>
</tr>
<tr>
<td>7.1 Conclusion</td>
<td>127</td>
</tr>
<tr>
<td>7.2 Proposed Future Work</td>
<td>130</td>
</tr>
<tr>
<td>References</td>
<td>132</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 A bipartite graph .................................................. 8
2.2 LDPC parity check matrix and its bipartite graph ................. 8
2.3 Depiction of the variable-node operation and message direction for a variable-node linked to three check-nodes ...................... 14
2.4 Depiction of the check-node operation and message direction for a check-node linked to three variable-nodes ...................... 15

3.1 System structure ................................................... 24
3.2 Receiver block diagram ............................................. 25
3.3 Distribution of BER vs. relative value of LLR ..................... 27
3.4 High LLR sequential locator ........................................ 28
3.5 (a) Prolate matrix surface plot for $m = 10$ (b) Prolate matrix surface plot for $m = 10$ (c) Normalized eigenvalues for $m = 10$ (d) Normalized eigenvalues for $m = 100$ ................................................. 34
3.6 Different curves for Tikhonov regularization (a) A particular model norm (b) A particular misfit norm ............................. 36
3.7 Graphical motion of the Steepest Descent algorithm ............... 38
3.8 Graphical motion of the Conjugate Gradient algorithm .......... 39
3.9 Nonuniform reconstruction of a bandlimited signal of 512 samples using
40 samples ................................................................. 44
3.10 MSE of SNR estimation error averaged over 1000 blocks for the pro-
posed methods with (a): $f_dT_S = 0.01$ and (b): $f_dT_S = 0.03$ ............ 50
3.11 Residual norm of the regularized Conjugate Gradient versus number
of iterations ............................................................... 57
3.12 Correlated Rayleigh random variate generation ............................. 58
3.13 Received correlated Rayleigh fading channel with AWGN ............... 59
3.14 Estimator MSE vs. Pilot spacing for different $f_dT_S$ with SNR=5dB . 60
3.15 Estimator MSE vs. $f_dT_S$ for various estimators with SNR=5dB .... 61
3.16 BER vs. $E_b/N_o$ for $f_dT_S=0.005$ and pilot separation 28 .......... 65
3.17 BER vs. $E_b/N_o$ for $f_dT_S=0.01$ and pilot separation 22 .......... 66
3.18 BER vs. $E_b/N_o$ for $f_dT_S=0.05$ and pilot separation 8 .......... 66
3.19 BER vs. $E_b/N_o$ for $f_dT_S=0.1$ and pilot separation 4 .......... 67
3.20 Number of multiplications per symbol per iteration vs. $f_dT_S$ ....... 67

4.1 (a) Equiprobable mapping of two 1D 16-ary constellation points and (c)
The corresponding joint PDF (b) Joint mapping for minimum average
energy and (d) The corresponding joint PDF .......................... 72
4.2 Huffman code with 21 Codewords ............................................ 74
4.3 Constellation for the Huffman codebook. The dashed line is for the
16-QAM constellation .................................................. 75
4.4 Shaping on regions ......................................................... 76
4.5 Shell mapping illustration .................................................. 76
4.6 Distribution of output stream for different regions ....................... 77
4.7 Nonequiprobable spacing for Gaussian shaping .......................... 78
4.8 Congruency points for 256-QAM / two 16x16 PAM .................... 79
4.9 Sign bit shaping as proposed in [17] ................................. 81
4.10 Sign-bit shaping system architecture ................................. 82
4.11 Polyhedron of a 3 active set point with a “1” in the RHS .......... 87
4.12 Induced marginal distribution of 64-point 1-D using the LP shaping over 100 iterations ................................................. 95
4.13 Shaping gain over the decoding for a 4-state sign-bit shaping scheme (Solid line), and for the proposed algorithm (dashed) .......... 96
4.14 Symbol error rate versus SNR over AWGN channel with QAM constellation of 32,64,128 and 256 points, respectively .................. 97

5.1 OFDM system transmitter ................................................. 100
5.2 PTS block diagram ....................................................... 103
5.3 Approximation of $\cos(\theta)$ and $\sin(\theta)$ in $\theta \in [0, \frac{\pi}{2}]$ .................... 105
5.4 Typical real baseband OFDM signal with no PAPR reduction ........ 106
5.5 OFDM signal after Full LP ............................................ 106
5.6 PAPR of the Full LP method and the no approximation approach for Eq.(5.15) versus .................................................. 108
5.7 PAPR for different values of $M$ for AP-PTS LP ........................ 111
5.8 Relative time consumed by Eq. (5.22) for different M .................. 111
5.9 PAPR for different M for IP-PTS LP .................................. 113
5.10 Relative time consumed by Eq. (5.27) for different M ................. 114
5.11 Performance of AP-PTS integer LP with $P=[2,6,12]$ for $M=8$ represented by the dashed lines and $M=32$ represented by the solid lines . 115
5.12 Performance Comparison of the ILP-PTS algorithm with other methods with 4 different sequences ........................................... 116
5.13 Performance Comparison of the ILP-PTS algorithm with other methods with 8 different sequences ........................................... 117
5.14 Relative time consumption for a 2048 subcarrier OFDM system . . . . . 118
6.1 Functional block diagram of the TMS320C6713 DSP . . . . . . . . . . . 121
6.2 Nonuniform LLR quantization .......................................................... 122
6.3 The tangent hyperbolic function ......................................................... 123
6.4 A view of the TMS320C6713 board with real-time processing and a controlling computer .......................................................... 124
6.5 Comparison between fixed point implementation and floating point implementation using the update in Eq. (6.1) ......................... 125
6.6 Comparison between fixed point implementation and floating point implementation using the update in Eq. (6.2) ......................... 125
6.7 BER vs. $E_b/N_0$ for $f_dT_S$=0.05 and pilot separation 8 for fixed point implementation and floating point implementation .................. 126
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACT</td>
<td>Adaptive weight Conjugate gradient Toeplitz method</td>
</tr>
<tr>
<td>ALLR</td>
<td>Absolute Log-Likelihood Ratio</td>
</tr>
<tr>
<td>AP-PTS</td>
<td>Adjacent Partition-Partial Transmit Sequences</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BICM</td>
<td>Bit Interleaved Coded Modulation</td>
</tr>
<tr>
<td>BP</td>
<td>Belief Propagation</td>
</tr>
<tr>
<td>CCDF</td>
<td>Complementary Cumulative Distribution Function</td>
</tr>
<tr>
<td>CER</td>
<td>Constellation Expansion Ratio</td>
</tr>
<tr>
<td>CG</td>
<td>Conjugate Gradient</td>
</tr>
<tr>
<td>CGM</td>
<td>Conjugate Gradient Method</td>
</tr>
<tr>
<td>conv(.)</td>
<td>convex hull</td>
</tr>
<tr>
<td>CSA</td>
<td>Cut-Search Algorithm</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DPSS</td>
<td>Discrete Prolate Spheroidal Sequences</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DVB-H</td>
<td>Digital Video Broadcasting - Handheld</td>
</tr>
</tbody>
</table>
DVB-T  Digital Video Broadcasting - Terrestrial
ETSI  European Telecommunications Standards Institute
EXIT  EXtrinsic Information Transfer charts
FFT  Fast Fourier Transform
FIR  Finite-Impulse-Response
GF  Galois Field
HIPERLAN High Performance Radio LAN
HSR  High-Speed Rail
I-component In-Phase
ICI  Inter-Carrier Interference
IDFT  Inverse Discrete Fourier Transform
IP-PTS Interleaved Partition-Prtial Transmit Sequences
ISI  Inter- Symbol Interference
ITU  International Telecommunication Union
LDPC Low-Density Parity-Check codes
LHS  Left Hand Side
LLR  Log-Likelihood Ratio
LP  Linear Programming
LSB  Least Significant Bit
MALP Modified Adaptive Linear Program
MCU  Micro-Controller Unit
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIMO</td>
<td>Multiple-Input Multiple-Output</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PA</td>
<td>Power Amplifier</td>
</tr>
<tr>
<td>PAM</td>
<td>Pulse Amplitude Modulation</td>
</tr>
<tr>
<td>PAPR</td>
<td>Peak-to-Average Power Ratio</td>
</tr>
<tr>
<td>PAR</td>
<td>Peak-to-Average energy Ratio</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PTS</td>
<td>Partial Transmit Sequences</td>
</tr>
<tr>
<td>Q-component</td>
<td>Quadrature phase</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RHS</td>
<td>Right Hand Side</td>
</tr>
<tr>
<td>SLM</td>
<td>Selective Mapping</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SP</td>
<td>Sum-Product</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Modern communication systems aim to transmit information from one point to another over a communication channel with high performance using efficiently the limited resources available. The need to transmit digital data over wireless channels when the user is moving at high speeds has become an important issue over the last years motivated by the wide proliferation of high speed public transport. The existence of high-speed users is becoming increasingly common, especially for users in high-speed rail (HSR) such as the 574.8 km/h V150 French TGV or the 431 km/h Shanghai Maglev Train. As of 2012, the maximum commercial speed was about 300 km/h for the majority of the high speed trains (Japan, China, Taiwan, Germany, Italy, and UK). The Doppler spread at such high speeds renders the channel as a fast-fading channel and requires special attention of the system designer to enable reliable wireless services under these conditions.

In the last decade, the number of smartphone users has dramatically increased with the advent of flashy gadgets and touch phones. The widespread use of such
1.1. MOTIVATION

devices means that the number of users demanding wireless high quality digital multimedia access is on the rise. This rise poses an obstacle in high speed public transportation since all the users are trying to drain the limited resources all at once.

In addition, the high mobility limits the achievable data rates, where it renders the underlying wireless communication channel as fast fading. The fast changes in the communication channel make the channel estimation part in the receiver side of tantamount importance. Nevertheless, good channel estimation performance implies high computational complexity and or time complexity. Many methods for channel estimation exist in the literature. They fall under two main categories. The first category requires pilot insertion at the transmitter. The second category employs blind/semi-blind estimation techniques at the receiver. The vast majority of pilot-assisted methods rely on uniform pilot insertion, meaning that the receiver estimates the channel based on the received pilots at the predesignated pilot positions set at the transmitter.

In recent years, a new technology based on the idea of femtocells/small cells (in this thesis, the two terms will be used interchangeably) has emerged, where a low-power cellular base station, typically designed for use in a home or small business, allows service providers to extend service coverage indoors or at the cell edge, especially where access would otherwise be limited or unavailable. This idea is currently being extended to public transportation, where a small cell is placed inside the transportation cabinet. The small cell connects the users inside the cabinet with the main base station. The communication protocol between the small cell and the base station is different than the protocol used between the small cell and the users. This placement of a small cell allows for a higher mobile data capacity. In this thesis, we
will concentrate on the case where the end-users employ Orthogonal Frequency Division Multiplexing (OFDM) for accessing the small cell, while the small cell employs advanced channel estimation methods to accommodate the underlying fast fading channel for the communication link with the base station.

In this thesis, we devise techniques that enhance the small cell performance on both ends of the link. In the first part of the thesis, we take the link between the small cell and the base station, where we propose a novel channel estimation technique based on nonuniform sampling methods using Low Density Parity Check (LDPC) codes. LDPC codes have the option of offering soft output values at the decoder. These values are used to extract high fidelity symbols. These symbols are considered pseudopilots and are dispersed irregularly in the code block. A nonuniform interpolation algorithm is implemented to estimate the underlying fading process. The estimated channel is fed back to the LDPC decoder to get better estimates and produce more reliable pseudopilots. This iterative process continues until a maximum number of iterations is reached. It is worth mentioning that other codes offer the possibility of soft-valued decoded stream, but LDPC codes were chosen because of their ease of implementation.

We propose an algorithm for signal shaping to minimize the average transmitted power that can be used by the small cell as well as by the end-users. The motivation behind using this algorithm is that the small cell has power consumption constraints as it runs on battery, and the end-users are bound by devices that run on battery. We employ another property of LDPC codes, the sparsity of the check matrix, to devise a Linear Programming (LP) method to minimize the average transmitted power.

For the link between the end-users and the small cell, we propose a scheme where
the system employs OFDM modulation. OFDM makes efficient use of the spectrum by allowing overlap, and channel equalization becomes simpler by using adaptive equalization techniques with single-carrier systems. Also, OFDM is computationally efficient by using Fast Fourier Transform (FFT) techniques to implement the modulation and demodulation functions. But OFDM suffers a major drawback: the OFDM signal has a noise like amplitude with a very large dynamic range; therefore, it requires Radio Frequency (RF) power amplifiers with a high dynamic range, which might not always be available. This noise-like amplitude is caused by what is called high peak-to-average power ratio (PAPR). We study the different methods used for OFDM peak-to-average power ratio reduction and propose a method relying on an LP formulation to reduce the PAPR of signal.

Also, we discuss an implementation of parts of the system on a DSP chip to examine the feasibility of such approach in real life. We use the Texas Instruments TMS320C6713 board. The comparison is made for fixed-point versus floating-point implementation of the LDPC decoding methods as well as the nonuniform estimation and decoding algorithm for various number of precision bits. The communication channel is implemented on the FPGA chip on the board.

1.2 Organization of the thesis

The remainder of the thesis is organized as follows: Chapter 2 gives a background about LDPC codes, their structure, and the most widely used methods for decoding. The decoding methods will be implemented in Chapter 3, Chapter 4, and Chapter 6.

In Chapter 3, we present an iterative algorithm for joint channel estimation and decoding using LDPC codes. The algorithm is explained and the the nonuniform
1.2. ORGANIZATION OF THE THESIS

interpolator is derived. A conditioning method for the interpolating algorithm is also presented and a convergence analysis of the proposed algorithm is derived.

Chapter 4 presents the work relying on LDPC block codes for minimizing the average transmitted power of modulated signals. The state-of-the-art methods are studied and the shaping method based on LP is presented. The similarities between the LP method and trellis shaping are shown.

In Chapter 5, we study the performance of PAPR reduction algorithms based on a LP formulation derived from the algorithms used in Chapter 4. Mainly, Chapter 5 presents two novel methods for PAPR reduction based on partial transmit sequences (PTS). The methods of Adjacent Partition (IP) PTS and Interleaved Partition (IP) PTS are derived and simplifications for the LP formulation are carried over.

Chapter 6 presents an implementation of the proposed system on a DSP chip. The system is implemented on the TMS320C6713 Texas Instruments board. Two different LDPC decoding methods are compared using fixed-point and floating-point implementation. The Joint channel decoder and estimator is implemented using one of the LDPC decoding methods. The channel is implemented on the FPGA chip on the board. Finally, conclusions and future work are given in Chapter 7.
Chapter 2

LDPC codes

This chapter provides a brief introduction to the basics of LDPC codes that are found in the literature. Three different decoding schemes are presented. Message-passing algorithms are discussed and illustrated in a graphical manner. The methods discussed in this chapter will be used in Chapter 3, Chapter 4, and Chapter 6, notably, the use of the LLR values and their implementation in decoding schemes.

2.1 Introduction

LDPC codes were first devised by Gallager in his PhD thesis [21]. These codes are characterized by a sparse parity-check matrix $H$. In the case of regular LDPC codes, the columns have a fixed number $d_v$ of nonzero elements and the rows have a fixed number $d_c$ of nonzero elements [2]. For this regular $(d_v, d_c)$ LDPC code, we denote by $N$ the number of columns, i.e., $N$ is the length of the codeword. Let $M < N$ denote the number of rows of the parity check matrix. It is assumed that $H$ is full rank, i.e., Rank$(H) = M$ [43]. Let $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ be the transmitted codeword. At the
receiver, the decoder verifies the operation

\[ H\mathbf{x}^T = 0 \]  

(2.1)

where \( \mathbf{0} \) is the all zero vector. If this operation is carried out at the receiver and yields a zero, then the receiver declares the message free of error. There is also the probability of an undetected error, caused by a codeword that satisfies the constraint. It is clear that the number of independent columns in \( H \) equals \( K = N - M \) and is also the number of information bits in the codeword. Since the sum of all ones in the columns equals the sum of ones in the rows, we get

\[ N d_v = M d_c = (N - K) d_c \]  

(2.2)

so that

\[ \frac{K}{N} = 1 - \frac{d_v}{d_c}. \]  

(2.3)

The term \( K/N \) is called the code rate, which is the average number of information bits per codeword binary symbol [43]. The Matrix \( H \) can be related to a bipartite graph. A bipartite graph is a graph whose vertices can be divided into two disjoint sets \( U \) and \( V \) such that every edge connects a vertex in \( U \) to one in \( V \); that is, \( U \) and \( V \) are independent sets as shown in Figure 2.1

### 2.2 Bipartite Graphs

In coding terminology, a variable-node is associated with a column of \( H \), and a check-node is associated with a row of \( H \). The bipartite graph associated with the parity
check matrix is shown in Figure 2.2. A node in the graph is associated with each row and each column, an edge is present between the nodes if and only if there exists a “one” in the intersection of the corresponding row and column of the nodes. This graph is bipartite because the edges connect between two disjoint sets (Figure 2.2) of checks nodes represented by squares- and variable-nodes- represented by circles.
Any even length path will end up in the same set and any odd length path will end up in the opposite set. A loop or a cycle is a closed path emanating from one node and returning to the same node, which indicates that it is of even length. Since the maximum number of connections between any two nodes is one, then the minimum loop length is 4. The girth of a graph is the length of the shortest loop. [21, 43, 20].

2.3 Irregular LDPC Codes

The design of irregular LDPC codes, ones that have a nonuniform number of ones in each column/row gives more degrees of freedom and flexibility to the system, since the potential of such codes depends on the maximum girth of the corresponding parity check matrix [41, 42]. The degree distribution of an irregular LDPC code is given by the polynomial [41, 42]

\[
(\lambda(x),\rho(x)) = \left(\sum_{i=1}^{\infty} \lambda_i x^{i-1}, \sum_{i=1}^{\infty} \rho_i x^{i-1}\right)
\]  

(2.4)

where \(\lambda_i\) denotes the fraction of the graph branches connected to degree-\(i\) variable-nodes and \(\rho_j\) denotes the fraction of the graph branches connected to degree-\(j\) check-nodes. The degree polynomial \(\lambda(x)\) is referred to as the variable-node degree distribution. Similarly, the degree polynomial \(\rho(x)\) is referred to as the check-node degree distribution [20, 50].
Since these polynomials represent distributions, the following constraints apply:

\[
0 \leq \rho_j \leq 1 \quad j \geq 1 \\
0 \leq \lambda_i \leq 1 \quad i \geq 1 \\
\sum_{j=1}^{\infty} \rho_j = 1 \\
\sum_{i=1}^{\infty} \lambda_i = 1
\]  \tag{2.5}

If a parity check matrix has a total of \( \ell \) nonzero elements, this means that the graph has a total of \( \ell \) branches. The number of degree-\( i \) variable-nodes \( v_i \) is

\[
v_i = \frac{l\lambda_i}{i}
\]  \tag{2.6}

and the number of degree-\( j \) check-nodes \( c_j \) is

\[
c_j = \frac{l\rho_j}{j}
\]  \tag{2.7}

which means that the number \( N \) of variable-nodes equals

\[
N = \sum_i v_i = l \sum_i \frac{\lambda_i}{i} = l \int_0^1 \lambda(x) \, dx
\]  \tag{2.8}

Similarly, the number \( M \) of check-nodes equals

\[
M = \sum_j c_j = l \sum_j \frac{\rho_j}{j} = l \int_0^1 \rho(x) \, dx
\]  \tag{2.9}
and the code rate $R$ is given by

$$R = \frac{K}{N} = \frac{N - M}{N} = 1 - \frac{M}{N}$$

$$= 1 - \frac{l \sum_{j} \frac{\rho_j}{\Lambda_j}}{\int \rho(x) dx} = 1 - \frac{\int \rho(x) dx}{\int \lambda(x) dx}$$

(2.10)

The performance of carefully designed LDPC codes outperforms the regular LDPC codes and in some cases it can perform close to the Shannon limit [9].

### 2.4 Decoding Algorithms

The decoding algorithms of LDPC codes are mostly probabilistic in nature and derivation. While some depend on hard decision bit-flipping algorithms, the ones that perform best are known to depend on iterative message passing of real valued reliabilities [20, 44]. Gallager [21] suggested three algorithms in his thesis. The idea behind the suggested decoding algorithms is to consider each variable-node and each check-node as a separate processor that takes information from connected processors and shares its calculations with others. This iterative fashion is easily implemented, since the LDPC parity check matrix is an instance of a Tanner graph [57]. The discussion on different algorithms for decoding LDPC codes flourished and multiple variants were suggested. In [43] Mackay reintroduced the codes and many of the relevant decoding algorithms, including the Belief Propagation (BP) algorithm. Kschischang et al. [37] present a general graph based algorithm, the sum-product (SP) algorithm, which can be useful for computing the marginalization of complex probability density functions [20].
Next, we give a summary of some of the most widely used algorithms in the literature.

### 2.4.1 The Gallager C algorithm

The likelihood of a binary random variable $y$ taking values in the set $\{0, 1\}$ is given by

$$\lambda_y = \frac{P\{y = 0\}}{P\{y = 1\}}$$  \hfill (2.11)

For practical and numerical reasons, this value is usually computed in the log domain, so that the log-likelihood ratio is expressed as

$$\Lambda_y = \log \lambda_y$$  \hfill (2.12)

The algorithm advances in two sub-iterations:

I . Each variable-node processes an output message for the check-nodes connected to it.

II . Each check-node processes an output message for the variable-nodes connected to it.

As mentioned earlier, the set of edges between the variable-nodes and check-nodes are determined by the existence of a nonzero element in the corresponding place in the parity check matrix. Next, we show the operation of the two sub-iterations. The derivation is shown for the regular LDPC case but can be easily extended to the irregular case.
Sub-iteration I:

For a variable-node with $d_v$ edges, we would like to derive the update equations for the case of a regular parity check matrix. This means that $d_v$ is constant for all the variable-nodes. Assuming independent observations of the $d_v$ edges $(\varsigma_1, \ldots, \varsigma_{d_v})$, the likelihood ratios of the probabilities of the observations are

$$\left( \frac{P(\varsigma_1 | y = 0)}{P(\varsigma_1 | y = 1)} \frac{P(\varsigma_{d_v} | y = 0)}{P(\varsigma_{d_v} | y = 1)} \right) = (\lambda_1, \ldots, \lambda_{d_v}) \tag{2.13}$$

Using Bayes rule, the probability of $y$ being zero given the $d_v$ observations is

$$P(y = 0 | \varsigma_1, \ldots, \varsigma_{d_v}) = \frac{P(\varsigma_1, \ldots, \varsigma_{d_v} | y = 0)P(y = 0)}{P(\varsigma_1, \ldots, \varsigma_{d_v})} \tag{2.14}$$

The Likelihood ratio is now given by

$$\lambda = \frac{P(y = 0 | \varsigma_1, \ldots, \varsigma_{d_v})}{P(y = 1 | \varsigma_1, \ldots, \varsigma_{d_v})}$$

$$= \frac{P(\varsigma_1, \ldots, \varsigma_{d_v} | y = 0)P(y = 0)}{P(\varsigma_1, \ldots, \varsigma_{d_v} | y = 1)P(y = 1)}$$

$$= \frac{P(y = 0)}{P(y = 1)} \prod_i \frac{P(\varsigma_i | y = 0)}{P(\varsigma_i | y = 1)}$$

$$= \frac{P(y = 0)}{P(y = 1)} \prod_i \lambda_i \tag{2.15}$$

Usually, the assumption of equiprobable inputs is assumed, i.e. $P(y = 0) = P(y = 1) = 1/2$. This gives the log-likelihood ratio

$$\Lambda = \sum_i \Lambda_i \tag{2.16}$$

where $\Lambda_i = \log \lambda_i$ [20].
2.4. DECODING ALGORITHMS

Each variable-node computes $d_v$ output messages

$$m_{j}^{v,k+1} = m_0 + \sum_{i=1, i \neq j}^{d_v} m_i^{v,k} \quad (2.17)$$

where $m_{j}^{v,k+1}$ is the $j^{th}$ output message at iteration $k + 1$, and $m_i^{v,k}$ is the $i^{th}$ input message at iteration $k$. In Figure 2.3, the message $m_0$ is the message relayed from the channel observation to the corresponding variable-node; it can also be some a priori knowledge about the corresponding bit. The summation is justified by the independence assumption mentioned earlier. The processor excludes the message coming from the node to which the message is relayed.

Sub-iteration II:

The check-node can be thought of as a constraint on the corresponding bits. The
constraint in binary codewords can be thought of as a modulo \(-2\) operation on all the connected edges with the variable-nodes. So, given the probabilities of all the \((d_c - 1)\) variable-nodes connected to the current check-node, we would like to compute the probability of their modulo \(-2\) summation being equal to zero (Figure 2.4). In other words, given \(d_c - 1\) random variables \([y_1, \ldots, y_{d_c-1}]\) with probabilities

\[
(P(y_1 = 1), \ldots, P(y_{d_c-1} = 1)) = (p_1, \ldots, p_{d_c-1})
\]  

(2.18)

we want to compute the probability \(P_0\) that their modulo \(-2\) sum is equal to zero

\[
P_0 = \sum_{y|\text{sum}(y) \text{ is even}} P(y)
\]

\[
= \sum_{y|\text{sum}(y) \text{ is even}} \prod_{k=1}^{d_c-1} p(y_k)
\]  

(2.19)
where the term \((\mathbf{y}|\sum \mathbf{y}) \text{ is even}\) means all the vectors \(\mathbf{y}\) of length \((d_c - 1)\) that sum up to an even number of ones. The multiplication in the second line is justified by the assumption of independent observations. Following [20], consider the polynomial

\[
q(t) = \alpha_0 + \alpha_1 t + \cdots + \alpha_{d_c-1} t^{d_c-1} = \prod_{j=1}^{d_c-1} (1 - p_j + p_j t)
\]  

(2.20)

where the coefficient \(\alpha_i\) is given by

\[
\alpha_i = \sum_{k_1 < k_2 < \cdots < k_i} p_{k_1} \cdots p_{k_i} \prod_{j=k_1,k_2,\ldots,k_i} (1 - p_j).
\]  

(2.21)

That means that \(\alpha_i\) equals the probability of having \(i\) ones and \((d_c - 1 - i)\) zeros. The summation of \(q(1)\) and \(q(-1)\) yields the probability of having an even sum, as can be seen from

\[
q(1) + q(-1) = \sum_{j=0}^{d_c-1} (1 + (-1)^j) \alpha_j = \sum_{k=0}^{[(d_c-1)/2]} 2\alpha_{2k}
\]  

(2.22)

which equals double the probability of having an even number of ones. \(P_0\) can be expressed as

\[
P_0 = \frac{q(1) + q(-1)}{2} = \frac{1 + \prod_{j=1}^{d_c-1} (1 - 2p_j)}{2}.
\]  

(2.23)

Since \(P_1 = 1 - P_0\), we compute the LLR

\[
\Lambda = \log \frac{P_0}{1 - P_0}
\]  

(2.24)

\[
= \log \frac{1 + \prod_{j=1}^{d_c-1} (1 - 2p_j)}{1 - \prod_{j=1}^{d_c-1} (1 - 2p_j)}
\]  

(2.25)

\[
= \log \frac{1 + \prod_{j=1}^{d_c-1} \left(1 - \frac{2}{e^{\lambda_j+1}}\right)}{1 - \prod_{j=1}^{d_c-1} \left(1 - \frac{2}{e^{\lambda_j+1}}\right)}
\]  

(2.26)
2.4. DECODING ALGORITHMS

\[ = \log \frac{1 + \prod_{j=1}^{d_c-1} \tanh (\Lambda_j/2)}{1 - \prod_{j=1}^{d_c-1} \tanh (\Lambda_j/2)} \]  
(2.27)

\[ = 2 \text{arctanh} \prod_{j=1}^{d_c-1} \tanh (\Lambda_j/2) \]  
(2.28)

where

\[ \text{arctanh} (x) = \frac{1}{2} \log \frac{1 + x}{1 - x} \]  
(2.29)

and

\[ \Lambda_j = \log \frac{1 - p_i}{p_i}. \]  
(2.30)

The check-node performs the LLR operation and feeds back this information to the intended variable-node. The check-node performs the computation

\[ m_{c,k+1}^c = 2 \text{arctanh} \prod_{i=1, i \neq j}^{d_c} \tanh \frac{m_{i,k}^c}{2} \]  
(2.31)

where \( m_{c,k+1}^c \) is the \( j^{th} \) output processed message at iteration \( k + 1 \), and \( m_{i,k}^c \) is the \( i^{th} \) input message coming from the variable-node to the check-node at iteration \( k \).

The two operations keep iterating between the corresponding variable-nodes and check-nodes. When the decoding process is terminated, every variable-node computes its own estimate of the LLR as follows

\[ m^v = m_0 + \sum_{i=1}^{d_v} m_i^v. \]  
(2.32)

The decoding estimates of the LLR can now be used to make a maximum a-posteriori probability (MAP) estimate of the corresponding bits. If the sign of the LLR is positive, then the probability of the bit being a zero is higher than its probability.
of being a one. Similarly, if the sign of the LLR is negative, then the probability of
the bit being a one is higher. The decoder can then decide, based on the sign of the
LLRs, to produce the desired estimates of the output vector.

The stopping criterion is determined by the application or designer. One possible
criterion is the decoded vector $y$ satisfying $Hy = 0$, $H$ being the parity check matrix
of the code. Another criterion is that a maximum number of iterations has been
reached. The previous iterative algorithm is optimal in the case where there are no
cycles in the graph of the code. If the graph has cycles, then some other schedule
may be more suited to minimize the probability of dependent observations.

2.4.2 The Min-Sum Algorithm

The previous algorithm relies heavily on the computation of complex nonlinear func-
tions that require a lot of memory and computational power. If one is willing to
accept a little degradation in the performance at the expense of smaller delays and
less complexity, then the resulting simplification can make the design easier and more
tenable to hardware implementation. One possible simplification comes from the use
of recursive operations in the computation of the arctanh function. The most ex-
pensive operation in terms of numerical complexity is Equation (2.31). We drop the
superscript $c$ in the following and use the vector of messages $(m_1, \ldots, m_{d-c-1})$ in what
follows, define the following

\[ m_1^* = m_1 \]
\[ m_{i+1}^* = 2 \text{arctanh} \left( \frac{m_i^*}{2} \text{tanh} \left( \frac{m_{i+1}^*}{2} \right) \right) \]  \hspace{1cm} (2.33)
and

\[ m_j^c = m_{d_c-1}^* \] (2.34)

as the last step of the recursion. The arctanh function can be simplified as

\[ 2 \text{arctanh} \left( \frac{x}{2} \right) \tanh \left( \frac{y}{2} \right) = \text{sgn}(x) \text{sgn}(y) \left( \min \{|x|, |y|\} + \log \frac{1 + e^{-|x|-|y|}}{1 + e^{-|x|-|y|}} \right) \] (2.35)

noticing that whenever \(||x| - |y|| \) is large, then

\[ \log \frac{1 + e^{-|x|-|y|}}{1 + e^{-|x|-|y|}} \] (2.36)

becomes very small. The proof of Equation (2.35) can be found in [20]. Using this identity, the approximation of the operation at the check-node becomes

\[ m_j^c = \left( \prod_{i=1, i \neq j}^{d_c} \text{sgn}(m_i^c) \right) \left( \min_{i \neq j} \{m_i^c\} \right). \] (2.37)

This yields the min-sum algorithm, which is an approximation of the Belief Propagation (BP) algorithm. For the two algorithms discussed previously, the choice of \( m_0 \) was not specified. In the case of binary input with AWGN, this initial value can be estimated as

\[ m_0 = \log \frac{\frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(y_k-1)^2}{2\sigma_w^2}}}{\frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(y_k+1)^2}{2\sigma_w^2}}} = \log e^{\frac{4y_k}{\sigma_w^2}} = \frac{2y_k}{\sigma_w^2} \] (2.38)

where \( y_k \) is the received value fed by the channel to the node, and \( \sigma_w^2 \) is the estimated noise variance. The knowledge of the noise variance is not always available at the
decoder, in which case the decoder must rely on $y_k$ as the only source of information available.

2.5 Conclusion

In this chapter, we made a short exposition of LDPC codes. Regular and irregular codes were explained and derivations of the most widely used decoding algorithms were carried out. The results in this chapter, especially the BP algorithm and its simplified version, the SP algorithm, will be employed in the next chapter.
Chapter 3

Nonuniform Sampling in Channel Estimation

Small cells mounted on high-speed trains exhibit rough and fast changes in the channel between them and the base station. In this chapter, we propose an algorithm for joint channel estimation and decoding using LDPC codes (presented in Chapter 2) in fast fading frequency-flat channels. The algorithm offers better performance in the high Doppler frequency region compared to other estimation methods. The transmission scheme is block-oriented and the estimation algorithm uses the time-varying channel parameters from previous blocks to initialize some parameters in subsequent blocks.

3.1 Pilot Symbol Assisted (PSA) Channel Estimation

Channel estimation methods can be classified into three major groups: pilot-assisted, blind and semi-blind methods. In this chapter, we deal with pilot-assisted channel estimation only. The interested reader is referred to [22, 28] and the references therein for literature on blind and semi-blind channel estimation. In pilot-assisted methods, a train of pilot symbols is inserted at regular points in the data stream. The pilot spacing is governed by the coherence time and the coherence bandwidth of the channel.
In fast fading channels, the coherence time of the channel is smaller than the symbol period. According to the sampling theory, this implies a high-frequency insertion of pilots since the maximum separation between the channel samples has to be no more than half the period of maximum variation of the channel [61, 46]. However, this limits the throughput of the information link. To avoid this limitation, it is possible to use the soft output of the decoded stream to generate extra high fidelity pilot symbols and employ a joint estimation and decoding scheme.

Pilot-assisted channel estimation techniques have been employed in many communication systems to obtain remarkable enhancements. For example, in [69] the authors use Discrete Prolate Spheroidal sequences to estimate the fading channel under the assumption that the Doppler spread is known. The estimation relies on solving a set of linear equations for each received frame. In [45], the authors use B-splines to estimate the fading channel. They employ local splines due to their computational simplicity. In [62], the channel estimation of pilot symbol assisted Turbo codes is carried out by solving the Wiener-Hopf equations. The set of filter coefficients is then used to approximate the channel response. In [39], a Minimum Mean Square Error (MMSE) Turbo equalizer is presented, and a practical method of computing the equalizer coefficients using the Fast Fourier Transform (FFT) is derived.

The use of LDPC codes in joint decoding and estimation was implemented in many different scenarios. For example, the Bit Interleaved Coded Modulation (BICM) concept was utilized in [48] and analyzed using density evolution and extrinsic information transfer (EXIT) charts. The authors in [4] presented a joint detector and decoder in which a Multiple-Input Multiple-Output (MIMO) channel detector and a finite geometry LDPC decoder iteratively exchange soft information. The use of
(non-binary) LDPC codes is considered in [65] where the proposed algorithm combines hard decision decoding using message passing with the signal detector iteratively.

All the aforementioned algorithms do not exploit the log-likelihood ratio (LLR) values of the underlying bits. These values are an indicator of reliability that can be used to extract the variations of the fading channel.

The idea of using irregular pilots and soft valued LLR positions as pseudopilots was studied in [61], where the authors proposed the use of an adaptive weights method to reconstruct the fading process and decode the packet. However, this method does not address the shortcomings of the reconstruction process, and the selection of the number of interpolation points is determined through simulations only.

### 3.2 Contributions

The system proposed in this chapter has the following attributes:

- Applies a nonuniform interpolation method for estimating the underlying time varying fading channel using the soft output of the LDPC decoder.

- Applies a fixed-point number implementation of a LLR locator algorithm that finds the maximum absolute values of the LLR stream from the LDPC decoder and determines their locations. This algorithm is suitable for implementation in real life scenarios and not limited to computer simulations.

- Proposes a regularized iterative conjugate gradient method at the core of the estimation block to mitigate the ill-conditioning of the reconstruction matrix used in the interpolation algorithm.
3.3 System Structure

Figure 3.1 depicts the system structure. It is composed of a transmitter, wireless fading channel and a receiver. The transmitter employs an LDPC encoder with periodic pilot insertion. The LDPC-coded sequence is then modulated and sent through the wireless channel. The received symbol is given by

\[ r[i] = c[i]x[i] + n[i] \quad i = 0, 1, \ldots, N - 1 \]

where \( x[i] \) is the transmitted modulated symbol, \( c[i] \) is a complex channel coefficient and \( n[i] \) is additive white Gaussian noise with variance \( \sigma_n^2 \) in both the in-phase and quadrature components.

The fading process \( c[i] \) is modeled as a zero-mean complex Gaussian sequence and has the autocorrelation function

\[ \rho_c[\tau] = E\{c[i]c^*[i + \tau]\} = \sigma_c^2 J_0(2\pi f_d T_s \tau), \]

where \( E[\cdot] \) denotes the statistical expectation, \( J_0(\cdot) \) is the zero order Bessel function of the first kind, \( f_d \) is Doppler spread, \( T_s \) is the symbol period, and \( \sigma_c^2 \) is the variance of the
fading process. The normalized Doppler spread is defined as the product $f_d T_s$ [29].

The receiver works in an iterative manner. The LDPC decoder generates a soft-output sequence that is used by the reliability locator to produce a nonuniform interpolation grid. The nonuniform interpolator uses the obtained grid for reconstructing the channel.

3.4 Estimation Using Nonuniform Samples

3.4.1 Overall Description

Figure 3.2 depicts the block diagram of the receiver. It consists of three main parts: a channel estimator (nonuniform interpolator), a soft output LDPC decoder and a LLR locator. Upon receiving the input stream, the estimator uses the pilot symbols
to give a rough estimate of the fading channel coefficients. The LDPC decoder then
uses that estimate to decode the input stream and generates a vector of LLRs. The
LLR locator uses these values to generate a vector containing the maximum absolute
LLR locations (pseudopilot locations) that are used by the nonuniform reconstruction
algorithm. This loop continues until the LDPC issues a vector with no errors or when
the maximum number of iterations is exceeded.

In this section, we first show the relationship between the LLR values and BER.
We then develop a simple method for obtaining the highest absolute LLR values.
After deriving the reconstruction algorithm, we explain the need for regularization by
exploring the ill-conditioning of the reconstruction methods already in use. Finally,
this section investigates the computational complexity of the proposed receiver.

3.4.2 Relationship between LLR and BER

To understand the motivation behind using the LLR values, and the assumption
of high fidelity of these values, we simulated the LDPC decoder under AWGN and
binary phase shift keying with different SNR values. The number of simulations was
set to 100,000. The maximum value of the LLR values was set to 2,048; on real-time
processor implementation, the maximum LLR value is governed by the number of bits
given for a signed integer. The absolute values of the LLRs were sectioned into 10 bins
with bin 0 having the lowest absolute LLR value and bin 10 having the highest value.
The BER of each bin was estimated. Figure 3.3 shows the BER for an SNR of 2 dB.
It can be clearly seen that there exists an inverse correlation between the LLR value
and the corresponding BER. The simulations were repeated for different SNR values
and the same trend was observed. Therefore, our proposition that those symbols with
high absolute LLR be regarded as pseudopilots seems natural. This can be justified by noting that the LLR value is just the log likelihood ratio of the probability of the received signal, given that the transmitted symbol is a zero, to the probability of the received signal, given the transmitted symbol is a one. The expression of the LLR was derived in Equation (2.30) and its use explained in Chapter 2. It is given by

$$\text{LLR} = \log \frac{P(r|x = 0)}{P(r|x = 1)}.$$  \hspace{1cm} (3.2)

As the credibility of the numerator (denominator) gets higher, the value of the \( \log(\cdot) \) function increases (decreases) correspondingly.

3.4.3 LLR Locator

In this section, we introduce the LLR locator shown at the receiver in Figure 3.2 and explain how it produces the vector of LLRs to be fed to the interpolator. A simple sorting algorithm gives the highest absolute LLR values in a vector denoted by \( r_\ell \).
The LLR values produced by the LDPC decoder are stored in signed integer notation. Each LLR value is stored in a column as shown in Figure 3.4. The first row holds the sign bits. The second row holds the most significant bits (MSB), while the last row holds the least significant bits (LSB). The locator algorithm starts with two lists, \(\text{list1}\) and \(\text{list2}\). \(\text{list1}\) holds the number location of each symbol and \(\text{list2}\) is an empty list to be filled with the reliable symbol locations.

The algorithm starts by disregarding the sign bits stored in the first row. Then scanning row by row, it locates the first appearance of a one in a row. If a one is found, it adds its location to \(\text{list2}\) and removes it from \(\text{list1}\). Upon finishing the scan process of the first row, the same procedure is done for the columns with indexes in \(\text{list1}\). This assures the elimination of any redundant locations. The algorithm terminates upon finding a predetermined number of LLR locations and stores them in the vector \(r_\ell\).

The highest absolute valued LLRs are in fact dispersed nonuniformly across the stream. The nonuniform interpolator uses those dispersed pilots, or “pseudopilots,” to further improve the estimates. The feedback from the estimated fading channel to the LDPC decoder further improves the performance of the estimated channel in terms of
MSE. The outer loops are defined as the iteration between the LDPC decoder and the nonuniform interpolator. The estimator internal loops are defined as the number of iterations that the nonuniform interpolator uses to reach a predefined tolerance set by the design. The performance is degraded with the effects of additive noise; however, this degradation is minimized through the regularization method (see section 3.4.4), which gives preference to solutions with smaller norms. This regularization improves the conditioning of the interpolation problem, and enables a numerical solution. The next section gives a thorough analysis of the nonuniform sampling and interpolation methods used in this chapter, and explains the need for regularization.

3.4.4 Nonuniform Interpolation

The LLR vector produced by the sequential bit locator is used to get the pseudopilots required for the reconstruction of the channel coefficients using the regularized Conjugate Gradient (CG) algorithm. To derive the regularized CG reconstruction algorithm and show why we need it, we have to show the ill-conditioning of the reconstruction matrix. The original channel fading vector is denoted by $c$. The channel is to be reconstructed using the pseudopilots provided by the ALLR locator. We denote by $f(t)$ the continuous counterpart of the discrete sampled channel vector $c$. Let $f$ be a bandlimited function sampled at a non-equispaced grid $t_j$ producing $f(t_j)$ . The function is bandlimited in the sense that $f$ belongs to the space $B_\Omega$ [53], where $B_\Omega$ is defined as

$$B_\Omega = \{ f \in L^2[\mathbb{R}] : \hat{f}(\omega) \text{ for } |\omega| > \Omega \}$$

(3.3)
3.4. ESTIMATION USING NONUNIFORM SAMPLES

and \( \hat{f} \) is the Fourier transform of \( f \) defined as

\[
\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-2\pi j \omega t} dt.
\]

(3.4)

The Shannon-Whittaker-Kolmogorov sampling theory states that the band-limited function \( f \) can be reconstructed using the formula

\[
f(t) = \sum_n f\left(\frac{n}{\Omega}\right) \text{sinc}(\Omega t - n)
\]

(3.5)

where \( \text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \). The grid \( n \) is equispaced and the maximum intersample spacing is governed by the bandwidth \( \Omega \). However, if the sampling is taken at irregular spacing, the reconstruction of the original signal will no longer follow Equation (3.5). Frame theory [53] presents the basic tool to analyze such cases. Next, we give a brief summary of the main results in frame theory needed in our analysis.

A family \( \{x_j\} \) in a Hilbert space \( H \) is said to be a frame for \( f \) if there exist constants \( 0 < A \leq B < \infty \) such that

\[
A\|f\|^2 \leq \sum_j |\langle f, x_j \rangle|^2 \leq B\|f\|^2.
\]

(3.6)

The constants \( A \) and \( B \) are called the lower and upper frame bounds, respectively. If \( A = B = \text{constant} \), the frame is called a tight frame. The analysis operator \( T \) is defined as

\[
T : f \in H \rightarrow Tf = \{\langle f, x_j \rangle\}
\]

(3.7)
and the synthesis operator $T^*$ is defined as
\begin{equation}
T^* : c \in \ell^2(\mathbb{Z}) \rightarrow T^*c = \sum_j c_j x_j.
\end{equation}

Combining Equation (3.7) and Equation (3.8) we get the frame operator $S$
\begin{equation}
S = T^*T
\end{equation}
\begin{equation}
Sf = \sum_j \langle f, x_j \rangle x_j.
\end{equation}

Given a frame $S$ and its image $S^{-1}$, we can reconstruct the function $f$
\begin{equation}
f = \sum_j \langle f, x_j \rangle y_j = \sum_j \langle f, y_j \rangle x_j
\end{equation}
where $y_j = S^{-1}x_j$ is called the dual frame. From Equation (3.10) it is clear that the moments $\langle f, x_j \rangle$ can be used to reconstruct the function $f$.

The reconstruction of a function from its sampled version at points $\{\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots\}$ is given by
\begin{equation}
f(t) = \sum_j c_j \text{sinc}(t - t_j)
\end{equation}
where $c = [\ldots, c_{-2}, c_{-1}, c_0, c_1, c_2, \ldots]$ is the solution of
\begin{equation}
Rc = b
\end{equation}
where $R$ is the Gram matrix with entries $R_{ij} = \text{sinc}(t_i - t_j)$ and $b_j = f(t_j)$.

The problems that need to be investigated are the ones related to stability and uniqueness of reconstruction. The stability issue arises when we traverse from the
infinite-dimensional case in theory to the finite-dimensional approximation used in practice. The truncation of frames and its effects on the stability of the reconstructed solution can be understood when the problem is solved in matrix form [52, 53]. Let the orthogonal projection operator $P_n$ be such that

$$P_n c = c_{<n>} = [c_{-n}, c_{-n+1}, \ldots, c_{n-1}, c_n].$$

(3.13)

This enables us to write the truncated Gram matrix $R_{<n>}$ as $R_{<n>} = P_n R P_n$ and $b_{<n>} = P_n b$:

$$R_{<n>} c(n) = b_{<n>}.$$  

(3.14)

However, if $f$ is sampled at $m$ times at the Nyquist rate with $t_j = j/m$, then $\text{sinc}(\cdot-t_j)$ is a tight frame with frame bounds $A = B = m$. The truncated function reconstruction is given by

$$f_n(t) = \sum_{j=-n}^{n} \frac{f(t_j)}{m} \text{sinc}(t-t_j).$$

(3.15)

In matrix form, the Gram matrix $R_{<n>}$ is Toeplitz with entries

$$R_{<n>}(i,j) = \frac{\sin\\((\pi/m)(i-j))\\}{(\pi/m)(i-j)}. $$

(3.16)

This matrix is also called the prolate matrix. Its behavior and properties are well studied and documented. The ill-conditioning of this matrix can be understood after plotting the eigenvalue distribution of such matrices. In what follows, we drop the subscript $<n>$ to denote finite dimensional vectors and matrices and assume all vectors and matrices are finite dimensional unless stated otherwise. Figure 3.5 shows the surface plot and the normalized eigenvalues of the prolate matrix for $m=10$ and
3.4. ESTIMATION USING NONUNIFORM SAMPLES

$m=100$. The magnitude of the eigenvalues for $m=100$ drops towards zero faster than that for $m=10$, indicating a clustering around 1 and an exponential decrease towards 0. This behavior illustrates the reason for classifying this matrix as ill-conditioned [51]. The nonuniform sampling case is no better than the uniform one. As $m$ increases, the ill-conditioning tends to aggravate the reconstruction problem due to round-off errors or additive noise effects on vector $b$. The low eigenvalues tend to amplify any noise or perturbations in the range space. This can be overcome by using the Moore-Penrose inverse of such matrices. If we denote by $R = S V U^*$ the singular value decomposition (SVD) of the prolate matrix, and by $\lambda_i$ the $i^{th}$ eigenvalue of $R$, then a modified version of the Moore-Penrose inverse of $R$ is $R^+ = U V^+ S^*$, where $U$ and $S^*$ are unitary matrices and $V^+$ is given by

$$V^+ = \text{diag} \left( \left\{ \lambda_j^+ \right\} \right), \quad \lambda_j^+ = \begin{cases} \frac{1}{\lambda_j} & \lambda_j > 0 \\ 0 & \text{otherwise} \end{cases}$$  (3.17)

It can be seen that eliminating the low values of $\lambda_j$ tends to lower the contributions of the perturbations in $b$ at the expense of solving a SVD decomposition at every iteration. There are some other solutions to this problem where regularization techniques are used. Some of these techniques are given in [3], where multiple regularization methods were studied and implemented. Tikhonov regularization was found to be more heuristic than systemic, and the conjugate gradient method was found to be the most suitable algorithm in terms of computational complexity requirements. Furthermore, the combination of Tikhonov regularization and the conjugate gradient method leads to a satisfactory solution to the severe ill-conditioning of interpolation matrices.
3.4. ESTIMATION USING NONUNIFORM SAMPLES

Figure 3.5: (a) Prolate matrix surface plot for \( m = 10 \) (b) Prolate matrix surface plot for \( m = 100 \) (c) Normalized eigenvalues for \( m = 10 \) (d) Normalized eigenvalues for \( m = 100 \)
3.5 Regularization Techniques

In this section, we will briefly discuss some of the well known regularization techniques for solving ill-conditioned inverse problems.

3.5.1 Tikhonov Regularization

In Tikhonov regularization, we consider all solutions with $\|Rc - b\|_2 \leq \delta$ for some constant $\delta$, and select the one that minimizes the norm of $c$ [3]:

$$
\min \|c\|_2 \\
\text{s.t.} \quad \|Rc - b\|_2 \leq \delta.
$$

(3.18)

An intuitive explanation for such a choice of minimization is that any nonzero feature in the regularized solution will increase the value of the norm of $c$. As $\delta$ increases, the set of feasible solution set of Equation (3.18) also increases, and the minimum value of $\|Rc - b\|_2$ increases as well. Thus, the values of $\delta$ and $\|c\|_2$ trace out a curve.

The compromise between a measure of fitness of the solution and the allowed noise level falls under the design option. The minimization can also be traced using the following optimization formulation:

$$
\min \|Rc - b\|_2 \\
\text{s.t.} \quad \|c\|_2 \leq \varepsilon
$$

(3.19)

The damped least squares is also considered as an option

$$
\min \|Rc - b\|_2^2 + \alpha^2 \|c\|_2^2
$$

(3.20)
3.5. REGULARIZATION TECHNIQUES

Figure 3.6: Different curves for Tikhonov regularization (a) A particular model norm (b) A particular misfit norm

which is a Lagrange multiplier implementation of Equation (3.18) and Equation (3.19). \(\alpha\) is called the regularization parameter. Figure 3.6 shows the plots of two curves for Tikhonov regularization with two different norms. It can be seen that the curve traced by these values takes on a characteristic L shape. This happens because \(\|b\|_2\) is a strictly decreasing function of \(\alpha\), and \(\|Rc - b\|_2\) is a strictly increasing function of \(\alpha\). The selection of a particular value of \(\alpha\) is called the L-curve criterion, in which the value of \(\alpha\) that gives the solution closest to the corner of the L-curve is selected [3].

3.5.2 Truncated Generalized Singular Value Decomposition (TGSVD)

The TGSVD is based on realizing the eigenvalue spread of the matrix \(R\). A truncation value is then set. It eliminates the eigenvalues that are very small in value and causes the solution to vary, in case there is some disturbance or noise. Let \(\tau\) denote this
3.5. REGULARIZATION TECHNIQUES

truncation level. The regularized pseudo-inverse $R^{1, \tau}$ is given by:

$$R^{1, \tau} = U \text{diag}(\{d_{k}^{\tau}\}) S^{\ast}, \quad d_{k}^{\tau} = \begin{cases} \frac{1}{\lambda_{j}} & \lambda_{j} > \tau \\
0 & \text{otherwise} \end{cases} \quad (3.21)$$

The optimum truncation level depends on the dimension of $R$ and the noise level.

3.5.3 The Conjugate Gradient Method (CGM)

The CGM is considered one of the most prominent iterative methods for solving sparse linear equations. The method starts by transforming the linear equation model into a quadratic form [32]. A quadratic form is simply a scalar quadratic function of a vector with the form

$$F(c) = \frac{1}{2} c^{T} R c - b^{T} c + \text{const} \quad (3.22)$$

where the vector $c = [c_{-n}, c_{-n+1}, \ldots, c_{n-1}, c_{n}]^{T}$. Since $R$ is symmetric Toeplitz, the minimizer of this form is given by $Rc - b$, which is the original system that we are seeking to solve. Minimizing $F$ means we are trying to find the gradient of $F$ defined as

$$\nabla F(c) = \begin{bmatrix} \partial F / \partial c_{-n} \\
\vdots \\
\partial F / \partial c_{n} \end{bmatrix}. \quad (3.23)$$

The steepest descent method uses this gradient and moves along the opposite direction in order to minimize the objective function $F$. The algorithm is summarized in these
3.5. REGULARIZATION TECHNIQUES

Figure 3.7: Graphical motion of the Steepest Descent algorithm

steps:

1. \( c_0 \): start with arbitrary \( c_0 \)
2. \( r_k = b - Rc_k \quad k = 1, 2, \ldots \)
3. \( \alpha_k = \frac{\langle r_k, r_k \rangle}{\langle r_k, Rr_k \rangle} = \frac{r_k^T r_k}{r_k^T Rr_k} \) (3.24)
4. \( c_{k+1} = c_k + \alpha_k r_k \)

where \( \langle r_k, r_k \rangle = r_k^T r_k \), and \( c_i = [c_n^{(i)}, c_{n+1}^{(i)}, \ldots, c_{n-1}^{(i)}, c_n^{(i)}] \).

The term \( r_k \) is the residual after each iteration and the term \( \alpha_k \) is the direction of steepest descent for the next iteration. With careful analysis, it was found that the paths traversed by the steepest descent method are redundant in many iterations. Figure 3.7 shows an example of a positive-definite system of equations with two
variables and the steepest descent line search for the minimum. The path can be clearly seen as redundant. One way to eliminate this redundancy is to make the residuals after each iteration orthogonal. This can be accomplished by introducing the concept of R-orthogonal vectors, or vectors orthogonal with respect to a matrix. Two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are called $R$-orthogonal or conjugate, if

$$\mathbf{x}_1^T R \mathbf{x}_2 = 0.$$  \hspace{1cm} (3.25)

This addition allows the iterative solution to arrive at the optimum point much faster than the steepest descent method (see Figure 3.8). With this addition and after some
derivations [32, 3], the conjugate gradient algorithm sums up to:

\[
\begin{align*}
\mathbf{d}(0) &= \mathbf{r}(0) = \mathbf{b} - R\mathbf{c}(0), \\
\alpha(i) &= \frac{\mathbf{r}^T(i)\mathbf{r}(i)}{\mathbf{d}^T(i)R\mathbf{d}(i)}, \\
\mathbf{c}(i+1) &= \mathbf{r}(i) + \alpha(i)\mathbf{d}(i), \\
\mathbf{r}(i+1) &= \mathbf{r}(i) - \alpha(i)R\mathbf{d}(i), \\
\beta(i+1) &= \frac{\mathbf{r}^T(i+1)\mathbf{r}(i+1)}{\mathbf{r}^T(i)\mathbf{r}(i)}, \\
\mathbf{d}(i+1) &= \mathbf{r}(i+1) + \beta(i+1)\mathbf{d}(i).
\end{align*}
\]

(3.26)

The analysis of the CGM and comparison with the steepest descent is discussed in [32] along with complexity and convergence issues.

### 3.5.4 Maximum Entropy Regularization

In the maximum entropy method, we use the form \(\sum_{i=1}^{n} c_i \ln(w_i c_i)\) to perform the regularization, where \(w_i\) is a positive weighting factor that favours a particular type of solution, and \(\{c_i\}_{i=1}^{n} \geq 0\). The term maximum entropy comes from a Bayesian approach in selecting a prior probability distribution, although the solutions in our case are not probabilistic. We maximize the entropy of \(\mathbf{c}\) subject to a constraint on the size of misfit measure \(\|R\mathbf{c} - \mathbf{b}\|_2\):

\[
\begin{align*}
\max \quad & -\sum_{i=1}^{n} c_i \ln(w_i c_i) \\
\text{s.t.} \quad & \|R\mathbf{c} - \mathbf{b}\|_2 \leq \delta \\
& \mathbf{c} \geq 0.
\end{align*}
\]

(3.27)
3.6 Reconstruction Algorithm

There exist many reconstruction algorithms for nonuniform sampling in the literature. Most of the methods are iterative and each has its advantages and disadvantages in terms of complexity, memory usage, speed, and stability. Some of the methods can be found in [23, 2, 7, 1, 11]. In this section we will focus on deriving the necessary formulas for use by the nonuniform interpolator. The reconstruction algorithm using Fourier methods will also be derived.

If the bandwidth of the original signal can be estimated (i.e., the Doppler spread estimated and fed to the demodulator to allow for a correct detection), then the use of this “prior” information can be incorporated into the reconstruction algorithm [52].

Let the sequence \( \{f(n)\}_{n=0}^{N-1} \) consisting of \( N \) samples with \( 2M + 1 \) nonzero Fourier coefficients be given by:

\[
f(n) = \frac{1}{N} \sum_{k=-M}^{M} \hat{f}(k) e^{i2\pi kn/N}
\]

which is the formula for the IDFT where \( \{\hat{f}(k)\}_{k=-M}^{M} \) are the Fourier coefficients. The bandlimited discrete signal can therefore be interpreted as a trigonometric polynomial of degree \( M \) and length \( N \) on the unit interval. If we introduce the function

\[
c(\ell) = \frac{1}{N} \sum_{k=-M}^{M} \hat{f}(k) e^{i2\pi k\ell/N}
\]

then \( f(n) = c\left(\frac{n}{N}\right) \). The problem at hand now is that of reconstructing \( c(t) \) from its nonuniform samples. Since the polynomial \( f \) is bandlimited to \( M \), then it can be

In the next section we will derive the reconstruction algorithm used by the nonuniform interpolator.
reconstructed using $2M + 1$ samples according to the fundamental theory of algebra. Let

$$D_M(t) = \sum_{k=-M}^{M} e^{i2\pi kt}$$

be the Dirichlet Kernel. According to the sampling theorem, the function $c(t)$ can be rewritten using its sampled points $\{c(0), \ldots, c(N-1)\}$ as

$$c(t) = \sum_{n=0}^{N-1} c(n) D_M(t - n).$$

Traversing from a uniformly dispersed set of sampled points $\{n\}_{n=0}^{N-1}$ to a nonuniform set of points $\{t_j\}_{j=0}^{r}$, such that the sequence $0 \leq t_1 \leq t_2 \leq \ldots \leq t_r < N$ is a subsequence of $\{0, 1, \ldots, N - 1\}$, and using adaptive weights to accommodate for the difference in sampling points separation, we would like to employ a frame operator on $c(t)$ in Equation (3.31) using the definition in Equation (3.7). The frame operator $S$ employed on $c(t)$ gives \[26\]

$$Sc(t) = \sum_{j=1}^{r} c(t_j) w_j D_M(t - t_j)$$

where the adaptive weights are defined as

$$w_j = \frac{f(t_j + 1) - f(t_j)}{2}$$

which gives higher weights to erratic or rough adjacent points and less weight to smooth or less erratic neighboring points. If $c(t)$ is given by

$$c(t) = \sum_{k=-M}^{M} y_k e^{i2\pi kt}$$

42
then

\[ S c(t) = \sum_{j=1}^{r} \left[ \left( \sum_{k=-M}^{M} y_k e^{i 2\pi k t_j} \right) w_j \left( \sum_{l=-M}^{M} e^{i 2\pi l (t-t_j)} \right) \right] \]

\[ = \sum_{l=-M}^{M} \left[ \sum_{k=-M}^{M} \left( \sum_{j=1}^{r} w_j e^{-i 2\pi j (l-k)} \right) y_k \right] e^{i 2\pi l t} . \]  

(3.35)

If we denote by \( A_w \) the \((2M + 1) \times (2M + 1)\) matrix with entries

\[ a_{kl} = a_{k-l} = \sum_{j=1}^{r} w_j e^{-i 2\pi j (k-l)}, \quad |k|, |l| \leq M \]  

(3.36)

we get

\[
A_w = \begin{bmatrix}
a_0 & a_1 & a_2 & \cdots & a_{2M} \\
a_1 & a_0 & a_1 & \cdots & a_{2M-1} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{2M} & a_{2M-1} & a_{2M-2} & \cdots & a_0
\end{bmatrix}
\]

and

\[ y = A_w^{-1} \hat{b} \]  

(3.37)

where the \( k^{th} \) element of \( \hat{b} \) is given by

\[ \hat{b}_k = \sum_{j=1}^{r} c(t_j) w_j e^{-i 2\pi k t_j}. \]  

(3.38)

The original signal can then be recovered using an \( N \)-point inverse discrete Fourier transform

\[ c(t) = \sum_{k=-M}^{M} y_k e^{i 2\pi k t}. \]  

(3.39)

The aforementioned algorithm is called the "Adaptive weight Conjugate gradient Toeplitz method" or ACT [52, 53] (the regularized conjugate gradient algorithm will be explained further in section 3.6.1). Figure 3.9 depicts the ACT reconstruction
Figure 3.9: Nonuniform reconstruction of a bandlimited signal of 512 samples using 40 samples

algorithm of a bandlimited signal of 512 samples using 40 samples. Fast implementation of this algorithm involves the use of Fourier methods in solving the Toeplitz system by augmenting the vector of unknowns by zeros and reflecting it at the origin. This allows for a great reduction in complexity. As noted before, if the sequence $0 \leq t_1 \leq t_2 \leq \cdots \leq t_r \leq N$ is a subsequence of $\{0, 1, \ldots, N-1\}$, and we define the sequences [52]

$$s_w(n) = \begin{cases} 
\frac{w_j}{N}, & n = t_j \\
0, & \text{otherwise}
\end{cases}$$

$$s_c(n) = \begin{cases} 
C(n)w_j, & n = t_j \\
0, & \text{otherwise}
\end{cases}$$

(3.40)
then it can be seen that

\[ \hat{b}_k = \hat{s}_c(k) = DFT\{s_c(k)\}, |k| \leq M \]  
(3.41)

\[ \hat{a}_k = \hat{s}_w(k) = DFT\{s_w(k)\}, k = 0, 1, 2, \ldots, 2M. \]  
(3.42)

The other “accelerating” method that we need to discuss is the exploitation of the special structure of Toeplitz matrices. Circulant matrices have the special property that multiplication of a vector by them is equivalent to passing the vector through a finite-impulse-response filter (FIR). The other property is that the eigenvectors are the columns of a unitary discrete Fourier transform matrix of the same size [24, 8]. If A is a Circulant matrix then

\[ A = \frac{1}{N} F_N \text{diag}(\hat{a}) F_N^{-1}. \]  
(3.43)

where \( \hat{a} \) is the discrete Fourier transform of the first row of the matrix A and

\[ F_N = \left\{ e^{-i2\pi k n \frac{N}{N}} \right\}_{k,n=0}^{N-1}. \]  
(3.44)

The solution of the system of equations \( Ay = \hat{b} \) is given by

\[ y = F_N \text{diag}(1/\hat{a}) F_N^{-1} \hat{b} \]  
(3.45)

where \((1/\hat{a})\) is the element-wise inverse of the vector \( \hat{a} \). \( A_w \) has a Toeplitz structure: to make it Circulant, the matrix is augmented by zeros to produce a Toeplitz Hermitian matrix \( A'_w \). The addition of extra zeros between the reflected vectors is to insure that
3.6. RECONSTRUCTION ALGORITHM

the length of the new vector is a power of two. This allows for the use of the FFT with optimum speed. The same augmentation and flipping is done for vector $\hat{b}$, the new vector $b = [\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_{N-1}, \hat{b}_N, \hat{b}_{N+1}, \ldots, \hat{b}_{2N-1}]^T$ is then incorporated in the CG method and the first $N$ samples are the ones used in the algorithm

$$A_w' = \begin{bmatrix} a'_0 & a'_1 & \ldots & a'_{N-2} & a'_{N-1} & \bar{a}'_{N-1} & \ldots & \bar{a}'_2 & \bar{a}'_1 \\ \bar{a}'_1 & a'_0 & a'_1 & \ldots & a'_{N-2} & a'_{N-1} & \bar{a}'_{N-1} & \ldots & \bar{a}'_2 \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{a}'_{N-2} & \bar{a}'_{N-1} & a'_N \end{bmatrix}$$

where

$$a'_i = \begin{cases} \hat{a}_i & \text{if } 0 \leq i \leq 2M \\ 0 & \text{if } 2M + 1 \leq i \leq N - 1 \end{cases} \quad (3.46)$$

and $\bar{a}$ is the conjugate of $a$.

3.6.1 Iterative Regularized Conjugate Gradient Method

The inverse problem in Equation (3.37) can be solved using the CG method. The ill-conditioning of matrix $A_w$ poses a problem as the number of outer iterations increases. The regularized CG method tackles Equation (3.37) by solving the Quadratic
Programming unconstrained cost function

\[
\min_y C(y, \alpha) = \|A_w y - b\|^2 + \alpha \|Hy\|^2 
\]  
(3.47)

where \(H\) is a highpass operator. Minimizing \(\|Hy\|^2\) means that we are seeking a smoother transition between samples. The iterative solution for this is given by

\[
(A_w + \alpha I)y_{k+1} = \alpha y_k + b . 
\]  
(3.48)

The operator \(H\) can be taken as a simple difference operator giving the cost function

\[
C(y, \alpha) = \|A_w y - b\|^2 + \alpha \|y - y_k\|^2 . 
\]  
(3.49)

The choice of the parameter \(\alpha\) can affect the solution of the minimization problem. A lower value emphasizes the original CG method while a higher value makes the algorithm work as a smoother of the solution. It puts higher penalty on solutions with higher norms. The tricky part involves the choice of a proper \(\alpha\) where it should be proportional to \(\|A_w y - b\|^2\), which means that \(\alpha\) is a function of \(y\)

\[
\alpha(y) \propto \|A_w y - b\|^2 . 
\]  
(3.50)

As the channel conditions worsen, the right hand side of Equation (3.50) increases. Therefore, the value of \(\alpha\) should be inversely proportional to the estimated SNR. Here we choose it as

\[
\alpha = \frac{1}{(E_{SNR} + \epsilon)} 
\]  
(3.51)
3.6. RECONSTRUCTION ALGORITHM

where $E_{SNR}$ is the estimated SNR and $\epsilon$ is a small positive number. The rationale behind this choice is that the better the channel condition, the less is our need for regularization, since the limitation on the output norm is no longer needed in the high SNR region. On the other hand, as the value of SNR decreases, the need for regularization manifests itself. The value of $\alpha$ should represent a solution between the two extremes: the noiseless case represents the generalized inverse as in Equation (3.37), and the other represents the smoothest attainable solution when the noise power increases to infinity. Other possible choices for $\alpha$ include some functionals of $C(y, \alpha(y))$ given in [70, 33]:

- $\alpha(y) = \frac{[A_wy-b]_+^2}{\gamma_1 - |Hy|^2}$
- $\alpha(y) = \ln(\gamma_2 M + 1)$

where $\gamma_1$ and $\gamma_2$ are constants, but these choices do not incorporate the values of the detected symbols in determining $\alpha$.

To measure the SNR, we propose using a projection matrix that spans the subspace of the signal modulated by the channel. The projection matrix $P_i$ is given by

$$P_i = \tilde{X}_i c^{(i)} \left( c^{(i)H} c^{(i)} \right)^{-1} c^{(i)H} \tilde{X}_i^H$$

(3.52)

where $\tilde{X}_i = diag\{\hat{x}^{(i)}[n]\}, n = \{0, 1, \ldots, N-1\}$. $\{\hat{x}^{(i)}[n]\}_{n=0}^{N-1}$ are the estimated values of the transmitted sequence $x[n]$ at iteration $i$, and $c^{(i)}$ is the vector of channel coefficients $c$ at iteration $i$. The SNR estimator is given by

$$E_{SNR} = \frac{r^H P_i r}{r^H P_i^H r} = \frac{r^H P_i r}{\|r\|^2 - r^H P_i r}$$

(3.53)
where $P_i^j = I - P_i$ and $r$ is the received sequence.

At each iteration, the channel $c$ is estimated at the uniform points grid $\{c_n : n = 0, 1, \ldots, N - 1\}$ using Equation (3.39). Assume the transmitted sequence $x$ is normalized to unity power. Then at each iteration $i$, the LDPC decoder generates the LLR values for the sequence $x$ denoted by $r_l^{(i)}$ given the estimated channel coefficients $c_n^{(i-1)}$. $c_n^{(0)}$ are the channel coefficients estimated using the pilot symbols. As the LDPC-estimator iterations increase, the condition-number of matrix $A_w$ deteriorates. This mandates a change of the value of $\alpha$ to incorporate the number of LDPC-estimator iterations as a factor.

To link the LLR values with the SNR estimator, a shaping filter $F$ is needed. The filter will act in 2 steps, shaping and then normalizing such that the output is in the interval $[-1, 1]$. Knowing that the LLR vector $r_l$ has a range of $[-\text{lim}, \text{lim}]$, we suggest the use of one of the following filters:

1. Step function: A hard decision decoder where the values are set to zero if they fall in the region $[-\xi, \xi]$ and hardly decoded otherwise

$$F_{\text{Step}}(r_l) = \begin{cases} 
\text{sign}(r_l[n]) & |r_l[n]| \geq \xi \\
0 & \text{otherwise}
\end{cases}$$

2. Piecewise linear step: The dead zone in the step function is relaxed to a linear function

$$F_{\text{LinStep}}(r_l) = \begin{cases} 
\text{sign}(r_l[n]) & |r_l[n]| \geq \xi \\
\frac{r_l[n]}{\xi} & \text{otherwise}
\end{cases}$$
3.6. RECONSTRUCTION ALGORITHM

Figure 3.10: MSE of SNR estimation error averaged over 1000 blocks for the proposed methods with
(a): $f_dT_S = 0.01$ and (b): $f_dT_S = 0.03$

3. Smooth: The transition is a smooth function

$$F_{Smooth}(r_t) = \tanh\left(\frac{r_t[n]}{\gamma \times lim}\right), \quad \gamma \in (0, 1]$$

where $\xi$ is a cutoff factor and $\gamma$ is a confidence factor.

Figures 3.10 (a) and (b) have been obtained for $f_dT_S = 0.01$ and 0.03, respectively. It can be seen that $F_{smooth}$ outperforms the other filters in terms of MSE. In particular, for the chosen $F_{smooth}$ filter, the estimator is robust especially at low SNR values, where we notice a 0.5dB gain over the other methods. The output of the smoothing filter will be used to calculate the estimated SNR according to Equation (3.52) and Equation (3.53).
3.6. RECONSTRUCTION ALGORITHM

After having determined the parameters needed for the regularization method, we now present the CG algorithm that will be used to find the solution of Equation (3.49). The CG algorithm begins by setting the maximum number of outer iterations $K_{out}$ and the stopping tolerance $\varsigma_{out}$, and the maximum number of inner iterations $K_{in}$ and the stopping tolerance $\varsigma_{in}$, as shown in Algorithm 1.

**Algorithm 1 Nonuniform Conjugate Gradient Reconstruction Algorithm**

1: Obtain the received signal $y$, the channel estimate $c$, and the LLR vector $r_{\ell}$
2: $k_o \leftarrow 0$
3: Calculate $E_{SNR} = \frac{r^H P r}{r^H P r_1}$, $\alpha(y)$
4: repeat
5: $res = b - A_w y$, $v = A^H w r$, $\psi^0 = \|v\|^2$
6: set the inner iteration counter $k_i \leftarrow 0$
7: while $k_i \leq K_{out}$ and $\sqrt{\psi^{k_i-1}} > \varsigma_{in}\sqrt{\psi^0}$ do
8: if $k_i = 0$ then
9: set $\beta \leftarrow 0$ and $p \leftarrow v$
10: else $\{k_i \neq 0\}$
11: $\beta \leftarrow \psi^{k_i-1}/\psi^{k_i-2}$ and $p \leftarrow v + \beta p$
12: end if
13: $q \leftarrow A_w p$
14: $w \leftarrow A^H q + \alpha p$
15: $\alpha \leftarrow \psi^{k_i-1}/p^H w$
16: $y \leftarrow y + \alpha p$
17: $v \leftarrow v + \alpha w$
18: $\psi^{k_i} \leftarrow \|res\|^2$
19: $k_i \leftarrow k_i + 1$
20: end while
21: Calculate $E_{SNR}$ and update $\alpha$
22: $k_o \leftarrow k_o + 1$
23: until ($k_o > K_{out}$)

In the next section we compare the complexity of the proposed algorithm with several other algorithms in the literature.
3.6. RECONSTRUCTION ALGORITHM

3.6.2 Computational Complexity

The computational complexity of the proposed algorithm is directly proportional to the SNR and the maximum Doppler spread $f_d$. As the SNR increases, the CG algorithm converges faster and the number of inner iterations decreases. The three main blocks of the receiver (as shown in Figure 3.2) are

- The channel estimator
- LDPC decoder
- The LLR locator

The channel estimator is based on the regularized CG algorithm described in section 3.6.1. The complexity of a matrix-vector multiplication is $O(m)$ where $m$ is the number of nonzero elements in matrix $A_w$. For the $E_{SNR}$ estimation, Equation (3.53) shows that one iteration requires $N(N+1)$ complex number multiplications and $N^2 - 1$ complex number additions. For the other part of the algorithm each iteration requires:

- One matrix vector multiplication
- Two dot products
- A number of scalar multiplications and vector additions

Matrix $A_w$ has nonzero elements in the order of the limit set by $O((f_d T S N)^2)$, where $N$ is the frame length. For an incremental number of pseudopilots, that is a fraction of $r_\ell$ given by $\ell_{frac} r_\ell$, the complexity per iteration per symbol is given by $O(\frac{K_{out} \ell_{frac} r_\ell (f_d T S N)^2}{N})$. Note that if matrix $A_w$ has condition number $\kappa$, then the time
3.7. CONVERGENCE ANALYSIS

The complexity of the CG algorithm is upper bound by \( \frac{1}{2} \kappa \ln(\frac{1}{\varsigma_{out}}) \), where \( \varsigma_{out} \) is the stopping tolerance of the CG algorithm.

The LDPC decoder complexity depends on the algorithm used for decoding. There exist many decoding algorithms and variants with tradeoffs between complexity and performance like Belief Propagation (BP), Sum-Product (SP), Lookup tables, and decoding with different scheduling schemes. For all the simulations, we employed the BP algorithm. For a rate \( \frac{1}{2} \) \((N, J, 2J)\) LDPC code, the total complexity associated with one iteration of BP consists of \(11NJ - 9N\) real multiplications, \(N(J + 1)\) real divisions, and \(N(3J + 1)\) real additions [19].

For the LLR locator, we need to estimate the number of comparisons needed to find the first \([2f_dT_s + 1]\) ones in the LLR bit matrix shown in Figure 3.4. If the probability distribution of bits in the input stream is uniform (assuming proper source coding is applied to the information stream), then the average number of ones in the first row is \(\frac{N}{2}\), and in row \(i\) is \(\frac{N}{2^i}\). We need to find the number of rows \(n_r\) such that:

\[
[2f_dT_s + 1] = N \sum_{i=1}^{n_r} \frac{1}{2^i} \tag{3.54}
\]

then

\[
n_r \propto \log_2 \left( 1 + \frac{2[2f_dT_s + 1]}{N} \right) \tag{3.55}
\]

which gives an average number of comparisons of \(n_rN\).

3.7 Convergence Analysis

Following the convergence performance analysis for the CG algorithm done in [34, p. 22], we analyze the convergence of the iterative regularized CG algorithm proposed
3.7. CONVERGENCE ANALYSIS

in section 3.6.1. Denote by \( y^* \) the optimal solution of Equation (3.48). If the CG method starts from a point \( y_0 \in \mathbb{R}^{2N} \), then applying the CG algorithm to solve the linear equation \( Ay = b \), where \( A \in \mathbb{R}^{2N \times 2N} \) is a symmetric positive definite matrix, would give an approximation \( y_k \) that satisfies \[ \|y_k - y^*\|_A \leq \min_{p_k \in P_k} \max_{z \in \Lambda(A)} \left| p_k(z) \right| \|y_0 - y^*\|_A \leq 2 \left( \frac{\sqrt{\lambda_{\text{max}}(A)} - \sqrt{\lambda_{\text{min}}(A)}}{\sqrt{\lambda_{\text{max}}(A)} + \sqrt{\lambda_{\text{min}}(A)}} \right)^k \|y_0 - y^*\|_A \] \quad (3.56)

where \( \|y\|_A \) is the \( A \)-norm defined as \( \|y\|_A = y^H Ay \). \( \Lambda(A) \) represents the spectrum set of \( A \), and \( \lambda_{\text{min}}(A) \) and \( \lambda_{\text{max}}(A) \) represent the smallest and largest eigenvalues respectively, while \( P_k = \{ p \mid p \) is a polynomial of degree \( k \) and \( p(0) = 1 \} \) is the set of \( k \)th degree residual polynomials. As a consequence of the inequality in Equation (3.56), the equivalent for the proposed iterative regularized CG method becomes

\[ \|y_k - y^*\|_{(A_w + \alpha I)} \leq 2 \left( \frac{\sqrt{\lambda_{\text{max}}^2(A_w) + \alpha} - \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}}{\sqrt{\lambda_{\text{max}}^2(A_w) + \alpha} + \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}} \right)^k \|y_0 - y^*\|_{(A_w + \alpha I)} \] \quad (3.57)

Let

\[ r_k = b - A_w y_k \]
\[ G(\alpha) = A_w + \alpha I \]
\[ D_k(\alpha) = \alpha y_k + b \]

then we get

\[
\begin{align*}
G(\alpha)y_{k+1} - D_k(\alpha) &= (A_w + \alpha I) y_{k+1} - (\alpha y_k + b) \\
&= \alpha (y_{k+1} + y_k) + (A_w y_{k+1} - b) \\
&= \alpha A_w^{-1} (r_k - r_{k+1}) - r_{k+1} \\
&= \alpha A_w^{-1} r_k - (\alpha A_w^{-1} + I) r_{k+1}
\end{align*}
\] \quad (3.58)
3.7. CONVERGENCE ANALYSIS

so that

\[
(\alpha A_w^{-1} + I) r_{k+1} = \alpha A_w^{-1} r_k - (D_k(\alpha) - G(\alpha)y_k) \\
\]

(3.59)

\[
r_{k+1} = (\alpha A_w^{-1} + I)^{-1} \left[ \alpha A_w^{-1} r_k - (D_k(\alpha) - G(\alpha)y_k) \right].
\]

But we know that

\[
\left\| \frac{1}{2} A_w^2 y \right\|_2 = \| y \|_{A_w}
\]

and

\[
\sqrt{\lambda_{\min}(A_w)} \| y \|_{A_w} \leq \| A_w y \|_2 \leq \sqrt{\lambda_{\max}(A_w)} \| y \|_{A_w}
\]

then

\[
\frac{\| D_k(\alpha) - G(\alpha)y_k \|_2}{2}
\]

\[
= \| D_k(\alpha) - (A_w + \alpha I) y_k \|_2
\]

\[
= \| (A_w + \alpha I) \left[ (A_w + \alpha I)^{-1} D_k(\alpha) - y_{k+1} \right] \|_2
\]

\[
= \| (A_w + \alpha I) \left[ y^* - y_{k+1} \right] \|_2
\]

\[
\leq \sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \| y^* - y_{k+1} \|_{(A_w+\alpha I)}
\]

\[
\leq 2\sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \left( \sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \right) \| y^* - y_k \|_{(A_w+\alpha I)}
\]

(3.60)

\[
\leq 2\sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \left( \sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \right) \| (A_w + \alpha I) \left[ y^* - y_k \right] \|_2
\]

\[
\leq 2\sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \left( \sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \right) \| D_k(\alpha) - (A_w + \alpha I) y_k \|_2
\]

\[
\leq 2\sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \left( \sqrt{\frac{\lambda_{\max}(A_w)}{\lambda_{\min}(A_w)}} + \alpha \right) \| r_k \|_2.
\]
Then the relation between the residuals in consecutive iterations is given by

\[ \|r_{k+1}\|_2 \leq \alpha \|(A_w + \alpha I)^{-1}\|_2 \|r_k\|_2 + \|(A_w + \alpha I)^{-1}\|_2 \|D_k(\alpha) - G(\alpha)y_k\|_2 \]

\[ \leq \frac{\alpha}{\lambda_{\text{max}}(A_w) + \alpha} \|r_k\|_2 + 2\frac{\lambda_{\text{max}}^2(A_w) + \alpha}{\lambda_{\text{min}}^2(A_w) + \alpha} \left( \frac{\lambda_{\text{max}}^2(A_w) + \alpha - \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}}{\lambda_{\text{max}}^2(A_w) + \alpha + \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}} \right)^k \|r_k\|_2 \]  

(3.61)

Note that the first term in Equation (3.61)

\[ \frac{\alpha}{\lambda_{\text{max}}^2(A_w) + \alpha} < 1 \]

and in the second term

\[ \left( \frac{\sqrt{\lambda_{\text{max}}^2(A_w) + \alpha} - \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}}{\sqrt{\lambda_{\text{max}}^2(A_w) + \alpha} + \sqrt{\lambda_{\text{min}}^2(A_w) + \alpha}} \right)^k < 1 \]

which means that for increasing iteration number \( k \) the algorithm converges.

Figure 3.11 shows the convergence of performance of the proposed regularized CG algorithm with high condition number averaged over 1000 simulations.

3.8 Numerical Simulations

In this section, the performance of the proposed regularized iterative nonuniform estimator is compared to several other iterative channel estimators. In the simulations, the LDPC encoder matrix used a rate \( R_c = 1/2 \) with length \( N = 816 \). The internal LDPC decoder number of iterations is set at 20 per loop for all estimators. The channel is generated by using the algorithm described in [67], in which the fast
3.8. NUMERICAL SIMULATIONS

frequency-flat, correlated fading channel is generated using the Discrete Fourier transform (DFT) with a specified length and normalized Doppler spread. This model does not account for the periodicity of the resulting channel, so, further investigation into different models should be pursued. Nevertheless, because of the fast implementation of this model, it will be used throughout the simulations. In all the simulations, the generated channel was setup such that the DFT of the channel spans a large number of coherence times.

3.8.1 Correlated Rayleigh Simulator

There are many potential candidates for the simulation of correlated Rayleigh fading channels in the literature. Figure 3.12 shows the simulator proposed in [68] where
Figure 3.12: Correlated Rayleigh random variate generation
Figure 3.13: Received correlated Rayleigh fading channel with AWGN

the authors used the inverse Fourier methods to generate the random-variate, which allowed for fast implementation of such a simulator.

Figure 3.13 shows a typical simulation of a received correlated Rayleigh fading channel plus additive noise at 10 dB; the parameters under control are the time span or sampling instances and the normalized Doppler shift. The interested reader is referred to [67] and references therein for the derivation of the generation process in Figure 3.12.

3.8.2 Pilot Spacing

Here, we want to evaluate the effects of pilot spacing over the estimation MSE. The influence of pilot spacing, denoted by $M_s$, is simulated for $f_dT_S = [0.005, 0.01,$
0.03, 0.05] for 1000 packets each. The SNR is estimated using the $F_{smooth}$ filter presented in Section 3.6.1. The number of outer iterations between the decoder and estimator is fixed at 5. It can be seen in Figure 3.14 that, for high $f_d T_S$, the curves are bowl-shaped. However, the effects of pilot spacing on MSE are less severe for low values of $f_d T_S$. For low values of $M_s$ the estimator incurs some degradation in MSE performance. The reason for this loss is that the energy is split between the pilots and the data symbols. The value of $E_b/N_0$ is given by $(\frac{R_s}{\sigma_n^2})(\frac{M_s-1}{M_s})$. Therefore, for low values of $M_s$, the loss in $E_b/N_0$ is high. On the other hand, for high values of $M_s$, the estimator loses track of the channel, and the number of pilots are not enough to make the estimator converge.
### 3.8. NUMERICAL SIMULATIONS

#### 3.8.3 MSE Performance

In this section, the MSE performance of the proposed algorithm is compared to the Wiener filter, Spheroidal basis expansion, B-spline interpolation, and the Turbo equalization.

Here, we briefly describe these techniques. For the Wiener filter approach, the coefficients \( w_{est} \) are found by solving the Wiener-Hopf equations:

\[
\sum_{i=-[K/2]}^{[K/2]} w_{est,i}\rho_c[m-i] + 2\sigma^2 w_{est,m} = \rho_c[m], \quad -[K/2] \leq m \leq [K/2]
\]  

(3.62)

where \( K \) is the order of the filter assumed to be odd [62]. The first estimate of the channel is given by approximating the channel coefficients at any instant to the nearest

---

**Figure 3.15:** Estimator MSE vs. \( f_dT_s \) for various estimators with SNR=5dB
pilot symbol. The Discrete Prolate Spheroidal Sequences method (DPSS) relies on the Slepian basis expansion functions \( \psi_d = [\psi_d[1], \psi_d[2], \ldots, \psi_d[N]], \ d = 1, \ldots, 2f_dT_SN+1 \), where \( N \) is the frame length. The Slepian sequences are characterized by being bandlimited to the frequency range \([\pm f_dT_S, f_dT_S]\) and simultaneously most concentrated in a certain time interval. These sequences \( \psi_d[\cdot] \) are designed specifically for each normalized Doppler frequency. Since the basis functions are orthonormal, the expansion coefficients are given by [69]:

\[
d_{est} = G^{-1} \sum_{m \in P} r[m]p^*[m]U^*[m]
\]  

(3.63)

where

\[
G = \sum_{\ell \in P} U[\ell]U^H[\ell]
\]

and

\[
U[\ell] = \begin{bmatrix}
\psi_0[\ell] \\
\vdots \\
\psi_{D-1}[\ell]
\end{bmatrix}
\]

where \( P \) is the pilot index set, \( p[i] \) is the \( i^{th} \) pilot symbol. The modification of this method to be iterative is done by incorporating the \( r_\ell \) vector and feeding it back to the DPSS estimator. Note that, each time the pseudopilot set increases, the number of the Slepian sequences used increases, and the corresponding MSE decreases.

The B-spline method [45] tries to capture the variations of the channel by using B-spline interpolation. The coefficients are derived to minimize the MSE using the
autocorrelation of the channel as input. The B-spline used is of order $\eta$ and

$$\hat{c}[m] = \sum_i h_i b_\eta[m - iT_{av}]$$  \hspace{1cm} (3.64)$$

where

$$h_i = \sum_{m=-L_1}^{L_2} a_m s_i - m$$

$$\sum_{k=-L_1}^{L_2} a_m A_{kl} = \gamma_\eta(l), l = -L_1, \ldots, L_2$$

where $\gamma_\eta(k)$ and $A_{kl}$ are given by

$$\gamma_\eta(k) = \frac{1}{T_{av}} \sum_{m=-(\eta+2)T_{av}/2}^{(\eta+2)T_{av}/2} b_{\eta+1}[m] \hat{\rho}_c[m + kT_{av}]$$

$$A_{kl} = \sum_{m=-\eta}^{\eta} b_{2\eta+1}[mT_{av}] G^\rho[lT_{av} - kT_{av} + mT_{av}]$$

and $G^\rho[kT_{av}]$ is given by

$$G^\rho[kT_{av}] = \frac{1}{T_{av}} \sum_{m=-T_{av}}^{T_{av}} b_{\eta+1}[m] \hat{\rho}_c[m - kT_{av}]$$

where $T_{av}$ is the averaging period and $\hat{\rho}_c$ is the estimated autocorrelation function from the estimated channel coefficients [45]. The method is also modified to incorporate the $r_\ell$ vector to use the pseudopilots in the interpolation process.

The employed Turbo equalizer [39] uses a recursive convolutional code with generator polynomial $[1, \frac{1+D+D^2+D^3}{1-D-D^3}]$ and a number of iterations equal to 5.
Figure 3.15 shows the MSE performance comparison of the Wiener filter, cubic B-spline interpolator, Spheroidal basis expansion, Turbo equalizer and our proposed nonuniform estimator for \( f_d T_S = [0.005, 0.01, 0.03, 0.05] \) at SNR level =5dB. It can be seen that the DPSS and B-spline methods are close in performance measures. The Wiener filter, with a filter order of 11, performs rather poorly. It will also be shown in the next section that our proposed method outperforms all other methods in the high Doppler spread region, noting that as the value of the normalized Doppler frequency increases to about 0.05, the MSE value begins to increase, but the proposed nonuniform algorithm maintains a gain of 0.9 dB over the closest competitor even at high fading rates.

### 3.8.4 BER Performance

The BER performance of the different techniques is explored for \( f_d T_S = [0.005, 0.01, 0.05, 0.1] \) in the \( E_b/N_0 \) range of \([0, 5]\) dB for BPSK modulation. The number of iterations for the LDPC min-sum algorithm are set at 20 for all of the estimators. The pilot separation \( M_s \) is decreased from the upper limit dictated by the sampling theory. It is set at \( M_s = 28, 22, 8, 4 \) for the indicated values of \( f_d T_S \), respectively. The value of \( E_b/N_0 \) is given by \( (\frac{R}{\sigma_n^2}) \cdot (\frac{M_s^{-1}}{M_s}) \), corresponding to a penalty of 0.1579, 0.202, 0.5799 and 1.2494 dB for the corresponding values of \( M_s \).

Figures 3.16, 3.17, 3.18, and 3.19 show the BER performance for the different techniques. The figures indicate that our proposed nonuniform estimator outperforms its competitors, especially at high SNR. This can be justified by the increase of the number of pseudopilots provided by the LLR locator for the interpolator, and by the observation that the other methods require a higher number of pilots per packet for
3.8. NUMERICAL SIMULATIONS

![Graph showing BER vs. E_b/N_o for f_dT_s=0.005 and pilot separation 28]

The initial estimation of the channel.

It should be noted that the performance of the proposed nonuniform estimator improves as the Doppler spread increases. This is due to the fact that, for high Doppler spread values, the spatial dispersion of the pseudopilots decreases. This is caused by the fast variations of the channel within a given packet. Note that the trend in BER improvement with an increasing Doppler spread was also observed in other works [63, 62].

3.8.5 Complexity

Figure 3.20 shows the computational complexity for the five estimators. It can be seen that the nonuniform estimator’s complexity falls between the DPSS and the B-spline estimators. The complexity grows with increasing f_dT_s as pointed out in section 3.6.2. However, one advantage of the proposed nonuniform method is that the reconstruction
3.8. NUMERICAL SIMULATIONS

Figure 3.17: BER vs. $E_b/N_o$ for $f_dT_s=0.01$ and pilot separation 22

Figure 3.18: BER vs. $E_b/N_o$ for $f_dT_s=0.05$ and pilot separation 8
3.8. NUMERICAL SIMULATIONS

Figure 3.19: BER vs. $E_b/N_o$ for $f_dT_S=0.1$ and pilot separation 4

Figure 3.20: Number of multiplications per symbol per iteration vs. $f_dT_S$
algorithm (excluding the inverse Fourier transform) itself depends on the number of pseudopilots generated and does not perform point by point interpolation.

3.9 Conclusion

In this chapter we presented an algorithm for joint channel estimation and decoding using nonuniform interpolation. We showed that the reconstruction matrix using nonuniform interpolation is ill-conditioned. A regularized Conjugate Gradient algorithm was developed to tackle this problem. The level of regularization was linked to the estimated SNR.

We studied the effect of pilot spacing on the estimator’s performance. Simulation results showed greater dependence of the reconstruction MSE on the pilot spacing for higher normalized Doppler spread.

The proposed scheme was compared to other state-of-the-art estimation algorithms including the Discrete Prolate Spheroidal Basis method, the B-Spline method, Wiener filtering, and Turbo equalization in terms of MSE, BER and complexity. The proposed nonuniform estimator showed better performance in terms of MSE and BER for different values of normalized Doppler spread. The complexity of the proposed scheme was derived analytically and the number of computations was shown to depend on the normalized Doppler spread. Extensive computer simulations and comparisons with other methods showed the competitiveness of the proposed scheme.
Chapter 4

Signal Shaping Using LDPC Codes

With small cells mounted on fast travelling vehicles, it is very important to conserve battery power as much as possible. In this chapter, we present an algorithm for the reduction of the average transmitted power using signal shaping. The algorithm solution relies on a LP formulation. The sparsity of the LDPC code offers a simple way to find a suboptimal but fast solution to the problem.

4.1 Introduction

The transmitted signal in a communication link should be tailored as much as possible to suit the requirements stipulated by the communication channel. Low transmit power is desirable in mobile communications due to lack of resources at the user terminal. The battery life is to be prolonged as much as possible without any downgrades to the quality of service. Moreover, because the bandwidth is shared by multiple users, the transmit power of one user translates directly into unwanted interference or noise to other users. Signal shaping is one of the methods used for this purpose. The most popular means of signal shaping is to transmit signals with the lowest-average power without any sacrifice in system performance.
4.1. INTRODUCTION

With the use of packetized digital communications comes the need for algorithms that can produce such minimization over blocks of information predetermined in length. The use of linear block codes in such situations is desirable. The algorithm must be able to search over all of the permitted sequences to choose the sequence that results in the lowest-average power.

The original idea of sign-bit shaping that was used by Forney [17] relies on two input streams. In [17], the input stream was divided into two streams, the first stream constituted the sign shaper and the second stream constituted the data bits. To avoid information loss due to shaping, Forney used syndrome vectors from coding theory at the transmitter. In his method, the shaping algorithm chooses the vector from the null space of the parity check matrix that results in the lowest average energy.

This chapter presents a novel algorithm for signal shaping over linear block codes using Linear Programming (LP) methods. The majority of the proposed algorithms for achieving shaping gain are mostly involved with convolutional codes. The rest involve using nonuniform spacing to achieve a Gaussian probability distribution on the transmitted ensemble [38]. The proposed algorithm relies on Forney’s trellis shaping methods [17]. In his paper, Forney proposed the use of 2’s complement representation of the constituent shaping bits to allow for the flexibility in choosing the sign bits, to change the power of the transmitted symbols. In our treatment of Linear Block codes, we extend the ideas of LP used in coding theory [54, 55, 56]. In the next section we will derive the equations needed to cast the signal shaping problem in the LP domain. A simplification of the problem in terms of $\ell_1$ minimization is also proposed. Numerical simulations show that the probability distribution of the transmitted signal approaches the Gaussian distribution after a small number of frames, where the
shaping gain is measured accordingly. The constellation expansion ratio is shown to
decrease as the number of points decrease, with no tangible increase in computational
complexity, except in calculating the energy vector needed for the minimization prob-
lem. The problem itself is not affected by the constellation size chosen. This makes
the algorithm flexible for adaptive coding and modulation methods.

4.2 Notion of Shaping

Signal shaping is principally modifications applied to a signal to alter some properties
of the transmitted waveform. The most popular application is the reduction of the
average transmitted power without any sacrifice in performance. If we conceptualize
a pair of i.i.d input streams (1D-PAM) to be transmitted, then the signal points will
be equiprobable given that the proper source coding is applied. The 2D histogram
of the constellation will take the shape of a square. Given that the number of con-
stellation points is a power of 2, Figure 4.1(a) shows the constellation mapping of
equiprobable two 16-ary streams and the associated joint probability density func-
tion (PDF)in Figure 4.1(c). By jointly mapping the two streams, the average power
can be lowered [13]. The points with high energy are reflected (mapped) to the lower
energy points. The highest energy points in the square 256-QAM constellation are the
corner points. Thus, eliminating these points results in the circular mapping shown
in Figure 4.1(b). As can be seen from the joint PDF, the high energy endpoints
have less probability. Considering N-dimensional PDF, the signal points should be
enclosed in an N-dimensional sphere in place of an equiprobable N-dimensional cube
[13].
4.3 Measures of Performance

There are many issues and effects to study when considering signal shaping (a detailed discussion of the subject can be found in [6, 16, 18, 13]):

I Shaping Gain: Defined as the ratio of average signal energy in equiprobable signalling to the average signal energy of the shaped signal.

II Constellation Expansion Ratio (CER): The ratio of the number of signal points in the shaped signal constellation to the number of signal points in the original equiprobable constellation. CER is always greater than unity.

III Peak-to-Average Energy Ratio (PAR): Is the ratio of the peak energy of the low-dimensional projected constellation to its average energy, in dB. It is always
4.3.1 Ultimate Shaping Gain

If the baseline system uses a uniform distribution and the shaped signal is constrained using constant variance, then the resulting ultimate PDF is a Gaussian distributed signal for the 1D projected signal. For both systems to transmit the same amount of information, their entropies must be equal. The differential entropy of the reference system transmitting symbols $x$ with average energy $E_s$ is given by

$$h(x) = \frac{1}{2} \log_2 (12E_s)$$

(4.1)

and the entropy of a Gaussian distributed system with average energy $E_o$ is

$$h(x) = \frac{1}{2} \log_2 (2\pi e E_o) .$$

(4.2)

Therefore, the ultimate shaping gain that can be sought for is

$$\frac{E_s}{E_o} = \frac{\pi e}{6} \approx 1.53 \text{dB} .$$

(4.3)

And this is achieved with a continuous Gaussian probability density function.

4.3.2 Shaping Methods

Since the idea of shaping is basically one of transforming the signal to deform the uniform PDF into a Gaussian-like shaped PDF, it can be related to source coding. Source coding tries to take a redundant stream of data and produce a non-redundant
4.3. MEASURES OF PERFORMANCE

Figure 4.2: Huffman code with 21 Codewords

equiprobable stream. So there exists a duality between source coding and shaping.

Figure 4.2 depicts a Huffman Code with 21 codewords. If the source is i.i.d., then each code word has a probability of $2^{-l_i}$, where $l_i$ is the length of the $i^{th}$ sequence. The entropy of such a codebook is 4. This means that it is equivalent to a 16-QAM equiprobable constellation.

The average energy of the (zero mean compensated) shaped constellation is $E=8.3$, giving a shaping gain of 0.77 dB. The CER is calculated as $21/16 = 1.31$. The PAR in 16-QAM equals $18/10$ or 2.55 dB, and the PAR for the shaped constellation is $25/8.38 = 4.76$ dB, an increase of 2 dB.
4.3. MEASURES OF PERFORMANCE

Figure 4.3: Constellation for the Huffman codebook. The dashed line is for the 16-QAM constellation

**Shaping on Regions**

Channel coding resorts to placing different codeword points in the hyperspace individually. Forney [14, 15] introduced the idea of coset coding. A shaping algorithm chooses the region and the channel code deals with the internal arrangement of the points, as in Ungerboeck’s multilevel coding [60]. Figure 4.4 illustrates the idea. The points with similar or approximately equal energy are grouped together in a single shell. A part of the input stream chooses the region. The rest of the stream is left for the channel encoder to place the symbol inside the designated region. Sell mapping is the extension of the idea of shaping on regions. It was used in the telephone line modem standard ITU recommendation V.34. The algorithm is discussed in detail in [10, 35, 36].

Figure 4.5 illustrates the idea, where the input stream is ordered based on total
Figure 4.4: Shaping on regions

Figure 4.5: Shell mapping illustration
4.3. MEASURES OF PERFORMANCE

Figure 4.6: Distribution of output stream for different regions
energy, and only the first K output-regions are chosen. Figure 4.6 shows a Monte Carlo simulation that illustrates the algorithm, where 100,000 runs were carried out.

**Nonuniform Spacing**

This method changes the spacing between the constellation points from uniform to nonuniform. It is sometimes called warping. It is also used in ITU V.34. The constellation is fed into a warping function that performs a nonlinear transformation of the interval $[-1, 1]$, as shown in Figure 4.7. There exist multiple warping functions that fit the criteria of maximising the noise immunity when placing the signal points. The same concept can be extended to passband transmission using complex-valued warping [13].
4.3. MEASURES OF PERFORMANCE

Figure 4.8: Congruency points for 256-QAM / two 16x16 PAM

Trellis Shaping

The method of trellis shaping was proposed by Forney [17]. Trellis shaping is basically a sign-bit shaping relying on the application of 2’s complement to the input symbols. The MSB is taken as the sign bit. Complementing that bit causes the symbol to be shifted instead of reflected. This allows the high energy symbols to be mapped to low-energy positions. The loss of information due to this mapping can be compensated by the use of ideas from coding theory. If we have a 16x16 square constellation, as shown in Figure 4.8, with a normalized square distance \( d_{\text{min}}^2 = 1 \), then we can consider each coordinate as a 16-point PAM signal belonging to \( \{\pm \frac{1}{2}, \pm \frac{3}{2}, \ldots, \pm \frac{15}{2}\} \). The 2’s complement of any point in this constellation is given by “\( zabc.1 \)”, where \( zabc \) represents any binary 4-tuple, \( z \) is the sign bit and the remaining bits \( abc \) are the least significant bits. The modification of this sign bit gives an equivalence class, where the LSBs are the same.
4.3. MEASURES OF PERFORMANCE

In order to minimize information loss due to shaping, Forney [17] suggested the use of syndromes from coding theory. The generator matrix of a channel code is the matrix $G_{K \times N}$. The respective parity check matrix is denoted $H_{(N-K)\times N}$ such that $GHT = 0_{K \times (N-K)}$. The $(N-K)$-digit syndrome formed by a word $x$ of length $N$ is

$$s = Hx.$$  
(4.4)

The matrix $H$ is called the syndrome former. Then $x$ can be formulated as

$$x = H^{-1}s.$$  
(4.5)

The particular property of interest in syndrome decoding, is that all of the members of the same coset give the same syndrome. So, if a codeword $c$ is to be transmitted then $c = G^Tr$, where $r$ is a $(K \times 1)$ vector. The syndrome formed by this codeword is $s = Hc = HG^Tr = 0_{(N-K)\times K}r = 0_{(N-K)\times 1}$. If the syndrome carries information, then the syndrome former can be used as a coset representative generator. Then it can be seen that

$$H(x \oplus c) = Hx \oplus Hc = s \oplus 0 = s.$$  
(4.6)

The choice of the best sequence $c$ to add to the sequence $x$ to yield the minimum average energy in the case of convolutional codes is achieved via a Viterbi Algorithm (VA) search on the corresponding trellis diagram. For each 2-tuple input $(x_1, x_2)$, the next path energy is determined by the state transition and the LSB’s of each corresponding symbol. The aim of the algorithm is to minimize the average energy within the memory span of the trellis diagram.
4.4 SYSTEM ARCHITECTURE

In this section, we propose a system that relies on a matrix arrangement of the input bits. The input is split into two streams. Stream 1 constitutes the sign bits and stream 2 the input to the matrix as shown in Figure 4.10. The stream is fed to the matrix row-wise and operated on by the shaping algorithm column-wise. The purpose

Figure 4.9: Sign bit shaping as proposed in [17]

Figure 4.9 illustrates an example of a basic block diagram of trellis shaping. Depending on the window size in the corresponding VA trellis diagram, a 4-state rate $\frac{1}{2}$ convolutional decoder can provide 0.97 dB of shaping gain. The price is an expansion in the constellation, as CER will be equal to 2. This can be lowered if some restrictions are made on the transitions. If the restrictions are implied on the allowed peak values, such that only even or odd parity transitions are allowed for the 2-tuple sign bits, they would result in a CER of 1.41 and approximately the same shaping gain.

4.4 System Architecture
behind this arrangement is to make the code resilient against burst errors as this is just a basic interleaver that also constitutes the signal shaper. The first column is left empty as it represents the sign bits that we need to determine to minimize the total average energy of the transmitted block code. Let the matrix be of dimension \( b \times N \). The relation between the code rate for the message bits denoted \( R_c \) and the dimensions is

\[
R_c = \frac{K_c}{(b - 1) \times N} \quad (4.7)
\]

where \( K_c \) denotes the length of the input stream. The shaping algorithm takes the first row. It determines the shaping bits of the constituting symbols. For the shaping bits, the length of the input vector is given by \( K_{sh} \). In order to use this as a syndrome vector, the parity check matrix must have a dimension of \( K_{sh} \times N \). The rate of the
corresponding code is given by $R_{sh}$, where

$$R_{sh} = 1 - \frac{K_{sh}}{N}. \quad (4.8)$$

The code rate for the system is given by

$$R = \frac{N(b - 1)R_e + N(1 - R_{sh})}{Nb}. \quad (4.9)$$

Stream 2 constitutes the raw uncoded bits with $R_e = 1$ and

$$R = 1 - \frac{R_{sh}}{b} \quad (4.10)$$

Since each symbol (constituting a single column) is represented using a 2’s complement notation, each sign bit determines the position within the congruency of the resulting point within the desired constellation. If each column constitutes a symbol with the first row representing the sign bits, then the objective of the algorithm is to achieve

$$\min \sum_i E_{S,i} \quad (4.11)$$

where $E_{S,i}$ is the symbol energy at column $i$. This is done by using the principles of coding theory. By using a rate $R_{sh}$ code for the shaping bits, we can devise an algorithm that chooses a vector from the null space of the parity check matrix such that the average energy is minimized. In the next section, we discuss such an algorithm and derive the necessary equations to perform a linear programming optimization method to reach that objective.
4.5 Shaping using Linear Programming

The main idea behind using LP methods for signal shaping is to enable the use of linear block codes. Some of the advantages of using LP methods include:

- Allowing for the convenience of using a predesignated packet size without the hassles of conventional convolutional coding.

- The optimization is done over the whole code block and does not use a moving window.

- The ease with which the number of points can be incremented without adding to the complexity of the algorithm.

Each and every column in the matrix represents a point in $\text{GF}(2^b)$ with $b$ rows in $\text{GF}(2)$. This vector is represented in 2's complement notation. If we denote by $[z_i, a_{i,1}, a_{i,2}, \ldots, a_{i,b-1}]$ the $i^{th}$ column, the $z_i$ is the sign bit. We denote by $E_0(E_1)$ the vector of energies produced by the ensemble of columns by replacing the first row given by the vector $z$ with zeros (respectively ones).

The input from the buffer register to the shaping algorithm is of length $K_{sh} = E_{sh}N$. It is denoted by the vector $\mathbf{x} \in \text{GF}(2)$. $\mathbf{x}$ can be thought of as the syndrome vector at the transmitter. Since this is an information bearing vector, we require that it satisfies the parity check matrix constraints:

$$H \otimes \mathbf{y} = \mathbf{x}$$ (4.12)

where $H_{M \times N}$ is the parity check matrix with $M = N - K$, $\mathbf{y}_{N \times 1}$ is the transmitted vector and $\otimes$ means the operations are carried over $\text{GF}(2)$. We need to choose $\mathbf{y}$ such
that the average energy of the transmitted block code is minimized. In other words we would like to

\[ \min \sum_i E_{0,i}(1 - y_i) + E_{1,i}y_i \]

\[ s.t. \ H \otimes y = x. \] (4.13)

where \( E_{0,i} \) (\( E_{1,i} \)) is the energy of the corresponding vector if \( y_i \) is equal to zero (one), respectively. The optimization problem can be further simplified to

\[ \min E_D^T y \]

\[ s.t. \ H \otimes y = x \] (4.14)

where \( E_D = E_1 - E_0 \) denotes the difference energy vector. The optimizer has to search in the null space of \( H \) to find the vector that minimizes the average energy of the block code. Since \( y \in GF(2) \), this turns out to be an integer optimization problem. This can be cast into an LP problem by writing the constraints in linear form.

There has been much research done in LP decoding of block codes in the literature. In [12, 56, 66], the devised algorithms rely on the decoding of a vector \( y \) depending on the log likelihood ratios and the underlying channel. The key point in this chapter is to modify such a decoder to accommodate for a non-zero vector \( x \) in the RHS of Equation (4.12), where all the operations are performed over GF(2).

The algorithm devised for LP decoding attempts to find the maximum likelihood codeword that satisfies Equation (4.15) below. Given a binary code \( C \in F_2^N \), a codeword \( x = [x_1, x_2, \ldots, x_N]^T \) is transmitted over a memoryless channel, \( y = [y_1, y_2, \ldots, y_N]^T \) is the received vector and the LLR vector is given by \( \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T \).
The problem is formulated as

\[
\begin{align*}
\min & \quad \lambda^T x \\
\text{s.t.} & \quad H \otimes x = 0 \\
& \quad x \in \{0, 1\}.
\end{align*}
\]

(4.15)

The constraint is split into M equations (the number of rows in \(H\))

\[
\begin{align*}
h_i \otimes x = 0.
\end{align*}
\]

(4.16)

where \(h_i\) is the \(i^{th}\) row of the matrix \(H\). The problem of casting Equation (4.16) into an LP problem was done in [12]. The extension of this paper lies in using the ideas presented previously and employing them in the shaping problem. Most importantly is when the RHS of Equation (4.16) is a “1” instead of a “0”. We will use the geometric interpretation in [12] to obtain a better understanding of the derivation when the RHS is a “1”. If row \(h_i\) has a total of \(m\) ones, then we say that it has \(m\) active constraints, denoted by the set \(N(m)\). The existence of a “1” in the RHS implies that the number of ones in the active points set must be odd.

Figure 4.11 shows the polyhedron that results from joining the eligible points on the unit cube. This polyhedron is the convex hull of the points \([BDEFG]\). The solution is contained in this polyhedron. The introduction of this relaxation makes the LP problem more tractable and easier to solve. It also introduces the undesirable fractional solutions obtained from the intersection of the resulting half-spaces [59]. To derive the needed equations for the case of a “1” in the RHS, let us take the surface \([BDE]\) in Figure 4.11. The normal to that surface is the vector pointing to the even
Figure 4.11: Polyhedron of a 3 active set point with a “1” in the RHS vertex $A$. The equation that defines this half space is

$$2 \mathbf{n}^T (\mathbf{x}_1 - \mathbf{x}_2) \leq 0 \quad (4.17)$$

where $\mathbf{n}$ is the normal to surface, the factor 2 is just for numerical convenience and $\mathbf{x}_1, \mathbf{x}_2$ are two points on the surface. The normal extends from the center of the cube with coordinates $[\frac{1}{2}, \frac{1}{2}, \cdots, \frac{1}{2}]$ to an even weight vertex $\mathbf{n} = V_e - \frac{1}{2} \mathbf{1}$, where $\mathbf{1}$ is the all ones vector. It can be seen that $2 \mathbf{n}^T$ has the following form

$$2 \mathbf{n}^T = [+1, +1, \cdots, +1, -1, -1, \cdots, -1]. \quad (4.18)$$

The positive ones in $2 \mathbf{n}^T$ are denoted by the set $V \in \mathcal{N}(m)$, and the negative ones are denoted by the set $\mathcal{N}(m) \setminus V$. If we choose $\mathbf{x}_2$ as a vertex, then the number of 1’s in $\mathbf{x}_2$ is $|V| \pm 1$ because moving from an odd vertex to an even vertex in one move takes
or adds a one. Then it can be seen that

\[ 2n^T x_2 = |V| - 1 \] (4.19)

and

\[ \sum_{n \in V} x_n - \sum_{n \in N \setminus V} x_n \leq |V| - 1 \]

\[ \forall V \in N(m), |V| \text{ even} \] (4.20)

\[ 0 \leq x_n \leq 1. \]

This completes what is needed for the setup of the shaping algorithm. Since the vector \( x \) in Equation (4.14) is an information source that has a random number of 1’s and 0’s, it was essential to derive the necessary equations to solve the LP for the case of a 1 in the RHS. As for the original case of a 0 in the RHS, that case has been treated [12, 56, 59], as mentioned before and will not be discussed further.

For the LP problem cast in Equation (4.14), a simple modification results in some numerical stability for the optimizer. For each equation, the restriction of even-odd parity should be separated. This results in two different interpretations of the solution space. For the even parity solution case, let \( C_m \) denote the set of all the solution points of check node \( m \). Then the solution space can be defined as \( \text{conv}(C_m) \), the convex hull of \( C_m \). The intersection of all the spaces generates the solution space for the whole problem.

\[ P = \text{conv}(C_A) \cap \text{conv}(C_B) \cap \text{conv}(C_C) \cdots. \] (4.21)

The optimum vector \( y \) satisfying the objective problem only in Equation (4.14) might not belong to \( P \). We already have the vector \( E_D \) that produces it. Let us denote this vector by \( y_{E_D} \). We can visualize a minimum \( \ell_1 \) norm vector stretching from \( y_{E_D} \).
to the nearest point belonging to \( P \). The vector \( y_{E_D} \) is obtained by substituting a “1” in each position that \( E_D \) is less than zero, and a “0” in each position that \( E_D \) is greater than or equal to zero. Then we get

\[
\begin{align*}
y &= y_{opt} + y_{E_D} \\
H \otimes y &= x \\
H \otimes (y_{opt} + y_{E_D}) &= x \\
H \otimes y_{opt} &= x \oplus (H \otimes y_{E_D})
\end{align*}
\]

The equivalent optimization problem transforms to

\[
\min \; \|y\|_{\ell_1} \quad \text{s.t.} \quad H \otimes y = x \oplus (H \otimes y_{E_D})
\]

The two methods are equivalent, but the numerical stability of the \( \ell_1 \) minimization is higher.

### 4.6 Adaptive Linear Programming

In the original formulation of LP decoding proposed by Feldman et al. [12], the number of constraints in the LP problem is linear in the block length, but exponential in the maximum check node degree. There are \( 2^{d_c-1} \) constraints for each check node of degree \( d_c \). The computational complexity of the LP formulation is prohibitively high, since the total number of constraints is exponential in terms of the maximum check node degree, denoted by \( d_c^{\text{max}} \).

Taghavi et al. [54] suggest an adaptive algorithm based on the observation that if one of the constraints of a check node in the LP matrix is violated, then all the other
4.6. ADAPTIVE LINEAR PROGRAMMING

constraints related to that node are satisfied with strict inequality.

Proof: [54].

Given a constraint of the form

\[ h_i x \leq b_i \]  (4.24)

and a vector \( x_0 \in \mathbb{R}^N \), then

\[ h_i x_0 = b_i \]  (4.25)

is called an active constraint at \( x_0 \). A violated constraint or a cut is given by

\[ h_i x_0 > b_i \]  (4.26)

Given a check node with neighborhood \( \mathcal{N} \subset \{0, 1, \ldots, N\} \) and two subsets \( V_1 \subset \mathcal{N} \) and \( V_2 \subset \mathcal{N} \) that introduce a cut at \( x \) (both \( V_1 \) and \( V_2 \) are assumed to be of odd sizes \( |V_1| \) and \( |V_2| \), respectively), we partition \( \mathcal{N} \) into four disjoint subsets:

\[ S = V_1 \cap V_2, \]
\[ \overline{V}_1 = V_1 \setminus V_2, \]
\[ \overline{V}_2 = V_2 \setminus V_1, \]
\[ \overline{\mathcal{N}} = \mathcal{N} \setminus (V_2 \cup V_1). \]

Equation (4.20) can be rewritten as

\[ \sum_{i \in S} x_i + \sum_{i \in \overline{V}_1} x_i - \sum_{i \in \overline{V}_2} x_i - \sum_{i \in \overline{\mathcal{N}}} x_i > |S| + |\overline{V}_1| - 1 \]  (4.27)
4.6. ADAPTIVE LINEAR PROGRAMMING

and

\[ \sum_{i \in S} x_i + \sum_{i \in V_2} x_i - \sum_{i \in V_1} x_i - \sum_{i \in N} x_i > |S| + |V_2| - 1. \]  (4.28)

Adding Equation (4.27) and Equation (4.28) and multiplying by \( \frac{1}{2} \) we get

\[ \sum_{i \in S} x_i - \sum_{i \in N} x_i > |S| + \frac{|V_1| + |V_2|}{2} - 1. \]  (4.29)

This simplifies to

\[ |S| + \frac{|V_1| + |V_2|}{2} - 1 < |S| \]  (4.30)

and

\[ |V_1| + |V_2| < 2. \]  (4.31)

On the other hand we have \(|V_1|\) and \(|V_2|\), both positive and odd numbers given by

\[ |V_1| = |S| + |V_1| \]  (4.32)

\[ |V_2| = |S| + |V_2| \]  (4.33)

which gives

\[ |V_1| - |V_2| = |V_1| - |V_2|. \]  (4.34)

\(|V_1| - |V_2|\) is an even number. Therefore, \(|V_1| + |V_2|\) is an even number. This means that Equation (4.31) holds iff \(|V_1| + |V_2| = 0\). Hence, \(|V_1| = |V_2| = 0\) and \(V_1\) and \(V_2\) are identical. ■

The adaptive linear programming (ALP) method [54] uses this property to cut consumption time and computational complexity of the original LP formulation. The

91
method is initialized with a box constraint to insure the solution belongs to \([0, 1]\):

\[
\begin{align*}
0 \leq x_i & \quad \text{if } E_{D,i} \geq 0 \\
x_i \leq 1 & \quad \text{if } E_{D,i} < 0
\end{align*}
\]

(4.35)

where \(E_{D,i}\) is the \(i^{th}\) element of the vector \(E_D\).

Using the box constraints and a basic decoding decision based on the sign of the vector \(E_D\), the algorithm outputs an uncoded bit-wise hard decision based on the received vector.

**Algorithm 2 Adaptive LP Decoding [54]**

1: Initialize the LP problem with the box constraints in (4.35)
2: \(k \leftarrow 0\)
3: Find the solution \(x^0\) to the initial LP problem by bit-wise hard decoding
4: repeat
5: \(k \leftarrow k + 1\)
6: Find the set \(S^k\) of all parity inequalities and box constraints that are violated at \(x^{k-1}\)
7: if \(|S^k| > 0\) then
8: Add the constraints in \(S^k\) to the LP problem and solve it to obtain \(x^k\)
9: end if
10: until \(|S^k| = 0\)

Using the observation that a violated constraint at a check node mandates that all the other check nodes are satisfied with strict inequality, the objective of the search algorithm is to find the violated parity inequalities can be performed without the need to examine all the \(M2^{d_{max}}\) parity inequalities generated by the original LP formulation. This leads to the modified adaptive LP (MALP) algorithm. The MALP algorithm has been extended using various simplifications [71]. The cut-search algorithm (CSA) introduced in [71] combines two different approaches for solving the
4.6. ADAPTIVE LINEAR PROGRAMMING

Algorithm 3 MALP Algorithm [55]

1: Initialize the LP problem with the box constraints in (4.35)
2: \( k \leftarrow 0 \)
3: Find the solution \( x^0 \) to the initial LP problem by bit-wise hard decoding
4: repeat
5: \( k \leftarrow k + 1; \ flag \leftarrow 0 \)
6: for \( j = 1 \) to \( m \) do
7: if there is no active parity inequality from check node \( j \) in the problem \textbf{then}
8: \( \textbf{if} \ \text{check node } j \text{ introduces a parity inequality that is violated at } x^{k-1} \textbf{then} \)
9: \( \text{Remove the parity inequalities of check node } j \text{ (if any) from the current} \)
10: \( \text{problem} \)
11: \( flag \leftarrow 1 \)
12: end if
13: end if
14: end for
15: \( \textbf{if} \ flag = 1 \textbf{ then} \)
16: \( \text{Solve the LP problem to obtain } \mathbf{x}^k \)
17: \( \textbf{end if} \)
18: until \( flag = 0 \)
19: return \( x^k \) as the solution to the LP decoding

LP problem. The first approach is the MALP algorithm. The second one, which we
state here without proof, is given in theorem 1 [71]: given a nonintegral vector \( \mathbf{x} \) and
a parity check node \( j \), let \( S = \{ i \in \mathcal{N}(j) | 0 < x_i < 1 \} \) be the set of nonintegral neighbors
of node \( j \) in the Tanner graph, and let \( T = \{ i \in \mathcal{N}(j) | x_i > \frac{1}{2} \} \). A necessary condition
for parity check \( j \) to induce a cut at \( \mathbf{x} \) is

\[
\sum_{i \in T} (1 - x_i) + \sum_{i \in \mathcal{N}(j) \setminus T} x_i < 1. \tag{4.36}
\]

This is equivalent to

\[
\sum_{i \in S} \left( \frac{1}{2} - x_i \right) > \frac{1}{2} |S| - 1. \tag{4.37}
\]
4.7. NUMERICAL SIMULATIONS

Algorithm 4 Cut-Search Algorithm [71]

1: Input parity check node \( j \) and vector \( \mathbf{x} \)
2: \( V \leftarrow T = \{ i \in \mathcal{N}(j) | x_i > \frac{1}{2} \} \)
3: \( S \leftarrow \{ i \in \mathcal{N}(j) | 0 < x_i < 1 \} \)
4: if \( |V| \) is even then
5: \( \text{if } S \neq \emptyset \text{ then} \)
6: \( \{ i^* \} \leftarrow \arg \min_{i \in S} \left( \frac{1}{2} - x_i \right) \)
7: else
8: \( \{ i^* \} \leftarrow \text{arbitrary } i \in \mathcal{N}(j) \)
9: end if
10: \( \text{if } i^* \in V \text{ then} \)
11: \( V \leftarrow V \setminus \{ i^* \} \)
12: else
13: \( V \leftarrow V \cup \{ i^* \} \)
14: end if
15: end if
16: if \( \sum_{i \in T} (1 - x_i) + \sum_{i \in \mathcal{N}(j) \setminus T} x_i < 1 \) then
17: Violated parity inequality found at node \( j \)
18: else
19: No violated parity inequality on parity-check node \( j \)
20: \( V \leftarrow \emptyset \)
21: end if
22: return \( V \)

4.7 Numerical Simulations

For the conventional \( M \times M \) 2D-QAM constellation with \( d_{\text{min}}^2 = 1 \) with data rate \( R \), average energy \( S \), and constellation size \( C_{2D} \) are given by

\[
R = \log_2 M^2 \text{ (bits per 2 dimensions)}
\]
\[
S = (M^2 - 1)/6 = (2^R - 1)/6
\]
\[
|C_{2D}| = M^2 = 2^R
\]
The shaping gain \( \gamma_s \) and the constellation expansion ratio are approximated as

\[
\gamma_s \approx \frac{2^R}{6S} \\
CER = |c_{2D}|/2^R.
\]

(4.38)

In order to use the LP shaping method in an efficient way, a low density parity check matrix is used for the sign bit shaping algorithm. The code used for the sign bits is the (96, 48) LDPC code. Stream 1 of length 96 * 5 = 480 uncoded bits is shifted serially row by row starting by row number 2 down to row number 6. Each row contains 96 bits. The rate used for the shaping algorithm is \( 1/2 \). The system supports a data rate of \( R = 5.5 \) bits per one dimension. The 2D constellation size is \( |C_{2D}| = 256 \), so the CER = 1.4142. This was accomplished without even using any restrictions on the high-energy corners of the constellation.
The shaping gain was measured experimentally and was found to be 127.17 for a rate 5.5 code per one-dimension. The baseline signal energy for the same bit rate is 170.63. The shaping gain was found to be 1.2665 dB.

Remarkably, this shaping gain obtained with the LP method is achieved without the use of the VA algorithm and large window sizes that consume a lot of time. Figure 4.13 shows the performance Forney’s sign-bit shaping. The shaping gain over a conventional 4-state sign-bit shaping scheme is shown for different decoding delays. Also shown is the shaping gain of the proposed algorithm for an equivalent amount of delay.

To measure the performance of the shaping algorithm in terms of error rate over a communication channel, we plot the symbol error rate versus the SNR for different QAM constellations using the LP shaping algorithm. The channel is an AWGN.
4.8. CONCLUSION

Figure 4.14: Symbol error rate versus SNR over AWGN channel with QAM constellation of 32, 64, 128 and 256 points, respectively.

channel. The QAM constellation size $\epsilon [32, 64, 128, 256]$.

4.8 Conclusion

A method for signal shaping on linear block codes using sign-bit shaping with Linear Programming was found to be superior to the existing conventional methods. The method was found to statistically approach the Gaussian PDF with few transmitted packets. Suggested future work includes the investigation of the impact of such shaping methods on Gaussian belief propagation networks used in modeling fading channels.
Chapter 5

PAPR Reduction Via Linear Programming

For real time implementation of communication systems, the designer must be aware of the limitations that cause degradations to the performance of the system. These limitations can render the system difficult to implement or even unusable. For the case of OFDM modulation used in today’s transmission systems, the challenges are not that many. One of these points is the problem of peak-to-average-power ratio (PAPR) that limits the use of linear amplifiers in the final design. In this chapter, we propose a method to minimize PAPR using linear programming methods.

5.1 Introduction

Orthogonal frequency division multiplexing (OFDM) is a very attractive scheme in wireless communications. Its ability to cope with frequency selective fading channels due to multipath scattering of the electromagnetic wave without the use of complex equalization methods renders it suitable for many applications. The OFDM output symbol rate is much lower than the input symbol rate because of its ability to split high-rate data streams into a number of lower rate streams that are transmitted
simultaneously over a number of subcarriers. This makes the insertion of guard intervals between OFDM symbols affordable thus eliminating most or all of the impeding inter-symbol interference (ISI).

OFDM has found many applications in both cable and wireless transmissions, ranging from the WLAN IEEE 802.11a, g, n and HIPERLAN/2 to terrestrial digital and mobile TV systems such as DVB-T and DVB-H respectively.

The OFDM transmission scheme suffers many disadvantages. The key challenges are in the time selectivity of the channel, high peak-to-average power ratio (PAPR) and phase-noise in oscillators.

High PAPR produced by modulated subcarriers combining constructively to form peaks can deteriorate the performance of OFDM if the signal passes through any nonlinearity. In order to minimize power consumption, the circuitry of RF transmitters usually employ non-linear power amplifiers (PA). Any nonlinear distortion of the transmitted OFDM signal can cause in-band and out-of-band interference. Out-of-band spurs or frequency leakage causes interference with adjacent users, while in-band leakage causes inter-carrier interference (ICI).

An apparent solution to this problem is to back-off the operating point of the non-linear PA. This, unfortunately, results in a significant power efficiency penalty. So there is a compelling need to reduce the PAPR while operating the PA in the nonlinear region.

Several methods of OFDM PAPR reduction exist in the literature, such as clipping [64, 25], Hadamard Transform [49], Partial Transmit Sequence (PTS) [30], Selective Mapping (SLM) [27, 58], and exponential companding [31], among many others. These schemes can mainly be categorized into signal distortion techniques, such as
clipping, and signal scrambling techniques such as PTS.

In this chapter, we propose a method to reduce the PAPR in real baseband OFDM signals by finding a suitable phase distortion vector that scrambles the subcarrier tones. The problem is formulated in terms of the subcarrier tones. Approximations are carried out to linearize constraints making the problem amenable to LP methods. As the proposed method is a LP optimization method, then it is considered a convex optimization method for which various solvers exist to find the solution. This makes the method attractive for implementation on real-time processors. Simulations are carried out to validate the performance of the proposed method when compared to other methods.
5.2. OFDM SYSTEM MODEL AND PAPR REDUCTION METHODS

5.2 OFDM System Model and PAPR Reduction Methods

Figure 5.1 shows a simple description of an OFDM transmission system. An OFDM symbol \( \hat{x}[n] \), \( 0 \leq n \leq N \), consists of \( N \) baseband data \([x_0, x_1, \ldots, x_{N-1}]\) carried on \( N \) subcarriers chosen to be orthogonal with constant frequency spacing \( \Delta f = \frac{1}{N} \). The OFDM symbol is given by

\[
\hat{x}[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{j \frac{2\pi kn}{N}}, \quad 0 \leq n < N \tag{5.1}
\]

Equation (5.1) can be equivalently represented in matrix form as \( \hat{x} = \frac{1}{\sqrt{N}} W x \), where

\[
W = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & e^{j2\pi/N} & \cdots & e^{j2\pi(N-1)/N} \\
\vdots & \vdots & \ddots & \vdots \\
1 & e^{j2\pi(N-1)/N} & \cdots & e^{j2\pi(N-1)(N-1)/N}
\end{bmatrix} \tag{5.2}
\]

There exists many different definitions for PAPR. The conventional definition of PAPR of an OFDM symbol in the time domain is given by

\[
\text{PAPR}(\hat{x}[n]) = \max_{0 \leq n < N} \left( \frac{\left|\hat{x}[n]\right|^2}{E(\left|\hat{x}[n]\right|^2)} \right) \tag{5.3}
\]

where \( E(\cdot) \) denotes the expectation operation.
5.3 PAPR Reduction Using Linear Programming

5.3.1 Partial Transmit Sequences (PTS)

The PTS algorithm is a technique for improving the statistics of a multicarrier signal. The basic idea of the PTS algorithm is to divide the original OFDM sequence into several sub-sequences, each sub-sequence is multiplied by different weights until an optimum value is chosen.

In PTS, the input data stream is divided into several disjoint subblocks \( \{x^{(m)}\}_{m=0,1,\ldots,M-1} \). The original vector is given by

\[
x = \begin{bmatrix} x^{(0)}_{L \times 1} & x^{(1)}_{L \times 1} & \cdots & x^{(M-1)}_{L \times 1} \end{bmatrix}^T
\]

[47], where \( L = \frac{N}{M} \). The phase sequence is denoted by \( \zeta = [\zeta^{(0)}, \zeta^{(1)}, \ldots, \zeta^{(M-1)}] \). Each phase subsequence \( \zeta^{(m)} \) has \( L \) elements

\[
\zeta^{(m)} = [\zeta_0^{(m)}, \zeta_1^{(m)}, \ldots, \zeta_{L-1}^{(m)}]
\] (5.4)

The two main methods in PTS are the adjacent partition (AP-PTS) and the interleaved partition (IP-PTS). In AP-PTS, the phase sequence is different with each block but does not change within the block, i.e.

\[
\zeta_i^{(m)} = \zeta_j^{(m)}, \quad 0 \leq i, j \leq L - 1.
\] (5.5)

For IP-PTS, the phase sequence is interleaved within separate blocks to produce the same sequence across blocks

\[
\zeta_i^{(m)} = \zeta_i^{(k)}, \quad 0 \leq m, k \leq M - 1.
\] (5.6)

Figure 5.2 shows the block diagram of a generic PTS PAPR reduction system.
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

5.3.2 Real Baseband OFDM PAPR reduction via LP methods

An OFDM signal can be mathematically given by

\[
\hat{x}[n] = \sum_{i=0}^{N-1} a_i \cos(2\pi f_i n) + \sum_{i=0}^{N-1} b_i \sin(2\pi f_i n), \quad 0 \leq n < N_T
\]  

(5.7)

where \( N \) denotes the number of subcarriers, \( f_i \) denotes the \( i^{th} \) subcarrier frequency, and \( N_T \) is the number of sampling points where \( N_T \geq 2N \). The number of sampling points is greater than \( 2N \) to insure a better minimization of the PAPR for the real signal after it passes through the digital to analog converter. In the most general case, we would like to perturbate this signal by shifting the In-phase (I) and Quadrature (Q) components by the same phase angle such that the resulting PAPR is minimized. Note that with this phase shift, the subcarriers remain orthogonal. The I and Q components remain orthogonal as well, as the phase shift is applied to both components as given

![Figure 5.2: PTS block diagram](image-url)
by

\[ \hat{x}[n] = \sum_{i=0}^{N-1} a_i \cos(2\pi f_i n + \theta_i) + \sum_{i=0}^{N-1} b_i \sin(2\pi f_i n + \theta_i). \] (5.8)

Equation (5.8) can be reformulated in terms of \( \cos\theta \) and \( \sin\theta \) as

\[
\hat{x}[n] = \sum_{i=0}^{N-1} a_i \cos(2\pi f_i n + \theta_i) + \sum_{i=0}^{N-1} b_i \sin(2\pi f_i n + \theta_i) \\
= \sum_{i=0}^{N-1} a_i \left[ \cos(2\pi f_i n) \cos(\theta_i) - \sin(2\pi f_i n) \sin(\theta_i) \right] \\
+ \sum_{i=0}^{N-1} b_i \left[ \cos(2\pi f_i n) \sin(\theta_i) + \sin(2\pi f_i n) \cos(\theta_i) \right] \\
= \sum_{i=0}^{N-1} \left[ a_i \cos(2\pi f_i n) + b_i \sin(2\pi f_i n) \right] \cos(\theta_i) \\
+ \sum_{i=0}^{N-1} \left[ b_i \cos(2\pi f_i n) - a_i \sin(2\pi f_i n) \right] \sin(\theta_i). \] (5.9)

In matrix form Equation (5.9) can be rewritten as

\[
\hat{x} = [\mathbf{CA} + \mathbf{SB}] \mathbf{cos(\theta)} + [\mathbf{CB} - \mathbf{SA}] \mathbf{sin(\theta)} \] (5.10)

\[
\hat{x} = \Psi \mathbf{cos(\theta)} + \Gamma \mathbf{sin(\theta)}
\]

where

\[
\mathbf{C}_{N_T \times N} = \\
\begin{bmatrix}
\cos(2\pi f_0) & \cos(2\pi f_1) & \cdots & \cos(2\pi f_{N-1}) \\
\cos(2\pi f_0) & \cos(2\pi f_1) & \cdots & \cos(2\pi f_{N-1}) \\
\vdots & \vdots & \ddots & \vdots \\
\cos(2\pi f_0(N_T-1)) & \cos(2\pi f_1(N_T-1)) & \cdots & \cos(2\pi f_{N-1}(N_T-1)) \\
\mathbf{S}_{N_T \times N} = \\
\begin{bmatrix}
\sin(2\pi f_0) & \sin(2\pi f_1) & \cdots & \sin(2\pi f_{N-1}) \\
\sin(2\pi f_0) & \sin(2\pi f_1) & \cdots & \sin(2\pi f_{N-1}) \\
\vdots & \vdots & \ddots & \vdots \\
\sin(2\pi f_0(N_T-1)) & \sin(2\pi f_1(N_T-1)) & \cdots & \sin(2\pi f_{N-1}(N_T-1)) \\
\mathbf{\theta} = \left[ \theta_0, \theta_1, \ldots, \theta_{N-1} \right]^T \\
A = \text{diag}(\left[ a_0, a_1, \ldots, a_{N-1} \right]^T) \\
B = \text{diag}(\left[ b_0, b_1, \ldots, b_{N-1} \right]^T) \\
\Psi = \mathbf{CA} + \mathbf{SB}, \quad \Gamma = \mathbf{CB} - \mathbf{SA}.
\]
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

The objective is to minimize the PAPR of $\hat{x}$. But Equation (5.10) is nonlinear in $\theta$.

To this end we propose to linearize the problem by making the approximation

\[
\cos(\theta_i) = \frac{2}{\pi} \left( \frac{\pi}{2} - \theta_i \right), \quad 0 \leq \theta_i \leq \frac{\pi}{2} \\
\sin(\theta_i) = \frac{2}{\pi} \left( \theta_i \right), \quad 0 \leq \theta_i \leq \frac{\pi}{2}.
\]

Using the approximation in Equation (5.11) in Equation (5.10) we get

\[
\hat{x} \approx \frac{2}{\pi} \Psi \left( \frac{\pi}{2} - \theta \right) + \frac{2}{\pi} \Gamma \theta \\
= \frac{2}{\pi} \left( \Gamma - \Psi \right) \theta + \Psi 1_{N \times 1}
\]

where $1$ is the all ones vector of size $N \times 1$. 

Figure 5.3: Approximation of $\cos(\theta)$ and $\sin(\theta)$ in $\theta \in [0, \pi/2]$
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

Figure 5.4: Typical real baseband OFDM signal with no PAPR reduction

Figure 5.5: OFDM signal after Full LP
The formulation of the PAPR problem is now cast as

\[
\min \max_{0 \leq n \leq N_T-1} \left| \frac{2}{\pi} (\Gamma - \Psi) \theta + \Psi 1_{N_N^T} \right|
\]

s.t. \( 0_{N_N^T} \leq \theta \leq \frac{\pi}{2} 1_{N_N^T} \). \hspace{1cm} (5.13)

where 0 is the all zeros vector of size \( N_N \times 1 \). The linear program in Equation (5.13) can be cast as a Chebyshev approximation problem \([5]\)

\[
\min \quad t \\
\text{s.t.} \quad \frac{2}{\pi} (\Gamma - \Psi) \theta + \Psi 1_{N_N^T} \leq t 1_{N_N^T} \quad (5.14)
\]

\[
\frac{2}{\pi} (\Gamma - \Psi) \theta + \Psi 1_{N_N^T} \geq -t 1_{N_N^T} \\
0_{N_N^T} \leq \theta \leq \frac{\pi}{2} 1_{N_N^T}
\]

which can be cast as

\[
\min \quad c^T \begin{bmatrix} \theta \\ t \end{bmatrix} \\
\text{s.t.} \quad \begin{bmatrix} \frac{2}{\pi} (\Gamma - \Psi) -1_{N_N^T} \\ \frac{2}{\pi} (\Gamma - \Psi) -1_{N_N^T} \\ -I_{N_N^T} \\ I_{N_N^T} \\ 0_{N_N^T} \end{bmatrix} \begin{bmatrix} \theta \\ t \end{bmatrix} \leq \begin{bmatrix} -\Psi 1_{N_N^T} \\ -\Psi 1_{N_N^T} \\ 0_{N_N^T} \hspace{0.2cm} 0_{N_N^T} \hspace{0.2cm} \frac{\pi}{2} 1_{N_N^T} \end{bmatrix}
\]

where \( c = [0_{1 \times N_N}, 1]^T \). Figure 5.4 and Figure 5.5 show a typical OFDM signal before and after applying the full LP algorithm of Equation (5.15). To compare the gain of using an optimization without limiting the phase to be limited between \([0, \frac{\pi}{2}]\), we conducted a simulation to minimize the PAPR of Equation (5.10) using nonlinear optimization methods. Figure 5.6 shows the complementary cumulative distribution function (CCDF) of the LP program given in Equation (5.15) compared to applying the optimization to Equation (5.10) with no approximations. It can be seen that
both methods have the same performance. This is due to the observation that high peaks usually occur in a few places across the frame, so any little displacement in the subcarriers is enough to make the peak collapse and reduce the PAPR of the resulting signal.

The CCDF of PAPR for an OFDM signal $y$ is the probability that the PAPR of an OFDM signal exceeds some clipping level $PAPR_0$ (this is referred to as the symbol clip probability and determined by the specific amplifiers used in the system)

$$CCDF(PAPR(y)) = \text{Prob}(PAPR(y) > PAPR_0).$$ (5.16)
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

AP-PTS

In AP-PTS, the vector $\theta$ is partitioned according to the subblocks of the partitioned input data stream. Each subblock has the same phase rotation repeated across the subblock. The phase value may differ from one subblock to the other. Let the vector $\theta^*$ be defined as

$$
\theta^*_{N \times 1} = \left[ \theta^{*(0)T}, \theta^{*(1)T}, \ldots, \theta^{*(M-1)T} \right]
$$

(5.17)

$$
\theta^*_{m} = \theta^*_{k}, \quad 0 \leq m, k \leq L - 1.
$$

(5.17)

where $\theta^*_{(i)}$ is of size $(L \times 1)$, $L = \frac{N}{M}$. In other words

$$
\theta^*_{N \times 1} = [\theta_0, \theta_0, \ldots, \theta_1, \theta_1, \ldots, \theta_{M-1}, \ldots, \theta_{M-1}]^T
$$

(5.18)

We need to modify Equation (5.15) and add the constraints that insure Equation (5.17) is satisfied. To that end we define the matrix

$$
\mathbb{R}_{L-1 \times L} = \begin{bmatrix}
1 & -1 & 0 & \cdots & 0 \\
1 & 0 & -1 & \ddots & \\
1 & 0 & 0 & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \\
1 & 0 & 0 & \cdots & -1
\end{bmatrix}
$$

(5.19)

$$
\mathbb{E}_{M(L-1) \times N} = \begin{bmatrix}
\mathbb{R} & 0 & \cdots & 0 \\
0 & \mathbb{R} & 0 & \\
\vdots & \ddots & \ddots & \\
0 & 0 & \cdots & \mathbb{R}
\end{bmatrix}
$$
This adds the constraint \(|\Xi^{\ast} \leq \varepsilon 1_{M(L-1)\times 1}|\) to Equation (5.14), where \(\varepsilon\) is a small positive number. The AP-PTS LP program is now cast as

\[
\min c^T \begin{bmatrix} \theta^* \\ t \end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix} \frac{-2}{\pi}(\Gamma-\Psi) & -1_{N_T\times 1} \\ \frac{-2}{\pi}(\Gamma-\Psi) & -1_{N_T\times 1} \\ -I_{N\times N} & 0_{N\times 1} \\ e & 0_{M(L-1)\times 1} \\ -e & 0_{M(L-1)\times 1} \end{bmatrix} \begin{bmatrix} \theta^* \\ t \end{bmatrix} \leq \begin{bmatrix} -\Psi 1_{N\times 1} + \Psi 1_{N\times 1} \\ 0_{N\times 1} \\ 0_{N\times 1} \end{bmatrix}
\]

(5.20)

This can be further simplified by introducing the matrix

\[
\Lambda_{N\times M} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & e \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}
\]

(5.21)

This simplifies the optimization problem to

\[
\begin{bmatrix} \frac{1}{\pi^2}(\Gamma-\Psi)\Lambda & -1_{N_T\times 1} \\ \frac{-2}{\pi}(\Gamma-\Psi)\Lambda & -1_{N_T\times 1} \\ -I_{M\times M} & 0_{M\times 1} \\ +I_{M\times M} & 0_{M\times 1} \end{bmatrix} \begin{bmatrix} \theta^* \\ t \end{bmatrix} \leq \begin{bmatrix} -\Psi 1_{N\times 1} + \Psi 1_{N\times 1} \\ 0_{N\times 1} \\ 0_{N\times 1} \end{bmatrix}
\]

(5.22)

where \(\theta^*\) is a vector of size \(M \times 1\) and represents the non-redundant values of the phase vector in the AP-PTS. Figure 5.7 shows the performance of the proposed algorithm for an OFDM signal having 256 carriers for \(M = 64, 32, 16, 8, 4\) and \(2\), respectively. Figure 5.8 shows the normalized time consumed by Equation (5.22) for the given subblock lengths.
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

Figure 5.7: PAPR for different values of $M$ for AP-PTS LP

Figure 5.8: Relative time consumed by Eq. (5.22) for different $M$
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

**IP-PTS**

In IP-PTS the subblocks of phase shift vector vector are interleaved such that

\[
\theta^* = [\theta_0, \theta_1, ..., \theta_{L-1}, \theta_0, \theta_1, ..., \theta_{L-1}, ..., \theta_0, \theta_1, ......., \theta_{L-1}]^T
\]

\[
\theta^{*}(m) = \theta^{*}(k), \quad 0 \leq m, k \leq M - 1
\]  

(5.23)

The modification to Equation (5.15) is again needed to satisfy Equation (5.23)

\[
\Xi_{(M-1)\times N} = \begin{bmatrix}
I_{L\times L} - I_{L\times L} & 0 & ... & 0 \\
I_{L\times L} & 0 & ... & 0 \\
0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 \\
\end{bmatrix}
\]  

(5.24)

This also adds the constraint \(|\Xi\theta^* \leq \varepsilon 1_{(M-1)\times L}|\) to Equation (5.14). The IP-PTS LP program is now cast as

\[
\min \quad c^T \begin{bmatrix} \theta^* \\ t \end{bmatrix}
\]

s.t.

\[
\begin{bmatrix}
\frac{\pi}{2}(\Gamma - \Psi) & -1_{N\times 1} \\
\frac{\pi}{2}(\Gamma - \Psi) & -1_{N\times 1} \\
-I_{N\times N} & 0_{N\times 1} \\
-I_{N\times N} & 0_{N\times 1} \\
\Xi & 0_{(M-1)\times L} \\
\Xi & 0_{(M-1)\times L} \\
\end{bmatrix} \begin{bmatrix} \theta^* \\ t \end{bmatrix} \leq \begin{bmatrix}
-\psi 1_{N\times 1} \\
+\psi 1_{N\times 1} \\
0_{N\times 1} \\
0_{N\times 1} \\
\varepsilon 1_{(M-1)\times L} \\
\varepsilon 1_{(M-1)\times L} \\
\end{bmatrix}
\]  

(5.25)

This can be further simplified with the introduction of the following matrix

\[
\Pi_{N\times L} = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
0 & 0 & \cdots & 1 \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\end{bmatrix}
\]  

(5.26)
5.3. PAPR REDUCTION USING LINEAR PROGRAMMING

and the optimization problem becomes

\[
\begin{bmatrix}
\frac{\pi}{2} (\Gamma - \Psi) \Pi & -1_{N_T \times 1} \\
-\frac{\pi}{2} (\Gamma - \Psi) \Pi & -1_{N_T \times 1} \\
-I_{L \times L} & 0_{L \times 1} \\
+I_{L \times L} & 0_{L \times 1}
\end{bmatrix}
\begin{bmatrix}
\theta''
\end{bmatrix}
\leq
\begin{bmatrix}
-\Psi 1_{N_T \times 1} \\
-\Psi 1_{N_T \times 1} \\
0_{L \times 1} \\
\frac{\pi}{2} 1_{L \times 1}
\end{bmatrix}
\]  \hspace{1cm} (5.27)

where \(\theta''\) is a vector of size \(L \times 1\) and represents the non-redundant values of the phase vector in the IP-PTS. Figure 5.9 shows the performance of Equation (5.27) having 256 carriers for \(M = 64, 32, 16, 8, 4, \) and \(2\), which is equivalent to \(L = 4, 8, 16, 32, 64, \) and \(128\), respectively. Figure 5.10 shows the normalized time consumed by Equation (5.27) for different values of \(M\).

5.3.3 Simplified Reformulation

The information in the phase vector \(\theta\) is considered a side information that needs to be sent to the receiver in order to correctly decode the given symbol. To minimize
the overhead presented by this data, a discrete set of allowable states that the vector can have is desirable.

To restrict the values of $\theta$ to be discrete, we propose the following extension

$$\theta \in \left\{ \frac{0}{P \pi}, \frac{2}{P \pi}, \ldots, \frac{2P}{P \pi} \right\}, P \in \mathbb{Z}^+$$

(5.28)

So, the problem now is cast as an integer linear program (ILP)

$$\min \ t$$

s.t. $\frac{1}{P} (\Gamma - \Psi) \theta + \Psi 1 \leq t 1$

$$\frac{1}{P} (\Gamma - \Psi) \theta + \Psi 1 \geq -t 1$$

(5.29)

$$\theta \in \{0, 1, \ldots, P\}.$$ 

The integer $P$ determines the achievable PAPR, but it also controls the complexity of
5.4 Numerical Simulations

In this section, we compare the performance of the proposed LP PTS scheme for real-valued OFDM systems with other systems using computer simulations. The simulations are carried out for OFDM symbols with 128 carriers. The comparisons
are made between the IP-PTS, AP-PTS, SLM and normal OFDM methods. The benchmarks to be compared are the CCDF performance and relative complexity. The complexity was measured as the average number of flops for each method averaged over the total number of iterations. We compare the performance against the original SLM and PTS methods [40, 47].

Figure 5.12 shows the performance comparison when $M$ is equal to 64, while Figure 5.13 shows the performance with $M = 32$. The SLM curve was obtained by using 64 different IFFT blocks to optimize the PAPR. The signal extension method found in [40] extends the complex conjugate sequence, where zeros are appended to allow the optimization to be carried out in the real domain. The method uses the set $\pm j, \pm 1$ to choose from the different subsequences, resulting in $4^N$ different sequences.
Figure 5.12 and Figure 5.13 also show that the performance of both algorithms is similar as the number of phase shifts increase. To show the attractiveness of the suggested algorithm in real time data, we ran the algorithm with an OFDM system of block length equal to 2048. Although in real life applications the number of subcarriers is less than that, we chose to demonstrate that the algorithm performs well in the case of high number of subcarriers. The number of subsequences was set to 64 for both the ILP and AP-PTS while for the SLM the total number of multiplication sequences was set to 1024. This selection makes the SLM complexity favorable, given its predefined small set of multiplication sequences. Figure 5.14 shows that the suggested ILP algorithm has less time complexity than the AP-PTS algorithm, making it attractive for online implementation.
5.5 Conclusion

In this chapter, we have shown how to lower the PAPR of a system employing real OFDM signals by using a set of phase shifts for the constituting subcarriers. The set of phase shifts was determined using a linear programming approach that minimizes the complexity when the block length is high. Two methods for the phase shift vector optimization were devised, mainly AP-PTS and IP-PTS. Both methods were compared to the state-of-the-art techniques in the literature and were found to outperform the other methods.

Figure 5.14: Relative time consumption for a 2048 subcarrier OFDM system
Chapter 6

Hardware Implementation

In this chapter, we briefly demonstrate an implementation of the algorithms presented in Chapters 3, 4, and 5. The implementation was carried out using Texas Instruments TMS320C6713 DSK board.

6.1 Introduction

With the development of high speed communication technology and the internet, there exists a high demand on algorithms that prove to be amenable to real-life hardware implementation. This is in contrast with the method used by most publications, where computer software simulation is usually the preferred way.

The benefit of implementing innovated algorithms on processor chips is in proving that the algorithm is useful to be applied in industry. Out of many ways to implement systems on hardware processors, Digital Signal Processors (DSP) prove to be more beneficial and are able to replace Micro-Controller Units (MCU) or other processors in many situations that demand higher speed and more complex algorithms, yet it probably will cost less.
6.2 HARDWARE IMPLEMENTATION OF A TRANSCEIVER SYSTEM

6.2 Hardware Implementation of a Transceiver System

In implementing the algorithms proposed in Chapters 3, 4, and 5, we chose the TMS320C6713. The TMS320C6000 family, which has a very long instruction word architecture, has members with clock rates up to 1 GHz and cost about 150 dollars. The speed increase is largely a result of reduced geometries and improved technology. In the last couple of years, DSP manufactures have been developing chips with multiple DSP cores and shared memory for use in high-end commercial applications like network access servers handling many voice and data channels. DSP chips with special purpose accelerators like Viterbi decoders, turbo code decoders, multimedia functions, and encryption/decryption functions are appearing.

One of the major advantages of the C6713 board is its support of 16-bit fixed-point and 32-bit floating-point arithmetic. But as with all hardware implementations, real-time DSP applications are limited to cases where the required signal sampling rate is sufficiently less than the DSP instruction rate, so a reasonable number of instructions can be performed between samples. For example, a wideband radio frequency (RF) signal with a high carrier frequency can not be directly sampled and demodulated with a DSP. However, when the bandwidth of the RF signal is sufficiently less than the instruction rate, analog front-end circuits can be used to demodulate it to baseband In-phase and Quadrature components. The resulting signal can then be sampled at a rate equal to the bandwidth and processed by a DSP. Alternatively, an analog filter can be used to form the Hilbert transform of the RF signal, and then the original signal and its Hilbert transform can be sampled at a rate equal to the bandwidth and processed with a DSP.

DSPs have been extensively used in audio frequency applications, where many
instructions can be performed between samples. However, they are being used to process increasingly wide-band signals as the instruction rates of new generations increase. Special purpose VLSI chips and FPGAs have been used to implement limited DSP functions at very high rates.

6.3 Comparison Between Fixed-Point and Floating-Point Implementation

In Chapter 2, we described three main algorithms used for decoding LDPC codes. Many other variants exist in the literature. The complexity of the decoding algorithm lies in the check-node update operation. The different implementations of the update calculation for the check-node messages were also derived. The decoding algorithm presented in Section 2.4.1 and the update calculation for the check-node message in Equation (2.31) and stated here is too complex to implement on a DSP chip.
check-node update message was given by

$$m_{j}^{c,k+1} = 2 \arctanh \prod_{i=1,i \neq j}^{d_c} \tanh \frac{m_{i}^{c,k}}{2}.$$  \hspace{1cm} (6.1)

A simplified approximation, which is called the min-sum algorithm, provides a good approximation for low order check nodes. The algorithm was derived in Section 2.4.2. The update message was given by

$$m_{j}^{c} = \left( \prod_{i=1,i \neq j}^{d_c} \text{sgn}(m_{i}^{c}) \right) \left( \min_{i \neq j} \{ m_{i}^{c} \} \right).$$  \hspace{1cm} (6.2)

The LLR values have a range $[-\text{lim}, \text{lim}]$. The LLR values with fixed point precision constitute one sign bit and $k-1$ quantization bit. This results in $2^k$ bins. The bin edges are placed at $\pm \frac{i}{2^k-1}$, $\forall i = 0, 1, \cdots, 2^{k-1}$. The quantized value is set to the center of the bin at $\pm \frac{(2i+1)}{2^k}$, $\forall i = 0, 1, \cdots, 2^{k-1} - 1$. Figure 6.2 shows the nonuniform quantization grid for the LLR values. The reason for choosing a nonuniform quantization grid can be understood if we look at a plot of the tanh function as shown in Figure 6.3.

Figure 6.5 shows the frame error rate performance of a rate $\frac{1}{2}$, [3, 6] LDPC code...
with a block length of 1008. The decoding algorithm uses the check-node update of Equation (6.1). The performance of a quantization level of 4, 5, and 7 quantization bits for the LLR messages is compared with the performance of a floating-point implementation. It can be seen that for a quantization scheme with 7 bits, the fixed-point and the floating-point performance are almost identical.

Figure 6.4 shows a typical setup of the TMS320C6713 board connected to MATLAB to buffer and monitor the results. Figure 6.6 shows the frame error rate performance for the same LDPC using the min-sum algorithm. The check-node update equation is given in Equation (6.2). The performance of a quantization level of 4, 5, and 6 quantization bits messages is compared with the performance of a floating-point implementation of both algorithms. The inferior performance of the min-sum algorithm can be clearly seen when compared to the sum-product algorithm.
For the implementation of the nonuniform LDPC decoder and estimator, Figure 6.7 shows the performance of the nonuniform interpolator introduced in Chapter 3. The settings for the hardware implementation were set for $f_d T_S = 0.05$. The channel was implemented using the FPGA chip on the board. The implementation method was discussed in Section 3.8.1. It can be seen that even at a precision of 10 bits for the fixed-point implementation, the performance of the floating-point implementation is still better.
6.3. COMPARISON BETWEEN FIXED-POINT AND FLOATING-POINT IMPLEMENTATION

Figure 6.5: Comparison between fixed point implementation and floating point implementation using the update in Eq. (6.1)

Figure 6.6: Comparison between fixed point implementation and floating point implementation using the update in Eq. (6.2)
6.4 Conclusion

In this chapter, we discussed the advantages of the TMS320C6713 Texas Instruments board for real-time algorithm implementation. We showed how the use of fixed-point implementation affects the performance of the system. The performance of LDPC decoding algorithm with different precision bits was compared with the floating-point implementation. Furthermore, the performance of the nonuniform LDPC joint decoder and estimator with fixed and floating point implementation was shown for a typical fast fading channel. The performance of the algorithms is affected by the number of precision bits but the performance shows a floor at high SNR even at 10 precision bits.
Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis introduced new methods and algorithms for enhancing the performance of wireless communications using small cells mounted on high-speed rails. The thesis addressed two parts: the link from the small cell to the base station, and the link from the end-users to the small cell.

The communication channel for the link between the small cell and the base station is considered to be a fast fading channel. This is due to the effect of mounting the small cell on a high-speed rail. To guarantee a good QoS for the end user, the channel estimation algorithm must be fast and agile. In chapter 3, we introduced an algorithm for joint detection and estimation over Rayleigh flat fast fading channels using nonuniform interpolation over LDPC codes. The introduction of nonuniform interpolation grid as opposed to the uniform interpolation methods used in the literature makes the convergence of the estimation algorithm faster.

While nonuniform interpolation exists in the literature, its implementation is
highly unstable in iterative decoders, as it diverges with increasing number of iterations. We proposed a regularization method for the nonuniform interpolator that used a combination of Tikhonov regularization and the conjugate gradient method and was found to give satisfactory results to remedy the effects of the severely ill-conditioned interpolation matrices. The degree of regularization depends on the SNR of the received stream, so we proposed a scheme for estimating the SNR recursively every iteration. With different estimation methods, we were able to choose the method that performs best under the given constraints while not being computationally expensive.

For the link between the end-user and the small cell, power efficiency was of main concern since end-users rely on battery-powered devices most of the time. To enhance power efficiency, we sought to devise a shaping algorithm in Chapter 4 that resembles the sign-bit shaping algorithm introduced by Forney. The main differences between the proposed algorithm and the state-of-the-art algorithms are in the coding scheme used and the optimization method. The proposed algorithm used block coding methods as opposed to convolutional coding methods found in the state of the are implementations of the sign-bit shaping. This gives a great flexibility to the system designer as block coding is more suited for packet oriented communication. The optimization method used to find the minimum power sequence to send was found using LP methods. This was enabled due to the exploitation of the sparsity of the LDPC codes. The algorithm used a LP sparse solver to minimize the average transmitted power. The performance of the proposed methods was evaluated based on the needed average power to achieve a specific error rate for different constellation sizes.
In Chapter 5 we introduced LP methods to minimize the PAPR of OFDM signals. If not addressed properly, PAPR affects OFDM signals and deteriorates the performance of the system. In the state-of-the-art systems, many communication protocols use OFDM as the base multiplexing method for multiple users. The effects of degraded error rates and ICI caused by high PAPR and clipping are of major concern in system design. The multitude of existing PAPR reduction algorithms made it crucial that the proposed algorithm outperforms competing methods while having a computational complexity attractive for real-life applications. The proposed method relied on approximating the angles of the sine and cosine for a phase angle between $[0, \frac{\pi}{2}]$ in order to minimize the PAPR. This was done with all the subcarriers. For practical reasons, the number of phase rotations must be finite. So, the algorithm grouped a finite number of subcarriers and imposed the same phase shift on them. This degraded the performance but increased the throughput of the channel as less side-information was needed to be sent over the communication link.

Numerical simulations were conducted to compare the SLM, AP-PTS, and IP-PTS methods used in the literature. The results showed that the proposed algorithm outperformed the other algorithms. Nevertheless, the computational complexity showed that the proposed LP algorithm lies between the SLM and PTS methods. At high constellation dimensions, the original PTS method is very computationally expensive that it is deemed inappropriate.

In Chapter 6, we showed a real-time implementation of the proposed algorithms using the TMS320C6713 board. The importance of using DSP boards to evaluate the proposed algorithms is to validate the possibility of real-time implementation of such algorithms. The performance of LDPC codes for different fixed-point precisions
versus floating-point implementation was shown. Furthermore, the performance of the proposed nonuniform estimator and decoder was shown to degrade as the number of precision bits decreased. The gap between floating-point implementation and fixed-point implementation exist even for higher precision.

7.2 Proposed Future Work

Three main areas of interest arise from such work, they were divided based on feasibility and time consumption for future work:

- Channel estimation: The algorithm for estimating the channel using nonuniform interpolation can be extended to special cases of Nakagami and Rayleigh fading channels by allowing the conjugate gradient method to find the vector of frequency coefficients that conforms with the autocorrelation function of the channel in the time domain. This suggests the use of second order cone programming in solving the problem. The difficulty with such an approach lies mainly in identifying the type of propagation and fading impeding the channel, and estimation of the required variables for each different fading model. Also, the incorporation of multipath fading channels should be considered. The incorporation of OFDM-LDPC channel estimation and decoding over fast fading channel should also be considered as OFDM is extremely sensitive to Doppler spread effects.

- Signal Shaping: Extending the shaping of signals into multidimensional constellations and incorporating the modulation scheme in the optimization process. This can be done by changing the LP optimization problem into a second order cone problem. The computational complexity is higher than LP, but the
extension allows for different types of modulation schemes. This is attractive for software defined radio systems where the transceiver interacts with different systems at the same time.

- **OFDM PAPR:** The extension of the LP PAPR of real OFDM signals to complex valued one should be investigated. The only problem that arises from that formulation is the introduction of linear matrix inequalities, which for high number of points, lead to a system with a large number of constraints. Obviously, a number of simplifications need to be postulated in order to solve the problem.
References


