Title
Energy Efficiency Maximization of Practical Wireless Communication Systems

Permalink
https://escholarship.org/uc/item/6kh0b5fx

Author
ERASLAN, EREN

Publication Date
2013-01-01

Peer reviewed|Thesis/dissertation
Energy Efficiency Maximization of Practical Wireless Communication Systems

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Electrical Engineering

by

Eren Eraslan

2013
Energy consumption of the modern wireless communication systems is rapidly growing due to the ever-increasing data demand and the advanced solutions employed in order to address this demand, such as multiple-input multiple-output (MIMO) and orthogonal frequency division multiplexing (OFDM) techniques. These MIMO systems are power hungry, however, they are capable of changing the transmission parameters, such as number of spatial streams, number of transmitter/receiver antennas, modulation, code rate, and transmit power. They can thus choose the best mode out of possibly thousands of modes in order to optimize an objective function. This problem is referred to as the link adaptation problem.

In this work, we focus on the link adaptation for energy efficiency maximization problem, which is defined as choosing the optimal transmission mode to maximize the number of successfully transmitted bits per unit energy consumed by the link. We model the energy consumption and throughput performances of a MIMO-OFDM link
and develop a practical link adaptation protocol, which senses the channel conditions and changes its transmission mode in real-time. It turns out that the brute force search, which is usually assumed in previous works, is prohibitively complex, especially when there are large numbers of transmit power levels to choose from. We analyze the relationship between the energy efficiency and transmit power, and prove that energy efficiency of a link is a single-peaked quasiconcave function of transmit power. This leads us to develop a low-complexity algorithm that finds a near-optimal transmit power and take this dimension out of the search space. We further prune the search space by analyzing the singular value decomposition of the channel and excluding the modes that use higher number of spatial streams than the channel can support. These algorithms and our novel formulations provide simpler computations and limit the search space into a much smaller set; hence reducing the computational complexity by orders of magnitude without sacrificing the performance.

The result of this work is a highly practical link adaptation protocol for maximizing the energy efficiency of modern wireless communication systems. Simulation results show orders of magnitude gain in the energy efficiency of the link. We also implemented the link adaptation protocol on real-time MIMO-OFDM radios and we report on the experimental results. To the best of our knowledge, this is the first reported testbed that is capable of performing energy-efficient fast link adaptation using PHY layer information.
The dissertation of Eren Eraslan is approved.

Mihaela van der Schaar

Gregory J. Pottie

Songwu Lu

Babak Daneshrad, Committee Chair

University of California, Los Angeles

2013
To mom and dad . . .
# Table of Contents

1 **Introduction** ................................................................. 1

   1.1 Importance of Energy Efficiency in Modern Wireless Communication Systems ................................................................. 2

   1.2 Research Efforts on Link Adaptation ........................................ 4

   1.3 Contributions of the Work .................................................. 8

   1.4 Organization of Dissertation .............................................. 11

2 **MIMO-OFDM System Model** ................................................ 13

3 **Link Adaptation for Energy Efficiency Maximization** .................. 17

   3.1 Throughput Maximization Problem ......................................... 19

   3.2 Energy Efficiency Maximization Problem .................................. 19

   3.3 Energy Consumption Model ................................................ 21

      3.3.1 Energy Consumption of the RF chains ................................. 22

      3.3.2 Energy Consumption of the Baseband Blocks ......................... 23

   3.4 Packet Error Rate (PER) Prediction Model ............................... 26

      3.4.1 PER Prediction Using Statistics from Upper Layers ............... 26

      3.4.2 PER Prediction Using Link Quality Metrics (LQM) ................ 27

      3.4.3 PER Prediction via Analytical Modeling .............................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 Link Adaptation Protocol Using Brute Force Search</td>
<td>32</td>
</tr>
<tr>
<td>3.5.1 Practicality and Complexity Issues</td>
<td>36</td>
</tr>
<tr>
<td>4 Practical Low-Complexity Link Adaptation for Energy Efficiency Max-</td>
<td>39</td>
</tr>
<tr>
<td>imization</td>
<td></td>
</tr>
<tr>
<td>4.1 SVD Based PPSNR Formulation</td>
<td>42</td>
</tr>
<tr>
<td>4.2 Energy Efficiency and Transmit Power Relationship</td>
<td>45</td>
</tr>
<tr>
<td>4.2.1 Low-complexity Iterative Solution for Finding Near-Optimal Trans-</td>
<td>50</td>
</tr>
<tr>
<td>mit Power</td>
<td></td>
</tr>
<tr>
<td>4.3 SVD Based Pruning of the Search Space</td>
<td>53</td>
</tr>
<tr>
<td>4.4 Low-Complexity Link Adaptation Protocol</td>
<td>57</td>
</tr>
<tr>
<td>4.5 How to Determine How Often We Should Be Sounding?</td>
<td>59</td>
</tr>
<tr>
<td>5 Simulations</td>
<td>62</td>
</tr>
<tr>
<td>5.1 Simulation Setup and Assumptions</td>
<td>63</td>
</tr>
<tr>
<td>5.1.1 Channel Model</td>
<td>64</td>
</tr>
<tr>
<td>5.1.2 Simulation Parameters</td>
<td>65</td>
</tr>
<tr>
<td>5.2 Simulation Results</td>
<td>68</td>
</tr>
<tr>
<td>5.2.1 Energy Efficiency Results and Mode Selection Trends</td>
<td>68</td>
</tr>
<tr>
<td>5.2.2 Comparison to Other Strategies and Throughput Maximization</td>
<td>73</td>
</tr>
<tr>
<td>Results</td>
<td></td>
</tr>
<tr>
<td>5.2.3 Adaptive Sounding Period Related Simulation Results</td>
<td>78</td>
</tr>
</tbody>
</table>
### List of Figures

1.1 Energy consumption breakdown of a base station and a smartphone. . . 3

2.1 Transmitter and receiver block diagrams for MIMO-OFDM system. . . 14

3.1 Transmitter and receiver block diagrams for energy consumption model. 22

3.2 Experimental SNR results. The transmitter was in the WISR laboratory and the receiver was pushed on a cart along the indicated path in the hallways. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

3.3 Energy-aware fast link adaptation protocol. The receiver estimates the channel using the sounding packet, then computes PPSNRs, finds the optimal mode $m^*$ and sends it to the transmitter using the ACK packet. The transmitter uses the optimal mode until the next sounding packet is transmitted. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

4.1 Packet success rate vs transmit power for four different 5-tuples. . . . 48

4.2 Energy efficiency vs transmit power for four different 5-tuples. Shaded area defines the feasible region for transmit power. . . . . . . . . . . . . 48

4.3 $\text{Prob}(\gamma_1 - \gamma_2 < \Delta) \times \sigma_2/\sigma_1$ for various $\Delta$ values. We generated 10 million instances of random $2 \times 2$ channel matrices each with entries that are ZMCSCGs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

4.4 Low-complexity link adaptation procedure. . . . . . . . . . . . . . . . 58
4.5 Sounding period determination algorithm. Values are examples only. $T_{max}$ and $T_{min}$ are upper and lower limits on $T_{sound}$. 60

5.1 Multipath mobile channel 66

5.2 Energy Efficiency vs time (Upper Plot), and received SNR for the 1st antenna vs time (measured via the sounding packet which uses the maximum $P_t$, the actual SNR during the data transmission is different since the link adaptation algorithm keeps adapting the transmit power). 70

5.3 Optimal mode choices vs time. 71

5.4 Throughput vs time (upper plot), PER vs time (middle plot), and SNR vs time (lower plot). Lower plot shows the received SNR at the 1st antenna when all 4 transmit antennas are used with maximum $P_t$. 76

5.5 Energy efficiency vs time (upper plot), and received SNR for the 1st antenna vs time (lower plot). 77

5.6 Adaptive sounding period. 79

5.7 Complexity reduction (left y-axis) and energy efficiency (right y-axis) performances for different $\tau$ values. 81

6.1 Detailed rx baseband block diagram. 83

6.2 High level block diagram of the testbed. 86

6.3 Screen-shot of the demo GUI, taken during an experiment. 86

6.4 Measurement environment and pictures of the transmitter and receiver testbeds. 88
6.5 Energy Efficiency vs time (Upper Plot), and received SNR for the 1st antenna (measured via the sounding packet which uses the maximum $P_t$) vs Time (lower plot)................................. 91

6.6 Optimal mode parameters vs time................................................... 92

6.7 Optimal mode statistics for the default experiment............................... 95

6.8 Optimal mode statistics for the experiment where we set $E_{\text{base}} = 0$... 96

6.9 Optimal mode statistics for the experiment where we set $E_{\text{rf}} = 0$..... 97

6.10 Optimal mode statistics for the experiment where we set the PA efficiency to $\eta = 0.70$.......................................................... 99

6.11 Optimal mode statistics for the experiment where we used fixed number of receive antennas, $N_r = 4$........................................ 100

6.12 Predicted and observed PER results. 0 dB PPSNR back off.................... 102

B.1 BER for BPSK. Transmit SNR is per bit and per tx antenna.................. 121

B.2 BER for 16QAM. Transmit SNR is per bit and per tx antenna................. 122

B.3 BER for 4×5 QPSK for different $\sigma_e$ values. Transmit SNR is per bit and per tx antenna. $\sigma_e = 0$ corresponds to perfect CE. ZF curves are included for comparison purposes with C. Wang’s paper [52]. These parameter values are taken from Fig. 2 of [52] for comparison purposes.............................. 123
# List of Tables

3.1 Transmission parameters and choices for the UCLA mode-rich MIMO testbed ........................................ 18

3.2 Coefficients for energy consumption calculations ........................................ 24

3.3 Coefficients for the log-linear approximation ........................................ 31

4.1 Computational complexities of PPSNR calculation methods .......................... 44

5.1 Simulation parameters ........................................ 67

5.2 Simulation results for energy efficiency maximization ................................ 72

5.3 Simulated protocols and results ........................................ 75

6.1 Default experiment parameters ........................................ 89

6.2 PER prediction performances ........................................ 103

6.3 Computational complexity comparison ........................................ 105
Acknowledgments

First and foremost, I would like to thank my advisor Professor Babak Daneshrad for his insightful guidance and encouragement throughout my doctoral studies. His sincerity and friendly approach to his students makes him a wonderful advisor. I am also grateful to my committee members Professor Mihaela van der Schaar, Professor Gregory Pottie and Professor Songwu Lu for agreeing to be in my doctoral committee and for their valuable inputs.

I would like to thank all of my friends who made my time at UCLA enjoyable and productive. In particular, I would like to thank my lab mates; Chung-Yu Lou, Johnny Wang, Mansour Rachid and Gaelen Pereira. The research discussions I had with Chung-Yu were very useful throughout the process, and without Johnny, the implementation of the algorithms on the radios could not be this easy. I would also like to thank my friends Alican Gök and Arya Ardakani, who are the first friends that I had in the US. Alican is one of the few people whom you can have informative and fun talks on a large variety of topics ranging from soccer to classical music. Arya has always challenged me with new ideas about the industry, which helped me in thinking outside the box. Felisha Corona deserves special thanks for making me feel at home away from home, and for proofreading my lengthy research papers.

Finally, I would like to thank my mom, dad and sister for their endless love and support. Their love makes me the wealthiest person in the world.
Vita

1987 Born, Kirsehir, Turkey.

2007–2008 Design Engineer, Part Time
Optisis Telecommunication Inc., Ankara, Turkey

2008 B.S., Electrical and Electronics Engineering, high honor
Bilkent University, Ankara, Turkey

2008–2013 Graduate Student Researcher
Electrical Engineering, University of California, Los Angeles

2009 Teaching Assistant
Electrical Engineering, University of California, Los Angeles

2010 M.S., Electrical Engineering
University of California, Los Angeles

2010–2013 PhD Student, Electrical Engineering
University of California, Los Angeles

2010 Wireless Communication Systems Intern
Silvus Technologies Inc., Los Angeles, CA

2012 Communication Systems Intern
Broadcom Corp., Irvine, CA


**Posters and Talks**


CHAPTER 1

Introduction

Multiple-input multiple-output (MIMO) based communication systems provide significant improvements in capacity by increasing the robustness of a link (space-time block codes - STBC) or by improving the spectral efficiency (spatial multiplexing - SM). Hence, MIMO techniques have found their way into many high-speed wireless communication standards such as IEEE802.11n [1], IEEE802.11ac and IEEE802.16e, to name a few. Most laptops nowadays have more than one antenna, Wi-Fi routers have at least two antennas, and cellular base stations are equipped with even more antennas. A secondary impact of integrating MIMO into a radio is that it has a multiplicative impact on the number of modes available to close the link, i.e. satisfying the throughput and delay requirements of the application. For instance, a MIMO radio can change system parameters, such as the number of spatial streams, number of transmitter/receiver antennas, modulation, code rate, and transmit power. A MIMO enabled mode-rich radio can thus sense its surroundings and choose the best mode that optimizes an objective function. This problem is generally referred to as the link adaptation problem.

Our aim in this work is to develop a practical link adaptation protocol for maximiz-
ing the energy efficiency or throughput of a wireless link. Our definition of practicality implies that the protocol should (1) track and respond to the fast fading changes in the channel, (2) be implementation friendly in terms of the hardware complexity, and (3) be applicable to a large variety of coded MIMO-OFDM systems.

1.1 Importance of Energy Efficiency in Modern Wireless Communication Systems

It has been reported that the total energy consumption of the information and communication technologies (ICT) sector takes up more than 3% of worldwide energy consumption [2] and this share is predicted to increase in the near future. The energy consumption of wireless networks is a major contributor to this and is growing much faster than the overall energy consumption growth of the sector. This is mainly due to the exponentially increasing data demand, and the advanced techniques, such as MIMO, employed in modern wireless systems to meet this demand.

High CO$_2$ emissions and expensive electricity bills for the cellular network operators are the inevitable results of this high energy consumption. The operators are the top energy consumers in many countries and their energy bill accounts for approximately 18% of the total operational expenditure in the European market and at least 32% in India [3, 4]. The most significant portion of the consumption comes from the communication related circuits of the base stations (BS). As shown in Fig. 1.1(a), 60%-70% of the energy consumption comes from the RF and baseband related circuits. Hence, making these radio related sections of the cellular networks will have big impact on the
environment and the energy costs of the operators.

At the user side, the energy consumption problem is also very significant as the modern smartphones are very power hungry and battery technology has not evolved as
fast as needed to meet the ever increasing energy needs of today’s mobile applications. Most of the smartphone users primarily complain about the battery up-time of their devices according to the customer surveys [4]. A significant portion, about 40%, of the total energy consumption comes from the wireless communication chips in smartphones as we show in Fig. 1.1(b) during a video streaming application [5]. Therefore, maximizing the energy efficiency of the wireless links will directly increase the up-time of the batteries and provide a much better user experience. Additionally, given that hundreds of millions of laptops and cell phones are sold each year, a reduction in the battery requirement would also have a huge environmental impact.

1.2 Research Efforts on Link Adaptation

Past work dealing with link adaptation has mostly focused on maximizing the throughput [6–10]. When the aim is to maximize the throughput of a link, one approach is to perform link adaptation at the MAC layer according to observed packet error rate (PER) statistics. The AutoRate Fallback [6] and its enhancements [7, 10] have been widely used in legacy WLANs for rate adaptation. In these MAC-based protocols, all the available modes are sorted in terms of their rates, and the radio switches to a lower/higher rate after collecting the PER statistics of these modes from previous trials. For example, in [10], a number of neighbor modes, where the neighborhood is defined in terms of their rates, are probed in a window, and the link adaptation algorithm switches to the higher performance mode based on the PER statistics of the probed modes. These algorithms are simple to implement, however they require probing of
modes. As a result, they are inherently slow in converging to the optimum mode and observe severe packet losses, especially in highly dynamic channels.

In order to make the adaptation faster, PHY layer metrics and their relationship to the PER performance have been considered. In [11], authors simulated the WiMAX system with a specific channel model and determined the SNR thresholds for switching from one mode to another. However, SNR information alone is not sufficient for determining the performance of a MIMO system. In some recent works, the post-processing SNR (PPSNR), which can be seen as the SNR at the output of the MIMO decoder, is leveraged to arrive at more accurate estimates of the link’s PER performance. PPSNR information is mapped to different link quality metrics (LQM) and the mapping from LQMs to the PER is determined via extensive simulations and curve fitting methods [9]. A major problem with these LQM methods is that they require extensive simulations to characterize the performance of the system. It may not be practical to simulate and calibrate the mappings for all channel models and all possible modes via simulations. A pure analytical modeling is therefore preferred to predict PER without extensive simulation and calibration. PPSNR based methods are more accurate in predicting the PER, however, they might be overly complex to implement as will be seen in the ensuing sections.

All the aforementioned works have focused on maximizing the throughput without considering the energy consumption of the link. In most systems, however, maximizing the energy efficiency of the link is as critical, if not more critical than maximizing its throughput, as we explained in the previous section. Given that the energy consumption of wireless communication components in both base stations and mobile devices is the
major contributor to the overall energy consumption, maximizing the energy efficiency of these links will have a significant impact on the environment, operator costs, and user experience. The energy efficiency maximization problem is arguably more challenging than the throughput maximization problem due to the larger search space, since a well designed energy-aware link adaption protocol must strike the proper balance between the energy consumption of the radios, the modulation coding and antenna scheme to be used and the application’s QoS requirements which includes throughput, latency and PER constraints. A survey of the literature reveals several papers dealing with the single-input single-output (SISO) link adaptation problem [12,13], and others such as [14] that consider an uncoded system or employ idealistic capacity results [15,16] in their derivations. However, to the best of our knowledge none of these works have closed the loop between theory and experimental feedback into the theory, as no testbeds have been reported that implement these algorithms.

In their initial work [12], Cui and Goldsmith formulated the link adaptation problem for minimizing the energy required for transmitting a certain amount of information. They modeled the energy consumption of the baseband and radio frequency (RF) circuits, as well as the transmit energy, which is consumed by the power amplifier (PA). A SISO, single carrier, narrowband system operating in an AWGN channel was assumed. The authors extended the energy-aware link adaptation problem to MIMO systems in [17]. The underlying system, however, was still a single carrier, narrowband system with one or two antennas at either the transmitter or receiver. Contrary to the traditional belief that MIMO systems are more energy efficient than SISO, it was shown that for short-range fixed-rate applications, a SISO system can beat both 2x1 and 2x2
Alamouti based MIMO systems as far as energy efficiency is concerned. This result is due to the fact that circuit energy dominates at short distances. The authors in [17] provide analysis and simulations for the average performance in Rayleigh fading channel, but do not address the practicality of the approach as the work assumes an uncoded system and brute force search for optimization. Other published energy-efficient link adaptation works also exist [13, 18], but are limited to SISO wideband systems in frequency selective channels and focus on energy-efficient waterfilling strategies, assuming the channel state is available at the transmitter.

Bougard et al. in [19], proposed link adaptation for WLANs to minimize the energy consumption of the link. The energy consumption model focused on the PA and ignored other sources of energy draw associated with various baseband and RF components. The optimization problem was formulated based on an idealistic channel capacity expression. The assumption was that the PER is equal to 1 if the capacity of the channel is less than a threshold, and assumed to be 0 otherwise. These thresholds were determined via simulations.

In [14], Kim et al. formulated the energy-efficient link adaptation problem as a geometric program. In this work, both STBC and SM modes were included in the search space. Additionally, bandwidth and transmit power were considered as optimization parameters. The bit error rate (BER) expressions were derived for the average performance in fading channels. However, it can be argued that significantly better performance can be achieved by responding to the fast fading instead of the average channel. Trends and guidelines were provided for short and long range applications. These trends give very useful insights. However, the underlying system was uncoded
and more importantly, only flat fading channels were considered. There are other works that focus on the energy efficiency problem in cellular networks [15, 16], and solve the problem using convex optimization techniques. Miao et al. in [20] solved the optimal power allocation and scheduling problems by closed-form expressions for energy efficient orthogonal frequency division multiple access (OFDMA) networks. The major drawback of these works is that they employ channel capacity instead of throughput in the problem formulation; therefore, their solution would pick unrealistically low transmit powers. Moreover, they can not adapt the modulation and code rate. Their search space is also usually limited to transmit power and MIMO configuration only. Another class of papers [21, 22] focus on base station switching, cell breathing and base station sleeping type of techniques to improve the energy efficiency in cellular networks.

Comprehensive literature surveys on energy efficient wireless communications can be found in [4, 23, 24].

1.3 Contributions of the Work

In this dissertation, we propose a low-complexity link adaptation protocol for energy efficiency maximization of practical MIMO-OFDM systems. The proposed protocol minimizes the total energy consumption of the link required to successfully transmit information bits while maintaining an application’s QoS requirements. We sound the channel periodically and calculate the metric called post-processing SNR (PPSNR). We then employ a fully mathematical model for the packet error rate (PER) performance of the system based on the PPSNR and we predict the energy efficiency performance
of each mode given the channel conditions.

It turns out that the brute force search over all possible modes, which is usually assumed in previous works, is prohibitively complex especially when there are large numbers of transmit power levels to choose from. For instance, the computation of the PPSNRs for every possible mode is a major challenge in terms of the complexity, since it requires many matrix inversions. In order to solve the complexity problem, we first derive a novel matrix-inversion-free PPSNR formulation by expressing it in terms of the singular values.

The search space in advanced MIMO systems can be very large, especially due to the large number of choices available for the transmit power and the MIMO configuration. In this work, we analyze the energy efficiency and transmit power relationship and prove that energy efficiency of the link is a single-peaked quasiconcave function of transmit power. This proof leads us to develop a low-complexity algorithm that finds a near-optimal transmit power and takes this dimension out of the search space. We further prune the search space by analyzing the singular value decomposition (SVD) of the channel and excluding the modes that use a higher number of spatial streams than the channel can support. These algorithms limit the search space into a much smaller set and hence reduce the computational complexity of the protocol by orders of magnitude without sacrificing the performance. Additionally, we propose solutions to other practical problems such as determination of the sounding period.

The result is a highly practical fast link adaptation algorithm which delivers orders of magnitude energy savings compared to poorly chosen static modes. The same algorithm, when used for throughput maximization, delivers significant performance
improvement over the MAC based approaches [10].

Below, we summarize the contributions of this work;

• We address the problems of the existing MAC based throughput maximization algorithms, which are slow in their response to the changing environmental conditions. We propose a fast-responsive link adaptation protocol for coded MIMO systems. It is more realistic than other published works on energy-aware link adaptation which assume uncoded systems or employ channel capacity based derivations.

• Practical problems of the proposed algorithm, such as computational complexity, and sounding period determination are addressed by novel closed form solutions and iterative algorithms.

  – We derive a novel matrix-inversion-free PPSNR formula based on the singular value decomposition of the channel matrix.

  – We analyze the energy efficiency and transmit power relationship and prove that the energy efficiency is a single-peaked quasiconcave function of transmit power. Based on this proof, we derive upper and lower bounds on the optimal transmit power and develop an iterative algorithm to find the optimal value.

  – We propose a simple method to further prune the search space using the singular value ratios of the channel.

  – We propose a sounding period determination algorithm to adapt the sounding period in mobile channels where the rate of the mobility also changes.
Proposed adaptive algorithm changes the sounding period without explicit knowledge of the Doppler frequency.

- Feasibility was proven via experimental validation of the algorithms on a real-time MIMO-OFDM testbed and we report the experimental results on mode selection trends. We present various real-time experiments that we performed in an indoor environment, and we show the mode selection trends and the energy efficiency results.

- Additionally, we derive the PPSNR for MMSE decoders operating in the presence of the channel estimation errors. This novel PPSNR formulation is a very accurate indicator of the error rate of a practical MIMO system operating with imperfect channel estimation matrices.

1.4 Organization of Dissertation

The remainder of the dissertation is organized as follows. We first describe the MIMO-OFDM system model in Chapter 2. The system model that we considered is meant to be generic and can be applied to many MIMO-OFDM systems, such as WiMAX and 802.11ac, with small modifications. In Chapter 3, we begin with the definition of the link adaptation problem. We then present the energy consumption and PER prediction models. A fast link adaptation protocol based on the brute force search method is presented in Chapter 3. We also discuss the computational complexity of the brute force search method in Chapter 3.

We present our solutions to the complexity problem in Chapter 4 by first introduc-
ing the novel PPSNR formulation. We then analyze the energy efficiency and transmit power relationship and develop an algorithm to find the optimal transmit power in a low complexity manner. In the last part of Chapter 4, we propose the search space pruning method based on the singular value ratios, and we finally present our low-complexity link adaptation protocol. Chapter 5 is devoted to realistic link level simulations where we analyze the performance of the proposed protocol and compare it to other methods. In Chapter 6, we present the implementation of the link adaptation protocols on real-time MIMO-OFDM testbeds and we discuss the experimental results. Chapter 7 concludes the dissertation.
CHAPTER 2

MIMO-OFDM System Model

We consider a generic MIMO-OFDM system, depicted in Fig. 2.1, where the transmitter is equipped with $N_t$ antennas, and the receiver uses $N_r$ antennas.

Information bits are first encoded with a convolutional encoder and punctured to achieve the desired code rate. The encoded bits are then parsed and interleaved over multiple spatial streams and subcarriers. The bits in each spatial stream are mapped to symbols by a quadrature-amplitude-modulator (QAM) which is followed by the spatial mapper. The spatial mapping (or antenna mapping) is a linear operation that transforms $N_{ss}$ (number of spatial streams) dimensional symbol vector into $N_t$ dimensional signal which is then transmitted after the inverse fast Fourier transform (IFFT) and cyclic prefix (CP) addition operations.

At the receiver, $N_r$ dimensional samples of the received signal in time domain are transformed back into frequency domain by the fast Fourier transform (FFT) operation after removal of the CP. MIMO decoder is then used for separating individual spatial streams from the total received signal. Soft bits are then fed to the Viterbi decoder after the deinterleaving, deparsing and depuncturing operations. The system model
is meant to be generic and applicable to most, if not all, MIMO-OFDM systems with minor modifications. The system model we assumed in this work is based on IEEE 802.11n, however can be aligned with WiMAX, 802.11ac and other MIMO-OFDM based standards with small modifications.

After the CP removal and FFT operations, the $N_r \times 1$ received signal at each subcarrier\(^1\) can be expressed as

$$y = \sqrt{P_t} \tilde{H} R x + n$$

(2.1)

where $x$ is the $N_{ss} \times 1$ data symbol vector, $R$ is the $N_t \times N_{ss}$ precoding matrix, which determines the mapping from spatial streams to transmit antennas, $\tilde{H}$ is the $N_r \times N_t$ channel matrix, and $n$ is the $N_r \times 1$ additive Gaussian noise vector with zero mean and covariance matrix $N_0 I$. Transmitted QAM symbols and precoding matrix are normal-

\(^1\)Time and subcarrier indices are omitted for brevity in the signal model here. The signal should also be scaled by number of subcarriers here if $P_t$ denotes the total power of the OFDM signal, however we dropped that notation for presentation simplicity. In the following chapters, the reader will easily be aware when we start using $P_t$ as the total power of the OFDM signal per antenna.
ized so that the transmitted signal has a power of $P_t$ at each antenna, i.e. $E[\mathbf{x}\mathbf{x}^*] = \mathbf{I}$ and $\|\mathbf{r}_l\|^2 = 1 \ \forall l \in 1, ..., N_t$ where $\mathbf{r}_l$ is the $l^{th}$ column of the precoding matrix, $\mathbf{R}$.

The task of the MIMO decoder at the receiver side is to achieve the estimate of the transmitted data symbol vector $\mathbf{x}$. The maximum likelihood (ML) decoder achieves the optimal error rate performance. However, these types of nonlinear decoders, including the near-optimal sphere decoder and its variants, are usually not suitable for practical systems due to their high complexity. Linear decoders, such as zero-forcing (ZF) and minimum mean-squared error (MMSE), achieve suboptimal performance; however, they are widely used in practical systems due to their low complexity implementations.

Linear decoders combine the elements of the received signal vector $\mathbf{y}$ by applying an $N_{ss} \times N_r$ matrix $\mathbf{W}$ to the received signal and obtain the estimate of the transmitted symbol vector $\hat{\mathbf{x}}$.

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} = \sqrt{P_t}\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}\mathbf{x} + \mathbf{Wn} \tag{2.2}$$

Hence, at the output of the decoder, the post-processing SNR (PPSNR) of the $k^{th}$ spatial stream is calculated as

$$\gamma_k = \frac{P_t \left| (\mathbf{W}\mathbf{H})_{k,k} \right|^2}{P_t \sum_{l \neq k} \left| (\mathbf{W}\mathbf{H})_{k,l} \right|^2 + N_0 \left( \mathbf{W}\mathbf{W}^* \right)_{k,k}} \tag{2.3}$$

where we denoted the effective channel as $\mathbf{H} = \tilde{\mathbf{H}}\mathbf{R}$ for simplicity, and $\left( ... \right)_{k,l}$ denotes the $(k,l)^{th}$ entry of the matrix of interest. $\mathbf{H}$ is either a tall or a square matrix as $N_r \geq N_{ss}$.

The MMSE decoder is optimal among the class of linear decoders in the sense of
minimizing the mean-squared error, and calculated as follows.

\[ W = \frac{1}{\sqrt{P_t}} \left[ H^*H + \frac{N_0}{P_t} I \right]^{-1} H^* \]  

(2.4)

For the MMSE decoder, the PPSNR can be simplified to [25]

\[ \gamma_k^{(MMSE)}(k) = \frac{1}{\left( \left[ \frac{P_t}{N_0} H^*H + I \right]^{-1} \right)_{k,k}} - 1. \]  

(2.5)

The zero-forcing (ZF) decoder, on the other hand, nulls out the interference from other spatial streams instead of minimizing the mean-squared error, and it is calculated as

\[ W = \frac{1}{\sqrt{P_t}} [H^*H]^{-1} H^*. \]  

(2.6)

For the ZF decoder, the PPSNR simplifies to

\[ \gamma_k^{(ZF)} = \frac{P_t/N_0}{\left( [H^*H]^{-1} \right)_{k,k}}. \]  

(2.7)

The PPSNR indicates the ratio of the power of the signal of interest to the power of the residual interference from other spatial streams plus the residual noise power. It is a good indicator of the system’s error rate performance and has been widely used for predicting the packet error rates for link adaptation purposes [9]. We employ the PPSNR for error rate calculations in link adaptation process.
CHAPTER 3

Link Adaptation for Energy Efficiency

Maximization

MIMO based communication systems provide significant improvements in performance by increasing the robustness of the link (space-time block codes - STBC), and/or by improving the spectrum efficiency (spatial multiplexing - SM). A secondary impact of integrating MIMO into a radio is that it has a multiplicative impact on the number of modes available to close the link, i.e. satisfying the throughput and delay requirements of the application. For instance, a MIMO radio can change system parameters, such as the number of spatial streams, number of transmitter/receiver antennas, modulation, code rate, and transmit power. We call these systems mode-rich radios.

As an example, we listed the parameter choices in Table 3.1 for the mode-rich radio testbed\(^1\) that we use in our laboratory. This MIMO-OFDM testbed has four antennas, hence it has sixteen choices for the transmitter/receiver antenna configuration. Like many other modern MIMO radios, it can also change the number of spatial streams

\(^1\)Details about the testbed are explained in Chapter 6 where we present the experiments on the link adaptation protocol.
Table 3.1: Transmission parameters and choices for the UCLA mode-rich MIMO testbed

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Choices</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of transmit antennas</td>
<td>$N_t$</td>
<td>4</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>Number of receive antennas</td>
<td>$N_r$</td>
<td>4</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>Number of spatial streams</td>
<td>$N_{ss}$</td>
<td>4</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>Antenna processing</td>
<td>–</td>
<td>3</td>
<td>SM, STBC, Delay Diversity</td>
</tr>
<tr>
<td>Modulation</td>
<td>$q$</td>
<td>4</td>
<td>BPSK, QPSK, 16QAM, 64QAM</td>
</tr>
<tr>
<td>Code rate</td>
<td>$r$</td>
<td>4</td>
<td>1/2, 2/3, 3/4, 5/6</td>
</tr>
<tr>
<td>Transmit power</td>
<td>$P_t$</td>
<td>64</td>
<td>-3 dBm to 20 dBm</td>
</tr>
<tr>
<td>RF carrier</td>
<td>–</td>
<td>2</td>
<td>2.4 GHz and 5 GHz bands</td>
</tr>
</tbody>
</table>

and the antenna processing method in order to adjust the data rate and robustness of the system. It can change the number of independent data streams from one up to four and can employ different methods of mapping these spatial streams to the physical antennas (antenna processing). In addition, like almost all modern high speed wireless communication systems, this testbed is capable of changing the modulation type (BPSK, QPSK, 16QAM and 64 QAM) and the code rate. The transmit power in modern communication systems usually have many choices depending on the power amplifiers used in the system. This testbed has sixty four choices for the transmit power.

When we consider all the combinations formed by these parameter choices shown in Table 3.1, we end up with thousands of modes available to be used for data transmission. A MIMO enabled mode-rich radio can thus sense its surroundings and choose the
best mode that optimizes an objective function. This problem is generally referred to as the link adaptation problem. In this work, we mainly focus on link adaptation for maximizing the energy efficiency of the link. We also included the throughput maximization both in simulations and implementation for comparison purposes.

3.1 Throughput Maximization Problem

Past work dealing with the link adaptation problem has mostly focused on maximizing the link throughput subject to QoS constraints [6–11]. The throughput maximization problem can be formally written as

\[
\text{maximize } \frac{(1 - \text{PER})L}{T} \text{ (bps)}
\]

subject to \( \text{PER} \leq \text{PER}_{\text{max}} \) (3.1)

where \( \text{PER}_{\text{max}} \) is the maximum allowed instantaneous (or short term) packet error rate which is usually determined by the application, \( L \) is the packet length in number of information bits, and \( T \) denotes the total time needed to transmit the packet including the packet overheads and the time spent on the MAC layer related tasks. We used the notion of packet error rate since we focus on a packet based communication system, however one can easily use PER, block error rate (BLER) or frame error rate (FER) interchangeably.

3.2 Energy Efficiency Maximization Problem

When our objective is to communicate \( L \) bits across a channel, a throughput maximization oriented approach will reduce the total transmission time, but it will not
necessarily maximize the energy efficiency of the system. The objective of energy-aware link adaptation on the other hand, is to minimize the total energy consumed in the link per successfully received bit, which is equivalent to maximizing the number of successfully received bits normalized by the total energy consumption of the link. The energy efficiency maximization problem can thus be formulated as

\[
\text{maximize } \quad EE = \frac{(1 - PER)L}{E_{\text{total}}} \quad \text{(bits/Joule)}
\]

subject to \( PER \leq PER_{\text{max}} \) \hspace{1cm} (3.2)

\[ TH \geq TH_{\text{target}} \]

where \( TH \) indicates the instantaneous throughput and \( TH_{\text{target}} \) is the target throughput enforced by the application. The energy efficiency \( (EE) \) here is defined as the number of successfully transmitted bits per unit energy consumption.

\[
EE = \frac{(1 - PER)L}{E_{\text{total}}} \quad \text{(3.3)}
\]

The numerator is simply the number of successfully transmitted bits and \( E_{\text{total}} \) is the total energy consumption of the link, which includes the energy consumed in the baseband portions and the RF transceiver sections both at the transmitter and receiver, to transmit a packet carrying \( L \) information bits\(^2\).

We try to maximize the \( EE \) by adapting these six system parameters: transmit power \( (P_t) \), number of transmit antennas \( (N_t) \), number of receive antennas \( (N_r) \), number of spatial streams \( (N_{ss}) \), modulation \( (q) \), and code rate \( (r) \). Search space of the energy

\(^2\)The energy efficiency can equivalently be defined as the throughput divided by the power consumption of the link, \( EE = \frac{(1 - PER)L}{T_{\text{on}}P_{\text{total}}} \), where \( P_{\text{total}} \) is the total power consumption of the link, and \( T_{\text{on}} \) denotes the total time needed by the PHY layer to transmit the packet including the packet overheads but excluding the MAC layer related delays.
efficiency maximization problem is all possible 6-tuples \((P_t, N_t, N_r, N_{ss}, q, r)\), and each realization of the 6-tuple is called a mode.

The energy efficiency maximization problem is definitely more complicated than the throughput maximization problem since the search space is much larger for the former. The throughput maximization problem has much smaller search space for point to point links since the maximum \(P_t\) and maximum \(N_r\) are always optimum. When we use a realistic \(PER\) expression in the formulation (instead of idealistic capacity expressions that were employed in \([15,16]\)) and include many optimization variables as we did in this case, the problem no longer fits into simpler convex forms as we will see in the ensuing sections. In such a case, achieving the optimal solution in a low complexity manner becomes the main obstacle in the implementation of an energy efficiency maximization protocol for practical systems.

As can be seen from the problem formulation, the \(PER\) and the total energy consumption \(E_{total}\) are the two parameters that need to be computed in order to calculate the energy efficiency of the system. Next, we describe the models used for computing \(PER\) and \(E_{total}\).

### 3.3 Energy Consumption Model

We employ a comprehensive energy consumption model, which includes the energy consumed in the RF and baseband portions of both the transmitter and the receiver. For the RF and baseband energy consumption calculation, we employed the models presented in \([14]\) (Section IV), which were obtained from already published link adaptation
Figure 3.1: Transmitter and receiver block diagrams for energy consumption model.

works or are based on actual implementation data for specific blocks. The total energy consumption of the link is a function of $N_{ss}, N_t, N_r, P_t$, modulation and code rate; therefore the link adaptation algorithm tries to optimize these parameters simultaneously for maximizing energy efficiency.

In Fig. 3.1, we show the high level block diagram of the components that we included in the energy consumption calculations. The total energy consumption in the link consists of the RF energy consumption, and the baseband energy consumption.

$$E_{total} = E_{rf} + E_{base}$$

$E_{base}$ is the energy consumption of the baseband blocks and $E_{rf}$ is total energy consumption of the RF chains at the transmitter and receiver.

### 3.3.1 Energy Consumption of the RF chains

For the RF energy consumption modeling, a direct conversion RF transceiver is assumed whose power consumption is a function of $P_t, N_t, N_r$ and the bandwidth (BW) according
to the model presented in [12, 14]. \( E_{rf} \) consists of the energy consumption of various components and computed as

\[
E_{rf} = \left[ p_{pa} N_t P_t + \left( b_{rf}^t BW + c_{rf}^t \right) N_t + \left( b_{rf}^r BW + c_{rf}^r \right) N_r + c_{rf} \right] T_{on}. \tag{3.5}
\]

The first term, \( p_{pa} N_t P_t \), is the power consumption of the power amplifiers which is proportional to the peak-to-average-power-ratio (PAPR) \( \psi \) of the waveform, and inversely proportional to the PA efficiency \( \eta \), \( p_{pa} = \psi / \eta \). The second and third terms represent the sum of the energy consumptions of the transmitter side RF components, and the receiver side RF components respectively. The definitions for the constant coefficients used in (3.5), \( b_{rf}^t \), \( c_{rf}^t \), \( b_{rf}^r \), \( c_{rf}^r \), \( c_{rf} \), and their values are shown in Table 3.2 [14].

It will later prove useful to group the terms in (3.5) based on their dependencies on \( P_t \) and rewrite \( E_{rf} \) as

\[
E_{rf} = \frac{P_t N_t \psi}{\eta} T_{on} + E_{rf, others} \tag{3.6}
\]

The first term is the power amplifiers’ energy consumption, which is transmit power dependent, and \( E_{rf, others} \) is the energy consumption of all other components, which is transmit power independent.

### 3.3.2 Energy Consumption of the Baseband Blocks

The baseband energy consumptions, \( E_{base}^t \) and \( E_{base}^r \), are modeled as linear functions of \( N_t \), \( N_r \) and the bandwidth. The bandwidth dependency is due to the fact that the operating clock frequency scales linearly with the signal bandwidth.
Table 3.2: Coefficients for energy consumption calculations

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{pa}$</td>
<td>$\frac{\psi}{\eta}$</td>
<td>Coefficient related to PA power consumption. It is a function of the modulation since it changes the PAPR of the OFDM system, and the efficiency of the PA used in the system. Typical commercial amplifier’s have around 35% efficiency.</td>
</tr>
<tr>
<td>$b_{rf}^t$</td>
<td>$1.3 \times 10^{-8}$</td>
<td>Coefficient related to transmitter RF components, such as DAC, whose power consumptions are proportional to $BW$ and $N_t$.</td>
</tr>
<tr>
<td>$b_{rf}^r$</td>
<td>$1.8 \times 10^{-8}$</td>
<td>Coefficient related to receiver RF components, such as ADC, whose power consumptions are proportional to $BW$ and $N_r$.</td>
</tr>
<tr>
<td>$c_{rf}$</td>
<td>0.1</td>
<td>Power consumption of the other RF components, such as filters, whose consumptions are constant.</td>
</tr>
<tr>
<td>$c_{rf}^t$</td>
<td>0.042</td>
<td>Coefficient related to transmitter RF components, such as transmit mixers, whose power consumptions are proportional to $N_t$.</td>
</tr>
<tr>
<td>$c_{rf}^r$</td>
<td>0.042</td>
<td>Coefficient related to receiver RF components, such as receive mixers, whose power consumptions are proportional to $N_r$.</td>
</tr>
<tr>
<td>$b_{base}^t$</td>
<td>$4.09 \times 10^{-9}$</td>
<td>Coefficient related to transmitter baseband signal processing power consumption.</td>
</tr>
<tr>
<td>$b_{base}^r$</td>
<td>$1.62 \times 10^{-9}$</td>
<td>Coefficient related to receiver baseband signal processing power consumption excluding MIMO decoder’s power consumption.</td>
</tr>
</tbody>
</table>
The baseband energy consumption at the transmitter side is

\[ E^t_{\text{base}} = b^t_{\text{base}} N_t BWT_{\text{on}}. \]  \hfill (3.7)

\( E^r_{\text{base}} \) is the energy consumption of the receiver baseband excluding the MIMO decoder’s energy consumption.

\[ E^r_{\text{base}} = b^r_{\text{base}} N_r BWT_{\text{on}}. \]  \hfill (3.8)

MIMO decoders can be power hungry in MIMO-OFDM systems especially for the systems employing higher number of antennas and subcarriers. The MIMO decoder’s energy consumption, \( E_{\text{mimo}} \) is assumed to be a linear function of \( N_t, N_r, N_{ss}, q \) and \( r \) and independent of \( P_t \).

The total baseband energy consumption, \( E_{\text{base}} = E^t_{\text{base}} + E^r_{\text{base}} + E_{\text{mimo}} \), is therefore a function of these 5 parameters, however it is independent of the transmit power.

Finally, we group all the terms in \( E_{\text{total}} \) based on their transmit power dependencies and rewrite (3.4) as the sum of PA energy consumption and all other energy draws, \( E_o \), which are transmit power independent.

\[ E_{\text{total}} = \frac{P_t N_t \psi}{\eta} T_{\text{on}} + E_o \]  \hfill (3.9)

\[ E_o = E^t_{\text{base}} + E^r_{\text{base}} + E_{\text{mimo}} + E_{\text{rf, others}} \]  \hfill (3.10)

Complete and accurate modeling of the system’s total energy consumption is an important task in an energy-aware link adaptation. Incomplete energy consumption models might lead the link adaptation engine to pick trivial solutions. For instance, if we include only the energy consumption of the power amplifiers and ignore all other
energy draws as was done in [19], the link adaptation algorithm will try to minimize the total transmit power without regard to the amount of energy consumed at the receiver, and will keep many receive chains active. However, it was shown in [17] that when a complete energy model is used, the SISO modes outperform MIMO modes at short distances due to the fact that the energy consumption of RF and baseband circuitry becomes comparable to the PA energy consumption and turning off the antenna chains saves energy.

### 3.4 Packet Error Rate (PER) Prediction Model

Prediction of the instantaneous PER of the link is the most critical task in a link adaptation process since the optimum mode is chosen based on the predicted PER. There has been a significant amount of research done in order to estimate or predict the PER of MIMO-OFDM systems.

#### 3.4.1 PER Prediction Using Statistics from Upper Layers

When the aim is to maximize the throughput of a link, one approach to link adaptation is to do the adaptation at the MAC Layer based on observed PER statistics. The AutoRate Fallback (ARF) link adaptation protocol [6] and its variants [7,10] have been widely used in legacy WLANs for maximizing throughput. In the basic form of ARF, all the possible modes are sorted in terms of their rates and the radio automatically switches to a lower rate after two consecutive packet errors and switches to a higher rate after a number of successful packet transmissions (typical number is 10). Some
modifications are made on this basic algorithm in order to make it more robust. For example, in [11], a number of neighbor modes (neighborhood is defined in terms of their rates) are probed in a window. The radio collects the PER statistics for the modes that were probed in the previous window and the link adaptation algorithm switches to the higher performance mode based on these PER statistics.

These algorithms are very simple to implement, however they require probing of modes and hence they are inherently slow in collecting the statistics especially in highly dynamic channels. In non-static channels, by the time these algorithms collect the PER statistics for a mode, the channel changes significantly and hence the statistics are no longer valid for the current state of the channel. This strategy would work when we have only handful of modes to probe and the channel is changing very slowly. Given that we might end up with thousands of modes available for energy efficiency maximization, this type of a PER prediction method would not be a good choice.

3.4.2 PER Prediction Using Link Quality Metrics (LQM)

In order to make faster and more accurate PER predictions, PHY layer information needs to be employed. There has been significant amount of research focusing on the relationship between the PHY layer metrics and the PER performance in MIMO-OFDM systems.

In [11], authors simulated the WiMAX system with a specific channel model and determined the SNR thresholds for switching from one mode to another. However, SNR information alone is not sufficient for determining the error rate performance of a MIMO system since there is interaction between the antennas and the spatial
streams. In [26], the authors proposed an SNR based look-up table (LUT) method with an additional dimension which considers the determinant of the channel matrix for switching between the space time coding and spatial multiplexing modes. Similarly, a channel condition number assisted SNR thresholding method is used in [27] where the MAC Layer statistics are also employed to update the switching thresholds in response to the channel changes.

In some recent works, the post-processing SNR (PPSNR), which can be seen as the SNR at the output of the MIMO decoder, is leveraged to arrive at more accurate estimates of the link’s PER performance. PPSNR information is mapped to different link quality metrics (LQM) and the mapping from LQMs to the PER is determined via extensive simulations and curve fitting methods [9].

The basic idea behind the LQM mapping methods is to average the PPSNRs from different spatial streams and subcarriers using an averaging function and map the resulting scalar metric, i.e. LQM, to the PER through a mapping function.

For a given mode, and a set of PPSNRs, \( \Gamma = \{ \gamma_{k,n} : k = 1, \ldots, N_{ss}; \ n = 1, \ldots, N_{sc} \} \), the PER of the mode is estimated as

\[
PER(\Gamma) \approx PER^{AWGN}(\gamma_{eff}) \tag{3.11}
\]

where \( PER^{AWGN} \) is the PER of the system operating in an AWGN channel with only one spatial stream, and \( N_{sc} \) is the number of data subcarriers. These \( PER^{AWGN} \) curves are generated offline for all different modulation and code rate schemes for a single spatial stream case and stored in a look-up table. \( \gamma_{eff} \) is the effective PPSNR which is
obtained by averaging all the $\gamma_{k,n}$’s using different functions.

$$\gamma_{\text{eff}} = \nu(q, r) \frac{1}{N_{ss}N_{sc}} \sum_{k=1}^{N_{ss}} \sum_{n=1}^{N_{sc}} \omega(\gamma_{k,n}, q, r)$$  \hspace{1cm} (3.12)

There are many different averaging methods proposed in the literature for obtaining the effective PPSNR $\gamma_{\text{eff}}$ [9]. One example is the exponential effective SNR mapping (EESM) method which averages the individual PPSNRs as follows

$$\gamma_{\text{eff}} = -\vartheta(q, r) \ln \left( \frac{1}{N_{ss}N_{sc}} \sum_{k=1}^{N_{ss}} \sum_{n=1}^{N_{sc}} e^{-\gamma_{k,n}/\vartheta(q,r)} \right),$$  \hspace{1cm} (3.13)

where the parameter $\vartheta(q, r)$ is calibrated using simulations for the best fitting.

Other methods, such as the mean mutual information per bit (MMIB) mapping [28], are very similar to the EESM, the only difference is the averaging functions $\omega$ and $\nu$.

A major problem with these LQM methods is that they require extensive simulations to characterize the performance of the system. It may not be practical to simulate and calibrate the mappings for all channel models and all possible modes via simulations. A pure analytical modeling is therefore preferred to predict PER without extensive simulation and calibration. PPSNR based methods are more accurate in predicting the PER, however they might be overly complex to implement as will be seen in the ensuing sections.

### 3.4.3 PER Prediction via Analytical Modeling

Instead of extensive simulations and calibrations, which are required for LQM based methods, another approach for the PER prediction task is to consider the structure of different blocks, such as channel encoder/decoder, puncturer, modulator, and come up
with analytical models for the error rate performances. A pure analytical model would definitely be preferred since it would not require extensive simulation and calibration steps to characterize the performance for different scenarios. However, the model might end up being overly complex to implement.

There have been research efforts focusing on complete analytical modeling of the PER for coded MIMO-OFDM systems. Authors in [29], for instance, considered the structure of the convolutional encoder, Viterbi decoder, interleaver and puncturer blocks and modeled the PER analytically using the PPSNRs, however the resulting expressions are way too complicated to be implemented for a practical real-time system. Below, we employ a rather simpler analytical method for obtaining the PER of a convolutionally coded MIMO-OFDM system using the PPSNR as the metric.

The residual interference plus noise at the output of the MIMO decoder can be approximated as Gaussian [30], hence the uncoded bit error rate (BER) of the $k^{th}$ spatial stream at the $n^{th}$ subcarrier can be calculated using the well known AWGN channel BER expressions as

$$\Pr_{k,n} \approx \alpha Q\left(\sqrt{\beta \gamma_{k,n}}\right)$$  \hspace{1cm} (3.14)

where $\gamma_{k,n}$ is the PPSNR of the $k^{th}$ spatial stream at the $n^{th}$ subcarrier, $\alpha$ and $\beta$ are the modulation dependent parameters and can be found in [31].

The overall uncoded BER of the MIMO-OFDM system can be found by averaging the BERs of individual streams and subcarriers as

$$\bar{\Pr}_{b-u} = \frac{1}{N_{ss} N_{sc}} \sum_{k=1}^{N_{ss}} \sum_{n=1}^{N_{sc}} \Pr_{k,n}$$  \hspace{1cm} (3.15)
Table 3.3: Coefficients for the log-linear approximation

<table>
<thead>
<tr>
<th>Code Rate</th>
<th>1/2</th>
<th>2/3</th>
<th>3/4</th>
<th>5/6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>9</td>
<td>6.4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$v$</td>
<td>15</td>
<td>12</td>
<td>10</td>
<td>8.5</td>
</tr>
</tbody>
</table>

Almost all commercial high-speed wireless communication systems use some sort of channel coding. For convolutional coded systems, the coded BER of the system at the output of the Viterbi decoder is modeled as a linear function of the uncoded BER in log domain \[32\]

$$
\ln \left( P_{b-c} \right) \approx u \ln \left( \bar{P}_{b-u} \right) + v
$$

where $u$ and $v$ are the coefficients of the log-linear approximation and listed in Table 3.3 for different code rates. Different mappings can be employed for systems that use coding strategies different than convolutional coding. For mathematical correctness, using the above log-linear relationship we express $P_{b-c}$ as

$$
P_{b-c} = \begin{cases} 
(\bar{P}_{b-u})^u e^v & \text{if } (\bar{P}_{b-u})^u e^v \leq 1 \\
1 & \text{if } (\bar{P}_{b-u})^u e^v \geq 1
\end{cases}
$$

Finally, PER of the system can be found by the relationship

$$
PER = 1 - (1 - P_{b-c})^L,
$$

and the packet success rate (PSR) is simply defined as

$$
PSR = 1 - PER.
$$
It is important to note here that the above PER prediction model is not exact due to the underlying assumptions about the coding gain and the assumption that the coded bit errors occur independently. However, we prefer this mathematical model over previously proposed LQM based PER mapping methods since the LQM methods require extensive simulations and curve fitting for different channel models and different modes even though there exist methods to deal with different packet lengths [9]. In Section 6.3.3, we compare our method and the LQM based methods experimentally and verify that our model performs similar to, or better than, the well-known and simulation intensive LQM based mapping methods [9] under realistic experimental conditions.

3.5 Link Adaptation Protocol Using Brute Force Search

The wireless channel changes rapidly due to the mobility of the transmitter, the receiver and the surrounding objects. As a result, the channel exhibits small scale fading (fast fading) in addition to the path loss and shadowing effects (large scale effects or slow fading). These effects can easily be observed in Fig. 3.2(b), where we plotted the received SNR at one of the receiver antennas vs time for an experiment that we conducted in an indoor environment.

The receiver was put on a cart and pushed away from the transmitter with pedestrian speeds along the path shown in Fig. 3.2(a). We observe the fast fading effects due to the mobility in the channel, the shadowing effects when we turn around the corners in the hallway, and the path loss effects as we move away from the transmitter and then as we approach to it. As the channel conditions change rapidly, our purpose is to
(a) Floorplan and the route of the experiment.

(b) Received SNR at the first receiver antenna vs time.

Figure 3.2: Experimental SNR results. The transmitter was in the WISR laboratory and the receiver was pushed on a cart along the indicated path in the hallways.
adapt to these changes fast enough and do link adaptation based on the instantaneous conditions.

The link adaptation work in [14] is aimed at finding the optimal mode based on the average error rate performance and as such, disregards the fast fading in the channel. However, significant performance improvements can be achieved if the link adaptation algorithms are designed to take into account the instantaneous (or short term) channel conditions, rather than making a decision based on the long term average behavior of the channel. This approach is called fast link adaptation, which aims to provide improved performance by tracking and responding to the rapid variations in the channel [33].

Our PPSNR based fast link adaptation protocol is based on sounding the full MIMO channel periodically and determining the optimal mode to transmit. A mode $m$ is a 6-tuple $(P_t, N_{ss}, N_t, N_r, q, r)$ containing the parameters to be optimized, and the optimal mode is defined as the one which has the maximum energy efficiency or throughput depending on the desired objective. The link adaptation protocol for maximizing the energy efficiency is depicted in Fig. 3.3(a) and summarized below:

(i) Send a channel sounding packet to estimate the full channel matrix, $\tilde{H}$, for each subcarrier. The sounding packet is sent using all transmit antennas with the highest possible transmit power and with identity precoding matrix, and all the antennas are enabled for reception at the receiver side. This enables us to estimate all possible channels between each tx-rx antenna pair. It should be noted that we only need the preamble portion of this packet to get the channel estimates. Depending on the design, the sounding packet can also be a specially
designed packet filled with channel estimation symbols to get a less noisy channel estimate.

(ii) Based on the estimated channel matrices, the PERs and the resulting throughputs for all available modes (all possible 6-tuples) are calculated using the method presented in Section 3.4.3.

(iii) The modes that cannot satisfy the QoS constraints, $TH_{\text{target}}$ or $PER_{\text{max}}$, are removed from the search space.

(iv) Energy efficiencies for the remaining modes are calculated based on the predicted PER values. We enumerate all possible 6-tuples such that the mode index $m$ distinctly maps to a realization of 6-tuple. Among the modes that satisfy the QoS constraints, we choose the mode with index $m^*$ that has the maximum predicted energy efficiency.

$$m^* = \arg \max_{m \in M} (EE_m) \quad (3.20)$$

We denoted the search space as $M$, where the cardinality of $M$ is number of available modes.

(v) If none of the available modes satisfy the QoS constraints, the algorithm chooses the mode that has the highest predicted throughput. A second implementation choice in such a case could be to let the application decide whether to relax the QoS constraints or stop the transmission. If all the predicted PERs are very high ($\approx 1$), which means that the channel is too bad, the algorithm forces both the transmitter and the receiver to enter sleep mode to save energy.
(vi) The receiver feeds back only the number \( m^* \) to the transmitter to reduce the amount of information to be fed back. The transmitter then uses the mode chosen by the link adaptation algorithm for the remainder of the session. The sounding period is defined as the duration between two sounding packets (See Fig. 3.3(b)).

(vii) The whole process is repeated when the next sounding packet is transmitted at the beginning of a new session.

3.5.1 Practicality and Complexity Issues

The link adaptation problem in (3.2) can be solved via performing brute force search over all possible realizations of the six parameters as we described above. We can enumerate all possible modes such that the mode index \( m \) distinctly maps to a realization of 6-tuple, i.e. the optimization variables \((P_t, N_{ss}, N_t, N_r, q, r)\). We can then perform brute force search by calculating \( EE_m \forall m \in M \) where the cardinality of \( M \) is number of available modes, and picking the mode \( m^* \) that has the maximum \( EE \).

Brute force search over the 6-tuple \((P_t, N_{ss}, N_t, N_r, q, r)\) can easily be prohibitively complex in a typical mode-rich radio. For instance, with the parameter choices considered in this work (shown in Table 3.1, similar to an 802.11n system) we end up with 240 choices for the 5-tuple \((N_{ss}, N_t, N_r, q, r)\) and a typical radio can have tens of transmit power levels (we assumed 32 or 64 levels in simulations) resulting in thousands of modes to choose from. Calculating \( PER \) for thousands of modes would have very high complexity especially due to the matrix inversions required for PPSNR calculations.
Figure 3.3: Energy-aware fast link adaptation protocol. The receiver estimates the channel using the sounding packet, then computes PPSNRs, finds the optimal mode $m^*$ and sends it to the transmitter using the ACK packet. The transmitter uses the optimal mode until the next sounding packet is transmitted.
In addition to the complexity issues, the proposed algorithm assumed that the system knows the Doppler speeds so that it sounds the channel frequently enough for the optimal performance. A practical algorithm needed for determining how often we should be sounding the channel.

We address the complexity and practicality related problems via closed form solutions and iterative algorithms in the following chapter and propose a practical low-complexity link adaptation algorithm.
We show in the simulations chapter that the proposed link adaptation protocol has great potential, however, the protocol employing the brute force search approach turns out to be of very high complexity. As we pointed out in the previous chapter, a mode-rich radio can easily have thousands of choices for the six transmission parameters that we try to optimize. A \( 4 \times 4 \) 802.11n like system, for instance, will have 240 choices for the 5-tuple \((N_{ss}, N_t, N_r, q, r)\), and the transmitter power alone can have tens to hundreds of choices. Computing the PPSNRs and then the PERs for this many number of modes is a challenging task due to the equations that need to be implemented.

The most complicated task in terms of the computational complexity is the PPSNR calculation since the form in (2.3) or (2.5) require matrix inversions for every 4-tuple \((P_t, N_{ss}, N_t, N_r)\) \(^1\). The transmit power has the major impact on the complexity as it generally has the highest number of choices. For every transmit power, we need to

\(^1\)PPSNR is independent of modulation and code rate, but it is a function of the antenna configuration even for the same \(N_{ss}\) since the effective channel changes.
recompute the PPSNR, then PER, and then EE. To address the complexity problem, in this chapter we derive a novel matrix-inversion-free PPSNR formulation by expressing it in terms of the singular values. With this new PPSNR formulation, we remove the necessity of repeating the matrix inversions needed for (2.5) for every transmit power level, hence the complexity does not scale with number of transmit power levels anymore.

Computing the PPSNRs is the first part of the PER prediction process. In the second part of the PER prediction task, we use the computed PPSNRs to estimate the uncoded and coded BERs, then we compute the PER. These computations have to be repeated for every 6-tuple \((P_t, N_{ss}, N_t, N_r, q, r)\), and hence bringing significant complexity. In order to further reduce the complexity, we investigate the relationship between the energy efficiency and transmit power and we try to find closed form or iterative solutions for the optimum transmit power level when we fix the other 5 parameters. We prove certain characteristics of the EE as a function of the transmit power, and derive upper and lower bounds on the optimal transmit power for each of the 5-tuples \((N_{ss}, N_t, N_r, q, r)\). After limiting our search into a much smaller region, we use an iterative algorithm to find the optimal transmit power. We therefore compute the optimal transmit power for each realization of the 5-tuple and take the transmit power out of the search space. Simulation results suggest that we find the near-optimal transmit power level in about 2.5 iterations instead of trying all \(N_p\) levels (32, 64 power levels are assumed in simulations). This reduces the search space by orders of magnitude.

Motivated by our novel singular value based PPSNR formulation, we propose to
prune the search space by using the singular value ratios. The basic idea behind this pruning is to exclude the spatial multiplexing modes ($N_{ss} > 1$) that will observe low singular value ratios in the channel, since these types of channels are not suitable for transmitting multiple spatial streams. This way we can significantly reduce the spatial stream dimension of the search space.

In addition to the computational complexity problems, we assumed in the previous chapter that the protocol sounds the channel several times within the coherence time in order to be able to adapt to the fast changes. This would require the knowledge of how fast the channel is changing in a mobile environment.

In summary, in this chapter we develop a practical, low-complexity link adaptation protocol by

- reformulating the PPSNR into a matrix-inversion-free form using the singular value decomposition,
- investigating the relationship between EE and $P_t$ and finding the optimal $P_t$ via an iterative algorithm (this takes the $P_t$ dimension out of the search space),
- pruning the search space using singular value ratios (this prunes the $N_{ss}$ dimension significantly),
- developing an adaptive channel sounding strategy.
4.1 SVD Based PPSNR Formulation

Direct calculation of the PPSNR using (2.3) or (2.5) is of very high complexity because of the necessity of matrix inversions. We need to redo the matrix inversions whenever we want to find $EE$ for a different transmit power. Therefore, the number of transmit power levels ($N_p$) has a multiplicative impact on the complexity. In this section, we reformulate the PPSNR and arrive at a solution which does not require any matrix inversions and hence reduces the complexity of the PPSNR calculations by orders of magnitude.

For convenience, we begin by writing the original PPSNR formula in (2.3),

$$\gamma_k = \frac{P_t |(WH)_{k,k}|^2}{P_t \sum_{l \neq k} |(WH)_{k,l}|^2 + N_0 (WW^*)_{k,k}},$$

which can also be expressed as the simpler form that we showed in (2.5), for the MMSE decoder.

$$\gamma_k^{(MMSE)} = \frac{1}{\left(\left[\frac{P_t}{N_0} H^* H + I\right]^{-1}\right)_{k,k}} - 1$$

Both forms of the PPSNR, (2.3) or (2.5), require matrix inversions.

In order to arrive at a simpler solution, we first take the singular value decomposition (SVD) of the $N_r \times N_{ss}$ effective channel matrix ($N_r \geq N_{ss}$) as

$$H = U\Sigma V^*$$

(4.1)

where $\Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}$, and the matrix $S = diag(\sigma_1, \sigma_2, \ldots, \sigma_{N_{ss}})$ is a diagonal matrix with diagonal entries equal to the singular values of the effective channel matrix, $H$. 

42
Using the SVD of $\mathbf{H}$, we are interested in computing the denominator of the first term in (2.5). We denote the matrix of interest as $\mathbf{K}$.

$$
\mathbf{K} = \left[ \frac{P_t}{N_0} \mathbf{H}^* \mathbf{H} + \mathbf{I} \right] 
$$

(4.2)

$$
= \frac{P_t}{N_0} \mathbf{V} \Sigma \mathbf{U}^* \mathbf{U} \Sigma \mathbf{V}^* + \mathbf{I} 
$$

(4.3)

$$
= \frac{P_t}{N_0} \mathbf{V} \mathbf{S} \mathbf{S}^* \mathbf{V}^* + \mathbf{I} 
$$

(4.4)

$$
= \mathbf{V} \left( \frac{P_t}{N_0} \mathbf{S}^* \mathbf{S} + \mathbf{I} \right) \mathbf{V}^* 
$$

(4.5)

Inverse of the matrix $\mathbf{K}$ now becomes a trivial operation since $\mathbf{V}$ is a unitary matrix, $\mathbf{V}^{-1} = \mathbf{V}^*$, and $\left( \frac{P_t}{N_0} \mathbf{S}^* \mathbf{S} + \mathbf{I} \right)$ is a diagonal matrix.

$$
\mathbf{K}^{-1} = \mathbf{V} \left( \frac{P_t}{N_0} \mathbf{S}^* \mathbf{S} + \mathbf{I} \right)^{-1} \mathbf{V}^* 
$$

(4.6)

The $(k,k)^{th}$ entry of $\mathbf{K}^{-1}$ can now be expressed in terms of the singular values as

$$
(K^{-1})_{k,k} = \sum_{j=1}^{N_{ss}} \frac{1}{\frac{P_t}{N_0} \sigma_j^2 + 1} |V_{k,j}|^2 
$$

(4.7)

where $\sigma_j$ is the $j^{th}$ singular value of $\mathbf{H}$ and $V_{k,j}$ is the $(k,j)^{th}$ entry of the $\mathbf{V}$ matrix.

Finally, we can formulate the PPSNR as

$$
\gamma_k = \left( \sum_{j=1}^{N_{ss}} \frac{1}{\frac{P_t}{N_0} \sigma_j^2 + 1} |V_{k,j}|^2 \right)^{-1} - 1. 
$$

(4.8)

Although this PPSNR formula requires SVD computation instead of matrix inversion, we don’t need to compute it for every transmit power level. Once we calculate it for a reference transmit power level, we can then simply scale the singular values whenever we need to calculate it for other transmit power levels. Whereas in the former PPSNR formulation (2.5), we had to redo the matrix inversions for each transmit power level.
Table 4.1: Computational complexities of PPSNR calculation methods

<table>
<thead>
<tr>
<th>$\times N_{sc}$</th>
<th>Original Method (B.2), (2.3)</th>
<th>New Method (4.8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex Mult.</td>
<td>$1352N_p$</td>
<td>$312N_p$</td>
</tr>
<tr>
<td>Complex Div.</td>
<td>$30N_p$</td>
<td>$154N_p$</td>
</tr>
<tr>
<td>Matrix Inv.</td>
<td>$1x1 \rightarrow 16N_p$, $2x2 \rightarrow 9N_p$, $3x3 \rightarrow 4N_p$, $4x4 \rightarrow N_p$</td>
<td>...</td>
</tr>
<tr>
<td>SVD</td>
<td>...</td>
<td>$1x1 \rightarrow 16$, $2x2 \rightarrow 9$, $3x3 \rightarrow 4$, $4x4 \rightarrow 1$</td>
</tr>
</tbody>
</table>

power level. It is also worthwhile to note that both SVD and matrix inversion have similar computational complexities.

Computational complexities of our new PPSNR formulation in (4.8) and the original PPSNR formulation in (2.3) are presented in Table 6.3 for a $4 \times 4$ MIMO system\(^2\). We achieve 66% reduction in the number of multiplications and divisions. More importantly, we eliminate the necessity of $N_{sc}N_p$ many $4 \times 4$ matrix inversions, plus $4N_{sc}N_p$ many $3 \times 3$ matrix inversions, and $9N_{sc}N_p$ many $2 \times 2$ matrix inversions. Instead, we need only $N_{sc}$ many $4 \times 4$, $4N_{sc}$ many $3 \times 3$, and $9N_{sc}$ many $2 \times 2$ SVDs no matter how many transmit power levels are used for link adaptation. We present the complexity results about these algorithms in Chapter 6.

\(^2\)The implementation results on the complexity reduction of algorithms and total savings are presented in Chapter 6. For the implementation of SVD, we used the Golub-Reinsch algorithm [34] which has $O(mn^2)$ time complexity for an mxn matrix.
4.2 Energy Efficiency and Transmit Power Relationship

We proposed a new method of computing the PPSNRs using the SVD, and the complexity of the PPSNR computation process is greatly reduced since it does not scale with the number of transmit power levels. However, the second part of the PER prediction process, which is obtaining the PER using computed PPSNRs, still needs to be recomputed for every transmit power level if we employ a brute force search method. Hence, the brute force search over the 6-tuple \((P_t, N_{ss}, N_t, N_r, q, r)\) becomes prohibitively complex for a typical mode-rich radio since the number of modes can be in the order of thousands. The 5-tuple \((N_{ss}, N_t, N_r, q, r)\) usually have limited number of choices in practical systems, however the transmit power is selected from a larger set and hence expands the search space. Therefore, we aim to take the \(P_t\) dimension out of the search space by finding the optimal transmit power \(P^*_t\) for each of the 5-tuples \((N_{ss}, N_t, N_r, q, r)\) in a low-complexity manner.

Let’s assume that we fix all the other parameters, the 5-tuple, and investigate the relationship between \(EE\) and \(P_t\) for a specific realization of the 5-tuple. We write the energy efficiency as a function of the single parameter \(P_t\) as

\[
EE(P_t) = 1 - \frac{\text{PER}(P_t)L}{E_{\text{total}}(P_t)}. 
\]  

(4.9)

Using the model we presented in Chapter 3, we express the total energy consumption, \(E_{\text{total}}(P_t)\), as a function of the transmit power as

\[
E_{\text{total}}(P_t) = \frac{\psi}{\eta} N_t P_t T_{\text{on}} + E_o, 
\]  

(4.10)

where the first term is the energy consumption of the power amplifiers which is pro-
portional to the transmit power, and the second term $E_o$ is the energy consumption of all other components including some of the RF and baseband sections which are independent of the transmit power.

We also replace $(1 - \text{PER}(P_t))$ with $PSR(P_t)$ and express $EE$ as

$$EE(P_t) = \frac{PSR(P_t) L}{\frac{\nu}{\eta} N_t P_t T_{on} + E_o}.$$  \hspace{1cm} (4.11)

Authors in [13, 15, 16] studied energy efficiency functions in the form of capacity divided by the power consumption. Analyzing the optimality of the transmit power in those cases is much simpler since the channel capacities are nice concave functions of the transmit power. As we stated in the introduction, these idealistic capacity based expressions are not practical since the protocol could pick unrealistically low transmit powers which might be way below the sensitivity level of a practical receiver. Another problem with these methods is that they can not adapt modulation and code rate since the capacity is independent of these two parameters.

However, in our case, $EE(P_t)$ or $PSR(P_t)$ are neither convex nor concave functions of $P_t$. In fact, $PSR(P_t)$ is a complicated function including summations of many $Q$-functions. For $(\bar{P}_{b-u})^u e^v \leq 1$, it becomes

$$PSR(P_t) = \left(1 - \left(\frac{1}{N_{ss} N_{sc}} \sum_{k=1}^{N_{ss}} \sum_{n=1}^{N_{sc}} \alpha Q\left(\sqrt{\beta \gamma_{k,n}(P_t)}\right) e^v\right)\right)^L$$ \hspace{1cm} (4.12)

and for $(\bar{P}_{b-u})^u e^v > 1$, it becomes 0 according to the model we employed.

We observe that the $PSR(P_t)$ is neither convex nor concave function, however we observe the following properties. It is an increasing function in $P_t \in [\delta, \infty]$ and its range is $[0, 1]$. It is an S-shaped (sigmoidal) function which starts out convex and smoothly
transitions to concave after an inflection point as shown in Fig. 4.1. We show the
PSR($P_t$) function for different realizations of the other 5 parameters in Fig. 4.1. We
fixed the 5-tuple in each case and plotted $PSR(P_t)$ only as a function of the transmit
power.

These properties are not unique only to the PER method that we presented. Far
from it, we can safely assume that any form of a $PSR(P_t)$ function will have these
properties due to the physical nature of the practical wireless communications. For very
low transmit power levels, a practical system observes zero or very low packet success
rates due to either packet detection failures or high bit error rates at the decoding stage
even if the packet is detected correctly. When the transmit power reaches a value that
satisfies the receiver sensitivity level, then these systems observe a very sharp transition
in the packet success rates from 0 to 1 for a given channel realization.

We state the following proposition about the energy efficiency and the transmit
power relationship given that the $PSR(P_t)$ function has the properties stated above.
Proof is given in Appendix A.

**Proposition:** For the S-shaped $PSR(P_t)$ function; $EE(P_t)$ is a single-peaked (strictly
increasing before the peak and strictly decreasing after the peak) quasiconcave function
in $P_t$, and there exists a unique optimal solution $P_t^* = \min(P_t^{\text{max}}, \max(P_t^{\text{min}}, \tilde{P}_t))$, where
$P_t^{\text{max}}$ and $P_t^{\text{min}}$ are the maximum and minimum constraints on the transmit power forced
by either the power amplifiers’ limitations or the QoS constraints, and $\tilde{P}_t$ is the solution
to (4.13).

$$\left. \frac{\partial EE(P_t)}{\partial P_t} \right|_{P_t=\tilde{P}_t} = 0 \Rightarrow \quad \tilde{P}_t = \frac{PSR(\tilde{P}_t)}{PSR'(\tilde{P}_t)} - \frac{E_o}{\frac{1}{\eta}N_tT_{\text{on}}}$$ (4.13)
Figure 4.1: Packet success rate vs transmit power for four different 5-tuples.

Figure 4.2: Energy efficiency vs transmit power for four different 5-tuples. Shaded area defines the feasible region for transmit power.
Proof: See Appendix A.

Fig. 4.1 shows the S-shaped $PSR(P_t)$ function and Fig. 4.2 shows the $EE(P_t)$ and optimal points for different realizations of the 5-tuple. The sharp rise of the energy efficiency functions before the peak is due to the sharp transition of the $PSR$ functions from 0 to 1 in a short $P_t$ range. The peak occurs at low $PER$ (usually around 0.1%–2%), then the $EE$ function starts decreasing with $\frac{L}{\frac{1}{2}N_tP_tF_{on}+E_o}$. This is because the $PSR$ function becomes 1 above certain $P_t$ and increasing the $P_t$ further decreases the energy efficiency. We illustrated a tight power constraint case in Fig. 4.2, hence $P_t^* = P_t^{max}$ for the $3 \times 4$ 64QAM mode since the peak occurs outside the feasible region, whereas $P_t^{min}$ becomes optimal for the QPSK mode. For the $1 \times 4$ 64QAM function, the peak occurs in the feasible region, hence the optimal $P_t^*$ is the one achieving the peak.

We proved the existence and uniqueness of the optimal $P_t$ when we fix the other five parameters. However, we have to solve (4.13) in order to find the optimal transmit power, $P_t^*$. As can be seen in (4.12) the $PSR(P_t)$ function is itself a quite complicated function of $P_t$. Moreover, (4.13) requires us to express the derivative, $PSR'(P_t)$, which does not result in a simple (in terms of low complexity) closed-form solution for $P_t^*$. Hence, it is not practical to directly compute the expression in (4.13). The results of the proposition, however, are still useful since it helps us to develop the low-complexity solution that we present in the following section.
4.2.1 Low-complexity Iterative Solution for Finding Near-Optimal Transmit Power

Since we have proven that \( EE(P_t) \) is a single-peaked function, which strictly increases before the peak and strictly decreases after the peak, instead of performing brute force search over the entire \( P_t \) range, we can limit our search around the peak if we can find upper and lower bounds on where the peak occurs.

Since we consider a MIMO-OFDM system, multiple subcarriers and possibly multiple spatial streams are carrying data, and in the uncoded BER formulation in (3.15) we were averaging the BER contributions of individual subcarriers and spatial streams. If we consider the individual spatial stream and subcarrier that has the lowest PPSNR, its BER will be higher than all the others. When we replace each of the terms in the uncoded BER summations in (3.15) and carry on the calculations using the minimum PPSNR term, we will end up with a higher PER than the actual. Below, we show that we can find a closed form upper bound on the transmit power, which will satisfy the \( \text{PER}_{\text{max}} \) criteria.

The average uncoded BER over all spatial streams and subcarriers, \( \bar{P}_{b-u} \), will be upper bounded by the uncoded BER of the individual subcarrier and spatial stream with the minimum PPSNR (\( \gamma_{\text{min}} = \min_{k,n}(\gamma_{k,n}) \), minimum over all subcarriers and spatial streams).

\[
\bar{P}_{b-u} \leq \alpha Q \left( \sqrt{\beta \gamma_{\text{min}}} \right) \quad (4.14)
\]

We can apply the Chernoff bound to the \( Q \) function for further simplification and
rewrite the upper bound as
\[ Q(x) \leq \frac{1}{2} e^{-\frac{-x^2}{2}} \] (4.15)
\[ \bar{P}_{b-u} \leq \frac{\alpha}{2} e^{-\frac{\beta}{2} \gamma_{\min}}. \] (4.16)

Plugging the above upper bound in the PER prediction equations (3.17)–(3.18), we derive an upper bound to the \( PER \) as
\[ PER \leq 1 - \left[ 1 - \frac{\alpha u}{2} e^{-\frac{\beta}{2} \gamma_{\min} + v} \right]^L \] (4.17)

Let’s assume that we are interested in finding the transmit power that meets the \( PER_{\text{max}} \) criteria. If we replace \( PER \) with the \( PER_{\text{max}} \) value in above inequality, and use the ZF PPSNR expression for the minimum PPSNR, \( \gamma_{\min} = \frac{P_t/N_0}{\|H^*H\|_{\text{max}}} \), we get an upper bound to the transmit power that satisfies the \( PER_{\text{max}} \) criteria.
\[ PER \leq 1 - \left[ 1 - \frac{\alpha u}{2} e^{-\frac{\beta}{2} \gamma_{\min} + v} \right]^L \] (4.18)
\[ P_t \leq N_0 \|H^*H\|_{\text{max}}^{-1} \frac{2}{\beta u} \left[ v - \ln \left( \frac{2u}{\alpha^u} \left( 1 - (1 - PER_{\text{max}})^\frac{1}{L} \right) \right) \right] \] (4.19)
where \( \|H^*H\|_{\text{max}}^{-1} \) is the maximum value of \( \|H^*H\|^{-1} \) over all spatial streams and subcarriers. We denote this upper bound as \( P_t^{\sup} \).
\[ P_t^{\sup} = N_0 \|H^*H\|_{\text{max}}^{-1} \frac{2}{\beta u} \left[ v - \ln \left( \frac{2u}{\alpha^u} \left( 1 - (1 - PER_{\text{max}})^\frac{1}{L} \right) \right) \right] \] (4.20)

Similarly, when we employ only the maximum PPSNR, \( \gamma_{\text{max}} \), in (3.15) and use the lower bound [35] on the \( Q \) function, the resulting transmit power will be a lower bound, \( P_t^{\inf} \).
\[ Q(x) \geq \frac{1}{4} e^{-\frac{\beta}{2} x^2} \] (4.21)
\[ P_t^{\ominus} = N_0 [\mathbf{H} \mathbf{H}^{-1} \left( \frac{\pi}{2 \beta u} \right) v - \ln \left( \frac{4^u}{\alpha^u} \left( 1 - (1 - PER_{\max})^{\frac{1}{\beta}} \right) \right) \]  

(4.22)

where \([\mathbf{H} \mathbf{H}^{-1}]_{\text{min}}^{-1}\) is the minimum value of \([\mathbf{H} \mathbf{H}^{-1}]^{-1}\) over all spatial streams and subcarriers.

Since we use \(\gamma_{\min}\) and \(\gamma_{\max}\) in derivations, for frequency selective channels, the resulting bounds \(P_t^{\oplus}\) and \(P_t^{\ominus}\) occur on the right and left hand side of the peak respectively\(^3\), as can be seen in Fig. 4.2. Hence, they restrict us to a much smaller region where we can try to find a near-optimal solution in various ways.

There are various ways to find the peak if we restrict our search around the peak. One way is to use Newton’s method [36], however it requires derivatives as in the case of (4.13). Instead, we employ the bisectioning approach as described in Algorithm 1.

The idea behind the algorithm is that we first restrict our transmit power search in a narrower region around the transmit power that satisfies \(PER_{\max}\) by using simple closed form solutions \(P_t^{\oplus}\) and \(P_t^{\ominus}\). We then move to the middle point \(\bar{P}_t = \frac{P_t^{\oplus} + P_t^{\ominus}}{2}\) and evaluate \(PER(\bar{P}_t)\). If \(PER(\bar{P}_t) \approx 0\), it means we are still at the right hand side of the peak and we can further reduce the transmit power. Thus, we declare \(\bar{P}_t\) as the new upper bound and continue iterations until we are sure that we satisfy \(PER_{\max}\) and we are not using unnecessarily high transmit power. We repeat the algorithm and find \(P_t^*\) for every 5-tuple that are remaining in the search space. We discuss the complexity reduction achieved via Algorithm 1 in Chapter 6.

We note here that the exact peak can also be found with the cost of additional

\(^3\)Note that we used \(s_k^{(ZF)}\) to be able to derive (4.20) and (4.22). For MMSE, \(P_t^{\ominus}\) is still valid since \(P_{b-u}^{\text{MMSE}} \leq P_{b-u}^{ZF}\), for the lower bound we can simply back-off slightly for the worst case which is the frequency-flat channel.
Algorithm 1 Transmit Power Search Algorithm

Calculate $P_t^{\uparrow}$ and $P_t^{\downarrow}$ in (4.20) and (4.22). Exclude the mode if $P_t^{\downarrow} < P_t^{\min}$ or $P_t^{\uparrow} > P_t^{\max}$.

while ¬(near-peak-found) do
  $\bar{P}_t = \frac{P_t^{\downarrow} + P_t^{\uparrow}}{2}$
  Calculate $PER(\bar{P}_t)$
  if $PER(\bar{P}_t) \approx 0$ then
    $P_t^{\downarrow} = \bar{P}_t$
  else if $PER(\bar{P}_t) > PER_{\text{target}}$ then
    $P_t^{\uparrow} = \bar{P}_t$
  else
    near-peak-found = 1
  end if
end while

$P_t^* = \bar{P}_t$ \hspace{1cm} ¬: logical NOT operator

complexity as in the case of Newton’s method, however our simulations show that
Algorithm 1 achieves almost the same energy efficiency as the exhaustive search method.

4.3 SVD Based Pruning of the Search Space

A second aspect of expressing PPSNR in the form of (4.8) is that it reveals the relationship between the channel singular values and the system performance. It was shown in [37] that the ratio between the singular values of different spatial streams determines the error rate performance of a MIMO system. If the ratios of the smaller singular values to the largest singular value are low, it was experimentally observed in [38] that the channel is not suitable for supporting the spatial streams corresponding to the low
singular values.

In order to illustrate the effect of the singular value ratios (or singular value spread) on the PPSNRs more explicitly, we begin by rewriting (4.8) in an inner product form as

$$
\gamma_k = \left[ \frac{P_t}{N_0} \sigma_{Nss}^2 + 1 \quad \frac{P_t}{N_0} \sigma_1^2 + 1 \quad \cdots \quad 1 \right] \left[ \begin{array}{c}
|V_{k,1}|^2 \\
|V_{k,2}|^2 \\
\vdots \\
|V_{k,Nss}|^2 
\end{array} \right]^{-1} \left( \frac{P_t}{N_0} \sigma_{Nss}^2 + 1 \right) - 1 \quad (4.23)
$$

Without loss of generality, assume that the singular values are sorted in decreasing order, i.e. \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{Nss} \). Then,

$$
\frac{P_t}{N_0} \sigma_{Nss}^2 + 1 \quad \frac{P_t}{N_0} \sigma_1^2 + 1 \quad \cdots \quad 1 
\leq \frac{P_t}{N_0} \sigma_{Nss}^2 + 1 \quad \frac{P_t}{N_0} \sigma_2^2 + 1 \quad \cdots \quad 1 
\quad (4.24)
$$

In order to get more insight to the effects of singular value ratios on PPSNRs, let’s consider a simple case of \( N_{ss} = 2 \) and analyze two extreme cases:

- **Case I**: Singular values are approximately equal, \( \sigma_1 \approx \sigma_2 \).

- **Case II**: One of the singular values is much smaller than the other, \( \sigma_1 >> \sigma_2 \).

**Case I**: Assume the two singular values are approximately equal, \( \sigma_1 \approx \sigma_2 \). The PP-SNRs, \( \gamma_1 \) and \( \gamma_2 \) are

$$
\gamma_1 = \left[ \sim 1 \quad 1 \right] \left[ \begin{array}{c}
|V_{1,1}|^2 \\
|V_{1,2}|^2 
\end{array} \right]^{-1} \left( \frac{P_t}{N_0} \sigma_2^2 + 1 \right) - 1, \quad (4.25)
$$

$$
\gamma_2 = \left[ \sim 1 \quad 1 \right] \left[ \begin{array}{c}
|V_{2,1}|^2 \\
|V_{2,2}|^2 
\end{array} \right]^{-1} \left( \frac{P_t}{N_0} \sigma_2^2 + 1 \right) - 1. \quad (4.26)
$$
We immediately notice that \( \gamma_1 \approx \gamma_2 \) for any choice of the unitary \( V \) matrix, since it has to satisfy the following

\[
|V_{1,1}|^2 + |V_{1,2}|^2 = |V_{1,1}|^2 + |V_{1,2}|^2 = 1, \quad (4.27)
\]

\[
V_{1,1}^* V_{2,1} + V_{1,2}^* V_{2,2} = 0 \quad (4.28)
\]

**Case II:** Now, let's assume that the singular value spread is high, i.e. \( \sigma_1 \gg \sigma_2 \). Then,

\[
\gamma_1 \approx \frac{1}{\sqrt{2}} e^{j\theta} \Gamma \quad (4.29)
\]

\[
\gamma_2 \approx \frac{1}{\sqrt{2}} e^{j\theta} \Gamma \quad (4.30)
\]

In this case, we have \( \gamma_1 \approx \gamma_2 \) only if \( |V_{1,2}|^2 \approx |V_{2,2}|^2 \) which implies \( |V_{1,1}|^2 \approx |V_{2,1}|^2 \) in addition. Considering also the requirement in (4.28), we observe that only the matrices of the form \( V \approx \pm \frac{1}{\sqrt{2}} e^{j\theta} \Gamma \) result in \( \gamma_1 \approx \gamma_2 \), where \( \Gamma \) is any \( 2 \times 2 \) Hadamard matrix, i.e. \( \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \). It is very unlikely to observe \( V \) matrices of this form.

As the singular value ratio of the small singular value to the large one decreases, it is very likely to observe big differences between the PPSNRs. When the singular values are close to each other (singular value ratio is higher), it is very likely to observe similar PPSNRs on the spatial streams. In order to illustrate this, we simulated 10 million uncorrelated \( 2 \times 2 \) channel matrices, each of which have entries that are zero mean circularly symmetric complex Gaussians (ZMCSCG) with unit variance.

Fig. 4.3 shows the probability that the difference between \( \gamma_1 \) and \( \gamma_2 \) (in dB) is less than \( \Delta \) as a function of the singular value ratio. As can be seen, if the singular...
Figure 4.3: Prob(|γ₁ − γ₂| < ∆) vs σ²/σ₁ for various ∆ values. We generated 10 million instances of random 2×2 channel matrices each with entries that are ZMCSCG.

value ratio is low, it is very likely to observe big differences between the PPSNRs of different streams. These channels with low singular value ratios are not very suitable for transmitting multiple spatial streams especially when the energy efficiency of the communication link is a concern. For such channels, the system would have to adjust the transmit power according to the stream with minimum PPSNR, since it dominates the error rate performance. This transmit power level might be unnecessarily high and reduces energy efficiency for other spatial streams which have significantly higher PPSNRs than the minimum one. As it will be more clear in the ensuing sections, the energy-efficient link adaptation protocol does not prefer higher number of spatial
streams for channels that have low singular value ratios.

Based on these observations, we propose to lower the complexity of the link adaptation process by excluding the modes, which can not be supported by the channel, from the search space due to their low singular value ratios. We exclude a mode from the search space if the ratio of its smallest singular value to the largest is less than a prescribed threshold $\tau$.

$$\frac{\sigma_{N_{ss}}}{\sigma_1} < \tau \implies \text{Exclude modes that use } N_{ss} \text{ spatial streams}$$ (4.31)

We will show in Chapter 5 that this simple SVD based pruning provides more than 35% reduction in the search space (and the complexity as a result) with less than 2% performance penalty for $\tau = 0.5$. Higher the threshold we pick, higher the reduction is, however it might have a worse impact on the performance. We explore the trade off between the performance and $\tau$ via simulations in Chapter 5 and find the optimal threshold value.

### 4.4 Low-Complexity Link Adaptation Protocol

The result of the complexity reduction, search space pruning and the transmit power search algorithms, that are presented in this chapter, is a low-complexity link adaptation protocol which is illustrated in Fig. 4.4.

We assume that the receiver estimates the full channel matrices for each subcarrier. The term *full* indicates that the receiver estimates the channel between all transmit-receive antenna pairs. In 802.11n like systems, this is done by communicating a special sounding packet which is transmitted using all available antennas and the maximum
available transmit power to ensure reception; and received by all available receive antennas to estimate the full channel matrices.

We then calculate the effective channels for each mode using precoding matrices, i.e. $H = \tilde{H}R$. We had denoted the search space spanned by all available combinations of the 6-tuple as $M$. We now denote $Z$ as the new search space containing all realizations of the 5-tuple after excluding the $P_t$ dimension. After taking the SVDs of channel matrices for a reference transmit power, we prune $Z$ by removing modes that have low singular value ratios. We then calculate the PPSNR, PER, and the $P_t^*$ using Algorithm 1 for all 5-tuples in $\tilde{Z}$. Finally, we choose the mode that maximizes the energy efficiency while satisfying the QoS constraints.

This whole process is repeated when the next sounding packet is transmitted at the beginning of a new session.

There are few details that we have not shown in Fig. 4.4 for presentation simplicity.
One of them is: if none of the modes in the search space are satisfying the QoS constraints and all of them are observing high PERs, which means the channel is too bad, we force the transmitter and the receiver to enter sleep mode for a predetermined period to save energy. Another one is: if none of the modes can satisfy the QoS constraints, however, some are observing reasonable PERs, we choose the mode that maximizes the throughput in order to continue the data transfer.

4.5 How to Determine How Often We Should Be Sounding?

The natural problem of determining the frequency of the sounding packets arises in our link adaptation protocol. This problem is important especially when the mobility in the environment changes over time. The channel needs to be sounded frequently enough so that it does not change significantly between two sounding packets. Doppler frequency in the channel should therefore be estimated in order to determine the sounding period. It could be estimated from the rate of change of the channel coefficients if we were able to account for any phase variation due to random sampling instances and packet detection timing uncertainties.

Instead of estimating the Doppler from channel coefficients, we measure the rate of change in PPSNR between sounding packets since it is the PPSNR that solely determines the performance of the link. Initially, we assume a value for the maximum Doppler \( (f_{d, \text{max}}) \) in the channel and set the starting sounding period to be \( \frac{1}{6f_{d, \text{max}}} \). Then as time progresses, the difference in PPSNR between two sounding packets is averaged.
\[ \Delta \gamma(i) = \frac{1}{N_{ss}N_{sc}} \sum_{k=1}^{N_{ss}} \sum_{n=1}^{N_{sc}} |\gamma_{k,n}(i) - \gamma_{k,n}(i-1)| \] (4.32)

If the averaged change in PPSNR, \( \Delta \gamma(i) \), is above a threshold, we decrease the sounding period, and if it is below a threshold, we increase the sounding period. This way, we can adapt to the mobility changes in the channel. This algorithm is depicted in Fig. 4.5. In the simulations chapter, we will present 3 set of simulations related to the channel sounding period determination:

- one with sufficiently small sounding period assuming prior knowledge of the maximum Doppler speeds in the channel,
- one with longer sounding period which observe performance losses due to insufficient sounding of the mobile channel,
• one with adaptive channel sounding period algorithm, which will adapt the duration of the sounding period based on the PPSNR changes.

We observe a performance loss when we employ a long fixed sounding period while the channel is rapidly varying. With the adaptive algorithm, however, we achieve much better performance without requiring any information about mobility in the environment.
CHAPTER 5

Simulations

We performed various end-to-end link level simulations in order to characterize the performance of the proposed link adaptation algorithms. In this chapter, we first describe the simulation environment that we developed, then we show the results of the realistic simulations.

The simulator has all the baseband transmitter and the receiver blocks of the MIMO-OFDM system that we have shown in Chapter 2. In addition to the baseband transmitter/receiver blocks, we have a realistic time-varying multipath channel model. The simulator also allows us to inject various RF impairments, such as IQ imbalance, phase noise, frequency offset, using their equivalent baseband models. We developed and used a bit-level simulator, meaning that we transmit and receive the bits without any abstraction about the BER, PER etc. Algorithms in the simulator are implemented in floating point format.
5.1 Simulation Setup and Assumptions

For simulation purposes, we assumed an IEEE 802.11n [1] like system from which we have taken the simulation parameters. The transmitter side consists of the baseband transmit blocks shown in Fig. 2.1 and also some other blocks related to preamble and packet generation that were not shown on the block diagram for simplicity purposes.

We assumed uniform modulation and uniform power allocation over all subcarriers and spatial streams. When we have more transmit antennas than the spatial streams, i.e. $N_t > N_{ss}$, the spatial expansion mode is evoked, and the spatial streams are mapped onto the transmit antennas in a round robin fashion. For instance, in the case of $N_t = 3$, $N_{ss} = 2$, the first spatial stream will be duplicated to the third transmit antenna with a proper cyclic delay [1]. This cyclic delay provides diversity in frequency domain, known as cyclic delay diversity, since it creates artificial multipaths and frequency selectivity in the channel.

As an example, the precoding matrix in the case of $N_t = 3$, $N_{ss} = 2$, with round robin processing becomes

$$R = D \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$  \hspace{1cm} (5.1)

where $D$ is the matrix applying cyclic delays to the transmitted signal, and it can be different for the data and the preamble portions of the packet. $D$ is an identity matrix if no cyclic delay is desired.

The receiver side has additional blocks, such as packet detection, timing and fre-
quency synchronization, and channel estimation, compared to the generic baseband receiver presented in Fig. 2.1. A detailed receiver block diagram is shown in Fig. 6.1 and details of the algorithms are explained in the experiments chapter. For simulation simplicity, we assumed ideal synchronization in most cases, unless otherwise stated. After finding the good starting sample for the FFT window, we take the FFT of the signal following the removal of the CP. For the MIMO decoding task, we employed the MMSE decoder to separate the individual streams from each other. The received symbols after the MIMO decoder are deinterleaved, deparsed and then passed through the depuncturer and the soft-decision Viterbi decoder.

5.1.1 Channel Model

We simulated a quasistatic multipath mobile channel, meaning that the channel remains constant during one packet, but changes from one packet to another due to the small scale effects (mobility - fast fading) and the large scale effects (path loss and shadowing - slow fading).

For a realistic channel simulation, we employed the simplified path loss and log-normal shadowing models in [31]. In addition to these large scale effects, the signal is affected by fast fading due to the mobility in the channel. For fast fading, we used the Jakes’ Doppler model in [39]. Each transmit-receive antenna pair is assumed to experience independent fading.

The power delay profiles (PDP) of the multipath channel are taken from 802.11n channel models\textsuperscript{1} [40]. We assumed each path observes an independent Doppler process,

\textsuperscript{1}We simplified the 802.11n channel models by using a smaller number of multipaths in order to
but the Doppler frequency is the same for all paths. The multipath profiles for each transmit-receive antenna pair are obtained from the same statistical PDP, but with a random drawing, hence they have different initial realizations. All the paths then change based on the Doppler model as time progresses.

In addition, we assumed that the mobility in the channel also changes over time, hence the Doppler frequency changes over time.

In Fig. 5.1, we plotted an example of the multipath profile for the channel type C, and how one of these paths is changing over time for different Doppler frequencies.

5.1.2 Simulation Parameters

Table 5.1 presents the system parameters that we used in our simulations. Since we assumed a 4×4 system, \( N_t, N_r \) and \( N_{ss} \) have choices from one to four. We have four different constellation and four different code rate options constituting (along with the \( N_{ss} \) choices) 32 modulation and coding schemes (MCS) defined by the 802.11n standard.

In every \( T_{\text{sounding}} \) seconds, a sounding packet is sent by the transmitter using all antennas and the highest possible \( P_t \). The receiver receives it using all of its antennas to estimate the full 4×4 channel matrices and \( N_0 \). We then calculate the optimal mode (\( m^* \)) using the proposed low-complexity algorithm and transmit using \( m^* \) until the next sounding period. We used both adaptive sounding and fixed sounding periods in simulations for comparison purposes.

We simulated a file transfer application, which requires all the lost packets to be

---

reduce the simulation time. Delay spreads are still the same.
(a) Multipath profile of an example channel.

(b) Magnitude of a path changing over time based on the Doppler rates.

Figure 5.1: Multipath mobile channel
Table 5.1: Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_t$</td>
<td>1, 2, 3, 4</td>
<td>Number of tx antennas</td>
</tr>
<tr>
<td>$N_r$</td>
<td>1, 2, 3, 4</td>
<td>Number of rx antennas</td>
</tr>
<tr>
<td>$N_{ss}$</td>
<td>1, 2, 3, 4</td>
<td>Number of spatial streams</td>
</tr>
<tr>
<td>$q$</td>
<td>1, 2, 4, 6</td>
<td>Modulation. $q = 1$: BPSK, $q = 2$: QPSK, $q = 4$: 16QAM, $q = 6$: 64QAM</td>
</tr>
<tr>
<td>$N_{sc}$</td>
<td>52</td>
<td>Number of data subcarriers</td>
</tr>
<tr>
<td>$BW$</td>
<td>20 MHz</td>
<td>Bandwidth</td>
</tr>
<tr>
<td>$P_t$</td>
<td>-3 – 20 dBm</td>
<td>Transmit power (32 or 64 equally spaced levels in dB)</td>
</tr>
<tr>
<td>$L$</td>
<td>1 KByte</td>
<td>Packet size (info bits)</td>
</tr>
<tr>
<td>$f_c$</td>
<td>2.4 GHz</td>
<td>Carrier Frequency</td>
</tr>
<tr>
<td>$f_d$</td>
<td>1–8 Hz</td>
<td>Doppler frequency (varies)</td>
</tr>
<tr>
<td>$\varsigma$</td>
<td>3.4</td>
<td>Path loss exponent (typical value for indoor channels)</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>2 dB</td>
<td>Log-normal shadowing variance</td>
</tr>
<tr>
<td>Channel</td>
<td>Type C [40]</td>
<td>802.11n channel type C, 30ns delay spread</td>
</tr>
<tr>
<td>$T_{silence}$</td>
<td>200 µsec</td>
<td>Inter-packet silence duration (for MAC tasks)</td>
</tr>
</tbody>
</table>

retransmitted until all of them are successfully received. A target time is defined for the transmission of the file. The link adaptation algorithm thus calculates the target throughput online based on the remaining file size and the remaining time. We simulated the transmission of files with different sizes and different target times.
5.2 Simulation Results

In this section, we present various simulation results corresponding to different scenarios and strategies. First, we show the energy efficiency performance of the proposed low-complexity protocol and discuss the mode selection trends for energy efficiency maximization. We also compare our low-complexity protocol to the brute force (or exhaustive search) method where we enumerate all possible choices for the adaptation parameters and we choose the mode that has the maximum predicted energy efficiency. The brute force search is guaranteed to achieve the optimal solution, hence providing an upper bound on the energy efficiency that any link adaptation protocol can achieve.

In the second part, we present the simulation results for different link adaptation approaches and compare our method against them. We show the simulation results for throughput maximization, and compare our proposed solution against the MAC layer based protocols taken from the literature.

5.2.1 Energy Efficiency Results and Mode Selection Trends

For the results that we present in this section, we simulated a realistic link scenario, where the receiver node moves away from the transmitter for the first 3 seconds, then it approaches the transmitter for 3 seconds, and then it moves away again for 3 seconds with a speed of 0.75 m/sec \((f_d = 6\, \text{Hz}, \text{pedestrian speed})\). See the SNR fluctuations in Fig. 5.2.

We plotted \(EE\) vs time, and the chosen mode vs time in Fig. 5.2 and Fig. 5.3, respectively. The proposed protocol changes its mode in an opportunistic manner to
track the fast fading changes in the channel. The optimal mode is therefore dependent on the instantaneous characteristics of the channel, therefore the energy efficiency fluctuates as the SNR or channel quality changes. However, it is still important to draw conclusions about general trends of the mode selection as a function of SNR changes (from low to high and vice versa).

The first thing to observe from Fig. 5.3 is that the protocol chooses the higher order constellations, 64QAM and 16QAM very often, 29% and 64.8% of the time respectively. Higher order constellations are preferred in terms of energy efficiency since the packets are transmitted in a shorter duration. The protocol switches to a lower constellation only when the channel quality can not support higher order constellations. We observe that as SNR decreases, the protocol starts picking 16QAM more often, then switches to QPSK (6.2% of the time) around the 3rd second when the SNR makes a dip. BPSK was never chosen by the algorithm for this scenario.

At high SNR regimes (shorter distance), \( E_{\text{base}} \) becomes comparable to \( E_{\text{rf}} \) due to the fact that we do not need high \( P_t \) to support 64QAM or 16QAM, and hence smaller \( N_{ss} \) and \( N_r \) are chosen to reduce the energy consumption of the baseband. As the SNR goes lower, the algorithm reacts with increasing \( P_t \) (boosts the received SNR) and \( N_r \) (achieves spatial diversity) first, to be able to maintain higher order constellations. Increasing \( N_t \) as SNR decreases is not preferred in the first place, since we employ CDD as a transmit diversity technique, which does not provide the strong diversity of other methods such as STBC or beamforming. The algorithm turns on additional tx antennas only when it needs to transmit additional spatial streams via lower constellations. It is costly in terms of the energy consumption to increase \( N_{ss} \) at short distances, and
Figure 5.2: Energy Efficiency vs time (Upper Plot), and received SNR for the 1st antenna vs time (measured via the sounding packet which uses the maximum $P_t$, the actual SNR during the data transmission is different since the link adaptation algorithm keeps adapting the transmit power).

the channel is not good enough to support multiple spatial streams at long distances. This is why we never observed three or four spatial streams in the experiment. It chose single spatial stream for 68% of the time.
In addition to our proposed protocol, we simulated different strategies for comparison purposes as shown in Table 5.2. We first compare our proposed protocol to the brute force search method and observe that it is only 2% less energy efficient (for $\tau = 0.5$) than the protocol employing the brute force search method. It achieves huge complexity reduction as we show in the next chapter.

Being able to adjust the transmit power is very critical in terms of energy efficiency as the power amplifiers consume a significant amount of energy. In order to save energy,
Table 5.2: Simulation results for energy efficiency maximization.

<table>
<thead>
<tr>
<th>Link Adaptation Method</th>
<th>Avg. EE (Mbits/J)</th>
<th>Total Time (sec)</th>
<th>Avg. PER</th>
<th>QoS Outage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brute force search</td>
<td>12.97</td>
<td>7.60</td>
<td>3.49%</td>
<td>0.28%</td>
</tr>
<tr>
<td>Proposed protocol with $\tau = 0.5$</td>
<td>12.70</td>
<td>7.74</td>
<td>3.53%</td>
<td>0.81%</td>
</tr>
<tr>
<td>Brute force with fixed $P_t = 20$ dBm</td>
<td>8.81</td>
<td>7.01</td>
<td>1.65%</td>
<td>0.30%</td>
</tr>
<tr>
<td>Brute force with upper bound $P_t = P_t^{\overline{1}}$</td>
<td>11.38</td>
<td>7.69</td>
<td>0.61%</td>
<td>0%</td>
</tr>
</tbody>
</table>

the EE maximization protocols pick just enough transmit power to deliver the packets. Once in a while the EE maximization protocol observes packet losses and a higher PER, however it obtains huge energy savings and still meet the QoS constraints ($PER < 5\%$) more than 99% of the time. We simulated a strategy where the link adaptation engine can not adjust the transmit power (only 20 dBm is available), but it does a brute force search over the other 5 parameters. This protocol achieves only 8.81 Mbits/J efficiency, which is 32% less than what the baseline system achieves.

We proposed the Algorithm 1 for finding the near optimal transmit power by performing a few iterations. Simulation results show that we find the near optimal level in 2.5 iterations on the average. Instead of using the Algorithm 1, another approach that has even smaller computational complexity would be to use the upper bound ($P_t = P_t^{\overline{1}}$) as the near optimal transmit power. We also simulated this approach as can be seen in Table 5.2. Although it has less complexity (complexity calculations are shown in the
experiments chapter), it achieves energy efficiency of 11.38 Mbits/J, which is 12% less than the baseline brute force method. Hence, it could also be a good candidate for the system designs that are limited by the complexity rather than the performance.

5.2.2 Comparison to Other Strategies and Throughput Maximization Results

We simulated different link adaptation strategies, such as brute force throughput maximization, a MAC layer based link adaptation that is widely used in WLAN radios, and some fixed modes (no link adaptation). This section presents the results and the discussions related to these approaches.

For the results that we present in this section, we simulated a mobile multipath channel with variable Doppler rates. The link scenario is similar to the one that we used in the previous section, however the mobility or the Doppler rate changes over time. The receiver node moves away from the transmitter and returns back at varying speeds. It moves away from the transmitter for the first 3 seconds with a speed of 0.75 m/sec (6 Hz Doppler), then it approaches the transmitter for 4 seconds with a speed of 0.50 m/sec (4 Hz doppler), then it moves away again for 2 seconds with a speed of 1 m/sec (8 Hz doppler), then returns back for 4 seconds with 0.50 m/sec speed and so on (See SNR changes in lower plot of Fig. 5.4).

The simulated approaches for this section are listed along with the results in Table 5.3. The MAC based protocol that we simulated is based on the Robust Rate Adaptation Algorithm (RRAA) [10]. This protocol measures the PER for several candidate modes (neighbor modes) within a short window of time, then switches to another mode
if the achievable throughput at that mode is higher than the throughput at the currently used mode. The packet error rates and the achievable throughputs are measured by transmitting several packets for each mode in the candidate set. First, all the modes are ordered in terms of their rates. We define six neighbor modes to the currently used mode as our candidate set. Three have lower rates than the current mode and the other three have higher rates, and all neighbor modes are probed along with the currently used mode. In each cycle, 60 packets are sent using the neighbor modes (10 for each neighbor) and 240 packets are sent with the currently used mode. For the next cycle, the MAC based link adaptation algorithm switches to the higher performance mode based on the PER statistics of the probed modes in the current cycle, and the neighborhood is also updated. These types of protocols are widely used in WLAN radios [6, 7, 10].

The second simulated protocol is the proposed protocol with the objective of maximizing the throughput. Fig. 5.4 shows the results comparing our protocol with the MAC based protocol in terms of their throughputs. The very first thing to observe is that the proposed protocol responds very quickly to the fast fading changes in the channel quality, since we sound the channel and pick the optimal mode to maximize the instantaneous throughput. On the other hand, a MAC based protocol is inherently slow since it needs to build PER statistics by trial and error. Once it finds a better performing mode to switch to, the channel has already changed. When the SNR decreases rapidly, the MAC based protocol will suffer from severe packet losses while it slowly tries to converge to a lower throughput mode. This fact is observed in Fig. 5.4 (middle plot) where we plot the PER of the protocols over time. When the channel
Table 5.3: Simulated protocols and results

<table>
<thead>
<tr>
<th>Protocol Name</th>
<th>Short Description</th>
<th>Energy</th>
<th>Avg TH</th>
<th>QoS Outage</th>
<th>Avg PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC based link adaptation - max TH</td>
<td>Protocol adapted from [10]. Uses observed PER statistics to maximize TH.</td>
<td>89.84 J</td>
<td>17.75 Mbps</td>
<td>18.84%</td>
<td>12.19%</td>
</tr>
<tr>
<td>Proposed PPSNR based protocol - max TH</td>
<td>Proposed PPSNR based link adaptation protocol with objective of maximizing TH.</td>
<td>51.19 J</td>
<td>23.8 Mbps</td>
<td>0.94%</td>
<td>0.49%</td>
</tr>
<tr>
<td>Proposed PPSNR based protocol - max EE</td>
<td>Proposed PPSNR based link adaptation protocol with objective of maximizing EE.</td>
<td>31.45 J</td>
<td>17.83 Mbps</td>
<td>8.01%</td>
<td>2.06%</td>
</tr>
<tr>
<td>4×4 16QAM, rate 1/2, Nss = 2 - Fixed Mode</td>
<td>A fixed mode, 2 spatial streams, no link adaptation. Pt = 20 dBm.</td>
<td>88.09 J</td>
<td>19.95 Mbps</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>3×4 64QAM, rate 2/3, Nss = 3 - Fixed Mode</td>
<td>A fixed mode, 3 spatial streams, no link adaptation. Pt = 20 dBm.</td>
<td>81.64 J</td>
<td>13.93 Mbps</td>
<td>58.77%</td>
<td>45.78%</td>
</tr>
<tr>
<td>1×4 16QAM, rate 3/4, Nss = 1 - Fixed Mode</td>
<td>A fixed mode, 1 spatial streams, no link adaptation. Pt = 20 dBm.</td>
<td>112.1 J</td>
<td>16.68 Mbps</td>
<td>4.32%</td>
<td>1.77%</td>
</tr>
</tbody>
</table>
quality gets better again, the MAC based protocol is once again sluggish in its response (see Fig. 5.4). The results show a 34% improvement in throughput over the MAC based protocol when we run our PPSNR based protocol in throughput maximization.
mode. We also included two fixed modes, which achieved reasonably high throughputs for comparison purposes. The gain over these well performing fixed modes are 20% and 71%, respectively.

The third simulated protocol is again the proposed PPSNR based protocol, but with the objective of maximizing the link’s energy efficiency. Energy efficiencies of the pro-
tocols are plotted over time in Fig. 5.5. We observe that the throughput maximization protocol finishes the transmission in a shorter time, however it is very energy inefficient, especially in high SNR regimes. This is because the throughput maximization protocols always use the maximum transmit power, whereas the energy-aware protocol reduces the transmit power to save energy as long as the target QoS constraints are satisfied. The proposed energy-aware link adaptation algorithm consumes 31.45 Joules to complete the transmission of a 40 MB file, which is significantly less compared to the consumption of the $TH$ maximization protocols. It is 4x more energy efficient than the $(1x4, 16QAM, \text{Rate } 3/4, N_{ss} = 1)$ fixed mode, 2.5x more efficient than $(4x4, 16QAM, \text{Rate } 1/2, N_{ss} = 2)$ and $(3x4, 64QAM, \text{Rate } 2/3, N_{ss} = 1)$ fixed modes (Table 5.3).

We have chosen to show the performance of reasonably well performing fixed modes for presentation simplicity (for instance the 4x4 16QAM mode did not observe any packet losses throughout the simulation). We observe, however, that the PPSNR based energy-aware protocol achieves orders of magnitude improvement in energy efficiency compared to poorly chosen fixed modes while satisfying the QoS constraints. It satisfies the QoS constraints ($PER < 10\%$) 92% of the time.

5.2.3 Adaptive Sounding Period Related Simulation Results

We performed 3 different set of simulations related to the channel sounding period:

- Constant, small enough sounding period, $T_{\text{sound}} = \frac{1}{48}$. This choice of sounding period ensures that we sound the channel 6 times during the smallest coherence time when $f_d = 8$ Hz, which is fast enough so that the channel does not change
Figure 5.6: Adaptive sounding period.

significantly between two sounding packets. We observed 2.09% PER during the entire simulation for this choice.

- Constant insufficiently long sounding period, $T_{\text{sound}} = \frac{1}{12}$. This choice leads to undersampling of the channel, hence we observed 13.96% PER which in return causes significant performance loss.
Adaptive channel sounding period algorithm. We show how $T_{\text{sound}}$ is adapted over time in Fig. 5.6 for the channel that has varying mobility. We observe that this simple algorithm reacts to the mobility changes in the channel without explicitly knowing the Doppler rate and observes 2.06% PER. It quickly switches to a lower sounding period when the mobility changes from 4 Hz to 8 Hz, and the value it settles is half of the value it settled for the 4 Hz case. We see noise like variations at the plateaus, which is due to temporal faster or slower variations in the channel.

5.2.4 Singular Value Ratio Threshold Related Simulation Results

The complexity reduction performance of the SVD based pruning method is clearly dependent on the SVR threshold, $\tau$. We simulated the proposed protocol for different values of the $\tau$ and we report the results in this section. The simulation assumptions and the scenario are the same as those used in section 5.2.1.

Fig. 5.7 shows the average energy efficiencies achieved throughout the simulation and the percentage of the modes excluded by the SVD based pruning algorithm versus different values of the threshold. We observe that the energy efficiency performance more or less stays constant for $\tau < 0.5$, however there is a big difference in complexity reduction between $\tau = 0.1$ and $\tau = 0.5$, as the percentage of modes that were excluded from the search space shows a linear behaviour as a function of $\tau$. We start observing a significant performance loss in energy efficiency for $\tau > 0.5$. Hence, $\tau = 0.5$ is a good candidate for implementation, as it does not cause any significant performance loss while we exclude about 40% of the modes from the search space without going through the PER calculations. Impacts of this on the complexity are discussed in the
Figure 5.7: Complexity reduction (left y-axis) and energy efficiency (right y-axis) performances for different $\tau$ values.

following chapter.
CHAPTER 6

Experiments

In order to validate the gains suggested by the simulation results and to test the algorithms in real life scenarios, we implemented the PPSNR based link adaptation protocol on a real time MIMO-OFDM testbed. In this chapter, we first present an overview of the testbeds used in experiments and the development of the link adaptation algorithms. We then present the results of experiments that were carried out in an indoor environment.

6.1 Testbed Overview

Experiments were carried out using a real time 4x4 MIMO-OFDM testbed [38, 41] developed by the UCLA Wireless Integrated Systems Research (WISR) laboratory researchers and Silvus Inc. engineers. The PHY layer baseband blocks of the transmitter are implemented on a Virtex-5 field-programmable gate array (FPGA) based on the IEEE 802.11n standard [1]. The transmitter uses up to 4 antennas in spatial multiplexing mode and follows the packet structure mandated by the standard.

A detailed receiver baseband block diagram is given in Fig. 6.1. The first task
of the receiver is to detect the presence of the packet. This is done by using the delay and correlate algorithm [42] which takes advantage of the periodicity of the short training symbols at the start of the preamble. Packet detection is declared when the autocorrelation value rises above a certain threshold.

Pilot aided frequency and timing synchronizations are performed using the training symbols embedded in the packet preamble [43]. Coarse frequency and timing offset estimations are done using the short training fields (STF). The estimations are then fine tuned using the long training fields (LTF).

The channel is first estimated using known legacy-LTF first for the legacy mode to be able to decode the signaling field. Then, the high throughput LTFs (HT-LTF) in the preamble are used for MIMO channel estimation [1]. For each subcarrier, the LTF
signals form an orthogonal matrix (containing 1’s and -1’s) in the frequency domain

\[ P = \begin{bmatrix}
1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 \\
-1 & 1 & 1 & 1 \\
\end{bmatrix} \quad (6.1) \]

which allows us to arrive at the maximum-likelihood (ML) estimate of the MIMO channel in a low-complexity manner [44].

The task of the MIMO decoder is to separate individual spatial streams from each other. As we mentioned earlier, the MMSE decoder is optimal among the class of linear detectors in the sense of minimizing the mean-squared error. Since the matrix inversions in the MMSE solution pose complexity problems for higher order MIMO configurations, a matrix inversion free MMSE MIMO decoder is implemented based on [45] in this testbed. The received symbols after the MIMO decoder are deinterleaved, deparsed and then passed through the depuncturer and the soft-decision Viterbi decoder. Since it is a packet based system, all the synchronization and estimations are performed individually on a per packet basis.

RF chains of the testbeds are using dual band Maxim MAX2829 RF transceiver ICs [46], and 12 bits Analog Devices AD9863 ICs [47] for ADC/DAC operations as well as other components such as automatic gain controller and various filters. Other hardware related details and typical RF impairment performances of the testbeds can be found in [38, 41].

Using the testbed as a building block, we implemented our energy efficient link adaptation protocol as follows. First, we implemented a modified version of the stop-
and-wait automatic-repeat-request (ARQ) protocol to ensure that no information is lost during a file (data) transfer scenario. The transmitter sends the data in 100 packet blocks along with a special sounding packet that the receiver uses for optimal mode calculation. A sequence number is assigned to each individual packet, and the receiver determines the status of the packets (whether they are lost) by examining the sequence numbers. The receiver then responds with a block acknowledgment (ACK) packet for every 100 packets transmitted. This feedback packet is transmitted using the highest power available to ensure reliable reception and contains information about the packets’ status in the previous block and the parameters for the optimal mode to be used for the next block. We assign a unique number, \( m \), for every 6-tuple and the receiver only feeds back the optimal mode’s number, \( m^* \), to the transmitter to minimize the amount of information that needs to be fed back. Finally, the transmitter reconfigures itself to implement the optimal mode that was just communicated to it by the receiver via the ACK packet. It then transmits the next block of data along with lost packets from the previous block.

Our proposed link adaptation protocol is implemented on the testbed’s microprocessor. The microprocessor communicates with the baseband FPGA in order to gather the inputs needed by the link adaptation engine as shown in Fig. 6.2. These include the estimated channel matrices for every subcarrier and the noise level estimates per antenna. Once the link adaptation engine gathers these inputs upon receiving the sounding packet, it computes the optimal mode with index \( m^* \), which is then transmitted back to the sender via ACK packet.

A laptop computer is used at each node to initialize and control the radio as well
as to save and display the statistics. Radio to laptop communication is established via Ethernet, and a graphical user interface (GUI) is developed to display the real time statistics about the link, such as the parameters of the optimal mode, SNR, PER,
energy consumption of the link, and status of the file transfer. A screen-shot of the GUI is shown in Fig. 6.3.

### 6.2 Testing Environment

Several experiments were conducted on the fifth floor of the UCLA Engineering IV building. In all experiments, the transmitter (Fig. 6.4(b)) was fixed at a location inside the WISR laboratory, while the receiver node was moved on a cart along the path shown in Fig. 6.4(a).

Carrier frequency of 2.412 GHz, at the lower edge of the ISM band, was used since it was the cleanest band in the testing environment, in terms of the interference caused by other Wi-Fi devices. The cart carrying the receiver was moved at a slow pedestrian speed along the path starting from the front door of the WISR LAB and ending at the same location for each experiment. Therefore, the channel shows distinctive non-line-of-sight characteristics for most of the time, and shows line-of-sight characteristics when the receiver is in front of the laboratory. There were also a few people walking by during the tests contributing to the dynamics in the channel, which we refer to as mobility.

The default link adaptation parameter values used in the experiment are listed in Table 6.1. There are 7 different choices for the per antenna transmit power due to hardware limitations. The system operates within a 20 MHz wide bandwidth using 52 active subcarriers out of 64 [1]. The rest of the subcarriers are used as pilots and guard bands. The packet length is chosen as 1 KBytes, in order to be able to make
Figure 6.4: Measurement environment and pictures of the transmitter and receiver testbeds.

comparisons with the other published work\textsuperscript{1}. We used the “complete” energy model for the majority of the experiments, but we also performed the same tests after changing

\textsuperscript{1}LQM to PER mapping values are given for $L = 1$KBytes in [9] and an interpolation method is proposed to find PER for different packet lengths using the $L = 1$ KBytes case as a reference.
Table 6.1: Default experiment parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_t$</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>$N_r$</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>$N_{ss}$</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>$q$</td>
<td>BPSK, QPSK, 16QAM, 64QAM</td>
</tr>
<tr>
<td>$N_{sc}$</td>
<td>52</td>
</tr>
<tr>
<td>$BW$</td>
<td>20 MHz</td>
</tr>
<tr>
<td>$P_t$</td>
<td>-3.36, -0.46, 3.54, 8.04, 12.5, 17.0, 22.0 dBm</td>
</tr>
<tr>
<td>$L$</td>
<td>1 KByte</td>
</tr>
<tr>
<td>$f_c$</td>
<td>2.412 GHz</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0.35</td>
</tr>
<tr>
<td>$\psi$</td>
<td>9.51, 9.76, 9.80, 9.83 (^2)</td>
</tr>
<tr>
<td>$E_{total}$</td>
<td>inclusive: $E_{rf} + E_{base}$</td>
</tr>
</tbody>
</table>

these parameters for comparison purposes.

### 6.3 Experimental Results

We performed various experiments for different purposes. In this section, we first present the results of the link adaptation experiments, where we used the default values

\(^2\)The PAPR, $\psi$, is modulation dependent and the values are for BPSK, QPSK, 16QAM, and 64QAM respectively. These values are obtained from the OFDM signals, including all the preamble and payload sections of the packet, in simulations. The transmit power, $P_t$, values here are per antenna and measured in a lab environment.
that are given in Table 6.1. We also performed the same set of experiments with incomplete energy consumption models and tweaked parameter sets. For instance, we set $E_{\text{base}} = 0$ in the link adaptation engine and observe the mode selection trends for such systems where the baseband energy consumption is ignored. In addition to the link adaptation experiments, we performed experiments with fixed modes (no link adaptation) in order to test the PER prediction performances of different methods such as the LQM approaches.

6.3.1 Link Adaptation Results

In this section, we present the energy efficiency maximization results along with our conclusions based on these results. The plots in Fig. 6.5 show the energy efficiency and the received SNR obtained from an experiment that we performed on the path shown in Fig. 6.4(a) using the default parameters shown in Table 6.1.

SNR decreases for the first 300 seconds due to the path loss as we move away from the transmitter and it increases as we return back towards to the tx location. In addition to the large scale path loss effects, we see the shadowing effects in the recorded SNR, due to turning around a corner and the fast fading effects due to the mobility in the channel. The link adaptation algorithm reacts to these channel changes inclusive of the fast fading by changing its mode of operation. Fig. 6.6 depicts the selected modes over the duration of the experiment.

As was mentioned earlier, the proposed link adaptation algorithm uses the time varying sounding periods to track the fast changes in the channel in an opportunistic manner. The optimal mode is therefore dependent on the instantaneous characteristics
Figure 6.5: Energy Efficiency vs time (Upper Plot), and received SNR for the 1st antenna (measured via the sounding packet which uses the maximum $P_t$) vs Time (lower plot).

of the channel and this is evident in Fig. 6.5 where the energy efficiency performance of the link closely follows the fluctuations in the channel quality. However, it is still instructive to identify the general trends in the mode selection algorithm as a function of SNR. These general trends can be useful for simpler implementations based on average
Figure 6.6: Optimal mode parameters vs time.

SNR information.

We included the pie charts that are showing how frequently we picked each specific value as optimal throughout the experiment. We observe from Fig. 6.6 and Fig. 6.7 that the algorithm chooses the highest order constellation, 64QAM, 68% of the time. This is because at high SNRs (short distances), higher order constellations are preferred in terms of energy efficiency since they reduce the transmission time of packets. The cost
of operating with a higher order constellation is less significant as the chosen transmit
drivers (and the PA energy consumptions as a result) are low. As the distance increases
(low SNR regime), the PA energy consumption dominates due to the necessity of using
higher transmit powers. Therefore, lower order constellations are preferred to be able
to operate in low SNR regime instead of increasing the transmit power even further.
This is evident in Fig. 6.6. We observe that as SNR decreases, the algorithm starts
picking 16QAM more often, and then switches to QPSK (10% of the time) around 300
seconds when the rx is at the farthest distance from the transmitter.

In the high SNR regimes (shorter distance), $E_{\text{base}}$ becomes comparable to $E_{\text{rf}}$ due
to the fact that we do not need high transmit power to support 64QAM or 16QAM,
and hence smaller $N_{ss}$, $N_t$, and $N_r$, are chosen to reduce the energy consumption as
long as the $TH_{\text{target}}$ is satisfied. As the SNR goes lower, the algorithm reacts with first
increasing $P_t$ (stabilizes the received SNR) and increasing $N_r$ (achieves spatial diversity)
to be able to maintain higher order constellations. The algorithm turns on additional
tx antennas only when it needs to transmit additional spatial streams in order to satisfy
the $TH_{\text{target}}$ via lower constellations. It is costly in terms of the energy consumption to
increase $N_{ss}$ at short distance, and the channel is not good enough to support multiple
spatial streams at long distances. This is why we never observed 3 or 4 spatial streams
in the experiment. The protocol chose single spatial stream for 83% of the time. It
should also be noted that increasing the $N_t$ as SNR decreases is not preferred in the first
place, since due to the radio implementation, we employ CDD as a transmit diversity
technique which does not provide the strong diversity gain of other methods, such as
STBC or beamforming.
6.3.2 Experiments with Tweaked Parameter Sets and Incomplete Energy Models

In addition to the default experiments that we presented in the previous section, we performed various experiments by tweaking the energy consumption model and by excluding some of the optimization parameters from the search space. In this section, we present how these changes affect the mode selection trends by comparing to the default experiment that was presented in the previous section. All the other parameters and models remain the same unless otherwise stated.

For the first set of experimental results that we presented in this chapter, we used the default settings presented in Table 6.1. We will refer to this experiment as the default experiment for presentation convenience. In Fig. 6.7, we show the percentages of the optimal parameters that were selected throughout the experiment. For instance, the link adaptation engine picked one spatial stream modes 83% of the time as seen in the relevant subplot.

We will present the results corresponding to the following modifications:

1. Set $E_{base} = 0$.
2. Set $E_{ref} = 0$.
3. Set $\eta = 0.70$.
4. Set $N_r = 4$, nonadaptive.

These modifications might be seen as hypothetical or extreme, however they allow us to understand the trends more clearly.
1. Set $E_{\text{base}} = 0$: In earlier chapters, we mentioned that there are papers employing incomplete energy consumption models, which include only the RF energy consumption and ignore other sources of energy draw. In some papers, such as [19], only PA consumption is considered, even the energy consumption of receiver RF sections are ignored. In order to understand how the energy efficiency maximization protocol picks optimal modes when we only ignore the baseband energy consumption, we set $E_{\text{base}} = 0$ in the link adaptation engine, performed the same
Figure 6.8: Optimal mode statistics for the experiment where we set $E_{base} = 0$.

experiment and observed the mode selection trends. Fig. 6.8 shows that the link adaptation protocol lowers the transmit power by 0.5 dB, and tends to use higher number of receive antennas. Removing the baseband consumption out of the picture makes the PA consumption a more significant contributor to the total energy consumption, hence we see a decrease in optimal $P_t$ values. Total energy consumption is still dependent on $N_r$ via the $E_{base}$, however $N_r$ becomes clearly less significant in the total consumption. As a result, the link adaptation protocol
uses more receive antennas on average to achieve higher spatial diversity in the channel.

2. Set $E_{rf} = 0$: This case might be very hypothetical and unrealistic for today’s communication systems, since in majority of the channel scenarios, the energy consumption of the RF sections dominates. However, baseband energy consumption can be a dominant factor for some short range communication systems in future. Regardless of the practical implications, we experimented this case and
presented the results in Fig. 6.9. As expected, since we are ignoring all the PA and other RF related contributions, the transmit power and $N_t$ are used at their maximum\(^3\) since there is no efficiency penalty for doing so. By boosting the transmit power and antenna configuration, we automatically increase the received SNR, which makes the link capable of supporting 64QAM most of the time. This shortens the transmission time and saves energy.

3. \textit{Set }$\eta = 0.70$: We considered the use of a PA with higher efficiency in this case. Even though 70\% efficiency may not seem very realistic for today’s PAs, we picked this high efficiency number for comparison clarity. For the same transmit power, this type of PA would consume half the energy compared to the default experiment. Results are similar to the above case, however not as extreme, as we show in Fig. 6.10. The link adaptation protocol uses a slightly higher $P_t$, but significantly higher $N_t$, since this is another way of increasing the total transmit power. Since higher $N_t$ already provides some sort of spatial diversity, a lower $N_r$ (compared to the default experiment) is preferred in order to decrease the energy consumption.

4. \textit{Set }$N_r = 4$: For this case, we fixed the number of receive antennas to 4. As a result, the link achieves higher spatial diversity and the channel is more suitable for carrying higher number of spatial streams. We have seen earlier that the link adaptation protocol usually picks lower $N_{ss}$ since it requires a lower operating

\(^3\)In only a few instances, the maximum was not used since the computed EE’s were the same for the maximum and one level lower $P_t$ as the PER’s were both 0. This is why the average $P_t$ was 21.88 dBm, instead of 22.04 dBm. The same situation was the reason for not observing $N_t = 4, 100\%$ of the time.
Figure 6.10: Optimal mode statistics for the experiment where we set the PA efficiency to $\eta = 0.70$.

SNR, which in turn translates into lowering the transmit power for higher energy efficiency. However, for this particular case of $N_r = 4$, the link observes higher received SNRs, and hence we can transmit multiple spatial streams with a higher modulation format to reduce the transmission time. We observe this in Fig. 6.11, which shows that unlike the default experiment, $N_{ss} = 2$ is chosen 84% of the time and 64QAM is chosen 76% of the time. We also observe that the transmit
Figure 6.11: Optimal mode statistics for the experiment where we used fixed number of receive antennas, $N_r = 4$.

power is significantly low (11.5 dBm) since the channel provides enough spatial diversity.

6.3.3 PER Prediction Results

Additional experiments were performed in order to test the performance of the PER prediction algorithm, which is an important part of the mode selection process.
The predicted and observed PER results are presented in Fig. 6.12 for the 4x4, QPSK, rate 3/4, 2 spatial streams mode. The experiment was carried out in the second half of the route in Fig. 6.4(a). The transmitter sends 1000 packets with the same mode in each sounding period, we predict the PER using the method presented in Section 3.4.3 and we record the real PER observed at the end of the sounding period. For comparison purposes, we also predict the PER using the exponential effective SNR mapping (EESM) method from [9], which was shown to perform very similarly to other link quality metric assisted look-up table methods in terms of prediction accuracy. The EESM method uses exponential mapping from PPSNR to link quality metric, which is then used to predict the PER using pre-simulated and curve fitted values stored in look-up tables [9]. The mapping function was given in Section 3.4.2.

As can be seen in Fig. 6.12, both the method we presented in Section 3.4.3 and the EESM method catch the PER trend when it jumps to 100% at low SNRs. However, both methods produce PER predictions that are off from the observed PER in (0%-100%) region. This is because none of the PER prediction methods are accurate in theory due to the complex nature of the task, and none of the published methods account for the hardware impairments when calculating the PER.

It can be argued that it is not very problematic when we predict a higher PER than the actual one, since it might only have a small degradation in the overall link adaptation performance if there are many other modes available in the search space. On the other hand, the case of predicting a much lower PER than the actual (outage) should be avoided, since it will cause violation of the QoS constraints and more importantly poor overall performance when the link adaptation algorithm happens to pick that specific
mode which observes higher PER than the predicted. In order to avoid this situation, we introduce a small back off in the calculated PPSNRs to make more pessimistic (safer) PER predictions. We compare the method in Section 3.4.3 using different amounts of back off with the EESM method in terms of the outage probability $P_{out}$ (6.2) and the

Figure 6.12: Predicted and observed PER results. 0 dB PPSNR back off.
Table 6.2: PER prediction performances

<table>
<thead>
<tr>
<th>PPSNR b.o.</th>
<th>$\bar{\epsilon}^2$</th>
<th>$P_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sec. 3.4.3</td>
<td>EESM [9]</td>
</tr>
<tr>
<td>0 dB</td>
<td>0.639</td>
<td>0.643</td>
</tr>
<tr>
<td>1 dB</td>
<td>0.609</td>
<td>0.608</td>
</tr>
<tr>
<td>2 dB</td>
<td>0.634</td>
<td>0.588</td>
</tr>
<tr>
<td>3 dB</td>
<td>0.694</td>
<td>0.632</td>
</tr>
</tbody>
</table>

mean-squared error\(^4\) $\bar{\epsilon}^2$ (6.3) in Table 6.2.

$$P_{out} = \text{Prob}(\text{PER}_{\text{predict}} < \text{PER}_{\text{observed}})$$ (6.2)

$$\bar{\epsilon}^2 = \frac{1}{N} \sum_{t=1}^{N} \left| \log (\text{PER}_{\text{predict}}(t)) - \log (\text{PER}_{\text{observed}}(t)) \right|^2$$ (6.3)

The results in Table 6.2 show the necessity of the PPSNR back off, since the outages occur frequently (15.8% and 18.8% of the time) for both methods when no back off is applied. 3 dB or higher back off is needed to reduce $P_{out}$ to a reasonably small number.

Both prediction methods perform similarly in terms of the MSE. However, the method presented in this work outperforms the EESM method in terms of the outage percentages. More importantly, it is a closed form mathematical expression and does not require any extensive simulations and/or curve fitting as is required for the EESM and other link quality metric based methods.

\(^4\)Since we send 1000 packets in each measurement window, our PER accuracy is good only in the 1%-100% range. We did not include the measurements, in which we observed $\text{PER}_{\text{observed}} < 1\%$, in the MSE or outage probability calculations.
6.4 Computational Complexities

In this section, we report on the computational complexity of the proposed link adaptation protocol and compare it to the computational complexity of the brute force search method in Table 6.3.

There are two processes that have dominant effects on the complexity of the link adaptation engine. The first is the PPSNR calculation and the second is the PER calculation. The brute force search method requires us to calculate PPSNR for every transmit power level and for direct calculation of PPSNR using (2.5), we need to perform matrix inversions for each transmit power level. The number of transmit power levels \( (N_p) \) thus has a multiplicative impact on the complexity. Although the new PPSNR formulation in (4.8) requires the computation of SVD, we don’t need to compute it for every transmit power level. Once we calculate it for a reference transmit power level, we can then simply scale the singular values for other levels.

Keeping in mind that the SVD and matrix inversion operations have roughly the same complexity, let’s assume that the calculation of PPSNRs (for all the subcarriers) for a specific \( P_t \) for all 5-tuples \( (N_{ss}, N_l, N_r, q, r) \) via either (2.5) or (4.8) is \( \Psi \). PPSNR calculation complexity of the brute force search using (2.5) will thus be \( N_p \Psi \). However the proposed method has the complexity \( \Psi \) since we do not perform SVDs for every transmit power level (so that \( N_p \) is not multiplied). Notice that the SVD based pruning algorithm excludes some of the PPSNR calculations. We do not calculate the PPSNR for the excluded modes after calculating the SVD, hence it provides further complexity reduction. However, this is not a huge reduction since we already calculated the SVD.
for those modes. Therefore, we did not include this reduction in this section as we are discussing the dominant factors rather than computing the exact complexities.

Now, let’s assume that the computational complexity of the PER calculations for all 5-tuples is \(\Xi\). The complexity for the brute force search method will therefore be \(N_p\Xi\). Instead of repeating the PER calculations \(N_p\) times, simulation results suggest that we achieve a near optimal mode in 2.5 \(P_t\) level iterations using Algorithm 1. This is only \(\frac{2.5}{32} \approx \frac{1}{13}\) th of the computational complexity of the brute force search method, if we use \(N_p = 32\). Moreover, the SVD pruning algorithm and Algorithm 1 excludes 38% of the 5-tuples from the search space throughout the simulation for \(\tau = 0.5\) without performing any PER calculations (See Fig. 5.7). The PER computational complexity of the proposed protocol is therefore \(\frac{38 \times 2.5}{100}\Xi \approx \Xi\).

In summary, assuming that inversion of the largest matrix roughly dominates the performance, our proposed protocol reduces the computational complexity by more than a factor of \(\frac{1}{N_p}\) while causing only about 2% performance degradation.
CHAPTER 7

Conclusions

In this work, we presented a low-complexity PPSNR based link adaptation protocol for throughput or energy efficiency maximization of practical MIMO-OFDM communication links. The proposed algorithm tracks the channel changes fast enough so as to react to fast fading and chooses the optimal mode for the current state of the channel.

The proposed solution employs a novel formulation of the PPSNR which does not require any matrix inversions, and greatly reduces the computational complexity. This PPSNR formulation could be used for any link adaptation algorithm that uses the PPSNR information as the error rate indicator metric such as those outlined in [9].

Transmit power usually has many choices in wireless communication systems, hence it expands the search space greatly. To solve this problem, we analyzed the energy efficiency and transmit power relationship isolated from all other parameters, and we proved the existence and uniqueness of the optimal transmit power when we fix other transmission parameters. The proof also reveals that the energy efficiency is a single-peaked quasiconcave function of the transmit power. Even though solving for the optimal transmit power requires computationally intense operations such as derivatives
of the sum of $Q$ functions, the iterative algorithm that we presented finds the close to optimal solution in 2-3 iterations without requiring any complicated operations, such as taking derivatives. This greatly reduces the computational complexity of the link adaptation process since we took the transmit power dimension out of the search space.

Inspired by the new PPSNR formulation, we proposed further pruning of the search space using the singular value ratios. The basic idea of the method is to remove the high spatial stream modes, that the channel can not support, from the search space. For the channels with low singular value ratios, we show that we expect to see big differences between the PPSNRs of different spatial streams. These types of channels are not suitable for transmitting multiple spatial streams, especially when energy efficiency is a concern. We look at the singular value ratios of the channel and exclude the high spatial stream modes from the search space before we start performing any mode selection related computations. This provided us an additional 30%-40% shrinkage of the search space.

Through realistic end-to-end bit level simulations, we showed that the performance of the proposed protocol is only about 2% lower than the brute force search method while the complexity is reduced by orders of magnitude.

In appendix B, we provided the derivation of the PPSNR for MMSE decoders operating in the presence of channel estimation error. In practical systems, the channel estimates are inherently noisy and/or outdated. As a result, the performance of the system is degraded. The PPSNR that we derived in Appendix B gives a very accurate estimate of what the performance of the system will be in such scenarios. Accuracy of the analysis is proven via simulations.
Aside from the algorithmic contributions, we reported on the development of the link adaptation protocol on a mode-rich 4x4 MIMO-OFDM testbed. To the best of our knowledge, this is the first reported testbed that is capable of performing energy-efficient fast link adaptation using the PHY layer information. This testbed allowed us to validate the operation of the link adaptation protocol, observe the mode selection trends, and experimentally test different PER prediction algorithms in an indoor environment under real-time constraints.
APPENDIX A

Proof of the Proposition

The derivative of $EE(P_t)$ with respect to $P_t$ is

$$\frac{\partial EE(P_t)}{\partial P_t} = - \left( LN_t T_{on} \frac{\psi}{\eta} \right) \frac{h(P_t)}{\left( \frac{\psi}{\eta} P_t N_t T_{on} + E_o \right)^2} (A.1)$$

For the rest of the proof, we define the function $h(P_t)$ as

$$h(P_t) = PSR(P_t) - PSR'(P_t) \left[ P_t + \frac{E_o}{\frac{\psi}{\eta} N_t T_{on}} \right] (A.2)$$

Let $P_t^f$ be the inflection point at which the S-shaped function $PSR$ transitions from convexity to concavity. For any $P_t \in [\delta, P_t^f)$, $PSR(P_t)$ is\(^1\) strictly convex. Hence, setting $y = \delta$ in the first order condition\(^2\) for convexity, we obtain the following condition.

$$PSR(\delta) > PSR(P_t) + PSR'(P_t)(\delta - P_t)$$

$$0 > PSR(P_t) - P_t PSR'(P_t) + \delta PSR'(P_t) \geq h(P_t) \geq 0 (A.3)$$

\(^1\)The purpose of defining $\delta$ as the minimum point of the domain instead of 0 is to be mathematically accurate and avoid the $P_t$ region where $PSR$ is constant and equal to 0 due to $(P_{b-u})^n e^n \geq 1$ in (3.17).

\(^2\)A differentiable function $f(x)$ is convex on interval $I \subset \mathbb{R}$ if and only if $f(y) \geq f(x) + f'(x)(y - x)$, and concave if and only if $f(y) \leq f(x) + f'(x)(y - x)$, $\forall x, y \in I$. These are known as the first order conditions.
where we used the fact that $PSR(\delta) = 0$ on the left hand side. It directly follows from (A.3) that $h(P_t) < 0$, since the third term in $h(P_t)$, i.e. $PSR'(P_t) \frac{E_o}{\frac{q}{q} Nt_{on}}$, is nonnegative due to the fact that the derivative of the PSR is positive for the defined region, $PSR'(P_t) > 0$, and $E_o \geq 0$, since it is the energy consumption of some blocks. Hence, we first state the following fact

(i) for $P_t \in [\delta, P_f^t)$, $h(P_t) < 0 \Rightarrow EE'(P_t) > 0$.

The equation of the straight line that is tangent to the $PSR(P_t)$ function at a point $(\dot{P}_t, PSR(\dot{P}_t))$ is

$$g(P_t) = PSR(\dot{P}_t) + PSR'(\dot{P}_t)(P_t - \dot{P}_t) = h(\dot{P}_t) + PSR'(\dot{P}_t) \left[ P_t + \frac{E_o}{\frac{q}{q} Nt_{on}} \right]. \quad (A.4)$$

Following the above definition, we see that $h(P_t)$ is the ordinate of the tangent line [48] when we set the abscissa to be $-\frac{E_o}{\frac{q}{q} Nt_{on}}$. As $P_t \to \infty$, the tangent line becomes $g(P_t) = 1$, therefore we state the following fact

(ii) $\lim_{P_t \to \infty} h(P_t) > 0$.

The facts (i) and (ii) together with the fact that $h(P_t)$ is a continuous function (for the defined domain) imply that there has to be a transmit power, $\tilde{P}_t$, that satisfies $h(\tilde{P}_t) = 0$ as the continuous function $h(P_t)$ has to cross zero, as it transitions from negative (i) to positive (ii). This proves the existence of $\tilde{P}_t$ that makes the derivative equal to zero, $\frac{\partial EE(P_t)}{\partial P_t} \bigg|_{P_t=\tilde{P}_t} = 0$.

The statement (i) also implies that any optimal point, which has to satisfy $h(P_t) = 0$, has to occur in the concave region of the $PSR(P_t)$. For any two points $\dot{P}_t$ and $\tilde{P}_t$ in the concave region, $\dot{P}_t > \tilde{P}_t$ implies $h(\dot{P}_t) > h(\tilde{P}_t)$, hence there must be only one optimal point satisfying $h(P_t) = 0$. This proves the uniqueness of $\tilde{P}_t$ which is the global
maximizer of the unconstrained problem of maximizing $EE(P_t)$.

Similar to the method that we used for (A.3), now we use the fact that for any transmit power greater than the inflection point, $P_t > P_f$, and obviously for $P_t > \tilde{P}_t$, the $PSR(P_t)$ function is strictly concave. We then obtain the following inequality for any $P_t > \tilde{P}_t$ by plugging $y = \tilde{P}_t$ in the first order concavity condition,

$$PSR(\tilde{P}_t) < PSR(P_t) + PSR'(P_t)\left(\tilde{P}_t - P_t\right)$$  \hspace{1cm} (A.5)

$$PSR'(\tilde{P}_t)\left[\tilde{P}_t + \frac{E_o}{\eta N_t T_{on}}\right] < PSR(P_t) + PSR'(P_t)\left(\tilde{P}_t - P_t\right)$$  \hspace{1cm} (A.6)

$$\tilde{P}_t\left[PSR'(\tilde{P}_t) - PSR'(P_t)\right] < PSR(P_t) - P_t PSR'(P_t) - \frac{E_o}{\eta N_t T_{on}}\left[PSR'(\tilde{P}_t)\right]$$  \hspace{1cm} (A.7)

where we used the fact $h(\tilde{P}_t) = 0$ in (A.6) to replace the left hand side of (A.5) and reorganized the terms to get (A.7). In (A.7), we used the fact that $PSR'(P_t)$ is a decreasing function in the concave region. $h(P_t)$ is clearly greater than the right hand side of (A.7), hence $h(P_t) > 0$ and $EE'(P_t) < 0$ for any $P_t > \tilde{P}_t$.

Similar to the previous two cases ($P_t < P_f$ and $P_t > \tilde{P}_t$), we can prove that $h(P_t) < 0$ and $EE'(P_t) > 0$ for $P_f < P_t < \tilde{P}_t$.

These prove that $EE(P_t)$ is a single-peaked function as it is strictly increasing for $P_t < \tilde{P}_t$ since $EE'(P_t) > 0$; and strictly decreasing for $P_t > \tilde{P}_t$ since $EE'(P_t) < 0$. This directly implies that $EE(P_t)$ is a quasiconcave function following the definition of quasiconcavity\(^3\).

In the above discussion, we proved that there exists a unique $\tilde{P}_t$ that maximizes the unconstrained problem. For the power constrained problem, if $\tilde{P}_t > P_t^{\text{max}}$ then $P_t^{\text{max}}$

\(^3\)The function $f(x)$ defined on a convex set $S$ is quasiconcave if $\forall x, y \in S$ and $\lambda \in [0, 1]$ we have $f(\lambda x + (1 - \lambda)y) \geq \min(f(x), f(y))$. 

111
becomes the optimal solution since $EE(P_t)$ is an increasing function of $P_t$ for $P_t < \tilde{P}_t$, and if $\tilde{P}_t < P_t^{\text{min}}$ then $P_t^{\text{min}}$ becomes the optimal solution since $EE(P_t)$ is a decreasing function for $P_t > \tilde{P}_t$. Hence, we obtain the global optimum for the constrained problem as

$$P_t^* = \min(P_t^{\text{max}}, \max(P_t^{\text{min}}, \tilde{P}_t)).$$

(A.8)
APPENDIX B

Derivation of the PPSNR for MMSE Decoders in the Presence of Channel Estimation Error

Perfect channel state information (CSI) is usually assumed in the literature when simulating or analyzing the performance of linear MIMO decoders [49, 50]. However, in practice the channel estimates are inherently noisy. Important work [51, 52] has characterized the error rate performance of ZF decoders in the presence of channel estimation error. Nevertheless, less is known for the case of MMSE decoders in practical scenarios. For ZF and MMSE decoders, the joint effect of phase noise and channel estimation error is considered in [53] and the performance is analyzed in terms of the degradation in signal-to-noise-plus-interference-ratio (SINR) without expressing the closed form performance indicators or error rate analysis. In this section of the dissertation, we analyze the MMSE based receivers in the presence of channel estimation error, and derive a closed form post-processing SNR expression, which provides an accurate estimate of the error rate performance. We believe that it is a very useful tool for link adaptation protocols and for error rate analysis in general. Accuracy of the analytical results is verified through simulations.
For the ensuing derivations, we assume that the precoding matrix \( R = I \), hence 
\( N_{ss} = N_t \), and we will use these two interchangeably in this section. However, the 
derivations and results in this section are very well applicable to any linear precoding 
matrix. Following the same notation with Chapter 2, for an individual spatial stream 
and subcarrier, we write the \( N_r \times 1 \) received signal vector \( y \) as

\[
y = \sqrt{P_t} H x + n
\]

(B.1)

where \( x \) is the transmitted signal vector, \( H \) is the \( N_r \times N_{ss} \) effective channel matrix, and 
\( n \) is the \( N_r \times 1 \) additive Gaussian noise vector with zero mean and covariance matrix 
\( E[nn^*] = N_0 I \). We assume an uncorrelated Rayleigh flat fading channel, i.e. entries of 
\( H \) are i.i.d. zero mean circularly symmetric complex Gaussians (ZMCSCG) with unit 
variance. The signal power at each transmit antenna is assumed to be equal to \( P_t \).

We assume that the receiver estimates the transmitted signal vector using the MMSE 
decoder which we denoted earlier as

\[
W = \frac{1}{\sqrt{P_t}} \left[ H^* H + \frac{N_0}{P_t} I \right]^{-1} H^*
\]

and the PPSNR is computed as

\[
\gamma_k = \frac{P_t \left| (WH)_{k,k} \right|^2}{P_t \sum_{l \neq k} \left| (WH)_{k,l} \right|^2 + N_0 \left( WW^* \right)_{k,k}}
\]

(B.3)

This definition of PPSNR holds if the channel is perfectly known at the receiver. 
However, in practice, the channel matrix has to be estimated by the receiver, and the 
estimated channel is inherently noisy in practical systems. We model the estimated 
channel matrix as

\[
\hat{H} = H + \Delta H
\]

(B.4)
where $\Delta H$ denotes the estimation error matrix which is uncorrelated with $H$, and its entries are ZMCSCG with variance $\sigma_e^2$. The quality of channel estimation is captured by $\sigma_e^2$, which can be appropriately estimated or computed depending on the channel estimation method.

### B.1 PPSNR Derivation for Practical Systems

Our aim in this section is to derive the PPSNR for practical MIMO systems which observe channel estimation errors. The receiver uses the estimated channel $\hat{H}$ to calculate the MMSE decoder as

$$
\hat{W} = \frac{1}{\sqrt{P_t}} \left[ (H + \Delta H)^* (H + \Delta H) + \frac{N_0}{P_t} I \right]^{-1} (H + \Delta H)^*
$$

(B.5)

We write the imperfect MMSE solution as $\hat{W} = W + \Delta W$. Now, the MMSE estimate of the signal vector becomes

$$
\hat{x} = (W + \Delta W) y = \sqrt{P_t} WHx + \sqrt{P_t} \Delta WHx + Wn + \Delta Wn
$$

(B.6)

We observe that there are additional interference and noise terms caused by the MMSE error matrix, $\Delta W$. We denote the post detection noise as $\hat{n} = \sqrt{P_t} \Delta WHx + Wn + \Delta Wn$. With this definition for the post detection noise, the PPSNR of the $k^{th}$ spatial stream in the presence of channel estimation error can be expressed as

$$
\tilde{\gamma}_k = \frac{P_t \left| (WH)_{k,k} \right|^2}{P_t \sum_{i \neq k} \left| (WH)_{k,l} \right|^2 + (E [\hat{n}^*])_{k,k}}
$$

(B.7)
where we replaced the original noise covariance in (B.3) with the covariance of \( \hat{n} \), which is calculated as

\[
E[\hat{n}\hat{n}^*] = P_t E[\Delta WHxx^*H^*\Delta W^*] + E[Wnn^*W^*] \\
+ E[Wnn^*\Delta W^*] + E[\Delta Wnn^*W^*] \\
+ E[\Delta Wnn^*\Delta W^*] \tag{B.8}
\]

In order to calculate the terms in (B.8), we need to first derive \( \Delta W \). For small \( \sigma_e^2 \), the \( \Delta H^*\Delta H \) term in (B.5) becomes negligible compared to others. Hence, we can rewrite (B.5) as

\[
\hat{W} \approx \frac{1}{\sqrt{P_t}} \left[ H^*H + \frac{N_0}{P_t} I + H^*\Delta H + \Delta H^*H \right]^{-1} (H + \Delta H)^* \tag{B.9}
\]

which can be further simplified using the matrix approximation \( (P + \epsilon^2 Q)^{-1} \approx P^{-1} - \epsilon^2 P^{-1}QP^{-1} \) for small \( \epsilon^2 \). Let us also define \( K = \left( H^*H + \frac{N_0}{P_t} I \right)^{-1} \) for brevity and simplify (B.9) as

\[
\hat{W} \approx \frac{1}{\sqrt{P_t}} \left[ K - K (H^*\Delta H + \Delta H^*H) K \right] (H + \Delta H)^* \tag{B.10}
\]

\[
= \frac{1}{\sqrt{P_t}} \left[ KH^* - K (H^*\Delta H + \Delta H^*H) KH^* \right. \\
\left. + K\Delta H^* - K (H^*\Delta H + \Delta H^*H) K\Delta H^* \right] \tag{B.11}
\]

small compared to the other terms

Finally the desired error matrix becomes

\[
\Delta W \approx \frac{1}{\sqrt{P_t}} (-K (H^*\Delta H + \Delta H^*H) KH^* + K\Delta H^*). \tag{B.12}
\]

Using the above approximation, we can now calculate the terms in (B.8). We first note that the third and fourth terms in (B.8) are zero since \( E[\Delta W] \approx 0 \). The second term is \( E[Wnn^*W^*] = N_0 WW^* \), and the first term becomes \( P_t E[\Delta WHxx^*H^*\Delta W^*] = \)
\[ P_t E[\Delta WHH^*\Delta W^*]. \] Below, we calculate the first and last terms in (B.8) by plugging the error matrix (B.12) into (B.8).

\[ P_t E[\Delta WHH^*\Delta W^*] \]
\[ \approx E[\Delta WHH^*\Delta W^*] + E[\Delta WHH^*\Delta W^*] - E[\Delta WHH^*\Delta W^*] + E[\Delta WHH^*\Delta W^*] \]
\[ - E[\Delta WHH^*\Delta W^*] + E[\Delta WHH^*\Delta W^*] + E[\Delta WHH^*\Delta W^*] - E[\Delta WHH^*\Delta W^*] \]
\[ + E[\Delta WHH^*\Delta W^*] \]
\[ (B.13) \]

It can be proven that \( E[\Delta HA\Delta H^*] = E[\Delta H^*A\Delta H^*] = 0 \) for any deterministic matrix \( A \). Hence the second, third, fourth and seventh terms in (13) are zero. For the remaining terms we use the fact that \( E[\Delta HA\Delta H^*] = \sigma_e^2 \text{Tr}(A) I \), where \( \text{Tr}(A) \) is the trace operation on the matrix \( A \), and we obtain

\[ P_t E[\Delta WHH^*\Delta W^*] \]
\[ \approx \sigma_e^2 \text{Tr}(KH^*HH^*HK^*)KH^*HK^* + \sigma_e^2 \text{Tr}(HH^*HK^*H^*)KK^* \]
\[ - \sigma_e^2 \text{Tr}(HK^*HH^*)KK^* - \sigma_e^2 \text{Tr}(HH^*HK^*H^*)KK^* \]
\[ + \sigma_e^2 \text{Tr}(HH^*)KK^* \]
\[ (B.14) \]

Similarly, the last term in (B.8), \( E[\Delta W_{nn}^*\Delta W^*] \), can be computed following the same way.

\[ E[\Delta W_{nn}^*\Delta W^*] = N_0 E[\Delta W\Delta W^*] \]
\[ (B.15) \]
\[ P_t \sum_{l \neq k} |(WH)_{k,l}|^2 + \left( \frac{\sigma^2_e}{P_t} \text{Tr}(KH^*HK^*) \right) \cdot KH^*HK^* \\
+ \left( \frac{\sigma^2_e}{P_t} \text{Tr}(HKH^*HK^*H^*) \right) \cdot KK^* \\
- \left( \frac{\sigma^2_e}{P_t} \text{Tr}(HKH^*HK^*) \right) \cdot KK^* \\
+ \left( \frac{\sigma^2_e}{P_t} \text{Tr}(HH^*) \right) \cdot KK^* + N_0 WW^* + \frac{N_0}{P_t} \sigma^2_e \text{Tr}(KH^*HK^*) \cdot KH^*HK^* \\
+ \frac{N_0}{P_t} \sigma^2_e \text{Tr}(HKH^*HK^*H^*) \cdot KK^* - \frac{N_0}{P_t} \sigma^2_e \text{Tr}(HKH^*) \cdot KK^* \\
- \frac{N_0}{P_t} \sigma^2_e \text{Tr}(HK^*H^*) \cdot KK^* + \frac{N_0}{P_t} \sigma^2_e N_r KK^* \right)_{k,k} \]

(B.17)

\[ E[\Delta W \Delta W^*] \approx \frac{\sigma^2_e}{P_t} \text{Tr}(KH^*HK^*) \cdot KH^*HK^* + \frac{\sigma^2_e}{P_t} \text{Tr}(HKH^*HK^*H^*) \cdot KK^* \\
- \frac{\sigma^2_e}{P_t} \text{Tr}(HKH^*) \cdot KK^* - \frac{\sigma^2_e}{P_t} \text{Tr}(HK^*H^*) \cdot KK^* + \frac{\sigma^2_e}{P_t} N_r KK^* \]  

(B.16)

Finally, we plug \( E[\tilde{n}\tilde{n}^*] \) into (B.7) and obtain the PPSNR in the presence of channel estimation error as (B.17).

The BER of the system in the presence of channel estimation error can be found simply by plugging \( \tilde{\gamma}_k \) as the symbol SNR into the AWGN BER formulas. For example, the BER of \( k^{th} \) stream for BPSK is \( P^k_b = Q\left(\sqrt{2 \tilde{\gamma}_k}\right) \), and \( P^k_b = \frac{3}{4} Q\left(\sqrt{\frac{\tilde{\gamma}_k}{3}}\right) + \frac{1}{2} Q\left(3 \sqrt{\frac{\tilde{\gamma}_k}{5}}\right) - \frac{1}{4} Q\left(5 \sqrt{\frac{\tilde{\gamma}_k}{5}}\right) \) for gray-coded 16QAM.
B.2 Simulation Results

In order to test the prediction performance of the PPSNR obtained via analysis, we simulated transmission of thousands of packets through uncorrelated Rayleigh flat fading channels. For each SNR point on the BER plots, we randomly generate 1000 i.i.d. realizations of the channel matrix $H$. For each specific realization of the channel, we transmit 500 packets each of which carries 2000 information symbols. We perform channel estimation for each packet as explained below in Case 1.

Case 1: In our simulations, we employed the maximum likelihood (ML) channel estimation (CE) algorithm, in which the channel estimate is obtained via training symbols that are known to the receiver. During the training phase, the $N_{ss} \times N_{tr}$ training matrix $X_{tr}$ is transmitted where $N_{tr} \geq N_{ss}$ is the number of training symbols. The $N_r \times N_{tr}$ received signal is $Y_{tr} = HX_{tr} + G$ where $G$ is the $N_r \times N_{tr}$ noise matrix. Then, the ML estimate of the channel is given as [54]

$$\hat{H} = Y_{tr}X_{tr}^* (X_{tr}X_{tr}^*)^{-1}$$

(B.18)

It was shown that the optimal training signal has the property of $X_{tr}X_{tr}^* = P_t N_{tr} I$. When this orthogonal training signal is employed, the entries of $\Delta H$ are i.i.d. with $CN(0, \sigma^2_e)$, and the channel estimation noise variance is $\sigma^2_e = \frac{1}{N_{tr} P_t / N_0}$ [54]. The estimation error is caused by the AWGN in this case.

The following training signal, which is taken from 802.11n standard [1], was em-

---

1 At the receiver: after the MMSE decoder, we scale the estimate $\hat{x}$ and generate decision statistic $z_k = \hat{x}_k / (WH)_{k,k}$ for $k^{th}$ spatial stream. We then project $z_k$ onto the nearest constellation point to decide on the symbols and the bits, i.e. hard decision with gain compensation.

2 $\sigma^2_e$ can also be defined as $\sigma^2_e = \frac{N_{ss} P_t}{N_0}$ depending on SNR definition.
ployed in the simulations. $\mathbf{X}_{tr} = \sqrt{P_{t}} \mathbf{P}$ where $\mathbf{P}$ is the submatrix formed by first $N_{ss}$ rows and first $N_{tr}$ columns of the bigger matrix\(^3\) $\mathbf{P}$, i.e. $\mathbf{P} = \mathbf{P} [1 : N_{ss}; 1 : N_{tr}]$.

$$
\mathbf{P} = \begin{bmatrix}
1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1
\end{bmatrix}
$$

(B.19)

It should be noted that with this choice of the training matrix, the ML channel estimation at the receiver becomes a very simple operation since the matrix inversion, $(\mathbf{X}_{tr}\mathbf{X}_{tr}^*)^{-1}$, is now a trivial operation.

We present the simulation results for BPSK in Fig. B.1 and 16QAM in Fig. B.2 with $1 \times 4$, $2 \times 4$, $4 \times 4$ MIMO configurations. $N_{tr} = 4$ is used in all the simulations. The case of $\sigma_{e}^2 = 0$, i.e. perfect channel estimation, is also included in the results.

For each channel instance, analytical BER results are obtained by using the PPSNR derived in the previous section. Then, these BERs are averaged over all realizations of the channel.

First thing to notice in Fig. B.1 and Fig. B.2 is that for $\sigma_{e}^2 = 0$, simulation and analysis curves closely match. Performance is significantly degraded for the systems experiencing channel estimation errors. This is particularly evident for the $4 \times 4$ configurations.

As it can be seen in Fig. B.1 and Fig. B.2, our analysis gives a very tight approximation of the real performance. For BPSK $4 \times 4$, and all of the 16QAM configurations

\(^3\)The $\mathbf{P}$ matrix here is for maximum of 4 spatial streams since the standard supports up to 4 streams. $N_{tr} = 4$ for $N_{ss} \geq 3$.\)
the analysis results match the simulated performances. For BPSK 1×4, and 2×4 configurations the analysis results are upper-bounds to the real performance at high SNR, however, they are still very close to the real performances. The analysis results become tighter for higher order modulations and higher order MIMO configurations. This is because of the fact that the Gaussian assumption, which is made for the post-detection noise, is more valid at higher order modulations and MIMO configurations. At low SNRs, the total post detection noise \( \hat{n} \) is dominated by the additive white Gaussian noise component \( n \) therefore the assumption is valid even for lower configurations. However, at high SNRs the residual interference from other spatial streams becomes
dominant and  \( \hat{\mathbf{n}} \) is loosely approximated as Gaussian for lower order constellations and MIMO configurations.

It is interesting to note that in contrast to the results obtained for ZF decoder by [52], we do not observe any error floor on the performance. This is due to the fact that the channel estimation error variance \( \sigma^2_e \) for ML estimation gets smaller as SNR increases. This is the situation that occurs in practical packet based or bursty communication systems where the channel estimation is performed for every packet prior to data detection, and hence experiences the same noise variance as the data transmission. Therefore the channel estimation quality is dependent on the SNR. On
Figure B.3: BER for $4 \times 5$ QPSK for different $\sigma_e$ values. Transmit SNR is per bit and per tx antenna. $\sigma_e = 0$ corresponds to perfect CE. ZF curves are included for comparison purposes with C. Wang’s paper [52]. These parameter values are taken from Fig. 2 of [52] for comparison purposes.

the other hand, error floors are observed in [52] because of the assumption that $\sigma_e^2$ remains constant independent of the SNR. This case is investigated below in Case 2.

Case 2: In addition to ML channel estimation results, we also performed simulations with constant $\sigma_e$. Unlike the first case, the channel estimation quality is independent of the SNR. This situation might arise either when there is a ready channel estimate to be used by the receiver formed elsewhere with a different additive noise variance, or the
channel estimation is outdated and the major error in the channel estimation comes from the mobility changes in the channel.

In Fig. B.3, the BER performance of a QPSK $4 \times 5$ system is investigated for $\sigma_e = 5\%, 10\%, 20\%$ using the estimation error model in (B.4). Each packet observes a different realization of the random matrix $\Delta H$ with the designated variance $\sigma_e^2$. As expected, we observe error floor in the performance due to the constant estimation error variance as in the ZF decoder case studied in [52]. More importantly, these error floors are the same as those observed by ZF decoder because of the fact that the MMSE and ZF decoders exhibit the same behavior at asymptotically high SNR. The simulation results in this case also agree with the analysis.
REFERENCES


