# Low Complexity Wireless Communication Digital Baseband Design

by

Shunyao Wu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2017 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Chair
Hyunseok Lee
Antonia Papandreou-Suppappola

ARIZONA STATE UNIVERSITY

August 2017

# ABSTRACT

This thesis addresses two problems in digital baseband design of wireless communication systems, namely, those in Internet of Things (IoT) terminals that support long range communications and those in full-duplex systems that are designed for high spectral efficiency.

IoT terminals for long range communications are typically based on Orthogonal Frequency-Division Multiple Access (OFDMA) and spread spectrum technologies. In order to design an efficient baseband architecture for such terminals, the workload profiles of both systems are analyzed. Since frame detection unit has by far the highest computational load, a simple architecture that uses only a scalar datapath is proposed. To optimize for low energy consumption, application-specific instructions that minimize register accesses and address generation units for streamlined memory access are introduced. Two parameters, namely, correlation window size and threshold value, affect the detection probability, the false alarm probability and hence energy consumption. Next, energy-optimal operation settings for correlation window size and threshold value are derived for different channel conditions. For both good and bad channel conditions, if target signal detection probability is greater than 0.9, the baseband processor has the lowest energy when the frame detection algorithm uses the longest correlation window and the highest threshold value.

A full-duplex system has high spectral efficiency but suffers from self-interference. Part of the interference can be cancelled digitally using equalization techniques. The cancellation performance and computation complexity of the competing equalization algorithms, namely, Least Mean Square (LMS), Normalized LMS (NLMS), Recursive Least Square (RLS) and feedback equalizers based on LMS, NLMS and RLS are analyzed, and a trade-off between performance and complexity established. NLMS linear equalizer is found to be suitable for resource-constrained mobile devices and NLMS decision feedback equalizer is more appropriate for base stations that are not energy constrained.

To my Mother and Father

ACKNOWLEDGMENTS

I would first like to express my sincere gratitude to my advisor Dr. Chaitali Chakrabarti, for her continuous guidance and patience during the past two years. I am lucky to have the opportunity to continue to work with her for my PhD degree. The light in Dr. Chakrabarti's office is on even on Sundays. She is very patient and is always there to help me. It is so wonderful to work with an advisor like her.

Besides my advisor, I would like to express my gratitude to the rest of my thesis committee, I would like to thank Dr. Antonia Papandreou-Suppappola for teaching me Random Signal Theory. She always answered my questions with great patience and humor. I would like to thank Dr. Hyunseok Lee for his continued guidance. I enjoyed the opportunity to work closely with him. His skills in hardware and communication systems have helped guide my research. Finally, I would like to thank Amir Ayati for pointing me to the right papers and also providing his full duplex system model.

As a member of the Low Power Systems Lab at Arizona State University, I would like to thank my lab mates, Hsing-Min Chen, Siyuan Wei, Manqing Mao, Jian Zhou and Jiang Xiang, for their excellent friendship.

Finally I would like to thank my friends and my family for their continuous support and encouragement. They have all helped me in my pursuit of research.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

# CHAPTER 1. INTRODUCTION

In the very near future, we anticipate that machine to machine (M2M) communication will dominate internet traffic [1]. Smart sensing devices in autonomous cars, surveillance cameras, smart meters, health monitors, etc. will talk to other sensing devices without human intervention. This is the new era of Internet of Things (IoT) [2], [3]. In many scenarios, the IoT devices will have to communicate with each other or with a central station in places that lack proper network connection and even power sources. Current systems, like LTE, are complex and not energy efficient. In this thesis, we address the problem of long range communication in remote areas by designing a low energy digital baseband processor that is optimized for such scenarios.

Another important problem in wireless communication systems is achieving high spectrum efficiency. A full duplex system achieves high spectrum efficiency by transmitting and receiving in the same bandwidth simultaneously. This technology has tremendous implications in network design, for example, cellular networks can cut their spectrum needs by half. However, these systems have self-interference caused by their own transmitted signal through direct path and reflected paths. Self-interference can be reduced at the antenna end, at the analog end and at the digital end. In this thesis we study the performance of several equalization algorithms with different levels of complexity to reduce noise due to self-interference at the digital end.

## 1.1 Long Range Wireless Communication Baseband Processor Design

### 1.1.1 Motivation

To support IoT communications, there are short range systems such as WiFi, Bluetooth, Zigbee, and z-wave, and long range systems such as LoRa, Sigfox, and LTE-M [4]. While short range systems are good for indoor applications, long range systems are designed for monitoring water, gas or infrastructure health without power and backbone network connectivity [5]. Legacy cellular networks, such as those based on LTE, are too expensive in terms of power and operation cost, and not applicable for long range systems.

Several architectures have been proposed for IoT terminals for short range communications. There are commercial designs such as those from CEVA which consists of DSP with SIMD units [6], DSP with two VLIW slots from Tensilica [7], ARM core-based ARTIK series from Samsung [8]. There are also designs from academia such as the custom SIMD architecture with flexible bit-width [9] and CISC processor with reconfigurable microcode [10]. For long range IoT applications, there are a number of commercial solutions from Silicon Lab, Semtech, and TI. These are all low power versions of conventional RISC processors.

### 1.1.2 Contribution

In this thesis, we present a baseband processor architecture for long range IoT systems based on OFDMA and spread spectrum technologies [11]. Since an IoT terminal spends most of its time in the idle mode, we design an architecture which is optimized for idle mode. We perform detailed workload analysis of the two technologies and find that frame

detection is by far the most dominant workload. Since frame detection implemented using sliding window is essentially a scalar operation, we propose a scalar processor based architecture. Reducing energy consumption is extremely important, and so we introduce (i) application-specific instructions that help reduce power in register files through instruction chaining (where instructions are executed one after one without storing intermediate results), and (ii) streamline data access in memories through address generating units which hide the overhead of address calculations. The proposed architecture was synthesized using Cadence in 65 nm technology node. Preliminary synthesis results show that the area of this architecture is 0.204$\mu m^2$ and that it consumes only 2.41 nJ/cycle when clocked at 3MHz with supply voltage of 1.08V. It is different from previous idle mode works [9] [12]in that it is designed for long range IoT terminals and is based on a scalar datapath.

Since the baseband processor spends 97%-99% of its time in frame detection, next we derive a set of energy-optimal frame detection algorithm settings for different channel conditions. The derivation is based on design space exploration that considers both algorithm performance and processor energy consumption. In contrast, previous works consider only detection algorithm design [13] or only implementation [14]. We implemented two versions of coarse-grain frame detection algorithm that have different detection capabilities. Since the energy consumption depends on detection performance and the detection performance of the algorithm depends on the effect of correlation window size and threshold value, we analyzed the effects of these parameters on the energy consumption of the baseband processor. When the frame detection probability is larger

than 0.9, we find that a simplified version is more energy-efficient for good channel conditions. But for bad channel condition (SNR < -9dB), we have to lower the frame detection probability to adapt the situation, in this scenario, the baseline algorithm is better in energy consumption. In addition, we showed that in both cases, longer correlation window size and higher threshold is more energy-efficient.

## 1.2 Full Duplex Systems

### 1.2.1 Motivation

Earlier it was believed that a radio cannot both transmit and receive in the same bandwidth simultaneously [15]. Communication systems deployed either time-division or frequency-division approach instead of bidirectional communication [16]. In recent years, full duplex systems that can transmit and receiver at the same bandwidth, have been proposed. These systems achieve high spectrum efficiency and have the potential to change our current wireless network with respect to cellular, antenna arrangement [17].

The basic challenge of full-duplex systems is handling of self-interference. Figure 1.1 describes the key blocks of a full duplex terminal. The self-interference is due to transmitted signal from direct path and reflected paths. To mitigate the effect of self-interference, in the antenna part, an architecture called balun which uses two transmit antennas with different path lengths, has been used in [15]. At the analog end, tuning algorithms have been developed in [18] based on copying the analog signal at the transmitter side and using parallel fixed lines of varying delays and tunable attenuators to cancel the self-interference. For the digital cancellation part, Least Mean Square (LMS)

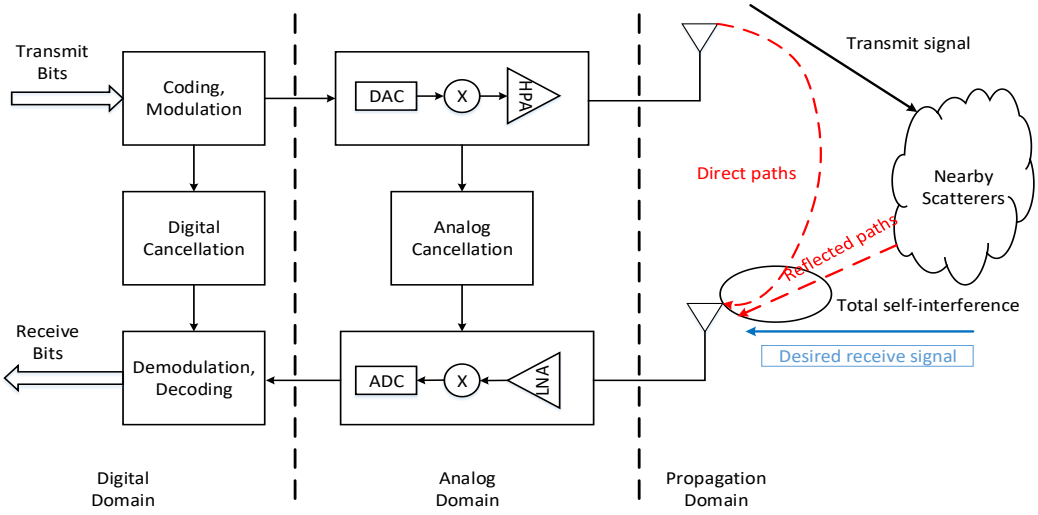


Figure 1.1 In-band Full-Duplex Terminal

[19], Least Square (LS) [20], and Maximum Likelihood Sequence Estimation (MLSE) [18] algorithms have been used.

### 1.2.2 Contribution

As outlined in [18], since the total cancellation for self-interference is about 110dB, the analog part should reduce about 60dB of noise and the digital part should eliminate the remaining 50dB of noise. To reduce the linear and non-linear noise, a strong equalization algorithm should be used. Unfortunately a strong algorithm has higher complexity and so the choice of the algorithm depends on the application requirement.

In this thesis, we studied the performance of different equalizer algorithms based on Least Mean Square (LMS), Normalized LMS (NLMS), and Recursive Least Square (RLS) for three different channel models, namely AWGN channel, Rayleigh Fading channel and real

indoor measured channel. We also studied the performance of feedback equalizers based on LMS, NLMS and RLS.

To study the effect of channel, we fixed the equalization algorithm to RLS and NLMS. We found that the RLS and NLMS performance for different channel models was similar. The performance for Rayleigh Fading channel was slightly worse and the performance for AWGN slightly better compared to the performance for real indoor channel.

Next we fixed the channel model to that of the real indoor channel, and studied the performance of the different algorithms. We found that the performance using NLMS decision feedback equalizer is the best. It achieves about 45dB noise cancellation. NLMS linear equalizer and RLS linear equalizer are next in performance. While both equalizers had noise cancellation of about 35dB, NLMS linear equalizer has lower computation complexity compared to RLS linear equalizer.

We also found that in indoor channel, equalizer with decision feedback can cancel 10 dB more non-linear noise due to the feedback loop. Due to the high noise floor in the input signal, normalization helps in improving the performance by limiting the scaling of input signal. Unfortunately, equalizers with decision feedback have higher complexity and may not be suitable for resource-constrained devices. Thus we conclude that NLMS decision feedback equalizer is suitable for base stations which are not resource constrained and

where cancellation performance is very important, and NLMS linear equalizer is suitable for terminals which are resource constrained.

## 1.3 Organization

The rest of thesis is organized as follows. Chapter 2 presents background information on long range wireless communication processor and full duplex systems. Chapter 3 presents the workload of different long range wireless communication protocols, the scalar baseband processor architecture for IoT terminal and design space exploration for deriving the energy optimal setting. Chapter 4 presents different equalizer algorithms for digital cancellation in full duplex systems and an analysis of their performance. Chapter 5 summarizes the thesis.

# CHAPTER 2. BACKGROUND

In this chapter, we briefly describe some of the existing protocols for long range wireless communications (Section 2.1) followed by basics of full duplex operation, and techniques to address interference related problems (Section 2.2).

## 2.1 Long Range Baseband Processor

Long range wireless communication protocols for IoT systems are designed to support communications between small sensor nodes and a network server with minimum power. LTE-M [21], LoRa [22], and Sigfox [23] are representative long range protocols. LTE-M [21] is a narrow band version of LTE protocol that is used for M2M communication. Narrow band allows it to achieve higher signal to noise ratio (SNR) while sacrificing data rate. LTE-M is based on OFDMA technology. LoRa is a new technology based on spread spectrum that targets low power long range (tens of kilometers) IoT terminals [22]. This technology is different from code division multiple access (CDMA) due to the use of relatively narrow band and chirp modulation scheme which is robust to frequency offset error. While the exact details of Sigfox are not known, we expect it to have characteristics similar to other narrow band protocols such as long idle mode and preamble based frame structure.

### 2.1.1 OFDMA-based IoT Terminal

Fig. 2.1 shows the receiver structure of an OFDMA-based IoT terminal. Frame detection consists of two operations: coarse-grain detection and fine-grain detection. Coarse-grain detection makes a decision on the existence of a predefined preamble pattern with minimal

computation cost. Fine-grain detection reconfirms the result of coarse-grain detection by using a computationally more expensive but more accurate algorithm, such as matched filtering. Thus, the fine-grain detection algorithm is conditionally called when a frame is detected by the coarse-grain algorithm.

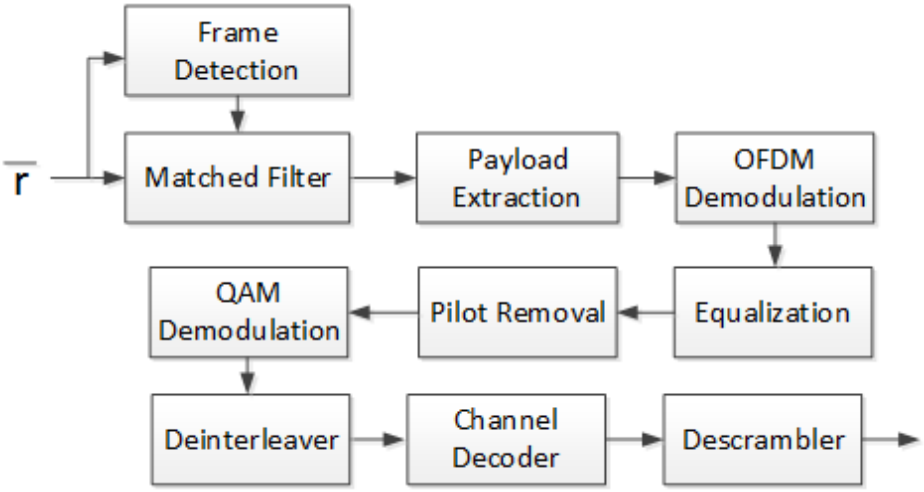


Figure 2.1 OFDMA-based IoT receiver

The computations in coarse-grain detection consist of auto-correlation of the received signal and its normalization with the signal energy. By exploiting the overlap in the computations of two adjacent autocorrelation values, $c_n$ can be implemented as $c_n = c_{n-1} + p_n - p_{n-L}$ where $p_n = r_n \cdot r_{n-N}^*$. This is referred to as the sliding window operation. Similarly the signal energy $e_n$ also can be implemented as

$e_n = e_{n-1} + q_n - q_{n-L}$, where $q_n = |r_n|^2$. Thus by using the sliding window approach, the computation overhead of each of these operations can be reduced from $L$ multiplications and $L - 1$ additions to one multiplication and two additions.

An OFDMA receiver generates a coarse-grain detection signal if the normalized auto-correlation value, referred to as $m_n$, is greater than a pre-defined threshold value, $Th$. This step is followed by fine-grain detection which is essentially matched filter computation. Matched filter is based on cross-correlation between reference preamble sequence and received signal, followed by selection of maximum value point. Although cross correlation requires many expensive multiplications, it is implemented on a small range.

The next step is payload extraction followed by OFDMA demodulation using FFT. The user data placed on sub-carriers is extracted and fed to the channel estimator/equalizer. For the estimator, least mean square (LMS) algorithm is selected because of its mid-range channel equalization performance without high computation load [24]. After pilot removal, N-QAM is used for demodulation. The data is then de-interleaved in order to achieve time diversity.

For forward error correction, convolutional code is used and decoding is done using the Viterbi algorithm. This algorithm can be represented by a set of vector operations, such as branch metric computation (BMC) and add compare select (ACS) operations [25]. Descrambler operation consists of bit-wise exclusive OR operations between the channels decoded data with a pseudo random sequence. Because bit wise operations can be performed independently, the descrambler operation can be computed using a bit vector operation.

### 2.1.2 Spread Spectrum-based IoT Terminal

Fig. 2.2 show the structure of spread spectrum-based IoT receiver. It consists of frame detection, searcher, Rake receiver, despreader, deinterleaver, channel decoder, and descrambler. The spread spectrum system uses a high frequency waveform to transmit information bits over the air. Its operation is similar to a CDMA system except that it uses chirp sequences. Using chirp sequence in modulation is advantageous for IoT terminals because it is robust to frequency offset error and does not require precise synchronization at base station. The searcher and Rake receiver are selected in order to get lower Bit Error Rate although they require additional computations.

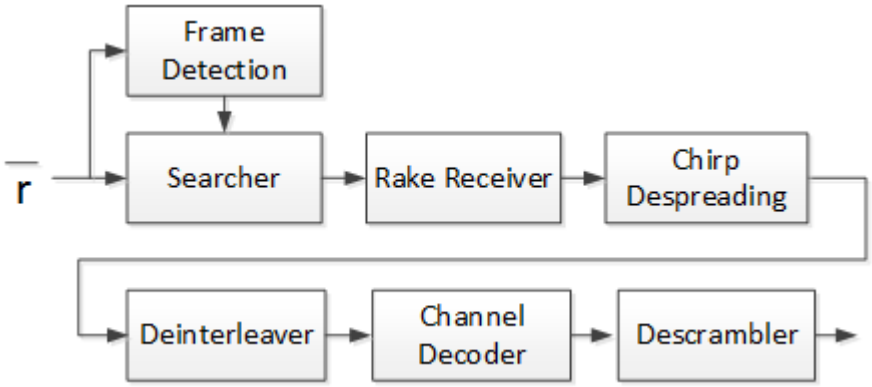


Figure 2.2 Spread Spectrum-based IoT receiver

After frame detection, the searcher estimates delay spread of received signal caused by multipath fading. The delay spread is estimated by continuous computation of correlations between received signal and chirp sequence. The high correlation peaks correspond to the location of the received signals. Rake receiver implements multiple demodulation paths with different delays. Each demodulation path (so called finger) performs matched filtering

and its computation pattern is similar to vector inner product. Chirp sequence with varied frequency is used for chirp despreading. Channel code used in the spread spectrum-based system is typically Reed-Solomon (RS) code which shows good correction performance for burst errors. The decoding consists of four blocks, namely, syndrome computation, Berlekamp-Messy algorithm, Chien Search algorithm, and Forney algorithm. Of these, syndrome computation and Chien search algorithm can be parallelized [26]. Other modules such as frame detection, deinterleaving, and descrambling are similar to those in the OFDMA-based IoT terminals, and are not described here. In Chapter 3, we present an analysis of the workload characteristics of the two protocols, followed by derivation of optimized algorithms and configuration of the parameters of the systems to minimize power consumption.

## 2.2 Full Duplex Systems

Traditionally, spectrum efficiency has been increased through advances in modulation, coding schemes, and Multiple Input Multiple Output (MIMO) technologies. The gains achieved by these methods have now saturated [27] and so researchers have turned to full duplex systems. In a full duplex system, receive and transmit signals occupy the same frequency band, thereby doubling the throughput.

Unfortunately a full-duplex network system suffers from all types of interference. When the interference is caused by the node's own transmissions, it is labeled self-interference. For short-range wireless systems, such as WiFi and small-cell systems, the path loss is not large, making self-interference reduction easier to achieve. When the interference to

reception is caused by a different node, it is labeled as inter-node interference. Inter-node interference can be within a cell (intra-cell) or across cells (inter-cell). In traditional half-duplex networks, intra-cell interference is either not an issue (e.g. in FDD networks) or a small part of an overall design challenge (e.g. in TDD networks like WiFi). With full-duplex transmissions, intra-cell interference is the dominant component, simply because uplink and downlink are simultaneously active.

### 2.2.1 Self-interference

A detailed spectral analysis of tones when transmitting and receiving was presented in [18]. Figure 2.3 borrowed from [18] shows that there are three main components. First is the linear component, caused by the reflection of the transmitted signal. Second is the non-linear component that mainly consists of harmonics. The third is transmitter noise, which is extremely high, about 50dBm, and caused by radio transmitter such as power amplifier. Local oscillators can also generate additional phase noise.

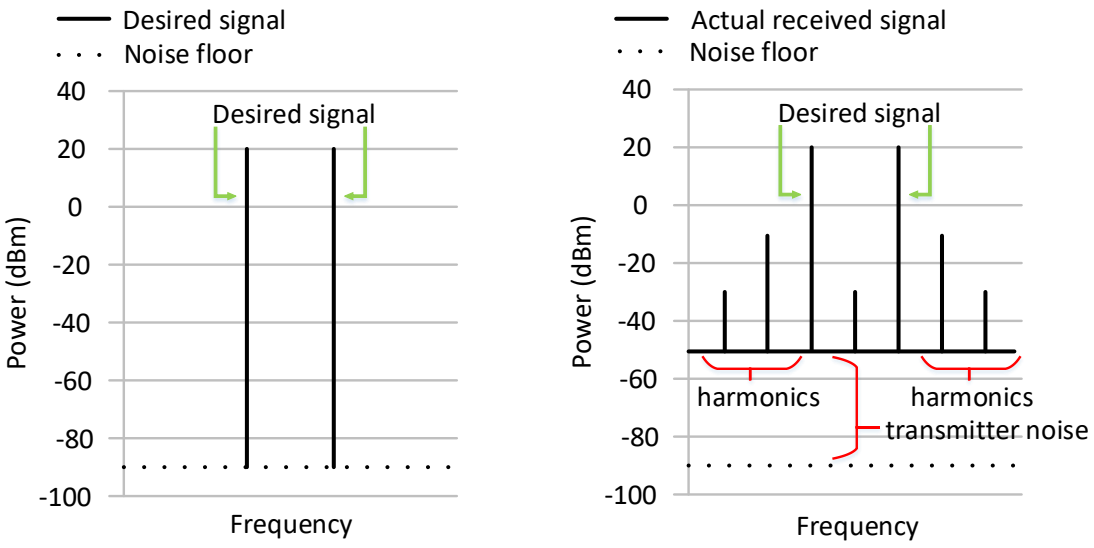


Figure 2.3 Signal spectrum of transmitter side and receiver side [18]

### 2.2.2 Requirements for Self-Interference Cancellation

In the experiments presented in [18], OFDM-wideband signals are used to quantify the power levels of different distortions. Using 80MHz bandwidth Wi-Fi radio, [18] showed that the noise floor in receiver side is -90dBm. First, for the main signal, 110dB of linear self-interference cancellation is required to achieve the -90dBm receiver noise floor.

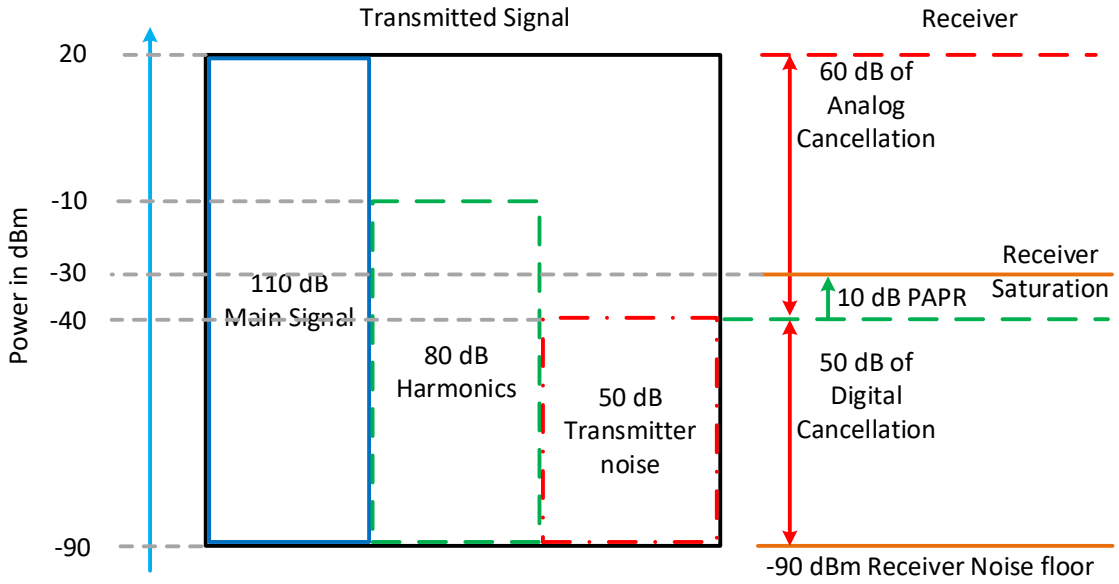


Figure 2.4 Power level requirements for full duplex system [18]

Second, since there are 80 dB harmonics above the noise floor, the full duplex technique has to provide at least 80dB non-linear self-interference cancellation. Third, transmitter noise is 50dB higher than the noise floor, which has to be addressed by analog noise cancellation. In summary, any full duplex technique should provide 110dB linear cancellation, 80dB non-linear cancellation, 60dB of analog cancellation and 50dB for digital cancellation. In this thesis, we focus on digital cancellation.

### 2.2.3 Current Full Duplex Systems

There are three well-known architectures for full duplex systems. The Stanford architecture [18], the Rice architecture [20] and the Tampere University of Technology (TUT) architecture [19].

A. The Stanford architecture

The Stanford architecture uses both analog and digital cancellation methods to suppress self-interference with analog cancellation playing a dominant role. There is a circulator connected to the antenna, which is a three port device to provide limited isolation between transmitted and received signals. It transforms a copy of the analog RF signal using analog components to cancel the self-interference signal at receiver side. It then uses MLSE to cancel the remaining noise. The Stanford architecture has good performance, but its analog circuitry is expensive. For MIMO systems, its performance and cost is likely to be quite high [28].

B. The Rice architecture

Figure 2.5 describes the full-duplex OFDM transceiver from Rice [20]. At the transmitter side, the base-band signal is modulated using an OFDM modulator, then up-converted to the carrier frequency $f_c$ and then amplified using a power amplifier. The oscillator at the transmitter side is assumed to have a random phase error represented by $\varphi^t(t)$. At the receiver side, the amplitude of the received signal is properly adjusted using a low noise amplifier (LNA). The signal is then down-converted from the carrier frequency to the base-

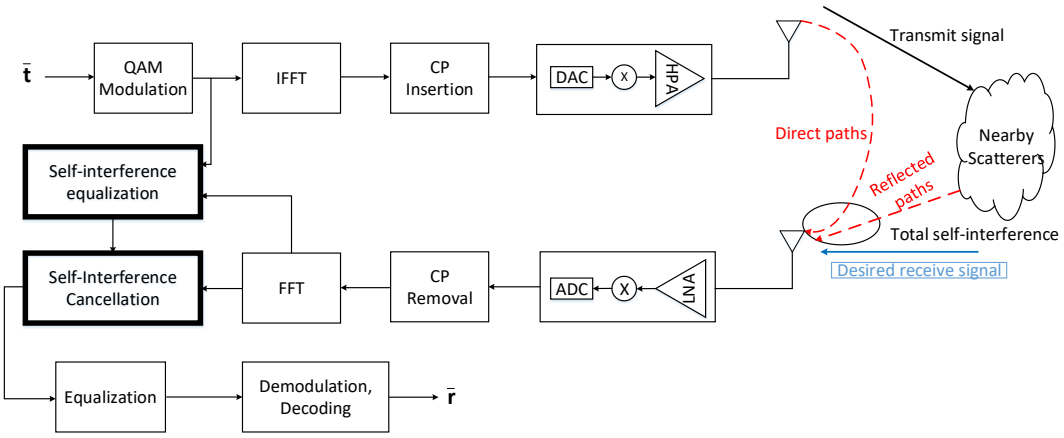


Figure 2.5 Rice architecture for full duplex system [20]

band. The down-conversion mixer is assumed to have a random phase error represented by $\varphi^{r}(t)$. The base-band signal is then quantized and converted to the frequency domain using Fourier transform. The Rice architecture, uses Least Square (LS) estimator to cancel self-interference. This architecture is able to suppress about 48dB self-interference. So its performance is not as good as the Stanford architecture.

C. TUT architecture

The TUT architecture uses a combination of techniques to reduce self-interference by 100dB [19]. It describes a novel antenna design to provide about 70dB cancellation. In addition, it uses multitap analog cancellation and LMS equalization algorithm for digital cancellation.

# CHAPTER 3. LONG RANGE IOT DIGITAL BASEBAND PROCESSOR

We propose a new baseband architecture for Internet of Things (IoT) terminals that support long range communications such as those based on Orthogonal Frequency-Division Multiple Access (OFDMA) and spread spectrum technologies. We analyze the workload profiles of both systems (Section 3.1). We introduce our baseband processor that optimized for frame detection in Section 3.2 and find that the frame detection unit has by far the highest computational load. We elaborate on the frame detection algorithm in Section 3.3 and derive energy-optimal operation settings for the frame detection unit for different channel conditions in Section 3.4.

## 3.1 Block Level Workload Characteristics

We analyzed the workload characteristics of major computation kernels of OFDMA-based (Figure 2.1) and spread spectrum-based IoT terminals (Figure 2.2) described in Chapter 2. Table 3.1 summarizes the computational characteristics of the two types of terminals. We see that while frame detection and deinterleaver are scalar, most of the other kernels are suitable for vector processing. The choice of the architecture--scalar vs vector--would depend on whether the system spends most of its time on the scalar kernels or on the vector kernels.

Table 3.1 Computational Characteristics of IoT Terminals

| Algorithm | Vector | Scalar | Short Vect. |
|---|---|---|---|
| Frame detection | | √ | |
| Matched filter | √ | | |
| FFT | √ | | |
| Channel estimation | √ | | |
| Equalization | √ | | |
| Demodulation | | | √ |
| Deinterleaver | | √ | |
| Viterbi decoder | √ | √ | |
| Descrambler | √ | | |
| Searcher | √ | | |
| Rake receiver | √ | | |
| RS decoder | √ | √ | √ |

Tables 3.2 and 3.3 present the workload profiles of OFDMA and spread spectrum-based IoT terminals respectively. To obtain the profiles, we built MATLAB models of the two IoT terminals and executed them on an X86 machine. We measured the cycle count of each algorithm, and computed the percentile contributions. In addition, to see the impact of long idle periods, we changed the ratio of operation time between idle state and active state from 10:1 to 100:1, though the ratio of idle to active state, in reality, is much longer.

Table 3.2 Workload Profile of OFDMA Terminal

| Block | Idle: Active=10:1 | Idle: Active=100:1 |
|---|---|---|
| Frame detection | 97.4763% | 99.1896% |
| Matched filter | 0.5408% | 0.0603% |
| Payload extraction | 0.4507% | 0.1206% |
| OFDM demodulation | 0.0451% | 0.0100% |
| Equalization | 0.1352% | 0.0502% |
| Pilot removal | 0.0901% | 0.0134% |
| QAM demodulation | 0.2253% | 0.0167% |
| Deinterleaving | 0.0451% | 0.0033% |
| Channel decoding | 0.9013% | 0.5291% |
| Descrambling | 0.0901% | 0.0067% |

In the OFDMA-based terminals, for ratio 10:1, frame detection is the dominant workload (>97%) followed by channel decoding. Consider a scenario where 1 out of 10 frames contains information. If each frame has 4,000 symbols, then the frame detection unit operates on 10×4,000 symbols. In contrast, the matched filter only operates on 100 symbols. After changing the idle to active ratio to 100:1, the frame detection block accounts for 99% of the workload.

Table 3.3 Workload Profile of Spread Spectrum Terminal

| Block | Idle: Active=10:1 | Idle: Active=100:1 |
|---|---|---|
| Frame detection | 99.4264% | 99.8456% |
| Searcher | 0.1941% | 0.1261% |
| Rake receiver | 0.0131% | 0.0004% |
| Chirp despreading | 0.0174% | 0.0008% |
| Deinterleaving | 0.0044% | 0.0002% |
| RS decoding | 0.3402% | 0.0264% |
| Descrambling | 0.0044% | 0.0004% |

In the spread spectrum-based terminals, frame detection is more than 99% of the workload even when the ratio is 10:1. This is because in spectrum spread systems, the frame size is much larger and so there are more calculations per frame. Note that in the active mode, our workload profile for OFDMA protocol is almost identical compared to that in [9].

From Tables 3.2 and 3.3, it is clear that for both systems frame detection is, by far, the most dominant workload. Since it can be implemented efficiently using low cost sliding window algorithm, the baseband processor has to be optimized for scalar processing.

## 3.2 Processor Architecture

Figure 3.1 shows the architecture of the proposed baseband processor designed for OFDMA and spread spectrum-based IoT terminals [11]. It is essentially a 32bit scalar processor which consists of arithmetic and logic unit, an accumulator, register file of size

32bit×16, and two data memories each of size 8Kbytes. Each data memory has two read ports and one write port and can be programmed to read two entries and write one entry in one cycle. Each read or write port has a dedicated address generation unit (AGU) to access memory with minimal address calculation overhead. The ALU can operate on data from

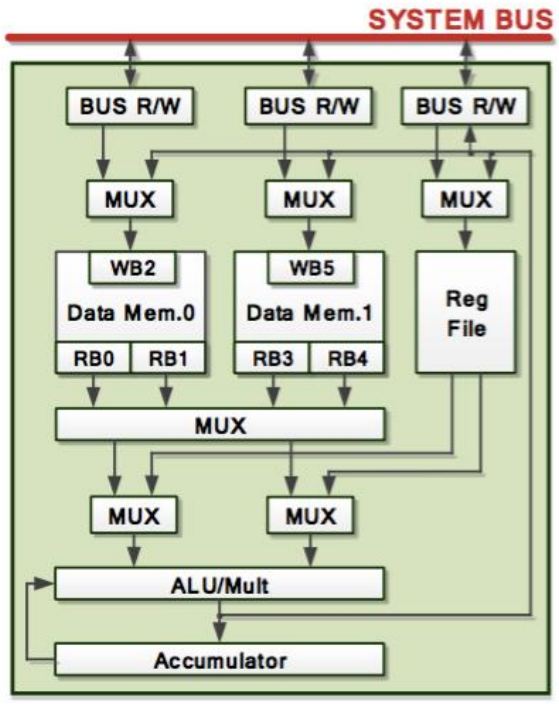


Figure 3.1 Architecture of processor for long range IoT terminal

the register files or data memories. Alternately, the data memories can load data from the ALU and the RF unit. The control path consists of instruction memory and instruction decoder which are not shown in this figure for simplicity.

The choice of scalar processor was derived from the workload analysis results which showed that even when idle to active period ratio is 10:1, frame detection (which is a sequential algorithm) accounts for 97%-99% of the workload. Apart from frame detection,

the other baseband signal processing algorithms can be represented by vector inner products and implemented by a SIMD style architecture. However, in reality, idle period is much longer than active period and thus the power saving and throughput enhancement that we can expect from using SIMD datapath, is almost negligible.

### 3.3 Frame Detection Algorithm

In Section 3.1, we observed that the frame detection block dominates the total workload of IoT terminals in both the spread spectrum-based and OFDMA-based IoT terminals. Here we discuss in detail the frame detection algorithm.

Frame detection estimates whether a frame was transmitted from a basestation or not. It is implemented as a binary hypothesis test. Let $m_n$ be the auto-correlation value $c_n$ of received data, normalized by its energy as shown in Eqn. (1).

$$m_n = \frac{|c_n|^2}{e_{n,0} \cdot e_{n,1}} \quad (1)$$

Here $c_n$ is implemented as the inner product of two vectors, $c_n = \sum_{i=0}^{L-1} r_{n-i} r_{n-i-N}^*$ where $r_n$ represents the received signal, $L$ is the correlation window length, and $N$ is the spacing between two correlation windows. Signal energy $e_{n,0}$ is computed as $\sum_{i=0}^{L-1} |r_{n-i}|^2$ and $e_{n,1}$ is computed as $\sum_{i=0}^{L-1} |r_{n-i-L}|^2$. Let $H_0$ be the case for frame not detected and $H_1$ be the case for frame detected. Then the actual test is to check whether the normalized correlation value mn is bigger than a predefined threshold $Th$.

The performance of frame detection is based on two probabilities: probability of detection, $P_D$, and probability of false alarm, $P_{FA}$. $P_D$ is the probability of detecting a packet that has

been transmitted from a basestation, and $P_{FA}$ is the probability that the binary test incorrectly decides the presence of frames, when actually no frames were transmitted. Let $F_{tx}$ and $F_{ntx}$ represent the event that a basestation transmits a frame or not. Then $P_D$ and $P_{FA}$ can be defined in terms of conditional probabilities as shown below.

$$P_D = P(m_n \geq Th | F_{tx}) \tag{2}$$

$$P_{FA} = P(m_n \geq Th | F_{ntx}) \tag{3}$$

Although high $P_D$ is desirable, it requires a complex algorithm and hence higher energy consumption. High $P_{FA}$ results in unnecessary activation of full receiver chain, and so increases idle mode energy consumption. Thus $P_{FA}$ should be as small as possible for energy-efficient operation, as will be illustrated in next subsection.

**Coarse and Fine-Grain Detection:** As shown in Figure 2.1 and Figure 2.2, the frame detection block consists of two parts, coarse-grain detection and fine-grain detection. Because fine-grain detection algorithm is conditionally called when a frame is detected by the coarse-grain detection algorithm, the impact of coarse-grain algorithm on energy consumption is more substantial. The coarse-grain detection algorithm can be implemented using sliding window. Let $p_n = r_n \cdot r_{n-N}^*$ be the intermediate parameter for sliding window. Then the auto-corelation value $c_n = c_{n-1} + p_n - p_{n-L}$. For more details, please refer to Section 2.1.

Algorithm 1 describes the sliding window algorithm for coarse-grain detection. Here $Th$ is the threshold value, $r$ is the receiver sequence, $n$ is index of signal, $N$ is correlation length,

$L$ is window size. It minimizes the number of computations by reusing previous multiplication results stored in arrays *pc* and *pe*.

**Algorithm 1** Sliding Window Algorithm

```
1:  procedure COARSE_GRAIN_DETECTION(Th, r, n, N, L)
2:      pc[n] = r[n] * complex_conj(r[n-N])
3:      pe[n] = r[n] * complex_conj(r[n])
4:      c[n] = c[n-1] + pc[n] - pc[n-L+1]
5:      e1[n] = e1[n-1] + pe[n] - pe[n-L+1]
6:      e2[n] = e2[n-1] + pe[n-N+1] - pe[n-N-L+1]
7:      m[n] = c[n] * c[n] / (e1[n] * e2[n])
8:      if(Th < m[n] ) return FRAME_DET
9:      else return FRAME_NO_DET
```

The implementation of the baseline coarse-grain detection algorithm, Algorithm 1, is shown in Eqn. (1). In order to reduce the number of computations in computing the energy terms used for normalization, we propose Algorithm 2 which implements $m_n = |c_n|^2 / e_{n,0}^2$. Thus, while Algorithm 1 uses energy terms corresponding to current correlation window, $e_{n,0}$, and previous correlation window, $e_{n,1}$ for normalization, Algorithm 2 just uses signal energy corresponding to only the current correlation window for normalization. In the pseudo code shown above, Algorithm 2 does not require address computation, addition, and subtraction in line 6 and simplifies computations in line 7 by converting complex multiplication to two real multiplications. In Section 3.4.3, we will compare the energy cost of these two algorithms.

## 3.4 Finding Energy-optimal Parameters

In this section, we find the most energy-optimal operation parameters for frame detection. We assume that the frame detection algorithm has been implemented on the proposed IoT

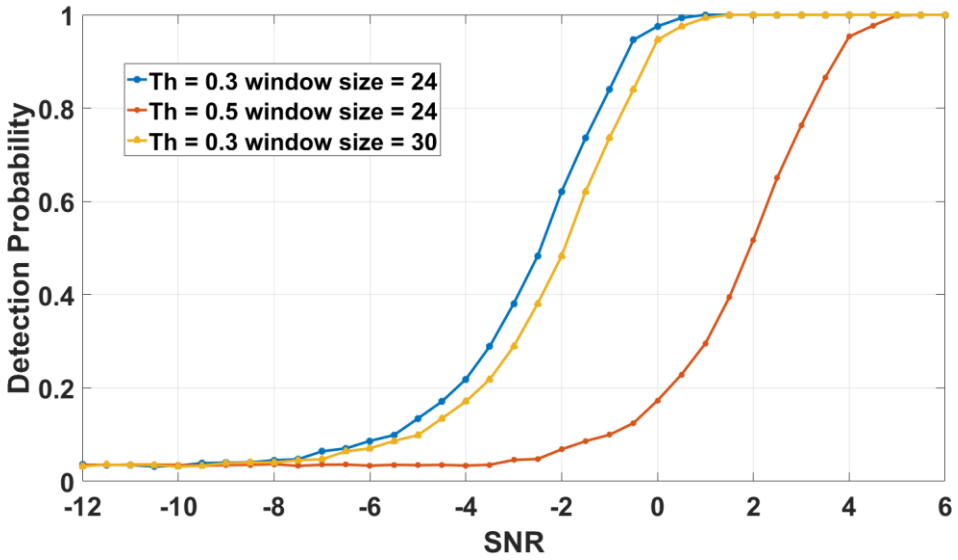


Figure 3.2 Detection probability as a function of SNR for Rayleigh fading channels with different parameter settings

processor. We first present the channel model in Section 3.4.1. Then we derive an energy model of the baseband processor for frame detection in Section 3.4.2. This is followed by design space exploration in Section 3.4.3.

### 3.4.1 Channel Model

We consider Rayleigh fading as our channel model because there is no line of sight in long range communication. Our Rayleigh fading channel model has 10 Hz Doppler shift and channel length 16 with frequency offset 2 KHz. We add AWGN noise to Rayleigh fading channel.

Figure 3.2 describes how detection probability $P_D$ varies with different SNR, where SNR is the ratio of the average signal power to average noise power. The blue curve (Figure 3.2) corresponds to the case when threshold = 0.3 and correlation window size of 24. For the

same window size (W = 24), if the threshold increases to 0.5, the curve moves to the right implying that the channel SNR has to be very good to achieve the same detection probability. If the threshold value stays the same (at 0.3), an increase in the window size shifts the curve to the right only slightly. Thus detection probability is very sensitive to threshold value. We found that this trend is similar for other fading channels and so here we only consider the Rayleigh fading channel model.

### 3.4.2 Energy Model

The total energy consumption of the baseband processor in the idle mode is the sum of its energy when it computes frame detection (and is on) and the energy when it is off. Let $P_{off}$ and $P_{on}$ represent the probability of the processor being in off-state and on-state, respectively, and let $E_{off}$ and $E_{on}$ be the corresponding off-state and on-state energy. Then the total idle mode energy $E_T$ is given as follows

$$E_T = P_{off} \cdot E_{off} + P_{on} \cdot E_{on} \tag{4}$$

A more precise energy consumption model is given by

$$E_{on} = E_c + P\left(m_{n,c} > Th_c\right) \cdot E_f + P\left(\left(m_{n,c} > Th_c\right) \cap \left(m_{n,f} > Th_f\right)\right) \cdot E_{rest} \tag{5}$$

In the above equation, $E_c$ and $E_f$ represent energy consumption for coarse-grain detection and fine-grain detection respectively, $m_{n,c}$ and $Th_c$ are the normalized correlation value and threshold value for coarse-grain detection, and $m_{n,f}$ and $Th_f$ are the normalized correlation value and threshold value for fine-grain detection. $E_{rest}$ is the energy consumption of blocks that are activated in the receiver chain after fine-grain detection. We keep $Th_f$ constant at 0.4 and vary $Th_c$, to derive the energy-efficient configuration. We refer to $Th_c$ as $Th$ in the rest of this section.

We implemented the coarse-grain and fine-grain frame detection algorithms and evaluated their energy consumption by accumulating the energy consumption of all blocks in the digital baseband processor (Section 3.2) that were activated during the execution of these algorithms. We found that the energy cost of coarse-grain detection, $E_c$, is 41.73 nJ for Algorithm 1 and 38.97 nJ for Algorithm 2. The energy cost of the fine-grain detection algorithm, $E_f$, is 27.4 × 16 × L nJ, where L is the correlation length. Thus, the energy cost of fine-grain detection is higher than that of coarse-grain detection, as expected. However, fine-grain detection is activated only when the normalized correlation value due to coarse-grain frame detection is greater than a threshold value.

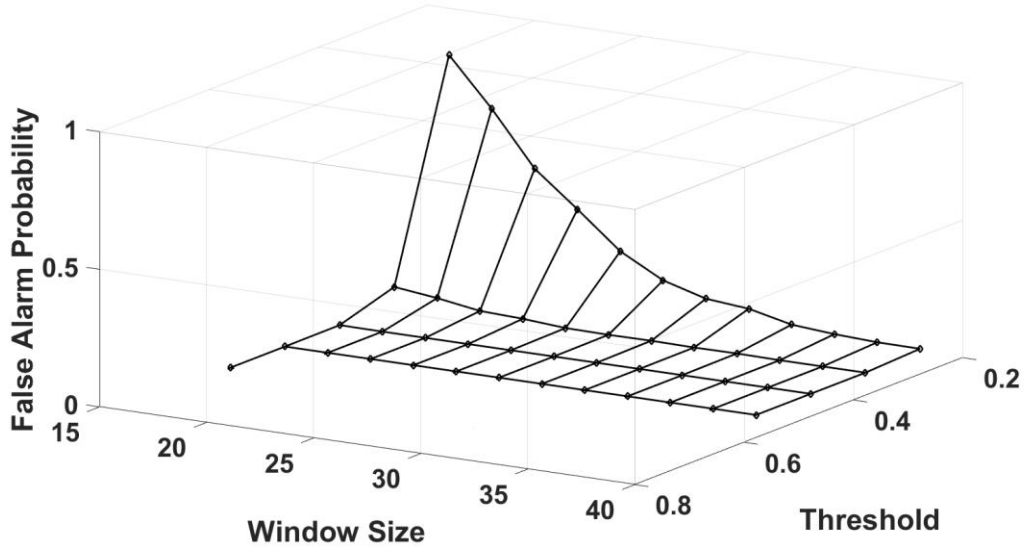


Figure 3.3 False alarm probability using Algorithm 1 when $P_D > 0.9$ and SNR = 5dB

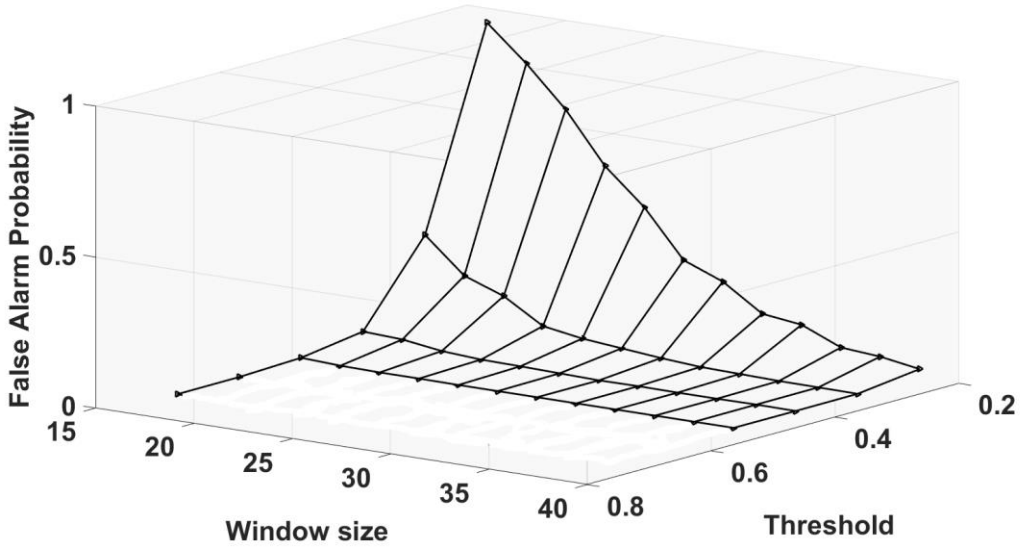


Figure 3.4 False alarm probability using Algorithm 2 when $P_D > 0.9$ and SNR = 5dB

### 3.4.3 Design Space Exploration

We study the effect of the following three operation parameters on the processor energy consumption: i) type of coarse grain detection algorithm (Algorithm 1 or Algorithm 2), ii) coarse-grain detection threshold *Th* and iii) correlation window size.

The frame detection probability $P_D$ and the false alarm probability $P_{FA}$, both affect the on-state energy, $E_{on}$. While $P_D$ can be improved by using a more complex algorithm, $P_D$ value gets saturated and increasing the complexity of the algorithm does not help. At the network level, low $P_D$ increases average packet transmission time but higher $P_D$ increases $E_{on}$. The tolerable packet delay range is determined by the network service provider. This typically corresponds to $P_D > 0.9$, however, when channel condition is bad, we may lower the $P_D$ requirement to 0.8 temporarily. So in this paper, we performed analysis for $P_D > 0.9$ in

good channel condition (SNR > 0dB) and $P_D$ > 0.8 for bad channel condition (SNR < -5dB).

In the energy consumption equation, eqn. (5), the first term $E_c$ is affected by detection probability and the second term is a function of the false alarm probability. When the system satisfies the detection probability requirement, the first term $E_c$ just depends on the algorithm type (Algorithm 1 or Algorithm 2). If we fix the algorithm type, then the energy consumption depends on second term. Since $E_f$ is constant, the energy consumption depends on the false alarm probability.

Figures 3.3 and 3.4 plot the false alarm probability as a function of threshold value and window size for Algorithm 1 and 2, respectively. From Figure 3.3, we see that for Algorithm 1, false alarm probability is low when threshold value is between 0.3 and 0.5

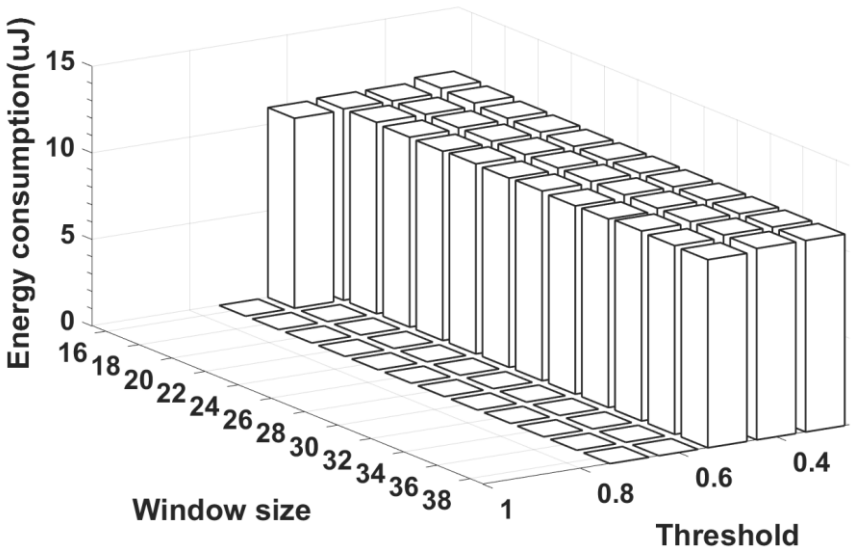


Figure 3.5 Energy consumption of Algorithm 1 when $P_D$ > 0.9 and SNR = 5dB

with window size from 20 to 38. For this set of settings, the false alarm probability is less than 0.021. Similarly from Figure 3.4, we see that for Algorithm 2, the low false alarm probability points correspond to threshold values ranging from 0.3 to 0.5 and window size ranging from 26 to 38. So in the rest of the section, we restrict the threshold value and window size range in order to guarantee that the false alarm probability is low.

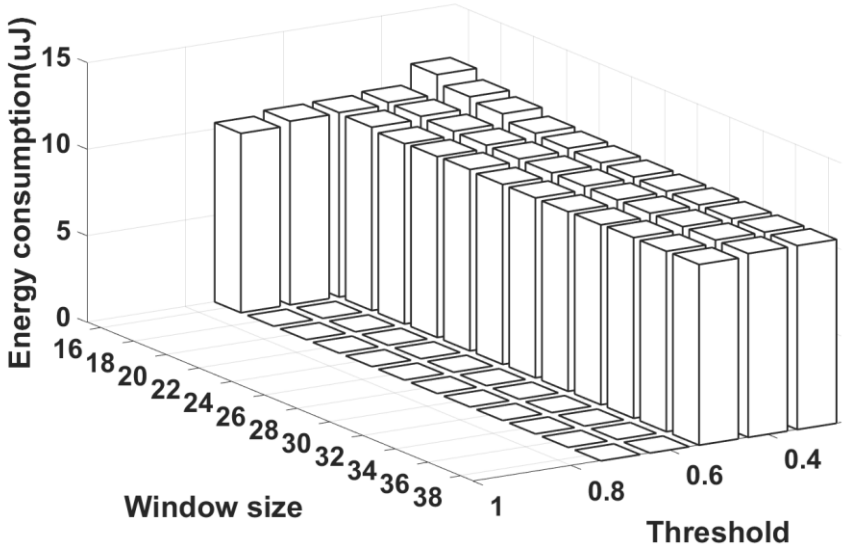


Figure 3.6 Energy consumption of Algorithm 2 when $P_D > 0.9$ and SNR = 5dB

Fig. 3.5 and Fig. 3.6 show energy consumption for different algorithms, window sizes, and threshold values for good channel conditions (SNR = 5dB). For the same window size, as threshold becomes higher, the energy consumption gets lower. However when the window size is larger than 16, and the threshold is larger than 0.5, the detection probability is less than 0.9 and is not shown in this figure. The lowest energy configuration of Algorithm 1 (Figure 3.5) corresponds to window size 38 and threshold 0.5 and the lowest energy

configuration of Algorithm 2 (Figure 3.6) corresponds to window size 32 and threshold 0.5.

**Effect of choice of algorithm:**

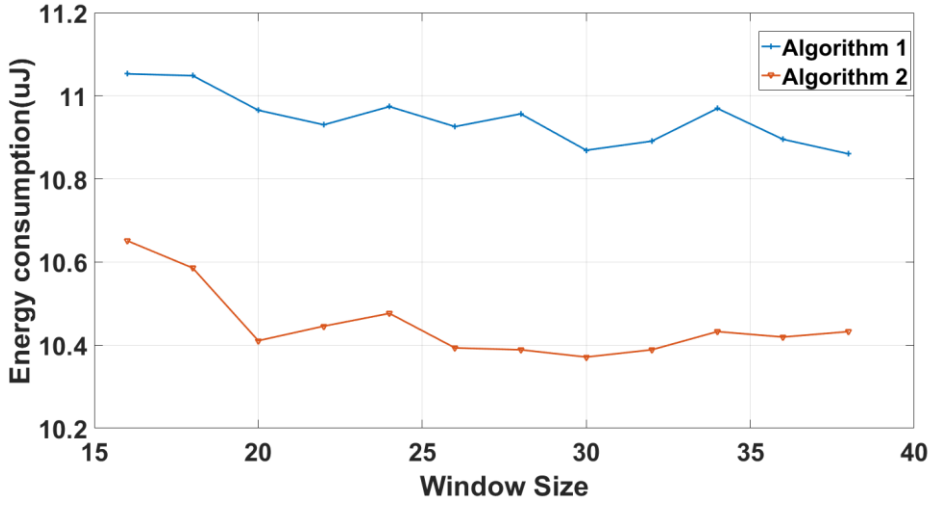


Figure 3.7 Energy consumption comparison of both algorithms when $P_D > 0.9$, SNR = 5dB and Threshold = 0.5

We investigate the energy performance of Algorithms 1 and 2 for good and bad channel conditions. As window size gets larger, the energy consumption reduces. This is because as window size increases, the false alarm probability becomes smaller in both algorithms. For the good channel case, shown in Figure 3.7, Algorithm 2 has lower energy than Algorithm 1. In contrast, for a very bad channel (SNR < -9dB), shown in Figure 3.8, we find that Algorithm 1 has lower energy than Algorithm 2. Thus when the channel is very poor, Algorithm 1 can be used. Once the base station reacts to the poor channel condition by increasing the signal strength, the receiver can switch to Algorithm 2.

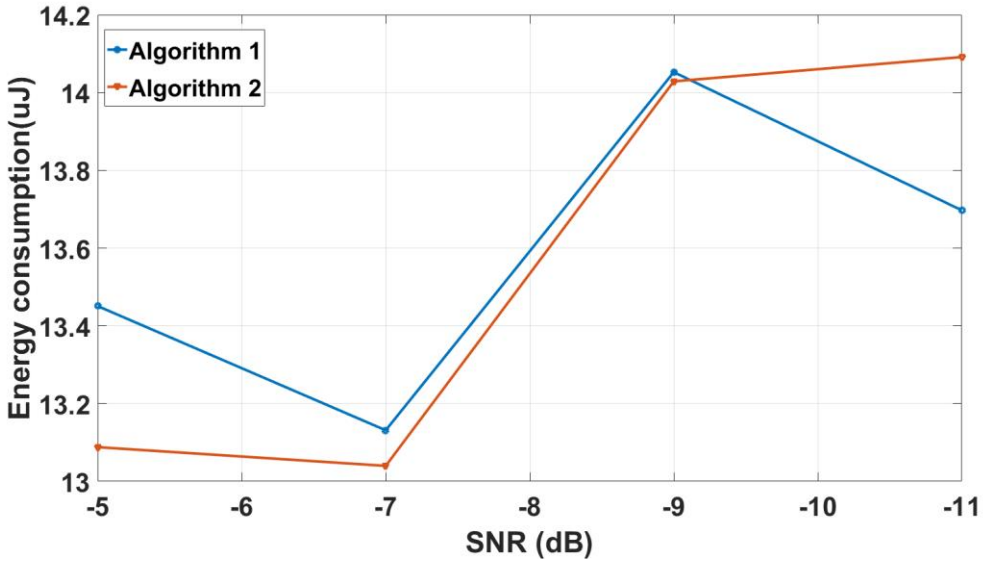


Figure 3.8 Energy consumption comparison of both algorithms when $P_D > 0.8$

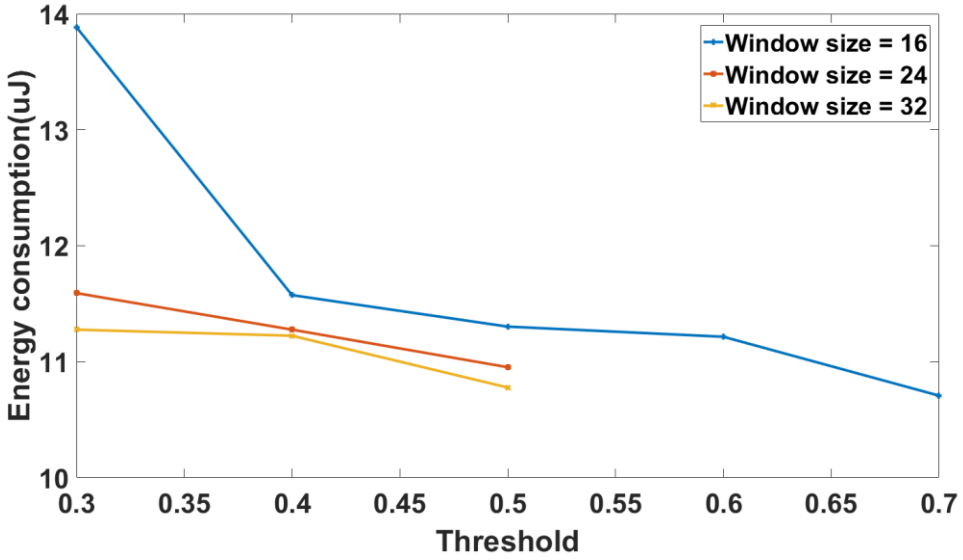


Figure 3.9 Energy consumption of Algorithm 2 when $P_D > 0.9$ and SNR = 5dB

We focus on Algorithm 2. Since the two factors (window size and threshold) contribute to the energy consumption, next, we provide a detailed analysis of the effect of these two parameters for Algorithm 2.

**Effect of threshold values:**

For the good channel condition case (SNR = 5dB), as threshold gets larger, the energy consumption reduces. This is shown in Figure 3.9 for Algorithm 2. When threshold changes from 0.3 to 0.4, the energy consumption decreases for small window size (W = 16). The reduction is not as dramatic for large window sizes. However when threshold value is larger than 0.4, for all window sizes, the energy consumption barely changes.

Recall that since threshold is the value that is compared with $m_n$ (see Eqn. (1)), it affects both detection probability and false alarm probability. For window sizes 32 or lower, the reason why energy consumption decreases sharply at first is that low threshold may cause a high false alarm detection. Since average noise level is near 0.2, if threshold is set close to 0.2, false alarm probability is very high. Every time there is a false alarm, the fine-grain detection block is activated. Since fine-grain detection is based on matched filter which is more complex, the energy overhead is high.

The false alarm probability is quite low when threshold is high. The false alarm probability is also low when the window size is large, and so the value of threshold has little effect on further reducing false alarm probability. This is why for large window size, the threshold value has little effect on energy consumption.

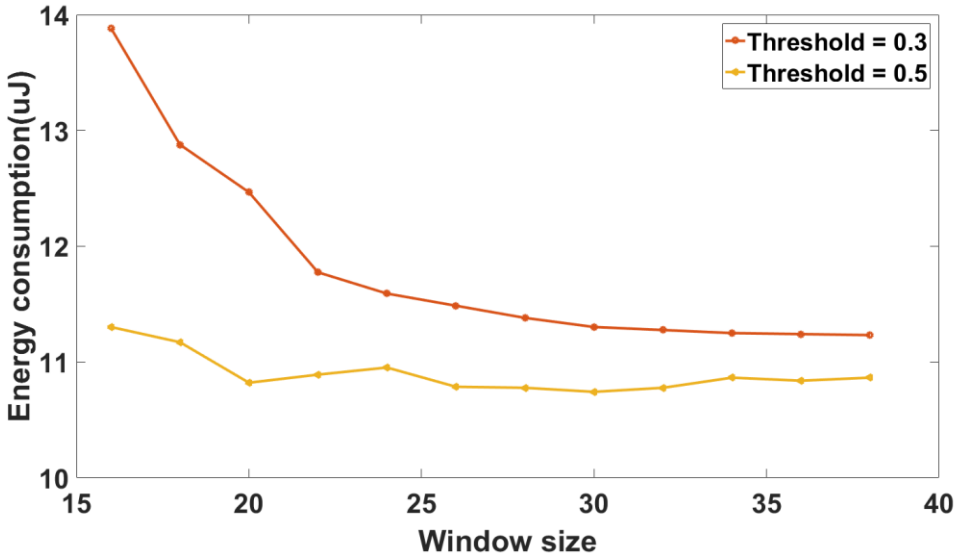


Figure 3.10 Energy consumption of Algorithm 2 when $P_D > 0.9$ and SNR = 5dB

Threshold value also affects the detection probability and its effect is evident when the window size is large. This is why when the threshold is higher than 0.5, only one case (window size = 16) can satisfy the detection probability requirement.

**Effect of window size:**

The effect of window size on energy consumption is illustrated in Figure 3.10. For the good channel condition shown in Figure 3.10, when window size increases, energy consumption decreases significantly at the beginning if threshold value is small. However when window size is larger than 36, the energy consumption is almost flat.

When window size gets larger, we use more samples to calculate the value $m_n$. Because the algorithm uses energy normalization, each time window size is increased, the value of signal energy term used for normalization gets bigger, but the correlation value does not

grow as fast. Thus the $m_n$ value becomes smaller with larger window size. While increasing window size can reduce false alarm probability, it can also affect the detection probability. For small threshold values, the reason why energy consumption decreases at first is because increasing window size decreases false alarm probability. As illustrated in previous subsection, false alarm probability affects energy consumption significantly. As window size increases, false alarm probability decreases to a point. After that there is no reduction in false alarm rate and the energy consumption does not change with large window sizes.

**Summary:**

We conclude that Algorithm 2 is the more energy-efficient choice for a wide range of channel conditions (up to SNR = -9dB) when $P_D > 0.8$ ( For low SNR, we need to lower detection probability requirement a little bit in order to receive the signal). For a certain detection probability, increasing window size and threshold can both reduce energy consumption, however, changing window size on the fly involves change in buffer size, addressing etc. In contrast, changing threshold value on the fly is trivial. So we propose to change threshold value based on the channel conditions for energy efficiency.

# CHAPTER 4. REDUCING SELF-INTERFERENCE THROUGH EQUALIZATION IN FULL DUPLEX SYSTEMS

In Chapter 2, we described how a full duplex system suffers from self-interference. We found that if the analog part can reduce 60dB noise, then the digital part should remove 50dB noise due to linear components and 20dB noise due to non-linear components. The linear noise components are due to the main transmitted signal which is received by itself and the reflected signal through the communication channel. After removing the linear components, the digital part still has to remove the non-linear components. It is not easy to model the non-linear noise, thus more complex equalization such as decision feedback may have to be used for this part. In this thesis, we focus on linear noise cancellation.

In this chapter, we first introduce the equalization algorithms in Sections 4.1 and 4.2. We present the channel models in Section 4.3 followed by performance results for different channel models in Section 4.4.

## 4.1 Algorithms for Equalization

### 4.1.1 Least Mean Square (LMS)

The LMS algorithm can be classified as an adaptive filter that mimics a desired filter by finding the filter coefficients that produce the least mean square of the error signal (difference between the desired and the actual signals) [29]. This equalizer uses stochastic gradient descent method.

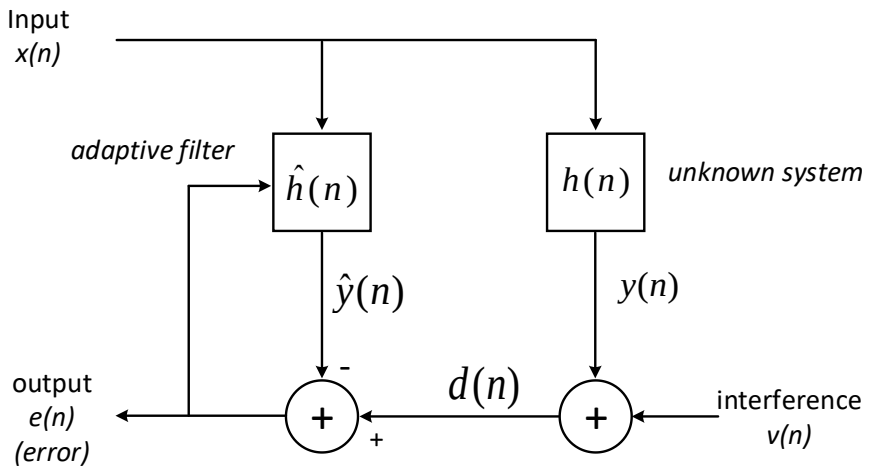


Figure 4.1 LMS block diagram

Let $\boldsymbol{h}(n)$ be the unknown system that we are trying to model, $\boldsymbol{x}(n)$ be the input with the number of the current input sample being $n$, and $\widehat{\boldsymbol{h}}(n)$ be the estimated filter. Both $\boldsymbol{h}(n)$ and $\widehat{\boldsymbol{h}}(n)$ are filters with $p$ taps. Define $\{.\}^T$ as matrix transpose and $\{.\}^*$ as conjugate operation. In each iteration, $\widehat{\boldsymbol{h}}(n)$ is updated to minimize the error. The algorithm is described below:

| Algorithm LMS | |
|---|---|
| Parameters: | μ = step size |
| Initialization: | $\widehat{\boldsymbol{h}}(\boldsymbol{0}) = zeros(p)$ |
| Computation: | for $\boldsymbol{n}$=0, 1, 2 … |
| | $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$ |
| | $e(n) = d(n) - \widehat{\boldsymbol{h}}^T(n)\boldsymbol{x}(n)$ |
| | $\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) + \mu e^*(n)\boldsymbol{x}(n)$ |

LMS has very low complexity and is easy to implement. Unfortunately, it has some stability problems.

### 4.1.2 Normalized Least Mean Square (NLMS)

The main disadvantage of LMS algorithm is that it is sensitive to the scaling of input $\boldsymbol{x}(n)$, which makes it hard to choose step size $\mu$ to guarantee the accuracy of the algorithm. Normalizing the power of the input signal $\boldsymbol{x}(n)$ can solve this problem [30]. The corresponding algorithm is referred to as NLMS. Define $\{.\}^H$ as Hermitian Transpose.

| Algorithm NLMS | |
|---|---|
| Parameters: | $p$ =filter order; μ = step size |
| Initialization: | $\widehat{\boldsymbol{h}}(\mathbf{0}) = zeros(p)$ |
| Computation: | for $\boldsymbol{n}$=0, 1, 2 … |
| | $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-p+1)]^T$ |
| | $e(n) = d(n) - \widehat{\boldsymbol{h}}^T(n)\boldsymbol{x}(n)$ |
| | $\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) + \frac{\mu e^*(n)\boldsymbol{x}(n)}{\boldsymbol{x}^H(n)\boldsymbol{x}(n)}$ |

Normalization of the input signal power introduces more matrix multiplication and division operations. But the algorithm is more stable compared to LMS.

### 4.1.3 Recursive Least Square (RLS)

The Recursive Least Squares (RLS) is an adaptive algorithm which recursively finds the coefficients that minimize a weighted linear least squares cost function [31]. This is in contrast to other algorithms such as LMS that aim to reduce the mean square error. In the

derivation of RLS, the input signals are considered deterministic, while for LMS and similar algorithms they are considered stochastic. Compared to most of its competitors, RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity.

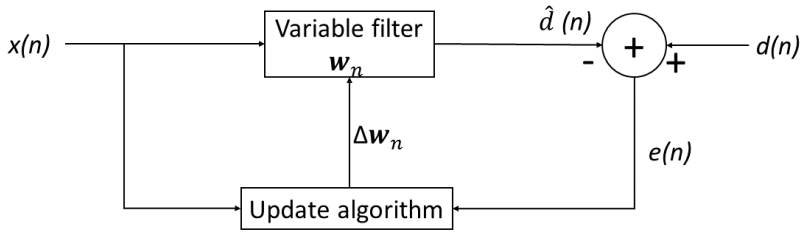


Figure 4.2 RLS block diagram

The idea behind RLS filters is to minimize a cost function *C* by appropriately selecting the filter coefficients $\mathbf{w}_n$ in each iteration n. The error signal *e(n)* and desired signal *d(n)* are defined in the negative feedback path shown in Figure 4.2. The error depends on the difference of estimated signal $\hat{d}(n)$ and desired signal $d(n)$ :

$$e(n) = d(n) - \hat{d}(n)$$

The weighted function *C* is a function of $e(n)$ :

$$C(\boldsymbol{w}_n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$$

Here $0 < \lambda \leq 1$ is the "forgetting factor" which gives exponentially less weight to previous error samples. The algorithm is shown below:

Algorithm RLS

Parameters: $\boldsymbol{p}$ = filter order; $\boldsymbol{\lambda}$ = forgetting factor; $\boldsymbol{\delta}$ = value to initialize **P(0)**

Initialization: $\boldsymbol{w}(n) = 0$ ,

$x(k) = 0, k = -p, \ldots, -1.$

$d(k) = 0, k = -p, \ldots, -1.$

$\boldsymbol{P}(0) = \delta^{-1} I$ where $I$ is the identity matrix of rank $\boldsymbol{p} + 1$

Computation: For $n = 1,2, \ldots$

$$\boldsymbol{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-p) \end{bmatrix}$$

$$\alpha(n) = d(n) - \boldsymbol{x^T}(n)\boldsymbol{w}(n-1)$$

$$\boldsymbol{g}(n) = \boldsymbol{P}(n-1)\boldsymbol{x}(n)\{\lambda + \boldsymbol{x^T}(n)\boldsymbol{P}(n-1)\boldsymbol{x}(\boldsymbol{n})\}^{-1}$$

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \boldsymbol{g}(n)\boldsymbol{x^T}(n)\lambda^{-1}\boldsymbol{P}(n-1)$$

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \alpha(n)\boldsymbol{g}(n)$$

RLS has very fast convergence but it is significantly more complex compared to LMS.

## 4.2 Decision Feedback Equalizer

A decision feedback equalizer (DFE) uses feedback of detected symbols to produce an estimate of the channel output. It cancels inter-symbol interference (ISI) while minimizing noise enhancement caused by inverting the channel frequency response [32]. Note that noise enhancement is a typical problem of the linear equalizers described earlier.

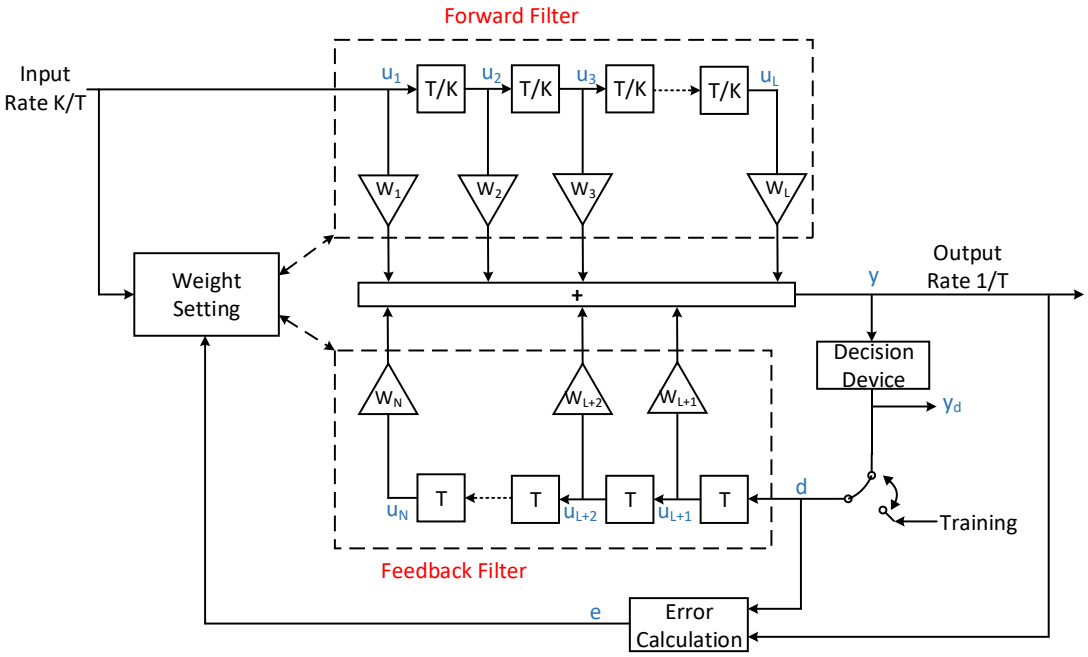


Figure 4.3 Decision Feedback Block Diagram

The block diagram of decision feedback equalizer is shown in Figure 4.3. The upper dashed box in the diagram is the forward filter, which is a delay line that outputs the weighted sum value of the delay signal. The order of the forward filter is L. The delay is given by T/K where T is symbol period, and K is an integer. Thus this equalizer receives K input samples before it produces one output sample and updates the weights. In our system, we set K = 1.

The lower dashed box is the feedback filter which contains a tapped delay line whose inputs are the decisions made on the equalized signal. The order of feedback filter is N. The weight setting block is an existing equalizer such as LMS, RLS, or NLMS. The error calculation block calculates error $e = d - y$ and uses $e$ to update the settings in the equalizer.

There are two operation modes for the equalizer: training mode and the other is decision-directed mode. In training mode, the reference is the known transmitted sequence; in decision-directed mode, the reference signal is signal generated by decision device, which is denoted as $y_d$ in the diagram. In a typical application, the equalizer begins in training mode to gather information about the channel, and then switches to decision directed mode.

### 4.2.1 Least Mean Square Decision Feedback Equalizer

The LMS Decision Feedback Equalizer is a decision feedback equalizer that uses the LMS algorithm to equalize a linearly modulated baseband signal through a dispersive channel. Basically, the weight setting block (see Figure 4.3) uses the LMS algorithm to update the weights, once per symbol.

### 4.2.2 Normalized Least Mean Square Decision Feedback Equalizer

The Normalized LMS Decision Feedback Equalizer is a decision feedback equalizer that uses the NLMS algorithm for equalization. The weight setting block now uses the NLMS algorithm (described in Section 4.1.2) to update the weights, once per symbol.

The advantage of NLMS feedback decision feedback equalizer compared to LMS decision feedback equalizer is that it can solve the scaling input problem. The LMS algorithm is sensitive to scaling of the input, and thus by using normalized operation in NLMS, the performance is stable. Compared to NLMS linear equalizer (Section 4.1.2), it has a feedback loop, which makes the performance more accurate, though it takes more calculations.

### 4.2.3 RLS Decision Feedback Equalizer

The RLS Decision Feedback Equalizer is a decision feedback equalizer that uses the RLS algorithm for equalization. The weight setting block uses the RLS algorithm (described in Section 4.1.3) to update the weights, once per symbol.

### 4.2.4 Summary

Table 4.1 Comparison of Equalization Algorithms

| Algorithm | Complexity | Stability |
|---|---|---|
| LMS | Low | Less stable |
| NLMS | Low | Stable |
| RLS | Medium | Stable |
| LMS Decision Feedback | High | Highly stable |
| NLMS Decision Feedback | High | Highly stable |
| RLS Decision Feedback | Very high | Highly stable |

Table 4.1 compares the complexity and stability of the different equalization algorithms. The first three algorithms (LMS, NLMS and RLS) have no feedback loop, and thus the complexity is relatively low. The three equalizers with feedback loop are more stable and better suited for reducing the non-linear noise in the receiver signal.

## 4.3 Channel Models

We studied the performance of the different equalization algorithms for three different types of channels.

### 4.3.1 AWGN Channel

Additive White Gaussian Noise (AWGN) is the basic noise model used in communication systems. Additive means it is added to any noise that might be intrinsic to the communication system, white refers to the idea that it has uniform power across the frequency band, Gaussian means it has a normal distribution in the time domain with an average time domain value zero. The AWGN channel is modeled by adding white Gaussian noise to input signal. There is no phase shift or frequency offset in this channel. In our simulations, we use AWGN channel with SNR = 10dB; the Gaussian has mean of 0 and variance of 1.

### 4.3.2 Rayleigh Fading Channel

Rayleigh fading models assume that the magnitude of a signal that has passed through such a transmission medium will vary randomly, or fade, according to a Rayleigh distribution — the radial component of the sum of two uncorrelated Gaussian random variables.

Rayleigh fading is a reasonable model when there are many objects in the environment that scatter the radio signal before it arrives at the receiver. The central limit theorem indicates that, if there is enough scatter, the channel impulse response will be well-modelled as a Gaussian process irrespective of the distribution of the individual components. If there is no dominant component to the scatter, then such a process will have zero mean and phase evenly distributed between 0 and $2\pi$ radians. The envelope of the channel response will therefore be Rayleigh distributed.

In our simulation, the Rayleigh fading channel has 2 paths with path gains of 0dB and -3dB and 10 Hz Doppler shift.

### 4.3.3 Indoor Real Channel

Real indoor channel is measured in a large room with many instruments. The distance between transmitter antenna and receiver antenna is 20cm. A known sine wave is transmitted and received, and the parameters calculated by using RF tools in Matlab.

## 4.4 Simulation Results and Analysis

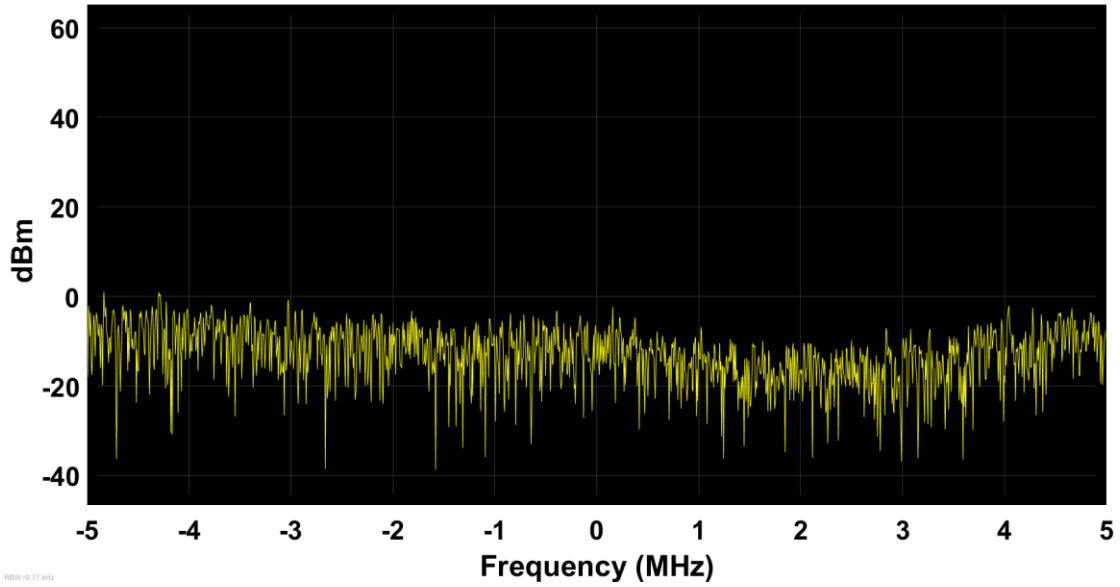


Figure 4.4 Signal before equalizer through AWGN Channel

We built a simple full-duplex system model based on the Rice architecture (see Figure 2.5). In this section we present simulation results to illustrate the performance of the different equalization algorithms for different channel models. We conduct three experiments. First,

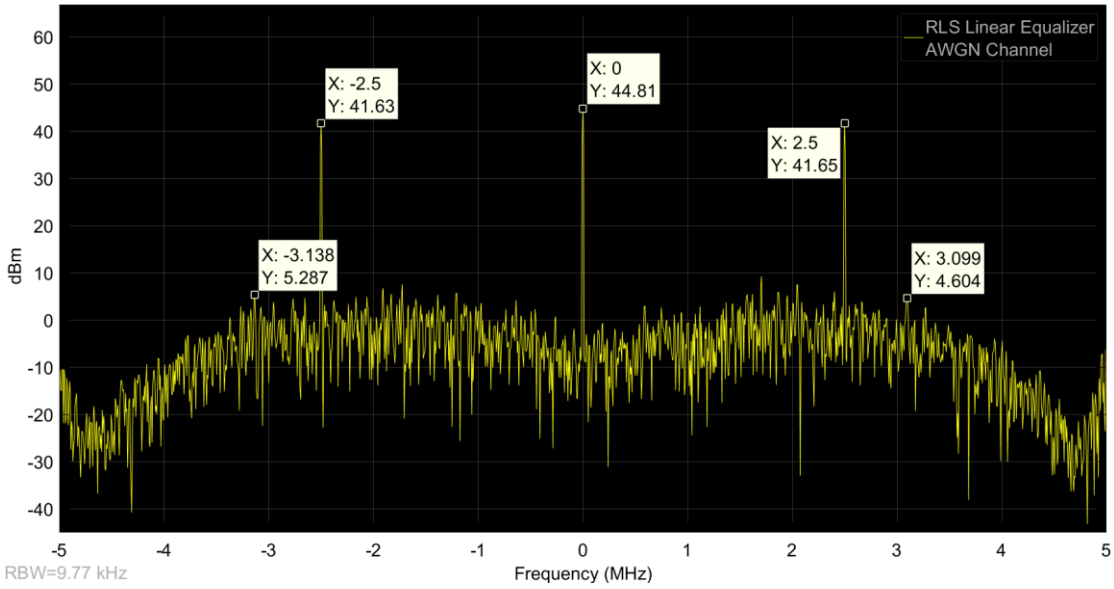


Figure 4.5 RLS Linear Equalizer in AWGN Channel

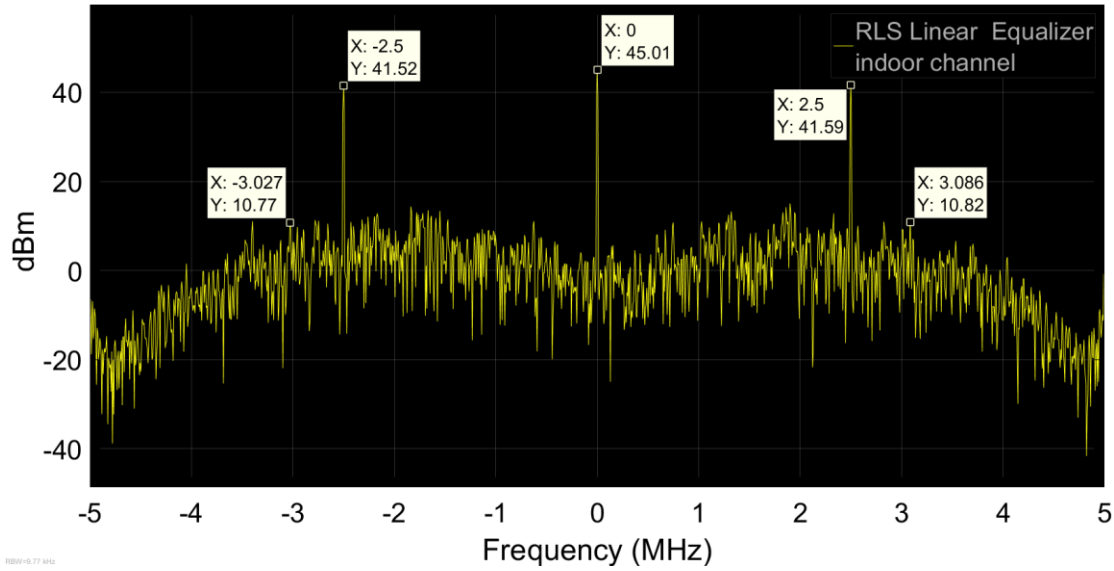


Figure 4.6 RLS Linear Equalizer in Indoor Channel

we fix the algorithm and change the channel to see the effect of channel. Next we fix the channel model and evaluate the performance of the three linear equalizers. Finally, we compare the performance of the different decision feedback equalizers.

**Experiment 1: Performance of linear equalizers for different channels**

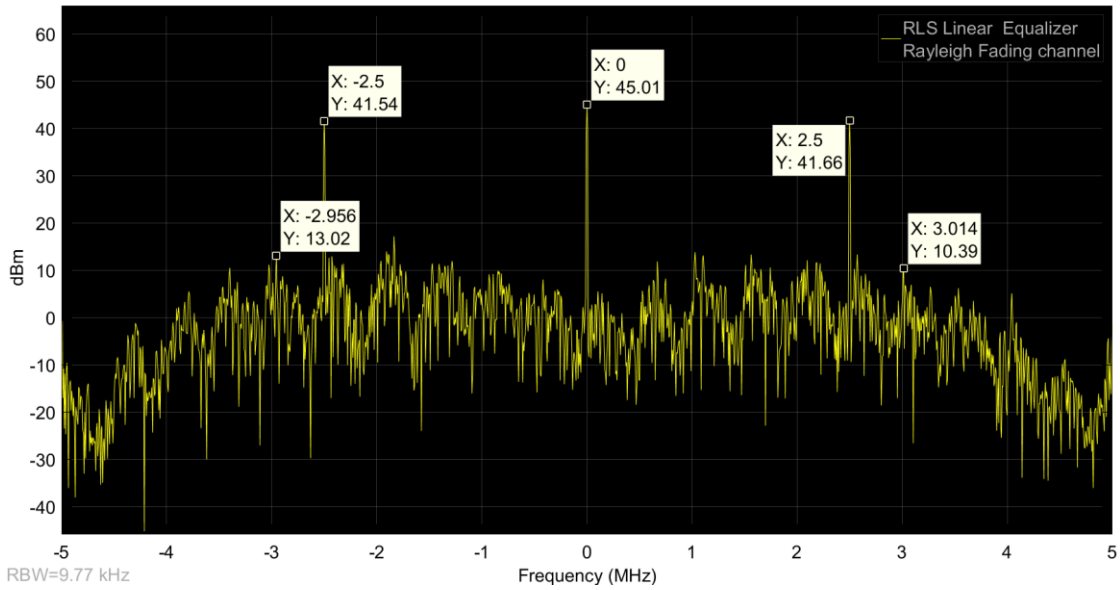


Figure 4.7 RLS Linear Equalizer in Rayleigh Fading Channel

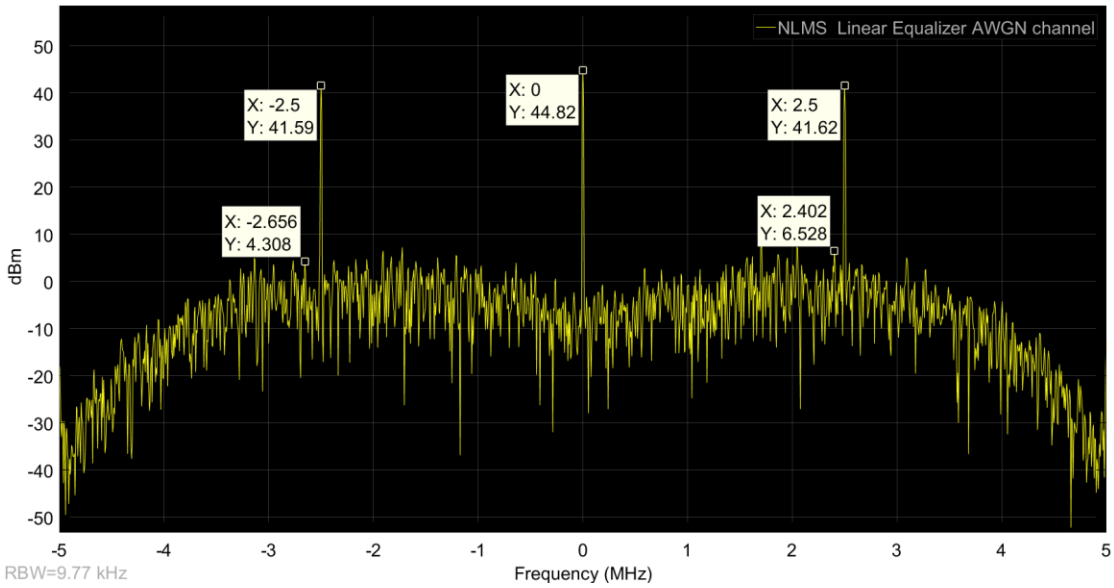


Figure 4.8 NLMS Linear Equalizer in AWGN Channel

To illustrate the need for equalization, we present Figure 4.4 which shows the signal before equalization. Figures 4.5, 4.6 and 4.7, show the performance of the RLS algorithm for

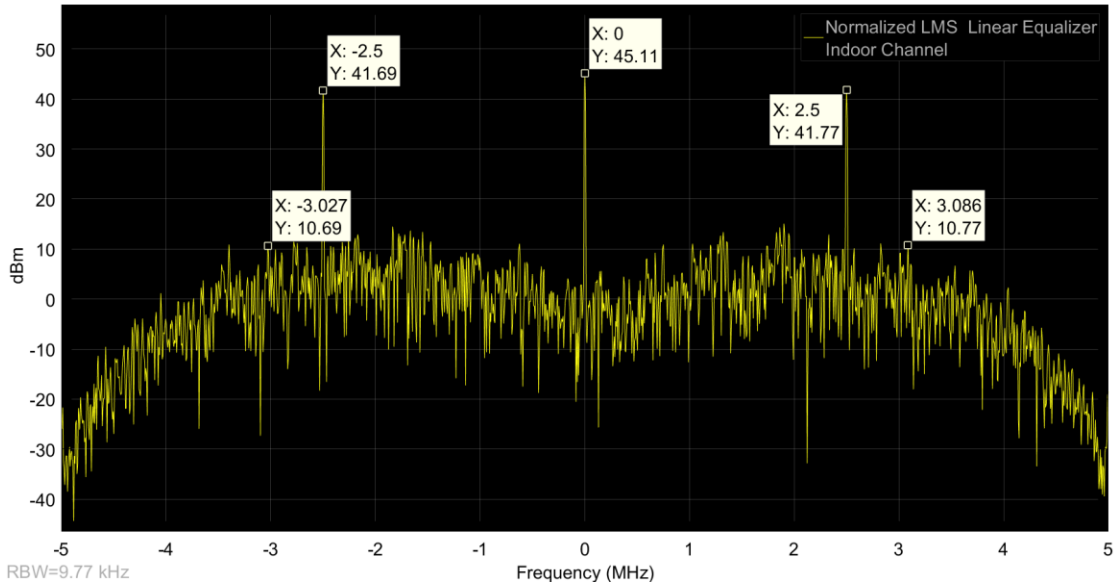


Figure 4.9 NLMS Linear Equalizer in Indoor Channel

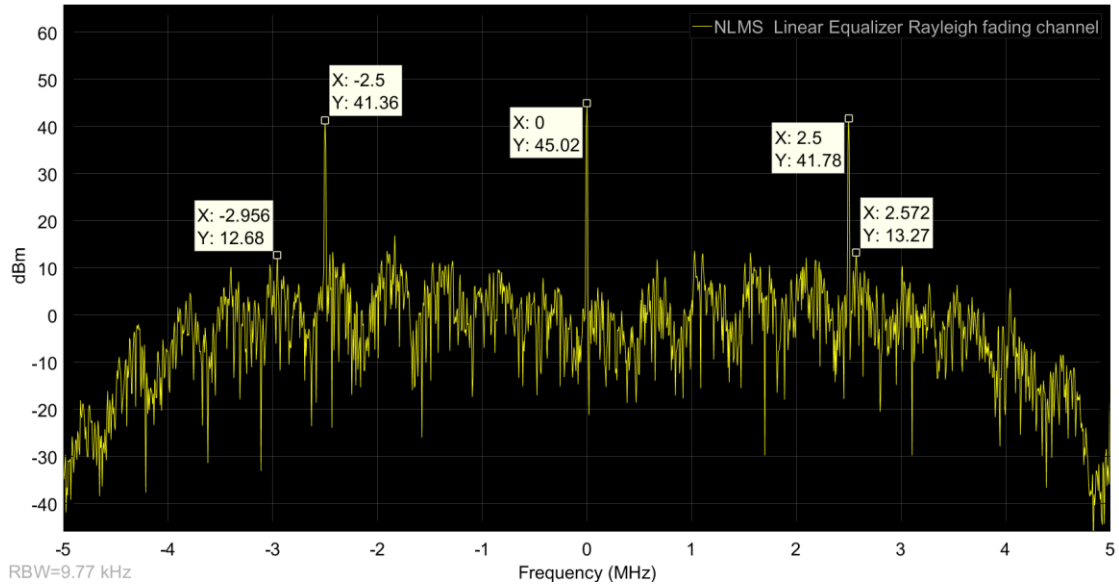


Figure 4.10 NLMS Linear Equalizer in Rayleigh Fading Channel

AWGN, Rayleigh and indoor channel, respectively. The highest peaks (which lay in 0Hz) are all around 45dBm, and the side peaks are 41.5dBm at 2.5MHz and -2.5MHz. There is significant reduction when the frequency is larger than ±4.6MHz. This shows that RLS

algorithm acts like a low pass filter and can reduce the harmonics outside ±4.6MHz. Figures 4.8, 4.9 and 4.10 show performance of the NLMS algorithm using AWGN, indoor and Rayleigh fading channels, respectively. The results for NLMS are almost the same as that of RLS with 45dBm for highest peak and 41.5dBm for side peaks at ±2.5 MHz.

After equalization, for both RLS and NLMS, the noise is 5dBm for AWGN, 11dBm for indoor channel and 12dBm for Rayleigh fading channel. Thus for AWGN channel, the highest peak is almost 40dB higher than noise, compared to indoor channel where the difference is about 35dB, and Rayleigh Fading channel where the difference is 32dB.

Thus we conclude that both RLS and NLMS algorithms have comparable results for different channel models. Since the real indoor channel has performance that is between AWGN and Rayleigh Fading channel and so the next set of experiments is conducted for the real indoor channel.

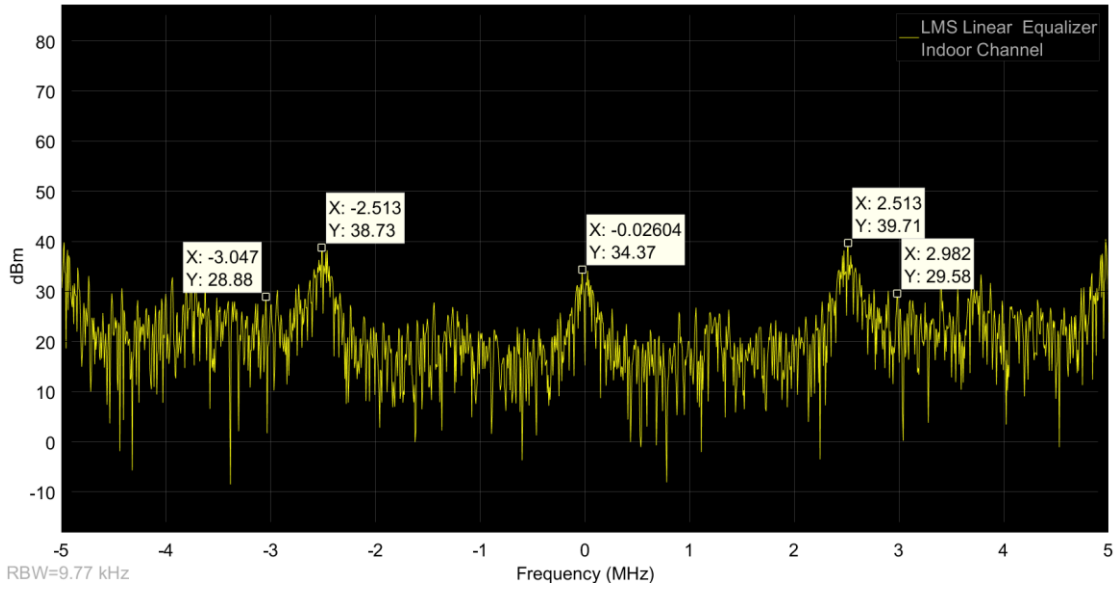


Figure 4.11 LMS Linear Equalizer in Indoor Channel

**Experiment 2: Performance of linear equalizers for indoor channel**

First we compare the performance of LMS (Figure 4.11) and NLMS (Figure 4.9) linear equalizers. The LMS linear equalizer has the worst performance; its noise floor is very high

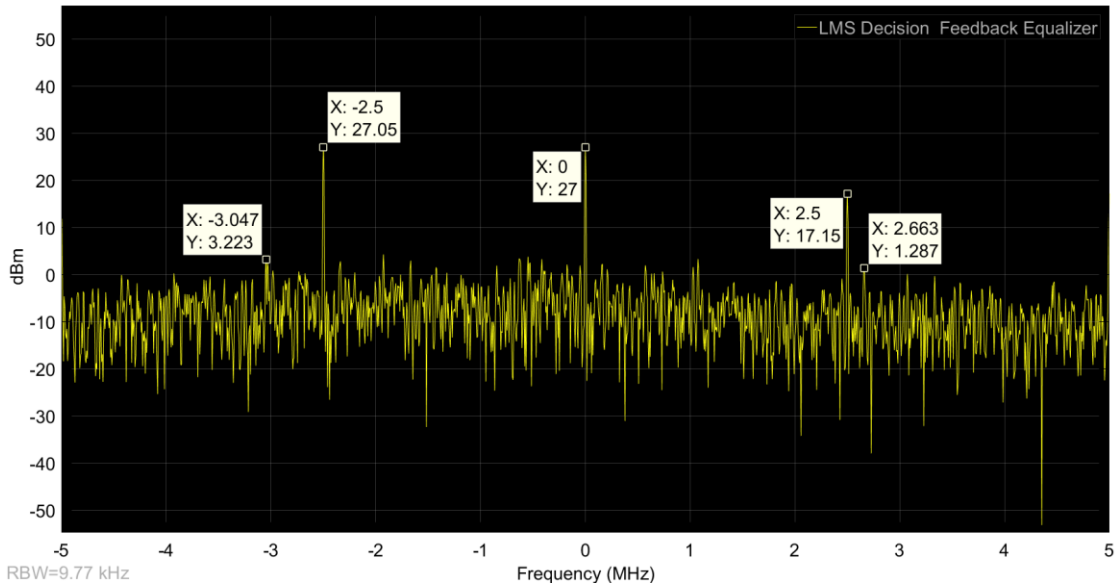


Figure 4.12 LMS Decision Feedback Equalizer in Indoor Channel

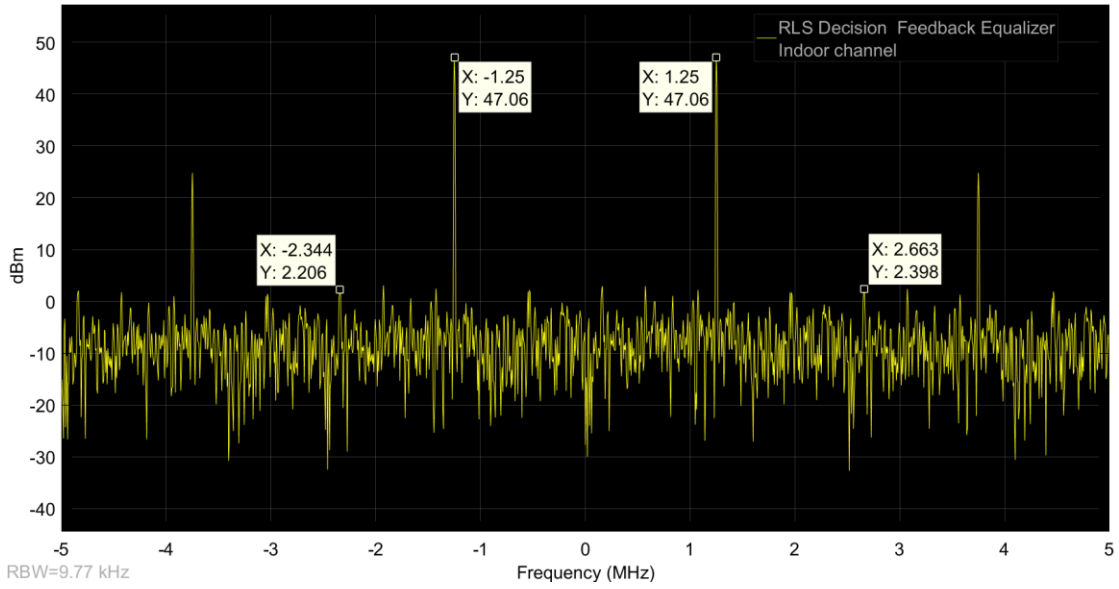


Figure 4.13 RLS Decision Feedback Equalizer in Indoor Channel

at almost 30dBm. In contrast, NLMS linear equalizer reduces the noise floor to 10dBm. Thus we can clearly see that with normalization, the performance can be improved a lot. In fact the performance of NLMS linear algorithm is compared to RLS linear algorithm (Figure 4.6), but with less computations. Also its low pass filter property is better than RLS.

**Experiment 3: Performance of decision feedback equalizers for indoor channel**

We compare the performance of the decision feedback equalizers. A comparison of LMS linear equalizer (Figure 4.11) and LMS decision feedback equalizer (Figure 4.12) shows that the noise is reduced about 30dB by using feedback path. Similarly, a comparison between NLMS linear equalizer (Figure 4.9) and NLMS decision feedback (Figure 4.13) shows that the feedback path can reduce about 10dB more noise. Finally, a comparison of

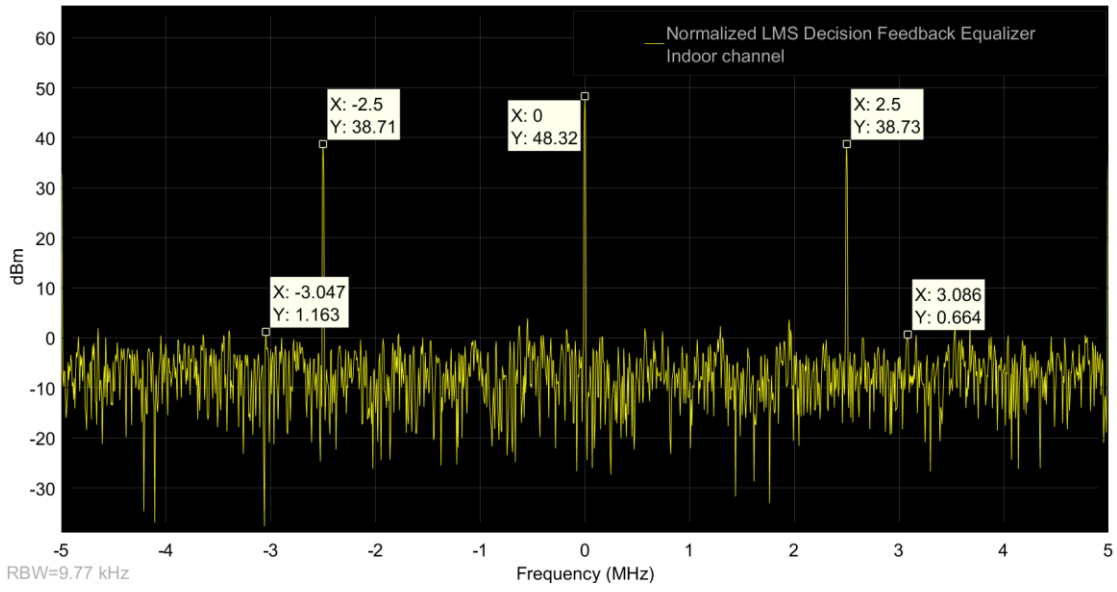


Figure 4.14 NLMS Decision Feedback Equalizer in Indoor Channel

RLS linear equalizer (Figure 4.6) and RLS decision feedback equalizer (Figure 4.13) also shows that feedback path can lower an additional 8 dB noise.

Of all the equalizers, the performance of NLMS decision feedback is the best, it cancels about 47dB of noise. Using normalization can minimize the input noise to some level, and using feedback loop can help increase the desired signal.

## 4.5 Summary

Table 4.2 summarizes the performance and noise floor reduction of the different equalization algorithm. We can see that NLMS decision feedback equalizer has the best performance and LMS linear equalizer has the worst performance. Thus NLMS decision feedback algorithm can be applied to base station, where cancellation performance is very important. For mobile terminals, we choose NLMS linear equalizer. It has good performance and lower complexity compared to RLS linear equalizer.

Table 4.2 Performance of Equalization Algorithms using Indoor Channel

| Algorithm | Performance(dB) | Noise floor(dBm) |
|---|---|---|
| LMS | 9 | 29 |
| NLMS | 34 | 11 |
| RLS | 34 | 11 |
| LMS Decision Feedback | 24 | 3 |
| NLMS Decision Feedback | 47 | 1 |
| RLS Decision Feedback | 45 | 2 |

# CHAPTER 5. CONCLUSION AND FUTURE WORK

This chapter summarizes the main contributions of this thesis and suggests possible future work.

## 5.1 Contributions

### 5.1.1 Long Range Wireless Communication Baseband Processor

In this thesis we proposed a baseband processor for long range wireless communication IoT terminals based on OFDMA and spread spectrum technologies. We first characterized the computation pattern of baseband processing operations in the two types of terminals and found that frame detection is, by far, the dominant workload. So we developed a scalar processor that is optimized for frame detection. It uses specialized chained instructions to reduce power overhead caused by register file accessing and address generation units to minimize address calculation overhead. Preliminary synthesis results in 65nm show that our processor architecture consumes 2.41nJ/cycle when running at 3MHz with 1.08V supply voltage and has an area of $0.204\mu m^2$. Based on our energy analysis, we find that for a good channel (SNR = 5dB), the simplified frame detection can be used and that the optimal setting corresponds to high threshold of 0.7, and window size of 16. For a bad channel (SNR = -11dB), the choices are limited, and it is better to choose the baseline algorithm with threshold of 0.2 and window size of 16 for energy efficiency.

### 5.1.2 Self-interference Mitigation in Full Duplex System

In this thesis we explore several equalization algorithms for digital noise cancellation in full duplex systems. Our goal is to achieve 50 dB noise reduction in the digital part. We

investigate the performance of RLS and NLMS linear equalizers for AWGN, Rayleigh fading and indoor channel models. We find that both algorithms have comparable performance for all three channel models. Since the real indoor (measured) channel has performance in between AWGN and Rayleigh fading, we run the remaining simulations on the indoor channel model. We find that the NLMS decision feedback equalizer can cancel 45dB of noise followed by NLMS and RLS linear equalizers which can cancel 35dB noise. NLMS linear equalizer has the lowest complexity of these three, and so we conclude that NLMS linear equalizer is suitable for resource-constrained mobile devices. NLMS decision feedback equalizer has the highest complexity and the best noise cancellation performance, so we conclude that it can be used in full duplex systems employed in base stations.

## 5.2 Future Work

The following items summarize possible research directions based on the work presented in this thesis. First, we plan to implement a more complex equalizer such as Maximum-Likelihood Sequence Estimation (MLSE) and analyze its performance and complexity. Since we had only considered linear noise components, next we plan to model the non-linear noise components due to harmonics. We will then derive noise cancellation algorithms in the digital domain to mitigate them.

# REFERENCES


[1] G. Lawton, "Machine-to-Machine Technology Gears Up For Growth," *Computer,* vol. 37, no. 9, pp. 12-15, 2004.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal,* vol. 1, no. 1, pp. 22-32, 2014.

[3] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys Tutorials,* vol. 16, no. 1, pp. 414-454, 2014.

[4] R. Ratasuk, N. Mangalvedhe and A. Ghosh, "Narrowband LTE-M System for M2M Communication," *IEEE 80th Vehicular Technology Conference (VTC2014-Fall),* pp. 1-5, 2014.

[5] T. I. Inc., "Long-range RF communication:Why narrowband is the factor standard," Texas Instruments Inc., 2014. [Online]. Available: http://www.ti.com/.

[6] C. Inc., "Ceva-xc5," CEVA Inc., 2015. [Online]. Available: http://www.ceva-dsp.com/CEVA-XC5.

[7] I. Cadence Design Systems, "Tensilica Fusion F1 DSP for IoT/Wearables," Cadence Design Systems, Inc, 2016. [Online]. Available: http://ip.cadence.com/ipportfolio/tensilicaip/fusion#fusion-features.

[8] S. Inc., "Artik modules overview," SAMSUNG Inc., 2016. [Online]. Available: https://www.artik.io/modules/overview/artik-1/.

[9] Y. Chen, S. Lu, H. S. Kim, D. Blaauw, R. G. Dreslinski and T. Mudge, "A low power software-defined-radio baseband processor for the Internet of Things," *IEEE International Symposium on High Performance Computer Architecture (HPCA),* pp. 40-51, 2016.

[10] N. Ma, Z. Zou, Z. Lu, L. Zheng and S. Blixt, "A hierarchical reconfigurable micro-coded multi-core processor for IoT applications," *9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC),* pp. 1-4, 2014.

[11] S. Wu, S. Kang, C. Chakrabarti and H. Lee, "Low power baseband processor for IoT terminals with long range wireless communications," *IEEE Global Conference on Signal and Information Processing (GlobalSIP),* pp. 728-732, 2016.

[12] H. Lee, T. Mudge and C. Chakrabarti, "Reducing idle mode power in software defined radio terminals," *Proceedings of the 2006 International Symposium on Low Power Electronics and Design (ISLPED),* pp. 101-106, 2006.

[13] Z. Li and G. Gielen, "Energy Normalized Correlation for Signal Acquisition in Power-Control-Absent UWB Networks," *IEEE Communications Letters,* vol. 14, no. 7, pp. 653-655, 2010.

[14] J. Oller, E. Garcia and E. Lopez, "IEEE 802.11-enabled wake-up radio system: design and performance evaluation," *Electronics Letters,* vol. 50, no. 20, pp. 1484-1486, 2014.

[15] J. I. Choi, M. Jain, K. Srinivasan, P. Levis and S. Katti, "Achieving single channel, full duplex wireless communication," *Proceedings of the sixteenth Annual International Conference on Mobile Computing and Networking,* pp. 1-12, 2010.

[16] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results," *Forty Fourth Asilomar Conference on Signals, Systems and Computers,* pp. 1558-1562, 2010.

[17] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti and P. Sinha, "Practical, Real-time, Full Duplex Wireless," *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking,* pp. 301-312, 2011.

[18] D. Bharadia, E. McMilin and S. Katti, "Full duplex radios," *Proceedings of ACM SIGCOMM 2013,* pp. 375-386, 2013.

[19] D. Korpi, M. Heino, C. Icheln, K. Haneda and M. Valkama, "Compact Inband Full-Duplex Relays With Beyond 100 dB Self-Interference Suppression: Enabling Techniques and Field Measurements," *IEEE Transactions on Antennas and Propagation,* vol. 65, no. 2, pp. 960-965, 2017.

[20] E. Ahmed, A. M. Eltawil and A. Sabharwal, "Self-interference cancellation with nonlinear distortion suppression for full-duplex systems," *Asilomar Conference on Signals, Systems and Computers,* pp. 1199-1203, 2013.

[21] "LINK LABS," LINK LABS, 27 January 2017. [Online]. Available: https://www.link-labs.com/blog/what-is-lte-m. [Accessed 27 January 2017].

[22] "LoRa Alliance," LoRa Alliance, 2017. [Online]. Available: https://www.lora-alliance.org/.

[23] "Link Labs," Link Labs, 13 January 2016. [Online]. Available: https://www.link-labs.com/blog/sigfox-vs-lora.

[24] X. Cai and G. B. Giannakis, "Error probability minimizing pilots for OFDM with M-PSK modulation over Rayleigh-fading channels," *IEEE Transactions on Vehicular Technology,* vol. 53, no. 1, pp. 146-155, 2004.

[25] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti and K. Flautner, "SODA: A Low-power Architecture For Software Radio," *33rd International Symposium on Computer Architecture (ISCA'06),* pp. 89-101, 2006.

[26] H. Lee, Y. Lee and H. Yang, "Implementation of the Chien search algorithm on application specific instruction set processor," *IEEE International Conference on Consumer Electronics (ICCE),* pp. 183-184, 2012.

[27] A. Sabharwal, P. Schniter, D. Guo, D. Bliss, S. Rangarajan and R. Wichman, "In-Band Full-Duplex Wireless: Challenges and Opportunities," *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 9, pp. 1637-1652, 2014.

[28] A. Balatsoukas-Stimming, P. Belanovic, K. Alexandris and A. Burg, "On self-interference suppression methods for low-complexity full-duplex MIMO," *Asilomar Conference on Signals, Systems and Computers,* pp. 992-997, 2013.

[29] B. Widrow, J. McCool, M. Larimore and C. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE,* vol. 64, no. 8, pp. 1151-1162, 1976.

[30] N. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 34, no. 4, pp. 793-806, 1986.

[31] J. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 32, no. 2, pp. 304-337, 1984.

[32] A. Goldsmith, Wireless Communication, New York: Cambridge University Press, 2005, pp. 105-113.