ABSTRACT

A SOFTWARE-DEFINED RADIO BASED ON THE UNIFIED SMSE FRAMEWORK

by Robert James Graessle

The purpose of this research was to implement a software-defined radio based on a recently developed framework for constructing various spectrally-modulated, spectrally-encoded (SMSE) signals. Two candidate waveforms (MC-CDMA and TDCS) are selected to demonstrate the capabilities of the framework, and they are modulated using antipodal signaling. A transmitter and receiver are each implemented on separate digital signal processor starting kits (DSK). A channel simulator consisting of additive white Gaussian noise and narrowband BPSK interferers is implemented on an FPGA. Burst transmissions from transmitter to receiver through the channel simulator are conducted to evaluate the bit-error rate performance of the system. Results from floating point simulation, fixed point simulation and hardware implementation are presented. The bit-error results from the hardware implementation closely match theoretical results. Also, TDCS is shown to mitigate effects of narrowband interference compared to MC-CDMA.

A SOFTWARE-DEFINED RADIO BASED ON THE UNIFIED SMSE FRAMEWORK

A Thesis

Submitted to the

Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Master of Science

Department of Electrical & Computer Engineering

by

Robert James Graessle

Miami University

Oxford, Ohio

2010

Advisor_____

Dr. Chi-Hao Cheng

Reader_____

Dr. Dmitriy Garmatyuk

Reader_____

Dr. Vasu Chakravarthy

Table of Contents

List of Tables

List of Figures

Acknowledgments

I am grateful to many individuals who have helped me over the course of my graduate school career. First, I would like to thank my advisor, Dr. Chi-Hao Cheng for his mentorship. Whether it was the late-night emails or driving back and forth to Dayton, his support was greatly appreciated, and his dedication to his students can never be questioned. Thank you to Dr. Dmitriy Garmatyuk and Dr. Vasu Chakravarthy for taking the time to serve on my thesis committee. I would also like to thank the rest of the faculty and staff of the Electrical and Computer Engineering department at Miami University for all of their support over the past 6 years.

I would like to express my gratitude to my colleagues at the Air Force Research Laboratory, Sensors Directorate for their support and guidance, including Cliff Bullmaster, George Gonczy, Jim Stephens, Ed Huling and Jill Johnson. A special thanks goes to the Dayton Area Graduate Studies Institute (DAGSI) for funding my graduate study.

Finally, I would like to thank my family (including my fiancée Becky Williams and her family) and friends for their love and support.

1. Introduction

1.1. Background

A spectrally modulated, spectrally encoded (SMSE) framework for generation of various spread spectrum and multicarrier communication waveforms has been developed by researchers at the Air Force Institute of Technology (AFIT) [30]. The development of the framework was motivated by the potential of software-defined cognitive radio (CR-based SDR). CR-based SDR will allow for novel wireless communications systems that will support multiple capabilities (cellular, Bluetooth, radio and TV, 802.11, etc.) operating on a single platform. These platforms will require waveform agility (the ability to generate multiple communications waveforms and operate seamlessly with all of them) as well as a cognitive ability (in order to select the most appropriate waveform to use, while making the selection process transparent to the user). As the electromagnetic spectrum becomes more congested, CR platforms will also require the ability to dynamically adjust the spectrum usage of the radio in response to its changing environment. Spread spectrum and multicarrier waveforms thus make a logical choice for inclusion in future CR platforms [4]. In published papers, the developed framework has been shown to correctly generate multicarrier waveforms and produce bit-error rate (BER) curves that match theoretical results [30]. However, most of this work has been done using computer simulation up until very recently. Hardware implementations based on the SMSE framework are an active area of research [41]. This thesis seeks to add to this body of knowledge by developing and studying an embedded hardware implementation of a transmitter/receiver pair based on the SMSE framework.

1.2. Problem Statement

This thesis represents an attempt to construct a transmitter/receiver pair based on the SMSE framework. Design decisions must be made while realizing the hardware implementation from a software simulation. This thesis will serve as an initial DSP implementation based on the SMSE framework and as a proof-of-concept, eventually leading to a fully functional SMSE implementation. While the SMSE framework is capable of implementing many communications waveforms, we have chosen to implement two spread spectrum waveforms

(MC-CDMA and TDCS) that we believe will give a good representation of the capabilities of the framework. The scope and assumptions of this project are discussed in more detail in Section 1.4.

1.3. Goals

There are three primary goals of this research:

1. To evaluate the bit-error rate performance of a hardware implementation based on the SMSE framework and compare it to previously published results.
2. To evaluate the capability of a digital signal processor (DSP) in implementing a SMSE transmitter or receiver and to analyze any difficulties encountered in transitioning from a simulated implementation to a hardware implementation.
3. To compare the performance of the hardware implementation when transmitting using MC-CDMA vs. TDCS in the presence of narrowband interference.

1.4. Scope & Assumptions

As an initial implementation and proof of concept, the scope of this research was narrowly defined. The major decisions and assumptions that were made in defining this research's scope are summarized below.

1. The SMSE transmitter and receiver operate at baseband. While this is unfeasible for practical wireless applications where RF transmission is required, it reduces the complexity of the transmitter/receiver implementation and the hardware required.
2. The channel is modeled simply as additive white Gaussian noise (AWGN) with variable signal-to-noise ratio, with the addition of several narrowband (BPSK) interferers spread across the bandwidth of interest. No fading or multipath effects are considered.
3. Since the channel does not contain fading or multipath effects, channel equalization at the receiver is not considered in this thesis.
4. While numerous SMSE waveforms can be generated using the framework, only two (MC-CDMA and TDCS) are considered in this thesis. MC-CDMA is chosen because it is a popular and well-studied transmission scheme due to its high spectral efficiency, multiple access capability and ease of implementation using the Fourier transform [14].

TDCS is chosen because it is a novel transmission scheme that has not received much attention in the literature and because it may offer performance improvements over MC-CDMA in the presence of narrowband interference. It is believed that these two transmission schemes have several fundamental differences [4] and thus can give an accurate representation of the capabilities of the SMSE framework.

5. Binary phase-shift keying (BPSK) is the only modulation method considered in this thesis. *M*-ary PSK will be considered as part of the future work.
6. No networking scheme or protocol is considered above the physical layer.
7. The transmitter and receiver are not implemented to operate in real-time. Instead, the transmitter sends a series of burst transmissions of finite length that are received and processed by the receiver. By performing multiple burst transmissions in succession, bit-error analysis may still be accurately performed to analyze the performance of the system.

1.5. Methodology

First a software simulation of a SMSE-based transmitter/receiver (with AWGN channel and narrowband interference) is developed. The outputs of the simulation (communication waveforms and BER curves) are compared to published results to verify the correctness of the simulation. Following this verification step, the simulation is modified to produce a fixed-point simulation that quantizes all variables to finite bit width in order to consider quantization effects. The outputs of the fixed point simulation are compared to those of the floating point simulation to investigate the finite-word effects and determine the expected hardware performance. The fixed point simulation is then used as a basis for writing the embedded code that will run on the DSP. The transmitter and receiver are each implemented on a separate DSP. The two DSPs are connected to separate host PCs for configuration and control purposes. A field-programmable gate array (FPGA), connected to one of the PCs, is used to add AWGN and narrowband interference into the transmitted signal in order to model the effects of the channel. The transmitter DSP, FPGA, and receiver DSP are connected by coaxial cables. For testing, burst transmissions are sent repeatedly from transmitter to receiver while varying the SNR produced by the channel simulator.

1.6. Materials

The transmitter/receiver pair is implemented on identical Spectrum Digital TMS320C6416 DSP Starter Kits (DSK). The kits contain a fixed-point DSP running at 1 GHz. The kits are programmed using Texas Instruments Code Composer Studio 3.1 and are connected to the host PCs via a USB interface. Analog-to-digital and digital-to-analog conversion is performed by an NDTech DSP Star AD/DA module that interfaces to the DSK via an EMIF interface. The module contains four 12-bit A/D converters operating at a combined maximum of 6 MSPS and two 12-bit D/A converters operating at a maximum of 165 MSPS. Each DSK contains an identical AD/DA module [23].

The channel simulator is implemented using a Nallatech XtremeDSP Development Kit-IV. This kit consists of a Xilinx Virtex-4 XC4SX35 FPGA, two 14-bit analog-to-digital converters operating at a maximum 105 MSPS, and two 14-bit digital-to-analog converters operating at a maximum of 160 MSPS [39]. The kit is programmed using Nallatech's FUSE software suite and connects to a host PC via a PCI interface. The FPGA designs are developed as Simulink models using Xilinx System Generator [38].

The initial floating-point and fixed-point simulations are developed using MATLAB R2009a. All software for simulations and programming/controlling the DSKs and FPGA runs on Dell desktop computers with an Intel Core 2 processor running at 2.1 GHz and 3 gigabytes of memory.

1.7. Overview

This thesis is organized into eight chapters. Chapter 2, the Literature Review, contains the following: a brief history of software-defined radio and cognitive radio, descriptions of the waveforms being considered in this thesis, a description of the unified SMSE framework, and a description of the DSP platform and design flow used for the implementation. Chapter 3 contains a description of the design process (from floating point simulation to hardware implementation) for the transmitter, receiver and AWGN generator. Chapters 4, 5 and 6 present detailed descriptions of the final implementations of the transmitter, channel model and receiver, respectively. Chapter 7 presents the results of the testing of the simulations and hardware

implementation and a discussion of the meaning of the results. Chapter 8 concludes the thesis and offers suggestions for future work to be performed.

2. Literature Review

2.1. A Brief History of Software-Defined Radio and Cognitive Radio

The concept of *software-defined radio* was first presented by Joseph Mitola in 1992 [20]. He defined a software-defined radio as simply a radio in which the typical hardware components (amplifiers, mixers, oscillators, filters, etc.) have been replaced by software that performs the same function. His idealized depiction of a software radio had no analog hardware parts; the antenna was directly connected to an analog-to-digital converter and all processing of the signal was done digitally via software. In practice, due to the high bandwidths and sample rates required for RF operation, conventional microprocessors were not powerful enough to handle processing for an idealized SDR. This is still the case for some applications even today, though faster computing devices such as FPGAs and ASICs have made the idealized software radio a more feasible possibility. Most current SDR applications make use of hardware components to demodulate and down-sample the incoming signal to a bandwidth and sample rate that can be processed in software by the computing hardware available [31].

One of the first major implementations of a software-defined radio was the U.S. Army's SpeakEasy project in the 1990's. This project sought to use software to provide interoperability between 10 different radio platforms over a bandwidth of 2 to 2000 MHz. Texas Instruments DSPs were used in addition to FPGAs. The final implementation consisted of several hundred processors and filled the back of a truck [19]. The U.S. military has continued its investigation into SDR with the Joint Tactical Radio System (JTRS). This is an ongoing project scheduled for completion in 2010 that will develop a new field radio for use by ground, sea and air forces [7]. A notable SDR endeavor outside of the military domain is the GNU Radio project, a software toolkit allowing for open-source development of radio software. Custom-designed software can interface with a Universal Software Radio Peripheral (USRP), a hardware board consisting of high-speed A/D and D/A converters and an FPGA for down-conversion and up-conversion. Processing is performed on general-purpose microprocessors. The GNU Radio toolkit and USRP, when coupled together, greatly increase the SDR functional capabilities of conventional

microprocessors [24]. In fact, the USRP has been used as the implementation platform of choice for previous implementations of SMSE-based software-defined radios [41].

In 1999, Mitola described *cognitive radio* as a radio that can adapt its transmission or reception parameters in response to a changing environment [21]. The so-called "Mitola radio" (or "full" cognitive radio) is capable of making decisions on transmission and reception parameters based on information gathered besides just the spectral environment. For example, it might be able to sense its location (physically and/or within the network topology) and identify other nearby users of the channel and adjust its operating parameters accordingly. In another type of CR, the spectrum-sensing cognitive radio, only the observed RF spectral energy is used as a basis for adjusting radio parameters. Such a radio is able to adapt its operations to improve the performance of its own communications while minimizing the negative effect on other spectrum users. It also seeks to maximize the efficient use of the spectrum. To these ends, there are 3 main tasks of a cognitive radio [15]:

1. Detection of the RF spectrum environment (including spectrum holes)
2. Estimation of channel state and prediction of channel capacity
3. Regulating its transmit power to manage the spectrum in real-time

The idea of cognitive radio is a natural extension of software-defined radio, since CR is made possible by SDR. Indeed, in recent years the two terms have become less and less distinct. A software-defined radio now may be expected to hold one or several of the characteristics commonly associated with CR; namely, the ability to operate in several different modes and to select the optimal mode dynamically based on its own observations.

2.2. Spectrally-Modulated, Spectrally-Encoded Waveforms

Spectrally-modulated, spectrally-encoded (SMSE) waveforms are modulated and encoded in the frequency domain as opposed to the time domain. After the modulation and encoding, an inverse Fourier transform operation is performed on the signal before transmission. At the receiver side, a Fourier transform is performed on the received signal. Demodulation and decoding is then performed using the frequency domain representation of the received signal.

SMSE waveforms are considered to be good candidates for cognitive radio [15] [4]. They allow for computationally efficient generation of modulating waveforms with dynamically changing spectral content. This is well-suited for a cognitive radio, which will need to adapt its transmissions to a changing spectral environment in order to avoid interfering with other users.

Myriad SMSE waveforms exist; however, only two are considered in this thesis: Multicarrier Code-Division Multiple Access (MC-CDMA) and the Transform Domain Communications System (TDCS). This section presents an overview of the history, operation and distinguishing features of each of these waveforms. First, descriptions of Orthogonal Frequency-Division Multiplexing (OFDM) and Code Division Multiple Access (CDMA) are presented as a precursor and motivation to discussions of MC-CDMA and TDCS.

2.2.1. Orthogonal Frequency-Division Multiplexing (OFDM)

OFDM is a specialization of Frequency-Division Multiplexing (FDM). In FDM, a single signal of a larger bandwidth is multiplexed into several signals (known as *subcarriers*) of a smaller bandwidth. The primary motivation for this is interference avoidance. For a system with data rate R and bandwidth B divided into N subcarriers, each subcarrier has a bandwidth $B_N = B/N$ and a data rate $R_N \approx R/N$ [10]. Each subcarrier also has reduced symbol duration when compared with the single-carrier system. This reduces the effect of multipath and frequency-selective channel fading [4] [10]. For a sufficiently large N, B_N can be made much less than the coherence bandwidth B_c of the channel. Or, when viewed in the time domain, the symbol time can be made much greater than the channel delay spread, reducing the amount of intersymbol interference caused by the channel [10]. It is even possible to implement a working FDM system without the channel equalizer typically required for wireless systems [14].

OFDM differs from FDM in that it requires less bandwidth. In FDM, each subcarrier is separated by a guard interval in order to avoid crosstalk between adjacent channels. In OFDM, adjacent channels overlap; however, crosstalk is mitigated by ensuring orthogonality between adjacent subcarriers. The minimum frequency separation allowed between subcarriers while maintaining orthogonality is $1/T_N$, where T_N is the symbol period. Each subcarrier can be expressed as:

$$\cos\left(2\pi\left(f_0 + \frac{i}{T_N}\right)t + \varphi_i\right), i = 0, 1, 2, \ldots \quad (2.1)$$

where f_0 is the frequency of the lowest-frequency subcarrier, i represents the subcarrier number, and φ represents any set of subcarrier phase offsets [10]. The spectrum of each subcarrier has a $sinc^2$ function shape, with the spectral peak of each subcarrier occurring in the spectral nulls of all other subcarriers. In this fashion, interference among subcarriers is mitigated [4].

To carry out OFDM transmission, a serial data stream is split into N parallel streams. Each stream is modulated (using a PSK or QAM scheme) onto a subcarrier. In practice, this is done by applying the data modulation to a frequency-domain vector with each element representing a subcarrier, then performing an inverse Fourier transform to generate the time-domain OFDM waveform [5]. Prior to the development of realizable hardware implementations of the discrete Fourier transform, OFDM transmission was performed using a bank of sinusoidal modems for each subcarrier, resulting in bulky and complex implementations. Therefore, the development of fast Fourier transform algorithms represented a major milestone in OFDM development and allowed for significant reductions in OFDM modem implementation complexity [14].

In order to mitigate detrimental performance effects (e.g. intersymbol interference) caused by multipath when the delay spread is longer than the symbol duration, a guard time is introduced between consecutive OFDM symbols. This guard time is filled by copying N_T time-domain samples from the end of each OFDM symbol to its beginning. The number of extension samples N_T required depends on the channel transient response and the number of modulation levels [14]. This extension of the time-domain symbol is known as a *cyclic extension* or *cyclic prefix*. The time-domain symbol has a length of $N + N_T$ samples with an overhead of N_T/N introduced by the cyclic extension (N_T samples of redundant data are transmitted). Alternatively, the cyclic extension may be filled with all zeros instead of repeating samples, thus reducing transmit power used by the system, although it results in a more complex implementation than using the traditional cyclic prefix [10]. The resultant time-domain vector is passed through a parallel-to-serial converter to generate a single data stream for transmission. Transmission is then performed using a sinusoidal modulator and a digital-to-analog converter [10].

At the receiver side, the signal is demodulated down to baseband, filtered, and passed through an analog-to-digital converter. Following synchronization, the cyclic prefix is removed from each symbol and the serial stream is once again split into *N* parallel streams. A Fourier transform is performed on the parallel vector yielding a frequency-domain vector representing the OFDM symbol with each vector element corresponding to a subcarrier. Demodulation (QAM or PSK) is performed on each vector element by comparing the received element to the symbol constellation and selecting the symbol value with the smallest distance to the received element [10].

2.2.2. Code Division Multiple Access (CDMA)

CDMA is a multiple access scheme where each user's transmissions are modulated by a different spreading code. The spreading code may be orthogonal (in which transmissions over the same time period and bandwidth do not interfere with one another) or nonorthogonal (in which transmissions over the same time period and bandwidth may interfere with one another, but the interference is mitigated due to the correlation properties of the spreading sequences in use) [10].

Typically, the spreading is accomplished by either Direct Sequence Spread Spectrum (DSSS) or Frequency Hopping Spread Spectrum (FHSS). In DSSS, the information being transmitted is modulated by a spreading code (with a value of +/- 1) with a chip rate that is faster than the information signal bit rate. Thus, the signal is spread in the frequency domain. Spreading the signal in this way makes it less susceptible to narrowband and intersymbol interference (ISI). When a portion of the bandwidth is affected by narrowband interference or frequency-selective fading, the signal can still be accurately recovered from the signal energy present in the unaffected bandwidth. In FHSS, the spreading code sequence is used to generate a *hop pattern* for each user. The FHSS transmission consists of a narrowband carrier signal that rapidly switches carrier frequencies across a defined bandwidth, in accordance with the hop pattern defined by the spreading sequence. If the spreading codes are orthogonal and the users are synchronized to one another, the users' transmissions never collide. However, if nonorthogonal codes are used or the users are not perfectly synchronized, multiple users' transmissions may collide by occupying the same channel at the same time [10].

These two flavors of CDMA are known as DS-CDMA and FH-CDMA, respectively. Studies indicate that DS-CDMA is superior to FH-CDMA in terms of the number of users that are capable of transmitting simultaneously while avoiding interference. Furthermore, interference between users in an FH-CDMA system occurs at much greater distance between users than in DS-CDMA. FH-CDMA does benefit from greater resistance to strong interference; by avoiding interference (hopping around it) instead of mitigating it (increasing the processing gain) the performance degradation caused by a strong interferer can be better avoided [10].

2.2.3. Multicarrier Code-Division Multiple Access (MC-CDMA)

MC-CDMA may be viewed as combining DS-CDMA with OFDM. Like OFDM, an MC-CDMA signal consists of several subcarriers. However, while OFDM transmits a different symbol on each subcarrier, MC-CDMA transmits the same symbol across all subcarriers. Prior to the inverse Fourier transform, each subcarrier is multiplied by a different chip of a binary spreading sequence (+/- 1). This is similar to direct-sequence spread spectrum (DSSS), except MC-CDMA performs the signal spreading in the frequency domain as opposed to the time domain. Key benefits of MC-CDMA include immunity to intersymbol interference and multiple access capability [10]. In a multiuser MC-CDMA system, the spreading sequence provides user separation; each transmitter modulates its signal with a unique spreading code such that the cross-correlation of one code with another does not yield any significant peaks. Similar to DSSS, this property allows multiple users to transmit simultaneously while ensuring that each user's transmissions can be separated out at the receiver. The users benefit from the reduction of fading effects due to the frequency diversity obtained via the use of subchannels [10].

In practice, an MC-CDMA transmitter can be implemented by concatenating a CDMA spreader and OFDM transmitter. The input data stream is split into several parallel streams (note, however, that each parallel stream contains one identical sample of the input serial stream. In other words, both the serial and parallel streams sample at the same rate). Each parallel stream is modulated by a chip from a binary spreading sequence. The process past this point is exactly identical to that of OFDM: each stream is modulated at baseband and the resulting vector is passed through an inverse Fourier transform to generate a time-domain waveform. Like

OFDM, a cyclic extension is introduced between symbols in order to reduce intersymbol interference [4].

The MC-CDMA receiver performs the reverse process. Synchronization is performed (based on the spreading sequence) following downconversion and digitization. The cyclic extension is removed from each symbol, a Fourier transform is performed, and the subcarriers are despread and demodulated [4] [10]. Prior to demodulation, each subcarrier i is multiplied by a weighting factor β_i. These factors are used to equalize the received signal based on the frequency response of the channel; some subchannels may experience more fading than others, and this disparity must be equalized at the receiver. Two common techniques for selecting the subcarrier weighting factors are Equal Gain Combining (EGC) and Maximum Ratio Combining (MRC) [14].

2.2.4. Transform Domain Communications System (TDCS)

TDCS is a novel communications scheme that has not yet received much attention in the literature [4]. It was developed as a joint effort between the Air Force Research Laboratory (AFRL) and the Air Force Institute of Technology (AFIT). The first technical report describing a system similar to TDCS was written by German in 1988 [8], leading to a patent describing a "Low probability of intercept communication system" that was filed by Andren on behalf of Harris Corporation in 1991 [1]. The TDCS itself was first described by Radcliffe in his M.S. thesis of 1996 [27], and the block diagram of the system he proposed then is the same as used in TDCS today [11]. Radcliffe demonstrated that, for a single stationary transmitter and receiver operating in an AWGN channel with various types of interference (including fixed tones, swept tones and partial band interference), TDCS was better at interference avoidance and mitigation than conventional direct sequence spread spectrum [27].

The goals of TDCS are to avoid interfering signals in the spectrum of operation while also ensuring low probability of intercept and detection (LPI/LPD) [12]. TDCS does this by dynamically altering the frequency components of the transmission waveform (known as the Fundamental Modulation Waveform, or FMW) based on the observed spectrum of the environment. A block diagram of the TDCS is shown in Figure 2.1. The first step in TDCS communications is spectrum identification, which is performed at both the transmitter and the

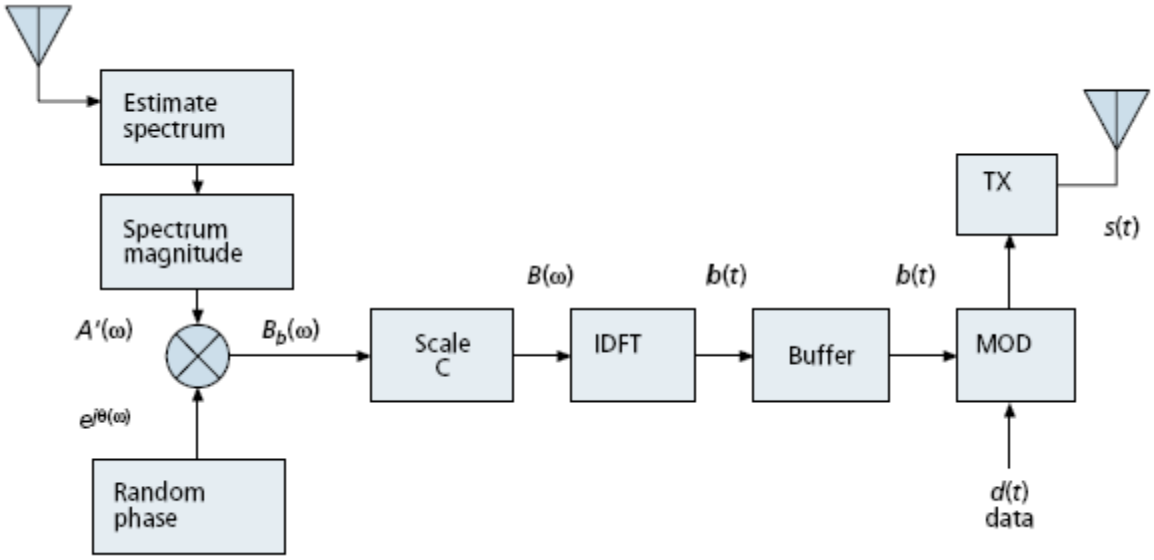


Figure 2.1 – TDCS Transmitter Block Diagram [4]

receiver. A spectral estimation technique such as a periodogram can be used to sample the spectrum activity at the transmitter's location. The result of the spectral estimation is fed to a spectrum magnitude block which places a hard threshold on the spectral estimation. Frequency components with power exceeding the threshold are set to zero (nulled) while frequency components that do not exceed the threshold are assigned a gain value of one. This block results in a vector $A(\omega)$ depicting the "clean" interference-free spectrum [4].

After the spectrum vector is generated, a pseudorandom (PR) phase vector is applied to $A(\omega)$ to generate $B_b(\omega)$. The purpose of the PR phase is to ensure that the time-domain waveform exhibits weak correlation properties, thus contributing to LPI/LPD. r bits of a binary PR sequence, which can be generated by a shift register, are mapped to one of 2^r phase values equally spaced around the unit circle, as shown in Figure 2.2. Another important benefit of the PR phase is multiple access capability: each user pair can be assigned a unique PR sequence, allowing each pair's transmissions to be distinguishable when transmitted simultaneously across the same bandwidth. This may appear similar to a spread spectrum multiple access scheme such as Direct Sequence CDMA (DS-CDMA). However, the key difference between DS-CDMA and TDCS is that in the former the PR code is used to spread the spectrum, while in the latter the PR code is used to randomize the phase of the subcarriers, leading to LPI/LPD [4].

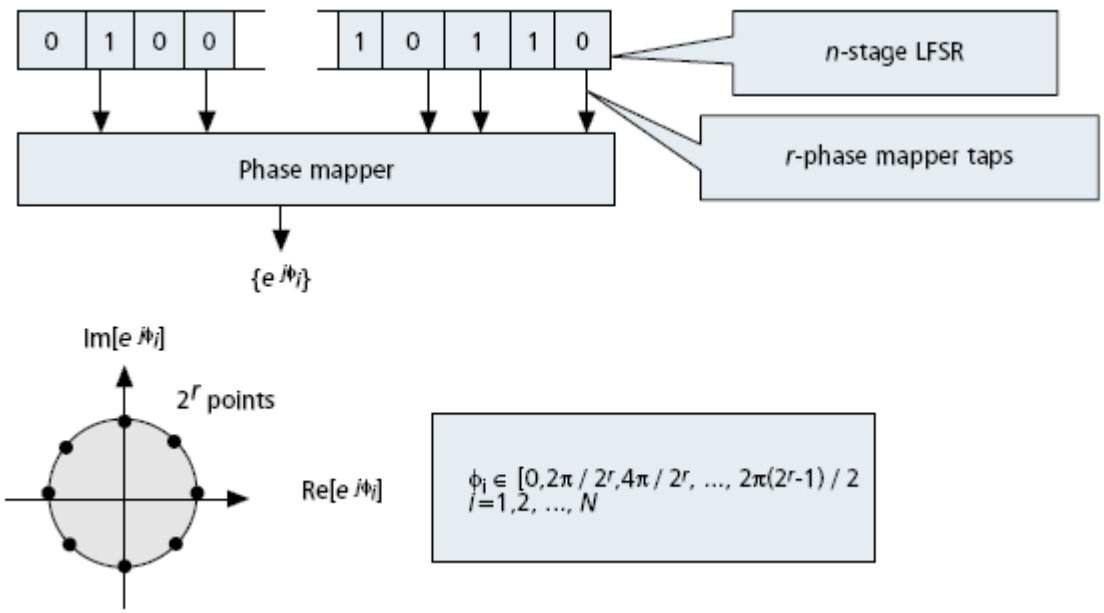


Figure 2.2 – Generation of PR code in TDCS [4]

The TDCS complex frequency-domain vector (with random phase components) is then scaled so that the transmitted energy of each symbol is relatively equal. Scaling helps to mitigate detrimental effects of a high peak-to-average power ratio caused in systems with nonlinear power amplifiers [14]. This scaling also allows for consistent measurement of E_b/N_0 and bit-error rate through the channel during experiments [11].

After scaling, the resultant complex vector is passed through an inverse Fourier transform to generate the time-domain FMW. This is done M times for M-ary modulation, generating M FMWs, one for each possible symbol to transmit. Each FMW is generated by rotating the original FMW to generate a FMW for the other symbols (e.g. for quarternary modulation, the original FMW is rotated by 90, 180, and 270 degrees to generate four total waveforms) [11]. If antipodal modulation is used, as is the case in this thesis, the second FMW will be 180 degrees shifted from the other. The FMWs are typically generated once, stored at the transmitter, and used to transmit multiple consecutive symbols. The FMWs are only updated when operational conditions dictate. This is different from OFDM, where an inverse Fourier transform is required for each symbol transmission [4]. Finally, the FMWs are concatenated appropriately for the bits in the information signal before being transmitted [11].

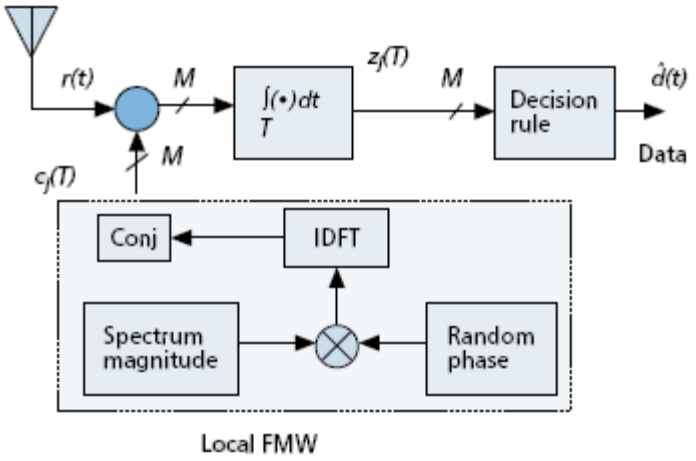


Figure 2.3 – Block diagram of TDCS receiver [4]

At the receiver, synchronization is performed, typically by correlating the received time-domain signal with a reference TDCS waveform. The receiver contains an identical structure to the transmitter for generating the reference FMWs (Figure 2.3). The received waveform is correlated with each reference FMW, and a maximum likelihood decision rule is used to estimate the received symbol [4]. Note that in order to generate the reference FMWs, it is necessary to sample the spectral environment to determine which frequency components need to be notched out. In previous work describing the TDCS, it is assumed that the transmitter and receiver exist in the same environment and thus will notch out identical components [11].

2.2.5. Differences between OFDM, MC-CDMA and TDCS

Chakravarthy et al [4] summarized the key differences between the aforementioned SMSE waveforms. More specifically, they sought to dispel the notion that TDCS was similar in principle to OFDM or MC-CDMA. Their points of emphasis were:

- TDCS was designed specifically for avoiding intentional interference (e.g. jamming), while OFDM and MC-CDMA were designed primarily to mitigate natural channel effects (e.g. fading).
- OFDM transmits multiple data symbols in each OFDM symbol, while in TDCS one data symbol is transmitted in each TDCS symbol.

- OFDM requires an inverse Fourier transform operation for each set of data symbols being transmitted, while TDCS can transmit multiple sets of data symbols using a single FMW and a single inverse Fourier transform operation. The number of inverse Fourier transforms required can be altered depending on environmental factors.
- OFDM and MC-CDMA subcarriers are individually modulated using PSK or QAM, while TDCS typically utilizes antipodal signaling or cyclic shift keying of the entire waveform.
- OFDM ensures orthogonality through subcarrier spacing, while TDCS ensures orthogonality through pseudorandom phase. TDCS makes use of a pseudorandom code in ensuring orthogonality while OFDM does not.
- TDCS and MC-CDMA both make use of pseudorandom phase for multiple access capability. However, MC-CDMA uses the phase for signal spreading while TDCS does not. TDCS uses the pseudorandom phase to make waveforms that are more noise-like with weak correlation properties. Furthermore, in MC-CDMA the phases are limited to binary values (0 or π), while in TDCS the phases can take on any of 2^r phase values (equally spaced around the unit circle) when using r bits of a PR sequence.
- In OFDM and MC-CDMA, the data symbols modulate the carrier bins directly, while in TDCS they do not modulate the carrier bins directly. Instead, they modulate the fundamental modulation waveforms.

2.2.6. SMSE Waveforms as Candidates for Cognitive Radio

The aforementioned spread spectrum techniques are considered to be good candidates for cognitive radio. Haykin [15] states that a cognitive radio must be able to fill in spectrum holes and adapt to dynamic conditions of the radio environment. He specifically mentions OFDM as a good candidate due to its spectral flexibility and computational efficiency. He uses OFDM to accomplish dynamic spectrum management by changing the bandwidths and center frequencies of the OFDM subcarriers and varying the number of bits per symbol for each subcarrier. TDCS accomplishes the goal of dynamic spectrum access by dynamically modifying the frequency component of the fundamental modulation waveform. MC-CDMA operates similarly to TDCS and OFDM (through modulation and encoding in the frequency domain followed by an inverse Fourier transform) despite being fundamentally different in some ways, so it is included in this

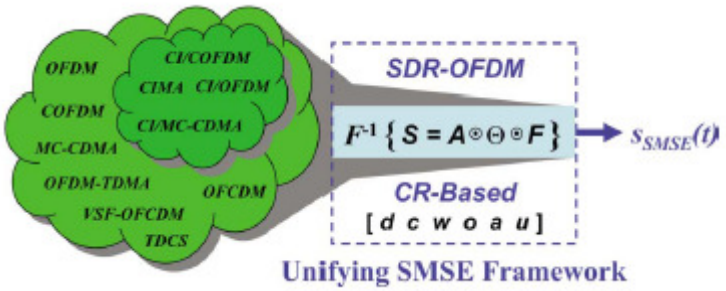


Figure 2.4 – Overview of the unified SMSE framework [30]

discussion of SMSE waveforms [4]. All 3 techniques possess the ability to dynamically modify the frequency domain representation of the waveform. This is well-suited for a cognitive radio, which will need to possess dynamic spectrum management capability in order to avoid interfering with other users.

2.3. The Unified SMSE Framework

In 2006 Roberts et al [28] proposed a single mathematical framework, from which several classes of SMSE waveforms can be generated. As can be seen in Figure 2.4, the framework allows for the generation of the waveforms discussed previously (OFDM, MC-CDMA, TDCS) as well as several others that are not considered in this thesis.

This framework was developed as an offshoot of an earlier attempt to develop a unifying analytic framework for ultra-wideband (UWB) signals. This UWB framework modeled a transmitter and Rake receiver allowing for general analysis, characterization and implementation of UWB signals [40]. The new SMSE framework differs from the UWB framework in that the signals are spectrally designed as opposed to temporally designed. The eventual goal is to combine the two types of signals into a unifying CR-based SDR framework for all signals [30]. The original proposal [28] included only a transmitter framework and only applied to three OFDM-based signals; in 2007 the framework was expanded to include a full transmitter-receiver structure and additional SMSE signals [30].

The framework consists of six complex-variable frequency domain vectors (*d, c, w, o, a, u*) that are combined via Hadamard multiplication as in (2.2) to generate the SMSE waveform.

$$s_k[m] = a_m u_m c_m d_{m,k} w_m e^{j(\theta_{c_m} + \theta_{d_{m,k}} + \theta_{w_m} + \theta_{o_{m,k}})} \quad (2.2)$$

In (2.2), **s** is the SMSE waveform represented in the frequency domain, k is the symbol number, and m denotes the m^{th} of N_{F} spectral components in the bandwidth of interest. The notation θ_x represents the phase component of x. It is worth noting that the framework accounts for multiple user pair, antenna or channel realizations by expanding the complex-variable vectors into matrices with one row for each realization. However, this thesis is only concerned with a single user pair realization and will thus use the vector representations.

The framework's complex-variable vectors are described in more detail below.

- a represents the spectral components assigned for use by the radio. In a full CR application, this information would come from a network controller and the information would be determined *a priori*. A magnitude of 1 indicates that the spectrum is available; a 0 indicates that the spectrum is unavailable. For this application and in [30], a is assumed to be 1 for all m.
- For TDCS, u represents the spectral components that are currently not in use by other users. This information is obtained through the spectral sensing component of the radio. A magnitude of 1 indicates that the spectrum is unoccupied (available); a 0 indicates that the spectrum is occupied (unavailable). For MC-CDMA, u is dependent on the parameter F which determines subcarrier spacing, defined by:

$$f_m = m(F/T_{sym}) \quad (2.3)$$

  In (2.3), f_m represents the carrier frequency corresponding to frequency bin m and T_{sym} represents the intra-symbol transmission time, where m takes on a value of 1 to N_F (the number of frequency bins in the bandwidth of interest). For $F = 1$, the subcarriers are equally spaced at an interval equal to the symbol rate. Also, for $F = 1$, MC-CDMA has a similar structure to OFDM except for transmitting the same symbol across all subcarriers (as opposed to transmitting a different symbol on each subcarrier as done in OFDM). In this thesis, it is assumed that $F = 1$, so $u_m = 1$ for all m for MC-CDMA.

- c represents the code at each frequency component. In MC-CDMA, the phase component of c can be either 0 or π (corresponding to a magnitude of 1 or -1) in order to

accomplish spectral spreading. In TDCS, the phase component of *c* takes on one of 2^r values from 0 to 2π, providing noise-like correlation properties to the waveform.

- *d* represents the data modulation being applied to the symbol being transmitted. This vector can take on different values based on the digital modulation scheme being employed. For example, in a QPSK scheme *d* can take on the following values: 1+*j*, -1+*j*, 1-*j*, -1-*j* (assuming Gray coding is used). For OFDM, *d* varies with the spectral component *m* in addition to the symbol number *k*. For MC-CDMA and TDCS, *d* varies only with the symbol number *k*.
- *w* represents a windowing function. This function may be used for spectral shaping if desired. For this application and in [30], *w* is assumed to have a magnitude of 1 for all *m*.
- *o* is a phase-only component vector used for providing orthogonality between users in the context of a carrier interferometry scheme. According to [30], for OFDM, MC-CDMA and TDCS, the phase component of *o* is assumed to be 0 for all *m* and *k*.

Operation	**TDCS**	**MC-CDMA**
Data Modulation	MPSK, MQAM relies on *k*	MPSK, MQAM relies on *k*
Coding	**c = 1** $\theta_{c_m} \in [0, 2\pi]$	**c = 1** $\theta_{c_m} \in \{0, \pi\}$
Windowing	**w = 1** $\boldsymbol{\theta_w = 0}$	**w = 1** $\boldsymbol{\theta_w = 0}$
Orthogonality	$\boldsymbol{\theta_o = 0}$	$\boldsymbol{\theta_o = 0}$
Frequency Assignment	**a**	**a**
Frequencies Used	u_m depends on spectrum sensing	u_m depends on *F*-parameter

Table 2.1 – Overview of the SMSE framework variable instantiations for TDCS and MC-CDMA [30]

Table 2.1 summarizes the variable instantiations used to generate the waveforms considered in this thesis. Again, the letter notation for each vector (**c, w, a,** etc.) represents the magnitude of the complex variable vector, while the notation θ represents the phase of each vector.

Once the frequency domain vector is generated from the six variable instantiations (2.2), an inverse Fourier transform is used to generate the time-domain SMSE waveform (2.4) [30].

$$s_k[n] = \frac{1}{N_f} Re\left\{ \sum_{m=0}^{N_F-1} a_m u_m c_m d_{m,k} w_m e^{j(2\pi f_m t_n + \theta_{c_m} + \theta_{d_{m,k}} + \theta_{w_m} + \theta_{o_{m,k}})} \right\} \qquad (2.4)$$

N_F, in addition to being the number of frequency bins in the bandwidth of interest, is also the length of the inverse Fourier transform. In practical applications, N_F will be a power of 2 in order to take advantage of Fast Fourier Transform (FFT) algorithms.

At the receiver, following synchronization a Fourier transform operation is performed to transform the signal into the frequency domain. Here, decoding and demodulation will be performed. The correct values for *a, c, w,* and *o* are assumed to be known at the receiver a priori. *u* must be determined by the receiver for spectral sensing. In [30] it is assumed that the transmitter and receiver observe identical spectra and thus generate identical *u* vectors. The values for *a, c, w, o,* and *u* are removed from the spectral components of the received signal. Then, the spectral components are averaged over the bins over which data is spread (1 bin for OFDM, N_F bins for MC-CDMA and TDCS), resulting in a complex value. This averaged value is compared to a constellation of *M* possible symbol values to determine the most likely symbol received.

2.4. DSP as Implementation Platform for Software-Defined Radio

Digital signal processors have been used to implement a wide variety of telecommunications applications, including spread spectrum radios [36]. Haker used a Texas Instruments DSP as an implementation platform for a TDCS transmitter/receiver pair [11]. Due to the sampling rate limitations of the DSP, this implementation was limited to an acoustic

channel instead of an RF channel. Indeed, this is often a limitation of the DSP when it comes to communications applications, leading to many developers selecting an FPGA due to the higher processing rates that are possible [31]. However, if the bandwidth of the RF signal is less than the instruction rate of the DSP, an analog RF front-end can be used to demodulate the RF signal to baseband at a sampling rate that can be processed by the DSP [36].

Zhou et al have used the USRP and GNU software-defined radio as an implementation platform for cognitive radios based on the SMSE framework [41]. The USRP hardware consists of an FPGA used for modulating/demodulating the signal between baseband and RF, and it is connected to a PC via a USB interface. The code describing the operation of the software-defined radio is written and run on the PC as opposed to on a digital signal processor. This GNU SDR-based implementation offers several benefits compared to a DSP implementation. It allows software developers to access the RF spectrum by using general-purpose computers and popular programming languages in a high-level development environment [24]. However, this architecture may not be appropriate for smaller devices or embedded applications where general-purpose processors are not available. It is believed that this thesis will represent the first formal implementation of a SMSE framework-based transmitter/receiver pair on a digital signal processor or embedded platform.

3. Design Flow

When implementing DSP algorithms on hardware, the design usually begins as an implementation in a high-level programming language (MATLAB and C/C++ are commonly chosen). This allows the designer to explore algorithmic design decisions without having to worry about the details of a low level implementation. This initial implementation makes use of floating point precision and various tools and libraries that may be available in the high level design environment. While floating point computation offers virtually unlimited range and precision, it is impractical for real-time hardware implementation due to speed requirements and the size of the hardware that would be required. Many popular implementation platforms for DSP algorithms, including digital signal processors and FPGAs, are able to perform fixed point computations at a much faster rate and with lower power consumption [7]. As a result, the floating point simulation is not sufficient to study the precise behavior of the algorithm running on the hardware.

After the functionality of the floating point implementation is verified, the numerical values in the program are quantized to fixed-point values based on the numerical precision allowed by the processor of choice. Also, the high level libraries and abstractions used in the floating point implementation are replaced with less abstract versions that are specific to the final processor of implementation. The functionality of the implementation is confirmed once again to analyze the effect of the quantization on the algorithm's performance [7].

This chapter gives an overview of the different milestones in the design flow that allowed the SMSE transmitter and receiver to come into fruition. A description of the developed floating point simulation program is presented first. The changes necessary to convert the floating point simulation into a fixed point implementation are discussed. Finally, the changes necessary to develop the final implementation from the fixed point simulation are described.

3.1. Floating Point Simulation

The first step in implementation was to develop what is known as the floating point simulation. This simulation was written in MATLAB and makes full use of the flexibility of the MATLAB language. Numbers are represented using 64-bit floating point precision. Specialized

MATLAB functions for the fast Fourier transform, random number generation, and cross-correlation are used to simplify the implementation.

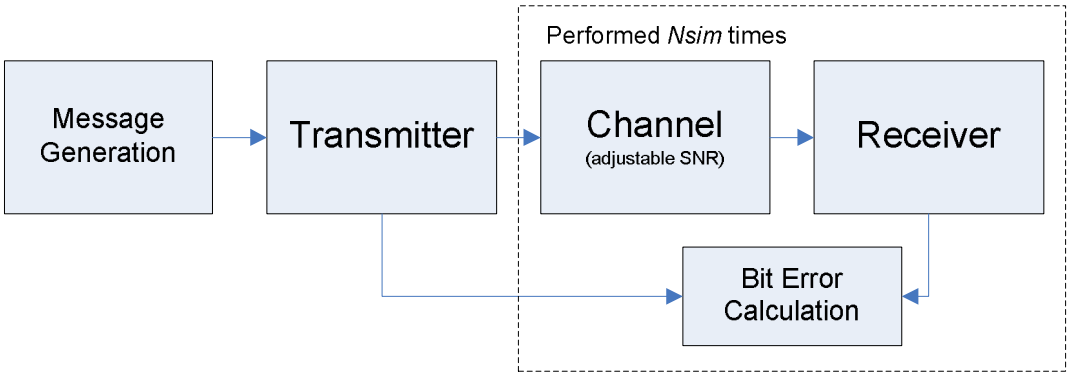


Figure 3.1 – SMSE Simulation Overview

The simulation is run from a MATLAB script which calls separate functions for the transmitter, channel and receiver. The simulation script allows the user to define the operating variables for the simulation, including:

- The number of Monte Carlo simulations to run (*Nsim*)
- The number of samples to transmit per simulation (*Nsamples*)
- The number of samples per symbol (*Nf*)
- The number of symbols to transmit (defined as *Nsamples/Nf*)
- A range of E_s/N_0 values over which to test (*awgn_EsN0*)
- The synchronization keyword: a sequence of bits that the receiver will use to synchronize with the transmitted message (*synch_key*)
- Channel type (an ideal pass-through, AWGN, or AWGN plus narrowband interferers)
- The number, position and power of interferers in the channel
- Code vectors (*c*) for both MC-CDMA and TDCS

An overview of the floating point simulation structure is shown in Figure 3.1. After defining the variables, the simulation begins by generating a vector of *Nbits* random bits. This is done using MATLAB's *rand* function. The synchronization keyword is then placed within this vector at a specified distance from the beginning. This will allow the receiver to search for the

synchronization keyword in the received signal in order to perform message and symbol synchronization. The bit vector is then passed to the transmitter function, which generates a sample output vector describing the SMSE signal to be transmitted. Note that the output vector of the transmitter will have a length of *Nf * Nbits* samples.

The output of the transmitter is then passed through the channel and receiver *Nsim* times for each E_s/N_0 value in *awgn_EsN0*. If a false synchronization occurs, that particular transmission is repeated until the synchronization is successful. The simulation keeps track of the number of correct and incorrect synchronizations recorded. The receiver produces a vector of the received bits, which is compared to the originally transmitted bit vector to determine the bit-error rate. The mean of the bit-error values obtained at each E_s/N_0 value is computed and used to plot the simulated bit-error curve alongside the theoretical bit-error curve for BPSK, determined by the equation:

$$P_b = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = \frac{1}{2}erfc\left(\sqrt{\frac{E_s}{N_0}}\right) \tag{3.1}$$

where E_s/N_0 is the signal-to-noise ratio and *erfc(x)* denotes the complementary error function. This process is performed for MC-CDMA and then repeated for TDCS. The variable *scheme* is used to toggle the transmitter and receiver between each method.

When simulating TDCS, the *u* vector for transmitter and receiver is determined a priori based on the number and position of interferers in the channel. This differs from the hardware implementation which uses a spectrum estimation method at the transmitter and receiver to determine the usage vector. The hardware implementation also uses the output of the spectrum estimation to select whether MC-CDMA or TDCS is to be used.

3.2. Fixed Point Simulation

To convert the floating point simulation into a fixed point version, several changes needed to be made. The goal was to convert the abstract implementation into one that was still developed at a high level, but that incorporated many lower level aspects. Then, converting the fixed point simulation into hardware implementation would be a more straightforward process.

3.2.1. Scaling and Quantization

The floating point simulation made use of floating point number representation, allowing for a large amount of range and precision. To develop the fixed point simulation, it was necessary to quantize numerical values to the integer values required by the fixed point processor. The range and precision allowed on a DSP are determined by several factors, but a common limiting factor is the resolution of the analog-to-digital converters and digital-to-analog converters. The DSP Star AD/DA modules used in this project have a resolution of 12 bits for both A/D and D/A, meaning they can represent twos-complement integer values from -2^{12-1} to $2^{12-1}-1$, or -2048 to 2047. The floating point simulation uses values between -1 and 1 for all variables and for the transmitted waveform. When transforming the floating point simulation to fixed point, these values are scaled to have a maximum value of 2047 and a minimum value of -2048. This results in variables that have a length of 12 bits with 11 bits used for fractional precision. The difference between the floating point and fixed point implementations is shown in Figure 3.2. Note the different scale between the floating point and fixed point waveforms.

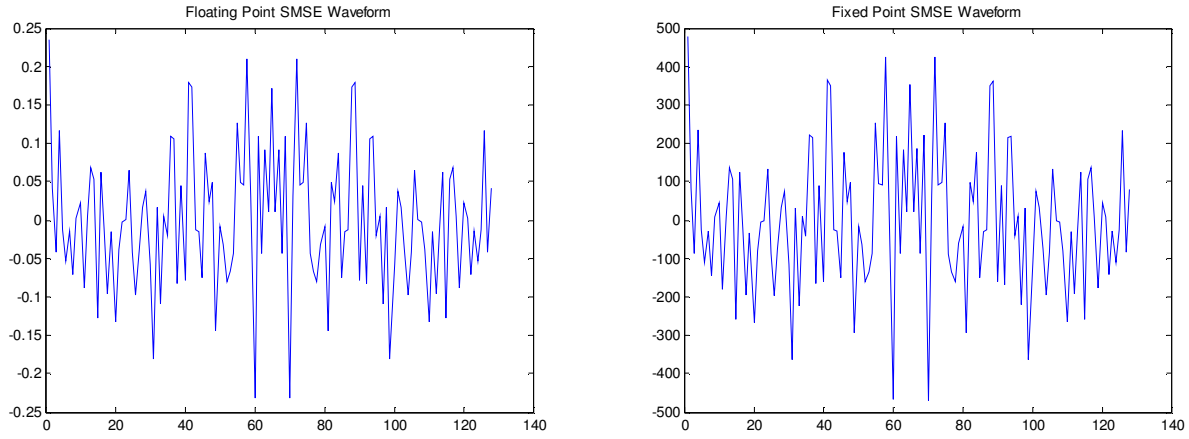


Figure 3.2 – Comparison between Floating Point and Fixed Point SMSE Waveforms

This change of scale affects not only the range of numbers allowed, but their precision. Values in the floating point simulation had virtually unlimited precision that was limited only by

the IEEE floating point standard. In the fixed point implementation, precision is limited to the value of the least significant bit:

$$2^{-11} = 4.883 \times 10^{-4}$$

The maximum deviation between any floating point value and its quantized value then becomes half of the precision, or 2.441×10^{-3} [34]. This difference between the desired floating point value and the quantized fixed point value is known as *quantization error*, and it may negatively impact the performance of an algorithm. This is why the floating point simulation is important, as it allows for analysis of how the hardware's quantization will affect the algorithm being developed.

3.2.2. Fixed Point Fast Fourier Transform and Cross-Correlation

The floating point simulation made use of MATLAB's built-in functions for computing the Fast Fourier Transform (FFT) of a vector and cross-correlation of two vectors. These functions are known within MATLAB as `fft` and `xcorr` respectively. Since these functions are not available on the DSP, different functions needed to be found to be incorporated into the fixed point simulation.

For the Fast Fourier Transform, the FFT code in C in [18] was adapted to be used in the hardware implementation. The function was changed from a 256 point FFT to a 128 point FFT. Also, since the C code was originally written for a floating point processor, some scaling was added to avoid overflows that would occur when implemented on the fixed point processor. In order to use the code in the fixed point simulation, it needed to be translated from C to MATLAB. This was done manually by hand. The results of the fixed-point FFT function were compared to those of MATLAB's built-in FFT function to ensure that the fixed-point function was implemented correctly.

To implement the cross-correlation function in the fixed point simulation, the C code in [3] was used after being translated from C to MATLAB. Like the FFT function, the results were compared to those of MATLAB's built-in cross-correlation function to verify that the fixed-point cross-correlation was correct.

3.2.3. Vectorization

One of the major benefits of implementing signal processing algorithms in MATLAB is the language's extensive use of vectorization. One can multiply two vectors together or find the maximum, minimum and mean value of a vector using one line of code. This can greatly aid in the manipulation and analysis of signals. No such intrinsic capability exists in a lower level language such as C; array operations must be performed element-by-element. This requires some changes when converting the simulation from floating point to fixed point. Instances of element-by-element multiplication of arrays need to be replaced by loops which performed each element's multiplication iteratively. Similarly, instances of finding the maximum and mean of an array were replaced by loops that performed the searches iteratively. After this conversion, the MATLAB code could be translated into C code line-by-line in a very straightforward fashion.

3.3. Hardware Implementation

For the most part, the fixed point simulation could be translated into the hardware implementation code with a few small adjustments. Since MATLAB arrays start with index 1 while C arrays start with index 0, this required adjustments to the array indexing used throughout the program. Also, some variables that were defined on-the-fly during simulation were hard-coded into header files to ease the computational burden on the hardware. These variables include the vector of bits to be transmitted, the twiddle factors for use by the FFT function, and the code vectors for MC-CDMA and TDCS.

4. Transmitter Implementation

The following 3 chapters describe the hardware implementation of the SMSE communications system in detail. This chapter begins with an overview of the system and then describes the implementation of the transmitter. Chapter 5 describes the implementation of the channel model, and Chapter 6 describes the implementation of the receiver.

4.1. System Overview

There are three major components to the SMSE communications system: the transmitter, channel model and receiver. The transmitter and receiver are implemented on separate Spectrum Digital DSKs each featuring a Texas Instruments C6416 digital signal processor. Each DSK also consists of a DSP Star AD/DA combo operating at 48 kHz. The channel is implemented on a Nallatech XtremeDSP development kit featuring a Xilinx Virtex-4 FPGA. The FPGA kit is located on a PCI card which is housed within a PC that is also used to control the transmitter DSK. A separate PC is used to control the receiver DSK. The DSKs are controlled by their respective PCs via a USB interface.

The transmitter DSK, channel FPGA, and receiver DSK are all interconnected via coaxial cable. The transmitter repeatedly sends one-second burst transmissions to the receiver via the channel model, which manipulates the signal based on the desired effect of the channel. The receiver performs processing and compares its estimate of the received bits to the transmitted bit sequence to determine the number of bit errors that occurred. A separate coaxial line is connected from the receiver to the transmitter. This is simply used by the receiver to signal the transmitter to begin another burst transmission. It is not used for any kind of timing synchronization. The transmitter and receiver both perform spectrum sensing to determine which spread spectrum scheme (MC-CDMA or TDCS) to use and which spectrum segments are occupied.

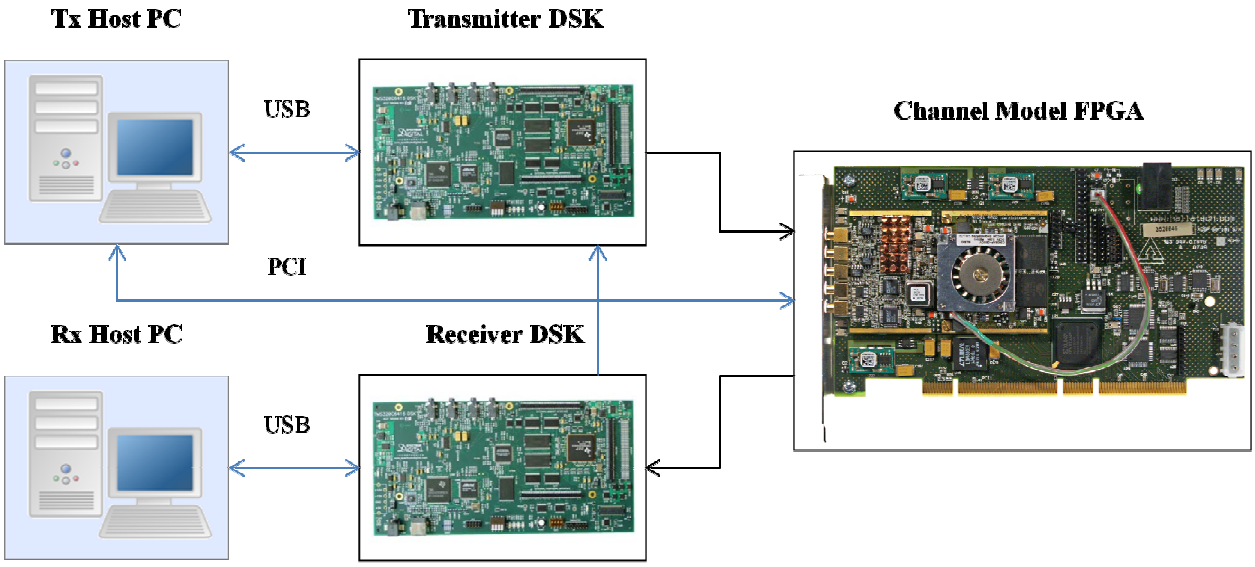


Figure 4.1 – SMSE System Overview

Figure 4.1 shows the system hardware, SMSE signal flow (black lines), and control interfaces (blue lines).

4.2. Transmitter

Figure 4.2 shows the overall signal flow of the transmitter. Spectrum estimation is used to determine the available and occupied spectrum and to determine whether MC-CDMA or TDCS should be used. The five complex-valued SMSE framework vectors (*a*, *u*, *w*, *o*, and *c*) are then generated based on the transmission scheme selected. These vectors are then Hadamard multiplied together to form the frequency domain vector describing the SMSE waveform; in the literature, this vector is referred to as $s_k[m]$ for the *k*th symbol being transmitted:

$$s_k[m] = \{a[m]\, u[m]\, w[m]\, c[m]\, o[m]\}_{m=0}^{N_F-1} \qquad (4.1)$$

To generate the time domain symbol waveform, this vector is passed through an inverse Fourier transform to create the discrete time-domain version of the SMSE symbol waveform, $s_k[n]$. This waveform is then modulated by the data bit being transmitted (either +1 or -1). The modulated waveforms for successive bits are concatenated and stored in a buffer for transmission. The one-second buffer containing the complete SMSE transmission $s[n]$ is transmitted by being passed

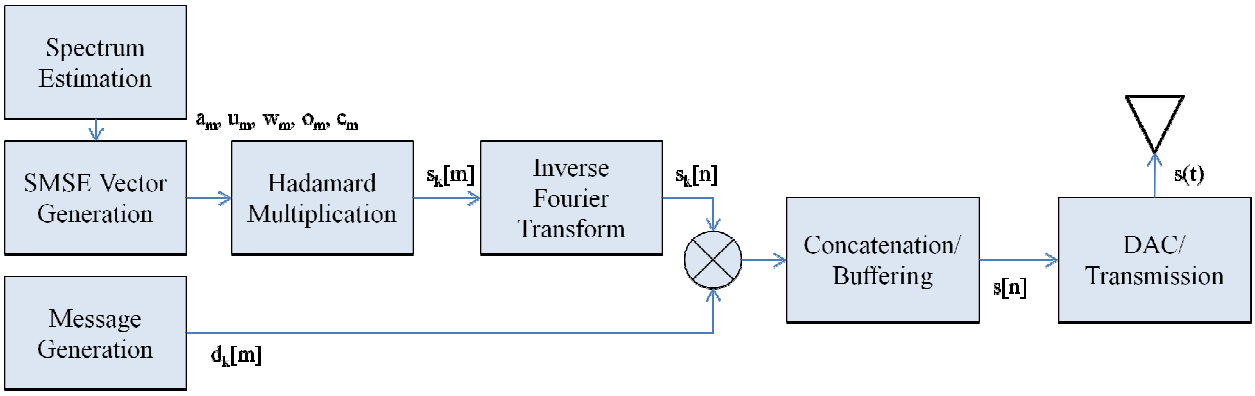


Figure 4.2 – SMSE Transmitter Block Diagram

through a digital-to-analog converter to generate the analog transmission *s(t)*. The following sections will describe the implementation of each segment of the transmitter in greater detail.

4.2.1. Message Generation

At the transmitter, a sequence of bits is prepared for transmission. In the MATLAB simulations, this was done by using the program's built-in random number generator. A 16 bit predefined sequence is then placed within the random sequence; this synchronization sequence is used by the receiver to find where the transmitted message begins. The synchronization sequence is derived from a 13-bit Barker sequence and was chosen due to uniform correlation sidelobes and a well-defined peak [2]. Since 48000 samples are being transmitted and each SMSE symbol contains 128 samples, 375 total bits are transmitted. The synchronization sequence is placed in the message sequence 25 bits from the beginning. This means 350 bits are received at the receiver, including the synchronization sequence.

To ease the computational burden of the hardware, a bit sequence generated by the MATLAB simulation is hard-coded and placed into a C header file. This header file is available to both the transmitter and receiver, which allows the receiver to determine the number of bit errors that occur following the demodulation process.

4.2.2. Spectrum Estimation

In order to perform TDCS transmissions, the spectrum must be sampled to determine the location of any interference and notch out that part of the spectrum in the TDCS waveform. In this implementation, the spectrum is sampled using Bartlett's method [25]. A buffer is used to store samples of the channel environment from the FPGA channel model, captured through the DSP Star analog-to-digital converter. Periodograms are performed on several subsequences (of length 128) of this buffer, and the periodogram results are averaged to yield a PSD estimate. Due to the high variability and poor estimates that a single periodogram may obtain, averaging several results is necessary to reduce the variability and obtain a more accurate estimate [32].

Once the spectrum estimate is obtained, it is used to determine the likely location of interferers. The TDCS transmitter can be configured to notch out 0, 1, 2 or 3 interferers. To do this, the element of the spectrum estimate vector with the highest magnitude is found. That bin and the two bins adjacent to it are notched out (replaced with zero). This process is repeated one or two times until the specified interferers are notched out. For each interferer found, 3 bins will be notched out. When creating the frequency use vector to be used in TDCS waveform generation, bins that were notched out are assigned a value of zero, along with the bins corresponding to DC and the Nyquist frequency. All other bins are assigned a value of one. Note that since the frequency vectors are at baseband, the first half of the vector represents the positive frequencies from 0 to $F_s/2$ while the second half of the vector is reflected and represents the negative frequencies from $-F_s/2$ to 0. This means that in order to generate a real waveform in the time domain, the bins that are notched out in the first half of the vector must also be notched out in the corresponding bins in the second half of the vector. Figure 4.3 shows a sample frequency use vector generated by the spectrum estimation block for a case of 3 interferers. Note how the second half of the vector is a reflection of the first half of the vector, and that the bins corresponding to DC and the Nyquist frequency are notched out.

In the case of TDCS, it is understood that there is the possibility of a spectrum mismatch, where the transmitter's estimate of the channel spectrum (and notching of the TDCS waveform) differs from that of the receiver. The transmitted TDCS waveform will then be different from the reference waveform computed at the receiver, leading to reduced accuracy of the correlation receiver. This phenomenon was studied in [11], and it was determined via simulation that when spectrum estimates are mismatched there will be a degradation in bit error performance of less

than 1 dB in AWGN. Existing implementations of TDCS have either assumed that the transmitter and receiver will generate identical spectrum estimates, or they have set a hard limit on the number of subcarriers that can be notched out as is done in this SMSE implementation. Haker used a similar procedure for notching out interferers in his study of a TDCS hardware implementation [12]. However, despite the transmitter and receiver sharing the same straightforward method for notching out subcarriers, there is still a possibility that a mismatch will occur.

For clarity, all TDCS bit error curves presented in subsequent chapters represent situations where the transmitter and receiver obtain identical spectrum estimates and subcarrier notching. This spectrum estimation method is not assumed to be optimal, and its optimality is not studied in this thesis. Future implementations may need to explore spectrum estimation and notching methods in greater detail. In any case, to ensure reliable communication a practical TDCS implementation will need to have a handshaking process by which transmitter and receiver exchange information about the spectra they are observing and decide upon the best transmission waveform.

The spectrum estimation block is also used to determine which spread spectrum scheme should be used. The presence of narrowband interference is used as the criterion for selecting the scheme. If narrowband interferers are found in the channel, TDCS is chosen; otherwise MC-CDMA is chosen. In order to estimate whether or not interference is present, the statistical mean and standard deviation of the channel PSD estimate are calculated. If the standard deviation of the PSD estimate is a sufficient percentage of the mean, it assumed that interferers are present; otherwise, there are no interferers present. This is a simple classification method drawing from the fact that the addition of strong tone interference will increase the variance of the PSD over the relatively low variance observed with just a flat noise floor.

4.2.3. SMSE Vector Generation

The SMSE transmitter, at its core, consists of the six frequency-domain SMSE vectors *a, u, w, o, d,* and *c*. Combinations of these vectors, followed by an inverse Fourier transform operation, are used to generate the time-domain waveforms corresponding to the 2 multicarrier modulations described in this thesis. In this implementation, the vectors *a, w,* and *o* are set to 1

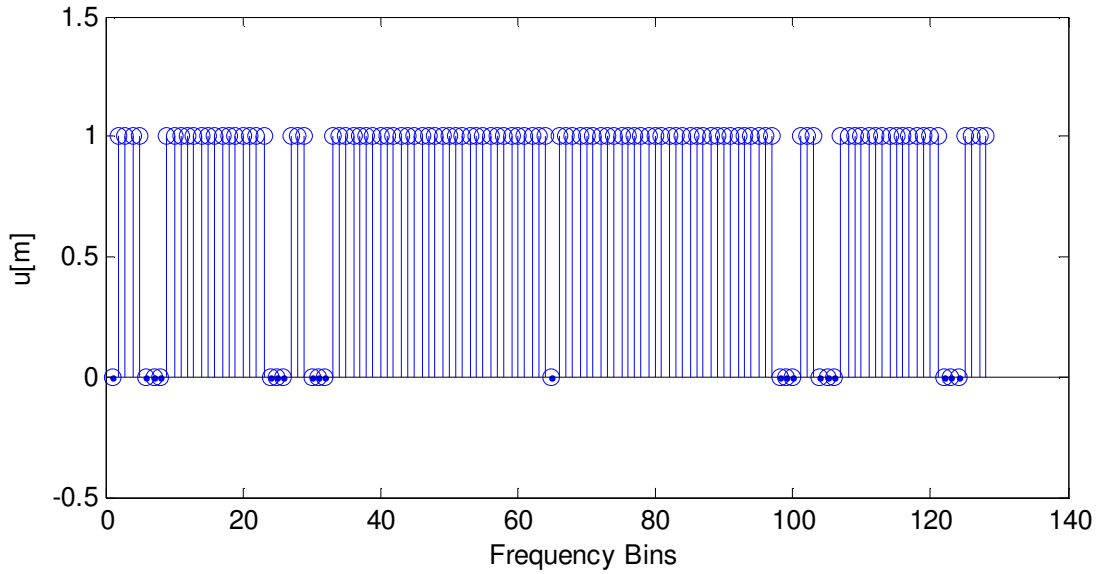


Figure 4.3 – Frequency Usage Vector for Three Interferers Scenario

for all subcarriers and are not dependent on the type of waveform being transmitted. These 3 vectors represent frequency assignment, spectral windowing and an orthogonality term, respectively [30]. For the BPSK modulation being used in this implementation, *d* can take on a value of +1 or -1, and it is the same across all subcarriers for each symbol. To simplify the implementation, the modulation occurs after the modulating waveform has been generated via the inverse Fourier transform. This puts the implementation in line with previous implementations of the TDCS [4] [11]. The coding vector (*c*) and frequency usage vector (*u*) are different depending on whether MC-CDMA or TDCS is being transmitted.

4.2.3.1. Coding Vector

The *c* vector is the coding vector that is used to apply coding to each subcarrier in the frequency domain. The vector is defined differently depending on whether MC-CDMA or TDCS is being used.

MC-CDMA accomplishes spreading in the frequency domain and achieves multiple access capability by toggling the phase of each subcarrier between 0 and π, equivalent to toggling the amplitude between +1 and -1. The real component of the coding vector therefore becomes +1 or -1 while the imaginary component is zero for all subcarriers. In this implementation, a Walsh-Hadamard code of length 64 is used as a spreading code in the

frequency domain. Walsh-Hadamard codes are popularly used in spread spectrum systems because of their orthogonality [14]. When properly aligned, all Walsh-Hadamard codes of a particular length are orthogonal to each other and produce a zero cross-correlation when they are aligned correctly. When each user pair is assigned a unique Walsh-Hadamard code, this allows for transmissions from multiple users to be separated out at the receiver with no interference between them [10]. Although Walsh-Hadamard coding is used in this implementation, Roberts et al [30] found there to be no difference in BER performance when random coding was used instead of a Walsh-Hadamard code. Walsh-Hadamard coding is only used as a more realistic example to demonstrate the capabilities of the SMSE framework in a possible multiple access scenario.

TDCS uses the coding to achieve multiple-access and to randomize the phase of the spectral components, making the TDCS waveform appear more noise-like [4]. As described in section 2.2.4, each subcarrier is assigned one of 2^r phase values equally spaced around the unit circle, based on the output of a linear feedback shift register (pseudorandom number generator). In this implementation, r is set to 3, so there are 8 possible phase values: 0, $\pi/4$, $\pi/2$, $3\pi/4$, π, $5\pi/4$, $3\pi/2$, and $7\pi/4$. In order to achieve these phase values around the unit circle, the real and imaginary components of the coding vector must be set to the values shown in Table 4.1.

Phase	**Real Component**	**Imaginary Component**
0	1	0
$\pi/4$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
$\pi/2$	0	1
$3\pi/4$	$\frac{-\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
π	-1	0
$5\pi/4$	$\frac{-\sqrt{2}}{2}$	$\frac{-\sqrt{2}}{2}$
$3\pi/2$	0	-1
$7\pi/4$	$\frac{\sqrt{2}}{2}$	$\frac{-\sqrt{2}}{2}$

Table 4.1 – Phase Values Used for TDCS

An example of the real component, imaginary component, and phase for the coding vector of a TDCS symbol is shown in Figure 4.4. Note that the phase is conjugate symmetric about the Nyquist frequency bin; that is, the phase values in the second half of the vector are a reflection of the first half about the Nyquist frequency bin, multiplied by -1. This is necessary due to the implicit use of the Hilbert transform in describing baseband signals in the frequency domain. In order to generate a real-valued time-domain signal via the inverse Fourier transform, the frequency domain vector must be conjugate symmetric [11]. Also note that on the phase plot, the phase values $5\pi/4$, $3\pi/2$, and $7\pi/4$ are represented as $-3\pi/4$, $-\pi/2$, and $-\pi/4$ respectively.

Table 4.2 summarizes the real and imaginary components of the coding vector for each SMSE waveform. For MC-CDMA and TDCS, instances of the coding vector are generated from the MATLAB simulation, hard-coded and stored in the C header file that is available to both transmitter and receiver. This eases the computational burden at both the transmitter and receiver.

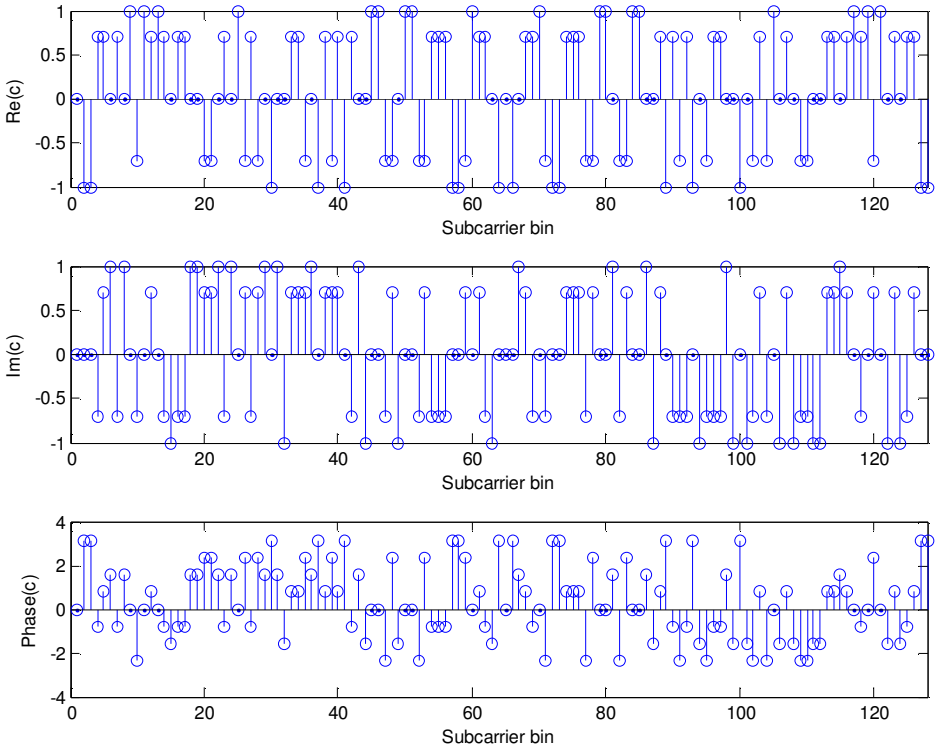


Figure 4.4 – TDCS Coding Vector Instantiation

Scheme	Real Component	Imaginary Component	Phase
MC-CDMA	± 1	0	0 or π
TDCS	$\pm 1, \pm \frac{\sqrt{2}}{2}$	$\pm 1, \pm \frac{\sqrt{2}}{2}$	$\frac{2\pi}{8} i$ for $i = 0 \ldots 7$

Table 4.2 – Coding Vector Values for SMSE Waveforms

4.2.3.2. Frequency Usage Vector

u is defined differently depending on whether MC-CDMA or TDCS is being used. For MC-CDMA, *u* is dependent on the *F* parameter (subcarrier spacing) as described in section 2.3. Subcarrier spacing is assumed to be *F*=1 for this application, so when MC-CDMA is the mode of transmission, *u* is set to one for all elements of the vector (except for the elements corresponding to DC and the Nyquist frequency, which are set to zero). For TDCS, *u* is the frequency use vector determined by the spectrum estimation method described in the previous section. In either case, these vectors are calculated on the fly by transmitter and receiver and are not hard-coded into the implementation.

4.2.4. Hadamard Multiplication

To generate the SMSE symbol waveform, a frequency domain vector $s_k[m]$ is generated by Hadamard multiplying the five SMSE framework variable vectors. Hadamard multiplication corresponds to element-by-element array multiplication [30]:

$$s_k[m] = \{a[m]\, u[m]\, c[m]\, w[m]\, o[m]\}_{m=0}^{N_F-1} \qquad (4.2)$$

Note that *c* consists of both real and imaginary components, and that *o* consists only of an imaginary component. In the implementation, the real and imaginary component vectors are multiplied separately to create real and imaginary baseband frequency-domain vectors describing the SMSE symbol waveform. These vectors will be passed to the inverse Fourier transform which will generate the time-domain SMSE symbol waveform.

Figures 4.5 and 4.6 show instances of the frequency-domain vectors describing MC-CDMA and TDCS waveforms, respectively. These plots were generated from the fixed point simulation. Note that the vectors are conjugate-symmetric about the Nyquist bin (bin 65), and that the bins associated with DC and the Nyquist frequency (bins 1 and 65) have zero magnitude. This is necessary due to the nature of the Hilbert transform in describing baseband signals [11]. Also note that the values have been scaled to fit inside a 12 bit word length.

4.2.5. Inverse Fourier Transform and Modulation

To generate a time domain SMSE waveform, an inverse Fourier transform is performed on the real and imaginary frequency domain vectors described in the previous section. As described in section 3.2.2, a C implementation of the Fast Fourier Transform from [18] was used in the hardware implementation. This function can be used to compute the inverse Fourier transform by adding a scaling factor of $1/N$ to the FFT result and conjugating the twiddle factors used in the FFT algorithm. This process yields the time-domain real-valued signal vector $s_k[n]$.

Plots of time-domain MC-CDMA and TDCS waveforms are shown in Figures 4.7 and 4.8. As BPSK signaling is being used, these time-domain waveforms are multiplied by either +1 or -1 before transmission, corresponding to each bit being transmitted. To allow for comparison between the simulation and the hardware implementation, Figures 4.9 and 4.10 show the same MC-CDMA and TDCS waveforms as they were generated on the hardware. As expected, the results of the hardware implementation exactly match those of the fixed point simulation.

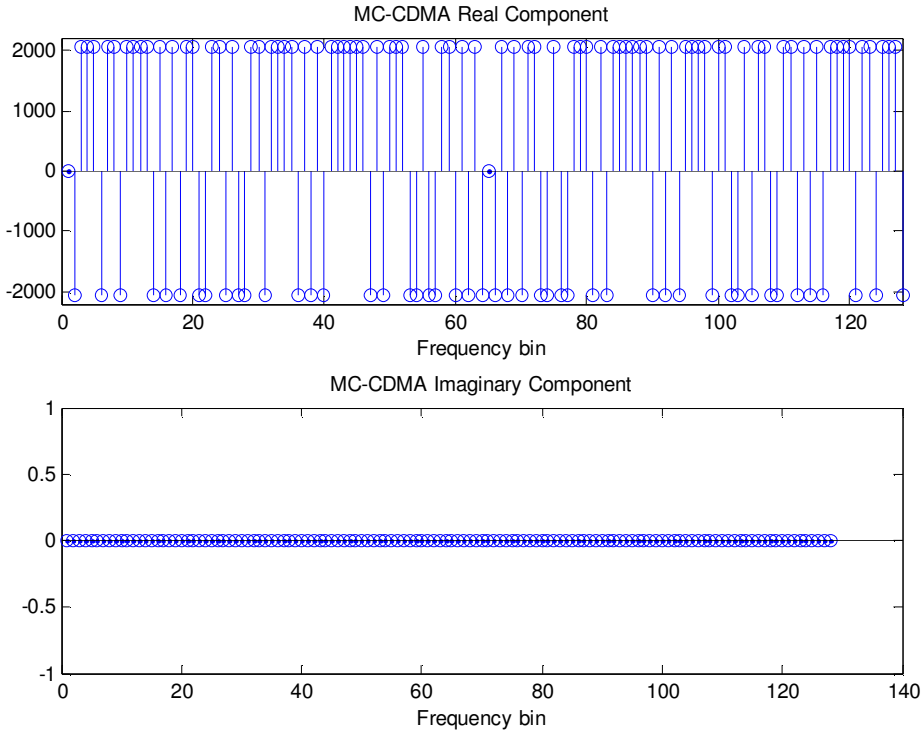


Figure 4.5 – MC-CDMA Frequency Domain Vectors

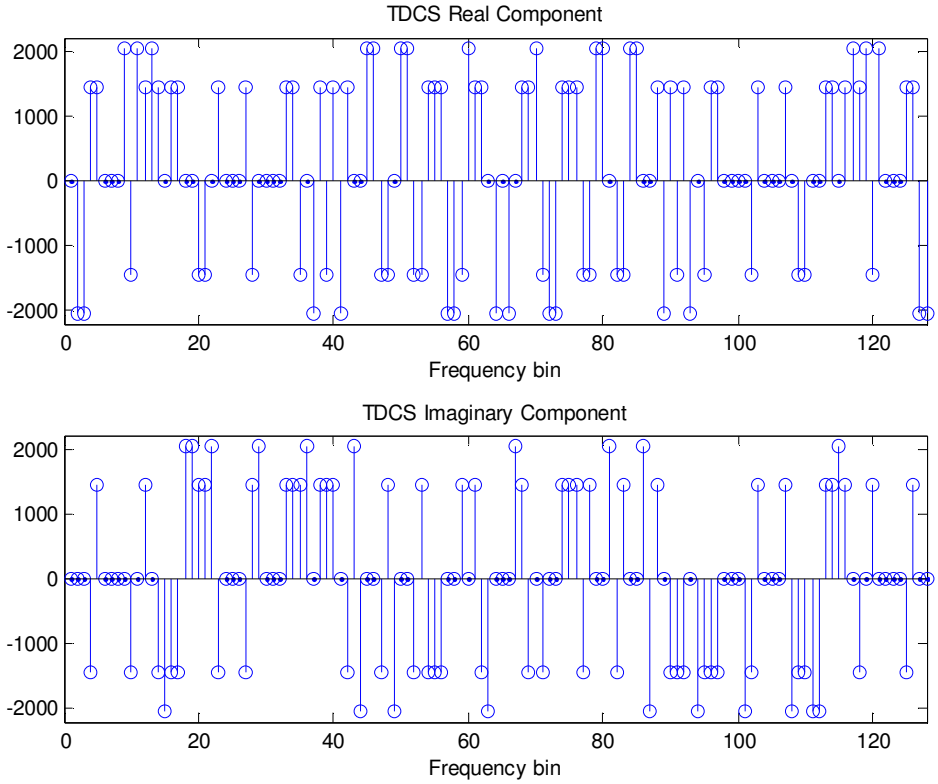


Figure 4.6 – TDCS Frequency Domain Vectors

4.2.6. Concatenation, Buffering & Transmission

Each of the 375 bits to be transmitted is modulated onto a symbol waveform with a length of 128 samples. These samples are then concatenated and stored into a one-second buffer for transmission. Once the buffer is full, the transmitter waits to receive a pulse from the receiver indicating it is ready for reception. Once this pulse is received, the transmitter begins to transmit the data stored in buffer via the digital-to-analog converter. The transmitter then returns to the ready state of waiting for a pulse from the receiver.

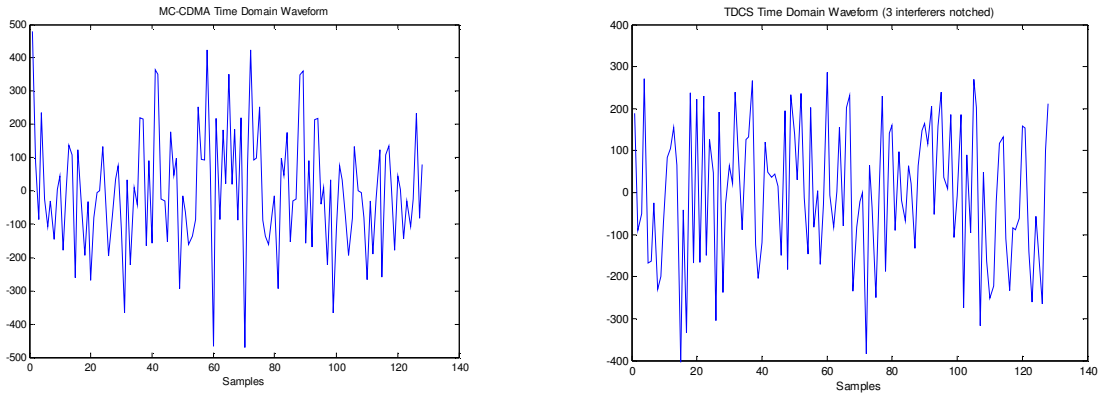


Figure 4.7 and 4.8 – MC-CDMA and TDCS Time Domain Waveforms after IFFT

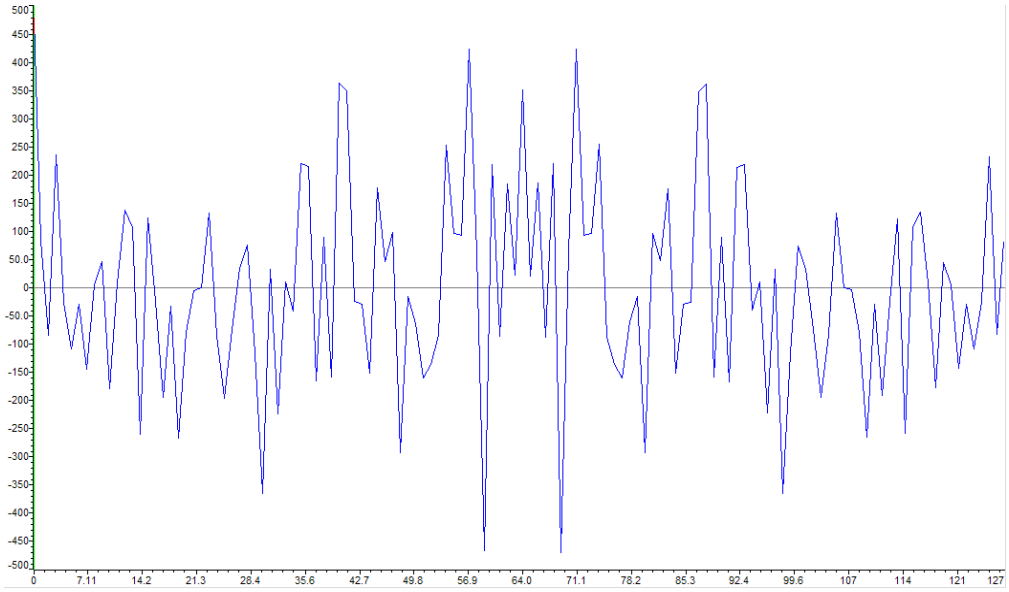


Figure 4.9 – MC-CDMA Time-domain Waveform from Hardware Implementation

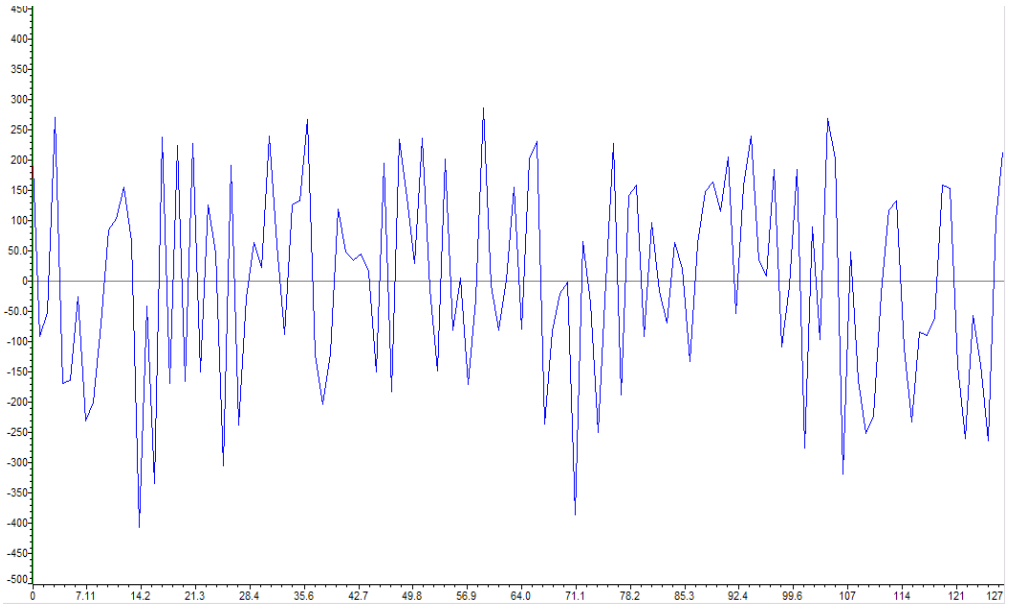


Figure 4.10 – TDCS Time-domain Waveform from Hardware Implementation

5. Channel Model Implementation

In order to study the performance of the SMSE transmitter and receiver, it is necessary to evaluate the system by attempting transmissions through a realistic, non-ideal communications channel. Since this transmitter/receiver implementation operates at baseband, transmission cannot occur wirelessly over the air. It was necessary to develop a new means to simulate the effects of the channel on the communications signal. Although there are numerous channel models that receive attention in the literature (flat fading, frequency selective fading), for this work an additive white Gaussian noise (AWGN) channel with narrowband interference was chosen. The AWGN allows for study of the system's capability with the metric being bit error probability as a function of signal-to-noise ratio (SNR). The addition of narrowband interference allows for testing of the effectiveness of TDCS, a communication scheme that possesses the ability to avoid narrowband interference.

5.1. Nallatech XtremeDSP Development Kit

The channel model was implemented on a Nallatech XtremeDSP development kit containing a Xilinx Virtex-4 FPGA (XC4VSX35) [39]. The kit comes equipped with two analog-to-digital converters (operating at up to 105 MHz) and two digital-to-analog converters (operating at up to 160 MHz). The kit is housed on a card that interfaces with a PC via a PCI interface, allowing for programming and control of the kit. The FPGA, ADCs and DACs make the kit ideal for communications applications. One of the ADCs is used as input to the channel model (from the transmitting DSK), while a DAC is used as the output of the channel model (sent to the receiving DSK).

5.2. System Generator Environment

The design for the channel model was developed using Xilinx System Generator [38]. System Generator is an add-on for The MathWorks' Simulink product that permits developing FPGA designs in a graphical interface. To use Simulink, the designer places blocks that perform various functions (math, signal processing, signal routing) into a design and connects them together in order to perform the desired functions of a system. System Generator extends this capability by providing its own special set of blocks that describe Xilinx intellectual property

designs for FPGA. FPGA designs can then be accurately simulated and analyzed within the Simulink environment. Examples of Xilinx blocks that are used in the channel model include the BPSK AWGN Channel, Look-up Table, Random Number Generator and Mux (multiplexer). After the design is simulated, System Generator can then generate working Hardware Description Language (HDL) code or a bitstream for programming to the FPGA hardware.

A special capability of System Generator is hardware co-simulation. This allows for running a simulation simultaneously on the PC and the FPGA hardware. The FPGA hardware can then be controlled via the simulation, and the outputs of the hardware can be viewed within the simulation. This approach is used for the channel model. The simulation is used to control the channel model (adjusting the signal-to-noise ratio, enabling interferers, etc.), while the actual channel model is implemented on the FPGA hardware. The maximum operating frequency of the hardware co-simulation is 40 MHz, which is more than sufficient for this application.

5.3. Additive White Gaussian Noise Channel

In an AWGN channel the received signal *r(t)* can be described as

$$r(t) = s(t) + n(t) \tag{5.1}$$

where *s(t)* is the transmitted signal and *n(t)* represents a white Gaussian random process with zero mean and a power spectral density of $N_0/2$. The signal-to-noise ratio of the received signal is defined as the ratio of the received signal power (P_r) to the power of the received noise within the bandwidth of the transmitted signal, which can be written as $2B(N_0/2)$, where B is the bandwidth of the complex envelope of the transmitted signal. This leads to

$$SNR = \frac{P_r}{N_0 B} \tag{5.2}$$

which can be expressed in terms of the signal energy per symbol (E_s) as

$$SNR = \frac{P_r}{N_0 B} = \frac{E_s}{N_0 B T_s} \tag{5.3}$$

where T_s is the symbol period [10]. For this application, the bandwidth B of the baseband SMSE signal is 24 kHz, and the symbol period T_s is 1/375 Hz. Therefore:

$$SNR = \frac{E_s}{N_0 B T_s} = \frac{1}{64}\frac{E_s}{N_0} \tag{5.4}$$

Thus, for SNR to equal E_s/N_0, the noise power N_0 must be scaled by a factor equal to the number of positive frequency subcarriers (64). This is the approach that is taken in the MATLAB simulations in order to obtain proper curves for BER vs. E_s/N_0. For BPSK, BER as a function of E_s/N_0 can be written as [10]:

$$P_b = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = \frac{1}{2}erfc\left(\sqrt{\frac{E_s}{N_0}}\right) \tag{5.5}$$

This is the equation used for calculating the theoretical BER curves that appear in Chapter 7.

5.3.1. Hardware Implementation of AWGN

The AWGN channel is implemented on the FPGA using an intellectual property (IP) core provided by Xilinx [37]. The core generates Gaussian noise for a BPSK AWGN channel using the Box-Muller method and central limit theorem as described in [9]. The desired SNR may be provided as an input to the core in the range from 0.0 dB to 15.9 dB in increments of 0.1 dB. For the core, the SNR is defined for BPSK with unit symbol energy ($E_s = 1$). The MC-CDMA and TDCS waveforms being transmitted may not have unit symbol energy. To account for this, the output of the AWGN core is scaled by a factor of $\sqrt{E_s}$, where E_s is the calculated symbol energy of the MC-CDMA/TDCS waveform. This adjusts the variance of the random number distribution by a factor of E_s; the square root is necessary because the output must be scaled by the desired standard deviation, which is the square root of the variance. The symbol energy E_s is calculated *a priori* in the floating point MATLAB simulation for each waveform being transmitted. The SNR input to the AWGN block may then be adjusted to vary E_s/N_0 of the channel, thus enabling the generation of BER curves. The SNR and gain of the AWGN block can be adjusted while the channel model is running.

5.4. Narrowband Interference

Narrowband interference is added to the channel model to allow for analysis of the interference mitigation properties of TDCS compared with MC-CDMA. In the floating point

and fixed point simulations, this is accomplished by creating a frequency domain vector describing the location of interferers. In this vector, bins corresponding to the positive frequencies of the interference are assigned a value of 1i, and bins corresponding to the negative frequencies of the interference are assigned a value of -1i. All other bins are assigned a value of zero. An inverse Fourier transform is performed on this complex-valued vector to generate a time-domain real sinusoidal signal describing the narrowband interference. Figure 5.1 shows an example of the frequency domain and time domain vectors describing a scenario with interferers contained in frequency bins 6, 24 and 30.

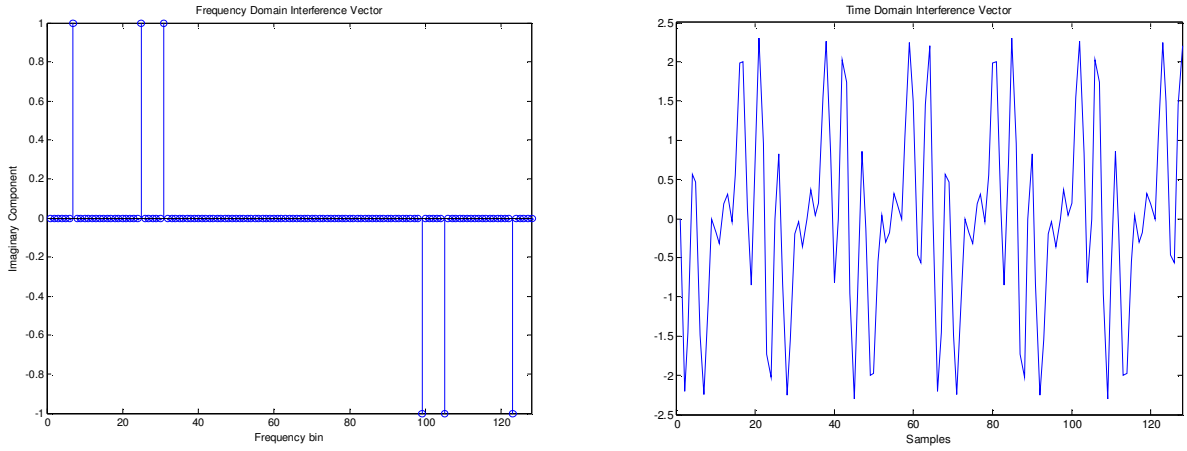


Figure 5.1 – Frequency Domain and Time Domain Interference Vectors

To simulate BPSK interference, each half of the time domain vector is multiplied by either a +1 or -1 (determined randomly) before being added to the channel. This results in BPSK interferers with a bit rate R of 48000/64 = 750 Hz. The minimum bandwidth of each interferer is then $3R/2$ = 1125 Hz [17]. This means that for TDCS to avoid transmitting in any area of the spectrum where interference is present, 3 frequency bins (centered at the center frequency of each interferer) must be notched out. This approach is taken in the spectrum estimation method described in section 4.2.2. Figure 5.2 shows the power spectral density of 3 BPSK interferers.

The amplitude of each sinusoidal interferer is adjusted for a desired signal-to-interferer ratio (SIR) based on the following equation, which relates the amplitude of a sinusoid to its power [33]:

$$A = \frac{\sqrt{2I_o}}{2} \tag{5.6}$$

where I_0 is the interferer power determined based on the desired SIR and symbol energy E_s:

$$I_0 = \frac{E_s}{SIR} \tag{5.7}$$

5.4.1. Hardware Implementation of BPSK Interferers

To implement the BPSK interferers in hardware, two main components were used: a sinusoid look-up table and a random number generator. Figure 5.3 shows the structure of the hardware implementation. The look-up table contains 128 values describing one cycle of a sinusoid. A counter, clocked at a rate corresponding to the frequency of the interferer divided by 128, increments the address to the look-up table, making the output of the look-up table correspond to a sinusoid at the frequency of the interferer. The output of the look-up table is then multiplied by either a +1 or -1, based on the output of a random number generator sampled at the desired bit rate of the interferer. This sinusoid is then multiplied by the desired interferer amplitude, which is determined *a priori* (via the MATLAB floating point simulation) and passed as an input to the hardware co-simulation. Finally, the output of the interferer block is selected by a multiplexer, allowing each individual interferer to be turned off and on.

The channel model is implemented with a total of four interferers from which to choose, centered on bins 6, 18, 24, and 30. The locations of these bins were chosen experimentally based on simulation results; it was found that these bin locations produce a noticeable impairment in the performance of MC-CDMA. Other bin combinations produce similar impairments, while a few bin combinations do not produce much impairment at all. The reason behind this was not studied in this thesis, but [13] offers some possible geometric explanations why some interferer locations produce different effects than others in multicarrier transmission schemes. The amplitude of the interferers can be adjusted during the co-simulation, and interferers can be turned off and on during the co-simulation as well. The locations of the interferers are fixed and cannot be adjusted during the co-simulation; a new hardware co-simulation block must be generated in order to adjust their position.

Figures 5.4 and 5.5 show screenshots of an oscilloscope displaying the PSD of the channel (running on FPGA hardware). The figures show the channel with 3 interferers and with 3 interferers plus noise, respectively. E_s/N_0 is 1 dB and the signal-to-interference ratio is -17 dB.

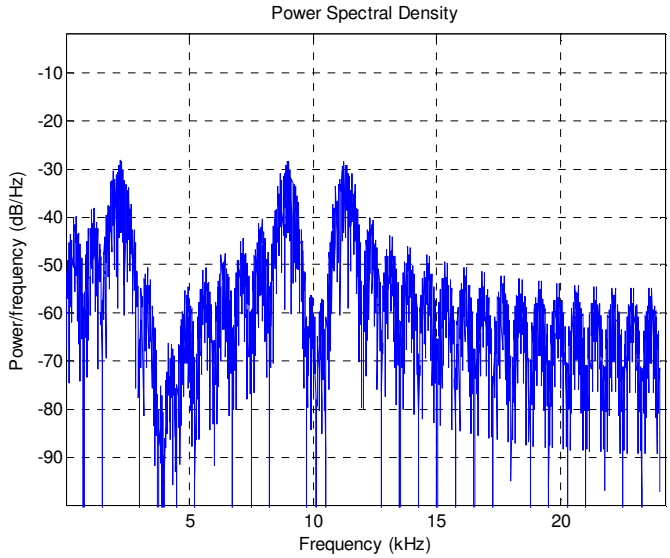


Figure 5.2 – Power Spectral Density of Interferers in Channel (Simulation)

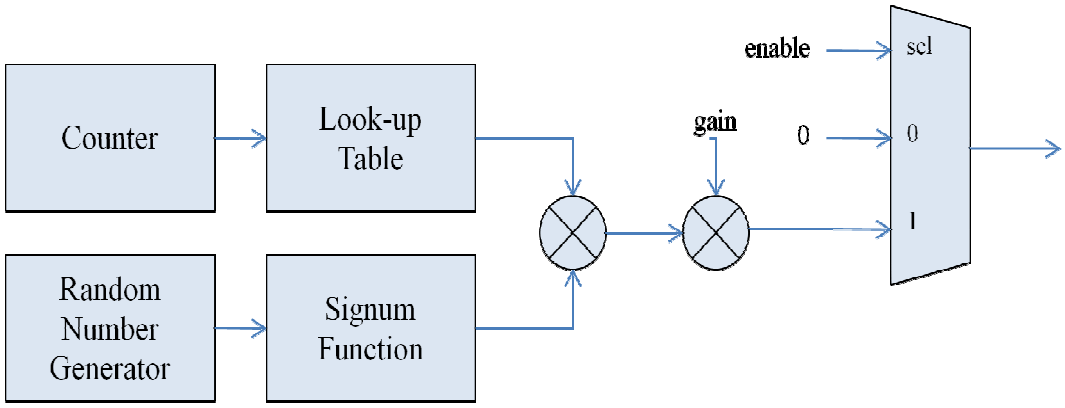


Figure 5.3 – Structure of Narrowband Interferer Block

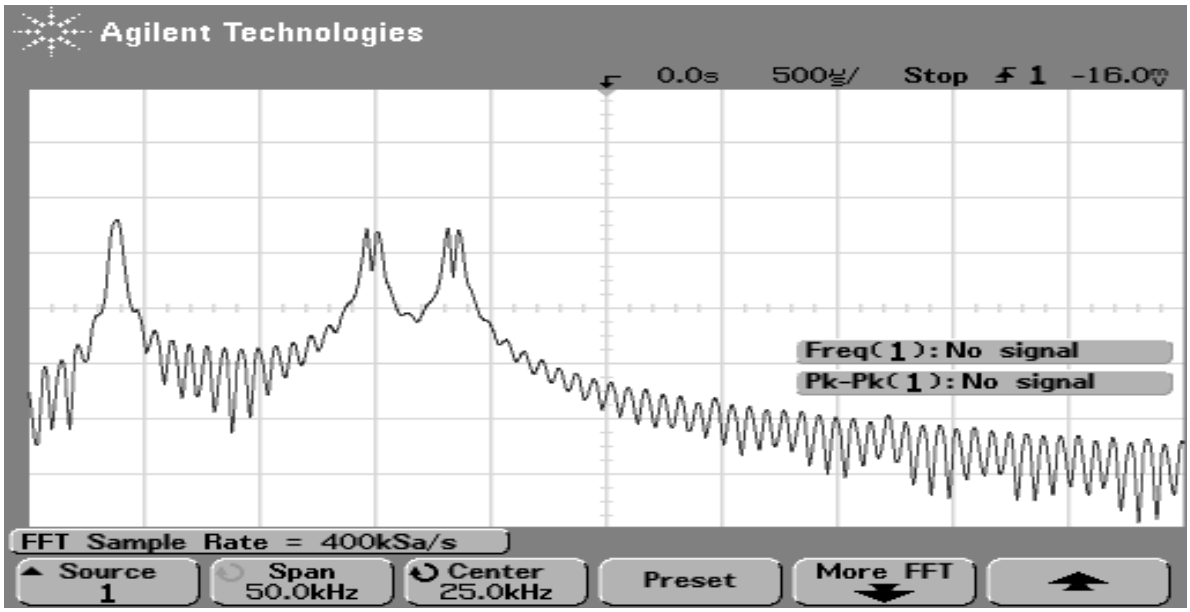


Figure 5.4 – PSD of 3 Interferer Channel Model running on FPGA

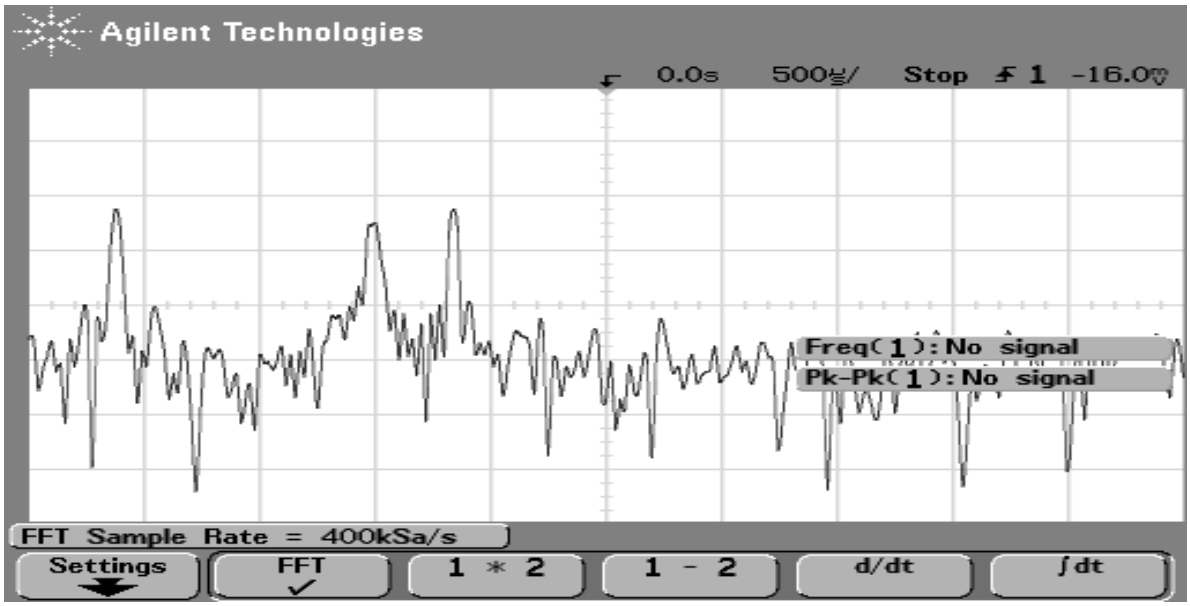


Figure 5.5 – PSD of 3 Interferer + Noise Channel Model running on FPGA

6. Receiver Implementation

Figure 6.1 shows the structure of the SMSE receiver implementation. Time domain reference signals are generated by the blocks enclosed within the dotted line on the figure. These components are used in the same manner as the similar blocks in the transmitter; however, instead of being transmitted, the waveform is used as a reference signal for determining the most likely symbol received. Synchronization of the receiver to the message signal is performed using Direct Time Correlation. The reference waveforms are used in combination with a correlator to determine a maximum likelihood estimate for the received bits. The bit estimates are compared to the original transmitted message to determine the number of bits received in error.

6.1. Receiver Operation

Operation of the SMSE transmitter/receiver pair is triggered by the receiver. On command from the user (triggered via the Code Composer Studio GUI on the controlling PC), the receiver begins by sampling the channel environment. It then commences operation by

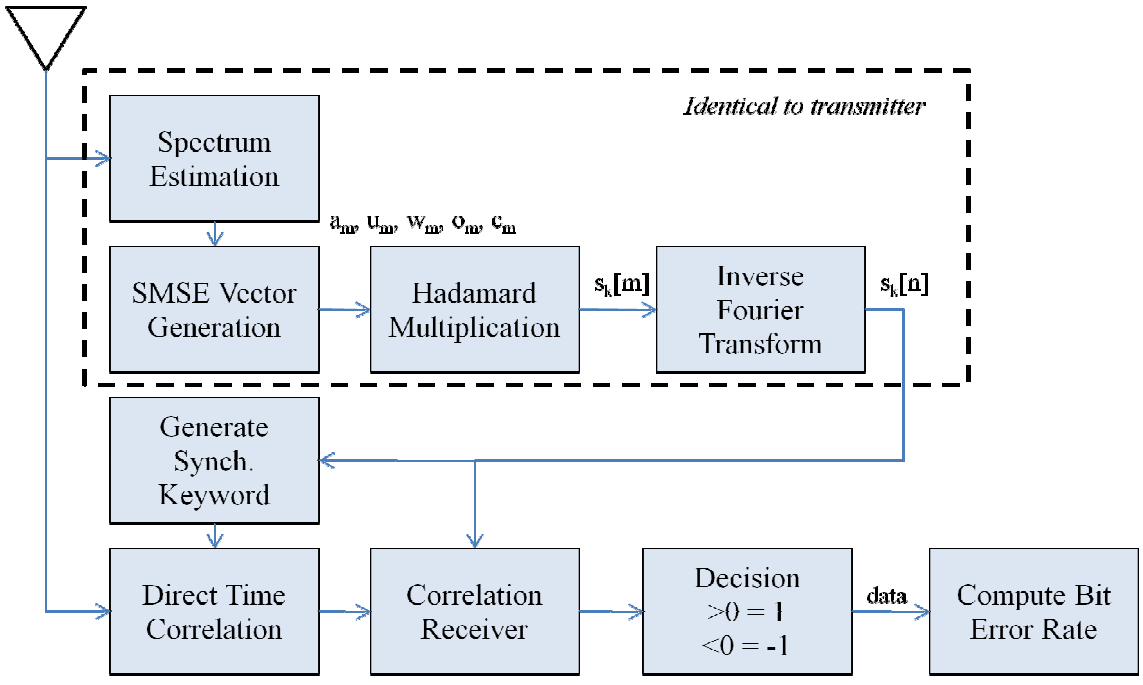


Figure 6.1 – SMSE Receiver Block Diagram

sending a pulse through its digital-to-analog converter to an analog-to-digital converter on the transmitter DSK, signaling that the receiver is ready to receive and that the transmitter should begin a burst transmission. The receiver then begins sampling from its analog-to-digital converter coming from the channel model FPGA and storing samples in a one second (48000 samples) buffer. Once the receiving buffer is full, the receiver stops sampling and moves into the processing phase. The received signal is processed to determine where the message signal begins and to extract the transmitted bits from the signal. The received bit estimates are compared to the transmitted bit sequence (a copy of which is stored *a priori* on the receiver DSK) to compute the bit-error rate observed during transmission. After this is complete, the receiver sends another pulse indicating it is ready to receive another transmission. This process continues for a predefined number of transmissions.

6.2. Reference Waveform Generation

The receiver and transmitter contain identical logic for generating the SMSE waveforms. While the transmitter uses this hardware for generating signals to transmit, the receiver uses this hardware to generate reference signals that are used to analyze the received signal. The spectrum estimation block in the receiver, like the transmitter, develops a frequency usage vector for TDCS based on spectrum estimation using Bartlett's method as described in section 4.2.2. Assuming the transmitter and receiver observe the same spectrum, the transmitter and receiver should generate identical frequency usage vectors. The frequency usage vector (for TDCS or MC-CDMA) is combined with the coding, frequency assignment, spectral windowing and an orthogonality vectors via Hadamard multiplication as described in section 4.2.4, and the time-domain reference waveform is generated via an inverse Fourier transform as described in section 4.2.5. This reference waveform serves two purposes in the receiver. First, the waveform is used to generate a time-domain synchronization waveform that is used in Direct Time Correlation to find the beginning of the message sequence in the received signal buffer. Second, the waveform is used in time-domain matched filter correlation with received symbol waveforms, yielding a maximum likelihood estimate of the bit received.

6.3. Acquisition and Synchronization of Received SMSE Signals

In this implementation, acquisition and synchronization are performed using Direct Time Correlation (DTC). DTC was used for synchronization in TDCS in [29] and is believed to approximate a matched filter implementation. A copy of a known synchronization keyword is correlated with the received signal to determine the point at which the receiver should synchronize to the transmitted signal. The DTC process indicates both the synchronization keyword boundaries and the symbol boundaries meaning the receiver may proceed directly to frame (symbol) processing [29]. Two possible detection techniques are examined in [29], peak detection and threshold detection. In peak detection, the output of the cross-correlation between received signal and reference waveform is assumed to represent the location of the synchronization keyword. In threshold detection, the received signal is continuously shifted and correlated with the reference waveform until the output of the cross-correlation exceeds a certain threshold indicating that the synchronization keyword boundary has been reached. This implementation makes use of the peak detection method since it is assumed that the SMSE signal will exist somewhere in the receiver's sampling buffer. For systems where the receiver may continually sample for a period of time until a transmitted SMSE signal is detected, the threshold

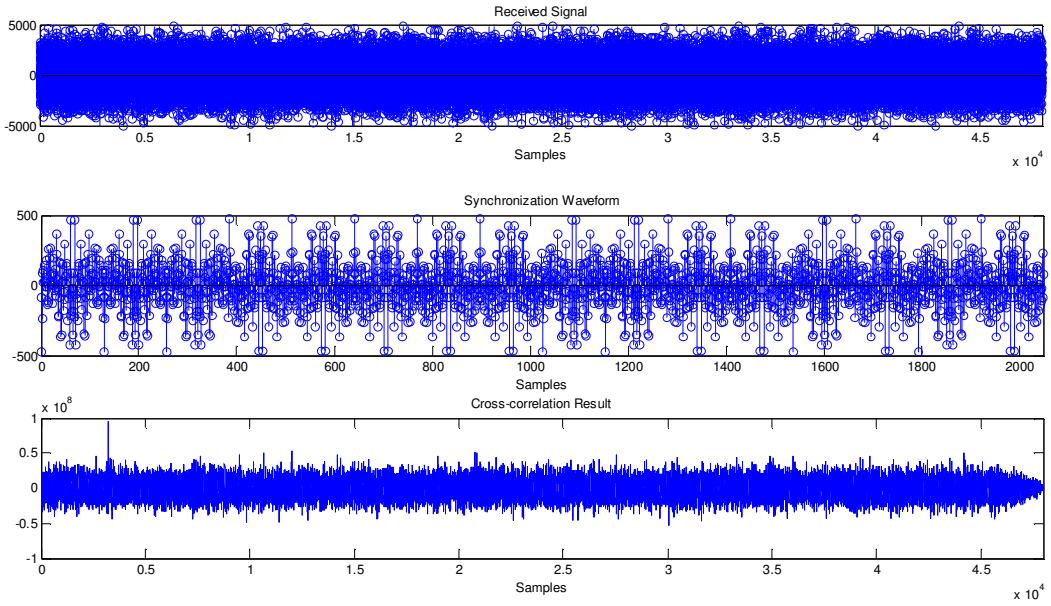


Figure 6.2 – Example of Direct Time Correlation

detection method may be better suited.

The reference SMSE symbol waveform generated at the receiver as described in the previous section is concatenated with itself to form a waveform describing the synchronization keyword bit sequence `0001111100110101` as described in section 4.2.1. To represent a 0 bit, the reference waveform is multiplied by -1 before concatenation. The concatenated waveforms are stored in a 2048 sample buffer (128 samples for each of 16 bits in the bit sequence) representing the synchronization waveform. This buffer is then cross-correlated with the received signal stored in the one-second buffer of samples from the ADC. The peak of the correlation output is found and used as the reference point indicating the beginning of the synchronization bit sequence in the received signal. All prior samples in the received signal are discarded, and the decoding and demodulation process begins at this point in the receiver's buffer. Figure 6.2 shows an example from the fixed point simulation of the direct time correlation process. The 48000-sample received signal, 2048-sample synchronization waveform and cross-correlation result are shown. The peak of the cross-correlation result is at sample 3200, corresponding to a delay of 25 bits (3200 samples divided by 128 samples per bit) before the synchronization bit sequence is transmitted.

6.4. Correlation Receiver

Once the beginning of the message signal has been located within the received signal

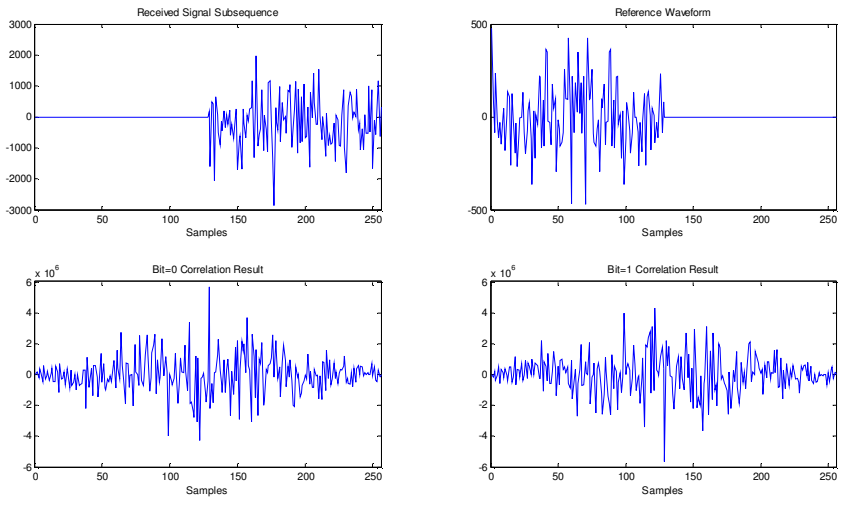


Figure 6.3 – Example of Correlation Receiver

buffer, the transmitted bits can be extracted from the received signal. The 128-sample reference waveform generated at the receiver is cross-correlated with 128-sample subsequences of the received signal buffer. The output of the correlator is sampled at points corresponding to the symbol boundaries, making this correlator receiver implementation equivalent to a matched filter receiver implementation described in [26]. These sample outputs yield test statistics that are then evaluated via a decision rule to determine the symbol most likely to have been received. Since antipodal BPSK modulation is being used, the received subsequences only need to be correlated with one reference waveform. If the correlator output is positive, a binary one is estimated to be received. If the correlator output is negative, a binary zero is estimated to be received. Subsequences of the received signal buffer are correlated in this manner until the buffer is exhausted.

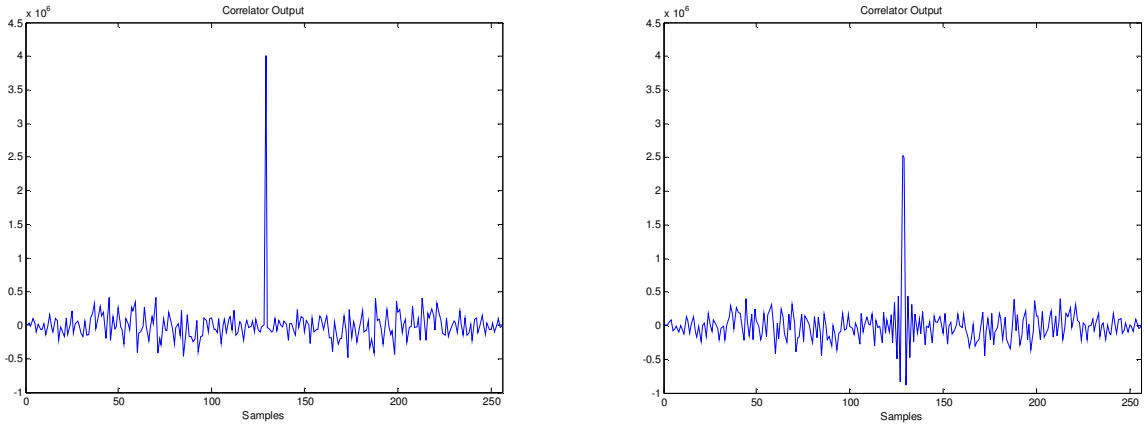
Figure 6.3 shows an example of the correlation process from the fixed point simulation. A subsequence of the received signal buffer is correlated with the reference waveform generated by the receiver. In order to determine the correlation output at both positive and negative delays, both signals are zero-padded before being correlated, turning the two 128-point sequences into 256-point sequences [22]. Knowing the correlation output at both positive and negative delays is important when symbol timing adjustments are taken into account as described in the following sections. The 129^{th} sample of the correlation output, corresponding to the symbol boundary and the peak of the correlation, is used as the test statistic for input to the maximum likelihood decision rule. In an ideal (noiseless) scenario the peak will be either positive or negative depending on the bit being transmitted.

6.5. Symbol Timing Recovery

Symbol timing synchronization is a problem that must be solved when the sampling clocks at the transmitter and receiver are not synchronized as is the case with this application. In such a situation, the sampling clocks at the transmitter and receiver may exhibit frequency offset or phase offset (jitter). The frequency offset is assumed to be relatively small because the transmitter and receiver possess identical hardware that is manufactured to tight standards. However, even when the frequency offset is small the phase offset still may accumulate over time and negatively affect the performance of the communications system [36]. If the

transmitter and receiver sampling clocks are out of phase with each other, the correlation receiver output will not be sampled at the ideal instant.

To solve this problem, an early-late gate synchronizer can be employed [26]. This synchronizer exploits the symmetry of the output of the correlation receiver. If the receiver sampling clock is sampling at the ideal time instants, the correlation receiver output will



(a) (b)

Figure 6.4 – Correlation Receiver Output with (a) ideal sampling time and (b) sampling phase offset

have a maximum absolute value at the sample corresponding to the symbol boundary, and the adjacent samples will have a smaller absolute value. Thus, the sample corresponding to the sample boundary can be used to find the peak of the correlation. However, if the receiver clock samples at some offset away from the ideal sampling point, the output of the correlation receiver will not yield the true peak of the correlation. The true peak of the correlation will instead lie at some point in between the sampling instants of the correlation output. Figure 6.4 illustrates this phenomenon. Note the shorter and wider correlation peak when there is a sampling phase offset between the transmitter and receiver. Since the phase of the sampling clock onboard the ADC cannot be adjusted in real-time, the sampling offset must be estimated by the receiver and steps must be taken in processing to mitigate the effect of the offset and determine where the peak of the correlator output should lie.

In implementing this early-late gate synchronizer, the output of the matched correlator surrounding the sample corresponding to the symbol boundary is interpolated (via Jain's method [16]) to determine the most likely location of the true peak of the correlator output. This information is used to decide whether to sample the correlator output at the sample corresponding the symbol boundary, or one of the adjacent samples. When the location of the estimated peak passes a certain threshold (0.9 samples offset) in either direction, the sampling of the correlator output shifts to the next sample in that direction. In hardware testing, it has been found that this method mitigates but does not completely eliminate errors caused by sampling clock phase offset. It is believed that the since the output of the correlation receiver for SMSE waveforms is not purely symmetric, the effectiveness of the early-late gate synchronizer is reduced.

6.6. Bit Error Rate Calculation

Each received bit estimate obtained by the correlation receiver is stored in a vector of received bits. After the reception process has been completed for the entire received signal buffer, the received bit vector is compared to the copy of the transmitted bit vector in the C header file stored at the receiver. If the estimate of a received bit differs from the bit that was transmitted, a bit error is recorded. The number of errors in each simulation is recorded in an array variable and a console output is displayed, both of which are accessible via the Code Composer Studio interface. Figure 6.5 shows a screenshot of Code Composer Studio and how the error results can be obtained through the Watch Window or the console output.

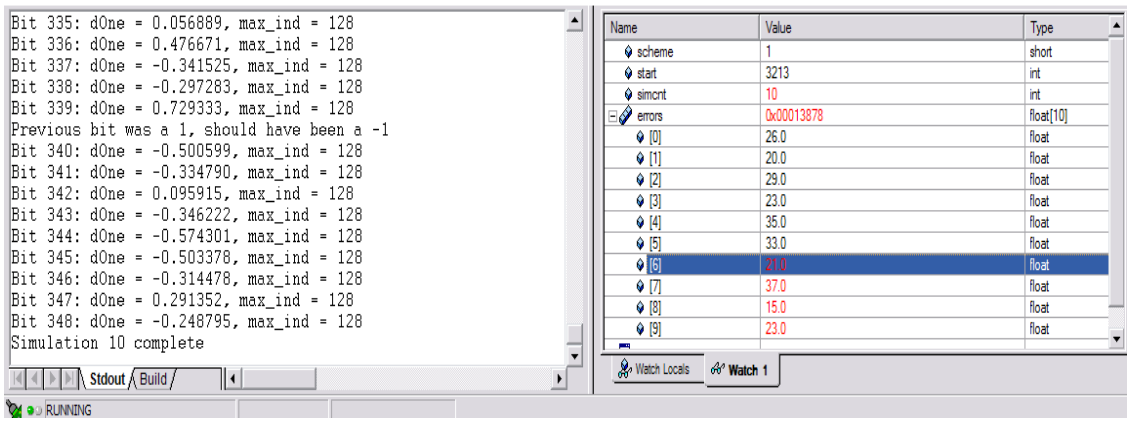


Figure 6.5 – Code Composer Studio showing console output and error count output

7. Results

This chapter presents the results of testing the floating point and fixed point simulations and the hardware implementation. Monte Carlo simulations are used to provide an estimate of the bit-error rate of the SMSE communications system [36]. The probability of bit error is expressed as:

$$P_E = \frac{N_e}{N} \tag{7.1}$$

where N_e is the number of bits received in error out of N total bits transmitted. If N is sufficiently large, P_E will exhibit small variance and thus the simulation will yield an accurate estimate of the bit error probability.

Bit error curves (P_E vs. E_s/N_0) are provided for each of the following scenarios.

- MC-CDMA
  - AWGN
  - 1 interferer
  - 2 interferers
  - 3 interferers
- TDCS
  - AWGN
  - 1 interferer
  - 2 interferers
  - 3 interferers

A separate scenario compares P_E vs. SIR for MC-CDMA with one interferer and constant noise power.

All results are obtained when the receiver is able to correctly synchronize to the message signal using the synchronization keyword embedded in the transmitted signal. In the event of a false synchronization, the bit-error rate obtained is not included in the results, and that simulation is repeated until a correct synchronization occurs. In the case of TDCS with narrowband

interference in the channel, all results are obtained when the TDCS transmitter and receiver pair notch out identical subcarriers that correspond to the correct location of the interferers. In all scenarios (with the exception of the BER vs. SIR plot), the signal-to-interferer power ratio is -12 dB. The narrowband interference remains at the same frequency and magnitude throughout the duration of each simulation. The table below summarizes the location of the interferers for each scenario.

Scenario	Interferer Bins
1 interferer	18
2 interferers	6, 24
3 interferers	6, 24, 30

Table 7.1 – Interferer Bin Locations

In all simulations, bit-error rate is plotted as a function of E_s/N_0, where N_0 includes only the non-colored Gaussian noise. The power of any interfering signals is not included in the calculation of E_s/N_0.

7.1. Floating Point Simulation Results

For the floating point simulation results, each point on the BER curve represents the average bit-error rate value obtained after 50 simulations. Since 350 bits are transmitted in each simulation, this means that 17500 bits are transmitted for each point on the BER curve. E_s/N_0 is incremented from 0 to 6 dB in increments of 0.5 dB. Results for MC-CDMA and TDCS are plotted on the same figure, along with a theoretical bit-error rate curve for BPSK, to allow for easy comparison between the two schemes.

Figures 7.1-7.4 show the performance of MC-CDMA and TDCS in the presence of AWGN and 1, 2, and 3 interferers, respectively. The addition of interference clearly has a detrimental effect on the performance of MC-CDMA. For one or two interferer scenarios, TDCS

is able to mitigate this effect and restore performance to that of BPSK in AWGN. However, when three interferers are present, TDCS is only able to partially mitigate the detrimental effect of the interferers. Performance is still worse than that of BPSK in AWGN. We believe this phenomenon is due to spectral leakage from the interferers raising the noise floor. This possibility will be explored in more detail in later sections. For either modulation scheme, as E_s/N_0 increases, deviations from the theoretical curve become more likely. This is due to the relatively small number of errors that are observed at high E_s/N_0 values. Running more simulations (thus transmitting more bits) would cause these deviations to decrease; however, the simulation time required to do this makes this prohibitive.

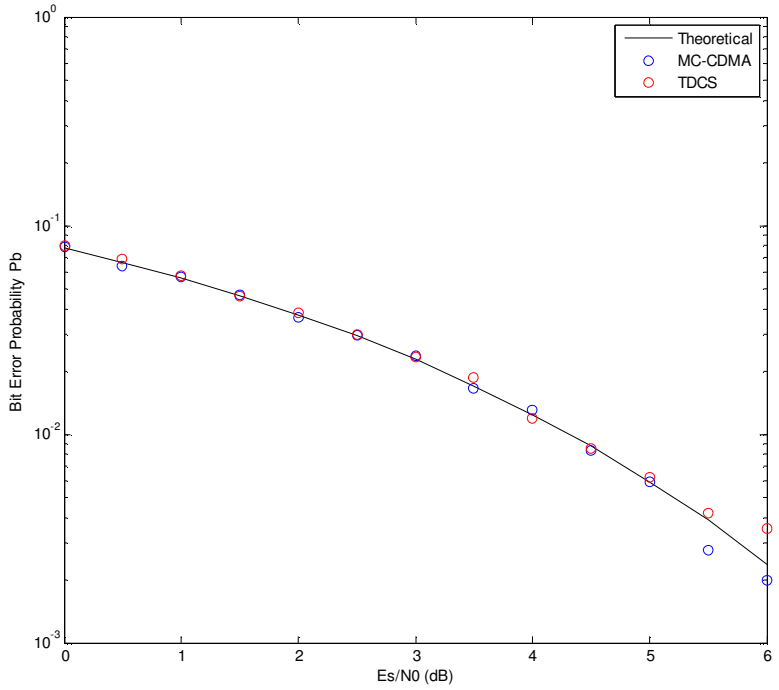


Figure 7.1 – Floating Point Performance of MC-CDMA and TDCS in AWGN Channel

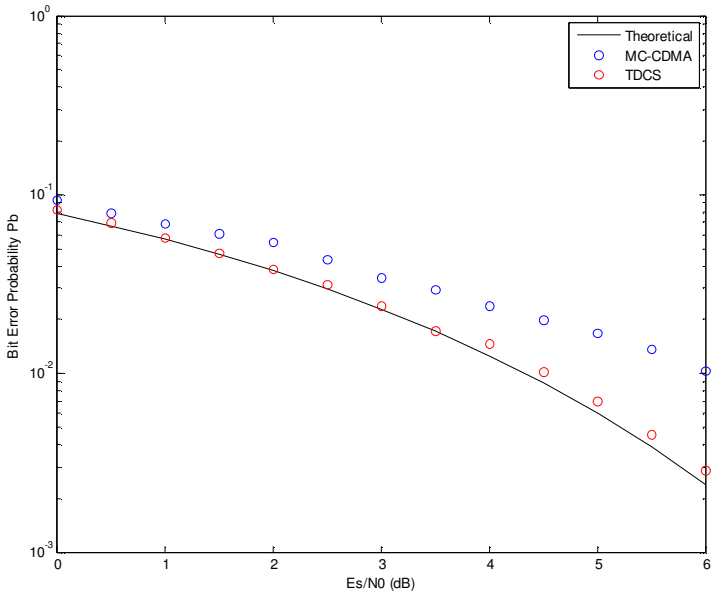


Figure 7.2 – Floating Point Performance of MC-CDMA and TDCS in AWGN Channel with 1 Interferer

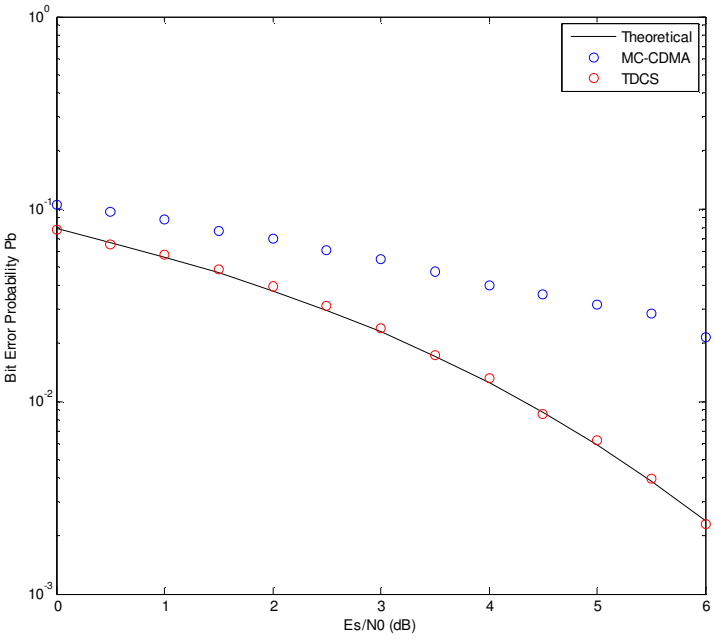


Figure 7.3 – Floating Point Performance of MC-CDMA and TDCS in AWGN Channel with 2 Interferers

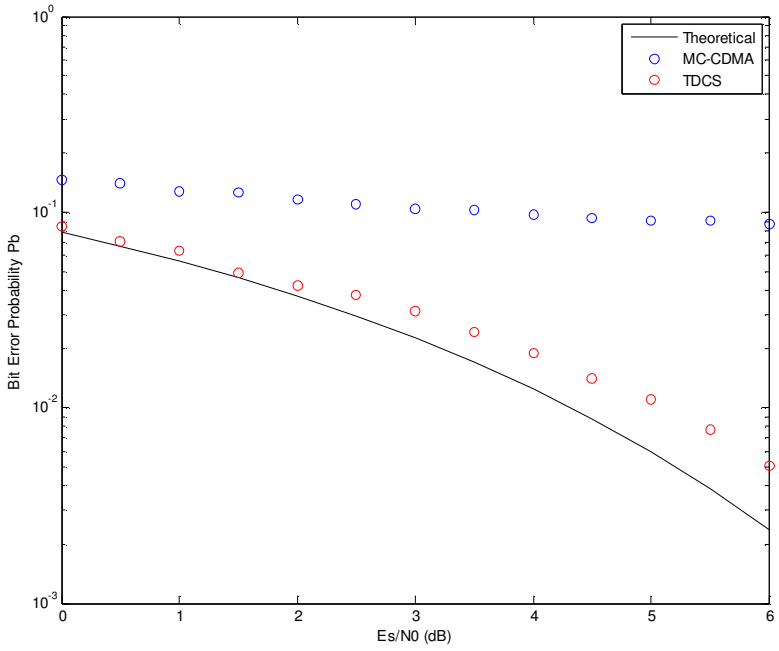


Figure 7.4 – Floating Point Performance of MC-CDMA and TDCS in AWGN Channel with 3 Interferers

7.2. Fixed Point Simulation Results

For the fixed point simulation results, each point on the BER curve represents the average bit-error rate value obtained after 50 simulations. Since 350 bits are transmitted in each simulation, this means that 17500 bits are transmitted for each point on the BER curve. E_s/N_0 is incremented from 0 to 6 dB in increments of 0.5 dB. Results for MC-CDMA and TDCS are plotted on the same figure, along with a theoretical bit-error rate curve for BPSK, to allow for easy comparison between the two schemes.

Figures 7.5-7.8 show the fixed point performance of MC-CDMA and TDCS in the presence of AWGN and 1, 2, and 3 interferers, respectively. The purpose of the fixed point simulation is to simulate the behavior of the code as it will run on the hardware and to verify the results obtained in the floating point simulation. The same trends from the floating point simulation discussed in the previous section, namely the impairment to MC-CDMA's

performance in the presence of narrowband interference, can be seen in the fixed point simulation. Thus, it can be concluded that the fixed point simulation is properly implemented. Furthermore, it is assumed that the hardware implementation will show results similar to both simulations. This will need to be verified when the hardware implementation is tested.

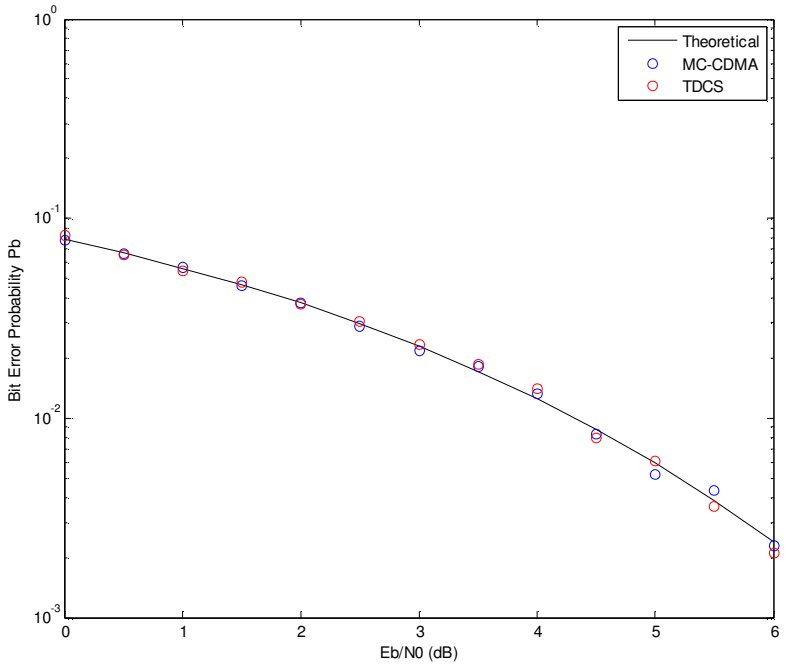


Figure 7.5 – Fixed Point Performance of MC-CDMA and TDCS in AWGN Channel

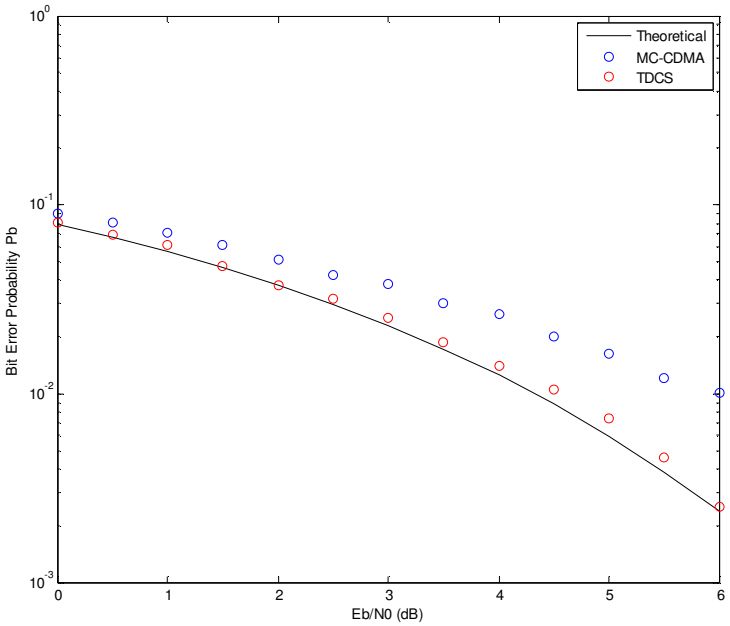


Figure 7.6 – Fixed Point Performance of MC-CDMA and TDCS in AWGN Channel with 1 Interferer

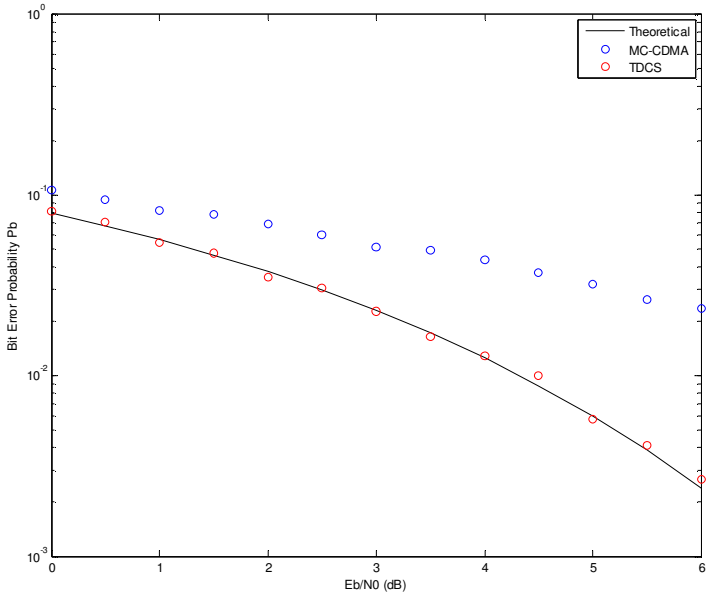


Figure 7.7 – Fixed Point Performance of MC-CDMA and TDCS in AWGN Channel with 2 Interferers

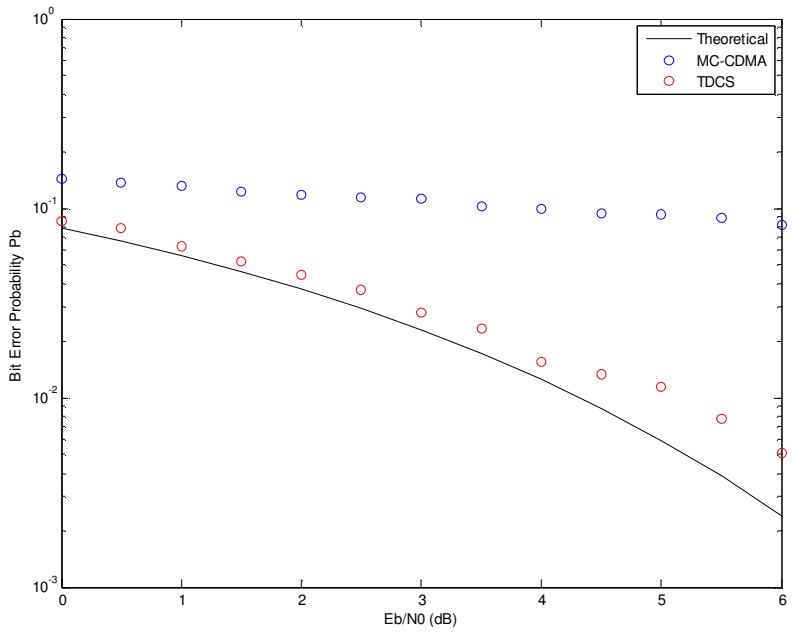


Figure 7.8 – Fixed Point Performance of MC-CDMA and TDCS in AWGN Channel with 3 Interferers

7.3. Hardware Implementation Results

For the hardware results, each point on the BER curve represents the average bit-error rate value obtained in 20 burst transmissions. Since 350 bits are transmitted in each burst, this means that 7000 bits are transmitted for each point on the BER curve. E_s/N_0 is incremented from 0 to 6 dB in increments of 1 dB. Results for MC-CDMA and TDCS are plotted on the same figure, along with a theoretical bit-error rate curve for BPSK, to allow for easy comparison between the two schemes.

Figures 7.9-7.12 show the performance of MC-CDMA and TDCS in the presence of AWGN and 1, 2, and 3 interferers, respectively. The hardware implementation results show the effectiveness of the SMSE system with the transmitter and receiver implemented on separate digital signal processors. The results observed in the hardware testing coincide closely with the results from the floating point and fixed point simulations. When narrowband interference is

introduced to the channel, MC-CDMA exhibits a performance impairment that is most notable at E_s/N_0 of 3 dB or greater. TDCS is able to mitigate this impairment almost completely for one or two interferers. When there are three narrowband interferers in the channel, TDCS is only able to partially mitigate the bit-error performance impairment observed in MC-CDMA.

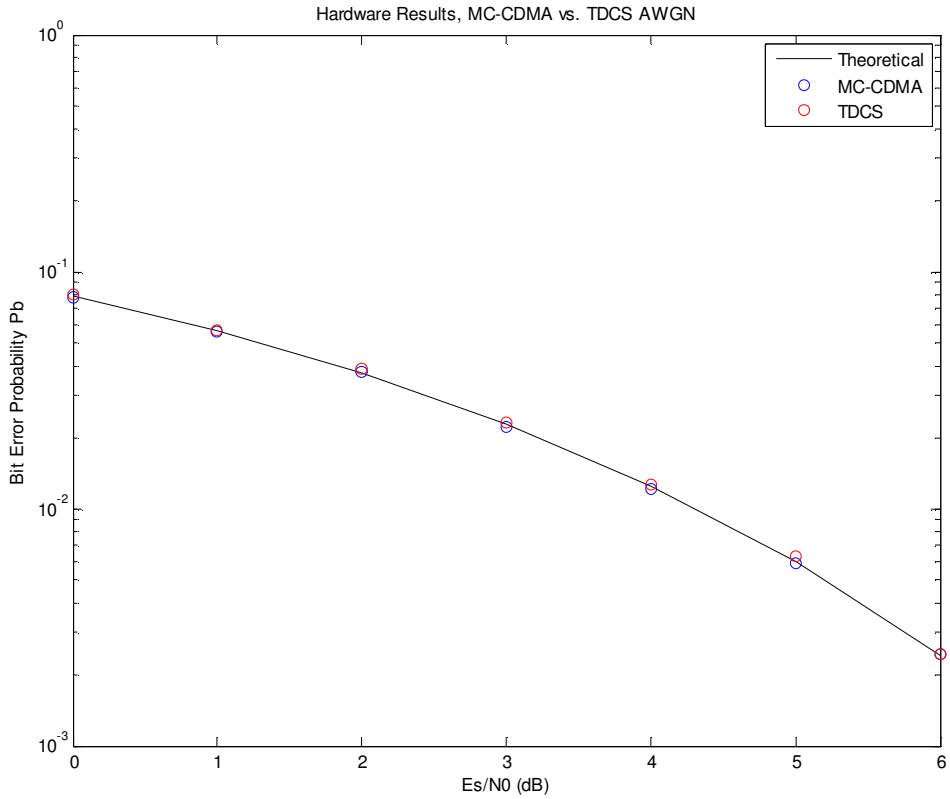


Figure 7.9 – Hardware Performance of MC-CDMA and TDCS in AWGN Channel

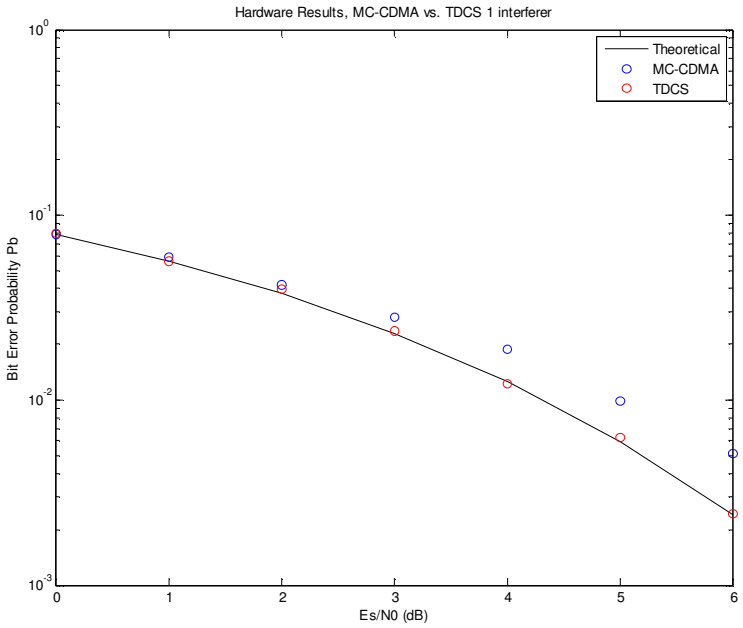


Figure 7.10 – Hardware Performance of MC-CDMA and TDCS in AWGN Channel with 1 Interferer

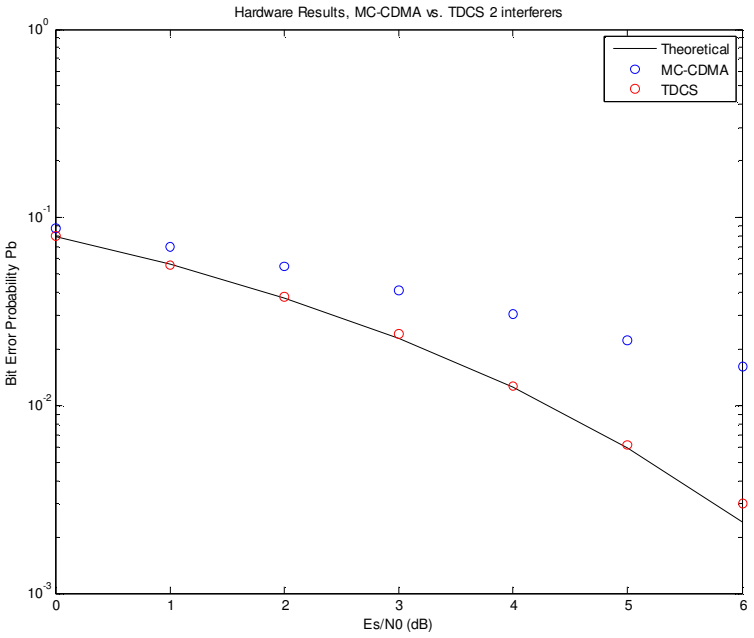


Figure 7.11 – Hardware Performance of MC-CDMA and TDCS in AWGN Channel with 2 Interferers

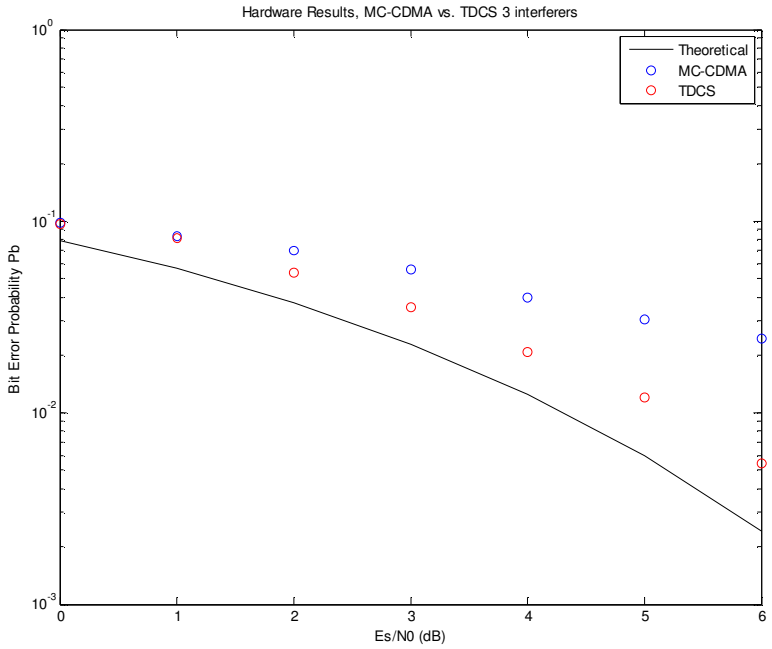


Figure 7.12 – Hardware Performance of MC-CDMA and TDCS in AWGN Channel with 3 Interferers

A separate test was done, evaluating MC-CDMA in the presence of AWGN and a single interferer with varying power. E_s/N_0 was held steady at 6 dB. Figure 7.13 shows the results from the hardware implementation. As with the previous results, each point represents 10 burst transmissions (3500 total bits transmitted per data point). The signal-to-interference ratio (S/I) was varied from -17 dB to -3 dB in increments of 2 dB. The results show that when the S/I is greater than around -5 dB, the performance of MC-CDMA closely matches that of BPSK in AWGN. This led to the decision to adopt a S/I of -12 dB for all scenarios, so that a significant performance detriment could be observed in MC-CDMA.

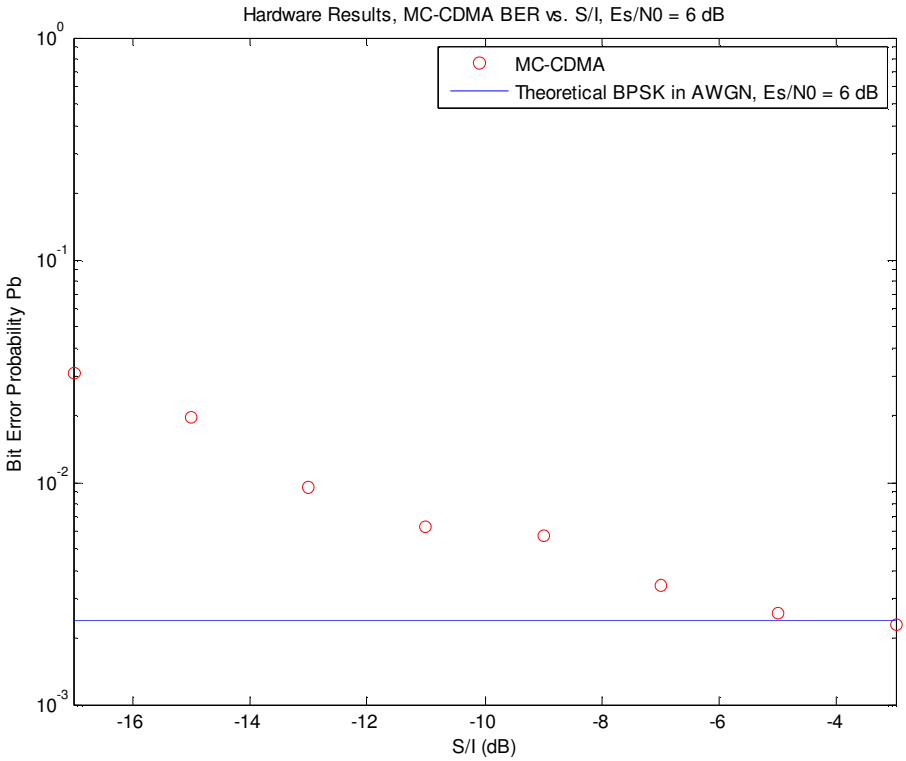


Figure 7.13 – Hardware Performance of MC-CDMA, BER vs. S/I

7.4. Discussion

Based on the results from the hardware implementation, it is believed that the SMSE architecture is properly implemented. Results for MC-CDMA and TDCS in AWGN (without narrowband interference) in Figure 7.9 closely match the theoretical bit-error results expected for BPSK in AWGN. As expected, the introduction of narrowband interference impairs the performance of MC-CDMA. TDCS is able to mitigate the effect of this interference by notching out the interference-laden spectrum in the transmission waveform. The simulation and hardware results indicate that TDCS only partially mitigates the negative effects caused by more than 2 interferers. This result was not expected based on previous results reported in the literature [14]. It is believed that this is due to the nature of the narrowband interference generation performed in the hardware. No kind of filtering is applied to the interferers. When the interferers are modulated, energy is allowed to spread across the entire channel bandwidth. In a more typical

channel scenario narrowband interference would be filtered in order to confine the released energy to the bandwidth of the interferer's message signal. It is believed that this spectral spreading across the bandwidth contributes to a rising of the noise level which impairs the bit-error performance as the number of interferers increases.

To test this theory, in simulation each of the narrowband interferers was put through a bandpass filter centered on the frequency of the interference with a 3 dB bandwidth equal to the bit rate of the BPSK interferer. The purpose of this was to reduce the amount of energy spread outside the message signals being transmitted by each interferer. It is believed that this will reduce the interference (and performance impairment) caused by the rising of the noise floor across the TDCS transmission bandwidth. Figure 7.14 shows the results of this test in floating point simulation. TDCS is now able to restore the performance to that of BPSK in AWGN when there are 3 interferers in the channel. Also note that even MC-CDMA exhibits better performance when the 3 interferers are filtered. Based on this result, it is believed that the originally observed results are due to the way by which the narrowband interference is created in the channel. Figure 7.15 shows an example of the power spectral density of the interference after filtering as observed in the floating point simulation. This figure can be compared to Figure 5.2 to show the difference that filtering makes.

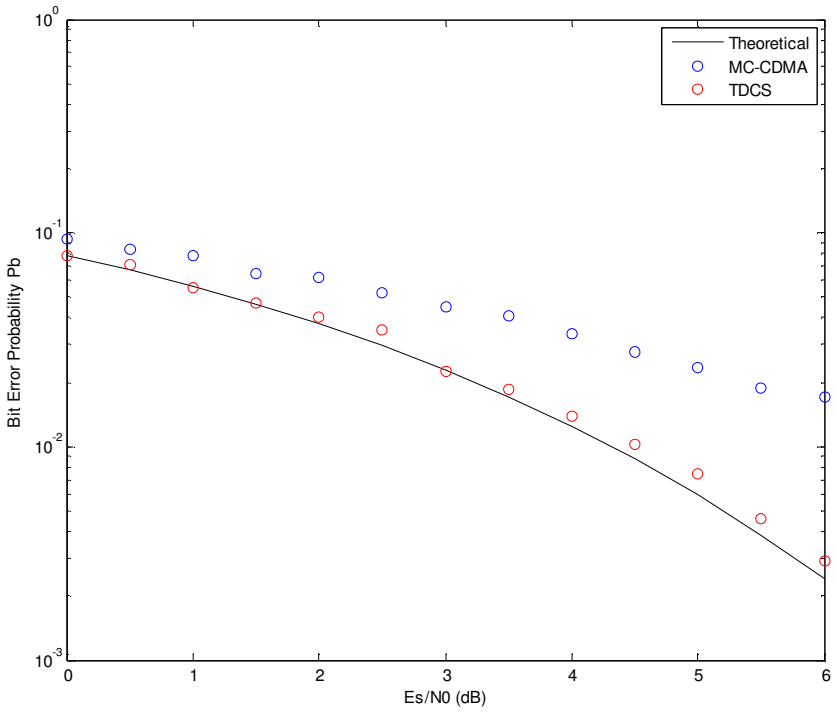


Figure 7.14 – Performance of MC-CDMA and TDCS in AWGN Channel with 3 Filtered Interferers

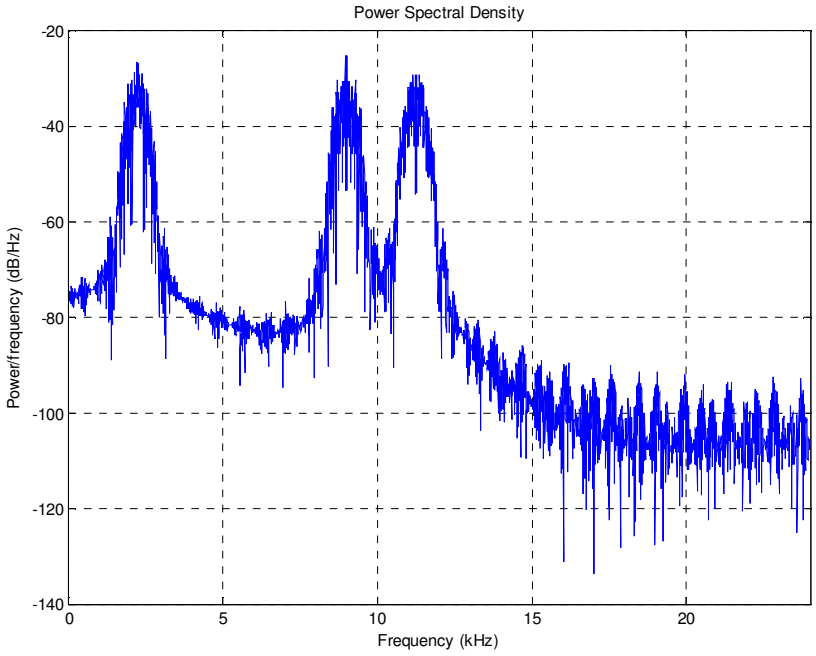


Figure 7.15 – Power Spectral Density of 3 Interferers after Filtering

8. Conclusion

In this research work, a transmitter/receiver pair based on the SMSE framework was successfully implemented in hardware. Two multi-carrier techniques, MC-CDMA and TDCS, were selected as candidates for study to exemplify the capabilities of the SMSE framework. The effects of narrowband interference on MC-CDMA and TDCS were studied, and it was found that TDCS (with its spectral notching capability) was more resistant to narrowband interference than MC-CDMA.

The three goals of this research effort listed in section 1.3 were accomplished. The first goal was to evaluate the bit-error rate performance of a hardware implementation based on the SMSE framework and compare it to previously published results. Chapter VII presented hardware bit-error results that closely match the results obtained in simulation in [30]. Thus, the first goal was met. The second goal was to evaluate the suitability of a DSP for implementation of an SMSE transmitter or receiver and to analyze difficulties encountered in transitioning from simulation to a hardware implementation. This goal has been met, and the DSP has shown to be a capable platform for implementing an SMSE-based transmitter or receiver. The issues encountered in developing the hardware implementation were described in Chapter III. The third and final goal was to compare the performance of the hardware implementation when transmitting using MC-CDMA vs. TDCS in the presence of narrowband interference. This goal was met through the implementation of the narrowband interference channel model (described in Chapter V), and the results were shown in Chapter VII. It was found that the hardware version of TDCS was capable of mitigating the detrimental performance effects caused by one or two narrowband interferers. Further analysis via simulation supported the hypothesis that the hardware results obtained were due to the particular implementation of the narrowband interference. This analysis suggested that TDCS was able to fully mitigate detrimental effects caused by more two interferers when the interference was filtered.

8.1. Recommendations for Future Work

In the course of completing this research, several areas of potential future study were identified. These areas include:

1. The implementation of higher order signaling (QPSK, 8-PSK, 16-PSK, etc.) in the SMSE transmitter/receiver pair.
2. The implementation of an RF front end, downconverter and upconverter to go with the SMSE transmitter/receiver pair. This would eventually lead to an implementation that could transmit wirelessly.
3. Study of the SMSE implementation in a multiple-access environment.
4. Study of the SMSE implementation in a fading channel. This would necessitate implementing channel equalization in the receiver and possibly a cyclic prefix for avoiding intersymbol interference.
5. Converting the SMSE implementation from a burst transmission mode to a streaming (real-time) mode. A different way of performing synchronization and acquisition will need to be implemented.
6. Incorporating more SMSE waveforms into the hardware implementation. Possible options include coded OFDM and the carrier interferometry (CI) variants of OFDM and MC-CDMA [30].
7. The future steps outlined here will help to move the current SMSE implementation from a proof-of-concept to a more robust implementation.

References

1. Andren, A.F. et al., "Low Probability of Intercept Communication System," Harris Corp., U.S. Patent 5029 184, 1991.

2. Andren, Carl. *Short PN Sequences for Direct Sequence Spread Spectrum Radios*. Palm Bay, Florida: Harris Semiconductor, 1997, http://sss-mag.com/pdf/shortpn.pdf.

3. Bourke, Paul. "Cross Correlation." http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlate/.

4. Chakravarthy, V., A. S. Nunez, J. P. Stephens, A. K. Shaw, and M. A. Temple. "TDCS, OFDM, and MC-CDMA: A Brief Tutorial," IEEE Communications Magazine, vol. 43, no. 9 pp. S11-S16, 2005.

5. Chakravarthy, Vasu. "Evaluation of Overlay/Underlay Waveform via SD-SMSE Framework for Enhancing Spectrum Efficiency," Doctor of Philosophy dissertation, Wright State University, 2008.

6. Cigan, Eric and Aaik van der Poel, Ir,. "Simplifying DSP Hardware Development within a MATLAB-Based Design Flow," Compiler Magazine, September 2005.

7. Feickert, Andrew,.." *The Joint Tactical Radio System (JTRS) and the Army's Future Combat System (FCS): Issues for Congress*.," CRS Report for Congress, Congressional Research Service, Washington, DC: 2005.

8. German, Edgar H, *"Transform Domain Signal Processing Study Final Report,"* Technical Report, Contract: Air Force F30602-86-C-0133, DTIC: ADB132635, Reisterstown, MD: August. 1988.

9. Ghazel, A., E. Boutillon, J. -L Danger, G. Gulak, and H. Laamari. "Design and Performance Analysis of a High Speed AWGN Communication Channel Emulator," IEEE PACRIM Conference, Victoria, B.C., August 2001.

10. Goldsmith, Andrea. *Wireless Communications*. New York: Cambridge University Press, 2005.

11. Haker, Marshall E. "Hardware Realization of a Transform Domain Communication System." M.S. Thesis, Air Force Institute of Technology, 2007.

12. Haker, Marshall E., Richard K. Martin, and Vasu Chakravarthy. . Comparison of Hardware Implementation of Transform Domain Communications to Theoretical Results. IEEE Military Communications Conference, (MILCOM) 2007.

13. Han, Chuan, Jun Wang, Shuping Gong, and Shaoqian Li. "Performance of the OFDM-Based Transform Domain Communication System in Cognitive Radio Contexts", The 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM 2006), Mykonos Island, Greece, June 8-10, 2006.

14. Hanzo, Lajos. *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs, and Broadcasting*. Piscataway, N.J.: IEEE Press, 2003.

15. Haykin, S. "Cognitive Radio: Brain-Empowered Wireless Communications." IEEE Journal on Selected Areas in Communications,; vol. 23, no. 2 pp. 201-220, 2005..

16. Jain, Vijay K., William L. Collins, and David C. Davis. "High-Accuracy Analog Measurements via Interpolated FFT." IEEE Transactions on Instrumentation and Measurement, vol. 28, no. 2 pp. 113-122, 1979.

17. Johnson, Don. "Binary Phase Shift Keying," http://cnx.org/content/m10280/2.14/ (accessed June 9, 2010).

18. Kumar, B. P. *Digital Signal Processing Laboratory*. Boca Raton, FL: CRC Press, 2005.

19. Lackey, R. I. and D. W. Upmal. "Speakeasy: The Military Software Radio." IEEE *Communications Magazine, vol.* 33, no. 5 pp. 56-61, 1995.

20. Mitola, J.,III. . *Software Radios-Survey, Critical Evaluation and Future Directions*. National Telesystems Conference (NTC-92), 1992.

21. Mitola, J.,III and G. Q. Maguire Jr. "Cognitive Radio: Making Software Radios More Personal." IEEE *Personal Communications,* vol. 6, no. 4, pp. 13-18, 1999..

22. Mitra, S. K. *Digital Signal Processing: A Computer-Based Approach*. 3rd ed. McGraw-Hill, 2005.

23. ND Tech. "AD/DA Daughter Card (AD6M)." http://www.nd-tech.com/web/main.html?menu=product&sub=sub6 (accessed June 9, 2010).

24. Norton, Quinn. GNU Radio Opens an Unseen World. *Wired Magazine,* June 5, 2006.

25. Proakis, J. G. and D. G. Manolakis. *Digital Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 1996.

26. Proakis, J. G. and M. Salehi. *Digital Communications*. 5th ed. McGraw-Hill, 2007.

27. Radcliffe, Rodney A. "Design and Simulation of a Transform Domain Communication System." M.S. Thesis, Air Force Institute of Technology, 1996.

28. Roberts, M. L., M. A. Temple, M. E. Oxley, R. F. Mills, and R. A. Raines. "A General Analytic Framework for Spectrally Modulated, Spectrally Encoded Signals," Proc. IEEE Int. Conf. Waveform Diversity and Design, Lihue, HI, Jan. 2006.

29. Roberts, M. L., M. A. Temple, R. A. Raines, and E. P. Magee. "Initial Acquisition Performance of a Transform Domain Communication System: Modeling and Simulation Results," 2000 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, October 2000, U33: pp 157-162.

30. Roberts, M. L., M. A. Temple, R. A. Raines, R. F. Mills, and M. E. Oxley. "Communication Waveform Design using an Adaptive Spectrally Modulated, Spectrally

Encoded (SMSE) Framework." IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 1, pp. 203-213, 2007.

31. Rudra, Angsuman, “FPGA-Based Applications for Software Radio,” *RF Design,* May 2004, pp 24.

32. Stranneby, Dag and Walker, William. "Spectral Analysis and Modulation, Part 2: Power Spectrum Analysis." http://www.dspdesignline.com/howto/206801391.

33. The MathWorks. "Measuring the Power of Deterministic Periodic Signals." http://www.mathworks.com/products/demos/shipping/signal/deterministicsignalpower.html (accessed June 9, 2010).

34. The MathWorks. "Precision and Range :: Fixed-Point Concepts." http://www.mathworks.co.uk/access/helpdesk/help/toolbox/fixedpoint/ug/f6415.html.

35. Tranter, W. H., K. S. Shamugan, T. S. Rappaport, and K. L. Kosbar. *Principles of Communication Systems Simulation with Wireless Applications*. Upper Saddle River, NJ: Prentice Hall, 2004.

36. Tretter, Steven A. *Communication System Design using DSP Algorithms with Laboratory Experiments for the TMS320C6713 DSK*. New York: Springer, 2008.

37. Xilinx Inc. *Additive White Gaussian Noise (AWGN) Core v1.0*: Xilinx Inc., 2002.

38. Xilinx Inc. "System Generator for DSP." http://www.xilinx.com/tools/sysgen.htm (accessed June 9, 2010).

39. Xilinx Inc. "XtremeDSP Development Kit — Virtex-4 Edition." http://www.xilinx.com/products/devkits/DO-DI-DSP-DK4-UNI-G.htm (accessed June 9, 2010).

40. Yang, Liuqing and Georgios B. Giannakis. "A General Model and SINR Analysis of Low Duty-Cycle UWB Access through Multipath with NBI and Rake Reception," IEEE Transaction on Wirless Commun., vol. 4, no. 4, pp. 1818-1833, Jul. 2005.

41. Zhou, Ruolin, Xue Li, V. Chakravarthy, C. Bullmaster, Bin Wang, R. Cooper, and Zhiqiang Wu. "Software Defined Radio Implementation of SMSE Based Overlay Cognitive Radio," IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks, Singapore, April 6-9, 2010.