Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

9-25-2008

Application of Optimization Techniques to Spectrally Modulated, Spectrally Encoded Waveform Design

Todd W. Beard

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Other Statistics and Probability Commons, and the Theory and Algorithms Commons

Recommended Citation

APPLICATION OF OPTIMIZATION TECHNIQUES TO
SPECTRALLY MODULATED, SPECTRALLY ENCODED
WAVEFORM DESIGN

DISSERTATION

Todd William Beard, Major, USAF

AFIT/DEE/ENG/08-16

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DEE/ENG/08-16

Application of Optimization Techniques to Spectrally Modulated, Spectrally Encoded Waveform Design

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Todd William Beard, B.S.E.E., M.S.E.E.

Major, USAF

September 2008

APPLICATION OF OPTIMIZATION TECHNIQUES TO
SPECTRALLY MODULATED, SPECTRALLY ENCODED
WAVEFORM DESIGN

Todd William Beard, B.S.E.E., M.S.E.E.

Major, USAF

Approved:

// Signed //	22 Sep 08
Dr. Michael A. Temple (Chairman)	Date
// Signed //	19 Sep 08
Dr. Robert F. Mills (Member)	Date
// Signed //	18 Sep 08
Dr. John O. Miller (Member)	Date
// Signed //	22 Sep 08
Dr. John F. Raquet (Member)	Date

Accepted:

// Signed //	25 Sep 08
M. U. Thomas	Date

Dean, Graduate School of Engineering and Management

Abstract

The Spectrally Modulated, Spectrally Encoded (SMSE) framework provides an effective means for implementing Orthogonal Frequency Division Multiplexing (OFDM) signals – a bedrock technology for future fourth generation (4G) communication systems based on Cognitive Radio (CR) and Software Defined Radio (SDR) techniques. As 4G SMSE communications emerge they must coexist with other systems while competing for available communication resources. Given a lack of inter-system orthogonality and limited available bandwidth, these signals must be designed to spectrally coexist while inducing "manageable" levels of mutual interference. The research goal was to demonstrate a structured means for SMSE waveform design using two techniques commonly employed in operations research: *Genetic Algorithm* (GA) and *Response Surface Methodology* (RSM).

The design process is demonstrated herein for a coexistent scenario containing SMSE and Direct Sequence Spread Spectrum (DSSS) signals. Coexistent SMSE-DSSS designs are addressed under both *perfect* and *imperfect* DSSS code tracking conditions using a non-coherent delay-lock loop (DLL). Under both conditions, the number of SMSE subcarriers N_f and subcarrier spacing Δf are the optimization variables of interest. For *perfect* DLL code tracking conditions, the GA and RSM optimization processes are considered independently with the objective function being end-to-end DSSS bit error rate P_b. A hybrid GA-RSM optimization process is used under more realistic *imperfect* DLL code tracking conditions. In this case, optimization is accomplished through a correlation degradation metric with the GA process being first applied to generate a "coarse" solution followed by RSM processing which provides the final optimized solution.

For all perfect and imperfect DLL code tracking scenarios considered, the optimized DSSS P_b *minimization* results yielded SMSE waveform designs and P_b performance that was consistent with scenarios having no coexistent SMSE signal present (best-case coexistent performance). For the optimized DSSS P_b *maximization* solutions, worst-case

SMSE-DSSS coexistence was achieved for SMSE waveform designs that were spectrally "matched" to the DSSS signal, i.e., greatest P_b degradation was experienced when the resultant SMSE subcarrier spacing Δf was an integer multiple of the spectral line spacing Δf_{Chip} of the DSSS spreading code.

This work has successfully expanded the practical utility of a previously developed tool, the original SMSE framework, by demonstrating a more efficient, structured means for coexistent waveform design that replaces previous trial and error methods. The research objective has been achieved in the sense that 4G communication design engineers now have one additional tool at their disposal and its significance has been acknowledged – the technical community is one step closer to actually hitting the bedrock of OFDM-based signaling using the SMSE framework. It is also important to note that the particular DLL implementation used here, and the metrics used to characterize various tracking conditions (perfect and imperfect), are sufficiently general such that the optimization demonstrations herein are broadly applicable to other non-communication applications employing DLL tracking, e.g., precision navigation, timing, geolocation, etc.

Acknowledgements

First, I would like to thank the United States Air Force for giving me the opportunity to continue my education. Also, I am grateful to AFRL for sponsoring the research and to AFIT for providing a first-rate educational experience.

I am also thankful for the committee. Their help and suggestions were always appreciated.

Several previous AFIT students had stressed to me the importance of selecting the right advisor. In retrospect, I feel extremely fortunate to have worked with Dr. Temple, who constantly went "above and beyond" for my sake.

Finally, and most importantly, I thank God and my family for their support throughout the program. They were always understanding, and I look forward to supporting them as the years go by.

Todd William Beard

Table of Contents

List of Figures

List of Tables

APPLICATION OF OPTIMIZATION TECHNIQUES TO SPECTRALLY MODULATED, SPECTRALLY ENCODED WAVEFORM DESIGN

I. Introduction

A quick glance at the US frequency allocation chart [33] shows that the radio spectrum between 9 KHz and 300 GHz is fully *allocated*. However, this provides no indication as to how *efficiently* the allocated spectrum is being utilized. As the need for increasing data rates, the number of users, and overall network capacity improvement has grown, the issue of spectrum efficiency versus spectrum allocation has gained much attention. This is highlighted by the following relevant sampling of recent literature addressing spectrum efficiency:

1. [39] indicates there is an "underutilization of the radio spectrum as revealed by extensive measurements of actual spectrum usage" in [12].

2. [38] cites measurements in [12] as well and conclude that "at any given time much of the prized spectrum lies idle ... spectrum shortage results from the spectrum management policy rather than the physical scarcity of usable frequencies."

3. [9] cites measurements in [12] as well and indicate that "... at any time roughly 10% of the unlicensed frequency spectrum is actively in use (leaving 90% unused)"

4. [8] provides an overview of the Defense Advanced Research Projects Agency (DARPA) Next Generation (XG) Communication Program and indicates that 94% of the spectrum was unused worldwide at the time of a 2002 study.

5. [1] indicates that "temporal and geographical variations in the utilization of assigned spectrum ranges from 15% to 85% with a high variance in time" and cite [25] in support of their conclusion.

1

While there may be some uncertainty of the current value of spectrum efficiency, as well as its terminology (percentage used, unused, underused, underutilized, etc.), the overall consensus is that spectrum efficiency is much poorer than what can be achieved, and the technical community is unified in its quest to improve it.

To some degree, the technical community has unified under the concept of Dynamic Spectrum Access (DSA) which is highlighted through recent events such as the IEEE's Dynamic Spectrum Access Networks (DySPAN) symposiums [10]. DySPAN has grown to be "the preeminent event to gather international economists, engineers, network architects, researchers and academic scholars together to share cutting edge research on and demonstrations of emerging wireless technology." The original DySPAN symposium was held in 2005 and activities since then have continued to have significant international influence on policy, technology, research and development of next generation wireless systems. From a technology focused perspective, the ability to effectively employ DSA techniques is highly linked to Software Defined Radio (SDR) and Cognitive Radio (CR) concepts.

1.1 Research Motivation

The fundamental DSA concept provides one means for fourth generation (4G) and subsequent communication systems to improve spectrum efficiency while minimizing adverse coexistence effects. This is done by monitoring and adapting to changing channel conditions, traffic requirements, coexisting signals, and spectrum availability by generating waveforms that dynamically respond to these conditions. The coding, modulation, and multiple access techniques of emerging systems will require this type of adaptivity, as enabled at the physical layer through SDR techniques. The SDR advantages are obvious given that the radio can be easily upgraded with changes in standards, frequency allocation, security practices and real-time environmental changes. To some degree, these advantages have been successfully exploited in some systems [13, 19, 23, 24]. However, these systems are somewhat restricted in the decision stage of adaptation, i.e., the "brain" that controls the adaptation based on observed environmental factors is somewhat limited.

2

To address this limitation, CR techniques have been proposed and investigated to increase a given radio's autonomy by increasing its ability to observe, "think", and decide the next best course of action. This represents one form of "brain empowered communications" referred to by Haykin [17]. In this context, CR principles are considered a means for enhancing and advancing SDR functionality and capability. By way of maintaining consistency with previous work in [26, 30] that forms the basis for this research, the synergistic union between CR and SDR will be referred to here as *CR-based SDR*. In this context, the SDR is a software controlled waveform generator that is controlled by CR principles designed to improve environmental assessment and spectrum usage efficiency. In the context of achieving smarter communications, Haykin identifies Orthogonal Frequency Division Multiplexing (OFDM) as a bedrock technology for CR-based SDR implementation [17]; as with previous work in [26, 30], this continues to provide the motivation for the research presented here.

While the concept of OFDM is relatively simple, the ability to generate OFDM signals has only recently become practical from a hardware perspective. The speed of modern digital processors now allows waveform characteristics to be defined in the frequency domain, with conversion to time domain waveforms via an Inverse Fast Fourier Transform (IFFT) occurring at rates that enable high-speed communications. As such, OFDM has proven to be very flexible and has generated significant interest throughout the research community [11, 20, 34–37, 40]. The rapidly expanding pool of emerging OFDM-based techniques, as illustrated in the cloud region of Fig. 1.1, drove the need to develop a unified framework to encapsulate OFDM variants. Elements of the resultant framework are captured in the analytic expression in Fig. 1.1 which effectively embodies a class of what are now called Spectrally Modulated, Spectrally Encoded (SMSE) waveforms [26–30].

Researchers continue to investigate expanded roles for applying the original SMSE framework. Some of the more recent efforts are focusing on overlay, underlay and hybrid overlay/underlay waveform implementations [5, 6]. In the context of an overlay waveform architecture where interference to primary users is mitigated by avoiding their spectral regions, these latest SMSE developments suggest that the original SMSE framework

3
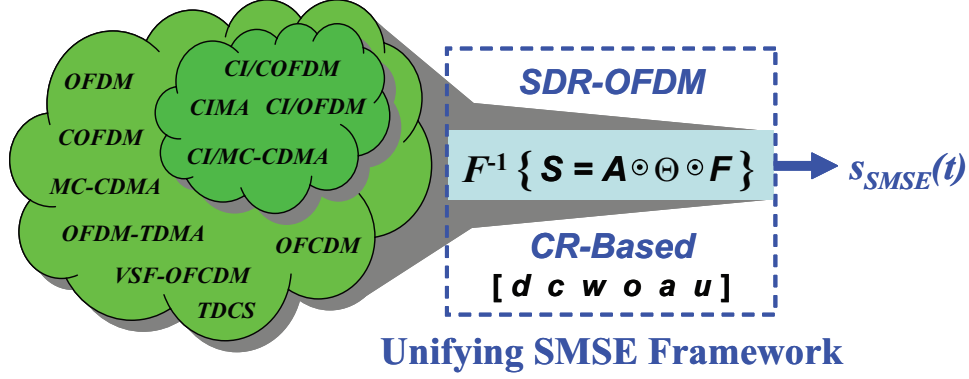
Figure 1.1: Unifying SMSE framework for OFDM-based signaling using a CR-based SDR architecture. Waveform adaptivity is provided through variation in design variables denoted by $[\,d\ c\ w\ o\ a\ u\,]$ [26].

is best characterized as applying *hard decision* criteria to spectral usage. By accounting for possible implementation of both overlay and underlay techniques, these recent activities have demonstrated preliminary success with what is being called *soft decision* SMSE (SD-SMSE). While the research focus in this dissertation is on the original SMSE framework and its optimization for coexistence with other systems, emergence of SD-SMSE techniques certainly warrants future investigation given that the methods considered here should to be directly applicable.

1.2 Research Assumptions

Definitions of mathematical symbols and terminology are provided throughout the document as they are introduced. Results of all work presented should be interpreted within limits and constraints imposed by the assumptions that have been made. The following summarizes the assumptions that made for the research.

1.2.1 Coexistent SMSE-DSSS Scenario.

- For all scenarios considered, the coexistent SMSE and DSSS signals are spectrally coexistent (same center frequency) and are operating over an Additive White Gaussian Noise (AWGN) channel. However, given the focus of the waveform design

4

demonstrations are on *process*, the procedures used are valid for other channel models.

- For coexistent and interference scenarios, the relative power ratios between the signal of interest, the interfering signal, and the channel noise are the dominant factors. Therefore, the research assumes these relative power levels are set such that SMSE parametric changes within can impact DSSS receiver performance.

- Except for spreading code tracking, the DSSS receiver was perfectly synchronized to the transmitted DSSS signal in terms of carrier tracking (frequency and phase) and communication symbol tracking. In addition, final results presented in Section 4.1 are based on *perfect* spreading code tracking while the results presented in Section 4.2 are based on *imperfect* spreading code tracking.

1.2.2   Optimization Techniques.

- Less-than-global optimal solutions were deemed acceptable for demonstration purposes. The optimal responses to SMSE variable changes were at least locally optimal and accepted given the research goal was to find "good" parameter values *without* requiring exhaustive testing.

- All GA experiments were designed to ensure that all possible combinations of SMSE parameters (N_f, Δf) were allowed, i.e., all combinations were in the optimization feasibility region.

- All RSM experiments used a second-order β model for optimization. The experiments consisted of a two-factor, three-level, full-factorial design with four additional center runs. The SMSE parameters were optimized using the steepest ascent/descent process until the response surface fit the second-order model, as determined by an ANOVA.

1.3 Research Sponsorship

II. Background

2.1 Digital Communications

Electronic communications can occur using either analog or digital waveforms. Analog communication applications include television (TV), AM radio, FM radio, and early cellular telephones. Digital communication techniques differ from analog techniques in that information is sent using a set of predefined quantized values that are commonly represented by either a binary value of 0 or 1. Digital communication applications include modern cellular telephone systems, high definition television (HDTV) and a multitude of wireless computer network implementations [31]. There are many reasons why modern communication system designers prefer digital techniques. First, the digital receiver must only decide between a finite number of transmit conditions (communication symbols). Thus, digital techniques perform very well in noisy environments since the original signal does not have to be reconstructed. Second, digital repeater systems may be implemented as regenerative receive-transmit nodes. They are able to receive, estimate symbols, perform error correction and duplicate information before passing it on, allowing reliable long distance communications. Also, digital hardware is often less expensive to manufacture than analog equipment [31].

A basic digital communications system model is shown in Fig. 2.1. The digital input sequence $\{d_k\}$, $d_k \in \{0, 1\}$, is mapped to a waveform $s(t)$ for transmission. This process of mapping information to a transmitted waveform is called *modulation*. The transmitted signal propagates through the channel, or transmission medium, and is corrupted by noise. The channel may be a transmission line, such a telephone line or coaxial cable, or the open airwaves as assumed for this research. The term *noise* describes the combined effect of interference from many sources, including the combined effect of additional signals in the channel, atmospheric effects and thermal effects within the receiver itself. Additive white Gaussian noise (AWGN) is a common channel model used for designing and analyzing electronic communications systems. As used for this research, the AWGN channel assumption provides a reasonable starting point for system design.

Figure 2.1:   Basic digital communication system model showing transmitter modulation and receiver demodulation functions. An AWGN propagation channel is illustrated.

M-ary Phase Shift Keying (MPSK) is one type of digital data modulation whereby the input data bits are mapped to a series of transmitted communication symbols given by

$$s(t) = A \cos \left[2\pi f_c t + \phi \left(t \right) \right] , \tag{2.1}$$

where $0 \leq t \leq T_{sym}$, T_{sym} is the symbol duration, and phase value $\phi(t)$ is determined by the input bit values. The special case where each symbol represents $M = 2$ bits, is known as Binary Phase Shift Keying (BPSK) which is the basis for the analysis in this dissertation. In this case, $\phi = 0°$ ($d_k = 1$) or $\phi_i = 180°$ ($d_k = 0$) and the expression in (2.1) can be rewritten as

$$s(t) = (-1)^{d_k} A \cos \left(2\pi f_c t \right) . \tag{2.2}$$

The plot in Fig. 2.2 shows a representative BPSK waveform spanning two symbol periods having different modulation values (either $d_k = 0$ or $d_k = 1$). This bit difference causes the $180°$ phase shift occurring at T_{sym} in the plot.

To use the available communication resources efficiently, digital waveforms are often *multiplexed* within or across specific signaling domains (time, frequency, space, polarization and/or code). Several multiplexing schemes exist that allow multiple users to effectively share communication resources, including *Time Division* (TDM), *Frequency Division* (FDM), *Space Division* (SDM), *Polarization Division* (PDM) and *Code Division*

8
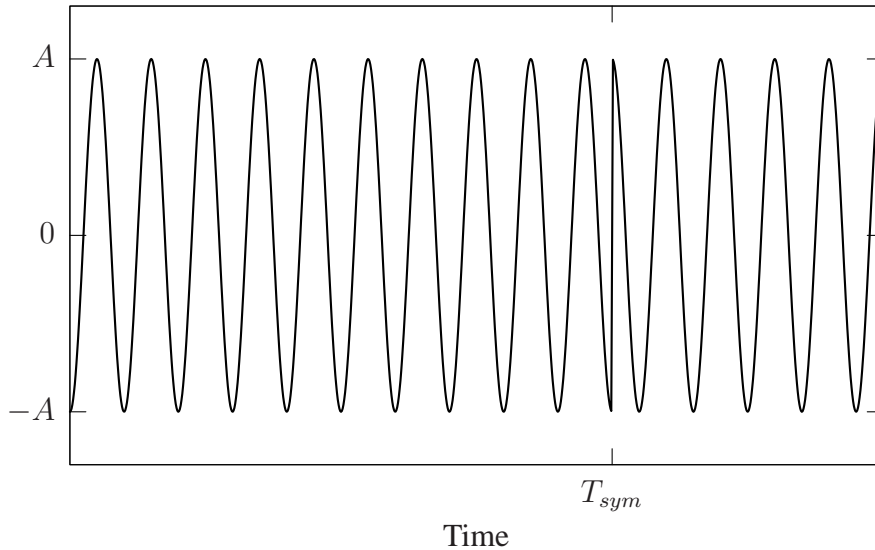
Figure 2.2:   Representative Binary Phase Shift Keyed (BPSK) waveform spanning to symbol periods having different data modulation values. The instantaneous $180°$ phase change at T_{sym} is due to this difference [31].

(CDM) [31]. Of particular interest to this research are FDM and CDM, each of which is introduced and discussed in greater detail where appropriate.

2.1.1 Spectrally Modulated, Spectrally Encoded (SMSE) Framework.   One method to allow more data through a given channel involves dividing the allocated frequency band into several narrower subbands, allowing multiple individual signals to coexist. This technique is known as Frequency Division Multiplexing (FDM) [31]. A representative FDM spectrum is illustrated in Fig. 2.3 for three subbands separated by Δf. The receiver for a particular signal in this scheme is able to extract the desired signal using an appropriate band pass filter centered on the subband of interest and having a bandwidth that is a fraction of Δf, say $W_{BP} = \Delta f/2$. In this case, a "manageable" amount of interfering signal power from the other two subbands passes through the filter and impacts demodulation [31]. Given that the signals are separated in frequency, the important design parameters for FDM include frequency separation Δf and W_{BP}.

One particularly efficient way to space FDM subcarrier frequencies is to assign Δf such that the individual subband center frequencies are spectrally coincident with the first

9

null of adjacent signal [34]. For this to occur, the symbol duration and subcarrier separation are related such that $T_{sym} = 1/\Delta f$, i.e., the individual subcarrier frequencies are harmonics of the fundamental frequency Δf. This particular subcarrier spacing is illustrated in Fig. 2.4. Ideally, this process results in individual signals being mathematically orthogonal with no mutual interference. This type of FDM scheme is known as Orthogonal Frequency Division Multiplexing (OFDM) [34].

While the concept of OFDM is relatively simple, the ability to generate OFDM signals has only recently become practical from a hardware perspective. The speed of modern digital processors now allows waveform characteristics to be defined in the frequency domain, with conversion to time domain waveforms via an Inverse Fast Fourier Transform (IFFT) occurring at speeds that enable high-speed communications. As such, OFDM has proven to be very flexible and has generated significant interest throughout the research community [11, 20, 34–37, 40]. The rapidly expanding pool of emerging OFDM-based techniques drove the need to develop a unified framework to encapsulate OFDM variants. The resultant framework effectively embodies a class of what are now called Spectrally Modulated, Spectrally Encoded (SMSE) waveforms [27–30].

Researchers continue to investigate expanded roles for applying the original SMSE framework, with some of the more recent efforts focusing on hybrid overlay/underlay waveform implementations [5, 6]. In the context of an overlay waveform architecture where interference to primary users is mitigated by avoiding their spectral regions, these latest SMSE developments suggest that the original SMSE framework is best characterized as applying *hard decision* criteria to spectral usage. By accounting for possible implementation of both overlay and underlay techniques, these recent activities have demonstrated preliminary success with what is being called *soft decision* SMSE (SD-SMSE). While the research focus in this dissertation is on the original SMSE framework and its optimization for coexistence with other systems, emergence of SD-SMSE techniques certainly warrants future investigation given that the methods considered here should be directly applicable. For completeness, the following SMSE development is provided and is based on the orig-
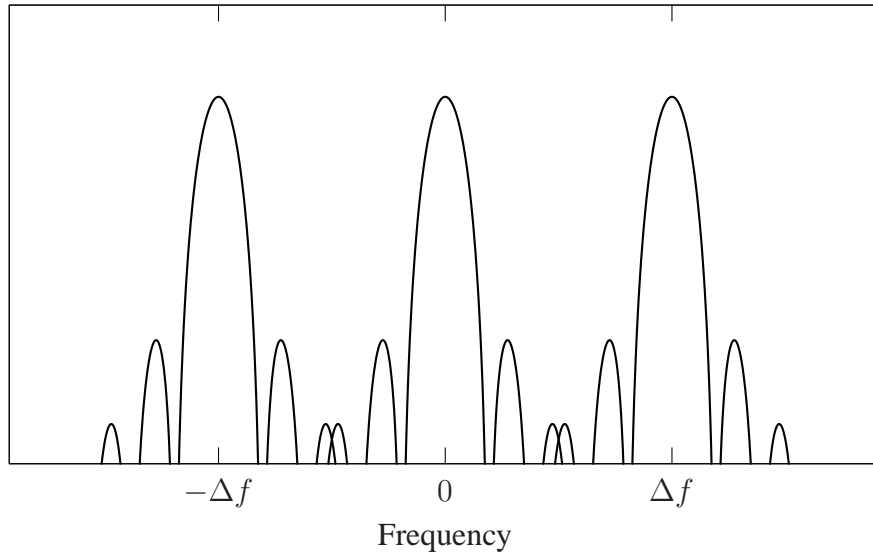
Figure 2.3:   Representative Frequency Division Multiplexed (FDM) spectrum for three subcarriers separated in frequency by Δf [34].
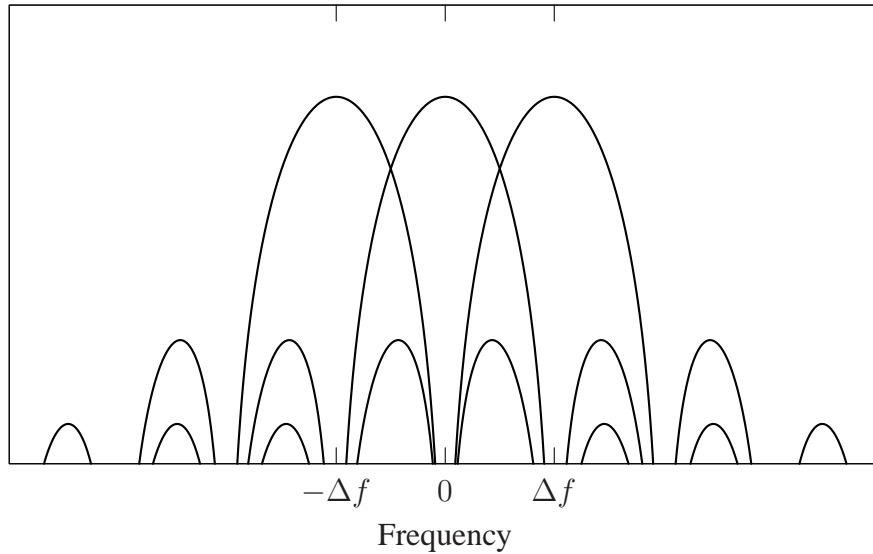
Figure 2.4:   Representative SMSE signal spectrum for three subcarriers separated in frequency by $\Delta f = 1/T_{sym}$, an integer multiple of the symbol interval [34].

inal SMSE framework presented in [30]. The reader is referred to the original SMSE framework documentation if additional details are required.

Accounting for all SMSE waveform design variables, the original SMSE framework provides a unified approach for generating and characterizing a host of OFDM-based signals. Using $\odot$ to denote Hadamard product (element-by-element multiplication), the spectral representation of the kth SMSE symbol is given by [28, 30]

$$\mathbf{s}_k = \mathbf{c} \odot \mathbf{d}_k \odot \mathbf{w} \odot \mathbf{o}_k , \tag{2.3}$$

where the design variables are defined as follows: *Coding* $\mathbf{c} = [c_1, c_2, \ldots, c_{N_f}], c_i \in \mathbf{C}$; *Data Modulation* $\mathbf{d} = [d_1, d_2, \ldots, d_{N_f}], d_i \in \mathbf{C}$; *Windowing* $\mathbf{w} = [w_1, w_2, \ldots, w_{N_f}], w_i \in \mathbf{C}$; *Orthogonality* $\mathbf{o} = [o_1, o_2, \ldots, o_{N_f}], o_i \in \mathbf{C}, |o_i| = 1 \ \forall \ i$. Each of these terms are introduced to functionally incorporate various waveform design characteristics commonly employed in communications.

All that remains to completely specify the SMSE waveform is the frequency component selection and symbol duration of the resulting waveform. The frequency component defines the number of carrier components that are spectrally modulated and encoded. Assuming an N_f-point inverse fast Fourier transform (IFFT), there are initially N_f possible carrier components available. Use of components from this pool of frequencies is controlled through frequency *assignment* and *use* variables. For example, a system may elect to assign a subset of N_f carriers to a given user. This *assignment* of frequencies is accounted for through variable $\mathbf{a} = [a_1, a_2, \ldots, a_{N_f}], a_i \in \{0, 1\}$, where zeros indicate unassigned carriers. From this assigned pool of carriers, some may be unused due to excessive interference, system design, etc. The remaining *used* carriers are accounted for through variable $\mathbf{u} = [u_1, u_2, \ldots, u_{N_f}], u_i \in \{0, 1\}$, where zeros indicate unused carriers and there are total of P used carriers ($P \leq N_f$). Thus, $\mathbf{u}$ is a subset of $\mathbf{a}$ ($\mathbf{u} \subseteq \mathbf{a}$) and only assigned carrier components exist SMSE symbols. The frequency assignment and use variables are incorporated into (2.3) as follows

$$\mathbf{s}_k = \mathbf{a} \odot \mathbf{u} \odot \mathbf{c} \odot \mathbf{d}_k \odot \mathbf{w} \odot \mathbf{o}_k \, , \tag{2.4}$$

where the mth carrier component of $\mathbf{s}_k$ is given by

$$\mathbf{s}_k[m] = a_m u_m c_m d_{m,k} w_m \mathrm{e}^{j(\theta_{d_{m,k}} + \theta_{c_m} + \theta_{w_m} + \theta_{o_{m,k}})} \, . \tag{2.5}$$

There are $m = 0, 1, \ldots, N_f - 1$ frequency components with c_m, θ_{c_m}, $d_{m,k}$, $\theta_{d_{m,k}}$, w_m, θ_{w_m} and $\theta_{o_{m,k}}$ being the corresponding magnitudes and phases of the design variables. As indicated by the subscripted k and m indices in the righthand side of (2.5), the coding and windowing terms only vary with frequency index m, whereas the data modulation and orthogonality terms vary with symbol index k as well.

The block diagram in Fig. 2.5 illustrates the functional processes that are commonly used to generate OFDM signals. Input data bits $d_k \in \{0, 1\}$ are mapped to discrete BPSK coefficients according to $b_k = (-1)^{d_k}$. The BPSK coefficients are first grouped using a serial-to-parallel (S/P) conversion process and then weighted by vector $\mathbf{w_k}$ prior to the IFFT operation. Following the IFFT operation, the grouped bits are converted back to a stream stream via parallel-to-serial (P/S) conversion. The resultant complex discrete samples are then converted to in-phase and quadrature baseband signals by digital-to-analog conversion (DAC). Finally, the complex components are combined in a quadrature modulator and up-converted to the desired carrier frequency f_c for transmission. For the coexistence analysis in this work, the key SMSE waveform design variables include the number of IFFT points N_f, the subcarrier frequency spacing Δf and the inter-subcarrier complex weighting $\mathbf{w_k}$.

2.1.2   Direct Sequence Spread Spectrum (DSSS) System.   Spread spectrum communications are a class of signals that employ Code Division Multiple Access (CDMA). As a mechanism for enabling multiple access, CDMA provides the ability to increase the number of users within a finite allocation of spectrum. In terms of multiple access performance, CDMA has an inherent advantage over Time Division Multiple Access (TDMA) given there is no need for precise timing between users in the network. Two other desirable
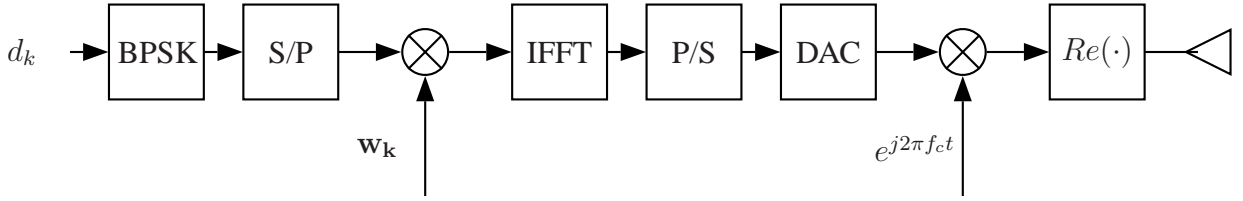
Figure 2.5: Block diagram illustrating functional processes commonly used for OFDM symbol generation and transmission [34].

characteristics of CDMA when implemented with spread spectrum techniques, include: 1) users enjoy some degree of privacy from unintended eavesdroppers given that the code is only shared with intended receivers, 2) spreading of information across a wide band of frequencies provides some inherent protection against fading channel effects, and 3) the initial despreading operation in the DSSS receiver provides some level of protection against coexisting interferers or jammers [25, 31]. Spectral spreading in a DSSS is generally achieved using a psuedorandom noise (PN) code. The PN code is actually deterministic but possesses some properties that are similar to random noise when observed without knowledge of the code [31].

A typical DSSS waveform generation process is illustrated in Fig. 2.6. For BPSK data modulation, the generation process first begins by mapping the sequence of input data bits $\{d_k\}$ to waveform values to created the data modulated signal $d(t)$ given by

$$d(t) = (-1)^{d_k} p(t) , \tag{2.6}$$

where

$$p(t) \equiv \begin{cases} 1, & t \in (0, T_{sym}] \\ 0, & \text{otherwise} \end{cases} .$$

The resultant data modulated waveform $d(t)$ modulates the carrier signal $A \cos{(2\pi f_c t)}$, where f_c is the carrier frequency. The carrier modulated signal is then spectrally spread by

14
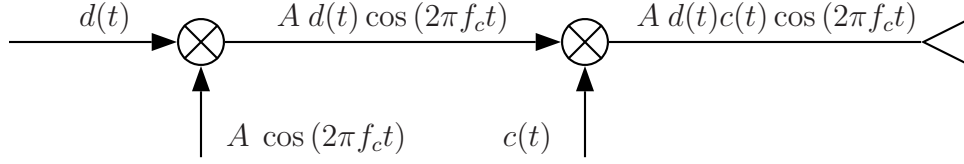
Figure 2.6:   Typical DSSS signal generation process. The carrier carrier f_c is first modulated by data dependent waveform $d(t)$ and then spectrally spread by coded waveform $c(t)$ [25].

the PN coded waveform $c(t)$. Although not a requirement, the spreading waveform $c(t)$ is generally created using the same BPSK technique as used for $d(t)$. Thus, given a binary PN spreading sequence of N_c elements, $\{c\} = \{c_1, c_2, ..., c_{N_c}\}$, one period of spreading waveform $c(t)$ contains N_c chip intervals of duration T_c and is given by

$$c(t) = (-1)^{c_k} p(t) , \tag{2.7}$$

where

$$p(t) \equiv \begin{cases} 1, & t \in (0, T_c] \\ 0, & \text{otherwise} \end{cases} .$$

The plots in Fig. 2.7 graphically illustrate the DSSS waveform construction process. The plots in Fig. 2.7a and Fig. 2.7b represent the baseband data modulated and PN coded signals $d(t)$ and $c(t)$, respectively. These signals were generated using the BPSK bit-to-waveform mapping processes detailed in (2.6) and (2.7). For visual clarity, only four chip intervals per symbol duration were used ($T_{sym} = 1/R_{sym} = 4 \times T_c = 4/R_c$). The signal in Fig. 2.7c is the result of multiplying the data modulated signal $d(t)$ with the RF carrier $A \cos(2\pi f_c t)$. For visual clarity, only eight carrier frequency cycles per symbol duration were used ($T_{sym} = 8/f_c$). The carrier phase transitions in modulated carrier are readily apparent and correspond directly with phase transitions in $d(t)$. Finally, the signal in

15
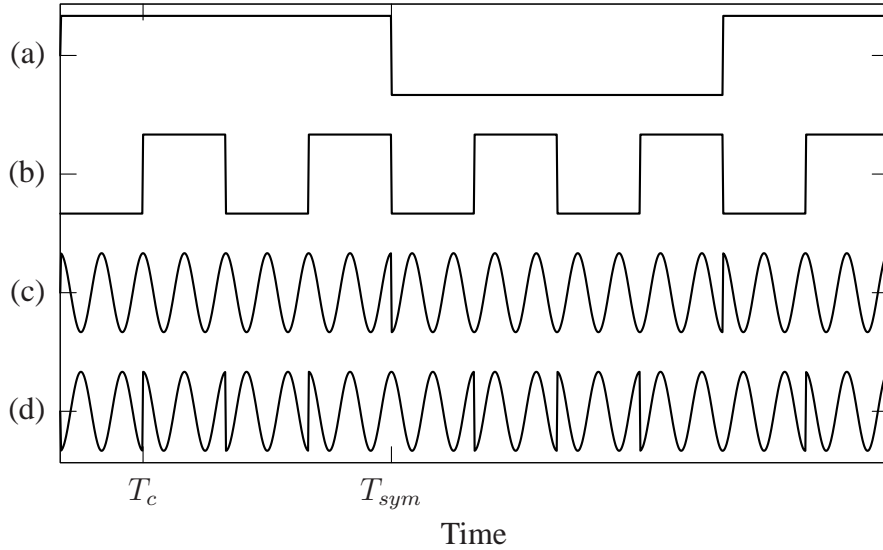
Figure 2.7: Waveform construction process for BPSK DSSS system: (a) baseband data modulated waveform $d(t)$, (b) baseband spreading modulation waveform $c(t)$, (c) carrier modulated waveform $d(t)\cos{(2\pi f_c t)}$, and (d) final DSSS waveform $c(t)d(t)\cos{(2\pi f_c t)}$ [25].

Fig. 2.7c is modulated by the PN coded signal $c(t)$ to create the transmitted signal shown in Fig. 2.7d.

The spectral spreading resulting from this process is illustrated in Fig. 2.8 which shows power spectral density (PSD) plots for the unspread data modulated carrier (dashed line with peak value P_{BPSK}) and the final spread waveform (solid line with peak value P_{DSSS}). Given that $T_{sym} = 4 \times T_c$ was used in this illustration, the spread signal PSD occupies four times more bandwidth than the unspread signal when measured between nulls. This bandwidth expansion is equivalent to what is commonly called the DSSS *processing gain* which is given by $G_p = R_c/R_{sym}$ for BPSK data modulation. It is also important to note that the peak value of P_{DSSS} one-fourth ($1/G_p$) the value of P_{BPSK} ($-6\ dB$ on a decibel scale). Thus, the DSSS signal power is effectively spread across a wider bandwidth than the original unspread signal.

The DSSS receiver essentially despreads and estimates communication symbols by repeating transmitter functions in reverse order. This is functionally illustrated in the block diagram shown in Fig. 2.9. The received DSSS signal enters the system and is first despread
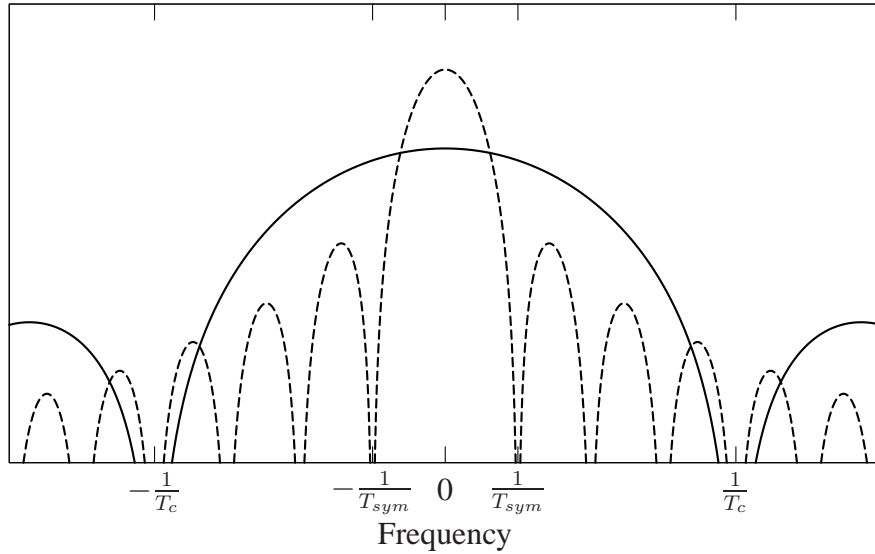
Figure 2.8:   PSD comparison of unspread data modulated carrier (dashed line) and final spread DSSS waveform (solid line). The amount of spectral spread is dictated by processing gain $GP_p = T_{sym}/T_c$ [25].

by mixing it with an estimate of the transmitted spreading waveform $c(t)$. To be effective when there is geographic separation between the transmitter and receiver, the receiver must have some *a priori* knowledge of the transmitted spreading code and thus the code cannot be purely random. The despread signal is bandpass filtered and the communication symbols estimated using techniques that are identical to systems employing no spread spectrum techniques.
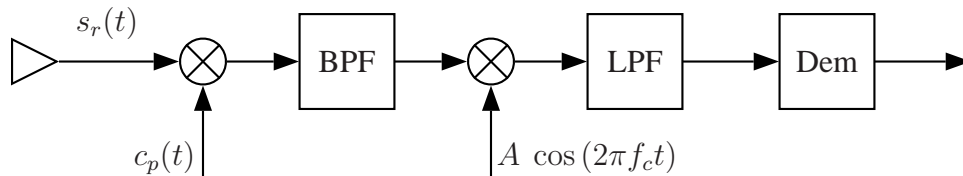
Figure 2.9:   Illustration of typical DSSS receiver architecture [25].

During DSSS receiver processing, the noise and other interfering signal components can be thought of as being *spread* versus *despread* given they do not contain the original spreading modulation. Thus, the composite *despread* waveform consists of two terms: 1) one due to the desired transmitted signal which returns to its original spectral shape and bandwidth, and 2) undesired noise and interfering signal contributions that are spectrally spread with reduced peak power levels. After post-despread filtering, a majority of the desired signal power remains (70% to 90% depending of the filter bandwidth) while only a fraction of the unwanted noise and interfering power remain. The portion of desired signal power that remains and the percentage of undesired power that is rejected is a function of the DSSS processing gain G_p. In general, the SNR improvement due to the despreading and filtering operation is proportional to G_p [25].

For a DSSS system to optimally estimate communication symbols, the DSSS receiver must have some knowledge of the received signal characteristics across the signaling domains: time, frequency, space, polarization, and code. Depending on the system, some of these parameters are known *a priori* while others must be estimated. In addition, the estimated parameters may change over time and must be constantly tracked and updated. For example, a communication system often operates at a pre-determined center frequency f_c. However, frequency variation in the received signal, due to Doppler frequency shift f_d resulting from relative transmitter-receiver motion and imperfect local oscillator behavior in both the transmitter and receiver, dictates that the receiver employ frequency and/or phase tracking which is typically accomplished with a Phase-Lock Loop (PLL). Next to PLL tracking stability, the next most important tracking requirement is perhaps the ability to reliably generate a local estimate of the received spreading code. The next section describes one common tracking method used for DSSS code tracking.

2.1.3 Delay-Lock Loop (DLL) Code Tracking. Spreading code tracking is perhaps the most important aspect in a DSSS system. While the transmitted code and code parameters are generally known *a priori* by the receiver, the relative time offset or delay of the code (sometimes called it phase) and chip-to-chip interval variation must be estimated

and tracked – this is referred to as *code tracking*. In a communication system, a lack of effective code tracking generally results in inefficient despreading of the received signal. This results in a lower SNR at the demodulator input and poorer bit error performance [15, 16]. In applications other than communications that also employ some form of code tracking, e.g., spread spectrum based navigation or radar systems, ineffective code tracking may result in poor delay estimates which ultimately lead to poorer position estimates [21].

To track the received PN coded waveform, the DSSS receiver must first search and find the relative code position using a process called acquisition. This process provides an initial course estimate of the code's temporal position and is used as a starting point for more accurate tracking by the code tracking loop. Once the tracking loop receives an initial temporal estimate of code position, it undergoes a process called pull-in, where it fine-tunes the estimated code position. Once it has adjusted the loop enters a state known a tracking, where it is providing a sufficiently accurate code estimate, in terms of temporal position, to effectively despread the received DSSS waveform. The required code tracking accuracy varies with system requirements.

Code tracking is generally performed using PLL techniques. The most commonly used architecture employs two parallel branches, including one representing an early (advanced) version of the current code estimate and the other representing a late (delayed) version of the current code estimate. This architecture is known as a Delay-Lock Loop (DLL) [32] and is the focus of this research. DLL implementations can be categorized as being either coherent or non-coherent. A coherent DLL uses knowledge of the received signal carrier frequency and phase to perform tracking. Therefore, the signal may be downconverted with the code tracking occurring at baseband. However, in many situations it can be difficult to estimate and track the carrier phase without first tracking the code for effective despreading. For this reason, the analysis in this dissertation concentrated on the non-coherent DLL, which typically operates at an intermediated frequency (IF) and does not require or assume knowledge of the carrier phase.
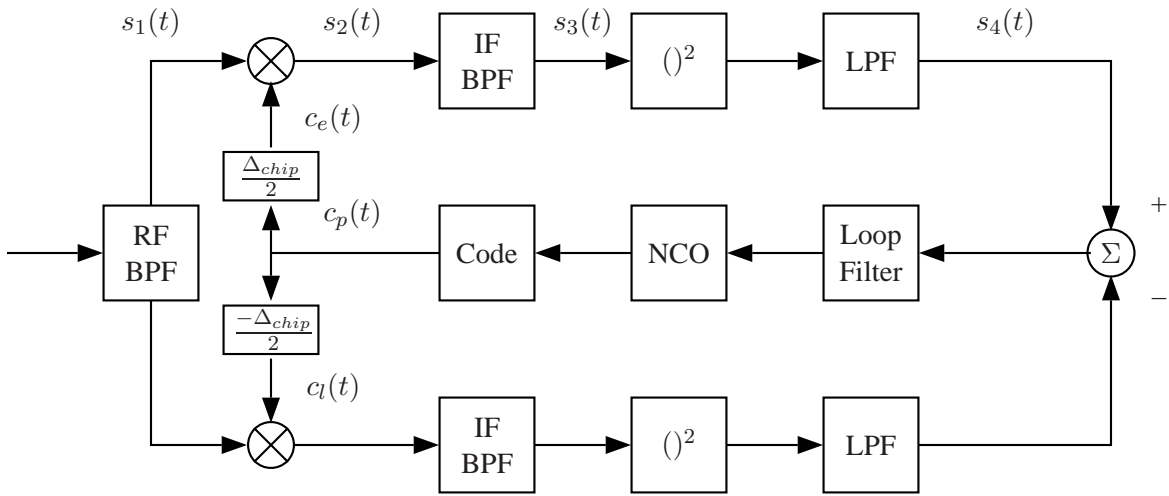
Figure 2.10:   Typical elements of non-coherent Delay-Lock Loop (DLL) [25].

The typical elements for implementing a non-coherent DLL are shown in Fig. 2.10. The two distinct parallel branches following the RF BPF are referred to as the *early* and *late* gate branches. These two branches are functionally equivalent beginning with the left-most despreading mixer and ending with the right-most LPF. What distinguishes the two branches is that the estimated code from the code generator is either delayed (arrives late) or advanced (arrives early) by some fixed amount before being applied to the despreading mixer. One intuitive way to understand DLL functionality is to observe how the time and frequency domain responses of a given input signal are modified as the signal progresses through the DLL. Given the early and late gate branches are functionally equivalent except for a code timing offset, the process is illustrated using $c_{e/l}(t)$ to represent either the early or late gate code.

Received signal $s_r(t) = d(t)c(t)cos\,[2\pi(f_c + f_d)t]$, where f_c is the transmitted carrier frequency and f_d is Doppler frequency shift, is first passed through the DLL RF BPF. Given an RF BPF bandwidth of $W_{RF} = 2R_{chip}$ and an arbitrary portion of $s_r(t)$ that spans two chip intervals, the DLL RF BPF produces the filtered time and frequency domain responses, $s_1(t)$ and $S_1(f)$, shown in Fig. 2.11 and Fig. 2.12, respectively. Selection of the RF BPF bandwidth is usually related to the main spectral response of the received signal.

20

This allows the filter to pass most of the desired signal energy while blocking undesired signal and noise responses falling outside the band of interest. Notice in Fig. 2.11 that less than ideal filtering removes the ideal rectangular shape across the chip boundary and that only the main spectral response remains in Fig. 2.12.

The remaining signal is then multiplied by an offset version of the baseband PN code. The time domain result $s_2(t)$ is shown in Fig. 2.13 and the frequency domain result $S_2(f)$ is in Fig. 2.14. The important observation at this point occurs in the frequency domain, where one can see that spectral lines are created at harmonics of the code repetition rate, R_c. The magnitude of the spectral line a f_c is used for code tracking. If the code is perfectly tracked, this spectral line will have the same magnitude in both the early and late branches of the DLL. If it is not perfectly tracked, then the magnitude of this spectral line in the two branches will differ.

To recover the desired spectral line at f_c, the signal is passed through the IF BPF. The bandwidth of this filter should be as narrow as possible to remove unwanted noise and harmonic effects, while being wide enough to ensure sufficient desired signal energy passes given that received signal frequency ($f_c + f_d$) is not precisely known. The resultant time and frequency domain responses of the DSSS signal at the DLL IF BPF output, $s_3(t)$ and $S_3(f)$ are shown in Fig. 2.15 and Fig. 2.16, respectively.

The magnitude (envelope) of the DLL IF BPF output effectively represents a measure of correlation between the received PN coded waveform and the DLL estimate of the same. The envelope is extracted using a basic energy detection process comprised of a squaring operation $(\bullet)^2$ followed by lowpass filtering (LPF). The result of applying this operation to the signal in Fig. 2.15 yields the time and frequency domain responses shown in Fig. 2.17 and Fig. 2.18, respectively.

The filtered difference between the early and late gate energy detector outputs, or discriminator output, provides a measure of relative code offset between the received and internally generated codes. The loop filter design is highly application specific and aims to maximize overall DLL stability. The discriminator output controls the numerically con-
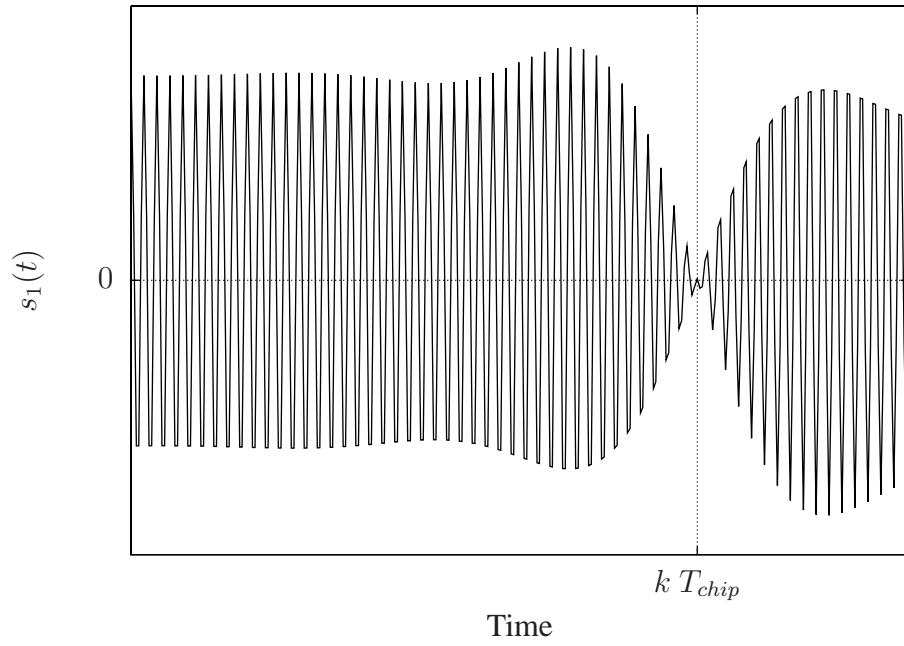
Figure 2.11: Time domain response of DSSS signal at the DLL RF BPF output.
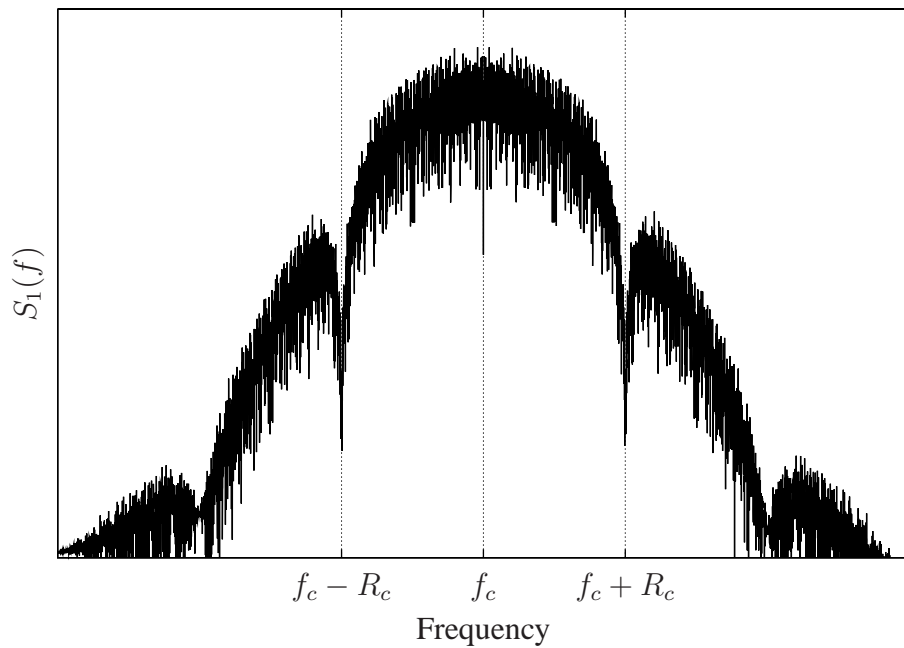
Figure 2.12: Frequency domain response of DSSS signal at the DLL RF BPF output.
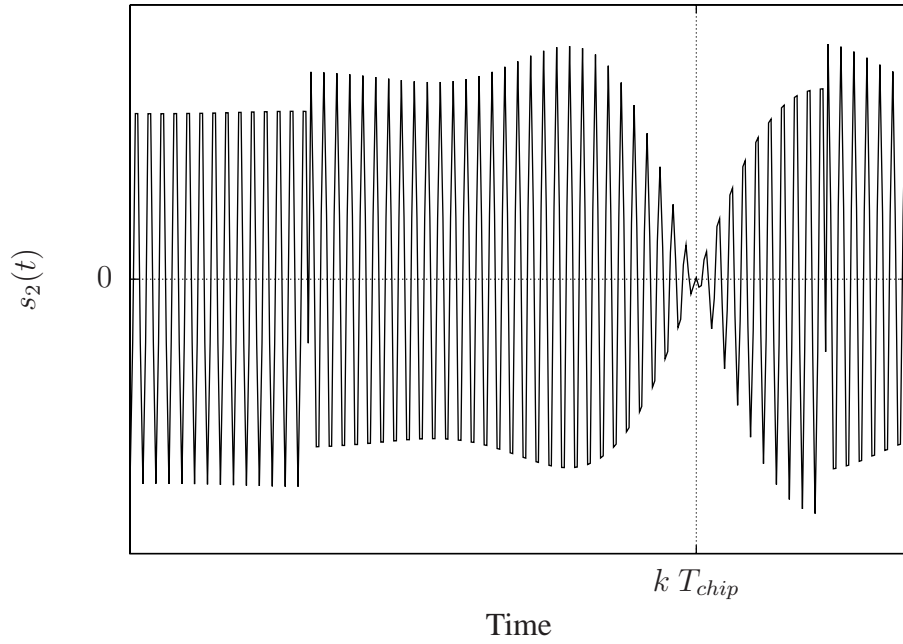
Figure 2.13:   *Unfiltered* time domain response of *despread* DSSS signal after multiplication by $c(t)$ under perfect code tracking conditions.
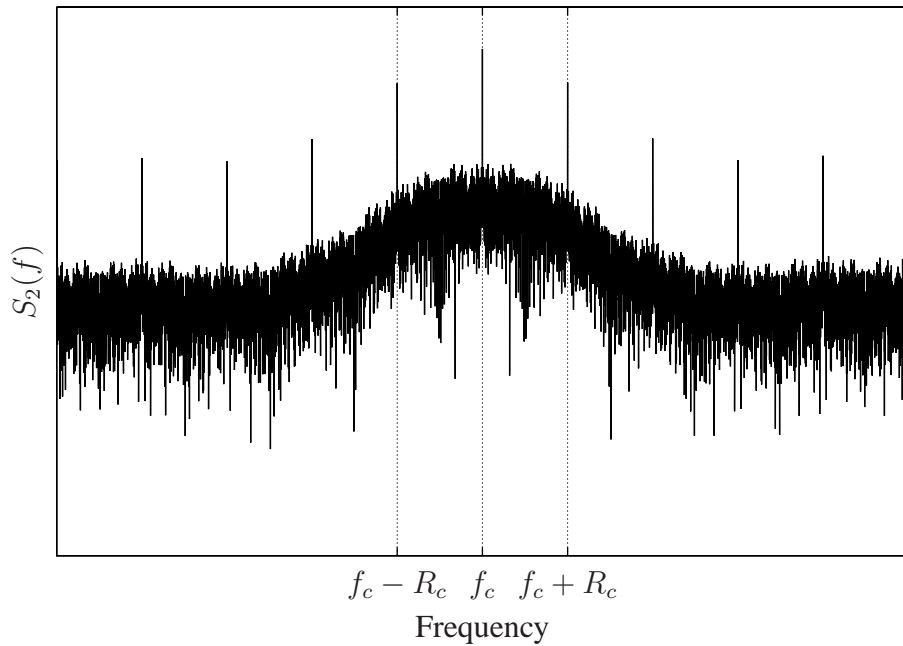
Figure 2.14:   *Unfiltered* frequency domain response of *despread* DSSS signal after multiplication by $c(t)$ under perfect code tracking conditions.
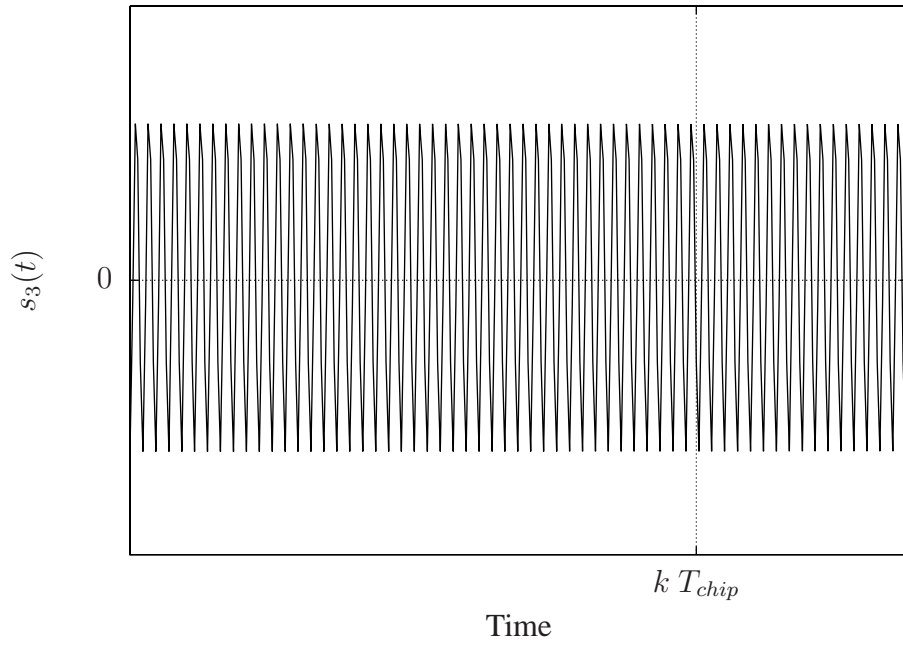
Figure 2.15:   *Filtered* time domain response of *despread* DSSS signal at DLL IF BPF output under perfect code tracking conditions.
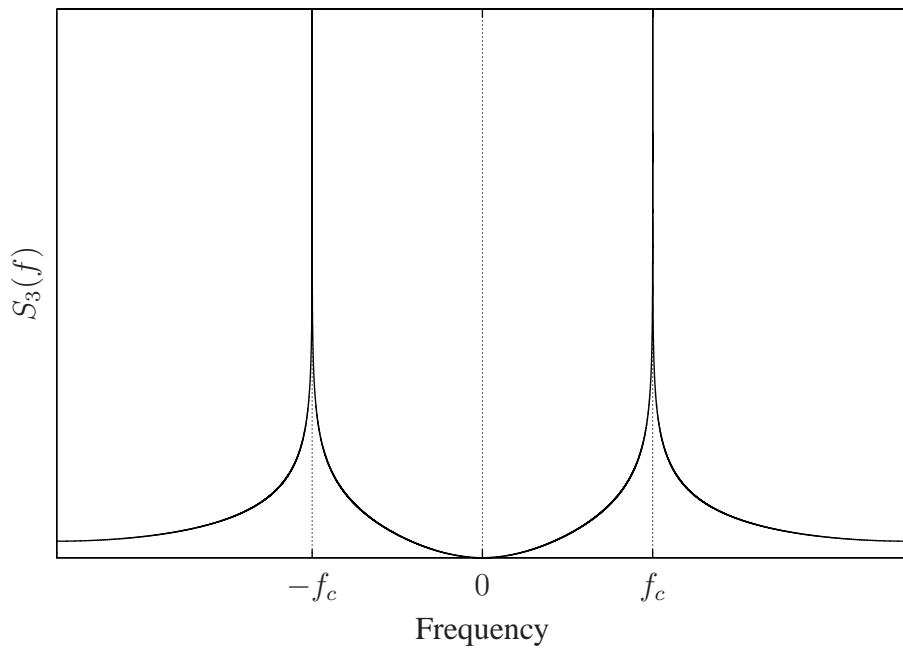
Figure 2.16:   *Filtered* frequency domain response of *despread* DSSS signal at DLL IF BPF output under perfect code tracking conditions.

trolled oscillator (NCO) which drives the PN code generator. If the current state of the internal code generator perfectly matches the received code, the discriminator output is zero and no change occurs in the NCO. Whenever the internal code generator state does not perfectly match the received code, a non-zero signed discriminator output is produced and the DLL code generator advances or delays its output accordingly (changes the code phase). For example, if the early gate correlates more closely with the received signal the discriminator output becomes positive and the NCO frequency increases. Likewise, if the late gate correlates more closely with the received signal the discriminator output becomes negative and the NCO frequency decreases. Thus, the DLL is constantly adapting to signal and channel conditions. This adaptation is desirable when compensating for actual varia-tions in $s_r(t)$ such as changes in carrier frequency f_c and Doppler frequency f_d. However, DLL tracking variation due to noise and/or interfering signals is generally undesirable. The DLL output discriminator response for changing signal conditions is generally described as an *S-curve*. An ideal *S-curve* response is shown in Fig. 2.19 for the case where no channel noise or interfering signals are present.
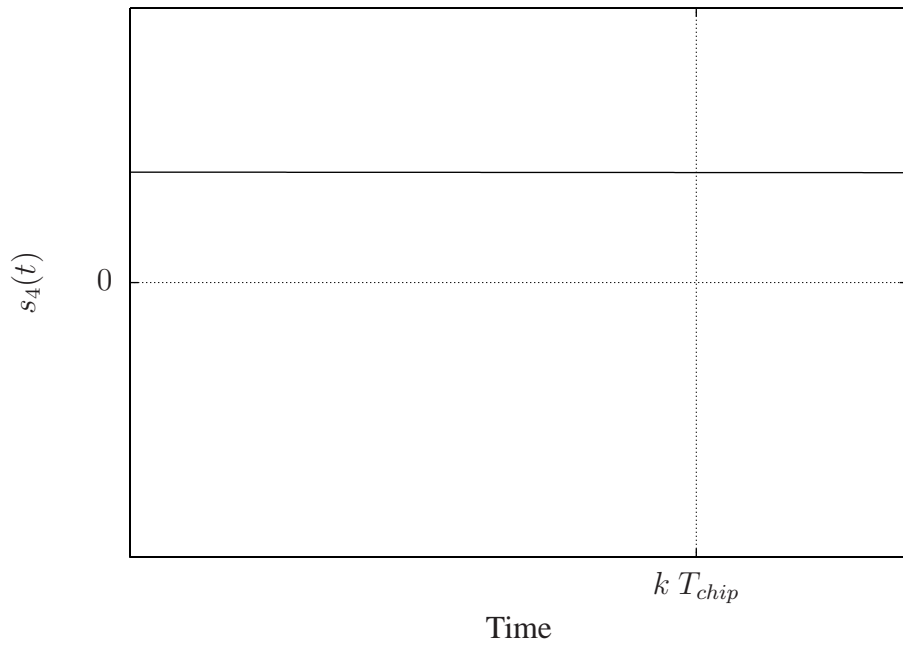
Figure 2.17:   Time domain response of DSSS signal at the energy detector output showing that double frequency and higher-order harmonics have been suppressed.
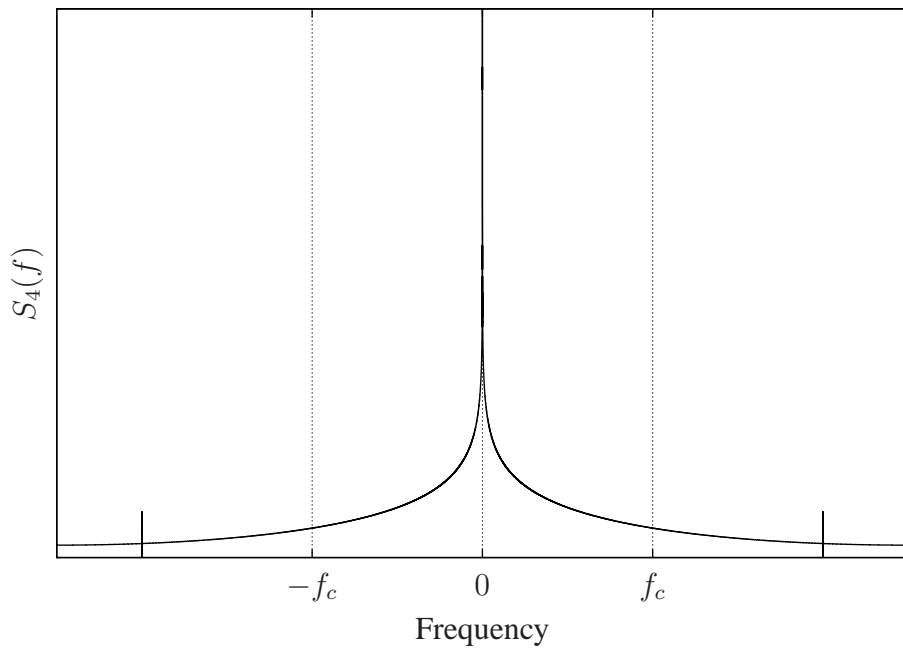
Figure 2.18:   Frequency domain response of DSSS signal at the energy detector output showing that double frequency and higher-order harmonics have been suppressed.
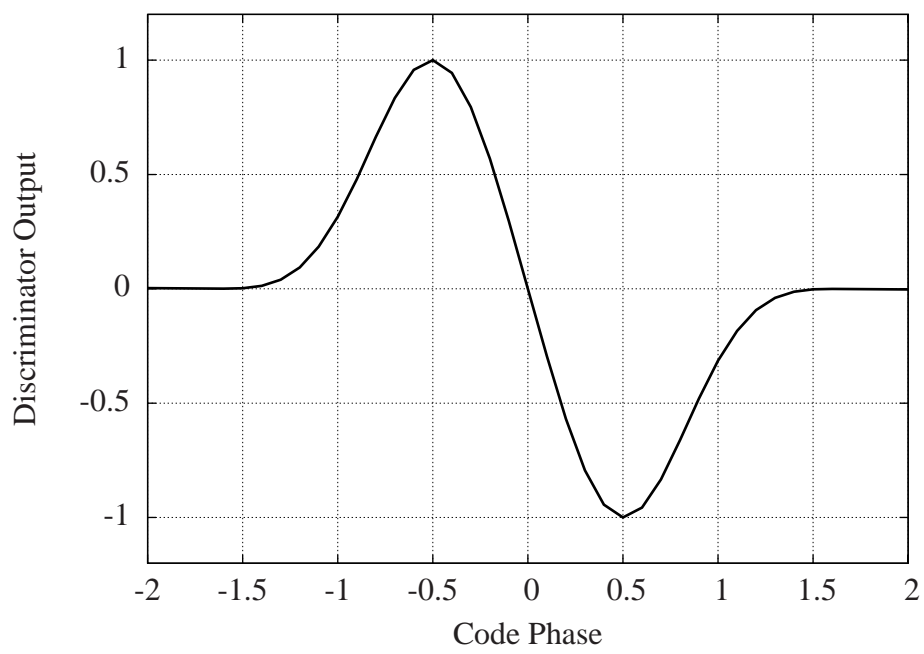
Figure 2.19:   Ideal S-curve response: DLL discriminator output versus code phase offset between the received and DLL estimated codes [25].

2.2 Optimization Techniques

Optimization techniques attempt to maximize or minimize the response of a mathematical model that represents a given physical system. The mathematical function to be maximized or minimized is called the *objective function*, which represents the system response to a particular set of *decision variables*. The relationship between the objective function and the decision variables is described by model *parameters* and *constraints*. In general, there may be many combinations of decision variables that satisfy the constraints, the collection of which is known as the *feasibility region*. Therefore, an optimized solution for the system is the "best choice" of decision variables that fall within the feasibility region – the particular definition of "best choice" is problem-specific [18].

2.2.1 Genetic Algorithm (GA). There are many optimization techniques which may be applied to a particular problem, with each having its own strengths and weaknesses. One large class of techniques is known as *heuristic methods* which generally attempt to find a "good" solution without necessarily striving to guarantee optimality [18]. While heuristic techniques tend to be very problem-specific, there are a handful of *metaheuristic* methods which may be applied to a broader range of problems. The GA is one popular class of metaheuristics which simulate the biological evolution process by describing the decision parameters using a binary string called a *gene*. The GA process essentially *mates* genes from an available population and retains the strongest *offspring* for subsequent mating in the new population. Some *parents* are retained in the new population and the possibility of *mutation* within offspring permitted [18]. The GA process can be summarized as follows:

1. The initial GA population is randomly generated and genes formed using the process illustrated in Fig. 2.20. The fitness of each generated gene is then calculated to ensure it is in the feasibility region. If in the feasibility region it is retained, else, it is rejected and another gene replaces it.

2. Some number of most fit (m) and least fit (l) genes, for ($l + m$) a multiple of two, are selected from the population to serve as parents. The selected parents are then randomly paired for mating to create $(m + l)/2$ parent pairs.

28
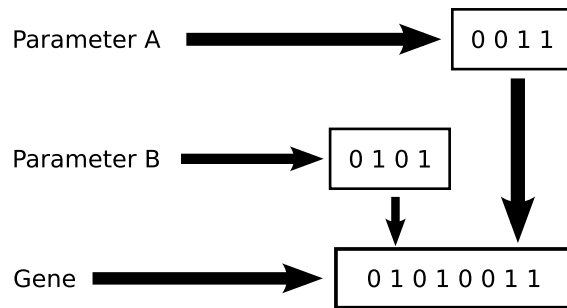
Figure 2.20: GA genes assembled for the initial population using two parameters with randomly generated binary digit values.

3. The binary digits of the parent genes are compared to generate offspring as shown in Fig. 2.21. When binary digits in the parents match, their corresponding values are passed directly to the offspring. When the binary digits differ (indicated by the **x** elements in the figure), the offspring digit values are randomly assigned with equal probability.

4. In addition, each digit in the selected offspring is subjected to some likelihood of mutation, i.e., there is some chance that its final value is complemented as illustrated in Fig. 2.22.

5. A new gene population is formed by retaining some of the previously mated parents and their offspring.

6. The selection-mating-selection process in Step 2 through Step 5 is repeated for several iterations (generations) until an exit criterion is met.

29
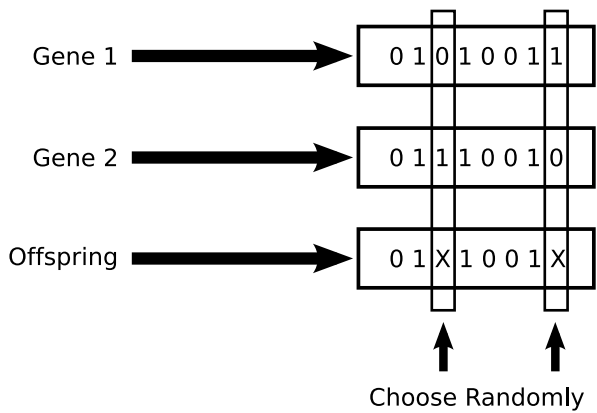
Figure 2.21: GA offspring generation process using a chosen parent gene pair from the population. The boxed columns denote dissimilar parent values where offspring values are randomly assigned. Matching parent values in unboxed columns are assigned directly to the offspring.
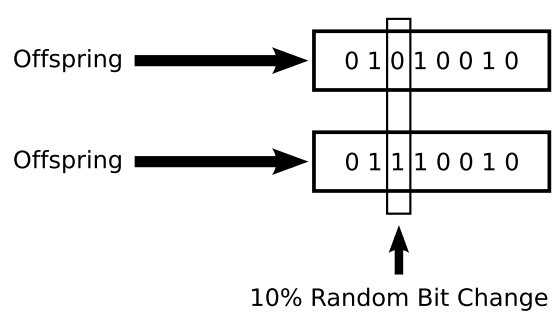
Figure 2.22: GA offspring mutation process. Each offspring digit is subjected to some likelihood of mutation.

2.2.2 Response Surface Methodology (RSM). RSM is a statistically-based alternative to GA optimization and proven itself in industrial experimentation and typically involves quality control. However, it is believed that this same approach may be useful in other types of engineering applications, including waveform design. Given it has a solid foundation in linear systems theory, it is readily accessible for communications design and consists of the following components [22]:

1. Design of Experiments (DOE), Section 2.2.2.1: A process for setting up experiments to ensure proper collection of data. If designed correctly, the experiment provides data that may be used for linear regression modeling and/or Analysis of Variance (ANOVA) processing [7, 22].

2. Linear Regression Modeling, Section 2.2.2.2: Used in RSM to estimate the shape of a response surface. If the response surface contains curvature over the tested region, the surface model may be used to solve for an optimal point. If the surface does not contain curvature, the regression model may still be used to search for a region that is quadratic [22].

3. Analysis of Variance (ANOVA), Section 2.2.2.3: A statistical analysis tool used to statistically compare the mean values of collected data. The ANOVA process is closely related to linear regression modeling and may be used to validate a given regression model, or to determine the significance of various model elements, including the presence of curvature in the response surface. Conclusions drawn from the ANOVA process are based on the assumption that the residual data is normally distributed and that the population variances are equal [7, 22].

4. Residual Testing, Section 2.2.2.4 through Section 2.2.2.7: Provides methods for testing ANOVA residual data distributions to verify whether or not the normality assumptions are satisfied. If the assumptions are not satisfied, various transformation techniques can be applied to remedy violations [14, 22].

5. Comparative Testing, Section 2.2.2.8: The ANOVA null hypothesis is that the mean values of the tested populations are equal. If the null hypothesis is rejected (means

31

are not equal), at least one of the population means is statistically different. In and of itself, the ANOVA process does not provide insight as to which population(s) are different. Thus, additional comparative testing is required to make this determination [14, 22].

2.2.2.1 Design of Experiments (DOE). DOE is a process that allows experimenters to craft trials that can statistically characterize or optimize a process or a system. The system under test may often be modeled as a linear process given by [22],

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{e} \,, \tag{2.8}$$

where $\mathbf{Y}$ is the system response, $\mathbf{X}$ is a matrix representing combinations of input parameters, β is the system model, and $\mathbf{e}$ is random error. Several goals may be accomplished with a design such as this. First, one may determine if the system model β adequately describes true system behavior. Second, one may be able to establish a model estimate $\hat{\beta}$. Third, if $\hat{\beta}$ can be determined, one can gain knowledge to optimize response $\mathbf{Y}$ given parametric variation in $\mathbf{X}$ [22].

The input parameters in matrix $\mathbf{X}$ are often expressed in terms of coded variables [22]. This is done by mapping between natural variables and coded parameters in $\mathbf{X}$. The input variables in this dissertation will generally be the SMSE waveform design variables N_f and Δf. Given these variables, a representative mapping from SMSE parameters (natural units) to DOE coded units is shown in Table 2.1.

Table 2.1:   Representative mapping from SMSE variables $(N_f, \ \Delta f)$ to DOE coded variables $(x_1, \ x_2)$.

Factor Level	N_f	x_1	Δf	x_2
Low	8	-1	17	-1
High	32	1	15	1

32

By way of illustrating the use of (2.8) with the mapping in Table 2.1, consider a first-order model with interaction. In this case, each element in the vector $\mathbf{Y}$ can be expressed as

$$y_{21} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + e . \tag{2.9}$$

Using this model to test the condition $N_f = 32$ and $\Delta f = 17$, the corresponding mapped values of $x_1 = 1$ and $x_2 = -1$ from Table 2.1 are used such that (2.9) is expressed as

$$y = \beta_0 + \beta_1(1) + \beta_2(-1) + \beta_{12}(1)(-1) + e.$$

A design that would test all possible combinations of $(N_f, \ \Delta f)$ as given in Table 2.1 is called a *full factorial design* [22] and would include all elements $\mathbf{Y}$ given by

$$
\begin{aligned}
y_{11} &= \beta_0 + \beta_1(-1) + \beta_2(-1) + \beta_{12}(-1)(-1) + e \\
y_{12} &= \beta_0 + \beta_1(-1) + \beta_2(1) + \beta_{12}(-1)(1) + e \\
y_{21} &= \beta_0 + \beta_1(1) + \beta_2(-1) + \beta_{12}(1)(-1) + e \\
y_{22} &= \beta_0 + \beta_1(1) + \beta_2(1) + \beta_{12}(1)(1) + e .
\end{aligned}
$$

This set of equations may be expressed in the matrix form given by (2.8) using

$$\mathbf{Y} = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{bmatrix}, \tag{2.10}$$

$$
\mathbf{X_{FF}} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{2.11}
$$

and

$$
\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \end{bmatrix}, \tag{2.12}
$$

where subscript values in $\mathbf{Y}$ and β correspond to ordered coded variables (x_1, x_2) and $\mathbf{X_{FF}}$ denotes a full factorial design.

If experimental replication is desired for statistical analysis, which is often the case for optimization, the input parameters in $\mathbf{X_{FF}}$ may be repeated. This is accounted for using

$$
\mathbf{X} = \begin{bmatrix} \mathbf{X_{FF}} \\ \mathbf{X_{FF}} \\ \vdots \\ \mathbf{X_{FF}} \end{bmatrix}, \tag{2.13}
$$

where $\mathbf{X}$ now represents the total experiment and $\mathbf{X_{FF}}$ represents one repetition of the full factorial design [22]. As provided, the expressions in (2.8), (2.10), (2.11), and (2.12) provide a full experimental description that may be used directly to analyze system response $\mathbf{Y}$ [22].

2.2.2.2 Linear Regression Modeling. Once an experiment has been completed according to (2.8), the system response $\mathbf{Y}$ (collected data) may be used to generate a

34

model estimate $\hat{\beta}$. This may be done using a *linear regression* process by solving the least squares normal equation given by [22]

$$\hat{\beta} = \left(\mathbf{X^TX}\right)\mathbf{X^TY} \,, \tag{2.14}$$

with the predicted values of $\mathbf{Y}$ then given by

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} \,. \tag{2.15}$$

The resultant *residuals* $\mathbf{e}$ (error) are then calculated as the difference between the collected and predicted values of $\mathbf{Y}$ according to [22]

$$\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} \,. \tag{2.16}$$

If the model estimate $\hat{\beta}$ provides a good prediction of response $\mathbf{Y}$ given input $\mathbf{X}$, then the residual elements in $\mathbf{e}$ will be normally distributed random variables. Furthermore, the variance of $\mathbf{e}$ under each condition in $\mathbf{X}$ should be approximately equal [22]. Tests for quantifying normality and variance equality among the residual elements are described in Section 2.2.2.5 and Section 2.2.2.6, respectively. If the model estimate $\hat{\beta}$ indicates there is curvature in the response surface $\mathbf{Y}$, it may be used to optimize $\mathbf{Y}$, i.e., $\hat{\beta}$ may be used to specify optimal input parameters in $\mathbf{X}$ to either maximize or minimize response surface $\mathbf{Y}$ [22].

Two β models are particularly useful for RSM: the second-order model and the first-order model [22]. The second-order model is given by

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 \,. \tag{2.17}$$

If the estimated model parameters from (2.17) adequately describes the data (as determined by the ANOVA), then the surface is assumed to contain curvature and optimal input param-

eters x_1 and x_2 may be calculated. Using elements expressed by (2.17), the second-order model may be expressed in matrix form as

$$\mathbf{Y} = \beta_0 + \mathbf{X^T b} + \mathbf{X^T B X} \,, \tag{2.18}$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \,,$$

$$\mathbf{b} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} \,,$$

and

$$\mathbf{B} = \begin{bmatrix} \hat{\beta}_{11} & \frac{\hat{\beta}_{12}}{2} \\ \frac{\hat{\beta}_{12}}{2} & \hat{\beta}_{22} \end{bmatrix} \,.$$

A *stationary point* for the second-order model is given by [22]

$$\mathbf{X_s} = -\frac{1}{2}\mathbf{B^{-1} b} \,. \tag{2.19}$$

The derivation of stationary point $\mathbf{X_s}$ is accomplished by setting the derivative of the estimated model $\hat{\beta}$ to zero and solving for $\mathbf{X}$ [22]. Therefore, the stationary point must be tested to determine if it creates a maximum, minimum, or saddle point in response surface $\mathbf{Y}$. This is done using Eigenvalue analysis. If all Eigenvalues of $\mathbf{B}$ are positive, $\mathbf{X_s}$ mini-

mizes $\mathbf{Y}$. If all Eigenvalues are negative, $\mathbf{X_s}$ maximizes $\mathbf{Y}$. If the Eigenvalues signs for $\mathbf{B}$ differ, the point $\mathbf{X_s}$ corresponds to a saddle point on $\mathbf{Y}$ [22].

If, however, when using the second-order model of (2.17) the resultant ANOVA does not determine the response surface $\mathbf{Y}$ contains curvature, a first-order model of the following form may be used [22]

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \,. \tag{2.20}$$

In this case, the path of steepest ascent/descent, given by

$$\frac{\Delta x_1}{\hat{\beta}_1} = \frac{\Delta x_2}{\hat{\beta}_2} \,, \tag{2.21}$$

may be followed in subsequent experiments until the second-order model becomes appropriate. Then, a stationary point may be found by (2.19) [22].

2.2.2.3 Analysis of Variance (ANOVA).   The ANOVA process is the heart of DOE optimization. ANOVA is a statistical technique which uses the sample variances of a data set to test the impact of input parameters on an output response variable. Specifically, given an experimental design $\mathbf{X}$ and model β, the ANOVA process uses the sample variances to test the null hypothesis, i.e., "Are the means of different conditions described by the model equal?" Thus, rejection of the null hypothesis implies that at least one of the means are different [22].

There are many ways to interpret a null hypothesis rejection. In this analysis, the ANOVA is primarily used in two ways. First, rejecting the null hypothesis with respect to the entire regression model signifies that $\hat{\beta}$ adequately describes the process [22]. Second, the ANOVA is used to evaluate the individual input parameters of $\mathbf{X}$. In this case, rejecting the null hypothesis for an given input parameter indicates that parameter has a significant effect on response surface $\mathbf{Y}$ and should be included in the model for optimization [22]. As a result, ANOVA provides insight into the response surface shape by including appropriate

model terms while eliminating those that are inappropriate. The ANOVA process begins by assuming a model for the test data [7]. For this illustration, the assumed model is *second-order* and expressed as [22]

$$
\begin{aligned}
y_{ijk} &= \beta_0 + \beta_1 x_{1_i} + \beta_2 x_{2_j} + \beta_{12} x_{1_i} x_{2_j} \\
&+ \beta_{11} x_{1_i}^2 + \beta_{22} x_{2_j}^2 + e_{ijk} ,
\end{aligned}
\tag{2.22}
$$

where y is the response variable, (x_1, x_2) are coded input variables, the β terms represent regression coefficients and e represents the error. The i and j subscripts correspond to index values of x_1 and x_2, respectively. The k subscript represents the experimental run number. The expression in (2.22) may be expressed more compactly in matrix form as

$$
\mathbf{Y} = \mathbf{X}\beta + \mathbf{e} .
\tag{2.23}
$$

Essentially, the ANOVA process performs a least-squares fit of the data to the model by applying [7, 22]

$$
\hat{\beta} = \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathbf{T}\mathbf{Y} .
\tag{2.24}
$$

Substituting this estimate for β into (2.23) yields a regression model of the form

$$
\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}\left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathbf{T}\mathbf{Y} ,
\tag{2.25}
$$

with the resultant error given by

$$
\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} .
\tag{2.26}
$$

After the least-squares fit to the data, the sample variance is partitioned into sub-spaces corresponding to the main effects, interactions and error. The main effects variance

and interaction variances are compared to the error variance as part of a significance test to determine their relative importance to the model. More specifically, assuming that the underlying model error is Gaussian, the ratio of a given factor's sample variance to the error's sample variance will be distributed according to the F distribution. Tabulated F distribution values may then be used to provide probabilities that the given factor under test is significant and should be included in the model. The larger the value of test statistic F_0, the more reasonable it becomes to reject the null hypothesis of equal means. An alternative approach uses a p-value which represents the probability that the statistic F_0 is the result of a population that corresponds to the null hypothesis. If the p-value is small, it casts doubt on the null hypothesis. Therefore, ANOVA provides a tool for deciding which factors and interactions are most important in a given system model, a task which is often left to intuition [7, 22]. More detailed information regarding ANOVA processing can be found in [22].

2.2.2.4 Normality and Variance. The normality and equal variance assumptions that were noted in Section 2.2.2.2 and Section 2.2.2.3 are important to both linear regression modeling and ANOVA processing [22]. The following subsections address each of these conditions.

2.2.2.5 Normality Testing. The residuals e are assumed to be normally distributed [22] in linear regressing modeling and ANOVA processing. A visual test such as a probability plot, or a numerical method such as the Shapiro-Wilks test, can be used for testing normality [14].

A representative normal probability plot of residuals is shown in Fig. 2.23. This is essentially a plot of the residuals (x-axis) versus the cumulative probability of the normal distribution (y-axis). If the residuals appear to lie on a straight line they are distributed approximately normally [14, 22].

The Shapiro-Wilks test provides a significance test for normality [14]. In this case, the null hypothesis is that the data is normally distributed. To execute the test, the residuals
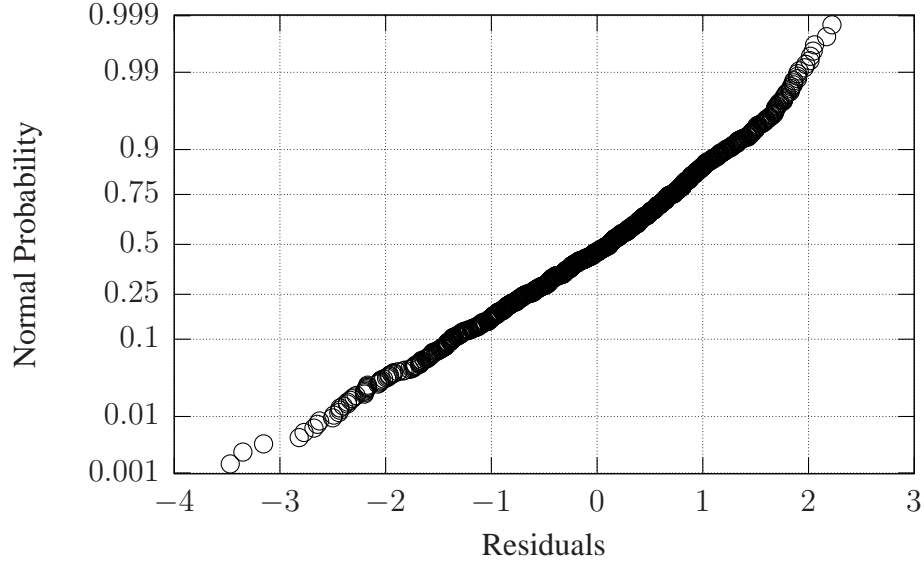
Figure 2.23: Normality test using residuals versus cumulative probability of normal distribution. If residuals lie along a straight line the data may be considered normally distributed [22].

e are ordered such that $e_1 \leq e_2 \leq \cdots \leq e_{N_e}$, where N_e represents the total number of residuals. Test statistic W is then calculated using

$$S^2 = \sum_{i=1}^{N_e} \left(e_i - \overline{e}\right)^2 \ , \tag{2.27}$$

$$k = \begin{cases} \frac{N_e}{2}, & N_e \text{ even} \\ \frac{N_e - 1}{2}, & N_e \text{ odd} \ , \end{cases} \tag{2.28}$$

$$b = \sum_{i=1}^{k} a_{N_e - i + 1} \left(e_{N_e - i + 1} - e_i\right) \ , \tag{2.29}$$

$$W = \frac{b^2}{S^2} \ , \tag{2.30}$$

where $\overline{e}$ in (2.27) is the residual mean and tabulated values of $a_{N_e - i + 1}$ in (2.29) are provided in [14]. The resultant value of W from (2.30) is then mapped to a tabulated p-value which can be found in [14]. Given that the null hypothesis is normally distributed data, a small p-

40

value from the Shapiro-Wilks casts doubt on this assumption and causes the null hypothesis to be rejected [14].

2.2.2.6   Variance Testing.   Variance equality testing can be done using a Bartlett's significant test. In this case, the null hypothesis is that the residual variances under all experimental conditions are equal.

Assuming there are a variances to compare with each having n_i samples, the total number of samples N_e is given by

$$N_e = \sum_{i=1}^{a} n_i .$$
(2.31)

To test the null hypothesis that $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_a^2$, the test statistic χ_0^2 is calculated by

$$S_p^2 = \frac{1}{N_e - a} \sum_{i=1}^{a} (n_i - 1) S_i^2,$$
(2.32)

$$q = (N_e - a) \log_{10} \left(S_p^2 \right) - \sum_{i=1}^{a} (n_i - 1) \log_{10} \left(S_i^2 \right),$$
(2.33)

$$c = 1 + \frac{1}{3(a-1)} \left[\sum_{i=1}^{a} (n_i - 1)^{-1} - (N_e - 1)^{-1} \right],$$
(2.34)

$$\chi_0^2 = 2.3026 \frac{q}{c},$$
(2.35)

where S_i^2 represents the sample variance of the ith condition in the experiment [22]. The test statistic χ_0^2 is used to find a tabulated p-value from a χ_{a-1}^2 distribution. If the p-value obtained from a χ_{a-1}^2 distribution is small, this suggests that the null hypothesis should be rejected and the data does not meet the variance equality condition [22].

2.2.2.7   Statistical Transformation.   If the normality conditions and/or variance equality conditions in Section 2.2.2.5 and Section 2.2.2.6 are not satisfied, it may be

possible to transform the data so the conditions are met. The Box-Cox transformation is one common transformation for accomplishing this. The Box-Cox transformation is given by [22]

$$\mathbf{T} = \begin{cases} \frac{\mathbf{Y}^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \ln(\mathbf{Y}) & \lambda = 0 \, , \end{cases} \tag{2.36}$$

where $\lambda \in R$. As indicated, data $\mathbf{Y}$ may be transformed using a range of λ values. If a value of λ exists such that the residuals of the transformed data $\mathbf{T}$ meet the requirements for normality and equality of variance, then linear regression and/or the ANOVA may be performed on the transformed data, $\mathbf{T}$ [22]. However, if the residuals from $\mathbf{T}$ still does not satisfy normality and variance equality conditions, the ranks of the data can be analyzed as part of the ANOVA process. When ranks are used, the test is known as the Kruskal-Wallis test [22].

2.2.2.8 Comparative Testing. The ANOVA process is a hypothesis test that provides information about the statistical data means for conditions under test. The null hypothesis is that all means are equal and its rejection indicates that at least one mean differs from the others. However, in and of itself the ANOVA process does not provide an indication of which mean(s) is different. Comparative testing can be performed to determine this. In addition to providing information about the ANOVA results, comparative testing is also important to quantify results of the optimization process [22].

One visual tool for comparing population statistics for random variables is the box and whisker plot as illustrated in Fig. 2.24 [22]. This plot shows statistical properties of the correlation degradation metric C_{Deg} under various SMSE parameter combinations. The box and whisker plot is interpreted as follows for a given SMSE parameter combination: 2) the box midline represents the median value, 2) the top and bottom box edges represent quartiles for the 25th and 75th percentiles of the populations, and 3) the extreme "whisker" ends represent the minimum and maximum population values. The box and whisker representation reveals general trends about the data means and data spread [22].
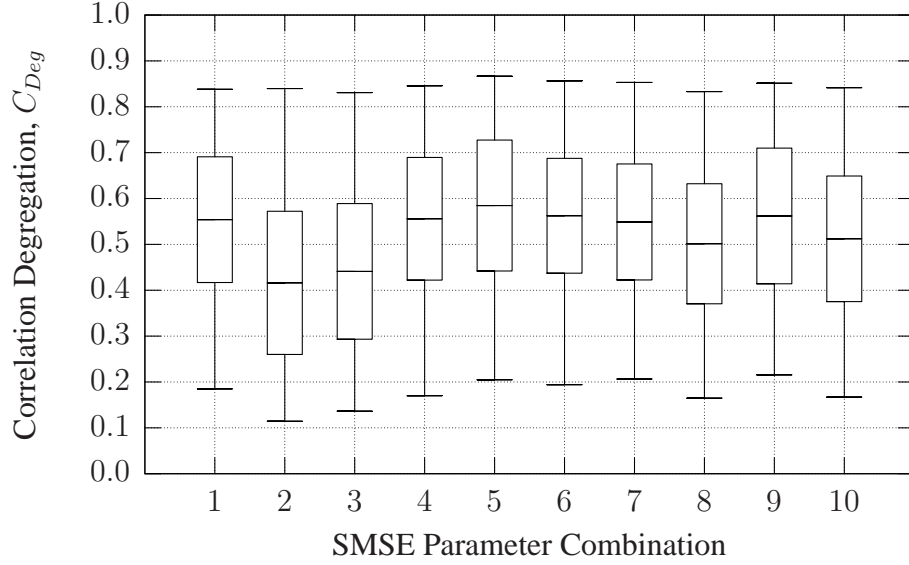
Figure 2.24: Representative box and whisker plot showing population medians (box midline), quartiles for 25th and 75th percentiles (box top and bottom), and population extreme values (whisker ends) [22].

To formally compare the means of two normally distributed, equal variance populations, a t-test may be used according to the following [22]. Given there are n_i samples in the ith population, with $\overline{y}_i$ being the mean value for the the ith population, test statistic t_0 for the two mean case can be calculated as follows:

$$S_i^2 = \frac{1}{n_i - 1} \sum_{k=1}^{n_i} \left(y_k - \overline{y}_i\right)^2 \quad i = 1, 2 \;, \tag{2.37}$$

$$S_p^2 = \frac{(n_1 - 1) S_1^2 + (n_2 - 1) S_2^2}{n_1 + n_2 - 2} \;, \tag{2.38}$$

$$t_o = \frac{\overline{y}_1 - \overline{y}_2}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \;. \tag{2.39}$$

Test statistic t_0 from (2.39) is used to find a tabulated p-value from a $t_{n_1+n_2-2}$ distribution. If the p-value obtained from the $t_{n_1+n_2-2}$ distribution is small, this suggests that the null hypothesis should be rejected and the means $\overline{y}_1$ and $\overline{y}_2$ do indeed differ [22].

43

To compare means for more than two populations, the Least Significant Difference (LSD) test may be used. Assuming there are a total means to compare with each based on n_i samples, the total number of samples N_e is given by

$$N_e = \sum_{i=1}^{a} n_i \,.$$

(2.40)

In this case, the LSD test for a full-factorial design is given by

$$\text{LSD} = t_{\frac{\alpha}{2}, N_e - a} \sqrt{\frac{2\, \mathbf{e}^T \mathbf{e}}{n(N_e - a)}} \,,$$

(2.41)

where α is the significance level and $\nu = N_e - a$ is the number of degrees of freedom [22]. When comparing any two populations in the experiment, the null hypothesis of equal means ($\overline{y}_i = \overline{y}_j$) would be rejected if the means differ by more than the LSD [22].

III. Methodology

In many respects, the goal of this dissertation is to develop and demonstrate a set of tools that waveform design engineers can use to design efficient communication waveforms. One of the basic tools adopted for this research is the Spectrally Modulated, Spectrally Encoded (SMSE) framework as described in Section 2.1.1. The SMSE framework has been successfully used to mathematically describe various types of Orthogonal Frequency Division (OFDM) waveforms. The quantitative nature of the SMSE framework makes it well-suited for applying design techniques from disciplines outside of communications and/or signal processing. More specifically, the SMSE framework enables a systematic approach to waveform design from *operations research* – a field of study dedicated to the various forms of optimization [18].

Coexistent interference occurs when two or more communication systems operate without orthogonality in frequency, time, space, polarization, and/or coding. With limited available bandwidth, it is often necessary for signals to spectrally coincide while inducing "manageable" levels of mutual interference. When user requirements dictate fundamentally different waveform modulations, the system design procedure often involves trial and error to find waveforms which can *coexist* [25, 31]. As commonly employed in the operations research field, the *Genetic Algorithm* (GA) and *Response Surface Methodology* (RSM) techniques, as described in Section 2.2.1 and Section 2.2.2, represent two approaches that can be considered for waveform design *optimization* to provide a more structured, optimal means for determining waveform design parameters [3, 4].

Each of these optimization techniques are employed here to demonstrate SMSE waveform design in a coexistent scenario containing an SMSE signal and a Direct Sequence Spread Spectrum (DSSS) system operating over an Additive White Gaussian Noise (AWGN) channel. The DSSS system employs a non-coherent Delay-Lock Loop (DLL) as described in Section 2.1.3 for code tracking prior to data demodulation. It is important to note that the particular DLL implementation used here, and the metric introduced in Section 3.2.3 to characterize various tracking conditions (perfect and imperfect), are sufficiently general such that the optimization demonstrations herein are broadly applicable to

other non-communication applications employing DLL tracking, e.g., precision navigation, timing, geolocation, etc.

The coexistent SMSE-DSSS waveform design process is demonstrated under two conditions, including: 1) *perfect* DLL code tracking as described in Section 3.1 and 2) *imperfect* DLL code tracking as described in Section 3.2. Under *perfect* DLL code tracking conditions, the GA and RSM optimization processes are considered independently in Section 3.1.3 and Section 3.1.4, respectively. In both cases, the optimization goal is to determine SMSE parameter values for the number of IFFT points N_f and subcarrier spacing Δf such that DSSS bit error rate P_b is optimized (minimized or maximized). Under *imperfect* DLL code tracking conditions in Section 3.2, the GA and RSM techniques are sequentially combined into a hybrid optimization process that includes: 1) the GA process being applied in Section 3.2.4 to generate a "coarse" solution for initial RSM processing, and 2) the RSM process providing the final optimized solution in Section 3.2.5. As in the perfect code tracking case, the end goal is to determine the $(N_f, \Delta f)$ pair that optimizes P_b. However, P_b optimization (minimization or maximization) is actually accomplished through a Correlation Degradation metric C_{Deg} as introduced in Section 3.2.3. The statistical behavior of C_{Deg} is used to capture and characterize overall DLL code tracking performance.

3.1 Perfect DLL Code Tracking

3.1.1 Coexistent SMSE-DSSS Scenario. The following signal conditions were used for coexistent SMSE-DSSS scenario under *perfect* DLL code tracking conditions.

The SMSE signal was generated according to the framework described in Section 2.1.1. Two SMSE factors (design parameters) were varied for the experiments, including: 1) the total number of IFFT points N_f and 2) the subcarrier frequency separation Δf. The remaining SMSE design parameters in (2.4) were fixed such that conventional OFDM was implemented [34], i.e., $\mathbf{s_k} = \mathbf{d_k}$. The complex baseband OFDM symbols were generated using independent BPSK data modulation on all subcarriers and carrier modulated to f_c for coexistent demonstrations. The carrier modulated SMSE waveform occupies a total bandwidth of $W_{SMSE} = 2 \times N_f \times \Delta f$ and has a duration of $T_{OFDM} = 1/R_{OFDM} = 1/\Delta f$.

The SMSE data bits were randomly generated with equal probability. Therefore, every transmitted SMSE symbol was random and there were no special bit sequences used to simulate frames, packets, etc. Finally, there was no cyclic prefix used for the waveform design demonstrations and analysis.

The coexistent DSSS system used BPSK for both data and spreading modulations. For demonstration purposes, the spreading code was an $N_c = 32$-bit Hadamard sequence with exactly one code period (32 chip intervals) occurring per data symbol. The DSSS symbol duration was $T_{sym} = 1 \; sec$, and the symbol rate was $R_{sym} = 1/T_{sym} = 1 \; Hz$. Consequently, the chip rate was $R_{chip} = N_c \times R_{sym} = 32 \; Hz$. The DSSS receiver was *perfectly synchronized* to the transmitted DSSS signal in terms of carrier tracking (frequency and phase), spreading code tracking, and communication symbol tracking. An ideal RF filter was used and communication symbols were estimated using a single channel correlation receiver under Maximum Likelihood (ML) conditions.

3.1.2   Optimization Metric.   The optimization metric under perfect DLL code tracking tracking conditions was end-to-end DSSS system P_b, as determined by Monte Carlo simulation of a physics-based analytic model. The model assumed that the coexistent SMSE and DSSS signals were spectrally coincident (same center frequency) and were operating over an AWGN channel. Therefore, the resultant DSSS bit errors are due to a combination of channel noise and the coexistent SMSE signal. However, the channel noise power was fixed during both the minimization and maximization demonstrations. Thus, the DSSS P_b curves in Section 3.1 correspond to bit error change as a function of interfering signal power and optimized input SMSE parameters. The changes are not due to differing noise power.

3.1.3   Genetic Algorithm (GA).   Each combination of the two optimization input parameters $(N_f, \Delta f)$ were represented using eight binary digits in a gene. Furthermore, the number of SMSE subcarriers was constrained to be an integer power of two with $N_f \in [1, 128]$. Similarly, the SMSE subcarrier spacing was assigned an integer value satisfying $\Delta f \in [1, 33]$. The feasibility region for optimization included all possible combinations

of N_f and Δf within these ranges. For the special case of $N_f = 1$, the resultant SMSE waveform is equivalent to a single BPSK modulated subcarrier and has an RF bandwidth of $W_{SMSE} = 2/T_{OFDM} = 2/\Delta f$, where Δf is used here and in presenting subsequent $N_f = 1$ results to more appropriately refer to waveform bandwidth and not subcarrier spacing. Finally, the objective function was end-to-end DSSS (P_b).

Consistent with the process described in Section 2.2.1, the GA process proceeded as follows:

1. The initial GA population consisted of ten randomly generated genes.

2. Using Monte Carlo simulation of a physics-based analytic model, the DSSS P_b was calculated for each case to judge each genes' *fitness*.

3. For mating, four of the five most fit genes and two of the least fit genes were chosen as parents. The resulting six parents were randomly assigned for mating, creating three pairs of parents.

4. To mate, the binary digits of the parents' genes were compared. When the binary values of parent genes matched, the same values were passed to the offspring. Where the values differed, the offspring values were randomly chosen with equal probability. Each pair of parents created two offspring.

5. After mating, each binary digit in the offspring's gene was subjected to a 10% chance of mutation, or complementing the bit value.

6. The next population of ten genes included the six children and the four most fit parents. The GA process then proceeded as it did with the initial/previous population of ten genes. For design demonstrations in this dissertation, the GA process was repeated for 100 generations.

The GA optimization process was used to both minimize and maximize DSSS P_b. Minimizing P_b creates what could be called "peaceful" coexistence while maximizing P_b creates worst-case coexistence conditions. Results for GA optimization under *perfect* code tracking conditions are provided in Section 4.1.2.

3.1.4 Response Surface Methodology (RSM). The RSM process was also used to optimize SMSE parameters $(N_f, \Delta f)$ such that coexistent DSSS P_b performance was both maximized and minimized. The RSM experiments assumed a second-order model as described in (2.17). As a result, the experimental design consisted of a two-factor, three-level, full-factorial design with four additional center runs. The corresponding system model from (2.8) for this demonstration is expressed as

$$
\mathbf{Y} = \begin{bmatrix}
1 & -1 & -1 & 1 & 1 & 1 \\
1 & -1 & 0 & 0 & 1 & 0 \\
1 & -1 & 1 & -1 & 1 & 1 \\
1 & 0 & -1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_{12} \\
\beta_{11} \\
\beta_{22}
\end{bmatrix} .
\tag{3.1}
$$

The SMSE parameters were optimized using the steepest ascent/descent process described in (2.21) until the response surface fit the second-order model, as determined by the ANOVA. Once the second-order model detected curvature, (2.19) was used to solve for the SMSE parameters in terms of their coded variables. Results for the RSM optimization process under *perfect* code tracking conditions are provided in Section 4.1.3.

3.2 *Imperfect DLL Code Tracking*

3.2.1 Coexistence Scenario. The following signal conditions were used for coexistent SMSE-DSSS scenario under *imperfect* DLL code tracking conditions.

The SMSE coexistent signal remained unchanged from that used in the perfect DLL code tracking demonstration in Section 3.1.1. Once gain, the DSSS system used BPSK for both data and spreading modulations. However, the spreading code used for demonstration was an $N_c = 1023$-chip Gold code sequence with exactly one code period (1023 chip intervals) occurring per data symbol. The symbol rate was set to $R_{sym} = 1\ kHz$, and therefore the chip rate was $R_{chip} = 1.023\ MHz$. The DSSS receiver was *perfectly synchronized* to the transmitted DSSS signal in terms of carrier tracking (frequency and phase) and communication symbol tracking.

The DSSS receiver used the non-coherent DLL described in Section 2.1.3 for code tracking. A non-coherent DLL was chosen for demonstration give it is suitable for general purpose DSSS receivers without requiring precise carrier tracking [25]. The DLL RF filter was an 8th-order Butterworth filter having a bandwidth equal to twice the chip rate, $W_{RF} = 2R_{chip} = 2.046\ MHz$. The bandpass filter following the despreading mixer in the early/late DLL branches was an 8th-order Butterworth filter having a bandwidth of $W_{BPF} = 5\ kHz$. The low-pass filter in the DLL energy detectors used a 4th-order Butterworth filter with a bandwidth of $W_{LPF} = 2.5\ MHz$. The loop filter was a first-order filter with $F_{loop}(s) = 1$. The mapping between the discriminator output and the NCO was linear, such that a maximum response from the discriminator resulted in a NCO code generation rate of $2R_{chip}$. Finally, the early (advanced) and late (delayed) codes were separated by $\Delta_{e-l} = 1$ chip.

3.2.2 Hybrid Optimization. The perfect DLL code tracking demonstration in Section 3.1 only accounted for a portion of the errors associated with coexistent interference. Under more realistic conditions, imperfect DLL code tracking due to interfering signals will further degrade DSSS P_b performance [15, 16]. The imperfect code tracking demonstration is designed to help isolate code tracking error effects resulting from a coexistent SMSE waveform being received by the DSSS system.

Unlike the perfect DLL code tracking demonstration in Section 3.1, the GA and RSM optimization processes are not performed independently under imperfect tracking conditions. Rather, they are used to complement each other using a hybrid optimization approach. In this process, the GA process is first used to determine an initial "coarse" solution that is then passed to the RSM process which determines the final optimized solution. This approach is advantageous for several reasons. First, given that GA is naturally a discrete process, it performs best at describing and optimizing SMSE design parameters that are discrete as well, e.g., the number of IFFT points N_f. Second, GA solutions are less accurate when the gene mapping is for continuous variables, such as subcarrier spacing Δf. In this respect, the RSM process is most advantageous given its solutions are based on a modeled surface response and the resultant optimized solution is not necessarily part of the input test matrix. However, one limitation of the RSM process is that it requires a good starting point or the search process may become too prolonged. Therefore, the final hybrid approach for optimizing $(N_f, \Delta f)$ selection exploits the strength of each process and consists of 1) using GA first to determine the most appropriate N_f value, followed by 2) the RSM process to find the optimized Δf value associated with the GA N_f value.

3.2.3 Optimization Metric. To isolate the impact of a coexistent SMSE signal on DSSS code tracking performance, an optimization metric is introduced. In this case, optimization is based on DLL code tracking performance versus end-to-end DSSS P_b. The DSSS Correlation Degradation metric is defined here as

$$C_{Deg} = 1 - \int_0^{N_c T_c} c_i(t) c_p(t) dt , \qquad (3.2)$$

where $c_p(t)$ is the DLL prompt code estimate shown in Figure 2.10, $c_i(t)$ is incident received code and integration is carried out over one full code period $N_c T_c$ ($N_c T_c = T_{sym} = 1\ msec$ for this demonstration). The minimum value of $C_{Deg} = 0$ indicates *perfect* code tracking, a value of $0 < C_{Deg} < 1$ indicates *imperfect* code tracking, and a maximum value of $C_{Deg} = 1$ indicates a DLL break-lock condition. Representative histograms for C_{Deg} in (3.2) under imperfect *manageable* and imperfect *severely degraded* DLL tracking conditions are shown in Fig. 3.1 and Fig. 3.2, respectively.

3.2.4 Genetic Algorithm (GA). Each combination of the two optimization input parameters $(N_f, \Delta f)$ were represented using nine binary digits in a gene. In addition, the parameters were constrained to $N_f \in \{16, 32, 64, 128\}$ and $\Delta f \in [6, 133] \cap \mathcal{I}$. The feasibility region was defined such that the total SMSE signal bandwidth was less than the DLL RF filter bandwidth ($N_f \Delta f < 2.046\ MHz$). The optimization objective function was the the correlation metric C_{Deg} given in (3.2).

Consistent with the process described in Section 2.2.1, the GA process proceeded as follows:
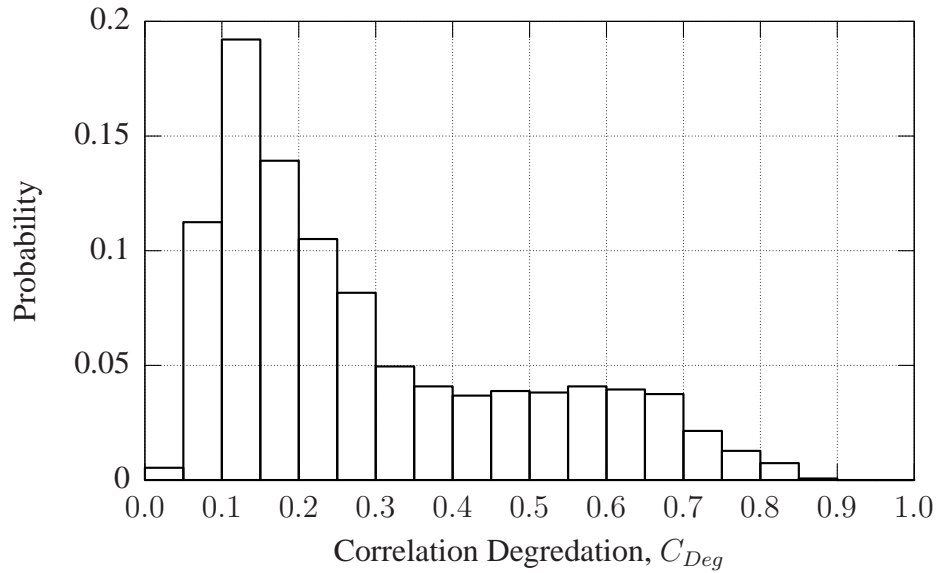
Figure 3.1: Representative histogram of correlation degradation metric C_{Deg} in (3.2) for imperfect *manageable* DLL tracking conditions.
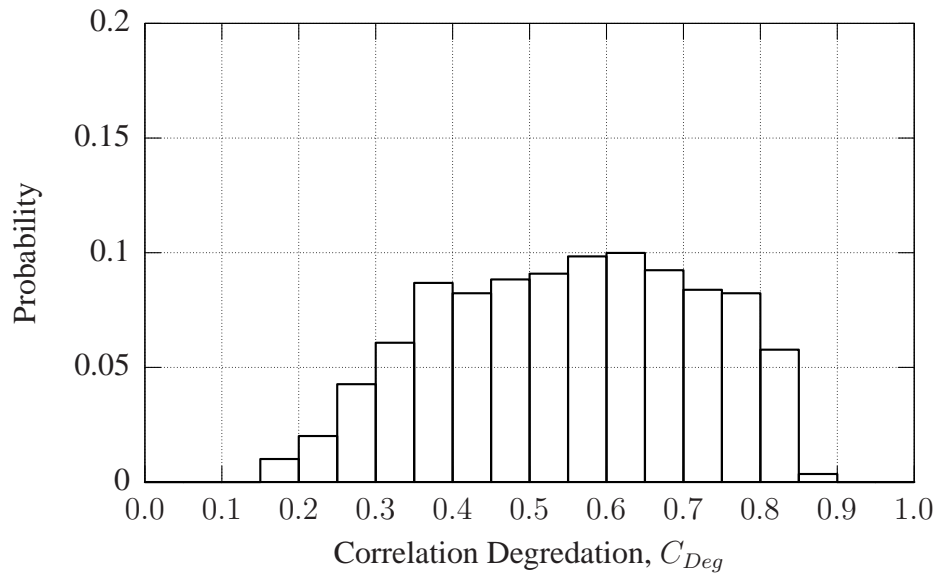
Figure 3.2: Representative histogram of correlation degradation metric C_{Deg} in (3.2) for imperfect *severely degraded* tracking conditions.

1. The initial population consisted of ten randomly generated genes.

2. Using a physics-based analytic model, C_{Deg} was calculated for each case to judge the genes' *fitness*.

3. For mating, four of the five most fit genes and two of the least fit genes were chosen as parents. The resulting six parents were randomly assigned for mating, creating three pairs of parents.

4. To mate, the binary digits of the parents' genes were compared. When the binary values of parent genes matched, the same values were passed to the offspring. Where the values differed, the offspring values were randomly chosen with equal probability. Each pair of parents created two offspring.

5. After mating, each binary digit in the offspring's gene was subjected to a 10% chance of mutation, or complementing the bit value.

6. The next population of ten genes included the six children and the four most fit parents. The GA process then proceeded as it did with the initial/previous population of ten genes. For design demonstrations in this dissertation, the GA process was repeated for 1000 generations.

The GA process was used to both minimize and maximize the DSSS C_{Deg}. Minimizing C_{Deg} corresponds to "peaceful" coexistence while maximizing C_{Deg} creates a worst-case coexistence scenario. The GA output included ten SMSE $(N_f, \Delta f)$ parameter combinations for both the minimization and maximization cases. These final populations were

compared using multi-comparison tests from Section 2.2.2.8. Optimization results for this process are found in Section 3.2.4.

3.2.5 *Response Surface Methodology (RSM).* Given that the GA process provided a "coarse" solution that maximized and minimized C_{Deg}, the RSM was next used to obtain final optimization results. Since N_f is a discrete parameter and every value is tested in the GA process, the GA-optimized values for N_f were accepted. The RSM process was then applied to find corresponding optimized values for Δf.

As in the perfect code tracking case, the RSM experiments used the ANOVA with a second-order model to detect curvature in the response surface. However, there was only one input variable in the model, x_1. Consequently, the experimental design consisted of a one-factor, five-level, full-factorial design. The matrix form of the system model from (2.8) is then expressed as

$$\mathbf{Y} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_{11} \end{bmatrix}. \tag{3.3}$$

The SMSE parameters were optimized using the steepest ascent/descent process described in (2.21) until the response surface fit the second-order model, as determined by the ANOVA. Once the second-order model detected curvature, (2.19) was used to solve for the SMSE parameters in terms of their coded variables. Final results for the RSM optimization process under *imperfect* code tracking conditions are provided in Section 4.2.3.

55

IV. Results and Analysis

The SMSE waveform design procedure is demonstrated in this chapter under two code tracking conditions, including perfect code tracking and imperfect code tracking. In both cases, the SMSE waveform is introduced into the DSSS system and resultant DSSS bit error rate (P_b) performance characterized as the SMSE parameters are varied. Under *perfect code tracking* conditions in Section 4.1, the DSSS receiver is assumed to maintain perfect code tracking such that the DLL produces an ideal prompt code $c_p(t)$ for despreading. Optimal SMSE parameter selection is addressed using independent Genetic Algorithm (GA) and Response Surface Methodology (RSM) approaches. Under *imperfect code tracking* conditions in Section 4.2, DLL code tracking is incorporated to assess performance under more realistic channel conditions using a less than ideal $c_p(t)$ estimate. Optimal SMSE parameter selection under these conditions is addressed using a hybrid GA-RSM technique. The reader should exercise caution when comparing optimization results across various sections in this chapter. This is especially important when considering perfect and imperfect DSSS code tracking results, given they were generated using dissimilar SMSE and DSSS parameter values as well as different objective functions and feasibility region constraints during optimization.

4.1 Perfect Code Tracking

4.1.1 Demonstration Procedure. The SMSE waveform design procedure is first demonstrated in a coexistent environment under *perfect* DSSS code tracking conditions. In this case, the estimated DLL prompt code $c_p(t)$ is considered to be ideal such that the DSSS receiver despreading code perfectly matches the transmitted spreading code. Optimal SMSE parameter selection is addressed using independent GA and RSM approaches. The goal is to find SMSE parameter values that optimize DSSS receiver performance in terms of end-to-end bit error rate (P_b).

4.1.1.1 Coexistent SMSE Signal. The coexisting SMSE signal was generated using the framework described in Section 2.1.1 all but two of the parameters fixed to implement conventional OFDM [34]. Performance of an SMSE OFDM implementa-

tion is dictated by two design parameters, including the total number of IFFT points N_f, which dictates the number of individual SMSE subcarriers, and the subcarrier frequency spacing Δf. Together, the value of these two parameters determines the overall waveform bandwidth.

4.1.1.2 Overall DSSS System. The DSSS system considered here uses BPSK for both data and spreading modulations. The spreading code was a $N_c = 32$ bit Hadamard sequence with exactly one code period occurring per data symbol. The DSSS P_b performance is evaluated over an Additive White Gaussian Noise (AWGN) channel. The DSSS receiver employs an ideal RF filter prior to the despreading mixer. Following the despreading mixer, the communication symbols are estimated using a single channel correlation process under maximum likelihood conditions.

4.1.2 Genetic Algorithm (GA). The analytic SMSE framework enables parametric optimization of OFDM waveforms using a GA process. As noted previously, the two SMSE optimization parameters included, 1) the number of IFFT points in the SMSE waveform generation (N_f), and 2) the subcarrier frequency separation (Δf). Each GA gene used eight binary digits to represent the possible SMSE parameter values. Furthermore, the value of N_f was constrained to be an integer power of two in the range [1,128]. For representation in the GA gene, Δf was assigned an integer value in the range [1,33]. The objective function to be optimized was DSSS bit error rate P_b. For demonstration purposes, the GA optimization process was carried out for two cases: (A) *Minimizing* DSSS P_b which represents best-case SMSE-DSSS coexistence performance, and (B) *Maximizing* DSSS P_b which represents worst-case SMSE-DSSS coexistence performance.

For both cases, the initial GA population consisted of ten randomly generated genes. Using Monte Carlo simulation of a physics-based analytic model, the DSSS P_b was calculated for each case to judge the genes' *fitness*, with the DSSS system P_b value being the fitness statistic. Tabulated GA results for the two optimization cases are presented in Table 4.1 and Table 4.2 [3]. Each table includes the initial random population values and the final optimized values. Table 4.1 shows the GA optimized parameter values that *min-*

imize DSSS P_b (best-case SMSE-DSSS coexistence) and Table 4.2 shows GA optimized parameter values that *maximize* DSSS P_b (worst-case SMSE-DSSS coexistence).

Corresponding bit error curves for the data presented in Table 4.1 and Table 4.2 are shown in Fig. 4.1 and Fig. 4.2, respectively [3]. In both cases, the signal power (S) and the noise power (N) in the DSSS system remained fixed while the interfering SMSE power (I) was varied to achieve the indicated Signal-to-Interference-plus-Noise Ratio $SINR = S/(I + N)$.

GA optimized results for Minimum DSSS P_b are shown in Fig. 4.1. The upper curve (unfilled boxes) is provided for comparison and shows the highest resultant P_b performance obtained from the initial non-optimized parameter population. The lower curve (filled boxes) is the resultant P_b performance using the final GA-optimized parameters from Table 4.1 ($N_f = 64, \Delta f = 11\ Hz, W_{SMSE} = 1.41\ KHz$) and represents best-case coexistence. In this case, the W_{SMSE} bandwidth greatly exceeds $W_{RF} = 2 \times R_{chip} = 64\ Hz$ and minimal received SMSE power actually enters the DSSS detector. Consequently, the P_b results approach the noise limited case, i.e., the resultant P_b is the same as if there were no SMSE signal present. This result is a direct consequence of the feasibility region being loosely constrained and allowing solutions where $W_{SMSE} > W_{RF}$. As such, there are additional GA solutions in Table 4.1 that were analyzed and produced similar noise limited results, e.g., the ($N_f = 128, \Delta f = 11\ Hz, W_{SMSE} = 2.82\ KHz$) solution uses the same subcarrier spacing with more carriers and an even wider bandwidth.

GA optimized results for Maximum DSSS P_b degradation are shown in Fig. 4.2. The lower curve (unfilled boxes) is provided for comparison and shows the lowest resultant P_b performance obtained from the initial non-optimized parameter population (noise limited performance). The upper curve (filled boxes) is the resultant P_b performance using the final GA-optimized parameters from Table 4.2 ($N_f = 1, f = 16\ Hz, W_{SMSE} = 32\ Hz$). This is the special case of $N_f = 1$ (single BPSK modulated subcarrier centered at f_c) and produces worst-case coexistence. Given the resultant bandwidth of $W_{SMSE} = 32\ Hz$, all SMSE power is within $W_{RF} = 64\ Hz$ and contributes to degraded performance. In addition to

Table 4.1: Initial and optimized SMSE parametric value populations using the GA process to *MINIMIZE* DSSS probability of bit error (P_b) – Best-Case SMSE-DSSS Coexistence Performance [3].

N_f		$\Delta f\ (kHz)$	
Initial	Optimized	Initial	Optimized
128	64	20	10
2	128	28	11
16	128	28	4
8	64	29	22
128	8	5	1
64	64	4	8
8	128	21	11
1	128	5	11
64	64	19	23
8	64	18	11

Table 4.2: Initial and optimized SMSE parametric value populations using the GA process to *MAXIMIZE* DSSS probability of bit error (P_b) – Worst-Case SMSE-DSSS Coexistence Performance [3].

N_f		$\Delta f\ (kHz)$	
Initial	Optimized	Initial	Optimized
128	1	20	16
2	4	28	16
168	32	28	13
8	1	29	16
128	1	5	32
64	1	4	16
8	1	21	16
1	1	5	16
64	1	19	16
8	1	18	16

Figure 4.1: SINR vs DSSS P_b performance for initial (unfilled boxes) and GA optimized (filled boxes) SMSE parameters in Table 4.1. GA optimized for *Minimum P_b* – Best-Case SMSE-DSSS Coexistence Performance [3].
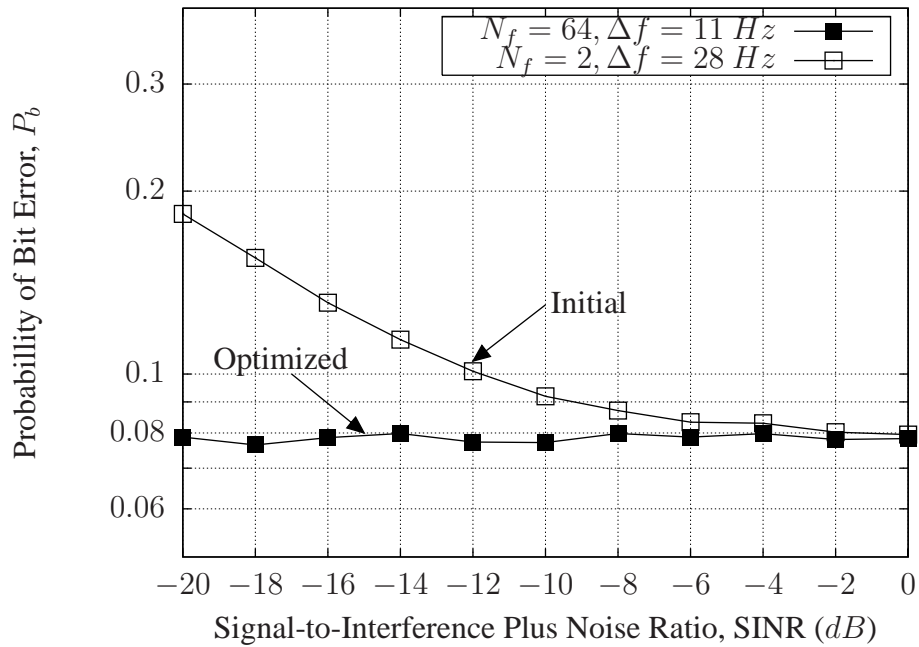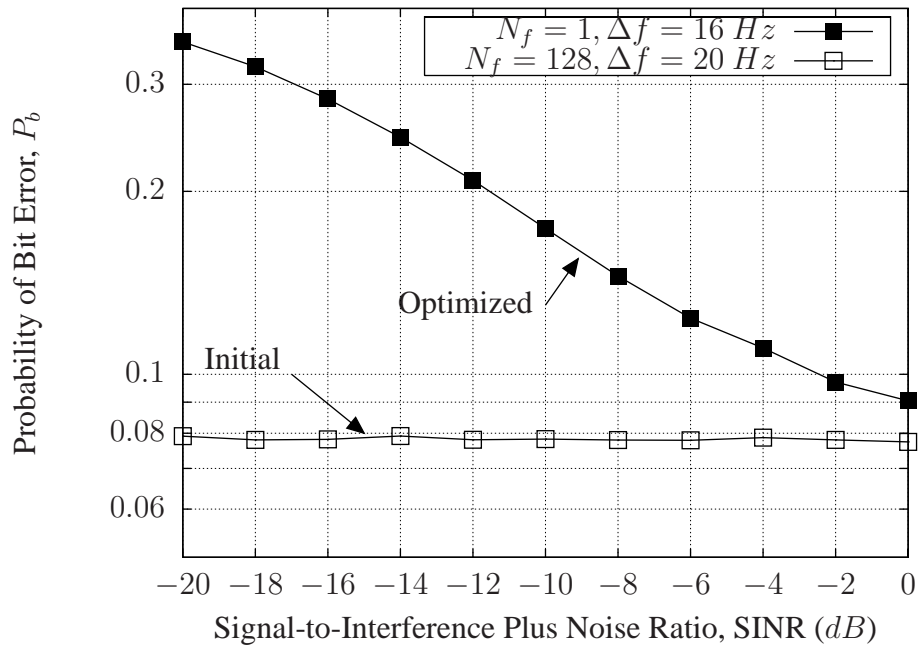

Figure 4.2: SINR vs DSSS P_b performance for initial (unfilled boxes) and GA optimized (filled boxes) SMSE parameters in Table 4.2. GA optimized for *Maximum P_b* – Worst-Case SMSE-DSSS Coexistence Performance [3].

power, the P_b degradation is also attributable to the spectral sinc $(f) = \sin(f)/f$ response of the SMSE waveform and the spectral structure of the DSSS spreading signal. More specifically, the discrete spectral lines in the $N_c = 32$-chip DSSS spreading code are spaced $f_{chip} = R_{chip}/N_c = 1\ Hz$ apart [25] and there are $W_{RF}/f_{chip} = 64$ total lines in $W_{RF} = 2 \times R_{chip} = 64\ Hz$, or 32 total lines in $W_{SMSE} = W_{RF}/2 = 32\ Hz$. The impact of this is illustrated by considering the power spectrum at the despreading mixer output (DSSS detector input) in the DSSS receiver, which is the convolution of the spreading code spectral lines and the SMSE subcarriers. For the shift-multiply-integrate operations of the spectral convolution process that occur near f_c, there are 32 contributing products in the correlation result. As a result, the amount of power in the "despread" spectral response that falls within the DSSS detector bandwidth ($W_{Det} = 2 \times R_{sym} = 2\ Hz$) is maximum and the interfering SMSE signal has maximum impact on DSSS P_b performance.

4.1.3   Response Surface Methodology (RSM).   The RSM was the second approach considered to optimize SMSE parameter selection under perfect DSSS code tracking conditions. In this case, the experiments assumed the P_b response surface fit a second-order model given by

$$\mathbf{Y} = \beta_0 + \beta_1 \mathbf{x_1} + \beta_2 \mathbf{x_2} + \beta_{12} \mathbf{x_1} \mathbf{x_2} + \beta_{11} \mathbf{x_{11}} + \beta_{22} \mathbf{x_{22}} \tag{4.1}$$

To properly characterize this model, the experimental design considered a two-factor, three-level, full-factorial design with four additional center runs. The matrix form of the experimental design is given by

$$\mathbf{Y} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \\ \beta_{11} \\ \beta_{22} \end{bmatrix}. \tag{4.2}$$

The mapping from SMSE parameter variables $(N_f, \Delta f)$ to coded variables (x_1, x_2) for the initial and final RSM experiments is shown in Table 4.3 and Table 4.4, respectively [4]. The final results were obtained using the method of steepest descent.

Table 4.3:   Mapping from SMSE parameters $(N_f, \ \Delta f)$ to coded variables $(x_1, \ x_2)$ for the *initial* RSM experiment [4].

Factor Level	N_f	x_1	Δf	x_2
Low	2^3	-1	17	-1
Medium	2^4	0	16	0
High	2^5	1	15	1

Table 4.4:   Mapping from SMSE parameters $(N_f, \ \Delta f)$ to coded variables $(x_1, \ x_2)$ for the *final* RSM experiment after applying the method of steepest descent [4].

Factor Level	N_f	x_1	Δf	x_2
Low	2^1	-1	16.37	-1
Medium	2^2	0	15.87	0
High	2^3	1	15.36	1

Table 4.5:   ANOVA table for initial and final RSM trials [4]

Source of Variation	Degrees of Freedom	Initial p-value	Final p-value
Model	5	3.8×10^{-6}	2.1×10^{-4}
x_1	1	1.6×10^{-8}	1.0×10^{-5}
x_2	1	1.0×10^{-1}	1.4×10^{-1}
$x_1 x_2$	1	3.1×10^{-1}	2.5×10^{-2}
x_1^2	1	6.6×10^{-3}	1.5×10^{-5}
x_2^2	1	3.8×10^{-1}	1.2×10^{-2}
Error	7		

Table 4.5 shows ANOVA results for the initial and final RSM experiments [4], with the table rows representing main effects and interactions according to the experimental model. The p-value indicates the significance for each factor. The smaller the p-value, in a given row, the more likely that term is significant. For the trials illustrated here, a p-value < 0.05 was considered significant, and results in rejection of the ANOVA null hypothesis of equal means.

As one may observe, results for first RSM trial indicate that the response surface was not quadratic given the p-value > 0.05. After moving to the final region, both of the x_1^2 and x_2^2 quadratic terms are significant and the stationary point x_s, found using (2.19), is determined to be $N_f = 2$ and $\Delta f = 15.87\ Hz$. However, additional Eigenvalue analysis of the resulting regression coefficients revealed that the stationary point x_s is neither a minimum nor a maximum, but rather a saddle point. If a global minimum or maximum is desired, the RSM process would need to be repeated using a different starting point. For purposes of this research, the saddle point solution is sufficient to demonstrate the practical utility of the RSM process.

To illustrate consistency between the RSM saddle point solution and physical wave-form level modeling, an end-to-end simulation was run for the SMSE-DSSS coexistent scenario. Simulation results are shown in Fig. 4.3 for the P_b corresponding to the RSM
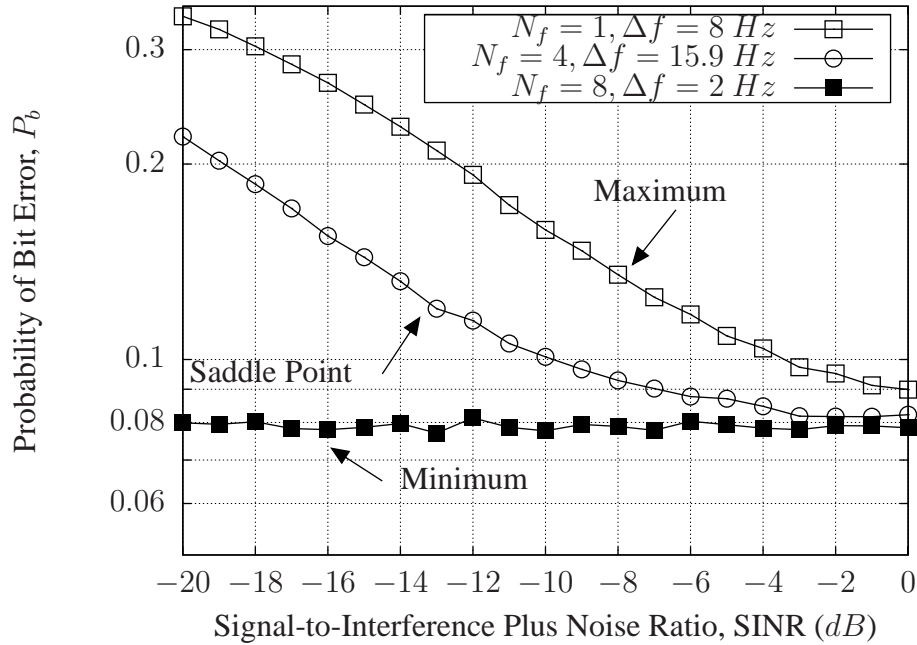
Figure 4.3: SINR vs DSSS P_b performance for RSM optimized SMSE parameters. Unfilled boxes represent Best-Case SMSE-DSSS Coexistence and filled boxes represent Worst-Case SMSE-DSSS Coexistence [4].

stationary point (unfilled circles) [4]. Additional best-case (unfilled boxes) and worst-case (filled boxes) results are also shown and were obtained from additional RSM searches.

As indicated in Fig. 4.3, the saddle point solution in Table 4.4 of ($N_f = 4, \Delta f = 15.87\ Hz, W_{SMSE} = 127\ Hz$) has approximately one-half of its power within $W_{RF} = 2 \times R_{chip} = 64\ Hz$ and is indeed non-optimal (neither maximum nor minimum degradation) given that its resultant P_b performance consistently fall between the two extremes of maximum degradation for ($N_f = 1, \Delta f = 8\ Hz, W_{SMSE} = 16\ Hz$) and minimum noise limited performance for ($N_f = 8, \Delta f = 2\ Hz, W_{SMSE} = 32\ Hz$). As with GA maximization results in Section 4.1.2, worst-case coexistence performance is once again achieved for the special case of $N_f = 1$ (single BPSK modulated subcarrier centered at f_c) and the physical interpretation as to why this occurs is as explained in that section.

Results in Section 4.1.2 and Section 4.1.3 suggest that the GA and RSM techniques are applicable for rigorous coexistence analysis of conventional, DSSS and OFDM-based SMSE waveforms. In applying both techniques, the independently optimized results con-
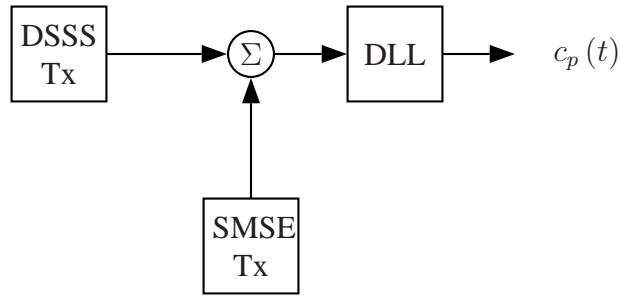
Figure 4.4:   Imperfect code tracking scenario for SMSE-DSSS coexistent optimization. SMSE parameters impact DLL performance and impact prompt code $c_p(t)$ estimation.

sistently migrated toward expected system performance and there were valuable "lessons learned" regarding how to best address more realistic constraints and conditions. As presented in the next section, these lessons learned drove the development a hybrid GA-RSM optimization process to address imperfect code tracking conditions.

4.2   Imperfect Code Tracking

The SMSE waveform design procedure is next demonstrated in a coexistent environment under *imperfect* code tracking conditions. As discussed in Section 2.1.3, receiver code tracking is critical to successful demodulation. The effects of code tracking are incorporated here to assess SMSE-DSSS coexistence performance under more realistic channel conditions. In this case, a less than ideal prompt code estimate $c_p(t)$ is used and optimal SMSE parameter selection is addressed using a hybrid GA-RSM technique. The hybrid technique first uses the GA process in Section 4.2.2 to find a "coarse" optimization solution. The course GA solution is then used as the initial starting point in the RSM process of Section 4.2.3 which provides the final, more precise optimized solution.

4.2.1   Demonstration Procedure.   The imperfect code tracking scenario for SMSE-DSSS optimization is depicted in Fig. 4.4 which shows the coexistent SMSE and DSSS transmitters. As shown, both signals are present while the DLL in the DSSS receiver tracks the received spreading code.
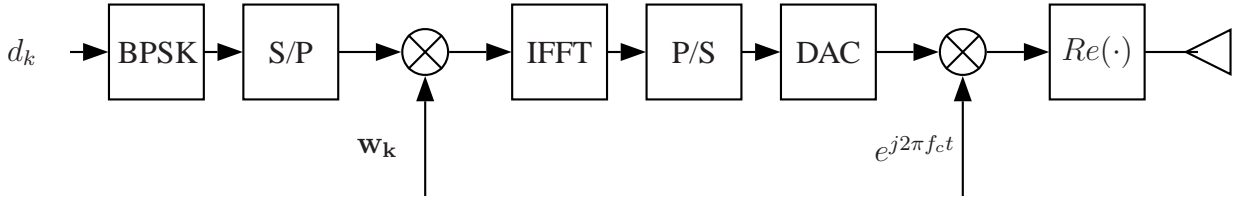
65

Figure 4.5:   Coexistent SMSE signal generation architecture.

The DLL prompt code output $c_p(t)$ in Fig. 4.4 represents the PN spreading code estimate that is used to despread the received signal prior to demodulation. In this scenario, the received DSSS waveform is corrupted by the SMSE waveform which causes the prompt code estimate to be an imperfect representation of the true spreading code. As a result, the incoming DSSS signal of interest is not perfectly despread and demodulated P_b performance is degraded. Demonstration details with respect to the coexistent SMSE signal, the overall DSSS system, and DLL code tracking are presented in Section 4.2.1.1, Section 4.2.1.2 and Section 4.2.1.3, respectively.

4.2.1.1   Coexistent SMSE Signal.   The SMSE signal was generated using the architecture shown in Fig. 4.5. The sequence of input data bits d_k are randomly generated with equal probability of being either a 0 or 1. The bits are then modulated using antipodal BPSK such that the kth bit b_k is mapped to $d_k \in \{-1, 1\}$ according to

$$d_k = (-1)^{b_k} .$$

Using a serial-to-parallel (S/P) conversion process, groups of N_f modulated bits passed to the IFFT operation after 1) element-by-element weighting by complex vector w_k and 2) zero padding. The first weighting coefficient is set to $w_1 = 0$ and the remaining $N_f - 1$ coefficients are set to unity such that $w_k = [0\ 1\ 1\ \cdots\ 1]$. This is consistent with common practice when implementing OFDM and effectively ensures that the resul-

66

tant SMSE waveform does not include a DC component. In the simulation, the resultant weighted vector of N_f elements is then zero padded with a total of $f_s/\Delta f - N_f$ zeros. This ensures that the resultant time domain SMSE symbol following the IFFT process is sampled at the same effective rate f_s as other signals in the simulation. In the simulation, this operation effectively replaces the Digital-to-Analog converter (DAC) in a real-world communication system.

Each IFFT operation creates one time domain SMSE symbol comprised of N_f carriers and having a symbol duration of $T_{sym} = 1/\Delta f$ and sample frequency f_s. The resultant SMSE symbols are complex baseband signals. The received interfering SMSE waveform was taken as real part of the carrier modulated signal, expressed as

$$s_{SMSE}(t) = Re\left[v(t)e^{j2\pi f_c t}\right] \ ,$$

where f_c is the carrier frequency, $v(t)$ is the complex baseband SMSE signal, and $s_{SMSE}(t)$ is the resultant coexisting SMSE signal.

The optimized SMSE variables used for the imperfect code tracking demonstration included: 1) the number of IFFT points N_f and 2) the subcarrier frequency separation Δf, which were allowed to take on values of $N_f \in \{16, 32, 64, 128\}$ and $\Delta f \in [6, 132]$ kHz.

4.2.1.2 Overall DSSS System.   The transmitted DSSS signal was generated using the architecture shown in Fig. 4.6. The input data modulated waveform $d(t)$ is based on BPSK modulation using randomly generated bits having equal probability of being a 0 or 1. The baseband data modulated waveform $d(t)$ is then carrier modulated to f_c and spread by $c(t)$ prior to transmission. Spreading waveform $c(t)$ is generated according to

$$c(t) = \sum_{m=1}^{N_c}(-1)^{c_m}p(t - mT_{chip}) \ ,$$

where c_m is an N_c length binary spreading code and T_{chip} is the chip duration. For the imperfect code tracking demonstration being considered here: 1) c_m was an $N_c = 1023$
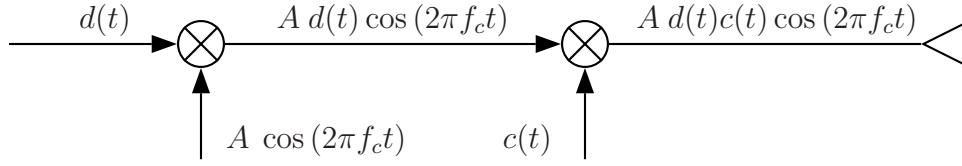
67

Figure 4.6:   DSSS signal generation architecture.

length Gold code, 2) the code period was set equal to the symbol duration ($N_c \times T_{chip} = T_{sym}$), 3) the chip rate was $R_{chip} = 1/T_{chip} = 1023 \times 10^3$ chips/sec, and 4) the code period start/stop boundaries were synchronous with T_{sym} transition boundaries.

   4.2.1.3 Delay-Lock Loop (DLL) Code Tracking. The DLL architecture used for the imperfect code tracking demonstration is shown in Fig. 4.7. The RF bandpass filter was implemented as an 8th-order Butterworth filter having a -3 dB bandwidth of $W_{RF} = 2 \times R_{chip} = 2.046$ MHz.

  The prompt code $c_p(t)$ is generated using the same spreading code sequence used by the DSSS transmitter but with its timing characteristics varied by the NCO in accordance with the DLL error voltage. The early code $c_e(t)$ is advanced $T_{chip}/2$ relative to the prompt code while the late code $c_l(t)$ is delayed $T_{chip}/2$ relative to the prompt code.

  The IF bandpass filters serves to remove all but the fundamental harmonic after multiplication by either $c_e(t)$ or $c_l(t)$. The IF filters were implemented as 8th-order Butterworth filters having a -3 dB bandwidth of $W_{BPF} = 5$ kHz. This bandwidth is somewhat wider than necessary to track the code of interest, but a wider bandpass was chosen for consistency with real-world conditions where Doppler shift is not precisely known or not tracked.

  The squaring operation following the IF bandpass filtering and subsequent low pass filtering comprise an envelope detector. The low pass filters are designed to remove double frequency terms resulting from the squaring operation and were implemented here 4th-order Chebychev filters having a -3 dB bandwidth of $W_{LPF} = 2.5$ MHz. The early and late gate low pass filter outputs are summed and filtered to provide the control signal (discrim-
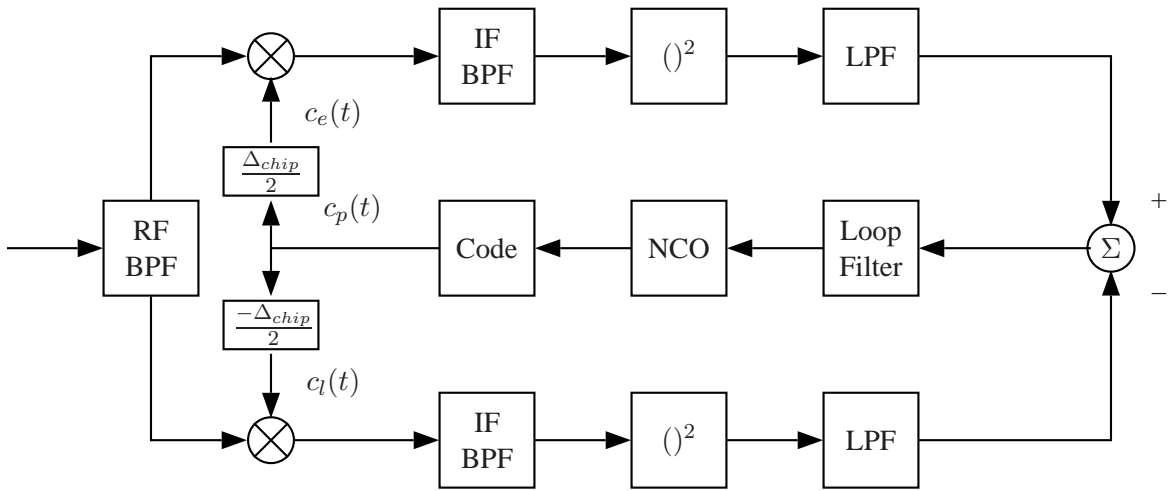
Figure 4.7:   Delay-Lock Loop (DLL) architecture used for SMSE-DSSS coexistence demonstration under imperfect code tracking conditions.

inator voltage) to the Numerically Controlled Oscillator (NCO). For simulation purposes, the NCO maintains its own time reference to generate the PN code. When the discriminator voltage is zero, the NCO sample rate matches the simulated sample rate of f_s and the estimated prompt code $c_p(t)$ perfectly matches the received code. When the signed discriminator value deviates positive/negative from zero, the NCO sample rate increases/decreases accordingly and the estimated prompt code $c_p(t)$ is altered. Ideally, the discriminator value varies linearly between its extremes. When viewed in real-time, an accordion-like effect is observed in $c_p(t)$ on a chip-by-chip basis, i.e., there is a clear compression and dilation of the chip intervals throughout the code period.

Perfect DLL tracking results in a prompt code estimate that is an exact replica (code phase and chip duration) of the received DSSS spreading waveform. The presence of a coexisting SMSE signal causes the estimated prompt code to vary in both phase and chip duration as the DLL compensates. This degradation is characterized and quantified using the cross-correlation metric C_{Deg} introduced in Section 3.2.3.

The correlation metric introduced in Section 3.2.3 is generated as follows to characterize DLL code tracking performance. The DSSS waveform is received by the DLL with an initial code phase difference of 20% between the prompt code and the true spreading code. The DLL processes this signal for two code cycles without adaptation to allow the DLL to stabilize to an accurate initial phase estimate. Adaption begins after two code periods. After five code periods the coexistent SMSE signal is applied to the loop along with the DSSS signal for five additional code periods. The prompt code and true spreading code from the five additional code periods are correlated per (3.2) using an integration interval of one code period. The mean value from the correlation process comprises one sample of the correlation metric, C_{T-P}. For clarity, the data is presented is terms of a normalized and shifted version of C_{T-P}, termed C_{Deg}, which represents the amount of tracking degradation caused by the coexistent SMSE signal.

Representative histograms of the C_{Deg} metric are shown in Fig. 3.1 and Fig. 3.2 for moderately and severely degraded DLL code tracking performance, respectively. Recall

that values of C_{Deg} approaching zero indicate better DLL tracking performance while values near one indicate poorer, more degraded tracking performance.

In addition, observations of C_{Deg} behavior in many such figures indicates that the SMSE waveform impact on DLL performance is highly dependent on ISR. Thus, care was taken to choose appropriate ISR values for subsequent waveform design demonstrations. For example, if the chosen ISR value is too high all of the potential SMSE waveform designs can cause the DLL to break lock. Likewise, if the chosen ISR value is too low, many of the potential waveform designs will have minimal impact. Observations of C_{Deg} for 15 dB $\leq ISR \leq$ 35 dB showed that the metric behaved best for optimization purposes at $ISR = 20$ dB. At that value, all of the SMSE parameter combinations degraded DLL code tracking performance, but none consistently caused the DLL to break lock.

4.2.2 Hybrid Optimization Step 1: GA Process. The hybrid optimization technique first uses the GA process in Section 3.2.4 to find a "coarse" optimized SMSE solution. For demonstration purposes, 1000 generations were used with optimization variables being the number of IFFT points $N_f \in \{16, 32, 64, 128\}$ and the subcarrier frequency spacing $\Delta f \in [1, 33] \cap \mathcal{I}$. The feasibility region was defined such that $N_f \Delta f \leq 1/T_{chip}$.

The GA process was used to both minimize and maximize C_{Deg} under imperfect DLL tracking conditions. Table 4.6 shows the initial random and final optimized populations using the GA process to *minimize C_{Deg}* . Box and whisker plots of C_{Deg} for the initial and final optimized populations are shown in Fig. 4.8 and Fig. 4.9, respectively.

Table 4.7 shows the initial random and final optimized populations using the GA process to *maximize C_{Deg}*. Box and whisker plots of C_{Deg} for the initial and final optimized populations are shown in Fig. 4.10 and Fig. 4.11, respectively.

The parameter combinations in Table 4.6 and Table 4.7 were compared via the LSD test described in Section 2.2.2.8. The SMSE parameters, ($N_f = 16, \Delta f = 6$ kHz), provided C_{Deg} means which were statistically lower than all other parameter combinations except for ($N_f = 16, \Delta f = 7$ kHz). These two SMSE parameter combinations were statistically indistinguishable. The combination ($N_f = 16, \Delta f = 6$ kHz) was chosen as the start-

Table 4.6: Initial random and final optimized SMSE parametric value populations using the GA process to *MINIMIZE* correlation degradation C_{Deg} – Best-Case Tracking and SMSE-DSSS Coexistence.

N_f		$\Delta f\ (kHz)$		Figs. 4.8 & 4.9
Initial	Optimized	Initial	Optimized	Combination
32	16	23	10	1
64	16	8	38	2
16	32	53	6	3
16	16	17	6	4
32	16	14	6	5
16	32	24	6	6
64	16	11	6	7
16	16	7	6	8
16	16	34	6	9
16	16	63	6	10

Table 4.7: Initial random and final optimized SMSE parametric value populations using the GA process to *MAXIMIZE* correlation degradation C_{Deg} – Worst-Case Tracking and SMSE-DSSS Coexistence.

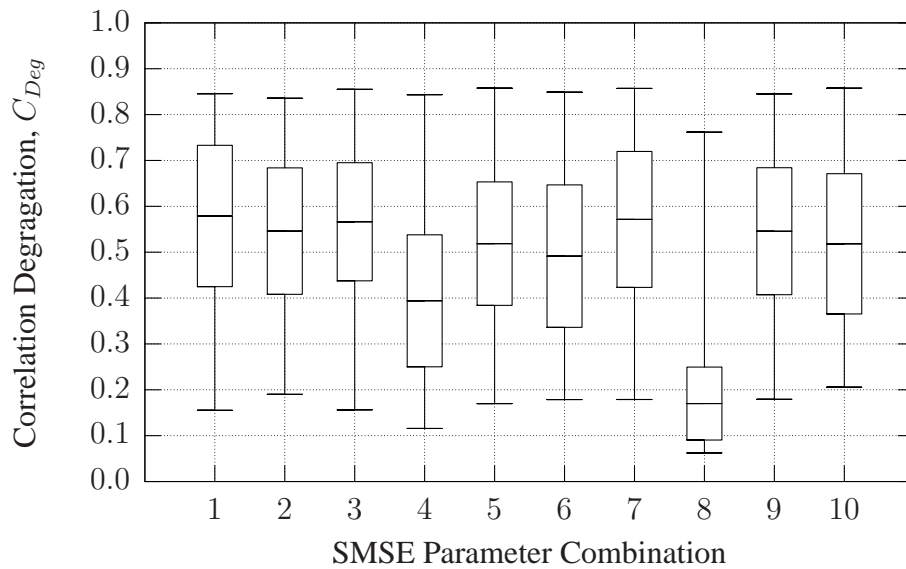| N_f		$\Delta f\ (kHz)$		Figs. 4.10 & 4.11
Initial	Optimized	Initial	Optimized	Combination
16	32	48	25	1
16	32	20	24	2
16	32	24	28	3
64	32	13	24	4
32	32	29	24	5
32	32	27	24	6
32	32	31	24	7
16	32	33	24	8
32	32	22	24	9
16	32	29	24	10

Figure 4.8: Box and whisker plot of C_{Deg} results for *initial random* population using the GA process to *MINIMIZE* C_{Deg}.
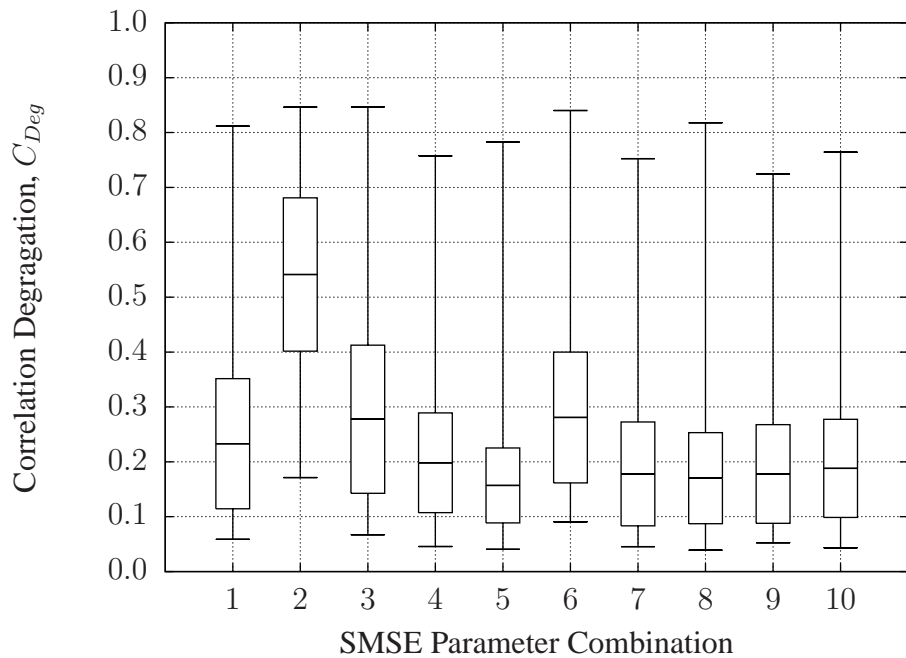

Figure 4.9: Box and whisker plot of C_{Deg} results for *final optimized* population using the GA process to *MINIMIZE* C_{Deg}.

Table 4.8:   Hybrid Optimization Process: GA Optimized SMSE parameters N_f and Δf maximizing and minimizing the correlation degradation metric, C_{Deg}.

	N_f	Δf (kHz)	W_{SMSE} (MHz)
Minimized C_{Deg}	16	6	0.19
Maximized C_{Deg}	32	24	1.54

ing point for the RSM minimization process instead of ($N_f = 16, \Delta f = 7$ kHz), because it was the result of GA convergence. However, the parameter values ($N_f = 16, \Delta f = 7$ kHz) were also explored with the RSM process, because of its proximity to ($N_f = 16, \Delta f = 6$ kHz).

The SMSE parameters ($N_f = 32, \Delta f = 24$ kHz) provided C_{Deg} means that were not statistically lower than any other parameter combinations. Eleven SMSE parameter combinations from Table 4.6 and Table 4.7 provided C_{Deg} means which were statistically indistinguishable from that of ($N_f = 32, \Delta f = 24$ kHz). The other parameter combinations were similar to ($N_f = 32, \Delta f = 24$ kHz) with respect to their relatively high bandwidths ($W_{SMSE} = N_f \times \Delta f$). The bandwidth range for these signals was $512 kHz \leq W_{SMSE} \leq 992$ kHz. For comparison purposes, the bandwidth of the minimization solution, ($N_f = 16, \Delta f = 6$ kHz) was $W_{SMSE} = 96$ kHz. The combination ($N_f = 32, \Delta f = 24$ kHz) was chosen as the starting point for the RSM maximization process instead of the other, similarly performing SMSE parameter combinations, because it was the result of GA convergence.

The resultant GA-optimized SMSE parameters are shown in Table 4.8, and their corresponding histograms for C_{Deg} are shown in Fig. 4.12 and Fig. 4.13.

4.2.3   Hybrid Optimization Step 2: RSM Process.   Given the "coarse" optimization solution from the GA process in Section 4.2.2, the RSM process in Section 3.2.5 is next applied to perform final minimization and maximization of correlation degradation C_{Deg}. This hybrid GA-RSM approach addressed two issues that emerged in Section 4.1.3.
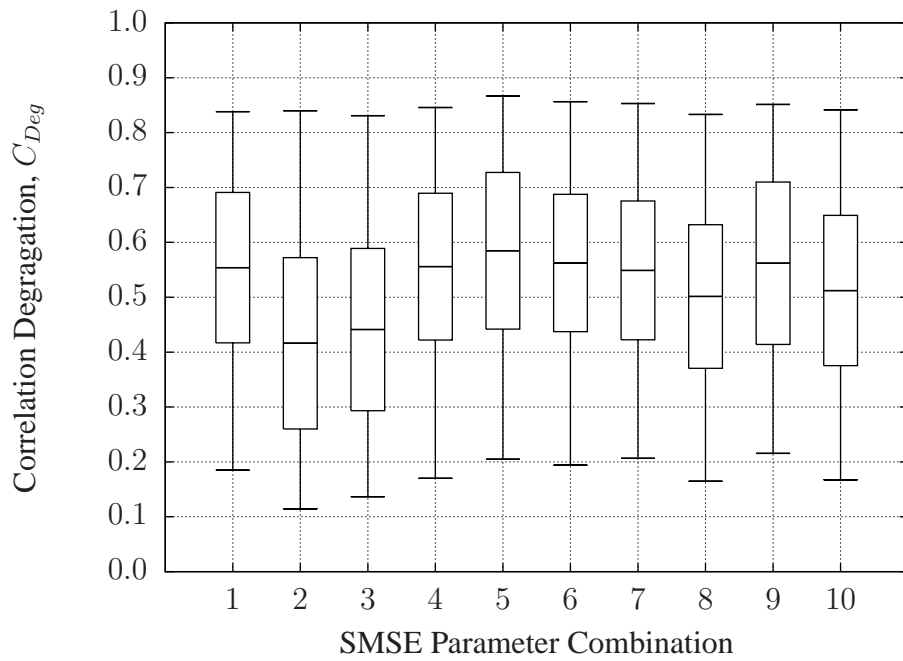
Figure 4.10:   Box and whisker plot of C_{Deg} results for *initial random* population using the GA process to *MAXIMIZE* C_{Deg}.
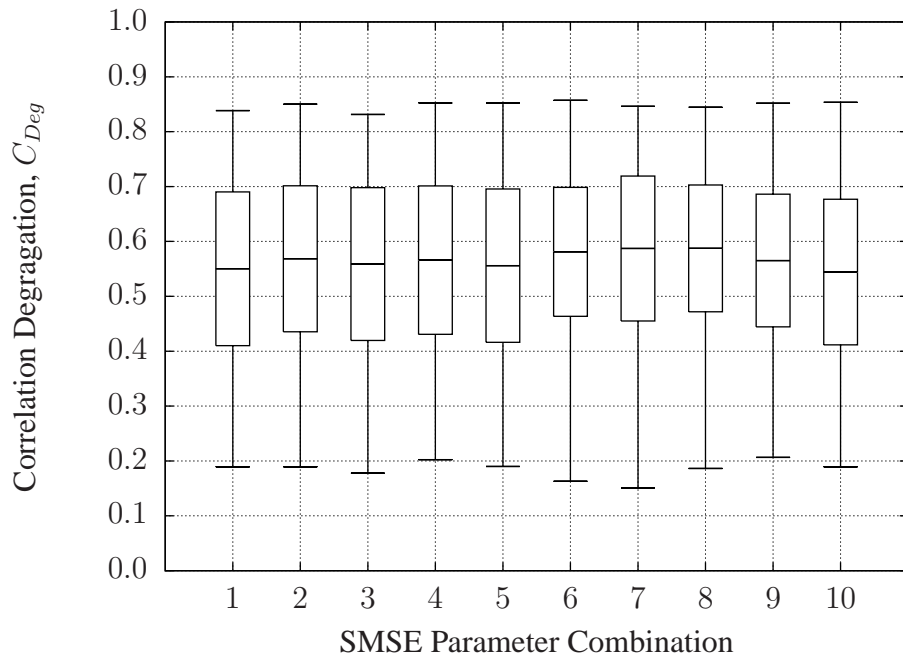
Figure 4.11:   Box and whisker plot of C_{Deg} results for *final optimized* population using the GA process to *MAXIMIZE* C_{Deg}.
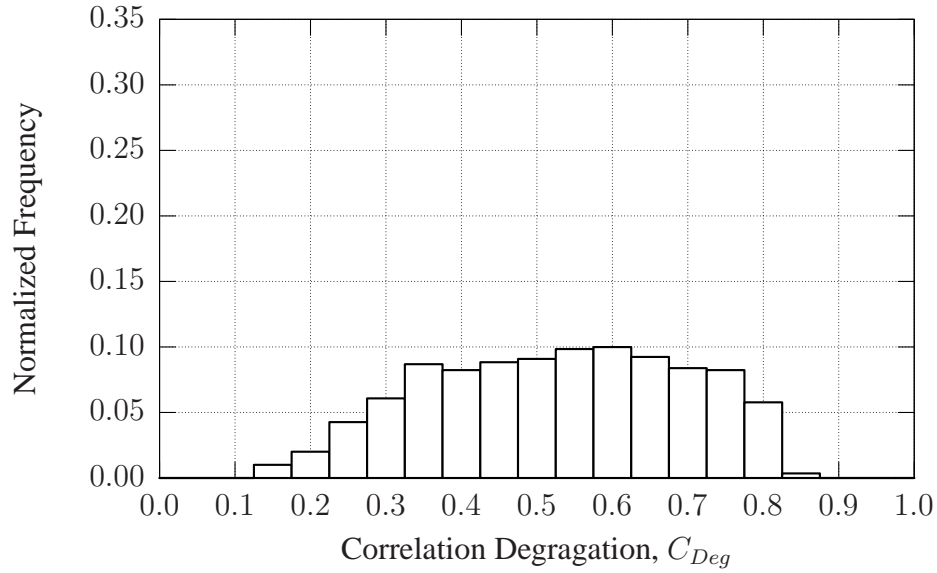
Figure 4.12: Histogram showing behavior of C_{Deg} for severely degraded DLL code tracking with ISR = 25 dB (at the output of the front-end RF filter) in coexistent SMSE-DSSS scenario with GA *MAXIMIZED* SMSE Parameters $(N_f, \Delta f)$.

Figure 4.13: Histogram showing behavior of C_{Deg} for moderately degraded DLL code tracking with ISR = 25 dB (at the output of the front-end RF filter) in coexistent SMSE-DSSS scenario with GA *MINIMIZED* SMSE Parameters $(N_f, \Delta f)$.

Table 4.9:   Hybrid Optimization Process: SMSE parameter-to-coded variable (Δf-to-x) mapping for initial RSM *MINIMIZATION* experiment under imperfect code tracking conditions.
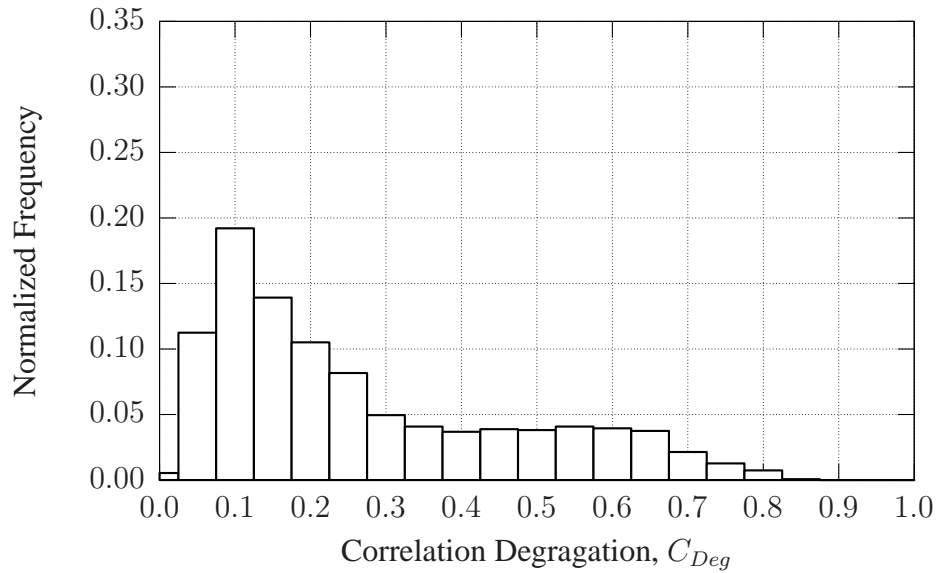
x	$\Delta f\ (kHz)$
-2	6.00
-1	6.23
0	6.47
1	6.71
2	6.94

The first issue had to do with answering the question, "What is a good starting point for the RSM process?" Given GA is the first step in the hybrid process, a systematic approach is now used to choose the RSM starting point. The second issue is simply related to the number of dimensions used in the optimization process. Given the GA solution effectively identifies the appropriate N_f value, the RSM optimization process is reduced to a one dimensional search for the appropriate Δf value. This is a desirable outcome given that RSM attempts to optimize on a continuous surface, while N_f is inherently discrete. Thus, resultant RSM design matrix has one factor with five levels. The number of SMSE subcarriers was set to $N_f = 16$ as determined by the initial GA process.

4.2.3.1   Minimizing *DLL Tracking Degradation*.   Final RSM optimization is first considered for the case where C_{Deg} is to be minimized for the coexistent SMSE-DSSS scenario. The values used for Δf are shown in Table 4.9 along with their coded counterparts in variable x.

The RSM design matrix was run with 1000 repetitions and the data transformed using the Box-Cox transformation with a value of $\lambda = 54.78$. This resulted in each condition in the trial being properly classified as normal according to the Shapiro-Wilks test described in Section 2.2.2.5.

The data was then fit to linear, pure quadratic, and quadratic models as described in Section 2.2.2.2. The resultant p-value test for all three models yielded $p < 10^{-3}$ indicating

Table 4.10:   Hybrid Optimization Process: SMSE parameter-to-coded variable (Δf-to-x) mapping for initial RSM *MAXIMIZATION* experiment under imperfect code tracking conditions.

x	Δf (kHz)
-2	23.03
-1	23.52
0	24.00
1	24.48
2	24.97

a good fit. The pure quadratic model provided the best fit based on its F statistic and p-value. However, the solutions for all three models were calculated since all were deemed to be statistically significant.

Based on these results, the final solution for a coexistent SMSE signal that is least disruptive to DSSS system performance, as indicated by minimal DSSS code tracking degradation, is $N_f = 16$ and $\Delta f = 6.47\ kHz$. This was accomplished by finding a coarse GA solution followed by fine-tuning with the RSM process.

4.2.3.2   Maximizing DLL Tracking Degradation.   Final RSM optimization is first considered for the case where C_{Deg} is to be maximized for the coexistent SMSE-DSSS scenario. The values used for Δf are shown in Table 4.10 along with their coded counterparts in variable x.

The RSM design matrix was run with 1000 repetitions and the data transformed using the Box-Cox transformation with a value of $\lambda = 10.88$. This resulted in each condition in the trial as being properly classified as normal according to the Shapiro-Wilks test described in Section 2.2.2.5.

The data was then fit to linear, pure quadratic, and quadratic models as described in Section 2.2.2.2. The resultant p-value test for the quadratic model yielded $p < 10^{-3}$ indicating a good fit.

Table 4.11: Hybrid Optimization Process: RSM Optimized SMSE parameters N_f and Δf maximizing and minimizing the correlation degradation metric, C_{Deg}.

	N_f	$\Delta f\ (kHz)$	$W_{SMSE}\ (MHz)$
Minimized C_{Deg}	16	6.47	0.2
Maximized C_{Deg}	32	25.02	1.6

Based on these results, the final solution for a coexistent SMSE signal that is the most disruptive to DSSS system performance, as indicated by maximal DSSS code tracking degradation, is $N_f = 32$ and $\Delta f = 25.02\ kHz$, as shown in Table 4.11. This was accomplished by finding a coarse GA solution followed by fine-tuning with the RSM process. To visually compare the final solutions of the RSM process, histograms of the final minimized and maximized C_{Deg} values are provided in Fig. 4.14 and Fig. 4.15, respectively.

Given the resultant W_{SMSE} for each optimized case in Table 4.11, all SMSE power is contained within $W_{RF} = 2.046\ MHz$ and thus the RF filtering effects contribute minimally to the performance difference. The performance difference is most attributable to the relationship between SMSE subcarrier spacing and the spectral structure of the DSSS spreading signal, i.e., the degree of SMSE-DSSS spectral coincidence. For parameters used in these imperfect tracking scenarios, the spectral characteristics of the DSSS spreading signal and received SMSE signals can be summarized as follows:

1. The spectral lines for the $N_c = 1023$-chip DSSS spreading code are spaced $\Delta f_{chip} = R_{chip}/N_c = 1\ KHz$ apart [25]. There are $W_{RF}/\Delta f_{chip} = 2,046$ total lines in $W_{RF} = 2 \times R_{chip} = 2.046\ MHz$ that are power weighted according to [25]

$$S_c\left(f - f_c\right) = \begin{cases} \frac{1}{N_c^2}, & f = f_c \\ \frac{N_c-1}{N_c^2} \operatorname{sinc}^2\left[\left(f - f_c\right)T_{chip}\right], & \text{Elsewhere} \end{cases},$$

where $\operatorname{sinc}(f) = \sin\left(f\right)/f$. The central line is located at f_c and remaining lines uniformly spaced on either side of f_c at intervals of $\Delta f_{chip} = 1\ KHz$.
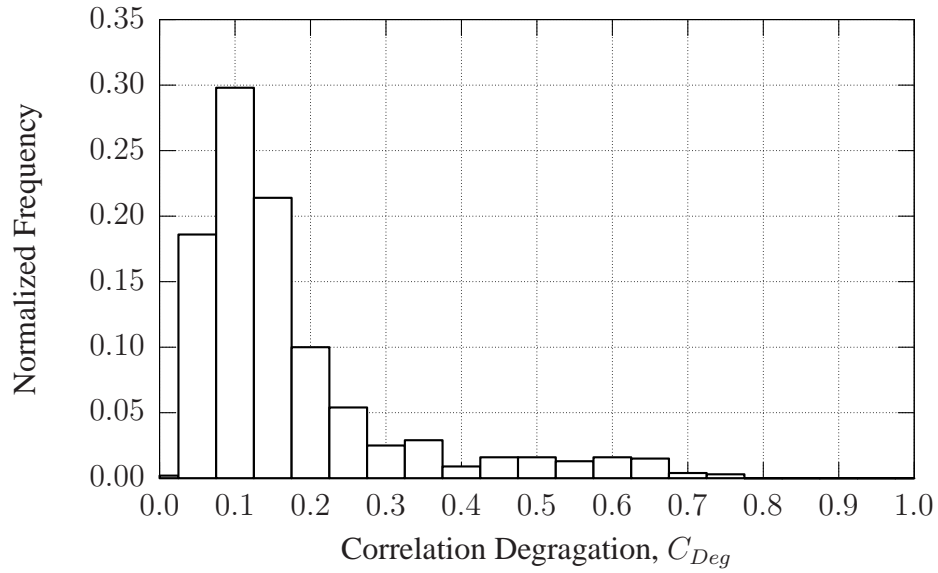
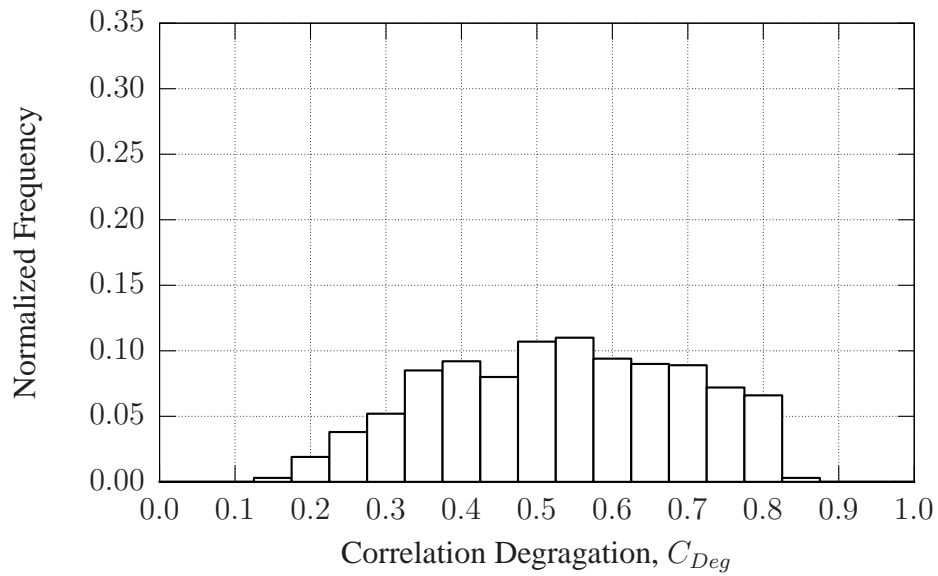Figure 4.14: Histogram showing behavior of C_{Deg} for the RSM solution that *MINIMIZES* C_{Deg}.

Figure 4.15: Histogram showing behavior of C_{Deg} for the RSM solution that *MAXIMIZES* C_{Deg}.

2. The coexisting SMSE signal spectrum is centered at f_c with an even number of BPSK data modulated subcarriers ($N_f/2$) uniformly spaced on either side of f_c. The two centrally located SMSE subcarriers are at Δf on either side of f_c and all remaining subcarriers spaced at uniform intervals of Δf from these. By design, the total SMSE power I is equally distributed across all subcarriers.

The degree of spectral coincidence between the DSSS spreading code, as used in the DSSS receiver to "despread" the received signals, and received SMSE signals dictates P_b performance. This is best characterized by considering the relationship between the DSSS spectral line spacing Δf_{chip} and the SMSE subcarrier spacing Δf, or more directly, by considering the resultant power spectrum at the despreading mixer output (DSSS detector input) in the DSSS receiver. The "despread" spectral response is the convolution of the uniformly spaced DSSS spectral lines at $\Delta f_{chip} = 1\ KHz$ with the uniformly spaced SMSE subcarriers spaced at Δf.

For the least disruptive solution in Table 4.11 the optimized subcarrier spacing of $\Delta f = 6.47\ KHz$ is a non-integer multiple of $\Delta f_{chip} = 1\ KHz$. Therefore, for a given shift-multiply-integrate operation of the spectral convolution process there is a most one DSSS spectral line that is aligned with an SMSE subcarrier and the peak response of all other SMSE subcarriers falls approximately midway between other spectral lines–minimal SMSE-DSSS spectral coincidence. As a result, the amount of power in the "despread" spectral response that falls within the DSSS detector bandwidth ($W_{Det} = 2 \times R_{Sym} = 2\ KHz$) is relatively low and the interfering SMSE signal has minimal impact on DSSS P_b performance.

On the other hand, for the most disruptive solution in Table 4.11, the optimized subcarrier spacing of $\Delta f = 25.02\ KHz$ is approximately an integer multiple of $\Delta f_{chip} = 1\ KHz$. Therefore, for a given shift-multiply-integrate operation of the spectral convolution process the peak responses of all SMSE subcarriers align near perfectly with specific DSSS spectral lines–maximum SMSE-DSSS spectral coincidence. As a result, the amount of power in the "despread" spectral response that falls within the DSSS detector bandwidth

($W_{Det} = 2 \times R_{Sym} = 2\ KHz$) is maximum and the interfering SMSE signal has maximum impact on DSSS P_b performance. Alternately stated, this SMSE solution in this case is more spectrally "matched" to the DSSS signal which causes greater degradation.

4.2.4   Optimal SMSE Parameter Demonstration.   Two additional metrics were calculated to demonstrate waveform optimization effects using the final RSM optimized solutions from Section 4.2. First, DSSS end-to-end bit error rate P_b was calculated to demonstrate the communication system applicability. Second, relative timing jitter, J_{code}, between the DLL-generated prompt code $c_p(t)$ and the incident DSSS code $c_i(t)$ was calculated to demonstrate applicability to DSSS-based navigation and precision timing system.

Communication system P_b results are shown in Table 4.12. These results were generated using the physics-based analytic DSSS receiver model with Monte Carlo simulation as shown in Section 4.1. The ideal results are provided for comparison and represent perfect code tracking conditions, i.e., the best that can be expected. The minimization and maximization results were obtained using the SMSE parameter solutions from the hybrid optimization process. For the degraded cases, the SMSE waveform effects on the DSSS receiver were isolated to DLL tracking performance only; no SMSE signal was present in the DSSS demodulator.

Timing jitter J_{code} results are shown in Table 4.13. For this analysis, J_{code} was defined as the the time difference between the coded pulse transition points in $c_i(t)$ and $c_p(t)$, with negative values for J_{code} indicating that $c_p(t)$ transitioned before $c_p(t)$ and positive values indicating that $c_p(t)$ transitioned after $c_p(t)$. Table 4.13 shows the standard deviation of the J_{code} metric.

Table 4.12:   Communication system bit error rate (P_b) for ideal code tracking and degraded code tracking using final SMSE parameters from hybrid GA-RSM optimization process in Section 4.2.

C_{Deg} Optimization	N_f	$\Delta f(kHz)$	P_b
Ideal	N/A	N/A	1.0×10^{-3}
Minimized	16	6.47	5.6×10^{-2}
Maximized	32	25.02	2.2×10^{-1}

Table 4.13:   Timing jitter standard deviation J_{code} for degraded code tracking using final SMSE parameters from hybrid GA-RSM optimization process in Section 4.2.

C_{Deg} Optimization	N_f	$\Delta f(kHz)$	J_{code} (sec)
Minimized	16	6.47	1.83×10^{-7}
Maximized	32	25.02	1.94×10^{-7}

V. Conclusion

5.1 Research Summary

The Spectrally Modulated, Spectrally Encoded (SMSE) framework provides an effective means for implementing Orthogonal Frequency Division Multiplexing (OFDM) signals and the ability to efficiently generate them has only recently become practical from a hardware perspective. As such, OFDM technology has been identified as a bedrock technology for fourth generation (4G) communications based on Cognitive Radio (CR) and Software Defined Radio (SDR) techniques [17]. The inherent flexibility of OFDM has generated significant research interest [11, 20, 34–37, 40] and the expanding pool of OFDM variants drove the need for a unified framework. As utilized under this research, the resultant SMSE framework effectively embodies the class of OFDM-based signals [27–30].

As 4G SMSE communications emerge they must coexist with other systems while competing to use available communication resources. Thus, coexistent interference becomes a concern, especially when these systems operate without orthogonality in frequency, time, space, polarization, and/or coding. Given a lack of orthogonality and limited available bandwidth, these signals must be designed to spectrally coincide while inducing "manageable" levels of mutual interference. This becomes particularly challenging when fundamentally different waveform modulations and the waveform design procedure often resorts to trial and error design methods [25, 31]. The goal of this research was to demonstrate a more structured, optimal means for SMSE waveform design using techniques commonly employed in the operations research field. A survey of general optimization techniques revealed that two methods were particularly applicable to the coexistent SMSE waveform design scenario, including *Genetic Algorithm* (GA) and *Response Surface Methodology* (RSM) optimization techniques.

Each of these optimization techniques are used to demonstrate SMSE waveform design in a coexistent scenario containing an SMSE signal and a Direct Sequence Spread Spectrum (DSSS) system operating over an Additive White Gaussian Noise (AWGN) channel. The DSSS system employs a non-coherent Delay-Lock Loop (DLL) for code tracking. The specific DLL implementation used here, along with defined correlation metrics

that characterize code tracking conditions, is sufficiently general such that the optimization demonstrations herein are broadly applicable to other non-communication applications employing DLL tracking, e.g., precision navigation, timing, geolocation, etc.

The coexistent SMSE-DSSS waveform design process is demonstrated herein under both *perfect* and *imperfect* DLL code tracking conditions. Under both conditions the number of subcarriers N_f and subcarrier spacing Δf are the SMSE design variables of interest. Under *perfect* DLL code tracking conditions, the GA and RSM optimization processes are considered independently with the objective function to be optimized (minimized and maximized) being DSSS bit error rate P_b. A hybrid GA-RSM optimization process is used under more realistic *imperfect* DLL code tracking conditions. In this case, P_b optimization is accomplished through a correlation degradation metric with the GA process being first applied to generate a "coarse" solution followed by RSM processing to provide the final optimized solution.

For all perfect and imperfect DLL code tracking scenarios considered, the optimized DSSS P_b *minimization* results yielded SMSE waveform designs and P_b performance that was consistent with scenarios having no coexistent SMSE signal present (best-case coexistent performance). For the optimized DSSS P_b *maximization* solutions, worst-case SMSE-DSSS coexistence was achieved for SMSE waveform designs that were spectrally "matched" to the DSSS signal, i.e., greatest P_b degradation was experienced when the resultant SMSE subcarrier spacing Δf was an integer multiple of the spectral line spacing Δf_{chip} of the DSSS spreading code.

The research objective has been achieved in the sense that 4G communications design engineers now have one additional tool at their disposal. This work has successfully expanded the practical utility of a previously developed tool, the original SMSE framework [26, 28, 30], by demonstrating a more efficient, structured means for coexistent waveform design that replaces previous trial and error methods. As such, the communications community is one step closer to actually hitting the bedrock of OFDM-based signaling using the SMSE framework. The significance of this has been acknowledged through ac-

ceptance of the independent GA [3] and RSM [4] optimization results under *perfect* DLL code tracking conditions. Applicability of the hybrid GA-RSM processing technique under *imperfect* DLL code tracking conditions has been acknowledged as well [2].

5.2   *Recommendations for Future Research*

Given that applicability of optimization techniques to SMSE waveform design has been demonstrated and well-received by the technical community, there are many additional research topics that could be investigated. Some of the most evident future research avenues include:

- The research assumptions in Chapter I could be removed and/or relaxed and the research process repeated. In some cases, this would represent a somewhat trivial exercise and produce results that might be expected. For example, the coexistent SMSE-DSSS demonstration results in Chapter V could easily be expanded by considering alternative OFDM-based communication variants (CI-OFDM, COFDM, MC-CDMA, etc.). In other cases, relaxing the research assumptions could prove to be far more challenging and could produce results which are far more significant.

- The demonstrations here focused on SMSE waveform design through appropriate selection of two variables, N_f and Δf. For all cases, the total SMSE power was equally distributed across all selected subcarriers. Research could be conducted that maintains the same coexistent SMSE-DSSS scenarios and GA/RSM optimization objective functions but with additional consideration given to incorporating the selection of SMSE weight vector $\mathbf{w_k}$. By treating each element of $\mathbf{w_k}$ as a model parameter, the ANOVA process could provide insight into the impact that individual subcarriers are having on the objective function. Assuming the optimization goal is to achieve best-case coexistence, the practical implication is that insignificant subcarriers (those inducing minimal interference) could transmit at appreciable power levels while power levels in significant subcarriers (those inducing maximum interference) could be reduced or set to zero (shut off).

- The approach to shutting off significant subcarriers is perhaps best characterized as being a *hard decision* process. Recent efforts have introduced overlay, underlay and hybrid overlay/underlay waveform design using the original SMSE framework with subcarrier selection and power distribution based on *soft decision* criteria [5, 6]. Given the inherent structure of the original SMSE framework is maintained in these efforts, the emergence of these new SD-SMSE techniques warrants future investigation given that the optimization methods demonstrated in this work should to be directly applicable.

- Demonstrations in this work are based on *internally* observable knowledge of how the coexistent DSSS system, and in particular its DLL code tracking, is responding to various SMSE parameter changes. This internal knowledge will generally be unavailable and thus alternatives need to be investigated. Two potential alternatives could be considered. First, cooperative exchange of system performance status among coexistent system(s) through communication back channels. While this approach generally requires additional communication resources, which may result in fewer resources being available for the primary function, similar techniques have been employed in communication networks. Second, the objective function behavior could be determined passively without any aid from coexistent systems. While this approach is technically more challenging, it is consistent with envisioned 4G communication goals for CR-based SDR communications that will use *externally* observable knowledge to adapt and optimize performance.

- The specific non-coherent DLL implementation used here, along with defined correlation metrics for characterizing code tracking conditions, are sufficiently general such that the optimization processes considered are broadly applicable to other non-communication applications employing DLL tracking, e.g., precision navigation, timing, geolocation, etc. Therefore, a similar SMSE coexistence analysis and demonstration could be conducted within each of these application areas.

5.3 Acknowledgement

Bibliography

1. Akyildiz, I.F., W. Lee, M.C. Vuren, and S. Mohanty. "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey". *Elsevier Computer Networks*, 50:2127-2159, May 2006.

2. Beard, T.W. and M.A. Temple. *Applications and Methods of Waveform Diversity*. SciTech Publishing, Inc. Chapter C-V-8: Coexistent SMSE Waveform Design Using Optimization Techniques, To appear: 2nd Quarter 2008.

3. Beard, T.W., M.A. Temple, J.O. Miller, and R.F. Mills. "Using Genetic Algorithms for Spectrally Modulated Spectrally Encoded waveform Design". *Int'l Waveform Diversity and Design Conf (WDD07)*, 265–269, Jun 2007.

4. Beard, T.W., M.A. Temple, and M.L. Roberts. "An Experimental Design Approach for Optimizing SMSE Waveforms to Minimize Coexistent Interference". *IEEE Int'l Conf on Comm (ICC07)*, 5581–5585, Jun 2007.

5. Chakravarthy, V., Z. Wu, A. Shaw, M.A. Temple, R. Kannan, and F. Garber. "A General Overlay/Underlay Analytic Expression Representing Cognitive Radio Waveform". 3rd *IEEE Int'l Waveform Diversity and Design Conf (WDD07)*, 69–73. 2007.

6. Chakravarthy, V., Z. Wu, M.A. Temple, A. Shaw, and F. Garber. "Novel Overlay/Underlay Cognitive Radio Waveforms Using SD-SMSE Framework to Enhance Spectrum Efficiency - Part I". *IEEE Trans on Comm*, Aug 2008, Under Review.

7. Christensen, Ronald. *Plane Answers to Complex Questions: The Theory of Linear Models*. Springer-Verlag, New York, New York, 2nd Edition, 1996.

8. Defense Research Projects Agency (DARPA). "XG Communications Program Overview". URL www.darpa.mil/sto/solicitations/WAND, Feb 2007.

9. Devroye, N. and P. Mitran. "Achievable Rates in Cognitive Radio Channels". *IEEE Trans on Info Thy*, 52(5):1813–1827, May 2006.

10. Dynamic Spectrum Access Networks (DySPAN). URL www.ieee-dyspan.org.

11. Engels, M. *Wireless OFDM Systems: How to Make Them Work?* Kluwer Academic Publishers, Boston, MA, 2002.

12. Federal Communications Commission. "Report of the Spectrum Efficiency Working Group". *Spectrum Policy Task Force*, 15 Nov 2002.

13. Glossner, J., D. Iancu, J. Lu, E. Hokenek, and M. Moudgill. "A Software-Defined Communications Baseband Design". *IEEE Comm Mag*, 41(1):120–128, Jan 2003.

14. Hahn, Gerald J. and Samuel S. Shapiro. *Statistical Models in Engineering*. John Wiley and Sons, Inc., 1967.

15. Hart, B.W., R.D.J Van Nee, and R. Prasad. "Bit Error Probability Degradation Due to Code Tracking Errors in Spread-Spectrum Communication Systems". 6th *IEEE Int'l Symp on Personal, Indoor and Mobile Radio Communications (PIMRC95)*, 3:1016–1024, Sep 1995.

16. Hart, B.W., R.D.J. Van Nee, and R. Prasad. "Performance Degradation Due to Code Tracking Errors in Spread-Spectrum Code-Division Multiple-Access Systems". *IEEE Jour on Sel Areas in Comm*, 14(8):1669–1679, Oct 1996.

17. Haykin, S. "Cognitive Radio: Brain-Empowered Wireless Communications". *IEEE Jour on Sel Areas in Comm*, 23(2):201–220, Feb 2005.

18. Hillier, F.S. and G.J. Leiberman. *Introduction to Operations Research*. McGraw-Hill, New York, New York, 8th Edition, 2005.

19. "Joint Tactical Radio System (JTRS) Program". URL `http://jtrs.army.mil`.

20. LeFloch, B., M. Alard, and C. Berrou. "Coded Orthogonal Frequency Division Multiplex". *Proc of IEEE*, 83(6):982–996, Jun 1995.

21. Misra, P., B.P. Burke, and M.M. Pratt. "GPS Performance in Navigation". *Proc of IEEE*, 87(1):65–85, Jan 1999.

22. Montgomery, D.C. *Design and Analysis of Experiments*. John Wiley and Sons, Hoboken, New Jersey, 6th Edition, 2005.

23. *Network Centric Warfare*. Report to Congress, Department of Defense, Washington, DC, July 31, 2001. URL http://www.dod.mil/nii/NCW/.

24. "Networking Will Drive ISR Success". *C4ISR Online Journal*. URL http://www.isrjournal.com, Aug 2004.

25. Peterson, R.L., R.E. Ziemer, and D.E. Borth. *Introduction to Spread Spectrum Communications*. Prentice Hall, Englewood Cliffs, NJ, 1995.

26. Roberts, M.L. "A General Framework for Analyzing, Characterizing, and Implementing Spectrally Modulated, Spectrally Encoded Signals". Doctoral Dissertation, AFIT/DS/ENG/ENG/06-06, Air Force Institute of Technology, 2006.

27. Roberts, M.L., M.A. Temple, R.F. Mills, and R.A. Raines. "Interference Suppression Characterization for Spectrally Modulated, Spectrally Encoded Signals". *IEE Electronic Letters*, 42(19):1103–1104, Sep 2006.

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". *IEEE Int'l Conf. on Waveform Diversity and Design (WDD06)*, Jan 2006.

29. Roberts, M.L., M.A. Temple, M.E. Oxley, R.F. Mills, and R.A. Raines. "A Spectrally Modulated, Spectrally Encoded Analytic Framework for Carrier Interferometry Signals". *ACM Int'l Wireless Commun. and Mobile Computing Conf (IWCMC)*, Jul 2006.

30. Roberts, M.L., M.A. Temple, R.A. Raines, R.F. Mills, and M.E. Oxley. "Communication Waveform Design Using an Adaprive Spectrally Modulated, Spectrally Encoded (SMSE) Framework". *IEEE Jour on Sel Topic in Sig Proc*, 1(1):203–213, Jun 2007.

31. Sklar, B. *Digital Communications: Fundamentals and Applications*. Prentice Hall, Englewood Cliffs, NJ, 2nd Edition, 2001.

32. Spilker, J.J. and D.T. Magill. "The Delay-Lock Discriminator–An Optimum Tracking Device". *Proceedings of the IRE*, 49(9):1403–1416, Sep 1961.

33. U.S. Department of Commerce, National Telecommunications and Information Administration (NTIA). "United States Frequency Allocation Chart". http://www.ntia.doc.gov/osmhome/allochrt.pdf, Oct 2003.

34. Van Nee, R.D.J. and R. Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, Boston, MA, 2000.

35. Wiegandt, D.A. and C.R. Nassar. "High Performance OFDM via Carrier Interferometry". *IEEE Int'l Conf on 3rd-Generation Wireless and Beyond (3GWireless01)*, 404–409. 2001.

36. Wiegandt, D.A., C.R. Nassar, and Z. Wu. "Overcoming Peak-to-Average Power Ratio Issues in OFDM via Carrier Interferometry Codes". *IEEE Vehicular Tech Conf (VTC)*, 660–663. 2001.

37. Wiegandt, D.A., Z. Wu, and C.R. Nassar. "High-Throughput, High-Performance OFDM via Pseudo-Orthogonal Carrier Interferometry Spreading Codes". *IEEE Trans on Comm*, 51(7):1123–1134, Jul 2003.

38. Zhao, Q. and B.M. Sadler. "A Survey of Dynamic Spectrum Access". *IEEE Trans on Communications*, 24(3):79–89, May 2007.

39. Zhao, Q. and A. Swami. "A Survey of Dynamic Spectrum Access: Signal Processing and Networking Perspectives". *IEEE Int'l Conf on Acoustics, Speach and Signal Processing (ICASSP07)*. Vol. 4, pp. 1349-1352, 2007.

40. Zou, W.Y. and Y. Wu. "COFDM: An Overview". *IEEE Trans on Broadcasting*, 41(1):1–8, Mar 1995.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
20-09-2008	**Doctoral Dissertation**	September 2005-September 2008

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Application of Optimization Techniques to Spectrally Modulated, Spectrally Encoded Waveform Design	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Beard, Todd W., Major, USAF	ENG 08-275
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765 DSN: 785-3636	AFIT/DEE/ENG/08-16

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
Air Force Research Laboratory Attn: AFRL/RYRE (Mr. Vasu Chakravarthy) 2241 Avionics Circle, Bldg 620 WPAFB OH 45433-7734 Vasu.Chakravarthy@wpafb.af.mil	AFRL
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A design process is demonstrated for a coexistent scenario containing Spectrally Modulated, Spectrally Encoded (SMSE) and Direct Sequence Spread Spectrum (DSSS) signals. Coexistent SMSE-DSSS designs are addressed under both *perfect* and *imperfect* DSSS code tracking conditions using a non-coherent delay-lock loop (DLL). Under both conditions, the number of SMSE subcarriers and subcarrier spacing are the optimization variables of interest. For *perfect* DLL code tracking conditions, the GA and RSM optimization processes are considered independently with the objective function being end-to-end DSSS bit error rate. A hybrid GA-RSM optimization process is used under more realistic *imperfect* DLL code tracking conditions. In this case, optimization is accomplished using a correlation degradation metric with the GA process being first applied to generate a "coarse" solution followed by RSM processing which provides the final optimized solution. This work has successfully expanded the practical utility of a previously developed tool, the original SMSE framework, by demonstrating a more efficient, structured means for coexistent waveform design that replaces previous trial and error methods.

15. SUBJECT TERMS

SMSE, OFDM, Waveform Design, Optimization, Genetic Algorithm, Response Surface Methodology,

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
REPORT **U**	**ABSTRACT** **U**	**c. THIS PAGE** **U**	**UU**	104	Dr. Michael A. Temple (ENG)
					19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x4279; email: Michael.Temple@afit.edu