Multi-Objective Optimization of FM Noise Waveforms via Generalized Frequency Template Error Metrics

By

Charles Mohr

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

Dr. Shannon Blunt, Chairperson

Committee members

Dr. Chris Allen

Dr. Jim Stiles

Date defended: _____

The Thesis Committee for Charles Mohr certifies
that this is the approved version of the following thesis :

Multi-Objective Optimization of FM Noise Waveforms via Generalized Frequency Template Error Metrics

Dr. Shannon Blunt, Chairperson

Date approved: _____

Abstract

FM noise waveforms have been experimentally demonstrated to achieve high time bandwidth products and low autocorrelation sidelobes while achieving acceptable spectral containment in physical implementation. Still, it may be necessary to further reduce sidelobe levels for detection or improve spectral containment in the face of growing spectral use. The Frequency Template Error (FTE) and the Logarithmic Frequency Template Error (Log-FTE) metrics were conceived as means to achieve FM noise waveforms with good spectral containment and good autocorrelation sidelobes. In practice, FTE based waveform optimizations have been found to produce better autocorrelation responses at the expense of spectral containment while Log-FTE optimizations achieve excellent spectral containment and interference rejection at the expense of autocorrelation sidelobe levels. In this work, the notion of the FTE and Log-FTE metrics are considered as subsets of a broader class of frequency domain metrics collectively termed as the Generalized Frequency Template Error (GFTE). In doing so, many different P-norm based variations of the FTE and Log-FTE cost functions are extensively examined and applied via gradient descent methods to optimize polyphase-coded FM (PCFM) waveforms. The performance of the different P-norm variations of the FTE and Log-FTE cost functions are compared amongst themselves, against each other, and relative to a previous FM noise waveform design approach called Pseudo-Random Optimized FM (PRO-FM). They are evaluated in terms of their autocorrelation sidelobes, spectral containment, and their ability to realize spectral notches within the 3 dB bandwidth for the purpose of interference rejection. These comparisons are performed in both simulation and experimentally in loopback where it was found that

P-norm values of 2 tend to provide the best optimization performance for both the FTE and Log-FTE optimizations except in the case of the Log-FTE optimization of a notched spectral template where a P-norm value of 3 provides the best results. In general, the FTE and Log-FTE cost functions as subsets of the GFTE provide diverse means to optimize physically robust FM noise waveforms while emphasizing different performance criteria in terms of autocorrelation sidelobes, spectral containment, and interference rejection.

Acknowledgements

Attending graduate school has been one of best decisions I have made, but at the same time it has never been easy. With this in mind I would like to thank the people most responsible for helping me learn and grow throughout this journey including Dr. Shannon Blunt, who for the past four years has been an outstanding mentor, my wife, Eva for keeping sane when I'm feeling overwhelmed, and my children Willemina and Felix for always bringing me joy.

Addional thanks is due to the sponsors of this work, as it was supported in part by the Office of Naval Research under Contract #N00014-16-C-2029 and by a subcontract with Matrix Research, Inc. for research sponsored by the Air Force Research Laboratory under Prime Contract #FA8650-14-D-1722. DISTRIBUTION STATEMENT A. Approved for Public Release.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

Radar as a concept is familiar to the general public. A radar system transmits some kind of energy. It "listens" for reflections and it uses those reflections to deduce information about the environment which is in principle similar to the echolocation process used by bats and dolphins. Conceptually the process is straightforward, but, in practice, designing a system which can accurately characterize the environment in this way spans numerous science and engineering disciplines and is the subject of decades of research. In part, this is because radar systems are often tasked with much more than the simple ranging and detection of objects. Radars have been used to perform such diverse tasks as tracking the motion and position of aircraft or ground based vehicles, detecting and mapping the entire life cycle of thunderstorms, imaging the surface of the Earth from space or even the surfaces of other planets such as Venus and Saturn's moon Titan, and much more. Understandably, performing any one of these tasks requires knowledge of electromagnetics, radio frequency (RF) design, mechanical engineering, and as the different tasks demand it, disciplines such as geology or meteorology and others. This list of disciplines is by no means exhaustive but it does give an idea of the diverse research space radar occupies. With this in mind, the purpose of this chapter is to present the radar problem in its most basic, generic terms and then to distill these ideas down to the parts which are most relevant to this work. This achieves two goals in that it provides the necessary background to understand the material and it gives context to the problem within the larger radar problem space.

1.1 The Radar Range Equation

The word radar is ubiquitous enough that the fact that it is an acronym has largely been lost to the general public, but originally the word "RADAR" stood for RAdio Detection And Ranging. Consider the simple diagram in Fig. 1.1 which shows a radar performing detection and ranging in a monostatic mode, that is, the transmitter and receiver are located together.

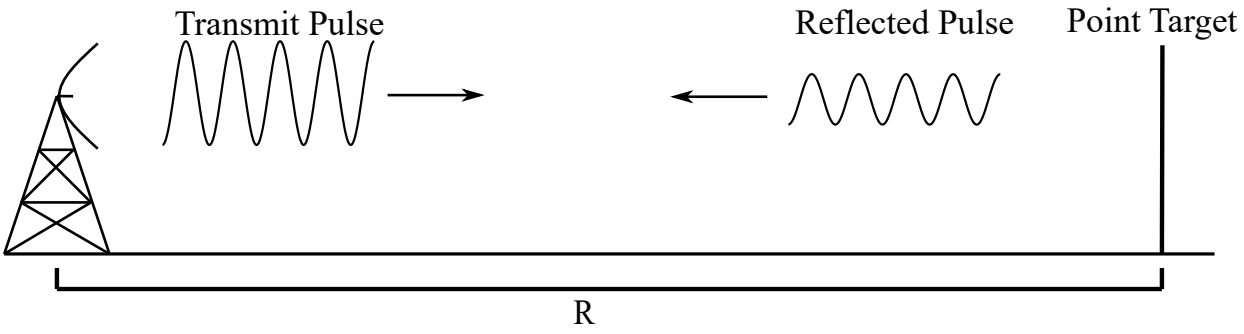

Figure 1.1: A monostatic radar transmits pulses and receives echoes from the environment

The radar transmits electromagnetic energy either in a pulsed manner (as in Fig. 1.1) or continuously and it "listens" for echoes from the environment. Despite this simple setup, the equation which describes the returned power from a generic object consists of several parts. To emphasize the role and origin of each term, the received power equation is written in both an expanded bistatic and compact monostatic form as [1]

$$P_r = \left(\frac{P_t G_t}{L_t}\right)\left(\frac{1}{4\pi R_t^2 L_{at}}\right)(\sigma)\left(\frac{1}{4\pi R_r^2 L_{ar}}\right)\left(\frac{G_r \lambda^2}{4\pi L_r}\right) = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L} \tag{1.1}$$

Each term in the expanded form of (1.1) represent either gains or losses from distinct parts of the transmit receive path in the order they are accrued.

$\frac{P_t G_t}{L_t}$ - The radar system transmits at a nominal power of P_t. This power is focused by a factor of G_t relative to an isotropic antenna, an antenna which radiates energy uniformly in all directions. Some portion of P_t is absorbed by the transmitter electronics which is accounted for in the term L_t.

$\frac{1}{4\pi R_t^2 L_{at}}$ - The transmit power is attenuated by spherical spreading losses over the distance from the transmitter to the object, R_t, and the atmospheric losses L_{at} over the same path due to such affects as absorption, diffraction, and refraction.

σ - The transmit signal is "reflected" by the object with some magnitude σ known as the radar cross section in units of square meters. The term reflected is used loosely here to indicate power is returned from the object. In reality, the factors and processes that contribute to the value of σ are various and complex. In fact, entire text books have been devoted to analyzing the radar cross sections of various media. For simplicity, it is simply considered a constant here.

$\frac{1}{4\pi R_r^2 L_{ar}}$ - The transmit power is attenuated by spherical spreading losses over the distance from the object to the receiver, R_r, and the atmospheric losses L_{ar} over the same path due to such affects as absorption, diffraction, and refraction.

$\frac{G_r \lambda^2}{4\pi L_r}$ - Finally, the power signal reaches the receive antenna and is modified by the antenna's effective aperture which is a function of both the receive antenna gain, G_r and the carrier frequency wavelength, λ. Additionally, energy is absorbed by the receivers electronics which is taken into account by the term L_r.

In (1.1), all the terms discussed above are written in a much more compact form by assuming a monostatic configuration in that the transmitter and receiver are collocated and the same antenna is used for both transmit and receive. Consequently, $G_t = G_r$ and $R_t = R_r$. Additionally, the various loss terms have been combined into the single loss term L. As a point of emphasis, which will be revisited later, the R^4 term in the denominator due to the two-way spherical spreading loss presents a significant challenge for the radar designer. An object at a distance $2R$ will only return $1/16$ the power as the same object located at a distance R. To account for this, radar systems have to operate at very high power levels to detect distant targets, and they must be able to handle a very large dynamic range as close targets will, in general, return much more energy than distant targets.

In the context of detection, however, it is more important to know the returned power relative to the power of other sources such as noise. In general, the noise due to the active electronics in the receiver dominates the noise due to other sources. Considering only the receiver noise, the noise power can be written as

$$P_n = kT_0FB \tag{1.2}$$

where k is Boltzmann's constant (1.38×10^{-23} watt-sec/K), T_0 is the standard temperature (290K), F is the receiver noise figure (unitless), and B is the receiver bandwidth (Hz). The form of (1.2) arises from the assumption that the receiver noise is *white*. That is the power spectral density, the power as a function of frequency, is constant over all frequency. Putting (1.1) and (1.2) together, the signal to noise ratio (SNR) is

$$SNR = \frac{P_tG^2\lambda^2\sigma}{(4\pi)^3R^4LkT_0FB} \tag{1.3}$$

In general, the radar SNR is what determines the performance of a radar system, since the SNR directly correlates with the ability of processing algorithms to correctly identify targets which in itself is an expansive field. In reality there can easily be other signals which complicate and detrimentally affect the detection process but for now it is instructive to only consider the noise as an interferer.

1.2 Pulsed and Continuous Wave Operation

Broadly speaking, radars are said to either operate in a continuous wave (CW) mode where the transmitter is always on and the receiver is always listening, or in a pulsed mode where the transmitter is turned on for only fractions of a second at a time and the receiver only listens while the transmitter is off.

CW radars are used in numerous applications such automotive collision avoidance, police radars, or level sensors in storage tanks [1]. In general however, they are limited by there trans-

mitter/receiver isolation. That is, signals from the transmitter directly pass to the receiver causing self interference and a loss in performance. Consequently, these systems are typically operated at a relatively low power level and thus short range mode. For this work, it is more critical to cover the operation of pulsed radars in more detail.

Consider the pulsed radar operation depicted in Fig. 1.1. At regular intervals, the radar transmitter is turned on for T seconds thereby emitting an electromagnetic pulse T seconds long. Typical values of T are on the order of microseconds, as small as nanoseconds, or as large as milliseconds. In between transmissions, the radar receiver listens for reflected pulses. Otherwise, the receiver would be overwhelmed by the high power of the transmitter.

More formally pulsed operation can be defined by several parameters. The pulse repetition interval (PRI), here refered to as T_{PRI}, is the time between pulses. The pulse repetition frequency (PRF), here refered to as f_{PRF}, is the reciprocal of the PRI. The PRF represents the number of pulses per second in Hz. Taking into account the pulse duration, the duty cycle d is the fraction of time the transmitter is turned on within a PRI defined as

$$d = \frac{T}{T_{PRI}} = T f_{PRF} \tag{1.4}$$

With the duty cycle defined it now makes sense to revisit equation (1.2). In (1.2), the transmit power P_t appears to be constant, but in a pulsed mode the transmit power varies as a function of time. In fact, it goes from a very high peak power during the pulse time down to zero in receive mode. The average of these two extremes can be written as

$$P_{avg} = P_t T f_{PRF} \tag{1.5}$$

where P_t is now explicitly the peak transmit power of the system. Then, combining equations (1.3) and (1.5) creates

$$SNR = \frac{P_t T f_{PRF} G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L k T_0 F B} \tag{1.6}$$

which takes into account the variable power (on or off) of pulsed operation.

1.3 Pulse Integration

Intuitively, it seems reasonable that by "staring" at a particular target longer, thus collecting a greater amount of returned energy, the SNR should improve. For CW this entails simply "staring" at the target longer while for pulsed operation, this corresponds to transmitting more pulses in the same direction. However, the form of (1.6) has no mechanism for this. If the numerator and denominator are converted to energy by multiplying them individually by some period of time, this factor would cancel and return the same expected SNR as in (1.6). The missing piece here, that is not evident from the discussion so far, is that the numerator (signal power) in (1.6) integrates more efficiently than the denominator. In fact, the SNR improves as a linear function of the number of pulses integrated or as a function of the dwell time, T_d in a CW context. To explain this briefly, consider the mean of a set of independent, identically distributed (iid), zero mean random variables such that

$$Y = \frac{1}{N}\sum_{n=1}^{N} X_n \tag{1.7}$$

The variance of Y is then

$$\text{var}[Y] = \frac{1}{N^2}\text{var}\left[\sum_{n=1}^{N} X_n\right] = \frac{1}{N^2}\text{E}\left[\left(\sum_{n=1}^{N} X_n\right)^2\right] \tag{1.8}$$

where the variance in this case can be simply be written as the expectation of the variable squared since it has been defined as zero mean. The $\frac{1}{N}$ has also been pulled out of the variance operator since $\text{var}[aV] = a^2\text{var}[V]$. Then by considering all the terms in the squared sum, (1.8) becomes

$$\text{var}[Y] = \frac{1}{N^2}\text{E}\left[\sum_{n=1}^{N}\sum_{i=1}^{N} X_n X_i\right] \tag{1.9}$$

Since the variables are independent, every covariance term is zero such that (1.9) becomes

$$\text{var}[Y] = \frac{1}{N^2}\text{E}\left[\sum_{n=1}^{N} X_n^2\right] \tag{1.10}$$

Finally, if the variance of each X is σ^2, the resulting variance of Y is

$$\text{var}[Y] = \frac{1}{N^2}(N\sigma^2) = \frac{\sigma^2}{N} \tag{1.11}$$

Thus, taking the mean of the sum of N iid random variables results in a factor of N reduction in the variance. For noise, the variance is the power. Thus coherent integration results in a factor of N decrease in noise power. A more thorough discussion of this topic and others related to the behavior of noise and noisy signals can be found in [2]. With this result, the radar range equation becomes

$$SNR = \frac{n_p P_t T f_{PRF} G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L k T_0 F B} \tag{1.12}$$

where n_p is the number of integrated pulses

1.4 Radar Ranging

Before discussing (1.12) further, it is necessary to define some fundamental radar parameters and their interactions. Consider a monostatic radar operating in a pulsed mode. At time t_0, it emits a pulse which then travels at the speed of light, c, or approximately 3×10^8 m/s. It hits some target at distance R and is immediately reflected back towards the radar covering the distance R once again. Stated succinctly, the round trip delay, t_d, of the pulse is

$$t_d = \frac{R}{2c} \tag{1.13}$$

With this equation measuring the distance to a target is as simple as measuring the arrival time of the leading edge of the pulse. Unfortunately, it isn't so simple in reality. When there are several

targets in close proximity to each other in addition to noise, simply measuring the arrival time of distinct reflections become very difficult as the different reflections will interfere with each other. To conceptualize this, we can define the notion of range resolution. Range resolution is the minimum distance at which two objects can be separated and still be unambiguously differentiated. In Fig. 1.2, the two point targets are separated by the distance ΔR, which as a function of c and the pulse length T is the minimum distance at which two targets can be spaced such that there is no overlap between their reflected pulses.

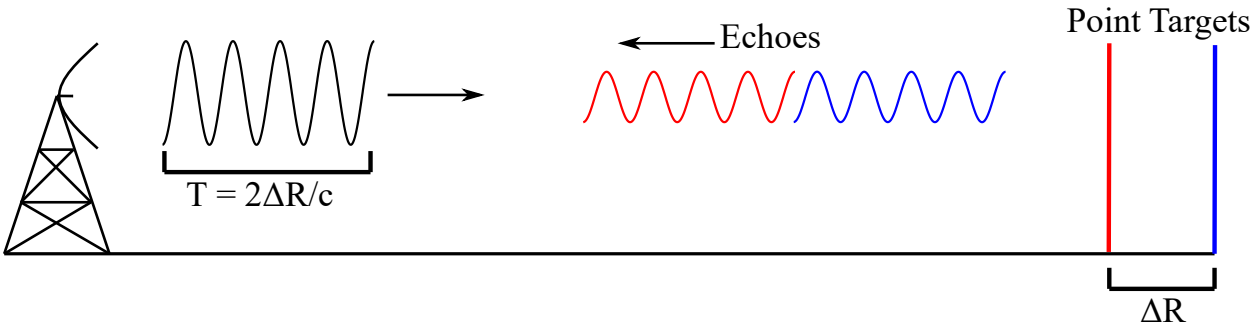

Figure 1.2: ΔR is the minimum distance at which reflected signals do not overlap

With this definition, the nominal range resolution of any pulsed radar system is then

$$\Delta R = \frac{Tc}{2} \tag{1.14}$$

In reality, the achievable range resolution of a radar system is more complicated than this simple formula, and through clever signal design and post processing it can even be decoupled from the length of the pulse allowing for overlap in the echoes. More will be said about this later.

Finally, consider (1.13) in the context of transmitting multiple pulses. Every T_{PRI} a new pulse is transmitted. In Fig. 1.3 one point target (PT1) is placed within the range R_{max} defined as

$$R_{max} = \frac{c}{2f_{PRF}} \tag{1.15}$$

R_{max} is the maximum distance a target can be located and have a pulse echo reach the receiver before the transmission of the next pulse. A second point target, (PT2), has been placed beyond

this range. Consequently, the echo of the first pulse from PT2 will not reach the receiver until *after* the transmission of the second pulse. Thus, if there is no way to differentiate between the first and the second pulse, it will be impossible to know whether that echo was a result of the first or the second pulse. If it is assumed to come from the second pulse, PT2 will appear much closer than it actually is. The range of PT2 is then ambiguous so R_{max} is the maximum unambiguous range. The obvious way to increase R_{max} is by decreasing f_{PRF}, such that the power returned by scatterers beyond R_{max} will be attenuated by the R^4 term in (1.12) to the point of irrelevance. However, decreasing f_{PRF} has other implications which will be discussed later.

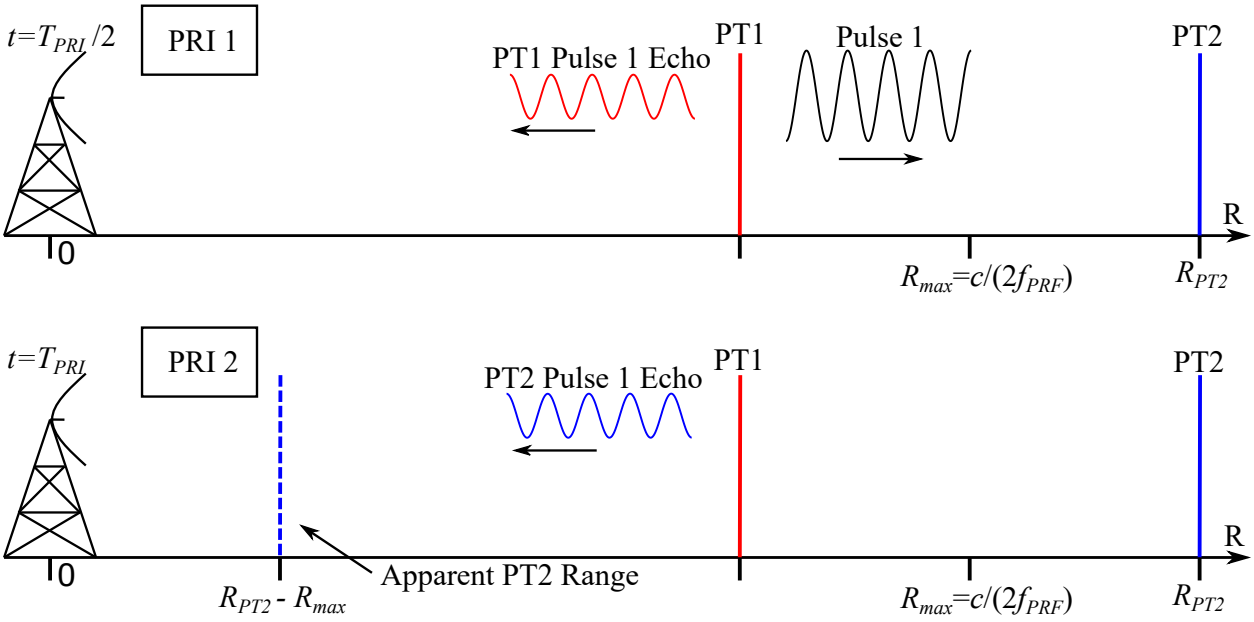

Figure 1.3: The second point target 2 (PT2) is place beyond the unambiguous range $R_{max} = c/(2f_{PRF})$. Consequently, PT2 appears much closer than it actually is.

1.5 Pulse Compression

From the R^4 term alone in (1.12) it is clear that pulsed radars either need to transmit at high peak power levels, transmit relatively long pulses, transmit pulses at a high f_{PRF}, integrate as many pulses as possible for a given look direction, or some combination thereof to simply detect targets at a desired distance. Unfortunately, the peak transmit power is limited by the capabilities of RF equipment, long pulse lengths correspond to poor range resolution as per (1.14), high pulse repe-

tition frequencies result in short unambiguous ranges, and, if a radar needs to scan a large angular area, it can only stare in one direction for so long thus limiting pulse integration. These considerations demonstrate the complex interrelationships at play in determining the ideal operating parameters for a radar. In reality, as with any complex engineering problem, the design goals have to be weighed against constraints and compromises will have to be made. However, as the topics have been laid out thus far, the most problematic relationship is actually between peak power, pulse length, and range resolution.

For example, consider a radar system operating at a pulse length of $T = 1\mu s$, a peak transmit power $P_t = 1$ MW, and a PRF of $f_{PRF} = 5$ kHz. These correspond to an average power of $P_{avg} = 5$ kW via (1.5), a range resolution of 150 m via (1.14), and an unambiguous range of 30 km via (1.15). Now consider these numbers in the context of the radar range equation in (1.12). Assuming the unambiguous range is the largest range of interest, the spherical spreading loss alone results in a loss factor of 1.2×10^{-18}, whereas at a close range of say .5 km, it results in a loss of 1.6×10^{-11}, representing a difference of almost 7 orders of magnitude. Amazingly, as a testament to the state of RF engineering, radars can and do operate at these dynamic ranges (or even greater), and at these levels of sensitivity. Other terms in the radar range equation help such as the radar cross section or the antenna gain, but the accomplishment is nonetheless impressive. The problem here though is that the range resolution is still 150 m which for many applications is unacceptably large. To make matters worse, many systems are power constrained such as on an aircraft in which case the radar may have to compensate by transmitting longer pulse leading to even worse range resolution. Up to now though, nothing has been said about what is being transmitted other than the signal has some nominal power level and pulse length. However, through the judicious design and post processing of radar signals the relationship in (1.14) can be greatly relaxed.

1.5.1 Signal Bandwidth

In discussing the bandwidth of radar signals it is useful to first think of the significance of the noise bandwidth term, B, in the denominator of (1.12). For the purpose of improving SNR, maximizing

(1.12) implies minimizing B thus minimizing the noise power. The obvious way to do this is to simply filter the receive signal either in analog or after digital sampling. Either way the noise power will be reduced. The problem here, however, is that the transmit and thus the desired signal also has some amount of bandwidth. Restricting the receive bandwidth too much will result in received signal loss and a net loss in SNR.

To begin this work's discussion of radar waveforms, first consider a time limited (pulsed) and constant frequency (tone) radar signal as

$$s(t) = \begin{cases} A\cos(2\pi f_c t + \phi) & t_0 < x < t_0 + T \\ 0 & \text{otherwise} \end{cases} \tag{1.16}$$

where A is some amplitude scale factor, f_c is the signal frequency, ϕ is some initial phase, and the pulse lasts for T seconds from time t_0 to $t_0 + T$. To determine the bandwidth of the signal we can directly apply the fourier transform to $s(t)$.

$$\mathbb{F}\{s(t)\} = S(f) = \frac{A}{2}\left[\frac{\sin(T\pi(f-f_c))}{\pi(f-f_c)} + \frac{\sin(T\pi(f+f_c))}{\pi(f+f_c)}\right]\exp[j(\phi - 2\pi f t_0)] \tag{1.17}$$

To simplify the analysis we can consider the baseband signal where $s(t)$ has been down-converted to DC since, ideally, this process preserves the spectral envelope. Then (1.17) becomes

$$S(f) = A\frac{\sin(T\pi f)}{\pi f}\exp[j(\phi - 2\pi f t_0)] \tag{1.18}$$

Finally, in the context of SNR we care about the power spectral density (PSD) of $s(t)$ so taking the absolute value squared of (1.18) yields

$$|S(f)|^2 = \left(A\frac{\sin(T\pi f)}{\pi f}\right)^2 \tag{1.19}$$

This function is plotted in Fig. 1.4, where a significant problem is made obvious. The point of deriving the PSD and plotting it to determine the signal bandwidth. However, from the form

of (1.19) and visually from Fig.1.4, there is some degree of spectral power for all frequencies. From a theoretical standpoint this result is unsurprising. It is well known in Fourier theory that no time limited signal is also frequency limited. From a practical and technical standpoint this presents a problem. If the bandwidth is defined to be the span of frequencies such that all signal power is contained within the band, the bandwidth will be infinite. This is known as the *absolute* bandwidth. Under this definition, the bandwidth of all radar signals would be identically infinite! After all, even CW systems turn on and off at some point. With this in mind, it is necessary to consider a more pragmatic definition of bandwidth. One such definition, is the 3dB bandwidth, or the bandwidth beyond which the PSD has decayed below 1/2 (3 dB) of the peak power of its peak power. This definition has its roots in filter theory where the bandwidth of a filter is often defined by its 3dB bandwidth. In the context of radar it additionally presents a convenient relationship when considering range resolution as will be discussed in the next section.

For the case of the unmodulated pulse in Fig. 1.4, the reciprocal of the pulse duration actually approximates the 4 dB bandwidth. In this way, the reciprocal of the duration of an unmodulated is considered to be the bandwidth of the pulse. Consider this in the context of range resolution in (1.14) where a shorter pulse results in better range resolution. Given the relationship between, the unmodulated pulse's duration and bandwidth we can additionally say that a larger bandwidth corresponds a better range resolution. It turns out that this is generally true for any waveform unmodulated or not. Thus, by increasing the bandwidth of the signal through modulation, a radar signal can be made to achieve a better range resolution than an unmodulated pulse of the same duration.

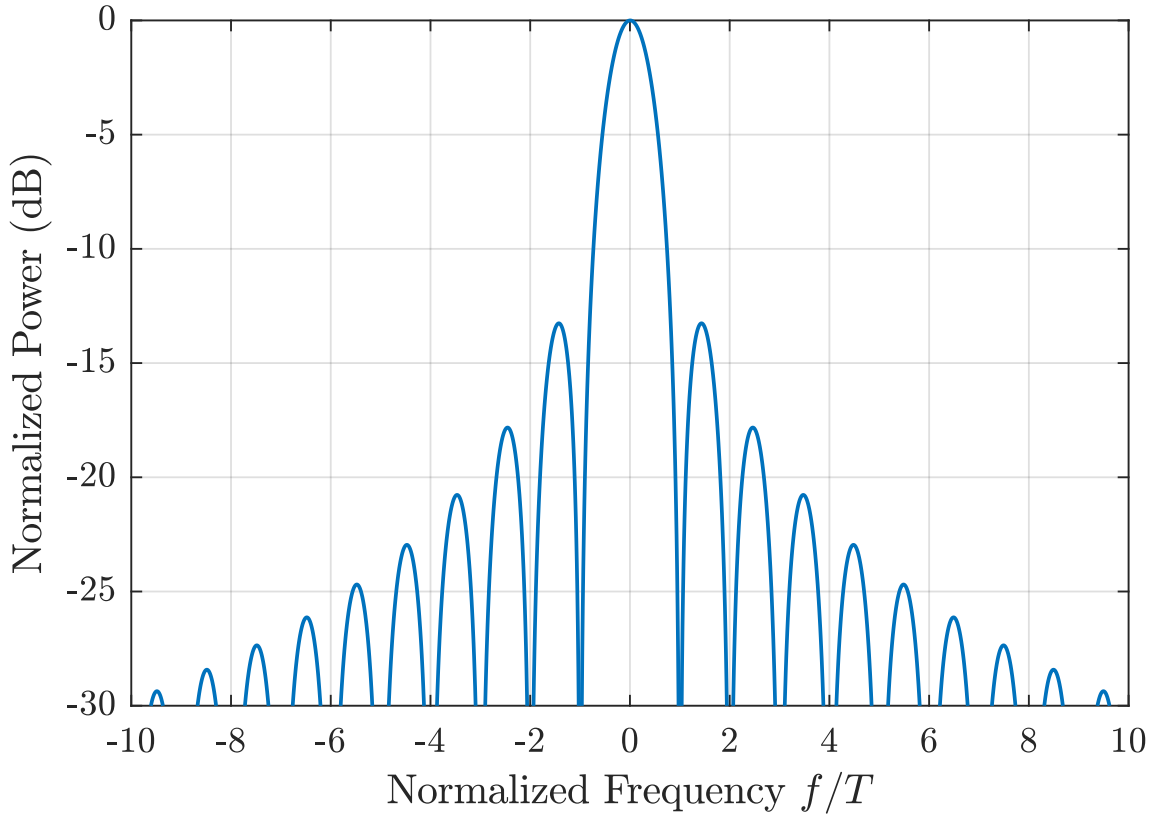

Figure 1.4: The power spectrum of an unmodulated pulse (tone) centered at baseband

1.5.2 The Matched Filter

To begin the discussion of the matched filter it is necessary to establish how radars receive and process data. This can either be done through analog processes or more likely through digital processes as is the case for modern radar systems. The goal is then to manipulate the received data such that the performance of the radar system is maximized in some sense. In general, especially as the radar problem has been presented thus far, this means maximizing the probability of detection. The question then arises. What filter can be applied to the received data such that the probability of detection is maximized? In the context of (1.12), the radar range equation, this means maximizing the SNR. It can be shown, as has been done in numerous works [3–6], the filter which maximizes

SNR in the presence of additive white Gaussian noise (as in (1.12)) takes the form

$$h(t) = as^*(-t) \quad 0 \leq t \leq T \tag{1.20}$$

where $s^*(-t)$ is the time reversed, complex conjugated version of the transmit signal and a is an arbitrary scale factor. Naturally, this is known as the matched filter since the filter is "matched" to the transmit signal. Filtering with the matched filter results in the autocorrelation response since the signal is correlated with itself. To demonstrate how this filter improves detectability consider the received signal in the vicinity of an echo in the presence of additive, white Gaussian noise. The filter is then applied as a convolution

$$y(t) = \int_{t_d}^{t_d+T} (as^*(t-\tau))(bs(t) + \nu(t))dt \tag{1.21}$$

where b is some scale factor corresponding to the channel loss of the scene and $\nu(t)$ is the additive, white Gaussian noise. Now examine (1.21) when $\tau = 0$ where the filter precisely overlaps the received echo. It becomes

$$y(t) = a\int_{t_d}^{t_d+T} b|s(t)|^2 + s^*(t)\nu(t)dt \tag{1.22}$$

According to the Schwartz inequality we can write from [1]

$$\int_a^b m_1(x)m_2(x)dx \leq \int_a^b m_1^2(x)dx \int_a^b m_2^2(x)dx \tag{1.23}$$

where $m_1(x)$ and $m_2(x)$ are arbitrary functions and the inequality holds with equality iff $m_1(x) = cm_2(x)$ where c is some constant. In (1.22) this is precisely the case. Thus, the signal component of the received data reaches is maximum value iff the matched filter is used. Since the noise is additive white Gaussian, its component will simply be proportional to the noise variance and the filter energy. Since the filter energy scales the signal component as well as this term, it cancels out

in the SNR equation. Thus the matched filter maximizes the SNR in the presence additive white Guassian noise. The proof presented here is not rigorous, but it is instructive for understanding the matched filter. A more rigourous treatment of the matched filter can be found in [3–6]

In the previous section, it was stated that range resolution is proportional to the signal bandwidth as was shown for the case of an unmodulated pulse. However, this is also true generally for any modulated pulse as will now be shown via the application of the matched filter. To begin this discussion consider a pulse whose instantaneous frequency is a linear function of time. Pulses of this type are said to be linearly modulated and are known as LFMs which stands for linear frequency modulation. Mathematically at baseband, these pulses are defined as [1]

$$s(t) = A\exp\left(j\pi\frac{B}{T}t^2\right) \quad -\frac{T}{2} \leq t \leq \frac{T}{2} \tag{1.24}$$

Over the course of the pulse, the instantaneous frequency linearly transitions from some f_1 to f_2 at a rate of $K = B/T$. If this were a sound wave this transition in frequency would sound like a chirp thus the value K is often referred to as the "chirp" rate and the pulses more generally as "chirped" LFM. The difference between f_1 and f_2 is approximately the 3 dB bandwidth of the LFM, B. The Fourier transform of an LFM pulse can be analytically calculated, however the exact solution is unwieldy as it contains Fresnel integrals [7]. The spectrum can be well approximated as [1]

$$S(f) = A\exp\left(-j\frac{T\pi f^2}{B}\right)\exp\left(j\frac{\pi}{4}\right) \quad -\frac{B}{2} \leq f \leq \frac{B}{2} \tag{1.25}$$

so long as the time-bandwidth product (BT) of the signal, that is the product of the pulse duration and the 3dB bandwith, is large (i.e. $BT > 10$).

To demonstrate how this modulation affects the signals bandwidth consider Fig. 1.5 where the spectrum of an LFM pulse is plotted against that of an unmodulated pulse in normalized terms.

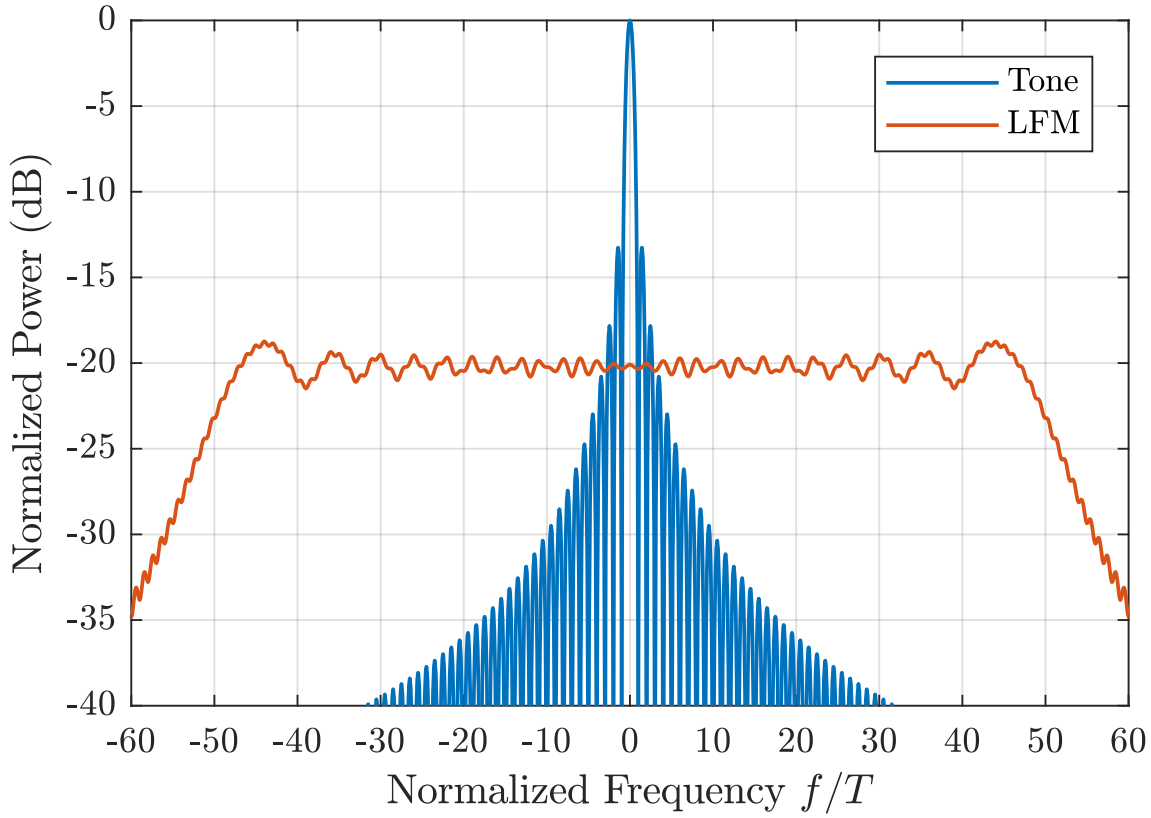

Figure 1.5: The power spectrum of an unmodulated pulse (tone) compared againts that of an LFM pulse

In Fig. 1.5, the LFM has been modulated such that it has about 100 times the bandwidth of an unmodulated pulse of the same duration and energy. The discrepancy in the normalized power between each pulse is that the power of the LFM is spread out over about 100 times the bandwidth of the unmodulated pulse. Thus, since each pulse has the same amount of power, the peak passband power of the LFM is about $10\log_{10}(100) = -20$ dB below the peak unmodulated pulse spectral power. To demonstrate how this increase in bandwidth relative to the pulse duration results in an improvement in the range resolution, Fig. 1.6 plots the matched filter responses for each case.

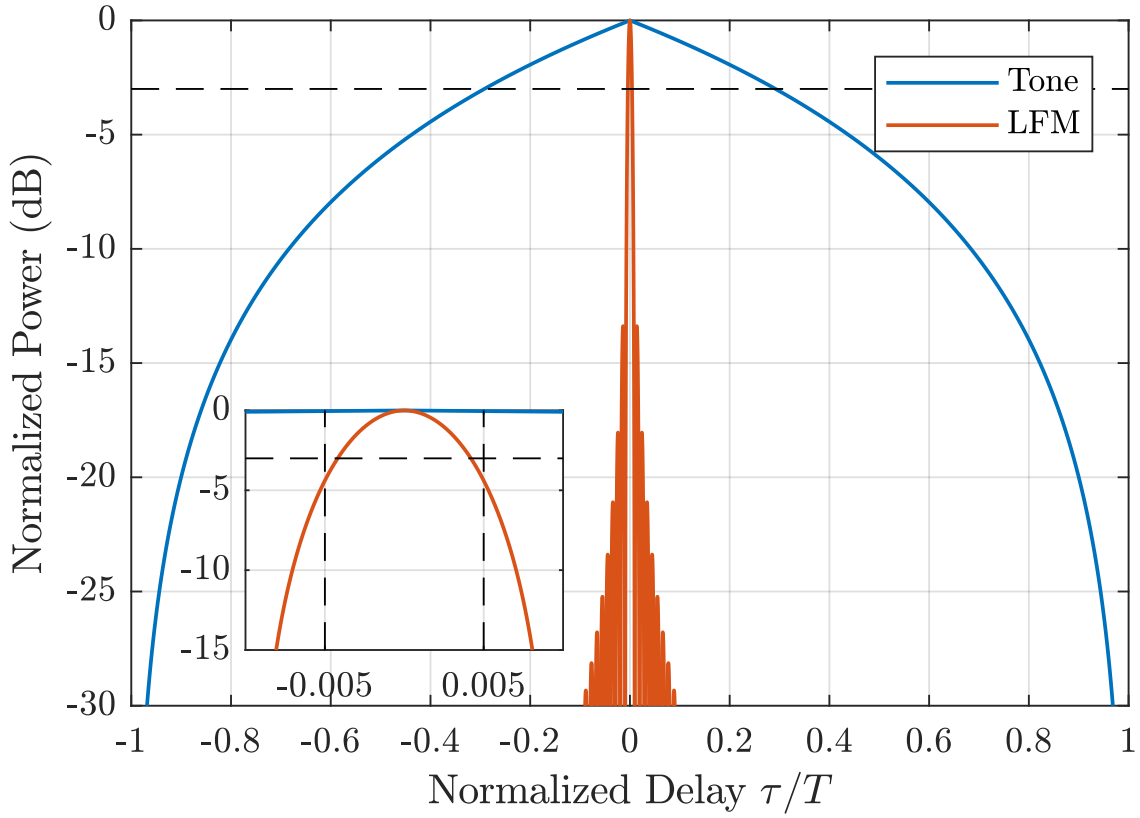

Figure 1.6: The matched filter responses of an unmodulated pulse and an LFM pulse

The matched filter responses of the unmodulated pulse and the LFM pulse are markedly different. The unmodulated match filter response consists of a single lobe extending over the entire $2T$ length filter response. The LFM response however consists of a lobing structure where there is a "mainlobe" centered at $\tau = 0$, and a series of smaller lobes, known as "sidelobes". As will be discussed later these sidelobes have important impacts on detection as they can obscure smaller power responses. Additionally, the response of the LFM demonstrates why matched filtering is also referred to as pulse compression. The pulse energy, which was spread over the entire duration of the pulse is now focused (mostly) within a narrow interval.

In the context of range resolution, consider the widths of the mainlobes. For the unmodulated pulse, this is the entire response. If the range resolution is taken to be the 3 dB bandwidth of the

mainlobe, then the unmodulated case corresponds to a time width of $T/2$ which lines up nicely with the range resolution definition given in (1.14). For the LFM case, an inset is provided which zooms in on the mainlobe response of the LFM. In the inset it is clear that the 3 dB point occurs at a value slightly less than $|\tau/T| = .005$. For LFMs the resolution of interest is often the *Rayleigh* resolution, or the null to null width of the mainlobe. In the normalized case in Fig. 1.6 this extends from $[-.01 < \tau/T < .01]$. Thus the Rayleigh resolution is approximately

$$\Delta R \approx \frac{c}{2B} \tag{1.26}$$

where for an LFM $B \approx \Delta f$ which is the change in frequency over the duration of the pulse. In general, the topic of range resolution is very complex. For the unmodulated case and the LFM case in isolation, as shown above, it is relatively straightforward. In crowded environments where there are any number of distinct returns or non-LFM like pulses are used the concept of resolution in the context of detection becomes much more complicated, but such a discussion on this topic is beyond the scope of this work. An in depth discussion of detection can be found in numerous radar textbooks such as [8–10]. For this work, the important takeaway is that range resolution is generally improved with increasing signal bandwidth and that this improvement, along with the maximization of SNR in the presence of additive white gaussian noise, motivates the use of the matched filter.

1.5.3 Autocorrelation Sidelobes

Figure 1.7 shows the matched filter (autocorrelation) response shown in Fig. 1.6 in isolation and zoomed in on the mainlobe response and several adjacent sidelobes. Regardless of BT, the LFM autocorrelation response always results in a peak sidelobe level (PSL) of -13.2 dB as shown in Fig. 1.7. In isolation, this is not an issue. Clearly, the true target response is indicated by the mainlobe. However, what if there is more than one target present? What if there are two close by and one is significantly smaller than the other in a returned power sense? As discussed earlier, it is not

unusual and in fact it is typical, that the power returned by targets in a radar scene will span several orders of magnitude. Consequently, it would not be at all unusual for there to be a relevant return buried in the sidelobes of a more powerful scatterer.

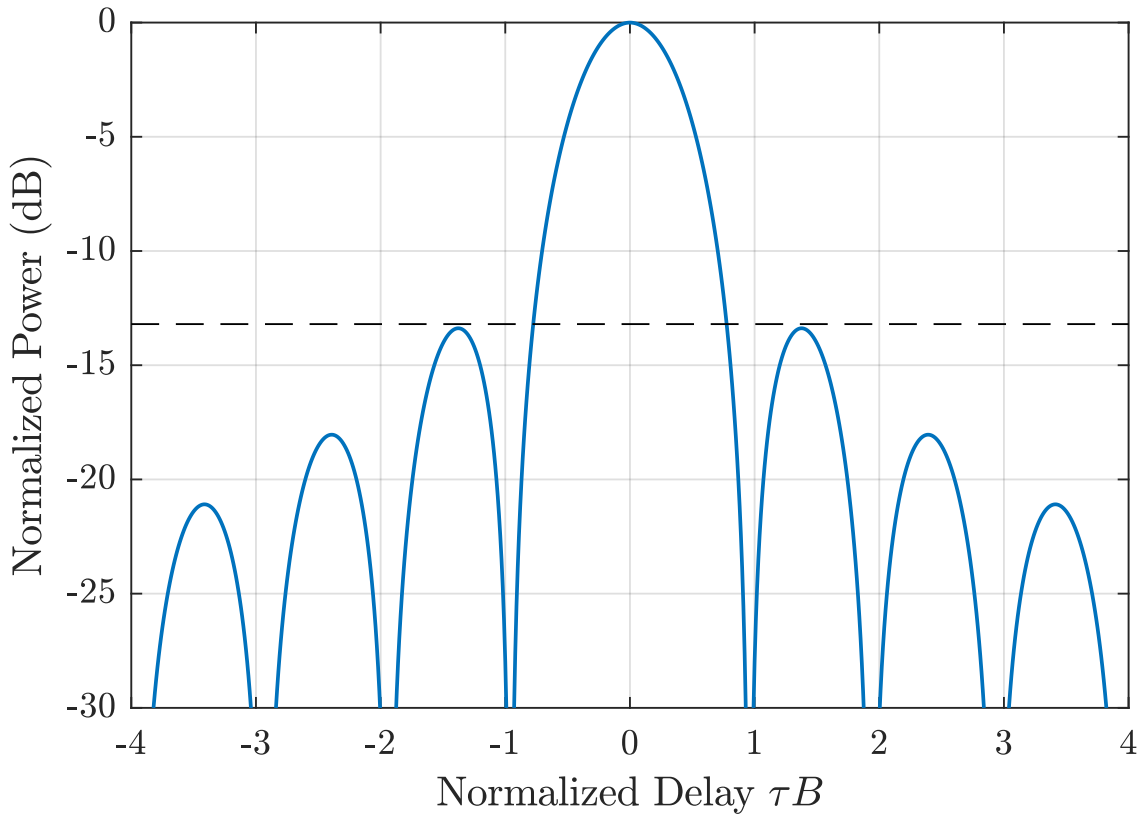

Figure 1.7: The autocorrelation response of an LFM pulse with $BT = 100$ near the autocorrelation mainlobe

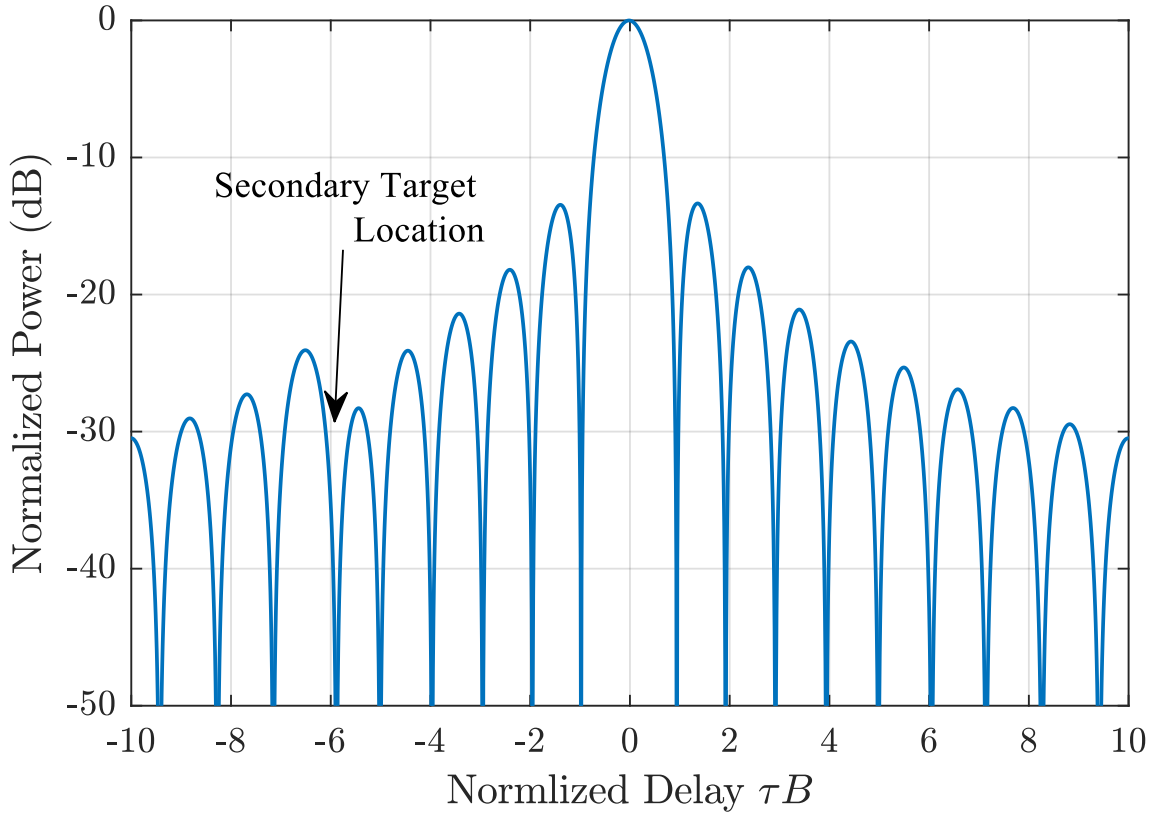

Figure 1.8: The secondary target response is completely obscured by the sidelobes of the much stronger scatterer

In Fig. 1.8, the autocorrelation response of the large target centered at 0 normalized delay completely obscures the location of a second weaker target. In fact, the only indication that anything else is anywhere is a small perturbation in the typical sidelobe response of an LFM. As shown in Fig. 1.8, there is no obvious way to pick out the location of a second target. The most obvious way to accommodate this issue is to use waveforms with lower autocorrelation sidelobes. Additionally, over the years various other methods have been developed to mitigate this issue. These methods include mismatch filtering schemes, where instead of a matched filter, a different filter is used which has been optimized to reduce the sidelobe levels at the expense of a loss in SNR [11, 12]. Sequential estimation schemes remove the responses of large scatterers to reveal the lower power targets [13]. Adaptive filtering schemes estimate the range profile and minimize sidelobe levels

in an iterative manner [14]. One method of sidelobe minimization which has proven effective for LFMs in particular is tapering where either the spectrum of the received data or the time envelope of the match filter are tapered by any one of a number of "window" functions. An in depth discussion on windowing functions can be found in [15, 16] , but to quickly demonstrate the effectiveness of this approach, Fig. 1.9 shows how applying a Hanning window significantly reduces the sidelobe response of the larger scatterer thus revealing the secondary target. The cost of doing so is a significant loss in SNR as indicated by the power level of the large scatterer which is at about -7 dB relative to the matched filter. Additionally, the mainlobe response has been somewhat widened indicating a modest loss in range resolution. Despite these drawbacks, tapering is frequently used in practice due to the relatively high sidelobe level of LFM waveforms.

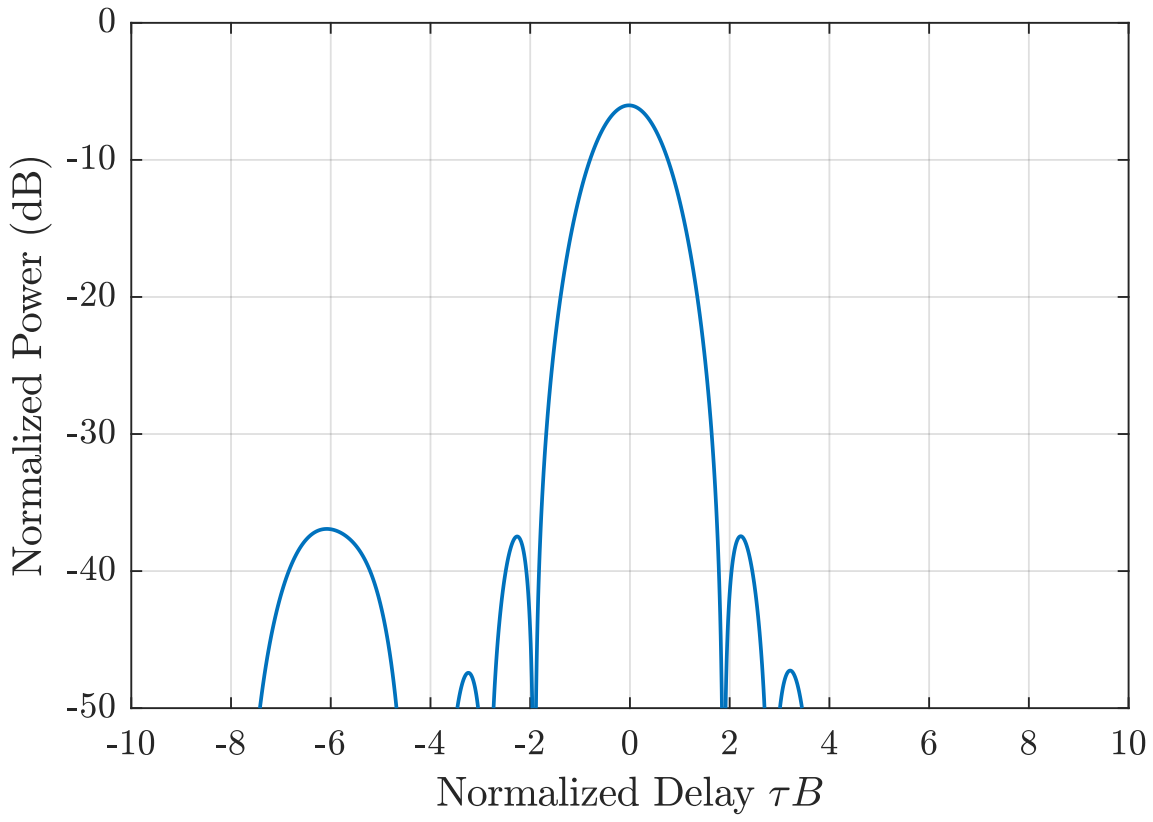

Figure 1.9: The secondary target response is revealed when the scene is filtered with a tapered matched filter

The sidelobes of the large target can be considered as a form of interference for the smaller secondary target. With this in mind we can define a new form of the radar range equation which measures not just the impact of noise, but also the interference due to nearby scatterers. Doing so brings about

$$\text{SINR} = \frac{P_S}{(1/n_p)P_N + \sum_{m=1}^{M} I_m} \tag{1.27}$$

where SINR is the signal to interference plus noise ratio, for succinctness the signal power has replaced with P_s and the noise power has been replaced by P_N and I_m is the sidelobe interference at the range of interest due to the m_{th} nearby scatterer (relative to the range of interest). In the case of Fig. 1.8, there was only one such interference term but it was enough to completely obscure the second target. Critically, the pulse integration term n_p has been inverted and moved into the denominator with the noise term where it can only mitigate the noise power. This is simply due to the fact that the noise power adds incoherently while the signal power, including the sidelobes, add coherently across multiple pulses when the same pulse is used every PRI as is assumed in (1.27).

1.5.4 Noise Radar

The transmission of the same pulse, or in the case of CW, some periodic waveform introduces ambiguities in the measurement. As discussed in Section 1.4, this notably occurs in ranging. As a means to address this issue, B. M. Horton suggested transmitting a signal modulated with random noise [17]. In Fig. 1.3, if the first pulse is different from the second pulse, the ambiguity would be resolved. Horton originally suggested this in the context of CW ranging measuring systems such as altimeters, but since then noise radar concepts have be considered for various purposes. [18] provides an overview of such applications.

In addition to mitigating ambiguities, noise radar operation provides an additional advantage in terms of sidelobe mitigation. Consider again (1.27) where the sidelobe responses of nearby scatterers interfere with a given range and pulse integration does not help here since the sidelobes add coherently. In the previous section several methods were introduced which have shown to be

effective, but here we consider another method wherein the coherent combination of the sidelobes of different waveforms results in a reduction of the sidelobe levels. This can be done explicitly as in the case of complementary codes where the sidelobe responses of different codes have been explicitly designed to completely cancel each other out [13, 19, 20]. Alternatively, in the case of noise radar, the cancellation occurs incidentally.

Pulsed noise radar is a fundamentally pulse agile scheme where each pulse is uncorrelated with every other pulse such that they exhibit unique, uncorrelated sidelobes. At a fundamental level, this allows the sidelobe samples of the different waveforms to be treated as iid random variables meaning the samples add together like the noise samples discussed in Section 1.3. In this mode, (1.29) can be modified as

$$\text{SINR} = \frac{n_p P_S}{P_N + \sum_{m=1}^{M} I_m} \tag{1.28}$$

The simplest approach to realizing this affect is to transmit noise as the waveform. In practice, the highly variable pulse amplitude of a waveform created this way leads to implementation issues as will be discussed in Section 1.6.1. In this work however, FM noise waveforms are considered. In contrast to simply transmitting random noise, FM noise waveforms are constant amplitude and can be designed to have good spectral containment and autocorrelation sidelobes as is done in Section 1.6.2.3 and is shown extensively in the results of Chapter 2.

1.6 Radar Signals

The only radar signals discussed in much detail so far have been the unmodulated pulse and the LFM chirp, but there are a multitude of different signals and more generally classes of signals that have been developed. In general, radar signals are designed to improve the detectability of objects in a radar scene. This is true for Synthetic Aperture Radar (SAR) where the goal is to form full fledged images of scenes which are comparable to optical images, for an airport radar where the goal is to know where aircraft are and where they are going, or for an automotive radar which just needs to know where the next car is and how fast it is approaching. What these cases

have in common is that there are objects to be detected and in general they are moving. In the previous section, we saw how autocorrelation sidelobes have an deleterious effect on the ability to detect low power objects in the presence of higher power ones, but the fact that the objects of interest are often moving further complicates this goal. Without going to far into the details, the motion of an object induces a frequency shift in the transmit signal that is proportional to the radial velocity of the object relative to the transmitter. This is known as a Doppler shift and it has important implications with respect to matched filter. Essentially, a Doppler shifted pulse is a fundamentally different pulse than its unshifted counter part. Unless the matched filter is adjusted to compensate for this affect (i.e. tuned to detect objects at a certain velocity), there will be some degree of mismatch loss. Some signals, such as the LFM, are particularly robust to this effect and are called "Doppler tolerant". Other waveforms, such as the ones designed in this work, are very sensitive to Doppler. In fact, if the autocorrelation response of these waveforms is plotted as a function of Doppler shift, these waveforms exhibit a "thumbtack" like response where they really only cohere with the matched filter at zero Doppler [6]. As a topic, Doppler shifts are very important to the radar problem as they can aid in detection and allow for the measuring of target velocity. Nevertheless, a detailed discussion into the implications of Doppler shifts is beyond the scope of this work, as here we are primarily concerned with the spectral, autocorrelation, and implementation characteristics of radar signals.

As final note before discussing specific radar signals and signal models, the scope of radar signals design is broad and only as much information on the topic is included here as is necessary to understand the motivation and implementation of the work in Chapter 2. There are numerous works which provide excellent overviews of radar signals such as [1, 6, 21]

1.6.1 Practical Considerations

In Section 1.5, signal bandwidth was discussed in the context of range resolution, but here it is examined in the context of practical implementation. Consider a radar transmitter. At each step in the physical implementation of the signal the bandwidth is constrained. In a digital system this

is limited by the sample rate. In the analog parts of the process the wiring, circuit design, up conversion, amplification and the antenna itself will all impose some amount of spectral filtering. Accordingly, it is important to design radar signals such that they stay within these spectral limits. Otherwise, energy will be dissipated and converted into heat within the transmitter. If this occurs after amplification the dissipated energy can be significant and it may damage the radar system. This problem occurs in addition to the distortion imposed on the signal by band limiting it below its designed bandwidth.

However, even if the transmitter bandwidth is not a concern, there can be problems elsewhere simply because the electromagnetic spectrum is a finite resource. To comply with regulations put forth by the FCC and to, in general, be a good spectral neighbor it is necessary to ensure any radar signal stays within its own band as much as possible [22]. Typically, this concerns what happens beyond the 3 dB bandwidth in what is known as the "roll-off" region. Ideally, all signals would exhibit a "brick wall" spectrum where the spectral content drops to zero beyond a certain frequency. However, just from basic Fourier theory this is impossible. As discussed earlier any finite length signal will have an infinite absolute bandwidth, so the key is to ensure the signal power decays (rolls off) as quickly as possible beyond the operational bandwidth. As the electromagnetic spectrum becomes further congested with the proliferation of connected, bandwidth hungry devices such as cell phones with video streaming capabilities spectral containment will become more and more important for radars and communications systems to coexist. In this work, we consider a more extreme case where there is an interferer within the radars operational bandwidth. This too can be addressed through appropriate radar signals design. Functionally speaking, interference due to other spectral users has severe consequences for the effectiveness of a radar system. In Fig. 1.10, the LFM autocorrelation is severely distorted by strong (10x the LFM energy) wideband interference to the point it looks like there are multiple scatterers at perhaps $\tau B = -4.5, 0, 3.5$ when in reality the only true return is located at $\tau B = 0$. Now the interference here is fairly strong but this is not an unlikely scenario. Consider the R^4 two-way spreading loss experienced the radar problem. Another transmitter interfering with the radar would only experience an R^2 one-way

spherical spreading loss so it would not be unusual for the interferer to exhibit a high power level relative to the desired returned signal. In terms of the radar range equation, this can be accounted for by adding an additional interference term.

$$\text{SINR} = \frac{P_t T f_{PRF} G^2 \lambda^2 \sigma}{(1/n_p)(4\pi)^3 R^4 L k T_0 F B + I_u + \sum_{m=1}^{M} I_m} \tag{1.29}$$

where I_u is the interference induced by other users. In general characterizing the behavior of I_u is a difficult problem as it will depend on numerous factors such as the modulation scheme employed which impacts statistics of the interference and consequently how it will cohere the matched filter at a given range. As a first approximation however, the amount of interference induced by another user will be in some way proportional to the cross spectral content as defined by Weiner-Khintchine theory. Thus the approach taken in this work to minimize interference from other users is to minimize the cross spectral content.

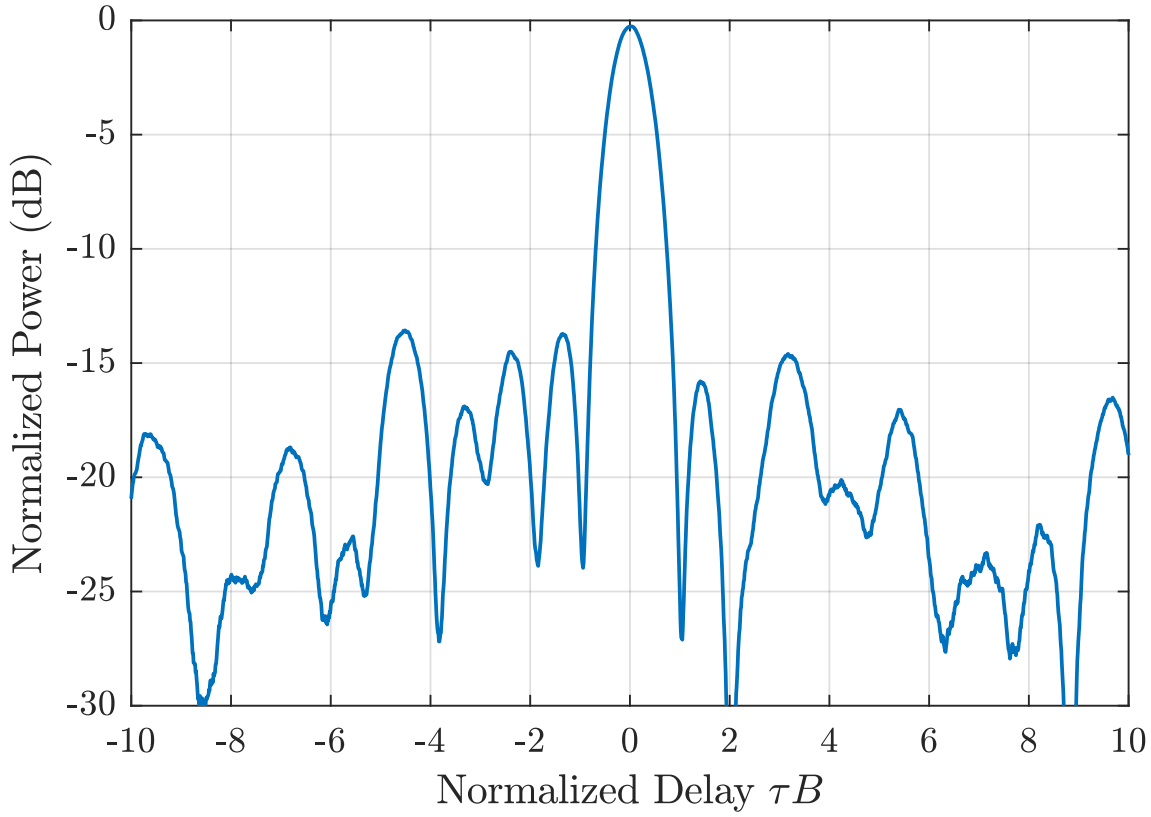

Figure 1.10: The Presence of strong, wideband interference distorts the LFM autocorrelation

More specific to radar or at least any system which utilizes high powered amplifiers, it is important to avoid the use of amplitude modulation (AM). For the sake of efficiency or simply because of the constraints of the component, high powered amplifiers tend to operate in the saturation region where they are most efficient [23]. Consequently, a signal with AM will invariably be distorted. As opposed to the filtering distortion discussed early, this distortion is non-linear in nature and can have difficult to characterize effects on the transmit signal [24]. One such affect is spectral regrowth where the roll off of the transmit signal is degraded leading to poor spectral containment. On the receive side this is problematic as this distortion will lead to degraded autocorrelation performance and thus poorer operational effectiveness overall.

1.6.2 Signal Models

The approach to radar signals design taken here is to progress from the most general signal model to more specific, tractable ones. With this in mind consider the most general mathematical model of a pulsed signal possible.

$$s(t) = \begin{cases} s_p(t) & 0 < t < T \\ 0 & \text{otherwise} \end{cases} \tag{1.30}$$

The radar signal $s(t)$ has some non-zero content $s_p(t)$ during the pulse interval $0 < t < T$. Otherwise the signal is 0. From a design standpoint, this model is far too general and thus impractical as there is an infinite continuum of possible signals. To pare this down we can consider what makes a radar signal "good" such as the autocorrelation characteristics or the practical considerations discussed above. Historically, and presently for that matter, the discretization imposed by digital systems has strongly influenced the models used to design radar systems. With this in mind, the signal model discussion begins with radar codes.

1.6.2.1 Radar Phase Codes

The discretization of a baseband continuous time signal results in a sequence of complex values. Considering modern systems are digital, this is the domain where the processing occurs. Thus, it is reasonable to design radar signals from this standpoint. Additionally, this pares down the signal model in (1.30) from a contiuum of possibilities to the discrete representation

$$s[n] = \begin{cases} s_p[n] & n = 1, 2, \cdots, N \\ 0 & \text{otherwise} \end{cases} \tag{1.31}$$

Even more succinctly, this can be written in vector form as

$$\mathbf{s} = [s_1 \; s_2 \; \cdots \; s_N]^T \tag{1.32}$$

where the 0 valued portions of $s[n]$ have been omitted for conciseness.

Over the decades numerous contributions have been made to the set of radar codes. First consider biphase, or codes whose elements are allowed to take on 2 antipodal states (i.e. π and $-\pi$). Among these are the well known Barker codes [25], minimum-peak-sidelobe (MPS) codes [26], maximal length sequences (MLS) [27], and others. Beyond biphase codes are polyphase codes where the elements are allowed to take any value within a set greater than two up to a continuum of values. Given their greater degree of design freedom there are a large number of polyphase codes. To name a few there are polyphase Barker codes [28], Frank codes [29], P-codes [30, 31] and many others. A detailed survey radar pulse compression codes can be found in [6].

The most common goal in designing these codes is the minimization of autocorrelation sidelobes with secondary considerations such as Doppler. For example consider the autocorrelation of the longest known ($N_b = 13$) Barker code in Fig. 1.11. Barker codes are optimal in the sense that they achieve a peak autocorrelation level of $1/N_b$ where N_b is the length of the Barker code.

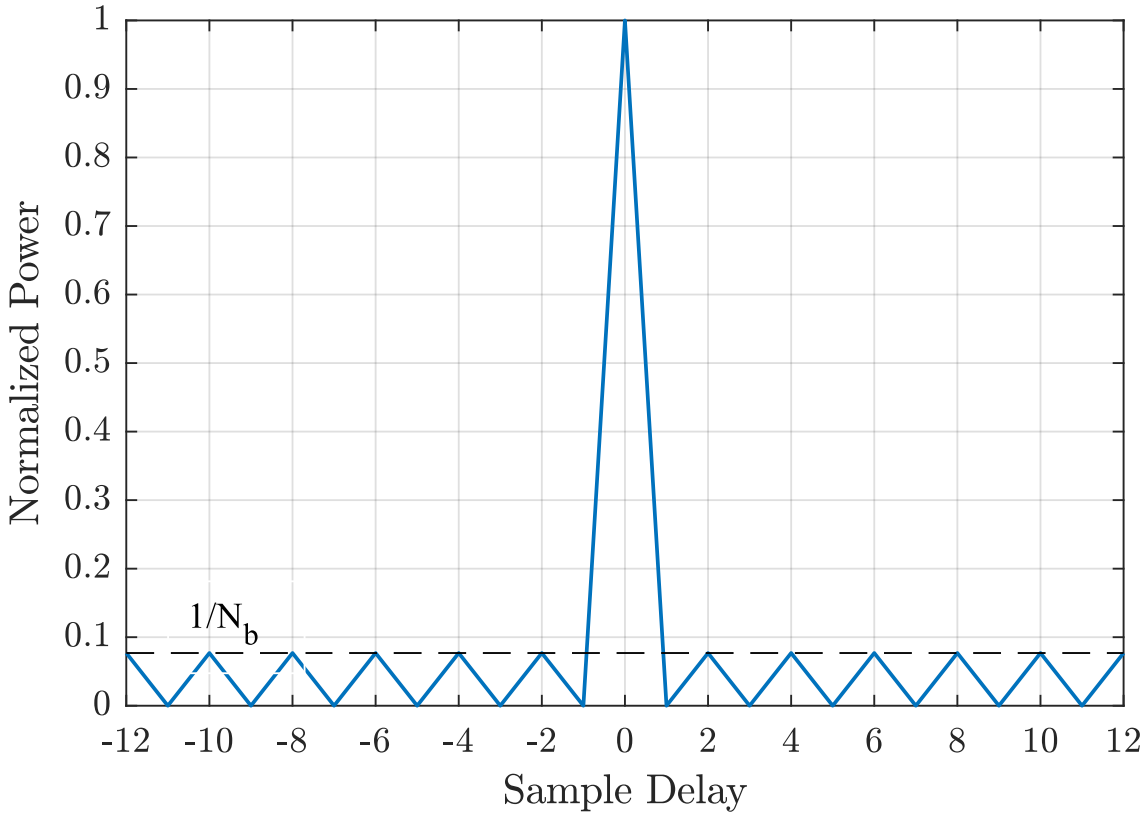

Figure 1.11: The autocorrelation of an $N_b = 13$ length Barker code

Now consider codes in the context of the practical considerations discussed earlier. They need to exhibit good spectral roll-off beyond some nominal transmit bandwidth and they need to be constant amplitude. However, to examine these properties it is necessary to model the continuous time signal, the signal that will be implemented on the hardware. (1.32) and (1.31) are precisely sampled versions of (1.30), but it is not clear how best to recover (1.30) from the sampled signal model (1.31). An intuitive approach is to model the continuous time signal as the concatenation of N, T_p second long, constant amplitude segments or "chips" such that the total pulse length is $T = NT_p$. Figure 1.12 shows this model and it is indeed constant modulus satisfying that condition as discussed earlier.

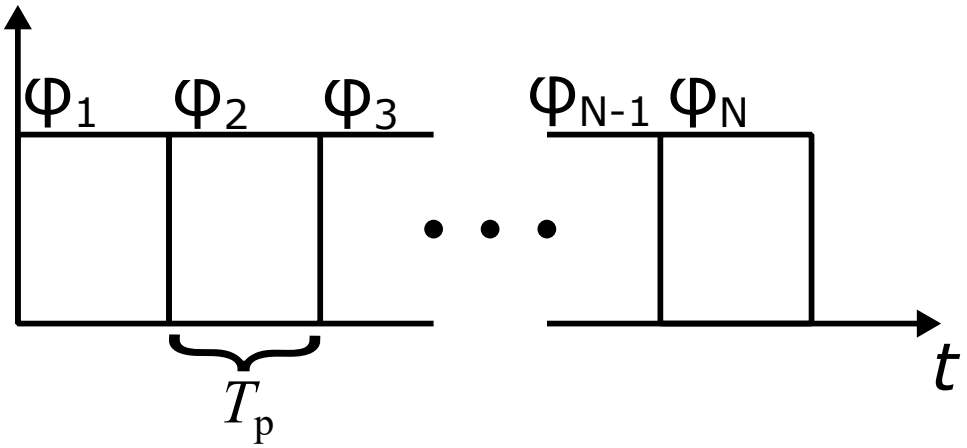

Figure 1.12: Continuous time phase coded waveform model

The next step however is to examine spectral containment. To do so, we write the phase coded model in an analytical, continuous time fashion which preserves its autocorrelation characteristics as

$$s(t;\mathbf{x}) = A\sum_{n=1}^{N} \text{rect}\left(\frac{t - T_p(n-1/2)}{T_p}\right)\exp(j\phi_n) \tag{1.33}$$

where $\mathbf{x}$ is the N-length vector of phase values and

$$\text{rect}(x) = \begin{cases} 1 & -1/2 < x < 1/2 \\ 0 & \text{otherwise} \end{cases} \tag{1.34}$$

Then, by recognizing (1.33) is a superposition of rectangular functions and the Fourier transform is a linear operator, the analytical Fourier transform of (1.33) is

$$S(f;\mathbf{x}) = \sum_{n=1}^{N} \frac{\sin(\pi f T_p)}{\pi f}\exp(j(\phi_n - 2\pi f(n-1/2)T_p)) \tag{1.35}$$

Since the Fourier transform of a rectangular function is a sinc function, it is unsurprising that the Fourier transform of the phase code signal model in (1.33) is a summation of phase shifted sinc functions. Using the Barker code above as an example it is common to consider the 3 dB bandwidth as the nominal bandwidth of a signal. In Fig. 1.13, the spectrum of the 13-length Barker code is plotted to 5 times its nominal 3 dB bandwidth (B_{3dB}) (in this case approx. 4 dB) according to

(1.35).

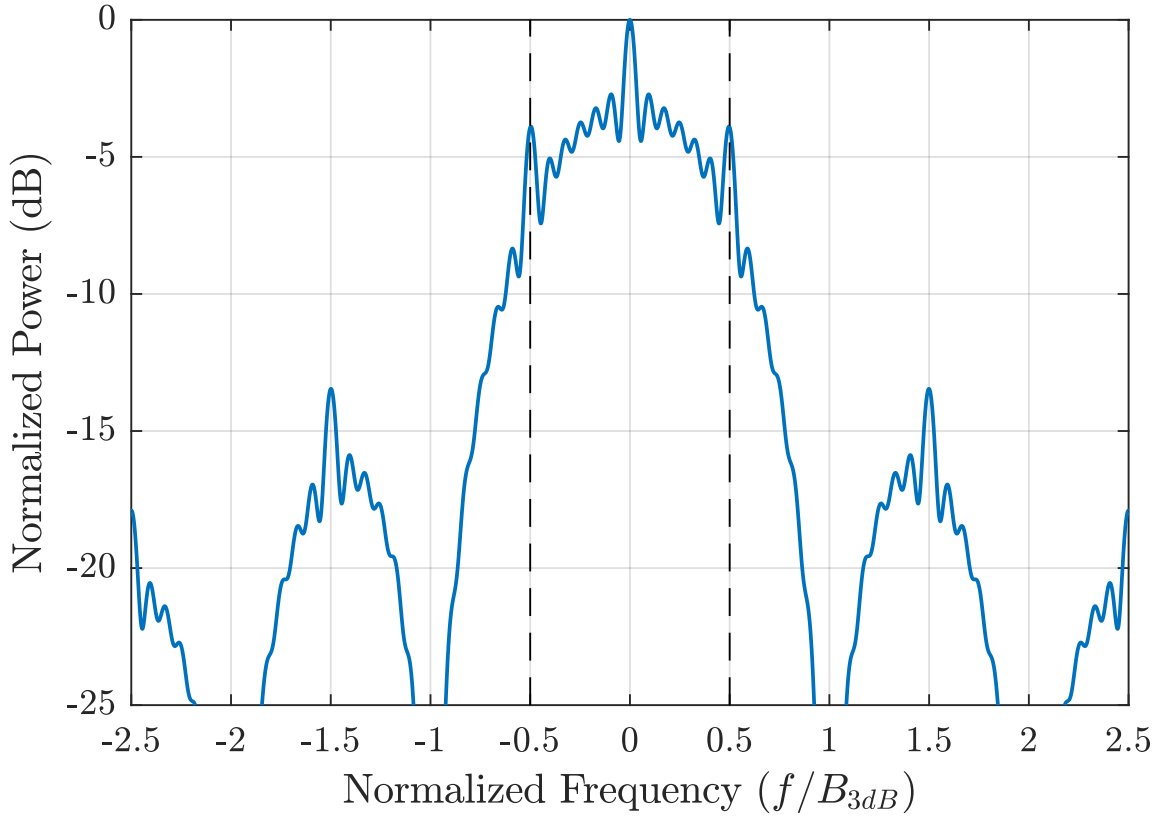

Figure 1.13: Power spectrum of the length 13 Barker code according to the analytical phase code spectrum

Interestingly, the spectrum rolls off precipitously just beyond the 3 dB bandwidth, as indicated by the vertical dashed lines, to a null at $|f/B_{3dB}| = 1$. However strong peaks then occur at odd integer multiples of B_{3dB}. These image like peaks will inevitably be filtered out somewhere in the transmit chain leading to signal distortion and heat dissipation. The culprit here is the instantaneous phase changes between the consecutive chips. An instantaneous change requires infinite frequency content which results in poor spectral roll-off. To some extent, this effect can be mitigated by simply limiting the magnitude of the phase change between the chips which in effect makes the phase more "continuous". Consider this approach in the context of an LFM signal approximated as a phase code. In Fig. 1.14, two different spectra of phase coded approximations of an LFM pulse

are plotted. In the first case, the LFM was complex sampled at a rate commensurate with its B_{3dB}. Thus for a BT of 100, this results in 100 phase code values with duration T_p. In the second case, the LFM was sampled at 10 times its B_{3dB}. Thus for a BT of 100, this results in 1000 phase code values with duration $T_p/10$ relative to the previous case. The difference in the roll-off of these two cases is stark. By simply subdividing the chips from the first case, to better approximate the LFM, the roll-off has been improved dramatically. Still, peaks will appear (not shown in Fig. 1.14) even in the case with good roll-off, but they will be spaced over greater intervals and in general be lower in power leading to less distortion when implemented.

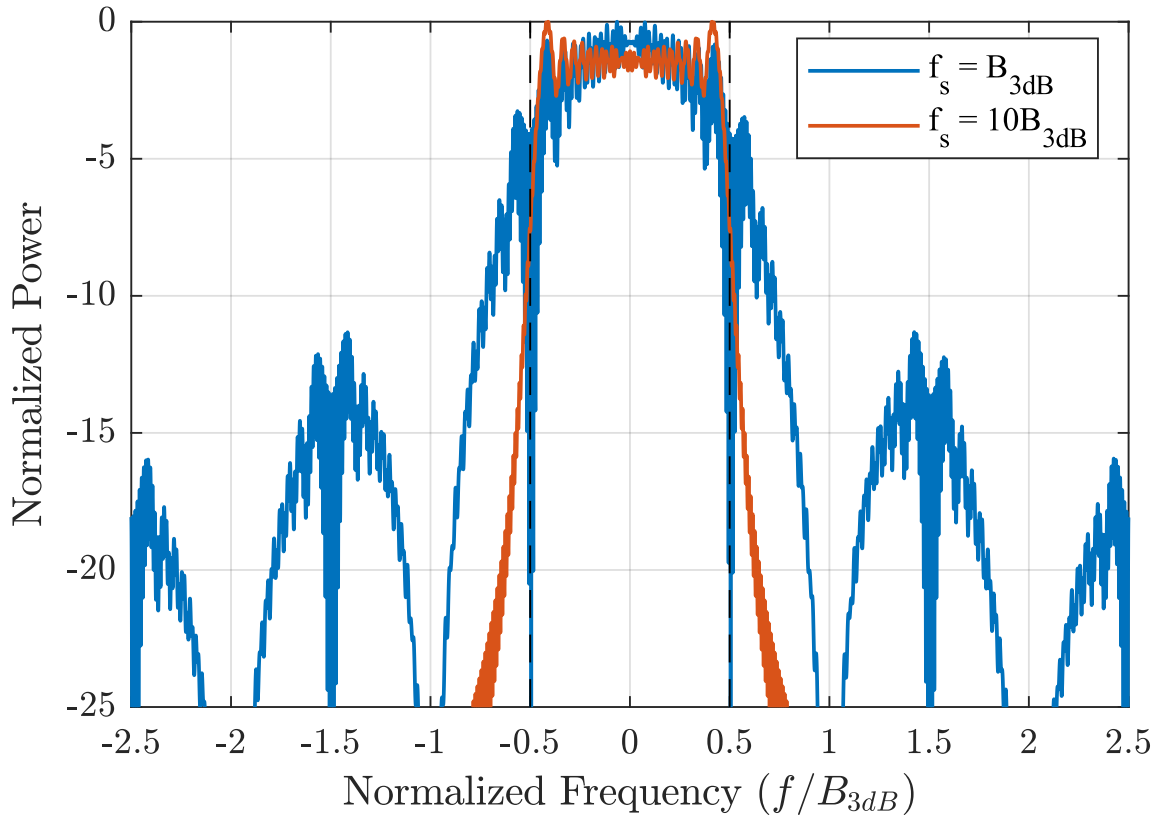

Figure 1.14: Power spectra of sampled LFM waveforms at two different sample rates implemented as phase codes

In the case of the LFM, subdividing the chips to better approximate the the LFM phase trajectory is a straightforward process as one just has to reference the continuous time phase function

for LFM waveforms. For the case of phase codes, which were designed as phase codes, there is no continuous model to reference so subdividing the chips for smoother phase transitions is less straightforward. However, to just give this a try, linearly interpolate the 13-length Barker code (in which case it is no longer a biphase code) by a factor of ten and examine how this impacts the spectrum and autocorrelation.

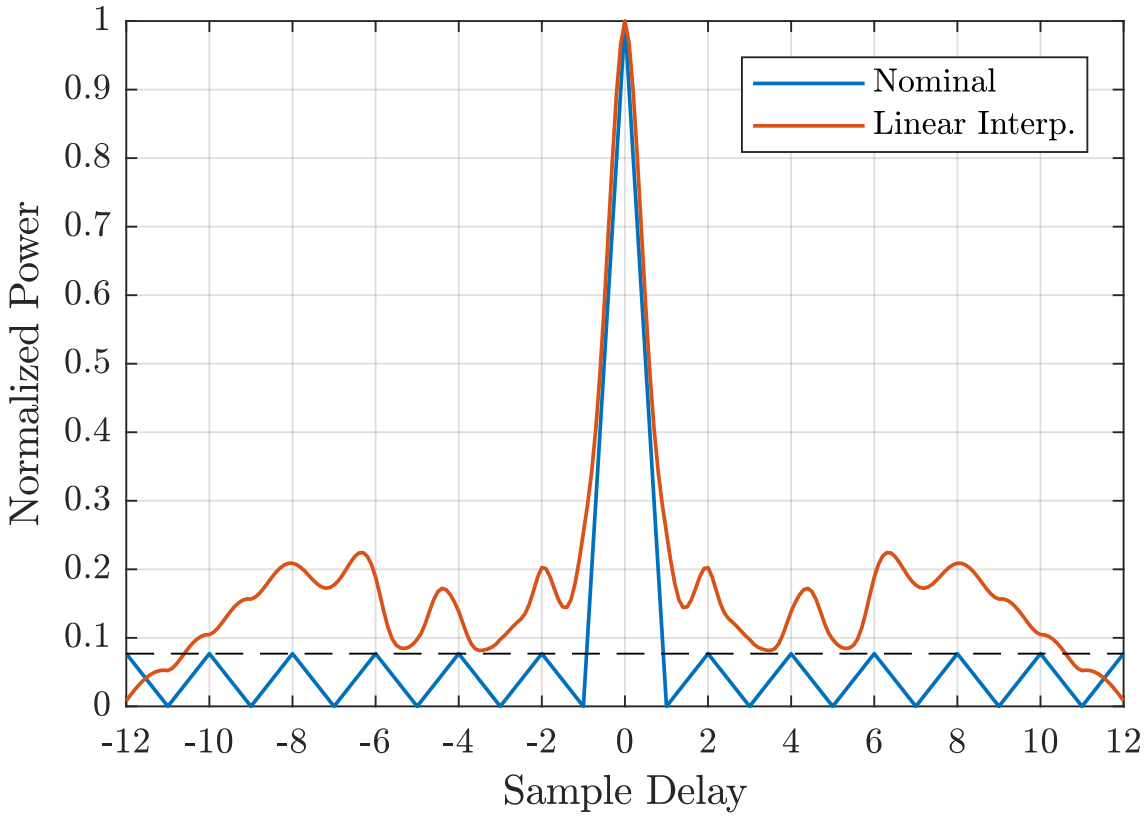

Figure 1.15: Autocorrelations of a Barker code and a 10 times linearly interpolated Barker code

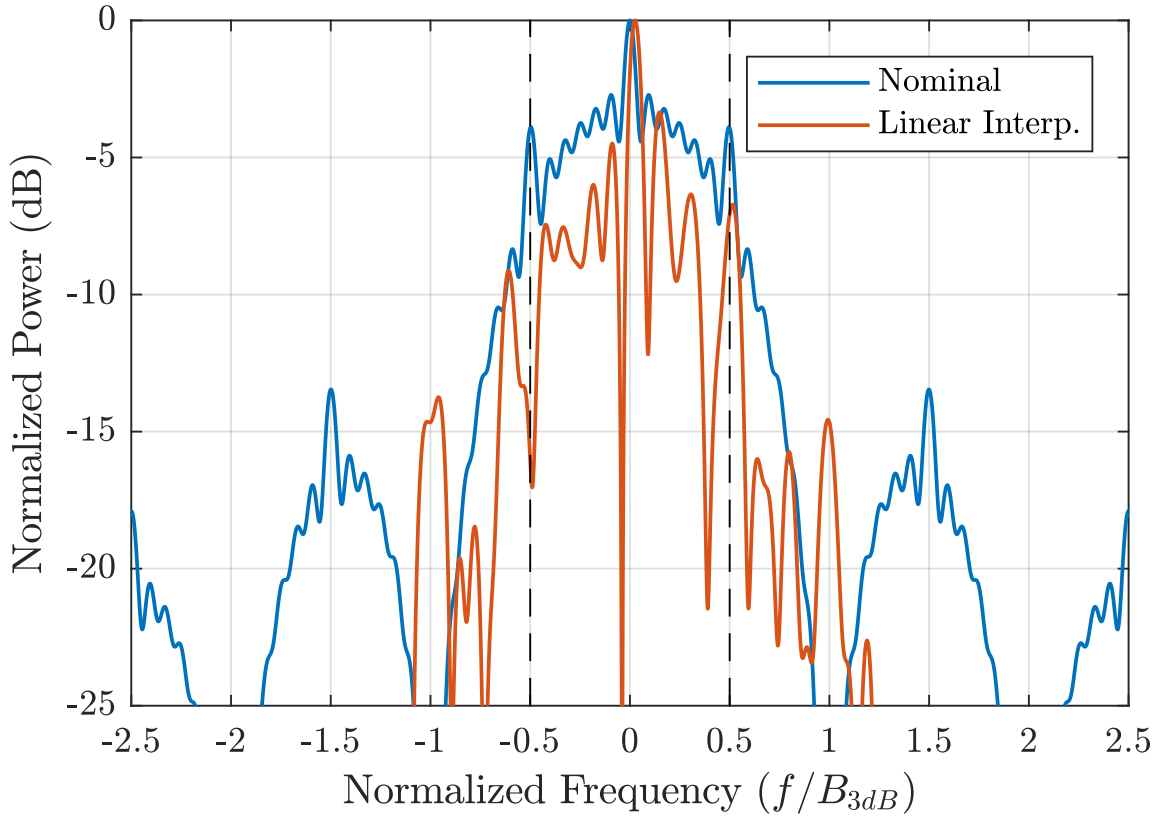

Figure 1.16: Power spectra of a Barker code and a 10 times linearly interpolated Barker code

While the spectral containment of the Barker code has certainly improved as indicated by Fig. 1.16, it is also extremely distorted and does not resemble the nominal Barker code spectrum within the 3 dB band. Additionally, the autocorrelation has also been degraded significantly relative to the $1/N_b$ performance afforded by Barker codes. These results indicate this is probably not a good way to implement this code. An alternative to interpolating the code to minimize phase transitions is to perhaps consider different shaping filters such as square root raised cosines as opposed to the square chips considered thus far since square root raised cosines exhibit better spectral roll-off characteristics as opposed to rect functions. The catch here however would be that the signal would no longer be constant modulus. As far a Barker codes are concerned and biphase codes in general, differential phase shift keying [32] and the biphase to quadriphase transformation [33]

have been shown as effective means of implementing biphase codes in a spectrally contained, continuous manner while largely preserving their autocorrelation characteristics. However, for polyphase codes other schemes have to be considered.

While phase codes often exhibit desirable characteristics such as low autocorrelation sidelobes, it can be difficult to implement them effectively due to the instantaneous phase transitions implicit in their structure. Such techniques as interpolating the phase values to mitigate the magnitude of these transitions can be effective at improving the spectral containment, but this technique will, in general, distort the characteristics which make the code desirable in first place as was seen in the case of the 13-length Barker code. Other waveform models seek to address this issue by assuming a continuous time model from the start, or by making small phase transition implicit in the waveform design. These techniques are discussed next.

1.6.2.2 Poly-Phase Coded Frequency Modulation (PCFM)

Poly-Phase Coded Frequency Modulation (PCFM) was adapted from Continuous Phase Modulation (CPM) as a means to address the physical implementation difficulties of poly-phase codes [34, 35] . CPM itself was developed as a spectrally contained and power efficient (constant amplitude) communications modulation scheme for especially power constrained applications such as the Bluetooth standard [36] where small, battery powered devices are often involved. These same attributes make CPM desirable from a radar implementation standpoint as it addresses the practical considerations discussed in Section 1.6.1. It is generally spectrally contained alleviating bandwidth concerns and its constant amplitude envelope makes it power efficient and amenable to high powered transmitters operating in the saturation region.

As the name implies, CPM achieves these goals by being a phase only modulation much like the codes discussed in Section 1.6.2.1 but the key caveat is that the phase is a continuous function of time. Consider the technique used in Section 1.6.2.1 to modify the LFM derived phase code for a more spectrally contained implementation. The phase code was broken into a greater number of shorter chips by interpolating the phase code values directly from the continuous time definition

of the LFM phase. Since CPM guarantees a continuous time phase structure, this phase code interpolation can be performed on any radar signal which has the CPM structure.

Adapting the CPM structure for use as a Radar signal design scheme is as straightforward as dropping the components of CPM which pertain specifically to communications. After all, the purpose of radar signals is not to transmit information, but to reveal information about the environment through its modulation which determines how well this task can be performed. Mathematically, PCFM is implemented as

$$s(t;\mathbf{x}) = \exp\left\{ j\left(\int_0^t g(\tau) * \left[\sum_{n=1}^{N} \alpha_n \delta(\tau - (n-1)T_p)) \right] d\tau \right) \right\} \tag{1.36}$$

were $g(\tau)$ is a shaping filter with time support over $[0, T_p]$. α_n is the n_{th} term in the PCFM code in the N length vector $\mathbf{x} = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_N]^T$. (1.36) guarantees the phase is continuous by twice integrating the delta functions modulated with the code values, $\mathbf{x}$, first by the convolution with $g(\tau)$ and secondly through the outer integral. Since all of this happens to the phase, the waveform is inherently constant modulus. The way (1.36) is constructed, it is possible to evaluate the convolution and integration as

$$\phi(t) = \sum_{n=1}^{N} b_n(t)\alpha_n \tag{1.37}$$

where

$$b_n(t) = \int_0^t g(\tau - (n-1)T_p) d\tau \tag{1.38}$$

In this way, the PCFM phase can be defined as a linear combination of weighted basis functions, where the particular form of $b_n(t)$ is dependent on the shaping filter chosen for $g(\tau)$. For the purposes of this work, and for what is known as first order PCFM [37], $g(\tau)$ will take the form of a rect function. As a result, the basis functions, $b_n(t)$ are time shifted ramp functions as shown in Fig. 1.17. Notice how every T_p seconds one basis function becomes constant while another begins to ramp. This guarantees that, when summed together, the phase is continuous and only one basis function is actively accumulating phase at a time. Additionally, the n_{th} basis function accumulates a total of α_n phase over the n_{th} T_p length interval as defined in (1.36) and more directly in (1.37).

In this way, the PCFM parameters are proportional to the instantaneous radial frequency over each T_p length interval.

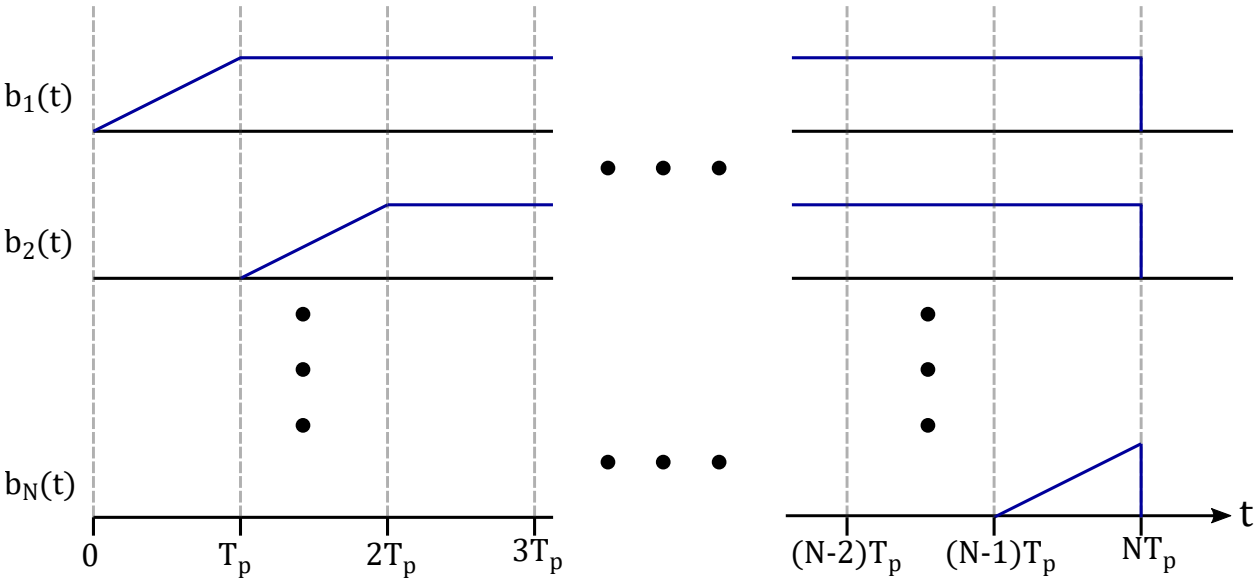

Figure 1.17: The basis functions of the first order PCFM form are time shifted ramp functions which remain constant after the ramping portion.

Still, while this model is succinct, constant amplitude, and continuous, it has to be discretized in order to represent it on a digital system. As in [38, 39], this can be accomplished via sampling the continuous time basis functions themselves such that

$$\mathbf{s} = \exp(j\mathbf{B}\mathbf{x}) \tag{1.39}$$

where the columns of the $M \times N$ matrix $\mathbf{B}$ are the sampled basis functions, $\mathbf{x}$ is the $N \times 1$ column vector of the PCFM parameters, and $\mathbf{s}$ is the $M \times 1$ sampled PCFM waveform. The total number of samples M is related to the number of PCFM parameters N by the relationship

$$M = KN \tag{1.40}$$

where K is the *oversampling* factor. Explicitly, K is the number of samples used to approximate each ramp. If the basis functions are normalized to reach a steady state value of 1. Then $\mathbf{B}$ will

take the form in Fig. 1.18 for $K = 3$. If the PCFM parameters in $\mathbf{x}$ span the entire interval $[-\pi,\ \pi]$, then K approximates the oversampling rate relative to the 3 dB bandwidth. That is, if the sample bandwidth is normalized to 1, the 3 dB bandwidth is approximately $1/K$. Conveniently, this also establishes the time bandwidth product as approximately equal to N. In practice, setting K equal to 2 or 3 is sufficient to adequately model the continuous time PCFM waveform such that a fast fourier transform (FFT) estimate of the spectrum will be a reasonably good estimate of the true continuous spectrum. Though, some degree of aliasing is inevitable since a pulsed waveform necessarily has an infinite bandwidth so true Nyquist sampling cannot be achieved. This aliasing can become problematic when extremely fine scale control of the spectrum is needed such as in the case of spectral notching. An alternative approach can mitigate the aliasing issue by, instead of sampling the time domain waveform, sampling an analytical spectral model of PCFM [40]. In general however, such an exact model is unnecessary.

$$\mathbf{B} = \begin{array}{c} \begin{matrix} \mathbf{b}_1 & \mathbf{b}_2 & & \mathbf{b}_N \end{matrix} \\ \left.\begin{matrix} 1/3 & 0 & \cdots & 0 \\ 2/3 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{matrix}\right\}\text{K=3} \\ \left.\begin{matrix} 1 & 1/3 & \cdots & 0 \\ 1 & 2/3 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1/3 \\ 1 & 1 & \cdots & 2/3 \\ 1 & 1 & \cdots & 1 \end{matrix}\right\}\text{M=KN} \end{array}$$

Figure 1.18: The structure of the first order PCFM basis matrix given $K = 3$

The discussion thus far has established the PCFM waveform structure as a continuous time waveform model which can be readily discretized such that is continuous time spectral containment properties are preserved. This behavior has been realized in numerous works as a means to optimize waveforms for such purposes as spectral containment, spectral notching, autocorrelation sidelobe mitigation, and even for nonlinear radar [38–45]. The optimization schemes typically involve some sort of iterative adjustment of the underlying PCFM parameters to minimize a cost function, (i.e. the integrated sidelobe level). The degrees of freedom is then equivalent to the number of parameters N.

In [46], it was found to be advantageous to relax the requirement that the PCFM parameters span only $[-\pi,\ \pi]$, and are instead allowed to exceed this interval. This is known as *over-phasing*. Over-phasing can result in a spreading of the spectrum, but only to a small degree if done rarely. Secondly, in [46], the relationship $N \approx BT$ was also relaxed. Such an effect can be achieved by limiting the span of the parameters such that the new span is $[-\pi/C,\ \pi/C]$, where C is the *over-coding* factor. The relationship between the number of parameters and the time bandwidth becomes $N \approx C(BT)$. In general, these additional degrees of freedom were found to lead to better optimized waveforms. The challenge however is that overcoding and overphasing can lead to poor spectral containment as they relax the constraints that make PCFM spectrally contained in the first place, so a well contained spectrum has to be guaranteed through means other than simply being a PCFM waveform. For the results in this paper, overphasing and *maximal* overcoding, where the number of PCFM parameters equals the number of waveform samples, are used, but, as will be shown, spectral containment is explicitly enforced by the optimization process itself.

1.6.2.3 Pseudo-Random Optimized Frequency Modulation (PRO-FM)

The PCFM model provides spectral containment and thus physical robustness through its underlying continuous model such that, even when the model is discretized, a sufficient oversampling factor adequately approximates the true, continuous waveform. The key to this process is that as the sample rate is increased, the phase difference between each consecutive sample is also decreased.

This is in contrast to phase codes where the phase change between chips is fixed regardless of the sample rate. With this in mind it is reasonable to ask if FM waveforms could be designed by simply limiting the phase change between samples.

Consider the algorithm defined by the alternating application of

$$r_{k+1}(t) = \mathbb{F}^{-1}\{|G(f)|\exp(j\angle\mathbb{F}\{p_k(t)\})\} \tag{1.41}$$

and

$$p_{k+1}(t) = u(t)\exp(j\angle(r_{k+1}(t)) \tag{1.42}$$

where $|G(f)|$ is some desired spectral envelope, $u(t)$ is a constant amplitude envelope, $\angle$ extracts the phase of the argument, and $\mathbb{F}$ and $\mathbb{F}^{-1}$ are the Fourier and inverse Fourier transforms respectively. In [47, 48], it was found that this was an effective means to produce FM noise waveforms with desirable, well contained spectra and good autocorrelation characteristics relative to unoptimized noise waveforms in either CW or pulsed operation. (1.41) and (1.42) are stated in continuous terms, but for implementation on a computer a discretized form is needed. The key to guaranteeing good spectral containment in this case is the choice of $|G(f)|$ and the degree of oversampling with respect to the 3dB bandwidth. Using the same notation as for PCFM, if K is set to 3 and $|G(f)|$ takes the form of a Gaussian spectrum then the discretized $|G(f)|^2$ is plotted in Fig. 1.19. The Fourier transform of a Gaussian spectrum is likewise Gaussian. Thus, if a waveform has a Gaussian spectrum then its autocorrelation will likewise be Gaussian and posses theoretically 0 autocorrelation sidelobes, which motivates the template choice.

The implementation of Pseudo-Random Optimized FM (PRO-FM) is straightforward (just a few lines of code) and computationally inexpensive as it largely consists of FFTs and IFFTs to carry out the alternating projections. To demonstrate the effectiveness of this optimization the integrated (summed) RMS and coherent autocorrelations of 1000 PRO-FM waveforms optimized according to the template in Fig. 1.19 are plotted alongside the RMS autocorrelation of 1000 random, unoptimized FM noise waveforms in Fig. 1.20 and the RMS PRO-FM spectrum in Fig.

1.21.

The meaning behind "Pseudo-Random Optimized" is apparent in Fig. 1.20. As evidenced by the RMS PRO-FM autocorrelation trace and the noise waveform trace, on an individual basis the PRO-FM waveforms demonstrate significantly improved sidelobe levels over the generic noise waveforms. Additionally, they are noise like in that the coherent autocorrelation sidelobes are decreased at a rate of $10\log_{10}(M)$, where M is the number of autocorrelations coherently summed, as evidenced by the coherent PRO-FM autocorrelation trace. Finally, Fig. 1.21 demonstrates the spectral containment of PRO-FM waveforms since the spectrum rolls off after the 3dB bandwidth.

An interesting result here is that despite the PRO-FM waveforms having no underlying continuous model, they are nonetheless physically realizable when optimized with an oversampled template [48]. In fact, they have proven to be quite robust and versatile. They have been used for such applications as practical spectral notching on [49–51], tandem-hopped radar and communications [49, 52–54], simultaneous dual-polarized emissions [55], a new form of nonlinear harmonic radar [56], and mimicking the random spatial actuation of the human eye [57]. The key to PRO-FM's implementability is that the oversampling of the template implicitly limits the magnitude of the phase change from sample to sample. Consequently, when the waveform is upsampled for implementation it does not matter how the phase is interpolated so long as a reasonable interpolation is used (e.g. linear, spline, cubic...). Since the waveform is already oversampled relative to the 3dB bandwidth, interpolation results in little distortion to the baseband spectrum facilitating high fidelity implementation. Owing to their ease of implementation and robustness, these waveforms are used as a basis of comparison to those produced for this thesis. While PRO-FM waveform are used for comparison in this work other schemes exist to produce physically amenable, FM noise waveforms such as temporal template error (TTE) optimized waveforms [58], stochastically implemented waveforms [59], and Phase-Attached Radar/Communications (PARC) waveforms [60–62].

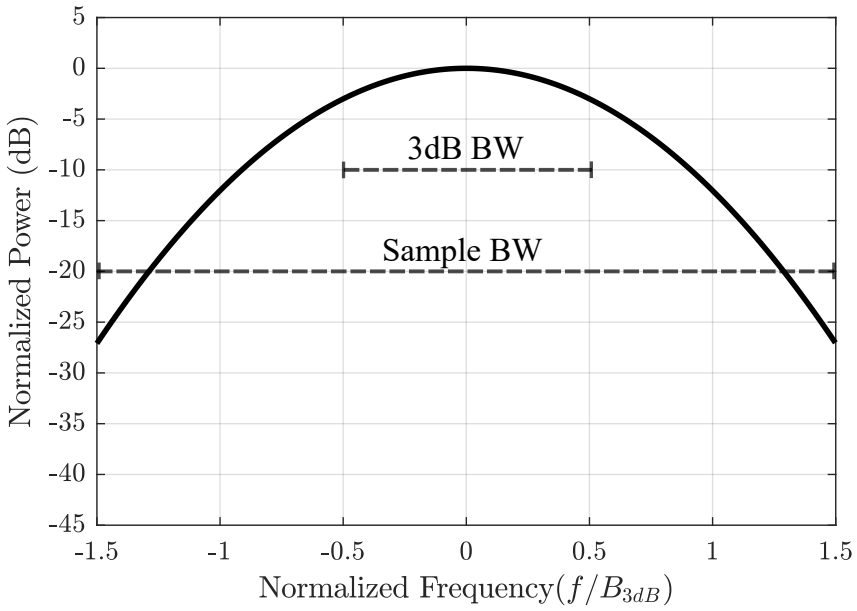

Figure 1.19: 3 time oversampled ($K = 3$), Gaussian spectral template for PRO-FM waveform optimization

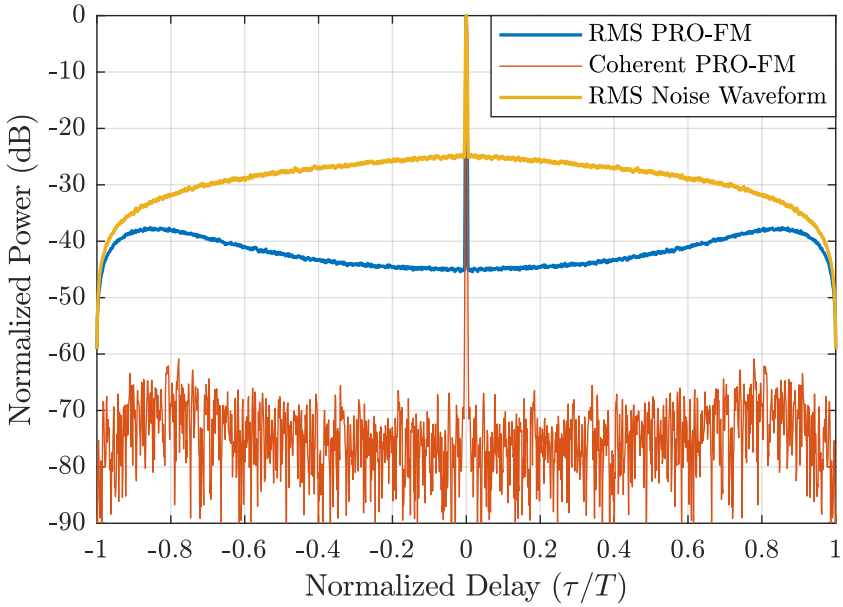

Figure 1.20: RMS and coherent autocorrelation functions of 1000 PRO-FM waveforms compared to the RMS autocorrelation function of generic noise waveforms with the same time bandwidth product

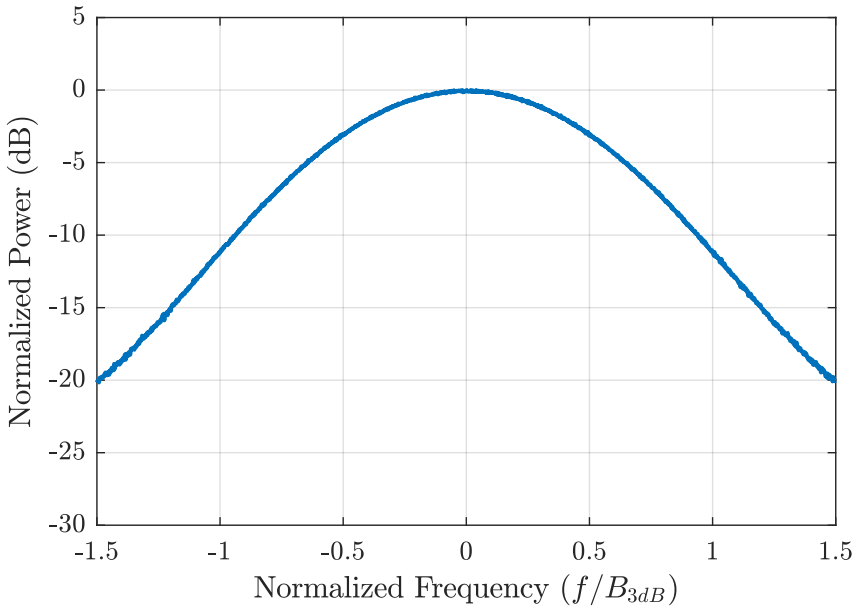

Figure 1.21: The RMS spectrum of 1000 PRO-FM waveforms optimized with the spectral template in Fig. 1.19.

1.7 Gradient Based Optimization

In general, the goal behind any optimization process is to design a system such that it meets a design goal or a set of goals to the "best" extent possible. Often times this take the form of minimizing losses, maximizing profits, maximizing efficiency, minimizing transit time, or maximizing or minimizing whatever metric is meaningful to the problem. The key words that appear again and again here are *minimization* and *maximization*. In numerical terms, this can be stated as [63]

$$\begin{aligned} &\text{minimize} \quad f_0(\mathbf{x}) \\ &\text{subject to} \quad f_i(\mathbf{x}) \leq b_i, \qquad i = 1,\ldots,m \end{aligned} \tag{1.43}$$

where $f_0(\mathbf{x})$ is some function which takes the vector of N optimization variables, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_N]$, and returns a scaler. The crux of the optimization problem is then to find the vector $\mathbf{x}$ such that $f_0(\mathbf{x})$ is minimized while the set of M inequalities, $f_i(\mathbf{x}) \leq b_i,\ i = 1,\ldots,m$, are also satisfied.

Although only minimization is mentioned here, a maximization problem can be made into a minimization problem by simply negating the cost function, $f_0(\mathbf{x})$. The entirety of cost function based optimization is described by (1.43).

Despite its simple appearance, (1.43) has been the subject of numerous publications, because of its deceptive complexity and monumental importance to various fields such as engineering or pure mathematics. Depending on the specific formulation, (1.43) can be solved in closed form or is completely intractable, has one optimal solution or numerous. For instance linear optimization considers only the cases where $f_0(\mathbf{x})$ and the constraints $f_i(\mathbf{x}) \leq b_i,\ i = 1,\ldots,m$ are linear functions of $\mathbf{x}$. In convex optimization, equation (1.43) has exactly one optimal solution or a family of co-located, optimal solutions. If the cost function and/or its constraints are nonlinear, this is sensibly called non-linear optimization. If there are no constraints at all then the optimization is said to be unconstrained. These are just a few of the major classes of numerical optimization, in fact to further complicate matters, some of these classes of problems are not even mutually exclusive. There are numerous resources on all these topics such as [63–65] just to name a few. For the purposes of this Thesis we only need to consider the form of (1.43) where $f_0(\mathbf{x})$ is non-linear, non-convex and there are no constraints.

1.7.1 Non-convex, Non-linear Cost Functions

One of the most powerful concepts in optimization theory is convexity. If convexity can be proven for an optimization problem, it means the problem has one optimal solution and this solution is a *global* minimum. That is, there is no other vector $\mathbf{x}$ such that $f_0(\mathbf{x})$ is minimized further. This is as apposed to a *local* minimum where there could be other, potentially lower, minimums to the problem [63]. To state this formally, a solution, $\mathbf{x}^*$, is a global minimizer if it satisfies

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \tag{1.44}$$

Conversely, for $\mathbf{x}^*$ to be a local minimizer it only has to satisfy the less strict requirement

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathscr{N} \tag{1.45}$$

where $\mathscr{N}$ is a neighborhood of $\mathbf{x}^*$ which itself is just an open set containing $\mathbf{x}^*$ [65]. Clearly, being able to identify whether a given solution is global is important from a practical standpoint and satisfying from an analytical perspective. However, doing so can often be difficult if not outright impossible. For example, Fig. 1.22 shows a one dimensional cut of the surface of one of the cost functions considered in this thesis. Anecdotally, this demonstrates two points. First just in this one dimensional cut of an N-dimensional problem there are numerous local minima which occur at the bottoms of the "valleys". Since most optimization algorithms only have local knowledge around a point $\mathbf{x}$ (i.e. the cost function value and its derivatives), the optimization can become "trapped" in the valley, and never find the better solutions elsewhere. In other words, without global knowledge of the cost function structure, it is very difficult to determine whether a local minima is or is not a global solution. The second point that Fig. 1.22 demonstrates is the non-convexity of the cost functions analyzed in this Thesis. Although, in the context of FM noise waveforms, this will be shown to be advantageous.

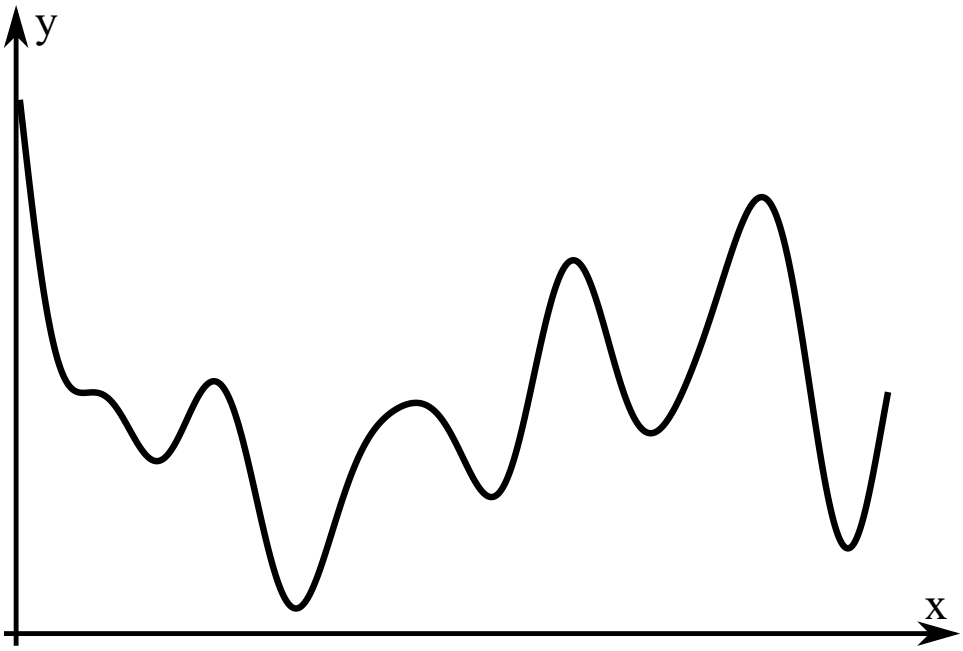

Figure 1.22: A one dimensional cut of the surface of one of the cost functions considered in this thesis

Convexity is such an important subtopic of optimization theory that entire textbooks have been devoted to exploring the convexity of different types of optimization problems and utilizing this to solve the problem. Often times, it is possible to cast or modify an optimization problem such that it is convex and can be solved using optimization algorithms designed for convex optimization problems. In this case, the optimization problem is more about how to formulate it in a solvable manner than necessarily solving the problem itself [63]. In general, convex problems generally have linear cost functions and linear constraints. There are exceptions, such as least squares optimization, but once the problem becomes non-linear, generally the best one can do is local optimization. As the name suggests rather than trying to find the global minimum, local optimization only seeks to find local minimums. In Fig. 1.22, this would mean stopping the optimization at any one the many minima. As is noted in [63], non-linear optimization and local optimization is often more of an art than a science. It relies on intuition, such as knowing what is and is not a good initialization, and the tweaking of various optimization parameters to produce good results in reasonable amounts of time. This process sometimes seems to be and often is an ad hoc process of just finding what works the best. As mentioned earlier and shown in Fig. 1.22 the cost functions considered in this

thesis are non-linear and non-convex. Accordingly, the "art" of non-linear, local optimization will become apparent in Chapter 2 where some parameters are seemingly chosen arbitrarily but were in fact chosen based on practical considerations and repeated trial and error.

This is not to say the entire process of non-linear, non-convex optimization is a game of guesswork. In fact, there are numerous algorithms that are exceptionally well defined and mathematically understood for performing local optimization [65]. As the next subsection describes, one class of methods utilizes gradients to iterate towards an optimal solution. The heuristic part of the process is deciding which algorithm to use for a problem and how to modify the algorithm to work best for the particular optimization.

1.7.2 Gradient Descent

In any iterative descent method, there are at least two steps. The first step computes a direction in which to change the cost function parameters and the next step determines how far to move in that direction. This iteration process can be written simply as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{1.46}$$

where $\mathbf{x}_k$ is the vector of the cost function parameters at iteration k, $\mathbf{x}_{k+1}$ is the vector of cost function parameters at iteration $k+1$, $\mathbf{p}_k$ is the direction in which $\mathbf{x}_k$ moves at iteration k, and finally, α_k is the step size or how far $\mathbf{x}_k$ moves along the direction $\mathbf{p}_k$. The difference between one iterative optimization method and another is generally how the search direction $\mathbf{p}_k$ is chosen. In gradient based methods, the search direction is in some way always a function of the gradient at the point $\mathbf{x}_k$. In the simplest of gradient methods, steepest descent, $\mathbf{p}_k$ is simply the negative of the current gradient $\mathbf{g}_k$.

To understand how this works, recall from high school and college calculus courses the definition of the gradient.

$$\nabla_{\mathbf{x}} J = \begin{bmatrix} \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial x_2} & \cdots & \frac{\partial J}{\partial x_N} \end{bmatrix}^T \tag{1.47}$$

In words, the gradient is simply a column vector of the first order partial derivatives of the cost function J with respect to each parameter x_n. Also covered in those courses is the significance of the gradient in terms what it says about the function J at the point $\mathbf{x}$, in that $\nabla_{\mathbf{x}} J$ is the direction of steepest *ascent*. To show this explicitly, find the direction u, where u is a unit vector, such that the directional derivative of J at the point $\mathbf{x}$ is maximized. Recall the definition of the directional derivative

$$\frac{dJ(\mathbf{x}+\alpha\mathbf{u})}{d\alpha} = \nabla_{\mathbf{x}} J \cdot \mathbf{u} \tag{1.48}$$

Using the definition of the dot product and the fact that $\mathbf{u}$ is a unit vector ,(1.48) can be rewritten as

$$\frac{dJ(\mathbf{x}+\alpha\mathbf{u})}{d\alpha} = |\nabla_{\mathbf{x}} J| \cos(\theta) \tag{1.49}$$

where θ is the angle between the vectors $\nabla_{\mathbf{x}} J$ and $\mathbf{u}$. Clearly, (1.49) is maximized where $\cos(\theta)$ is maximized. This occurs only when θ is equal to 0 and consequently when $\mathbf{u}$ and $\nabla_{\mathbf{x}} J$ are pointing in the same direction. Thus, the gradient is the direction of steepest ascent. It follows then that the direction of steepest *descent* occurs where $\cos(\theta)$ is equal to -1. In other words where $\mathbf{u}$ and $\nabla_{\mathbf{x}} J$ are pointing in opposite directions. This is the intuition behind steepest descent. By only moving in directions which decrease the cost function value, the value of J will reach some minimum value. Of course arriving at this intuition and proving it are two entirely different matters.

There are numerous caveats to consider. If the problem is constrained, what if the negative gradient moves the cost function into unfeasible space? Mathematically, decreasing at every step does not mean it will ever actually make it arbitrarily close to a local minimum. The gradient is an instantaneous measure of slope. How does one determine how far to move such that a decrease in the cost function value is actually realized? What if the function is discontinuous at the point $\mathbf{x}$ and no gradient even exists? For this thesis, the cost functions considered are both unconstrained and continuous at all points so these issues can be neglected here. Then, within theses conditions, it can be shown that steepest descent will converge to a local minimum, though the proof of this is quite involved and is not provided here, but can be found in [64].

1.7.2.1 Line Search

With a choice of direction, (some gradient derived vector), the next step is to determine how far to move in that direction such that the decrease in J is maximized or to at least guarantee J does not increase. For convenience, we define a new function

$$J(\mathbf{x}+\alpha\mathbf{p}) = w(\alpha) \tag{1.50}$$

$w(\alpha)$ is then the value of J along the direction $\mathbf{p}$. Accordingly, a value of $\alpha = 0$ simply returns the original cost function value at the point $\mathbf{x}$. In Fig. 1.22 one could imagine the y-axis as the value of $w(\alpha)$ and the x-axis as α. This presents a new one dimensional, optimization subproblem where the goal is to find the value of α such that $w(\alpha)$ is minimized. Unfortunately, in the general case, even with just one parameter to consider, this is not an easy problem. However, it is useful to first consider a much simpler trace than in Fig. 1.22, such as a simple quadratic. Such a function has exactly one minimum (convex) and can be easily minimized using techniques covered in early calculus courses, such as setting the derivative equal to zero and solving for α. As the function deviates from being quadratic, and even worse becomes non-convex such as in Fig. 1.22, exact methods may not be feasible.

Generally speaking there are two different types of line searches. The first uses curve fitting techniques. For example, the curve in Fig. 1.22 could be modeled using a Taylor series approximation. The minimum of this simpler function could then be determined and chosen as the step size. However, problems arise with this method when the curve fitting is a poor approximation of the cost function surface. In this case, the step size selected may not be the minimum of $w(\alpha)$, or even worse, it may increase the cost function value.

The other type of line search is a more heuristic approach. Rather than trying to model $w(\alpha)$ in any way it simply tries to find an α which satisfies certain conditions which guarantee a sufficient decrease in the cost function value and that this decrease is near a minimum of $w(\alpha)$. These conditions are known as the Wolfe conditions. The first of these is the sufficient decrease condition

or the *Armijo condition* [65] which is

$$J(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq J(f\mathbf{x}_k) + c_1 \alpha \nabla J_k^T \mathbf{p}_k \tag{1.51}$$

where c_1 is some constant on $c \in (0,1)$. This condition requires that the step size α must provide at least as much decrease as a first order Taylor series approximation of the cost function in the direction $\mathbf{p}_k$ with a scaled slope. Since the slope is scaled to smaller than the instantaneous directional derivative at $J(\mathbf{x}_k)$, the Armijo condition is guaranteed to be met for some sufficiently small value of α. It is good that there will always be a satisfactory value of α, however if this value is too small the potential for a larger decrease in the line search is lost and it may take an impractical number of iterations for the overall optimization problem to converge.

To address this issue we introduce the sufficient progress condition [65] stated as

$$\nabla J(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k \geq c_2 \nabla J_k^T \mathbf{p}_k \tag{1.52}$$

where c_2 is some constant on $c_2 \in (c_1, 1)$. This condition states the instantaneous directional derivative of J at $\mathbf{x}_k + \alpha \mathbf{p}_k$ in the direction $\mathbf{p}_k$ must be greater than or equal to a scaled version of the instantaneous directional derivative of J at $\mathbf{x}_k$ in the direction $\mathbf{p}_k$. Assuming $\mathbf{p}_k$ is a valid step direction, the initial $\nabla J_k^T \mathbf{p}_k$ will always be negative. In terms of the new one dimensional optimization subproblem, by requiring the derivative at $w(\alpha)$ to be greater than the derivative at $c_2 w(0)$, the step size α must then be near a stationary point or on a positive slope past the first minimum. Intuitively, if $w'(\alpha)$ is very negative for some value of α then increasing α will realize a decrease in J.

The sufficient progress condition can be modified by placing absolute value signs on either side of the equation as

$$|\nabla J(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k| \geq |c_2 \nabla J_k^T \mathbf{p}_k| \tag{1.53}$$

As (1.52) is written, $\nabla J(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k$ could be some very positive number and satisfy the sufficient

progress condition. If a very negative $w'(\alpha)$ means the cost function can be decreased further by increasing α, then a very positive $w'(\alpha)$ indicates the cost function value can be decreased further by decreasing α. (1.53) accounts for this with the absolute value signs. (1.51) and (1.53) are collectively known as the *Strong Wolfe Conditions*. Meeting these conditions effectively guarantees a "good" step length as measured by the parameters c_1 and c_2. [65] provides a few examples of algorithms which will satisfy these conditions in any line search.

Thus far, no mention has been made in terms of what values to actually use for the parameters c_1 and c_2. The only stipulation so far is that they satisfy $0 < c_1 < c_2 < 1$. This vagueness comes from the discussion in Section 1.7.1 about the "art" of non-linear optimization. Depending on the cost function, line search iterations may be computationally expensive so it may be more effective to take relatively sub-optimal steps rather than spending significant amounts of time getting every little bit of decrease possible in every line search. Conversely, a "more" convex problem could benefit from efficient line searches. The bottom line is that finding the best values of c_1 and c_2 is very problem dependent and can be an onerous process of trial and error. As far as the cost functions in this thesis are concerned, it may not even be worth it to satisfy the Strong Wolfe conditions at every iteration.

1.7.2.2 Non-linear Conjugate Gradient Descent

The crux of any gradient descent formulation is how the gradient factors into the selection of a search direction. In general, the formulation for any gradient descent method can be written as

$$\mathbf{p}_k = \begin{cases} -\nabla J_0 & \text{when } k = 0 \\ -\nabla J_k + \beta_k \mathbf{p}_{k-1} & \text{otherwise} \end{cases} \tag{1.54}$$

such that the search direction at the initial iteration is the negative of the gradient itself and the search direction at subsequent iterations is a linear combination of the negative of the current gradient and the previous search direction. The difference between each gradient method is then how the parameter β is chosen. The simplest of choices for β is 0. In which case, the descent

method is known as steepest descent where, at every iteration, the search direction is the negative of the current gradient. While steepest descent converges in general like any gradient descent algorithm, it often does so extremely slowly [65]. To improve the convergence rate, various forms of β have been considered.

In 1952, Hestenes and Stiefel proposed a β defined as [66]

$$\beta_k = \frac{\nabla J_k^T (\nabla J_{k+1} - \nabla J_k)}{\mathbf{p}_k^T (\nabla J_{k+1} - \nabla J_k)} \tag{1.55}$$

This was the first of what are known as conjugate gradient (CG) methods. This formulation in particular was made for quadratic cost functions. For an N dimensional quadratic (and therefore convex) cost function, the conjugate gradient method is guaranteed to converge in at most N steps. However, since this thesis considers non-linear cost functions it is more pertinent to discuss non-linear conjugate gradient methods (NLCG). In 1964, as a means to improve the convergence rate of non-linear cost functions, Fletcher and Reeves proposed β to be defined as [67]

$$\beta_k = \frac{||\nabla J_{k+1}||^2}{||\nabla J_k||^2} \tag{1.56}$$

Fletcher and Reeves where able to show under certain conditions how this form of β is an efficient for minimizing non-linear cost functions and so this is considered to be the original NLCG method. Still, non-linear cost functions are difficult to characterize so various other definitions of β have been proposed as means to minimize different sorts of cost functions. [68] provides an excellent survey of such methods.

As discussed early, the optimization of non-linear cost functions can be a somewhat heuristic process, consequently the most straightforward way to determine the best form of β for a problem is to simply try various forms and compare their convergence rates. Perhaps the epitome of this somewhat ad hoc choice of β is found with heavy ball gradient descent where

$$0 < \beta < 1 \tag{1.57}$$

The logic here is that the search direction of steepest descent can often be in almost opposite directions in consecutive steps leading to a poor convergence rate. The heavy ball method gives the search direction some "inertia" such that it can not change especially quickly. This makes the current search direction a sort of running average of previous search directions which will always converge to a minimum for a smooth, unconstrained cost function [69]. Interestingly enough for the cost functions in this thesis, heavy ball gradient was found to be the most efficient optimization algorithm with β equal to about .98. More will be said about this in Chapter 2.

Interestingly, another concern arises with NLCG and heavy ball methods. The search direction may become ill-conditioned and it may in fact become an ascent direction. An easy way to test this is to calculate

$$\nabla J_k^T \mathbf{p}_k = |\nabla J_k||\mathbf{p}_k|\cos(\theta) \tag{1.58}$$

If the result of (1.58) is positive then the search direction is a direction of ascent. The easiest way to address this issue is to reset the search direction the negative of the current gradient as is done in this thesis. The effects of this reseting operation will be seen in Chapter 2.

The final consideration when implementing an iterative optimization is when to stop. In general, these approaches will converge to some kind of minimum either local or global which is also a stationary point where the gradient consists entirely of 0s, but they will never make it to that point exactly with some exceptions. Because of this, a heuristic choice has to be made about when to halt the optimization procedure. The most straightforward options are to run for a set number of iterations or halt the optimization when per iteration decrease of the cost function falls below a certain limit or when the magnitude of the gradient becomes small (indicating proximity to a stationary point). Any of these options are valid, and the best option will likely depend on the behavior of the cost function itself.

Chapter 2

Generalized Frequency Template Error Optimization

In Chapter 1, practical radar waveforms were shown to be constant amplitude to minimize the distortion imposed by high power amplifiers operating in the saturation region. Secondly, practical radar waveforms were shown to be spectrally contained such that their spectrum rolls off quickly beyond the 3dB bandwidth. This permits constant amplitude upsampling schemes which minimally distort the waveform's in band frequencies as discussed in 1.6.2.3. Additionally, this prevents transmitter distortion due to band limiting effects as well at interference to and from other spectral users.

However, achieving these goals while also ensuring the waveforms themselves are good radar waveforms (i.e. low autocorrelation sidelobes) is often a difficult task. This chapter builds upon the work in [38] by defining a generalized frequency domain cost function structure, here denoted as the *generalized frequency template error* (GFTE) cost function. This cost function is minimized for many specific forms, which are shown to produce FM noise waveforms which achieve varying degrees of autocorrelation sidelobe minimization, spectral containment, and interference rejection.

2.1 The Generalized Frequency Template Error Cost Function

The general frequency template error cost function is the P-norm of the error between the transformed spectrum of some waveform and the transformed version of some desired spectrum. In continuous time, the GFTE can be written as

$$J = \left(\int_{-\infty}^{+\infty} [G(S(f)) - G(U(f))]^P df \right)^{(1/P)} \tag{2.1}$$

where $S(f)$ is the Fourier transform of the signal $s(t)$, $U(f)$ is the desired spectrum, and $G(\cdot)$ is some function which transforms $S(f)$ and $U(f)$ into a different domain. By minimizing (2.1), the error between the transformed spectrum of the signal $s(t)$ and the transformed, desired spectrum $G(U(f))$ is minimized in the p-norm sense. It will be shown that by careful choice of P and $G(\cdot)$, (2.1) can be tailored to optimize $s(t)$ in unique ways which favor different outcomes (i.e better spectral containment over autocorrelation sidelobes).

First however, the practical considerations discussed in Chapter 1 need to be considered. The waveform $s(t)$ needs to be constant modulus, pulsed, and continuous. Spectral containment will be obtained through the optimization. More succinctly, $s(t)$ is constrained such that

$$|s(t)| = \begin{cases} A & 0 < t \leq T \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where A is some constant. From this constraint, a natural choice for the form of $s(t)$ is PCFM. As discussed in Section 1.6.2.2, PCFM waveforms are capable of satisfying all the practical implementation requirements. Additionally, they are characterized by a finite set of parameters, making it possible to evaluate gradients with respect to these parameters and minimize cost functions using gradient methods as in [38–40, 42].

In order to minimize the GFTE, it is necessary to discretize (2.1) for representation on a digital system. With this in mind (2.1) can be written in a discrete form as

$$J = ||G(\bar{\mathbf{s}}_f) - G(\mathbf{u})||_P \tag{2.3}$$

where $\mathbf{u}$ is the length L sampled representation of the desired continuous spectrum, $\bar{\mathbf{s}}_f$ is the length L discrete Fourier transform (DFT) of the sampled signal $\bar{\mathbf{s}}$ which can be written as

$$\bar{\mathbf{s}}_f = \mathbf{A}^H \bar{\mathbf{s}} \tag{2.4}$$

where $\mathbf{A}^H$ is the $L \times L$ DFT matrix and $\mathbf{A}$ is the inverse DFT. $\bar{\mathbf{s}}$ is a zero padded sampled representation of the continuous time signal $s(t)$ defined as

$$\bar{\mathbf{s}} = [\mathbf{s}^T \;\; \mathbf{0}_{1\times(L-M)}]^T \tag{2.5}$$

The sampled signal $\mathbf{s}$ is zero-padded to length $L \geq (2M-1)$ samples to account for the full $2M-1$ length autocorrelation. $\mathbf{s}$ is a discretized PCFM waveform and thus takes the form of equation (1.39) in Section 1.6.2.2 ($\mathbf{s} = \exp(j\mathbf{Bx})$). Now, with a signal model selected, specific forms of $G(\cdot)$ are considered.

2.2 Frequency Template Error and Logarithmic Frequency Template Error

Specific forms of the general frequency template error cost function differ in their choice $G(\cdot)$. In this thesis, two forms of $G(\cdot)$ are considered. The first is simply called the frequency template error and it was originally explored in [41] and optimized using a greedy search algorithm. For the Frequency Template Error (FTE), $G(\cdot)$ takes the form

$$G_{FTE}(\cdot) = |\cdot|^2 \tag{2.6}$$

which is simply the element wise magnitude squared operation. In this way, the FTE metric measures the difference between the power spectrum of a candidate waveform and the desired power spectrum. Then, by minimizing J_{FTE}, the error between the power spectrum of the signal $\mathbf{s}$ and the desired spectrum $\mathbf{u}$ is minimized in a P-norm sense. By plugging (2.6) into (2.3), the FTE cost function is

$$J_{FTE} = ||\; |\mathbf{A}^H\mathbf{s}|^2 - |\mathbf{u}|^2||_P \tag{2.7}$$

The second variation considered here is called the logarithmic frequency template error (Log-FTE) which was first considered in [38] (though only the 2-norm was evaluated). In this variation, $G(\cdot)$

is defined as

$$G_{LogFTE}(\cdot) = \log_a(|\cdot|^2) \tag{2.8}$$

Here, $G(\cdot)$ takes the element wise base a logarithm of the magnitude squared. Rather than calculating the P-norm error between power spectra directly, it takes the P-norm error between the power spectra in the $\log_a$ domain. By plugging (2.8) into (2.3), the Log-FTE is

$$J_{LogFTE} = ||\log_a(|\mathbf{A}^H\mathbf{s}|^2) - \log_a(|\mathbf{u}|^2)||_P \tag{2.9}$$

Another way to think about this is to take advantage of the difference property of logarithms and rewrite (2.3) as

$$J_{LogFTE} = ||\log_a\left(|\mathbf{A}^H\mathbf{s}|^2 \odot |\mathbf{u}|^{-2}\right)||_P \tag{2.10}$$

where $\odot$ is the hadamard (element wise) product. In effect, the Log-FTE metric calculates the ratio between the candidate signal's power spectrum and the desired power spectrum and then takes the P-norm of this. This performs a kind of power normalization between the two spectra, such that the low power terms are weighted commensurately with the high power terms in the cost function on a logarithmic scale. For example, each individual error term in any kind of error metric such as the the GFTE is known as a residual. Consider the residuals for two different sample residuals of $\bar{\mathbf{s}}_f$ and $\bar{\mathbf{u}}$ under both the FTE and the Log-FTE.

$\lvert s_f\rvert^2$	$\lvert u\rvert^2$	FTE	Log-FTE
100	10	90	1
10	1	9	1

Table 2.1: Comparison between how the FTE and Log-FTE weight residuals

In Table 2.1, the log base used for the Log-FTE is 10. Clearly, the differing power levels of the individual samples of $\bar{\mathbf{s}}_f$ and $\bar{\mathbf{u}}$ in the first case (top row) versus the second case (bottom row) have

a dramatic effect on the size of the residual terms for the FTE. However, since the ratios between the samples of $\bar{\mathbf{s}}_f$ and $\bar{\mathbf{u}}$ are the same in each case, the Log-FTE provides the same result despite the order of magnitude difference in power levels. This is the primary difference between these two variations of the GFTE, and this will be shown to have a dramatic effect on the behavior and optimal solutions of each cost function.

2.3 Gradient Descent Implementation

The most important part and often the most computationally expensive step of any gradient descent method is the calculation of the gradient itself. A detailed derivation of the gradient with respect to the PCFM parameters of equation (2.3) is in Appendix A, but the final result is included here for convenience.

$$\nabla_{\mathbf{X}} J = 2\overline{\mathbf{B}}^T \Im \left\{ \bar{\mathbf{s}}^* \odot \mathbf{A} \left([G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u})] \odot \left| G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u}) \right|^{P-2} \odot G'_*(\mathbf{A}^H \mathbf{s}) \right) \right\} J^{(1-P)} \tag{2.11}$$

The next step is to plug in (2.6) for the FTE and (2.8) for the Log-FTE along with their derivatives which are derived in Section A.2 of Appendix A. Applying those results, the gradient of (2.7), the FTE, is

$$\nabla_{\mathbf{X}} J_{FTE} = 2\overline{\mathbf{B}}^T \Im \left\{ \bar{\mathbf{s}}^* \odot \mathbf{A} \left([|\mathbf{A}^H \bar{\mathbf{s}}|^2 - |\mathbf{u}|^2] \odot \left| |\mathbf{A}^H \bar{\mathbf{s}}|^2 - |\mathbf{u}|^2 \right|^{P-2} \odot (\mathbf{A}^H \mathbf{s}) \right) \right\} J^{(1-P)} \tag{2.12}$$

and the gradient with respect to (2.9), the Log-FTE, is

$$\nabla_{\mathbf{X}} J_{LogFTE} = 2\overline{\mathbf{B}}^T \Im \left\{ \bar{\mathbf{s}}^* \odot \mathbf{A} \left(\mathbf{g} \odot |\mathbf{g}|^{P-2} \odot (\mathbf{A}^T \bar{\mathbf{s}}^*)^{-1} \right) \right\} \frac{J^{(1-P)}}{\ln(a)} \tag{2.13}$$

where

$$\mathbf{g} = \log_a(|\mathbf{A}^H \bar{\mathbf{s}}|^2) - \log_a(|\mathbf{u}|^2) \tag{2.14}$$

Given the relative similarity of the FTE and the Log-FTE cost functions, the similarity of the gradients is unsurprising. One very important quality which both gradients posses is that they can be computed using only FFTs and matrix vector multiplies which has been proven useful for these types of optimization problems as in [38, 39, 70, 71]. In terms of computational expense, the biggest bottleneck is then the multiplication of $\overline{\mathbf{B}}^T$ with the rest of equation. This operation is $O(N^2)$. Conveniently however, the form of the first order PCFM basis matrix $\mathbf{B}$ has a large amount of redundancy as shown in Fig. 1.17. In fact, the only entries which are non-zero or non-unity are restricted to the main diagonal, and the $K-1$ diagonals below. This structure can be exploited to reduce the computation time of the matrix multiplication to insignificance when compared to the FFT execution time. Therefore, for first order PCFM waveforms, (2.12) and (2.13) can be calculated in $O(N\log(N))$. Being able to calculate the gradient quickly and efficiently is extremely important for gradient based optimization and particularly in this case. As will be shown, the FTE and Log-FTE cost functions are highly non-convex and thus require a large number of iterations relative to their dimensionality to converge. Therefore, each iteration needs to be as fast as possible to optimize these waveforms efficiently.

Beyond the speed of the gradient calculation, there are more issues related to computation of the gradient and the cost function itself. For example, numerical stability can become an issue when using higher values of P. This is simply because high values of P create both very large and very small numbers which are subject to numerical precision errors. Additionally, the gradient of the Log-FTE, (2.13), requires an element wise inversion. For small values, this can cause certain residuals to explode in size and dominate the cost function leading to slow convergence and/or a poor solution. Mitigating this issue is discussed in Section 2.4 since it becomes an issue for highly oversampled optimization.

A simple modification can be made to the cost functions themselves to prevent some precision errors, simplify their calculation slightly as well as the calculation of their gradients, and ensure the cost functions are differentiable at their solutions. (Though the differentiability of the cost function at a solution is not a problem in practice.) The FTE and Log-FTE cost functions can be modified

by simply taking the cost functions to the P_{th} power. The FTE and Log-FTE cost functions become

$$J_{FTE} = || \, |\mathbf{A}^H \mathbf{s}|^2 - |\mathbf{u}|^2||_P^P \tag{2.15}$$

and

$$J_{Log-FTE} = ||\log_a(|\mathbf{A}^H \mathbf{s}|^2) - \log_a(|\mathbf{u}|^2)||_P^P \tag{2.16}$$

In practice, this just means omitting the P_{th} root operation when calculating the cost function values. Additionally, with this change the gradients, (2.12) and (2.13) are unchanged except for the omission of the J^{1-P} term.

2.4 Optimization Parameters

In optimizing with FTE and Log-FTE cost functions via gradient descent methods there are numerous parameters to consider. Some of these parameters come from the waveform model and the cost function itself, while others pertain to the optimization method. Table 2.2 outlines these parameters.

Cost Function	Waveform Model (PCFM)	Optimization Method
Template	Time-Bandwidth Product	Gradient Method
P-Norm	Overcoding Factor	Line Search
	Initialization	Stopping Conditions

Table 2.2: Optimization Parameters

Exhaustively testing each of these parameters is simply impractical, especially considering some of these parameters have additional sub-parameters. For example, oversampling with respect to the 3dB bandwidth is tied into the choice of template and then the choice of line search and stopping conditions can also introduce additional parameters as discussed in Sections 1.7.2.1 and 1.7.2.2

For purely pragmatic reasons, we need to determine which parameters are the most meaningful in the context of this thesis and fix the rest. Considering this work is primarily about the behavior and utility of the FTE and Log-FTE cost functions, the particular optimization method used is just the means to an end, so the choice of a particular gradient method, line search, and stopping conditions will be fixed for all optimizations. The independent variables will then be the cost function and waveform parameters.

2.4.1 Gradient Descent Parameters

Starting with the fixed parameters, in the course of performing the optimization, it was found that the heavy ball gradient descent method provided the best convergence rate while still being very stable. Several different variants of NLCG methods were implemented as well, but they did not converge much more quickly than the heavy ball method, if at all, and, especially in the case of the Log-FTE, they had issues with numerical stability and frequent resets to the steepest descent direction. The fact that NLCG methods would be slower than heavy ball is surprising given the demonstrated performance of these methods in general [68]. However, qualitatively speaking, the FTE and Log-FTE are extremely non-convex functions and conjugate gradient methods tend to work better the closer the cost function is to being convex. For these reasons, the heavy ball method is used for all optimizations.

The next optimization parameter is the line search. Considering how quickly the gradient can be calculated and how non-convex the cost functions are, large decreases in the cost function should not be expected on any give iteration. Consequently, performing more total iterations is more important than achieving the maximum decrease on any given iteration. With this in mind, a quick and inexact form of backtracking was used to minimize the time spent on the line search at the expense of more total iterations, but these iterations are fast anyway due to the efficiency of the gradient calculation. To briefly describe this method, an initial step size is chosen arbitrarily. If the step size is too large it is decreased by a factor ρ, and if the step size is found to decrease the function, it is increased by a factor μ. Typical values of ρ and μ are .9 and 1.01 respectively.

This method is a poor choice for finding the best maximum decrease at each step, but at least for the FTE and Log-FTE, it was found to be stable, very fast, and still provide good convergence.

Finally, the last optimization method specific parameter is the choice of stopping conditions. As discussed in Section 1.7.2.2, there are several choices, but once again we make a very pragmatic decision. For the sake of convenience, the number of iterations is set to a fixed number. This can be done since iterations can be performed quickly enough that, for example, >10000 iterations can performed on 1000 waveforms in just a few minutes. Additionally, comparing the convergence rate of different cost functions is very straightforward this way. For the purposes of this thesis, this is sufficient.

Fixing the optimization method specific parameters was not done to downplay their importance. There is certainly a lot of interesting future work to be done optimizing the optimization, that is to make the optimization of these cost functions as fast and efficient as possible. It is just that there are more details and more results than could be fit into this thesis while still considering the overall characteristics and performance of the cost functions themselves. To reiterate, all optimizations performed in this thesis use heavy ball gradient descent, a simple inexact backtracking line search and a fixed number of iterations.

2.4.2 PCFM Waveform Parameters

For the PCFM waveform, the time-bandwidth product (BT), the overcoding factor, and the oversampling rate are all interrelated. First, all waveforms will be maximally overcoded. As discussed in Section 1.6.2.2, this means the number of waveform samples is equivalent to the number of PCFM parameters. Doing so affords the maximum number of design degrees of freedom leading to the best optimized waveforms. One concern with maximal overcoding is spectral containment since some of the continuous time fidelity is sacrificed for more design freedom, but because the cost functions used in this thesis explicitly shape the spectrum this will not be a problem. For the time-bandwidth product and the oversampling factor, the number of samples and consequently the number of maximally overcoded PCFM parameters was set to 900 for all optimizations. This

ensures that all comparisons between the cost functions are fair and physically implementing the waveforms stays within time and memory constraints. This means the oversampling factor K and the BT are related by $900 = BT \times K$. Throughout all the results in this thesis, two different oversampling factors of $K = 3$ and $K = 6$ are considered. Changing the oversampling factor in this way changes both the BT by the relationship above and more importantly the spectral templates below.

2.4.3 Cost Function Parameters

Since the FTE and Log-FTE cost functions match a waveform to a particular spectrum, the most important choice for these cost functions is the template itself. For the purposes of FM noise waveforms, the ideal spectrum rolls off quickly for good spectral containment, and corresponds to a good autocorrelation. For this reason, as explained in Section 1.6.2.3, PRO-FM typically optimizes waveforms to match a Gaussian spectrum. A Gaussian spectrum naturally rolls off quickly and because the inverse Fourier transform of a Guassian is likewise Gaussian, the autocorrelation is also Gaussian and theoretically possesses no sidelobes. For this reason, the spectral envelopes considered here are Gaussian in shape. Additionally, as will be shown, the FTE and Log-FTE differ primarily in how they optimize low power frequency bins. To demonstrate this, two different oversampling factors are considered, $K = 3$ and $K = 6$ since these templates have very different power levels across the entire sample bandwidth. The corresponding template shapes are shown in Figs. 2.1 and 2.2.

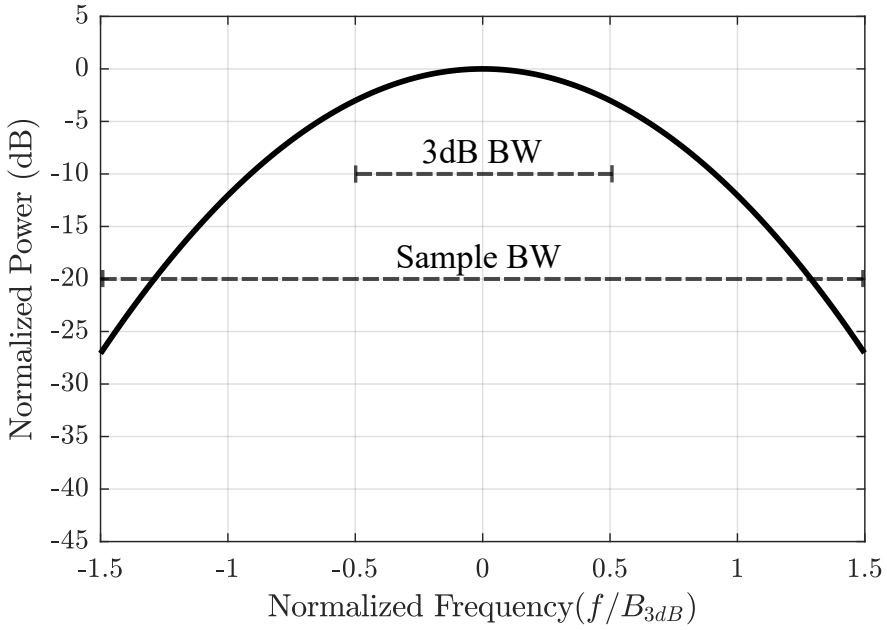

Figure 2.1: Gaussian spectral template with 3 times oversampling ($K = 3$)

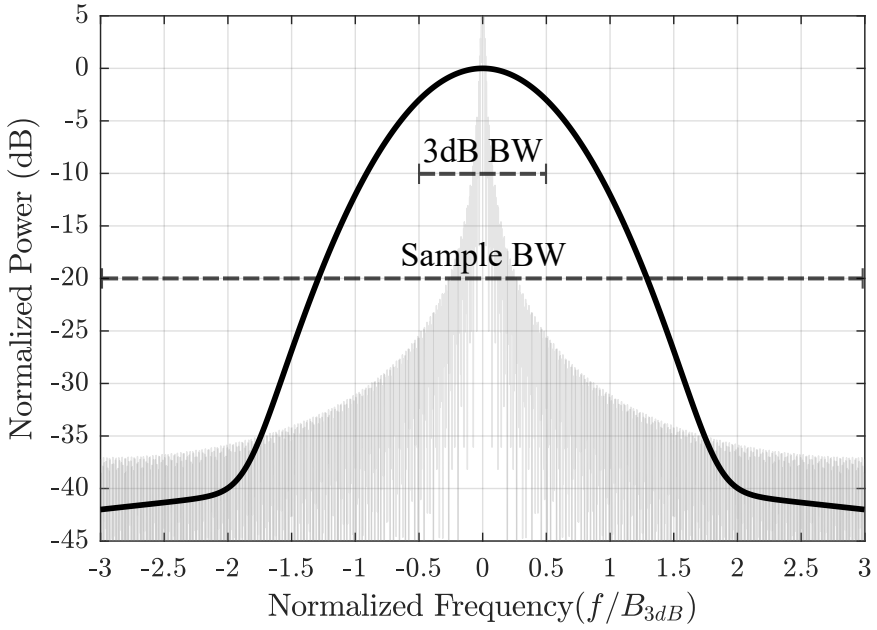

Figure 2.2: Gaussian spectral template with 6 times oversampling ($K = 6$) with a superimposed Sinc function

The oversampling factor K refers to the oversampling relative to the 3dB bandwidth. The frequency axes in Figs. 2.1 and 2.2 have been normalized to the 3dB bandwidth. The maximum positive and negative frequencies represent the sample bandwidth which is the highest frequency the DFT can measure given the oversampling factor and sample rate. Additionally, the $K = 6$ template in Fig. 2.2 has been modified by adding a smooth floor with a slight slope as it approaches the sample bandwidth. If left unmodified, the $K = 6$ template would approach extremely small values near -80 dB at the edges of the sample bandwidth. In the course of optimizing waveforms using the Log-FTE cost function, it was found that these power levels are simply unobtainable given the square, constant modulus pulse shape. In fact, the Log-FTE cost function would converge to very poor solutions, very slowly as it tried in vain to match the waveform spectrum to these extremely low power levels. Generally, the lowest levels it could attain were around -40dB. To accommodate this issue and to facilitate a good solution over the entire template, the slow rolloff below -40dB was added. Interestingly, this floor corresponds remarkably well to a sinc spectrum corresponding to a square, unmodulated pulse of the same power and duration as the waveform being optimized. This sinc correspondence is shown in Fig. 2.2 and likely has to do with the square shaped pulse and individually square-shaped chips of first order PCFM structure as defined in Section 1.6.2.2.

Beyond simply achieving good spectral containment or low autocorrlation sidelobes, it may also be beneficial to entirely avoid certian frequency bands, even inside the 3 dB bandwidth. This may be necessary when trying to avoid interference or operate simultaneously with communications systems [72]. The template in Fig. 2.3 is identical to the $K = 3$ oversampling template in Fig. 2.1 except that a -42 dB deep spectral notch has been placed within the passband. In various previous works, square spectral notches have been shown to have deliterious effects on the autocorrelations by creating a $\sin(x)/x$ roll-off. To mitigate this effect, the notch shape has been modified by adding a Tukey taper to the edges [50]. This has been shown to at least somewhat mitigate the problematic increase in sidelobe levels.

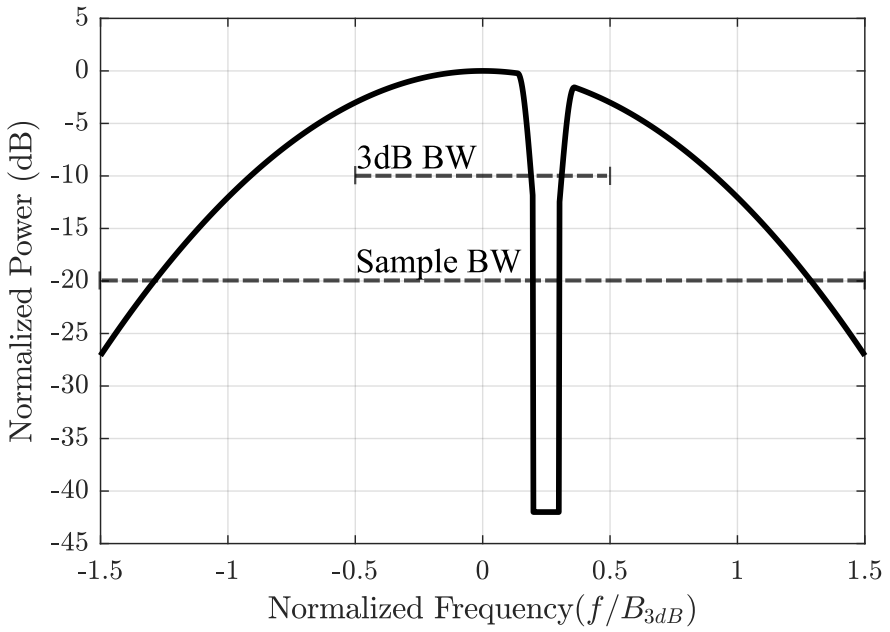

Figure 2.3: Gaussian spectral template with 3 times oversampling ($K = 3$) and with a notch located within the passband

The last parameter and the first one to be explored in this thesis is the the P-Norm parameter. This value determines how each cost function measures "closeness" to the template. Other works have used the P-Norm as a way of generalizing autocorrelation metrics, in particular integrated sidelobe level (ISL) and peak sidelobe power (PSL) [42]. In these works, $P = 2$ corresponds to the ISL metric, the total sidelobe power, and $P = \infty$ corresponds to the PSL metric, the peak sidelobe power. Values of P in between 2 and ∞ measure something in between these metrics. In the same way, for the FTE and Log-FTE $P = 2$ measures the total deviation from the template in a squared error sense, whereas $P = \infty$ measures the peak deviation from the template. Clearly, setting $P = \infty$ is not physically possible, but sufficiently large values of P have a similar effect. From experimentation, "sufficiently large" is the upper single digits. Alternatively, one could use the 1-norm, but this results in a discontinuous cost function with respect to the first derivative resutling in poor convergence, instability, and overall poor outcomes. Considering the large range of possible values of P it is useful to find the best value in terms of converging closest the template

before directly comparing the FTE and Log-FTE. Doing so cuts down on the number of test cases when comparing FTE, Log-FTE, and PRO-FM directly.

2.5 P-norm Evaluation

To examine the behavior of the FTE and Log-FTE cost functions for different values of P, an extensive number of optimizations were performed. For both the FTE and Log-FTE cost functions, sets of 1000, randomly initialized PCFM waveforms were optimized for each of the cases encompassed by Table 2.3. That is, for each of the spectral templates in Figs. 2.1, 2.2, and 2.3, 1000, randomly initialized waveforms were optimized for each value of P from 2-10. In each case the time bandwidth product was defined as $BT = N/K$ such that the number of samples in each waveform is fixed to $N = 900$. To ensure each optimization was sufficiently converged, each optimization was run for a fixed number of iterations totaling 100000, which as will be discussed shortly, is generally on the order of 5 times more iterations than necessary. In total, this resulted in the optimization of 54000 individual waveforms.

Template	Time Bandwidth Product	P-Norms
Gaussian (K=3)	300	2-10
Gaussian (K=6)	150	
Gaussian with Notch (K=3)	300	

Table 2.3: P-Norm simulation test cases.

Unfortunately, comparing the relative performance of each value of P is not as straightforward as comparing the final values of the cost functions in each case. Although, for each value of P the FTE and Log-FTE cost functions fundamentally measure the error between a candidate waveform's spectrum and a desired spectrum, they do so in different senses making it difficult to come to any conclusion based on their raw values. Instead, it is more instructive to compare the quality of the

resulting optimized waveforms against common metrics such as the peak sidelobe level, integrated sidelobe level, and in general how well the waveforms match the desired spectral template in both linear and dB scales.

Figures 2.4 - 2.6 show the resulting mean power spectra and RMS autocorrelations. Since the waveforms optimized are sets of 1000 FM noise waveforms it is appropriate to present the results of the optimizations as averages. Additionally, before comparing the FTE and Log-FTE cost functions directly it is important to establish which values of P are the most effective in each case, (i.e. if the 7-norm performs best for FTE for one template while the 4-norm performs best for the Log-FTE cost function for the same template then these are the respective FTE and Log-FTE P-norm valued cost functions that should be compared directly for that template). To make judging the ideal P-norm values more straightforward and for the sake of clarity, the FTE and Log-FTE results are presented separately with Figs. 2.4 and 2.5 pertaining to the FTE cost function and Figs. 2.6 and 2.7 pertaining to the Log-FTE cost function.

In Figs. 2.4a and 2.4b, the resulting mean power spectra over the sets of 1000 optimized FTE waveforms are plotted for each P-norm value from 2-10 along with the initial power spectrum and the desired spectral templates which are the $K = 3$ and $K = 6$ cases respectively. In Figs. 2.4c and 2.4d, the corresponding RMS autocorrelations are plotted along with the RMS autocorrelation of the random initialization. In Fig. 2.5, the same respective traces are plotted for the FTE results but for the notched template. To emphasize the most extreme P-norm values (2 and 10), the $P = 2$ and $P = 10$ cases are plotted in red and blue respectively, while the other P-norm values are plotted in light gray to aid in the understanding how the behavior of the FTE cost function changes as the P-norm value is increased, but without cluttering the plot area. Lastly, the spectral traces are normalized to the template power, while the autocorrelation traces are normalized to the peak autocorrelation power.

Starting with the $K = 3$ template, Figs. (2.4a and 2.4c), there is little difference in the resulting mean spectra in any case. It could be argued that the $P = 2$ sticks to the template a little better than the others as the normalized frequency increases and that $P = 10$ case achieves slightly better

spectral containment towards the edge of the sample bandwidth than any of the other traces, but these differences aren't especially significant. Interestingly however, in the RMS autocorrelation plot in Fig. (2.4c), the $P = 2$ FTE cost functions clearly achieves the best RMS autocorrelation, while $P = 3$ is second best, $P = 4$ is third best and so on until the $P = 10$ case demonstrates the poorest autocorrelation results. Turning our attention to the $K = 6$ template results in Figs. 2.4c and 2.4d), the RMS autocorrelation results are similar to the $K = 3$ template results albeit with slightly poorer sidelobe performance due to the time bandwidth product being divided by two. The $K = 6$ mean spectra however tell a different story. The $P = 2$ FTE cost function clearly provided the poorest spectral containment towards the sample bandwidth edges and throughout the entirety of the rolloff region while $P = 10$ clearly provides the best containment. For the higher oversampling rate case then this opens up a trade space the lower p-norm values provide superior autocorrelation sidelobes, but the higher ones provide superior spectral containment. Next, we consider the notched template in Fig. 2.5. In Fig. 2.5a, the $P = 2$ trace clearly demonstrates best notch depth (though still poor) while each subsequent value of P provides poorer and poorer depth. Otherwise, the spectral results are similar to the those seen in Fig. 2.4a. The autocorrelation results on the other hand, demonstrate a new effect where there are sin(x)/x like sidelobes emanating from the mainlobe, without the Tukey taper applied to the notch these sidelobe would be much worse [50]. Still, even with these shoulder lobes, the $P = 2$ trace clearly demonstrates the best RMS autocorrelation sidelobes. With these results in mind for comparisons to the Log-FTE and PRO-FM waveforms in Section 2.6, the $P = 2$ FTE cost function will be used for each template. For the $K = 3$ and notched templates this choice is obvious based on this discussion, but the $K = 6$ template presents a little more nuance. Although, the 10-norm provides superior spectral containment, the 2-norm plays more directly to the strengths of the FTE cost function as it provides excellent autocorrelation characteristics. This makes for a more stark contrast to Log-FTE results in Section 2.6.

Figures 2.6 and 2.7 are organized in a manner similar to the P-norm FTE results in Figs. 2.4 and 2.5 but for the P-norm Log-FTE optimized waveforms. Beginning with the $K = 3$ template,

in Fig. 2.6a each P-norm Log-FTE cost function provides roughly the same template match with perhaps the higher P-norm values providing a slightly poorer match but not significantly so. The RMS autocorrelation results in Fig. 2.6c however do show more of a difference where the $P = 2$ trace provides roughly 5 dB average lower sidelobes near and around mainlobe than the most extreme norm. Similar results are seen for the $K = 6$ template in Figs. 2.6b and 2.6d where the spectral matches are relatively equivalent across all norms but the $P = 2$ norm provides slightly better autocorrelation sidelobes albeit somewhat less in this more oversampled case. The notched template results in Fig. 2.7 tell a more nuanced story. First, in Fig. 2.7a notice how the $P = 2$ does not provide the best template match. In general each P-norm Log-FTE cost function achieves roughly the same notch depth with the higher values maybe being slightly worse, but in this case the $P = 3$ trace is highlighted in green to show that it actually provides the best template match outside of the notch region. Likewise, in the RMS autocorrelation results of Fig. 2.7b, we see the same sort of shoulder lobes as in the FTE case, but here the 2-norm Log-FTE cost function provides, objectively, the poorest autocorrelation while the $P = 3$ trace demonstrates the best RMS autocorrelation. This is in contrast to every other case examined for both the P-norm FTE and Log-FTE where the 2-norm was strictly superior in terms of the RMS autocorrelation.

To provide insight into this surprising result and to, in general, provide insight into the behavior of the optimization process, Fig. 2.8 plots the average cost function value (over each set of 1000 waveforms) as a function of iteration for each one of the P-norm results shown in Figs. 2.4 - 2.7. First though, it is important to note how the y-axis has been normalized. As mentioned earlier it is not meaningful to directly compare the absolute cost function values across different P-norms as these represent fundamentally different cost functions. In an effort to make the traces more comparable, the y-axis has been normalized to show the *convergence fraction* defined as

$$\text{Convergence Fraction} = \frac{J_i - J_F}{J_0 - J_F} \tag{2.17}$$

where J_i is the value of the cost function at the i_{th} iteration, J_0 is the initial cost function value,

and J_F is the cost function value after the final iteration which in this case is iteration 100000. Accordingly, each y-axis in Fig. 2.8 goes from 0 to 1 and each trace begins at 1 and ends at 0. In this way, if a trace at the i_{th} iteration has a value of .2 then the optimization is 80% of the way to its value at the last iteration. Before getting into specific details it is interesting to note that in pretty much each case the optimizations did pretty much nothing in the first 1000 or so iterations. This is a result of a conscience choice to make the initial step size extremely small (10^{-12}). This was done as a precaution to prevent a problematic corner case where the initial step size and gradient are poorly scaled. As far as the cost functions are concerned, the initializations are extremely poor meaning a jump in any direction by any amount has a good chance of resulting in a cost function decrease. If this jump is too large, it can badly scale the PCFM parameters leading to an poor overall solution. The result of the compensation (making the initial step size extremely small) is that the first 1000 or so iterations are spent increasing the step size through the process described in Section 2.4.1. One way to mitigate this is to use a more elegant step size algorithm, but for the purposes of this work this was unnecessary as the overarching goal is to compare the effectiveness of the FTE and Log-FTE as waveforms optimization cost functions. The convergence rate as a function of the iteration number is only a concern here inasmuch as it impedes this comparison. Thankfully, the forms of the gradients for the FTE and Log-FTE make iterations cheap. Hence, 100000 iterations were performed in each case.

Examining the convergence itself now, in Figs. 2.8a-e, the $P = 2$, trace takes the longest to begin meaningfully optimizing, but once it does it quickly overtakes the traces of the other norms. Figs. 2.8a-e encompass all the P-norm FTE results as well as the $K = 3$ and $K = 6$ results for the Log-FTE. These cases indicate that the 2-norm not only provides the best waveforms, but it does so more quickly. Figure 2.8f on the other hand, corresponding to the notched P-norm Log-FTE case in Fig. 2.7, demonstrates a more chaotic situation. Each trace exhibits numerous "kinks" where the magnitude of the slope abruptly decreases. These correspond to reset events where the search direction was found to actually be an ascent direction. In these instances, the search direction was reset to the steepest descent direction, the negative of the gradient. In addition to this there a

clustering phenomena occurs where several of the P-norm exhibit similar convergence rates. The most glaring results however is the remarkably poor convergence of the 2-norm trace. There are some difference between the other ones as well but in general they are all more or less converged to 95% of their final value by 20000 iterations, but it takes the 2-norm trace almost 80000 iterations to reach the same point. Still, as mentioned above since the optimization scheme itself has not been rigorously optimized these convergence traces need to be analyzed with caution. However, the 2-norm trace for the Log-FTE notched template, Fig. 2.8f, converges so much more poorly than any other case for either the FTE or Log-FTE that it is hard to imagine that kind of tweaking of the underlying optimization could bring it on par with the other P-norms. Instead this is likely a quirk of the Log-FTE cost function when including a notch.

To explain this, consider how the different P-norms quantize the error. The 2-norm measure the total squared template error, while higher norms measure something closer and closer to the maximum template error. A plausible explanation for the poor performance of the 2-norm in the Log-FTE notched case is that the optimization is having trouble reducing the error by deepening the notch without inducing a greater error elsewhere. The higher norms on the other hand can very effectively decrease the cost function value by deepening the notch since the peak error terms induced by the notch are far greater than anywhere else across the template. Thus, the higher norms can deepen the notch without worrying too much about inducing errors elsewhere. In simulation, the ideal compromise appears to be the 3-norm as this produces the best spectrum and RMS autocorrelation in Fig. 2.5. With this in mind the best optimization performance for the Log-FTE cost function is achieved by the 2-norm for the $K = 3$ and $K = 6$ templates and by the 3-norm for the notched template.

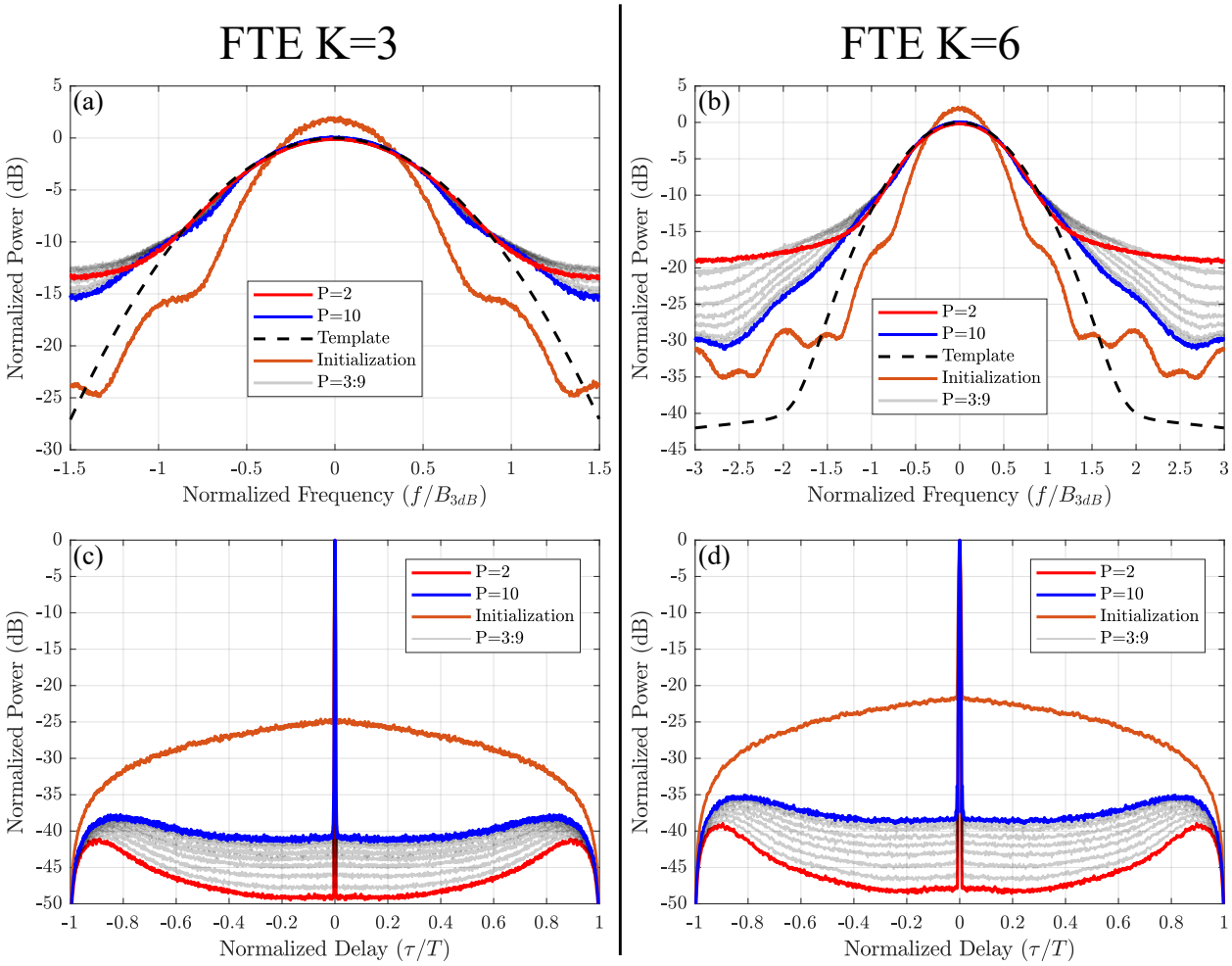

Figure 2.4: Mean power spectrum for 1000 P-norm FTE optimized waveforms for each value of P from 2-10 for both the $K = 3$ template (a) and the $K = 6$ template (b). RMS autocorrelation function for 1000 P-norm FTE optimized waveforms for each value of P from 2-10 for both the $K = 3$ template (c) and the $K = 6$ template (d).

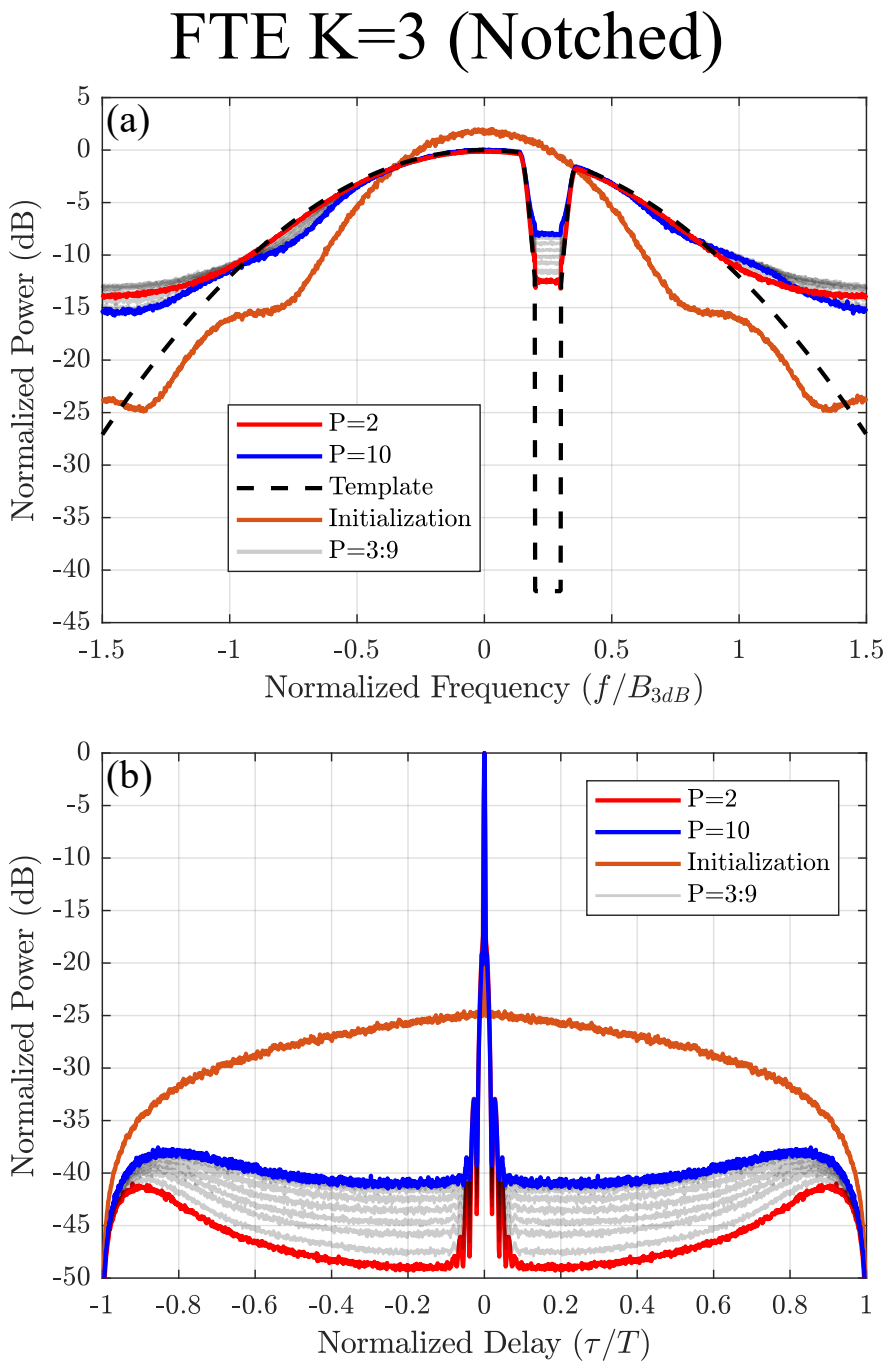

Figure 2.5: Mean power spectrum for 1000 P-norm FTE optimized waveforms for each value of P from 2-10 for the notched $K = 3$ template (a). RMS autocorrelation function for 1000 P-norm FTE optimized waveforms for each value of P from 2-10 for the notched $K = 3$ template (b).

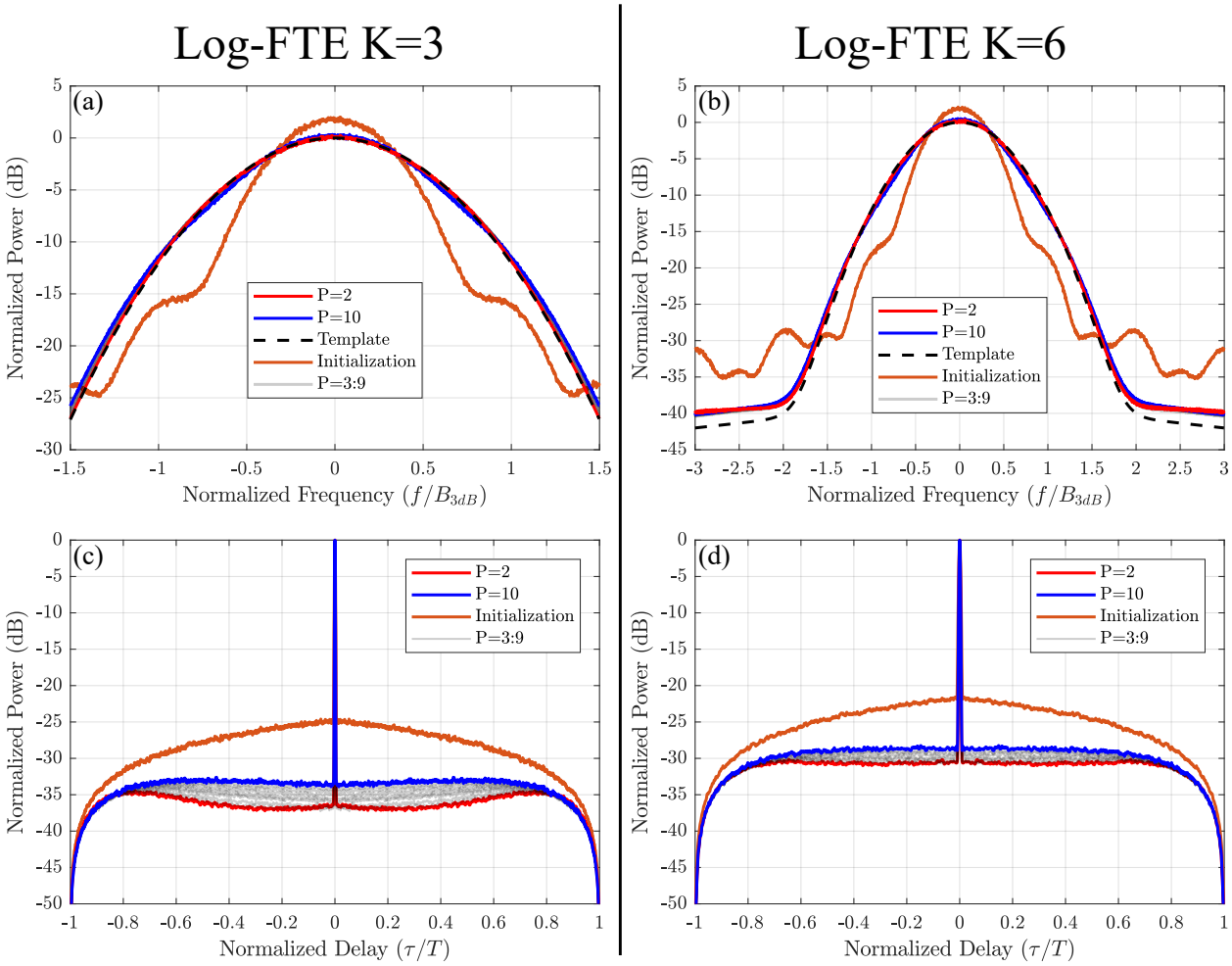

Figure 2.6: Mean power spectrum for 1000 P-norm Log-FTE optimized waveforms for each value of P from 2-10 for both the $K = 3$ template (a) and the $K = 6$ template (b). RMS autocorrelation function for 1000 P-norm Log-FTE optimized waveforms for each value of P from 2-10 for both the $K = 3$ template (c) and the $K = 6$ template (d).

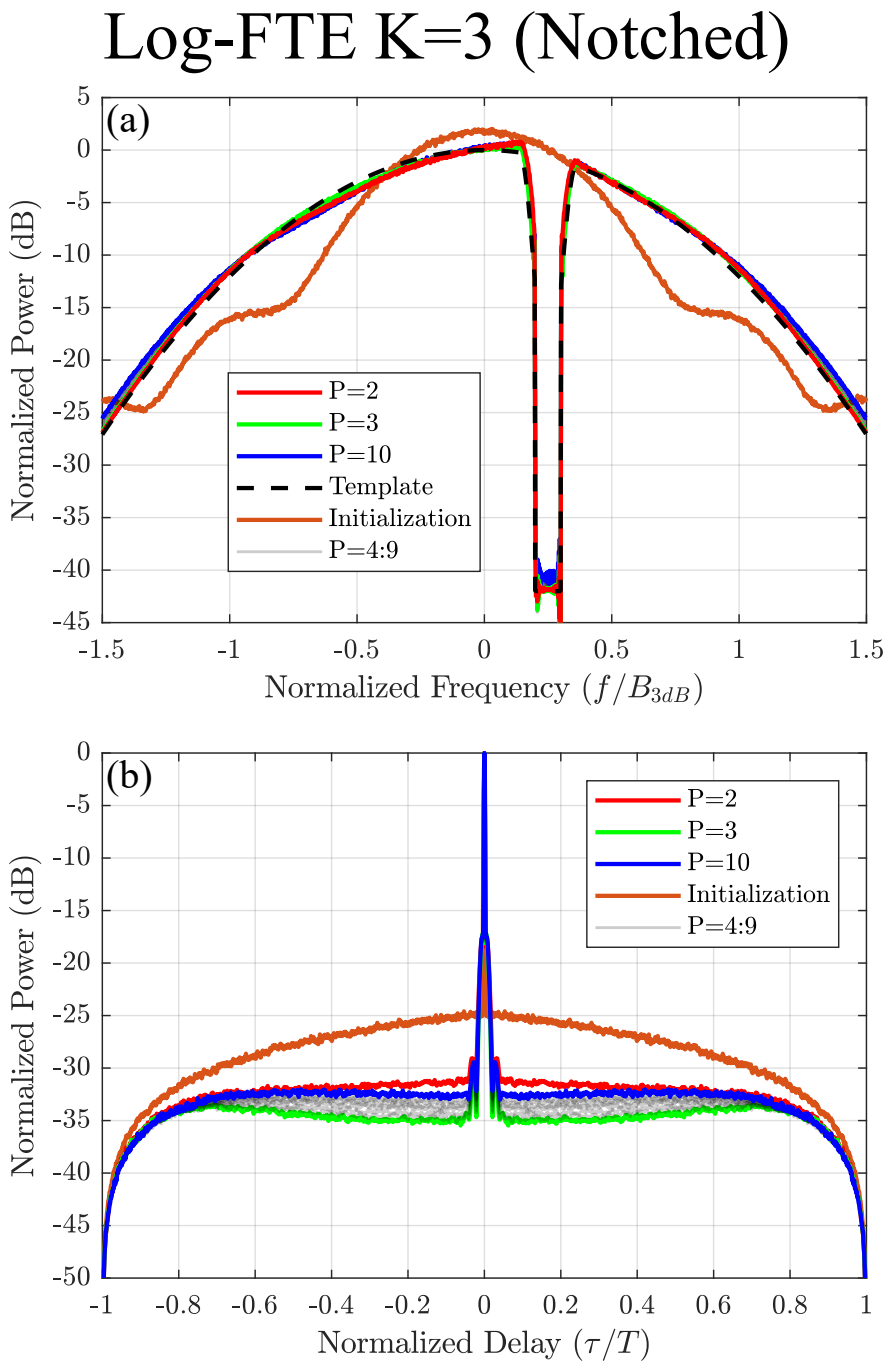

Figure 2.7: Mean power spectrum for 1000 P-norm Log-FTE optimized waveforms for each value of P from 2-10 for the notched $K = 3$ template (a). RMS autocorrelation function for 1000 P-norm Log-FTE optimized waveforms for each value of P from 2-10 for the notched $K = 3$ template (b).

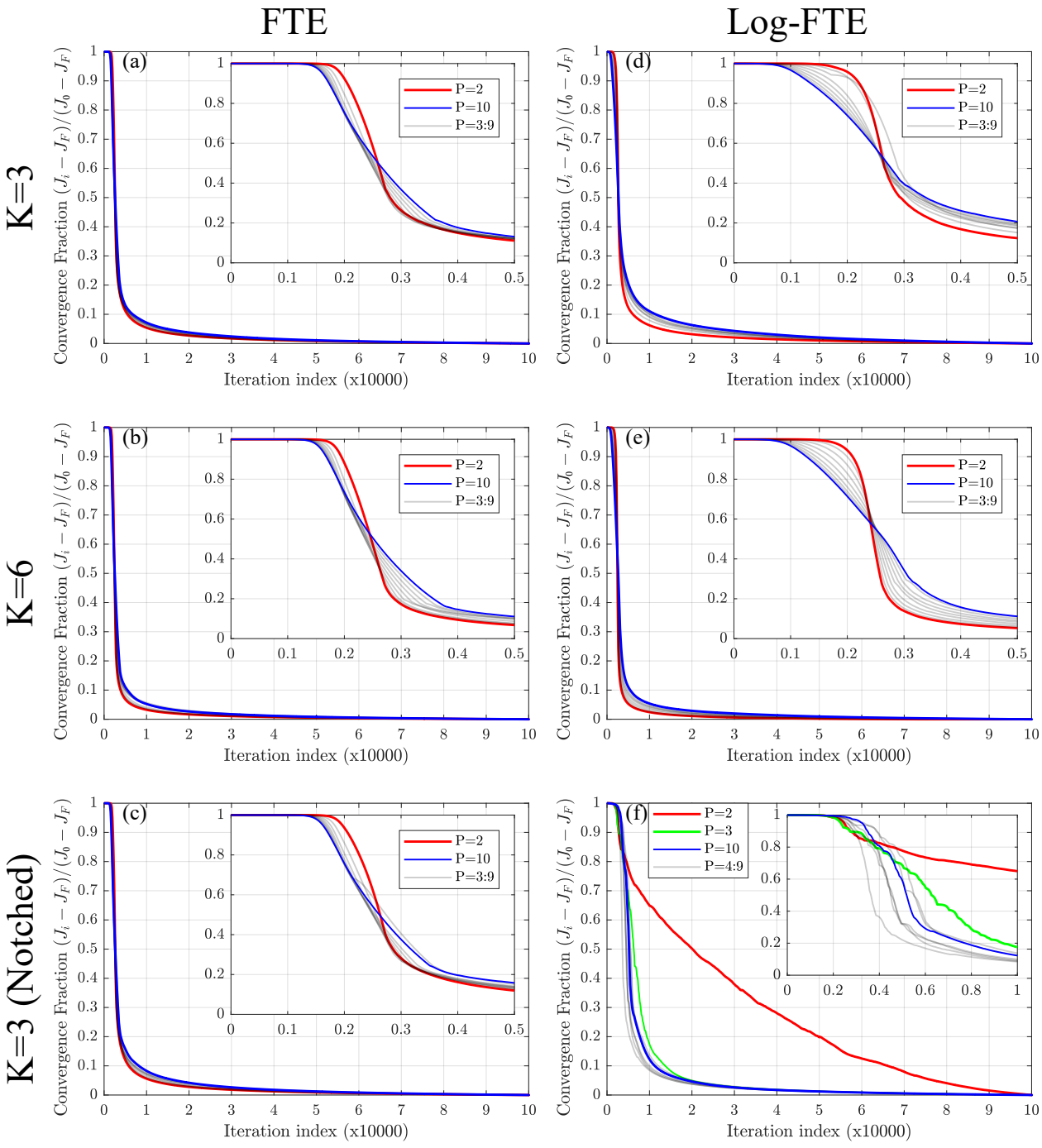

Figure 2.8: Normalized convergence fraction curve for each of the P-norm FTE and Log-FTE optimizations described in Table 2.3.

2.6 FTE and Log-FTE Evaluation

In the previous section, the performance of the FTE and Log-FTE cost functions was analyzed separately as a function of the P-norm value. In this section, the merits of the FTE and Log-FTE cost functions are directly compared to each other and to the PRO-FM waveforms discussed in Section 1.6.2.3, as a basis of comparison to previous FM noise waveforms. Based on the results of the previous section, Table 2.4 summarizes the P-norms used to compare the FTE and Log-FTE cost functions.

Cost function	Template		
	Gaussian ($K = 3$)	Gaussian ($K = 6$)	Notched Gaussian ($K = 3$)
FTE	2	2	2
Log-FTE	2	2	3

Table 2.4: P-norm values used as a function of each template for comparing the FTE and Log-FTE cost functions

In each case, the 2-norm is used with the exception of the Log-FTE cost function with the notched template where the 3-norm was found to provide significantly better performance. The waveforms used in this section are the same ones as in the last section for their respective norms. Sets of 1000 PRO-FM waveforms were additionally optimized for the same templates. In the case of the notched template, the PRO-FM optimization is unable to produce acceptably deep notches. Consequently, a projection based algorithm called Reiterative Uniform Weighting Optimization (RUWO) was additionally applied to the PRO-FM waveforms to improve their notch depth [73].

In Figs. 2.9, 2.10, and 2.11, the different FM noise optimizations are compared directly for the $K = 3$, $K = 6$, and notched $K = 3$ templates respectively. In each figure, the RMS power spectrum, RMS autocorrelation, and coherent autocorrelation are plotted in subplots a, b, and c respectively. In Figs. 2.12, 2.13, and 2.14, the same mean power spectra are plotted respectively, but on a linear scale.

Starting with the $K = 3$ template power spectra results in Fig. 2.9a each optimization scheme has clearly improved the template match significantly relative to the common initialization. However, there is a clear hierarchy of spectral containment. The Log-FTE trace is almost indistinguishable from the template trace all the way to the sample bandwidth while the PRO-FM and FTE traces begin to deviate from the template at roughly $|f/B_{3dB}| > 1$. The FTE then slowly rolls off to about -13 dB in normalized power, while the PRO-FM realizes a template match between these two extremes. However, in the RMS autocorrelation plot, Fig. 2.9b, the FTE cost function provides the best autocorrelation, PRO-FM provides the second best while the Log-FTE waveforms demonstrate the poorest autocorrelation performance. Regardless of their relative performance, each scheme improved over the initialization significantly and even the Log-FTE waveforms beat the initialization by roughly 12 dB near the mainlobe. Finally, the coherent autocorrelation functions in Fig. 2.9c demonstrate that, despite the optimization, the waveforms are still FM noise in that the coherent integrated sidelobe level is roughly 30 dB lower than the RMS autocorrelation levels in each case. Although the variance in the sidelobe levels of the coherently integrated traces make it difficult to say precisely what the improvement was relative to the RMS case. The 30 dB improvement corresponds to $10\log_{10}(M)$, where M is the number of waveforms being integrated which in this case $M = 1000$. This result demonstrates an important characteristic about the FTE and Log-FTE cost functions. The diverse initializations have resulted in diverse solutions indicating the non-convexity of the cost functions.

Moving onto the $K = 6$ template results in Fig. 2.10, we see similar results as the $K = 3$, but to a greater extreme. The difference between the spectral containment at the sample bandwidth of the FTE and Log-FTE cost function is now more 20 dB, while the PRO-FM waveforms achieves a level in between these two, but still closer to the Log-FTE results on a dB scale. In the RMS autocorrelation, the sidelobes for each scheme have increased by a few dB, but this is likely do to the time bandwidth product being cut in half going from 300 to 150. (The number of samples was kept the same for the higher sampling at $N = 900$ samples.) However, the Log-FTE waveform sidelobes went up by about 6 dB, the PRO-FM sidelobe went up about about 3-4 dB, while the FTE

sidelobes only went up about 2-3 dB. In the context of how the FTE and Log-FTE cost functions are effected by the additional low power bins these results seem reasonable. For the FTE cost function, these lower power bins (at the edges of the sample bandwidth of $K = 6$ template) have little impact on the cost function value since it measures the error on a linear scale. For the Log-FTE cost function this is actually much more restrictive as it is virtually required to match the template at all points which is difficult given the square pulse envelope. The penalty here for the significantly improved spectral containment over the FTE cost function is a further increase in the sidelobe level over what could be expected by halving the time bandwidth product ($10log_{10}(2) = 3$). Lastly, like the $K = 3$ template, each set of waveforms experiences a similar reduction in sidelobes of about 30 dB when coherently integrated. Though, the Log-FTE and the initial waveforms due seems to exhibit a higher degree of variance across the coherently integrated autocorrelation.

For the notched $K = 3$ template case in Fig. 2.11, there is a significant spread in the notch depth achieved by each method in Fig. 2.11a. First the PRO-FM+RUWO approach does not try to match any kind of template. Instead it simply attempts to minimize the spectral power in the notched region. It does so very effectively and manages to achieve a normalized notch depth of almost 60 dB relative to the peak template power. The Log-FTE approach manages to match the 42 dB of notch depth of the template while the FTE waveforms only manage about -12 dB which is similar to its spectral containment at the edges of the sample bandwidth. Similar to the other templates, the FTE cost function once again achieves the best RMS autocorrelation sidelobes in Fig. 2.11b while PRO-FM comes in second and Log-FTE comes in third, but all are significantly improved over the initialization. The notch induced shoulder lobes are also present. An interesting aspect of the coherent autocorrelation in this case is that as the sidelobes are pushed further down, more shoulder lobes are revealed and they are not pushed down with the coherent integration. This indicates the notch imposes a correlation affect across all the randomly initialized waveforms since the template is no longer a perfect Gaussian.

From these results several trends are clear. The FTE cost function provides the best autocorrelation performance and the worst spectral containment, the Log-FTE cost function provides the

best spectral containment and the worst autocorrelation performance, and the PRO-FM waveforms fall in between these extremes. From a trade-off standpoint, these results seem reasonable. They indicate that the FTE waveforms sacrifice containment for a better autocorrelation and the Log-FTE waveforms do the opposite. However, recall why the Gaussian template was chosen in the first place in that it theoretically provides an autocorrelation with no sidelobes. Intuitively, the better template match provided by the Log-FTE cost function should result in lower autocorrelation sidelobes. To get an idea of why this is not the case, the RMS spectral results from Figs. 2.9 - 2.11 are plotted again in Figs. 2.12 - 2.14 respectively, but on a linear scale. Additionally, subplots b and c within these figures provide zoomed in versions of the 3dB bandwidth and the roll-off regions respectively. The most striking difference between the linear spectral plots and the dB plots is the amount of variance present in the 3 dB bandwidth of the Log-FTE waveforms. The dB domain plots compress this variance since, relative to the power in these spectral bins, the variance is small. The issue is, relatively speaking, the power in the sidelobes is also small but still significant from a detection standpoint. Thus judging from the dB domain spectral plots and the RMS autocorrelation plots in Figs. 2.9 - 2.11 and the linear spectral plots in Figs. 2.12 - 2.14, for the purpose of minimizing autocorrelation sidelobes, it is more important to match the desired template on a linear scale than on a logarithmic scale. From a spectral containment standpoint, it is more important to match the desired template on a logarithmic scale than on a linear scale. The FTE and Log-FTE cost functions in conjunction with gradient descent optimization methods provide excellent means to do one or the other but not both.

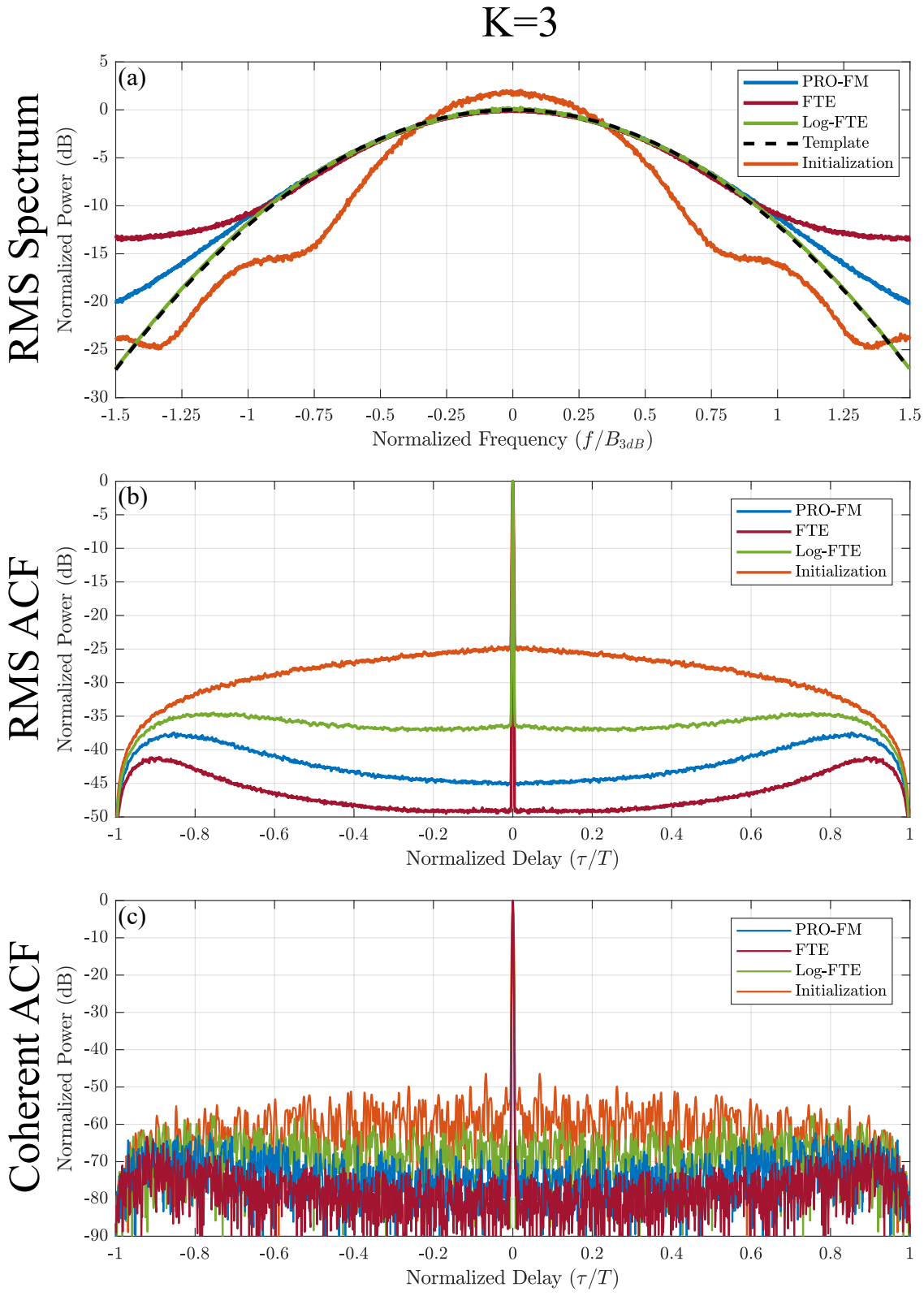

Figure 2.9: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c), for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 3$ template.

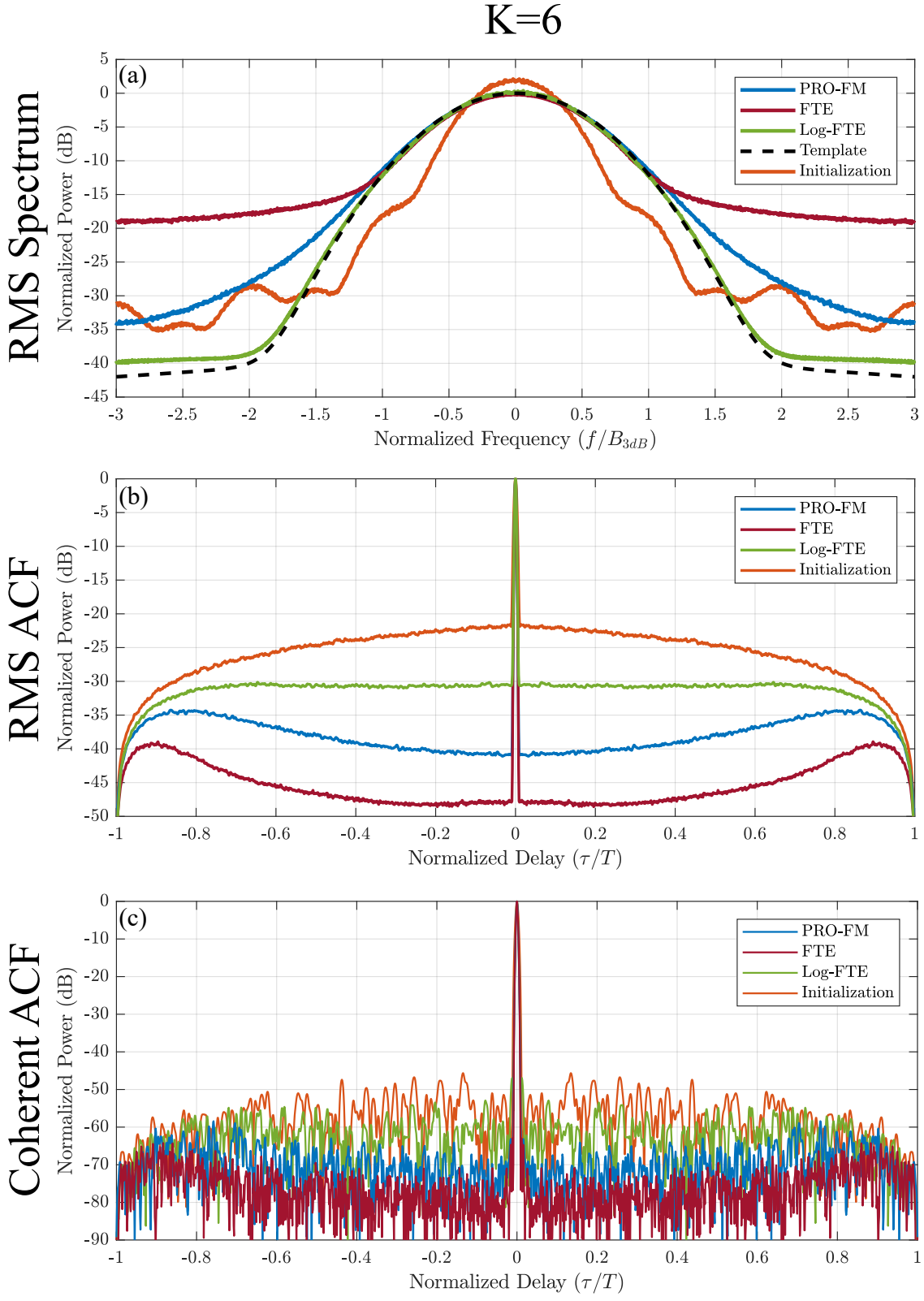

Figure 2.10: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c), for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 6$ template.

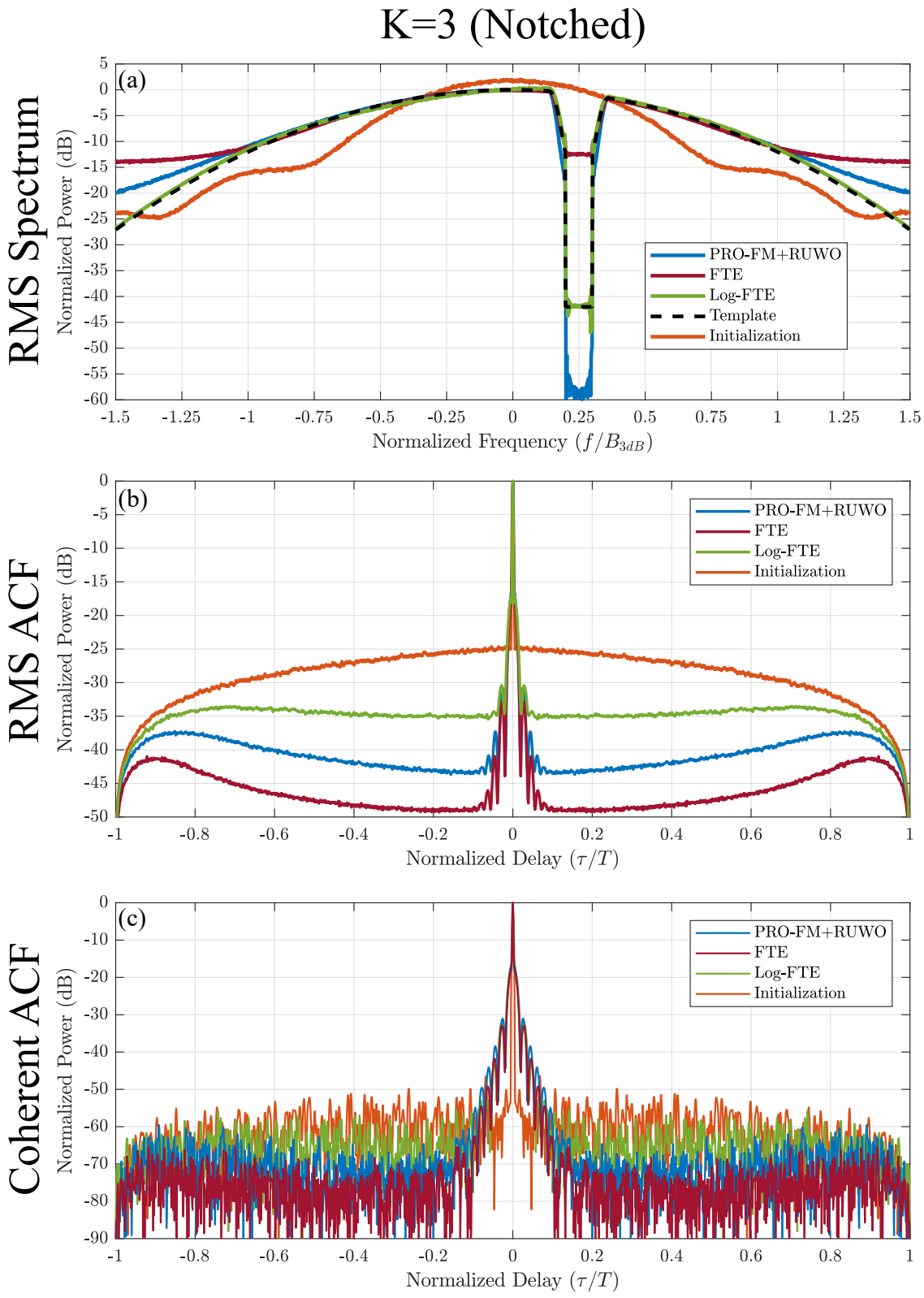

Figure 2.11: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c), for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the notched $K = 3$ template.

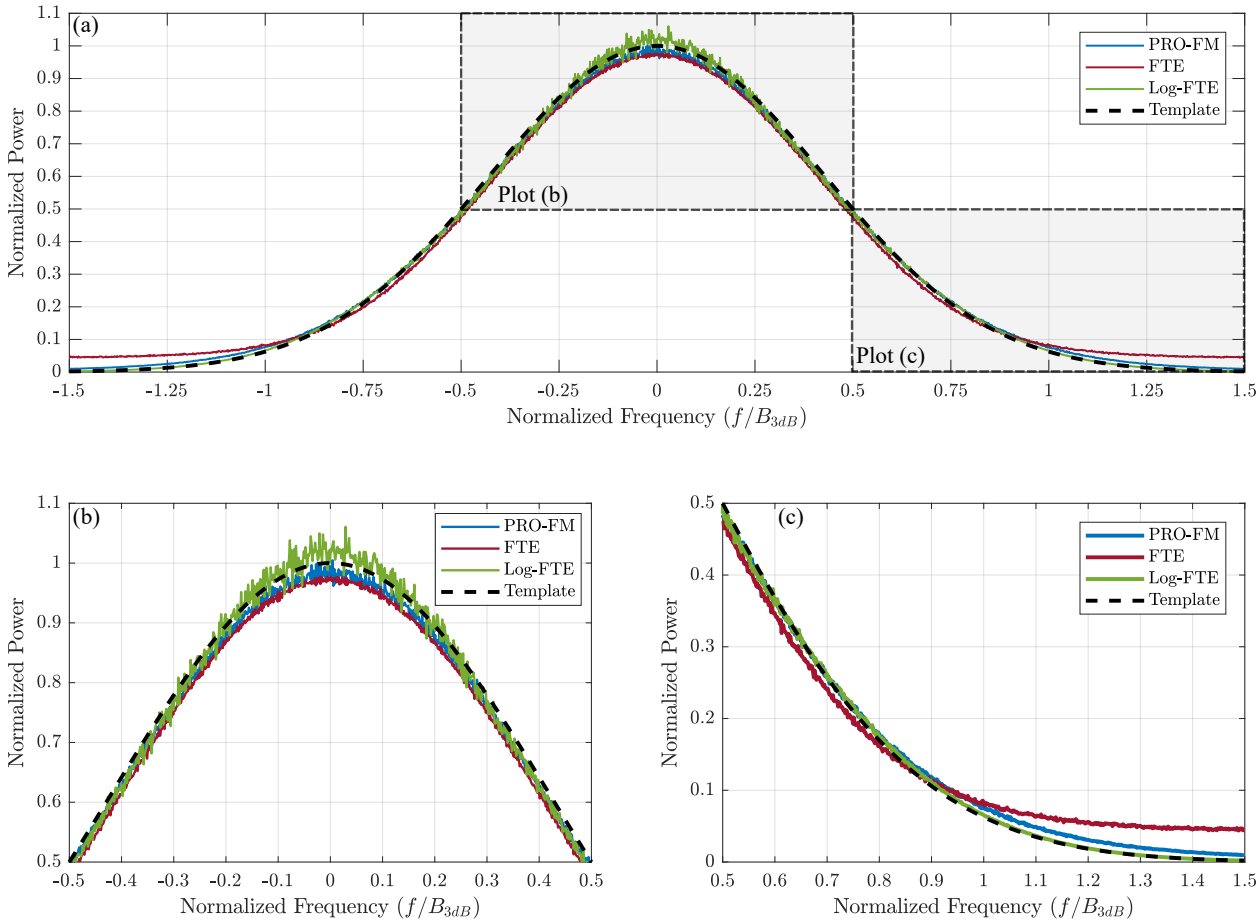

Figure 2.12: RMS spectrum on a linear scale for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 3$ template. entire spectrum (a), passband region (b), roll-off region (c)

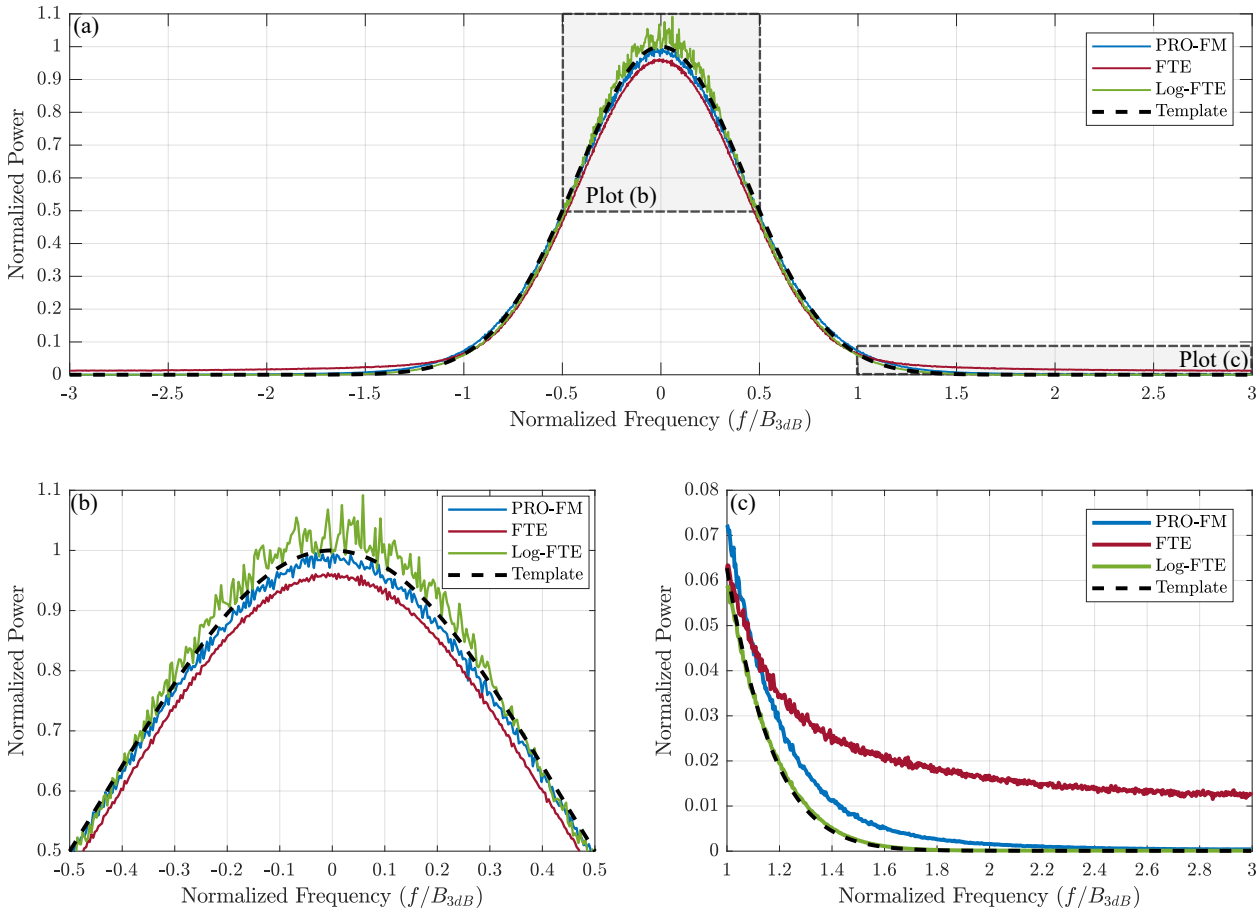

Figure 2.13: RMS spectrum on a linear scale for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 6$ template. entire spectrum (a), passband region (b), portion of the roll-off region (c)

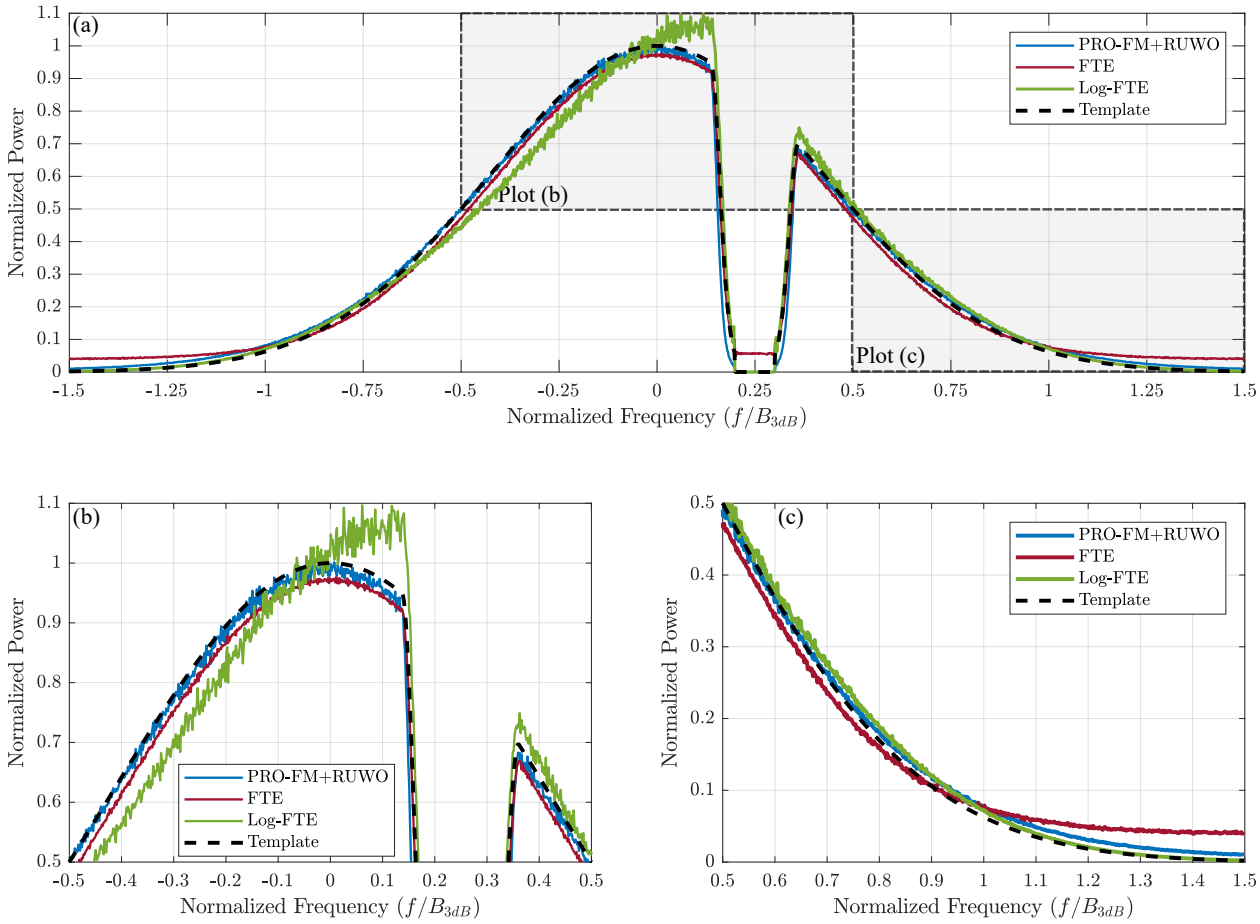

Figure 2.14: RMS spectrum on a linear scale for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the notched $K = 3$ template. entire spectrum (a), passband region (b), roll-off region (c)

2.7 Experimental Loopback Evaluation

The final step in evaluating the FTE and Log-FTE optimized waveforms is to examine their robustness to physical implementation. To do so, the waveforms were interpolated via a spline interpolation of the phase to a sample rate of 10 GSamples/sec and digitally upconverted to a center frequency of 3.55 GHz. They were then implemented on a Tektronix AWG70002A arbitrary waveform generator (AWG), passed through a class A linear amplifier, attenuated and passed through a low noise amplifier before being recorded on a Rhode & Schwarz FSW26 real time spectrum analyzer (RSA) at a sample rate of 200 MSamples/sec. The amplification and attenuation stages were included to emulate a real transmit receive chain. To maintain the time bandwidth products and accommodate the 200 MSamples/sec receiver sample rate of the (RSA) for both the $K = 3$ and the $K = 6$ templates, the bandwidths were 66.6 MHz and 33.3 MHz respectively and the pulse duration was fixed to 4.5 μs in each case. This achieves the desired time bandwidth products of 300 and 150 respectively.

To aid in comparing the simulated results to the loopback measured results, the simulated results from the previous section have been placed to the left of the loopback results in Figs. 2.15 - 2.17 which correspond to the $K = 3$, $K = 6$, and notched $K = 3$ templates respectively. In each figure, subplots a, b, and c correspond to the simulated results while the loopback results correspond to subplots d, e, and f. For clarity, the initialization traces have been removed from the simulation plots.

Beginning with the $K = 3$ loopback results in Fig. 2.15d, there is an extremely sharp attenuation which occurs shortly before the edge of the sample bandwidth (100 MHz). This attenuation is a result of the RSA's anti-aliasing filter. This filtering effect disproportionately attenuates the FTE waveforms as they have the greatest power content at these frequencies. The implications of this can be seen in Figs. 2.15b and 2.15e where the RMS autocorrelation of the FTE waveforms is still the best in loopback, but when comparing it to its simulated counterpart the RMS sidelobes have increased by roughly 4 dB indicated waveform distortion due to band limiting effects. Likewise, the PRO-FM waveforms experienced about a 2 dB sidelobe increase while the Log-FTE sidelobes

increased by maybe a single dB. These results indicate that superior spectral containment leads to superior implementation fidelity.

Figure 2.16 shows similar results but in a more dramatic fashion as the FTE waveforms in this case experience a much larger degree of attenuation at the sample bandwidth edges than either the PRO-FM or especially the Log-FTE waveforms. Although, in terms of total amount of attenuated power, the amount attenuated here is less than in the $K = 3$ case as the $K = 6$ FTE waveforms reach a minimum power of about -20 dB while the $K = 3$ FTE waveforms only reached a minimum of about -13 dB. Accordingly, the FTE RMS autocorrelation is degraded to a lesser degree than in the $K = 3$ case. The sidelobe increase is on the order of 2 dB. The PRO-FM and Log-FTE loopback RMS autocorrelations are almost indistinguishable from their simulated counterparts owing to their superior spectral containment and thus less severe distortion.

Perhaps the most dramatic impact of physical implementation is seen in the case of the notched template in Fig. 2.17. Comparing the loopback spectra in Fig. 2.17d to the simulated spectra in Fig. 2.17a there has been a notable loss in notch depth for the PRO-FM+RUWO waveforms and for the Log-FTE waveforms. The PRO-FM+RUWO waveforms went from about -58 dB in notch depth to roughly -28 dB while the Log-FTE notch went from -42 dB to -32 dB. The FTE waveforms did not lose any notch depth, but then again it was not very good in the first place. This loss in notch depth is not surprising as it has been observed before when implementing PRO-FM+RUWO waveforms [50]. In practice there are effective ways to mitigate this problem. In [72], a second and computationally expensive application of RUWO at passband was applied to preserve notch depth. In [40], an analytical model of the first order PCFM spectra was used to account for the baseband aliasing effects which make the physical realization of deep spectral notches difficult. Here however, it is notable that the Log-FTE waveforms achieved a better notch depth and much less loss overall (compared to their own simulation results) than the PRO-FM+RUWO waveforms without any other optimization. Finally, in terms of the RMS autocorrelations similar effects are seen as in the $K = 3$ template where the FTE experiences about a 4 dB increase in sidelobe level, though here there are the additional notch induced shoulder lobes. As a final note, due to which

waveforms were prepared for testing at the time of loopback experiments, the 2-norm Log-FTE waveforms were implemented in loopback while the 3-norm Log-FTE waveforms where used in the simulation plots. This is the reason for the discrepancy in the RMS Log-FTE sidelobes in simulation and in loopback.

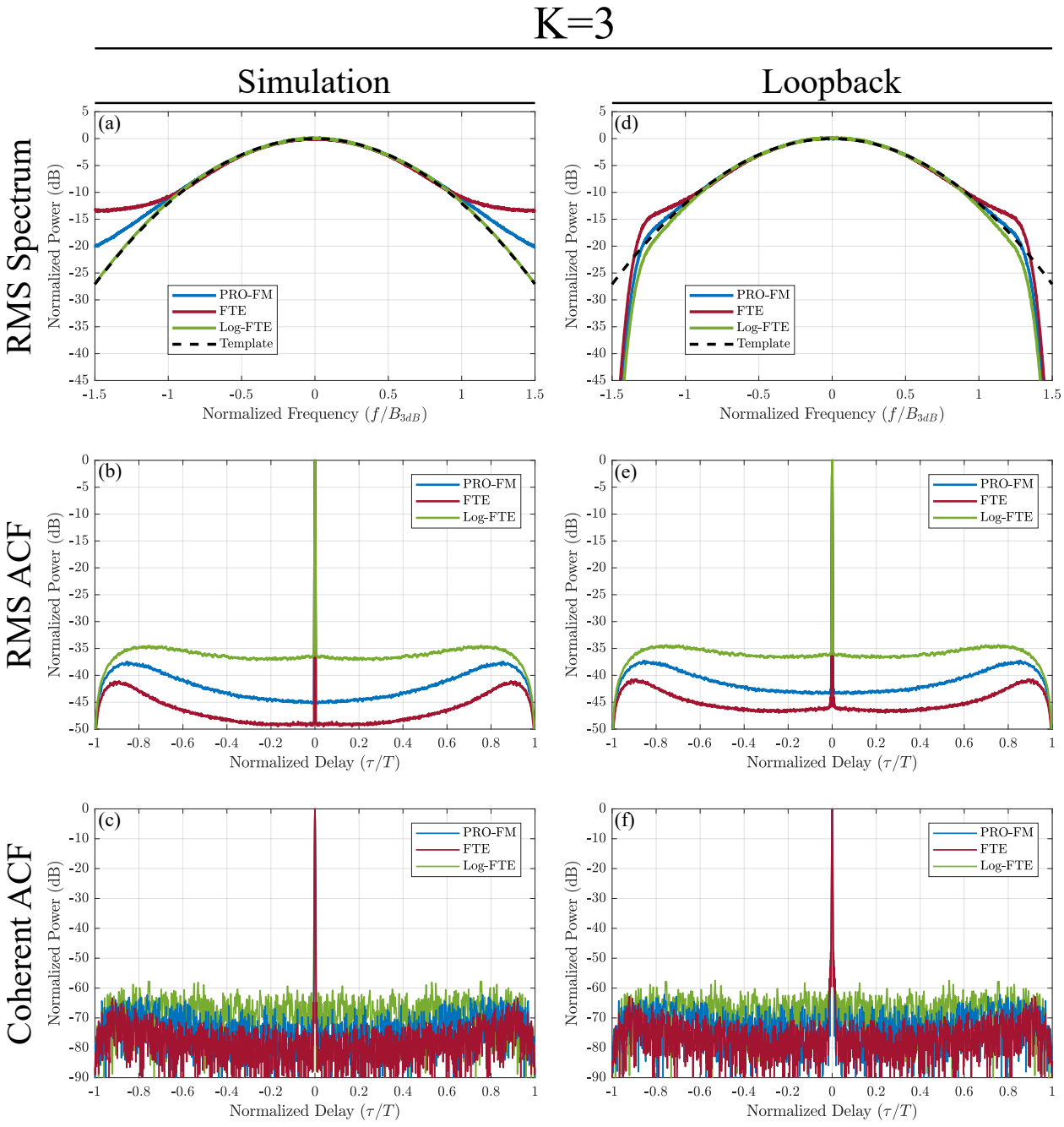

Figure 2.15: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c) in simulation and for experimental loopback results in (d), (e), and (f) respectively, for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 3$ template.

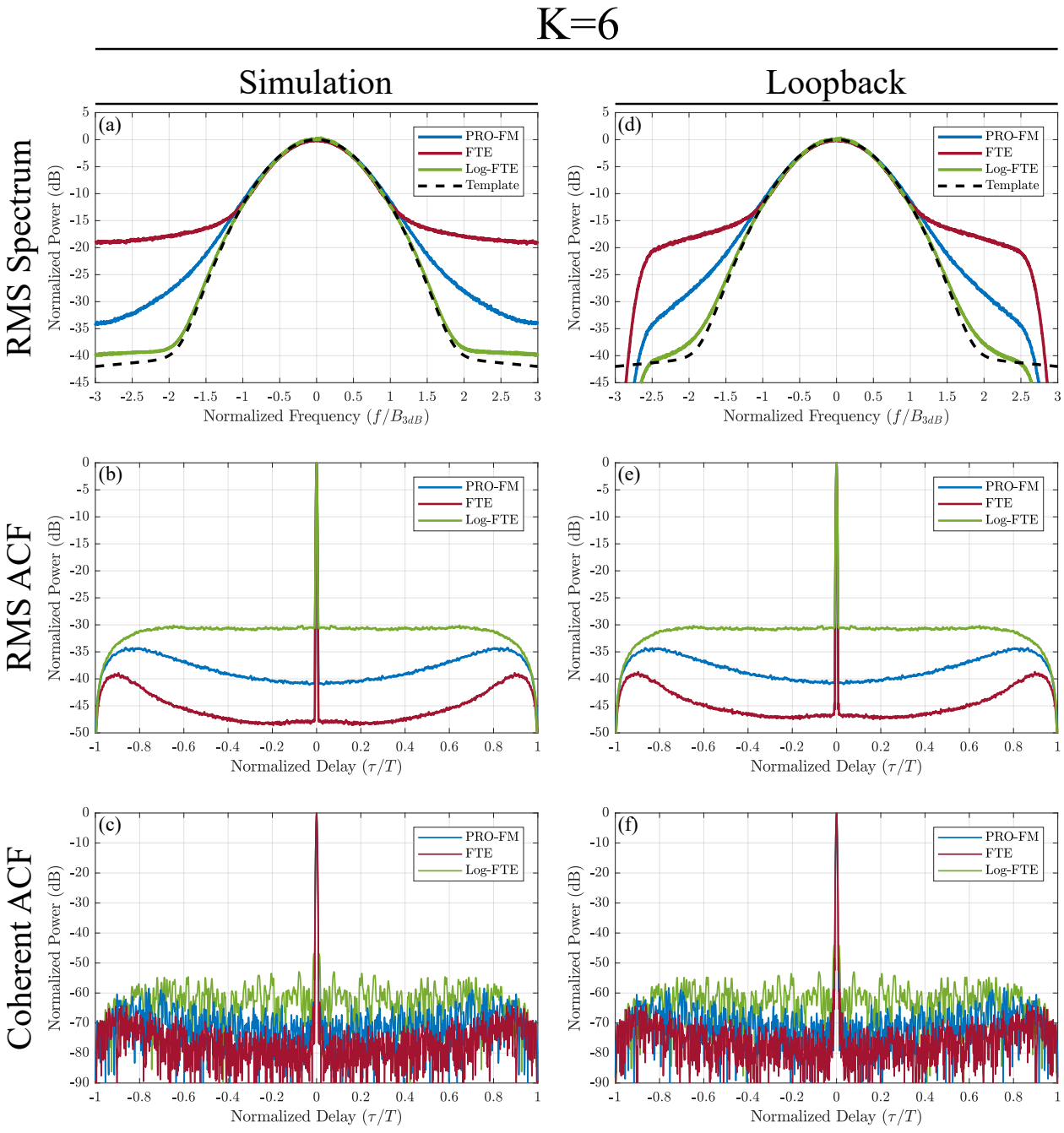

Figure 2.16: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c) in simulation and for experimental loopback results in (d), (e), and (f) respectively, for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the $K = 6$ template.

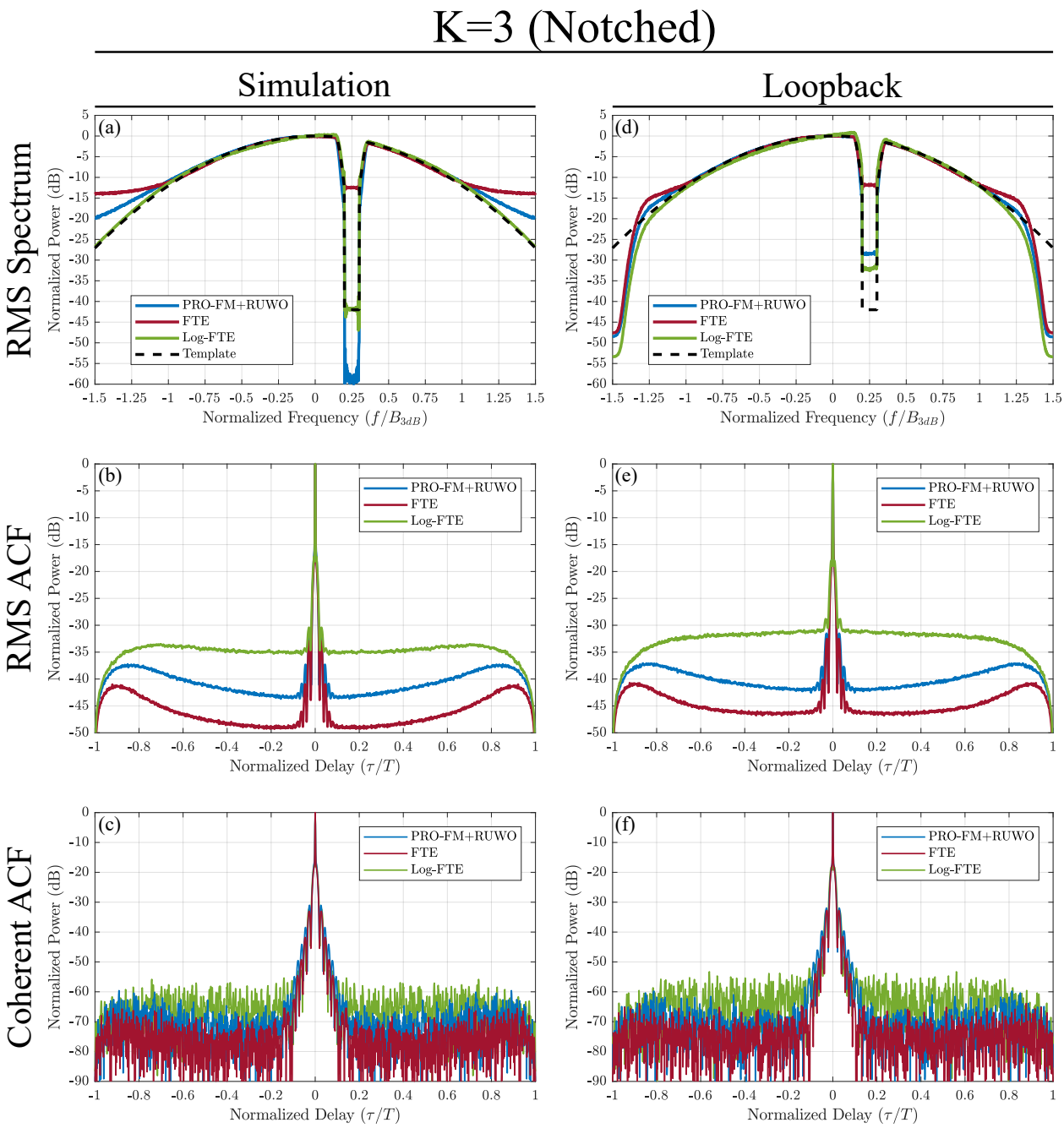

Figure 2.17: RMS spectrum (a), RMS autocorrelation function (b), and coherent autocorrelation function (c) in simulation and for experimental loopback results in (d), (e), and (f) respectively, for each of the 1000 optimized FTE, Log-FTE, and PRO-FM waveforms for the notched $K = 3$ template. (The loopback Log-FTE results used the 2-norm optimized waveforms.)

Chapter 3

Conclusions & Future Work

The constant amplitude envelope, continuous phase, and parameterization of the PCFM waveform structure was found to be convenient for the minimization of specific forms of the generalized frequency template error (GFTE) cost function. Specifically, the frequency template error (FTE) and the logarithmic frequency template error (Log-FTE) cost functions measure the error between a signals power spectrum and some desired spectral template in the linear and logarithmic domains respectively. By minimizing these cost functions using gradient descent methods, candidate waveforms were made to more closely approximate a desired spectrum in either a linear or logarithmic sense. The FTE cost function was found to optimize waveforms to achieve an excellent spectral match within the passband, but it also tended towards poorer spectral containment. The Log-FTE cost function would, to some degree, sacrifice matching the passband frequencies to achieve excellent spectral containment. These behaviors were demonstrated for several "P-norm" variations of both the FTE and Log-FTE cost functions and against Gaussian spectral templates exhibiting different degrees of "oversampling" relative to the 3 dB bandwidth and even a Gaussian template with a notch in the passband. The 2-norm was shown be the ideal norm in all cases but one. Interestingly, when trying to notch a spectrum for the purpose of interference rejection via the Log-FTE, it was shown to be best to use the 3-norm.

These optimization schemes were shown to open up a trade space between autocorrelation sidelobes and spectral containment, as the FTE waveforms exhibit excellent autocorrelation sidelobes and poor spectral containment while the Log-FTE waveforms provide excellent spectral containment with relatively modest sidelobe improvement over the initial waveforms. Additionally, the Log-FTE optimization were shown to be an excellent option for creating waveform with notched

passband spectra such as for interference avoidance. To establish their robustness to implementation the optimized waveforms were measured in loopback and shown to undergo only modest autocorrelation performance degradation with the FTE waveforms showing greater degradation due to their poorer spectral containment and thus susceptibility to filtering effects.

The FTE and Log-FTE optimizations represent a valuable addition to the extensive repertoire of existing waveform optimization schemes as they allow for the more extreme favoring for FM noise waveforms with either excellent autocorrelation sidelobes, excellent spectral containment, or even effective passband notching allowing for the tailoring of waveforms for specific applications. Future work should include considering further forms of the GFTE cost function and a deeper look into the specific structure and ideal optimization schemes of the GFTE cost function in an effort to critically analyze execution time of these optimizations to eventually achieve real time optimization of FM noise waveforms with diverse design goals in operational settings.

Appendix A

Gradient Derivations

A.1 The General Frequency Template Error Gradient

To minimize (2.3), the gradient with respect to the PCFM parameters, $\nabla_{\mathbf{x}} J$, needs to be evaluated. To do this, it is easier to calculate the gradient with respect to the conjugate of the waveform, $\mathbf{s}^*$, first, and then with respect to the PCFM parameters $\mathbf{x}$. Doing so requires the application of the traditional chain rule as well the complex chain rule in (A.1). For a function $g(w) : \mathbb{C} \rightarrow \mathbb{R}$ where $g(w) = g(f(z))$

$$\frac{d}{dz} g(f(z)) = \frac{dg(w)}{dw}\frac{df(z)}{dz} + \frac{dg(w)}{dw^*}\frac{df^*(z)}{dz} \tag{A.1}$$

In other words, the gradient of a real valued function which takes complex valued arguments can be evaluated using the complex chain rule. This is precisely the form of equation 2.3 which takes the complex valued signal as an argument and outputs a real scalar.

To evaluate the gradient, begin by rewriting (2.3) as

$$J = ||G(\mathbf{A}^H \bar{\mathbf{s}}_f) - G(\mathbf{u})||_P = \left(\sum_{k=1}^{2N-1} |G(\mathbf{a}_k^H \mathbf{s}) - G(u_k)|^P \right)^{(1/P)} \tag{A.2}$$

where $\mathbf{a}_k$ is the k_{th} column of the DFT matrix, $\mathbf{A}$. Next, define a set of intermediate variables to facilitate the use of the chain rule and the complex chain rule.

$$w = \sum_{k=1}^{2N-1} w_k \qquad w_k = |G(\mathbf{a}_k^H \mathbf{s}) - G(u_k)|^P \qquad g_k = G(\mathbf{a}_k^H \mathbf{s}) - G(u_k) \tag{A.3}$$

With these terms defined and the application of the chain rule, it is possible to state the gradient of

(2.3) as

$$\nabla_{\mathbf{s}^*} J = \frac{dJ}{dw} \left(\sum_{k=1}^{2N-1} \frac{dw_k}{dg_k} \nabla_{\mathbf{s}^*} g_k \right) \tag{A.4}$$

Evaluation of the term $\nabla_{\mathbf{s}^*} g_k$ requires the complex chain rule as

$$\nabla_{\mathbf{s}^*} g_k = G'(\mathbf{a}_k^H \bar{\mathbf{s}}) \cdot \mathbf{0} + G'_*(\mathbf{a}_k^H \mathbf{s}) \cdot \mathbf{a}_k \tag{A.5}$$

After evaluating the other derivatives with respect to the intermediate variables and putting them into (A.4), the gradient of (2.3) with respect to $\mathbf{s}^*$ becomes

$$\nabla_{\mathbf{s}^*} J = \left(\frac{1}{P} w^{(P-1)/P} \right) \left[\sum_{k=1}^{2N-1} \left(P g_k |g_k|^{(P-2)} \right) \left(G'_*(\mathbf{a}_k^H \mathbf{s}) \cdot \mathbf{a}_k \right) \right] \tag{A.6}$$

Now replace the intermediate variables and simplify.

$$\nabla_{\mathbf{s}^*} J = J^{(1-P)} \left[\sum_{k=1}^{2N-1} [G(\mathbf{a}_k^H \mathbf{s}) - G(u_k)] |G(\mathbf{a}_k^H \mathbf{s}) - G(u_k)|^{(P-2)} \left(G'_*(\mathbf{a}_k^H \mathbf{s}) \cdot \mathbf{a}_k \right) \right] \tag{A.7}$$

The summation can be replaced by a matrix multiply.

$$\nabla_{\mathbf{s}^*} J = J^{(1-P)} \mathbf{A} \left[[G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u})] \odot |G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u})|^{(P-2)} \odot G'_*(\mathbf{A}^H \bar{\mathbf{s}}) \right] \tag{A.8}$$

The last step is to take the gradient with respect to the PCFM parameters. It can be shown the gradient with respect to $\mathbf{x}$ in terms of the signal $\mathbf{s}$ is

$$\nabla_{\mathbf{x}} = 2\mathbf{B}^T \Im \{ s^* \odot \nabla_{s^*} \} \tag{A.9}$$

Finally, by plugging (A.8) into (A.9), the gradient of the GFTE with respect to the PCFM parameters is

$$\nabla_{\mathbf{x}} J = 2\overline{\mathbf{B}}^T \Im \left\{ \bar{\mathbf{s}}^* \odot \mathbf{A} \left([G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u})] \odot \left| G(\mathbf{A}^H \bar{\mathbf{s}}) - G(\mathbf{u}) \right|^{P-2} \odot G'_*(\mathbf{A}^H \mathbf{s}) \right) \right\} J^{(1-P)} \tag{A.10}$$

where $\overline{\mathbf{B}}$ is a zero-padded version of $\mathbf{B}$ as

$$\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{B}^T & \mathbf{0}_{N\times(L-M)} \end{bmatrix}^T \tag{A.11}$$

A.2 Derivatives of selected forms of $G(\cdot)$

For the frequency template error (FTE), $G(\cdot)$ takes the form

$$G_{FTE}(b) = |b|^2 = bb^* \tag{A.12}$$

Where b is a complex number. Then the derivative with respect to b^* using Wirtinger calculus is

$$G'_{FTE}(b) = b \tag{A.13}$$

For the Logarithmic Frequency template error (Log-FTE), $G(\cdot)$ takes the form

$$G_{LogFTE}(b) = \log_a(|b|^2) = \log_a(bb^*) \tag{A.14}$$

Where b is a complex number. Then the derivative with respect to b^* using Wirtinger calculus and the chain rule is

$$G'_{LogFTE}(b) = \frac{1}{\ln a(bb^*)}(b) = \frac{1}{\ln a(b^*)} \tag{A.15}$$

References

[1] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, *Principles of modern radar*. Citeseer, 2010.

[2] K. S. Shanmugan and A. M. Breipohl, *Random signals: detection, estimation, and data analysis*. Wiley, 1988.

[3] G. Turin, "An introduction to matched filters," *IRE Transactions on Information Theory*, vol. 6, pp. 311–329, June 1960.

[4] M. A. Richards, *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.

[5] P. M. Woodward, *Probability and Information Theory, with Applications to Radar: International Series of Monographs on Electronics and Instrumentation*, vol. 3. Elsevier, 2014.

[6] N. Levanon and E. Mozeson, *Radar signals*. John Wiley & Sons, 2004.

[7] J. R. Klauder, A. C. Price, S. Darlington, and W. J. Albersheim, "The theory and design of chirp radars," *The Bell System Technical Journal*, vol. 39, pp. 745–808, July 1960.

[8] J. DiFranco and B. Rubin, *Radar detection*. The Institution of Engineering and Technology, 2004.

[9] D. K. Barton, "Modern radar system analysis," *Norwood, MA, Artech House, 1988, 607 p.*, 1988.

[10] D. P. Meyer and H. A. Mayer, "Radar target detection- handbook of theory and practice," *New York, Academic Press, Inc., 1973. 508 p*, 1973.

[11] M. H. Ackroyd and F. Ghani, "Optimum mismatched filters for sidelobe suppression," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, pp. 214–218, March 1973.

[12] D. Henke, P. McCormick, S. D. Blunt, and T. Higgins, "Practical aspects of optimal mismatch filtering and adaptive pulse compression for fm waveforms," in *2015 IEEE Radar Conference (RadarCon)*, pp. 1149–1155, May 2015.

[13] J. Tsao and B. D. Steinberg, "Reduction of sidelobe and speckle artifacts in microwave imaging: the CLEAN technique," *IEEE Transactions on Antennas and Propagation*, vol. 36, pp. 543–556, April 1988.

[14] S. D. Blunt and K. Gerlach, "Adaptive pulse compression via MMSE estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 572–584, April 2006.

[15] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, pp. 51–83, Jan 1978.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice Hall Press, 2009.

[17] B. M. Horton, "Noise-modulated distance measuring systems," *Proceedings of the IRE*, vol. 47, pp. 821–828, May 1959.

[18] G. H. Liu Guosui and S. Weimin, "Development of random signal radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, pp. 770–777, July 1999.

[19] M. Golay, "Complementary series," *IRE Transactions on Information Theory*, vol. 7, pp. 82–87, April 1961.

[20] R. Sivaswamy, "Digital and analog subcomplementary sequences for pulse compression," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, pp. 343–350, March 1978.

[21] S. D. Blunt and E. L. Mokole, "Overview of radar waveform diversity," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, pp. 2–42, November 2016.

[22] H. Griffiths, L. Cohen, S. Watts, E. Mokole, C. Baker, M. Wicks, and S. Blunt, "Radar spectrum engineering and management: Technical and regulatory issues," *Proceedings of the IEEE*, vol. 103, pp. 85–102, Jan 2015.

[23] F. H. Raab, P. Asbeck, S. Cripps, P. B. Kenington, Z. B. Popovic, N. Pothecary, J. F. Sevic, and N. O. Sokal, "Power amplifiers and transmitters for RF and microwave," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, pp. 814–826, Mar 2002.

[24] F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," *IEEE Microwave Magazine*, vol. 10, pp. 52–64, Dec 2009.

[25] R. Baker, "Group synchronization of binary digital systems," *Communication Theory, Willis and Jackson (Butterworth's Scientific Publications, 1953)*, 1953.

[26] G. Coxson and J. Russo, "Efficient exhaustive search for optimal-peak-sidelobe binary codes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 302–308, Jan 2005.

[27] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proceedings of the IEEE*, vol. 64, pp. 1715–1729, Dec 1976.

[28] L. Bomer and M. Antweiler, "Polyphase barker sequences," *Electronics Letters*, vol. 25, pp. 1577–1579, Nov 1989.

[29] R. Frank, "Polyphase codes with good nonperiodic correlation properties," *IEEE Transactions on Information Theory*, vol. 9, pp. 43–45, January 1963.

[30] B. L. Lewis and F. F. Kretschmer, "A new class of polyphase pulse compression codes and techniques," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-17, pp. 364–372, May 1981.

[31] B. L. Lewis and F. F. Kretschmer, "Linear frequency modulation derived polyphase pulse compression codes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, pp. 637–641, Sep. 1982.

[32] H. H. Faust, B. Connolly, T. M. Firestone, R. C. Chen, B. H. Cantrell, and E. L. Mokole, "A spectrally clean transmitting system for solid-state phased-array radars," in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509)*, pp. 140–144, April 2004.

[33] J. W. Taylor and H. J. Blinchikoff, "Quadriphase code-a radar pulse compression signal with unique characteristics," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, pp. 156–170, March 1988.

[34] S. D. Blunt, M. Cook, J. Jakabosky, J. D. Graaf, and E. Perrins, "Polyphase-coded FM (PCFM) radar waveforms, part I: implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, pp. 2218–2229, July 2014.

[35] S. Blunt, M. Cook, E. Perrins, and J. de Graaf, "CPM-based radar waveforms for efficiently bandlimiting a transmitted spectrum," in *2009 IEEE Radar Conference*, pp. 1–6, May 2009.

[36] B. S. I. Group, "Specifications of the bluetooth system," 2003.

[37] P. S. Tan, J. Jakabosky, J. M. Stiles, and S. D. Blunt, "On higher-order representations of polyphase-coded FM radar waveforms," in *2015 IEEE Radar Conference (RadarCon)*, pp. 0467–0472, May 2015.

[38] C. A. Mohr, P. M. McCormick, S. D. Blunt, and C. Mott, "Spectrally-efficient FM noise radar waveforms optimized in the logarithmic domain," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0839–0844, April 2018.

[39] C. A. Mohr, P. M. McCormick, and S. D. Blunt, "Optimized complementary waveform subsets within an FM noise radar CPI," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0687–0692, April 2018.

[40] C. A. Mohr and S. D. Blunt, "Analytical spectrum representation for physical waveform optimization requiring extreme fidelity," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0839–0844, April 2019.

[41] S. D. Blunt, J. Jakabosky, M. Cook, J. Stiles, S. Seguin, and E. L. Mokole, "Polyphase-coded FM (PCFM) radar waveforms, part II: optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, pp. 2230–2241, July 2014.

[42] P. M. McCormick and S. D. Blunt, "Nonlinear conjugate gradient optimization of polyphase-coded FM radar waveforms," in *2017 IEEE Radar Conference (RadarConf)*, pp. 1675–1680, May 2017.

[43] J. Jakabosky, P. Anglin, M. R. Cook, S. D. Blunt, and J. Stiles, "Non-linear FM waveform design using Marginal Fisher's Information within the CPM framework," in *2011 IEEE Radar-Con (RADAR)*, pp. 513–518, May 2011.

[44] J. Jakabosky, S. D. Blunt, M. R. Cook, J. Stiles, and S. A. Seguin, "Transmitter-in-the-loop optimization of physical radar emissions," in *2012 IEEE Radar Conference*, pp. 0874–0879, May 2012.

[45] J. Owen, C. A. Mohr, S. D. Blunt, and K. Gallagher, "Nonlinear radar via intermodulation of jointly optimized fm noise waveform pairs," in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–6, Apr 2019.

[46] J. Jakabosky, S. D. Blunt, and B. Himed, "Optimization of "over-coded" radar waveforms," in *2014 IEEE Radar Conference*, pp. 1460–1465, May 2014.

[47] J. Jakabosky, S. D. Blunt, and B. Himed, "Waveform design and receive processing for non-recurrent nonlinear FMCW radar," in *2015 IEEE Radar Conference (RadarCon)*, pp. 1376–1381, May 2015.

[48] J. Jakabosky, S. D. Blunt, and B. Himed, "Spectral-shape optimized FM noise radar for pulse agility," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–6, May 2016.

[49] J. Jakabosky, S. D. Blunt, and A. Martone, "Incorporating hopped spectral gaps into non-recurrent nonlinear FMCW radar emissions," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 281–284, Dec 2015.

[50] J. Jakabosky, B. Ravenscroft, S. D. Blunt, and A. Martone, "Gapped spectrum shaping for tandem-hopped radar/communications cognitive sensing," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–6, May 2016.

[51] J. W. Owen, B. Ravenscroft, B. H. Kirk, S. D. Blunt, C. T. Allen, A. F. Martone, K. D. Sherbondy, and R. M. Narayanan, "Experimental demonstration of cognitive spectrum sensing notching for radar," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0957–0962, April 2018.

[52] B. Ravenscroft, P. M. McCormick, S. D. Blunt, J. Jakabosky, and J. G. Metcalf, "Tandem-hopped OFDM communications in spectral gaps of FM noise radar," in *2017 IEEE Radar Conference (RadarConf)*, pp. 1262–1267, May 2017.

[53] B. Ravenscroft, P. M. McCormick, S. D. Blunt, E. Perrins, and J. G. Metcalf, "A power-efficient formulation of tandem-hopped radar communications," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 1061–1066, April 2018.

[54] B. Ravenscroft, P. McCormick, S. D. Blunt, E. Perrins, C. Sahin, and J. Metcalf, "Experimental assessment of tandem-hopped radar and communications (THoRaCs)," in *2019 International Conference on Radar (RADAR)*, pp. 1–6, Sep 2019.

[55] G. Zook, P. M. McCormick, S. D. Blunt, C. Allen, and J. Jakabosky, "Dual-polarized FM noise radar," in *International Conference on Radar Systems (Radar 2017)*, pp. 1–5, Oct 2017.

[56] J. Owen, S. D. Blunt, K. Gallagher, P. McCormick, C. Allen, and K. Sherbondy, "Nonlinear radar via intermodulation of FM noise waveform pairs," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0951–0956, April 2018.

[57] G. Zook, P. McCormick, and S. D. Blunt, "Fixational eye movement radar: Random spatial modulation," in *2018 IEEE Radar Conference (RadarConf18)*, pp. 0827–0832, April 2018.

[58] C. A. Mohr and S. D. Blunt, "FM noise waveforms optimized according to a temporal template error (TTE) metric," in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–6, Apr 2019.

[59] C. A. Mohr and S. D. Blunt, "Design and generation of stochastically defined, pulsed FM noise waveforms," in *2019 International Conference on Radar (RADAR)*, pp. 1–6, Sep 2019.

[60] C. Sahin, J. Jakabosky, P. M. McCormick, J. G. Metcalf, and S. D. Blunt, "A novel approach for embedding communication symbols into physical radar waveforms," in *2017 IEEE Radar Conference (RadarConf)*, pp. 1498–1503, May 2017.

[61] C. Sahin, J. G. Metcalf, A. Kordik, T. Kendo, and T. Corigliano, "Experimental validation of phase-attached radar/communication (parc) waveforms: Radar performance," in *2018 International Conference on Radar (RADAR)*, pp. 1–6, Aug 2018.

[62] P. McCormick, C. Sahin, S. D. Blunt, and J. G. Metcalf, "FMCW implementation of phase-attached radar-communications (PARC)," in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–6, Apr 2019.

[63] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[64] D. G. Luenberger, Y. Ye, *et al.*, *Linear and nonlinear programming*, vol. 2. Springer, 1984.

[65] J. Nocedal and S. J. Wright, *Nonlinear Equations*. Springer, 2006.

[66] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, vol. 49. NBS Washington, DC, 1952.

[67] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.

[68] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.

[69] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson, "Global convergence of the heavy-ball method for convex optimization," in *2015 European Control Conference (ECC)*, pp. 310–315, IEEE, 2015.

[70] B. O'Donnell and J. M. Baden, "Fast gradient descent for multi-objective waveform design," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–5, May 2016.

[71] D. Zhao, Y. Wei, and Y. Liu, "Spectrum optimization via FFT-based conjugate gradient method for unimodular sequence design," *Signal Processing*, vol. 142, pp. 354–365, 2018.

[72] B. Ravenscroft, J. W. Owen, J. Jakabosky, S. D. Blunt, A. F. Martone, and K. D. Sherbondy, "Experimental demonstration and analysis of cognitive spectrum sensing and notching for radar," *IET Radar, Sonar Navigation*, vol. 12, no. 12, pp. 1466–1475, 2018.

[73] T. Higgins, T. Webster, and A. K. Shackelford, "Mitigating interference via spatial and spectral nulls," in *IET International Conference on Radar Systems (Radar 2012)*, pp. 1–6, Oct 2012.