

Higher-order Implementations of Polyphase-Coded FM Radar Waveforms

Peng Seng Tan, *Member, IEEE*, John Jakobosky, *Member, IEEE*,
James M. Stiles, *Senior Member, IEEE*, Shannon D. Blunt, *Fellow, IEEE*

Abstract—The recently developed polyphase-code FM (PCFM) implementation for physical radar waveforms is generalized to higher-order representations to facilitate greater design freedom. Being FM, waveforms realized with these implementations have the benefit of being readily amenable to a high-power radar transmitter while possessing parameterized structures that are advantageous for optimization. Here various attributes of these implementations are examined. Specifically, it is shown that higher-order representations can, in special cases, be made equivalent, and through these relationships appropriate signal structure attributes can be inferred. Higher-order coding guidelines are also derived based on the need to ensure spectral containment. Example waveforms are optimized for each particular implementation to highlight their individual properties, towards the ultimate goal of establishing new ways to realize waveform-diverse emission structures that are physically realizable.

Index Terms—radar waveforms, continuous phase modulation, nonlinear FM, waveform implementation, waveform diversity

I. INTRODUCTION

Frequency modulation (FM) represents one of the earliest [2, 3] and by far most widely used means of generating a radar waveform for use in pulse compression. After the establishment of the linear frequency modulated (LFM) chirp, the prospective benefits of nonlinear FM (NLFM) waveforms were realized, followed by a litany of important contributions (e.g. [4-12], many of which are summarized in [13, 14]). In short, FM waveforms are attractive because they can be generated in hardware with very wide bandwidths (particularly LFM), are constant amplitude, and are well-contained spectrally, thus making them readily amenable to high-power radar transmitters. Further, in the case of LFM the benefits of stretch processing [15] can be leveraged on receive (with recent work [16] demonstrating how

Peng Seng Tan is now with Eaton Cooper Lighting Corporate Campus, 1121 Hwy 74 S, Peachtree City, GA 30269 USA (e-mail: isaactan@eaton.com).

John Jakobosky is now with the Radar Division – U.S. Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC, 20375.

James M. Stiles and Shannon D. Blunt are with the Electrical Engineering and Computer Science Department, University of Kansas, 2335 Irving Hill Road, Lawrence, KS, 66045.

This work was supported by the US Air Force Office of Scientific Research. A portion of this work appeared at the 2015 IEEE International Radar Conference [1].

chirp-like NLFM can employ stretch processing as well).

A separate class of waveforms that has attracted significant attention is that of phase-coded sequences (e.g. [17-20]), particularly with the emergence of waveform diversity [14, 21-23] and the prospect of incorporating the dimensions of space (azimuth and elevation), Doppler, polarization, and frequency into the waveform design process in new ways. While not directly implementable without distortion (see [14]), phase codes are very important because they represent the means with which to parameterize the structure of a signal in a way that can be optimized, an attribute that was not readily available for traditional FM waveform design [2-10].

It has recently been shown that a modified form [24,25] of the continuous phase modulation (CPM) scheme used in some communication applications [26] can be employed to make the connection between the mathematically attractive structure of phase codes to the physically realizable structure of FM waveforms. Denoted as polyphase-coded FM (PCFM) [25], this framework facilitates the conversion of an arbitrary polyphase code into an FM waveform that can be readily transmitted by a high-power radar. More importantly, this approach enables the direct design of FM waveforms through optimization of the underlying parameterized code structure based on desired properties of the resulting FM waveform (i.e. the code-to-waveform implementation is part of the design process) [27]. In so doing, undesirable distortion-inducing effects of the transmitter (most notably by the power amplifier) can also be incorporated into the waveform design process for at least partial compensation [27,28], and spatial and polarization degrees-of-freedom can be physically coupled to waveforms [29-32]. It has recently even been shown that the PCFM implementation enables new forms of radar-embedded communication [33,34], the design of FM waveforms via gradient descent optimization of the coded parameters [35], the collective optimization of spectrally disjoint FM waveforms [36], the optimization of a particular form of FM noise waveform [37], and even the optimization and experimental demonstration of complementary FM waveforms [38] that are more robust to Doppler and transmitter distortion than traditional complementary codes. The very structure of PCFM itself has also been examined via the notion of “over-coding” [39], higher-order implementations [1], and through the use of Legendre polynomials as alternative phase basis functions [40].

In short, while there has been a tremendous amount of work on waveform design (see [13, 14, 21-23] and numerous references therein), it has only been recently that a *parametrized FM waveform implementation* [25] has been available with which to explore the myriad ways in which physical waveform diversity could be achieved. Because waveform diversity represents a vital enabler to address problems in radar spectrum engineering and

containment [41], radar spectrum sharing and multi-function operation [42], and cognitive radar [43, 44], it is beneficial to determine the ways in which useful waveforms can be optimized and physically generated with as much design freedom as possible. As such, here we provide a detailed examination of the higher-order PCFM implementations initially described in [1] that were inspired by the polynomial NLFM waveforms proposed by Doerry [11]. This examination involves exploration of the relationships between different orders, deriving guidelines for permissible higher-order code values based on spectral containment, and exemplifying the performance that can be achieved when optimizing waveforms realized by the different PCFM implementation structures.

Table I describes an order-based categorization of constant amplitude, code-parameterized radar waveforms, where the phase codes (e.g. [17-20]) correspond to a zeroth-order representation since they possess constant phase chips that undergo abrupt transitions between chips. The PCFM scheme described in [25] and subsequently employed in various ways in [27-39, 45-48] can then be viewed as a first-order representation since the use of a rectangular shaping filter produces piece-wise linear phase trajectories following the single integration stage. The well-known LFM waveform, along with waveforms constructed in a piecewise manner from LFM waveforms having different chirp-rates, are specific examples of a second-order representation that correspond to quadratic phase trajectories. A parameterized second-order PCFM implementation will likewise be defined here. Finally, third-order and higher representations are also possible, with a parameterized third-order PCFM implementation also presented here.

Table I. Waveform Representations (phase structure)

Waveform representation	Equivalent approaches in radar waveform generation / phase structure
0 th order (not differentiable)	Phase codes / abrupt phase transitions
1 st order (differentiable)	PCFM via [25] / piece-wise linear phase trajectories
2 nd order (twice differentiable)	LFM and piecewise NLFM / quadratic phase trajectories
3 rd order & higher (3+ derivatives exist)	Higher order NLFM / higher order phase trajectories

Given a finite pulsewidth and practical spectral roll-off – noting that the spectral support is theoretically infinite for a finite pulsewidth – the possible phase trajectories of a continuous waveform is limited only by the precision with which the waveform can be physically generated. Consequently, these additional PCFM implementations, and combinations thereof, provide an expansion of the number of possible waveforms that can be physically realized. This increased design freedom therefore represents a larger design space within which to construct waveform-diverse

emission structures such as those discussed above and beyond.

II. HIGHER-ORDER PCFM IMPLEMENTATIONS

In [25], it was shown how the CPM implementation that has been used to provide power efficient and spectrally efficient communications [26] could be modified to enable the conversion of arbitrary polyphase codes (specifically, *zeroth-order codes*) into physically realizable FM radar waveforms. Such waveforms may also be directly optimized by incorporating the code-to-waveform implementation into the design process [27,28,35-38]. Inspired by [11], we hereby expand this code-based FM structure to facilitate new ways in which such waveforms can be implemented, with the ultimate goal of establishing potential new avenues for physical waveform optimization.

Due to the nature of this code-to-waveform implementation, the polyphase-coded FM (PCFM) scheme [25] corresponds to a first-order representation in which the phase function of waveform $s(t) = \exp\{j\phi_1(t)\}$ can be expressed as

$$\phi_1(t) = \int_0^t \left[\sum_{n=1}^N a_n g_1(t' - (n-1)T_p) \right] dt' + \bar{\phi}_1, \quad (1)$$

where the set of phase-change values a_n for $n=1, 2, \dots, N$ constitute a *first-order code* (which may or may not be derived from a zeroth-order code of length $N+1$ per [25]) that produces a continuous waveform of pulsewidth T . The term $g_1(t)$ is a shaping filter that integrates to unity over the time support $[0, T_p]$ for $T_p = T/N$, and $\bar{\phi}_1$ is the initial phase of the waveform. If $g_1(t)$ is a rectangular filter, the phase function in (1) is piece-wise linear.

Let the continuous, first-order coded function inside the brackets of (1) be

$$\chi_1(t) = \sum_{n=1}^N a_n g_1(t - (n-1)T_p), \quad (2)$$

which represents the time-varying frequency of the waveform. Thus (1) can also be written as

$$\phi_1(t) = \int_0^t \chi_1(t') dt' + \bar{\phi}_1. \quad (3)$$

From [25], the first-order phase function of (1)-(3) can be implemented as shown in Figure 1 (with the MATLAB™ script provided in Appendix A). The first-order code a_n represents the (normalized) time-varying frequency with permissible values in $[-\pi, \pi]$. Because $g_1(t)$ integrates to unity over $[0, T_p]$, the maximum phase change in T_p seconds is $\pm\pi$. Thus the 3 dB bandwidth is

$$\pm \frac{\pi}{T_p} = \pm \frac{\pi N}{T} \cong \pm \frac{\pi BT}{T} = \pm \pi B \text{ radians/s,} \quad (4)$$

or $\pm B/2$ Hz (at baseband), where we have used the relationship $T = NT_p$ and the fact that the time-bandwidth product BT is well approximated by N . Note that the permissible region of $[-\pi, \pi]$ for first-order code values can be expanded to provide greater design freedom as long as appropriate spectral containment measures are also enforced to prevent expansion of the aggregate spectral content (see [38]).

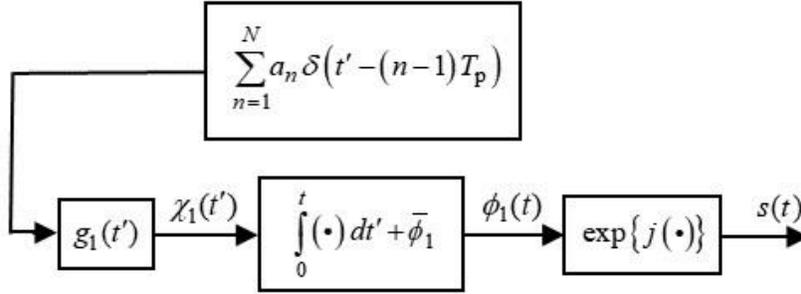


Fig. 1. First-order implementation of polyphase-coded FM (PCFM) waveforms (see Appendix A for MATLABTM script)

A. Formulation of second/third-order PCFM implementations

By using the same format as (3), and taking inspiration from the polynomial NLFM formulation of [11], generalization to second-order and third-order waveform phase functions can be expressed as

$$\phi_2(t) = \int_0^t \int_0^{t'} \chi_2(t'') dt'' dt' + \int_0^t \bar{\omega}_2 dt' + \bar{\phi}_2 \quad (5)$$

and

$$\phi_3(t) = \int_0^t \int_0^{t'} \int_0^{t''} \chi_3(t''') dt''' dt'' dt' + \int_0^t \int_0^{t'} \bar{\beta}_3 dt'' dt' + \int_0^t \bar{\omega}_3 dt' + \bar{\phi}_3, \quad (6)$$

respectively, where $\bar{\phi}_2$ and $\bar{\omega}_2$ are the second-order initial phase and frequency, and $\bar{\phi}_3$, $\bar{\omega}_3$, and $\bar{\beta}_3$ are the third-order initial phase, frequency, and chirp-rate. Like the first-order coded function $\chi_1(t)$ in (2), the second-order coded function from (5) can be defined as

$$\chi_2(t) = \sum_{n=1}^N b_n g_2(t - (n-1)T_p) \quad (7)$$

and the third-order coded function from (6) as

$$\chi_3(t) = \sum_{n=1}^N c_n g_3(t - (n-1)T_p). \quad (8)$$

In (7), the *second-order code* b_n for $n=1, 2, \dots, N$ represents a time-varying chirp-rate. Likewise, the *third-order code* c_n for $n=1, 2, \dots, N$ in (8) represents a time-varying *chirp-acceleration*. As with the first-order formulation, $g_2(t)$ and $g_3(t)$ are shaping filters defined on the interval $[0, T_p]$. Imposing the same bandwidth as determined by (4) onto these higher-order implementations requires that the compounding effect of the additional integration stages be taken into account, which impacts the selection of the coding values b_n and c_n as well as the associated shaping filters $g_2(t)$ and $g_3(t)$. Also note that b_n , c_n , $\bar{\omega}_2$, $\bar{\omega}_3$, and $\bar{\beta}_3$ are in angular units (i.e. scaled by 2π), with the permissible initial frequencies $\bar{\omega}_2, \bar{\omega}_3 \in [-\pi/T_p, +\pi/T_p]$ and chirp-rates $\bar{\beta}_3 \in [-2\pi/NT_p^2, +2\pi/NT_p^2]$.

Figures 2 and 3 illustrate the implementation of these second-order and third-order phase functions (with MATLABTM scripts provided in Appendix A). Clearly, fourth-order and higher phase functions could also be formulated in this manner, though such have not been found to be beneficial with regard to generating useful radar waveforms. In fact, it will be shown that while the second-order formulation tends to facilitate the design of waveforms with marked sidelobe level improvement relative to the first-order implementation for a commensurate BT , the same cannot be said for the third-order scheme, which thus far has been found to provide only a rather modest benefit when used in combination with the first and second orders.

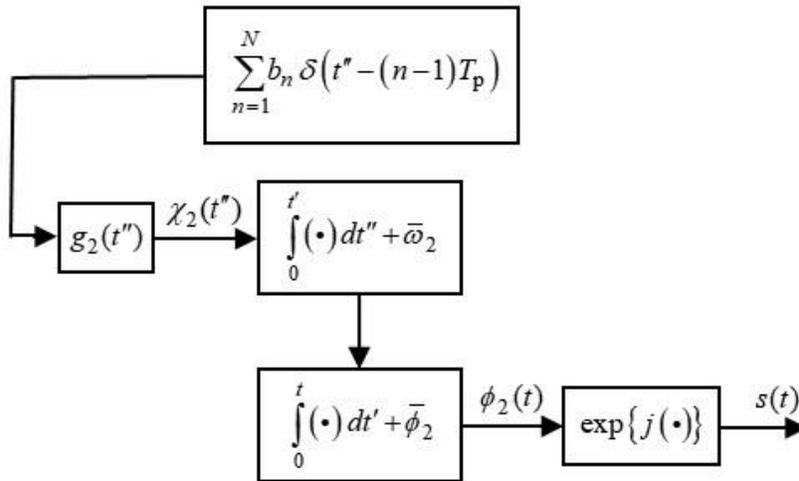


Fig. 2. Second-order PCFM waveform implementation
(see Appendix A for MATLABTM script)

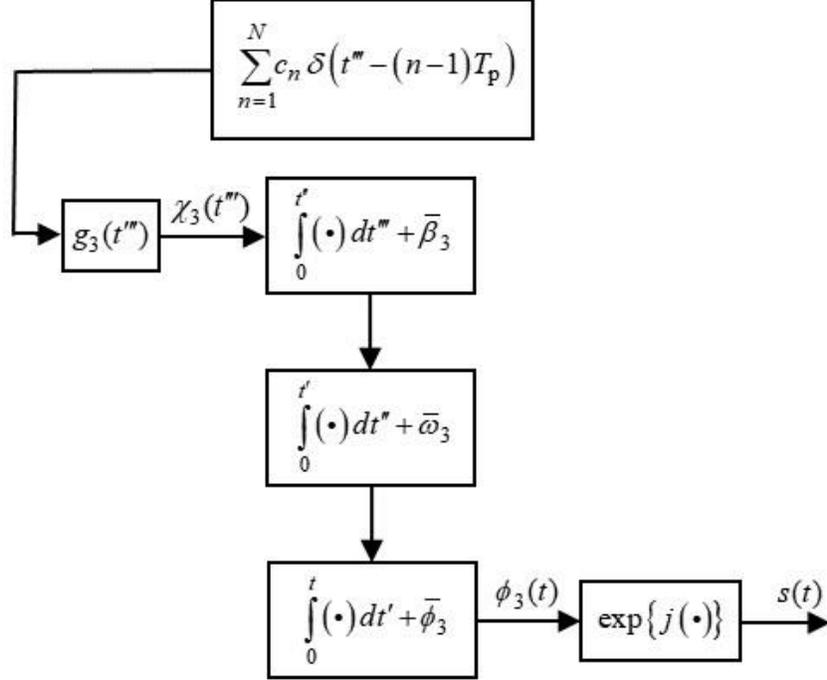


Fig. 3. Third-order PCFM waveform implementation
(see Appendix A for MATLAB™ script)

B. Relationships between different PCFM implementations

The relationships between different implementation orders offer some insight into the relative waveform design freedom of each, their permissible code values, and appropriate optimization approaches. Where (3), (5), and (6) provide the instantaneous phase function for each of these implementations, the instantaneous frequency function of each can be obtained by substituting in the respective coding structures of (2), (7), and (8) and then taking a derivative as

$$\dot{\phi}_1(t) = \frac{d\phi_1(t)}{dt} = \sum_{n=1}^N a_n g_1(t - (n-1)T_p) \quad (9)$$

$$\dot{\phi}_2(t) = \frac{d\phi_2(t)}{dt} = \int_0^t \left[\sum_{n=1}^N b_n g_2(t' - (n-1)T_p) \right] dt' + \bar{\omega}_2 \quad (10)$$

and

$$\begin{aligned} \dot{\phi}_3(t) &= \frac{d\phi_3(t)}{dt} \\ &= \int_0^t \int_0^{t'} \left[\sum_{n=1}^N c_n g_3(t'' - (n-1)T_p) \right] dt'' dt' + \int_0^t \bar{\beta}_3 dt' + \bar{\omega}_3 \end{aligned} \quad (11)$$

respectively. Likewise, the instantaneous chirp-rate of each implementation is obtained via an additional derivative as

$$\ddot{\phi}_1(t) = \frac{d^2\phi_1(t)}{dt^2} = \sum_{n=1}^N a_n \dot{g}_1(t - (n-1)T_p) \quad (12)$$

$$\ddot{\phi}_2(t) = \frac{d^2\phi_2(t)}{dt^2} = \sum_{n=1}^N b_n g_2(t - (n-1)T_p) \quad (13)$$

and

$$\ddot{\phi}_3(t) = \frac{d^2\phi_3(t)}{dt^2} = \int_0^t \left[\sum_{n=1}^N c_n g_3(t' - (n-1)T_p) \right] dt' + \bar{\beta}_3, \quad (14)$$

noting that the first-order representation in (12) necessitates the derivative of the shaping filter $g_1(t)$, where we assume this derivative exists for the purpose of the following analysis.

As an illustrative example, consider the linear FM (LFM) chirp, which possesses a rather simple structure and is an easy waveform to generate in hardware (e.g. via a swept local oscillator). For 3 dB bandwidth B and pulsewidth T , the LFM chirp-rate is

$$\beta_{\text{LFM}} = \frac{B}{T} \text{ Hz/s} = \frac{2\pi B}{T} \text{ rad/s}^2, \quad (15)$$

with associated time-bandwidth product $BT = \beta_{\text{LFM}} T^2$. Since the LFM phase is known to be quadratic in time, this waveform clearly represents a form of second-order implementation (a piece-wise linear approximation to LFM using a first-order implementation was presented in [25]). Recalling (4), for an up-chirp LFM at baseband the initial angular frequency is $-\pi/T_p$ and the final angular frequency at the end of the pulse is $+\pi/T_p$. The waveform therefore traverses a total angular frequency interval of $2\pi/T_p$ radians/s over the pulsewidth, for an associated bandwidth of $B = 1/T_p = N/T$ Hz. Substituting this result into (15) yields $\beta_{\text{LFM}} = N/T^2$ Hz/s which, when converted to angular frequency and again using $T = NT_p$, realizes $\beta_{\text{LFM}} = 2\pi/NT_p^2$ rad/s² that can be equated to the right side of (13) since LFM has a constant chirp-rate.

Given the time-bandwidth product $BT \cong N$, an exact LFM up-chirp could thus be realized with the second-order implementation of (5) and (7) by setting $\bar{\omega}_2 = -\pi/T_p$ and $b_n = 2\pi/N$ for $n = 1, 2, \dots, N$ and using a rectangular second-order shaping filter defined as $g_2(t) = (1/T_p^2) \text{rect}[0, T_p]$. Hence, the amount of angular frequency traversed

during an interval of T_p seconds is $2\pi/NT_p$ rad/s. The constant chirp-rate code b_n combined with the rectangular shaping filter realizes a constant instantaneous chirp-rate via (13) and thus a linear instantaneous frequency in (10), as expected for an LFM waveform. Note that the initial phase term $\bar{\phi}_2$ from (5) is arbitrary. While this coded implementation of an LFM is clearly more complicated than well-known analog waveform generation methods (e.g. the swept local oscillator mentioned above) it is useful as a means to establish a well understood basis with which to compare different implementation orders and particular waveform implementation structures.

Now consider generation of the same LFM waveform using a third-order implementation. With $\bar{\omega}_3 = -\pi/T_p$ and with $\bar{\phi}_3$ again being arbitrary, the obvious way would be to ignore the coding altogether (i.e. set all $c_n = 0$) by simply setting the initial chirp rate to $\bar{\beta}_3 = 2\pi/NT_p^2$. Alternatively, we could set $\bar{\beta}_3 = 0$ and determine the code values c_n and shaping filter $g_3(t)$ that, when combined and integrated as in (14), produce a constant LFM chirp rate. This result can be accomplished by taking the derivative of the second-order instantaneous chirp rate from (13), where the derivative of $g_2(t) = (1/T_p^2)\text{rect}[0, T_p]$ realizes a positive unit impulse function at $t = 0$, or $(1/T_p^2)\delta(t)$, and a negative unit impulse function at $t = T_p$, or $-(1/T_p^2)\delta(t - T_p)$. Consequently, an equivalent third-order shaping filter is

$$g_3(t) = (1/T_p^2)[\delta(t) - \delta(t - T_p)], \quad (16)$$

while the coding $c_n = b_n = 2\pi/N$ does not change. Because c_n is a constant, the bracketed term in (14) becomes

$$\sum_{n=1}^N c_n g_3(\tau - (n-1)T_p) = \frac{2\pi}{NT_p^2}[\delta(t) - \delta(t - T)], \quad (17)$$

where all but the first and last impulses over the pulsewidth cancel out. Figure 4 illustrates the normalized frequency content of these second-order and third-order implementations of an LFM waveform, which are identical as expected.

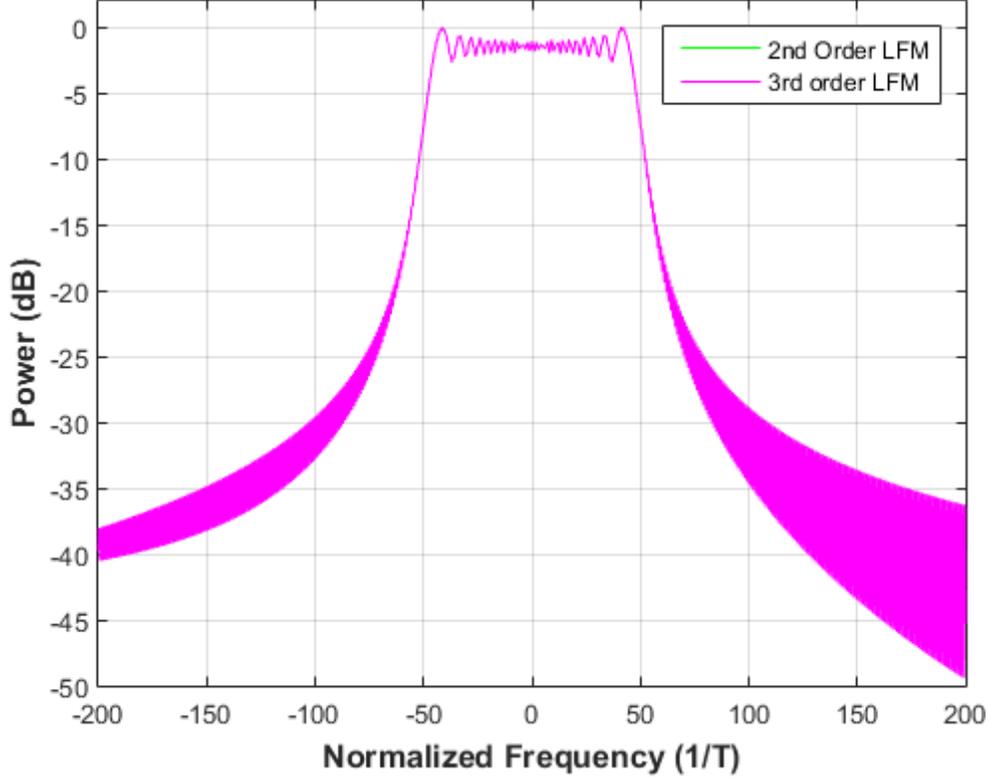


Fig. 4. Spectral content of second-order and third-order implementations of LFM with $BT = 100$

Now consider how the same could be achieved with first-order PCFM. The first-order implementation of (1) using a rectangular shaping filter can only realize a piece-wise linear (small stepped frequency) approximation to LFM, and thus the exact generation of LFM requires some modification to the first-order structure. A modified first-order implementation can be expressed as

$$\phi_{1,\text{mod}}(t) = \int_0^t \left[\sum_{n=1}^N a_n h_1(t' - (n-1)T_p) \right] dt' + \int_0^t \bar{\omega}_1 dt' + \bar{\phi}_1, \quad (18)$$

which includes an initial frequency offset $\bar{\omega}_1$ and the new shaping filter $h_1(t - (n-1)T_p)$. Comparing (9) with (10) and (12) with (13), we find that an exact LFM can be generated using this modified first-order implementation by setting $a_n = b_n = 2\pi/N$ and $\bar{\omega}_1 = -\pi/T_p$, with the new shaping filter defined as

$$\begin{aligned} h_1(t - (n-1)T_p) &= \int_0^t g_2(t' - (n-1)T_p) dt' \\ &= (1/T_p^2) \int_0^t \text{rect}[(n-1)T_p, nT_p] dt' \end{aligned}, \quad (19)$$

which simplifies to the ramp function

$$h_1(t - (n-1)T_p) = \begin{cases} 0, & 0 \leq t \leq (n-1)T_p \\ (t - (n-1)T_p)/T_p^2, & (n-1)T_p \leq t \leq nT_p \\ 1/T_p, & nT_p \leq t \leq NT_p \end{cases} . \quad (20)$$

Figure 5 illustrates the spectral content of this first-order ramp implementation that, once again, is found to be precisely that of an LFM waveform. In the next section, we use these LFM-based relationships to establish that the second-order shaping filter should integrate to $1/T_p$ over $[0, T_p]$.

More than just arcane new ways of generating an LFM, these different representations are special cases that establish how these implementation structures are related. However, the true utility in the different PCFM orders lies in the different continuous phase trajectories of physical waveforms that can be realized using finite first-, second-, and/or third-order codes and associated shaping filters. Consequently, the general structure of higher-order PCFM provides greater freedom to generate physical waveforms that possess desirable attributes.

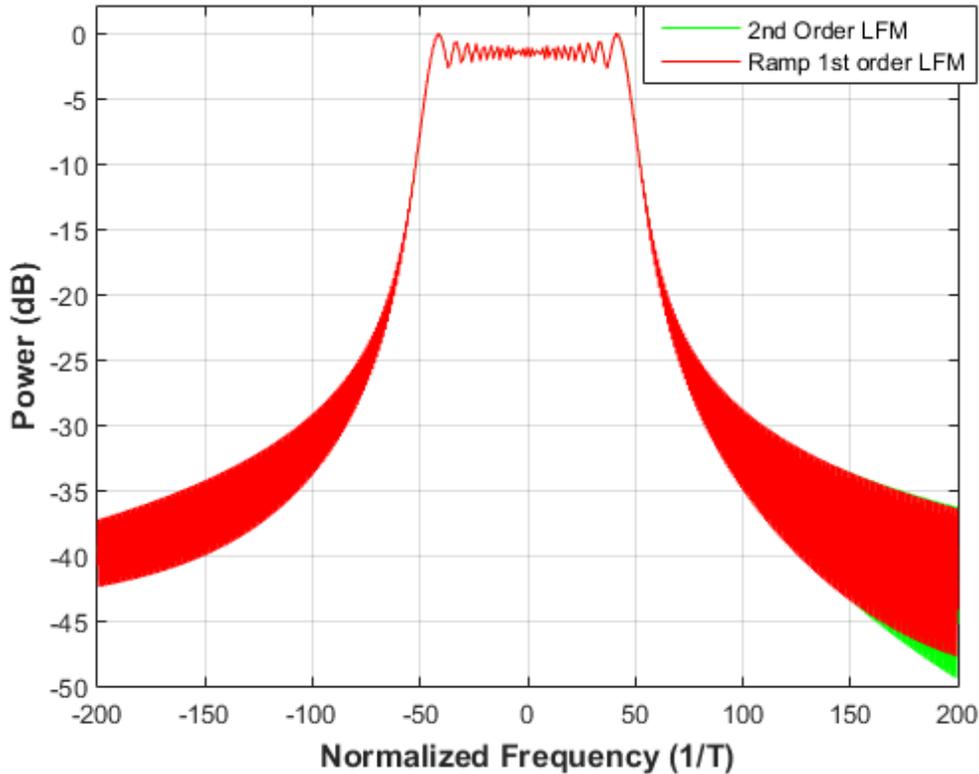


Fig. 5. Spectral content of second-order and first-order (RAMP) implementation of LFM with $BT = 100$

C. Permissible values for higher-order coding

As discussed in [27], permissible values of the first-order code α_n lie in $[-\pi, \pi]$ and these values correspond to normalized instantaneous frequencies, which translate into the (angular) edge frequencies of $\pm\pi/T_p$. As mentioned above, additional care must be taken with regard to selection of the code values for higher-order implementations.

To explore the permissible higher-order code values, let us begin by likewise constraining the derivatives in (9)-(11) as

$$-\pi/T_p \leq \sum_{n=1}^N a_n g_1(t - (n-1)T_p) \leq +\pi/T_p \quad (21)$$

$$-\pi/T_p \leq \int_0^t \left[\sum_{n=1}^N b_n g_2(t' - (n-1)T_p) \right] dt' + \bar{\omega}_2 \leq +\pi/T_p \quad (22)$$

and

$$-\pi/T_p \leq \int_0^t \int_0^{t'} \left[\sum_{n=1}^N c_n g_3(t'' - (n-1)T_p) \right] dt'' dt' + \int_0^t \bar{\beta}_3 dt' + \bar{\omega}_3 \leq +\pi/T_p. \quad (23)$$

Starting with the first-order case, due to the requirement that $\alpha_n \in [-\pi, \pi]$, the condition in (21) therefore necessitates that $\max\{g_1(t)\} \leq (1/T_p)$. Given the additional stipulation that $g_1(t)$ integrate to unity over the time support $[0, T_p]$, the first-order code constraint is met with equality when $g_1(t) = (1/T_p)\text{rect}[0, T_p]$. Any other shaping filter besides rectangular that also integrates to unity over $[0, T_p]$ would exceed this constraint. Thus a first-order code bound for an arbitrary shaping filter that also integrates to unity over $[0, T_p]$ is

$$\left[\frac{-\pi}{T_p \max\{g_1(t)\}} \right] \leq a_n \leq \left[\frac{+\pi}{T_p \max\{g_1(t)\}} \right]. \quad (24)$$

For the second-order case, consider the instantaneous angular frequency during the n th code interval by expanding the derivative in (10) as

$$\begin{aligned}
& \dot{\phi}_2((n-1)T_p \leq t \leq nT_p) \\
&= \int_0^t \left[\sum_{\ell=1}^n b_\ell g_2(t' - (\ell-1)T_p) \right] dt' + \bar{\omega}_2 \\
&= \int_{(n-1)T_p}^t b_n g_2(t' - (n-1)T_p) dt' + \int_0^{(n-1)T_p} \left[\sum_{\ell=1}^{n-1} b_\ell g_2(t' - (\ell-1)T_p) \right] dt' + \bar{\omega}_2, \\
&= \int_{(n-1)T_p}^t b_n g_2(t' - (n-1)T_p) dt' + \omega_{2,n-1}
\end{aligned} \tag{25}$$

where

$$\begin{aligned}
\omega_{2,n-1} &= \int_0^{(n-1)T_p} \left[\sum_{\ell=1}^{n-1} b_\ell g_2(t' - (\ell-1)T_p) \right] dt' + \bar{\omega}_2 \\
&= \left(\sum_{\ell=1}^{n-1} b_\ell \right) \left(\int_0^{T_p} g_2(t') dt' \right) + \bar{\omega}_2 \\
&= \left(\sum_{\ell=1}^{n-1} b_\ell \right) h_2(T_p) + \bar{\omega}_2 \\
&= \left(\sum_{\ell=1}^{n-1} b_\ell \right) \left(\frac{1}{T_p} \right) + \bar{\omega}_2
\end{aligned} \tag{26}$$

is the angular frequency at the beginning of the n th code interval, with $\omega_{2,0} = \bar{\omega}_2$ for $n = 1$ and

$$h_2(t) = \begin{cases} 0, & t \leq 0 \\ \int_0^t g_2(t') dt', & 0 \leq t \leq T_p \\ 1/T_p, & t \geq T_p \end{cases}. \tag{27}$$

Here we have made use of 1) the integration of the shaping filter is the same for each code interval, 2) the finite time support of $g_2(t)$ on $[0, T_p]$ realizes a constant for $t \geq T_p$ when integrated, and 3) this constant value is $h_2(T_p) = 1/T_p$ for consistency with (20). Likewise, evaluation of the integral in (25) over the n th code interval of $t \in [(n-1)T_p, nT_p]$ can be written as

$$\int_{(n-1)T_p}^t b_n g_2(t' - (n-1)T_p) dt' = b_n h_2(t - (n-1)T_p). \tag{28}$$

For instance, if $g_2(t) = (1/T_p^2) \text{rect}[0, T_p]$, then $h_2(t)$ would be a ramp function just like in (20). In general, if

$g_2(t)$ is non-negative for all time t , then $\max\{h_2(t)\} = h_2(T_p) = 1/T_p$ due to monotonicity. Therefore, the second-order code constraint in (22) can be simplified to

$$-\pi/T_p \leq (1/T_p)b_n + (1/T_p)\left(\sum_{\ell=1}^{n-1} b_\ell\right) + \bar{\omega}_2 \leq +\pi/T_p, \quad (29)$$

yielding the permissible values for the n th element of the second-order code as

$$\left[-\pi - \left(\sum_{\ell=1}^{n-1} b_\ell\right) - T_p \bar{\omega}_2\right] \leq b_n \leq \left[+\pi - \left(\sum_{\ell=1}^{n-1} b_\ell\right) - T_p \bar{\omega}_2\right]. \quad (30)$$

As an illustrative example, consider the second-order implementation of an LFM up-chirp waveform as discussed in the previous section where $\bar{\omega}_2 = -\pi/T_p$ and $b_n = 2\pi/N$ is a constant for $n = 1, 2, \dots, N$. Using these parameters as initialization, we can simplify (30) to determine permissible values for subsequent optimization, where the n th code element could take on values in the interval

$$\left[\frac{-2\pi(n-1)}{N}\right] \leq b_n \leq \left[\frac{+2\pi(N-n+1)}{N}\right]. \quad (31)$$

Specifically, for the first element ($n = 1$) the constraint is

$$0 \leq b_1 \leq +2\pi \quad (32)$$

and for the last element ($n = N$) the constraint is

$$-2\pi\left(1 - \frac{1}{N}\right) \leq b_N \leq +2\pi\left(\frac{1}{N}\right). \quad (33)$$

Likewise, in the center ($n = N/2 + 1$, assuming N is even) the constraint is

$$-\pi \leq b_{N/2+1} \leq +\pi. \quad (34)$$

Collectively, (32), (33), and (34) imply that, depending on the starting frequency and previous ($n - 1$) code values, the permissible values for b_n consist of an interval spanning 2π radians within the overall possible range of

$$-2\pi \leq b_n \leq +2\pi, \quad (35)$$

based on the hard constraint on frequency content defined by (22).

Finally, for the third-order case, expand the derivative in (11) for the n th code interval as

$$\begin{aligned}
& \dot{\phi}_3((n-1)T_p \leq t \leq nT_p) \\
&= \int_0^t \int_0^{t'} \left[\sum_{\ell=1}^n c_\ell g_3(t'' - (\ell-1)T_p) \right] dt'' dt' + \int_0^t \bar{\beta}_3 dt' + \bar{\omega}_3 \\
&= \int_0^t \int_0^{t'} c_n g_3(t'' - (n-1)T_p) dt'' dt' + \int_0^t \int_0^{t'} \left[\sum_{\ell=1}^{n-1} c_\ell g_3(t'' - (\ell-1)T_p) \right] dt'' dt' + \bar{\beta}_3 t + \bar{\omega}_3 \\
&= \int_0^t \int_0^{t'} c_n g_3(t'' - (n-1)T_p) dt'' dt' + \beta_{3,n-1} t + \omega_{3,n-1}
\end{aligned} \tag{36}$$

where $\beta_{3,n-1}$ and $\omega_{3,n-1}$ will be defined shortly. Note that the integral over the previous $(n-1)$ code intervals can be expressed as

$$\begin{aligned}
\int_0^t \int_0^{t'} \left[\sum_{\ell=1}^{n-1} c_\ell g_3(t'' - (\ell-1)T_p) \right] dt'' dt' &= \int_0^t \left[\sum_{\ell=1}^{n-1} c_\ell h_3(t' - (\ell-1)T_p) \right] dt' \\
&= \sum_{\ell=1}^{n-1} c_\ell f_3(t - (\ell-1)T_p)
\end{aligned} \tag{37}$$

in which

$$h_3(t) = \begin{cases} 0, & t \leq 0 \\ \int_0^t g_3(t') dt', & 0 \leq t \leq T_p, \\ 1/T_p^2, & t \geq T_p \end{cases} \tag{38}$$

noting the different constant for $h_3(t \geq T_p) = 1/T_p^2$ to remain consistent with the first-order and second-order formulations (and since there are still two additional integration stages). Subsequently, the final function in (37) has the general form

$$f_3(t) = \begin{cases} 0, & t \leq 0 \\ \int_0^t h_3(t') dt', & 0 \leq t \leq T_p, \\ t/T_p^2 + C, & t \geq T_p \end{cases} \tag{39}$$

with the constant $C = \int_0^{T_p} h_3(t') dt' - 1/T_p$ needed to maintain continuity of $f_3(t)$ at $t = T_p$. For example, if we set

$g_3(t) = (1/T_p^3) \text{rect}[0, T_p]$, then $h_3(t)$ would again be a ramp function like in (20), albeit scaled by an additional

factor of $1/T_p$. As a result, (39) takes on the particular form

$$f_3(t) = \begin{cases} 0, & t \leq 0 \\ t^2/(2T_p^3), & 0 \leq t \leq T_p \\ t/T_p^2 - 1/(2T_p), & t \geq T_p \end{cases} \quad (40)$$

in this case. In general, for $g_3(t)$ non-negative for all time t as in the first- and second-order instantiations, then the maximum value of $f_3(t)$ within $[0, T_p]$ is $\int_0^{T_p} h_3(t') dt' = \int_0^{T_p} \int_0^{t'} g_3(t'') dt'' dt'$. For the particular case of a rectangular shaping filter in (40), this maximum value is found to be $0.5/T_p$, which we shall use below.

For $t \geq T_p$, the function $f_3(t)$ increases linearly regardless of the particular shaping filter employed because it encompasses the double integral in (37). In other words, since (37) represents the contribution to instantaneous frequency, it is observed that during the n th code interval the previous third-order code values c_1 to c_{n-1} introduce piecewise linear chirping components in addition to constant frequency offsets, which can collectively be expressed as

$$\begin{aligned} \sum_{\ell=1}^{n-1} c_\ell f_3(t - (\ell-1)T_p) &= \sum_{\ell=1}^{n-1} c_\ell \left[(t - (\ell-1)T_p)/T_p^2 - 0.5/T_p \right] \\ &= \sum_{\ell=1}^{n-1} c_\ell \left[(t - (\ell-1)T_p)/T_p^2 \right] - \sum_{\ell=1}^{n-1} c_\ell \left[0.5/T_p \right]. \\ &= \left(\sum_{\ell=1}^{n-1} c_\ell \right) (t/T_p^2) - \sum_{\ell=1}^{n-1} c_\ell \left[(\ell-0.5)/T_p \right] \end{aligned} \quad (41)$$

Inserting (41) into (36) and associating the time-varying and constant frequency terms prior to the n th code interval therefore yields the preceding chirp-rate

$$\beta_{3,n-1} = \left(\sum_{\ell=1}^{n-1} c_\ell \right) (1/T_p^2) + \bar{\beta}_3 \quad (42)$$

and angular frequency

$$\omega_{3,n-1} = - \sum_{\ell=1}^{n-1} c_\ell \left[(\ell-0.5)/T_p \right] + \bar{\omega}_3 \quad (43)$$

from the last line of (36), where $\beta_{3,0} = \bar{\beta}_3$ and $\omega_{3,0} = \bar{\omega}_3$ for $n = 1$. The remaining portion of (36) can likewise be expressed as

$$\begin{aligned}
\int_0^t \int_0^{t'} c_n g_3(t'' - (n-1)T_p) dt'' dt' &= \int_0^t c_n h_3(t' - (n-1)T_p) dt' \\
&= c_n f_3(t - (n-1)T_p)
\end{aligned} \tag{44}$$

for the n th code interval. Using this result along with (42) and (43), the instantaneous frequency from (36) becomes

$$\begin{aligned}
&\dot{\phi}_3((n-1)T_p \leq t \leq nT_p) \\
&= \int_0^t \int_0^{t'} c_n g_3(t'' - (n-1)T_p) dt'' dt' + \beta_{3,n-1} t + \omega_{3,n-1} \\
&= c_n f_3(t - (n-1)T_p) + \left[\left(\sum_{\ell=1}^{n-1} c_\ell \right) \left(1/T_p^2 \right) + \bar{\beta}_3 \right] t - \sum_{\ell=1}^{n-1} c_\ell \left[(\ell-0.5)/T_p \right] + \bar{\omega}_3
\end{aligned} \tag{45}$$

Because the center (chirp) component in (45) changes linearly with time while the first (coded) component changes nonlinearly, and assuming $g_3(t)$ is non-negative for all time t , the point of maximum possible frequency deviation during the n th code interval occurs at $t = nT_p$. Also using the result $f_3(T_p) = 0.5/T_p$ for the rectangular shaping filter, the hard frequency constraint from (23) becomes

$$-\pi/T_p \leq \left\{ 0.5c_n/T_p + \left[\left(\sum_{\ell=1}^{n-1} c_\ell \right) \left(1/T_p^2 \right) + \bar{\beta}_3 \right] nT_p - \sum_{\ell=1}^{n-1} c_\ell \left[(\ell-0.5)/T_p \right] + \bar{\omega}_3 \right\} \leq +\pi/T_p \tag{46}$$

which can be rearranged to establish the permissible code values as

$$2 \left[-\pi - \sum_{\ell=1}^{n-1} [n - \ell + 0.5] c_\ell - \bar{\beta}_3 n T_p^2 - \bar{\omega}_3 T_p \right] \leq c_n \leq 2 \left[+\pi - \sum_{\ell=1}^{n-1} [n - \ell + 0.5] c_\ell - \bar{\beta}_3 n T_p^2 - \bar{\omega}_3 T_p \right]. \tag{47}$$

As an illustrative example similar to that for the second-order case, set $\bar{\beta}_3 = 0$ and $\bar{\omega}_3 = -\pi/T_p$. Applying (47), the permissible values for the first term ($n = 1$) in the third-order code should therefore reside within

$$0 \leq c_1 \leq +4\pi. \tag{48}$$

If the maximum value of $+4\pi$ is selected for c_1 , the bound on the next term ($n = 2$) is, via (47),

$$-12\pi \leq c_2 \leq -8\pi. \tag{49}$$

If the minimum value of -12π is then selected for c_2 , the subsequent bound according to (47) for the $n = 3$ term is

$$+16\pi \leq c_3 \leq +20\pi. \tag{50}$$

The point of this exercise is to show that the possible values for a given third-order code element span 4π radians within an interval that can shift significantly from one element to the next. This possible shifting arises because the

accrued memory from previous chirp acceleration terms as illustrated in (45) may necessitate successively larger code values as compensation to ensure the instantaneous frequency stays within the specified bounds of $[-\pi, \pi]$. Of course, small third-order code values require far less compensation by later code terms, which is actually the type of result that has (thus far) been obtained in optimization as shown in the next section. It remains to be seen how optimization of the third-order parameterization can be exploited to the full extent possible.

In short, the second-order and third-order implementations become increasingly more complex, particularly with regard to determination of their feasible code values. For the second-order case this complexity involves a “frequency memory” term $\omega_{2,n-1}$ from (26) that includes the effect of previous code values. The third-order case likewise involves $\omega_{3,n-1}$ from (43), as well as a “chirp memory” term $\beta_{3,n-1}$ from (42). Unlike the first-order implementation where every code element is established independently, these memory terms for the second and third-order implementations can have a significant impact on later code values.

It is also important to note that the frequency constraints for the second-order and third-order implementations, originally stated in (22) and (23), are extensions of the first-order constraint and thus rely on an implicit assumption of constant frequency during a code element time interval of T_p . However, these higher-order implementations clearly permit frequency to change during this interval. Thus for nonlinear chirp-like waveforms that tend to exhibit rapid frequency changes near the pulse edges (which typify what are arguably the “best” waveforms in terms of low autocorrelation sidelobes based on a *conservation of ambiguity* notion [14]) these higher-order “constraints” should instead be viewed as useful guidelines for parameter selection. Like in [39], aggregate spectral content can be used in lieu of hard frequency constraints to provide greater freedom for higher-order code design while maintaining necessary spectral containment.

D. Aggregate spectral containment

While the higher-order constraints derived above provide general guidelines for feasible code values, a more practical way to ensure spectral containment is to constrain the aggregate spectral content of the entire waveform. For example, in [27] the frequency template error (FTE) metric was defined which takes the form

$$\Phi_{\text{FTE}}[S(f)] = \left(\frac{1}{f_{\text{H}} - f_{\text{L}}} \right) \int_{f_{\text{L}}}^{f_{\text{H}}} \left| |S(f)|^p - |W(f)|^p \right|^q df, \quad (51)$$

where $S(f)$ is the Fourier transform of waveform $s(t)$, the integration limits f_L and f_H demarcate the frequency interval of interest (including sufficient spectral roll-off beyond the 3-dB bandwidth), and $W(f)$ is some desired frequency template such as a Gaussian window. The values p and q control the emphasis placed on in-band and out-of-band frequencies, with $p=1$ and $q=2$ defining a frequency-domain mean-square error (MSE) metric. A logarithmic version of (51) was similarly employed in [37] and other such frequency domain metrics are likely to be useful as well. Paraphrasing [8], a waveform having an aggregate spectral shape that decreases towards the band edges is known to also possess low autocorrelation sidelobes, and thus this manner of metric address both spectral containment and sidelobe response.

As discussed in [39] where the notion of PCFM “over-coding” was introduced, it is possible to exceed (even first-order) frequency constraints as long as the aggregate spectral content still adheres to the desired spectral template. This idea is really just an extension of the well-known *principle of stationary phase* conceived by the earliest developers of nonlinear FM (NLFM) waveforms [5] that states that the energy spectral density at a particular instantaneous frequency is inversely proportional to the chirp rate at that instant in time. More generally, one can say that the amount of time (relative to pulsewidth T) that a waveform resides in a given frequency directly relates to the relative amount of energy placed at that frequency. Consequently, where traditional NLFM waveform design has focused on the determination of nonlinear time-frequency functions with which to implement the stationary phase principle [13], these parameterized PCFM structures permit use of various optimization methods to search for waveforms that achieve the desired aggregate spectral content. Of course, the time-frequency functions previously developed also provide very good starting points from which to initialize for further optimization.

It is worth noting that this spectral density design perspective is quite powerful as a means to obtain individual waveforms that provide very low sidelobes. For instance, a Gaussian shaped spectrum was used in [49] to jointly optimize an FM waveform having $BT = 128$ and low-loss amplitude taper that was experimentally demonstrated to have a peak sidelobe 83 dB below the mainlobe match point while incurring only 0.26 dB of loss relative to a constant amplitude pulse (and in simulation the peak sidelobe was 108 dB below the mainlobe).

However, an even more significant benefit of spectral density based design, emphasizing the necessity to account for sufficient over-sampling relative to 3-dB bandwidth so as to preserve spectral roll-off characteristics, is that such an approach naturally supports the optimization of numerous waveform-diverse emission schemes that are physically

realizable. Examples include a variety of FM noise waveform arrangements [37, 50] including the first experimental demonstration of practical complementary structure [38] and simultaneous dual-polarized emissions that are separable on receive [51], practical spectral notching on transmit for cognitive interference avoidance and associated forms of radar-embedded communications [52, 53], a practical form of intermodulation-based nonlinear harmonic radar [54], and wideband/widebeam MIMO emissions that avoid “transmission” into the array invisible space that could otherwise potentially damage the radar [55]. The obvious practical utility of this spectrum-based design perspective to facilitate these and more new sensing capabilities likewise make it rather attractive in the context of higher-order PCFM implementations.

E. Multi-order PCFM implementations

Given the ability to generate higher-order PCFM waveforms as described in the previous sections, it is then also possible to combine them in a multi-order formulation to take advantage of the additional design freedom while maintaining the same aggregate spectral content. For instance, the first-order and second-order schemes from (3) and (5), respectively, can be combined as

$$\phi_{21}(t) = \int_0^t \chi_1(t') dt' + \int_0^t \int_0^{t'} \chi_2(t'') dt'' dt' + \int_0^t \bar{\omega}_{21} dt' + \bar{\phi}_{21}, \quad (52)$$

for $\chi_1(t)$ and $\chi_2(t)$ the first-order and second-order coded signals defined in (2) and (7), respectively, and $\bar{\omega}_{21}$ and $\bar{\phi}_{21}$ the initial frequency and phase. Likewise, all three orders can be employed by combining (6) with (52) as

$$\begin{aligned} \phi_{321}(t) = & \int_0^t \chi_1(t') dt' + \int_0^t \int_0^{t'} \chi_2(t'') dt'' dt' + \int_0^t \int_0^{t'} \int_0^{t''} \chi_3(t''') dt''' dt'' dt' \\ & + \int_0^t \int_0^{t'} \bar{\beta}_{321} dt'' dt' + \int_0^t \bar{\omega}_{321} dt' + \bar{\phi}_{321} \end{aligned}, \quad (53)$$

for $\chi_3(t)$ the third-order coded signals defined in (8) and with $\bar{\beta}_{321}$, $\bar{\omega}_{321}$, and $\bar{\phi}_{321}$ now the initial chirp-rate, frequency, and phase. In light of the derivation of the code constraints and subsequent discussion regarding their complexity and relaxation in Section II.C, it is clear that metrics involving aggregate spectral content as described in Section II.D should be employed when designing multi-order waveforms.

The general form for the multi-order PCFM implementation is depicted in Fig. 6, with the Matlab™

implementation provided in Appendix A. Any combination of first, second, and third orders, including single order implementations, can be obtained by simply setting the unused code(s) and/or parameters to zero.

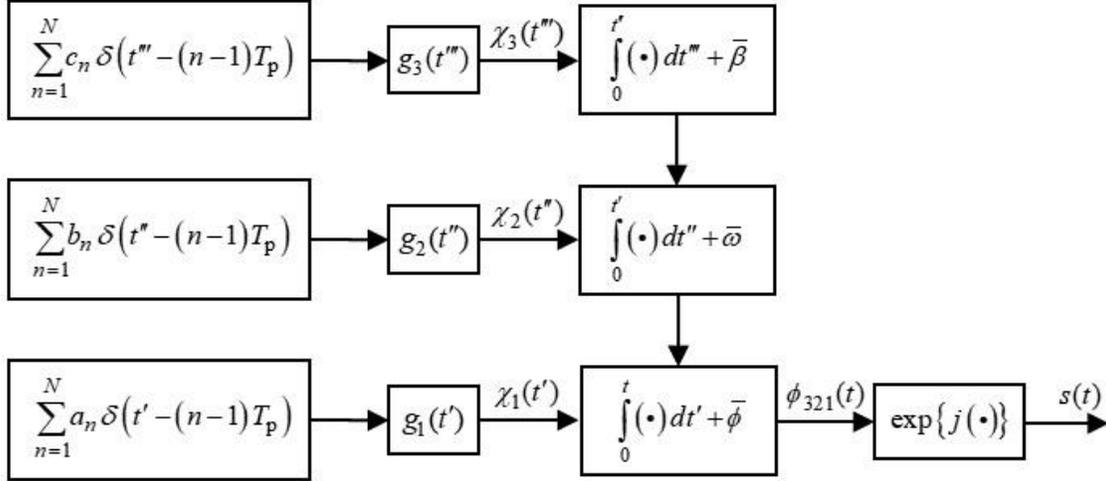


Fig. 6. Multi-order PCFM implementation for first-order at the bottom up to third-order at the top (see Appendix A for MATLAB™ script)

III. HIGHER-ORDER PCFM WAVEFORM DESIGN

The parameterized structure of the first-order PCFM implementation [25, 27] necessitates determination of the underlying code to realize the resulting physical waveform. The same is true for these higher-order and multi-order PCFM implementations as well. Determination of the corresponding code(s) and the initial frequency and chirp-rate as appropriate requires a search of the high-dimensional space these codes parameterize according to some prescribed cost function, which generally involves some measure of the waveform ambiguity function (the zero-Doppler cut is typical). There are myriad ways in which this optimization could be performed and it remains to be seen what types of waveforms can be obtained. For instance, it has been observed [4-14] that the design of an individual FM waveform with low range sidelobes tends to realize one that is relatively chirp-like and has a roughly “sideways S” time-frequency profile. In contrast, more recent developments involving FM noise [32, 37, 38, 50-54] produce waveforms with more random structure that benefit from the incoherent combining of sidelobes in the slow-time Doppler domain after pulse compression. With the prospect of many different design metrics, different approaches to optimize those metrics, and the different applications (and thus constraints and requirements) to which a waveform or set of waveforms may be applied, we shall limit the discussion here to the delineation of general attributes of these new physically realizable waveform implementations as opposed to considering how best to use them.

As an illustrative example to highlight these different structures, the “performance diversity” paradigm introduced in [27] was employed to optimize prototype waveforms for each implementation. This approach exploits the fact that different ambiguity function metrics that specifically evaluate the waveform autocorrelation tend to complement one another. Such metrics include the peak sidelobe level (PSL), the integrated sidelobe level (ISL), and the FTE summarized in Section II.D. Simply put, this optimization approach exploits the fact that these metrics respond relatively similarly in terms of waveform goodness, yet can be expected to possess different local minima. Consequently, a greedy search strategy involving the selection and update of the single code element whose change would provide the most improvement can alternate between the metrics to help avoid some of the local minima. Details of this particular approach can be found in [27]. No claim is made as to whether this search strategy is better or worse than any others. It is used here for the simple reason that it can be performed in essentially the same manner for each of the implementations so as to provide a means of comparison between them.

For waveform $s(t) = \exp\{j\phi(t)\}$ based on one of the PCFM implementations from (1), (5), (6), (52), or (53), or any continuous waveform for that matter, the autocorrelation (matched filter response) is

$$r(\tau) = \int_{t=0}^T s(t) s^*(t + \tau) dt \quad (54)$$

as a function of delay $-T \leq \tau \leq T$, with the interval $-\tau_m \leq \tau \leq \tau_m$ delimiting the mainlobe. For ease of reference, the well-known PSL and ISL metrics are

$$\text{PSL} = \max_{\tau} \left| \frac{r(\tau)}{r(0)} \right| \quad (55)$$

for $\tau \in [\tau_m, T]$ and

$$\text{ISL} = \frac{\int_{\tau_m}^T |r(\tau)|^2 d\tau}{\int_0^{\tau_m} |r(\tau)|^2 d\tau}. \quad (56)$$

For zeroth-order codes the denominator of (56) is often omitted because it is a constant, but this term is required for evaluation of continuous waveforms.

For all cases considered here a time-bandwidth product of $BT \cong N = 100$ is used and the FTE frequency weighting template from (51) is set to have a Gaussian shape. The performance diversity optimization process [27]

alternates between the metrics of PSL, ISL, and FTE, with the particular metric changed after a predetermined number of iterations or once no further improvement is obtained with the current metric. The higher-order implementations likewise rely on the frequency template to ensure containment of the aggregate spectral content. In addition, the value of the initial frequency for the higher-order implementations is rescaled after each iteration so that the waveform is centered on a baseband frequency of 0 prior to comparing the waveform power spectrum within the FTE metric.

A. Optimization of example waveforms for second/third-order PCFM implementations

For the following optimization results, the first-order case was initialized with a piece-wise LFM waveform, which has been found [27] to yield good final results by starting with an established delay-Doppler ridge (based on the “conservation of ambiguity” [14]). Leveraging the principle of stationary phase, the second-order waveform optimization is initialized with the scaled inverse of Taylor window coefficients possessing -40 dB range sidelobes. Likewise, the piece-wise difference of these initial second-order coefficients (thus approximating a derivative) was used for the third-order initialization.

Figures 7-9 illustrate the autocorrelation, aggregate spectral content, and instantaneous frequency (or time-frequency function) for optimized first-order (red) and second-order (green) PCFM implemented waveforms based on (1) and (5), respectively. In Fig. 7 it is observed that, for this example, the second-order waveform exhibits a PSL that is ~ 2.5 dB lower than that of the first-order waveform. While we cannot (yet) say that optimized waveforms obtained using the second-order implementation always outperform those using the first-order implementation for the same BT , strong anecdotal evidence has been observed to this effect over many different initializations and optimization processes (i.e. performance diversity ordering of metrics).

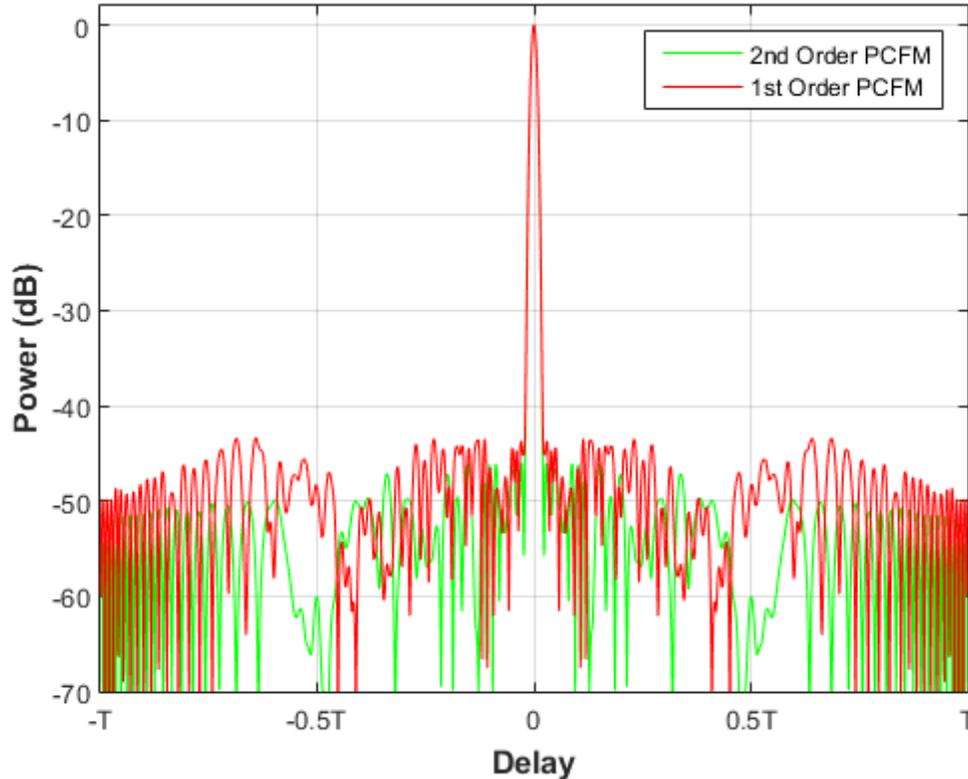


Fig. 7. Autocorrelations of optimized first-order and second-order PCFM implemented waveforms for $BT = 100$

The spectral content (Fig. 8) for the optimized second-order waveform is also found to exhibit a slightly broader roll-off from -15 dB down to about -40 dB. This effect can be explained by examining the instantaneous frequency (Fig. 9), where the second-order waveform is shown to possess the rapid chirping behavior typified by NLFM waveforms that are designed according to the principle of stationary phase [4, 13]. It is interesting to note, however, that unlike traditional smooth time-frequency functions (e.g. [4-12]), the optimized second-order PCFM instantaneous frequency does exhibit small perturbations that serve to further break up sidelobe coherence. That said, the second-order perturbations are less pronounced than for the first-order case due to the presence of the additional integration stage (per Fig. 2) that smooths out such effects to some degree.

A similar broader roll-off of the spectrum was previously observed when using an “over-coded” version of the first-order implementation [39], with commensurate sidelobe reduction. In fact, optimization of the over-coded first-order implementation was found to achieve even greater improvement in terms of sidelobe reduction relative to the second-order scheme. However, while both do require the use of a template to ensure good spectral containment, the over-coded arrangement also involves a multiplicative increase in the number of parameters to optimize. The benefits of the expanded parameterization (while preserving the same BT), and the associated relationships between spectral

roll-off and range sidelobes, is part of the justification for exploring these alternative/combined implementations.

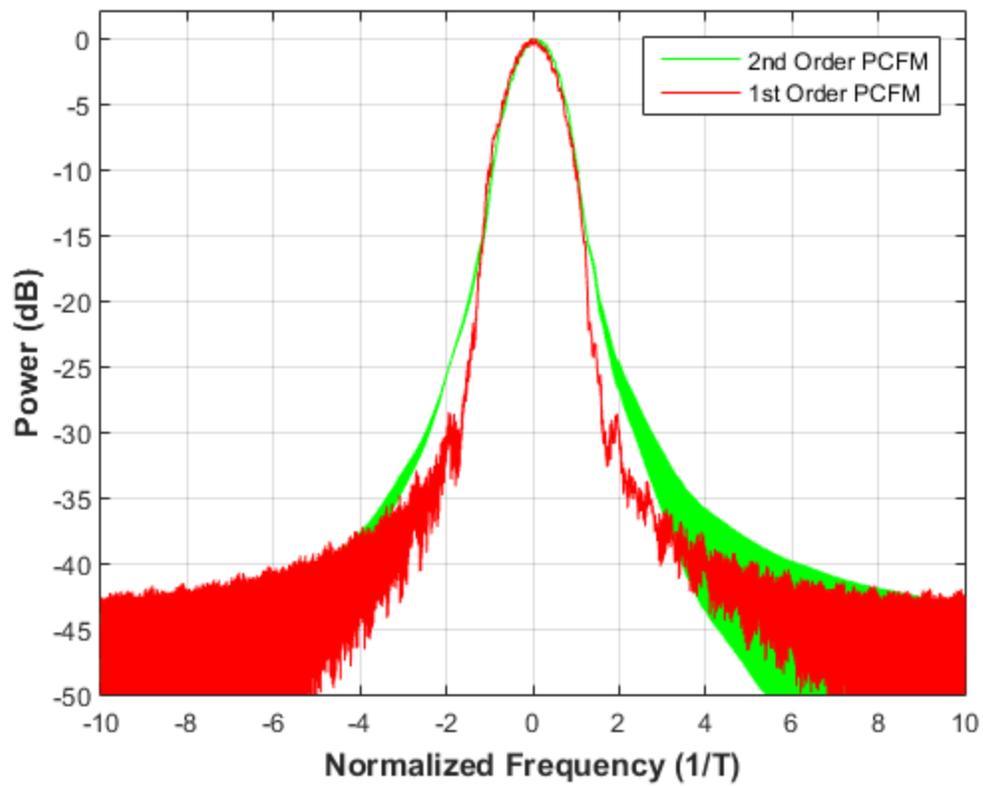


Fig. 8. Spectral content of optimized first-order and second-order PCFM implemented waveforms for $BT = 100$

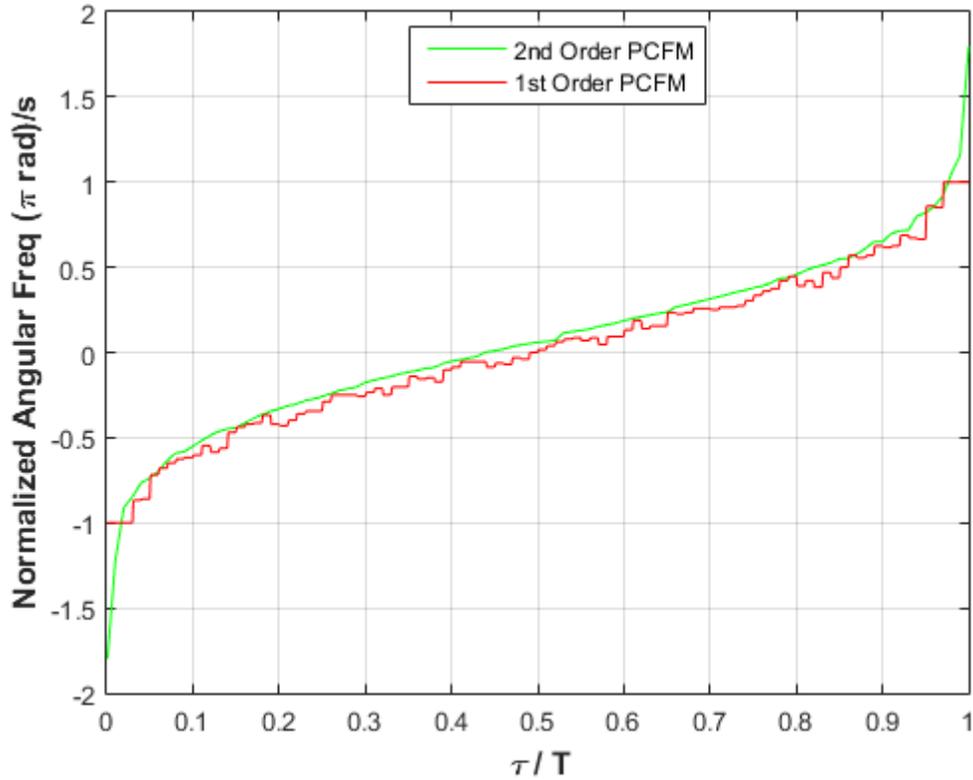


Fig. 9. Instantaneous frequency of optimized first-order and second-order PCFM implemented waveforms for $BT = 100$

Figures 10-12 subsequently illustrate the autocorrelation, aggregate spectral content, and instantaneous frequency for an optimized third-order (blue) PCFM implemented waveform based on (6), with the first-order implemented waveform (red) included again for comparison. Where the second-order waveform demonstrated a sidelobe reduction relative to first-order in Fig. 7, it is conversely observed in Fig. 10 that the optimized third-order waveform realizes poorer performance, with the sidelobes ~ 5 dB greater than for the first-order waveform. As with the second-order implementation, we cannot make the claim that this performance relationship always occurs for the same BT , though it has anecdotally been observed in all the cases thus far examined. As discussed with the example via (48)-(50), expanding the code search space according to the derived limits could possibly be a way to achieve better performance for the third-order implementation.

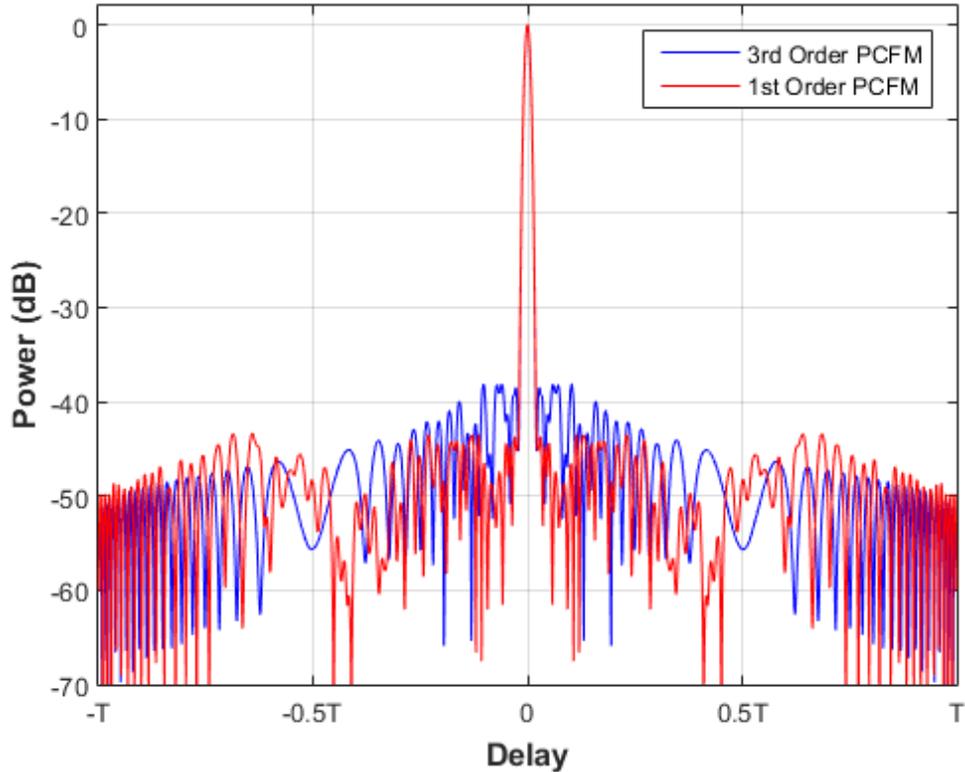


Fig. 10. Autocorrelations of optimized first-order and third-order PCFM implemented waveforms for $BT = 100$

The spectral roll-off (Fig. 11) for third-order does more closely match that of the first-order, albeit with a frequency-offset that arises due to the difficulty to optimize the more complex waveform implementation of (6). Recalling the implications of the principle of stationary phase, the reason for this frequency offset is observed in the instantaneous frequency plot (Fig. 12), where the beginning of the waveform is found to exhibit a slower nonlinear chirping behavior than at the end of the waveform. Further, the inclusion of another integration stage (now three) also produces an even smoother time-frequency function than either second-order or first-order, which tends to restrict the presence of small perturbations that could otherwise break up sidelobe coherence. It stands to reason that other methods, such as the gradient descent approach in [35] or other possible search methods, may be better suited for the optimization of third-order parameterized FM waveforms due to the increased complexity.

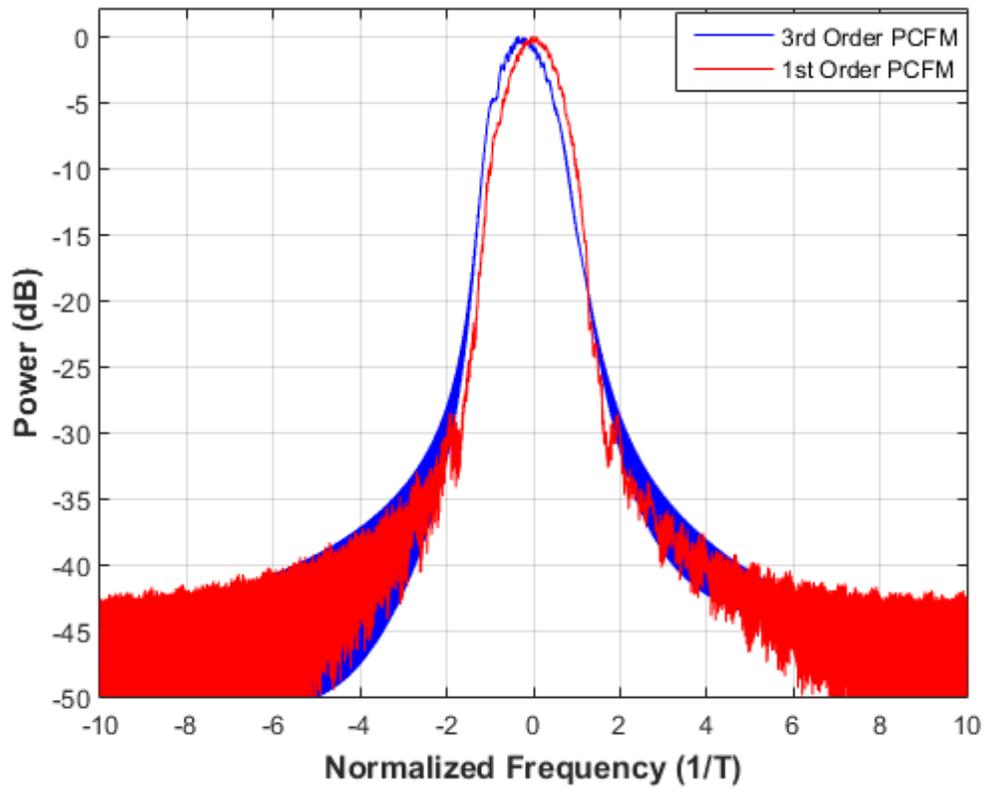


Fig. 11. Spectral content of optimized first-order and third-order PCFM waveforms for $BT = 100$

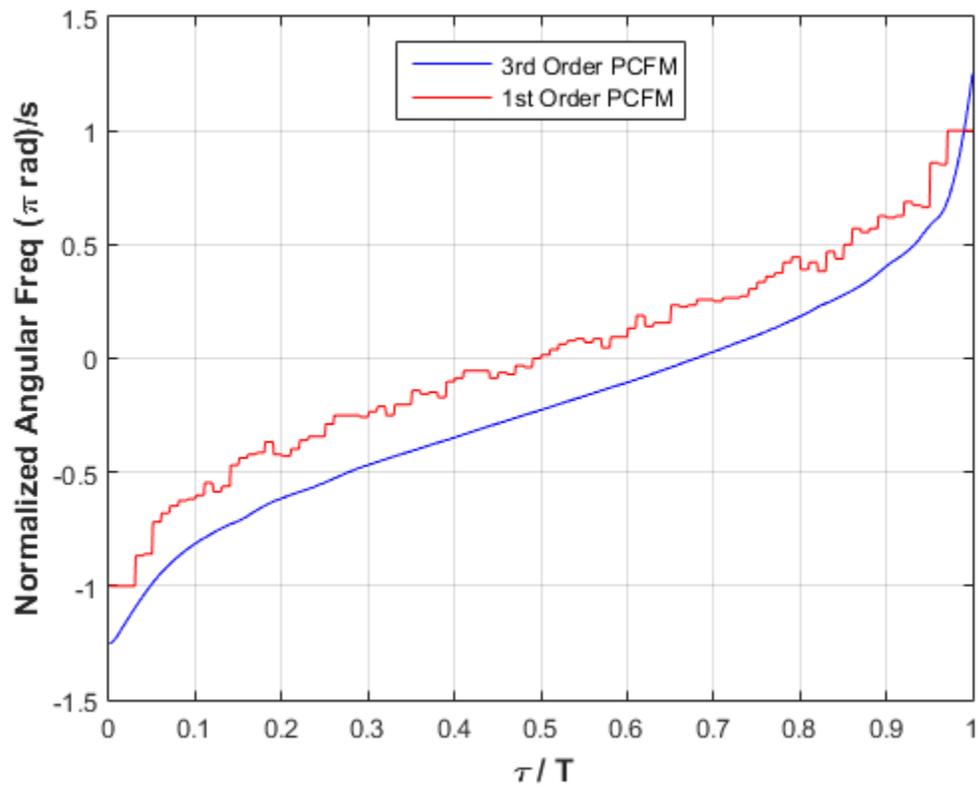


Fig. 12. Instantaneous frequency of optimized first-order and third-order PCFM waveforms for $BT = 100$

For the particular instantiation using the performance diversity optimization scheme and the initializations discussed above, Table II quantifies the PSL and ISL values for the example waveforms resulting from first-order, second-order, and third-order implementations for $BT = 100$, with B the 3-dB bandwidth. As a useful benchmark, the PSL bound for hyperbolic FM (HFM) waveforms, which is $-20\log_{10}(BT) - 3$ dB [8, 13, 14], is also included for the same BT . As observed previously in [27], the first-order implementation is able to exceed the HFM bound by a small margin. The second-order implementation, however, surpasses the bound by 3 dB. Of course, while the 3-dB bandwidth remains constant across these implementations, Fig. 9 reveals that the second-order case achieves this improvement in part due to a greater *swept bandwidth* arising from the sharp nonlinear chirping at the beginning and end of the pulse (about $1.8\times$ greater), which translates into the modest broadening observed in the aggregate spectrum of Fig. 8. The associated HFM PSL bound corresponding to that increased bandwidth is -48.1 dB.

Table II. PSL and ISL for 1st, 2nd and 3rd order optimized waveforms for $BT=100$

	1 st order	2 nd order	3 rd order	HFM bound
PSL (dB)	-43.4	-46.0	-38.1	-43.0
ISL (dB)	-59.5	-63.5	-57.4	N/A

It is also instructive to examine the structures of the different optimized codes underlying these implementations. The first-order code is essentially shown in Figs. 9 and 12 because the values of the code scale the rectangular shaping filters, thus yielding a stair-step signal that realizes a continuous, piece-wise linear phase trajectory after the single integration stage in (1). While recent work [28-38] has demonstrated just how capable this first-order implementation can be for the optimization and generation of physical waveform-diverse emissions, it stands to reason that expansion beyond this piece-wise linear phase structure may hold even greater potential due to the increased design freedom, smoother phase trajectories, and the prospect of producing more rapid chirping behaviors in a parameterized manner.

Figure 13 shows the optimized second-order code from the above example along with the associated upper/lower derived limits from (30) based on these code values, with the limits spanning an interval of 2π radians. Note that at the beginning and end of the pulse, the code values do exceed these limits by a modest amount, which occurs because the rapid chirping in these regions (see Fig. 9) provides a sidelobe lowering benefit [8] by facilitating a good match to the Gaussian spectral template used for optimization via (51). It is likewise observed that these limits, which as discussed at the end of Sect. II.C should instead be viewed as general guidelines, have a shape that would appear to

typify a good NLFM down-chirp (the optimized second-order NLFM is an up-chirp), if not for the fact that these code values represent instantaneous chirp rate instead of instantaneous frequency.

Further, while the general instantaneous frequency structure (Fig. 9) for this example optimized waveform from a second-order implementation is rather typical of NLFM waveforms obtained in the past [4-14], it is observed that most of the second-order code values in Fig. 13 are nearly constant, which makes sense given that the central portion of the waveform has a near-linear time-frequency relationship like LFM. However, the largely unused freedom of movement implied by Fig. 13 also speaks to untapped potential for this implementation as a means to realize new agile FM waveform constructs like in [28-39, 44-54].

Similar behavior is found in Fig. 14 for the third-order optimized code and limits from (47), where the former again modestly exceeds the latter at the ends of the pulse. Here the limits span 4π radians but otherwise still have a shape that looks like an NLFM down-chirp, but is in fact related to chirp acceleration. These shapes arise for both of these implementations because the actual waveforms associated with the codes in Fig. 13 and 14 are up-chirp NLFM and the corresponding code limits subsume the frequency and chirp memory terms that produce those waveforms.

Note that, as discussed in Sect. II.C, the actual third-order code values obtained by optimization are rather small compared to the bounds, yet this code produces a waveform with the instantaneous frequency function depicted in Fig. 12, which again is clearly recognizable as the “sideways S” shape one generally associates with a good NLFM waveform. Due to the accumulation of memory terms, exploitation of the greater freedom available for the third-order implementation, particularly as a means to achieve new agile waveform arrangements such as mentioned above, remains a topic of ongoing investigation.

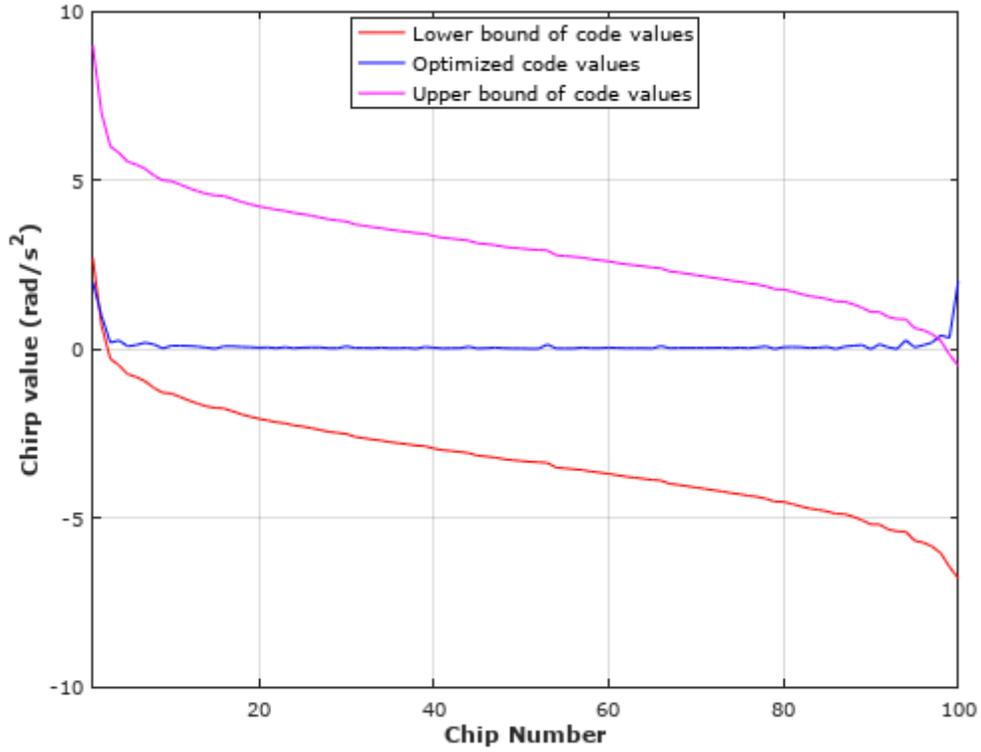


Fig. 13. Optimized second-order code (instantaneous chirp rate) and derived upper/lower limits

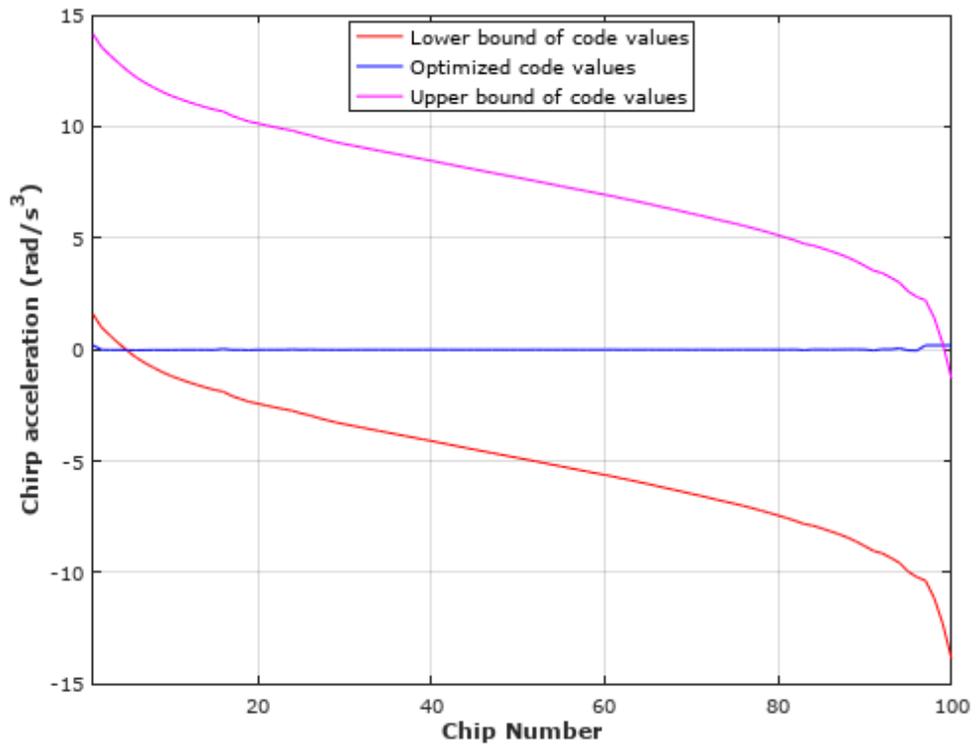


Fig. 14. Optimized third-order code (instantaneous chirp acceleration) and derived upper/lower limits

B. Optimization of example waveforms for multi-order PCFM implementations

Using (52) and (53), now consider optimization of waveforms based on the multi-order PCFM implementations of the same $BT = 100$ dimensionality. For joint optimization, either both (for first/second-order combined) or all three of the codes are designed according to the same greedy search used above (and in [27]) in which the single code element whose change would provide the greatest improvement is updated at each stage. We also examine sequential optimization whereby the different order codes are each optimized until no further improvement is possible (with the other codes set to 0 initially) and then that code is fixed while the optimization of a different code commences. Perhaps not surprising given the results above, it has been found that starting with the second-order code, followed by either the first-order or third-order codes, generally realizes waveforms with the best performance in terms of sidelobe reduction for the sequential approach. Like before, these resulting waveforms are examples of what could be achieved, with the expectation that there are better ways to optimize these structures as well as new agile waveform arrangements that could be realized.

Figures 15-17 depict the autocorrelation, spectral content, and instantaneous frequency, respectively, for joint greedy optimization of (52) and (53). Both cases realize PSL values (Fig. 15) that are nearly 8 dB better than the previous first-order result and 5 dB better than the second-order result alone. The shape of the spectral content (Fig. 16) is basically the same as the previous cases, with a small asymmetry caused by inclusion of the third-order coding that is present. Finally, the usual “sideways S” shape is again observed for the instantaneous frequency, with extensions of the *swept bandwidth* by factors of $2\times$ and $2.25\times$ for the first/second combined orders and first/second/third combined orders, respectively. These bandwidths correspond to HFM PSL bounds of -49.0 dB and -50.0 dB, respectively. Per Table III, it is interesting to note that the PSL values achieved by these multi-order waveforms do in fact surpass these bounds, an effect that can likely be largely attributed to the additional presence of the first-order perturbations observed in Fig. 17 that serve to break up sidelobe coherency and arise naturally from the optimization process. It remains to be seen whether the second or third-order implementations alone can realize these beneficial perturbations through greater exploitation of their individual code design spaces (per Figs. 13 and 14).

Plots for sequential optimization are not included because they are negligibly different from joint optimization, as evidenced by the resulting PSL and ISL values in Table III. However, the sequential optimization is more computationally efficient to implement since it does not require a greedy search over all the codes simultaneously.

The joint first/second/third-order instantiation clearly provides virtually no PSL benefit over the joint first/second-order case and only a modest ISL improvement. This result is included to demonstrate that little additional improvement has been observed by adding the third-order code, which does incur a higher cost to optimize due to greater dimensionality. A fourth-order implementation (not shown) has also been examined within this multi-order context and it likewise provided no discernible additional benefit. With all that said, it remains to be seen whether gradient-based optimization [35] or other possible code search procedures may alter this arrangement.

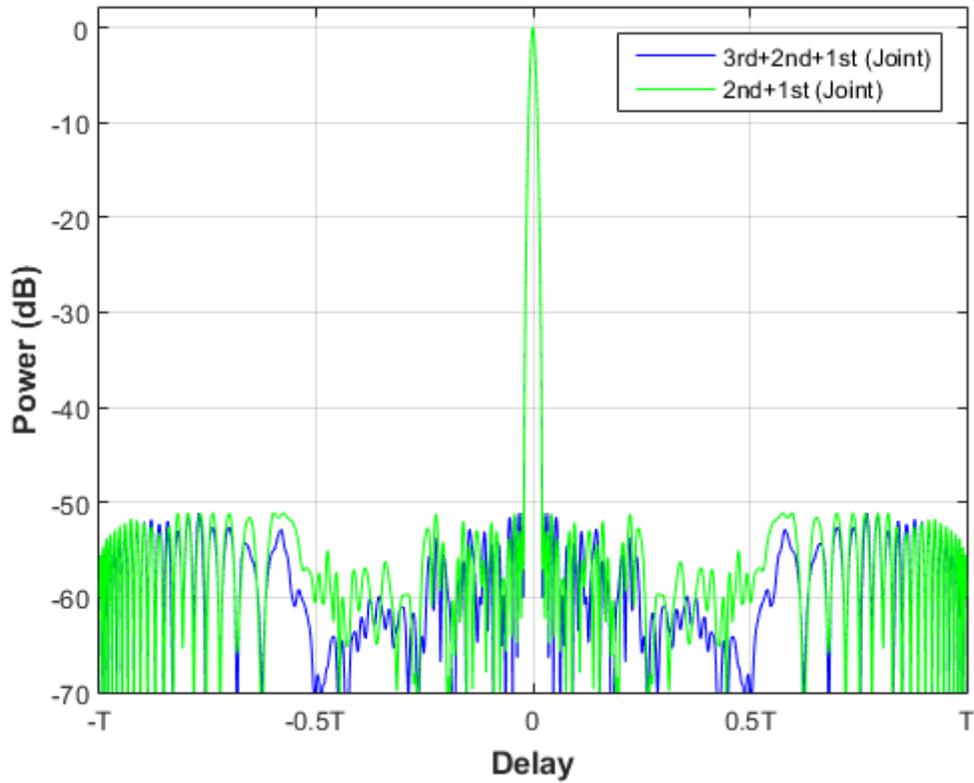


Fig. 15. Autocorrelation of jointly optimized waveforms via (52) and (53)

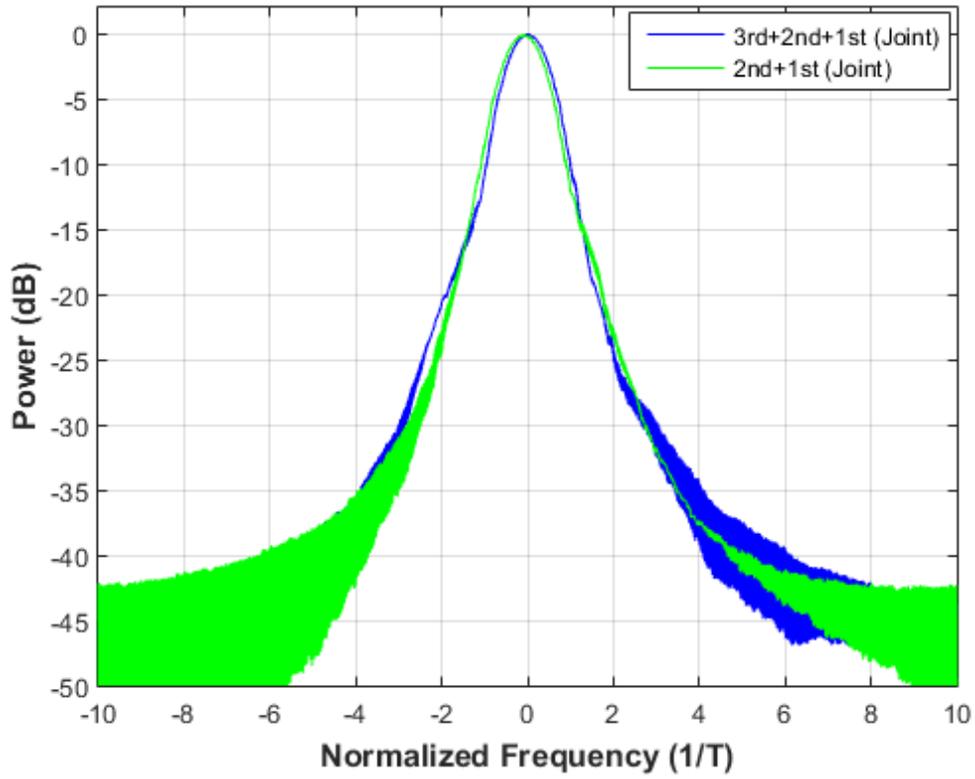


Fig. 16. Spectral content of jointly optimized waveforms via (52) and (53)

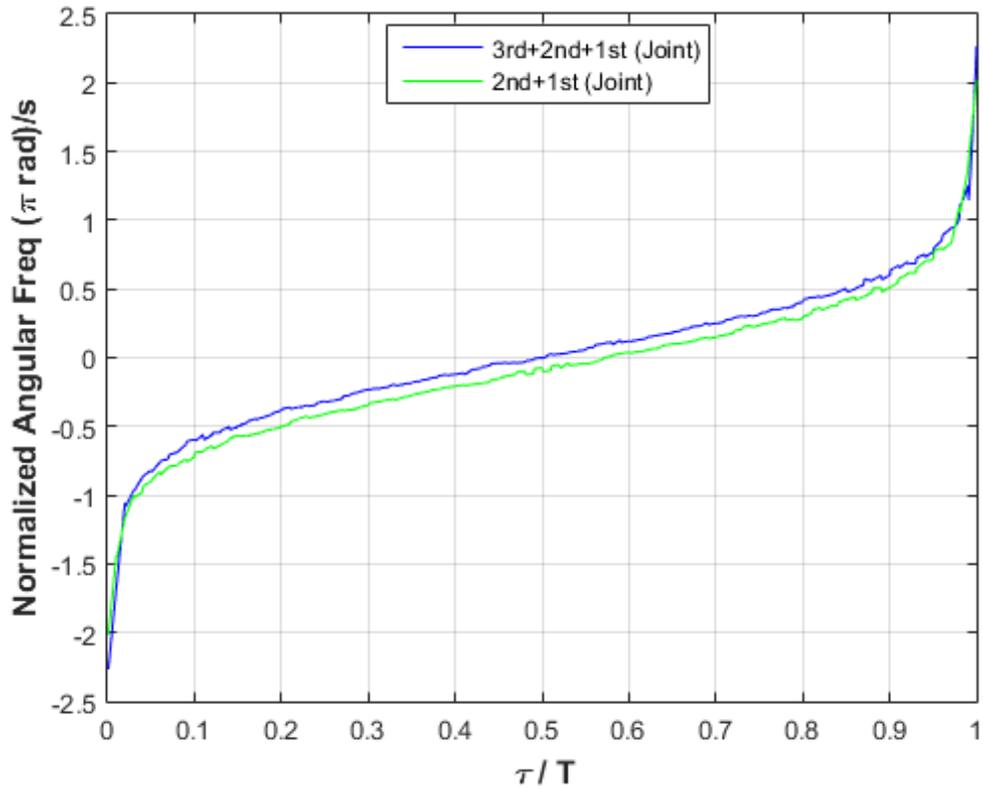


Fig. 17. Instantaneous frequency of jointly optimized waveforms via (52) and (53). Note the small perturbations provided by the first-order coding.

Table III. PSL and ISL for Sequential and Joint Optimization of multiple orders for $BT=100$

	Joint 1st & 2nd	Joint 1st, 2nd & 3rd	Sequential 1st & 2nd	Sequential 1st, 2nd & 3rd
PSL (dB)	-51.1	-51.1	-50.7	-51.2
ISL (dB)	-66.4	-67.9	-66.0	-66.8

By way of comparison with single-order implementations, Figs. 18 and 19 illustrate the optimized codes determined for the joint first/second-order implementation. It is interesting to note that the second-order joint code in Fig. 19 is quite similar to the structure of the second-order individual code depicted in Fig. 13, with the flared ends corresponding to the rapid chirping observed at the ends of the pulse in Fig. 17. However, the first-order joint code in Fig. 18 looks quite different from the first-order individual code in Fig. 9 (or 12) because it now only serves to provide a small dithering to the phase to break up sidelobe coherence instead of providing the overall waveform structure.

It is likewise noted that the theoretical limits were derived for the individual implementations and thus are not exactly applicable in this multi-order context. That said, as general guidelines they may still be useful when considering the trade-offs between different codes (such as observed for Figs. 18 and 19).

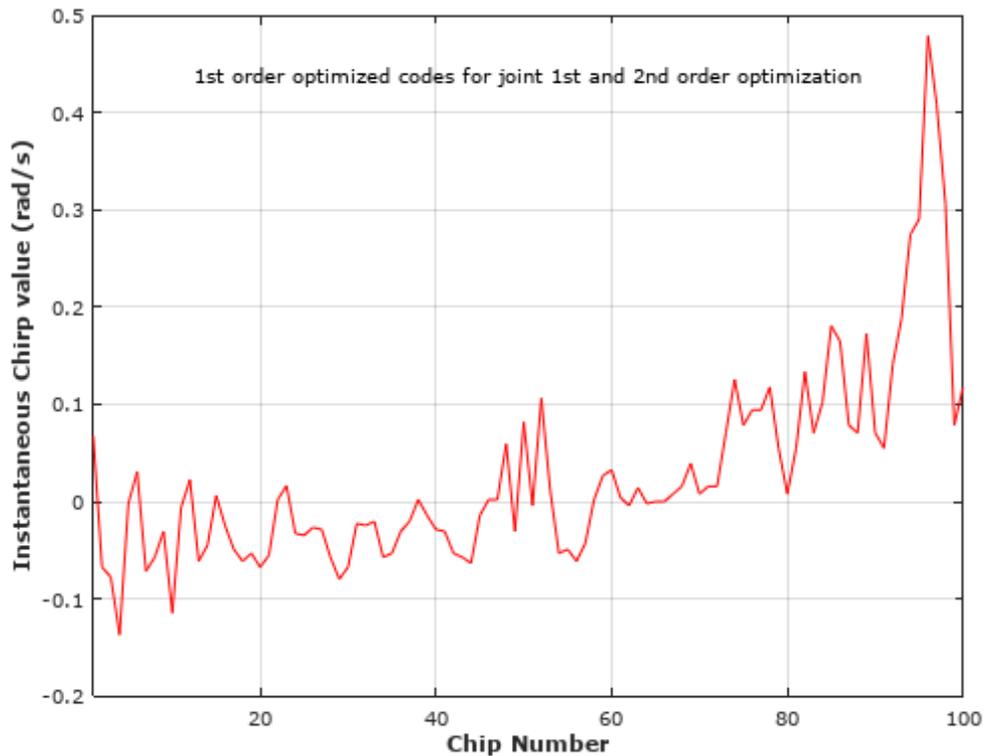


Fig. 18. First-order code coefficients for a jointly optimized first/second-order PCFM waveform

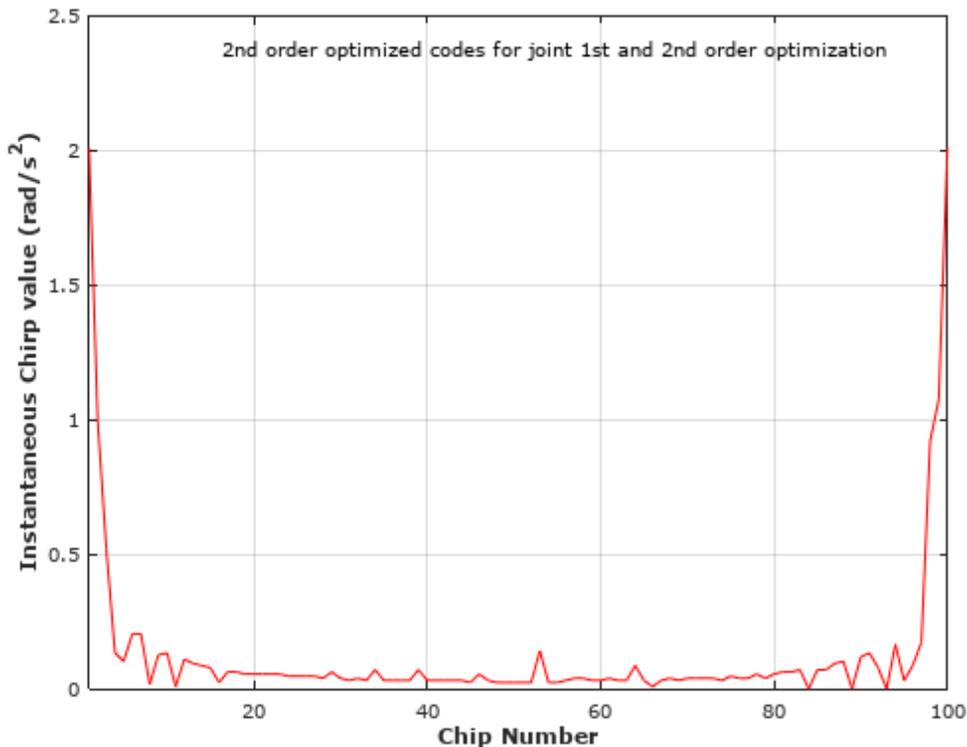


Fig. 19. Second code coefficients for a jointly optimized first/second-order PCFM waveform

C. Delay-Doppler Ambiguity Function

Finally, while the purpose of this paper is to explore these higher-order waveform implementation structures, it is interesting to examine the delay-Doppler ambiguity functions [13, 14] of the example waveforms that have been obtained from optimization. The ambiguity functions for the jointly optimized multi-order PCFM waveforms constructed from (52) and (53) are depicted in Figs. 20 and 21. As expected, based on the rather similar instantaneous frequency functions for these two waveforms (Fig. 17), the ambiguity functions are likewise quite similar in appearance, with small differences only visible across Doppler at a normalized delay of ± 0.5 . Given the fact that they had the same initialization and that the third-order component has been found to provide little benefit, this similarity is not surprising. Both exhibit the prominent delay-Doppler ridge that is typical of chirp-like waveforms and if one were to zoom out the surrounding Fresnel lobes would likewise be visible. While not to the same degree as LFM, these waveforms also still provide much of the Doppler tolerance exhibited by LFM, which is to be expected since Fig. 17 shows that they have (for the most part) time-frequency responses that are rather linear through the middle of the pulse. While there are certainly practical limits to the ambiguity function responses that can be achieved by physically realizable waveforms, the expanded design freedom afforded by these new parameterized structures may

very well enhance our ability to control the placement of delay/Doppler ambiguity in useful ways (e.g. to leverage incoherent sidelobe combining [37] and/or exploit complementary FM structures [38] across both delay and Doppler).

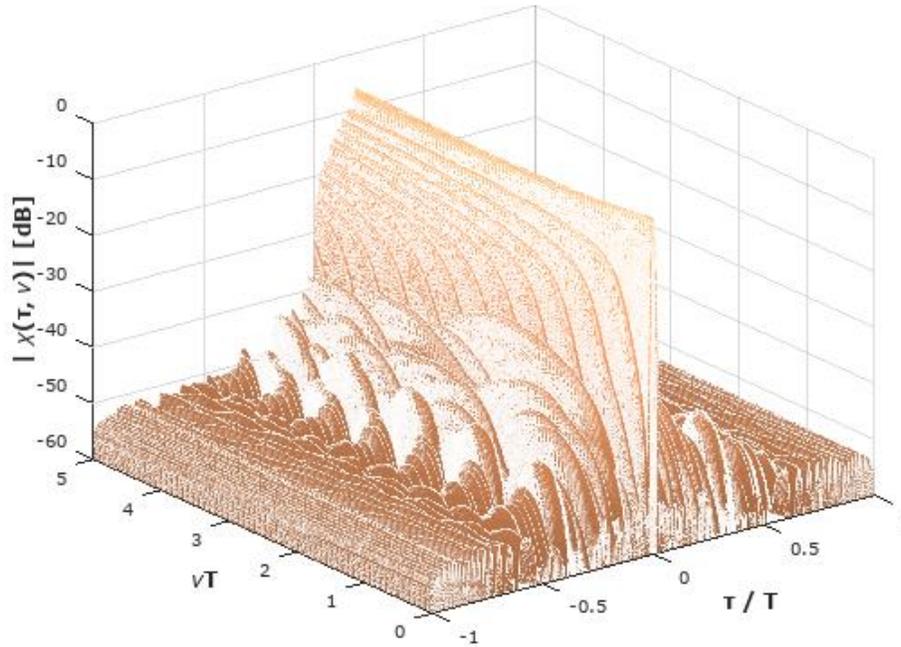


Fig. 20. Delay-Doppler ambiguity function for optimized multi-order PCFM waveform implemented via (52)

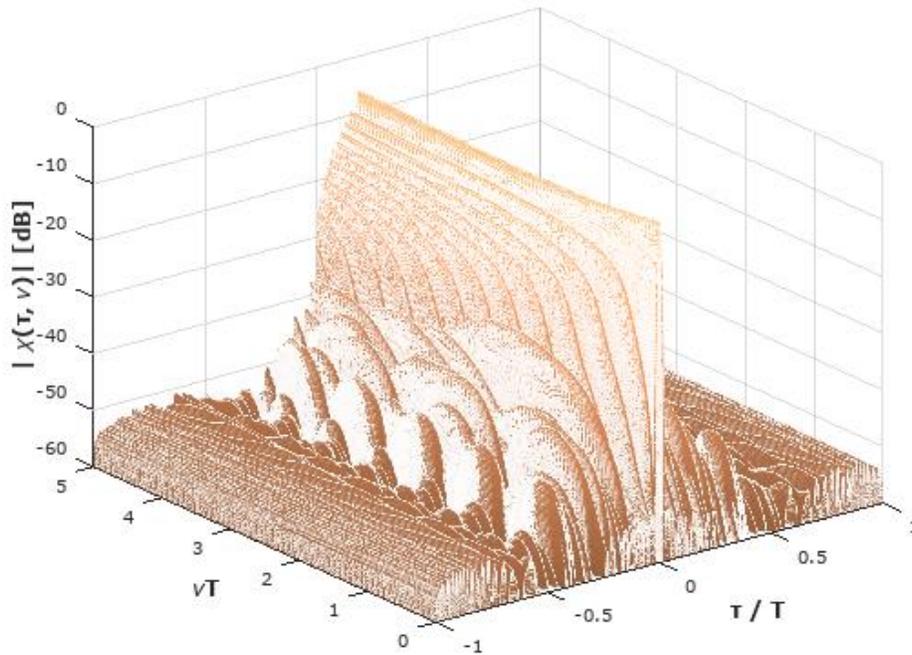


Fig. 21. Delay-Doppler ambiguity function for optimized multi-order PCFM waveform implemented via (53)

IV. CONCLUSIONS

The polyphase-coded FM (PCFM) framework for radar waveform implementation/optimization, which was previously derived from continuous phase modulation (CPM) used in communication, can be viewed as a first-order waveform representation when compared to the zeroth-order representation of a polyphase code by itself. Inspired by polynomial nonlinear FM [11], here it has been shown that higher-order schemes can also be employed as a means to obtain more degrees of freedom for physical waveform design with which to explore the continuum of possible phase trajectories. It has been observed through a variety of trials and anecdotally demonstrated here that waveforms designed using the second-order PCFM implementation tend to provide the best performance in terms of sidelobe reduction for the different single-order schemes. Multi-order implementations may likewise be employed to provide even better PSL/ISL performance, though beyond second-order does not appear to provide much additional benefit. The mathematical relationships between these different orders has also been investigated and subsequently used to derive general guidelines for selection of appropriate code values according to spectral containment constraints.

The particular optimization approach from [27] was used here to assess example waveforms obtained using these implementation structures. However, the variety and ultimate capabilities of these structures remains to be explored given the vast number of optimization strategies that could potentially be employed. Moreover, the emergence of new agile waveform-diverse modes establishes a need for greater waveform flexibility and design freedom that is readily supported by these alternative parameterizations. Further, these compact coded representations could prove useful for on-the-fly waveform design and deployment on low-cost software-defined radar platforms to facilitate cognitive radar (e.g. [52]). Finally, when combined with new optimal and adaptive receive processing schemes for FM waveforms such as [56-59] truly outstanding sensitivity and flexibility can be achieved for practical sensing applications.

APPENDIX A

The following MATLAB™ scripts provide the means to generate waveforms via the PCFM and HO-PCFM implementations defined in Section II. Because they are parameterized with codes whose lengths are on the order of BT , a diverse set of physically realizable waveforms can be produced from a compact set of parameters that may be optimized via arbitrary means. Note that the initial phase terms are omitted since a resulting waveform can simply be phase-rotated to achieve any desired starting phase.

To use these schemes with existing phase codes, a piece-wise phase difference like that described in [25] can be employed to obtain first-order codes, and subsequently repeated to obtain higher-order codes. That said, because the resulting FM waveforms possess better spectral containment than one can expect from a phase code (i.e. a zeroth-order representation), attaining good FM waveforms would be better achieved by designing the underlying coding with the corresponding code-to-waveform implementation as part of the optimization process (see [27,28,35-38]).

The resulting discrete vector \mathbf{s} for each implementation can be prepared for loading onto an arbitrary waveform generator (AWG) in one of two ways. The value of the over-sampling factor **over** in the script can be set sufficiently high to directly produce the AWG-ready version of the discretized waveform. Alternatively, **over** can be set relatively low (but still greater than 1) and then the vector \mathbf{s} can be up-sampled using phase interpolation, though this interpolation may introduce small errors that may limit the ultimate waveform fidelity that can be achieved. While the former approach is more precise, the latter tends to be more conducive to optimization due to the lower number of samples needed to represent the quasi-continuous representation of the waveform.

MATLAB™ FUNCTION TO IMPLEMENT A FIRST-ORDER PCFM WAVEFORM

```

% a_code: first-order phase-change code, bound between  $\pm\pi$ 
% over: "over-sampling" with respect to 3 dB bandwidth, integer > 1
% freq: instantaneous frequency of the waveform

function [s, freq] = PCFM1(a_code, over)
    N = length(a_code); % length of code
    g = ones(over,1); % define rectangular shaping filter
    g1 = g./sum(g); % normalize shaping filter to integrate to unity
    train = zeros(1,over*N); % define impulse train
    train(1:over:end) = a_code; % weight impulse train with code values
    chi1 = filter(g1,1,train); % apply shaping filter to weighted impulse train
    freq = chi1.*over; % instantaneous frequency for plotting
    phi1 = filter(1,[1 -1],chi1); % integrate frequency to phase
    s = exp(j*phi1); % resulting complex baseband waveform

```

MATLAB™ FUNCTION TO IMPLEMENT A SECOND-ORDER PCFM WAVEFORM

```

% b_code: second-order frequency-change code, bound between  $\pm 2\pi$ 
% over: "over-sampling" with respect to 3 dB bandwidth, integer > 1
% omega2: second-order initial frequency (in radians/Tp), here Tp = over
% freq: instantaneous frequency of the waveform

function [s, freq] = PCFM2(b_code, omega2, over)
    N = length(b_code); % length of code
    g = ones(over,1); % define rectangular shaping filter
    g2 = g./sum(g)^2; % normalize shaping filter to integrate to unity
    train2 = zeros(1,over*N); % define impulse train
    train2(1:over:end) = b_code; % weight impulse train with code values
    chi2 = filter(g2,1,train2); % apply shaping filter to weighted impulse train
    chi2dt = filter(1,[1 -1],chi2); % integrate chirp rate to frequency
    freq = (chi2dt + omega2) .*over; % instantaneous frequency for plotting
    phi2 = filter(1,[1 -1], (chi2dt + omega2)); % integrate frequency to phase
    s = exp(j*phi2); % resulting complex baseband waveform

```

MATLAB™ FUNCTION TO IMPLEMENT A THIRD-ORDER PCFM WAVEFORM

```

% c_code: third-order chirp-change code, bound between  $\pm 4\pi$ 
% over: "over-sampling" with respect to 3 dB bandwidth, integer > 1
% omega3: third-order initial frequency (in radians/Tp), here Tp = over
% beta3: third-order initial chirp rate (in radians/N/ Tp^2)
% freq: instantaneous frequency of the waveform

function [s, freq] = PCFM3(c_code, omega3, beta3, over)
    N = length(c_code); % length of code
    g = ones(over,1); % define rectangular shaping filter
    g3 = g./sum(g)^3; % normalize shaping filter to integrate to unity
    train3 = zeros(1,over*N); % define impulse train
    train3(1:over:end) = c_code; % weight impulse train with code values
    chi3 = filter(g3,1,train3); % apply shaping filter to weighted impulse train
    chi3dt = filter(1,[1 -1],chi3); % integrate chirp acceleration to chirp rate
    chi3dtdt = filter(1,[1 -1], (chi3dt + beta3)); % integrate chirp rate to frequency
    freq = (chi3dtdt + omega3) .*over; % instantaneous frequency for plotting
    phi3 = filter(1,[1 -1], (chi3dtdt + omega3)); % integrate frequency to phase
    s = exp(j*phi3); % resulting complex baseband waveform

```

As discussed in Section II.B, an LFM up-chirp with time-bandwidth product $BT \cong N$ could be exactly realized using the second-order PCFM function above by setting $\mathbf{b_code} = 2*\pi / N.*\mathbf{ones}(N,1)$ and $\mathbf{omega2} = -\pi / \mathbf{over}$, for the value of \mathbf{over} an arbitrary integer greater than 1. Precisely the same waveform could be obtained using the third-order PCFM function by setting $\mathbf{c_code} = \mathbf{zeros}(N,1)$, the initial frequency $\mathbf{omega3} = -\pi / \mathbf{over}$, and the chirp rate to $\mathbf{beta3} = 2*\pi / N / \mathbf{over}^2$. To use the coded form of the third-order implementation from Sect. II.B would require changing the shaping filter in the MATLAB™ script. These values should provide a sense of what is required to realize arbitrary waveforms with these implementations. Scripts for the mixed-order implementations are also provided below.

MATLAB™ FUNCTION TO IMPLEMENT MIXED SECOND+FIRST-ORDER PCFM WAVEFORM

```

% a_code: first-order phase-change code, bound between  $\pm\pi$ 
% b_code: second-order frequency-change code, bound between  $\pm 2\pi$ 
% over: "over-sampling" with respect to 3 dB bandwidth, integer > 1
% omega21: initial frequency
% freq: instantaneous frequency of the waveform
% this implementation requires that a_code and b_code are the same length

function s = PCFM21(a_code, b_code, omega21, over)
    N = length(a_code); % length of code
    g = ones(over,1); % define rectangular shaping filter
    g1 = g./sum(g); % normalize 1st order shaping filter to integrate to unity
    g2 = g./(sum(g))^2; % normalize 2nd order shaping filter
    train1 = zeros(1,over*N); % define impulse train
    train2 = zeros(1,over*N); % define impulse train
    train1(1:over:end) = a_code; % weight impulse train with 1st order code values
    train2(1:over:end) = b_code; % weight impulse train with 2nd order code values
    chi1 = filter(g1,1,train1); % apply shaping filter to 1st order weighted impulse train
    chi2 = filter(g2,1,train2); % apply shaping filter to 2nd order weighted impulse train
    chi2dt = filter(1,[1 -1],chi2); % integrate chirp rate to frequency
    freq = (chi1 + chi2dt + omega21).*over; % instantaneous frequency for plotting
    phi21 = filter(1,[1 -1], (chi1 + chi2dt + omega21)); % integrate frequency to phase
    s = exp(j*phi21); % resulting complex baseband waveform

```

MATLAB™ FUNCTION TO IMPLEMENT MIXED THIRD+SECOND+FIRST-ORDER PCFM WAVEFORM

```

% a_code: first-order phase-change code, bound between  $\pm\pi$ 
% b_code: second-order frequency-change code, bound between  $\pm 2\pi$ 
% c_code: third-order chirp-change code, bound between  $\pm 4\pi$ 
% omega321: initial frequency
% beta321: initial chirp rate
% over: "over-sampling" with respect to 3 dB bandwidth, integer > 1
% freq: instantaneous frequency of the waveform
% this implementation requires that a_code, b_code, and c_code are the same length

function s = PCFM321(a_code, b_code, c_code, beta321, omega321, over)
    N = length(a_code); % length of code
    g = ones(over,1); % define rectangular shaping filter
    g1 = g./sum(g); % normalize 1st order shaping filter to integrate to unity
    g2 = g./(sum(g))^2; % normalize 2nd order shaping filter
    g3 = g./(sum(g))^3; % normalize 3rd order shaping filter
    train1 = zeros(1,over*N); % define impulse train
    train2 = zeros(1,over*N); % define impulse train
    train3 = zeros(1,over*N); % define impulse train
    train1(1:over:end) = a_code; % weight impulse train with 1st order code values
    train2(1:over:end) = b_code; % weight impulse train with 2nd order code values
    train3(1:over:end) = c_code; % weight impulse train with 3rd order code values
    chi1 = filter(g1,1,train1); % apply shaping filter to 1st order weighted impulse train
    chi2 = filter(g2,1,train2); % apply shaping filter to 2nd order weighted impulse train
    chi3 = filter(g3,1,train3); % apply shaping filter to 3rd order weighted impulse train
    chi3dt = filter(1,[1 -1],chi3); % integrate 3rd order chirp acceleration to chirp rate
    chi32dt = filter(1,[1 -1], (chi2 + chi3dt + beta321)); % integrate chirp rate to frequency
    freq = (chi1 + chi32dt + omega321).*over; % instantaneous frequency for plotting
    phi321 = filter(1,[1 -1], (chi1 + chi32dt + omega321)); % integrate frequency to phase
    s = exp(j*phi321); % resulting complex baseband waveform

```

REFERENCES

- [1] Tan, P.S., Jakabosky, J., Stiles, J.M., Blunt, S.D., "On higher-order representations of polyphase-coded FM radar waveforms," *IEEE International Radar Conference*, Arlington, VA, May 2015.
- [2] Klauder, J.R., Price, A.C., Darlington, S., Albersheim, W.J., "The theory and design of chirp radars," *The Bell System Technical Journal*, vol. XXXIX, no. 4, pp. 745-808, July 1960.
- [3] Cook, C.E., "Pulse Compression - key to more efficient radar transmission," *Proc. IRE*, vol. 48, no. 3, pp. 310-316, Mar 1960.
- [4] Cook, C.E., "A class of nonlinear FM pulse compression signals," *Proc. IEEE*, vol. 52, no.11, pp.1369-1371, Nov 1964.
- [5] Fowle, E., "The design of FM pulse compression signals," *IEEE Trans. Information Theory*, vol. 10, no. 1, pp. 61-67, Jan. 1964.
- [6] Johnston, J.A., Fairhead, A.C., "Waveform design and Doppler sensitivity analysis for nonlinear FM chirp pulses," *IEEE Proceedings for Communications, Radar & Signal Processing and its Applications*, vol. 1333, no. 2, pp. 163-175, April 1986.
- [7] Labitt, M., "Obtaining low sidelobes using non-linear FM pulse compression," *MIT Lincoln Lab Project Report*, ATC-223, Nov. 1994.
- [8] Collins, T., Atkins, P., "Nonlinear frequency modulation chirps for active sonar," *IEEE Proceedings on Radar, Sonar & Navigation*, vol. 146, no. 6, pp. 312-316, Dec. 1999.
- [9] Gladkova, I., "Design of frequency modulated waveforms via the Zak transform," *IEEE Trans. Aerospace & Electronic Systems*, vol. 40, no. 1, pp. 355-359, Jan 2004.
- [10] De Witte E., Griffiths, H.D., "Improved ultra-low range sidelobe pulse compression waveform design," *Electronics Letters*, vol. 40, no. 22, pp. 1448-1450, Oct. 2004.
- [11] Doerry, A.W., "Generating nonlinear FM chirp waveforms for radar," *Sandia Report*, SAND2006-5856, Sept. 2006.
- [12] Kurdzo, J.M., Cheong, B.L., Palmer, R.D., Zhang, G., Meier, J.B., "A pulse compression waveform for improved-sensitivity weather radar observations," *AMS J. Atmospheric & Oceanic Technology*, vol. 31, pp. 2713-2731, Dec. 2014.
- [13] Levanon, N., Mozeson, E., *Radar Signals*, John Wiley & Sons, 2004.
- [14] Blunt, S.D., Mokole, E.L., "An overview of radar waveform diversity," *IEEE AESS Systems Magazine*, vol. 31, no. 11, pp. 2-42, Nov. 2016.
- [15] Caputi, W.J., "Stretch: a time-transformation technique," *IEEE Trans. Aerospace & Electronic Systems*, vol. AES-7, no. 2, pp. 269-278, Mar. 1971.
- [16] Hemmingsen, D.M., McCormick, P.M., Blunt, S.D., Allen, C., Martone, A., Sherbondy, K., Wikner, D., "Waveform-diverse stretch processing," *IEEE Radar Conference*, Oklahoma City, OK, Apr. 2018.
- [17] Frank, R.L., "Polyphase codes with good non-periodic correlation properties," *IEEE Trans. Information Theory*, vol. 9, no. 1, pp. 43-45, Jan. 1963.
- [18] Lewis, B.L., Kretschmer, F.F., "A new class of polyphase pulse compression codes and techniques," *IEEE Trans. Aerospace & Electronic Systems*, vol. AES-17, no. 3, pp. 364-372, May 1981.
- [19] Cohen, M.N., Fox, M.R., Baden, J.M., "Minimum peak sidelobe pulse compression codes," *IEEE Intl. Radar Conf.*, Arlington, VA, May 1990.
- [20] Nunn, C.J., Coxson, G.E., "Polyphase pulse compression codes with optimal peak and integrated sidelobes," *IEEE Trans. Aerospace & Electronic Systems*, vol. 45, no. 2, pp. 775-781, Apr. 2009.
- [21] Wicks, M., Mokole, E., Blunt, S.D., Amuso, V., Schneible, R., eds., *Principles of Waveform Diversity & Design*, SciTech Publishing, 2010.
- [22] Pillai, S., Li, K.Y., Selesnick, I., Himed, B., eds., *Waveform Diversity: Theory & Applications*, McGraw-Hill, 2011.
- [23] Gini, F., De Maio, A., Patton, L.K., *Waveform Design and Diversity for Advanced Radar Systems*, IET Press, 2012.

- [24] Blunt, S.D., Cook, M., Perrins, E., J. de Graaf, J., "CPM-based radar waveforms for efficiently bandlimiting a transmitted spectrum," *IEEE Radar Conf.*, Pasadena, CA, May 2009.
- [25] Blunt, S.D., Cook, M., Jakobosky, J., de Graaf, J., Perrins, E., "Polyphase-coded FM (PCFM) waveforms: part I: implementation," *IEEE Trans. Aerospace & Electronic Systems*, vol. 50, no. 3, pp. 2219-2230, July 2014.
- [26] Anderson, J.B., Aulin, T., Sundberg, C.-E., *Digital Phase Modulation*, Plenum Press, New York, NY, 1986.
- [27] Blunt, S.D., Cook, M., Jakobosky, J., de Graaf, J., Perrins, E., "Polyphase-coded FM (PCFM) waveforms: part II: optimization," *IEEE Trans. Aerospace & Electronic Systems*, vol. 50, no. 3, pp. 2231-2242, July 2014.
- [28] Ryan, L., Jakobosky, J., Blunt, S.D., Allen, C., Cohen, L., "Optimizing polyphase-coded FM waveforms within a LINC transmit architecture," *IEEE Radar Conf.*, Cincinnati, OH, May 2014.
- [29] Blunt, S.D., McCormick, P., Higgins, T., Rangaswamy, M., "Physical emission of spatially-modulated radar," *IET Radar, Sonar & Navigation*, vol. 8, no. 12, pp. 1234-1246, Dec. 2014.
- [30] McCormick, P., Jakobosky, J., Blunt, S.D., Allen, C., Himed, B., "Joint polarization/waveform design and adaptive receive processing," *IEEE Intl. Radar Conf.*, Washington, DC, May 2015.
- [31] Jakobosky, J., McCormick, P., Blunt, S.D., "Implementation & design of physical radar waveform diversity," *IEEE AESS Systems Magazine*, vol. 31, no. 12, pp. 26-33, Dec. 2016.
- [32] Zook, G., McCormick, P., Blunt, S.D., "Fixational eye movement radar: random spatial modulation," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [33] Sahin, C., Jakobosky, J., McCormick, P., Metcalf, J., Blunt, S., "A novel approach for embedding communication symbols into physical radar waveforms," *IEEE Radar Conf.*, Seattle, WA, May 2017.
- [34] Sahin, C., Metcalf, J.G., Kordik, A., Kendo, T., Corigliano, T., "Experimental validation of phase-attached radar/communication (PARC) waveforms: radar performance," *IEEE Intl. Conf. on Radar*, Brisbane, Australia, Aug. 2018.
- [35] McCormick, P.M., Blunt, S.D., "Nonlinear conjugate gradient optimization of polyphase-coded FM radar waveforms," *IEEE Radar Conf.*, Seattle, WA, May 2017.
- [36] Tan, P.S., Stiles, J.M., Blunt, S.D., "Physically realizing an optimized sparse spectrum via joint design of a collection of FM waveforms," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [37] Mohr, C.A., McCormick, P.M., Blunt, S.D., Mott, C., "Spectrally-efficient FM noise radar waveforms optimized in the logarithmic domain," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [38] Mohr, C.A., McCormick, P.M., Blunt, S.D., "Optimized complementary waveform subsets within an FM noise radar CPI," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [39] Jakobosky, J., Blunt, S.D., Himed, B., "Optimization of "over-coded" radar waveforms," *IEEE Radar Conference*, Cincinnati, OH, May 2014.
- [40] McCormick, P.M., Blunt, S.D., "Gradient-based coded-FM waveform design using Legendre polynomials," *IET Intl. Conf. Radar Systems*, Belfast, UK, Oct. 2017
- [41] Griffiths, H., Cohen, L., Watts, S., Mokole, E., Baker, C., Wicks, M., Blunt, S., "Radar spectrum engineering and management: technical and regulatory issues," *Proc. IEEE*, vol. 103, no. 1, pp. 85-102, Jan. 2015.
- [42] Blunt, S.D., Perrins, E.S., eds., *Radar & Communication Spectrum Sharing*, IET, 2018.
- [43] Guerci, J.R., *Cognitive Radar: The Knowledge-Aided Fully Adaptive Approach*, Artech House, 2010.
- [44] Haykin, S., *Cognitive Dynamic Systems: Perception-Action Cycle, Radar and Radio*, Cambridge University Press, 2012.
- [45] Tigrek, R.C., Doyuran, U.C., "Utilization of Laurent decomposition for CPM radar waveform design," *IEEE Radar Conf.*, Ottawa, Canada, Apr./May 2013.
- [46] Zhao, D., Wei, Y., Liu, Y., "Correlation performance analysis for waveforms with spectral notches," *IET Radar, Sonar & Navigation*, vol. 11, no. 11, pp. 1644-1651, Nov. 2017.
- [47] Zhao, D., Wei, Y., Liu, Y., "PCFM radar waveform design with spectral and correlation considerations," *IEEE Trans. Aerospace & Electronic Systems*, vol. 53, no. 6, pp. 2885-2898, Dec. 2017.
- [48] Hekrdla, M., Stukovska, P., "Spectrally compact SFSK radar waveforms," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.

- [49] Jakobosky, J., Blunt, S.D., Higgins, T., "Ultra-low sidelobe waveform design via spectral shaping and LINC transmit architecture," *IEEE Intl. Radar Conf.*, Washington, DC, May 2015.
- [50] Jakobosky, J., Blunt, S.D., Himed, B., "Spectral-shape optimized FM noise radar for pulse agility," *IEEE Radar Conf.*, Philadelphia, PA, May 2016.
- [51] Zook, G., McCormick, P.M., Blunt, S.D., Allen, C., Jakobosky, J., "Dual-polarized FM noise radar," *IET Intl. Conf. Radar Systems*, Belfast, UK, Oct. 2017.
- [52] Ravenscroft, B., Owen, J.W., Jakobosky, J., Blunt, S.D., Martone, A.F., Sherbondy, K.D., "Experimental demonstration and analysis of cognitive spectrum sensing & notching for radar," *IET Radar, Sonar and Navigation*, vol. 12, no. 12, pp. 1466-1475, Dec. 2018.
- [53] Ravenscroft, B., McCormick, P.M., Blunt, S.D., Perrins, E., Metcalf, J.G., "A power-efficient formulation of tandem-hopped radar & communications," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [54] Owen, J., Blunt, S.D., Gallagher, K., McCormick, P.M., Allen, C., Sherbondy, K., "Nonlinear radar via intermodulation of FM noise waveform pairs," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [55] McCormick, P.M., Blunt, S.D., Metcalf, J.G., "Wideband MIMO frequency-modulated emission design with space-frequency nulling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 2, pp. 363-378, Mar. 2017.
- [56] Henke, D., McCormick, P., Blunt, S.D., Higgins, T., "Practical aspects of optimal mismatch filtering and adaptive pulse compression for FM waveforms," *IEEE Intl. Radar Conf.*, Washington, DC, May 2015.
- [57] McCormick, P.M., Blunt, S.D., Higgins, T., "A gradient descent implementation of adaptive pulse compression," *IEEE Radar Conf.*, Philadelphia, PA, May 2016.
- [58] Harnett, L., Hemmingsen, D., McCormick, P., Blunt, S.D., Allen, C., Martone, A., Sherbondy, K., Wikner, D., "Optimal and adaptive mismatch filtering for stretch processing," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.
- [59] McCormick, P.M., Blunt, S.D., "Shared-spectrum multistatic radar: experimental demonstration using FM waveforms," *IEEE Radar Conf.*, Oklahoma City, OK, Apr. 2018.