3.0 WAVEFORM CODING

3.1 Introduction

We now want to look at waveform coding. We specifically want to look at phase and frequency coding. Our first exposure to waveform coding was our study of LFM pulses. In that case the frequency and phase changed continuously across the pulse. In the cases we consider here, the frequency or phase changes in discrete steps across the pulse. As with LFM, the amplitude is constant across the pulse. In our studies we will subdivide the pulse into a series of subpulses, or chips, where the duration of the chips is the same. This is not a requirement but a convenience for our purposes, and in most practical applications. We then assign a different phase and/or frequency to each chip according to some rule defined by the frequency or phase coding algorithm. We assume that the amplitude of all chips is the same. This is also a “semi requirement” of most phase and frequency coding schemes in that they were developed under the assumption that the amplitudes of the chips are the same.

In most applications, the chips are adjacent. That is, the waveform has a 100% duty cycle. Again, this is not a hard requirement but is a standard to which waveform designers adhere. An exception to this is the step frequency waveform. In this case the spacing between pulses is usually large enough to permit unambiguous range operation, although this is not a hard requirement.

An example of a 7-chip, Barker coded waveform is shown in Figure 1. This figure can be considered an ideal, base-band signal in that the pulses are perfectly rectangular and the “voltage” values are normalized values of either +1 or -1.

![Figure 1 – 7-chip Barker Coded Waveform](image)

We usually resort to waveform coding when we simultaneously desire fine range resolution and high target energy. The desire for fine range resolution dictates a narrow pulse while the desire for high energy dictates a wide pulse. With properly chosen phase or frequency coding we can usually achieve both desires.

The most common form of waveform coding is LFM. LFM waveforms are reasonably easy to generate and process. They also lend themselves to reasonably straightforward analysis. In spite of this, researchers have studied, and still study, many other waveform coding methods. Interestingly, several of these are based on the LFM waveform. Examples include frequency step waveforms and Frank polyphase coded waveforms.
In here we will present several different waveform coding schemes and discuss the characteristics of their ambiguity functions. To that end, I've sent you a Matlab script that generates the ambiguity function. Some of you may already have such a program and, if so, I encourage you to use your routine.

3.2 Frank Polyphase Waveforms

We will start by examining a digital form of LFM that is termed Frank polyphase coding. In essence, a Frank polyphase code is a digital representation of a quadratic phase shift, the phase shift exhibited by LFM. Frank polyphase codes have lengths that are perfect squares; i.e. \( N = L^2 \) where \( L \) is an integer. The code can be formed by first creating an \( L \times L \) matrix of the form

\[
F_L = \begin{bmatrix}
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 1 & 2 & 3 & \ldots & L-1 \\
0 & 2 & 4 & 6 & \ldots & 2(L-1) \\
0 & 3 & 6 & 9 & \ldots & 3(L-1) \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & (L-1) & 2(L-1) & 3(L-1) & \ldots & (L-1)^2
\end{bmatrix}.
\]

(1)

Next the rows or columns are concatenated to form a vector of length \( N = L^2 \). Finally, the phase is determined by multiplying each element by

\[
\Delta \phi = 2\pi/L.
\]

(2)

We will illustrate this by an example. We consider \( L = 4 \) which produces an \( N = L^2 = 16 \) element Frank polyphase code. The Frank polyphase matrix is

\[
F_4 = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 \\
0 & 2 & 4 & 6 \\
0 & 3 & 6 & 9
\end{bmatrix}
\]

(3)

and

\[
\Delta \phi = 2\pi/4 = \pi/2.
\]

(4)

The vector of phase shifts is

\[
\phi(m) = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 0 \ 2 \ 4 \ 6 \ 0 \ 3 \ 6 \ 9] \cdot \pi/2.
\]

(5)

The resulting Frank polyphase coded waveform is

\[
u_{FP}(t) = \sum_{m=0}^{N-1} e^{j\phi(m)} \text{rect} \left( \frac{t - mt_p}{\tau_p} \right).
\]

(6)

The range resolution of \( u_{FP}(t) \) is \( \tau_p \).
The ambiguity function of a $u_{FP}(t)$ with the 16 element Frank polyphase code of Equation (5) is shown in Figure 2. In the plot of Figure 2, Doppler ranges from 0 to $1/\tau_p$ and range delay goes from $-16\tau_p$ to $+16\tau_p$ where $16\tau_p$ is the total duration of the pulse.

![Figure 2 – Ambiguity function of a 16 Chip Frank Polyphase Waveform](image)

The depiction of the ambiguity function shown in Figure 2 allows us to visualize the structure of the overall ambiguity function while still being able to visualize the matched-Doppler range cut (the plot of $\chi(\tau, f)$, the ambiguity function, vs. $\tau$ for $f = 0$ and hereinafter termed the range cut).

It will be noted that the ambiguity function of the Frank polyphase waveform exhibits some semblance of the ridge characteristic of LFM waveforms. We might have expected this since the Frank polyphase waveform is a sort of discrete version of the LFM waveform.

Figure 3 contains a plot of $\phi(m)$ (with appropriate phase unwrapping) for the 16 chip example above. It also contains a plot of the phase shift of an LFM waveform that has a $BT$ product of 16, the same as the $BT$ product of the 16 chip Frank polyphase waveform. As can be seen, the Frank polyphase waveform has approximately the same quadratic phase characteristic as an equivalent LFM waveform.
3.3 Zadoff-Chu Waveform

There are several variants on Frank polyphase (see Reference 1\(^1\)) coding. One that is almost directly derived from LFM is termed a “minimal chirp slope Zadoff-Chu” code. This code can be any length, \( N \). The phase shifts are given by

\[
\phi(m) = \begin{cases} 
\frac{\pi}{N} m^2 & N \text{ even} \\
\frac{\pi}{N} (m+1)m & N \text{ odd}
\end{cases} \quad m = 0, 1, \ldots, N-1. \tag{7}
\]

Figure 4 contains plots of \( \phi(m) \) for a 16 chip Frank polyphase, a 16 chip Zadoff-Chu coded waveform and an LFM waveform with a \( BT \) product of 16. As can be seen, the phase shift of the Zadoff-Chu waveform more closely approximates the phase shift of the LFM waveform than does the phase shift of the Frank polyphase waveform.

---


© 2014 M. C. Budge - merv@thebudges.com
Frank polyphase and Zadoff-Chu codes are called polyphase codes because the phase shifts of the various chips can take on a multitude of values. This makes their generation slightly complicated in that they require a quadratic modulator to obtain the phases. They also require that the signal processor have the ability to process complex signals. With modern hardware this is not an issue. In older hardware it posed problems.

3.4 Barker Coded Waveforms

A simplification of polyphase coded waveforms are waveforms that use only two phase shifts of 0 and \( \pi \). These are termed binary phase codes. In older hardware these were easy to generate and process since they only involved sign reversal. A common set of binary phase codes found in radar texts are the Barker codes. These codes have the interesting property that the peak sidelobe level of the range cut (the range sidelobes) is \( 1/N \). (This assumes that the peak of the ambiguity function is normalized to unity.) Although Barker codes have low range sidelobes, their sidelobe levels off of matched Doppler can be high as shown in Figure 5.
There only 7 known Barker codes. They have lengths of 2, 3, 4, 5, 7, 11 and 13. The phase shifts for the 7 codes are shown in Table 1.

Table 1 – Phase Shifts for Barker Codes

<table>
<thead>
<tr>
<th>Code Length</th>
<th>Phase Shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0 0 or 0 (\pi)</td>
</tr>
<tr>
<td>3</td>
<td>0 0 (\pi)</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 (\pi) or 0 0 (\pi) 0</td>
</tr>
<tr>
<td>5</td>
<td>0 0 0 (\pi) 0</td>
</tr>
<tr>
<td>7</td>
<td>0 0 0 (\pi) (\pi) 0 (\pi)</td>
</tr>
<tr>
<td>11</td>
<td>0 0 0 (\pi) (\pi) (\pi) 0 (\pi) 0 (\pi)</td>
</tr>
<tr>
<td>13</td>
<td>0 0 0 0 (\pi) (\pi) 0 0 (\pi) 0 (\pi) (\pi) 0 (\pi) 0</td>
</tr>
</tbody>
</table>

Figure 5 – Ambiguity Function of a 11-chip Barker Coded Waveform
The reference indicated in footnote 1 contains a tabulation of *polyphase* Barker codes that range in length from 4 to 45. As implied by their name, these waveforms use polyphase coding and exhibit peak sidelobe levels of about $1/N$.

### 3.5 PRN Coded Waveforms

Another type of binary phase coded waveforms is pseudo-random-noise, or PRN, waveforms. They derive their name from the fact that the algorithm used to generate the 0 and $\pi$ phase shifts is also used to generate pseudo random numbers. PRN waveforms are also used in spread spectrum communications and to scramble (or whiten) signals in HDTV and other digital transmission systems.

PRN waveforms most often have lengths of $N = 2^M - 1$ where $M$ is an integer. The sequence of phase shifts is determined by a feedback shift register device similar to that shown in *Figure 6*. In that figure the boxes represent binary shift register elements (flip-flops) and the adder is a modulo-2 adder. The characteristics of the sequence of ones and zeros produced by the feedback shift register are determined by which shift register stages are summed and fed back to the left-most shift register stage. Ideally, we want to choose the stages so that the sequence of ones and zeros repeats only after $N = 2^M - 1$ samples, where $M$ is the number of stages in the shift register. Such a sequence of ones and zeros is termed a *maximal length* PRN sequence.

Skolnik\(^2\) has, on page 10.20, a tabulation of feedback configurations needed to generate maximal length sequences with lengths up to $2^{20}-1$. The table contains only one feedback configuration for each code length. However, it indicates that there are generally several to many feedback configurations for each length. There are texts and web-sites that provide other feedback configurations. When searching for these you might use keywords such as “maximal length sequences” “shift register sequences”, “linear shift register sequences” and “pseudo random noise”. To generate a PRN sequence, one must initially load the shift register with any binary number except zero.

---

Figure 7 contains a plot of the ambiguity function for a 15-chip PRN waveform where the phase coding was generated with the feedback configuration from your text and an initial load of 0001. It will be noted that the range cut doesn’t have sidelobes that are as low as for the Frank polyphase or Barker waveforms. However, the ambiguity function doesn’t have the large peaks that the other two waveforms exhibit. This is a characteristic of PRN coded waveforms: their sidelobe levels are generally “okay” but not extremely low or high. Long PRN coded waveforms have ambiguity functions that approach the ideal, “thumbtack” ambiguity function.

We noted above that we generated the phase code by using an initial load of 0001. It turns out that the initial load can have a fairly significant impact on the matched-Doppler range sidelobes of the ambiguity function. It also has a lesser impact on the other range-Doppler sidelobes. The only known way to choose an initial load that provides the desired sidelobe characteristics is to experiment. As a side note: changing the initial load results in a phase coding sequence that is a cyclic shift of the original sequence. This is a property of maximal length PRN codes: they are periodic with a period of $2^{M-1}$.

![Figure 7 – Ambiguity Function of a 15-chip PRN Coded Waveform](image)
3.6 Mismatched PRN Processing

We now want to investigate a special type of processing of PRN coded waveforms that takes advantage of an interesting property of PRN codes. The property we refer to is that the circular autocorrelation of a PRN coded sequence has a value of either N or -1. With a circular correlation, when we shift the sequence to the right by n chips we take the n chips that “fall off” of the end of the shifted sequence and place them at the beginning of the shifted sequence. This is illustrated in Figure 8. In this figure we have used the 7-bi PRN code of 1001110 to generate the PRN coded sequence of -111-1-1-11.

![Figure 8 – Illustration of a 2-bit Circular Shift](image)

To perform the circular autocorrelation we make a copy of the sequence to produce two sequences. We then circularly shift one sequence by n chips, multiply the result in N chip pairs and form a sum across the N chip result. This is illustrated in Figure 9. Mathematically we can write the circular correlation as

$$ R(k) = \sum_{n=0}^{N-1} r_n r_{(n+k)} $$

(8)

where \( (m)_N \) denotes evaluation of \( m \) modulo \( N \). The interesting property of PRN coded sequences referenced above is that

$$ R(k) = \begin{cases} N & k = 0, N, 2N, \ldots \\ -1 & \text{otherwise} \end{cases} $$

(9)
We now want to apply this property to examine a special type of PRN coded waveforms. We will assume that we encode a 0 of the PRN code to a phase of 0 and a 1 of the PRN code to a phase shift of $\theta$. Thus a single PRN coded pulse corresponding to the 7-bit PRN code of Figures 8 and 9 would be as shown in Figure 10.

We assume that the transmit waveform, $u(t)$, is as shown in Figure 10. We define a matched filter that is matched to a signal $v(t)$ where $v(t)$ is a concatenation of three $u(t)$’s. Thus $v(t)$ would be as shown in Figure 11. The $t = 0$ reference points in Figures 10 and 11 are used to denote the time alignment for matched range. Thus, when the received signal (a scaled version of Figure 10) is aligned with the center of the three PRN coded pulses of Figure 11 the matched filter is matched in range to the received pulse.
The matched filter output, which is also the range cut of the cross ambiguity function of \( u(t) \) and \( v(t) \), is

\[
v_{MF}(\tau) = \left. \int_{-\infty}^{\infty} u(t) h(\tau-t) \, dt \right|_{f=0} = \left. \int_{-\infty}^{\infty} u(t) v^*(t+\tau) e^{j2\pi ft} \, dt \right|_{f=0}. \tag{10}
\]

We want to examine \( v_{MF}(\tau) \) for \( \tau = n\tau_p \) where \( \tau_p \) is the chip width and \( n \) is an integer between \(- (N-1)\) and \( N-1 \). We particularly want to examine the form of \( u(t) v^*(t+n\tau_p) \). This is illustrated in Figure 12 for the aforementioned 7-chip PRN coded waveform and \( n = 2 \).

![Diagram](image)

When we form \( u(t) v^*(t+2\tau_p) \) we get

\[
u(t) v^*(t+2\tau_p) = \sum_{k=0}^{N-1} e^{jkh} e^{-j(k+2)h} \text{rect} \left( \frac{t-k\tau_p}{\tau_p} \right) \tag{11}
\]

and

\[
v_{MF}(2\tau_p) = \int_{-\infty}^{\infty} \sum_{k=0}^{N-1} e^{jkh} e^{-j(k+2)h} \text{rect} \left( \frac{t-k\tau_p}{\tau_p} \right) \, dt = \tau_p \sum_{k=0}^{N-1} e^{j(\phi_k-\phi_{k+2})h}. \tag{12}
\]

We note from Figure 12 that \( \phi_k - \phi_{(k+2)h} \) is equal to either \( 0, \theta \) or \( -\theta \). We note further that there are 3 cases where \( \phi_k - \phi_{(k+2)h} = 0 \), 2 cases where \( \phi_k - \phi_{(k+2)h} = \theta \) and 2 cases where \( \phi_k - \phi_{(k+2)h} = -\theta \). With this we get

\[
v_{MF}(2\tau_p) = 3e^{j0} + 2e^{j\theta} + 2e^{-j\theta} = 3 + 4\cos \theta. \tag{13}
\]

It turns out that for all \( \tau_p < |\tau| < (N-1)\tau_p \).
\[ v_{MF}(\tau) = 3 + 4 \cos \theta. \] (14)

In fact, for any N-chip \((N=2^m-1)\) PRN coded waveform with \(u(t)\) and \(v(t)\) chosen by the above rule,

\[ v_{MF}(\tau) = \frac{N - 1}{2} + \frac{N + 1}{2} \cos \theta \quad \tau_p < |\tau| < (N - 1)\tau_p. \] (15)

Stated in words, the range sidelobes within \(N-1\) chips of the mainlobe have a constant value as given by Equation (15).

As an interesting extension of the above, if we choose \(\theta\) such that

\[ \frac{N - 1}{2} + \frac{N + 1}{2} \cos \theta = 0 \] (16)

or

\[ \theta = \cos^{-1} \left( \frac{1 - N}{1 + N} \right) \] (17)

we get

\[ v_{MF}(\tau) = 0 \quad \tau_p < |\tau| < (N - 1)\tau_p. \] (18)

In words, the range sidelobes within \(N-1\) chips of the mainlobe are zero. This has the potential of being useful in the situation a radar must be able to detect a very small target in the presence of a very large target, provided both targets are at the same Doppler.

Figure 13 contains a plot of the ambiguity function for the 7-chip PRN example above. In the case the phase, \(\theta\), was chosen to be

\[ \theta = \cos^{-1} \left( \frac{1 - 7}{1 + 7} \right) = \cos^{-1} (-0.75) = 138.59^\circ. \] (19)
It will be noted that the range sidelobes around the central peak are, indeed, zero. However, it will be noted that the sidelobes off of matched Doppler rise significantly. It will also be noted that the range cut contains two extra peaks. These peaks are termed range ambiguities and are due to the fact that \( u(t) \) correlates with each of the other two PRN coded pulses that make-up \( v(t) \). The range sidelobes adjacent to these range ambiguities are the normal range sidelobes associated with PRN coded waveforms.

### 3.7 Complementary Coded Pulses

The reference in footnote 1 contains a discussion of complementary pulses, or waveforms using complementary coded chips. These waveforms are not PRN-based but also exhibit zero range sidelobes around the central peak. A binary, complementary coded waveform is illustrated in Figure 14 and its ambiguity function is illustrated in Figure 15. As noted with the PRN-coded waveform, the range sidelobes around the central peak are zero. However, the sidelobes off of matched Doppler are fairly large. The region of zero sidelobe level can be extended by extending the duration of the zero part of the waveform of Figure 14.
Complementary coded waveforms derive their name from the fact that the components that makeup the waveform have range sidelobe levels that are negatives of each other. For example, the range sidelobes of the left-hand, four-pulse waveform of Figure 14 are the negative of the sidelobes of the right-hand four-pulse waveform. The fact that the waveforms are combined into a single waveform, with a dead space equal to at least the duration of each waveform, causes the sidelobes of one to add to the sidelobes of the other. Since the sidelobes have opposite signs, their sum is zero.

Figure 14 – Example of a Complementary Coded Waveform

Figure 15 – Ambiguity Function of the Complementary Coded Waveform of Figure 14
3.8 Generation and Processing of Phase Coded Waveforms

To conclude our discussion of phase coded waveforms, we want to discuss, at least conceptually, how such waveforms are generated and how a matched filter might be implemented. Figure 16 contains a block diagram of a method that might be used to generate a phase coded waveform with arbitrary phase shifts on each pulse. The IF Oscillator generates quadrature signals at some intermediate frequency. The output of the synchronizer consists of two sequences of pulses where one sequence has amplitudes that depend upon the cosine of the phases on the individual pulses and the other sequence has amplitudes that depend upon the sine of the phases on the individual pulses. For the case of the complimentary coded waveform of Figure 14, the \( \cos \phi_k \) would appear as shown in Figure 14 and the \( \sin \phi_k \) output would be zero. If the phases of the were other than 0 and \( \pi \) the \( \sin \phi_k \) output would be non-zero.

The left two mixers of Figure 16 are termed baseband mixers and, for all practical purposes, are multipliers. The output of the top mixer is \( \cos \phi_k \cos \omega_{IF} t \) and the output of the bottom mixer is \( \sin \phi_k \sin \omega_{IF} t \). The output of the summer is

\[
\cos \phi_k \cos \omega_{IF} t - \sin \phi_k \sin \omega_{IF} t = \cos(\omega_{IF} t + \phi_k).
\]

As can be seen from Equation (20), the combination of the IF oscillator, synchronizer, two mixers and summer produce an IF signal that has the proper phase modulation imposed upon it.

![Figure 16 – Signal Generator for Phase Coded Waveforms](image)

The right mixer is a single-sideband mixer that produces the carrier signal with the proper phase modulation.
A block diagram of a conceptual matched filter for a phase coded pulse is shown in Figure 17. The mixer in the upper left heterodynes the received signal, with the phase modulation, to some IF. The IF signal is then processed by a matched filter matched to a single chip. The output of the matched filter is then converted to digital data via a complex ADC. The ADC samples once per chip. That is, the spacing between samples is \( \tau_p \). The samples are then further compressed by the pulse compressor.

![Figure 17 – Conceptual Matched Filter for a Phase Coded Waveform](image)

The pulse compressor first converts the digital word to a complex number and feeds it to an N-1 stage digital delay line (a multi-bit, N-1 stage shift register). The outputs of the various stages of the digital delay line are multiplied by complex exponentials with negative phase shifts. It will be noted that the order of the phase shifts is reversed. This, in effect, is an implementation of \( v^*(t) \). Finally, the outputs multipliers are summed to form the compressed output. It will be noted that the pulse compressor is a correlator (or convolver).

The pulse compressor of Figure 17 is designed as an all-range processor. That is, it continually operates over the entire waveform PRI. Because of this, the multiplies and the sum must all be completed within a single chip duration,
or $\tau_p$. Further, the multipliers and the summer are complex operators. Given this, very high speed digital hardware is needed to process phase coded waveforms of even moderate bandwidth.

The processing load can be relieved if the waveform is used over only a limited range window. In this case, the correlator in the bottom part of Figure 17 would be replaced with a combination of an FFT, a complex, vector multiplier, and an inverse FFT. The length of the FFT, multiplier and inverse FFT would depend upon the size of the range window, in chips and the number of chips in the phase coded waveform. Specifically, if the range window is $N_{RW}$ chips long and the phases coded waveform contains $N$ chips, or subpulses, then the minimum FFT length would be

$$N_{FFT} = 2N + N_{RW}.$$  \hspace{1cm} (21)

The factor of $2N$ is needed to account for the fact that the output of the matched filter is two times longer than the overall pulse width.

As an example, suppose one wanted to process a range window of 15 Km and was using a phase coded waveform with a bandwidth of 10 MHz and a duration of 10 $\mu$s. The 10 MHz bandwidth translates to a chip width of 0.1 $\mu$s, or 15 m. With this we get

$$N_{RW} = \frac{15000}{15} = 1000$$  \hspace{1cm} (22)

and

$$N = \frac{10 \mu s}{0.1 \mu s} = 100.$$  \hspace{1cm} (23)

From these two values we get

$$N_{FFT} = 2N + N_{RW} = 2(100) + 1000 = 1200.$$  \hspace{1cm} (24)

This means that the pulse compressor will require a 2048 point FFT and IFFT, and will require 2048 complex multiplies. The combination of the FFT, IFFT and multiply would need to be completed in one PRI. If one uses a PRI of say, 500 $\mu$s, then the operations would need to be completed in 500 $\mu$s. This is still stressing but may be easier than the all-range implementation of Figure 17. In some applications where latency is not a major issue, the FFT, multiply and IFFT operations can be pipelined to ease computing requirements. In such an application the FFT would be performed on one PRI, the multiply on the next PRI and the IFFT on the third PRI. Further, while the IFFT is being performed on the $K^{th}$ pulse, the multiply would be performed on the $(K+1)^{th}$ pulse and the FFT would be performed on the $(K+2)^{th}$ pulse.

If the phase coding is restricted to binary, with phase shifts of 0 and $\pi$, rather than polyphase, the generation and processing of the phase coded waveforms becomes considerably simpler. In Figure 16 the sine channel can be eliminated and in the pulse compressor the arithmetic becomes real instead of complex. In some instances the arithmetic is further simplified to single-bit arithmetic.
3.9 Step Frequency Waveforms

3.9.1 The Basics

We now want to discuss some of the math behind, and properties of, step frequency waveforms. The transmit signal we are interested in is shown in Figure 18. For this analysis, and in most practical applications, we assume that the radar operates unambiguously in range. That is, the signal from pulse k is received before pulse k+1 is transmitted. Thus, we can think of processing one pulse at a time and saving the results for later, further, processing. This is similar to the approach we used when we studied SAR.

![Figure 18 – Step Frequency Waveform](image)

We assume that the frequency, \( f_k \), of the k\textsuperscript{th} pulse is given by

\[
f_k = f_o + k\Delta f
\]

where \( f_o \) is the carrier frequency and \( \Delta f \) is the frequency step. For now we assume that the individual pulses are unmodulated. With this we can write the normalized transmit signal for the k\textsuperscript{th} pulse as

\[
v_{tk}(t) = e^{j2\pi(f_o + k\Delta f)t} \text{rect} \left[ \frac{t}{\tau_p} \right].
\]

In this equation, \( t = 0 \) for each pulse is referenced to the center of the pulse. This is important in terms of implementation because it carries the tacit assumption that the signal generator that creates the transmit and heterodyne signals are perfectly coherent, i.e. derived from the same source.

The normalized signal returned from a target at a range delay of \( \tau_R \) is

\[
v_{rk}(t) = e^{j2\pi(f_o + k\Delta f)(t-\tau_k)} \text{rect} \left[ \frac{t-\tau_R}{\tau_p} \right].
\]

We assume that we know \( \tau_k \) well enough to be able to sample the matched filter output near its peak. A more accurate measurement of \( \tau_R \) will be obtained from the output of the step waveform signal processor. For now we assume that the radar and target are fixed so that \( \tau_R \) is constant.
The heterodyne signal is given by
\[ h_k(t) = e^{j 2 \pi f_o t + j k \Delta f t}. \]  

(28)

We note that the frequency of the heterodyne signal is different for every pulse and that \( h_k(t) \) is perfectly coherent with \( v_{jk}(t) \). With this, the output of the heterodyne operation is
\[ v_{hk}(t) = v_{rk}(t) h_k^*(t) = e^{-j 2 \pi f_o t} e^{-j 2 \pi k \Delta f t} \text{rect} \left[ \frac{t - \tau_R}{\tau_p} \right]. \]  

(29)

The first term is a constant phase shift that will be common to all pulses. We will lump it into some constant that we will drop.

For the next step we pass \( v_{hk}(t) \) through a matched filter matched to \( \text{rect} \left[ \frac{t}{\tau_p} \right] \) to produce a normalized output of
\[ v_{mk}(t) = e^{-j 2 \pi k \Delta f t} \text{tri} \left[ \frac{t - \tau_R}{\tau_p} \right]. \]  

(30)

where \( \text{tri}[x] \) is a triangle centered at \( x = 0 \) with a base width of 2 and a height of unity.

Finally, we sample \( v_{mk}(t) \) at some \( \tau \), close to \( \tau_R \), to obtain
\[ v_{mk}(\tau) = e^{-j 2 \pi k \Delta f \tau} \text{tri} \left[ \frac{\tau - \tau_R}{\tau_p} \right]. \]  

(31)

After we obtain \( v_{mk}(\tau) \) from N pulses we form the sum
\[ V(\tau - \tau_R) = \sum_{k=0}^{N-1} a_k v_{mk}(\tau) = \text{tri} \left[ \frac{\tau - \tau_R}{\tau_p} \right] \sum_{k=0}^{N-1} a_k e^{-j 2 \pi k \Delta f \tau} \]  

(32)

where the \( a_k \) are complex weight coefficients that we choose to maximize \( |V(\tau - \tau_R)| \). We recognize Equation (32) as the form of the sum we have encountered in our antenna, stretch processing and SAR analyses. We can use this knowledge to postulate a form of the \( a_k \) as
\[ a_k = e^{j 2 \pi k \Delta f \tau} \]  

(33)

and write \( V(\tau - \tau_R) \) as
\[ V(\tau - \tau_R) = \text{tri} \left[ \frac{\tau - \tau_R}{\tau_p} \right] \sum_{k=0}^{N-1} a_k e^{j 2 \pi k \Delta f (\tau - \tau_k)} \]  

(34)
We can evaluate and normalize Equation 34 to yield

\[ V(\tau - \tau_R) = \text{tri} \left[ \frac{\tau - \tau_R}{\tau_p} \right] \frac{\sin(N\pi\Delta f (\tau - \tau_R))}{\sin(\pi\Delta f (\tau - \tau_R))}. \]  

(35)

A plot of \(|V(\tau - \tau_R)|\) vs. \((\tau - \tau_R)\Delta f\) is shown in Figure 19 for \(N = 10\) and without the \(\text{tri}[x]\) function. It will be noted that the central peak occurs at \((\tau - \tau_R)\Delta f = 0\) and that the first null occurs at \(|(\tau - \tau_R)\Delta f| = 1/N = 0.1\). The other peaks, which are range ambiguities, are located at integer values of \((\tau - \tau_R)\Delta f\). This tells us that the range resolution of the waveform is

\[ \Delta \tau = \frac{1}{N\Delta f} \]  

(36)

and that the range ambiguities are located at

\[ \tau_{amb} = \frac{k}{\Delta f}. \]  

(37)

In the above we ignored the \(\text{tri}[x]\) function to emphasize the location of range ambiguities. If we now include it, we can quantify the effect of the single-pulse matched filter on \(|V(\tau - \tau_R)|\). We will add the extra step of recognizing that \(V(\tau - \tau_R)\) is the matched-Doppler range cut of the ambiguity function of \(v_R(t)\) and, in future references, use \(V(\tau - \tau_R) = \chi(\tau - \tau_R, 0)\). With this, Figure 20 contains plots of \(|\chi(\tau - \tau_R, 0)|\) for \(\Delta f \tau_p = 0.5, 1\) and 2. The top plot
corresponds to the case of $\Delta f \tau_p = 0.5$ and the bottom plot corresponds to the case of $\Delta f \tau_p = 2$. The dashed triangles are the single-pulsed matched filter responses.

It will be noted that for $\Delta f \tau_p = 0.5$ and 1 the single-pulse matched filter response attenuates the range ambiguities. However, when $\Delta f \tau_p = 2$ the range ambiguities are present. This interaction of the single-pulse matched filter response is a limitation that must be considered when designing step frequency waveforms. Specifically, if one increases $\Delta f$ in an attempt to improve range resolution one runs the risk of introducing range ambiguities. From this we see that, for a given $N$ there is direct relation between $\Delta f$ and the width of the individual pulses. To improve resolution by increasing $\Delta f$, and avoid range ambiguities, we must assure that

$$\Delta f \tau_p \leq 1. \quad (38)$$

It turns out that a means of effectively reducing $\tau_p$ is to use a pulse compression waveform on the individual pulses. In that case $\tau_p$ would be the compressed pulse width. Another means of improving range resolution would be to increase the number of pulses that are processed. However, this must be done with care in that it can have negative consequences in terms of timelines and the potential impact of target motion.
3.9.2 Doppler Effects

We now want to consider the effects of target motion. For now we will be concerned with only target Doppler. To include target Doppler, we write the target range as

\[ R(t) = R_0 + \dot{R} t \]  

and the target range delay as

\[ \tau_a(t) = \tau_{R0} + \left(2\dot{R}/c\right)t = \tau_{R0} + \left(f_d/f_o\right)t. \]

We can write the target range delay at the time of the \( k^{th} \) transmit pulse as

\[ \tau_a(kT) = \tau_{Rk} = \tau_{R0} + \left(f_d/f_o\right)kT \]

where \( T \) is the PRI (see Figure 18).

If the transmit signal defined in Equation (26) then the received signal is

\[ v_{Rk}(t) = e^{j2\pi(f_o + k\Delta f)(t - \tau_{Rk})/\tau_p} \text{rect}\left[ \frac{t - \tau_{Rk}}{\tau_p} \right] \]

\[ = e^{j2\pi(f_o + k\Delta f)(t - \tau_{R0} - (f_d/f_o)kT)/\tau_p} \text{rect}\left[ \frac{t - \tau_{R0}}{\tau_p} \right]. \]

If we manipulate the above using \( (f_o + k\Delta f)(f_d/f_o) \approx f_d \) with \( \tau_{Rk} \approx \tau_{R0} \) in the \text{rect}[x] function we get

\[ v_{Rk}(t) = e^{j2\pi(f_o + k\Delta f)T_{\text{PRI}}/\tau_p} e^{-j2\pi f_d T_{\text{PRI}}} \text{rect}\left[ \frac{t - \tau_{R0}}{\tau_p} \right]. \]

We note that the approximation of \( (f_o + k\Delta f)(f_d/f_o) \approx f_d \) may not be very good. However, it allows us to focus directly on the effects of target Doppler and not be concerned with the potential smearing effects the approximation could cause. This would need to be considered in a more complete analysis.

If we compare Equation 43 with Equation (27) we note that the only difference is the appearance of the term related to Doppler. Thus, if we repeated the heterodyning and matched filtering math from above we would get

\[ v_{Mk}(\tau - \tau_{R0}) = e^{-j2\pi k\Delta f \tau_{R0}} e^{-j2\pi f_d k T} \text{tri}\left[ \frac{\tau - \tau_{R0}}{\tau_p} \right]. \]

If we form a weighted sum of the \( v_{Mk}(\tau - \tau_{R0}) \) as we did before we would get

\[ V(\tau - \tau_{R0}) = \sum_{k=0}^{N-1} v_{Mk}(\tau) = \sum_{k=0}^{N-1} b_k e^{-j2\pi(\Delta f \tau_{R0} + f_d k T)} \text{tri}\left[ \frac{\tau - \tau_{R0}}{\tau_p} \right]. \]
As we did earlier, we want to choose the $b_k$ to maximize $|V(\tau - \tau_{R0})|$. However, a more general form would be to use
\[
b_k = e^{j2\pi(\Delta f + f_d)\tau}.
\] (46)
which would yield
\[
\chi(\tau - \tau_{R0}, f - f_d) = \text{tri}\left(\frac{\tau - \tau_{R0}}{\tau_p}\right) \sum_{k=0}^{N-1} e^{j2\pi((\tau - \tau_{R0})\Delta f + (f - f_d)\tau)p}.
\] (47)
Or, evaluating the sum,
\[
\chi(\tau - \tau_{R0}, f - f_d) = \text{tri}\left(\frac{\tau - \tau_{R0}}{\tau_p}\right) \sin\left(N\pi\left((\tau - \tau_{R0})\Delta f + (f - f_d)T\right)\right) \sin\left(\pi\left((\tau - \tau_{R0})\Delta f + (f - f_d)T\right)\right).
\] (48)

Figure 21 contains a plot of $\chi(\tau - \tau_{R0}, f - f_d)$ for the 10-pulse waveform discussed earlier and a PRI of 500 $\mu$s. We further use $\tau_p = 1 \mu$s and $\Delta f = 1$ MHz so that $\tau_p \Delta f = 1$. It will be noted that the ambiguity function has the ridge associated with LFM waveforms. This is expected because the step frequency waveform is a discrete representation of a waveform with linear frequency modulation.

![Figure 21 – Ambiguity Function of a Step Frequency Waveform](image-url)
Figure 22 contains a matched-range, Doppler cut plot, a plot of \( \chi(0, f - f_d) \) vs. \( f - f_d \), of the step frequency waveform. It will be noted that the Doppler resolution of this waveform is 20 KHz or \( 1/NT \), as expected.

![Figure 22 – Matched-Range, Doppler Cut for a Step Frequency Waveform](image)

Figure 23 contains plots of range cuts at matched Doppler and at a Doppler offset of one Doppler resolution cell (200 Hz). It will be noted that a Doppler offset of one Doppler resolution cell causes a range error of one range resolution cell. This means that the step frequency waveform is very sensitive to Doppler and that, if we want accurate absolute range measurement, the range shift due to target Doppler must be removed. This can be done if the target is in track and the relative velocity between the radar and target is known with reasonable accuracy.

If the step frequency waveform is used in its more common role of target imaging (or SAR), the various scatterers of the target should be moving at about the same range-rate so that range errors due to Doppler differences of the scatterers should be small. One would still want to remove the gross Doppler so as to minimize losses due to Doppler mismatch. (It will be noted from Figure 23 that the range cut at \( f = 1/NT \) is down about 1 dB (-10log(0.9)).)
Figure 23 – Range Cuts of a Step Frequency Waveform