Phase Coded Waveforms for Radar

Richard M. Davis
Ronald L. Fante
Richard P. Perry

The MITRE Corporation
Bedford, MA, 01730

Abstract – This paper focuses on the use of phase coded waveforms for radar. The authors present tradeoffs between types of codes, matched and mismatched filters, loss in signal-to-noise ratio, time sidelobe levels, and Doppler intolerance. A multiple hypothesis approach is introduced for countering Doppler intolerance and a frequency stepped waveform is presented for use in applications requiring large pulse compression gain.

Index Terms – Phase coded waveforms, mismatched filters, Doppler intolerance, multiple hypotheses, phase-coded frequency-stepped waveform

NOMENCLATURE

ISLR	Integrated sidelobe ratio
LFM	Linear frequency modulated
PC	Phase coded
PCFSW	Phase coded frequency stepped waveform
PSLR	Peak sidelobe ratio
RMSSLR	Root mean square sidelobe ratio

I. Introduction

Although the linear frequency modulated (LFM) waveform continues to be the work horse of modern radars, there is growing interest in the use of phase coded (PC) waveforms. The PC waveform allows the radar to operate as a communications terminal sending and receiving short messages. Phase coding can also be used to reduce radio frequency interference (RFI) between adjacent radars. PC waveforms divide the pulse into N time segments, referred to as chips, and apply a different phase to each. Binary phase codes limit the chip phase to 0 or π, while polyphase codes support more levels.

PC waveforms have their shortcomings, notably computational complexity, high time sidelobes, and Doppler intolerance. Pulse compression of PC waveforms can become computationally burdensome in wideband modes. PC waveforms preclude the use of spectral windowing functions to reduce time sidelobes, and Doppler introduces a phase slope across the pulse destroying it’s coherence.

Key, Fowle, and Haggarty were among the first to suggest using mismatched filters to lower time sidelobes in PC waveforms [1]. The approach was later generalized by Ackroyd and Gahni [2]. Baden and Cohen proposed a modification for controlling the peak sidelobe level [3]. Baden and Cohen also looked at sidelobe levels of different phase codes which use matched filters [4]. The prior art focuses on short PC waveforms for use in communication systems. Our focus is on long codes applicable to radar. We present tradeoffs between the type of code, code length, mismatched filter length, loss, and sidelobe level. Similar trade studies are presented relating to Doppler intolerance and an approach for countering the intolerance using multiple hypotheses is presented. In addition, a stepped frequency PC waveform is suggested for use in applications requiring large pulse compression gains (large time-bandwidth products).

In Section II we discuss measures of performance to be used in comparing different PC waveforms. In Section III we derive mismatched pulse compression filters for PC waveforms, where the filters are designed to lower integrated and peak time sidelobes. The performance measures for random and cyclic PC codes are compared in Section IV. The Doppler intolerance issue is reviewed in Section V and the use of multiple hypotheses is discussed for countering the problem. Section VI presents a phase coded frequency stepped waveform for use in applications requiring large pulse compression gain. The work is summarized in Section VII.

II. Mismatched Filters and Measures of Performance

It is well known that a matched filter (a passive network whose frequency response equals the conjugate of the transmitted radar pulse) provides the highest possible signal-to-noise ratio (SNR), when the background noise is Gaussian [5]. Although the matched filter guarantees maximum SNR against Gaussian backgrounds, it leaves the designer with high time sidelobes. For an LFM waveform, these sidelobes are generally controlled using a frequency weighting function.

The matched filter for a PC waveform is a noisy signal that produces relatively large time sidelobes. The PC waveform has large phase discontinuities which preclude the use of windowing with an appropriate weighting function to reduce time sidelobes. Mismatched filters, therefore, have been proposed for use with PC waveforms to lower time sidelobes. Designs can be found in the literature to lower both integrated and peak sidelobe levels [2,3]. Performance measures for various combinations of phase coding and mismatched filtering include integrated, peak, and rms sidelobe ratios (*ISLR*, *PSLR*, and *RMSSLR*), and SNR loss (*Loss)* relative to matched filtering. We define the performance measures below:

$$ISLR = \frac{1}{\left|y_o\right|} \sum_{n \neq o} \left| y_n \right|^2 \qquad (1)$$

$$PSLR = \frac{\left| \left(y_n\right)_{peak\ sidelobe} \right|^2}{\left| y_o \right|^2} \qquad (2)$$

$$RMSSLR = \frac{ISLR}{M+S-2} \tag{3}$$

$$Loss = \frac{\left|y_o\right|^2 mismatched\ \ filter}{\left|y_o\right|^2 matched\ \ filter} \tag{4}$$

where y_o = peak voltage response out of filter
y_n = voltage of nth range (time) sidelobe
S = number of chips in code (code length)
M = number of coefficients in mismatched filter (filter length)

ISLR is observed to equal the ratio of the total sidelobe energy to the energy in the chip at the peak of the filter response. Similarly *PSLR* is equal to the ratio of the energy in the chip corresponding to the highest sidelobe normalized by the energy in the chip at the peak response. *RMSSLR* equals *ISLR* divided by the number of sidelobes out of the mismatched pulse compression filter. Loss is defined as the peak output of the mismatched filter divided by the peak output of the matched filter.

III. Derivation of Mismatched Filters

Letting h_n denote the coefficients of the mismatched pulse compression filter and c_n the code values, the voltage samples at the output of the filter are given by the convolution

$$y_n = \sum_{m=1}^{n} h_m c_{n-m+1} \quad \text{for } n = 1 : M+S-1 \tag{5a}$$

where $c_{n-m+1} = 0$ for $n-m+1 \le 0$ *or* $n-m+1 > S$. Switching to vector notation

$$y_n = hx_n^T \tag{5b}$$

where $h = (h_1, h_2, \dots, h_M)$
$x_n = (x_n(1), x_n(2), \dots, x_n(M))$
$x_n(i) = c_{n-i+1}$ *when* $0 < n\text{-}i+1 \le S$
$= 0$ otherwise
T = transpose

The total energy in the time sidelobes is

$$E = \sum_{n=1}^{M+S-1} w_n^2 \left|y_n\right|^2 - \left|y_K\right|^2 \tag{6}$$

where *K=(M+S)/2.* Following [3], we have allowed for a sidelobe shaping function by including the scalar weights w_n. It is assumed that w_K=1. Substituting (5b) into (6)

$$E = hRh^H \tag{7}$$

where

$$R = \sum_{n=1}^{M+S-1} w_n^2 x_n^T x_n - w_K^2 x_K^T x_K \tag{8}$$

and *H* denotes hermitian. We next minimize *E* subject to the constraint that the peak voltage out of the mismatched filter equal the peak gain out of the matched filter i.e.

$$y_K = S = hx_K^T \tag{9}$$

Minimizing (7) subject to (9) is equivalent to minimizing Γ where

$$\Gamma = hRh^H + \lambda(S - hx_K^T) = h(Rh^H - \lambda x_K^T) + \lambda S \tag{10}$$

and λ denotes Lagrange Multiplier. The optimum value of *h* (which minimizes Γ) is

$$h_{opt} = \lambda h_0 \tag{11a}$$

where

$$h_0^H = R^{-1} x_K^T \tag{11b}$$

For purposes of comparison, it is desireable to force the thermal noise power out of the mismatched filter to be the same as the thermal noise power out the matched filter, i.e. we desire

$$hh^H = S \tag{12}$$

Using (12) to solve for λ we obtain

$$h_{opt} = \left(\frac{S}{h_0 h_0^H} \right)^{\frac{1}{2}} h_0 \qquad (13)$$

The SNR at the output of the pulse compressor will be $|y_K|^2/S$.

Eqution (13) can be used to design mismatched filters which minimize the integrated time sidelobes. Baden and Cohen [3] have shown that using an iterative procedure, mismatched filters can also be designed to minimize peak sidelobe level. Their procedure entails first designing a mismatched filter with shaping weights equal to unity and then, on subsequent iterations, changing the weights to

$$w_n = w_n (|y_n| + b) \qquad (14)$$

where *b* is a real scalar bias.

Finally, we note that instead of using the convolution in (5) to calculate the output of the mismatched filter, we can use transform processing, i.e.

$$y_n = DFT^{-1} (DFT(h_n) \, x \, DFT(c_n)) \qquad (15)$$

where *DFT* denotes Discrete Fourier Transform. The size of the *DFTs* will be $M+S-1$. It is not necessary to conjugate $DFT(h_n)$, as is traditionally done, because the mismatched filter design does it for us by deriving the optimum filter.

IV. Performance Measures for Different Types of Codes

A. Random Codes

It was found that loss introduced by the mismatched filter can be lowered by diagonally loading the covariance matrix *R*. *R* is a square, hermitian, band-diagonal matrix of dimension *M*. Our loading took the form

$$Diagonal\ Load = \frac{(\alpha) trace(R)}{M} \qquad (16)$$

where α is a real scalar. Figures 1 and 2 show sensitivity of the performance measures to diagonal loading where the abscissa equals $10\log_{10}(\alpha)$. Figure 1 was generated for random polyphase codes with c_n equal to $exp(j\Psi_n)$, *and* Ψ_n uniform *0 to* 2π. Figure 2 was generated using random bi-phase codes with Ψ_n equal to *0* or π. No attempt was made to exploit the band-diagonal property when inverting *R*. An *LU Decomposition* routine found in [6] was used to solve equation (11b). Results are shown for code lengths (*S*) of 256, 512, and 1024. In all cases the length of the mismatched filter (*M*) was set equal to 3*S* and no sidelobe shaping was done (shaping weights w_n set to unity).

The performance measures shown in Figures 1 and 2 for the two code types are observed to be nearly identical. We expect to see some randomness in the performance measures due to the stochastic nature of the codes. Figure 3, for example, shows typical variation across five different random codes of length 256. The variance was observed to be significantly less for the 512 and 1024 codes (results not shown). The filter coefficients (h_n) for the polyphase codes are complex, while those for the binary codes are real. Less computation is, therefore, required to perform pulse compression when using a biphase (binary) random code compared to a polyphase code. A diagonal loading of -10 dB appears to represent a compromise between filter loss and sidelobe level. The loss numbers may seem high, but it should be noted that windowing functions used with an LFM waveform typically impose 1-2 dB loss and cause a broadening of the mainlobe response – which is not observed with PC waveforms.

Figure 4 compares compressed pulse output patterns for a 1024 bit random binary code for: (a) a matched filter, (b) a mismatched filter designed to minimize *ISLR* (shaping weights set to unity) and, (c) a mismatched filter designed to minimize *PSLR*. We obtained identical patterns whether we used Equation (5) or (15) to calculate the patterns. We found that the recursive procedure of Equation (14) acts to whiten the sidelobes. It was effective in lowering sidelobe peaks, but raised rms and integrated sidelobe levels by a small amount. We found that for the long codes used in this study, flat sidelobe responses occurred after only three iterations following the initial design in which the shaping weights were set to unity. Thus minimizing *PSLR* required formation and inversion of four covariance matrices. We also found that the flattest sidelobe responses occurred when the bias (*b*) in Equation (14) was set at a few dB below the expected peak sidelobe level.

Figure 5 plots the performance measures versus the ratio of mismatched filter length to code length (M/S) for random binary codes. The diagonal loading was set at -10 dB when generating the figure. Increasing filter length is observed to lower sidelobe responses at the expense of a small increase in loss.

B. Cyclic Codes

Cyclic codes are block codes which can be generated using shift registers with feedback [7]. The codes are often denoted (n,k) where n is the length of the code and k the number of information bits. The n-k parity bits can be used for error detection and correction. An (n,k) code can be generated using a shift register with k stages. The length of the code is related to the number of information bits by

$$n = 2^{k} - 1 \tag{17}$$

The cyclic codes are also maximal length sequences. Figure 6 shows a shift register that can be used to generate the (1023,10) code. The numbers T_1 to T_{10} denote the 10 taps on the shift register. The register would be clocked at a rate f_C, where f_C^{-1} is the width of a chip. The number of possible codes for a given k is limited to the number of factors of the polynomial x^n-1 .

The loss resulting from designing mismatched filters for the cyclic codes was found to be lower than the loss for random codes. But diagonal loading the covariance matrix did not further reduce the loss for cyclic codes – as it did for random codes. Figure 7 shows sensitivity of the performance measures for the cyclic codes to the ratio of the length of the mismatched filter to the code length (M/S), where S is given by (17). Figure 8 presents a set of compressed pulse patterns for a phase coded waveform that uses the 1023 bit cyclic code. The figure compares the output of a matched filter (left insert) with the output of a mismatched filter designed to minimize *ISLR* (center insert), and the output of a mismatched filter designed to minimize *PSLR*. The performance measures for the cyclic codes are observed to be better than those for the random codes except for the peak sidelobe level in the mismatched filter designed to minimize *ISLR*. We observed large uniformly spaced sidelobe peaks at multiples of *S* samples on either side of the peak response for all cyclic codes. In all cases, however, we were able to suppress the large lobes by minimizing *PSLR*.

It is noteworthy that the loss observed when using mismatched filters with cyclic codes is comparable to the loss due to windowing when using an LFM waveform. Figure 9 shows the pulse compressor output for an LFM pulse having a time-bandwidth product of 1024 (the pulse compression gain), where a 35 dB, nbar 4, Taylor Weighting was applied to the DFT of the reference waveform. The weighting function caused a loss of 1.1 dB and an increase in mainlobe null-to-null spacing of roughly a factor of two. We excluded the peak and one sample on each side when calculating *ISLR* for the umweighted case, and three samples on each side for the weighted case. No broadening in null-to-null spacing was observed when comparing matched and mismatched PC waveform filters. While the weighting used in LFM waveforms takes energy out of the sidelobes and puts it into the mainlobe, the use of mismatched filters with PC waveforms spreads the energy over a wider time extent without putting it into the mainlobe. The observation indicates that PC waveforms should support improved resolution and detection of small targets in close proximity to large targets compared to LFM waveforms. The observation also explains the drop in *ISLR* between the left (unweighted) and right (weighted) inserts in Figure 9. Additional mainlobe broadening of both LFM and PC waveforms will occur due to IF filtering prior to digitization. Our results do not include any effects due to IF filtering.

V. Doppler Intolerance of Phase Coded Waveforms

Target Doppler introduces a phase shift ϕ_n in each of the S chips within the uncompressed coded pulse, where

$$\phi_n = 2\pi\, f_d (n-1)\frac{T_u}{S-1} \qquad for\;\; n = 1:S \tag{18}$$

and f_d = Doppler shift, T_u= uncompressed pulse width. Figure 10 plots performance measures versus total phase shift ($2\pi f_d T_u$) introduced across the radar pulse for random binary codes. Significant degradation in sidelobe level is observed to occur for phase shifts exceeding 30-40 degrees. The sensitivity to Doppler induced phase shift for the random polyphase and cyclic codes is roughly the same as for random binary codes.

One way to mitigate the effect of Doppler is to use shorter uncompressed pulses. Assuming duty cycle is fixed, energy on target could be held constant by increasing the PRF. Increasing PRF, however, can cause a waveform that is unambiguous in range to become ambiguous.

An alternative approach for mitigating Doppler intolerance when using PC waveforms is to use multiple hypotheses. The latter approach entails adding additional pulse compression filters tuned to different Doppler frequencies. The coefficients for the new mismatched filters ($h_{p,n}$) can be obtained from those of the original filter (h_n) by imposing a linear phase term, ie

$$h_{p,n} = h_n \exp\left[\mathrm{j(n\,\text{-}1)}\; p\Delta\phi/(S-1)\right] \quad for\;\; n = 1:M \tag{19}$$

where

$$\begin{aligned}
p\Delta\phi &= 0, \pm 1 : \pm (N_H - 1)/2 \\
N_H &= \text{number of hypotheses (limited to odd number to include } p = 0\text{)} \\
\Delta\phi &= 2\pi\, f_d T_u / N_H = 4\pi V_{\max} T_u / (\lambda N_H) \\
V_{\max} &= \text{maximum expected targt velocity} \\
\lambda &= \text{wavelength at center frequency}
\end{aligned}$$

If multiple hypothesis testing is used, the output will be a bank of N range-Doppler maps. The N maps must be reduced to a single composite map by comparing the power in the same range-Doppler pixel in each of the maps. If the power does not exceed a preset threshold above the background (like 8 dB), the cell is declared to be target-free and the amplitude and phase in the test pixel in the map corresponding to the zero Doppler hypothesis ($p=0$) is placed in the composite map. Otherwise, the voltage of the pixel having the largest power is placed in the composite map. The “greatest of” algorithm described above guarantees that, even with Doppler fold-over, the largest return will be preserved. Doppler ambiguities must be resolved using the composite maps resulting from different coherent processing intervals.The question of whether or not to use constant false alarm rate processing prior to determining the “greatest of” pixel warrants further research.

Figure 11 shows performance measures for the same conditions as Figure 10, but for a case where we used three hypotheses and the “greatest of” algorithm when making the composite response. The three hypotheses corresponded to total phase shifts across the pulse of 0 and plus and minus 60 degrees. Although the authors have modeled the “greatest of” algorithm with Doppler processing included, we deleted the Doppler processing in the example shown. For each value of the abscissa in Figure 11, we used the filter output having the largest peak value $(|y_K|^2)$. We have also not shown results for negative values of phase – since the curves are symmetric about zero degrees. A significant improvement is observed in time sidelobe levels using only three hypotheses.

When using multiple hypotheses, the number of floating point operations per second (FLOPs) may be able to be reduced by reversing the order of pulse compression and Doppler processing. Figures 12a and 12b show two possible implementations for an airborne radar which does Doppler processing to detect moving targets. The traditional approach shown in Figure 12a requires duplicating all processing (pulse compression filters, Doppler processing filters, and clutter cancellation) N times, where N is the number of hypotheses. Using the reverse processing of Figure 12b, only the pulse compression filters need be duplicated N times. Displaced Phase Center Antenna (DPCA) and Space Time Adaptive Processing (STAP) are complex processing paradigms used to cancel clutter [8]. Two clutter-free range-Doppler maps are required to make angle measurements.

VI. A Phase Coded Frequency Stepped Waveform

The number of floating point operations per second (FLOPS) becomes excessive when using PC waveforms in applications requiring large time-bandwidth product waveforms. If, for example we needed 100 μsec pulse and 1000 MHz of bandwidth, the number of chips in the code would have to be 100,000. If we used a mismatched filter that was twice the code length, we would have to store 200,000 coefficients. Fortunately, in most wideband applications, the range to the target is known. Knowing the start of the range window allows us to consider using a phase coded frequency stepped waveform (PCFSW).

Sitler, et al [9] proposed using a phase coded stepped frequency waveform with interpulse frequency steps and matched filter processing. We propose stepping the frequency intrapulse and using a mismatched filter. A notional PCFSW for a single pulse is depicted in Figure 13. The pulse is divided into N (nearly) contiguous subpulses of width T where each subpulse has an instantaneous bandwidth B. The spacing between subpulses is chosen to be $2\Delta R/c$, where ΔR is the largest expected range extent of the target. The spacing ensures that returns from adjacent subpulses don’t overlap. In an aircraft detection mission, for example, we might choose ΔR equal to 45 meters - producing a spacing between pulses of 0.3 μsec. The local oscillator is stepped B Hz every T' seconds beginning at the start of the receive window, where $T' = T + 2\Delta R / c$. This allows the sampling rate to be reduced from NB complex samples per second to B. A PC waveform having total bandwidth NB and pulse width NT will have N^2BT chips, while a frequency stepped waveform with *N* subpulses and the same total bandwidth and pulse width, will require only *NBT* chips.

Using the same code on each subpulse, we will show that the composite compressed pulse waveform equals the product of the compressed subpulse with a comb function which we term the *pulse factor*. The product of the *pulse factor* with the compressed subpulse yields the compressed subpulse amplified by the pulse factor gain. The performance measures for the composite waveform will be the same as those of the subpulse, but the computational burden in performing the pulse compression. A longer mismatched filter (larger M) can be used to obtain additional reduction in time sidelobe performance if needed. The PCFSW is analyzed below. To simplify the analysis, we have assumed a point target ($\Delta R=0$).

Letting the subpulse equal s(t), the transmitted waveform is

$$v(t) = \sum_{n=0}^{N-1} s(t-nT)exp(\text{-}j2\pi(f_o - NB/2+nB)(t-nT)), \tag{20}$$

where each subpulse contains S chips, f_0 is the center frequency, N the number of subpulses, T the time extent of a subpulse, and B the subpulse bandwidth. The return from a target at range R is $v(t\text{-}\tau)$ where $\tau = 2R/c$. After down-converting with $exp(j2\pi f_n(t-nT))$, where $f_n = f_0 - NB/2+nB$

$$v_d(t) = \sum_{n=o}^{N-1} s(t-\tau-nT)\exp(j2\pi f_n\tau) \tag{21}$$

Recall that since the target is in track, the range window is short and returns from adjacent subpulses never overlap. We will derive the compressed pulse return $v_{d,n}(t)$ from pulse *n* and then sum over all *N* pulses to obtain the composite return. $v_{d,n}(t)$ can be sampled at a rate of *B* complex samples per second. Taking the Fourier Transform of $v_{d,n}(t)$

$$V_{d,n}(f) = \exp(j2\pi f_n\tau) \int_{nT+\tau}^{(n+1)T+\tau} s(t-\tau-nT)\exp(-j2\pi f\, t)dt \tag{22}$$

Letting $t' = t - \tau - nT$ and performing the integration

$$V_{d,n}(f) = \exp(j2\pi f_n\tau)S(f)\exp(-j2\pi f(\tau+nT)) \tag{23}$$

In the frequency domain the samples of $V_{d,n}(f)$ are spaced $\Delta f=1/T$, so that there are $K=BT$ samples. Next let the matched filter for $V_{d,n}(f)$ be

$$G_n(f) = H(f)\exp(j2\pi fnT)\,, \tag{24}$$

where $H(f)$ is the Fourier Transform of $h_{opt}(t)$ as given by Equation (13). Forming the product of (23) with (24)

$$Y_n(f) = S(f)H(f)\exp(j2\pi(f_n - f)\tau) \tag{25}$$

We next wish to take the inverse Fourier Transform of $Y_n(f)$. Since in practice the transform will be performed using a DFT (discrete Fourier Transform), it will be necessary to pad $Y_n(f)$ with at least $K(N\text{-}1)$ zeros, so that in the time domain the sample spacing will be $\Delta t \le 1/(NB)$ and the resolution of the composite compressed waveform will be determined by the total bandwidth (NB). Taking the inverse transform of (25)

$$y_n(t) = \rho(t-\tau)\exp(j2\pi f_n\tau) \tag{26}$$

where

$$\rho(t) = \int_{-\infty}^{\infty} H(f)S(f)\exp[j2\pi f(t-\tau)]df \tag{27}$$

is the compressed subpulse voltage. Finally, we multiply $y_n(t)$ by $P_n(t)=exp(-j2\pi nBt)$ to obtain $z_n(t)$ for each subpulse and then sum over n. It should be noted that rather than multiplying $y_n(t)$ by $P_n(t)$ in the time domain, we could have convolved the Fourier Transform of $P_n(t)$ with $Y_n(f)$ in the frequency domain. Noting that

$$\sum_{n=0}^{N-1} \exp(j2\pi\, nB(t-\tau)) \;= \exp(-j\pi(N-1)B(t-\tau))\frac{\sin \pi NB(t-\tau)}{\sin \pi B(t-\tau)} \tag{28}$$

and summing $z_n(t)$ over n

$$v_o(t) = \rho(t-\tau)\exp[j\phi(t)]\frac{\sin \pi NB(t-\tau)}{\sin \pi B(t-\tau)} \tag{29}$$

where $\phi(t) = 2\pi(f_0 - NB/2)\tau + \pi B(N-1)t$.

The output power is

$$\left|v_o(t)\right|^2 = \left|\rho(t-\tau)\right|^2 \frac{\sin^2 \pi NB(t-\tau)}{\sin^2 \pi B(t-\tau)} \tag{30}$$

Note that in the digital domain the samples of the output waveform must be spaced by 1/NB seconds or less. The latter observation explains why we padded $Y_n(f)$ with at least $S(N\text{-}1)$ zeros. A block diagram of the processing is shown in Figure 14. It is also noteworthy that we can draw an analogy between Equation (30) and the pattern of an array antenna comprised of N subarrays. The antenna pattern can be expressed as the product of a subarray pattern with an array factor. In our case the compressed pulse waveform can be decomposed into the product of a compressed subpulse with a *pulse factor*. The $\sin\pi NB(t-\tau)/\sin\pi B(t\text{-}\tau)$ pulse factor samples the compressed subpulse at intervals of B^{-1} with each sample having width $(NB)^{-1}$.

As a numerical example, assume a point target and a PC cyclic waveform consisting of 10 contiguous subpulses each having width 10 μsec, bandwidth 102.3 MHz and 1023 chips. The composite PCFSW waveform (sum of all subpulses) will have a bandwidth of 1023 MHz and a chip count of 10230. Assume all subpulses are identical and are pulse compressed separately by passing the 1023 chip subpulse through a mismatched filter of length 3069 designed to minimize *PSLR.* The compressed pattern of a single subpulse was shown in Figure 8. The composite compressed waveform resulting from carrying out the processing shown in Figure 14 is presented in Figure 15. As expected, the composite compressed waveform has the same shape as a compressed subpulse but has 10 times as many samples. Although the peak response increased by 20 dB (by N^2), SNR will increase by only 10 dB since the noise powers from each subpulse add noncoherently to raise the noise power 10 dB. Figures 8 and 14 only plot target power and do not include thermal noise. Thus the peak of the compressed subpulse shown in the right insert in Figure 8 is about 60 dB while the peak of the compressed composite waveform (sum of all 10 pulses) shown in Figure 15 is about 80 dB. The noise power in Figure 8 would be about 30 dB (due to the constraint of Equation (12)), while the noise power in Figure 15 would be about 40 dB.

Had we implemented a single long cyclic code across the entire 100 μsec pulse, the waveform would have had 102,300 chips and we would have had to clock the analog-to-digital converter at a rate of 1023 MHz – compared to having a total of 10,230 chips and clocking the converter at 102.3 MHz using the PCFSW. We end up with the same SNR as we would have had by implementing a single long code, but the sidelobe performance is not as good, in fact, the performance measures for the composite PCFSW are the same as those of a single subpulse.

VII Summary

We defined and compared performance measures for phase coded waveforms which use mismatched filters to lower time sidelobes. We considered long codes for use in radar applications. Three types of codes were examined: random polyphase codes, random binary codes, and cyclic codes. We presented sensitivity trades to diagonal loading of the covariance matrix used to design the mismatched filters, to the ratio of code length to mismatched filter length, and to phase shift across the uncompressed pulse due to Doppler. We presented a multiple hypothesis concept for compensating for Doppler intolerance. Finally, we showed that a frequency stepped phase coded waveform, consisting of multiple subpulses can be used to reduce computational burden, when using waveforms having large time-bandwidth products, but at the cost of higher time sidelobes compared to using a single long code.

References

[1] E.L. Key, E.N. Fowle, and R.D. Haggarty, “A Method of Sidelobe Suppression in Phase-Coded Pulse Compression Systems”, Technical Report No 209, MIT Lincoln Laboratory, August 1959

[2] M.H. Ackroyd and F. Gahni, “Optimum Mismatched Filters for Sidelobe Suppression”, IEEE Transactions Aerospace Electronics Systems, Vol AES-9, March 1973

[3] J.M. Baden and M.N. Cohen, “Optimal Peak Sidelobe Filters for Biphase Pulse Compression”, *IEEE International Radar Conference,* 1990

[4] J.M. Baden, M.N. Cohen, “Optimal Sidelobe Suppression for Biphase Codes,” *National Telesystems Conf,* Mar 1991

[5] D.O. North, “An Analysis of the Factors Which Determin Signal/Noise Discrimination in Pulses Carrier Systems, RCA Lab Rept PTR-6C, June 25, 1943, reprinted Proc IEEE Vol 51, July 1963

[6] W.H. Press, S.A. Teukolsky, W.T. Vettering, B.B. Flannery, “Numerical Recipes in Fortran 77”, Second Edition, Cambridge University Press, 1992

[7] W.W. Peterson, “Error Correcting Codes”, MIT Press, 1961

[8] M.I. Skolnik “Radar Handbook”, Second Edition, McGraw-Hill, 1990

[9] K.I. Sitler, M.A. Temple, R.C. Novack, J.A. Hughes, “High Range Resolution Profiling Using Phase-Coded Stepped-Frequency Waveforms, Electronic Letters, Jan 2002, Vol 38, No 1

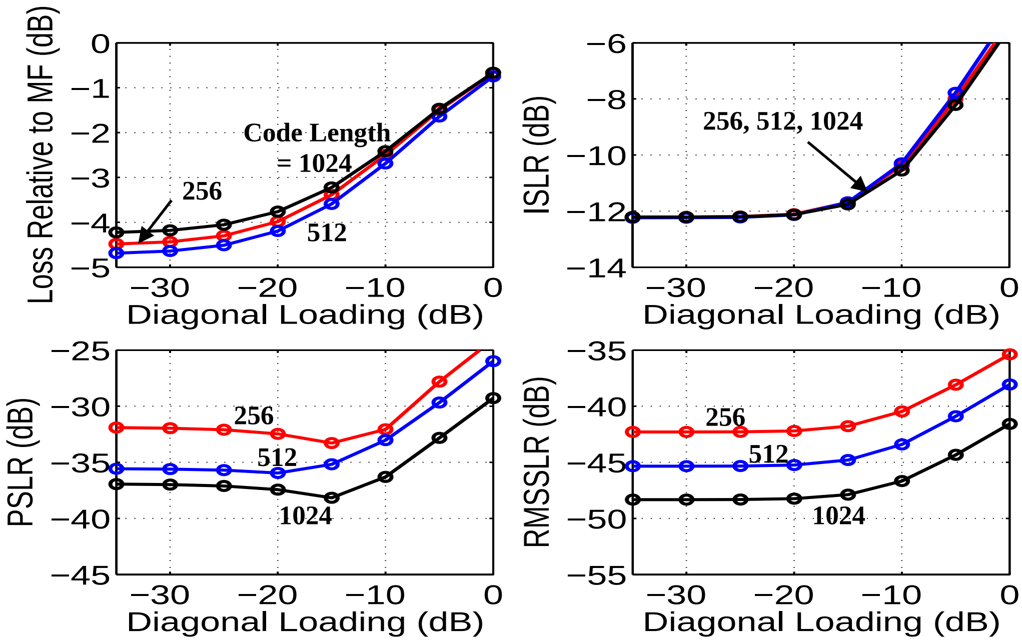


Figure 1. Performance Measures for Random Polyphase (0-2π) Codes
Designed to Minimize ISLR: M=3S

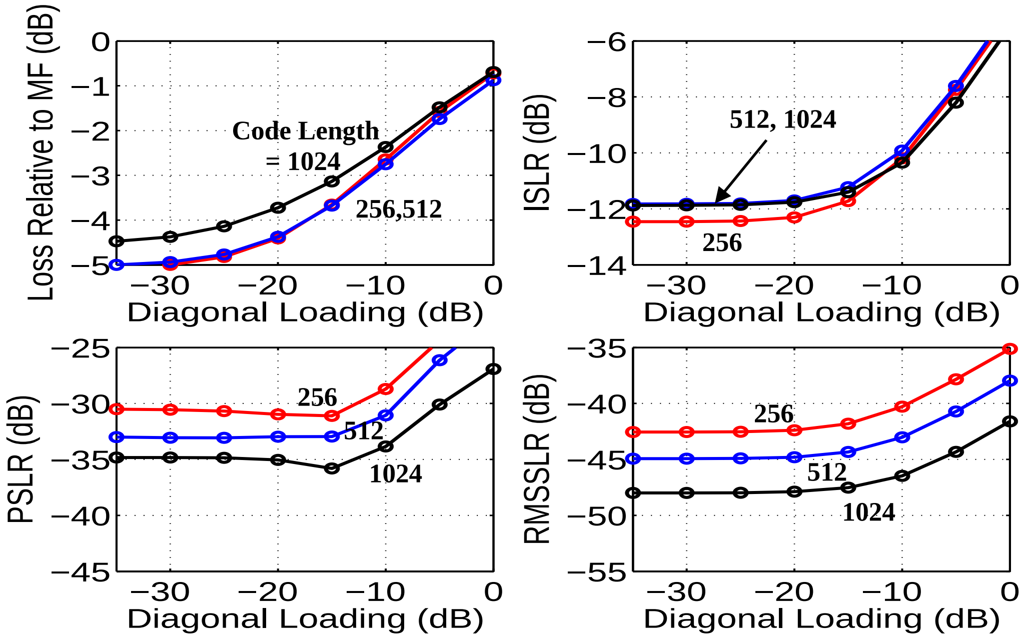


Figure 2. Performance Measures for Random Binary Codes (0 or π Phase) Designed to
Minimize ISLR: M=3S

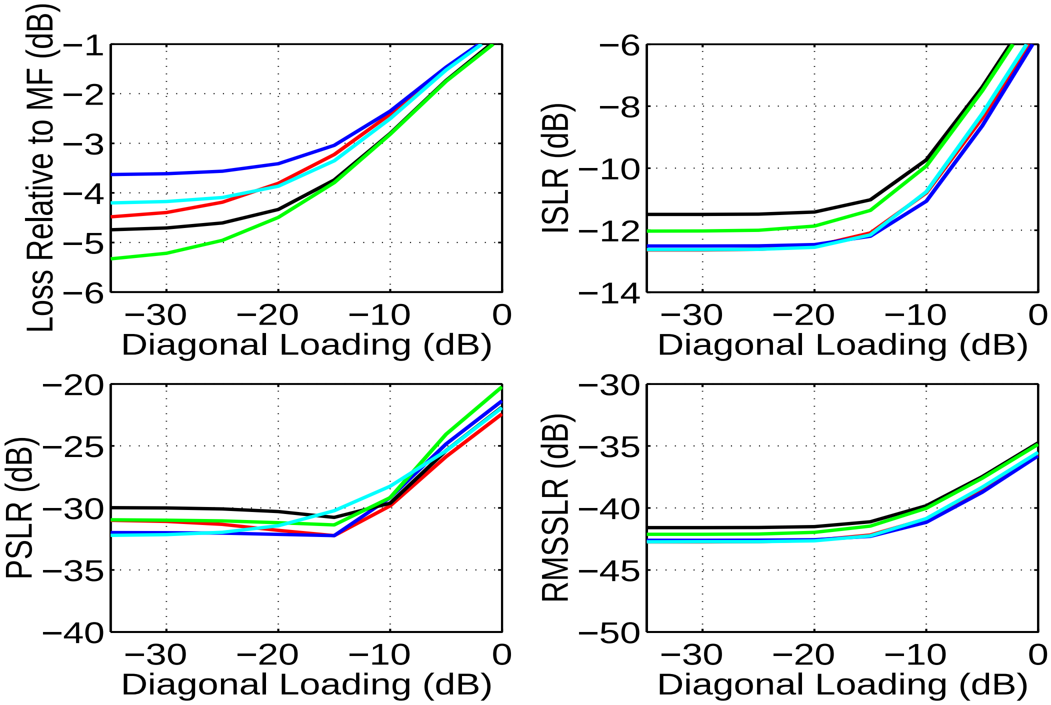


Figure 3. Performance Measures for Four Random Binary Codes (0 or π Phase) of Length 256 Designed to Minimize ISLR: M=768

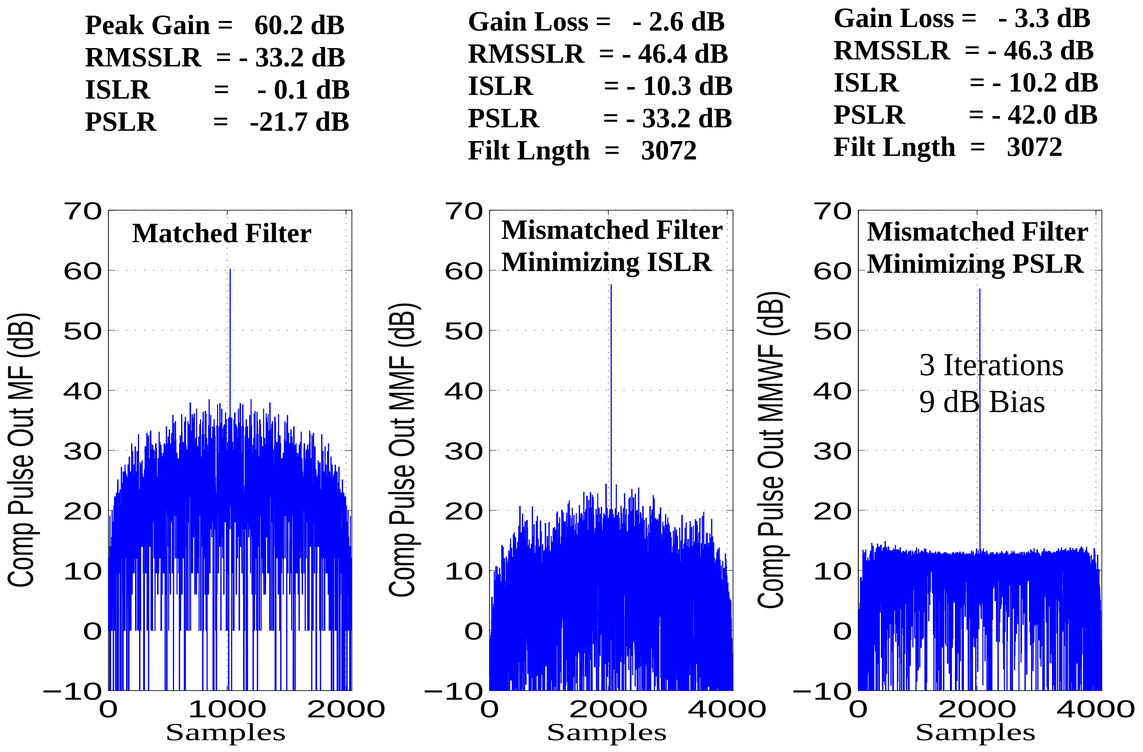


Figure 4. Comparison of Compressed Pulse Patterns for a 1024 Bit Random Binary Code

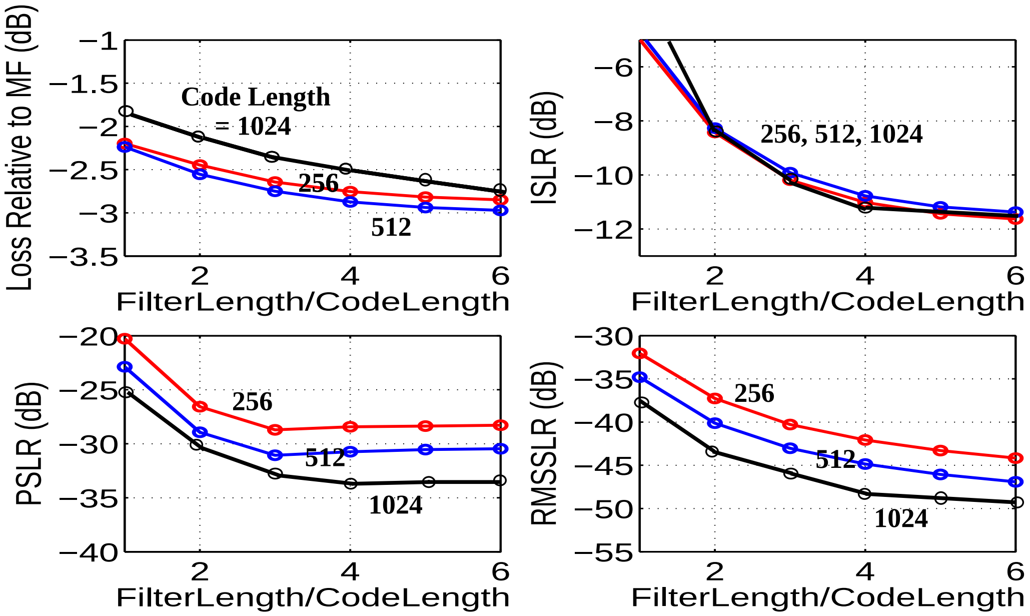


Figure 5. Performance Measures vs Ratio of Mismatched Filter Length to Code Length for Random Binary (0 or π Phase) Codes Designed to Minimize ISLR

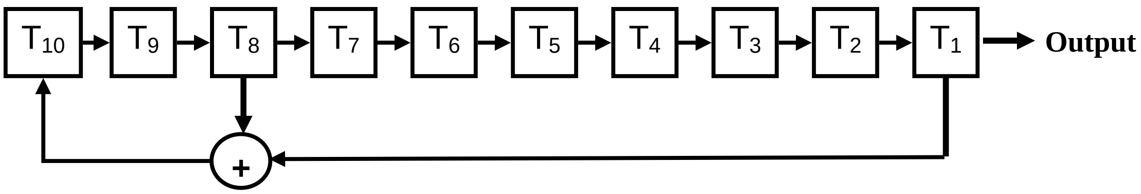


Figure 6. Shift Register Used to Generate (1023,10) Cyclic Code

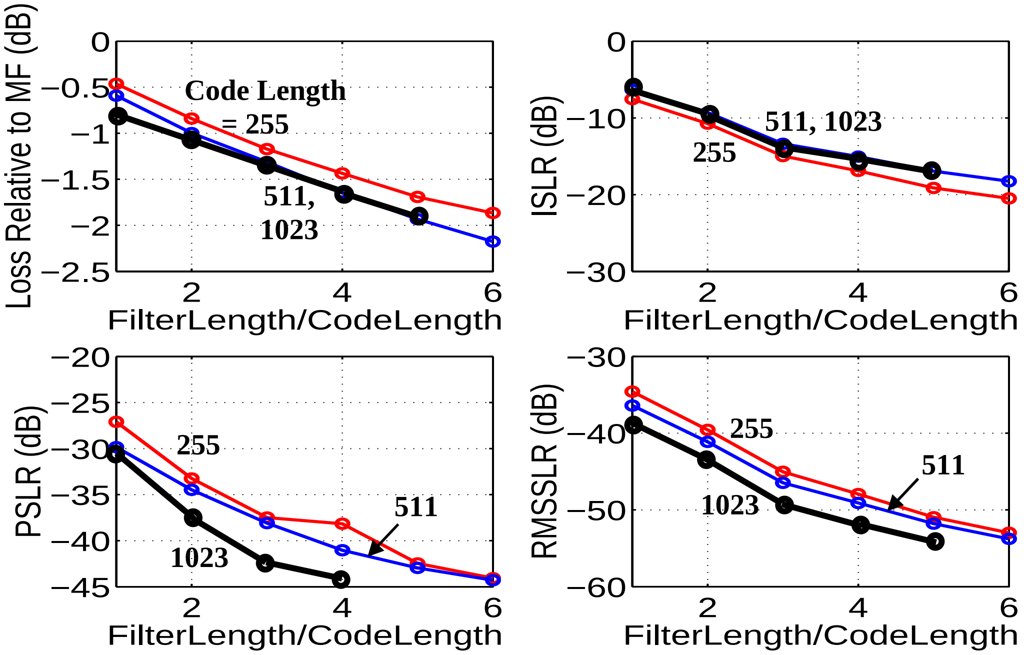


Figure 7. Performance Measures for Cyclic Codes Designed to Minimize PSLR vs Ratio of Length of Mismatched Filter to Code Length (M/S)

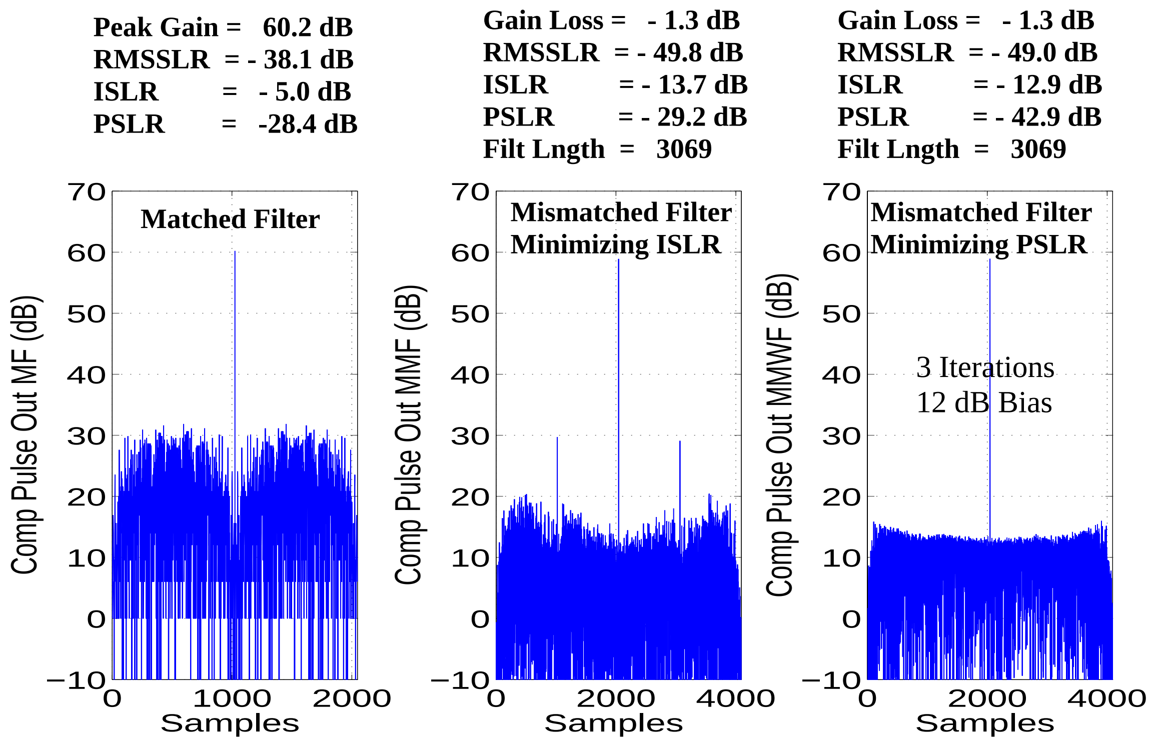


Figure 8. Comparison of Compressed Pulse Patterns for a 1023 Bit Cyclic Code

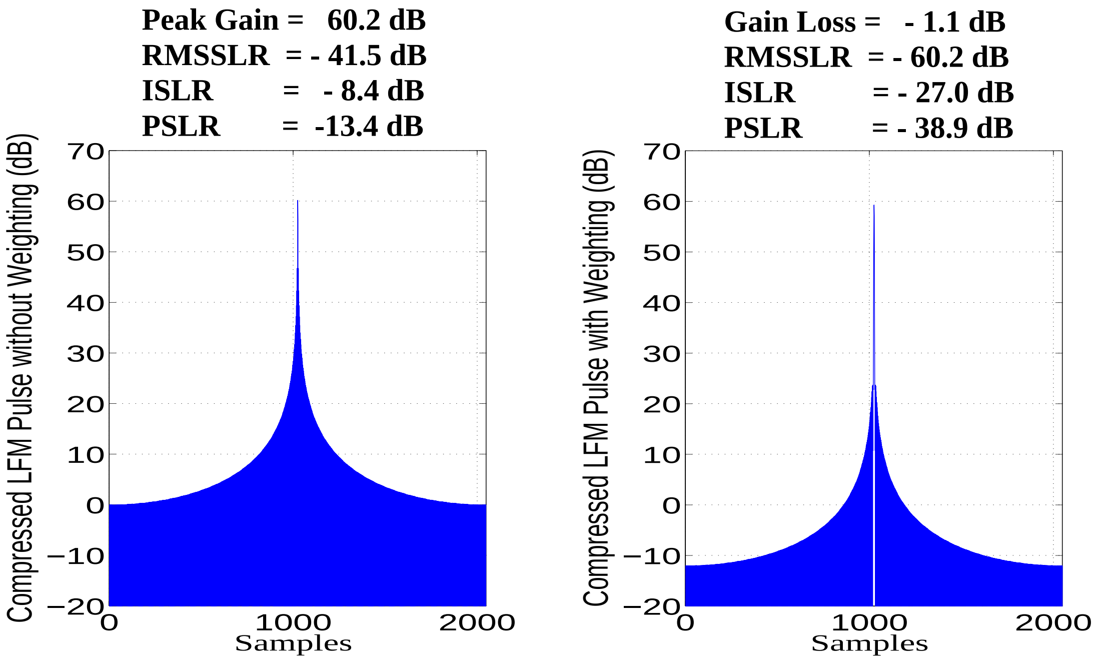


Figure 9. Compressed LFM Pulse with and without 40 dB, nbar = 4, Taylor Weighting: (Pulse Width) x Bandwidth = 1024

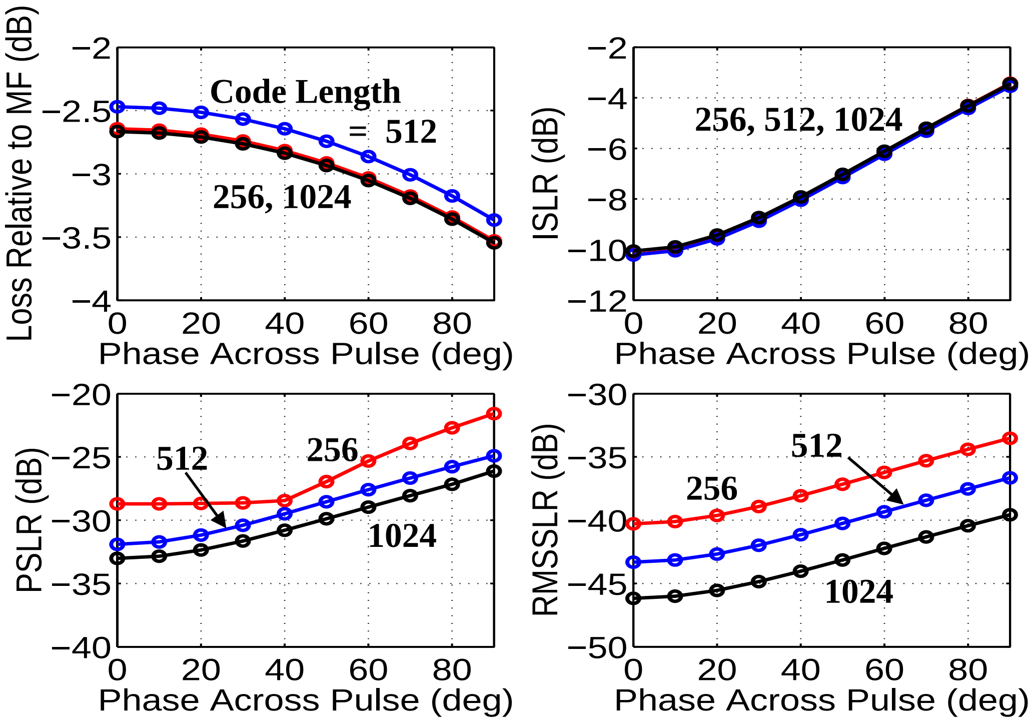


Figure 10. Performance Measures vs Total Doppler Induced Phase Shift Across Uncompressed Pulse: Random Binary Codes with Diagonal Loading = Trace/(10 M); Designed to Minimize ISLR; M=3S

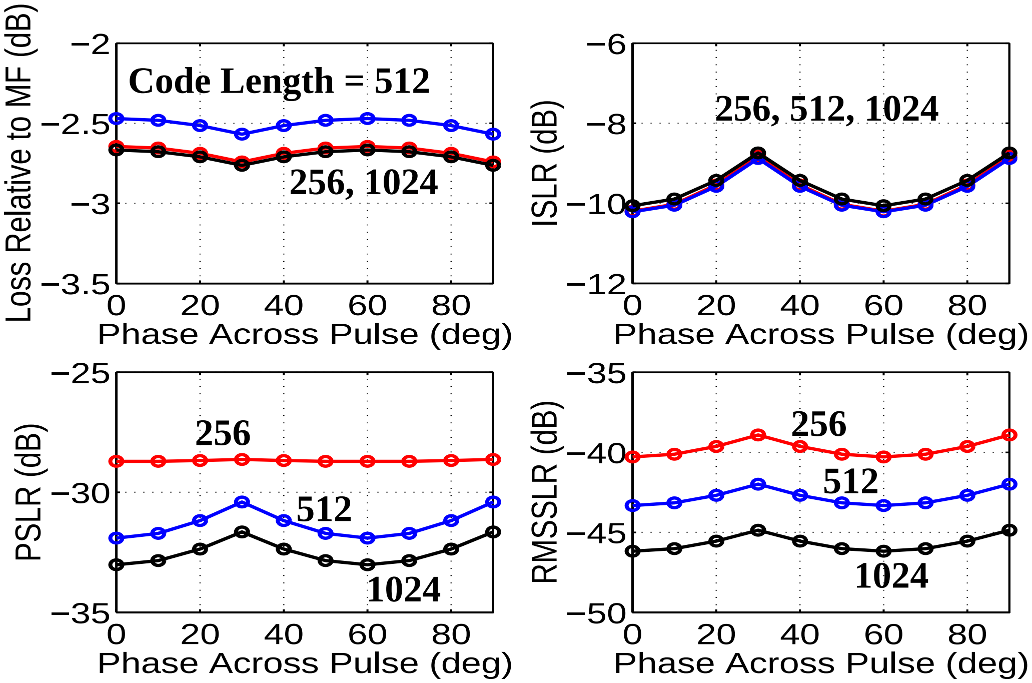


Figure 11. Figure 9 with 3 Hypotheses

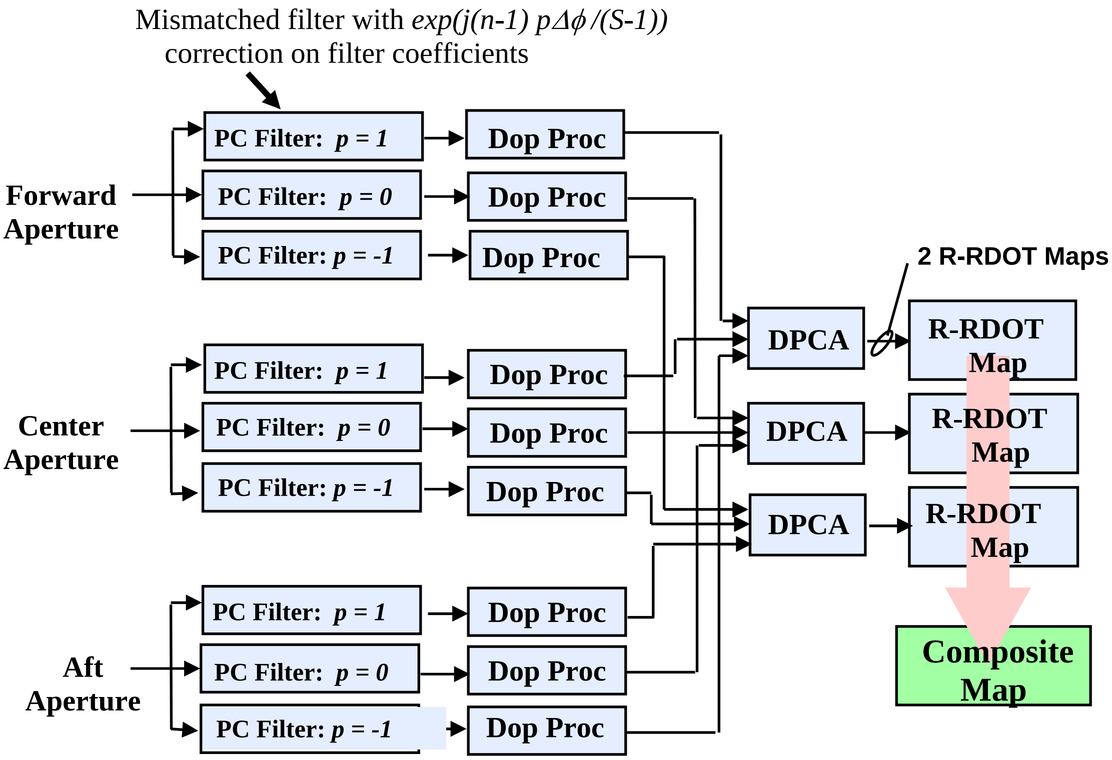
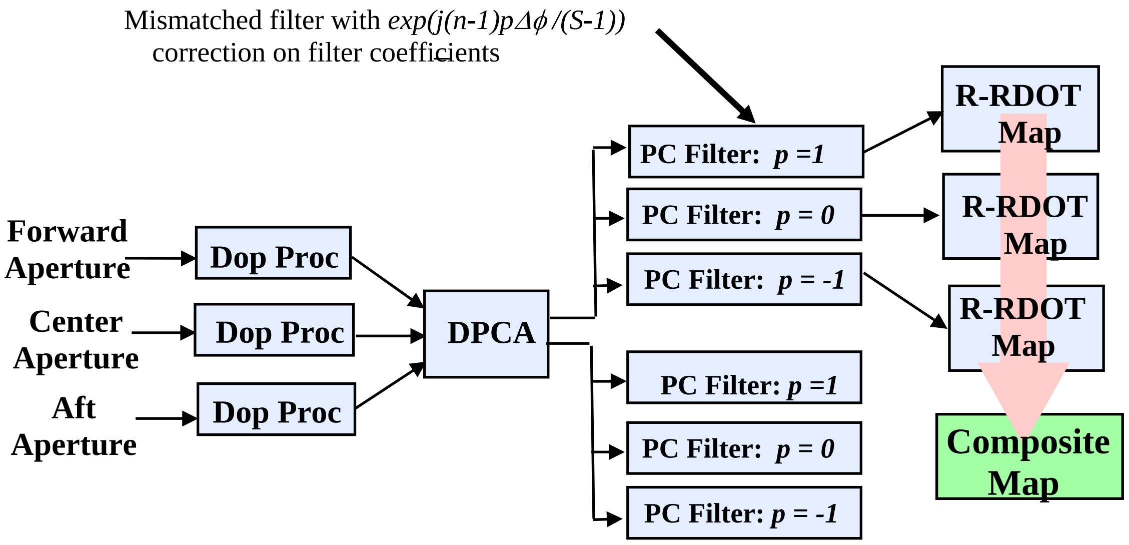


Figure 12a. Traditional Processing Using Three Multiple Hypotheses to Mitigate Doppler Intolerance: Pulse Compression Precedes Doppler Processing



Mismatched filter with $exp(j(n-1)p\Delta\phi/(S-1))$ correction on filter coefficients

Figure 12b. Reverse Processing Using Three Multiple Hypotheses to Mitigate Doppler Intolerance: Doppler Processing Precedes Pulse Compression

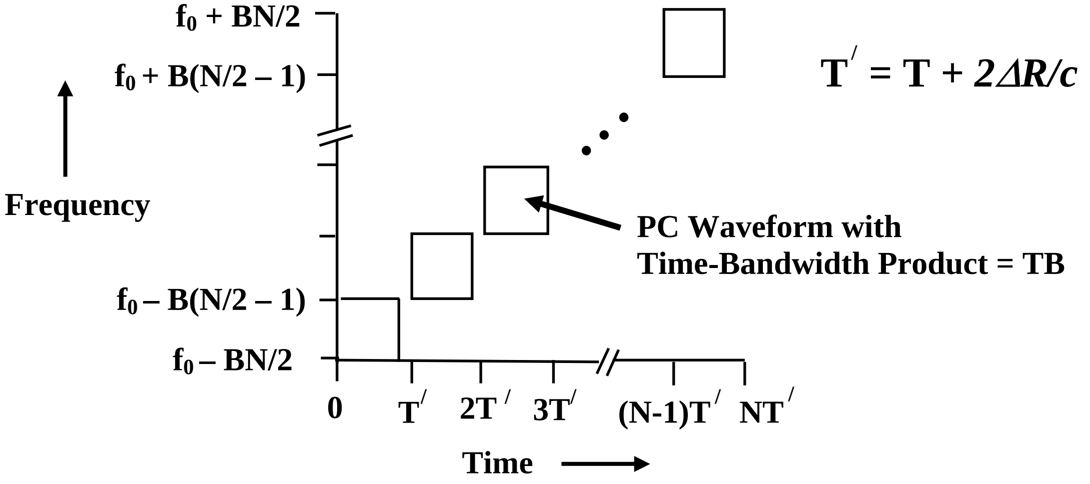
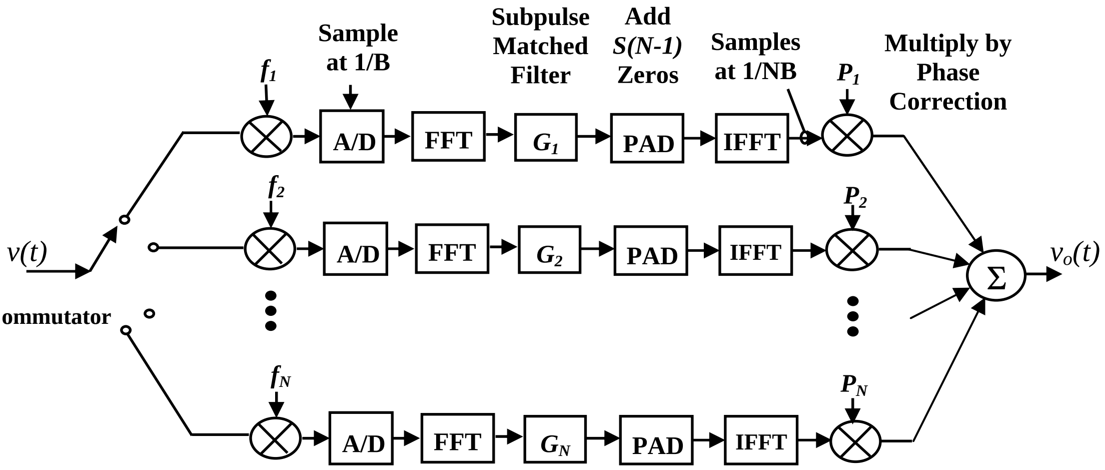


Figure 13. Phase Coded Frequency Stepped Waveform



$$f_n = f_o - NB/2 + nB$$
$$G_n(f) = FFT\,[h_{opt}(t)]\,exp(j2\pi f\,T)$$
$$P_n = exp(-j2\pi nBt)$$

Figure 14. Block Diagram of PCFSW Processing on Receive

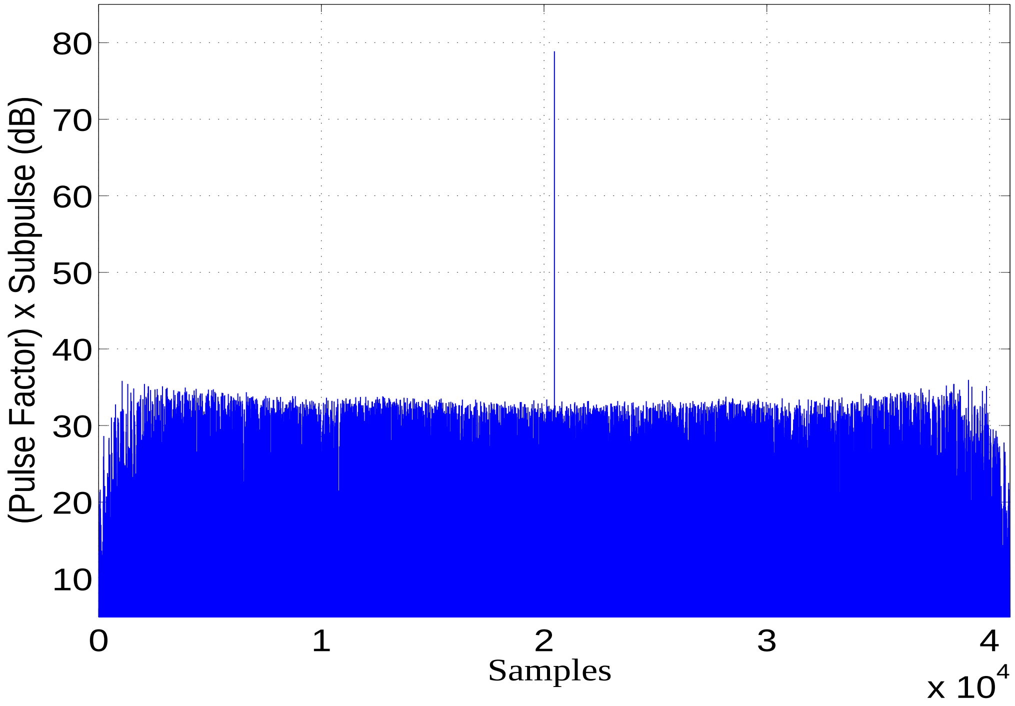


Figure 15. Composite PCFSW Equal to *Pulse Factor* x Subpulse: Subpulse is 1023 Chip Cyclic Code and Mismatched Filter Length is 3069 (see Figure 8)