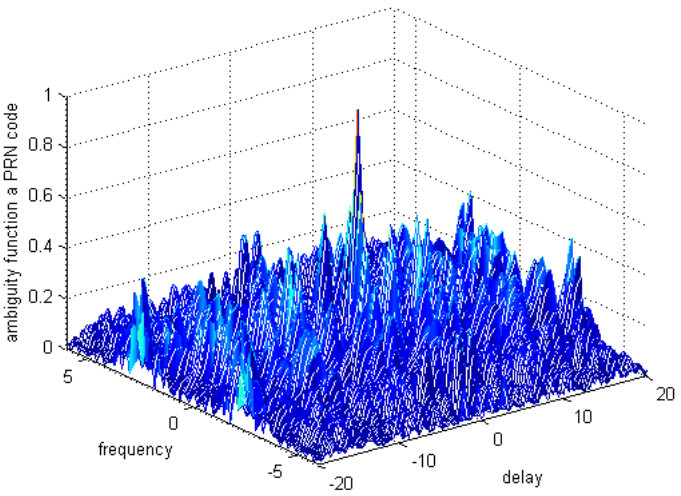


Figure 6.10a. Ambiguity function for B_{54} Barker code.

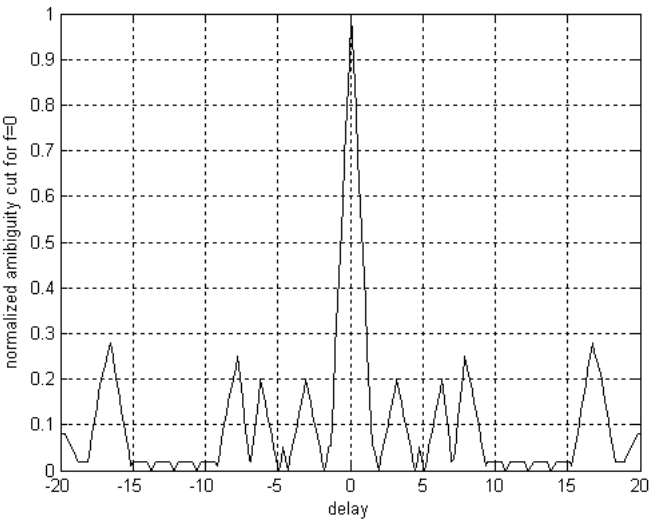


Figure 6.10b. Zero Doppler cut for the B_{54} ambiguity function.

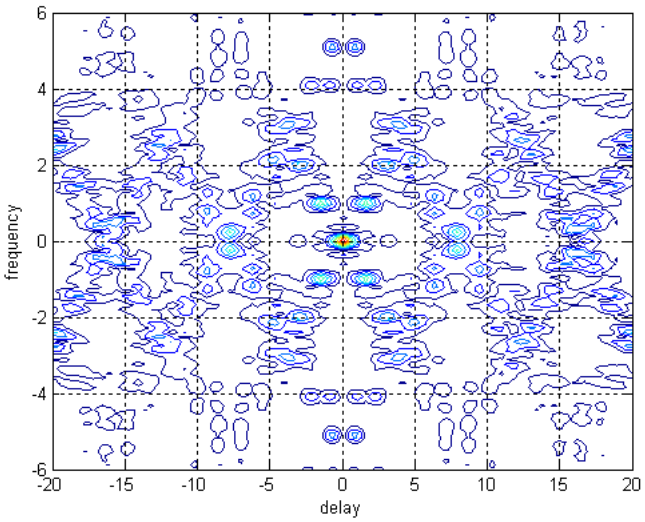


Figure 6.10c. Contour plot corresponding to Fig. 6.10a.

Pseudo-Random Number (PRN) Codes

Pseudo-Random Number (PRN) codes are also known as Maximal Length Sequences (MLS) codes. These codes are called pseudo-random because the statistics associated with their occurrence are similar to those associated with the coin-toss sequences. Maximum length sequences are periodic. The MLS codes have the following distinctive properties:

1. The number of ones per period is one more than the number of minus ones.
2. Half the runs (consecutive states of the same kind) are of length one and one fourth are of length two.
3. Every maximal length sequence has the "shift and add" property. This means that, if a maximal length sequence is added (modulo 2) to a shifted version of itself, then the resulting sequence is a shifted version of the original sequence.
4. Every *n*-tuple of the code appears once and only once in one period of the sequence.
5. The correlation function is periodic and is given by

$$\phi(n) = \begin{cases} L & n = 0, \pm L, \pm 2L, \ldots \\ -1 & elsewhere \end{cases} \qquad \textbf{(6.34)}$$

Figure 6.11 shows a typical sketch for an MLS autocorrelation function. Clearly these codes have the advantage that the compression ratio becomes very large as the period is increased. Additionally, adjacent peaks (grating lobes) become farther apart.

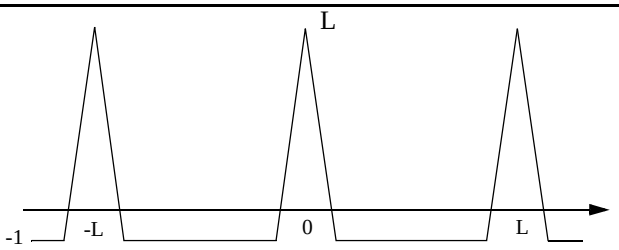


Figure 6.11. Typical autocorrelation of an MLS code of length L.

Linear Shift Register Generators

There are numerous ways to generate MLS codes. The most common is to use linear shift registers. When the binary sequence generated using a shift register implementation is periodic and has maximal length, it is referred to as an MLS binary sequence with period L, where

$$L = 2^n - 1 \qquad \textbf{(6.35)}$$

n is the number of stages in the shift register generator. A linear shift register generator basically consists of a shift register with modulo-two adders added to it. The adders can be connected to various stages of the register, as illustrated in Fig. 6.12 for $n = 4$ (i.e., $L = 15$). Note that the shift register initial state cannot be 0.

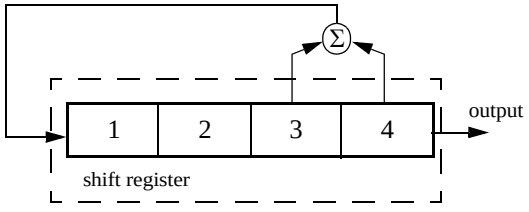


Figure 6.12. Circuit for generating an MLS sequence of length $L = 15$.

The feedback connections associated with a shift register generator determine whether the output sequence will be maximal. For a given size shift register, only a few feedback connections lead to maximal sequence outputs. In order to illustrate this concept, consider the two 5-stage shift register generators shown in Fig. 6.13. The shift register generator shown in Fig. 6.13 a generates a maximal length sequence, as clearly depicted by its state diagram. However, the shift register generator shown in Fig. 6.13 b produces three non-maximal length sequences (depending on the initial state).

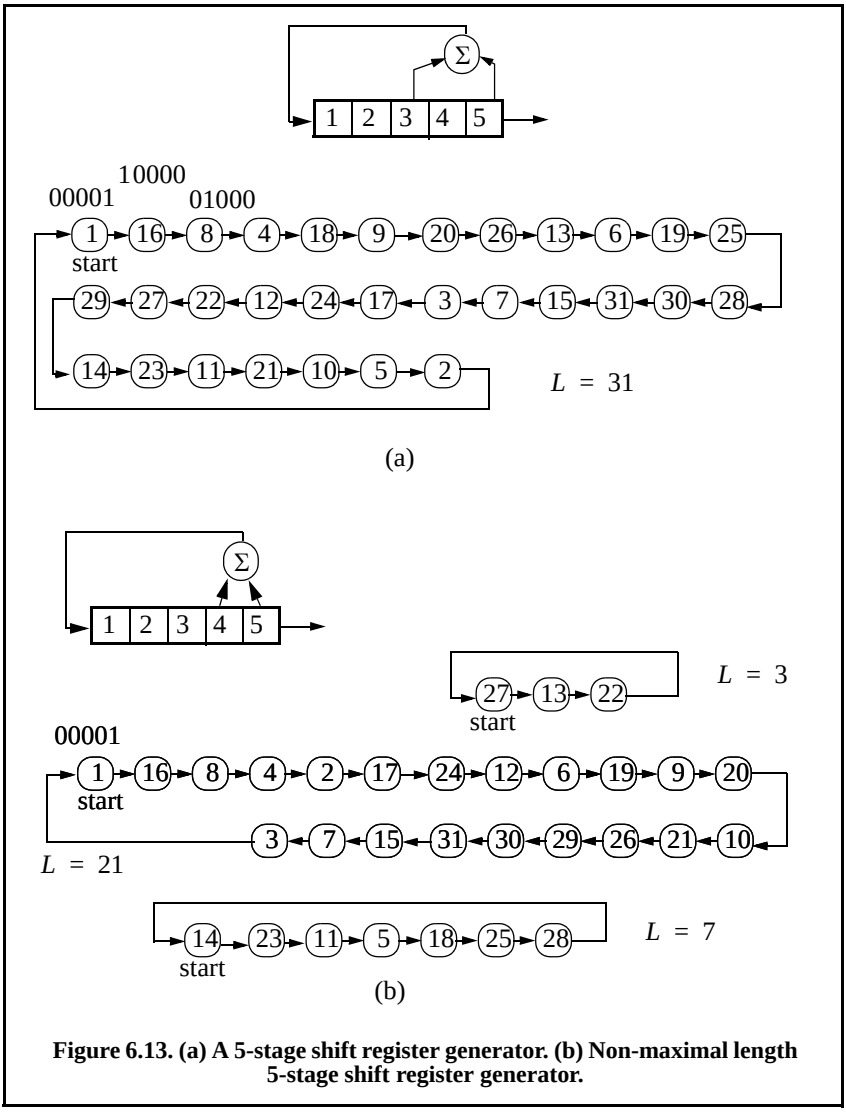


Figure 6.13. (a) A 5-stage shift register generator. (b) Non-maximal length 5-stage shift register generator.

Given an n-stage shift register generator, one would be interested in knowing how many feedback connections will yield maximal length sequences. Zierler[1] showed that the number of maximal length sequences possible for a given n-stage linear shift register generator is given by

$$N_L = \frac{\varphi(2^n - 1)}{n} \tag{6.36}$$

φ is the Euler's totient (Euler's phi) function and is defined by

$$\varphi(k) = k \prod_i \frac{(p_i - 1)}{p_i} \tag{6.37}$$

where p_i are the prime factors of k. Note that when p_i has multiples, only one of them is used. Also note that when k is a prime number, the Euler's phi function is

$$\varphi(k) = k - 1 \tag{6.38}$$

For example, a 3-stage shift register generator will produce

$$N_L = \frac{\varphi(2^3 - 1)}{3} = \frac{\varphi(7)}{3} = \frac{7 - 1}{3} = 2 \tag{6.39}$$

and a 6-stage shift register,

$$N_L = \frac{\varphi(2^6 - 1)}{6} = \frac{\varphi(63)}{6} = \frac{63}{6} \times \frac{(3 - 1)}{3} \times \frac{(7 - 1)}{7} = 6 \tag{6.40}$$

Maximal Length Sequence Characteristic Polynomial

Consider an n-stage maximal length linear shift register whose feedback connections correspond to n, k, m, etc. This maximal length shift register can be described using its characteristic polynomial defined by

$$x^n + x^k + x^m + \ldots + 1 \tag{6.41}$$

where the additions are modulo 2. Therefore, if the characteristic polynomial for an n-stage shift register is known, one can easily determine the register feedback connections and consequently deduce the corresponding maximal length sequence. For example, consider a 6-stage shift register whose characteristic polynomial is

$$x^6 + x^5 + 1 \tag{6.42}$$

1. Zierler, N., *Several Binary-Sequence Generators*, MIT Technical Report No. 95, Sept. 1955.

It follows that the shift register which generates a maximal length sequence is shown in Fig. 6.14.

One of the most important issues associated with generating a maximal length sequence using a linear shift register is determining the characteristic polynomial. This has been and continues to be a subject of research for many radar engineers and designers. It has been shown that polynomials which are both irreducible (not factorable) and primitive will produce maximal length shift register generators.

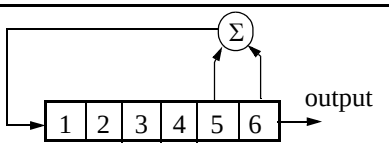


Figure 6.14. Linear shift register whose characteristic polynomial is $x^6 + x^5 + 1$.

A polynomial of degree n is irreducible if it is not divisible by any polynomial of degree less than *n*. It follows that all irreducible polynomials must have an odd number of terms. Consequently, only linear shift register generators with an even number of feedback connections can produce maximal length sequences. An irreducible polynomial is primitive if and only if it divides $x^n - 1$ for no value of n less than $2^n - 1$.

The MATLAB function "*prn_ambig.m*" calculates and plots the ambiguity function associated with a given PRN code. Figure 6.15 shows the output of this function for

u31 = [1 -1 -1 -1 -1 1 -1 1 -1 1 1 1 -1 1 1 -1 -1 -1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 -1]

Figure 6.16 is similar to Fig. 6.15, except in this case the input maximal length sequence is

u15=[1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 -1 -1]

6.3.2. Polyphase Codes

The signal corresponding to polyphase codes is as that given in Eq. (6.22) and the corresponding ambiguity function was given in Eq. (6.24). The only exception being that the phase θ_n is no longer restricted to $(0, \pi)$. Hence, the coefficient D_n are no longer equal to ± 1 but can be complex depending on the value of θ_n. Polyphase Barker codes have been investigated by many scientists and much is well documented in the literature. In this chapter the discussion will be limited to Frank codes.

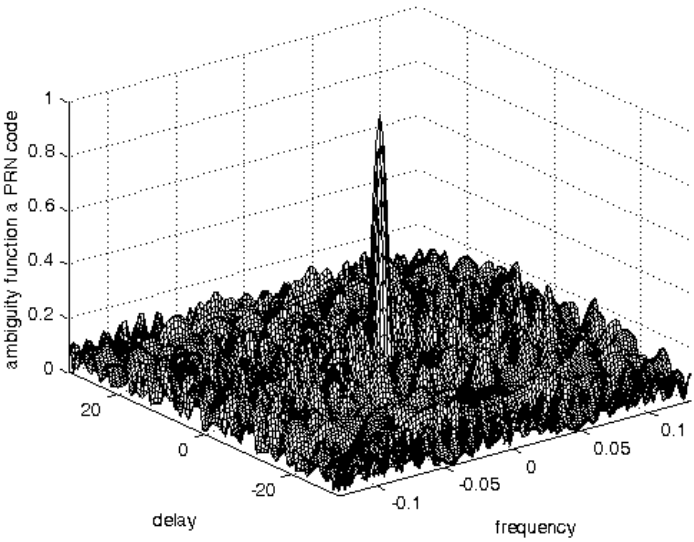


Figure 6.15a. Ambiguity function corresponding to a 31-bit PRN code.

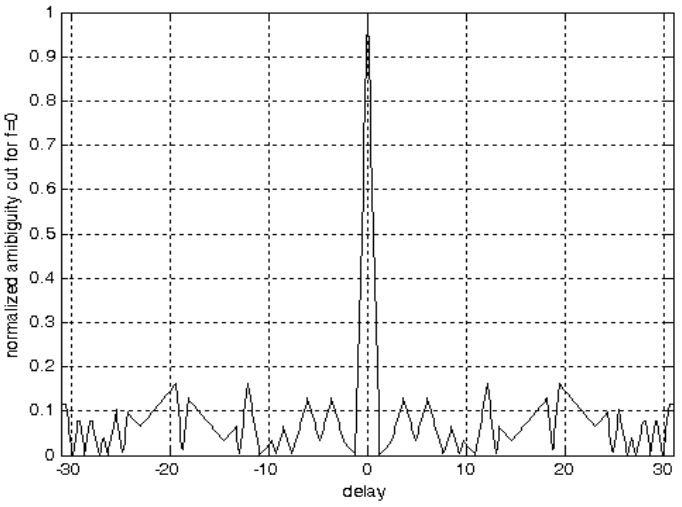


Figure 6.15b. Zero Doppler cut corresponding to Fig. 6.15a.

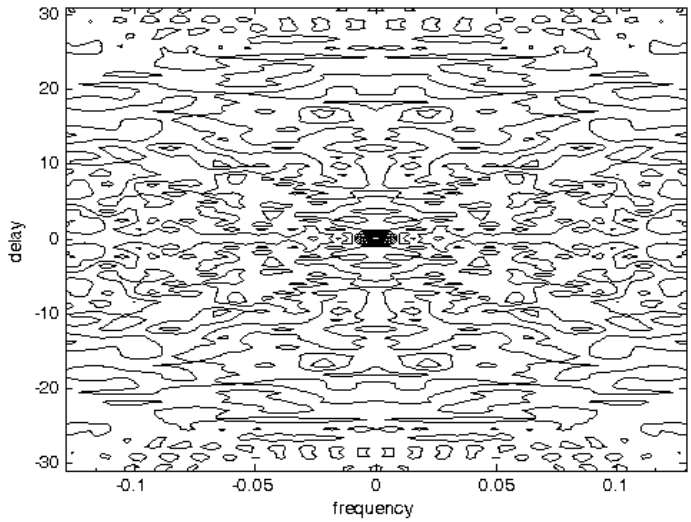


Figure 6.15c. Contour plot corresponding to Fig. 6.15a.

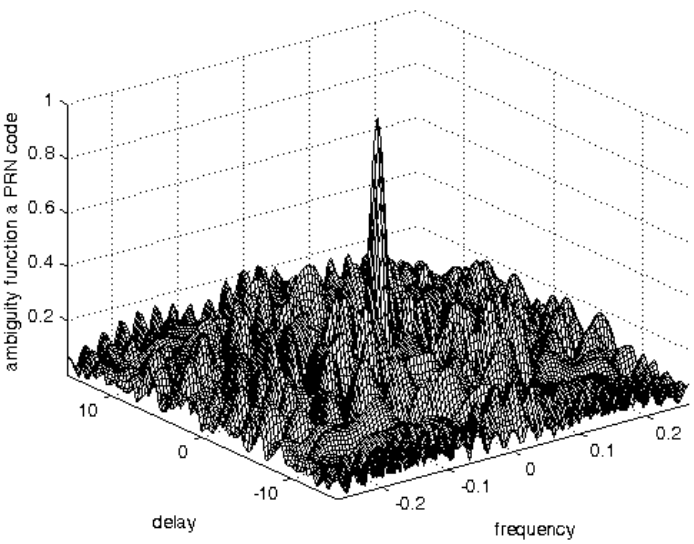


Figure 6.16a. Ambiguity function corresponding to a 15-bit PRN code.

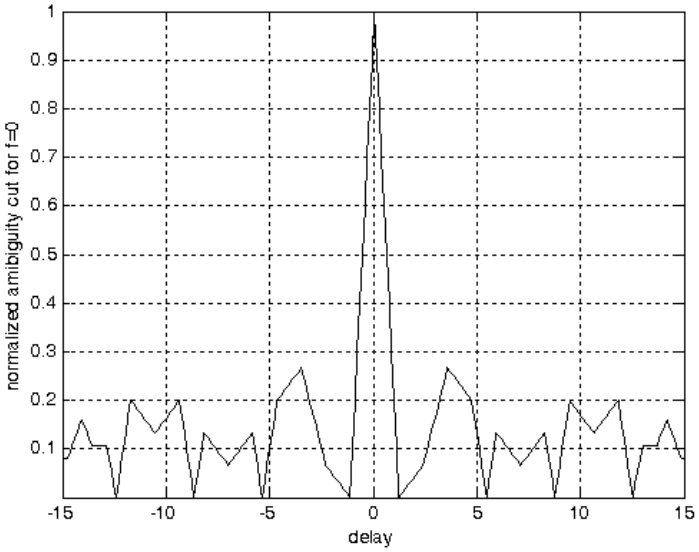


Figure 6.16b. Zero Doppler cut corresponding to Fig. 6.16a.

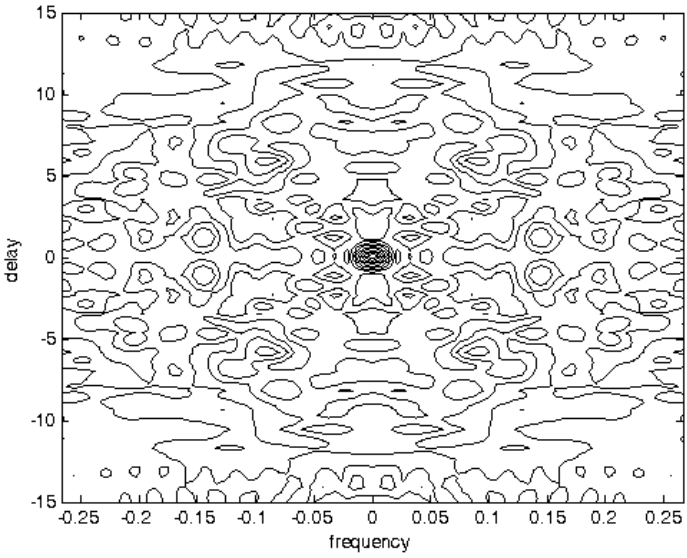


Figure 6.16c. Contour plot corresponding to Fig. 6.16a.

Frank codes

In this case, a single pulse of width T_p is divided into N equal groups; each group is subsequently divided into other N subpulses each of width τ_0. Therefore, the total number of subpulses within each pulse is N^2, and the compression ratio is $\xi = N^2$. As previously, the phase within each subpulse is held constant with respect to some CW reference signal.

A Frank code of N^2 subpulses is referred to as an N-phase Frank code. The first step in computing a Frank code is to divide $360°$ by N and define the result as the fundamental phase increment $\Delta\varphi$. More precisely,

$$\Delta\varphi = 360°/N \tag{6.43}$$

Note that the size of the fundamental phase increment decreases as the number of groups is increased, and because of phase stability, this may degrade the performance of very long Frank codes. For N-phase Frank code the phase of each subpulse is computed from

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 2 & 3 & \dots & N-1 \\ 0 & 2 & 4 & 6 & \dots & 2(N-1) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & (N-1) & 2(N-1) & 3(N-1) & \dots & (N-1)^2 \end{pmatrix} \Delta\varphi \tag{6.44}$$

where each row represents a group, and a column represents the subpulses for that group. For example, a 4-phase Frank code has $N = 4$, and the fundamental phase increment is $\Delta\varphi = (360°/4) = 90°$. It follows that

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 90° & 180° & 270° \\ 0 & 180° & 0 & 180° \\ 0 & 270° & 180° & 90° \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \tag{6.45}$$

Therefore, a Frank code of 16 elements is given by

$$F_{16} = \{1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; j \;\; -1 \;\; -j \;\; 1 \;\; -1 \;\; 1 \;\; -1 \;\; 1 \;\; -j \;\; -1 \;\; j\} \tag{6.46}$$

A plot of the ambiguity function for F_{16} is shown in Fig. 6.17. Note the thumb-tack shape of the ambiguity function. This plot can be reproduced using the following MATLAB code. The phase increments within each row represent a step-wise approximation of an up-chirp LFM waveform. The phase increments for subsequent rows increase linearly versus time. Thus, the correspond-

ing LFM chirp slopes also increase linearly for subsequent rows. This is illustrated in Fig. 6.18, for F_{16}.

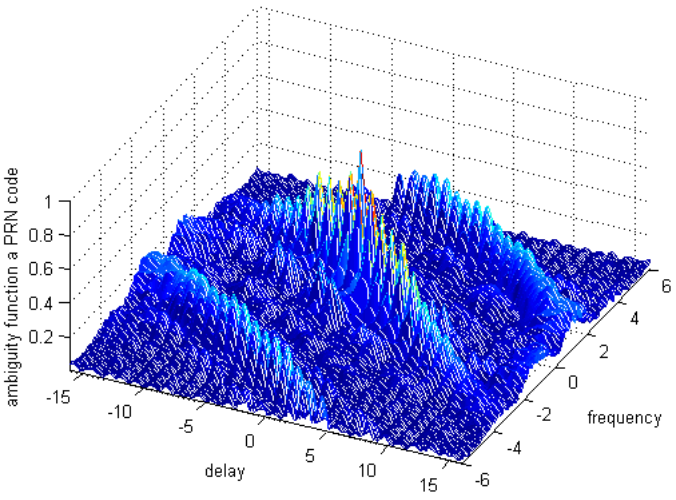


Figure 6.17a. Ambiguity plot for Frank code F_{16}.

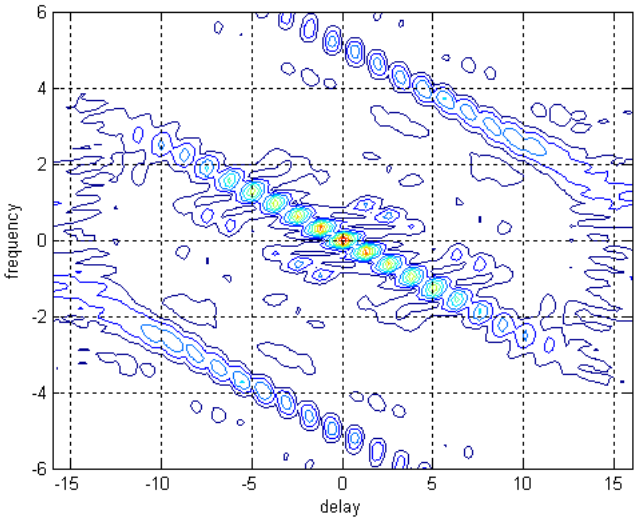


Figure 6.17b. Contour plot corresponding to Fig. 6.17a.

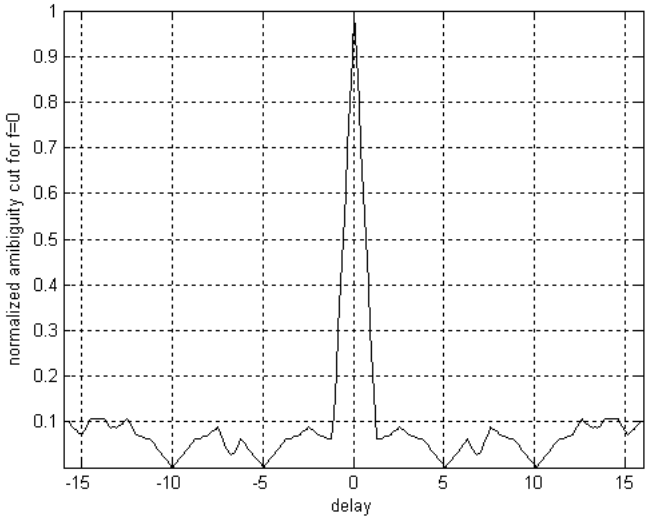


Figure 6.17c. Zero Doppler cut corresponding to Fig. 6.17a.

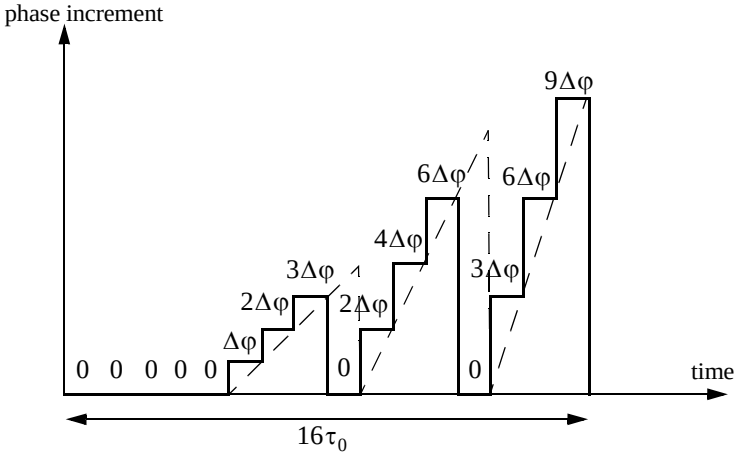


Figure 6.18. Step-wise approximation of an up-chirp waveform, using a Frank code of 16 elements.

6.4. Frequency Codes

Frequency codes are derived from Eq. (6.1) under the condition stated in Eq. (6.9) (i.e., $\theta_n = 0 \quad ;and \;\; a_n = 1, or \;\; 0$). The Stepped Frequency Waveform (SFW) discussed in the previous chapter is considered to be a code under this class of discrete coded waveforms. The ambiguity function was derived in Chapter 5 for SFW. In this chapter the focus is on another type of frequency codes that is called the Costas frequency code.

6.4.1. Costas Codes

Construction of Costas codes can be understood in the context of SFW. In SFW, a relatively long pulse of length T_p is divided into N subpulses, each of width τ_0 ($T_p = N\tau_0$). Each group of N subpulses is called a burst. Within each burst the frequency is increased by Δf from one subpulse to the next. The overall burst bandwidth is $N\Delta f$. More precisely,

$$\tau_0 = T_p / N \tag{6.47}$$

and the frequency for the ith subpulse is

$$f_i = f_0 + i\Delta f; \;\; i = 1, N \tag{6.48}$$

where f_0 is a constant frequency and $f_0 \gg \Delta f$. It follows that the time-bandwidth product of this waveform is

$$\Delta f T_p = N^2 \tag{6.49}$$

Costas[1] signals (or codes) are similar to SFW, except that the frequencies for the subpulses are selected in a random fashion, according to some predetermined rule or logic. For this purpose, consider the $N \times N$ matrix shown in Fig. 6.19 b. In this case, the rows are indexed from $i = 1, 2, \ldots, N$ and the columns are indexed from $j = 0, 1, 2, \ldots, (N-1)$. The rows are used to denote the subpulses and the columns are used to denote the frequency. A *dot* indicates the frequency value assigned to the associated subpulse. In this fashion, Fig. 6.19 a shows the frequency assignment associated with an SFW. Alternatively, the frequency assignments in Fig. 6.19b are chosen randomly. For a matrix of size $N \times N$, there are a total of $N!$ possible ways of assigning the dots (i.e., $N!$ possible codes).

The sequences of dot assignments for which the corresponding ambiguity function approaches an ideal or a *thumb-tack* response are called Costas codes.

1. Costas, J. P., A Study of a Class of Detection Waveforms Having Nearly Ideal Range-Doppler Ambiguity Properties, *Proc. IEEE* 72, 1984, pp. 996-1009.

A near thumb-tack response was obtained by Costas using the following logic: There is only one frequency per time slot (row) and per frequency slot (column). Therefore, for an $N \times N$ matrix the number of possible Costas codes is drastically less than $N!$. For example, there are $N_c = 4$ possible Costas codes for $N = 3$, and $N_c = 40$ possible codes for $N = 5$. It can be shown that the code density, defined as the ratio $N_c/N!$, gets significantly smaller as N becomes larger

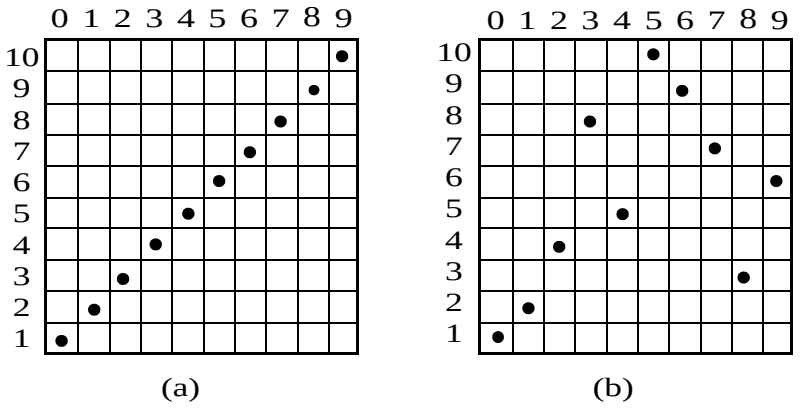


Figure 6.19. Frequency assignment for a burst of N subpulses. (a) SFW (stepped LFM); (b) Costas code of length Nc = 10.

There are numerous analytical ways to generate Costas codes. In this section we will describe two of these methods. First, let q be an odd prime number, and choose the number of subpulses as

$$N = q - 1 \tag{6.50}$$

Define γ as the primitive root of q. A primitive root of q (an odd prime number) is defined as γ such that the powers $\gamma, \gamma^2, \gamma^3, \ldots, \gamma^{q-1}$ modulo q generate every integer from 1 to $q - 1$.

In the first method, for an $N \times N$ matrix, label the rows and columns, respectively, as

$$\begin{aligned} i &= 0, 1, 2, \ldots, (q-2) \\ j &= 1, 2, 3, \ldots, (q-1) \end{aligned} \tag{6.51}$$

Place a dot in the location (i, j) corresponding to f_i if and only if

$$i = (\gamma)^j \ (modulo \ q) \tag{6.52}$$

In the next method, Costas code is first obtained from the logic described above; then by deleting the first row and first column from the matrix a new code is generated. This method produces a Costas code of length $N = q - 2$.

Define the normalized complex envelope of the Costas signal as

$$x(t) = \frac{1}{\sqrt{N\tau_0}} \sum_{l=0}^{N-1} x_l(t - l\tau_0) \tag{6.53}$$

$$x_l(t) = \begin{pmatrix} \exp(j2\pi f_l t) & 0 \le t \le \tau_0 \\ 0 & elsewhere \end{pmatrix} \tag{6.54}$$

Costas showed that the output of the matched filter is

$$\chi(\tau, f_d) = \frac{1}{N} \sum_{l=0}^{N-1} \exp(j2\pi l f_d \tau) \left\{ \Phi_{ll}(\tau, f_d) + \sum_{\substack{q=0 \\ q \ne l}}^{N-1} \Phi_{lq}(\tau - (l-q)\tau_0, f_d) \right\} \tag{6.55}$$

$$\Phi_{lq}(\tau, f_d) = \left(\tau_0 - \frac{|\tau|}{\tau_0}\right) \frac{\sin\alpha}{\alpha} \exp(-j\beta - j2\pi f_q \tau) \quad , \ |\tau| \le \tau_1 \tag{6.56}$$

$$\alpha = \pi(f_l - f_q - f_d)(\tau_0 - |\tau|) \tag{6.57}$$

$$\beta = \pi(f_l - f_q - f_d)(\tau_0 + |\tau|) \tag{6.58}$$

Three-dimensional plots of the ambiguity function of Costas signals show the near thumb-tack response of the ambiguity function. All side-lobes, except for a few around the origin, have amplitude $1/N$. Few sidelobes close to the origin have amplitude $2/N$, which is typical of Costas codes. The compression ratio of a Costas code is approximately N.

6.5. Ambiguity Plots for Discrete Coded Waveforms

Plots of the ambiguity function for a given code and the corresponding cuts along zero delay and zero Doppler provide strong indication about the code's characteristics in range and Doppler. Earlier, it was stated that the *goodness* of a given code is measured by its range and Doppler resolution characteristics. Therefore, plotting the ambiguity function of a given code is a key part of the design and analysis of radar waveforms. Unfortunately, some of the formulas for the ambiguity function are rather complicated and fairly difficult to code by the nonexpert programmer. In this section, a numerical technique for plotting

the ambiguity function of any code is presented. This technique takes advantage of the computation power of MATLAB by exploiting one of the properties of the ambiguity function. Three-dimensional plots are built successively from cuts of the ambiguity function as different Doppler mismatches.

For this purpose, consider the ambiguity function property given in Eq. (5.8) and repeated here as Eq. (6.59)

$$\left|\chi(\tau;f_d)\right|^2 = \left|\int X^*(f)X(f-f_d)e^{-j2\pi f\tau}df\right|^2 \qquad \textbf{(6.59)}$$

where $X(f)$ is the Fourier transform of the signal $x(t)$. Using Eq. (6.59), one can compute the ambiguity function by first computing the FT of the signal under consideration, delaying it by some value f_d, and then taking the inverse FT. When the signal under consideration is a discrete coded waveform then the Fast Fourier transform is utilized. From this one can compute plots of the ambiguity function using the following technique:

1. Determine the code U under consideration. Note that U may have complex values in accordance with the class of code being considered.
2. Extend the length of the code to the next power of 2 by zero padding (see Chapter 2 for details on interpolation).
3. For better display utilize an FFT whose size is 8 times or higher than the power integer of 2 computed in step 2.
4. Compute the FFT of the extended sequence.
5. Generate vectors of frequency mismatches and delay cuts.
6. Calculate using vector notation the value of $X(f-f_d)$.
7. Compute and store the vector resulting from the point by point multiplication $X^*(f)X(f-f_d)$.
8. Compute the inverse FFT of the product in step 7 for each delay value and store in a two-dimensional (2-D) array.
9. Plot the amplitude square of the resulting 2-D array to generate the ambiguity plot for the specific code under consideration.

An implementation of this algorithm using MATLAB was completed; this program is called *"ambiguity_code.m."* The listing of this program is as follows:

```
function [ambig] = ambiguity_code(uinput)
% Compute and plot the ambiguity function for any give code u
% Compute the ambiguity function by utilizing the FFT
% through combining multiple range cuts
N = size(uinput,2);
tau = N;
code = uinput;
```

```
samp_num = size(code,2) * 10;
n = ceil(log(samp_num) / log(2));
nfft = 2^n;
u(1:nfft) = 0;
j = 0;
for index = 1:10:samp_num
   index;
   j = j+1;
   u(index:index+10-1) = code(j);
end
% set-up the array v
v = u;
delay = linspace(0,5*tau,nfft);
freq_del = 12 / tau /100;
j = 0;
vfft = fft(v,nfft);
for freq = -6/tau:freq_del:6/tau;
   j = j+1;
   exf = exp(sqrt(-1) * 2. * pi * freq .* delay);
   u_times_exf = u .* exf;
   ufft = fft(u_times_exf,nfft);
   prod = ufft .* conj(vfft);
   ambig(j,:) = fftshift(abs(ifft(prod))');
end
freq = linspace(-6,6, size(ambig,1));
delay = linspace(-N,N,nfft);
figure(1)
mesh(delay,freq,(ambig ./ max(max(ambig))))
% colormap([.5 .5 .5])
% colormap(gray)
axis tight
ylabel('frequency')
xlabel('delay')
zlabel('ambiguity function a PRN code')
figure(2)
plot(delay,ambig(51,:)/(max(max(ambig))),'k')
xlabel('delay')
ylabel('normalized amibiguity cut for f=0')
grid
axis tight
figure(3)
contour(delay,freq,(ambig ./ max(max(ambig))))
axis tight
% colormap([.5 .5 .5])
% colormap(gray)
ylabel('frequency')
xlabel('delay')
grid
```

Problems

6.1. Define $\{x_I(n) = 1, -1, 1\}$ and $\{x_Q(n) = 1, 1, -1\}$. (a) Compute the discrete correlations: R_{x_I}, R_{x_Q}, $R_{x_I x_Q}$, and $R_{x_Q x_I}$. (b) A certain radar transmits the signal $s(t) = x_I(t)\cos 2\pi f_0 t - x_Q(t)\sin 2\pi f_0 t$. Assume that the autocorrelation $s(t)$ is equal to $y(t) = y_I(t)\cos 2\pi f_0 t - y_Q(t)\sin 2\pi f_0 t$. Compute and sketch $y_I(t)$ and $y_Q(t)$.

6.2. Consider the 7-bit Barker code, designated by the sequence $x(n)$. (a) Compute and plot the autocorrelation of this code. (b) A radar uses binary phase coded pulses of the form $s(t) = r(t)\cos(2\pi f_0 t)$, where $r(t) = x(0),\ for\ 0 < t < \Delta t$, $r(t) = x(n),\ for\ n\Delta t < t < (n+1)\Delta t$, and $r(t) = 0,\ for\ t > 7\Delta t$. Assume $\Delta t = 0.5\mu s$. (a) Give an expression for the autocorrelation of the signal $s(t)$, and for the output of the matched filter when the input is $s(t - 10\Delta t)$; (b) compute the time bandwidth product, the increase in the peak SNR, and the compression ratio.

6.3. (a) Perform the discrete convolution between the sequence ϕ_{11} defined in Eq. (6.31), and the transversal filter impulse response; and (b) sketch the corresponding transversal filter output.

6.4. Repeat the previous problem for $N = 13$ and $k = 6$. Use Barker code of length 13.

6.5. Develop a Barker code of length 35. Consider both B_{75} and B_{57}.

6.6. The smallest positive primitive root of $q = 11$ is $\gamma = 2$; for $N = 10$ generate the corresponding Costas matrix.

6.7. Compute the discrete autocorrelation for an F_{16} Frank code.

6.8. Generate a Frank code of length 8, i.e., F_8.

6.9. Using the MATLAB program developed in this chapter, plot the matched filter output for a 3-, 4-, and 5-bits Barker code.

Chapter 7 Target Detection and Pulse Integration

7.1. Target Detection in the Presence of Noise

A simplified block diagram of a radar receiver that employs an envelope detector followed by a threshold decision is shown in Fig. 7.1. The input signal to the receiver is composed of the radar echo signal $s(t)$ and additive zero mean white Gaussian noise random process $n(t)$, with variance σ^2. The input noise is assumed to be spatially incoherent and uncorrelated with the signal.

The output of the bandpass intermediate frequency (IF) filter is the signal $v(t)$, which can be written as a bandpass random process. That is,

$$\begin{aligned} v(t) &= v_I(t)\cos\omega_0 t + v_Q(t)\sin\omega_0 t = r(t)\cos(\omega_0 t - \Phi(t)) \\ v_I(t) &= r(t)\cos\Phi(t) \\ v_Q(t) &= r(t)\sin\Phi(t) \end{aligned} \tag{7.1a}$$

$$\begin{aligned} r(t) &= \sqrt{[v_I(t)]^2 + [v_Q(t)]^2} \\ \Phi(t) &= \left[\tan\left(\frac{v_Q(t)}{v_I(t)}\right)\right]^{-1} \end{aligned} \tag{7.1b}$$

where $\omega_0 = 2\pi f_0$ is the radar operating frequency, $r(t)$ is the envelope of $v(t)$, the phase is $\Phi(t) = \text{atan}(v_Q / v_I)$, and the subscripts I, and Q, respectively, refer to the in-phase and quadrature components.

A target is detected when $r(t)$ exceeds the threshold value v_T, where the decision hypotheses are

$$\begin{aligned} s(t) + n(t) > v_T &\Rightarrow Detection \\ n(t) > v_T &\Rightarrow False\ \ alarm \end{aligned}$$

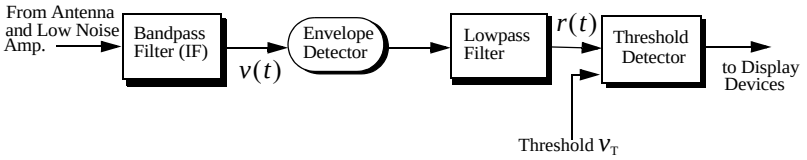


Figure 7.1. Simplified block diagram of an envelope detector and threshold receiver.

The case when the noise subtracts from the signal (while a target is present) to make $r(t)$ smaller than the threshold is called a miss. Radar designers seek to maximize the probability of detection for a given probability of false alarm.

The IF filter output is a complex random variable that is composed of either noise alone or noise plus target return signal (sine wave of amplitude A). The quadrature components corresponding to the case of noise alone are

$$\begin{aligned} v_I(t) &= n_I(t) \\ v_Q(t) &= n_Q(t) \end{aligned} \tag{7.2}$$

and for the second case,

$$\begin{aligned} v_I(t) = A + n_I(t) = r(t)\cos\Phi(t) \Rightarrow n_I(t) &= r(t)\cos\Phi(t) - A \\ v_Q(t) = n_Q(t) &= r(t)\sin\Phi(t) \end{aligned} \tag{7.3}$$

where the noise quadrature components $n_I(t)$ and $n_Q(t)$ are uncorrelated zero mean lowpass Gaussian noise with equal variances, σ^2. The joint Probability Density Function (*pdf)* of the two random variables $n_I;n_Q$ is

$$\begin{aligned} f_{n_I n_Q}(n_I, n_Q) &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{n_I^2 + n_Q^2}{2\sigma^2}\right) \\ &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(r\cos\varphi - A)^2 + (r\sin\varphi)^2}{2\sigma^2}\right) \end{aligned} \tag{7.4}$$

The *pdfs* of the random variables $r(t)$ and $\Phi(t)$, respectively, represent the modulus and phase of $v(t)$. The joint *pdf* for the two random variables $r(t);\Phi(t)$ are derived using a similar approach to that developed in Chapter 3. More precisely,

$$f_{R\Phi}(r, \varphi) = f_{n_I n_Q}(n_I, n_Q)|\boldsymbol{J}| \tag{7.5}$$

where $\boldsymbol{J}$ is a matrix of derivatives defined by

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial n_I}{\partial r} & \dfrac{\partial n_I}{\partial \varphi} \\ \dfrac{\partial n_Q}{\partial r} & \dfrac{\partial n_Q}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -r\sin\varphi \\ \sin\varphi & r\cos\varphi \end{bmatrix} \tag{7.6}$$

The determinant of the matrix of derivatives is called the Jacobian, and in this case it is equal to

$$|\boldsymbol{J}| = r(t) \tag{7.7}$$

Substituting Eq. (7.4) and Eq. (7.7) into Eq. (7.5) and collecting terms yield

$$f_{R\Phi}(r, \varphi) = \frac{r}{2\pi\sigma^2}\exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right)\exp\left(\frac{rA\cos\varphi}{\sigma^2}\right) \tag{7.8}$$

The *pdf* for $r(t)$ alone is obtained by integrating Eq. (7.8) over φ

$$f_R(r) = \int_0^{2\pi} f_{R\Phi}(r, \varphi)d\varphi = \frac{r}{\sigma^2}\exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right)\frac{1}{2\pi}\int_0^{2\pi}\exp\left(\frac{rA\cos\varphi}{\sigma^2}\right)d\varphi \tag{7.9}$$

where the integral inside Eq. (7.9) is known as the modified Bessel function of zero order,

$$I_0(\beta) = \frac{1}{2\pi}\int_0^{2\pi} e^{\beta\cos\theta}\, d\theta \tag{7.10}$$

Thus,

$$f_R(r) = \frac{r}{\sigma^2}I_0\left(\frac{rA}{\sigma^2}\right)\exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) \tag{7.11}$$

which is the Rician probability density function. The case when $A/\sigma^2 = 0$ (noise alone) was analyzed in Chapter 3 and the resulting *pdf* is a Rayleigh probability density function

$$f_R(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{7.12}$$

When (A/σ^2) is very large, Eq. (7.11) becomes a Gaussian probability density function of mean A and variance σ^2:

$$f_R(r) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r-A)^2}{2\sigma^2}\right) \tag{7.13}$$

Figure 7.2 shows plots for the Rayleigh and Gaussian densities.

The density function for the random variable Φ is obtained from

$$f_\Phi(\varphi) = \int_0^r f_{R\Phi}(r, \varphi)\ dr \tag{7.14}$$

While the detailed derivation is left as an exercise, the result of Eq. (7.14) is

$$f_\Phi(\varphi) = \frac{1}{2\pi} \exp\left(\frac{-A^2}{2\sigma^2}\right) + \frac{A\cos\varphi}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(A\sin\varphi)^2}{2\sigma^2}\right) F\left(\frac{A\cos\varphi}{\sigma}\right) \tag{7.15}$$

where

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}}\ e^{-\zeta^2/2}\ d\xi \tag{7.16}$$

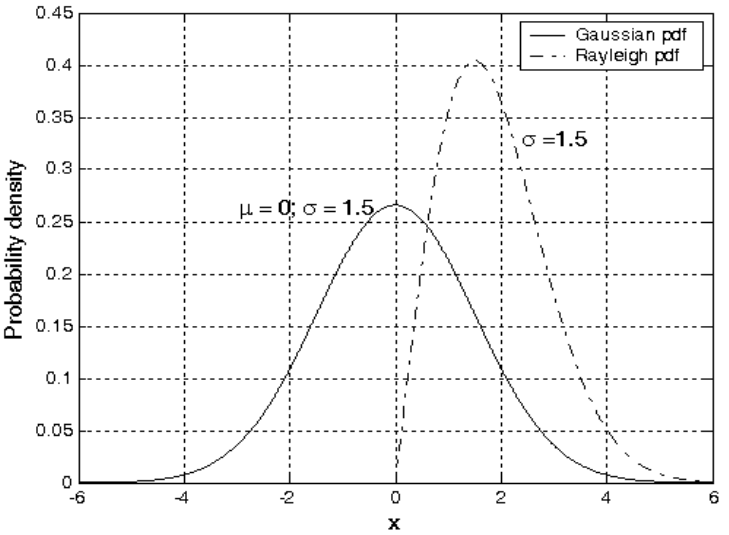


Figure 7.2. Gaussian and Rayleigh probability densities.

The function $F(x)$ can be found tabulated in most mathematical formula reference books. Note that for the case of noise alone ($A = 0$), Eq. (7.15) collapses to a uniform *pdf* over the interval $\{0, 2\pi\}$. One excellent approximation for the function $F(x)$ is

$$F(x) = 1 - \left(\frac{1}{0.661x + 0.339\sqrt{x^2 + 5.51}} \right) \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \qquad x \geq 0 \tag{7.17}$$

and for negative values of x

$$F(-x) = 1 - F(x) \tag{7.18}$$

7.2. Probability of False Alarm

The probability of false alarm P_{fa} is defined as the probability that a sample r of the signal $r(t)$ will exceed the threshold voltage v_T when noise alone is present in the radar:

$$P_{fa} = \int_{v_T}^{\infty} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2} \right) dr = \exp\left(\frac{-v_T^2}{2\sigma^2} \right) \tag{7.19}$$

$$v_T = \sqrt{2\sigma^2 \ln\left(\frac{1}{P_{fa}} \right)} \tag{7.20}$$

Figure 7.3 shows a plot of the normalized threshold versus the probability of false alarm. It is evident from this figure that P_{fa} is very sensitive to small changes in the threshold value. The false alarm time T_{fa} is related to the probability of false alarm by

$$T_{fa} = t_{int} / P_{fa} \tag{7.21}$$

where t_{int} represents the radar integration time, or the average time that the output of the envelope detector will pass the threshold voltage. Since the radar operating bandwidth B is the inverse of t_{int}, by substituting Eq. (7.19) into Eq. (7.20), we can write T_{fa} as

$$T_{fa} = \frac{1}{B} \exp\left(\frac{v_T^2}{2\sigma^2} \right) \tag{7.22}$$

Minimizing T_{fa} means increasing the threshold value, and as a result the radar maximum detection range is decreased. The choice of an acceptable value for T_{fa} becomes a compromise depending on the radar mode of operation.

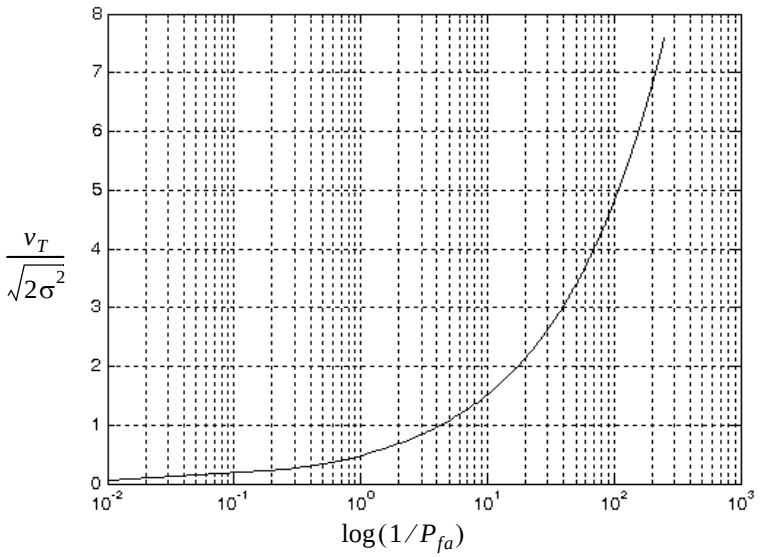


Figure 7.3. Normalized detection threshold versus probability of false alarm.

The false alarm number is defined as

$$n_{fa} = \frac{-\ln(2)}{\ln(1-P_{fa})} \approx \frac{\ln(2)}{P_{fa}} \tag{7.23}$$

Other slightly different definitions for the false alarm number exist in the literature, causing a source of confusion for many non-expert readers. Other than the definition in Eq. (7.23), the most commonly used definition for the false alarm number is the one introduced by Marcum (1960). Marcum defines the false alarm number as the reciprocal of P_{fa}. In this text, the definition given in Eq. (7.23) is always assumed. Hence, a clear distinction is made between Marcum's definition of the false alarm number and the definition in Eq. (7.23).

7.3. Probability of Detection

The probability of detection P_D is the probability that a sample r of $r(t)$ will exceed the threshold voltage in the case of noise plus signal,

$$P_D = \int_{v_T}^{\infty} \frac{r}{\sigma^2} I_0\left(\frac{rA}{\sigma^2}\right) \exp\left(-\frac{r^2+A^2}{2\sigma^2}\right) dr \tag{7.24}$$

If we assume that the radar signal is a sine waveform with amplitude A, then its power is $A^2/2$. Now, by using $SNR = A^2/2\sigma^2$ (single-pulse SNR) and $(v_T^2/2\sigma^2) = \ln(1/P_{fa})$, then Eq. (7.24) can be rewritten as

$$P_D = \int_{\sqrt{2\sigma^2 \ln(1/p_{fa})}}^{\infty} \frac{r}{\sigma^2} I_0\left(\frac{rA}{\sigma^2}\right) \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) dr = Q\left[\sqrt{\frac{A^2}{\sigma^2}}, \sqrt{2\ln\left(\frac{1}{P_{fa}}\right)}\right] \quad \textbf{(7.25)}$$

$$Q[\alpha, \beta] = \int_{\beta}^{\infty} \zeta I_0(\alpha\zeta) e^{-(\zeta^2 + \alpha^2)/2} \, d\zeta \quad \textbf{(7.26)}$$

Q is called Marcum's Q-function. When P_{fa} is small and P_D is relatively large so that the threshold is also large, Eq. (7.25) can be approximated by

$$P_D \approx F\left(\frac{A}{\psi} - \sqrt{2\ln\left(\frac{1}{P_{fa}}\right)}\right) \quad \textbf{(7.27)}$$

where $F(x)$ is given by Eq. (7.16). Many approximations for computing Eq. (7.25) can be found throughout the literature. One very accurate approximation presented by North (1963) is given by

$$P_D \approx 0.5 \times erfc(\sqrt{-\ln P_{fa}} - \sqrt{SNR + 0.5}) \quad \textbf{(7.28)}$$

where the complementary error function is

$$erfc(z) = 1 - \frac{2}{\sqrt{\pi}} \int_0^z e^{-v^2} dv \quad \textbf{(7.29)}$$

The integral given in Eq. (7.25) is complicated and can be computed using numerical integration techniques. Parl[1] developed an excellent algorithm to numerically compute this integral. It is summarized as follows:

$$Q[a, b] = \begin{cases} \frac{\alpha_n}{2\beta_n} \exp\left(\frac{(a-b)^2}{2}\right) & a < b \\ 1 - \left(\frac{\alpha_n}{2\beta_n} \exp\left(\frac{(a-b)^2}{2}\right)\right. & \left. a \geq b \right) \end{cases} \quad \textbf{(7.30)}$$

1. Parl, S., A New Method of Calculating the Generalized Q Function, *IEEE Trans. Information Theory,* Vol. IT-26, January 1980, pp. 121-124.

$$\alpha_n = d_n + \frac{2n}{ab}\alpha_{n-1} + \alpha_{n-2} \tag{7.31}$$

$$\beta_n = 1 + \frac{2n}{ab}\beta_{n-1} + \beta_{n-2} \tag{7.32}$$

$$d_{n+1} = d_n d_1 \tag{7.33}$$

$$\alpha_0 = \begin{cases} 1 & a < b \\ 0 & a \geq b \end{cases} \tag{7.34}$$

$$d_1 = \begin{cases} a/b & a < b \\ b/a & a \geq b \end{cases} \tag{7.35}$$

$\alpha_{-1} = 0.0$, $\beta_0 = 0.5$, and $\beta_{-1} = 0$. The recursive Eq. (7.30) through Eq. (7.33) are computed continuously until $\beta_n > 10^p$ for values of $p \geq 3$. The accuracy of the algorithm is enhanced as the value of p is increased. The MATLAB function "*marcumsq.m*" implements Parl's algorithm to calculate the probability of detection defined in Eq. (7.24). The syntax is as follows:

Pd = marcumsq(alpha, beta)

where *alpha* and *beta* are from Eq. (7.26). Figure 7.4 shows plots of the probability of detection, P_D, versus the single pulse SNR, with the P_{fa} as a parameter using this function. The following MATLAB program can be used to reproduce Fig. 7.4. It uses the function "*marcumsq.m.*"

```
% This program is used to produce Fig. 7.4
close all; clear all;
for nfa = 2:2:12
  b = sqrt(-2.0 * log(10^(-nfa)));
  index = 0;
  hold on
  for snr = 0:.1:18
    index = index +1;
    a = sqrt(2.0 * 10^(.1*snr));
    pro(index) = marcumsq(a,b);
  end
  x = 0:.1:18;
  set(gca,'ytick',[.1 .2 .3 .4 .5 .6  .7 .75 .8 .85 .9 .95 .9999])
  set(gca,'xtick',[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18])
   loglog(x, pro,'k');
end
hold off
xlabel ('Single pulse SNR in dB'); ylabel ('Probability of detection')
grid
```

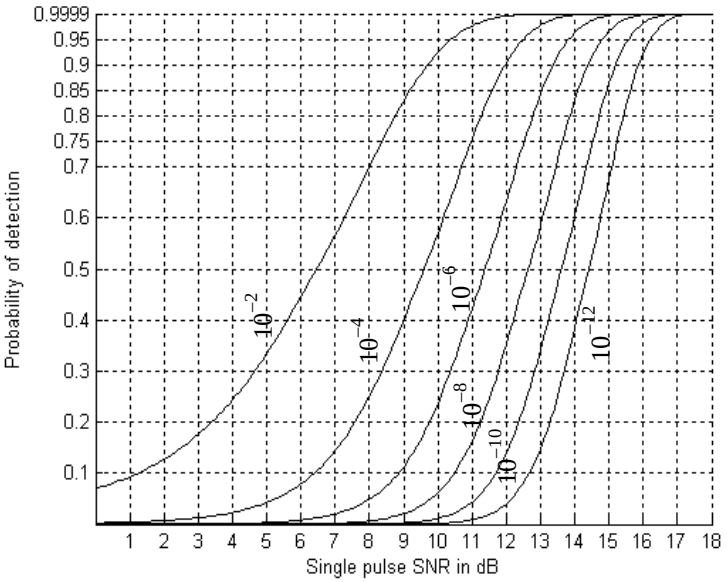


Figure 7.4. Probability of detection versus single pulse SNR, for several values of P_{fa}.

7.4. Pulse Integration

When a target is located within the radar beam during a single scan, it may reflect several pulses. By adding the returns from all pulses returned by a given target during a single scan, the radar sensitivity (SNR) can be increased. The number of returned pulses depends on the antenna scan rate and the radar PRF. More precisely, the number of pulses returned from a given target is given by

$$n_P = \frac{\theta_a T_{sc} f_r}{2\pi} \tag{7.36}$$

where θ_a is the azimuth antenna beamwidth, T_{sc} is the scan time, and f_r is the radar PRF. The number of reflected pulses may also be expressed as

$$n_P = \frac{\theta_a f_r}{\dot{\theta}_{scan}} \tag{7.37}$$

where $\dot{\theta}_{scan}$ is the antenna scan rate in degrees per second. Note that when using Eq. (7.36), θ_a is expressed in radians, while when using Eq. (7.37), it is expressed in degrees. As an example, consider a radar with an azimuth antenna

beamwidth $\theta_a = 3°$, antenna scan rate $\dot{\theta}_{scan} = 45°/\text{sec}$ (antenna scan time, $T_{sc} = 8\,\text{sec}$), and a PRF $f_r = 300Hz$. Using either Eq. (7.36) or Eq. (7.37) yields $n_P = 20$ pulses.

The process of adding radar returns from many pulses is called radar pulse integration. Pulse integration can be performed on the quadrature components prior to the envelope detector. This is called coherent integration or predetection integration. Coherent integration preserves the phase relationship between the received pulses. Thus a buildup in the signal amplitude is achieved. Alternatively, pulse integration performed after the envelope detector (where the phase relation is destroyed) is called noncoherent or postdetection integration.

Radar designers should exercise caution when utilizing pulse integration for the following reasons. First, during a scan a given target will not always be located at the center of the radar beam (i.e., have maximum gain). In fact, during a scan a given target will first enter the antenna beam at the 3-dB point, reach maximum gain, and finally leave the beam at the 3-dB point again. Thus, the returns do not have the same amplitude even though the target RCS may be constant and all other factors that may introduce signal loss remain the same.

Other factors that may introduce further variation to the amplitude of the returned pulses include target RCS and propagation path fluctuations. Additionally, when the radar employs a very fast scan rate, an additional loss term is introduced due to the motion of the beam between transmission and reception. This is referred to as scan loss. A distinction should be made between scan loss due to a rotating antenna (which is described here) and the term scan loss that is normally associated with phased array antennas (which takes on a different meaning in that context).

Finally, since coherent integration utilizes the phase information from all integrated pulses, it is critical that any phase variation between all integrated pulses be known with a great level of confidence. Consequently, target dynamics (such as target range, range rate, tumble rate, RCS fluctuation) must be estimated or computed accurately so that coherent integration can be meaningful. In fact, if a radar coherently integrates pulses from targets without proper knowledge of the target dynamics, it suffers a loss in SNR rather than the expected SNR buildup. Knowledge of target dynamics is not as critical when employing noncoherent integration; nonetheless, target range rate must be estimated so that only the returns from a given target within a specific range bin are integrated. In other words, one must avoid range walk (i.e., having a target cross between adjacent range bins during a single scan).

A comprehensive analysis of pulse integration should take into account issues such as the probability of detection P_D, probability of false alarm P_{fa}, the target statistical fluctuation model, and the noise or interference of statistical models. This is the subject of the rest of this chapter.

7.4.1. Coherent Integration

In coherent integration, when a perfect integrator is used (100% efficiency), to integrate n_P pulses, the SNR is improved by the same factor. Otherwise, integration loss occurs, which is always the case for noncoherent integration. Coherent integration loss occurs when the integration process is not optimum. This could be due to target fluctuation, instability in the radar local oscillator, or propagation path changes.

Denote the single pulse SNR required to produce a given probability of detection as $(SNR)_1$. The SNR resulting from coherently integrating n_P pulses is then given by

$$(SNR)_{CI} = n_P(SNR)_1 \tag{7.38}$$

Coherent integration cannot be applied over a large number of pulses, particularly if the target RCS is varying rapidly. If the target radial velocity is known and no acceleration is assumed, the maximum coherent integration time is limited to

$$t_{CI} = \sqrt{\lambda / 2a_r} \tag{7.39}$$

where λ is the radar wavelength and a_r is the target radial acceleration. Coherent integration time can be extended if the target radial acceleration can be compensated for by the radar.

In order to demonstrate the improvement in the SNR using coherent integration, consider the case where the radar return signal contains both signal plus additive noise. The mth pulse is

$$y_m(t) = s(t) + n_m(t) \tag{7.40}$$

where $s(t)$ is the radar signal return of interest and $n_m(t)$ is white uncorrelated additive noise signal with variance σ^2. Coherent integration of n_P pulses yields

$$z(t) = \frac{1}{n_P}\sum_{m=1}^{n_P} y_m(t) = \sum_{m=1}^{n_P}\frac{1}{n_P}[s(t) + n_m(t)] = s(t) + \sum_{m=1}^{n_P}\frac{1}{n_P}n_m(t) \tag{7.41}$$

The total noise power in $z(t)$ is equal to the variance. More precisely,

$$\sigma^2_{n_P} = E\left[\left(\sum_{m=1}^{n_P}\frac{1}{n_P}n_m(t)\right)\left(\sum_{l=1}^{n_P}\frac{1}{n_P}n_l(t)\right)^*\right] \tag{7.42}$$

where E is the expected value operator. It follows that

$$\sigma_{n_P}^2 = \frac{1}{n_P^2} \sum_{m,\, l = 1}^{n_P} E[n_m(t) n_l^*(t)] = \frac{1}{n_P^2} \sum_{m,\, l = 1}^{n_P} \sigma_{ny}^2 \delta_{ml} = \frac{1}{n_P} \sigma_{ny}^2 \qquad \textbf{(7.43)}$$

where σ_{ny}^2 is the single pulse noise power and δ_{ml} is equal to zero for $m \neq l$ and unity for $m = l$. Observation of Eqs. (7.41) and (7.42) shows that the desired signal power after coherent integration is unchanged, while the noise power is reduced by the factor $1/n_P$. Thus, the SNR after coherent integration is improved by n_P.

7.4.2. Noncoherent Integration

When the phase of the integrated pulses is not known so that coherent integration is no longer possible, another form of pulse integration is done. In this case, pulse integration is performed by adding (integrating) the individual pulses' envelopes or the square of their envelopes. Thus, the term noncoherent integration is adopted. A block diagram of radar receiver utilizing noncoherent integration is illustrated in Fig. 7.5.

The performance difference (measured in SNR) between the linear envelope detector and the quadratic (square law) detector is practically negligible. Robertson (1967) showed that this difference is typically less than $0.2dB$; he showed that the performance difference is higher than $0.2dB$ only for cases where $n_P > 100$ and $P_D < 0.01$. Both of these conditions are of no practical significance in radar applications. It is much easier to analyze and implement the square law detector in real hardware than is the case for the envelope detector. Therefore, most authors make no distinction between the type of detector used when referring to noncoherent integration, and the square law detector is almost always assumed. The analysis presented in this book will always assume, unless indicated otherwise, noncoherent integration using the square law detector.

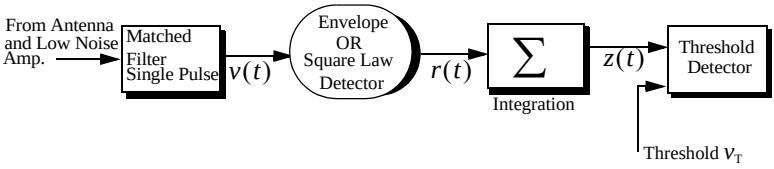


Figure 7.5. Simplified block diagram of a radar detector when noncoherent integration is used.

7.4.3. Improvement Factor and Integration Loss

Noncoherent integration is less efficient than coherent integration. Actually, the noncoherent integration gain is always smaller than the number of noncoherently integrated pulses. This loss in integration is referred to as postdetection or square-law detector loss.

Define $(SNR)_{NCI}$ as the SNR required to achieve a specific P_D given a particular P_{fa} when n_P pulses are integrated noncoherently. Also denote the single pulse SNR as $(SNR)_1$. It follows that

$$(SNR)_{NCI} = (SNR)_1 \times I(n_P) \tag{7.44}$$

where $I(n_P)$ is called the integration improvement factor. An empirically derived expression for the improvement factor that is accurate within $0.8dB$ is reported in Peebles (1998) as

$$[I(n_P)]_{dB} = 6.79(1 + 0.253P_D)\left(1 + \frac{\log(1/P_{fa})}{46.6}\right)\log(n_P) \tag{7.45}$$
$$(1 - 0.140\log(n_P) + 0.018310(\log n_P)^2)$$

The top part of Fig. 7.6 shows plots of the integration improvement factor as a function of the number of integrated pulses with P_D and P_{fa} as parameters using Eq. (7.45). The integration loss in *dB* is defined as

$$[L_{NCI}]_{dB} = 10\log n_P - [I(n_P)]_{dB} \tag{7.46}$$

The lower part of Fig. 7.6 shows plots of the corresponding integration loss versus n_P with P_D and P_{fa} as parameters. This figure can be reproduced using the following MATLAB code which uses MATLAB function *"improv_fac.m."*

```
% This program is used to produce Fig. 7.6
% It uses the function "improv_fac.m".
clear all;
close all;
Pfa = [1e-2, 1e-6, 1e-8, 1e-10];
Pd = [.5 .8 .95 .99];
np = linspace(1,1000,10000);
I(1,:) = improv_fac (np, Pfa(1), Pd(1));
I(2,:) = improv_fac (np, Pfa(2), Pd(2));
I(3,:) = improv_fac (np, Pfa(3), Pd(3));
I(4,:) = improv_fac (np, Pfa(4), Pd(4));
index = [1 2 3 4];
L(1,:) = 10.*log10(np) - I(1,:);
L(2,:) = 10.*log10(np) - I(2,:);
L(3,:) = 10.*log10(np) - I(3,:);
```

```
L(4,:) = 10.*log10(np) - I(4,:);
subplot(2,1,2);
semilogx (np, L(1,:), 'k:', np, L(2,:), 'k-.', ...
np, L(3,:), 'k-.', np, L(4,:), 'k')
xlabel ('Number of pulses');
ylabel ('Integration loss in dB')
axis tight; grid
subplot(2,1,1);
semilogx (np, I(1,:), 'k:', np, I(2,:), 'k-.', np, ...
 I(3,:), 'k--', np, I(4,:), 'k')
xlabel ('Number of pulses');
ylabel ('Improvement factor in dB')
legend ('pd=.5, Pfa=1e-2','pd=.8, Pfa=1e-6','pd=.95, ...
Pfa=1e-8','pd=.99, Pfa=1e-10');
grid;
axis tight
```

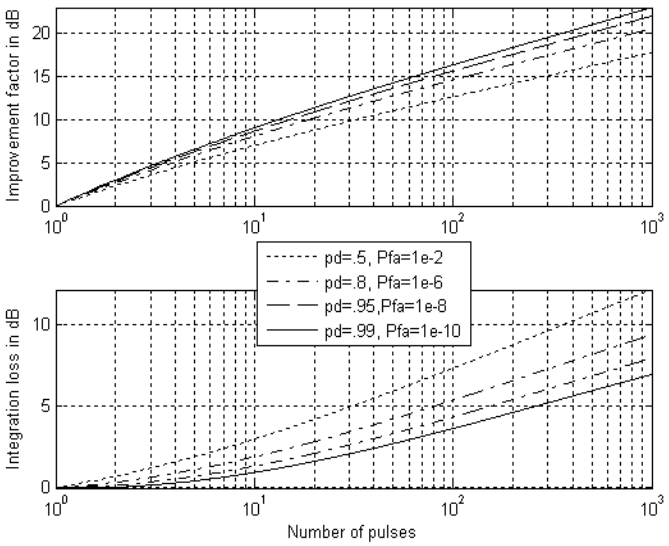


Figure 7.6. Typical plots for the improvement factor and integration loss versus number of noncoherently integrated pulses.

7.5. Target Fluctuation

Target detection utilizing the square law detector was first analyzed by Marcum[1], where he assumed a constant RCS (nonfluctuating target). This work was extended by Swerling[2] to four distinct cases of target RCS fluctuation. These cases have come to be known as Swerling models. They are Swerling I, Swerling II, Swerling III, and Swerling IV. The constant RCS case analyzed by Marcum is widely known as Swerling 0 or equivalently Swerling V. Target fluctuation introduces an additional loss factor in the SNR as compared to the case where fluctuation is not present given the same P_D and P_{fa}.

Swerling I targets have constant amplitude over one antenna scan or observation interval; however, a Swerling I target amplitude varies independently from scan to scan according to a chi-square probability density function with two degrees of freedom. The amplitude of Swerling II targets fluctuates independently from pulse to pulse according to a chi-square probability density function with two degrees of freedom. Target fluctuation associated with a Swerling III model is from scan to scan according to a chi-square probability density function with four degrees of freedom. Finally, the fluctuation of Swerling IV targets is from pulse to pulse according to a chi-square probability density function with four degrees of freedom.

Swerling showed that the statistics associated with Swerling I and II models apply to targets consisting of many small scatterers of comparable RCS values, while the statistics associated with Swerling III and IV models apply to targets consisting of one large RCS scatterer and many small equal RCS scatterers. Noncoherent integration can be applied to all four Swerling models; however, coherent integration cannot be used when the target fluctuation is either Swerling II or Swerling IV. This is because the target amplitude decorrelates from pulse to pulse (fast fluctuation) for Swerling II and IV models, and thus phase coherency cannot be maintained.

The chi-square *pdf* with $2N$ degrees of freedom can be written as

$$f_X(x) = \frac{N}{(N-1)!\ \sqrt{\sigma_x^2}} \left(\frac{Nx}{\sigma_x}\right)^{N-1} \exp\left(-\frac{Nx}{\sigma_x}\right) \qquad \textbf{(7.47)}$$

where σ_x is the standard deviation for the RCS value. Using this equation, the *pdf* associated with Swerling I and II targets can be obtained by letting $N = 1$, which yields a Rayleigh *pdf*. More precisely,

---

1. Marcum, J. I., A Statistical Theory of Target Detection by Pulsed Radar, *IRE Transactions on Information Theory*, Vol IT-6, pp. 59-267, April 1960.
2. Swerling, P., Probability of Detection for Fluctuating Targets, *IRE Transactions on Information Theory*, Vol IT-6, pp. 269-308, April 1960.

$$f_X(x) = \frac{1}{\sigma_x}\exp\left(-\frac{x}{\sigma_x}\right) \qquad x \geq 0 \tag{7.48}$$

Letting $N = 2$ yields the *pdf* for Swerling III and IV type targets,

$$f_X(x) = \frac{4x}{\sigma_x^2}\exp\left(-\frac{2x}{\sigma_x}\right) \qquad x \geq 0 \tag{7.49}$$

7.6. Probability of False Alarm Formulation for a Square Law Detector

Computation of the general formula for the probability of false alarm P_{fa} and subsequently the rest of square law detection theory requires knowledge and good understating of the incomplete Gamma function. Hence, those readers who are not familiar with this function are advised to read Appendix 7.A before proceeding with the rest of this chapter.

DiFranco and Rubin[1] derived a general form relating the threshold and P_{fa} for any number of pulses when noncoherent integration is used. The square law detector under consideration is shown in Fig. 7.7. There are $n_P \geq 2$ pulses integrated noncoherently and the noise power (variance) is σ^2.

The complex envelope in terms of the quadrature components is given by

$$\tilde{r}(t) = r_I(t) + jr_Q(t) \tag{7.50}$$

thus, the square of the complex envelope is

$$|\tilde{r}(t)|^2 = r_I^2(t) + r_Q^2(t) \tag{7.51}$$

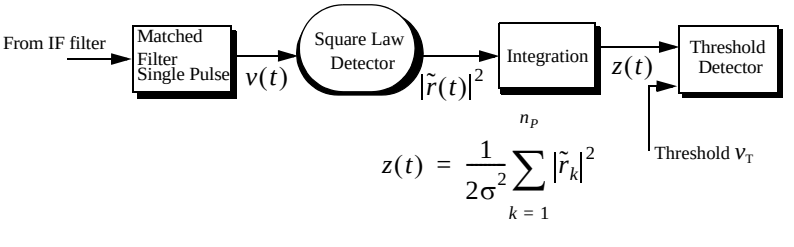


Figure 7.7. Square law detector.

1. DiFranco, J. V. and Rubin, W. L., *Radar Detection*, Artech House, Norwood, MA 1980.

The samples $|\tilde{r}_k|^2$ are computed from the samples of $\tilde{r}(t)$ evaluated at $t = t_k;\ k = 1, 2, \ldots, n_P$. It follows that

$$Z = \frac{1}{2\sigma^2} \sum_{k=1}^{n_P} [r_I^2(t_k) + r_Q^2(t_k)] \tag{7.52}$$

The random variable Z is the sum of $2n_P$ squares of random variables, each of which is a Gaussian random variable with variance σ^2. Thus, using the analysis developed in Chapter 3, the *pdf* for the random variable Z is given by

$$f_Z(z) = \left\{ \begin{array}{ll} \dfrac{z^{n_P-1} e^{-z}}{\Gamma(n_P)} & z \geq 0 \\ 0 & z < 0 \end{array} \right\} \tag{7.53}$$

Consequently, the probability of false alarm given a threshold value v_Y is

$$P_{fa} = Prob\{Z \geq v_T\} = \int_{v_T}^{\infty} \frac{z^{n_P-1} e^{-z}}{\Gamma(n_P)}\, dz \tag{7.54}$$

and using analysis provided in Appendix 7.A yields

$$P_{fa} = 1 - \Gamma_I\left(\frac{v_T}{\sqrt{n_P}}, n_P - 1\right) \tag{7.55}$$

Using the algebraic expression for the incomplete Gamma function, Eq. (7.55) can be written as

$$P_{fa} = e^{-v_T} \sum_{k=0}^{n_P-1} \frac{v_T^k}{k!} = 1 - e^{-v_T} \sum_{k=n_P}^{\infty} \frac{v_T^k}{k!} \tag{7.56}$$

The threshold value v_T can then be approximated by the recursive formula used in the Newton-Raphson method. More precisely,

$$v_{T,m} = v_{T,m-1} - \frac{G(v_{T,m-1})}{G'(v_{T,m-1})} \quad ;\ m = 1, 2, 3, \ldots \tag{7.57}$$

The iteration is terminated when $|v_{T,m} - v_{T,m-1}| < v_{T,m-1}/10000.0$. The functions G and G' are

$$G(v_{T,m}) = (0.5)^{n_P/n_{fa}} - \Gamma_I(v_T, n_P) \tag{7.58}$$

$$G'(v_{T,m}) = -\frac{e^{-v_T} v_T^{n_P-1}}{(n_P-1)!} \tag{7.59}$$

The initial value for the recursion is

$$v_{T,0} = n_P - \sqrt{n_P} + 2.3\sqrt{-\log P_{fa}}\left(\sqrt{-\log P_{fa}} + \sqrt{n_P} - 1\right) \tag{7.60}$$

Figure 7.8 shows plots of the threshold value versus the number of integrated pulses for several values of n_{fa}; remember that $P_{fa} \approx \ln(2)/n_{fa}$. This figure can be reproduced using the following MATLAB code which utilizes the MATLAB function *"threshold.m"*

```
% Use this program to reproduce Fig. 7.8 of text
clear all; close all;
for n= 1: 1:10000
  [pfa1 y1(n)] = threshold(1e4,n);
 [pfa2 y3(n)] = threshold(1e8,n);
 [pfa3 y4(n)] = threshold(1e12,n);
end
n =1:1:10000;
loglog(n,y1,'k',n,y3,'k--',n,y4,'k-.');
xlabel ('Number of pulses');
ylabel ('Threshold');
legend('nfa=1e4','nfa=1e8','nfa=1e12'); grid
```

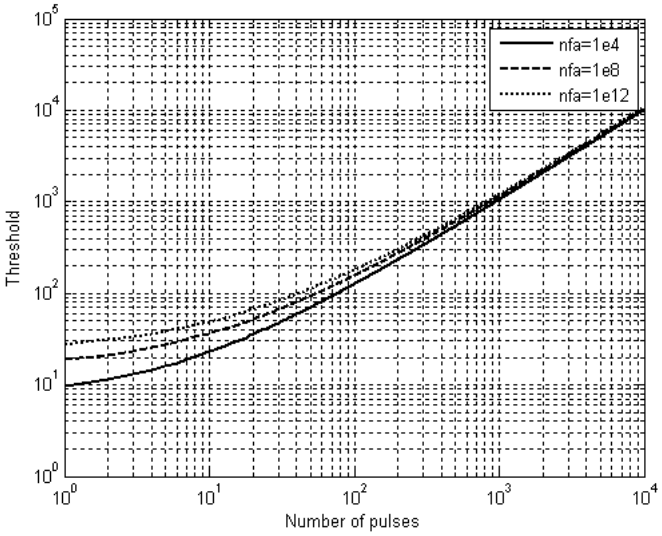


Figure 7.8. Threshold v_T versus n_p for several values of n_{fa}.

7.6.1. Square Law Detection

The *pdf* for the linear envelope $r(t)$ was derived earlier and it is given in Eq. (7.11). Define a new dimensionless variable y as

$$y_n = r_n/\sigma \tag{7.61}$$

and also define

$$\Re_p = A^2/\sigma^2 = 2SNR \tag{7.62}$$

σ^2 is the noise variance. It follows that the *pdf* for the new variable is

$$f_{Y_n}(y_n) = f_{R_n}(r_n)\left|\frac{dr_n}{dy_n}\right| = y_n \; I_0(y_n\sqrt{\Re_p}) \; \exp\left(\frac{-(y_n^2 + \Re_p)}{2}\right) \tag{7.63}$$

The output of a square law detector for the nth pulse is proportional to the square of its input. Thus, it is convenient to define a new change variable,

$$z_n = \frac{1}{2}y_n^2 \tag{7.64}$$

The *pdf* for the variable at the output of the square law detector is given by

$$f_{Z_n}(x_n) = f(y_n)\left|\frac{dy_n}{dz_n}\right| = \exp\left(-\left(z_n + \frac{\Re_p}{2}\right)\right) I_0(\sqrt{2z_n\Re_p}) \tag{7.65}$$

Noncoherent integration of n_p pulses is implemented as

$$z = \sum_{n=1}^{n_P} \frac{1}{2}y_n^2 \tag{7.66}$$

Again, $n_P \geq 2$. Since the random variables y_n are independent, the *pdf* for the variable z is

$$f(z) = f((y_1) \otimes f(y_2) \otimes \ldots \otimes f(y_{n_p})) \tag{7.67}$$

The operator $\otimes$ symbolically indicates convolution. The characteristic functions for the individual *pdf*s can then be used to compute the joint *pdf* for Eq. (7.69). The result is

$$f_Z(z) = \left(\frac{2z}{n_P\Re_p}\right)^{(n_P-1)/2} \exp\left(-z - \frac{1}{2}n_P\Re_p\right) I_{n_P-1}(\sqrt{2n_Pz\Re_p}) \tag{7.68}$$

I_{n_P-1} is the modified Bessel function of order $n_P - 1$. Substituting Eq. (7.62) into (7.68) yields

$$f_Z(z) = \left(\frac{z}{n_P SNR}\right)^{(n_P-1)/2} e^{(-z-n_P SNR)} I_{n_p-1}(2\sqrt{n_P z SNR}) \quad \textbf{(7.69)}$$

When target fluctuation is not present (i.e., Swerling 0), the probability of detection is obtained by integrating $f_Z(z)$ from the threshold value to infinity. The probability of false alarm is obtained by letting $\Re_p$ be zero and integrating the *pdf* from the threshold value to infinity. More specifically,

$$P_D\big|_{SNR} = \int_{v_T}^{\infty} \left(\frac{z}{n_P SNR}\right)^{(n_P-1)/2} e^{(-z-n_P SNR)} I_{n_p-1}(2\sqrt{n_P z SNR})dz \quad \textbf{(7.70)}$$

Which can be rewritten as

$$P_D\big|_{SNR} = e^{-n_P SNR}\left(\sum_{k=0}^{\infty}\frac{(n_P SNR)^k}{k!}\right)\left(\sum_{j=0}^{n_P-1+k}\frac{e^{-v_T}\, v_T^j}{j!}\right) \quad \textbf{(7.71)}$$

Alternatively, when target fluctuation is present, then the *pdf* is calculated using the conditional probability density function of Eq. (7.70) with respect to the SNR value of the target fluctuation type. In general, given a fluctuating target with SNR^F, where the superscript indicates fluctuation, the expression for the probability of detection is

$$P_D\big|_{SNR^F} = \int_0^{\infty} P_D\big|_{SNR}\, f_Z(z^F/SNR^F)dz = \quad \textbf{(7.72)}$$

$$\int_0^{\infty} P_D\big|_{SNR}\left(\frac{z^F}{n_P SNR^F}\right)^{(n_P-1)/2} e^{(-z^F-n_P SNR^F)} I_{n_p-1}(2\sqrt{n_P z^F SNR^F})dz$$

Remember that target fluctuation introduces an additional loss term in the SNR. It follows that for the same P_D given the same P_{fa} and the same n_P, $SNR^F > SNR$. One way to calculate this additional SNR is to first compute the required SNR given no fluctuation then add to it the amount of target fluctuation loss to get the required value for SNR^F. How to calculate this fluctuation loss will be addressed later on in this chapter. Meanwhile, hereon after, the superscript $\{^F\}$ will be dropped and it will always be assumed.

7.7. Probability of Detection Calculation

Marcum defined the probability of false alarm for the case when $n_P > 1$ as

$$P_{fa} \approx \ln(2)(n_P/n_{fa}) \quad \textbf{(7.73)}$$

The single pulse probability of detection for nonfluctuating targets is given in Eq. (7.25). When $n_P > 1$, the probability of detection is computed using the Gram-Charlier series. In this case, the probability of detection is

$$P_D \cong \frac{erfc(V/\sqrt{2})}{2} - \frac{e^{-V^2/2}}{\sqrt{2\pi}}[C_3(V^2-1) + C_4V(3-V^2) - C_6V(V^4 - 10V^2 + 15)] \quad \textbf{(7.74)}$$

where the constants C_3, C_4, and C_6 are the Gram-Charlier series coefficients, and the variable V is

$$V = \frac{v_T - n_P(1+SNR)}{\varpi} \quad \textbf{(7.75)}$$

In general, values for C_3, C_4, C_6, and ϖ vary depending on the target fluctuation type.

7.7.1. Swerling 0 Target Detection

For Swerling 0 (Swerling V) target fluctuations, the probability of detection is calculated using Eq. (7.74). In this case, the Gram-Charlier series coefficients are

$$C_3 = -\frac{SNR + 1/3}{\sqrt{n_p}(2SNR+1)^{1.5}} \quad \textbf{(7.76)}$$

$$C_4 = \frac{SNR + 1/4}{n_p(2SNR+1)^2} \quad \textbf{(7.77)}$$

$$C_6 = C_3^2/2 \quad \textbf{(7.78)}$$

$$\varpi = \sqrt{n_p(2SNR+1)} \quad \textbf{(7.79)}$$

Figure 7.9 shows a plot for the probability of detection versus SNR for cases $n_p = 1, 10$. Note that it requires less SNR, with ten pulses integrated noncoherently, to achieve the same probability of detection as in the case of a single pulse. Hence, for any given P_D the SNR improvement can be read from the plot. Equivalently, using the function *"improv_fac.m"* leads to about the same result. For example, when $P_D = 0.8$, the function *"improv_fac.m"* gives an SNR improvement factor of $I(10) \approx 8.55dB$. Figure 7.9 shows that the ten pulse SNR is about $6.03dB$. Therefore, the single pulse SNR is about $14.5dB$, which can be read from the figure.

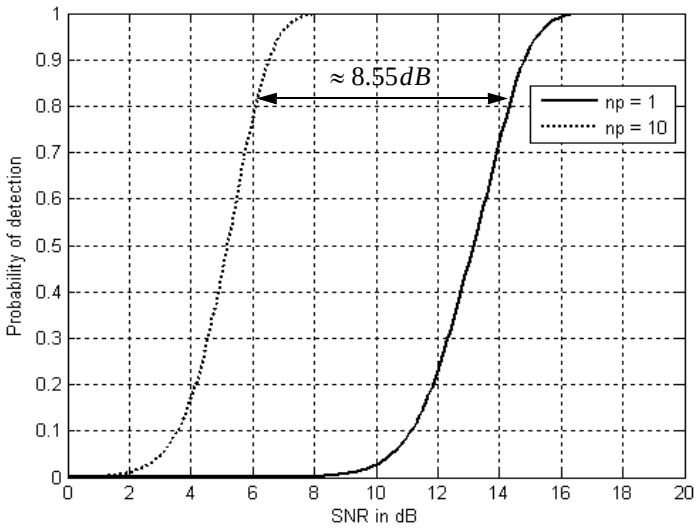


Figure 7.9. Probability of detection versus SNR, $P_{fa} = 10^{-9}$, and noncoherent integration; Swerling 0.

7.7.2. Detection of Swerling I Targets

The exact formula for the probability of detection for Swerling I type targets was derived by Swerling. It is

$$P_D = e^{-(v_T)/(1+SNR)} \quad ; \; n_P = 1 \tag{7.80}$$

$$P_D = 1 - \Gamma_I(v_T, n_P - 1) + \left(1 + \frac{1}{n_P SNR}\right)^{n_p - 1} \Gamma_I\left(\frac{v_T}{1 + \frac{1}{n_P SNR}}, n_P - 1\right) \tag{7.81}$$

$$\times \; e^{-v_T/(1+n_P SNR)} \quad ; \; n_P > 1$$

Figure 7.10 shows a plot of the probability of detection as a function of SNR for $n_p = 1$ and $P_{fa} = 10^{-9}$ for both Swerling I and V (Swerling 0) type fluctuations. Note that it requires more SNR, with fluctuation, to achieve the same P_D as in the case with no fluctuation. This figure can be reproduced using the following MATLAB code.

```
% Generate Figure 7.10
close all;
clear all;
pfa = 1e-9;
nfa = log(2) / pfa;
b = sqrt(-2.0 * log(pfa));
index = 0;
for snr = 0:.01:22
   index = index +1;
   a = sqrt(2.0 * 10^(.1*snr));
  swer0(index) = marcumsq(a,b);
   swer1(index) =  pd_swerling1 (nfa, 1, snr);
end
x = 0:.01:22;
%figure(10)
plot(x, swer0,'k',x,swer1,'k:');
axis([2 22 0 1])
xlabel ('SNR in dB')
ylabel ('Probability of detection')
legend('Swerling 0','Swerling I')
grid
```

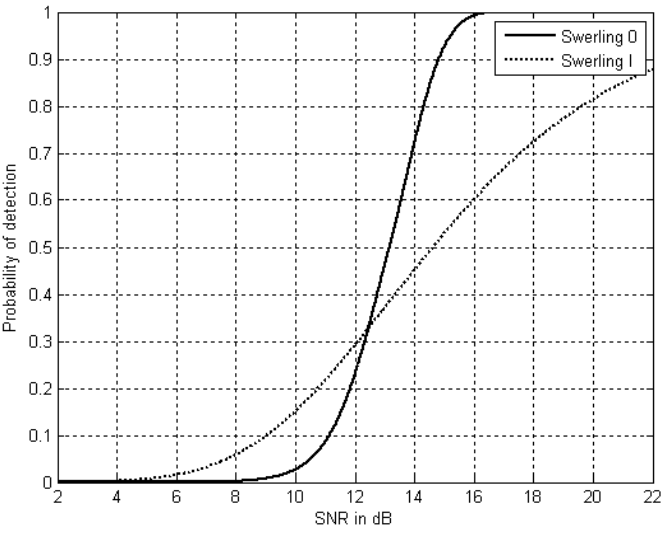


Figure 7.10. Probability of detection versus SNR, single pulse. $P_{fa} = 10^{-9}$.

Figure 7.11 is similar to Fig. 7.10 except in this case $P_{fa} = 10^{-6}$ and $n_P = 5$. This figure can be reproduced using the following MATLAB code

```
% Generate Figure 7.11
clear all
close all
pfa = 1e-6;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
  prob1(index) =  pd_swerling1 (nfa, 5, snr);
  prob0(index) =  pd_swerling5 (nfa, 2, 5, snr);
 end
x = -10:.5:30;
plot(x, prob1,'k',x,prob0,'k:');
axis([-10 30 0 1])
xlabel ('SNR in dB')
ylabel ('Probability of detection')
legend('Swerling I','Swerling 0')
title('Pfa =1e-6;  n=5')
grid
```

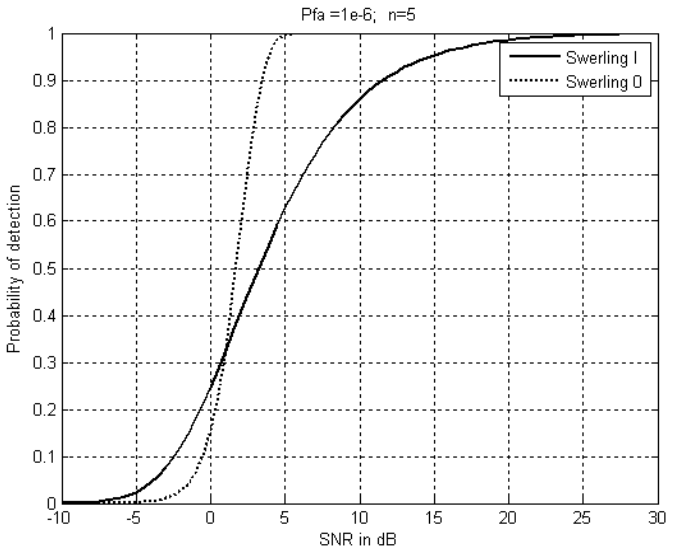


Figure 7.11. Probability of detection versus SNR. Swerling I and Swerling 0.

7.7.3. Detection of Swerling II Targets

In the case of Swerling II targets, the probability of detection is given by

$$P_D = 1 - \Gamma_I\left(\frac{v_T}{(1+SNR)}\ , n_p\right) \qquad ;\ n_P \le 50 \tag{7.82}$$

For the case when $n_P > 50$ the probability of detection is computed using the Gram-Charlier series. In this case,

$$C_3 = -\frac{1}{3\sqrt{n_p}} \qquad ,\ C_6 = \frac{C_3^2}{2} \tag{7.83}$$

$$C_4 = \frac{1}{4n_P} \tag{7.84}$$

$$\varpi = \sqrt{n_P}\ (1+SNR) \tag{7.85}$$

Figure 7.12a shows a plot of the probability of detection for Swerling 0, Swerling I, and Swerling II with $n_P = 5$, where $P_{fa} = 10^{-7}$. This figure can be reproduced using the following MATLAB code. Figure 7.12b is similar to Fig. 7.12a except in this case $n_P = 2$.

```
% Generate Figure 7.12
clc
clear all;
close all;
pfa = 1e-7;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
  prob1(index) =  pd_swerling1 (nfa, 5, snr); % Fig. 7.12a
  prob0(index) =  pd_swerling5 (nfa, 2, 5, snr); % Fig. 7.12a
  prob2(index) =  pd_swerling2 (nfa, 5, snr); % Fig. 7.12a
  % prob1(index) =  pd_swerling1 (nfa, 2, snr); % Fig. 7.12b
  % prob0(index) =  pd_swerling5 (nfa, 2, 2, snr);  % Fig. 7.12b
  % prob2(index) =  pd_swerling2 (nfa, 2, snr);  % Fig. 7.12b
end
x = -10:.5:30;
plot(x, prob0,'k',x,prob1,'k:',x,prob2,'k--');
axis([-10 30 0 1])
xlabel ('SNR in dB')
ylabel ('Probability of detection')
legend('Swerling 0','Swerling I','Swerling II')
title('Pfa =1e-7;  n=5')
grid
```

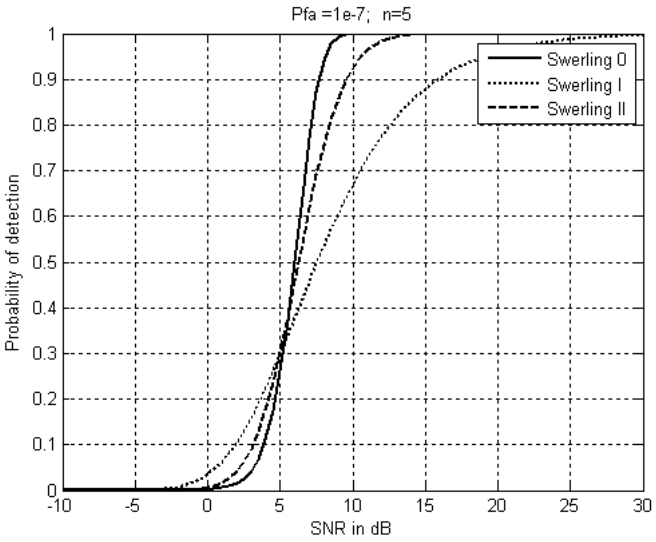


Figure 7.12a. Probability of detection versus SNR. Swerling II, Swerling I and Swerling 0.

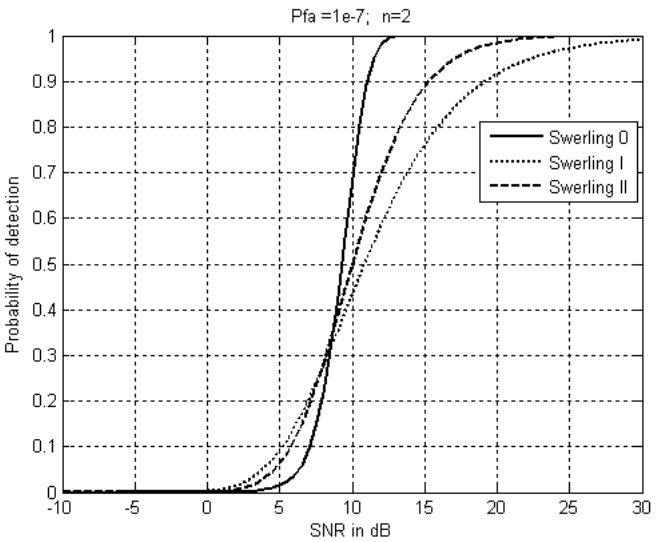


Figure 7.12b. Probability of detection versus SNR. Swerling II, Swerling I, and Swerling 0.

7.7.4. Detection of Swerling III Targets

The exact formulas, developed by Marcum, for the probability of detection for Swerling III type targets when $n_P = 1, 2$

$$P_D = \exp\left(\frac{-v_T}{1 + n_P SNR/2}\right)\left(1 + \frac{2}{n_P SNR}\right)^{n_P - 2} \times K_0$$
$$K_0 = 1 + \frac{v_T}{1 + n_P SNR/2} - \frac{2}{n_P SNR}(n_P - 2) \quad \textbf{(7.86)}$$

For $n_P > 2$ the expression is

$$P_D = \frac{v_T^{n_P - 1} \; e^{-V_T}}{(1 + n_P SNR/2)(n_P - 2)!} + 1 - \Gamma_I(v_T, n_P - 1) + K_0 \quad \textbf{(7.87)}$$
$$\times \; \Gamma_I\left(\frac{v_T}{1 + 2/n_p SNR}, n_p - 1\right)$$

Figure 7.13a shows a plot of the probability of detection as a function of SNR for $n_P = 1, 10, 50, 100$, where $P_{fa} = 10^{-9}$. Figure 7.13b shows a plot of the probability of detection for Swerling 0, Swerling I, Swerling II, and Swerling III with $n_P = 5$ and $P_{fa} = 10^{-7}$. Figure 7.13a can be reproduced using the following MATLAB code.

```
% Generate Figure 7.13a
close all;
clear all;
pfa = 1e-9;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
  prob1(index) =  pd_swerling3 (nfa, 1, snr);
  prob10(index) =  pd_swerling3 (nfa, 10, snr);
  prob50(index) =  pd_swerling3(nfa, 50, snr);
  prob100(index) =  pd_swerling3 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', x, prob100,'k-.');
axis([-10 30 0 1])
xlabel ('SNR in dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

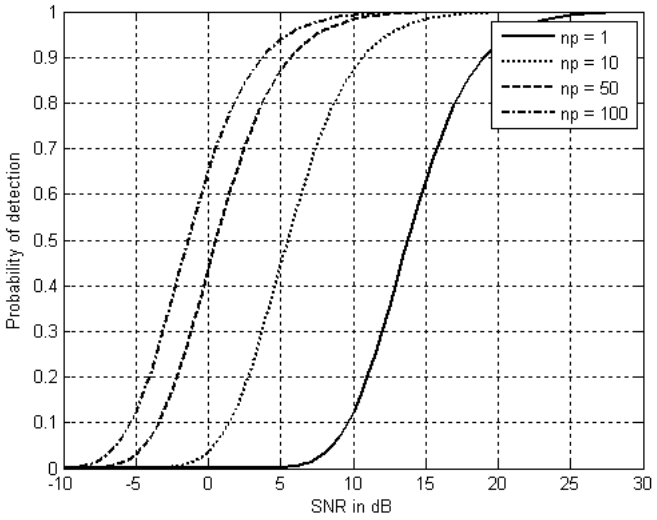


Figure 7.13a. Probability of detection versus SNR. Swerling III. $P_{fa} = 10^{-9}$.

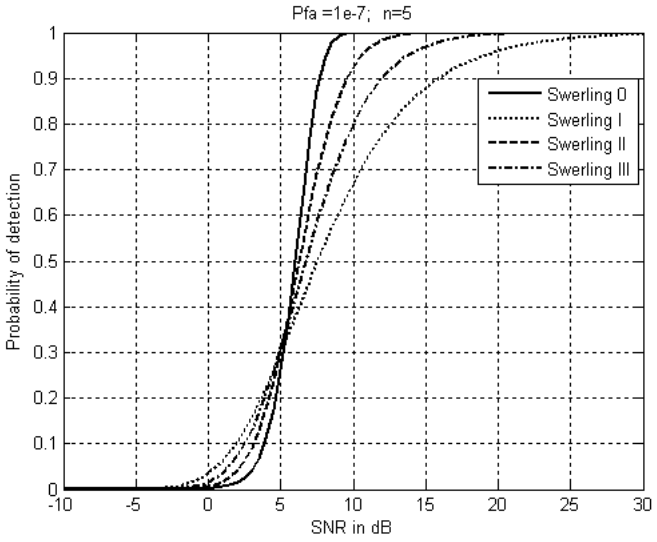


Figure 7.13b. Probability of detection versus SNR. Swerling III, Swerling II, Swerling I, and Swerling 0.

7.7.5. Detection of Swerling IV Targets

The expression for the probability of detection for Swerling IV targets for $n_P < 50$ is

$$P_D = 1 - \left[\gamma_0 + \left(\frac{SNR}{2}\right) n_P \gamma_1 + \left(\frac{SNR}{2}\right)^2 \frac{n_P(n_P-1)}{2!}\gamma_2 + \ldots + \left(\frac{SNR}{2}\right)^{n_P} \gamma_{n_P}\right]\left(1+\frac{SNR}{2}\right)^{-n_P} \tag{7.88}$$

$$\gamma_i = \Gamma_I\left(\frac{v_T}{1+(SNR)/2}\ , n_P + i\right) \tag{7.89}$$

By using the recursive formula

$$\Gamma_I(x, i+1) = \Gamma_I(x, i) - \frac{x^i}{i!\exp(x)} \tag{7.90}$$

then only γ_0 needs to be calculated using Eq. (7.89) and the rest of γ_i are calculated from the following recursion:

$$\gamma_i = \gamma_{i-1} - A_i \qquad ;\ i > 0 \tag{7.91}$$

$$A_i = \frac{v_T/(1+(SNR)/2)}{n_P + i - 1}\ A_{i-1} \qquad ;\ i > 1 \tag{7.92}$$

$$A_1 = \frac{(v_T/(1+(SNR)/2))^{n_P}}{n_P!\exp(v_T/(1+(SNR)/2))} \tag{7.93}$$

$$\gamma_0 = \Gamma_I\left(\frac{v_T}{(1+(SNR)/2)}\ , n_P\right) \tag{7.94}$$

For the case when $n_P \geq 50$, the Gram-Charlier series can be used to calculate the probability of detection. In this case,

$$C_3 = \frac{1}{3\sqrt{n_P}}\frac{2\beta^3 - 1}{(2\beta^2-1)^{1.5}} \qquad ;\ C_6 = \frac{C_3^2}{2} \tag{7.95}$$

$$C_4 = \frac{1}{4n_P}\frac{2\beta^4 - 1}{(2\beta^2-1)^2} \tag{7.96}$$

$$\varpi = \sqrt{n_P(2\beta^2 - 1)} \tag{7.97}$$

$$\beta = 1 + (SNR)/2 \tag{7.98}$$

Figure 7.14 shows plots of the probability of detection as a function of SNR for $n_P = 1, 10, 25, 75$, where $P_{fa} = 10^{-6}$. This figure can be reproduced using the following MATLAB code.

```
clear all; close all;
pfa = 1e-6;
nfa = log(2) / pfa;
index = 0;
for snr = -7:.15:10
  index = index +1;
  prob1(index) =  pd_swerling4 (nfa, 5, snr);
  prob10(index) =  pd_swerling4 (nfa, 10, snr);
  prob25(index) =  pd_swerling4(nfa, 25, snr);
  prob75(index) =  pd_swerling4 (nfa, 75, snr);
end
x = -7:.15:10;
plot(x, prob1,'k',x,prob10,'k.',x,prob25,'k:',x, prob75,'k-.','linewidth',1);
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 5','np = 10','np = 25','np = 75')
grid; axis tight
```

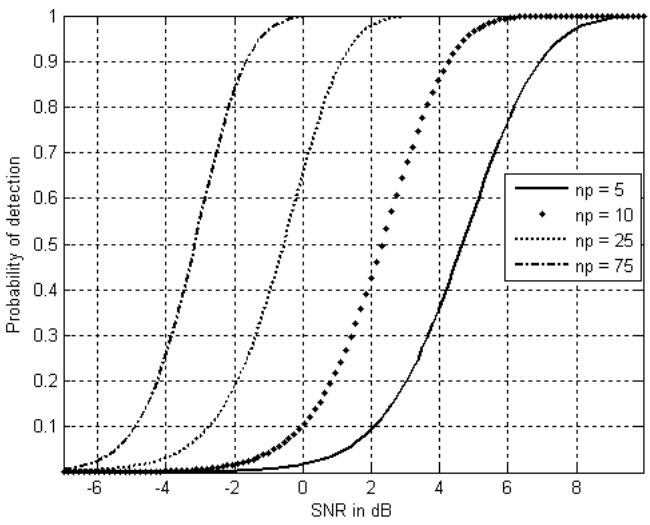


Figure 7.14. Probability of detection versus SNR. Swerling IV. $P_{fa} = 10^{-6}$.

7.8. Computation of the Fluctuation Loss

The fluctuation loss, L_f, can be viewed as the amount of additional SNR required to compensate for the SNR loss due to target fluctuation, given a specific P_D value. Kanter[1] developed an exact analysis for calculating the fluctuation loss. In this text the author will take advantage of the computational power of MATLAB and the MATLAB functions developed for this text to numerically calculate the amount of fluctuation loss. For this purpose consider the MATALB function "*fluct.m*", where its syntax is as follows:

[SNR] = fluct(pd, pfa, np, sw_case)

where

Symbol	Description	Units	Status
pd	*desired probability of detection*	*none*	*input*
nfa	*desired number of false alarms*	*none*	*input*
np	*number of pulses*	*none*	*input*
sw_case	*0, 1, 2, 3, or 4 depending on the desired Swerling case*	*none*	*input*
SNR	*Resulting SNR*	*dB*	*output*

For example, using the syntax

[SNR0] = fluct(0.8, 1e6, 5, 0)

will calculate the *SNR0* corresponding to a Swerling 0. If one would use this SNR in the function "*pd_swerling5.m*" with following syntax

[pd] = pd_swerling5 (1e6, 1, 5, SNR0)

the resulting P_D will be equal to 0.8. Similarly, if the following syntax is used

[SNR1] = fluct(.8, 1-e-6, 5, 1)

then the value *SNR1* will be that of Swerling 1. Of course, if one would use this *SNR1* value in the function "*pd_swerling1.m*" with following syntax

[pd] = pd_swerling1(1e6, 5, .8, SNR1)

the same P_D of 0.8 will be calculated. Therefore, the fluctuation loss for this case, is equal to *SNR0 - SNR1*.

1. Kanter, I., Exact Detection Probability for Partially Correlated Rayleigh Targets, *IEEE Trans*, AES-22, pp. 184-196, March 1986.

7.9. Cumulative Probability of Detection

Denote the range at which the single pulse SNR is unity (0 dB) as R_0, and refer to it as the reference range. Then, for a specific radar, the single pulse SNR at R_0 is defined by the radar equation and is given by

$$(SNR)_{R_0} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_0 B F L R_0^4} = 1 \tag{7.99}$$

The single pulse SNR at any range R is

$$SNR = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_0 B F L R^4} \tag{7.100}$$

Dividing Eq. (7.100) by Eq. (7.99) yields

$$\frac{SNR}{(SNR)_{R_0}} = \left(\frac{R_0}{R}\right)^4 \tag{7.101}$$

Therefore, if the range R_0 is known, then the SNR at any other range R is

$$(SNR)_{dB} = 40\log\left(\frac{R_0}{R}\right) \tag{7.102}$$

Also, define the range R_{50} as the range at which $P_D = 0.5 = P_{50}$. Normally, the radar unambiguous range R_u is set equal to $2R_{50}$.

The cumulative probability of detection refers to detecting the target at least once by the time it is at range R. More precisely, consider a target closing on a scanning radar, where the target is illuminated only during a scan (frame). As the target gets closer to the radar, its probability of detection increases since the SNR is increased. Suppose that the probability of detection during the nth frame is P_{D_n}; then, the cumulative probability of detecting the target at least once during the nth frame (see Fig. 7.15) is given by

$$P_{C_n} = 1 - \prod_{i=1}^{n} (1 - P_{D_i}) \tag{7.103}$$

P_{D_1} is usually selected to be very small. Clearly, the probability of not detecting the target during the nth frame is $1 - P_{C_n}$. The probability of detection for the ith frame, P_{D_i}, is computed as discussed in the previous section.

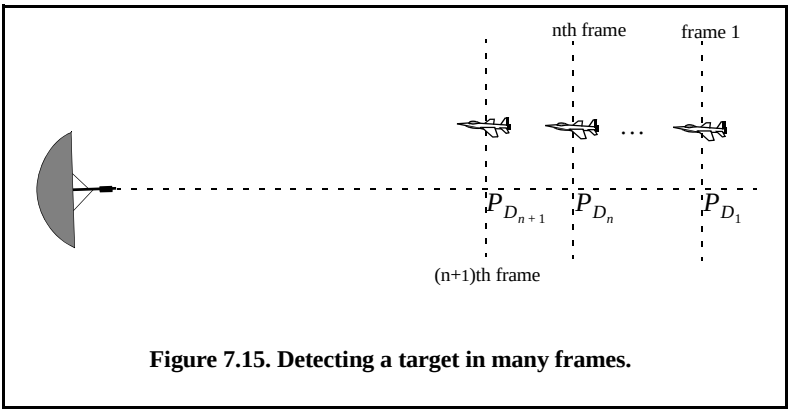


Figure 7.15. Detecting a target in many frames.

Example:

A radar detects a closing target at $R = 10Km$*, with probability of detection* P_D *equal to* 0.5*. Assume* $P_{fa} = 10^{-7}$*. Compute and sketch the single look probability of detection as a function of normalized range (with respect to* $R = 10Km$*), over the interval* $(2-20)Km$*. If the range between two successive frames is* $1Km$*, what is the cumulative probability of detection at* $R = 8Km$*?*

Solution:

From the function "marcumsq.m" the SNR corresponding to $P_D = 0.5$ *and* $P_{fa} = 10^{-7}$ *is approximately 12dB. By using a similar analysis to that which led to Eq. (7.102), we can express the SNR at any range* R *as*

$$(SNR)_R = (SNR)_{10} + 40 \log\frac{10}{R} = 52 - 40 \log R$$

By using the function "marcumsq.m" we can construct the following table:

R *Km*	(SNR) dB	P_D
2	39.09	0.999
4	27.9	0.999
6	20.9	0.999
8	15.9	0.999
9	13.8	0.9
10	12.0	0.5
11	10.3	0.25

R *Km*	(SNR) dB	P_D
12	8.8	0.07
14	6.1	0.01
16	3.8	ε
20	0.01	ε

where ε *is very small. A sketch of* P_D *versus normalized range is shown in Fig. 7.16.*

The cumulative probability of detection is given in Eq. (7.104), where the probability of detection of the first frame is selected to be very small. Thus, we can arbitrarily choose frame 1 to be at $R = 16Km$. *Note that selecting a different starting point for frame 1 would have a negligible effect on the cumulative probability (we only need* P_{D_1} *to be very small). Below is a range listing for frames 1 through 9, where frame 9 corresponds to* $R = 8Km$.

frame	1	2	3	4	5	6	7	8	9
range in Km	16	15	14	13	12	11	10	9	8

The cumulative probability of detection at 8 Km is then

$$P_{C_9} = 1-(1-0.999)(1-0.9)(1-0.5)(1-0.25)(1-0.07)$$
$$(1-0.01)(1-\varepsilon)^2 \approx 0.9998$$

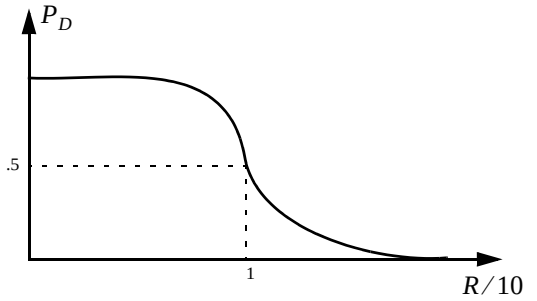


Figure 7.16. Cumulative probability of detection versus normalized range.

7.10. Constant False Alarm Rate (CFAR)

The detection threshold is computed so that the radar receiver maintains a constant predetermined probability of false alarm. Equation (7.20) gives the relationship between the threshold value V_T and the probability of false alarm P_{fa}, and for convenience is repeated here as Eq. (7.104):

$$v_T = \sqrt{2\sigma^2 \ln\left(\frac{1}{P_{fa}}\right)} \tag{7.104}$$

If the noise power σ^2 is constant, then a fixed threshold can satisfy Eq. (7.104). However, due to many reasons this condition is rarely true. Thus, in order to maintain a constant probability of false alarm, the threshold value must be continuously updated based on the estimates of the noise variance. The process of continuously changing the threshold value to maintain a constant probability of false alarm is known as Constant False Alarm Rate (CFAR).

Three different types of CFAR processors are primarily used. They are adaptive threshold CFAR, nonparametric CFAR, and nonlinear receiver techniques. Adaptive CFAR assumes that the interference distribution is known and approximates the unknown parameters associated with these distributions. Nonparametric CFAR processors tend to accommodate unknown interference distributions. Nonlinear receiver techniques attempt to normalize the root-mean-square amplitude of the interference. In this book only analog Cell-Averaging CFAR (CA-CFAR) technique is examined. The analysis presented in this section closely follows Urkowitz[1].

7.10.1. Cell-Averaging CFAR (Single Pulse)

The CA-CFAR processor is shown in Fig. 7.17. Cell averaging is performed on a series of range and/or Doppler bins (cells). The echo return for each pulse is detected by a square-law detector. In analog implementation these cells are obtained from a tapped delay line. The Cell Under Test (CUT) is the central cell. The immediate neighbors of the CUT are excluded from the averaging process due to a possible spillover from the CUT. The output of M reference cells ($M/2$ on each side of the CUT) is averaged. The threshold value is obtained by multiplying the averaged estimate from all reference cells by a constant K_0 (used for scaling). A detection is declared in the CUT if

$$Y_1 \geq K_0 Z \tag{7.105}$$

1. Urkowitz, H., Decision and Detection Theory, unpublished lecture notes. Lockheed Martin Co., Moorestown, NJ.

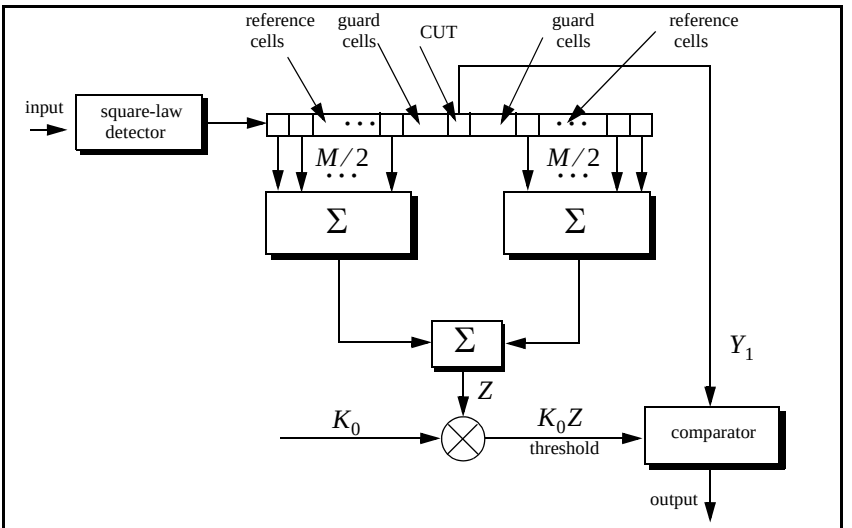


Figure 7.17. Conventional CA-CFAR.

CA-CFAR assumes that the target of interest is in the CUT and all reference cells contain zero-mean independent Gaussian noise of variance ψ^2. Therefore, the output of the reference cells, Z, represents a random variable with gamma probability density function (special case of the chi-square) with $2M$ degrees of freedom. In this case, the gamma *pdf* is

$$f(z) = \frac{z^{(M/2)-1} e^{(-z/2\psi^2)}}{2^{M/2}\ \sigma^M \Gamma(M/2)} \qquad ;\ z > 0 \tag{7.106}$$

The probability of false alarm corresponding to a fixed threshold was derived earlier. When CA-CFAR is implemented, then the probability of false alarm can be derived from the conditional false alarm probability, which is averaged over all possible values of the threshold in order to achieve an unconditional false alarm probability. The conditional probability of false alarm when $y = V_T$ can be written as

$$P_{fa}(v_T = y) = e^{-y/2\sigma^2} \tag{7.107}$$

It follows that the unconditional probability of false alarm is

$$P_{fa} = \int_0^\infty P_{fa}(v_T = y) f(y) dy \tag{7.108}$$

where $f(y)$ is the *pdf* of the threshold, which except for the constant K_0 is the same as that defined in Eq. (7.106). Therefore,

$$f(y) = \frac{y^{M-1} e^{(-y/2K_0\psi^2)}}{(2K_0\sigma^2)^M \Gamma(M)} \quad ; \; y \geq 0 \tag{7.109}$$

Performing the integration in Eq. (7.108) yields

$$P_{fa} = 1/(1+K_0)^M \tag{7.110}$$

Observation of Eq. (7.110) shows that the probability of false alarm is now independent of the noise power, which is the objective of CFAR processing.

7.10.2. Cell-Averaging CFAR with Noncoherent Integration

In practice, CFAR averaging is often implemented after noncoherent integration, as illustrated in Fig. 7.18. Now, the output of each reference cell is the sum of n_P squared envelopes. It follows that the total number of summed reference samples is Mn_P. The output Y_1 is also the sum of n_P squared envelopes. When noise alone is present in the CUT, Y_1 is a random variable whose *pdf* is a gamma distribution with $2n_p$ degrees of freedom. Additionally, the summed output of the reference cells is the sum of Mn_P squared envelopes. Thus, Z is also a random variable which has a gamma *pdf* with $2Mn_P$ degrees of freedom.

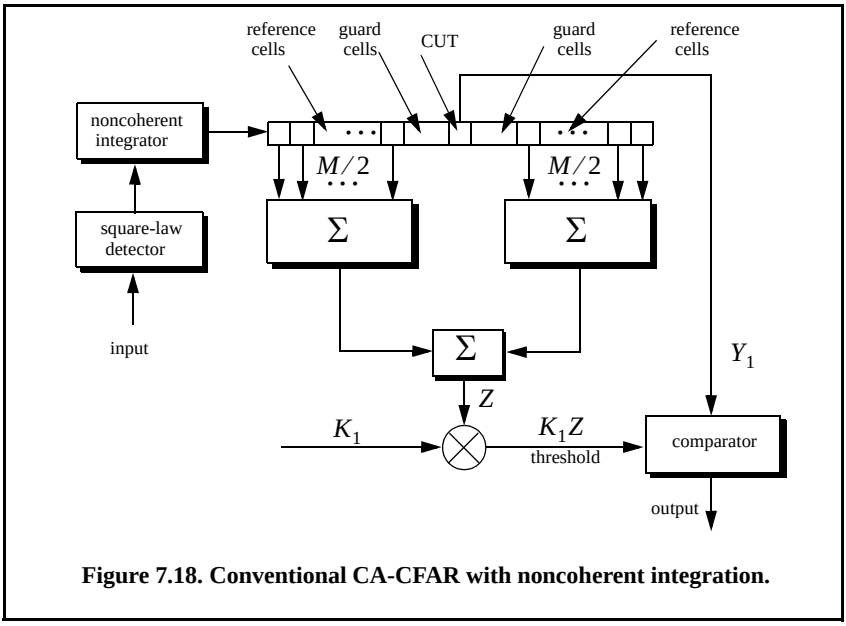


Figure 7.18. Conventional CA-CFAR with noncoherent integration.

The probability of false alarm is then equal to the probability that the ratio Y_1/Z exceeds the threshold. More precisely,

$$P_{fa} = Prob\{Y_1/Z > K_1\} \tag{7.111}$$

Equation (7.111) implies that one must first find the joint *pdf* for the ratio Y_1/Z. However, this can be avoided if P_{fa} is first computed for a fixed threshold value V_T, then averaged over all possible values of the threshold. Therefore, let the conditional probability of false alarm when $y = v_T$ be $P_{fa}(v_T = y)$. It follows that the unconditional false alarm probability is

$$P_{fa} = \int_0^\infty P_{fa}(v_T = y) f(y) dy \tag{7.112}$$

where $f(y)$ is the *pdf* of the threshold. In view of this, the probability density function describing the random variable $K_1 Z$ is given by

$$f(y) = \frac{(y/K_1)^{Mn_P - 1} e^{(-y/2K_0\sigma^2)}}{(2\sigma^2)^{Mn_P} K_1 \ \Gamma(Mn_P)} \qquad ; \ y \geq 0 \tag{7.113}$$

It can be shown that in this case the probability of false alarm is independent of the noise power and is given by

$$P_{fa} = \frac{1}{(1 + K_1)^{Mn_P}} \sum_{k=0}^{n_P - 1} \frac{1}{k!} \frac{\Gamma(Mn_P + k)}{\Gamma(Mn_P)} \left(\frac{K_1}{1 + K_1}\right)^k \tag{7.114}$$

which is identical to Eq. (7.110) when $K_1 = K_0$ and $n_P = 1$.

7.11. MATLAB Programs and Routines

This section presents listings for all the MATLAB programs used to produce all of the MATLAB-generated figures in this chapter. Additionally, other specific MATLAB functions are also presented. They are listed in the same order they appear in the chapter.

7.11.1. MATLAB Function "que_func.m"

The function "*que_func.m*" computes $F(x)$ using Eqs. (7.17) and (7.18). The syntax is as follows:

fofx = que_func (x)

MATLAB Function** "que_func.m" **Listing

```
function fofx = que_func(x)
% This function computes the value of the Q-function
% It uses the approximation in Eqs. (7.17) and (7.18)
if (x >= 0)
   denom = 0.661 * x + 0.339 * sqrt(x^2 + 5.51);
  expo = exp(-x^2 /2.0);
  fofx = 1.0 - (1.0 / sqrt(2.0 * pi)) * (1.0 / denom) * expo;
else
  denom = 0.661 * x + 0.339 * sqrt(x^2 + 5.51);
  expo = exp(-x^2 /2.0);
  value = 1.0 - (1.0 / sqrt(2.0 * pi)) * (1.0 / denom) * expo;
  fofx = 1.0 - value;
end
```

7.11.2. MATLAB Function "marcumsq.m"

This function utilizes Parl's method to compute P_D. The syntax is as follows:

$$Pd = marcumsq(a,b)$$

MATLAB Function** "marcumsq.m" **Listing

```
function Pd = marcumsq (a,b); % This function uses Parl's method to compute PD
max_test_value = 5000.;
if (a < b)
  alphan0 = 1.0;
  dn = a / b;
else
  alphan0 = 0.;
  dn = b / a;
end
alphan_1 = 0.;
betan0 = 0.5;
betan_1 = 0.;
D1 = dn;
n = 0;
ratio = 2.0 / (a * b);
r1 = 0.0;
betan = 0.0;
alphan = 0.0;
while betan < 1000.,
  n = n + 1;
  alphan = dn + ratio * n * alphan0 + alphan;
  betan = 1.0 + ratio * n * betan0 + betan;
  alphan_1 = alphan0;
  alphan0 = alphan;
  betan_1 = betan0;
```

```
  betan0 = betan;
  dn = dn * D1;
end
PD = (alphan0 / (2.0 * betan0)) * exp( -(a-b)^2 / 2.0);
if ( a >= b)
  PD = 1.0 - PD;
end
return
```

7.11.3. MATLAB Function "improv_fac.m"

The function "*improv_fac.m*" calculates the improvement factor using Eq. (7.45). The syntax is as follows:

[impr_of_np] = improv_fac (np, pfa, pd)

where

Symbol	Description	Units	Status
np	*number of integrated pulses*	*none*	*input*
pfa	*probability of false alarms*	*none*	*input*
pd	*probability of detection*	*none*	*input*
impr_of_np	*improvement factor*	*output*	*dB*

MATLAB Function** "improv_fac.m" **Listing

```
function impr_of_np = improv_fac (np, pfa, pd)
% This function computes the noncoherent integration improvement
% factor using the empirical formula defined in Eq. (7.54)
fact1 = 1.0 + log10( 1.0 / pfa) / 46.6;
fact2 = 6.79 * (1.0 + 0.235 * pd);
fact3 = 1.0 - 0.14 * log10(np) + 0.0183 * (log10(np))^2;
impr_of_np = fact1 * fact2 * fact3 * log10(np);
return
```

7.11.4. MATLAB Function "threshold.m"

The function "*threshold.m*" calculates the threshold value given the algorithm described in Section 7.6. The syntax is as follows:

[pfa, vt] = threshold (nfa, np)

where

Symbol	Description	Units	Status
nfa	*number of false alarm*	*none*	*input*
np	*number of pulses*	*none*	*input*
pfa	*probability of alarm*	*none*	*output*
vt	*threshold value*	*none*	*output*

MATLAB Function** "threshold.m" **Listing

```
function [pfa, vt] = threshold (nfa, np)
% This function calculates the threshold value from nfa and np.
% The Newton-Raphson  recursive formula is used
% This function uses "gammainc.m".
delmax = .00001;
eps = 0.000000001;
delta =10000.;
pfa = np * log(2) / nfa;
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (abs(delta) >= vt0)
  igf = gammainc(vt0,np);
  num = 0.5^(np/nfa) - igf;
  temp = (np-1) * log(vt0+eps) - vt0 - factor(np-1);
  deno = exp(temp);
  vt = vt0 + (num / (deno+eps));
  delta = abs(vt - vt0) * 10000.0;
  vt0 = vt;
```

7.11.5. MATLAB Function "pd_swerling5.m"

The function "*pd_swerling5.m*" calculates the probability of detection for Swerling 0 targets. The syntax is as follows:

[pd] = pd_swerling5 (input1, indicator, np, snr)

where

Symbol	Description	Units	Status
input1	P_{fa} *or* n_{fa}	*none*	*input*
indicator	*1 when input1 =* P_{fa} *2 when input1 =* n_{fa}	*none*	*input*

Symbol	Description	Units	Status
np	*number of integrated pulses*	*none*	*input*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

MATLAB Function** "pd_swerling5.m" **Listing

```
function pd = pd_swerling5 (input1, indicator, np, snrbar)
% This function is used to calculate the probability of detection
% for Swerling 5 or 0 targets for np>1.
if(np == 1)
  'Stop, np must be greater than 1'
  return
end
format long
snrbar = 10.0.^(snrbar./10.);
eps = 0.00000001;
delmax = .00001;
delta =10000.;
% Calculate the threshold Vt
if (indicator ~=1)
  nfa = input1;
  pfa =  np * log(2) / nfa;
else
  pfa = input1;
  nfa = np * log(2) / pfa;
end
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (abs(delta) >= vt0)
  igf = incomplete_gamma(vt0,np);
  num = 0.5^(np/nfa) - igf;
  temp = (np-1) * log(vt0+eps) - vt0 - factor(np-1);
  deno = exp(temp);
  vt = vt0 + (num / (deno+eps));
  delta = abs(vt - vt0) * 10000.0;
  vt0 = vt;
end
% Calculate the Gram-Chrlier coefficients
temp1 = 2.0 .* snrbar + 1.0;
omegabar = sqrt(np .* temp1);
c3 = -(snrbar + 1.0 / 3.0) ./ (sqrt(np) .* temp1.^1.5);
c4 = (snrbar + 0.25) ./ (np .* temp1.^2.);
c6 = c3 .* c3 ./2.0;
```

```
V = (vt - np .* (1.0 + snrbar)) ./ omegabar;
Vsqr = V .*V;
val1 = exp(-Vsqr ./ 2.0) ./ sqrt( 2.0 * pi);
val2 = c3 .* (V.^2 -1.0) + c4 .* V .* (3.0 - V.^2) -...
  c6 .* V .* (V.^4 - 10. .* V.^2 + 15.0);
q = 0.5 .* erfc (V./sqrt(2.0));
pd =  q - val1 .* val2;
return
```

7.11.6. MATLAB Function "pd_swerling1.m"

The function "*pd_swerling1.m*" calculates the probability of detection for Swerling I type targets. The syntax is as follows:

[pd] = pd_swerling1 (nfa, np, snr)

where

Symbol	Description	Units	Status
nfa	*Marcum's false alarm number*	*none*	*input*
np	*number of integrated pulses*	*none*	*input*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

MATLAB Function** "pd_swerling1.m" **Listing

```
function [pd] = pd_swerling1 (nfa, np, snrbar)
% This function is used to calculate the probability of detection
% for Swerling 1 targets.
format long
snrbar = 10.0^(snrbar/10.);
eps = 0.00000001;
delmax = .00001;
delta =10000.;
% Calculate the threshold Vt
pfa =  np * log(2) / nfa;
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (delta < (vt0/10000));
  igf = gammainc(vt0,np);
  num = 0.5^(np/nfa) - igf;
  deno = -exp(-vt0) * vt0^(np-1) /factorial(np-1);
  vt = vt0 - (num / (deno+eps));
  delta = abs(vt - vt0);
  vt0 = vt;
```

```
end
if (np == 1)
  temp = -vt / (1.0 + snrbar);
  pd = exp(temp);
  return
end
  temp1 = 1.0 + np * snrbar;
  temp2 = 1.0 / (np *snrbar);
  temp = 1.0 + temp2;
  val1 = temp^(np-1.);
  igf1 = gammainc(vt,np-1);
  igf2 = gammainc(vt/temp,np-1);
  pd = 1.0 - igf1 + val1 * igf2 * exp(-vt/temp1);
  return
```

7.11.7. MATLAB Function "pd_swerling2.m"

The function *"pd_swerling2.m"* calculates P_D for Swerling II type targets. The syntax is as follows:

[pd] = pd_swerling2 (nfa, np, snr)

where

Symbol	Description	Units	Status
nfa	*Marcum's false alarm number*	*none*	*input*
np	*number of integrated pulses*	*none*	*input*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

MATLAB Function** "pd_swerling2.m" **Listing

```
function [pd] = pd_swerling2 (nfa, np, snrbar)
% This function is used to calculate the probability of detection
% for Swerling 2 targets.
format long
snrbar = 10.0^(snrbar/10.);
eps = 0.00000001;
delmax = .00001;
delta =10000.;
% Calculate the threshold Vt
pfa =  np * log(2) / nfa;
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (delta < (vt0/10000));
```

```
  igf = gammainc(vt0,np);
  num = 0.5^(np/nfa) - igf;
  deno = -exp(-vt0) * vt0^(np-1) /factorial(np-1);
  vt = vt0 - (num / (deno+eps));
  delta = abs(vt - vt0);
  vt0 = vt;
end
if (np <= 50)
  temp = vt / (1.0 + snrbar);
  pd = 1.0 - gammainc(temp,np);
  return
else
  temp1 = snrbar + 1.0;
  omegabar = sqrt(np) * temp1;
  c3 = -1.0 / sqrt(9.0 * np);
  c4 = 0.25 / np;
  c6 = c3 * c3 /2.0;
  V = (vt - np * temp1) / omegabar;
  Vsqr = V *V;
  val1 = exp(-Vsqr / 2.0) / sqrt( 2.0 * pi);
  val2 = c3 * (V^2 -1.0) + c4 * V * (3.0 - V^2) - ...
    c6 * V * (V^4 - 10. * V^2 + 15.0);
  q = 0.5 * erfc (V/sqrt(2.0));
  pd =  q - val1 * val2;
end
return
```

7.11.8. MATLAB Function "pd_swerling3.m"

The function *"pd_swerling3.m"* calculates P_D for Swerling III type targets. The syntax is as follows:

[pd] = pd_swerling3 (nfa, np, snr)

where

Symbol	Description	Units	Status
nfa	*Marcum's false alarm number*	*none*	*input*
np	*number of integrated pulses*	*none*	*input*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

MATLAB Function** "pd_swerling3.m" **Listing

```
function [pd] = pd_swerling3 (nfa, np, snrbar)
% This function is used to calculate the probability of detection
% for Swerling 3 targets.
```

```
format long
snrbar = 10.0^(snrbar/10.);
eps = 0.00000001;
delmax = .00001;
delta =10000.;
% Calculate the threshold Vt
pfa =  np * log(2) / nfa;
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (delta < (vt0/10000));
  igf = gammainc(vt0,np);
  num = 0.5^(np/nfa) - igf;
  deno = -exp(-vt0) * vt0^(np-1) /factorial(np-1);
  vt = vt0 - (num / (deno+eps));
  delta = abs(vt - vt0);
  vt0 = vt;
end
temp1 = vt / (1.0 + 0.5 * np *snrbar);
temp2 = 1.0 + 2.0 / (np * snrbar);
temp3 = 2.0 * (np - 2.0) / (np * snrbar);
ko = exp(-temp1) * temp2^(np-2.) * (1.0 + temp1 - temp3);
if (np <= 2)
  pd = ko;
  return
else
  ko = exp(-temp1) * temp2^(np-2.) * (1.0 + temp1 - temp3);
  temp4 = vt^(np-1.) * exp(-vt) / (temp1 * (factorial(np-2.)));
  temp5 =  vt / (1.0 + 2.0 / (np *snrbar));
  pd = temp4 + 1.0 - gammainc(vt,np-1.) + ko * gammainc(temp5,np-1.);
end
return
```

7.11.9. MATLAB Function "pd_swerling4.m"

The function *"pd_swerling4.m"* calculates P_D for Swerling IV type targets. The syntax is as follows:

[pd] = pd_swerling4 (nfa, np, snr)

where

Symbol	Description	Units	Status
nfa	*Marcum's false alarm number*	*none*	*input*
np	*number of integrated pulses*	*none*	*input*

Symbol	Description	Units	Status
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

MATLAB Function** "pd_swerling4.m" **Listing

```
function [pd] = pd_swerling4 (nfa, np, snrbar)
% This function is used to calculate the probability of detection
% for Swerling 4 targets.
format long
snrbar = 10.0^(snrbar/10.);
eps = 0.00000001;
delmax = .00001;
delta =10000.;
% Calculate the threshold Vt
pfa =  np * log(2) / nfa;
sqrtpfa = sqrt(-log10(pfa));
sqrtnp = sqrt(np);
vt0 = np - sqrtnp + 2.3 * sqrtpfa * (sqrtpfa + sqrtnp - 1.0);
vt = vt0;
while (delta < (vt0/10000));
  igf = gammainc(vt0,np);
  num = 0.5^(np/nfa) - igf;
  deno = -exp(-vt0) * vt0^(np-1) /factorial(np-1);
  vt = vt0 - (num / (deno+eps));
  delta = abs(vt - vt0);
  vt0 = vt;
end
h8 = snrbar /2.0;
beta = 1.0 + h8;
beta2 = 2.0 * beta^2 - 1.0;
beta3 = 2.0 * beta^3;
if (np >= 50)
  temp1 = 2.0 * beta -1;
  omegabar = sqrt(np * temp1);
  c3 = (beta3 - 1.) / 3.0 / beta2 / omegabar;
  c4 = (beta3 * beta3 - 1.0) / 4. / np /beta2 /beta2;
  c6 = c3 * c3 /2.0;
  V = (vt - np * (1.0 + snrbar)) / omegabar;
  Vsqr = V *V;
  val1 = exp(-Vsqr / 2.0) / sqrt( 2.0 * pi);
  val2 = c3 * (V^2 -1.0) + c4 * V * (3.0 - V^2) - c6 * V * (V^4 - 10. * V^2 + 15.0);
  q = 0.5 * erfc (V/sqrt(2.0));
  pd =  q - val1 * val2;
  return
else
```

```
  gamma0 = gammainc(vt/beta,np);
  a1 = (vt / beta)^np / (factorial(np) * exp(vt/beta));
  sum = gamma0;
  for i = 1:1:np
    temp1 = gamma0;
    if (i == 1)
      ai = a1;
    else
      ai = (vt / beta) * a1 / (np + i -1);
    end
    gammai = gamma0 - ai;
    gamma0 = gammai;
    a1 = ai;
    for ii = 1:1:i
      temp1 = temp1 * (np + 1 - ii);
    end
    term = (snrbar /2.0)^i * gammai * temp1 / (factorial(i));
    sum = sum + term;
  end
  pd = 1.0 - (sum / beta^np);
end
pd = max(pd,0.);
return
```

7.11.10. MATLAB Function "fluct_loss.m"

This functions has been described in Section 7.8.

***MATLAB Function** "fluct_loss.m"*

```
function [SNR] = fluct(pd, nfa, np, sw_case)
% This function calculates the SNR fluctuation loss for Swerling models
% A negative Lf value indicates SNR gain instead of loss
format long
% ********** Swerling 5 case **********
% check to make sure that np>1
pfa =  np * log(2) / nfa;
if (sw_case == 0)
if (np ==1)
   nfa = 1/pfa;
   b = sqrt(-2.0 * log(pfa));
   Pd_Sw5 = 0.001;
   snr_inc = 0.1 - 0.005;
   while(Pd_Sw5 <= pd)
      snr_inc = snr_inc + 0.005;
      a = sqrt(2.0 * 10^(.1*snr_inc));
      Pd_Sw5 = marcumsq(a,b);
   end
```

```
   PD_SW5 = Pd_Sw5;
   SNR = snr_inc;
else
   % np > 1 use MATLAB function pd_swerling5.m
   snr_inc = 0.1 - 0.001;
   Pd_Sw5 = 0.001;
   while(Pd_Sw5 <= pd)
      snr_inc = snr_inc + 0.001;
      Pd_Sw5 = pd_swerling5(pfa, 1, np, snr_inc);
   end
   PD_SW5 = Pd_Sw5;
   SNR = snr_inc;
end
end
 % ********** End Swerling 5 case **********
% ********** Swerling 1 case **********
% compute the false alarm number
if (sw_case==1)
   Pd_Sw1 = 0.001;
   snr_inc = 0.1 - 0.001;
   while(Pd_Sw1 <= pd)
      snr_inc = snr_inc + 0.001;
      Pd_Sw1 = pd_swerling1(nfa, np, snr_inc);
   end
   PD_SW1 = Pd_Sw1;
   SNR = snr_inc;
end
 % ********** End Swerling 1 case **********
% ********** Swerling 2 case **********
if (sw_case == 2)
   Pd_Sw2 = 0.001;
   snr_inc = 0.1 - 0.001;
   while(Pd_Sw2 <= pd)
      snr_inc = snr_inc + 0.001;
      Pd_Sw2 = pd_swerling2(nfa, np, snr_inc);
   end
   PD_SW2 = Pd_Sw2;
   SNR = snr_inc;
end
 % ********** End Swerling 2 case **********
% ********** Swerling 3 case **********
if (sw_case == 3)
   Pd_Sw3 = 0.001;
   snr_inc = 0.1 - 0.001;
   while(Pd_Sw3 <= pd)
      snr_inc = snr_inc + 0.001;
      Pd_Sw3 = pd_swerling3(nfa, np, snr_inc);
   end
```

```
   PD_SW3 = Pd_Sw3;
   SNR = snr_inc;
end
 % ********** End Swerling 3 case **********
% ********** Swerling 4 case **********
if (sw_case == 4)
   Pd_Sw4 = 0.001;
   snr_inc = 0.1 - 0.001;
   while(Pd_Sw4 <= pd)
      snr_inc = snr_inc + 0.001;
      Pd_Sw4 = pd_swerling4(nfa, np, snr_inc);
   end
   PD_SW4 = Pd_Sw4;
   SNR = snr_inc;
end
```

Appendix 7.A The Incomplete Gamma Function

The Gamma Function

Define the Gamma function (not the incomplete Gamma function) of the variable z (generally complex) as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx \tag{7.115}$$

and when z is a positive integer, then

$$\Gamma(z) = (z-1)! \tag{7.116}$$

One very useful and frequently used property is

$$\Gamma(z+1) = z\Gamma(z) \tag{7.117}$$

The Incomplete Gamma Function

The incomplete gamma function. $\Gamma_I(u, q)$ used in this text is given by

$$\Gamma_I(u, q) = \int_0^{u\sqrt{q+1}} \frac{e^{-x} x^q}{q!} dx \tag{7.118}$$

Another definition, which is often used in the literature, for the incomplete Gamma function is

$$\Gamma_I[z, q] = \int_q^{\infty} x^{z-1} e^{-x} dx \tag{7.119}$$

It follows that

$$\Gamma(z) = \Gamma_I[z, 0] = \int_0^{\infty} x^{z-1} e^{-x} dx \tag{7.120}$$

which is the same as Eq. (7.115). Furthermore, for a positive integer n, the incomplete Gamma function can be represented by

$$\Gamma_I[n, z] = (n-1)! e^{-z} \sum_{k=0}^{n-1} \frac{z^k}{k!} \tag{7.121}$$

In order to relate $\Gamma_I[n, z]$ and $\Gamma_I(u, q)$ compute the following relation

$$\Gamma_I[a, 0] - \Gamma_I[a, z] = \int_0^{\infty} x^{a-1} e^{-x} dx - \int_z^{\infty} x^{a-1} e^{-x} dx = \int_0^{z} x^{a-1} e^{-x} dx \tag{7.122}$$

Applying the change of variables $a = q + 1$ and $z = u\sqrt{q+1}$ yields

$$\Gamma_I[q+1, 0] - \Gamma_I[q+1, u\sqrt{q+1}] = \int_0^{u\sqrt{q+1}} x^q e^{-x} dx \tag{7.123}$$

and if q is a positive integer then

$$\frac{\Gamma_I[q+1, 0] - \Gamma_I[q+1, u\sqrt{q+1}]}{q!} = \int_0^{u\sqrt{q+1}} \frac{x^q e^{-x}}{q!} dx = \Gamma_I(u, q) \tag{7.124}$$

Using Eq. (7.116) and (7.121) in Eq. (7.124) yields

$$\Gamma_I(u, q) = 1 - \frac{(q+1-1)! e^{-u\sqrt{q+1}}}{q!} \sum_{k=0}^{q} \frac{(u\sqrt{q+1})^k}{k!} \tag{7.125}$$

Finally, the incomplete Gamma function can be written as

$$\Gamma_I(u, q) = 1 - e^{-u\sqrt{q+1}} \sum_{k=0}^{q} \frac{(u\sqrt{q+1})^k}{k!} \tag{7.126}$$

The two limiting values for Eq. (7.126) are

$$\Gamma_I(0, q) = 0 \qquad \Gamma_I(\infty, q) = 1 \tag{7.127}$$

Figure 7A.1 shows the incomplete gamma function for $q = 1, 3, 5, 8$. This figure can be reproduced using the following MATLAB code which utilizes the built-in MATLAB function "*gammainc.m*".

```
% This program can be used to reproduce Fig. 7A.1
close all; clear all
x=linspace(0,20,200);
y1 = gammainc(x,1);
y2 = gammainc(x,3);
y3 = gammainc(x,5);
y4 = gammainc(x,8);
plot(x,y1,'k',x,y2,'k:',x,y3,'k--',x,y4,'k-.')
legend('q = 1','q = 3','q = 5','q = 8')
xlabel('x'); ylabel('Incomplete Gamma function (x,q)')
grid
```

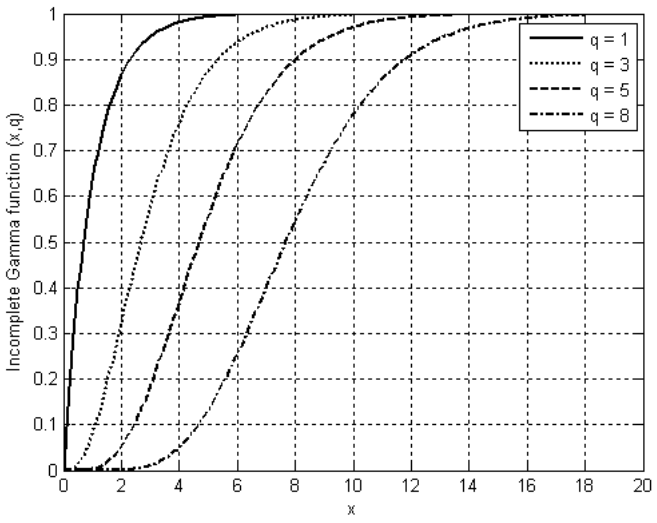


Figure 7A.1. The incomplete Gamma function for four values of q.

Problems

7.1. In the case of noise alone, the quadrature components of a radar return are independent Gaussian random variables with zero mean and variance σ^2. Assume that the radar processing consists of envelope detection followed by threshold decision. (a) Write an expression for the *pdf* of the envelope; (b) determine the threshold V_T as a function of σ that ensures a probability of false alarm $P_{fa} \leq 10^{-8}$.

7.2. (a) Derive Eq. (7.13); (b) derive Eq. (7.15).

7.3. A pulsed radar has the following specifications: time of false alarm $T_{fa} = 10\ min$, probability of detection $P_D = 0.95$, operating bandwidth $B = 1MHz$. (a) What is the probability of false alarm P_{fa}? (b) What is the single pulse SNR? (c) Assuming noncoherent integration of 100 pulses, what is the SNR reduction so that P_D and P_{fa} remain unchanged?

7.4. An L-band radar has the following specifications: operating frequency $f_0 = 1.5GHz$, operating bandwidth $B = 2MHz$, noise figure $F = 8dB$, system losses $L = 4dB$, time of false alarm $T_{fa} = 12\ minutes$, detection range $R = 12Km$, probability of detection $P_D = 0.5$, antenna gain $G = 5000$, and target RCS $\sigma = 1m^2$. (a) Determine the PRF f_r, the pulse width τ, the peak power P_t, the probability of false alarm P_{fa}, and the minimum detectable signal level S_{min}. (b) How can you reduce the transmitter power to achieve the same performance when 10 pulses are integrated noncoherently? (c) If the radar operates at a shorter range in the single pulse mode, find the new probability of detection when the range decreases to $9Km$.

7.5. (a) Show how you can use the radar equation to determine the PRF f_r, the pulse width τ, the peak power P_t, the probability of false alarm P_{fa}, and the minimum detectable signal level S_{min}. Assume the following specifications: operating frequency $f_0 = 1.5MHz$, operating bandwidth $B = 1MHz$, noise figure $F = 10dB$, system losses $L = 5dB$, time of false alarm $T_{fa} = 20\ min$, detection range $R = 12Km$, probability of detection $P_D = 0.5$ (three pulses). (b) If post detection integration is assumed, determine the SNR.

7.6. Show that when computing the probability of detection at the output of an envelope detector, it is possible to use Gaussian probability approximation when the SNR is very large.

7.7. A radar system uses a threshold detection criterion. The probability of false alarm $P_{fa} = 10^{-10}$. (a) What must be the average SNR at the input of a linear detector so that the probability of miss is $P_m = 0.15$? Assume large SNR approximation. (b) Write an expression for the *pdf* at the output of the envelope detector.

7.8. An X-band radar has the following specifications: received peak power $10^{-10}W$, probability of detection $P_D = 0.95$, time of false alarm $T_{fa} = 8\ min$, pulse width $\tau = 2\mu s$, operating bandwidth $B = 2MHz$, operating frequency $f_0 = 10GHz$, and detection range $R = 100Km$. Assume single pulse processing. (a) Compute the probability of false alarm P_{fa}. (b) Determine the SNR at the output of the IF amplifier. (c) At what SNR would the probability of detection drop to 0.9 (P_{fa} does not change)? (d) What is the increase in range that corresponds to this drop in the probability of detection?

7.9. A certain radar utilizes 10 pulses for noncoherent integration. The single pulse SNR is $15dB$ and the probability of miss is $P_m = 0.15$. (a) Compute the probability of false alarm P_{fa}. (b) Find the threshold voltage V_T.

7.10. Consider a scanning low PRF radar. The antenna half-power beam width is $1.5°$, and the antenna scan rate is $35°$ per second. The pulse width is $\tau = 2\mu s$, and the PRF is $f_r = 400Hz$. (a) Compute the radar operating bandwidth. (b) Calculate the number of returned pulses from each target illumination. (c) Compute the SNR improvement due to post-detection integration (assume 100% efficiency). (d) Find the number of false alarms per minute for a probability of false alarm $P_{fa} = 10^{-6}$.

7.11. Using the equation

$$P_D = 1 - e^{-SNR} \int_{P_{fa}}^{1} I_0(\sqrt{-4SNR \ln u})du$$

calculate P_D when $SNR = 10dB$ and $P_{fa} = 0.01$. Perform the integration numerically.

7.12. Write a MATLAB program to compute the CA-CFAR threshold value. Use similar approach to that used in the case of a fixed threshold.

7.13. A certain radar has the following specifications: single pulse SNR corresponding to a reference range $R_0 = 200Km$ is $10dB$. The probability of detection at this range is $P_D = 0.95$. Assume a Swerling I type target. Use the radar equation to compute the required pulse widths at ranges $R = 220Km, 250Km, 175Km$, so that the probability of detection is maintained.

7.14. Repeat Problem 7.14 for Swerling IV type target.

7.15. Utilizing the MATLAB functions presented in this chapter, plot the actual value for the improvement factor versus the number of integrated pulses. Pick three different values for the probability of false alarm.

7.16. Develop a MATLAB program to calculate the cumulative probability of detection.

7.17. A certain radar has the following parameters: Peak power $P_t = 500KW$, total losses $L = 12dB$, operating frequency $f_o = 5.6GHZ$, PRF $f_r = 2KHz$, pulse width $\tau = 0.5\mu s$, antenna beamwidth $\theta_{az} = 2°$ and $\theta_{el} = 7°$, noise figure $F = 6dB$, scan time $T_{sc} = 2s$. The radar can experience one false alarm per scan. (a) What is the probability of false alarm? Assume that the radar searches a minimum range of 10 Km to its maximum unambiguous range. (b) Plot the detection range versus RCS in dBsm. The detection range is defined as the range at which the single scan probability of detection is equal to 0.94. Generate curves for Swerling I, II, III, and IV type targets. (c) Repeat part (b) above when noncoherent integration is used.

7.18. A certain circularly scanning radar with a fan beam has a rotation rate of 3 seconds per revolution. The azimuth beamwidth is 3 degrees and the radar uses a PRI of 600 microseconds. The radar pulse width is 2 microseconds and the radar searches a range window that extends from 15 Km to 100 Km. It is desired that the false alarm rate not be higher than two false alarms per revolution. What is the required probability of false alarm? What is the minimum SNR so that minimum probability of false alarm can be maintained?

7.19. Derive Eq(7.63).

Chapter 8 Pulse Compression

Range resolution for a given radar can be significantly improved by using very short pulses. Unfortunately, utilizing short pulses decreases the average transmitted power, hence reducing the SNR. Since the average transmitted power is directly linked to the receiver SNR, it is often desirable to increase the pulse width (i.e., the average transmitted power) while simultaneously maintaining adequate range resolution. This can be made possible by using pulse compression techniques and the matched filter receiver. Pulse compression allows us to achieve the average transmitted power of a relatively long pulse, while obtaining the range resolution corresponding to a short pulse. In this chapter, two pulse compression techniques are discussed. The first technique is known as correlation processing which is predominantly used for narrow band and some medium band radar operations. The second technique is called stretch processing and is normally used for extremely wide band radar operations.

8.1. Time-Bandwidth Product

Consider a radar system that employs a matched filter receiver. Let the matched filter receiver bandwidth be denoted as B. Then the noise power available within the matched filter bandwidth is given by

$$N_i = 2 \frac{\eta_0}{2} B \tag{8.1}$$

where the factor of two is used to account for both negative and positive frequency bands, as illustrated in Fig. 8.1. The average input signal power over a pulse duration τ_0 is

$$S_i = \frac{E_x}{\tau_0} \tag{8.2}$$

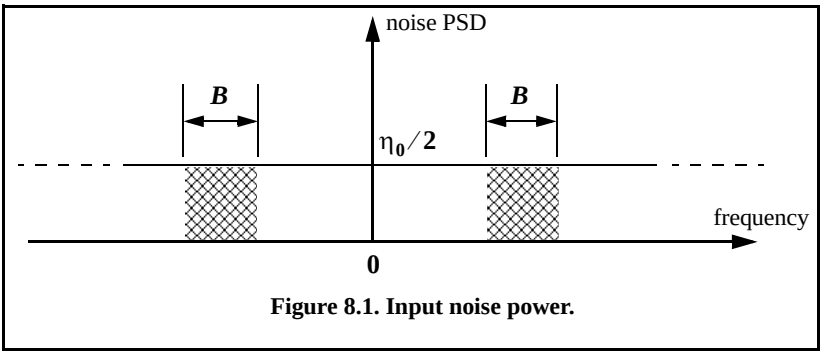


Figure 8.1. Input noise power.

E_x is the signal energy. Consequently, the matched filter input SNR is given by

$$(SNR)_i = \frac{S_i}{N_i} = \frac{E}{\eta_0 B\tau_0} \tag{8.3}$$

The output peak instantaneous SNR to the input SNR ratio is

$$\frac{SNR(t_0)}{(SNR)_i} = 2B\tau_0 \tag{8.4}$$

The quantity $B\tau_0$ is referred to as the time-bandwidth product for a given waveform or its corresponding matched filter. The factor $B\tau_0$ by which the output SNR is increased over that at the input is called the matched filter gain, or simply the compression gain.

In general, the time-bandwidth product of an unmodulated pulse approaches unity. The time-bandwidth product of a pulse can be made much greater than unity by using frequency or phase modulation. If the radar receiver transfer function is perfectly matched to that of the input waveform, then the compression gain is equal to $B\tau_0$. Clearly, the compression gain becomes smaller than $B\tau_0$ as the spectrum of the matched filter deviates from that of the input signal.

8.2. Radar Equation with Pulse Compression

The radar equation for a pulsed radar can be written as

$$SNR = \frac{P_t \tau_0 G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 F L} \tag{8.5}$$

where P_t is peak power, τ_0 is pulse width, G is antenna gain, σ is target RCS, R is range, k is Boltzmann's constant, T_0 is 290 degrees Kelvin, F is noise figure, and L is total radar losses.

Pulse compression radars transmit relatively long pulses (with modulation) and process the radar echo into very short pulses (compressed). One can view the transmitted pulse as being composed of a series of very short subpulses (duty is 100%), where the width of each subpulse is equal to the desired compressed pulse width. Denote the compressed pulse width as τ_c. Thus, for an individual subpulse, Eq. (8.5) can be written as

$$(SNR)_{\tau_c} = \frac{P_t \tau_c G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 F L} \tag{8.6}$$

The SNR for the uncompressed pulse is then derived from Eq. (8.6) as

$$SNR = \frac{P_t(\tau_0 = n_P \tau_c) G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 F L} \tag{8.7}$$

where n_P is the number of subpulses. Equation (8.7) is denoted as the radar equation with pulse compression.

Observation of Eq. (8.5) and Eq.(8.7) indicates the following (note that both equations have the same form): For a given set of radar parameters, and as long as the transmitted pulse remains unchanged, the SNR is also unchanged regardless of the signal bandwidth. More precisely, when pulse compression is used, the detection range is maintained while the range resolution is drastically improved by keeping the pulse width unchanged and by increasing the bandwidth. Remember that range resolution is proportional to the inverse of the signal bandwidth:

$$\Delta R = \frac{c}{2B} \tag{8.8}$$

8.3. Basic Principal of Pulse Compression

For this purpose, consider a long pulse with LFM modulation and assume a matched filter receiver. The output of the matched filter (along the delay axis, i.e., range) is an order of magnitude narrower than that at its input. More precisely, the matched filter output is compressed by a factor $\xi = B\tau_0$, where τ_0 is the pulse width and B is the bandwidth. Thus, by using long pulses and wideband LFM modulation, large compression ratios can be achieved.

Figure 8.2 shows an ideal LFM pulse compression process. Part (a) shows the envelope of a pulse, part (b) shows the frequency modulation (in this case it is an upchirp LFM) with bandwidth $B = f_2 - f_1$. Part (c) shows the matched filter time-delay characteristic while part (d) shows the compressed pulse envelope. Finally part (e) shows the matched filter input/output waveforms.

Figure 8.3 illustrates the advantage of pulse compression using a more realistic LFM waveform. In this example, two targets with RCS, $\sigma_1 = 1m^2$ and $\sigma_2 = 0.5m^2$, are detected. The two targets are not separated enough in time to be resolved. Figure 8.3a shows the composite echo signal from those targets. Clearly, the target returns overlap, and thus, they are not resolved. However, after pulse compression the two pulses are completely separated and are resolved as two distinct targets. In fact, when using LFM, returns from neighboring targets are resolved as long as they are separated in time by τ_c, the compressed pulse width.

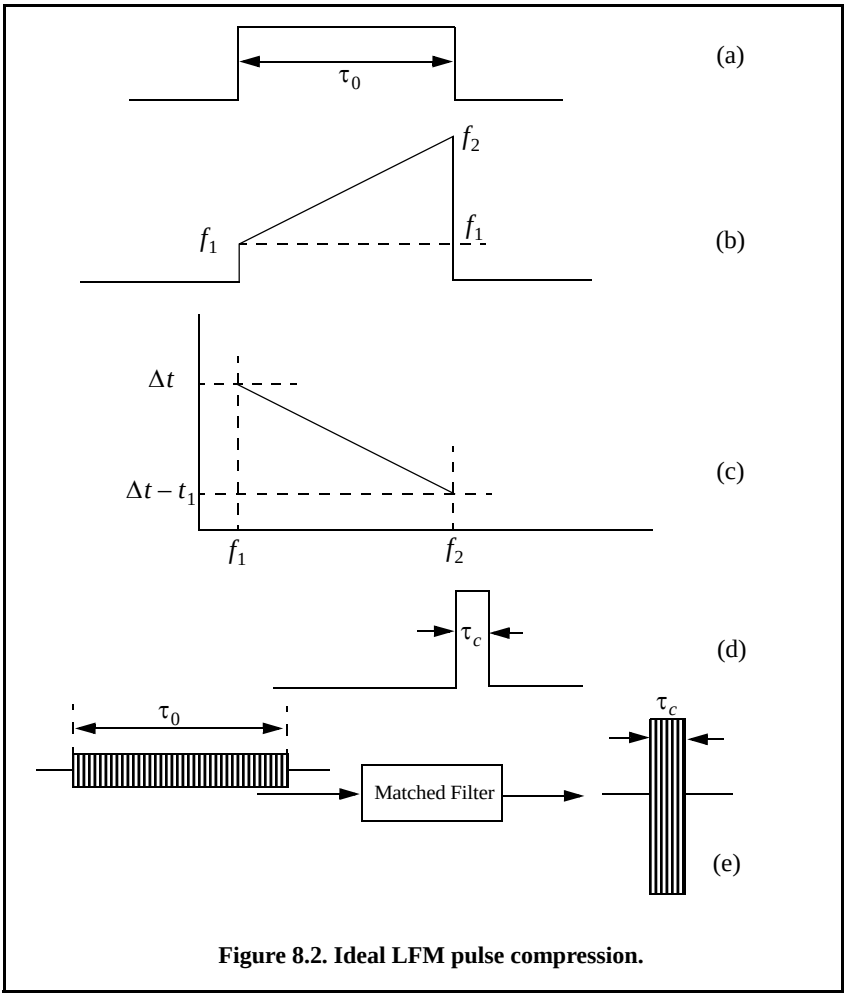


Figure 8.2. Ideal LFM pulse compression.

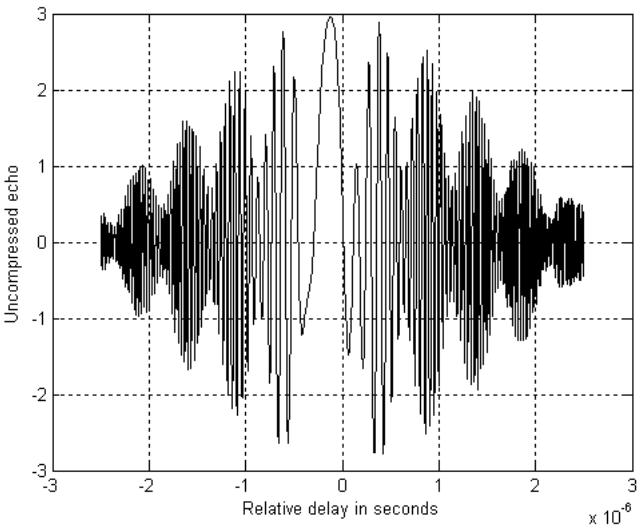


Figure 8.3a. Composite echo signal for two unresolved targets.

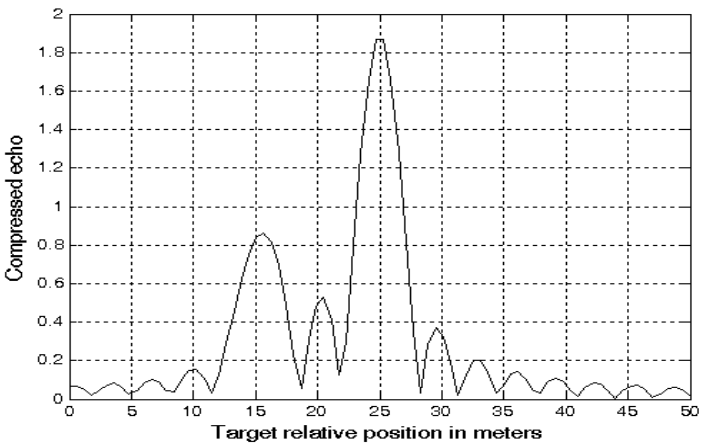


Figure 8.3b. Composite echo signal corresponding to Fig. 8.3a after pulse compression.

8.4. Correlation Processor

Radar operations (search, track, etc.) are usually carried out over a specified range window, referred to as the receive window and defined by the difference between the radar maximum and minimum range. Returns from all targets within the receive window are collected and passed through matched filter circuitry to perform pulse compression. One implementation of such analog processors is the Surface Acoustic Wave (SAW) devices. Because of the recent advances in digital computer development, the correlation processor is often performed digitally using the FFT. This digital implementation is called Fast Convolution Processing (FCP) and can be implemented at base band. The fast convolution process is illustrated in Fig. 8.4.

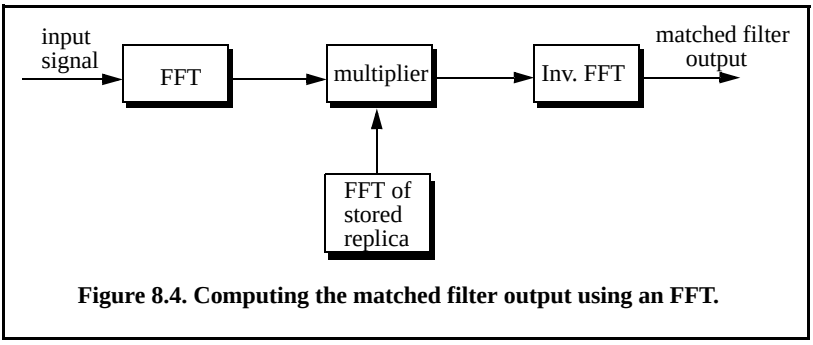


Figure 8.4. Computing the matched filter output using an FFT.

Since the matched filter is a linear time invariant system, its output can be described mathematically by the convolution between its input and its impulse response,

$$y(t) = s((t) \otimes h(t)) \tag{8.9}$$

where $s(t)$ is the input signal, $h(t)$ is the matched filter impulse response (replica), and the ($\otimes$) operator symbolically represents convolution. From the Fourier transform properties,

$$FFT\{s((t) \otimes h(t))\} = S(f) \cdot H(f) \tag{8.10}$$

and when both signals are sampled properly, the compressed signal $y(t)$ can be computed from

$$y = FFT^{-1}\{S \cdot H\} \tag{8.11}$$

where FFT^{-1} is the inverse FFT. When using pulse compression, it is desirable to use modulation schemes that can accomplish a maximum pulse compression ratio and can significantly reduce the sidelobe levels of the compressed waveform. For the LFM case the first sidelobe is approximately

$13.4dB$ below the main peak, and for most radar applications this may not be sufficient. In practice, high sidelobe levels are not preferable because noise and/or jammers located at the sidelobes may interfere with target returns in the main lobe.

Weighting functions (windows) can be used on the compressed pulse spectrum in order to reduce the sidelobe levels. The cost associated with such an approach is a loss in the main lobe resolution, and a reduction in the peak value (i.e., loss in the SNR). Weighting the time domain transmitted or received signal instead of the compressed pulse spectrum will theoretically achieve the same goal. However, this approach is rarely used, since amplitude modulating the transmitted waveform introduces extra burdens on the transmitter.

Consider a radar system that utilizes a correlation processor receiver (i.e., matched filter). The receive window in meters is defined by

$$R_{rec} = R_{max} - R_{min} \tag{8.12}$$

where R_{max} and R_{min}, respectively, define the maximum and minimum range over which the radar performs detection. Typically R_{rec} is limited to the extent of the target complex. The normalized complex transmitted signal has the form

$$s(t) = \exp\left(j2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le \tau_0 \tag{8.13}$$

τ_0 is the pulse width, $\mu = B/\tau_0$, and B is the bandwidth.

The radar echo signal is similar to the transmitted one with the exception of a time delay and an amplitude change that correspond to the target RCS. Consider a target at range R_1. The echo received by the radar from this target is

$$s_r(t) = a_1 \exp\left(j2\pi\left(f_0(t-t_1) + \frac{\mu}{2}(t-t_1)^2\right)\right) \tag{8.14}$$

where a_1 is proportional to target RCS, antenna gain, and range attenuation. The time delay t_1 is given by

$$t_1 = 2R_1/c \tag{8.15}$$

The first step of the processing consists of removing the frequency f_0. This is accomplished by mixing $s_r(t)$ with a reference signal whose phase is $2\pi f_0 t$. The phase of the resultant signal, after lowpass filtering, is then given by

$$\phi(t) = 2\pi\left(-f_0 t_1 + \frac{\mu}{2}(t-t_1)^2\right) \tag{8.16}$$

and the instantaneous frequency is

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\phi(t) = \mu(t - t_1) = \frac{B}{\tau_0}\left(t - \frac{2R_1}{c}\right) \tag{8.17}$$

The quadrature components are

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} \cos\phi(t) \\ \sin\phi(t) \end{pmatrix} \tag{8.18}$$

Sampling the quadrature components is performed next. The number of samples, N, must be chosen so that foldover (ambiguity) in the spectrum is avoided. For this purpose, the sampling frequency, f_s (based on the Nyquist sampling rate), must be

$$f_s \geq 2B \tag{8.19}$$

and the sampling interval is

$$\Delta t \leq 1/2B \tag{8.20}$$

Using Eq. (8.17) it can be shown that (the proof is left as an exercise) the frequency resolution of the FFT is

$$\Delta f = 1/\tau_0 \tag{8.21}$$

The minimum required number of samples is

$$N = \frac{1}{\Delta f \Delta t} = \frac{\tau_0}{\Delta t} \tag{8.22}$$

Equating Eqs. (8.20) and (8.22) yields

$$N \geq 2B\tau_0 \tag{8.23}$$

Consequently, a total of $2B\tau_0$ real samples, or $B\tau_0$ complex samples, is sufficient to completely describe an LFM waveform of duration τ_0 and bandwidth B. For example, an LFM signal of duration $\tau_0 = 20\ \mu s$ and bandwidth $B = 5\ MHz$ requires 200 real samples to determine the input signal (100 samples for the I-channel and 100 samples for the Q-channel).

For better implementation of the FFT N is extended to the next power of two, by zero padding. Thus, the total number of samples, for some positive integer m, is

$$N_{FFT} = 2^m \geq N \tag{8.24}$$

The final steps of the FCP processing include (1) taking the FFT of the sampled sequence, (2) multiplying the frequency domain sequence of the signal

with the FFT of the matched filter impulse response, and (3) performing the inverse FFT of the composite frequency domain sequence in order to generate the time domain compressed pulse. Of course, weighting, antenna gain, and range attenuation compensation must also be performed.

Assume that I targets at ranges R_1, R_2, and so forth are within the receive window. From superposition, the phase of the down-converted signal is

$$\phi(t) = \sum_{i=1}^{I} 2\pi\left(-f_0 t_i + \frac{\mu}{2}(t - t_i)^2\right) \qquad \textbf{(8.25)}$$

The times $\{t_i = (2R_i/c);\ i = 1, 2, \ldots, I\}$ represent the two-way time delays, where t_1 coincides with the start of the receive window. As an example, consider the case where

# targets	R_{rec}	*pulse width*	*Band-width*	*targets range*	*Target RCS*	*Window type*
3	*200m*	*0.005ms*	*100e6 Hz*	*[10 75 120] m*	*[1 2 1]m^2*	*Ham-ming*

Note that the compressed pulsed range resolution is $\Delta R = 1.5m$. Figure 8.5 and Fig. 8.6 shows the real part and the amplitude spectrum of the replica used for this example. Figure 8.7 shows the uncompressed echo, while Fig. 8.8 shows the compressed MF output. Note that the scatterer amplitude attenuation is a function of the inverse of the scatterer's range within the receive window. Figure 8.9 is similar to Fig. 8.8, except in this case the first and second scatterers are less than 1.5 meter apart (they are at 70 and 71 meters).

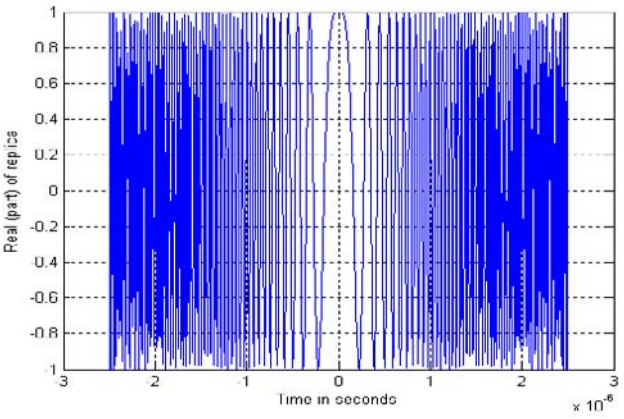


Figure 8.5. Real part of replica.

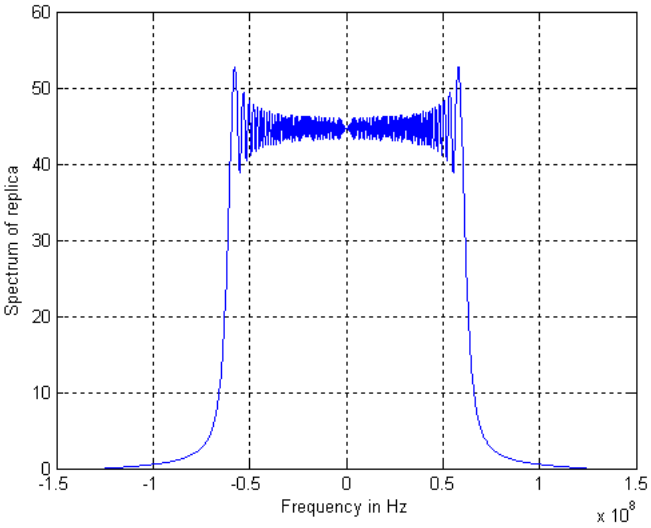


Figure 8.6. Replica spectrum.

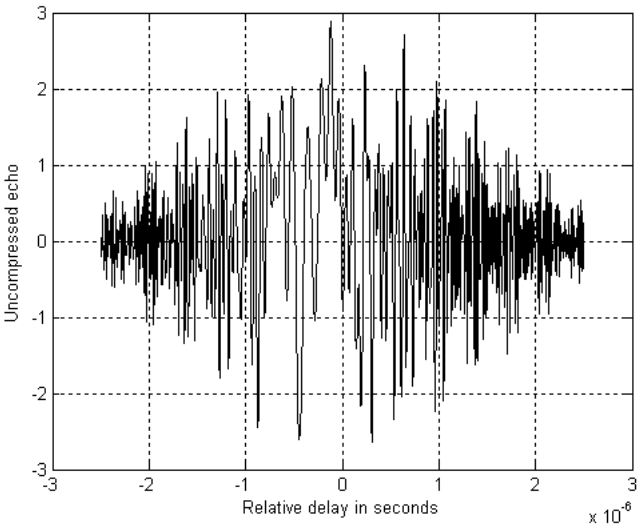


Figure 8.7. Uncompressed echo signal. Scatterers are not resolved.

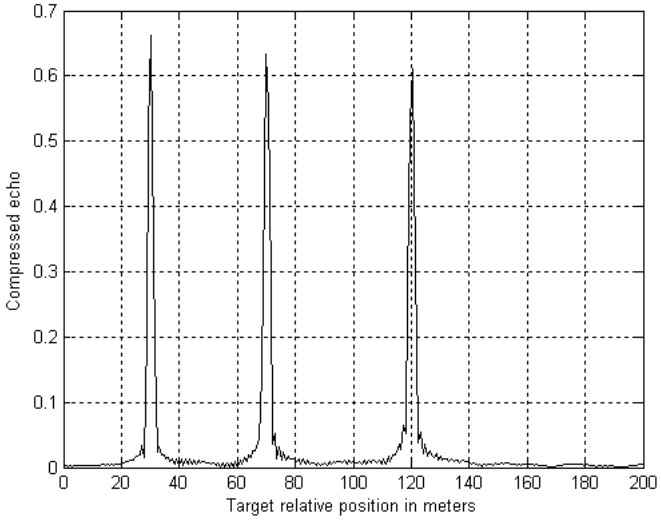


Figure 8.8. Compressed echo signal corresponding to Fig. 8.7. Scatterers are completely resolved.

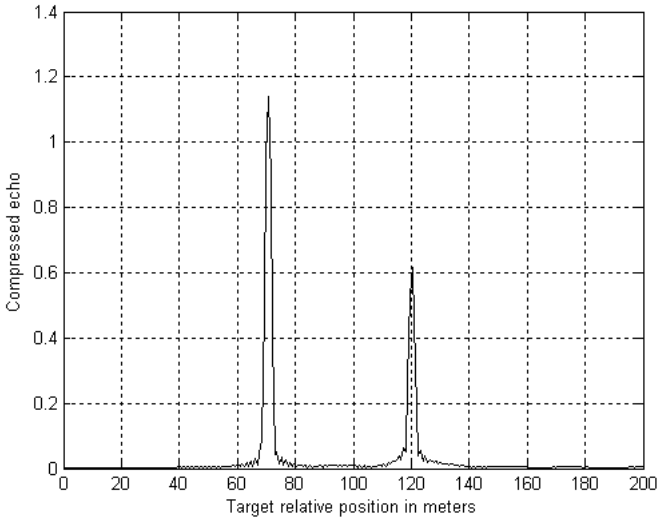


Figure 8.8. Compressed echo signal corresponding to Fig. 5.7. Scatterers are completely resolved.

8.5. Stretch Processor

Stretch processing, also known as *active correlation*, is normally used to process extremely high-bandwidth LFM waveforms. This processing technique consists of the following steps: First, the radar returns are mixed with a replica (reference signal) of the transmitted waveform. This is followed by Low Pass Filtering (LPF) and coherent detection. Next, Analog-to-Digital (A/D) conversion is performed; and finally, a bank of Narrow-Band Filters (NBFs) is used in order to extract the tones that are proportional to target range, since stretch processing effectively converts time delay into frequency. All returns from the same range bin produce the same constant frequency.

8.5.1. Single LFM Pulse

Figure 8.10 shows a block diagram for a stretch processing receiver. The reference signal is an LFM waveform that has the same LFM slope as the transmitted LFM signal. It exists over the duration of the radar "receive-window," which is computed from the difference between the radar maximum and minimum range. Denote the start frequency of the reference chirp as f_r. Consider the case when the radar receives returns from a few close (in time or range) targets, as illustrated in Fig. 8.10. Mixing with the reference signal and performing lowpass filtering are effectively equivalent to subtracting the return frequency chirp from the reference signal. Thus, the LPF output consists of constant tones corresponding to the targets' positions. The normalized transmitted signal can be expressed by

$$s_1(t) = \cos\left(2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le \tau_0 \tag{8.26}$$

where $\mu = B/\tau_0$ is the LFM coefficient and f_0 is the chirp start frequency. Assume a point scatterer at range R. The signal received by the radar is

$$s_r(t) = a\cos\left[2\pi\left(f_0(t-t_0) + \frac{\mu}{2}(t-t_0)^2\right)\right] \tag{8.27}$$

where a is proportional to target RCS, antenna gain, and range attenuation. The time delay t_0 is

$$t_0 = 2R/c \tag{8.28}$$

The reference signal is

$$s_{ref}(t) = 2\cos\left(2\pi\left(f_r t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le T_{rec} \tag{8.29}$$

The receive window in seconds is

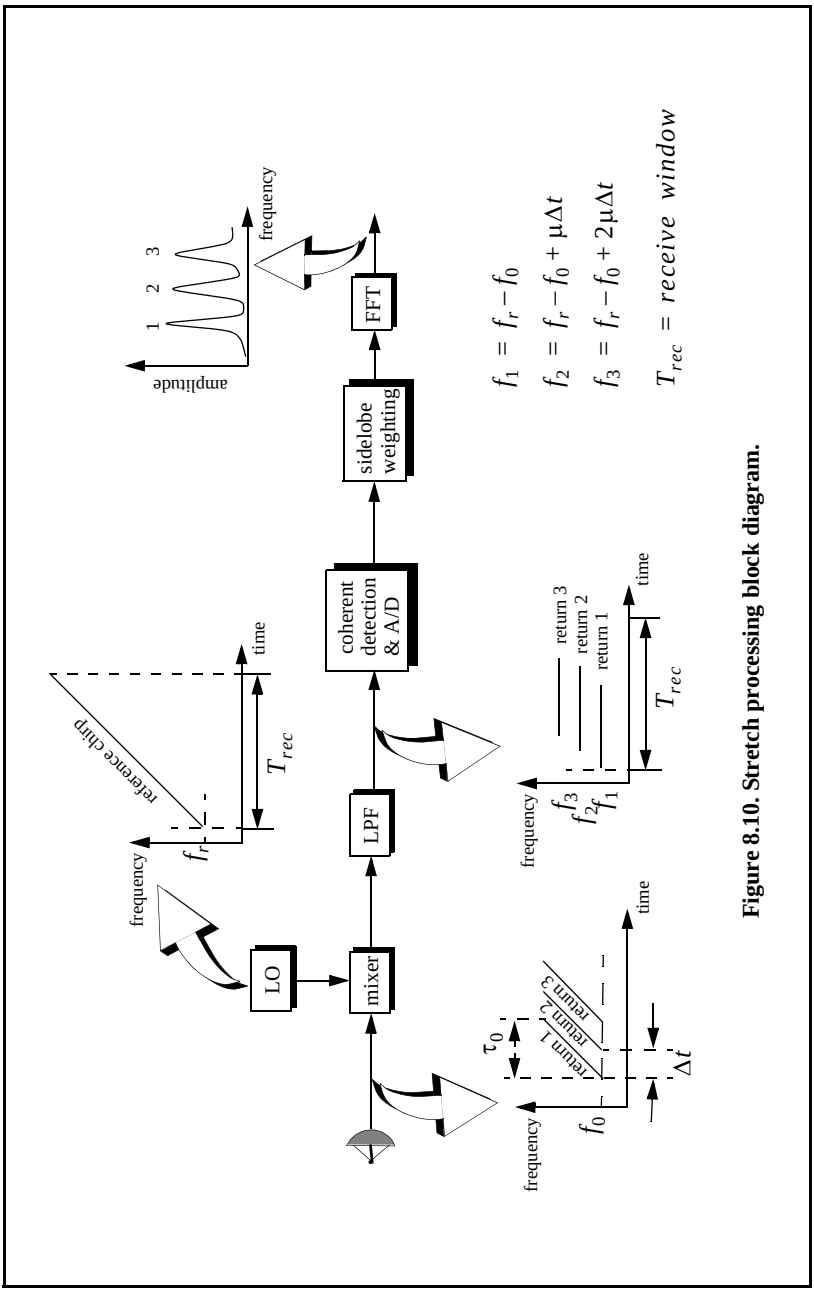


Figure 8.10. Stretch processing block diagram.

$$T_{rec} = \frac{2(R_{max} - R_{min})}{c} = \frac{2R_{rec}}{c} \tag{8.30}$$

It is customary to let $f_r = f_0$. The output of the mixer is the product of the received and reference signals. After lowpass filtering the signal is

$$s_0(t) = a\cos(2\pi f_0 t_0 + 2\pi\mu t_0 t - \pi\mu(t_0)^2) \tag{8.31}$$

Substituting Eq. (8.28) into Eq. (8.31) and collecting terms yield

$$s_0(t) = a \cos\left[\left(\frac{4\pi BR}{c\tau_0}\right)t + \frac{2R}{c}\left(2\pi f_0 - \frac{2\pi BR}{c\tau_0}\right)\right] \tag{8.32}$$

and since $\tau_0 \gg 2R/c$, Eq. (8.32) is approximated by

$$s_0(t) \approx a \cos\left[\left(\frac{4\pi BR}{c\tau_0}\right)t + \frac{4\pi R}{c} f_0\right] \tag{8.33}$$

The instantaneous frequency is

$$f_{inst} = \frac{1}{2\pi}\frac{d}{dt}\left(\left(\frac{4\pi BR}{c\tau_0}t + \frac{4\pi R}{c}f_0\right)\right) = \frac{2BR}{c\tau_0} \tag{8.34}$$

which clearly indicates that target range is proportional to the instantaneous frequency. Therefore, proper sampling of the LPF output and taking the FFT of the sampled sequence lead to the following conclusion: a peak at some frequency f_1 indicates presence of a target at range

$$R_1 = f_1 c\tau_0/2B \tag{8.35}$$

Assume I close targets at ranges R_1, R_2, and so forth ($R_1 < R_2 < \ldots < R_I$). From superposition, the total signal is

$$s_r(t) = \sum_{i=1}^{I} a_i(t)\cos\left[2\pi\left(f_0(t - t_i) + \frac{\mu}{2}(t - t_i)^2\right)\right] \tag{8.36}$$

where $\{a_i(t);\ i = 1, 2, \ldots, I\}$ are proportional to the targets' cross sections, antenna gain, and range. The times $\{t_i = (2R_i/c);\ i = 1, 2, \ldots, I\}$ represent the two-way time delays, where t_1 coincides with the start of the receive window. Using Eq. (8.32) the overall signal at the output of the LPF can then be described by

$$s_o(t) = \sum_{i=1}^{I} a_i \cos\left[\left(\frac{4\pi BR_i}{c\tau_0}\right)t + \frac{2R_i}{c}\left(2\pi f_0 - \frac{2\pi BR_i}{c\tau_0}\right)\right] \tag{8.37}$$

Hence, target returns appear as constant frequency tones that can be resolved using the FFT. Consequently, determining the proper sampling rate and FFT size is very critical. The rest of this section presents a methodology for computing the proper FFT parameters required for stretch processing.

Assume a radar system using a stretch processor receiver. The pulse width is τ_0 and the chirp bandwidth is B. Since stretch processing is normally used in extreme bandwidth cases (i.e., very large B), the receive window over which radar returns will be processed is typically limited to from a few meters to possibly less than 100 meters. The compressed pulse range resolution is computed from Eq. (8.8). Declare the FFT size to be N and its frequency resolution to be Δf. The frequency resolution can be computed using the following procedure: Consider two adjacent point scatterers at ranges R_1 and R_2. The minimum frequency separation, Δf, between those scatterers so that they are resolved can be computed from Eq. (8.34). More precisely,

$$\Delta f = f_2 - f_1 = \frac{2B}{c\tau_0}(R_2 - R_1) = \frac{2B}{c\tau_0}\Delta R \tag{8.38}$$

Substituting Eq. (8.8) into Eq. (8.38) yields

$$\Delta f = \frac{2B}{c\tau_0}\ \frac{c}{2B} = \frac{1}{\tau_0} \tag{8.39}$$

The maximum frequency resolvable by the FFT is limited to the region $\pm N\Delta f/2$. Thus, the maximum resolvable frequency is

$$\frac{N\Delta f}{2} > \frac{2B(R_{max} - R_{min})}{c\tau_0} = \frac{2BR_{rec}}{c\tau_0} \tag{8.40}$$

Using Eqs. (8.30) and (8.39) into Eq. (8.40) and collecting terms yield

$$N > 2BT_{rec} \tag{8.41}$$

For better implementation of the FFT, choose an FFT of size

$$N_{FFT} \geq N = 2^m \tag{8.42}$$

where m is a nonzero positive integer. The sampling interval is then given by

$$\Delta f = \frac{1}{T_s N_{FFT}} \Rightarrow T_s = \frac{1}{\Delta f N_{FFT}} \tag{8.43}$$

As an example, consider the case where

# targets	*3*
pulsewidth	*10 ms*
center frequency	*5.6 GHz*
bandwidth	*1 GHz*
receive window	*30 m*
relative target's range	*[2 5 10] m*
target's RCS	*[1, 1, 2] m^2*
window	*2 (Kaiser)*

Note that the compressed pulse range resolution, without using a window, is $\Delta R = 0.15m$. Figure 8.11 and Fig. 8.12, respectively, show the uncompressed and compressed echo signals corresponding to this example. Figures 8.13 a and b are similar to Fig. 8.11 and Fig. 8.12 except in this case two of the scatterers are less than 15 cm apart (i.e., unresolved targets at $R_{relative} = [3, 3.1]m$).

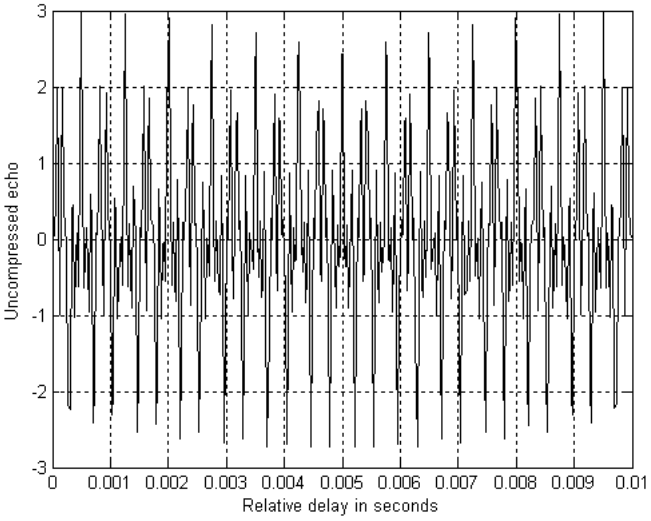


Figure 8.11. Uncompressed echo signal. Three targets are unresolved.

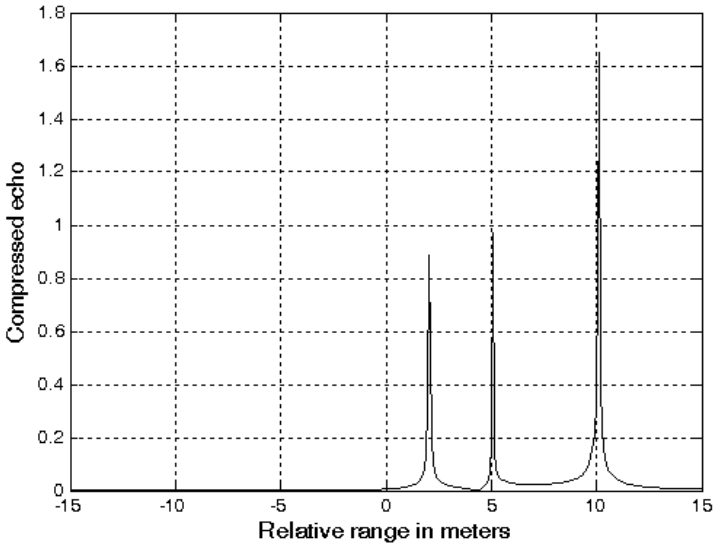


Figure 8.12. Compressed echo signal. Three targets are resolved.

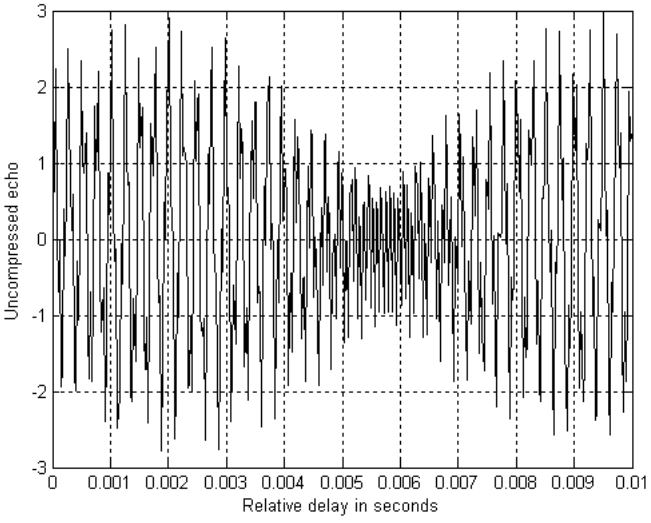


Figure 8.13a. Uncompressed echo signal. Three targets.

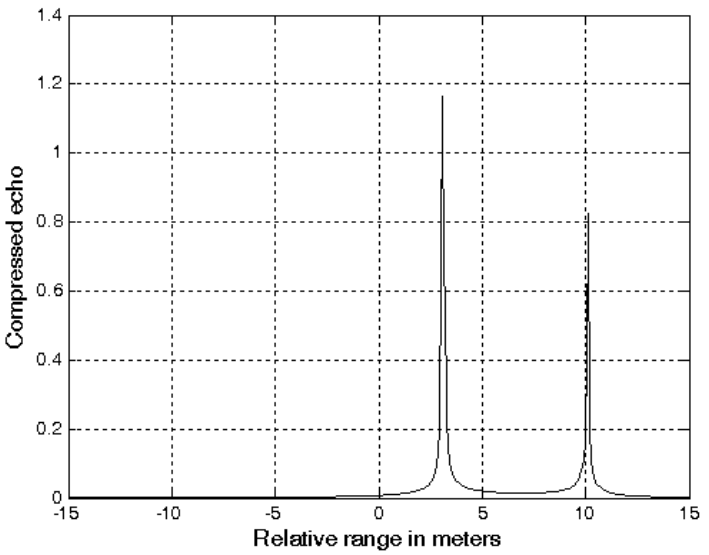


Figure 8.13b. Compressed echo signal. Three targets, two are not resolved.

8.5.2. Stepped Frequency Waveforms

Stepped Frequency Waveforms (SFW) are used in extremely wide band radar applications where very large time bandwidth product is required. Generation of SFW was discussed in Chapter 5. For this purpose, consider an LFM signal whose bandwidth is B_i and whose pulsewidth is T_i and refer to it as the primary LFM. Divide this long pulse into N subpulses each of width τ_0 to generate a sequence of pulses whose PRI is denoted by T. It follows that $T_i = (n-1)T$. Define the beginning frequency for each subpulse as that value measured from the primary LFM at the leading edge of each subpulse, as illustrated in Fig. 8.14. That is

$$f_i = f_0 + i\Delta f; \;\; i = 0, N-1 \tag{8.44}$$

where Δf is the frequency step from one subpulse to another. The set of n subpulses is often referred to as a burst. Each subpulse can have its own LFM modulation. To this end, assume that each subpluse is of width τ_0 and bandwidth B, then the LFM slope of each pulse is

$$\mu = \frac{B}{\tau_0} \tag{8.45}$$

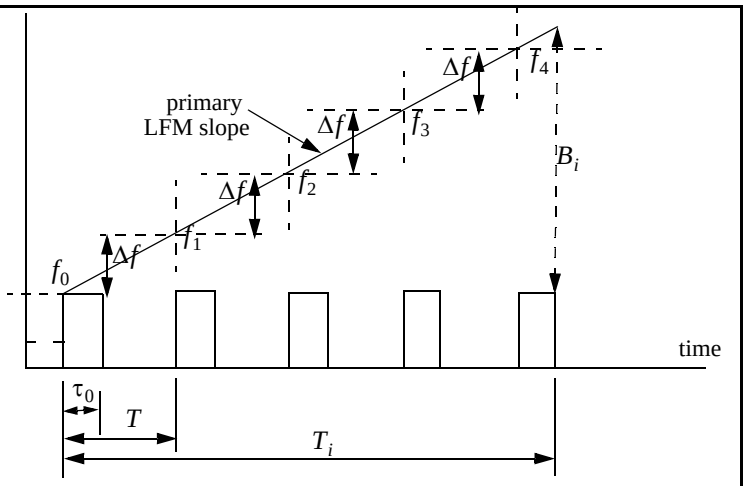


Figure 8.14. Example of stepped frequency waveform burst; $N = 5$.

The SFW operation and processing involve the following steps:

1. A series of N narrow-band LFM pulses is transmitted. The chirp beginning frequency from pulse to pulse is stepped by a fixed frequency step Δf, as defined in Eq. (8.44). Each group of N pulses is referred to as a burst.
2. The LFM slope (quadratic phase term) is first removed from the received signal, as described in Fig. 8.10. The reference slope must be equal to the combined primary LFM and single subpulse slopes. Thus, the received signal is reduced to a series of subpulses.
3. These subpulses are then sampled at a rate that coincides with the center of each pulse, sampling rate equivalent to $(1/T)$.
4. The quadrature components for each burst are collected and stored.
5. Spectral weighting (to reduce the range sidelobe levels) is applied to the quadrature components. Corrections for target velocity, phase, and amplitude variations are applied.
6. The IDFT of the weighted quadrature components of each burst is calculated to synthesize a range profile for that burst. The process is repeated for M bursts to obtain consecutive high resolution range profiles.

Within a burst, the transmitted waveform for the i^{th} step can be described as

$$x_i(t) = \begin{cases} C_i \dfrac{1}{\sqrt{\tau_0}} Rect\left(\dfrac{t}{\tau_0}\right) e^{j2\pi\left(f_i t + \frac{\mu}{2}t^2\right)} & ; \quad iT \le t \le iT + \tau_0 \\ 0 & elsewhere \end{cases} \qquad \textbf{(3.46)}$$

where C_i are constants. The received signal from a target located at range R_0 is then given by

$$x_{ri}(t) = C_i' e^{j2\pi\left[f_i(t-\Delta(t))-\frac{\mu}{2}(t-\Delta(t))^2\right]}, \quad iT+\Delta(t) \le t \le iT+\tau_0+\Delta(t) \tag{8.47}$$

where C_i' are constant and the round trip delay $\Delta(t)$ is given by

$$\Delta(t) = \frac{R_0 - vt}{c/2} \tag{8.48}$$

where c is the speed of light and v is the target radial velocity.

In order to remove the quadratic phase term, mixing is first performed with the reference signal given by

$$y_i(t) = e^{j2\pi\left(f_i t+\frac{\mu}{2}t^2\right)} \quad ; \quad iT \le t \le iT+\tau_0 \tag{8.49}$$

Next lowpass filtering is performed to extract the quadrature components. More precisely, the quadrature components are given by

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} A_i \cos\phi_i(t) \\ A_i \sin\phi_i(t) \end{pmatrix} \tag{8.50}$$

where A_i are constants, and

$$\phi_i(t) = -2\pi f_i\left(\frac{2R_0}{c} - \frac{2vt}{c}\right) \tag{8.51}$$

where now $f_i = \Delta f$. For each pulse, the quadrature components are then sampled at

$$t_i = iT + \frac{\tau_r}{2} + \frac{2R_0}{c} \tag{8.52}$$

τ_r is the time delay associated with the range that corresponds to the start of the range profile.

The quadrature components can then be expressed in complex form as

$$X_i = A_i e^{j\phi_i} \tag{8.53}$$

Equation (8.53) represents samples of the target reflectivity, due to a single burst, in the frequency domain. This information can then be transformed into a series of range delay reflectivity (i.e., range profile) values by using the IDFT. It follows that

$$H_l = \frac{1}{N}\sum_{i=0}^{N-1} X_i \ \exp\left(j\frac{2\pi li}{N}\right) \qquad ; \ 0 \le l \le N-1 \tag{8.54}$$

Substituting Eq. (8.51) and Eq. (8.53) into (8.54) and collecting terms yield

$$H_l = \frac{1}{N}\sum_{i=0}^{N-1} A_i \ \exp\left\{j\left(\frac{2\pi li}{N} - 2\pi f_i\left(\frac{2R_0}{c} - \frac{2vt_i}{c}\right)\right)\right\} \tag{8.55}$$

By normalizing with respect to N and by assuming that $A_i = 1$ and that the target is stationary (i.e., $v = 0$), then Eq. (8.55) can be written as

$$H_l = \sum_{i=0}^{N-1} \exp\left\{j\left(\frac{2\pi li}{N} - 2\pi f_i\frac{2R_0}{c}\right)\right\} \tag{8.56}$$

Using $f_i = i\Delta f$ inside Eq. (8.56) yields

$$H_l = \sum_{i=0}^{N-1} \exp\left\{j\frac{2\pi i}{N}\left(-\frac{2NR_0\Delta f}{c} + l\right)\right\} \tag{8.57}$$

which can be simplified to

$$H_l = \frac{\sin\pi\zeta}{\sin\frac{\pi\zeta}{N}} \ \exp\left(j\frac{N-1}{2} \ \frac{2\pi\zeta}{N}\right) \tag{8.58}$$

where

$$\zeta = \frac{-2NR_0\Delta f}{c} + l \tag{8.59}$$

Finally, the synthesized range profile is

$$\left|H_l\right| = \left|\frac{\sin\pi\zeta}{\sin\frac{\pi\zeta}{N}}\right| \tag{8.60}$$

Range Resolution and Range Ambiguity in SFW

As usual, range resolution is determined from the overall system bandwidth. Assuming an SFW with N steps and step size Δf, then the corresponding range resolution is equal to

$$\Delta R = \frac{c}{2N\Delta f} \tag{8.61}$$

Range ambiguity associated with an SFW can be determined by examining the phase term that corresponds to a point scatterer located at range R_0. More precisely,

$$\phi_i(t) = 2\pi f_i \frac{2R_0}{c} \tag{8.62}$$

It follows that

$$\frac{\Delta\phi}{\Delta f} = \frac{4\pi(f_{i+1} - f_i)}{(f_{i+1} - f_i)}\frac{R_0}{c} = \frac{4\pi R_0}{c} \tag{8.63}$$

or equivalently,

$$R_0 = \frac{\Delta\phi}{\Delta f}\frac{c}{4\pi} \tag{8.64}$$

It is clear from Eq. (8.64) that range ambiguity exists for $\Delta\phi = \Delta\phi + 2N\pi$. Therefore,

$$R_0 = \frac{\Delta\phi + 2N\pi}{\Delta f}\frac{c}{4\pi} = R_0 + N\left(\frac{c}{2\Delta f}\right) \tag{8.65}$$

and the unambiguous range window is

$$R_u = \frac{c}{2\Delta f} \tag{8.66}$$

A range profile synthesized using a particular SFW represents the relative range reflectivity for all scatterers within the unambiguous range window, with respect to the absolute range that corresponds to the burst time delay. Additionally, if a specific target extent is larger than R_u, then all scatterers falling outside the unambiguous range window will fold over and appear in the synthesized profile. This fold-over problem is identical to the spectral fold-over that occurs when using a Fast Fourier Transform (FFT) to resolve certain signal frequency contents. For example, consider an FFT with frequency resolution $\Delta f = 50Hz$ and size $NFFT = 64$. In this case, this FFT can resolve frequency tones between $-1600Hz$ and $1600Hz$. When this FFT is used to resolve the frequency content of a sine-wave tone equal to $1800Hz$, fold-over occurs and a spectral line at the fourth FFT bin (i.e., $200Hz$) appears. Therefore, in order to avoid fold-over in the synthesized range profile, the frequency step Δf must be

$$\Delta f \le c/2E \tag{8.67}$$

where E is the target extent in meters.

Additionally, the pulsewidth must also be large enough to contain the whole target extent. Thus,

$$\Delta f \leq 1/\tau_0 \tag{8.68}$$

and in practice,

$$\Delta f \leq 1/2\tau_0 \tag{8.69}$$

This is necessary in order to reduce the amount of contamination of the synthesized range profile caused by the clutter surrounding the target under consideration.

For example, assume that the range profile starts at $R_0 = 900m$ and that

# targets	*pulsewidth*	*N*	Δf	*1/T*	*v*
3	100μsec	*64*	$10MHz$	$100KHz$	*0.0*

In this case,

$$\Delta R = \frac{3\times 10^8}{2\times 64\times 10\times 10^6} = 0.235m, \text{ and } R_u = \frac{3\times 10^8}{2\times 10\times 10^6} = 15m$$

Thus, scatterers that are more than 0.235 meters apart will appear as distinct peaks in the synthesized range profile. Assume two cases; in the first case, [scat_range] = [908, 910, 912] meters, and in the second case, [scat_range] = [908, 910, 910.2] meters. In both cases, let [scat_rcs] = [100, 10, 1] meters squared. Figure 8.15 shows the synthesized range profiles generated using the function "SWF.m" and the first case when the Hamming window is not used. Figure 8.16 is similar to Fig. 8.15, except in this case the Hamming window is used. Figure 8.17 shows the synthesized range profile that corresponds to the second case (Hamming window is used). Note that all three scatterers were resolved in Fig. 8.15 and Fig. 8.16; however, the last two scatterers are not resolved in Fig. 8.17, because they are separated by less than ΔR.

Next, consider another case where *[scat_range] = [908, 912, 916] meters.* Figure 8.18 shows the corresponding range profile. In this case, foldover occurs, and the last scatterer appears at the lower portion of the synthesized range profile. Also, consider the case where [scat_range] = [908, 910, 923] meters. Figure 8.19 shows the corresponding range profile. In this case, ambiguity is associated with the first and third scatterers since they are separated by $15m$. Both appear at the same range bin.

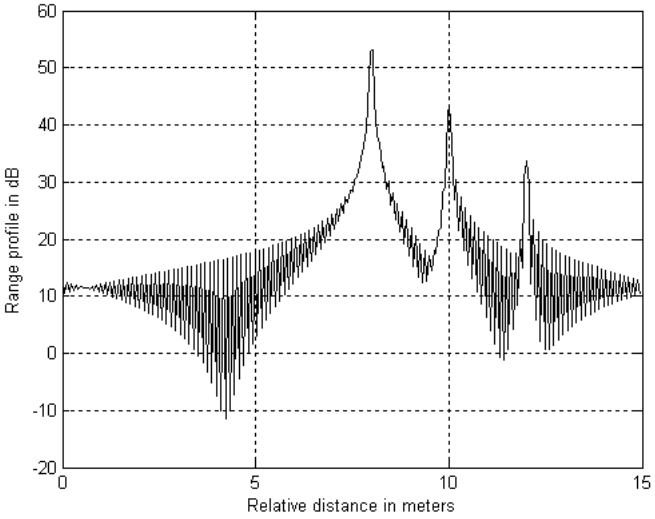


Figure 8.15. Synthetic range profile for three resolved scatterers. No window.

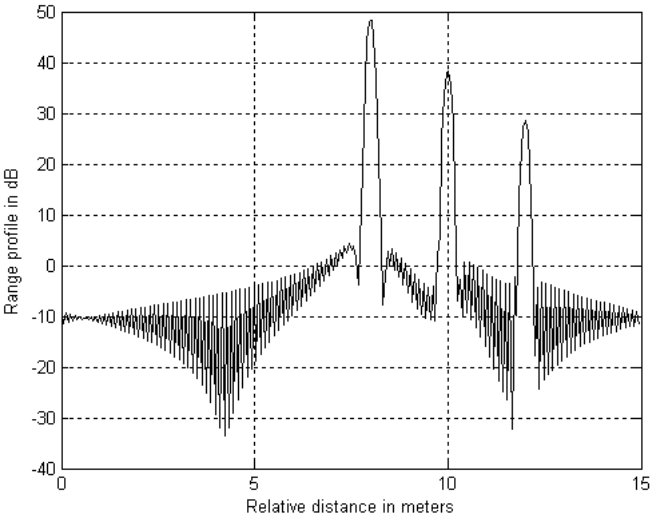


Figure 8.16. Synthetic range profile for three scatterers. Hamming window.

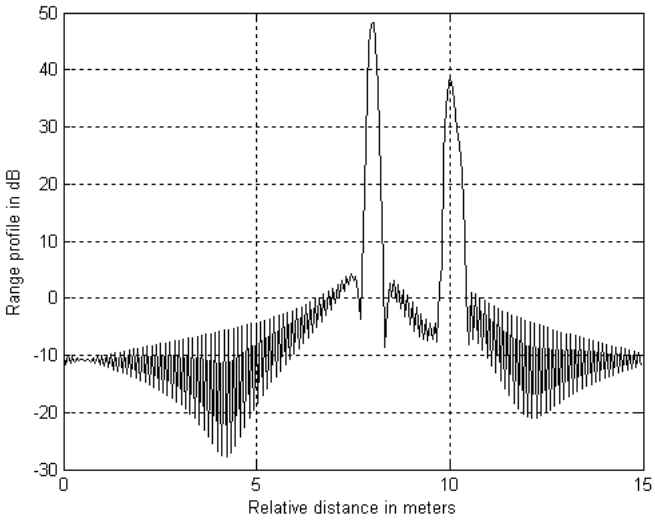


Figure 8.17. Synthetic range profile for three scatterers. Two are unresolved.

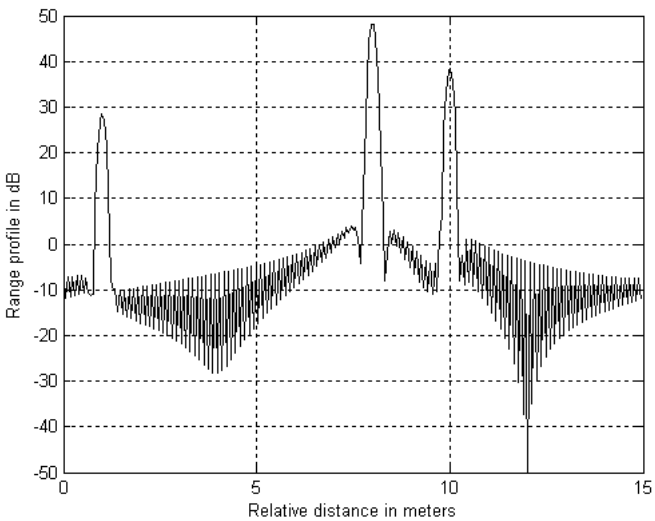


Figure 8.18. Synthetic range profile for three scatterers. Third scatterer folds over.

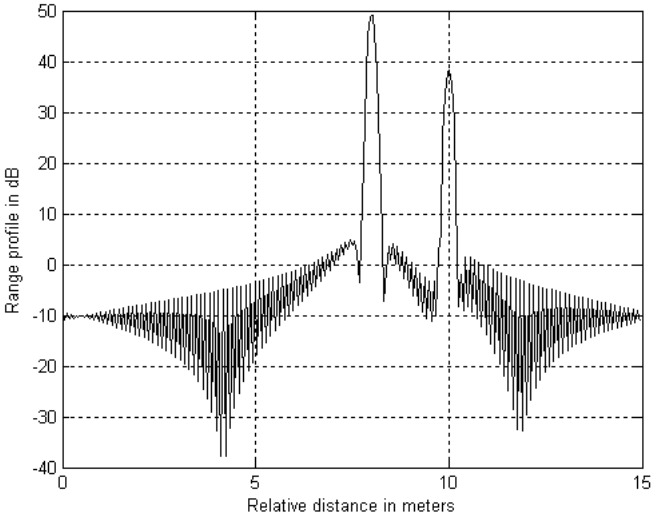


Figure 8.19. Synthetic range profile for three scatterers. The first and third scatterers appear in the same FFT bin.

8.5.2.1. Effect of Target Velocity

The range profile defined in Eq. (8.60) is obtained by assuming that the target under examination is stationary. The effect of target velocity on the synthesized range profile can be determined by starting with Eq. (8.55) and assuming that $v \neq 0$. Performing similar analysis as that of the stationary target case yields a range profile given by

$$H_l = \sum_{i=0}^{N-1} A_i \exp\left\{ j\frac{2\pi li}{N} - j2\pi f_i \left[\frac{2R}{c} - \frac{2v}{c}\left(iT + \frac{\tau_r}{2} + \frac{2R}{c} \right) \right] \right\} \qquad \textbf{(8.70)}$$

The additional phase term present in Eq. (8.70) distorts the synthesized range profile. In order to illustrate this distortion, consider the SFW described in the previous section, and assume the three scatterers of the first case. Also, assume that $v = 200 m/s$. Figure 8.20 shows the synthesized range profile for this case. Comparisons of Figs. 8.16 and 8.20 clearly show the distortion effects caused by the uncompensated target velocity. Figure 8.21 is similar to Fig. 8.20 except in this case, $v = -200 m/s$. Note in either case, the targets have moved from their expected positions (to the left or right) by $Disp = 2 \times n \times v/PRF$ *(1.28 m).*

This distortion can be eliminated by multiplying the complex received data at each pulse by the phase term

$$\Phi = \exp\left(-j2\pi f_i\left[\frac{2\hat{v}}{c}\left(iT + \frac{\tau_r}{2} + \frac{2\hat{R}}{c}\right)\right]\right) \quad \textbf{(3.71)}$$

$\hat{v}$ and R are, respectively, estimates of the target velocity and range. This process of modifying the phase of the quadrature components is often referred to as "phase rotation." In practice, when good estimates of $\hat{v}$ and $\hat{R}$ are not available, then the effects of target velocity are reduced by using frequency hopping between the consecutive pulses within the SFW. In this case, the frequency of each individual pulse is chosen according to a predetermined code. Waveforms of this type are often called Frequency Coded Waveforms (FCW). Costas waveforms or signals are a good example of this type of waveform.

Figure 8.22 shows a synthesized range profile for a moving target whose RCS is $\sigma = 10m^2$ and $v = 10m/s$. The initial target range is at $R = 912m$. All other parameters are as before. This figure can be reproduced using the following MATLAB code.

```
clear all;
close all;
nscat = 1;
scat_range = 912;
scat_rcs = 10;
n =64;
deltaf = 10e6;
prf = 10e3;
v = 10;
rnote = 900,
winid = 1;
count = 0;
for time = 0:.05:3
   count = count +1;
   hl = SFW (nscat, scat_range, scat_rcs, n, deltaf, prf, v, rnote, winid);
   array(count,:) = transpose(hl);
   hl(1:end) = 0;
   scat_range =  scat_range - 2 * n * v / prf;
end
figure (1)
 numb = 2*256;% this number matches that used in hrr_profile.
 delx_meter = 15 / numb;
 xmeter = 0:delx_meter:15-delx_meter;
 imagesc(xmeter, 0:0.05:4,array)
 colormap(gray)
ylabel ('Time in seconds')
xlabel('Relative distance in meters')
```

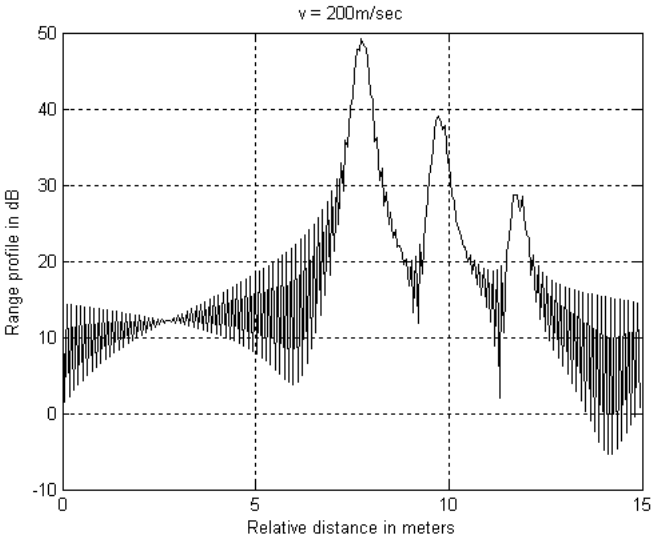


Figure 8.20. Illustration of range profile distortion due to target velocity.

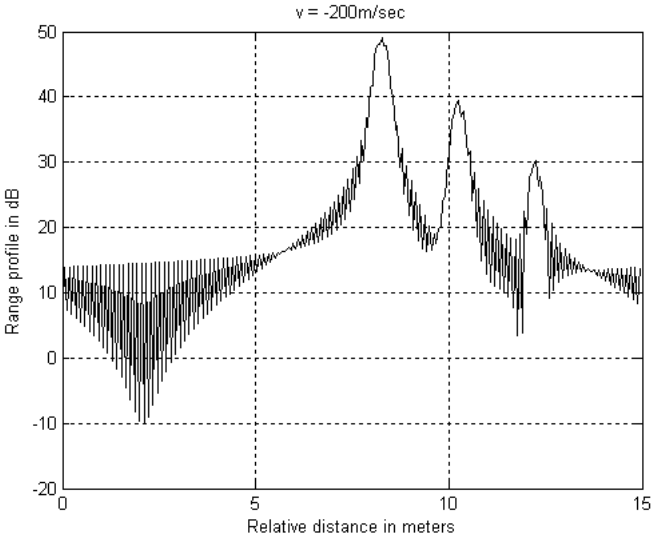


Figure 8.21. Illustration of range profile distortion due to target velocity.

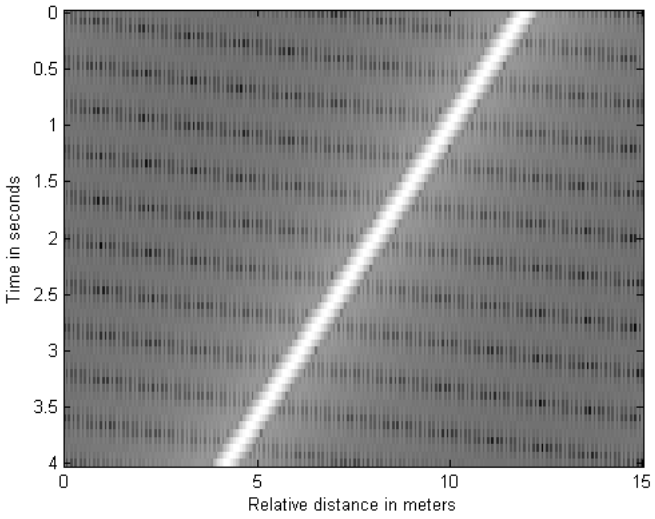


Figure 8.22. Synthesized range profile for a moving target (4 seconds long).

8.6. MATLAB Program Listings

This section presents listings for all the MATLAB programs used to produce all of the MATLAB-generated figures in this chapter.

8.6.1. MATLAB Function "matched_filter.m"

The function *"matched_filter.m"* performs fast convolution processing. The user can access this function either by a MATLAB function call or by executing the MATLAB program *"matched_filter_gui.m,"* which utilizes a MATLAB-based GUI. The work space associated with this program is shown in Fig. 8.23. The outputs for this function include plots of the compressed and uncompressed signals as well as the replica used in the pulse compression process. This function utilizes the function *"power_integer_2.m."*

The function *"matched_filter.m"* syntax is as follows:

[y] = matched_filter(nscat, rrec, taup, b, scat_range, scat_rcs, win)

where

Symbol	Description	Units	Status
nscat	*number of point scatterers within the received window*	*none*	*input*
rrec	*receive window size*	*m*	*input*
taup	*uncompressed pulse width*	*seconds*	*input*
b	*chirp bandwidth*	*Hz*	*input*
scat_range	*vector of scatterers' relative range (within the receive window)*	*m*	*input*
scat_rcs	*vector of scatterers' RCS*	m^2	*input*
win	*0 = no window* *1 = Hamming* *2 = Kaiser with parameter pi* *3 = Chebychev side-lobes at -60dB*	*none*	*input*
y	*normalized compressed output*	*volts*	*output*

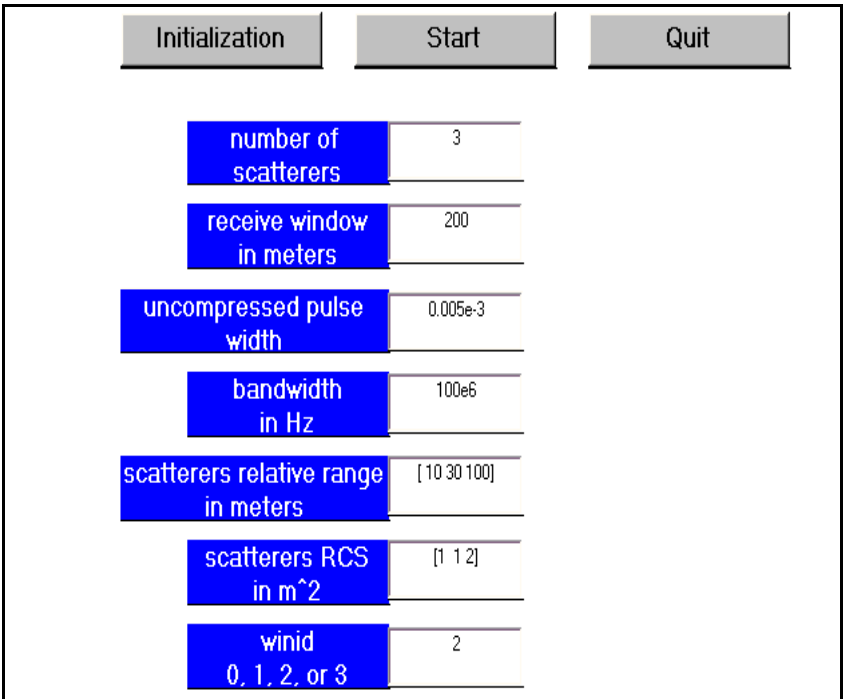


Figure 8.23. GUI workspace associated with the function *"matched_filter_gui.m."*

MATLAB Function *"matched_filter.m"* ***Listing***

```
function [y] = matched_filter(nscat,taup,b,rrec,scat_range,scat_rcs,winid)
eps = 1.0e-16;
% time bandwidth product
time_B_product = b * taup;
if(time_B_product < 5 )
   fprintf('************* Time Bandwidth product is TOO SMALL ***************')
   fprintf('\n Change b and or taup')
  return
end
% speed of light
c = 3.e8;
% number of samples
n = fix(5 * taup * b);
% initialize input, output, and replica vectors
x(nscat,1:n) = 0.;
y(1:n) = 0.;
replica(1:n) = 0.;
% determine proper window
if( winid == 0.)
  win(1:n) = 1.;
end
if(winid == 1.);
   win = hamming(n)';
end
if( winid == 2.)
   win = kaiser(n,pi)';
end
if(winid == 3.)
   win = chebwin(n,60)';
end
% check to ensure that scatterers are within recieve window
index = find(scat_range > rrec);
if (index ~= 0)
   'Error. Receive window is too large; or scatterers fall outside window'
  return
end
% calculate sampling interval
t = linspace(-taup/2,taup/2,n);
replica = exp(i * pi * (b/taup) .* t.^2);
figure(1)
subplot(2,1,1)
plot(t,real(replica))
ylabel('Real (part) of replica')
xlabel('Time in seconds')
grid
subplot(2,1,2)
```

```
sampling_interval = taup / n;
freqlimit = 0.5/ sampling_interval;
freq = linspace(-freqlimit,freqlimit,n);
plot(freq,fftshift(abs(fft(replica))));
ylabel('Spectrum of replica')
xlabel('Frequency in Hz')
grid
 for j = 1:1:nscat
   range = scat_range(j) ;
   x(j,:) = scat_rcs(j) .* exp(i * pi * (b/taup) .* (t +(2*range/c)).^2) ;
   y = x(j,:)  + y;
end
figure(2)
 y = y .* win;
plot(t,real(y),'k')
xlabel ('Relative delay in seconds')
ylabel ('Uncompressed echo')
grid
out =xcorr(replica, y);
out = out ./ n;
s = taup * c /2;
Npoints = ceil(rrec * n /s);
dist =linspace(0, rrec, Npoints);
delr = c/2/b;
figure(3)
plot(dist,abs(out(n:n+Npoints-1)),'k')
xlabel ('Target relative position in meters')
ylabel ('Compressed echo')
grid
return
```

MATLAB Function** "power_integer_2.m" **Listing

```
function n = power_integer_2 (x)
m = 0.;
for j = 1:30
  m = m + 1.;
  delta = x - 2.^m;
  if(delta < 0.)
    n = m;
    return
  else
  end
end
return
```

8.6.2. MATLAB Function "stretch.m"

The function "*stretch.m*" presents a digital implementation of stretch processing. The syntax is as follows:

[y] = stretch (nscat, taup, f0, b, scat_range, rrec, scat_rcs, win)

where

Symbol	Description	Units	Status
nscat	*number of point scatterers within the receive window*	*none*	*input*
taup	*uncompressed pulse width*	*seconds*	*input*
f0	*chirp start frequency*	*Hz*	*input*
b	*chirp bandwidth*	*Hz*	*input*
scat_range	*vector of scatterers' range*	*m*	*input*
rrec	*range receive window*	*m*	*input*
scat_rcs	*vector of scatterers' RCS*	m^2	*input*
win	*0 = no window* *1 = Hamming* *2 = Kaiser with parameter pi* *3 = Chebychev side-lobes at -60dB*	*none*	*input*
y	*compressed output*	*volts*	*output*

The user can access this function either by a MATLAB function call or by executing the MATLAB program "*stretch_gui.m,*" which utilizes MATLAB-based GUI and is shown in Fig. 8.24. The outputs of this function are the complex array *y* and plots of the uncompressed and compressed echo signal versus time.

MATLAB Function** "stretch.m" **Listing

```
function [y] = stretch(nscat, taup, f0, b, scat_range, rrec, scat_rcs, winid)
eps = 1.0e-16;
htau = taup / 2.;
c = 3.e8;
trec = 2. * rrec / c;
n = fix(2. * trec * b);
m = power_integer_2(n);
nfft = 2.^m;
x(nscat,1:n) = 0.;
y(1:n) = 0.;
if( winid == 0.)
  win(1:n) = 1.;
```

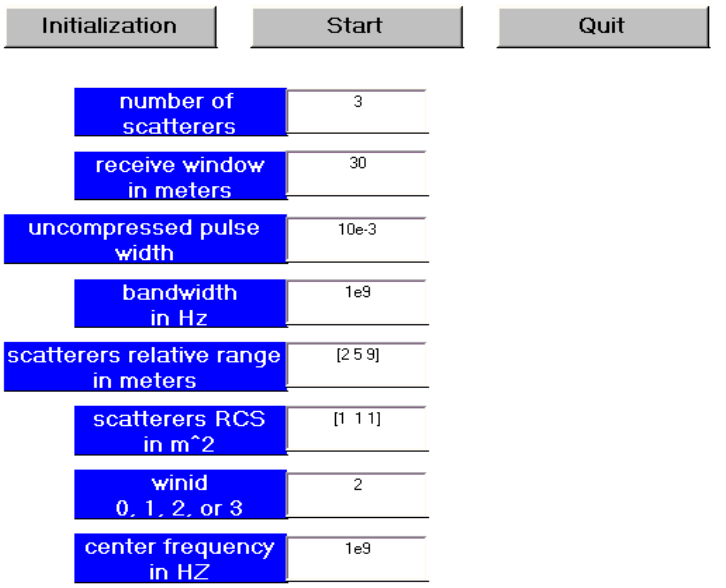


Figure 8.24. GUI workspace associated with the function *"stretch_gui.m."*

```
 win =win';
else
  if(winid == 1.)
    win = hamming(n);
  else
    if( winid == 2.)
      win = kaiser(n,pi);
    else
      if(winid == 3.)
        win = chebwin(n,60);
      end
    end
  end
end
deltar = c / 2. / b;
max_rrec = deltar * nfft / 2.;
maxr = max(scat_range);
if(rrec > max_rrec | maxr >= rrec )
  'Error. Receive window is too large; or scatterers fall outside window'
  return
end
```

```
t = linspace(0,taup,n);
for j = 1:1:nscat
   range = scat_range(j);% + rmin;
  psi1 = 4. * pi * range * f0 / c - ...
    4. * pi * b * range * range / c / c/ taup;
  psi2 = (2*4. * pi * b * range / c / taup) .* t;
  x(j,:) = scat_rcs(j) .* exp(i * psi1 + i .* psi2);
  y = y + x(j,:);
end
figure(1)
plot(t,real(y),'k')
xlabel ('Relative delay in seconds')
ylabel ('Uncompressed echo')
grid
ywin = y .* win';
yfft = fft(y,n) ./ n;
out= fftshift(abs(yfft));
figure(2)
delinc = rrec/ n;
%dist = linspace(-delinc-rrec/2,rrec/2,n);
dist = linspace((-rrec/2), rrec/2,n);
plot(dist,out,'k')
xlabel ('Relative range in meters')
ylabel ('Compressed echo')
axis auto
grid
```

8.6.3. MATLAB Function "SFW.m"

The function *"SFW.m"* computes and plots the range profile for a specific SFW. This function utilizes an Inverse Fast Fourier Transform (IFFT) of a size equal to twice the number of steps. Hamming window of the same size is also assumed. The syntax is as follows:

[hl] = SFW (nscat, scat_range, scat_rcs, n, deltaf, prf, v, r0, winid)

where

Symbol	Description	Units	Status
nscat	*number of scatterers that make up the target*	*none*	*input*
scat_range	*vector containing range to individual scatterers*	*meters*	*input*
scat_rcs	*vector containing RCS of individual scatterers*	*meter square*	*input*
n	*number of steps*	*none*	*input*

Symbol	Description	Units	Status
deltaf	*frequency step*	*Hz*	*input*
prf	*PRF of SFW*	*Hz*	*input*
v	*target velocity*	*meter/second*	*input*
r0	*profile starting range*	*meters*	*input*
winid	*number>0 for Hamming window* *number < 0 for no window*	*none*	*input*
hl	*range profile*	*dB*	*output*

MATLAB Function "SFW.m" Listing

```
function [hl] = SFW (nscat, scat_range, scat_rcs, n, deltaf, prf, v, rnote,winid)
% Range or Time domain Profile
% Range_Profile returns the Range or Time domain plot of a simulated
% HRR SFWF returning from a predetermined number of targets with a predetermined
% RCS for each target.
c=3.0e8;  % speed of light (m/s)
num_pulses  = n;
SNR_dB = 40;
nfft = 256;
% carrier_freq = 9.5e9; %Hz (10GHz)
freq_step    = deltaf; %Hz (10MHz)
V = v;  % radial velocity (m/s)  -- (+)=towards radar (-)=away
PRI = 1. / prf; % (s)
if (nfft > 2*num_pulses)
   num_pulses = nfft/2;
else
end
Inphase = zeros((2*num_pulses),1);
Quadrature = zeros((2*num_pulses),1);
Inphase_tgt    = zeros(num_pulses,1);
Quadrature_tgt = zeros(num_pulses,1);
IQ_freq_domain = zeros((2*num_pulses),1);
Weighted_I_freq_domain = zeros((num_pulses),1);
Weighted_Q_freq_domain = zeros((num_pulses),1);
Weighted_IQ_time_domain = zeros((2*num_pulses),1);
Weighted_IQ_freq_domain = zeros((2*num_pulses),1);
abs_Weighted_IQ_time_domain = zeros((2*num_pulses),1);
dB_abs_Weighted_IQ_time_domain = zeros((2*num_pulses),1);
taur = 2. * rnote / c;
for jscat = 1:nscat
  ii = 0;
  for i = 1:num_pulses
    ii = ii+1;
```

```
    rec_freq = ((i-1)*freq_step);
    Inphase_tgt(ii) = Inphase_tgt(ii) + sqrt(scat_rcs(jscat)) * cos(-2*pi*rec_freq*...
      (2.*scat_range(jscat)/c - 2*(V/c)*((i-1)*PRI + taur/2 + 2*scat_range(jscat)/c)));
            Quadrature_tgt(ii) = Quadrature_tgt(ii) + sqrt(scat_rcs(jscat))*sin(-
2*pi*rec_freq*...
      (2*scat_range(jscat)/c - 2*(V/c)*((i-1)*PRI + taur/2 + 2*scat_range(jscat)/c)));
  end
end
if(winid >= 0)
   window(1:num_pulses) = hamming(num_pulses);
else
   window(1:num_pulses) = 1;
end
Inphase = Inphase_tgt;
Quadrature = Quadrature_tgt;
Weighted_I_freq_domain(1:num_pulses) = Inphase(1:num_pulses).* window';
Weighted_Q_freq_domain(1:num_pulses) = Quadrature(1:num_pulses).* window';
Weighted_IQ_freq_domain(1:num_pulses)= Weighted_I_freq_domain + ...
  Weighted_Q_freq_domain*j;
Weighted_IQ_freq_domain(num_pulses:2*num_pulses)=0.+0.i;
Weighted_IQ_time_domain = (ifft(Weighted_IQ_freq_domain));
abs_Weighted_IQ_time_domain = (abs(Weighted_IQ_time_domain));
dB_abs_Weighted_IQ_time_domain =
20.0*log10(abs_Weighted_IQ_time_domain)+SNR_dB;
% calculate the unambiguous range window size
Ru = c /2/deltaf;
hl = dB_abs_Weighted_IQ_time_domain;
 numb = 2*num_pulses;
delx_meter = Ru / numb;
xmeter = 0:delx_meter:Ru-delx_meter;
plot(xmeter, dB_abs_Weighted_IQ_time_domain,'k')
xlabel ('Relative distance in meters')
ylabel ('Range profile in dB')
grid
```

Chapter 9 Radar Clutter

Clutter is a term used to describe any object that may generate unwanted radar returns that may interfere with normal radar operations. Parasitic returns that enter the radar through the antenna's mainlobe are called main-lobe clutter; otherwise they are called sidelobe clutter. Clutter can be classified into two main categories: surface clutter and airborne or volume clutter. Surface clutter includes trees, vegetation, ground terrain, man-made structures, and sea surface (sea clutter). Volume clutter normally has a large extent (size) and includes chaff, rain, birds, and insects. Surface clutter changes from one area to another, while volume clutter may be more predictable.

Clutter echoes are random and have thermal noise-like characteristics because the individual clutter components (scatterers) have random phases and amplitudes. In many cases, the clutter signal level is much higher than the receiver noise level. Thus, the radar's ability to detect targets embedded in high clutter background depends on the Signal-to-Clutter Ratio (SCR) rather than the SNR.

9.1. Clutter Cross Section Density

Since clutter returns are target-like echoes, the only way a radar can distinguish target returns from clutter echoes is based on the target RCS σ_t and the anticipated clutter RCS σ_c. Clutter RCS can be defined as the equivalent radar cross section attributed to reflections from a clutter area, A_c. The average clutter RCS is given by

$$\sigma_c = \sigma^0 A_c \tag{9.1}$$

where σ^0 is the clutter scattering coefficient, a dimensionless quantity that is often expressed in dB. The equivalent of Eq. (9.1) for volume clutter is

$$\sigma_c = \eta^0 V_w \tag{9.2}$$

where V_w is the clutter volume and η^0 is the volume clutter scattering coefficient. Note that η^0 units are m^{-1}, and because of this, it is typically expressed in dB/meter units.

9.2. Surface Clutter

Surface clutter includes both land and sea clutter, and is often called area clutter. Area clutter manifests itself in airborne radars in the look-down mode. It is also a major concern for ground-based radars when searching for targets at low grazing angles. The grazing angle ψ_g is the angle from the surface of the earth to the main axis of the illuminating beam, as illustrated in Fig. 9.1.

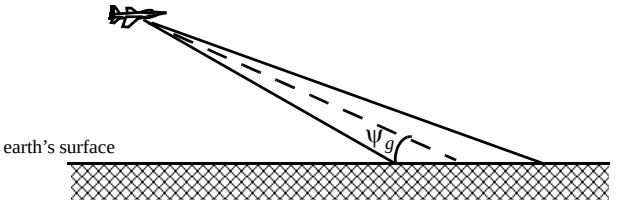


Figure 9.1. Definition of a grazing angle.

Factors that affect the radar performance due to the presence of clutter include clutter reflectivity which is function of radar wavelength, polarization, and of course shape and size of the clutter itself. The amount of clutter RCS in the radar beam depends heavily on the grazing angle, surface roughness, and spatial characteristics of clutter and its time fluctuation characteristics. Typically, the clutter scattering coefficient σ^0 is larger for smaller wavelengths. Figure 9.2 shows a sketch describing the dependency of σ^0 on the grazing angle. Three regions are identified; they are the low grazing angle region, the flat or plateau region, and the high grazing angle region.

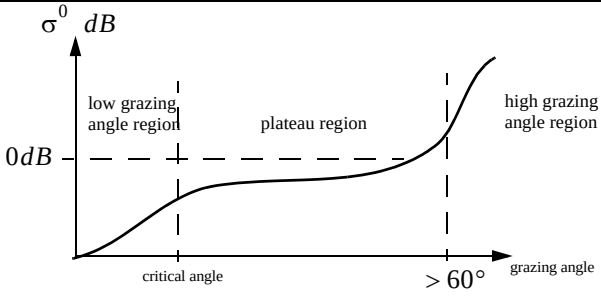


Figure 9.2. Clutter regions.

The low grazing angle region extends from zero to about the critical angle. The critical angle is defined by Rayleigh as the angle below which a surface is considered to be smooth and above which a surface is considered to be rough; Denote the root mean square (rms) of a surface height irregularity as h_{rms}; then according to the Rayleigh criteria, the surface is considered to be smooth if

$$\frac{4\pi h_{rms}}{\lambda}\sin\psi_g < \frac{\pi}{2} \tag{9.3}$$

Consider a wave incident on a rough surface, as shown in Fig. 9.3. Due to surface height irregularity (surface roughness), the rough path is longer than the smooth path by a distance $2h_{rms}\sin\psi_g$. This path difference translates into a phase differential $\Delta\psi$:

$$\Delta\psi = \frac{2\pi}{\lambda}\ 2h_{rms}\sin\psi_g \tag{9.4}$$

The critical angle ψ_{gc} is then computed when $\Delta\psi = \pi$ (first null); thus,

$$\frac{4\pi h_{rms}}{\lambda}\ \sin\psi_{gc} = \pi \tag{9.5}$$

or equivalently,

$$\psi_{gc} = \operatorname{asin}\frac{\lambda}{4h_{rms}} \tag{9.6}$$

In the case of sea clutter, for example, the rms surface height irregularity is

$$h_{rms} \approx 0.025 + 0.046\ \ S_{state}^{1.72} \tag{9.7}$$

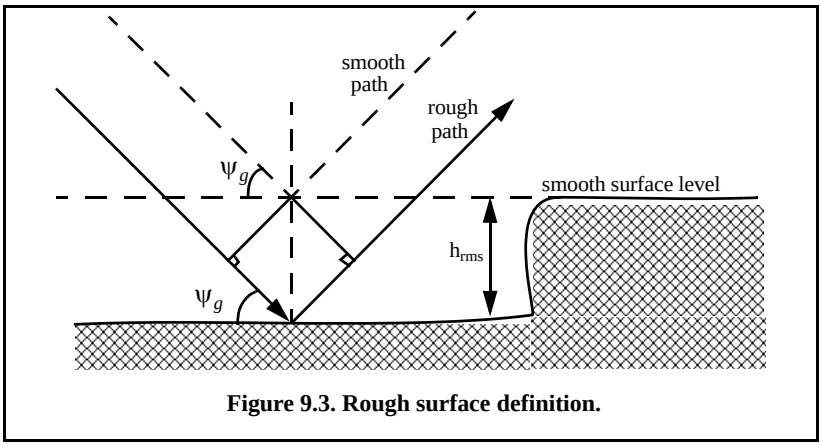


Figure 9.3. Rough surface definition.

where S_{state} is the sea state, which is tabulated in several cited references. The sea state is characterized by the wave height, period, length, particle velocity, and wind velocity. For example, $S_{state} = 3$ refers to a moderate sea state, in which the wave height is approximately $0.9144\ to\ 1.2192\ m$, the wave period 6.5 *to* 4.5 seconds, wave length $1.9812\ to\ 33.528\ m$, wave velocity $20.372\ to\ 25.928\ Km/hr$, and wind velocity $22.224\ to\ 29.632\ Km/hr$.

Clutter at low grazing angles is often referred to as diffuse clutter, where there are a large number of clutter returns in the radar beam (noncoherent reflections). In the flat region the dependency of σ^0 on the grazing angle is minimal. Clutter in the high grazing angle region is more specular (coherent reflections) and the diffuse clutter components disappear. In this region the smooth surfaces have larger σ^0 than rough surfaces, the opposite of the low grazing angle region.

9.2.1. Radar Equation for Surface Clutter

Consider an airborne radar in the look-down mode shown in Fig. 9.4. The intersection of the antenna beam with the ground defines an elliptically shaped footprint. The size of the footprint is a function of the grazing angle and the antenna $3dB$ beamwidth θ_{3dB}, as illustrated in Fig. 9.5. The footprint is divided into many ground range bins each of size $(c\tau/2)\sec\psi_g$, where τ is the pulse width. From Fig. 9.5, the clutter area A_c is

$$A_c \approx R\theta_{3dB}\frac{c\tau}{2}\sec\psi_g \tag{9.8}$$

The power received by the radar from a scatterer within A_c is given by the radar equation as

$$S_t = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4} \tag{9.9}$$

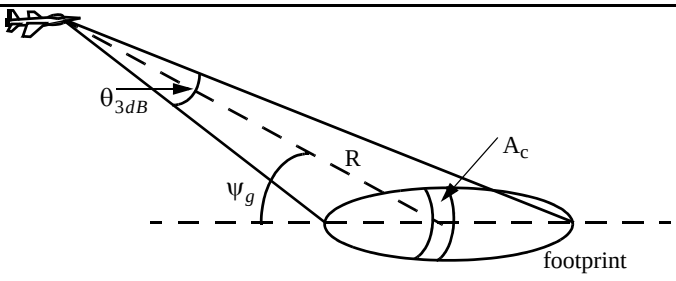


Figure 9.4. Airborne radar in the look-down mode.

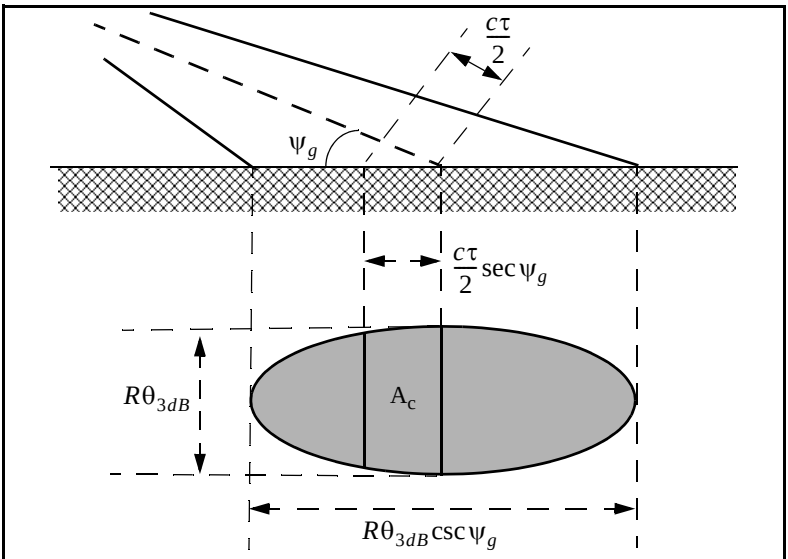


Figure 9.5. Footprint definition.

where, as usual, P_t is the peak transmitted power, G is the antenna gain, λ is the wavelength, and σ_t is the target RCS. Similarly, the received power from clutter is

$$S_C = \frac{P_t G^2 \lambda^2 \sigma_c}{(4\pi)^3 R^4} \tag{9.10}$$

where the subscript C is used for area clutter. Substituting Eq. (9.1) for σ_c into Eq. (9.10), we can then obtain the SCR for area clutter by dividing Eq. (9.9) by Eq. (9.10). More precisely,

$$(SCR)_C = \frac{2\sigma_t \cos\psi_g}{\sigma^0 \theta_{3dB} R c\tau} \tag{9.11}$$

Example:

Consider an airborne radar shown in Fig. 9.4. Let the antenna 3dB beamwidth be $\theta_{3dB} = 0.02 rad$, *the pulse width* $\tau = 2\mu s$, *range* $R = 20 Km$, *and grazing angle* $\psi_g = 20°$. *The target RCS is* $\sigma_t = 1 m^2$. *Assume that the clutter reflection coefficient is* $\sigma^0 = 0.0136$. *Compute the SCR.*

Solution:

The SCR is given by Eq. (9.11) as

$$(SCR)_C = \frac{2\sigma_t \cos\psi_g}{\sigma^0 \theta_{3dB} Rc\tau} \Rightarrow$$

$$(SCR)_C = \frac{(2)(1)(\cos 20°)}{(0.0136)(0.02)(20000)(3\times 10^8)(2\times 10^{-6})} = 5.76\times 10^{-4}$$

It follows that

$$(SCR)_C = -32.4dB$$

Thus, for reliable detection the radar must somehow increase its SCR by at least $(32 + X)dB$*, where* X *is on the order of* 13 *to* $15dB$ *or better.*

9.3. Volume Clutter

Volume clutter has large extents and includes rain (weather), chaff, birds, and insects. The volume clutter coefficient is normally expressed in square meters (RCS per resolution volume). Birds, insects, and other flying particles are often referred to as angle clutter or biological clutter.

Weather or rain clutter can be suppressed by treating the rain droplets as perfect small spheres. We can use the Rayleigh approximation of a perfect sphere to estimate the rain droplets' RCS. The Rayleigh approximation, without regard to the propagation medium index of refraction is

$$\sigma = 9\pi r^2 (kr)^4 \qquad r \ll \lambda \tag{9.12}$$

where $k = 2\pi/\lambda$, and r is radius of a rain droplet.

Electromagnetic waves when reflected from a perfect sphere become strongly co-polarized (have the same polarization as the incident waves). Consequently, if the radar transmits, for example, a right-hand-circular (RHC) polarized wave, then the received waves are left-hand-circular (LHC) polarized because they are propagating in the opposite direction. Therefore, the back-scattered energy from rain droplets retains the same wave rotation (polarization) as the incident wave, but has a reversed direction of propagation. It follows that radars can suppress rain clutter by co-polarizing the radar transmit and receive antennas.

Denote η as RCS per unit resolution volume V_w. It is computed as the sum of all individual scatterers RCS within the volume

$$\sigma_w = \sum_{i=1}^{N} \sigma_i \tag{9.13}$$

where N is the total number of scatterers within the resolution volume. Thus, the total RCS of a single resolution volume is

$$\sigma_W = \sum_{i=1}^{N} \sigma_i V_W \tag{9.14}$$

A resolution volume is shown in Fig. 9.6 and is approximated by

$$V_W \approx \frac{\pi}{8} \theta_a \theta_e R^2 c\tau \tag{9.15}$$

where θ_a and θ_e are, respectively, the antenna azimuth and elevation beamwidths in radians, τ is the pulse width in seconds, c is the speed of light, and R is range.

Consider a propagation medium with an index of refraction m. The ith rain droplet RCS approximation in this medium is

$$\sigma_i \approx \frac{\pi^5}{\lambda^4} K^2 D_i^6 \tag{9.16}$$

where

$$K^2 = \left| \frac{m^2 - 1}{m^2 + 2} \right|^2 \tag{9.17}$$

and D_i is the ith droplet diameter. For example, temperatures between $32°F$ and $68°F$ yield

$$\sigma_i \approx 0.93 \frac{\pi^5}{\lambda^4} D_i^6 \tag{9.18}$$

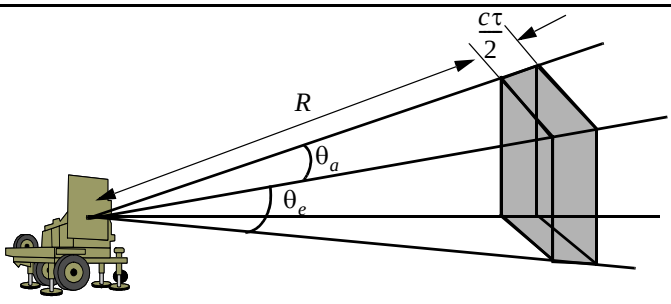


Figure 9.6. Definition of a resolution volume.

and for ice Eq. (9.18) can be approximated by

$$\sigma_i \approx 0.2\frac{\pi^5}{\lambda^4}D_i^6 \tag{9.19}$$

Substituting Eq. (9.19) into Eq. (9.14) yields

$$\sigma_w = \frac{\pi^5}{\lambda^4}K^2 Z \tag{9.20}$$

where the weather clutter coefficient Z is defined as

$$Z = \sum_{i=1}^{N} D_i^6 \tag{9.21}$$

In general, a rain droplet diameter is given in millimeters and the radar resolution volume is expressed in cubic meters; thus the units of Z are often expressed in $millimeter^6/m^3$.

9.3.1. Radar Equation for Volume Clutter

The radar equation gives the total power received by the radar from a σ_t target at range R as

$$S_t = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4} \tag{9.22}$$

where all parameters in Eq. (9.22) have been defined earlier. The weather clutter power received by the radar is

$$S_w = \frac{P_t G^2 \lambda^2 \sigma_w}{(4\pi)^3 R^4} \tag{9.23}$$

It follows that

$$S_w = \frac{P_t G^2 \lambda^2}{(4\pi)^3 R^4}\ \frac{\pi}{8} R^2 \theta_a \theta_e c\tau \sum_{i=1}^{N} \sigma_i \tag{9.24}$$

The SCR for weather clutter is then computed by dividing Eq. (9.22) by Eq. (9.24). More precisely,

$$(SCR)_V = \frac{S_t}{S_w} = (8\sigma_t) \Big/ \left(\pi \theta_a \theta_e c\tau R^2 \sum_{i=1}^{N} \sigma_i \right) \tag{9.25}$$

where the subscript V is used to denote volume clutter.

Example:

A certain radar has target RCS $\sigma_t = 0.1m^2$, *pulse width* $\tau = 0.2\mu s$, *antenna beamwidth* $\theta_a = \theta_e = 0.02 radians$. *Assume the detection range to be* $R = 50Km$, *and compute the SCR if* $\sum \sigma_i = 1.6 \times 10^{-8}(m^2/m^3)$.

Solution:

From Eq. (9.25) we have

$$(SCR)_V = \frac{8\sigma_t}{\pi\theta_a\theta_e c\tau R^2 \sum_N \sigma_i}$$

Substituting the proper values we get

$$(SCR)_V = \frac{(8)(0.1)}{\pi(0.02)^2(3\times10^8)(0.2\times10^{-6})(50\times10^3)^2(1.6\times10^{-8})} = 0.265$$

$$(SCR)_V = -5.76dB.$$

9.4. Clutter RCS

9.4.1. Single Pulse - Low PRF Case

Again the received power from clutter is also calculated using Eq. (9.9). However, in this case the clutter RCS σ_c is computed differently. It is

$$\sigma_c = \sigma_{MBc} + \sigma_{SLc} \qquad \textbf{(9.26)}$$

where σ_{MBc} is the main-beam clutter RCS and σ_{SLc} is the sidelobe clutter RCS, as illustrated in Fig. 9.7.

In order to calculate the total clutter RCS given in Eq. (9.11), one must first compute the corresponding clutter areas for both the main beam and the sidelobes. For this purpose, consider the geometry shown in Fig. 9.8. The angles θ_A *and* θ_E represent the antenna 3-dB azimuth and elevation beamwidths, respectively. The radar height (from the ground to the phase center of the antenna) is denoted by h_r, while the target height is denoted by h_t. The radar slant range is R, and its ground projection is R_g. The range resolution is ΔR and its ground projection is ΔR_g. The main beam clutter area is denoted by A_{MBc} and the sidelobe clutter area is denoted by A_{SLc}.

From Fig. 9.8, the following relations can be derived

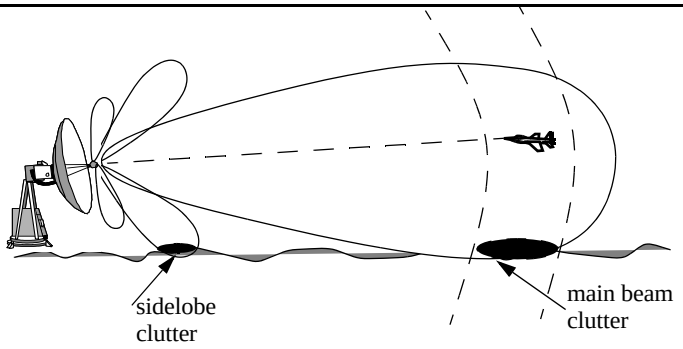


Figure 9.7. Geometry for ground based radar clutter

$$\theta_r = \text{asin}(h_r/R) \tag{9.27}$$

$$\theta_e = \text{asin}((h_t - h_r)/R) \tag{9.28}$$

$$\Delta R_g = \Delta R \cos\theta_r \tag{9.29}$$

where ΔR is the radar range resolution. The slant range ground projection is

$$R_g = R\cos\theta_r \tag{9.30}$$

It follows that the main beam and the sidelobe clutter areas are

$$A_{MBc} = \Delta R_g \; R_g \; \theta_A \tag{9.31}$$

$$A_{SLc} = \Delta R_g \; \pi R_g \tag{9.32}$$

Assume a radar antenna beam $G(\theta)$ of the form

$$G(\theta) = \exp\left(-\frac{2.776\theta^2}{\theta_E^2}\right) \Rightarrow Gaussian \tag{9.33}$$

$$G(\theta) = \left\{ \begin{array}{ll} \left(\dfrac{\sin\left(\dfrac{\theta}{\theta_E}\right)}{\left(\dfrac{\theta}{\theta_E}\right)}\right)^2 & ;|\theta| \le \dfrac{\pi\theta_E}{2.78} \\ 0 & ;elsewhere \end{array} \right\} \Rightarrow \left(\frac{\sin(x)}{x}\right)^2 \tag{9.34}$$

Then the main-beam clutter RCS is

$$\sigma_{MBc} = \sigma^0 A_{MBc} G^2(\theta_e + \theta_r) = \sigma^0 \Delta R_g \; R_g \; \theta_A G^2(\theta_e + \theta_r) \qquad \textbf{(9.35)}$$

and the sidelobe clutter RCS is

$$\sigma_{SLc} = \sigma^0 A_{SLc} (SL_{rms})^2 = \sigma^0 \Delta R_g \; \pi R_g (SL_{rms})^2 \qquad \textbf{(9.36)}$$

where the quantity SL_{rms} is the rms for the antenna sidelobe level.

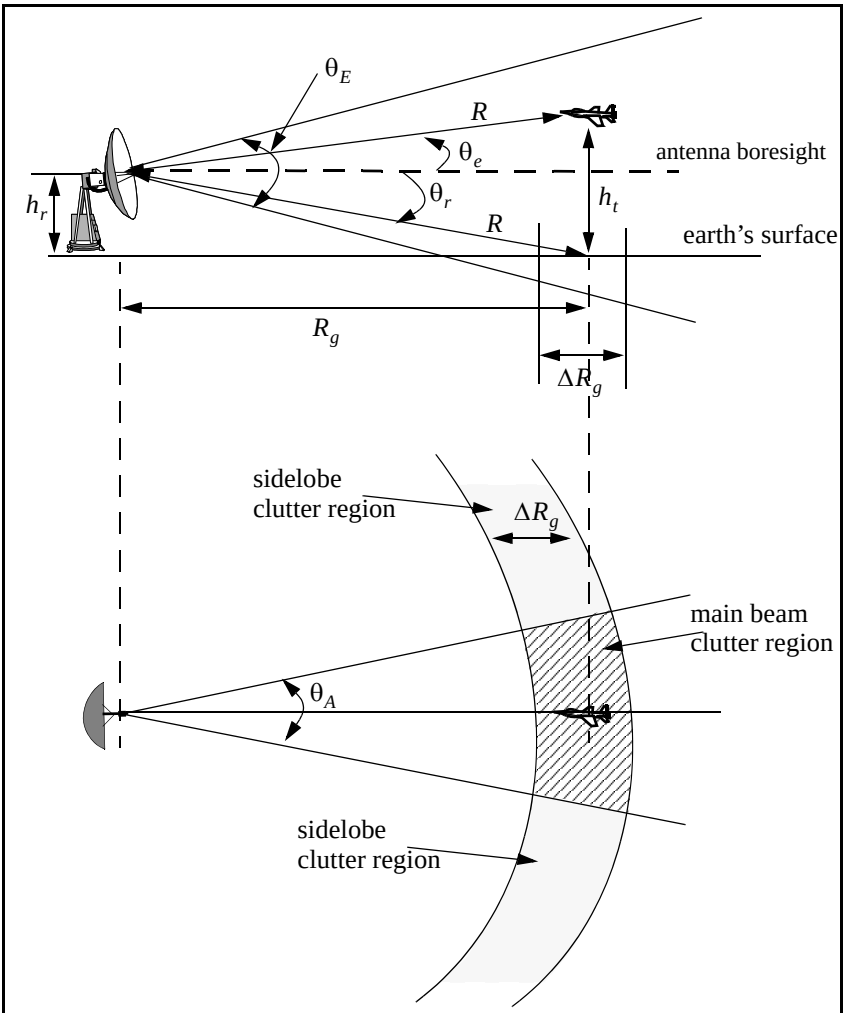


Figure 9.8. Clutter geometry for ground based radar. Side view and top view.

Finally, in order to account for the variation of the clutter RCS versus range, one can calculate the total clutter RCS as a function of range. It is given by

$$\sigma_c(R) = \frac{\sigma_{MBc} + \sigma_{SLc}}{(1 + (R/R_h)^4)} \tag{9.37}$$

where R_h is the radar range to the horizon calculated as

$$R_h = \sqrt{8h_r r_e/3} \tag{9.38}$$

where r_e is the Earth's radius equal to $6371Km$. The denominator in Eq. (9.37) is put in that format in order to account for refraction and for round (spherical) Earth effects.

The radar SNR due to a target at range R is

$$SNR = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4 k T_o B F L} \tag{9.39}$$

where, as usual, P_t is the peak transmitted power, G is the antenna gain, λ is the wavelength, σ_t is the target RCS, k is Boltzmann's constant, T_0 is the effective noise temperature, B is the radar operating bandwidth, F is the receiver noise figure, and L is the total radar losses. Similarly, the Clutter-to-Noise Ratio (CNR) at the radar is

$$CNR = \frac{P_t G^2 \lambda^2 \sigma_c}{(4\pi)^3 R^4 k T_o B F L} \tag{9.40}$$

where the σ_c is calculated using Eq. (9.37).

When the clutter statistic is Gaussian, the clutter signal return and the noise return can be combined, and a new value for determining the radar measurement accuracy is derived from the Signal-to-Clutter+Noise Ratio, denoted by SIR. It is given by

$$SIR = \frac{SNR}{1 + CNR} \tag{9.41}$$

Note that the CNR is computed from Eq. (9.40).

9.4.2. High PRF Case

High PRFs are typically used by pulsed Doppler radars. Pulsed Doppler radars use very short unmodulated train of pulses, and hence, range resolution is limited by the pulsewidth, which forces the radar to use extremely short duration pulses. High PRF radars make up for the loss of average transmitted power due to using short pulses by coherently processing a train of these pulses

within one coherent processing interval (integration time or dwell interval). Although high PRF radars although are ambiguous in range, they provide excellent capability to measuring Doppler frequency. Range ambiguity can be dealt with by using multiple PRF (PRF staggering) which will be addressed later section. One major drawback of using high PRFs (or pulsed Doppler radars) is the fact that pulsed Doppler radars have to contend with much more clutter than do low PRF radars.

Consider the illustrations shown in Fig. 9.9. The low PRF case is shown in Fig. 9.9a. In this case, the target is at maximum detection range which corresponds to an unambiguous range

$$R_u = \frac{cT}{2} = \frac{c}{2f_r} \tag{9.42}$$

where T is the pulse repetition interval and f_r is the radar PRF. The amount of clutter entering the radar through its main-beam corresponds only to the clutter patch located at the target's range. Alternatively, in Fig. 9.9b the high PRF case is depicted. In this case, the radar is range ambiguous and the amount of main-beam clutter entering the radar corresponds to many more clutter patches as shown in Fig. 9.9b. Consequently, the amount of clutter competing with target detection in an order of magnitude larger than the case of low PRF. This is typically referred to as clutter folding.

Denote the clutter power entering the radar due to a single pulse for the target at range R_0 as P_{C_1}, then because of the high PRF operation, the total clutter power entering the radar is

$$P_{C_{folded}} = \sum_{n=0}^{N-1} P_{C_1} Rect\left(\frac{t - nT}{\tau_0}\right) \tag{9.43}$$

where N is the number of pulses in one coherent processing interval (dwell), T is the PRI, and τ_0 is the pulsewidth. Note that since the radar receiver is shut off during transmission of a given pulse, Eq. (9.43) is computed only at delays (range) that correspond to

$$\{(nT + 2\tau_0) < t < (n+1)T - \tau_0;\ 0 \le n \le N-1\} \tag{9.44}$$

where in this case, the transmitter is assumed to be shut off not only during the transmission of each pulse but also for one pulsewidth before and after each transmission. Thus, one would expect the folded clutter RCS to not be continuous versus the range, but rather to exist over intervals of length T seconds with gaps that correspond to three times the pulsewidth. This is illustrated in the following few examples for both low and high PRF cases.

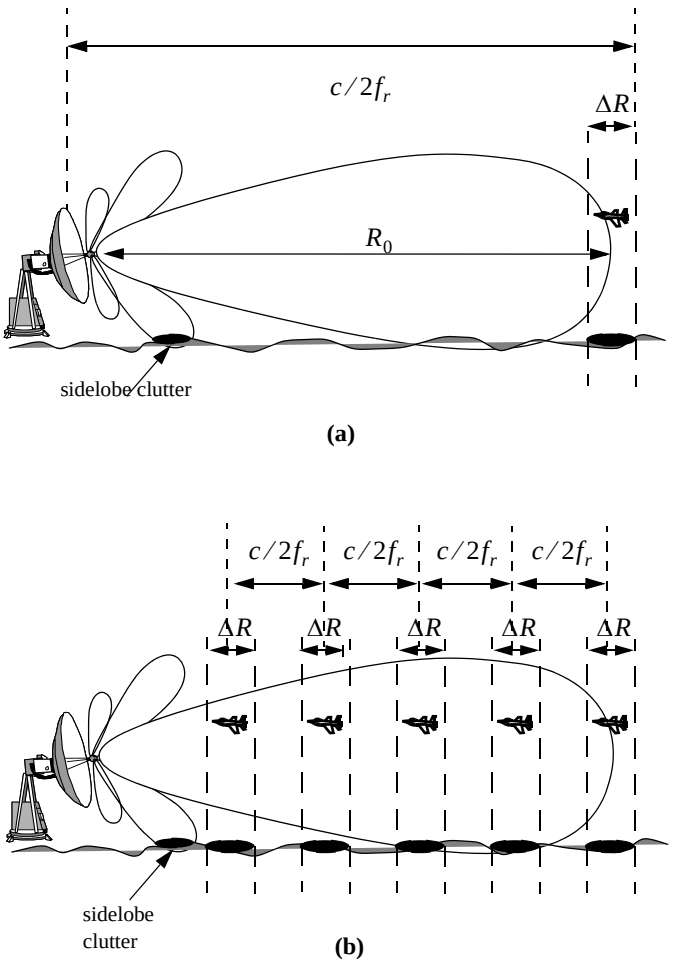


Figure 9.9. Mainbeam clutter entering radar. (a) Low PRF case; (b) high PRF case.

As an example consider the case with the following parameters

clutter back scatterer coefficient	*-20 dB*
antenna 3dB elevation beamwidth	*1.5 degrees*
antenna 3dB azimuth beamwidth	*2 degrees*
antenna sidelobe level	*-25 dB*
radar height	*3 meters*

target height	*150 meters*
radar peak power	*45 KW*
radar operating frequency	*50 KHz*
pulsewidth	*1 micro sec*
effective noise temperature	*290 Kelvins*
noise figure	*6 dB*
radar losses	*10 dB*
target RCS	*-10 dBsm*
radar center frequency	*5 GHz*

Figure 9.10 is concerned with a low PRF case (i.e, single pulse, no clutter folding). Figure 9.10a shows the clutter RCS versus range when a sin(x)/x antenna pattern is used, and Fig. 9.10b shows the resulting SNR, CNR, and SCR. Figure 9.11 is similar to Fig. 9.10 except in this case the antenna has a Gaussian shape. These plots can be reproduced using the following MATLAB code which uses the function *"clutter_rcs.m."*

```
%Use this code to generate Fig. 9.10 and 9.11
clear all;
close all;
k = 1.38e-23; % Boltzman's constant
pt = 45e3;
theta_AZ = 1.5;
theta_EL = 2;
F = 6;
L = 10;
tau = 1e-6;
B = 1/tau;
sigmmat = -10;
sigmma0 = -20;
SL = -25;
hr = 3;
ht = 150;
f0 = 5e9;
lambda = 3e8/f0;
range = linspace(2,50, 120);
[sigmmaC] = clutter_rcs(sigmma0, theta_EL, theta_AZ, SL, range, hr, ht, B,1);
sigmmaC = 10.^(sigmmaC./10);
range_m = 1000 .* range;
F = 10.^(F/10); % noise figure is 6 dB
T0 = 290; % noise temperature 290K
g = 26000 /theta_AZ /theta_EL; % antenna gain
Lt = 10.^(L/10); % total radar losses 13 dB
sigmmat = 10^(sigmmat/10)
```

```
CNR = pt*g*g*lambda^2 .* sigmmaC ./ ((4*pi)^3 .* (range_m).^4 .* k*T0*F*Lt*B); %
CNR
SNR = pt*g*g*lambda^2 .* sigmmat ./ ((4*pi)^3 .* (range_m).^4 .* k*T0*F*L*B); %
SNR
SCR = SNR ./ CNR; % Signal to clutter ratio
SIR = SNR ./ (1+CNR); % Signal to interference ratio
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
subplot(3,1,1)
plot(range,10*log10(SNR));
ylabel('SNR in dB');
grid on;
axis tight
subplot(3,1,2)
plot(range,10*log10(CNR));
ylabel('CNR in dB');
grid on;
axis tight
subplot(3,1,3)
plot(range,10*log10(SCR));
ylabel('SCR in dB') ;
grid on;
axis tight
xlabel('Range in Km')
```

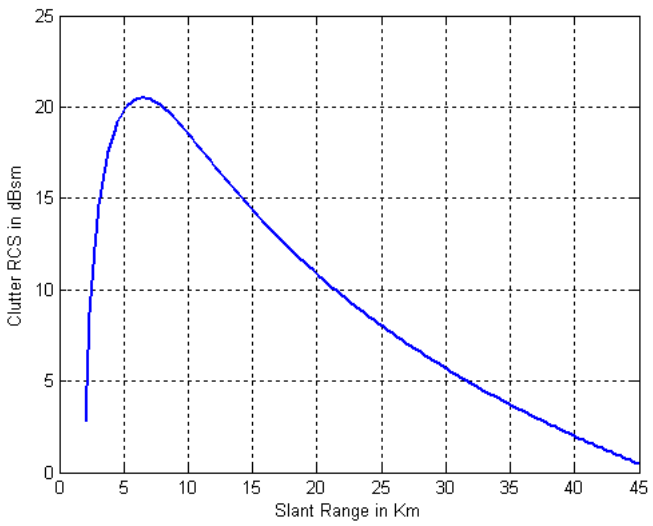


Figure 9.10a. Clutter RCS versus range with sin(x)/x antenna pattern. Single pulse case.

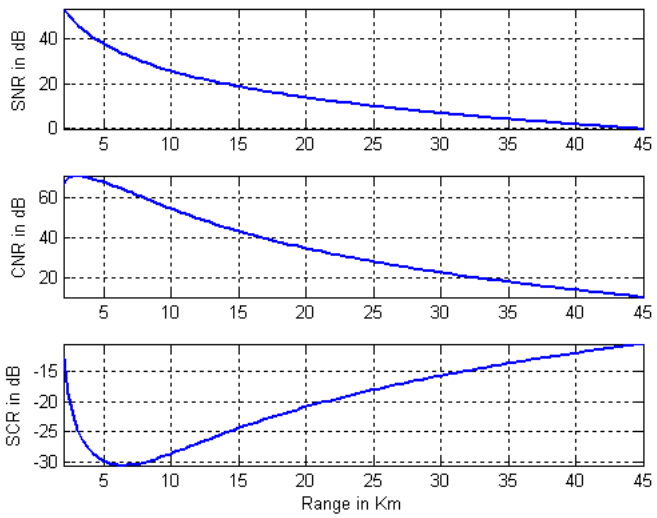


Figure 9.10b. SNR, CNR, and SCR corresponding to Fig. 9.10a.

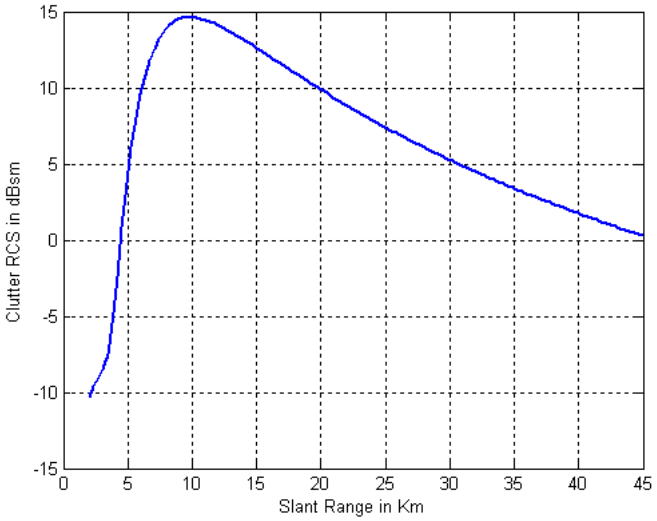


Figure 9.11a. Clutter RCS versus range with Gaussian antenna pattern. Single pulse case.

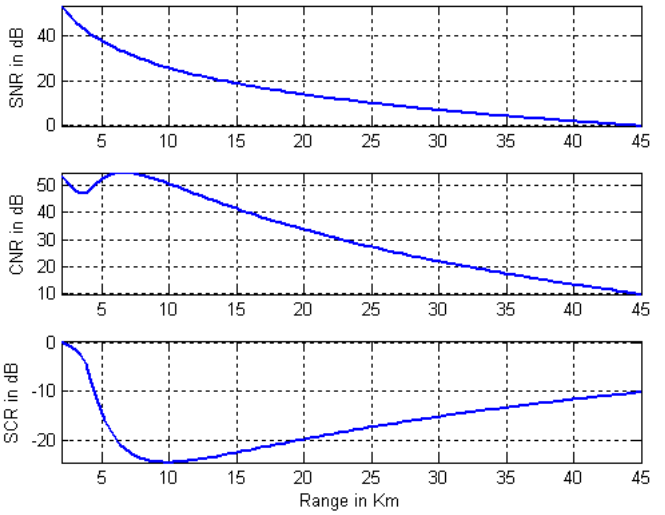


Figure 9.11b. SNR, CNR, and SCR corresponding to Fig. 9.11a.

Figure 9.12 shows the SNR, CNR, and SCR for the high PRF case (i.e, pulse Doppler radar, clutter folding). In this figure the antenna pattern has a sin(x)/x shape. Figure 9.13 is similar to Fig. 9.12 except in this case the antenna pattern is Gaussian. These plots can be reproduced using the following MATLAB code.

```
% Use this code to generate Fig. 9.12 or 9. 13 of text
clear all
close all
k = 1.38e-23; % Boltzmann's constant
T0 = 290; % degrees Kelvin
ant_id = 1; % use 1 for sin(x)/x antenna pattern and use 2 for Gaussian pattern
theta_ref = 0.75; % reference angle of radar antenna in degrees
re = 6371000 * 4 /3; % 4/3rd earth radius in Km
c = 3e8; % speed of light
theta_EL = 1.5; % Antenna elevation beamwidth in degrees
theta_AZ = 2.; % Antenna azimuth beamwidth in degrees
SL_dB = -25; % Antenna RMS sidelobe level
hr = 3; % Radar antenna height in meters
ht = 150; % Target height in meters
Sigmmat = -10; % Target RCS in dB
Sigmma0 = -20; % Clutter backscatter coefficient
P = 45e3; % Radar peak power in Watts
tau = 1e-6; % Pulse width (unmodulated)
```

```
fr = 50e3; % PRF in Hz
f0 = 5e9; % Radar center frequency
F = 6; % Noise figure in dB
L = 10; % Radar losses in dB
lambda = c /f0;
SL = 10^(SL_dB/10);
sigmma0 = 10^(Sigmma0/10);
F = 10^(F/10);
L = L^(L/10);
sigmmat = 10^(Sigmmat/10);
T = 1/fr; % PRI
B = 1/tau; % Bandwidth
delr = c * tau /2; % Range resolution;
Rh = sqrt(2*re*hr); % Range to Horizon
R1 = [2*delr:delr:c/2*(T-tau)];
Rclut = sqrt(R1.^2 + hr^2); % Range to clutter patches
G = 26000 /theta_EL /theta_AZ; % Antenna gain
for j = 0:40
   Rtgt = [c/2*(j*T+2*tau):delr:c/2*((j+1)*T-tau)];
   thetaR = asin(hr./Rclut); % Ele angle from radar to clutter patch target is present
   thetae = theta_ref *pi/180;
   d = Rclut .* cos(thetaR); % Ground range to center of clutter at range Rclut
   del_d = delr .* cos(thetaR);
   % claculte clutter RCS
   theta_sum = thetaR+thetae;
   if(ant_id ==1) % use sinc^2 antenna pattern
      ant_arg = ( theta_sum ) ./ (pi*theta_EL/180);
      gain = (sinc(ant_arg)).^2;
   else
      gain = exp(-2.776 .*(theta_sum./(pi*theta_EL/180)).^2);
   end
   % clutter RCS
    sigmmac = (pi*SL^2+(theta_AZ*pi/180).*gain.*sigmma0.*d.*del_d) ./ (1+(Rclut/
Rh).^4);
    CNR = P*G*G*lambda^2 .* sigmmac ./ ((4*pi)^3 .* Rclut.^4 .* k*T0*F*L*B); %
CNR
   SNR = P*G*G*lambda^2 .* sigmmat ./ ((4*pi)^3 .* Rtgt.^4 .* k*T0*F*L*B); % SNR
   SCR = SNR ./ CNR; % Signal to clutter ratio
   SIR = SNR ./ (1+CNR); % Signal to interfernce ratio
   figure(2)
   subplot(4,1,1),
   hold on
   plot(Rtgt/1000,10*log10(SNR));
   ylabel('SNR - dB');
  grid on
  subplot(4,1,2),
  hold on
   plot(Rtgt/1000,10*log10(CNR));
```

```
    ylabel('CNR - dB');
    grid on
    subplot(4,1,3),
    hold on
    plot(Rtgt/1000,10*log10(SCR));
    ylabel('SCR - dB') ;
    grid on
    subplot(4,1,4),
    hold on
    plot(Rtgt/1000,10*log10(SIR));
    xlabel('Range - Km')
    ylabel('SIR - dB');
   grid on
end
subplot(4,1,1)
axis([0 50 -10 100])
subplot(4,1,2)
axis([0 50 60 90]);
subplot(4,1,3)
axis([0 50 -100 0])
subplot(4,1,4)
axis([0 50 -100 0])
```

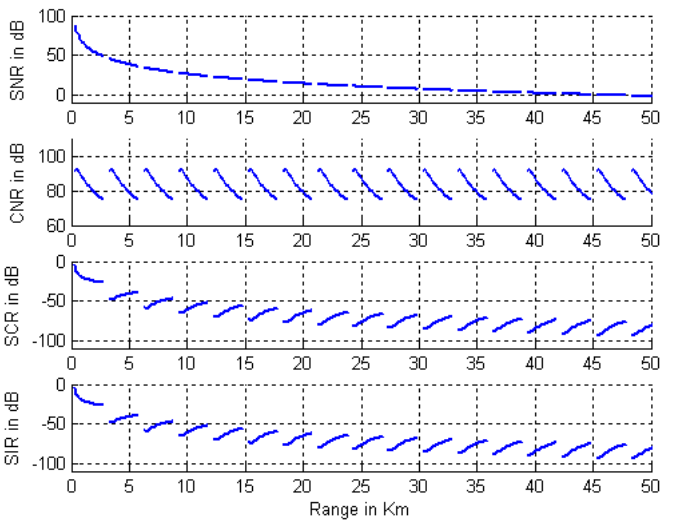


Figure 9.12. SIR, SCR, CNR, and SNR for a pulse Doppler radar with sin(x)/x antenna pattern.

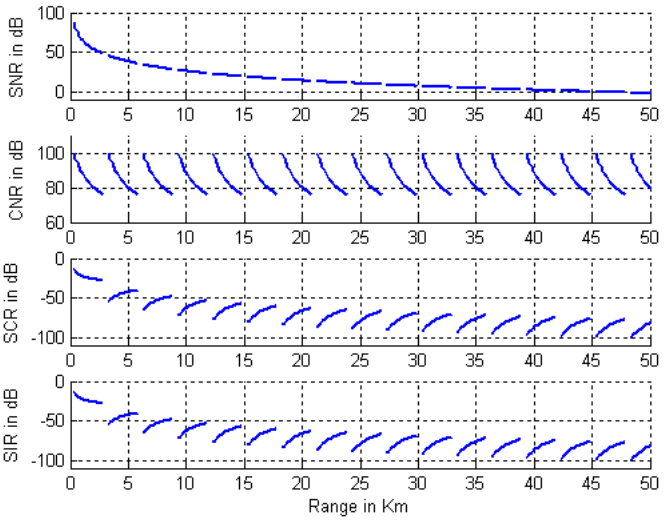


Figure 9.13. SIR, SCR, CNR, and SNR for a pulse Doppler radar with Gaussian antenna pattern.

9.5. Clutter Spectrum

9.5.1. Clutter Statistical Models

Since clutter within a resolution cell or volume is composed of a large number of scatterers with random phases and amplitudes, it is statistically described by a probability distribution function. The type of distribution depends on the nature of clutter itself (sea, land, volume), the radar operating frequency, and the grazing angle.

If sea or land clutter is composed of many small scatterers when the probability of receiving an echo from one scatterer is statistically independent of the echo received from another scatterer, then the clutter may be modeled using a Rayleigh distribution,

$$f(x) = \frac{2x}{x_0}\exp\left(\frac{-x^2}{x_0}\right) \quad ; \quad x \geq 0 \tag{9.45}$$

where x_0 is the mean-squared value of x.

The log-normal distribution best describes land clutter at low grazing angles. It also fits sea clutter in the plateau region. It is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}\ x}\ \exp\left(-\frac{(\ln x - \ln x_m)^2}{2\sigma^2}\right)\ ;\ x > 0 \qquad \textbf{(9.46)}$$

where x_m is the median of the random variable x, and σ is the standard deviation of the random variable $\ln(x)$.

The Weibull distribution is used to model clutter at low grazing angles (less than five degrees) for frequencies between 1 and $10GHz$. The Weibull probability density function is determined by the Weibull slope parameter a (often tabulated) and a median scatter coefficient $\bar{\sigma}_0$, and is given by

$$f(x) = \frac{bx^{b-1}}{\bar{\sigma}_0}\ \exp\left(-\frac{x^b}{\bar{\sigma}_0}\right);\ x \geq 0 \qquad \textbf{(9.47)}$$

where $b = 1/a$ is known as the shape parameter. Note that when $b = 2$ the Weibull distribution becomes a Rayleigh distribution.

9.5.2. Clutter Components

It was established earlier that the complex envelope of the signal received by the radar comprise the target returns and additive bandlimited white noise. In the presence of clutter, the complex envelope is now composed of target, noise, and clutter returns. That is,

$$\tilde{x}(t) = \tilde{s}(t) + \tilde{n}(t) + \tilde{w}(t) \qquad \textbf{(9.48)}$$

where $\tilde{s}(t)$, $\tilde{n}(t)$, and $\tilde{w}(t)$ are, respectively, the target, noise, and clutter complex envelope echoes. Noise is typically modeled (as discussed in earlier chapters) as a bandlimited white Gaussian random process. Furthermore, noise samples are consider statistically independent of each other and of clutter measurements.

Clutter arises from reflections of unwanted objects within the radar beam. Since many objects comprose the clutter returns, clutter may also be molded as a Gaussian random process. In other words, clutter samples from one radar measurement to another constitute a joint set of Gaussian random variables. However, because of the clutter fluctuation and due to antenna mechanical scanning, wind speed, and radar platform motion (if applicable), these random variables are not statistically independent.

More precisely, because of the antenna mechanical scanning, clutter returns in the radar mainbeam do not have the same amplitude from pulse to pulse. This will effectively add amplitude modulation to the clutter returns. This additional modulation is governed by the shape of the antenna pattern, the rate of mechanical scanning, and the radar PRF. Denote the antenna two-way azimuth $3dB$ beamwidth as θ_a and the antenna scan rate as $\dot{\theta}_{scan}$. It follows that the

contribution of antenna scanning to the standard deviation of the clutter fluctuation is

$$\sigma_s = 0.399\frac{\dot{\theta}_{scan}}{\theta_a} \tag{9.49}$$

Another contributor to the clutter spectral spreading is caused by motion of the clutter itself, due to wind. Trees, vegetation, and sea waves are the main contributors to this effect. This relative motion, although relatively small, introduces additional Doppler shift in the clutter returns. Earlier, it was established that Doppler frequency due to a relative velocity v is given by

$$f_d = 2v/\lambda \tag{9.50}$$

where λ is the radar operating wavelength. It follows that if the apparent rms velocity due to wind is v_{rms}, then the standard deviation is

$$\sigma_w = 2v_{rms}/\lambda \tag{9.51}$$

Finally, if the radar platform is in motion, then the relative motion between the platform and the stationary clutter will cause a Doppler shift given by

$$f_c = (2v_{radar}\cos\theta)/\lambda \tag{9.52}$$

where $v_{radar}\cos\theta$ is the radial velocity component of the platform in the direction of clutter. Since the radar beam has a finite width, not all clutter components have the same radial velocity at all times. More specifically, if the angles θ_1 and θ_2 represent the edges of the radar beam, then Eq. (9.52) ca be written as

$$f_c = \frac{2v_{radar}}{\lambda}(\cos\theta_2 - \cos\theta_1) \approx \frac{2v_{radar}}{\lambda}\theta_a \sin\theta \tag{9.53}$$

and the standard deviation due to platform motion is given by

$$\sigma_v = \frac{v_{radar}}{\lambda}\sin\theta \tag{9.54}$$

Finally, the overall clutter spreading is denoted by σ_f, where

$$\sigma_f^2 = \sigma_v^2 + \sigma_s^2 + \sigma_w^2 \tag{9.55}$$

The overall value of the clutter spreading defined in Eq. (9.55) is relatively small.

9.5.3. Clutter Power Spectrum Density

Clutter primarily comprises stationary ground unwanted reflections with limited relative motion with respect to the radar. Therefore, its power spectrum density will be concentrated around $f = 0$. However, because σ_f (see Eq. (9.55)) is not always zero, clutter actually exhibits some Doppler frequency spread. The clutter power spectrum can be written as the sum of fixed (stationary) and random (due to frequency spreading) components, as

$$S_c(f) = \frac{P_c}{T\sigma_f\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(f-k/T)^2}{2\sigma_f^2}\right) \tag{9.56}$$

where T is the PRI (i.e., $1/f_r$, f_r is the PRF), P_c is the clutter power or clutter mean square value, and σ_f is the clutter spectral spreading parameter as defined in Eq. (9.55). As clearly indicated by Eq. (9.56), the clutter PSD is periodic with period equal to f_r. Furthermore, the clutter PSD extends about each multiple integer of the PRF in accordance with Eq. (9.55). It must be noted that this spread is relatively small and thus the relation $\sigma_f \ll f_r$ is always true. This is illustrated in Fig. 9.14. The mean square value can be calculated from

$$P_c = T \int_{-f_r/2}^{f_r/2} S_c(f)df \tag{9.57}$$

Let $S_{c0}(f)$ denote the central portion of Eq. (9.56); then P_c is be expressed by

$$P_c = T \int_{-\infty}^{\infty} S_{c0}(f)df \tag{9.58}$$

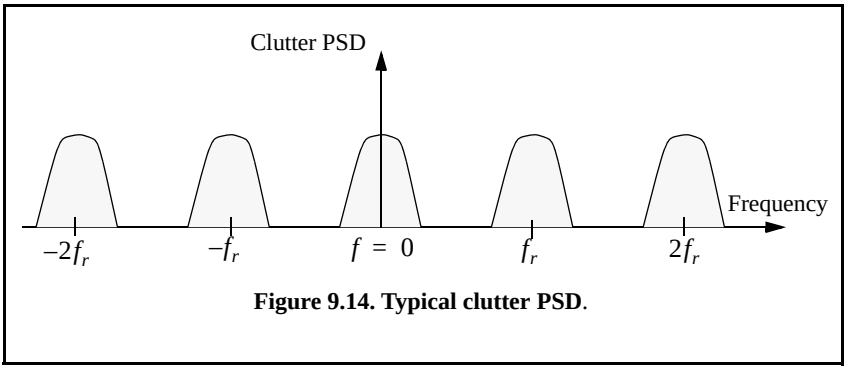


Figure 9.14. Typical clutter PSD.

where $S_{c0}(f)$ is a Gaussian shape function given by

$$S_{c0}(f) = \frac{k}{\sigma_f\sqrt{2\pi}} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) \tag{9.59}$$

and $k = P_c/T$.

9.6. Moving Target Indicator (MTI)

The clutter spectrum is concentrated around DC ($f = 0$) and multiple integers of the radar PRF f_r, as was illustrated in Fig. 9.14. In CW radars, clutter is avoided or suppressed by ignoring the receiver output around DC, since most of the clutter power is concentrated about the zero frequency band. Pulsed radar systems may utilize special filters that can distinguish between slow-moving or stationary targets and fast-moving ones. This class of filter is known as the Moving Target Indicator (MTI). In simple words, the purpose of an MTI filter is to suppress target-like returns produced by clutter and allow returns from moving targets to pass through with little or no degradation. In order to effectively suppress clutter returns, an MTI filter needs to have a deep stop-band at DC and at integer multiples of the PRF. Figure 9.15b shows a typical sketch of an MTI filter response, while Fig. 9.15c shows its output when the PSD shown in Fig. 9.15a is the input.

MTI filters can be implemented using delay line cancelers. As we will show later in this chapter, the frequency response of this class of MTI filter is periodic, with nulls at integer multiples of the PRF. Thus, targets with Doppler frequencies equal to nf_r are severely attenuated. Since Doppler is proportional to target velocity ($f_d = 2v/\lambda$), target speeds that produce Doppler frequencies equal to integer multiples of f_r are known as blind speeds. More precisely,

$$v_{blind} = (n\lambda f_r)/2; \quad n \geq 0 \tag{9.60}$$

Radar systems can minimize the occurrence of blind speeds either by employing multiple PRF schemes (PRF staggering) or by using high PRFs in which the radar may become range ambiguous. The main difference between PRF staggering and PRF agility is that the pulse repetition interval (within an integration interval) can be changed between consecutive pulses for the case of PRF staggering.

9.6.1. Single Delay Line Canceler

A single delay line canceler can be implemented as shown in Fig. 9.16. The canceler's impulse response is denoted as $h(t)$. The output $y(t)$ is equal to the convolution between the impulse response $h(t)$ and the input $x(t)$. The single delay canceler is often called a two-pulse canceler since it requires two distinct input pulses before an output can be read.

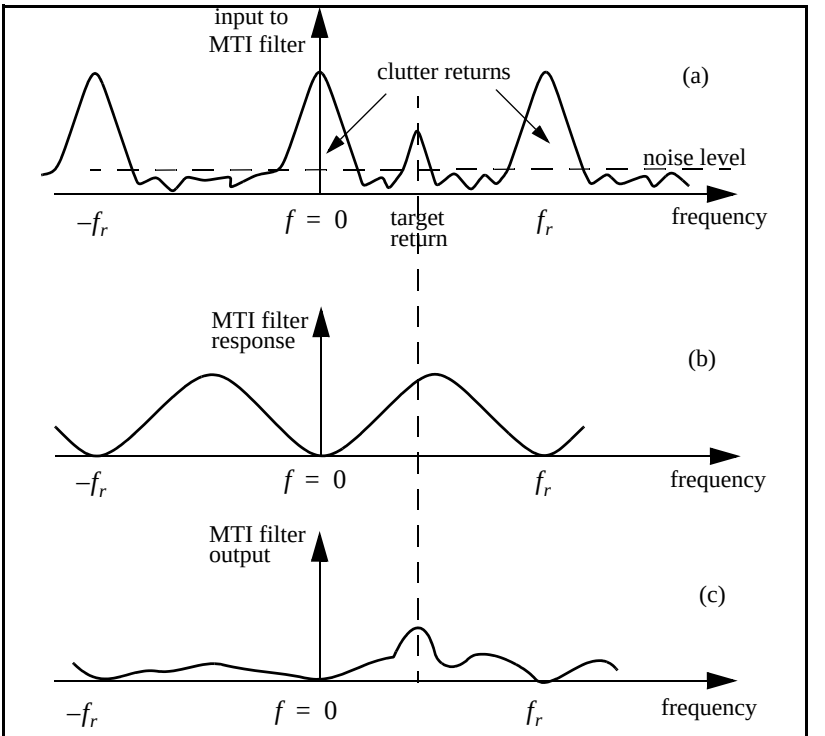


Figure 9.15. (a) Typical radar return PSD when clutter and target are present. (b) MTI filter frequency response. (c) Output from an MTI filter.

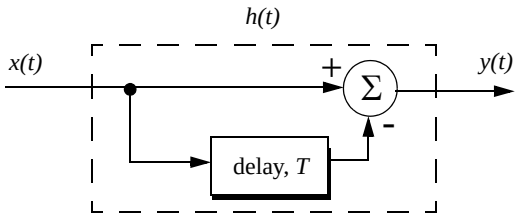


Figure 9.16. Single delay line canceler.

The delay T is equal to the radar PRI ($1/f_r$). The output signal $y(t)$ is

$$y(t) = x(t) - x(t - T) \tag{9.61}$$

The impulse response of the canceler is given by

$$h(t) = \delta(t) - \delta(t - T) \tag{9.62}$$

where $\delta(\ )$ is the delta function. It follows that the Fourier transform (FT) of $h(t)$ is

$$H(\omega) = 1 - e^{-j\omega T} \tag{9.63}$$

where $\omega = 2\pi f$. In the z-domain, the single delay line canceler response is

$$H(z) = 1 - z^{-1} \tag{9.64}$$

The power gain for the single delay line canceler is given by

$$|H(\omega)|^2 = H(\omega)H^*(\omega) = (1 - e^{-j\omega T})(1 - e^{j\omega T}) \tag{9.65}$$

It follows that

$$|H(\omega)|^2 = 1 + 1 - (e^{j\omega T} + e^{-j\omega T}) = 2(1 - \cos\omega T) \tag{9.66}$$

and using the trigonometric identity $(2 - 2\cos 2\vartheta) = 4(\sin\vartheta)^2$ yields

$$|H(\omega)|^2 = 4(\sin(\omega T/2))^2 \tag{9.67}$$

The amplitude frequency response for a single delay line canceller is shown in Fig. 9.17. Clearly, the frequency response of a single canceler is periodic with a period equal to f_r. The peaks occur at $f = (2n + 1)/(2f_r)$, and the nulls are at $f = nf_r$, where $n \geq 0$. In most radar applications the response of a single canceler is not acceptable since it does not have a wide notch in the stop-band. A double delay line canceler has better response in both the stop- and pass-bands, and thus it is more frequently used than a single canceler. In this book, we will use the names *single delay line canceler* and *single canceler* interchangeably.

9.6.2. Double Delay Line Canceler

Two basic configurations of a double delay line canceler are shown in Fig. 9.18. Double cancelers are often called three-pulse cancelers since they require three distinct input pulses before an output can be read. The double line canceler impulse response is given by

$$h(t) = \delta(t) - 2\delta(t - T) + \delta(t - 2T) \tag{9.68}$$

Again, the names *double delay line canceler* and *double canceler* will be used interchangeably. The power gain for the double delay line canceler is

$$|H(\omega)|^2 = |H_1(\omega)|^2 |H_1(\omega)|^2 \tag{9.69}$$

where $|H_1(\omega)|^2$ is the single line canceler power gain given in Eq. (9.55). It follows that

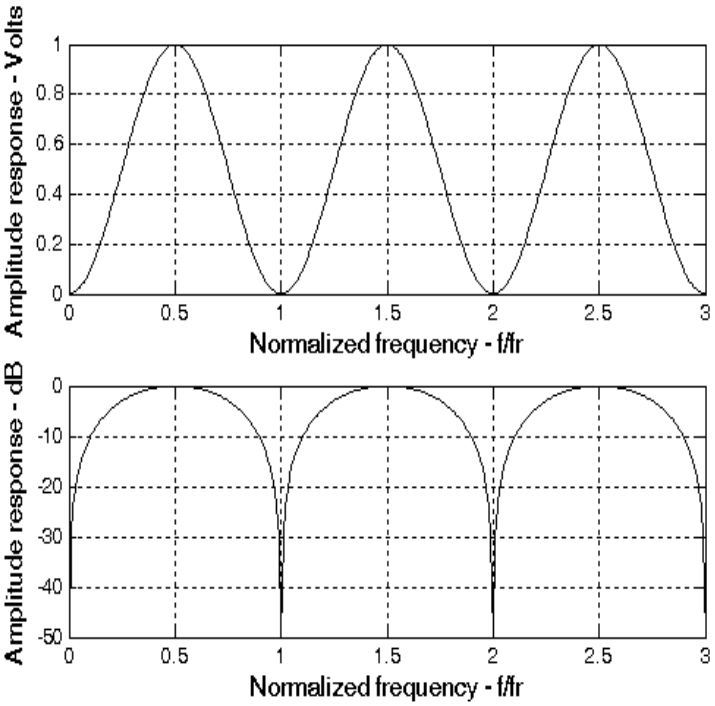


Figure 9.17. Single canceler frequency response.

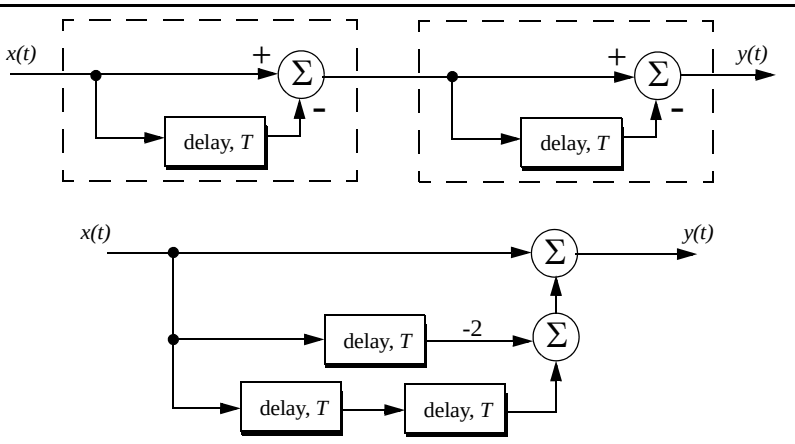


Figure 9.18. Two configurations for a double delay line canceler.

$$|H(\omega)|^2 = 16\left(\sin\left(\omega\frac{T}{2}\right)\right)^4 \quad \textbf{(9.70)}$$

And in the z-domain, we have

$$H(z) = (1-z^{-1})^2 = 1-2z^{-1}+z^{-2} \quad \textbf{(9.71)}$$

Figure 9.19 shows typical output from this function. Note that the double canceler has a better response than the single canceler (deeper notch and flatter pass-band response).

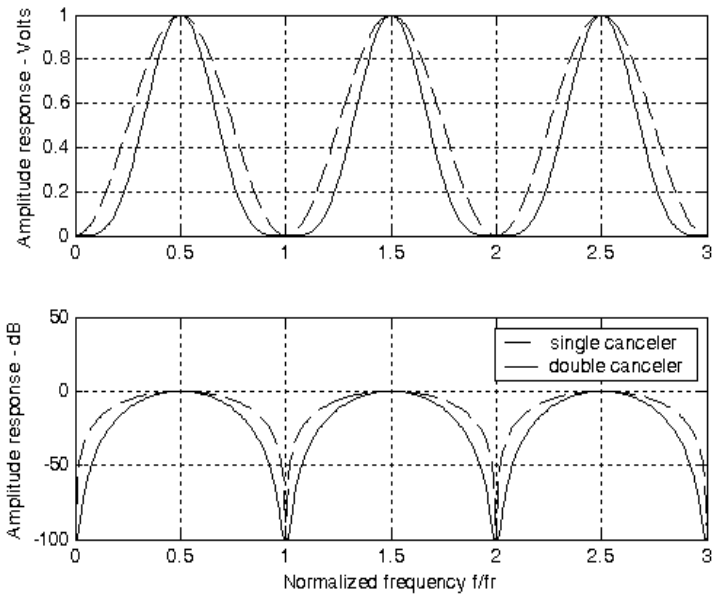


Figure 9.19. Normalized frequency responses for single and double cancelers.

9.6.3. Delay Lines with Feedback (Recursive Filters)

Delay line cancelers with feedback loops are known as recursive filters. The advantage of a recursive filter is that through a feedback loop, we will be able to shape the frequency response of the filter. As an example, consider the single canceler shown in Fig. 9.20. From the figure we can write

$$y(t) = x(t) - (1-K)w(t) \quad \textbf{(9.72)}$$

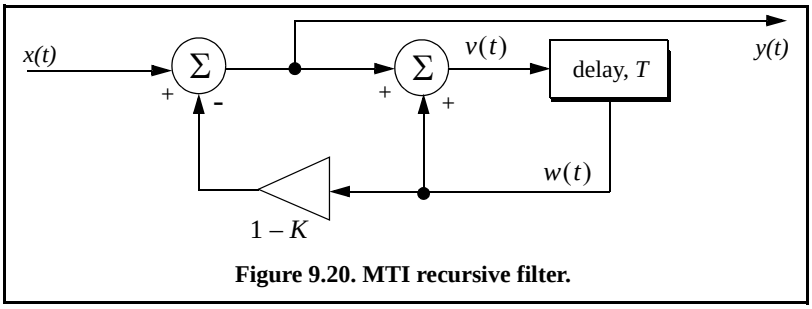


Figure 9.20. MTI recursive filter.

$$v(t) = y(t) + w(t) \tag{9.73}$$

$$w(t) = v(t - T) \tag{9.74}$$

Applying the z-transform to the above three equations yields

$$Y(z) = X(z) - (1 - K)W(z) \tag{9.75}$$

$$V(z) = Y(z) + W(z) \tag{9.76}$$

$$W(z) = z^{-1}V(z) \tag{9.77}$$

Solving for the transfer function $H(z) = Y(z)/X(z)$ yields

$$H(z) = \frac{1 - z^{-1}}{1 - Kz^{-1}} \tag{9.78}$$

The modulus square of $H(z)$ is then equal to

$$|H(z)|^2 = \frac{(1 - z^{-1})(1 - z)}{(1 - Kz^{-1})(1 - Kz)} = \frac{2 - (z + z^{-1})}{(1 + K^2) - K(z + z^{-1})} \tag{9.79}$$

Using the transformation $z = e^{j\omega T}$ yields

$$z + z^{-1} = 2\cos\omega T \tag{9.80}$$

Thus, Eq. (9.79) can now be rewritten as

$$\left|H(e^{j\omega T})\right|^2 = \frac{2(1 - \cos\omega T)}{(1 + K^2) - 2K\cos(\omega T)} \tag{9.81}$$

Note that when $K = 0$, Eq. (9.81) collapses to Eq. (9.67) (single line canceler). Figure 9.21 shows a plot of Eq. (9.81) for $K = 0.25, 0.7, 0.9$. Clearly, by changing the gain factor K one can control the filter response. This plot can be reproduced using the following MATLAB code.

```
clear all;
fofr = 0:0.001:1;
arg = 2.*pi.*fofr;
nume = 2.*(1.-cos(arg));
den11 = (1. + 0.25 * 0.25);
den12 = (2. * 0.25) .* cos(arg);
den1 = den11 - den12;
den21 = 1.0 + 0.7 * 0.7;
den22 = (2. * 0.7) .* cos(arg);
den2 = den21 - den22;
den31 = (1.0 + 0.9 * 0.9);
den32 = ((2. * 0.9) .* cos(arg));
den3 = den31 - den32;
resp1 = nume ./ den1;
resp2 = nume ./ den2;
resp3 = nume ./ den3;
plot(fofr,resp1,'k',fofr,resp2,'k-.',fofr,resp3,'k--');
xlabel('Normalized frequency')
ylabel('Amplitude response')
legend('K=0.25','K=0.7','K=0.9')
grid
axis tight
```

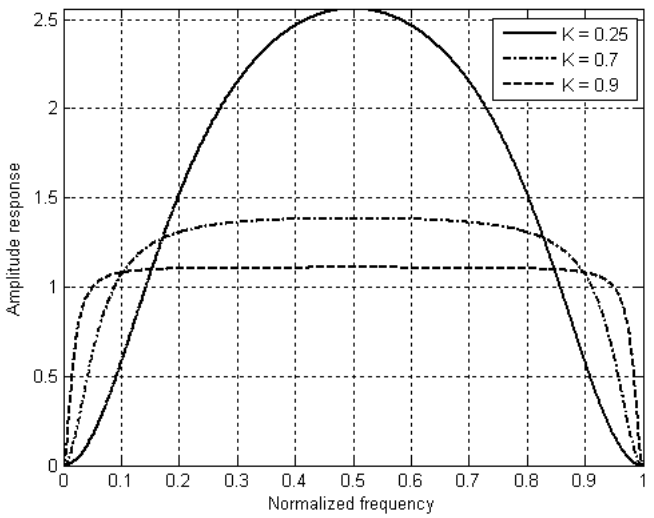


Figure 9.21. Frequency response corresponding to Eq. (9.81).

In order to avoid oscillation due to the positive feedback, the value of K should be less than unity. The value $(1-K)^{-1}$ is normally equal to the number of pulses received from the target. For example, $K = 0.9$ corresponds to ten pulses, while $K = 0.98$ corresponds to about fifty pulses.

9.7. PRF Staggering

Target velocities that correspond to multiple integers of the PRF are referred to as blind speeds. This terminology is used since an MTI filter response is equal to zero at these values. Blind speeds can pose serious limitations on the performance of MTI radars and their ability to perform adequate target detection. Using PRF agility by changing the pulse repetition interval between consecutive pulses can extend the first blind speed to more tolerable values. In order to show how PRF staggering can alleviate the problem of blind speeds, let us first assume that two radars with distinct PRFs are utilized for detection. Since blind speeds are proportional to the PRF, the blind speeds of the two radars would be different. However, using two radars to alleviate the problem of blind speeds is a very costly option. A more practical solution is to use a single radar with two or more different PRFs.

For example, consider a radar system with two interpulse periods T_1 and T_2, such that

$$\frac{T_1}{T_2} = \frac{n_1}{n_2} \tag{9.82}$$

where n_1 and n_2 are integers. The first true blind speed occurs when

$$\frac{n_1}{T_1} = \frac{n_2}{T_2} \tag{9.83}$$

This is illustrated in Fig. 9.22 for $n_1 = 4$ and $n_2 = 5$. The ratio

$$k_s = \frac{n_1}{n_2} \tag{9.84}$$

is known as the stagger ratio. Using staggering ratios closer to unity pushes the first true blind speed farther out. However, the dip in the vicinity of $1/T_1$ becomes deeper. In general, if there are N PRFs related by

$$\frac{n_1}{T_1} = \frac{n_2}{T_2} = \cdots = \frac{n_N}{T_N} \tag{9.85}$$

and if the first blind speed to occur for any of the individual PRFs is v_{blind1}, then the first true blind speed for the staggered waveform is

$$v_{blind} = \frac{n_1 + n_2 + \ldots + n_N}{N} \; v_{blind1} \tag{9.86}$$

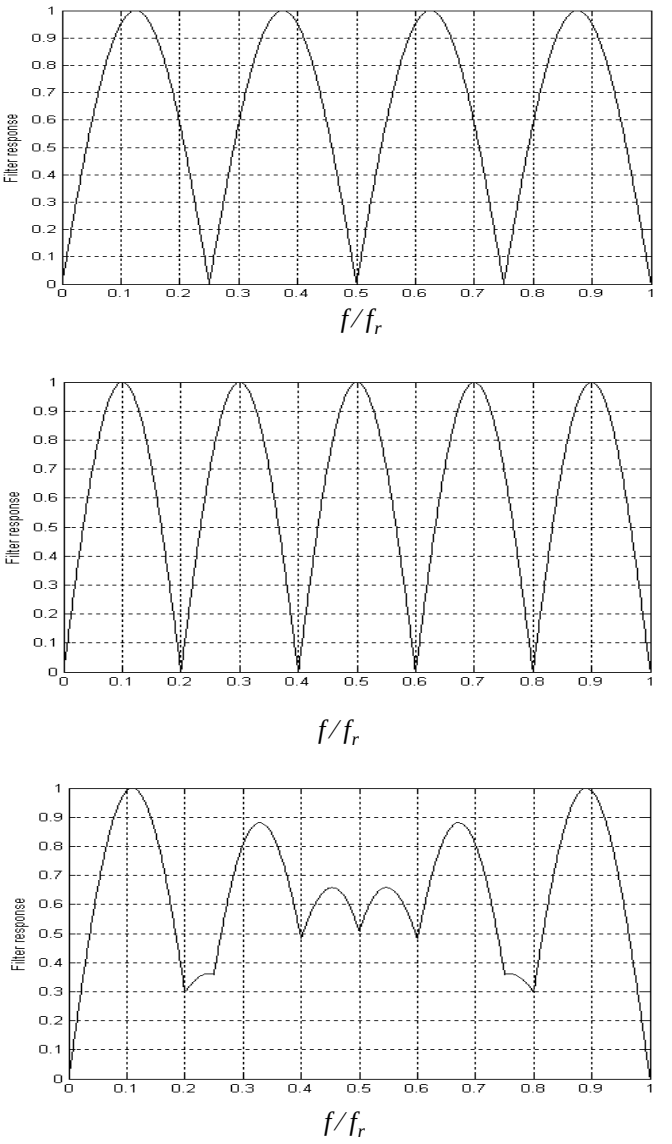


Figure 9.22. Frequency responses of a single canceler. Top plot corresponds to T_1, middle plot corresponds to T_2, bottom plot corresponds to stagger ratio $T_1/T_2 = 4/3$.

To better determine the frequency response of an MTI filter with staggered PRFs consider a three-pulse canceler with two PRFs, or equivalently two PRIs, T_1 and T_2. In this case, the impulse response will be given by

$$h(t) = [\delta(t) - \delta(t - T_1)] - [\delta(t - T_1) - \delta(t - T_1 - T_2)] \quad \textbf{(9.87)}$$

which can be written as

$$h(t) = \delta(t) - 2\delta(t - T_1) + \delta(t - T_1 - T_2) \quad \textbf{(9.88)}$$

Note that PRF staggering requires a minimum of two PRFs.

Make the change of variables $u = t - T_1$ in Eq. (9.88), and it follows

$$h(u + T_1) = \delta(u + T_1) - 2\delta(u) + \delta(u - T_2) \quad \textbf{(9.89)}$$

The Z-transform of the impulse response in Eq. (9.89) is then given by

$$H(z)z^{-T_1} = z^{T_1} - 2 + z^{-T_2} \quad \textbf{(9.90)}$$

and the amplitude frequency response for the staggered double delay line canceller is then given by

$$|H(z)|^2\Big|_{z = e^{j\omega T}} = (z^{T_1} - 2 + z^{-T_2})(z^{-T_1} - 2 + z^{T_2}) \quad \textbf{(9.91)}$$

Performing the algebraic manipulation in Eq. (9.91) and using the t trigonometric identity $(e^{j\omega T} + e^{-j\omega T}) = 2\cos\omega T$ yields

$$|H(\omega)|^2 = 6 - 4\cos(2\pi f T_1) - 4\cos(2\pi f T_2) + 2\cos(2\pi f(T_1 + T_2)) \quad \textbf{(9.92)}$$

It is customary to normalize the amplitude frequency response, thus

$$|H(\omega)|^2 = 1 - \frac{2}{3}\cos(2\pi f T_1) - \frac{2}{3}\cos(2\pi f T_2) + \frac{1}{3}\cos(2\pi f(T_1 + T_2)) \quad \textbf{(9.93)}$$

To determine the characteristics of higher stagger ratio MTI filters, adopt the notion of having several MTI filters, one for each combination of two staggered PRFs. Then the overall filter response is computed as the average of all individual filters. For example, consider the case where a PRF stagger is required with PRIs T_1, T_2, T_3, and T_4. First, compute the filter response using T_1 T_2 and denote by H_1. Then compute H_2 using T_2 and T_3, the filter H_3 is computed using T_3 T_4 and the filter H_4 is computed using T_4 and T_1. Finally compute the overall response as

$$H(f) = \frac{1}{4}[H_1(f) + H_2(f) + H_3(f) + H_4(f)] \quad \textbf{(9.94)}$$

Figure 9.23 shows the MTI filter response for a 4 stagger ratio defined. The overall response is computed as the average of 4 individual filters each corresponding to one combination of the stagger ratio. In the top portion of the figure the individual filters used were 2-pulse MTIs, while the bottom portion used 4-pulse individual MTI filters. This plot can be reproduced using the following MATLAB code.

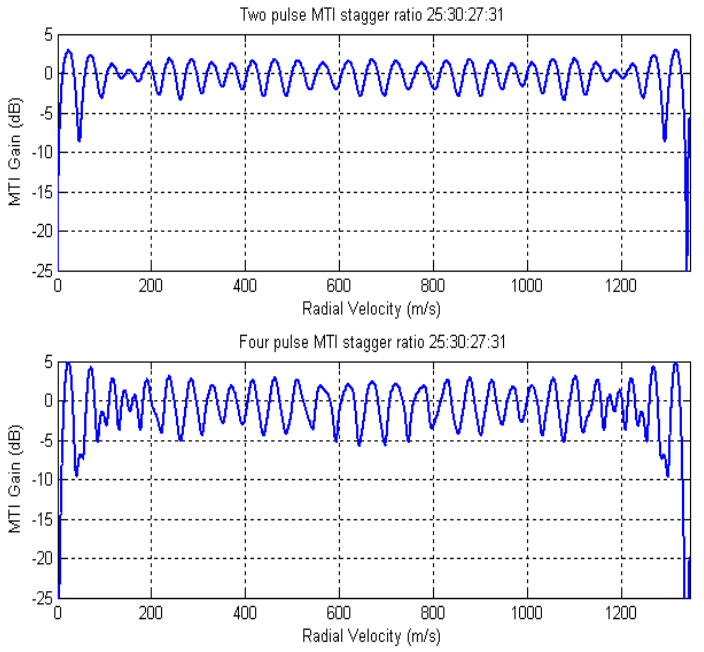


Figure 9.23. MTI responses with PRF staggering.

```
%Reproduce Fig 9.23 of text
k = .00035/25; a = 25*k; b = 30*k; c = 27*k; d = 31*k;
v2 = linspace(0,1345,10000);
f2 = (2.*v2)/.0375;
% H1(f)
T1 = exp(-j*2*pi.*f2*a); X1 = 1/2.*(1 - T1).*conj(1 - T1); H1 = 10*log10(abs(X1));
% H2(f)
T2 = exp(-j*2*pi.*f2*b); X2 = 1/2.*(1 - T2).*conj(1 - T2); H2 = 10*log10(abs(X2));
% H3(f)
T3 = exp(-j*2*pi.*f2*c); X3 = 1/2.*(1 - T3).*conj(1 - T3); H3 = 10*log10(abs(X3));
% H4(f)
T4 = exp(-j*2*pi.*f2*d); X4 = 1/2.*(1 - T4).*conj(1 - T4); H4 = 10*log10(abs(X4));
```

```
% Plot of the four components of H(f)
figure(1)
subplot(2,1,1)
% H(f) Average
ave2 = abs((X1 + X2 + X3 + X4)./4);
Have2 = 10*log10(abs((X1 + X2 + X3 + X4)./4));
plot(v2,Have2);
axis([0 1345 -25 5]);
 title('Two pulse MTI stagger ratio 25:30:27:31');
xlabel('Radial Velocity (m/s)');
 ylabel('MTI Gain (dB)'); grid on
% %Mean value of H(f)
v4 = v2; f4 = (2.*v4)/.0375;
% H1(f)
T1 = exp(-j*2*pi.*f4*a);
 T2 = exp(-j*2*pi.*f4*(a + b));
T3 = exp(-j*2*pi.*f4*(a + b + c));
X1 = 1/20.*(1 - 3.*T1 + 3.*T2 - T3).*conj(1 - 3.*T1 + 3.*T2 - T3);
H1 = 10*log10(abs(X1));
% H2(f)
T3 = exp(-j*2*pi.*f4*b);
T4 = exp(-j*2*pi.*f4*(b + c));
T5 = exp(-j*2*pi.*f4*(b + c + d));
X2 = 1/20.*(1 - 3.*T3 + 3.*T4 - T5).*conj(1 - 3.*T3 + 3.*T4 - T5);
H2 = 10*log10(abs(X2));
% H3(f)
T6 = exp(-j*2*pi.*f4*c);
T7 = exp(-j*2*pi.*f4*(c + d));
T8 = exp(-j*2*pi.*f4*(c + d + a));
X3 = 1/20.*(1 - 3.*T6 + 3.*T7 - T8).*conj(1 - 3.*T6 + 3.*T7 - T8);
H3 = 10*log10(abs(X3));
% H4(f)
T9 = exp(-j*2*pi.*f4*d); T10 = exp(-j*2*pi.*f4*(d + a));
T11 = exp(-j*2*pi.*f4*(d + a + b));
X4 = 1/20.*(1 - 3.*T9 + 3.*T10 - T11).*conj(1 - 3.*T9 + 3.*T10 - T11);
H4 = 10*log10(abs(X4));
% H(f) Average
ave4 = abs((X1 + X2 + X3 + X4)./4);
Have4 = 10*log10(abs((X1 + X2 + X3 + X4)./4));
% Plot of H(f) Average
subplot(2,1,2)
plot(v4,Have4);
axis([0 1345 -25 5]);
title('Four pulse MTI stagger ratio 25:30:27:31');
xlabel('Radial Velocity (m/s)');
ylabel('MTI Gain (dB)');
grid on
```

9.8. MTI Improvement Factor

In this section two quantities that are normally used to define the performance of MTI systems are introduced. They are Clutter Attenuation (CA) and the Improvement Factor. The MTI CA is defined as the ratio between the MTI filter input clutter power C_i to the output clutter power C_o,

$$CA = C_i / C_o \tag{9.95}$$

The MTI improvement factor is defined as the ratio of the SCR at the output to the SCR at the input,

$$I = \left(\frac{S_o}{C_o}\right) / \left(\frac{S_i}{C_i}\right) \tag{9.96}$$

which can be rewritten as

$$I = \frac{S_o}{S_i} CA \tag{9.97}$$

The ratio S_o / S_i is the average power gain of the MTI filter, and it is equal to $|H(\omega)|^2$. In this section, a closed form expression for the improvement factor using a Gaussian-shaped power spectrum (see Eq. (9.59)) is developed. A Gaussian-shaped clutter power spectrum is given by

$$S(f) = \frac{P_c}{\sqrt{2\pi}\ \sigma_f} \exp(-f^2 / 2\sigma_f^2) \tag{9.98}$$

where P_c is the clutter power (constant), and σ_f is the clutter rms frequency (which describes the clutter spectrum spread in the frequency domain, see Eq. (9.55)).

The clutter power at the input of an MTI filter is

$$C_i = \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\ \sigma_f} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) df \tag{9.99}$$

Factoring out the constant P_c yields

$$C_i = P_c \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_f} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) df \tag{9.100}$$

It follows that

$$C_i = P_c \tag{9.101}$$

The clutter power at the output of an MTI is

$$C_o = \int_{-\infty}^{\infty} S(f)|H(f)|^2 \, df \tag{9.102}$$

9.8.1. Two-Pulse MTI Case

In this section we will continue the analysis using a single delay line canceler. The frequency response for a single delay line canceler is

$$|H(f)|^2 = 4\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^2 \tag{9.103}$$

It follows that

$$C_o = \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\ \sigma_f} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) 4\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^2 \, df \tag{9.104}$$

Now, since clutter power will only be significant for small f, the ratio f/f_r is very small (i.e., $\sigma_f \ll f_r$). Consequently, by using the small angle approximation, Eq. (9.104) is approximated by

$$C_o \approx \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\ \sigma_f} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) 4\left(\frac{\pi f}{f_r}\right)^2 \, df \tag{9.105}$$

which can be rewritten as

$$C_o = \frac{4P_c\pi^2}{f_r^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_f^2}} \exp\left(-\frac{f^2}{2\sigma_f^2}\right) f^2 \, df \tag{9.106}$$

The integral part in Eq. (9.106) is the second moment of a zero-mean Gaussian distribution with variance σ_f^2. Replacing the integral in Eq. (9.106) by σ_f^2 yields

$$C_o = \frac{4P_c\pi^2}{f_r^2}\ \sigma_f^2 \tag{9.107}$$

Substituting Eq. (9.107) and Eq. (9.101) into Eq. (9.95) produces

$$CA = \frac{C_i}{C_o} = \left(\frac{f_r}{2\pi\sigma_f}\right)^2 \tag{9.108}$$

It follows that the improvement factor for a single canceler is

$$I = \left(\frac{f_r}{2\pi\sigma_f}\right)^2 \frac{S_o}{S_i} \tag{9.109}$$

The power gain ratio for a single canceler is (remember that $|H(f)|$ is periodic with period f_r)

$$\frac{S_o}{S_i} = |H(f)|^2 = \frac{1}{f_r}\int_{-f_r/2}^{f_r/2} 4\left(\sin\frac{\pi f}{f_r}\right)^2 df \tag{9.110}$$

Using the trigonometric identity $(2 - 2\cos 2\vartheta) = 4(\sin\vartheta)^2$ yields

$$|H(f)|^2 = \frac{1}{f_r}\int_{-f_r/2}^{f_r/2} \left(2 - 2\cos\frac{2\pi f}{f_r}\right) df = 2 \tag{9.111}$$

It follows that

$$I = 2(f_r/2\pi\sigma_f)^2 \tag{9.112}$$

The expression given in Eq. (9.112) is an approximation valid only for $\sigma_f \ll f_r$. When the condition $\sigma_f \ll f_r$ is not true, then the autocorrelation function needs to be used in order to develop an exact expression for the improvement factor.

Example:

A certain radar has $f_r = 800Hz$. If the clutter rms is $\sigma_f = 6.4Hz$, find the improvement factor when a single delay line canceler is used.

Solution:

The clutter attenuation CA is

$$CA = \left(\frac{f_r}{2\pi\sigma_f}\right)^2 = \left(\frac{800}{(2\pi)(6.4)}\right)^2 = 395.771 = 25.974dB$$

and since $S_o/S_i = 2 = 3dB$ we get

$$I_{dB} = (CA + S_o/S_i)_{dB} = 3 + 25.97 = 28.974dB.$$

9.8.2. The General Case

A general expression for the improvement factor for the n-pulse MTI (shown for a 2-pulse MTI in Eq. (9.112)) is given by

$$I = \frac{1}{Q^2(2(n-1)-1)!!}\left(\frac{f_r}{2\pi\sigma_f}\right)^{2(n-1)} \tag{9.113}$$

where the double factorial notation is defined by

$$(2n-1)!! = 1 \times 3 \times 5 \times \ldots \times (2n-1) \tag{9.114}$$

$$(2n)!! = 2 \times 4 \times \ldots \times 2n \tag{9.115}$$

Of course $0!! = 1$; Q is defined by

$$Q^2 = \frac{1}{\sum_n A_i^2} \tag{9.116}$$

where A_i are the binomial coefficients for the MTI filter. It follows that Q^2 for a 2-pulse, 3-pulse, and 4-pulse MTI are, respectively,

$$\left\{\frac{1}{2}, \frac{1}{20}, \frac{1}{70}\right\} \tag{9.117}$$

Using this notation, then the improvement factor for a 3-pulse and 4-pulse MTI are, respectively, given by

$$I_{3-pulse} = 2\left(\frac{f_r}{2\pi\sigma_t}\right)^4 \tag{9.118}$$

$$I_{4-pulse} = \frac{4}{3}\left(\frac{f_r}{2\pi\sigma_t}\right)^6 \tag{9.119}$$

9.9. Subclutter Visibility (SCV)

Subclutter Visibility (SCV) describes the radar's ability to detect nonstationary targets embedded in a strong clutter background, for some probabilities of detection and false alarm. It is often used as a measure of MTI performance. For example, a radar with $10dB$ SCV will be able to detect moving targets whose returns are ten times smaller than those of clutter. A sketch illustrating the concept of SCV is shown in Fig. 9.24.

If a radar system can resolve the areas of strong and weak clutter within its field of view, then Interclutter Visibility (ICV) describes the radar's ability to detect nonstationary targets between strong clutter points. The subclutter visibility is expressed as the ratio of the improvement factor to the minimum MTI

output SCR required for proper detection for a given probability of detection. More precisely,

$$SCV = I/(SCR)_o \tag{9.120}$$

When comparing the performance of different radar systems on the basis of SCV, one should use caution since the amount of clutter power is dependent on the radar resolution cell (or volume), which may be different from one radar to another. Thus, only if the different radars have the same beamwidths and the same pulse widths can SCV be used as a basis of performance comparison.

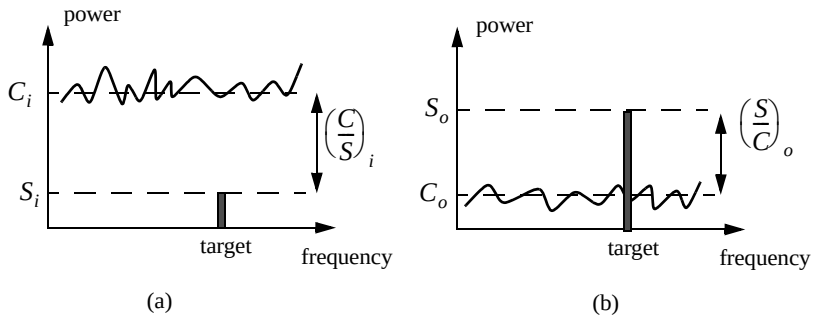


Figure 9.24. Illustration of SCV. (a) MTI input. (b) MTI output.

9.10. Delay Line Cancelers with Optimal Weights

The delay line cancelers discussed in this chapter belong to a family of transversal Finite Impulse Response (FIR) filters widely known as the "tapped delay line" filters. Figure 9.25 shows an N-stage tapped delay line implementation. When the weights are chosen such that they are the binomial coefficients (coefficients of the expansion $(1-x)^N$) with alternating signs, then the resultant MTI filter is equivalent to N-stage cascaded single line cancelers. This is illustrated in Fig. 9.26 for $N = 4$. In general, the binomial coefficients are given by

$$w_i = (-1)^{i-1} \frac{N!}{(N-i+1)!(i-1)!} \; ; \; i = 1, \ldots, N+1 \tag{9.121}$$

Using the binomial coefficients with alternating signs produces an MTI filter that closely approximates the optimal filter in the sense that it maximizes the improvement factor, as well as the probability of detection. In fact, the difference between an optimal filter and one with binomial coefficients is so small that the latter one is considered to be optimal by most radar designers. How-

ever, being optimal in the sense of the improvement factor does not guarantee a deep notch or a flat pass-band in the MTI filter response. Consequently, many researchers have been investigating other weights that can produce a deeper notch around DC, as well as a better pass-band response.

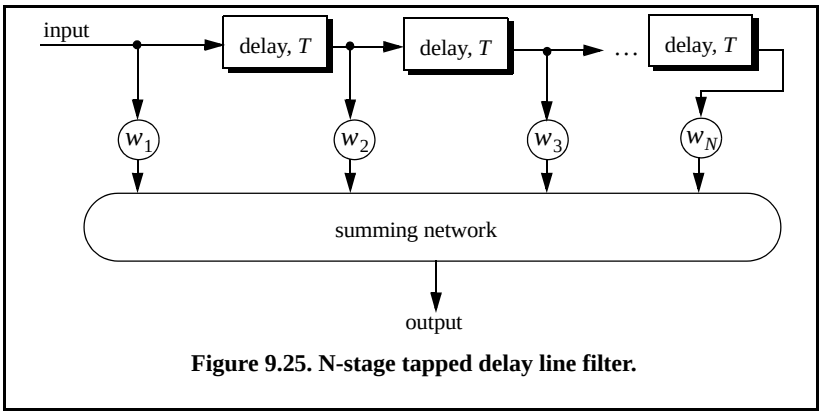


Figure 9.25. N-stage tapped delay line filter.

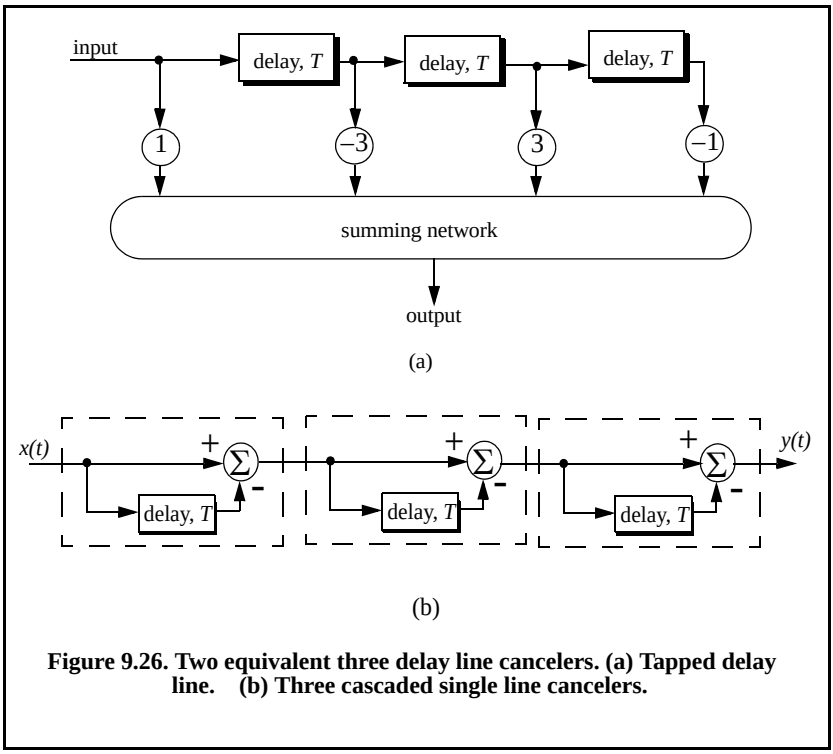


Figure 9.26. Two equivalent three delay line cancelers. (a) Tapped delay line. (b) Three cascaded single line cancelers.

In general, the average power gain for an N-stage delay line canceler is

$$\frac{S_o}{S_i} = \prod_{i=1}^{N} |H_1(f)|^2 = \prod_{i=1}^{N} 4\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^2 \tag{9.122}$$

For example, $N = 2$ (double delay line canceler) gives

$$\frac{S_o}{S_i} = 16\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^4 \tag{9.123}$$

Equation (9.123) can be rewritten as

$$\frac{S_o}{S_i} = |H_1(f)|^{2N} = 2^{2N}\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^{2N} \tag{9.124}$$

As indicated by Eq. (9.124), blind speeds for an N-stage delay canceler are identical to those of a single canceler. It follows that blind speeds are independent from the number of cancelers used. It is possible to show that Eq. (9.124) can be written as

$$\frac{S_o}{S_i} = 1 + N^2 + \left(\frac{N(N-1)}{2!}\right)^2 + \left(\frac{N(N-1)(N-2)}{3!}\right)^2 + \ldots \tag{9.125}$$

A general expression for the improvement factor of an N-stage tapped delay line canceler is reported by Nathanson[1] to be

$$I = \frac{(S_o/S_i)}{\sum_{k=1}^{N}\sum_{j=1}^{N} w_k w_j^* \rho\left(\frac{(k-j)}{f_r}\right)} \tag{9.126}$$

where the weights w_k and w_j are those of a tapped delay line canceler, and $\rho((k-j)/f_r)$ is the correlation coefficient between the kth and jth samples. For example, $N = 2$ produces

$$I = \frac{1}{1 - \frac{4}{3}\rho T + \frac{1}{3}\rho 2T} \tag{9.127}$$

1. Nathanson, F. E., *Radar Design Principles*, 2nd edition, McGraw-Hill, Inc., NY, 1991.

9.11. MATLAB Program Listings

This section presents listings for all the MATLAB programs used to produce all of the MATLAB-generated figures in this chapter. They are listed in the same order they appear in the text.

9.11.1. MATLAB Function "clutter_rcs.m"

The function *"clutter_rcs.m"* implements Eq. (9.37). It generates plots of the clutter RCS versus the radar slant range. Its outputs include the clutter RCS in dBsm. The syntax is as follows:

function [sigmaC] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, ht, b,ant_id)

where

Symbol	Description	Units	Status
sigma0	*clutter back scatterer coefficient*	*dB*	*input*
thetaE	*antenna 3dB elevation beamwidth*	*degrees*	*input*
thetaA	*antenna 3dB azimuth beamwidth*	*degrees*	*input*
SL	*antenna sidelobe level*	*dB*	*input*
range	*range; can be a vector or a single value*	*Km*	*input*
hr	*radar height*	*meters*	*input*
ht	*target height*	*meters*	*input*
b	*bandwidth*	*Hz*	*input*
ant_id	*1 for (sin(x)/x)^2 pattern* *2 for Gaussian pattern*	*none*	*input*
sigmac	*clutter RCS; can be either vector or single value depending on "range"*	*dB*	*output*

A GUI called *"clutter_rcs_gui"* was developed for this function. Executing this GUI generates plots of the σ_c versus range. Figure 9.26 shows the GUI workspace associated with this function.

MATLAB Function** "clutter_rcs.m" **Listing

```
function [sigmaC] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, ht, b,ant_id)
% This unction calculates the clutter RCS and the CNR for a ground based radar.
thetaA = thetaA * pi /180; % antenna azimuth beamwidth in radians
thetaE = thetaE * pi /180.; % antenna elevation beamwidth in radians
re = 6371000; % earth radius in meter
rh = sqrt(8.0*hr*re/3.); % range to horizon in meters
```

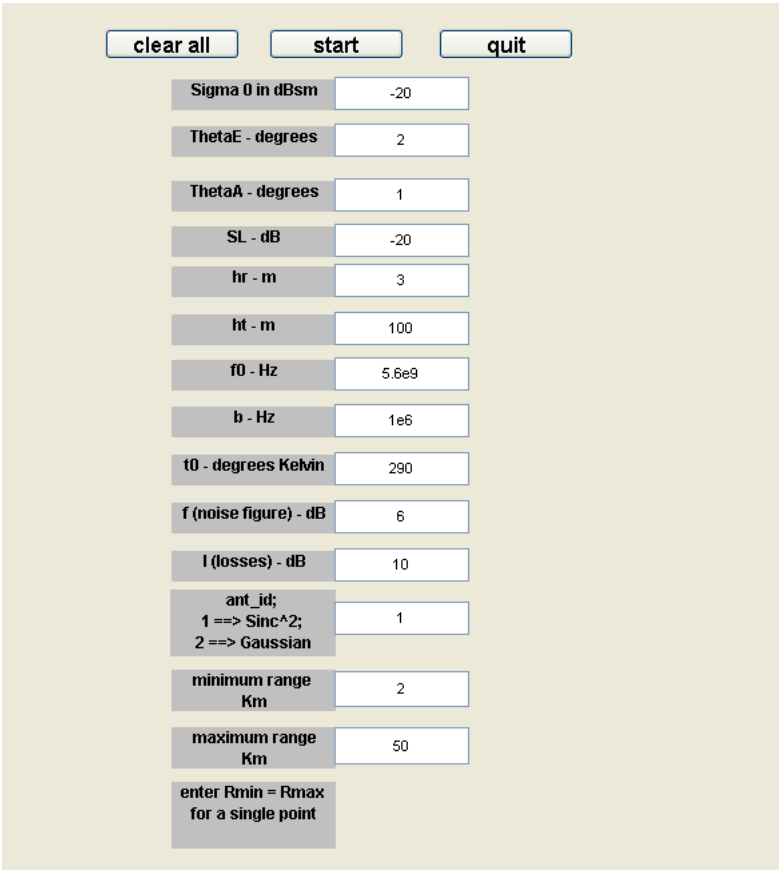


Figure 9.27. GUI workspace for *"clutter_rcs_gui.m."*

```
SLv = 10.0^(SL/10); % radar rms sidelobes in volts
sigma0v = 10.0^(sigma0/10); % clutter backscatter coefficient
deltar = 3e8 / 2 / b; % range resolution for unmodulated pulse
range_m = 1000 .* range;  % range in meters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
thetar = asin(hr ./ range_m);
thetae = asin((ht-hr) ./ range_m);
% propagation attenuation due to round earth
propag_atten = 1. + ((range_m ./ rh).^4);
Rg = range_m .* cos(thetar);
deltaRg = deltar .* cos(thetar);
theta_sum = thetae + thetar;
% use sinc^2 antenna pattern when ant_id=1
```

```
% use Gaussian antenna pattern when ant_id=2
if(ant_id ==1) % use sinc^2 antenna pattern
   ant_arg = (theta_sum ) ./ (pi*thetaE);
   gain = (sinc(ant_arg)).^2;
else
   gain = exp(-2.776 .*(theta_sum./thetaE).^2);
end
% compute sigmac
sigmac = (sigma0v .* Rg .* deltaRg) .* ...
(pi * SLv * SLv + thetaA .* gain.^2) ./ propag_atten;
sigmaC = 10*log10(sigmac);
figure(1)
plot(range, sigmaC,'linewidth',1.5)
grid
xlabel('Slant Range in Km')
ylabel('Clutter RCS in dBsm')
%
```

9.11.2. MATLAB Function "single_canceler.m"

The function *"single_canceler.m"* computes and plots (as a function of f/f_r) the amplitude response for a single delay line canceler. The syntax is as follows:

[resp] = single_canceler (fofr)

where *"fofr"* is the number of periods desired.

MATLAB Function** "single_canceler.m" **Listing

```
function [resp] = single_canceler (fofr1)
% single delay canceller
eps = 0.00001;
fofr = 0:0.01:fofr1;
arg1 = pi .* fofr;
resp = 4.0 .*((sin(arg1)).^2);
max1 = max(resp);
resp = resp ./ max1;
subplot(2,1,1)
plot(fofr,resp,'k')
xlabel ('Normalized frequency in f/fr')
ylabel( 'Amplitude response in Volts')
grid
subplot(2,1,2)
resp=10.*log10(resp+eps);
plot(fofr,resp,'k');
axis tight
grid
xlabel ('Normalized frequency in f/fr')
```

```
ylabel( 'Amplitude response in dB')
```

9.11.3. MATLAB Function "double_canceler.m"

The function *"double_canceler.m"* computes and plots (as a function of f/f_r) the amplitude response for a double delay line canceler. The syntax is as follows:

[resp] = double_canceler (fofr)

where *"fofr"* is the number of periods desired.

MATLAB Function** "double_canceler.m" **Listing

```
function [resp] = double_canceler(fofr1)
eps = 0.00001;
fofr = 0:0.01:fofr1;
arg1 = pi .* fofr;
resp = 4.0 .* ((sin(arg1)).^2);
max1 = max(resp);
resp = resp ./ max1;
resp2 = resp .* resp;
subplot(2,1,1);
plot(fofr,resp,'k--',fofr, resp2,'k');
ylabel ('Amplitude response - Volts')
resp2 = 20. .* log10(resp2+eps);
resp1 = 20. .* log10(resp+eps);
subplot(2,1,2)
plot(fofr,resp1,'k--',fofr,resp2,'k');
legend ('single canceler','double canceler')
xlabel ('Normalized frequency f/fr')
ylabel ('Amplitude response in dB')
```

Problems

9.1. Compute the signal-to-clutter ratio (SCR) for the radar described in Section 9.2.1. In this case, assume antenna 3dB beam width $\theta_{3dB} = 0.03rad$, pulse width $\tau = 10\mu s$, range $R = 50Km$, grazing angle $\psi_g = 15°$, target RCS $\sigma_t = 0.1m^2$, and clutter reflection coefficient $\sigma^0 = 0.02(m^2/m^2)$.

9.2. Repeat the example in Section 9.3 for target RCS $\sigma_t = 0.15m^2$, pulse width $\tau = 0.1\mu s$, antenna beam width $\theta_a = \theta_e = 0.03radians$; the detection range is $R = 100Km$, and $\sum \sigma_i = 1.6 \times 10^{-9}(m^2/m^3)$.

9.3. The quadrature components of the clutter power spectrum are, respectively, given by

$$\bar{S}_I(f) = \delta(f) + \frac{C}{\sqrt{2\pi}\sigma_c}\exp(-f^2/2\sigma_c^2)$$

and

$$\overline{S_Q}(f) = \frac{C}{\sqrt{2\pi}\sigma_c}\exp(-f^2/2\sigma_c^2).$$

Compute the D.C. and A.C. power of the clutter. Let $\sigma_c = 10Hz$.

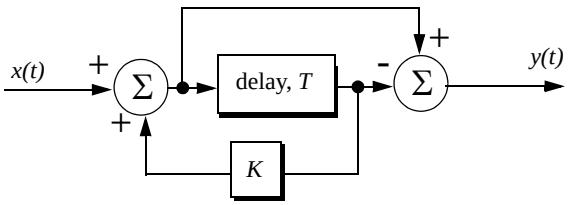
9.4. A certain radar has the following specifications: pulse width $\tau' = 1\mu s$, antenna beam width $\Omega = 1.5°$, and wavelength $\lambda = 3cm$. The radar antenna is $7.5m$ high. A certain target is simulated by two point targets (scatterers). The first scatterer is $4m$ high and has RCS $\sigma_1 = 20m^2$. The second scatterer is $12m$ high and has RCS $\sigma_2 = 1m^2$. If the target is detected at $10Km$, compute (a) SCR when both scatterers are observed by the radar, (b) SCR when only the first scatterer is observed by the radar. Assume a reflection coefficient of -1, and $\sigma^0 = -30dB$.

9.5. A certain radar has range resolution of $300m$ and is observing a target somewhere in a line of high towers each having RCS $\sigma_{tower} = 10^6 m^2$. If the target has RCS $\sigma_t = 1m^2$, (a) how much signal-to-clutter ratio should the radar have? (b) Repeat part (a) for range resolution of $30m$.

9.6. (a) Derive an expression for the impulse response of a single delay line canceler. (b) Repeat for a double delay line canceler.

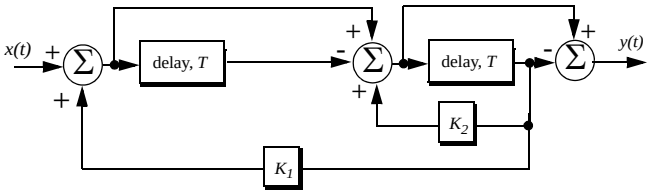
9.7. (a) What is the transfer function, $H(z)$? (b) If the clutter power spectrum is $W(f) = w_0\exp(-f^2/2\sigma_c^2)$, find an exact expression for the filter power gain. (c) Repeat part (b) for small values of frequency, f. (d) Compute the clutter attenuation and the improvement factor in terms of K and σ_c.

9.8. One implementation of a single delay line canceler with feedback is shown below



9.9. Plot the frequency response for the filter described in the previous problem for $K = -0.5, 0, and\ 0.5$.

9.10. An implementation of a double delay line canceler with feedback is shown below.



(a) What is the transfer function, $H(z)$? (b) Plot the frequency response for $K_1 = 0 = K_2$, and $K_1 = 0.2, K_2 = 0.5$.

9.11. Consider a single delay line canceler. Calculate the clutter attenuation and the improvement factor. Assume that $\sigma_c = 4Hz$ and PRF $f_r = 450Hz$.

9.12. Develop an expression for the improvement factor of a double delay line canceler.

9.13. Repeat Problem 9.10 for a double delay line canceler.

9.14. An experimental expression for the clutter power spectrum density is $W(f) = w_0 \exp(-f^2/2\sigma_c^2)$, where w_0 is a constant. Show that using this expression leads to the same result obtained for the improvement factor as developed in Section 9.8.

9.15. A certain radar uses two PRFs with stagger ratio 63/64. If the first PRF is $f_{r1} = 500Hz$, compute the blind speeds for both PRFs and for the resultant composite PRF. Assume $\lambda = 3cm$.

9.16. A certain filter used for clutter rejection has an impulse response $h(n) = \delta(n) - 3\delta(n-1) + 3\delta(n-2) - \delta(n-3)$. (a) Show an implementation of this filter using delay lines and adders. (b) What is the transfer function? (c) Plot the frequency response of this filter. (d) Calculate the output when the input is the unit step sequence.

9.17. The quadrature components of the clutter power spectrum are given in Problem 9.3. Let $\sigma_c = 10Hz$ and $f_r = 500Hz$. Compute the improvement of the signal-to-clutter ratio when a double delay line canceler is utilized.

9.18. Develop an expression for the clutter improvement factor for single and double line cancelers using the clutter autocorrelation function.

Chapter 10 Doppler Processing

In this chapter Doppler processing is analyzed in the context of continuous wave (CW) radars and pulsed Doppler radars. Continuous wave radars utilize CW waveforms, which may be considered to be a pure sinewave of the form $\cos 2\pi f_0 t$. Spectra of the radar echo from stationary targets and clutter will be concentrated at f_0. The center frequency for the echoes from moving targets will be shifted by f_d, the Doppler frequency. Thus, by measuring this frequency difference CW, radars can very accurately extract target radial velocity. Because of the continuous nature of CW emission, range measurement is not possible without some modifications to the radar operations and waveforms, which will be discussed later.

Alternatively, pulsed radars utilize a stream of pulses with a specific PRI (or PRF) to generate what is known as range-Doppler maps. Each map is divided into resolution cells. The dimensions of these resolution cells are range resolution along the time axis and Doppler resolution along the frequency axis.

10.1. CW Radar Functional Block Diagram

In order to avoid interruption of the continuous radar energy emission, two antennas are used in CW radars, one for transmission and one for reception. Figure 10.1 shows a simplified CW radar block diagram. The appropriate values of the signal frequency at different locations are noted on the diagram. The individual Narrow Band Filters (NBF) must be as narrow as possible in bandwidth in order to allow accurate Doppler measurements and minimize the amount of noise power. In theory, the operating bandwidth of a CW radar is infinitesimal (since it corresponds to an infinite duration continuous sinewave). However, systems with infinitesimal bandwidths cannot physically exist, and thus, the bandwidth of CW radars is assumed to correspond to that of a gated CW waveform.

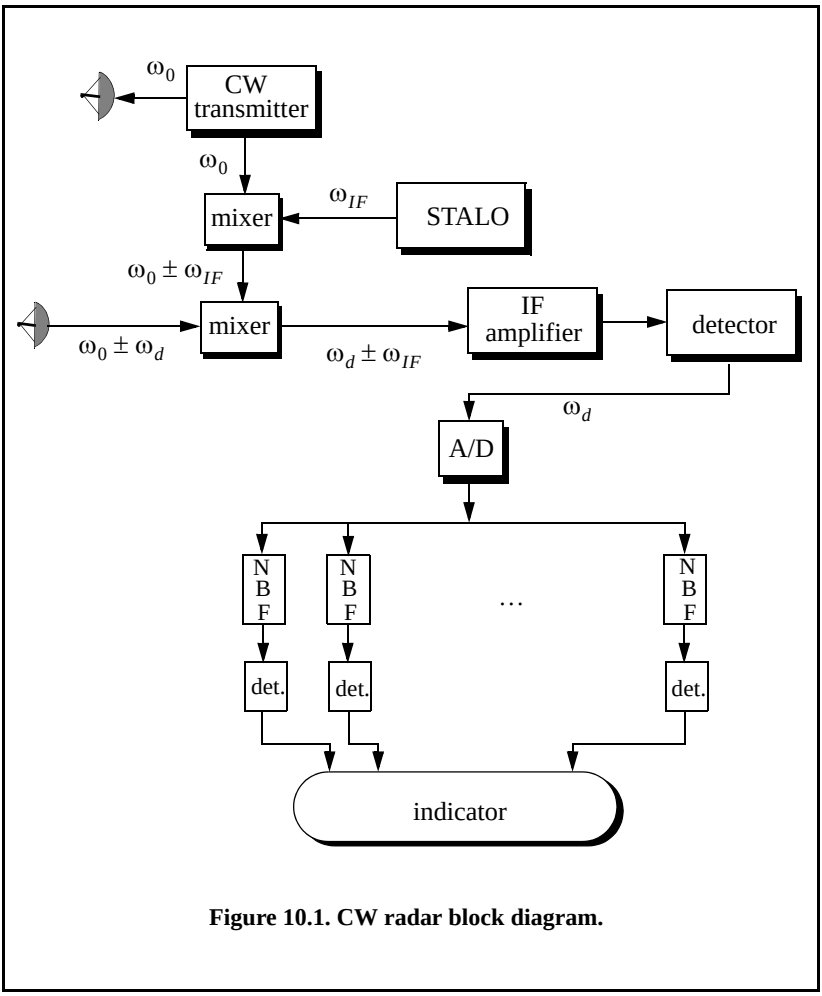


Figure 10.1. CW radar block diagram.

The NBF bank (Doppler filter bank) can be implemented using a Fast Fourier Transform (FFT). If the Doppler filter bank is implemented using an FFT of size N_{FFT}, and if the individual NBF bandwidth (FFT bin) is Δf, then the effective radar Doppler bandwidth is $N_{FFT}\Delta f/2$. The reason for the one-half factor is to account for both negative and positive Doppler shifts. The frequency resolution Δf is proportional to the inverse of the integration time.

Since range is computed from the radar echoes by measuring a two-way time delay, single frequency CW radars cannot measure target range. In order for CW radars to be able to measure target range, the transmit and receive waveforms must have some sort of timing marks. By comparing the timing marks at transmit and receive, CW radars can extract target range.

The timing mark can be implemented by modulating the transmit waveform, and one commonly used technique is Linear Frequency Modulation (LFM). Before we discuss LFM signals, we will first introduce the CW radar equation and briefly address the general Frequency Modulated (FM) waveforms using sinusoidal modulating signals.

10.1.1. CW Radar Equation

As indicated by Fig. 10.1, the CW radar receiver declares detection at the output of a particular Doppler bin if that output value passes the detection threshold within the detector box. Since the NBF bank is implemented by an FFT, only finite length data sets can be processed at a time. The length of such blocks is normally referred to as the dwell interval, integration time, or coherent processing interval. The dwell interval determines the frequency resolution or the bandwidth of the individual NBFs. More precisely,

$$\Delta f = 1/T_{Dwell} \tag{10.1}$$

T_{Dwell} is the dwell interval. Therefore, once the maximum resolvable frequency by the NBF bank is chosen the size of the NBF bank is computed as

$$N_{FFT} = 2B/\Delta f \tag{10.2}$$

B is the maximum resolvable frequency by the FFT. The factor 2 is needed to account for both positive and negative Doppler shifts. It follows that

$$T_{Dwell} = N_{FFT}/2B \tag{10.3}$$

The CW radar equation can now be derived. Consider the radar equation developed in Chapter 1. That is

$$SNR = \frac{P_{av}TG^2\lambda^2\sigma}{(4\pi)^3R^4kT_oFL} \tag{10.4}$$

where $P_{av} = (\tau/T)P_t$, τ/T, and P_t is the peak transmitted power. In CW radars the average transmitted power over the dwell interval P_{CW}, and T must be replaced by T_{Dwell}. Thus, the CW radar equation can be written as

$$SNR = \frac{P_{CW}T_{Dwell}G_tG_r\lambda^2\sigma}{(4\pi)^3R^4kT_oFLL_{win}} \tag{10.5}$$

where G_t and G_r are the transmit and receive antenna gains, respectively. The factor L_{win} is a loss term associated with the type of window (weighting) used in computing the FFT.

10.1.2. Linear Frequency Modulated CW Radar

CW radars may use LFM waveforms so that both range and Doppler information can be measured. In practical CW radars, the LFM waveform cannot be continually changed in one direction, and thus, periodicity in the modulation is normally utilized. Figure 10.2 shows a sketch of a triangular LFM waveform. The modulation does not need to be triangular; it may be sinusoidal, saw-tooth, or some other form. The dashed line in Fig. 10.2 represents the return waveform from a stationary target at range R. The beat frequency f_b is also sketched in Fig. 10.2. It is defined as the difference (due to heterodyning) between the transmitted and received signals. The time delay Δt is a measure of target range; that is,

$$\Delta t = \frac{2R}{c} \tag{10.6}$$

In practice, the modulating frequency f_m is selected such that

$$f_m = \frac{1}{2t_0} \tag{10.7}$$

The rate of frequency change, $\dot{f}$, is

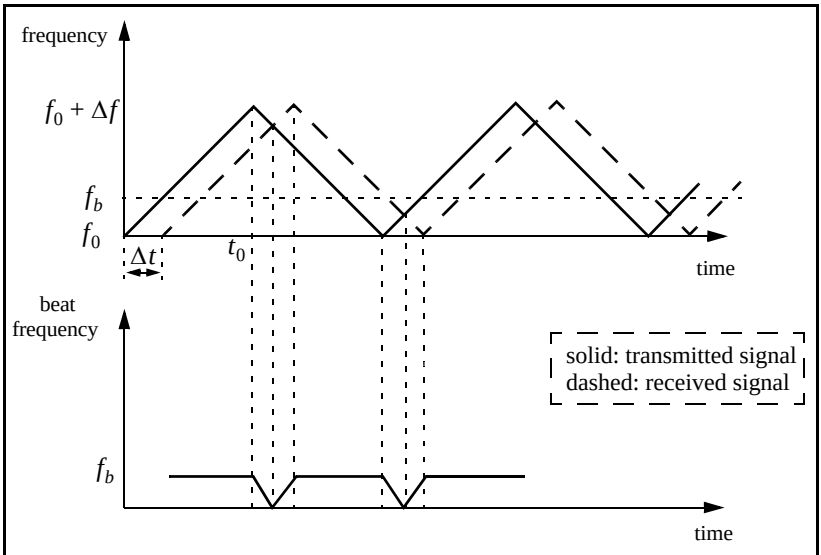


Figure 10.2. Transmitted and received triangular LFM signals and beat frequency for stationary target.

$$\dot{f} = \frac{\Delta f}{t_0} = \frac{\Delta f}{(1/2f_m)} = 2f_m \Delta f \tag{10.8}$$

where Δf is the peak frequency deviation. The beat frequency f_b is given by

$$f_b = \Delta t \dot{f} = \frac{2R}{c} \dot{f} \tag{10.9}$$

Equation (10.9) can be rearranged as

$$\dot{f} = \frac{c}{2R} f_b \tag{10.10}$$

Equating Eqs. (10.8) and (10.10) and solving for f_b yield

$$f_b = \frac{4Rf_m \Delta f}{c} \tag{10.11}$$

Now consider the case when Doppler is present (i.e., nonstationary target). The corresponding triangular LFM transmitted and received waveforms are sketched in Fig. 10.3, along with the corresponding beat frequency. As previously noted the beat frequency is defined as

$$f_b = f_{received} - f_{transmitted} \tag{10.12}$$

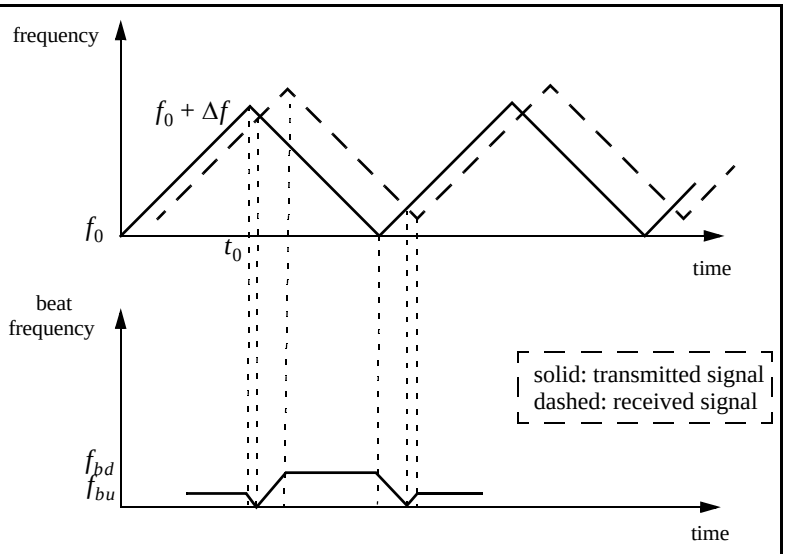


Figure 10.3. Transmitted and received LFM signals and beat frequency, for a moving target.

When the target is not stationary the received signal will contain a Doppler shift term in addition to the frequency shift due to the time delay Δt. In this case, the Doppler shift term subtracts from the beat frequency during the positive portion of the slope. Alternatively, the two terms add up during the negative portion of the slope. Denote the beat frequency during the positive (up) and negative (down) portions of the slope, respectively, as f_{bu} and f_{bd}. It follows that

$$f_{bu} = \frac{2R}{c}\dot{f} - \frac{2\dot{R}}{\lambda} \tag{10.13}$$

where $\dot{R}$ is the range rate or the target radial velocity as seen by the radar. The first term of the right-hand side of Eq. (10.13) is due to the range delay defined by Eq. (10.6), while the second term is due to the target Doppler. Similarly,

$$f_{bd} = \frac{2R}{c}\dot{f} + \frac{2\dot{R}}{\lambda} \tag{10.14}$$

Range is computed by adding Eq. (10.12) and Eq. (10.14). More precisely,

$$R = \frac{c}{4\dot{f}}(f_{bu} + f_{bd}) \tag{10.15}$$

The range rate is computed by subtracting Eq. (10.14) from Eq. (10.13),

$$\dot{R} = \frac{\lambda}{4}(f_{bd} - f_{bu}) \tag{10.16}$$

As indicated by Eq. (10.15) and Eq. (10.16), CW radars utilizing triangular LFM can extract both range and range rate information. In practice, the maximum time delay Δt_{max} is normally selected as

$$\Delta t_{max} = 0.1 t_0 \tag{10.17}$$

Thus, the maximum range is given by

$$R_{max} = \frac{0.1 c t_0}{2} = \frac{0.1c}{4f_m} \tag{10.18}$$

and the maximum unambiguous range will correspond to a shift equal to $2t_0$.

10.1.3. Multiple Frequency CW Radar

Continuous wave radars do not have to use LFM waveforms in order to obtain good range measurements. Multiple frequency schemes allow CW radars to compute very adequate range measurements without using frequency

modulation. In order to illustrate this concept, first consider a CW radar with the following waveform

$$x(t) = A\sin 2\pi f_0 t \tag{10.19}$$

The received signal from a target at range R is

$$x_r(t) = A_r \sin(2\pi f_0 t - \varphi) \tag{10.20}$$

where the phase φ is equal to

$$\varphi = 2\pi f_0 (2R/c) \tag{10.21}$$

Solving for R we obtain

$$R = \frac{c\varphi}{4\pi f_0} = \frac{\lambda}{4\pi}\varphi \tag{10.22}$$

Clearly, the maximum unambiguous range occurs when φ is maximum, i.e., $\varphi = 2\pi$. Therefore, even for relatively large radar wavelengths, R is limited to impractical small values. Next, consider a radar with two CW signals, denoted by $s_1(t)$ and $s_2(t)$. More precisely,

$$x_1(t) = A_1 \sin 2\pi f_1 t \tag{10.23}$$

$$x_2(t) = A_2 \sin 2\pi f_2 t \tag{10.24}$$

The received signals from a moving target are

$$x_{1r}(t) = A_{r1} \sin(2\pi f_1 t - \varphi_1) \tag{10.25}$$

and

$$x_{2r}(t) = A_{r2} \sin(2\pi f_2 t - \varphi_2) \tag{10.26}$$

where $\varphi_1 = (4\pi f_1 R)/c$ and $\varphi_2 = (4\pi f_2 R)/c$. After heterodyning (mixing) with the carrier frequency, the phase difference between the two received signals is

$$\varphi_2 - \varphi_1 = \Delta\varphi = \frac{4\pi R}{c}(f_2 - f_1) = \frac{4\pi R}{c}\Delta f \tag{10.27}$$

Again R is maximum when $\Delta\varphi = 2\pi$; it follows that the maximum unambiguous range is now

$$R = c/2\Delta f \tag{10.28}$$

and since $\Delta f \ll c$, the range computed by Eq. (10.28) is much greater than that computed by Eq. (10.22).

10.2. Pulsed Radars

Pulsed radars transmit and receive a train of modulated pulses. Range is extracted from the two-way time delay between a transmitted and received pulse. Doppler measurements can be made in two ways. If accurate range measurements are available between consecutive pulses, then Doppler frequency can be extracted from the range rate $\dot{R} = \Delta R / \Delta t$. This approach works fine as long as the range is not changing drastically over the interval Δt. Otherwise, pulsed radars utilize a Doppler filter bank.

Pulsed radar waveforms can be completely defined by the following: (1) carrier frequency which may vary depending on the design requirements and radar mission; (2) pulse width, which is closely related to the bandwidth and defines the range resolution; (3) modulation; and finally (4) the pulse repetition frequency. Different modulation techniques are usually utilized to enhance the radar performance, or to add more capabilities to the radar that otherwise would not have been possible. The PRF must be chosen to avoid Doppler and range ambiguities as well as maximize the average transmitted power.

Radar systems employ low, medium, and high PRF schemes. Low PRF waveforms can provide accurate, long, unambiguous range measurements, but exert severe Doppler ambiguities. Medium PRF waveforms must resolve both range and Doppler ambiguities; however, they provide adequate average transmitted power as compared to low PRFs. High PRF waveforms can provide superior average transmitted power and excellent clutter rejection capabilities. Alternatively, high PRF waveforms are extremely ambiguous in range. Radar systems utilizing high PRFs are often called Pulsed Doppler Radars (PDR). Range and Doppler ambiguities for different PRFs are summarized in Table 10.1.

Distinction of a certain PRF as low, medium, or high PRF is almost arbitrary and depends on the radar mode of operations. For example, a $3KHz$ PRF is considered low if the maximum detection range is less than $30Km$. However, the same PRF would be considered medium if the maximum detection range is well beyond $30Km$.

Radars can utilize constant and varying (agile) PRFs. For example, Moving Target Indicator (MTI) radars use PRF agility to avoid blind speeds, as discussed in Chapter 9. This kind of agility is known as PRF staggering. PRF agility is also used to avoid range and Doppler ambiguities, as will be explained in the next three sections. Additionally, PRF agility is also used to prevent jammers from locking onto the radar's PRF. These two last forms of PRF agility are sometimes referred to as PRF jitter.

Figure 10.4 shows a simplified pulsed radar block diagram. The range gates can be implemented as filters that open and close at time intervals that corre-

spond to the detection range. The width of such an interval corresponds to the desired range resolution. The radar receiver is often implemented as a series of contiguous (in time) range gates, where the width of each gate is achieved through pulse compression. The clutter rejection can be implemented using MTI or other forms of clutter rejection techniques. The NBF bank is normally implemented using an FFT, where bandwidth of the individual filters corresponds to the FFT frequency resolution.

TABLE 10.1. PRF ambiguities.

PRF	Range Ambiguous	Doppler Ambiguous
Low PRF	No	Yes
Medium PRF	Yes	Yes
High PRF	Yes	No

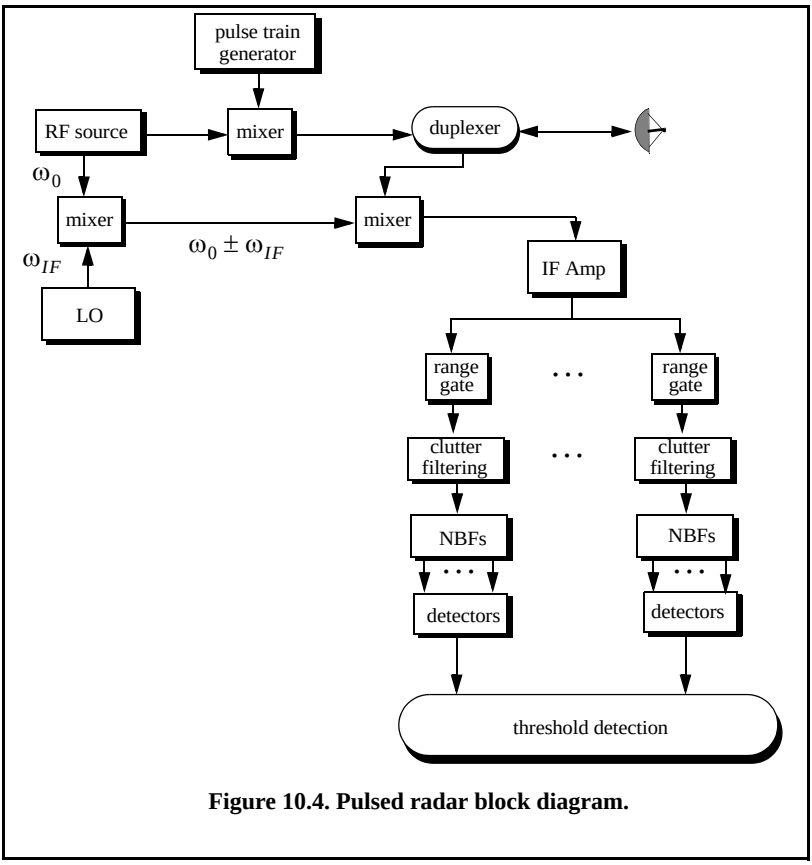


Figure 10.4. Pulsed radar block diagram.

10.2.1. Pulse Doppler Radars

In ground based radars, the amount of clutter in the radar receiver depends heavily on the radar-to-target geometry. The amount clutter is considerably higher when the radar beam has to face toward the ground. Furthermore, radars employing high PRFs have to deal with an increased amount of clutter due to folding in range. Clutter introduces additional difficulties for airborne radars when detecting ground targets and other targets flying at low altitudes. This is illustrated in Fig. 10.5. Returns from ground clutter emanate from ranges equal to the radar altitude to those which exceed the slant range along the mainbeam, with considerable clutter returns in the sidelobes and mainbeam. The presence of such large amounts of clutter interferes with radar detection capabilities and makes it extremely difficult to detect targets in the look-down mode. This difficulty in detecting ground or low altitude targets has led to the development of pulse Doppler radars where other targets, kinematics such as Doppler effects are exploited to enhance detection.

Pulse Doppler radars utilize high PRFs to increases the average transmitted power and rely on target's Doppler frequency for detection. The increase in the average transmitted power leads to an improved SNR which helps the detection process. However, using high PRFs compromise the radar's ability to detect long range target because of range ambiguities associated with high PRF applications.

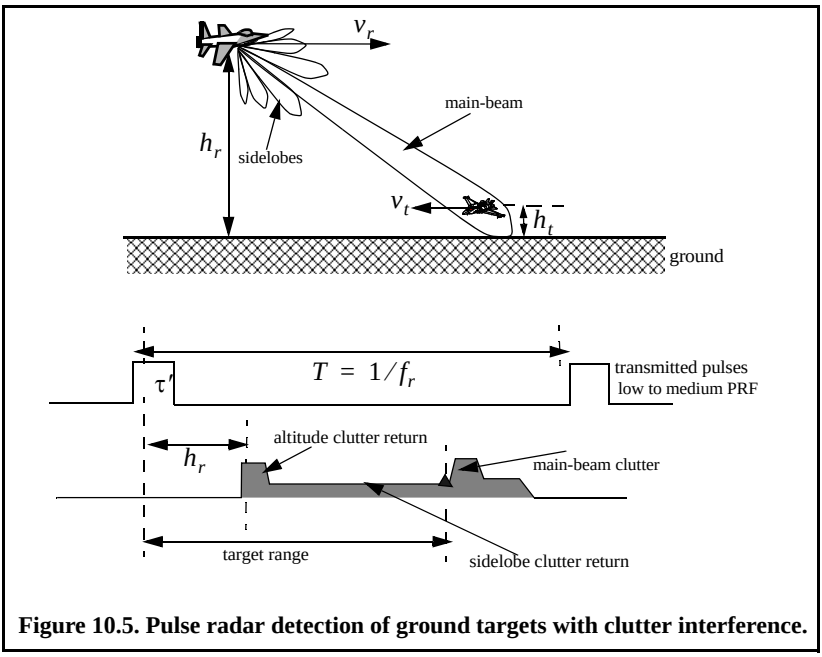


Figure 10.5. Pulse radar detection of ground targets with clutter interference.

As was explained in Chapter 9, pulse Doppler radars (or high PRF radars) have to deal with the additional increase in clutter power due to clutter folding. This has led to the development of a special class of airborne MTI filters, often referred to as AMTI. Techniques such as using specialized Doppler filters to reject clutter are very effective and are often employed by pulse Doppler radars. Pulse Doppler radars can measure target Doppler frequency (or its range rate) fairly accurately and use the fact that ground clutter typically possesses limited Doppler shift when compared with moving targets to separate the two returns. This is illustrated in Fig. 10.6. Clutter filtering (i.e., AMTI) is used to remove both main-beam and altitude clutter returns, and fast moving target detection is done effectively by exploiting its Doppler frequency. In many modern pulse Doppler radars the limiting factor in detecting slow moving targets is not clutter but rather another source of noise referred to as phase noise generated from the receiver local oscillator instabilities.

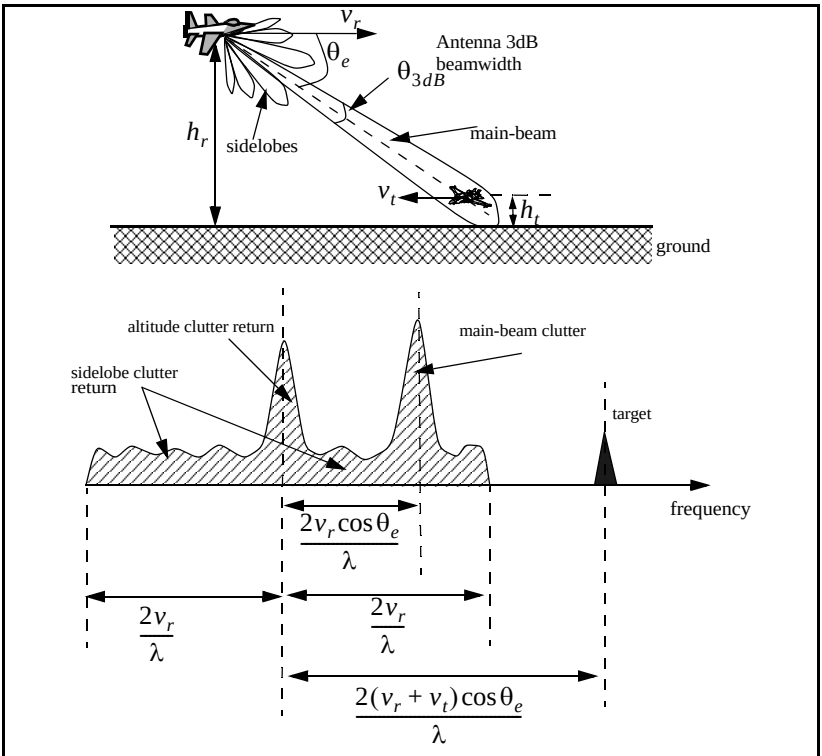


Figure 10.6. Cartoon illustrating frequency characteristics of pulse Doppler radar echoes.

10.2.2. High PRF Radar Equation

Consider a high PRF radar that uses a periodic train of very short pulses. The pulse width is τ and the period is T. This pulse train can be represented using an exponential Fourier series. The central power spectrum line (DC component) for this series contains most of the signal's power. Its value is $(\tau/T)^2$, and it is equal to the square of the transmit duty factor. Thus, the single pulse radar equation for a high PRF radar (in terms of the DC spectral power line) is

$$SNR = \frac{P_t G^2 \lambda^2 \sigma d_t^2}{(4\pi)^3 R^4 k T_o B F L d_r} \tag{10.29}$$

where, in this case, one can no longer ignore the receive duty factor since its value is comparable to the transmit duty factor. In fact, $d_r \approx d_t = \tau f_r$. Additionally, the operating radar bandwidth is now matched to the radar integration time (time on target), $B = 1/T_i$. It follows that

$$SNR = \frac{P_t \tau f_r T_i G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_o F L} \tag{10.30}$$

and finally,

$$SNR = \frac{P_{av} T_i G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_o F L} \tag{10.31}$$

where P_{av} was substituted for $P_t \tau f_r$. Note that the product $P_{av} T_i$ is a "kind of energy" product, which indicates that high PRF radars can enhance detection performance by using relatively low power and longer integration time.

Example:

Compute the single pulse SNR for a high PRF radar with the following parameters: peak power $P_t = 100KW$, *antenna gain* $G = 20dB$, *operating frequency* $f_0 = 5.6GHz$, *losses* $L = 8dB$, *noise figure* $F = 5dB$, *dwell interval* $T_i = 2s$, *duty factor* $d_t = 0.3$. *The range of interest is* $R = 50Km$. *Assume target RCS* $\sigma = 0.01m^2$.

Solution:

From Eq. (10.31) we have

$$(SNR)_{dB} = (P_{av} + G^2 + \lambda^2 + \sigma + T_i - (4\pi)^3 - R^4 - kT_o - F - L)_{dB}$$

The following table gives all parameters in dB:

P_{av}	λ^2	T_i	kT_0	$(4\pi)^3$	R^4	σ
44.771	–25.421	3.01	–23.977	32.976	187.959	–20

$$(SNR)_{dB} = 44.771 + 40 - 25.421 - 20 + 3.01 - 32.976 + 203.977 - 187.959 - 5 - 8 = 12.4dB$$

The same answer can be obtained by using the function "hprf_req.m" (see Section 10.3.2) with the following syntax:

hprf_req (100e3, 2, 20, 5.6e9, 0.01, .3, 50e3, 5, 8)

10.2.3. Pulse Doppler Radar Signal Processing

The main idea behind pulse Doppler radar signal processing is to divide the footprint (the intersection of the antenna 3dB beamwidth with the ground) into resolution cells that constitute a range Doppler map, MAP. The sides of this map are range and Doppler, as illustrated in Fig. 10.7. Fine range resolution, ΔR, is accomplished in real time by utilizing range gating and pulse compression. Frequency (Doppler) resolution is obtained from the coherent processing interval.

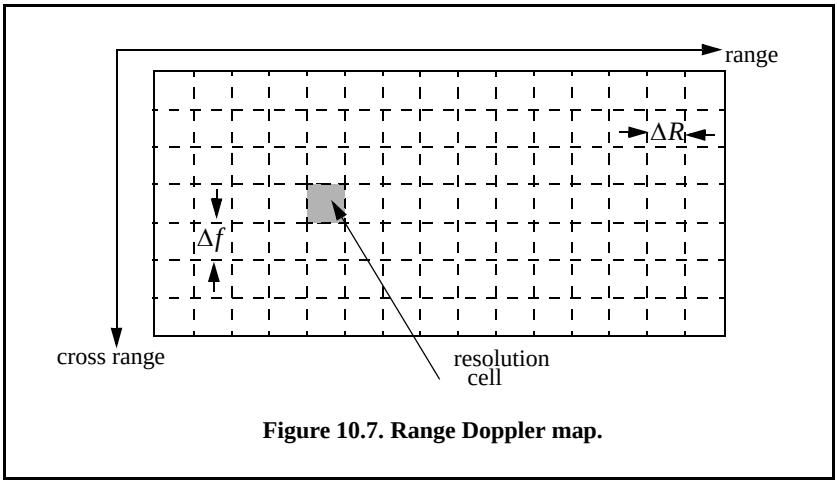


Figure 10.7. Range Doppler map.

To further illustrate this concept, consider the case where N_a is the number of azimuth (Doppler) cells, and N_r is the number of range bins. Hence, the MAP is of size $N_a \times N_r$, where the columns refer to range bins and the rows refer to azimuth cells. For each transmitted pulse within the dwell, the echoes

from consecutive range bins are recorded sequentially in the first row of MAP. Once the first row is completely filled (i.e., returns from all range bins have been received), all data (in all rows) are shifted downward one row before the next pulse is transmitted. Thus, one row of MAP is generated for every transmitted pulse. Consequently, for the current observation interval, returns from the first transmitted pulse will be located in the bottom row of MAP, and returns from the last transmitted pulse will be in the top row of MAP.

Referring to Fig. 10.4, fine range resolution is achieved using the matched filter. Clutter rejection (filtering) is performed on each range bin (i.e, rows in the MAP). Then all samples from one dwell within each range bin are processed using an FFT to resolve targets in Doppler. It follows that a peak in a given resolution cell corresponds to a specific target detection at that range and Doppler frequency. Selection of the proper size FFT and its associated parameters were discussed in Chapter 2.

10.2.4. Resolving Range Ambiguities in Pulse Doppler Radars

Pulse Doppler radars exhibit serve range ambiguities because they use high PRF pulse streams. In order to resolve these ambiguities, pulse Doppler radars utilize multiple high PRFs (PRF staggering) within each processing interval (dwell). For this purpose, consider a pulse Doppler radar that uses two PRFs, f_{r1} and f_{r2}, on transmit to resolve range ambiguity, as shown in Fig. 10.8. Denote R_{u1} and R_{u2} as the unambiguous ranges for the two PRFs, respectively. Normally, these unambiguous ranges are relatively small and are short of the desired radar unambiguous range R_u (where $R_u \gg R_{u1}, R_{u2}$). Denote the radar desired PRF that corresponds to R_u as f_{rd}.

The choice of f_{r1} and f_{r2} is such that they are relatively prime with respect to one another. One choice is to select $f_{r1} = Nf_{rd}$ and $f_{r2} = (N+1)f_{rd}$ for some integer N. Within one period of the desired PRI ($T_d = 1/f_{rd}$) the two PRFs f_{r1} and f_{r2} coincide only at one location, which is the true unambiguous target position. The time delay T_d establishes the desired unambiguous range. The time delays t_1 and t_2 correspond to the time between the transmit of a pulse on each PRF and receipt of a target return due to the same pulse.

Let M_1 be the number of PRF1 intervals between transmit of a pulse and receipt of the true target return. The quantity M_2 is similar to M_1 except it is for PRF2. It follows that over the interval 0 to T_d, the only possible results are $M_1 = M_2 = M$ or $M_1 + 1 = M_2$. The radar needs only to measure t_1 and t_2. First, consider the case when $t_1 < t_2$. In this case,

$$t_1 + \frac{M}{f_{r1}} = t_2 + \frac{M}{f_{r2}} \tag{10.32}$$

for which we get

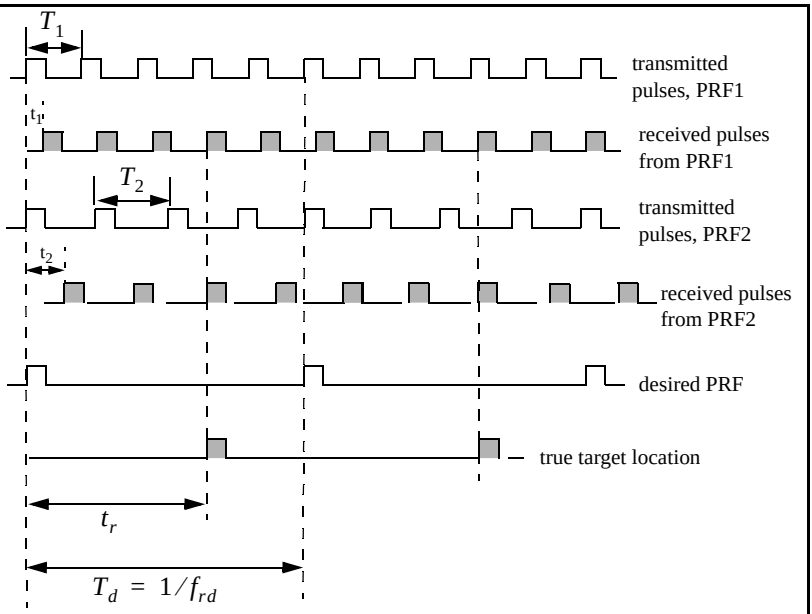


Figure 10.8. Resolving range ambiguity.

$$M = \frac{t_2 - t_1}{T_1 - T_2} \tag{10.33}$$

where $T_1 = 1/f_{r1}$ and $T_2 = 1/f_{r2}$. It follows that the round-trip time to the true target location is

$$\begin{aligned} t_r &= MT_1 + t_1 \\ t_r &= MT_2 + t_2 \end{aligned} \tag{10.34}$$

and the true target range is

$$R = ct_r/2 \tag{10.35}$$

Now, if $t_1 > t_2$, then

$$t_1 + \frac{M}{f_{r1}} = t_2 + \frac{M+1}{f_{r2}} \tag{10.36}$$

Solving for M we get

$$M = \frac{(t_2 - t_1) + T_2}{T_1 - T_2} \tag{10.37}$$

and the round-trip time to the true target location is

$$t_{r1} = MT_1 + t_1 \tag{10.38}$$

and in this case, the true target range is

$$R = \frac{ct_{r1}}{2} \tag{10.39}$$

Finally, if $t_1 = t_2$, then the target is in the first ambiguity. It follows that

$$t_{r2} = t_1 = t_2 \tag{10.40}$$

and

$$R = ct_{r2}/2 \tag{10.41}$$

Since a pulse cannot be received while the following pulse is being transmitted, these times correspond to blind ranges. This problem can be resolved by using a third PRF. In this case, once an integer N is selected, then in order to guarantee that the three PRFs are relatively prime with respect to one another, we may choose $f_{r1} = N(N+1)f_{rd}$, $f_{r2} = N(N+2)f_{rd}$, and $f_{r3} = (N+1)(N+2)f_{rd}$.

10.2.5. Resolving Doppler Ambiguity

In the case where the pulse Doppler radar is utilizing medium PRFs, it will be ambiguous in both range and Doppler. Resolving range ambiguities was discussed in the previous section. In this section Doppler ambiguity is addressed. Remember that the line spectrum of a train of pulses has $\sin x/x$ envelope (see Chapter 2), and the line spectra are separated by the PRF, f_r, as illustrated in Fig. 10.9. The Doppler filter bank is capable of resolving target Doppler as long as the anticipated Doppler shift is less than one half the bandwidth of the individual filters (i.e., one half the width of an FFT bin). Thus, pulsed radars are designed such that

$$f_r = 2f_{dmax} = (2v_{rmax})/\lambda \tag{10.42}$$

where f_{dmax} is the maximum anticipated target Doppler frequency, v_{rmax} is the maximum anticipated target radial velocity, and λ is the radar wavelength.

If the Doppler frequency of the target is high enough to make an adjacent spectral line move inside the Doppler band of interest, the radar can be Doppler ambiguous. Therefore, in order to avoid Doppler ambiguities, radar systems require high PRF rates when detecting high speed targets. When a long-range radar is required to detect a high speed target, it may not be possible to be both range and Doppler unambiguous. This problem can be resolved by using multiple PRFs. Multiple PRF schemes can be incorporated sequentially within each

dwell interval (scan or integration frame) or the radar can use a single PRF in one scan and resolve ambiguity in the next. The latter technique, however, may have problems due to changing target dynamics from one scan to the next.

The Doppler ambiguity problem is analogous to that of range ambiguity. Therefore, the same methodology can be used to resolve Doppler ambiguity. In this case, we measure the Doppler frequencies f_{d1} and f_{d2} instead of t_1 and t_2.

If $f_{d1} > f_{d2}$, then we have

$$M = \frac{(f_{d2} - f_{d1}) + f_{r2}}{f_{r1} - f_{r2}} \tag{10.43}$$

And if $f_{d1} < f_{d2}$,

$$M = \frac{f_{d2} - f_{d1}}{f_{r1} - f_{r2}} \tag{10.44}$$

and the true Doppler is

$$f_d = Mf_{r1} + f_{d1} \qquad ; \; f_d = Mf_{r2} + f_{d2} \tag{10.45}$$

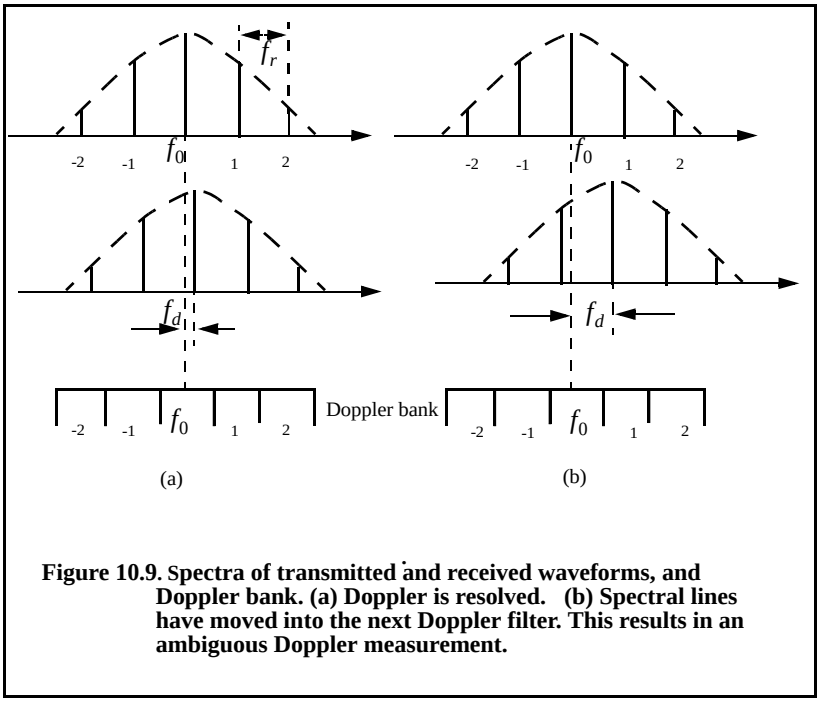


Figure 10.9. Spectra of transmitted and received waveforms, and Doppler bank. (a) Doppler is resolved. (b) Spectral lines have moved into the next Doppler filter. This results in an ambiguous Doppler measurement.

Finally, if $f_{d1} = f_{d2}$, then

$$f_d = f_{d1} = f_{d2} \tag{10.46}$$

Again, blind Dopplers can occur, which can be resolved using a third PRF.

Example:

A certain radar uses two PRFs to resolve range ambiguities. The desired unambiguous range is $R_u = 100Km$. *Choose* $N = 59$. *Compute* f_{r1}, f_{r2}, R_{u1}, *and* R_{u2}.

Solution:

First let us compute the desired PRF, f_{rd}

$$f_{rd} = \frac{c}{2R_u} = \frac{3 \times 10^8}{200 \times 10^3} = 1.5KHz$$

It follows that

$$f_{r1} = Nf_{rd} = (59)(1500) = 88.5KHz$$

$$f_{r2} = (N+1)f_{rd} = (59+1)(1500) = 90KHz$$

$$R_{u1} = \frac{c}{2f_{r1}} = \frac{3 \times 10^8}{2 \times 88.5 \times 10^3} = 1.695Km$$

$$R_{u2} = \frac{c}{2f_{r2}} = \frac{3 \times 10^8}{2 \times 90 \times 10^3} = 1.667Km.$$

Example:

Consider a radar with three PRFs; $f_{r1} = 15KHz$, $f_{r2} = 18KHz$, *and* $f_{r3} = 21KHz$. *Assume* $f_0 = 9GHz$. *Calculate the frequency position of each PRF for a target whose velocity is* $550m/s$. *Calculate* f_d *(Doppler frequency) for another target appearing at* $8KHz$, $2KHz$, *and* $17KHz$ *for each PRF.*

Solution:

The Doppler frequency is

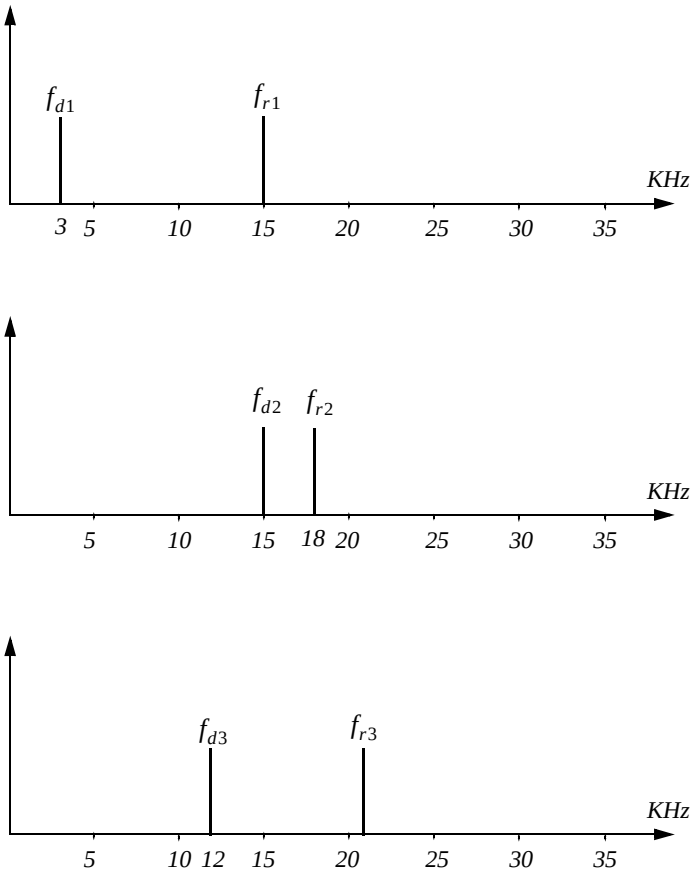
$$f_d = 2\frac{vf_0}{c} = \frac{2 \times 550 \times 9 \times 10^9}{3 \times 10^8} = 33KHz$$

Then by using Eq. (10.42) $n_i f_{ri} + f_{di} = f_d$ *where* $i = 1, 2, 3$, *we can write*

$$n_1 f_{r1} + f_{d1} = 15n_1 + f_{d1} = 33$$

$$n_2 f_{r2} + f_{d2} = 18n_2 + f_{d2} = 33$$
$$n_3 f_{r3} + f_{d3} = 21n_3 + f_{d3} = 33$$

We will show here how to compute n_1, *and leave the computations of* n_2 *and* n_3 *to the reader. First, if we choose* $n_1 = 0$, *that means* $f_{d1} = 33KHz$, *which cannot be true since* f_{d1} *cannot be greater than* f_{r1}. *Choosing* $n_1 = 1$ *is also invalid since* $f_{d1} = 18KHz$ *cannot be true either. Finally, if we choose* $n_1 = 2$ *we get* $f_{d1} = 3KHz$, *which is an acceptable value. It follows that the minimum* n_1, n_2, n_3 *that may satisfy the above three relations are* $n_1 = 2$, $n_2 = 1$, *and* $n_3 = 1$. *Thus, the apparent Doppler frequencies are* $f_{d1} = 3KHz$, $f_{d2} = 15KHz$, *and* $f_{d3} = 12KHz$, *as seen below.*



Now for the second part of the problem. Again by using Eq. (10.61) we have

$$n_1 f_{r1} + f_{d1} = f_d = 15n_1 + 8$$

$$n_2 f_{r2} + f_{d2} = f_d = 18n_2 + 2$$

$$n_3 f_{r3} + f_{d3} = f_d = 21n_3 + 17$$

We can now solve for the smallest integers n_1, n_2, n_3 *that satisfy the above three relations. See the table below.*

n	**0**	**1**	**2**	**3**	**4**
f_d from f_{r1}	8	23	<u>38</u>	53	68
f_d from f_{r2}	2	20	<u>38</u>	56	
f_d from f_{r3}	17	<u>38</u>	39		

Thus, $n_1 = 2 = n_2$, *and* $n_3 = 1$, *and the true target Doppler is* $f_d = 38KHz$. *It follows that*

$$v_r = 38000 \times \frac{0.0333}{2} = 632.7 \frac{m}{\text{sec}}$$

10.3. MATLAB Programs and Routines

10.3.1. MATLAB Program "range_calc.m"

The program "*range_calc.m*" solves the radar range equation of the form

$$R = \left(\frac{P_t \tau f_r T_i G_t G_r \lambda^2 \sigma}{(4\pi)^3 k T_0 F L (SNR)_o} \right)^{\frac{1}{4}} \qquad \textbf{(10.47)}$$

where P_t is peak transmitted power, τ is pulse width, f_r is PRF, G_t and G_r are respectively the transmitting and receiving antenna gain, λ is wavelength, σ is target cross section, k is Boltzman's constant, T_0 is 290 kelvin, F is system noise figure, L is total system losses, and $(SNR)_o$ is the minimum SNR required for detection.

One can choose either CW or pulsed radars. In the case of CW radars, the terms $P_t \tau f_r$ is replaced within the code by the average CW power P_{CW}. Additionally, the term T_i refers to the dwell interval. Alternatively, in the case of pulse radars T_i denotes the time on target. The plot inside Fig. 10.10 shows an example of the SNR versus the detection range for a pulse radar using the parameters shown in the figure. A MATLAB-based Graphical User Interface

(GUI) (see Fig. 10.10) is utilized in inputting and editing all input parameters. The outputs include the maximum detection range versus minimum SNR plots. The following MATLAB function is used by this GUI to generate the desired outputs.

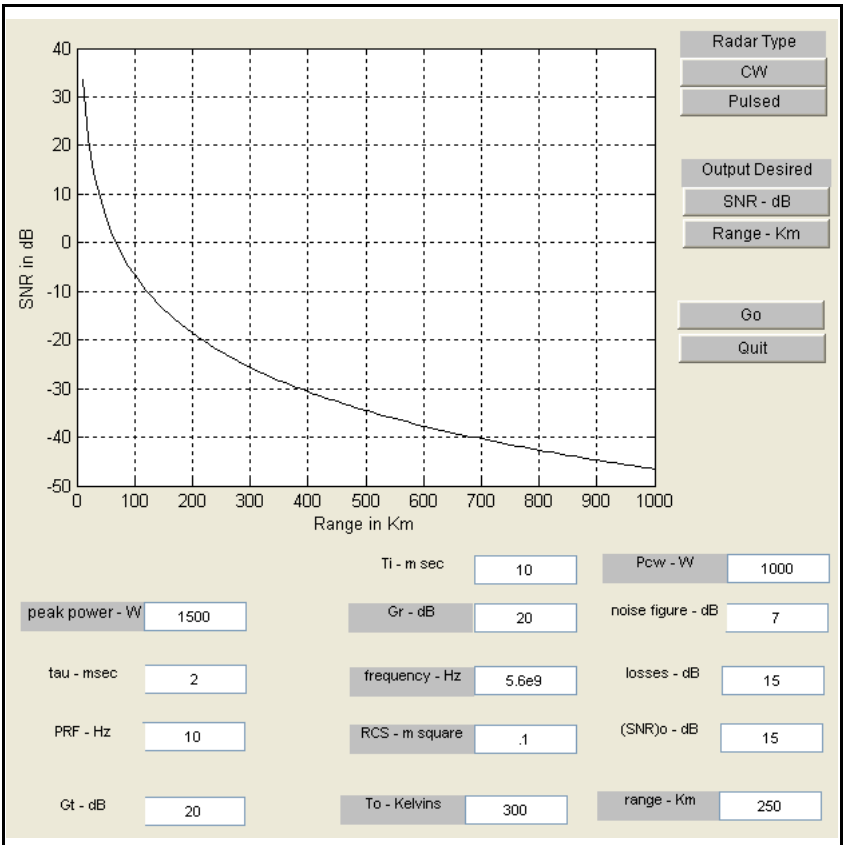


Figure 10.10. GUI work space associated with the program *"range_calc.m."*

```
function [output_par] = range_calc (pt, tau, fr, time_ti, gt, gr, freq, ...
  sigma, te, nf, loss, snro, pcw, range, radar_type, out_option)
c = 3.0e+8;
lambda = c / freq;
if (radar_type == 0)
  pav = pcw;
else
  % Compute the duty cycle
  dt = tau * 0.001 * fr;
  pav = pt * dt;
```

```
end
pav_db = 10.0 * log10(pav);
  lambda_sqdb = 10.0 * log10(lambda^2);
  sigmadb = 10.0 * log10(sigma);
  for_pi_cub = 10.0 * log10((4.0 * pi)^3);
 k_db = 10.0 * log10(1.38e-23);
 te_db = 10.0 * log10(te);
 ti_db = 10.0 * log10(time_ti);
 range_db = 10.0 * log10(range * 1000.0);
if (out_option == 0)
   %compute SNR
   snr_out = pav_db + gt + gr + lambda_sqdb + sigmadb + ti_db - ...
   for_pi_cub - k_db - te_db - nf - loss - 4.0 * range_db
  index = 0;
  for range_var = 10:10:1000
     index = index + 1;
     rangevar_db = 10.0 * log10(range_var * 1000.0);
     snr(index) = pav_db + gt + gr + lambda_sqdb + sigmadb + ti_db - ...
       for_pi_cub - k_db - te_db - nf - loss - 4.0 * rangevar_db;
  end
  var = 10:10:1000;
  plot(var,snr,'k')
  xlabel ('Range in Km');
  ylabel ('SNR in dB');
  grid
else
 range4 = pav_db + gt + gr + lambda_sqdb + sigmadb + ti_db - ...
    for_pi_cub - k_db - te_db - nf - loss - snro;
 range = 10.0^(range4/40.) / 1000.0
 index = 0;
 for snr_var = -20:1:60
    index = index + 1;
    rangedb = pav_db + gt + gr + lambda_sqdb + sigmadb + ti_db - ...
      for_pi_cub - k_db - te_db - nf - loss - snr_var;
    range(index) = 10.0^(rangedb/40.) / 1000.0;
 end
 var = -20:1:60;
 plot(var,range,'k')
 xlabel ('Minimum SNR required for detection in dB');
 ylabel ('Maximum detection range in Km');
 grid
end
return
```

10.3.2. MATLAB Function "hprf_req.m"

The function "*hprf_req.m*" implements the high PRF radar equation. Its syntax is as follows:

[snr] = hprf_req (pt, Ti, g, freq, sigma, dt, range, nf, loss)

where

Symbol	Description	Units	Status
pt	*peak power*	*W*	*input*
Ti	*time on target*	*seconds*	*input*
g	*antenna gain*	*dB*	*input*
freq	*frequency*	*Hz*	*input*
sigma	*target RCS*	m^2	*input*
dt	*duty cycle*	*none*	*input*
range	*target range (can be a single value or a vector)*	*m*	*input*
nf	*noise figure*	*dB*	*input*
loss	*radar losses*	*dB*	*input*
snr	*SNR (can be a single value or a vector)*	*dB*	*output*

MATLAB Function** "hprf_req.m" **Listing

```
function [snr] = hprf_req (pt, Ti, g, freq, sigma, dt, range, nf, loss)
% This program implements Eq. (10.31)
c = 3.0e+8; % speed of light
lambda = c / freq; % wavelength
pav = 10*log10(pt*dt); % compute average power in dB
Ti_db = 10*log10(Ti); % time on target in dB
lambda_sqdb = 10*log10(lambda^2); % compute wavelength square in dB
sigmadb = 10*log10(sigma); % convert sigma to dB
four_pi_cub = 10*log10((4.0 * pi)^3); % (4pi)^3 in dB
k_db = 10*log10(1.38e-23); % Boltzman's constant in dB
to_db = 10*log10(290); % noise temp. in dB
range_pwr4_db = 10*log10(range.^4); % vector of target range^4 in dB
% Implement Equation (1.72)
num = pav + Ti_db + g + g + lambda_sqdb + sigmadb;
den = four_pi_cub + k_db + to_db + nf + loss + range_pwr4_db;
snr = num - den;
return
```

Problems

10.1. In a multiple frequency CW radar, the transmitted waveform consists of two continuous sinewaves of frequencies $f_1 = 105KHz$ and $f_2 = 115KHz$. Compute the maximum unambiguous detection range.

10.2. Consider a radar system using linear frequency modulation. Compute the range that corresponds to $\dot{f} = 20, 10MHz$. Assume a beat frequency $f_b = 1200Hz$.

10.3. A certain radar using linear frequency modulation has a modulation frequency $f_m = 300Hz$ and frequency sweep $\Delta f = 50MHz$. Calculate the average beat frequency differences that correspond to range increments of 10 and 15 meters.

10.4. A CW radar uses linear frequency modulation to determine both range and range rate. The radar wavelength is $\lambda = 3cm$, and the frequency sweep is $\Delta f = 200KHz$. Let $t_0 = 20ms$. (a) Calculate the mean Doppler shift; (b) compute f_{bu} and f_{bd} corresponding to a target at range $R = 350Km$, which is approaching the radar with radial velocity of $250m/s$.

10.5. Consider a medium PRF radar on board an aircraft moving at a speed of 350 m/s with PRFs $f_{r1} = 10KHz$, $f_{r2} = 15KHz$, and $f_{r3} = 20KHz$; the radar operating frequency is $9.5GHz$. Calculate the frequency position of a nose-on target with a speed of 300 m/s. Also calculate the closing rate of a target appearing at 6, 5, and $18KHz$ away from the center line of PRF 10, 15, and $20KHz$, respectively.

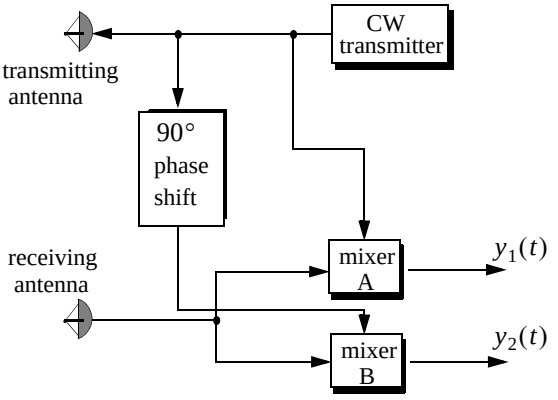
10.6. A certain radar operates at two PRFs, f_{r1} and f_{r2}, where $T_{r1} = (1/f_{r1}) = T/5$ and $T_{r2} = (1/f_{r2}) = T/6$. Show that this multiple PRF scheme will give the same range ambiguity as that of a single PRF with PRI T.

10.7. Consider an X-band radar with wavelength $\lambda = 3cm$ and bandwidth $B = 10MHz$. The radar uses two PRFs, $f_{r1} = 50KHz$ and $f_{r2} = 55.55KHz$. A target is detected at range bin 46 for f_{r1} and at bin 12 for f_{r2}. Determine the actual target range.

10.8. A certain radar uses two PRFs to resolve range ambiguities. The desired unambiguous range is $R_u = 150Km$. Select a reasonable value for N. Compute the corresponding f_{r1}, f_{r2}, R_{u1}, and R_{u2}.

10.9. A certain radar uses three PRFs to resolve range ambiguities. The desired unambiguous range is $R_u = 250Km$. Select $N = 43$. Compute the corresponding f_{r1}, f_{r2}, f_{r3}, R_{u1}, R_{u2}, and R_{u3}.

10.10. In Chapter 1 we developed an expression for the Doppler shift associated with a CW radar (i.e., $f_d = \pm 2v/\lambda$, where the plus sign is used for closing targets and the negative sign is used for receding targets). CW radars can use the system shown below to determine whether the target is closing or receding. Assuming that the emitted signal is $A\cos\omega_0 t$ and the received signal is $kA\cos((\omega_0 \pm \omega_d)t + \varphi)$, show that the direction of the target can be determined by checking the phase shift difference in the outputs $y_1(t)$ and $y_2(t)$.



Chapter 11 Adaptive Array Processing

11.1. Introduction

The emphasis in this chapter is on adaptive array processing. For this purpose, a top level overview of phased array antennas is first introduced. Phased array antennas are capable of forming multiple beams at the transmitting or receiving modes. Beamforming can be carried out at the Radio frequency (RF), Intermediate Frequency (IF), base band, or digital levels. RF beamforming is the simplest and most common technique. In this case, multiple narrow beams are formed through the use of phase shifters. IF and base band beamforming require complex coherent hardware. However, the system is operated at lower frequencies where tolerance is not as critical. Digital beamforming is more flexible than RF, IF, or base band techniques, but it requires a demanding level of processing hardware.

Adaptive arrays mostly employ phased arrays to automatically sense and eliminate unwanted signals entering the radar's Field of View (FOV) while enhancing reception about the desired target returns. For this purpose, adaptive arrays utilize a rather complicated combination of hardware and require demanding levels of software implementation. Through feedback networks, a proper set of complex weights is computed and applied to each channel of the array. A successful implementation of adaptive arrays depends heavily on two factors: first, a proper choice of the reference signal, which is used for comparison against the received target/jammer returns. A good estimate of the reference signal makes the computation of the weights systematic and effective. On the other hand, a bad estimate of the reference signal increases the array's adapting time and limits the system to impractical (non-real time) situations. Second, a fast (real time) computation of the optimum weights is essential. There have been many algorithms developed for this purpose. Nevertheless, they all share a common problem, that is, the computation of the inverse of a complex matrix. This drawback has limited the implementation of adaptive arrays to experimental systems or small arrays.

11.2. General Arrays

An array is a composite antenna formed from two or more basic radiators. Each radiator is denoted as an element. The elements forming an array could be dipoles, dish reflectors, slots in a wave guide, or any other type of radiator. Array antennas synthesize narrow directive beams that may be steered, mechanically or electronically, in many directions. Electronic steering is achieved by controlling the phase of the current feeding the array elements. Arrays with electronic beam steering capability are called phased arrays. Phased array antennas, when compared with other simple antennas such as dish reflectors, are costly and complicated to design. However, the inherent flexibility of phased array antennas to steer the beam electronically and also the need for specialized multifunction radar systems have made phased array antennas attractive for radar applications.

Figure 11.1 shows the geometrical fundamentals associated with this problem. Consider the radiation source located at (x_1, y_1, z_1) with respect to a phase reference at $(0, 0, 0)$. The electric field measured at far field point P is

$$E(\theta, \phi) = I_0 \frac{e^{-jkR_1}}{R_1} f(\theta, \phi) \tag{11.1}$$

where I_0 is the complex amplitude, $k = 2\pi/\lambda$ is the wave number, and $f(\theta, \phi)$ is the radiation pattern.

Now, consider the case where the radiation source is an array made of many elements, as shown in Fig. 11.2. The coordinates of each radiator with respect to the phase reference are (x_i, y_i, z_i), and the vector from the origin to the ith element is given by

$$\vec{r}_i = \hat{a}_x x_i + \hat{a}_y y_i + \hat{a}_z z_i \tag{11.2}$$

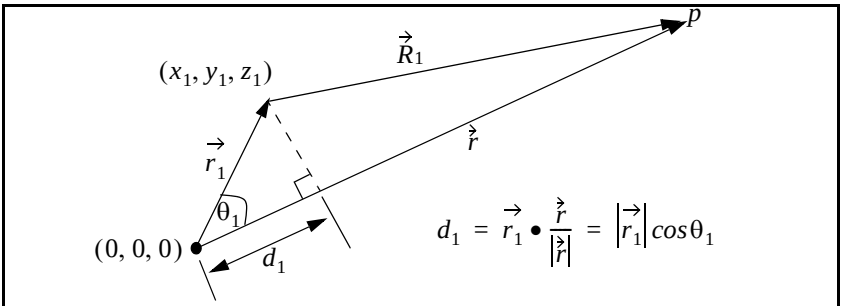


Figure 11.1. Geometry for an array antenna. Single element.

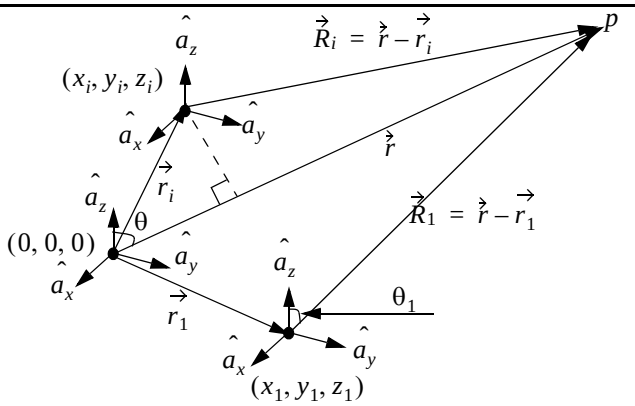


Figure 11.2. Geometry for an array antenna.

The far field components that constitute the total electric field are

$$E_i(\theta, \phi) = I_i \frac{e^{-jkR_i}}{R_i} f(\theta_i, \phi_i) \tag{11.3}$$

where

$$\begin{aligned} R_i &= \left|\vec{R}_i\right| = \left|\vec{r} - \vec{r}_i\right| = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \\ &= r\sqrt{1 + (x_i^2 + y_i^2 + z_i^2)/r^2 - 2(xx_i + yy_i + zz_i)/r^2} \end{aligned} \tag{11.4}$$

Using spherical coordinates, where $x = r\sin\theta\cos\varphi$, $y = r\sin\theta\sin\varphi$, and $z = r\cos\theta$, yields

$$\frac{(x_i^2 + y_i^2 + z_i^2)}{r^2} = \frac{\left|\vec{r}_i\right|^2}{r^2} \ll 1 \tag{11.5}$$

Thus, a good approximation (using binomial expansion) for Eq. (11.4) is

$$R_i = r - r(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi + z_i\cos\theta) \tag{11.6}$$

It follows that the phase contribution at the far field point from the ith radiator with respect to the phase reference is

$$e^{-jkR_i} = e^{-jkr}\ e^{jk(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi + z_i\cos\theta)} \tag{11.7}$$

Remember, however, that the unit vector $\hat{r}_0$ along the vector $\vec{r}$ is

$$\hat{r}_0 = \frac{\vec{r}}{|\vec{r}|} = \hat{a}_x \sin\theta\cos\phi + \hat{a}_y \sin\theta\sin\phi + \hat{a}_z \cos\theta \tag{11.8}$$

Hence, we can rewrite Eq. (11.7) as

$$e^{-jkR_i} = e^{-jkr}\ e^{jk(\vec{r}_i \bullet \hat{r}_0)} = e^{-jkr} e^{j\Psi_i(\theta,\phi)} \tag{11.9}$$

Finally, by virtue of superposition, the total electric field is

$$E(\theta,\phi) = \sum_{i=1}^{N} I_i e^{j\Psi_i(\theta,\phi)} \tag{11.10}$$

which is known as the array factor for an array antenna where the complex current for the ith element is I_i.

In general, an array can be fully characterized by its array factor. This is true since knowing the array factor provides the designer with knowledge of the array's (1) 3-dB beamwidth, (2) null-to-null beamwidth, (3) distance from the main peak to the first side-lobe, (4) height of the first side-lobe as compared to the main beam, (5) location of the nulls, (6) rate of decrease of the side-lobes, and (7) grating lobes' locations.

11.3. Linear Arrays

Figure 11.3 shows a linear array antenna consisting of N identical elements. The element spacing is d (normally measured in wavelength units). Let element #1 serve as a phase reference for the array. From the geometry, it is clear that an outgoing wave at the nth element leads the phase at the $(n+1)th$ element by $kd\sin\theta$, where $k = 2\pi/\lambda$. The combined phase at the far field observation point P is independent of ϕ and can be written as

$$\Psi(\theta,\phi) = k(\vec{r}_n \bullet \hat{r}_0) = (n-1)kd\sin\theta \tag{11.11}$$

Thus, from Eq. (11.10), the electric field at a far field observation point with direction-sine equal to $\sin\theta$ (assuming isotropic elements) is

$$E(\sin\theta) = \sum_{n=1}^{N} e^{j(n-1)(kd\sin\theta)} \tag{11.12}$$

Expanding the summation in Eq. (11.12) yields

$$E(\sin\theta) = 1 + e^{jkd\sin\theta} + \ldots + e^{j(N-1)(kd\sin\theta)} \tag{11.13}$$

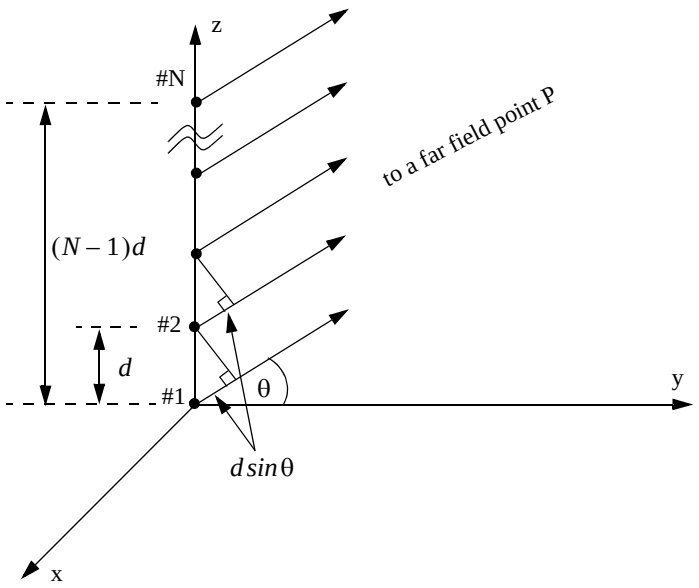


Figure 11.3. Linear array of equally spaced elements.

The right-hand side of Eq. (11.13) is a geometric series, which can be expressed in the form

$$1 + a + a^2 + a^3 + \ldots + a^{(N-1)} = \frac{1 - a^N}{1 - a} \tag{11.14}$$

Replacing a by $e^{jkd\sin\theta}$ yields

$$E(\sin\theta) = \frac{1 - e^{jNkd\sin\theta}}{1 - e^{jkd\sin\theta}} = \frac{1 - (\cos Nkd\sin\theta) - j(\sin Nkd\sin\theta)}{1 - (\cos kd\sin\theta) - j(\sin kd\sin\theta)} \tag{11.15}$$

The far field array intensity pattern is then given by

$$|E(\sin\theta)| = \sqrt{E(\sin\theta)E^*(\sin\theta)} \tag{11.16}$$

Substituting Eq. (11.15) into Eq. (11.16) and collecting terms yield

$$|E(\sin\theta)| = \sqrt{\frac{(1 - \cos Nkd\sin\theta)^2 + (\sin Nkd\sin\theta)^2}{(1 - \cos kd\sin\theta)^2 + (\sin kd\sin\theta)^2}} \tag{11.17}$$

which can be written as

$$|E(\sin\theta)| = \sqrt{\frac{1-\cos Nkd\sin\theta}{1-\cos kd\sin\theta}} \tag{11.18}$$

and using the trigonometric identity $1-\cos\theta = 2(\sin\theta/2)^2$ yields

$$|E(\sin\theta)| = \left|\frac{\sin(Nkd\sin\theta/2)}{\sin(kd\sin\theta/2)}\right| \tag{11.19}$$

which is a periodic function of $kd\sin\theta$, with a period equal to 2π.

The maximum value of $|E(\sin\theta)|$, which occurs at $\theta = 0$, is equal to N. It follows that the normalized intensity pattern is equal to

$$|E_n(\sin\theta)| = \frac{1}{N}\left|\frac{\sin((Nkd\sin\theta)/2)}{\sin((kd\sin\theta)/2)}\right| \tag{11.20}$$

The normalized two-way array pattern (radiation pattern) is given by

$$G(\sin\theta) = |E_n(\sin\theta)|^2 = \frac{1}{N^2}\left(\frac{\sin((Nkd\sin\theta)/2)}{\sin((kd\sin\theta)/2)}\right)^2 \tag{11.21}$$

Figure 11.4 shows a plot of Eq. (11.21) versus $\sin\theta$ for $N = 8$. This plot can be reproduced using the following MATLAB code.

```
% Use this code to produce figure 11.4a and 11.4b
clear all; close all;
eps = 0.00001;
k = 2*pi;
theta = -pi : pi / 10791 : pi;
var = sin(theta);
nelements = 8;
d = 1;        %  d = 1;
num = sin((nelements * k * d * 0.5) .* var);
if(abs(num) <= eps)
  num = eps;
end
den = sin((k* d * 0.5) .* var);
if(abs(den) <= eps)
  den = eps;
end
pattern = abs(num ./ den);
maxval = max(pattern);
pattern = pattern ./ maxval;
figure(1)
plot(var,pattern)
xlabel('sine angle - dimensionless')
ylabel('Array pattern')
grid
```

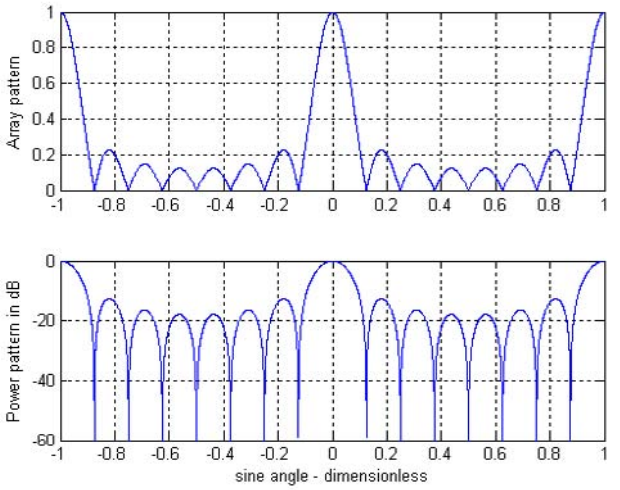


Figure 11.4a. Normalized radiation pattern for a linear array; $N = 8$; $d = \lambda$.

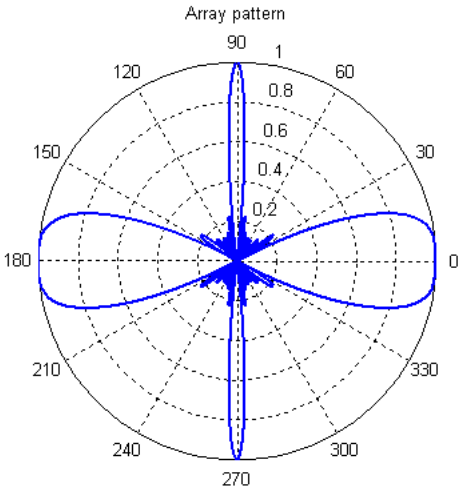


Figure 11.4b. Polar plot for the array pattern in Fig. 11.4a.

```
figure(2)
plot(var,20*log10(pattern))
axis ([-1 1 -60 0])
xlabel('sine angle - dimensionless')
ylabel('Power pattern in dB')
grid;
figure(3)
theta = theta +pi/2;
polar(theta,pattern)
title ('Array pattern')
```

The radiation pattern $G(sin\theta)$ has cylindrical symmetry about its axis $(sin\theta = 0)$ and is independent of the azimuth angle. Thus, it is completely determined by its values within the interval $(0 < \theta < \pi)$. The main beam of an array can be steered electronically by varying the phase of the current applied to each array element. Steering the main beam into the direction-sine $sin\theta_0$ is accomplished by making the phase difference between any two adjacent elements equal to $kdsin\theta_0$. In this case, the normalized radiation pattern can be written as

$$G(sin\theta) = \frac{1}{N^2}\left(\frac{sin[(Nkd/2)(sin\theta - sin\theta_0)]}{sin[(kd/2)(sin\theta - sin\theta_0)]}\right)^2 \quad \textbf{(11.22)}$$

If $\theta_0 = 0$, then the main beam is perpendicular to the array axis, and the array is said to be a broadside array. Alternatively, the array is called an endfire array when the main beam points along the array axis. The radiation pattern maxima are computed using L'Hopital's rule when both the denominator and numerator of Eq. (11.22) are zeros. More precisely,

$$\left(\frac{kdsin\theta}{2} = \pm m\pi\right);\ m = 0, 1, 2, \ldots \quad \textbf{(11.23)}$$

Solving for θ yields

$$\theta_m = asin\left(\pm\frac{\lambda m}{d}\right);\ m = 0, 1, 2, \ldots \quad \textbf{(11.24)}$$

where the subscript m is used as a maxima indicator. The first maximum occurs at $\theta_0 = 0$ and is denoted as the main beam (lobe). Other maxima occurring at $|m| \geq 1$ are called grating lobes. Grating lobes are undesirable and must be suppressed. The grating lobes occur at non-real angles when the absolute value of the arc-sine argument in Eq. (11.24) is greater than unity; it follows that $d < \lambda$. Under this condition, the main lobe is assumed to be at $\theta = 0$ (broadside array). Alternatively, when electronic beam steering is considered, the grating lobes occur at

$$\left|\sin\theta - \sin\theta_0\right| = \pm\frac{\lambda n}{d};\ \ n = 1, 2, \ldots \tag{11.25}$$

Thus, in order to prevent the grating lobes from occurring between $\pm 90°$, the element spacing should be $d < \lambda/2$.

The radiation pattern attains secondary maxima (side-lobes) when the numerator of Eq. (11.24) is maximum, or equivalently

$$\frac{Nkd\sin\theta}{2} = \pm(2l+1)\frac{\pi}{2};\ \ l = 1, 2, \ldots \tag{11.26}$$

Solving for θ yields

$$\theta_l = \operatorname{asin}\left(\pm\frac{\lambda}{2d}\frac{2l+1}{N}\right);\ \ l = 1, 2, \ldots \tag{11.27}$$

where the subscript l is used as an indication of side-lobe maxima. The nulls of the radiation pattern occur when only the numerator of Eq. (11.24) is zero. More precisely,

$$\frac{N}{2}kd\sin\theta = \pm n\pi;\quad \begin{matrix} n = 1, 2, \ldots \\ n \neq N, 2N, \ldots \end{matrix} \tag{11.28}$$

Again solving for θ yields

$$\theta_n = \operatorname{asin}\left(\pm\frac{\lambda}{d}\frac{n}{N}\right);\quad \begin{matrix} n = 1, 2, \ldots \\ n \neq N, 2N, \ldots \end{matrix} \tag{11.29}$$

where the subscript n is used as a null indicator. Define the angle that corresponds to the half power point as θ_h. It follows that the half power (3-dB) beamwidth is $2\left|\theta_m - \theta_h\right|$. This occurs when

$$\frac{N}{2}kd\sin\theta_h = 1.391\ \ radians \Rightarrow \theta_h = \operatorname{asin}\left(\frac{\lambda}{2\pi d}\frac{2.782}{N}\right) \tag{11.30}$$

In order to reduce the side-lobe levels, the array must be designed to radiate more power toward the center and much less at the edges. This can be achieved through tapering (windowing) the current distribution over the face of the array. There are many possible tapering sequences that can be used for this purpose. However, as known from spectral analysis, windowing reduces side-lobe levels at the expense of widening the main beam. Thus, for a given radar application, the choice of the tapering sequence must be based on the trade-off between side-lobe reduction and main-beam widening.

Figures 11.5 through Fig. 11.13 show plots of the array gain pattern versus steering angle for a few. These plots can be reproduced using the following MATLAB code

```
% produce figures 11.5 through 11.13
clear all; close all; clc
win = hamming(19);
[theta,patternr,patterng] = linear_array(19, 0.5, 0, -1, -1, -3);
figure(5)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 0.5\lambda; \theta = 0 degrees; Perfect phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 0, 1, win, -3);
figure(6)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle - degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 0 degrees; Perfect phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, -15, -1, -1, 3);
figure(7)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 0.5\lambda; \theta = -15 degrees; 3-bits phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 5, 1, win, 3);
figure(8)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle - degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 5 degrees; 3-bits phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 25, 1, win, 3);
figure(9)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 25 degrees; 3-bits phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, 48, -1, -1, -3);
figure(10)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = 48 degrees; Perfect phase shifters')
```

```
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, 48, 1, win, -3);
figure(11)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = 48 degrees; Perfect phase shifters; Hamming
window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, -53, -1, -1, 3);
figure(12)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = -53 degrees; 3-bits phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, -33, 1, win, 3);
figure(13)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 1.5\lambda; \theta = -33 degrees; 3-bits phase shifters; ...
Hamming window')
grid on;
axis tight
```

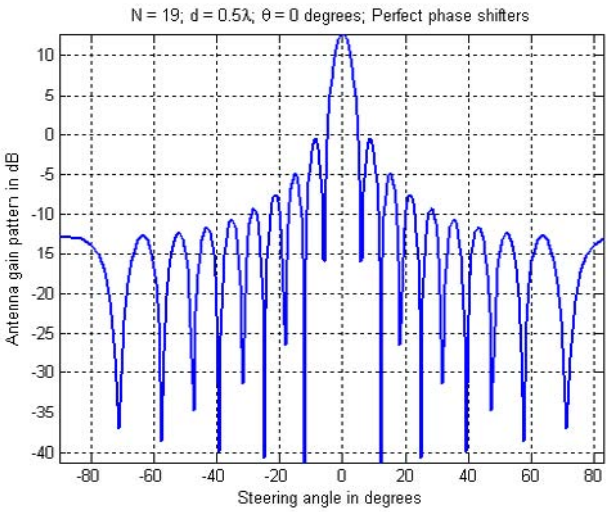


Figure 11.5. Array gain pattern.

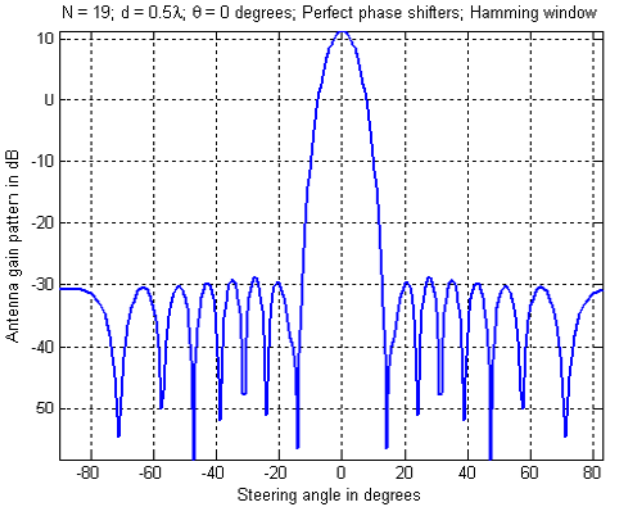


Figure 11.6. Array gain pattern.

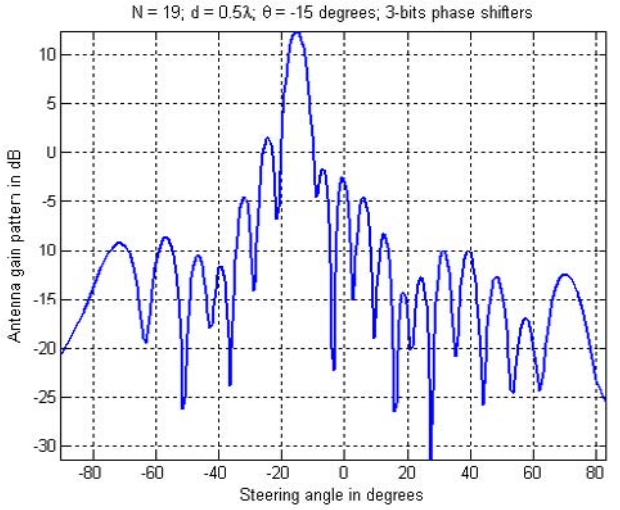


Figure 11.7. Array gain pattern.

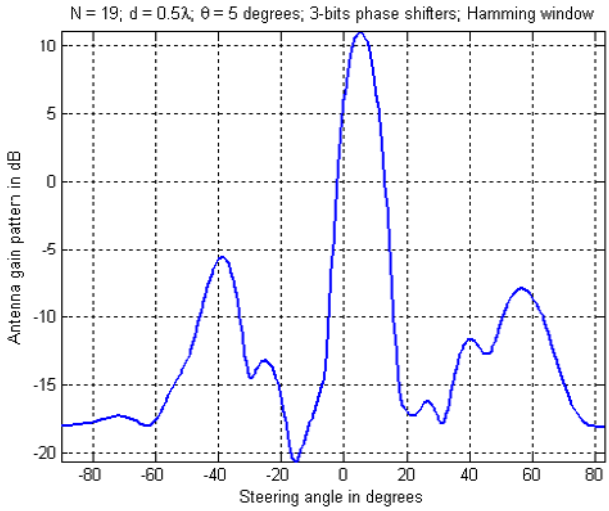


Figure 11.8. Array gain pattern.

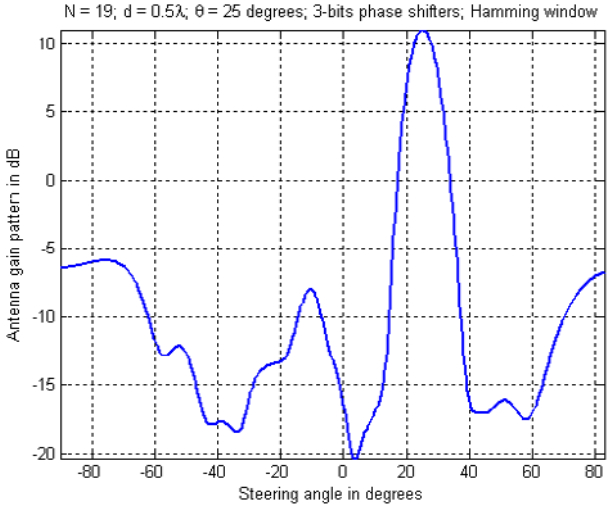


Figure 11.9. Array gain pattern.

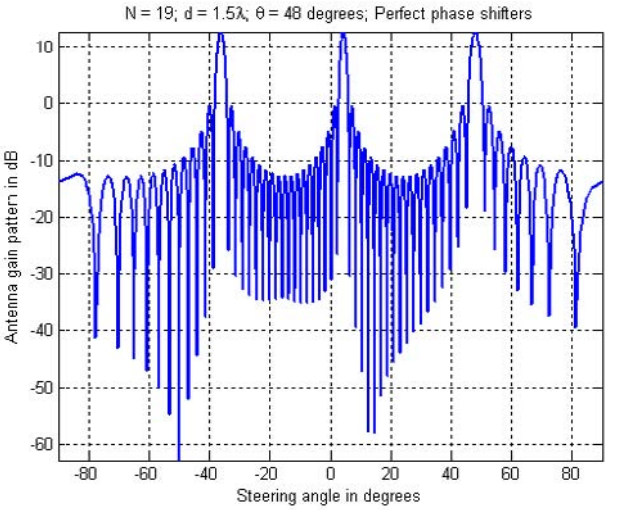


Figure 11.10. Array gain pattern.

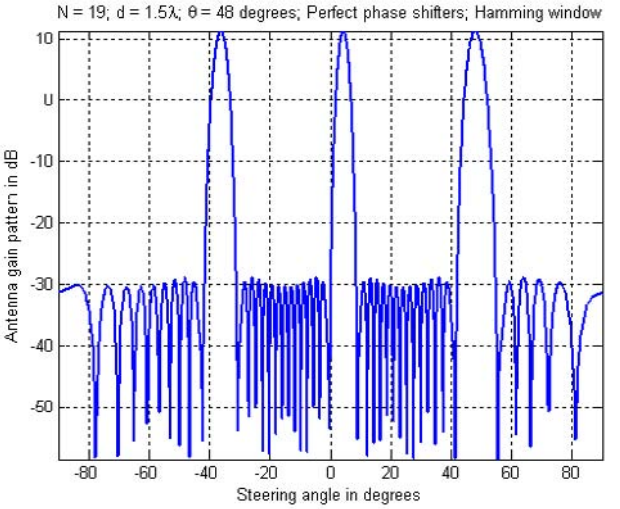


Figure 11.11. Array gain pattern.

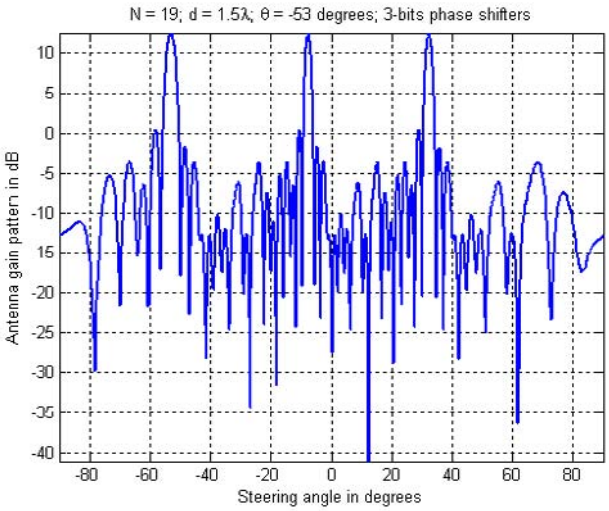


Figure 11.12. Array gain pattern.

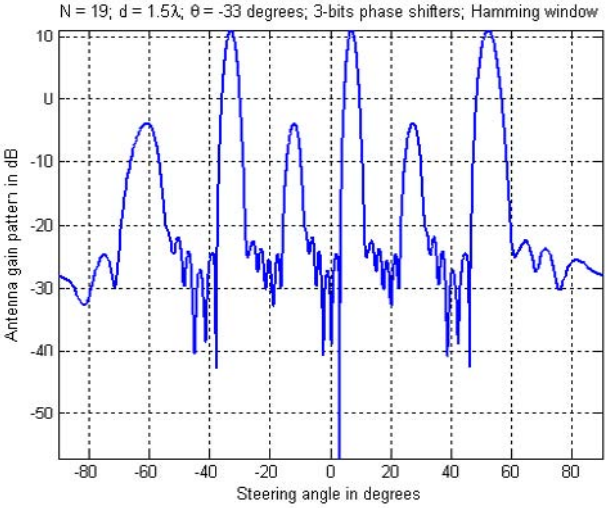


Figure 11.13. Array gain pattern.

11.4. Nonadaptive Beamforming

In adaptive beamforming the beam of interest is formed (generated) by continuously changing a set of weights through feedback circuits to minimize an output error signal. Nonadaptive or conventional beamformers do the same thing in the sense that the beam of interest is generated using a set of unique weights. Except in this case, these weights are determined a priori so that interference from a specific angle of arrival is minimized or eliminated. Different sets of weights will produce nulls in different directions in the array's field of view.

Consider a linear array of N equally spaced elements, and a plane wave $exp(j2\pi f_0 t))$ incident on the aperture with direction-sine $sin\theta$, as shown in Fig. 11.14. The weights $w_i,\ i = 0, 1, \ldots N-1$ are, in general, complex constants. The output of the beamformer is

$$y(t) = \sum_{n=0}^{N-1} w_n x_n(t-\tau_n) \tag{11.31}$$

$$\tau_n = n\frac{d}{c}sin\theta;\ n = 0, 1, \ldots, (N-1) \tag{11.32}$$

where d is the element spacing and c is the speed of light. Fourier transformation of Eq. (11.31) yields

$$Y(\omega) = \sum_{n=0}^{N-1} w_n X_n(\omega) exp(-j\omega\tau_n) = \sum_{n=0}^{N-1} w_n X_n(\omega) e^{-jn\Delta\theta} \tag{11.33}$$

The phase term $\Delta\theta$ is defined as

$$\Delta\theta = 2\pi f_0 \frac{d}{c} sin\theta = \frac{2\pi}{\lambda} d sin\theta \tag{11.34}$$

$\omega = 2\pi f_0$ and $f_0/c = 1/\lambda$. Eq. (11.33) can be written in vector form as

$$Y = \mathbf{s}^{\dagger}\mathbf{x} \tag{11.35}$$

$$\mathbf{s}^{\dagger} = \left[1 \quad e^{j\Delta\theta} \quad \ldots \quad e^{j(N-1)\Delta\theta}\right] \tag{11.36}$$

$$\mathbf{x}^{\dagger} = \left[w_0 X_o \ w_1 X_1 \ \ldots \ \ldots w_{N-1} X_{N-1}\right]^{*} \tag{11.37}$$

where the superscripts * and †, respectively, indicate complex conjugate and complex conjugate transpose.

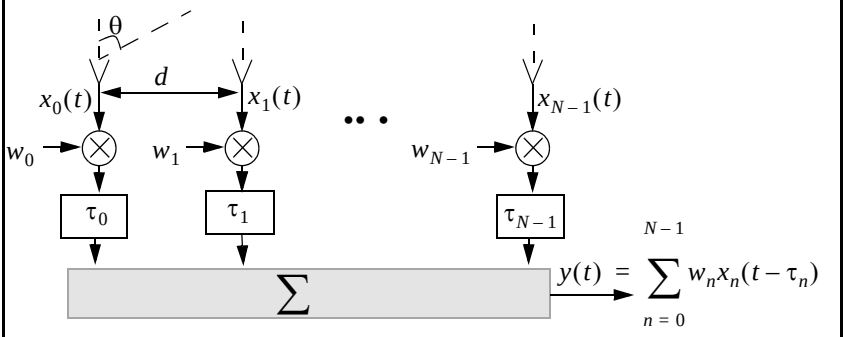


Figure 11.14. A linear array of size N, element spacing d, and an incident plane wave defined by $sin\theta$.

Let A_1 be the amplitude of the wavefront defined by $sin\theta_1$; it follows that the vector $\mathbf{x}$ is given by

$$\mathbf{x} = A_1 \mathbf{s}_1^* \tag{11.38}$$

where $\mathbf{s}_1$ is a steering vector can be written as,

$$\mathbf{s}^\dagger{}_1 = \left[w_0 \quad w_1 e^{-j\Delta\theta_1} \ldots w_{N-1} e^{-j(N-1)\Delta\theta_1} \right] \; ; \; \Delta\theta_1 = \frac{2\pi d}{\lambda} \cdot sin\theta_1 \tag{11.39}$$

Using this notation, Eq. (11.35) can be expressed in the form

$$Y = \mathbf{s}^\dagger \mathbf{x} = A_1 \mathbf{s}^\dagger \mathbf{s}^*_1 \tag{11.40}$$

The array pattern of the beam steered at θ_1 is computed as the expected value of Y. In other words, the power spectrum density for the beamformer output is given by

$$S(k) = E[Y Y^\dagger] = P_1 \mathbf{s}^\dagger \Re \mathbf{s} \tag{11.41}$$

where $P_1 = E[|A_1|^2]$ and $\Re$ is the correlation matrix given by

$$\Re = E\{\mathbf{s}_1 \mathbf{s}^\dagger{}_1\} \tag{11.42}$$

Consider L incident plane waves with directions of arrival defined by

$$\Delta\theta_i = \frac{2\pi d}{\lambda} sin\theta_i; \; i = 1, L \tag{11.43}$$

The nth sample at the output of the mth sensor is

$$y_m(n) = \upsilon(n) + \sum_{i=1}^{L} A_i(n) exp(-jm\Delta\theta_i);\ m = 0, N-1 \tag{11.44}$$

where $A_i(n)$ is the amplitude of the ith plane wave and $\upsilon(n)$ is white, zero-mean noise with variance σ_υ^2, and it is assumed to be uncorrelated with the signals. Equation (11.44) can be written in vector notation as

$$\mathbf{y}(n) = \upsilon(n) + \sum_{i=1}^{L} A_i(n)\mathbf{s}_i^* \tag{11.45}$$

A set of L steering vectors is needed to simultaneously form L beams. Define the steering matrix $\aleph$ as

$$\aleph = \left[\mathbf{s}_1\ \mathbf{s}_2\ \ldots\ \mathbf{s}_L\right] \tag{11.46}$$

Then the autocorrelation matrix of the field measured by the array is

$$\Re = E\{\mathbf{y}_m(n)\mathbf{y}_m^\dagger(n)\} = \sigma_\upsilon^2 \boldsymbol{I} + \aleph C \aleph^\dagger \tag{11.47}$$

where $C = dig\left[P_1\ P_2\ \ldots\ P_L\right]$, and $\boldsymbol{I}$ is the identity matrix.

For example, consider the case depicted in Fig. 11.15, where an interfering signal located at angle $\theta_i = \pi/6$ off the antenna boresight. The desired signal is at $\theta_t = 0°$. The desired output should contain only the signal $s(t)$. From Eq. (11.33) and Eq. (11.34) the desired output is

$$y_d(t) = \sum_{n=0}^{1} w_n x_n(t - \tau_{n_t}) = w_0 x_0 + w_1 x_1 e^{-j\frac{2\pi}{\lambda} d \sin\theta_t} \tag{11.48}$$

Since the angle $\theta_t = 0°$, it follows that

$$y_d(t) = \{Ae^{j2\pi f_0 t}\}\{(w_{0R} + jw_{0I}) + (w_{1R} + jw_{1I})\} \tag{11.49}$$

$$\begin{aligned} w_0 &= w_{0R} + jw_{0I} \\ w_1 &= w_{1R} + jw_{1I} \end{aligned} \tag{11.50}$$

Thus, in order to produce the desired signal, $s(t)$, at the output of the beamformer, it is required that

$$\begin{aligned} w_{0R} + w_{1R} = 1 &\Rightarrow w_{0R} = 1 - w_{1R} \\ w_{0I} + w_{1I} = 0 &\Rightarrow w_{0I} = -w_{1I} \end{aligned} \tag{11.51}$$

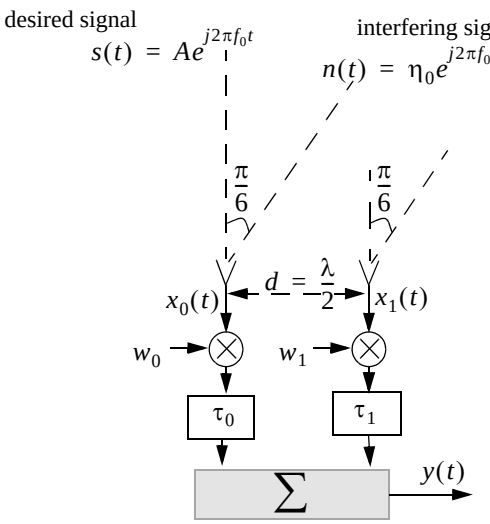


Figure 11.15. Two element array with an interfering signal at $\theta_i = \pi/6$.

Next, the output due to the interfering signal is

$$y_i(t) = \sum_{n=0}^{1} w_n x_n(t-\tau_{n_t}) = w_0 x_0 + w_1 x_1 e^{-j\frac{2\pi}{\lambda} d \sin\theta_i} \tag{11.52}$$

Since the angle $\theta_i = \pi/6$, it follows that

$$y_d(t) = \{\eta_0 e^{j2\pi f_0 t}\}\{(w_{0R} + jw_{0I}) - j(w_{1R} + jw_{1I})\} \tag{11.53}$$

and in order to eliminate the interference signal from the output of the beamformer, it is required that

$$\begin{aligned} w_{0R} + w_{1I} = 0 &\Rightarrow w_{0R} = -w_{1I} \\ w_{0I} - w_{1R} = 0 &\Rightarrow w_{0I} = w_{1R} \end{aligned} \tag{11.54}$$

Solving Eq. (11.51) and Eq. (11.54) yields

$$w_{0R} = \frac{1}{2}; w_{0I} = \frac{1}{2}; w_{1R} = \frac{1}{2}; w_{1I} = \frac{-1}{2} \tag{11.55}$$

Using the weights given in Eq. (11.55) will allow the desired signal to get through the beamformer unaffected; however, the interference signal will be completely eliminated from the output.

11.5. Adaptive Array Processing

11.5.1. Adaptive Signal Processing Using Least Mean Squares (LMS)

Adaptive signal processing evolved as a natural evolution from adaptive control techniques of time varying systems. Advances in digital processing computation techniques and associated hardware have facilitated maturing adaptive processing techniques and algorithms. Consider the basic adaptive digital system shown in Fig. 11.16. The system input is the sequence $x[k]$ and its output is the sequence $y[k]$. What differentiates adaptive from nonadaptive systems is that in adaptive systems the transfer function $H_k(z)$ is now time varying. The arrow through the transfer function box is used to indicate adaptive processing (or time varying transfer function). The sequence $d[k]$ is referred to as the *desired* response sequence. The error sequence is the difference between the desired response and the actual response. Remember that the desired sequence is not completely known; otherwise, if it were completely known, one would not need any adaptive processing to compute it. The definition of this desired response is dependent on the system specific requirements.

Many different techniques and algorithms have been developed to minimize the error sequence. Using one technique over another depends heavily on the operating environment under consideration. For example, if the input sequence is a stationary random process, then minimizing the error signal is nothing more than solving the least mean squares problem. However, in most adaptive processing systems the input signal is a nonstationary process. In this section the least mean squares technique is examined.

The least mean squares (LMS) algorithm is the most commonly utilized algorithm in adaptive processing, primary because of its simplicity. The time varying transfer function of order L can be written as a Finite Impulse Response (FIR) filter defined by

$$H_k(z) = b_0 + b_1 z^{-1} + \ldots + b_L z^{-L} \tag{11.56}$$

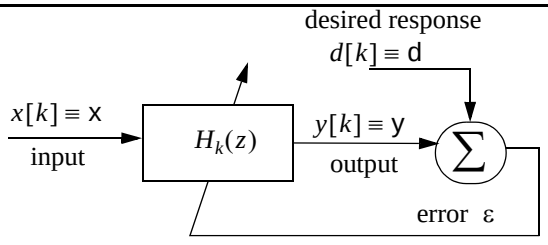


Figure 11.16. Basic adaptive system.

The input output relationship is given by the discrete convolution

$$y(k) = \sum_{n=0}^{L} b_n(k)x(k-n) \tag{11.57}$$

The goal of the adaptive LMS process is to adjust the filter coefficients toward an optimum minimum mean square error (MMSE). The most common approach to achieving this MMSE utilizes the method of steepest descent. For this purpose, define the filter coefficients in vector notation as

$$\mathbf{b}_k = \left[b_o(k)\ b_1(k)\ldots\quad b_L(k)\right]^{\dagger} \tag{11.58}$$

then

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \mu\nabla_k \tag{11.59}$$

where μ is a parameter that controls how fast the error converges to the desired MMSE value, and the gradient vector ∇_k is defined by

$$\nabla_k = \frac{\partial}{\partial \mathbf{b}_k}E[\varepsilon_k^2] = \left[\frac{\partial}{\partial \mathbf{b}_0(k)}E[\varepsilon_k^2]\ \ldots\ \frac{\partial}{\partial \mathbf{b}_L(k)}E[\varepsilon_k^2]\right]^{\dagger} \tag{11.60}$$

As clearly indicated by Eq. (11.59) the adaptive filter coefficients update rate is proportional to the negative gradient; thus, if the gradient is known at each step of the adaptive process, then better computation of the coefficient is obtained. In other words, the MMSE decreases from step k to step $k+1$. Of course, once the solution is found the gradient becomes zero and the coefficient will not change any more.

When the gradient is not known, estimates of the gradient are used based only on the instantaneous squared error. These estimates are defined by

$$\hat{\nabla}_k = \frac{\partial}{\partial \mathbf{b}_k}[\varepsilon_k^2] = 2\varepsilon_k\frac{\partial}{\partial \mathbf{b}_k}(d_k - y_k) \tag{11.61}$$

Since the desired sequence $d[k]$ is independent from the output $y[k]$, Eq. (11.61) can be written as

$$\hat{\nabla}_k = -2\varepsilon_k\mathbf{x}_k \tag{11.62}$$

where the vector $\mathbf{x}_k$ is the input signal sequence. Substituting Eq. (11.62) into Eq. (11.59) yields

$$\mathbf{b}_{k+1} = \mathbf{b}_k + 2\varepsilon_k\mu\mathbf{x}_k \tag{11.63}$$

The choice of the convergence parameter μ plays a significant role in determining the system performance. This is clear because as indicated by Eq. (11.63), a successful implementation of the LMS algorithm depends on the input signal, the choice of the desired signal, and the convergence parameter. Much research and effort has been devoted toward selecting the optimal value for μ. Nonetheless, no universal value has been found. However, a range for this parameter has been determined to be $0 < \mu < 1$.

Often, a normalized value for the convergence parameter μ_N can be used instead of its absolute value. That is,

$$\mu_N = \frac{\mu}{(L+1)\sigma^2} \tag{11.64}$$

where L is the order of the adaptive FIR filter and σ^2 is the variance (power) of the input signal. When the input signal is not stationary and its variance is varying with time, a time varying estimate of σ^2 is used. That is

$$\hat{\sigma}_k^2 = \alpha x_k^2 + (1-\alpha)\hat{\sigma}_{k-1}^2 \tag{11.65}$$

where α is a factor selected such that $0 < \alpha < 1$. Finally, Eq. (11.63) can be written as

$$b_{k+1} = b_k + \frac{2\varepsilon_k \mu x_k}{(L+1)\hat{\sigma}_k^2} \tag{11.66}$$

As an example and in reference to Fig. 11.15, let the input and desired signals be defined as

$$x[k] = \sqrt{2}\ \sin\left(\frac{2\pi k}{20}\right) + n[k] \qquad ;\ k = 0, 1, \ldots, 500 \tag{11.67}$$

$$d[k] = \sqrt{2}\ \sin\left(\frac{2\pi k}{20}\right) \qquad ;\ k = 0, 1, \ldots, 500 \tag{11.68}$$

where $n[k]$ is additive white noise with zero mean and variance $\sigma_n^2 = 2$. Figure 11.17 shows the output of the LMS algorithm defined in Eq. (11.66) when $\mu = 0.1$ and $\alpha = 0$. Figure 11.18 is similar to Fig. 11.17 except in this case, $\mu = 0.01$ and $\alpha = 0.1$. Note that in Fig. 11.18 the rate of convergences is reduced since μ is smaller than that used in Fig. 11.17; however, the filter's output is less noise because α is greater than zero which allows for more accurate updates of the noise variance as defined in Eq. (11.65). These plots can be reproduced using the following MATLAB code which utilizes the function *"LMS.m"* (see Section 11.6.2).

```
% Figures 11.17 and 11.18
close all; clear all
N = 501;
mu = 0.1; % convergence parameter
L = 20; % FIR filter order
B = zeros(1,L+1); % FIR coefficients
sigma = 2; %Initial estimate for noise power
alpha = .00; % forgetting factor
 k = 1:N;
noise = rand(1, length(k)) - .5; % Random noise
D = sqrt(2)*sin(2*pi*k/20);
X = D + sqrt(7)*noise;
Y = LMS(X, D, B, mu, sigma, alpha);
subplot(3,1,1)
plot(D,'linewidth',1); xlim([0 501]); grid on;
ylabel('Desired response'); title('\mu = 0.1; \alpha = 0.')
subplot(3,1,2)
plot(X,'linewidth',1); xlim([0 501]); grid on;
ylabel('Corrupted signal')
subplot(3,1,3)
plot(Y,'linewidth',1); xlim([0 501]); grid on;
xlabel('time in sec');
ylabel('LMS output')
```

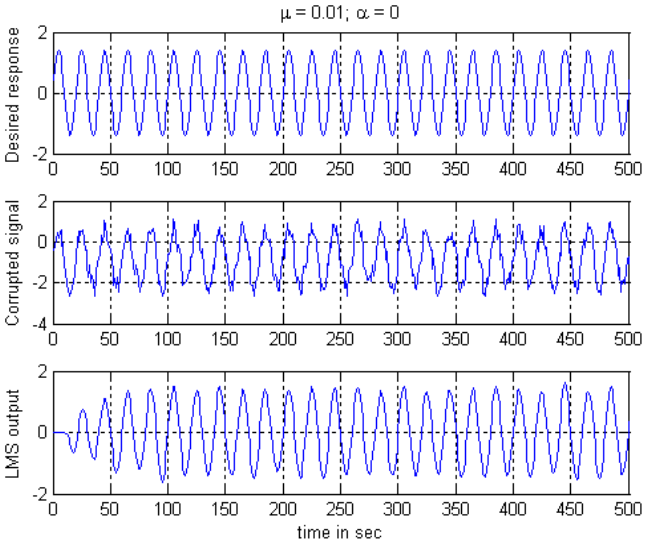


Figure 11.17. Input signal, desired response, and output response of an LMS filter.

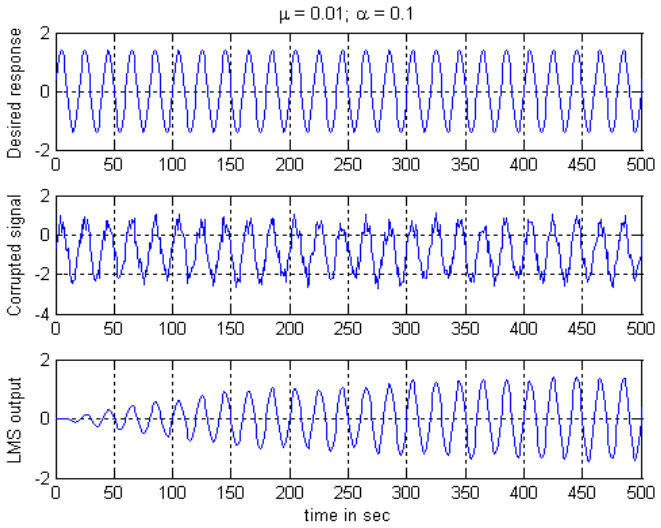


Figure 11.18. Input signal, desired response, and output response of an LMS filter.

11.5.2. The LMS Adaptive Array Processing

Consider the LMS adaptive array shown in Fig. 11.19. The difference between the reference signal and the array output constitutes an error signal. The error signal is then used to adaptively calculate the complex weights, using a predetermined convergence algorithm. The reference signal is assumed to be an accurate approximation of the desired signal (or desired array response). This reference signal can be computed using a training sequence or spreading code which is supposed to be known at the radar receiver. The format of this reference signal will vary from one application to another. But in all cases, the reference signal is assumed to be correlated with the desired signal. An increased amount of this correlation significantly enhances the accuracy and speed of the convergence algorithm being used. In this section, the LMS algorithm is assumed.

In general, the complex envelope of a bandpass signal and its corresponding analytical (pre-envelope) signal can be written using the quadrature components pair $(x_I(t), x_Q(t))$. Recall that the quadrature components are related using the Hilbert transform as follows:

$$x_Q(t) = \hat{x}_I(t) \tag{11.69}$$

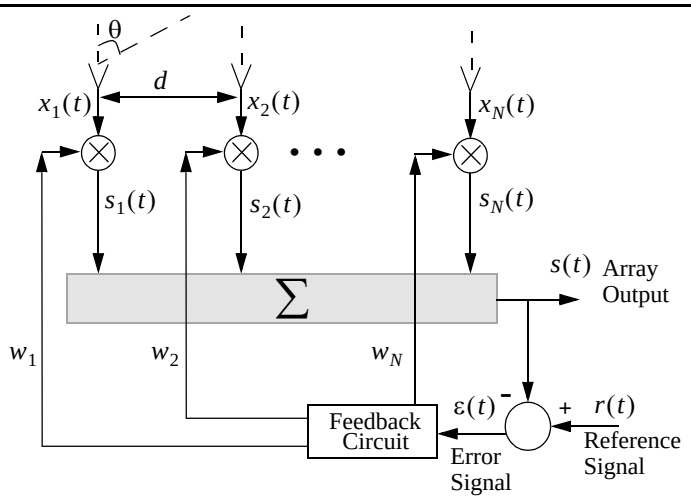


Figure 11.19. A linear adaptive array.

where $\hat{x}_I$ is the Hilbert transform of x_I. A bandpass signal $x(t)$ can be expressed as follows (see Chapter 2):

$$x(t) = x_I(t)\cos 2\pi f_0 t - x_Q(t)\sin 2\pi f_0 t \tag{11.70}$$

$$\psi(t) = x(t) + j\hat{x}(t) \equiv \tilde{x}(t)e^{j2\pi f_0 t} \tag{11.71}$$

$$\tilde{x}(t) = x_I(t) + jx_Q(t) \tag{11.72}$$

where $\psi(t)$ is the pre-envelope and $\tilde{x}(t)$ is the complex envelope. Equation (11.72) can be written using Eq. (11.69) as

$$\tilde{x}(t) = x_I(t) + jx_Q(t) = x_I(t) + j\hat{x}_I(t) \tag{11.73}$$

Using this notation, the adaptive array output signal, its reference signal, and the error signal can also be written using the same notation as

$$\tilde{s}(t) = s(t) + j\hat{s}(t) \tag{11.74}$$

$$\tilde{r}(t) = r(t) + j\hat{r}(t) \tag{11.75}$$

$$\tilde{\varepsilon}(t) = \varepsilon(t) + j\hat{\varepsilon}(t) \tag{11.76}$$

Referencing Fig. 11.19, denote the output of the n^{th} array input signal as $s_n(t)$ and assume complex weights given by

$$w_n = w_{nI} + jw'_{nQ} = w_{nI} - jw_{nQ} \tag{11.77}$$

It follows that

$$s_n(t) = w_{nI}\ x_{nI}(t) + w_{nQ}\ x_{nQ}(t) \tag{11.78}$$

Taking the Hilbert transform of Eq. (11.78) yields

$$\hat{s}_n(t) = w_{nI}\ \hat{x}_{nI}(t) + w_{nQ}\ \hat{x}_{nQ}(t) \tag{11.79}$$

By using Eq. (11.67) into Eq. (11.79), one gets

$$\hat{\hat{x}}_n(t) = w_{nI}\ \hat{x}_{nQ}(t) - w_{nQ}\ \hat{x}_{nI}(t) \tag{11.80}$$

The n^{th} channel analytic signal is

$$\psi_n(t) = s_n(t) + j\hat{s}_n(t) \tag{11.81}$$

Substituting Eq. (11.78) and Eq. (11.79) into Eq. (11.80) gives

$$\psi_n(t) = w_{nI}\ x_{nI}(t) + w_{nQ}\ x_{nQ}(t) + j[w_{nI}\ \hat{x}_{nQ}(t) - w_{nQ}\ \hat{x}_{nI}(t)] \tag{11.82}$$

Collecting terms yields, using complex notation,

$$\psi_n(t) = w_n\tilde{x}_n(t) \tag{11.83}$$

Therefore, the output of the entire adaptive array is

$$\tilde{s}(t) = \sum_{n=1}^{N} \tilde{s}_n(t) = \sum_{n=1}^{N} w_n\tilde{x}_n(t) \tag{11.84}$$

which can be written using vector notation as

$$\tilde{\mathsf{s}} = \mathsf{w}^t\tilde{\mathsf{x}} = \tilde{\mathsf{x}}^t\mathsf{w} \tag{11.85}$$

where the vectors $\tilde{\mathsf{x}}$ and w are given by

$$\tilde{\mathsf{x}} = \left[\tilde{x}_1(t)\ \tilde{x}_2(t)\ \ldots\tilde{x}_N(t)\right]^t \tag{11.86}$$

$$\mathsf{w} = \left[w_1\ w_2\ \ldots w_N\right]^t \tag{11.87}$$

The superscript $\{\ ^t\}$ indicates the transpose operation.

As discussed earlier, one common technique to achieving the MMSE of an LMS algorithm is to use steepest descent. Thus, the complex weights in the LMS adaptive array are related as defined in Eq. (11.59). That is,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\nabla_k \tag{11.88}$$

where again μ is the convergence parameter. The subscript k indicates time samples. In this case, the gradient vector ∇_k is defined by

$$\nabla_k = \frac{\partial}{\partial \mathbf{w}_k} E[\tilde{\varepsilon}_k^2] = \left[\frac{\partial}{\partial w_0(k)} E[\tilde{\varepsilon}_k^2] \; \cdots \; \frac{\partial}{\partial w_N(k)} E[\tilde{\varepsilon}_k^2]\right]^t \tag{11.89}$$

Rearranging Eq. (11.88) so that the rate of change between consecutive estimates of the complex weights is on one side of the equation yields

$$\mathbf{w}_{k+1} - \mathbf{w}_k = -\mu \frac{\partial}{\partial \mathbf{w}_k}(E[\tilde{\varepsilon}_k^2]) \tag{11.90}$$

where the middle portion of Eq. (11.89) was also substituted for the gradient vector. In this format, the left hand side of Eq. (11.90) represents the rate of change of the complex weights with respect to time (i.e., the derivative of the weights with respect to time). It follows that

$$\frac{d}{dt}\mathbf{w} = -\mu \frac{\partial}{\partial \mathbf{w}}(E[\tilde{\varepsilon}^2(t)]) \tag{11.91}$$

However, see from Fig. 11.18, that the error signal complex envelope is

$$\tilde{\varepsilon}(t) = \tilde{r}(t) - \sum_{n=1}^{N} w_n \tilde{x}_n(t) \Rightarrow \tilde{\varepsilon}(t) = \tilde{r}(t) - \tilde{\mathbf{x}}^t \mathbf{w} \tag{11.92}$$

It can be shown (see Problem 11.6) that

$$\frac{\partial}{\partial \mathbf{w}} E[\tilde{\varepsilon}^2(t)] = -E[\tilde{\mathbf{x}}^* \tilde{\varepsilon}(t)] \tag{11.93}$$

Therefore, Eq. (11.91) can be written as

$$\frac{d}{dt}\mathbf{w} = \mu E[\tilde{\mathbf{x}}^* \tilde{\varepsilon}(t)] \tag{11.94}$$

substituting Eq. (11.92) into Eq. (11.94) gives

$$\frac{d}{dt}\mathbf{w} = \mu E[\tilde{\mathbf{x}}^* (\tilde{r}(t) - \tilde{\mathbf{x}}^t \mathbf{w})] \tag{11.95}$$

Equivalently,

$$\frac{d}{dt}\mathbf{w} + \mu E[\tilde{\mathbf{x}}^*\tilde{\mathbf{x}}^t]\mathbf{w} = \mu E[\tilde{\mathbf{x}}^*\tilde{r}(t)] \tag{11.96}$$

The covariance matrix is by definition

$$C = E[\tilde{\mathbf{x}}^*\tilde{\mathbf{x}}^t] = \begin{bmatrix} \tilde{x}_1^*\tilde{x}_1 & \tilde{x}_1^*\tilde{x}_2 & \cdots \\ \tilde{x}_2^*\tilde{x}_1 & \tilde{x}_2^*\tilde{x}_2 & \cdots \\ \cdots & \cdots & \end{bmatrix} \tag{11.97}$$

and the reference signal correlation vector $\mathbf{s}$ is

$$\mathbf{s} = E[\tilde{\mathbf{x}}^*\tilde{d}(t)] = E\left[\tilde{x}_1^*\tilde{r} \;\; \tilde{x}_2^*\tilde{r} \;\; \cdots\right]^t \tag{11.98}$$

Using Eq. (11.98) and Eq. (11.97), one can rewrite the differential equation (DE) given Eq. (11.96) as

$$\frac{d}{dt}\mathbf{w} + \mu C\mathbf{w} = \mu\mathbf{s} \tag{11.99}$$

The steady state solution for the DE defined in Eq. (11.99) (provided that the covariance matrix is not singular) is

$$\mathbf{w} = C^{-1}\,\mathbf{S} \tag{11.100}$$

As the size of the covariance matrix increase (i.e., number of channels in the adaptive array) so does the complexity associated with computing the adaptive weights in real time. This is true because computing the inverse of large matrices in real time can be extremely challenging and demands significant amount of computing power. Consequently, the effectiveness of adaptive arrays has been limited to small-sized arrays, where only a few interfering signals can be eliminated (cancelled). Additionally, computing of a good estimate of the covariance matrix in real time is also difficult in practical applications. In order to mitigate that effect, a reasonable estimate for $E\{x_i x_j^*\}$ (the i,j element of the covariance matrix) is derived by averaging m independent samples of data from the same distribution. This approach can be extended to the entire covariance matrix by collecting M independent "snapshots" of data from N channels. Thus, the estimate of the covariance matrix can be given as,

$$\tilde{C} \approx (\tilde{\mathbf{x}}^\dagger\tilde{\mathbf{x}})/M \tag{11.101}$$

The transient solution of Eq. (11.99) (see Problem 11.7) is

$$\mathbf{w}(t) = \sum_{n=1}^{N} \mathbf{p}_i e^{-\mu\lambda_i t} \tag{11.102}$$

where the vectors $\mathbf{p}_i$ are constants that depend on the initial value of $\mathbf{w}(t)$, and λ_i are the eigenvalues of the matrix $\mathbf{C}$. It follows that the complete solution of Eq. (11.99) is

$$\mathbf{w}(t) = \sum_{n=1}^{N} \mathbf{p}_i e^{-\mu\lambda_i t} + \mathbf{C}^{-1}\,\mathbf{s} \tag{11.103}$$

A very common measure of effectiveness of an adaptive array is the ratio of the total output interference power, S_o to the internal noise power, S_n.

Example:

Consider the two-element array in Section 11.4. Assume the desired signal is at directional-sine $sin(\theta_t)$ and the interference signal is at $sin(\theta_i)$. Calculate the adaptive weights so that the interference signal is cancelled.

Solution:

From Fig. 11.19

$$\tilde{x}_1(t) = \tilde{d}_1(t) + \tilde{n}_1(t) + \tilde{I}_1(t)$$

$$\tilde{x}_2(t) = \tilde{d}_2(t) + \tilde{n}_2(t) + \tilde{I}_2(t)$$

where d is the desired response, n is the noise, signal, and I is the interference signal. The noise signal is spatially incoherent, more specifically

$$E[\tilde{n}_i^*(t)\tilde{n}_j(t)] = \begin{cases} 0 & i \neq j \\ \sigma_n^2 & i = j \end{cases}$$

Also

$$E[\tilde{d}_i^*(t)\tilde{n}_i(t)] = 0 \qquad for\ all\ (i,j)$$

The desired signal is

$$\tilde{d}(t) = \tilde{d}_1(t) + \tilde{d}_2(t) = A_d e^{j2\pi f_0 t} e^{j\Theta_d} + A_d e^{j2\pi f_0 t} e^{j\Theta_d} e^{-j\pi \sin\theta_d}$$

where Θ_d is a uniform random variable. The interference signal is

$$\tilde{I}(t) = \tilde{I}_1(t) + \tilde{I}_2(t) = A_i e^{j2\pi f_0 t} e^{j\Theta_i} + A_i e^{j2\pi f_0 t} e^{j\Theta_i} e^{-j\pi \sin\theta_i}$$

where Θ_i is a uniform random variable. Of course the random variables Θ_d and Θ_i are assumed to be statistically independent. In vector format

$$\tilde{\mathbf{x}}_d = A_d e^{j2\pi f_0 t} e^{j\Theta_d} \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_d} \end{bmatrix}$$

$$\tilde{\mathbf{x}}_i = A_i e^{j2\pi f_0 t} e^{j\Theta_i} \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_i} \end{bmatrix}$$

Of course the noise vector is

$$\tilde{\mathbf{x}}_n = \begin{bmatrix} \tilde{n}_1(t) \\ \tilde{n}_2(t) \end{bmatrix}$$

and the reference signal is (this is an assumption so that the desired and reference signal are correlated)

$$\tilde{r}(t) = A_r e^{j2\pi f_0 t} e^{j\Theta_d}$$

Note that the input SNR is

$$SNR_d = A_d^2 / \sigma_n^2$$

and the interference to noise ratio is

$$SNR_i = A_i^2 / \sigma_n^2$$

The input signal can be written using vector notation as

$$\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_d + \tilde{\mathbf{x}}_i + \tilde{\mathbf{x}}_n$$

The covariance matrix is computed from Eq. (11.97) as

$$\mathbf{C} = E[\tilde{\mathbf{x}}_d^* \tilde{\mathbf{x}}_d^t] = \begin{bmatrix} A_d^2 + A_i^2 + \sigma_n^2 & A_d^2 e^{-j\pi \sin\theta_d} + A_i^2 e^{-j\pi \sin\theta_i} \\ A_d^2 e^{j\pi \sin\theta_d} + A_i^2 e^{j\pi \sin\theta_i} & A_d^2 + A_i^2 + \sigma_n^2 \end{bmatrix}$$

In order to compute the covariance matrix eigenvalue, one needs to compute the determinant first

$$|C| = 4A_d^2A_i^2\left(sin\left(\frac{\theta_d + \theta_i}{s}\right)\right)^2 + 2A_d^2\sigma_n^2 + 2A_i^2\sigma_n^2 + \sigma_n^4$$

Thus,

$$C^{-1} = \frac{1}{|C|}\begin{bmatrix} A_d^2 + A_i^2 + \sigma_n^2 & -A_d^2 e^{-j\pi sin\theta_d} - A_i^2 e^{-j\pi sin\theta_i} \\ -A_d^2 e^{j\pi sin\theta_d} - A_i^2 e^{j\pi sin\theta_i} & A_d^2 + A_i^2 + \sigma_n^2 \end{bmatrix}$$

The reference correlation vector is

$$\mathbf{s} = E[\tilde{x}^*\tilde{r}(t)] = A_dA_r\begin{bmatrix} 1 \\ e^{j\pi sin\theta_d} \end{bmatrix}$$

It follows that the weights are

$$\mathbf{w} = \frac{A_dA_r}{|C|}\begin{bmatrix} A_i^2 + \sigma_n^2 - A_i^2 e^{j\pi(sin\theta_d - sin\theta_i)} \\ e^{j\pi sin\theta_d}\{A_i^2 + \sigma_n^2 - A_i^2 e^{j\pi(sin\theta_i - sin\theta_d)}\} \end{bmatrix}$$

11.5.3. Sidelobe Cancelers (SLC)

Sidelobe cancelers typically consist of a main antenna (which can be a phased array or a single element) and one or more auxiliary antennas. The main antenna is referred to as the main channel; it is assumed to be highly directional and is pointed toward the desired signal angular location. The interfering signal is assumed to be located somewhere off the main antenna boresight (in the sidelobes). Because of this configuration the main channel receives returns from both the desired and the interfering signals. However, returns from the interfering signal in the main channel are weak because of the low main antenna sidelobe gain in the direction of the interfering signal. Also the auxiliary antenna returns are primarily from the interfering signal. This is illustrated in Fig. 11.20.

Referring to Fig. 11.20, $\tilde{s}(t)$ is the desired signal, $\tilde{n}(t)$ is the main channel noise signal which is primarily from the interfering signal, while $\tilde{n}'(t)$ is the interfering signal in the auxiliary array. It is assumed that the signals $\tilde{s}(t)$ and $\tilde{n}(t)$ are uncorrelated. It is also assumed that the interfering signal is highly correlated with the noise signal in the main channel. The basic idea behind SLC is to have the adaptive auxiliary channel produce an accurate estimate of the noise signal first, then to subtract that estimate from the main channel signal so that the output signal is mainly the desired signal.

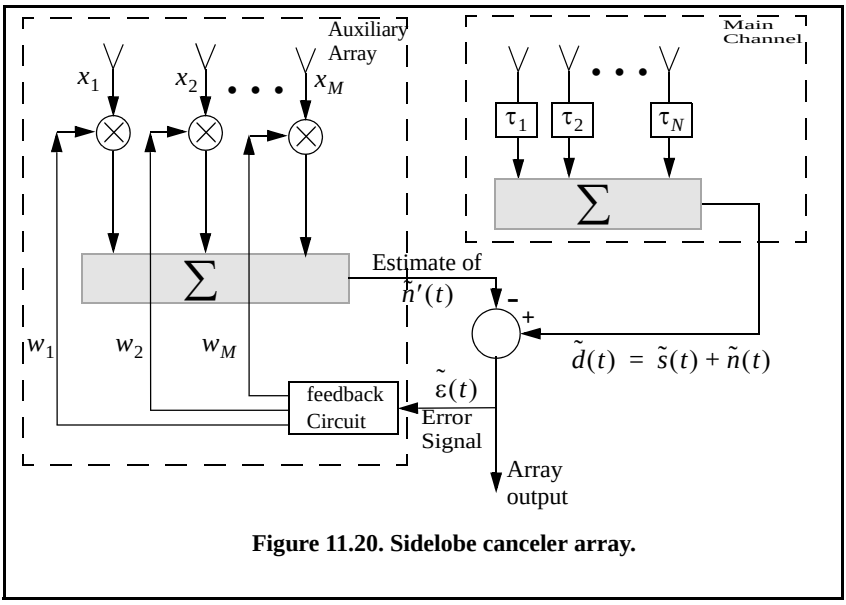


Figure 11.20. Sidelobe canceler array.

The error signal is

$$\tilde{\varepsilon} = \tilde{\mathrm{d}} - \mathrm{w}^t \tilde{\mathrm{x}} \tag{11.104}$$

where $\tilde{\mathrm{x}}$ is the vector of auxiliary array signal, w is the adapted weights. The vector $\tilde{\mathrm{d}}$ of size M. The residual power is

$$P_{res} = E[\varepsilon\varepsilon^{\dagger}] \tag{11.105}$$

$$P_{res} = E[(\tilde{\mathrm{d}} - \mathrm{w}^t \tilde{\mathrm{x}})(\tilde{\mathrm{d}}^* - \tilde{\mathrm{x}}^{\dagger} \mathrm{w}^*)] \tag{11.106}$$

It follows that

$$P_{res} = E[|\tilde{\mathrm{d}}|^2] - E[\tilde{\mathrm{d}}\,\tilde{\mathrm{x}}^{\dagger}\mathrm{w}^*] - E[\tilde{\mathrm{d}}^* \mathrm{w}^t \tilde{\mathrm{x}}] - \mathrm{w}^t E[\tilde{\mathrm{x}}\,\tilde{\mathrm{x}}^{\dagger}]\mathrm{w}^* \tag{11.107}$$

Differentiate the residual power with respect to w and setting the answer equal to zero (to compute the optimal weights that minimize the power residual) yields

$$\frac{\partial P_{res}}{\partial \mathrm{w}} = 0 = -\tilde{\mathrm{x}}\,\tilde{\mathrm{d}} + C_a \mathrm{w} \tag{11.108}$$

where C_a is the covariance matrix of the auxiliary channel. Finally, the optimal weights are given by

$$\mathbf{w} = \mathbf{C}_a^{-1} \tilde{\mathbf{x}}\tilde{\mathbf{d}} \quad \textbf{(11.109)}$$

Note that the vector $\tilde{\mathbf{x}}\tilde{\mathbf{d}}$ represents the components that are common to both main and auxiliary channels. Note that Eq. (11.109) makes intuitive sense where the objective is to isolate the components in the data which are common to the main and auxiliary channels and we then wish to give them some heavy attenuation (which comes from inverting $\mathbf{C}_a$).

11.6. MATLAB Program Listings

This section presents listings for all the MATLAB programs used in this chapter. They are listed in the same order they appear in the text.

11.6.1. MATLAB Function "linear_array.m"

The function *"linear_array.m"* computes and plots the linear array gain pattern as a function of real sine-space. The syntax is as follows:

[theta, patternr, patterng] = linear_array(Nr, dolr, theta0, winid, win, nbits)

where

Symbol	Description	Units	Status
Nr	*number of elements in array*	*none*	*input*
dolr	*element spacing in lambda units*	*wavelengths*	*input*
theta0	*steering angle*	*degrees*	*input*
winid	*-1: No weighting is used* *1: Use weighting defined in win*	*none*	*input*
win	*window for side-lobe control*	*none*	*input*
nbits	*negative #: perfect quantization* *positive #: use* 2^{nbits} *quantization levels*	*none*	*input*
theta	*real angle available for steering*	*degrees*	*output*
patternr	*array pattern*	*dB*	*output*
patterng	*gain pattern*	*dB*	*output*

MATLAB Function** "linear_array.m" **Listing

```
function [theta,patternr,patterng] = linear_array(Nr,dolr,theta0,winid,win,nbits);
% This function computes and returns the gain radiation pattern for a linear array
% It uses the FFT to computes the pattern
%%%% *INPUTS ********** %%%%%%%%%%%%%
```

```
% Nr ==> number of elements; dolr ==> element spacing (d) in lambda units divided
by lambda
% theta0 ==> steering angle in degrees; winid ==> use winid negative for no window,
winid positive to enter your window of size(Nr)
% win is input window, NOTE that win must be an NrX1 row vector; nbits ==> number
of bits used in the pahse shifters
% negative nbits mean no quantization is used
%%%% *OUTPUTS ********** %%%%%%%%%%%%%%%
% theta ==> real-space angle; patternr ==> array radiation pattern in dBs
% patterng ==> array directive gain pattern in dBs
%%%%%%%% ******************** %%%%%%%%%%%
eps = 0.00001;
n = 0:Nr-1;
i = sqrt(-1);
%if dolr is > 0.5 then; choose dol = 0.25 and compute new N
if(dolr <=0.5)
  dol = dolr;
  N = Nr;
else
  ratio = ceil(dolr/.25);
  N = Nr * ratio;
  dol = 0.25;
end
% choose proper size fft, for minimum value choose 256
Nrx = 10 * N;
nfft = 2^(ceil(log(Nrx)/log(2)));
if nfft < 256
   nfft = 256;
end
% convert steering angle into radians; and compute the sine of angle
theta0 = theta0 *pi /180.;
sintheta0 = sin(theta0);
% detrmine and comput quantized steering angle
if nbits < 0
  phase0 = exp(i*2.0*pi .* n * dolr * sintheta0);
else
   % compute and add the phase shift terms (WITH nbits quantization)
   % Use formula thetal = (2*pi*n*dol) * sin(theta0) divided into 2^nbits
   % and rounded to the nearest quantization level
   levels = 2^nbits;
   qlevels = 2.0 * pi / levels; % compute quantization levels
% compute the phase level and round it to the closest quantization level
   angleq = round(dolr .* n * sintheta0 * levels) .* qlevels; % vector of possible angles
   phase0 = exp(i*angleq);
end
% generate array of elements with or without window
if winid < 0
   wr(1:Nr) = 1;
```

```
else
   wr = win';
end
% add the phase shift terms
 wr =  wr .* phase0;
 % determine if interpolation is needed (i.e N > Nr)
if N > Nr
   w(1:N) = 0;
   w(1:ratio:N) = wr(1:Nr);
else
   w = wr;
end
% compute the sine(theta) in real space that correspond to the FFT index
arg = [-nfft/2:(nfft/2)-1] ./ (nfft*dol);
idx = find(abs(arg) <= 1);
sinetheta = arg(idx);
theta = asin(sinetheta);
% convert angle into degrees
theta = theta .* (180.0 / pi);
% Compute fft of w (radiation pattern)
patternv = (abs(fftshift(fft(w,nfft)))).^2;
% convert radiation pattern to dBs
patternr = 10*log10(patternv(idx) ./Nr +  eps);
% Compute directive gain pattern
rbarr  = 0.5 *sum(patternv(idx)) ./ (nfft * dol);
patterng = 10*log10(patternv(idx) + eps) - 10*log10(rbarr + eps);
return
```

11.6.2. MATLAB Function "LMS.m"

The function *"LMS.m"* implements Eq. (11.66). Its syntax is as follows

$$Y = LMS(X, D, B, mu, sigma, alpha)$$

where X is the corrupted sequence, D is the desired response, B is a vector containing the FIR filter coefficients (its initial value can be set to zero), *mu* is the convergence parameter, *sigma* is the SNR, and *alpha* is the forgetting factor.

MATLAB Function** "LMS.m" **Listing

```
function X = LMS(X, D, B, mu, sigma, alpha)
%   This program was written by Stephen Robinson a senior radar
%   engineer at deciBel Research, Inc. in Huntsville, AL
%   X = data vector ; size = 1 x N
%   D = desired signal vector; size = 1 x N
%   N = number of data samples and of adaptive iterations
%   B = adaptive coefficients of Lht order fFIRfilter; size = 1 x L
%   L = order of adaptive system
%   mu = convergence parameter
```

```
% sigma = input signal power estimate
% alpha = exponential forgetting factor
N = size(X,2)
L = size(B,2)-1
px = B;
for k = 1:N
   px(1) = X(k);
   X(k) = sum(B.*px);
   E = D(k) - X(k);
   sigma = alpha*(px(1)^2) + (1 - alpha)*sigma;
   tmp = 2*mu/((L+1)*sigma);
   B = B + tmp*E*px;
   px(L+1:-1:2) = px(L:-1:1);
end
return
```

Problems

11.1. Consider an antenna whose diameter is $d = 3m$. What is the far field requirement for an X-band or an L-band radar that is using this antenna?

11.2. Consider an antenna with electric field intensity in the *xy*-plane $E(\varsigma)$. This electric field is generated by a current distribution $D(y)$ in the *yz*-plane. The electric field intensity is computed using the integral

$$E(\varsigma) = \int_{-r/2}^{r/2} D(y) exp\left(2\pi j \frac{y}{\lambda} sin\varsigma\right) dy$$

where λ is the wavelength and r is the aperture. (a) Write an expression for $E(\varsigma)$ when $D(Y) = d_0$ (a constant). (b) Write an expression for the normalized power radiation pattern and plot it in dB.

11.3. A linear phased array consists of 50 elements with $\lambda/2$ element spacing. (a) Compute the 3dB beam width when the main-beam steering angle is $0°$ and $45°$. (b) Compute the electronic phase difference for any two consecutive elements for steering angle $60°$.

11.4. A linear phased array antenna consists of eight elements spaced with $d = \lambda$ element spacing. (a) Give an expression for the antenna gain pattern (assume no steering and uniform aperture weighting). (b) Sketch the gain pattern versus sine of the off-boresight angle β. What problems do you see is using $d = \lambda$ rather than $d = \lambda/2$?

11.5. In Section 10.4.2 we showed how a DFT can be used to compute the radiation pattern of a linear phased array. Consider a linear of 64 elements at half wavelength spacing, where an FFT of size 512 is used to compute the pattern. What are the FFT bins that correspond to steering angles $\beta = 30°, 45°$?

11.6. Derive Eq. (11.93).

11.7. Compute the transient solution of the DE defined in Eq. (11.99).

11.8. Compute the interference power to the intput power ratio of the example in Section 11.5.3.

11.9. To generate the sum and difference patterns for a linear array of size N follow this algorithm: To form the difference pattern, multiply the first $N/2$ elements by -1 and the second $N/2$ elements by +1. Plot the sum and difference patterns for a linear array of size 60.

11.10. Generate the delta/sum patterns for a 21-element linear array using the form $\frac{\Delta}{\Sigma} = j\frac{V_\Delta}{\sqrt{|V_\Delta|^2 + |V_\Sigma|^2}}$ where V_Δ is the difference voltage pattern and V_Σ is the sum voltage pattern.

Bibliography

Abramowitz, M. and Stegun, I. A., eds., *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, NY, 1970.

Balanis, C. A., *Antenna Theory, Analysis and Design*, Harper & Row, New York, 1982.

Barkat, M., *Signal Detection and Estimation*, Artech House, Norwood, MA, 1991.

Barton, D. K., *Modern Radar System Analysis*, Artech House, Norwood, MA, 1988.

Benedict, T. and Bordner, G., Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations, *IRE Transaction on Automatic Control, Ac-7*, July 1962, pp. 27-32.

Berkowitz, R. S., *Modern Radar - Analysis, Evaluation, and System Design*, John Wiley & Sons, Inc, New York, 1965.

Beyer, W. H., *CRC Standard Mathematical Tables*, 26th edition, CRC Press, Boca Raton, FL, 1981.

Billetter, D. R., *Multifunction Array Radar*, Artech House, Norwood, MA, 1989.

Blackman, S. S., *Multiple-Target Tracking with Radar Application*, Artech House, Norwood, MA, 1986.

Blake, L. V., *A Guide to Basic Pulse-Radar Maximum Range Calculation. Part-I: Equations, Definitions, and Aids to Calculation*, Naval Res. Lab. Report 5868, 1969.

Blake, L. V., *Radar-Range Performance Analysis*, Lexington Books, Lexington, MA, 1980.

Boothe, R. R., *A Digital Computer Program for Determining the Performance of an Acquisition Radar Through Application of Radar Detection Probability Theory*, U.S. Army Missile Command, Report No. RD-TR-64-2, Redstone Arsenal, Alabama, 1964.

Bowman, J. J., Piergiorgio, L. U., and Senior, T. B., *Electromagnetic and Acoustic Scattering by Simple Shapes*, North-Holland Pub. Co, Amsterdam, 1969.

Brookner, E., ed., *Aspects of Modern Radar*, Artech House, Norwood, MA, 1988.

Brookner, E., ed., *Practical Phased Array Antenna System*, Artech House, Norwood, MA, 1991.

Brookner, E., *Radar Technology*, Lexington Books, Lexington, MA, 1996.

Burdic, W. S., *Radar Signal Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1968.

Brookner, E., *Tracking and Kalman Filtering Made Easy*, John Wiley & Sons, New York, 1998.

Cadzow, J. A., *Discrete-Time Systems, An Introduction with Interdisciplinary Applications*, Prentice Hall, Englewood Cliffs, NJ, 1973.

Carlson, A. B., *Communication Systems, An Introduction to Signals and Noise in Electrical Communication*, 3rd edition, McGraw-Hill, New York, 1986.

Carpentier, M. H., *Principles of Modern Radar Systems*, Artech House, Norwood, MA, 1988.

Compton, R. T., *Adaptive Antennas*, Prentice Hall, Englewood Cliffs, NJ, 1988.

Cook, E. C. and Bernfeld, M., *Radar Signals An Introduction to Theory and Application*, Artech House, Norwood, MA, 1993.

Costas, J. P., A Study of a Class of Detection Waveforms Having Nearly Ideal Range-Doppler Ambiguity Properties, *Proc. IEEE 72*, 1984, pp. 996-1009.

Curry, G. R., *Radar System Performance Modeling*, Artech House, Norwood, 2001.

DiFranco, J. V. and Rubin, W. L., *Radar Detection*. Artech House, Norwood, MA, 1980.

Dillard, R. A. and Dillard, G. M., *Detectability of Spread-Spectrum Signals*, Artech House, Norwood, MA, 1989.

Edde, B., *Radar Principles, Technology, Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

Elsherbeni, A., Inman, M. J., and Riley, C., Antenna Design and Radiation Pattern Visualization, *The 19th Annual Review of Progress in Applied Computational Electromagnetics*, ACES'03, Monterey, CA, March 2003.

Fehlner, L. F., *Marcum's and Swerling's Data on Target Detection by a Pulsed Radar*, Johns Hopkins University, Applied Physics Lab. Rpt. # TG451, July 2, 1962, and Rpt. # TG451A, September 1964.

Fielding, J. E. and Reynolds, G. D., *VCCALC: Vertical Coverage Calculation Software and Users Manual*, Artech House, Norwood, MA, 1988.

Gabriel, W. F., Spectral Analysis and Adaptive Array Superresolution Techniques, *Proc. IEEE*, Vol. 68, June 1980, pp. 654-666.

Gelb, A., ed., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

Goldman, S. J., *Phase Noise Analysis in Radar Systems, Using Personal Computers*, John Wiley & Sons, New York, NY, 1989.

Grewal, M. S. and Andrews, A. P., *Kalman Filtering - Theory and Practice Using MATLAB*, 2nd edition, Wiley & Sons Inc., New York, 2001.

Hamming, R. W., *Digital Filters*, 2nd edition, Prentice Hall, Englewood Cliffs, NJ, 1983.

Hanselman, D. and Littlefield, B., *Mastering MATLAB 5, A Complete Tutorial and Reference*, MATLAB Curriculum Series, Prentice Hall, Englewood Cliffs, NJ, 1998.

Hirsch, H. L. and Grove, D. C., *Practical Simulation of Radar Antennas and Radomes*, Artech House, Norwood, MA, 1987.

Hovanessian, S. A., *Radar System Design and Analysis*, Artech House, Norwood, MA, 1984.

James, D. A., *Radar Homing Guidance for Tactical Missiles*, John Wiley & Sons, New York, 1986.

Jin, J., *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 2002.

Kanter, I., Exact Detection Probability for Partially Correlated Rayleigh Targets, *IEEE Trans, AES-22*, March 1986, pp. 184-196.

Kay, S. M., *Fundamentals of Statistical Signal Processing - Estimation Theory*, Volume I, Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, 1993.

Kay, S. M., *Fundamentals of Statistical Signal Processing - Detection Theory*, Volume II, Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, 1993.

Keller, J. B., Geometrical Theory of Diffraction, *Journal Opt. Soc. Amer.*, Vol. 52, February 1962, pp. 116-130.

Klauder, J. R., Price, A. C., Darlington, S., and Albershiem, W. J., The Theory and Design of Chirp Radars, *The Bell System Technical Journal*, Vol. 39, No. 4, 1960.

Klemm, R., *Principles of Space-Time Adaptive Processing*, 3rd Ed, IET, London UK, 2006.

Knott, E. F., Shaeffer, J. F., and Tuley, M. T., *Radar Cross Section*, 2nd edition, Artech House, Norwood, MA, 1993.

Lativa, J., Low-Angle Tracking Using Multifrequency Sampled Aperture Radar, *IEEE-AES Trans.*, Vol. 27, No. 5, September 1991, pp.797-805.

Lee, S. W. and Mittra, R., Fourier Transform of a Polygonal Shape Function and Its Application in Electromagnetics, *IEEE Trans. Antennas and Propagation*, Vol. 31, January 1983, pp. 99-103.

Levanon, N., *Radar Principles*, John Wiley & Sons, New York, 1988.

Levanon, N. and Mozeson, E., Nullifying ACF Grating Lobes in Stepped-frequency Train of LFM Pulses, *IEEE-AES Trans*., Vol. 39, No. 2, April 2003, pp. 694-703.

Levanon, N. and Mozeson, E., *Radar Signals*, John Wiley-Interscience, Hoboken, NJ, 2004.

Lewis, B. L., Kretschmer, Jr., F. F., and Shelton, W. W., *Aspects of Radar Signal Processing*, Artech House, Norwood, MA, 1986.

Long, M. W., *Radar Reflectivity of Land and Sea*, Artech House, Norwood, MA, 1983.

Lothes, R. N., Szymanski, M. B., and Wiley, R. G., *Radar Vulnerability to Jamming*, Artech House, Norwood, MA, 1990.

Mahafza, B. R., *Introduction to Radar Analysis*, CRC Press, Boca Raton, FL, 1998.

Mahafza, B. R., *Radar Systems Analysis and Design Using MATLAB*, 2nd Ed, Taylor & Francis, Boca Raton, FL, 2005.

Mahafza, B. R. and Polge, R. J., Multiple Target Detection Through DFT Processing in a Sequential Mode Operation of Real Two-Dimensional Arrays, *Proc. of the IEEE Southeast Conf.'90*, New Orleans, LA, April 1990, pp. 168-170.

Mahafza, B. R., Heifner, L.A., and Gracchi, V. C., Multitarget Detection Using Synthetic Sampled Aperture Radars (SSAMAR), *IEEE-AES Trans.*, Vol. 31, No. 3, July 1995, pp. 1127-1132.

Mahafza, B. R. and Sajjadi, M., Three-Dimensional SAR Imaging Using a Linear Array in Transverse Motion, *IEEE-AES Trans.*, Vol. 32, No. 1, January 1996, pp. 499-510.

Marchand, P., *Graphics and GUIs with MATLAB*, 2nd edition, CRC Press, Boca Raton, FL, 1999.

Marcum, J. I., A Statistical Theory of Target Detection by Pulsed Radar, Mathematical Appendix, *IRE Trans.*, Vol. IT-6, April 1960, pp. 259-267.

Medgyesi-Mitschang, L. N. and Putnam, J. M., Electromagnetic Scattering from Axially Inhomogenous Bodies of Revolution, *IEEE Trans. Antennas and Propagation*., Vol. 32, August 1984, pp. 797-806.

Meeks, M. L., *Radar Propagation at Low Altitudes*, Artech House, Norwood, MA, 1982.

Melsa, J. L. and Cohn, D. L., *Decision and Estimation Theory*, McGraw-Hill, New York, 1978.

Mensa, D. L., *High Resolution Radar Imaging*, Artech House, Norwood, MA, 1984.

Meyer, D. P. and Mayer, H. A., *Radar Target Detection: Handbook of Theory and Practice*, Academic Press, New York, 1973.

Monzingo, R. A. and Miller, T. W., *Introduction to Adaptive Arrays*, John Wiley & Sons, New York, 1980.

Morchin, W., *Radar Engineer's Sourcebook*, Artech House, Norwood, MA, 1993.

Morris, G. V., *Airborne Pulsed Doppler Radar*, Artech House, Norwood, MA, 1988.

Nathanson, F. E., *Radar Design Principles*, 2nd edition, McGraw-Hill, New York, 1991.

Navarro, Jr., A. M., *General Properties of Alpha Beta and Alpha Beta Gamma Tracking Filters*, Physics Laboratory of the National Defense Research Organization TNO, Report PHL 1977-92, January 1977.

North, D. O., An Analysis of the Factors Which Determine Signal/Noise Discrimination in Pulsed Carrier Systems, *Proc. IEEE 51*, No. 7, July 1963, pp. 1015-1027.

Oppenheim, A. V. and Schafer, R. W., *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

Oppenheim, A. V., Willsky, A. S., and Young, I. T., *Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ, 1983.

Orfanidis, S. J., *Optimum Signal Processing, an Introduction*, 2nd edition, McGraw-Hill, New York, 1988.

Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, 2nd edition, McGraw-Hill, New York, 1984.

Parl, S. A., New Method of Calculating the Generalized Q Function, *IEEE Trans. Information Theory*, Vol. IT-26, No. 1, January 1980, pp. 121-124.

Peebles, Jr., P. Z., *Probability, Random Variables, and Random Signal Principles*, McGraw-Hill, New York, 1987.

Peebles, Jr., P. Z., *Radar Principles*, John Wiley & Sons, New York, 1998.

Pettit, R. H., *ECM and ECCM Techniques for Digital Communication Systems*, Lifetime Learning Publications, New York, 1982.

Polge, R. J., Mahafza, B. R., and Kim, J. G., *Extension and Updating of the Computer Simulation of Range Relative Doppler Processing for MM Wave Seekers*, Interim Technical Report, Vol. I, prepared for the U.S. Army Missile Command, Redstone Arsenal, Alabama, January 1989.

Polge, R. J., Mahafza, B. R., and Kim, J. G., Multiple Target Detection Through DFT Processing in a Sequential Mode Operation of Real or Synthetic Arrays, *IEEE 21st Southeastern Symposium on System Theory*, Tallahassee, FL, 1989, pp. 264-267.

Poularikas, A., *Signals and Systems Primer with MATLAB*, Taylor & Francis, Boca Raton, FL, 2007.

Poularikas, A. and Ramadan, Z. M., *Adaptive Filtering Primer with MATLAB*, Taylor & Francis, Boca Raton, FL, 2006.

Poularikas, A. and Seely, S., *Signals and Systems*, PWS Publishers, Boston, MA, 1984.

Putnam, J. N. and Gerdera, M. B., CARLOS TM: A General-Purpose Three-Dimensional Method of Moments Scattering Code, *IEEE Trans. Antennas and Propagation*, Vol. 35, April 1993, pp. 69-71

Reed, H. R. and Russell, C. M., *Ultra High Frequency Propagation*, Boston Technical Publishers, Inc., Lexington, MA, 1964.

Resnick, J. B., *High Resolution Waveforms Suitable for a Multiple Target Environment*, MS Thesis, MIT, Cambridge, MA, June 1962.

Richards, M. A., *Fundamentals of Radar Signal Processing*, McGraw-Hill, New York, 2005.

Rihaczek, A. W., *Principles of High Resolution Radars*, McGraw-Hill, New York, 1969.

Robertson, G. H., Operating Characteristics for a Linear Detector of CW Signals in Narrow-band Gaussian Noise, *Bell Sys. Tech. Journal*, Vol. 46 April 1967, pp. 755-774.

Ross, R. A., Radar Cross Section of Rectangular Flat Plate as a Function of Aspect Angle, *IEEE Trans*. AP-14, 1966, p. 320.

Ruck, G. T., Barrick, D. E., Stuart, W. D., and Krichbaum, C. K., *Radar Cross Section Handbook*, Volume 1, Plenum Press, New York, 1970.

Ruck, G. T., Barrick, D. E., Stuart, W. D., and Krichbaum, C. K., *Radar Cross Section Handbook*, Volume 2, Plenum Press, New York, 1970.

Rulf, B. and Robertshaw, G. A., *Understanding Antennas for Radar, Communications, and Avionics*, Van Nostrand Reinhold, 1987.

Scanlan, M.J., ed., *Modern Radar Techniques*, Macmillan, New York, 1987.

Scheer, J. A. and Kurtz, J. L., ed., *Coherent Radar Performance Estimation*, Artech House, Norwood, MA, 1993.

Shanmugan, K. S. and Breipohl, A. M., *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, New York, 1988.

Shatz, M. P. and Polychronopoulos, G. H., *An Algorithm for Evaluation of Radar Propagation in the Spherical Earth Diffraction Region*. IEEE Transactions on Antenna and Propagation, VOL. 38, NO.8, August 1990, pp. 1249-1252.

Sherman, S. M., *Monopulse Principles and Techniques*, Artech House, Norwood, MA.

Singer, R. A., Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets, *IEEE Transaction on Aerospace and Electronics, AES-5*, July 1970, pp. 473-483.

Skillman, W. A., *DETPROB: Probability of Detection Calculation Software and User's Manual*, Artech House, Norwood, MA, 1991.

Skolnik, M. I., *Introduction to Radar Systems*, McGraw-Hill, New York, 1982.

Skolnik, M. I., ed., *Radar Handbook*, 2nd edition, McGraw-Hill, New York, 1990.

Song, J. M., Lu, C. C., Chew, W. C., and Lee, S. W., Fast Illinios SolverCode (FSIC), *IEEE Trans. Antennas and Propagation*, Vol. 40, June 1998, pp. 27-34.

Stearns, S. D. and David, R. A., *Signal Processing Algorithms*, Prentice Hall, Englewood Cliffs, NJ, 1988.

Stimson, G. W., *Introduction to Airborne Radar*, Hughes Aircaft Company, El Segundo, CA, 1983.

Stratton, J. A., *Electromagnetic Theory*, McGraw-Hill, New York, 1941.

Stremler, F. G., *Introduction to Communication Systems*, 3rd edition, Addison-Wesley, New York, 1990.

Stutzman, G. E., Estimating Directivity and Gain of Antennas, *IEEE Antennas and Propagation Magazine 40*, August 1998, pp. 7-11.

Swerling, P., Probability of Detection for Fluctuating Targets, *IRE Transaction*

on Information Theory, Vol. IT-6, April 1960, pp. 269-308.

Taflove, A., *Computional Electromagnetics: The Finite-Difference Time-Domain Method*, Artech House, Norwood, MA, 1995.

Van Trees, H. L., *Detection, Estimation, and Modeling Theory*, Part I, Wiley & Sons, Inc., New York, 2001.

Van Trees, H. L., *Detection, Estimation, and Modeling Theory*, Part III, Wiley & Sons, New York, 2001.

Van Trees, H. L., *Optimum Array Processing*, Part IV of *Detection, Estimation, and Modeling Theory*, Wiley & Sons, New York, 2002.

Tzannes, N. S., *Communication and Radar Systems*, Prentice Hall, Englewood Cliffs, NJ, 1985.

Urkowitz, H., *Decision and Detection Theory*, Unpublished Lecture Notes, Lockheed Martin Co., Moorestown, NJ.

Urkowtiz, H., *Signal Theory and Random Processes*, Artech House, Norwood, MA, 1983.

Vaughn, C. R., Birds and Insects as Radar Targets: A Review, *Proc. IEEE*, Vol. 73, No. 2, February 1985, pp. 205-227.

Wehner, D. R., *High Resolution Radar*, Artech House, Norwood, MA, 1987.

Weiner, M. M., ed., *Adaptive Antennas and Recivers*, Taylor & Francis, Boca Raton, FL, 2006.

White, J. E., Mueller, D. D., and Bate, R. R., *Fundamentals of Astrodynamics*, Dover Publications, New York, NY, 1971.

Ziemer, R. E. and Tranter, W. H., *Principles of Communications, Systems, Modulation, and Noise*, 2nd edition, Houghton Mifflin, Boston, MA, 1985.

Zierler, N., *Several Binary-Sequence Generators*, MIT Technical Report No. 95, Sept. 1955.

Index

Z

Electrical Engineering

RADAR SIGNAL ANALYSIS AND PROCESSING USING MATLAB®

Offering radar-related software for the analysis and design of radar waveform and signal processing, **Radar Signal Analysis and Processing Using MATLAB®** provides a comprehensive source of theoretical and practical information on radar signals, signal analysis, and radar signal processing with companion MATLAB® code.

After an overview of radar systems operation and design, the book reviews elements of signal theory relevant to radar detection and radar signal processing, along with random variables and processes. The author then presents the unique characteristic of the matched filter and develops a general formula for the output of the matched filter that is valid for any waveform. He analyzes several analog waveforms, including the linear frequency modulation pulse and stepped frequency waveforms, as well as unmodulated pulse-train, binary, polyphase, and frequency codes. The book explores radar target detection and pulse integration emphasizing the constant false alarm rate. It also covers the stretch processor, the moving target indicator, radar Doppler processing, beamforming, and adaptive array processing.

Using configurable MATLAB® code, this book demonstrates how to apply signal processing to radar applications. It includes many examples and problems to illustrate the practical application of the theory.

Features

- Provides comprehensive coverage of radar signals and signal processing techniques and algorithms
- Presents easy-to-follow mathematical derivations of all equations and formulas
- Includes a complete set of MATLAB® functions and routines, which can be downloaded online
- Contains numerous graphical plots, common radar-related functions, table format outputs, and end-of-chapter problems

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

C6643

ISBN: 978-1-4200-6643-2
90000
9 781420 066432