MATLAB®

Simulations for Radar Systems Design

Bassem R. Mahafza, Ph.D.
Decibel Research, Inc.
Huntsville, Alabama

Atef Z. Elsherbeni
Professor
Electrical Engineering Department
The University of Mississippi
Oxford, Mississippi

CHAPMAN & HALL/CRC

A CRC Press Company
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Mahafza, Bassem R.
MATLAB simulations for radar systems design / Bassem R. Mahafza, Atef Z. Elsherbeni
p. cm.
Includes bibliographical references and index.
ISBN 1-58488-392-8 (alk. paper)
1. Radar–Computer simulation. 2. Radar–Equipment and supplies–Design and construction–Data processing. 3. MATLAB. I. Elsherbeni, Atef Z. II. Title
TK6585.M34 2003
621.3848′01′13—dc22 2003065397

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

No claim to original U.S. Government works
International Standard Book Number 1-58488-392-8
Library of Congress Catalog Number 2003065397
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

To: My wife and four sons;

Wayne and Shirley;

and

in the memory of my parents

Bassem R. Mahafza

To: My wife and children;

my mother;

and

in the memory of my father

Atef Z. Elsherbeni

Preface

The emphasis of *"MATLAB Simulations for Radar Systems Design"* is on radar systems design. However, a strong presentation of the theory is provided so that the reader will be equipped with the necessary background to perform radar systems analysis. The organization of this book is intended to teach a conceptual design process of radars and related trade-off analysis and calculations. It is intended to serve as an engineering reference for radar engineers working in the field of radar systems. The MATLAB®[1] code provided in this book is designed to provide the user with hands-on experience in radar systems, analysis and design.

A radar design case study is introduced in Chapter 1 and carried throughout the text, where the authors' view of how to design this radar is detailed and analyzed. Trade off analyses and calculations are performed. Additionally, several mini design case studies are scattered throughout the book.

"MATLAB Simulations for Radar Systems Design" is divided into two parts: Part I provides a comprehensive description of radar systems, analyses and design. A design case study, which is carried throughout the text, is introduced in Chapter 1. In each chapter the authors' view of how to design the case-study radar is presented based on the theory covered up to that point in the book. As the material coverage progresses through the book, and new theory is discussed, the design case-study requirements are changed and/or updated, and of course the design level of complexity is also increased. This design process is supported by a comprehensive set of MATLAB 6 simulations developed for this purpose. This part will serve as a valuable tool to students and radar engineers in helping them understand radar systems, design process. This includes 1) learning how to go about selecting different radar parameters to meet the design requirements; 2) performing detailed trade-off analysis in the context of radar sizing, modes of operations, frequency selection, waveforms and signal processing; 3) establishing and developing loss and error budgets associated with the design; and 4) generating an in-depth understanding of radar operations and design philosophy. Additionally, Part I includes several mini design case studies pertinent to different chapters in order to help enhance understanding of radar design in the context of the material presented in different chapters.

Part II includes few chapters that cover specialized radar topics, some of which is authored and/or coauthored by other experts in the field. The material

[1] MATLAB is a registered trademark of the The MathWorks, Inc. For product information, please contact: The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098 USA. Web: *www.mathworks.com*.

included in Part II is intended to further enhance the understanding of radar system analysis by providing detailed and comprehensive coverage of these radar related topics. For this purpose, MATLAB 6 code has also been developed and made available.

All MATLAB programs and functions provided in this book can be downloaded from the CRC Press Web site (*www.crcpress.com*). For this purpose, follow this procedure: 1) from your Web browser type *"http://www.crcpress.com"*, 2) click on *"Electronic Products"*, 3) click on *"Download & Updates"*, and finally 4) follow instructions of how to download a certain set of code off that Web page. Furthermore, this MATLAB code can also be downloaded from The MathWorks Web site by following these steps: 1) from your Web browser type: *"http://mathworks.com/matlabcentral/fileexchange/",* 2) place the curser on *"Companion Software for Books"* and click on *"Communications"*. The MATLAB functions and programs developed in this book include all forms of the radar equation: pulse compression, stretch processing, matched filter, probability of detection calculations with all Swerling models, High Range Resolution (HRR), stepped frequency waveform analysis, ghk tracking filter, Kalman filter, phased array antennas, clutter calculations, radar ambiguity functions, ECM, chaff, and many more.

Chapter 1 describes the most common terms used in radar systems, such as range, range resolution, and Doppler frequency. This chapter develops the radar range equation. Finally, a radar design case study entitled *"MyRadar Design Case Study"* is introduced. Chapter 2 is intended to provide an overview of the radar probability of detection calculations and related topics. Detection of fluctuating targets including Swerling I, II, III, and IV models is presented and analyzed. Coherent and non-coherent integration are also introduced. Cumulative probability of detection analysis is in this chapter. Visit 2 of the design case study *"MyRadar"* is introduced.

Chapter 3 reviews radar waveforms, including CW, pulsed, and LFM. High Range Resolution (HRR) waveforms and stepped frequency waveforms are also analyzed. The concept of the Matched Filter (MF) is introduced and analyzed. Chapter 4 presents in detail the principles associated with the radar ambiguity function. This includes the ambiguity function for single pulse, Linear Frequency Modulated pulses, train of unmodulated pulses, Barker codes, and PRN codes. Pulse compression is introduced in Chapter 5. Both the MF and the stretch processors are analyzed.

Chapter 6 contains treatment of the concepts of clutter. This includes both surface and volume clutter. Chapter 7 presents clutter mitigation using Moving Target Indicator (MTI). Delay line cancelers implementation to mitigate the effects of clutter is analyzed.

Chapter 8 presents detailed analysis of Phased Arrays. Linear arrays are investigated and detailed and MATLAB code is developed to calculate and plot the associated array patterns. Planar arrays, with various grid configurations, are also presented.

Chapter 9 discusses target tracking radar systems. The first part of this chapter covers the subject of single target tracking. Topics such as sequential lobing, conical scan, monopulse, and range tracking are discussed in detail. The second part of this chapter introduces multiple target tracking techniques. Fixed gain tracking filters such as the $\alpha\beta$ and the $\alpha\beta\gamma$ filters are presented in detail. The concept of the Kalman filter is introduced. Special cases of the Kalman filter are analyzed in depth.

Chapter 10 is coauthored with Mr. J. Michael Madewell from the US Army Space and Missile Defense Command, in Huntsville, Alabama. This chapter presents an overview of Electronic Counter Measures (ECM) techniques. Topics such as self screening and stand off jammers are presented. Radar chaff is also analyzed and a chaff mitigation technique for Ballistic Missile Defense (BMD) is introduced.

Chapter 11 is concerned with the Radar Cross Section (RCS). RCS dependency on aspect angle, frequency, and polarization is discussed. The target scattering matrix is developed. RCS formulas for many simple objects are presented. Complex object RCS is discussed, and target fluctuation models are introduced. Chapter 12 is coauthored with Dr. Brian Smith from the US Army Aviation and Missile Command (AMCOM), Redstone Arsenal in Alabama. This chapter presents the topic of Tactical Synthetic Aperture Radar (SAR). The topics of this chapter include: SAR signal processing, SAR design considerations, and the SAR radar equation. Finally Chapter 13 presents an overview of signal processing.

Using the material presented in this book and the MATLAB code designed by the authors by any entity or person is strictly at will. The authors and the publisher are neither liable nor responsible for any material or non-material losses, loss of wages, personal or property damages of any kind, or for any other type of damages of any and all types that may be incurred by using this book.

Bassem R. Mahafza
Huntsville, Alabama
July, 2003

Atef Z. Elsherbeni
Oxford, Mississippi
July, 2003

Acknowledgment

The authors first would like to thank God for giving us the endurance and perseverance to complete this work. Many thanks are due to our families who have given up and sacrificed many hours in order to allow us to complete this book. The authors would like to also thank all of our colleagues and friends for their support during the preparation of this book. Special thanks are due to Brian Smith, James Michael Madewell, Patrick Barker, David Hall, Mohamed Al-Sharkawy, and Matthew Inman who have coauthored and/or reviewed some of the material in this reference book.

Table of Contents

PART I

Chapter 1

Introduction to Radar Basics

Chapter 3

Chapter 4

Chapter 5

Pulse Compression

Chapter 6

Surface and Volume Clutter

Chapter 7

Moving Target Indicator (MTI) and Clutter Mitigation

Chapter 8

Phased Arrays

Chapter 9

Target Tracking

Single Target Tracking

Multiple Target Tracking

PART II

Chapter 1

Introduction to Radar Basics

1.1. Radar Classifications

The word radar is an abbreviation for RAdio Detection And Ranging. In general, radar systems use modulated waveforms and directive antennas to transmit electromagnetic energy into a specific volume in space to search for targets. Objects (targets) within a search volume will reflect portions of this energy (radar returns or echoes) back to the radar. These echoes are then processed by the radar receiver to extract target information such as range, velocity, angular position, and other target identifying characteristics.

Radars can be classified as ground based, airborne, spaceborne, or ship based radar systems. They can also be classified into numerous categories based on the specific radar characteristics, such as the frequency band, antenna type, and waveforms utilized. Another classification is concerned with the mission and/or the functionality of the radar. This includes: weather, acquisition and search, tracking, track-while-scan, fire control, early warning, over the horizon, terrain following, and terrain avoidance radars. Phased array radars utilize phased array antennas, and are often called multifunction (multimode) radars. A phased array is a composite antenna formed from two or more basic radiators. Array antennas synthesize narrow directive beams that may be steered mechanically or electronically. Electronic steering is achieved by controlling the phase of the electric current feeding the array elements, and thus the name phased array is adopted.

Radars are most often classified by the types of waveforms they use, or by their operating frequency. Considering the waveforms first, radars can be Continuous Wave (CW) or Pulsed Radars (PR).[1] CW radars are those that continuously emit electromagnetic energy, and use separate transmit and receive antennas. Unmodulated CW radars can accurately measure target radial velocity (Doppler shift) and angular position. Target range information cannot be extracted without utilizing some form of modulation. The primary use of unmodulated CW radars is in target velocity search and track, and in missile guidance. Pulsed radars use a train of pulsed waveforms (mainly with modulation). In this category, radar systems can be classified on the basis of the Pulse Repetition Frequency (PRF) as low PRF, medium PRF, and high PRF radars. Low PRF radars are primarily used for ranging where target velocity (Doppler shift) is not of interest. High PRF radars are mainly used to measure target velocity. Continuous wave as well as pulsed radars can measure both target range and radial velocity by utilizing different modulation schemes.

Table 1.1 has the radar classifications based on the operating frequency.

TABLE 1.1. Radar frequency bands.

Letter designation	Frequency (GHz)	New band designation (GHz)
HF	*0.003 - 0.03*	*A*
VHF	*0.03 - 0.3*	*A<0.25; B>0.25*
UHF	*0.3 - 1.0*	*B<0.5; C>0.5*
L-band	*1.0 - 2.0*	*D*
S-band	*2.0 - 4.0*	*E<3.0; F>3.0*
C-band	*4.0 - 8.0*	*G<6.0; H>6.0*
X-band	*8.0 - 12.5*	*I<10.0; J>10.0*
Ku-band	*12.5 - 18.0*	*J*
K-band	*18.0 - 26.5*	*J<20.0; K>20.0*
Ka-band	*26.5 - 40.0*	*K*
MMW	*Normally >34.0*	*L<60.0; M>60.0*

High Frequency (HF) radars utilize the electromagnetic waves' reflection off the ionosphere to detect targets beyond the horizon. Very High Frequency (VHF) and Ultra High Frequency (UHF) bands are used for very long range Early Warning Radars (EWR). Because of the very large wavelength and the sensitivity requirements for very long range measurements, large apertures are needed in such radar systems.

[1] See Appendix 1A.

Radars in the L-band are primarily ground based and ship based systems that are used in long range military and air traffic control search operations. Most ground and ship based medium range radars operate in the S-band. Most weather detection radar systems are C-band radars. Medium range search and fire control military radars and metric instrumentation radars are also C-band.

The X-band is used for radar systems where the size of the antenna constitutes a physical limitation; this includes most military multimode airborne radars. Radar systems that require fine target detection capabilities and yet cannot tolerate the atmospheric attenuation of higher frequency bands may also be X-band. The higher frequency bands (Ku, K, and Ka) suffer severe weather and atmospheric attenuation. Therefore, radars utilizing these frequency bands are limited to short range applications, such as police traffic radar, short range terrain avoidance, and terrain following radar. Milli-Meter Wave (MMW) radars are mainly limited to very short range Radio Frequency (RF) seekers and experimental radar systems.

1.2. Range

Figure 1.1 shows a simplified pulsed radar block diagram. The time control box generates the synchronization timing signals required throughout the system. A modulated signal is generated and sent to the antenna by the modulator/transmitter block. Switching the antenna between the transmitting and receiving modes is controlled by the duplexer. The duplexer allows one antenna to be used to both transmit and receive. During transmission it directs the radar electromagnetic energy towards the antenna. Alternatively, on reception, it directs the received radar echoes to the receiver. The receiver amplifies the radar returns and prepares them for signal processing. Extraction of target information is performed by the signal processor block. The target's range, R, is computed by measuring the time delay, Δt, it takes a pulse to travel the two-way path between the radar and the target. Since electromagnetic waves travel at the speed of light, $c = 3 \times 10^8 m/\sec$, then

$$R = \frac{c\Delta t}{2} \tag{1.1}$$

where R is in meters and Δt is in seconds. The factor of $\frac{1}{2}$ is needed to account for the two-way time delay.

In general, a pulsed radar transmits and receives a train of pulses, as illustrated by Fig. 1.2. The Inter Pulse Period (IPP) is T, and the pulsewidth is τ. The IPP is often referred to as the Pulse Repetition Interval (PRI). The inverse of the PRI is the PRF, which is denoted by f_r,

$$f_r = \frac{1}{PRI} = \frac{1}{T} \tag{1.2}$$

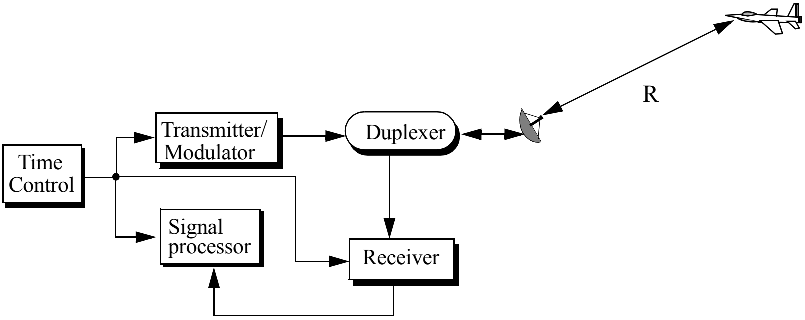


Figure 1.1. A simplified pulsed radar block diagram.

During each PRI the radar radiates energy only for τ seconds and listens for target returns for the rest of the PRI. The radar transmitting duty cycle (factor) d_t is defined as the ratio $d_t = \tau / T$. The radar average transmitted power is

$$P_{av} = P_t \times d_t , \tag{1.3}$$

where P_t denotes the radar peak transmitted power. The pulse energy is $E_p = P_t \tau = P_{av} T = P_{av} / f_r$.

The range corresponding to the two-way time delay T is known as the radar unambiguous range, R_u. Consider the case shown in Fig. 1.3. Echo 1 represents the radar return from a target at range $R_1 = c\Delta t / 2$ due to pulse 1. Echo 2 could be interpreted as the return from the same target due to pulse 2, or it may be the return from a faraway target at range R_2 due to pulse 1 again. In this case,

$$R_2 = \frac{c\Delta t}{2} \qquad or \qquad R_2 = \frac{c(T + \Delta t)}{2} \tag{1.4}$$

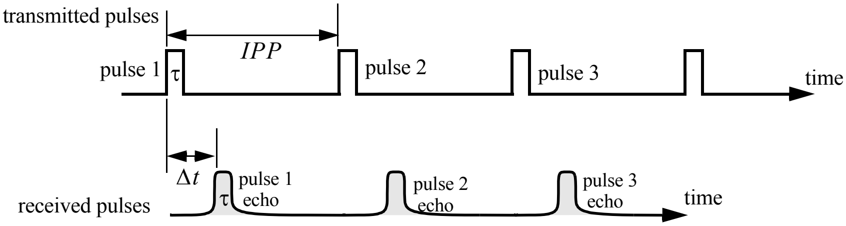


Figure 1.2. Train of transmitted and received pulses.

Clearly, range ambiguity is associated with echo 2. Therefore, once a pulse is transmitted the radar must wait a sufficient length of time so that returns from targets at maximum range are back before the next pulse is emitted. It follows that the maximum unambiguous range must correspond to half of the PRI,

$$R_u = c\frac{T}{2} = \frac{c}{2f_r} \tag{1.5}$$

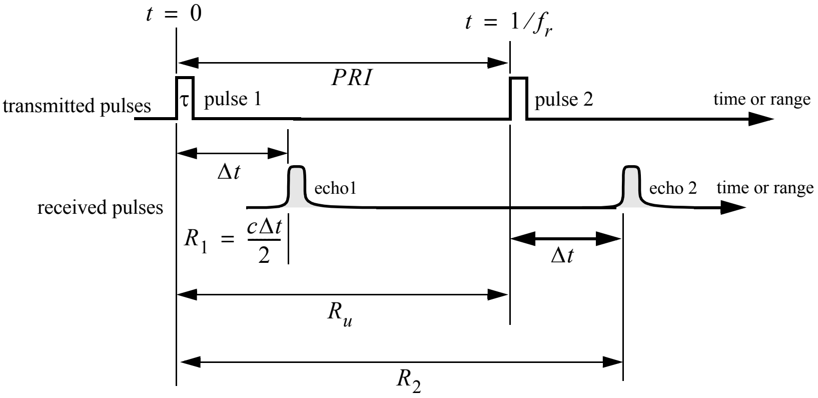


Figure 1.3. Illustrating range ambiguity.

1.3. Range Resolution

Range resolution, denoted as ΔR, is a radar metric that describes its ability to detect targets in close proximity to each other as distinct objects. Radar systems are normally designed to operate between a minimum range R_{min}, and maximum range R_{max}. The distance between R_{min} and R_{max} is divided into M range bins (gates), each of width ΔR,

$$M = (R_{max} - R_{min})/\Delta R \tag{1.6}$$

Targets separated by at least ΔR will be completely resolved in range. Targets within the same range bin can be resolved in cross range (azimuth) utilizing signal processing techniques. Consider two targets located at ranges R_1 and R_2, corresponding to time delays t_1 and t_2, respectively. Denote the difference between those two ranges as ΔR:

$$\Delta R = R_2 - R_1 = c\frac{(t_2 - t_1)}{2} = c\frac{\delta t}{2} \tag{1.7}$$

Now, try to answer the following question: What is the minimum δt such that target 1 at R_1 and target 2 at R_2 will appear completely resolved in range (different range bins)? In other words, what is the minimum ΔR?

First, assume that the two targets are separated by $c\tau/4$, where τ is the pulsewidth. In this case, when the pulse trailing edge strikes target 2 the leading edge would have traveled backwards a distance $c\tau$, and the returned pulse would be composed of returns from both targets (i.e., unresolved return), as shown in Fig. 1.4a. However, if the two targets are at least $c\tau/2$ apart, then as the pulse trailing edge strikes the first target the leading edge will start to return from target 2, and two distinct returned pulses will be produced, as illustrated by Fig. 1.4b. Thus, ΔR should be greater or equal to $c\tau/2$. And since the radar bandwidth B is equal to $1/\tau$, then

$$\Delta R = \frac{c\tau}{2} = \frac{c}{2B} \qquad \textbf{(1.8)}$$

In general, radar users and designers alike seek to minimize ΔR in order to enhance the radar performance. As suggested by Eq. (1.8), in order to achieve fine range resolution one must minimize the pulsewidth. However, this will reduce the average transmitted power and increase the operating bandwidth. Achieving fine range resolution while maintaining adequate average transmitted power can be accomplished by using pulse compression techniques.

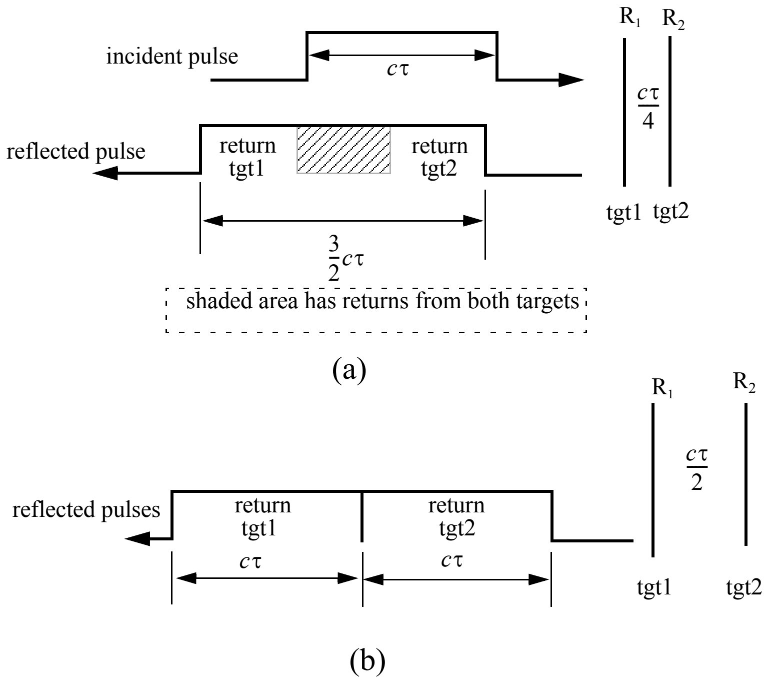


Figure 1.4. (a) Two unresolved targets. (b) Two resolved targets.

1.4. Doppler Frequency

Radars use Doppler frequency to extract target radial velocity (range rate), as well as to distinguish between moving and stationary targets or objects such as clutter. The Doppler phenomenon describes the shift in the center frequency of an incident waveform due to the target motion with respect to the source of radiation. Depending on the direction of the target's motion, this frequency shift may be positive or negative. A waveform incident on a target has equiphase wavefronts separated by λ, the wavelength. A closing target will cause the reflected equiphase wavefronts to get closer to each other (smaller wavelength). Alternatively, an opening or receding target (moving away from the radar) will cause the reflected equiphase wavefronts to expand (larger wavelength), as illustrated in Fig. 1.5.

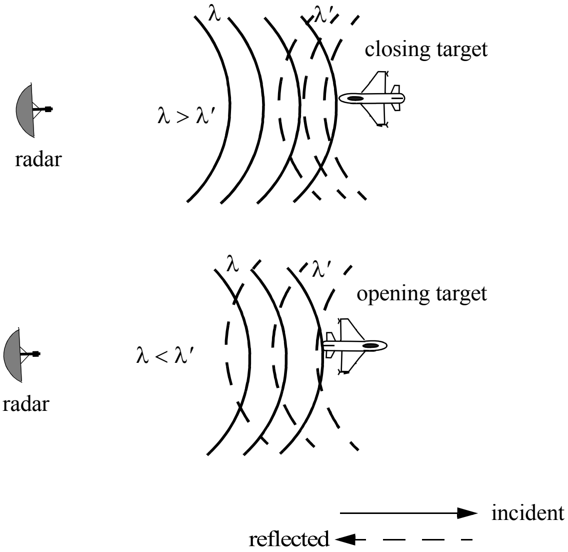


Figure 1.5. Effect of target motion on the reflected equiphase waveforms.

Consider a pulse of width τ (seconds) incident on a target which is moving towards the radar at velocity v, as shown in Fig. 1.6. Define d as the distance (in meters) that the target moves into the pulse during the interval Δt,

$$d = v\Delta t \tag{1.9}$$

where Δt is equal to the time between the pulse leading edge striking the target and the trailing edge striking the target. Since the pulse is moving at the speed of light and the trailing edge has moved distance $c\tau - d$, then

$$c\tau = c\Delta t + v\Delta t \tag{1.10}$$

and

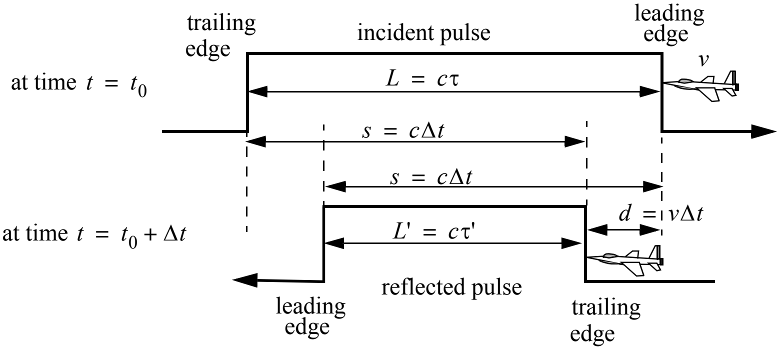


Figure 1.6. Illustrating the impact of target velocity on a single pulse.

$$c\tau' = c\Delta t - v\Delta t \tag{1.11}$$

Dividing Eq. (1.11) by Eq. (1.10) yields,

$$\frac{c\tau'}{c\tau} = \frac{c\Delta t - v\Delta t}{c\Delta t + v\Delta t} \tag{1.12}$$

which after canceling the terms c and Δt from the left and right side of Eq. (1.12) respectively, one establishes the relationship between the incident and reflected pulses widths as

$$\tau' = \frac{c - v}{c + v}\tau \tag{1.13}$$

In practice, the factor $(c - v)/(c + v)$ is often referred to as the time dilation factor. Notice that if $v = 0$, then $\tau' = \tau$. In a similar fashion, one can compute τ' for an opening target. In this case,

$$\tau' = \frac{v + c}{c - v}\tau \tag{1.14}$$

To derive an expression for Doppler frequency, consider the illustration shown in Fig. 1.7. It takes the leading edge of pulse 2 Δt seconds to travel a distance $(c/f_r) - d$ to strike the target. Over the same time interval, the leading edge of pulse 1 travels the same distance $c\Delta t$. More precisely,

$$d = v\Delta t \tag{1.15}$$

$$\frac{c}{f_r} - d = c\Delta t \tag{1.16}$$

solving for Δt yields

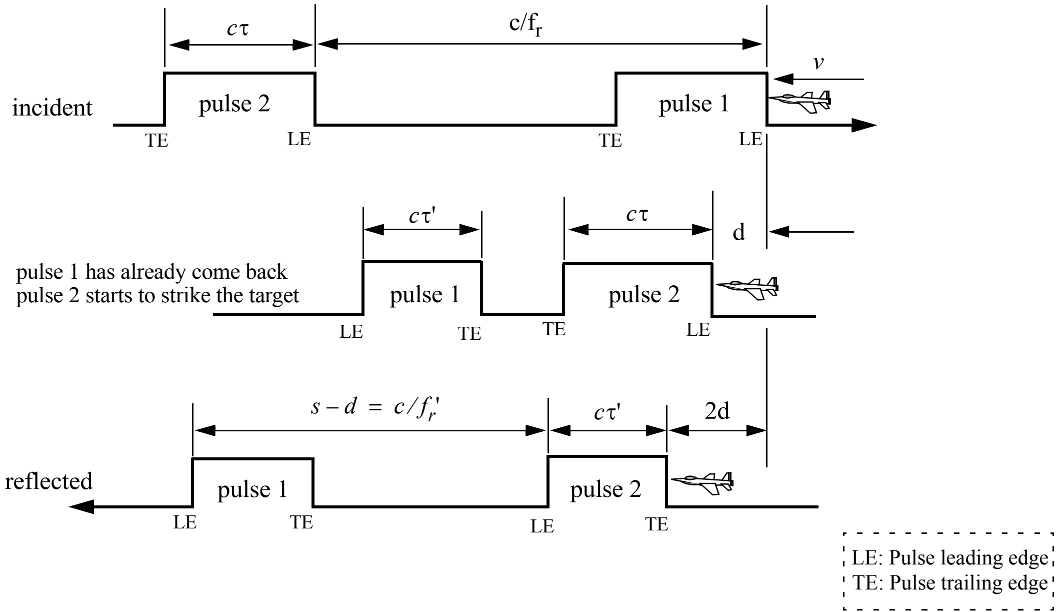


Figure 1.7. Illustration of target motion effects on the radar pulses.

$$\Delta t = \frac{c/f_r}{c+v} \tag{1.17}$$

$$d = \frac{cv/f_r}{c+v} \tag{1.18}$$

The reflected pulse spacing is now $s-d$ and the new PRF is f_r', where

$$s-d = \frac{c}{f_r'} = c\Delta t - \frac{cv/f_r}{c+v} \tag{1.19}$$

It follows that the new PRF is related to the original PRF by

$$f_r' = \frac{c+v}{c-v}\,f_r \tag{1.20}$$

However, since the number of cycles does not change, the frequency of the reflected signal will go up by the same factor. Denoting the new frequency by f_0', it follows

$$f_0' = \frac{c+v}{c-v}\,f_0 \tag{1.21}$$

where f_0 is the carrier frequency of the incident signal. The Doppler frequency f_d is defined as the difference $f_0' - f_0$. More precisely,

$$f_d = f_0' - f_0 = \frac{c+v}{c-v}\,f_0 - f_0 = \frac{2v}{c-v}\,f_0 \tag{1.22}$$

but since $v \ll c$ and $c = \lambda f_0$, then

$$f_d \approx \frac{2v}{c}\,f_0 = \frac{2v}{\lambda} \tag{1.23}$$

Eq. (1.23) indicates that the Doppler shift is proportional to the target velocity, and, thus, one can extract f_d from range rate and vice versa.

The result in Eq. (1.23) can also be derived using the following approach: Fig. 1.8 shows a closing target with velocity v. Let R_0 refer to the range at time t_0 (time reference); then the range to the target at any time t is

$$R(t) = R_0 - v(t-t_0) \tag{1.24}$$

The signal received by the radar is then given by

$$x_r(t) = x(t-\psi(t)) \tag{1.25}$$

where $x(t)$ is the transmitted signal, and

$$\psi(t) = \frac{2}{c}(R_0 - vt + vt_0) \tag{1.26}$$

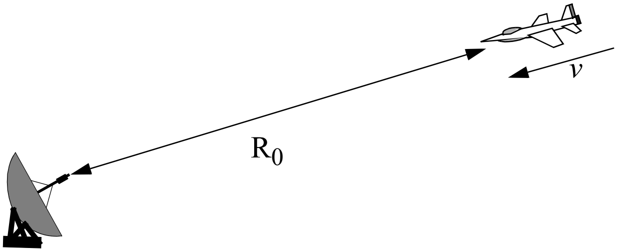


Figure 1.8. Closing target with velocity *v*.

Substituting Eq. (1.26) into Eq. (1.25) and collecting terms yield

$$x_r(t) = x\left(\left(1 + \frac{2v}{c}\right)t - \psi_0\right) \tag{1.27}$$

where the constant phase ψ_0 is

$$\psi_0 = \frac{2R_0}{c} + \frac{2v}{c}\ t_0 \tag{1.28}$$

Define the compression or scaling factor γ by

$$\gamma = 1 + \frac{2v}{c} \tag{1.29}$$

Note that for a receding target the scaling factor is $\gamma = 1 - (2v/c)$. Utilizing Eq. (1.29) we can rewrite Eq. (1.27) as

$$x_r(t) = x(\gamma t - \psi_0) \tag{1.30}$$

Eq. (1.30) is a time-compressed version of the return signal from a stationary target ($v = 0$). Hence, based on the scaling property of the Fourier transform, the spectrum of the received signal will be expanded in frequency to a factor of γ.

Consider the special case when

$$x(t) = y(t)\cos\omega_0 t \tag{1.31}$$

where ω_0 is the radar center frequency in radians per second. The received signal $x_r(t)$ is then given by

$$x_r(t) = y(\gamma t - \psi_0)\cos(\gamma\omega_0 t - \psi_0) \tag{1.32}$$

The Fourier transform of Eq. (1.32) is

$$X_r(\omega) = \frac{1}{2\gamma}\left(Y\left(\frac{\omega}{\gamma} - \omega_0\right) + Y\left(\frac{\omega}{\gamma} + \omega_0\right)\right) \tag{1.33}$$

where for simplicity the effects of the constant phase ψ_0 have been ignored in Eq. (1.33). Therefore, the bandpass spectrum of the received signal is now centered at $\gamma\omega_0$ instead of ω_0. The difference between the two values corresponds to the amount of Doppler shift incurred due to the target motion,

$$\omega_d = \omega_0 - \gamma\omega_0 \tag{1.34}$$

ω_d is the Doppler frequency in radians per second. Substituting the value of γ in Eq. (1.34) and using $2\pi f = \omega$ yield

$$f_d = \frac{2v}{c} f_0 = \frac{2v}{\lambda} \tag{1.35}$$

which is the same as Eq. (1.23). It can be shown that for a receding target the Doppler shift is $f_d = -2v/\lambda$. This is illustrated in Fig. 1.9.

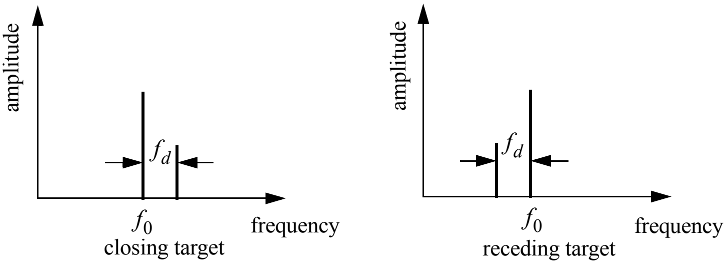


Figure 1.9. Spectra of received signal showing Doppler shift.

In both Eq. (1.35) and Eq. (1.23) the target radial velocity with respect to the radar is equal to v, but this is not always the case. In fact, the amount of Doppler frequency depends on the target velocity component in the direction of the radar (radial velocity). Fig. 1.10 shows three targets all having velocity v: target 1 has zero Doppler shift; target 2 has maximum Doppler frequency as defined in Eq. (1.35). The amount of Doppler frequency of target 3 is $f_d = 2v\cos\theta/\lambda$, where $v\cos\theta$ is the radial velocity; and θ is the total angle between the radar line of sight and the target.

Thus, a more general expression for f_d that accounts for the total angle between the radar and the target is

$$f_d = \frac{2v}{\lambda}\cos\theta \tag{1.36}$$

and for an opening target

$$f_d = \frac{-2v}{\lambda}\cos\theta \tag{1.37}$$

where $\cos\theta = \cos\theta_e \ \cos\theta_a$. The angles θ_e and θ_a are, respectively, the elevation and azimuth angles; see Fig. 1.11.

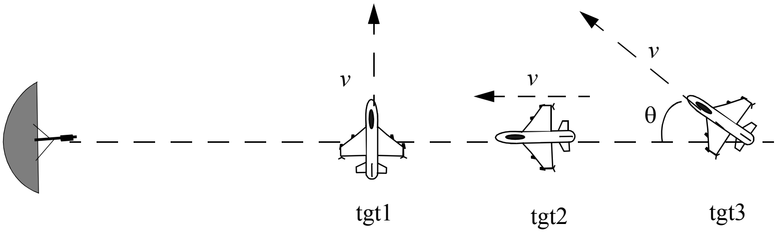


Figure 1.10. Target 1 generates zero Doppler. Target 2 generates maximum Doppler. Target 3 is in between.

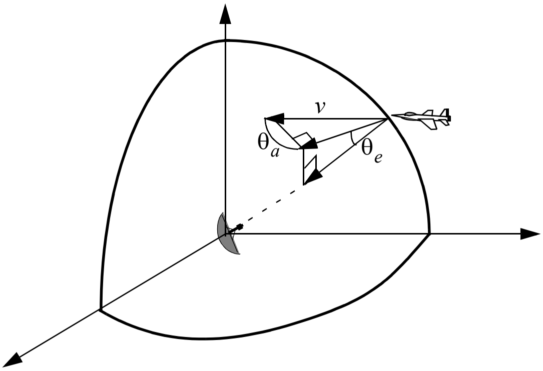


Figure 1.11. Radial velocity is proportional to the azimuth and elevation angles.

1.5. The Radar Equation

Consider a radar with an omni directional antenna (one that radiates energy equally in all directions). Since these kinds of antennas have a spherical radiation pattern, we can define the peak power density (power per unit area) at any point in space as

$$P_D = \frac{Peak\ \ transmitted\ \ power}{area\ \ of\ \ a\ \ sphere} \qquad \frac{watts}{m^2} \tag{1.38}$$

The power density at range R away from the radar (assuming a lossless propagation medium) is

$$P_D = \frac{P_t}{4\pi R^2} \tag{1.39}$$

where P_t is the peak transmitted power and $4\pi R^2$ is the surface area of a sphere of radius R. Radar systems utilize directional antennas in order to increase the power density in a certain direction. Directional antennas are usually characterized by the antenna gain G and the antenna effective aperture A_e. They are related by

$$G = \frac{4\pi A_e}{\lambda^2} \tag{1.40}$$

where λ is the wavelength. The relationship between the antenna's effective aperture A_e and the physical aperture A is

$$\begin{gathered} A_e = \rho A \\ 0 \le \rho \le 1 \end{gathered} \tag{1.41}$$

ρ is referred to as the aperture efficiency, and good antennas require $\rho \to 1$. In this book we will assume, unless otherwise noted, that A and A_e are the same. We will also assume that antennas have the same gain in the transmitting and receiving modes. In practice, $\rho = 0.7$ is widely accepted.

The gain is also related to the antenna's azimuth and elevation beamwidths by

$$G = k\frac{4\pi}{\theta_e\theta_a} \tag{1.42}$$

where $k \le 1$ and depends on the physical aperture shape; the angles θ_e and θ_a are the antenna's elevation and azimuth beamwidths, respectively, in radians. An excellent approximation of Eq. (1.42) introduced by Stutzman and reported by Skolnik is

$$G \approx \frac{26000}{\theta_e\theta_a} \tag{1.43}$$

where in this case the azimuth and elevation beamwidths are given in degrees.

The power density at a distance R away from a radar using a directive antenna of gain G is then given by

$$P_D = \frac{P_t G}{4\pi R^2} \tag{1.44}$$

When the radar radiated energy impinges on a target, the induced surface currents on that target radiate electromagnetic energy in all directions. The amount of the radiated energy is proportional to the target size, orientation, physical shape, and material, which are all lumped together in one target-specific parameter called the Radar Cross Section (RCS) denoted by σ.

The radar cross section is defined as the ratio of the power reflected back to the radar to the power density incident on the target,

$$\sigma = \frac{P_r}{P_D} \; m^2 \tag{1.45}$$

where P_r is the power reflected from the target. Thus, the total power delivered to the radar signal processor by the antenna is

$$P_{Dr} = \frac{P_t G \sigma}{(4\pi R^2)^2} A_e \tag{1.46}$$

Substituting the value of A_e from Eq. (1.40) into Eq. (1.46) yields

$$P_{Dr} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{1.47}$$

Let S_{min} denote the minimum detectable signal power. It follows that the maximum radar range R_{max} is

$$R_{max} = \left(\frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 S_{min}} \right)^{1/4} \tag{1.48}$$

Eq. (1.48) suggests that in order to double the radar maximum range one must increase the peak transmitted power P_t sixteen times; or equivalently, one must increase the effective aperture four times.

In practical situations the returned signals received by the radar will be corrupted with noise, which introduces unwanted voltages at all radar frequencies. Noise is random in nature and can be described by its Power Spectral Density (PSD) function. The noise power N is a function of the radar operating bandwidth, B. More precisely

$$N = Noise \;\; PSD \times B \tag{1.49}$$

The input noise power to a lossless antenna is

$$N_i = kT_e B \tag{1.50}$$

where $k = 1.38 \times 10^{-23}$ $joule/degree \;\; Kelvin$ is Boltzman's constant, and T_e is the effective noise temperature in degrees Kelvin. It is always desirable that the minimum detectable signal (S_{min}) be greater than the noise power. The fidelity of a radar receiver is normally described by a figure of merit called the noise figure F (see Appendix 1B for details). The noise figure is defined as

$$F = \frac{(SNR)_i}{(SNR)_o} = \frac{S_i/N_i}{S_o/N_o} \tag{1.51}$$

$(SNR)_i$ and $(SNR)_o$ are, respectively, the Signal to Noise Ratios (SNR) at the input and output of the receiver. S_i is the input signal power; N_i is the input noise power. S_o and N_o are, respectively, the output signal and noise power. Substituting Eq. (1.50) into Eq. (1.51) and rearranging terms yields

$$S_i = kT_eBF(SNR)_o \tag{1.52}$$

Thus, the minimum detectable signal power can be written as

$$S_{min} = kT_eBF(SNR)_{o_{min}} \tag{1.53}$$

The radar detection threshold is set equal to the minimum output SNR, $(SNR)_{o_{min}}$. Substituting Eq. (1.53) in Eq. (1.48) gives

$$R_{max} = \left(\frac{P_tG^2\lambda^2\sigma}{(4\pi)^3kT_eBF(SNR)_{o_{min}}}\right)^{1/4} \tag{1.54}$$

or equivalently,

$$(SNR)_{o_{min}} = \frac{P_tG^2\lambda^2\sigma}{(4\pi)^3kT_eBFR_{max}^4} \tag{1.55}$$

In general, radar losses denoted as L reduce the overall SNR, and hence

$$(SNR)_o = \frac{P_tG^2\lambda^2\sigma}{(4\pi)^3kT_eBFLR^4} \tag{1.56}$$

Although it may take on many different forms, Eq. (1.56) is what is widely known as the Radar Equation. It is a common practice to perform calculations associated with the radar equation using decibel (dB) arithmetic. A review is presented in Appendix A.

MATLAB Function "radar_eq.m"

The function *"radar_eq.m"* implements Eq. (1.56); it is given in Listing 1.1 in Section 1.10. The syntax is as follows:

[snr] = radar_eq (pt, freq, g, sigma, te, b, nf, loss, range)

where

Symbol	Description	Units	Status
pt	*peak power*	*Watts*	*input*
freq	*radar center frequency*	*Hz*	*input*
g	*antenna gain*	*dB*	*input*
sigma	*target cross section*	m^2	*input*
te	*effective noise temperature*	*Kelvin*	*input*

Symbol	Description	Units	Status
b	*bandwidth*	*Hz*	*input*
nf	*noise figure*	*dB*	*input*
loss	*radar losses*	*dB*	*input*
range	*target range (can be either a single value or a vector)*	*meters*	*input*
snr	*SNR (single value or a vector, depending on the input range)*	*dB*	*output*

The function *"radar_eq.m"* is designed such that it can accept a single value for the input *"range"*, or a vector containing many range values. Figure 1.12 shows some typical plots generated using MATLAB program *"fig1_12.m"* which is listed in Listing 1.2 in Section 1.10. This program uses the function *"radar_eq.m",* with the following default inputs: Peak power $P_t = 1.5MW$, operating frequency $f_0 = 5.6GHz$, antenna gain $G = 45dB$, effective temperature $T_e = 290K$, radar losses $L = 6dB$, noise figure $F = 3dB$. The radar bandwidth is $B = 5MHz$. The radar minimum and maximum detection range are $R_{min} = 25Km$ and $R_{max} = 165Km$. Assume target cross section $\sigma = 0.1m^2$.

Note that one can easily modify the MATLAB function *"radar_eq.m"* so that it solves Eq. (1.54) for the maximum detection range as a function of the minimum required SNR for a given set of radar parameters. Alternatively, the radar equation can be modified to compute the pulsewidth required to achieve a certain SNR for a given detection range. In this case the radar equation can be written as

$$\tau = \frac{(4\pi)^3 kT_eFLR^4 SNR}{P_tG^2\lambda^2\sigma} \tag{1.57}$$

Figure 1.13 shows an implementation of Eq. (1.57) for three different detection range values, using the radar parameters used in MATLAB program *"fig1_13.m"*. It is given in Listing 1.3 in Section 1.10.

When developing radar simulations, Eq. (1.57) can be very useful in the following sense. Radar systems often utilize a finite number of pulsewidths (waveforms) to accomplish all designated modes of operations. Some of these waveforms are used for search and detection, others may be used for tracking, while a limited number of wideband waveforms may be used for discrimination purposes. During the search mode of operation, for example, detection of a certain target with a specific RCS value is established based on a pre-determined probability of detection P_D. The probability of detection, P_D, is used to calculate the required detection SNR (this will be addressed in Chapter 2).

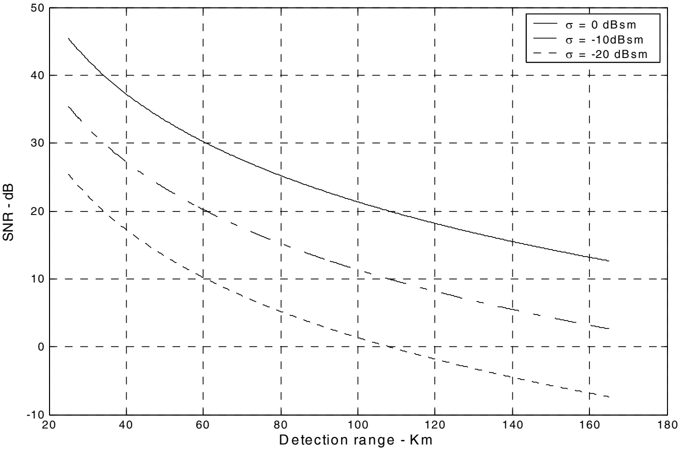


Figure 1.12a. SNR versus detection range for three different values of RCS.

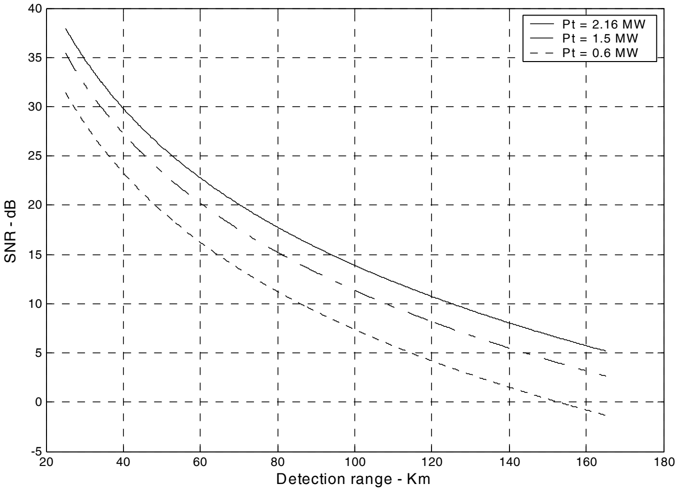


Figure 1.12b. SNR versus detection range for three different values of radar peak power.

Once the required SNR is computed, Eq. (1.57) can then be used to find the most suitable pulse (or waveform) that achieves the required SNR (or equivalently the required P_D). Often, it may be the case that none of the available radar waveforms may be able to guarantee the minimum required SNR for a particular RCS value at a particular detection range. In this case, the radar has to wait until the target is close enough in range to establish detection, otherwise pulse integration (coherent or non-coherent) can be used. Alternatively, cumulative probability of detection can be used. All these issues will be addressed in Chapter 2.

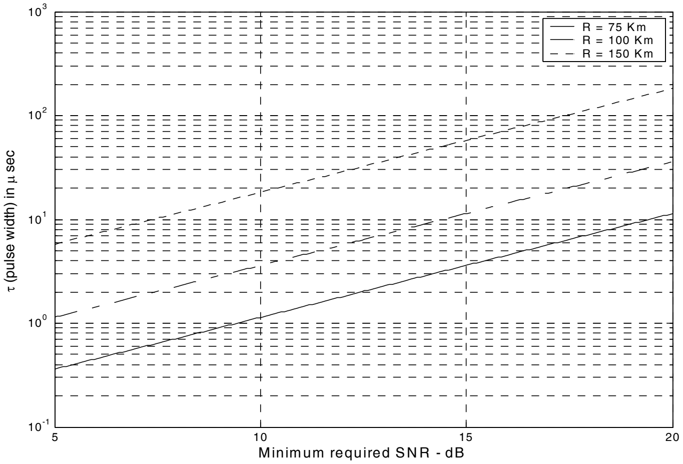


Figure 1.13. Pulsewidth versus required SNR for three different detection range values.

1.5.1. Radar Reference Range

Many radar design issues can be derived or computed based on the radar reference range R_{ref} which is often provided by the radar end user. It simply describes that range at which a certain SNR value, referred to as SNR_{ref}, has to be achieved using a specific reference pulsewidth τ_{ref} for a pre-determined target cross section, σ_{ref}. Radar reference range calculations assume that the target is on the line defined by the maximum antenna gain within a beam (broad side to the antenna). This is often referred to as the radar line of sight, as illustrated in Fig. 1.14.

The radar equation at the reference range is

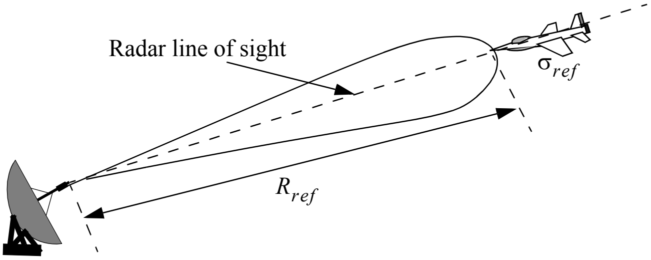


Figure 1.14. Definition of radar line of sight and radar reference range.

$$R_{ref} = \left(\frac{P_t G^2 \lambda^2 \sigma_{ref} \tau_{ref}}{(4\pi)^3 k T_e F L (SNR)_{ref}} \right)^{1/4} \tag{1.58}$$

The radar equation at any other detection range for any other combination of SNR, RCS, and pulsewidth can be given as

$$R = R_{ref} \left(\frac{\tau}{\tau_{ref}} \frac{\sigma}{\sigma_{ref}} \frac{SNR_{ref}}{SNR} \frac{1}{L_p} \right)^{1/4} \tag{1.59}$$

where the additional loss term L_p is introduced to account for the possibility that the non-reference target may not be on the radar line of sight, and to account for other losses associated with the specific scenario. Other forms of Eq. (1.59) can be in terms of the SNR. More precisely,

$$SNR = SNR_{ref} \frac{\tau}{\tau_{ref}} \frac{1}{L_p} \frac{\sigma}{\sigma_{ref}} \left(\frac{R_{ref}}{R} \right)^4 \tag{1.60}$$

As an example, consider the radar described in the previous section, in this case, define $\sigma_{ref} = 0.1 m^2$, $R_{ref} = 86 Km$, and $SNR_{ref} = 20 dB$. The reference pulsewidth is $\tau_{ref} = 0.1 \mu\sec$. Using Eq. (1.60) we compute the SNR at $R = 120 Km$ for a target whose RCS is $\sigma = 0.2 m^2$. Assume that $L_p = 2 dB$ to be equal to $(SNR)_{120Km} = 15.2 dB$. For this purpose, the MATLAB program *"ref_snr.m"* has been developed; it is given in Listing 1.4 in Section 1.10.

1.6. Search (Surveillance)

The first task a certain radar system has to accomplish is to continuously scan a specified volume in space searching for targets of interest. Once detection is established, target information such as range, angular position, and possibly target velocity are extracted by the radar signal and data processors. Depending on the radar design and antenna, different search patterns can be adopted. A two-dimensional (2-D) fan beam search pattern is shown in Fig.1.15a. In this case, the beamwidth is wide enough in elevation to cover the desired search volume along that coordinate; however, it has to be steered in azimuth. Figure 1.15b shows a stacked beam search pattern; here the beam has to be steered in azimuth and elevation. This latter kind of search pattern is normally employed by phased array radars.

Search volumes are normally specified by a search solid angle Ω in steradians. Define the radar search volume extent for both azimuth and elevation as Θ_A and Θ_E. Consequently, the search volume is computed as

$$\Omega = (\Theta_A \Theta_E)/(57.296)^2 \ steradians \tag{1.61}$$

where both Θ_A and Θ_E are given in degrees. The radar antenna $3dB$ beamwidth can be expressed in terms of its azimuth and elevation beamwidths θ_a and θ_e, respectively. It follows that the antenna solid angle coverage is $\theta_a \theta_e$ and, thus, the number of antenna beam positions n_B required to cover a solid angle Ω is

$$n_B = \frac{\Omega}{(\theta_a \theta_e)/(57.296)^2} \tag{1.62}$$

In order to develop the search radar equation, start with Eq. (1.56) which is repeated here, for convenience, as Eq. (1.63)

$$SNR = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_e B F L R^4} \tag{1.63}$$

Using the relations $\tau = 1/B$ and $P_t = P_{av} T/\tau$, where T is the PRI and τ is the pulsewidth, yields

$$SNR = \frac{T}{\tau} \frac{P_{av} G^2 \lambda^2 \sigma \tau}{(4\pi)^3 k T_e F L R^4} \tag{1.64}$$

Figure 1.15. (a) 2-D fan search pattern; (b) stacked search pattern.

Define the time it takes the radar to scan a volume defined by the solid angle Ω as the scan time T_{sc}. The time on target can then be expressed in terms of T_{sc} as

$$T_i = \frac{T_{sc}}{n_B} = \frac{T_{sc}}{\Omega}\theta_a\theta_e \tag{1.65}$$

Assume that during a single scan only one pulse per beam per PRI illuminates the target. It follows that $T_i = T$ and, thus, Eq. (1.64) can be written as

$$SNR = \frac{P_{av}G^2\lambda^2\sigma}{(4\pi)^3kT_eFLR^4}\frac{T_{sc}}{\Omega}\theta_a\theta_e \tag{1.66}$$

Substituting Eqs. (1.40) and (1.42) into Eq. (1.66) and collecting terms yield the search radar equation (based on a single pulse per beam per PRI) as

$$SNR = \frac{P_{av}A_e\sigma}{4\pi kT_eFLR^4}\frac{T_{sc}}{\Omega} \tag{1.67}$$

The quantity $P_{av}A$ in Eq. (1.67) is known as the power aperture product. In practice, the power aperture product is widely used to categorize the radar's ability to fulfill its search mission. Normally, a power aperture product is computed to meet a predetermined SNR and radar cross section for a given search volume defined by Ω.

As a special case, assume a radar using a circular aperture (antenna) with diameter D. The 3-dB antenna beamwidth θ_{3dB} is

$$\theta_{3dB} \approx \frac{\lambda}{D} \tag{1.68}$$

and when aperture tapering is used, $\theta_{3dB} \approx 1.25\lambda/D$. Substituting Eq. (1.68) into Eq. (1.62) yields

$$n_B = \frac{D^2}{\lambda^2}\ \Omega \tag{1.69}$$

For this case, the scan time T_{sc} is related to the time-on-target by

$$T_i = \frac{T_{sc}}{n_B} = \frac{T_{sc}\lambda^2}{D^2\Omega} \tag{1.70}$$

Substitute Eq. (1.70) into Eq. (1.64) to get

$$SNR = \frac{P_{av}G^2\lambda^2\sigma}{(4\pi)^3R^4kT_eFL}\frac{T_{sc}\lambda^2}{D^2\Omega} \tag{1.71}$$

and by using Eq. (1.40) in Eq. (1.71) we can define the search radar equation for a circular aperture as

$$SNR = \frac{P_{av} A \sigma}{16 R^4 k T_e L F} \frac{T_{sc}}{\Omega} \tag{1.72}$$

where the relation $A = \pi D^2 / 4$ (aperture area) is used.

MATLAB Function "power_aperture.m"

The function *"power_aperture.m"* implements the search radar equation given in Eq. (1.67); it is given in Listing 1.5 in Section 1.10. The syntax is as follows:

PAP = power_aperture (snr, tsc, sigma, range, te, nf, loss, az_angle, el_angle)

where

Symbol	**Description**	**Units**	**Status**
snr	*sensitivity snr*	*dB*	*input*
tsc	*scan time*	*seconds*	*input*
sigma	*target cross section*	*m^2*	*input*
range	*target range (can be either single value or a vector)*	*meters*	*input*
te	*effective temperature*	*Kelvin*	*input*
nf	*noise figure*	*dB*	*input*
loss	*radar losses*	*dB*	*input*
az_angle	*search volume azimuth extent*	*degrees*	*input*
el_angle	*search volume elevation extent*	*degrees*	*input*
PAP	*power aperture product*	*dB*	*output*

Plots of the power aperture product versus range and plots of the average power versus aperture area for three RCS choices are shown in Figure 1.16. MATLAB program *"fig1_16.m"* was used to produce these figures. It is given in Listing 1.6 in Section 1.10. In this case, the following radar parameters were used

σ	T_{sc}	$\theta_e = \theta_a$	R	T_e	$nf \times loss$	snr
$0.1 \ m^2$	$2.5 \sec$	$2°$	$250 Km$	$900 K$	$13 dB$	$15 dB$

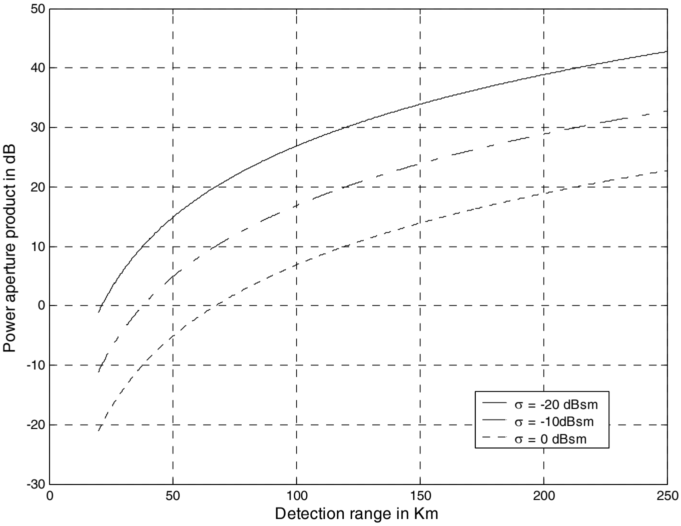


Figure 1.16a. Power aperture product versus detection range.

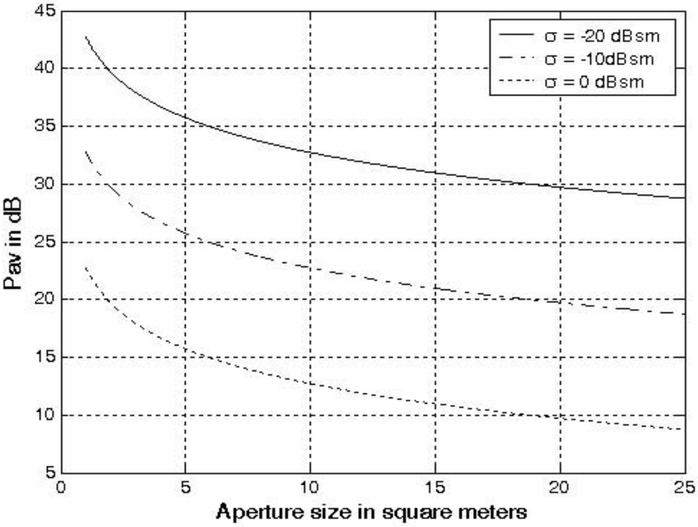


Figure 1.16b. Radar average power versus power aperture product.

Example:

Compute the power aperture product corresponding to the radar that has the following parameters: Scan time $T_{sc} = 2\sec$*, Noise figure* $F = 8dB$*, losses* $L = 6dB$*, search volume* $\Omega = 7.4\ steradians$*, range of interest is* $R = 75Km$*, and the required SNR is* $20dB$*. Assume that* $T_e = 290Kelvin$ *and* $\sigma = 3.162m^2$.

Solution:

Note that $\Omega = 7.4\ steradians$ *corresponds to a search sector that is three fourths of a hemisphere. Thus, using Eq. (1.61) we conclude that* $\theta_a = 180°$ *and* $\theta_e = 135°$*. Using the MATLAB function "power_aperture.m" with the following syntax:*

PAP = power_aperture(20, 2, 3.162, 75e3, 290, 8, 6, 180, 135)

we compute the power aperture product as 36.7 dB.

1.6.1. Mini Design Case Study 1.1

Problem Statement:

Design a ground based radar that is capable of detecting aircraft and missiles at 10 Km and 2 Km altitudes, respectively. The maximum detection range for either target type is 60 Km. Assume that an aircraft average RCS is 6 dBsm, and that a missile average RCS is -10 dBsm. The radar azimuth and elevation search extents are respectively $\Theta_A = 360°$ *and* $\Theta_E = 10°$*. The required scan rate is 2 seconds and the range resolution is 150 meters. Assume a noise figure F = 8 dB, and total receiver noise L = 10 dB. Use a fan beam with azimuth beamwidth less than 3 degrees. The SNR is 15 dB.*

A Design:

The range resolution requirement is $\Delta R = 150m$*; thus by using Eq. (1.8) we calculate the required pulsewidth* $\tau = 1\mu\sec$*, or equivalently require the bandwidth* $B = 1MHz$*. The statement of the problem lends itself to radar sizing in terms of power aperture product. For this purpose, one must first compute the maximum search volume at the detection range that satisfies the design requirements. The radar search volume is*

$$\Omega = \frac{\Theta_A \Theta_E}{(57.296)^2} = \frac{360 \times 10}{(57.296)^2} = 1.097\ steradians \tag{1.73}$$

At this point, the designer is ready to use the radar search equation (Eq. (1.67)) to compute the power aperture product. For this purpose, one can mod-

ify the MATLAB function "power_aperture.m" to compute and plot the power aperture product for both target types. To this end, the MATLAB program "casestudy1_1.m", which is given in Listing 1.7 in Section 1.10, was developed. Use the parameters in Table 1.2 as inputs for this program. Note that the selection of $T_e = 290 Kelvin$ is arbitrary.

TABLE 1.2: Input parameters to MATLAB program "*casestudy1_1.m*".

Symbol	Description	Units	Value
snr	*sensitivity snr*	*dB*	*15*
tsc	*scan time*	*seconds*	*2*
sigma_tgtm	*missile radar cross section*	*dBsm*	*-10*
sigma_tgta	*aircraft radar cross section*	*dBsm*	*6*
rangem	*missile detection range*	*Km*	*60*
rangea	*aircraft detection range*	*Km*	*60*
te	*effective temperature*	*Kelvin*	*290*
nf	*noise figure*	*dB*	*8*
loss	*radar losses*	*dB*	*10*
az_angle	*search volume azimuth extent*	*degrees*	*360*
el_angle	*search volume elevation extent*	*degrees*	*10*

Figure 1.17 shows a plot of the output produced by this program. The same program also calculates the corresponding power aperture product for both the missile and aircraft cases, which can also be read from the plot,

$$\begin{aligned} PAP_{missile} &= 38.53dB \\ PAP_{aircraft} &= 22.53dB \end{aligned} \tag{1.74}$$

Choosing the more stressing case for the design baseline (i.e., select the power-aperture-product resulting from the missile analysis) yields

$$P_{av} \times A_e = 10^{3.853} = 7128.53 \Rightarrow A_e = \frac{7128.53}{Pav} \tag{1.75}$$

Choose $A_e = 1.75m^2$ to calculate the average power as

$$P_{av} = \frac{7128.53}{1.75} = 4.073KW \tag{1.76}$$

and assuming an aperture efficiency of $\rho = 0.8$ yields the physical aperture area. More precisely,

$$A = \frac{A_e}{\rho} = \frac{1.75}{0.8} = 2.1875m^2 \tag{1.77}$$

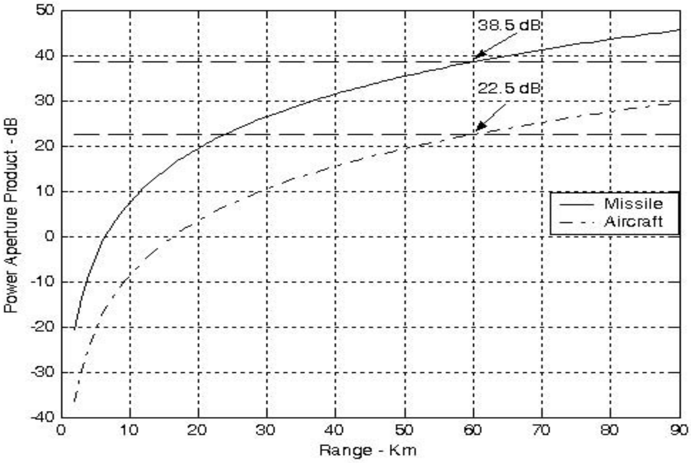


Figure 1.17. Power aperture product versus detection range for radar in mini design case study 1.1.

Use $f_0 = 2.0GHz$ *as the radar operating frequency. Then by using* $A_e = 1.75m^2$ *we calculate using Eq. (1.40)* $G = 29.9dB$*. Now one must determine the antenna azimuth beamwidth. Recall that the antenna gain is also related to the antenna 3-dB beamwidth by the relation*

$$G = \frac{26000}{\theta_e\theta_a} \tag{1.78}$$

where (θ_a, θ_e) *are the antenna 3-dB azimuth and elevation beamwidths, respectively. Assume a fan beam with* $\theta_e = \Theta_E = 15°$*. It follows that*

$$\theta_a = \frac{26000}{\theta_e G} = \frac{26000}{10 \times 977.38} = 2.66° \Rightarrow \theta_a = 46.43mrad \tag{1.79}$$

1.7. 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

$$n_P = \frac{\theta_a T_{sc} f_r}{2\pi} \tag{1.80}$$

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

$$n_P = \frac{\theta_a f_r}{\dot{\theta}_{scan}} \tag{1.81}$$

where $\dot{\theta}_{scan}$ is the antenna scan rate in degrees per second. Note that when using Eq. (1.80), θ_a is expressed in radians, while when using Eq. (1.81) it is expressed in degrees. As an example, consider a radar with an azimuth antenna beamwidth $\theta_a = 3°$, antenna scan rate $\dot{\theta}_{scan} = 45° / \sec$ (antenna scan time, $T_{sc} = 8\sec onds$), and a PRF $f_r = 300Hz$. Using either Eq.s (1.80) or (1.81) 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 pre-detection integration. Coherent integration preserves the phase relationship between the received pulses. Thus a build up in the signal amplitude is achieved. Alternatively, pulse integration performed after the envelope detector (where the phase relation is destroyed) is called non-coherent or post-detection 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 which may introduce signal loss remain the same. This is illustrated in Fig. 1.18, and is normally referred to as antenna beam-shape loss.

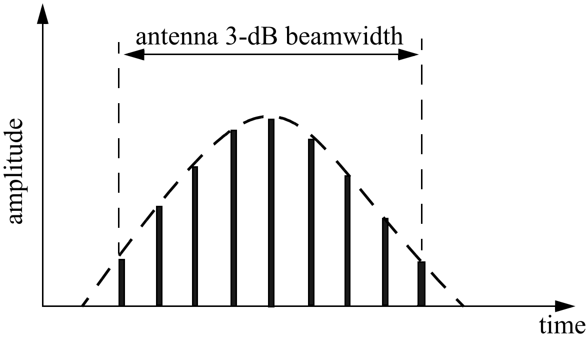


Figure 1.18. Pulse returns from a point target using a rotating (scanning) antenna

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). These topics will be discussed in more detail in other chapters.

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, etc.) 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 build up. Knowledge of target dynamics is not as critical when employing non-coherent 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., avoid 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 statistical models. These topics will be discussed in Chapter 2. However, in this section an overview of pulse integration is introduced in the context of radar measurements as it applies to the radar equation. The basic conclusions presented in this chapter concerning pulse integration will still be valid, in the general sense, when a more comprehensive analysis of pulse integration is presented; however, the exact implementation, the mathematical formulation, and /or the numerical values used will vary.

1.7.1. 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 non-coherent 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

$$(SNR)_{CI} = n_P(SNR)_1 \tag{1.82}$$

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

$$t_{CI} = \sqrt{\lambda/2a_r} \tag{1.83}$$

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.

1.7.2. Non-Coherent Integration

Non-coherent integration is often implemented after the envelope detector, also known as the quadratic detector. Non-coherent integration is less efficient than coherent integration. Actually, the non-coherent integration gain is always smaller than the number of non-coherently integrated pulses. This loss in integration is referred to as post detection or square law detector loss. Marcum and Swerling showed that this loss is somewhere between $\sqrt{n_P}$ and n_P. DiFranco and Rubin presented an approximation of this loss as

$$L_{NCI} = 10\log(\sqrt{n_p}) - 5.5 \quad dB \tag{1.84}$$

Note that as n_P becomes very large, the integration loss approaches $\sqrt{n_P}$.

The subject of integration loss is treated in great levels of detail in the literature. Different authors use different approximations for the integration loss associated with non-coherent integration. However, all these different approximations yield very comparable results. Therefore, in the opinion of these authors the use of one formula or another to approximate integration loss becomes somewhat subjective. In this book, the integration loss approximation reported by Barton and used by Curry will be adopted. In this case, the non-coherent integration loss which can be used in the radar equation is

$$L_{NCI} = \frac{1 + (SNR)_1}{(SNR)_1} \tag{1.85}$$

It follows that the SNR when n_P pulses are integrated non-coherently is

$$(SNR)_{NCI} = \frac{n_P(SNR)_1}{L_{NCI}} = n_P(SNR)_1 \times \frac{(SNR)_1}{1 + (SNR)_1} \tag{1.86}$$

1.7.3. Detection Range with Pulse Integration

The process of determining the radar sensitivity or equivalently the maximum detection range when pulse integration is used is as follows: First, decide whether to use coherent or non-coherent integration. Keep in mind the issues discussed in the beginning of this section when deciding whether to use coherent or non-coherent integration.

Second, determine the minimum required $(SNR)_{CI}$ or $(SNR)_{NCI}$ required for adequate detection and track. Typically, for ground based surveillance radars that can be on the order of 13 to 15 dB. The third step is to determine how many pulses should be integrated. The choice of n_P is affected by the radar scan rate, the radar PRF, the azimuth antenna beamwidth, and of course by the target dynamics (remember that range walk should be avoided or compensated for, so that proper integration is feasible). Once n_P and the required SNR are known one can compute the single pulse SNR (i.e., the reduction in SNR). For this purpose use Eq. (1.82) in the case of coherent integration. In the non-coherent integration case, Curry presents an attractive formula for this calculation, as follows

$$(SNR)_1 = \frac{(SNR)_{NCI}}{2n_P} + \sqrt{\frac{(SNR)^2_{NCI}}{4n_P^2} + \frac{(SNR)_{NCI}}{n_P}} \tag{1.87}$$

Finally, use $(SNR)_1$ from Eq. (1.87) in the radar equation to calculate the radar detection range. Observe that due to the integration reduction in SNR the radar detection range is now larger than that for the single pulse when the same SNR value is used. This is illustrated using the following mini design case study.

1.7.4. Mini Design Case Study 1.2

Problem Statement:

A MMW radar has the following specifications: Center frequency $f = 94GHz$, pulsewidth $\tau = 50 \times 10^{-9}\sec$, peak power $P_t = 4W$, azimuth coverage $\Delta\alpha = \pm 120°$, Pulse repetition frequency $PRF = 10KHz$, noise figure $F = 7dB$; antenna diameter $D = 12in$; antenna gain $G = 47dB$; radar cross section of target is $\sigma = 20m^2$; system losses $L = 10dB$; radar scan time $T_{sc} = 3\sec$. Calculate: The wavelength λ; range resolution ΔR; bandwidth B; antenna half power beamwidth; antenna scan rate; time on target. Compute the range that corresponds to 10 dB SNR. Plot the SNR as a function of range. Finally, compute the number of pulses on the target that can be used for integration and the corresponding new detection range when pulse integration is used, assuming that the SNR stays unchanged (i.e., the same as in the case of a single pulse). Assume $T_e = 290 \ Kelvin$.

A Design:

The wavelength λ *is*

$$\lambda = \frac{c}{f} = \frac{3 \times 10^8}{94 \times 10^9} = 0.00319m$$

The range resolution ΔR *is*

$$\Delta R = \frac{c\tau}{2} = \frac{(3 \times 10^8)(50 \times 10^{-9})}{2} = 7.5m$$

Radar operating bandwidth B *is*

$$B = \frac{1}{\tau} = \frac{1}{50 \times 10^{-9}} = 20MHz$$

The antenna 3-dB beamwidth is

$$\theta_{3dB} = 1.25\frac{\lambda}{D} = 0.7499°$$

Time on target is

$$T_i = \frac{\theta_{3dB}}{\dot{\theta}_{scan}} = \frac{0.7499°}{80°/\sec} = 9.38m\sec$$

It follows that the number of pulses available for integration is calculated using Eq. (1.81),

$$n_P = \frac{\theta_{3dB}}{\dot{\theta}_{scan}}f_r = 9.38 \times 10^{-3} \times 10 \times 10^3 \Rightarrow 94 \ \ pulses$$

Coherent Integration case:

Using the radar equation given in Eq. (1.58) yields $R_{ref} = 2.245Km$*. The SNR improvement due to coherently integrating 94 pulses is 19.73dB. However, since it is requested that the SNR remains at 10dB, we can calculate the new detection range using Eq. (1.59) as*

$$R_{CI}\big|_{n_P = 94} = 2.245 \times (94)^{1/4} = 6.99Km$$

Using the MATLAB Function "radar_eq.m" with the following syntax

[snr] = radar_eq (4, 94e9, 47, 20, 290, 20e6, 7, 10, 6.99e3)

yields SNR = -9.68 dB. This means that using 94 pulses integrated coherently at 6.99 Km where each pulse has a SNR of -9.68 dB provides the same detection criteria as using a single pulse with SNR = 10dB at 2.245Km. This is illustrated in Fig. 1.19, using the MATLAB program "fig1_19.m", which is given in Listing 1.8 in Section 1.10. Figure 1.19 shows the improvement of the detection range if the SNR is kept constant before and after integration.

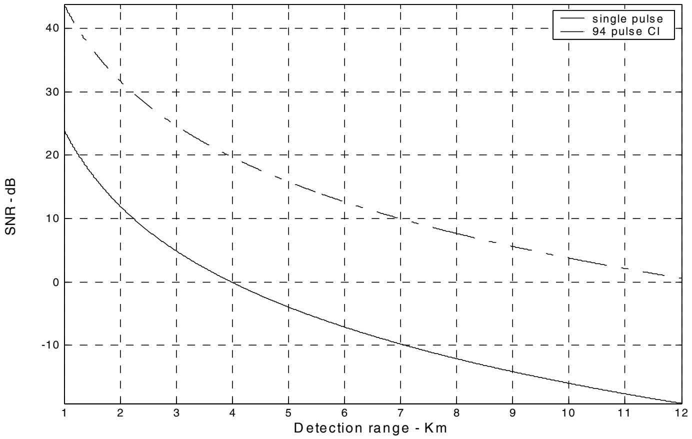


Figure 1.19. SNR versus detection range, using parameters from example.

Non-coherent Integration case:

Start with Eq. (1.87) with $(SNR)_{NCI} = 10dB$ *and* $n_P = 94$*,*

$$(SNR)_1 = \frac{10}{2 \times 94} + \sqrt{\frac{(10)^2}{4 \times 94^2} + \frac{10}{94}} = 0.38366 \Rightarrow -4.16dB$$

Therefore, the single pulse SNR when 94 pulses are integrated non-coherently is -4.16dB. You can verify this result by using Eq. (1.86). The integration loss L_{NCI} *is calculated using Eq. (1.85). It is*

$$L_{NCI} = \frac{1 + 0.38366}{0.38366} = 3.6065 \Rightarrow 5.571dB$$

Therefore, the net non-coherent integration gain is

$$10 \times \log(94) - 5.571 = 14.16dB \Rightarrow 26.06422$$

and, consequently, the maximum detection range is

$$R_{NCI}\big|_{n_P = 94} = 2.245 \times (26.06422)^{1/4} = 5.073Km$$

This means that using 94 pulses integrated non-coherently at 5.073 Km where each pulse has SNR of -4.16dB provides the same detection criterion as using a single pulse with SNR = 10dB at 2.245Km. This is illustrated in Fig. 1.20, using the MATLAB program "fig1_19.m".

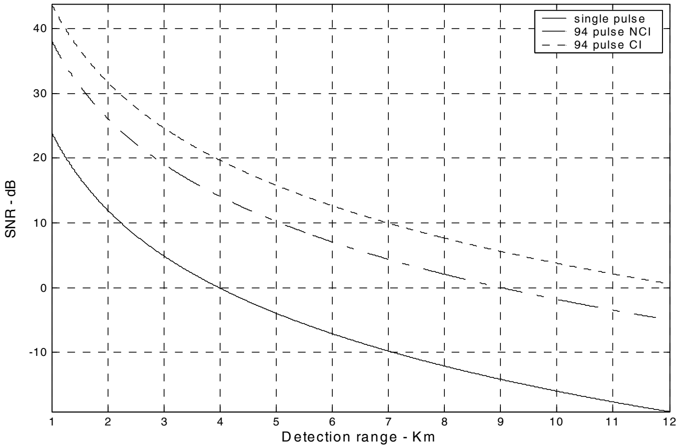


Figure 1.20. SNR versus detection range, for the same example.

MATLAB Function "pulse_integration.m"

Figure 1.21 shows the SNR gain versus the number of integrated pulses for both coherent and non-coherent integration. This figure corresponds to parameters from the previous example at $R = 5.01 Km$. Figure 1.22 shows the general case SNR improvement versus number of integrated pulses. Both figures were generated using MATLAB program *"fig1_21.m"* which is given in Listing 1.9 in Section 1.10. For this purpose the MATLAB function *"pulse_integration.m"* was developed. It is given in Listing 1.10 in Section 1.10. This function calculates the radar equation given in Eq. (1.56) with pulse integration. The syntax for MATLAB function *"pulse_integration.m"* is as follows

[snr] = pulse_integration (pt, freq, g, sigma, te, b, nf, loss, range, np, ci_nci)

where

Symbol	Description	Units	Status
pt	*peak power*	*Watts*	*input*
freq	*radar center frequency*	*Hz*	*input*
g	*antenna gain*	*dB*	*input*
sigma	*target cross section*	m^2	*input*
te	*effective noise temperature*	*Kelvin*	*input*
b	*bandwidth*	*Hz*	*input*

Symbol	Description	Units	Status
nf	*noise figure*	*dB*	*input*
loss	*radar losses*	*dB*	*input*
range	*target range (can be either a single value or a vector)*	*meters*	*input*
np	*number of integrated pulses*	*none*	*input*
ci_nci	*1 for CI; 2 for NCI*	*none*	*input*
snr	*SNR (single value or a vector, depending on the input range)*	*dB*	*output*

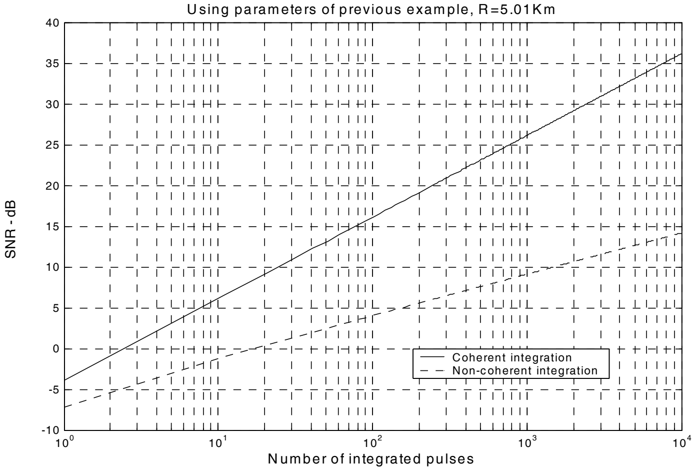


Figure 1.21. SNR improvement when integration is utilized.

1.8. Radar Losses

As indicated by the radar equation, the receiver SNR is inversely proportional to the radar losses. Hence, any increase in radar losses causes a drop in the SNR, thus decreasing the probability of detection, as it is a function of the SNR. Often, the principal difference between a good radar design and a poor radar design is the radar losses. Radar losses include ohmic (resistance) losses and statistical losses. In this section we will briefly summarize radar losses.

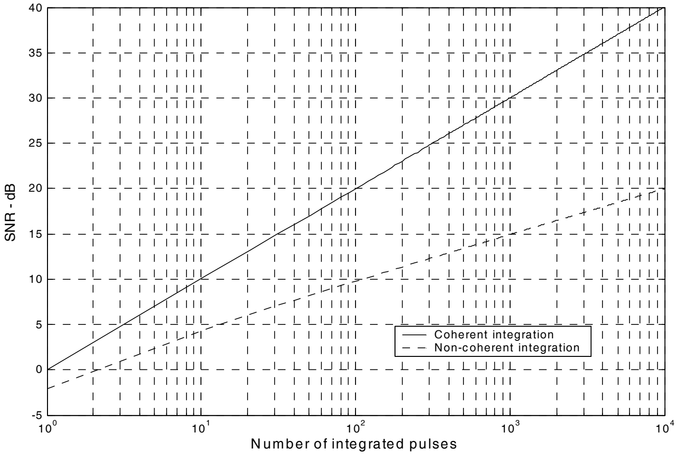


Figure 1.22. SNR improvement when integration is utilized.

1.8.1. Transmit and Receive Losses

Transmit and receive losses occur between the radar transmitter and antenna input port, and between the antenna output port and the receiver front end, respectively. Such losses are often called plumbing losses. Typically, plumbing losses are on the order of 1 to 2 dB.

1.8.2. Antenna Pattern Loss and Scan Loss

So far, when we used the radar equation we assumed maximum antenna gain. This is true only if the target is located along the antenna's boresight axis. However, as the radar scans across a target the antenna gain in the direction of the target is less than maximum, as defined by the antenna's radiation pattern. The loss in SNR due to not having maximum antenna gain on the target at all times is called the antenna pattern (shape) loss. Once an antenna has been selected for a given radar, the amount of antenna pattern loss can be mathematically computed.

For example, consider a $\sin x / x$ antenna radiation pattern as shown in Fig. 1.23. It follows that the average antenna gain over an angular region of $\pm\theta/2$ about the boresight axis is

$$G_{av} \approx 1 - \left(\frac{\pi r}{\lambda}\right)^2 \frac{\theta^2}{36} \tag{1.88}$$

where r is the aperture radius and λ is the wavelength. In practice, Gaussian antenna patterns are often adopted. In this case, if θ_{3dB} denotes the antenna 3dB beamwidth, then the antenna gain can be approximated by

$$G(\theta) = \exp\left(-\frac{2.776\theta^2}{\theta_{3dB}^2}\right) \tag{1.89}$$

If the antenna scanning rate is so fast that the gain on receive is not the same as on transmit, additional scan loss has to be calculated and added to the beam shape loss. Scan loss can be computed in a similar fashion to beam shape loss. Phased array radars are often prime candidates for both beam shape and scan losses.

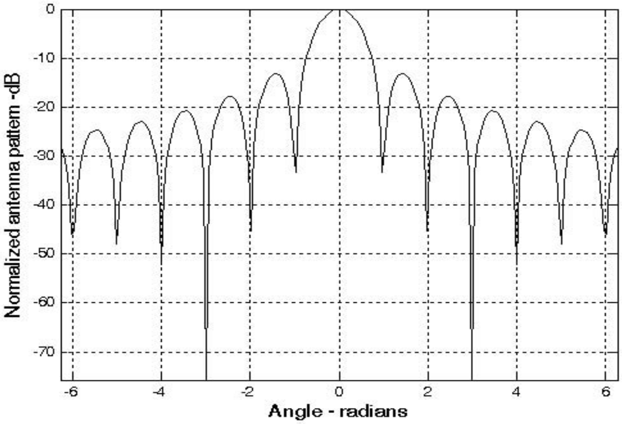


Figure 1.23. Normalized (sin x / x) antenna pattern.

1.8.3. Atmospheric Loss

Detailed discussion of atmospheric loss and propagation effects is in a later chapter. Atmospheric attenuation is a function of the radar operating frequency, target range, and elevation angle. Atmospheric attenuation can be as high as a few dB.

1.8.4. Collapsing Loss

When the number of integrated returned noise pulses is larger than the target returned pulses, a drop in the SNR occurs. This is called collapsing loss. The collapsing loss factor is defined as

$$\rho_c = \frac{n+m}{n} \tag{1.90}$$

where n is the number of pulses containing both signal and noise, while m is the number of pulses containing noise only. Radars detect targets in azimuth, range, and Doppler. When target returns are displayed in one coordinate, such as range, noise sources from azimuth cells adjacent to the actual target return converge in the target vicinity and cause a drop in the SNR. This is illustrated in Fig. 1.24.

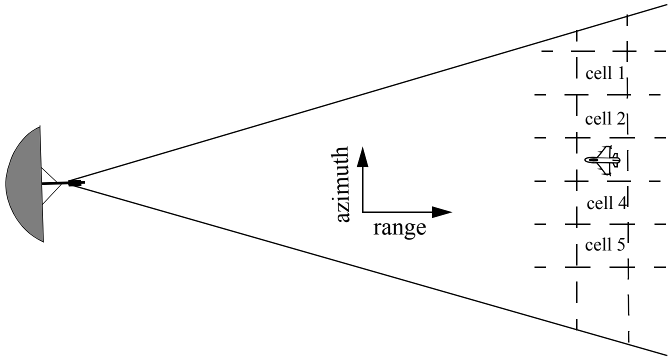


Figure 1.24. Illustration of collapsing loss. Noise sources in cells 1, 2, 4, and 5 converge to increase the noise level in cell 3.

1.8.5. Processing Losses

a. Detector Approximation:

The output voltage signal of a radar receiver that utilizes a linear detector is

$$v(t) = \sqrt{v_I^2(t) + v_Q^2(t)} \tag{1.91}$$

where (v_I, v_Q) are the in-phase and quadrature components. For a radar using a square law detector, we have $v^2(t) = v_I^2(t) + v_Q^2(t)$.

Since in real hardware the operations of squares and square roots are time consuming, many algorithms have been developed for detector approximation. This approximation results in a loss of the signal power, typically 0.5 to 1 dB.

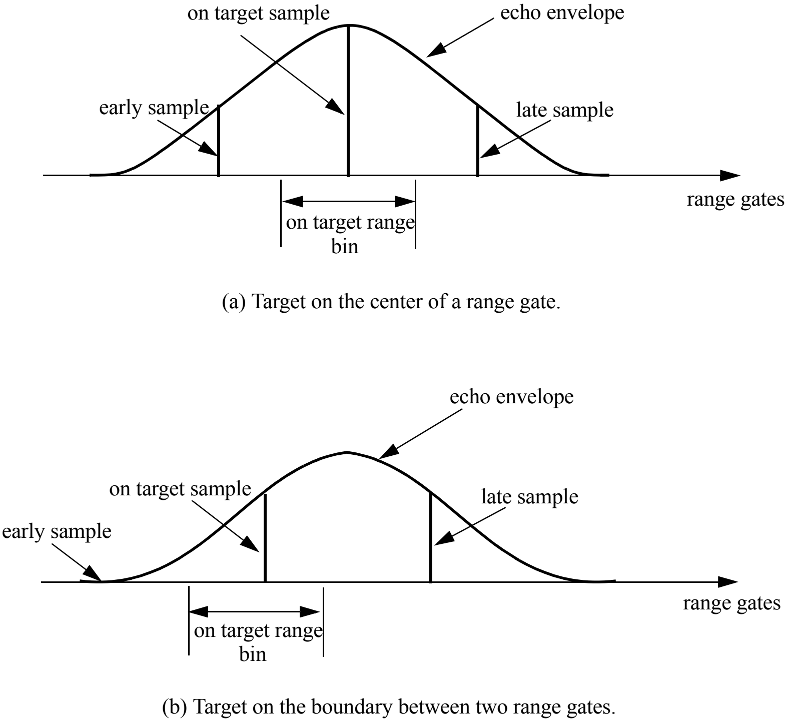
b. Constant False Alarm Rate (CFAR) Loss:

In many cases the radar detection threshold is constantly adjusted as a function of the receiver noise level in order to maintain a constant false alarm rate. For this purpose, Constant False Alarm Rate (CFAR) processors are utilized in order to keep the number of false alarms under control in a changing and unknown background of interference. CFAR processing can cause a loss in the SNR level on the order of 1 dB.

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.

c. Quantization Loss:

Finite word length (number of bits) and quantization noise cause an increase in the noise power density at the output of the Analog to Digital (A/D) converter. The A/D noise level is $q^2/12$, where q is the quantization level.



(a) Target on the center of a range gate.

(b) Target on the boundary between two range gates.

Figure 1.25. Illustration of range gate straddling.

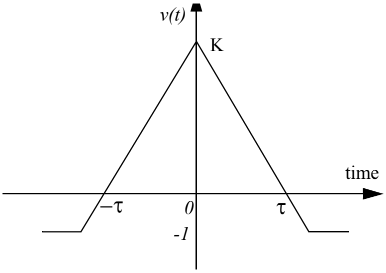
d. Range Gate Straddle:

The radar receiver is normally mechanized as a series of contiguous range gates (bins). Each range bin is implemented as an integrator matched to the transmitted pulsewidth. Since the radar receiver acts as a filter that smears (smooths), the received target echoes. The smoothed target return envelope is normally straddled to cover more than one range gate.

Typically, three gates are affected; they are called the early, on, and late gates. If a point target is located exactly at the center of a range gate, then the early and late samples are equal. However, as the target starts to move into the next gate, the late sample becomes larger while the early sample gets smaller. In any case, the amplitudes of all three samples should always roughly add up to the same value. Fig. 1.25 illustrates the concept of range straddling. The envelope of the smoothed target echo is likely to be Gaussian shaped. In practice, triangular shaped envelopes may be easier and faster to implement. Since the target is likely to fall anywhere between two adjacent range bins, a loss in the SNR occurs (per range gate). More specifically, a target's returned energy is split between three range bins. Typically, straddle loss of about 2 to 3 dB is not unusual.

Example:

Consider the smoothed target echo voltage shown below. Assume 1Ω *resistance. Find the power loss due to range gate straddling over the interval* $\{0, \tau\}$.



Solution:

The smoothed voltage can be written as

$$v(t) = \left\{ \begin{array}{ll} K + \left(\dfrac{K+1}{\tau}\right)t & ; \ t < 0 \\ K - \left(\dfrac{K+1}{\tau}\right)t & ; \ t \geq 0 \end{array} \right\}$$

The power loss due to straddle over the interval $\{0, \tau\}$ *is*

$$L_s = \frac{v^2}{K^2} = 1 - 2\left(\frac{K+1}{K\tau}\right)t + \left(\frac{K+1}{K\tau}\right)^2 t^2$$

The average power loss is then

$$\bar{L}_s = \frac{2}{\tau}\int_0^{\tau/2} \left(1 - 2\left(\frac{K+1}{K\tau}\right)t + \left(\frac{K+1}{K\tau}\right)^2 t^2\right) \quad dt$$

$$= 1 - \frac{K+1}{2K} + \frac{(K+1)^2}{12K^2}$$

and, for example, if $K = 15$, then $\bar{L}_s = 2.5dB$.

e. Doppler Filter Straddle:

Doppler filter straddle is similar to range gate straddle. However, in this case the Doppler filter spectrum is spread (widened) due to weighting functions. Weighting functions are normally used to reduce the sidelobe levels. Since the target Doppler frequency can fall anywhere between two Doppler filters, signal loss occurs.

1.8.6. Other Losses

Other losses may include equipment losses due to aging radar hardware, matched filter loss, and antenna efficiency loss. Tracking radars suffer from crossover (squint) loss.

1.9. “MyRadar” Design Case Study - Visit 1

In this section, a design case study, referred to as *“MyRadar”* design case study, is introduced. For this purpose, only the theory introduced in this chapter is used to fulfill the design requirements. Note that since only a limited amount of information has been introduced in this chapter, the design process may seem illogical to some readers. However, as new material is introduced in subsequent chapters, the design requirements are updated and/or new design requirements are introduced based on the particular material of that chapter. Consequently, the design process will also be updated to accommodate the new theory and techniques learned in that chapter.

1.9.1. Authors and Publisher Disclaimer

The design case study *“MyRadar”* is a ground based air defense radar derived and based on Brookner’s[1] open literature source. However, the design approach introduced in this book is based on the authors’ point of view of how to design such radar. Thus, the design process takes on a different flavor than that introduced by Brookner. Additionally, any and all design alternatives presented in this book are based on and can be easily traced to open literature sources.

Furthermore, the design approach adopted in this book is based on modeling many of the radar system components with no regards to any hardware constraints nor to any practical limitations. The design presented in this book is intended to be tutorial and academic in nature and does not adhere to any other requirements. Finally, the MATLAB code presented in this book is intended to be illustrative and academic and is not designed nor intended for any other uses.

Using the material presented in this book and the MATLAB code designed by the authors of this book by any entity or person is strictly at will. The authors and the publisher are neither liable nor responsible for any material or non-material losses, loss of wages, personal or property damages of any kind, or for any other type of damages of any and all types that may be incurred by using this book.

1.9.2. Problem Statement

You are to design a ground based radar to fulfill the following mission: Search and Detection. The threat consists of aircraft with an average RCS of 6 dBsm ($\sigma_a = 4m^2$), and missiles with an average RCS of -3 dBsm ($\sigma_m = 0.5m^2$). The missile altitude is 2Km, and the aircraft altitude is about 7 Km. Assume a scanning radar with 360 degrees azimuth coverage. The scan rate is less than or equal to 1 revolution every 2 seconds. Assume L to X band. We need range resolution of 150 m. No angular resolution is specified at this time. Also assume that only one missile and one aircraft constitute the whole threat. Assume a noise figure F = 6 dB, and total receiver loss L = 8 dB. For now use a fan beam with azimuth beamwidth of less than 3 degrees. Assume that 13 dB SNR is a reasonable detection threshold. Finally, assume flat earth.

1.9.3. A Design

The desired range resolution is $\Delta R = 150m$. Thus, using Eq. (1.8) one calculates the required pulsewidth as $\tau = 1\mu\sec$, or equivalently the required bandwidth $B = 1MHz$. At this point a few preliminary decisions must be made. This includes the selection of the radar operating frequency, the aperture size, and the single pulse peak power.

[1] Brookner, Eli, Editor, *Practical Phased Array Antenna Systems*, Artech House, 1991, Chapter 7.

The choice of an operating frequency that can fulfill the design requirements is driven by many factors, such as aperture size, antenna gain, clutter, atmospheric attenuation, and the maximum peak power, to name a few. In this design, an operating frequency $f = 3GHz$ is selected. This choice is somewhat arbitrary at this point; however, as we proceed with the design process this choice will be better clarified.

Second, the transportability (mobility) of the radar drives the designer in the direction of a smaller aperture type. A good choice would be less than 5 meters squared. For now choose $A_e = 2.25m^2$. The last issue that one must consider is the energy required per pulse. Note that this design approach assumes that the minimum detection SNR (13 dB) requirement is based on pulse integration. This condition is true because the target is illuminated with several pulses during a single scan, provided that the antenna azimuth beamwidth and the PRF choice satisfy Eq. (1.81).

The single pulse energy is $E = P_t\tau$. Typically, a given radar must be designed such that it has a handful of pulsewidths (waveforms) to choose from. Different waveforms (pulsewidths) are used for definite modes of operations (search, track, etc.). However, for now only a single pulse which satisfies the range resolution requirement is considered. To calculate the minimum single pulse energy required for proper detection, use Eq. (1.57). More precisely,

$$E = P_t\tau = \frac{(4\pi)^3 kT_e FLR^4 SNR_1}{G^2\lambda^2\sigma} \tag{1.92}$$

All parameters in Eq. (1.92) are known, except for the antenna gain, the detection range, and the single pulse SNR. The antenna gain is calculated from

$$\left(G = \frac{4\pi A_e}{\lambda^2} = \frac{4\pi \times 2.25}{(0.1)^2} = 2827.4\right) \Rightarrow G = 34.5dB \tag{1.93}$$

where the relation ($\lambda = c/f$) was used.

In order to estimate the detection range, consider the following argument. Since an aircraft has a larger RCS than a missile, one would expect an aircraft to be detected at a much longer range than that of a missile. This is depicted in Fig. 1.26, where R_a refers to the aircraft detection range and R_m denotes the missile detection range. As illustrated in this figure, the minimum search elevation angle θ_1 is driven by the missile detection range, assuming that the missiles are detected, with the proper SNR, as soon as they enter the radar beam. Alternatively, the maximum search elevation angle θ_2 is driven the aircraft's position along with the range that corresponds to the defense's last chance to intercept the threat (both aircraft and missile). This range is often called "keep-out minimum range" and is denoted by R_{min}. In this design approach,

$R_{min} = 30Km$ is selected. In practice, the keep-out minimum range is normally specified by the user as a design requirement.

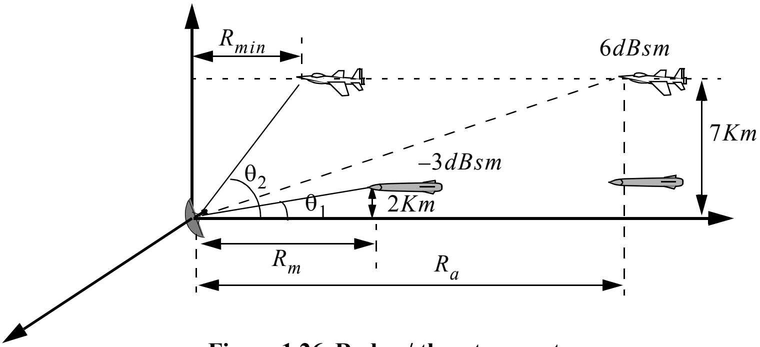


Figure 1.26. Radar / threat geometry.

The determination of R_a and R_m is dictated by how fast can a defense interceptor reach the keep-out minimum range and kill the threat. For example, assume that the threatening aircraft velocity is $400m/s$ and the threatening missile velocity is $150m/s$. Alternatively, assume that an interceptor average velocity is $250m/s$. It follows that, the interceptor time of flight, based on $R_{min} = 30Km$, is

$$T_{interceptor} = \frac{30 \times 10^3}{250} = 120\sec \tag{1.94}$$

Therefore, an aircraft and a missile must be detected by the radar at

$$\begin{aligned} R_a &= 30Km + 120 \times 400 = 78Km \\ R_m &= 30Km + 120 \times 150 = 48Km \end{aligned} \tag{1.95}$$

Note that these values should be used only as a guide. The actual detection range must also include a few more kilometers, in order to allow the defense better reaction time. In this design, choose $R_m = 55Km; and\ R_a = 90Km$. Therefore, the maximum PRF that guarantees an unambiguous range of at least 90Km is calculated from Eq. (1.5). More precisely,

$$f_r \le \frac{c}{2R_u} = \frac{3 \times 10^8}{2 \times 90 \times 10^3} = 1.67KHz \tag{1.96}$$

Since there are no angular resolution requirements imposed on the design at this point, then Eq. (1.96) is the only criterion that will be used to determine the radar operating PRF. Select,

$$f_r = 1000Hz \tag{1.97}$$

The minimum and maximum elevation angles are, respectively, calculated as

$$\theta_1 = \operatorname{atan}\left(\frac{2}{55}\right) = 2.08° \tag{1.98}$$

$$\theta_2 = \operatorname{atan}\left(\frac{7}{30}\right) = 13.13° \tag{1.99}$$

These angles are then used to compute the elevation search extent (remember that the azimuth search extent is equal to $360°$). More precisely, the search volume Ω (in steradians) is given by

$$\Omega = \frac{\theta_2 - \theta_1}{(57.296)^2} \times 360 \tag{1.100}$$

Consequently, the search volume is

$$\Omega = 360 \times \frac{\theta_2 - \theta_1}{(57.296)^2} = 360 \times \frac{13.13 - 2.08}{(57.296)^2} = 1.212 \ \ steradians \tag{1.101}$$

The desired antenna must have a fan beam; thus using a parabolic rectangular antenna will meet the design requirements. Select $A_e = 2.25m^2$; the corresponding antenna 3-dB elevation and azimuth beamwidths are denoted as θ_e, θ_a, respectively. Select

$$\theta_e = \theta_2 - \theta_1 = 13.13 - 2.08 = 11.05° \tag{1.102}$$

The azimuth 3-dB antenna beamwidth is calculated using Eq. (1.42) as

$$\theta_a = \frac{4\pi}{G\theta_e} = \frac{4 \times \pi \times 180^2}{2827.4 \times \pi^2 \times 11} = 1.33° \tag{1.103}$$

It follows that the number of pulses that strikes a target during a single scan is calculated using Eq. (1.81) as

$$n_p \le \frac{\theta_a f_r}{\dot{\theta}_{scan}} = \frac{1.33 \times 1000}{180} = 7.39 \Rightarrow n_p = 7 \tag{1.104}$$

The design approach presented in this book will only assume non-coherent integration (the reader is advised to re-calculate all results by assuming coherent integration, instead). The design requirement mandates a 13 dB SNR for detection. By using Eq. (1.87) one calculates the required single pulse SNR,

$$(SNR)_1 = \frac{10^{1.3}}{2 \times 7} + \sqrt{\frac{(10^{1.3})^2}{4 \times 7^2} + \frac{10^{1.3}}{7}} = 3.635 \Rightarrow (SNR)_1 = 5.6dB \tag{1.105}$$

Furthermore the non-coherent integration loss associated with this case is computed from Eq. (1.85),

$$L_{NCI} = \frac{1 + 3.635}{3.635} = 1.27 \Rightarrow L_{NCI} = 1.056dB \tag{1.106}$$

It follows that the corresponding **single pulse** energy for the missile and the aircraft cases are respectively given by

$$E_m = \frac{(4\pi)^3 kT_e FLR_m^4 (SNR)_1}{G^2 \lambda^2 \sigma_m} \Rightarrow$$

$$E_m = \frac{(4\pi)^3 (1.38 \times 10^{-23})(290)(10^{0.8})(10^{0.6})(55 \times 10^3)^4 10^{0.56}}{(2827.4)^2 (0.1)^2 (0.5)} = 0.1658 \ \ Joules \tag{1.107}$$

$$E_a = \frac{(4\pi)^3 kT_e FLR_a^4 (SNR)_1}{G^2 \lambda^2 \sigma_a} \Rightarrow$$

$$E_a = \frac{(4\pi)^3 (1.38 \times 10^{-23})(290)(10^{0.8})(10^{0.6})(90 \times 10^3)^4 10^{0.56}}{(2827.4)^2 (0.1)^2 (4)} = 0.1487 \ \ Joules \tag{1.108}$$

Hence, the peak power that satisfies the single pulse detection requirement for both target types is

$$P_t = \frac{E}{\tau} = \frac{0.1658}{1 \times 10^{-6}} = 165.8KW \tag{1.109}$$

The radar equation with pulse integration is

$$SNR = \frac{P_t^1 G^2 \lambda^2 \sigma}{(4\pi)^3 kT_e BFLR^4} \frac{n_p}{L_{NCI}} \tag{1.110}$$

Figure 1.27 shows the SNR versus detection range for both target-types with and without integration. To reproduce this figure use MATLAB program *"fig1_27.m"* which is given in Listing 1.12 in Section 1.10.

1.9.4. A Design Alternative

One could have elected not to reduce the single pulse peak power, but rather keep the single pulse peak power as computed in Eq. (1.109) and increase the radar detection range. For example, integrating 7 pulses coherently would improve the radar detection range by a factor of

$$R_{imp} = (7)^{0.25} = 1.63 \tag{1.111}$$

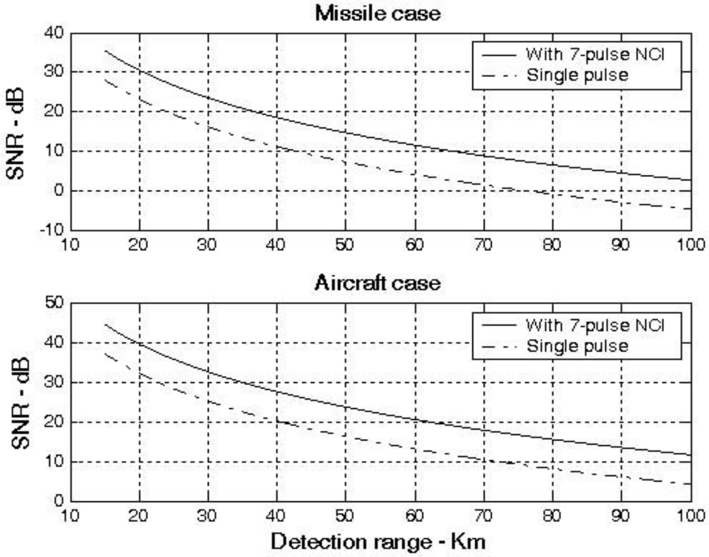


Figure 1.27. SNR versus detection range for both target types with and without pulse integration.

It follows that the new missile and aircraft detection ranges are

$$R_a = 78 \times 1.63 = 126.9Km$$
$$R_m = 48 \times 1.63 = 78.08Km \tag{1.112}$$

Note that extending the minimum detection range for a missile to $R_m = 78Km$ would increase the size of the extent of the elevation search volume. More precisely,

$$\theta_1 = \operatorname{atan}\left(\frac{2}{78}\right) = 1.47° \tag{1.113}$$

It follows that the search volume Ω (in steradians) is now

$$\Omega = 360 \times \frac{\theta_2 - \theta_1}{(57.296)^2} = 360 \times \frac{13.13 - 1.47}{(57.296)^2} = 1.279 \ \ steradians \tag{1.114}$$

Alternatively, integrating 7 pulses non-coherently with $(SNR)_{NCI} = 13dB$ yields

$$(SNR)_1 = 5.6dB \tag{1.115}$$

and the integration loss is

$$L_{NCI} = 1.057dB \tag{1.116}$$

Then, the net non-coherent integration gain is

$$NCI_{gain} = 10 \times \log(7) - 1.057 = 7.394dB \Rightarrow NCI_{gain} = 5.488 \tag{1.117}$$

Thus, the radar detection range is now improved due to a 7-pulse non-coherent integration to

$$\begin{aligned} R_a &= 78 \times (5.488)^{0.25} = 119.38Km \\ R_m &= 48 \times (5.488)^{0.25} = 73.467Km \end{aligned} \tag{1.118}$$

Again, the extent of the elevation search volume is changed to

$$\theta_1 = \operatorname{atan}\left(\frac{2}{73.467}\right) = 1.56° \tag{1.119}$$

It follows that the search volume Ω (in steradians) is now

$$\Omega = 360 \times \frac{\theta_2 - \theta_1}{(57.296)^2} = 360 \times \frac{13.13 - 1.56}{(57.296)^2} = 1.269 \ \ steradians \tag{1.120}$$

1.10. MATLAB Program and Function Listings

This section presents listings for all MATLAB functions and programs used in this chapter. Users are encouraged to vary the input parameters and rerun these programs in order to enhance their understanding of the theory presented in the text. All selected parameters and variables follow the same nomenclature used in the text; thus, understanding the structure and hierarchy of the presented code should be an easy task once the user has read the chapter.

Note that all MATLAB programs and functions developed in this book can be downloaded from CRC Press Web Site “www.crcpress.com”. Additionally, all MATLAB code developed for this book was developed using MATLAB 6.5 Release 13 for Microsoft Windows.

Listing 1.1. MATLAB Function “radar_eq.m”

```
function [snr] = radar_eq(pt, freq, g, sigma, te, b, nf, loss, range)
% This program implements Eq. (1.56)
c = 3.0e+8; % speed of light
lambda = c / freq; % wavelength
p_peak = 10*log10(pt); % convert peak power to 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
te_db = 10*log10(te); % noise temp. in dB
b_db = 10*log10(b); % bandwidth in dB
range_pwr4_db = 10*log10(range.^4); % vector of target range^4 in dB
% Implement Equation (1.56)
num = p_peak + g + g + lambda_sqdb + sigmadb;
den = four_pi_cub + k_db + te_db + b_db + nf + loss + range_pwr4_db;
snr = num - den;
return
```

Listing 1.2. MATLAB Program “fig1_12.m”

```
% Use this program to reproduce Fig. 1.12 of text.
close all
clear all
pt = 1.5e+6; % peak power in Watts
freq = 5.6e+9; % radar operating frequency in Hz
g = 45.0; % antenna gain in dB
sigma = 0.1; % radar cross section in m squared
te = 290.0; % effective noise temperature in Kelvins
b = 5.0e+6; % radar operating bandwidth in Hz
nf = 3.0; %noise figure in dB
loss = 6.0; % radar losses in dB
range = linspace(25e3,165e3,1000); % traget range 25 -165 Km, 1000 points
snr1 = radar_eq(pt, freq, g, sigma, te, b, nf, loss, range);
snr2 = radar_eq(pt, freq, g, sigma/10, te, b, nf, loss, range);
snr3 = radar_eq(pt, freq, g, sigma*10, te, b, nf, loss, range);
% plot SNR versus range
figure(1)
rangekm  = range ./ 1000;
plot(rangekm,snr3,'k',rangekm,snr1,'k -.',rangekm,snr2,'k:')
grid
legend('\sigma = 0 dBsm','\sigma = -10dBsm','\sigma = -20 dBsm')
xlabel ('Detection range - Km');
ylabel ('SNR - dB');
snr1 = radar_eq(pt, freq, g, sigma, te, b, nf, loss, range);
snr2 = radar_eq(pt*.4, freq, g, sigma, te, b, nf, loss, range);
snr3 = radar_eq(pt*1.8, freq, g, sigma, te, b, nf, loss, range);
figure (2)
plot(rangekm,snr3,'k',rangekm,snr1,'k -.',rangekm,snr2,'k:')
grid
legend('Pt = 2.16 MW','Pt = 1.5 MW','Pt = 0.6 MW')
xlabel ('Detection range - Km');
ylabel ('SNR - dB');
```

Listing 1.3. MATLAB Program “fig1_13.m”

```
% Use this program to reproduce Fig. 1.13 of text.
close all
clear all
pt = 1.e+6; % peak power in Watts
freq = 5.6e+9; % radar operating frequency in Hz
g = 40.0; % antenna gain in dB
sigma = 0.1; % radar cross section in m squared
te =300.0; % effective noise temperature in Kelvins
nf = 5.0; %noise figure in dB
loss = 6.0; % radar losses in dB
range = [75e3,100e3,150e3]; % three range values
snr_db = linspace(5,20,200); % SNR values from 5 dB to 20 dB 200 points
snr = 10.^(0.1.*snr_db); % convert snr into base 10
gain = 10^(0.1*g); %convert antenna gain into base 10
loss = 10^(0.1*loss); % convert losses into base 10
F = 10^(0.1*nf); % convert noise figure into base 10
lambda = 3.e8 / freq; % compute wavelength
% Implement Eq.(1.57)
den = pt * gain * gain * sigma * lambda^2;
num1 = (4*pi)^3 * 1.38e-23 * te * F * loss * range(1)^4 .* snr;
num2 = (4*pi)^3 * 1.38e-23 * te * F * loss * range(2)^4 .* snr;
num3 = (4*pi)^3 * 1.38e-23 * te * F * loss * range(3)^4 .* snr;
tau1 = num1 ./ den ;
tau2 = num2 ./ den;
tau3 = num3 ./ den;
% plot tau versus snr
figure(1)
semilogy(snr_db,1e6*tau1,'k',snr_db,1e6*tau2,'k -.',snr_db,1e6*tau3,'k:')
grid
legend('R = 75 Km','R = 100 Km','R = 150 Km')
xlabel ('Minimum required SNR - dB');
ylabel ('\tau (pulsewidth) in \mu sec');
```

Listing 1.4. MATLAB Program “ref_snr.m”

```
% This program implements Eq. (1.60)
clear all
close all
Rref = 86e3; % ref. range
tau_ref = .1e-6; % ref. pulsewidth
SNRref = 20.; % Ref SNR in dB
snrref = 10^(SNRref/10);
```

```
Sigmaref = 0.1; % ref RCS in m^2
Lossp = 2; % processing loss in dB
lossp = 10^(Lossp/10);
% Enter desired value
tau = tau_ref;
R = 120e3;
rangeratio = (Rref / R)^4;
Sigma = 0.2;
% Implement Eq. (1.60)
snr = snrref * (tau / tau_ref) * (1. / lossp) * ...
   (Sigma / Sigmaref) * rangeratio;
snr = 10*log10(snr)
```

Listing 1.5. MATLAB Function "power_aperture.m"

```
function PAP =
                        power_aperture(snr,tsc,sigma,range,te,nf,loss,az_angle,el_
                        angle)
% This program implements Eq. (1.67)
Tsc = 10*log10(tsc); % convert Tsc into dB
Sigma = 10*log10(sigma); % convert sigma to dB
four_pi = 10*log10(4.0 * pi); % (4pi) in dB
k_db = 10*log10(1.38e-23); % Boltzman's constant in dB
Te = 10*log10(te); % noise temp. in dB
range_pwr4_db = 10*log10(range.^4); % target range^4 in dB
omega = az_angle * el_angle / (57.296)^2; % compute search volume in stera-
                        dians
Omega = 10*log10(omega) % search volume in dB
% implement Eq. (1.67)
PAP = snr + four_pi + k_db + Te + nf + loss + range_pwr4_db + Omega ...
   - Sigma - Tsc;
return
```

Listing 1.6. MATLAB Program "fig1_16.m"

```
% Use this program to reproduce Fig. 1.16 of text.
close all
clear all
tsc = 2.5; % Scan time is 2.5 seconds
sigma = 0.1; % radar cross section in m squared
te = 900.0; % effective noise temperature in Kelvins
snr = 15; % desired SNR in dB
nf = 6.0; %noise figure in dB
loss = 7.0; % radar losses in dB
```

```
az_angle = 2; % search volume azimuth extent in degrees
el_angle = 2; % search volume elevation extent in degrees
range = linspace(20e3,250e3,1000); % range to target 20 Km 250 Km, 1000
                          points
pap1 = power_aperture(snr,tsc,sigma/10,range,te,nf,loss,az_angle,el_angle);
pap2 = power_aperture(snr,tsc,sigma,range,te,nf,loss,az_angle,el_angle);
pap3 = power_aperture(snr,tsc,sigma*10,range,te,nf,loss,az_angle,el_angle);
% plot power aperture product versus range
% figure 1.16a
figure(1)
rangekm  = range ./ 1000;
plot(rangekm,pap1,'k',rangekm,pap2,'k -.',rangekm,pap3,'k:')
grid
legend('\sigma = -20 dBsm','\sigma = -10dBsm','\sigma = 0 dBsm')
xlabel ('Detection range in Km');
ylabel ('Power aperture product in dB');
% generate Figure 1.16b
lambda = 0.03; % wavelength in meters
G = 45; % antenna gain in dB
ae = linspace(1,25,1000);% aperture size 1 to 25 meter squared, 1000 points
Ae = 10*log10(ae);
range = 250e3; % range of interest is 250 Km
pap1 = power_aperture(snr,tsc,sigma/10,range,te,nf,loss,az_angle,el_angle);
pap2 = power_aperture(snr,tsc,sigma,range,te,nf,loss,az_angle,el_angle);
pap3 = power_aperture(snr,tsc,sigma*10,range,te,nf,loss,az_angle,el_angle);
Pav1 = pap1 - Ae;
Pav2 = pap2 - Ae;
Pav3 = pap3 - Ae;
figure(2)
plot(ae,Pav1,'k',ae,Pav2,'k -.',ae,Pav3,'k:')
grid
xlabel('Aperture size in square meters')
ylabel('Pav in dB')
legend('\sigma = -20 dBsm','\sigma = -10dBsm','\sigma = 0 dBsm')
```

Listing 1.7. MATLAB Program “casestudy1_1.m”

```
% This program is used to generate Fig. 1.17
% It implements the search radar equation defined in Eq. 1.67
clear all
close all
snr = 15.0;        % Sensitivity SNR in dB
tsc = 2.;          % Antenna scan time in seconds
sigma_tgtm = -10;    % Missile RCS in dBsm
```

```
sigma_tgta = 6;      % Aircraft RCS in dBsm
range = 60.0; % Sensitivity range in Km,
te = 290.0;         % Effective noise temperature in Kelvins
nf = 8;             % Noise figure in dB
loss = 10.0;        % Radar losses in dB
az_angle = 360.0;   % Search volume azimuth extent in degrees
el_angle = 10.0;  % Search volume elevation extent in degrees
c = 3.0e+8;         % Speed of light
% Compute Omega in steradians
omega = (az_angle / 57.296) * (el_angle /57.296);
omega_db = 10.0*log10(omega); % Convert Omega to dBs
k_db = 10.*log10(1.38e-23);
te_db = 10*log10(te);
tsc_db = 10*log10(tsc);
factor = 10*log10(4*pi);
rangemdb = 10*log10(range * 1000.);
rangeadb = 10*log10(range * 1000.);
PAP_Missile = snr - sigma_tgtm - tsc_db + factor + 4.0 * rangemdb + ...
   k_db + te_db + nf + loss + omega_db
PAP_Aircraft = snr - sigma_tgta - tsc_db + factor + 4.0 * rangeadb + ...
   k_db + te_db + nf + loss + omega_db
index = 0;
% vary range from 2Km to 90 Km
for rangevar = 2 : 1 : 90
   index = index + 1;
   rangedb = 10*log10(rangevar * 1000.0);
   papm(index) = snr - sigma_tgtm - tsc_db + factor + 4.0 * rangedb + ...
      k_db + te_db + nf + loss + omega_db;
  missile_PAP(index) = PAP_Missile;
  aircraft_PAP(index) = PAP_Aircraft;
   papa(index) = snr - sigma_tgta - tsc_db + factor + 4.0 * rangedb + ...
      k_db + te_db + nf + loss +omega_db;
end
var = 2 : 1 : 90;
figure (1)
plot (var,papm,'k',var,papa,'k-.')
legend ('Missile','Aircraft')
xlabel ('Range - Km');
ylabel ('Power Aperture Product - dB');
hold on
plot(var,missile_PAP,'k:',var,aircraft_PAP,'k:')
grid
hold off
```

Listing 1.8. MATLAB Program “fig1_19.m”

```
% Use this program to reproduce Fig. 1.19 and Fig. 1.20 of text.
close all
clear all
pt = 4; % peak power in Watts
freq = 94e+9; % radar operating frequency in Hz
g = 47.0; % antenna gain in dB
sigma = 20; % radar cross section in m squared
te = 293.0; % effective noise temperature in Kelvins
b = 20e+6; % radar operating bandwidth in Hz
nf = 7.0; %noise figure in dB
loss = 10.0; % radar losses in dB
range = linspace(1.e3,12e3,1000); % range to target from 1. Km 12 Km, 1000
                      points
snr1 = radar_eq(pt, freq, g, sigma, te, b, nf, loss, range);
Rnewci = (94^0.25) .* range;
snrCI = snr1 + 10*log10(94); % 94 pulse coherent integration
% plot SNR versus range
figure(1)
rangekm  = range ./ 1000;
plot(rangekm,snr1,'k',Rnewci./1000,snr1,'k -.')
axis([1 12 -20 45])
grid
legend('single pulse','94 pulse CI')
xlabel ('Detection range - Km');
ylabel ('SNR - dB');
% Generate Figure 1.20
snr_b10 = 10.^(snr1./10);
SNR_1 = snr_b10./(2*94) + sqrt(((snr_b10.^2) ./ (4*94*94)) + (snr_b10 ./
                      94)); % Equation 1.80 of text
LNCI = (1+SNR_1) ./ SNR_1; % Equation 1.78 of text
NCIgain = 10*log10(94) - 10*log10(LNCI);
Rnewnci = ((10.^(0.1*NCIgain)).^0.25) .* range;
snrnci = snr1 + NCIgain;
figure (2)
plot(rangekm,snr1,'k',Rnewnci./1000,snr1,'k -.', Rnewci./1000,snr1,'k:')
axis([1 12 -20 45])
grid
legend('single pulse','94 pulse NCI','94 pulse CI')
xlabel ('Detection range - Km');
ylabel ('SNR - dB');
```

Listing 1.9. MATLAB Program “fig1_21.m”

```
%use this figure to generate Fig. 1.21 of text
clear all
close all
np = linspace(1,10000,1000);
snrci = pulse_integration(4,94.e9,47,20,290,20e6,7,10,5.01e3,np,1);
snrnci = pulse_integration(4,94.e9,47,20,290,20e6,7,10,5.01e3,np,2);
semilogx(np,snrci,'k',np,snrnci,'k:')
legend('Coherent integration','Non-coherent integration')
grid
xlabel ('Number of integrated pulses');
ylabel ('SNR - dB');
```

Listing 1.10. MATLAB Function “pulse_integration.m”

```
function [snrout] = pulse_integration(pt, freq, g, sigma, te, b, nf, loss,
                        range,np,ci_nci)
 snr1 = radar_eq(pt, freq, g, sigma, te, b, nf, loss, range) % single pulse SNR
if (ci_nci == 1) % coherent integration
  snrout = snr1 + 10*log10(np);
else % non-coherent integration
   if (ci_nci == 2)
      snr_nci = 10.^(snr1./10);
      val1 = (snr_nci.^2) ./ (4.*np.*np);
      val2 = snr_nci ./ np;
      val3 = snr_nci ./ (2.*np);
      SNR_1 = val3 + sqrt(val1 + val2); % Equation 1.87 of text
      LNCI = (1+SNR_1) ./ SNR_1; % Equation 1.85 of text
      snrout = snr1 + 10*log10(np) - 10*log10(LNCI);
   end
end
return
```

Listing 1.11. MATLAB Program “myradarvisit1_1.m”

```
close all
clear all
pt = 724.2e+3; % peak power in Watts
freq = 3e+9; % radar operating frequency in Hz
g = 37.0; % antenna gain in dB
sigmam = 0.5; % missile RCS in m squared
sigmaa = 4.0; % aircraft RCS in m squared
te = 290.0; % effective noise temperature in Kelvins
b = 1.0e+6; % radar operating bandwidth in Hz
```

```
nf = 6.0; %noise figure in dB
loss = 8.0; % radar losses in dB
range = linspace(5e3,125e3,1000); % range to target from 25 Km 165 Km,
                          1000 points
snr1 = radar_eq(pt, freq, g, sigmam, te, b, nf, loss, range);
snr2 = radar_eq(pt, freq, g, sigmaa, te, b, nf, loss, range);
% plot SNR versus range
figure(1)
rangekm  = range ./ 1000;
plot(rangekm,snr1,'k',rangekm,snr2,'k:')
grid
legend('Misssile','Aircraft')
xlabel ('Detection range - Km');
ylabel ('SNR - dB');
```

Listing 1.12. MATLAB Program “fig1_27.m”

```
% Use this program to reproduce Fig. 1.27 of text.
close all
clear all
np = 7;
pt = 165.8e3; % peak power in Watts
freq = 3e+9; % radar operating frequency in Hz
g = 34.5139; % antenna gain in dB
sigmam = 0.5; % missile RCS m squared
sigmaa = 4; % aircraft RCS m squared
te = 290.0; % effective noise temperature in Kelvins
b = 1.0e+6; % radar operating bandwidth in Hz
nf = 6.0; %noise figure in dB
loss = 8.0; % radar losses in dB
% compute the single pulse SNR when 7-pulse NCI is used
SNR_1 = (10^1.3)/(2*7) + sqrt((((10^1.3)^2) / (4*7*7)) + ((10^1.3) / 7));
% compute the integration loss
LNCI = 10*log10((1+SNR_1)/SNR_1);
loss_total = loss + LNCI;
range = linspace(15e3,100e3,1000); % range to target from 15 to 100 Km,
                          1000 points
% modify pt by np*pt to account for pulse integration
snrmnci = radar_eq(np*pt, freq, g, sigmam, te, b, nf, loss_total, range);
snrm = radar_eq(pt, freq, g, sigmam, te, b, nf, loss, range);
snranci = radar_eq(np*pt, freq, g, sigmaa, te, b, nf, loss_total, range);
snra = radar_eq(pt, freq, g, sigmaa, te, b, nf, loss, range);
% plot SNR versus range
rangekm  = range ./ 1000;
```

```
figure(1)
subplot(2,1,1)
plot(rangekm,snrmnci,'k',rangekm,snrm,'k -.')
grid
legend('With 7-pulse NCI','Single pulse')
ylabel ('SNR - dB');
title('Missile case')
subplot(2,1,2)
plot(rangekm,snranci,'k',rangekm,snra,'k -.')
grid
legend('With 7-pulse NCI','Single pulse')
ylabel ('SNR - dB');
title('Aircraft case')
xlabel('Detection range - Km')
```

Appendix 1A Pulsed Radar

1A.1. Introduction

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) pulsewidth, 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 in Table 1A.1.

TABLE 1A.1. PRF ambiguities.

PRF	Range Ambiguous	Doppler Ambiguous
Low PRF	No	Yes
Medium PRF	Yes	Yes
High PRF	Yes	No

Radars can utilize constant and varying (agile) PRFs. For example, Moving Target Indicator (MTI) radars use PRF agility to avoid blind speeds. 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 latter forms of PRF agility are sometimes referred to as PRF jitter.

Fig. 1A.1 shows a simplified pulsed radar block diagram. The range gates can be implemented as filters that open and close at time intervals that correspond 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 matched to the radar pulsewidth. The NBF bank is normally implemented using an FFT, where bandwidth of the individual filters corresponds to the FFT frequency resolution.

1A.2. Range and Doppler Ambiguities

As explained earlier, a pulsed radar can be range ambiguous if a second pulse is transmitted prior to the return of the first pulse. In general, the radar PRF is chosen such that the unambiguous range is large enough to meet the radar's operational requirements. Therefore, long-range search (surveillance) radars would require relatively low PRFs.

The line spectrum of a train of pulses has $\sin x / x$ envelope, and the line spectra are separated by the PRF, f_r, as illustrated in Fig. 1A.2. 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

$$f_r = 2f_{dmax} = \frac{2v_{rmax}}{\lambda} \tag{1A.1}$$

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.

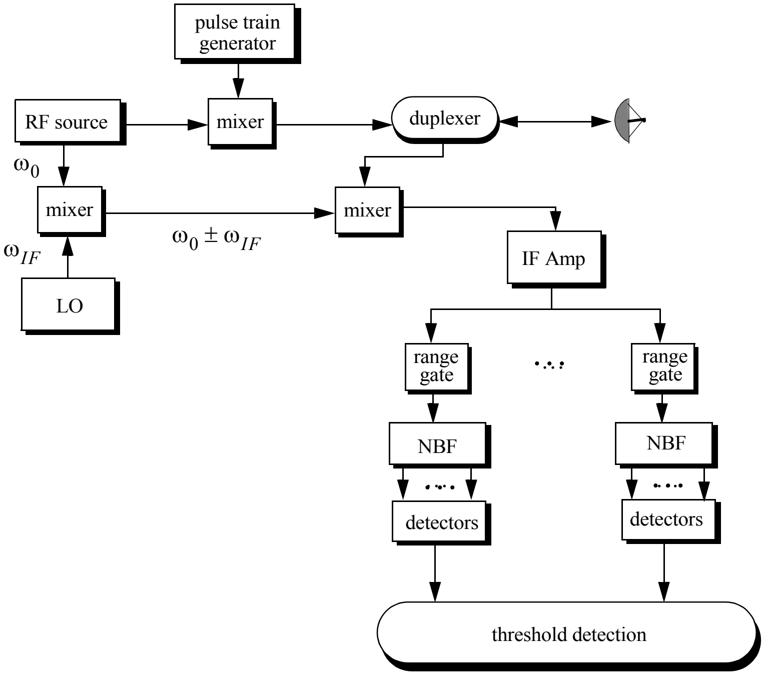


Figure 1A.1. Pulsed radar block diagram.

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.

1A.3. Resolving Range Ambiguity

Consider a radar that uses two PRFs, f_{r1} and f_{r2}, on transmit to resolve range ambiguity, as shown in Fig. 1A.3. 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}.

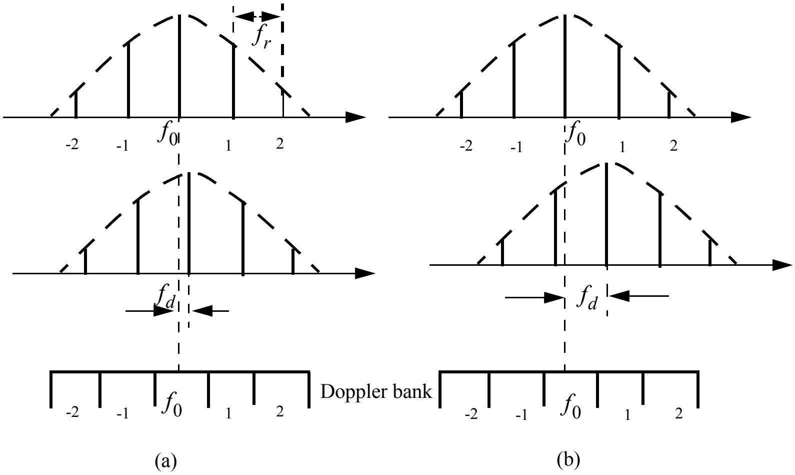


Figure 1A.2. **Spectra of transmitted and received waveforms, and Doppler bank. (a) Doppler is resolved. (b) Spectral lines have moved into the next Doppler filter. This results in an ambiguous Doppler measurement.**

We choose f_{r1} and f_{r2} 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,

$$t_1 + \frac{M}{f_{r1}} = t_2 + \frac{M}{f_{r2}} \tag{1A.2}$$

for which we get

$$M = \frac{t_2 - t_1}{T_1 - T_2} \tag{1A.3}$$

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

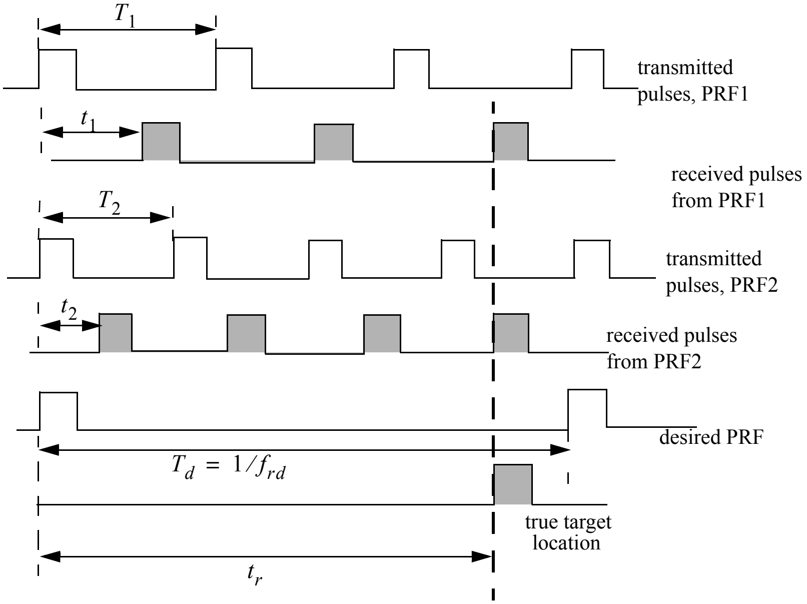


Figure 1A.3. Resolving range ambiguity.

$$
\begin{aligned}
t_r &= MT_1 + t_1 \\
t_r &= MT_2 + t_2
\end{aligned}
\tag{1A.4}
$$

and the true target range is

$$R = ct_r/2 \tag{1A.5}$$

Now if $t_1 > t_2$, then

$$t_1 + \frac{M}{f_{r1}} = t_2 + \frac{M+1}{f_{r2}} \tag{1A.6}$$

Solving for M we get

$$M = \frac{(t_2 - t_1) + T_2}{T_1 - T_2} \tag{1A.7}$$

and the round-trip time to the true target location is

$$t_{r1} = MT_1 + t_1 \tag{1A.8}$$

and in this case, the true target range is

$$R = \frac{ct_{r1}}{2} \tag{1A.9}$$

Finally, if $t_1 = t_2$, then the target is in the first ambiguity. It follows that

$$t_{r2} = t_1 = t_2 \tag{1A.10}$$

and

$$R = ct_{r2}/2 \tag{1A.11}$$

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. In this case, one may choose $f_{r1} = N(N+1)f_{rd}$, $f_{r2} = N(N+2)f_{rd}$, and $f_{r3} = (N+1)(N+2)f_{rd}$.

1A.4. Resolving Doppler Ambiguity

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

$$M = \frac{(f_{d2} - f_{d1}) + f_{r2}}{f_{r1} - f_{r2}} \tag{1A.12}$$

And if $f_{d1} < f_{d2}$,

$$M = \frac{f_{d2} - f_{d1}}{f_{r1} - f_{r2}} \tag{1A.13}$$

and the true Doppler is

$$\begin{aligned} f_d &= Mf_{r1} + f_{d1} \\ f_d &= Mf_{r2} + f_{d2} \end{aligned} \tag{1A.14}$$

Finally, if $f_{d1} = f_{d2}$, then

$$f_d = f_{d1} = f_{d2} \tag{1A.15}$$

Again, blind Doppler 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

$$f_d = 2\frac{vf_0}{c} = \frac{2 \times 550 \times 9 \times 10^9}{3 \times 10^8} = 33KHz$$

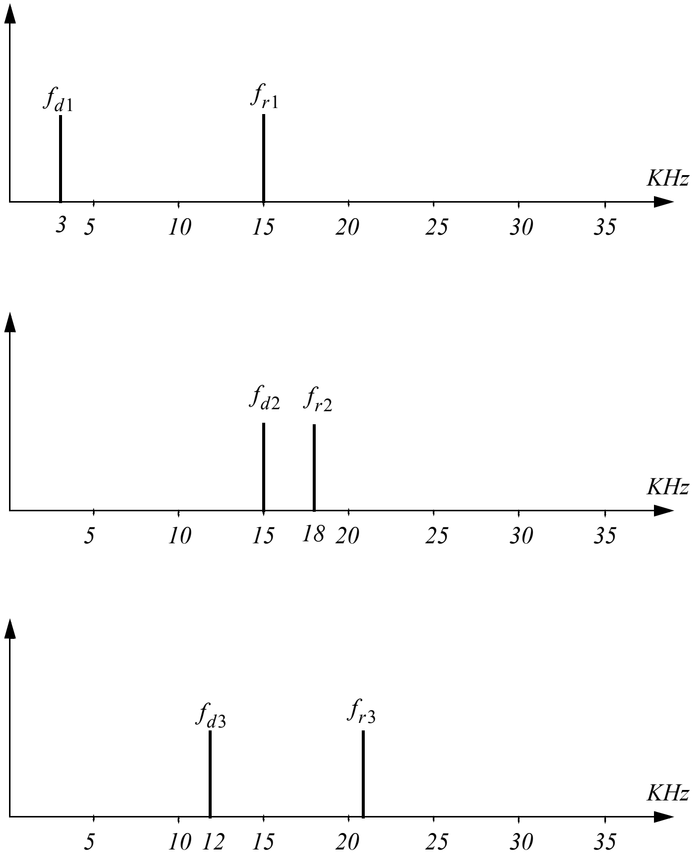
Then by using Eq. (1A.14) $n_i f_{ri} + f_{di} = f_d$ *where* $i = 1, 2, 3$ *, we can write*

$$n_1 f_{r1} + f_{d1} = 15n_1 + f_{d1} = 33$$

$$n_2 f_{r2} + f_{d2} = 18n_2 + f_{d2} = 33$$

$$n_3 f_{r3} + f_{d3} = 21n_3 + f_{d3} = 33$$

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* $f_{d3} = 12KHz$ *.*



Now for the second part of the problem. Again by using Eq. (1A.14) 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	$\underline{38}$	53	68
f_d from f_{r2}	2	20	$\underline{38}$	56	
f_d from f_{r3}	17	$\underline{38}$	39		

Thus, $n_1 = 2 = n_2$ *, and* $n_3 = 1$*, and the true target Doppler is* $f_d = 38KHz$ *. It follows that*

$$v_r = 38000 \times \frac{0.0333}{2} = 632.7\frac{m}{\text{sec}}$$

Appendix 1B Noise Figure

1B.1. Noise Figure

Any signal other than the target returns in the radar receiver is considered to be noise. This includes interfering signals from outside the radar and thermal noise generated within the receiver itself. Thermal noise (thermal agitation of electrons) and shot noise (variation in carrier density of a semiconductor) are the two main internal noise sources within a radar receiver.

The power spectral density of thermal noise is given by

$$S_n(\omega) = \frac{|\omega|h}{\pi\left[\exp\left(\frac{|\omega|h}{2\pi kT}\right) - 1\right]} \tag{1B.1}$$

where $|\omega|$ is the absolute value of the frequency in radians per second, T is the temperature of the conducting medium in degrees Kelvin, k is Boltzman's constant, and h is Plank's constant ($h = 6.625 \times 10^{-34}$ $joule$ $\sec onds$). When the condition $|\omega| \ll 2\pi kT/h$ is true, it can be shown that Eq. (1B.1) is approximated by

$$S_n(\omega) \approx 2kT \tag{1B.2}$$

This approximation is widely accepted, since, in practice, radar systems operate at frequencies less than 100 GHz; and, for example, if $T = 290K$, then $2\pi kT/h \approx 6000$ GHz.

The mean square noise voltage (noise power) generated across a $1\ ohm$ resistance is then

$$\langle n^2 \rangle = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} 2kT \quad d\omega = 4kTB \tag{1B.3}$$

where B is the system bandwidth in hertz.

Any electrical system containing thermal noise and having input resistance R_{in} can be replaced by an equivalent noiseless system with a series combination of a noise equivalent voltage source and a noiseless input resistor R_{in} added at its input. This is illustrated in Fig. 1B.1.

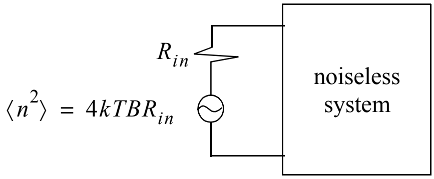


Figure 1B.1. Noiseless system with an input noise voltage source.

The amount of noise power that can physically be extracted from $\langle n^2 \rangle$ is one fourth the value computed in Eq. (1B.3). The proof is left as an exercise.

Consider a noisy system with power gain A_P, as shown in Fig. 1B.2. The noise figure is defined by

$$F_{dB} = 10 \log \frac{total\ noise\ power\ out}{noise\ power\ out\ due\ to\ R_{in}\ alone} \tag{1B.4}$$

More precisely,

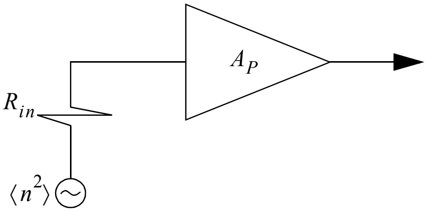


Figure 1B.2. Noisy amplifier replaced by its noiseless equivalent and an input voltage source in series with a resistor.

$$F_{dB} = 10 \ \log \frac{N_o}{N_i \ A_p} \tag{1B.5}$$

where N_o and N_i are, respectively, the noise power at the output and input of the system.

If we define the input and output signal power by S_i and S_o, respectively, then the power gain is

$$A_P = \frac{S_o}{S_i} \tag{1B.6}$$

It follows that

$$F_{dB} = 10\log\left(\frac{S_i/Ni}{S_o/N_o}\right) = \left(\frac{S_i}{N_i}\right)_{dB} - \left(\frac{S_o}{N_o}\right)_{dB} \tag{1B.7}$$

where

$$\left(\frac{S_i}{N_i}\right)_{dB} > \left(\frac{S_o}{N_o}\right)_{dB} \tag{1B.8}$$

Thus, it can be said that the noise figure is the loss in the signal-to-noise ratio due to the added thermal noise of the amplifier $((SNR)_o = (SNR)_i - F \ \ in \ \ dB)$.

We can also express the noise figure in terms of the system's effective temperature T_e. Consider the amplifier shown in Fig. 1B.2, and let its effective temperature be T_e. Assume the input noise temperature is T_o. Thus, the input noise power is

$$N_i = kT_oB \tag{1B.9}$$

and the output noise power is

$$N_o = kT_oB \ A_p + kT_eB \ A_p \tag{1B.10}$$

where the first term on the right-hand side of Eq. (1B.10) corresponds to the input noise, and the latter term is due to thermal noise generated inside the system. It follows that the noise figure can be expressed as

$$F = \frac{(SNR)_i}{(SNR)_o} = \frac{S_i}{kT_oB} \ kBA_p \ \frac{T_o + T_e}{S_o} = 1 + \frac{T_e}{T_o} \tag{1B.11}$$

Equivalently, we can write

$$T_e = (F - 1)T_o \tag{1B.12}$$

Example:

An amplifier has a 4dB noise figure; the bandwidth is $B = 500\ KHz$ *. Calculate the input signal power that yields a unity SNR at the output. Assume* $T_o = 290$ *degrees Kelvin and an input resistance of one ohm.*

Solution:

The input noise power is

$$kT_oB = 1.38 \times 10^{-23} \times 290 \times 500 \times 10^3 = 2.0 \times 10^{-15} w$$

Assuming a voltage signal, then the input noise mean squared voltage is

$$\langle n_i^2 \rangle = kT_oB = 2.0 \times 10^{-15}\ v^2$$

$$F = 10^{0.4} = 2.51$$

From the noise figure definition we get

$$\frac{S_i}{N_i} = F\left(\frac{S_o}{N_o}\right) = F$$

and

$$\langle s_i^2 \rangle = F\langle n_i^2 \rangle = 2.51 \times 2.0 \times 10^{-15} = 5.02 \times 10^{-15}\ v^2$$

Finally,

$$\sqrt{\langle s_i^2 \rangle} = 70.852 nv$$

Consider a cascaded system as in Fig. 1B.3. Network 1 is defined by noise figure F_1, power gain G_1, bandwidth B, and temperature T_{e1}. Similarly, network 2 is defined by F_2, G_2, B, and T_{e2}. Assume the input noise has temperature T_0.

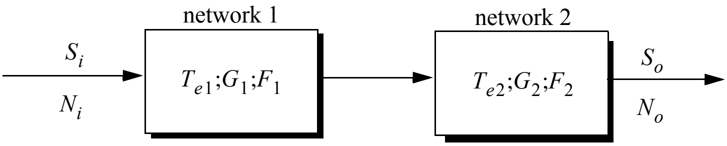


Figure 1B.3. Cascaded linear system.

The output signal power is

$$S_o = S_i G_1 G_2 \tag{1B.13}$$

The input and output noise powers are, respectively, given by

$$N_i = kT_o B \tag{1B.14}$$

$$N_o = kT_0 B G_1 G_2 + kT_{e1} B G_1 G_2 + kT_{e2} B G_2 \tag{1B.15}$$

where the three terms on the right-hand side of Eq. (1B.15), respectively, correspond to the input noise power, thermal noise generated inside network 1, and thermal noise generated inside network 2.

Now if we use the relation $T_e = (F-1)T_0$ along with Eq. (1B.13) and Eq. (1B.14), we can express the overall output noise power as

$$N_o = F_1 N_i G_1 G_2 + (F_2 - 1) N_i G_2 \tag{1B.16}$$

It follows that the overall noise figure for the cascaded system is

$$F = \frac{(S_i/N_i)}{(S_o/N_o)} = F_1 + \frac{F_2 - 1}{G_1} \tag{1B.17}$$

In general, for an n-stage system we get

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \cdot \cdot \cdot + \frac{F_n - 1}{G_1 G_2 G_3 \cdot \cdot \cdot G_{n-1}} \tag{1B.18}$$

Also, the n-stage system effective temperatures can be computed as

$$T_e = T_{e1} + \frac{T_{e2}}{G_1} + \frac{T_{e3}}{G_1 G_2} + \cdot \cdot \cdot + \frac{T_{en}}{G_1 G_2 G_3 \cdot \cdot \cdot G_{n-1}} \tag{1B.19}$$

As suggested by Eq. (1B.18) and Eq. (1B.19), the overall noise figure is mainly dominated by the first stage. Thus, radar receivers employ low noise power amplifiers in the first stage in order to minimize the overall receiver noise figure. However, for radar systems that are built for low RCS operations every stage should be included in the analysis.

Example:

A radar receiver consists of an antenna with cable loss $L = 1dB = F_1$, an RF amplifier with $F_2 = 6dB$, and gain $G_2 = 20dB$, followed by a mixer whose noise figure is $F_3 = 10dB$ and conversion loss $L = 8dB$, and finally, an integrated circuit IF amplifier with $F_4 = 6dB$ and gain $G_4 = 60dB$. Find the overall noise figure.

Solution:

From Eq. (1B.18) we have

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3}$$

G_1	G_2	G_3	G_4	F_1	F_2	F_3	F_4
$-1dB$	$20dB$	$-8dB$	$60dB$	$1dB$	$6dB$	$10dB$	$6dB$
0.7943	100	0.1585	10^6	1.2589	3.9811	10	3.9811

It follows that

$$F = 1.2589 + \frac{3.9811 - 1}{0.7943} + \frac{10 - 1}{100 \times 0.7943} + \frac{3.9811 - 1}{0.158 \times 100 \times 0.7943} = 5.3629$$

$$F = 10\log(5.3628) = 7.294dB$$

Chapter 2 *Radar Detection*

2.1. 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. 2.1. The input signal to the receiver is composed of the radar echo signal $s(t)$ and additive zero mean white Gaussian noise $n(t)$, with variance ψ^2. The input noise is assumed to be spatially incoherent and uncorrelated with the signal.

The output of the bandpass IF filter is the signal $v(t)$, which can be written as

$$\begin{aligned} v(t) &= v_I(t)\cos\omega_0 t + v_Q(t)\sin\omega_0 t = r(t)\cos(\omega_0 t - \varphi(t)) \\ v_I(t) &= r(t)\cos\varphi(t) \\ v_Q(t) &= r(t)\sin\varphi(t) \end{aligned} \tag{2.1}$$

where $\omega_0 = 2\pi f_0$ is the radar operating frequency, $r(t)$ is the envelope of $v(t)$, the phase is $\varphi(t) = \mathrm{atan}(v_Q/v_I)$, and the subscripts I, 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

$$\begin{array}{ll} s(t) + n(t) > V_T & \quad Detection \\ n(t) > V_T & \quad False\ \ alarm \end{array}$$

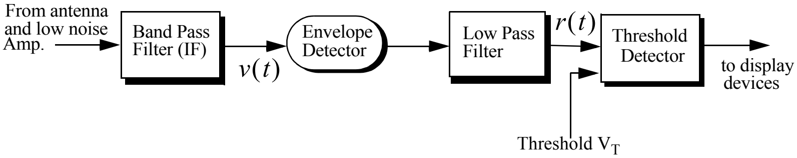


Figure 2.1. Simplified block diagram of an envelope detector and threshold receiver.

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 first case are

$$\begin{aligned} v_I(t) &= n_I(t) \\ v_Q(t) &= n_Q(t) \end{aligned} \tag{2.2}$$

and for the second case,

$$\begin{gathered} v_I(t) = A + n_I(t) = r(t)\cos\varphi(t) \Rightarrow n_I(t) = r(t)\cos\varphi(t) - A \\ v_Q(t) = n_Q(t) = r(t)\sin\varphi(t) \end{gathered} \tag{2.3}$$

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

$$\begin{aligned} f(n_I, n_Q) &= \frac{1}{2\pi\psi^2}\exp\left(-\frac{n_I^2 + n_Q^2}{2\psi^2}\right) \\ &= \frac{1}{2\pi\psi^2}\exp\left(-\frac{(r\cos\varphi - A)^2 + (r\sin\varphi)^2}{2\psi^2}\right) \end{aligned} \tag{2.4}$$

The *pdfs* of the random variables $r(t)$ and $\varphi(t)$, respectively, represent the modulus and phase of $v(t)$. The joint *pdf* for the two random variables $r(t);\varphi(t)$ is given by

$$f(r, \varphi) = f(n_I, n_Q)|J| \tag{2.5}$$

where $[J]$ is a matrix of derivatives defined by

$$[J] = \begin{bmatrix} \frac{\partial n_I}{\partial r} & \frac{\partial n_I}{\partial \varphi} \\ \frac{\partial n_Q}{\partial r} & \frac{\partial n_Q}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -r\sin\varphi \\ \sin\varphi & r\cos\varphi \end{bmatrix} \tag{2.6}$$

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

$$|J| = r(t) \tag{2.7}$$

Substituting Eqs. (2.4) and (2.7) into Eq. (2.5) and collecting terms yield

$$f(r, \varphi) = \frac{r}{2\pi\psi^2}\exp\left(-\frac{r^2 + A^2}{2\psi^2}\right)\exp\left(\frac{rA\cos\varphi}{\psi^2}\right) \tag{2.8}$$

The *pdf* for r alone is obtained by integrating Eq. (2.8) over φ

$$f(r) = \int_0^{2\pi} f(r, \varphi)d\varphi = \frac{r}{\psi^2}\exp\left(-\frac{r^2 + A^2}{2\psi^2}\right)\frac{1}{2\pi}\int_0^{2\pi}\exp\left(\frac{rA\cos\varphi}{\psi^2}\right)d\varphi \tag{2.9}$$

where the integral inside Eq. (2.9) is known as the modified Bessel function of zero order,

$$I_0(\beta) = \frac{1}{2\pi}\int_0^{2\pi} e^{\beta\cos\theta}\ d\theta \tag{2.10}$$

Thus,

$$f(r) = \frac{r}{\psi^2} I_0\left(\frac{rA}{\psi^2}\right)\exp\left(-\frac{r^2 + A^2}{2\psi^2}\right) \tag{2.11}$$

which is the Rician probability density function. If $A/\psi^2 = 0$ (noise alone), then Eq. (2.11) becomes the Rayleigh probability density function

$$f(r) = \frac{r}{\psi^2}\exp\left(-\frac{r^2}{2\psi^2}\right) \tag{2.12}$$

Also, when (A/ψ^2) is very large, Eq. (2.11) becomes a Gaussian probability density function of mean A and variance ψ^2:

$$f(r) \approx \frac{1}{\sqrt{2\pi\psi^2}} \exp\left(-\frac{(r-A)^2}{2\psi^2}\right) \tag{2.13}$$

Fig. 2.2 shows plots for the Rayleigh and Gaussian densities. For this purpose, use MATLAB program *"fig2_2.m"* given in Listing 2.1 in Section 2.11. This program uses MATLAB functions *"normpdf.m"* and *"raylpdf.m"*. Both functions are part of the MATLAB Statistics toolbox. Their associated syntax is as follows

normpdf(x,mu,sigma)

raylpdf(x,sigma)

"x" is the variable, *"mu"* is the mean, and *"sigma"* is the standard deviation.

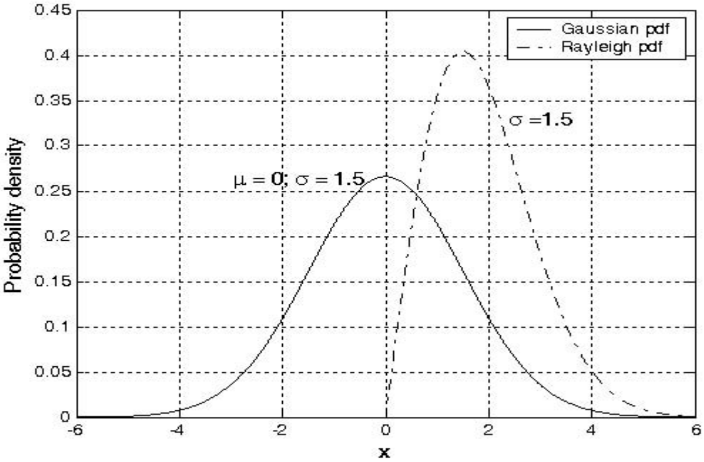


Figure 2.2. Gaussian and Rayleigh probability densities.

The density function for the random variable φ is obtained from

$$f(\varphi) = \int_0^r f(r, \varphi)\ dr \tag{2.14}$$

While the detailed derivation is left as an exercise, the result of Eq. (2.14) is

$$f(\varphi) = \frac{1}{2\pi} \exp\left(\frac{-A^2}{2\psi^2}\right) + \frac{A\cos\varphi}{\sqrt{2\pi\psi^2}} \exp\left(\frac{-(A\sin\varphi)^2}{2\psi^2}\right) F\left(\frac{A\cos\varphi}{\psi}\right) \tag{2.15}$$

where

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \; e^{-\zeta^2/2} \; d\xi \tag{2.16}$$

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. (2.15) collapses to a uniform *pdf* over the interval $\{0, 2\pi\}$.

One excellent approximation for the function $F(x)$ is

$$F(x) = 1 - \left(\frac{1}{0.661x + 0.339\sqrt{x^2 + 5.51}} \right) \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \qquad x \ge 0 \tag{2.17}$$

and for negative values of x

$$F(-x) = 1 - F(x) \tag{2.18}$$

MATLAB Function "que_func.m"

The function *"que_func.m"* computes $F(x)$ using Eqs. (2.17) and (2.18) and is given in Listing 2.2 in Section 2.11. The syntax is as follows:

fofx = que_func (x)

2.2. 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,

$$P_{fa} = \int_{V_T}^{\infty} \frac{r}{\psi^2} \exp\left(-\frac{r^2}{2\psi^2} \right) \; dr = \exp\left(\frac{-V_T^2}{2\psi^2} \right) \tag{2.19a}$$

$$V_T = \sqrt{2\psi^2 \ln\left(\frac{1}{P_{fa}} \right)} \tag{2.19b}$$

Fig. 2.3 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. This figure can be reproduced using MATLAB program *"fig2_3.m"* given in Listing 2.3 in Section 2.11.

The false alarm time T_{fa} is related to the probability of false alarm by

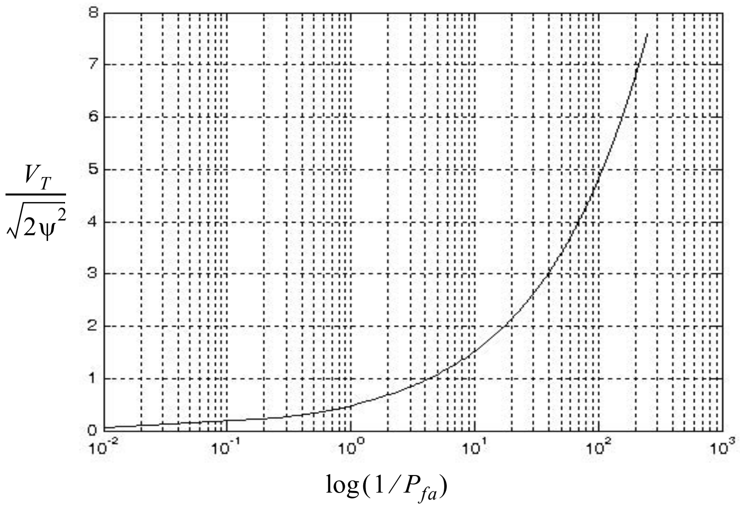


Figure 2.3. Normalized detection threshold versus probability of false alarm.

$$T_{fa} = t_{int}/P_{fa} \tag{2.20}$$

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}, then by substituting Eq. (2.19) into Eq. (2.20) we can write T_{fa} as

$$T_{fa} = \frac{1}{B}\exp\left(\frac{V_T^2}{2\psi^2}\right) \tag{2.21}$$

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

Fehlner[1] defines the false alarm number as

$$n_{fa} = \frac{-\ln(2)}{\ln(1-P_{fa})} \approx \frac{\ln(2)}{P_{fa}} \tag{2.22}$$

[1] 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.

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. (2.22), 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. (2.22) is always assumed. Hence, a clear distinction is made between Marcum's definition of the false alarm number and the definition in Eq. (2.22).

2.3. 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,

$$P_D = \int_{V_T}^{\infty} \frac{r}{\psi^2} I_0\left(\frac{rA}{\psi^2}\right) \exp\left(-\frac{r^2 + A^2}{2\psi^2}\right) dr \qquad \textbf{(2.23)}$$

If we assume that the radar signal is a sine waveform with amplitude A, then its power is $A^2/2$. Now, by using $SNR = A^2/2\psi^2$ (single-pulse SNR) and $(V_T^2/2\psi^2) = \ln(1/P_{fa})$, then Eq. (2.23) can be rewritten as

$$P_D = \int_{\sqrt{2\psi^2 \ln(1/p_{fa})}}^{\infty} \frac{r}{\psi^2} I_0\left(\frac{rA}{\psi^2}\right) \exp\left(-\frac{r^2 + A^2}{2\psi^2}\right) dr = \qquad \textbf{(2.24)}$$

$$Q\left[\sqrt{\frac{A^2}{\psi^2}}, \sqrt{2\ln\left(\frac{1}{P_{fa}}\right)}\right]$$

$$Q[\alpha, \beta] = \int_{\beta}^{\infty} \zeta I_0(\alpha\zeta) e^{-(\zeta^2 + \alpha^2)/2}\, d\zeta \qquad \textbf{(2.25)}$$

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. (2.24) can be approximated by

$$P_D \approx F\left(\frac{A}{\psi} - \sqrt{2\ln\left(\frac{1}{P_{fa}}\right)}\right) \qquad \textbf{(2.26)}$$

where $F(x)$ is given by Eq. (2.16). Many approximations for computing Eq. (2.24) can be found throughout the literature. One very accurate approximation presented by North (see bibliography) is given by

$$P_D \approx 0.5 \times erfc(\sqrt{-\ln P_{fa}} - \sqrt{SNR + 0.5}) \qquad \textbf{(2.27)}$$

where the complementary error function is

$$erfc(z) = 1 - \frac{2}{\sqrt{\pi}} \int_0^z e^{-v^2} dv \tag{2.28}$$

MATLAB Function "marcumsq.m"

The integral given in Eq. (2.24) 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:

$$Q[a, b] = \left\{ \begin{array}{ll} \frac{\alpha_n}{2\beta_n} \exp\left(\frac{(a-b)^2}{2}\right) & a < b \\ 1 - \left(\frac{\alpha_n}{2\beta_n} \exp\left(\frac{(a-b)^2}{2}\right) \right. & \left. a \ge b\right) \end{array} \right\} \tag{2.29}$$

$$\alpha_n = d_n + \frac{2n}{ab}\alpha_{n-1} + \alpha_{n-2} \tag{2.30}$$

$$\beta_n = 1 + \frac{2n}{ab}\beta_{n-1} + \beta_{n-2} \tag{2.31}$$

$$d_{n+1} = d_n d_1 \tag{2.32}$$

$$\alpha_0 = \left\{ \begin{array}{ll} 1 & a < b \\ 0 & a \ge b \end{array} \right\} \tag{2.33}$$

$$d_1 = \left\{ \begin{array}{ll} a/b & a < b \\ b/a & a \ge b \end{array} \right\} \tag{2.34}$$

$\alpha_{-1} = 0.0$, $\beta_0 = 0.5$, and $\beta_{-1} = 0$. The recursive Eqs. (2.30) through (2.32) are computed continuously until $\beta_n > 10^p$ for values of $p \ge 3$. The accuracy of the algorithm is enhanced as the value of p is increased. The MATLAB function *"marcumsq.m"* given in Listing 2.4 in Section 2.11 implements Parl's algorithm to calculate the probability of detection defined in Eq. (2.24). The syntax is as follows:

Pd = marcumsq (alpha, beta)

where *alpha* and *beta* are from Eq. (2.25). Fig. 2.4 shows plots of the probability of detection, P_D, versus the single pulse SNR, with the P_{fa} as a parameter. This figure can be reproduced using the MATLAB program *"prob_snr1.m"* given in Listing 2.5 in Section 2.11.

[1] 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.

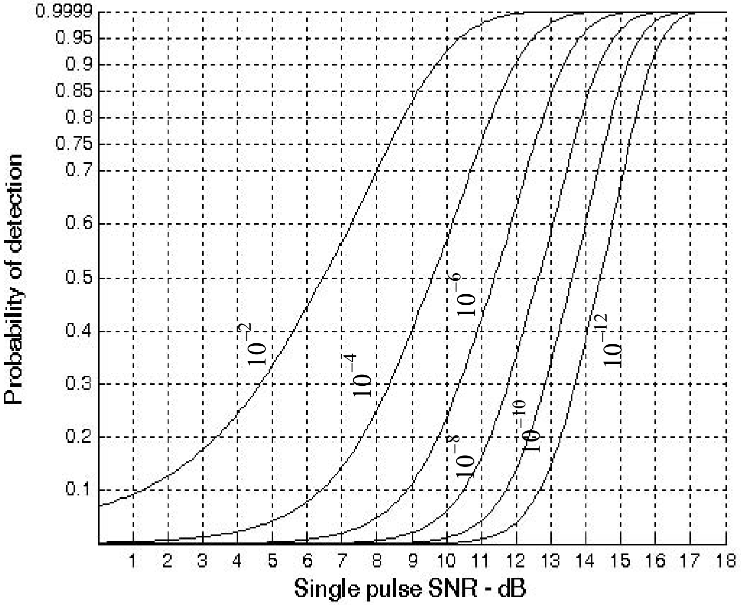


Figure 2.4. Probability of detection versus single pulse SNR, for several values of P_{fa}.

2.4. Pulse Integration

Pulse integration was discussed in Chapter 1 in the context of radar measurements. In this section a more comprehensive analysis of this topic is introduced in the context of radar detection. The overall principles and conclusions presented earlier will not change; however, the mathematical formulation and specific numerical values will change. Coherent integration preserves the phase relationship between the received pulses, thus achieving a build up in the signal amplitude. Alternatively, pulse integration performed after the envelope detector (where the phase relation is destroyed) is called non-coherent or post-detection integration.

2.4.1. Coherent Integration

In coherent integration, if a perfect integrator is used (100% efficiency), then integrating n_P pulses would improve the SNR by the same factor. Otherwise, integration loss occurs which is always the case for non-coherent integration. In order to demonstrate this signal buildup, consider the case where the radar return signal contains both signal plus additive noise. The m^{th} pulse is

$$y_m(t) = s(t) + n_m(t) \tag{2.35}$$

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

$$z(t) = \frac{1}{n_P}\sum_{m=1}^{n_P} y_m(t) = \sum_{m=1}^{n_P} \frac{1}{n_P}[s(t) + n_m(t)] = s(t) + \sum_{m=1}^{n_P} \frac{1}{n_P} n_m(t) \tag{2.36}$$

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

$$\psi_{nz}^2 = E\left[\left(\sum_{m=1}^{n_P} \frac{1}{n_P} n_m(t)\right)\left(\sum_{l=1}^{n_P} \frac{1}{n_P} n_l(t)\right)^*\right] \tag{2.37}$$

where $E[\ ]$ is the expected value operator. It follows that

$$\psi_{nz}^2 = \frac{1}{n_P^2} \sum_{m,l=1}^{n_P} E[n_m(t) n_l^*(t)] = \frac{1}{n_P^2} \sum_{m,l=1}^{n_P} \psi_{ny}^2 \delta_{ml} = \frac{1}{n_P} \psi_{ny}^2 \tag{2.38}$$

where ψ_{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. (2.36) and (2.38) 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.

Denote the single pulse SNR required to produce a given probability of detection as $(SNR)_1$. Also, denote $(SNR)_{n_P}$ as the SNR required to produce the same probability of detection when n_P pulses are integrated. It follows that

$$(SNR)_{n_P} = \frac{1}{n_P}(SNR)_1 \tag{2.39}$$

The requirements of knowing the exact phase of each transmitted pulse as well as maintaining coherency during propagation is very costly and challenging to achieve. Thus, radar systems would not utilize coherent integration during search mode, since target micro-dynamics may not be available.

2.4.2. Non-Coherent Integration

Non-coherent integration is often implemented after the envelope detector, also known as the quadratic detector. A block diagram of radar receiver utilizing a square law detector and non-coherent integration is illustrated in Fig. 2.5. In practice, the square law detector is normally used as an approximation to the optimum receiver.

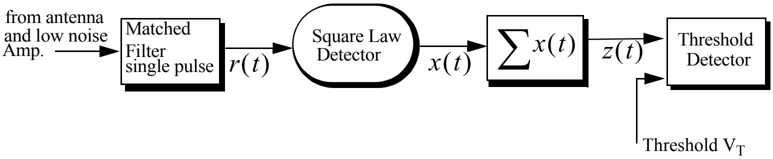


Figure 2.5. Simplified block diagram of a square law detector and non-coherent integration.

The *pdf* for the signal $r(t)$ was derived earlier and it is given in Eq. (2.11). Define a new dimensionless variable y as

$$y_n = \frac{r_n}{\psi} \tag{2.40}$$

and also define

$$\Re_p = \frac{A^2}{\psi^2} = 2SNR \tag{2.41}$$

It follows that the *pdf* for the new variable is then given by

$$f(y_n) = f(r_n)\left|\frac{dr_n}{dy_n}\right| = y_n \ I_0(y_n\sqrt{\Re_p}) \ \exp\left(\frac{-(y_n^2 + \Re_p)}{2}\right) \tag{2.42}$$

The output of a square law detector for the n^{th} pulse is proportional to the square of its input, which, after the change of variable in Eq. (2.40), is proportional to y_n. Thus, it is convenient to define a new change variable,

$$x_n = \frac{1}{2}y_n^2 \tag{2.43}$$

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

$$f(x_n) = f(y_n)\left|\frac{dy_n}{dx_n}\right| = \exp\left(-\left(x_n + \frac{\Re_p}{2}\right)\right)I_0(\sqrt{2x_n\Re_p}) \tag{2.44}$$

Non-coherent integration of n_p pulses is implemented as

$$z = \sum_{n=1}^{n_P} x_n \tag{2.45}$$

Since the random variables x_n are independent, the *pdf* for the variable z is

$$f(z) = f(x_1) \bullet f(x_2) \bullet \dots \bullet f(x_{n_p}) \tag{2.46}$$

The operator $\bullet$ symbolically indicates convolution. The characteristic functions for the individual *pdf*s can then be used to compute the joint *pdf* in Eq. (2.46). The details of this development are left as an exercise. The result is

$$f(z) = \left(\frac{2z}{n_P \Re_p}\right)^{(n_P - 1)/2} \exp\left(-z - \frac{1}{2} n_P \Re_p\right) I_{n_P - 1}(\sqrt{2 n_P z \Re_p}) \tag{2.47}$$

$I_{n_P - 1}$ is the modified Bessel function of order $n_P - 1$. Therefore, the probability of detection is obtained by integrating $f(z)$ from the threshold value to infinity. Alternatively, the probability of false alarm is obtained by letting $\Re_p$ be zero and integrating the *pdf* from the threshold value to infinity. Closed form solutions to these integrals are not easily available. Therefore, numerical techniques are often utilized to generate tables for the probability of detection.

Improvement Factor and Integration Loss

Denote the SNR that is required to achieve a specific P_D given a particular P_{fa} when n_P pulses are integrated non-coherently by $(SNR)_{NCI}$. And thus, the single pulse SNR, $(SNR)_1$, is less than $(SNR)_{NCI}$. More precisely,

$$(SNR)_{NCI} = (SNR)_1 \times I(n_P) \tag{2.48}$$

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[1] as

$$[I(n_P)]_{dB} = 6.79(1 + 0.235 P_D)\left(1 + \frac{\log(1/P_{fa})}{46.6}\right)\log(n_P) \tag{2.49}$$
$$(1 - 0.140\log(n_P) + 0.018310(\log n_P)^2)$$

Fig. 2.6a 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. (2.49). This plot can be reproduced using the MATLAB program *"fig2_6a.m"* given in Listing 2.6 in Section 2.11. Note this program uses the MATLAB function *"improv_fac.m"*, which is given in Listing 2.7 in Section 2.11.

MATLAB Function "improv_fac.m"

The function *"improv_fac.m"* calculates the improvement factor using Eq. (2.49). It is given in Listing 2.7 in Section 2.11. The syntax is as follows:

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

[1] Peebles Jr., P. Z., *Radar Principles*, John Wiley & Sons, Inc., 1998.

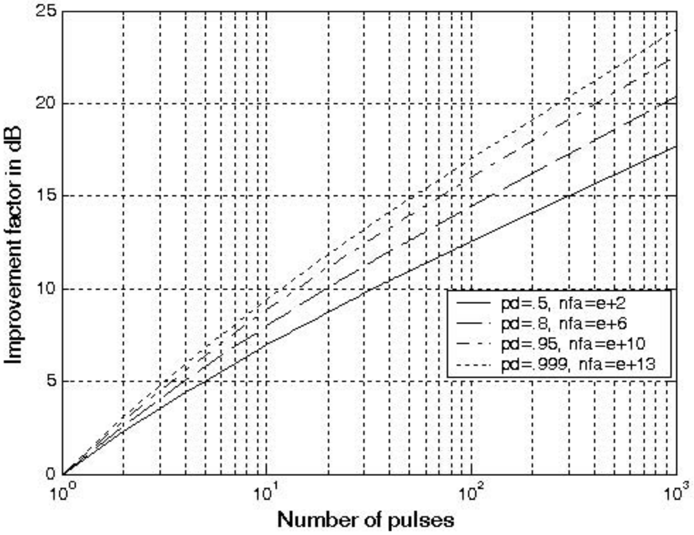


Figure 2.6a. Improvement factor versus number of non-coherently integrated pulses.

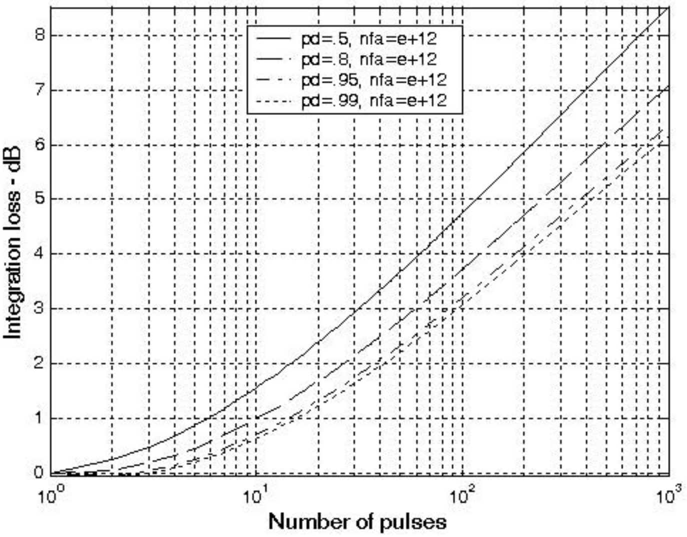


Figure 2.6b. Integration loss versus number of non-coherently integrated pulses.

where

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

The integration loss is defined as

$$L_{NCI} = n_P / I(n_P) \tag{2.50}$$

Figure 2.6b shows a plot of the integration loss versus n_P. This figure can be reproduced using MATLAB program *"fig2_6b.m"* given in Listing 2.8 in Section 2.11. It follows that, when non-coherent integration is utilized, the corresponding SNR required to achieve a certain P_D given a specific P_{fa} is now given by

$$(SNR)_{NCI} = (n_P \times (SNR)_1) / L_{NCI} \tag{2.51}$$

which is very similar to Eq. (1.86) derived in Chapter 1.

2.4.3. Mini Design Case Study 2.1

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 $R = 12Km$, the minimum required SNR is $SNR = 13.85dB$, antenna gain $G = 5000$, and target RCS $\sigma = 1m^2$. (a) Determine the PRF f_r, the pulsewidth τ, the peak power P_t, the probability of false alarm P_{fa}, the corresponding P_D, 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 non-coherently? (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 $9Km$.

A Solution

Assume that the maximum detection corresponds to the unambiguous range. From that the PRF is computed as

$$f_r = \frac{c}{2R_u} = \frac{3 \times 10^8}{2 \times 12000} = 12.5KHz$$

The pulsewidth is proportional to the inverse of the bandwidth,

$$\tau = \frac{1}{B} = \frac{1}{2 \times 10^6} = 0.5\mu s$$

The probability of false alarm is

$$P_{fa} = \frac{1}{BT_{fa}} = \frac{1}{2 \times 10^6 \times 12 \times 60} = 6.94 \times 10^{-10}$$

It follows that by using MATLAB function "marcumsq.m" the probability of detection is calculated from

$$Q\left[\sqrt{\frac{A^2}{\psi^2}}, \sqrt{2\ln\left(\frac{1}{P_{fa}}\right)}\right]$$

with the following syntax

$$marcumsq(alpha,\ beta)$$

where

$$alpha = \sqrt{2} \times \sqrt{10^{13.85/10}} = 6.9665$$

$$beta = \sqrt{2\ln\left(\frac{1}{6.94 \times 10^{-10}}\right)} = 6.494$$

Remember that $(A^2/\psi^2) = 2SNR$. *Thus, the detection probability is*

$$P_D = marcumsq(6.9665, 6.944) = 0.508$$

Using the radar equation one can calculate the radar peak power. More precisely,

$$P_t = SNR\ \frac{(4\pi)^3 R^4 kT_0 BFL}{G^2 \lambda^2 \sigma} \Rightarrow$$

$$P_t = 10^{1.385} \frac{(4\pi)^3 \times 12000^4 \times 1.38 \times 10^{-23} \times 290 \times 2 \times 10^6 \times 6.309 \times 2.511}{5000^2 \times 0.2^2 \times 1}$$

$$= 126.61\,Watts$$

And the minimum detectable signal is

$$S_{min} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L} = \frac{126.61 \times 5000^2 \times 0.2^2 \times 1}{(4\pi)^3 \times 12000^4 \times 2.511} = 1.2254 \times 10^{-12} Volts$$

When 10 pulses are integrated non-coherently, the corresponding improvement factor is calculated from the MATLAB function "improv_fac.m" using the following syntax

improv_fac (10,1e-11,0.5)

which yields $I(10) = 6 \Rightarrow 7.78dB$. *Consequently, by keeping the probability of detection the same (with and without integration) the SNR can be reduced by a factor of almost 6 dB (13.85 - 7.78). The integration loss associated with a 10-pulse non-coherent integration is calculated from Eq. (2.50) as*

$$L_{NCI} = \frac{n_P}{I(10)} = \frac{10}{6} = 1.67 \Rightarrow 2.2dB$$

Thus the net single pulse SNR with 10-pulse non-coherent integration is

$$(SNR)_{NCI} = 13.85 - 7.78 + 2.2 = 8.27dB .$$

Finally, the improvement in the SNR due to decreasing the detection range to 9 Km is

$$(SNR)_{9Km} = 10\log\left(\frac{12000}{9000}\right)^4 + 13.85 = 18.85dB .$$

2.5. Detection of Fluctuating Targets

So far the probability of detection calculations assumed a constant target cross section (non-fluctuating target). This work was first analyzed by Marcum.[1] Swerling[2] extended Marcum's work to four distinct cases that account for variations in the target cross section. 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 lowers the probability of detection, or equivalently reduces the SNR.

[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.

Swerling I targets have constant amplitude over one antenna scan; 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 similar to Swerling I, except in this case the target power fluctuates independently from pulse to pulse 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. Non-coherent 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

$$f(\sigma) = \frac{N}{(N-1)!\ \bar{\sigma}} \left(\frac{N\sigma}{\bar{\sigma}}\right)^{N-1} \exp\left(-\frac{N\sigma}{\bar{\sigma}}\right) \tag{2.52}$$

where $\bar{\sigma}$ is the average 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,

$$f(\sigma) = \frac{1}{\bar{\sigma}} \exp\left(-\frac{\sigma}{\bar{\sigma}}\right) \qquad \sigma \geq 0 \tag{2.53}$$

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

$$f(\sigma) = \frac{4\sigma}{\bar{\sigma}^2} \exp\left(-\frac{2\sigma}{\bar{\sigma}}\right) \qquad \sigma \geq 0 \tag{2.54}$$

The probability of detection for a fluctuating target is computed in a similar fashion to Eq. (2.23), except in this case $f(r)$ is replaced by the conditional *pdf* $f(r/\sigma)$. Performing the analysis for the general case (i.e., using Eq. (2.47)) yields

$$f(z/\sigma) = \left(\frac{2z}{n_P\sigma^2/\psi^2}\right)^{(n_P-1)/2} \exp\left(-z - \frac{1}{2}n_P\frac{\sigma^2}{\psi^2}\right) I_{n_P-1}\left(\sqrt{2n_Pz\frac{\sigma^2}{\psi^2}}\right) \tag{2.55}$$

To obtain $f(z)$ use the relations

$$f(z, \sigma) = f(z/\sigma)f(\sigma) \tag{2.56}$$

$$f(z) = \int f(z, \sigma) d\sigma \tag{2.57}$$

Finally, using Eq. (2.56) in Eq. (2.57) produces

$$f(z) = \int f(z/\sigma)f(\sigma) d\sigma \tag{2.58}$$

where $f(z/\sigma)$ is defined in Eq. (2.55) and $f(\sigma)$ is in either Eq. (2.53) or (2.54). The probability of detection is obtained by integrating the *pdf* derived from Eq. (2.58) from the threshold value to infinity. Performing the integration in Eq. (2.58) leads to the incomplete Gamma function.

2.5.1. Threshold Selection

When only a single pulse is used, the detection threshold V_T is related to the probability of false alarm P_{fa} as defined in Eq. (2.19). DiFranco and Rubin[1] derived a general form relating the threshold and P_{fa} for any number of pulses when non-coherent integration is used. It is

$$P_{fa} = 1 - \Gamma_I\left(\frac{V_T}{\sqrt{n_P}}, n_P - 1\right) \tag{2.59}$$

where Γ_I is used to denote the incomplete Gamma function. It is given by

$$\Gamma_I\left(\frac{V_T}{\sqrt{n_P}}, n_P - 1\right) = \int_0^{V_T/\sqrt{n_P}} \frac{e^{-\gamma}\ \gamma^{(n_P - 1) - 1}}{((n_P - 1) - 1)!}\ d\gamma \tag{2.60}$$

Note that the limiting values for the incomplete Gamma function are

$$\Gamma_I(0, N) = 0 \qquad \Gamma_I(\infty, N) = 1 \tag{2.61}$$

For our purposes, the incomplete Gamma function can be approximated by

$$\Gamma_I\left(\frac{V_T}{\sqrt{n_P}}, n_P - 1\right) = 1 - \frac{V_T^{n_P - 1}\ e^{-V_T}}{(n_P - 1)!}\left[1 + \frac{n_P - 1}{V_T} + \frac{(n_P - 1)(n_P - 2)}{V_T^2} + \right. \tag{2.62}$$

$$\left. \dots + \frac{(n_P - 1)!}{V_T^{n_P - 1}}\right]$$

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

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

$$V_{T,m} = V_{T,m-1} - \frac{G(V_{T,m-1})}{G'(V_{T,m-1})} \qquad ; \; m = 1, 2, 3, \ldots \tag{2.63}$$

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

$$G(V_{T,m}) = (0.5)^{n_P/n_{fa}} - \Gamma_I(V_T, n_P) \tag{2.64}$$

$$G'(V_{T,m}) = -\frac{e^{-V_T} \; V_T^{n_P - 1}}{(n_P - 1)!} \tag{2.65}$$

The initial value for the recursion is

$$V_{T,0} = n_P - \sqrt{n_P} + 2.3 \sqrt{-\log P_{fa}} \; (\sqrt{-\log P_{fa}} + \sqrt{n_P} - 1) \tag{2.66}$$

MATLAB Function "incomplete_gamma.m"

In general, the incomplete Gamma function for some integer N is

$$\Gamma_I(x, N) = \int_0^x \frac{e^{-v} \; v^{N-1}}{(N-1)!} dv \tag{2.67}$$

The function *"incomplete_gamma.m"* implements Eq. (2.67). It is given in Listing 2.9 in Section 2.11. Note that this function uses the MATLAB function *"factor.m"* which is given in Listing 2.10. The function *"factor.m"* calculates the factorial of an integer. Fig. 2.7 shows the incomplete Gamma function for $N = 1, 3, 6, 10$. This figure can be reproduced using the MATLAB program *"fig2_7.m"* given in Listing 2.11. The syntax for this function is as follows:

[value] = incomplete_gamma (x, N)

where

Symbol	Description	Units	Status
x	*variable input to* $\Gamma_I(x, N)$	*units of x*	*input*
N	*variable input to* $\Gamma_I(x, N)$	*none / integer*	*input*
value	$\Gamma_I(x, N)$	*none*	*output*

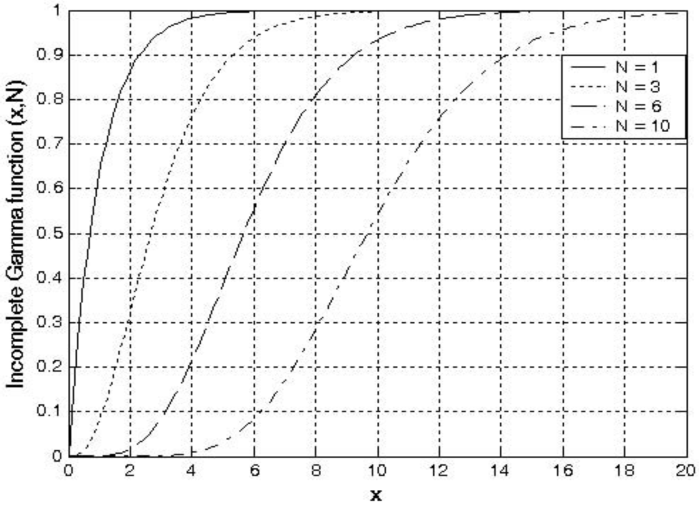


Figure 2.7. The incomplete Gamma function for four values of *N*.

MATLAB Function "threshold.m"

The function *"threshold.m"* calculates the threshold using the recursive formula used in the Newton-Raphson method. It is given in Listing 2.12 in Section 2.11. The syntax is as follows:

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

where

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

Fig. 2.8 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 MATLAB program *"fig2_8.m"* given in Listing 2.13. This program uses both *"threshold.m"* and "*incomplete_gamma*".

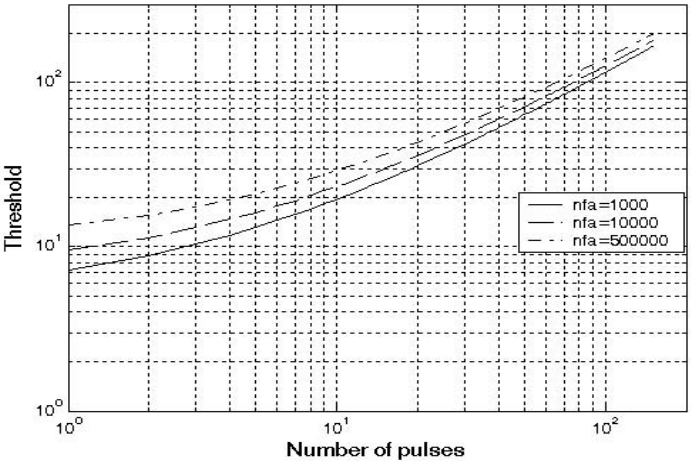


Figure 2.8. Threshold V_T versus n_p for several values of n_{fa}.

2.6. Probability of Detection Calculation

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

$$P_{fa} \approx \ln(2)(n_P/n_{fa}) \tag{2.68}$$

The single pulse probability of detection for non-fluctuating targets is given in Eq. (2.24). When $n_P > 1$, the probability of detection is computed using the Gram-Charlier series. In this case, the probability of detection is

$$P_D \cong \frac{erfc(V/\sqrt{2})}{2} - \frac{e^{-V^2/2}}{\sqrt{2\pi}}[C_3(V^2-1) + C_4V(3-V^2) \\ - C_6V(V^4 - 10V^2 + 15)] \tag{2.69}$$

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

$$V = \frac{V_T - n_P(1 + SNR)}{\varpi} \tag{2.70}$$

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

2.6.1. Detection of Swerling V Targets

For Swerling V (Swerling 0) target fluctuations, the probability of detection is calculated using Eq. (2.69). In this case, the Gram-Charlier series coefficients are

$$C_3 = -\frac{SNR + 1/3}{\sqrt{n_p}(2SNR + 1)^{1.5}} \tag{2.71}$$

$$C_4 = \frac{SNR + 1/4}{n_p(2SNR + 1)^2} \tag{2.72}$$

$$C_6 = C_3^2/2 \tag{2.73}$$

$$\varpi = \sqrt{n_p(2SNR + 1)} \tag{2.74}$$

MATLAB Function "pd_swerling5.m"

The function *"pd_swerling5.m"* calculates the probability of detection for Swerling V targets. It is given in Listing 2.14. 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*
np	*number of integrated pulses*	*none*	*input*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

Fig. 2.9 shows a plot for the probability of detection versus SNR for cases $n_p = 1, 10$. This figure can be reproduced using the MATLAB program *"fig2_9.m"*. It is given in Listing 2.15 in Section 2.11.

Note that it requires less SNR, with ten pulses integrated non-coherently, 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 $I(10) \approx 8.55dB$. Fig. 2.9 shows that the ten pulse SNR is about $6.03dB$. Therefore, the single pulse SNR is about (from Eq. (2.49)) $14.5dB$, which can be read from the figure.

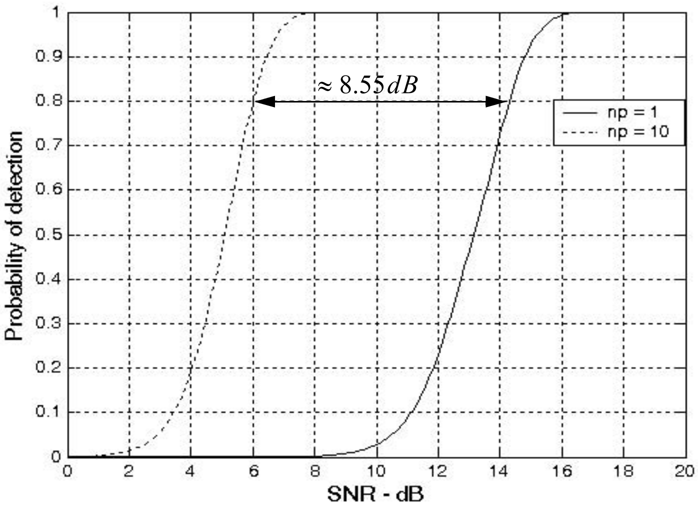


Figure 2.9. Probability of detection versus SNR, $P_{fa} = 10^{-9}$, and non-coherent integration.

2.6.2. Detection of Swerling I Targets

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

$$P_D = e^{-V_T/(1+SNR)} \qquad ; \; n_P = 1 \tag{2.75}$$

$$P_D = 1 - \Gamma_I(V_T, n_P - 1) + \left(1 + \frac{1}{n_P SNR}\right)^{n_p - 1} \Gamma_I\left(\frac{V_T}{1 + \frac{1}{n_P SNR}}, n_P - 1\right) \tag{2.76}$$

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

MATLAB Function "pd_swerling1.m"

The function *"pd_swerling1.m"* calculates the probability of detection for Swerling I type targets. It is given in Listing 2.16 in Section 2.11. 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*

Fig. 2.10 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 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 MATLAB program *"fig2_10.m"* given in Listing 2.17.

Fig. 2.11a shows a plot of the probability of detection versus SNR for $n_P = 1, 10, 50, 100$, where $P_{fa} = 10^{-8}$. Fig. 2.11b is similar to Fig. 2.11a; in this case $P_{fa} = 10^{-11}$. These figures can be reproduced using MATLAB program *"fig2_11ab.m"* given in Listing 2.18.

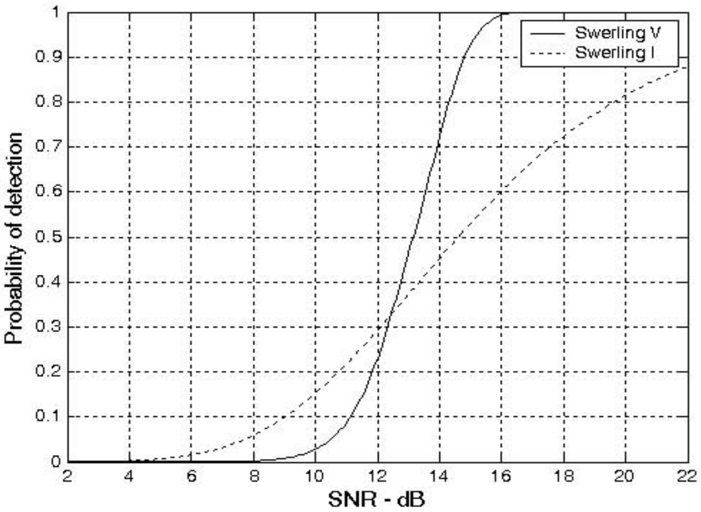


Figure 2.10. Probability of detection versus SNR, single pulse. $P_{fa} = 10^{-9}$**.**

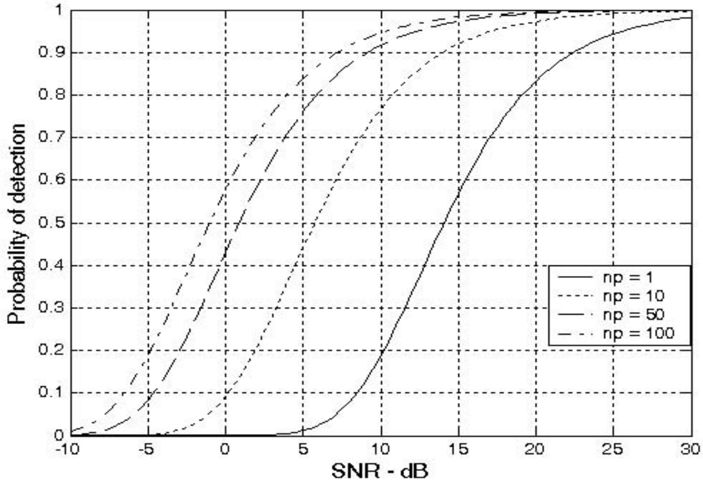


Figure 2.11a. Probability of detection versus SNR. Swerling I. $P_{fa} = 10^{-8}$**.**

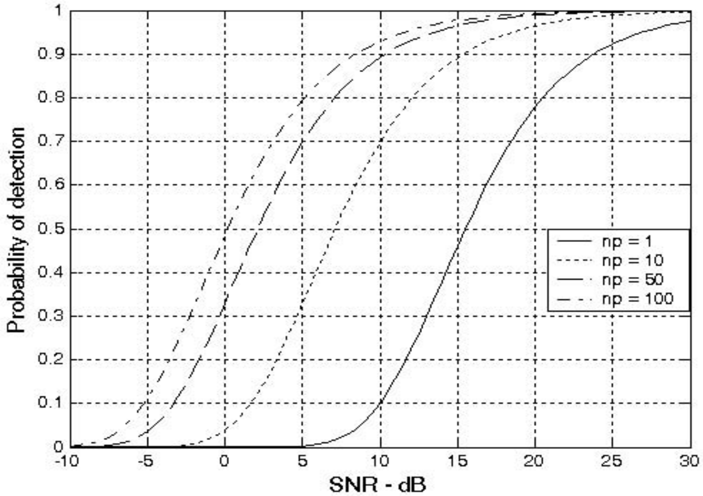


Figure 2.11b. Probability of detection versus SNR. Swerling I. $P_{fa} = 10^{-11}$**.**

2.6.3. Detection of Swerling II Targets

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

$$P_D = 1 - \Gamma_I\left(\frac{V_T}{(1+SNR)}\ ,n_p\right) \qquad ; \ n_P \le 50 \tag{2.77}$$

For the case when $n_P > 50$ Eq. (2.69) is used to compute the probability of detection. In this case,

$$C_3 = -\frac{1}{3\sqrt{n_p}} \qquad , \ C_6 = \frac{C_3^2}{2} \tag{2.78}$$

$$C_4 = \frac{1}{4n_P} \tag{2.79}$$

$$\varpi = \sqrt{n_P}\ (1+SNR) \tag{2.80}$$

MATLAB Function "pd_swerling2.m"

The function *"pd_swerling2.m"* calculates P_D for Swerling II type targets. It is given in Listing 2.19 in Section 2.11. 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*

Fig. 2.12 shows a plot of the probability of detection as a function of SNR for $n_P = 1, 10, 50, 100$, where $P_{fa} = 10^{-10}$. This figure can be reproduced using MATLAB program *"fig2_12.m"* given in Listing 2.20.

2.6.4. Detection of Swerling III Targets

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

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

$$K_0 = 1 + \frac{V_T}{1+n_P SNR/2} - \frac{2}{n_P SNR}(n_P - 2) \tag{2.81}$$

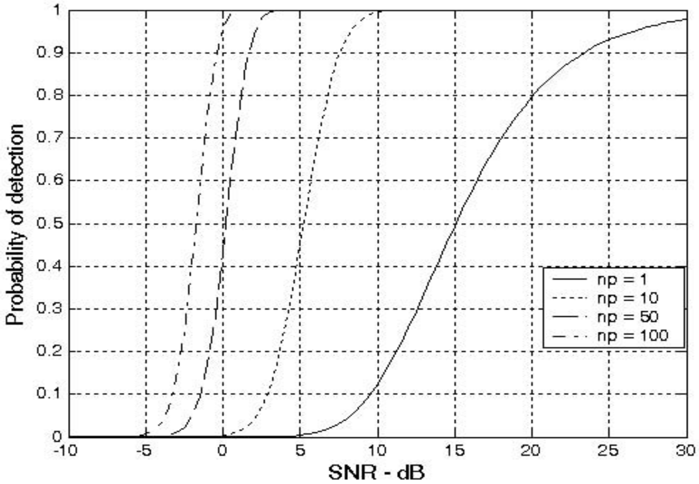


Figure 2.12. Probability of detection versus SNR. Swerling II. $P_{fa} = 10^{-10}$**.**

For $n_P > 2$ the expression is

$$P_D = \frac{V_T^{n_P - 1} \; e^{-V_T}}{(1 + n_P SNR/2)(n_P - 2)!} + 1 - \Gamma_I(V_T, n_P - 1) + K_0 \tag{2.82}$$

$$\times \; \Gamma_I\left(\frac{V_T}{1 + 2/n_p SNR}, n_p - 1\right)$$

MATLAB Function "pd_swerling3.m"

The function *"pd_swerling3.m"* calculates P_D for Swerling III type targets. It is given in Listing 2.21 in Section 2.11. 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*

Fig. 2.13 shows a plot of the probability of detection as a function of SNR for $n_P = 1, 10, 50, 100$, where $P_{fa} = 10^{-9}$. This figure can be reproduced using MATLAB program *"fig2_13.m"* given in Listing 2.22.

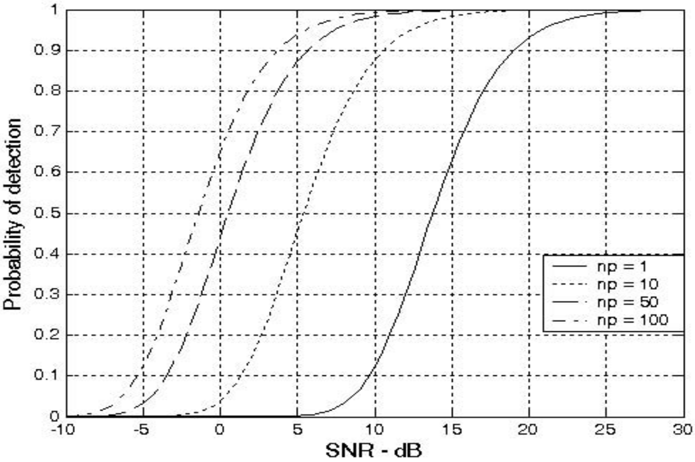


Figure 2.13. Probability of detection versus SNR. Swerling III. $P_{fa} = 10^{-9}$**.**

2.6.5. Detection of Swerling IV Targets

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

$$P_D = 1 - \left[\gamma_0 + \left(\frac{SNR}{2}\right) n_P \gamma_1 + \left(\frac{SNR}{2}\right)^2 \frac{n_P(n_P - 1)}{2!} \gamma_2 + \ldots + \left(\frac{SNR}{2}\right)^{n_P} \gamma_{n_P}\right]\left(1 + \frac{SNR}{2}\right)^{-n_P} \tag{2.83}$$

where

$$\gamma_i = \Gamma_I\left(\frac{V_T}{1 + (SNR)/2}, n_P + i\right) \tag{2.84}$$

By using the recursive formula

$$\Gamma_I(x, i + 1) = \Gamma_I(x, i) - \frac{x^i}{i! \exp(x)} \tag{2.85}$$

then only γ_0 needs to be calculated using Eq. (2.84) and the rest of γ_i are calculated from the following recursion:

$$\gamma_i = \gamma_{i-1} - A_i \qquad ; \; i > 0 \tag{2.86}$$

$$A_i = \frac{V_T/(1+(SNR)/2)}{n_P + i - 1} A_{i-1} \qquad ; \; i > 1 \tag{2.87}$$

$$A_1 = \frac{(V_T/(1+(SNR)/2))^{n_P}}{n_P! \exp(V_T/(1+(SNR)/2))} \tag{2.88}$$

$$\gamma_0 = \Gamma_I\left(\frac{V_T}{(1+(SNR)/2)}, n_P\right) \tag{2.89}$$

For the case when $n_P \geq 50$, the Gram-Charlier series and Eq. (2.69) can be used to calculate the probability of detection. In this case,

$$C_3 = \frac{1}{3\sqrt{n_P}} \frac{2\beta^3 - 1}{(2\beta^2 - 1)^{1.5}} \qquad ; \; C_6 = \frac{C_3^2}{2} \tag{2.90}$$

$$C_4 = \frac{1}{4n_P} \frac{2\beta^4 - 1}{(2\beta^2 - 1)^2} \tag{2.91}$$

$$\varpi = \sqrt{n_P(2\beta^2 - 1)} \tag{2.92}$$

$$\beta = 1 + \frac{SNR}{2} \tag{2.93}$$

MATLAB Function "pd_swerling4.m"

The function *"pd_swerling4.m"* calculates P_D for Swerling IV type targets. It is given in Listing 2.23 in Section 2.11. 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*
snr	*SNR*	*dB*	*input*
pd	*probability of detection*	*none*	*output*

Figure 2.14 shows a plot of the probability of detection as a function of SNR for $n_P = 1, 10, 50, 100$, where $P_{fa} = 10^{-9}$. This figure can be reproduced using MATLAB program *"fig2_14.m"* given in Listing 2.24.

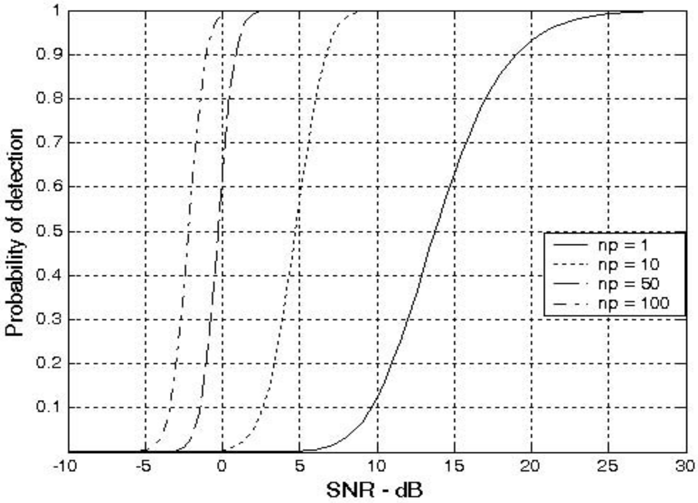


Figure 2.14. Probability of detection versus SNR. Swerling IV. $P_{fa} = 10^{-9}$.

2.7. The Radar Equation Revisited

The radar equation developed in Chapter 1 assumed a constant target RCS and did not account for integration loss. In this section, a more comprehensive form of the radar equation is introduced. In this case, the radar equation is given by

$$R^4 = \frac{P_{av} G_t G_r \lambda^2 \sigma I(n_P)}{(4\pi)^3 k T_e F B \tau f_r L_t L_f \ (SNR)_1} \tag{2.94}$$

where $P_{av} = P_t \tau f_r$ is the average transmitted power, P_t is the peak transmitted power, τ is pulsewidth, f_r is PRF, G_t is transmitting antenna gain, G_r is receiving antenna gain, λ is wavelength, σ is target cross section, $I(n_P)$ is improvement factor, n_P is the number of integrated pulses, k is Boltzman's constant, T_e is effective noise temperature, F is the system noise figure, B is receiver bandwidth, L_t is total system losses including integration loss, L_f is loss due to target fluctuation, and $(SNR)_1$ is the minimum single pulse SNR required for detection.

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. This was demonstrated for a Swerling I fluctuation in Fig.

2.10. Kanter[1] developed an exact analysis for calculating the fluctuation loss. In this text the authors 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 with an accuracy of $0.005dB$ or better. For this purpose the MATLAB function *"fluct_loss.m"* was developed. It is given in Listing 2.25 in Section 2.11. Its syntax is as follows:

[Lf, Pd_Sw5] = fluct_loss(pd, pfa, np, sw_case)

where

Symbol	Description	Units	Status
pd	*desired probability of detection*	*none*	*input*
pfa	*probability of false alarm*	*none*	*input*
np	*number of pulses*	*none*	*input*
sw_case	*1, 2, 3, or 4 depending on the desired Swerling case*	*none*	*input*
Lf	*fluctuation loss*	*dB*	*output*
Pd_Sw5	*Probability of detection corresponding to a Swerling V case*	*none*	*output*

For example, using the syntax

[Lf,Pd_Sw5]=fluct_loss(0.65, 1e-9, 10,1)

will calculate the SNR corresponding to both Swerling V and Swerling I fluctuation when the desired probability of detection $P_D = 0.65$ and probability of false alarm $P_{fa} = 10^{-9}$ and 10 pulses of non-coherent integration. The following is a reprint of the output:

```
PD_SW5 = 0.65096989459928
SNR_SW5 = 5.52499999999990
PD_SW1 = 0.65019653294095
SNR_SW1 = 8.32999999999990
Lf = 2.80500000000000
```

Note that a negative value for L_f indicates a fluctuation SNR gain instead of loss. Finally, it must be noted that the function *"fluct_loss.m"* always assumes non-coherent integration. Fig. 2.15 shows a plot for the additional SNR (or fluctuation loss) required to achieve a certain probability of detection. This figure can be reproduced using MATLAB program *"fig2_16.m"* given in Listing 2.26 in Section 2.11.

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

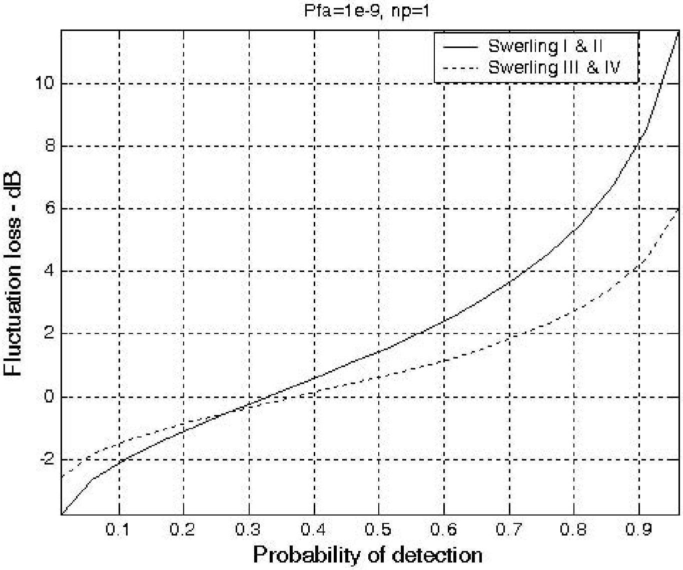


Figure 2.15. Fluctuation loss versus probability of detection.

2.8. 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

$$(SNR)_{R_0} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_0 BFLR_0^4} = 1 \tag{2.95}$$

The single pulse SNR at any range R is

$$SNR = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_0 BFLR^4} \tag{2.96}$$

Dividing Eq. (2.96) by Eq. (2.95) yields

$$\frac{SNR}{(SNR)_{R_0}} = \left(\frac{R_0}{R}\right)^4 \tag{2.97}$$

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

$$(SNR)_{dB} = 40\log\left(\frac{R_0}{R}\right) \tag{2.98}$$

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. 2.16) is given by

$$P_{C_n} = 1 - \prod_{i=1}^{n}(1 - P_{D_i}) \tag{2.99}$$

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.

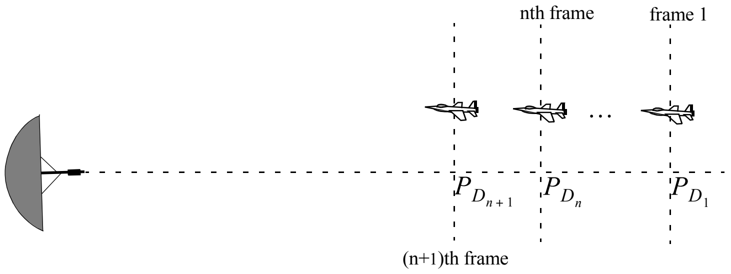


Figure 2.16. Detecting a target in many frames.

2.8.1. Mini Design Case Study 2.2

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 $R = 8Km$?

A 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. (2.98), 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
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. 2.17.*

The cumulative probability of detection is given in Eq. (2.95), 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* $R = 8Km$. *The cumulative probability of detection at 8 Km is then*

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

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

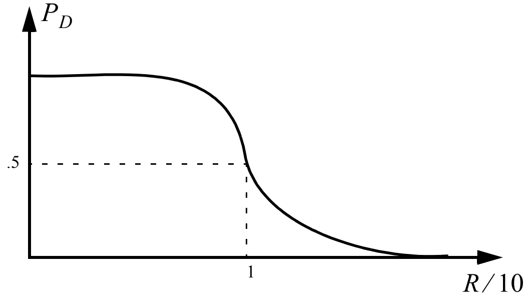


Figure 2.17. Cumulative probability of detection versus normalized range.

2.9. Constant False Alarm Rate (CFAR)

The detection threshold is computed so that the radar receiver maintains a constant pre-determined probability of false alarm. Eq. (2.19b) 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. (2.100):

$$V_T = \sqrt{2\psi^2 \ln\left(\frac{1}{P_{fa}}\right)} \tag{2.100}$$

If the noise power ψ^2 is assumed to be constant, then a fixed threshold can satisfy Eq. (2.100). 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].

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

2.9.1. Cell-Averaging CFAR (Single Pulse)

The CA-CFAR processor is shown in Fig. 2.18. 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

$$Y_1 \geq K_0 Z \tag{2.101}$$

Cell-averaging 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

$$f(z) = \frac{z^{(M/2)-1} e^{(-z/2\psi^2)}}{2^{M/2} \quad \psi^M \Gamma(M/2)} \qquad ; \ z > 0 \tag{2.102}$$

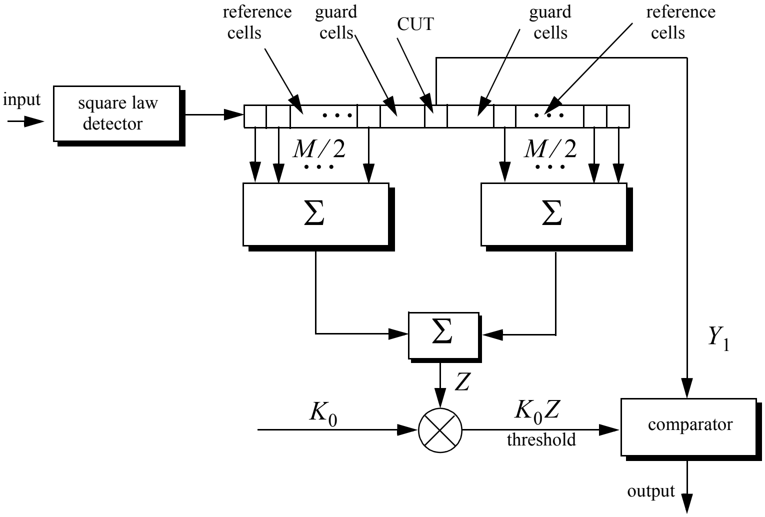


Figure 2.18. Conventional CA-CFAR.

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

$$P_{fa}(V_T = y) = e^{-y/2\psi^2} \tag{2.103}$$

It follows that the unconditional probability of false alarm is

$$P_{fa} = \int_0^\infty P_{fa}(V_T = y) f(y) dy \tag{2.104}$$

where $f(y)$ is the *pdf* of the threshold, which except for the constant K_0 is the same as that defined in Eq. (2.102). Therefore,

$$f(y) = \frac{y^{M-1} e^{(-y/2K_0\psi^2)}}{(2K_0\psi^2)^M \Gamma(M)} \quad ; \; y \ge 0 \tag{2.105}$$

Performing the integration in Eq. (2.104) yields

$$P_{fa} = \frac{1}{(1+K_0)^M} \tag{2.106}$$

Observation of Eq. (2.106) shows that the probability of false alarm is now independent of the noise power, which is the objective of CFAR processing.

2.9.2. Cell-Averaging CFAR with Non-Coherent Integration

In practice, CFAR averaging is often implemented after non-coherent integration, as illustrated in Fig. 2.19. 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.

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

$$P_{fa} = Prob\{Y_1/Z > K_1\} \tag{2.107}$$

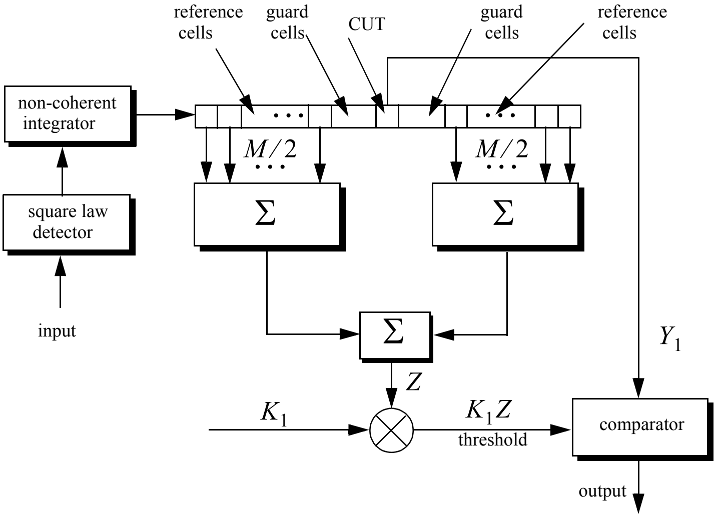


Figure 2.19. Conventional CA-CFAR with non-coherent integration.

Eq. (2.107) 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 given by

$$P_{fa} = \int_0^{\infty} P_{fa}(V_T = y) f(y) dy \qquad \textbf{(2.108)}$$

where $f(y)$ is the *pdf* of the threshold. In view of this, the probability density function describing the random variable K_1Z is given by

$$f(y) = \frac{(y/K_1)^{Mn_P - 1} e^{(-y/2K_0\psi^2)}}{(2\psi^2)^{Mn_P} K_1 \ \Gamma(Mn_P)} \quad ; \ y \geq 0 \qquad \textbf{(2.109)}$$

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

$$P_{fa} = \frac{1}{(1+K_1)^{Mn_P}} \sum_{k=0}^{n_P - 1} \frac{1}{k!} \frac{\Gamma(Mn_P + k)}{\Gamma(Mn_P)} \left(\frac{K_1}{1+K_1}\right)^k \qquad \textbf{(2.110)}$$

which is identical to Eq. (2.106) when $K_1 = K_0$ and $n_P = 1$.

2.10. "MyRadar" Design Case Study - Visit 2[1]

2.10.1. Problem Statement

Modify the design introduced in Chapter 1 for the "MyRadar" design case study so that the effects of target RCS fluctuations are taken into account. For this purpose modify the design such that: The aircraft and missile target types follow Swerling I and Swerling III fluctuations, respectively. Also assume that a $P_D \geq 0.995$ *is required at maximum range with* $P_{fa} = 10^{-7}$ *or better. You may use either non-coherent integration or cumulative probability of detection. Also, modify any other design parameters if needed.*

2.10.2. A Design

The missile and the aircraft detection ranges were calculated in Chapter 1. They are $R_a = 90Km$ for the aircraft and $R_m = 55Km$ for the missile. First, determine the probability of detection for each target type with and without the 7-pulse non-coherent integration. For this purpose, use MATLAB program *"myradar_visit2_1.m"* given in Listing 2.27. This program first computes the improvement factor and the associated integration loss. Second it calculates the single pulse SNR. Finally it calculates the SNR when non-coherent integration is utilized. Executing this program yields:

SNR_single_pulse_missile = 5.5998 dB
SNR_7_pulse_NCI_missile = 11.7216 dB
SNR_single_pulse_aircraft = 6.0755 dB
SNR_7_pulse_NCI_aircrfat = 12.1973 dB

Using these values in functions *"pd_swerling1.m"* and *"pd_swerling3.m"* yields

Pd_single_pulse_missile = 0.013
Pd_7_pulse_NCI_missile= 0.9276
Pd_single_pulse_aircraft = 0.038
Pd_7_pulse_NCI_aircraft = 0.8273

Clearly in all four cases, there is not enough SNR to meet the design requirement of $P_D \geq 0.995$.

[1] Please read disclaimer in Section 1.9.1.

Instead, resort to accomplishing the desired probability of detection by using cumulative probabilities. The single frame increment for the missile and aircraft cases are

$$R_{Missile} = scan\ \ rate \times v_m = 2 \times 150 = 300m \tag{2.111}$$

$$R_{Aircraft} = scan\ \ rate \times v_a = 2 \times 400 = 800m \tag{2.112}$$

2.10.2.1 Single Pulse (Per Frame) Design Option

As a first design option, consider the case where during each frame only a single pulse is used for detection (i.e., no integration). Consequently, if the single pulse detection does not achieve the desired probability of detection at 90 Km for the aircraft or at 55 Km for the missile, then non-coherent integration of a few pulses per frame can then be utilized. Keep in mind that only non-coherent integration can be used in the cases of Swerling type I and III fluctuations (see Section 2.4).

Assume that the first frame corresponding to detecting the aircraft is 106 Km. This assumption is arbitrary and it provides the designer with 21 frames. It follows that the first frame, when detecting the missile, is at 61 Km. Furthermore, assume that the SNR at $R = 90Km$ is $(SNR)_{aircraft} = 8.5dB$, for the aircraft case. And, for the missile case assume that at $R = 55Km$ the corresponding SNR is $(SNR)_{missile} = 9dB$. Note that these values are simply educated guesses, and the designer may be required to perform several iterations in order to accomplish the desired cumulative probability of detection, $P_D \geq 0.995$. In order to calculate the cumulative probability of detection at a certain range, the MATLAB program *"myradar_visit2_2.m"* was developed. This program is given in Listing 2.28 in Section 2.11.

Initialization of the program *"myradar_visit2_2.m"* includes entering the following inputs: The desired P_{fa}; the number of pulses to be used for non-coherent integration per frame; the range at which the desired cumulative operability of detection must be achieved; the frame size; and finally the target fluctuation type. For notational purposes, denote the range at which the desired cumulative probability of detection must be achieved as R_0. Then for each frame, the following list includes the outputs of this program: SNR, probability of detection, fluctuation loss, and cumulative probability of detection.

The logic used by this program for calculating the proper probability of detection at each frame and for computing the cumulative probability of detection is described as follows:

1. Initialize the program, by entering the desired input values. Assume Swerling V fluctuation and use Eq. (2.98) to calculate the frame-SNR, $(SNR)_i$.

1.1. For the *"MyRadar"* design case study, use $n_P = 1$, $R_0 = 90Km$, and $(SNR_0)_{aircraft} = 8.5dB$. Alternatively use $R_0 = 55Km$ and $(SNR)_{missile} = 9dB$ for the missile case. Note that the selected SNR values are best estimates or educated guesses, and it may require going through few iterations before finally selecting an acceptable set.

2. The program will then calculate the number of frames and their associated ranges. The program uses the function *"fluct_loss.m"* to calculate the Swerling V P_D at each frame and the additional SNR required to accomplish the same probability of detection when target fluctuation is included.

3. Depending on the fluctuation type, the program will then use the proper MATLAB function to calculate the probability of detection for each frame, P_{D_i}.

3.1. For the *"MyRadar"* design case study, these functions are *"pd_swerling1.m"* and *"pd_swerling 3.m"*.

4. Finally, the program uses Eq. (2.99) to compute the cumulative probability of detection, P_{D_n}.

A Graphical User Interface (GUI) has been developed for this program; Fig. 2.20 shows its associated GUI workspace. To use this GUI, from the MATLAB command window type *"myradar_visit2_2_gui"*. Executing the program "*myradar_visit2_2.m"* using the input values stated above yields the following cumulative probabilities of detection for the aircraft and missile cases,

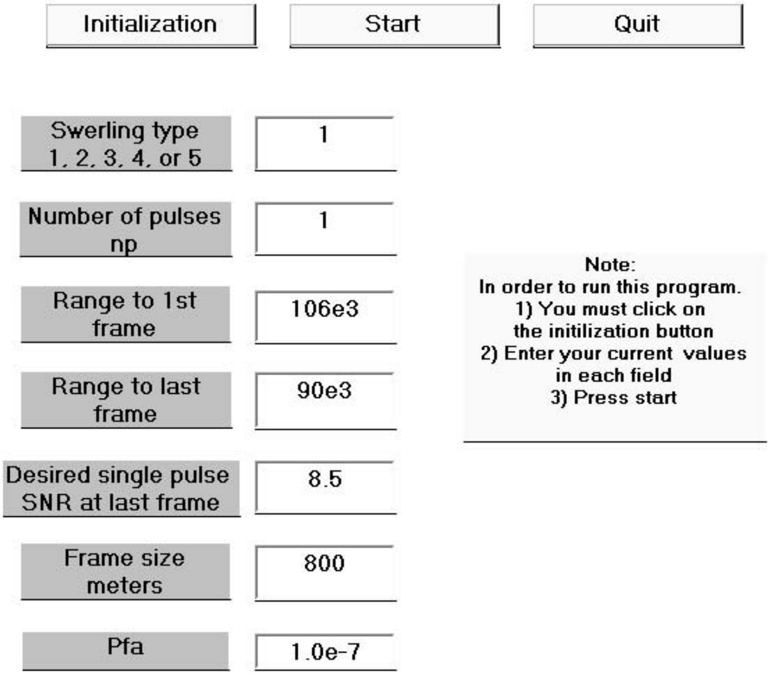
$$P_{DC_{Missile}} = 0.99872$$

$$P_{DC_{aircraft}} = 0.99687$$

These results clearly satisfy the design requirement of $P_D \ge 0.995$. However, one must re-validate the peak power requirement for the design. To do that, go back to Eq.s (1.107) and (1.108), and replace the SNR values used in Chapter 1 by the values adopted in this chapter (i.e., $(SNR_0)_{aircraft} = 8.5dB$ and $(SNR)_{missile} = 9dB$). It follows that the corresponding single pulse energy for the missile and the aircraft cases are respectively given by

$$E_m = 0.1658 \times \frac{10^{0.9}}{10^{0.56}} = 0.36273 Joules \qquad \textbf{(2.113)}$$

$$E_a = 0.1487 \times \frac{10^{0.85}}{10^{0.56}} = 0.28994 Joules \qquad \textbf{(2.114)}$$



Swerling type 1, 2, 3, 4, or 5: 1

Number of pulses np: 1

Range to 1st frame: 106e3

Range to last frame: 90e3

Desired single pulse SNR at last frame: 8.5

Frame size meters: 800

Pfa: 1.0e-7

Note:
In order to run this program.
1) You must click on
the initilization button
2) Enter your current values
in each field
3) Press start

Figure 2.20. GUI workspace associated with program *"myradar_visit2_2_gui.m"*.

This indicates that the stressing single pulse peak power requirement (i.e., missile detection) exceeds $362KW$. This value for the single pulse peak power is high for a mobile ground based air defense radar and practical constraints would require using less peak power.

In order to bring the single pulse peak power requirement down, one can use non-coherent integration of a few pulses per frame prior to calculating the frame probability of detection. For this purpose, the program "*myradar_visit2_2.m"* can be used again. However, in this case $n_P > 1$. This is analyzed in the next section.

2.10.2.2. Non-Coherent Integration Design Option

The single frame probability of detection can be improved significantly when pulse integration is utilized. One may use coherent or non-coherent integration to improve the frame cumulative probability of detection. In this case, caution should be exercised since coherent integration would not be practical when the target fluctuation type is either Swerling I or Swerling III. Alternatively, using non-coherent integration will always reduce the minimum required SNR.

Rerun the MATLAB program "*myradar_visit2_2_gui*". Use $n_P = 4$ and use $SNR = 4dB$ (single pulse) for both the missile and aircraft single pulse SNR[1] at their respective reference ranges, $R_{0_{missile}} = 55Km$ and $R_{0_{aircraft}} = 90Km$. The resulting cumulative probabilities of detection are

$$P_{DC_{Missile}} = 0.99945$$

$$P_{DC_{aircraft}} = 0.99812$$

which are both within the desired design requirements. It follows that the corresponding minimum required single pulse energy for the missile and the aircraft cases are now given by

$$E_m = 0.1658 \times \frac{10^{0.4}}{10^{0.56}} = 0.1147 Joules \tag{2.115}$$

$$E_a = 0.1487 \times \frac{10^{0.4}}{10^{0.56}} = 0.1029 Joules \tag{2.116}$$

Thus, the minimum single pulse peak power (assuming the same pulsewidth as that given in Section1.9.2) is

$$P_t = \frac{0.1147}{1 \times 10^{-6}} = 114.7KW \tag{2.117}$$

Note that the peak power requirement will be significantly reduced while maintaining a very fine range resolution when pulse compression techniques are used. This will be discussed in a subsequent chapter.

Fig. 2.21 shows a plot of the SNR versus range for both target types. This plot assumes 4-pulse non-coherent integration. It can be reproduced using MATLAB program *"fig2_21.m"*. It is given in Listing 2.29 in Section 2.11.

2.11. MATLAB Program and Function Listings

This section presents listings for all MATLAB programs/functions used in this chapter. The user is advised to rerun these programs with different input parameters.

[1] Again these values are educated guesses. The designer my be required to go through a few iterations before arriving at an acceptable set of design parameters.

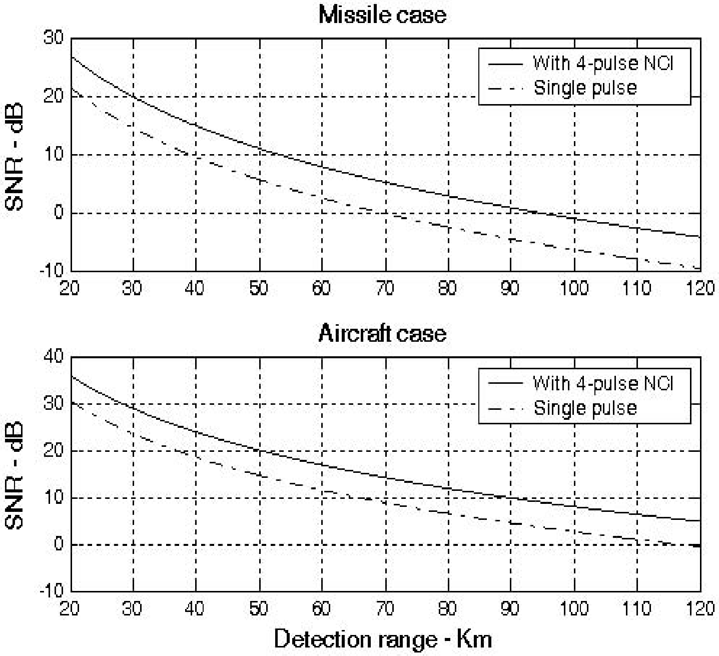


Figure 2.21. SNR versus detection range for both target types. The 4-pulse NCI curves correspond to 21 frame cumulative detection with the last frame at: 55 Km for the missile and 90 Km for the aircraft.

Listing 2.1. MATLAB Program "fig2_2.m"

```
% This program can be used to reproduce Figure 2.2 of the text
clear all
close all
xg = linspace(-6,6,1500); % random variable between -6 and 6
xr = linspace(0,6,1500); % random variable between 0 and 6
mu = 0; % zero mean Gaussian pdf mean
sigma = 1.5; % standard deviation (sqrt(variance))
ynorm = normpdf(xg,mu,sigma); % use MATLAB function normpdf
yray = raylpdf(xr,sigma); % use MATLAB function raylpdf
plot(xg,ynorm,'k',xr,yray,'k-.');
grid
legend('Gaussian pdf','Rayleigh pdf')
xlabel('x')
```

```
ylabel('Probability density')
gtext('\mu = 0; \sigma = 1.5')
gtext('\sigma =1.5')
```

Listing 2.2. MATLAB Function "que_func.m"

```
function fofx = que_func(x)
% This function computes the value of the Q-function
% listed in Eq.(2.16). It uses the approximation in Eqs. (2.17) and (2.18)
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
```

Listing 2.3. MATLAB Program "fig2_3.m"

```
%This program generates Figure 2.3.
close all
clear all
logpfa = linspace(.01,250,1000);
var = 10.^(logpfa ./ 10.0);
vtnorm =  sqrt( log (var));
semilogx(logpfa, vtnorm,'k')
grid
```

Listing 2.4. MATLAB Function "marcumsq.m"

```
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
```

Listing 2.5. MATLAB Program “prob_snr1.m”

```
% This program is used to produce Fig. 2.4
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 - dB')
ylabel ('Probability of detection')
grid
```

Listing 2.6. MATLAB program “fig2_6a.m”

```
% This program is used to produce Fig. 2.6a
% It uses the function "improv_fac"
clear all
close all
pfa1 = 1.0e-2;
pfa2 = 1.0e-6;
pfa3 = 1.0e-10;
pfa4 = 1.0e-13;
pd1 = .5;
pd2 = .8;
pd3 = .95;
pd4 = .999;
index = 0;
for np = 1:1:1000
   index = index + 1;
   I1(index) = improv_fac (np, pfa1, pd1);
   I2(index) = improv_fac (np, pfa2, pd2);
   I3(index) = improv_fac (np, pfa3, pd3);
   I4(index) = improv_fac (np, pfa4, pd4);
end
np = 1:1:1000;
semilogx (np, I1, 'k', np, I2, 'k--', np, I3, 'k-.', np, I4, 'k:')
xlabel ('Number of pulses');
ylabel ('Improvement factor I - dB')
legend ('pd=.5, nfa=e+2','pd=.8, nfa=e+6','pd=.95, nfa=e+10','pd=.999, nfa=e+13');
grid
```

Listing 2.7. MATLAB Function “improv_fac.m”

```
function impr_of_np = improv_fac (np, pfa, pd)
% This function computes the non-coherent integration improvement
% factor using the empirical formula defined in Eq. (2.49)
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
```

Listing 2.8. MATLAB Program “fig2_6b.m”

```
% This program is used to produce Fig. 2.6b
% It uses the function "improv_fac".
clear all
close all
pfa1 = 1.0e-12;
pfa2 = 1.0e-12;
pfa3 = 1.0e-12;
pfa4 = 1.0e-12;
pd1 = .5;
pd2 = .8;
pd3 = .95;
pd4 = .99;
index = 0;
for np = 1:1:1000
   index = index+1;
   I1 = improv_fac (np, pfa1, pd1);
   i1 = 10.^(0.1*I1);
   L1(index) = -1*10*log10(i1 ./ np);
   I2 = improv_fac (np, pfa2, pd2);
   i2 = 10.^(0.1*I2);
   L2(index) = -1*10*log10(i2 ./ np);
   I3 = improv_fac (np, pfa3, pd3);
   i3 = 10.^(0.1*I3);
   L3(index) = -1*10*log10(i3 ./ np);
   I4 = improv_fac (np, pfa4, pd4);
   i4 = 10.^(0.1*I4);
   L4 (index) = -1*10*log10(i4 ./ np);
end
np = 1:1:1000;
semilogx (np, L1, 'k', np, L2, 'k--', np, L3, 'k-.', np, L4, 'k:')
axis tight
xlabel ('Number of pulses');
ylabel ('Integration loss - dB')
legend ('pd=.5, nfa=e+12','pd=.8, nfa=e+12','pd=.95, nfa=e+12','pd=.99,
nfa=e+12');
grid
```

Listing 2.9. MATLAB Function “incomplete_gamma.m”

```
function [value] = incomplete_gamma ( vt, np)
% This function implements Eq. (2.67) to compute the Incomplete Gamma
Function
% This function needs "factor.m" to run
format long
eps = 1.000000001;
% Test to see if np = 1
if (np == 1)
  value1 = vt * exp(-vt);
  value = 1.0 - exp(-vt);
  return
end
sumold = 1.0;
sumnew =1.0;
calc1 = 1.0;
calc2 = np;
xx = np * log(vt+0.0000000001) - vt - factor(calc2);
temp1 = exp(xx);
temp2 = np / (vt+0.0000000001);
diff = .0;
ratio = 1000.0;
if (vt >= np)
  while (ratio >= eps)
    diff = diff + 1.0;
    calc1 = calc1 * (calc2 - diff) / vt ;
    sumnew = sumold + calc1;
    ratio = sumnew / sumold;
    sumold = sumnew;
  end
  value = 1.0 - temp1 * sumnew * temp2;
  return
else
  diff = 0.;
  sumold = 1.;
  ratio = 1000.;
  calc1 = 1.;
  while(ratio >= eps)
    diff = diff + 1.0;
    calc1 = calc1 * vt / (calc2 + diff);
    sumnew = sumold + calc1;
    ratio = sumnew / sumold;
    sumold = sumnew;
```

```
   end
   value = temp1 * sumnew;
end
```

Listing 2.10. MATLAB Function “factor.m”

```
function [val] = factor(n)
% Compute the factorial of n using logarithms to avoid overflow.
format long
n = n + 9.0;
n2 = n * n;
temp = (n-1) * log(n) - n + log(sqrt(2.0 * pi * n)) ...
   + ((1.0 - (1.0/30. + (1.0/105)/n2)/n2) / 12) / n;
val = temp - log((n-1)*(n-2)*(n-3)*(n-4)*(n-5)*(n-6) ...
   *(n-7)*(n-8));
return
```

Listing 2.11. MATLAB Program “fig2_7.m”

```
% This program can be used to reproduce Fig. 2.7
close all
clear all
format long
ii = 0;
for x = 0:.1:20
   ii = ii+1;
   val1(ii) = incomplete_gamma(x , 1);
   val2(ii) = incomplete_gamma(x , 3);
   val = incomplete_gamma(x , 6);
   val3(ii) = val;
   val = incomplete_gamma(x , 10);
   val4(ii) = val;
end
xx = 0:.1:20;
plot(xx,val1,'k',xx,val2,'k:',xx,val3,'k--',xx,val4,'k-.')
legend('N = 1','N = 3','N = 6','N = 10')
xlabel('x')
ylabel('Incomplete Gamma function (x,N)')
grid
```

Listing 2.12. MATLAB Function “threshold.m”

```
function [pfa, vt] = threshold (nfa, np)
% This function calculates the threshold value from nfa and np.
% The Newton-Raphson recursive formula is used (Eqs. (2-63) through (2-66))
% This function uses "incomplete_gamma.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 = 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
```

Listing 2.13. MATLAB Program “fig2_8.m”

```
% Use this program to reproduce Fig. 2.8 of text
clear all
for n= 1: 1:150
   [pfa1 y1(n)] = threshold(1000,n);
   [pfa2 y3(n)] = threshold(10000,n);
   [pfa3 y4(n)] = threshold(500000,n);
end
n =1:1:150;
loglog(n,y1,'k',n,y3,'k--',n,y4,'k-.');
axis([0 200 1 300])
xlabel ('Number of pulses');
ylabel('Threshold')
legend('nfa=1000','nfa=10000','nfa=500000')
grid
```

Listing 2.14. MATLAB Function “pd_swerling5.m”

```
function pd = pd_swerling5 (input1, indicator, np, snrbar)
% This function is used to calculate the probability of
% 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-Charlier 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;
```

Listing 2.15. MATLAB Program “fig2_9.m”

```
% This program is used to produce Fig. 2.9
close all
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
b = sqrt(-2.0 * log(pfa));
index = 0;
for snr = 0:.1:20
   index = index +1;
   a = sqrt(2.0 * 10^(.1*snr));
   pro(index) = marcumsq(a,b);
   prob205(index) =  pd_swerling5 (pfa, 1, 10, snr);
end
x = 0:.1:20;
plot(x, pro,'k',x,prob205,'k:');
axis([0 20 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10')
grid
```

Listing 2.16. MATLAB Function “pd_swerling1.m”

```
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 (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
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 = incomplete_gamma(vt,np-1);
  igf2 = incomplete_gamma(vt/temp,np-1);
  pd = 1.0 - igf1 + val1 * igf2 * exp(-vt/temp1);
```

Listing 2.17. MATLAB Program “fig2_10.m”

```
% This program is used to reproduce Fig. 2.10
close all
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
b = sqrt(-2.0 * log(pfa));
index = 0;
for snr = 0:.1:22
  index = index +1;
  a = sqrt(2.0 * 10^(.1*snr));
  pro(index) = marcumsq(a,b);
  prob(index) =  pd_swerling1 (nfa, 1, snr);
end
x = 0:.1:22;
plot(x, pro,'k',x,prob,'k:');
axis([2 22 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('Swerling V','Swerling I')
grid
```

Listing 2.18. MATLAB Program “fig2_11ab.m”

```
% This program is used to produce Fig. 2.11a&b
clear all
pfa = 1e-11;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
   index = index +1;
   prob1(index) =  pd_swerling1 (nfa, 1, snr);
   prob10(index) =  pd_swerling1 (nfa, 10, snr);
   prob50(index) =  pd_swerling1 (nfa, 50, snr);
   prob100(index) =  pd_swerling1 (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 - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

Listing 2.19. MATLAB Function “pd_swerling2.m”

```
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 (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
if (np <= 50)
   temp = vt / (1.0 + snrbar);
   pd = 1.0 - incomplete_gamma(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
```

Listing 2.20. MATLAB Program "fig2_12.m"

```
% This program is used to produce Fig. 2.12
clear all
pfa = 1e-10;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
   index = index +1;
   prob1(index) =  pd_swerling2 (nfa, 1, snr);
   prob10(index) =  pd_swerling2 (nfa, 10, snr);
   prob50(index) =  pd_swerling2 (nfa, 50, snr);
   prob100(index) =  pd_swerling2 (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 - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

Listing 2.21. MATLAB Function "pd_swerling3.m"

```
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 (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
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
   temp4 = vt^(np-1.) * exp(-vt) / (temp1 * exp(factor(np-2.)));
   temp5 =  vt / (1.0 + 2.0 / (np *snrbar));
   pd = temp4 + 1.0 - incomplete_gamma(vt,np-1.) + ko * ...
      incomplete_gamma(temp5,np-1.);
end
```

Listing 2.22. MATLAB Program "fig2_13.m"

```
% This program is used to produce Fig. 2.13
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 - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

Listing 2.23. MATLAB Function “pd_swerling4.m”

```
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 (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
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
  snr = 1.0;
  gamma0 = incomplete_gamma(vt/beta,np);
  a1 = (vt / beta)^np / (exp(factor(np)) * exp(vt/beta));
  sum = gamma0;
  for i = 1:1:np
    temp1 = 1;
    if (i == 1)
      ai = a1;
    else
      ai = (vt / beta) * a1 / (np + i -1);
    end
    a1 = ai;
    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 / exp(factor(i));
    sum = sum + term;
  end
  pd = 1.0 - sum / beta^np;
end
pd = max(pd,0.);
```

Listing 2.24. MATLAB Program "fig2_14.m"

```
% This program is used to produce Fig. 2.14
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
  prob1(index) =  pd_swerling4 (nfa, 1, snr);
  prob10(index) =  pd_swerling4 (nfa, 10, snr);
  prob50(index) =  pd_swerling4(nfa, 50, snr);
  prob100(index) =  pd_swerling4 (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 1.1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
axis tight
```

Listing 2.25. MATLAB Function "fluct_loss.m"

```
function [Lf,Pd_Sw5] = fluct_loss(pd, pfa, 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
% compute the false alarm number
nfa =  log(2) / pfa;
% *************** Swerling 5 case ****************
% check to make sure that np>1
if (np ==1)
   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_SW5 = snr_inc
else
   % np > 1 use MATLAB function pd_swerling5.m
   snr_inc = 0.1 - 0.005;
   Pd_Sw5 = 0.001;
   while(Pd_Sw5 <= pd)
      snr_inc = snr_inc + 0.005;
      Pd_Sw5 = pd_swerling5(pfa, 1, np, snr_inc);
   end
   PD_SW5 = Pd_Sw5
   SNR_SW5 = snr_inc
end
if sw_case == 5
   Lf = 0.
   return
end
% ************** End Swerling 5 case ***********
% ************** Swerling 1 case ***************
if (sw_case == 1)
   Pd_Sw1 = 0.001;
   snr_inc = 0.1 - 0.005;
   while(Pd_Sw1 <= pd)
      snr_inc = snr_inc + 0.005;
      Pd_Sw1 = pd_swerling1(nfa, np, snr_inc);
   end
   PD_SW1 = Pd_Sw1
   SNR_SW1 = snr_inc
   Lf = SNR_SW1 - SNR_SW5
end
% ************** End Swerling 1 case ***********
% ************** Swerling 2 case ***************
if (sw_case == 2)
   Pd_Sw2 = 0.001;
   snr_inc = 0.1 - 0.005;
   while(Pd_Sw2 <= pd)
      snr_inc = snr_inc + 0.005;
      Pd_Sw2 = pd_swerling2(nfa, np, snr_inc);
   end
   PD_SW2 = Pd_Sw2
   SNR_SW2 = snr_inc
   Lf = SNR_SW2 - SNR_SW5
end
% ************** End Swerling 2 case ***********
% ************** Swerling 3 case ***************
```

```
if (sw_case == 3)
   Pd_Sw3 = 0.001;
   snr_inc = 0.1 - 0.005;
   while(Pd_Sw3 <= pd)
      snr_inc = snr_inc + 0.005;
      Pd_Sw3 = pd_swerling3(nfa, np, snr_inc);
   end
   PD_SW3 = Pd_Sw3
   SNR_SW3 = snr_inc
   Lf = SNR_SW3 - SNR_SW5
end
% *************** End Swerling 3 case ************
% *************** Swerling 4 case ****************
if (sw_case == 4)
   Pd_Sw4 = 0.001;
   snr_inc = 0.1 - 0.005;
   while(Pd_Sw4 <= pd)
      snr_inc = snr_inc + 0.005;
      Pd_Sw4 = pd_swerling4(nfa, np, snr_inc);
   end
   PD_SW4 = Pd_Sw4
   SNR_SW4 = snr_inc
   Lf = SNR_SW4 - SNR_SW5
end
% *************** End Swerling 4 case ************
return
```

Listing 2.26. MATLAB Program “fig2_15.m”

```
% Use this program to reproduce Fig. 2.15 of text
clear all
close all
index = 0.;
for pd = 0.01:.05:1
   index = index + 1;
   [Lf,Pd_Sw5] = fluct_loss(pd, 1e-9,1,1);
   Lf1(index) = Lf;
   [Lf,Pd_Sw5] = fluct_loss(pd, 1e-9,1,4);
   Lf4(index) = Lf;
end
pd = 0.01:.05:1;
figure (2)
plot(pd, Lf1, 'k',pd, Lf4,'K:')
xlabel('Probability of detection')
```

```
ylabel('Fluctuation loss - dB')
legend('Swerling I & II','Swerling III & IV')
title('Pfa=1e-9, np=1')
grid
```

Listing 2.27. MATLAB Program “myradar_visit2_1.m”

```
% Myradar design case study visit 2_1
close all
clear all
pfa = 1e-7;
pd = 0.995;
np = 7;
pt = 165.8e3; % peak power in Watts
freq = 3e+9; % radar operating frequency in Hz
g = 34.5139; % antenna gain in dB
sigmam = 0.5; % missile RCS m squared
sigmaa = 4; % aircraft RCS m squared
te = 290.0; % effective noise temperature in Kelvins
b = 1.0e+6; % radar operating bandwidth in Hz
nf = 6.0; %noise figure in dB
loss = 8.0; % radar losses in dB
% compute the improvement factor due to 7-pulse non-coherent integration
Improv = improv_fac (np, pfa, pd);
% calculate the integration loss
lossnci = 10*log10(np) - Improv;
% calculate net gain in SNR due to integration
SNR_net = Improv - lossnci;
loss_total = loss + lossnci;
rangem = 55e3;
rangea = 90e3;
SNR_single_pulse_missile = radar_eq(pt, freq, g, sigmam, te, b, nf, loss, rangem)
SNR_7_pulse_NCI_missile = SNR_single_pulse_missile + SNR_net
SNR_single_pulse_aircraft = radar_eq(pt, freq, g, sigmaa, te, b, nf, loss, rangea)
SNR_7_pulse_NCI_aircraft = SNR_single_pulse_aircraft + SNR_net
```

Listing 2.28. MATLAB Program “myradar_visit2_2.m”

```
%clear all
% close all
% swid = 3;
% pfa = 1e-7;
% np = 1;
% R_1st_frame = 61e3; % Range for first frame
% R0 = 55e3; % range to last frame
% SNR0 = 9; % SNR at R0
% frame = 0.3e3; % frame size
nfa = log(2) / pfa;
range_frame = R_1st_frame:-frame:R0; % Range to each frame
% implement Eq. (2.98)
SNRi = SNR0 + 40 .* log10((R0 ./ range_frame));
% calculate the Swerling 5 Pd at each frame
b = sqrt(-2.0 * log(pfa));
if np ==1
   for frame = 1:1:size(SNRi,2)
      a = sqrt(2.0 * 10^(.1*SNRi(frame)));
      pd5(frame) = marcumsq(a,b);
   end
else
   [pd5] = pd_swerling5(pfa, 1, np, SNRi);
end
% compute additional SNR needed due to fluctuation
for frame = 1:1:size(SNRi,2)
   [Lf(frame),Pd_Sw5] = fluct_loss(pd5(frame), pfa, np, swid);
end
% adjust SNR at each frame
SNRi = SNRi - Lf;
% compute the frame Pd
for frame = 1:1:size(SNRi,2)
   if(swid==1)
      Pdi(frame) = pd_swerling1 (nfa, np, SNRi(frame));
   end
    if(swid==2)
      Pdi(frame) = pd_swerling2 (nfa, np, SNRi(frame));
   end
   if(swid==3)
      Pdi(frame) = pd_swerling3 (nfa, np, SNRi(frame));
   end
   if(swid==4)
      Pdi(frame) = pd_swerling4 (nfa, np, SNRi(frame));
   end
    if(swid==5)
      Pdi(frame) = pd5(frame);
   end
end
```

```
Pdc(1:size(SNRi,2)) = 0;
Pdc(1) = 1 - Pdi(1);
% compute the cumulative Pd
for frame = 2:1:size(SNRi,2)
   Pdc(frame) = (1-Pdi(frame)) * Pdc(frame-1);
end
PDC = 1 - Pdc(21)
```

Listing 2.29. MATLAB Program “fig2_21.m”

```
% Use this program to reproduce Fig. 2.20 of text.
close all
clear all
np = 4;
pfa = 1e-7;
pdm = 0.99945;
pda = 0.99812;
% calculate the improvement factor
Im = improv_fac(np,pfa, pdm);
Ia = improv_fac(np, pfa, pda);
% caculate the integration loss
Lm = 10*log10(np) - Im;
La = 10*log10(np) - Ia;
pt = 114.7e3; % peak power in Watts
freq = 3e+9; % radar operating frequency in Hz
g = 34.5139; % antenna gain in dB
sigmam = 0.5; % missile RCS m squared
sigmaa = 4; % aircraft RCS m squared
te = 290.0; % effective noise temperature in Kelvins
b = 1.0e+6; % radar operating bandwidth in Hz
nf = 6.0; % noise figure in dB
loss = 8.0; % radar losses in dB
losstm = loss + Lm; % total loss for missile
lossta = loss + La; % total loss for aircraft
range = linspace(20e3,120e3,1000); % range to target from 20 to 120 Km,
1000 points
% modify pt by np*pt to account for pulse integration
snrmnci = radar_eq(np*pt, freq, g, sigmam, te, b, nf, losstm, range);
snrm = radar_eq(pt, freq, g, sigmam, te, b, nf, loss, range);
snranci = radar_eq(np*pt, freq, g, sigmaa, te, b, nf, lossta, range);
snra = radar_eq(pt, freq, g, sigmaa, te, b, nf, loss, range);
% plot SNR versus range
rangekm  = range ./ 1000;
figure(1)
```

```
subplot(2,1,1)
plot(rangekm,snrmnci,'k',rangekm,snrm,'k -.')
grid
legend('With 4-pulse NCI','Single pulse')
ylabel ('SNR - dB');
title('Missile case')
subplot(2,1,2)
plot(rangekm,snranci,'k',rangekm,snra,'k -.')
grid
legend('With 4-pulse NCI','Single pulse')
ylabel ('SNR - dB');
title('Aircraft case')
xlabel('Detection range - Km')
```

Chapter 3 Radar Waveforms

Choosing a particular waveform type and a signal processing technique in a radar system depends heavily on the radar's specific mission and role. The cost and complexity associated with a certain type of waveform hardware and software implementation constitute a major factor in the decision process. Radar systems can use Continuous Waveforms (CW) or pulsed waveforms with or without modulation. Modulation techniques can be either analog or digital. Range and Doppler resolutions are directly related to the specific waveform frequency characteristics. Thus, knowledge of the power spectrum density of a waveform is very critical. In general, signals or waveforms can be analyzed using time domain or frequency domain techniques. This chapter introduces many of the most commonly used radar waveforms. Relevant uses of a specific waveform will be addressed in the context of its time and frequency domain characteristics. In this book, the terms waveform and signal are used interchangeably to mean the same thing.

3.1. Low Pass, Band Pass Signals, and Quadrature Components

Signals that contain significant frequency composition at a low frequency band including DC are called Low Pass (LP) signals. Signals that have significant frequency composition around some frequency away from the origin are called Band Pass (BP) signals. A real BP signal $x(t)$ can be represented mathematically by

$$x(t) = r(t)\cos(2\pi f_0 t + \phi_x(t)) \qquad \textbf{(3.1)}$$

where $r(t)$ is the amplitude modulation or envelope, $\phi_x(t)$ is the phase modulation, f_0 is the carrier frequency, and both $r(t)$ and $\phi_x(t)$ have frequency components significantly smaller than f_0. The frequency modulation is

$$f_m(t) = \frac{1}{2\pi}\frac{d}{dt}\phi_x(t) \tag{3.2}$$

and the instantaneous frequency is

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}(2\pi f_0 t + \phi_x(t)) = f_0 + f_m(t) \tag{3.3}$$

If the signal bandwidth is B, and if f_0 is very large compared to B, the signal $x(t)$ is referred to as a narrow band pass signal.

Band pass signals can also be represented by two low pass signals known as the quadrature components; in this case Eq. (3.1) can be rewritten as

$$x(t) = x_I(t)\cos 2\pi f_0 t - x_Q(t)\sin 2\pi f_0 t \tag{3.4}$$

where $x_I(t)$ and $x_Q(t)$ are real LP signals referred to as the quadrature components and are given, respectively, by

$$\begin{aligned} x_I(t) &= r(t)\cos\phi_x(t) \\ x_Q(t) &= r(t)\sin\phi_x(t) \end{aligned} \tag{3.5}$$

Fig. 3.1 shows how the quadrature components are extracted.

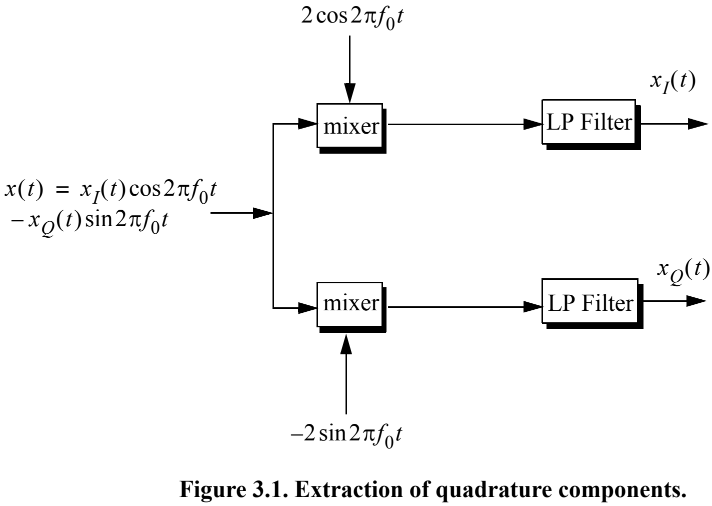


Figure 3.1. Extraction of quadrature components.

3.2. The Analytic Signal

The sinusoidal signal $x(t)$ defined in Eq. (3.1) can be written as the real part of the complex signal $\psi(t)$. More precisely,

$$x(t)= Re\{\psi(t)\} = Re\{r(t)e^{j\phi_x(t)}e^{j2\pi f_0 t}\} \tag{3.6}$$

Define the "analytic signal" as

$$\psi(t) = v(t)e^{j2\pi f_0 t} \tag{3.7}$$

where

$$v(t) = r(t)e^{j\phi_x(t)} \tag{3.8}$$

and

$$\Psi(\omega) = \begin{cases} 2X(\omega) & \omega \ge 0 \\ 0 & \omega < 0 \end{cases} \tag{3.9}$$

$\Psi(\omega)$ is the Fourier transform of $\psi(t)$ and $X(\omega)$ is the Fourier transform of $x(t)$. Eq. (3.9) can be written as

$$\Psi(\omega) = 2U(\omega)X(\omega) \tag{3.10}$$

where $U(\omega)$ is the step function in the frequency domain. Thus, it can be shown that $\psi(t)$ is

$$\psi(t) = x(t) + j\tilde{x}(t) \tag{3.11}$$

$\tilde{x}(t)$ is the Hilbert transform of $x(t)$.

Using Eqs. (3.6) and (3.11), one can then write (shown here without proof)

$$x(t) = u_{0I}(t)\cos\omega_0 t - u_{0Q}(t)\sin\omega_0 t \tag{3.12}$$

which is similar to Eq. (3.4) with $\omega_0 = 2\pi f_0$.

Using Parseval's theorem it can be shown that the energy associated with the signal $x(t)$ is

$$E_x = \frac{1}{2}\int_{-\infty}^{\infty} x^2(t)\ dt = \frac{1}{2}\int_{-\infty}^{\infty} u^2(t)\ dt = \frac{1}{2}E_\psi \tag{3.13}$$

3.3. CW and Pulsed Waveforms

The spectrum of a given signal describes the spread of its energy in the frequency domain. An energy signal (finite energy) can be characterized by its Energy Spectrum Density (ESD) function, while a power signal (finite power) is characterized by the Power Spectrum Density (PSD) function. The units of the ESD are Joules per Hertz and the PSD has units Watts per Hertz.

The signal bandwidth is the range of frequency over which the signal has a nonzero spectrum. In general, any signal can be defined using its duration (time domain) and bandwidth (frequency domain). A signal is said to be band-limited if it has finite bandwidth. Signals that have finite durations (time-limited) will have infinite bandwidths, while band-limited signals have infinite durations. The extreme case is a continuous sine wave, whose bandwidth is infinitesimal.

A time domain signal $f(t)$ has a Fourier Transform (FT) $F(\omega)$ given by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\ dt \tag{3.14}$$

where the Inverse Fourier Transform (IFT) is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}\ d\omega \tag{3.15}$$

The signal autocorrelation function $R_f(\tau)$ is

$$R_f(\tau) = \int_{-\infty}^{\infty} f^*(t)f(t+\tau)\ dt \tag{3.16}$$

The asterisk indicates the complex conjugate. The signal amplitude spectrum is $|F(\omega)|$. If $f(t)$ were an energy signal, then its ESD is $|F(\omega)|^2$; and if it were a power signal, then its PSD is $\bar{S}_f(\omega)$ which is the FT of the autocorrelation function

$$\bar{S}_f(\omega) = \int_{-\infty}^{\infty} \bar{R}_f(\tau)e^{-j\omega\tau}\ d\tau \tag{3.17}$$

First, consider a CW waveform given by

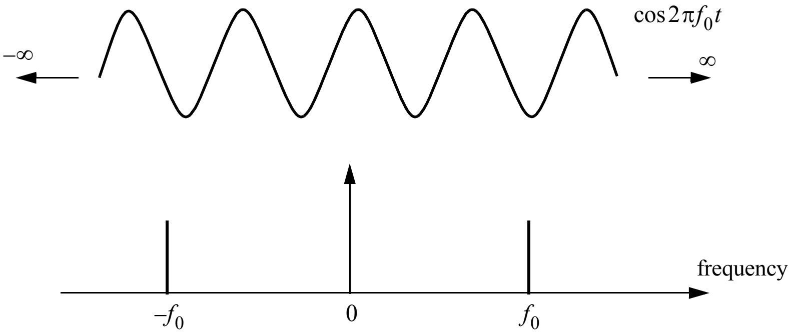


Figure 3.2. Amplitude spectrum for a continuous sine wave.

$$f_1(t) = A\cos\omega_0 t \tag{3.18}$$

The FT of $f_1(t)$ is

$$F_1(\omega) = A\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] \tag{3.19}$$

where $\delta(\;\cdot\;)$ is the Dirac delta function, and $\omega_0 = 2\pi f_0$. As indicated by the amplitude spectrum shown in Fig. 3.2, the signal $f_1(t)$ has infinitesimal bandwidth, located at $\pm f_0$.

Next consider the time domain signal $f_2(t)$ given by

$$f_2(t) = ARect\left(\frac{t}{\tau}\right) = \begin{Bmatrix} A & -\frac{\tau}{2} \le t \le \frac{\tau}{2} \\ 0 & otherwise \end{Bmatrix} \tag{3.20}$$

It follows that the FT is

$$F_2(\omega) = A\tau Sinc\left(\frac{\omega\tau}{2}\right) \tag{3.21}$$

where

$$Sinc(x) = \frac{\sin(\pi x)}{\pi x} \tag{3.22}$$

The amplitude spectrum of $f_2(t)$ is shown in Fig. 3.3. In this case, the bandwidth is infinite. Since infinite bandwidths cannot be physically implemented, the signal bandwidth is approximated by $2\pi/\tau$ radians per second or $1/\tau$

Hertz. In practice, this approximation is widely accepted since it accounts for most of the signal energy.

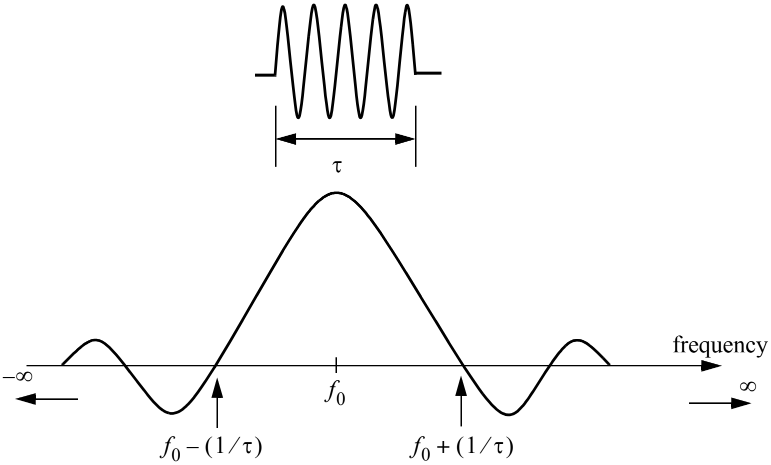


Figure 3.3. Amplitude spectrum for a single pulse, or a train of non-coherent pulses.

Now consider the coherent gated CW waveform $f_3(t)$ given by

$$f_3(t) = \sum_{n=-\infty}^{\infty} f_2(t - nT) \tag{3.23}$$

Clearly $f_3(t)$ is periodic, where T is the period (recall that $f_r = 1/T$ is the PRF). Using the complex exponential Fourier series we can rewrite $f_3(t)$ as

$$f_3(t) = \sum_{n=-\infty}^{\infty} F_n e^{\frac{j2\pi nt}{T}} \tag{3.24}$$

where the Fourier series coefficients F_n are given by

$$F_n = \frac{A\tau}{T}\ Sinc\left(\frac{n\tau\pi}{T}\right) \tag{3.25}$$

It follows that the FT of $f_3(t)$ is

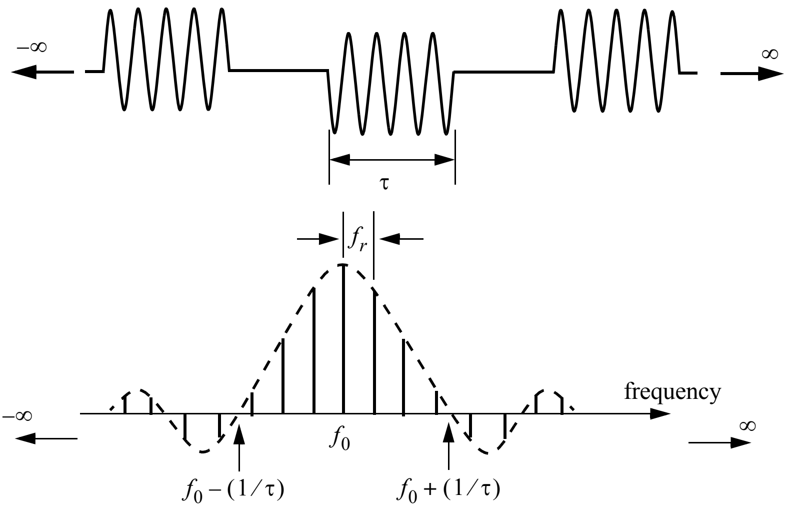


Figure 3.4. Amplitude spectrum for a coherent pulse train of infinite length.

$$F_3(\omega) = 2\pi \sum_{n=-\infty}^{\infty} F_n \delta(\omega - 2n\pi f_r) \tag{3.26}$$

The amplitude spectrum of $f_3(t)$ is shown in Fig. 3.4. In this case, the spectrum has a $\sin x/x$ envelope that corresponds to F_n. The spacing between the spectral lines is equal to the radar PRF, f_r.

Finally, define the function $f_4(t)$ as

$$f_4(t) = \sum_{n=0}^{N} f_2(t - nT) \tag{3.27}$$

Note that $f_4(t)$ is a limited duration of $f_3(t)$. The FT of $f_4(t)$ is

$$F_4(\omega) = AN\tau \left(Sinc\left(\omega \frac{NT}{2}\right) \bullet \sum_{n=-\infty}^{\infty} Sinc(n\pi\tau f_r)\delta(\omega - 2n\pi f_r) \right) \tag{3.28}$$

where the operator $(\ \bullet\ )$ indicates convolution. The spectrum in this case is shown in Fig. 3.5. The envelope is still a $\sin x/x$ which corresponds to the pulsewidth. But the spectral lines are replaced by $\sin x/x$ spectra that correspond to the duration NT.

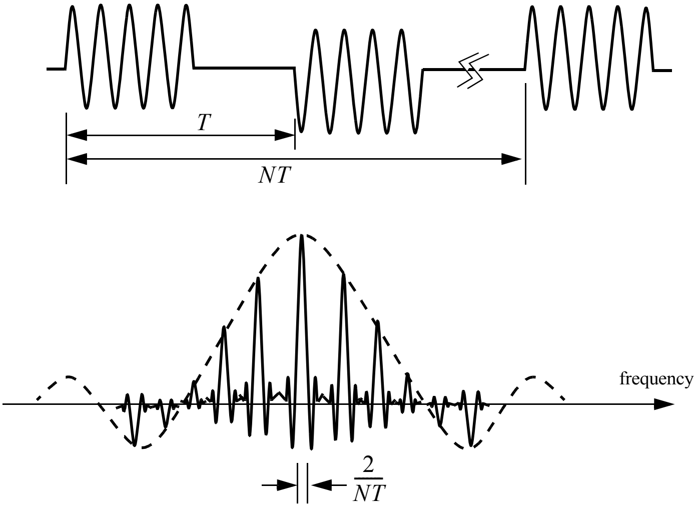


Figure 3.5. Amplitude spectrum for a coherent pulse train of finite length.

3.4. Linear Frequency Modulation Waveforms

Frequency or phase modulated waveforms can be used to achieve much wider operating bandwidths. Linear Frequency Modulation (LFM) is commonly used. In this case, the frequency is swept linearly across the pulsewidth, either upward (up-chirp) or downward (down-chirp). The matched filter bandwidth is proportional to the sweep bandwidth, and is independent of the pulsewidth. Fig. 3.6 shows a typical example of an LFM waveform. The pulsewidth is τ, and the bandwidth is B.

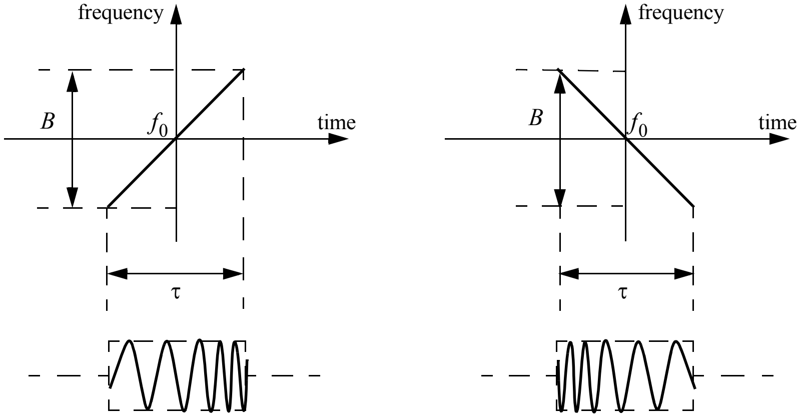
The LFM up-chirp instantaneous phase can be expressed by

$$\psi(t) = 2\pi\left(f_0 t + \frac{\mu}{2}t^2\right) \qquad -\frac{\tau}{2} \le t \le \frac{\tau}{2} \tag{3.29}$$

where f_0 is the radar center frequency, and $\mu = (2\pi B)/\tau$ is the LFM coefficient. Thus, the instantaneous frequency is

$$f(t) = \frac{1}{2\pi}\frac{d}{dt}\psi(t) = f_0 + \mu t \qquad -\frac{\tau}{2} \le t \le \frac{\tau}{2} \tag{3.30}$$

Similarly, the down-chirp instantaneous phase and frequency are given, respectively, by



(a)

(b)

Figure 3.6. Typical LFM waveforms. (a) up-chirp; (b) down-chirp.

$$\psi(t) = 2\pi\left(f_0 t - \frac{\mu}{2}t^2\right) \qquad -\frac{\tau}{2} \le t \le \frac{\tau}{2} \tag{3.31}$$

$$f(t) = \frac{1}{2\pi}\frac{d}{dt}\psi(t) = f_0 - \mu t \qquad -\frac{\tau}{2} \le t \le \frac{\tau}{2} \tag{3.32}$$

A typical LFM waveform can be expressed by

$$s_1(t) = Rect\left(\frac{t}{\tau}\right)e^{j2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)} \tag{3.33}$$

where $Rect(t/\tau)$ denotes a rectangular pulse of width τ. Eq. (3.33) is then written as

$$s_1(t) = e^{j2\pi f_0 t}s(t) \tag{3.34}$$

where

$$s(t) = Rect\left(\frac{t}{\tau}\right)e^{j\pi\mu t^2} \tag{3.35}$$

is the complex envelope of $s_1(t)$.

The spectrum of the signal $s_1(t)$ is determined from its complex envelope $s(t)$. The complex exponential term in Eq. (3.34) introduces a frequency shift about the center frequency f_o. Taking the FT of $s(t)$ yields

$$S(\omega) = \int_{-\infty}^{\infty} Rect\left(\frac{t}{\tau}\right) e^{j\pi\mu t^2} \; e^{-j\omega t} dt = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \exp\left(\frac{j2\pi\mu t^2}{2}\right) \; e^{-j\omega t} dt \qquad \textbf{(3.36)}$$

Let $\mu' = 2\pi\mu = 2\pi B/\tau$, and perform the change of variable

$$x = \sqrt{\frac{\mu'}{\pi}}\left(t - \frac{\omega}{\mu'}\right) \qquad ; \; dx = \sqrt{\frac{\mu'}{\pi}} \; dt \qquad \textbf{(3.37)}$$

Thus, Eq. (3.36) can be written as

$$S(\omega) = \sqrt{\frac{\pi}{\mu'}} \; e^{-j\omega^2/2\mu'} \int_{-x_1}^{x_2} e^{j\pi x^2/2} \; dx \qquad \textbf{(3.38)}$$

$$S(\omega) = \sqrt{\frac{\pi}{\mu'}} \; e^{-j\omega^2/2\mu'} \left\{ \int_0^{x_2} e^{j\pi x^2/2} \; dx - \int_0^{-x_1} e^{j\pi x^2/2} \; dx \right\} \qquad \textbf{(3.39)}$$

where

$$x_1 = \sqrt{\frac{\mu'}{\pi}}\left(\frac{\tau}{2} + \frac{\omega}{\mu'}\right) = \sqrt{\frac{B\tau}{2}}\left(1 + \frac{f}{B/2}\right) \qquad \textbf{(3.40)}$$

$$x_2 = \sqrt{\frac{\mu'}{\pi}}\left(\frac{\tau}{2} - \frac{\omega}{\mu'}\right) = \sqrt{\frac{B\tau}{2}}\left(1 - \frac{f}{B/2}\right) \qquad \textbf{(3.41)}$$

The Fresnel integrals, denoted by $C(x)$ and $S(x)$, are defined by

$$C(x) = \int_0^x \cos\left(\frac{\pi\upsilon^2}{2}\right) d\upsilon \qquad \textbf{(3.42)}$$

$$S(x) = \int_0^x \sin\left(\frac{\pi\upsilon^2}{2}\right) d\upsilon \qquad \textbf{(3.43)}$$

Fresnel integrals are approximated by

$$C(x) \approx \frac{1}{2} + \frac{1}{\pi x}\sin\left(\frac{\pi}{2}x^2\right) \qquad ; \; x \gg 1 \tag{3.44}$$

$$S(x) \approx \frac{1}{2} - \frac{1}{\pi x}\cos\left(\frac{\pi}{2}x^2\right) \qquad ; \; x \gg 1 \tag{3.45}$$

Note that $C(-x) = -C(x)$ and $S(-x) = -S(x)$. Fig. 3.7 shows a plot of both $C(x)$ and $S(x)$ for $0 \le x \le 4.0$. This figure can be reproduced using MATLAB program *"fig3_7.m"* given in Listing 3.1 in Section 3.12.

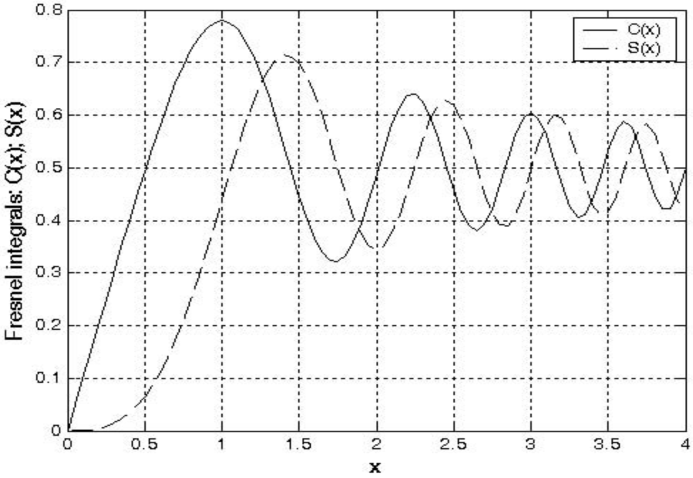


Figure 3.7. Fresnel integrals.

Using Eqs. (3.42) and (3.43) into (3.39) and performing the integration yield

$$S(\omega) = \tau\sqrt{\frac{1}{B\tau}}\; e^{-j\omega^2/(4\pi B)}\left\{\frac{[C(x_2) + C(x_1)] + j[S(x_2) + S(x_1)]}{\sqrt{2}}\right\} \tag{3.46}$$

Fig. 3.8 shows typical plots for the LFM real part, imaginary part, and amplitude spectrum. The square-like spectrum shown in Fig. 3.8c is widely known as the Fresnel spectrum. This figure can be reproduced using MATLAB program *"fig3_8.m"*, given in Listing 3.2 in Section 3.12.

A MATLAB GUI (see Fig. 3.8d) was developed to input LFM data and display outputs as shown in Fig. 3.8. It is called *"LFM_gui.m"*. Its inputs are the uncompressed pulsewidth and the chirp bandwidth.

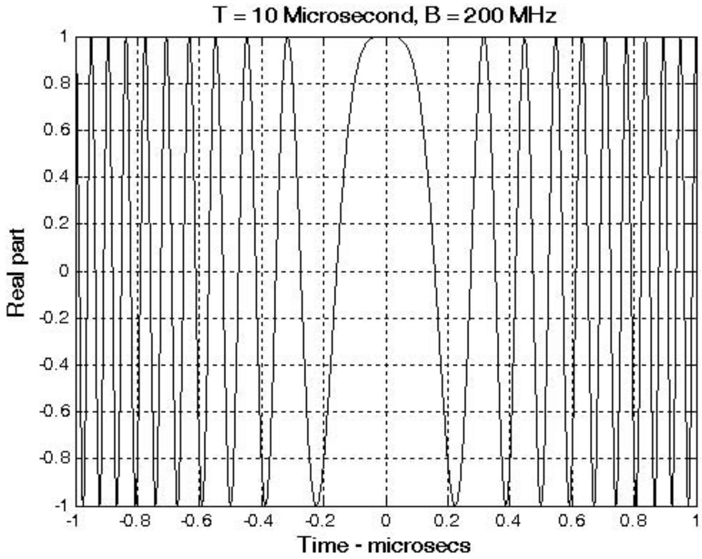


Figure 3.8a. Typical LFM waveform, real part.

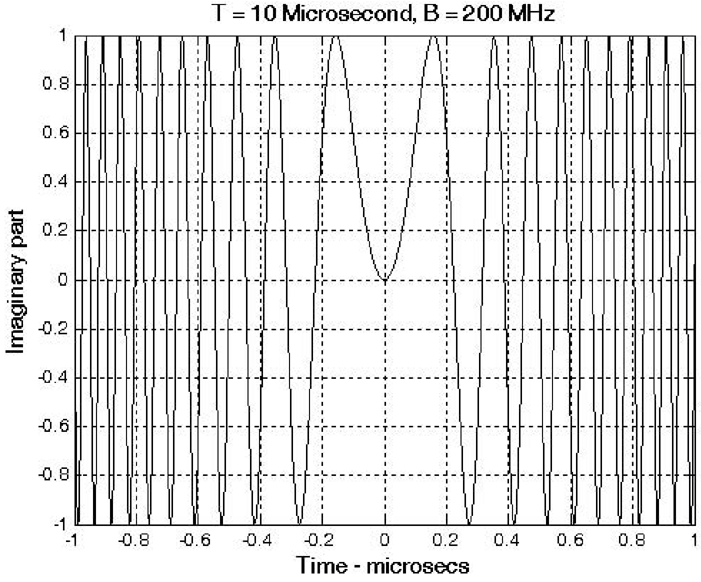


Figure 3.8b. Typical LFM waveform, imaginary part.

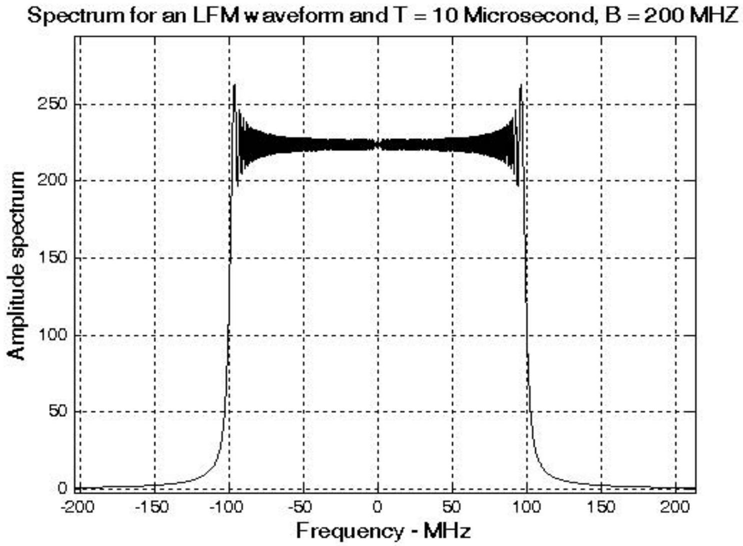
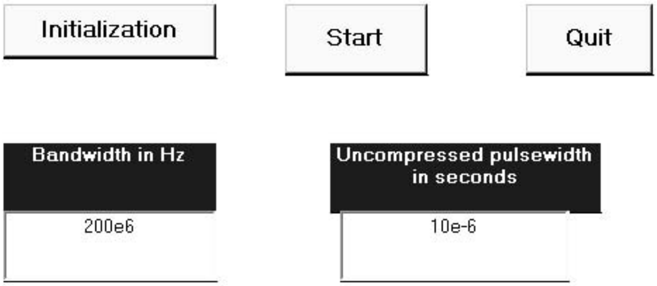


Figure 3.8c. Typical spectrum for an LFM waveform.



Bandwidth in Hz
200e6

Uncompressed pulsewidth in seconds
10e-6

Figure 3.8d. GUI workspace *"LFM_gui.m"*.

3.5. High Range Resolution

An expression for range resolution ΔR in terms of the pulsewidth τ was derived in Chapter 1. When pulse compression is not used, the instantaneous bandwidth B of radar receiver is normally matched to the pulse bandwidth, and in most radar applications this is done by setting $B = 1/\tau$. Therefore, range resolution is given by

$$\Delta R = (c\tau)/2 = c/(2B) \tag{3.47}$$

Radar users and designers alike seek to accomplish High Range Resolution (HRR) by minimizing ΔR. However, as suggested by Eq. (3.47) in order to achieve HRR one must use very short pulses and consequently reduce the average transmitted power and impose severe operating bandwidth requirements. Achieving fine range resolution while maintaining adequate average transmitted power can be accomplished by using pulse compression techniques, which will be discussed in Chapter 5. By means of frequency or phase modulation, pulse compression allows us to achieve the average transmitted power of a relatively long pulse, while obtaining the range resolution corresponding to a very short pulse. As an example, consider an LFM waveform whose bandwidth is B and un-compressed pulsewidth (transmitted) is τ. After pulse compression the compressed pulsewidth is denoted by τ', where $\tau' \ll \tau$, and the HRR is

$$\Delta R = \frac{c\tau'}{2} \ll \frac{c\tau}{2} \tag{3.48}$$

Linear frequency modulation and Frequency-Modulated (FM) CW waveforms are commonly used to achieve HRR. High range resolution can also be synthesized using a class of waveforms known as the “Stepped Frequency Waveforms” (SFW). Stepped frequency waveforms require more complex hardware implementation as compared to LFM or FM-CW; however, the radar operating bandwidth requirements are less restrictive. This is true because the receiver instantaneous bandwidth is matched to the SFW sub-pulse bandwidth which is much smaller than the LFM or FM-CW bandwidth. A brief discussion of SFW waveforms is presented in the following section.

3.6. Stepped Frequency Waveforms

Stepped Frequency Waveforms (SFW) produce Synthetic HRR target profiles because the target range profile is computed by means of Inverse Discrete Fourier Transformation (IDFT) of frequency domain samples of the actual target range profile. The process of generating a synthetic HRR profile is described in Wehner.[1] It is summarized as follows:

1. A series of n narrow-band pulses are transmitted. The frequency from pulse to pulse is stepped by a fixed frequency step Δf. Each group of n pulses is referred to as a burst.

[1] Wehner, D. R., *High Resolution Radar,* second edition, Artech House, 1993.

2. The received signal is sampled at a rate that coincides with the center of each pulse.
3. The quadrature components for each burst are collected and stored.
4. Spectral weighting (to reduce the range sidelobe levels) is applied to the quadrature components. Corrections for target velocity, phase, and amplitude variations are applied.
5. 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 N bursts to obtain consecutive synthetic HRR profiles.

Fig. 3.9 shows a typical SFW burst. The Pulse Repetition Interval (PRI) is T, and the pulsewidth is τ'. Each pulse can have its own LFM, or other type of modulation; in this book LFM is assumed. The center frequency for the i^{th} step is

$$f_i = f_0 + i\Delta f \qquad ; \; i = 0, n-1 \tag{3.49}$$

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

$$s_i(t) = \begin{pmatrix} C_i \cos 2\pi f_i t + \theta_i & ; & iT \le t \le iT + \tau' \\ 0 & & elsewhere \end{pmatrix} \tag{3.50}$$

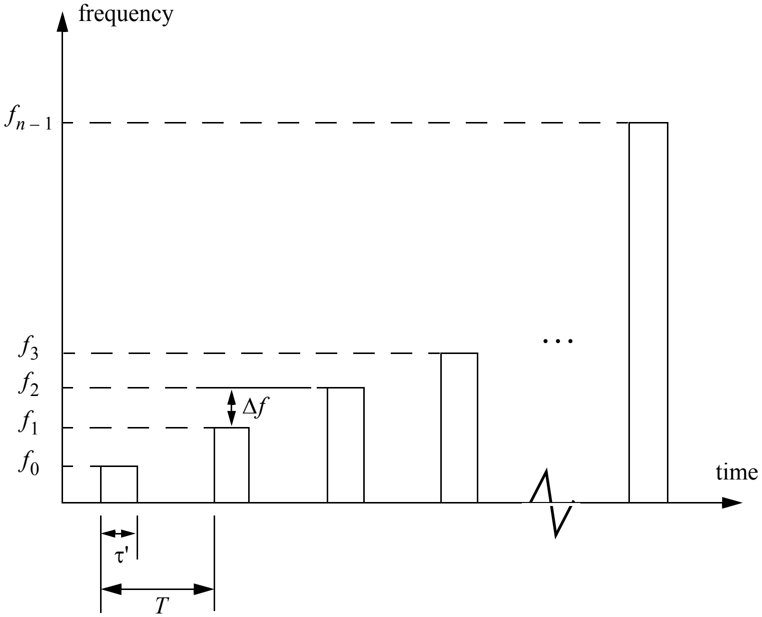


Figure 3.9. Stepped frequency waveform burst.

where θ_i are the relative phases and C_i are constants. The received signal from a target located at range R_0 at time $t = 0$ is then given by

$$s_{ri}(t) = C_i' \cos(2\pi f_i(t - \tau(t)) + \theta_i) \qquad ; \; iT + \tau(t) \le t \le iT + \tau' + \tau(t) \tag{3.51}$$

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

$$\tau(t) = \frac{R_0 - vt}{c/2} \tag{3.52}$$

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

The received signal is down-converted to base-band in order to extract the quadrature components. More precisely, $s_{ri}(t)$ is mixed with the signal

$$y_i(t) = C \cos(2\pi f_i t + \theta_i) \qquad ; \; iT \le t \le iT + \tau' \tag{3.53}$$

After low pass filtering, the quadrature components are given by

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} A_i \cos\psi_i(t) \\ A_i \sin\psi_i(t) \end{pmatrix} \tag{3.54}$$

where A_i are constants, and

$$\psi_i(t) = -2\pi f_i\left(\frac{2R_0}{c} - \frac{2vt}{c}\right) \tag{3.55}$$

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

$$t_i = iT + \frac{\tau_r}{2} + \frac{2R_0}{c} \tag{3.56}$$

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

$$X_i = A_i e^{j\psi_i} \tag{3.57}$$

Eq. (3.57) 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

$$H_l = \frac{1}{n}\sum_{i=0}^{n-1} X_i \ \exp\left(j\frac{2\pi li}{n}\right) \qquad ; \ 0 \le l \le n-1 \tag{3.58}$$

Substituting Eqs. (3.57) and (3.55) into (3.58) and collecting terms yield

$$H_l = \frac{1}{n}\sum_{i=0}^{n-1} A_i \ \exp\left\{j\left(\frac{2\pi li}{n} - 2\pi f_i\left(\frac{2R_0}{c} - \frac{2vt_i}{c}\right)\right)\right\} \tag{3.59}$$

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. (3.59) can be written as

$$H_l = \sum_{i=0}^{n-1} \exp\left\{j\left(\frac{2\pi li}{n} - 2\pi f_i\frac{2R_0}{c}\right)\right\} \tag{3.60}$$

Using $f_i = i\Delta f$ inside Eq. (3.60) yields

$$H_l = \sum_{i=0}^{n-1} \exp\left\{j\frac{2\pi i}{n}\left(-\frac{2nR_0\Delta f}{c} + l\right)\right\} \tag{3.61}$$

which can be simplified to

$$H_l = \frac{\sin\pi\chi}{\sin\frac{\pi\chi}{n}} \ \exp\left(j\frac{n-1}{2} \ \frac{2\pi\chi}{n}\right) \tag{3.62}$$

where

$$\chi = \frac{-2nR_0\Delta f}{c} + l \tag{3.63}$$

Finally, the synthesized range profile is

$$|H_l| = \left|\frac{\sin\pi\chi}{\sin\frac{\pi\chi}{n}}\right| \tag{3.64}$$

3.6.1. Range Resolution and Range Ambiguity in SFW

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

$$\Delta R = \frac{c}{2n\Delta f} \tag{3.65}$$

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

$$\psi_i(t) = 2\pi f_i \frac{2R_0}{c} \tag{3.66}$$

It follows that

$$\frac{\Delta\psi}{\Delta f} = \frac{4\pi(f_{i+1} - f_i)R_0}{(f_{i+1} - f_i)\ \ c} = \frac{4\pi R_0}{c} \tag{3.67}$$

or equivalently,

$$R_0 = \frac{\Delta\psi}{\Delta f}\frac{c}{4\pi} \tag{3.68}$$

It is clear from Eq. (3.68) that range ambiguity exists for $\Delta\psi = \Delta\psi + 2n\pi$. Therefore,

$$R_0 = \frac{\Delta\psi + 2n\pi}{\Delta f}\frac{c}{4\pi} = R_0 + n\left(\frac{c}{2\Delta f}\right) \tag{3.69}$$

and the unambiguous range window is

$$R_u = \frac{c}{2\Delta f} \tag{3.70}$$

Hence, 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

$$\Delta f \le c / 2E \tag{3.71}$$

where E is the target extent in meters.

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

$$\Delta f \le 1 / \tau' \tag{3.72}$$

and, in practice,

$$\Delta f \le 1 / 2\tau' \tag{3.73}$$

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.

MATLAB Function "hrr_profile.m"

The function *"hrr_profile.m"* computes and plots the synthetic HRR profile for a specific SFW. It is given in Listing 3.3 in Section 3.12. 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] = hrr_profile (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*
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*

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

nscat	*tau*	*n*	*deltaf*	*prf*	*v*
3	$100\mu\sec$	*64*	$10MHz$	$10KHz$	*0.0*

In this case,

$$\Delta R = \frac{3 \times 10^8}{2 \times 64 \times 10 \times 10^6} = 0.235m$$

$$R_u = \frac{3 \times 10^8}{2 \times 10 \times 10^6} = 15m$$

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.*

Fig. 3.10 shows the synthesized range profiles generated using the function *"hrr_profile.m"* and the first case when the Hamming window is not used. Fig. 3.11 is similar to Fig. 3.10, except in this case the Hamming window is used. Fig. 3.12 shows the synthesized range profile that corresponds to the second case (Hamming window is used). Note that all three scatterers were resolved in Figs. 3.10 and 3.11; however, the last two scatterers are not resolved in Fig. 3.12, since they are separated by less than ΔR .

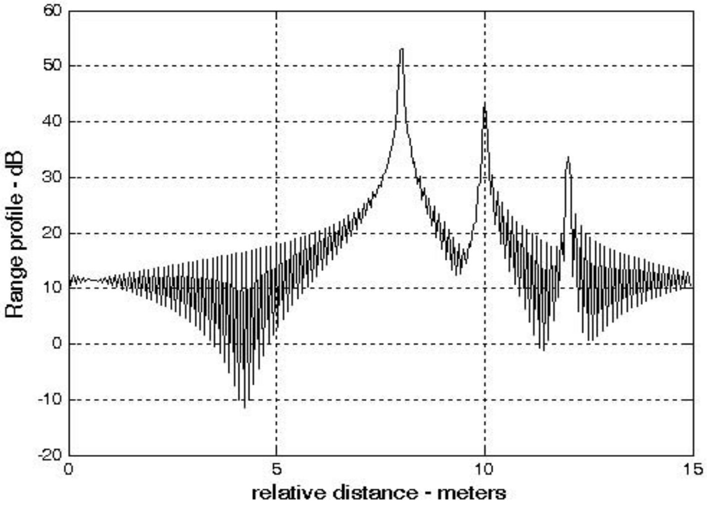


Figure 3.10. Synthetic range profile for three resolved scatterers. No window.

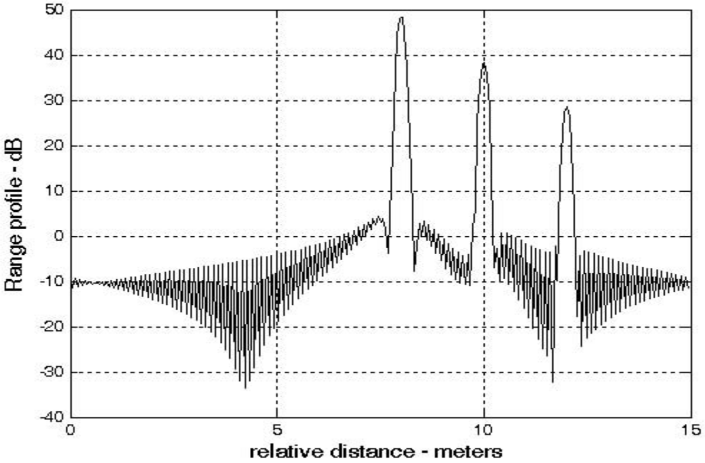


Figure 3.11. Synthetic range profile for three scatterers. Hamming window.

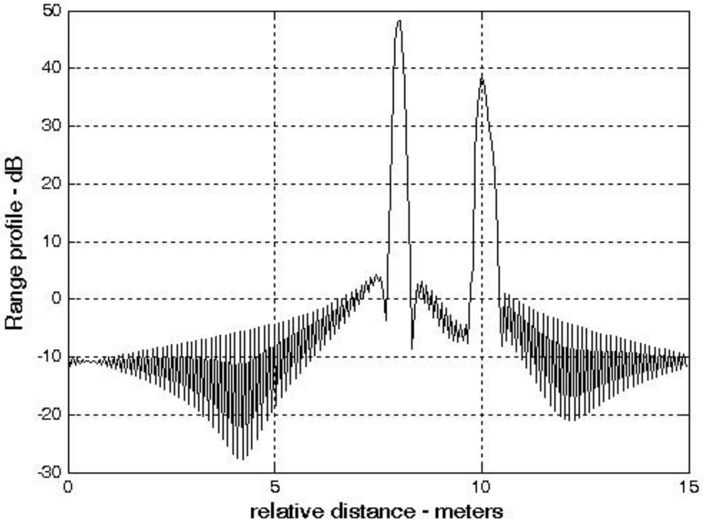


Figure 3.12. Synthetic range profile for three scatterers. Two are unresolved.

Next, consider another case where *[scat_range] = [908, 912, 916] meters.* Fig. 3.13 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

Fig. 3.14 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.

3.6.2. Effect of Target Velocity

The range profile defined in Eq. (3.64) 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 substituting Eqs. (3.55) and (3.56) into Eq. (3.58), which yields

$$H_l = \sum_{i=0}^{n-1} A_i \exp\left\{j\frac{2\pi l i}{n} - j2\pi f_i\left[\frac{2R}{c} - \frac{2v}{c}\left(iT + \frac{\tau_1}{2} + \frac{2R}{c}\right)\right]\right\} \qquad \textbf{(3.74)}$$

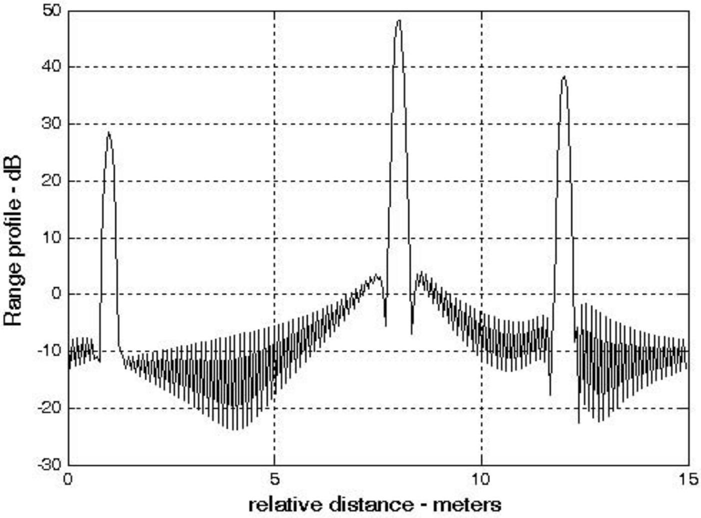


Figure 3.13. Synthetic range profile for three scatterers. Third scatterer folds over.

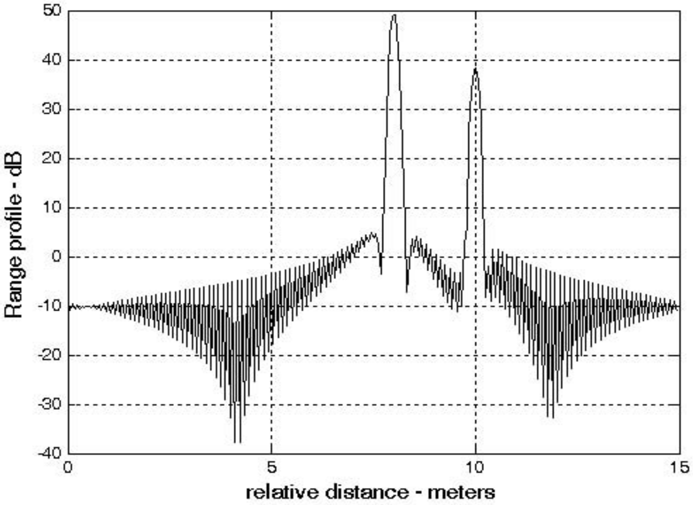


Figure 3.14. Synthetic range profile for three scatterers. The first and third scatterers appear in the same FFT bin.

The additional phase term present in Eq. (3.74) 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 $v = 100m/s$. Fig. 3.15 shows the synthesized range profile for this case. Comparisons of Figs. 3.11 and 3.15 clearly show the distortion effects caused by the uncompensated target velocity. Figure 3.16 is similar to Fig. 3.15 except in this case, $v = -100m/s$. 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

$$\Phi = \exp\left(-j2\pi f_i\left[\frac{2\underset{\sim}{v}}{c}\left(iT + \frac{\tau_1}{2} + \frac{2\underset{\sim}{R}}{c}\right)\right]\right) \qquad \textbf{(3.75)}$$

$\underset{\sim}{v}$ and $\underset{\sim}{R}$ 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 $\underset{\sim}{v}$ and $\underset{\sim}{R}$ 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.

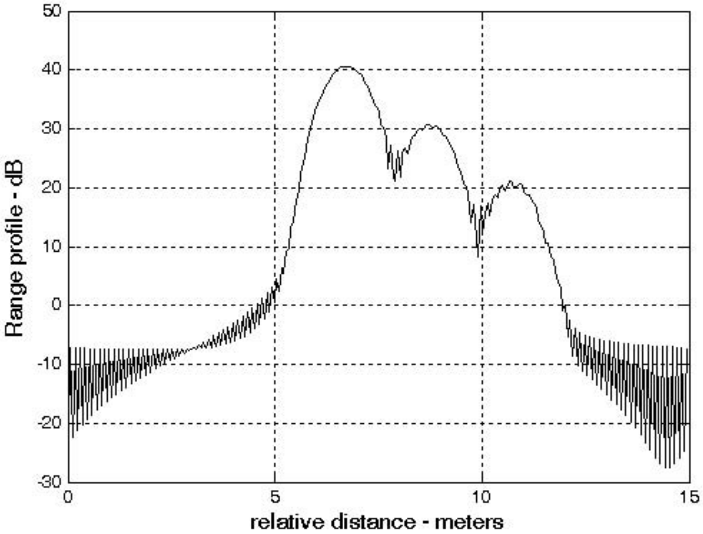


Figure 3.15. Illustration of range profile distortion due to target velocity.

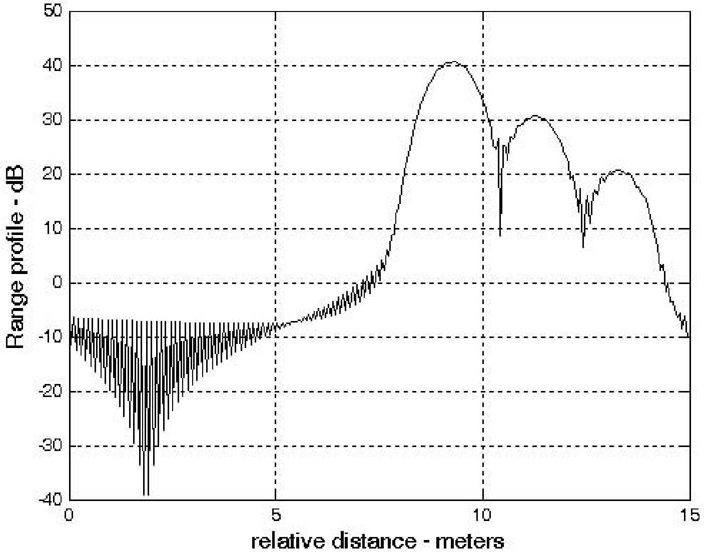


Figure 3.16. Illustration of range profile distortion due to target velocity.

Figure 3.17 shows a synthesized range profile for a moving target whose RCS is $\sigma = 10m^2$ and $v = 15m/s$. The initial target range is at $R = 912m$. All other parameters are as before. This figure can be reproduced using the MATLAB program *"fig3_17.m"* given in Listing 3.4 in Section 3.12.

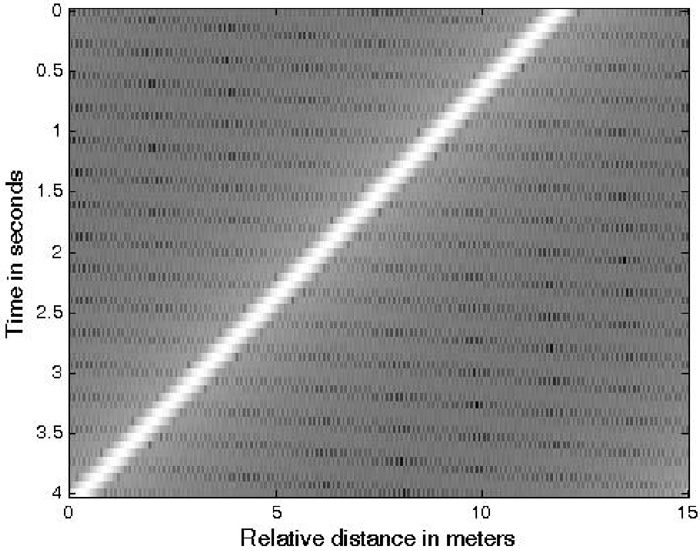


Figure 3.17. Synthesized range profile for a moving target (4 seconds long).

3.7. The Matched Filter

The most unique characteristic of the matched filter is that it produces the maximum achievable instantaneous SNR at its output when a signal plus additive white noise is present at the input. The noise does not need to be Gaussian. The peak instantaneous SNR at the receiver output can be achieved by matching the radar receiver transfer function to the received signal. We will show that the peak instantaneous signal power divided by the average noise power at the output of a matched filter is equal to twice the input signal energy divided by the input noise power, regardless of the waveform used by the radar. This is the reason why matched filters are often referred to as optimum filters in the SNR sense. Note that the peak power used in the derivation of the radar equation (SNR) represents the average signal power over the duration of the pulse, not the peak instantaneous signal power as in the case of a matched filter. In practice, it is sometimes difficult to achieve perfect matched filtering. In such cases, sub-optimum filters may be used. Due to this mismatch, degradation in the output SNR occurs.

Consider a radar system that uses a finite duration energy signal $s_i(t)$. Denote the pulsewidth as τ', and assume that a matched filter receiver is utilized. The main question that we need to answer is: What is the impulse, or frequency, response of the filter that maximizes the instantaneous SNR at the output of the receiver when a delayed version of the signal $s_i(t)$ plus additive white noise is at the input?

The matched filter input signal can then be represented by

$$x(t) = C \; s_i(t - t_1) + n_i(t) \tag{3.76}$$

where C is a constant, t_1 is an unknown time delay proportional to the target range, and $n_i(t)$ is input white noise. Since the input noise is white, its corresponding autocorrelation and Power Spectral Density (PSD) functions are given, respectively, by

$$\bar{R}_{n_i}(t) = \frac{N_0}{2} \; \delta(t) \tag{3.77}$$

$$\bar{S}_{n_i}(\omega) = \frac{N_0}{2} \tag{3.78}$$

where N_0 is a constant. Denote $s_o(t)$ and $n_o(t)$ as the signal and noise filter outputs, respectively. More precisely, we can define

$$y(t) = C \; s_o(t - t_1) + n_o(t) \tag{3.79}$$

where

$$s_o(t) = s_i(t) \bullet h(t) \tag{3.80}$$

$$n_o(t) = n_i(t) \bullet h(t) \tag{3.81}$$

The operator ($\bullet$) indicates convolution, and $h(t)$ is the filter impulse response (the filter is assumed to be linear time invariant).

Let $R_h(t)$ denote the filter autocorrelation function. It follows that the output noise autocorrelation and PSD functions are

$$\bar{R}_{n_o}(t) = \bar{R}_{n_i}(t) \bullet R_h(t) = \frac{N_0}{2} \; \delta(t) \bullet R_h(t) = \frac{N_0}{2} \; R_h(t) \tag{3.82}$$

$$\bar{S}_{n_o}(\omega) = \bar{S}_{n_i}(\omega) |H(\omega)|^2 = \frac{N_0}{2} \; |H(\omega)|^2 \tag{3.83}$$

where $H(\omega)$ is the Fourier transform for the filter impulse response, $h(t)$. The total average output noise power is equal to $\bar{R}_{n_o}(t)$ evaluated at $t = 0$. More precisely,

$$\bar{R}_{n_o}(0) = \frac{N_0}{2} \int_{-\infty}^{\infty} |h(u)|^2 du \tag{3.84}$$

The output signal power evaluated at time t is $|Cs_o(t - t_1)|^2$, and by using Eq. (3.80) we get

$$s_o(t - t_1) = \int_{-\infty}^{\infty} s_i(t - t_1 - u)\ \ h(u)\ \ du \tag{3.85}$$

A general expression for the output SNR at time t can be written as

$$SNR(t) = \frac{|Cs_o(t - t_1)|^2}{\bar{R}_{n_o}(0)} \tag{3.86}$$

Substituting Eqs. (3.84) and (3.85) into Eq. (3.86) yields

$$SNR(t) = \frac{C^2 \left| \int_{-\infty}^{\infty} s_i(t - t_1 - u)\ \ h(u)\ \ du \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |h(u)|^2 du} \tag{3.87}$$

The Schwartz inequality states that

$$\left| \int_{-\infty}^{\infty} P(x)Q(x)dx \right|^2 \le \int_{-\infty}^{\infty} |P(x)|^2 dx \int_{-\infty}^{\infty} |Q(x)|^2 dx \tag{3.88}$$

where the equality applies only when $P = kQ^*$, where k is a constant and can be assumed to be unity. Then by applying Eq. (3.88) on the numerator of Eq. (3.87), we get

$$SNR(t) \le \frac{C^2 \int_{-\infty}^{\infty} |s_i(t - t_1 - u)|^2\ \ du \int_{-\infty}^{\infty} |h(u)|^2\ \ du}{\frac{N_0}{2} \int_{-\infty}^{\infty} |h(u)|^2 du} = \frac{2C^2 \int_{-\infty}^{\infty} |s_i(t - t_1 - u)|^2\ \ du}{N_0} \tag{3.89}$$

Eq. (3.89) tells us that the peak instantaneous SNR occurs when equality is achieved (i.e., from Eq. (3.88) $h = ks_i^*$). More precisely, if we assume that equality occurs at $t = t_0$, and that $k = 1$, then

$$h(u) = s_i^*(t_0 - t_1 - u) \tag{3.90}$$

and the maximum instantaneous SNR is

$$SNR(t_0) = \frac{2C^2 \int\limits_{-\infty}^{\infty} |s_i(t_0 - t_1 - u)|^2 \; du}{N_0} \tag{3.91}$$

Eq. (3.91) can be simplified using Parseval's theorem,

$$E = C^2 \int\limits_{-\infty}^{\infty} |s_i(t_0 - t_1 - u)|^2 \; du \tag{3.92}$$

where E denotes the energy of the input signal; consequently we can write the output peak instantaneous SNR as

$$SNR(t_0) = \frac{2E}{N_0} \tag{3.93}$$

Thus, we can draw the conclusion that the peak instantaneous SNR depends only on the signal energy and input noise power, and is independent of the waveform utilized by the radar.

Finally, we can define the impulse response for the matched filter from Eq. (3.90). If we desire the peak to occur at $t_0 = t_1$, we get the non-causal matched filter impulse response,

$$h_{nc}(t) = s_i^*(-t) \tag{3.94}$$

Alternatively, the causal impulse response is

$$h_c(t) = s_i^*(\tau - t) \tag{3.95}$$

where, in this case, the peak occurs at $t_0 = t_1 + \tau$. It follows that the Fourier transforms of $h_{nc}(t)$ and $h_c(t)$ are given, respectively, by

$$H_{nc}(\omega) = S_i^*(\omega) \tag{3.96}$$

$$H_c(\omega) = S_i^*(\omega)e^{-j\omega\tau} \tag{3.97}$$

where $S_i(\omega)$ is the Fourier transform of $s_i(t)$. Thus, the moduli of $H(\omega)$ and $S_i(\omega)$ are identical; however, the phase responses are opposite of each other.

Example:

Compute the maximum instantaneous SNR at the output of a linear filter whose impulse response is matched to the signal $x(t) = \exp(-t^2/2T)$.

Solution:

The signal energy is

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} e^{(-t^2)/T} dt = \sqrt{\pi T} \ Joules$$

It follows that the maximum instantaneous SNR is

$$SNR = \frac{\sqrt{\pi T}}{N_0/2}$$

where $N_0/2$ *is the input noise power spectrum density.*

3.8. The Replica

Again, consider a radar system that uses a finite duration energy signal $s_i(t)$, and assume that a matched filter receiver is utilized. The input signal is given in Eq. (3.76) and is repeated here as Eq. (3.98),

$$x(t) = C \ s_i(t - t_1) + n_i(t) \tag{3.98}$$

The matched filter output $y(t)$ can be expressed by the convolution integral between the filter's impulse response and $x(t)$,

$$y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)du \tag{3.99}$$

Substituting Eq. (3.95) into Eq. (3.99) yields

$$y(t) = \int_{-\infty}^{\infty} x(u)s_i^*(\tau - t + u)du = \bar{R}_{xs_i}(t - \tau) \tag{3.100}$$

where $\bar{R}_{xs_i}(t-\tau)$ is a cross-correlation between $x(t)$ and $s_i(\tau - t)$. Therefore, the matched filter output can be computed from the cross-correlation between the radar received signal and a delayed replica of the transmitted waveform. If the input signal is the same as the transmitted signal, the output of the matched filter would be the autocorrelation function of the received (or transmitted) signal. In practice, replicas of the transmitted waveforms are normally computed and stored in memory for use by the radar signal processor when needed.

3.9. Matched Filter Response to LFM Waveforms

In order to develop a general expression for the matched filter output when an LFM waveform is utilized, we will consider the case when the radar is tracking a closing target with velocity v. The transmitted signal is

$$s_1(t) = Rect\left(\frac{t}{\tau'}\right)e^{j2\pi\left(f_0t + \frac{\mu}{2}t^2\right)} \tag{3.101}$$

The received signal is then given by

$$s_{r_1}(t) = s_1(t - \Delta(t)) \tag{3.102}$$

$$\Delta(t) = t_0 - \frac{2v}{c}(t - t_0) \tag{3.103}$$

where t_0 is the time corresponding to the target initial detection range, and c is the speed of light. Using Eq. (3.103) we can rewrite Eq. (3.102) as

$$s_{r_1}(t) = s_1\left(t - t_0 + \frac{2v}{c}(t - t_0)\right) = s_1(\gamma(t - t_0)) \tag{3.104}$$

and

$$\gamma = 1 + 2\frac{v}{c} \tag{3.105}$$

is the scaling coefficient. Substituting Eq. (3.101) into Eq. (3.104) yields

$$s_{r_1}(t) = Rect\left(\frac{\gamma(t - t_0)}{\tau'}\right)e^{j2\pi f_0\gamma(t - t_0)}\ e^{j\pi\mu\gamma^2(t - t_0)^2} \tag{3.106}$$

which is the analytical signal representation for $s_{r_1}(t)$. The complex envelope of the signal $s_{r_1}(t)$ is obtained by multiplying Eq. (3.106) by $\exp(-j2\pi f_0 t)$. Denote the complex envelope by $s_r(t)$; then after some manipulation we get

$$s_r(t) = e^{-j2\pi f_0 t_0}Rect\left(\frac{\gamma(t - t_0)}{\tau'}\right)e^{j2\pi f_0(\gamma - 1)(t - t_0)}\ e^{j\pi\mu\gamma^2(t - t_0)^2} \tag{3.107}$$

The Doppler shift due to the target motion is

$$f_d = \frac{2v}{c}f_0 \tag{3.108}$$

and since $\gamma - 1 = 2v/c$, we get

$$f_d = (\gamma - 1)f_0 \tag{3.109}$$

Using the approximation $\gamma \approx 1$ and Eq. (3.109), Eq. (3.107) is rewritten as

$$s_r(t) \approx e^{j2\pi f_d(t - t_0)} \; s(t - t_0) \tag{3.110}$$

where

$$s(t - t_0) = e^{-j2\pi f_0 t} s_1(t - t_0) \tag{3.111}$$

$s_1(t)$ is given in Eq. (3.101). The matched filter response is given by the convolution integral

$$s_o(t) = \int_{-\infty}^{\infty} h(u)s_r(t - u)du \tag{3.112}$$

For a non-causal matched filter the impulse response $h(u)$ is equal to $s^*(-t)$; it follows that

$$s_o(t) = \int_{-\infty}^{\infty} s^*(-u)s_r(t - u)du \tag{3.113}$$

Substituting Eq. (3.111) into Eq. (3.113), and performing some algebraic manipulations, we get

$$s_o(t) = \int_{-\infty}^{\infty} s^*(u) \; e^{j2\pi f_d(t + u - t_0)} \; s(t + u - t_0)du \tag{3.114}$$

Finally, making the change of variable $t' = t + u$ yields

$$s_o(t) = \int_{-\infty}^{\infty} s^*(t' - t)s(t' - t_0)e^{j2\pi f_d(t' - t_0)} \; dt' \tag{3.115}$$

It is customary to set $t_0 = 0$. It follows that

$$s_o(t;f_d) = \int_{-\infty}^{\infty} s(t')s^*(t' - t)e^{j2\pi f_d t'} \; dt' \tag{3.116}$$

where we used the notation $s_o(t;f_d)$ to indicate that the output is a function of both time and Doppler frequency.

3.10. Waveform Resolution and Ambiguity

As indicated by Eq. (3.93) the radar sensitivity (in the case of white additive noise) depends only on the total energy of the received signal and is independent of the shape of the specific waveform. This leads us to ask the following question: If the radar sensitivity is independent of the waveform, then what is the best choice for a transmitted waveform? The answer depends on many factors; however, the most important consideration lies in the waveform's range and Doppler resolution characteristics.

As discussed in Chapter 1, range resolution implies separation between distinct targets in range. Alternatively, Doppler resolution implies separation between distinct targets in frequency. Thus, ambiguity and accuracy of this separation are closely associated terms.

3.10.1. Range Resolution

Consider radar returns from two stationary targets (zero Doppler) separated in range by distance ΔR. What is the smallest value of ΔR so that the returned signal is interpreted by the radar as two distinct targets? In order to answer this question, assume that the radar transmitted pulse is denoted by $s(t)$,

$$s(t) = A(t)\cos(2\pi f_0 t + \phi(t)) \tag{3.117}$$

where f_0 is the carrier frequency, $A(t)$ is the amplitude modulation, and $\phi(t)$ is the phase modulation. The signal $s(t)$ can then be expressed as the real part of the complex signal $\psi(t)$, where

$$\psi(t) = A(t)e^{j(\omega_0 t - \phi(t))} = u(t)e^{j\omega_0 t} \tag{3.118}$$

and

$$u(t) = A(t)e^{-j\phi(t)} \tag{3.119}$$

It follows that

$$s(t) = Re\{\psi(t)\} \tag{3.120}$$

The returns from both targets are respectively given by

$$s_{r1}(t) = \psi(t - \tau_0) \tag{3.121}$$

$$s_{r2}(t) = \psi(t - \tau_0 - \tau) \tag{3.122}$$

where τ is the difference in delay between the two returns. One can assume that the reference time is τ_0, and thus without any loss of generality one may set $\tau_0 = 0$. It follows that the two targets are distinguishable by how large or small the delay τ can be.

In order to measure the difference in range between the two targets consider the integral square error between $\psi(t)$ and $\psi(t-\tau)$. Denoting this error as ε_R^2, it follows that

$$\varepsilon_R^2 = \int_{-\infty}^{\infty} |\psi(t) - \psi(t-\tau)|^2 \ dt \tag{3.123}$$

Eq. (3.123) can be written as

$$\varepsilon_R^2 = \int_{-\infty}^{\infty} |\psi(t)|^2 \ dt + \int_{-\infty}^{\infty} |\psi(t-\tau)|^2 \ dt - \int_{-\infty}^{\infty} \{(\psi(t)\psi^*(t-\tau) + \psi^*(t)\psi(t-\tau)) \ dt\} \tag{3.124}$$

Using Eq. (3.118) into Eq. (3.124) yields

$$\varepsilon_R^2 = 2\int_{-\infty}^{\infty} |u(t)|^2 \ dt - 2Re\left\{ \int_{-\infty}^{\infty} \psi^*(t)\psi(t-\tau) \ dt \right\} = 2\int_{-\infty}^{\infty} |u(t)|^2 \ dt - 2Re\left\{ e^{-j\omega_0\tau} \int_{-\infty}^{\infty} u^*(t)u(t-\tau) \ dt \right\} \tag{3.125}$$

The first term in the right hand side of Eq. (3.125) represents the signal energy, and is assumed to be constant. The second term is a varying function of τ with its fluctuation tied to the carrier frequency. The integral inside the right-most side of this equation is defined as the “range ambiguity function,”

$$\chi_R(\tau) = \int_{-\infty}^{\infty} u^*(t)u(t-\tau) \ dt \tag{3.126}$$

The maximum value of $\chi_R(\tau)$ is at $\tau = 0$. Target resolvability in range is measured by the squared magnitude $|\chi_R(\tau)|^2$. It follows that if $|\chi_R(\tau)| = \chi_R(0)$ for some nonzero value of τ, then the two targets are indistinguishable. Alternatively, if $|\chi_R(\tau)| \neq \chi_R(0)$ for some nonzero value of τ, then the two targets may be distinguishable (resolvable). As a consequence, the most desirable shape for $\chi_R(\tau)$ is a very sharp peak (thumb tack shape) centered at $\tau = 0$ and falling very quickly away from the peak.

The time delay resolution is

$$\Delta\tau = \frac{\int_{-\infty}^{\infty} |\chi_R(\tau)|^2 \; d\tau}{\chi_R^2(0)} \tag{3.127}$$

Using Parseval's theorem, Eq. (3.127) can be written as

$$\Delta\tau = 2\pi \frac{\int_{-\infty}^{\infty} |U(\omega)|^4 \; d\omega}{\left[\int_{-\infty}^{\infty} |U(\omega)|^2 \; d\omega\right]^2} \tag{3.128}$$

The minimum range resolution corresponding to $\Delta\tau$ is

$$\Delta R = c\Delta\tau / 2 \tag{3.129}$$

However, since the signal effective bandwidth is

$$B = \frac{\left[\int_{-\infty}^{\infty} |U(\omega)|^2 \; d\omega\right]^2}{2\pi \int_{-\infty}^{\infty} |U(\omega)|^4 \; d\omega} \tag{3.130}$$

the range resolution is expressed as a function of the waveform's bandwidth as

$$\Delta R = c / (2B) \tag{3.131}$$

The comparison between Eqs. (3.116) and (3.126) indicates that the output of the matched filter and the range ambiguity function have the same envelope (in this case the Doppler shift f_d is set to zero). This indicates that the matched filter, in addition to providing the maximum instantaneous SNR at its output, also preserves the signal range resolution properties.

3.10.2. Doppler Resolution

It was shown in Chapter 1 that the Doppler shift corresponding to the target radial velocity is

$$f_d = \frac{2v}{\lambda} = \frac{2vf_0}{c} \tag{3.132}$$

where v is the target radial velocity, λ is the wavelength, f_0 is the frequency, and c is the speed of light.

Let

$$\Psi(f) = \int_{-\infty}^{\infty} \psi(t) e^{-j2\pi ft}\ dt \tag{3.133}$$

Due to the Doppler shift associated with the target, the received signal spectrum will be shifted by f_d. In other words the received spectrum can be represented by $\Psi(f-f_d)$. In order to distinguish between the two targets located at the same range but having different velocities, one may use the integral square error. More precisely,

$$\varepsilon_f^2 = \int_{-\infty}^{\infty} \left|\Psi(f) - \Psi(f-f_d)\right|^2\ df \tag{3.134}$$

Using similar analysis as that which led to Eq. (3.125), one should minimize

$$2Re\left\{\int_{-\infty}^{\infty} \Psi^*(f)\Psi(f-f_d)\ df\right\} \tag{3.135}$$

By using the analytic signal in Eq. (3.118) it can be shown that

$$\Psi(f) = U(2\pi f - 2\pi f_0) \tag{3.136}$$

Thus, Eq. (3.135) becomes

$$\int_{-\infty}^{\infty} U^*(2\pi f)U(2\pi f - 2\pi f_d)\ df = \int_{-\infty}^{\infty} U^*(2\pi f - 2\pi f_0)U(2\pi f - 2\pi f_0 - 2\pi f_d)\ df \tag{3.137}$$

The complex frequency correlation function is then defined as

$$\chi_f(f_d) = \int_{-\infty}^{\infty} U^*(2\pi f)U(2\pi f - 2\pi f_d)\ df = \int_{-\infty}^{\infty} |u(t)|^2\ ^{j2\pi f_d t}\ dt \tag{3.138}$$

and the Doppler resolution constant Δf_d is

$$\Delta f_d = \frac{\int_{-\infty}^{\infty} |\chi_f(f_d)|^2 df_d}{\chi_f^2(0)} = \frac{\int_{-\infty}^{\infty} |u(t)|^4 dt}{\left[\int_{-\infty}^{\infty} |u(t)|^2 dt\right]^2} = \frac{1}{\tau'} \tag{3.139}$$

where τ' is pulsewidth.

Finally, one can define the corresponding velocity resolution as

$$\Delta v = \frac{c\Delta f_d}{2f_0} = \frac{c}{2f_0\tau'} \tag{3.140}$$

Again observation of Eqs. (3.138) and (3.116) indicate that the output of the matched filter and the ambiguity function (when $\tau = 0$) are similar to each other. Consequently, one concludes that the matched filter preserves the waveform Doppler resolution properties as well.

3.10.3. Combined Range and Doppler Resolution

In this general case, one needs to use a two-dimensional function in the pair of variables (τ, f_d). For this purpose, assume that the complex envelope of the transmitted waveform is

$$\psi(t) = u(t)e^{j2\pi f_0 t} \tag{3.141}$$

Then the delayed and Doppler-shifted signal is

$$\psi'(t-\tau) = u(t-\tau)e^{j2\pi(f_0-f_d)(t-\tau)} \tag{3.142}$$

Computing the integral square error between Eqs. (3.142) and (3.141) yields

$$\varepsilon^2 = \int_{-\infty}^{\infty} |\psi(t) - \psi'(t-\tau)|^2 \; dt = 2\int_{-\infty}^{\infty} |\psi(t)|^2 dt - 2Re\left\{\int_{-\infty}^{\infty} \psi^*(t) - \psi'(t-\tau)dt\right\} \tag{3.143}$$

which can be written as

$$\varepsilon^2 = 2\int_{-\infty}^{\infty} |u(t)|^2 \; dt - 2Re\left\{e^{j2\pi(f_0-f_d)\tau}\int_{-\infty}^{\infty} u(t)u^*(t-\tau)e^{j2\pi f_d t}dt\right\} \tag{3.144}$$

Again, in order to maximize this squared error for $\tau \neq 0$ one must minimize the last term of Eq. (3.144).

Define the combined range and Doppler correlation function as

$$\chi(\tau, f_d) = \int_{-\infty}^{\infty} u(t)u^*(t-\tau)e^{j2\pi f_d t}dt \tag{3.145}$$

In order to achieve the most range and Doppler resolution, the modulus square of this function must be minimized for $\tau \neq 0$ and $f_d \neq 0$. Note that the output of the matched filter in Eq. (3.116) is identical to that given in Eq. (3.145). This means that the output of the matched filter exhibits maximum instantaneous SNR as well as the most achievable range and Doppler resolutions.

3.11. "MyRadar" Design Case Study - Visit 3

3.11.1. Problem Statement

Assuming a matched filter receiver, select a set of waveforms that can meet the design requirements as stated in the previous two chapters. Assume linear frequency modulation. Do not use more than a total of 5 waveforms. Modify the design so that the range resolution $\Delta R = 30m$ during the search mode, and $\Delta R = 7.5m$ during tracking.

3.11.2. A Design

The major characteristics of radar waveforms include the waveform's energy, range resolution, and Doppler (or velocity) resolution. The pulse (waveform) energy is

$$E = P_t \tau \tag{3.146}$$

where P_t is the peak transmitted power and τ is the pulsewidth. Range resolution is defined in Eq. (3.131), while the velocity resolution is in Eq. (3.140).

Close attention should be paid to the selection process of the pulsewidth. In this design we will assume that the pulse energy is the same as that computed in Chapter 2. The radar operating bandwidth during search and track are calculated from Eq. (3.131) as

$$\begin{Bmatrix} B_{search} \\ B_{track} \end{Bmatrix} = \begin{Bmatrix} 3 \times 10^8 / (2 \times 30) = 5 \ \ MHz \\ 3 \times 10^8 / (2 \times 7.5) = 20 \ \ MHz \end{Bmatrix} \tag{3.147}$$

Since the design calls for a pulsed radar, then for each pulse transmitted (one PRI) the radar should not be allowed to receive any signal until that pulse has been completely transmitted. This limits the radar to a minimum operating range defined by

$$R_{min} = \frac{c\tau}{2} \tag{3.148}$$

In this design choose $R_{min} \geq 15Km$. It follows that the minimum acceptable pulsewidth is $\tau_{max} \leq 100\mu s$.

For this design select 5 waveforms, one for search and four for track. Typically search waveforms are longer than track waveforms; alternatively, tracking waveforms require wider bandwidths than search waveforms. However, in the context of range, more energy is required at longer ranges (for both track and search waveforms), since one would expect the SNR to get larger as range becomes smaller. This was depicted in the example shown in Fig. 1.13 in Chapter 1.

Assume that during search and initial detection the single pulse peak power is to be kept under 10 KW (i.e., $P_t \le 20KW$). Then by using the single pulse energy calculated using Eq. (2.115) in Chapter 2, one can compute the minimum required pulsewidth as

$$\tau_{min} \ge \frac{0.1147}{20 \times 10^3} = 5.735\mu s \tag{3.149}$$

Choose $\tau_{search} = 20\mu s$, with bandwidth $B = 5MHz$ and use LFM modulation. Fig. 3.18 shows plots of the real part, imaginary part, and the spectrum of this search waveform. This figure was produced using the GUI workspace *"LFM_gui.m"*. As far as the track waveforms, choose four waveforms of the same bandwidth ($B_{track} = 20MHz$) and with the following pulsewidths.

TABLE 3.1. *"MyRadar"* design case study track waveforms.

Pulsewidth	**Range window**
$\tau_{t1} = 20\mu s$	$R_{max} \to 0.75R_{max}$
$\tau_{t2} = 17.5\mu s$	$0.75R_{max} \to 0.5R_{max}$
$\tau_{t3} = 15\mu s$	$0.5R_{max} \to 0.25R_{max}$
$\tau_{t4} = 12.5\mu s$	$R \le 0.25R_{max}$

Note that R_{max} refers to the initial range at which track has been initiated. Fig. 3.19 is similar to Fig. 3.18 except it is for τ_{t3}.

For the waveform set selected in this design option, the radar duty cycle varies from 1.25% to 2.0%. Remember that the PRF was calculated in Chapter 1 as $f_r = 1KHz$; thus the PRI is $T = 1ms$.

At this point of the design, one must verify that the selected waveforms provide the radar with the desired SNR that meets or exceeds what was calculated in Chapter 2, and plotted in Fig. 2.21. In other words, one must now re-run these calculations and verify that the SNR has not been degraded. This task will be postponed until Chapter 5, where the radar equation with pulse compression is developed.

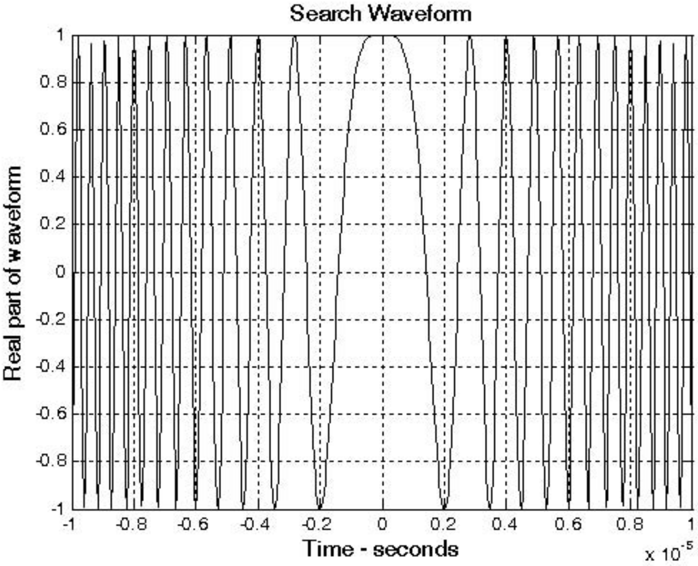


Figure 3.18a. Real part of search waveform.

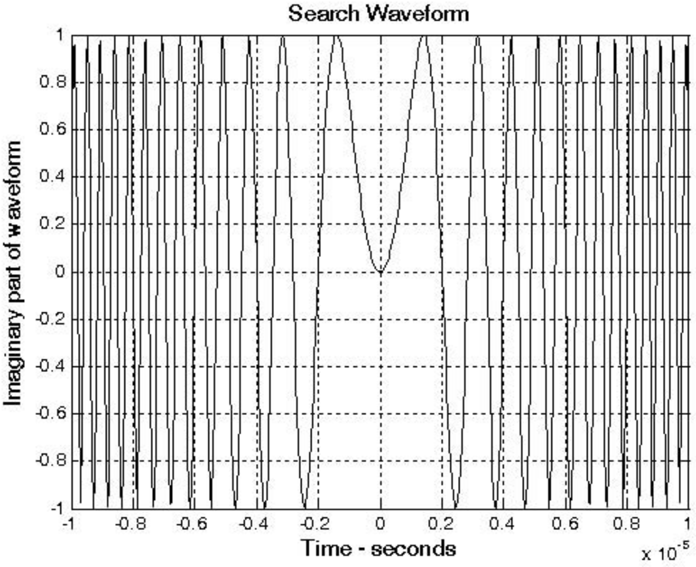


Figure 3.18b. Imaginary part of search waveform.

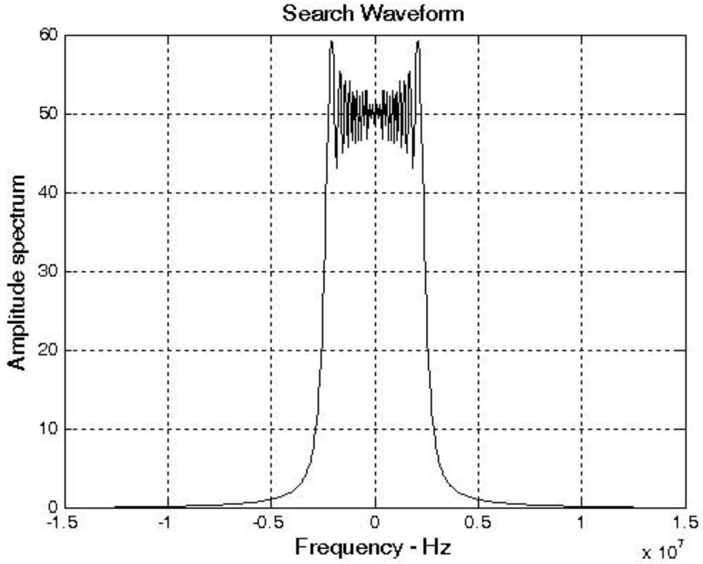


Figure 3.18c. Amplitude spectrum.

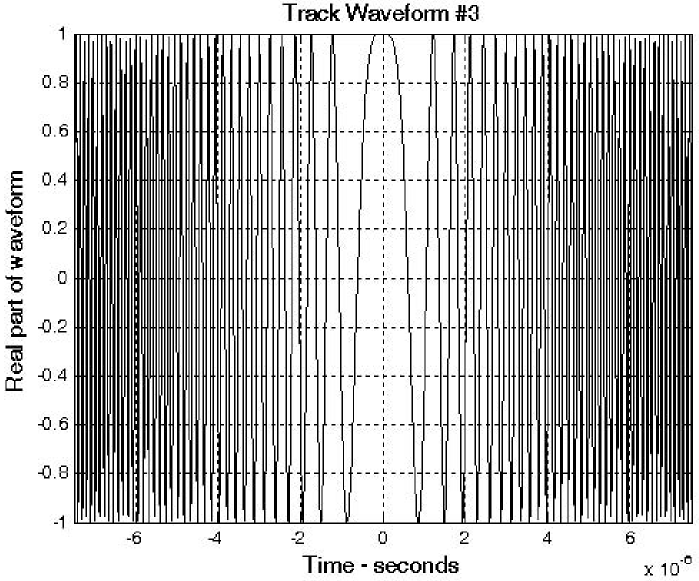


Figure 3.19a. Real part of waveform.

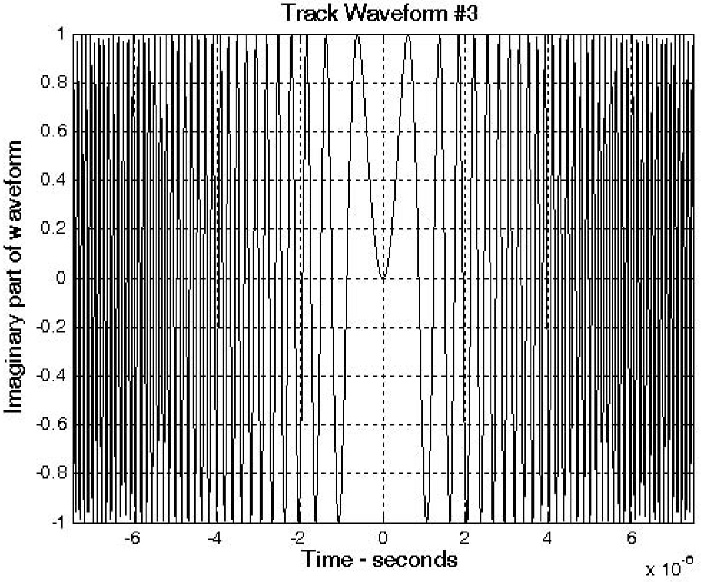


Figure 3.19b. Imaginary part of waveform.

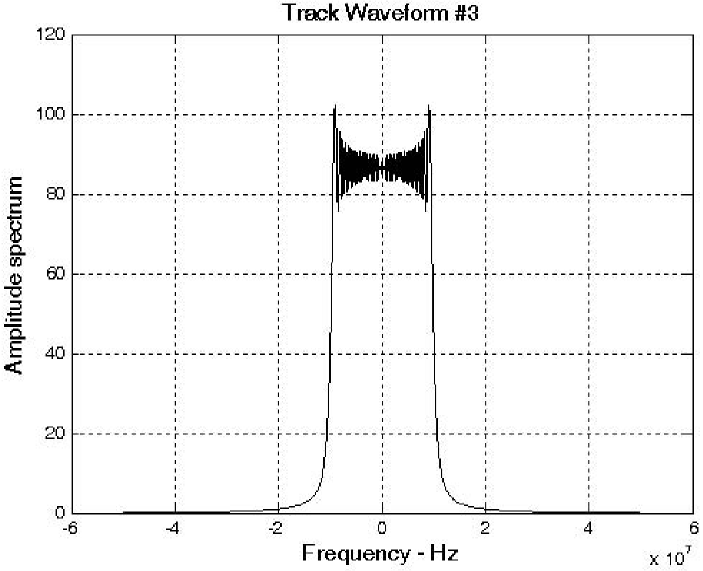


Figure 3.19c. Amplitude spectrum.

3.12. MATLAB Program and Function Listings

This section presents listings for all MATLAB programs/functions used in this chapter.

Listing 3.1. MATLAB Program "fig3_7.m"

```
% Use this program to reproduce Fig 3.7 from text
clear all
close all
n = 0;
 for x = 0:.05:4
    n = n+1;
   sx(n) = quadl('fresnels',.0,x);
   cx(n) = quadl('fresnelc',.0,x);
end
 plot(cx)
 x=0:.05:4;
 plot(x,cx,'k',x,sx,'k--')
 grid
 xlabel ('x')
 ylabel ('Fresnel integrals: C(x); S(x)')
 legend('C(x)','S(x)')
```

Listing 3.2. MATLAB Program "fig3_8.m"

```
% Use this program to reproduce Fig. 3.8 of text
close all
clear all
eps = 0.000001;
%Enter pulsewidth and bandwidth
B = 200.0e6; %200 MHZ bandwidth
T = 10.e-6; %10 micro second pulse;
% Compute alpha
mu = 2. * pi * B / T;
% Determine sampling times
delt = linspace(-T/2., T/2., 10001); % 1 nano second sampling interval
% Compute the complex LFM representation
Ichannal = cos(mu .* delt.^2 / 2.); % Real part
Qchannal = sin(mu .* delt.^2 / 2.); % Imaginary Part
LFM = Ichannal + sqrt(-1) .* Qchannal; % complex signal
%Compute the FFT of the LFM waveform
LFMFFT = fftshift(fft(LFM));
% Plot the real and Imaginary parts and the spectrum
```

```
freqlimit = 0.5 / 1.e-9;% the sampling interval 1 nano-second
freq = linspace(-freqlimit/1.e6,freqlimit/1.e6,10001);
figure(1)
plot(delt*1e6,Ichannal,'k');
axis([-1 1 -1 1])
grid
xlabel('Time - microsecs')
ylabel('Real part')
title('T = 10 Microsecond, B = 200 MHz')
figure(2)
plot(delt*1e6,Qchannal,'k');
axis([-1 1 -1 1])
grid
xlabel('Time - microsecs')
ylabel('Imaginary part')
title('T = 10 Microsecond, B = 200 MHz')
figure(3)
plot(freq, abs(LFMFFT),'k');
%axis tight
grid
xlabel('Frequency - MHz')
ylabel('Amplitude spectrum')
title('Spectrum for an LFM waveform and T = 10 Microsecond, ...
B = 200 MHZ')
```

Listing 3.3. MATLAB Function “hrr_profile.m”

```
function [hl] = hrr_profile (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 prede-
                        termined
% 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;
```

```
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).* win-
                         dow';
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 - meters')
ylabel ('Range profile - dB')
grid
```

Listing 3.4. MATLAB Program "fig3_17.m"

```
% use this program to reproduce Fig. 3.17 of text
clear all
close all
nscat = 1;
scat_range = 912;
scat_rcs = 10;
n =64;
deltaf = 10e6;
prf = 10e3;
v = 15;
rnote = 900,
winid = 1;
count = 0;
for time = 0:.05:3
   count = count +1;
   hl = hrr_profile (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')
```

Chapter 4

The Radar Ambiguity Function

4.1. Introduction

The radar ambiguity function represents the output of the matched filter, and it describes the interference caused by the range and/or Doppler shift of a target when compared to a reference target of equal RCS. The ambiguity function evaluated at $(\tau, f_d) = (0, 0)$ is equal to the matched filter output that is matched perfectly to the signal reflected from the target of interest. In other words, returns from the nominal target are located at the origin of the ambiguity function. Thus, the ambiguity function at nonzero τ and f_d represents returns from some range and Doppler different from those for the nominal target.

The radar ambiguity function is normally used by radar designers as a means of studying different waveforms. It can provide insight about how different radar waveforms may be suitable for the various radar applications. It is also used to determine the range and Doppler resolutions for a specific radar waveform. The three-dimensional (3-D) plot of the ambiguity function versus frequency and time delay is called the radar ambiguity diagram. The radar ambiguity function for the signal $s(t)$ is defined as the modulus squared of its 2-D correlation function, i.e., $|\chi(\tau;f_d)|^2$. More precisely,

$$|\chi(\tau;f_d)|^2 = \left| \int_{-\infty}^{\infty} s(t)s^*(t-\tau)e^{j2\pi f_d t}\,dt \right|^2 \tag{4.1}$$

In this notation, the target of interest is located at $(\tau, f_d) = (0, 0)$, and the ambiguity diagram is centered at the same point. Note that some authors define the ambiguity function as $|\chi(\tau;f_d)|$. In this book, $|\chi(\tau;f_d)|$ is called the uncertainty function.

Denote E as the energy of the signal $s(t)$,

$$E = \int_{-\infty}^{\infty} |s(t)|^2 dt \tag{4.2}$$

The following list includes the properties for the radar ambiguity function:

1) The maximum value for the ambiguity function occurs at $(\tau, f_d) = (0, 0)$ and is equal to $4E^2$,

$$max\{|\chi(\tau;f_d)|^2\} = |\chi(0;0)|^2 = (2E)^2 \tag{4.3}$$

$$|\chi(\tau;f_d)|^2 \le |\chi(0;0)|^2 \tag{4.4}$$

2) The ambiguity function is symmetric,

$$|\chi(\tau;f_d)|^2 = |\chi(-\tau;-f_d)|^2 \tag{4.5}$$

3) The total volume under the ambiguity function is constant,

$$\int\int |\chi(\tau;f_d)|^2 \; d\tau \;\; df_d = (2E)^2 \tag{4.6}$$

4) If the function $S(f)$ is the Fourier transform of the signal $s(t)$, then by using Parseval's theorem we get

$$|\chi(\tau;f_d)|^2 = \left|\int S^*(f) S(f-f_d) e^{-j2\pi f\tau} df \;\right|^2 \tag{4.7}$$

4.2. Examples of the Ambiguity Function

The ideal radar ambiguity function is represented by a spike of infinitesimally small width that peaks at the origin and is zero everywhere else, as illustrated in Fig. 4.1. An ideal ambiguity function provides perfect resolution between neighboring targets regardless of how close they may be to each other. Unfortunately, an ideal ambiguity function cannot physically exist. This is because the ambiguity function must have finite peak value equal to $(2E)^2$ and a finite volume also equal to $(2E)^2$. Clearly, the ideal ambiguity function cannot meet those two requirements.

4.2.1. Single Pulse Ambiguity Function

Consider the normalized rectangular pulse $s(t)$ defined by

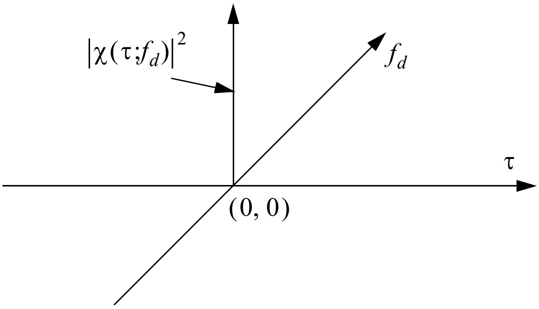


Figure 4.1. Ideal ambiguity function.

$$s(t) = \frac{1}{\sqrt{\tau'}} Rect\left(\frac{t}{\tau'}\right) \tag{4.8}$$

From Eq. (4.1) we have

$$\chi(\tau;f_d) = \int_{-\infty}^{\infty} s(t)s^*(t-\tau)e^{j2\pi f_d t}\,dt \tag{4.9}$$

Substituting Eq. (4.8) into Eq. (4.9) and performing the integration yield

$$|\chi(\tau;f_d)|^2 = \left|\left(1 - \frac{|\tau|}{\tau'}\right) \frac{\sin(\pi f_d(\tau' - |\tau|))}{\pi f_d(\tau' - |\tau|)}\right|^2 \qquad |\tau| \le \tau' \tag{4.10}$$

MATLAB Function "single_pulse_ambg.m"

The function *"single_pulse_ambg.m"* implements Eq. (4.10). It is given in Listing 4.1 in Section 4.6. The syntax is as follows:

single_pulse_ambg [taup]

taup is the pulsewidth. Fig 4.2 (a-d) show 3-D and contour plots of single pulse uncertainty and ambiguity functions. These plots can be reproduced using MATLAB program *"fig4_2.m"* given in Listing 4.2 in Section 4.6.

The ambiguity function cut along the time delay axis τ is obtained by setting $f_d = 0$. More precisely,

$$|\chi(\tau;0)|^2 = \left(1 - \frac{|\tau|}{\tau'}\right)^2 \qquad |\tau| \le \tau' \tag{4.11}$$

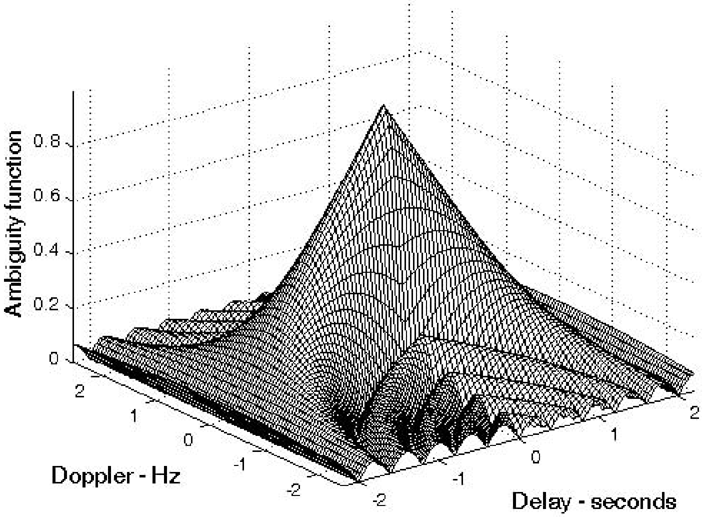


Figure 4.2a. Single pulse 3-D uncertainty plot. Pulsewidth is 2 seconds.

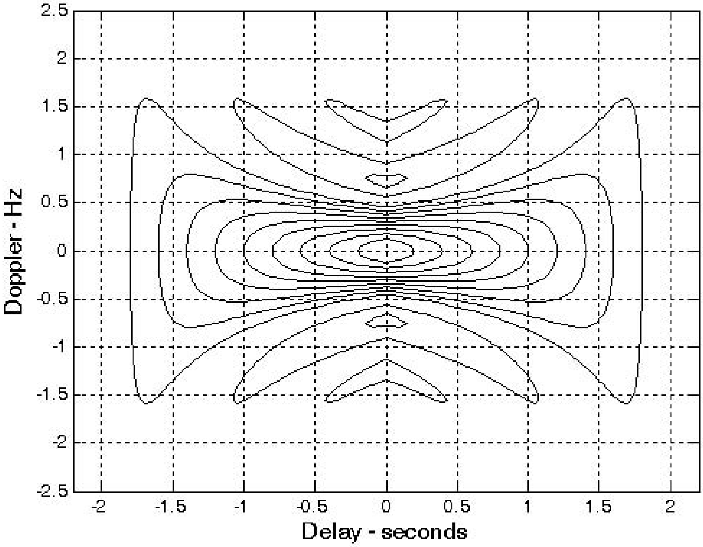


Figure 4.2b. Contour plot corresponding to Fig. 4.2a.

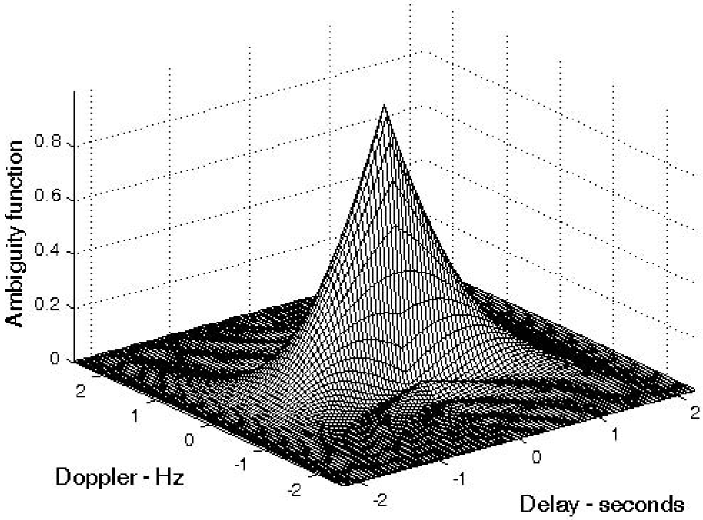


Figure 4.2c. Single pulse 3-D ambiguity plot. Pulsewidth is 2 seconds.

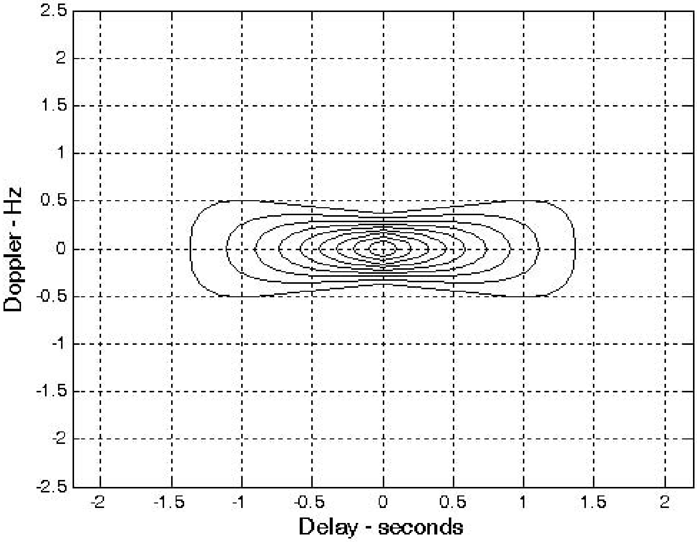


Figure 4.2d. Contour plot corresponding to Fig. 4.2c.

Note that the time autocorrelation function of the signal $s(t)$ is equal to $\chi(\tau;0)$. Similarly, the cut along the Doppler axis is

$$|\chi(0;f_d)|^2 = \left|\frac{\sin\pi\tau' f_d}{\pi\tau' f_d}\right|^2 \tag{4.12}$$

Figs. 4.3 and 4.4, respectively, show the plots of the uncertainty function cuts defined by Eqs. (4.11) and (4.12). Since the zero Doppler cut along the time delay axis extends between $-\tau'$ and τ', then, close targets would be unambiguous if they are at least τ' seconds apart.

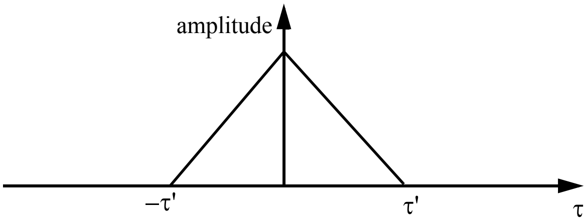


Figure 4.3. Zero Doppler uncertainty function cut along the time delay axis.

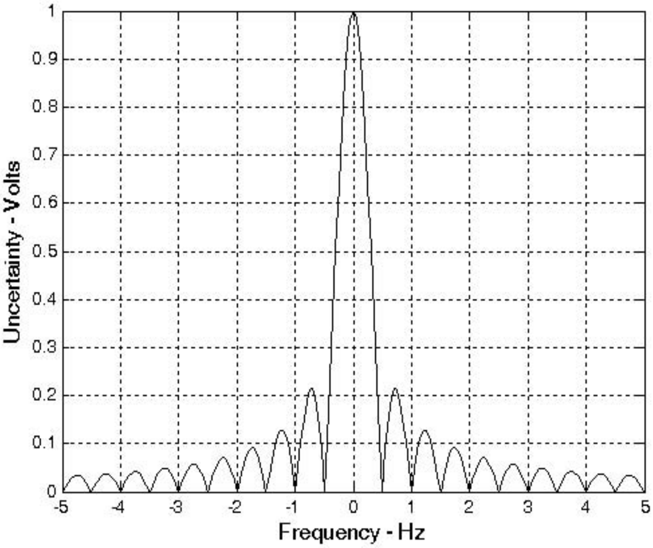


Figure 4.4. Uncertainty function of a single frequency pulse (zero delay). This plot can be reproduced using MATLAB program *"fig4_4.m"* given in Listing 4.3 in Section 4.6.

The zero time cut along the Doppler frequency axis has a $(\sin x/x)^2$ shape. It extends from $-\infty$ to ∞. The first null occurs at $f_d = \pm 1/\tau'$. Hence, it is possible to detect two targets that are shifted by $1/\tau'$, without any ambiguity.

We conclude that a single pulse range and Doppler resolutions are limited by the pulsewidth τ'. Fine range resolution requires that a very short pulse be used. Unfortunately, using very short pulses requires very large operating bandwidths, and may limit the radar average transmitted power to impractical values.

4.2.2. LFM Ambiguity Function

Consider the LFM complex envelope signal defined by

$$s(t) = \frac{1}{\sqrt{\tau'}} Rect\left(\frac{t}{\tau'}\right) \; e^{j\pi\mu t^2} \tag{4.13}$$

In order to compute the ambiguity function for the LFM complex envelope, we will first consider the case when $0 \le \tau \le \tau'$. In this case the integration limits are from $-\tau'/2$ to $(\tau'/2) - \tau$. Substituting Eq. (4.13) into Eq. (4.9) yields

$$\chi(\tau;f_d) = \frac{1}{\tau'} \int_{-\infty}^{\infty} Rect\left(\frac{t}{\tau'}\right) Rect\left(\frac{t-\tau}{\tau'}\right) e^{j\pi\mu t^2} e^{-j\pi\mu(t-\tau)^2} \; e^{j2\pi f_d t} dt \tag{4.14}$$

It follows that

$$\chi(\tau;f_d) = \frac{e^{-j\pi\mu\tau^2}}{\tau'} \int_{\frac{-\tau'}{2}}^{\frac{\tau'}{2}-\tau} e^{j2\pi(\mu\tau + f_d)t} dt \tag{4.15}$$

Finishing the integration process in Eq. (4.15) yields

$$\chi(\tau;f_d) = e^{j\pi\tau f_d}\left(1 - \frac{\tau}{\tau'}\right) \frac{\sin\left(\pi\tau'(\mu\tau + f_d)\left(1 - \frac{\tau}{\tau'}\right)\right)}{\pi\tau'(\mu\tau + f_d)\left(1 - \frac{\tau}{\tau'}\right)} \qquad 0 \le \tau \le \tau' \tag{4.16}$$

Similar analysis for the case when $-\tau' \le \tau \le 0$ can be carried out, where in this case the integration limits are from $(-\tau'/2) - \tau$ to $\tau'/2$. The same result can be obtained by using the symmetry property of the ambiguity function ($|\chi(-\tau, -f_d)| = |\chi(\tau, f_d)|$). It follows that an expression for $\chi(\tau;f_d)$ that is valid for any τ is given by

$$\chi(\tau;f_d) = e^{j\pi\tau f_d}\left(1-\frac{|\tau|}{\tau'}\right)\frac{\sin\left(\pi\tau'(\mu\tau+f_d)\left(1-\frac{|\tau|}{\tau'}\right)\right)}{\pi\tau'(\mu\tau+f_d)\left(1-\frac{|\tau|}{\tau'}\right)} \qquad |\tau| \le \tau' \tag{4.17}$$

and the LFM ambiguity function is

$$|\chi(\tau;f_d)|^2 = \left|\left(1-\frac{|\tau|}{\tau'}\right)\frac{\sin\left(\pi\tau'(\mu\tau+f_d)\left(1-\frac{|\tau|}{\tau'}\right)\right)}{\pi\tau'(\mu\tau+f_d)\left(1-\frac{|\tau|}{\tau'}\right)}\right|^2 \qquad |\tau| \le \tau' \tag{4.18}$$

Again the time autocorrelation function is equal to $\chi(\tau, 0)$. The reader can verify that the ambiguity function for a down-chirp LFM waveform is given by

$$|\chi(\tau;f_d)|^2 = \left|\left(1-\frac{|\tau|}{\tau'}\right)\frac{\sin\left(\pi\tau'(\mu\tau-f_d)\left(1-\frac{|\tau|}{\tau'}\right)\right)}{\pi\tau'(\mu\tau-f_d)\left(1-\frac{|\tau|}{\tau'}\right)}\right|^2 \qquad |\tau| \le \tau' \tag{4.19}$$

MATLAB Function “lfm_ambg.m”

The function *“lfm_ambg.m”* implements Eqs. (4.18) and (4.19). It is given in Listing 4.4 in Section 4.6. The syntax is as follows:

lfm_ambg [taup, b, up_down]

where

Symbol	Description	Units	Status
taup	*pulsewidth*	*seconds*	*input*
b	*bandwidth*	*Hz*	*input*
up_down	*up_down = 1 for up chirp* *up_down = -1 for down chirp*	*none*	*input*

Fig. 4.5 (a-d) shows 3-D and contour plots for the LFM uncertainty and ambiguity functions for

taup	*b*	*up_down*
1	*10*	*1*

These plots can be reproduced using MATLAB program *“fig4_5.m”* given in Listing 4.5 in Section 4.6. This function generates 3-D and contour plots of an LFM ambiguity function.

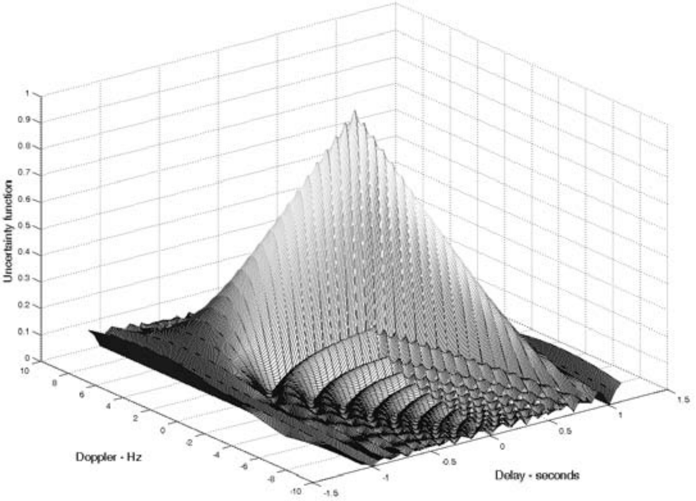


Figure 4.5a. Up-chirp LFM 3-D uncertainty plot. Pulsewidth is 1 second; and bandwidth is 10 Hz.

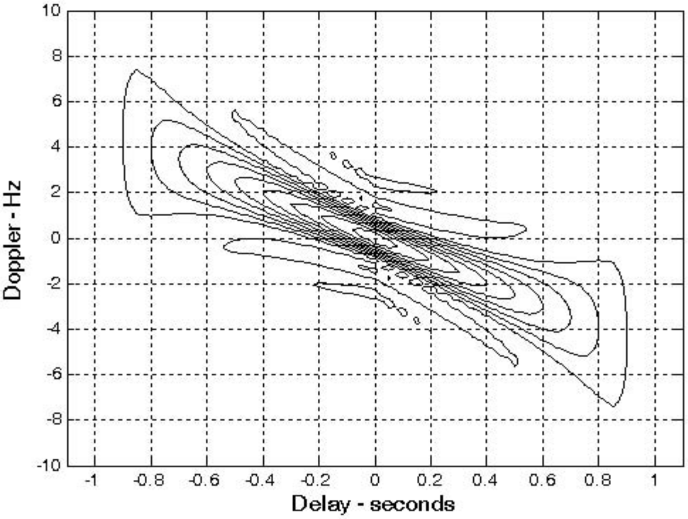


Figure 4.5b. Contour plot corresponding to Fig. 4.5a.

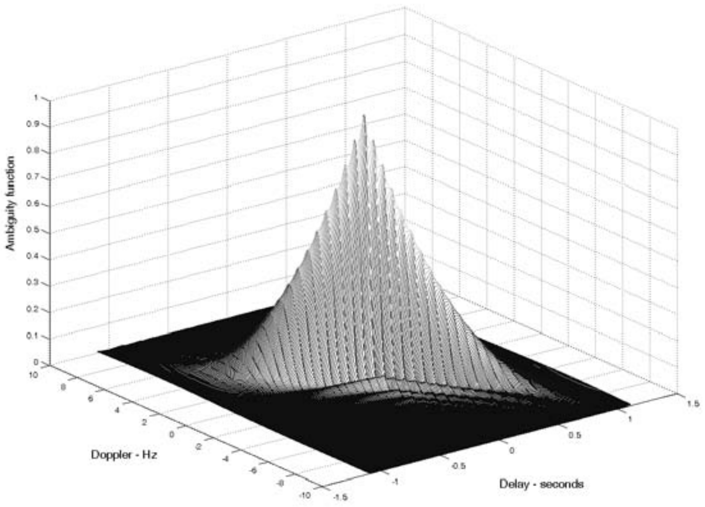


Figure 4.5c. Up-chirp LFM 3-D ambiguity plot. Pulsewidth is 1 second; and bandwidth is 10 Hz.

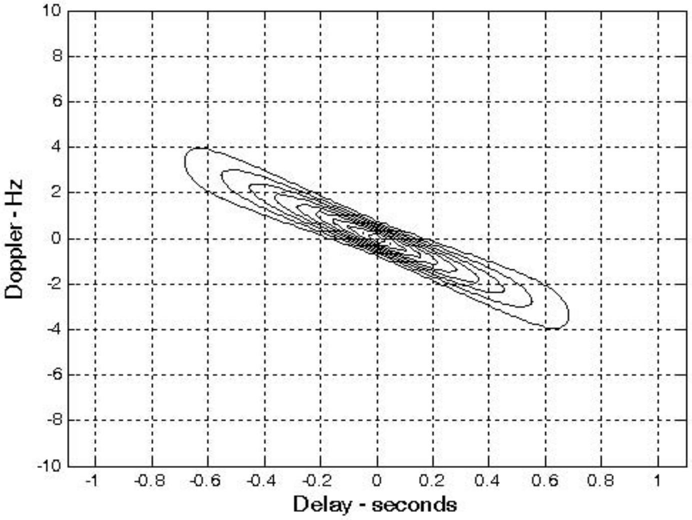


Figure 4.5d. Contour plot corresponding to Fig. 4.5c.

The up-chirp ambiguity function cut along the time delay axis τ is

$$|\chi(\tau;0)|^2 = \left|\left(1-\frac{|\tau|}{\tau'}\right)\frac{\sin\left(\pi\mu\tau\tau'\left(1-\frac{|\tau|}{\tau'}\right)\right)}{\pi\mu\tau\tau'\left(1-\frac{|\tau|}{\tau'}\right)}\right|^2 \qquad |\tau| \le \tau' \tag{4.20}$$

Fig. 4.6 shows a plot for a cut in the uncertainty function corresponding to Eq. (4.20). Note that the LFM ambiguity function cut along the Doppler frequency axis is similar to that of the single pulse. This should not be surprising since the pulse shape has not changed (we only added frequency modulation). However, the cut along the time delay axis changes significantly. It is now much narrower compared to the unmodulated pulse cut. In this case, the first null occurs at

$$\tau_{n1} \approx 1/B \tag{4.21}$$

which indicates that the effective pulsewidth (compressed pulsewidth) of the matched filter output is completely determined by the radar bandwidth. It follows that the LFM ambiguity function cut along the time delay axis is narrower than that of the unmodulated pulse by a factor

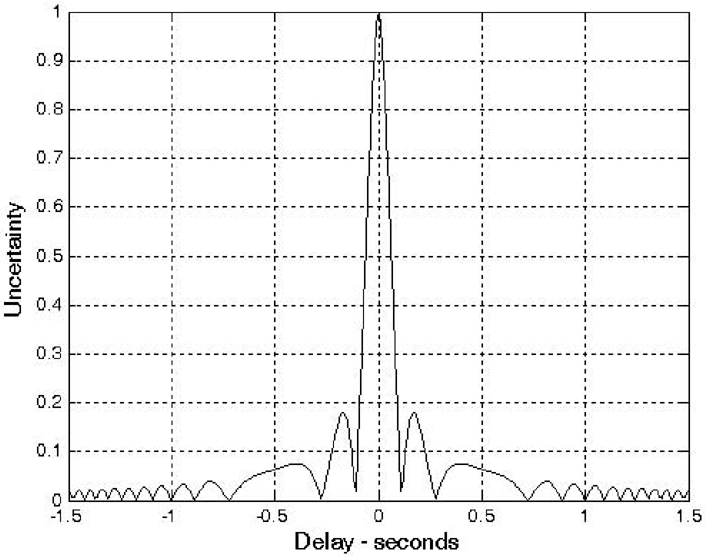


Figure 4.6. Zero Doppler uncertainty of an LFM pulse ($\tau' = 1$, $b = 20$). This plot can be reproduced using MATLAB program *"fig4_6.m"* given in Listing 4.6 in Section 4.6.

$$\xi = \frac{\tau'}{(1/B)} = \tau' B \tag{4.22}$$

ξ is referred to as the compression ratio (also called time-bandwidth product and compression gain). All three names can be used interchangeably to mean the same thing. As indicated by Eq. (4.22) the compression ratio also increases as the radar bandwidth is increased.

Example:

Compute the range resolution before and after pulse compression corresponding to an LFM waveform with the following specifications: Bandwidth $B = 1GHz$*; and pulsewidth* $\tau' = 10ms$.

Solution:

The range resolution before pulse compression is

$$\Delta R_{uncomp} = \frac{c\tau'}{2} = \frac{3 \times 10^8 \times 10 \times 10^{-3}}{2} = 1.5 \times 10^6 \ \ meters$$

Using Eq. (4.21) yields

$$\tau_{n1} = \frac{1}{1 \times 10^9} = 1 \ \ ns$$

$$\Delta R_{comp} = \frac{c\tau_{n1}}{2} = \frac{3 \times 10^8 \times 1 \times 10^{-9}}{2} = 15 \ \ cm .$$

4.2.3. Coherent Pulse Train Ambiguity Function

Fig. 4.7 shows a plot of a coherent pulse train. The pulsewidth is denoted as τ' and the PRI is T. The number of pulses in the train is N; hence, the train's length is $(N-1)T$ seconds. A normalized individual pulse $s(t)$ is defined by

$$s_1(t) = \frac{1}{\sqrt{\tau'}} Rect\left(\frac{t}{\tau'}\right) \tag{4.23}$$

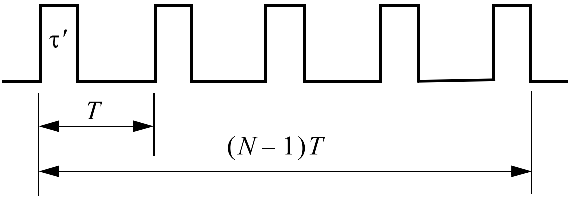


Figure 4.7. Coherent pulse train. N=5.

When coherency is maintained between the consecutive pulses, then an expression for the normalized train is

$$s(t) = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} s_1(t - iT) \tag{4.24}$$

The output of the matched filter is

$$\chi(\tau;f_d) = \int_{-\infty}^{\infty} s(t) s^*(t+\tau) e^{j2\pi f_d t} dt \tag{4.25}$$

Substituting Eq. (4.24) into Eq. (4.25) and interchanging the summations and integration yield

$$\chi(\tau;f_d) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \int_{-\infty}^{\infty} s_1(t - iT)\ \ s_1^*(t - jT - \tau) e^{j2\pi f_d t} dt \tag{4.26}$$

Making the change of variable $t_1 = t - iT$ yields

$$\chi(\tau;f_d) = \frac{1}{N} \sum_{i=0}^{N-1} e^{j2\pi f_d iT} \sum_{j=0}^{N-1} \int_{-\infty}^{\infty} s_1(t_1)\ \ s_1^*(t_1 - [\tau - (i-j)T]) e^{j2\pi f_d t_1} dt_1 \tag{4.27}$$

The integral inside Eq. (4.27) represents the output of the matched filter for a single pulse, and is denoted by χ_1. It follows that

$$\chi(\tau;f_d) = \frac{1}{N} \sum_{i=0}^{N-1} e^{j2\pi f_d iT} \sum_{j=0}^{N-1} \chi_1[\tau - (i-j)T;f_d] \tag{4.28}$$

When the relation $q = i - j$ is used, then the following relation is true:[1]

$$\sum_{i=0}^{N} \sum_{m=0}^{N} = \sum_{q=-(N-1)}^{0} \sum_{i=0}^{N-1-|q|} \Bigg|_{for\ \ j = i - q} + \sum_{q=1}^{N-1} \sum_{j=0}^{N-1-|q|} \Bigg|_{for\ \ i = j + q} \tag{4.29}$$

Using Eq. (4.29) into Eq. (4.28) gives

[1] Rihaczek, A. W., *Principles of High Resolution Radar*, Artech House, 1994.

$$\chi(\tau;f_d) = \frac{1}{N}\sum_{q=-(N-1)}^{0}\left\{\chi_1(\tau-qT;f_d)\sum_{i=0}^{N-1-|q|}e^{j2\pi f_d iT}\right\} \tag{4.30}$$

$$+\frac{1}{N}\sum_{q=1}^{N-1}\left\{e^{j2\pi f_d qT}\chi_1(\tau-qT;f_d)\sum_{j=0}^{N-1-|q|}e^{j2\pi f_d jT}\right\}$$

Setting $z = \exp(j2\pi f_d T)$, and using the relation

$$\sum_{j=0}^{N-1-|q|} z^j = \frac{1-z^{N-|q|}}{1-z} \tag{4.31}$$

yield

$$\sum_{i=0}^{N-1-|q|} e^{j2\pi f_d iT} = e^{[j\pi f_d(N-1-|q|T)]}\ \frac{\sin[\pi f_d(N-1-|q|T)]}{\sin(\pi f_d T)} \tag{4.32}$$

Using Eq. (4.32) in Eq. (4.30) yields two complementary sums for positive and negative q. Both sums can be combined as

$$\chi(\tau;f_d) = \frac{1}{N}\sum_{q=-(N-1)}^{N-1}\chi_1(\tau-qT;f_d)e^{[j\pi f_d(N-1+q)T]}\ \frac{\sin[\pi f_d(N-|q|T)]}{\sin(\pi f_d T)} \tag{4.33}$$

Finally, the ambiguity function associated with the coherent pulse train is computed as the modulus square of Eq. (4.33). For $\tau' < T/2$, the ambiguity function reduces to

$$\chi(\tau;f_d) = \frac{1}{N}\sum_{q=-(N-1)}^{N-1}|\chi_1(\tau-qT;f_d)|\ \left|\frac{\sin[\pi f_d(N-|q|T)]}{\sin(\pi f_d T)}\right| \tag{4.34}$$

Thus, the ambiguity function for a coherent pulse train is the superposition of the individual pulse's ambiguity functions. The ambiguity function cuts along the time delay and Doppler axes are, respectively, given by

$$|\chi(\tau;0)|^2 = \left|\sum_{q=-(N-1)}^{N-1}\left(1-\frac{|q|}{N}\right)\left(1-\frac{|\tau-qT|}{\tau'}\right)\right|^2 \quad ; \ |\tau-qT|<\tau' \tag{4.35}$$

$$|\chi(0;f_d)|^2 = \left|\frac{1}{N}\frac{\sin(\pi f_d\tau')}{\pi f_d\tau'}\;\frac{\sin(\pi f_d NT)}{\sin(\pi f_d T)}\right|^2 \tag{4.36}$$

MATLAB Function "train_ambg.m"

The function *"train_ambg.m"* implements Eq. (4.34). It is given in Listing 4.7 in Section 4.6. The syntax is as follows:

train_ambg [taup, n, pri]

Symbol	Description	Units	Status
taup	*pulsewidth*	*seconds*	*input*
n	*number of pulses in train*	*none*	*input*
pri	*pulse repetition interval*	*seconds*	*input*

Fig. 4.8 (a-d) shows typical outputs of this function, for

taup	*n*	*pri*
0.2	*5*	*1*

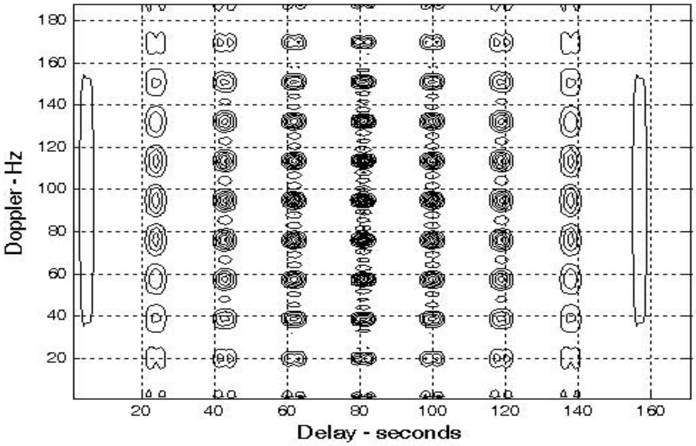


Figure 4.8a. Three-dimensional ambiguity plot for a five pulse equal amplitude coherent train. Pulsewidth is 0.2 seconds; and PRI is 1 second, N=5. This plot can be reproduced using MATLAB program *"fig4_8.m"* given in Listing 4.8 in Section 4.6.

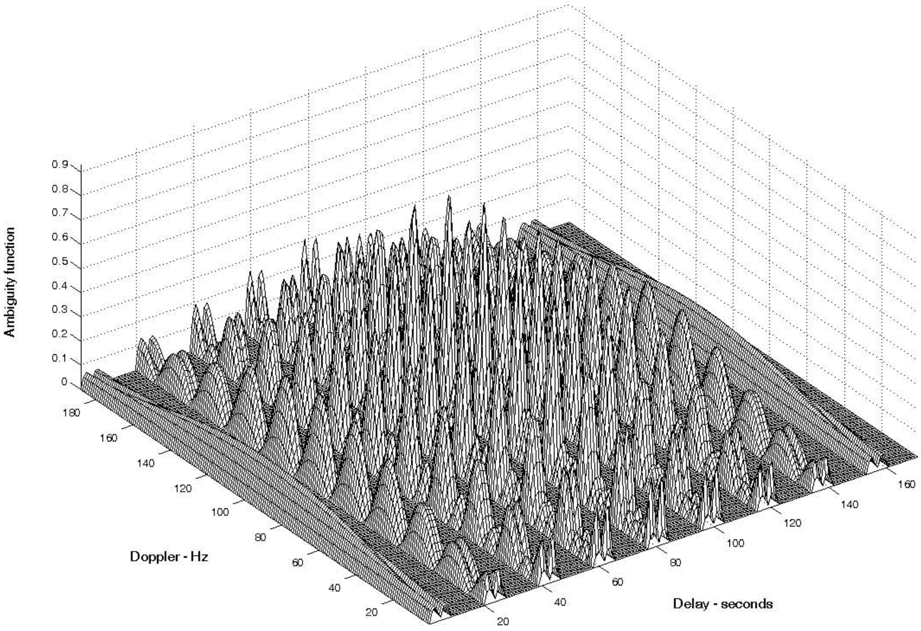


Figure 4.8b. 3-D plot corresponding to Fig. 4.8a.

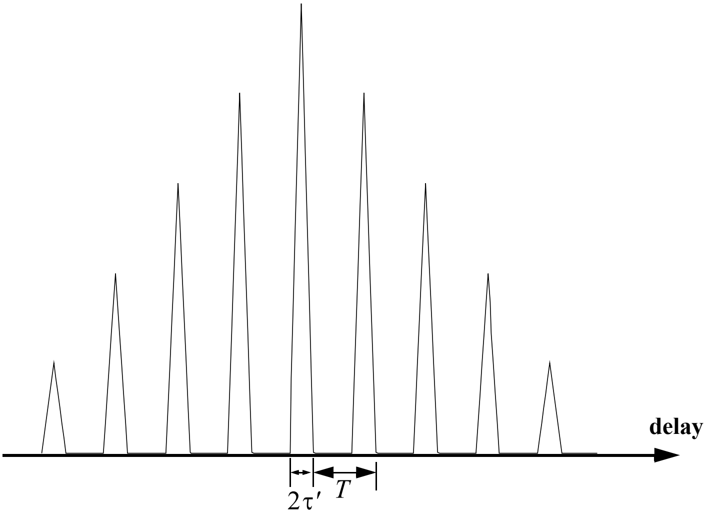


Figure 4.8c. Zero Doppler cut corresponding to Fig. 4.8a.

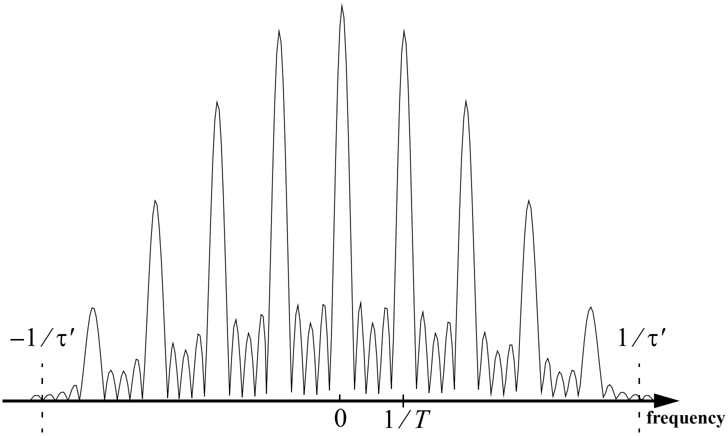


Figure 4.8d. Zero delay cut corresponding to Fig. 4.8a.

4.3. Ambiguity Diagram Contours

Plots of the ambiguity function are called ambiguity diagrams. For a given waveform, the corresponding ambiguity diagram is normally used to determine the waveform properties such as the target resolution capability, measurement (time and frequency) accuracy, and its response to clutter. Three-dimensional ambiguity diagrams are difficult to plot and interpret. This is the reason why contour plots of the 3-D ambiguity diagram are often used to study the characteristics of a waveform. An ambiguity contour is a 2-D plot (frequency/time) of a plane intersecting the 3-D ambiguity diagram that corresponds to some threshold value. The resultant plots are ellipses. It is customary to display the ambiguity contour plots that correspond to one half of the peak autocorrelation value.

Fig. 4.9 shows a sketch of typical ambiguity contour plots associated with a gated CW pulse. It indicates that narrow pulses provide better range accuracy than long pulses. Alternatively, the Doppler accuracy is better for a wider pulse than it is for a short one. This trade-off between range and Doppler measurements comes from the uncertainty associated with the time-bandwidth product of a single sinusoidal pulse, where the product of uncertainty in time (range) and uncertainty in frequency (Doppler) cannot be much smaller than unity. Note that an exact plot for Fig. 4.9 can be obtained using the function *"single_pulse_ambg.m"* and the MATLAB command *contour*.

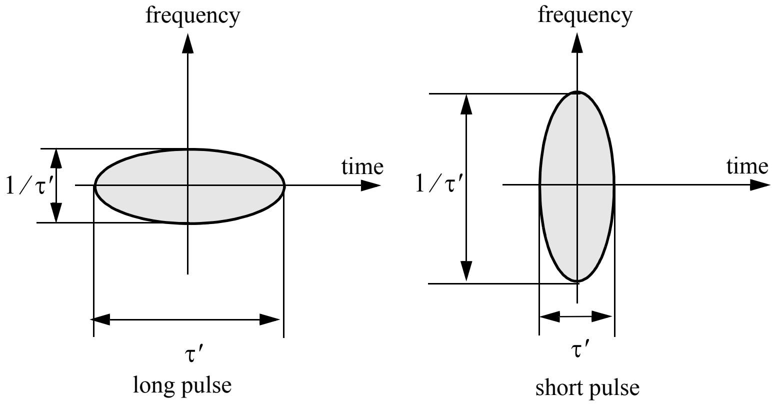


Figure 4.9. Ambiguity contour plot associated with a sinusoid modulated gated CW pulse. See Fig. 4.2.

Multiple ellipses in an ambiguity contour plot indicate the presence of multiple targets. Thus, it seems that one may improve the radar resolution by increasing the ambiguity diagram threshold value. This is illustrated in Fig.

4.10. However, in practice this is not possible for two reasons. First, in the presence of noise we lack knowledge of the peak correlation value; and second, targets in general will have different amplitudes.

Now consider the case of a coherent pulse train described in Fig. 4.7. For a pulse train, range accuracy is still determined by the pulsewidth, in the same manner as in the case of a single pulse, while Doppler accuracy is determined by the train length. Thus, time and frequency measurements can be made independently of each other. However, additional peaks appear in the ambiguity diagram which may cause range and Doppler uncertainties (see Fig. 4.11).

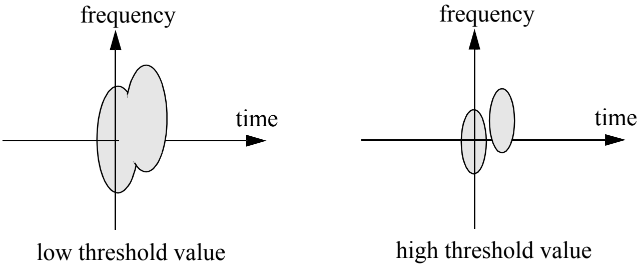


Figure 4.10. Effect of threshold value on resolution.

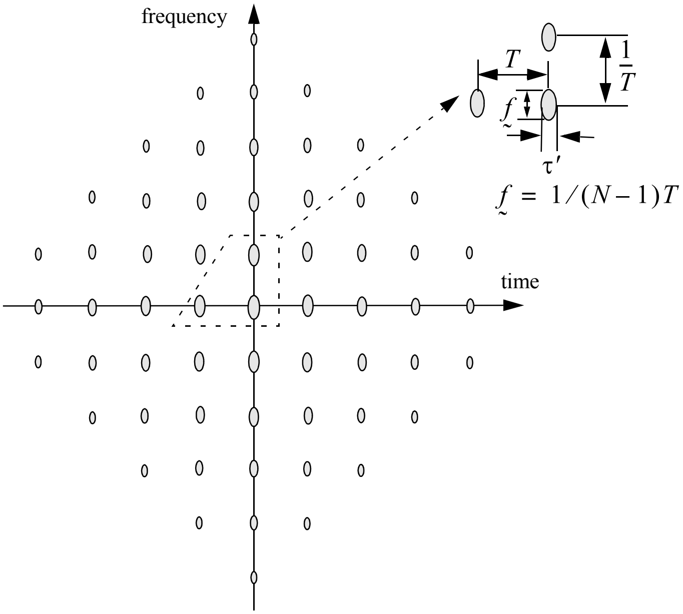


Figure 4.11. Ambiguity contour plot corresponding to Fig. 4.7. For an exact plot see Fig. 4.8a.

As one would expect, high PRF pulse trains (i.e., small T) lead to extreme uncertainty in range, while low PRF pulse trains have extreme ambiguity in Doppler. Medium PRF pulse trains have moderate ambiguity in both range and Doppler, which can be overcome by using multiple PRFs. It is possible to avoid ambiguities caused by pulse trains and still have reasonable independent control on both range and Doppler accuracies by using a single modulated pulse with a time-bandwidth product that is much larger than unity. Fig. 4.12 shows the ambiguity contour plot associated with an LFM waveform. In this case, τ' is the pulsewidth and B is the pulse bandwidth. The exact plots can be obtained using the function *"lfm_ambg.m"*.

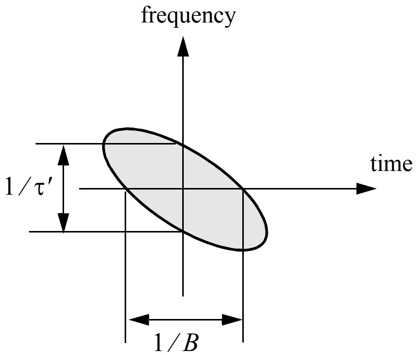


Figure 4.12. Ambiguity contour plot associated with an up-chirp LFM waveform. For an exact plot see Fig. 4.5b.

4.4. Digital Coded Waveforms

In this section we will briefly discuss the digital coded waveform. We will determine the waveform range and Doppler characteristics on the basis of its autocorrelation function, since in the absence of noise, the output of the matched filter is proportional to the code autocorrelation.

4.4.1. Frequency Coding (Costas Codes)

Construction of Costas codes can be understood from the construction process of Stepped Frequency Waveforms (SFW) described in Chapter 3. In SFW, a relatively long pulse of length τ' is divided into N subpulses, each of width τ_1 ($\tau' = N\tau_1$). 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,

$$\tau_1 = \tau'/N \qquad (4.37)$$

and the frequency for the *ith* subpulse is

$$f_i = f_0 + i\Delta f \qquad ; \; i = 1, N \qquad (4.38)$$

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

$$\Delta f \tau' = N^2 \qquad (4.39)$$

Costas 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. 4.13b. 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. 4.13a shows the frequency assignment associated with a SFW. Alternatively, the frequency assignments in Fig. 4.13b 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).

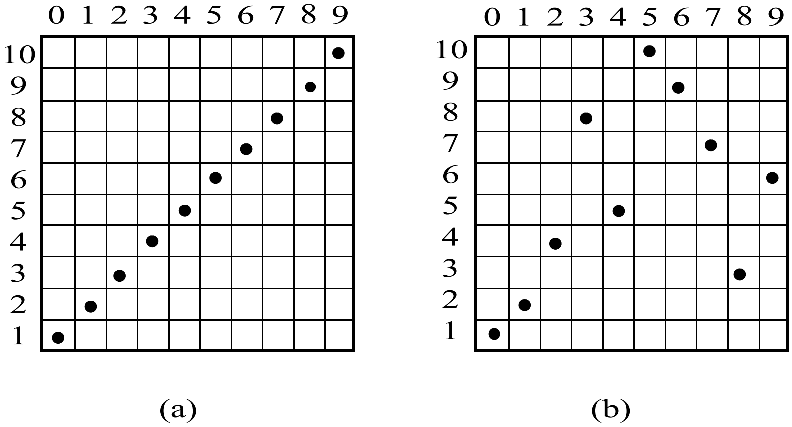


Figure 4.13. Frequency assignment for a burst of N subpulses. (a) SFW (stepped LFM); (b) Costas code of length Nc = 10.

The sequences of *"dot"* assignments for which the corresponding ambiguity function approaches an ideal or a *"thumbtack"* response are called Costas codes. A near thumbtack response was obtained by Costas[1] 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.

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

$$N = q - 1 \tag{4.40}$$

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

$$\begin{aligned} i &= 0, 1, 2, \ldots, (q-2) \\ j &= 1, 2, 3, \ldots, (q-1) \end{aligned} \tag{4.41}$$

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

$$i = (\gamma)^j \ (modulo \ \ q) \tag{4.42}$$

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

$$s(t) = \frac{1}{\sqrt{N\tau_1}} \sum_{l=0}^{N-1} s_l(t - l\tau_1) \tag{4.43}$$

$$s_l(t) = \begin{pmatrix} \exp(j2\pi f_l t) & 0 \le t \le \tau_1 \\ 0 & elsewhere \end{pmatrix} \tag{4.44}$$

[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.

Costas showed that the output of the matched filter is

$$\chi(\tau, f_D) = \frac{1}{N}\sum_{l=0}^{N-1} \exp(j2\pi l f_D \tau)\left\{ \Phi_{ll}(\tau, f_D) + \sum_{\substack{q=0 \\ q \neq l}}^{N-1} \Phi_{lq}(\tau - (l-q)\tau_1, f_D) \right\} \tag{4.45}$$

$$\Phi_{lq}(\tau, f_D) = \left(\tau_1 - \frac{|\tau|}{\tau_1}\right)\frac{\sin\alpha}{\alpha} \ \exp(-j\beta - j2\pi f_q \tau) \qquad , \ |\tau| \le \tau_1 \tag{4.46}$$

$$\alpha = \pi(f_l - f_q - f_D)(\tau_1 - |\tau|) \tag{4.47}$$

$$\beta = \pi(f_l - f_q - f_D)(\tau_1 + |\tau|) \tag{4.48}$$

Three-dimensional plots of the ambiguity function of Costas signals show the near thumbtack response of the ambiguity function. All sidelobes, except for 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.

4.4.2. Binary Phase Codes

Consider the case of binary phase codes in which a relatively long pulse of width τ' is divided into N smaller pulses; each is of width $\Delta\tau = \tau'/N$. Then, the phase of each sub-pulse is randomly chosen as either 0 or π radians relative to some CW reference signal. It is customary to characterize a sub-pulse that has 0 phase (amplitude of +1 Volt) as either "1" or "+." Alternatively, a sub-pulse with phase equal to π (amplitude of -1 Volt) is characterized by either "0" or "-." The compression ratio associated with binary phase codes is equal to $\xi = \tau'/\Delta\tau$, and the peak value is N times larger than that of the long pulse. The goodness of a compressed binary phase code waveform depends heavily on the random sequence of the phases of the individual sub-pulses.

One family of binary phase codes that produces compressed waveforms with constant sidelobe levels equal to unity is the Barker code. Fig. 4.14 illustrates a Barker code of length seven. A Barker code of length n is denoted as B_n. There are only seven known Barker codes that share this unique property; they are listed in Table 4.1. Note that B_2 and B_4 have complementary forms that have the same characteristics. Since there are only seven Barker codes, they are not used when radar security is an issue.

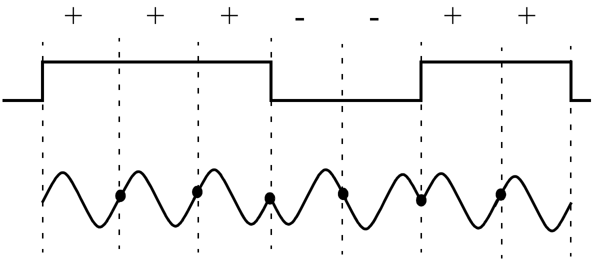


Figure 4.14. Binary phase code of length 7.

TABLE 4.1. Barker codes.

Code symbol	**Code length**	**Code elements**	**Side lode reduction (dB)**
B_2	*2*	*1 -1* *1 1*	*6.0*
B_3	*3*	*1 1 -1*	*9.5*
B_4	*4*	*1 1 -1 1* *1 1 1 -1*	*12.0*
B_5	*5*	*1 1 1 -1 1*	*14.0*
B_7	*7*	*1 1 1 -1 -1 1 -1*	*16.9*
B_{11}	*11*	*1 1 1 -1 -1 -1 1 -1 -1 1 -1*	*20.8*
B_{13}	*13*	*1 1 1 1 1 -1 -1 1 1 -1 1 -1 1*	*22.3*

In general, the autocorrelation function (which is an approximation of the matched filter output) for a B_N Barker code will be $2N\Delta\tau$ wide. The main lobe is $2\Delta\tau$ wide; the peak value is equal to N. There are $(N-1)/2$ sidelobes on either side of the main lobe. This is illustrated in Fig. 4.15 for a B_{13}. Notice that the main lobe is equal to 13, while all sidelobes are unity.

The most sidelobe reduction offered by a Barker code is $-22.3dB$, which may not be sufficient for the desired radar application. However, Barker codes can be combined to generate much longer codes. In this case, a B_m code can be used within a B_n code (m within n) to generate a code of length mn. The compression ratio for the combined B_{mn} code is equal to mn. As an example, a combined B_{54} is given by

$$B_{54} = \{11101, 11101, 00010, 11101\} \tag{4.49}$$

and is illustrated in Fig. 4.16. Unfortunately, the sidelobes of a combined Barker code autocorrelation function are no longer equal to unity.

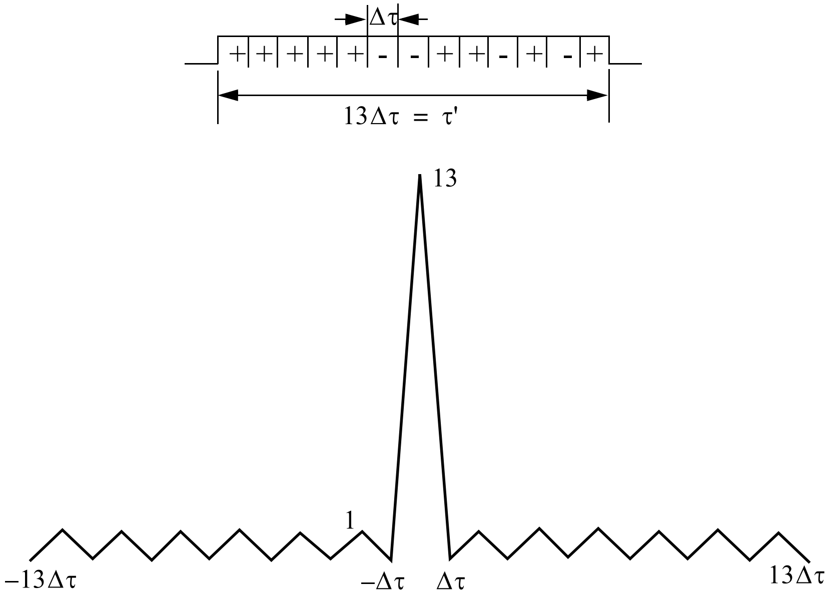


Figure 4.15. Barker code of length 13, and its corresponding autocorrelation function.

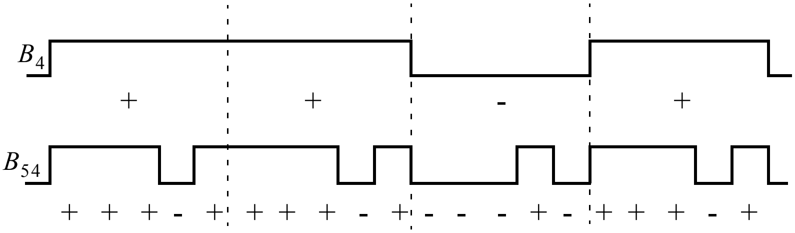


Figure 4.16. A combined B_{54} Barker code.

MATLAB Function "barker_ambig.m"

The MATLAB function *"barker_ambig.m"* calculates and plots the ambiguity function for Barker code. It is given in Listing 4.9 in Section 4.6. The syntax as follows:

$$[ambiguity] = barker\_ambig(u)$$

where u is a vector that defines the input code in terms of *"1's"* and *"-1's."* For example, using $u = [1 \ \ 1 \ \ 1 \ \ -1 \ \ -1 \ \ 1 \ \ -1]$ as an input, the function will calculate and plot the ambiguity function corresponding to B_7. Fig. 4.17 shows the output of this function when B_{13} is used as an input. Fig. 4.18 is similar to Fig. 4.17, except in this case B_7 is used as an input.

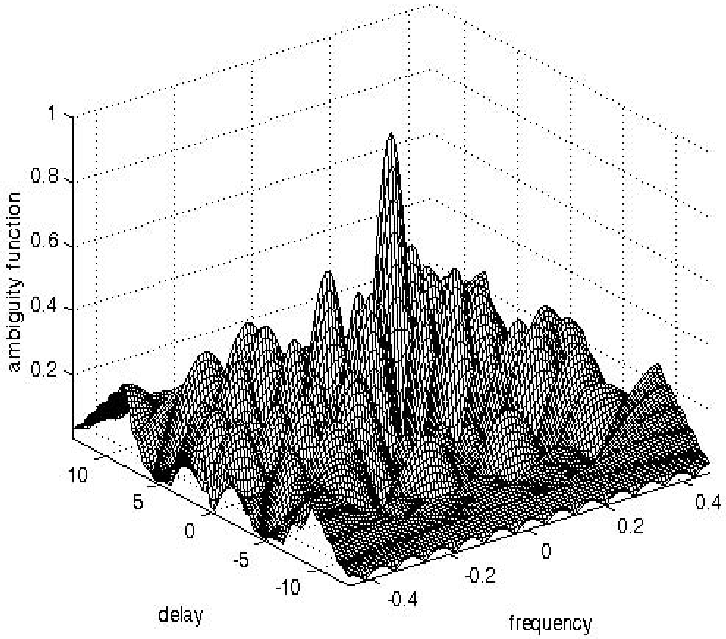


Figure 4.17a. Ambiguity function for B_{13} Barker code.

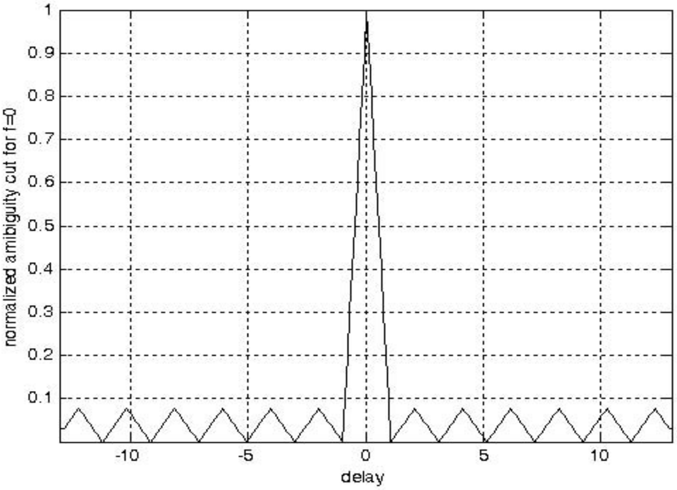


Figure 4.17b. Zero Doppler cut for the B_{13} ambiguity function.

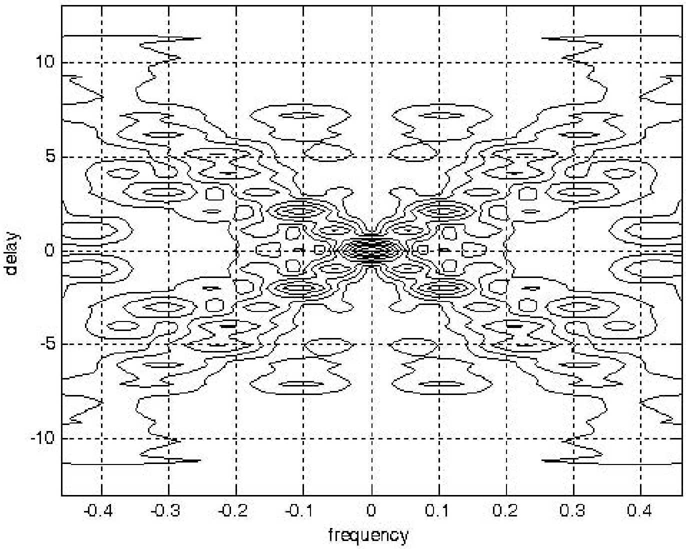


Figure 4.17c. Contour plot corresponding to Fig. 4.17a.

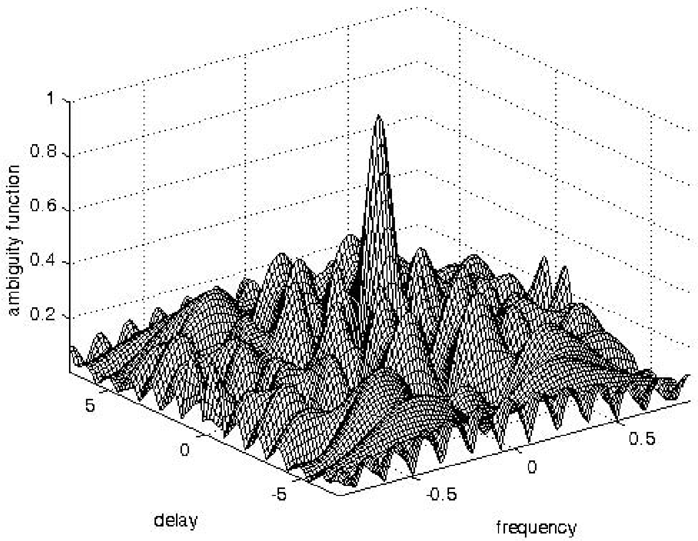


Figure 4.18a. Ambiguity function for B_7 Barker code.

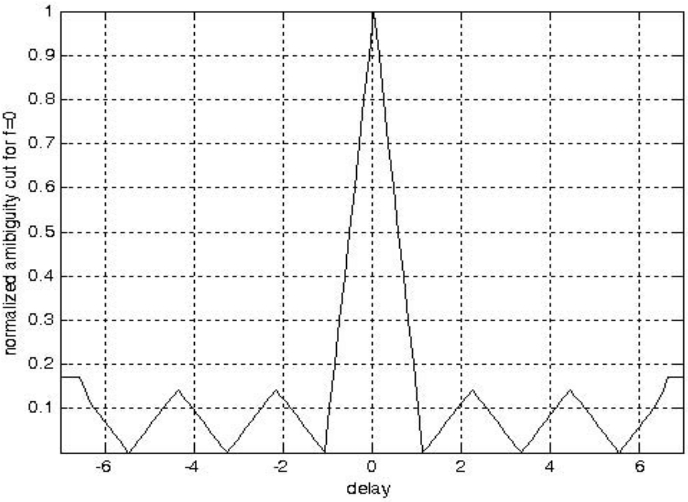


Figure 4.18b. Zero Doppler cut for the B_7 ambiguity function.

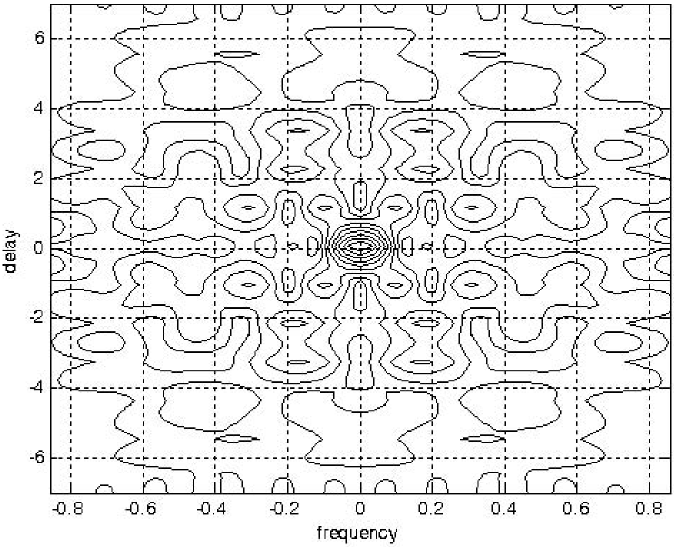


Figure 4.18c. Contour plot corresponding to Fig. 4.18a.

4.4.3. 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 that 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

$$\phi(n) = \begin{Bmatrix} L & n = 0, \pm L, \pm 2L, \ldots \\ -1 & elsewhere \end{Bmatrix} \tag{4.50}$$

Fig. 4.19 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.

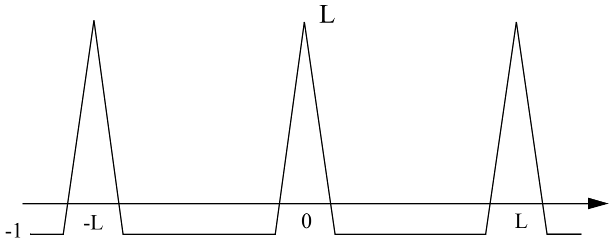


Figure 4.19. Typical autocorrelation of an MLS code of length L.

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

$$L = 2^n - 1 \tag{4.51}$$

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. 4.20 for $n = 4$ (i.e., $L = 15$). Note that the shift register initial state cannot be "zero."

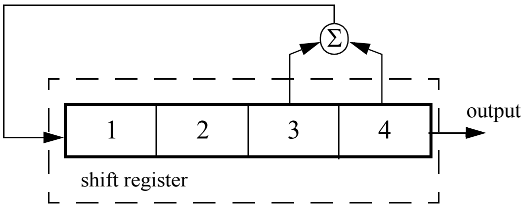


Figure 4.20. Circuit for generating an MLS sequence of length $L = 15$.

The feedback connections associated with a shift register generator determine whether the output sequence will be maximal or not. For a given size shift register, only 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. 4.21. The shift register generator shown in Fig. 4.21a generates a maximal length sequence, as clearly depicted by its state diagram. However, the shift register generator shown in Fig. 4.21b produces three non-maximal length sequences (depending on the initial state).

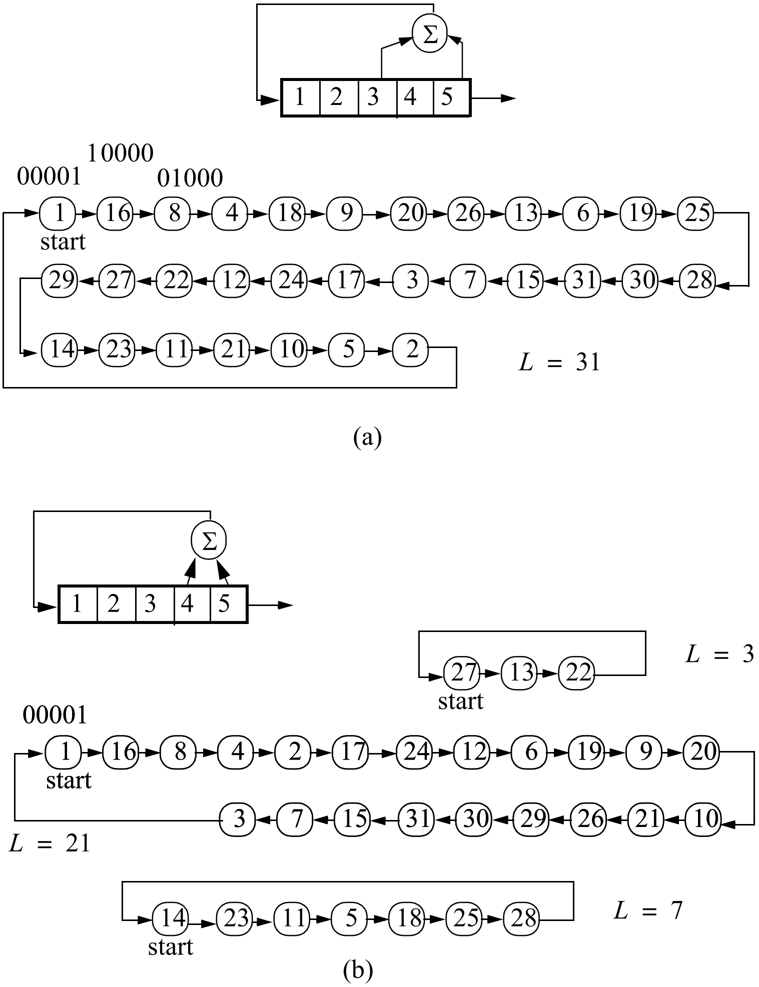


Figure 4.21. (a) A 5-stage shift register generator. (b) Non-maximal length 5 stage shift register generator.

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

$$N_L = \frac{\varphi(2^n - 1)}{n} \tag{4.52}$$

where φ is the Euler's totient (or Euler's phi) function. Euler's phi function is defined by

$$\varphi(k) = k \prod_i \frac{(p_i - 1)}{p_i} \tag{4.53}$$

where p_i are the prime factors of k. Note that when p_i has multiples, then only one of them is used (see example in Eq. (4.56)). Also note that when k is a prime number, then the Euler's phi function is

$$\varphi(k) = k - 1 \tag{4.54}$$

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

$$N_L = \frac{\varphi(2^3 - 1)}{3} = \frac{\varphi(7)}{3} = \frac{7-1}{3} = 2 \tag{4.55}$$

and a 6-stage shift register,

$$N_L = \frac{\varphi(2^6 - 1)}{6} = \frac{\varphi(63)}{6} = \frac{63}{6} \times \frac{(3-1)}{3} \times \frac{(7-1)}{7} = 6 \tag{4.56}$$

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

$$x^n + x^k + x^m + \ldots + 1 \tag{4.57}$$

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

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

$$x^6 + x^5 + 1 \tag{4.58}$$

It follows that the shift register which generates a maximal length sequence is shown in Fig. 4.22.

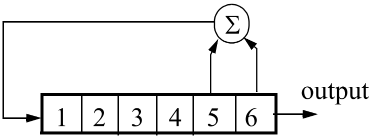


Figure 4.22. Linear shift register whose characteristic polynomial is $x^6 + x^5 + 1$**.**

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.

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 $2^n - 1$.

MATLAB Function "prn_ambig.m"

The MATLAB function *"prn_ambig.m"* calculates and plots the ambiguity function associated with a given PRN code. It is given in Listing 4.10 in Section 4.6. The syntax is as follows:

[ambiguity] = prn_ambig(u)

where u is a vector that defines the input maximal length code (sequence) in terms of *"1's"* and *"-1's."* Fig. 4.23 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]

Fig. 4.24 is similar to Fig. 4.23, 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]

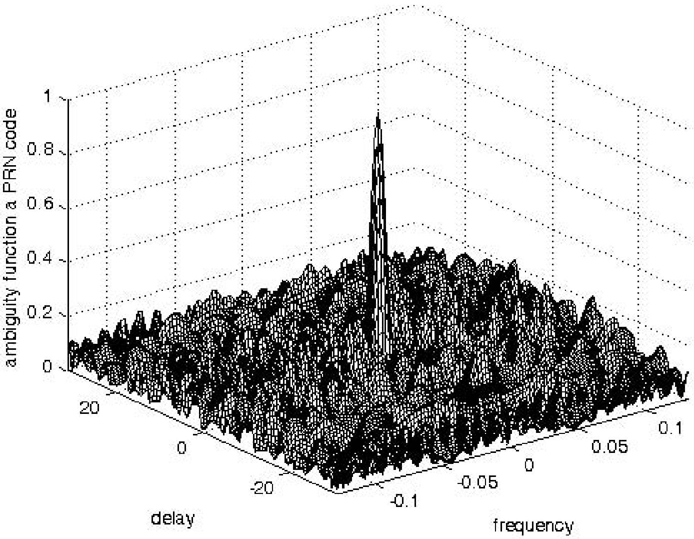


Figure 4.23a. Ambiguity function corresponding to a 31-bit PRN code.

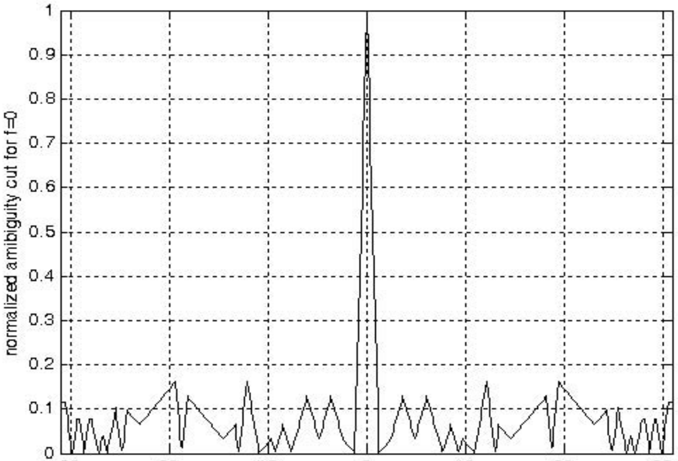


Figure 4.23b. Zero Doppler cut corresponding to Fig. 4.23a.

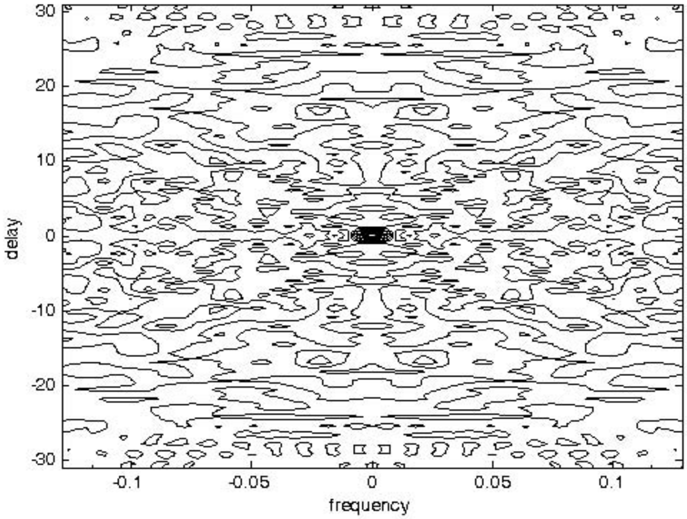


Figure 4.23c. Contour plot corresponding to Fig. 4.23a.

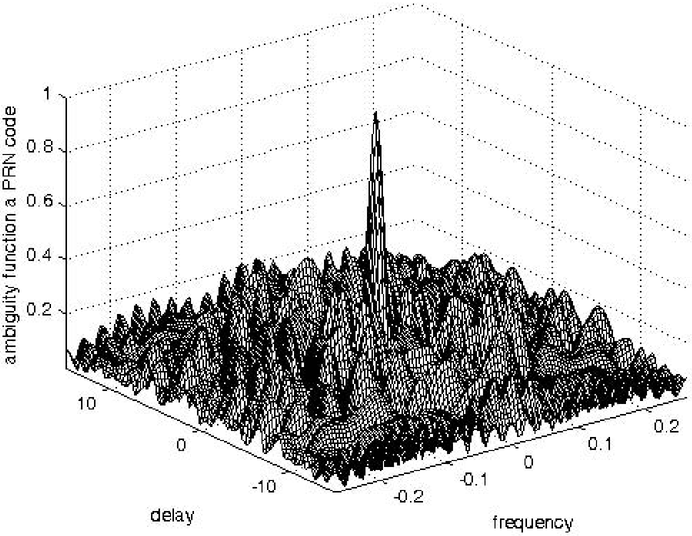


Figure 4.24a. Ambiguity function corresponding to a 15-bit PRN code.

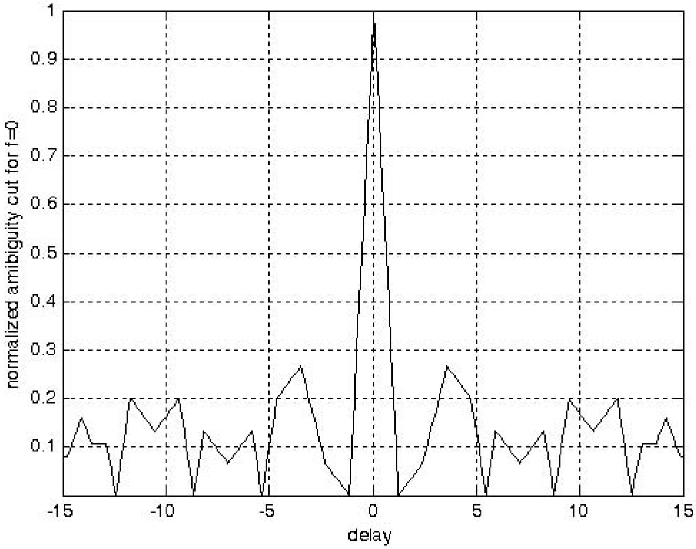


Figure 4.24b. Zero Doppler cut corresponding to Fig. 4.24a.

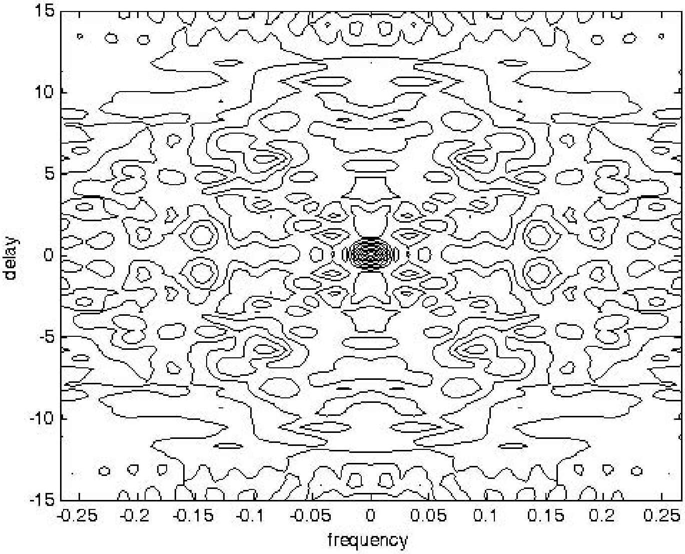


Figure 4.24c. Contour plot corresponding to Fig. 4.24a.

4.5. "MyRadar" Design Case Study - Visit 4

4.5.1. Problem Statement

Generate the ambiguity plots for the waveforms selected in Chapter 3 for this design case study.

4.5.2. A Design

In this section we will show the 3-D ambiguity diagram and the corresponding contour plot for only the search waveform. The user is advised to do the same for the track waveforms. For this purpose, use the MATLAB program *"myradar_visit4.m"*. It is given in Listing 4.11 in Section 4.6.

Figs. 4.25 and 4.26 show the output figures produced by the program *"myradar_visit4.m"* that correspond to the search waveform.

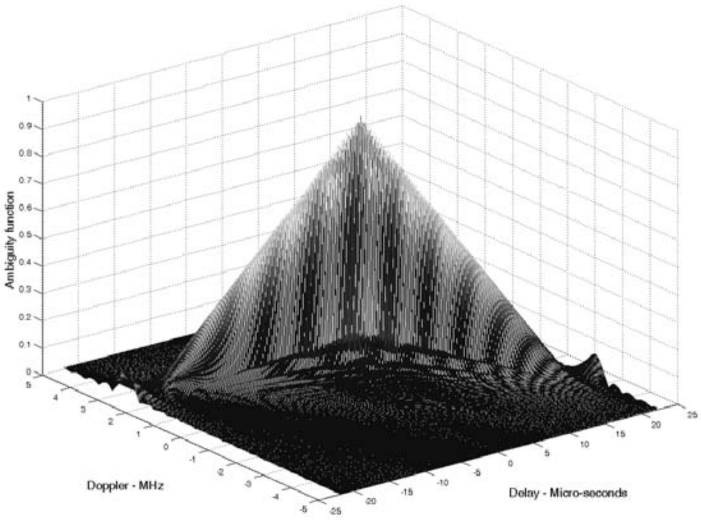


Figure 4.25. Ambiguity plot for *"MyRadar"* search waveform.

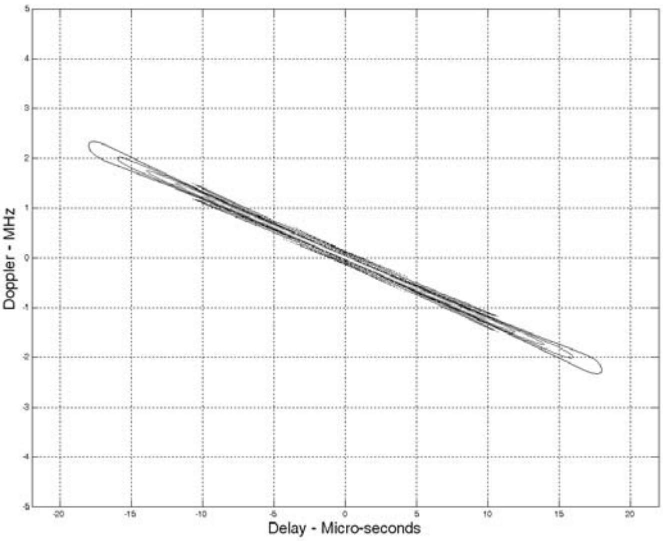


Figure 4.26. Contour of the ambiguity plot for *"MyRadar"* search waveform.

4.6. MATLAB Program and Function Listings

This section presents listings for all MATLAB programs/functions used in this chapter. The user is strongly advised to rerun the MATLAB programs in order to enhance his understanding of this chapter's material.

Listing 4.1. MATLAB Function "single_pulse_ambg.m"

```
function x = single_pulse_ambg (taup)
colormap (gray(1))
eps = 0.000001;
i = 0;
taumax = 1.1 * taup;
taumin = -taumax;
for tau = taumin:.05:taumax
  i = i + 1;
  j = 0;
  for fd = -5/taup:.05:5/taup %-2.5:.05:2.5
    j = j + 1;
    val1 = 1. - abs(tau) / taup;
    val2 = pi * taup * (1.0 - abs(tau) / taup) * fd;
    x(j,i) = abs( val1 * sin(val2+eps)/(val2+eps));
  end
end
```

Listing 4.2. MATLAB Program “fig4_2.m”

```
% Use this program to reproduce Fig. 4.2 of text
close all
clear all
eps = 0.000001;
taup = 2.;
taumin = -1.1 * taup;
taumax = -taumin;
x = single_pulse_ambg(taup);
taux = taumin:.05:taumax;
fdy = -5/taup:.05:5/taup;
figure(1)
mesh(taux,fdy,x);
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
zlabel ('Ambiguity function')
colormap([.5 .5 .5])
colormap (gray)
figure(2)
contour(taux,fdy,x);
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
colormap([.5 .5 .5])
colormap (gray)
grid
y = x.^2;
figure(3)
mesh(taux,fdy,y);
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
zlabel ('Ambiguity function')
colormap([.5 .5 .5])
colormap (gray)
figure(4)
contour(taux,fdy,y);
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
```

```
colormap([.5 .5 .5])
colormap (gray)
grid
```

Listing 4.3. MATLAB Program “fig4_4.m”

```
% Use this program to reproduce Fig 4.4 of text
close all
clear all
eps = 0.0001;
taup = 2.;
fd = -10./taup:.05:10./taup;
uncer = abs( sinc(taup .* fd));
ambg = uncer.^2;
plot(fd, ambg,'k')
xlabel ('Frequency - Hz')
ylabel ('Ambiguity - Volts')
grid
figure(2)
plot (fd, uncer,'k');
xlabel ('Frequency - Hz')
ylabel ('Uncertainty - Volts')
grid
```

Listing 4.4. MATLAB Function “lfm_ambg.m”

```
ffunction x = lfm_ambg(taup, b, up_down)
eps = 0.000001;
i = 0;
mu = up_down * b / 2. / taup;
delt = 2.2*taup/250;
delf = 2*b /250;
for tau = -1.1*taup:.05:1.1*taup
  i = i + 1;
  j = 0;
  for fd = -b:.05:b
    j = j + 1;
    val1 = 1. - abs(tau) / taup;
    val2 = pi * taup * (1.0 - abs(tau) / taup);
    val3 = (fd + mu * tau);
    val = val2 * val3;
    x(j,i) = abs( val1 * (sin(val+eps)/(val+eps))).^2;
  end
end
```

Listing 4.5. MATLAB Program “fig4_5.m”

```
% Use this program to reproduce Fig. 4.5 of text
close all
clear all
eps = 0.0001;
taup = 1.;
b =10.;
up_down = 1.;
x = lfm_ambg(taup, b, up_down);
taux = -1.1*taup:.05:1.1*taup;
fdy = -b:.05:b;
figure(1)
mesh(taux,fdy,x)
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
zlabel ('Ambiguity function')
figure(2)
contour(taux,fdy,x)
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
y = sqrt(x);
figure(3)
mesh(taux,fdy,y)
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
zlabel ('Uncertainty function')
figure(4)
contour(taux,fdy,y)
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
```

Listing 4.6. MATLAB Program “fig4_6.m”

```
% Use this program to reproduce Fig. 4.6 of text
close all
clear all
taup = 1;
b =20.;
up_down = 1.;
taux = -1.5*taup:.01:1.5*taup;
fd = 0.;
mu = up_down * b / 2. / taup;
ii = 0.;
```

```
for tau = -1.5*taup:.01:1.5*taup
   ii = ii + 1;
   val1 = 1. - abs(tau) / taup;
   val2 = pi * taup * (1.0 - abs(tau) / taup);
   val3 = (fd + mu * tau);
   val = val2 * val3;
   x(ii) = abs( val1 * (sin(val+eps)/(val+eps)));
end
figure(1)
plot(taux,x)
grid
xlabel ('Delay - seconds')
ylabel ('Uncertainty')
figure(2)
plot(taux,x.^2)
grid
xlabel ('Delay - seconds')
ylabel ('Ambiguity')
```

Listing 4.7. MATLAB Function "train_ambg.m"

```
function x = train_ambg (taup, n, pri)
if( taup > pri / 2.)
   'ERROR. Pulsewidth must be less than the PRI/2.'
   return
end
gap = pri - 2.*taup;
eps = 0.000001;
b = 1. / taup;
ii = 0.;
for q = -(n-1):1:n-1
   tauo = q - taup ;
   index = -1.;
   for tau1 = tauo:0.0533:tauo+gap+2.*taup
      index = index + 1;
      tau = -taup + index*.0533;
      ii = ii + 1;
      j = 0.;
      for fd = -b:.0533:b
         j = j + 1;
         if (abs(tau) <= taup)
            val1 = 1. -abs(tau) / taup;
            val2 = pi * taup * fd * (1.0 - abs(tau) / taup);
            val3 = abs(val1 * sin(val2+eps) /(val2+eps));
```

```
        val4 = abs((sin(pi*fd*(n-abs(q))*pri+eps))/(sin(pi*fd*pri+eps)));
        x(j,ii)=  val3 * val4 / n;
      else
        x(j,ii) = 0.;
      end
    end
  end
end
```

Listing 4.8. MATLAB Program “fig4_8.m”

```
% Use this program to reproduce Fig. 4.8 of text
close all
clear all
taup =0.2;
pri=1;
n=5;
x = train_ambg (taup, n, pri);
figure(1)
mesh(x)
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
zlabel ('Ambiguity function')
figure(2)
contour(x);
xlabel ('Delay - seconds')
ylabel ('Doppler - Hz')
```

Listing 4.9. MATLAB Function “barker_ambig.m”

```
function [ambig] = barker_ambig(uinput)
% Compute and plot the ambiguity function for a Barker code
%Compute the ambiguity function
% by utilizing the FFT through combining multiple range cuts
N = size(uinput,2);
tau = N;
Barker_code = uinput;
samp_num = size(Barker_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) = Barker_code(j);
end
v = u;
delay = linspace(-tau, 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 = -6/tau:freq_del:6/tau;
delay = linspace(-N,N,nfft);
figure (1)
mesh(freq,delay,ambig ./ max(max(ambig)))
colormap([.5 .5 .5])
colormap(gray)
axis tight
xlabel('frequency')
ylabel('delay')
zlabel('ambiguity function')
figure (2)
value = 10 * N ;
plot(delay,ambig(:,51)/value,'k')
xlabel('delay')
ylabel('normalized amibiguity cut for f=0')
grid
axis tight
figure (3)
contour(freq,delay,ambig ./ max(max(ambig)))
colormap([.5 .5 .5])
colormap (gray)
xlabel('frequency')
ylabel('delay')
grid on
```

Listing 4.10. MATLAB Function “prn_ambig.m”

```
function [ambig] = prn_ambig(uinput)
```

```
% Compute and plot the ambiguity function for a PRN code
% Compute the ambiguity function by utilizing the FFT
% through combining multiple range cuts

N = size(uinput,2);
tau = N;
PRN = uinput;
samp_num = size(PRN,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) = PRN(j);
end
% set-up the array v
v = u;
delay = linspace(0,5*tau,nfft);
freq_del = 8 / tau /100;
j = 0;
vfft = fft(v,nfft);
for freq = -4/tau:freq_del:4/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 = -4/tau:freq_del:4/tau;
delay = linspace(-N,N,nfft);
figure(1)
mesh(freq,delay,ambig ./ max(max(ambig)))
colormap([.5 .5 .5])
colormap(gray)
axis tight
xlabel('frequency')
ylabel('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(freq,delay,ambig ./ max(max(ambig)))
axis tight
colormap([.5 .5 .5])
colormap(gray)
xlabel('frequency')
ylabel('delay')
```

Listing 4.11. MATLAB Program “myradar_visit4.m”

```
% Use this program to reproduce Figs. 4.25 to 4.27 of the text
close all
clear all
eps = 0.0001;
taup = 20.e-6;
b =1.e6;
up_down = 1.;
i = 0;
mu = up_down * b / 2. / taup;
delt = 2.2*taup /250;
delf = 2*b /300;
for tau = -1.1*taup:delt:1.1*taup
  i = i + 1;
  j = 0;
  for fd = -b:delf:b
    j = j + 1;
    val1 = 1. - abs(tau) / taup;
    val2 = pi * taup * (1.0 - abs(tau) / taup);
    val3 = (fd + mu * tau);
    val = val2 * val3;
    x(j,i) = abs( val1 * (sin(val+eps)/(val+eps))).^2;
  end
end
taux = linspace(-1.1*taup,1.1*taup,251).*1e6;
fdy = linspace(-b,b,301) .* 1e-6;
figure(1)
mesh(taux,fdy,sqrt(x))
xlabel ('Delay - Micro-seconds')
ylabel ('Doppler - MHz')
zlabel ('Ambiguity function')
figure(2)
```

```
contour(taux,fdy,sqrt(x))
xlabel ('Delay - Micro-seconds')
ylabel ('Doppler - MHz')
grid
```

Chapter 5 *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, which can hinder the radar's normal modes of operation, particularly for multi-function and surveillance radars. Since the average transmitted power is directly linked to the receiver SNR, it is often desirable to increase the pulsewidth (i.e., increase the average transmitted power) while simultaneously maintaining adequate range resolution. This can be made possible by using pulse compression techniques. 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, we will analyze analog and digital pulse compression techniques.

Two LFM pulse compression techniques are discussed in this chapter. 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.

5.1. 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

$$N_i = 2\frac{N_0}{2}B \tag{5.1}$$

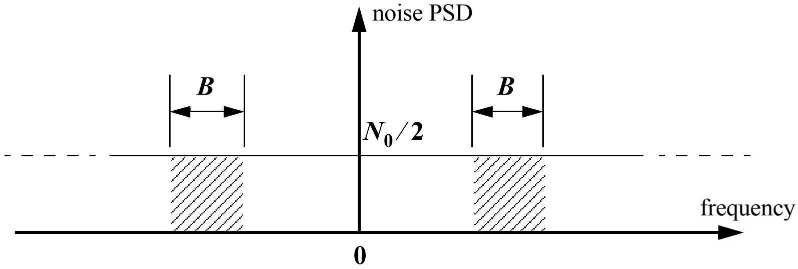


Figure 5.1. Input noise power.

where the factor of two is used to account for both negative and positive frequency bands, as illustrated in Fig. 5.1. The average input signal power over a pulse duration τ' is

$$S_i = \frac{E}{\tau'} \tag{5.2}$$

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

$$(SNR)_i = \frac{S_i}{N_i} = \frac{E}{N_0 B\tau'} \tag{5.3}$$

The output peak instantaneous SNR to the input SNR ratio is

$$\frac{SNR(t_0)}{(SNR)_i} = 2B\tau' \tag{5.4}$$

The quantity $B\tau'$ is referred to as the "time-bandwidth product" for a given waveform or its corresponding matched filter. The factor $B\tau'$ 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 $B\tau'$. Clearly, the compression gain becomes smaller than $B\tau'$ as the spectrum of the matched filter deviates from that of the input signal.

5.2. Radar Equation with Pulse Compression

The radar equation for a pulsed radar can be written as

$$SNR = \frac{P_t \tau' G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_e F L} \tag{5.5}$$

where P_t is peak power, τ' is pulsewidth, G is antenna gain, σ is target RCS, R is range, k is Boltzman's constant, T_e is effective noise temperature, 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 pulsewidth. Denote the compressed pulsewidth as τ_c. Thus, for an individual subpulse, Eq. (5.5) can be written as

$$(SNR)_{\tau_c} = \frac{P_t \tau_c G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_e F L} \tag{5.6}$$

The SNR for the uncompressed pulse is then derived from Eq. (5.6) as

$$SNR = \frac{P_t(\tau' = n\tau_c) G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_e F L} \tag{5.7}$$

where n is the number of subpulses. Equation (5.7) is denoted as the radar equation with pulse compression.

Observation of Eqs. (5.5) and (5.7) 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 pulsewidth unchanged and by increasing the bandwidth. Remember that range resolution is proportional to the inverse of the signal bandwidth,

$$\Delta R = c/2B \tag{5.8}$$

5.3. LFM Pulse Compression

Linear FM pulse compression is accomplished by adding frequency modulation to a long pulse at transmission, and by using a matched filter receiver in order to compress the received signal. As a result, the matched filter output is compressed by a factor $\xi = B\tau'$, where τ' is the pulsewidth and B is the bandwidth. Thus, by using long pulses and wideband LFM modulation large compression ratios can be achieved.

Figure 5.2 shows an ideal LFM pulse compression process. Part (a) shows the envelope for a wide 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.

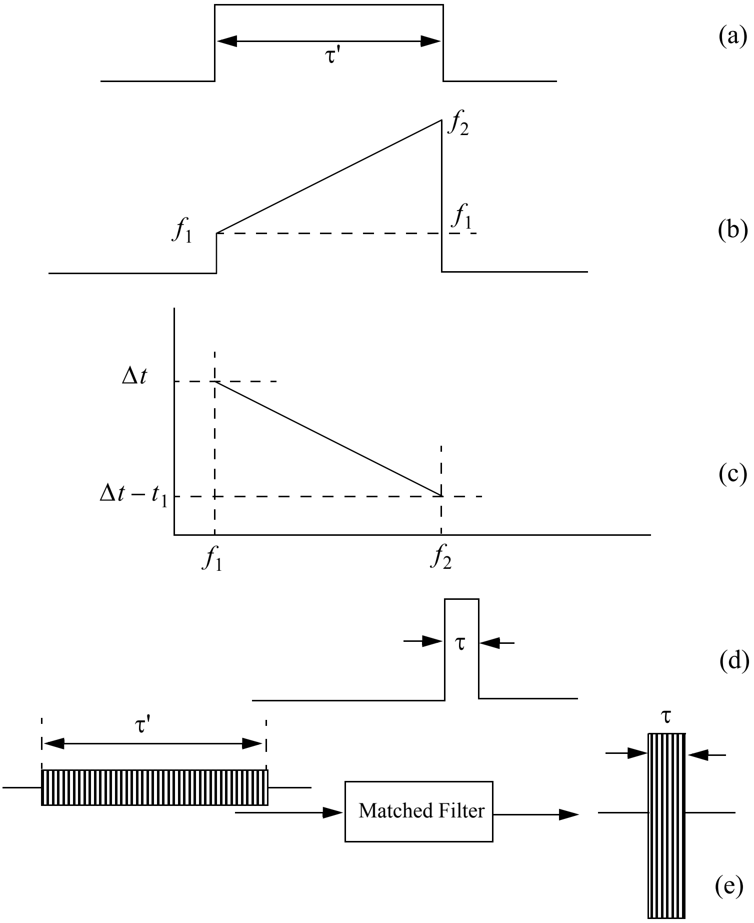


Figure 5.2 Ideal LFM pulse compression.

Fig. 5.3 illustrates the advantage of pulse compression using 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. Fig. 5.3a 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_{n1}, the compressed pulsewidth. This figure can be reproduced using MATLAB program *"fig5_3.m"* given in Listing 5.1 in Section 5.5.

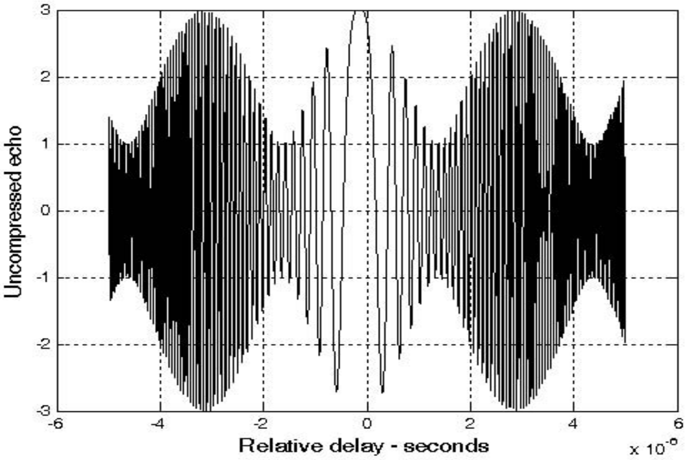


Figure 5.3a. Composite echo signal for two unresolved targets.

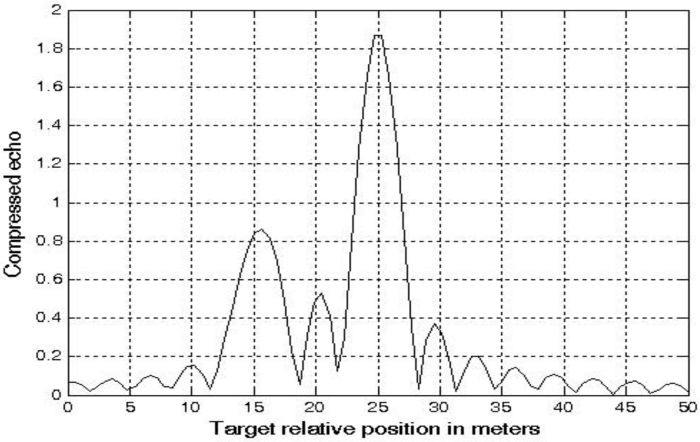


Figure 5.3b. Composite echo signal corresponding to Fig. 5.3a, after pulse compression.

5.3.1. 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. 5.4

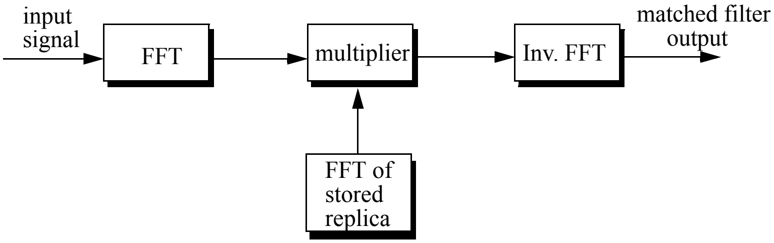


Figure 5.4. Computing the matched filter output using an FFT.

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,

$$y(t) = s(t) \bullet h(t) \tag{5.9}$$

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

$$FFT\{s(t) \bullet h(t)\} = S(f) \cdot H(f) \tag{5.10}$$

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

$$y = FFT^{-1}\{S \cdot H\} \tag{5.11}$$

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

$$R_{rec} = R_{max} - R_{min} \tag{5.12}$$

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

$$s(t) = \exp\left(j2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le \tau' \tag{5.13}$$

τ' is the pulsewidth, $\mu = B/\tau'$, 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

$$s_r(t) = a_1 \exp\left(j2\pi\left(f_0(t-\tau_1) + \frac{\mu}{2}(t-\tau_1)^2\right)\right) \tag{5.14}$$

where a_1 is proportional to target RCS, antenna gain, and range attenuation. The time delay τ_1 is given by

$$\tau_1 = 2R_1/c \tag{5.15}$$

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 low pass filtering, is then given by

$$\psi(t) = 2\pi\left(-f_0\tau_1 + \frac{\mu}{2}(t-\tau_1)^2\right) \tag{5.16}$$

and the instantaneous frequency is

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\psi(t) = \mu(t-\tau_1) = \frac{B}{\tau'}\left(t - \frac{2R_1}{c}\right) \tag{5.17}$$

The quadrature components are

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} \cos\psi(t) \\ \sin\psi(t) \end{pmatrix} \tag{5.18}$$

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

$$f_s \geq 2B \tag{5.19}$$

and the sampling interval is

$$\Delta t \leq 1/2B \tag{5.20}$$

Using Eq. (5.17) it can be shown that (the proof is left as an exercise) the frequency resolution of the FFT is

$$\Delta f = 1/\tau' \tag{5.21}$$

The minimum required number of samples is

$$N = \frac{1}{\Delta f \Delta t} = \frac{\tau'}{\Delta t} \tag{5.22}$$

Equating Eqs. (5.20) and (5.22) yields

$$N \geq 2B\tau' \tag{5.23}$$

Consequently, a total of $2B\tau'$ real samples, or $B\tau'$ complex samples, is sufficient to completely describe an LFM waveform of duration τ' and bandwidth B. For example, an LFM signal of duration $\tau' = 20\ \mu s$ 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

$$N_{FFT} = 2^m \geq N \tag{5.24}$$

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 (HRR profile). 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

$$\psi(t) = \sum_{i=1}^{I} 2\pi\left(-f_0\tau_i + \frac{\mu}{2}(t-\tau_i)^2\right) \tag{5.25}$$

The times $\{\tau_i = (2R_i/c);\ i = 1, 2, \ldots, I\}$ represent the two-way time delays, where τ_1 coincides with the start of the receive window.

MATLAB Function "matched_filter.m"

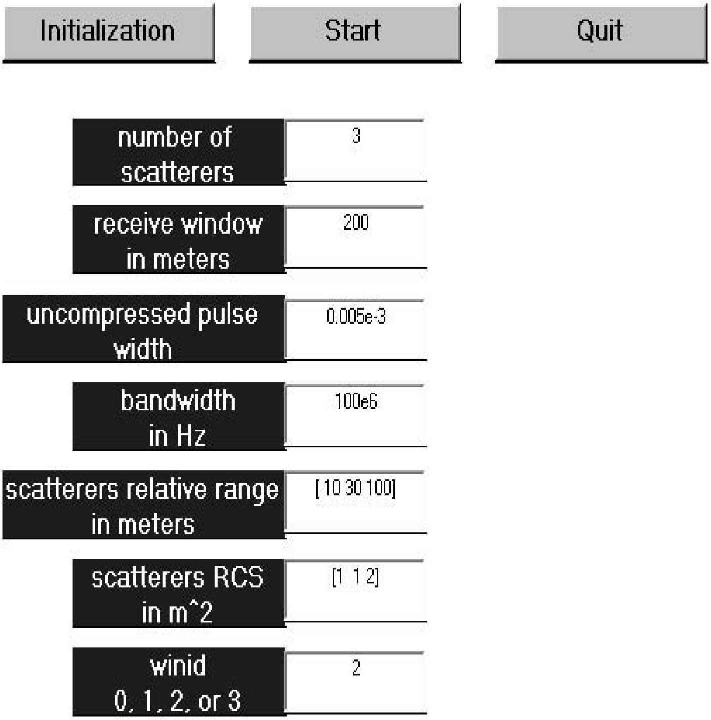
The function *"matched_filter.m"* performs fast convolution processing. It is given in Listing 5.2 in Section 5.5. The syntax is as follows:

[y] = matched_filter(nscat, taup, b, rrec, 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 pulsewidth*	*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 - sidelobes at -60dB*	*none*	*input*
y	*normalized compressed output*	*volts*	*output*

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. 5.5. 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"* which implements Eq. (5.24). It is given in Listing 5.3 in Section 5.5.



number of scatterers	3
receive window in meters	200
uncompressed pulse width	0.005e-3
bandwidth in Hz	100e6
scatterers relative range in meters	[10 30 100]
scatterers RCS in m^2	[1 1 2]
winid 0, 1, 2, or 3	2

Figure 5.5. GUI workspace associated with the function *"matched_filter_gui.m"*.

As an example, consider the case where

nscat	*3*
rrec	*200 m*
taup	*0.005 ms*
b	*100e6 Hz*
scat_range	*[10 75 120] m*
scat_rcs	$[1\ 2\ 1]m^2$
win	*2*

Note that the compressed pulsed range resolution, without using a window, is $\Delta R = 1.5m$. Figs. 5.6 shows the real part and the amplitude spectrum for the replica used in the pulse compression. Fig. 5.7 shows the uncompressed echo, while Fig. 5.8 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.

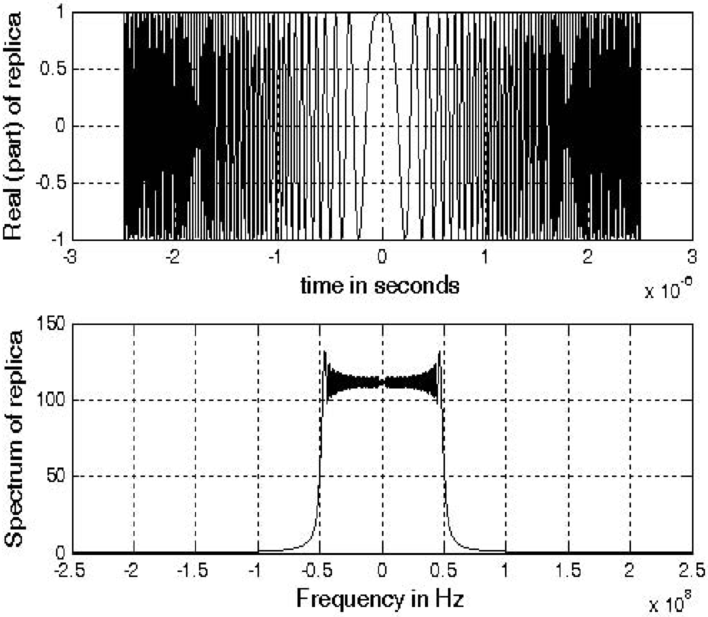


Figure 5.6. Real part and amplitude spectrum of replica.

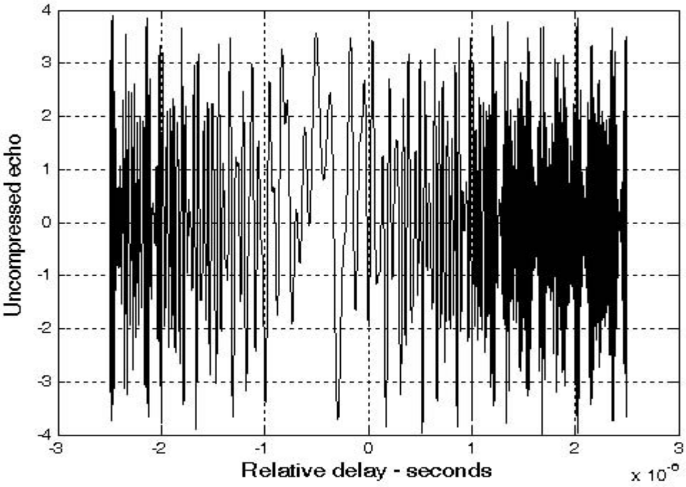


Figure 5.7. Uncompressed echo signal. Scatterers are not resolved.

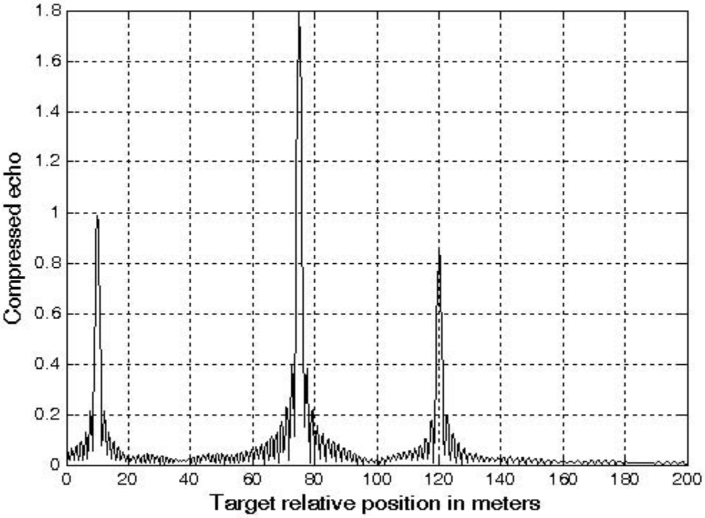


Figure 5.8. Compressed echo signal corresponding to Fig. 5.7. Scatterers are completely resolved.

Fig. 5.9 is similar to Fig. 5.8, except in this case the first and second scatterers are less than 1.5 meter apart (they are at 70 and 71 meters within the receive window).

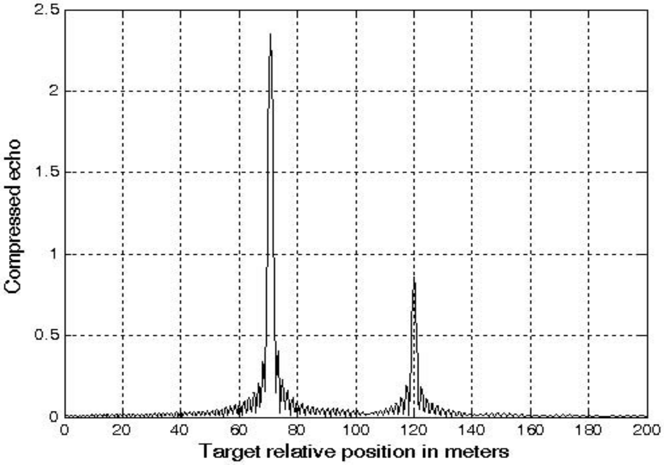


Figure 5.9. Compressed echo signal of three scatterers, two of which are not resolved.

5.3.2. 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. Fig. 5.10a 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. 5.10a. Mixing with the reference signal and performing low pass 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

$$s_1(t) = \cos\left(2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le \tau' \tag{5.26}$$

where $\mu = B/\tau'$ 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

$$s_r(t) = a\cos\left[2\pi\left(f_0(t-\Delta\tau) + \frac{\mu}{2}(t-\Delta\tau)^2\right)\right] \tag{5.27}$$

where a is proportional to target RCS, antenna gain, and range attenuation. The time delay $\Delta\tau$ is

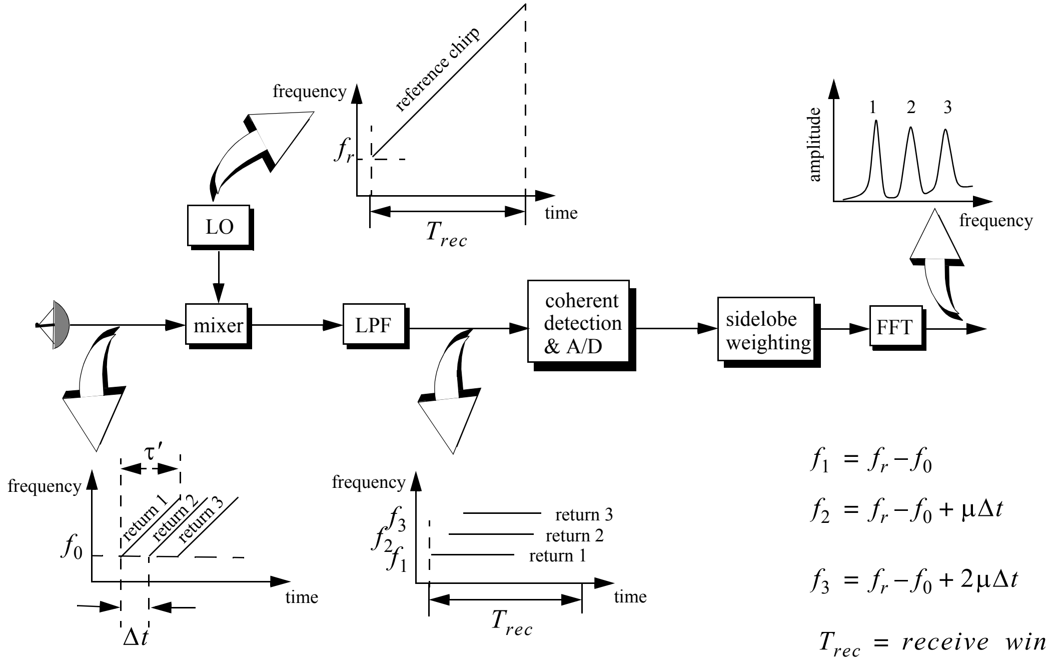
$$\Delta\tau = 2R/c \tag{5.28}$$

The reference signal is

$$s_{ref}(t) = 2\cos\left(2\pi\left(f_r t + \frac{\mu}{2}t^2\right)\right) \qquad 0 \le t \le T_{rec} \tag{5.29}$$

The receive window in seconds is

$$T_{rec} = \frac{2(R_{max} - R_{min})}{c} = \frac{2R_{rec}}{c} \tag{5.30}$$



$$f_1 = f_r - f_0$$

$$f_2 = f_r - f_0 + \mu\Delta t$$

$$f_3 = f_r - f_0 + 2\mu\Delta t$$

$$T_{rec} = receive\ \ window$$

Figure 5.10a. Stretch processing block diagram.

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

$$s_0(t) = a\cos(2\pi f_0\Delta\tau + 2\pi\mu\Delta\tau t - \pi\mu(\Delta\tau)^2) \tag{5.31}$$

Substituting Eq. (5.28) into (5.31) and collecting terms yield

$$s_0(t) = a \cos\left[\left(\frac{4\pi BR}{c\tau'}\right)t + \frac{2R}{c}\left(2\pi f_0 - \frac{2\pi BR}{c\tau'}\right)\right] \tag{5.32}$$

and since $\tau' » 2R/c$, Eq. (5.32) is approximated by

$$s_0(t) \approx a \cos\left[\left(\frac{4\pi BR}{c\tau'}\right)t + \frac{4\pi R}{c}f_0\right] \tag{5.33}$$

The instantaneous frequency is

$$f_{inst} = \frac{1}{2\pi}\frac{d}{dt}\left(\frac{4\pi BR}{c\tau'}t + \frac{4\pi R}{c}f_0\right) = \frac{2BR}{c\tau'} \tag{5.34}$$

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

$$R_1 = f_1 c\tau'/2B \tag{5.35}$$

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

$$s_r(t) = \sum_{i=1}^{I} a_i(t)\cos\left[2\pi\left(f_0(t-\tau_i) + \frac{\mu}{2}(t-\tau_i)^2\right)\right] \tag{5.36}$$

where $\{a_i(t);\ i = 1, 2, \ldots, I\}$ are proportional to the targets' cross sections, antenna gain, and range. The times $\{\tau_i = (2R_i/c);\ i = 1, 2, \ldots, I\}$ represent the two-way time delays, where τ_1 coincides with the start of the receive window. Using Eq. (5.32) the overall signal at the output of the LPF can then be described by

$$s_o(t) = \sum_{i=1}^{I} a_i\cos\left[\left(\frac{4\pi BR_i}{c\tau'}\right)t + \frac{2R_i}{c}\left(2\pi f_0 - \frac{2\pi BR_i}{c\tau'}\right)\right] \tag{5.37}$$

And 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 pulsewidth is τ' 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. (5.8). 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 range R_1 and R_2. The minimum frequency separation, Δf, between those scatterers so that they are resolved can be computed from Eq. (5.34). More precisely,

$$\Delta f = f_2 - f_1 = \frac{2B}{c\tau'}(R_2 - R_1) = \frac{2B}{c\tau'}\Delta R \tag{5.38}$$

Substituting Eq. (5.8) into Eq. (5.38) yields

$$\Delta f = \frac{2B}{c\tau'}\ \frac{c}{2B} = \frac{1}{\tau'} \tag{5.39}$$

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

$$\frac{N\Delta f}{2} > \frac{2B(R_{max} - R_{min})}{c\tau'} = \frac{2BR_{rec}}{c\tau'} \tag{5.40}$$

Using Eqs. (5.30) and (5.39) into Eq. (5.40) and collecting terms yield

$$N > 2BT_{rec} \tag{5.41}$$

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

$$N_{FFT} \geq N = 2^m \tag{5.42}$$

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

$$\Delta f = \frac{1}{T_s N_{FFT}} \Rightarrow T_s = \frac{1}{\Delta f N_{FFT}} \tag{5.43}$$

MATLAB Function "stretch.m"

The function *"stretch.m"* presents a digital implementation of stretch processing. It is given in Listing 5.4 in Section 5.5. The syntax is as follows:

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

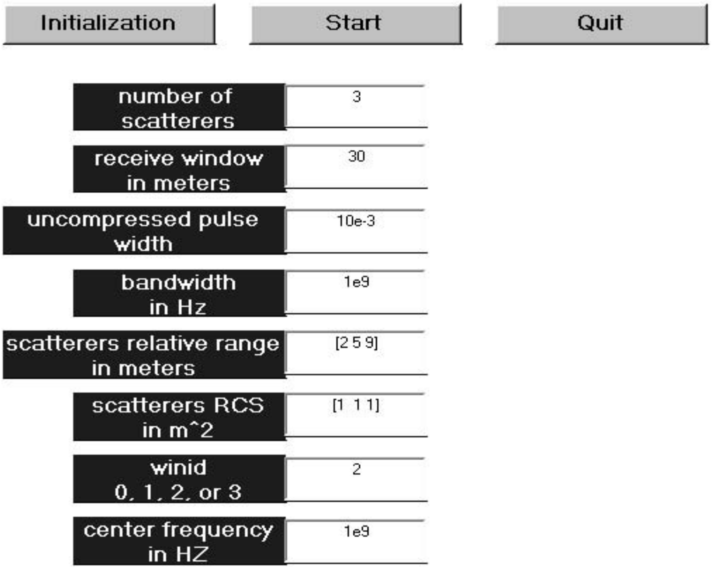
where

Symbol	**Description**	**Units**	**Status**
nscat	*number of point scatterers within the received window*	*none*	*input*
taup	*uncompressed pulsewidth*	*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 - sidelobes 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. 5.10b. The outputs of this function are the complex array y and plots of the uncompressed and compressed echo signal versus time. As an example, consider the case where

nscat	*3*
taup	*10 ms*
f0	*5.6 GHz*
b	*1 GHz*
rrec	*30 m*
scat_range	*[2 5 10] m*
scat_rcs	*[1, 1, 2]* m^2
win	*2 (Kaiser)*

Note that the compressed pulse range resolution, without using a window, is $\Delta R = 0.15m$. Figs. 5.11 and 5.12, respectively, show the uncompressed and compressed echo signals corresponding to this example. Fig. 5.13 is similar to Figs. 5.11 and 5.12 except in this case two of the scatterers are less than 15 cm apart (i.e., unresolved targets at $R_{relative} = [3, 3.1]m$).



number of scatterers: 3

receive window in meters: 30

uncompressed pulse width: 10e-3

bandwidth in Hz: 1e9

scatterers relative range in meters: [2 5 9]

scatterers RCS in m^2: [1 1 1]

winid 0, 1, 2, or 3: 2

center frequency in HZ: 1e9

Figure 5.10b. GUI workspace associated with the function *"stretch_gui.m"*.

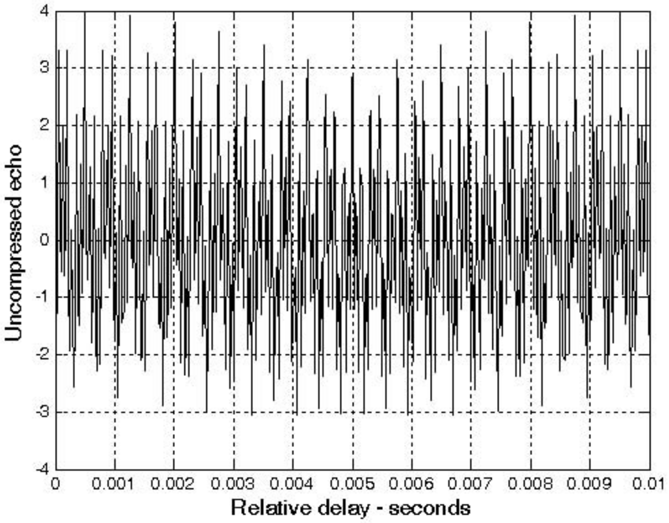


Figure 5.11. Uncompressed echo signal. Three targets are unresolved.

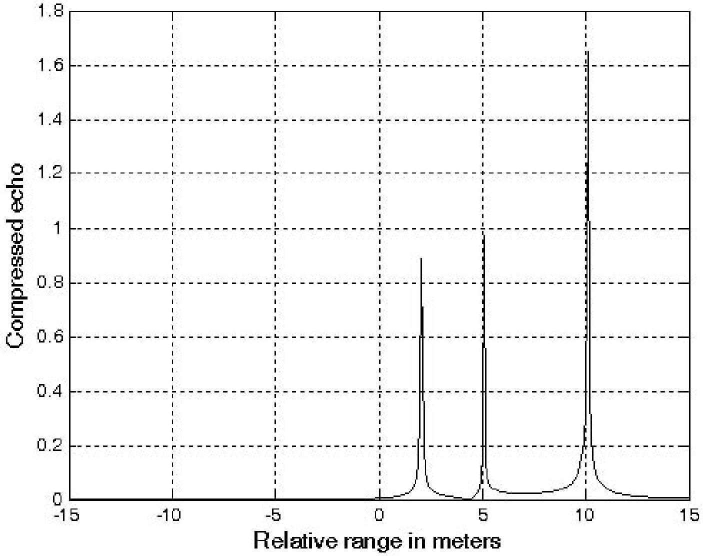


Figure 5.12. Compressed echo signal. Three targets are resolved.

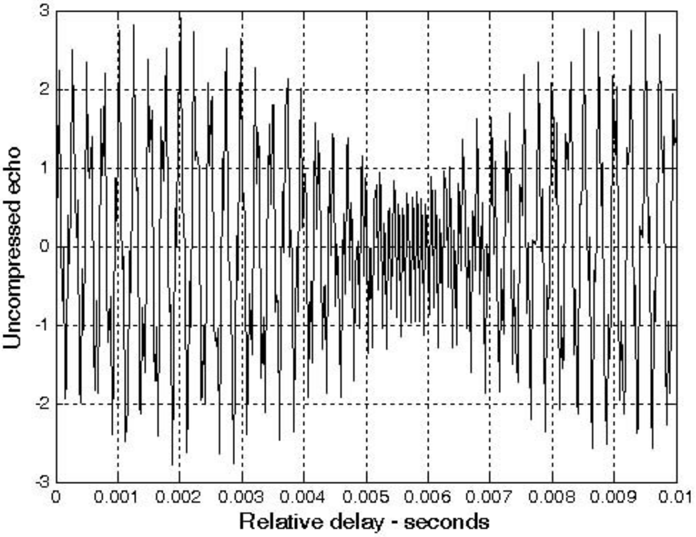


Figure 5.13a. Uncompressed echo signal. Three targets.

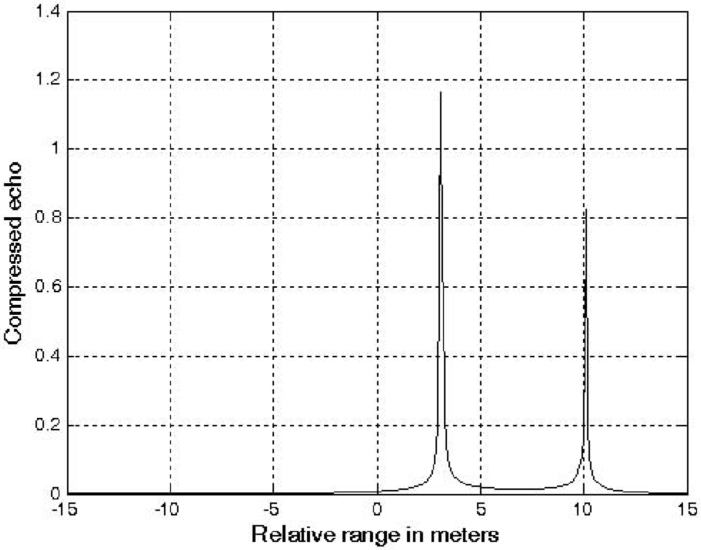


Figure 5.13b. Compressed echo signal. Three targets, two are not resolved.

5.3.3. Distortion Due to Target Velocity

Up to this point, we have analyzed pulse compression with no regard to target velocity. In fact, all analyses provided assumed stationary targets. Uncompensated target radial velocity, or equivalently Doppler shift, degrades the quality of the HRR profile generated by pulse compression. In Chapter 3, the effects of radial velocity on SFW were analyzed. Similar distortion in the HRR profile is also present with LFM waveforms when target radial velocity is not compensated for.

The two effects of target radial velocity (Doppler frequency) on the radar received pulse were developed in Chapter 1. When the target radial velocity is not zero, the received pulsewidth is expanded (or compressed) by the time dilation factor. Additionally, the received pulse center frequency is shifted by the amount of Doppler frequency. When these effects are not compensated for, the pulse compression processor output is distorted. This is illustrated in Fig. 5.14. Fig. 5.14a shows a typical output of the pulse compression processor with no distortion. Alternatively, Figs. 5.14b, 5.14c, and 5.14d show the output of the pulse compression processor when 5% shift of the chirp center frequency, 10% time dilation, and both are present.

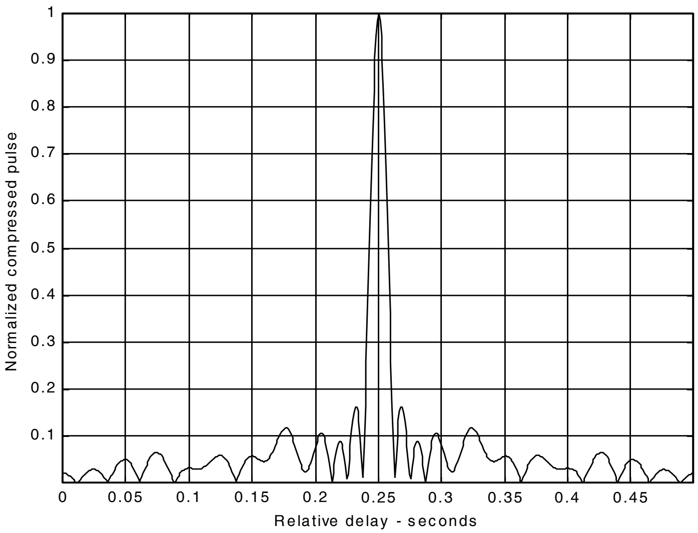


Figure 5.14a. Compressed pulse output of a pulse compression processor. No distortion is present. This figure can be reproduced using MATLAB program *"fig5_14"* given in Listing 5.5 in Section 5.5.

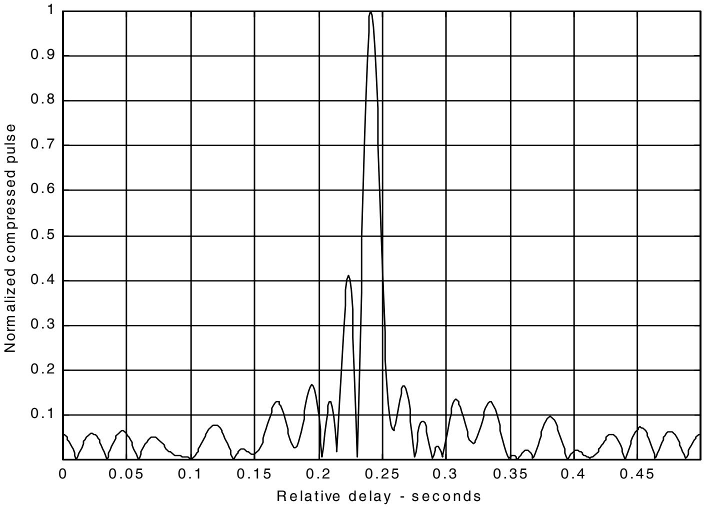


Figure 5.14b. Mismatched compressed pulse; 5% Doppler shift.

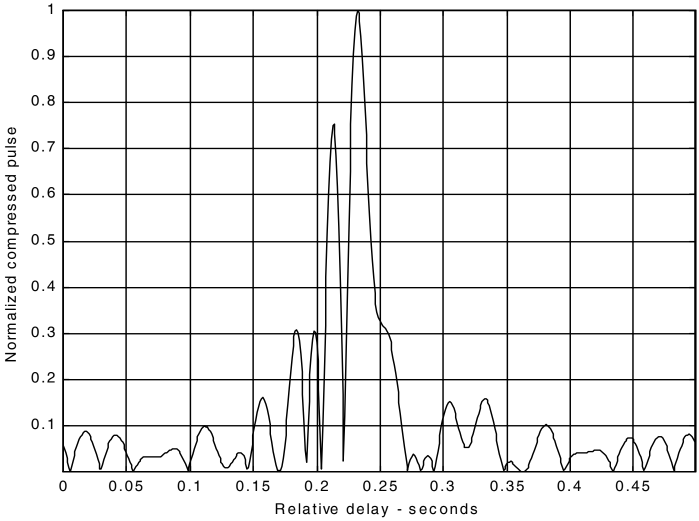


Figure 5.14c. Mismatched compressed pulse; 10% time dilation.

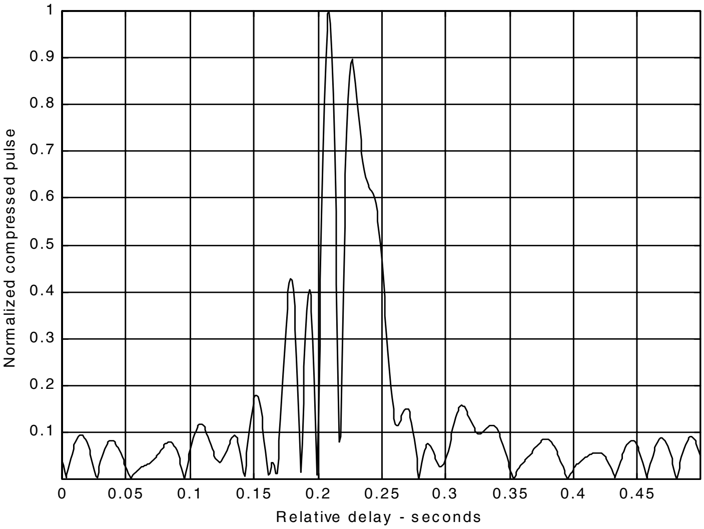


Figure 5.14d. Mismatched compressed pulse; 10% time dilation and 5% Doppler shift.

Correction for the distortion caused by the target radial velocity can be overcome by using the following approach. Over a period of a few pulses, the radar data processor estimates the radial velocity of the target under track. Then, the chirp slope and pulsewidth of the next transmitted pulse are changed to account for the estimated Doppler frequency and time dilation.

5.4. "MyRadar" Design Case Study - Visit 5

5.4.1. Problem Statement

Assume that the threat may consist of multiple aircraft and missiles. Show how the matched filter receiver can resolve multiple targets with a minimum range separation of 50 meters. Also verify that the waveforms selected in Chapter 3 are adequate to maintain proper detection and tracking (i.e., provide sufficient SNR).

5.4.2. A Design

It was determined in Chapter 3 that the pulsed compressed range resolutions during search and track are respectively given by

$$\Delta R_{search} = 30m ; \ B_{search} = 5MHz \tag{5.44}$$

$$\Delta R_{track} = 7.5m ; \ B_{track} = 20MHz \tag{5.45}$$

It was also determined that a single search waveform and 4 track waveforms would be used.

Assume that track is initiated once detection is declared. Aircraft target type are detected at $R^a_{max} = 90Km$ while the missile is detected at $R^m_{max} = 55Km$. It was shown in Section 2.10.2.2 that the minimum SNR at these ranges for both target types is $SNR \geq 4dB$ when 4-pulse non-coherent integration is utilized along with cumulative detection. It was also determined that a single pulse option was not desirable since it required prohibitive values for the peak power. At this point one should however take advantage of the increased SNR due to pulse compression. From Chapter 3, the pulse compression gain, for the selected waveforms, is equal to 100 (10 dB). One should investigate this SNR enhancement in the context of eliminating the need for pulse integration.

The pulsed compressed SNR can be computed using Eq. (5.7), which is repeated here as Eq. (5.46)

$$SNR = \frac{P_t \tau' G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_e F L} \tag{5.46}$$

where $G = 34.5dB$, $\lambda = 0.1m$, $T_e = 290Kelvin$, $F = 6dB$, $L = 8dB$, $\sigma_m = 0.5m^2$, $\sigma_a = 4m^2$, and $P_t = 20KW$ (from Chapter 3). The search pulsewidth is $\tau' = 20\mu s$ and the track waveforms are $12.5\mu s \le \tau'_i \le 20\mu s$. First consider the missile case. The single pulse SNR at the maximum detection range $R^m_{max} = 55Km$ is given by

$$SNR_m = \frac{20 \times 10^3 \times 20 \times 10^{-6} \times (10^{3.45})^2 \times (0.1)^2 \times 0.5}{(4\pi)^3 \times (55 \times 10^3)^4 \times 1.38 \times 10^{-23} \times 290 \times 10^{0.8} \times 10^{0.6}} = \tag{5.47}$$

$$8.7028 \Rightarrow SNR_m = 9.39dB$$

Alternatively, the single pulse SNR, with pulse compression, for the aircraft is

$$SNR_a = \frac{20 \times 10^3 \times 20 \times 10^{-6} \times (10^{3.45})^2 \times (0.1)^2 \times 4}{(4\pi)^3 \times (90 \times 10^3)^4 \times 1.38 \times 10^{-23} \times 290 \times 10^{0.8} \times 10^{0.6}} = \tag{5.48}$$

$$9.7104 \Rightarrow SNR_m = 9.87dB$$

Using these calculated SNR values into the MATLAB program *“myradar_visit2_2.m”* (see Chapter 2) yields

$$\begin{aligned} P_{DC_{Aircarft}} &= 0.999 \\ P_{DC_{Missile}} &= 0.9984 \end{aligned} \tag{5.49}$$

which clearly satisfies the design requirement of $P_D \ge 0.995$.

Next, consider the matched filter and its replicas and pulsed compressed outputs (due to different waveforms). For this purpose use the program *“matched_filter_gui.m”*. Assume a receive window of 200 meters during search and 50 meters during track.

Fig. 5.15 shows the replica and the associated uncompressed and compressed signals. The targets consist of two aircraft separated by 50 meters. Fig. 5.16 is similar to Fig. 5.15, except it is for track waveform number 4 and the target separation is 20 m.

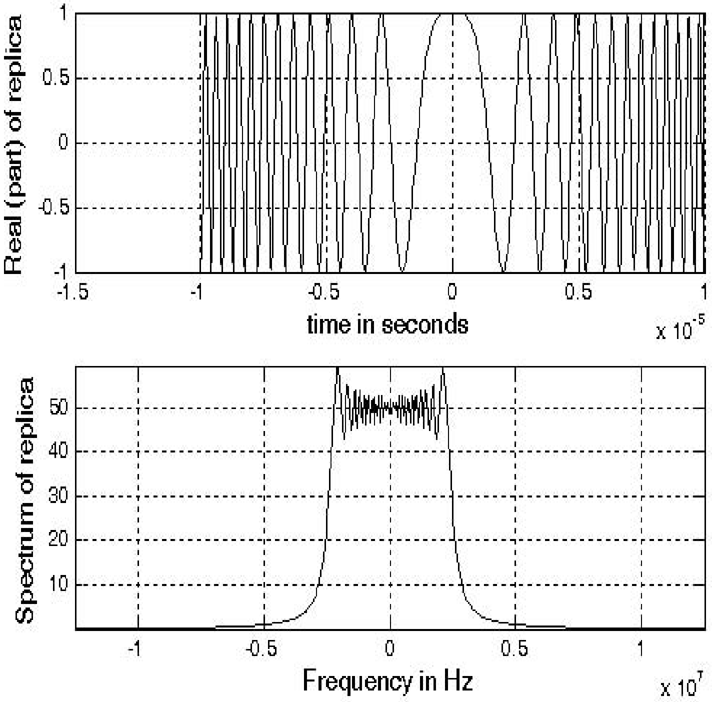


Figure 5.15a. Replica associated with search waveform.

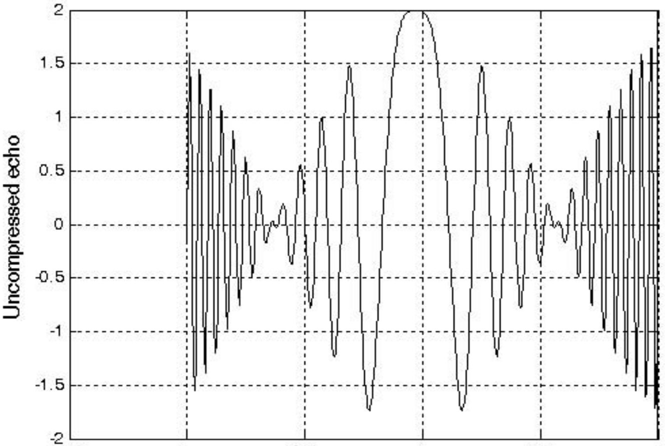


Figure 5.15b. Uncompressed signal of two aircraft separated by 50 m.

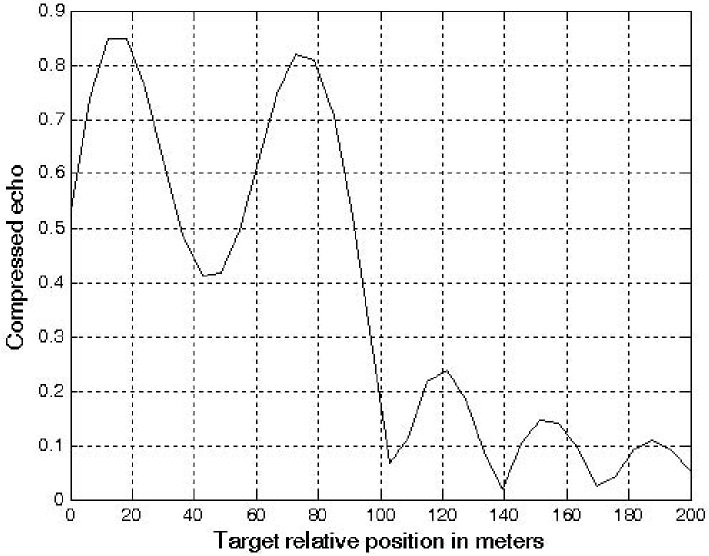


Figure 5.15c. Compressed signal corresponding to Fig. 5.15b. No window.

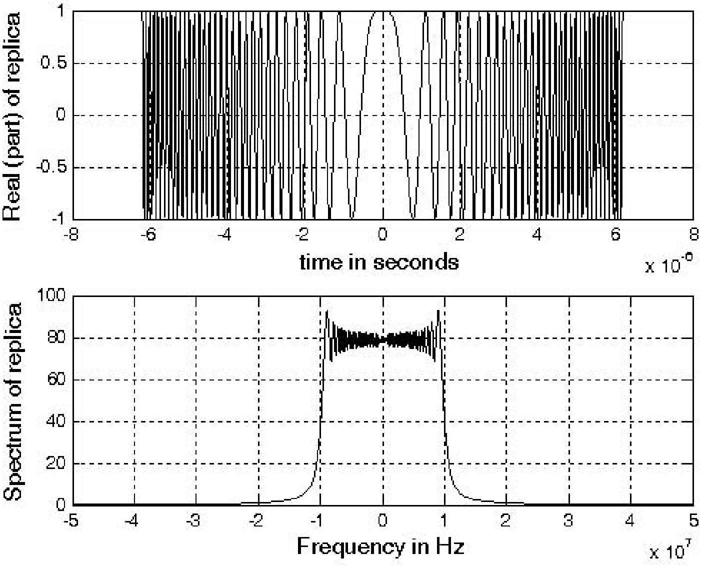


Figure 5.16a. Replica associated with track waveform number 4.

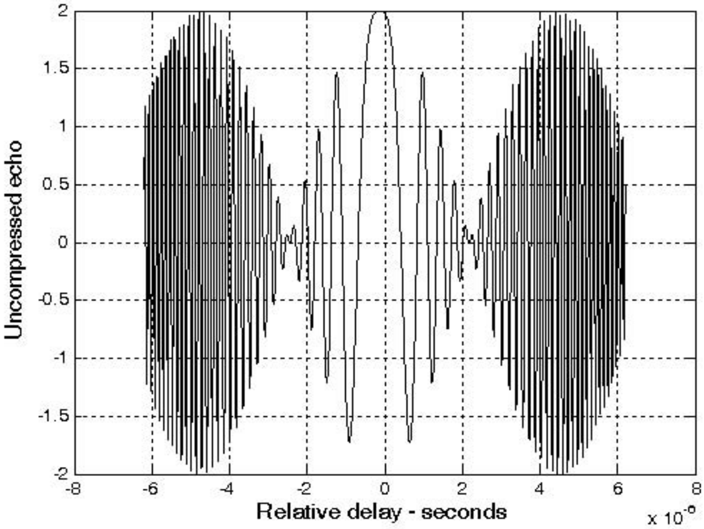


Figure 5.16b. Uncompressed signal of two aircraft separated by 20 m.

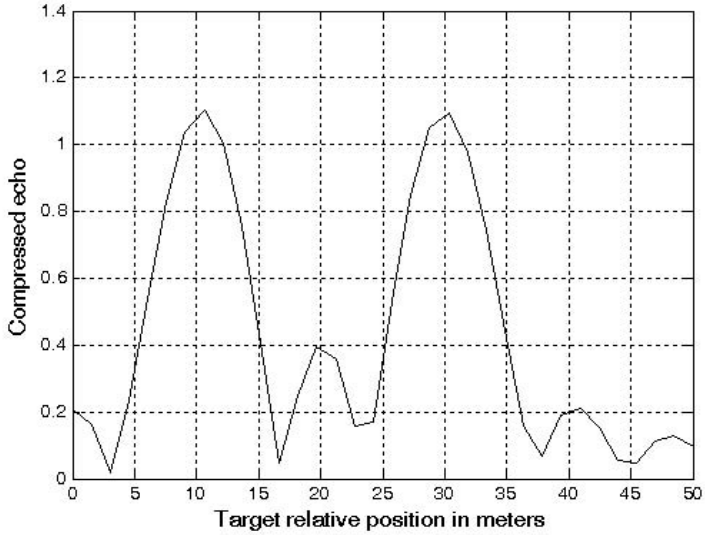


Figure 5.16b. Compressed signal of two aircraft separated by 20 m.

5.5. MATLAB Program and Function Listings

This section presents listings for all MATLAB programs/functions used in this chapter. The user is strongly advised to rerun the MATLAB programs in order to enhance his understanding of this chapter's material.

Listing 5.1. MATLAB Program "fig5_3.m"

```
% use this program to reproduce Fig. 5.3 of text
clear all
close all
nscat = 2; %two point scatterers
taup = 10e-6; % 10 microsecond uncompressed pulse
b = 50.0e6; % 50 MHz bandwidth
rrec = 50 ; % 50 meter processing window
scat_range = [15 25] ; % scatterers are 15 and 25 meters into window
scat_rcs = [1 2]; % RCS 1 m^2 and 2m^2
winid = 0; %no window used
[y] = matched_filter(nscat,taup,b,rrec,scat_range,scat_rcs,winid);
```

Listing 5.2. MATLAB Function "matched_filter.m"

```
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.;
  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
% check to ensure that scatterers are within receive 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)
plot(t,real(y),'k')
xlabel ('Relative delay - 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
```

Listing 5.3. MATLAB Function “power_integer_2.m”

```
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
```

Listing 5.4. MATLAB Function “stretch.m”

```
function [y] = stretch(nscat,taup,f0,b,rrec,scat_range,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.;
  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 - seconds')
ylabel ('Uncompressed echo')
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
```

Listing 5.5. MATLAB Program “fig5_14.m”

```
% use this program to reproduce Fig. 5.14 of text
clear all
eps = 1.5e-5;
t = 0:0.001:.5;
y = chirp(t,0,.25,20);
figure(1)
plot(t,y);
yfft = fft(y,512) ;
ycomp = fftshift(abs(ifft(yfft .* conj(yfft))));
maxval = max (ycomp);
ycomp = eps + ycomp ./ maxval;
figure(1)
del = .5 /512.;
tt = 0:del:.5-eps;
plot (tt,ycomp,'k')
axis tight
xlabel ('Relative delay - seconds');
ylabel('Normalized compressed pulse')
grid
y1 = chirp (t,0,.25,21); % change center frequency
y1fft = fft(y1,512);
y1comp = fftshift(abs(ifft(y1fft .* conj(yfft))));
maxval = max (y1comp);
y1comp = eps + y1comp ./ maxval;
figure(2)
plot (tt,y1comp,'k')
axis tight
xlabel ('Relative delay - seconds');
ylabel('Normalized compressed pulse')
grid
t = 0:0.001:.45; % change pulsewidth
y2 = chirp (t,0,.225,20);
y2fft = fft(y2,512);
y2comp = fftshift(abs(ifft(y2fft .* conj(yfft))));
maxval = max (y2comp);
y2comp = eps + y2comp ./ maxval;
figure(3)
plot (tt,y2comp,'k')
axis tight
xlabel ('Relative delay - seconds');
ylabel('Normalized compressed pulse')
grid
```

Chapter 6 *Surface and Volume Clutter*

6.1. Clutter Definition

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 main lobe 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.

White noise normally introduces the same amount of noise power across all radar range bins, while clutter power may vary within a single range bin. 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 (via clutter map). 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

$$\sigma_c = \sigma^0 A_c \tag{6.1}$$

where $\sigma^0(m^2/m^2)$ is the clutter scattering coefficient, a dimensionless quantity that is often expressed in dB. Some radar engineers express σ^0 in terms of squared centimeters per squared meter. In these cases, σ^0 is $40dB$ higher than normal.

6.2. 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. 6.1.

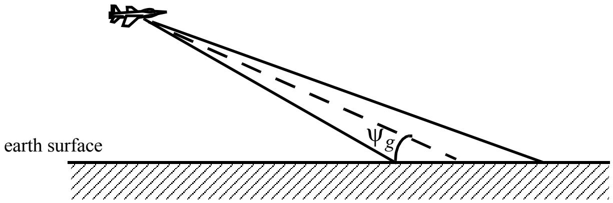


Figure 6.1. Definition of grazing angle.

Three factors affect the amount of clutter in the radar beam. They are the grazing angle, surface roughness, and the radar wavelength. Typically, the clutter scattering coefficient σ^0 is larger for smaller wavelengths. Fig. 6.2 shows a sketch describing the dependency of σ^0 on the grazing angle. Three regions are identified; they are the low grazing angle region, flat or plateau region, and the high grazing angle region.

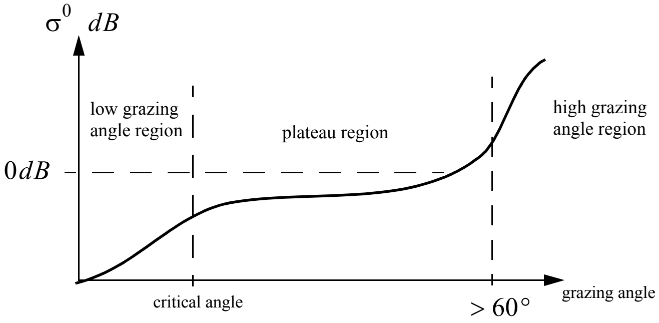


Figure 6.2. Clutter regions.

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

$$\frac{4\pi h_{rms}}{\lambda} \sin\psi_g < \frac{\pi}{2} \tag{6.2}$$

Consider a wave incident on a rough surface, as shown in Fig. 6.3. 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 $\Delta\psi$:

$$\Delta\psi = \frac{2\pi}{\lambda}\ 2h_{rms}\sin\psi_g \tag{6.3}$$

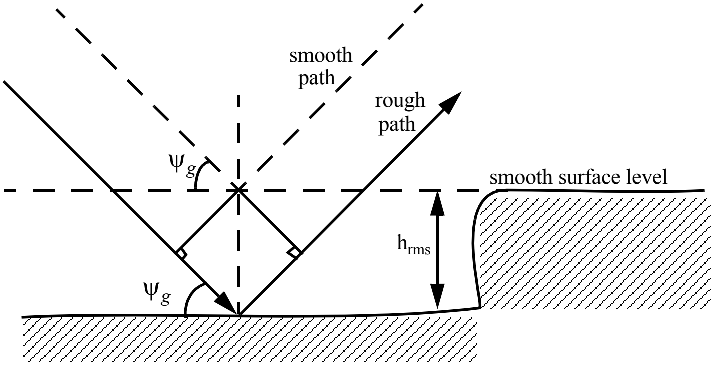


Figure 6.3. Rough surface definition.

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

$$\frac{4\pi h_{rms}}{\lambda}\ \sin\psi_{gc} = \pi \tag{6.4}$$

or equivalently,

$$\psi_{gc} = \operatorname{asin}\frac{\lambda}{4h_{rms}} \tag{6.5}$$

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

$$h_{rms} \approx 0.025 + 0.046 \ S_{state}^{1.72} \tag{6.6}$$

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, where in this case the wave height is approximately between $0.9144 \ to \ 1.2192 \ m$, 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 (non-coherent 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, opposite of the low grazing angle region.

6.2.1. Radar Equation for Area Clutter - Airborne Radar

Consider an airborne radar in the look-down mode shown in Fig. 6.4. 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. 6.5. The footprint is divided into many ground range bins each of size $(c\tau/2)\sec\psi_g$, where τ is the pulsewidth.

From Fig. 6.5, the clutter area A_c is

$$A_c \approx R\theta_{3dB}\frac{c\tau}{2}\sec\psi_g \tag{6.7}$$

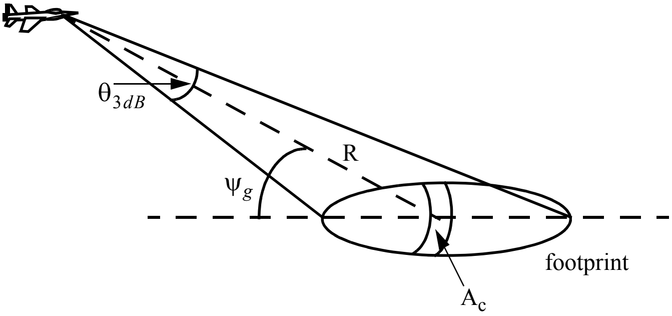


Figure 6.4. Airborne radar in the look-down mode.

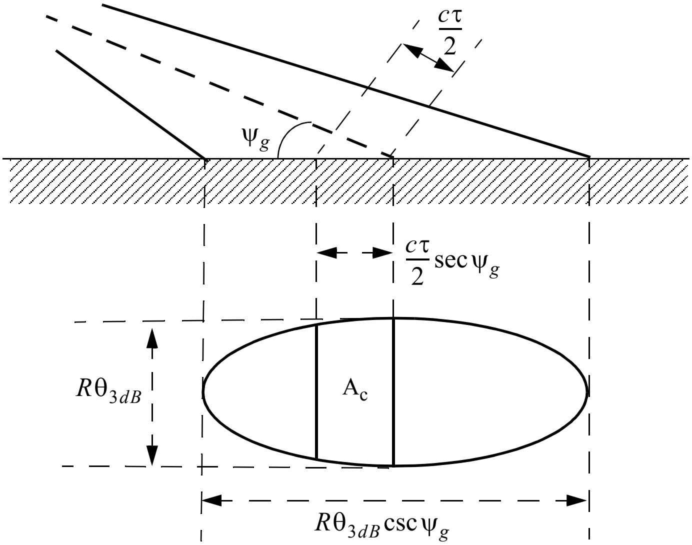


Figure 6.5. Footprint definition.

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

$$S_t = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4} \tag{6.8}$$

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

$$S_C = \frac{P_t G^2 \lambda^2 \sigma_c}{(4\pi)^3 R^4} \tag{6.9}$$

where the subscript C is used for area clutter. Substituting Eq. (6.1) for σ_c into Eq. (6.9), we can then obtain the SCR for area clutter by dividing Eq. (6.8) by Eq. (6.9). More precisely,

$$(SCR)_C = \frac{2\sigma_t \cos\psi_g}{\sigma^0 \theta_{3dB} R c\tau} \tag{6.10}$$

Example:

Consider an airborne radar shown in Fig. 6.4. Let the antenna 3dB beamwidth be $\theta_{3dB} = 0.02rad$*, the pulsewidth* $\tau = 2\mu s$*, range* $R = 20Km$*, and* *grazing angle* $\psi_g = 20°$. *The target RCS is* $\sigma_t = 1m^2$. *Assume that the clutter reflection coefficient is* $\sigma^0 = 0.0136$. *Compute the SCR.*

Solution:

The SCR is given by Eq. (6.10) 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.*

6.2.2. Radar Equation for Area Clutter - Ground Based Radar

Again the received power from clutter is also calculated using Eq. (6.9). However, in this case the clutter RCS σ_c is computed differently. It is

$$\sigma_c = \sigma_{MBc} + \sigma_{SLc} \tag{6.11}$$

where σ_{MBc} is the main beam clutter RCS and σ_{SLc} is the sidelobe clutter RCS, as illustrated in Fig. 6.6.

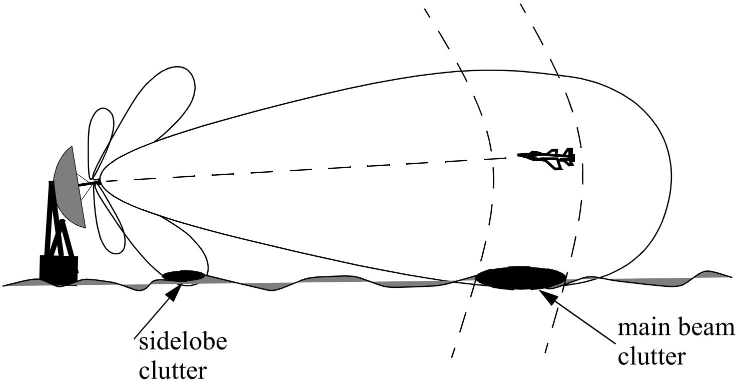


Figure 6.6. Geometry for ground based radar clutter

In order to calculate the total clutter RCS given in Eq. (6.11), 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. 6.7. 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}.

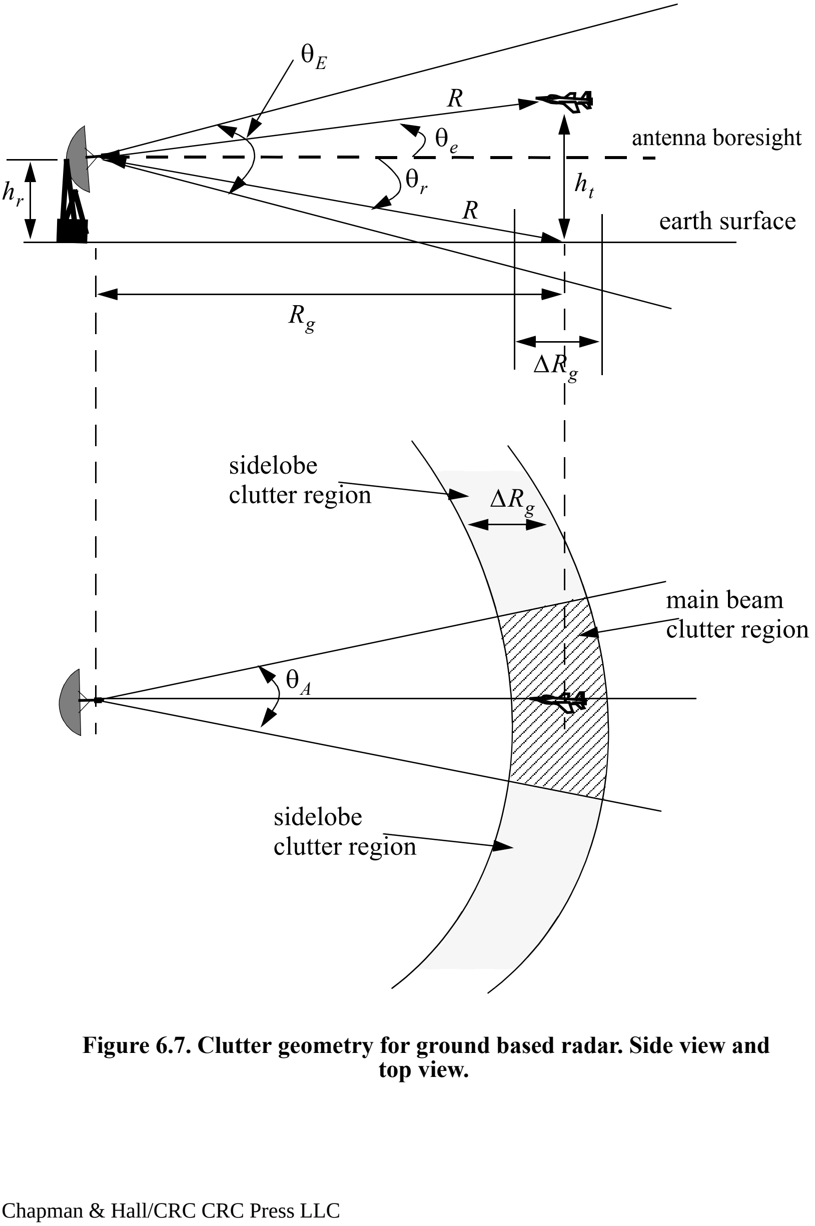


Figure 6.7. Clutter geometry for ground based radar. Side view and top view.

From Fig. 6.7 the following relations can be derived

$$\theta_r = \operatorname{asin}(h_r/R) \tag{6.12}$$

$$\theta_e = \operatorname{asin}((h_t - h_r)/R) \tag{6.13}$$

$$\Delta R_g = \Delta R \cos\theta_r \tag{6.14}$$

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

$$R_g = R\cos\theta_r \tag{6.15}$$

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

$$A_{MBc} = \Delta R_g \; R_g \; \theta_A \tag{6.16}$$

$$A_{SLc} = \Delta R_g \; \pi R_g \tag{6.17}$$

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

$$G(\theta) = \exp\left(-\frac{2.776\theta^2}{\theta_E^2}\right) \Rightarrow Gaussian \tag{6.18}$$

$$G(\theta) = \begin{cases} \left(\dfrac{\sin\left(2.78\dfrac{\theta}{\theta_E}\right)}{\left(2.78\dfrac{\theta}{\theta_E}\right)}\right)^2 & ;|\theta| \le \dfrac{\pi\theta_E}{2.78} \\ 0 & ;elsewhere \end{cases} \Rightarrow \left(\frac{\sin(x)}{x}\right)^2 \tag{6.19}$$

Then the main beam clutter RCS is

$$\sigma_{MBc} = \sigma^0 A_{MBc} G^2(\theta_e + \theta_r) = \sigma^0 \Delta R_g \; R_g \; \theta_A G^2(\theta_e + \theta_r) \tag{6.20}$$

and the sidelobe clutter RCS is

$$\sigma_{SLc} = \sigma^0 A_{SLc}(SL_{rms})^2 = \sigma^0 \Delta R_g \; \pi R_g (SL_{rms})^2 \tag{6.21}$$

where the quantity SL_{rms} is the root-mean-square (rms) for the antenna sidelobe level.

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

$$\sigma_c(R) = \frac{\sigma_{MBc} + \sigma_{SLc}}{(1 + (R/R_h)^4)} \tag{6.22}$$

where R_h is the radar range to the horizon calculated as

$$R_h = \sqrt{\frac{8h_r r_e}{3}} \tag{6.23}$$

where r_e is the Earth's radius equal to $6371Km$. The denominator in Eq. (6.22) 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

$$SNR = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4 k T_o B F L} \tag{6.24}$$

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 Boltzman'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 (CNR) at the radar is

$$CNR = \frac{P_t G^2 \lambda^2 \sigma_c}{(4\pi)^3 R^4 k T_o B F L} \tag{6.25}$$

where the σ_c is calculated using Eq. (6.21).

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

$$SIR = \frac{1}{\frac{1}{SNR} + \frac{1}{SCR}} \tag{6.26}$$

Note that the SCR is computed by dividing Eq.(6.24) by Eq. (6.25).

MATLAB Function "clutter_rcs.m"

The function *"clutter_rcs.m"* implements Eq. (6.22); it is given in Listing 6.1 in Section 6.6. It also generates plots of the clutter RCS and the CNR versus the radar slant range. Its outputs include the clutter RCS in dBsm and the CNR in dB. The syntax is as follows:

[sigmaC,CNR] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, ht, pt, f0, b, t0, f, l, 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*
pt	*radar peak power*	*KW*	*input*
f0	*radar operating frequency*	*Hz*	*input*
b	*bandwidth*	*Hz*	*input*
t0	*effective noise temperature*	*Kelvins*	*input*
f	*noise figure*	*dB*	*input*
l	*radar losses*	*dB*	*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*
CNR	*clutter to noise ratio; 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 and CNR versus range. Figure 6.8 shows typical plots produced by this GUI using the antenna pattern defined in Eq. (6.18). Figure 6.9 is similar to Fig. 6.8 except in this case Eq. (6.19) is used for the antenna pattern. Note that the dip in the clutter RCS (at very close range) occurs at the grazing angle corresponding to the null between the main beam and the first sidelobe. Fig. 6.9c shows the GUI workspace associated with this function.

In order to reproduce those two figures use the following MATLAB calls:

[sigmaC,CNR] = clutter_rcs(-20, 2, 1, -20, linspace(2,50,100), 3, 100, 75, 5.6e9, 1e6, 290, 6, 10, 1) **(6.27)**

[sigmaC,CNR] = clutter_rcs(-20, 2, 1, -25, linspace(2,50,100), 3, 100, 100, 5.6e9, 1e6, 290, 6, 10, 2) **(6.28)**

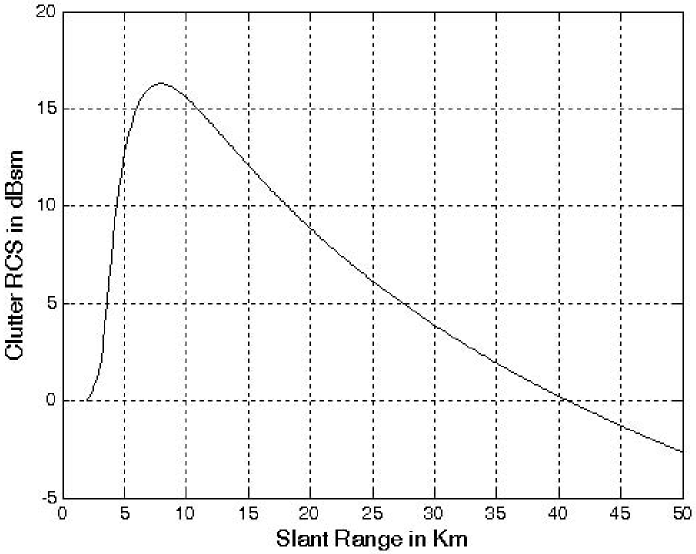


Figure 6.8a. Clutter RCS versus range using the function call in Eq. (6.27).

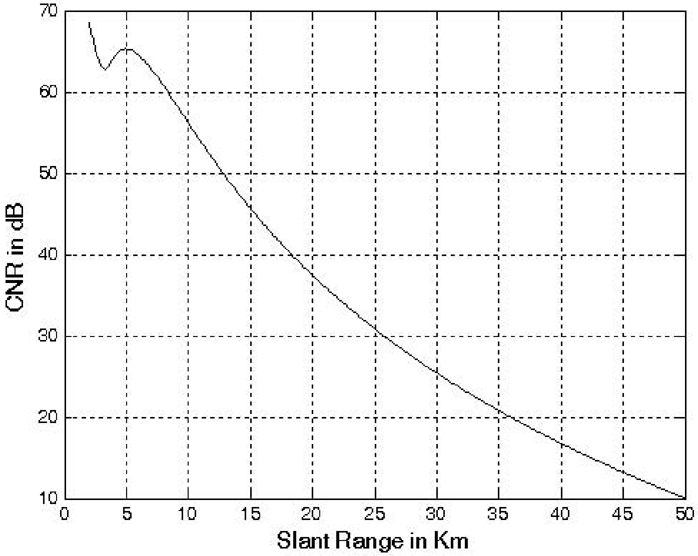


Figure 6.8b. CNR versus range using the function call in Eq. (6.27).

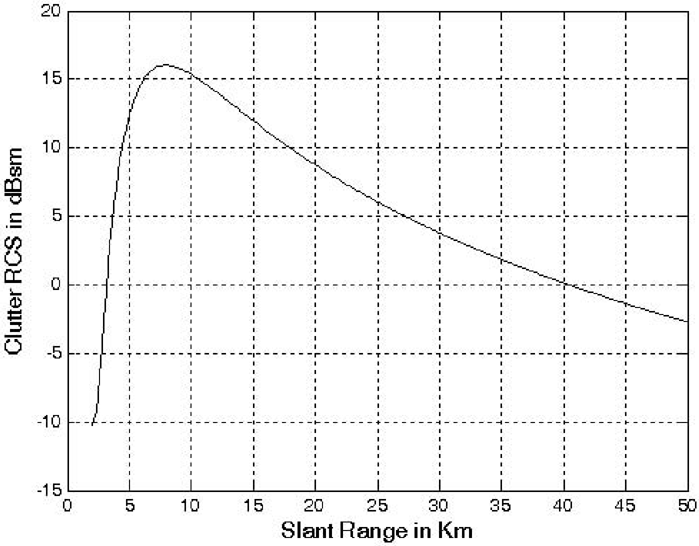


Figure 6.9a. Clutter RCS versus range using the function call in Eq. (6.28).

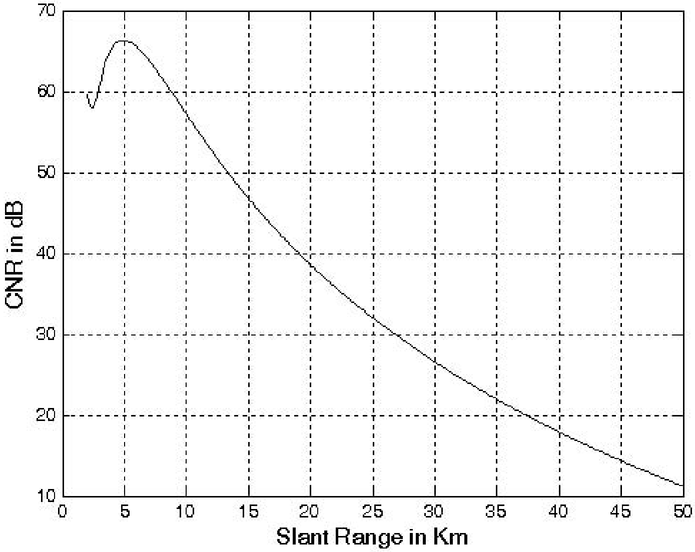
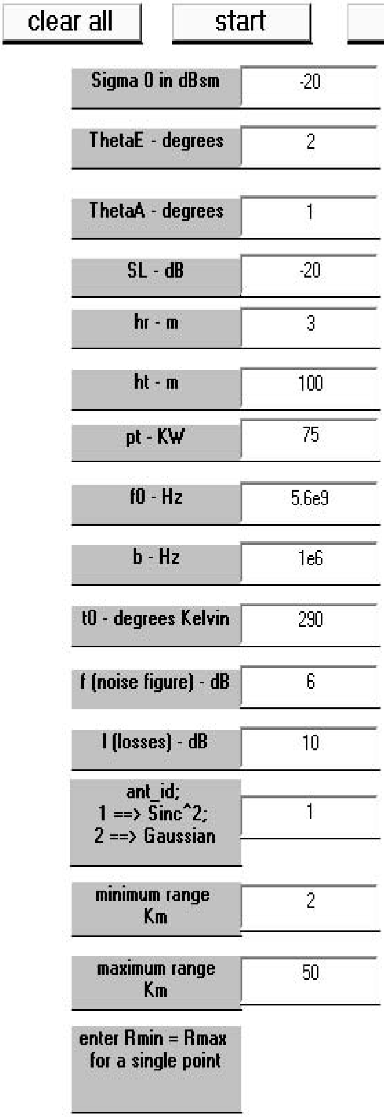


Figure 6.9b. CNR versus range using the function call in Eq. (6.28).



Sigma 0 in dBsm	-20
ThetaE - degrees	2
ThetaA - degrees	1
SL - dB	-20
hr - m	3
ht - m	100
pt - KW	75
f0 - Hz	5.6e9
b - Hz	1e6
t0 - degrees Kelvin	290
f (noise figure) - dB	6
l (losses) - dB	10

ant_id;
1 ==> Sinc^2;
2 ==> Gaussian
	1

minimum range Km	2
maximum range Km	50
enter Rmin = Rmax for a single point	

Figure 6.9c. GUI workspace for *"clutter_rcs_gui.m"*.

6.3. 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.

As mentioned earlier, chaff is used as an ECM technique by hostile forces. It consists of a large number of dipole reflectors with large RCS values. Historically, chaff was made of aluminum foil; however, in recent years most chaff is made of the more rigid fiberglass with conductive coating. The maximum chaff RCS occurs when the dipole length L is one half the radar wavelength.

Weather or rain clutter is easier to suppress than chaff, since rain droplets can be viewed 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:

$$\sigma = 9\pi r^2 (kr)^4 \qquad r \ll \lambda \tag{6.29}$$

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,

$$\eta = \sum_{i=1}^{N} \sigma_i \tag{6.30}$$

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

$$\sigma_W = \sum_{i=1}^{N} \sigma_i V_W \tag{6.31}$$

A resolution volume is shown in Fig. 6.10, and is approximated by

$$V_W \approx \frac{\pi}{8}\theta_a\theta_e R^2 c\tau \tag{6.32}$$

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

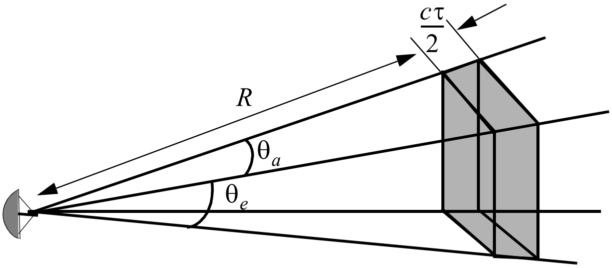


Figure 6.10. Definition of a resolution volume.

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

$$\sigma_i \approx \frac{\pi^5}{\lambda^4}K^2 D_i^6 \tag{6.33}$$

where

$$K^2 = \left|\frac{m^2-1}{m^2+2}\right|^2 \tag{6.34}$$

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

$$\sigma_i \approx 0.93\frac{\pi^5}{\lambda^4}D_i^6 \tag{6.35}$$

and for ice Eq. (6.33) can be approximated by

$$\sigma_i \approx 0.2\frac{\pi^5}{\lambda^4}D_i^6 \tag{6.36}$$

Substituting Eq. (6.33) into Eq. (6.30) yields

$$\eta = \frac{\pi^5}{\lambda^4} K^2 Z \tag{6.37}$$

where the weather clutter coefficient Z is defined as

$$Z = \sum_{i=1}^{N} D_i^6 \tag{6.38}$$

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 $millimeter^6/m^3$.

6.3.1. 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

$$S_t = \frac{P_t G^2 \lambda^2 \sigma_t}{(4\pi)^3 R^4} \tag{6.39}$$

where all parameters in Eq. (6.39) have been defined earlier. The weather clutter power received by the radar is

$$S_W = \frac{P_t G^2 \lambda^2 \sigma_W}{(4\pi)^3 R^4} \tag{6.40}$$

Using Eq. (6.31) and Eq. (6.32) in Eq. (6.40) and collecting terms yield

$$S_W = \frac{P_t G^2 \lambda^2}{(4\pi)^3 R^4} \frac{\pi}{8} R^2 \theta_a \theta_e c\tau \sum_{i=1}^{N} \sigma_i \tag{6.41}$$

The SCR for weather clutter is then computed by dividing Eq. (6.39) by Eq. (6.41). More precisely,

$$(SCR)_V = \frac{S_t}{S_W} = \frac{8\sigma_t}{\pi \theta_a \theta_e c\tau R^2 \sum_{i=1}^{N} \sigma_i} \tag{6.42}$$

where the subscript V is used to denote volume clutter.

Example:

A certain radar has target RCS $\sigma_t = 0.1m^2$*, pulsewidth* $\tau = 0.2\mu s$*, 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. (6.42) we have

$$(SCR)_V = \frac{8\sigma_t}{\pi\theta_a\theta_e c\tau R^2 \sum_{i=1}^{N} \sigma_i}$$

Substituting the proper values we get

$$(SCR)_V = \frac{(8)(0.1)}{\pi(0.02)^2(3 \times 10^8)(0.2 \times 10^{-6})(50 \times 10^3)^2(1.6 \times 10^{-8})} = 0.265$$

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

6.4. 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,

$$f(x) = \frac{2x}{x_0}\exp\left(\frac{-x^2}{x_0}\right) \quad ; \ x \geq 0 \tag{6.43}$$

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

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}\ x}\ \exp\left(-\frac{(\ln x - \ln x_m)^2}{2\sigma^2}\right) \quad ; \ x > 0 \tag{6.44}$$

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 $10 GHz$. 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

$$f(x) = \frac{bx^{b-1}}{\bar{\sigma}_0} \exp\left(-\frac{x^b}{\bar{\sigma}_0}\right) \quad ; \; x \geq 0 \tag{6.45}$$

where $b = 1/a$ is known as the shape parameter. Note that when $b = 2$ the Weibull distribution becomes a Rayleigh distribution.

6.5. "MyRadar" Design Case Study - Visit 6

6.5.1. Problem Statement

Analyze the impact of ground clutter on "MyRadar" design case study. Assume a Gaussian antenna pattern. Assume that the radar height is 5 meters. Consider an antenna sidelobe level $SL = -20 \;\; dB$ and a ground clutter coefficient $\sigma^0 = -15 \;\; dBsm$. What conclusions can you draw about the radar's ability to maintain proper detection and track of both targets? Assume a radar height $h_r \geq 5m$.

6.5.2. A Design

From the design processes established in Chapters 1 and 2, it was determined that the minimum single pulse SNR required to accomplish the design objectives was $SNR \geq \;\; 4dB$ when non-coherent integration (4 pulses) and cumulative detection were used. Factoring in the surface clutter will degrade the SIR. However, one must maintain $SIR \geq \;\; 4dB$ in order to achieve the desired probability of detection.

Figure 6.11 shows a plot of the clutter RCS versus range corresponding to *"MyRadar"* design requirements. This figure can be reproduced using the MATLAB GUI *"clutter_rcs_gui"* with the following inputs:

Symbol	**Value**	**Units**
sigma0	*-15*	*dB*
thetaE	*11 (see page 45)*	*degrees*

Symbol	Value	Units
thetaA	*1.33 (see page 45)*	*degrees*
SL	*-20*	*dB*
range	*linspace(10,120,1000)*	*Km*
hr	*5*	*meter*
ht	*2000 for missile; 10000 for aircraft*	*meter*
pt	*20*	*KW*
f0	*3e9*	*Hz*
b	*5e6*	*Hz*
t0	*290*	*Kelvins*
f	*6*	*dB*
l	*8*	*dB*
ant_id	*2 for Gaussian pattern*	*none*

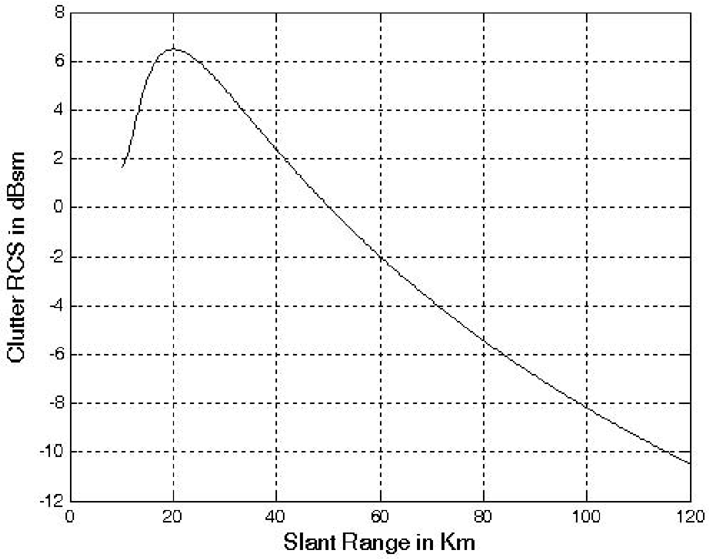


Figure 6.11a. Clutter RCS entering the radar for the missile case.

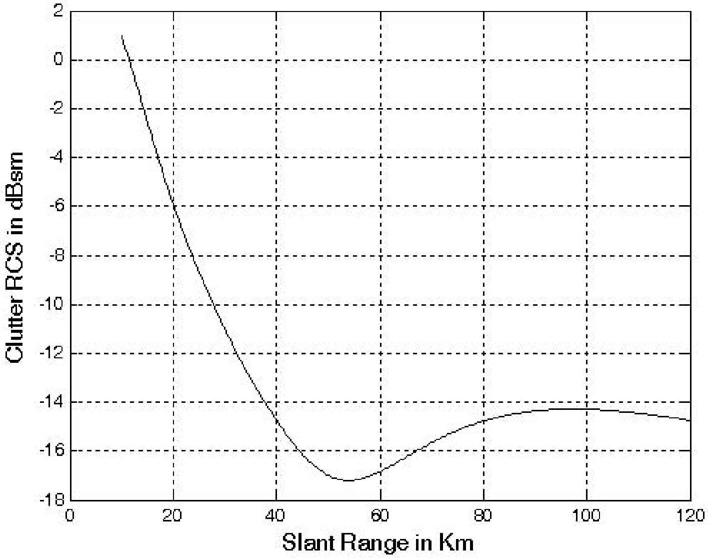


Figure 6.11b. Clutter RCS entering the radar for the aircraft case.

The MATLAB program *"myradar_visit6.m"* was developed to calculate and plot the CNR and SIR for *"MyRadar"* design case study. It is given in Listing 6.2 in Section 6.6. This program assumes the design parameters derived in Chapters 1 and 2. More precisely:

Symbol	**Description**	**Value**
σ^0	*clutter backscatter coefficient*	*-15 dBsm*
SL	*antenna sidelobe level*	*-20 dB*
σ_m	*missile RCS*	$0.5m^2$
σ_a	*aircraft RCS*	$4m^2$
θ_E	*antenna elevation beamwidth*	*11 deg*
θ_A	*antenna azimuth beamwidth*	*1.33 deg*
hr	*radar height*	*5 m*
hta	*target height (aircraft)*	*10 Km*
htm	*target height (missile)*	*2 Km*

Symbol	Description	Value
P_t	*radar peak power*	*20 KW*
f_0	*radar operating frequency*	*3GHz*
T_0	*effective noise temperature*	*290 degrees Kelvin*
F	*noise figure*	*6 dB*
L	*radar total losses*	*8 dB*
τ'	*Uncompressed pulsewidth*	*20 microseconds*

Figure 6.12 shows a plot of the CNR and the SIR associated with the missile. Figure 6.13 is similar to Fig. 6.12 except it is for the aircraft case. It is clear from these figures that the required SIR has been degraded significantly for the missile case and not as much for the aircraft case. This should not be surprising, since the missile's altitude is much smaller than that of the aircraft. Without clutter mitigation, the missile would not be detected at all. Alternatively, the aircraft detection is compromised at $R \le 80 Km$. Clutter mitigation is the subject of the next chapter.

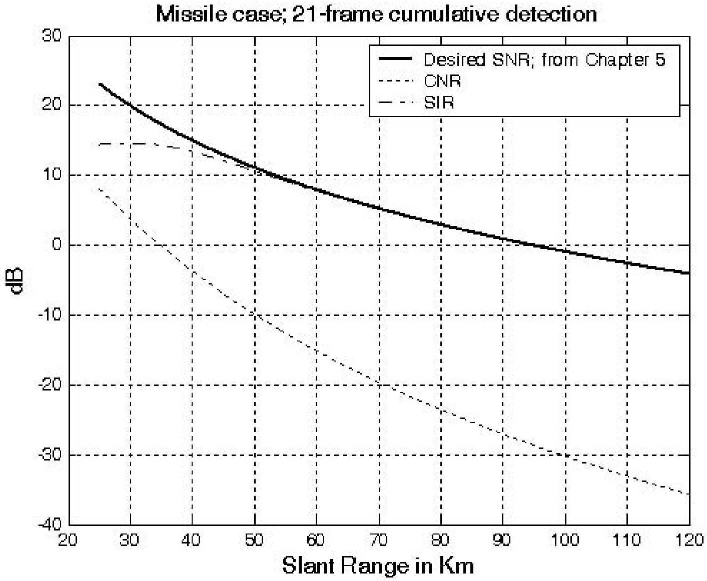


Figure 6.12. SNR, CNR, and SIR versus range for the missile case.

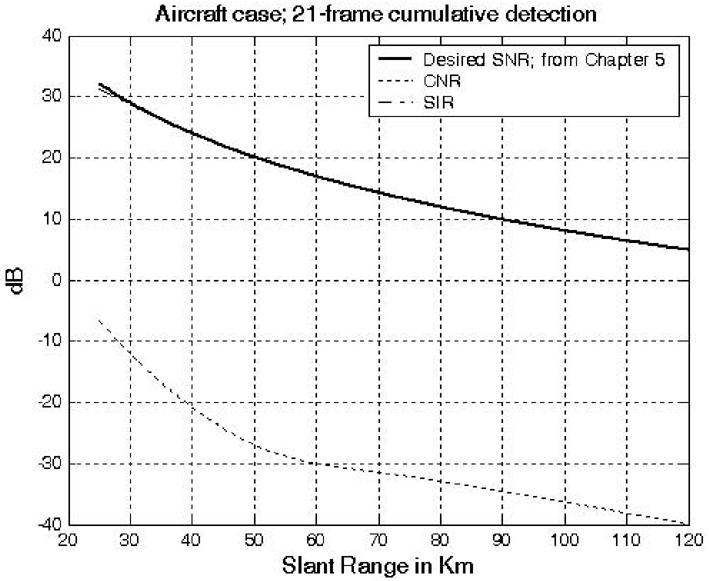


Figure 6.13. SNR, CNR and SIR versus range for the aircraft case.

6.6. MATLAB Program and Function Listings

This section presents listings for all MATLAB programs/functions used in this chapter. The user is advised to rerun these programs with different input parameters.

Listing 6.1. MATALB Function "clutter_rcs.m"

```
function [sigmaC,CNR] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, ht,
pt, f0, b,  t0, f, l,ant_id)
% This function calculates the clutter RCS and the CNR for a ground based
radar.
clight = 3.e8; % speed of light in meters per second
lambda = clight /f0;
thetaA_deg = thetaA;
thetaE_deg = thetaE;
thetaA = thetaA_deg * pi /180; % antenna azimuth beamwidth in radians
thetaE = thetaE_deg * pi /180.; % antenna elevation beamwidth in radians
re = 6371000; % earth radius in meters
rh = sqrt(8.0*hr*re/3.); % range to horizon in meters
```

```
SLv = 10.0^(SL/10); % radar rms sidelobes in volts
sigma0v = 10.0^(sigma0/10); % clutter backscatter coefficient
tau = 1/b; % pulsewidth
deltar = clight * tau / 2.; % range resolution for unmodulated pulse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
range_m = 1000 .* range;  % range in meters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
thetar = asin(hr ./ range_m);
thetae = asin((ht-hr) ./ range_m);
propag_atten = 1. + ((range_m ./ rh).^4); % propagation attenuation due to
round earth
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 = (2.78 * 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);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (size(range,2)==1)
   fprintf('Sigma_Clutter='); sigmaC
else
   figure(1)
   plot(range, sigmaC)
   grid
   xlabel('Slant Range in Km')
   ylabel('Clutter RCS in dBsm')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate CNR
pt = pt * 1000;
g = 26000 / (thetaA_deg*thetaE_deg); % antenna gain
F = 10.^(f/10); % noise figure is 6 dB
Lt = 10.^(l/10); % total radar losses 13 dB
k = 1.38e-23; % Boltzman’s constant
T0 = t0; % noise temperature 290K
```

```
argnumC = 10*log10(pt*g*g*lambda*lambda*tau .* sigmac);
argdem = 10*log10(((4*pi)^3)*k*T0*Lt*F .*(range_m).^4);
CNR = argnumC - argdem;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (size(range,2) ==1)
   fprintf('Cluuter_to_Noise_ratio='); CNR
 else
   figure(2)
   plot(range, CNR,'r')
   grid
   xlabel('Slant Range in Km')
   ylabel('CNR in dB')
 end
```

Listing 6.2. MATLAB Program “myradar_visit6.m”

```
clear all
close all
thetaA= 1.33; % antenna azimuth beamwidth in degrees
thetaE = 11; % antenna elevation beamwidth in degrees
hr = 5.; % radar height to center of antenna (phase reference) in meters
htm = 2000.; % target (missile) high in meters
hta = 10000.; % target (aircraft) high in meters
SL = -20; % radar rms sidelobes in dB
sigma0 = -15; % clutter backscatter coefficient
b = 1.0e6; %1-MHz bandwidth
t0 = 290; % noise temperature 290 degrees Kelvin
f0 = 3e9; % 3 GHz center frequency
pt = 114.6; % radar peak power in KW
f = 6; % 6 dB noise figure
l = 8; % 8 dB radar losses
range = linspace(25,120,500); % radar slant range 25 to 120 Km, 500 points
% calculate the clutter RCS and the associated CNR for both targets
[sigmaCa,CNRa] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, hta, pt,
f0, b, t0,  f, l, 2);
[sigmaCm,CNRm] = clutter_rcs(sigma0, thetaE, thetaA, L, range, hr, htm, pt,
f0, b, t0, f, l, 2);
close all
%%%%%%%%%%%%%%%%%%%%%%%%
np = 4;
pfa = 1e-7;
pdm = 0.99945;
pda = 0.99812;
% calculate the improvement factor
```

```
Im = improv_fac(np,pfa, pdm);
Ia = improv_fac(np, pfa, pda);
% calculate the integration loss
Lm = 10*log10(np) - Im;
La = 10*log10(np) - Ia;
pt = pt * 1000; % peak power in watts
range_m = 1000 .* range; % range in meters
g = 34.5139; % antenna gain in dB
sigmam = 0.5; % missile RCS m squared
sigmaa = 4; % aircraft RCS m squared
nf = f; %noise figure in dB
loss = l; % radar losses in dB
losstm = loss + Lm; % total loss for missile
lossta = loss + La; % total loss for aircraft
% modify pt by np*pt to account for pulse integration
SNRm = radar_eq(np*pt, f0, g, sigmam, t0, b, nf, losstm, range_m);
SNRa = radar_eq(np*pt, f0, g, sigmaa, t0, b, nf, lossta, range_m);
snrm = 10.^(SNRm./10);
snra = 10.^(SNRa./10);
cnrm = 10.^(CNRm./10);
cnra = 10.^(CNRa./10);
SIRm = 10*log10(snrm ./ (1+cnrm));
SIRa = 10*log10(snra ./ (1+cnra));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
plot(range, SNRm,'k', range, CNRm,'k :', range,SIRm,'k -.')
grid
legend('Desired SNR; from Chapter 5','CNR','SIR')
xlabel('Slant Range in Km')
ylabel('dB')
title('Missile case; 21-frame cumulative detection')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(4)
plot(range, SNRa,'k', range, CNRa,'k :', range,SIRa,'k -.')
grid
legend('Desired SNR; from Chapter 5','CNR','SIR')
xlabel('Slant Range in Km')
ylabel('dB')
title('Aircraft case; 21-frame cumulative detection')
```

Chapter 7

Moving Target Indicator (MTI) and Clutter Mitigation

7.1. Clutter Spectrum

The power spectrum of stationary clutter (zero Doppler) can be represented by a delta function. However, clutter is not always stationary; it actually exhibits some Doppler frequency spread because of wind speed and motion of the radar scanning antenna. In general, the clutter spectrum is concentrated around $f = 0$ and integer multiples of the radar PRF f_r, and may exhibit a small amount of spreading.

The clutter power spectrum can be written as the sum of fixed (stationary) and random (due to frequency spreading) components. For most cases, the random component is Gaussian. If we denote the stationary-to-random power ratio by W^2, then we can write the clutter spectrum as

$$S_c(\omega) = \bar{\sigma}_0\left(\frac{W^2}{1+W^2}\right)\delta(\omega_0) + \frac{\bar{\sigma}_0}{(1+W^2)\sqrt{2\pi\sigma_\omega^2}}\ \exp\left(-\frac{(\omega-\omega_0)^2}{2\sigma_\omega^2}\right) \tag{7.1}$$

where $\omega_0 = 2\pi f_0$ is the radar operating frequency in radians per second, σ_ω is the rms frequency spread component (determines the Doppler frequency spread), and $\bar{\sigma}_0$ is the Weibull parameter.

The first term of the right-hand side of Eq. (7.1) represents the PSD for stationary clutter, while the second term accounts for the frequency spreading. Nevertheless, since most of the clutter power is concentrated around zero Doppler with some spreading (typically less than 100 Hz), it is customary to model clutter using a Gaussian-shaped power spectrum (which is easier to analyze than Eq. (7.1)). More precisely,

$$S_c(\omega) = \frac{P_c}{\sqrt{2\pi\sigma_\omega^2}} \exp\left(-\frac{(\omega - \omega_0)^2}{2\sigma_\omega^2}\right) \tag{7.2}$$

where P_c is the total clutter power; σ_ω^2 and ω_0 were defined earlier. Fig. 7.1 shows a typical PSD sketch of radar returns when both target and clutter are present. Note that the clutter power is concentrated around DC and integer multiples of the PRF.

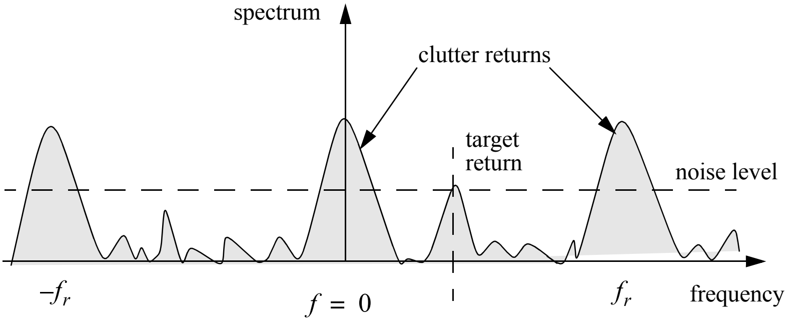


Figure 7.1. Typical radar return PSD when clutter and target are present.

7.2. Moving Target Indicator (MTI)

The clutter spectrum is normally concentrated around DC ($f = 0$) and multiple integers of the radar PRF f_r, as illustrated in Fig. 7.2a. 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 slowly 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. Fig. 7.2b shows a typical sketch of an MTI filter response, while Fig. 7.2c shows its output when the PSD shown in Fig. 7.2a 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,

$$v_{blind} = \frac{\lambda f_r}{2} \;\; ; \;\; n \geq 0 \tag{7.3}$$

Radar systems can minimize the occurrence of blind speeds by either employing multiple PRF schemes (PRF staggering) or by using high PRFs where in this case 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.

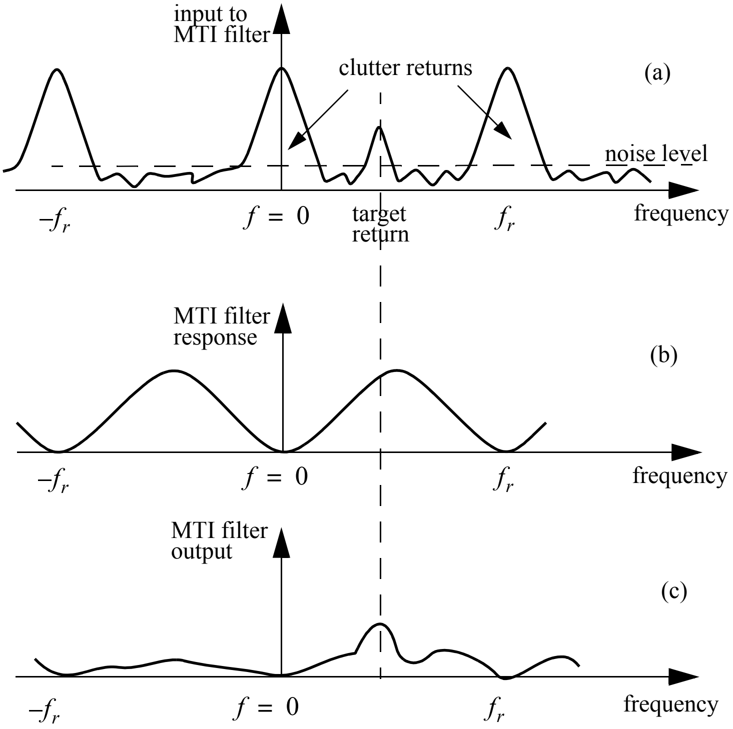


Figure 7.2. (a) Typical radar return PSD when clutter and target are present. (b) MTI filter frequency response. (c) Output from an MTI filter.

Fig. 7.3 shows a block diagram of a coherent MTI radar. Coherent transmission is controlled by the STAble Local Oscillator (STALO). The outputs of the STALO, f_{LO}, and the COHerent Oscillator (COHO), f_C, are mixed to produce the transmission frequency, $f_{LO} + f_C$. The Intermediate Frequency (IF), $f_C \pm f_d$, is produced by mixing the received signal with f_{LO}. After the IF amplifier, the signal is passed through a phase detector and is converted into a base band. Finally, the video signal is inputted into an MTI filter.

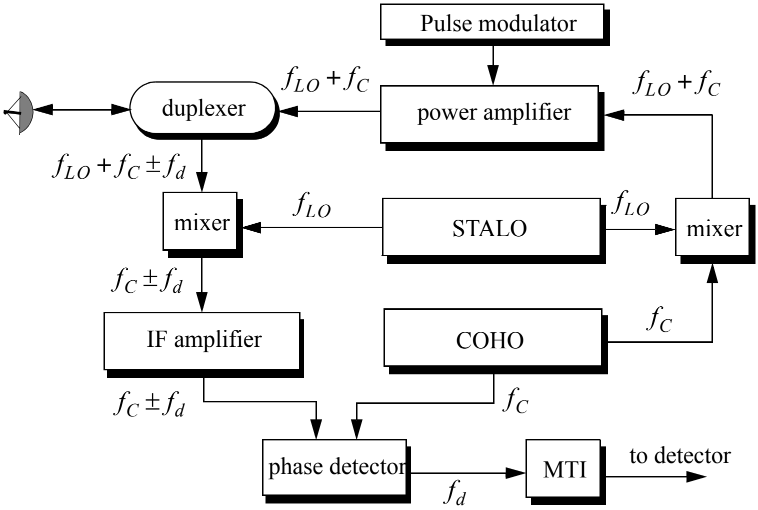


Figure 7.3. Coherent MTI radar block diagram.

7.3. Single Delay Line Canceler

A single delay line canceler can be implemented as shown in Fig. 7.4. 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.

The delay T is equal to the PRI of the radar $(1/f_r)$. The output signal $y(t)$ is

$$y(t) = x(t) - x(t - T) \tag{7.4}$$

The impulse response of the canceler is given by

$$h(t) = \delta(t) - \delta(t - T) \tag{7.5}$$

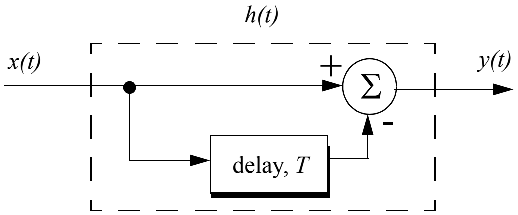


Figure 7.4. Single delay line canceler.

where $\delta(\quad\cdot\quad)$ is the delta function. It follows that the Fourier transform (FT) of $h(t)$ is

$$H(\omega) = 1 - e^{-j\omega T} \tag{7.6}$$

where $\omega = 2\pi f$.

In the z-domain, the single delay line canceler response is

$$H(z) = 1 - z^{-1} \tag{7.7}$$

The power gain for the single delay line canceler is given by

$$|H(\omega)|^2 = H(\omega)H^*(\omega) = (1 - e^{-j\omega T})(1 - e^{j\omega T}) \tag{7.8}$$

It follows that

$$|H(\omega)|^2 = 1 + 1 - (e^{j\omega T} + e^{-j\omega T}) = 2(1 - \cos\omega T) \tag{7.9}$$

and using the trigonometric identity $(2 - 2\cos 2\vartheta) = 4(\sin\vartheta)^2$ yields

$$|H(\omega)|^2 = 4(\sin(\omega T/2))^2 \tag{7.10}$$

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. It is given in Listing 7.1 in Section 7.11. The syntax is as follows:

[resp] = single_canceler (fofr)

where *fofr* is the number of periods desired. Typical output of the function *"single_canceler.m"* is shown in Fig. 7.5. 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 $n \geq 0$.

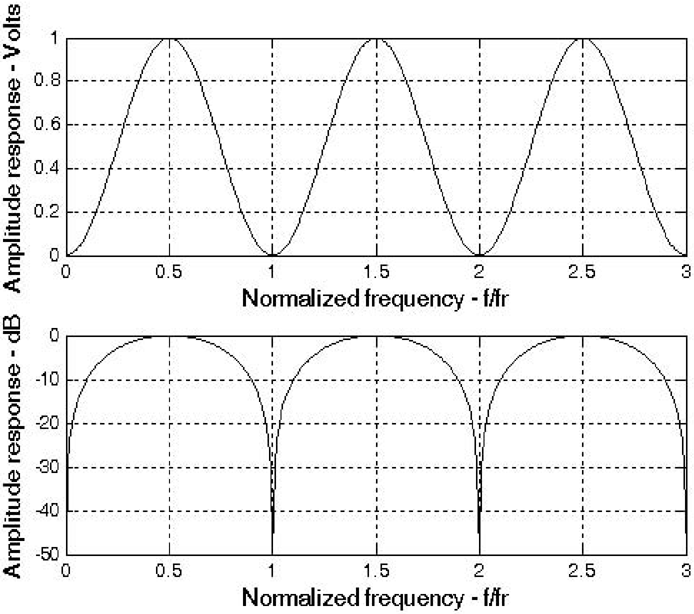


Figure 7.5. Single canceler frequency response.

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.

7.4. Double Delay Line Canceler

Two basic configurations of a double delay line canceler are shown in Fig. 7.6. 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

$$h(t) = \delta(t) - 2\delta(t-T) + \delta(t-2T) \tag{7.11}$$

Again, the names "double delay line" canceler and "double canceler" will be used interchangeably. The power gain for the double delay line canceler is

$$|H(\omega)|^2 = |H_1(\omega)|^2 |H_1(\omega)|^2 \tag{7.12}$$

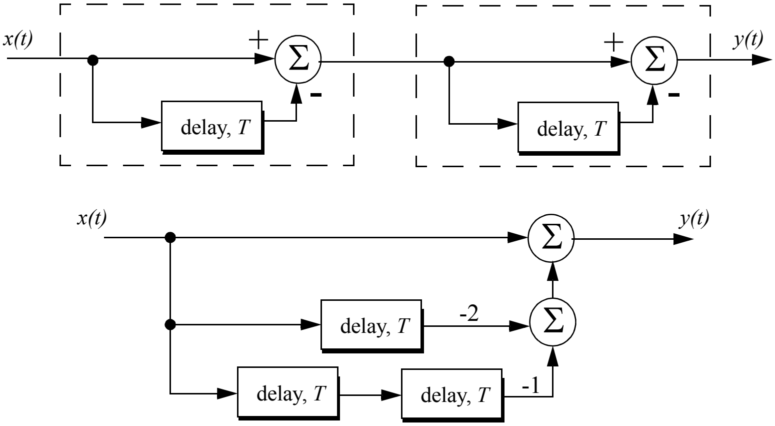


Figure 7.6. Two configurations for a double delay line canceler.

where $|H_1(\omega)|^2$ is the single line canceler power gain given in Eq. (7.10). It follows that

$$|H(\omega)|^2 = 16\left(\sin\left(\omega\frac{T}{2}\right)\right)^4 \tag{7.13}$$

And in the z-domain, we have

$$H(z) = (1 - z^{-1})^2 = 1 - 2z^{-1} + z^{-2} \tag{7.14}$$

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. It is given in Listing 7.2 in Section 7.11. The syntax is as follows:

[resp] = double_canceler (fofr)

where *fofr* is the number of periods desired.

Fig. 7.7 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).

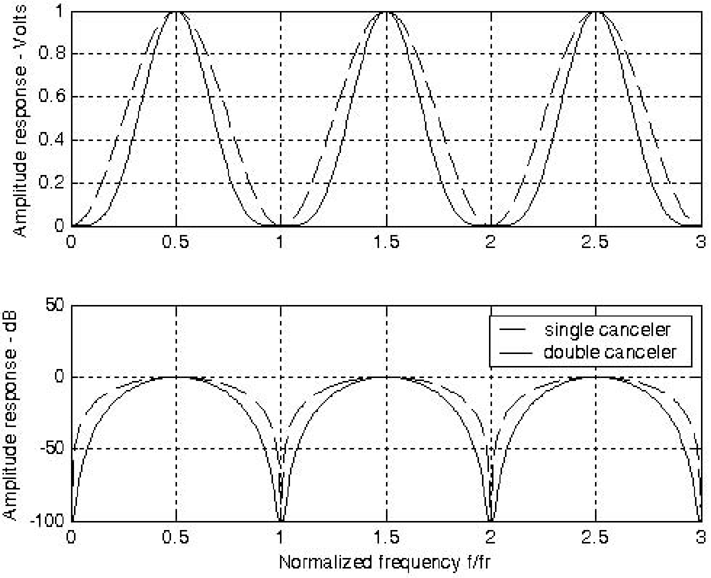


Figure 7.7. Normalized frequency responses for single and double cancelers.

7.5. 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. 7.8. From the figure we can write

$$y(t) = x(t) - (1-K)w(t) \tag{7.15}$$

$$v(t) = y(t) + w(t) \tag{7.16}$$

$$w(t) = v(t-T) \tag{7.17}$$

Applying the z-transform to the above three equations yields

$$Y(z) = X(z) - (1-K)W(z) \tag{7.18}$$

$$V(z) = Y(z) + W(z) \tag{7.19}$$

$$W(z) = z^{-1}V(z) \tag{7.20}$$

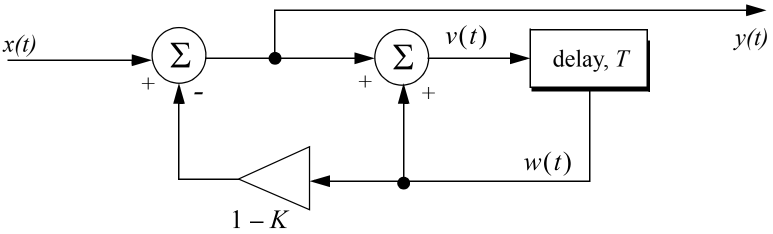


Figure 7.8. MTI recursive filter.

Solving for the transfer function $H(z) = Y(z)/X(z)$ yields

$$H(z) = \frac{1 - z^{-1}}{1 - Kz^{-1}} \tag{7.21}$$

The modulus square of $H(z)$ is then equal to

$$|H(z)|^2 = \frac{(1 - z^{-1})(1 - z)}{(1 - Kz^{-1})(1 - Kz)} = \frac{2 - (z + z^{-1})}{(1 + K^2) - K(z + z^{-1})} \tag{7.22}$$

Using the transformation $z = e^{j\omega T}$ yields

$$z + z^{-1} = 2\cos\omega T \tag{7.23}$$

Thus, Eq. (7.22) can now be rewritten as

$$\left|H(e^{j\omega T})\right|^2 = \frac{2(1 - \cos\omega T)}{(1 + K^2) - 2K\cos(\omega T)} \tag{7.24}$$

Note that when $K = 0$, Eq. (7.24) collapses to Eq. (7.10) (single line canceler). Fig. 7.9 shows a plot of Eq. (7.24) for $K = 0.25, 0.7, 0.9$. Clearly, by changing the gain factor K one can control the filter response.

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.

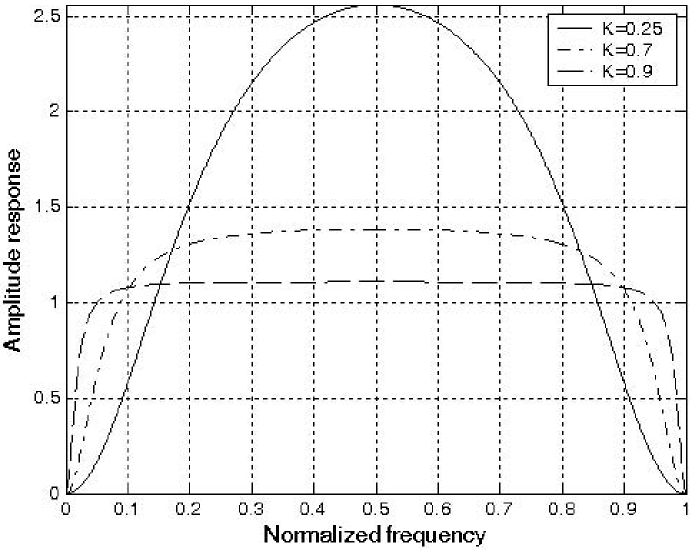


Figure 7.9. Frequency response corresponding to Eq. (7.24). This plot can be reproduced using MATLAB program *"fig7_9.m"* given in Listing 7.3 in Section 7.11.

7.6. 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 (see Fig. 7.7). 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 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

$$\frac{T_1}{T_2} = \frac{n_1}{n_2} \tag{7.25}$$

where n_1 and n_2 are integers. The first true blind speed occurs when

$$\frac{n_1}{T_1} = \frac{n_2}{T_2} \tag{7.26}$$

This is illustrated in Fig. 7.10 for $n_1 = 4$ and $n_2 = 5$. Note that if $n_2 = n_1 + 1$, then the process of PRF staggering is similar to that discussed in Chapter 3. The ratio

$$k_s = \frac{n_1}{n_2} \tag{7.27}$$

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, as illustrated in Fig. 7.11 for stagger ratio $k_s = 63/64$. In general, if there are N PRFs related by

$$\frac{n_1}{T_1} = \frac{n_2}{T_2} = \dots = \frac{n_N}{T_N} \tag{7.28}$$

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

$$v_{blind} = \frac{n_1 + n_2 + \dots + n_N}{N} v_{blind1} \tag{7.29}$$

7.7. 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 MTI “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,

$$CA = C_i/C_o \tag{7.30}$$

The MTI improvement factor is defined as the ratio of the Signal to Clutter (SCR) at the output to the SCR at the input,

$$I = \left(\frac{S_o}{C_o}\right) / \left(\frac{S_i}{C_i}\right) \tag{7.31}$$

which can be rewritten as

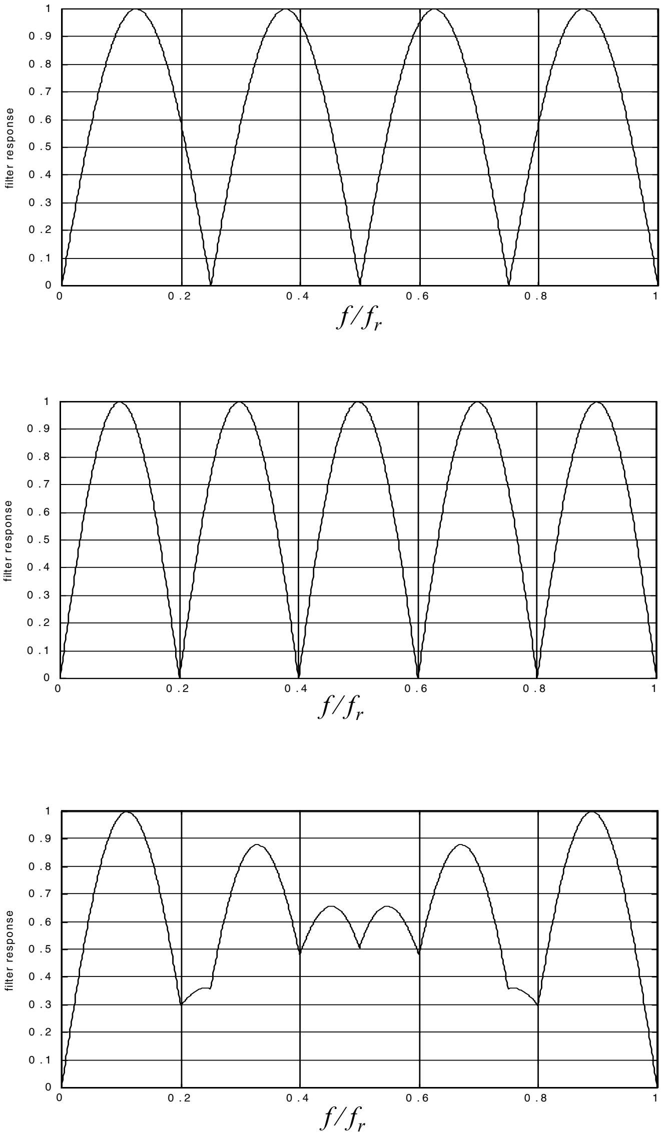


Figure 7.10. Frequency responses of a single canceler. Top plot corresponds to T_1, middle plot corresponds to T_2, bottom plot corresponds to stagger ratio $T_1/T_2 = 4/3$. This plot can be reproduced using MATLAB program *"fig7_10.m"* given in Listing 7.4 in Section 7.11.

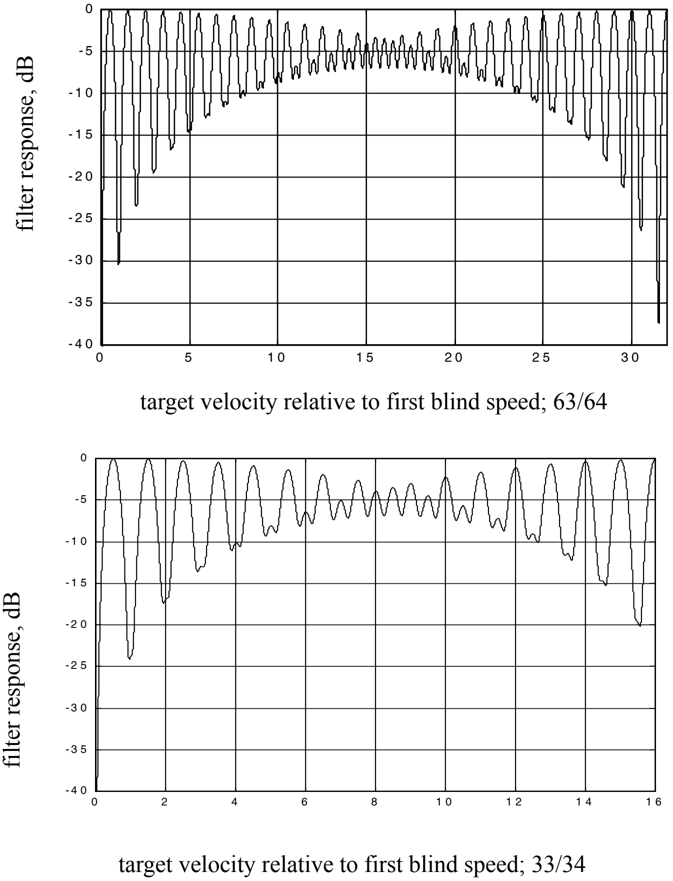


Figure 7.11. MTI responses, staggering ratio 63/64. This plot can be reproduced using MATLAB program *"fig7_11.m"* given in Listing 7.5 in Section 7.11.

$$I = \frac{S_o}{S_i} CA \tag{7.32}$$

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 is developed. A Gaussian-shaped clutter power spectrum is given by

$$W(f) = \frac{P_c}{\sqrt{2\pi}\ \sigma_t} \exp(-f^2/2\sigma_t^2) \tag{7.33}$$

where P_c is the clutter power (constant), and σ_t is the clutter rms frequency (which describes the clutter spectrum spread in the frequency domain). It is given by

$$\sigma_t = \sqrt{\sigma_v^2 + \sigma_s^2 + \sigma_w^2} \tag{7.34}$$

σ_v is the standard deviation for the clutter spectrum spread due to wind velocity; σ_s is the standard deviation for the clutter spectrum spread due to antenna scanning; and σ_v is the standard deviation for the clutter spectrum spread due to the radar platform motion (if applicable). It can be shown that[1]

$$\sigma_v = \frac{2\sigma_w}{\lambda} \tag{7.35}$$

$$\sigma_s = 0.265\left(\frac{2\pi}{\Theta_a T_{scan}}\right) \tag{7.36}$$

$$\sigma_s \approx \frac{v}{\lambda} \sin\theta \tag{7.37}$$

where λ is the wavelength and σ_w is the wind rms velocity; Θ_a is the antenna 3-db azimuth beamwidth (in radians); T_{scan} is the antenna scan time; v is the platform velocity; and θ is the azimuth angle (in radians) relative to the direction of motion.

The clutter power at the input of an MTI filter is

$$C_i = \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\ \sigma_t} \exp\left(-\frac{f^2}{2\sigma_t^2}\right)\ df \tag{7.38}$$

Factoring out the constant P_c yields

$$C_i = P_c \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{f^2}{2\sigma_t^2}\right) df \tag{7.39}$$

It follows that

[1] Berkowtiz, R. S., *Modern Radar, Analysis, Evaluation, and System Deign*, John Wiley & Sons, New York, 1965.

$$C_i = P_c \tag{7.40}$$

The clutter power at the output of an MTI is

$$C_o = \int_{-\infty}^{\infty} W(f)|H(f)|^2 \; df \tag{7.41}$$

7.7.1. 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 given by Eq. (7.6). The single canceler power gain is given in Eq. (7.10), which will be repeated here, in terms of f rather than ω, as Eq. (7.42),

$$|H(f)|^2 = 4\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^2 \tag{7.42}$$

It follows that

$$C_o = \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\;\sigma_t} \exp\left(-\frac{f^2}{2\sigma_t^2}\right) 4\left(\sin\left(\frac{\pi f}{f_r}\right)\right)^2 \; df \tag{7.43}$$

Now, since clutter power will only be significant for small f, then the ratio f/f_r is very small (i.e., $\sigma_t \ll f_r$). Consequently, by using the small angle approximation, Eq. (7.43) is approximated by

$$C_o \approx \int_{-\infty}^{\infty} \frac{P_c}{\sqrt{2\pi}\;\sigma_t} \exp\left(-\frac{f^2}{2\sigma_t^2}\right) 4\left(\frac{\pi f}{f_r}\right)^2 \; df \tag{7.44}$$

which can be rewritten as

$$C_o = \frac{4P_c\pi^2}{f_r^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{f^2}{2\sigma_t^2}\right) f^2 \; df \tag{7.45}$$

The integral part in Eq. (7.45) is the second moment of a zero mean Gaussian distribution with variance σ_t^2. Replacing the integral in Eq. (7.45) by σ_t^2 yields

$$C_o = \frac{4P_c\pi^2}{f_r^2}\;\sigma_t^2 \tag{7.46}$$

Substituting Eqs. (7.46) and (7.40) into Eq. (7.30) produces

$$CA = \frac{C_i}{C_o} = \left(\frac{f_r}{2\pi\sigma_t}\right)^2 \tag{7.47}$$

It follows that the improvement factor for a single canceler is

$$I = \left(\frac{f_r}{2\pi\sigma_t}\right)^2 \frac{S_o}{S_i} \tag{7.48}$$

The power gain ratio for a single canceler is (remember that $|H(f)|$ is periodic with period f_r)

$$\frac{S_o}{S_i} = |H(f)|^2 = \frac{1}{f_r}\int_{-f_r/2}^{f_r/2} 4\left(\sin\frac{\pi f}{f_r}\right)^2 df \tag{7.49}$$

Using the trigonometric identity $(2 - 2\cos 2\vartheta) = 4(\sin\vartheta)^2$ yields

$$|H(f)|^2 = \frac{1}{f_r}\int_{-f_r/2}^{f_r/2} \left(2 - 2\cos\frac{2\pi f}{f_r}\right) df = 2 \tag{7.50}$$

It follows that

$$I = 2\left(\frac{f_r}{2\pi\sigma_t}\right)^2 \tag{7.51}$$

The expression given in Eq. (7.51) is an approximation valid only for $\sigma_t \ll f_r$. When the condition $\sigma_t \ll f_r$ 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 $\sigma_v = 6.4Hz$ (wooded hills with $\sigma_w = 1.16311Km/hr$), find the improvement factor when a single delay line canceler is used.

Solution:

In this case $\sigma_t = \sigma_v$. It follows that the clutter attenuation CA is

$$CA = \left(\frac{f_r}{2\pi\sigma_t}\right)^2 = \left(\frac{800}{(2\pi)(6.4)}\right)^2 = 395.771 = 25.974dB$$

and since $S_o/S_i = 2 = 3dB$ we get

$$I_{dB} = (CA + S_o/S_i)_{dB} = 3 + 25.97 = 28.974dB.$$

7.7.2. The General Case

A general expression for the improvement factor for the n-pulse MTI (shown for a 2-pulse MTI in Eq. (7.51)) is given by

$$I = \frac{1}{Q^2(2(n-1)-1)!!}\left(\frac{f_r}{2\pi\sigma_t}\right)^{2(n-1)} \tag{7.52}$$

where the double factorial notation is defined by

$$(2n-1)!! = 1 \times 3 \times 5 \times \ldots \times (2n-1) \tag{7.53}$$

$$(2n)!! = 2 \times 4 \times \ldots \times 2n \tag{7.54}$$

Of course $0!! = 1$; Q is defined by

$$Q^2 = \frac{1}{\displaystyle\sum_{i=1}^{n} A_i^2} \tag{7.55}$$

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

$$\left\{\frac{1}{2}, \frac{1}{20}, \frac{1}{70}\right\} \tag{7.56}$$

Using this notation, then the improvement factor for a 3-pulse and 4-pulse MTI are respectively given by

$$I_{3-pulse} = 2\left(\frac{f_r}{2\pi\sigma_t}\right)^4 \tag{7.57}$$

$$I_{4-pulse} = \frac{4}{3}\left(\frac{f_r}{2\pi\sigma_t}\right)^6 \tag{7.58}$$

7.8. "MyRadar" Design Case Study - Visit 7

7.8.1. Problem Statement

The impact of surface clutter on the "MyRadar" design case study was analyzed. Assume that the wind rms velocity $\sigma_w = 0.45 m/s$. *Propose a clutter mitigation process utilizing a 2-pulse and a 3-pulse MTI. All other parameters are as calculated in the previous chapters.*

7.8.2. A Design

In earlier chapters we determined that the wavelength is $\lambda = 0.1m$, the PRF is $f_r = 1KHz$, the scan rate is $T_{scan} = 2s$, and the antenna azimuth 3-db beamwidth is $\Theta_a = 1.3°$. It follows that

$$\sigma_v = \frac{2\sigma_w}{\lambda} = \frac{2 \times 0.45}{0.1} = 9Hz \tag{7.59}$$

$$\sigma_s = 0.265\left(\frac{2\pi}{\Theta_a T_{scan}}\right) = 0.265 \times \frac{2 \times \pi}{1.32 \times \frac{\pi}{180} \times 2} = 36.136Hz \tag{7.60}$$

Thus, the total clutter rms spectrum spread is

$$\sigma_t = \sqrt{\sigma_v^2 + \sigma_s^2} = \sqrt{81 + 1305.810} = \sqrt{1386.810} = 37.24Hz \tag{7.61}$$

The expected clutter attenuation using a 2-pulse and a 3-pulse MTI are respectively given by

$$I_{2pulse} = 2\left(\frac{f_r}{2\pi\sigma_t}\right)^2 = 2 \times \left(\frac{1000}{2 \times \pi \times 37.24}\right)^2 = 36.531\frac{W}{W} \Rightarrow 15.63dB \tag{7.62}$$

$$I_{3pulse} = 2\left(\frac{f_r}{2\pi\sigma_t}\right)^4 = 2 \times \left(\frac{1000}{2 \times \pi \times 37.24}\right)^4 = 667.247\frac{W}{W} \Rightarrow 28.24dB \tag{7.63}$$

To demonstrate the effect of a 2-pulse and 3-pulse MTI on *"MyRadar"* design case study, the MATLAB program *"myradar_visit7.m"* has been developed. It is given in Listing 7.6 in Section 7.5. This program utilizes the radar equation with pulse compression. In this case, the peak power was established in Chapter 5 as $P_t \leq 10KW$. Figs. 7.12 and 7.13 show the desired SNR and the calculated SIR using a 2-pulse and a 3-pulse MTI filter respectively, for the missile case. Figs. 7.14 and 7.15 show similar output for the aircraft case.

One may argue, depending on the tracking scheme adopted by the radar, that for a tracking radar

$$\sigma_t = \sigma_v = 9Hz \tag{7.64}$$

since $\sigma_s = 0$ for a radar that employes a monopulse tracking option. In this design, we will assume a Kalman filter tracker. For more details the reader is advised to visit Chapter 9.

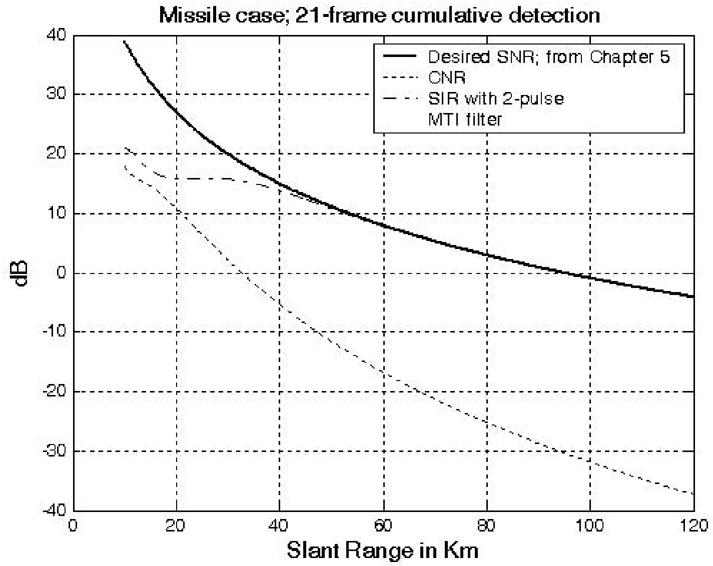


Figure 7.12. SIR for the missile case using a 2-pulse MTI filter.

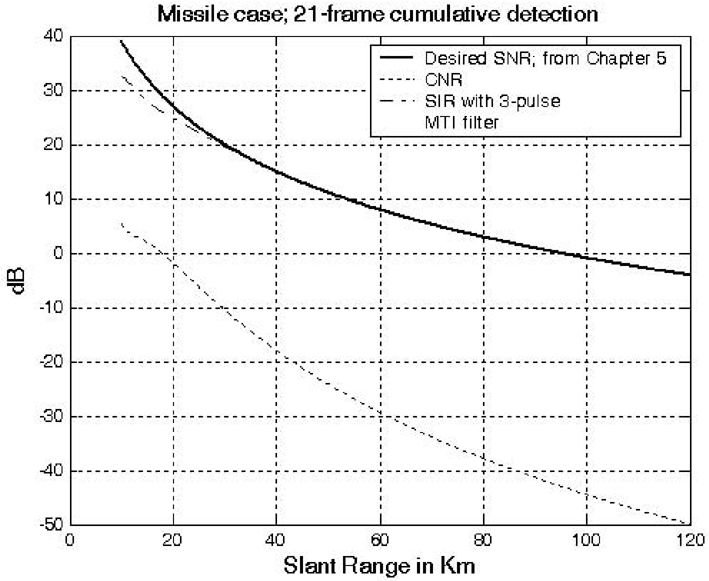


Figure 7.13. SIR for the missile case using a 3-pulse MTI filter.

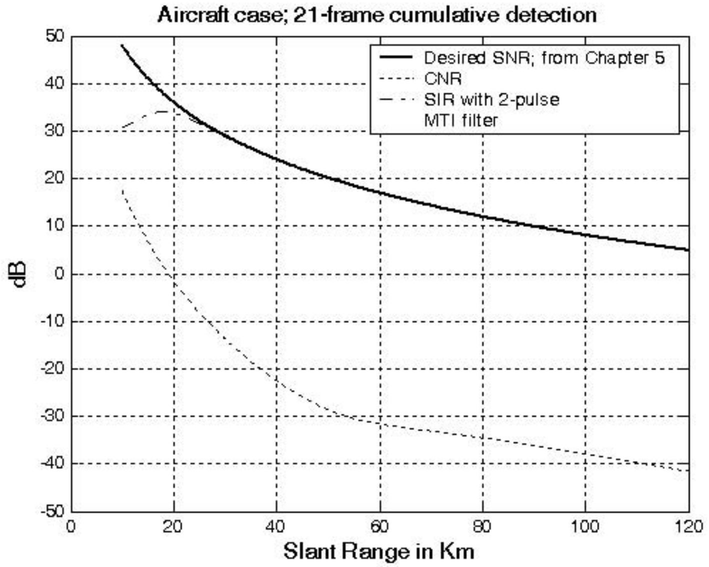


Figure 7.14. SIR for the aircraft case using a 2-pulse MTI filter.

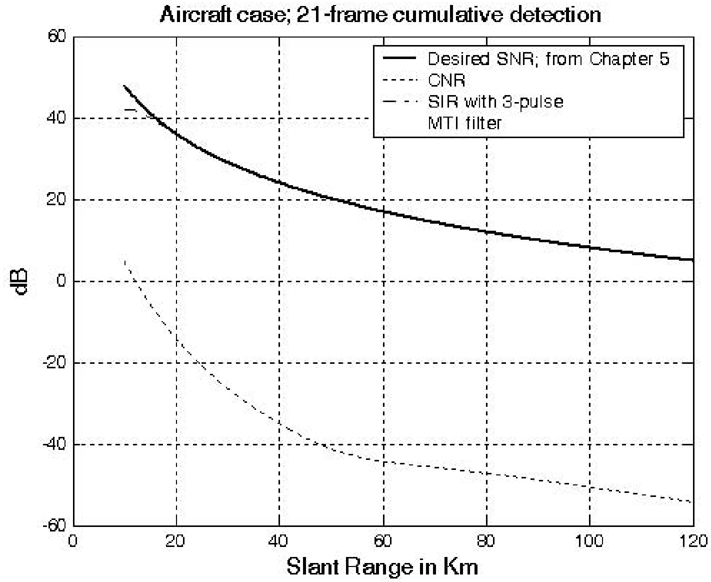


Figure 7.15. SIR for the aircraft case using a 3-pulse MTI filter.

As clearly indicated by the previous four figures, a 3-pulse MTI filter would provide adequate clutter rejection for both target types. However, if we assume that targets are detected at maximum range (90 Km for aircraft and 55 Km for missile) and then are tracked for the rest of the flight, then 2-pulse MTI may be adequate. This is true since the SNR would be expected to be larger during track than it is during detection, especially when pulse compression is used. Nonetheless, in this design a 3-pulse MTI filter is adopted.

7.9. MATLAB Program and Function Listings

This section contains listings of all MATLAB programs and functions used in this chapter. Users are encouraged to rerun this code with different inputs in order to enhance their understanding of the theory.

Listing 7.1. MATLAB Function "single_canceler.m"

```
function [resp] = single_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;
subplot(2,1,1)
plot(fofr,resp,'k')
xlabel ('Normalized frequency - f/fr')
ylabel( 'Amplitude response - Volts')
grid
subplot(2,1,2)
resp=10.*log10(resp+eps);
plot(fofr,resp,'k');
axis tight
grid
xlabel ('Normalized frequency - f/fr')
ylabel( 'Amplitude response - dB')
```

Listing 7.2. MATLAB Function "double_canceler.m"

```
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 - dB')
```

Listing 7.3. MATLAB Program “fig7_9.m”

```
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
```

Listing 7.4. MATLAB Program “fig7_10.m”

```
clear all
fofr = 0:0.001:1;
```

```
f1 = 4.0 .* fofr;
f2 = 5.0 .* fofr;
arg1 = pi .* f1;
arg2 = pi .* f2;
resp1 = abs(sin(arg1));
resp2 = abs(sin(arg2));
resp = resp1+resp2;
max1 = max(resp);
resp = resp./max1;
plot(fofr,resp1,fofr,resp2,fofr,resp);
xlabel('Normalized frequency f/fr')
ylabel('Filter response')
```

Listing 7.5. MATLAB Program "fig7_11.m"

```
clear all
fofr = 0.01:0.001:32;
a = 63.0 / 64.0;
term1 = (1. - 2.0 .* cos(a*2*pi*fofr) + cos(4*pi*fofr)).^2;
term2 = (-2. .* sin(a*2*pi*fofr) + sin(4*pi*fofr)).^2;
resp = 0.25 .* sqrt(term1 + term2);
resp = 10. .* log(resp);
plot(fofr,resp);
axis([0 32 -40 0]);
grid
```

Listing 7.6. MATLAB Program "myradar_visit7.m"

```
clear all
close all
clutter_attenuation = 28.24;
thetaA= 1.33; % antenna azimuth beamwidth in degrees
thetaE = 11; % antenna elevation beamwidth in degrees
hr = 5.; % radar height to center of antenna (phase reference) in meters
htm = 2000.; % target (missile) height in meters
hta = 10000.; % target (aircraft) height in meters
SL = -20; % radar rms sidelobes in dB
sigma0 = -15; % clutter backscatter coefficient in dB
b = 1.0e6; %1-MHz bandwidth
t0 = 290; % noise temperature 290 degrees Kelvin
f0 = 3e9; % 3 GHz center frequency
pt = 114.6; % radar peak power in KW
f = 6; % 6 dB noise figure
l = 8; % 8 dB radar losses
```

```
range = linspace(25,120,500); % radar slant range 25 to 120 Km, 500 points
% calculate the clutter RCS and the associated CNR for both targets
[sigmaCa,CNRa] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, hta, pt,
f0, b, t0, f, l,2);
[sigmaCm,CNRm] = clutter_rcs(sigma0, thetaE, thetaA, SL, range, hr, htm, pt,
f0, b, t0, f, l,2);
close all
%%%%%%%%%%%%%%%%%%%%%%%%
np = 4;
pfa = 1e-7;
pdm = 0.99945;
pda = 0.99812;
% calculate the improvement factor
Im = improv_fac(np,pfa, pdm);
Ia = improv_fac(np, pfa, pda);
% caculate the integration loss
Lm = 10*log10(np) - Im;
La = 10*log10(np) - Ia;
pt = pt * 1000; % peak power in watts
range_m = 1000 .* range; % range in meters
g = 34.5139; % antenna gain in dB
sigmam = 0.5; % missile RCS m squared
sigmaa = 4; % aircraft RCS m squared
nf = f; %noise figure in dB
loss = l; % radar losses in dB
losstm = loss + Lm; % total loss for missile
lossta = loss + La; % total loss for aircraft
% modify pt by np*pt to account for pulse integration
SNRm = radar_eq(np*pt, f0, g, sigmam, t0, b, nf, losstm, range_m);
SNRa = radar_eq(np*pt, f0, g, sigmaa, t0, b, nf, lossta, range_m);
snrm = 10.^(SNRm./10);
snra = 10.^(SNRa./10);
CNRm = CNRm - clutter_attenuation;
CNRa = CNRa - clutter_attenuation;
cnrm = 10.^(CNRm./10);
cnra = 10.^(CNRa./10);
SIRm = 10*log10(snrm ./ (1+cnrm));
SIRa = 10*log10(snra ./ (1+cnra));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
plot(range, SNRm,'k', range, CNRm,'k :', range,SIRm,'k -.')
grid
legend('Desired SNR; from Chapter 5','CNR','SIR with 3-pulse','MTI filter')
xlabel('Slant Range in Km')
```

```
ylabel('dB')
title('Missile case;  21-frame cumulative detection')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
plot(range, SNRa,'k', range, CNRa,'k :', range,SIRa,'k -.')
grid
legend('Desired SNR; from Chapter 5','CNR','SIR with 3-pulse','MTI filter')
xlabel('Slant Range in Km')
ylabel('dB')
title('Aircraft case; 21-frame cumulative detection')
```