Eyung W. Kang

radar system

analysis, design, and simulation

CD-ROM Included

Radar System Analysis, Design, and Simulation

Eyung W. Kang

ARTECH
HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalog record for this book is available from the British Library.

ISBN-13: 978-1-59693-347-7

685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

Radar System Analysis, Design, and Simulation

DISCLAIMER OF WARRANTY

The technical descriptions, procedures, and computer programs in this book have been developed with the greatest of care and they have been useful to the author in a broad range of applications; however, they are provided as is, without warranty of any kind. Artech House, Inc. and the author and editors of the book titled *Radar System Analysis, Design, and Simulation* make no warranties, expressed or implied, that the equations, programs, and procedures in this book or its associated software are free of error, or are consistent with any particular standard of merchantability, or will meet your requirements for any particular application. They should not be relied upon for solving a problem whose incorrect solution could result in injury to a person or loss of property. Any use of the programs or procedures in such a manner is at the user's own risk. The editors, author, and publisher disclaim all liability for direct, incidental, or consequent damages resulting from use of the programs or procedures in this book or the associated software.

For a listing of recent titles in the *Artech House Radar Library*, turn to the back of this book.

Preface

This book aims to reach radar system engineers who strive to optimize system performance and graduate students who wish to learn radar system design. It addresses the need to master system analysis and design skills and to verify that the analysis is correct and that the design is optimal through simulation on a digital computer. The prerequisites for understanding most of topics in this book are rather minimal: an elementary knowledge of probability theory and algebraic matrix operations and a basic familiarity with the computer.

Achieving the right balance between the depth and breadth of each chapter subject has been a difficult task. Readers may recognize that full coverage of each chapter title could fill its own a separate book of substantial volume. An attractive feature of this book is that it allows readers to build a solid foundation and increase their level of sophistication as their experience grows.

This book is intended for those readers who have an impatient urge to learn radar system analysis and design and C++ programming with applications. All examples are explained in detail in the text, and the numerical results are either displayed on screen or stored in .DAT files and shown in graphic form when appropriate. This book is not C++ for dummies; this book is for smart students and diligent engineers who are overloaded with other burdens.

So let's get started! The mountaintop is not that far away, and the path to be traversed to reach the top is not too arduous a climb.

Acknowledgments

This introductory book will help readers to build a solid foundation for further explorations in their chosen topics.

Many friends and colleagues helped me and encouraged me to complete the manuscript. I owe them a large debt of gratitude. I wish I could acknowledge them all by name, but they are too numerous.

I am most grateful to Mrs. Mary K. Darcy who patiently taught me how to type the mathematic equations clearly and concisely. Her indefatigable tutoring in presenting the graphs and diagrams made this book stand apart from all others with a similar title.

Introduction

Chapter 1: Matrix, Vector, and Linear Equations

Chapter 1 introduces matrix-vector operations. We start with how to solve a set of simultaneous linear equations by Gauss's elimination methods of the back-substitution and forward substitution that we learned in earlier years in school. When the dimension of equations is two or three we can solve for the unknowns with pencil and paper. When the dimension is higher than, say, four or higher we would rather resort to a computer program.

We encounter immediately the task of matrix inversion. Matrix inversion is usually permissible, but sometime it is not, depending upon the structure of the matrix. We learn when it is permissible by means of matrix factorization (decomposition). The factored matrices would clearly indicate whether the inversion is permissible. Several popular factorization routines and the corresponding inversions of factored matrices are programmed.

Vector operation is treated as a subset of matrix operations. Two header files, VECTOR.H and MATRIX.H, are constructed in order to determine when the matrix-vector operations are called for in the main driver.

Chapter 2: Pseudorandom Number (PRN), Noise, and Clutter Generation

The goal of this chapter is to generate various noises and clutters we would encounter in signal processing. The noise and clutter are characterized by the probability density function and its mean and variance. (For those who prefer electrical terms mean is DC voltage level and variance is AC power.) The basic building components of noise and clutter are unit uniform random variables, and the unit uniform variables are, in turn, generated from pseudo-random numbers (PRNs). There are a few prepackaged random number generators; however, careless use of these generators causes some intractable confusion in the results of signal processing.

We start with PRN generation by mixed congruential method. This method generates a random number vector without missing data nor duplicated data in a specified population N after a correction. It is contiguous in the given range. Some of the off-the-shelf packages seldom meet the requirement of contiguity.

The typical noise and clutter we encounter most often in signal processing, such as Gaussian, Rayleigh, Rician, exponential, and chi-squared, and lognormal, Weibull clutters, are generated. The computation of noise power or clutter power is emphasized. These noises or clutters are used in applications programs in such areas as filter design, pulse compression, Kalman filters, and the Monte Carlo technique.

Chapter 3: Filters, FIR, and IIR

In this chapter we program filter designs. Filters are broadly classified into two groups: finite duration impulse response filters, and infinite duration impulse response filters. The word duration is deleted, and we call them the finite impulse response (FIR) filters, or infinite impulse response (IIR) filters for short.

The design of a FIR filter is based on the inverse Fourier transform of the impulse response of lowpass, highpass, bandpass, and bandstop. The structure of the FIR filter is nonrecursive, and the phase response is a linear function of frequency. The design of the IIR filter is based on the analog filter designs abundantly available in textbooks and tables. The analog frequency response is transformed to the z-domain using bilinear transform and prewarping the response. The structure is recursive, and the phase versus frequency is nonlinear.

Several window functions (the envelope functions) to control the level of sidelobes are described in detail and incorporated in the design, and the advantages and disadvantages of FIR and IIR are discussed.

Chapter 4: Fast Fourier Transform (FFT) and IFFT

The real-time processing of Fourier transforms became possible when the fast Fourier transform algorithm was discovered in the early 1960s. Prior to this discovery the "real-time on-line" computation of frequency content had been impractical.

The fast Fourier transform is implemented either by decimation-in-time (DIT) or decimation-in-frequency (DIF) algorithm, with a suitable bit-reversal operation. The proper sequence of the bit-reverse operation is shown in a block diagram to show the proper order of bit-reversal in DIT and DIF.

An improper sampling of signals would produce spectral leakage and unfaithful reproduction of time function. A detailed discussion to remedy these problems are given, and examples are shown.

Applications of the fast Fourier transform and inverse transform in signal processing are numerous and include determining how to extract a signal buried in noise, how to compress a frequency-modulated pulse to improve resolution, and how to correct an unbalanced and mismatched I/Q channels of a coherent receiver. We present a few of these interesting applications.

A clear explanation together with a system block diagram will help readers to understand the concept involved. The results of programs are stored in .DAT files, and the corresponding graphs are shown.

Chapter 5: Ambiguity Function

This chapter is written for those who would like to explore the pulse compression in depth. We start with a rectangular pulse with constant carrier frequency and learn basic characteristics of a simple pulsed signal.

A popular signal for pulse compression is a coherent linear frequency-modulated (CLFM) pulse. Most high-performance radar systems have adopted the CLFM pulse, and the returned signal is compressed via FFT and IFFT algorithms in real time.

Analysis of the ambiguity function has indicated that there are many more modulation schemes to increase compression gain, to reduce sidelobe level, to improve resolution, and to maintain the low probability of intercept.

One ideal modulated pulse is the Costas-coded frequency-hopping modulation. We have programmed the Costas-coded signal and concluded that the Costas-coded signal is very nearly an ideal pulse. The correct sequence of frequency hopping is analyzed, and the result is presented in an ambiguity surface in three dimensions.

Chapter 6: Array Antennas

This chapter presents the array antenna design. A simple line array antenna is analyzed and programmed. A circular or elliptical array design is an extension of a line array design. The amplitude distributions of array elements are controlled by window functions such as Hamming, Chebyshev, Taylor, and Lambda. The maximum gain, 3-dB beamwidth, and the sidelobe levels are compared with a rectangle window. A similarity between the window functions in filter design is mentioned. A monopulse array antenna, an essential component of a tracking radar system, is followed.

Elimination (or cancelation) of mutual reactance coupling among array elements is excluded from the discussion, for the coupling is entirely dependent upon the physical structure of the element of radiation source and the geometric distribution of elements on the array aperture.

This chapter does not cover the intricacies and difficulties in designing and manufacturing an array antenna. However, it provides, under an ideal condition, what we expect of the maximum gain, the beamwidth, and the sidelobe levels when the dimension of the array aperture and window function are specified. The references are cited for those problems we have not addressed.

Chapter 7: Target Detection

This chapter presents the theory of the probability of detection and the probability of false alarm. The detection probability and the false alarm probability are introduced heuristically at the beginning in order to put readers at ease.

The detection processing involved in the heuristic model is shown in a recirculating accumulator. The relationship between the detection probability P_d, the false alarm probability P_{fa} and the threshold level V_{th} is programmed and discussed. The threshold level V_{th} (or bias level y_b) is a function of the number of pulse N to be accumulated (summed) in the recirculator-accumulator; the larger the number of pulses the lower the V_{th} (or y_b) for a specified P_{fa}.

The mathematic expression for P_{fa} is formally derived as a function of the bias level y_b. An expression of P_d is derived, in turn, as a function of both the number of pulses N and the signal-to-noise ratio S/N (in power) per pulse.

Four factors, P_d, P_{fa}, N, and S/N, maintain an interlocked relationship; that is, if we specify any combination of three factors, the fourth will be determined uniquely. One remaining problem is how to characterize the target signals. The retuned signals are grouped into five Marcum's and Swerling's targets in honor of the pioneer investigators.

Marcum's target is nonfluctuating, a constant cross-section target, like a large metallic sphere of several wavelengths in diameter. A large spherical target reflects a constant power irrespective of the aspect angle of the antenna beam. Two programs are written for Marcum's target, and the results are shown in a table as well as in the following graphic forms:

- Pd_(0)_1.CPP Detection probability, Marcum's target, N=1
- Pd_(0)_N.CPP Detection probability, Marcum's target, N≥2

Swerling has grouped the fluctuating targets in four different types: target model 1, 2, 3, and 4.

The probability density function of the target cross-section is assumed to be Rayleigh-distributed for target models 1 and 2, and the rate of fluctuation is either slow or rapid. Rayleigh-slow is Swerling's target 1, and Rayleigh-rapid is Swerling target 2.

The target cross-section of Swerling's 3 and 4 is assumed to be distributed chi-squared with four degrees of freedom, and the fluctuation is either slow or rapid. Slow or rapid fluctuation is a relative term with respect to the dynamics of target and the radar operation parameters such as transmitting frequency, the pulse repetition frequency (f_{PRF}), and the antenna rotation rate.

Eight programs are written for Swerling's targets. The false alarm probability is set at 1.0E-6, adjustable to 1.0E-5. The detection probability is computed as a function of S/N per pulse with the number of pulse N integrable as a variable parameter. The eight programs are listed as follows:

- Pd_(l)_1.CPP: Swerling target 1, N=1;
- Pd_(l)_N.CPP: Swerling target 1, N=2, 4, 8, 16, . . . ;
- Pd_(2)_1.CPP: Swerling target 2, N=1;
- Pd_(2)_N.CPP: Swerling target 2, N=2, 4, 8, 16, . . . ;
- Pd_(3)_1.CPP: Swerling target 3, N=1;
- Pd_(3)_N.CPP: Swerling target 3, N=2, 4, 8, 16, . . . ;
- Pd_(4)_1.CPP: Swerling target 4, N=1;
- Pd_(4)_N.CPP: Swerling target 4, N=2, 4, 8, 16, ...

The noise is for all programs Gaussian with zero mean and unity variance, which transformed to exponential after a square-law detector as presented in Chapter 2.

An example of a preliminary system design is given at the end. For instance, when we specify the transmitter frequency and power, the antenna gain, the hardware plumbing loss the RF section and the noise figure, the predetection bandwidth of the receiver, the number of pulses N integrable, at what distance might we be able to detect a target of $10m^2$ with a detection probability of 90% (or 50%) if the target is a nonfluctuating Marcum's target? We answer this question clearly. The detection range for the fluctuating Swerling's targets can be deduced from the programs given above.

The detection probability when the Gaussian noise is replaced by clutter returns is separately analyzed and programmed in Chapter 10, since the detection probability under clutter intrusion requires an additional theoretical development and processing implementation.

Chapter 8: Kalman Filter

We define the symbols and notations used in this chapter at the outset. This is to eliminate the confusion and frustration that might stem from reading a few textbooks and research papers by various authors.

The bold letters represent vectors or matrices. Some vectors and matrices have an overhead symbol, a caret (^) or a tilde (~). The caret signifies that the vector or matrix is an estimate. The tilde, on the other hand, signifies the difference between an estimate and the true state. In addition, the vector and matrix have subscript k–1, k, or k+1. The k–1 connotes the vector or matrix is a state of immediate past, the k the present. The k+1 is a predicted state, one sample time in the future. The vector or matrix without a subscript is time-invariant, and the vector or matrix without an overhead symbol is the true state, unknowable to the observer.

Example 1

We derive seven equations of Kalman filter through an example. A simple example of Kalman filter processing is a radar system that tracks a passenger airliner whose flight trajectory is a straight line at a constant altitude without any abrupt maneuver. We assume that the airliner experiences a small random deviation from an ideal straight line due to atmospheric disturbance and nonuniform engine thrust. We define the state equation of the airliner and the measurement (observation) equation by a radar. The atmospheric disturbance and nonuniform thrust are assumed to be small in magnitude, and the probability distribution is Gaussian. Two sources of measurement errors are due to finite transmitter pulsewidth (range error) and finite antenna beamwidth (azimuth error). The probability density function of the measurement errors is a unit uniform; that is, the error is most likely distributed equally and uniformly. Numerical examples are given whenever a new vector or matrix is introduced.

Three programs are written: an equation of a straight-line trajectory in x-y coordinates; a Gaussian deviation of zero mean and variance of 0.5g, which is added to the straight line to simulate the actual flight path; and the measurement equation by the radar. The estimated and predicted error covariance matrices $\mathbf{P}_k^+$ and $\mathbf{P}_{k+1}^-$ are derived as well as the Kalman gain matrix $\mathbf{k}_k$. The estimated state and predicted state, $\mathbf{x}_k^+$ and $\mathbf{x}_{k+1}^-$ (the position of airliner) are derived, with the Kalman gain matrix $\mathbf{k}_k$ as an input.

The filter processing is shown in a flow diagram. The filter has two recursive loops; one computes the estimated and predicted error covariance matrices, and the other loop computes the estimated and predicted states. The two loops are connected by a Kalman gain matrix.

A post-flight analysis is programmed to evaluate the performance of the Kalman filter. The error between the true trajectory and the estimated and the predicted positions of the airliner are presented for discussion. An improved initialization of the error covariance matrix and that of the state vector are discussed.

Example 2

The second example is air traffic control (ATC) radar. There are two types of ATC radars: airport surveillance radar (ASR) and air route surveillance radar (ARSR). The former is for a short range, the latter for a longer range. An ASR system with a Kalman tracking filter is programmed in the line-of-sight (LOS) coordinates while an airliner approaches an airport.

The transmitter pulsewidth (0.83 μs) and the antenna beamwidth (1.5 degrees), the sources of range measurement error, and the azimuth error are incorporated in the numerical example to demonstrate how a practical Kalman filter really works.

The elements of estimated and predicted error covariance matrices are in terms of LOS coordinates, as are the estimated and predicted airliner's state. A post-flight error analysis is programmed, and the results are presented in graphic form. The advantages and disadvantages of formulating the filter in LOS coordinates system are discussed.

Example 3 and 4

The third and fourth examples are that of a short-range ground-based air defense radar; one operates in the LOS coordinates and the other in the Cartesian coordinates system (CCS). The target is a fighter-bomber on a ground-attack mission with a "turn-dive-attack and turn-climb-escape" maneuver with a maximum acceleration of ± 3g. The random acceleration noise matrix $\mathbf{Q}_k$ and the measurement error matrix $\mathbf{R}_k$ are derived, and the coordinate transformation between LOS and CCS is given. The initialization of the estimate of the error covariance matrix $P_k(k=2)$ instead of $P_k(k=0)$ is discussed in detail.

The tracking performances in the LOS and CCS systems are presented for comparison. The recursive processing sequences are shown in a block diagram.

Example 5

The fifth example emphasizes the problems associated with a matrix inversion, and a Kalman filter without a matrix inversion is presented. We note the special characteristic of the error covariance matrix P_k; it is always symmetric and positive definite. The covariance matrix is factored (decomposed), and we recognize that an inversion is equivalent to a scalar division after factorization.

The Kalman filter without a matrix inversion is programmed. The target is identical to example 3 or 4. The result shows that the difference between "with inversion" and "without inversion" is practically nil. The computation load of no-inversion is, however, slightly higher.

A block diagram of "no-matrix-inversion" is given. The basic processing sequences are not altered; they are two recursive loops, one for the error covariance matrices and the other for state vectors, connected by Kalman gain matrix.

Chapter 9: Monte Carlo Method and Function Integration

In this chapter we learn the basic principles of the Monte Carlo method through examples. We begin with the most simple technique of the "hit-or-miss" method. From this method we learn that the number of sampled data must be on the order of tens of thousands or more to obtain an acceptable level of precision.

The hit-or-miss method is followed by the "ordered sample method" and the "sample mean method" to reduce the number of replications. We present the subject of the coefficient of dispersion (CD) to determine the replication number required for a desired level of precision. The last method we study is the "importance sampling method," a most popular technique among researchers in the field. Mathematics involved in the importance sampling are derived in detail, and two examples are given.

This chapter concludes with several function integration routines found in the elementary calculus books, since the Monte Carlo technique is a numerical experimental mathematic process of integration and a probabilistic determination of a rare event occurrence out of a large number of trials.

Chapter 10: Constant False Alarm Rate (CFAR) Processing

Equations are derived to show the interlocking relationship among the probability of detection, the probability of false alarm, the threshold level, and the number of samples stored in the reference window.

We entertain a scenario of radar in surveillance mode when an antenna sweeps from one sector to another. The antenna will receive different noise or clutter power locally varying. The threshold (or bias level) must be adjusted automatically to maintain the false alarm at a constant level. In this chapter we derive mathematic equations for the threshold when statistically stationary and uniformly distributed noise or clutter is encountered. The crux of the solution is an accurate estimate of the noise or clutter power.

Example 1: CA-CFAR

We present the "cell-average" technique in estimating the noise power in detail. The noise is assumed to be Gaussian-distributed with zero mean and unknown variance. CA-CFAR processing is simple in concept and easy to implement; however, it has a sluggish response to a rapid change in noise level. The CA-CFAR incurs a processing loss inversely proportional to the number of reference cells employed in estimating the noise power. The loss in decibels as a function of the number of reference cells is derived. The most detrimental to a successful operation is the absence of an adequate means to deal with multiple targets in the reference cells, since these spurious targets would raise the threshold erroneously higher.

Example 2: OS-CFAR

We investigate a CFAR technique with a censoring mechanism, called "order-statistics" (OS) processing, or OS-CFAR for short, to handle the multiple-target situation. The received noise data stored in the reference window is rank-ordered (rearranged) by increasing order of magnitude, and we censor (discard) one, two, three, or four of the highest data in estimating the noise power. The underlying principle comes from the theory of order statistics. The distribution of noise after the censor has been known to us. We have applied the theory to the problem.

We have derived the equations of the false alarm probability, the threshold multiplication factor T, and the signal-to-noise (S/N) ratio (in power) per pulse. The results indicate that OS-CFAR has eliminated the problem of multiple targets in the reference window with a small fraction of additional processing loss. We also found that the response to a sudden change in noise level is faster than that of CA-CFAR. The OS-CFAR offers the most robust processing, provided that the noise (or clutter) is Gaussian-distributed, variance-unknown.

CFAR processing in clutter. The reflected power from the land and sea are reported to be more likely distributed as log-normal or Weibull rather than a benign Gaussian. The log-normal distribution is characterized by tall spiky waveforms. The probability density function has the longest tail.

The Weibull is a two-parameter distribution: the shape parameter c and the scale parameter b. Depending on the shape parameter, the Weibull probability density function may be an exponential, or Rayleigh, or a Gaussian look-alike with a small skew to the right. Weibull is a versatile function to characterize the clutter returns that vary spatially or temporally.

We study some statistics of Weibull distribution (i.e., mean, variance, moments, median, mode, and quantiles). Weibull clutters that pass through a square-law detector remain Weibulls, though the shape parameter is halved and the scale parameter is squared. We take advantage of this unique transformation of Weibull in CFAR processing.

Example 3: Weber-Haykin CFAR, WH-CFAR

Weber and Haykin proposed CFAR processing in Weibull clutter. We follow their analysis and present a few numerical results, since they have not provided any. The task is to estimate the two parameters simultaneously from the clutter returns stored in the reference window. The false alarm probability and the threshold are programmed. We present two results with uncensored and censored algorithms. The processing losses are compared with CA-CFAR and OS-CFAR, and we found that WH-CFAR exacts a very high loss in Weibull clutter.

Example 4: Maximum Likelihood CFAR, ML-CFAR

The principle of maximum likelihood estimation is applied to CFAR processing. Mathematic expressions for an estimate of the shape parameter and the scale parameter are given. An estimate of the shape parameter can only be obtained through an iterative procedure since the expression is in a transcendental form. An estimate of scale parameter is obtained by using the iteratively estimated shape parameter.

The shortcoming of the maximum likelihood estimate of the shape parameter is a large biased result when the number of sampled data (the stored clutter samples in the reference window) is relatively small (i.e., 8, 16, or 32).

The biased estimate must be corrected before computing the scale parameter.

The bias is negligible when the samples are larger than 100. (We rarely have a window length of 100.) We have programmed ML-CFAR without a censor and found that the loss is less than that of WH-CFAR; however, it is doubtful that real-time on-line implementation due to the computation burden imposed by the iterative procedure.

Example 5: Minimum Mean Square Error CFAR, MMSE-CFAR

To avoid the laborious iterations in ML-CFAR we apply the minimum mean square error analysis to estimate the shape and scale parameter of Weibull-distributed clutters. The received clutter data stored in the reference window are rank-ordered (rearranged) in ascending order of magnitude. The rank-ordered clutter samples in a natural logarithm is a linear function of the parameters with small deviations from a straight line. A regressive straight line with the minimum mean squared error discloses two parameters; the slope of the straight line is an estimate of the shape parameter, the intercept point an estimate of the scale parameter. The computations involved are simple enough for a real-time implementation.

Chapter 11: Moving Target Indicator (MTI)

A moving target indicator (MTI) is one of perhaps half a dozen indispensable signal processing methods for the successful operation of a surveillance or tracking radar. A MTI is designed on the principle of Doppler frequency detection (the phase differential with respect to time). All moving objects produce Doppler frequency proportional to the relative velocity of the target and the observer.

We present the implementations of recursive and nonrecursive delay line cancelers of various orders and discuss their performance. We mention the origin of blind speed and design approaches to mitigate it by staggering the pulse repetition frequency, f_{PRF}. An example of ATC radar is given for stagger management.

The definition of clutter attenuation (CA) and the improvement factor (I) in decibels is given to evaluate the performance of various MTI filters. The clutter attenuation and the improvement factor would deteriorate when the clutter distribution has a nonzero mean velocity and a broader variance. The system instabilities also cause deterioration. Two examples are given.

Chapter 12: Miscellaneous Program Routines

This chapter has collected two scores of C^{++} programs that do not fit with the chapter title or with the program routines in preliminary preparation for the main program. I hope that readers will benefit from the collections in this chapter.

Contents