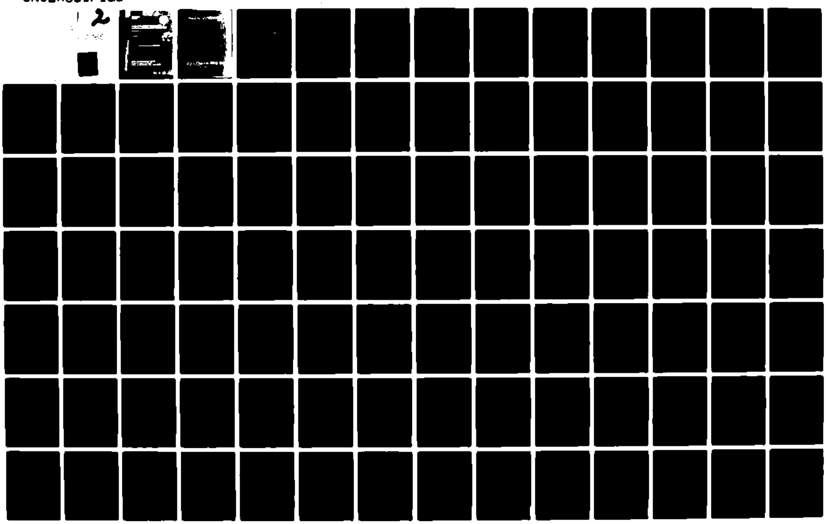
AD-A097 765 MARK RESOURCES INC MARINA DEL REY CA F/G 17/9

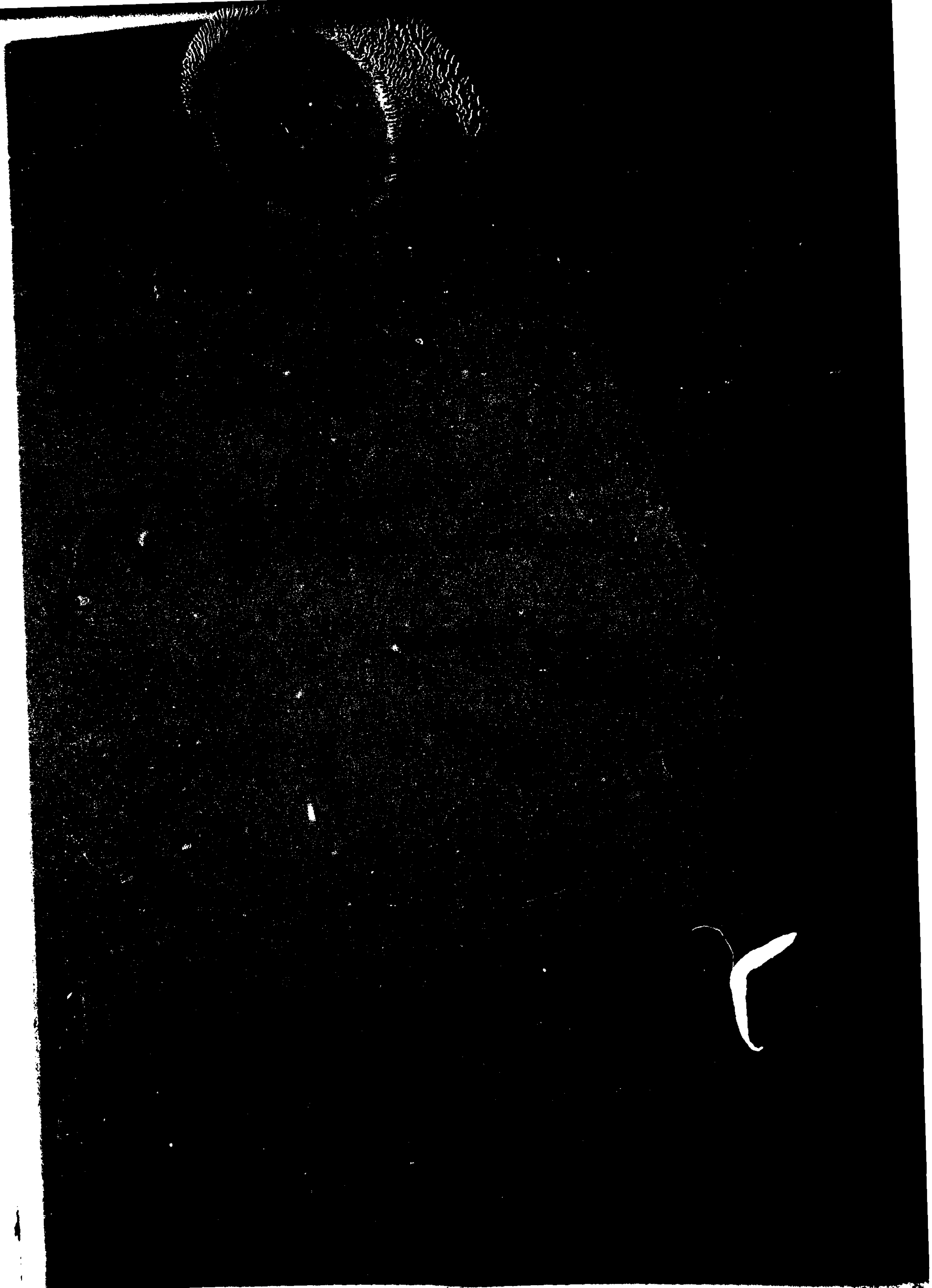
SIMULATING RADAR SIGNALS FOR DETECTION PERFORMANCE EVALUATION.(U)

FEB 81 R L MITCHELL F19628-79-C-0012

UNCLASSIFIED RADC-TR-80-410 NL



AD A097765



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

Field	Entry
1. REPORT NUMBER	RADC-TR-80-410
2. GOVT ACCESSION NO.	AD-A097765
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	SIMULATING RADAR SIGNALS FOR DETECTION PERFORMANCE EVALUATION.
5. TYPE OF REPORT & PERIOD COVERED	Final Technical Report, 1 Oct 78 — 31 Dec 80
6. PERFORMING ORG. REPORT NUMBER	N/A
7. AUTHOR(s)	R. L. Mitchell
8. CONTRACT OR GRANT NUMBER(s)	F19628-79-C-0012
9. PERFORMING ORGANIZATION NAME AND ADDRESS	MARK Resources, Inc. 4676 Admiralty Way, Suite 303 Marina del Rey CA 90291
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	61102F 23051435
11. CONTROLLING OFFICE NAME AND ADDRESS	Deputy for Electronic Technology (RADC/EEC) Hanscom AFB MA 01731
12. REPORT DATE	Feb [redacted] 81
13. NUMBER OF PAGES	55
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	Same
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Capt William Buchanan, USAF

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Detection
Radar Simulation
Importance Sampling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A simulation procedure is developed for determining detection performance of a ground-based radar against airborne targets under a wide variety of environment and system conditions. Quadrative video signals are simulated as they are transmitted, reflected from the scattering environment, and processed in the radar receiver. Importance sampling techniques are used to simulate low false alarm rates.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391766

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
1. INTRODUCTION AND SUMMARY	1
2. IMPORTANCE SAMPLING	3
2.1 CONVENTIONAL SAMPLING THEORY	3
2.2 IMPORTANCE SAMPLING THEORY	7
2.3 SPECIFIC EXAMPLES	9
2.4 CONCLUSIONS	29
3. MODULATION OF CLUTTER WITH A SCANNING ANTENNA	32
3.1 FORMULATION OF THE CLUTTER SIGNAL	33
3.2 FURTHER APPROXIMATIONS	35
3.3 THE GAUSSIAN BEAM	37
4. GENERATING CLUTTER SEQUENCES FOR A GROUND-BASED RADAR	42
4.1 FFT APPROACH	43
4.2 BRUTE-FORCE APPROACH	44
4.3 FFT APPROACH WITH INTERPOLATION	45
5. SIMULATION PROGRAM	49
5.1 INPUT PARAMETERS	49
5.2 INTERPRETING THE RESULTS	53
5.3 EXAMPLES	54
6. REFERENCES	59
APPENDIX A. RADAR SIGNAL SIMULATION PROGRAM TO DETERMINE DETECTION PERFORMANCE (Fortran Listings)	60

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

LIST OF FIGURES

		Page
1.	Sketch of Operations Performed with Conventional Sampling	6
2.	Sketch of Operations Performed with Importance Sampling	8
3.	Application of Importance Sampling to Exponential Distribution ..	12
4.	Application of Importance Sampling to Gaussian Distribution	15
5.	Application of Importance Sampling to Sum of Exponential Variate.	19
6.	Application of Importance Sampling to Sum of Arbitrarily Weighted Exponential Variates	20
7.	Application of Importance Sampling to Processor Consisting of Two Exponential Variates on Input but only One Used on Output ...	22
8.	Application of Importance Sampling to Sum of Log-Normal Variates	23
9.	Application of Importance Sampling to Sum of Log-Normal Phasors .	25
10.	Application of Importance Sampling to Example of Mixed Statistics	27
11.	Application of Importance Sampling to CFAR Example	30
12.	Format of Block Correlated Squares on the Ground	51

EVALUATION

The objective of the study was to develop a method of determining the detection performance of ground-based radars against airborne targets under a wide variety of environment and system conditions.

This effort involved the development of a novel computationally efficient method for simulating ground-based coherent radar system performance subjected to target and clutter signals which are partially correlated.

The shortcoming of conventional simulation techniques used to compute detection performance is that an extremely large number of statistical replications are required to establish false alarm performance. This work advanced the art by adapting the technique of importance sampling which significantly reduces the required computer time, even when simulating nonlinear systems. In most cases it is possible to simulate false alarm probabilities as low as 10^{-8} with only about 10^3 to 10^4 replications. A report, entitled "Importance Sampling Applied to Radar False-Alarms," summarizing results and application examples has been accepted for publication in the IEEE Transactions of the AES.

J Leon Poirier
J. LEON POIRIER
Contract Monitor

1. INTRODUCTION AND SUMMARY

The objective of the study described in this report was to develop a method of determining detection performance for a ground-based radar against an airborne target under a wide variety of environment and system conditions. This objective was met by constructing a digital simulation of the radar signals as they are transmitted, reflected from the scatterers representing targets and clutter, and received by and processed in the radar receiver. Realistic ground, rain, and chaff clutter environments are modeled, as well as the processing features of modern radars such as moving target indication (MTI), both coherent and noncoherent integration, constant false alarm rate (CFAR) processing, and nonlinear operations. The detection performance is determined by Monte Carlo sampling techniques, in which the target and clutter scattering models are described statistically, as well as the location of the target with respect to the center of the antenna beam, range gate, and Doppler filter.

The shortcoming of conventional sampling techniques applied to detection performance in radar is the extremely large number of statistical replications required to establish false alarm performance. In order to overcome this problem a considerable effort was expended in developing importance sampling techniques that could be applied to the wide variety of signals in radar, including the non-Gaussian and non-Rayleigh signals characteristic of clutter. The results of this effort are described in Section 2. In most cases it is possible to simulate false alarm rates as low as 10^{-8} with only about 10^{3} to 10^{4} replications of the experiment. Techniques for handling mixed statistics as well as CFAR are also described.

It is well known that antenna motion during the coherent processing time of the radar causes the clutter to be amplitude modulated; the effect is a broadening of the clutter spectrum that would be observed if there were no antenna motion. It is straightforward to compute the resultant spectrum if the clutter is spatially homogeneous, especially if the antenna pattern and original clutter spectrum are Gaussian shaped. In Section 3 we extend the analysis to nonhomogeneous clutter and arbitrary beamshapes. While the signals are nonstationary, the resulting algorithms are in a form that is amenable to efficient digital simulation.

In Section 4 a general procedure is derived for generating correlated random signal sequences that are characteristic of clutter in a ground-based radar. The procedure is extremely efficient as it is based on fast Fourier transform (FFT) combined with interpolation. It can be applied to any number of signal samples and the shape of the power spectral density is arbitrary.

The simulation program that was developed to determine detection performance is described in Section 5, and a Fortran listing is given in Appendix A.

2. IMPORTANCE SAMPLING

Importance sampling is a technique that can be applied to the simulation of low-probability events without incurring the computation costs usually associated with such simulations. With importance sampling one can modify the probability distribution of the underlying random process in order to make the low-probability events (false alarms) occur more frequently. The desired probabilities at the output of the process are then found by weighting each event by a factor that is a function of only the state of the input; this factor is independent of the process itself [1-6].

The basic principle of importance sampling is straightforward as described above. However, it is not so well known just how the technique can be made to work in a particular application. For example, what if there are multiple random inputs to some processor where the inputs might belong to different statistical processes? Or if the processor is nonlinear? Or if there is not a unique relationship between the input and output of the processor? These and other issues will be addressed in this report. We will concentrate on applications to the simulation of signals in radar and communication systems in order to limit the scope of the study. We begin with a tutorial discussion of both conventional and importance sampling.

2.1 CONVENTIONAL SAMPLING THEORY

Let us designate the output of a statistical process as y. We wish to estimate the probability density function of this process, p(y), or its cumulative distribution function, P(y), with a finite set of observations. The conventional procedure is to sort the output samples into preselected bins that cover the range of interest in the variable y. This operation of sorting results in a histogram, which can be integrated to form the sample distribution function.

For the purpose of examining the upper tail of the distribution function it is more convenient to work with the complement to P(y), namely

$$Q(y) = 1 - P(y) = \int_y^\infty p(\xi)\, d\xi \quad . \tag{1}$$

In estimating Q(y) with a finite set of observations, we can define the operation applied to each sample as

$$D_Y(y) = 1 \quad , \quad y \geq Y$$
$$= 0 \quad , \quad y < Y \quad , \qquad (2)$$

where Y is a preset threshold. In practice, this operation will be applied to many values of Y simultaneously, for each sample of y, in the prior step of computing the histogram. We note that all samples receive the same weight.

The result of applying the operation in (2) to each sample of y is also a statistical process. The mean value of (2) is

$$\overline{D_Y(y)} = \int D_Y(y)p(y)\,dy$$
$$= \int_Y^\infty p(y)\,dy = Q(Y) \quad , \qquad (3)$$

and the second moment is

$$\overline{D_Y^2(y)} = \int D_Y^2(y)p(y)\,dy$$
$$= \int_Y^\infty p(y)\,dy = Q(Y) \quad . \qquad (4)$$

The variance is given by

$$\mathrm{var}[D_Y(y)] = \overline{D_Y^2(y)} - [\overline{D_Y(y)}]^2$$
$$= Q(Y) - Q^2(Y) = Q(Y)P(Y) \quad . \qquad (5)$$

In the following analysis we will be interested in the upper tail of the distribution where $P(Y) \approx 1$, so we can essentially assume that

$$\mathrm{var}\,[D_Y(y)] \approx Q(Y) \quad . \tag{6}$$

In order to estimate $Q(Y)$ the operation in (2) will be repeated for N samples of y and the estimate will be formed as

$$\hat{Q}(Y) = \frac{1}{N} \sum_{i=1}^{N} D_Y(y_i) \quad , \tag{7}$$

where $\{y_i\}$ is the set of N observations (statistical samples), which we assume to be independent. It follows from (3) and (7) that $\overline{\hat{Q}(Y)} = Q(Y)$, which means that (7) is an unbiased estimate. The variance of (7) is given by

$$\mathrm{var}\,[\hat{Q}(Y)] = \frac{1}{N} Q(Y)P(Y)$$

$$\approx \frac{1}{N} Q(y) \quad . \tag{8}$$

In order to estimate $Q(Y)$ with high precision, the standard deviation of the estimate must be small compared with the mean value. In other words, $NQ(Y)>>1$. If, for example, $Q(Y) = 10^{-6}$ then N must be at least as large as 10^6 in order to achieve any precision at all in the estimate. This requirement for an impractically large number of samples is the dilemma faced when one applies conventional sampling techniques to the estimate of low-probability events.

Importance sampling will be a solution to this dilemma, but before we jump to that subject let us expand our discussion to include a description of the process itself. As sketched in Figure 1, the input to some processor will be a random variable x with a known probability density function $p(x)$. The output is y, for which we wish to estimate $Q(y)$, the complement to the distribution function, with a finite set of observations. The estimate will be formed by taking the average value of the observable z, which is the operation $D_Y(y)$ applied to y. The transfer function of the processor will be designated by $y = F(x)$, which implies that a given value of x is mapped into a unique value of y. Since each x results in a particular value of z we can write

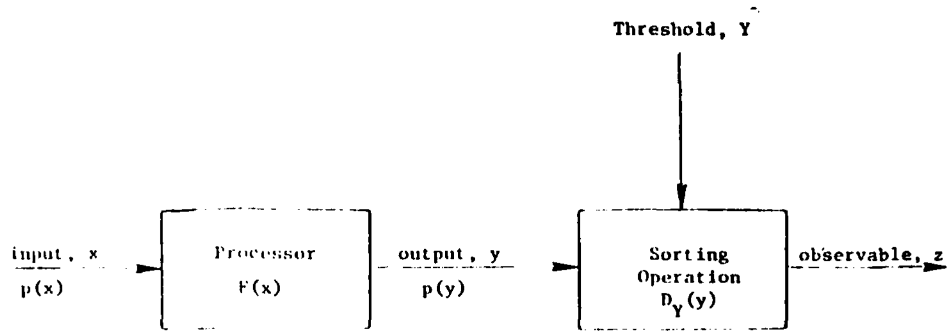


Figure 1. Sketch of Operations Performed with Conventional Sampling

$$z = D_Y[F(x)] \quad . \tag{9}$$

2.2 IMPORTANCE SAMPLING THEORY

The principle of importance sampling is to distort or modify the input random process {x} in order to make the original low-probability events occur more frequently. This action will be compensated by weighting the event by a factor that is a function of only the particular value of x on input [1,2,3]. The functional flow is sketched in Figure 2 where

$$z_m = D_Y[F(x)]w(x) \tag{10}$$

The weight is designated as $w(x)$ and the modified probability density function on input is designated as $p_m(x)$. In order for z_m to be an unbiased estimate of $Q(Y)$ we must have $\bar{z}_m = \bar{z}$, the latter quantity being the mean value of (9), so that

$$\int D_Y[F(x)]w(x)p_m(x)dx = \int D_Y[F(x)]p(x)dx \quad . \tag{11}$$

thus

$$w(x) = p(x)/p_m(x) \quad . \tag{12}$$

For each replication of the experiment a specific value of x will be generated. With the use of (12) we can then compute the weight on the basis of the ratio of known input probability density functions.

So far we have been working with a one-dimensional process. Actually the process could be multi-dimensional. We will assume that there is still a single output, y, so that we can write

$$z = D_Y[F(x_1, \ldots, x_K)] \tag{13}$$

and

$$z_m = D_Y[F(x_1, \ldots, x_K)]w(x_1, \ldots, x_K) \quad . \tag{14}$$

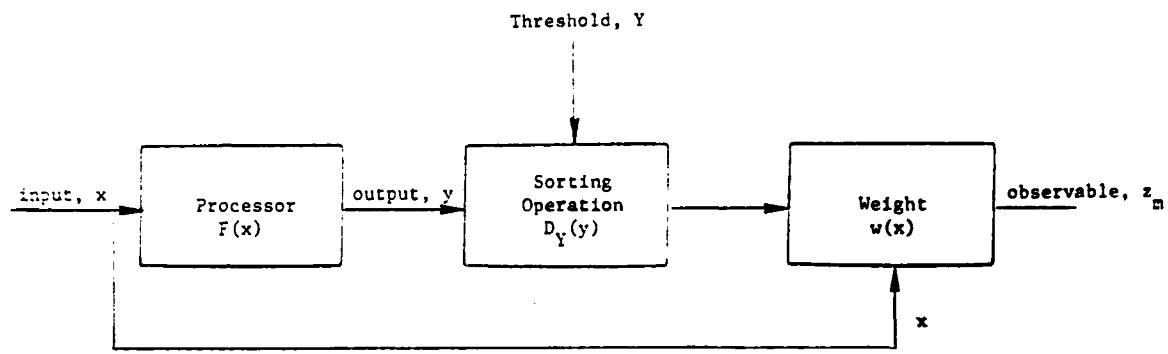


Figure 2. Sketch of Operations Performed with Importance Sampling

And since $\bar{z}_m = \bar{z}$, we have

$$w(x_1, \ldots, x_K) = \frac{p(x_1, \ldots, x_K)}{p_m(x_1, \ldots, x_K)} \qquad (15)$$

where the right side of (15) is the ratio of two joint probability density functions. If the input samples are independent and belong to the same distribution function,

$$w(x_1, \ldots, x_K) = \prod_{k=1}^{K} p(x_k)/p_m(x_k) \quad . \qquad (16)$$

2.3 SPECIFIC EXAMPLES

In order to demonstrate the utility and power of importance sampling, several examples will be given. First, we will examine linear processors where the output statistics are known, and then we will analyze some nonlinear processors.

Example 1: Exponential Distribution

Let x be exponentially distributed with y = F(x) = x. We can define

$$p(x) = (1/\bar{x})e^{-x/\bar{x}} \quad , \quad x \geq 0$$

$$= 0 \quad , \quad x < 0 \quad , \qquad (17)$$

where x is the mean value of the original distribution. Upon integration we obtain

$$Q(x) = e^{-x/\bar{x}} \qquad (18)$$

for $x \geq 0$. For importance sampling we will modify the input distribution by changing (increasing) the mean value as

$$p_m(x) = \frac{1}{\bar{x}_m} \, e^{-x/\bar{x}_m} \quad , \quad x \geq 0$$

$$= 0 \quad , \quad x < 0 \quad . \tag{19}$$

The weight is now given by

$$w(x) = \frac{p(x)}{p_m(x)} = \frac{\bar{x}_m}{\bar{x}} \, e^{-(1/\bar{x} - 1/\bar{x}_m)x} \quad , \tag{20}$$

which is a simple calculation that will be performed for each sample on input.

For the case when $y = F(x) = x$ as we have here, the second moment of z_m can be evaluated easily. For a single observation

$$\overline{z_m^2} = \int D_Y^2(x) w^2(x) p_m(x) \, dx$$

$$= \int_Y^{\infty} w(x) p(x) \, dx \quad . \tag{21}$$

For the exponential distribution we can substitute (17) and (20) to obtain

$$\overline{z_m^2} = \frac{\bar{x}_m}{\bar{x}(2 - \bar{x}/\bar{x}_m)} \, e^{-(2/\bar{x} - 1/\bar{x}_m)Y} \quad . \tag{22}$$

Since $\bar{z} = Q(Y) = e^{-Y/\bar{x}}$, the variance for a single observation is given by

$$\mathrm{var}[z_m] = \left[\frac{\bar{x}_m}{\bar{x}(2 - \bar{x}/\bar{x}_m)} \, e^{Y/\bar{x}_m} - 1 \right] e^{-2Y/\bar{x}} \quad . \tag{23}$$

For N independent observations in estimating the distribution function, the variance will be reduced by a factor of N over that of a single observation. Thus we can write the following ratio for N independent observations:

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = \frac{1}{N}\left[\frac{\bar{x}_m}{\bar{x}(2 - \bar{x}/\bar{x}_m)} e^{Y/\bar{x}_m} - 1\right] \quad . \tag{24}$$

When we work with the expression in (24) it is convenient to simplify it slightly by anticipating the result $\bar{x}_m >> \bar{x}$. Thus

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = \frac{1}{N}\left[\frac{\bar{x}_m}{2\bar{x}} e^{Y/\bar{x}_m} - 1\right] \quad . \tag{25}$$

Now we can ask the question as to what the optimum value of $\bar{x}_m$ is. If we differentiate (25) with respect to $\bar{x}_m$ and set the result equal to zero we obtain $\bar{x}_m = Y$ as the solution. At the optimum,

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = \frac{1}{N}\left[\frac{e}{2}\frac{Y}{\bar{x}} - 1\right] \quad . \tag{26}$$

For illustration, suppose we are interested in estimating Q(Y) in the neighborhood of $Q(Y) = 10^{-6}$. With (18) we obtain $\bar{x}_m = Y = 13.8\bar{x}$ (which means that the approximation made in obtaining (25) was indeed valid) and (26) reduces to

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = 17.8/N \quad . \tag{27}$$

In Section 2 with conventional sampling techniques this ratio was $10^6/N$. We have thus reduced the number of samples required to obtain a specified precision by a factor of over 50,000. With N = 1000 samples the relative standard deviation in the estimate of Q(Y) will be 13.3% with importance sampling, an acceptable error for practically any application.

In Figure 3 we compare the experimental distribution function with N = 1000

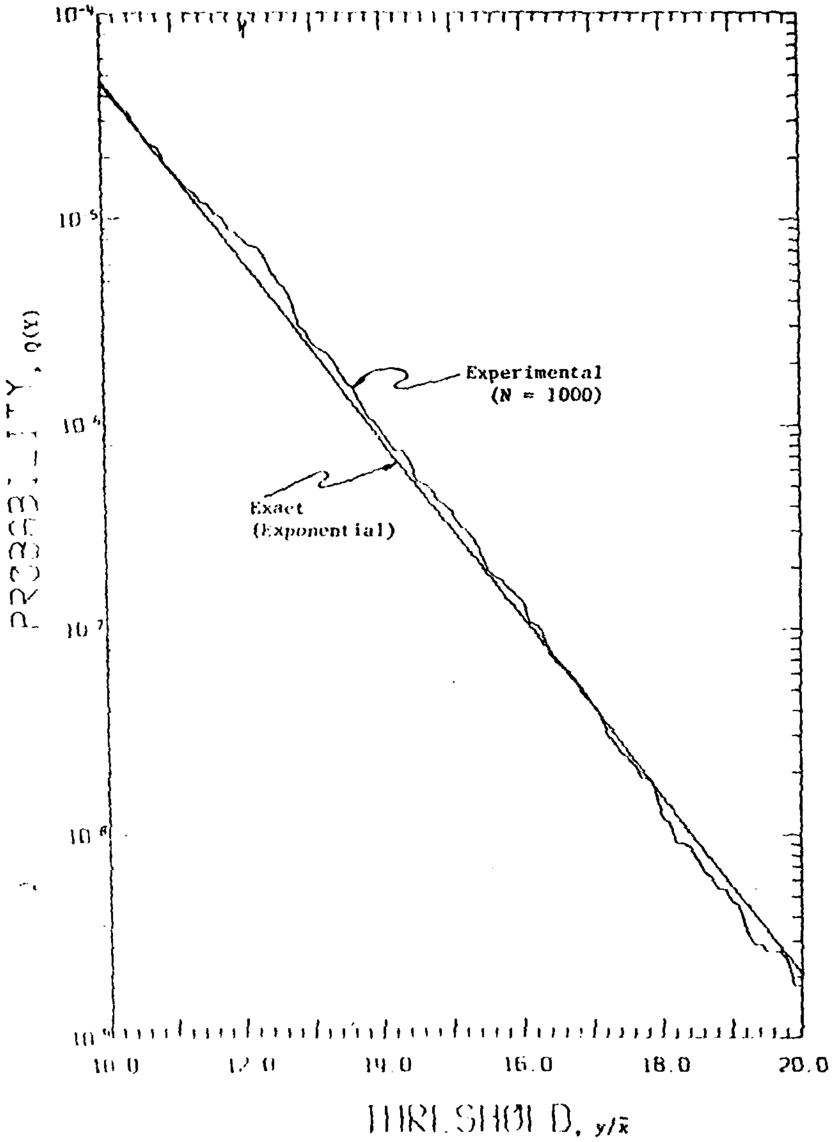


Figure 3. Application of Importance Sampling to Exponential Distribution

and the exact distribution function.* The value of $\bar{x}_m = 13.8\bar{x}$ was used to first compute the histogram, which was then integrated to obtain the sample distribution function. While the procedure was optimized for $Z(Y) \approx 10^{-6}$, we observe that relatively small errors exist throughout the range in probability that covers five orders of magnitude. Thus the procedure is relatively insensitive to the precise value of $\bar{x}_m$ (which could also be established by examining (25)).

Example 2: Gaussian Distribution

Let x be a zero-mean Gaussian random variable, again with $y = F(x) = x$, where

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} , \qquad (28)$$

and

$$Q(Y) = \int_Y^\infty p(x)\, dx . \qquad (29)$$

For importance sampling we will modify $p(x)$ by changing (increasing) σ so that

$$p_m(x) = \frac{1}{\sqrt{2\pi}\,\sigma_m} e^{-x^2/2\sigma_m^2} \qquad (30)$$

The weight is given by

$$w(x) = \frac{p(x)}{p_m(x)} = \frac{\sigma_m}{\sigma} e^{-(1/\sigma^2 - 1/\sigma_m^2)x^2/2} . \qquad (31)$$

*The inputs to the process were exponentially distributed random variables. If u is a uniformly distributed random variable (0,1), then $x = -\ln(u)$ generates an exponential random variable with unit mean. With importance sampling the precise statistical properties of the pseudo-random numbers, u, are no longer critical as they would be with conventional sampling techniques.

From (21) we obtain for a single observation

$$\overline{z_m^2} = \frac{\sigma_m/\sigma}{\sqrt{2 - \sigma^2/\sigma_m^2}}\, Q\left(Y\sqrt{2 - \sigma^2/\sigma_m^2}\right) , \tag{32}$$

where Q(Y) is given by (29). The relative variance for N independent observation is given by

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = \frac{1}{N}\left[\frac{\sigma_m/\sigma}{\sqrt{2 - \sigma^2/\sigma_m^2}}\,\frac{Q\left(Y\sqrt{2 - \sigma^2/\sigma_m^2}\right)}{Q^2(Y)} - 1\right] . \tag{33}$$

In anticipation of the result $\sigma_m >> \sigma$ we might be tempted to simplify (33) by setting $\sqrt{2 - \sigma^2/\sigma_m^2} = \sqrt{2}$; however, the resulting approximation would not yield an optimum solution as a function of σ_m. As it stands, (33) is best handled numerically.

We can find an optimum value of σ_m that minimizes (33) for a specific value of Y. For example, if $Y = 4.7\sigma$, $Q(Y) \approx 10^{-6}$. A value of $\sigma_m = 4.8\sigma$ was found to be the optimum for this value of Y, although (33) was nearly flat over the interval $4.0 \lesssim \sigma_m/\sigma \lesssim 5.5$ so that we can conclude that $\sigma_m = Y$ is essentially the optimum. For $\sigma_m = Y = 4.7\sigma$, the relative variance is

$$\frac{\mathrm{var}[z_m]}{(\bar{z}_m)^2} = 49/N , \tag{34}$$

which means that N = 1000 observations would result in a relative standard deviation of 22% in the estimate of Q(Y). In Figure 4 we show the comparison between the experimental (N = 1000 and $\sigma_m = 4.7\sigma$) and exact distribution functions for the Gaussian case. While σ_m was chosen to optimize the procedure for $Q(Y) \sim 10^{-6}$, we note that the error is relatively small over most of the range in probability plotted in the figure. We see also that the error is somewhat larger than in Figure 3 for the exponential distribution, as we could predict from (27) and (34).

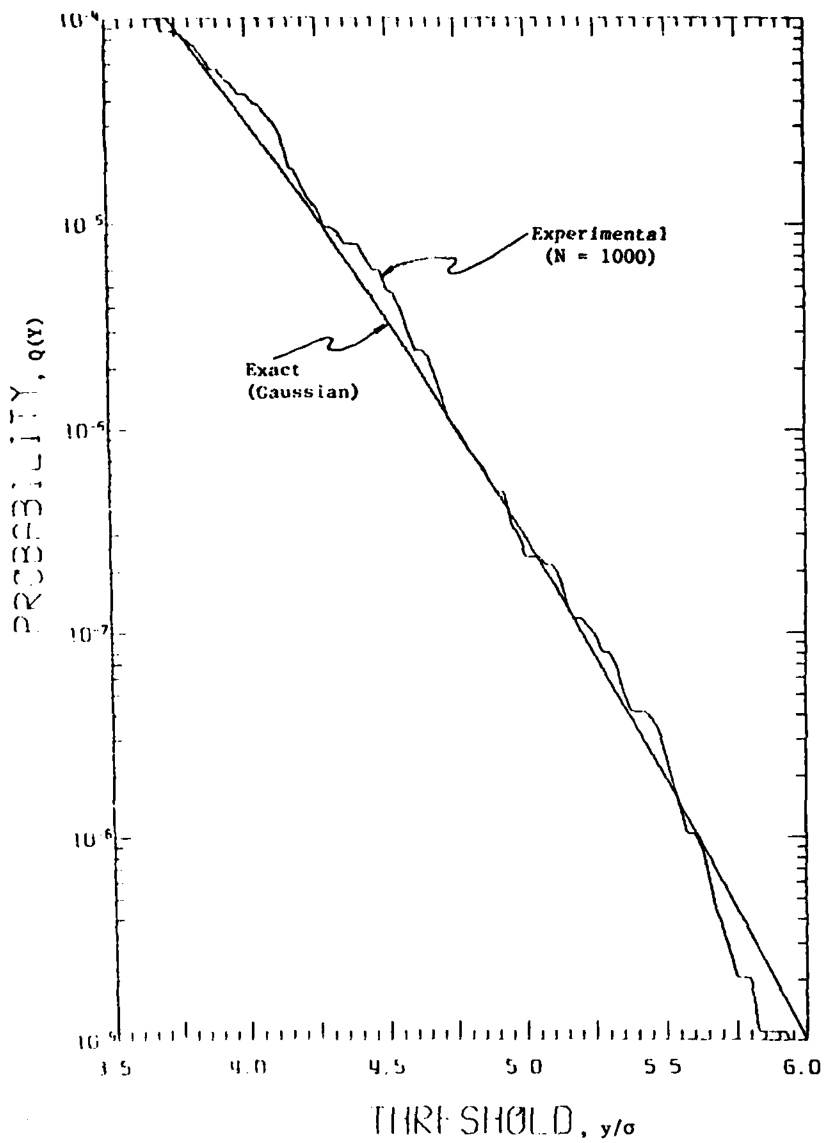


Figure 4. Application of Importance Sampling to Gaussian Distribution

Practically every situation of interest in the analysis of radar and communication systems can be handled by either the exponential or Gaussian random variable as the input to a process (no counter examples are known to the author). For example, the log-normal distribution is a simple transformation of the Gaussian, and the Weibull of an exponential. In fact, one can create an exponential random variable from the Gaussian and vice versa as we will show in the next two examples.

Example 3: Sum-Square of Two Gaussian Variates

Let x_1 and x_2 be two independent, zero-mean Gaussian random variables at the input to our processor, and let

$$y = F(x_1,x_2) = x_1^2 + x_2^2 \quad . \tag{35}$$

If we select $\sigma^2 = 0.5$ then y will be an exponential random variable with $\bar{y} = 1$. With importance sampling the weight applied to each outcome will be a function of only x_1 and x_2. From (16), (28), and (30) we can write

$$w(x_1,x_2) = 2\sigma_m^2 \, e^{-(1 - 1/2\sigma_m^2)(x_1^2 + x_2^2)} \quad , \tag{36}$$

and with (35)

$$w(x_1,x_2) = 2\sigma_m^2 \, e^{-(1 - 1/2\sigma_m^2)y} \quad . \tag{37}$$

We have applied the above procedure for generating the exponential random variable and with importance sampling we have computed the experimental distribution function. We chose $\sigma_m = 2.63$ (or $2\sigma_m^2 = 13.8$) and N = 1000. The experimental result is essentially identical to that in Figure 3, at least in the statistical sense.

Example 4: Generation of Gaussian Variate from Exponential

Let x be an exponential random variable at the input to our processor. Within the processor we will create

$$y = F(x) = \sqrt{2x} \cos \theta \quad , \tag{38}$$

where θ is a uniformly distributed random variable $(0,2\pi)$.* The random variable y will be Gaussian distributed, and if $\bar{x} = 1$ the mean value of y will be zero and the variance unity. With importance sampling the weight applied to each outcome will be a function of only x in this example, since the generation of θ is performed within the processor and will not be considered as an input variable. The weight is given by (20) for $\bar{x} = 1$.

We have generated Gaussian random samples by means of (38) and have used importance sampling to estimate the distribution function in the range of $10^{-4} \leq Q(Y) \leq 10^{-9}$. The result is statistically consistent with that of the direct method in Figure 4. A value of $\bar{x}_m = 22.1$ $(= 4.7^2)$ corresponds to optimizing the procedure at $Q(Y) = 10^{-6}$.

With this example we no longer have a unique mapping of x into y, which ought to suggest other possibilities in simulation. In addition, we can configure the processors so that all inputs are either exponential or Gaussian random variables as we will show in Example 9.

Example 5: Sum of Exponential Variates

The sum of K independent random variables that are exponentially distributed is a chi-square random variable with 2K degrees of freedom. We can write

$$y = F(x_1, \ldots, x_K) = \sum_{k=1}^{K} x_k \quad , \tag{39}$$

where $\{x_k\}$ is the set of K exponential random variables on input. With importance sampling the weight will be a function of the specific values of x_k. From (16), (17), and (19) we can write

$$\begin{aligned} w(x_1, \ldots, x_K) &= (\bar{x}_m/\bar{x})^K \exp\left[-(1/\bar{x} - 1/\bar{x}_m) \sum_{k=1}^{K} x_k\right] \\ &= (\bar{x}_m/\bar{x})^K e^{-(1/\bar{x} - 1/\bar{x}_m)\, y} \end{aligned} \tag{40}$$

*The use of (38) with $\sin\theta$ substituted for $\cos\theta$ will produce a second, independent Gaussian random variable.

In Figure 5 we show the experimental distribution obtained by using importance sampling on (39) for $K = 5$ and $N = 1000$ replications. The mean value of the input distribution was modified as $\bar{x}_m = 4.7\bar{x}$, which optimized the procedure at $Q(Y) = 10^{-6}$. Note that the error is extremely small throughout the five orders of magnitude in probability shown in Figure 5.

Example 6: Sum of Exponential Variates with Arbitrary Weights

For the previous example all input random variables were uniformly weighted in (39). Let us generalize the situation by assuming arbitrary weights in the summation as

$$y = F(x_1, \ldots, x_K) = \sum_{k=1}^{K} a_k x_k \quad , \tag{41}$$

where $\{x_k\}$ is the set of independent exponential random variables on input with a mean value $\bar{x}$. With importance sampling the weight is a function of only the input, so the weight remains the same as (40). The distribution function of y is given by

$$Q(y) = \sum_{k=1}^{K} \frac{a_k^{K-1}}{\prod_{i\neq k}(a_k - a_i)} e^{-y/a_k\bar{x}} \quad . \tag{42}$$

In Figure 6 we have used importance sampling to estimate $Q(Y)$ for $K=2$ and $N = 1000$. The ratio of weights used in (41) varied from .48/.52 (resulting in essentially a chi-square distribution with 4 degrees of freedom) to .05/.95, with the sum of weights being unity in all cases. For each of the cases, the procedure was optimized in the neighborhood of $Q(Y) = 10^{-6}$ (e.g., $\bar{x}_m = 10.7\bar{x}$ for the .25/.75 case).

With this example we can create the following interesting situation: let $K=2$ so that the weight used in importance sampling in (40) will be based on two input variables; however, let one of the weights a_k in (41) be zero so that the output y will be exponentially distributed. From one replication to the other we can even interchange the a_k without affecting the distribution function of y. Does importance sampling work in this case? Indeed it does,

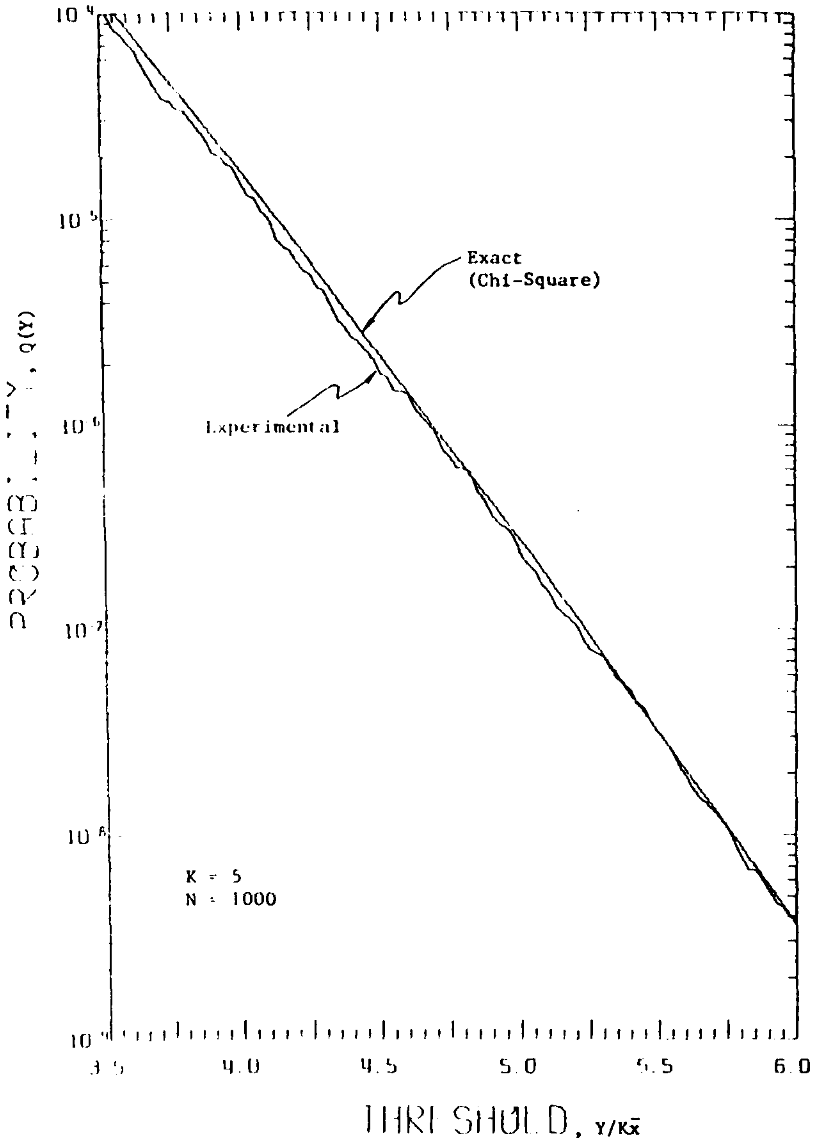


Figure 5. Application of Importance Sampling to Sum of Exponential Variates

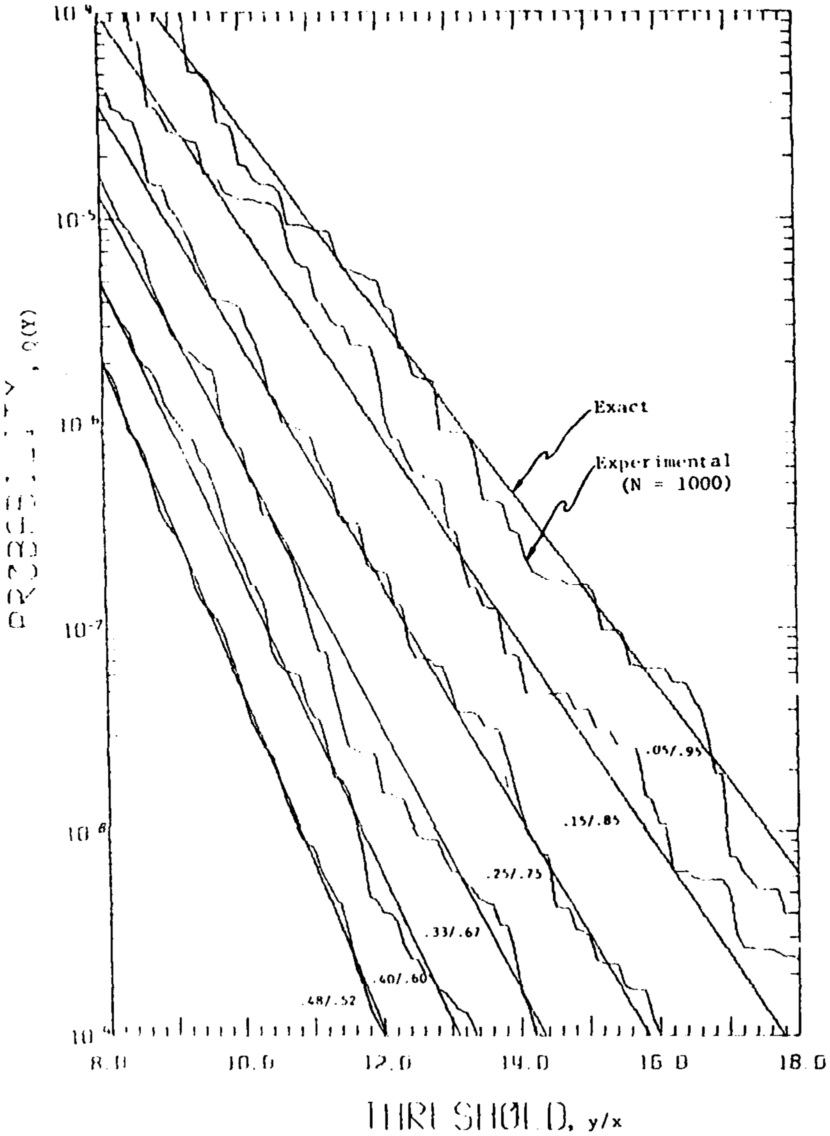


Figure 6. Application of Importance Sampling to Sum of Arbitrarily Weighted Exponential Variates (K=2; ratio of weights is indicated)

as we show in Figure 7, with N = 1000 and $\bar{x}_n$ = 13.8. The error is not as small as it was in Figure 3, but this could be compensated by increasing N.

Example 7: Sum of Log-Normal Variates

The distribution of the sum of log-normal random variables is difficult to compute by conventional numerical methods. Let us begin with an exponentially distributed random variable x_k as one of K inputs to the processor in Figure 1, with $\bar{x} = 1$. We will then generate a Gaussian random variable of unit variance as in Example 4,

$$g_k = \sqrt{2x_k}\cos\theta \quad , \tag{43}$$

where θ is a uniformly distributed random variable $(0,2\pi)$. The log-normal random variable is generated as

$$\ell_k = e^{\sigma_L g_k} \quad , \tag{44}$$

where σ_L is the standard deviation of the log-variate. The median value of ℓ_k will be unity. The output of the processor will then be

$$y = \sum_{k=1}^{K} \ell_k \quad , \tag{45}$$

for which we will use importance sampling to estimate its distribution function. The weight will be given by (40).

Because of the complicated processor above, there is no straightforward way to choose $\bar{x}_m$ which modifies the input distribution for importance sampling. Therefore, we will try some arbitrary values. In Figure 8 we show the experimental distribution functions for eight cases where we have varied x_m from 2 through 30 in steps of 4. In each case K = 2, σ_L = 1.0, and N = 1000 observations. We see that all but one of the cases tend to be clustered about a straight line in the format plotted. The exception is the $\bar{x}_m$ = 2 case, which

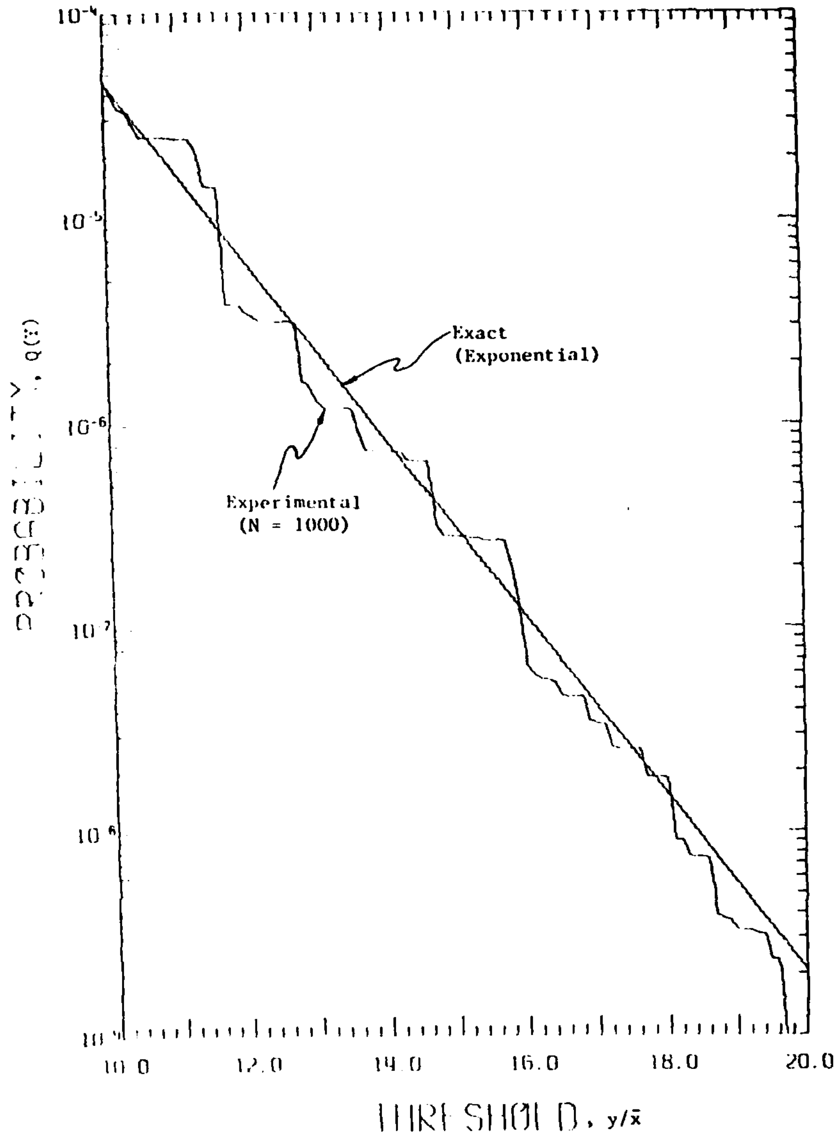


Figure 7. Application of Importance Sampling to Processor Consisting of Two Exponential Variates on Input but only One Used on Output

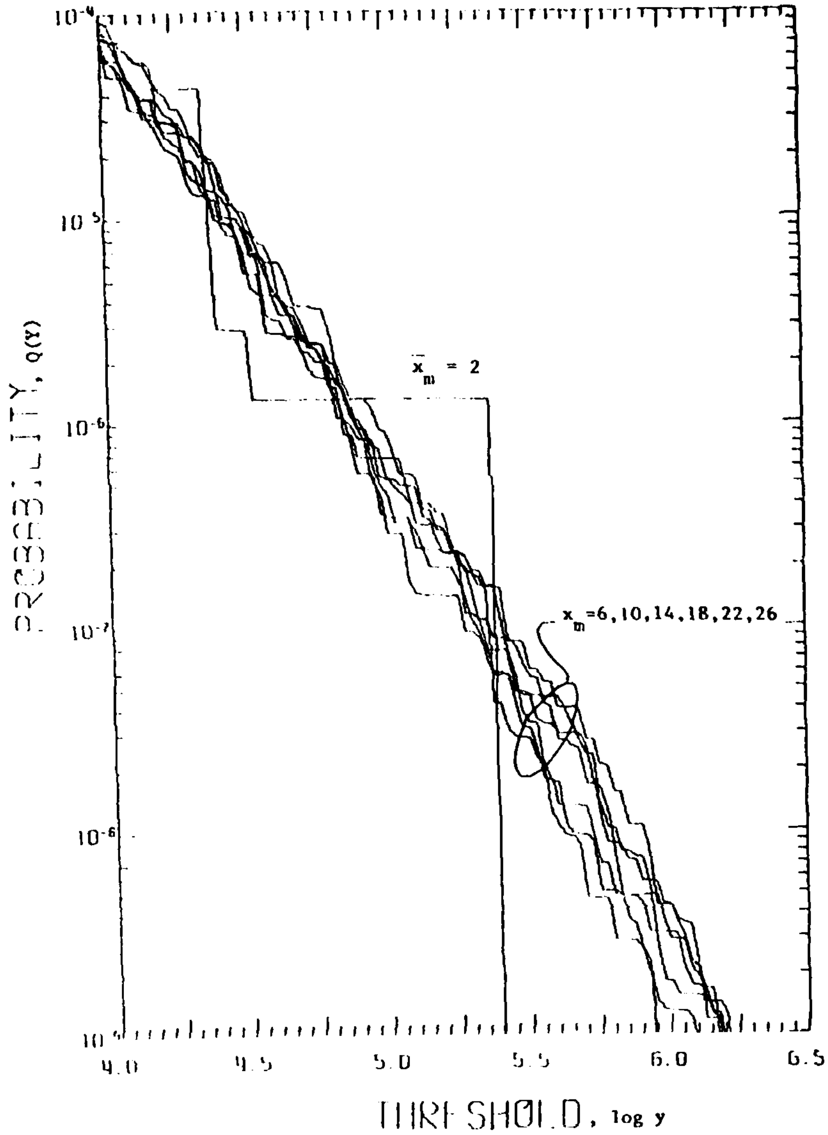


Figure 8. Application of Importance Sampling to Sum of Log Normal Variates ($K = 2$, $\sigma_1 = 1.0$, $N = 1000$)

is too small to produce values of y frequently enough in the range of interest. The remaining cases establish the validity of importance sampling for this example since each estimate is unbiased; the spread of the cluster is a measure of the standard deviation of the estimate. By inference, $\bar{x}_m = 1$ (which is conventional sampling) with a sufficient number of observations would also produce a result that would fall within the cluter in Figure 8. Almost any value of $\bar{x}_m$, especially in the interval $6 \leq \bar{x}_m \leq 30$, could be used with importance sampling to produce an acceptable result. However, since $\bar{x}_m \approx 22$ optimizes the procedure for a single Gaussian random variable at $Q(Y) = 10^{-6}$, there is probably no requirement to exceed this value in the general case. We could also repeat the above procedure whenever K or σ_L were changed.

We have also used importance sampling on (45) by beginning with independent Gaussian random variables as the input to the processor. The above conclusions were unaffected by this change, however (the range $2 \leq \sigma_m \leq 8$ proved satisfactory).

Example 8: Sum of Log-Normal Phasors

A situation that is common in the analysis of radar clutter is the summation of random phasors where the amplitude of each is log-normally distributed. We will generate the log-normal random amplitude ℓ_k as in the previous section (but σ_L now refers to the signal amplitude) and then generate the random phasor as

$$v_k = \ell_k e^{j\phi_k} \quad , \tag{46}$$

where ϕ_k is a uniformly distributed random variable $(0, 2\pi)$. Finally, we form

$$y = \left| \sum_{k=1}^{K} v_k \right|^2 \quad , \tag{47}$$

for which we will use importance sampling to estimate its distribution function.

In Figure 9 we show the experimental distribution functions for three cases, $x_m = 18$, 22, and 26, where $K=2$, $\sigma_L = 0.5$, and $N = 1000$ for all cases. The experimental curves again cluster fairly closely throughout the entire range in probability plotted.

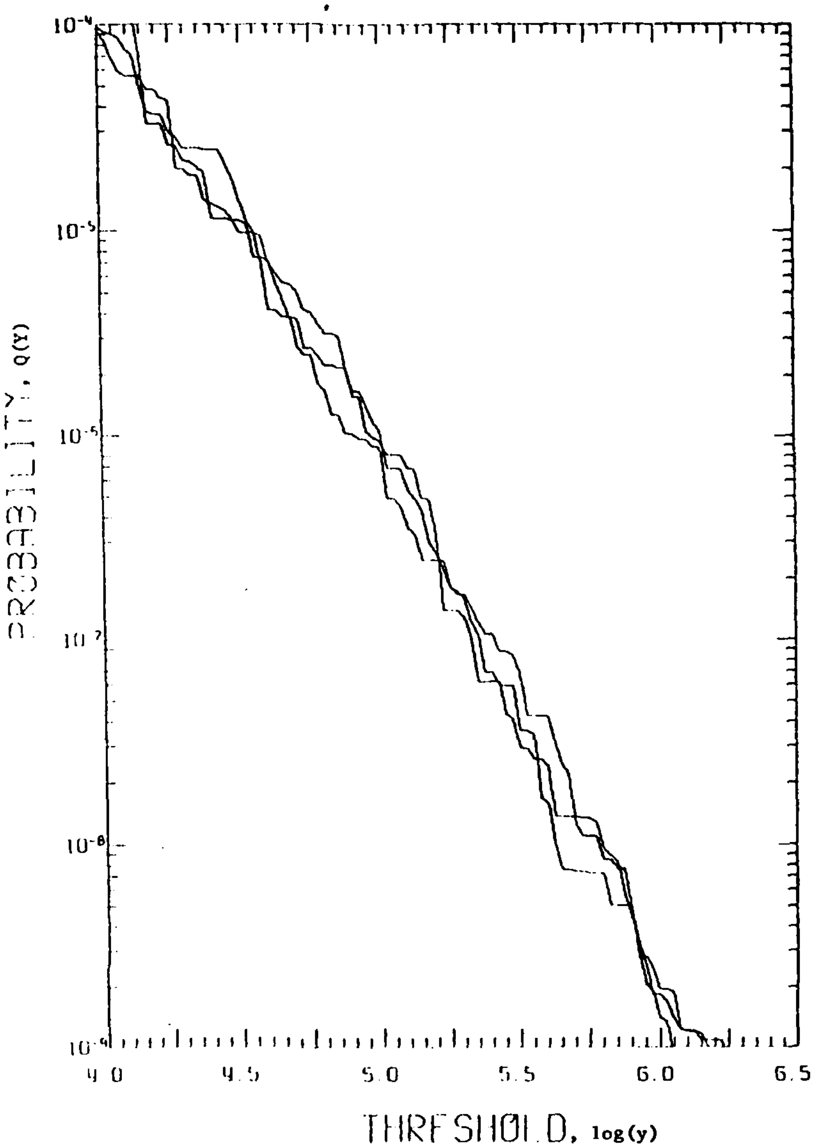


Figure 9. Application of Importance Sampling to Sum of Log-Normal Phasors ($K = 2$, $\sigma_L = 0.5$, $N = 1000$)

Example 9: Mixed Statistics

In Examples 5 through 8 we have superimposed random samples from the same statistical distribution. But in radar and communication systems, signals are often combined with mixed statistics. For example, radar clutter, which is often assumed to be log-normally distributed, would be combined with thermal noise, which is Rayleigh-amplitude distributed. In this example we will simulate this case and use importance sampling to estimate the distribution function. We will again begin with two exponentially distributed random variables as the input to the processor in Figure 1. A log-normal phasor, V_1, will be generated from one of the exponentially distributed inputs, x_1, by means of (43), (44), and (46). The Rayleigh-amplitude phasor will be generated from the second exponentially distributed input, x_2, as

$$V_2 = \sqrt{x_2}\,(\cos\phi + j\sin\theta) \quad , \tag{48}$$

where θ is a uniformly distributed random variable $(0, 2\pi)$ that is generated within the processor. The output of the processor will be

$$y = |V_1 + aV_2|^2 \quad , \tag{49}$$

where a is a factor used to scale one process with the other. We will constrain $\bar{x} = 1$ for both inputs.

In Figure 10 we show the application of importance sampling used to estimate the distribution function of y for a = 3 and $N = 10^4$ observations. The four values of $\bar{x}_m$ = 14, 18, 22, and 26 were used to generate the four experimental curves. Note that they are tightly clustered, especially for the higher probabilities shown in the figure. We have also shown the exact distribution functions for the components that make up the sum in (49).

Since the four experimental curves in Figure 10 are so closely clustered throughout five orders of magnitude in probability, even with such widely different values of the importance sampling scaling parameter, $\bar{x}_m$, we can conclude with high probability that the true distribution also lies within the cluster. Thus we can also conclude that importance sampling works when the process

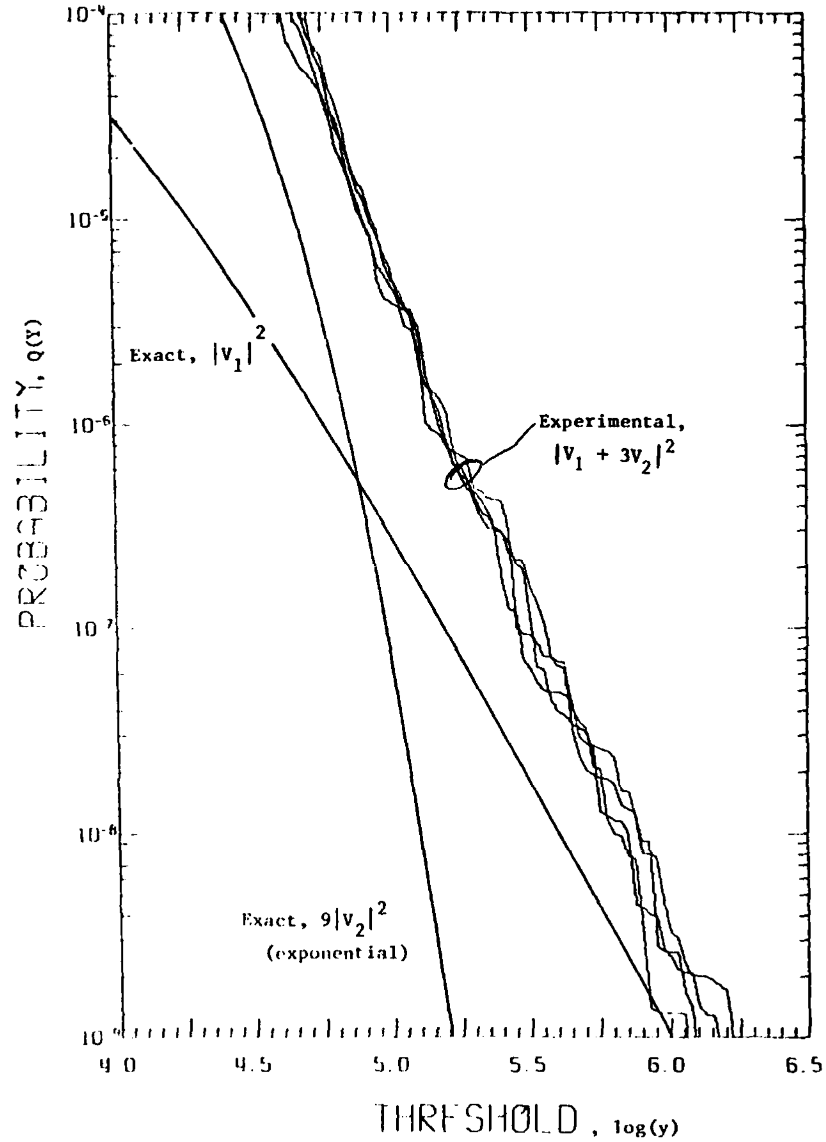


Figure 10. Application of Importance Sampling to Example of Mixed Statistics ($N = 10^4$)

under investigation involves a combination of random variables with different statistics. In the application of importance sampling one probably should create all random variables by some transformation of a set of input random variables with a common distribution.

Example 10: CFAR

A simple approach in radar to achieving a constant false alarm rate (CFAR) in the presence of nonstationary noise is to set the detection threshold on the basis of the average noise power in a number of reference samples where each of these samples is assumed to represent noise only. Such a scheme is denoted as cell averaging CFAR. The output of the processor might be

$$y = \frac{x}{\sum_{i=1}^{R} x_i} \tag{50}$$

where x is the sample under test and the set of samples $\{x_i\}$ form the CFAR reference. Note that if we modify all R+1 input variables equally by changing the common mean value, the output y remains unchanged and importance sampling would seem to be inapplicable in this case. One could apply importance sampling to y if p(y) were known; however, this would be a severe limit to the approach.

In order to find a method that works on the input variables it was decided to treat the numerator in (50) as the input process; the denominator will be generated internally to the processor, just as we generated θ in Example 4. The reasoning behind this choice is that the variance of the denominator is much less than the numerator, and the low probability behavior of the numerator has a much greater impact on the outcome than does the denominator. However, unless R is reasonably large we will not get much of an improvement with importance sampling compared to the conventional approach. If the random variables are exponentially distributed, the following number of observations N will be needed for each value of R to provide the same quality of result with importance sampling $[Q(y) = 10^{-6}]$:

R	∞	200	100	50	20
N	1000	2300	4700	15,000	170,000

In most applications of cell averaging CFAR in radar we will also employ some noncoherent integration of samples prior to forming the CFAR ratio. In such cases we can write the output as

$$y = \frac{\sum_{k=1}^{K} x_k}{\sum_{i=1}^{R} x_i} \tag{51}$$

Usually there will be M reference samples for each sample in the numerator, so we can write R = KM. In a typical situation we might have K = 6 and M = 10 so that R = 60, which permits us to use a reasonably small number of replications to determine Q(Y) with importance sampling applied to only the numerator of (51). In Figure 11 we show the results of this case with N = 10,000 observations; all random variables are exponentially distributed and $\overline{x}_m = 5\overline{x}$ for the K = 6 input variables. Thus importance sampling also works with cell averaging CFAR, but we have been forced to redefine the input to the processor.

2.4 CONCLUSIONS

Importance sampling has wide application in the simulation of signals in radar and communication systems. It is robust and efficient, producing reliable estimates of the low-probability tail of the distribution function with typically 1000 replications of the experiment. It has been shown to work with multiple input processes when all inputs belong to the same distribution and are distorted equally. For a process involving a combination of several types of signals, each from a different statistical family, one can still apply importance sampling by redefining the procedure in which the signals are generated. In effect, a new processor is created so that all input signals will

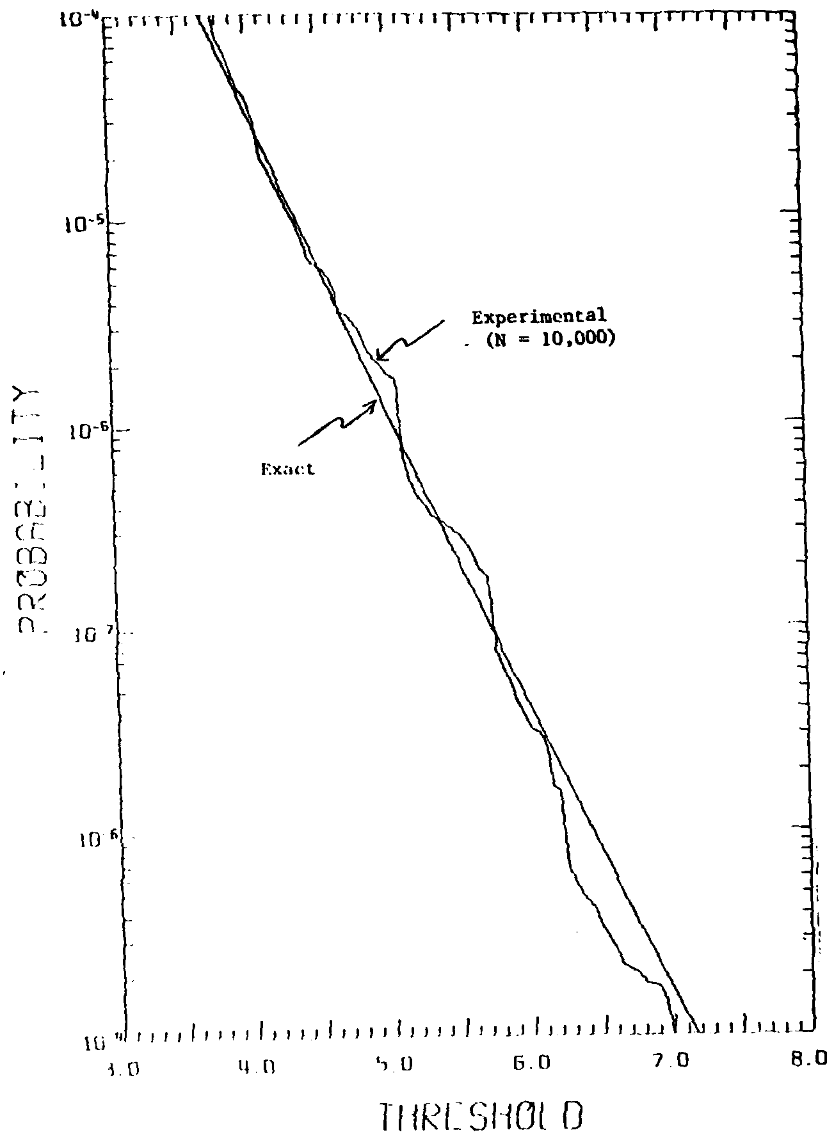


Figure 11. Application of Importance Sampling to CFAR Example

belong to the same statistical family and all will be distorted equally in importance sampling. In some cases, such as with cell averaging CFAR, we will even have to redefine the input to the processor. It has been shown that one can begin with either Gaussian or exponentially distributed signals and generate practically any combination of statistical signals used in the simulation of radar and communication systems. We are unaware of any situation of practical interest where this procedure would fail.

3. MODULATION OF CLUTTER WITH A SCANNING ANTENNA

There are two basic ways to scan the antenna in a ground based search radar: either continuously or step-wise. In the latter case we can assume that there is no antenna motion relative to the ground for a fixed time which we designate as the on-target time. Received pulses from the ground clutter will be partially correlated which is due entirely to the internal motion of the clutter itself (the wind-induced motion of trees, etc.). If the antenna moves from pulse to pulse, then the return fron any point on the ground will be amplitude modulated as a result of the time-varying antenna gain in the direction of that point. The correlation properties of ground clutter will then be affected by the scanning modulation in addition to the internal motion of the clutter. In many cases, the scan modulation effect dominates the internal motion effect, and for simulating clutter signals the scan modulation severely complicates an otherwise straightforward implementation.

For the purposes of this discussion we will distinguish between two types of radar signal processing: continuous and batch. With continuous processing the radar generally utilizes all past data in a recursive filter to process each pulse, and there is usually one output signal for each pulse. Continuous processing is often implemented in continuously scanning systems, at least for older radars. With batch processing the radar collects a sequence of pulses before any processing is implemented, and the processing from one batch to the next is independent. Batch processing is implemented on all step scan systems (as known to the author) and some of the newer continuous scan systems, especially the 3-D search radars utilizing an electronic scan in elevation. The FFT processor is one example of a batch processor.

The distinction between continuous and batch processing has a dramatic effect on the techniques used to simulated clutter signals, so much so that the techniques and computer algorithms used for one system will bear little similarity to those used for the other. In effect, there will be two distinctly different computer programs required to simulate both cases. The reason is that the continuous processing system demands almost a pulse-by-pulse approach to computing the interaction of the antenna with the clutter geometry, while the batch processing system usually needs to be updated only once for each batch of pulses. The latter case is simpler and cheaper to simulate, and few questionable approximations will be required out of necessity. For this reason we will concentrate our discussion on the continuous processing system. We will assume in the following analysis that the radar has 2-D resolution and the antenna has a fan beam in elevation.

3.1 FORMULATION OF THE CLUTTER SIGNAL

Let us designate the complex signal associated with the k^{th} scatterer in a range resolution cell as $V_k(t)$, which we have made time-varying in order to account for internal motion of the clutter. In general, $|V_k(t)|^2$ will have the dimensions of radar cross section (RCS), or some power scaled version of it according to the radar range equation. In addition, let $g(\theta)$ be the one-way voltage gain of the antenna in the azimuth, θ, direction. The composite signal (complex voltage) received by the radar when the antenna scans continuously in azimuth will be proportional to

$$V(t) = \sum_k V_k(t) g^2(\theta_o + \dot{\theta} t - \theta_k) \tag{52}$$

where θ_o is some reference angle, $\dot{\theta}$ is the scan rate, θ_k is the azimuth of the k^{th} scatterer, and the summation is performed over all scatterers within a range resolution cell. Differences in range among scatterers that would effect the phase of superposition in (52) are implicitly included in the definition of $V_k(t)$.

As written (52) is nontrivial to implement because $V_k(t)$ is a two dimensional process: *spatial and temporal*. Without any simplification (which can be done only through approximations) we must evaluate (52) for each pulse, and at the very least we must have one scatterer (in a range ring) for each pulse throughout a 360° scan. For a PRF of 400 Hz and a 10 sec scan we will have 4000 total pulses and at least 4000 scatterers. The process $V_k(t)$ is thus described by a

4000 × 4000 matrix (neglecting the "edge effects" at the beginning and end of the scan). Clearly, some kind of approximation is in order at this point to reduce the size of the problem.

One approach would be to assume that $g(\theta)$ is negligibly small for $|\theta|$ larger than some value (where $\theta = 0$ corresponds to the mainlobe axis.) We could simulate only the mainlobe or the near sidelobes of the antenna, and assume that there is no contribution from sidelobes further out than some point. The problem is that in some cases (in older systems) just simulating the mainlobe might be adequate but in other cases (in newer, high-performance systems) we might have to include many sidelobes. Just where we should stop simulating sidelobes is not readily apparent since it depends on the system being simulated. Few ground rules can be given either because very little analysis has been performed on this subject.

Another approach to simplifying (52) is to focus our attention on $V_k(t)$. In general, this process can be assumed to be random (for the objective of determining detection performance), with the statistics of one scatterer being independent from any other. The random process will be spatially uncorrelated. It is fairly common in the literature to break up the scattering properties of ground clutter into two components: an ac component and a dc component. The ac component, the time-varying one, is the result of the motion of trees, brush, grass, etc., and is generally made up of many individual scatterers (e.g., the leaves on a tree) where no single one dominates. Such a process is easily justified to be Gaussian, at least within a relatively small area on the ground. On the other hand, the dc component is associated with rigid objects, such as bare ground, tree trunks, man-made structures, etc., and non-Gaussian statistics usually prevail in this case. In general, the dc component is stronger (higher power) than the ac component.

With the above distinction between the two components let us define

$$V_k(t) = V_k + \tilde{V}_k(t) \tag{53}$$

where V_k is the dc component of the k^{th} scatterer and $\tilde{V}_k(t)$ is the ac component. If we substitute (2) into (1) we obtain

$$V(t) = \sum_k V_k g^2(\theta_o+\dot{\theta}t-\theta_k) + \sum_k \tilde{V}_k(t) g^2(\theta_o+\dot{\theta}t-\theta_k) \tag{54}$$

At first glance it appears that we have complicated the situation because we now have two summations insted of one. However, the first one is straightforward; it is a circular convolution, something that is easily implemented. For the second summation, which has all the inherent complexity of (52) we will truncate the antenna pattern as described previously, but now the effect of the truncation will be much less significant since $\tilde{V}_k(t)$ will usually be much weaker than $V_k(t)$.

Equation (54), or at least the second term of it, is not particularly economical to implement in terms of computer time, especially when computing false-alarm performance. For this reason we will pursue further approximations.

3.2 FURTHER APPROXIMATIONS

In general (52) and (54) describe a nonstationary process because clutter samples in one azimuth sector can be much stronger than in another. This is another way of saying that clutter is spatially nonhomogeneous. If this property of clutter applies to both terms in (54) then there is little more we can do to simplify the expression. However, if the ac component of clutter is homogeneous (mere speculation at this time) we can treat the second term in (54) as a stationary process. With this assumption we can apply Fourier analysis to determine the spectrum of this term which we rewrite as

$$V_{ac}(t) = \sum_k \tilde{V}_k(t) g^2(\theta_o+\dot{\theta}t-\theta_k) \tag{55}$$

In determining the second moment, let us write

$$V_{ac}(t_1)V_{ac}^*(t_2) = \sum_k \tilde{V}_k(t_1)\tilde{V}_\ell^*(t_2) g^2(\theta_o+\dot{\theta}t-\theta_k) g^2(\theta_o+\dot{\theta}t-\theta_\ell) \tag{56}$$

When we take the ensemble average of (56) the cross terms vanish as

$$\overline{\tilde{V}_k(t_1)\tilde{V}_\ell^*(t_2)} = 0\ ,\ k \neq \ell$$

so that

$$\underbrace{\overline{V_{ac}(t_1)V^*_{ac}(t_2)}}_{\substack{\text{autocorrelation function}\\ \text{of composite process}}} = \sum_k \underbrace{\overline{\tilde{V}_k(t_1)\tilde{V}^*_k(t_2)}}_{\substack{\text{autocorrelation function of}\\ \text{clutter internal motion}}} g^2(\theta_o+\dot{\theta}t_1-\theta_k)g^2(\theta_o+\dot{\theta}t_2-\theta_k) \tag{57}$$

The autocorrelation functions above will be assumed to be nonstationary (even though we will later assume that the autocorrelation function of the clutter internal motion is stationary). Thus we will define

$$\tilde{R}(t_1,t_2) = \overline{\tilde{V}_k(t_1)\tilde{V}^*_k(t_2)} \tag{58}$$

$$R_{ac}(t_1,t_2) = \overline{V_{ac}(t_1)V^*_{ac}(t_2)} \tag{59}$$

and

$$h_k(t) = g^2(\theta_o + \dot{\theta}t - \theta_k) \tag{60}$$

Note that we are assuming that (58) is independent of the subscript k so that the clutter process is spatially homogeneous. With this notation, (57) becomes

$$R_{ac}(t_1,t_2) = \sum_k \tilde{R}(t_1,t_2)\, h_k\,(t_1)\, h_k\,(t_2) \quad . \tag{61}$$

At this point we will introduce the two-dimensional Fourier transform as used in Reference 7, so that

$$\tilde{\Gamma}(f_1,f_2) = \iint \tilde{R}(t_1,t_2)\, e^{-j2\pi(f_1t_1 - f_2t_2)}\, dt_1dt_2 \quad , \tag{62}$$

and

$$\tilde{R}(t_1,t_2) = \iint \Gamma(f_1,f_2)\, e^{j2\pi(f_1t_1 - f_2t_2)}\, df_1 df_2 \quad , \tag{63}$$

with a similar set of expressions for $\Gamma_{ac}(f_1,f_2) \leftrightarrow R_{ac}(t_1,t_2)$. The two-dimensional Fourier transform of (61) is now

$$\Gamma_{ac}(f_1,f_2) = \sum_k \iint \tilde{R}(t_1,t_2)\, h_k\,(t_1)\, h_k\,(t_2)\, e^{-j2\pi(f_1t_1-f_2t_2)}\, dt_1dt_2 \quad . \tag{64}$$

Let $t_2 = t_1 - \tau$ and assume that $\tilde{R}(t_1,t_2) = \tilde{R}(\tau)$, which is the definition of a stationary process. After some manipulation we can write

$$\Gamma_{ac}(f_1,f_2) = \sum_k \int \tilde{R}(\tau)\, e^{-j2\pi f_2\tau} \left\{ \int h_k(t_1)h_k(t_1-\tau)e^{-j2\pi t_1(f_1-f_2)}\, dt_1 \right\} d\tau \quad . \tag{65}$$

The quantity within the braces is the ambiguity function of the waveform $h_k(t)$. Let us define

$$\chi_k(\tau,f) = \int h_k(t)h_k(t-\tau)\, e^{-j2\pi ft}\, dt \quad , \tag{66}$$

so that (65) becomes

$$\Gamma_{ac}(f_1,f_2) = \sum_k \int \tilde{R}(\tau)\, \chi_k(\tau,f_1-f_2)\, e^{-j2\pi f_2\tau}\, d\tau \quad . \tag{67}$$

This is about as far as we can conveniently go without solving for a specific example.

3.3 THE GAUSSIAN BEAM

If we assume that the antenna beamshape is Gaussian then the integrations become relatively tractable. In particular, let

$$h_k(t) = e^{-at^2} \tag{68}$$

for the moment. Then it can be shown that

$$X_k(\tau,f) = \sqrt{\frac{\pi}{2a}}\; e^{-\pi^2 f^2/2a}\; e^{-a\tau^2/2}\; e^{-j\pi f\tau} \quad . \tag{69}$$

Now if the antenna pattern is defined by

$$g^2(\theta) = e^{-(\alpha\theta/\theta_{3dB})^2} \quad , \tag{70}$$

where $\alpha = 1.6651$ and θ_{3dB} is the one-way half-power beamwidth, we note that from (60) and (68)

$$a = (\alpha\dot{\theta}/\theta_{3dB})^2 \quad , \tag{71}$$

and

$$\tau \rightarrow \tau + (\theta_o - \theta_k)/\dot{\theta} \quad . \tag{72}$$

Thus (69) is given by

$$X_k(\tau,f) = \sqrt{\frac{\pi}{2}}\,\frac{\theta_{3dB}}{\alpha\dot{\theta}}\; e^{-\frac{1}{2}(\pi f\theta_{3dB}/\alpha\dot{\theta})^2}\; e^{-\frac{1}{2}(\alpha\dot{\theta}\tau/\theta_{3dB})^2}\; e^{j2\pi f((\theta_o-\theta_k)/\dot{\theta}-\tau/2)} \quad . \tag{73}$$

Note that in (67) the subscript k is only associated with the ambiguity function, and in (73) the subscript k only appears in the last term. Thus if we sum (73) over k we must evalute the following expression

$$\sum_k e^{j2\pi f((\theta_o-\theta_k)/\dot{\theta}-\tau/2)} \quad . \tag{74}$$

However, as the increments become small so that the summation approaches an integration, (74) will have a significant value only for $f \approx 0$. As the limits on the integration get larger and larger, (74) becomes more like a δ-function, $\delta(f)$. For our purposes we will assume that it is a δ-function because we will never employ long enough integration times to be able to resolve the difference in shape of (74) from a δ-function. We must also include a scale factor in the evaluation of (74) as

$$\sum_k e^{j2\pi f((\theta_o - \theta_k)/\dot{\theta} - \tau/2)} = \frac{\dot{\theta}}{\Delta\theta}\,\delta(f) \tag{75}$$

where $\Delta\theta$ is the azimuth spacing of the clutter samples around the range ring (or the average spacing if the samples are nonuniformly spaced). With this assumption the summation of (73) over k becomes

$$\sum_k \chi_k(\tau,f) = \sqrt{\frac{\pi}{2}}\,\frac{\theta_{3dB}}{\alpha\Delta\theta}\,e^{-\frac{1}{2}(\alpha\dot{\theta}\tau/\theta_{3dB})^2}\,\delta(f) \quad . \tag{76}$$

Furthermore, (67) reduces to

$$\Gamma_{ac}(f_1,f_2) = \delta(f_1-f_2)\cdot\int \tilde{R}(\tau)\chi(\tau)\,e^{-j2\pi f_2\tau}\,d\tau \quad , \tag{77}$$

where

$$\chi(\tau) = \sqrt{\frac{\pi}{2}}\,\frac{\theta_{3dB}}{\alpha\Delta\theta}\,e^{-\frac{1}{2}(\alpha\dot{\theta}\tau/\theta_{3dB})^2} \quad . \tag{78}$$

Note that (77) is just the autocorrelation function of the beam scan modulation function in (60) with the Gaussian beam assumption. Furthermore, (77) is singular along the $f_1 = f_2$ axis, which means that the resultant process is stationary [7]. If we designate $S_{ac}(f)$ as the power spectrum of this stationary process, then we can write

$$S_{ac}(f) = \int \tilde{R}(\tau)\chi(\tau)\,e^{-j2\pi f\tau}\,d\tau \quad .$$

Now let us define the following Fourier transform pairs

$$S_{im}(f) \leftrightarrow \tilde{R}(\tau) \tag{80}$$

$$G(f) \leftrightarrow \chi(\tau) \tag{81}$$

so that the spectrum of the composite process in (79) can be written as

$$S_{ac}(f) = S_{im} \bigstar G(f) \tag{82}$$

where $S_{im}(f)$ is the spectrum of the clutter internal motion, G(f) is the scan modulation spectrum, and the star designates a convolution.

Let us continue with the assumption of a Gaussian beam. We will define

$$f_g = \frac{\alpha^2 \dot{\theta}}{\sqrt{2}\pi\theta_{3dB}} \tag{83}$$

so that when we take the Fourier transform of (77) we obtain

$$G(f) = \sqrt{\frac{\pi}{2}}\ \frac{\theta_{3dB}}{\alpha\Lambda\theta}\ \underbrace{\frac{\alpha}{\sqrt{\pi}\, f_g}\ e^{-(\alpha f/f_g)^2}}_{\text{normalized to unit power}} \tag{84}$$

Note that f_g is the two-sided half-power spectral width. Let us also assume that $\tilde{R}(\tau)$ is a Gaussian-shaped autocorrelation function

$$\tilde{R}(\tau) = \frac{P_T \Lambda\theta}{2\pi}\ e^{-(b\tau)^2} \tag{85}$$

Note that this autocorrelation function applies to a single clutter sample; therefore we have chosen P_T as the average power associated with all clutter within a range ring throughout 2π radians in azimuth. The Fourier transform of (85) is also Gaussian shaped as

$$S_{im}(f) = \frac{P_T \Lambda\theta}{2\pi}\ \underbrace{\frac{\alpha}{\sqrt{\pi}\, f_{im}}\ e^{-(\alpha f/f_{im})^2}}_{\text{normalized to unit power}} \tag{86}$$

where

$$b = \pi f_{im}/\alpha$$

and f_{im} is the two-sided half-power width of the internal motion spectrum. The convolution of (84) and (86) is

$$S_{ac}(f) = \underbrace{P_T}_{\text{total power in range ring}} \; \underbrace{\frac{\sqrt{\pi}}{\alpha}}_{1.064} \; \underbrace{\frac{\theta_{3dB}}{\sqrt{2} \cdot 2\pi}}_{\text{2-way half-power beamwidth : } 2\pi} \; \underbrace{\frac{\alpha}{\sqrt{\pi}\, f_{ac}} \; e^{-(\alpha f/f_{ac})^2}}_{\text{normalized to unit power}} \qquad (87)$$

where

$$f_{ac}^2 = f_{im}^2 + f_g^2 \qquad (88)$$

We can neglect the scan modulation when $f_{im} >> f_g$, or from (83) when

$$f_{im} >> .624\ \dot{\theta}/\theta_{3dB} \qquad (89)$$

We can replace the $>>$ sign with a > 4.0 without any noticeable effect on performance.

4. GENERATING CLUTTER SEQUENCES FOR A GROUND-BASED RADAR

In Reference 8 several general techniques are described for generating clutter sequences. They are all based on properties of the discrete Fourier transform and the fact that samples in the frequency domain will be statistically independent of each other. We will generate a set of independent random phasors $\{X(n\Delta f)\}$ at uniformly spaced increments, Δf, in the frequency domain, such that

$$\overline{|X(n\Delta f)|^2} = \Delta f\, S(n\Delta f) \tag{90}$$

where $S(f)$ is the desired power spectral density and the bar on the left side of (90) designates an ensembel average. The amplitude distribution of the phasors $\{X(n\Delta f)\}$ need not be Rayleigh [8]. The correlated time sequence is obtained by taking the discrete Fourier transform as

$$x(k\Delta t) = \sum_n X(n\Delta f)\; e^{j2\pi kn\Delta f\Delta t} \tag{91}$$

Usually the sample spacing in the time domain, Δt, will be given (e.g., the pulse repetition period in a pulsed radar), but Δf is under our control. Before we define how Δf should be chosen we note that the time sequence repeats with a period given by

$$T_r = 1/\Delta f \tag{92}$$

Let us also define the two sided half-power width of the spectrum as f_{3dB}. The correlation distance is then approximately

$$T_c = 1/f_{3dB} \tag{93}$$

Since the time sequence repeats with a period T_r, we can utilize only a portion of the period without having the beginning of the desired sequence being correlated with the end. We should therefore choose

$$T_r \geq T + T_c \tag{94}$$

where T is the duration of the desired sequence. If we make use of (92) and (93) we can rewrite (94) as

$$\Delta f \leq \frac{f_{3dB}}{1 + f_{3dB}T} \tag{95}$$

There is one other constraint on Δf: We must be able to resolve the spectrum. From Reference 8 (p. 120) we must have $\Delta f \leq 0.63\ f_{3dB}$ for a Gaussian-shaped spectrum. The constant will be slightly different for other spectral shapes, but it will probably not be less than 0.63 for ground-based radar clutter spectra. Therefore we can write

$$\Delta f \leq 0.63\ f_{3dB} \cdot \min\left\{1, \frac{1.60}{1 + f_{3dB}T}\right\} \tag{96}$$

In general, $f_{3dB}T$ will be less than 0.60 for a ground-based surveillance radar, so the simple constraint $\Delta f \leq 0.63\ f_{3dB}$ will apply in most cases; however, (96) will always be applicable.

There are two general ways to implement (91): the fast Fourier transform (FFT) and brute-force approaches. We will now discuss each.

4.1 FFT APPROACH

Let us define the repetition frequency as

$$f_r = 1/\Delta t \tag{97}$$

Next, let us divide this interval into N_r equal increments so that

$$\Delta f = f_r/N_r \tag{98}$$

From (97) we note that

$$N_r = 1/\Delta f \Delta t \tag{99}$$

so that (91) can be written as

$$x(k\Delta t) = \sum_{n=0}^{N_r-1} X(n\Delta f)\ e^{j2\pi kn/N_r} \tag{100}$$

which is the conventional definition of the discrete Fourier transform that can be implemented as an FFT.

In order to determine how large N_r should be let us utilize (98) and the inequality $\Delta f \leq 0.63\ f_{3dB}$, and write

$$N_r \geq 1.60\ f_r/f_{3dB} \tag{101}$$

with the understanding that N_r might have to be even larger if (95) were to apply. Now let us work with some numbers. A general rule of thumb is that the spectral width of ground clutter for a ground-based radar with a non-scanning antenna is about 3% of the maximum wind Doppler [8]. Thus we can write

$$f_{3dB} = .06\ V_w/\lambda$$

where V_w is the wind velocity and λ is the wavelength. For V_w = 10 m/sec and λ = .12 m (S-band) we have f_{3dB} = 5 Hz. If the PRF is 1 kHz (f_r = 1000 Hz) then (101) becomes $N_r \geq 320$, which is a large number, especially considering that we might utilize at most only about 16 samples in the time domain. The disadvantage of the FFT approach, at least as defined so far, is that most of the spectral samples will be zero. Even though the FFT is efficient, it must still implement the multiplies by zero. The next approach avoids this shortcoming.

[illegible].2 BRUTE-FORCE APPROACH

The brute-force approach is a direct implementation of (91). We define the samples in the frequency domain only over a limited region of the power spectral density. If we utilize $\Delta f = 0.63\ f_{3dB}$, then as few as 5 or 7 samples will be sufficient to define the spectral process (as determined in Reference 8 for Gaussian shaped spectra; the rule might be different for other spectral shapes). Moreover (91) is implemented only for the desired number of time samples. If N_f and N are the number of frequency and time domain samples, respectively, then the computation time will be proportional to the product NN_f, instead of $N_r \log_2 N_r$ for the FFT approach. But since $N \ll N_r$ and $N_f \ll N_r$ will usually be true for ground-based radars, the brute force approach will usually be faster than the FFT approach for generating clutter spectra.

4.3 FFT APPROACH WITH INTERPOLATION

If $f_r >> f_{3dB}$, as it is with most ground-based radars, then many consecutive time-domain samples will be correlated. We can increase the sample spacing Δt in (91) or (100) to reduce the computation time for the Fourier transform, and then utilize interpolation to obtain samples at the desired rate. Let us define the reduced sample spacing in the time domain as $\Delta t'$, and

$$h = \Delta t/\Delta t' \tag{102}$$

Similarly, we will define $f_r' = 1/\Delta t'$, so that

$$h = f_r'/f_r \tag{103}$$

Furthermore, we will define the number of samples in one repetition interval in the frequency domain as

$$N_r' = f_r'/\Delta f \tag{104}$$

and as a result

$$N_r = N_r'/h \tag{105}$$

Interpolation in the time domain causes spurious responses in the frequency domain [8]. If we wish to hold these spurious responses to 50 dB below the desired clutter power, we must choose $f_r' \geq 10\ f_{3dB}$ if linear interpolation is used [8, Eq. 8.61]. Now we can use (103) and write $h \geq 10\ f_{3dB}/f_r$; however, if this ratio is greater than unity we will not implement interpolation. Thus we can write an equality that will provide -50 dB spurious responses for linear interpolation as

$$h = \min\{10\ f_{3dB}/f_r,\ 1\} \tag{106}$$

If we use $\Delta f \leq 0.63\ f_{3dB}$ in (104) we can also write

$$N_r' \geq 1.60\ f_r'/f_{3dB} \tag{107}$$

which is equivalent to (101) applied to the reduced sampling rate. If we combine (107) with (103) we obtain

$$N_r' \geq 1.60\, h \cdot f_r/f_{3dB} \tag{108}$$

and with (106) we have

$$N_r' \geq 16 \qquad , \text{ if } f_{3dB}/f_r \leq 0.1 \tag{109a}$$

$$\geq 1.6\, f_r/f_{3dB}, \text{ otherwise} \quad . \tag{109b}$$

Note that (109a) provides the largest lower bound on N_r', so that we can write for all cases $N_r' \geq 16$. This is a factor of 20 less than N_r in the example in Section 4.1.

Let us define $N = T/\Delta t$. Then from (93), (94), and (97) we can write

$$N_r \geq N + f_r/f_{3dB} \tag{110}$$

With (105) and $h = 10\, f_{3dB}/f_r$, we obtain another lower limit on N_r' as

$$N_r' \geq (N + 10)h \tag{111}$$

Usually for a ground based surveillance radar this limit will be lower than (109a).

In Table 1 we list the performance that can be obtained for various parameter choices. The use of the table will begin with a specification of the desired spurious spectral response level and accuracy. For efficient computation the lowest value of N_r' should be used, although some consideration should be given to the use of $N_r' = 16, 32, 64$, etc., because these values are especially efficient FFT sizes. We will be given N, the number of time samples being utilized, and the ratio f_{3dB}/f_r. We will then compute

$$h = (f_r'/f_{3dB})\, f_{3dB}/f_r \tag{112}$$

where (f_r'/f_{3dB}) is given in the table. If $h \geq 1$ we will not implement interpolation. Next we will check the constraint on N in (111), which we rewrite as

$$N \leq N_r'/h - (f_r'/f_{3dB}) \tag{113}$$

Table 1. Performance for Various Interpolation Parameters

Option	N_r'	f_r'/f_{3dB}	$\Delta f/f_{3dB}$	Spurs*	Accuracy**	Constraint
A	16	10	0.625	-50 dB	2.0%	$N \leq 16/h-10$
B	20	10	0.50	-50	0.1	$\leq 20/h-10$
C	20	12	0.60	-53	1.5	$\leq 20/h-12$
D	24	12	0.50	-53	0.1	$\leq 24/h-12$
E	32	16	0.50	-58	0.1	$\leq 32/h-16$
F	40	16	0.40	-58	--	$\leq 40/h-16$
G	48	16	0.333	-58	--	$\leq 48/h-16$
H	64	16	0.25	-58	--	$\leq 64/h-16$

*Spurious spectral responses for linear interpolation [8, Eq. 8.61]

**Sampling error [8, Figure 8.5 for $M = 7$ and $m = 2$]

If it is satisfied then we can use that particular option; otherwise, we will have to use a larger value of N_r'. Let us now consider the example $f_{3dB}/f_r = .01$ and $N = 32$, and assume that Option A would be suitable if it satisfies the constraint on N. We obtain $h = 0.10$ and the constraint $N \leq 150$, which works. On the other hand, suppose that $f_{3dB}/f_r = .05$ and the performance of Option E were the minimum acceptable. Now we obtain $h = 0.80$ and $N \leq 64$, which works (Options F and G would also work). We should comment that when h is close to unity, as in the last example, interpolation may not offer any computational advantage over the straight FFT approach. We could have chosen $\Delta f = f_r/64 = .3125\ f_{3dB}$, and we could have utilized $N = 64 - 20 = 44$ time samples without interpolation, which would have resulted in a slightly faster computation. Interpolation pays off when $h < 0.5$, especially so for very small values of h. When $0.5 \leq h < 1.0$ the issue is not so decisive as many factors come into play.

We can compare the computation times for the brute-force and FFT (with interpolation) approaches. For the former the computation time is proportional to $N \cdot N_f$, where N_f is the number of samples defining the power spectral density. For the FFT the computation time is proportional to $N_r' \log_2 N_r'$. On one particular computer the brute-force approach will be faster when N satisfies the following:

N_r'	$N_f = 5$	$N_f = 7$
16	$N \leq 5$	$N \leq 4$
32	$N \leq 14$	$N \leq 10$

For $N_r' = 16$ the constraints of $N \leq 4$ or $N \leq 5$ will not usually be of practical interest in radar simulations; even if we were interested in simulating a few samples it would be still relatively efficient to utilize the FFT approach with interpolation. For $N_r' = 32$ there might be a few situations in which the brute-force approach would offer a computational advantage, but the advantage will never be great.

5. SIMULATION PROGRAM

A very convenient, flexible, and computationally efficient Fortran program has been developed to simulate detection performance for a ground-based radar against airborne targets. The quadrature components of the video signal are simulated in the neighborhood of the target, and the received signal is processed in the receiver in the same manner and sequence as it would be in an actual radar. The scattering environment is described in a statistical manner, as well as the fine-scale location of the target within the azimuth beam, range gate, and Doppler filter. By repeating the run with independent random inputs, the detection statistics are accumulated in the form of cumulative probability versus threshold setting. If a target is absent, then the output is probability of false alarm versus threshold setting. Importance sampling techniques can be used to increase the efficiency of this calculation.

Practically any situation of interest can be simulated with parameters specified on input. Listings of the program as given in Appendix A include a description of all input parameters, as well as the procedures, assumptions, and limitations within each simulation step. The main program (MAIN) acts as a driver, in which calls are made to subroutines that generate specific signals such as

- the target (TARGET)
- clutter (CLUTTR)
- receiver noise (NOISE)

in addition to a subroutine that processes the received signal (PROCES) and accumulates detection performance statistics (DSTINL,DSTPNT,DSTFUN). The simulation procedure can be easily modified. For example, as written, the main program permits two types of clutter to be generated, ground plus rain or ground plus chaff; all three types can be generated with three calls to CLUTTR (which also requires additional parameters to be defined on input).

5.1 INPUT PARAMETERS

All input parameters are capable of being defined in DATA statements in the main program or via NAMELIST /VALUES/ for each simulation run. We will discuss these parameters as they relate to each simulation function.

The Target

The nominal location of the target is defined in terms of its range (R) and altitude above a spherical earth (HT). Its velocity is incorporated in the simulation by defining in which Doppler filter (KFILT) the target appears (KFILT = 1 for dc). The average radar cross section (RCS) of the target is also specified, as well as a fluctuation parameter (KSWER). The fluctuation is assumed to be slow (pulse-to-pulse correlated but scan-to-scan independent), and the following cases can be handled:

KSWER	Case
0	nonfluctuating
1	Swerling Case 1
2	Swerling Case 3
>0	chi-square with 2*KSWER degrees of freedom

The target position as described above is nominal; for each statistical replication the actual position is varied randomly throughout the azimuth beam, the range gate, and the Doppler filter. The beam scanning and straddling losses are thus incorporated in the simulation.

Ground Clutter

Spatially nonhomogeneous ground clutter can be simulated. The model used in this program is a block correlated one, in which a value of σ_o (the backscatter coefficient) is generated for one square of size RCOR from a statistical distribution that can be log-normal (SVP>0.) or Weibull (SVP<0.). The σ_o for adjacent squares in the format of Figure 12 is generated independently, but within any square the σ_o is constant (homogeneous) with an assumed Rayleigh amplitude distribution applying to each elemental area within the square. The global average σ_o is defined as SIGØ on input. For homogeneous ground clutter set SVP=0.

Two models for the clutter spectrum (not including the scan modulation) are included. They are a Gaussian shape (IT=0) and a general shape (IT>0) of

$$s(f) = \frac{1}{1 + (f/f_o)^n}$$

where n = IT (the main program restricts the choice to IT = 2 or 3). The half-power spectral width relative to the PRF is defined as FWTR. The above description for the clutter spectrum is designated as the "ac component." There is also a "dc component," the power of which is DCAC times the power in the ac component.

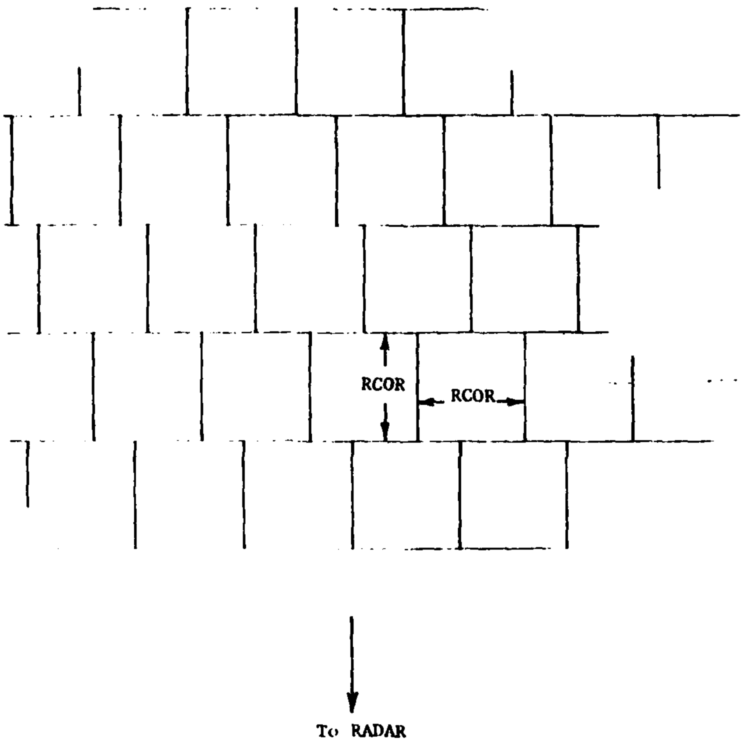


Figure 12. Format of Block Correlated Squares on the Ground (there is random staggering from one row to the next)

The user must determine if there will be any ground clutter at all because of terrain masking, and whether there will be any enhancement of short range clutter in range-ambiguous situations (R>C/(2.*PRF), where C is the propagation velocity).

Rain/Chaff Clutter

The volumetric clutter is assumed to be homogeneously distributed in a layer between two altitudes (H1 and H2, H2 > H1) relative to a spherical earth. The reflectivity, or backscatter cross section per unit volume, is defined as REF. For rain clutter set H1=0. A Gaussian shape is assumed for the fluctuation spectrum.

Wind

The spectral width of all types of clutter, and the mean Doppler of volumetric clutter, are functions of the wind speed (VW) and direction with respect to the antenna beam axis (ANGL).

The System

The radar system is defined by the following parameters:

PT = peak transmit power
GØ = peak antenna gain, one-way
WL = wavelength
BW = pulse (resolution) bandwidth
PRF = pulse repetition frequency
FL = product of noise figure and rf losses (L>0)
AZ3DB = half-power width of azimuth beam, one way
ITYPE = type of scan (=1 for step, =2 for continuous)
AZDWEL = azimuth angle through which the antenna steps (ITYPE=1) or scans (ITYPE=2) from dwell to dwell*
PSAT = receiver saturation level, after pulse cancellation but before Doppler filtering
NCOH = number of pulses coherently integrated
NNCOH = number of pulse groups noncoherently integrated at same frequency
NCNCL = number of stages of pulse cancellation
NR = number of range gates simulated (=1 for conventional threshold detection, >1 but odd for CFAR processing--the number of CFAR reference cells is NR-1)
LAW = envelope detection law (=1 for linear, =2 for square law)

*a dwell consists of NCOH*NNCOH+NCNCL pulses.

The processing stages in the receiver are assumed to be in the following order, beginning at the front end:

- range gating
- pulse cancellation
- saturation
- Doppler filtering
- envelope detection
- threshold detection or CFAR processing

The resultant clutter spectrum for the scanning antenna (ITYPE=2) is modeled on the basis of the Gaussian shape for the beam and input spectrum.

Simulation

The number of statistical replications under the same nominal conditions is specified as NREP. For false alarm analysis set RCS=0., and importance sampling can be invoked with KSW=1 (conventional sampling will be implemented with KSW=0). The distortion parameter for importance sampling is XM, which should be established by trial and error (too small of a value will result in infrequent false alarms, while too large of a value will cause underflow/overflow or other diagnostics; XM=2. is a reasonable first guess). As set by a DATA statement the number of azimuth samples simulated for clutter is NA=21, and the spacing of the samples is computed in MAIN as DA=.2*AZ3DB. The user can decrease the running time by choosing NA=11 and DA=.4*AZ3DB, or NA=9 and DA=.5*AZ3DB, to encompass $\pm$ 2 beamwidths of the azimuth mainbeam. Further out sidelobes can be accommodated by choosing NA and DA accordingly.

Debug printouts can be obtained by setting IDBG>0. In addition, it is possible to obtain a printout of the signal spectrum in the first range gate (or any range gate with suitable program modifications).

5.2 INTERPRETING THE RESULTS

For conventional threshold detection (non-CFAR) the output of the Doppler filter(s) under test is normalized by the input noise power divided by NCOH, which is the approximate (receiver) noise power in a Doppler filter on output (it would be the noise power if the filter weights were uniform). These detected outputs are then accumulated in a histogram, and the histogram is integrated to obtain the cumulative distribution function versus relative threshold setting. For CFAR processing the output of the Doppler filter under test is normalized by the average of the NR-1 range gated outputs forming the CFAR reference (half of the gates are in front of the range gate under test, and half are behind it).

In order to determine what the threshold setting should be set RCS=0. to get the simulated false alarm probability versus threshold setting. Usually, the false alarm probability will be given. Then with RCS>0. we can read out the probability of detection that corresponds to the desired threshold setting.

5.3 EXAMPLES

If NNCOH=1, KSWER=1, LAW=2, and NR=1, all signals appearing in a non-clutter region of the Doppler spectrum will be Rayleigh amplitude and the probability of exceeding a threshold that is normalized by the average output noise power is given by

$$P(T) = e^{-T/(1 + SNR)} \tag{114}$$

where T is the normalized threshold and SNR is the output signal-to-noise ratio.

For the first example we will examine the false alarm performance with importance sampling. The simulated parameter values that deviate from the built-in data values are RCS=0., SIGO=0., NCOH=1, KFILT=1, NREP=1000, KSW=1, and XM=13.8. The simulation output is given in Table 2. Since SNR=0 in (114) we can easily compute what the theoretical results should be. For example, P(8.0) = .000335, P(10.0) = .0000454, and P(12.0) = .00000614; all values fall within 10% of the corresponding values in Table 2.

For the second example we will repeat the above conditions except we will utilize four Doppler filters (NCOH=4, KFILT=3). The simulation output is given in Table 3. Because of Hamming Weighting in the Doppler filtering process, the noise bandwidth of a Doppler filter is about 25% larger than the PRF divided by *the number of filters (NCOH)*. If we thus multiply the values of T in Table 3 by 0.80 prior to the use of (114), we will get good statistical agreement.

For the third example, we add a target (RCS=4.) so that the input signal-to noise ratio is a factor NCOH higher (4 × 8.0 = 32.0), reduced by the following effects

Antenna scan	.69	(-1.6 dB)
Doppler filter straddle	.90	(-0.5 dB)
Range gate straddle	.58	(-2.3 dB)
Total	.36	(-4.4 dB)

Table 2

THRESHOLD CROSSING STATISTICS

THRESHOLD T	FROM THAT DATA.GE.T
0.00	1.21380057
.20	.84770106
.40	.71958792
.60	.59584591
.80	.45942613
1.00	.35374626
1.20	.30398993
1.40	.24210467
1.60	.19860592
1.80	.17010872
2.00	.13864017
2.20	.11651527
2.40	.09599186
2.60	.08122836
2.80	.06652662
3.00	.04646717
3.20	.03632204
3.40	.03108665
3.60	.02610827
3.80	.02130364
4.00	.01740784
4.20	.01350029
4.40	.01123024
4.60	.00911666
4.80	.00774567
5.00	.00632966
5.20	.00534393
5.40	.00392760
5.60	.00310967
5.80	.00258871
6.00	.00208503
6.20	.00184439
6.40	.00162270
6.60	.00142592
6.80	.00117393
7.00	.00092323
7.20	.00070414
7.40	.00062227
7.60	.00046485
7.80	.00042215
8.00	.00031433
8.20	.00026295
8.40	.00022604
8.60	.00015499
8.80	.00013776
9.00	.00011658
9.20	.00009863
9.40	.00007433
9.60	.00005582
9.80	.00004729
10.00	.00004165
10.20	.00003933
10.40	.00003326
10.60	.00002719
10.80	.00002040
11.00	.00001635
11.20	.00001351
11.40	.00001195
11.60	.00001074
11.80	.00000806
12.00	.00000652
12.20	.00000539
12.40	.00000464
12.60	.00000413
12.80	.00000309
13.00	.00000273
13.20	.00000221
13.40	.00000185
13.60	.00000126
13.80	.00000086
14.00	.00000073
14.20	.00000064
14.40	.00000057
14.60	.00000047
14.80	.00000039
15.00	.00000031
15.20	.00000025
15.40	.00000022
15.60	.00000017
15.80	.00000014
16.00	.00000012
16.20	.00000010
16.40	.00000008
16.60	.00000007
16.80	.00000006
17.00	.00000005
17.20	.00000003
17.40	.00000002
17.60	.00000002
17.80	.00000001
18.00	.00000001
18.20	.00000001
18.40	.00000001
18.60	.00000001
18.80	.00000001
19.00	.00000001
19.20	.00000000
19.40	.00000000
19.60	.00000000
19.80	.00000000
20.00	.00000000

Table 3

THRESHOLD CROSSING STATISTICS

THRESHOLD T	PROB THAT DATA.GE.T
0.00	.18263636
.20	.17811398
.40	.17805438
.60	.15218296
.80	.15214634
1.00	.15110649
1.20	.15108545
1.40	.15106213
1.60	.14669242
1.80	.14644394
2.00	.14641651
2.20	.13592534
2.40	.13583332
2.60	.12456066
2.80	.09873398
3.00	.09224525
3.20	.06018307
3.40	.06012936
3.60	.06005682
3.80	.05990731
4.00	.05272144
4.20	.04537119
4.40	.03748572
4.60	.01141535
4.80	.01138476
5.00	.01138207
5.20	.01014165
5.40	.01013869
5.60	.01012378
5.80	.00997911
6.00	.00986891
6.20	.00983875
6.40	.00983671
6.60	.00972320
6.80	.00959998
7.00	.00157811
7.20	.00152927
7.40	.00141015
7.60	.00140948
7.80	.00131221
8.00	.00090462
8.20	.00088224
8.40	.00088083
8.60	.00081441
8.80	.00080644
9.00	.00080610
9.20	.00059735
9.40	.00051021
9.60	.00050608
9.80	.00050546
10.00	.00039346
10.20	.00039271
10.40	.00039266
10.60	.00021992
10.80	.00021918
11.00	.00021917
11.20	.00021861
11.40	.00006844
11.60	.00006216
11.80	.00005073
12.00	.00005031
12.20	.00005029
12.40	.00002847
12.60	.00002766
12.80	.00002690
13.00	.00002644
13.20	.00002644
13.40	.00002506
13.60	.00002506
13.80	.00002505
14.00	.00002505
14.20	.00002418
14.40	.00002337
14.60	.00002337
14.80	.00002337
15.00	.00002331
15.20	.00000202
15.40	.00000201
15.60	.00000161
15.80	.00000161
16.00	.00000161
16.20	.00000161
16.40	.00000160
16.60	.00000160
16.80	.00000158
17.00	.00000158
17.20	.00000158
17.40	.00000156
17.60	.00000151
17.80	.00000145
18.00	.00000145
18.20	.00000143
18.40	.00000143
18.60	.00000143
18.80	.00000143
19.00	.00000139
19.20	.00000139
19.40	.00000139
19.60	.00000137
19.80	.00000132
20.00	.00000050

Thus the output signal-to-noise ratio is SNR = 11.5. In Table 4 we show the simulation results, which are in good statistical agreement with (114).

Table 4

THRESHOLD CROSSING STATISTICS

THRESHOLD T	PROB THAT DATA.GE.T	THRESHOLD T	PROB THAT DATA.GE.T
0.00	1.00000000	10.20	.54000000
.20	.99000000	10.40	.54000000
.40	.97000000	10.60	.53000000
.60	.96000000	10.80	.52000000
.80	.95000000	11.00	.51000000
1.00	.94000000	11.20	.51000000
1.20	.93000000	11.40	.50000000
1.40	.93000000	11.60	.49000000
1.60	.92000000	11.80	.48000000
1.80	.92000000	12.00	.46000000
2.00	.90000000	12.20	.46000000
2.20	.89000000	12.40	.46000000
2.40	.89000000	12.60	.46000000
2.60	.89000000	12.80	.46000000
2.80	.87000000	13.00	.46000000
3.00	.86000000	13.20	.45000000
3.20	.86000000	13.40	.44000000
3.40	.86000000	13.60	.43000000
3.60	.85000000	13.80	.42000000
3.80	.84000000	14.00	.42000000
4.00	.84000000	14.20	.42000000
4.20	.82000000	14.40	.42000000
4.40	.80000000	14.60	.41000000
4.60	.79000000	14.80	.40000000
4.80	.79000000	15.00	.39000000
5.00	.78000000	15.20	.39000000
5.20	.77000000	15.40	.39000000
5.40	.77000000	15.60	.39000000
5.60	.76000000	15.80	.39000000
5.80	.76000000	16.00	.39000000
6.00	.75000000	16.20	.39000000
6.20	.73000000	16.40	.39000000
6.40	.71000000	16.60	.39000000
6.60	.69000000	16.80	.38000000
6.80	.67000000	17.00	.38000000
7.00	.66000000	17.20	.37000000
7.20	.65000000	17.40	.37000000
7.40	.65000000	17.60	.35000000
7.60	.65000000	17.80	.35000000
7.80	.63000000	18.00	.35000000
8.00	.63000000	18.20	.34000000
8.20	.63000000	18.40	.33000000
8.40	.61000000	18.60	.32000000
8.60	.61000000	18.80	.32000000
8.80	.60000000	19.00	.31000000
9.00	.60000000	19.20	.31000000
9.20	.59000000	19.40	.31000000
9.40	.59000000	19.60	.30000000
9.60	.59000000	19.80	.30000000
9.80	.58000000	20.00	.29000000
10.00	[illegible]		

6. REFERENCES

1. Hansen, V. G., "Importance Sampling in Computer Simulation of Signal Processors," *Computers and Electrical Engineering*, Vol. 1, pp 545-550, Pergamon Press, 1974.

2. Hansen, V. G., "Detection Performance of the Narrow-Band Wilcoxon Detector Against Gaussian Noise," *IEEE Trans. Information Theory*, Vol. IT-18, pp 664-667, Sept. 1972.

3. Hansen, V. G. and H. R. Ward, "Detection Performance of the Cell Averaging LOG/CFAR Receiver," *IEEE Trans. AES*, Vol. AES-8, pp 648-652, Sept. 1972.

4. Hansen, V. G., "Detection Performance of Some Nonparametric Rank Tests and an Application to Radar," *IEEE Trans. Information Theory*, Vol. IT-16, pp 309-318 (especially Section V), May 1970.

5. Cantrell, B. H., and G. V. Trunk, "Correction to Angular Accuracy of a Scanning Radar Employing a Two-Pole Filter," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-10, pp 878-880, November 1974.

6. Dahlquist, G., and A. Björck, *Numerical Methods*, Englewood Cliffs, NJ: Prentice-Hall, pp 460-461, 1974.

7. Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill, 1965.

8. Mitchell, R. L., *Radar Signal Simulation*, Artech House, 1976.

APPENDIX A
RADAR SIGNAL SIMULATION PROGRAM TO DETERMINE DETECTION PERFORMANCE

FORTRAN LISTINGS

```
      PROGRAM MAIN(INPUT,OUTPUT)
C
C        ********************************************
C        *                                          *
C        *     RADAR SIGNAL SIMULATION PROGRAM      *
C        *     TO DETERMINE DETECTION PERFORMANCE   *
C        *                                          *
C        *     BY RL MITCHELL, MARK RESOURCES INC   *
C        *     PREPARED FOR RADC/HANSCOM AFB        *
C        *     SEPT 1980                            *
C        *                                          *
C        ********************************************
C
C THE OBJECTIVE OF THIS SIMULATION PROGRAM IS TO DETERMINE DETECTION
C PERFORMANCE FOR A GROUND BASED RADAR AGAINST AN AIRBORNE TARGET.  THE
C QUADRATURE VIDEO SIGNALS ARE GENERATED FOR THE TARGET, GROUND CLUTTER,
C RAIN/CHAFF CLUTTER, AND RECEIVER NOISE, AND THE COMBINED SIGNAL IS
C PROCESSED IN A SIMULATED RECEIVER THAT INCLUDES SOME OR ALL OF THE
C FOLLOWING OPERATIONS..... PULSE CANCELLATION MTI, RECEIVER SATURATION,
C DOPPLER FILTERING VIA FFT, ENVELOPE DETECTION, NONCOHERENT INTEGRA-
C TION AND CELL-AVERAGING CFAR PROCESSING.  IN THE ABSENCE OF A TARGET,
C IMPORTANCE SAMPLING CAN BE USED TO EFFICIENTLY AND RELIABLY ESTABLISH
C FALSE ALARM PERFORMANCE AT LOW FALSE ALARM RATES.  THIS SIMULATION
C PROGRAM ALSO INCORPORATES MANY NOVEL FEATURES TO COVER A WIDE VARIETY
C OF ENVIRONMENTAL AND SYSTEM CONDITIONS.....
C
C       1. TARGET FLUCTUATION STATISTICS.
C             THE TARGET IS ASSUMED TO BE SLOWLY FLUCTUATING (COMPLETE
C       CORRELATION PULSE-TO-PULSE, BUT SCAN-TO-SCAN INDEPENDENT).  THE
C       STATISTICS ARE SPECIFIED AS ONE OF THE CHI-SQUARE MODELS THAT IN-
C       CLUDE NONFLUCTUATING AND SWERLING CASES 1 AND 3.  NO FREQUENCY
C       DIVERSITY IS SIMULATED.
C
C       2. RANDOM GROUND TERRAIN.
C             THE GROUND CLUTTER IS ASSUMED TO BE BLOCK CORRELATED.
C       WITHIN A SQUARE CELL ON THE GROUND (OF DIMENSIONS RCORR) A RANDOM
C       NUMBER IS GENERATED TO REPRESENT THE BACKSCATTER COEFFICIENT
C       (SIGMA-ZERO) FOR THAT CELL.  THESE CELLS WILL BE DESIGNATED AS
C       CLUTTER CELLS.  INDEPENDENT RANDOM NUMBERS ARE GENERATED FOR ALL
C       OTHER CELLS.  HOMOGENEOUS TERRAIN CAN ALSO BE ASSUMED.
C
C       3. NONGAUSSIAN STATISTICS FOR CLUTTER CELLS.
C             THE RANDOM NUMBERS ASSIGNED TO THE CLUTTER CELLS IN THE
C       NONHOMOGENEOUS TERRAIN CAN BE GENERATED FROM WEIBULL OR LOG-
C       NORMAL DISTRIBUTION FUNCTIONS.  AN OVERALL AVERAGE BACKSCATTER
C       COEFFICIENT IS SPECIFIED ON INPUT.
C
C       4. GROUND CLUTTER SPECTRUM.
C             THE SHAPE OF THE GROUND CLUTTER SPECTRUM CAN BE EITHER
C       GAUSSIAN OR 1/(1+(F/F0)**N).  THE WIDTH OF THIS SPECTRUM IS PRE-
```

```
C        PORTIONAL TO THE WIND VELOCITY.  THERE IS ALSO A CAPABILITY TO
C        ADD A DC COMPONENT THAT WOULD REPRESENT STATIONARY CLUTTER.
C
C
C        5. RAIN/CHAFF CLUTTER.
C              HOMOGENEOUS VOLUMETRIC CLUTTER IS ASSUMED BETWEEN TWO ALT-
C        ITUDES, THE LOWER ONE BEING AT GROUND LEVEL FOR RAIN.  THE
C        SPECTRUM FOR RAIN/CHAFF CLUTTER IS GAUSSIAN SHAPED, WITH THE
C        WIDTH BEING PROPORTIONAL TO THE WIND VELOCITY.....MANY FACTORS
C        ENTER INTO THE CONSTANT OF PROPORTIONALITY AS DISCUSSED BY
C        NATHANSON (REF 2), AND ONLY A FIRST-CUT IS PROVIDED IN THIS
C        PROGRAM.
C
C        6. ANTENNA SCAN.
C              TWO TYPES OF ANTENNA SCANS CAN BE SIMULATED, A CONTINUOUS
C        SCAN OR A STEP SCAN.  THE CONTINUOUS SCAN MODULATION FOR THE
C        TARGET IS SIMULATED ON A PULSE-BY-PULSE BASIS, BUT FOR CLUTTER
C        AN EQUIVALENT SPECTRAL BROADENING FACTOR IS COMPUTED ON THE BASIS
C        OF AN ASSUMED GAUSSIAN-SHAPED AZIMUTH BEAM.  THE ELEVATION BEAM
C        IS ASSUMED TO BE A FAN BEAM, WITH A COSECANT-SQUARE SHAPE.
C
C        7. LOCATION OF TARGET.
C              THE TARGET IS ASSUMED TO BE RANDOMLY LOCATED WITHIN.....
C                 A. THE AZIMUTH SCAN INCREMENT CORRESPONDING TO A DWELL
C                 B. THE RANGE RESOLUTION CELL
C                 C. A DOPPLER FILTER (IF DOPPLER FILTERING IS EMPLOYED)
C                    OR THE PRF (IF NO DOPPLER FILTERING IS EMPLOYED).
C
C        8. DOPPLER SEARCH.
C              WHEN NO TARGET IS PRESENT (FALSE ALARM SIMULATION) A
C        SEARCH IS PERFORMED OVER THOSE DOPPLER FILTERS THAT HAVE BEEN
C        CALCULATED TO CONTAIN NO CLUTTER.  FOR SOME LOWER-FREQUENCY
C        RADARS, SAY AT L-BAND, THE RAIN CLUTTER THAT APPEARS IN A DOPPLER
C        FILTER MAY BE LESS THAN THERMAL NOISE SO THAT THE ALGORITHM USED
C        IN THIS PROGRAM TO DETERMINE THE FIRST AND LAST FILTERS (K1 AND
C        K2) WILL NOT BE CORRECT.....THE USER MUST ANALYZE THIS SITUATION,
C        AS WELL AS THE SPECTRAL WIDTH OF RAIN CLUTTER.
C
C
C THE FOLLOWING PARAMETERS ARE READ FROM NAMELIST /VALUES/.....
C
C            R = RANGE OF TARGET OR CELL UNDER TEST
C           HT = TARGET ALTITUDE
C          RCS = TARGET RCS (AVERAGE)
C         SIGC = GROUND CLUTTER BACKSCATTER COEF (SIGMA-ZERO)
C          SVP   SPATIAL VARIABILITY PARAMETER                      SPTVAR
C         RDCR   CLUTTER DECORRELATION DISTANCE
C         DCAC = RATIO OF DC POWER TO AC POWER
C          ETA = VOLUMETRIC CLUTTER (RAIN/CHAFF) REFLECTION COEF
C                (RCS PER UNIT VOLUME)
```

```
C        H1 = LOWER ALTITUDE OF RAIN/CHAFF CLUTTER (H1=0 FOR RAIN)
C        H2 = UPPER ALTITUDE OF RAIN/CHAFF CLUTTER
C        VW = WIND SPEED
C      ANGL = ANGLE BETWEEN WIND AND ANTENNA BORESIGHT
C        PT = PEAK TRANSMIT POWER
C        GG = PEAK ANTENNA POWER GAIN (ONE-WAY)
C        WL = WAVELENGTH
C        BW = PULSE BANDWIDTH
C       PRF = PULSE REPETITION FREQUENCY
C        FL = PRODUCT OF NOISE FIGURE AND SYSTEM LOSSES
C     AZ3DB = ONE-WAY HALF-POWER WIDTH (RAD)                          AZGAIN
C    AZDWEL = AZ STEP BETWEEN DWELLS (ITYPE=1) OR
C             AZ SCAN DURING DWELL    (ITYPE=2)
C      PSAT = SATURATION POWER LEVEL                                  PROCES
C        XM = DISTORTION OF SAMPLE MEAN FOR IMPORTANCE SAMP           GAUSSI
C        NR = TOTAL NUMBER OF RANGE CELLS PROCESSED                   PROCES
C      NCOH = SIZE OF FFT IN DOPPLER PROCESSOR                        PROCES
C     NNCOH = NUMBER OF PULSE GROUPS NONCOH INTEGRATED                PROCES
C     NCNCL = STAGES OF CANCELLATION                                  PROCES
C       LAW = LAW OF FIRST DETECTOR                                   PROCES
C        JJ = PARAMETER SPECIFYING SPECTRAL SHAPE                     RANSEC
C     KSWER = SWERLING CHI-SQUARE PARAMETER (SEE REF.3)               TARGET
C     KFILT = DOPPLER FILTER NUMBER IN WHICH TARGET APPEARS           TARGET
C      NREP = NUMBER OF REPETITIONS OF SIMULATION
C       KSW = SWITCH FOR IMPORTANCE SAMPLING (ON=1, OFF=0)
C     ITYPE = TYPE OF SCAN (1 FOR STEP, 2 FOR CONTINUOUS)
C      IDIM = DIMENSION OF XR,XI,YR,YI ARRAYS
C      IDBG = DEBUG FLAG (OFF IDBG.LE.0, ON IDBG.GT.0)
C
C THESE PARAMETERS ARE COMPUTED AND PRINTED IN NAMELIST /PARAM1/.....
C
C      RAMB = FIRST RANGE AMBIGUITY (=C/(2.*PRF))
C        DR = RANGE RESOLUTION CELL SIZE (=C/(2.*BW))
C      PSCL = SCALING FACTOR, CONVERTS RCS TO RECEIVED POWER
C        EL = TARGET ELEVATION ANGLE
C      TPOW = RECEIVED POWER FOR TARGET
C      CPOW = RECEIVED POWER FOR UNIT AREA OF GROUND CLUTTER
C      RPOW = RECEIVED POWER FOR UNIT AREA OF RAIN/CHAFF CLUTTER
C    SIGDBG = EQUIVALENT BACKSCATTER COEF FOR RAIN/CHAFF CLUTTER
C        PN = RECEIVER NOISE POWER
C       SNR = SINGLE-PULSE SIGNAL-TO-NOISE RATIO (RECEIVER NOISE ONLY)
C       SCR = SINGLE-PULSE SIGNAL-TO-CLUTTER RATIO (GROUND CLUTTER ONLY)
C       SVR = SINGLE-PULSE SIGNAL-TO-CLUTTER RATIO (VOLUME CLUTTER ONLY)
C      AREA = CLUTTER CELL AREA ON GROUND (=.707*R*DR*AZ3DB)
C       DAZ = AZIMUTH SCAN INCREMENT PER PULSE REPETITION PERIOD
C    AZRATE = ANTENNA SCAN RATE
C      FMAX = MAXIMUM DOPPLER OF WIND
C    FMAXTR = FMAX/PRF
C      FDTR = FMAXTR*COS(ANGLE)
C    RATPGR = SPECTRAL WIDTH OF GROUND CLUTTER NORMALIZED TO PRF
```

```
C    FWTRVL = SPECTRAL WIDTH OF RAIN/CHAFF CLUTTER NORMALIZED TO PRF
C    FWTRSN = SPECTRAL BROADENING OF CLUTTER DUE TO SCAN MODULATION
C     TNORM = THRESHOLD NORMALIZATION (USED FOR CONVENTIONAL DETECTION)
C     NPULS = NUMBER OF PULSES PROCESSED FOR DWELL (=NCOH*NNCOH*NCNCL)
C      NTOT = NPULS*NS
C        K1 = FIRST FILTER FOR DOPPLER SEARCH                        PROCES
C        K2 = LAST  FILTER FOR DOPPLER SEARCH                        PROCES
C
C THESE PARAMETERS ARE SET IN DATA STATEMENTS.....
C
C       XKT = BOLTZMANNS CONSTANT TIMES NOISE TEMPERATURE
C         C = PROPAGATION VELOCITY
C        RE = EFFECTIVE RADIUS OF EARTH (4/3 TIMES ACTUAL)
C        NA = NUMBER OF AZIMUTH SAMPLES COMPUTED FOR CLUTTER         CLUTTR
C        NT = NUMBER OF THRESHOLD SAMPLES
C        T1 = THRESHOLD VALUE FOR FIRST SAMPLE
C        DT = SPACING OF THRESHOLD SAMPLES
C        NS = NUMBER OF SPECTRAL SAMPLES IN SPECTRUM ANALYSIS
C
C ALL UNITS MUST BE CONSISTENT THROUGHOUT.....
C
C              LENGTH - METERS
C              AREA   - SQ. METERS
C              ANGLE  - RADIANS
C              TIME   - SEC
C              FREQ   - HZ
C              POWER  - WATTS
C
C NOTE.....NO CHECK IS MADE FOR RANGE AMBIGUOUS OPERATION.  IF R.GT.RAMB
C THERE MAY BE ENHANCED CLUTTER DUE TO FOLDOVER OF SHORT-RANGE CLUTTER.
C THIS ENHANCEMENT CAN BE ACCOMMODATED BY REDEFINING SIGO AND REF.
C
C THERE IS NO PROVISION IN THIS PROGRAM FOR.....
C
C       1. PULSE COMPRESSION
C       2. FREQUENCY DIVERSITY
C       3. PULSE REPETITION INTERVAL STAGGERING
C
C THE XR,XI,YR, AND YI ARRAYS MUST BE DIMENSIONED AS LARGE AS NTOT.
C ARRAY TT MUST BE DIMENSIONED AS LARGE AS NT.
C ARRAYS S AND WGT MUST BE DIMENSIONED AS LARGE AS NS.
C
C THE FOLLOWING REFERENCES ARE NOTED.....
C
C     1. R. MITCHELL, RADAR SIGNAL SIMULATION, ARTECH HOUSE, 1976
C     2. F. NATHANSON, RADAR DESIGN PRINCIPLES, MC GRAW-HILL,1969
C     3. R. MITCHELL, ET AL., RECURSIVE METHODS FOR ......,IEEE/AES,JUL 71
C     4. ABRAMOWITZ + STEGUN, HANDBOOK OF MATHEMATICAL FUNCTIONS
C
      DIMENSION XR(256),XI(256),YR(256),YI(256),S(128),WGT(128)
```

```
      DIMENSION TI(101)
      COMMON /ZQ4G6/ IQ56
      COMMON /ISAM/ KSW,ISW,XM,NSUM,WSUM
      COMMON /COM1/ NF,NPULS,NA,NCOH,F,DF,DA,AZ3DB
      COMMON /TEMP/ WORK(256)

      NAMELIST /VALUES/ R,HT,RCS,SIGO,SVP,RCOR,DCAC,KEF,H1,H2,VW,ANGL,
     1                  PT,GC,WL,PW,PRF,FL,AZ3DB,AZDWEL,PSAT,XP,
     2                  NF,NCOH,NNCOH,NCNCL,LAW,IT,KSWER,KFILT,NREP,KSW,
     3                  ITYPE,IDIM,IDBG
      NAMELIST /PARAM1/ RAMP,DR,PSCL,EL,TPOW,CPOW,RPOW,SIGCEO,PN,SNR,
     1                  SCR,SZZ,AREA,DAZ,AZRATE,FMAX,FMAXTR,FOTR,
     2                  FWTRGR,FWTRVL,FWTRSN,TNORM,NPULS,NTOT,K1,K2
      NAMELIST /PARAM2/ DA,NA,XKT,C,PI,NT,T1,DT

      DATA R/1.E5/          ,HT/1000./      ,RCS/1./        ,KSWER/1/
      DATA KFILT/9/         ,PT/1.E5/       ,GO/4.E3/       ,WL/.1/
      DATA PW/1.E6/         ,PRF/1000./     ,FL/10./        ,PSAT/1.E99/
      DATA NF/1/            ,NCOH/16/       ,NNCOH/1/       ,NCNCL/0/
      DATA LAW/2/           ,SIGO/1.E-4/    ,SVP/0./        ,RCOR/1.E6/
      DATA DCAC/0./         ,IT/0/          ,VW/10./        ,ANGL/0./
      DATA KEF/0./          ,H1/0./         ,H2/1000./      ,NREP/1/
      DATA KSW/0/           ,XM/2./         ,AZ3DB/.05/     ,AZDWEL/.05/
      DATA XKT/4.-21/       ,C/3.E8/        ,RE/8.5E6/      ,NA/21/
      DATA ITYPE/1/         ,IDIM/256/      ,IDBG/0/
      DATA NT/101/          ,T1/0./         ,DT/.2/         ,NS/128/
      DATA FOURPI/12.5664/

    5 READ VALUES
      IF(NREP.LE.0) STOP
      PRINT VALUES
      NPULS=NCOH*NNCOH+NCNCL
      NTOT=NPULS*NF
      XR(IDIM+1)=123.
C     IF(XI(1).NE.123.)                    STOP 30
      IF(NNCOH.LE.0)                       STOP 31
      IF(NTOT.GT.IDIM)                     STOP 32
      IF(IT.LT.0.OR.IT.GT.3)               STOP 33
      IF(KFILT.LT.1.OR.KFILT.GT.NCOH)      STOP 34
      IF(MOD(NF,2).EQ.0)                   STOP 35
      IF(LAW.LT.1.OR.LAW.GT.2)             STOP 36
      IF(KSW.LT.0.OR.KSW.GT.1)             STOP 37
C COMPUTE AZIMUTH SAMPLE SPACING
      DA=.2*AZ3DB
C COMPUTE POWER SCALING IN RADAR RANGE EQUATION
      PSCL=PT*(GO*WL)**2/(FOURPI**3)/R**4
C COMPUTE ELEVATION ANGLE OF TARGET
      EL=HT/R-R/(2.*RE)
C COMPUTE RECEIVED TARGET POWER
      TPOW=0.
```

```
      IF(EL.GT.0.) TPOW=PSCL*RCS*ELGAIN(EL)**2
C COMPUTE RECEIVED CLUTTER POWER FROM UNIT AREA ON GROUND (ZERO ELEVA-
C TION ANGLE IS ASSUMED)
      CPOW=PSCL*SIGO*ELGAIN(0.)
C COMPUTE EQUIVALENT SIGMA-ZERO OF RAIN/CHAFF
      DH=.01*(H2-H1)
      H=H1
      SUM=0.
      IF(REF.LE.0.) GO TO 22
      DO 20 J=1,101
      EL=H/R-R/(2.*RE)
      IF(EL.GT.0.) SUM=SUM+ELGAIN(EL)**2
      H=H+DH
   20 CONTINUE
   22 SIGOEC=REF*SUM*DH
C COMPUTE RECEIVED RAIN/CHAFF CLUTTER POWER PER UNIT CROSS-SECTION AREA
      RPOW=PSCL*SIGOEC
C COMPUTE NOISE POWER
      PN=XKT*BW*FL
C COMPUTE SIGNAL-TO-NOISE RATIO (RECEIVER NOISE ONLY)
      SNR=1.E99
      IF(PN.GT.0.) SNR=TPOW/PN
C COMPUTE RANGE CELL WIDTH
      RW=C/(2.*BW)
C COMPUTE FIRST RANGE AMBIGUITY
      RAMB=C/(2.*PRF)
C COMPUTE CLUTTER CELL AREA ON GROUND
      AREA=.707*R*RW*AZ3DB
C COMPUTE SIGNAL-TO-CLUTTER RATIO (GROUND CLUTTER)
      SCR=1.E99
      IF(CPOW.GT.0.) SCR=TPOW/(CPOW*AREA)
C COMPUTE SIGNAL-TO-CLUTTER RATIO (VOLUME CLUTTER)
      SVR=1.E99
      IF(RPOW.GT.0.) SVR=TPOW/(RPOW*AREA)
C COMPUTE REMAINING SCAN PARAMETERS
      DAZ=0.
      IF(ITYPE.LT.11) DAZ=AZDWEL/FLOAT(NPULS)
      AZRATE=DAZ*PRF
C COMPUTE WIND PARAMETERS
      FMAX=2.*VW/WL
      FMAXIP=FMAX/PRF
      ICTF=FMAXIP*[illegible](CANCL)
C COMPUTE GROUND AND VOLUME CLUTTER SPECTRAL WIDTHS (SEE NATHANSONS BOOK
C SEE [illegible] SECTIONS 7.4, 7.4, 7.5, 7.12 FOR MORE INFORMATION)
      FWIDGR=.01*FMAXIP
      FWIDVL=.20*FMAXIP
C COMPUTE CLUTTER SPECTRAL BROADENING DUE TO SCAN MODULATION
      FASN=.024*AZRATE/AZ3DB
      FWIDSN=FASN/PRF
C RECOMPUTE SPECTRAL WIDTHS
```

```
      FXI=(C*SQRT(2.)*(FSX/(2.*(2*SNX*2)
      FXIN=(C*SQRT(2.)*(FSX/(2.*(SNX*2)
C COMPUTE FILTER NUMBERS FOR DOPPLER SEARCH
      K1=KFILT
      K2=KFILT
      IF(FCS.GT.0.) GO TO 25
      K1=1
      K2=NCOH
      IF(SIGD.LE.0.) GO TO 24
      K1=3
      K2=NCOH-1
   24 IF(SIGDF.LE.0.) GO TO 25
      FMX=AMAX1(FCT+FWTRVL,0.)+100.
      FMN=AMIN1(FCT-FWTRVL,0.)+100.
      K1=FMX*NCOH
      K2=FMN*NCOH
      K1=MOD(K1+1,NCOH)
      K2=MOD(K2-1,NCOH)
   25 CONTINUE
C COMPUTE THRESHOLD NORMALIZATION
      TNORM=1.
      IF(NF.EQ.1) TNORM=SQRT(PN/FLOAT(NCOH))**LA
      PRINT PARAM1
      PRINT PARAM2

C SIMULATE SIGNALS, PROCESSOR, AND THRESHOLD STATISTICS
      CALL XMIT(-NS,0.,S)
      CALL ESTIN(LTI,CT,NT)
      DO 70 IREP=1,NREP
      IF(IREP.GT.10) IDEBG=0
      NSUM=0
      WSUM=0.
      CALL XMIT(-NTOT,0.,XR)
      CALL XMIT(-NTOT,0.,XI)
      IF(PN.LE.0.) GO TO 32
      CALL NOISE(YR,YI,PN)
      CALL ASUM(NTOT,XR,YR,XR)
      CALL ASUM(NTOT,XI,YI,XI)
      IF(IDEBG.LE.0) GO TO 32
      CALL PRNT(YR,NTOT,-4,6H    YR,20HOUTPUT OF NOISE       )
      CALL PRNT(YI,NTOT,-4,6H    YI,20HOUTPUT OF NOISE       )
   32 IF(SIGD.LE.0.) GO TO 34
      CALL CLUTR(YR,YI,CPDX,RLOX,SVP,IT,DLAL,FWTRVL,0.)
      CALL ASUM(NTOT,XR,YR,XR)
      CALL ASUM(NTOT,XI,YI,XI)
      IF(IDEBG.LE.0) GO TO 34
      CALL PRNT(YR,NTOT,-4,6H    YR,20HOUTPUT OF CLUTTR-GR )
      CALL PRNT(YI,NTOT,-4,6H    YI,20HOUTPUT OF CLUTTR-GR )
   34 IF(SIGDF.LE.0.) GO TO 36
      CALL CLUTTR(YR,YI,FCOS,1.,1.,0.,0.,FWTRVL,FOT)
```

```
      CALL ASUM(NTOT,XR,YR,XR)
      CALL ASUM(NTOT,XI,YI,XI)
      IF(IPRG.LE.0) GO TO 36
      CALL PRNT(YR,NTOT,-4,6H      YR,20HOUTPUT OF CLUTTR-RN )
      CALL PRNT(YI,NTOT,-4,6H      YI,20HOUTPUT OF CLUTTR-RN )
   36 IF(-(S.LE.0.)) GO TO 38
      AZI=.5*AZBWEL*(RAN(0.)-1.)
      CALL TARGET(YR,YI,TPDW,KSWER,KFILT,AZI,DAZ)
      CALL ASUM(NTOT,XR,YR,XR)
      CALL ASUM(NTOT,XI,YI,XI)
      IF(IPRG.LE.0) GO TO 38
      CALL PRNT(YR,NTOT,-4,6H      YR,20HOUTPUT OF TARGET     )
      CALL PRNT(YI,NTOT,-4,6H      YI,20HOUTPUT OF TARGET     )
C HERE WE PERFORM A SPECTRAL ANALYSIS ON THE FIRST RANGE RING
   38 CALL XMIT(-NS,0.,YR)
      CALL XMIT(-NS,0.,YI)
      CALL XMIT(NPULS,XR,YR)
      CALL XMIT(NPULS,XI,YI)
      CALL SPCTRM(YR,YI,NPULS,NS,.08,WGT)
      CALL ASUM(NS,YR,S,S)
C HERE WE PROCESS THE SIGNAL
      CALL PROCES(XR,XI,K1,K2,LAW,NNCUM,NCNCL,PSAT)
      NK=K2-K1+1
      W=1.
      IF(KSW.EQ.1) W=(XMS*NSUM)*EXP(-(1.-1./XM)*WSUM)
      IF(W.LT.0.) GO TO 65
      DO 40 K=1,NK
      CALL PSTPNT(X(K)/XNORM,W)
   40 CONTINUE
   50 CONTINUE
C PRINT OUT SPECTRUM OF FIRST RANGE RING
      SNORM=1./ZSUM(NS,S)
      CALL XMIT(-NS,SNORM,S,S)
      CALL PRNT(S,NS,6,6H      S,20HSPECTRUM OF 1ST GATE)
C PRINT OUT THRESHOLD STATISTICS
      CALL PSTFGN(TT)
      PRINT 110
      DO 60 J=1,NT
      T=T1+(J-1)*DT
      PROB=1.-TT(J)
      PRINT 111,T,PROB
   60 CONTINUE
      GO TO 5
   65 PRINT 112,NSUM,WSUM,W
      STOP

  110 FORMAT(///30H THRESHOLD CROSSING STATISTICS//
     1      ...HTHRESHOLD...PROB THAT/ 1OX1HT,7X9HDATA.GE.T/)
  111 FORMAT(F13.2,F14.7)
  112 FORMAT(////54H ERROR IMPORTANCE SAMPLING WEIGHT IS NEGATIVE/
     1      ...NSUM,WSUM,W.....I5,2E13.6)
      END
```

```
      SUBROUTINE TARGET(XR,XI,TPOW,KSWER,KFILT,AZ1,DAZ)
C
C IN THIS SUBROUTINE THE COMPLEX VIDEO SIGNAL FOR THE TARGET IS CREATED
C IN THE ARRAY-PAIR (XR,XI).  THE TARGET DOPPLER IS RANDOM WITHIN THE
C FILTER NUMBER KFILT (IF NCOH.GT.1) OR WITHIN THE PRF (IF NCOH=1).
C THE AZIMUTH BEAM SWEEPS PAST THE TARGET DURING THE DWELL OF NPULS
C PULSES BEGINNING AT AZ1 WITH AN AZIMUTH INCREMENT OF DAZ FOR EACH
C PULSE.  THE RCS PARAMETERS ARE.....
C
C       TPOW = TARGET AVERAGE RECEIVED POWER (=PSCL*RCS*ELGAIN**2)
C      KSWER = SWERLING CHI-SQUARE PARAMETER (SEE REF.3)
C
C THE FORMAT OF THE XR,XI-ARRAYS IS AS IF THEY WERE DIMENSIONED
C
C                       XR(NPULS,NR)        XI(NPULS,NR)
C
      DIMENSION XR(1),XI(1)
      COMMON /COM1/ NR,NPULS,NA,NCOH
      COMMON /DEBUG/ IDBG
      DATA TWOPI/6.2831853/
      POW=TPOW
      IF(KSWER.LE.0) GO TO 22
      PROD=1.
      DO 20 J=1,KSWER
      PROD=PROD*RANF(0.)
   20 CONTINUE
      POW=TPOW*(-ALOG(PROD))
   22 A=SQRT(POW)
      H=.5*RANF(0.)
      F=(KFILT+RANF(0.)-1.5)/FLOAT(NCOH)
      L=((NR-1)/2)*NPULS+1
      IF(IDBG.GT.0) PRINT 100,A,H,F
      CALL XMIT(-NR*NPULS,0.,XR)
      CALL XMIT(-NR*NPULS,0.,XI)
      AZ=AZ1
      DO 30 K=1,NPULS
      ARG=TWOPI*K*F
      SR=COS(ARG)
      SI=SIN(ARG)
      AMP=A*AZGAIN(AZ)
      XR(L)=AMP*(1.-H)*SR
      XI(L)=AMP*(1.-H)*SI
      IF(NR.LE.1) GO TO 25
      XR(L+NPULS)=AMP*H*SR
      XI(L+NPULS)=AMP*H*SI
   25 L=L+1
      AZ=AZ+DAZ
   30 CONTINUE
      RETURN
  100 FORMAT(///21H TARGET     A,H,F.....F12.4,2F12.4)
      END
```

```
      SUBROUTINE CLUTTR(XR,XI,CPOW,RCOR,SVP,IT,DCAC,FWTR,FDTR)
C
C IN THIS SUBROUTINE WE GENERATE THE COMPLEX VIDEO SIGNAL FOR CLUTTER
C WITH A STEP SCANNING ANTENNA.  THE CLUTTER SIGNAL IS OUTPUT TO THE
C ARRAY PAIR (XR,XI).  THE PARAMETERS ARE.....
C
C     CPOW = CLUTTER RCS PER UNIT AREA ON THE GROUND SCALED TO RECEIVED
C            POWER (=SIG0*FSCL)
C        R = RANGE OF INTEREST
C       DR = RANGE RESOLUTION CELL SIZE
C       DA = AZIMUTH SAMPLE SPACING
C       NA = NUMBER OF AZIMUTH SAMPLES
C     RCOR = CLUTTER CORRELATION DISTANCE
C      SVP = SPATIAL VARIABILITY PARAMETER
C     FDTR = MEAN DOPPLER OF CLUTTER RELATIVE TO THE PRF
C     FWTR = DOPPLER WIDTH OF CLUTTER RELATIVE TO THE PRF
C       IT = PARAMETER SPECIFYING SPECTRAL SHAPE (SEE SUB. RANSEQ)
C     DCAC = RATIO OF DC POWER TO AC POWER
C
C IN ADDITION N IS THE NUMBER OF SPECTRAL SAMPLES USED IN THE GENERATION
C OF THE CORRELATED RANDOM SEQUENCES.  IT SHOULD BE AT LEAST AS LARGE
C AS    NPULS+1./FWTR.
C
C THERE ARE TWO GENERAL CASES OF INTEREST.....
C
C                 GROUND CLUTTER                       RAIN/CHAFF CLUTTER
C        ---------------------------              -------------------------
C        SVP.NE.0.    (NONHOMOGENEOUS)            SVP=0.   (HOMOGENEOUS)
C        FDTR=0.                                  FDTR=(2*VW/WL)*COS(ANGL)
C        DCAC.GT.0.                               DCAC=0.
C                                                 IT=0
C
C IF THE ELEVATION BEAM DOES NOT HAVE PEAK GAIN ON THE GROUND AT THE
C RANGE OF INTEREST THEN CPOW MUST INCLUDE THE ELEVATION BEAM WEIGHTING.
C
C THE FORMAT OF THE XR,XI-ARRAYS IS AS IF THEY WERE DIMENSIONED
C
C                    XR(NPULS,NR)       XI(NPULS,NR)
C
      DIMENSION XR(1),XI(1)
      COMMON /COM1/ NR,NPULS,NA,NCOH,R,DR,DA
      COMMON /ISAM/ KSW,ISW
      COMMON /DBUG/ IDBG
      DATA TWOPI/6.2831853/
      NC=(NR+1)/2
      P=CPOW*R*DR*DA
      KR=RANF(0.)
      MR=-1
      L1=1
      DO 40 I=1,NR
```

```
      IF(MR.EQ.IFIX(RR)) GO TO 25
      SUM=0.
      AZ=-.5*(NA-1)*DA
      AA=RANF(0.)
      MA=0
      W=SPTVAR(SVP)
      DO 20 J=1,NA
      IF(MA.EQ.IFIX(AA)) GO TO 15
      W=SPTVAR(SVP)
      MA=AA
   15 SUM=SUM+P*W*AZGAIN(AZ)**2
      AZ=AZ+DA
      AA=AA+R*DA/RCOF
   20 CONTINUE
      MR=RR
   25 PAV=SUM
      IF(IDBG.GT.0) PRINT 100,I,PAV
      ISW=0
      IF(KSW.EQ.1.AND.I.EQ.NC) ISW=1
      CALL RANSEQ(XR(L1),XI(L1),NPULS,FWTR,IT,DCAC,PAV)
      IF(FDTR.EQ.0.) GO TO 35
      L=L1
      DO 30 K=1,NPULS
      ARG=TWOPI*FDTR*(K-1)
      C=COS(ARG)
      S=SIN(ARG)
      YR=C*XR(L)-S*XI(L)
      YI=C*XI(L)+S*XR(L)
      XR(L)=YR
      XI(L)=YI
      L=L+1
   30 CONTINUE
   35 RR=RR+DR/RCOR
      L1=L1+NPULS
   40 CONTINUE
      RETURN
  100 FORMAT(///21H CLUTTR     I,PAV.....I5,E12.4)
      END
```

```
      SUBROUTINE NOISE(XR,XI,PN)
C
C IN THIS SUBROUTINE THE THERMAL NOISE IS GENERATED IN THE ARRAY-PAIR
C (XR,XI).
C
C        PN = AVERAGE NOISE POWER
C
C THE FORMAT OF THE XR,XI-ARRAYS IS AS IF THEY WERE DIMENSIONED
C
C                         XR(NPULS,NR)       XI(NPULS,NR)
C
      DIMENSION XR(1),XI(1)
      COMMON /ISAM/ KSW,ISW
      COMMON /COM1/ NR,NPULS
      NC=(NR+1)/2
      L=1
      DO 20 I=1,NR
      ISW=0
      IF(KSW.EQ.1.AND.I.EQ.NC) ISW=1
      DO 20 K=1,NPULS
      CALL GAUSSI(XR(L),XI(L),PN)
      L=L+1
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PROCES(XR,XI,K1,K2,LAW,NNCOH,NCNCL,PSAT)
C
C IN THIS SUBROUTINE THE SIGNAL PROCESSING IN THE RECEIVER IS IMPLE-
C MENTED ON THE ARRAY-PAIR (XR,XI).  THE ARGUMENTS ARE.....
C
C        K1 = FIRST FILTER FOR DOPPLER SEARCH
C        K2 = LAST  FILTER FOR DOPPLER SEARCH
C       LAW = LAW OF FIRST DETECTOR
C     NNCOH = NUMBER OF PULSE GROUPS NONCOHERENTLY INTEGRATED
C     NCNCL = STAGES OF CANCELLATION
C      PSAT = SATURATION POWER LEVEL
C
C THE FOLLOWING CASES CAN BE HANDLED.....
C
C     PULSE CANCELLATION MTI WITH BINARY WEIGHTS
C
C         NCNCL = STAGES OF CANCELLATION (NCNCL+1 = NUMBER OF PULSES)
C         NCNCL = 0 CORRESPONDS TO NO PULSE CANCELLATION
C
C     DOPPLER FILTERING VIA FFT
C
C          NCOH = SIZE OF FFT
C          NCOH = 1 CORRESPONDS TO NO DOPPLER FILTERING
C
C     SATURATION BETWEEN PULSE CANCELLATION AND DOPPLER FILTERING
C
C          PSAT = SATURATION POWER LEVEL
C          PSAT = 1.E99 CORRESPONDS TO LINEAR PROCESSING (NO SATURATION)
C
C     LAW OF FIRST DETECTOR
C
C           LAW = 1 CORRESPONDS TO LINEAR DETECTOR
C           LAW = 2 CORRESPONDS TO SQUARE-LAW DETECTOR
C
C     NONCOHERENT INTEGRATION AT SAME CENTER FREQUENCY
C
C         NNCOH = NUMBER OF PULSE GROUPS NONCOHERENTLY INTEGRATED
C         NNCOH = 1 CORRESPONDS TO NO NONCOHERENT INTEGRATION
C
C     CFAR THRESHOLD REGULATION
C
C            NR = TOTAL NUMBER OF RANGE CELLS PROCESSED INCLUDING TARGET
C          NR-1 = NUMBER OF CFAR REFERENCE CELLS
C            NR = 1 CORRESPONDS TO CONVENTIONAL THRESHOLD PROCESSING
C
C THE INPUT COMPLEX SIGNAL IS IN THE ARRAY-PAIR (XR,XI), THE FORMAT OF
C WHICH IS AS IF THE DIMENSION WERE  XR(NPULS,NR),    XI(NPULS,NR).
C
C THE NONCOHERENTLY INTEGRATED OUTPUT OF FILTERS K1 THROUGH K2 APPEARS
C IN ARRAY XR IN SAMPLES 1 THROUGH K2-K1+1.  IF NO DOPPLER FILTERING
```

```
C VIA FFT IS IMPLEMENTED THEN NCOH=K1=K2=1.  USUALLY IF A TARGET IS
C PRESENT THEN K1=K2 WILL CORRESPOND TO THE DOPPLER SAMPLE IN WHICH THE
C TARGET WAS PLACED.  IF NO TARGET IS PRESENT THEN THERE ARE SEVERAL
C POSSIBILITIES.....
C
C     1. NO CLUTTER
C
C              SET K1=1,  K2=NCOH
C
C     2. GROUND CLUTTER ONLY WITH HAMMING FILTER WEIGHTS
C
C              SET K1=3,  K2=NCOH-1
C
C     3. GROUND PLUS WEATHER CLUTTER
C
C              SET K1 = FIRST DOPPLER SAMPLE NOT BLANKED
C                  K2 = LAST  DOPPLER SAMPLE NOT BLANKED
C
C FOR CFAR PROCESSING THE RANGE SAMPLE UNDER TEST IS ALWAYS THE CENTER
C SAMPLE.
C
      DIMENSION XR(1),XI(1)
      DIMENSION Z(128)
      COMMON /COM1/ NR,NPULS,NA,NCOH
      COMMON /DBUG/ IDBG
      IF(NPULS.NE.NNCOH*NCOH+NCNCL) STOP 1
      IF(MOD(NR,2).NE.1)            STOP 2
      IF(NCOH.EQ.1.AND.K1.NE.1)     STOP 3
      IF(NCOH.EQ.1.AND.K2.NE.1)     STOP 4
      NREF=NR-1
      NP=NPULS
      DO 50 I=1,NR
      L1=(I-1)*NPULS+1
      IF(IDBG.LE.0.OR.I.GT.IDBG) GO TO 22
      CALL PRNT(XR(L1),NPULS,-4,6H     XR,20HINPUT TO PROCES     )
      CALL PRNT(XI(L1),NPULS,-4,6H     XI,20HINPUT TO PROCES     )
   22 IF(NCNCL.EQ.0) GO TO 26
C HERE WE IMPLEMENT PULSE CANCELATION
      DO 25 M=1,NCNCL
      L=L1
      NP=NP-1
      DO 20 K=1,NP
      XR(L)=XR(L)-XR(L+1)
      XI(L)=XI(L)-XI(L+1)
      L=L+1
   20 CONTINUE
   25 CONTINUE
      IF(IDBG.LE.0.OR.I.GT.IDBG) GO TO 26
      CALL PRNT(XR(L1),NP,-4,6H     XR,20HAFTER PULSE CANCEL  )
      CALL PRNT(XI(L1),NP,-4,6H     XI,20HAFTER PULSE CANCEL  )
```

```
   26 IF(PSAT.GT.1.E90) GO TO 33
C HERE WE IMPLEMENT SATURATION
      L=L1
      DO 30 K=1,NP
      P=XR(L)**2+XI(L)**2
      IF(P.LE.PSAT) GO TO 30
      A=SQRT(PSAT/P)
      XR(L)=A*XR(L)
      XI(L)=A*XI(L)
      L=L+1
   30 CONTINUE
      IF(IDBG.LE.0.OR.I.GT.IDBG) GO TO 33
      CALL PRNT(XR(L1),NP,-4,6H    XR,20HAFTER SATURATION    )
      CALL PRNT(XI(L1),NP,-4,6H    XI,20HAFTER SATURATION    )
C HERE WE IMPLEMENT COHERENT FILTERING AND NONCOHERENT INTEGRATION
   33 L=L1
      CALL XMIT(-NCOH,0.,Z)
      DO 40 M=1,NNCOH
      CALL COHFLT(XR(L),XI(L),LAW,NCNCL)
      CALL ASUM(NCOH,XR(L),Z,Z)
      L=L+NCOH
   40 CONTINUE
      IF(IDBG.LE.0.OR.I.GT.IDBG) GO TO 42
      CALL PRNT(XR(L1),NF,-4,6H    XR,20HAFTER COH FILTER    )
   42 CALL XMIT(NCOH,Z,XR(L1))
   50 CONTINUE
      KK=K2-K1+1
      L1=K1
      M1=1
      DO 55 I=1,NR
      CALL XMIT(KK,XR(L1),XR(M1))
      L1=L1+NPULS
      M1=M1+KK
   55 CONTINUE
      IF(IDBG.LE.0) GO TO 57
      CALL PRNT(XR,NR*KK,-4,6H    XR,20HAFTER DETECTION     )
   57 IF(NR.EQ.1) GO TO 75
C HERE WE IMPLEMENT CFAR PROCESSING
      NRC=(NR+1)/2
      DO 70 K=1,KK
      L=K
      SUM=0.
      DO 60 I=1,NR
      IF(I.NE.NRC) SUM=SUM+XR(L)
      L=L+KK
   60 CONTINUE
      L=(NRC-1)*KK+K
      XR(K)=NREF*XR(L)/SUM
   70 CONTINUE
      IF(IDBG.LE.0) GO TO 75
      CALL PRNT(XR,KK,4,6H    XR,20HAFTER CFAR          )
   75 RETURN
      END
```

```
      SUPROUTINE CCHFLT(XR,XI,LAW,NCNCL)
C
C IN THIS SUBROUTINE WE IMPLEMENT COHERENT FILTERING CN THE COPPLEX-VIDEC
C SIGNAL IN THE ARRAY-PAIR (XR,XI) OF LENGTH NCOH.  THE DETECTED OUTPUT
C APPEARS IN ARRAY XR OF LENGTH NCOH.
C
C      NCOH = SIZE OF FFT IN COPPLER FILTER BANK
C     NCNCL = NUMBER OF STAGES OF PULSE CANCELLATION
C       LAW = ENVELOPE DETECTION LAW (1=LINEAP, 2=SQUARE-LAW)
C
C PULSE WEIGHTING IS APPLIED PRIOR TO COPPLER FILTERING.  THE WEIGHTING
C FUNCTION IS COSINE ON A PEDESTAL, WHERE
C
C     ALPHA = 1.   FOR UNIFORM WEIGHTING
C           = .08 FOR HAMMING WEIGHTING
C
      DIMENSION XR(1),XI(1)
      DIMENSION C(128),W(128)
      COMMON /COM1/ NR,NPULS,NA,NCCH
      DATA ALPHA/.08/
      DATA C(1)/1./
      DATA N1,N2/0,0/,PI/3.14159265/
      IF(NCOH.EQ.1) GO TO 17
      IF(N1.EQ.NCCH.AND.N2.EQ.NCNCL) GC TO 15
      N1=NCOH
      N2=NCNCL
      CALL WEIGHT(W,NCOH,ALPHA)
      C(1)=1.
      DO 12 K=2,NCCH
      C(K)=SIN(PI*(K-1)/FLOAT(NCOH))**(2*NCNCL)
   12 CONTINUE
   15 CALL PROD(NCCH,W,XR,XR)
      CALL PROD(NCOH,W,XI,XI)
      CALL FFT2(XR,XI,NCCH,-1)
   17 DO 25 K=1,NCOH
      XR(K)=XR(K)**2+XI(K)**2
      XR(K)=XR(K)/C(K)
      IF(LAW.EQ.1) XR(K)=SQRT(XR(K))
   25 CONTINUE
      RETURN
      END
```

```
      FUNCTION AZGAIN(AZ)
C
C COMPUTES ONE-WAY POWER GAIN AT AZIMUTH ANGLE AZ (RAD).  GAIN IS NORM-
C ALIZED TO A PEAK VALUE OF UNITY.
C
C IN THIS EXAMPLE A GAUSSIAN BEAMSHAPE IS ASSUMED WHERE
C
C     AZ3DB = ONE-WAY HALF-POWER WIDTH (RAD)
C
      DIMENSION G(42)
      COMMON /COM1/ DUMMY(7),AZ3DB
      DATA NN/0/
      IF(NN.GT.0) GO TO 25
      NN=1
      DO 20 I=1,42
      G(I)=EXP(-(1.6651*(.05*(I-1)))**2)
   20 CONTINUE
   25 A=20.*ABS(AZ)/AZ3DB
      IF(A.GT.40.) A=40.
      IA=A
      A=A-IA
      IA=IA+1
      AZGAIN=(1.-A)*G(IA)+A*G(IA+1)
      RETURN
      END
```

```
      FUNCTION ELGAIN(EL)
C
C COMPUTES ONE-WAY POWER GAIN AT ELEVATION ANGLE EL (RAD). GAIN IS NORM-
C ALIZED TO A PEAK VALUE OF UNITY.
C
C IN THIS EXAMPLE A COSECANT-SQUARE BEAMSHAPE IS ASSUMED.
C
      DIMENSION G(42)
      DATA PI2/1.57079632/
      DATA EL0/.1/
      DATA NN/0/
      IF(NN.GT.0) GO TO 25
      NN=1
      DO 20 I=1,41
      E=PI2*(I-1)/40.
      G(I)=1.
      IF(E.GT.EL0) G(I)=(SIN(EL0)/SIN(E))**2
   20 CONTINUE
      G(42)=G(41)
   25 E=40.*EL/PI2
      IF(E.GT.40.) E=40.
      IF(E.LT.0.)  E=0.
      IE=E
      E=E-IE
      IE=IE+1
      ELGAIN=(1.-E)*G(IE)+E*G(IE+1)
      RETURN
      END
```

```
      FUNCTION SPTVAR(SVP)
C
C THE RANDOM NUMBERS CORRESPONDING TO THE TERRAIN SPATIAL VARIABILITY
C ARE GENERATED IN THIS SUBROUTINE.  THERE ARE THREE CASES.....
C
C         SVP.LE.0.      WEIBULL DISTRIBUTION, -SVP=WEIBULL PARAMETER
C         SVP.EQ.0.      HOMOGENEOUS TERRAIN
C         SVP.GT.0.      LOG-NORMAL DISTRIBUTION, SVP=STD DEV OF
C                        LOG-VARIATE (NEPERS)
C
C IN ALL CASES THE MEAN VALUE OF THE RANDOM NUMBER IS UNITY.
C
C SEE REF.1, EQS 9.10 AND 9.11
C
      DATA SS/0./
      SPTVAR=1.
      IF(SVP.EQ.0.) RETURN
      IF(SVP.GT.0.) GO TO 20
      IF(SS.EQ.SVP) GO TO 10
      SS=SVP
      A=-SVP
      GA=GAMMA(A+1.)
   10 E=-ALOG(RANF(0.))
      SPTVAR=E**A/GA
      RETURN
   20 G=GAUSS(DUMMY)
      SPTVAR=EXP(SVP*(G-.5*SVP))
      RETURN
      END
```

```
      SUBROUTINE RANSEQ(XR,XI,N,FWTR,IT,DCAC,PAV)
C
C IN THIS SUBROUTINE WE GENERATE A CORRELATED SEQUENCE OF GAUSSIAN
C RANDOM PHASORS IN THE ARRAY-PAIR (XR,XI) OF LENGTH N.  THERE ARE TWO
C CASES FOR THE SPECTRAL SHAPE.....
C
C           IT=0         GAUSSIAN SPECTRAL SHAPE
C
C           IT.GT.0      SPECTRAL SHAPE IS   1/(1+(2F/FWTR)**IT)
C
C IN EITHER CASE FWTR IS THE RATIO OF THE 2-SIDED HALF-POWER SPECTRAL
C WIDTH AND THE PRF.  THE ABOVE SPECTRUM IS DESIGNATED AS THE AC-COMPON-
C ENT.  IT IS CENTERED AT DC.  THERE IS ALSO A DC-COMPONENT ADDED TO THE
C ABOVE, WHERE DCAC IS THE RATIO OF DC TO AC POWER.  THE AVERAGE POWER
C OF THE OUTPUT SAMPLES IS PAV.
C
C THE RANDOM SEQUENCE IS GENERATED BY THE FFT-METHOD WITH INTERPOLATION.
C THE PARAMETERS CHOSEN BELOW WILL PRODUCE A MAXIMUM OF -50DB SPURIOUS
C SPECTRAL RESPONSES WITH AN ACCURACY OF ABOUT 2-PERCENT.
C
      DIMENSION XR(1),XI(1),AR(257),AI(257),S(257)
      DATA FF/-1./,II/-1/,NN/-1/
      IF(FWTR.EQ.FF.AND.IT.EQ.II.AND.N.EQ.NN) GO TO 35
      FF=FWTR
      II=IT
      NN=N
      IF(FWTR.LE.0.) GO TO 65
      H=AMIN1(10.*FWTR,1.)
      NF=16
   10 NFH=NF/H-9.5
      IF(N.LE.NFH) GO TO 12
      NF=2*NF
      GO TO 10
   12 IF(NF.GT.256) STOP 11
      NF2=NF/2+1
      LIM=3*NF*FWTR/H
      S(1)=1.
      DO 20 K=2,NF2
      S(K)=0.
      IF(K.GT.LIM) GO TO 15
      IF(IT.LE.0) S(K)=EXP(-(1.6651*(K-1)*H/(FWTR*NF))**2)
      IF(IT.GT.0) S(K)=1./(1.+(2.*(K-1)*H/(FWTR*NF))**IT)
   15 S(NF+2-K)=S(K)
   20 CONTINUE
      SUM=ZSUM(NF,S)
      S(1)=S(1)+DCAC*SUM
      SUM=SUM*(1.+DCAC)/PAV
      CALL PROD(-NF,1./SUM,S,S)
   35 IF(FWTR.LE.0.) GO TO 65
      DO 40 K=1,NF
```

```
      CALL GAUSSI(AR(K),AI(K),S(K))
   40 CONTINUE
      CALL FFT2(AR,AI,NF,1)
      AR(NF+1)=AR(1)
      AI(NF+1)=AI(1)
      T=0.
      DO 50 K=1,N
      I=T
      TT=T-I
      I=I+1
      XR(K)=(1.-TT)*AR(I)+TT*AR(I+1)
      XI(K)=(1.-TT)*AI(I)+TT*AI(I+1)
      T=T+H
   50 CONTINUE
      GO TO 70
   65 CALL GAUSSI(BR,BI,FAV)
      CALL XMIT(-N,BR,XR)
      CALL XMIT(-N,BI,XI)
   70 RETURN
      END
```

```
      SUBROUTINE GAUSSI(X,Y,P)
C
C GENERATES RANDOM NUMBERS FOR THE FOLLOWING TWO DISTRIBUTION FUNCTIONS
C
C        CALL GAUSSI(X,Y,P)      GENERATES PAIR OF GAUSSIAN RANDOM
C                                PHASOR COMPONENTS OF AVERAGE POWER P
C
C        CALL EXPI(X)            GENERATES EXPONENTIAL RANDOM VARIABLE
C                                WITH UNIT AVERAGE POWER
C
C THIS SUBROUTINE SET CAN ALSO BE USED WITH IMPORTANCE SAMPLING IF WE
C SET.....
C
C      ISW = 1 TO ACTIVATE IMPORTANCE SAMPLING
C       XM = DISTORTION OF MEAN POWER
C
C FOR THE FIRST CALL WHEN IMPORTANCE SAMPLING IS IN EFFECT WE MUST SET
C
C     WSUM = 0.
C     NSUM = 0
C
C AFTER THE LAST CALL THE IMPORTANCE SAMPLING WEIGHT IS GIVEN BY
C
C        W = (XM**NSUM)*EXP(-(1.-1./XM)*WSUM)
C
      COMMON /ISAM/ KSW,ISW,XM,NSUM,WSUM
      DATA Z/0./
      ITYPE=1
      GO TO 10
      ENTRY EXPI
      ITYPE=2
   10 XMM=1.
      IF(ISW.EQ.1) XMM=XM
      E=XMM*(-ALOG(RANF(Z)))
      X=E
      IF(ISW.NE.1) GO TO 12
      WSUM=WSUM+E
      NSUM=NSUM+1
   12 IF(ITYPE.GE.2) RETURN
      E=SQRT(P*E)
   15 A=RANF(Z)
      A=A+A-1.
      B=RANF(Z)
      B=B+B-1.
      A2=A*A
      B2=B*B
      C=A2+B2
      IF(C.GT.1.) GO TO 15
      X=E*(A2-B2)/C
      Y=2.*E*A*B/C
      RETURN
      END
```

```
      FUNCTION GAUSS(DUMMY)
C
C GENERATES GAUSSIAN RANDOM NUMBER OF ZERO MEAN, UNIT VARIANCE.
C
      DATA I/0/
      IF(I) 1,1,2
    1 CALL GAUSSI(A,B,2.)
      GAUSS=A
      I=1
      GO TO 3
    2 GAUSS=B
      I=0
    3 RETURN
      END
```

```
      FUNCTION APROC(N,A,B,C)
C
C THIS SUBROUTINE PACKAGE PROCESSES ARRAYS.  IN EVERY CASE IABS(N) IS
C THE LENGTH OF ALL ARRAYS.  THERE ARE SEVERAL ENTRIES.....
C
C        CALL XMIT(-N,A,B)          B(K)=A(1)   FOR  K=1,N
C
C        CALL XMIT(N,A,B)           B(K)=A(K)   FOR  K=1,N
C
C        ZS=ZSUM(N,A)               ZS=SUM(A(K))   FOR  K=1,N
C
C        CALL ASUM(-N,A,B,C)        C(K)=A(1)+B(K)   FOR  K=1,N
C
C        CALL ASUM(N,A,B,C)         C(K)=A(K)+B(K)   FOR  K=1,N
C
C        DD=DOT(N,A,B)              DD=SUM(A(K)*B(K))   FOR  K=1,N
C
C        CALL PROD(-N,A,B,C)        C(K)=A(1)*B(K)   FOR  K=1,N
C
C        CALL PROD(N,A,B,C)         C(K)=A(K)*B(K)   FOR  K=1,N
C
C        E=ENGY(N,AR,AI)            E=SUM(AR(K)**2+AI(K)**2)   FOR  K=1,N
C
C        CALL POWR(N,AR,AI,P)       P(K)=AR(K)**2+AI(K)**2   FOR  K=1,N
C
C        CALL SWAP(N,A,B)           A(K) AND B(K) ARE SWAPPED FOR K=1,N
C
C SOME EXAMPLES ARE.....
C
C        SUM=ZSUM(N,A)              NORMALIZE A-ARRAY BY SUM
C        CALL PROD(-N,1./SUM,A,A)
C
C        E=ENGY(N,AR,AI)            NORMALIZE (AR,AI)-ARRAYS
C        ANORM=1./SQRT(E)           BY TOTAL ENERGY
C        CALL PROD(-N,ANORM,AR,AR)
C        CALL PROD(-N,ANORM,AI,AI)
C
C        SUMSQ=DOT(N,A,A)           SUM-SQUARE OF ELEMENTS IN A-ARRAY
C
C NOTE THAT.....
C
C        E=ENGY(N,AR,AI)=DOT(N,AR,AR)+DOT(N,AI,AI)
C
      DIMENSION A(1),B(1),C(1)
      ENTRY PROD
      IF(N) 10,12,15
   10 NN=-N
      AA=A(1)
      DO 11 K=1,NN
      C(K)=AA*B(K)
```

```
11 CONTINUE
12 RETURN
15 DO 18 K=1,N
   C(K)=A(K)*B(K)
18 CONTINUE
   RETURN
   ENTRY ENGY
   EE=0.
   DO 25 K=1,N
   EE=EE+A(K)**2+B(K)**2
25 CONTINUE
   APROC=EE
   RETURN
   ENTRY POWR
   DO 30 K=1,N
   C(K)=A(K)**2+B(K)**2
30 CONTINUE
   RETURN
   ENTRY ZSUM
   SUM=0.
   DO 35 K=1,N
   SUM=SUM+A(K)
35 CONTINUE
   APROC=SUM
   RETURN
   ENTRY ASUM
   IF(N) 40,42,45
40 NN=-N
   AA=A(1)
   DO 41 K=1,NN
   C(K)=AA+B(K)
41 CONTINUE
42 RETURN
45 DO 48 K=1,N
   C(K)=A(K)+B(K)
48 CONTINUE
   RETURN
   ENTRY SWAP
   DO 50 K=1,N
   AA=A(K)
   A(K)=B(K)
   B(K)=AA
50 CONTINUE
   RETURN
   ENTRY DOT
   DP=0.
   DO 60 K=1,N
   DP=DP+A(K)*B(K)
60 CONTINUE
   APROC=DP
```

```
      RETURN
      ENTRY XMIT
      IF(N) 62,66,67
   62 NN=-N
      AA=A(1)
      DO 65 K=1,NN
      B(K)=AA
   65 CONTINUE
   66 RETURN
   67 DO 70 K=1,N
      B(K)=A(K)
   70 CONTINUE
      RETURN
      END
```

```
      FUNCTION GAMMA(Z)
C
C COMPUTES GAMMA FUNCTION OF A REAL ARGUMENT
C SEE REF.4, ECS 6.1.15, 6.1.16, AND 6.1.35.
C
      DATA A1/-.5748646/,A2/.9512363/,A3/-.6998588/,A4/.4245549/,
     1     A5/-.1010678/
      X=Z-1.
      IF(X.LT.0.) STOP
      GAMMA=1.
   10 IF(X.LT.1.) GO TO 20
      GAMMA=GAMMA*X
      X=X-1.
      GO TO 10
   20 GAMMA=GAMMA*(((((A5*X+A4)*X+A3)*X+A2)*X+A1)*X+1.)
      RETURN
      END
```

```
      SUBROUTINE PRNT(X,N,NPL,A,WHERE)
C
C THIS SUBROUTINE PRINTS OUT ARRAY-X OF LENGTH N.  THE FORMAT IS.....
C
C        NPL.GT.0          F-FORMAT, NPL DECIMAL PLACES
C        NPL.LT.0          E-FORMAT,-NPL DECIMAL PLACES
C
C IN ADDITION.....
C
C          A = HOLLERITH NAME OF ARRAY BEING PRINTED (6H, RIGHT JUSTIFIED)
C      WHERE = HOLLERITH LABEL (20H)
C
      DIMENSION X(1),WHERE(2)
      PRINT 100,A,N,WHERE,A,(A,K,K=1,9)
      I1=1
      I2=MIN0(N,10)
   10 NP=IABS(NPL)
      IF(NPL.GT.0) PRINT 101,I1,NP,(X(I),I=I1,I2)
      IF(NPL.LT.0) PRINT 102,I1,NP,(X(I),I=I1,I2)
      I1=I1+10
      I2=MIN0(I1+9,N)
      IF(I1.LE.N) GO TO 10
      RETURN
  100 FORMAT(////13H PRINTOUT OF ,A6,8H(K),K=1,I4,5X2A10//
     1     5H     K,6XA6,3H(K),9(1XA6,3H(K+,I1,1H))/)
  101 FORMAT(I5,3X10F12.=)
  102 FORMAT(I5,3X10E12.=)
      END
```

```
      SUBROUTINE SPCTRM(XR,XI,NIN,NOUT,ALPHA,W)
C
C IN THIS SUBROUTINE WE COMPUTE THE POWER SPECTRUM OF THE COMPLEX TIME
C SEQUENCE IN THE ARRAY-PAIR (XR,XI) OF LENGTH NIN.  THE POWER SPECTRUM
C IS RETURNED IN ARRAY XR, AND IT IS NOW OF LENGTH NOUT.
C
C THE SAMPLE SPACING OF THE POWER SPECTRUM IS 1/NOUT OF THE REPETITION
C FREQUENCY.
C
C A COSINE-ON-A-PEDESTAL WEIGHTING IS APPLIED TO THE INPUT SAMPLES.
C ALPHA IS THE RATIO OF THE WEIGHTING FUNCTION AT THE EDGE TO THE
C CENTER.  ALPHA=.08 FOR HAMMING AND ALPHA=1.0 FOR UNIFORM WEIGHTING.
C THE USER CAN SUPPLY HIS OWN WEIGHTING FUNCTION IN ARRAY W OF LENGTH
C NIN, BUT HE MUST SET ALPHA TO A NEGATIVE VALUE.
C
C ARRAY W IS A WORKING ARRAY AND IT MUST BE DIMENSIONED AS LARGE AS.....
C
C               NIN             IF NOUT.EQ.2**INTEGER
C               NIN+2*NOUT      IF NOUT.NE.2**INTEGER
C
C THE WEIGHTS ARE NORMALIZED SO THAT THE SUM IS UNITY.
C
      DIMENSION XR(1),XI(1),W(1)
      DATA NINO/0/,NOUTO/0/
      IF(NIN.EQ.NINO.AND.NOUT.EQ.NOUTO) GO TO 10
      IF(ALPHA.GE.0.) CALL WEIGHT(W,NIN,ALPHA)
      NINO=NIN
      NOUTO=NOUT
   10 N=NOUT-NIN
      IF(N) 40,15,12
   12 CALL XMIT(-N,0.,XR(NIN+1))
      CALL XMIT(-N,0.,XI(NIN+1))
   15 CALL PROD(NIN,W,XR,XR)
      CALL PROD(NIN,W,XI,XI)
      CALL FFT2(XR,XI,NOUT,-1)
      CALL PCWR(NOUT,XR,XI,XR)
      RETURN
   40 PRINT 100,NIN,NOUT
      STOP
  100 FORMAT(/1X49H***ERROR IN SPECTRM, NOUT IS SMALLER THAN NIN ***/
     1     /10X4HNIN=I6,10X5HNOUT=I6)
      END
```

AD-A097 765 MARK RESOURCES INC MARINA DEL REY CA F/G 17/9
SIMULATING RADAR SIGNALS FOR DETECTION PERFORMANCE EVALUATION.(U)
FEB 81 R L MITCHELL F19628-79-C-0012
UNCLASSIFIED RADC-TR-80-410 NL

2 of 2

END
DATE
FILMED
8
DTIC

```
      SUBROUTINE WEIGHT(W,N,ALPHA)
C
C THIS SUBROUTINE COMPUTES COSINE-ON-A-PEDESTAL WEIGHTS IN ARRAY W OF
C LENGTH N (N IS ALSO THE NUMBER OF PULSES).  ALPHA IS THE RATIO OF THE
C WEIGHTING FUNCTION AT THE EDGE TO THE CENTER.  ALPHA=.08 FOR HAMMING
C AND 1.0 FOR UNIFORM WEIGHTING.  THE WEIGHTS ARE NORMALIZED SO THE SUM
C IS UNITY.
C
C IF THE WEIGHTS ARE TO BE USED AS PART OF A LONGER ARRAY OF LENGTH NR
C (WITH ZERO FILL), THE CALLING SEQUENCE IS.....
C
C           CALL XMIT(-NR,0.,W)
C           CALL WEIGHTS(W,N,ALPHA)
C
C TO CENTER THE WEIGHTS AT THE FIRST SAMPLE OF THE W-ARRAY (BEST DONE
C WHEN N IS ODD), FOLLOW THE ABOVE SEQUENCE WITH.....
C
C           CALL SHAFT(W,NR,-N/2)
C
      DIMENSION W(1)
      DATA TWOPI/6.2831853/
      A=(1.+ALPHA)/2.
      B=(1.-ALPHA)/2.
      CN=(N+1)/2.
      XN=N
      DO 20 K=1,N
      W(K)=A+B*COS(TWOPI*(K-CN)/XN)
   20 CONTINUE
      WNORM=1./ZSUM(N,W)
      CALL FROD(-N,WNORM,W,W)
      RETURN
      END
```

```
      FUNCTION DBX(A,N)
C COMPUTES DECIBEL VALUES.  THERE ARE TWO ENTRIES.....
C
C         D=DB(A)              D=10.*ALOG10(A)
C
C         CALL DBN(A,N)        A(K)=DB(A(K)), K=1,N
C
C THE OUTPUT IS TRUNCATED TO THE INTERVAL -99, +99 DB.
C
      DIMENSION A(1)
      DBZ(Z)=10.*ALOG10(AMAX1(AMIN1(7.943E9,Z),1.2589E-10))
      ENTRY DB
      DBX=DBZ(A(1))
      RETURN
      ENTRY DBN
      DO 20 I=1,N
      A(I)=DBZ(A(I))
   20 CONTINUE
      RETURN
      END
```

```
      SUBRCUTINE DSTGEN(A,B,NN)
C
C COMPUTES CUMULATIVE DISTRIBUTICN FUNCTION OF SAMPLE DATA
C
C TO INITIALIZE.....
C
C        CALL DSTINL(X1,XINC,N)           X1 = FIRST VALUE
C                                       XINC = VALUE INCREMENT
C                                          N = NUMBER OF INCREMENTS
C
C FOR EACH DATA POINT.....
C
C        CALL DSTPNT(X,W)                  X = DATA POINT VALUE
C                                          W = WEIGHT (USUALLY =1.)
C
C TO COMPUTE THE DISTRIBUTION FU  TION (AFTER LAST CALL TO DSTPNT).....
C
C        CALL DSTFUN(P)                    P = DISTRIBUTICN FUNCTION
C
C THE DISTRIBUTION FUNCTION IN ARRAY P IS.....
C
C      P(1) = PROB(DATA.LT.X1)
C      P(2) = PROB(DATA.LT.X1+XINC)
C             .
C             .
C             .
C      P(N) = PROB(DATA.LT.X1+(N-1)*XINC)
C
C ARRAY P IS DIMENSICNED FOR A MAX VALUE OF N=201
C
      DIMENSION A(1),F(202)
      ENTRY DSTINL
      N=NN
      IF(N.GT.201) STOP 22
      NS=0
      NP1=N+1
      X1=A(1)
      XINC=B
      CALL XMIT(-NP1,0.,F)
      RETURN
      ENTRY DSTFUN
      DO 10 L=2,NP1
      F(L)=F(L-1)+F(L)
   10 CONTINUE
      DO 20 L=1,N
      F(L)=F(NP1)-F(L)
      A(L)=1.-F(L)/FLOAT(NS)
   20 CONTINUE
C     PRINT 100,F(NP1)
  100 FORMAT(F16.6)
```

```
RETURN
ENTRY DSTPNT
NS=NS+1
L=(A(1)-X1)/XINC+2.
L=MAX0(L,1)
L=MIN0(L,NF1)
F(L)=F(L)+F
RETURN
END
```

```
      SUBRCUTINE FFT2(AR,AI,N,ISGN)
C
C THIS SUBROUTINE COMPUTES THE FFT IN THE ARRAY-PAIR (AR,AI) OF LENGTH N
C AND RETURNS THE RESULT IN THE SAME ARRAY PAIR.
C
C        ISGN = SIGN OF PHASE ARGUMENT IN FFT
C
C ARRAY W IN COMMON MUST BE DIMENSIONED AT LEAST AS LARGE AS 2*N IF
C N.NE.2**INTEGER.
C
      DIMENSION AR(1),AI(1)
      COMMON /TEMP/ W(1)
      I=1
      IF(ISGN.GT.0) I=0
      CALL FOURT(AR,AI,N,1,I,1,W(1),W(N+1))
      RETURN
      END
```

```
      SUBROUTINE FOURT(DATAR,DATAI,NN,NDIM,IFRWD,ICPLX,WORKR,WORKI)
      DIMENSION DATAR(1),DATAI(1),NN(1),WORKR(1),WORKI(1),IFACT(20)
C
C     THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C
C     EVALUATES COMPLEX FOURIER SERIES FOR COMPLEX OR REAL FUNCTIONS.
C     THAT IS, IT COMPUTES
C     FTRAN(J1,J2,...)=SUM(DATA(I1,I2,...)*W1**(I1-1)*(J1-1)
C                                *W2**(I2-1)*(J2-1)*...),
C     WHERE W1=EXP(-2*PI*SQRT(-1)/NN(1)), W2=EXP(-2*PI*SQRT(-1)/NN(2)),
C     ETC. AND I1 AND J1 RUN FROM 1 TO NN(1), I2 AND J2 RUN FROM 1 TO
C     NN(2), ETC.  THERE IS NO LIMIT ON THE DIMENSIONALITY (NUMBER OF
C     SUBSCRIPTS) OF THE ARRAY OF DATA.  THE PROGRAM WILL PERFORM
C     A THREE-DIMENSIONAL FOURIER TRANSFORM AS EASILY AS A ONE-DIMEN-
C     SIONAL ONE, THO IN A PROPORTIONATELY GREATER TIME.  AN INVERSE
C     TRANSFORM CAN BE PERFORMED, IN WHICH THE SIGN IN THE EXPONENTIALS
C     IS +, INSTEAD OF -.  IF AN INVERSE TRANSFORM IS PERFORMED UPON
C     AN ARRAY OF TRANSFORMED DATA, THE ORIGINAL DATA WILL REAPPEAR,
C     MULTIPLIED BY NN(1)*NN(2)*...  THE ARRAY OF INPUT DATA MAY BE
C     REAL OR COMPLEX, AT THE PROGRAMMERS OPTION, WITH A SAVING OF
C     ABOUT THIRTY PER CENT IN RUNNING TIME FOR REAL OVER COMPLEX.
C     (FOR FASTEST TRANSFORM OF REAL DATA, NN(1) SHOULD BE EVEN.)
C     THE TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED IN THE
C     ORIGINAL ARRAY OF DATA, REPLACING THE INPUT DATA.  THE LENGTH
C     OF EACH DIMENSION OF THE DATA ARRAY MAY BE ANY INTEGER.  THE
C     PROGRAM RUNS FASTER ON COMPOSITE INTEGERS THAN ON PRIMES, AND IS
C     PARTICULARLY FAST ON NUMBERS RICH IN FACTORS OF TWO.
C
C     TIMING IS IN FACT GIVEN BY THE FOLLOWING FORMULA.  LET NTOT BE THE
C     TOTAL NUMBER OF POINTS (REAL OR COMPLEX) IN THE DATA ARRAY, THAT
C     IS, NTOT=NN(1)*NN(2)*...  DECOMPOSE NTOT INTO ITS PRIME FACTORS,
C     SUCH AS 2**K2 * 3**K3 * 5**K5 * ...  LET SUM2 BE THE SUM OF ALL
C     THE FACTORS OF TWO IN NTOT, THAT IS, SUM2 = 2*K2.  LET SUMF BE
C     THE SUM OF ALL OTHER FACTORS OF NTOT, THAT IS, SUMF = 3*K3+5*K5+..
C     THE TIME TAKEN BY A MULTIDIMENSIONAL TRANSFORM ON THESE NTOT DATA
C     IS T = T0 + T1*NTOT + T2*NTOT*SUM2 + T3*NTOT*SUMF.  FOR THE PAR-
C     TICULAR IMPLEMENTATION FORTRAN 32 ON THE CDC 3300 (FLOATING POINT
C     ADD TIME = SIX MICROSECONDS),
C     T = 3000 + 600*NTOT + 50*NTOT*SUM2 + 175*NTOT*SUMF MICROSECONDS
C     ON COMPLEX DATA.
C
C     IMPLEMENTATION OF THE DEFINITION BY SUMMATION WILL RUN IN A TIME
C     PROPORTIONAL TO NTOT**2.  FOR HIGHLY COMPOSITE NTOT, THE SAVINGS
C     OFFERED BY COOLEY-TUKEY CAN BE DRAMATIC.  A MATRIX 100 BY 100 WILL
C     BE TRANSFORMED IN TIME PROPORTIONAL TO 10000*(2+2+2+2+5+5+5+5) =
C     280,000 (ASSUMING T2 AND T3 TO BE ROUGHLY COMPARABLE) VERSUS
C     10000**2 = 100,000,000 FOR THE STRAIGHTFORWARD TECHNIQUE.
C
C     THE COOLEY-TUKEY ALGORITHM PLACES TWO RESTRICTIONS UPON THE
C     NATURE OF THE DATA BEYOND THE USUAL RESTRICTION THAT
```

```
C       THE DATA FROM ONE CYCLE OF A PERIODIC FUNCTION.  THEY ARE--
C       1.  THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES
C       MUST BE THE SAME.
C       2.  CONSIDERING THE DATA TO BE IN THE TIME DOMAIN,
C       THEY MUST BE EQUI-SPACED AT INTERVALS OF DT.  FURTHER, THE TRANS-
C       FORM VALUES, CONSIDERED TO BE IN FREQUENCY SPACE, WILL BE EQUI-
C       SPACED FROM 0 TO 2*PI*(NN(I)-1)/(NN(I)*DT) AT INTERVALS OF
C       2*PI/(NN(I)*DT) FOR EACH DIMENSION OF LENGTH NN(I).  OF COURSE,
C       DT NEED NOT BE THE SAME FOR EVERY DIMENSION.
C
C       THE CALLING SEQUENCE IS--
C       CALL FOURT(DATAR,DATAI,NN,NDIM,IFRWD,ICPLX,WORKR,WORKI)
C
C       DATAR AND DATAI ARE THE ARRAYS USED TO HOLD THE REAL AND IMAGINARY
C       PARTS OF THE INPUT DATA ON INPUT AND THE TRANSFORM VALUES ON
C       OUTPUT.  THEY ARE FLOATING POINT ARRAYS, MULTIDIMENSIONAL WITH
C       IDENTICAL DIMENSIONALITY AND EXTENT.  THE EXTENT OF EACH DIMENSION
C       IS GIVEN IN THE INTEGER ARRAY NN, OF LENGTH NDIM.  THAT IS,
C       NDIM IS THE DIMENSIONALITY OF THE ARRAYS DATAR AND DATAI.
C       IFRWD IS AN INTEGER USED TO INDICATE THE DIRECTION OF THE FOURIER
C       TRANSFORM.  IT IS NON-ZERO TO INDICATE A FORWARD TRANSFORM
C       (EXPONENTIAL SIGN IS -) AND ZERO TO INDICATE AN INVERSE TRANSFORM
C       (SIGN IS +).  ICPLX IS AN INTEGER TO INDICATE WHETHER THE DATA
C       ARE REAL OR COMPLEX.  IT IS NON-ZERO FOR COMPLEX, ZERO FOR REAL.
C       IF IT IS ZERO (REAL) THE CONTENTS OF ARRAY DATAI WILL BE ASSUMED
C       TO BE ZERO, AND NEED NOT BE EXPLICITLY SET TO ZERO.  AS EXPLAINED
C       ABOVE, THE TRANSFORM RESULTS ARE ALWAYS COMPLEX AND ARE STORED
C       IN DATAR AND DATAI ON RETURN.  WORKR AND WORKI ARE ARRAYS USED
C       FOR WORKING STORAGE.  THEY ARE NOT NECESSARY IF ALL THE DIMENSIONS
C       OF THE DATA ARE POWERS OF TWO.  IN THIS CASE, THE ARRAYS MAY BE
C       REPLACED BY THE NUMBER 0 IN THE CALLING SEQUENCE.  THUS, USE OF
C       POWERS OF TWO CAN FREE A GOOD DEAL OF STORAGE.  IF ANY DIMENSION
C       IS NOT A POWER OF TWO, THESE ARRAYS MUST BE SUPPLIED.  THEY ARE
C       FLOATING POINT, ONE DIMENSIONAL OF LENGTH EQUAL TO THE LARGEST
C       ARRAY DIMENSION, THAT IS, TO THE LARGEST VALUE OF NN(I).
C       WORKR AND WORKI, IF SUPPLIED, MUST NOT BE THE SAME ARRAYS AS DATAR
C       OR DATAI.  ALL SUBSCRIPTS OF ALL ARRAYS BEGIN AT 1.
C
C       EXAMPLE 1.  THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A
C       COMPLEX ARRAY DIMENSIONED 100 BY 16 BY 13.
C       DIMENSION DATAR(100,16,13),DATAI(100,16,13),WORKR(100),WORKI(100)
C       DIMENSION NN(3)
C       NN(1)=100
C       NN(2)=16
C       NN(3)=13
C       CALL FOURT(DATAR,DATAI,NN,3,1,1,WORKR,WORKI)
C
C       EXAMPLE 2.  ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF
C       LENGTH 64.
C       DIMENSION DATAR(64),DATAI(64)
```

```
C     CALL FOURT(DATAR,DATAI,64,1,1,0,0,0)
C
C     THERE ARE NO ERROR MESSAGES OR ERROR HALTS IN THIS PROGRAM.  THE
C     PROGRAM RETURNS IMMEDIATELY IF NDIM OR ANY NN(I) IS LESS THAN ONE.
C
C     THE SINE AND COSINE VALUES REQUIRED FOR THE TRANSFORM ARE
C     GENERATED RECURSIVELY.  IF DOUBLE PRECISION IS AVAILABLE, IT IS
C     STRONGLY URGED THAT THE FOLLOWING VARIABLES BE SO DECLARED TO
C     REDUCE ACCUMULATION OF ROUNDOFF ERROR--
C     DOUBLE PRECISION TWOPI,THETA,WSTPR,WSTPI,WMINR,WMINI,WR,WI,WTEMP
C    *  ,THETM,WMSTR,WMSTI,TWOWR,SR,SI,OLDSR,OLDSI,STMPR,STMPI
C     IN ADDITION, TWOPI SHOULD BE ASSIGNED A SUFFICIENTLY PRECISE
C     VALUE AND THE VARIOUS CALLS TO THE FUNCTIONS COS AND SIN
C     SHOULD BE CHANGED TO DCOS AND DSIN.
C
C     PROGRAM BY NORMAN BRENNER FROM THE BASIC ALGORITHM BY CHARLES
C     RADER (BOTH OF MIT LINCOLN LABORATORY).  MAY 1967.  THE IDEA
C     FOR THE BIT REVERSAL WAS SUGGESTED BY RALPH ALTER (ALSO MIT LL).
C     ADAPTED FROM THE WORK OF JAMES W. COOLEY AND JOHN W. TUKEY,
C     AN ALGORITHM FOR THE MACHINE CALCULATION OF COMPLEX FOURIER
C     SERIES, MATH. COMPUT. 19, 90 (APRIL 1965), 297-301.
C
      IF(NDIM-1)920,1,1
1     NTOT=1
      DO 2 IDIM=1,NDIM
2     NTOT=NTOT*NN(IDIM)
      TWOPI=6.283185307
C
C     MAIN LOOP FOR EACH DIMENSION
C
      NP1=1
      DO 910 IDIM=1,NDIM
      N=NN(IDIM)
      NP2=NP1*N
      IF(N-1)920,900,5
C
C     IS N A POWER OF TWO AND IF NOT, WHAT ARE ITS FACTORS
C
5     M=N
      NTWO=NP1
      IF=1
      IDIV=2
10    IQUOT=M/IDIV
      IREM=M-IDIV*IQUOT
      IF(IQUOT-IDIV)50,11,11
11    IF(IREM)20,12,20
12    NTWO=NTWO+NTWO
      IFACT(IF)=IDIV
      IF=IF+1
      M=IQUOT
```

```
      GO TO 10
20    IDIV=3
      INON2=IF
30    IQUOT=M/IDIV
      IREM=M-IDIV*IQUOT
      IF(IQUOT-IDIV)60,31,31
31    IF(IREM)40,32,40
32    IFACT(IF)=IDIV
      IF=IF+1
      M=IQUOT
      GO TO 30
40    IDIV=IDIV+2
      GO TO 30
50    INON2=IF
      IF(IREM)60,51,60
51    NTWO=NTWO+NTWO
      GO TO 70
60    IFACT(IF)=M
70    NON2P=NP2/NTWO
C
C     SEPARATE FOUR CASES--
C        1. COMPLEX TRANSFORM
C        2. REAL TRANSFORM FOR THE 2ND, 3RD, ETC. DIMENSION.  METHOD--
C           TRANSFORM HALF THE DATA, SUPPLYING THE OTHER HALF BY CON-
C           JUGATE SYMMETRY.
C        3. REAL TRANSFORM FOR THE 1ST DIMENSION, N ODD.  METHOD--
C           SET THE IMAGINARY PARTS TO ZERO.
C        4. REAL TRANSFORM FOR THE 1ST DIMENSION, N EVEN.  METHOD--
C           TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 WHOSE REAL PARTS
C           ARE THE EVEN NUMBERED REAL VALUES AND WHOSE IMAGINARY PARTS
C           ARE THE ODD-NUMBERED REAL VALUES.  UNSCRAMBLE AND SUPPLY
C           THE SECOND HALF BY CONJUGATE SYMMETRY.
C
      ICASE=1
      IFMIN=1
      IF(ICPLX)100,71,100
71    ICASE=2
      IF(IDIM-1)72,72,100
72    ICASE=3
      IF(NTWO-NP1)100,100,73
73    ICASE=4
      IFMIN=2
      NTWO=NTWO/2
      N=N/2
      NP2=NP2/2
      NTOT=NTOT/2
      I=1
      DO 80 J=1,NTOT
      DATAR(J)=DATAR(I)
      DATAI(J)=DATAR(I+1)
```

```
80    I=I+2
C
C     SHUFFLE DATA BY BIT REVERSAL, SINCE N=2**K.  AS THE SHUFFLING
C     CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED
C
100   IF(NON2P-1)101,101,200
101   NP2HF=NP2/2
      J=1
      DO 150 I2=1,NP2,NP1
      IF(J-I2)121,130,130
121   I1MAX=I2+NP1-1
      DO 125 I1=I2,I1MAX
      DO 125 I3=I1,NTOT,NP2
      J3=J+I3-I2
      TEMPR=DATAR(I3)
      TEMPI=DATAI(I3)
      DATAR(I3)=DATAR(J3)
      DATAI(I3)=DATAI(J3)
      DATAR(J3)=TEMPR
125   DATAI(J3)=TEMPI
130   M=NP2HF
140   IF(J-M)150,150,141
141   J=J-M
      M=M/2
      IF(M-NP1)150,140,140
150   J=J+M
      GO TO 300
C
C     SHUFFLE DATA BY DIGIT REVERSAL FOR GENERAL N
C
200   DO 270 I1=1,NP1
      DO 270 I3=I1,NTOT,NP2
      J=I3
      DO 260 I=1,N
      IF(ICASE-3)210,220,210
210   WORKR(I)=DATAR(J)
      WORKI(I)=DATAI(J)
      GO TO 240
220   WORKR(I)=DATAR(J)
      WORKI(I)=0.
240   IFP2=NP2
      IF=IFMIN
250   IFP1=IFP2/IFACT(IF)
      J=J+IFP1
      IF(J-I3-IFP2)260,255,255
255   J=J-IFP2
      IFP2=IFP1
      IF=IF+1
      IF(IFP2-NP1)260,260,250
260   CONTINUE
```

```
      I2MAX=I3+NP2-NP1
      I=1
      DO 270 I2=I3,I2MAX,NP1
      DATAR(I2)=WORKR(I)
      DATAI(I2)=WORKI(I)
270   I=I+1
C
C     SPECIAL CASE--  W=1
C
300   I1RNG=NP1
      GO TO(302,301,302,302),ICASE
301   I1RNG=NP0*(1+NPREV/2)
302   IF(NTWO-NP1)600,600,303
303   DO 430 I1=1,I1RNG
      IMIN=NP1+I1
      ISTEP=2*NP1
      GO TO 330
310   J=I1
      DO 320 I=IMIN,NTOT,ISTEP
      TEMPR=DATAR(I)
      TEMPI=DATAI(I)
      DATAR(I)=DATAR(J)-TEMPR
      DATAI(I)=DATAI(J)-TEMPI
      DATAR(J)=DATAR(J)+TEMPR
      DATAI(J)=DATAI(J)+TEMPI
320   J=J+ISTEP
      IMIN=IMIN+IMIN-I1
      ISTEP=ISTEP+ISTEP
330   IF(ISTEP-NTWO)310,310,331
C
C     SPECIAL CASE--  W=-SQRT(-1)
C
331   IMIN=3*NP1+I1
      ISTEP=4*NP1
      GO TO 420
400   J=IMIN-ISTEP/2
      DO 410 I=IMIN,NTOT,ISTEP
      IF(IFFWD)401,402,401
401   TEMPR=DATAI(I)
      TEMPI=-DATAR(I)
      GO TO 403
402   TEMPR=-DATAI(I)
      TEMPI=DATAR(I)
403   DATAR(I)=DATAR(J)-TEMPR
      DATAI(I)=DATAI(J)-TEMPI
      DATAR(J)=DATAR(J)+TEMPR
      DATAI(J)=DATAI(J)+TEMPI
410   J=J+ISTEP
      IMIN=IMIN+IMIN-I1
      ISTEP=ISTEP+ISTEP
```

```
420   IF(ISTEP-NTWO)400,400,430
430   CONTINUE
C
C     MAIN LOOP FOR FACTORS OF TWO.  W=EXP(-2*PI*SQRT(-1)*M/MMAX)
C
      THETA=-TWOPI/8.
      WSTPR=0.
      WSTPI=-1.
      IF(IFRWD)502,501,502
501   THETA=-THETA
      WSTPI=1.
502   MMAX=8*NP1
      GO TO 540
500   WMINR=COS(THETA)
      WMINI=SIN(THETA)
      WR=WMINR
      WI=WMINI
      MMIN=MMAX/2+NP1
      MSTEP=NP1+NP1
      DO 530 M=MMIN,MMAX,MSTEP
      DO 525 I1=1,I1RNG
      ISTEP=MMAX
      IMIN=M+I1
510   J=IMIN-ISTEP/2
      DO 520 I=IMIN,NTOT,ISTEP
      TEMPR=DATAR(I)*WR-DATAI(I)*WI
      TEMPI=DATAR(I)*WI+DATAI(I)*WR
      DATAR(I)=DATAR(J)-TEMPR
      DATAI(I)=DATAI(J)-TEMPI
      DATAR(J)=DATAR(J)+TEMPR
      DATAI(J)=DATAI(J)+TEMPI
520   J=J+ISTEP
      IMIN=IMIN+IMIN-I1
      ISTEP=ISTEP+ISTEP
      IF(ISTEP-NTWO)510,510,525
525   CONTINUE
      WTEMP=WR*WSTPI
      WR=WR*WSTPR-WI*WSTPI
530   WI=WI*WSTPR+WTEMP
      WSTPR=WMINR
      WSTPI=WMINI
      THETA=THETA/2.
      MMAX=MMAX+MMAX
540   IF(MMAX-NTWO)500,500,600
C
C     MAIN LOOP FOR FACTORS NOT EQUAL TO TWO.
C     W=EXP(-2*PI*SQRT(-1)*(J2-I3)/IFP2)
C
600   IF(NON2P-1)700,700,601
601   IFP1=NTWO
```

```
      IF=INON2
610   IFP2=IFACT(IF)*IFP1
      THETA=-TWOPI/FLOAT(IFACT(IF))
      IF(IFRWD)612,611,612
611   THETA=-THETA
612   THETM=THETA/FLOAT(IFP1/NP1)
      WSTPR=COS(THETA)
      WSTPI=SIN(THETA)
      WMSTR=COS(THETM)
      WMSTI=SIN(THETM)
      WMINR=1.
      WMINI=0.
      DO 660 J1=1,IFP1,NP1
      I1MAX=J1+I1RNG-1
      DO 650 I1=J1,I1MAX
      DO 650 I3=I1,NTOT,NP2
      I=1
      WR=WMINR
      WI=WMINI
      J2MAX=I3+IFP2-IFP1
      DO 640 J2=I3,J2MAX,IFP1
      TWOWR=WR+WR
      JMIN=I3
      J3MAX=J2+NP2-IFP2
      DO 630 J3=J2,J3MAX,IFP2
      J=JMIN+IFP2-IFP1
      SR=DATAR(J)
      SI=DATAI(J)
      OLDSR=0.
      OLDSI=0.
      J=J-IFP1
620   STMPR=SR
      STMPI=SI
      SR=TWOWR*SR-OLDSR+DATAR(J)
      SI=TWOWR*SI-OLDSI+DATAI(J)
      OLDSR=STMPR
      OLDSI=STMPI
      J=J-IFP1
      IF(J-JMIN)621,621,620
621   WORKR(I)=WR*SR-WI*SI-OLDSR+DATAR(J)
      WORKI(I)=WI*SR+WR*SI-OLDSI+DATAI(J)
      JMIN=JMIN+IFP2
630   I=I+1
      WTEMP=WR*WSTPI
      WR=WR*WSTPR-WI*WSTPI
640   WI=WI*WSTPR+WTEMP
      I=1
      DO 650 J2=I3,J2MAX,IFP1
      J3MAX=J2+NP2-IFP2
      DO 650 J3=J2,J3MAX,IFP2
```

```
      DATAR(J3)=WORKR(I)
      DATAI(J3)=WORKI(I)
650   I=I+1
      WTEMP=WMINR*WMSTI
      WMINR=WMINR*WMSTR-WMINI*WMSTI
660   WMINI=WMINI*WMSTR+WTEMP
      IF=IF+1
      IFP1=IFP2
      IF(IFP1-NP2)610,700,700
C
C     COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON-
C     JUGATE SYMMETRIES.
C
700   GO TO (500,800,900,701),ICASE
701   NHALF=N
      N=N+N
      THETA=-TWOPI/FLOAT(N)
      IF(IFRWD)703,702,703
702   THETA=-THETA
703   WSTPR=COS(THETA)
      WSTPI=SIN(THETA)
      WR=WSTPR
      WI=WSTPI
      IMIN=2
      JMIN=NHALF
      GO TO 725
710   J=JMIN
      DO 720 I=IMIN,NTOT,NP2
      SUMR=(DATAR(I)+DATAR(J))/2.
      SUMI=(DATAI(I)+DATAI(J))/2.
      DIFR=(DATAR(I)-DATAR(J))/2.
      DIFI=(DATAI(I)-DATAI(J))/2.
      TEMPR=WR*SUMI+WI*DIFR
      TEMPI=WI*SUMI-WR*DIFR
      DATAR(I)=SUMR+TEMPR
      DATAI(I)=DIFI+TEMPI
      DATAR(J)=SUMR-TEMPR
      DATAI(J)=-DIFI+TEMPI
720   J=J+NP2
      IMIN=IMIN+1
      JMIN=JMIN-1
      WTEMP=WR*WSTPI
      WR=WR*WSTPR-WI*WSTPI
      WI=WI*WSTPR+WTEMP
725   IF(IMIN-JMIN)710,730,740
730   IF(IFRWD)731,740,731
731   DO 735 I=IMIN,NTOT,NP2
735   DATAI(I)=-DATAI(I)
740   NP2=NP2+NP2
      NTOT=NTOT+NTOT
```

```
      J=NTOT+1
      IMAX=NTOT/2+1
  745 IMIN=IMAX-NHALF
      I=IMIN
      GO TO 755
  750 DATAR(J)=DATAR(I)
      DATAI(J)=-DATAI(I)
  755 I=I+1
      J=J-1
      IF(I-IMAX)750,760,760
  760 DATAR(J)=DATAR(IMIN)-DATAI(IMIN)
      DATAI(J)=0.
      IF(I-J)770,780,780
  765 DATAR(J)=DATAR(I)
      DATAI(J)=DATAI(I)
  770 I=I-1
      J=J-1
      IF(I-IMIN)775,775,765
  775 DATAR(J)=DATAR(IMIN)+DATAI(IMIN)
      DATAI(J)=0.
      IMAX=IMIN
      GO TO 745
  780 DATAR(1)=DATAR(1)+DATAI(1)
      DATAI(1)=0.
      GO TO 900
C
C     COMPLETE A REAL TRANSFORM FOR THE 2ND, 3RD, ETC. DIMENSION BY
C     CONJUGATE SYMMETRIES.
C
  800 IF(NPREV-2)900,900,805
  805 DO 860 I3=1,NTOT,NP2
      I2MAX=I3+NP2-NP1
      DO 860 I2=I3,I2MAX,NP1
      IMAX=I2+NP1-1
      IMIN=I2+I1RNG
      JMAX=I3+I3+NP1-IMIN
      IF(I2-I3)820,820,810
  810 JMAX=JMAX+NP2
  820 IF(IDIM-2)850,850,830
  830 J=JMAX+NP0
      DO 840 I=IMIN,IMAX
      DATAR(I)=DATAR(J)
      DATAI(I)=-DATAI(J)
  840 J=J-1
  850 J=JMAX
      DO 860 I=IMIN,IMAX,NP0
      DATAR(I)=DATAR(J)
      DATAI(I)=-DATAI(J)
  860 J=J-NP0
C
```

```
C     END OF LOOP ON EACH DIMENSION
C
900   NP0=NP1
      NP1=NP2
910   NPREV=N
920   RETURN
      END
```

DATE
FILMED

8