

A PC-BASED RADAR DESIGN PACKAGE

Cyril Zaki Harari

A Project Report submitted to the Faculty of Engineering,
University of the Witwatersrand, Johannesburg, in partial
fulfillment of the requirements for the degree of Master of
Science in Engineering.

Johannesburg, 1987

DOCUMENT :

A Project Report submitted in partial-
fulfilment of the requirements for the
Degree of Masters of Science in
Electrical Engineering

PROJECT TITLE :

A PC-BASED RADAR DESIGN PACKAGE

SUBMITTED TO :

Assistant Dean (Post Graduate Affairs)

PROJECT SUPERVISOR :

Mr Mike Archer (Electrical Engineering
Department)

DONE BY :

Cyril Zaki Harari
Student Number: 81-0362/4

APPROVED BY :

.....

REMARKS :

.....
.....
.....

DECLARATION

I declare that this thesis is my own unaided work. It is being submitted for the Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

C. Z. Harari

(Cyril Zaki Harari)

2nd day of *October* 198*7*

ABSTRACT

This report deals with the development of a software package for evaluating the performance of a radar for single and multiple return pulse integration and fluctuating and non-fluctuating targets, as defined by Blake [1 and 2] and Hovanessian [3].

The software provides the user with a Personal Computer based package allowing him to calculate :

- i. the signal-to-noise ratio required, to give a specified target detection probability for a specified false alarm probability, and number of integrated pulses.
- ii. the effective target cross-section for various fluctuation models.
- iii. the detection performance versus range of a given radar using i and ii.

DEDICATION

To my mother, who for 24 years
inspired, encouraged, fed, argued
and in all ways lived with me. To
this person I dedicate this thesis.

ACKNOWLEDGMENTS

I wish to thank Mike Archer, lecturer at the University of the Witwatersrand, for his guidance into the radar field and for the many hours of challenging discussion, encouragement and support which he gave me.

Cyril Zaki Harari

CONTENTS	Page
DECLARATION.....	ii
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
CONTENTS.....	vi+vii
LIST OF FIGURES.....	viii
LIST OF TABLES.....	ix
LIST OF SYMBOLS.....	x
1. INTRODUCTION.....	1
2. BACKGROUND AND SELECTED TOPICS.....	4
2.1. The Radar Equation.....	4
2.2. Detection in the Radar Receiver.....	6
2.3. Probability of Detection and Signal to noise ratio.....	7
2.4. Radar Target Cross Section.....	7
2.5. Coverage Diagrams.....	8
3. FORMULATION OF THE PROBLEM.....	10
3.1. Calculation of the Radar Range.....	11
3.2. Integration Package.....	11
3.2.1. The Rectangular Formula.....	11
3.2.2. The Trapezoidal Formula.....	12
3.2.3. Simpson's Formula (Parabolic formula) for n even.....	12
3.2.4. The Runge Kutta Method.....	12
3.3. Bessel Function Evaluation.....	13
3.4. Probability of detection.....	14
3.4.1. Single Pulse Detection.....	14
3.4.2. Detection with Many Pulses Integrated.....	22
3.5. Target Fluctuation Effects.....	27
3.6. Coverage Diagram.....	29
4. SOLUTION TO THE PROBLEM.....	31
4.1. Software Development.....	31
4.1.1. Flow Chart Description.....	31
4.1.1.1. Calculation of the Radar Range.....	31
4.1.1.1.2. Pd, Pfa Curves for Single Pulse Integration.....	33

CONTENTS (Continued)	Page	
4.1.1.3.	P_d , P_{fa} Curves for Multiple Pulse Integration.....	34
4.1.1.4.	Curve Plot Procedure for Tables Generated.....	34
4.1.1.5.	Coverage Diagram.....	35
5.	PROBLEMS ENCOUNTERED AND THEIR SOLUTIONS.....	36
6.	RESULTS.....	39
7.	CONCLUSION.....	40
 APPENDICES		
APPENDIX A	A graph (taken from Blake[2]) showing the required signal to noise ratio at the input terminals of a linear rectifier detector as a function of probability of detection for a <u>single pulse</u> , with the false alarm probability (P_{fa}) as a parameter, calculated for a non-fluctuating signal.....	41
APPENDIX B	Substitution of equation (10) into (9).....	42
APPENDIX C	Substitution of equation (10) to obtain the value for v_{av}	43
APPENDIX D	Replacing v with $u = v/v_{av}$ and substituting the formula into equation (11).....	45
APPENDIX E	Source listing of pascal programs.....	46
APPENDIX F	Probability of detection versus signal to noise ratio (Hovanessian [3]).....	47
APPENDIX G	Examples of graphs plotted for varying S/N and probability of false alarm for Single pulse integration.....	48
APPENDIX H	A typical coverage diagram showing the radar vertical coverage using a cosecant-squared beam.....	49
APPENDIX I	Example of a graph Blake [1 and 2] of the required signal-to noise ratio for a linear detector as a function of number of pulses integrated, for 0.5 probability of detection calculated for a non fluctuating signal and five values of false-alarm probability (P_{fa}).....	50
APPENDIX J	Flow Chart of the software package.....	51
REFERENCES.....		52

LIST OF FIGURES

Figure	Page
1. Block diagram of a radar employing a power amplifier transmitter and a superheterodyne receiver.....	2
2. Radar receiver output as a function of time.....	6
3. Simple Flow Chart of the Problem.....	10
4. Signal processing circuitry.....	18

LIST OF TABLES

Table		Page
1.	Values of Single-Pulse Normalized Threshold Voltage for a set of False Alarm Probabilities.....	17
2.	Values of Inverse Error Function, k_{pfa} for Some Standard Value of False Alarm Probability.....	25
3.	Values of Inverse Error Function, k_{pd} for Some Standard Values of Probability of Detection, P_d	27

LIST OF SYMBOLS

- R_{\max} = maximum radar range.
- P_t = transmitted power, Watts.
- G_t = transmitting antenna gain, dB.
- G_r = receiving antenna gain, dB.
- Σ = target radar cross section, m^2
- λ = wavelength, metres.
- S_{\min} = minimum signal to noise ratio required for a specific P_{fa} and P_d .
- F_t = pattern propagation factor, transmitting antenna to target path.
- F_r = pattern propagation factor, target to receiving antenna path.
- k = Boltzmann's constant = 1.38×10^{-23} J.K⁻¹.
- T = equivalent noise temperature (290°K).
- B = receiver bandwidth, Hertz.
- L = System Losses (both transmit and receive).
- S = (S/N) = signal-to-noise ratio.
- $I_0(x)$ = Modified Bessel function of order zero.
- v = the receiver output voltage.
- P_d = the probability of detection.
- $v_t = E_t$ = fixed threshold voltage level.
- P_{sn} = probability density function of v .
- P_{fa} = the probability of false alarm.
- $P_n(v)$ = probability density of the detector output voltage when the input is noise.
- σ^2 = the noise power at the detector input.

1. INTRODUCTION

Radar is a prime example of the use of electromagnetic waves for remote sensing. Its development has been stimulated primarily by military needs for surveillance, navigation, and weapon control, but it is widely used in many important civilian applications.

Modern commercial air travel relies heavily on radar navigation on board aircraft and for air-traffic control on the ground. Radar is found on almost all the commercial shipping and on many pleasure boats.

As a remote sensor, it responds to targets as close as tens of feet, (as in police speedometers), to ranges of hundreds of millions of miles, as in large radars used by astronomers to probe the planets.

Radar is simple in concept even though in most instances its practical implementation is not. It operates by radiating electromagnetic energy and detecting the presence and character of the echo return from reflecting objects.

As an active device it utilizes its own controlled illumination to detect the target and to probe the target characteristics. It does not depend on energy radiated by the target itself, as does the radiometer, or on the energy reflected from uncontrolled sources, as does the optical camera.

The ability to detect a target at great distances and to locate its position with relatively high accuracy are the two chief attributes of radar. Although radar technology has been advanced primarily by the military, benefits have spilled over into many important civilian applications, of which a principal example is the navigation and control of ships and aircraft.

The radar principle has been applied from frequencies of a few Megahertz (MF) upto ultraviolet (laser radar). This represents a frequency ratio of about $10^9:1$. The particular techniques used in implementing the radar concept differ markedly over this wide range of frequencies, but the basic principle remains the same.

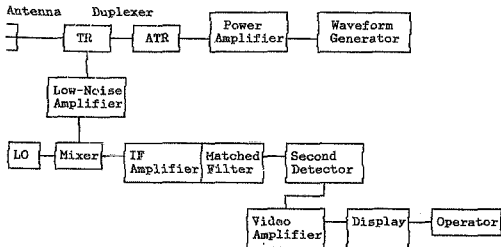


Figure 1 Block diagram of a radar employing a power-amplifier transmitter and a superheterodyne receiver.

The figure shown above illustrates the block diagram of one form of simple radar. The radar signal, usually in the form of a repetitive train of short pulses, is generated by a transmitter and radiated into space by an antenna.

The duplexer permits a single antenna to be used for both transmission and reception. Reflecting objects, or targets, intercept and reradiate a portion of the radar signal; a small amount returns in the direction of the radar.

Some of the returned signal, or echo, is collected by the antenna and detected in the receiver. The presence of reflected energy reveals the presence of a target.

Comparison of the returned signal with that transmitted yields information about the target, such as its location, size, shape, and whether it is in motion relative to the radar. In most radars the extracted information is displayed on a cathode-ray-tube where it is viewed by an operator. In some applications, for example weapon control, the radar output may bypass the operator and actuate a control system without human intervention.

The form of the electromagnetic signal radiated by the radar depends on the desired information that is wanted about the target. A pulse radar for aircraft surveillance, for instance, may generate a repetitive train of short pulses, each a few microseconds in width, at a repetition rate of several hundred per second.

For accurate range measurement short pulses, which occupy a small volume of space, or longer 'chirped' pulses which occupy a wide spectral bandwidth are used.

To sense accurately the doppler frequency shift introduced in the reflected signal by a moving target, the signal waveform must be long in duration. An example is the continuous wave (CW) doppler radar.

In addition to the widely used pulse train, there are many other possible waveforms, including CW with frequency modulation, pulses with frequency or phase modulation, and bursts of pulses.

Theory prescribes the optimum waveform for a particular set of constraints on measurement accuracy, ambiguity, resolution, and clutter discrimination.

2.1. The Radar Equation

Radars are electromagnetic devices used for detection of targets by radiating electromagnetic energy and examining the reflected energy. The term 'target' will refer to any reflecting object which interferes with the transmitted wave and reflects part of its energy. The definition deviates from the common notion of targets as hostile aircraft or missiles.

It will be seen that a comparison of the properties of the transmitted energy with that of the reflected energy from a target will result in parameters relating the relative radar target positions. The frequency shift between transmitted and reflected wave is proportional to radar target closing rate and the delay between transmitted and received electromagnetic energy is proportional to the distance between the radar and the target.

The detection range of a radar system is primarily a function of three parameters: 1) transmitted power, 2) antenna gain, and 3) receiver sensitivity. Increasing the transmitted power will increase the radiated energy which, in turn will result in a stronger target return. The antenna gain is a measure of the radiated energy in the direction of the target as compared to uniform radiation of energy. Receiver sensitivity is a measure of the capability of the receiver in detecting target returns.

The form of the radar equation shown below expresses the maximum radar range R_{max} in terms of radar and target parameters:

$$R_{max} = \left[\frac{P_t G_t G_r \Sigma \text{Lambda}^2}{(4\pi)^3 S_{min} F_t F_r (kTB) L} \right]^{1/4}$$

[EQUATION 1]

where P_t = transmitted power, Watts
 P_{fa} = probability of false alarm
 P_d = probability of detection
 G_t = transmitting antenna gain, dB
 G_r = receiving antenna gain, dB
 Σ = radar target cross section, m²
 Lambda = operating wavelength, metres
 S_{min} = minimum signal to noise ratio for a given P_{fa} and P_d
 F_t = pattern propagation factor, transmitting antenna to target path
 F_r = pattern propagation factor, target to receiving antenna path
 k = Boltzmann's constant = 1.38×10^{-23} J/°K
 T = equivalent noise temperature (290°K)
 B = receiver bandwidth, Hertz
 L = System Losses (both transmit and receive)

All the parameters are to some extent under the control of the radar designer, except for the target cross section 'Σ'. The radar equation states that if long range detection is required then

- 1) the transmitted power must be large,
- 2) the radiated energy must be concentrated into a narrow beam (high transmitting antenna gain),
- 3) the received echo energy must be collected with a large antenna aperture (also synonymous with high gain),
- 4) and the receiver must be sensitive to weak signals.

In practice, the radar equation does not predict the range performance of actual radar equipment to a satisfactory degree of accuracy. The predicted values of radar range are usually optimistic.

In some cases the particular radar detection range might can be half of that predicted by Equation 1.

Part of this discrepancy is due to the failure of Equation 1 to explicitly include the various losses that occur throughout the system or the loss in performance usually experienced when electronic equipment is operated in the field rather than under laboratory-type conditions.

Another important factor that must be considered in the radar equation is the statistical or unpredictable nature of several of the parameters.

The minimum detectable signal ' S_{min} ' and the target cross section ' Σ ' are both statistical in nature and are expressed in statistical terms.

2.2. Detection in the Radar Receiver

The ability of a radar receiver to detect a weak echo signal is limited by the noise energy that occupies the same portion of the frequency spectrum as does the signal energy. The weakest signal the receiver can detect is called the minimum detectable signal (MDS).

The specification of the minimum detectable signal is sometimes difficult because of its statistical nature and because the criterion for deciding whether a target is present or not, may not be too well defined.

Detection is based on establishing a threshold level at the output of the receiver. If the receiver output exceeds the threshold, a signal is assumed to be present. This is called threshold detection.

In the detection circuitry of the radar, the target return power is processed in the presence of receiver noise. The figure shown below is a qualitative description of target detection in the presence of receiver noise.

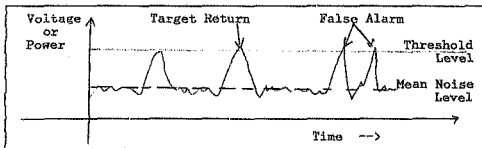


Figure 2 Radar receiver output as a function of time

A target is said to be detected if the envelope crosses the set threshold. The threshold is set so as to result in an acceptable number of false alarm (threshold crossings due to noise per given time, for example, five per minute).

Too low a threshold increases the likelihood that noise alone will rise above the threshold and be taken for a real signal. Such an occurrence is called a false alarm. Thus, the settings of thresholds is a trade-off between the desired false alarm rate and the detection capability.

Increasing the threshold level will, of course, decrease the number of false alarms, but it will also decrease target detection capability.

In defining false alarm rates and threshold settings, the type of displays, the required display density, and human factors such as the experience of operators can also be considered.

2.3. Probability of Detection and Signal To Noise Ratio

The probability of target detection depends on the value of target echo signal-to-noise ratio (S/N) in the receiver and the threshold setting in the receiver detection circuitry. High values of S/N correspond to high values of target return power. Higher thresholds raise the level of power at which target returns will be detected thereby degrading detection capability.

On the other hand higher detection thresholds reduce the rate of false alarms (noise crossing threshold levels) making detection of strong echo targets on the display easier.

Given the equations defining probability of detection as a function of signal-to-noise-ratio with probability of false alarm as a variable, the designer can calculate the range performance of a given radar for a given target using Equation 1.

2.4. Radar Target Cross Section

The discussion of the minimum signal-to-noise ratio assumes that the echo signal received from a particular target does not vary with time. In practice, however, the echo signal from a target in motion is never constant.

The cross sections of complex targets (the usual type of radar target) are very sensitive to target aspect. Therefore, as the target aspect changes relative to the radar, variations in the echo signal will result.

A method of accounting for a fluctuating cross section in the radar equation is to select a lower bound, that is, a value of

cross section that is exceeded for some specified (large) fraction of time. For all practical purposes the value selected is a minimum and the target will always present a cross section greater than that selected. This procedure results in a conservative prediction of radar range and has the advantage of simplicity.

To properly account for target cross-section fluctuations, the probability density function and the correlation properties with time must be known for the particular target and type of trajectory.

Curves of radar cross section as a function of aspect angle and a knowledge of trajectory with respect to the radar are needed to obtain an exact value for the dynamic radar cross section of a target at any time.

Thus we define the probability-density function, which gives the probability of finding any particular value of target cross section between the values of Σ and $\Sigma + d\Sigma$, whilst the auto-correlation function describes the degree of correlation of the cross section with time or number of pulses.

2.5. Coverage Diagrams

The vertical coverage diagram is typically a range-height-angle chart.

Plots of the radar range as a function of elevation angle can be made on such charts.

Such plots are called coverage diagrams.

In general, the curvature of the earth cannot be neglected when predicting radar coverage. This is especially true for coverage at low elevation angles near the horizon. The two regions of interest in radar propagation are the interference region and the diffraction region.

The interference, or optical, region is located within line of sight of the radar. The direct and reflected waves interfere to produce a lobed radiation pattern similar to that for a plane earth.

Lobing, however, is not as pronounced in the case of the round earth; the minima are not as deep nor are the maxima as great since a wave reflected from a curved surface is more divergent than one reflected from a plane.

The other region of interest is that which lies just beyond the interference region below the radar line of sight and is the diffraction, or the shadow, region.

Here radar signals are rapidly attenuated. Very few microwave radars have the capability of penetrating the diffraction region to any extent because of the severe losses.

The effect of the round earth on radar coverage can be predicted by analytical means for the idealized case of a 'smooth' earth of known, uniform properties.

There exists in the literature the necessary graphs and nomographs which simplify the computation of the radar coverage.

3. FORMULATION OF THE PROBLEM

A personal computer based radar design package was to be developed which allows the designer to input the known parameters with regards to a particular radar.

Information such as the radar transmitting power, the transmitting and receiving antenna gains, the target radar cross section, the pattern propagation factor (PPF) for the transmitting antenna-to-target-path and the PPF for the target-to-receiving antenna path, the receiver bandwidth, the generalised system losses, the wavelength and the signal to noise ratio are typical input parameters.

Those that can typically be calculated (if not available) are the probability of detection, and minimum signal to noise ratio given the probability of false alarm and also the losses due to target cross section fluctuation.

A simple flow chart of the problem is given below.

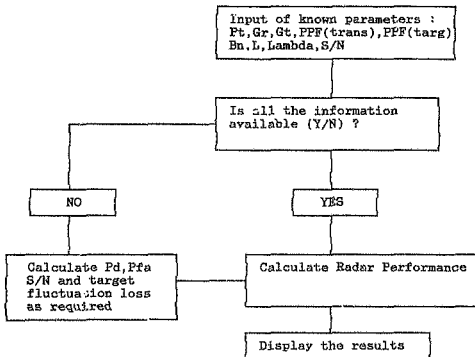


Figure 3 Simple Flow Chart of the Problem

3.1. Calculation of the Radar Range

The derivation and the formulation of the radar equation as described in section 2.1 was done over a long period of time. The particular equation which is used in the software package which was developed is described later.

(A complete and detailed discussion of all those factors that influence the prediction of the radar range is beyond the scope of this thesis. For this reason many subjects will be treated only lightly. More detailed information will be found in subsequent sections or in the references listed at the end of this thesis.)

The development of the software for evaluation of the radar range was very quick. All that was needed was a definition of the variables to be used, prompting of the user for the necessary inputs, evaluating Equation 1 and then displaying the results obtained. Note that here it was assumed that S_{min} and Σ were easily evaluated or available to the user. Routines were developed later allowing the user to evaluate S_{min} and Σ using techniques described by Blake [1 and 2] and Hovanessian [3].

3.2. Integration Package

The development of a numerical integration package was clearly necessary. Simpson's rule was to be considered. The Rectangular, Trapezoidal and Runge-Kutta rules were also considered so that the most efficient and optimum method of numerical integration could be determined.

In the rules given below, the interval from $x = a$ to $x = b$ is subdivided into n equal parts by the points $a = x_0, x_1, x_2, \dots, x_{n-1}, x_n = b$ and we let $y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n), h = (b-a)/n$.

3.2.1. The Rectangular Formula

$$\int_b^a f(x) dx \approx h(y_0 + y_1 + y_2 + \dots + y_{n-1})$$

[EQUATION 2]

The error in the rectangular rule is of the order h , i.e.
 $E = O(h)$.

3.2.2. The Trapezoidal Formula

$$\int_b^a f(x) dx \approx h/2(y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n)$$

[EQUATION 3]

The error in the trapezoidal rule is of the order h^2 , i.e.
 $E = O(h^2)$.

3.2.3. Simpson's Formula (Parabolic formula) for n even

$$\int_b^a f(x) dx \approx h/3(y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + 2y_{n-2} + 4y_{n-1} + y_n)$$

[EQUATION 4]

The error in Simpson's rule is of the order h^4 , i.e.
 $E = O(h^4)$. This implies that it integrates cubics exactly.

3.2.4. The Runge Kutta Method

This method is actually the four point method, often called the Runge Kutta method.

It is performed as follows:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$k_3 = hf(x_n + h/2, y_n + k_2/2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

} 4 points

$$\int_b^a f(x) dx \quad \equiv \text{is approximated and given as follows:}$$

$$y_{n+1} = y_n \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

[EQUATION 5]

The local truncation error has order 6 i.e. it behaves like $O(h^6)$, and the global truncation error behaves like $O(h^4)$.

Each of the equations for the approximation of definite integrals was implemented and the results obtained were very favourable.

The Runge Kutta technique clearly proved the best method. It allows the step length h to be varied as it is increased between the lower and upper limits of the integral to be evaluated.

It was therefore decided that this technique would be the one used for the numerical integration of the probability of detection function.

3.3. Bessel Function Evaluation

The need to evaluate the Modified Bessel function arose because the equations for the determination of the probability of detection for a radar used the Bessel function.

The Modified Bessel function of the first kind of order n is expressed as follows:

$I_n(x) = i^{-n} J(ix)$, which can also be expressed as an infinite series.

For $n = 0$, we have

$$I_0(x) = 1 + \frac{x^2}{2^2} + \frac{x^4}{2^2 \cdot 4^2} + \frac{x^6}{2^2 \cdot 4^2 \cdot 6^2} + \dots$$

[EQUATION 6]

The evaluation of the infinite series when implemented was taken to an accuracy for values up to x^{26} (where x ranges from zero to ten). However, it was decided to use the integral representation of the Bessel function (see below) which allows the Bessel function to be accurately calculated at any value of x and not be limited in the range from zero to ten.

The integral representation is as follows:

$$I_{\rho}(x) = (1/2\pi) \int_0^{2\pi} e^{x \sin \beta} d\beta$$

[EQUATION 7]

3.4. Probability of detection

A large quantity of experimental and theoretical work has been carried out by several persons in this particular field. It is here that most of this report is concentrated on.

The main sources of background information obtained was that of Blake and also Hovanessian. Most of the work done by Blake dates back to the Second World War and papers on this topic of research were finally released in 1969.

An outline of Blake's equations and methods used to obtain the graph included as Appendix A is given below. The main aim of this report was to prove the validity of these curves and to try to reproduce them to a reasonable accuracy. The method used by Hovanessian will be discussed later.

3.4.1. Single Pulse Detection

Equations as derived by Blake

The curve shown in Appendix A was machine computed by Blake (and machine plotted), using mathematical results derived by others.

The probability of detection is formally given by the expression

$$P_d = \int_{v_t}^{\infty} P_{SN}(v) dv,$$

[EQUATION 8]

where v is the receiver output voltage (after all predecision processing, including integration or averaging of successive pulses in the case of pulse radar, doppler filtering in the case of CW or pulse doppler radar). The quantity v_t is a fixed threshold level of voltage that depends on the allowable rate of

'false alarms' (decision that a signal is present when in fact there is no signal). The function P_{SN} is the probability density of v , which is a random variable because it is the resultant receiver output when both signal (s) and noise (n) are present in the output.

The probability of false alarm is

$$P_{fa} = \int_{v_t}^{\infty} P_N(v) dv,$$

[EQUATION 9]

where $P_N(v)$ is the probability density of the detector output voltage when the input is noise, with no signal present. The function p_{SN} must of course reduce to p_N when the signal power has zero value.

The value of P_d , equation (8) obviously depends on the value selected for v_t ; this value is chosen, as is evident from equation (9), by first deciding what value of P_{fa} is desired, or required, and then solving equation (9) for the value of v_t that results in this value of P_{fa} .

Of course, this solution can be obtained analytically only if $p_N(v)$ is an integrable function.

The graph included as Appendix A gives values of P_d as a function of the signal to noise power ratio S/N , for detection of a single pulse (no integration), and a linear rectifier detector (sometimes referred to as an envelope detector). The equations for $P_N(v)$ and $P_{SN}(v)$ for this case, as has been shown by North [4] and others, are

$$P_N(v) = (v/\sigma^2) e^{-v^2/2\sigma^2}$$

[EQUATION 10]

and

$$P_{SN}(v) = (v/\sigma^2) e^{-(v^2/2\sigma^2 + S)} I_0\{(v/\sigma)\sqrt{2S}\}$$

[EQUATION 11]

where σ^2 is the noise power at the detector input, and S is the signal to noise power ratio at the same point. These equations assume a Gaussian distribution for the amplitude of the RF noise

voltage. This assumption is valid for the noise sources ordinarily present.

I_0 denotes the Modified Bessel Function of the order zero, which is related to the ordinary zero-order Bessel function J_0 by: $I_0(x) = J_0(ix)$; $i = \sqrt{-1}$. Equation (10) is the probability density function of a Raleigh distribution, and that of equation (11) is sometimes referred to as the Rician distribution (after S.O. Rice), who described its properties in detail, although it had been described previously by North and others.

The function of equation (10) is integrable. Therefore equation (9) can be solved for v_t . (The integration is included as Appendix B). The result is

$$P_{fa} = e^{-v_t^2/2\sigma^2} \quad \text{[EQUATION 12]}$$

Therefore,

$$(v_t/\sigma) = \sqrt{-2\ln(P_{fa})} \quad \text{[EQUATION 13]}$$

The ratio v_t/σ is the ratio of the threshold voltage to the predetection rms noise voltage.

It is more practical to express v_t in relation to the average noise voltage at the detector output (since this quantity is easily measured; it is the dc output voltage of the detector).

Since, (see Appendix C for the derivation)

$$v_{av} = \int_0^{\infty} v P_n(v) dv = \sigma \sqrt{\pi/2} \quad \text{[EQUATION 14]}$$

it follows that

$$v_t = (v_t/v_{av}) = \sqrt{-4 \ln(P_{fa})/\pi} \quad \text{[EQUATION 15]}$$

Robertson has normalized the threshold voltage differently; the

relationship of his threshold voltage, u_r to the one used here, u_t , is

$$u_r = \{(u_t - 1) \sqrt{M}\} / \sqrt{4/\pi - 1} \quad \text{[EQUATION 10]}$$

where M is the number of pulses integrated ($M = 1$ for equation (16)).

Results for u_t and also u_r , obtained by Blake by digital computer solutions of equations (15) and (16) are listed in Table 1 below.

P_{fa}	u_t	u_r
10^{-1}	1.7122	1.3625
10^{-2}	2.4215	2.7193
10^{-3}	2.9657	3.7605
10^{-4}	3.4245	4.6381
10^{-5}	3.8287	5.4114
10^{-6}	4.1941	6.1105
10^{-7}	4.5301	6.7534
10^{-8}	4.8429	7.3517
10^{-9}	5.1367	7.9137
10^{-10}	5.4146	8.4453
10^{-11}	5.6788	8.9509
10^{-12}	5.9313	9.4340
10^{-13}	6.1735	9.8973
10^{-14}	6.4066	10.3431
10^{-15}	6.6315	10.7733
10^{-16}	6.8489	11.1893

Table 1 Values of Single-Pulse Normalized Threshold Voltage for a set of False Alarm Probabilities

Blake then used these values of u_t for computing the curves given in the graph included as Appendix A, by numerical integration of equation (11) with v replaced by $u = v/v_{av}$; that is

$$P_d(S) = \int_{u_t}^{\infty} (\pi/2) u e^{-[(\pi u^2/4)+S]} I_0[u \sqrt{\pi S}] du \quad \text{[EQUATION 17]}$$

The derivation of equation (17) is included in Appendix D.

Note that this is where the discrepancy with Blake's equations and those derived occurs. The lower limit derived does not correspond.

Blake's equation for a single pulse was the first to be considered. When implemented, the algebraic derivation was not checked, and thus the results obtained were incorrect. The equations given by Hovanessian [3] were thus considered.

Equations derived by Hovanessian

In this section the expressions relating the single look probability of detection P_D , as a function of signal-to-noise ratio S/N with false alarm probability P_n as a parameter will be derived.

The single look probability of detection refers to detection probability attained by a single continuous illumination of the target as opposed to scan-to-scan illumination.

The effect of integration of several pulses on the probability of detection will be discussed later in this report.

Note that the false alarm probability P_n refers to the condition where noise exceeds the threshold setting resulting in a false detection.

Consider a sinusoidal signal augmented by Gaussian noise, $v(t)$, to constitute the input of the IF filter of figure 2 below.

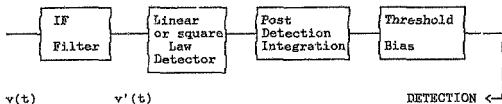


Figure 4 Signal processing circuitry

The output $v'(t)$ will be modified by filter characteristics which can be expressed using Fourier analysis. Assuming $v'(t)$ to be given by:

$$v'(t) = x(t)\cos 2\pi f_c t + y(t)\sin 2\pi f_c t$$

[EQUATION 18]

where f_c is the IF carrier frequency and $x(t)$ and $y(t)$ are orthogonal amplitudes of signal and noise components, the detector output can be expressed as a function of $x(t)$ and $y(t)$ as follows:

$$v = \sqrt{x(t)^2 + y(t)^2} \quad \text{[EQUATION 19]}$$

for a linear detector and v^2 for a square law detector.

The probability density and the cumulative probability for the envelope v are determined in a classic paper by S.O. Rice [17 and 18] and are expressed as follows:

$$p(v) = v e^{-\frac{(v^2 + (2 S/N))}{2}} I_0\{v \sqrt{2 S/N}\} dv \quad \text{[EQUATION 20]}$$

and

$$\int_0^v p(v) dv = \int_0^v v e^{-\frac{(v^2 + (2 S/N))}{2}} I_0\{v \sqrt{2 S/N}\} dv \quad \text{[EQUATION 21]}$$

where

v = envelope of signal plus noise

S/N = signal-to-noise power ratio (the multiplier 2 is used to convert rms signal power to peak signal power)

I_0 = hyperbolic Bessel function of zero order

$p(v)$ = probability density of v

$\int_0^v p(v) dv$ = cumulative probability distribution of v

To develop a direct relationship between P_s , P_n , and S/N ,

consider the following. From equation (21), it is seen that the probability of v lying between 0 and a fixed threshold E_t is given by:

$$\int_0^{E_t} p(v) dv = \int_0^{E_t} v e^{-\frac{(-v^2 - (2 S/N))}{2}} I_0\{v \sqrt{2 S/N}\} dv$$

[EQUATION 22]

so that the probability of detection P_D of a voltage consisting of a signal plus noise exceeding E_t is:

$$P_D = 1 - \int_0^{E_t} p(v) dv$$

[EQUATION 23]

$$P_D = 1 - \int_0^{E_t} v e^{-\frac{(-v^2 - (2 S/N))}{2}} I_0\{v \sqrt{2 S/N}\} dv$$

[EQUATION 24]

For the special case of when no signal is present, equation (22) yields the probability P_n , of a noise only voltage exceeding E_t , that is:

$$P_n = 1 - \int_0^{E_t} v e^{-v^2/2} dv = \int_{E_t}^{\infty} v e^{-v^2/2} dv$$

[EQUATION 25]

Therefore,

$$P_n = e^{-E_t^2/2}$$

[EQUATION 26]

Defining a new variable, u such that

$$u = e^{-\frac{v^2}{2}}, \quad v = \sqrt{-2 \ln(u)}$$

[EQUATION 27]

It is seen that

$$du = -v e^{-\frac{v^2}{2}} dv, \quad \text{and } E_t = P_n$$

[EQUATION 28]

so that equation (24) yields

$$P_S = 1 + \int_1^{P_n} e^{-S/N} I_0\{\sqrt{-4 S/N \ln(u)}\} du$$

[EQUATION 29]

or equivalently

$$P_S = 1 - e^{-S/N} \int_{p(n)}^1 I_0\{\sqrt{-4 S/N \ln(u)}\} du$$

[EQUATION 30]

Note from equations (29) and (30) that P_S is no longer explicitly dependent upon threshold function E_t . The effect of the threshold upon P_S is completely described by the probability of false alarm, P_n . Equation (30) can be put in an infinite series form and evaluated as a function of S/N with P_n as a parameter.

Equation (26) can thus be used to evaluate E_t given a value of the probability of false alarm.

Having this value, equation (24) can then be used to calculate the probability of detection p_S of a radar for a single pulse.

3.4.2. Detection with Many Pulses Integrated

Equations as derived by Blake

The equations described by Blake [1 and 2] for the detection of two pulses cannot practically be applied, (for even three pulses), and certainly not for many pulses.

However certain approximations can be made for the functions P_n and P_{sn} of Equations (8) and (9) in the many-pulse case which become very good if the number of pulses is sufficiently large.

These approximations are described by North [4]. They make use of the central limit theorem of probability theory, which states that the distribution of the sum of n independent random variables tends to a Gaussian (or normal) form as n tends to infinity, no matter what the distributions are for the individual variables, subject to some fairly mild conditions that are well satisfied in the present situation.

The variance and average values of these sum distributions are found by applying the following two well-known theorems:

- i. the standard deviation σ_M for the sum of M independent random variables is given by

$$\sigma_M^2 = \sum_{i=1}^M \sigma_i^2,$$

[EQUATION 31]

where σ_i is the standard deviation of the i th variable. Note that here 'Σ' is the 'SUM' which must not be confused with the RCS (radar cross-section).

- ii. the average sum, x_M , is given by

$$x_M = \sum_{i=1}^M x_i,$$

[EQUATION 32]

where x_i is the average value of the i th variable.

The desired Gaussian density functions for the detector output voltage v are therefore

$$p(v) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-(\sigma - v)^2 / 2\sigma_0^2} \quad \text{[EQUATION 33]}$$

where σ_0 and v_{av} are found by applying equations (31) and (32). Since a steady signal and constant noise power are assumed, all the v_1 's and σ_1 's in these equations are equal; hence for M pulses integrated the results are

$$v_{av} = M v_1 \quad \text{[EQUATION 34]}$$

and

$$\sigma_M = \sqrt{M} \sigma_1. \quad \text{[EQUATION 35]}$$

The quantity σ_0 in equation (33) is the standard deviation of the output detector voltage, which is not the same as that of the input voltage, denoted σ in equations (10) and (11).

The relationships of v_1 and σ_1 to the detector-input quantity σ for the noise-only case are

$$v_{1n} = \sigma \sqrt{\pi/2} \quad \text{[EQUATION 36]}$$

and

$$\sigma_{1n} = \sigma \sqrt{2 - \pi/2} = v_{1n} \sqrt{4/\pi - 1} \quad \text{[EQUATION 37]}$$

For the signal-and-noise case the relationships are

$$v_{1sn} = b \sigma \sqrt{\pi/2}, \text{ and} \quad \text{[EQUATION 38]}$$

$$\sigma_{1sn} = \sqrt{1n} \sqrt{(4/\pi)(1+S) - b^2},$$

[EQUATION 39]

where

$$b = e^{-S/2} [(1+S) I_0(S/2) + S I_1(S/2)]$$

[EQUATION 40]

(Note that for $S = 0$, $b = 1$, since $I_0(0) = 1$ and $I_1(0) = 0$)

The false-alarm probability is then given by

$$P_{fa}(S) = 1 / (\sqrt{2\pi}\sigma_{Msn}) \int_{v_t}^{\infty} e^{-(v-v_{Msn})^2 / 2\sigma_{Msn}^2} dv$$

[EQUATION 41]

and the probability of detection is

$$P_d(S) = 1 / (\sqrt{2\pi}\sigma_{Msn}) \int_{v_t}^{\infty} e^{-(v-v_{Msn})^2 / 2\sigma_{Msn}^2} dv$$

[EQUATION 42]

where v_t is the selected threshold voltage.

Integrals of the type shown can be written in terms of the error function,

$$\text{erf}(u) = 2/\sqrt{\pi} \int_0^u e^{-t^2} dt$$

[EQUATION 43]

which is a tabulated function. For computer programs, standard subroutines for erf(u) are available; alternatively direct

numerical integration can be employed.

In terms of this function, and with quantities in the arguments of the error function normalised to the average noise output (for M added samples), and with the normalised threshold voltage written as u_t , the equation for false-alarm probability and probability of detection become

$$P_{fa} = 1/2 - 1/2 \operatorname{erf} \left[\frac{u_t - 1}{\sqrt{2(4/\pi - 1)/M}} \right] \quad \text{[EQUATION 44]}$$

This equation can be solved for u_t by iteration, or by interpolation using a table of the error function. For some standard values of P_{fa} the iteration can be done 'in general' to find a constant k_{pfa} such that

$$u_t = 1 + k_{pfa} \sqrt{2(4/\pi - 1)/M} \quad \text{[EQUATION 45]}$$

The formula for k_{pfa} is

$$k_{pfa} = \operatorname{erf}^{-1} (1 - 2 P_{fa}) \quad \text{[EQUATION 46]}$$

where the notation erf^{-1} denotes the 'inverse error function'.

Some useful values of k_{pfa} , which were obtained by Blake [1 and 2] by interpolation using a 15-place table of the error function are given in the table below:

P_{fa}	k_{pfa}
10^{-4}	2.630
10^{-6}	3.361
10^{-8}	3.958
10^{-10}	4.498
10^{-12}	4.974

Table 2 Values of Inverse Error Function, k_{pfa} for Some Standard Value of False Alarm Probability

The corresponding equation for P_d is

$$P_d = 1/2 - 1/2 \operatorname{erf} \left[\frac{u_t - b}{\sqrt{2[4(1+S)/\pi - b^2]/M}} \right] \quad \text{[EQUATION 47]}$$

Again, a constant can be found for a specified value of P_d :

$$k_{pd} = \operatorname{erf}^{-1} (1 - 2 P_d) \quad \text{[EQUATION 48]}$$

The formula for $k_{1,d}$ is

$$k_{pd} = \frac{u_t - b}{\sqrt{2[4(1+S)/\pi - b^2]/M}} \quad \text{[EQUATION 49]}$$

It must be noted that this equation cannot be solved explicitly for S , because b (equation (40)) is a function of S . If the explicit expression for b is used, it becomes evident that an algebraic solution is not possible.

Consequently iteration is required for numerical solution of equation (49) for S , just as it would be if equation (47) were used directly.

Nevertheless, the procedure of finding k_{pd} and writing equation (49) is advantageous when solutions for several or many values of N are desired for the same value of P_d , because the iterative solution of equation (49) does not involve numerical integration, as does that of equation (47).

Values of k_{pd} for some standard values of P_d are given in Table 4 overleaf. These were obtained by Blake [1 and 2] by iteration and an error function subroutine.

It is noteworthy that

$$k_{pd}(P_d) = -k_{pd}(1 - P_d) \quad \text{[EQUATION 50]}$$

so that one iteration procedure gives values of k_{pd} for two values of P_d . Also for $P_d = 0.5$, $k_{pd} = 0$ exactly.

P_d	k_{pd}
0.01	1.6449760
0.05	1.1630870
0.10	0.9061940
0.25	0.4769363
0.50	0.0000000
0.75	-0.4769363
0.90	-0.9061940
0.95	-1.1630870
0.99	-1.6449760

Table 3 Values of Inverse Error Function, k_{pd} , for Some Standard Values of Probability of Detection, P_d

As the number of pulses M is increased one would expect the results to become more accurate - because the assumption of Gaussian density curves for P_n and P_{gn} (equations (10) and (11) is more nearly correct as the number of pulses integrated increases (in accordance with the central limit theorem).

It is probably a good rule of thumb that the method is virtually exact for numbers of pulses in the thousands, very good for numbers in the hundreds (error of order 0.1 dB), and in error by roughly 0.5 dB at about ten pulses integrated.

3.5. Target Fluctuation Effects

The previous discussions have considered a steady target signal in a noise background. Yet, most real radar targets are not steady, but rather fluctuating signals, introducing a further statistical uncertainty in the detection process.

In general, the effect of fluctuation is to require higher signal-to-noise ratios for high probability of detection, and lower values for low probability of detection, than those required with nonfluctuating signals.

Swirling has considered four cases, which differ in the assumed rate of fluctuation and the assumed statistical distribution of the cross section.

The two assumed rates are:

- i. a relatively slow fluctuation, such that the values of Σ for successive scans of the radar beam past the target are statistically independent but Σ remains constant from one pulse to the next.

- ii. a relatively fast fluctuation, such that the values of Σ are independent from pulse to pulse within one beamwidth of the scan (i.e., during the integration time).

The first of the two assumed distributions for the received-signal is the Rayleigh form, which means that the target cross section Σ has a probability-density function given by

$$p(\Sigma) = \frac{1}{\Sigma} e^{-\Sigma/\Sigma_{av}}$$

[EQUATION 51]

where Σ_{av} is the average cross-section. (This is called a negative-exponential density function, but a target having this distribution is called a Raleigh target.)

The second assumed cross-section density function is

$$p(\Sigma) = \frac{4\Sigma}{\Sigma_{av}^2} e^{-2\Sigma/\Sigma_{av}}$$

[EQUATION 52]

(Note that the Raleigh density function for a voltage v is

$$p(v) = \frac{2v}{r^2} e^{-v^2/r^2} \quad (v \geq 0)$$

[EQUATION 53]

where r is the rms value of v ; i.e., $r^2 = v_{av}^2$).

The first distribution is observed when the target consists of many independent scattering elements, of which no single one (or just a few) predominate. Many aircraft have approximately this characteristic at microwave frequencies, and large complicated targets are usually of this nature. (This result is predicted by the central limit theorem of probability theory.)

The second distribution corresponds to that of a target having one main scattering element that predominates, together with many smaller independent scattering elements.

In summary, the cases considered by Swerling (information obtained from references [11] and [15]) are as follows:

Case 1: Equation (51), slow fluctuation

Case 2: Equation (51), fast fluctuation

Case 1: Equation (52), slow fluctuation

Case 1: Equation (52), fast fluctuation

Swerling's case 1 is the most often assumed when range prediction is to be made for a nonspecific fluctuating target.

Cases 2 and 4 are of lesser interest because they are less frequently encountered and because for more than about 10 pulses integrated the results are very nearly the same as those for the steady-signal case.

3.6. Coverage Diagram

The factors F_t and F_r in the range equation account for the facts that the target may not be in the antenna pattern maximum and that the wave propagation between antennas and target and may not be free-space propagation.

Thus F_t and F_r ($F = F_t$ or F_r , since a monostatic radar uses the same antenna for both transmitting and receiving) contain antenna pattern factors and also account for reflection-interference (multipath) effects, diffraction and shadowing, absorption losses, and abnormal refraction effects.

The ultimate goal of pattern-propagation-factor calculations is usually a plot of the radar maximum range (R_{max}) as a function of either the target elevation (θ_1) or the target height (h_2).

For distant targets, F can be computed as a function of θ by using the flat earth formulas. Here h must be less than 1000 feet and the following inequality $\tan(\theta_1) > h_1 \times 10^{-3}$ must be satisfied, where h_1 is the height of the antenna.

However, when the inequality is satisfied the earth's curvature must be taken into account.

The range-height-angle relationship, for ranges very much smaller than the earth's radius a , is

$$h_2 = \frac{R_{max} \cos^2(\theta_1)}{2ka} + R_{max} \sin(\theta_1) + h_1$$

[EQUATION 54]

In summary, the cases considered by Swerling (information obtained from references [11] and [15]) are as follows:

Case 1: Equation (51), slow fluctuation

Case 2: Equation (51), fast fluctuation

Case 1: Equation (52), slow fluctuation

Case 1: Equation (52), fast fluctuation

Swerling's case 1 is the most often assumed when range prediction is to be made for a nonspecific fluctuating target.

Cases 2 and 4 are of lesser interest because they are less frequently encountered and because for more than about 10 pulses integrated the results are very nearly the same as those for the steady-signal case.

3.6. Coverage Diagram

The factors F_t and F_r in the range equation account for the facts that the target may not be in the antenna pattern maximum and that the wave propagation between antennas and target and may not be free-space propagation.

Thus F_t and F_r ($F = F_t$ or F_r , since a monostatic radar uses the same antenna for both transmitting and receiving) contain antenna pattern factors and also account for reflection-interference (multipath) effects, diffraction and shadowing, absorption losses, and abnormal refraction effects.

The ultimate goal of pattern-propagation-factor calculations is usually a plot of the radar maximum range (R_{max}) as a function of either the target elevation (θ_1) or the target height (h_2).

For distant targets, F can be computed as a function of θ_1 by using the flat earth formulas. Here h must be less than 1000 feet and the following inequality $\tan(\theta_1) > h_1 \times 10^{-3}$ must be satisfied, where h_1 is the height of the antenna.

However, when the inequality is satisfied the earth's curvature must be taken into account.

The range-height-angle relationship, for ranges very much smaller than the earth's radius a , is

$$h_2 = \frac{R_{max} \cos^2(\theta_1)}{2ka} + R_{max} \sin(\theta_1) + h_1$$

[EQUATION 54]

A range height chart can be constructed on the basis of equation (54), in which the rays are straight lines and (if the range height scales are in the same units) the constant-range lines are circles with the origin as centre.

(If the units are not the same, the constant-range lines will be ellipses.) The constant-height contours on this chart curve downward, with a curvature that is equal to the fictitious earth of radius $a_e = k(a + h_1)$.

Plots of radar range as a function of the elevation angle can be made on such charts; such plots are called coverage diagrams.

A typical coverage diagram using a cosecant-squared shaped beam is shown in appendix H.

4. SOLUTION OF THE PROBLEM

4.1. Software Development

The software package which was developed is structured as shown in the flow chart included as Appendix J. It should be pointed out that in the development of the software package an emphasis was placed on a 'user friendly' and menu driven type package.

Help menu's are included whenever a user is required to input information, thus aiding the radar designer.

The software was written in Turbo PASCAL (Borland). The use of procedures allows changes to the overall system to be made. If at a later stage further procedures need to be incorporated the modular structure of the program caters for this adequately.

4.1.1. Flow Chart Description

Referring to the flow chart included as Appendix J, the user is presented with the **Main Menu**. The various options which are presented to the user are as follows:

- 1) Calculation of the Radar Range.
- 2) Pfa (probability of false-alarm), Pd (probability of detection) curves for **Single Pulse**.
- 3) Pfa (probability of false-alarm), Pd (probability of detection) curves for **Multiple Pulse**.
- 4) Plotting of Curves
- 5) Help (Y/N)
- 6) QUIT ?

Each of the above mentioned options will now be discussed in greater detail.

4.1.1.1. Calculation of the Radar Range

On selection of this option the user is prompted on whether he needs help or further information on the selection just made.

In order to calculate the radar range, the user is prompted for the following information, namely:-

- 1) P_t (the transmitted power in watts).
- 2) G_t (the transmitting antenna gain in dB).
- 3) G_r (the receiving antenna gain in dB).
- 4) Σ (the radar target cross section in m^2). If the target cross section is fluctuating it is calculated according to cases 1 to 4 as specified by Swerling (refer to by Skolnik [15]). The user is prompted for the average radar cross-section which is then used to calculate the probability density function for the cross-section.

(Refer to equations 51 and 52).

- 5) F_t (the pattern propagation factor for the transmitting antenna-to-target path).
- 6) F_r (the pattern propagation factor for the target-to-receiving antenna path).
- 7) B_n (the receiver bandwidth in Hertz).
- 8) L (the generalised system losses in dB). This includes:

plumbing loss (a loss experienced in the transmission lines which connect the output of the transmitter to the radar),

beam-shape loss (this is a loss brought into account as the maximum gain is employed in the radar equation rather than a gain that changes pulse to pulse).

limiting loss (loss caused by limiting in the radar receiver thus lowering the probability of detection).

collapsing loss (if the radar integrates additional noise samples along with the wanted signal-to-noise pulses, the added noise results in a further degradation).

other losses due to non-ideal equipment, operator loss, and field degradation.

- 9) Λ (the wavelength in metres).
- (Refer to equation 1 for calculating the radar range).

10) (S/N) (the signal-to-noise ratio). If (S/N) needs to be calculated depending on the type of integration to be performed, the user is prompted for:

- a. Single Pulse Integration
The probability of detection and the probability of false alarm.

(Refer to equation 8 to 30).

- b. Multiple Pulse Integration
The number of pulses to be integrated, the probability of detection and the probability of false alarm.

(Refer to equations 31 to 50).

4.1.1.2. P_d, P_{fa} Curves for Single Pulse Integration

The user is prompted on whether he needs help or further information on the selection just made.

In order to generate the probability of detection versus signal-to-noise ratio curves for a given probability of false alarm for single pulse (non-coherent) integration, the user is prompted for the following information, namely:-

- 1). Table Name under which the calculated data can be stored. (Several checks are made to see whether the file exists and if the user wants to create a new one).
- 2). $SdB(Lower)$, $SdB(Upper)$ the lower and upper values of the signal-to-noise ratio range to be plotted must be entered.
- 3). P_{fa} , P_d for a given probability of false alarm the corresponding probability of detection is calculated and written to the specified table name.

(Refer to equations 13,18 and 24,26).

Note:- Turbo PASCAL saves its files in a hexadecimal (HEX) format, and when the TYPE command is used under DOS the equivalent graphics characters are seen. A utility was written to save the calculated values in ASCII format allowing the user to TYPE the file and see the values tabulated.

10) (S/N) (the signal-to-noise ratio). If (S/N) needs to be calculated depending on the type of integration to be performed, the user is prompted for:

a. Single Pulse Integration
The probability of detection and the probability of false alarm.

(Refer to equation 8 to 30).

b. Multiple Pulse Integration
The number of pulses to be integrated, the probability of detection and the probability of false alarm.

(Refer to equations 31 to 50).

4.1.1.2. P_d, P_{fa} Curves for Single Pulse Integration

The user is prompted on whether he needs help or further information on the selection just made.

In order to generate the probability of detection versus signal-to-noise ratio curves for a given probability of false alarm for single pulse (non-coherent) integration, the user is prompted for the following information, namely:-

- 1) Table Name under which the calculated data can be stored. (Several checks are made to see whether the file exists and if the user wants to create a new one).
- 2). SdB(Lower), SdB(Upper) the lower and upper values of the signal-to-noise ratio range to be plotted must be entered.
- 3). P_{fa}, P_d for a given probability of false alarm the corresponding probability of detection is calculated and written to the specified table name.

(Refer to equations 13,16 and 24,26).

Note:- Turbo PASCAL saves its files in a hexadecimal (HEX) format, and when the TYPE command is used under DOS the equivalent graphics characters are seen. A utility was written to save the calculated values in ASCII format allowing the user to TYPE the file and see the values tabulated.

4.1.1.3. P_d , P_{fa} Curves for Multiple Pulse Integration

Once again the user is prompted on whether he needs help or further information on the selection just made.

In order to generate the probability of detection versus signal-to-noise ratio curves for a given probability of false alarm, for multiple pulse (non-coherent) integration the user is prompted for the following information, namely:-

1. **Table name** under which the calculated data can be stored. (Several checks are made to see whether the file exists and if the user wants to create a new one).
2. **SdB(lower), SdB(upper)** values of the signal-to-noise ratio range to be plotted must be entered.
3. **N** the number of pulses to be integrated.
4. **P_{fa} , P_d** for a given probability of false alarm the corresponding probability of detection for multiple pulse integration is calculated and written to the specified table name.

Note:- At present this option is not working satisfactorily. The only equations which could be obtained for the integration of multiple pulses are taken from Blake [1 and 2]. The equations which Blake has given could not be derived.

(Refer to equations 31 to 50).

4.1.1.4. Curve Plot Procedure for Tables Generated

The user is prompted on whether he needs help or further information on the selection just made.

In order to plot the probability of detection versus signal-to-noise ratio curves for a given probability of false alarm, for both single and multiple pulse (non-coherent) integration the user is prompted for the following information, namely:-

1. **Table name** under which the calculated data was stored. A check is made to see whether the file exists and if it does it is plotted.
2. **Printout** The user is given the option to obtain a printout of the `graph` appearing on the console. The printer must be able to print graphic characters.

4.1.1.5. Coverage Diagram

The user is prompted for the for the angle at which the maximum antenna-pattern gain occurs.

The transmitting and receiving antenna gain at the varying angles must then be entered.

The values are input in a file. The file is then read and using equation (1) with all the necessary parameters the radar range or varying angle of elevation and height is calculated using equation (54).

This option has not yet been completed in the software package presently available

(Refer to equation (54))

5. PROBLEMS ENCOUNTERED AND THEIR SOLUTIONS

The development of the software package went quite quickly at first.

This involved developing code to evaluate the radar range equation.

The user would be prompted for the different parameters and the radar range calculated.

At this point in time the values of the signal-to-noise ratio and the fluctuating targets had not been considered.

With the given equations, a decision was made to implement the system using TURBO Pascal as the development language and the TURBO Graphics utilities for the graphics functions.

The integral technique which was used was the 4th order Runge Kutta, giving good results for most functions integrated.

The equations that Blake derived were implemented first. Blake's equation for determining the probability of detection for a single pulse is given as equation (17).

Many problems were encountered.

- 1) Firstly Blake does not define the units which he uses for the various variables in his equations.
- 2) Typically the signal-to-noise ratio, is this in decibels (dB), a voltage ratio, a power ratio, or is it a numerical ratio.
- 3) The value for the predetection rms noise voltage σ is not given.
- 4) The evaluation of the Bessel function was evaluated using an infinite series taking 26 terms, and giving accurate answers for x ranging 0 to 10. If x was greater than 10 a floating point overflow error would be given, implying that the particular version of Pascal could not handle numbers greater than 10^{37} .
- 5) It was decided to verify Blake's equations by deriving each one carefully, as given in appendices 2 to 4.

In Appendix D a discrepancy with Blake's equations and those derived occurs. The lower limit for the evaluation of the probability of detection P_d , equation (17) does not correspond with that derived. Following the derivation one sees that the lower limit is:

$$u_t \sigma \sqrt{\pi/2}$$

and not just u_t as given in equation (17).

This error in Blake's equation led to many of the problems encountered whilst trying to obtain the curves for the probability of detection, given the probability of detection and signal-to-noise ratio.

It was therefore decided to try and obtain another reference to develop the software package to evaluate the probability of detection for single pulse integration.

The equation as derived by Hovanessian [3] was the one used. The equation which was used to obtain the probability of detection for a single pulse is equation (24).

It appears at first that the integral for p_d is easy to evaluate, but on closer examination it is not so.

The Bessel function $I_0(x)$ also contains an integral and therefore a double integral must be evaluated.

It must also be noted that as the interval in equation (24) is incremented from 0 to E_t so the Bessel function must be evaluated for each value in a change in the variable u , that is du .

The function to be integrated is defined under the procedure **derivative**. As the Bessel Function must also be evaluated, a second function namely **derivativo** is also defined. (See Appendix E, the program called 'FLOTPD.COM'.)

The package requests the upper and lower limits of the signal-to-noise ratio, and the probability of false alarm for single pulse integration. It then calculates the value of E_t (threshold voltage), the upper limit of the integral. Note that the S/N ratio is converted from decibels (dB) to a numerical ratio, assuming it to be a voltage ratio.

The Bessel function is evaluated first, and then the probability of detection ($p_d = P_d$). This is done in a loop until the value of steps is reached whereupon the loop terminates and an answer obtained. The value for the signal-to-noise ratio is increased if a curve is to be plotted and the values are written to a file, and can be plotted at a later stage.

It must be noted that as a personal computer is used, the evaluation of the double integral is rather slow.

When the program was run at first, the answers which were returned were far from those previously determined by Blake and Hovanessian.

On closer examination of the of the numerical integration process, the value of the truncation error was monitored, noting that it grew as the program progressed.

To overcome this problem, the step length was decreased (Dorn and McCracken [16]).

Eventually a so-called 7 to 8 point method is used.

The step length h used to evaluate the Modified Bessel function is approximately 0.5, whereas that for the p_3 (probability of detection) integral is typically in the range 0.019 to 0.004.

The step length varies because as the value of the probability of false alarm varies so does E_t and thus the upper limit of the integral. It therefore influences the step length.

The software development for evaluation of the probability of detection and signal-to-noise for a given probability of false alarm and number of pulses to be integrated caused many problems. The equations that were implemented are those derived by Blake. Several attempts were made to try and derive these equations, namely equations 31 to 50 and all attempts were unsuccessful. The implementation of the equations were performed and reasonable results obtained for a single pulse ($M=1$). Values for pulses greater than 1 to be integrated led to run-time errors. The error that occurred was an overflow error, caused by a division by a very small number (tending to zero). This was due to the constant 'b' in equation 40. As 'S' (the signal to noise ratio, a voltage ratio) is increased, so 'b' tends to zero. This problem could not be overcome.

6.

RESULTS

The graphs as obtained by Blake and Hovnessian for single pulse integration are included in appendices A and Z respectively.

Although the package developed returns the probability of detection for a given probability of false alarm and a given signal-to-noise ratio it was extended to so that the full curves for a particular p_n could be plotted.

Running the program PLOTPD.COM will prompt the user for a range of signal-to-noise ratio that must be calculated for a given probability of false alarm (P_n). The results are written to an ASCII file which can then be read using the package called SYMPHONY and plotted. Examples are included in Appendix G and compare favourably with those obtained by Blake and Hovnessian.

Using the plot option in the radar program allows the users to plot a single curve, using the HardCopy facility available in the Turbo Graphics package.

Running the program MULTIPLE.COM will prompt the user for the probability of false alarm and number of pulses to be integrated. The user can watch the program executing and see that the threshold voltage is calculated correctly. This value is then used as the upper limit to the integral to calculate the probability of detection for multiple pulse integration. This facility is not giving the correct results. The equations could not be verified, thus the implementation proved to be unsuccessful.

7.

CONCLUSION

Developing a software package for determining the probability of detection of a radar for a single pulse had been tackled previously, by Blake [1 and 2], Hovannesian [3] and Robertson, but not on a personal computer.

The determination of the probability of detection using multiple pulse integration will only be possible if the equations as derived by Blake [1 and 2] can be verified. Derivations of these equations could not be found or successfully determined. It appears that the equations presented by Blake could be incorrect, a similar occurrence when developing the package for the single pulse integration.

The personal computer, although a very useful tool clearly has its limitations when so called 'number crunching' must be performed.

But as PC's are available to most designers of radar systems, once the curves have been plotted using the integration package, the results obtained are accurate and compare favourably with those obtained in [1,2 and 3].

The problem appeared to be easy at first but the experience gained and showing that the work done previously was correct proved to be a worthwhile learning experience.

APPENDIX A

A graph (taken from Blake[2]) showing the required signal to noise ratio at the input terminals of a linear rectifier detector as a function of probability of detection for a single pulse, with the false alarm probability (P_{fa}) as a parameter, calculated for a non-fluctuating signal.

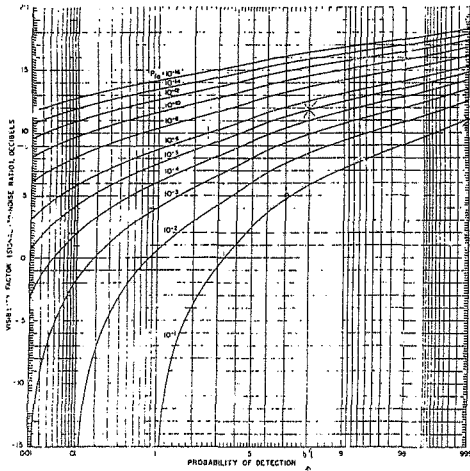


Fig. 4 - Required signal-to-noise ratio (visibility factor) at the input terminals of a linear-rectifier detector as a function of probability of detection for a single pulse, with the false-alarm probability (P_{fa}) as a parameter, calculated for a non-fluctuating signal. (Note: This figure also appears in an appendix at the end of the report.)

APPENDIX B

Substitution of equation (10) into equation (9) yields the following:

$$\begin{aligned} P_{fa} &= \int_{v_t}^{\infty} (v/\sigma^2) e^{-v_t^2/2\sigma^2} dv \\ &= -e^{-v^2/2\sigma^2} \Big|_{v_t}^{\infty} \\ &= -e^{-\infty} - (-e^{-v_t^2/2\sigma^2}) \\ &= e^{-v_t^2/2\sigma^2} \end{aligned}$$

Therefore $P_{fa} = e^{-v_t^2/2\sigma^2}$ which yields the result given as equation (12).

APPENDIX C

Substitution of equation (10) to obtain the value for v_{av} , we have

$$v_{av} = \int_0^{\infty} v P_n(v) dv$$

$$= \int_0^{\infty} (v^2/\sigma^2) e^{-v^2/2\sigma^2} dv$$

$$= (1/\sigma^2) \times \frac{\Gamma(3/2)}{2 \times (1/2\sigma^2)^{3/2}}$$

$$= (1/\sigma^2) \times (\sqrt{\pi}/2) / (2 \times (1/2)^{3/2} \times (1/\sigma^2)^{3/2})$$

$$= \frac{1}{\sigma^2} \frac{\sqrt{\pi}}{\sqrt{2}} \frac{1}{\sqrt{2\sigma^2}}$$

$$= \frac{\sqrt{\pi} \sqrt{2} \sigma^2}{4\sigma^2}$$

$$= \frac{\sigma \sqrt{\pi} \sqrt{2} \cdot 2}{4}$$

$$= \sigma \sqrt{\pi} \frac{1}{\sqrt{2}}$$

$$= \sigma \sqrt{\pi/2} \quad \text{as given by equation (14).}$$

APPENDIX C (CONTINUED)

From reference 5 the equation for the 'gamma' function was obtained.

$$\int_0^{\infty} x^m e^{-ax^2} dx = \frac{\Gamma[(m+1)/2]}{2a^{(m+1)/2}}$$

and also $\Gamma(m+1/2) = \frac{1 \cdot 3 \cdot 5 \dots (2m-1)}{2^m}$

When substituting $m = 2$, and $a = 1/2\sigma^2$, which yields $\Gamma(3/2) = \frac{\sqrt{\pi}}{2}$

APPENDIX D

Replacing v with $u = v/v_{av}$ and substituting the formula into equation (11) yields the following calculations:

$$P_d(S) = \int_{v_t}^{\infty} (v/\sigma^2) e^{-\frac{(v^2/2\sigma^2 + S)}{\sigma^2}} I_0\{v/\sigma \sqrt{2S}\} dv$$

replacing v by $u_t = v_t/v_{av}$

where $v_{av} = \sigma \sqrt{\pi/2}$

$$\text{i.e. } u_t = (v_t/\sigma) \sqrt{2/\pi}$$

$$\Rightarrow v_t = u_t \sigma \sqrt{\pi/2}$$

$$\Rightarrow dv_t = \sigma \sqrt{\pi/2} du_t$$

$$P_d = \int_{u_t \sigma \sqrt{\pi/2}}^{\infty} u \sigma \sqrt{\pi/2} e^{-\frac{(u^2 \sigma^2 \pi/2 + S)}{2\sigma^2}} I_0\{(u \sigma \sqrt{\pi/2})/\sigma \sqrt{2S}\} \sigma \sqrt{\pi/2} du$$

$$= \int_{u_t}^{\infty} \frac{1}{\sigma \sqrt{\pi/2}} e^{-\frac{(u^2 \pi/4 + S)}{\sigma^2}} I_0\{u \sqrt{\pi S}\} du$$

This is not u_t only, as given by Blake [1 and 3] !!!!

APPENDIX E

Source listing of pascal program

The following programs and procedures have been included in this appendix. They are listed in alphabetical order.

1. BESSEL.PAS
2. DISPVAL.PAS
3. GAINTHET.PAS
4. GETSELEC.PAS
5. GRAFLOT.PAS
6. MAINHELP.PAS
7. MULTIPLE.PAS
8. PDERF.PAS
9. PDI1.PAS
10. PDRKINTG.PAS
11. PDTABLE.PAS
12. PFAFILE.PAS
13. PLOTDP.PAS
14. POWER.PAS
15. PROBDET.PAS
16. RADAR.PAS
17. RANGE.PAS
18. TABLESEL.PAS
19. VARIABLE.PAS

The first comment specifies briefly what the program/procedure is used for. Comments are interspersed to allow for easier understanding of how the program was developed.

Program Name: C:\HDV\BESSEL.PAS
Date: Sunday, August 23, 1987
Time: 22:37

Page: 1

```
1 A B (
2 A B |
3 A B | This procedure is used to evaluate the Bessel function by using the
4 A B | integral function for the Bessel Function. The user is prompted for
5 A B | the value to be calculated and the result is obtained. The
6 A B | procedure is used extensively in determining the probability of
7 A B | false alarm and probability of detection for both single and
8 A B | multiple pulse integration. The 4th order Runge-Kutta technique
9 A B | is used for the integration. The answer is returned as the value
10 A B | INTEGR. The function is specified in the procedure Derivative_B
11 A B |
12 A B | Written By : Cyril Harari
13 A B |
14 A B |
15 A B |
16 B | procedure Bessel_Integral;
17 B |
18 B | BEGIN
19 B | Clrscr
20 B | gotoxy(2,6);
21 B | writeln('Bessel function evaluation by using integral function');
22 B | gotoxy(8,7);
23 B | writeln('');
24 B | result:=0;
25 B | xi:=0;
26 B | gotoxy(10,10);
27 B | write('Enter value for x : ');
28 B | readln(xi);
29 B | st:=0;
30 B | fi:=2*pi;
31 B | NumSteps:=20;
32 B | delta := (fi - st) / NumSteps;
33 B | integr := 0;
34 B | End;
35 A |
36 B | procedure Derivative_B;
37 B | BEGIN
38 B | x := st;
39 B | deriv := explsin(x);
40 B |
41 B | End;
42 A |
43 B | procedure R_K_Loop_B;
44 B |
45 B | ( inner 4th order Runge-Kutta loop. The number of times this routine )
46 B | ( is called depends on the variable 'NumSteps'. )
47 B |
48 B | BEGIN
49 B | For rkc2 := 1 To 4 Do
50 B | BEGIN
51 B | Derivative_B;
52 B | Case rkc2 Of
53 B | 1 BEGIN
54 B | k4 := integr;
55 B | k1 := deriv * delta;
```

Program Name: C:\HDD\B\SSEL.PAS
Date: Sunday, August 23, 1987
Time: 22:37

Page: 2

```
55 B 4      integr := k4 + k1 / 2;  
57 B 4      st := st + delta / 2;  
58 B 4      End;  
59 B 4      2: Begin  
60 B 4      k2 := deriv * delta;  
61 B 4      integr := k4 + k2 / 2;  
62 B 4      End;  
63 B 4      3: Begin  
64 B 4      k3 := deriv * delta;  
65 B 4      integr := k4 + k3;  
66 B 4      st := st + delta / 2;  
67 B 4      End;  
68 B 4      4: Begin  
69 B 4      integr := st + (k1 + k2 * 2 + k3 * 2 + deriv * delta) / 6;  
70 B 4      End;  
71 B 3      End;  
72 B 2      End;  
73 B 1      End;  
74 A 0  
75 A 0  
76 B 0      Procedure Runge_Kutta;  
77 B 0  
78 B 1      Begin  
79 B 1      For rkcl := 1 To NumSteps Do  
80 B 1          R_K_Loop_B;  
81 B 1      End;  
82 A 0  
83 B 0      Procedure Bessel_Integral_Results;  
84 B 0  
85 B 1      Begin  
86 B 1      gotoxy(8, 15);  
87 B 1      writeln('RESULT');  
88 B 1      gotoxy(8, 16);  
89 B 1      writeln('=====');  
90 B 1      gotoxy(10, 20);  
91 B 1      integr:=integr/(2*pi);  
92 B 1      write('Io(', Io:3:1, ') = ', integr:5:3);  
93 B 1      gotoxy(25, 25);  
94 B 1      write('Hit any key to continue...');  
95 B 2      repeat until keypressed;  
96 B 1      End;
```

Program Name: C:\HVV\DISPVAL.PAS
Date: Sunday, August 23, 1987
Time: 22:44

Page: 1

```
1 P 0 (
2 A 0 |
3 A 0 | This procedure is used to display the transmitter and receiver
4 A 0 | gain values with varying angle of elevation which were stored using
5 A 0 | the file GAINTHET.PAS
6 A 0 |
7 A 0 |
8 A 0 |
9 A 0 |
10 A 0 |
11 A 0 |
12 A 0 |
13 A 0 |
14 A 0 |
15 A 0 |
16 B 0 Procedure Display_Values;
17 B 1 begin
18 B 1 ( First open the table file. )
19 B 1
20 B 1 Assign(TabFile,FileName);
21 B 1 Reset(TabFile);
22 B 1
23 B 1 gotoxy(9,5);
24 B 1 Write'GAIN - THETA Table for ',FileName, ' file.';
25 B 1 gotoxy(14,6);
26 B 1 writeln('');
27 B 1 writeln
28 B 1
29 B 1 gotoxy(5,10);
29 B 1 Write'Gain Receiver Gain Transmitter Theta';
30 B 1 writeln
31 B 1
31 B 1 While not eof(TabFile) do
32 B 2 begin
33 B 2 Read(TabFile,MemRec);
34 B 2 Writeln(' ',MemRec.GainR:4,' ',MemRec.GainT:4,' ',MemRec.theta:4);
35 B 2 end;
36 B 1 gotoxy(25,25);
37 B 1 Write'Hit any key to continue...';
38 B 2 repeat until keypressed;
39 B 1 Close(TabFile);
40 B 1 end;
```


Program Name: C:\NOVA\BETSLEC.PAG
Date: Sunday, August 23, 1987
Time: 22:54

Pages: 1

```
1 A 0 ( ..... )  
2 A 0 t .....  
3 A 0 t This procedure is the procedure which displays the main menu on  
4 A 0 t the screen. The user is prompted for his choice and once made it is  
5 A 0 t executed. This procedure is executed within a repeat until loop  
6 A 0 t so that once a key is pressed that option is executed.  
7 A 0 t .....  
8 A 0 t .....  
9 A 0 t .....  
10 A 0 t .....  
11 A 0 t .....  
12 A 0 t Written By : Cyril Harari  
13 A 0 t ..... )  
14 A 0 ..... )  
15 A 0  
16 0 PROCEDURE get_selection ;  
17 0 ( ..... )  
18 0  
19 0 BEGIN  
20 0 clrscr;  
21 0 gotoxy(1,3);  
22 0 writeln(' :28,'RADAR SIMULATION SYSTEM');  
23 0 writeln(' :28,'.....');  
24 0 gotoxy(1,6);  
25 0 writeln(' :5,'1) Calculation of the Range (Complex formula)');  
26 0 writeln;  
27 0 writeln(' :5,'2) Gain - Theta table definition');  
28 0 writeln;  
29 0 writeln(' :5,'3) Non-coherent integration for a SINGLE pulse');  
30 0 writeln;  
31 0 writeln(' :5,'4) Non-coherent integration for MULTIPLE pulses');  
32 0 writeln  
  
33 0 writeln(' :5,'5) Calculation of the Receiver Power');  
34 0 writeln;  
35 0 writeln(' :5,'6) Bessel Function using integrati-');  
36 0 writeln  
  
37 0 writeln  
  
38 0 writeln  
  
39 0 writeln(' :5,'8) HELP ');  
40 0 writeln;  
41 0 writeln(' :5,'9) Quit:');  
42 0 gotoxy(28,25);  
43 0 write 'Enter choice : ?';  
44 0 END ; ( get_selection )
```

Program Name: C:\MSV\GRAPHLOT.PAS
Date: Sunday, August 23, 1987
Time: 23:0

Page: 1

```
1 A 0 ( !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! )
2 A 0 |
3 A 0 | This program is used to plot the values generated by the program |
4 A 0 | PLOTTPD.COM It generates a graph using the Turbo Graphics utility |
5 A 0 | POLYGEN.HGH and the procedure SPLINE.HGH. It is also used so that a |
6 A 0 | smooth curve between the given points can be plotted. Note that |
7 A 0 | the Turbo Pascal standard file format with extension '.dat' must |
8 A 0 | be entered so that the file can be plotted. Note that certain |
9 A 0 | *.sys and *.hgh, namely Turbo Graphics utilities must be included |
10 A 0 | for the program to run. The HardCopy procedure is used to generate |
11 A 0 | a printout of the graphics screen. The value of false, f in this |
12 A 0 | procedure implies that the active screen is plotted in EPSON |
13 A 0 | Graphics mode ( 960 point per line). |
14 A 0 |
15 A 0 |                               Written By : Cyril Harari |
16 A 0 |
17 A 0 ( !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! )
18 A 0
19 A 0 program Interpolate;
20 A 0
21 A 0
22 A 0 ($I c:\toolbox\typedef.sys)           (these files must be)
23 A 0 ($I c:\toolbox\graphix.sys)         (included in this order)
24 A 0 ($I c:\toolbox\kernel.sys)
25 A 0 ($I c:\toolbox\windows.sys)
26 A 0 ($I c:\toolbox\finder\d.hgh)
27 A 0 ($I c:\toolbox\axis.hgh)
28 A 0 ($I c:\toolbox\polygon.hgh)
29 A 0 ($I c:\toolbox\spline.hgh)
30 A 0 ($I ($I c:\toolbox\duany.inc) )
31 A 0 ($I variable.pas)
32 A 0
33 D 0 procedure Spline0ns;
34 B 0
35 B 0 var x,compres;
36 B 0 dx,dy,i,o,n,lines,scale;integer;
37 B 0 xi,yi,y2,z2;integer;
38 B 0 b,x;Plotarray;
39 B 0
40 B 0 begin
41 B 0
42 B 0 DefineWindow(1,0,0,XMax61b,YMax61b); (define both windows as whole screen)
43 B 0 DefineWindow(2,0,0,XMax61b,YMax61b);
44 B 0 DefineWorld(1,0,1000,1000,81);      (give a world to the screen)
45 B 0 DefineHeader(2,GraphName);        (window where curves will go)
46 B 0 SetHeaderOn;
47 B 0 l:=0;
48 B 0 while not eof(tabfile2) do         (fill polygon array)
49 B 2 begin
50 B 2   i:=i+1;
51 B 2   readtabfile2,mearec2;
52 B 2   all[i]:=mearec2.sdbj
53 B 2   all[2i]:=mearec2.pobj;
54 B 2 end;
55 B 0 close(tabfile2);
```

```
56 B 1  
57 B 1  n:=1;  
58 B 1  s:=50; (generate spline with 50 points)  
59 B 1  spline(a,n,s(2,l),s(n-1,l),b,e);  
60 B 1  FindWorld(2,b,n,1,1.00); (make world 2 the right size)  
61 B 1  with world(2) do (flip the found world vertically)  
62 B 2  begin;  
63 B 2    cexp:=y1;  
64 B 2    y1:=y2;  
65 B 2    v2:=temp;  
66 B 2  end;  
67 I 1  SelectWindow(2); (select it and draw border)  
68 B 1  DrawBorder;  
69 B 1  
70 B 1  dx:=-8; (draw axis (inset from window edge))  
71 B 1  dy:=7;  
72 B 1  X1:=3;  
73 B 1  Y1:=5;  
74 B 1  X2:=25;  
75 B 1  Y2:=18;  
76 B 1  lines:=8;  
77 B 1  scale:=8;  
78 B 1  
79 B 1  SetLineStyle(1); (draw initial curve as dotted line)  
80 B 1  DrawAxis(dx,dy,X1,Y1,X2,Y2,lines,scale,false);  
81 B 1  DrawPolygon(2,n-1,7,2,8); (don't draw the endpoints)  
82 B 1  
83 B 1  SetLineStyle(0); (draw interpolated curve as solid line)  
84 B 1  DrawAxis(0,0,X1,Y1,X2,Y2,8,false);  
85 B 1  DrawPolygon(0,1,-n,0,0,0); (spline is not good on endpoints)  
86 B 1  
87 B 1  SelectWorld(1); (select outside window)  
88 B 1  SelectWindow(1);  
89 B 1  
90 B 1  DrawTextW(730,400,1,'?702 The data'); (print legend)  
91 B 1  DrawTextW(730,500,1,'.. The initial data values');  
92 B 1  DrawTextW(730,600,1,'... The interpolated values');  
93 B 1  
94 B 1  end;  
95 A 0  
96 A 0  BEGIN  
97 A 1  TextColor(15);  
98 A 1  TextBackground(1);  
99 A 1  ClrScr  
  
100 A 1  gotoxy(8,15);  
101 A 1  write('Enter table name to be plotted : ');  
102 A 1  readln(filename);  
103 A 1  filename := filename+'.dat';  
104 A 1  graphname := 'Plot of graph '+ filename + ', Pd (%) vs S/N (dB)';  
105 A 1  Assign(TabFile2,filename);  
106 A 1  ($I-)  
107 A 1  Reset(TabFile2);  
108 A 1  ($I+)  
109 A 1  Exists := (IOResult= 0);  
110 A 1  gotoxy(8,17);
```

Program Name: C:\HDV\GRAPHLOT.PAS
Date: Sunday, August 23, 1987
Time: 23:8

Page: 3

```
111 A 1 if not Exists then
112 A 1   write('File does not exist. Check the filename to be plotted')
113 A 1 else
114 A 2   Begin
115 A 2     InitGraphic;           (initialize the graphics system)
116 A 2     SplinDex;             (do the deco)
117 A 2     SelectWorld(1);
118 A 2     SelectWindow(1);
119 A 2     DrawTextW(720,500,1,'S/N (dB)');
120 A 2     DrawTextW(10,200,1,'Pd');
121 A 2     DrawTextW(10,250,1,'X');
122 A 2     DrawTextW(730,700,1,'Printout (Y/N) ? ');
123 A 3     repeat
124 A 3       ReadKbd,answer);
125 A 3     until answer IN ['y','Y','n','N'];
126 A 2     IF Answer IN ['Y','y']
127 A 2       THEN HardCopy(false,1)
128 A 2     ELSE LeaveGraphic     (leave the graphics system)
129 A 2     End;
130 A 1 END.
131 @ 0
132 @ 0
133 @ 0
```

Program Name: C:\HDV\MAIN\HELP.PAS
Date: Sunday, August 23, 1987
Time: 23:8

Page: 1

```
1 A 0 ( *****  
2 A 0 |  
3 A 0 | This procedure is used give the user a HELP facility in the main |  
4 A 0 | menu. It gives the user background to the software package and |  
5 A 0 | tells him of the various features of the system. The screen is |  
6 A 0 | displayed until any key is pressed and the user is then returned to |  
7 A 0 | menu.  
8 A 0 |  
9 A 0 |  
10 A 0 |  
11 A 0 |  
12 A 0 | Written By : Cyril Harari |  
13 A 0 |  
14 A 0 ( ***** )  
15 B 0 PROCEDURE Main_Menu_Help ;  
16 B 0 ( ***** )  
17 B 0  
18 B 0 BEGIN  
19 B 1 clrscr;  
20 B 1 gotoxy(1,3) ;  
21 B 1 writeln(' :10,'The "RADAR SIMULATION SYSTEM" allows the user to calculate ');  
22 B 1 writeln(' :10,'the radar range. The range equation includes the ');  
23 B 1 writeln(' :10,'signal-to-noise which is determined by specifying the ');  
24 B 1 writeln(' :10,'probability of detection (Pd) and false alarm (Pfa) for a ');  
25 B 1 writeln(' :10,'given radar.');26 B 1  
27 B 1  
28 B 1 writeln(' :10,'Pd and Pfa are calculated using the method described by ');  
29 B 1 writeln(' :10,'Hovanessian for a single pulse, and Blakes technique is ');  
30 B 1 writeln(' :10,'used to calculate Pd and Pfa for non-coherent multiple ');  
31 B 1 writeln(' :10,'pulse integration. ');  
32 B 1  
33 B 1 gotoxy(20,25);  
34 B 1 write' Hit any key to continue ... ');  
35 B 2 REPEAT UNTIL keypressed  
36 B 1 ( NO;
```

Program Name: C:\NDV\MULTIPLE.PAS
Date: Sunday, August 23, 1987
Time: 23:15

Page: 1

```
1 A 0 { *****  
2 A 0 {  
3 A 0 { This program is used to evaluate the probability of detection for  
4 A 0 { multiple pulse integration. The user is prompted for the required  
5 A 0 { Pfa (probability of false alarm), number of pulses to be integrated,  
6 A 0 { and the Pd (probability of detection). The Bessel function J0 and  
7 A 0 { I1 are also used here. At each value of S/N the value of Pd is  
8 A 0 { calculated until the specified value of Pd is obtained whereupon the  
9 A 0 { results are displayed. This program is not functioning correctly.  
10 A 0 { It appears the Blake's (I and Z) equations for multiple pulse  
11 A 0 { integration are incorrect.  
12 A 0 {  
13 A 0 {  
14 A 0 { Written By : Cyril Harari  
15 A 0 { ***** }  
16 A 0  
17 A 0 program Prob_Det_Multiple;  
18 A 0  
19 A 0 var deltaErf, derivErf, integrErf, endErf, fErf, stErf, e, tiErf, k2Erf,  
20 A 0 k3Erf, k4Erf, Pfa, PfaReq, PfaCalc, PfaCalc, PfaReq, kpfa, deltai, ul,  
21 A 0 derivI1, integrI1, I1, fI1, sI1, kI1, k2I1, k3I1, k4I1, deltaI, derivI,  
22 A 0 integrI, fI, sI, sI, kI, k2I, k3I, k4I, s, h, h,  
23 A 0 PfaCalcI, h, endErfI, PfaCalcI, sDval : real;  
24 A 0 NumStepsI, rkc1Erf, rkc2Erf, NumStepsI, rkcI1, rkc2I1,  
25 A 0 NumStepsI, rkcI1, rkc2I1 : Integer;  
26 A 0  
27 A 0 procedure Initialize;  
28 A 0  
29 A 0 { BCB/N  
30 A 0 I TextColor(15);  
31 A 0 I TextBackground(1);  
32 A 0 I ClrScr  
  
33 A 0 I gotoxy(8,1);  
34 A 0 I writeLn'Calculation of Kpfa and Pd - 'Multiple Pulse Integration';  
35 A 0 I gotoxy(8,2);  
36 A 0 I writeLn'*****';  
37 A 0 I xi:=0;  
38 A 0 I Pfa:=0;  
39 A 0 I PfaCalc:=0;  
40 A 0 I PfaReq:=0;  
41 A 0 I gotoxy(8,4);  
42 A 0 I write'Enter Required Pfa :';  
43 A 0 I readlnPfaReq;  
44 A 0 I gotoxy(8,5);  
45 A 0 I write'Enter number of pulses to be added :';  
46 A 0 I readlnI;  
47 A 0 I gotoxy(8,6);  
48 A 0 I write'What is the probability of detection ?';  
49 A 0 I readlnPdReq;  
50 A 0 I endErf:=0;  
51 A 0 I If PfaReq <= 1e-04 then endErf:=2.0;  
52 A 0 I If PfaReq <= 1e-05 then endErf:=3.0;  
53 A 0 I If PfaReq <= 1e-06 then endErf:=3.5;  
54 A 0 I If PfaReq <= 1e-07 then endErf:=3.75;  
55 A 0 I If PfaReq <= 1e-08 then endErf:=4.4;
```

```
56 B 1 If PfaReq <= 1e-12 then sdrErf:=4.95;  
57 B 1 End;  
58 A 0  
59 B 0 Procedure Derivative_Erf;  
60 B 0  
61 B 1 Begin  
62 B 1 x := stErf;  
63 B 1 derivErf := (2/sqrt(pi))*exp(-x*x);  
64 B 1 End;  
65 A 0  
66 B 0 Procedure R_K_Loop_Erf;  
67 B 0  
68 B 1 Begin  
69 B 1 For rkc2Erf := 1 To 4 Do  
70 B 2 Begin  
71 B 2 Derivative_Erf;  
72 B 3 Case rkc2Erf Of  
73 B 4 1: Begin  
74 B 4 k1Erf := IntgrErf;  
75 B 4 k1Erf := derivErf * k1Erf;  
76 B 4 IntgrErf := k1Erf * 2;  
77 B 4 stErf := stE * 2;  
78 B 4 End;  
79 B 4 2: Begin  
80 B 4 k2Erf := derivErf * deltaErf;  
81 B 4 IntgrErf := k1Erf + k2Erf / 2;  
82 B 4 End;  
83 B 4 3: Begin  
84 B 4 k3Erf := derivErf * deltaErf;  
85 B 4 IntgrErf := k1Erf + k3Erf;  
86 B 4 stErf := stErf + deltaErf / 2;  
87 B 4 End;  
88 B 4 4: Begin  
89 B 4 IntgrErf := k1Erf + (k1Erf + k2Erf * 2 + k3Erf * 2 + derivErf * deltaErf) / 6;  
90 B 4 End;  
91 B 3 End;  
92 B 2 End;  
93 B 1 End;  
94 A 0  
95 B 0 Procedure Runga_KuttaErf;  
96 B 0  
97 B 1 Begin  
98 B 1 For rkc1Erf := 1 To NumStepsErf Do  
99 B 1 R_K_Loop_Erf;  
100 B 1 End;  
101 A 0  
102 B 0 Procedure Derivative_I0;  
103 B 1 Begin  
104 B 1 x := stI0;  
105 B 1 derivI0 := 1/(2*pi)*exp(10*asin(x));  
106 B 1 End;  
107 A 0  
108 B 0 Procedure R_I_Loop_I0;  
109 B 0  
110 B 1 Begin
```

Program Name: C:\VHDV\MULTIPLE.PAS
Date: Sunday, August 23, 1987
Time: 23:15

Page: 3

```
111 B 1 For rkc218 := 1 To 4 Do
112 B 2 Begin
113 B 2 Derivative_10;
114 B 3 Case rkc218 Of
115 B 4 1: Begin
116 B 4 k410 := integr10;
117 B 4 k110 := deriv10 * delta10;
118 B 4 integr10 := k410 + k110 / 2;
119 B 4 st10 := st10 + delta10 / 2;
120 B 4 End;
121 B 4 2: Begin
122 B 4 k210 := deriv10 * delta10;
123 B 4 integr10 := k410 + k210 / 2;
124 B 4 End;
125 B 4 3: Begin
126 B 4 k310 := deriv10 * delta10;
127 B 4 integr10 := k410 + k310;
128 B 4 st10 := st10 + delta10 / 2;
129 B 4 End;
130 B 4 4: Begin
131 B 4 integr10 := k410 + (k110 + k210 * 2 + k310 * 2 + deriv10 * delta10) / 4;
132 B 4 End;
133 B 3 End;
134 B 2 End;
135 B 1 End;
136 A 0
137 B 0 Procedure Runge_Kutta10;
138 B 0
139 B 1 Begin
140 B 1 For rkc110 := 1 To NumSteps10 Do
141 B 1 R_K_Loop_10;
142 B 1
143 B 1 End;
144 A 0
145 B 0 Procedure Derivative_11;
146 B 1 Begin
147 B 1 x := st11;
148 B 1 deriv11 := 1/(2*pi)*((sin(x)*exp(11*(sin(x)))));
149 B 1 End;
150 A 0
151 B 0 Procedure R_K_Loop_11;
152 B 0
153 B 1 Begin
154 B 1 For rkc211 := 1 To 4 Do
155 B 2 Begin
156 B 2 Derivative_11;
157 B 3 Case rkc211 Of
158 B 4 1: Begin
159 B 4 k11 := integr11;
160 B 4 k111 := deriv11 * delta11;
161 B 4 integr11 := k11 + k111 / 2;
162 B 4 st11 := st11 + delta11 / 2;
163 B 4 End;
164 B 4 2: Begin
165 B 4 k211 := deriv11 * delta11;
```


Program Name: C:\NOV\MULTIPLE.PAS
Date: Sunday, August 23, 1987
Time: 23:15

Page: 5

```
-----  
221 A 1  sdB:=15;  
222 A 1  bt:=1e-36;  
223 A 2  repeat  
224 A 2    s:=exp(sdB*ln10)/20;  
225 A 2    gotoxy(8,15);  
226 A 2    write'B(number) = ', s(9:7);  
227 A 2  
228 A 2  (0 := s/2)                ( Procedure to calculate constants 10 )  
229 A 2  st10:=0;  
230 A 2  f10:=2*p1;  
231 A 2  NumSteps10:=50;  
232 A 2  delta10 := (f10 - st10) / NumSteps10;  
233 A 2  integr10 := 0;  
234 A 2  Runge_Kutta10;  
235 A 2  gotoxy(8,16);  
236 A 2  write'10('f,10:4:3,') = ',integr10(5:4);  
237 A 2  
238 A 2  I1:=s/2;                ( Procedure to calculate constants 11 )  
239 A 2  st11:=0;  
240 A 2  f11:=2*p1;  
241 A 2  NumSteps11:=50;  
242 A 2  delta11 := (f11 - st11) / NumSteps11;  
243 A 2  integr11 := 0;  
244 A 2  Runge_Kutta11;  
245 A 2  gotoxy(8,17);  
246 A 2  write'11('f,11:4:3,') = ',integr11(5:4);  
247 A 2  
248 A 2  ( procedure to calculate Pd with many pulses integrated )  
249 A 2  
250 A 2  if b>=1e-37 then b := exp((-s/2)*((1+s)*intgr(8)+(s*intgr11)))  
251 A 2    else b :=0;  
252 A 2  gotoxy(8,17);  
253 A 2  writeln'b = ',b);  
254 A 2  kpd := (ut-b)/sqrt(2*((4*(1+s)/p1)-(b*b))/M);  
255 A 2  gotoxy(8,18);  
256 A 2  write'kpd = ',kpd(9:7);  
257 A 2  stErf:=0;  
258 A 2  fErf:=kpd;  
259 A 2  NumStepsErf:=50;  
260 A 2  deltaErf := (fErf - stErf) / NumStepsErf;  
261 A 2  integrErf := 0;  
262 A 2  Runge_KuttaErf;  
263 A 2  PdCalc:=8.5-8.5*intgrErf;  
264 A 2  sdb := sdb + 1;  
265 A 2  gotoxy(8,19);  
266 A 2  writeln'sdb = ',sdb(4:2);  
267 A 2  gotoxy(8,20);  
268 A 2  writeln'PdCalc = ',PdCalc(9:7);  
269 A 2  gotoxy(8,21);  
270 A 2  write'H = ',H(3:1);  
271 A 2  
272 A 2  untill PdCalc >=PdReq;  
273 A 1  
274 A 1  gotoxy(8,22);  
275 A 1  writeln'Pd = ',PdCalc(5);
```

Program Name: C:\MOV\MULTIPLE.PAS
Date: Sunday, August 23, 1987
Time: 23:15

Page: 6

276 A 1 sdval := 281ns)/ln(8);
277 A 1 gotoxy(18,23);
278 A 1 writeln'sdB = ',sdval/2:4:2); (NOTE FACTOR OF 2 HERE !!!!!!!!)
279 A 1 gotoxy(25,23);
280 A 1 write'Hit any key to continue...';
281 A 2 repeat until keypressed;
282 A 1 END.

Program Name: C:\HDV\PSERF.PAS
Date: Sunday, August 23, 1987
Time: 23:23

Page: 1

```
1 A 0 ( .....  
2 A 0 ( .....  
3 A 0 | This program was written to evaluate the error function whic is used |  
4 A 0 | in the program for calculating the multiple pulse integration. The |  
5 A 0 | result is returned as integrErf. Again the 4th order Runge-Kutta |  
6 A 0 | integration technique is used. |  
7 A 0 | .....  
8 A 0 | .....  
9 A 0 | .....  
10 A 2 | .....  
11 A 0 | .....  
12 A 0 | .....  
13 A 0 | .....  
14 A 0 ( ..... )  
15 A 0 .....  
16 A 0 program Prob_Det_Multiple;  
17 A 0 .....  
18 A 0 var deltaErf, derivErf, integrErf, endErf, fErf, stErf, x, k1Erf, k2Erf,  
19 A 0 k3Erf, k4Erf, Pi, Rresult : real;  
20 A 0 NumStepsErf, rk1Erf, rk2Erf : integer;  
21 A 0 .....  
22 B 0 procedure Initializer;  
23 B 0 .....  
24 B 1 BEGIN  
25 B 1 .....  
26 B 1 TextColor(15);  
27 B 1 TextBackground(1);  
28 B 1 ClrScr  
29 B 1 gotoxy(5);  
30 B 1 writeln('!18,'Error function evaluation');  
31 B 1 writeln('!18,'.....');  
32 B 1 result:=0;  
33 B 1 x:=0;  
34 B 1 gotoxy(8,10);  
35 B 1 write('Enter value for x : ');  
36 B 1 readln(endErf);  
37 B 1 stErf:=0;  
38 B 1 fErf:=endErf;  
39 B 1 NumStepsErf:=100;  
40 B 1 deltaErf := (fErf - stErf) / NumStepsErf;  
41 B 1 integrErf := 0;  
42 B 1 End;  
43 A 0 .....  
44 B 0 Procedure Derivative_Erf;  
45 B 1 Begin  
46 B 1 x := stErf;  
47 B 1 derivErf := (2/sqrt(1-x*x));  
48 B 1 .....  
49 B 1 End;  
50 A 0 .....  
51 B 0 Procedure R_K_Loop_Erf;  
52 B 0 .....  
53 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )  
54 B 0 ( is called depends on the variable 'NumStepsErf'. )  
55 B 0
```

Program Name: C:\HDV\PMBERF.PAS
Date: Sunday, August 23, 1987
Time: 23:23

Page: 2

```
56 B 1 Begin
57 B 1 For krc2Erf := 1 To 4 Do
58 B 2 Begin
59 B 2 Derivative_Erf;
60 B 2
61 B 3 Case krc2Erf Of
62 B 4 1: Begin
63 B 4 k4Erf := IntegrErf;
64 B 4 k1Erf := derivErf * deltaErf;
65 B 4 integrErf := k4Erf + k1Erf / 2;
66 B 4 stErf := stErf * deltaErf / 2;
67 B 4 End;
68 B 4 2: Begin
69 B 4 k2Erf := derivErf * deltaErf;
70 B 4 integrErf := k4Erf + k2Erf / 2;
71 B 4 End;
72 B 4 3: Begin
73 B 4 k3Erf := derivErf * deltaErf;
74 B 4 integrErf := k4Erf + k3Erf;
75 B 4 stErf := stErf + deltaErf / 2;
76 B 4 End;
77 B 4 4: Begin
78 B 4 integrErf := k4Erf + (k1Erf + k2Erf * 2 + k3Erf * 2 + derivErf * deltaErf) / 6;
79 B 4 End;
80 B 2 End;
81 B 1 End;
82 A 0
83 A 0
84 B 0 Procedure Runqa_KattaErf;
85 B 0
86 B 1 Begin
87 B 1 For krc1Erf := 1 To NumStepsErf Do
88 B 1 R_K_Loop_Erf;
89 B 1
90 B 1 End;
91 A 0
92 A 0
93 A 1 Begin
94 A 1 Initialize;
95 A 1
96 A 1 Runqa_KattaErf;
97 A 1 ClrScr
98 A 1 gotoxy(30,1);
99 A 1 writeln('RESULT');
100 A 1 gotoxy(30,2);
101 A 1 writeln('-----');
102 A 1 gotoxy(5,10);
103 A 1 Pfa:=0.0-8.5*IntegrErf;
104 A 1 writeln('erf',endErf:4:3,') = ',integrErf:5:1);
105 A 1 gotoxy(5,12);
106 A 1 writeln('Pfa = ',Pfa:9);
107 A 1 gotoxy(25,25);
108 A 1 write('Hit any key to continue...');
109 A 2 repeat until keypressed;
110 A 1 End.
```

Program Name: C:\HDV\PD11.PAS
Date: Sunday, August 25, 1987
Time: 23:25

Page: 1

```
1 A 0 ( ..... )
2 A 0 |
3 A 0 | This program was used to evaluate the Bessel function I1 using the |
4 A 0 | integral function for the Bessel Function. The user is prompted for |
5 A 0 | the value to be calculated and the result is obtained. The |
6 A 0 | procedure is used extensively in determining the probability of |
7 A 0 | false alarm and probability of detection for both single and |
8 A 0 | multiple pulse integration. The 4th order Runge-Kutta technique |
9 A 0 | is used for the integration. The answer is returned as the value |
10 A 0 | I1BRI1. The function is specified in the procedure Derivative_I1. |
11 A 0 |
12 A 0 |                               Written By : Cyril Horari |
13 A 0 |
14 A 0 ( ..... )
15 A 0
16 A 0 Program Bessel_I1;
17 A 0
18 A 0 var deltaI, derivI, integrI, endI, fI, stI, x, kI1, k2I1,
19 A 0     k3I1, k4I1, Pfa, result : real;
20 A 0     NumSt I1, rkcI1, rkc2I1 : integer;
21 A 0
22 A 0 Const pf = 5.1415926535;
23 A 0
24 A 0 procedure Initialize;
25 A 0
26 A 0 | BEGIN
27 A 0 |
28 A 0 |   TextColor(15);
29 A 0 |   TextBackground(1);
30 A 0 |   ClrScr
31 A 0 |
32 A 0 |   gotoxy(5,5);
33 A 0 |   writeln' :10,'Evaluation of the I1(x) Bessel Function';
34 A 0 |   writeln' :10,'.....';
35 A 0 |   result:=0;
36 A 0 |   x:=0;
37 A 0 |   gotoxy(8,10);
38 A 0 |   write'Enter value for x : ';
39 A 0 |   readln(endI);
40 A 0 |   stI:=0;
41 A 0 |   fI:=2*pi;
42 A 0 |   NumStepsI:=100;
43 A 0 |   deltaI := (fI - stI) / NumStepsI;
44 A 0 |   integrI := 0;
45 A 0 | End;
46 A 0
47 A 0 | Procedure Derivative_I1;
48 A 0 | Begin
49 A 0 |   x := stI;
50 A 0 |   derivI := 1/(2*pi)*ln(x)*expend(1/sinx));
51 A 0 | End;
52 A 0
53 A 0 | Procedure R_K_Loop_I1;
54 A 0 |
55 A 0 | ( Inner 4th order Runge-Kutta loop. The number of times this routine )
```

Program Name: C:\HDV\PO11.PAS
Date: Sunday, August 23, 1987
Time: 23:28

Page: 2

```
56 B 0 ( is called depends on the variable 'RunSteps11'. )
57 B 0
58 B 1 Begin
59 B 1 For rkc211 := 1 To 4 Do
60 B 2 Begin
61 B 2 Derivative_11;
62 B 3 Case rkc211 Of
63 B 4 1: Begin
64 B 4 k411 := intgr11;
65 B 4 k111 := deriv11 & deltall;
66 B 4 intgr11 := k411 + k111 / 2;
67 B 4 st11 := st11 + deltall / 2;
68 B 4 End;
69 B 4 2: Begin
70 B 4 k211 := deriv11 & deltall;
71 B 4 intgr11 := k411 + k211 / 2;
72 B 4 End;
73 B 4 3: Begin
74 B 4 k311 := deriv11 & deltall;
75 B 4 intgr11 := k411 + k311;
76 B 4 st11 := st11 + deltall / 2;
77 B 4 End;
78 B 4 4: Begin
79 B 4 intgr11 := k411 + (k111 + k211 & 2 + k311 & 2 + deriv11 & deltall) / 6;
80 B 4 End;
81 B 3 End;
82 B 2 End;
83 B 1 End;
84 A 0
85 A 0
86 B 0 Procedure Runge_Kutta11;
87 B 0
88 B 1 Begin
89 B 1 For rkc111 := 1 To RunSteps11 Do
90 B 1 R_K_Loop_11;
91 B 1
92 B 1 End;
93 A 0
94 A 0
95 A 1 Begin
96 A 1 Initialize;
97 A 1
98 A 1 Runge_Kutta11;
99 A 1 ClrScr
100 A 1 gotoxy(38,1);
101 A 1 writeLn('RESULT');
102 A 1 gotoxy(38,2);
103 A 1 writeLn('=====');
104 A 1 gotoxy(5,10);
105 A 1 writeLn('11(' ,end11:4;3,') = ',intgr11:5:4);
106 A 1 gotoxy(25,25);
107 A 1 write'Hit any key to continue...';
108 A 2 repeat until keypressed;
109 A 1 End.
```


Program Name: C:\HDV\PROGRAMS\PAS
Date: Monday, August 24, 1997
Time: 0:0

Page:

```
56 B 1 k4 := 0;  
57 B 1  
58 R f End;  
59 A 0  
60 B 0 Procedure Derivative;  
61 B 1 Begin  
62 B 1 deriv := exp(sqrt(2 * s) * x * sinf);  
63 B 1 End;  
64 A 0  
65 B 0 Procedure R_K_Loop_lo;  
66 B 0  
67 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )  
68 B 0 is called depends on the variable 'steps'. )  
69 B 0  
70 B 1 Begin  
71 B 1 For rkc2lo := 1 To 4 Do  
72 B 2 Begin  
73 B 2 Derivative;  
74 B 3 Case rkc2lo Of  
75 B 4 1: Begin  
76 B 4 k4lo := integrlo;  
77 B 4 k1lo := derivlo + delta;  
78 B 4 integr1 := k1lo + k1lo / 2;  
79 B 4 f := f + delta / 2;  
80 B 4 End;  
81 B 4 2: Begin  
82 B 4 k2lo := derivlo + delta;  
83 B 4 integrlo := k4lo + k2lo / 2;  
84 B 4 End;  
85 B 4 3: Begin  
86 B 4 k3lo := derivlo + delta;  
87 B 4 integrlo := k4lo + k3lo;  
88 B 4 f := f + delta / 2;  
89 B 4 End;  
90 B 4 4: Begin  
91 B 4 integrlo := k4lo + (k1lo + k2lo + 2 + k3lo * 2 + derivlo + delta) / 6;  
92 B 4 End;  
93 B 3 End;  
94 B 2 End;  
95 B 1 End;  
96 A 0  
97 B 0 Procedure Runge_Kutta_lo;  
98 B 0  
99 B 1 Begin  
100 B 1 For rkc1lo := 1 To steps Do  
101 B 2 Begin  
102 B 2 R_K_Loop_lo;  
103 B 2 End;  
104 B 1 End;  
105 A 0  
106 B 0 Procedure Derivative;  
107 B 1 Begin  
108 B 1 Deriv := x * Bessel * exp((-x * x) - 2/s) / 2;  
109 B 1 End;  
110 A 0
```

Program Name: C:\NDV\FORINTG.PAS
Date: Monday, August 24, 1967
Time: 9:8

Page:

```
111 B 0 Procedure R_K_Loop;
112 B 0
113 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )
114 B 0 ( is called depends on the variable 'steps'. )
115 B 0
116 B 1 Begin
117 B 1 For rkc2 := 1 To 4 Do
118 B 2 Begin
119 B 2 Derivatives;
120 B 3 Case rkc2 Of
121 B 4 1: Begin
122 B 4 k4 := integr;
123 B 4 k1 := deriv k delta;
124 B 4 integr := k4 + k1 / 2;
125 B 4 x := x + delta / 2;
126 B 4 End;
127 B 4 2: Begin
128 B 4 k2 := deriv k delta;
129 B 4 integr := k4 + k2 / 2;
130 B 4 End;
131 B 4 3: Begin
132 B 4 k3 := deriv k delta;
133 B 4 integr := k4 + k3;
134 B 4 x := x + delta / 2;
135 B 4 End;
136 B 4 4: Begin
137 B 4 integr := k4 + (k1 + k2 + 2 + k3 + 2 + deriv k delta) / 6;
138 B 4 End;
139 B 3 End;
140 B 2 End;
141 B 1 End;
142 A 0
143 A 1 Begin
144 A 1 Initialize;
145 A 1 For rkc1 := 1 to steps do
146 A 2 Begin
147 A 2 k1lo := 0;
148 A 2 k2lo := 0;
149 A 2 k3lo := 0;
150 A 2 k4lo := 0;
151 A 2 integrlo := 0;
152 A 2 f := 0;
153 A 2 Runge_Kutta_Loop;
154 A 2 Bessel := integrlo / (2 + pi);
155 A 2 R_K_Loop;
156 A 2 gotoxy35,15;
157 A 2 Write'Busy... ', rkc1;
158 A 2 end;
159 A 1 gotoxy1,18;
160 A 1 ClrScr
161 A 1 gotoxy36,1;
162 A 1 WriteLn'RESULTS';
163 A 1 gotoxy36,2;
164 A 1 WriteLn'=====';
165 A 1 gotoxy5,18;
```

Program Name: C:\HDV\POCKINTG.PAS
Date: Monday, August 24, 1987
Time: 0:0

Page:

```
166 A 1 writeLn('For R/N = ', sdb:3:1, ' dB', '          S(ratio) = ', si:3:1);  
167 A 1 gotoxy(5,12);  
168 A 1 writeLn('For Pfa = ', Pfas:6, '          Et = ', endx:4:2, ' Volts');  
169 A 1 gotoxy(5,14);  
170 A 1 WriteLn('Pd = ', (1 - intgr) * 100:5:2, ' X ');  
171 A 1 gotoxy(25,25);  
172 A 1 write 'Hit any key to continue...';  
173 A 2 repeat until keypressed;  
174 A 1 End.
```

```
1 A B : ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2 A B :
3 A B : This procedure is used to open and read a file specified by the
4 A B : variable FileName. The values of the signal to noise ratio, the
5 A B : probability of false alarm and the probability of detection are then
6 A B : displayed in tabular format on the screen. The file is then closed,
7 A B : thus saving the file. Note that if the file does not exist an error
8 A B : condition will occur.
9 A B :
10 A B :
11 A B :
12 A B :
13 A B :
14 A B :
15 A B :
16 B B Procedure Pd_Table_Values;
17 B B Begin
18 B B ( First open the table file. )
19 B B
20 B B Assign(TabFile2,FileName);
21 B B Reset(TabFile2);
22 B B gotoxy(8,1);
23 B B Write'SiNoMin - Pd Table for Pfa = ',FileName, ' file.';
24 B B gotoxy(8,2);
25 B B WriteLn('#####');
26 B B WriteLn
27 B B
28 B B
29 B B
30 B B
31 B B
32 B B
33 B B
34 B B
35 B B
36 B B
37 B B
38 B B
39 B B
40 B B
41 B B
42 B B
43 B B
44 B B
45 B B
46 B B
47 B B
48 B B
49 B B
50 B B
51 B B
52 B B
53 B B
54 B B
55 B B
56 B B
57 B B
58 B B
59 B B
60 B B
61 B B
62 B B
63 B B
64 B B
65 B B
66 B B
67 B B
68 B B
69 B B
70 B B
71 B B
72 B B
73 B B
74 B B
75 B B
76 B B
77 B B
78 B B
79 B B
80 B B
81 B B
82 B B
83 B B
84 B B
85 B B
86 B B
87 B B
88 B B
89 B B
90 B B
91 B B
92 B B
93 B B
94 B B
95 B B
96 B B
97 B B
98 B B
99 B B
100 B B
101 B B
102 B B
103 B B
104 B B
105 B B
106 B B
107 B B
108 B B
109 B B
110 B B
111 B B
112 B B
113 B B
114 B B
115 B B
116 B B
117 B B
118 B B
119 B B
120 B B
121 B B
122 B B
123 B B
124 B B
125 B B
126 B B
127 B B
128 B B
129 B B
130 B B
131 B B
132 B B
133 B B
134 B B
135 B B
136 B B
137 B B
138 B B
139 B B
140 B B
141 B B
142 B B
143 B B
144 B B
145 B B
146 B B
147 B B
148 B B
149 B B
150 B B
151 B B
152 B B
153 B B
154 B B
155 B B
156 B B
157 B B
158 B B
159 B B
160 B B
161 B B
162 B B
163 B B
164 B B
165 B B
166 B B
167 B B
168 B B
169 B B
170 B B
171 B B
172 B B
173 B B
174 B B
175 B B
176 B B
177 B B
178 B B
179 B B
180 B B
181 B B
182 B B
183 B B
184 B B
185 B B
186 B B
187 B B
188 B B
189 B B
190 B B
191 B B
192 B B
193 B B
194 B B
195 B B
196 B B
197 B B
198 B B
199 B B
200 B B
201 B B
202 B B
203 B B
204 B B
205 B B
206 B B
207 B B
208 B B
209 B B
210 B B
211 B B
212 B B
213 B B
214 B B
215 B B
216 B B
217 B B
218 B B
219 B B
220 B B
221 B B
222 B B
223 B B
224 B B
225 B B
226 B B
227 B B
228 B B
229 B B
230 B B
231 B B
232 B B
233 B B
234 B B
235 B B
236 B B
237 B B
238 B B
239 B B
240 B B
241 B B
242 B B
243 B B
244 B B
245 B B
246 B B
247 B B
248 B B
249 B B
250 B B
251 B B
252 B B
253 B B
254 B B
255 B B
256 B B
257 B B
258 B B
259 B B
260 B B
261 B B
262 B B
263 B B
264 B B
265 B B
266 B B
267 B B
268 B B
269 B B
270 B B
271 B B
272 B B
273 B B
274 B B
275 B B
276 B B
277 B B
278 B B
279 B B
280 B B
281 B B
282 B B
283 B B
284 B B
285 B B
286 B B
287 B B
288 B B
289 B B
290 B B
291 B B
292 B B
293 B B
294 B B
295 B B
296 B B
297 B B
298 B B
299 B B
300 B B
301 B B
302 B B
303 B B
304 B B
305 B B
306 B B
307 B B
308 B B
309 B B
310 B B
311 B B
312 B B
313 B B
314 B B
315 B B
316 B B
317 B B
318 B B
319 B B
320 B B
321 B B
322 B B
323 B B
324 B B
325 B B
326 B B
327 B B
328 B B
329 B B
330 B B
331 B B
332 B B
333 B B
334 B B
335 B B
336 B B
337 B B
338 B B
339 B B
340 B B
341 B B
342 B B
343 B B
344 B B
345 B B
346 B B
347 B B
348 B B
349 B B
350 B B
351 B B
352 B B
353 B B
354 B B
355 B B
356 B B
357 B B
358 B B
359 B B
360 B B
361 B B
362 B B
363 B B
364 B B
365 B B
366 B B
367 B B
368 B B
369 B B
370 B B
371 B B
372 B B
373 B B
374 B B
375 B B
376 B B
377 B B
378 B B
379 B B
380 B B
381 B B
382 B B
383 B B
384 B B
385 B B
386 B B
387 B B
388 B B
389 B B
390 B B
391 B B
392 B B
393 B B
394 B B
395 B B
396 B B
397 B B
398 B B
399 B B
400 B B
401 B B
402 B B
403 B B
404 B B
405 B B
406 B B
407 B B
408 B B
409 B B
410 B B
411 B B
412 B B
413 B B
414 B B
415 B B
416 B B
417 B B
418 B B
419 B B
420 B B
421 B B
422 B B
423 B B
424 B B
425 B B
426 B B
427 B B
428 B B
429 B B
430 B B
431 B B
432 B B
433 B B
434 B B
435 B B
436 B B
437 B B
438 B B
439 B B
440 B B
441 B B
442 B B
443 B B
444 B B
445 B B
446 B B
447 B B
448 B B
449 B B
450 B B
451 B B
452 B B
453 B B
454 B B
455 B B
456 B B
457 B B
458 B B
459 B B
460 B B
461 B B
462 B B
463 B B
464 B B
465 B B
466 B B
467 B B
468 B B
469 B B
470 B B
471 B B
472 B B
473 B B
474 B B
475 B B
476 B B
477 B B
478 B B
479 B B
480 B B
481 B B
482 B B
483 B B
484 B B
485 B B
486 B B
487 B B
488 B B
489 B B
490 B B
491 B B
492 B B
493 B B
494 B B
495 B B
496 B B
497 B B
498 B B
499 B B
500 B B
501 B B
502 B B
503 B B
504 B B
505 B B
506 B B
507 B B
508 B B
509 B B
510 B B
511 B B
512 B B
513 B B
514 B B
515 B B
516 B B
517 B B
518 B B
519 B B
520 B B
521 B B
522 B B
523 B B
524 B B
525 B B
526 B B
527 B B
528 B B
529 B B
530 B B
531 B B
532 B B
533 B B
534 B B
535 B B
536 B B
537 B B
538 B B
539 B B
540 B B
541 B B
542 B B
543 B B
544 B B
545 B B
546 B B
547 B B
548 B B
549 B B
550 B B
551 B B
552 B B
553 B B
554 B B
555 B B
556 B B
557 B B
558 B B
559 B B
560 B B
561 B B
562 B B
563 B B
564 B B
565 B B
566 B B
567 B B
568 B B
569 B B
570 B B
571 B B
572 B B
573 B B
574 B B
575 B B
576 B B
577 B B
578 B B
579 B B
580 B B
581 B B
582 B B
583 B B
584 B B
585 B B
586 B B
587 B B
588 B B
589 B B
590 B B
591 B B
592 B B
593 B B
594 B B
595 B B
596 B B
597 B B
598 B B
599 B B
600 B B
601 B B
602 B B
603 B B
604 B B
605 B B
606 B B
607 B B
608 B B
609 B B
610 B B
611 B B
612 B B
613 B B
614 B B
615 B B
616 B B
617 B B
618 B B
619 B B
620 B B
621 B B
622 B B
623 B B
624 B B
625 B B
626 B B
627 B B
628 B B
629 B B
630 B B
631 B B
632 B B
633 B B
634 B B
635 B B
636 B B
637 B B
638 B B
639 B B
640 B B
641 B B
642 B B
643 B B
644 B B
645 B B
646 B B
647 B B
648 B B
649 B B
650 B B
651 B B
652 B B
653 B B
654 B B
655 B B
656 B B
657 B B
658 B B
659 B B
660 B B
661 B B
662 B B
663 B B
664 B B
665 B B
666 B B
667 B B
668 B B
669 B B
670 B B
671 B B
672 B B
673 B B
674 B B
675 B B
676 B B
677 B B
678 B B
679 B B
680 B B
681 B B
682 B B
683 B B
684 B B
685 B B
686 B B
687 B B
688 B B
689 B B
690 B B
691 B B
692 B B
693 B B
694 B B
695 B B
696 B B
697 B B
698 B B
699 B B
700 B B
701 B B
702 B B
703 B B
704 B B
705 B B
706 B B
707 B B
708 B B
709 B B
710 B B
711 B B
712 B B
713 B B
714 B B
715 B B
716 B B
717 B B
718 B B
719 B B
720 B B
721 B B
722 B B
723 B B
724 B B
725 B B
726 B B
727 B B
728 B B
729 B B
730 B B
731 B B
732 B B
733 B B
734 B B
735 B B
736 B B
737 B B
738 B B
739 B B
740 B B
741 B B
742 B B
743 B B
744 B B
745 B B
746 B B
747 B B
748 B B
749 B B
750 B B
751 B B
752 B B
753 B B
754 B B
755 B B
756 B B
757 B B
758 B B
759 B B
760 B B
761 B B
762 B B
763 B B
764 B B
765 B B
766 B B
767 B B
768 B B
769 B B
770 B B
771 B B
772 B B
773 B B
774 B B
775 B B
776 B B
777 B B
778 B B
779 B B
780 B B
781 B B
782 B B
783 B B
784 B B
785 B B
786 B B
787 B B
788 B B
789 B B
790 B B
791 B B
792 B B
793 B B
794 B B
795 B B
796 B B
797 B B
798 B B
799 B B
800 B B
801 B B
802 B B
803 B B
804 B B
805 B B
806 B B
807 B B
808 B B
809 B B
810 B B
811 B B
812 B B
813 B B
814 B B
815 B B
816 B B
817 B B
818 B B
819 B B
820 B B
821 B B
822 B B
823 B B
824 B B
825 B B
826 B B
827 B B
828 B B
829 B B
830 B B
831 B B
832 B B
833 B B
834 B B
835 B B
836 B B
837 B B
838 B B
839 B B
840 B B
841 B B
842 B B
843 B B
844 B B
845 B B
846 B B
847 B B
848 B B
849 B B
850 B B
851 B B
852 B B
853 B B
854 B B
855 B B
856 B B
857 B B
858 B B
859 B B
860 B B
861 B B
862 B B
863 B B
864 B B
865 B B
866 B B
867 B B
868 B B
869 B B
870 B B
871 B B
872 B B
873 B B
874 B B
875 B B
876 B B
877 B B
878 B B
879 B B
880 B B
881 B B
882 B B
883 B B
884 B B
885 B B
886 B B
887 B B
888 B B
889 B B
890 B B
891 B B
892 B B
893 B B
894 B B
895 B B
896 B B
897 B B
898 B B
899 B B
900 B B
901 B B
902 B B
903 B B
904 B B
905 B B
906 B B
907 B B
908 B B
909 B B
910 B B
911 B B
912 B B
913 B B
914 B B
915 B B
916 B B
917 B B
918 B B
919 B B
920 B B
921 B B
922 B B
923 B B
924 B B
925 B B
926 B B
927 B B
928 B B
929 B B
930 B B
931 B B
932 B B
933 B B
934 B B
935 B B
936 B B
937 B B
938 B B
939 B B
940 B B
941 B B
942 B B
943 B B
944 B B
945 B B
946 B B
947 B B
948 B B
949 B B
950 B B
951 B B
952 B B
953 B B
954 B B
955 B B
956 B B
957 B B
958 B B
959 B B
960 B B
961 B B
962 B B
963 B B
964 B B
965 B B
966 B B
967 B B
968 B B
969 B B
970 B B
971 B B
972 B B
973 B B
974 B B
975 B B
976 B B
977 B B
978 B B
979 B B
980 B B
981 B B
982 B B
983 B B
984 B B
985 B B
986 B B
987 B B
988 B B
989 B B
990 B B
991 B B
992 B B
993 B B
994 B B
995 B B
996 B B
997 B B
998 B B
999 B B
1000 B B
```

Program Name: C:\HDV\FP\FAFILE.PAS
Date: Monday, August 24, 1987
Time: 8:48

Page:

```
1 A 0 (#####.#####)#####  
2 A 0 #  
3 A 0 # The file for storing the values of probability of false alarm, #  
4 A 0 # probability of detection and signal-to-noise ratio is created here. #  
5 A 0 # The user is prompted for a table name. If it does not exist it is #  
6 A 0 # created, else the user is requested for another file name. Note #  
7 A 0 # that an ASCII file is also created to allow the user to type the #  
8 A 0 # file to see what it contains using the DOS command TYPE filename.ext #  
9 A 0 #  
10 A 0 #  
11 A 0 #  
12 A 0 # Written By : Yil Harari #  
13 A 0 #  
14 A 0 (#####)#####  
15 A 0  
16 B # Procedure #FAfile;  
17 B #  
19 B # Begin  
19 B # Repeat ( until the file name exists )  
20 B # ClrScr  
  
21 B # gotoxy(8,18);  
22 B # Write ' Enter table name : ' ;  
23 B # Readln(Filename);  
24 B # Filename2 := Filename + '.asc';  
25 B # Filename := Filename + '.dat';  
26 B # gotoxy(8,20); ( extension. )  
27 B # write('File in use is : ',Filename);  
28 B # Delay(500);  
29 B # Assign(TabFile2,Filename);  
30 B # Assign(TabFile4,Filename2);  
31 B # Rewrite(TabFile4);  
32 B # {#-} ( Now we check that the )  
33 B # ReSet(TabFile2); ( file exists. )  
34 B # {#-} ( If it does not exist, )  
35 B # Exists := (IOresult= 0); ( the user can create it. )  
36 B #  
37 B # If not exists then  
38 B # begin  
39 B # ClrScr  
  
40 B # gotoxy(8,20);  
41 B # Write'File does not exist. Would you like to create it ? (Y/N) ' ;  
42 B # ReadKbd,ch);  
43 B # If UpCase(ch) = 'Y' then  
44 B # begin  
45 B # Rewrite(TabFile2);  
46 B # Exists := True;  
47 B # End;  
48 B # end;  
49 B #  
50 B # Until Exists;  
51 B # end;
```

Program Name: C:\HDV\PL0T9D.PAS
Date: Monday, August 24, 1987
Time: 1:21

Page:

```
1 A B ( *****  
2 A B *  
3 A B * This program is used to generate the values of the probability of  
4 A B * detection for a single pulse given the signal to noise ratio and the  
5 A B * probability of false alarm. A range of S/N is requested and the  
6 A B * values are stored to a file created by the procedure in the file  
7 A B * PFAFILE.PAS. It saves both ASCII and Turbo Pascal files each with  
8 A B * ".asc" and ".dat" extension respectively. The program GRAPLOT.COM  
9 A B * can then be used to plot the data values in the file filename.dat.  
10 A B *  
11 A B *  
12 A B *  
13 A B *  
14 A B *  
15 A B *  
16 A B PROGRAM Prob_Detection_Curve_Plot;  
17 A B  
18 A B ($I variable.pas)  
19 A B ($I PfaFile.pas)  
20 A B ($I PdTable.pas)  
21 A B  
22 B B Procedure DerivativeLo;  
23 B B  
24 B B Begin  
25 B B   derivLo := exp(sqrt(2 * s) * x * sin(f));  
26 B B End;  
27 A B  
28 B B Procedure R_K_Loop_Lo;  
29 B B  
30 B B ( Inner 4th order Runge-Kutta loop. The number of times this routine )  
31 B B ( is called depends on the variable "steps". )  
32 B B  
33 B B Begin  
34 B B   For rkc2Lo := 1 To 4 Do  
35 B B     Begin  
36 B B       DerivativeLo;  
37 B B       Case rkc2Lo Of  
38 B B         1: Begin  
39 B B           k1Lo := integrLo;  
40 B B           k1Lo := derivLo * deltaLo;  
41 B B           integrLo := k1Lo * k1Lo / 2;  
42 B B           f := f + deltaLo / 2;  
43 B B         End;  
44 B B         2: Begin  
45 B B           k2Lo := derivLo * deltaLo;  
46 B B           integrLo := k1Lo * k2Lo / 2;  
47 B B         End;  
48 B B         3: Begin  
49 B B           k3Lo := derivLo * deltaLo;  
50 B B           integrLo := k1Lo * k3Lo;  
51 B B           f := f + deltaLo / 2;  
52 B B         End;  
53 B B         4: Begin  
54 B B           integrLo := k1Lo * (k1Lo + k2Lo * 2 + k3Lo * 2 + derivLo * deltaLo) / 6;  
55 B B         End;
```

Program Name: C:\HDV\PLDTPD.PAS
Date: Monday, August 24, 1987
Time: 11:24

Page:

```
1 A 0 ( ..... )
2 C 0 *
3 A 2 * This program is used to generate the values of the probability of
4 A 0 * detection for a single pulse given the signal to noise ratio and the
5 A 0 * probability of false alarm. A range of S/N is requested and the
6 A 0 * values are stored to a file created by the procedure in the
7 A 0 * PFAFILE.PAS. It saves both ASCII and turbin pascal files each with
8 A 0 * ".asc" and ".dat" extension respectively. The program GRAPLOT.COM
9 A 0 * can then be used to plot the data values in the file filename.dat.
10 A 0 *
11 A 0 *
12 A 0 *                               Written By : Cyril Harari
13 A 0 *
14 A 0 * ..... )
15 A 0
16 A 0 PROGRAM Prob_Detection_Curve_Plot;
17 A 0
18 A 0 ($I variable.pas)
19 A 0 ($I PfaFile.pas)
20 A 0 ($I P0Table.pas)
21 A 0
22 B 0 Procedure DerivativeLo;
23 B 0
24 B 1 Begin
25 B 1   derivLo := exp(sqrt(2 * sl * x * x * sin(f)));
26 B 1 End;
27 A 0
28 B 0 Procedure R_K_LoopLo;
29 B 0
30 B 0 { Inner 4th order Runge-Kutta loop. The number of times this routine }
31 B 0 { is called depends on the variable 'steps'. }
32 B 0
33 B 1 Begin
34 B 1   For rkc2Lo := 1 To 4 Do
35 B 2   Begin
36 B 2     DerivativeLo;
37 B 3   Case rkc2Lo Of
38 B 4     1: Begin
39 B 4       k1Lo := integrLo;
40 B 4       k1Lo := derivLo * deltaLo;
41 B 4       integrLo := k1Lo + k1Lo / 2;
42 B 4       f := f + deltaLo / 2;
43 B 4     End;
44 B 4     2: Begin
45 B 4       k2Lo := derivLo * deltaLo;
46 B 4       integrLo := k1Lo + k2Lo / 2;
47 B 4     End;
48 B 4     3: Begin
49 B 4       k3Lo := derivLo * deltaLo;
50 B 4       integrLo := k1Lo + k3Lo;
51 B 4       f := f + deltaLo / 2;
52 B 4     End;
53 B 4     4: Begin
54 B 4       integrLo := k1Lo + (k1Lo + k2Lo * 2 + k3Lo * 2 + derivLo * deltaLo) / 6;
55 B 4     End;
```

Program Name: C:\HDV\PLDTPD.PAS
Date: Monday, August 24, 1987
Time: 1:24

Page:

```
56 B 3 End;
57 B 2 End;
58 B 1 End;
59 A 0
60 B 0 Procedure Runge_Kutta_1o;
61 B 0
62 B 1 Begin
63 B 1 For rkc1o := 1 To stepslo Do
64 B 2 Begin
65 B 2 R_K_Loop_1o;
66 B 2 End;
67 B 1 End;
68 A 0
69 B 0 Procedure Derivative;
70 B 1 Begin
71 B 1 Deriv := x + Bessel * exp(((x + x) - 2*)/2!);
72 B 1 End;
73 A 0
74 B 0 Procedure R_K_Loop;
75 B 0
76 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )
77 B 0 ( is called depends on the variable 'steps'. )
78 B 0
79 B 1 Begin
80 B 1 For rkc2 := 1 To 4 Do
81 B 2 Begin
82 B 2 Derivative;
83 B 3 Case rkc2 Of
84 B 4 1: Begin
85 B 4 k4 := integr;
86 B 4 k1 := deriv * delta;
87 B 4 integr := k4 + k1 / 2;
88 B 4 x := x + delta / 2;
89 B 4 End;
90 B 4 2: Begin
91 B 4 k2 := deriv * delta;
92 B 4 integr := k4 + k2 / 2;
93 B 4 End;
94 B 4 3: Begin
95 B 4 k3 := deriv * delta;
96 B 4 integr := k4 + k3;
97 B 4 x := x + delta / 2;
98 B 4 End;
99 B 4 4: Begin
100 B 4 integr := k4 + (k1 + k2 * 2 + k3 * 2 + deriv * delta) / 6;
101 B 4 End;
102 B 3 End;
103 B 2 End;
104 B 1 End;
105 A 0
106 A 0 ( Program MAIN Body )
107 A 0
108 A 1 Begin
109 A 1 TextColor(15);
110 A 1 TextBackground(1);
```

Program Name: C:\HDV\PL079.PAS
Date: Monday, August 29, 1987
Time: 1:24

Page:

```
111 A 1 gotoxy(10,15);
112 A 1 PfaFile;
113 A 1 Clrscr;
114 A 1 gotoxy(10,15);
115 A 1 Write ('What is the probability of false alarm (Pfa) ? ');
116 A 1 Readln (Pfa);
117 A 1 Clrscr;
118 A 1 gotoxy(10,15);
119 A 1 Write ('What is the lower signal to noise ratio limit (db) ? ');
120 A 1 Readln (sdblower);
121 A 1 Clrscr;
122 A 1 gotoxy(10,15);
123 A 1 Write ('What is the upper signal to noise ratio limit (db) . ');
124 A 1 Readln (sdbupper);
125 A 1 Clrscr;
126 A 1 for sdb := sdblower to sdbupper do
127 A 2 begin
128 A 2   s:=exp((sdb*(ln(10)/10)));
129 A 2   ut := 0;
130 A 2   x := 0;
131 A 2   u:=x := sqrt(-2 * ln(Pfa));
132 A 2   f := 2 * pi;
133 A 2   steps := 500;
134 A 2   stepsize := 10;
135 A 2   delta := (f - ut) / steps;
136 A 2   delta := (endx - x) / steps;
137 A 2   intgrlo := 0;
138 A 2   intgr := 0;
139 A 2   k1 := 0;
140 A 2   k2 := 0;
141 A 2   k3 := 0;
142 A 2   k4 := 0;
143 A 2
144 A 2   for rkc1 := 1 to steps do
145 A 3 begin
146 A 3   k1fo := 0;
147 A 3   k2fo := 0;
148 A 3   k3fo := 0;
149 A 3   k4fo := 0;
150 A 3   intgrfo := 0;
151 A 3   f := 0;
152 A 3   Runge_Kutta_lo;
153 A 3   Bessel := intgrfo / (2 * pi);
154 A 3   R_K_Loop;
155 A 3   gotoxy(1,21);
156 A 3   Write('S/N ... ',sdb,' Count ... ',rkc1);
157 A 3   Prob := (1 - intgr) * 100;
158 A 3   end;
159 A 2
160 A 2 NSINo := sdb;
161 A 2 MSRatio := s;
162 A 2 NPfa := Pfa;
163 A 2 NEL := endx;
164 A 2 MProb := Prob;
165 A 2
```

Program Name: C:\HDV\PL0TFR.PAG
Date: Monday, August 24, 1987
Time: 11:24

Page:

```
166 A 2   with MesRec2 do
167 A 3     begin
168 A 3       sdb := MSINo;
169 A 3       s := MSRatio;
170 A 3       Pfa := NPfa;
171 A 3       Endz := NEL;
172 A 3       Prob := NProb;
173 A 3       Write(TabFile2, MesRec2);      ( write the record to the file.)
174 A 3       with mesrec3 do
175 A 4         begin
176 A 4           str(sdb:4, sdbAsc);
177 A 4           str(s:4, sAsc);
178 A 4           str(Pfa:6, PfaAsc);
179 A 4           str(Endz:4, EndzAsc);
180 A 4           str(Prob:3:2, ProbAsc);
181 A 4           write#(Tabfile4, sdbasc, ' ', vasc, ' ', pfaasc, ' ', endzasc, ' ', probasc, ' ');
182 A 4         end;
183 A 3       end;
184 A 2     end;
185 A 1   Close (TabFile2);
186 A 1   Close (Tabfile4);
187 A 1   ClrScr;
188 A 1   Pd_Table_Values;
189 A 1   End.
```

Program Name: C:\NOV\PROBDET.PAS
Date: Monday, August 24, 1987
Time: 1:24

```
1 A 0 ( *****  
2 A 0 *  
3 A 0 * This procedure is used to generate the values of the probability of *  
4 A 0 * detection for a single pulse given the signal to noise ratio and the *  
5 A 0 * probability of false alarm. The S/N is requested and the values are *  
6 A 0 * displayed on the screen. This procedure is invoked to calculate *  
7 A 0 * one value of Pd for a given Pfa and S/N ratio. It is up to the user *  
8 A 0 * to check the given data files to see whether the calculation that *  
9 A 0 * has been requested has not been performed previously. The user *  
10 A 0 * must refer to the graphs which have been plotted or look at the data *  
11 A 0 * files, extension 'asc'. *  
12 A 0 * *  
13 A 0 * Written By : Cyril Harari *  
14 A 0 * *  
15 A 0 ***** )  
16 A 0  
17 B 0 Procedure Initialize;  
18 B 0  
19 B 0 Begin  
20 B 1 TextColor(13);  
21 B 1 TextBackground(1);  
22 B 1 ClrScr;  
23 B 1 writeLn(' :20,'Determination of the Pd function');  
24 B 1 writeLn(' :20,'*****');  
25 B 1 gotoxy(1,6);  
26 B 1 Write ('What is the signal-to-noise ratio (dB) ? ');  
27 B 1 Readln (sdbval);  
28 B 1 s:=exp(sdbval*ln(10)/10);  
29 B 1 writeLn;  
30 B 1 Write ('What is the probability of false alarm (Pfa) ? ');  
31 B 1 Readln (Pfa);  
32 B 1 st := 0;  
33 B 1 x := 0;  
34 B 1 endx := sqrt(-2 * ln(Pfa));  
35 B 1 f := 2 * pi;  
36 B 1 steps := 500;  
37 B 1 stepsl0 := 10;  
38 B 1 deltaf0 := (f - st) / stepsl0;  
39 B 1 delta := (endx - x) / steps;  
40 B 1 Intgrf0 := 0;  
41 B 1 intgr := 0;  
42 B 1 k1 := 0;  
43 B 1 k2 := 0;  
44 B 1 k3 := 0;  
45 B 1 k4 := 0;  
46 B 1  
47 B 1 End;  
48 A 0  
49 B 0 Procedure Derivative0;  
50 B 0 Begin  
51 B 1 deriv0 := exp(sqrt(2 * pi * x * x * sdb(f)));  
52 B 1 End;  
53 A 0  
54 B 0 Procedure R_K_Loop_0;  
55 B 0
```

Program Name: C:\HDV\PROBDET.PAS
Date: Monday, August 24, 1987
Time: 1:21

Page: 2

```
56 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )  
57 B 0 ( is called depends on the variable 'steps'. )  
58 B 0  
59 B 1 Begin  
60 B 1 For rkc2lo := 1 To 4 Do  
61 B 2 Begin  
62 B 2 DerivativeLo;  
63 B 3 Case rkc2lo Of  
64 B 4 1: Begin  
65 B 4 k4lo := intgrlo;  
66 B 4 k1lo := derivlo + deltaLo;  
67 B 4 intgrlo := k4lo + k1lo / 2;  
68 B 4 f := f + deltaLo / 2;  
69 B 4 End;  
70 B 4 2: Begin  
71 B 4 k2lo := derivlo + deltaLo;  
72 B 4 intgrlo := k4lo + k2lo / 2;  
73 B 4 End;  
74 B 4 3: Begin  
75 B 4 k3lo := derivlo + deltaLo;  
76 B 4 intgrlo := k4lo + k3lo;  
77 B 4 f := f + deltaLo / 2;  
78 B 4 End;  
79 B 4 4: Begin  
80 B 4 intgrlo := k4lo + (k1lo + k2lo + 2 + k3lo + 2 + derivlo + deltaLo) / 6;  
81 B 4 End;  
82 B 3 End;  
83 B 2 End;  
84 B 1 End;  
85 A 0  
86 B 0 Procedure Runge_Kutta_lo;  
87 B 0  
88 B 1 Begin  
89 B 1 For rkc1lo := 1 To stepslo Do  
90 B 2 Begin  
91 B 2 R_K_Loop_lo;  
92 B 2 End;  
93 B 1 End;  
94 A 0  
95 B 0 Procedure Derivative;  
96 B 1 Begin  
97 B 1 Deriv := x + Bessel * exp((-x + x) - 2ts/2);  
98 B 1 End;  
99 A 0  
100 B 0 Procedure R_K_Loop;  
101 B 0  
102 B 0 ( Inner 4th order Runge-Kutta loop. The number of times this routine )  
103 B 0 ( is called depends on the variable 'steps'. )  
104 B 0  
105 B 1 Begin  
106 B 1 For rkc2 := 1 To 4 Do  
107 B 2 Begin  
108 B 2 Derivatives;  
109 B 3 Case rkc2 Of  
110 B 4 1: Begin
```


Program Name: C:\HDV\PROBDET.PAS
Date: Monday, August 24, 1987
Time: 1:24

Page: 4

166 A 0

Program Name: C:\MOVIRADAR.PAS
Date: Monday, August 24, 1987
Time: 1:25

Page: 1

```
1 A 0 (*****  
2 A 0 *  
3 A 0 * This program is the main procedure. It invokes the various options *  
4 A 0 * as requested by the user. These include calculation of the radar *  
5 A 0 * range, generating the Pd, Pfa curves for single and multiple pulse *  
6 A 0 * integration, and also plotting these curves. *  
7 A 0 * *  
8 A 0 * *  
9 A 0 * *  
10 A 0 * *  
11 A 0 * *  
12 A 0 * *  
13 A 0 * Written By : Cyril Harari *  
14 A 0 (*****  
15 A 0  
16 A 0 PROGRAM radar;  
17 A 0  
18 A 0 ($I Variable.pas )  
19 A 0 ($I GetSelec.pas)  
20 A 0 ($I MainHelp.pas)  
21 A 0 ($I Range.pas)  
22 A 0 ($I Power.pas)  
23 A 0  
24 A 0 ( The procedures included below are used for input of the Theta-Gain Table )  
25 A 0  
26 A 0 ($I GainThet.pas)  
27 A 0 ($I DispVal.pas)  
28 A 0 ($I TableSel.pas)  
29 A 0  
30 A 0 ( The procedure included below is used to calculate the threshold voltage )  
31 A 0 ( given a required probability of false alarm and also to determine the )  
32 A 0 ( probability of detection for a single pulse, Hovnessian method. )  
33 A 0  
34 A 0 ($I ProbDet.pas)  
35 A 0  
36 A 0 ( The procedure included below is used to determine the Bessel integral )  
37 A 0 ( using the function approximation: )  
38 A 0  
39 A 0 ($I Bessel.pas)  
40 A 0  
41 A 0 ( main program )  
42 A 1 BEGIN  
43 A 2 REPEAT  
44 A 2 TextColor(15);  
45 A 2 TextBackground(1);  
46 A 2 get_selection ;  
47 A 2 REPEAT  
48 A 3 read(kbd,op);  
49 A 3 UNTIL op IN options;  
50 A 2 IF (op <> '9')  
51 A 2 THEN  
52 A 3 BEGIN  
53 A 4 CASE op OF  
54 A 5 '1' : BEGIN  
55 A 5 Range;
```

Program Name: C:\HOV\RADAR.PAS
Date: Monday, August 24, 1987
Time: 1:25

Pages 2

```
-----  
56 A S      END;  
57 A S      '2' : BEGIN  
58 A S          clrscr;  
59 A S          Table_Select;  
60 A S          Main_Theta_Input;  
61 A S          Display_Values;  
62 A S      END;  
63 A S      '3' : BEGIN  
64 A S          Prob_Detection;  
65 A S      END;  
66 A S      '4' : BEGIN  
67 A S          END;  
68 A S      '5' : BEGIN  
69 A S          Power;  
70 A S          END;  
71 A S      '6' : BEGIN  
72 A S          Bessel_Integral;  
73 A S          Runge_Kutta;  
74 A S          Bessel_Integral_Results;  
75 A S          END;  
76 A S      '8' : BEGIN  
77 A S          Main_Menu_Help;  
78 A S          END;  
79 A S      END;  
80 A S      END;  
81 A S      UNTIL op IN ('9');  
82 A S      clrscr;  
83 A S      END.
```

Program Name: C:\HDV\ RANGE.PAS
Date: Monday, August 24, 1987
Time: 1:25

Page: 1

```
1 A 0 ( *****  
2 A 0 *  
3 A 0 * This procedure is used to obtain the necessary values so that the *  
4 A 0 * range equation can be evaluated. The help file is also included *  
5 A 0 * giving the user further information about the selected option. *  
6 A 0 *  
7 A 0 *  
8 A 0 *  
9 A 0 *  
10 A 0 *  
11 A 0 *  
12 A 0 * Written By : Cyril Naras *  
13 A 0 *  
14 A 0 ***** )  
15 A 0  
16 0 PROCEDURE Range;  
17 0  
18 0 { The procedure that follows is used to obtain the relevant inputs  
19 0 so that the radar range(simple formula) can be evaluated.  
20 0  
21 0 Definition of variables:-  
22 0  
23 0 Pr = received signal power (at antenna terminals)  
24 0 Pt = transmitted signal power (at antenna terminals)  
25 0 Gt = transmitting antenna power gain (NOTE:- isotropic antenna G = 1)  
26 0 Gr = receiving antenna power gain  
27 0  $\sigma$  = radar cross section  
28 0 Lambda = wavelength  
29 0 Ft = pattern-propagation factor for the transmitting-antenna-to-target path.  
30 0 Ft = pattern-propagation factor for target-to-receiving-antenna path.  
31 0 R = radar-to-target distance (range)  
32 0 dBa = the voltage corresponding to 1 mW into 600 ohms load(0.707 volts rms).  
33 0 output power  
34 0 dB = 10 log  $\frac{\text{output power (dB)}}{\text{input power}}$  P = 10 log (1000xP)  
35 0 dBa  
36 0  
37 0 output power (dB) ln (a)  
38 0 dB = 20 log  $\frac{\text{output power (dB)}}{\text{input power (dB)}}$  log a =  $\frac{\ln (a)}{\ln (10)}$   
39 0 ln (10)  
40 0  
41 0 Tk = temperature (xK) = 290xK  
42 0 -23  
43 0 k = Boltzmann's constant (1.38 x 10-23 J/deg)  
44 0 Bn = noise bandwidth  
45 0 Fn = noise figure of a receiver  
46 0  
47 0 Typical Values to be input : Pt=150kW  
48 0 Freq=10GHz  
49 0 St=Gr=30dB  
50 0 Ft=Fr=1  
51 0 Bn=10Hz  
52 0 L=0dB  
53 0 SINAFn=12dB  
54 0  $\sigma$ =1mJ  
55 0
```

Program Name: C:\HOW\RANGE.PAS
Date: Monday, August 24, 1987
Time: 1:25

Page: 2

```
-----  
56 B 1 BEGIN  
57 B 1 ( Pt:=0;  
58 B 1 Gr:=0;  
59 B 1 St:=0;  
60 B 1 Signal:=0;  
61 B 1 Lambda:=0;  
62 B 1 Freq:=0;  
63 B 1 Ft:=0;  
64 B 1 Fr:=0;  
65 B 1 SInMin:=0;  
66 B 1 Ss:=0;  
67 B 1 Bs:=0;  
68 B 1 Ls:=0;  
69 B 1 Rt:=0;  
70 B 1 R_kar:=0;  
71 B 1 TextColor(15);  
72 B 1 TextBackground(1);  
73 B 1 clrscr;  
74 B 1 gotoxy(1,2);  
75 B 1 write(' :10,'PARAMETER INPUT FOR CALCULATION OF THE RADAR RANGE');  
76 B 1 gotoxy(1,3);  
77 B 1 write(' :10,'-----');  
78 B 1 gotoxy(1,5);  
79 B 1 write('Do you want some help ? (Y/N)');  
80 C 1 read(kbd,ch);  
81 B 1 gotoxy(1,5);  
82 B 1 clrEOL;  
83 B 1 if ch IN ('Y','N')  
84 B 2 then begin  
85 B 2 gotoxy(1,7);  
86 B 2 write(' :10,'The range equation can be evaluated here. Typical inputs ');  
87 B 2 write(' :10,'include transmitted power, transmitting and receiving power ');  
88 B 2 write(' :10,'gain, target cross section, pattern propagation factor (PPF)');  
89 B 2 write(' :10,'for the transmitting-antenna-to-target path, PPF for the ');  
90 B 2 write(' :10,'target-to-receiving-antenna path, noise bandwidth of the ');  
91 B 2 write(' :10,'receiver predetection filter, the generalized loss factors, ');  
92 B 2 write(' :10,'signal-to-noise ratio, and the frequency or wavelength. ');  
93 B 2 write(' ');  
94 B 2 gotoxy(28,23);  
95 B 2 write(' Hit any key to continue ... ');  
96 B 3 REPEAT UNTIL keypressed  
97 B 2 end;  
98 B 1 for count := 7 to 1 do  
99 B 2 begin  
100 B 2 gotoxy(1,count);  
101 B 2 clrEOL;  
102 B 2 end;  
103 B 1 gotoxy(1,5);  
104 B 1 write('Transmitted signal power (Pt) (Watts) ? ');  
105 B 1 readln(Pt);  
106 B 1 write('Transmitting antenna power gain (Gt) (dB) ? ');  
107 B 1 readln(Gt_dB);  
108 B 1 Gt:=exp(Gt_dB*ln(10)/10);  
109 B 1 write('Receiving antenna gain (Gr) (dB) ? ');  
110 B 1 readln(Gr_dB);
```

Program Name: C:\HOW\RANGE.PAS
Date: Monday, August 24, 1987
Time: 1:25

Page: 3

```
111 B 1 Cr:=exp(i*OrdB*is(i0)/i0);
112 B 1 write('Radar target cross section (e) (a) ? ');
113 B 1 readln(Signal);
114 B 1 write('PPF for the transmitting-antenna-to-target path (Ft) ? ');
115 B 1 readln(Ft);
116 B 1 write('PPF for the target-to-receiving-antenna path (Fr) ? ');
117 B 1 readln(Fr);
118 B 1 write('Noise bandwidth of the receiver predetection filter (Bn) (Hertz) ? ');
119 B 1 readln(Bn);
120 B 1 write('Value for the generalized loss factors (L) (dB) ? ');
121 B 1 readln(LdB);
122 B 1 L:=exp(LdB*is(i0)/i0);
123 B 1 write('Signal-to-noise ratio (S/N)ain (dB) ? ');
124 B 1 readln(SiNoHindB);
125 B 1 SiNoMin:=exp(SiNoHindB*is(i0)/20);
126 B 1 write('Do you have the frequency or the wavelength (F or W) ? ');
127 B 1 readln(Freq_wave);
128 B 1 Freq_wave:=0;Case(Freq_wave);
129 B 1 if Freq_wave = 'F' then
130 B 2 begin
131 B 2 write('What is the frequency of transmission (Freq) (Hertz) ? ');
132 B 2 readln(Freq);
133 B 2 Lambda:=3000/Freq;
134 B 2 end
135 B 1 else
136 B 2 begin
137 B 2 write('Enter the wavelength (Lambda) (metres) ? ');
138 B 2 readln(Lambda);
139 B 2 write(L);
140 B 2 end;
141 B 1 R:=sqrt(sqrt((Pt*Gt*Gr+Signal*sq(Lambda)*sqr(Ft)*sqr(Fr))/(16*pi*is(pi)*SiNoMin*k*k*Bw*L));
142 B 1 R_k:=R/1000;
143 B 1 ClrScr;
144 B 1 gotoxy(1,31);
145 B 1 write(' :30, 'RESULTS');
146 B 1 write(' :30, "=====');
147 B 1 gotoxy(10,8);
148 B 1 write('For 1-');
149 B 1 gotoxy(10,8);
150 B 1 write('Pt = ',Pt:5:1, ' watts');
151 B 1 gotoxy(40,8);
152 B 1 write('Gt = ',Gt:8:5:1);
153 B 1 gotoxy(10,9);
154 B 1 write('Gr = ',Gr:8:5:1);
155 B 1 gotoxy(40,9);
156 B 1 write('e = ',Signal:1, ' m');
157 B 1 gotoxy(5,10);
158 B 1 write('Lambda = ',Lambda:4:1, ' metres');
159 B 1 gotoxy(40,10);
160 B 1 write('Ft = ',Ft:3:1);
161 B 1 gotoxy(10,11);
162 B 1 write('Fr = ',Fr:3:1);
163 B 1 gotoxy(40,11);
164 B 1 write('SiNoMin = ',SiNoHindB:5:1, ' dB');
165 B 1 gotoxy(10,12);
```


Program Name: C:\HDV\TABLESEL.PAS
Date: Monday, August 24, 1987
Time: 1:25

Page: 1

```
1 0 0 Procedure Table_Select;
2 0 0
3 0 1 Begin
4 0 2 Repeat      ( until the file name exists )
5 0 2   ClrScr;
6 0 2   gotoxy(10,10);
7 0 2   Write(' Enter table name : ');
8 0 2   Readln(Filename);
9 0 2   Filename := Filename + '.dat';          ( table files have a *.dat )
10 0 2  gotoxy(10,20);                          ( extension. )
11 0 2  write 'File in use is : ',FileName;
12 0 2  Delay(1500);
13 0 2  Assign (TabFile,FileName);
14 0 2
15 0 2  (SI-)                                     ( Now we check that the )
16 0 2  Reset(TabFile);                          ( file exists. )
17 0 2  (I+)                                     ( If it does not exist, )
18 0 2  Exists := (IOresult = 0);                ( the user can create it. )
19 0 2
20 0 2  If not exists then
21 0 3  begin
22 0 3  ClrScr;
23 0 3  gotoxy(10,20);
24 0 3  Write('File does not exist. Would you like to create it ? (Y/N) ');
25 0 3  Read(KbD,ch);
26 0 3  If UpCase(ch) = 'Y' then
27 0 4  begin
28 0 4  ReWrite(TabFile);
29 0 4  Exists := True;
30 0 4  End;
31 0 3  end;
32 0 2
33 0 2  Until Exists;
34 0 1 end;
```

Program Name: C:\MOV\VARIBLE.PAS
Date: Monday, August 24, 1987
Time: 1:26

Page: 1

```
1 A 0 ( *****  
2 A 0 *  
3 A 0 * This file has all the global variables which are used in all the  
4 A 0 * procedures. It includes TYPE, String, variable, constant and label  
5 A 0 * definitions.  
6 A 0 *  
7 A 0 *  
8 A 0 *  
9 A 0 *  
10 A 0 *  
11 A 0 *  
12 A 0 * Written By : Dyril Harari  
13 A 0 *  
14 A 0 ( ***** )  
15 A 0  
16 A 0 TYPE  
17 A 0 txt = text;  
18 A 0  
19 A 0 RecPfa = RECORD  
20 A 0 sdb : Integer;  
21 A 0 s, Pfa, ends, Prob : Real;  
22 A 0 END;  
23 A 0  
24 A 0 Rec = RECORD  
25 A 0 Theta : Integer;  
26 A 0 GainR, GainT : Real;  
27 A 0 END;  
28 A 0  
29 A 0 strPfa = string [10];  
30 A 0 RecPfaAsc = RECORD  
31 A 0 sdbAsc, sAsc, PfaAsc, endsAsc, ProbAsc : strPfa;  
32 A 0 END;  
33 A 0  
34 A 0 VAR d, x, Pr, Pt, Gr, R, igma, Gt, Ac, Lambda, Pr_db, Pr_dbc, Freq,  
35 A 0 Ft, Fr, Prob, Pfa, S, dn, L, R_max, R_kc, In, temp, (a,) Ul, Nrus, p,  
36 A 0 result, res, stt, fin, stop, SINMin, SdB, GrdB, SINMinD, LdB, x2,  
37 A 0 integr1, deriv1a, k1a, k2a, k3a, k4a, inter, deriv, k1, k2, k3, k4,  
38 A 0 st, f, delta, s2, Et, ends, Bessel, end_st, MQaj,2, MQajT, GainR,  
39 A 0 GainT, delta1a, SNumber, MSRatio, MEI, MPfa, MProb, sdbval : real;  
40 A 0 ch, op, freq_wave, op1, op3, par, answer : char;  
41 A 0 ends, odd, periodic, limits_smpc, Exisit : boolean;  
42 A 0 (*****  
43 A 0 rule_string : string [50];  
44 A 0 TabFile : File of Rec;  
45 A 0 NewRec : Rec;  
46 A 0 TabFile2 : File of RecPfa;  
47 A 0 NewRec2 : RecPfa;  
48 A 0 TabFile3 : File of RecPfaAsc;  
49 A 0 TabFile4 : txt;  
50 A 0 NewRec3 : RecPfaAsc;  
51 A 0 HTheta, Theta, (G,) steps, rk1, rk2, steps1a, rk1a, rk2a, NumSteps,  
52 A 0 count, value, MSINa, sdblower, sdbupper, sdb : Integer;  
53 A 0 FileName, FileName2 : string [14];  
54 A 0 GraphName : string [50];  
55 A 0 vx, vs : array [1..13] of real;
```

Program Name: C:\HDV\ARIABLE.PAS
Date: Monday, August 24, 1987
Time: 1:26

Page: 2

```
-----  
56 A 0 FuncKey : Boolean;  
57 A 0 label loop_1, loop_2;  
58 A 0  
59 A 0 CONST Tk=298;  
60 A 0 options : set of char = ('1','2','3','4','5','6','7','8','9');  
61 A 0 maxallowed - 1E-9;  
62 A 0 k=1.38e-23;  
63 A 0 pi=3.1415926535;
```

APPENDIX F

Probability of detection versus signal to noise ratio (Hovnessian [3])

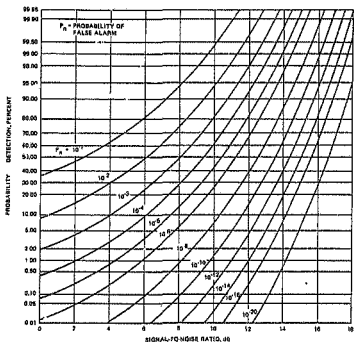
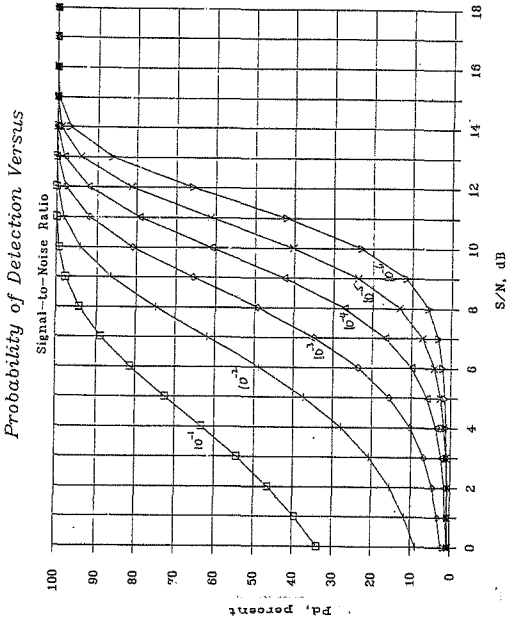


Figure 6-3. Probability of detection versus signal-to-noise ratio.

APPENDIX G

Examples of graphs plotted for varying S/N and probability of false alarm for single pulse integration



APPENDIX H A typical coverage diagram showing the radar vertical coverage using a cosecant-squared beam

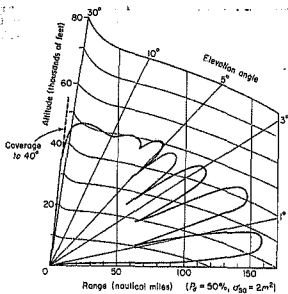


Figure 8.5 AN/UPS-1 with extended vertical coverage using cosecant-squared beam.

APPENDIX I

Example of a graph Blake [1 and 2] of the required signal-to noise ratio for a linear detector as a function of number of pulses integrated, for 0.1 probability of detection, calculated for a nonfluctuating signal for five values of false-alarm probability (P_{fa})

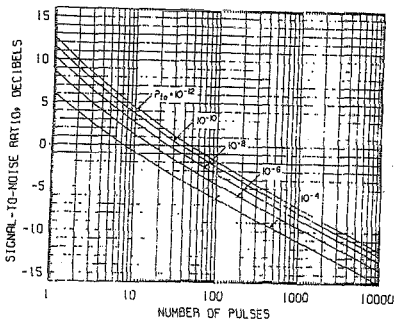



Fig. 5a - Required signal-to-noise ratio (visibility factor) for a linear detector as a function of number of pulses integrated, for 0.1 probability of detection, calculated for a nonfluctuating signal for five values of false-alarm probability (P_{fa}). (Note: This figure also appears in a larger size in an appendix at the end of the report.)



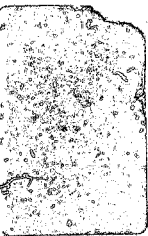
APPENDIX J Flow Chart of the software package

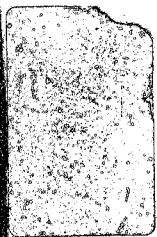
REFERENCES

1. Blake L. V., "A Guide to Basic Pulse-Radar Maximum-Range Calculation Part 1: Equations, Definitions, and Aids to Calculations - Naval Research Laboratory, Washington, D.C., December 1969.

Blake L. V., "A Guide to Basic Pulse-Radar Maximum-Range Calculation Part 2: Derivations of Equations, Bases of Graphs, and Additional Explanations" - Naval Research Laboratory, Washington, D.C., December 1969.
3. Hovanesian S. A., "Radar System Design and Analysis" - Artech House, Inc. Washington, 1984.
4. North D. O., "An Analysis of the Factors which Determine Signal/Noise Discrimination in Pulsed-Carrier Systems" - Proc. IEEE 51 (No. 7), 1016-1027, July 1963 (originally published as RCA Laboratories Report PTR-6C, June 1943).
5. Marcum J. I., "A Statistical Theory of Target Detection by Pulsed Radars - Johns Hopkins University Applied Physics Laboratory Report TG-451, July 1962, and Supplement, TG-451A, September 1964.
6. Spiegel M. R., "Mathematical Handbook of Formulas and Tables" - Schaum's Outline Series, McGraw-Hill, New York.
7. Freeman J. J., "Principles of Noise" - John Wiley & Sons, Inc. London, 1958.
8. Skolnik M. I. "Introduction to Radar Systems" - McGraw Hill International Book Company, 1983.
9. Bachman C. G., "Radar Targets" - Lexington Books, 1982.
10. Barton D. K., "Radar System Analysis" - Prentice-Hall, International, Inc., 1965.
11. Brookner E., "Radar Technology" - Artech House, Inc., 1977.

12. Rusnak I., "Search Radar Evaluation by the Normalised Cumulative Probability of Detection Curves" - IEEE Transactions on Aerospace and Electronic Systems Vol. AES-22, No. 4, July 1986.
13. Nathanson F. E., "Radar Design Principles - Signal Processing and the Environment" - McGraw-Hill Book Company, Inc., 1969.
14. Knott E. F., Shaffer J. F. and Tuley M. T., "Radar Cross Section - Its Prediction, Measurement and Reduction" - Artech House, Inc, Washington, 1985.
15. Skolnik M. I., "Radar Handbook" - McGraw-Hill Book Company, Inc., 1970.
16. Dorn, W. S. and McCracken D. D., "Numerical Methods with FORTRAN, IV Case Studies" - Wiley International, 1972 (Canada).
17. Rice, S. O., "Mathematical Analysis of Random Noise" - The Bell System Technical Journal, Volume XXIII, July 1944, pages 282 to 336.
18. Rice, S. O., "Mathematical Analysis of Random Noise (Concluded)" - The Bell System Technical Journal, Volume XXIV, January 1945, pages 46 to 157.







Author Harari Cyril Zaki

Name of thesis A Pc-based Radar Design Package. 1987

PUBLISHER:

University of the Witwatersrand, Johannesburg

©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.