Report No. FAA-RD-71-70

INVESTIGATION OF VIDEO PROCESSING CONCEPTS FOR ARTS III BASIC RADAR BEACON TRACKING LEVEL SYSTEM

P.E. Steichen and M.W. Wold
Defense Systems Division
UNIVAC
St. Paul, Minn. 55101

transpo 72
U.S. International Transportation Exposition
Dulles International Airport
Washington, D.C.
May 27-June 4, 1972

OCTOBER 1970

FINAL REPORT

Availability is unlimited. Document may be released to the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151, for sale to the public.

Prepared for

DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20591

150

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-71-70	-	-

4. Title and Subtitle	5. Report Date
INVESTIGATION OF VIDEO PROCESSING CONCEPTS FOR ARTS III BASIC RADAR BEACON TRACKING LEVEL SYSTEM	October 1970
	6. Performing Organization Code: -
7. Author(s): P.E. Steichen, M.W. Wold	8. Performing Organization Report No.: PX 6367
9. Performing Organization Name and Address: UNIVAC Defense Systems Division St. Paul, Minn. 55101	10. Work Unit No.: Project No. 19180
	11. Contract or Grant No.: DOT FA70WA-2289
12. Sponsoring Agency Name and Address: Federal Aviation Administration System Research & Development Service Washington, D.C. 20590	13. Type of Report & Period Covered: Final Report
	14. Sponsoring Agency Code: -

15. Supplementary Notes

16. Abstract

The report is an investigation to determine the role of a general purpose digital computer in the radar video processing area of ARTS III. Radar video processing is divided by functions into the following areas: Video Selection, Video Quantization, Target Detection, Position Estimation, and Radar/Beacon Target Report Correlation. Numerous alternatives for allocating these functions between a special purpose hardware device and a general purpose digital computer are explored to determine corresponding computer time and core requirements. With the exception of Radar/Beacon Target Report Correlation, the analysis explores only radar processing requirements. The algorithms and processing concepts considered are for the most part proven techniques. Performance evaluations are provided in the appendices for the few algorithms which have been modified to facilitate computer implementation. Within the appendices are a Markov Chain analysis of a quantizer threshold control system, a Monte Carlo simulation approach to the evaluation of target detection, and a statistical formulation of a particular target detection method.

17. Key Words	18. Distribution Statement
Radar Video Processing Beacon Tracking Level System Radar Target Detector Sliding Window Detection	Availability is unlimited. Document may be released to the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151, for sale to the public.

19. Security Classif. (of this rpt)	20. Security Classif. (of this pg)	21. No. of Pages	22. Price
Unclassified	Unclassified	150	$3.00 PC .95 MF

"This report has been prepared by UNIVAC for the Systems Research and Development Service, Federal Aviation Administration, under Contract No. DOT FA70WA-2289. The contents of this report reflect the views of the contractor, who is responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official view or policy of the FAA. This report does not constitute a standard, specification or regulation."

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACE - Automatic Clutter Eliminator

ACP - Azimuth Change Pulse

ARP - Azimuth Reference Pulse

ARTS III - Automated Radar Terminal System III

ASR - Airport Surveillance Radars

BDAS - Beacon Data Aquisition Subsystem

db - Decibels

ET - Early Termination

FTC - Fast Time Constant

IOP - Input/Output Processor

MS - Milliseconds

MTI - Moving Target Indicator

NAFEC - National Airspace Facility Engineering Center

NAS - National Airspace System

nm - Nautical miles

nsec - Nanoseconds

PCD - Production Common Digitizer

P_d - Probability of Detection

PD/FD - Predetector/Final Detector

P_{fa} - Probability of False Detection

P_n - Probability of Quantization Due to Noise

PRF - Pulse Repitition Frequency

RDAS - Radar Data Aquisition Subsystem

RMS - Root mean square

rpm - revolutions per minute

SCF - Scan Correlation Feedback

SWD - Sliding Window Detector

TMRVDP - Terminal Modified Radar Video Data Processor

μs - microseconds

Introduction

(A) Description and Objectives. The study documented by this report is an investigation of the computer processing requirements related to radar video processing. A demonstration of practical and impractical computer processing approaches is brought out through evaluating numerous algorithms for the video processing functions. The end result of the study is a delineation of the computer requirements for several variations of function allocations between a special hardware device and the digital computer. Throughout the report, this hardware device will be called the Radar Digitizer, and the computer will be called the Input/Output Processor (IOP). Except for the process of combining radar and beacon data, the study is limited to exploration of radar processing. The beacon processing is assumed to be that of the Automated Radar Terminal System (ARTS) III, currently under development. The study is undertaken as part of the ARTS III Expansion Contract DOT FA70WA-2289.

The principal objective of this study is to help establish the role of the digital computer in radar processing for ARTS III. Specifically of interest is the computer role in what is called the Basic Radar Data Aquisition Subsystem (RDAS). The Basic RDAS will be an interim demonstration model which will establish a design for an early operational radar tracking system, suitable for installation at critical terminal sites. A second purpose of the Basic RDAS is to provide an experimental tool for definition of a follow-on improved radar tracking design. The objective of this study is not to present a totally defined system design for the Basic RDAS. The detailed definition and refinement of concepts presented here will be accomplished in a design effort to follow this present study. On the other hand, this study provides basic information as to what approaches are feasible from the standpoint of computer processing and also serves as a stimulant for generation of design ideas. Consequently this report is not a design document, but is considered as a concept study which will provide guidance for the later design effort. Beyond considering the investigation of the overall role of the computer, a second objective is to develop a repertoire of feasible processing algorithms. That is, for many of the radar processing functions, there are numerous algorithms which could be applied. It is desired to identify those for which computer processing is practical.

The algorithm and processing concepts considered are for the most part limited to tried and proven techniques. For example, a large number of the processing concepts are taken from the Terminal Modified Radar Video Data Processor (TMRVDP) and the Production Common Digitizer (PCD) developed for the enroute (NAS) system. However, some consideration has also been given to concepts demonstrated in certain military systems.

Performance analysis of the radar processing algorithms is not an objective of this study because of the restraint that the concepts are mainly limited to tried and proven techniques. However, where an algorithm is slightly changed to better suit computer implementation, a performance evaluation is provided to demonstrate the impact of such a change. A notable example of this is the analysis of the Scan Correlated Feedback (SCF) concept utilized in the TMRVDP.

The method for estimating the computer processing time and memory requirements is to investigate the actual techniques of programming the functions in the computer. To accomplish this, actual computer instruction sequences and data layouts were investigated for each of the functions examined. Based on the above method and on system capacity goals, estimates of processing time is generally given in two forms. The first form is an algebraic expression defining the required computer time per second as a function of certain variables, e.g., the number of targets. The percentage of time required on the basis of a given set of variables is the second form. These variables are identified in a later paragraph.

Where provided, the analysis of algorithm performance is based either on mathematical models or on Monte Carlo type simulation. The mathematical models are generally from the area of probability theory. Detailed explanations of the models are not usually provided in order to avoid deviating from the main topics. Monte Carlo simulation of the target detection function is presented in one of the appendices to compare two significantly different approaches to that function.

(B) System Parameters. Throughout the report, computer requirements are stated in terms of variables which are parameters relating to the radar system, the beacon system, and the surveillance area covered by these systems. These variables and their specific values are given below. The values listed represent the system requirements to which the Basic RDAS will be designed. For easy reference, this set of variables with their assigned values, are referred to as the standard parameters. They are as follows:

Radar	Symbol	Value
Range Quantization (in nautical miles)	q	1/16 nm
Maximum Range	R_m	60 nm
Number of 30-bit Quantized Video Words per Sweep (trigger)	N_W	32
Pulse Repetion Frequency (PRF)	f	1200/second
Beamwidth at Half Power Points (degrees)	θ_R	1.5°
(number of sweeps)	BW_3	20
Scanning Rate (revolutions per minute)	ω	15 rpm

Beacon	Symbol	Value
Pulse Repetition Frequency (PRF)	f_B	400/second
Beamwidth at Half Power Points	θ_B	3°

Environment	Symbol	Value
Number of Radar Targets/Second	N_R	125
Number of Beacon Targets/Second	N_B	63
Number of Beacon Fruit Responses/Sec	N_F	100
Miscellaneous		
Target Detection Window Length	W_L	17
Computer Memory Cycle Time	μ	750 nanoseconds

All references to miles in the above table and throughout the entire report are understood to be nautical miles. The value stated for N_R is to be interpreted as representing processed (reported) targets both from actual aircraft and other interference, e.g., clutter. The fruit rate stated pertains to the fruit transferred to the computer via the Beacon Data Acquisition Subsystem (BDAS). In practice, this would be the fruit not eliminated by a hardware defruiter. Based on a field data obtained through tests at the National Airspace Facility Engineering Center (NAFEC), a fruit rate of 100/second is considered to be higher than necessary; however this rate is considered acceptable for purposes of this study. The time and memory requirements for the computer processing are based on the assumption that the computer is the one which is specified for ARTS III. In particular, the time calculation assumes execution with the IOP operating with a dedicated memory module. The statement of these assumptions is not intended to constrain the possible operation of the Basic RDAS in a multiprocessing computer configuration. Rather, the significance is to provide a firm basis upon which time estimates can be made.

The processing time is calculated on the basis of memory cycle times (μ) which is taken to be 750 nanoseconds (nsec). The purpose of this policy is to simplify the calculation of execution time for program sequences. The actual instruction execution times, as stated in the design document (Reference 1) for the IOP, vary somewhat for different instructions and even for different modes within an instruction. For example, an add instruction requires 1.4 microseconds (μs) if executed with a constant within the instruction word, and 1.5 μs otherwise. Similarly, a logical product instruction requires 2.05 μs if executed with a constant within the instruction word or 2.15 μs, otherwise. The execution times assigned in this report are values which are integral multiples of memory cycle times (750 nsec). In all cases except multiply instructions, these multiples are upper bounds. Hence, the assigned values yield execution times greater than or equal to the actual times. For the instructions mentioned above, the assigned execution times are two memory cycles (1.5 μs) for the add, and three memory cycles (2.25 μs) for the logical product. Multiply instructions are assigned execution times of 12 memory cycles; whereas divide instructions are assigned 16 memory cycles. Transfers of 30-bit data words either into or out of the computer are defined as one memory cycle (750 nsec) per word.

One further assumption (constraint) is that only one IOP and one memory module may be assigned to the complete radar/beacon processing task. That is, the total processing time must be limited to that which can be handled by one IOP. Memory requirements must be less than 16,384 words. As such, the feasibility of an approach implies the above criteria must be satisfied.

(C) Report Organization. The report addresses each of the major radar processing functions somewhat independently, and then in the final section summarizes the total system processing requirements. In the following section, Video Processing Requirements, a brief discussion of the radar processing problem is given for the purpose of identifying and defining what functions must be accomplished. The subsequent section, Video Selection and Quantization, discusses functions related to preparing the analog video for computer processing. This includes the discussion of the type of video which should be transferred to the computer, how the computer can assist in the control of thresholds for the quantization process, and the impact of handling two types of video. The next major section, Target Detection and Beamsplitting, addresses the principal radar processing function, which is the detection and location of targets. Considerable attention is given to the problem of eliminating radar noise data. The remaining processing functions are discussed in the section entitled Ancillary Functions. Processing of beacon data, correlation of radar and beacon data, and utilization of tracking feedback information are the three main topics. Examples of total processing requirements are given in the Conclusion. These examples illustrate how the processing time and memory requirements change under different allocations of functions between the digitizer and the IOP. Three appendices are provided to address topics somewhat extraneous to the main theme of the report. The intent of the first two appendices is to show that essentially equivalent performance can be expected even though the computer implementation of certain functions differs from what has been done in the past. The final appendix is simply a review of the statistical formulation for one of the target detection algorithms.

The level of content in the test assumes that the reader has some background in the functional requirements of radar processing and also some understanding of digital computer capabilities. In particular, extensive reference is made to the TMRVDP system, and it is assumed that the reader is familiar with the concepts applied there. Except for the appendices, detailed mathematical discussions of the radar processing concepts are avoided. However, within the appendices, a fairly extensive background in probability and statistical theory is assumed. With respect to the digital computer capabilities, a general understanding of the data buffering concept and the elementary concepts of what is meant by programs, data bases, and Boolean operations is all that is required. This knowledge, of course, is with reference to the ARTS III IOP.

Video Processing Requirements

Stated succinctly, the purpose of a radar video processing system is to statistically detect targets by operating on the analog video signal from a radar receiver and to output target position estimates (range and azimuth). In order to accomplish this task effectively, the system must fulfill certain basic requirements. These requirements can be divided by function into five areas:

1. Video Selection
2. Video Quantization
3. Target Detection
4. Position Estimation
5. Radar/Beacon Correlation

Figure 1 is a simplified block diagram illustrating the relationship among the functional requirements mentioned above.

(A) Video Selection. Several types of video could be available to the system. Among these are normal, moving target indicator (MTI), their respective log videos, and their respective log-fast time constant (FTC)-antilog videos. The first stage in video processing involves choosing the one to be used over a given region of the radar surveillance area. For example, it might be desirable to use MTI video in an area of ground clutter or log-FTC-antilog video in areas of weather clutter due to its non-saturating properties. The video selection mechanism may be an automatic procedure or may be based on manual entries. Automatic procedures are discussed in the section entitled Video Selection and Quantization.

(B) Video Quantization. After video selection has been performed, quantization of the analog video signal takes place. Figure 2 is an illustration of a portion of a typical return pattern generated by one radar sweep.

The time lag of the return signal is directly proportional to the distance traveled by the radar pulse. It is therefore possible, using timing circuitry, to partition the amplitude curve into 1/16 mile range intervals. Quantization is accomplished by comparing the amplitude of the radar return in a given 1/16 mile interval to a voltage threshold, β. If the amplitude exceeds β, a logical ONE is generated for the position corresponding to the range interval in question. In this manner the analog signal from each radar sweep is quantized into a binary sequence.

The control of the quantization threshold, β, is extremely important to the performance of the entire system. The level at which it is set determines the percentage of ones which will appear in the quantization sequence. Ideally, this threshold should be controlled so as to minimize the probability of a one being set by noise alone (P_n), while insuring that the probability of a one being set by an actual target return is high. It is desirable from a target detection standpoint to control the β so that P_n

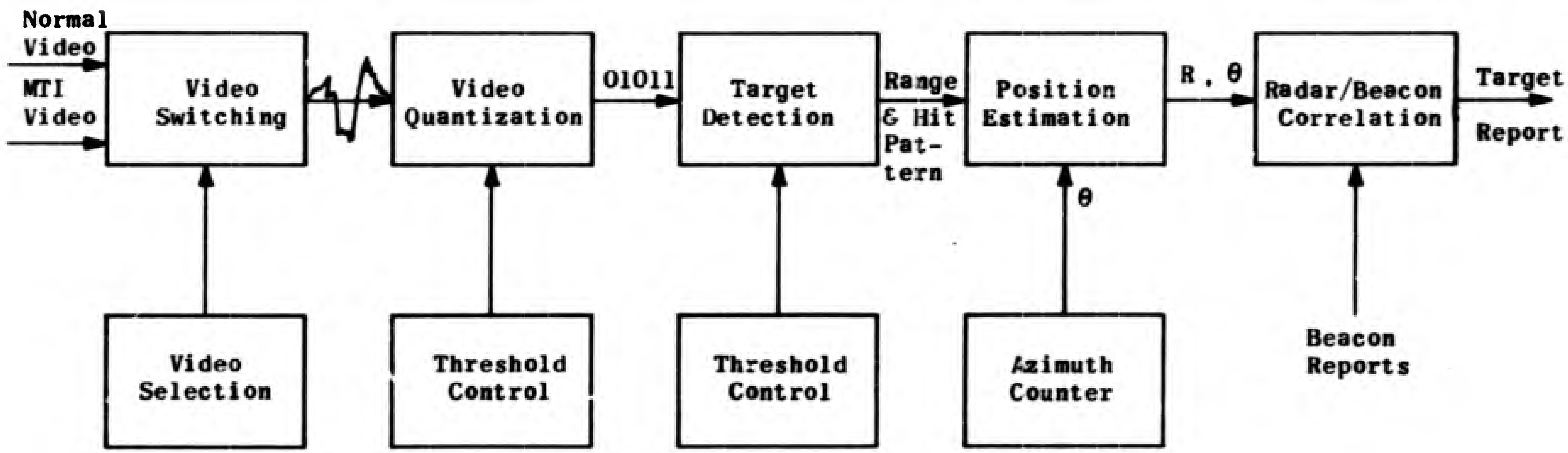


Figure 1. Simplified Block Diagram of Functional Requirements for Radar Video Processing

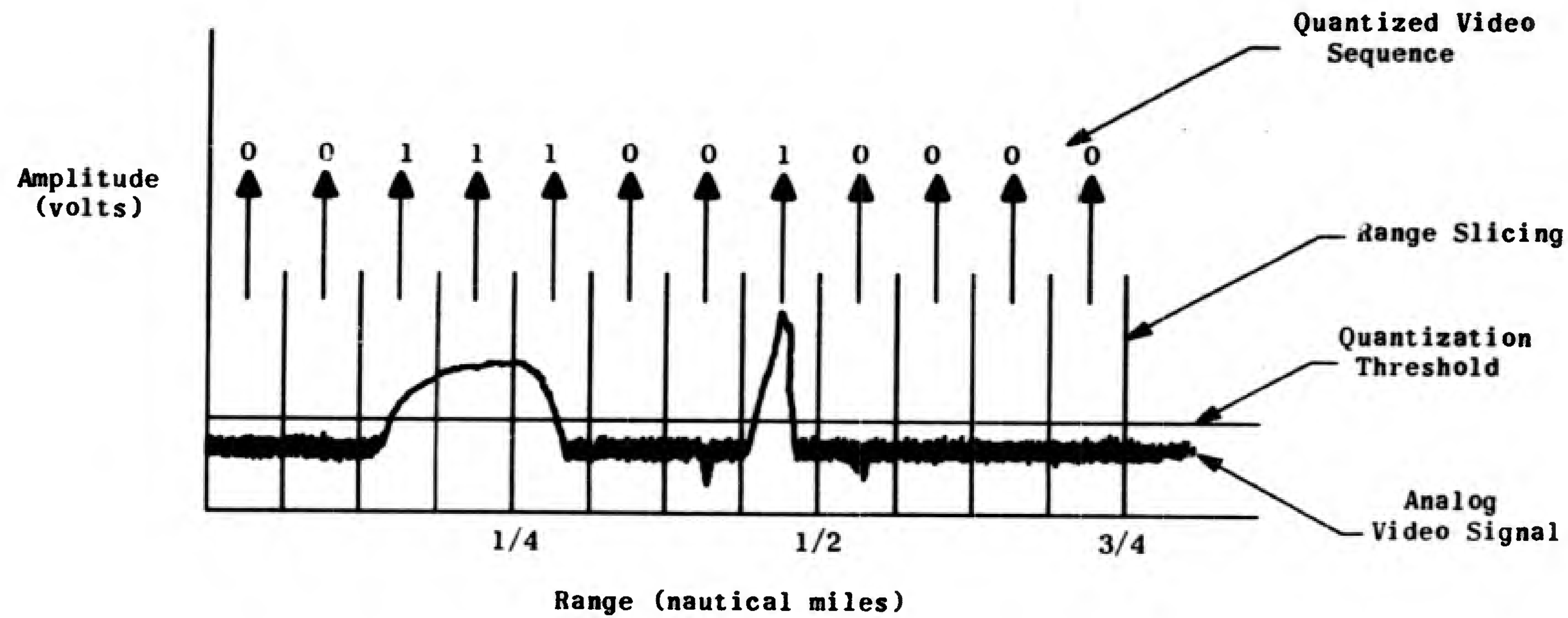


Figure 2. Quantization of the Analog Video Signal

is constant over the entire radar surveillance area. This is usually accomplished through the use of some sort of feedback system. Various methods of threshold control are discussed in the section entitled Video Selection and Quantization.

(C) Target Detection. The next step in video processing is to analyze the quantized video to determine whether or not a target is present. Target detection is accomplished on the basis of the density of the hit pattern located at the same range on consecutive sweeps. If the density is high, a target is declared at the range in question. This density may be measured in various ways. The most common method is to count the number of hits in a fixed number of adjacent sweeps — this number being approximately equal to the radar beam width — and to compare this count to a threshold. If the threshold is exceeded, a target is declared. The proper setting of this threshold is very important. If it is too low, many false alarms will result; whereas if it is too high, weak targets will not be detected. Methods of controlling this threshold are discussed in the section entitled Target Detection and Beamsplitting.

(D) Position Estimation. The result of the target detection stage is a declaration that a target is present at a certain range. In the position estimation stage, the azimuth of the target is estimated and a position estimate is provided in terms of range and azimuth (R,θ). Several methods of position estimation are discussed in the section entitled Target Detection and Beamsplitting. Each of these methods investigates the hit pattern of the target to make an estimate of target azimuth position.

(E) Radar/Beacon Correlation. Finally, the radar position estimate is correlated with the beacon position estimate. A single target report is produced if the radar and beacon reports are sufficiently close together,

Video Selection and Quantization

This section contains an investigation of processing requirements for selecting the type of video, control of the quantizer threshold, and transfer of radar data to the computer. The prime objectives are to determine what input, control, and selection techniques are feasible for implementation in the digital computer. As part of the objectives, the sensitivity of methods to changing parameters, e.g., range quantization and amount of data stored, is brought out. The use of the term "sensitivity" is interpreted as the change in computer processing time, memory storage requirements, and the capability of a method to perform its intended function.

The problems of video selection, quantization, and input of data into the computer are separated into four topics. The first main topic of discussion is that of inputting data to the computer. Here the discussion settles on what type of data should be input, how much must be stored, and how much computer time is consumed for buffering and controlling the input process. The second topic addresses control of the quantization process. Only one concept, Scan Correlated Feedback (SCF) is treated; however, two alternatives of the basic concept are discussed and evaluated for time and core requirements. Next, the implementation of video selection is considered through another application of (SCF) concept. An alternative, that of inputting two types of video to the computer, is also considered. The final topic investigates methods of implementing range strobe elimination.

(A) Input of Radar Data. The determination of what and how much radar data to bring into the computer is strongly dependent on the type of processing considered for the data. Therefore, this discussion could be put off until after the topic of target detection and beamsplitting; however this is not necessary since the amount of knowledge required about the detection process is slight. The reader need only have an understanding of the conventional sliding window target detection process to follow the discussion of this subsection. Furthermore, discussing the topic of input first, establishes practical limits on the amount of data a target detection algorithm can call upon.

Two types of processing may be imposed on the radar data in the computer. The first and most important is to detect and locate the position of radar targets. The second is for control of the conversion from radar analog video to digital (binary) form. The data for the target detection function places the greater demand on the computer. Hence, the discussion in this subsection will be directed primarily towards furnishing data for this latter requirement. As will be seen, the additional requirements for data with which to control the quantization will either be small or of the same form as the data for target detection.

The type and amount of data required in the computer for target processing depends on whether the detection function is accomplished in the Radar Digitizer or in the computer. A detailed investigation of the data load for both of these schemes is provided in the following paragraphs.

(1) Detection of Radar Targets in the Digitizer. Receiving only detected target reports would be a feasible task for the computer. The average data rate would be low relative to the computer capacity. The only problem would be to handle two or more target detections occuring during the same sweep and at nearly equal ranges (e.g. within ½ mile). The average data rate would simply be related to the design parameter N_R, that is, the number of radar targets per second. Considering the maximum range of 60 miles and a range accuracy of 1/16 of a mile, then 10 bits would be needed for expressing range. With an azimuth granularity of 4096 Azimuth Change Pulses (ACP) per revolution, 12 bits are required for azimuth (figure 3). Hence, one 30-bit computer would would be sufficient for each report. The corresponding average data rate for N_R = 125/seconds presents no problem. With respect to peak data rate, the maximum input rate for the IOP is 420,000 words/second or one word every 2.4 μs. This period of time corresponds to about 3/16 of a mile of data. Assuming a capability of resolving targets separated by only 1/8 miles, some provision for queueing reports in the hardware would be necessary. This requirement could easily be handled with one or two holding registers. Lastly, the problem of storing target reports in the computer is also simple because of the small average input rate. The input time, which consists primarily of controlling the input sequence, is 1.1 percent. Memory requirements are on the order of 100 words. In summary, this type of input would place no significant demands on the computer. In fact, the only required computer processing of the target reports would be to correlate the radar data with the beacon data.

A minor modification of the above concept would be to have the hardware detect the leading edge of the target and then transfer quantized video from that vicinity. For example, the hardware could detect leading edge using the sliding window concept. At this point the hardware could transfer range, azimuth, leading edge threshold, and the present quantized video within the sliding window. See figure 4. The hardware would then set a target in process flag and collect another window of quantized data over the next W_L sweeps. This too would be transferred to the computer with range, azimuth and leading edge threshold. This process could be repeated perhaps once more to insure the computer has all of the quantized data from the target. The benefit sought through this type of input scheme would be improvement in rejection of clutter data and beamsplitting accuracy. It would also permit estimating target signal strength. This would presumably be obtained through application of a sophisticated processing algorithm, via the computer. However, at this point the merits of the scheme are not the prime concern. Rather, computer requirements are the prime concern.

The input and storage requirements for this method would be similar to the previous, and again the only significant problem would be handling targets at nearly equal ranges. Using the example of the previous paragraph, 10 bits of range, 12 bits of azimuth, 5 bits of leading edge threshold, and W_L-bits of video would be transferred three times per leading edge detection. Each such transfer would require two 30-bit words. Hence, the average data rate would be $6N_R$. For N_R = 125 targets per second, the average rate is 750 words/second. At 750 nsec/word, this corresponds to less than 0.1 percent of the IOP time. Conflicts due to targets at similar ranges would have to be handled by queuing arrangement in the hardware.

Bit Position 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Spare	Azimuth	Range

Ave. Data Rate = 125 wds/sec
Max. Data Rate = 420,000 wds/sec

Figure 3. Format for Input of Target Reports to IOP

Bit Position 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Spare	T_L	Azimuth 1	Range
Spare		W_L Bits of Video	

Spare	T_L	Azimuth 2	Range
Spare		W_L Bits of Video	

Spare	T_L	Azimuth 3	Range
Spare		W_L Bits of Video	

Azimuth 1, 2, and 3 are the respective azimuths of the 3 windows of video

Ave. Data Rate = 750 wds/sec

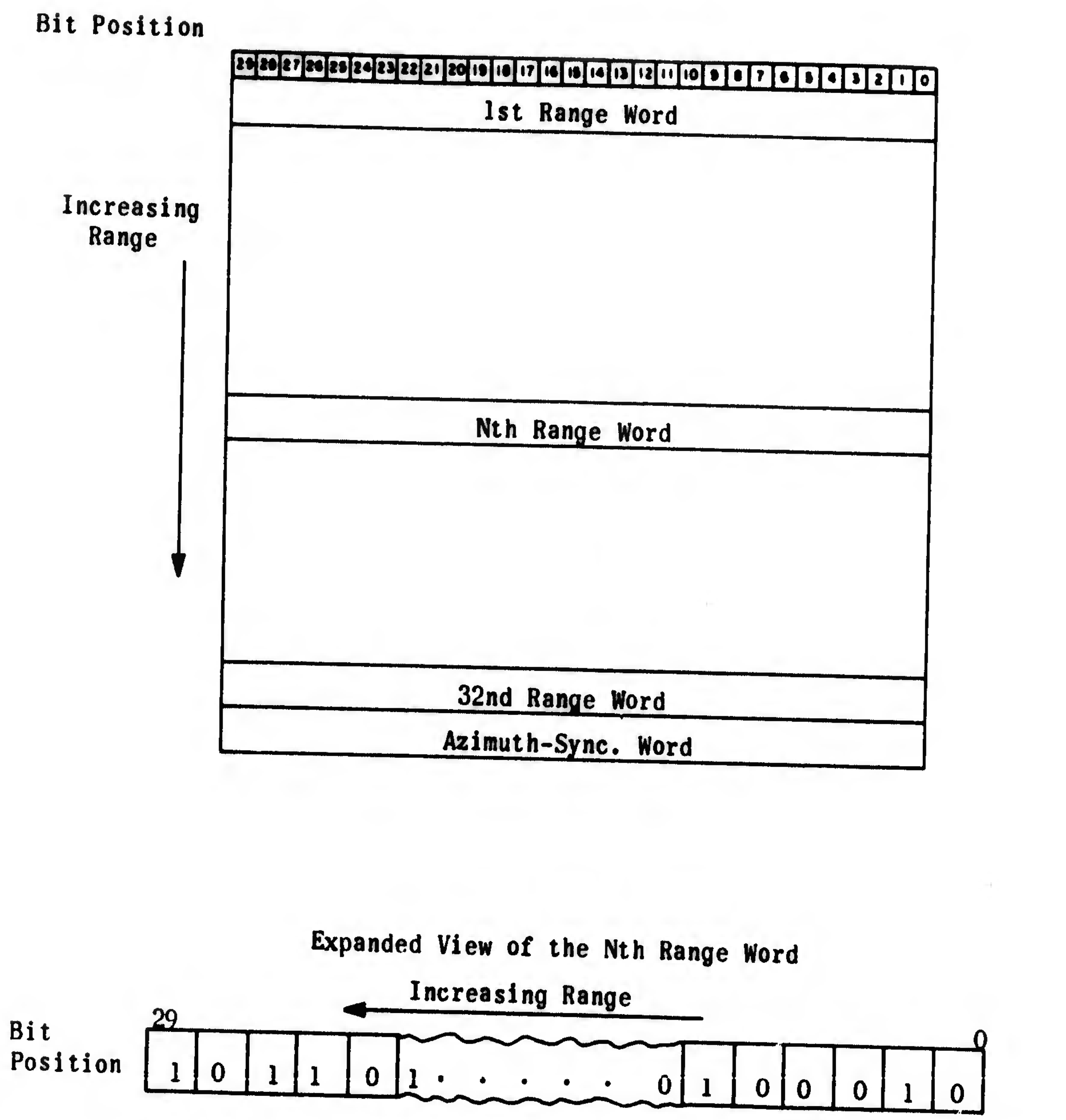
Figure 4. Format for Input of Target Lead Edge and Quantized Video in the Vicinty of the Target to the IOP

The delay between the first and third transfers of the target data would be $2W_L$ sweeps. Therefore, sufficient storage area would have to be provided in the computer to save all the data generated during the corresponding period of time ($T = 2W_L/f$). For purposes of estimating the storage area, consider that targets and clutter reports are distributed in azimuth according to a Poisson probability distribution with mean rate of N_R. Then N_RT is the average number of reports produced by the hardware in T seconds. A safe estimate of the worst case requirement is three times this average. Therefore, at six words per target, 18 N_RT is the estimated core requirements. For the standard parameters stated earlier, this amounts to 64 words.

Control of the input process could be handled via a double buffer scheme wherein one of two alternating buffers are initiated to handle the data over a given period. A period could be one or more sweeps, preferably more, such as three sweeps. The initiation of a period would be sensed via an interrupt from the hardware. Alternate switching between two buffers would require transferring the report data to the holding area mentioned above. On the basis of three sweeps per period and a report rate of $N_R = 125$ reports/second, ($12\ f + 54\ N_R$) memory cycles or 1.6 percent of the computer time would be consumed. The input processing program would be small, requiring only about 30 instructions. Two buffers of about 10 words each would probably be sufficient for receiving the report data of three sweeps. In summary, the input requirements for this alternative are 1.6 percent for time and less than 200 words of memory.

(2) Input of Quantized Video. Departing significantly from the first two approaches is the idea of bringing all quantized video into the computer. This approach places a considerably higher demand on the computer, but as will become evident, the load is well within the data handling capacity of the computer. The quantized video for each sweep would be formatted into 30-bit words and transferred to the computer. For example, consider the quantization of radar data into two-level (0 or 1) for each 1/16 mile range interval, called range cells. A radar having a maximum range of 60 miles would have 960 such range cells. Since the amplitude is quantized into two levels, only one bit of a computer word is required for each range cell. For the case at hand, the number of 30-bit video words per sweep (N_W) would be 32. The range data would be serially shifted into 30-bit words serially as quantized; hence range order would be preserved. The zero bit of the first word for each sweep would represent the range interval 0 to 1/16 of mile. Bits 1 to 29 of this word would contain the data starting at 1/16 of a mile and increase consecutively in 1/16 of a mile increments to the range of 1 7/8 miles. The quantized video for the next 1 7/8 miles would be consecutively formatted into the second word beginning at bit 0. The form for the third to the 32rd word would be similar. The layout of one sweep of data is illustrated in figure 5. The word (WD_r) and bit (B_r) position of a given range (r) would satisfy the following equation:

$$r = (30WD_r + B_r)q$$



Each bit represents the quantization of 1/16 radar miles in range

Ave. Data Rate = 45, 000 wds/sec.

Figure 5. Quantization of One Range Sweep into Serialized Range Words

where q is the length of the range interval (1/16 miles) and the words are assigned numbers beginning with 0 and ending with N_W-1. The hardware would transfer each word to the computer as soon as all 30 bits have been filled. Thus, a word will be transferred, approximately, every 22.5 μs, since round trip time for radar data, which travels at the speed of light, is approximately 12 μs/radar mile. The 22.5 μs/word corresponds to a rate of roughly 45,000 words/second which is well within the single channel capacity of the computer (420,000 words/second).

For the purpose of maintaining computer synchronization with the hardware and for determining the azimuth corresponding to a given sweep of radar data, an azimuth/synchronization word would be transferred after the last video word of each sweep. Thus, the total words for a sweep would be $N_W + 1$. The question of how much of this quantized video should be held is a subject of the following paragraphs.

By the nature of the serial processing characteristics of a general purpose digital computer, it is desirable to operate on block of data at a time. That is, it is more effective to buffer a moderate size quantity of data and process in blocks of the same size than to attempt to operate on each word directly as it enters the computer. The reason for this is that the requirement of the computer program to determine when a word is in memory would expend essentially all of the processing time. So then the question is, what is a reasonable size for the data? The answer to this question involves trading off processing time and flexibility against storage requirements. The discussion in the following paragraphs begins with consideration of a single sweep, and then moves on to the multiple-sweep case. The result is a recommendation of about 24 sweeps for the Airport Surveillance Radars (ASR).

To input one sweep of data while processing another is not practical for the ASR. The period between sweeps is very small — for f = 1,200, the period is 833 μs. This short period would impose severe constraints on the length of time that could be used in searching the data for targets. As will be seen in the discussion of Target Detection and Beam Splitting, much of the quantized video can be rapidly scanned. However, upon indication of a target, the processing time to make the target detection decision is on the order of 300 μs. This is a large portion of the 833 μs per sweep period. Furthermore, the target detection techniques, e.g. the sliding window, require a number of sweeps of video equal to the window length. Therefore, at least W_L sweeps of quantized video must be held in the computer. If this is the case, there is little to be gained in inputting on a one sweep basis.

Consider next an input scheme with three buffers, each of which is long enough to store N sweeps of data. Imagine the buffers arranged as in figure 6 and assign three functions to the buffers as follows:

1. Buffer A — history area
2. Buffer B — processing area
3. Buffer C — input buffer area.

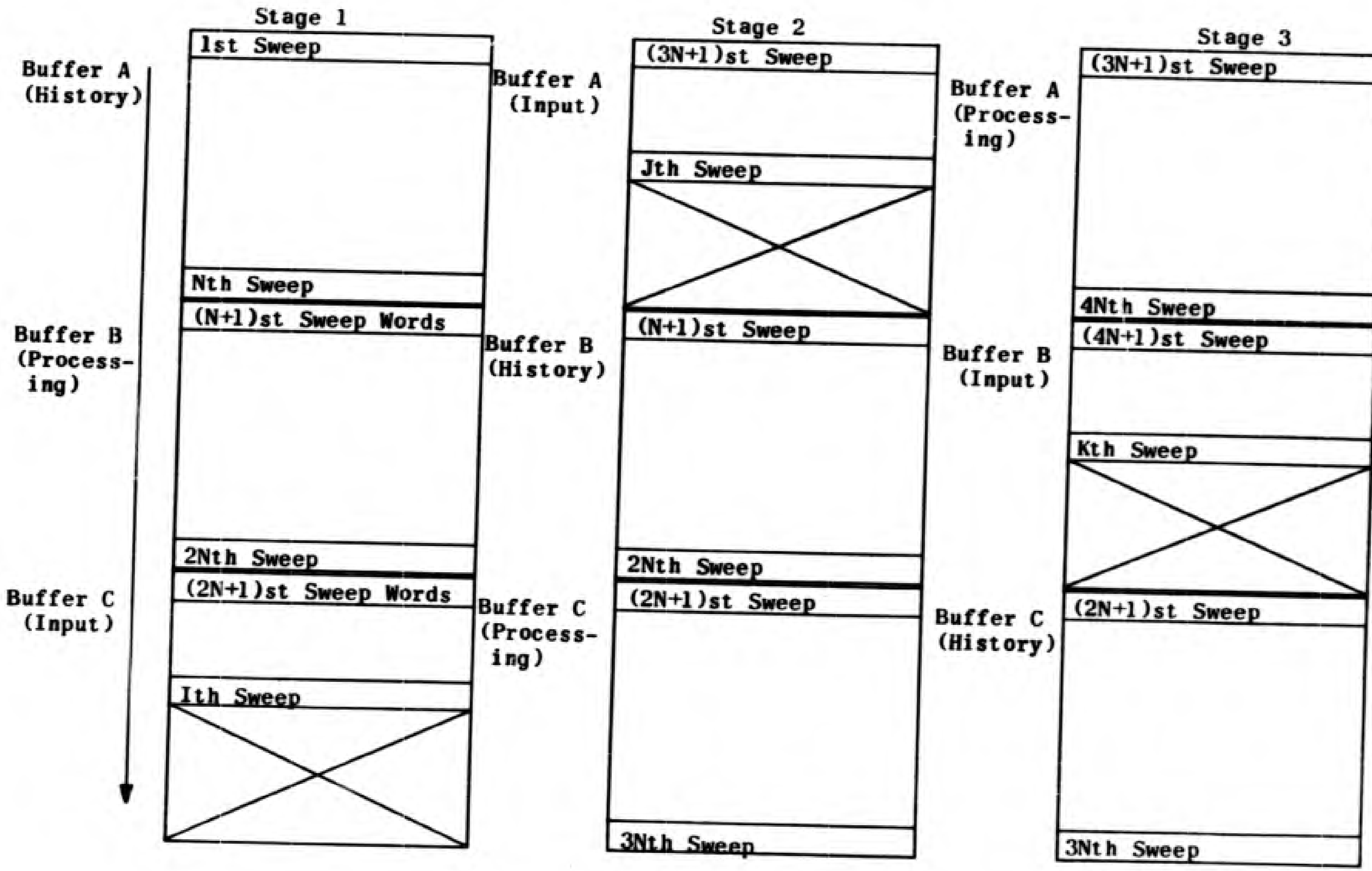


Figure 6. Three Buffer Cyclical Method of Quantized Video Storage

The input buffer area is reserved for N sweeps of new quantized video. The processing area is N sweeps of quantized video to be processed while the input buffer is filling. The history area is used in conjunction with the processing area to provide sufficient quantized video for the sliding window detectors. That is, the detectors require W_L sweeps of video. The joint processing and history areas help to fulfill this requirement. When Buffer C is filled, the processing of data in Buffer B has to be complete so that the assignment of buffer functions can be permuted as follows:

1. Buffer A — input buffer area
2. Buffer B — history area
3. Buffer C — processing area

After the next N sweeps, the functions are again permuted. The input sequence continues in this manner with assignments being permuted every N sweeps.

The choice of the number (N) of sweeps per buffer is now discussed. Although $N = W_L$ seems like a logical choice, it is possible to design target detection algorithms which require that only $W_L/2$ sweeps be held in the history buffer. Hence, the lower limit on the choice of N is $W_L/2$. At the other extreme, the maximum buffer length of the IOP is 1,024 words. This requires that N be limited to $1{,}024/(N_W + 1)$, since $(N_W + 1)$ is the total number of words received each sweep. Therefore N must satisfy the following inequality condition:

$$W_L/2 \leq N \leq \frac{1{,}024}{N_W + 1}$$

Considering a target detection window length of 17 and 32 words of quantized video per sweep, N is bounded as follows:

$$9 \leq N \leq 31$$

Some other system design criteria may also be applied. It is intended that the RDAS operate in a time-sharing mode initially with beacon processing and eventually with other functions. This imposes another restriction on the length of the buffer. For successful time-sharing operation, it is desired that the repetition rate or calling frequency of a function be kept low to minimuze program overhead. Although the ARTS III criteria has not yet been specified, a repetition rate of 50 times per second (every 20 milliseconds) would probably be reasonable.

Applying the above criteria to choosing a buffer length would result in an N of 24. That is, choosing a processing repetition rate of $R_r = 50$/second and having a radar PRF of $f = 1200$ yields $N = f/R_r = 24$. With these guidelines it is now possible to get estimates of processing time for controlling the input sequence and also for the amount of memory storage required.

By using a continually running input/output (I/O) command chain, the time required for controlling the input buffer is minimal. The chain would be set up to initialize Buffers A, B and then C sequentially. Following Buffer C, the chain would loop back to the beginning and reinitiate Buffer A. To provide indicators for coordinating input with processing, the chain would set a buffer-in-processing flag at the beginning of each buffer and clear this flag at the end of that buffer. During the processing, this flag would be tested to see if the processing is in proper sequence with input, that is, whether the processing was either ahead or behind the corresponding input. This would take on the order of 20 instructions and about 0.1 percent of the IOP time for both the processing program and the input chain.

The memory space for the triple buffer scheme is a significant factor. The number of words is $3N(N_W+1)$ where 3N is the total number of sweeps required by the triple buffer scheme and (N_W+1) is the number of words per sweep. Table I lists the memory requirements for three choices of N and three different range quantization factors. Input time for buffering video data and controlling the sequence is also shown.

Table I. Input Buffer Memory Space and Time Requirements

Quantization	N_W	Memory Storage (words)			Buffer Time (percent)
		Minimum N	Preferred N	Maximum N	
1/8	16	459	1,224	3,060	1.6
1/16	32	918	2,448	3,069	3.1
1/32	64	1,836	2,925	2,925	6.0

(3) Summary of Input Analysis. Two basic approaches have been considered. The first approach is to input data from the hardware only in the vicinity of a hardware detected target. For the two alternatives considered there, computer requirements for the input function are small. On the second approach, which is to input all quantized video, the memory requirements are significantly larger; however the processing time remains small. Table II provides a summary of the trade-off considerations for the three input schemes. Because of the interrelationship between input to the computer and subsequent computer processing, it is not meaningful to choose an input scheme at this point. For example, target detection could be performed in the computer only if all quantized video is input. However, since the processing for target detection has not yet been analyzed no firm recommendation can be made. The only significant conclusion at this point is that all three input schemes are feasible.

Table II. Trade-Off Summary of Input Techniques

Method	Data Transferred	Input Time (%)	Computer Memory (30-Bit Words)	Hardware Memory* (Bits)	Advantages	Disadvantages
Input of Target Reports (Digitizer Target Detection and Beamsplitting)	Range, azimuth	1.1	200	15,360	Digitizer detection and beamsplitting of target implies minimal processing in computer	Restrictive to changes in target detection and beamsplitting algorithms
Input of Target Leading Edge Reports (Digitizer Target Detection, IOP Beamsplitting)	Range, azimuth, leading edge threshold, quantized video in neighborhood of target	1.6	200	15,360	Digitizer detection of targets implies minimal processing in computer Permits flexibility in choosing beamsplitting algorithm	Restrictive to changes in target detection algorithm
Input of Quantized Video (IOP Target Detection and Beamsplitting)	Quantized video, azimuth, synchronizing code	3.1	2500	None	Permits flexibility in choosing target detection and beamsplitting algorithms	Induces heavy processing load in computer

*This considers only the memory required for holding quantized video used in target detection (960 range cells x 16 sweeps).

(B) Control of Quantizer Thresholds. One of the most critical functions of a digital video processing system is the control of the clip level used in quantizing the analog radar video to binary data. Numerous techniques are available for this function. For example, the following are under consideration for the Basic RDAS:

a. Fast/Slow Feedback (from the PCD)
b. Digital Fast Time Constant (from the Weather Outline Generator)
c. Scan Correlated Feedback-SCF (from the TMRVDP)

Of these, the only one which is considered as a candidate for computer implementation is the SCF. The other two require too fast a response for the computer to participate and also require operation on analog radar video. Nevertheless, these techniques are important candidates for hardware implementation. In the discussion which follows, the processing time and core requirements for the TMRVDP version of the SCF and two alternatives are examined.

(1) Scan Correlated Feedback. The SCF system is a quantizer threshold control system which partitions the entire radar surveillance area into zones and permits independent selection of thresholds in each zone. Its operation in each zone is essentially that of an infinite history controller. The number of ones produced in the quantized video of a zone during one radar scan is compared with a control parameter. On the basis of whether the count is greater than or less than this parameter, the quantizer threshold is raised or lowered for the next scan. The concept utilizes 64 threshold levels covering the voltage interval of the analog radar video. A block diagram of this concept is shown in figure 7 and analysis of the SCF can be found in Appendix A.

Computer implementation of the SCF concept as defined for the TMRVDP requires a considerable expenditure of processing time. An actual SCF program instruction sequence was derived and analyzed for execution time. The program design follows the TMRVDP, except for the size of the zones. Because the computer operates on 30-bit words, the length of a zone in range is taken as 1 7/8 miles. That is, for 1/16 mile quantization the number of range cells in a zone would be 30. The program design also assumes 32 Azimuth Change Pulses (ACP) per zone. For zones of other multiples than 1 7/8 miles and 32 ACPs the time would vary slightly. Further assumptions are that the noise level would be individually selectable in steps of 0.10 percent on a zone by zone basis. Noise levels in the interval of 0 to 25 percent may be chosen. The resulting formula for the computer processing time per second is

$$(8f + 18{,}040\ N_W + 18216)\mu$$

Using the standard parameters, the corresponding computer time is 45.2 percent. The memory requirements for program and data is as follows:

Program	200 words
Quantizer Threshold Levels for 4096 Zones	2048 words

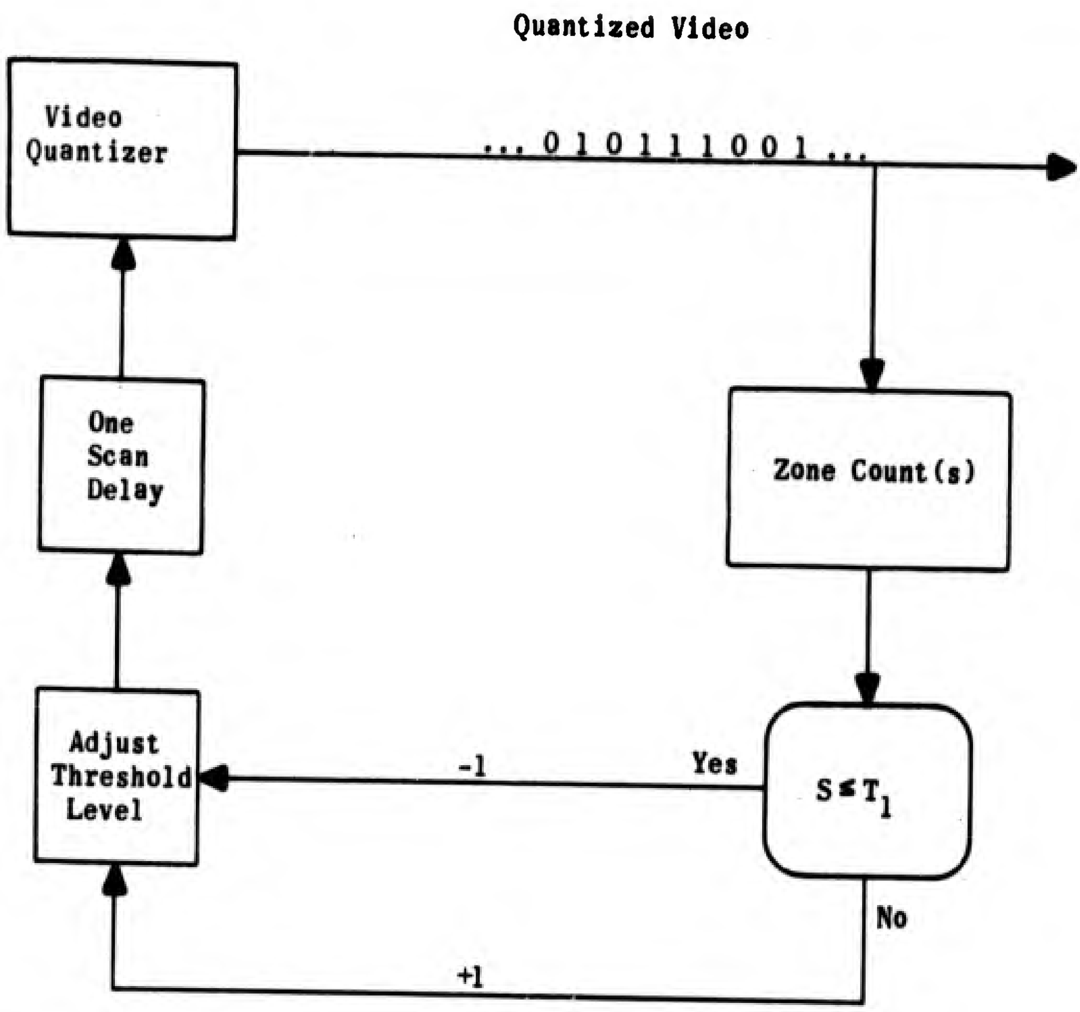


Figure 7. Block Diagram of Single Stage Thresholding for Scan Correlated Feedback

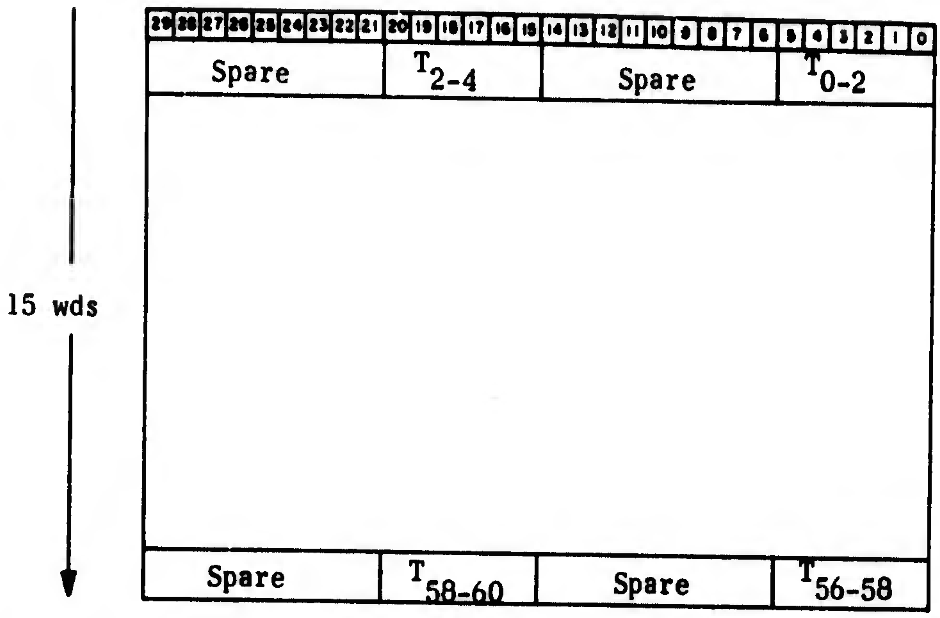
Figures given to this point state the requirements for determining the thresholds, but do not include transferring the thresholds back to the digitizer for the next radar scan. This is the subject of the next paragraph.

Because the thresholds for a given range can change, at most, once every 32 ACPs, two approaches regarding the transfer of quantized threshold data to the digitizer are considered. The first approach requires the digitizer to hold the full range ensemble of thresholds (32 values) for the 32 ACPs, and then request that the next aximuth sector be transferred. The second approach has the computer repeatedly transferring the same data to the digitizer for each sweep in the 32 ACP sector, and then switch to the data for the next azimuth sector. Note the analysis for both variants of this method assumes two thresholds are packed into each 30-bit words (figure 8). Hence one azimuth sector requires 16 words.

The output requirements for the first approach can be handled by a very simple output chain. This chain would buffer directly out of the table used in computing the quantizer thresholds. Further, it would utilize a sequence of 128 output commands (corresponding to the 128 azimuth sectors per scan) and automatically jump back to the beginning of the chain. Once in synchronization with the digitizer, the chain could run continuously and unattended. Provision for monitoring synchronization could be provided by having a routine to check on the processing once per scan. This monitoring function could be initiated via digitizer interrupt notifying the computer of the Azimuth Reference Pulse (ARP). Such a chain would require 129 memory locations (plus a small monitor routine) and a negligible amount of processing time (on the order of 0.01 percent). In addition, this approach would require 192 bits of memory in the digitizer.

The second approach eliminates the memory in the digitizer. On each sweep of an azimuth sector (32 ACPs), the same set of thresholds data is buffered to the digitizer. The output scheme for this approach also utilizes an automatic recycling output chain; however, one cycle of the chain would keep reinitiating the same buffer. Furthermore, once during each sector the radar processing program would be required to set up the buffer limits for the next azimuth sector. Memory requirements for the chain would be about 90 words. The time including buffering and output chain control would be about 1.6 percent. This output scheme is preferred over the one described above since the small increase in computer time would save on memory in the digitizer.

The total computer requirements for this version of the SCF using the repeated buffering of thresholds is roughly 3,100 words and 47 percent of the IOP time. (See table III). This time is extremely high. The next paragraph addresses the modification of the SCF concept which makes it more practical for computer implementation.



T_{0-2} = Quantizer Threshold for SCF Zone from 0 to 2 mi.

.
.
.

T_{58-60} = Quantizer Threshold for SCF Zone from 58 to 60 mi.

Max. Data Rate = 20,800 wds/sec.

Figure 8. Format for Transfer of Thresholds to Digitizer from IOP (TMRVDP SCF)

Table III. Scan Correlated Feedback Concept
Terminal Modified RVDP Version

Time		Memory	
Processing Program	45.2%	Processing Program	201 words
		Output Chain	90
Output	1.6	Zone Thresholds	768
		Quantizer Thresholds	2048
	46.8%		3107 words

Note: The figures for processing time do not include the time to input quantized video.

(2) Two-Stage SCF. The two-stage SCF, illustrated in figure 9, places a portion of the processing in the digitizer. The operation of determining whether the threshold for a given zone should be raised or lowered is performed in a two-level decision process. First, on each azimuth sweep the digitizer counts the number of video ones in each 2 mile interval. On the basis of a parameter called the range density threshold (R1), a binary one or zero is generated for each range interval of a sweep. If the number of ones in an interval exceeds R1, a one is generated; otherwise a zero is generated. At the end of each sweep the digitizer transfers to the computer the data for each two mile interval (30 bits) (Figure 10). The second level of test is performed in the computer. For each zone (2 miles by 32 ACPs), the computer checks to see if the number of times the range density was exceeded is above another parameter called the azimuth density threshold (A1). If the count is above, the quantizer threshold is incremented by one level; otherwise, the threshold is decremented by one.

It has been determined that the two-stage technique provides nearly equivalent performance to the TMRVDP version. The substantiation of this claim is given in Appendix A. Note also that the number of zones for this approach is 3840 because the number of range segments (two mile intervals) is taken as 30 instead of 32. Noise levels for each zone are individually selectable in steps of about ½ percent from zero to 24 percent.

Because the digitizer does the counting of the quantized video, the processing time in the computer is greatly reduced. The time and core requirements for the two-stage SCF along with a number of special alternatives are given in the following paragraphs.

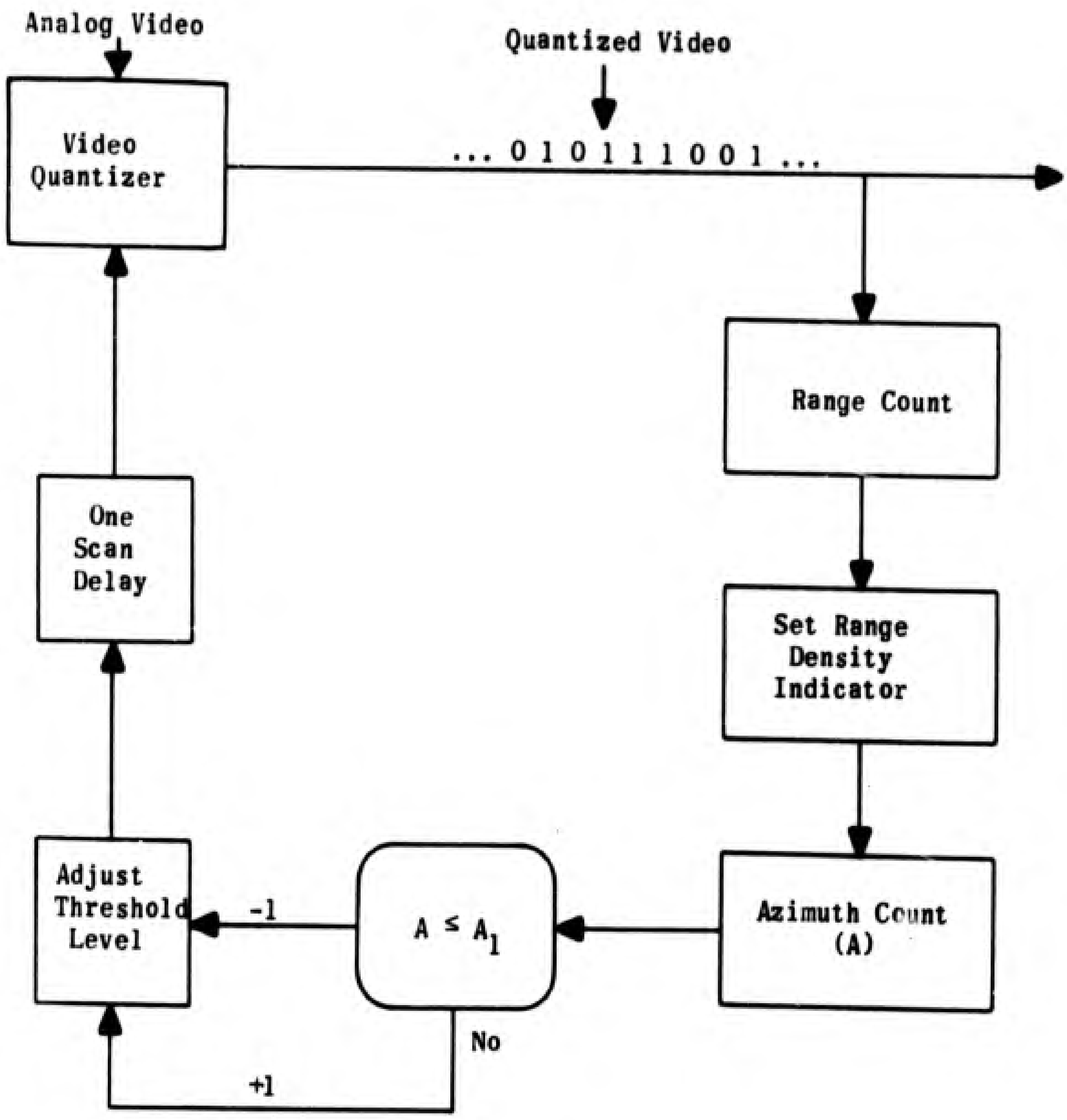


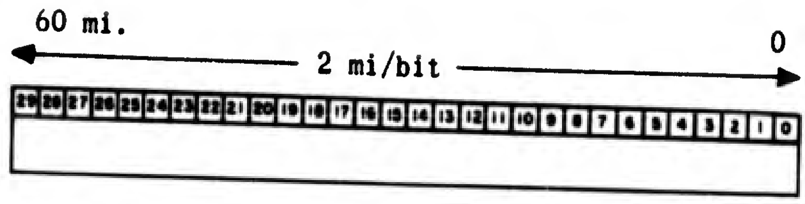
Figure 9. Block Diagram of Two Stage Thresholding for Scan Correlated Feedback

The basic process of determining whether to increment or decrement the quantizer threshold requires only 5.0 percent computer time. However, there are a number of remarks to make about the total requirement. First, as indicated in the description of the method, the digitizer must transfer the range density word at the end of each sweep. The time for this is a mere 0.09 percent, if it is input along with the regular quantized video. Furthermore, no additional input control logic is required. If the triple buffer input scheme (see section entitled Input of Quantized Video) is used with 24 sweeps per buffer, another 72 words of buffer space is required. The second remark is that the processing time and core requirements reduce as more thresholds are packed into one word. Recall that for the SCF method of the previous approach, two thresholds are packed per word. The values stated earlier in this paragraph are upon the basis of five thresholds per word. If 64 levels of quantizer thresholds are assumed, six bits are required for each threshold (figure 11). Hence, five is the maximum number of thresholds that can be put in a word. Furthermore, as a consequence of the five thresholds per word, the number of words per sweep decreases to six and the output time decreases. Assuming the output method which buffers the quantizer thresholds every sweep, the output time is only 0.72 percent. The total cost for the two-stage SCF is 5.8 percent in processing time and 1,885 in memory space. See table IV.

Table IV. Basic Two-Stage SCF (5 Quantizer Thresholds/Word)

Time		Memory	
Processing Program	5.0 %	Processing Program	187 words
		Output Chain	90
Input	0.09	Input Buffer	72
Output	0.72	Azimuth Density Thresholds	768
		Quantizer Thresholds	768
	5.8 %		1885 words

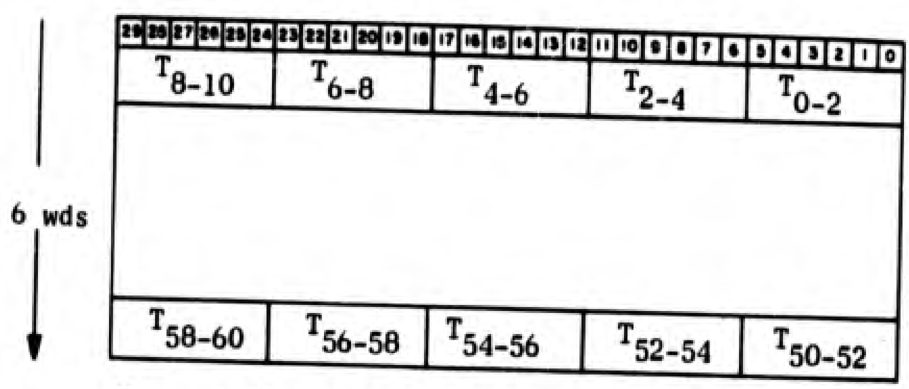
One item which has not yet been brought out is the method for control of the digitizer range density threshold (R1). The previous discussion tacitly assumes that R1 was already available to the digitizer; however it is important to consider just what options are available in choosing R1. First R1 is the same for all zones in the surveillance area. The second option would be to have two possible values which would be selectable on a per zone basis. This could be handled by preceding the quantized threshold data for each sweep with one 30-bit word defining the R1 for each of the thirty zones (figure 12). The program for setting up these words would be small and the increased output buffering time and core



Range Density Crossing Word
(Each word represents one sweep)

Data Rate = 1200 wds/sec.

Figure 10. Format for Range Density Crossing Word



T_{0-2} = Quantizer Threshold for SCF Zone from 0 to 2 mi.
⋮
T_{58-60} = Quantizer Threshold for SCF Zone from 58 to 60 mi.

Max. Data Rate = 8,300 wds/sec.

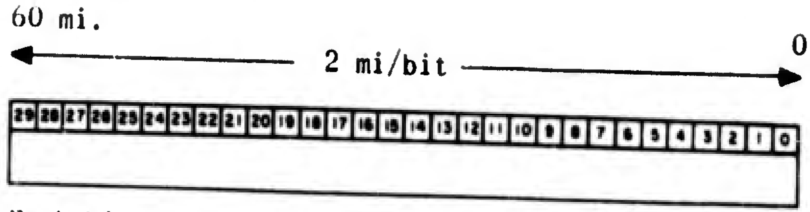
Figure 11. Format for Transferring SCF Thresholds to Digitizer from IOP (5 Thresholds/Word)

would be 0.09 percent and 128 words, respectively. Finally the third and most sophisticated option would provide the Rl value individually for each zone. A suitable range of Rl values is zero to seven. This would imply three bits are required per zone or a total of ninety bits per azimuth sector. Because three words would be required for the Rl variables, it is not economical to transfer all Rls to the digitizer at the beginning of each sweep. The alternative is to pair the Rls and quantizer thresholds on a zone by zone basis. With three bits for the Rl and six bits for the quantizer threshold, only the parameters for three zones could be formated into one 30-bit word (figure 13). Hence, the total number of words transferred in each sweep would be ten. The most significant fact, however, is the increased processing time, which is 6.6 percent. The total effect of the option can be seen by reviewing table V. The second option, that of having two possible Rls for each zone is preferred because it allows flexibility without a high cost in processing time.

Table V. Two-Stage SCF(Independent Rl for Each Zone)

Time		Memory	
Processing Program	6.6 %	Processing Program	165 words
		Output Chain	90
Input	0.09	Input Buffer	72
Output	1.08	Azimuth Density Thresholds	768
		Rl and Quantizer Thresholds	1280
	7.8 %		2375 words

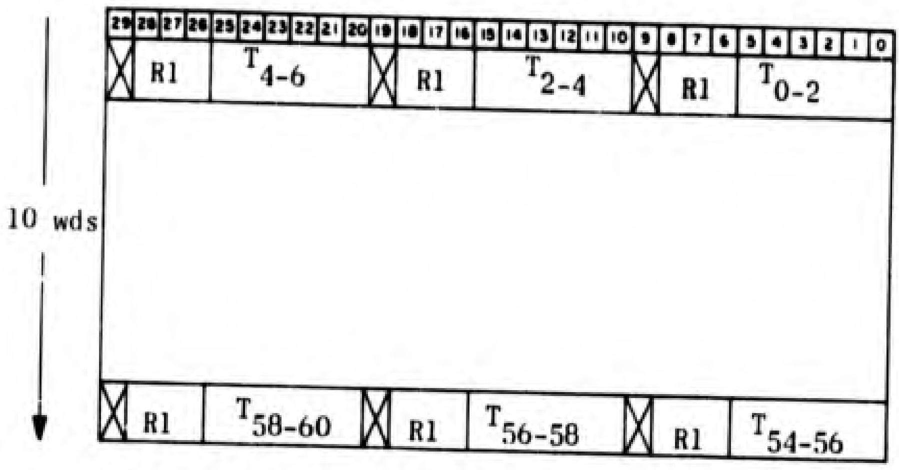
Another alternative to the two-stage SCF is to have the quantizer threshold levels held in the digitizer and have the computer simply determine whether to increment or decrement the thresholds. The operation would be almost the same as previously discussed except that the computer produces only one 30-bit control word per azimuth sector (figure 14). Each bit would inform the digitizer to increment the quantizer threshold for a corresponding zone if set to one, or to decrement the threshold if zero. As could be expected both the computer requirements for time and core decreases slightly. See table VI. However, 23,040 bits of memory would be needed in the digitizer to hold the quantizer threshold levels. For this alternative, a further unsettled point is how to select and control the Rl parameters. Options for this are similar to that discussed in the previous paragraph. The option of having the digitizer hold the thresholds is not recommended because the computer time and core are not significantly reduced and the option is costly in digitizer memory usage.



Each bit position allows a choice of one of two R1's.

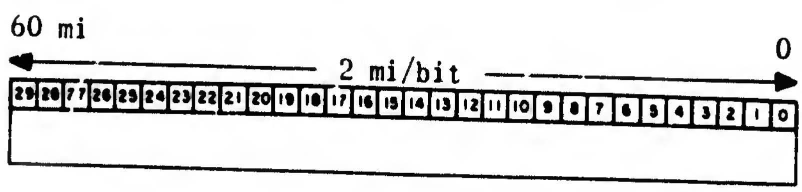
Data Rate = 1200 wds/sec

Figure 12. Format for R1 Selection Word (Two Choices)



Max. Data Rate = 13,900 wds/sec

Figure 13. Format for Transfer of SCF Threshold and Independent R1 for Each Zone from IOP to Digitizer



0 in bit position = decrement threshold
1 in bit position = increment threshold

Data Rate = 1200 wds/sec

Figure 14. Format for Increment/Decrement Word

Table VI. Two-Stage SCF (Quantizer Threshold in the Digitizer)

Time		Memory	
Processing Program	4.7 %	Processing Program	145 words
Input	0.09	Output Chain	90
Output (once/azimuth sector)	0.01	Input Buffer	72
		Azimuth Density Thresholds	768
		Increment/Decrement Words	128
	4.8 %		1203 words

The final option to the two-stage SCF provides improved performance. In this method, which is called the tri-level two-stage SCF, the quantizer threshold is permitted to be incremented, decremented, or left unchanged as a result of examining the range density data. That is, the quantizer threshold for a particular zone is incremented only if the number of sweeps having range density crossing is greater than an upper azimuth density threshold (A2). A threshold is decremented if the corresponding number is less than the lower azimuth density threshold (A1). If the count is between A1 and A2, there is no change in the level. The analysis of this method in Appendix A shows that improved performance is attained. A possible modification to this method is to set secondary limits B1 < A1 and B2 > A2. If the count is less than B1 the threshold is decremented by more than one level, whereas if the count is greater than B2, the threshold is incremented by more than one level. However neither performance analysis nor computer requirements are provided for this latter modification. It is noteworthy that the SCF system implemented in the TMRVDP uses only the bi-level process.

The processing program for the tri-level option is similiar to the basic two-stage SCF (bi-level) except that two thresholds are checked for each zone. For I/O formats see figures 12 and 13. As a consequence both the time and core requirements increase. The details of these requirements are given in table VII. Note that the packing of five quantized thresholds per word is assumed for this method. Also the noise level in the quantized video would be controllable from zero to 24 percent in steps of ½ percent. The tri-level option is recommended over the bi-level because it significantly increases performance without excessive increases in process time.

Table VII. Two-Stage SCF (Tri-Level Option)

Time		Memory	
Processing Program	5.5 %	Processing Program	234 words
Input	0.09	Output Chain	90
Output	0.72	Input Buffer	72
		Azimuth Density Thresholds	1536
		Quantizer Thresholds	768
	6.3 %		2700 words

In summary, the two-stage SCF is a feasible method of implementing quantizer threshold control in the computer. Figure 15 is a block diagram illustrating the quantizer threshold control options discussed in the previous paragraphs. The options recommended for design consideration are shown in the smooth cornered retangles and are connected by double lines. It must be emphasized that the recommendations are made from the standpoint of efficient computer implementation without sacrificing performance. In review, the Scan Correlated Feedback concept is the only quantizer threshold control system which lends itself to computer implementation. The fast response time required in the other methods mentioned rules out computer implementation. Two-stage SCF is recommended because it provides essentially equivalent performance to that of the single-stage SCF, but conserves computer time. The transfer of thresholds on every sweep to the digitizer is recommended because hardware memory is conserved with negligible increase in IOP processing time. The use of two RIs per zone is recommended because it allows flexibility without a large amount of processing time. It is also recommended that quantizer thresholds be held in the IOP instead of the digitizer in order to conserve digitizer memory. Finally, tri-level incrementation is recommended because of the improvement in performance. Table VIII lists the characteristics, processing time, and core requirements for the tri-level two-stage SCF method, when all the recommended options are included. For data rates and I/O formats refer to figures 12 and 13.

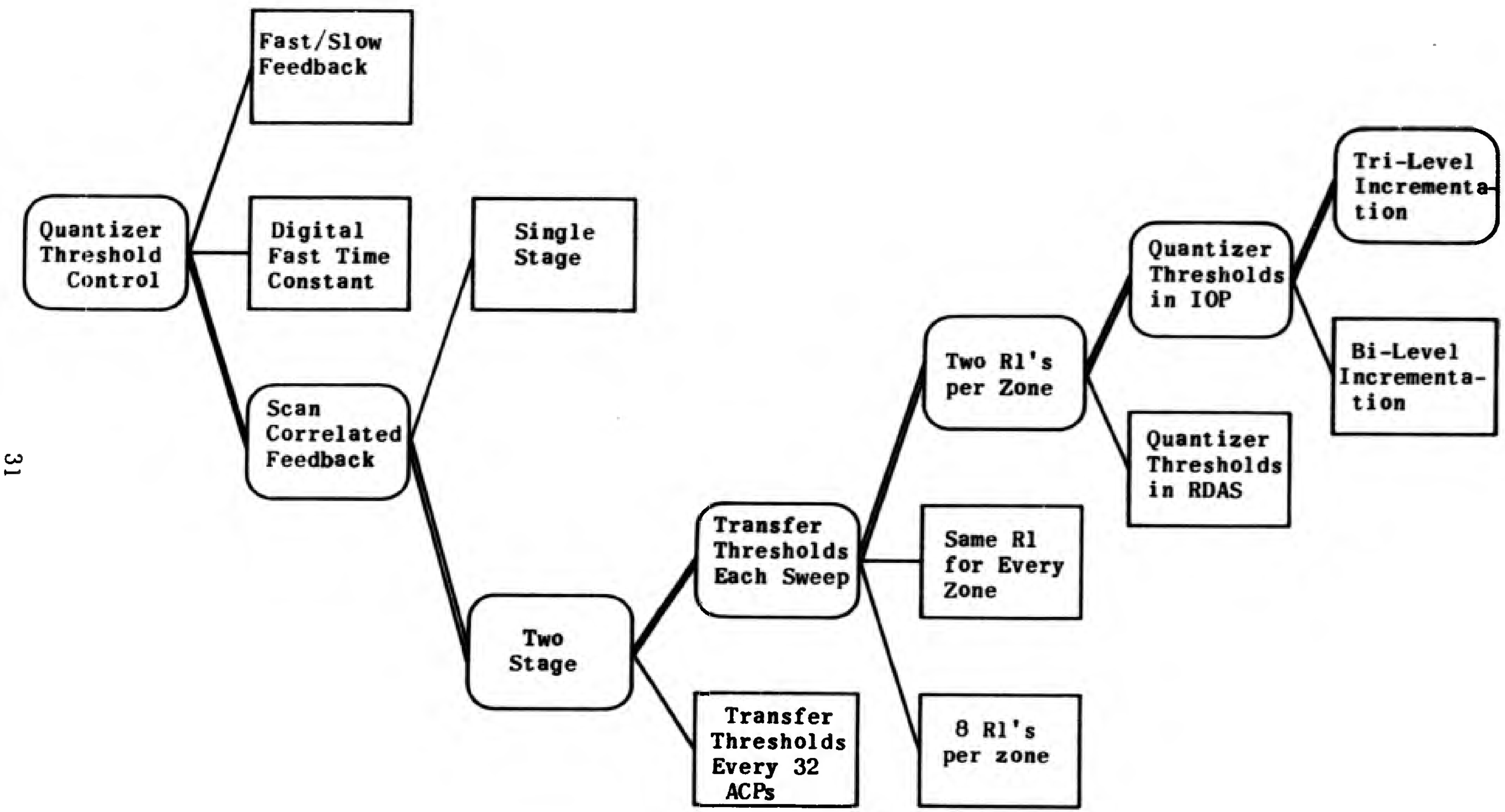


Figure 15. Block Diagram of Quantizer Threshold Control Options

Table VIII. Tri-Level Two-Stage SCF

Time		Memory	
Processing Program	5.5 %	Processing Program	234 words
Input	0.09	Output Chain	90
Output	0.81	Input Buffer	72
		Azimuth Density Thresholds	1536
		Range Density Threshold Selector	128
		Quantizer Thresholds	768
Total	6.4 %	Total	2828 words

Characteristics: Two-Stage decision process
Tri-Level Incrementation of Thresholds
Maintain Thresholds in IOP
Two Range Density Thresholds per Zone
Output Quantizer Thresholds Every Sweep

(C) Selection of Radar Video. To this point the discussion has concentrated on the extent of processing which should be performed on the quantized video before it is brought into the computer and how to control the threshold used in the quantization process. Consideration of the video type, such as normal or moving target indicator (MTI), has been ignored. It is true, however, that the techniques considered this far apply equally well to different types of video. Now the question of how to handle the presence of two video types is addressed. Although the concept is not limited to two video types, the discussion is restricted to two types for the sake of expedience. These are considered as normal and MTI.

Two approaches regarding the selection of video are treated. The first is a concept of switching between the two choices on a zone-by-zone basis. The alternative approach is to bring into the computer both video types, thereby permitting the target detection process to operate on both videos. The time and core implications of these two approaches are examined in the following paragraphs.

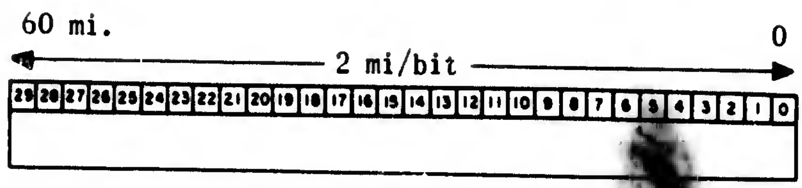
(1) Switching Between Two Videos. The switching problem can be handled by another application of the two-stage scan correlated feedback concept. In the digitizer there would be a special register and logic which counts the number of 1/16 mile range cells wherein the normal video exceeds the upper limit of quantizer threshold control. The threshold applied in this quantization would normally be set very close to the saturation, or

limit value of the radar normal video. Therefore a high number of ones over a small range interval would tend to indicate saturation in the normal video. The count would be tested and reset every two miles. The digitizer would generate a binary one as the indication that the count has exceeded a control parameter, called the range density threshold R2, and a zero, if otherwise. The values of R2 would be between zero and 32. The 30 indicator bits -- corresponding to the 60 mile range -- would be transferred to computer at the end of each sweep (figure 16). The computer would perform an azimuth density check on the basis on a 32 ACP sector. The two mile by the 32 ACP zone is choosen for compatibility with the quantizer threshold control zones. If the number of sweeps having the range indicator set is high, this would indicate that the normal video had saturated throughout the zone and that no useful information could be extracted. Therefore a switch to the alternative video (MTI) would be warranted. The decision would be implemented by checking the number of range indicators having one against another control parameter, called the azimuth density V1. The value of V1 would be selectable between zero and 32. The same value would be applied in all zones. The decision would be indicated by setting a binary one for each zone wherein MTI is desired and setting a zero for selection of normal video (figure 17). These video selection words would then be transferred to the digitizer at the appropriate time during the next revolution. The selection in each zone would be controlled individually on the basis of how much clutter existed for that zone.

There is some question as to how appropriate the switch from normal to MTI video would be if the saturation was due to weather clutter. Additional logic may have to be incorporated to prevent inappropriate switching. In any case, the performance of the above concept would have to be evaluated experimentally. Irrespective of performance, it is of interest to determine the computer requirements if such a scheme were to be implemented. This is provided in the following paragraphs.

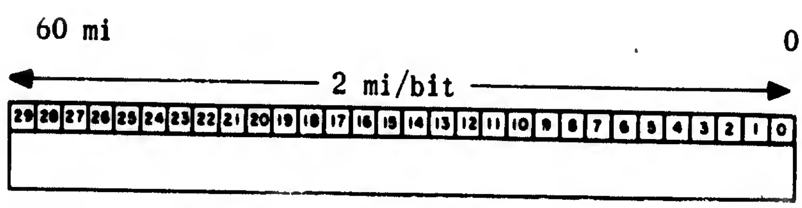
The program for implementing this concept would be very similar to the two-stage SCF suggested for the quantizer threshold control. Two possibilities exist. First, the selection function is incorporated into the quantizer control function. That is, the functions for video selection would be added to the basic program for quantizer threshold control. Second, the selection function operates independent of the quantizer control. The latter would be sensible only if the SCF concept was not utilized for threshold control. The time and core requirements for both possibilities are tabulated in table IX.

Another important issue is what happens to the quantizer threshold process when the video is switched. It is unreasonable to assume that the threshold levels appropriate to one video would also apply to the other. Two alternatives also exist for this issue. First, the switching problem could be ignored under the assumption that the switching is infrequent and that the SCF process would automatically seek out the new level. Thus a transition period would exist wherein either an abnormally high or low video noise level would prevail. The low noise level would tend to degrade



Data Rate = 1200 wds/sec

Figure 16. Format for Video Switching Range Density Threshold Crossing Word



Data Rate = 1200 wds/sec

Figure 17. Format for Video Switching Word

target detection probability; whereas the high noise level would tend to saturate the detection process. If the switching is on an infrequent basis, the first form (low noise) would be preferred. This situation could be assured by setting the threshold to a high level, when switching videos. The choice of the high threshold level could be determined through experimentation during the evaluation of the system. The second alternative to regulating the quantizer thresholds at switch over would be to provide an SCF loop for both types of video. This concept would involve adding another range density register in the digitizer, another input word every sweep and further processing in the computer. In terms of cost, this add-on would be very similar to the video selection add-on. The pertinent costs for a bi-level process are shown in table X. The incremental cost for a tri-level process would be the same as that indicated on the basic two-stage SCF.

In summary, the switching between two types of video can be provided through another application of the Scan Correlated Feedback concept. Furthermore, the quantizer threshold control for the second video can be handled by sharing a single control loop between the two videos or by providing an independent feedback system for each video. Since the frequency of switching videos is unknown, the latter approach is preferred.

As in the discussion for the quantizer threshold control, the question of how to control the R2 threshold is present. The solutions for the video selection case are analogous to those promoted for the quantizer control, so nothing new would be learned by repeating them here. Further, the analysis of time and core would produce similar results. The preferred approach is to provide two R2 values in the digitizer, and to permit the selection of these on a zone by zone basis.

Table IX. Two-Stage SCF Concept for Video Selection
Add-on to Quantizer Threshold Control

Time (percent)		Memory (words)	
Option 1:		Processing Program	115
Processing Program	2.7	Input Buffer	72
Input	0.09	Azimuth Density Thresholds	768
Output	0.09	Video Select Indicator Words	128
	2.9		1083
Option 2: Independent Operation		Processing Program	145
Processing Program	4.7	Output Chain	90
Input	0.09	Input Buffer	72
Output	0.27	Azimuth Density Thresholds	768
		Video Select Indicator Words	128
	5.1		1203

Table X. Two-Stage SCF Additional Requirements for Second Quantizer Feedback System

Time		Memory	
Processing Program	3.0 %	Processing Program	115 words
Input	0.09	Input Buffer	72
Output	0.54	Azimuth Density Thresholds	768
		Quantizer Thresholds	768
	3.6 %		1723 words

(2) Parallel Input of Two Videos. The second approach to the video selection problem is to input the second video in parallel with the first video. For example, both normal and MTI video could input for each range cell. The net effect of this approach is to double the input time and buffer requirements. Figure 18 gives format and data rate for parallel input. For example, under the assumption that q = 1/16 mile, the requirement to input two types of video would be the same as if only one video type was input, but that the quantization was 1/32 mile with these assumptions. the time is 6.0 percent and the buffer memory space is 4,896 words.--see table I. The control of the second video quantizer threshold would impose the further requirements previously discussed.

(3) Summary of Video Selection. The comparison of the two basic approaches for video selection is shown in table XI. The time and core values represent the incremental cost to add on the stated functions, under the assumption that the first video type is already being input and that the two-stage SCF is utilized for the quantizer control of the first video. Because the processing requirements for target detection on two videos has not been established, it is not meaningful to make a choice of video selection methods at this point.

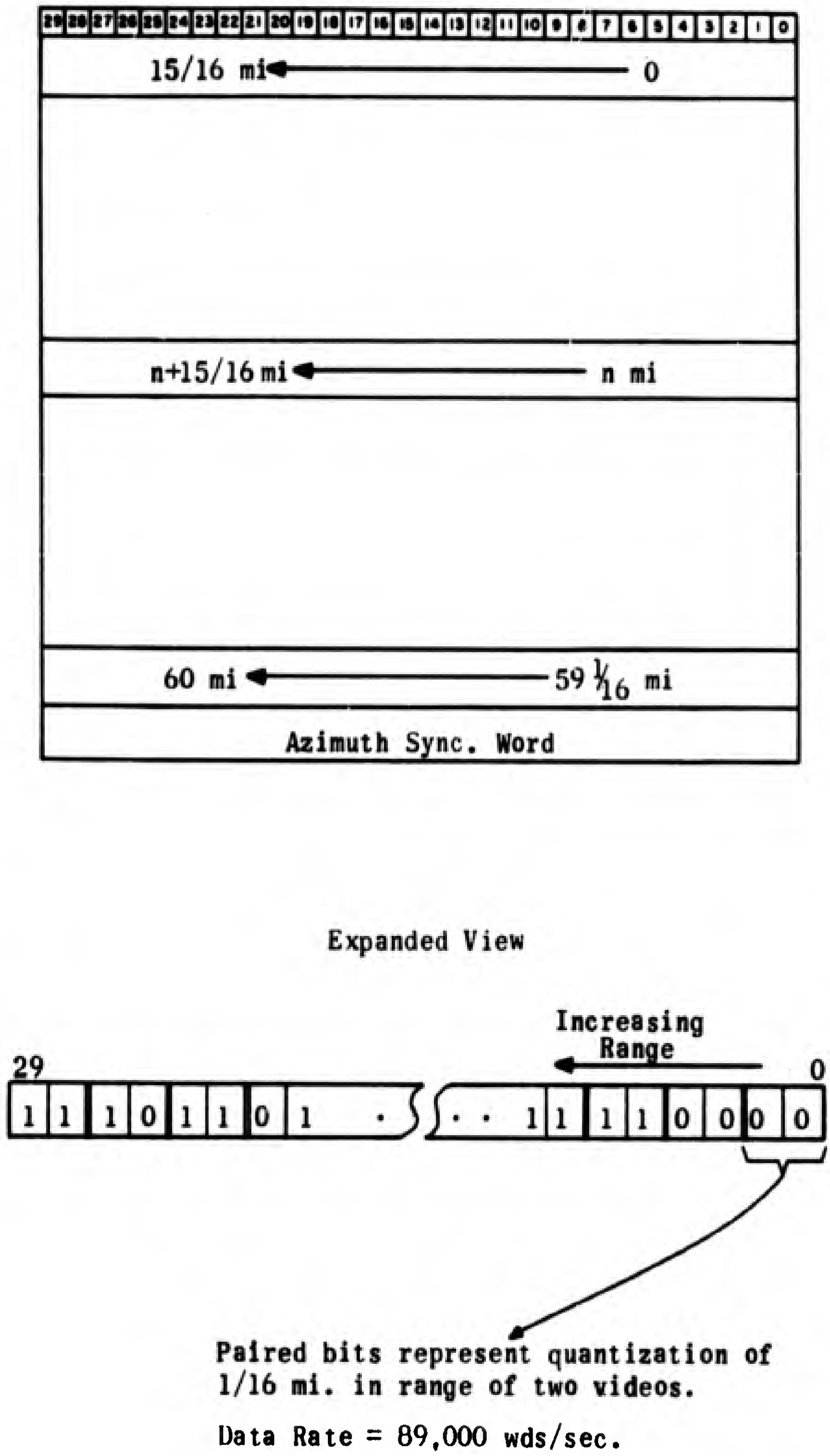


Figure 18. Format for Parallel Input of Two Videos

Table XI. Comparison of Video Selection Approaches

Option	Time	Memory
Video Switching		
Video Selection via SCF Concept	2.9 %	1083 words
Second Quantizer Threshold Control	3.6	1723
	6.5 %	2806 words
Parallel Input of Two Videos		
Input of Second Video	2.9 %	2448 words
Second Quantizer Threshold Control	3.6	1723
	6.5 %	4171 words

(D) Range Strobe Elimination. Occasionally, some type of electronic interference will cause the radar video to become saturated for a period of time. This interference may be from another radar or from some other electronic device emitting at a frequency near that of the radar. The period may be as short as a few microseconds or last for several sweeps. The latter case may cause perturbations in the target detection process. The counteraction to this problem is called range strobe elimination. Basically, then, the function of this process is to detect sweeps during which the number of quantized ones is abnormally high and to eliminate this data. Further, whenever a significant number of range strobes occur together, it is desirable to inform the controller that the target detection data in this area is degraded.

The first approach considered for detecting range strobes is simply to count the number of quantized ones generated during a sweep. If this number exceeds a critical value, then all of the quantized video for that sweep is set to zero. To provide for detecting persistent range strobes over a number of sweeps, the bit for the first range cell could be set to one for each strobe detected. If a range strobe occurs with sufficient frequency, the target detection process would sense the range strobe in the same way as for ordinary targets. Because of the 0 range, the declared target would be marked as a range strobe report.

Computer implementation of a range strobe detector is impractical because of processing time. The processing time per second for the process as described above is given by

$$f(7 + 53N_W)\mu$$

For the standard parameters the processing time is 153 percent. This, of course, is excessive and range strobe elimination by this method is much more economically accomplished in hardware.

An alternative is to group the quantized video and then pick out the number of groups having a certain characteristic. For example, the IOP has the capability to effectively test for all zeros in a 15-bit word. Hence, a process can be defined which would count the number of non-zero half words of quantized video. If this number exceeds a statistically determined threshold, the sweep is said to contain a range strobe and the data is eliminated. The processing time for this approach is given by

$$f(7 + 11N_W)\mu$$

which corresponds to 32 percent for the standard parameters.

If the counting of the video is undertaken in the digitizer, the computer time required reduces to a practical value. For example, let the digitizer count the number of quantized ones per sweep and then transfer this count to the IOP at the end of each sweep. The processing in the IOP would then be reduced to checking the count against the threshold and clearing the sweep data if the count is high. The processing time per second for this approach is

$$f(11 + 5N_W P_{RS})\mu$$

where P_{RS} is the probability of a range strobe on a given sweep. A reasonable estimate for P_{RS} would be 0.01. With this assumption and considering the standard parameters, 1.1 percent processing time is required. The memory requirements would be less than 100 words including program and additional buffer space for the video count. Note that even for a P_{RS} of 0.10, which operationally would be excessively high, the processing for this method is still only about 2.4 percent.

Of the three approaches considered, only the last two satisfy the constraints of feasability for computer implementation. Further, the placing of the counting process in the digitizer provides a significant pay-off in reducing computer time. Therefore, the last approach is greatly preferred over the second approach. Note further that the second method assumes that all quantized video is buffered to the computer; whereas the last approach does not have this requirement. Rather it requires only that the count be transferred each sweep.

In addition to the software approaches already given, there is a completely hardware approach which permits more precise control. A sliding range window would be utilized to censor the quantized video on a cell by cell basis. The number of ones in the window would be compared with a threshold. If the number exceeds the threshold, the quantized video corresponding to the center of the range window would be zeroed. This procedure would be repeated for each range cell of the sweep. Therefore, this approach would permit more selective control for the range strobe elimination process. Instead of always deleting a full sweep of quantized video sometimes only a portion would be deleted. The window length would determine the sensitivity. For example, the window length corresponding to two miles might be appropriate. Comparing the hardware approach (sliding range window) to the preferred approach involving software (hardware counting; software decision process) results in a recommendation for the former. The principal reason is the improvement in selectivity of the video censoring.

Target Detection and Beam Splitting

This section deals with the main topic of Radar Video Processing, which involves determining if a target signal is present in the quantized video, and estimating its position. The description begins with a review of the conventional method of target detection and beam splitting, which is the sliding window detector. This is followed by a discussion of how target detection can be accomplished in the computer. Here a unique approach called the Predetector/Final Detector Concept is introduced. Several target detection algorithms based on slightly different statistical formulations are considered. The section ends with a discussion of two approaches for estimating target position. Again in this section the prinicipal objective is to investigate the computer requirements of the various approaches.

Historically, many target detection algorithms have been tried, but the one which is most commonly applied is called the sliding window detector (SWD). To provide a background and understanding of the target detection concept to be developed later in this section, a brief discussion of this conventional SWD is presented.

(A) Sliding Window Detector. The concept of the SWD is that the quantized video within a small region is examined for the number of hits (binary ones). When the hit count is sufficiently high a target is said to be present. The concept of window stems from the fact that the quantized video is taken at a fixed range and over several sweeps in azimuth Thus, the sample has the appearance of a little slit or window over the data, see figure 19. One window is required for each range cell (1/16 of a mile). As each new sweep of radar data becomes available, the window is moved one position clockwise; this sliding of the window repeats with every sweep.

The threshold of the sliding window is chosen so as to control the number of cases in which noise causes a detection. Noise here is interpreted as ones which were quantized strictly from a random spike in the analog video. Under the assumption that noise is stationary and of known level, the probability for the number of hits, k, in a window of size, W_L, is given by:

$$\binom{W_L}{k} p^k q^{W_L-k}$$

where p is the probability of a single range cell containing a one and q = 1-p. A false alarm, that is a target which is declared from noise alone, is generated whenever the hit count is above the threshold. For purposes of discussion in this section the probability of a false alarm, P_{fa}, will be taken as the probability that t or more noise hits are contained in the window, i.e.

$$P_{fa} = \sum_{k=t}^{W_L} \binom{W_L}{k} p^k q^{W_L-k}$$

Other more sophisticated definitions of P_{fa} are sometimes used; however, for this report the above definition is adequate.

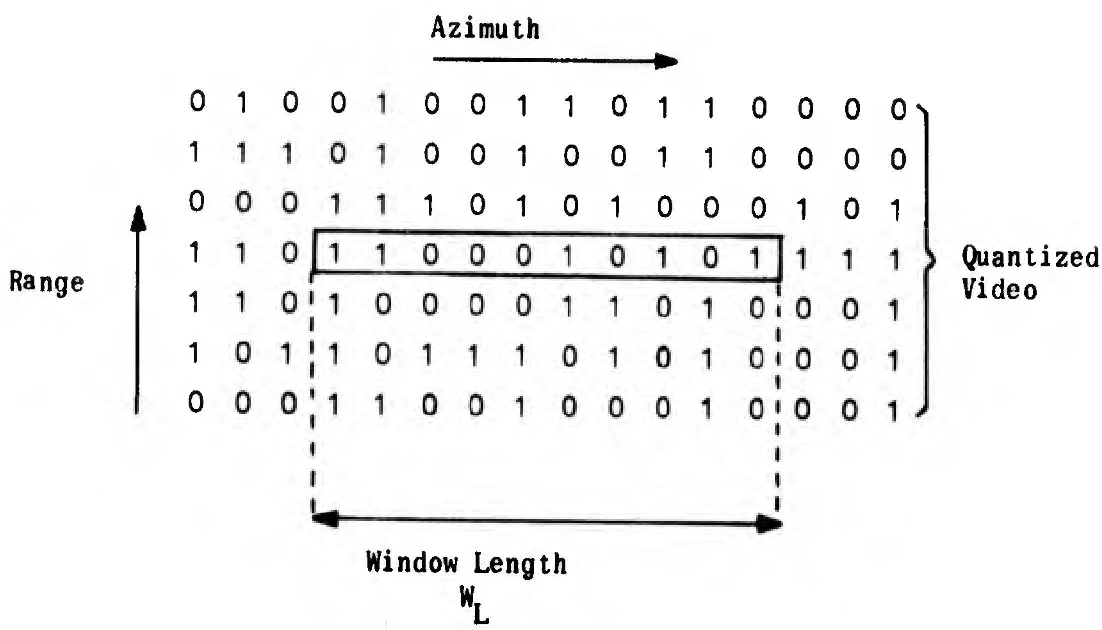


Figure 19. Illustration of a Typical Sliding Window

With respect to the probability of detection in the vicinity of targets consider a window at a fixed range. The probability that a one occurs changes with each sweep position. Commonly in literature this probability (p_s) is treated as a constant over all sweeps covering the target signal. In reality this probability increases as the radar points more directly at the target and then decreases as it sweeps beyond the target. Probability curves showing this pattern are given in figure 20. Based on a quantizer threshold set for ten percent noise, graph 1 shows how the probability varies with signal strength. Graph 2 shows how the probability curves for a target of fixed signal strength change with thresholds corresponding to five and ten percent noise. The computation of the curves assumes the radar parameters stated earlier and a $\frac{\sin X}{X}$ antenna pattern.

Based on the probability curves described above the probability P_d of detecting a target with a sliding window can be determined. The probability P_d naturally changes with target signal strength, noise probability, and position of the window relative to the beam center. Figure 21 illustrates several examples of how P_d is affected by the above variables.

When a target is declared with SWD, its range is determined by the range position of the window; however the azimuth requires more calculation. The conventional manner of determining azimuth is to consider the position of the window at target detection as the leading edge of the target. Subsequently the trailing edge of the target is declared when the hit count falls below a second threshold. The target azimuth is then taken as the midpoint of the leading and trailing azimuth, adjusted for a possible bias.

Straight forward implementation of SWD in the computer would be impractical because of the processing time requirements. Considering the detector as described above and subject to the limit that the threshold is 15 or less, the processing time per second is given by the following formula:

$$(f(69N_W+7)+121N_R)\mu$$

Using the standard parameters, the time amounts to about 200 percent. This clearly would not be a reasonable approach even if two IOPs could be allocated to the radar video processing. An approach which makes computer implementation of target detection feasible is described in the next paragraph.

(B) Predetector/Final Detector. As seen in the last paragraph, the straightforward implementation of SWD in a digital computer is not economical because of the inordinate amount of time required. The predetector/final detector (PD/FD) approach circumvents this problem by using a statistcally weak, quick test followed by a conventional window type test.

This concept is further brought out in figure 22, which schematically illustrates the flow of noise and target data through the process. Considering the amount of quantized video received relative to the amount of

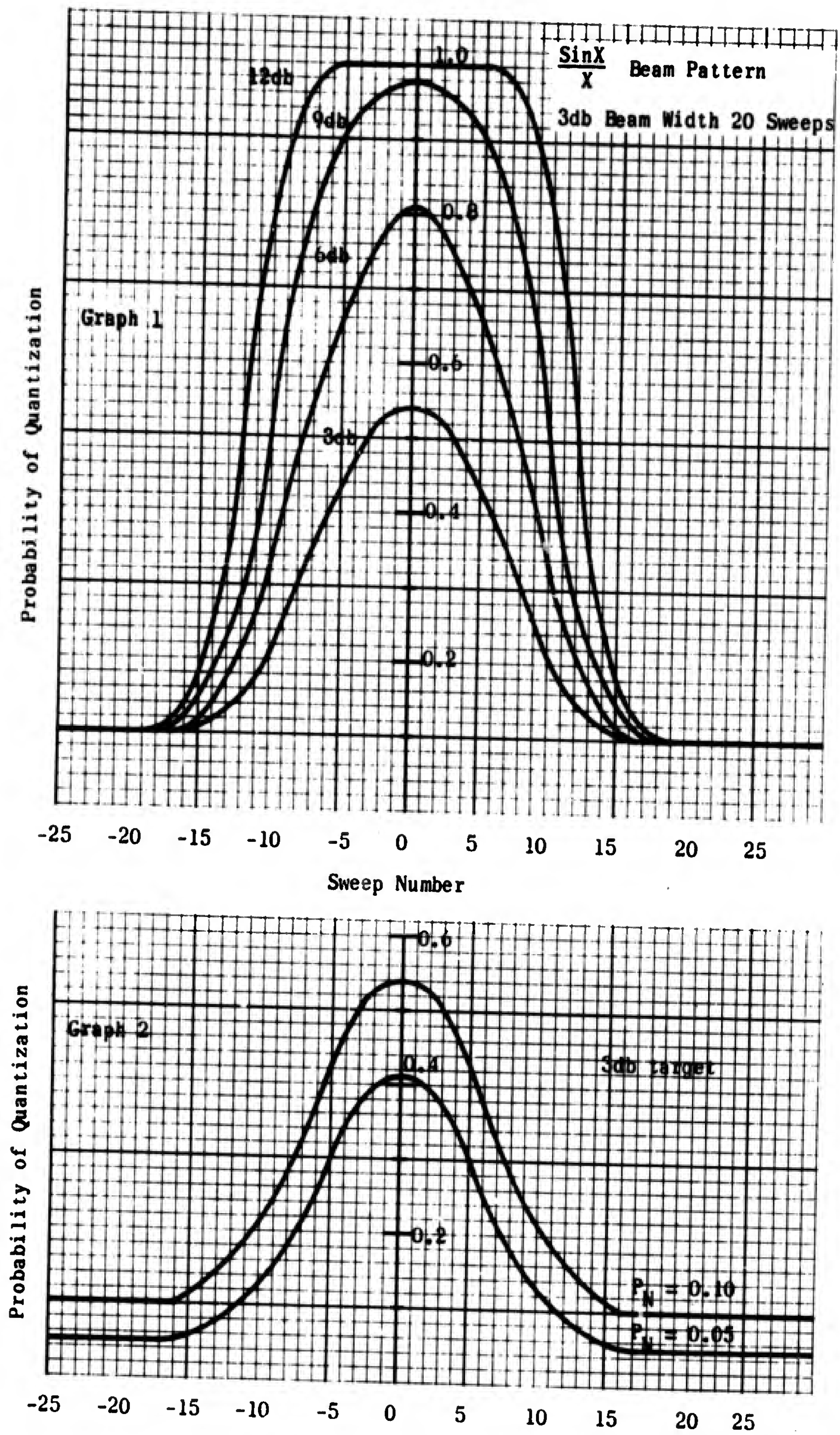
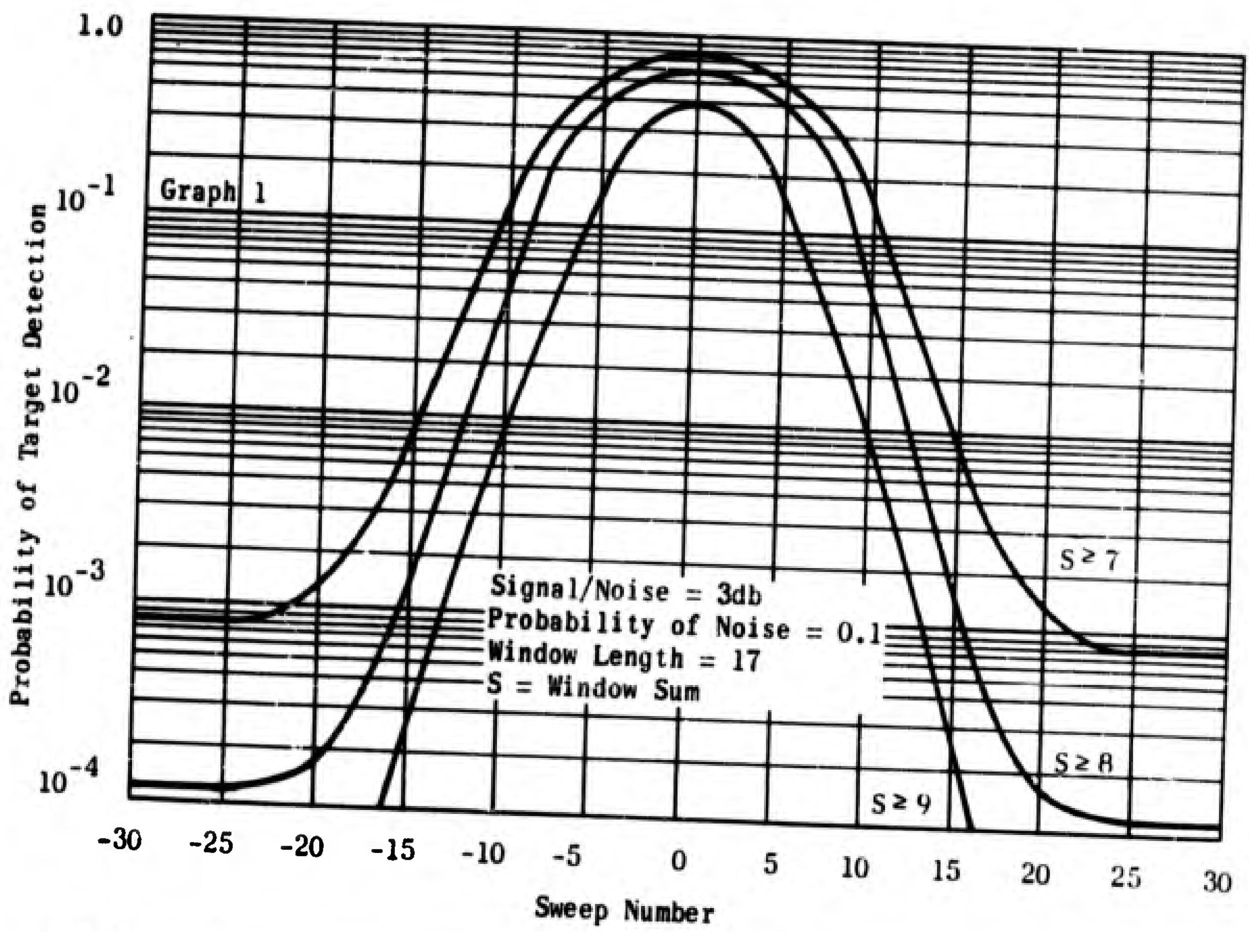


Figure 20. Probability Ensembles



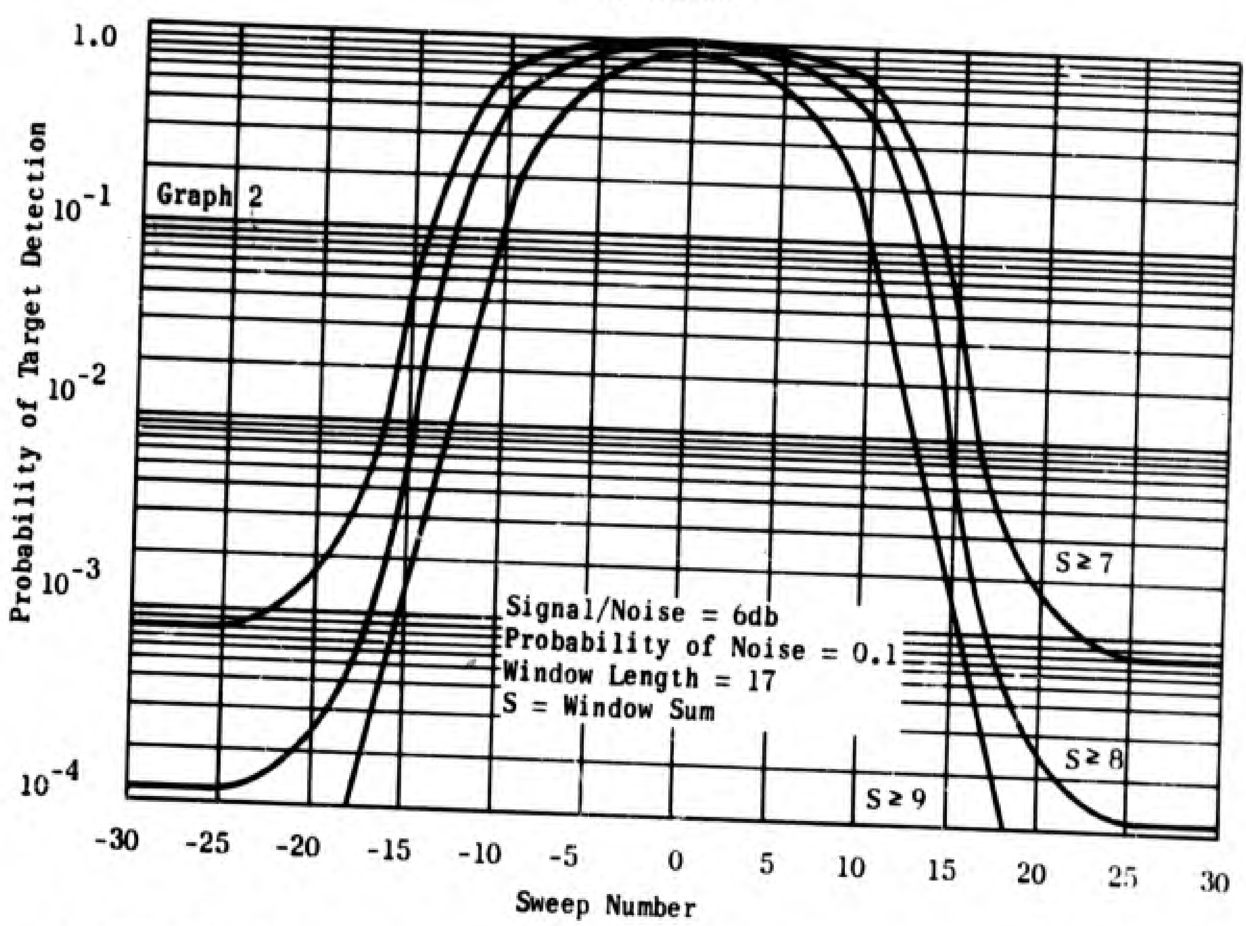


Figure 21. Variation of Target Detection Probability with Sweep Number

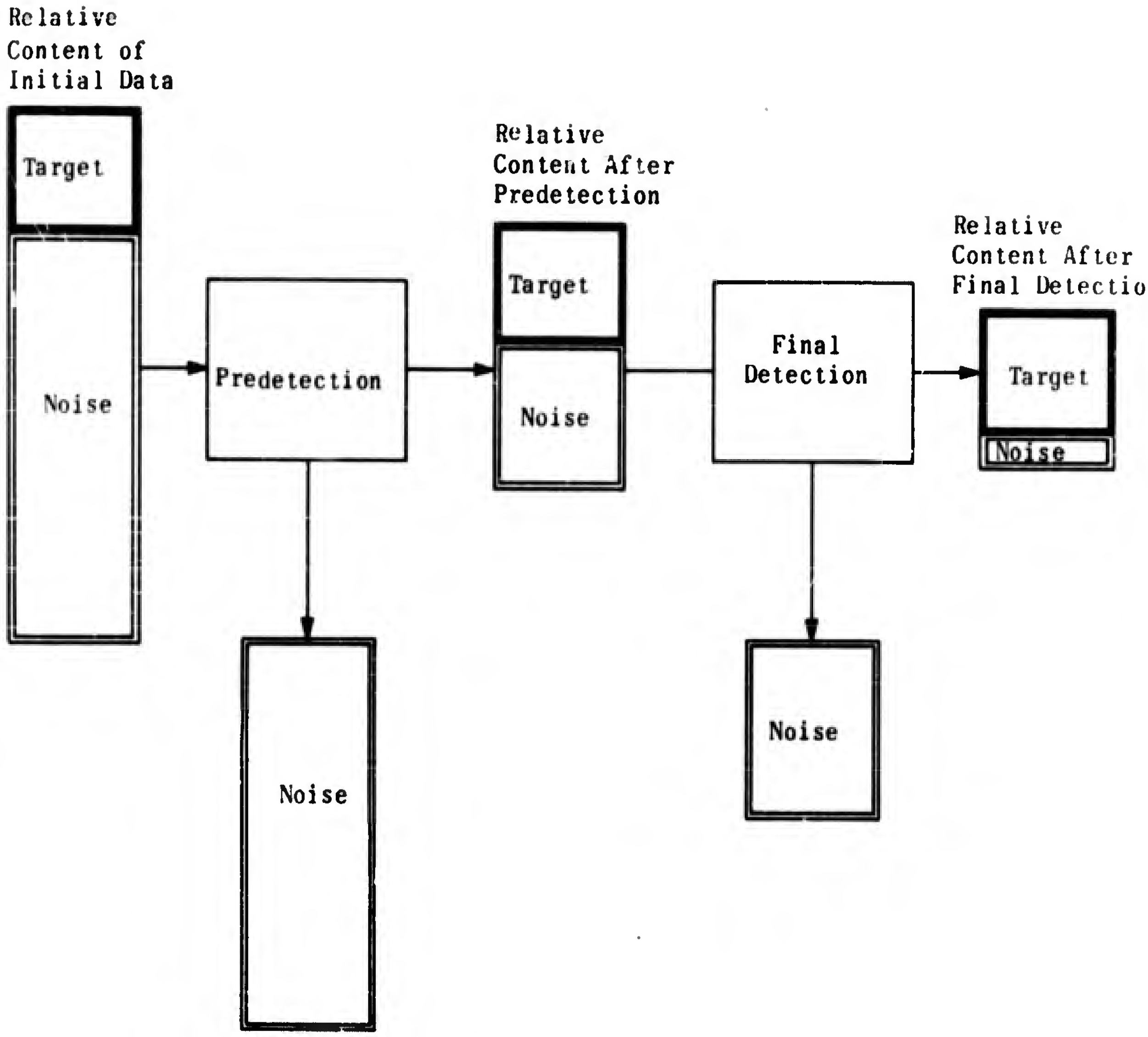


Figure 22. Flow of Noise and Target Data

target information, the input to the predetector is principally noise. (For 500 targets per scan with 20 sweeps of signal per target, only 0.2 percent of the quantized video contains target information). The function of the predetector is to significantly alter this percentage, to approximately 50 percent. The predetector would rapidly eliminate most of the noise data while not significantly rejecting target data. The final detector then operates on the quantized video passed by the predetector. The function of the final detector is to reject more noise so as to attain the desired probability of false alarm (P_{fa}). Normally a false alarm probability on the order of 10^{-6} would be sought.

The predetection process can be allocated to either the digitizer or the IOP. Different algorithms might be used for the hardware as opposed to the software approach; however the function of the predetector would remain the same. Both of these approaches are investigated in a later paragraph.

The final detector is similar to the sliding window detector, except that this process is engaged only when a predetection has occurred. Various algorithms can be utilized for the final detector. These are investigated in the next paragraph. However, for all the algorithms the operation is basically the same. On the basis of the range and azimuth detected by the predetector, a window of quantized video is examined. The placement of this window is such that it is generally centered over the predetection location; however this is not a requirement. Based on the observed data a decision is made as to whether a target is present. Intuitively one might expect that the PD/FD concept might cause some loss in the probability of detecting targets; however as demonstrated by the empirical analysis in Appendix B, the loss is insignificant for the two types of predetectors considered. For example, if the P_{fa} is in the range of 10^{-7} to 10^{-5}, the detectors analyzed yielded probabilities are within two percent of one another.

(1) Final Detection Algorithms. Three final detection algorithms are investigated. The first is similar to the conventional sliding window detector, except that the sliding operation is replaced by the predetection process. This algorithm, called the Fixed Threshold Detection, takes a window of data and compares the number of binary ones against a fixed threshold M. The second detector is the Automatic Clutter Eliminator technique which is employed in the Terminal Modified Radar Video Data Processor (TMRVDP). The last method is called the Weighted Summation Detector. Each of these techniques is analyzed to determine the amount of computer time and memory requirements for a specific number of targets per second. The purpose of this analysis is to bring out the relative processing costs among the various methods. In a later paragraph, the computer processing regarding the impact of cascading the final detector with the predetector will be explored.

• Fixed Threshold Detector. The basic idea of the fixed threshold detector is that a high portion of ones in a window is an indication of a target. The window covers N sweeps at a fixed range. Anytime the number of ones equals or exceeds a threshold a target is declared. The threshold is chosen so as to attain a given probability of false alarm, i.e. to control the number of cases in which targets are declared from noise.

For the analysis which follows a window length $W_L = 17$ and probability of false alarms $P_{fa} = 10^{-5}$ are chosen. The threshold (T) corresponding to this P_{fa} and W_L depends on the noise level in the quantized video. Table XII gives the four noise levels and the corresponding thresholds considered in the following analysis.

Table XII. Fixed Threshold Detector Data

Noise Level (%)	Threshold	Processing Time for Early Termination (%)
10	9	1.92
7.5	8	1.98
5	7	2.04
2.5	6	2.09

Two ways of programming the fixed threshold detector are considered. The first method examines all W_L range cells in the window. In the second method, called early termination, both the number of ones and the number of zeros are counted until either count reaches a critical value. These values are T for the ones and $W_L - T + 1$ for the zeros. The processing time for the fixed length sampling method (first method) is

$$N_R(10W_L + 36)\mu$$

or about 1.9 percent for the standard parameters. The time for the early termination method varies with the strength of the target and the thresholds. Most targets are strong and produce nearly solid sequences of ones, hence the threshold T should be attained soon after counting a minimum of T cells. However, for this time analysis; it is assumed that the last required one (Tth) is obtained in the last cell of the window (W_L-th cell). The formula for processing time is then

$$N_R(13W_L - 6T + 38)\mu$$

which corresponds approximately to 1.9 percent for T = 9. Processing times for other thresholds are given in table XII.

Hence, under the foregoing assumptions which are worst cases, the early termination method is not significantly longer than the fixed length sampling method. As will be seen later the average processing time for rejecting noise is considerably less for the early termination method. Both methods provide identical performance in terms of accepting or rejecting targets. Both methods would permit selection of thresholds such that false alarm probabilit'es in the interval of 10^{-7} to 10^{-3} could be provided. In particular any value of T between 0 and 17 would be possible.

- Automatic Clutter Eliminator. The fixed threshold detector as described above has the drawback that the noise probability for quantized video must be constant and known. This is required to determine the threshold corresponding to a fixed probability of false alarm. This problem can be avoided by using the method referred to as the Automatic Clutter Eliminator (ACE) in the TMRDVP. With ACE a sample of quantized video surrounding the target window is used to estimate the noise level. In this way, the detection threshold (M) may be adjusted to compensate for a change in the noise level and still maintain the specified P_{fa}. The formulation of this technique as a statistical hypothesis is presented in Appendix C.

The operation of ACE involves counting the number of ones in a target window and also in windows on each side of the target window, see figure 23. That is, if W_0 represents the target window, then the total number of ones contained in windows W_{-k} through W_k is obtained. Based on this count, called the noise sum, a detection threshold is determined such that the probability of false alarm remains constant. If the count in the target window exceeds the detection threshold, a target is declared. The thresholds can be determined via an expression given in Appendix C.

The processing for the ACE technique places greater demands on the computer; although special techniques can be used to make the processing feasible. In addition to being a function of W_L, the processing time and core requirements are dependent on the number of windows used in the noise sum. For purposes of this analysis, a clutter sum utilizing nine windows centered in range on the target window is assumed. The expression for computer processing time is

$$N_R(17W_L + 80)\mu$$

This amounts to 3.4 percent for the standard parameters. The memory required is 1,132 words under the following assumptions:

1. Nine windows for the noise sum
2. Four false alarm levels (P_{fa} curves), e.g. 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}
3. Individual selection of P_{fa} curves on the basis of 2 mile by 32 Azimuth Change Pulse (ACP) zones.

The threshold values would vary over 0 to 17. The processing time would not change if less than nine windows were used, but the memory requirements would be reduced.

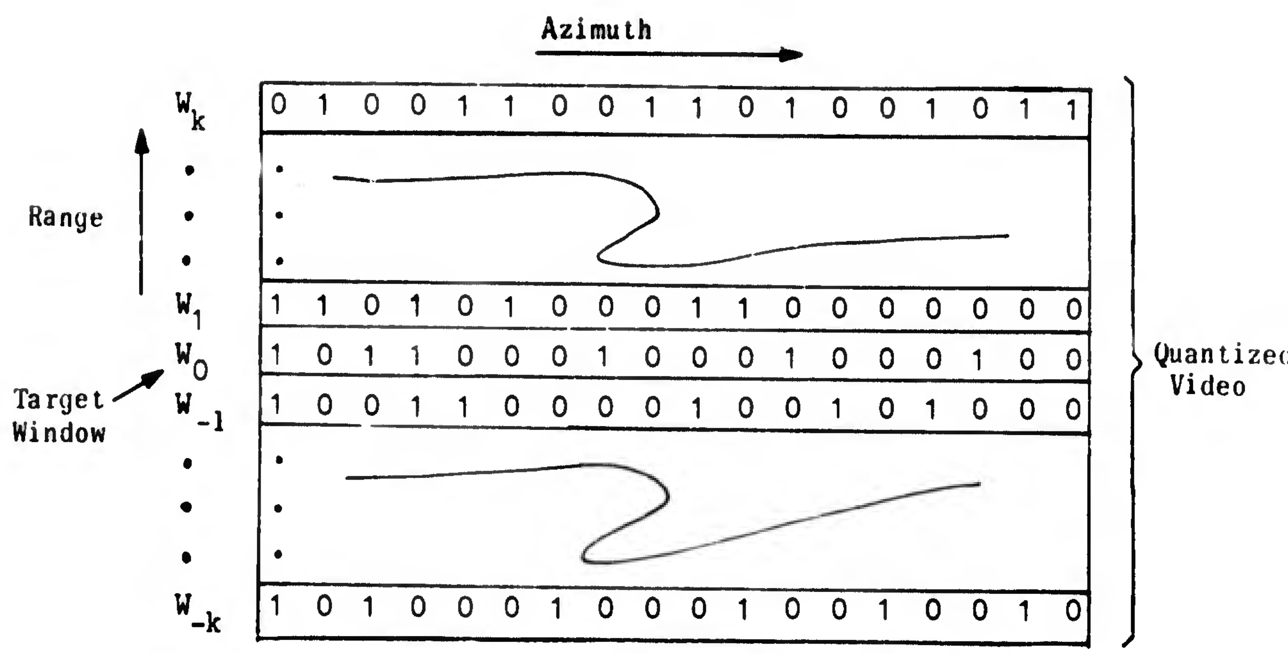


Figure 23. Illustration of Windows Used in the ACE Technique

• Weighted Summation. The third method for final detection permits weighting the quantized video differently for each cell. For example, targets hit patterns tend to have more dense concentration of ones in the center and somewhat less dense concentrations at the edges. Quantized video from radar receiver noise would tend to have the same density of ones everywhere in the window. By giving more weight to the data in the center of a detection window, it is statistically possible to define a more powerful decision process. However, to achieve this, additional assumptions must be made about the quantized video. In particular, an estimate of the target signal strength is necessary. At any rate, it is of interest to determine the computer requirements for this type of detector under the following type of detection criteria:

$$\sum_j w_j X_j \geq T_{wt}$$

where the w_j are arbitrary weights, X_j represents the value of the binary video and T_{wt} is the threshold chosen to attain a given level of false alarms on a known noise level, e.g. P_{fa} for 10^{-7} to 10^{-3}. The weights are assumed to be positive and normalize such that their sum is one, i.e. $\sum W_j = 1$. The computer requirements for this detector are very similar to the fixed threshold, fixed length detector discussed earlier. The processing time is given by

$$N_R(11W_L + 36)\mu$$

which equals 2.1 percent for the standard parameters. The memory requirements are slightly higher because an additional table of weighting factors are required. Nevertheless the memory is still less than 100 words.

(2) Predetectors. The predetection process could be assigned to either the RDAS or the IOP. If performed in the IOP it is necessary to find a predetector which requires little processing time and yet performs well in the sense of predetecting a target. If assigned to the digitizer, hardware logic is the principal concern. In either case the predetector must reject most of the noise data so that the number of calls for the final detector is small.

The analysis which follows examines the requirements for processing time and the capability of the predetector to pick out targets. The processing time is determined by computing the time to perform both the predetection process and the final detection process (when the predetection condition is satisfied) under a known noise level in the quantized video. The noise level is defined by the probability that the quantizer produces a binary one when the analog signal only contains receiver noise. Four noise levels are considered, 10, 7.5, 5 and 2.5 percent. The processing times provided are average values based on probability models for the predetectors. To avoid the problem of quoting predetection time values for each combination of final detectors and noise levels, a constant execution time of the final detector is applied. This constant is 400 memory cycles per call. In a later paragraph, the actual processing time for two final detectors is examined. A further assumption on the processing is that the quantization of video is 1/16 of a mile. The capability of a predetector to pass a target is examined by determining the probability that a target of a given signal strength would be predetected. Two signal strengths are considered -- 3 decibels (db) and 6db. The probability is calculated for the event that at least one predetection occurs as the predetector slides over the target data. The probability curves of figure 20 are used to describe the target. Software (IOP) predetectors are investigated in the next paragraph. The analysis of hardware predetectors follows.

- Software Predetector. Rapid processing is obtained by the software predetector through parallel operation on thirty range cells. The predetector utilizes Boolean logic to operate on the thirty range cells of quantized video within a computer word. In addition the predetector utilizes a very short window. The predetectors presented have lengths of three, four, and eight sweeps. The operation of the software predetector is discussed more explicitly in the following paragraphs.

The first predetector, called the three-in-a-row, is presented primarily for illustrative purposes. Conceptionally, it is easier to understand than the two types to be presented later. Further, the three-in-a row predetector provides a good reference for comparing the target detectability for software and hardware predetectors.

The criterion for this predetector is that a predetection is declared if on three consecutive sweeps the quantized video at a constant range is all ones, i.e., three ones in a row as in figure 24. Considering the programming, this operation can be accomplished on all thirty range cells at a time by taking a composite logical product of three quantized video words. Note that the words must relate to the same range interval and be from three consecutive sweeps.

The processing time for the three-in-a-row predetector is the sum of, i) the average time to perform the three logical products and ii) the average time to process predetections via the final detector. In figure 25, graph 1, this processing time is plotted as a function of noise level. Five noise levels (0, 2.5, 5, 7.5 and 10 percent) are used to determine the curve. Linear interpolation is used to represent other values. Notice that the processing time is high even for low noise levels. A companion, target detectibility curve is shown in graph 2 of figure 25. Here the performance for 3db and 6db targets is shown. It can be noticed that even for the 3db targets, which are considered to be very weak, the probability of a predetection is fairly high for noise levels above five percent.

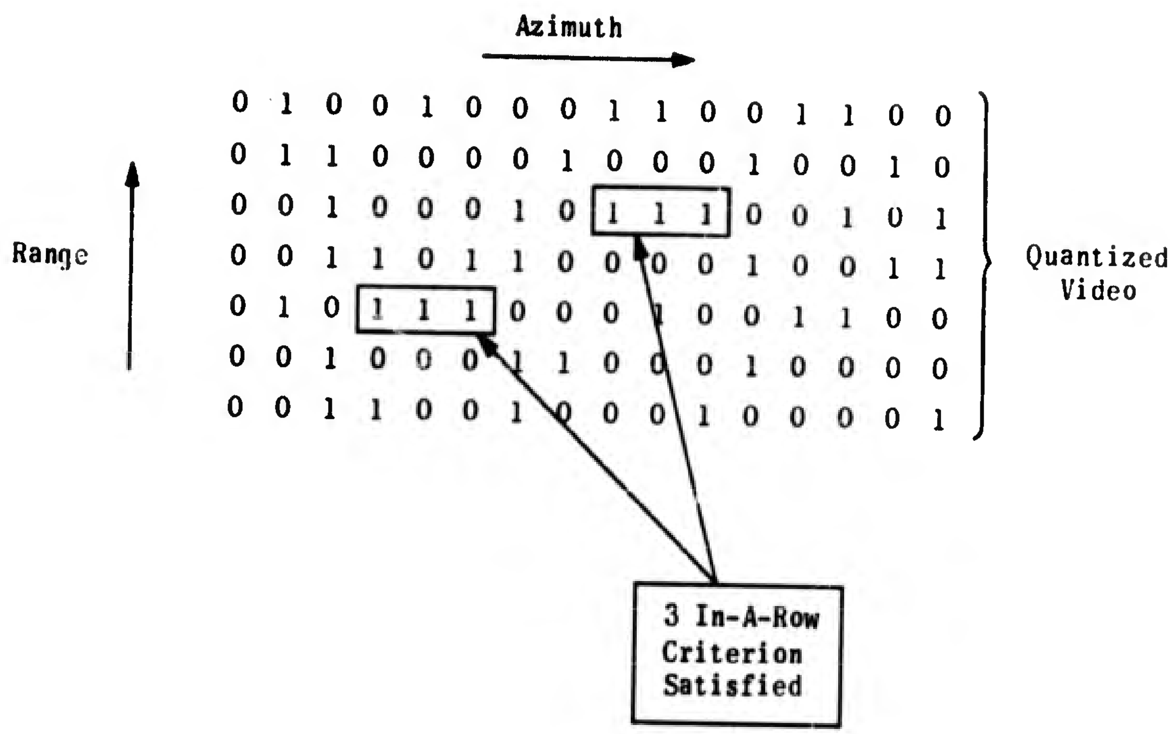
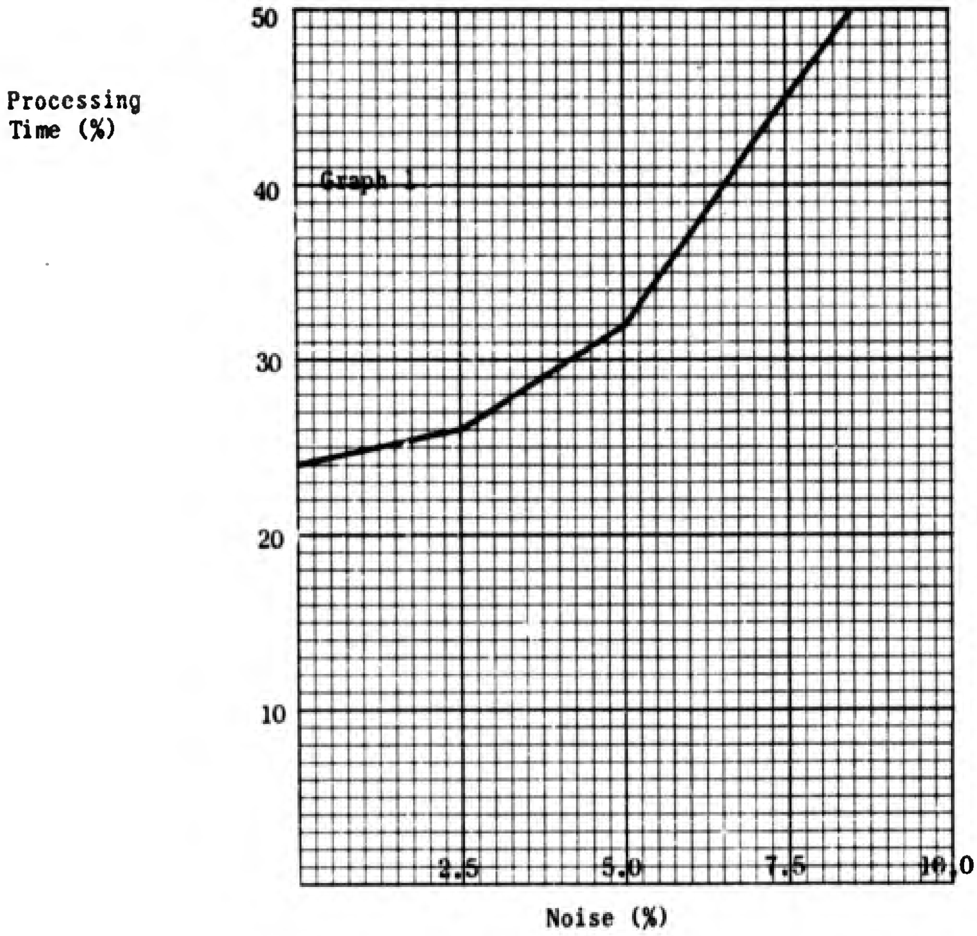


Figure 24. Illustration of a Three-In-A-Row Predetector



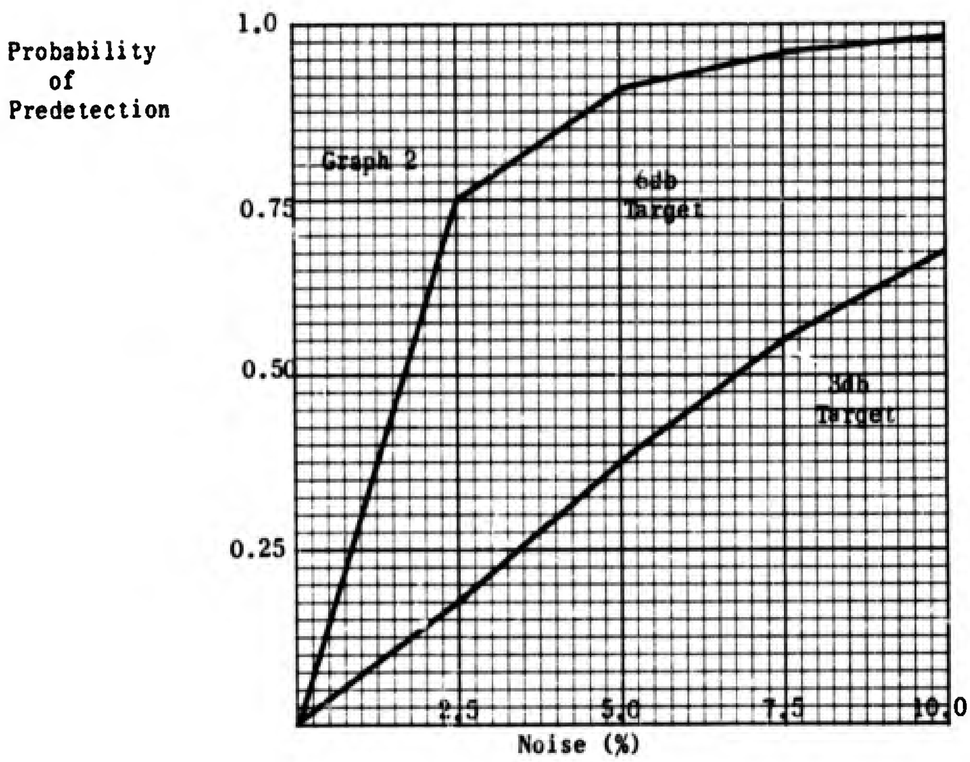


Figure 25. Processing Time and Performance of the 3-In-A-Row Predetector

From the standpoint of computation, the sequential 3/4 is a more efficient predetector. This predetector picks up all three-in-a-row conditions; hence with respect to target detectability the performance is identical to the foregoing predetector. However, instead of using a window of length three and applying the test every sweep, the sequential 3/4 uses a window of length four and applies the test every other sweep. The criteria for predetection is such that out of the four window cells the center two must contain ones and that at least one of the adjacent cells must also contain a one. The Boolean expression for this predetector is:

$$Y = X_2 \bullet X_3 \bullet (X_1 + X_4)$$

where X represents 1 or 0 for window cell j, the plus (+) represents a logical sum operation and the dot (•) represents a logical product operation. In interpretation, Y = 1 implies a predetection; whereas Y = 0 implies none.

The processing time of the sequential 3/4 is considerably less than the three-in-a-row. Graph 1 of figure 26 illustrates the time for this predetector as well as the next one to be discussed. (For reference, the three-in-a-row predetector is indicated by the dashed line.) Note the time for the sequential 3/4 approaches one half of the time for the three-in-a-row at low noise levels. Nevertheless, at the ten percent noise level the processing time is still high. Since all three-in-a-row conditions are picked up, the probability of predetection for the two is identical.

The final software technique is called the sequential 4/8. It operates much like the sequential 3/4 except that a larger window is used. The predetection criteria is that there must be two binary ones in the center of the window and, with respect to the three cells on each side of the center, there must be at least one binary one per side. The Boolean expression for this predetector is

$$Y = (X_1 + X_2 + X_3) \bullet X_4 \bullet X_5 \bullet (X_6 + X_7 + X_8)$$

As in the sequential 3/4, this criteria is applied every other sweep. In the sense of rejecting noise, the sequential 4/8 is a stronger predetector than the three-in-a-row. The effect of this is to reduce processing time for the high noise levels and also to slightly reduce target detectability for a weak (3db) target, see figure 26. Considering the decrease in processing time, the slight loss on weak targets seems acceptable. For a signal strength of 6 db, the loss in detectability nearly disappears. Appendix B provides an empirical comparison of the detection probabilities for a SWD versus a PD/FD technique with the three-in-a-row and sequential 4/8 predetectors. It can be noted that the losses shown there are of the same relative magnitude as those seen in figure 26. In summary, target detection using software predetection does appear to be a feasible approach. There do exist predetectors which reduce processing time to a reasonable level without causing significant loss in target detectability. Of the software predetectors presented, the sequential 4/8 appears to be the best choice.

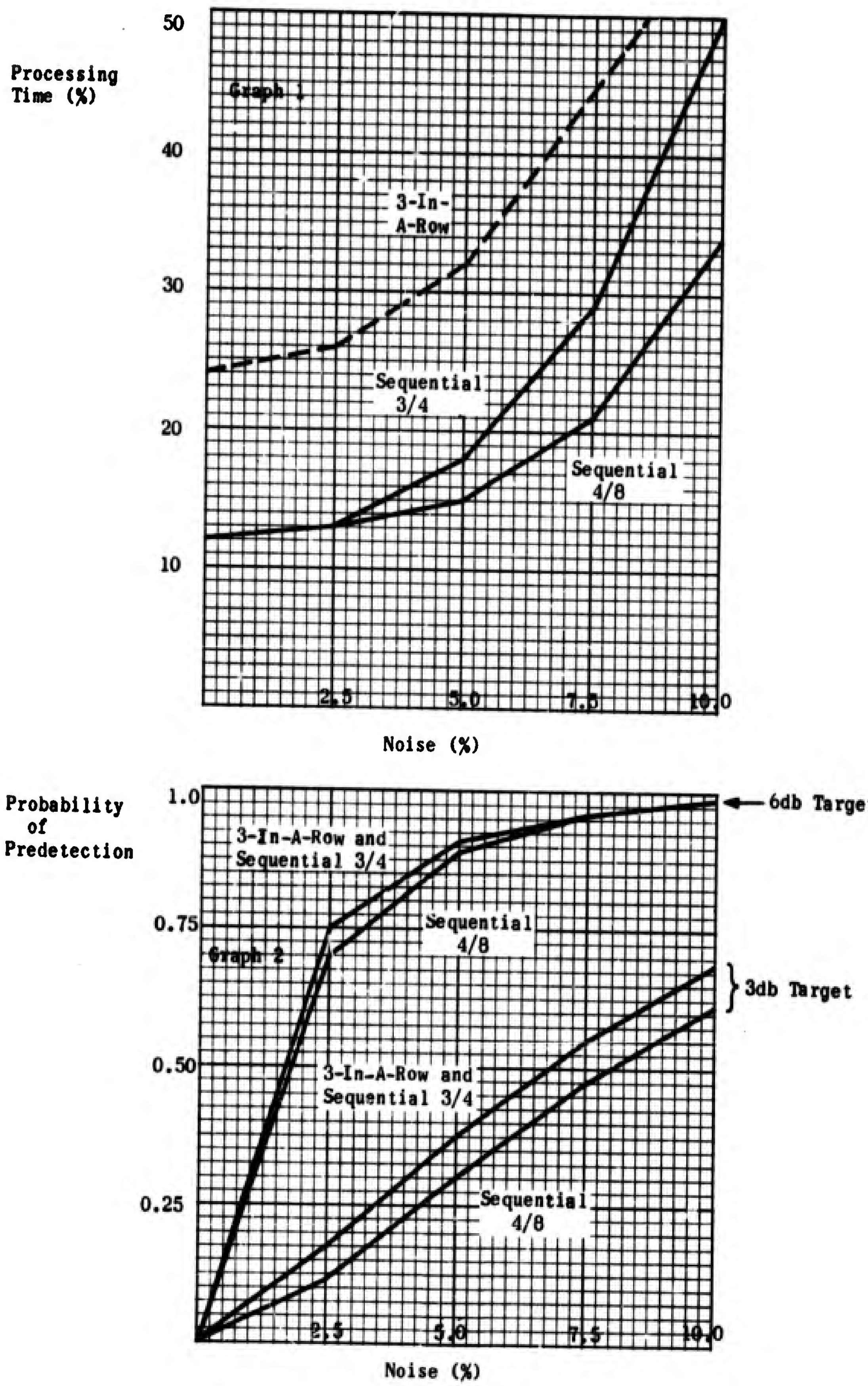


Figure 26. Processing Time and Performance of the 3-In-a-Row, Sequential 3/4 and 4/8 Predetectors

• Hardware Predetector. Putting the predetector in the hardware reduces the computer processing because it no longer has to operate on every quantized video word. Only the quantized video in the vicinity of a predetection is examined via the final detection process. On the other hand, additional predetection data must be transferred to the computer; where it can be examined to determine the location of predetections.

The transfer of predetection data to the computer can be accomplished by transferring the range of each predetection to a computer. This could be done at the end of each sweep. Since ten bits is sufficient to define the range, up to three predetections could be packed into one 30-bit word. More than one predetection word (3 predetections) may be sent to the computer each sweep; however one appears to be sufficient. The justification of this claim follows.

The purpose of predetection is to avoid calling the final detection process when no target information is present in the quantized video. Now, if the number of noise predetections per sweep is high, then the function of the predetector is not fulfilled. To establish an upper limit for this number, consider that a final detector requires about 400 memory cycles for a single execution. If one noise predetection is received each sweep, the target detection process would consume 36 percent of the computer time (based on 3/4 microsecond per memory cycle and 1200 sweeps/second). Now, with respect to targets, assuming an average of 125 per second, one predetection per target, and a pulse repetition frequency (PRF) of 1200, there would be an average of only about one target predetection per ten sweeps. Even under the liberal assumption of nine predetections per target, the average is less than one predetection per sweep. Furthermore, to inhibit the event of producing multiple predetections when over target video, the hardware logic could lock out the predetector for a number of sweeps at a given range whenever a predetection occurs. Therefore, the average number of predetections per sweep is well within the three per sweep limitation.

To account for the possible event of more than three predetections per sweep, the hardware could provide a holding feature so that the additional predetection could be transferred on the following sweep. One suggested scheme would have the hardware set a flip/flop when a predetection occurs and the predetector word is already holding three range values. On the next sweep, a predetection from this range cell would automatically be declared independent of the new video.

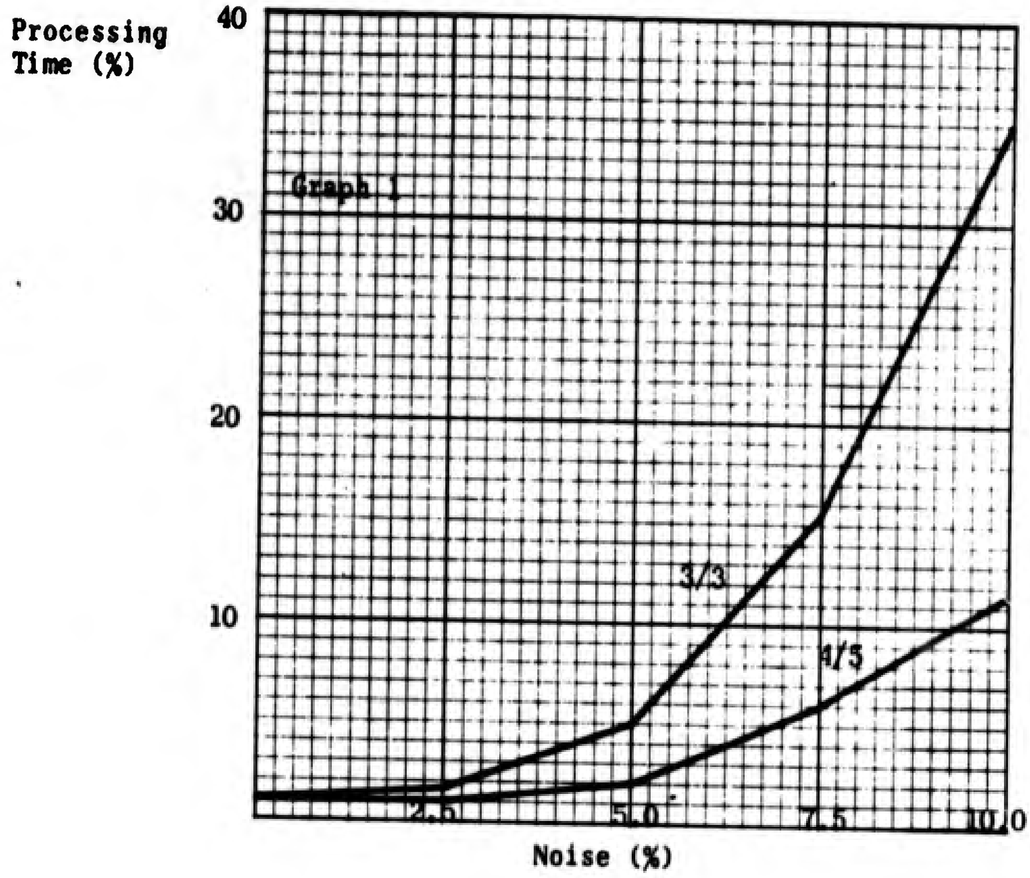
Two classes of hardware predetectors are considered. The first is simply a sliding window detector of very short length. The technique called the sequential observer makes up the second class. The analysis of these hardware predetectors parallels that of the software predetectors. That is, both processing time and target detectability are examined. The processing time due to noise data is considered as the time to examine the predetector word each sweep and the average time spent in the final detector. The assumption of one predetection word per sweep is also used.

Two short sliding window predetectors are presented. The first is the 3 out of 3 (3/3) predetector. This, of course, yields identical target detection performance as the three-in-a-row software predetector. The second short sliding window predetector requires that 4 out of 5 (4/5) window cells must have binary ones. The processing time and the probability of predetecting targets are shown in figure 27. where again the curves are based on five points. It can be observed that the processing time is less than that observed for the software predetectors. Further, the target detectability of the 4/5 predetector is almost the same as the 3/3 predetector, so that the performance of these hardware compares favorably with the software predetectors examined.

The sequential observer is an appealing predetection technique for hardware implementation because it requires less memory per range cell than the short sliding window technique. A sequential observer is simply a counter which is incremented by a given value (K1) for a one in the quantized video and decremented by K0 for a zero. The counter is inhibited from taking on values less than zero by setting the value to zero to zero in such cases. Whenever the counter equals or exceeds a threshold value, T, a predetection is declared and the counter is reset to zero. The performance of two sequential observers is shown in figure 28. Both of these predetectors have K1 = K0 = 1; however the threshold values are 3 and 4, respectively. As an illustration, the sequential observer with threshold three would be incremented one for each one, and decremented by one for each zero. The count of three would indicate a predetection. Both the processing time and the probability of predetecting a target is slightly higher for the sequential observer with threshold three than the 3/3 hardware predetector. On the other hand, with threshold four, the performance indicators both drop significantly. Hence, significant processing time would be saved at the cost of losing some weak targets. Other values of K1, K0 and the threshold could be chosen to obtain performance in between the two sequential observers presented here.

(3) Summary of Target Detection. To illustrate the processing time differences, a comparison is made of two final detectors each operating with two predetectors. The two final detectors are the ACE and the fixed threshold window using the Early Termination (ET) option. These choices illustrate the cost of using the most time-consuming final detector algorithm (ACE) against the fastest final detector algorithm (ET). The predetector choices are the sequential 4/8 software predetector and the sequential observer (T = 3) hardware predetector. The objective here is to bring out the difference in processing time for the best choice of software against the best choice hardware predetector. The processing time for each of the four combinations is shown in figure 29. The time accounts for the processing of both noise and target data (125/second).

First, all four of the combinations are reasonable approaches in terms of processing time, particularly at noise levels of 7.5 percent or lower. Hardware predetection does in general provide significant savings in processing time. Finally, significant time can be saved if the fixed threshold detector could be used. On the other hand, ACE offers considerable protection in the sense of maintaining a fixed false alarm level and does not require unreasonable processing time.



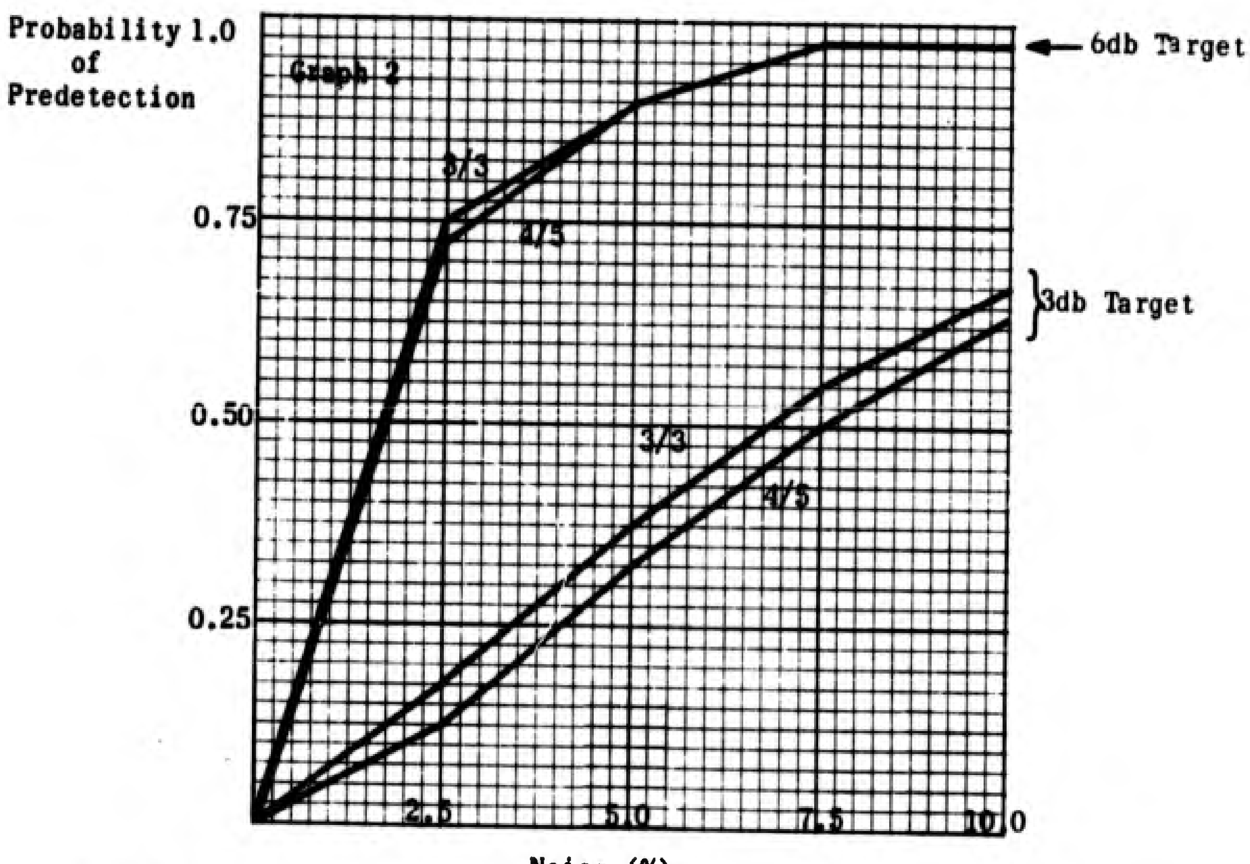


Figure 27. Processing Time and Performance of Short Sliding Windows

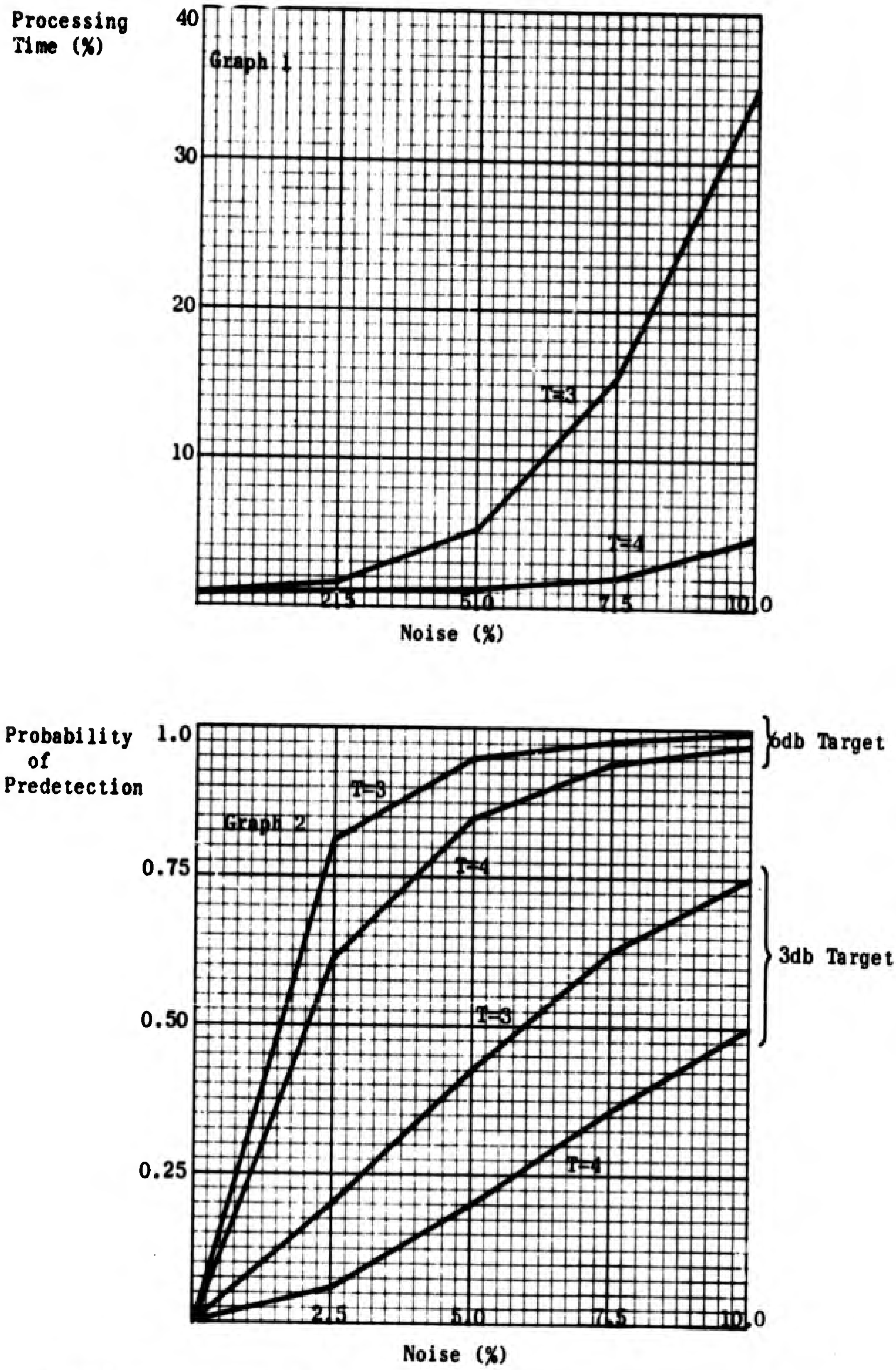


Figure 28. Processing Time and Performance of Sequential Observers

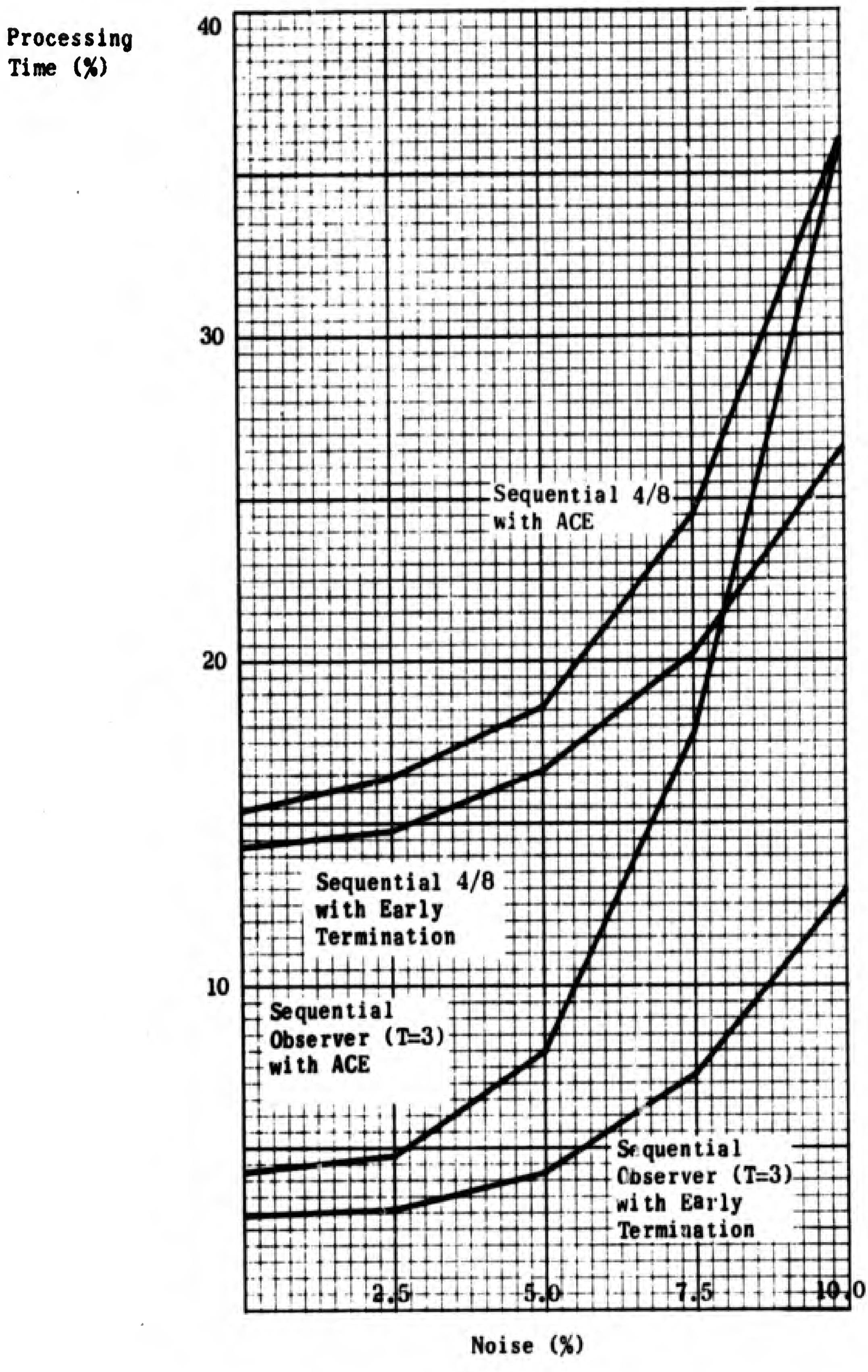


Figure 29. Processing Time for Various Predetector/Final Detector Combinations

(4) Dual Detection on Two Videos. The section entitled Video Selection mentioned the idea that two types of video could be buffered into the computer in parallel. Target detection could be independently attempted on both of these videos. The resulting target reports could then be correlated to produce one report per actual target. This concept enhances the probability of detecting targets since two nearly independent (in the statistical sense) trials for target detection would be made each scan.

The consequence of this concept would be to approximately double both the processing time and the probability of false alarm. The doubling of false alarms could be tolerated if the number of false alarms from a single video is on the order of one to ten per scan. For the processing time there exist other tradeoffs. For example, because of the increased probability of target detection it may be possible to reduce the noise level in the quantized video and still realize the improved detectability. As may be recalled from the data in figure 29, the lower the noise level the greater the reduction in processing time. Considering the ACE detector with a sequential observer, dropping the noise from 7.5 percent to 5 percent decreases the processing time from 17.8 to 8.0 percent. The processing time of the dual scheme at the 5 percent noise level, plus the additional input time (2.9 percent) would be just slightly more than single processing at the 7.5 percent noise level. In particular, the time values would be 18.9 percent and 17.8 percent, respectively. The analysis of the improvement in target detection probability is not available, but it could be anticipated that a net improvement would be realized.

Another interesting alternative would be to let the digitizer perform the target detection function on one video and let the IOP perform the detection on the second video. The two sets of reports would then be correlated in the same manner as considered earlier. In this case, the processing time would be essentially the same as if software detection of only a single video was considered. This results because the additional time to input target reports from the digitizer is insignificant, i.e. approximately one percent. In conclusion, the concept of processing two types of videos in parallel warrants further study and consideration.

(C) Beamsplitting. The technique of beamsplitting via the conventional sliding window has already been discussed. It is of interest to consider another approach called the center of density beamsplitter. This paragraph investigates the processing requirements for these two methods. The input for the beamsplitting process is considered as being the detection range and azimuth. The function of beamsplitting is then to find a best estimate of the target azimuth.

Beamsplitting via the conventional sliding window takes the mid-point of the target leading and trailing edges as an estimate of the target azimuth. Determination of the leading and trailing edges in the computer requires successive sliding of the window and testing the count against the threshold. The processing time is given by the expression

$$42\ W_L\ N_R\ \mu$$

which corresponds to 6.7 percent for the standard parameters.

The center of density concept computes the arithmetic average of the azimuth position at which hits are received. The operation is analogous to finding the balance point of a bar load with unit masses at various points along its length. The azimuth estimate ($\bar{\theta}$) is computed by

$$\bar{\theta} = \frac{\Sigma\, \theta_j X_j}{\Sigma\, X_j}$$

where θ_j is the azimuth of the j-th sweep and X_j is the binary value for the j-th sweep. To insure that the azimuth estimate is not biased due to the initial placement of the window, an iterative process is incorporated. That is, on the basis of the data in the initial target detection window, a center of density azimuth is calculated. Then, another window is centered about this new point, and a second azimuth is calculated. This process is repeated until two successive azimuth estimates are essentially the same. To avoid the recounting of the video data within successive windows, the window length is enlarged at each iteration. As a consequence, the window at the end of the iteration process generally covers the entire target and thus helps to minimize biasing the estimate. The number of iterations is generally about four. The processing time is determined by

$$(20\, W_L + 176)\, N_R\, \mu$$

which corresponds to 4.8 percent. Furthermore, when this beamsplitting method is combined with the fixed threshold window or weighted summation detection, the processing time is reduced. This happens because $\Sigma\, \theta_j X$ and $\Sigma\, X_j$ can be calculated during the detection process at no extra cost. The beamsplitting time in this case is given by

$$(10\, W_L + 176)\, N_R\, \mu$$

or 3.2 percent when using the standard parameters.

Ancillary Functions

This section addresses the remaining processing functions to be considered in the report. These are the beacon processing, correlation of radar and beacon data, and the utilization of tracking feedback information.

(A) Beacon Video Processing. The assumption of this study is that the beacon processing is accomplished in the manner presently defined for the ARTS III System. Briefly in review, this concept utilizes a hardware device called Data Aquisition Subsystem (DAS) to extract beacon replies from the analog video. The role of the digital computer is to detect and locate valid beacon reports.

The computation requirements for beacon processing are taken from an ARTS III design document (Reference 7). The processing time as a percentage is stated as

$$6.4 + 0.236N_B + 0.028N_F$$

where N_B is the number of beacon reports per second ($N_B = 63$) and N_F is the number of fruit responses per second ($N_F = 100$). Hence, for the standard parameters of this report, the processing time is 24.1 percent. The memory requirement is given by

$$880 + 9N_B$$

which corresponds to roughly 1500 words.

(B) Radar/Beacon Correlation. The goal of the Basic Radar Tracking Level System is to add a radar tracking capability to the present ARTS III Beacon System. Thus far the discussion has concentrated on strictly the radar problem. Now it is of interest to determine what the costs are for combining radar and beacon reports. The correlation criteria considered for this analysis are straightforward. The range of a radar and a beacon report must be within 1/16 mile. The azimuth of the radar report must lie within some fixed interval about the beacon report. If this criteria is met, the two reports are merged into one.

The processing time is not very sensitive to the method of choosing which of the two ranges and which of the two azimuths to use, so the detailed calculation of various options are not provided. The expression for processing time is

$$(9.3f + 70N_R + N_B(26W_L - 23))\mu$$

using the standard parameters, the processing time is about 3.5 percent. The memory requirements would be on the order of 200 words including program and data areas.

(C) Tracking Feedback. If an estimate of the position for a target is known, it is possible that improved system performance can be obtained by applying this information in the video processing task. The assumption for this investigation is that the predicted position of targets handled by a tracking program is available. The question of interest is what could be done with this information and what would be the computer costs.

Two uses of tracking feedback are considered. The first is the adjustment of the threshold used in the quantization process. For optimum system performance it is usually considered best to operate at a fixed noise level. This optimum noise level may be different for regulation of clutter than for receiver noise alone. However, to enhance detection in the vicinity of a known target, another threshold may be preferred. Furthermore, if the target in question produces a weak signal, then it is certainly desirable to improve the probability of detection, even though a slightly higher level of false alarms results. Second, even if the target in question is strong, it is of interest to maximize the detection of other weak but perhaps untracked targets in the vicinity. The objective here is to provide better data for target collision avoidance. Also with respect to the Scan Correlated Feedback (SCF) threshold control, information that a target is in a particular zone could be utilized to inhibit the raising of the threshold because of the added number of ones received from the target.

The second use of tracking feedback is to adjust the threshold during the target detection process. For example, suppose that ACE is the chosen detection technique. When a target is known to be in a given zone, it might be desirable to relax the target detection threshold by switching to a higher P_{fa} curve, e.g., 10^{-4}.

Whatever the motivation of the threshold adjustment, the principal objective of this section is to estimate the computer requirements for this type of operation. Implementation of this function can be handled by literally constructing a range-azimuth map of the predicted target positions. This map would have the same granularity as the zones utilized in the SCF and ACE functions, e.g., 2 miles by 32 Azimuth Change Pulses (ACPs). The map would then indicate the zones in which threshold adjustment is needed.

The map would be constructed in the computer on a scan-by-scan basis. One bit would be required for each zone. The bit would be set to one when a target is predicted in the zone. Furthermore, when the track is very near the boundary of the zone, it would be desirable to also set the bit for the neighboring zone to one. This would compensate somewhat for the possible inaccuracy of the predicted position. Figure 30 illustrates four cases in which the zone would be expanded. The first is that the predicted position is in the interior of the zone, hence only that zone is marked. The second is that the position is near a range boundary. Then the adjacent zone in range is also marked. A similar situation is permitted when the position is near an azimuth boundary. Last, when the position is in a corner of one zone, three adjacent zones are marked.

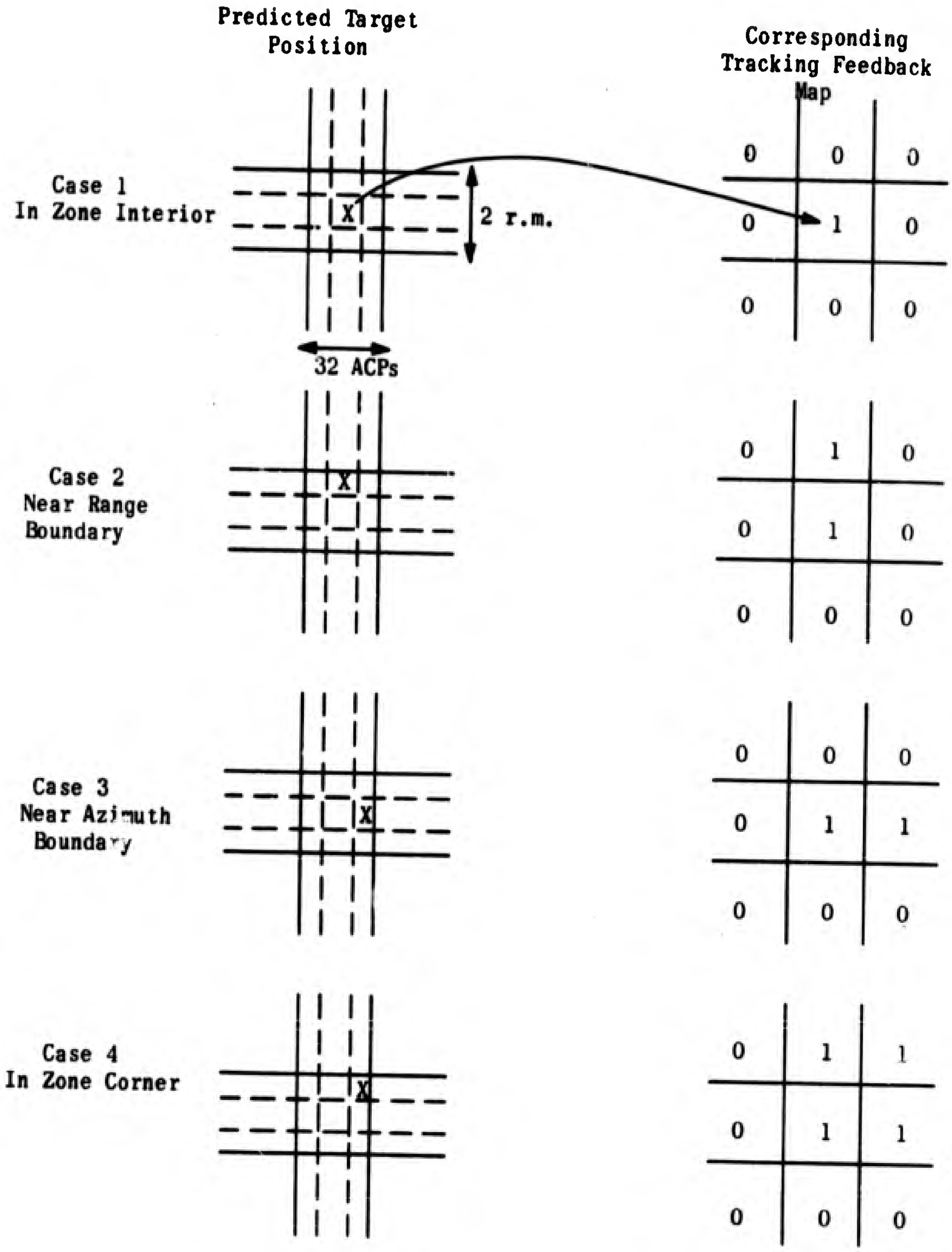


Figure 30. Illustration of Mapping Predicted Target Position

Computer requirements for the mapping process are small. The map would be constructed in segments where each segment is one radar quadrant (90°). The processing time under the assumption that 2 miles by 32 ACP zones are utilized and that a list of tracks within a radar quadrant is available, the processing time is:

$$(170 + 51\ N_R)\mu$$

This corresponds to 0.5 percent for a track capacity of 125 per second. The memory requirements would be about 200 words including program and map areas.

Adjusting the quantizer thresholds for SCF would be accomplished through establishing a new threshold based on the present threshold, the presently designated noise level, and the desired noise level. This calculation would be of the form

$$T_2 = \frac{T_1\ K_1}{K_2}$$

where T_1 and T_2 are the old and new thresholds respectively, K_1 and K_2 are the constants determined by the old and new commanded noise levels. On the basis of 2 miles by 32 ACP zones, the processing time is

$$(102\ N_R + 35{,}520)\mu$$

which is 3.6 percent for the standard parameters. The memory requirements are under 100 words.

Adjustment of the threshold for the ACE detector requires selecting a different P_{fa} curve when a target is predetected in a zone. The time for this operation when added to the normal ACE logic is

$$(6\ N_R + 0.09\ N_W\ f)\mu$$

which comes to 0.3 percent. The additional memory requirement is insignificant. In summary, the total computer requirements for the tracking feedback functions described above are 4.4 percent for processing time and about 300 words of memory.

The tracking feedback technique described above is a very simple approach which provides a binary indication for each zone, i.e., target present or not present. Thus, selection of one of two P_{fa} (or P_n) values is provided. More sophisticated techniques could be implemented which would provide for a greater number of P_{fa} (or P_n) selection. Such a method might be one which makes use of a track quality measure.

Conclusion

The material of this report is consolidated through presenting numerous approaches to the allocation of functions into hardware and software. These ideas are brought out by illustrating several complete system configuration models.

In the first approach, the Radar Digitizer is virtually the same as the Terminal Modified Radar Video Data Processor (TMRVDP); in conjunction, the software functions are basically limited to beacon processing and its integration with radar. In succeeding models, the processing functions of the computer are progressively increased by transferring functions from the digitizer to the Input/Output Processor (IOP). The final model illustrates the extreme of this scheme. That is, all of the radar functions described in the main body of the report are allocated to the computer. The models presented are not the limit of possibilities. Many other combinations would be possible by considering minor changes in the processing approaches. Nevertheless, the models chosen sufficiently cover the field such that the impact of certain allocations can be assessed.

The presentation of the system models is through a series of tables and figures which summarize the allocation of functions and the computer requirements. All time and memory values are based on the standard parameters used throughout the report. For the purposes of review, the principal parameters are reiterated here:

Range Quantization	1/16 mile
Pulse Repetition Frequency	1200/seconds
Maximum Range	60 miles
Target Detection Window Length (radar)	17
Radar Target Reports per second	125
Beacon Target Reports per second	63
Fruit Density (responses per second)	100

Figure 31 provides a review of the alternatives investigated for the principal radar video processing functions. The alternatives given are the ones which have been shown to be feasible and practical. Furthermore, in cases where minor modifications were discussed, the alternatives represent the preferred techniques. For example, in the discussion of the two-stage Scan Correlation Feedback (SCF) system, techniques for output of thresholds and selection of range densities thresholds were presented. The bi-level and tri-level alternatives of figure 31 utilize the techniques

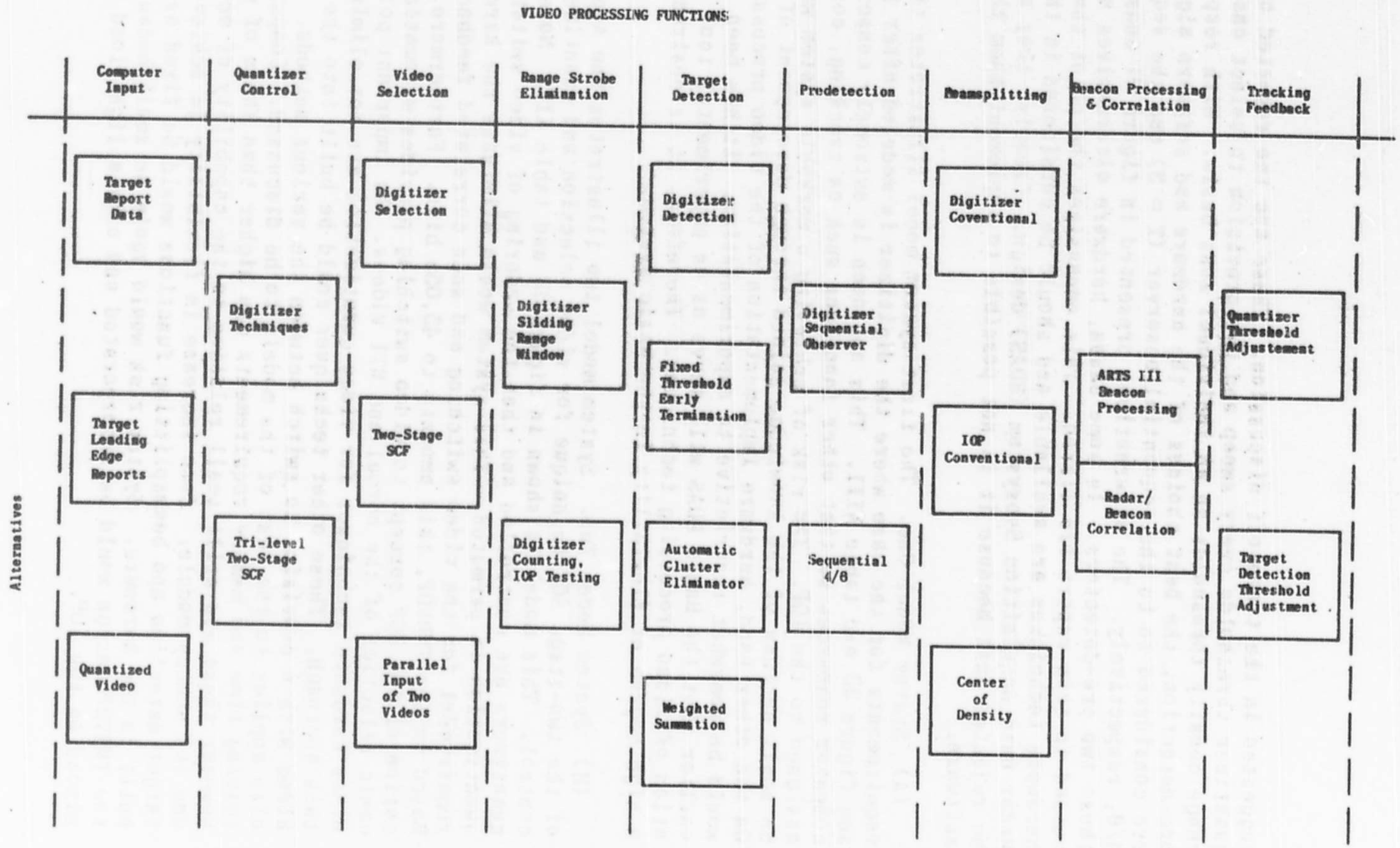


Figure 31. Summary of Video Processing Alternatives

suggested in the trade-off discussion. These are the repeated output of quantizer thresholds every sweep and the provision to select one of two range density thresholds on an individual zone basis. With respect to pre-detection, the best choices of the hardware and software algorithms are considered to be the sequential observer (T = 3) and the sequential 4/8, respectively. The alternatives presented in figure 31 consider only these two pre-detectors. In some cases, hardware alternatives not discussed in this report are listed. This emphasizes the point that other hardware techniques are available and should be considered in the Basic Radar Data Acquisition Subsystem (RDAS) design. Clearly, they are not to be rejected just because it is not possible to implement them through software.

(A) System Model One. The first system model illustrates the computer requirements for the case where the digitizer is modeled after the TMRVDP, see figure 32 and table XIII. This approach is extremely conservative of computer resources so that other functions such as tracking, could be assigned to the IOP. The risk of achieving a workable system would be small because of the knowledge gained through development of the TMRVDP. On the other hand, hardware implementation of the video processing functions would be somewhat restrictive to experimentation. It has been noted earlier that the Basic RDAS will serve as an experimental tool for evaluation of video processing techniques. Therefore, it is desirable to have a high degree of flexibility in the basic system.

(B) System Model Two. System model two illustrates the application of the two-stage SCF technique for video selection and quantizer threshold control. This model is shown in figure 33 and table XIV. Note that two quantizers are controlled and that the ordering of video switching and quantization is permuted. This system would eliminate the hardware memory requirement for the video switching and scan correlated feedback function. Based on the TMRVDP, this amounts to 45,056 bits. Furthermore, the application of the SCF concept to video switching provides automatic and dynamic selection of the normal and MTI videos. An important point is that other hardware techniques for video quantization are not eliminated by this approach. These other techniques could be built into the digitizer along with a provision to switch between the various methods. This point also applies to the rest of the models to be discussed. Computer processing time and memory requirements are higher than those of model one; however these are still small relative to the capability of one IOP and one 16K memory module. Some increase in flexibility is achieved, but target detection and beamsplitting functions would be fixed as initially built in the hardware. System risk would again be small because most of the TMRVDP design would be incorporated and only a light load would be placed on the IOP.

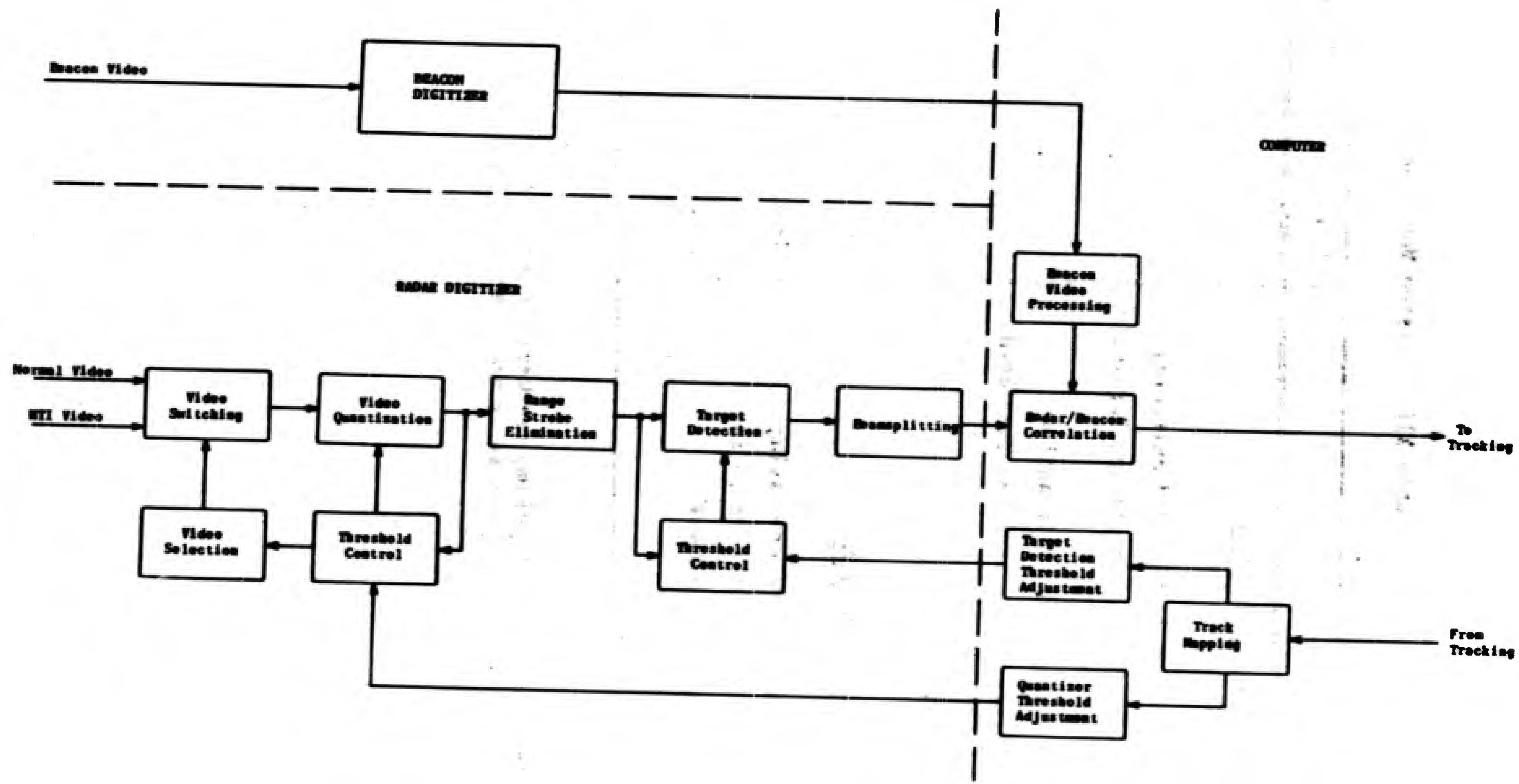


Figure 32. Block Diagram of System Model One

Table XIII. System Model One

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Selection and Switching		
Video Quantization and Threshold Control		
Range Strobe Elimination		
Target Detection and Threshold Control		
Beamsplitting		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Correlation	3.5	200
Tracking Feedback*	4.4	300
Total	32.0	2,000

* The figures for tracking feedback assume scan correlated feedback and target detection would be in the computer; hence these figures are low by about one percent.

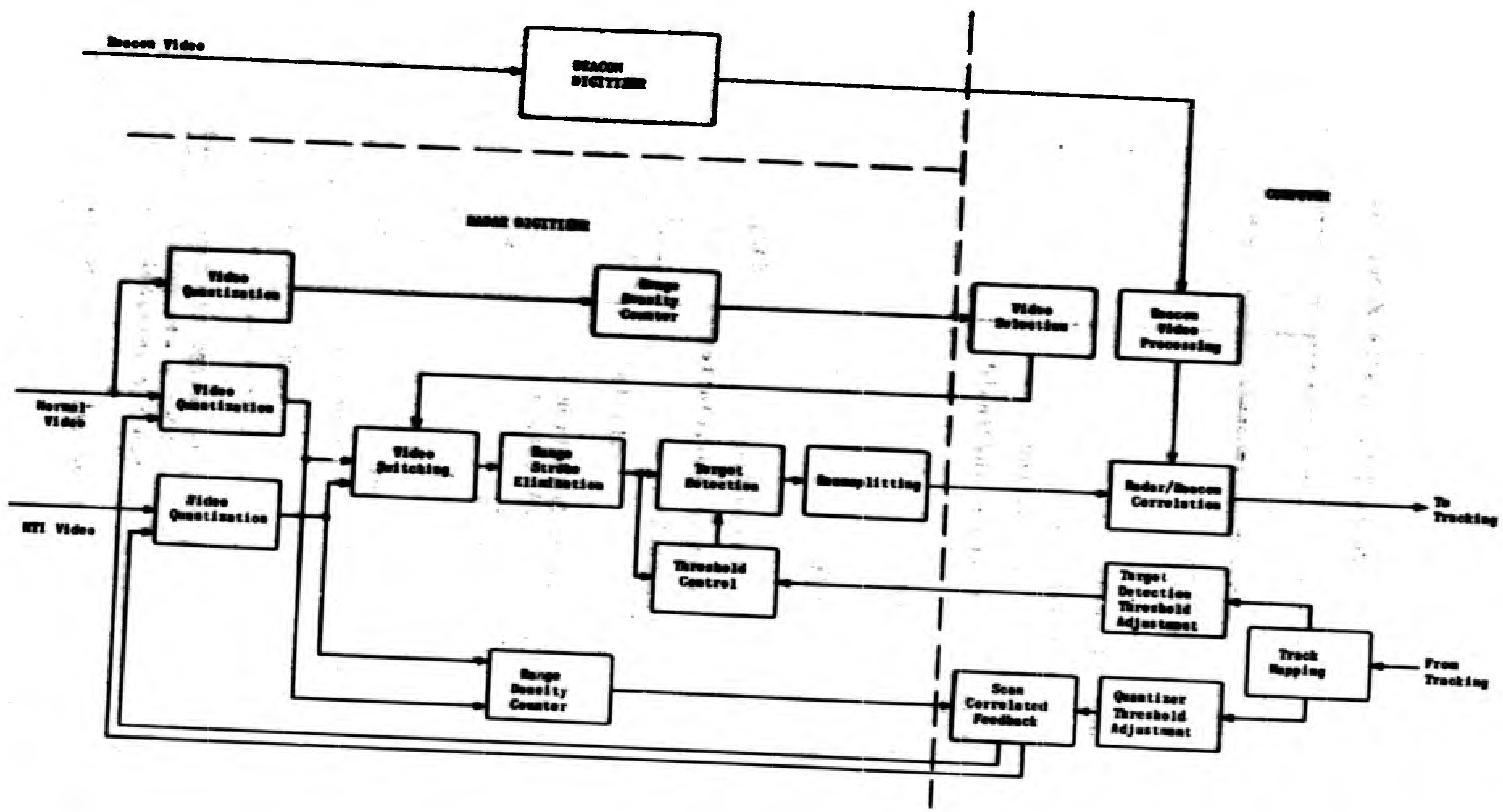


Figure 33. Block Diagram of System Model Two

Table XIV. System Model Two

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Switching		
Video Quantization		
Range Strobe Elimination		
Target Detection and Threshold Control		
Beamsplitting		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	3.5	200
Tracking Feedback*	4.4	300
Subtotal	32.0	2,000
Control of Video Selection and Quantization Threshold		
Video Selection	2.9	1,100
Threshold Control (normal video)**	6.4	2,800
Threshold Control (MTI video)	4.2	2,400
Subtotal	13.5	6,300
Total	45.5	8,300

* Tracking feedback figures assume the target detection function is in the computer; hence about a one percent increase in the values would be required to transfer threshold data to the digitizer.

** Quantizer threshold control is via the tri-level, two-stage SCF concept.

(C) System Model Three. The beamsplitting function is allocated to the IOP in system model three, refer to figure 34 and table XV. With this model the digitizer would perform leading edge detection and transfer a small amount of quantized video to the computer. The computer would locate the target center and perhaps perform additional target report screening, such as run length discrimination. Other functions would be the same as model two.

This model has some interesting features; however the overall appeal is small. Because the computer could perform some target detection functions, a little flexibility is gained over model two. For example, by considering the digitizer leading edge detector as a pre-detector, the IOP could participate in the target detection function. Any of the final detector algorithms described earlier could be applied here. Nevertheless, the degree of flexibility is quite limited. Computer requirements, under the assumption that the digitizer performs target detection, are approximately 52 percent for processing time and 9,000 words of memory. Hence, the computer processing capability is not significantly exploited. System risk increases over model two, but no significant problems related to achieving a workable system would be anticipated.

(D) System Model Four. In this model, a hardware predetector replaces the digitizer leading edge detector, and quantized video is transferred to the computer wherein final detection is performed. This model applies the Automatic Clutter Eliminator (ACE) as the final detector. Other functions remain as given in system model three. Figure 35 and table XVI illustrate this model. Both processing time and memory requirements take a sizeable jump; however the total requirement is still well within the limits established earlier as goals. A high degree of flexibility is offered because most of the video processing functions are in the software. On the other hand, the system risk is greater than the earlier models because the software target detection approach does not have a predecessor in ARTS III. Nevertheless, the feasibility of the software approach has been demonstrated in a previously developed military system so that it is not without credentials.

(E) System Model Four-A. The concept of dual target detection on two videos is demonstrated in system model four-A. Both normal and moving target indicator (MTI) quantized video is transferred to the computer. The digitizer performs predetection on each of the videos. Similarly, the computer provides parallel target detection and beamsplitting for the two videos. The two sets of target reports resulting from this parallelism are associated in the Radar/Beacon Correlation process. Figure 36 and table XVII further describe this model.

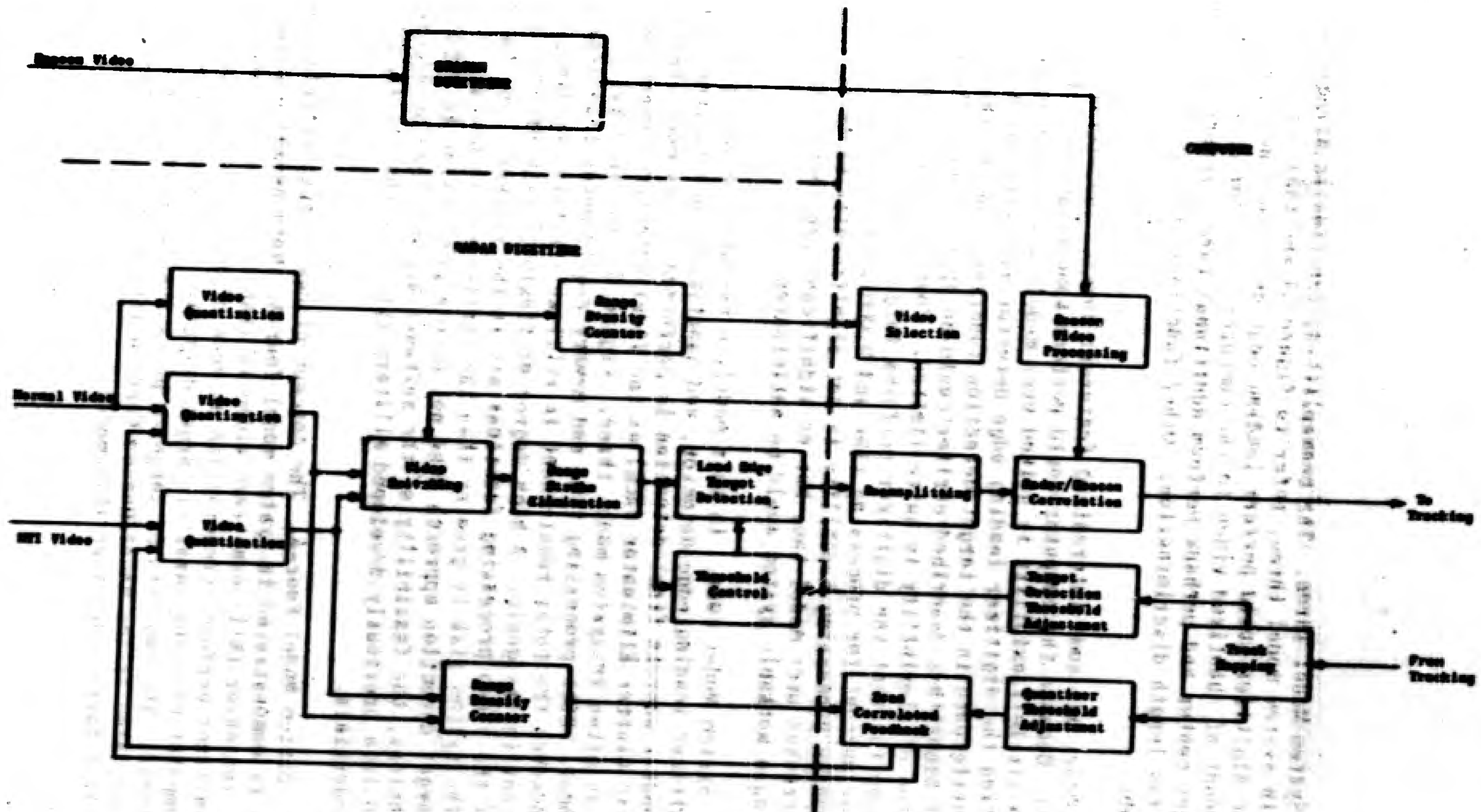


Figure 34. Block Diagram of System Model Three

Table XV. System Model Three

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Switching		
Video Quantization		
Range Strobe Elimination		
Target Detection and Threshold Control		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	3.5	200
Tracking Feedback*	4.4	300
Control of Video Selection and Quantization Thresholds**	13.5	6,300
Subtotal	45.5	8,300
Beamsplitting		
Input (small regions of quantized video)	1.6	200
Center of Density Algorithm	4.8	100
Subtotal	6.4	300
Total	51.9	8,600

*The figures for tracking feedback assume the target detection function is in the computer; hence about a one percent increase in the values would be required to transfer threshold data to the digitizer.

**The quantizer threshold control is via the tri-level two-stage SCF and is for two video types.

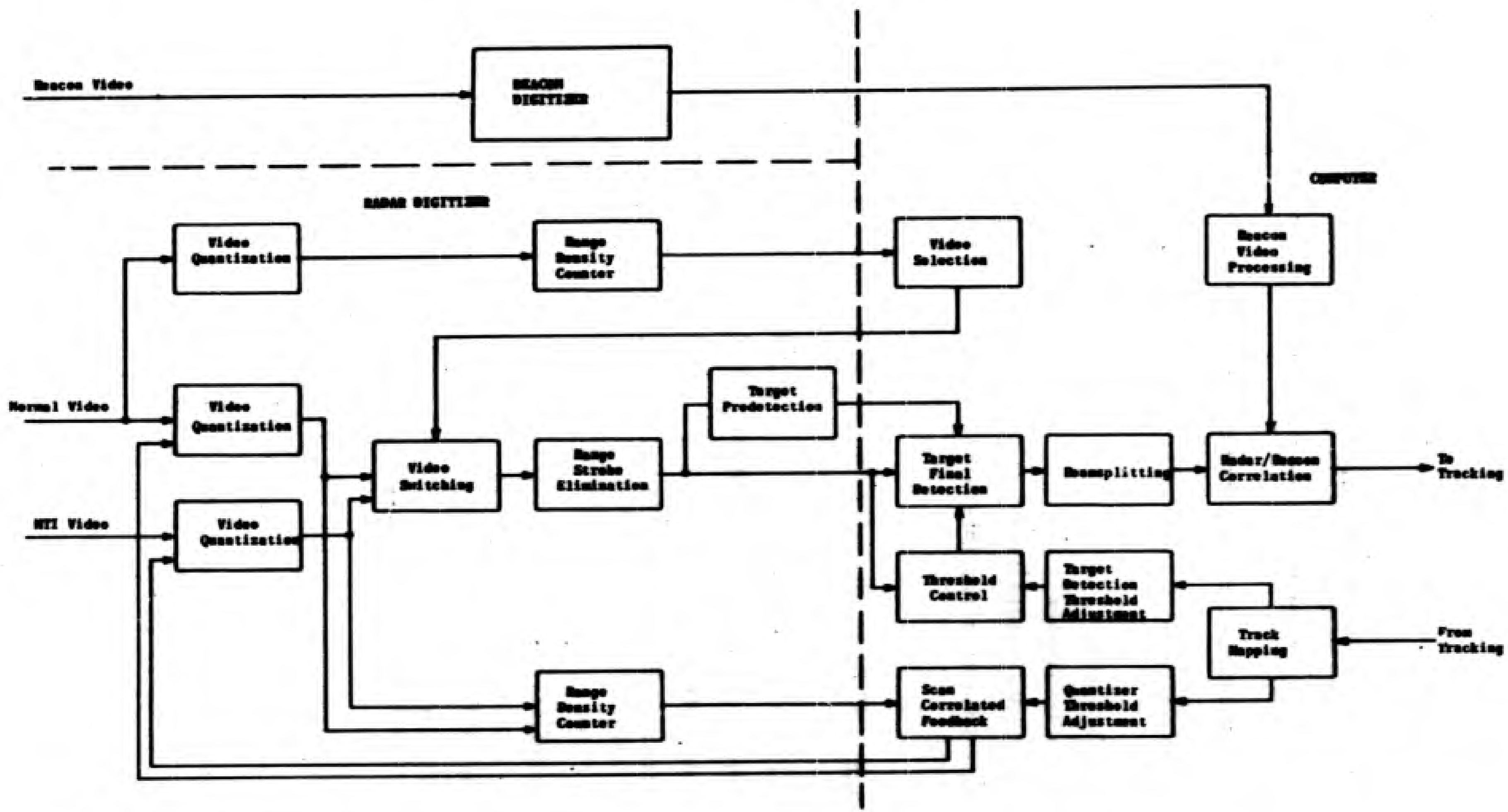


Figure 35. Block Diagram of System Model Four

Table XVI. System Model Four

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Switching		
Video Quantization		
Range Strobe Elimination		
Predetection		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	3.5	200
Tracking Feedback	4.4	300
Control of Video Selection and Quantizer Thresholds*	13.5	6,300
Input of Quantized Video	3.0	2,500
Target Detection via ACE**	17.8	1,200
Totals	66.3	12,000

*The quantizer threshold control is via the tri-level two-stage SCF and is for two video types.

**Figures for target detection are based on a sequential observer (T = 3) predetector operating on 7.5 percent noise.

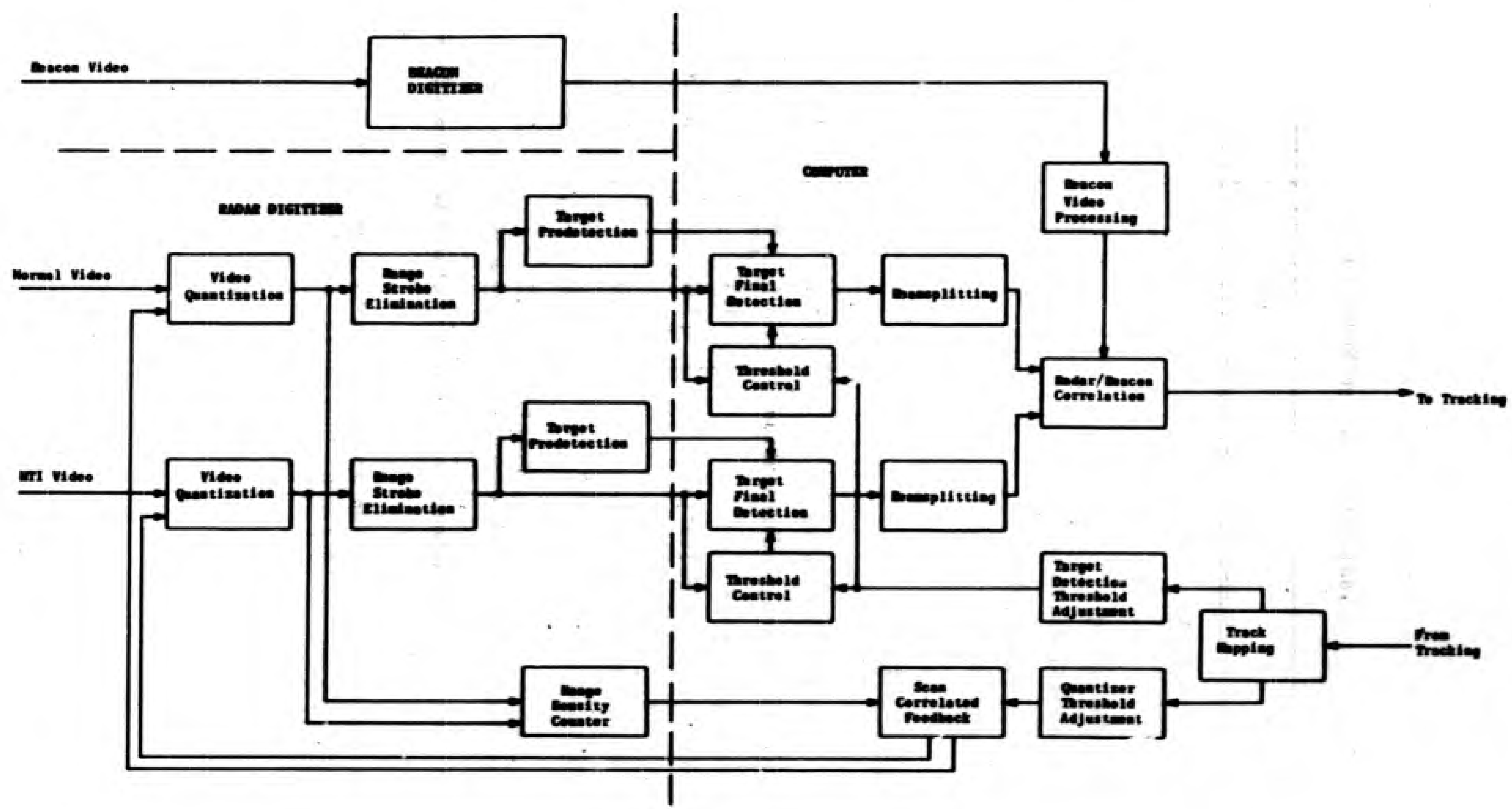


Figure 36. Block Diagram of System Model Four-A

Table XVII. System Model Four-A, Dual Target Detection of Two Videos

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Switching		
Parallel Quantization of Two Videos		
Range Strobe Elimination		
Predetection		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	4.7	200
Tracking Feedback	4.7	300
Control of Quantizer Thresholds*	10.6	5,200
Input of Two Quantized Video Types	6.0	5,000
Target Detection via ACE**	16.0	1,200
Beamsplitting	9.6	100
Totals	75.7	13,500

*Quantizer threshold control is via the tri-level two-stage SCF and is for two video types.

**Figures for target detection are based on the sequential observer (T = 3) predetector operating on five percent noise.

The trade off considerations of this model are similar to those of model four although from an experimental stand point this model has great appeal. The dual target detection could significantly enhance the radar tracking level performance. The risk for this system would be similar to that of model four, since that model could be considered as a fall back position. The computer requirements, which are about 76 percent processing time and 14K words, pose no problems. It is appropriate to note that the processing time quoted assumes operation at a five percent noise level in the quantized video.

(F) System Model Five. System model five illustrates the opposite extreme of system model one. As indicated in figure 37 and table XVIII most of the radar processing functions are allocated to the computer. This model differs from model four in the allocation of the predetector; whereas ACE is again assumed as the final detector.

The model is definitely a feasible approach; however it is subject to higher risk than other models already presented. As in system model four, a great deal of flexibility is offered since most of the processing functions are in the software. Various approaches to target detection and threshold control could easily be incorporated through program changes. This feature would be significant during the experimentation and evaluation of the system at a test site.

On the other hand, this approach is new to the ARTS III environment. As such, a degree of risk prevails. For example, if the computer became overloaded due to an unforeseen processing requirement, the fall back would only be through a change of system parameters, e.g. noise level. This could affect system performance. Because the Basic RDAS model may be used as an interim system for critical terminal areas, the risk of system model five is of concern.

(G) System Model Five-A. Dual target detection on two videos is again demonstrated by system model five-A, refer to figure 38 and table XIX. Except for the dual detection capability, this model is similar to system model five. It dramatically illustrates the processing power of the digital computer, but because of the heavy processing load, it would not be practical for implementation in a single IOP system. It would, however, be feasible in a multi-IOP system. As in system model four-A, it would be extremely interesting to evaluate the performance improvement during the experimental phase at a test site. On the other hand, this model would pose the same risk as model five and, as such, may not be desirable as a solution for an interim system at terminal areas.

(H) Summary of Functional Allocations. Figure 39 provides a breakdown of the functional allocations for the system models discussed in the preceding pages. the abbreviation RD refers to functions performed in the Radar Digitizer (hardware), whereas the IOP abbreviation refers to the functions performed in the computer (software).

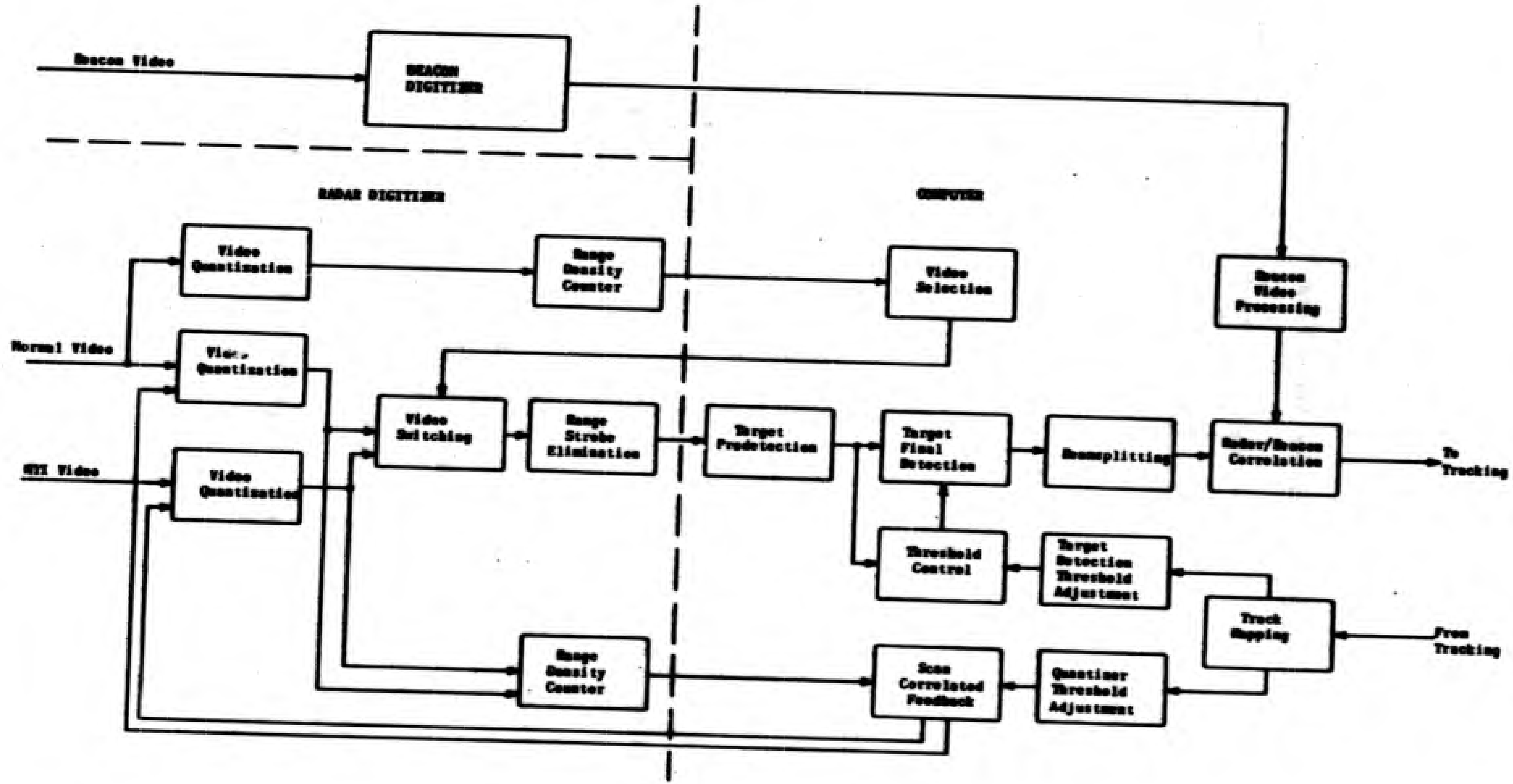


Figure 37. Block Diagram for System Model Five

Table XVIII. System Model Five

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Switching		
Video Quantization		
Range Strobe Elimination		
COMPUTER FUNCTIONS		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	3.5	200
Tracking Feedback	4.4	300
Control of Video Selection and Quantizer Thresholds*	13.5	6,300
Input of Quantized Video	3.0	2,500
Predetection and Target Detection via ACE**	24.3	1,300
Beamsplitting	4.8	100
Totals	77.6	12,200

*The quantizer threshold control is via the tri-level two-stage SCF and is for two video types.

**Figures for target detection are based on the sequential predetector operating on 7.5 percent noise.

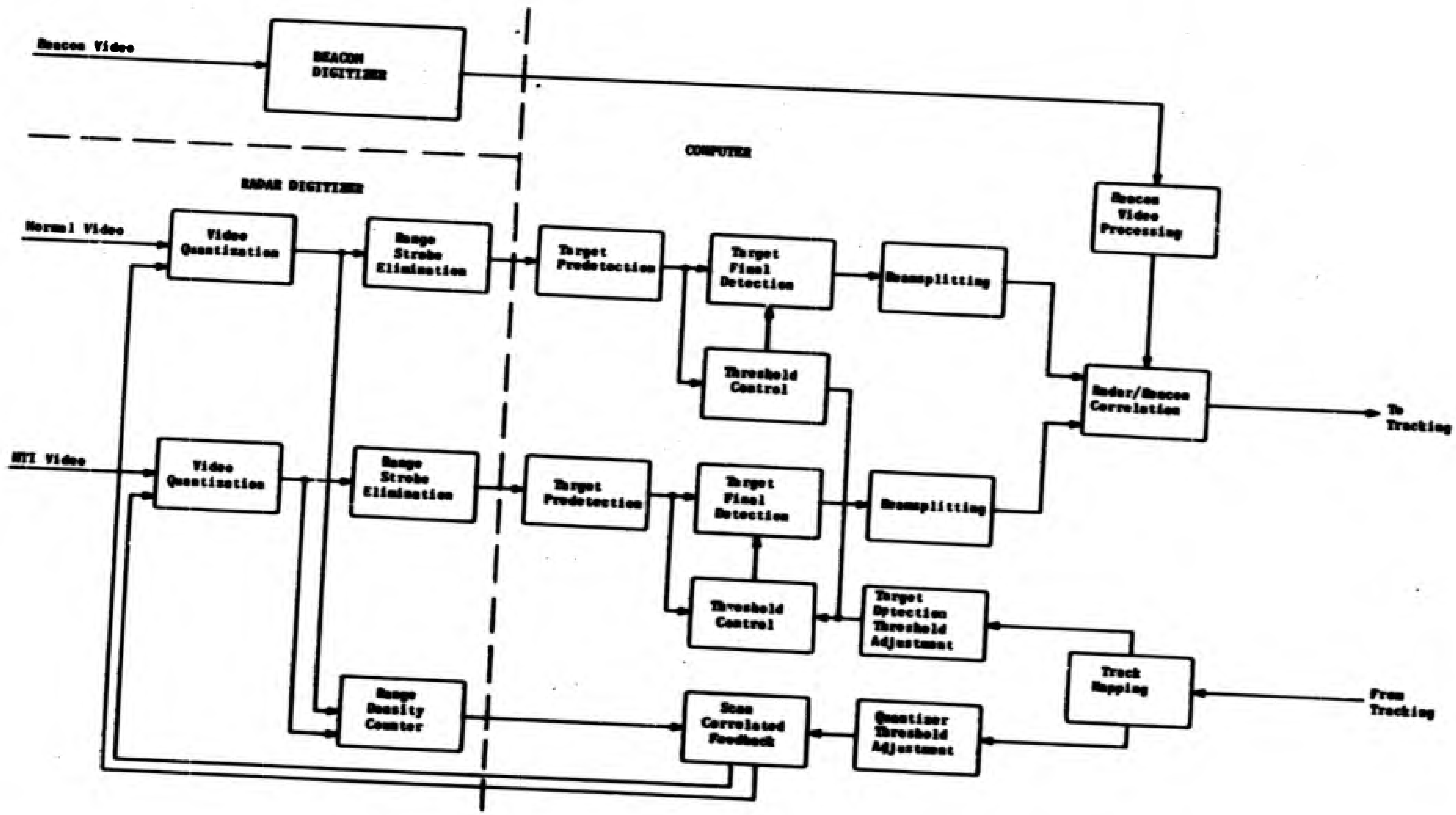


Figure 38. Block Diagram of System Model Five-A

Table XIX. System Model Five-A, Dual Target Detection on Two Videos

Functions	Time (%)	Memory (words)
<u>DIGITIZER FUNCTIONS</u>		
Video Switching		
Parallel Quantization of Two Videos		
Range Strobe Elimination		
<u>COMPUTER FUNCTIONS</u>		
Beacon Video Processing	24.1	1,500
Radar/Beacon Report Correlation	4.7	200
Tracking Feedback	4.7	300
Control of Quantizer Thresholds*	10.6	5,200
Input of Two Quantized Videos	5.9	5,000
Predetection and Target Detection via ACE**	37.2	1,300
Beamsplitting	9.6	100
Totals	97.8	13,600

*Quantizer threshold control is via the tri-level two-stage SCF and is for two video types.

**Figures for target detection are based on the sequential 4/8 predetector operating on five percent noise.

Functions	System Models 1	2	3	4	4A	5	5A
1. Video Selection	RD	IOP	IOP	IOP		IOP	
2. Video Switching	RD	RD	RD	RD		RD	
3. Quantization	RD	RD	RD	RD	RD	RD	RD
4. Parallel Quantization of Two Videos					RD		RD
5. Control of Quantization Thresholds	RD	IOP	IOP	IOP	IOP	IOP	IOP
6. Range Stobe Elimination	RD	RD	RD	RD	RD	RD	RD
7. Predetection				RD	RD	IOP	IOP
8. Target Detection & Threshold Control	RD	RD	RD	IOP	IOP	IOP	IOP
9. Beamsplitting	RD	RD	IOP	IOP	IOP	IOP	IOP
10. Beacon Video Processing	IOP	IOP	IOP	IOP	IOP	IOP	IOP
11. Radar/Beacon Correlation	IOP	IOP	IOP	IOP	IOP	IOP	IOP
12. Tracking Feedback	IOP	IOP	IOP	IOP	IOP	IOP	IOP
13. Input of Quantized Video In Area of Target				IOP	IOP		
14. Input of All Quantized Video						IOP	IOP

Figure 39. Summary of Functional Allocation for System Models

Recommendations

The objectives of the Basic Radar Data Acquisition Subsystem (RDAS) are two-fold. The first is to provide an experimental system in which different video processing concepts can be readily implemented and evaluated. Satisfaction of this objective dictates that a high degree of flexibility be possible in the Basic RDAS. This capability can best be achieved with a system which emphasizes the software approach. The second objective is to provide an interim operational system which can be installed and exercised at terminals in critical need of an automated system. This objective requires a design in which the risk of developing a workable system is minimal. The satisfaction of this objective could perhaps be best accomplished through a design which is an improved version of the Terminal Modified Radar Video Data Processor (TMRVDP). Although these objectives may appear to oppose one another, it is possible to design the Basic RDAS to satisfy both objectives.

The Basic RDAS approach recommended for consideration in the system design is to develop a system which incorporates features possessed by both system model one and system model five. That is, the digitizer should be capable of detecting and locating targets in addition to transferring quantized video to the computer. Similarly, the computer should be capable of handling digitizer target reports or developing target reports from the quantized video. Furthermore, the system should be designed such that either or both capabilities can be exercised. Because of the hardware target detection capability, the system risk would be minimal. Hence, the system would be suitable for operation at terminals. The provision for performing target detection within the computer would enhance the flexibility of the system during the experimental phase. In addition, the evaluation of dual target detection on two videos can be accomplished through operating the digitizer target detector on one video simultaneously with computer target detection on another video. Figure 40 is a block diagram of the system design recommended for consideration, and table XX lists the time and core requirements for the three possible configurations. Other design considerations are as follows:

1. Implementation of two tri-level, two-stage Scan Correlated Feedback (SCF) loops for quantizer threshold control of normal and moving target indicator (MTI) videos, respectively.
2. Application of the two-stage SCF concept for video selection.
3. Hardware implementation of a sliding range window for range strobe elimination.
4. Application of Automatic Clutter Eliminator (ACE) for target detection in the digitizer.
5. Application of the sequential 4/8 predetector followed by the ACE final detector for the software approach. (In addition the software should be designed such that the predetector and final detector can be easily changed).
6. Application of conventional beamsplitting in the hardware and center of density beam splitting in the software.
7. Incorporation of tracking feedback for threshold adjustment, at least in the software approach.

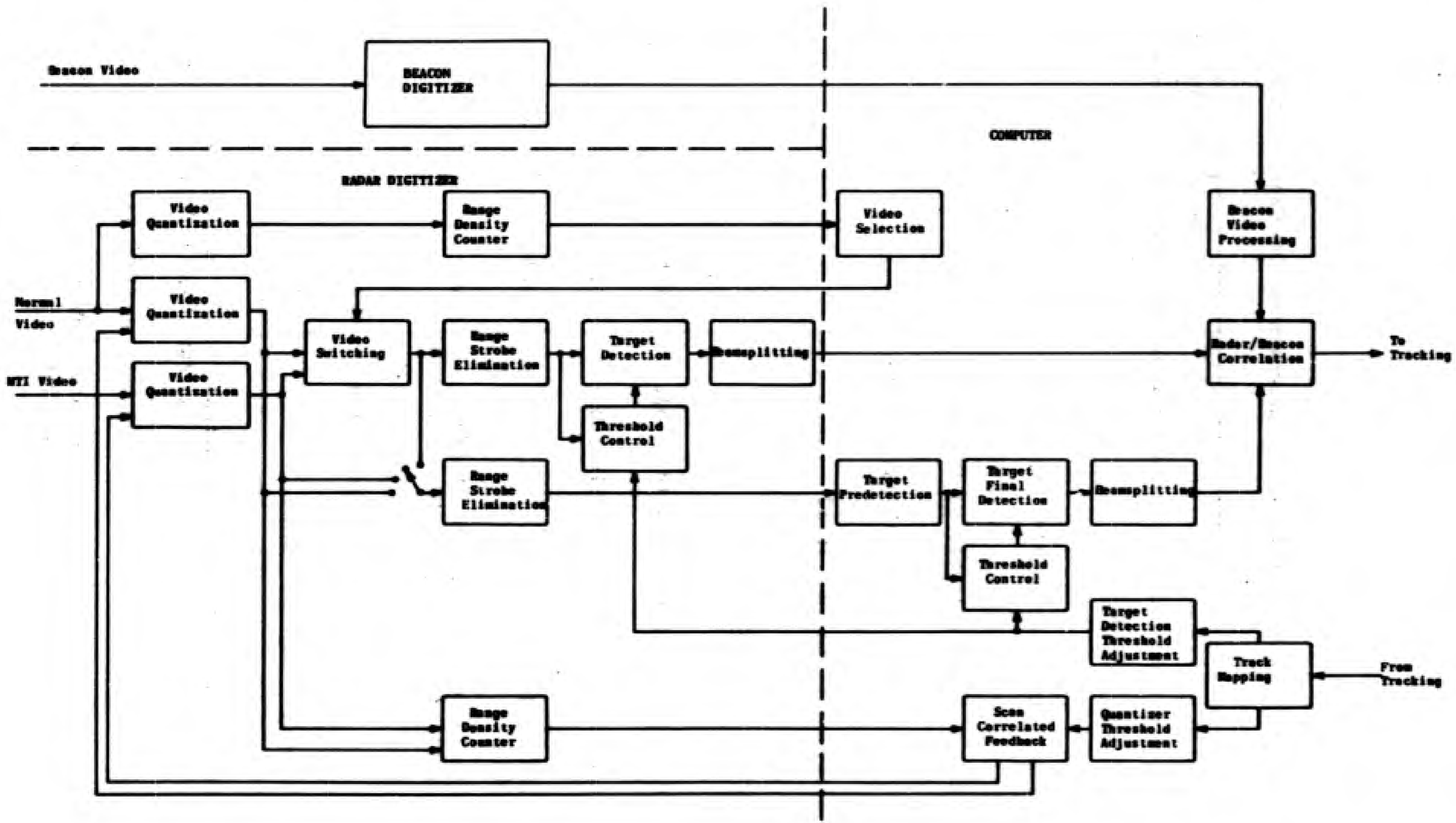


Figure 40. Recommended Basic RDAS System

Table XX. Basic RDAS Design Recommended for Consideration

Functions	Time (%)	Memory (words)
DIGITIZER FUNCTIONS		
Video Quantization(H)*, (S), (H/S)		
Video Switching(H), (S), (H/S)		
Range Strobe Elimination (H), (S), (H/S)		
Target Detection and Threshold Control (H), (H/S)		
Beamsplitting (H), (H/S)		
COMPUTER FUNCTIONS		
Beacon Video Processing (H), (S), (H/S)	24.1	1,500
Radar/Beacon Report Correlation (H), (S), (H/S)	4.7	200
Tracking Feedback (H), (S), (H/S)	4.4	300
Control of Video Selection and Quantizer Threshold** (H), (S), (H/S)	13.5	6,300
Input of Quantized Video (S), (H/S)	3.0	2,500
Predetection and Target Detection via ACE*** (S), (H/S)	24.3	1,300
Beamsplitting (S), (H/S)	4.8	100

*Three possible system configurations: (H) - hardware detection, (S) - software detection, (H/S) - hardware and software detection.

**Quantizer threshold control is via the two-stage tri-level SCF.

***Figures for target detection are based upon the sequential 4/8 predetector operating on 7.5 percent noise.

Bibliography

1. "ARTS III System Design Data," Volume I, FAA Contract DOT FA 69WA-2071, 1 November 1969.

2. Final Report, FAA Contract No. DOT FA 69WA-2071, Letter Amendment No. 3, "Expansion of the ARTS III System to the Radar Tracking Level ARTS III A," March, 1970.

3. Final Report, FAA Project No. 231-001-01X, Report No. NS-67-1, "Test and Evaluation of the Radar Video Data Processor," June, 1967.

4. Karlin, Samuel, A First Course in Stochastic Processes, New York: Academic Press Inc., 1966.

5. Lindgren, B. W., Statistical Theory, New York: The Macmillan Co., 1968.

6. Skolnik, Merrill I., Introduction to Radar Systems, New York: McGraw Hill Book Co. Inc., 1962.

7. "ARTS III Coding Specifications," Volume I, FAA Contract DOT FA 69WA-2071, 18 August 1970.

BLANK PAGE

APPENDIX A
ANALYSIS OF THE SCAN CORRELATED FEEDBACK CONCEPT

The Scan Correlation Feedback (SCF) Concept is a control system for setting the clipping level threshold in the video quantizer. By operating on binary quantized video produced during one scan, the SCF determines what threshold should be applied on the next radar revolution. The objective is to manipulate the threshold in such a manner that the percentage of ones produced from noise and clutter video is regulated to a predetermined value.

The purpose of this analysis is to compare the functional performance for three methods of the SCF concept. The SCF as implemented in the Terminal Modified Radar Data Video Processor (TMRVDP) is taken as the reference system. Modified versions which have as a purpose more efficient computer implementation are compared against this reference system. A measure of comparison is the statistical variance of the noise level produced by the quantization process.

The analysis is an application of the theory of Markov Chains. The three SCF systems are first modeled as finite Markov Chains, and then the models are evaluated under steady-state conditions for the purpose of comparing long-term performance. Following that is an investigation of short-term transient response.

(A) Description of SCF Models. Before proceeding with the analysis, the three SCF methods are described.

(1) TMRVDP System. The functions of the SCF systems are the quantization of analog video, evaluation of the percentage of binary ones produced and adjustment of the quantization threshold. A block diagram of these functions for the reference SCF system is illustrated in figure A-1. The functions are described in the following paragraphs.

The quantizer compares radar analog video V(t) against a voltage threshold and produces binary video corresponding to range intervals--called range cells. The SCF of the TMRDVP utilizes range intervals of 1/8 of a mile; however in this analysis range cells corresponding to 1/16 of a mile are treated.

A zone counter determines the number of ones produced in each zone having a dimension of 2 miles by 32 ACPs. The partitioning of the whole radar surveillance area into zones permits regional control of the video quantization. The selection of 1/16 radar mile quantization implies that 1,024 range cells are contained in each zone.

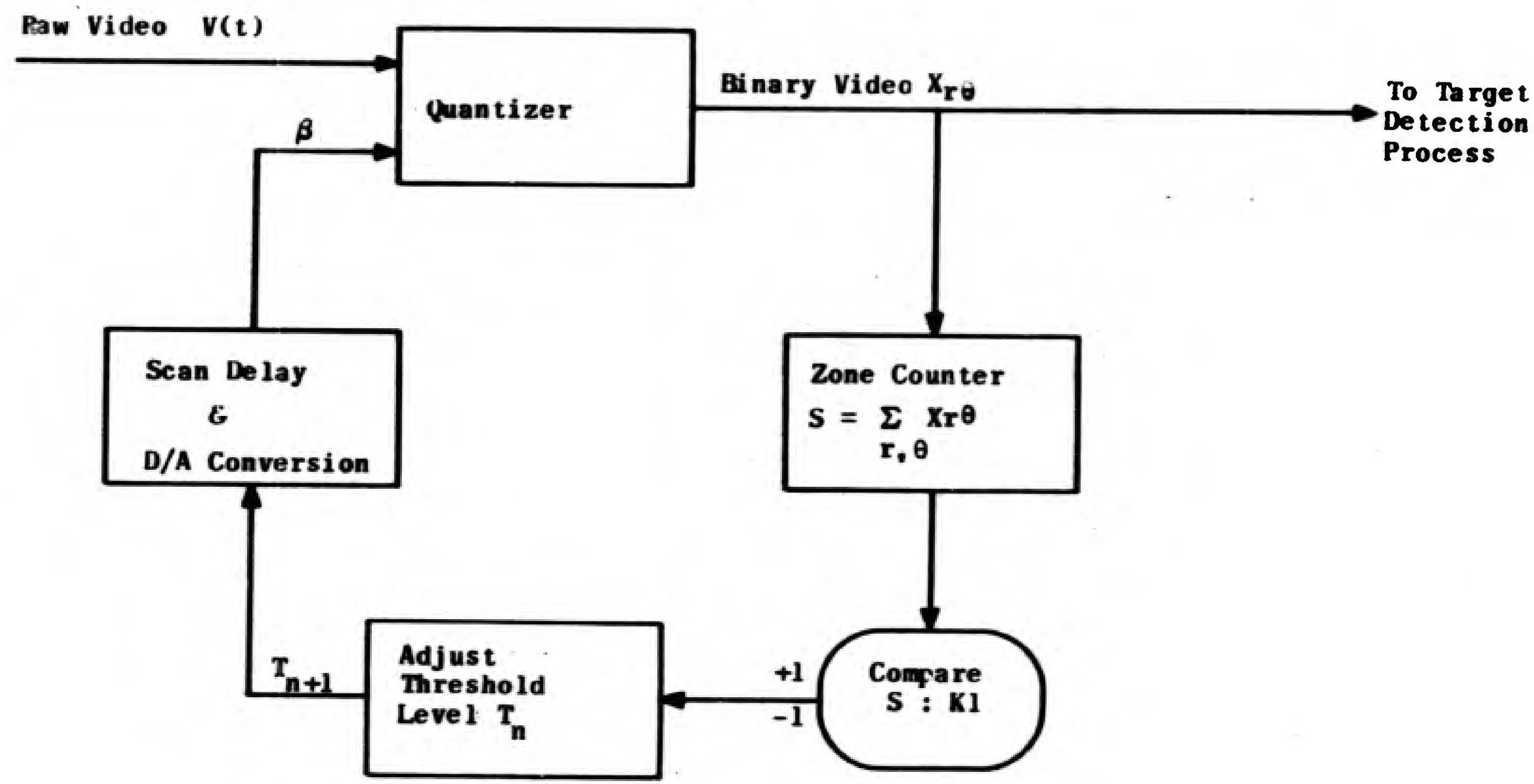


Figure A-1. Bi-level Single-state SCF (TM RVDP)

In the next operation, the zone count (S) is compared to a threshold (K1). This threshold is chosen on the basis of the noise level (percentage of ones) desired for the zone. A later paragraph describes how K1 is determined. The condition that $S > K1$ is interpreted as an indication that the voltage threshold β is too low and too many video ones are produced. The complementary condition indicates that β is too high and too few ones are produced.

Based on the above conditions, a threshold level (T_n) is either incremented or decremented, respectively, by one count. The resulting threshold level T_{n+1} determines the voltage threshold to be applied on the next radar scan of the zone. Sixty-four levels are available for T_n. The sixty-four levels are transformed into voltage levels β which cover the expected voltage range of the radar analog video. In this analysis, this range is assumed to be zero to four times the average level ($\bar{V}$) of the video. The transformation is linear, that is,

$$\beta = B(T_n) = T_n \cdot \frac{\bar{V}}{16}$$

In the TMRVDP, an estimate of the average video level taken as the output of an integrater having a fast time constant, on the order of five microseconds.

The SCF described above is that which is implemented in the TMRVDP an is called the bi-level single-stage SCF. The meaning of the term "bi-level" is that the outcome of the decision process has two possibilities, i.e. T_n is either incremented or decremented. Use of the modifier "Single-Stage" relates to the zone counting and decision processing. In particular, only a single decision is made for each zone on each scan. The following paragraph describes the second SCF system in which the decision process is modified.

(2) Bi-level Two-Stage SCF. The bi-level two-stage SCF incorporates a two-level decision process to determine whether to increment or decrement the threshold level T_n. This process replaces the zone count and compare operation of the single-stage SCF. The first decision level involves only the quantized video from a single sweep. The number of ones in a two mile range interval is counted and compared to a range density threshold (R1). For each two mile interval, a binary range density indicator (Y) is set to one if the range count (R) exceeds R1; otherwise Y is set to zero. This process is repeated for each of the sweeps in the zone (32 ACPs). The second decision process for each zone operates on the range density indicators within that zone. In particular, the number of Y's containing ones is determined, and this azimuth count (A) is checked against the threshold called the azimuth density threshold (A1). When the condition that $A > A1$, the threshold level T_n is incremented by one; alternately, T_n is decremented if $A \le A1$. Figure A-2 provides a block diagram of the bi-level two-stage SCF.

The assertion to be proven by this analysis is that performance of the two-stage SCF closely approximates that of the single-stage SCF. In addition, it will be shown that a further modification of the two-stage SCF provides even better performance. This modification is in the method of incrementing T_n.

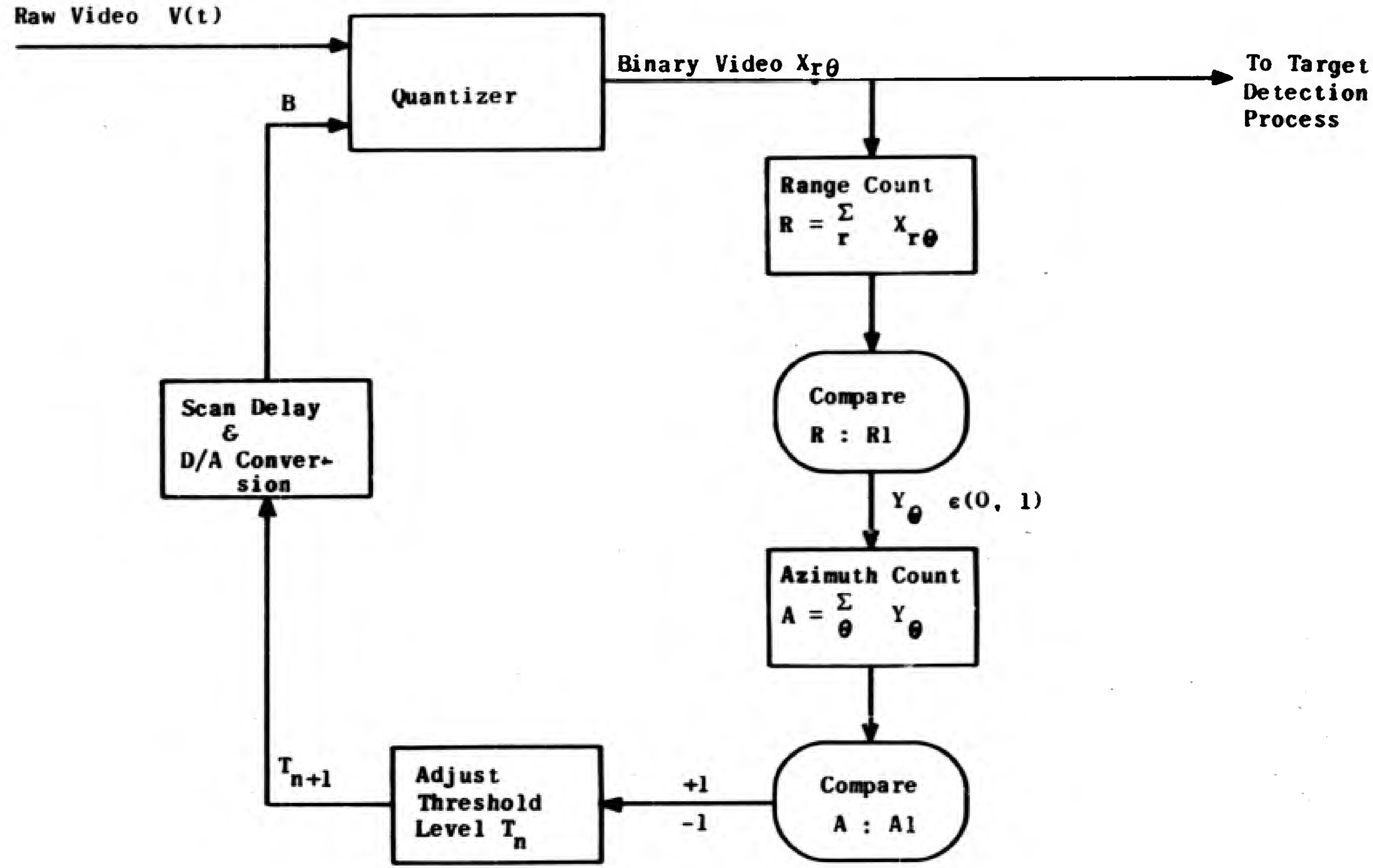


Figure A-2. Bi-level Two-Stage SCF

(3) Tri-level Two-Stage SCF. The incrementation in the tri-level two-stage SCF differs from the bi-level incrementation discussed above in that three possibilities exist for the adjustment of T_n. Two azimuth density thresholds (A1 and A2) are required. If the azimuth count (A) of the range density indicators with in a zone exceeds A2, the threshold level T_n is incremented. If $A < A1$, then T_n is decremented; otherwise T_n is left unchanged. Hence, whenever the count (A) takes on values within an interval of acceptance, the threshold level is assumed to be properly set, and no change is permitted. This differs from the bi-level processes in which T_n is changed every scan.

(B) Math Models. The SCF concepts can be modeled as a Markov Chain. Specifically, the process can be considered as a Random Walk with reflecting barriers. The 64 threshold levels are defined as states of the Markov Chain. The objective is then to determine the properties of the stochastic random variable T_n. This is achieved through determining the probability distribution for T_n. It should be noticed that in each zone, the SCF system operates independent of the neighboring zones. Hence in the analysis, it is sufficient to consider the characteristics of only one zone. The following paragraphs define the three versions of the SCF concepts in mathematical terms.

(1) Bi-level Single-Stage SCF. Let T_n be a stochastic random variable defining the states of the bi-level single-stage SCF at epoch n, that is, on the n-th scan. The probability distribution of T_n defines the properties regarding the selection of the threshold level. In particular, the distribution describes the likelihood that a threshold level takes on a certain value at scan n. This probability distribution can be determined if the initial distribution for the threshold levels T_0 and the transition probability matrix (M) of the Markov Chain are known. Through most of the discussion, the initial distribution is unimportant; therefore it will be left undefined until it is needed. The transition matrix M can be established by examining the processes of the SCF concept.

The quantization process is considered first. The analog video entering the quantizer is assumed to be a stationary stochastic process V(t) defined by the Rayleigh Distribution. That is,

$$P(V(t) \leq \beta) = \int_0^{\beta} \frac{v e^{\frac{v^2}{-2\psi^2}}}{\psi^2} dv$$

where β is the voltage threshold and ψ is a parameter directly proportional to the average voltage level of the analog video. It is further assumed that the autocorrelation function of this process approaches zero with sufficient speed that the video in adjacent range cells can be treated as statistically independent. With these assumptions, the probability distribution of the binary video produced by the quantizer can be defined.

Let $X_{r\theta}$ be a Bernoulli random variable representing the binary video in a range cell of the zone. This cell is interpreted as the one corresponding to the r-th range interval of the sweep θ. Because it is assumed V(t) is stationary every $X_{r\theta}$ has the same distribution, i.e. a Bernoulli distribution defined by parameter (p_n) where,

$$p_n = P(X_{r\theta} = 1) = P(V(t) > \beta) = \int_{\beta}^{\infty} \frac{v}{\psi^2} e^{\frac{v^2}{-2\psi^2}} dv = e^{\frac{\beta^2}{-2\psi^2}}$$

Note the p_n is a function of the threshold level (T_n) utilized on the n-th scan, since $\beta = B(T_n)$. This parameter P_n is called the noise probability of epoch n. A graph of p_n for $\psi = ½$ volt is provided in figure A-3.

At the n-th scan, the zone count S_n can now be described as a random variable with a binomial distribution. The parameters of this distribution are the noise probability p_n and N where N is the number of range cells in a zone. In this analysis N is taken as 1,024.

The comparison of S_n against threshold K1 can be constructed as a statistical hypotheses test; however a precise formulation is not necessary. Consider that a noise probability p, called the Command p_n, is desired out of the quantizer. A sample of video whose underlying noise probability is p_n is taken from a zone. The decision is whether $p_n > p$ or $p_n \leq p$. Because the zone count S_n has a binomial distribution, the Neyman-Pearson Test of the hypothesis ($p_n > p$) has an acceptance region of the form:

$$S_n > K1$$

where K1 is a constant which has yet to be determined. Consider taking K1 as the mean (expectation) of the binomial distribution having parameters N, p, i.e.

$$K1 = Np.$$

This choice of K1 produces a test whose size is on the order of one half, i.e. the probability of Type I error is close to one half. Therefore, when $p_n = p$, the probability of incrementing T_n is approximately the same as the probability of decrementing T_n. This is clearly a desirable characteristic for bi-level incrementation.

With this ground work, the transition probability matrix M for the Markov process can be constructed. Because of the assumed stationarity of the analog video, the Markov process is stationary. Therefore, the transition matrix M is independent of the time (n). The transition matrix corresponding to the bi-level single-stage SCF with 64 threshold levels is as follows:

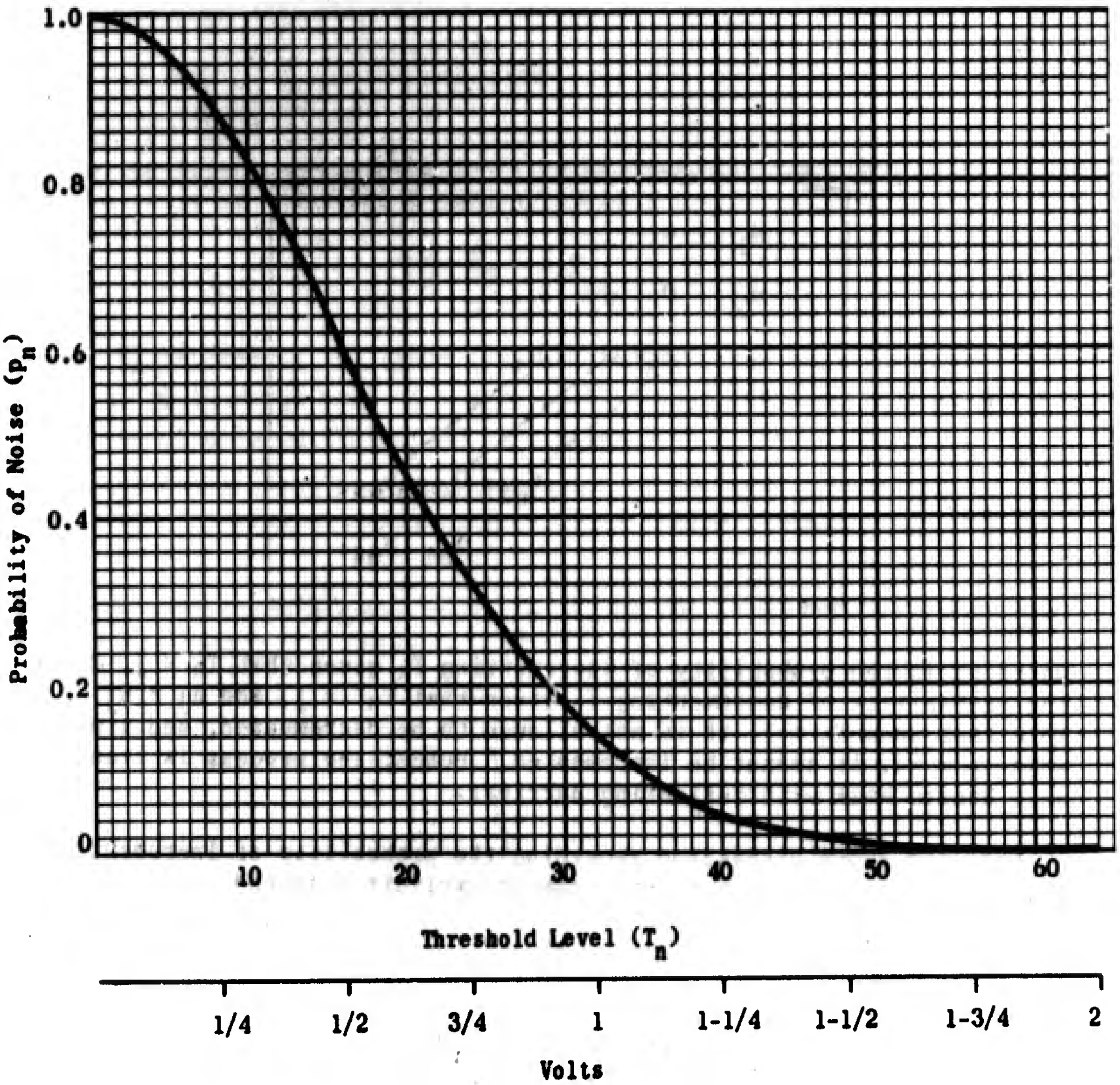


Figure A-3. Probability of Noise Vs. Threshold Levels (ψ = 0.5 Volts)

$$M = \begin{bmatrix} d_1 & u_1 & & & & & \\ d_2 & 0 & u_2 & & & & \\ & d_3 & 0 & u_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & d_{63} & 0 & u_{63} \\ & & & & d_{64} & u_{64} \end{bmatrix}$$

where u_j is the probability of incrementing T_n given that $T_n = j$, d_j is the probability of decrementing T_n given that $T_n = j$, and $u_j + d_j = 1$. Note that when $T_n = 1$, it is not allowed to be decremented, and similarly when $T_n = 64$, it cannot be incremented. Hence, the process is treated as a Random Walk with reflecting barriers.

From the above discussion regarding the probability of incrementing T_n, the expression for u_j and d_j can be explicitly stated.

$$u_j = P(S_n > Kl \mid T_n = j)$$

$$= \sum_{k>Kl} \binom{N}{k} p_n{}^k (1 - p_n)^{N - k}$$

$$d_j = 1 - u_j$$

It should be noted that the dependence of u_j on T_n is through p_n. Recall that

$$p_n = e^{\frac{\beta(T_n)^2}{-2\sigma^2}}$$

The Markov Chain formulation of the bi-level single-stage SCF is now complete. The following paragraphs describe the modifications to this Markov Chain resulting from the bi-level and tri-level two-stage SCF.

(2) Two Stage SCF Markov Chain. The only change in the Markov process for the bi-level two-stage SCF is in the construction of the transition probabilities, u_j and d_j. These probabilities are established by investigating the two-stage decision process.

In the first level of the decision process, the quantized video is counted on the basis of independent sweeps, i.e.,

$$R_\theta = \sum_r X_{r\theta}$$

where R_θ represents the range count for sweep θ and $X_{r\theta}$ is the quantized video for the $r\theta$-th range cell of a zone. This range count R_θ is a random variable from the binomial distribution with parameters p_n and N_R where N_R is the number of range cells along the range dimension of the zone. Based on the zone size and range intervals already stated, it is evident that $N_R = 32$ for this analysis.

With respect to a given zone, let Y_θ be the range density indicator for sweep θ. Then

$$Y_\theta = \begin{cases} 0 & , \text{ if } R_\theta \le R_1 \\ 1 & , \text{ if } R_\theta > R1 \end{cases}$$

where R1 is the range density threshold whose value is to be specified later. Now, Y_θ is a Bernoulli random variable with

$$P(Y_\theta = 1) = P(R_\theta > R1)$$

$$= \sum_{r > R1} \binom{NR}{r} p_n{}^r (1 - p_n)^{N_R - r}$$

Because the $X_{r\theta}$ have been assumed to be independent and identically distributed, it is evident that the Y_θ's for zone are also independent and identically distributed. For convenience, let $\emptyset = P(Y_\theta = 1)$, then it can be seen that the azimuth count for scan n, i.e.

$$A_n = \sum_\theta Y_\theta$$

is a binomial random variable with parameters $\emptyset$ and N_θ (the number of sweeps in a zone, $N_\theta = 32$).

The decision to increment T_n is based on the criteria

$$A_n > A1$$

where A1 is a constant which determines the size of the hypothesis test. Therefore, the two-stage SCF has, as could be expected, two critical parameters (A1 and R1) which have to be specified. Just as in the bi-level single-stage process, it is desired to have the probability of incrementing T_n approximately equal the probability of decrementing T_n when $p_n = p$. There exists a number of A1 and R1 combinations which would satisfy this objective. Perhaps the most evident choice is to take the means (expectations) for R_θ and A_n. That is, let

$$R1 = E(R_\theta) = N_R p$$

and

$$A1 = E(A_n) = N_\theta \phi'$$

where p is the Command p_n and

$$\phi' = \sum_{r>R1} \binom{N_R}{r} p^r (1 - p)^{N_R - r}$$

The transition probabilities u_j and d_j can now be defined. Let

$$\phi_j = P\,(Y_\theta = 1 \mid T_n = j)$$

$$= P\,(R_\theta > R1)$$

$$= \sum_{r>R1} \binom{N_R}{r} p_n^{\ r} (1 - p_n)^{N_R - r}$$

then

$$u_j = P\,(A_n > A1 \mid T_n = j)$$

$$= \sum_{k>A1} \binom{N_\theta}{k} \phi_j^{\ k} (1 - \phi_j)^{N_\theta - k}$$

and

$$d_j = 1 - u_j$$

where R1 and A1 are taken as defined in the previous paragraph.

(3) Tri-Level Two-Stage Markov Chain. The tri-level Two-Stage SCF is formulated in much the same manner as the bi-level two-stage SCF, except that an additional transition probability is introduced. The tri-level process permits T_n to remain unchanged if the sampled video is within a certain confidence interval, i.e.

$$T_{n+1} = T_n \quad \text{if } A1 \le S_n \le A2$$

The transition probability matrix for this process has the following form

$$M = \begin{bmatrix} s_1 & u_1 & & & & \\ d_2 & s_2 & u_2 & & & \\ & d_3 & s_3 & u_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & d_{63} & s_{63} & u_{63} \\ & & & & d_{64} & s_{64} \end{bmatrix}$$

where d_j is the probability of decrementing T_n given that $T_n = j$, s_j is the probability of leaving T_n unchanged given that $T_n = j$, and u_j is the probability of incrementing T_n given that $T_n = j$. With the definition that $d_1 = 0$ and $u_{64} = 0$ then the following expression must be satisfied for all j's.

$$d_j + s_j + u_j = 1$$

Notice again that this process is a Random Walk with reflecting barriers.

There now exists three parameters, R1, A1, A2, which determine the process. Consider choosing R1 as before, i.e.

$$R1 = E(R_\theta) = N_R p$$

Then the definition of Y_θ as a Bernoulli random variable remains unchanged. Further consider that if $A_n = \Sigma_\theta Y_\theta$ is within a one sigma confidence interval about the expectation, $E(A_n)$, then the present value of T_n is taken as adequate and T_n does not change. The form of A1 and A2 is then as follows

$$A1 = N_\theta \emptyset - \sqrt{N_\theta \emptyset (1-\emptyset)}$$

$$A2 = N_\theta \emptyset + \sqrt{N_\theta \emptyset (1-\emptyset)}$$

$$\emptyset = \sum_{r>R1} \binom{N_R}{r} p^r (1-p)^{N_R - r}$$

and p is the command p_n.

The expressions for determining the transition probabilities can now be stated, i.e.

$$u_j = P(A_n > A2 \mid T_n = j)$$

$$= \sum_{k>A2} \binom{N_\theta}{k} \emptyset_j^{\,k} (1-\emptyset_j)^{N_\theta - k}$$

$$d_j = P(A_n < A1 \mid T_n = j)$$

$$= \sum_{k<A1} \binom{N_\theta}{k} \emptyset_j^{\,k} (1-\emptyset_j)^{N_\theta - k}$$

$$s_j = 1 - u_j - d_j$$

where $\emptyset_j$ is the conditional probability for $Y_\theta = 1$ given $T_n = j$, i.e.

$$\emptyset_j = P(Y_\theta = 1 \mid T_n = j)$$

This concludes the formulation of the SCF concept as a Markov Chain. It has been seen that each SCF version can be defined as a Random Walk with reflecting barriers. Further the transition probability matrices are determined by the assumptions on the input noise process for the analog video and the choice of parameters for the decision process. The next paragraph discusses how information regarding the long-term operating characteristics of T_n can be established from the knowledge of the transition matrices.

(C) Stationary Distribution. It is of interest to investigate what happens after a SCF system has run for a long period of time. For example, does the system tend to settle on values of T_n which produce the desired probability of noise? If the above occurs, what is the expected noise level produced and how much variation in noise level would be typical?

The answers to the above questions are found in the theory of Markov Chains. Specifically because the described SCF systems have a finite number of states (none of which is absorbing), it is known that a steady-state condition does occur. Further, this steady-state can be described by what is called the stationary probability distribution of T_n. Let Z be a row vector made up of non-negative elements whose sum is one, i.e.,

$$Z = (z_1\ z_2\ \ldots.\ z_{64})$$

where $z_j \geq 0$ and $\Sigma z_j = 1$. There exists a unique vector C with the above properties that satisfies the following matrix equation.

$$C = (c_1\ c_2\ \ldots.\ c_{64}) = CM$$

where M is the transition probability matrix of the Markov chain. The elements of this C define the stationary probability distribution of $T = \lim_{n \to \infty} T_n$. That is, consider the system is turned on and run for a long period of time (theoretically an infinite period). An observer examines the system for the threshold level then in effect. The element c_j is the probability that the threshold level (T) is equal to j.

When stationary probability distribution of T is known, the expectation, E(T), and the variance, $\sigma^2(T)$, can be determined. Clearly both of these are defined, and both exist because the distribution is discrete and has a finite number of states. In particular, the expectation is

$$E(T) = \sum_j jc_j$$

and the variance is

$$\sigma^2(T) = \sum_j (j - E(T))^2 c_j$$

Clearly E(T) is the average value about which the threshold level fluctuates. Additionally $\sigma(T)$, the standard deviation of the stationary distribution, is a measure of how much the threshold levels vary around E(T).

The stationary distribution of T completely defines the steady-state operation of the system; however a greater appreciation of the system performance can be obtained by evaluating the process in terms of noise probabilities. That is, the expected value and the standard deviation of the noise probabilities produced by the quantizer is of greater interest than the corresponding variables in terms of threshold levels. Fortunately, this information is also readily available. It was stated earlier that quantizer output in terms of noise probability p_n is a function of threshold level T_n i.e.,

$$p_n = e^{-\frac{B(T_n)^2}{2\psi^2}}$$

Hence, the noise probability produced is a random variable η which can be construed as a single valued transformation of random variable T. Since T is a discrete random variable, η is also discrete, so that

$$E(\eta) = \sum_j \eta_j c_j$$

and

$$\sigma 2(\eta) = \sum_j (\eta_j - E(\eta))^2 c_j$$

where η_j is the noise probability associated with the threshold level j. Note that the parameter ψ is treated as an implied, fixed parameter of the transformation. The comparison of SCF concepts is primarily made in terms of these latter variables.

(D) Application. The comparison of the three SCF versions can now be made. Two main points are brought out. First, the performance of the bi-level two-stage SCF closely approximates that of the bi-level single-stage SCF. Second, application of the tri-level two-stage SCF can provide a significant improvement over that of the bi-level approaches. As mentioned in the math model discussion, the comparison is presented in terms of the expectation and standard deviation of the random noise probabilities η.

The data provided in the following discussions and illustrations was obtained through numerical evaluation of the math model via a digital computer. Evaluation is made for Command p_n values from 1.5 percent to 24 percent in steps of 1.5 per cent.

With respect to the comparison of single versus two-stage processes with bi-level incrementation, it is found that the expectations from the two approaches are in close agreement. The expectation of the noise probability, $E(\eta)$, for each method is tabulated in table A-I. Notice that the single-stage $E(\eta)$ is biased slightly higher than the Command p_n and that the bi-level two-stage $E(\eta)$ is slightly higher than that of the single-stage process. In both cases the magnitude of the bias is small, and is of no real consequence in operation. The significant element is that the expected noise level closely follows the Command p_n.

Table A-I. List of Expected η for Various Command P_n

Command P_n (%)	Expected η (%)		
	Bi-Level Single-Stage	Bi-Level Two-Stage	Tri-Level Two-Stage
1.5	1.549	1.829	1.623
3.0	3.118	3.185	3.012
4.5	4.587	4.941	4.495
6.0	6.054	6.361	6.142
7.5	7.622	7.839	7.531
9.0	9.080	9.113	9.046
10.5	10.645	10.967	10.637
12.0	12.110	12.392	12.108
13.5	13.577	13.721	13.354
15.0	15.142	15.294	14.982
16.5	16.607	17.105	16.736
18.0	18.067	18.224	17.889
19.5	19.629	19.998	19.594
21.0	21.099	21.178	21.111
22.5	22.557	22.894	22.475
24.0	24.122	24.147	24.073

Note: Zone Size 32 range cells x 32 ACPs, ψ = ½ volt

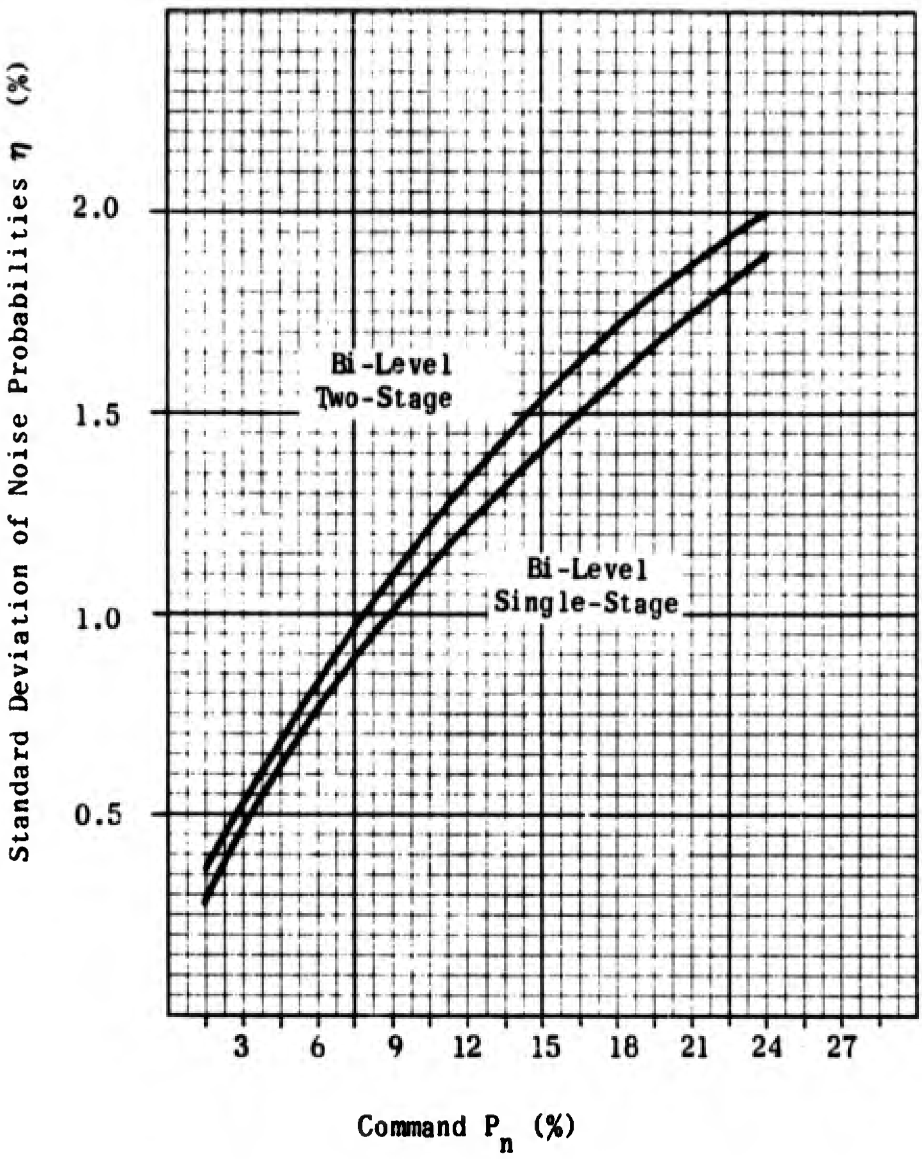
The next item of interest is the relationship of the standard deviation from the two bi-level processes. Plots of the standard deviation are shown in figure A-4. Here there is fairly close agreement in the values over the whole range of Command p_n. As might be expected the standard deviation for the single-stage process is uniformly smaller than that of the two-stage process, but the relative difference is small. The two-stage standard deviation is generally 10 percent higher than that for the single stage. This would not be a noticeable factor in actual operation of the system.

Further comparison of the bi-level process is via a direct look at the stationary probability distributions of η. Figure A-5 provides a side-by-side comparison for the several values of Command p_n. For convenience of presentation, the distributions are given in terms of the threshold level (T) random variable. The solid colored bars represent the single-stage distributions; whereas the open bars represent the two-stage distributions. Note that the general trend of the single-stage and the two-stage distributions are the same. Individual values (probabilities) within the distributions vary slightly, but not significantly.

In summary, the comparison of the single- and two-stage processes with bi-level incrementation has shown that nearly equivalent long term properties could be anticipated. For both processes the expectations, standard deviations and stationary distributions agree remarkably well. The next issue is whether the adaption of a tri-level incrementation process improves the overall performance. This is addressed in the following paragraphs. The tri-level two-stage SCF system is compared against the bi-level single-stage SCF system. Again, the approach is to investigate the long-term operating characteristics. The expectations and standard deviations of the noise probabilities produced by the quantizer are first reviewed, then the stationary probability distributions are discussed.

The tri-level two-stage $E(\eta)$ are more nearly equal to the bi-level single-stage $E(\eta)$ than was seen for the bi-level two-stage SCF system. In some cases, the tri-level two-stage $E(\eta)$ are closer to the Command p_n than the corresponding expectations for the bi-level single-stage process. This assertion can be verified by reviewing the second and fourth columns of table A-I. As illustrated by comparing the standard deviations of the subject processes, the tri-level two-stage SCF provides more precise control of the noise probabilities than does the bi-level single-stage process. The standard deviations of the noise probabilities are plotted in figure A-6. For example, at a Command p_n of 9 percent the bi-level single-stage process has a standard deviation of 1 percent; whereas for the tri-level two-stage process, the standard deviation drops to 0.8 percent. The reduction in standard deviation increases with higher command p_n values.

Review of the stationary probability distributions for the subject processes also shows that tighter control of the noise probabilities is possible with the tri-level two-stage SCF. The distributions which are shown in figure A-7 illustrate the manner in which the tri-level two-stage distributions are concentrated on a few values of the threshold levels. The peaks



Zone Size = 32 range cells X 32 ACPs

ψ = 1/2 volt

Figure A-4. Comparison of Standard Deviations of Noise Probabilities for Single and Two-Stage SCF Systems

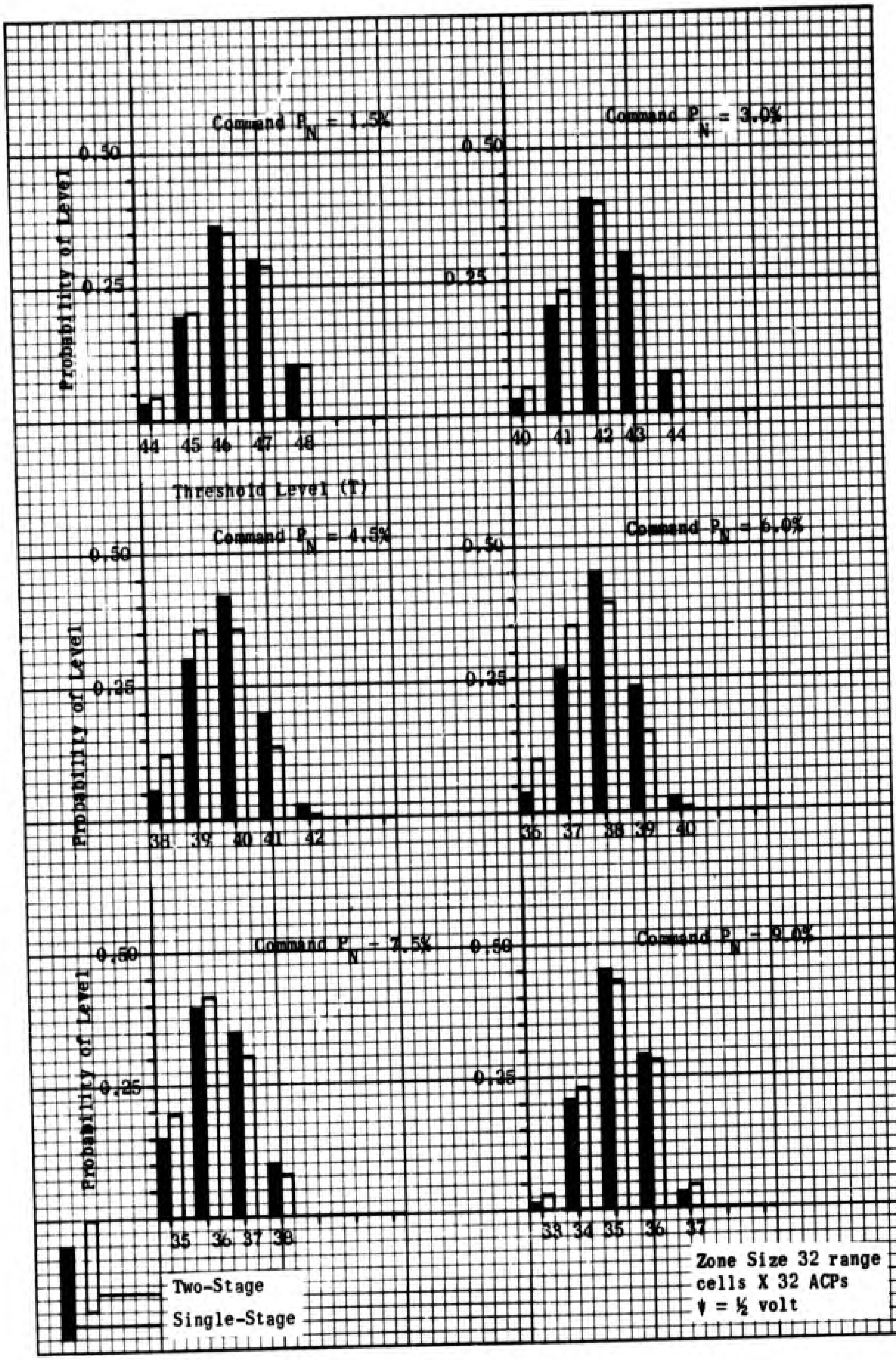
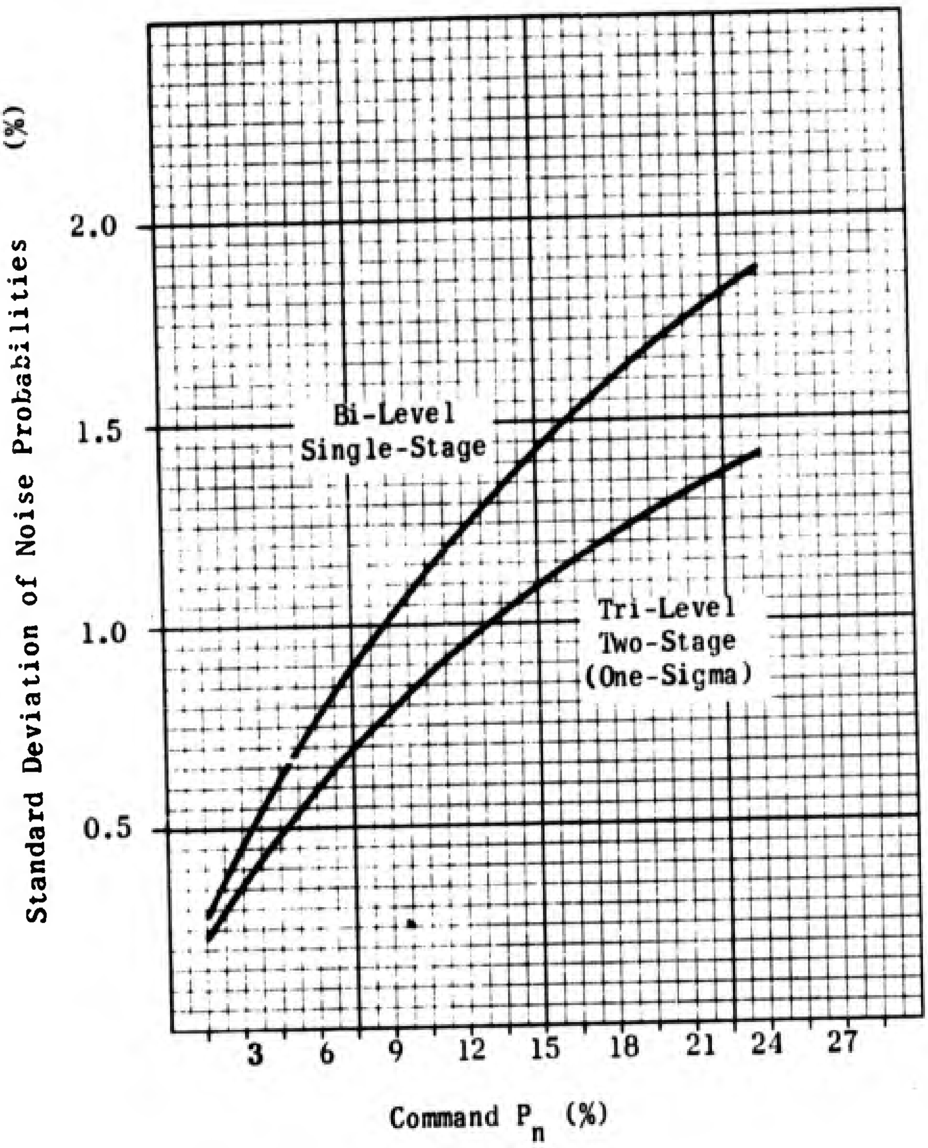


Figure A-5. Comparison of Stable Distributions (Bi-level)



Zone Size = 32 range cells X 32 ACPs

ψ = 1/2 volt

Figure A-6. Comparison of Standard Deviations of Noise Probabilities for Bi-Level Single Stage SCF and Tri-Level Two Stage SCF (One-Sigma)

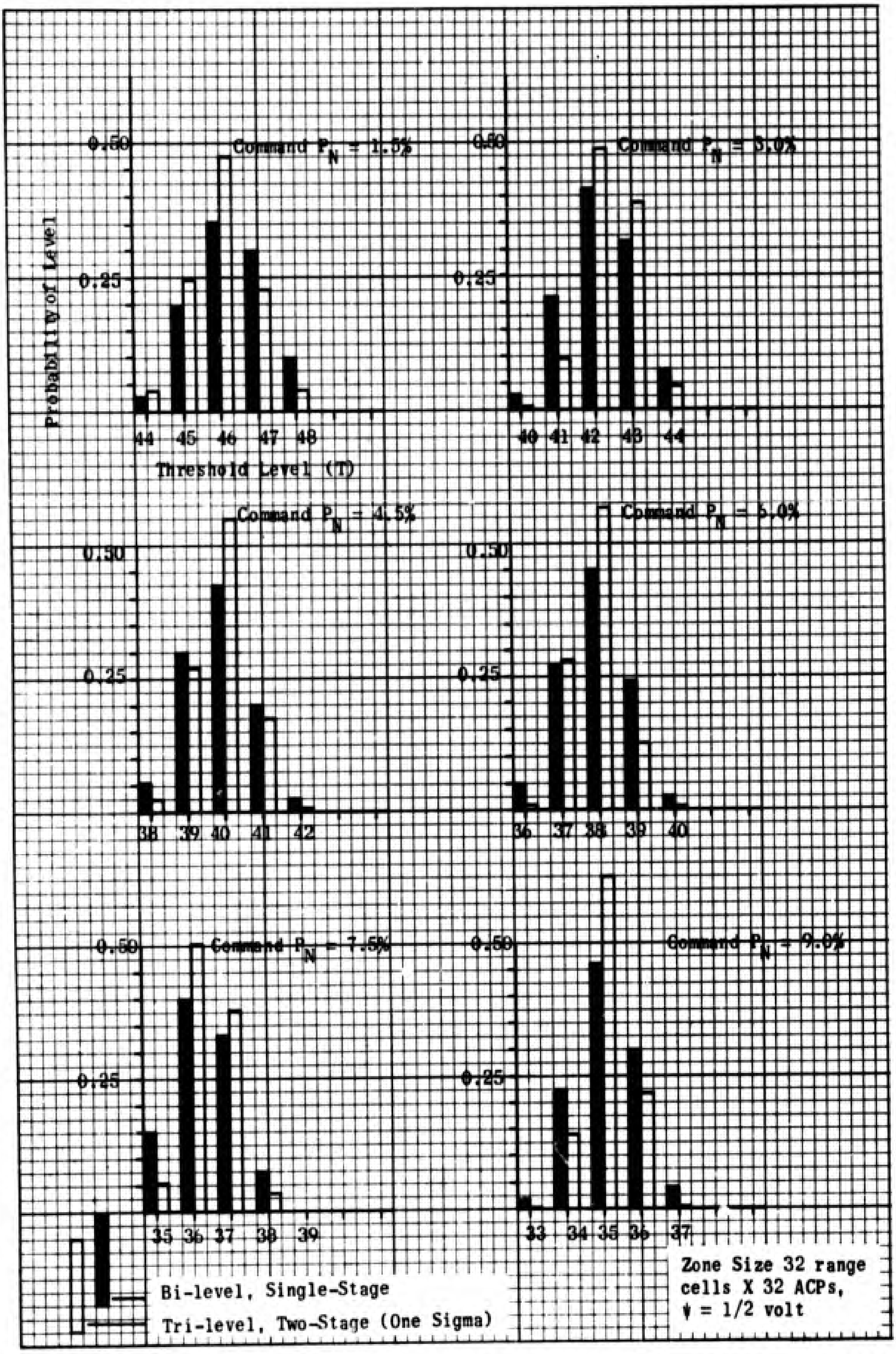


Figure A-7. Comparison of Stable Distributions Bi-Level (Single-Stage) vs. Tri-Level (Two-Stage) (One Sigma) of these distributions are much more pronounced than those of the bi-level single-stage process.

In conclusion, the tri-level two-stage SCF is a means of improving the control of the quantizer output. Nevertheless, a question arises as to the choice of confidence interval (A1, A2) used in the tri-level decision process. Recall that in the discussion of the tri-level process a one-sigma interval was designated. If a larger confidence interval had been chosen, even more dramatic improvement would have been attained in the steady-state condition. However, larger intervals would cause slower responses over the short-term. This issue is explored more in the following paragraphs.

(E) Transient Response. Just as in the investigation of steady-state conditions, the theory of Markov Chains provides information regarding the short-term characteristics of the system. Specifically, probability distributions can be determined at each step of the process, i.e., every scan. It is through the examination of these distributions that the short-term response of the SCF approaches are evaluated. In particular, the following situation is studied.

Assume that the SCF system has been operating long enough to reach steady-state. Further suppose the Command p_n is a specific value, say 4.5 percent. By previous analysis, the stationary probability distributions are known. Now let a new Command p_n, e.g., 10.5 percent, be selected.

The evaluation of the step-by-step probability distributions as they approach the new steady-state distribution is a means of determining the transient response. However, a sufficient characterization can be attained by simply tracing the means (expectations) of these distributions. The calculation of the n-th step distributions is as follows:

$$C_n = C_{n-1} P_2$$

where C_n is a row vector whose elements give the probability of the system being in a certain state at epoch n, and P_2 is the transition probability matrix for the new Command p_n. The vector C_0 is defined as the stationary probability distribution in effect before the switching of Command p_n.

Naturally, from C_n the expected threshold levels, $E(T_n)$ can be readily computed. Figure A-8 illustrates such expectations for 17 scans after the switch from a Command p_n of 4.5 percent to 10.5 percent. The SCF systems represented are the following:

- Bi-level two-stage SCF.
- Tri-level two-stage SCF (1σ confidence interval).
- Tri-level two-stage SCF (2σ confidence interval).

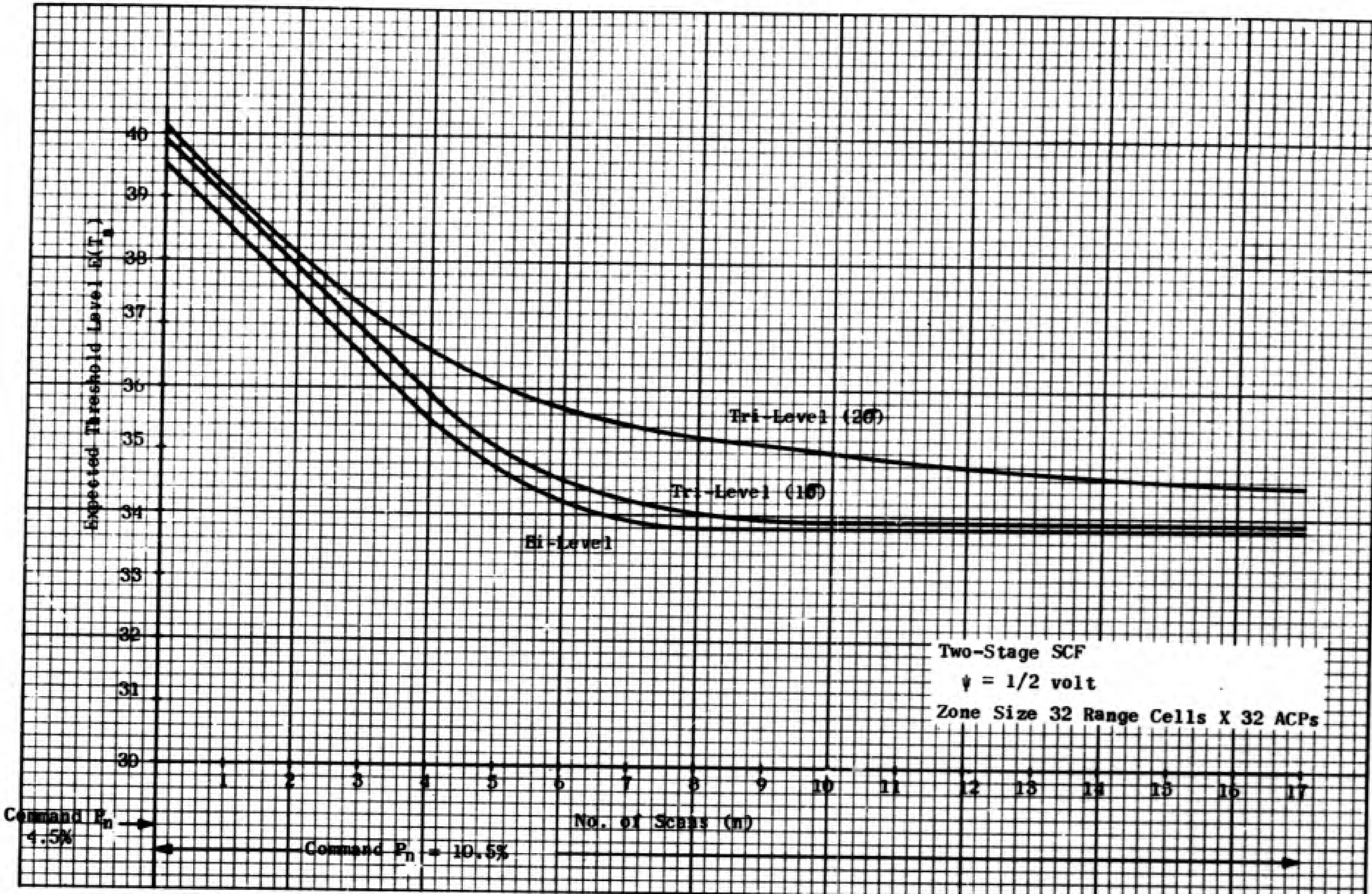


Figure A-8. Transient Response - Switch Command P_n from 4.5 to 10.5 Percent

It is evident that the tri-level (1σ) process has essentially the same transient response as the bi-level two-stage process. It should be noted that the initial separation of the two expected value curves is due to the difference in steady-state levels at the Command p_n of 4.5 percent. The response of the tri-level (2σ) process is slow relative to the other two processes. Even after 17 scans the tri-level (2σ) process has not reached its steady state-value which is 34.2. Therefore, it is concluded that for confidence intervals of two-sigma or greater the transient response is too slow. Further attempts to optimize the size of the confidence interval is not provided in this analysis. The significant point is that for confidence intervals on the order of one-sigma, no great lag is introduced into the system.

(F) Conclusions. Three versions of controlling the quantizer threshold via the SCF concept have been described and analyzed. It has been shown that each version can be modeled as a Random Walk with reflecting barriers. As a consequence, it has been possible to analyze both the long-term steady-state and short-term transient response via the theory of Markov Chains. By investigating descriptive parameters of the steady-state conditions, the three versions of the SCF systems have been compared.

The first argument has shown that the system with single- and two-stage decision processes for bi-level incrementation have nearly equivalent operating characteristics. As a result, the two-stage process could serve as an acceptable replacement of the single-stage process. Next, the assertion has been proved that a tri-level two-stage SCF could improve the control system by reducing the variation in the noise level produced by the quantizer. However, the tri-level process introduces another system parameter, namely the size of the confidence interval for the azimuth density count. By investigating the transient response resulting from a change in Command p_n, it has been shown that for two-sigma and larger confidence intervals a sluggishness in reaching the steady-state can be anticipated. On the other hand, the process with one-sigma confidence interval responds essentially the same as the bi-level process. Therefore, the tri-level two-stage SCF system with one-sigma confidence intervals is asserted as being the best choise of the SCF systems presented.

Application of Markov Chains to the analysis of the SCF concepts has demonstrated the power of this discipline. Further performance characteristics can be obtained via this analytical approach. For example, the following are important issues which can be evaluated through Markov modeling:

1. Sensitivity of this system to the number of threshold levels.
2. Transient effect on the threshold levels due to targets in the zones.
3. Effect of fixing the range density parameters R1 independent of Command p_n.

4. Anticipated SCF response under changing noise levels in the analog video.

5. Sensitivity of this system to zone size.

APPENDIX B
ANALYSIS OF TARGET DETECTION PERFORMANCE VIA THE PREDETECTOR/FINAL DETECTOR CONCEPT

Target detection in radar video processing systems is most commonly accomplished through the use of a Sliding Window Detector (SWD). An illustration of how a sliding window is formed is given in figure B-1. An infinite binary sequence is generated by the quantization of analog video which is at a constant range and on successive sweeps. A sliding window is the finite binary sequence formed by considering n consecutive positions in the infinite sequence at a time. This window is successively shifted clockwise one position on each sweep. At each position, the window is summed, i.e., the ones are counted. If this sum exceeds a fixed threshold, t, a target detection is declared at range R (figure B-2). The threshold is chosen to achieve a fixed probability of a false alarm (P_{fa}) i.e., probability of declaring a target when none is present.

A modification of the conventional SWD described above involves the use of a two-stage detection method employing a predetector followed by a final detector (PD/FD). Instead of a sliding window being shifted and summoned at each sweep position, it is required that a predetection criterion be satisfied before the window is summed. Final detection is still based on a window sum exceeding a threshold, but the difference is that a window is summed only if a predetection occurs.

An example of a typical predetector is illustrated in figure B-3. This predetector is called the three-out-of-three (3/3). The criterion for predetection is simply that three video ones are present in three consecutive positions of the infinite binary sequence. If the criterion is satisfied, a window centered about the middle position of the three consecutive ones is summed. If this sum exceeds a fixed threshold, t, a target detection is declared at range R.

A more complicated predetector is the sequential 4/8 shown in figure B-4. For this method three things are required:

1. Two video ones in a row.

2. At least one video one in any of the three positions to the immediate left of the two consecutive video ones.

3. At least one video one in any of the three positions to the immediate right of the two consecutive video ones.

The sequential 4/8 predetector is applied at every other sweep position in the infinite sequence. If the criterion is satisfied, a window centered on the predetector is summed. This sum is then compared to a fixed threshold for final detection.

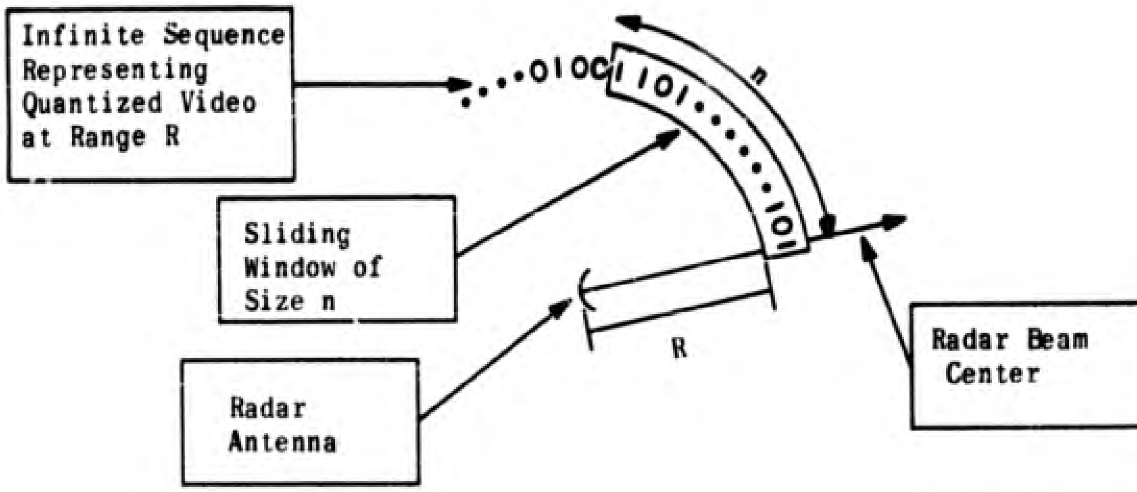


Figure B-1. Illustration of Formation of Sliding Window

. . . . 1 0 0 0 1 0 1 [0 1 0 . . . . . . 1 0 1] 0 1 1 0 0 0 . . .

$X_1 X_2 X_3 \quad \ldots \quad X_{n-2} X_{n-1} X_n$

If $\sum_{i=1}^{n} X_i \geq t$, a target detection is declared.

Figure B-2. Sliding Window Detector

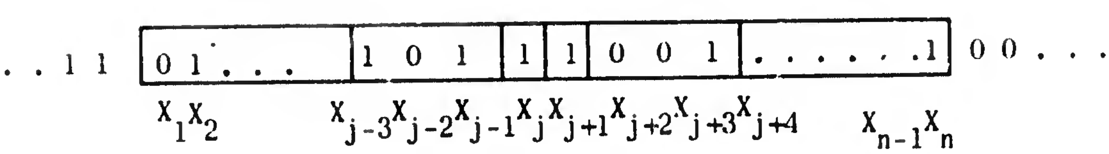
. . . . 0 1 1 0 [1 0 . . . . [1][1][1] . . . . 0 1] 0 1 0 0 . . . .

$X_1 X_2 \quad \ldots \quad X_{j-1} X_j X_{j+1} \quad \ldots \quad X_{n-1} X_n$

If $X_{j-1} = X_j = X_{j+1} = 1$ (where X_j is the center position of the window)

and $\sum_{i=1}^{n} X_i \geq t$, a target detection is declared.

Figure B-3. PD/FD with 3/3 Predetector



If $X_j = X_{j+1} = 1$ (where X_j is the center position in the window) and $X_{j-3} + X_{j-2} + X_{j-1} \geq 1$ and $X_{j+2} + X_{j+3} + X_{j+4} \geq 1$, and $\sum_{i=1}^{n} X_i \geq t$, a target detection is declared.

Figure B-4. PD/FD With Sequential 4/8 Predetector

Predetector/final detector schemes are of great importance when target detection is performed in a computer. They use substantially less computer processing time than the conventional SWD. The following analysis is provided to appraise whether there is a loss in target detection performance when utilizing a PD/FD concept. It must be emphasized that no attempt is made to find a best PD/FD method. The analysis simply illustrates two particular PD/FD methods and compares their performance to that of the conventional SWD.

(A) Approach

(1) Monte Carlo Simulation. The analysis employs Monte Carlo techniques to simulate the quantized radar video representative of a target. In the present analysis, a mathematical model is used to describe a target return pattern. A future study to be conducted by Westinghouse will provide more information regarding actual target return patterns.

(2) Model Assumptions. The targets simulated are assumed to be in a clutter free environment, and the probability of quantization , P_n, in any range cell due to noise alone is assumed to be constant. A standard one-way SinX/X antenna pattern is assumed. This leads to the observation that as the radar beam center crosses a target, the mean signal return voltage increases proportional to the function $(SinX/X)^2$ until the beam is directed at the target. At this point the return reaches its maximum. The voltage then decreases proportional to $(SinX/X)^2$.

In this analysis target strength is measured in terms of decibles (db). The formula used for conversion of mean return voltage to db is the following:

$$db = 20\log(V_2/V_1) \qquad (B\text{-}1)$$

where db = target strength in decibles
V_2 = mean target return voltage
V_1 = RMS noise level

When the term "3db target" is used in this analysis it refers to a target whose mean signal return voltage, V_2, at beam position zero satisfies the equation

$$3db = 20\log(V_2/V_1). \qquad (B\text{-}2)$$

The definition of "6db target" is analogous to the 3db target definition.

(3) Target Simulation. For a target of a given strength it is possible to determine the probability of quantization, P_q, for each range cell in a neighborhood of the target. A graph of these probabilities is called the probability ensemble for the target. The probability ensemble illustrated in figure B-5 is formed by using the voltage return at each beam position to compute the corresponding Pq. This probability can be computed using the following formula:

$$P_q = \int_{V_T}^{\infty} \frac{R}{V_1^2} \exp.\left(-\frac{R^2+V_2^2}{2V_1^2}\right) I_o\left(\frac{RV_2}{V_1^2}\right) dR$$

where V_T = voltage threshold corresponding to the desired P_n

V_1 = RMS Noise Level

V_2 = RMS voltage of signal return

$$I_o(Z) = \sum_{n=0}^{\infty} \frac{1}{(n!)^2} \left(\frac{Z}{2}\right)^{2n}$$

Finally, random numbers between zero and one are generated for each beam position and compared to the value of the probability ensemble at the beam position in question. If a random number is higher than the value of the probability ensemble at a given beam position, a zero is set for this position, otherwise a one is set. The result is a binary sequence representing the quantized video in a neighborhood of the target.

(4) Analysis Parameters. The parameters used in the model are chosen to make it as realistic as possible. Terminal radars have a pulse repetition frequency (prf) of 1200, a scan rate of 15 revolutions per minute (rpm) and a 3db beamwidth of 1½ degrees. From this information the 3db beamwidth, in terms of azimuth sweeps, is calculated to be 20. A P_n of 0.1 is used in the analysis. All windows considered in the analysis are of odd lengths. The choice of odd length windows is convenient for centering predetectors and, if desired, the performance of even sized windows can be easily interpolated from the odd length data.

A weak target is used to measure the relative performance of the detection criteria. From figure 20 in the main body of the report it can be seen that the maximum probability of quantization of a 3db target is 0.54. This low value indicates that the 3db target is weak for the radar system is used in the analysis. For this reason a 3db target is chosen for performance comparisons.

(B) Performance Comparisons

(1) Predetectors evaluated. Simulated targets are generated in the manner described above using the parameters discussed in the previous paragraph. The performance of the PD/FD approach utilizing the 3/3 and sequential 4/8 predetectors is compared to that of the conventional SWD.

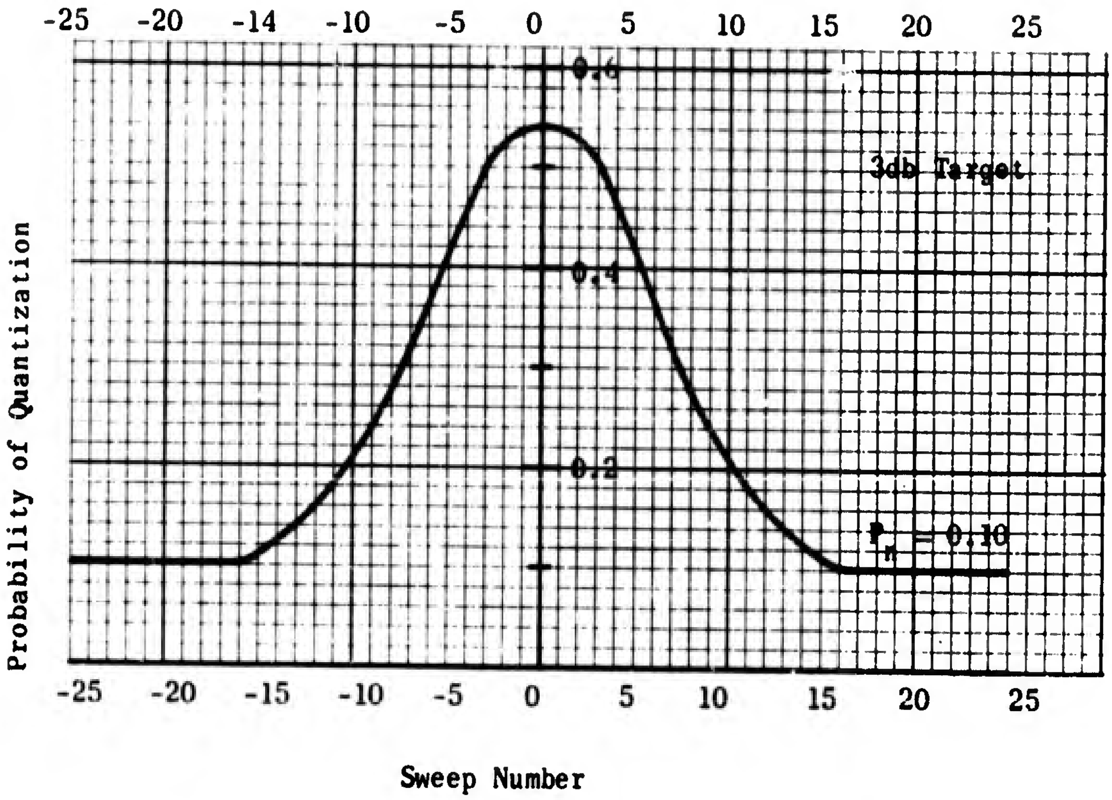


Figure B-5. Probability Ensemble Curve

(2) Probability of Detection vs. Probability of False Alarm. Performance comparisons of detection schemes take into account both probability of detection, P_d, and probability of false alarm, P_{fa}. P_d is simply the probability that a target will be declared when one is actually present. P_{fa} is the probability that a target will be declared when only noise is present. A target detection scheme can be said to perform better than another scheme at a given P_{fa}, if its P_d at that P_{fa} is higher than the P_d of the other scheme.

For this analysis the P_d is empirically calculated by generating one thousand targets and counting the number of detections declared. The choice of one thousand targets gives a 95 percent confidence interval of approximately $\pm.03$ for P_d. The accuracy of the data measuring the relative performance of the detection methods is further enhanced by the fact that each comparison was made using the same simulated target patterns.

The calculation of P_{fa} for the three detection methods studied is given in the following paragraphs. Basic to the calculation is the observation that the quantization of distinct range cells due to noise alone is a stochastically independent event.

For a sliding window of size w and a threshold t, the P_{fa} for the SWD is computed from the binomial distribution as follows:

$$P_{fa} = P(S \geq t)$$

In this formula, S is the window sum, a random variable which has the binomial distribution with parameters $N = w$ and $p = P_n = 0.1$.

The P_{fa} for the PD/FD scheme using a 3/3 predetector is computer as follows:

$$P_{fa} = (P_n)^3 \, P(S - 3 \geq t - 3) = (.1)^3 \, P(S - 3 \geq t - 3)$$

In this formula $(P_n)^3$ is the probability that a predetection has taken place, i.e. three video ones in a row; S is the window sum; and t is a fixed threshold. The random variable (S - 3) has a binomial probability distribution with parameters $N = w - 3$ and $p = P_n = 0.1$.

The calculation of P_{fa} for the PD/FD scheme employing the sequential 4/8 predetector is a bit more complicated. A final detection can occur following predetections wherein the number of ones out of the 8 positions considered in the predetection stage of this scheme is 4, 5, 6, 7, or 8. The formula for the P_{fa} with a final detector window size w, threshold t, and $P_n = 0.1$ is given as follows:

$$P_{fa} = \sum_{i=4}^{8} P_i \cdot P(S - i \geq t - i)$$

In the formula, P_i equals the probability of predetection with i ones in the eight positions considered, and S equals the window sum, where i = 4, 5, 6, 7, and 8. The random variable (S - i) has a binomial distribution with parameters N = w - i and p = P_n = 0.1.

(3) Comparisons Made. The analysis is performed in such a way that the P_d is empirically determined for the conventional SWD and the two PD/FD schemes at P_{fa}'s running from 10^{-6} to 10^{-3}. The P_{fa} is controlled by varying the detection threshold, t.

In general the performance comparision can be viewed as consisting of two stages. The first stage involves the determination of the window sizes which perform best for a given detection scheme. The second stage involves comparing the performance of all three detection schemes, each one operating with its maximum performance window sizes.

(D) Results.

(1) Optimum Performance Window Sizes. Figure B-6 is a graph of P_{fa} versus P_d for the SWD operating on simulated 3db targets with window sizes 9 through 17. The data points are connected by interpolation lines for ease in interpretation. The points for window sizes 15 and 17 are interconnected because their performance is nearly identical. Window sizes 15 and 17 perform better than window sizes 9 through 13 over the entire range of P_{fa}'s considered in the analysis. The low P_d's reinforce the assertion made earlier that 3db is a weak target for the radar system.

Figure B-7 is a similar graph for window sizes 15 through 25. Window sizes 15 and 17 perform better than window sizes 19 through 25. Note that optimum performance for the SWD is attained when using window size 15 or 17.

Figure B-8 illustrates the performance of the 3/3 predetector followed by a final detector windows of size 9 through 21. Final detector window sizes 19 and 21 perform better than window sizes 9 through 17. Figure B-9 indicates that final detector window sizes 19 and 21 perform better than window sizes 23 and 25. Hence optimum performance with the 3/3 Predetector is obtained using a final detector window size of 19 or 21.

Figure B-10 and B-11 indicate that optimum performance when using the sequential 4/8 predetector is attained with window sizes 19 or 21.

(2) Comparison of All Methods Evaluated. Figure B-12 is a comparison of the performance of the SWD with PD/FD schemes using the 3/3 predetector and the sequential 4/8 predetector. The window sizes used are the best performing window sizes for each detection scheme. The difference in performance is negligible in the 10^{-6} to 10^{-5} P area. The performance of the PD/FD schemes drops off slightly in the 10^{-4} to 10^{-3} P_{fa} area. Figure B-13 is a comparison of the performance of the three schemes on 6db targets. Note that the scale is expanded in order to facilitate the plotting of the data points. The difference in performance

is less than .02 over the entire range of P_{fa}'s considered in the analysis. This figure is negligible, especially when compared to the high P_d's.

(E) Conclusions. There is negligible detecting loss when using PD/FD schemes on 3db target with P_{fa} in the 10^{-6} to 10^{-5} range. A slight loss is realized in the 10^{-4} to 10^{-3} P_{fa} range. On 6 db targets there is negligible target detection loss over the P_{fa} range from 10^{-6} to 10^{-3}.

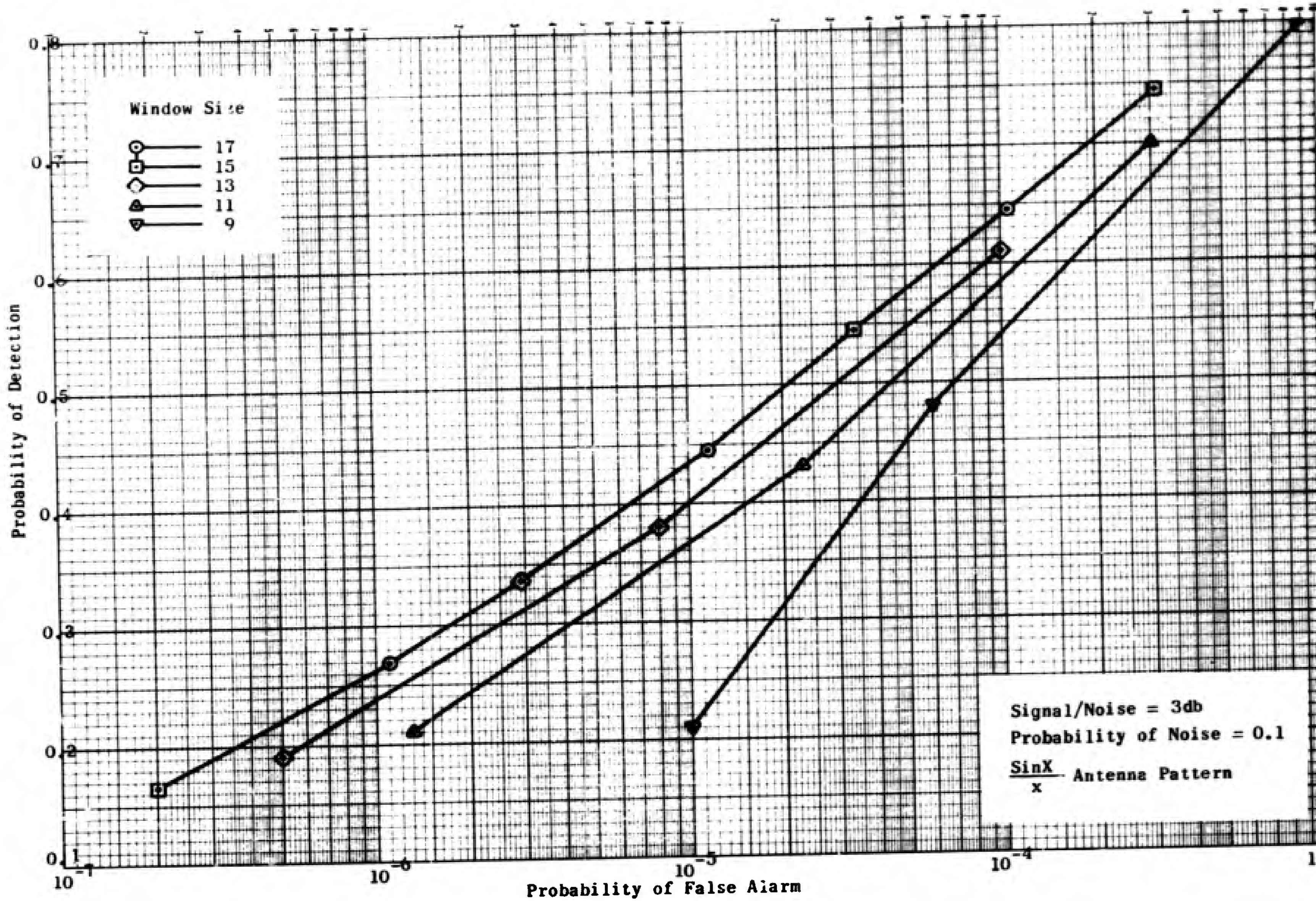


Figure B-6. Performance of Sliding Window Detector (Window Sizes 9-17)

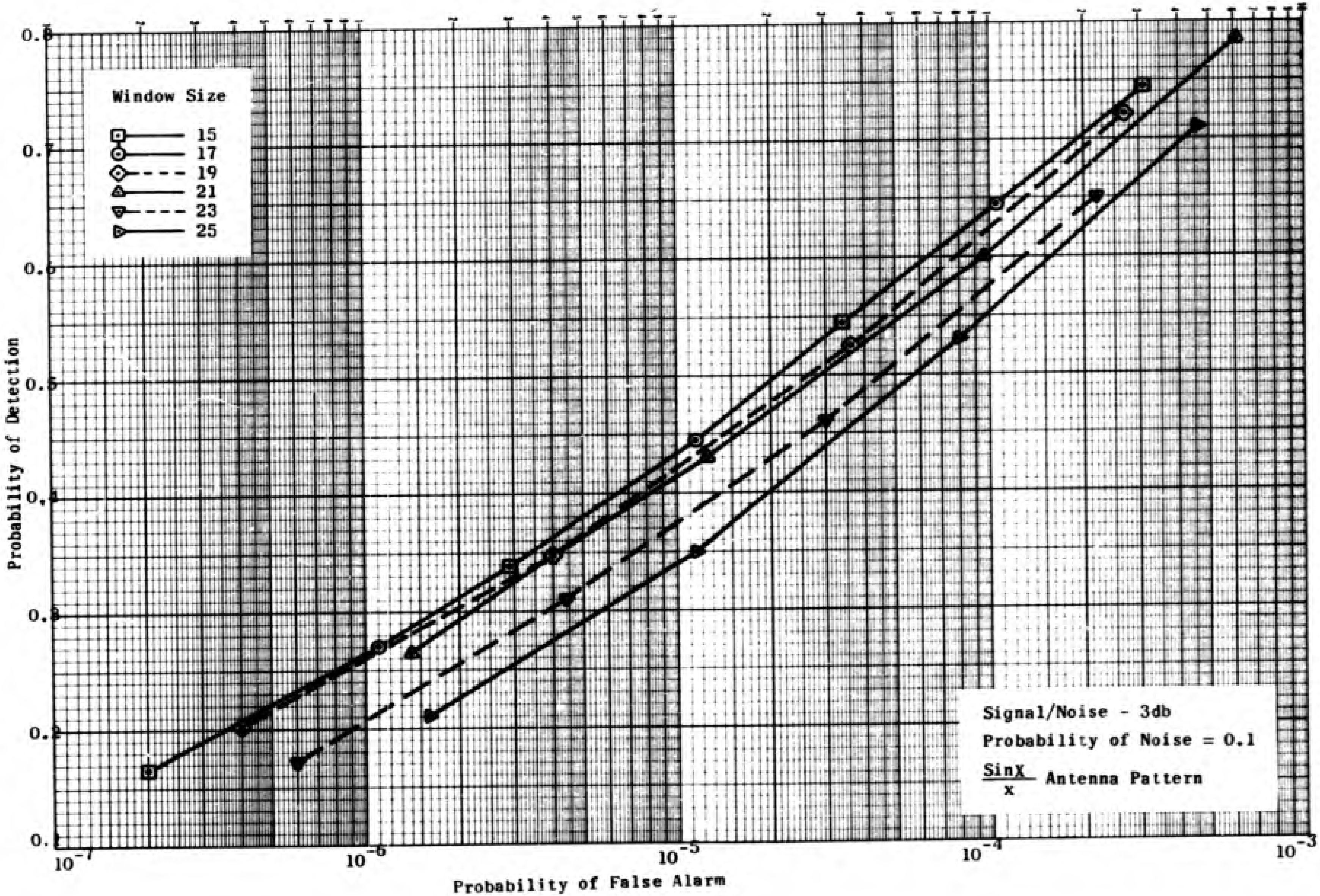


Figure B-7. Performance of Sliding Window Detector (Window Sizes 15-25)

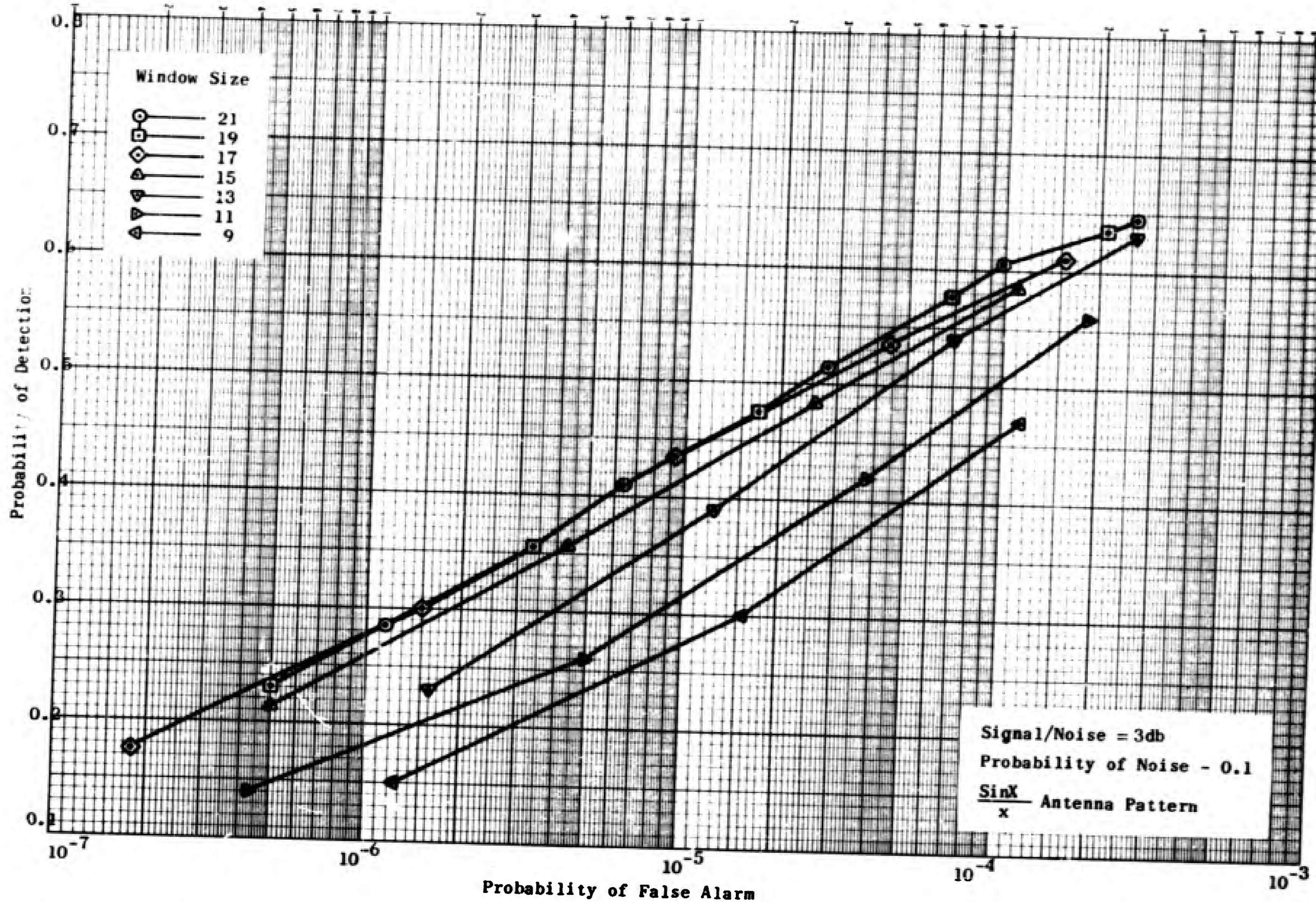


Figure B-8. Performance of 3/3 Predetector (Final Detector Window Sizes 9-21)

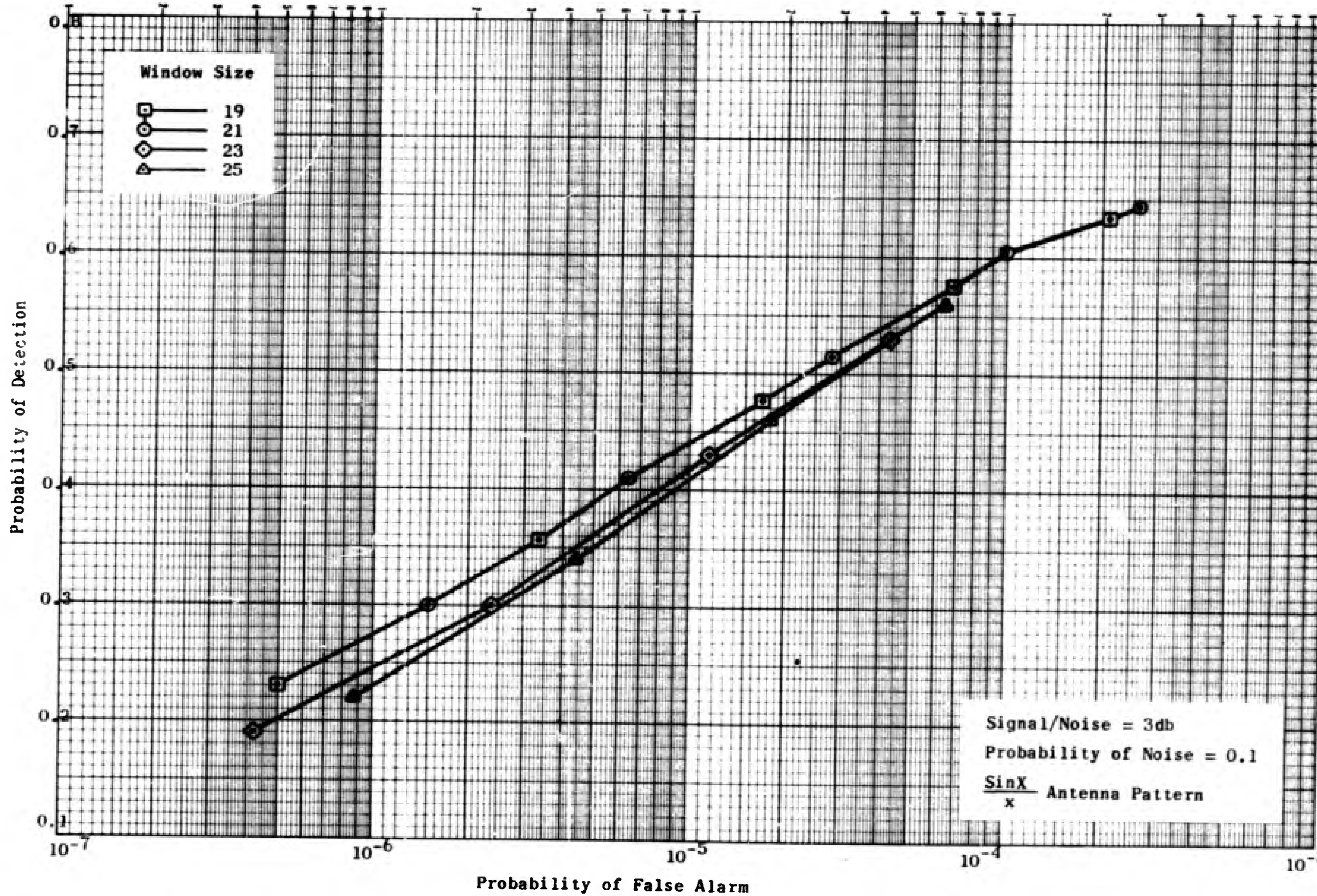


Figure B-9. Performance of 3/3 Predetector (Final Detector Window Sizes 19-25)

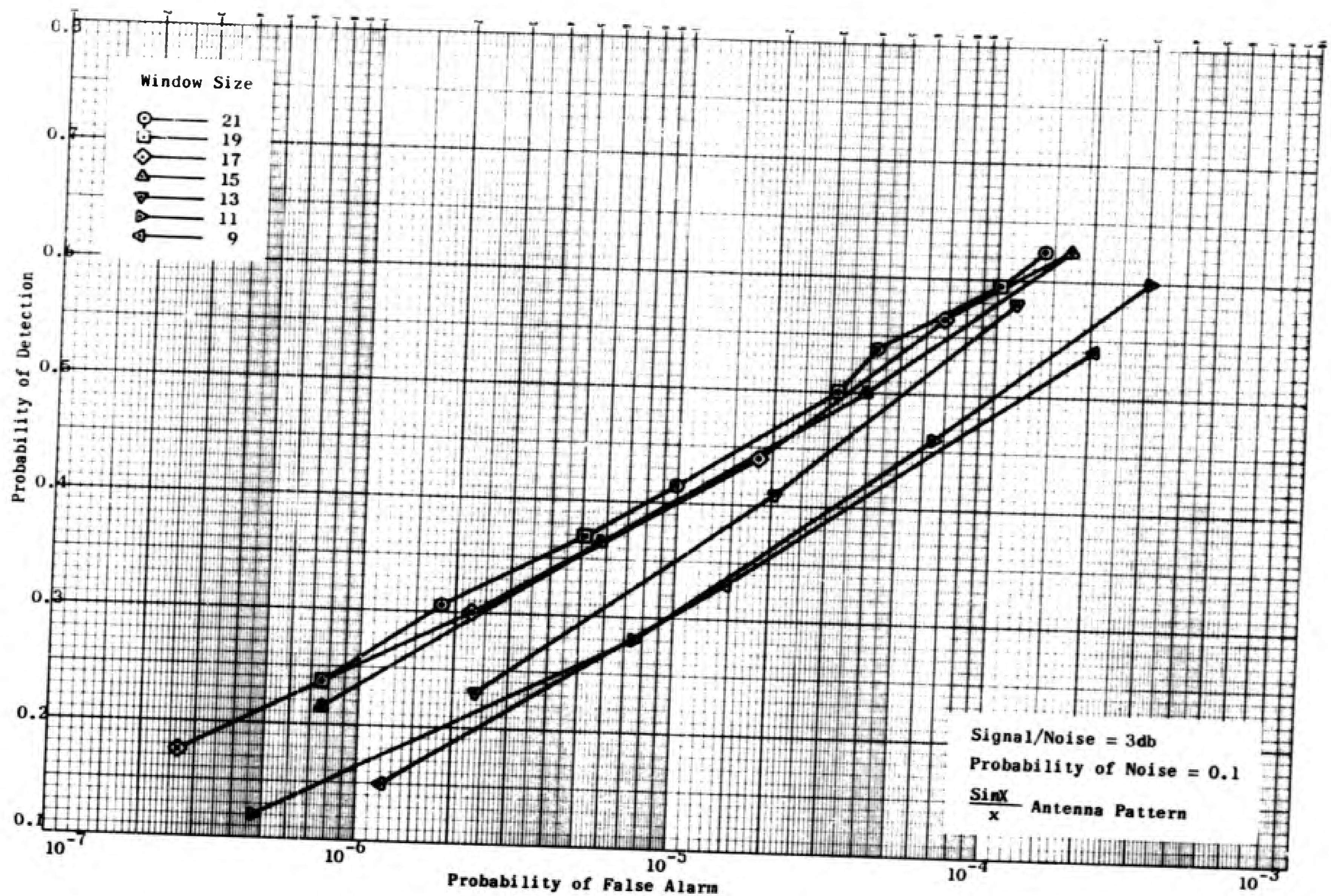


Figure B-10. Performance of Sequential 4/8 Predetector (Final Detector Window Sizes 9-21)

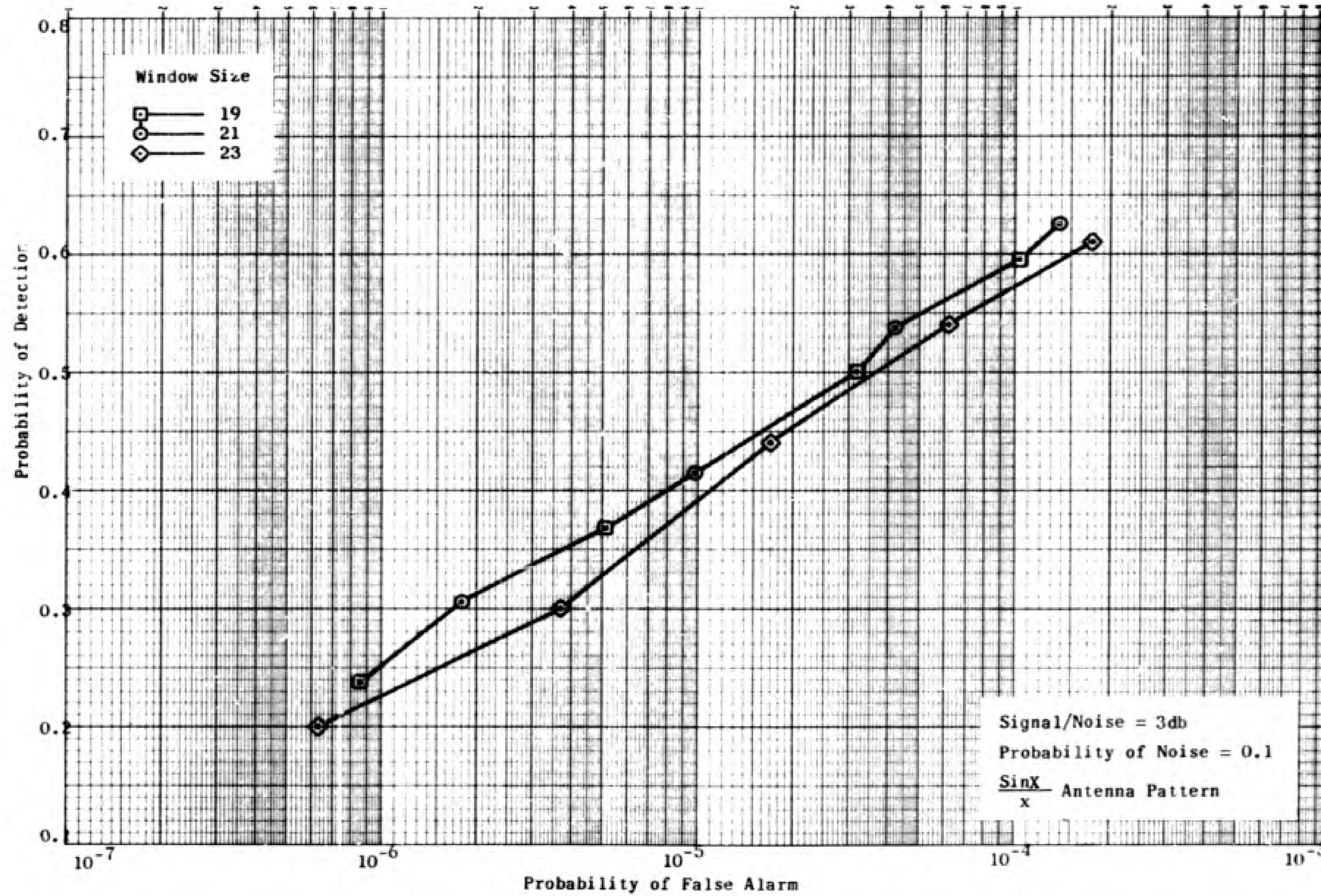


Figure B-11. Performance of Sequential 4/8 Predetector (Final Detector Window Sizes 19-23)

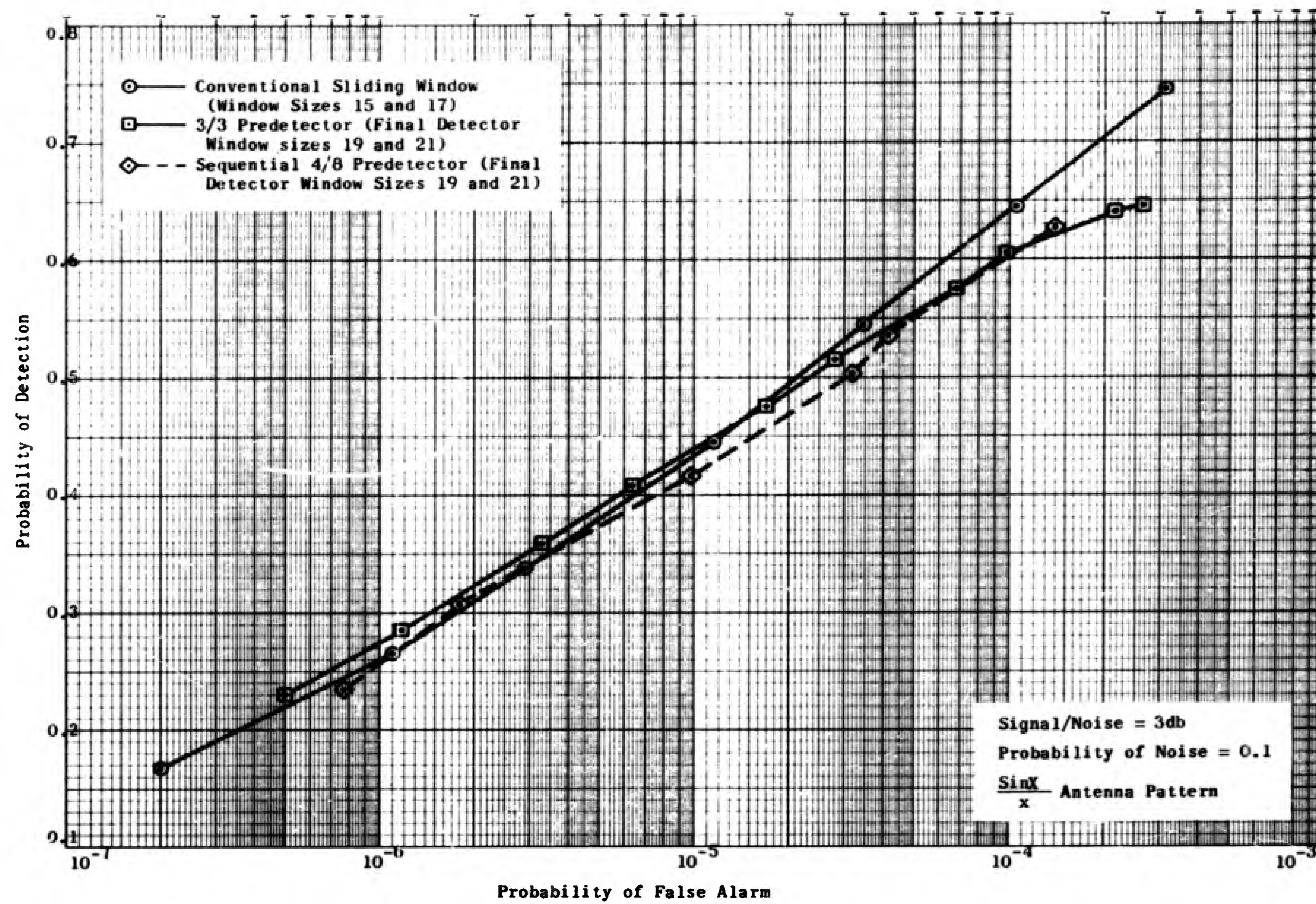


Figure B-12. Performance Comparison of Sliding Window Detector and Two PD/FDs (3db Target)

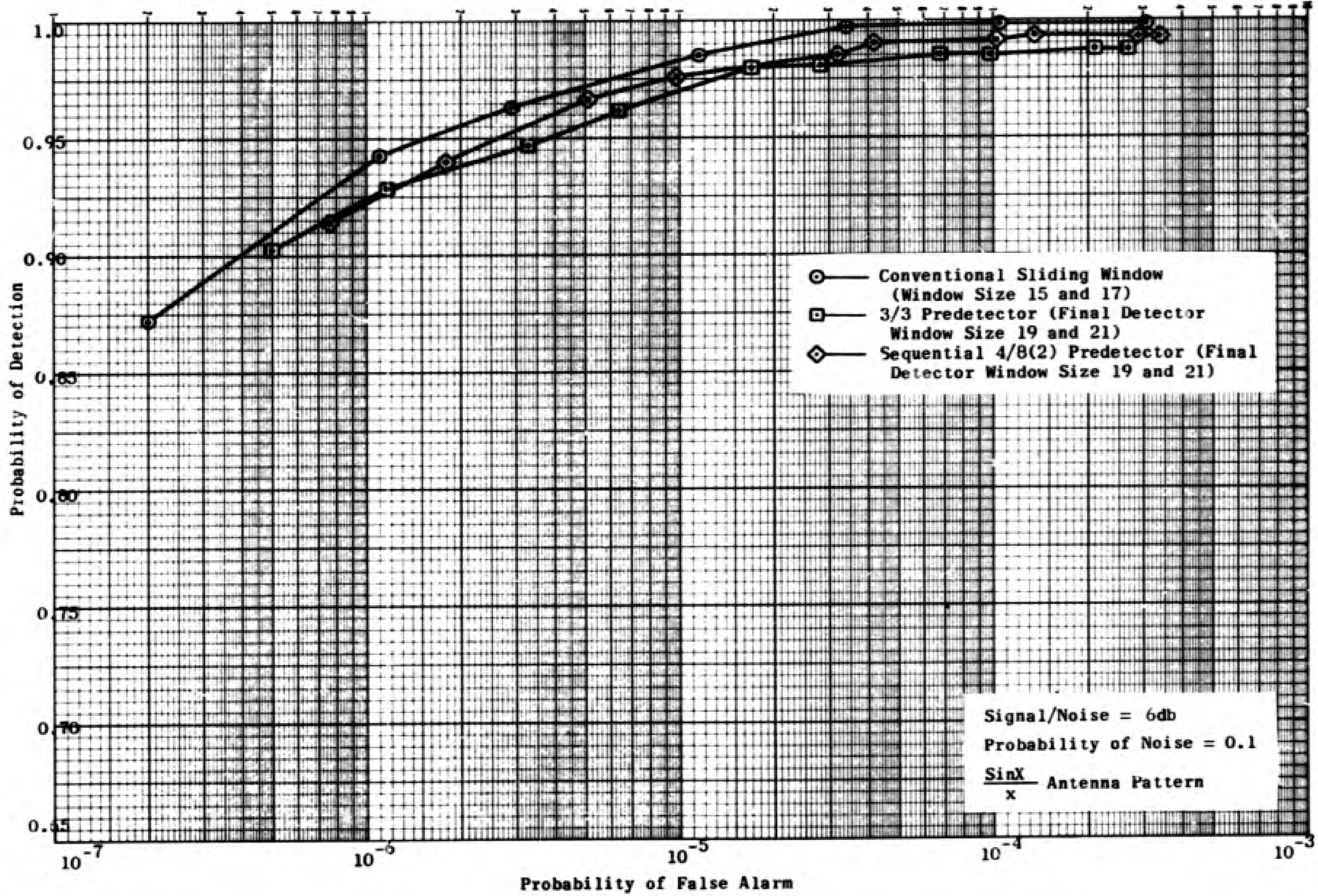


Figure B-13. Performance Comparison of Sliding Window and Two PD/FDs (6db Target)

BLANK PAGE

APPENDIX C
STATISTICAL FORMULATION OF THE AUTOMATIC CLUTTER ELIMINATOR TECHNIQUE

The method of target lead edge declaration employed in the Terminal Modified Radar Video Data Processor (TMRVDP) system makes use of a technique called the Automatic Clutter Eliminator (ACE). One of the basic problems in any radar system is to detect targets in a clutter environment. Target declaration in the TMRVDP is based upon the event that the number or ones in a window exceeds a variable threshold. The performance of this detection system is a function of how well the threshold is chosen. If the threshold is set too high, a weak target might not be detected. If the threshold is set too low, too many false detections result from noise and clutter. The ACE technique makes the assumption that in an (m x n) region of range cells, the probability of quantization in any of these cells due to noise and clutter alone is nearly constant. Figure C-1 illustrates such a region. By statistically sampling the region it is possible to set a target detection threshold for the center window of size m such that the probability of false alarm is a fixed value.

(A) Basic Assumptions. Before setting down the statistical formulation of the ACE technique, certain assumptions must be made concerning (m x n) range neighborhood in question (figure C-2). First it must be assumed that the quantization of each of the range cells in the neighborhood is accomplished with statistical independence. It must also be assumed that the probability of quantization in any of the range cells in the neighborhood other than the cells in the center window, w_0, is equal to some value P_1. Furthermore, the probability of quantization in each of the range cells within w_0 is assumed to be equal to some parameter p_2.

(B) ACE as a Hypothesis Test. The ACE technique can be formulated as a conditional statistical hypothesis test (Reference 5). First consider the (m x n) range cell neighborhood, less the range cells in w_0, as m(n-1) random samples from a Bernoulli distribution with parameter $p = p_1$. Then consider the range cells within w_0 as a sample of size m from a Bernoulli distribution with $p = p_2$. The null hypothesis, H_0, is simply that p_1 equals p_2. This is equivalent to saying that the quantized range cells in the center window, w_0, have the same distribution as the quantized range cells in the rest of the m x n neighborhood. If H_0 is accepted, a target is not declared in w_0.

The alternative hypothesis, H_1, is that $p_2 > p_1$. The rejection of H_0 in favor of H_1 is equivalent to saying that the probability of quantization of the range cells within w_0 is higher than it is in the range cells in the rest of the neighborhood. This phenomenon is assumed to be due to the presence of a target, and a target declaration is then made.

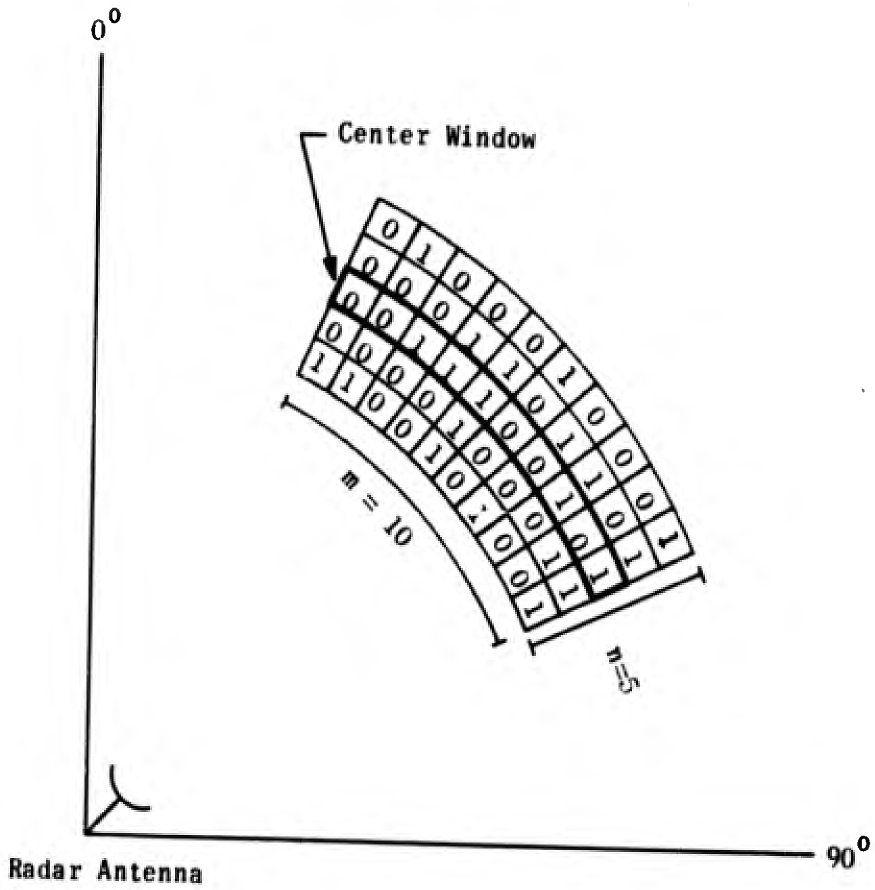


Figure C-1. Illustration of a m x n Range Cell Region

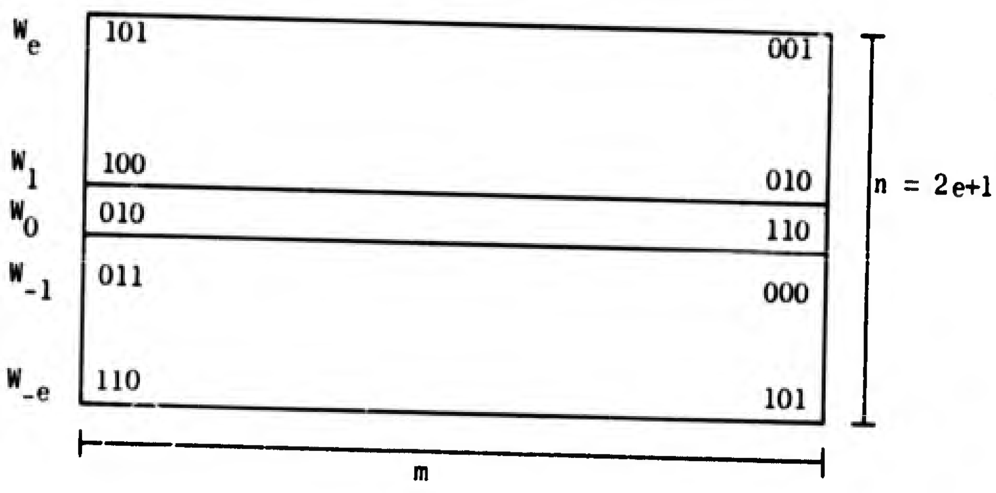


Figure C-2. Generalized m x n Range Cell Region

Let the random variable J equal the number of ones in w_0, the random variable K equal the number of ones in the rest of the m x n neighborhood, and the random variable S equal the number of ones in the entire m x n neighborhood. A hypothesis test of H_0 against H_1 will be constructed on the basis of the observed values of J and S. The probability of a type I error, α, in this test (i.e., the probability of rejecting H_0 when H_0 is true) is actually the probability of declaring a target when only noise and clutter are present. Putting this another way α = probability of false alarm (P_{fa}). The critical region defining the test will be set up as follows: H_0 will be rejected in favor of H_1 if $J \geq t$ where t is a threshold dependent upon S, and chosen to attain a fixed P_{fa}.

The test can be set up to run under any value of P_{fa} (= α) by the appropriate choice of t. Let s be the observed value of S. Then

$$\begin{aligned}
P_{fa} &= \alpha = P\ (\text{reject } H_0;\ H_0 \text{ true}) \\
&= P(J \geq t;\ S = s,\ p_1 = p_2) \\
&= P(J = t,\ K = s - t;\ S = s,\ p_1 = p_2) \\
&+ P\ (J = t + 1,\ K = s - t - 1;\ S = s,\ p_1 = p_2) \\
&+ \ldots + P\ (J = \min\ (m,s),\ K = \max\ (s - m,\ 0);\ S = s,\ p_1 = p_2) \\
&= \frac{P(J = t,\ K = s - t,\ S = s,\ p_1 = p_2)}{P(S = s,\ p_1 = p_2)} \\
&+ \frac{P(J = t + 1,\ K = s - t - 1,\ S = s,\ p_1 = p_2)}{P(S = s,\ p_1 = p_2)} \\
&+ \ldots + \frac{P(J = \min\ (m,s),\ K = \max\ (s - m,\ 0),\ S = s,\ p_1 = p_2)}{P(S = s,\ p_1 = p_2)}
\end{aligned} \tag{C-1}$$

Letting $p_1 = p_2 = p_0$ and observing that under the null hypothesis the random variable J is binomial with $N = m$, $p = p_0$; K is binomial with $N = m\ (n - 1)$, $p = p_0$; and S is binomial with $N = m \times n$, $p = p_0$, we get that

$$P_{fa} = \alpha = \frac{\binom{m}{s} p_0^{\,t} (1-p_0)^{m-t} \binom{m(n-1)}{s-t} p_0^{\,s-t} (1-p_0)^{m(n-1)-s+t}}{\binom{mn}{s} p_0^{\,s} (1-p_0)^{mn-s}}$$

$$+ \frac{\binom{m}{t+1} p_0^{\,t+1} (1-p_0)^{m-t-1} \binom{m(n-1)}{s-t-1} p_0^{\,s-t-1} (1-p_0)^{m(n-1)-s+t+1}}{\binom{mn}{s} p_0^{\,s} (1-p_0)^{mn-s}} + \ldots$$

$$+ \frac{\binom{m}{\min(m,s)} p_0^{\,\min(m,s)} (1-p_0)^{m-\min(m,s)} \binom{m(n-1)}{\max(s-m,0)} p_0^{\,\max(s-m,0)} (1-p_0)^{m(n-1)-\max(s-m,0)}}{\binom{mn}{s} p_0^{\,s} (1-p_0)^{mn-s}}$$

$$= \frac{\binom{m}{t}\binom{m(n-t)}{s-t}}{\binom{mn}{s}} + \frac{\binom{m}{t+1}\binom{m(n-1)}{s-t-1}}{\binom{mn}{s}} + \ldots + \frac{\binom{m}{\min(m,s)}\binom{m(n-1)}{\max(s-m,0)}}{\binom{mn}{s}} \qquad \text{(C-2)}$$

Note that $P_{fa} = \alpha$ is dependent only on the dimensions of the range cell neighborhood (m x n), the observed value of S, and the threshold t. Moreover, $P_{fa} = \alpha$ is computed from a hypergeometric distribution. Thus for a given m and n, and an observed s, a threshold t can be determined which will test H_0 against H_1 with a given P_{fa}.

As an example of how this test can be applied, consider the range cell neighborhood with m = 17 and n = 5 (figure C-3). The conditional hypothesis test will be applied to window w_0 of size 17 given that s = 10 with a desired $P_{fa} \leq 10^{-4}$. Thus the threshold, t, must be chosen so that $P_{fa} = \alpha \leq 10^{-4}$. From equation (C-2), t must be chosen so that

$$P_{fa} = \alpha = \frac{\binom{17}{t}\binom{68}{10-t}}{\binom{85}{10}} + \frac{\binom{17}{t+1}\binom{68}{10-t-1}}{\binom{85}{10}} + \ldots + \frac{\binom{17}{10}\binom{68}{10}}{\binom{85}{10}} \geq 10^{-4}$$

(C-3)

In the above formula if $t = 7$, $P_{fa} = 3.77 \times 10^{-4} > 10^{-4}$ and if $t = 8$, $P_{fa} = .567 \times 10^{-4} < 10^{-4}$. Thus given $s = 10$ and using 17 x 5 range cell neighborhood, a threshold of 8 would be used as a detection criterion to insure that $P_{fa} \leq 10^{-4}$.

W_2	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0
W_1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0
W_0	0	1	1	0	0	1	0	1	0	1	1	1	0	1	1	0	0
W_{-1}	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
W_{-2}	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0

$n = 5$

$m = 17$

Figure C-3. A 17x5 Range Cell Region

(C) Conclusion. It has been demonstrated how the ACE technique can be formulated in terms of a conditional statistical hypothesis test with the probability of type I error equal to the probability of false alarm.

20309