FAA-RD-74-123

Project Report
ATC-28

A Simulation of the DABS Sensor for Evaluating Reply Processor Performance

R. J. McAulay
V. Vitto

16 September 1974

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

I. INTRODUCTION

In the design of an upgraded sensor for the third generation Air Traffic Control system many options are available ranging from antenna pattern design to a variety of signal processing techniques. One of the major difficulties confronted by the designer is to determine which of these options provides the most cost-effective solution to the problem of providing valid target detection, reliable data transfer and monopulse azimuth estimation in the context of the Discrete Address Beacon System (DABS). One of the most severe problems which the sensor must be able to deal with is the interference that will be generated by the present-day Air Traffic Control Radar Beacon System (ATCRBS) with which the DABS system will have to coexist during the transition period. Since detailed analysis of the cumulative effect of this interference on the performance of a DABS sensor is difficult, it was necessary to develop a computer simulation program for both the sensor functions and the interference background.

This report describes the elements of a simulation program that was designed to perform a realistic evaluation of a variety of reply processing techniques, antenna design parameters and receiver characteristics for a DABS sensor. The reply processing techniques are limited to the generation of information bit and monopulse off-boresight azimuth estimates for DABS downlink messages. The report describes the detailed characteristics of two elements of the simulation program; the data generator and reply processor, and the high degree of versatility incorporated within these elements to allow for a great many performance tradeoff studies.

In addition, a model of the fruit environment expected to be observed by a DABS sensor located in the NAFEC area in 1980 is presented. This model is used, along with some typical simulation results for a particular reply processor configuration operating in that fruit environment to show how future DABS sensor performance can be predicted and suitable designs chosen.

II. SIMULATION PROGRAM

In order to perform systems studies of the performance of a DABS sensor using projected estimates of the ATCRBS interference environment it was essential that a realistic computer simulation be developed of the DABS sensor hardware, of the DABS reply processor, and of the target and interference signals that would be processed by the sensor. In this section a brief description of the simulation program will be given to show that realistic measures of performance can be obtained. The block diagram in Fig. 2.1 illustrates the essential features of the program. The individual blocks will now be described.

DABS Waveform Generator

It has been established [1] that the DABS downlink waveforms are to be made up of a 16 chip preamble and either 56 or 112 bit messages using 0.5 μsec chips. A typical waveform is shown in Fig. 2.2. Each message bit is encoded into two chips using delay and complement coding so that each DABS waveform consists of either 128, or 240 chips. In the simulation program the message bits can be set to correspond to the all "one's" condition or they can be randomized with a one and a zero bit being equally likely. In practice the DABS message bits are parity encoded [1]. The efficacy of this coding algorithm can be evaluated by applying it to the above sequence of message bits. The DABS signal-to-noise ratio (SNR) and azimuth are read into the program as parameters. Using the given value of the SNR, the DABS signal amplitude prior to the antenna is computed. Then using the DABS azimuth and the antenna pattern subroutine, which will be described later, the DABS signal amplitude at the outputs of the antenna ports can be obtained.

ATCRBS Waveform Generator

At the same time that a DABS waveform is received there may be one or more overlapping ATCRBS replies. In the simulation program these are

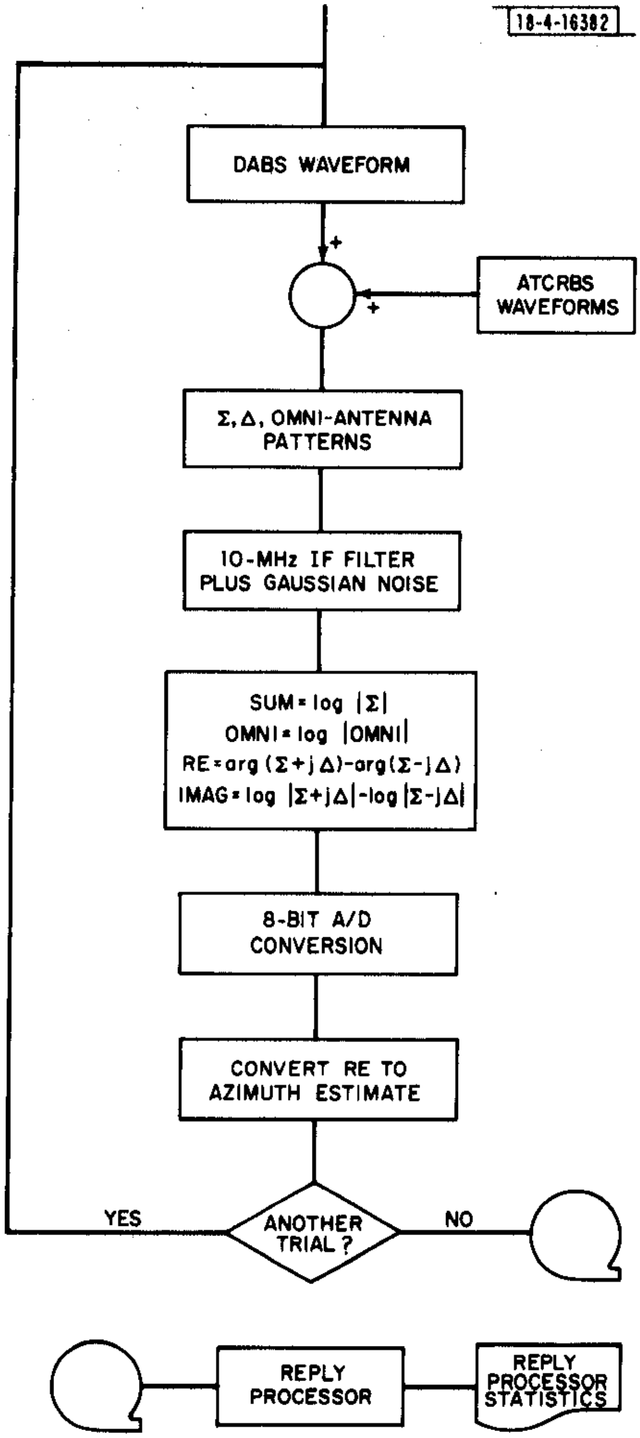

18-4-16382
DABS WAVEFORM
+
ATCRBS WAVEFORMS
+
Σ, Δ, OMNI-ANTENNA PATTERNS
10-MHz IF FILTER PLUS GAUSSIAN NOISE
SUM = log |Σ|
OMNI = log |OMNI|
RE = arg (Σ + j Δ) - arg (Σ - j Δ)
IMAG = log |Σ + j Δ| - log |Σ - j Δ|
8-BIT A/D CONVERSION
CONVERT RE TO AZIMUTH ESTIMATE
YES
ANOTHER TRIAL ?
NO
REPLY PROCESSOR
REPLY PROCESSOR STATISTICS

Fig. 2.1. Simulation program block diagram.

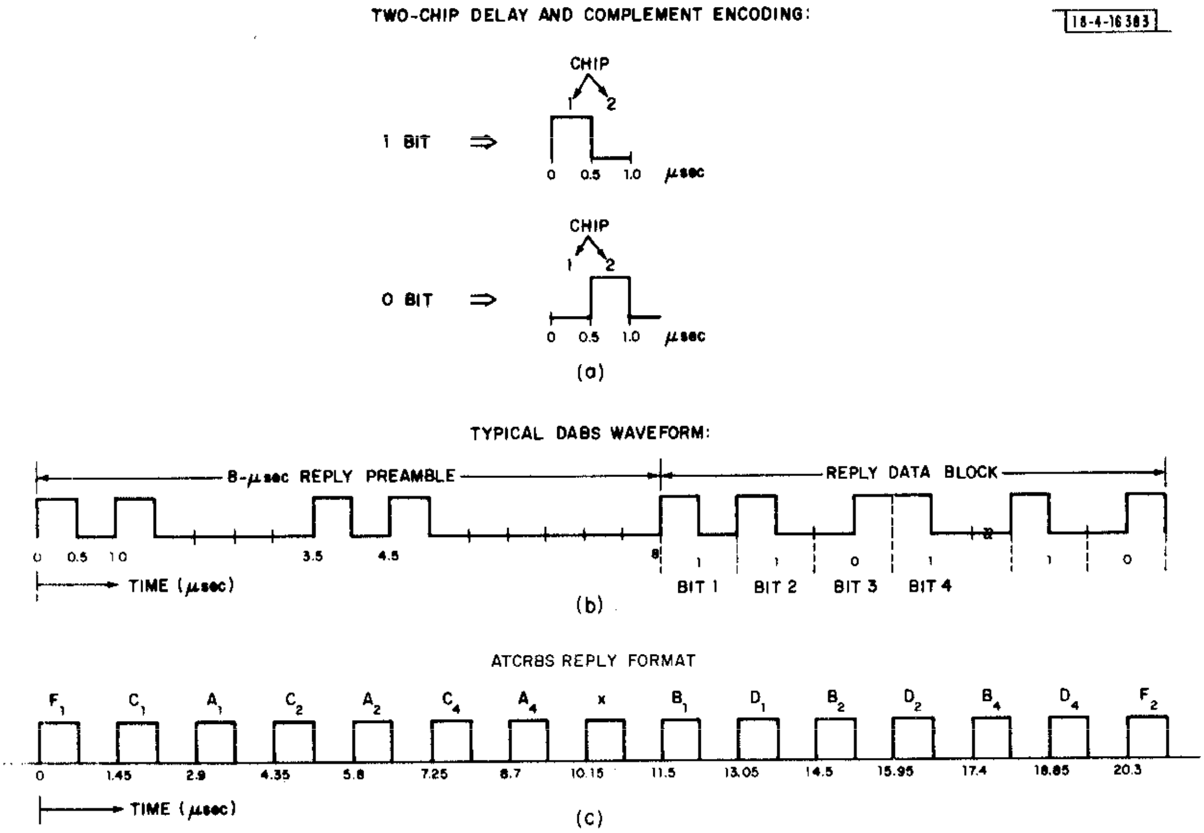


Fig. 2.2. Two-chip delay and complement encoding.

specified in terms of 15 bit messages where each bit is a .45 μsec pulse, as shown in Fig. 2.2b. Except for the F1 and F2 pulses which are always on, and the X pulse, which is always off, the remaining pulses in the message can be randomized. In all of the results reported here, all message pulses were turned on to represent a worst case situation.

As in the DABS waveform case, the ATCRBS waveform characterization is completely specified by the signal amplitude, azimuth and arrival time. These parameters must be chosen according to models for the ATCRBS fruit environment. These will be discussed in greater detail in the next section but for the moment, suffice it to say that an ATCRBS signal amplitude, azimuth and time of arrival can be chosen in a probabilistic way from models for the ATCRBS fruit, hence the waveform before and after antenna filtering can be completely specified.

Antenna Pattern Processing

Since the goal of the simulation program was to perform sensor parameter tradeoff studies, it was necessary to characterize the antenna patterns in a parametric way. For example, one important tradeoff study is the exploration of the effects of varying the sidelobe level while keeping a fixed beamwidth. Fortunately, the Taylor Pattern Illumination Functions [2] can be specified in such a way to permit this type of tradeoff study. In Fig. 2.3 we have illustrated typical even and odd antenna patterns designed to achieve -26 dB sidelobe levels. The omni antenna pattern model is also shown. Notice that idealized antenna patterns are used. However, the program has been generalized to allow for modelling of hardware errors in the amplitude and phase taper illumination functions, although this option was rarely used in the actual simulation runs.

Data Generator

This program is the heart of the simulation as it takes the DABS and ATCRBS message specifications and forms sampled-data sequences which

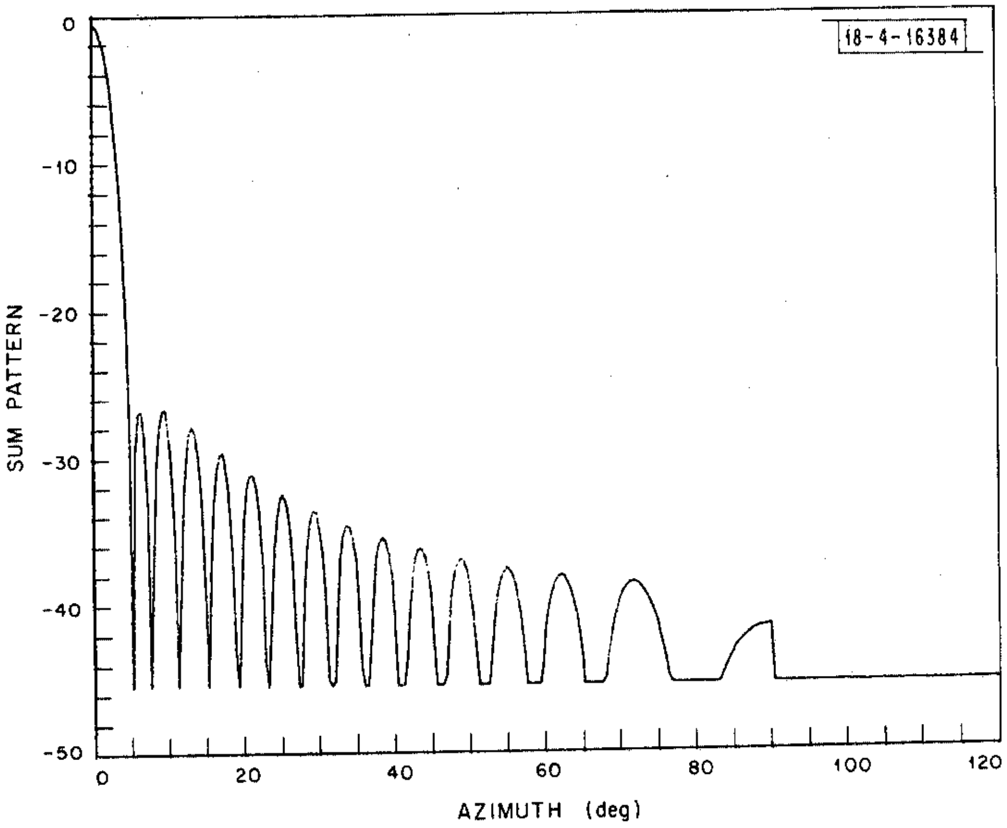


Fig. 2.3a. "Sum" beam antenna pattern.

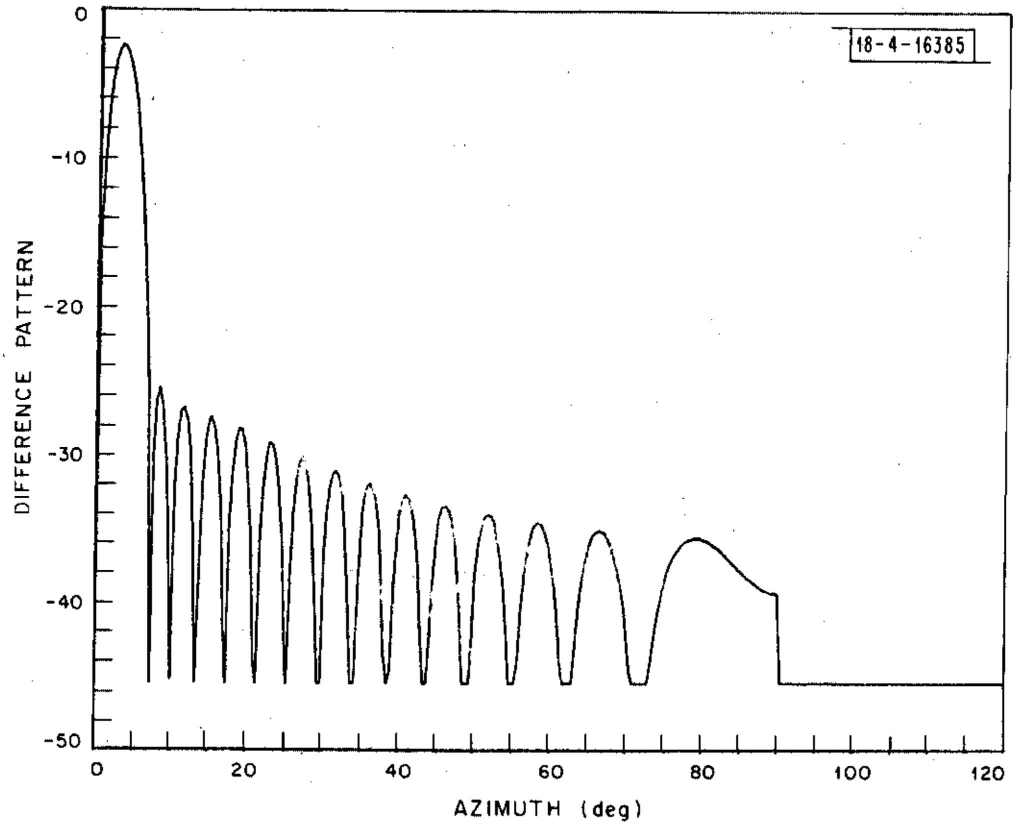


Fig. 2.3b. "Difference" beam antenna pattern.

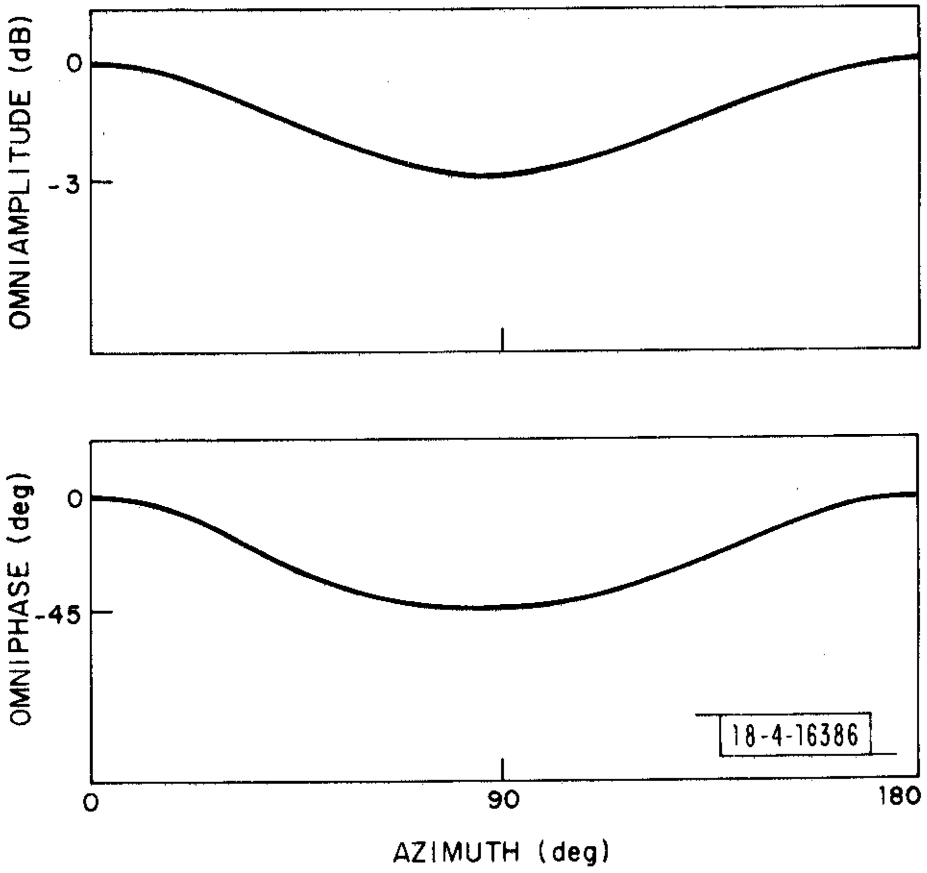


Fig. 2.3c. Omni antenna pattern.

correspond to the four essential hardware outputs of the DABS sensor: log sum, log omni, angle channel and interference channel. These quantities are defined in the following way: the signals at the output of the sum, difference and omni antennas at any particular sampling point, the kth say, are given by

$$\Sigma(k) = A_s e^{j\varphi_s} G_\Sigma(\theta_s) + \sum_{m=1}^{M} A_m e^{j\varphi_m} G_\Sigma(\theta_m) + n_\Sigma \qquad (2\text{-}1)$$

$$\Delta(k) = A_s e^{j\varphi_s} G_\Delta(\theta_s) + \sum_{m=1}^{M} A_m e^{j\varphi_m} G_\Delta(\theta_m) + n_\Delta \qquad (2\text{-}2)$$

$$0(k) = A_s e^{j\varphi_s} G_0(\theta_s) + \sum_{m=1}^{M} A_m e^{j\varphi_m} G_0(\theta_m) + n_0 \qquad (2\text{-}3)$$

The definition of the parameters in these equations are the following:

1. A_s and θ_s represent the DABS amplitude and azimuth. These are specified input parameters.
2. φ_s represents the DABS phase. From pulse to pulse within a reply, this is an independent uniformly distributed random variable on $(0, 2\pi)$. From sample to sample within a pulse this phase can be made to increase or decrease linearly to represent a fixed frequency offset with respect to 1090 MHz.
3. $G_\Sigma(\theta)$, $G_\Delta(\theta)$, $G_0(\theta)$ represent the attenuation due to the sum, difference and omni antenna beam patterns respectively.
4. A_m, θ_m and φ_m represent the amplitude, azimuth and phase for the mth of M ATCRBS replies. These quantities are chosen probabilistically using models for the projected ATCRBS fruit environments. It should

be noted that A_m and θ_m are random from reply to reply, but once picked, are constant throughout the reply.

5. n_Σ, n_Δ and n_0 represent the additive white Gaussian noise samples due to the front end of the mixer preamplifier of the sum, difference and omni antenna channels. When we use the term signal-to-noise ratio (SNR), we are referring to the quantity $A_s^2/2\sigma^2$ where

$$\overline{|n_\Sigma|^2} = \overline{|n_\Delta|^2} = 2\sigma^2$$

All of these parameters are brought together in the data generator subroutine to form the simulated sampled data RF signals $\Sigma(k)$, $\Delta(k)$ and $0(k)$. As in the real-world version of the DABS sensor, these signals are first filtered at RF, thereby rendering the DABS and ATCRBS waveforms sequences of non-square pulses. In the simulation this is accomplished using a first order filter. Letting y(k) denote the sum, difference or omni signals at time k, then

$$y(k+1) = x(k+1) \qquad (2\text{-}4)$$

where

$$\begin{aligned} x(k+1) &= \alpha x(k) + (1-\alpha)\, y(k) \\ x(0) &= 0 \qquad (2\text{-}5) \\ \alpha &= \exp(f_c T) \end{aligned}$$

where f_c is the bandwidth of the filter and T is the time between samples. Typically we use the values f_c = 10 MHz and T = .1 μsec, where the latter quantity corresponds to a 10 MHz sampling rate.

Logarithmic Amplification

At this point the DABS sensor hardware has generated filtered versions of the sum, difference and omni RF signals. From a practical point of view the dynamic range variations of these signals can be quite large and it is necessary to perform further hardware processing before the A/D conversion can be performed. These additional operations result in the four channel outputs: log-sum, log-omni, angle and interference. The log-sum and log-omni channel outputs, denoted by $y_{\Sigma}(k)$ and $y_0(k)$, respectively, are obtained by passing the signals* $|\Sigma(k)|$ and $|0(k)|$ through log-amplifiers. The characteristic used in the simulation is shown in Fig. 2.4. In general, for a log-amplifier of D dB dynamic range and V_{max} maximum output voltage, the characteristic is given by

$$y = \log x = \begin{cases} ax & 0 \leq x \leq 1 \\ a[1 + \ln(x)] & 1 < x \leq 10^{D/20} \end{cases} \qquad (2\text{-}6a)$$

where

$$a = \frac{V_{max}}{1 + \frac{D}{20}\ln 10} \qquad (2\text{-}6b)$$

In the simulation program we let V_{max} = 3 volts and D = 80 dB resulting in a = .29382.

Azimuth Estimation and Interference Detection

The angle channel output represents the signal from which the azimuth estimate is to be derived. In hardware it is obtained by making a phase comparison of the signals $\Sigma(k) \pm j\Delta(k)$. In other words, if

* We now let $\Sigma(k)$, $\Delta(k)$ and $0(k)$ denote the filtered versions of the sum, difference and omni signals.

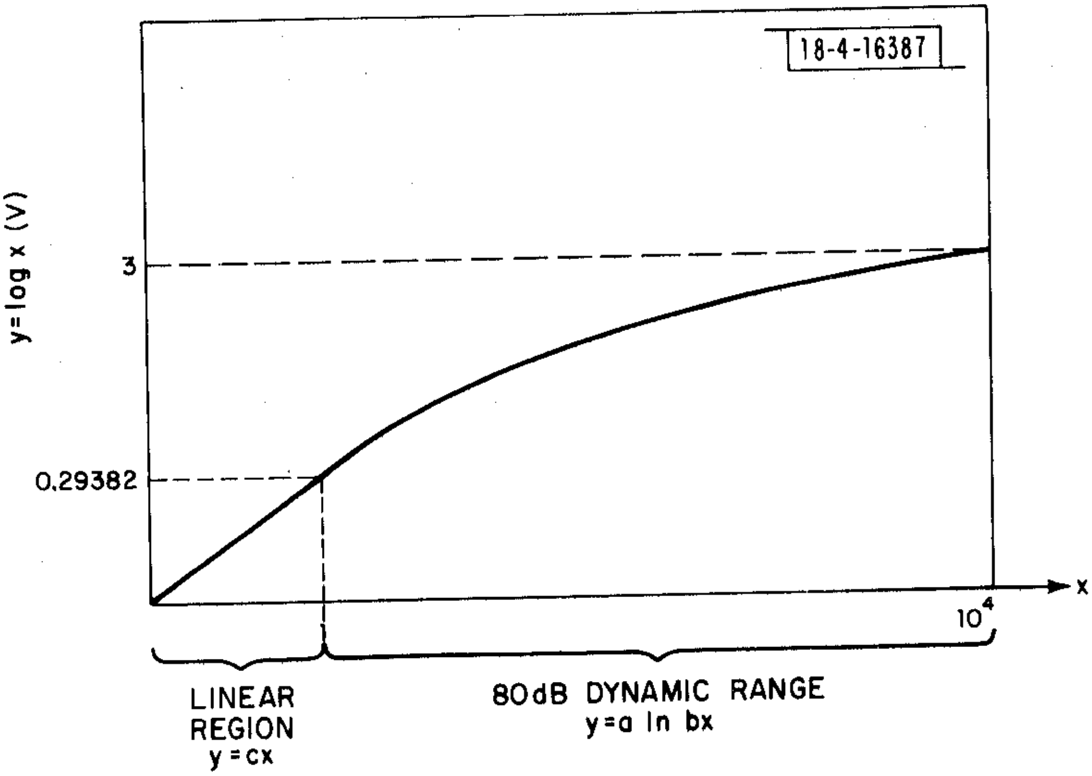


Fig. 2.4. Log amplifier characteristic.

$$\alpha(k) = \arg[\Sigma(k) + j\Delta(k)] - \arg[\Sigma(k) - j\Delta(k)] \tag{2-7}$$

then the outputs of the phase comparators are

$$y_c(k) = \cos \alpha(k) \tag{2-8a}$$

$$y_s(k) = \sin \alpha(k) \tag{2-8b}$$

which provides enough information to permit an unambiguous azimuth estimate over the entire width of the antenna's mainbeam.

The interference channel output, which is used to indicate the presence of more than one signal in the receiver [3, 4], is given by y_I where

$$y_I(k) = \log |\Sigma(k) + j\Delta(k)| - \log |\Sigma(k) - j\Delta(k)| \tag{2-9}$$

The log amplifiers used to generate y_I have the same characteristics that were described previously.

Since all of the processing in the DABS sensor is to be done using digital hardware, the above signals are quantized to represent the effects of A/D conversion. Finally the azimuth estimate, θ, is generated using the quantized versions of $y_c(k)$ and $y_s(k)$ according to the maximum likelihood algorithm [5]

$$\frac{G_\Delta(\hat{\theta})}{G_\Sigma(\hat{\theta})} = \tan[\alpha(k)/2] \tag{2-10}$$

where $\alpha(k)$ is given by (2-7) and $G_\Delta(\theta)/G_\Sigma(\theta)$ represents the normalized difference pattern. Using the half-angle formula, this becomes

$$\frac{G_\Delta(\hat{\theta})}{G_\Sigma(\hat{\theta})} = \frac{\sin \alpha(k)}{1 + \cos \alpha(k)} \approx \frac{y_s(k)}{1 + y_c(k)} \tag{2-11}$$

where the approximation indicates a negligible loss in accuracy due to the A/D quantization. The estimate is found using a table look-up for stored values of the normalized difference pattern.

Summary

At this point the simulation program would have completed a single trial. A plot of the output of a typical trial for the log sum (amplitude) and Re (azimuth) channel outputs is shown in Fig. 2.5 for a portion of a DABS reply whose message is overlapped by a stronger ATCRBS reply. Figure 2.5a shows the idealized DABS and ATCRBS waveforms. Figure 2.5b represents the resultant composite waveform that would appear at the output of the log sum channel. This waveform incorporates the effects of antenna attenuation, IF mixer preamplifier noise, IF filtering, logarithmic amplification and A/D conversion at a 10 MHz rate. The scale is adjusted so that the minimum trigger level (MTL) is set to correspond to a 12 dB SNR. In this case, the DABS waveform has a 20 dB SNR. Figure 2.5c shows the angle channel data for a 2 MHz sampling rate. This corresponds to samples taken in the center of each of the DABS chips. There are clearly two populations of azimuth estimates, one corresponding to the DABS target, the other to the ATCRBS reply. It is the goal of the reply processor to try to separate the two populations so that the DABS azimuth will be estimated on the basis of ATCRBS-interference-free data.

In other words, data would have been obtained for a single set of values for the DABS and ATCRBS amplitudes and azimuths, for the DABS and ATCRBS carrier phases and for the preamplifier noises. Since these are random variables drawn from statistical distributions, data for many such trials must be obtained. This is done by looping back to the ATCRBS signal generator so that a different set of amplitudes, azimuths and arrival times can be drawn from the fruit model and repeating the entire procedure.

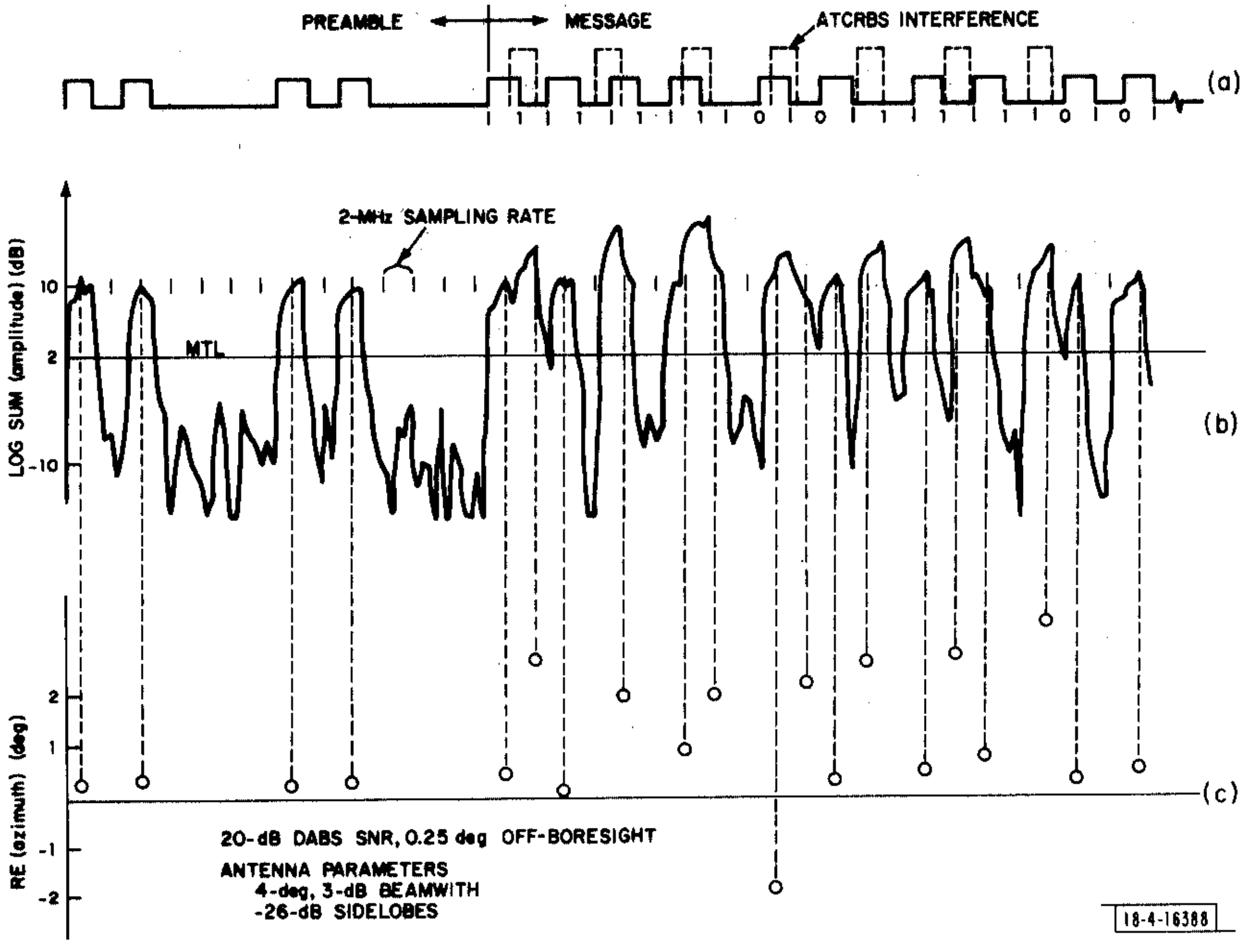


Fig. 2.5. Typical simulation data for one mainbeam ATCRBS interference.

Ultimately a data tape is generated that represents many configurations of interfering and signalling situations. These stored data are then used as input to the DABS reply processor where independent design parameter trade-off studies can be made. The algorithm for the reply processor will be discussed in a subsequent section, while in the next section the fruit models, from which the ATCRBS waveform parameters are selected, will be described.

III. ATCRBS FRUIT MODELS

In the last section we described the computer program that generated sampled data waveform sequences which represented typical DABS reply waveforms in a background of receiver noise and interference. Initiation of a specific trial required the selection of amplitudes, azimuths and times of arrival for the overlapping ATCRBS interference. In this section we shall describe the fruit models from which these random variables are chosen.

Time of Arrival

Let us use T to denote the length of a DABS reply in microseconds, where T = 64 or 120 μsec depending on whether a short or long message is transmitted. If t_0 denotes the time of arrival of the leading edge of the first pulse in the DABS preamble, then an ATCRBS reply will be a potential source of interference whenever the leading edge of the F1 pulse lies in the interval $(t_0 - 20.3, t_0 + T)$. For convenience we chose our time scale so that $t_0 = 20.3$ μsec and then pick the leading edge of the F1 pulses of the ATCRBS replies to be uniformly distributed random variables in the interval $(0, T + 20.3)$.

Angle of Arrival

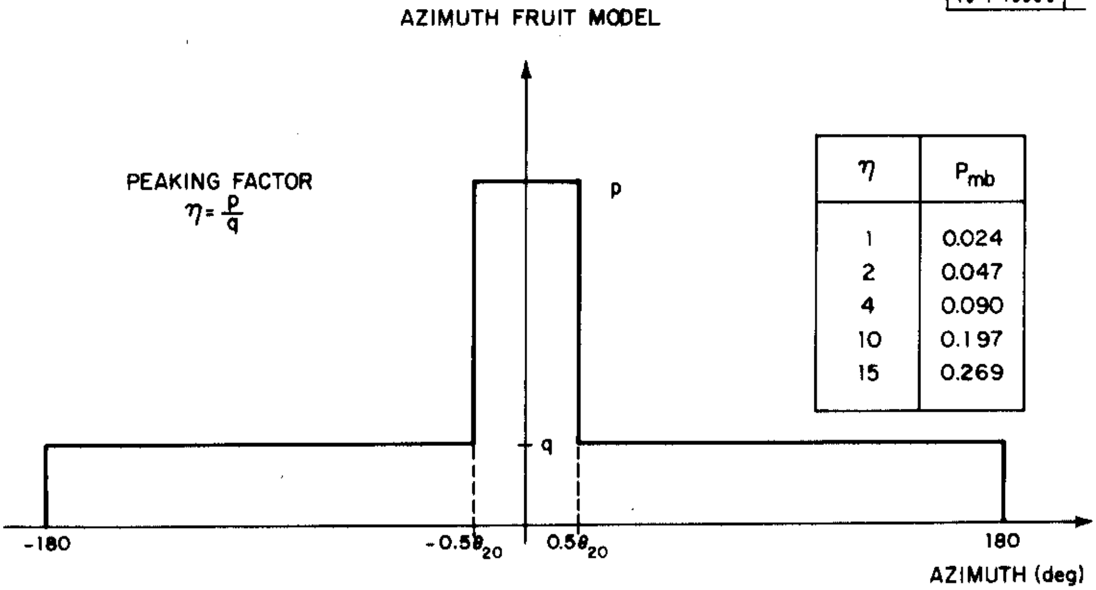
Since the DABS targets of interest are always located within the mainbeam of a highly directional antenna, the effects of the ATCRBS interference will be radically different depending on whether the interferer is located within the mainbeam or the sidelobe. The reason for this is the significant attenuation that the ATCRBS signal undergoes when it is received in the sidelobes of the antenna pattern. Therefore, it is reasonable to describe the ATCRBS azimuth by a two step distribution as shown in Fig. 3.1. In the figure, θ_{20} represents the 20 dB beamwidth of the antenna's mainbeam, p and q represent the ordinates of the mainbeam and sidelobe distribution. It is convenient to introduce the notion of a mainbeam peaking factor $\eta = p/q$

which measures the predominance of mainbeam fruit. For example, if the aircraft are equally likely to be located at any particular azimuth, then the azimuth distribution is uniform and we set $p = q$, or $\eta = 1$. There may be situations, however, where there may be more aircraft per beamwidth in a certain direction; for example, when an interrogator sweeps past a distant airport. To model this situation we simply increase the peaking factor η by some appropriate amount. Given a peaking factor, it is relatively easy to show that the probability that any particular fruit reply originated from within the mainbeam of the antenna is given by

$$p_{mb} = \frac{\eta \theta_{20}}{360 + \theta_{20}(\eta - 1)} \qquad (3\text{-}1)$$

In Fig. 3.1 values of η and p_{mb} have been tabulated for an antenna having -26 dB peak sidelobes, 4° 3 dB beamwidth and 8.46° 20 dB beamwidth. Hence, in order to generate an ATCRBS azimuth, one first draws a sample from a binomial distribution with events "mainbeam" or "sidelobe" where the probability that the event "mainbeam" occurs is p_{mb} given by (3-1). If the event "mainbeam" occurs, then we draw the actual ATCRBS azimuth from a uniform distribution on the interval $(-.5\ \theta_{20}, .5\ \theta_{20})$. On the other hand, if the event "sidelobe" occurs, then the ATCRBS azimuth is drawn from another uniform distribution on the intervals $(-180, -.5\ \theta_{20})$ $(.5\ \theta_{20}, 180)$. In the actual implementation of the simulation, results were obtained conditioned on a given number of mainbeam and sidelobe fruits and the binomial weighting applied later. This aspect of the study will be discussed in greater detail in a later section.

18-4-16389



η	P_{mb}
1	0.024
2	0.047
4	0.090
10	0.197
15	0.269

$$P_{mb} = \text{PROB. MAINBEAM FRUIT} = \frac{\eta\theta_{20}}{360 + \theta_{20}(\eta - 1)}$$

Fig. 3.1. ATCRBS fruit azimuth distribution.

ATCRBS Amplitude

In the analysis of measured fruit statistics, a model for the ATCRBS amplitudes has evolved that corresponds reasonably well to the observed data. For convenience, we express the ATCRBS amplitude in terms of its corresponding signal-to-noise ratio which we denote by SNR. The probabilistic model for the SNR is given by the rule

$$\mathrm{SNR} = \mathrm{SNR}_{\min} - 20 \log_{10} U \qquad (3\text{-}2)$$

where $\mathrm{SNR}_{\min}$ is a minimum expected SNR corresponding to a minimum power transponder at maximum range and where U is a uniformly distributed random variable on (0, 1). It is then easy to show that the probability distribution function of the SNR is

$$\Pr\{\mathrm{SNR} \leqq \psi\} = \begin{cases} 1 - 10^{(\mathrm{SNR}_{\min} - \psi)/20} & \psi \geqq \mathrm{SNR}_{\min} \\ 0 & \text{otherwise} \end{cases} \qquad (3\text{-}3)$$

This is illustrated by the dashed line in Fig. 3.2 where we have set $\mathrm{SNR}_{\min}$ equal to 32 dB. The scale is adjusted so that 10 dB SNR corresponds to a received power level of − 85 dBm. Since this gives only the SNR distribution prior to the antenna, it is of interest to show the effects of antenna processing. It is easy to see that if SNR_0 refers to the SNR at the output of the antenna, and SNR_i the SNR at the input, then

$$\mathrm{SNR}_0 = \mathrm{SNR}_i + 10 \log_{10} |G_\Sigma(\theta)|^2 \qquad (3\text{-}4)$$

Restricting the ATCRBS azimuths to be either mainbeam or sidelobe and using our trial antenna pattern, we empirically determined the distribution

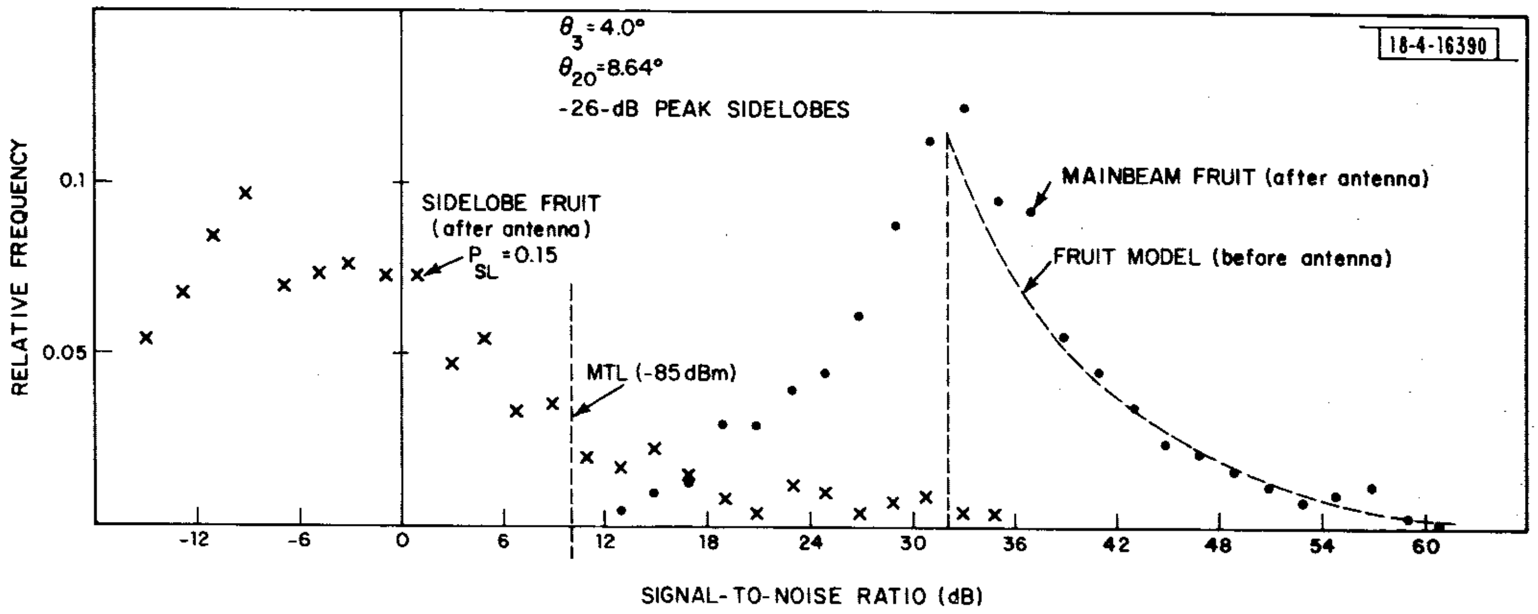


Fig. 3.2. ATCRBS amplitude fruit model.

of SNR's due to mainbeam and sidelobe fruit. These results are also illustrated in Fig. 3.2 and can be used to determine the probability that sidelobe fruit will exceed minimum threshold level (MTL), P_{SL}). This can be done by computing the area under the tail of the relative frequency curve that lies above the MTL. For this case it turns out that $P_{SL} = .15$, hence of all of the omni fruits that are being received in the sidelobes of the antenna, only 15% of them will be detectable in the sense that their amplitudes will exceed the minimum threshold level.

Summarizing the preceding results we see that for each ATCRBS fruit we pick an arrival time at random to guarantee an overlap with the DABS waveform. All of the ATCRBS pulses are then turned on, except for the X pulse and the amplitude of the pulses is given by

$$A = \sqrt{2}\ \sigma\ 10^{SNR/20} \tag{3-5}$$

where the SNR in dB is chosen according to (3-2). The ATCRBS azimuth is then chosen uniformly within the antenna 20 dB beamwidth if the particular fruit in question is mainbeam or from the entire 360° interval less the 20 dB beamwidth otherwise. This azimuth parameter is then used to obtain the gains of the sum and difference antenna patterns which in turn modulate the ATCRBS amplitude given in (3. 5). Using the fruit models in the simulation program described in the preceding section, we are now able to generate data tapes, which, after sufficiently many trials, should produce a large variety of DABS-ATCRBS interference conditions from which meaningful processor performance statistics can be evaluated. In the next section the DABS reply processor and its associated performance statistics will be described.

IV. THE REPLY PROCESSING ALGORITHM

The purpose of the simulation program described in Section II is to generate a data tape containing the sample by sample amplitude, monopulse azimuth, interference and omnidirectional channel outputs for each of a multiplicity of replies for a fixed set of parameters which define the characteristics of the antenna, receiver, A/D conversion and DABS signal characteristics. The second aspect of the program is to use these data to evaluate the performance of a variety of signal processing options to determine a cost-effective processor that can reliably decode the DABS message and estimate the aircraft's azimuth. By simulating the various reply processing algorithms in software a great many options can be examined in considerable detail. In this section one class of reply processing algorithm will be described that makes use of amplitude and azimuth consistency checks derived from the preamble. The object is to illustrate the methodology that can be used to analyze and predict the DABS sensor performance in realistic fruit environments.

Preamble Processing

For each DABS reply to be processed it is assumed that the preamble has been detected since the performance of the preamble detector has been explored in detail elsewhere [6]. An estimate of the reference amplitude and azimuth are derived from the preamble by examining the four pulses that make up the preamble and comparing the amplitude and azimuth estimates of the first and fourth, first and second, second and third and third and fourth preamble pulses. In practice, this is done by sampling the preamble at a 10 MHz rate, determining the location of the pulse edges and then taking the sample that corresponded to the middle of the pulse. In the simulation program the locations of the pulse edges are assumed known so that the "middle" samples, which occur at a 2 MHz rate, can be examined directly. The amplitudes and azimuths determined from the middle sample on each pulse are required to correlate within specified confidence intervals defined by the program input parameters. The reference amplitude is obtained by averaging the amplitude samples for the pair of pulses that correlate and have minimum amplitude [6]. If azimuth correlation also occurs on the selected pulse pair, then the average azimuth value determines the preamble reference azimuth

If an amplitude or azimuth reference cannot be obtained from the preamble, the condition is flagged and a more conservative reply processing strategy is followed.

Chip Decoding

The amplitude and azimuth preamble estimates, so derived, are then used as references to establish confidence intervals for decoding the estimates that are obtained for the remaining chips within the DABS message field. If an amplitude estimate on some chip is above MTL but outside of the established confidence interval, then it is likely due to an interference pulse and can be flagged as a one chip with low confidence. The azimuth confidence window can be used in the same way. Therefore, if two chips for a particular bit are detected above MTL, then a potential ambiguity occurs because the first level decoder must declare a one-one chip situation when only zero-one or one-zero are allowed by the delay-and-complement encoding procedure. The conflict can be resolved by examining the confidence flags, since if interference is present it is likely to be outside the amplitude and/or azimuth confidence windows and the low confidence bit will be set. Then the one-one chip situation, can be deciphered as a one-zero if the low confidence bit is set on the second chip and zero-one otherwise.

In the general case, an interference bit (I_c) is set for all chips declared above the threshold by examining the middle sample of the chip of the amplitude, azimuth, interference and omnidirectional signals. A failure to correlate the present value of amplitude and azimuth with the preamble estimates results in an interference flag setting of $I_c = 1$ indicating the presence of interference. In addition, if the interference channel indicates the presence of sidelobe interference, then I_c is also set to 1. Computationally these conditions are written as:

$$\begin{aligned}
&|\log|\Sigma| - A_p| > \lambda_o \\
&|\hat{\theta} - \hat{\theta}_p| > \lambda_1 \\
&|Q| > \lambda_2 \\
&|\log|\Sigma| - \log|\Omega|| > \lambda_3
\end{aligned} \qquad (4\text{-}1)$$

where Σ is the present amplitude measurement, $\hat{\theta}$ is the monopulse azimuth estimate (given by 2.11), Q is the interference flag (given by 2.9) and Ω is the omnidirectional channel output for the chip under study. A_p and $\hat{\theta}_p$ are the amplitude and azimuth preamble estimates. If any of the above inequalities are satisfied, the interference bit is set for that chip. Any of the above four tests can be eliminated from the processor by setting the appropriate threshold (λ_i) to zero.

Bit Decoding and Azimuth Estimation

The DABS message block consists of a sequence of non-return to zero pulse amplitude modulation (NRZ-PAM) signals where each information bit is encoded into two chips using a delay and complement signal format as described in Section II, Fig. 2.2a. An information bit equal to one is formed by a one chip followed by a zero chip and a zero bit is formed by a zero chip followed by a one chip. The bit decoding algorithm takes advantage of the PPM format and the interference flag setting for each chip to make bit decisions (E) and assign a confidence flag (C) to each bit (C = 1 implies high confidence, C = 0 low confidence). The rules for the bit decision and monopulse accumulation process used in the bit decoding process are given in Table 4-1. In the table, M_i is a 1 if the i^{th} chip exceeds MTL and is a 0 otherwise.

From this table it is clear that when preamble estimates $(\widehat{A}_p$ and $\widehat{\theta}_p)$ are available, the algorithm is willing to declare high confidence bit decisions (C = 1) when interference is present on only one of the two chips making up the information bit. In the absence of preamble estimates the processor will declare low confidence bit decisions when interference is observed on either or both of the chips.

The azimuth estimate is obtained by determining those chips in the message that are flagged as being free of interference and result in an unambiguous bit decision. The individual azimuth estimate samples $(\widehat{\theta}_1$ or $\widehat{\theta}_2)$ taken from each interference-free chip are accumulated over the entire message and the final azimuth estimate for the reply is obtained by dividing by the number of samples accumulated.

A Reply Processor Example

In Figure 2. 5 the data for a typical Monte Carlo trial was illustrated. We now use that same data sample to illustrate the reply processor algorithm. For convenience we have redrawn the data in Fig. 4. 1. First we note that all of the preamble pulses are received free of interference and hence all pairs of samples correlate within the 2 dB correlation window. The smallest average value is taken as the reference amplitude and a 2 dB confidence window drawn about it. This is illustrated by the lines drawn in Fig. 4. 1b. Since the pair of azimuth samples also correlate, a $.25^o$ azimuth confidence window is drawn about the reference azimuth. Now we examine the first pair of chips whose amplitudes both exceed MTL because of the presence of an ATCRBS pulse overlapping the second chip. Therefore a one-one chip decision is made and a potential ambiguity exists. However, the amplitude

Chip Decision		Interference Indicator		Information Bit Estimate (E)	Confidence Flag Estimate (C)		Monopulse Sample To Be Accumulated To Form Monopulse Estimate
M_1	M_2	I_{c_1}	I_{c_2}		$\hat{A}_p$, $\hat{\theta}_p$ available	$\hat{A}_p$, $\hat{\theta}_p$ available	
1	0	0	0	1	1	1	θ_1
1	0	1	0	1	1	0	None
0	1	0	0	0	1	1	θ_2
0	1	0	1	0	1	0	None
1	1	0	0	1	0	0	None
1	1	0	1	1	1	0	θ_1
1	1	1	0	0	1	0	θ_2
1	1	1	1	1	0	0	None
0	0	0	0	0	0	0	None

TABLE 4-1 REPLY PROCESSOR DECISION MATRIX

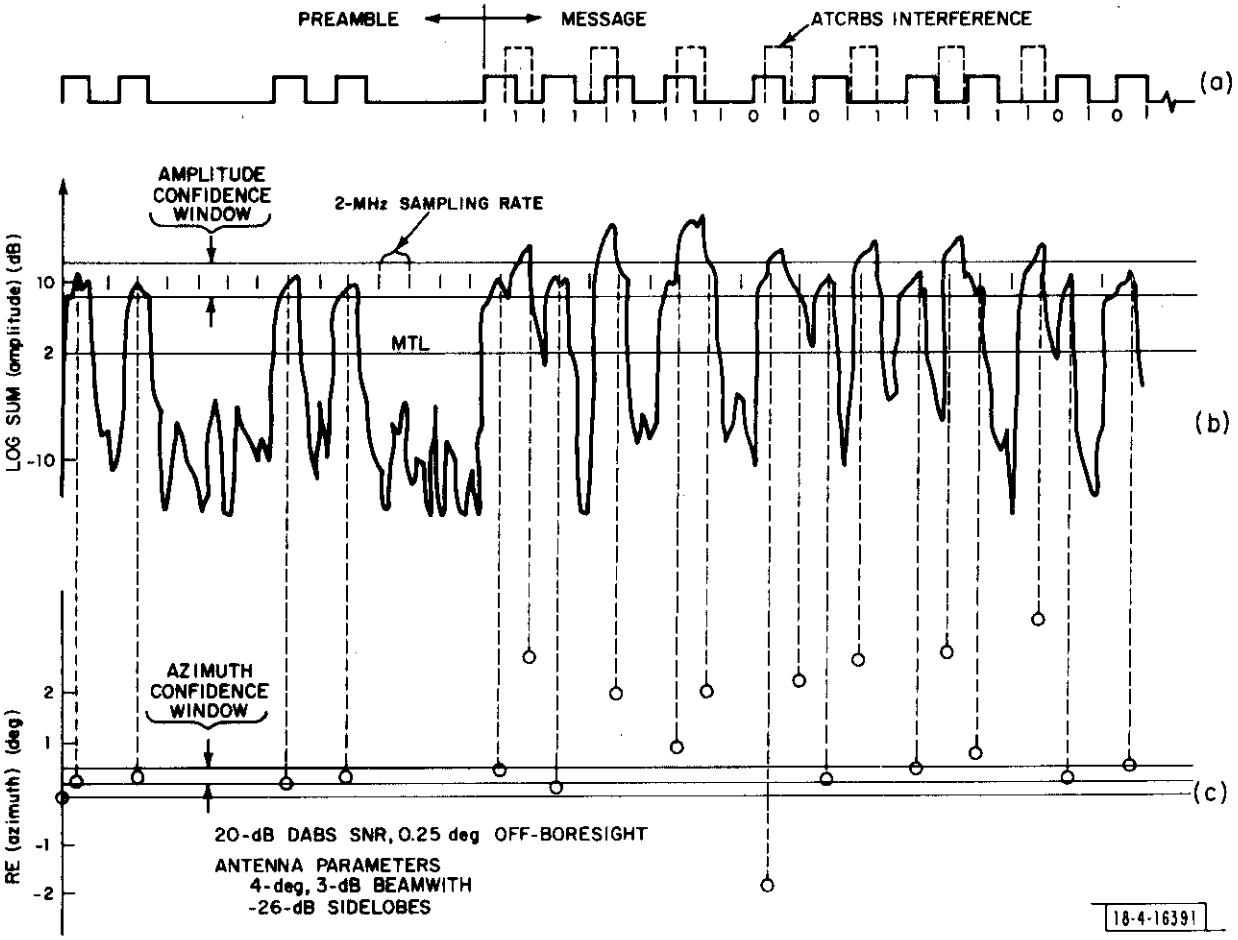


Fig. 4.1. Typical simulation data for one mainbeam ATCRBS interference.

of the first chip lies within the confidence window while the second does not, hence the processor declares a one-zero chip configuration (a one bit) with high confidence. Proceeding to the second chip pair we see that only the first chip exceeds MTL hence a one-zero chip declaration is made with high confidence. Similarly on the third chip pair, since only the first chip amplitude exceeds MTL, then the one-zero chip decision is made with high confidence even though the amplitude lies outside the confidence window due to the overlapping ATCRBS pulse. The fifth chip pair is likewise decoded as a zero-one situation since the amplitude sample on the first chip does not exceed MTL. One can proceed through the entire message this way to decode the reply. Although there was no need for the azimuth confidence window for this case, it sometimes happens that a one-one chip declaration cannot be resolved using only the amplitude confidence window as both values either lie within it or outside it, although the latter situation is a rare event. In this case, the ambiguity is resolved by examining the azimuth estimates on each chip and declaring a one for the chip whose azimuth sample falls within the confidence window.

This example typifies the excellent performance that can be achieved when amplitude and azimuth consistency checks are available from the preamble. In fact the performance is so good that it probably suggests that simpler reply processing schemes might suffice. Although not within the scope of this report, further studies have shown that indeed it is not necessary to use the consistency checking methods outlined here. However, the analytical approach to evaluating system performance is identical and we shall continue to use the above reply processor as our baseline example.

Reply Processor Performance Statistics

After the reply processor has produced the information bits, confidence flags and azimuth estimate for a particular reply read from the input data tape, a set of statistical measures are generated to allow for a final evaluation of the processor. In order to evaluate the performance of the error detection and decoding algorithm, it is important to determine the probability distributions of the number of bit errors in a reply, N_B, and the span of these bit errors, S_B. The number of bit errors the processor makes on a particular reply is obtained by comparing the transmitted DABS information bit pattern to the information bit estimates. Statistics are accumulated for the number of replies having $N_B = 0, 1, 2, \ldots, K$. The span of the bit errors is determined by counting the number of bits between the first and the last bit errors in a given reply. The resulting value of S_B is used to update statistics on the number of replies having $S_B = 0, 1, \ldots, K$. In a similar manner the number of low confidence flags that were set within the reply (N_c) and their bit span (S_c) are used to update corresponding probability functions for N_c and S_c. The number of bit errors flagged with high confidence (N_{Bc}) is also determined and a probability function updated for that parameter.

The reply processor is designed to function with an error detection and correction algorithm which will detect and correct information bit errors with high probability if the following conditions are met:

1. The algorithm has a priori knowledge of the 24-bit address of the expected DABS reply, information which will be available for all DABS targets for which track files have been established.
2. The span of information bit errors within the reply is less than or equal to 24 bits.
3. An information bit error is not flagged with high confidence.
4. The number of low confidence bits within the reply is less than some specified value. We take this value to be one half the number of information bits within the reply.

The probability functions for the numbered span of bit errors and the number of low confidence bits within a reply are not updated by any reply which fails to meet the conditions required for successful error correction given in 2 through 4 above. Any reply which fails to meet conditions 2 through 4 above will not have a monopulse estimate produced by the reply processor and will thus not be included in the final monopulse estimate statistics.

Each time the individual probability functions given for conditions 2, 3 and 4 are updated a counter is incremented. These counts are used to determine conditional probabilities for the final processor statistics to be defined later.

There is another condition which will cause no update for all of the probability functions and no monopulse estimate will be made available for the reply being processed. That condition exists when the preamble amplitude estimate ($\hat{A}_1$) does not fall within a window centered around the true amplitude of the reply. The size of the window is equal to the $\pm 2\sigma$ error expected from noise and quantization effects. If this test fails the preamble estimate is assumed to have been "captured" by interference and further processing of the reply would lead to grossly erroneous results since the span of low confidence bits would exceed the number required for reliable decoding. A count of the number of times this occurs is maintained by the program to produce an overall preamble capture for the set of Monte Carlo trials.

When all of the Monte Carlo trials have been evaluated by the reply processor, the program provides a set of overall statistics defining the performance of the processor under the interference conditions, signal-to-noise ratio, antenna and receiver characteristics and reply processing techniques specified at the time the data tape was generated. The pertinent parameters defining all of the above characteristics are output by the program followed by the overall performance measures.

The rate at which the preamble estimation algorithm fails to provide amplitude and the azimuth estimates and the preamble capture rate are given. The probability functions for the number of bits in error and their bit span, the number of low confidence bits and their bit span and the probability that an information bit error is flagged with high confidence are also printed out.

On each reply the azimuth estimation error is obtained by comparing the estimated azimuth with the true value. Statistics are then accumulated for the bias and standard deviation of the azimuth errors from which the overall rms value is computed.

One of the most important performance measures evaluated by the simulation program is the failure probability of the processor, (P_F). The processor fails whenever there is at least one bit error remaining after the application of the error detection and decoding algorithm [1]. Therefore, a reply failure does not occur whenever (i) the span of the message bit errors is no greater than 24 bits, (ii) an information bit error is not given a high confidence rating and (iii) the number of low confidence bit settings is less than one-half the number of information bits within the reply, since when these three conditions are satisfied it is assured that the error detection and correction algorithm will correct all of the low confidence bits which are in error. The failure probability can therefore be evaluated by counting the relative number of replies which lead to a violation of the above conditions. For example, if the interference captures the preamble estimates, then the number of low confidence bit settings will be greater than one-half the number of information bits within the reply. By defining the events

A = the event that interference captures the preamble estimates

B = the event that the span of the bit errors exceeds 24 bits

C = the event that one or more bit errors are flagged with high confidence

D = the event that the number of low confidence bit errors is greater than one-half the number of information bits

it is easy to show that the failure probability is given by

$$P_F = P(A) + P(\overline{A})\left\{P(B/\overline{A}) + P(\overline{B}/\overline{A})\left[P(C/\overline{B},\overline{A}) + P(D/\overline{C},\overline{B},\overline{A})\ P(\overline{C}/\overline{B},\overline{A})\right]\right\} \qquad (4-2)$$

To evaluate the failure probability and to examine the failure modes of the processor the program generates the probabilities $P(A)$, $P(B/\overline{A})$, $P(C/\overline{B},\overline{A})$, $P(D/\overline{C},\overline{B},\overline{A})$. This is done by counting the relative number of times each of the above events occur.

V. SENSOR PERFORMANCE ANALYSIS

In the preceding sections we have described the computer simulation of the DABS sensor and the reply processor algorithm to be used for decoding the DABS message and generating the azimuth estimate. The performance can be evaluated by specifying a given number of ATCRBS mainbeam and sidelobe fruits. In this section we want to develop quantitatively the number of ATCRBS fruits that must be considered and then combine the results of the simulation trials in a probabilistic way to predict the performance for the DABS sensor operating in the NAFEC area in 1980.

Omni Fruit Rate

To begin with, we introduce the notion of the omni fruit rate. This represents the number of fruits per second that exceed the minimum triggering level (MTL) that would be received by an omnidirectional antenna having the same gain as the directional antenna with which actual fruit rates were measured. In other words, the omni fruit rate, denoted λ_o measures all of the potentially detectable fruits within the 360^o scan of the antenna. The measured fruit rate, denoted λ_m, represents the number of fruits per second that exceed MTL at the output of the directional antenna. Obviously, $\lambda_m \leqq \lambda_0$ because the sidelobes render some of the omni fruits undetectable by reducing their signal strengths below MTL. In fact, the exact relation between λ_0 and λ_m is

$$\lambda_m = \lambda_0 \text{ Pr (fruit} \geqq \text{MTL)} \tag{5-1}$$

We would like to be able to make an estimate of λ_0 using measured values of λ_m. We can go further than (5-1) by noting that any particular fruit reply can come from the mainbeam with probability p_{mb} given by (3-1) or from the sidelobes with probability $1 - p_{mb}$. Therefore,

$$\Pr(\text{fruit} \geqq \text{MTL}) = \Pr(\text{fruit} \geqq \text{MTL} | \text{mainbeam})\, p_{mb} + \Pr(\text{fruit} \geqq \text{MTL} | \text{sidelobe})\,(1 - p_{mb}) \qquad (5\text{-}2)$$

Using the amplitude fruit model discussed in Section III, the minimum signal-to-noise ratio for an omni fruit was 32 dB. Since the mainbeam is defined by the 20 dB beamwidth, then clearly all mainbeam fruits must have SNR's greater than 12 dB. Since MTL for all of the cases studied was set at -85 dBm, corresponding to a 10 dB SNR, it is clear that all mainbeam fruits will exceed MTL, hence,

$$\Pr(\text{fruit} \geqq \text{MTL mainbeam}) = 1 \qquad (5\text{-}3)$$

Combining these results we see that the omni fruit rate is given by

$$\lambda_0 = \lambda_m / [p_{mb} + \Pr(\text{fruit} \geqq \text{MTL} | \text{sidelobe})\,(1 - p_{mb})] \qquad (5\text{-}4)$$

Unfortunately, a program has never been conducted that provides measurements of either p_{mb} or Pr(fruit ≧ MTL sidelobe) and to go further we shall simply have to make estimates of these quantities. To do this we shall make use of the simulation program and the 4^o, -26 dB Taylor illuminated antenna pattern discussed previously. We have already used this special case to relate the probability that any fruit originated within the antenna mainbeam to the peaking factor and found, for example, that for a 4:1 peaking factor, that $p_{mb} = 0.1$. Furthermore, we obtained an empirical distribution for the sidelobe SNR which was shown in Figure 3.2 for MTL set at a 10 dB SNR. It was shown that the area under the tail above MTL was .15 which gives the probability that any fruit arises from the sidelobes has an amplitude that exceeds MTL. Therefore

$$\Pr(\text{fruit} \geqq \text{MTL} | \text{sidelobe}) = .15 \qquad (5\text{-}5)$$

Using these results in (5-4) we estimate the omni fruit rate as

$$\lambda_0 = 4.26\ \lambda_m \qquad (5\text{-}6)$$

Therefore, for a measured fruit rate of 10,000 fruit/sec which corresponds to a 1980 projection of a reasonably high density area, the omni fruit rate is of the order of 40,000 fruits/sec. It should be noted that this is only an estimate since it strongly depends on the actual mainbeam peaking factor and the sidelobes of the antenna pattern that was used in making the measurements. Neither of these parameters are known with any certainty.

Number of Overlapping Omni Fruits

Once the omni fruit rate is known, it is possible to estimate the total number of ATCRBS replies that will likely overlap any one DABS reply. Assuming the ATCRBS arrival times are Poisson distributed, which is not a particularly accurate assumption either, then the probability that k ATCRBS replies overlap a DABS message of length T μsec is

$$P(k) = \frac{[\lambda_0 (T + 20.3)]}{k} \exp[-\lambda_0 (T + 20.3)] \qquad k = 0, 1, 2, \ldots \quad (5\text{=}7)$$

where the omni fruit rate is now measured in units of megafruit per second. The average number of overlapping fruits is therefore $\lambda_0 (T + 20.3)$. For a DABS short message, $T = 64$ μsec, we find that the average number of overlapping ATCRBS replies is $\lambda_0 (T + 20.3) = 3.4$. In Table 5.1 we have tabulated the probability of k overlaps which is just the cumulative Poisson probability for this average value.

k	p(k)	$\sum_{j=0}^{k} p(j)$
0	.033	.033
1	.113	.146
2	.193	.339
3	.200	.539
4	.186	.725
5	.126	.851
6	.072	.923
7	.035	.958
8	.015	.973
9	.006	.979
10	.002	.981

Table 5.1

From this we see that at the 3-sigma point of the Poisson distribution there are 6 omni fruit replies and the probability that there will be more than 6 is less than .08.

In order to obtain an independent check on this number we approach the problem from another point of view. It has been estimated from aircraft density predictions for the region around New York City in 1980 [7] that there will be a maximum of 800 aircraft within line of sight of a DABS sensor at Philadelphia in 1980. It has also been estimated from current measurements of that environment [8] and under the assumption that by 1980 all ATCRBS interrogators will be SLS equipped, that there will be, on the average, 100

ATCRBS fruit replies per second per aircraft. Therefore, the peak omni fruit rate must be less than 80,000 fruits/sec. This in turn leads to the result that for a DABS short message there may be at most 80,000 x (64 + 20.3) x 10^{-6} = 6.7 ATCRBS fruits overlapping the DABS reply. It should be noted that this is a conservative result, but is consistent with the 3-sigma estimate of 6 omni fruit replies obtained from the Poisson model.

Global Performance

It is tempting at this point to analyze the sensor performance for only the worst case of 6 omni fruits, since, presumably the performance should improve as fewer and fewer fruit overlaps occur. It is not clear that this is a good assumption, however, because the effects of a single fruit reply can be quite different from the effects of numerous fruit overlaps, since in the latter case the fruit takes on noise-like qualities. Therefore, if there are K potential fruit overlaps, we shall study the performance for each case of k omni fruit overlaps where k = 0, 1, 2, . . . K and then combine the results using the Poisson weighting to obtain the overall global performance of the sensor.

We have already noted that system performance depends dramatically on whether or not the omni fruit actually enters the sensor via the mainbeam or through the sidelobes. Therefore, it is necessary to subdivide the number of cases studied even further, since if there are k omni fruits, j of them may be mainbeam where j = 0, 1, 2, . . . k. The probability that any one omni fruit comes through the mainbeam is given by (3-1) as

$$p_{mb} = \frac{\eta\theta_{20}}{360 + \theta_{20}(\eta - 1)} \qquad (5\text{-}8)$$

and since the fruit azimuths are independent, then the number of mainbeam fruits out of k omni fruits, j, has a binomial distribution. Therefore,

$$P(j|k) = \binom{k}{j} (p_{mb})^j (1 - p_{mb})^{k-j} \qquad j = 0, 1, \ldots k \qquad (5\text{-}9)$$

Then to evaluate the global performance of some statistic, μ say, we operate the simulation program for the case of j mainbeam fruits out of k omni fruits and obtain the sample value $\mu(j,k)$. We do this for all possible values of $k = 0, 1, \ldots K$ and for each of these, all values of $j = 0, 1, \ldots k$. The global value of the statistic is then given by

$$\overline{\mu} = \sum_{k=0}^{K} \sum_{j=0}^{k} \mu(j,k)\ P(j|k)\ P(k) \tag{5-10}$$

where $P(j|k)$ is the probability that of k omni fruit overlaps, j of them will be within the mainbeam and is given by the binomial distribution in (5-9) and where $P(k)$ is the probability that there will be k omni fruit overlaps and this is given by the Poisson distribution in (5-7).

To perform a complete evaluation requires that there be $(K+1)(K+2)/2$ simulation trials, which in the case of $K = 6$, necessitates 28 computer runs. This is far too many cases to be evaluated in practice, but reasonable bounds on performance can be obtained by considering only the most likely cases. For example, for the 4° antenna pattern with -26 dB sidelobes, a 4:1 peaking factor leads to the probability of a mainbeam fruit of $p_{mb} = .1$. We have used this number in the binomial distribution (5-9) to compute the probability that of k omni fruits, j of them are mainbeam. These probabilities are tabulated in Table 5.2 from which it can be concluded that there will be no need to consider all possible cases. In fact, even for the case of 6 omni fruits, we need only study the results for 0, 1 or 2 mainbeam fruits since the probability of any other situation occurring is negligible.

j \ k	1	2	3	4	5	6	7	8
0	.91	.828	.754	.686	.624	.568	.517	.470
1	.09	.164	.224	.271	.309	.337	.358	.372
2		.008	.022	.040	.061	.083	.106	.129
3			0	.003	.006	.008	.017	.025
4				0	0	0	.002	.004
5					0	0	0	0
6						0	0	0
7							0	0
8								0

Table 5.2

Probability j fruits of k omni fruits are mainbeam

$$P(j|k) = \binom{k}{j} p_{mb}^{j} (1 - p_{mb})^{k-j} \qquad j = 0, 1, \ldots k$$

VI. NUMERICAL RESULTS

The simulation program has been used to generate a set of data which is intended to characterize the performance of a DABS reply processor operating in the ATCRBS fruit environment which will be observed by a sensor located in the vicinity of NAFEC in 1980. It should be noted that these data provide a measure of the performance of a particular, rather sophisticated reply processor configuration operating in that environment. The class of reply processor considered utilized amplitude and azimuth confidence windows derived from the preamble correlation algorithm in addition to the RSLS interference flag.

The data were generated for two ATCRBS fruit conditions. Table 5-1 shows that if the omni fruit rate is 40,000 fruits/sec the average number of ATCRBS fruit expected to overlap a DABS 56-bit downlink message is about 3 if a Poisson arrival model is assumed. In addition, the 3-sigma point on the Poisson distribution is 6 replies. This latter number also corresponds to a peak omni fruit rate of 80,000 fruits/sec which corresponds to an environment of 800 aircraft, each producing an average of 100 replies/sec. This represents a 1980 projection for a reasonably high density area. Therefore, three fruit replies overlapping the DABS message represents an average environment while six replies can be considered a worst case for the above model.

A series of data sets were generated for three and six ATCRBS replies randomly overlapping a 56-bit DABS message. These data were generated using an antenna with a four degree 3 dB beamwidth and 26 dB sidelobes. The DABS message content was randomized while the ATCRBS replies had all code pulses present. The receiver used a first order Butterworth filter and the amplitude and monopulse azimuth outputs were quantized to 256 levels. The SNR for the DABS replies were set at 30 dB (approximately the average SNR for a DABS target) and 15 dB (the minimum usable SNR for a DABS target).

At each SNR three data sets were generated for three and six ATCRBS fruits (a total of 12 cases). In the three cases, the ATCRBS fruits were distributed in the following ways:

a. all ATCRBS fruits in the sidelobes and backlobes

b. one fruit reply in the mainbeam and all others in the sidelobes and backlobes, and

c. two fruit replies in the mainbeam and the others in the sidelobes and backlobes.

These data sets provided an input for the reply processor simulation and the results were combined according to the binomial distribution shown in Table 5-1 for azimuthal peaking factors equal to two and four.

The reply processor configuration used to analyze these data assumed successful preamble detection and produced estimates of amplitude and monopulse azimuth from the preamble according to the algorithm, described in Section IV. Correlation was assumed if the amplitude samples agreed within ± 2 dB and if the azimuth samples agreed to within ± 0. 5 degrees. The relative gain of the omnidirectional antenna was chosen at 20 dB down from the peak of the primary sum beam antenna. The interference detection channel output (Q) was not used to process the data as this parameter was found to have limited use when amplitude and azimuth consistency checks were employed in the generation of information bit, confidence flag and monopulse azimuth estimates [4].

Table 6-1 presents the results of running the simulation under the above conditions. The failure probability implies that the information bit estimates would not be capable of resulting in a corrected information bit sequence according to the error correction criteria defined in Section IV.

The results show that for a nominal fruit rate expected for a reasonably high density area in 1980, there is a high probability that a DABS target at a nominal SNR will have its downlink reply successfully decoded and an accurate monopulse azimuth estimate made.

Number of ATCRBS Replies Overlapping DABS Reply	DABS SNR (dB)	Azimuthal Peaking Factor = 2		Azimuthal Peaking Factor = 4	
		P_F	Azimuth rms (degrees)	P_F	Azimuth rms (degrees)
3	30	.03	.01	.03	.01
	15	.17	.14	.21	.15
6	30	.07	.01	.08	.02
	15	.31	.16	.35	.17

Table 6-1

Typical Simulation Results

VII. CONCLUSIONS

This report has presented a description of a simulation program which allows the evaluation of the design parameters associated with the antenna, receiver and DABS reply processing functions of a DABS sensor. The following conclusions are based on the information presented in this report and the knowledge gained during the detailed analyses of particular interference conditions and failure modes of the reply processor:

1. The simulation program is an effective tool which can be employed to produce realistic interference conditions for sensor performance analyses.
2. The program has the capacity for the analysis of sensor design parameter tradeoffs as the selections of antenna and receiver parameters allow for considerable flexibility.
3. The reply processing algorithms also allow a great deal of flexibility in parameter selection and represent a valuable tool for performance tradeoffs of processor complexity versus interference environment. Although the only algorithm evaluated in this report utilized amplitude and azimuth consistency checks, other simpler algorithms can be analyzed simply by revising the software in the reply processor subroutine.
4. The results presented in Section VI show the performance of a DABS reply processor configuration in a conservative estimate of the NAFEC 1980 interference environment. These results indicate good performance is attainable in the generation of DABS downlink message and monopulse azimuth estimates.

In summary, the simulation program has such a great degree of parameter selection and flexibility that a large number of tradeoff studies can be envisioned in the analysis of the performance of a DABS sensor. In fact, the program has been a valuable tool in the selection of DABS reply processing techniques for the DABS Sensor Engineering Requirement.

ACKNOWLEDGEMENT

The authors would like to thank J. R. Johnson of the M.I.T. Lincoln Laboratory for his assistance in developing the ATCRBS fruit models and the reply processing algorithm.

REFERENCES

[1] "Provisional Signal Formats for the Discrete Address Beacon System"Project Report ATC-30, Lincoln Laboratory, M.I.T. (9 November 1973), DDC AD-770052/9.

[2] D. K. Barton and H. R. Ward, Handbook of Radar Measurement, Appendix A (Prentice-Hall, Englewood Cliffs, New Jersey, 1969).

[3] R. J. McAulay and T. P. McGarty, "An Optimum Interference Detector for DABS Monopulse Data Editing," Technical Note 1973-48, Lincoln Laboratory, M.I.T. (26 September 1973), DDC AD-769337/7.

[4] R. J. McAulay, "Suboptimal Detection of Unresolved Radar Targets and Multipath," private communication.

[5] E. M. Hofstetter and D. F. DeLong, Jr., "Detection and Parameter Estimation in an Amplitude-Comparison Monopulse Radar, "IEEE Trans. Inform. Theory V. IT-15, 22 (1969), DDC AD-689885.

[6] T. J. Macdonald and J. R. Johnson, "DABS Downlink Preamble Study," private communication.

[7] R. L. Honigman, "Aircraft Spatial Distribution, New York, 1980," private communication.

[8] "Development of a Discrete Address Beacon System," Quarterly Technical Summary, Lincoln Laboratory, M.I.T. (1 April 1973), Fig. II-3, DDC AD-762071.