(NASA-CR-158714) OPTIMIZATION OF MLS RECEIVERS FOR MULTIPATH ENVIRONMENTS Final Report (Virginia Univ.) 303 p HC A14/MF A01 CSCL 17G
N79-24973
Unclas
G3/04 23468

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

UNIVERSITY OF VIRGINIA

Charlottesville, Virginia 22901

A Final Report

OPTIMIZATION OF MLS RECEIVERS FOR MULTIPATH ENVIRONMENTS

NASA Grant NSG 1128

Submitted to:

NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport

Submitted by:

G. A. McAlpine
Associate Professor

J. H. Highfill III
Senior Scientist

Report No. UVA/528062/EE79/107

June 1979

303

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

Members of the faculty who teach at the undergraduate and graduate levels and a number of professional engineers and scientists whose primary activity is research generate and conduct the investigations that make up the school's research program. The School of Engineering and Applied Science of the University of Virginia believes that research goes hand in hand with teaching. Early in the development of its graduate training program, the School recognized that men and women engaged in research should be as free as possible of the administrative duties involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

The director of RLES—himself a faculty member and researcher—maintains familiarity with the support requirements of the research under way. He is aided by an Academic Advisory Committee made up of a faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research program.

In addition to administrative support, RLES is charged with providing certain technical assistance. Because it is not practical for each department to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.

A Final Report

OPTIMIZATION OF MLS RECEIVERS FOR MULTIPATH ENVIRONMENTS

NASA Grant NSG 1128

Submitted to:

NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport
Baltimore, MD 21240

Submitted by:

G. A. McAlpine
Associate Professor

J. H. Highfill III
Senior Scientist

Department of Electrical Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528062/EE79/107

June 1979

Copy No. 1

ACKNOWLEDGEMENT

The authors would like to acknowledge the contributions of the following technical assistants who served on the project at various times and without whose collective effort, little would have been accomplished.

R. A. Hale
S. M. Harper
K. C. Henderson
T. C. Hu
S. H. Irwin, Jr.
G. Koleyni
R. E. Nelson
R. C. Newman
J. E. Padgett
S. V. Pizzi
R. D. Steed
J. R. Troxel
C. P. Tzeng
P. D. Via

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF TABLES

SECTION I

INTRODUCTION

This is the final report of research under Grant NSG 1128, awarded to the University in December 1974. The project is concerned principally with the angle-tracking problems in Microwave Landing System (MLS) receivers, the goal of the research being a receiver design capable of optimal performance in the multipath environments found in air terminal areas. The scope of the work included various theoretical and evaluative studies associated with the project goal, e.g.

i. Signal model development

ii. Derivation of optimal receiver structures

iii. Development and use of computer simulations for receiver algorithm evaluation

and also, at least initially, the development of an experimental receiver for flight testing. Reference is made to the progress reports [1-5] for details of the research. This report provides an overview of the work and a summary of principal results and conclusions.

During the 1976-77 annum a study of the DME was also undertaken along with the ongoing angle receiver research. The DME work included some preliminary theoretical analysis but mostly simulation studies of the various designs, i.e. fixed threshold, adaptive threshold and delay and compare DME receivers. The simulations showed the delay and compare receiver to be the most robust of the three designs simulated. The DME study was peripheral to the focal research on the angle receiver and is not discussed further in this report. Reference is made to [4, pp. 51-100] for more detail.

The early study established that multipath propagation in air terminal areas is due principally to reflections from hangars and other buildings, other aircraft and the ground -- all cases where the multipath interference is very nearly specular in nature and either an actual or virtual specular point can be identified. A math model for the received signal in such a case contains parameters which, together with selected time derivatives, constitute a vector which is easily conceptualized into the "state" of a state space model of the problem dynamics. And the latter provides the mathematical framework necessary for applying recursive state estimation techniques to the optimal receiver design problem. This was the approach taken in the research.

Two algorithms resulting were studied extensively:

i. An Optimal receiver: The state vector included signal-to-noise ratios, angular coordinates, angular rates, the phase difference and scalloping frequency parameters.

ii. A Suboptimal receiver: A structure similar in some respects to the optimal receiver but the phase difference and scalloping frequency parameters were not estimated.

Both algorithms generally outperformed in simulation a "threshold receiver" design which was approximately representative of the current Phase III Receiver. The Optimal design generally was the best, at least when the scalloping rate was less than half the angle function repetition rate and hence the phase difference parameter could be tracked successfully. The suboptimal design, at least in the simulation, suffered from extremely complex and length computations. The optimal design (and presumably the suboptimal design, would also) exhibited a high sensitivity to error in the (presumed) beam width of the MLS ground antenna. These and other simulation results are discussed at length in the report, and approaches are suggested for possible improvements in performance.

Some considerations relating to an experimental receiver design are also offered.

SECTION II

SIGNAL MODEL AND ESTIMATOR DESIGNS

The signal available to the processor is a log video signal with bandwidth approximately half that of the i-f channel of the receiver. We begin at this point with the adaptation of this signal, in part A below, for use by the processor. Part B presents a state space model of the evolving problem geometry. Part C describes the general approach used to optimal receiver design, and Part D develops specific designs studied in the research. The reader is assumed to be familiar with the MLS system concepts and signal formats; for this orientation, refer to [6], [7] and [8].

A. THE AMPLITUDE-SQUARED ENVELOPE SIGNAL

The receiver log-envelope signal, a continuous-time signal within a scan, is sampled throughout a window on each semi-scan (centered on the expected centroid of the direct path pulse) at a sampling rate approximately equal to the i-f bandwidth and then suitably exponentiated and squared; the resulting J samples of the amplitude-squared envelope taken within a given scan are then normalized to a suitable measure of receiver noise power and assembled into an observations, or measurement, vector u, which clearly is nonlinear in the problem parameters of interest and also corrupted non-additively by receiver noise. Specifically, for the kth scan, k = 0,1,2,---, and in terms of a discrete-time variable τ_j local to the scan, and assuming the presence of a direct-path component, a single multipath component and receiver noise, the jth component of u, say u_j, j = 1, ---, J is modeled as follows:

$$u_j = \{\alpha p[\theta - \theta_A(\tau_j)] + \alpha_R p[\theta_R - \theta_A(\tau_j)]\cos\beta_j + n_{c_j}\}^2$$

$$+ \{\alpha_R p\ [\theta_R - \theta_A(\tau_j)]\ \sin\beta_j + n_{s_j}\}^2 \qquad (2.1)$$

where

$\alpha = \alpha(k)$ = direct path signal-to-noise ratio (2.2)

$\theta = \theta(k)$ = angular coordinate of own A/C (2.3)

$\alpha_R = \alpha_R(k)$ = multipath signal-to-noise ratio (2.4)

$\theta_R = \theta_R(k)$ = angular coordinate of reflector specular point (2.5)

$\beta_j = \beta(k,\tau_j)$ = direct path-to-multipath phase difference at the jth sample of the kth scan, propagating as follows:

$$\beta_j = \beta(k) + \omega_{sc}\tau_j + \frac{\dot{\omega}_{sc}}{2}\ \tau_j^2 + --- \qquad (2.6a)$$

where

$\beta(k)$ = phase difference at the start of the kth scan, propagating similarly, i.e.

$$\beta(k+1) = \beta(k) + \omega_{sc}\ T_k + \frac{\dot{\omega}_{sc}}{2}\ T_k^2 + --- \qquad (2.6b)$$

in which

T_k = time interval, kth-to-(k+1)th scans. (2.7)

ω_{sc} = the scalloping rate (also derivatives may be present) (2.8)

θ_A (.) = the transmitting antenna scanning function (2.9)
(see equation (2.124) below)

p[.] = the transmitting antenna selectivity function (2.10)
(square root of power density at a constant radius) and

n_{c_j}, n_{s_j} are independent Gaussian random variables with mean zero, variance 0.5. (2.11)

This model neglects second-order effects ([3], p. 9) chiefly

a. Doppler effects not influencing the scalloping rate

b. Differential propagation delays (e.g. $\left(\frac{r_{reflect} - r_{direct}}{c}\right)$, which tend to distort the mirror symmetry that exists between the TO-, FRO-scan signals.

B. THE STATE-SPACE MODEL

The various parameters appearing in u_j, (2.1) above, together with time derivatives of interest, are assembled into an N_s-dimensional state vector, x, modeled as the solution of a suitable linear difference equation evolving in discrete-time, from scan-to-scan, and excited by a white, zero-mean random process, representing external influences. Hence, the composite generator-of-observations model is taken as a state space formulation with form as follows:

$$x(k+1) = F(k)x(k) + G(k)z(k)$$

$$u(k) = h(x(k),n(k)) \tag{2.12}$$

where

$x(k)$ = state N_s-vector (2.13)

$z(k)$ = a representation of the various external influences on the modeled environment, taken as an M-vector random process, white with mean zero and covariance matrix $Q(k)$, where (2.14)

$$Q(k) \triangleq \langle z(k)z^T(k)\rangle \tag{2.15}$$

$F(k)$ = state 1-step transition matrix (2.16)

$G(k)$ = an $N_s \times M$ input constraint matrix of rank M, where

$$M \leq N_s \tag{2.17}$$

$u(k)$ = observations J-vector (2.18)

$n(k)$ = a 2J-vector of the quadrature components n_{c_j}, n_{s_j} of receiver noise associated with the J samples of the noisy envelope ($n(j)$ and $n(k)$ are independent if $j \neq k$). (2.19)

$h(\cdot,\cdot)$ = a nonlinear vector-valued function of its arguments, constructing $u(k)$ as a J-vector of envelope samples u_j, equation (2.1). (2.20)

C. THE ESTIMATION APPROACH

In developing a procedure for estimating the angular coordinate θ, it was clear from (2.1) that performance would be better if more of the state variables than just θ were estimated. It was also clear that not all of the state variables influence the vector $u(k)$ of observation's taken within a single scan; in the present formulation, for example, θ and θ_R are considered constants during the active scan, hence $\dot{\theta}$ and $\dot{\theta}_R$ are absent from (2.1), nevertheless they are necessarily included as variables in the state vector to afford modeling tractably a varying geometry over a sequence of scans. The functional dependence of $u(k)$ on only a subset of the state variables posed a choice:

1. Estimate only the subset of state variables associated with $u(k)$.

2. Estimate the full state vector $x(k)$.

The first option very likely would result in severe degeneration in performance, during signal fades, including possibly loss of lock. The benefits promised by a modest filter memory, however, (represented by rate estimates in a receiver state estimator, e.g.) prompted the second choice, and we began to consider the criterion under which a suitable estimate of the full state x(k) could be calculated recursively, given the complexity of the observations.

Ideally, given a sequence of observations u, (2.12b), say from some initial scan through the present (kth) scan, represented as follows:

$$U(k) \triangleq \{u(1), u(2), ---, u(k)\}, \quad (2.21)$$

we might adopt as a candidate criterion of performance the mean square error criterion and seek to calculate the conditional mean $\langle x(k) | U(k) \rangle$ as a basis for an optimal MLS receiver design. Without loss of generality, however, the conditional mean can be written as

$$\langle x(k) | U(k) \rangle = \langle x(k) | U(k-1) \rangle + \langle E(k) | U(k) \rangle \quad (2.22)$$

where

$$\langle x(k) | U(k-1) \rangle = F(k+1) \langle x(k-1) | U(k-1) \rangle \quad (2.23)$$

is the extrapolation of the prior estimate to the present, and $\langle E(k) | U(k) \rangle$ is the conditional mean of the error

$$E(k) \triangleq x(k) - \langle x(k) | U(k-1) \rangle \quad (2.24)$$

in the extrapolated estimate (2.23). Hence, via (2.22), a conditional mean receiver is essentially equivalent to a calculation of $\langle E(k) | U(k) \rangle$, a task which, generally,

1. As the notation suggests, may involve individually each and every observation constituting the sequence U(k), and in addition,
2. May not even be computable in a finite number of operations, or even easily approximated.

A class of notable exceptions exists in which not only is $\langle E(k) | U(k) \rangle$ computable or easily approximated, but all the historical information in dated observations necessary for the calculation of $\langle E(k) | U(k) \rangle$ is carried in the extrapolated prior state estimate $\langle x(k) | U(k-1) \rangle$, (2.23); that is $\langle E(k) | U(k) \rangle$ can be written

$$\langle E(k) | U(k) \rangle = f(\langle x(k) | U(k-1) \rangle, u(k)), \text{ for some } f(\cdot,\cdot). \tag{2.25}$$

Two examples are the following:

1. The Kalman filter, applicable when $h(x,n)$ is linear in both x and n, gives an exact calculation of $(\langle E(k) | U(k) \rangle$;

2. The Extended Kalman filter, applicable when $h(x,n)$ represents an additive corruption of the observations by n, gives an approximat calculation of $\langle E(k) | U(k) \rangle$ when $h(x,n)$ is nonlinear in x.

Because of the complexity of the observation $h(x,n)$ [recall h_j, (2.1)] and the computability requirement (for simulation and also, potentially, hardware), the estimation approach used in this research was, of necessity, an approximation. An adaptation of the preceeding was used, producing a state estimate (generally now suboptimal), denoted $\hat{x}(k|k)$ to distinguish it from the exact conditional mean, and obtained, as follows:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \hat{\xi}(k|k) \tag{2.26}$$

where

$$\hat{x}(k|k-1) = F(k-1)\hat{x}(k-1|k-1) \tag{2.27}$$

and $\hat{\xi}(k|k)$ is a "suitable" estimate to be defined below, of the error $\xi(k)$ in $\hat{x}(k|k-1)$, given $\hat{x}(k|k-1)$ and $u(k)$, where

$$\xi(k) \triangleq x(k) - \hat{x}(k|k-1) \tag{2.28}$$

The estimate $\hat{\xi}(k|k)$ will be functionally dependent only upon $\hat{x}(k|k-1)$ and $u(k)$. A final error definition needed and an easily proved result of interest are, as follows:

$$e(k) \triangleq \xi(k) - \hat{\xi}(k|k) \tag{2.29}$$

$$= x(k) - \hat{x}(k|k) \tag{2.30}$$

i.e., the error in the estimate $\hat{\xi}(k|k)$ is the residual error in the updated state estimate $\hat{x}(k|k)$.

Since the observations u(k) are functionally dependent upon only a subset of the state variables, the calculation of $\hat{\xi}(k|k)$, i.e. the estimation of the error $\xi(k)$ in $\hat{x}(k|k-1)$, given $\hat{x}(k|k-1)$ and u(k), was accomplished in 2 stages, characterized respectively as the Scan Data Processor (SDP) and the Tracking Loop Filter. The SDP essentially does a curve-fitting of a noiseless, internal version of the observations with the noisy, actual ones, calculating perturbations (error estimates) of the associated elements of $\hat{x}(k|k-1)$ to improve the fit. The Tracking Loop, closed around the SDP in a conventional recursive structure, develops on estimate of the full error vector $\xi(k)$, taking the assumed state evolution dynamics into account. A detailed discussion of these two stages is given below.

The approach taken was modified, in part, by two factors, as follows:

1. The presumed low-bandwidth of the state evolution model wrt the repetition rate, implying, quantitatively,

$$x(k) \simeq F(k-1)x(k-1) \tag{2.31}$$

(i.e. $G(k-1)\omega(k-1)$ in (2.12) is small).

2. The "tracking" nature of the estimation task implying, presumably,

$$x(k) \simeq \hat{x}(k|k) \tag{2.32}$$

(i.e. the estimation error e(k), (2.31), is "small").

Equations (2.27), (2.28), (2.30) and (2.31) above imply that $\xi(k)$ may be approximated, as follows:

$$\xi(k) \cong F(k-1)e(k-1) \tag{2.33}$$

and this, with (2.32) above implies that

$$\xi(k) \text{ is "small"}, \tag{2.34}$$

a result important to the design of the SDP described next.

Scan Data Processor

Let γ denote the parameter vector comprising the subset of N_G state variables on which u_j, (2.1), is functionally dependent. The general relation

$$\gamma = Hx \tag{2.35}$$

then defines a masking matrix H, $N_G \times N_s$, having rank $N_G \leq N_s$ and consisting appropriately of 1's and 0's. Other N_G-vector quantities of interest are obtained, as follows:

Extrapolated Prior Est: $\hat{\gamma}(k|k-1) = H\hat{x}(k|k-1)$ (2.36)

Error in $\hat{\gamma}(k|k-1)$: $\varepsilon(k) = H\xi(k)$ (2.37)

By (2.34), $\varepsilon(k)$ in (2.37) is "small", and Murphy's Locally Optimum Estimation (LOE) theory, [9], was brought to bear on the calculation of an estimate which, around $\varepsilon=0$, should be optimal in an intuitively appealing sense. The LOE criterion is summarized, in the notation of the SDP, as follows:

> Locally Optimum Estimation: The estimate $\hat{\varepsilon}$ of the error ε (in the present estimate $\hat{\gamma}$ of the parameter γ) is locally optimum at the point $\varepsilon=0$, if and only if (iff) the following two conditions are satisfied:
>
> 1) $\hat{\varepsilon}$ is a locally unbiased estimate of ε at the point of $\varepsilon=0$, and

2) $\hat{\varepsilon}$ is a locally minimum mean-squared error (MMSE) estimate of ε at the point $\varepsilon=0$,

where 'locally unbiased' and 'locally MMSE' estimations are defined as follows:

Locally Unbiased Estimation: Defining the error in the estimate $\hat{\varepsilon}$ of the quantity ε as follows:

$$\eta(k) \triangleq \hat{\varepsilon}(k|k) - \varepsilon(k) \tag{2.38}$$

and then defining the bias of the estimate $\hat{\varepsilon}$ of the error ε (in the estimate $\hat{\gamma}$ of the parameter γ), as follows:

$$b(\varepsilon) \triangleq \langle\eta(k)|\gamma - \hat{\gamma} = \varepsilon\rangle \tag{2.39}$$

then the estimate $\hat{\varepsilon}$ of the error ε (in the estimate $\hat{\gamma}$ of the parameter γ) is locally unbiased at the point $\varepsilon=0$ iff the following two conditions are satisfied:

1) $b(0) = 0$, on N_G-vector (2.40a)

2) $\left[\frac{db(\varepsilon)}{d\varepsilon}\right]\Big|_{\varepsilon=0} = 0$, on $N_G \times N_G$ matrix (2.40b)

and

Locally MMSE Estimation: Defining the mean-squared error of $\hat{\varepsilon}$ in terms of η, (2.38) above, as follows:

$$\Sigma_{\hat{\varepsilon}}(\varepsilon) \triangleq \langle\eta(k)\eta^T(k) \,|\, \gamma - \hat{\gamma} = \varepsilon\rangle,\ (\ )^T = \text{transpose}, \tag{2.41}$$

then the estimate $\hat{\varepsilon}$ of the error ε (in the estimate $\hat{\gamma}$ of the parameter γ) is locally MMSE at the point $\varepsilon=0$ iff, for any estimate, $\hat{\delta}$, if ε locally unbiased at $\varepsilon=0$, the mean-squared errors of $\hat{\varepsilon}$ and $\hat{\delta}$ satisfy, in the usual non-negative definite sense,

$$\Sigma_{\hat{\delta}}(0) \geq \Sigma_{\hat{\varepsilon}}(0), \quad N_G \times N_G \text{ matrices} \tag{2.42}$$

The error $\eta(k)$ is induced by the noise $n(k)$ which is white (recall (2.11) and (2.19)); hence, clearly, local to the point $\varepsilon=0$, when $\langle n \rangle = 0$ via (2.39), it is true also that $\eta(k)$ is white, i.e.

$$\langle \eta(j)\eta^T(k) \rangle = 0, \quad j \neq k \tag{2.43}$$

We take note, in passing, of the important and beneficial property given in (2.40b), requiring that errors made in estimating the various components of the vector ε be decoupled when $\varepsilon=0$. In addition to making the estimate unique, this is probably effective in extending the properties in (2.40a) and (2.42) into the open region around the point $\varepsilon=0$.

Murphy has meticulously expounded the theory and solution of the locally optimum estimation problem in his scholarly work [9] and illustrated his results in diverse examples in communications. The solution, applied to the SDP design problem at hand, involves, first, the definition of several additional quantities:

1. The noiseless quadratic envelope vector q with element q_j.
2. The linear envelope vectors m and v (and associated elements), corresponding respectively to quadratic envelopes q and u.
3. The conditional probability density function (pdf) $p(v|m)$
4. The likelihood ration $\lambda(u|\gamma)$ Let m_{c_j} and m_{g_j} respectively be the linear envelope functions associated with a cosine and sine orthogonal decomposition of the noiseless i-f (or r-f) signal:

$$m_{c_j} = \alpha p[\theta - \theta_A(\tau_j)] + \alpha_R p[\theta_R - \theta_A(\tau_j)]\cos\beta_j \tag{2.44a}$$

$$m_{s_j} = \alpha_R p[\theta - \theta_A(\tau_j]\sin\beta_j \tag{2.44b}$$

various parameters of which are as defined, following (2.1). Then, in the same manner that the J-vector u of observations u_j was constructed, a noiseless quadratic envelope vector q is defined with elements q_j, j=1,2,...,J, where

$$q_j = m_{c_j}^2 + m_{s_j}^2 \tag{2.45}$$

$$= \alpha^2 p_j^2(\theta) + 2\alpha\alpha_R p_j(\theta) p_j(\theta_R) \cos\beta_j + \alpha_R^2 p_j^2(\theta_R) \tag{2.46}$$

in which $p_j(\theta)$ is short-hand for $p[\theta-\theta_A(\tau_j)]$, and similarly for $p_j(\theta_R)$. The observations sample u_j, (2.1) may then be written as

$$u_j = (m_{c_j} + n_{c_j})^2 + (m_{s_j} + n_{s_j})^2 \tag{2.47}$$

or, equivalently

$$u_j = q_j + 2n_{c_j}[q_j]^{\frac{1}{2}} + n_{c_j}^2 + n_{s_j}^2 \tag{2.48}$$

Now, let m and v respectively represent noiseless and noisey <u>linear</u> envelope vectors with elements m_j and v_j, respectively, for j=1,2,...,J, where

$$m_j \triangleq q_j^{\frac{1}{2}} = [m_{c_j}^2 + m_{s_j}^2]^{\frac{1}{2}} \tag{2.49}$$

$$v_j \triangleq u_j^{\frac{1}{2}} = [(m_{c_j} + n_{c_j})^2 + (m_{s_j} + n_{s_j})^2]^{\frac{1}{2}} \tag{2.50}$$

Since the sampling rate within the scan equals the i-f bandwidth, the noise samples are all nearly independent (and zero mean, Gaussian with variance 0.5; recall (2.11)). Hence, referring to [10, eq. (8-115)] for the conditional pdf $p(v_j|m_j)$, the conditional pdf $p(v|m)$ can be written

$$p(v|m) = \prod_{j=1}^{J} p(v_j|m_j) \tag{2.51}$$

$$= 2^J \prod_{j=1}^{J} v_j I_o(2m_j v_j)\exp(-m_j^2 - v_j^2) \tag{2.52}$$

where $I_o(\cdot)$ is the modified Bessel function of the first kind, zeroth order. The likelihood ratio of interest is the following:

$$\lambda = \frac{p(v|m)}{p(v|0)} = \prod_{j=1}^{J} I_o(2m_j v_j)\exp(-m_j^2) \tag{2.53}$$

or, defining a new function $M_o(\cdot)$: $R^+ \rightarrow R^+$, as follows, in relation to the even function $I_o(\cdot)$ ([3], p. 13):

$$M_o(x^2) = I_o(x), \; x \in R^1 \tag{2.54}$$

then, in terms of q_j and u_j, we may write

$$\lambda(u|\gamma) = \prod_{j=1}^{J} M_o(4q_j u_j)\exp(-q_j) \tag{2.55}$$

$$= \prod_{j=1}^{J} \lambda_j(u_j|q_j) \tag{2.56}$$

where the conditioning variable on the <u>left</u> is shown as γ, rather than $q(=q(\gamma))$, to emphasize the parameter values.

The theory provides, further, that if one of the parameters upon which q is dependent is, in fact, a random variable, say ζ, in which there is no estimation interest, then it is to be averaged out before proceeding, i.e. the average likelihood ratio

$$\lambda(u|\gamma) = \langle \prod_{j=1}^{J} \lambda_j(u_j|q_j(\gamma,\zeta)) \,|u,\gamma\rangle \tag{2.57}$$

is used in the work below. In this approach (which formed the basis of one variant of MLS receiver design studied) clearly the noiseless envelope vector q has no further significance.

The Scan Data Processor design by the LOE approach can now be completed. In the notation of the SDP design problem (but otherwise quite generally) the estimate $\hat{\varepsilon}(k|k)$ of the error $\varepsilon(k)$ (in the estimate

$\hat{\gamma}(k|k-1)$ of the parameter vector γ) which is locally optimum at $\varepsilon=0$ is given by

$$\hat{\varepsilon}(k|k) = \Phi^{-1}(\hat{\gamma})\Lambda(u|\hat{\gamma}) \tag{2.58}$$

where, recognizing $u(k) = u(\gamma(k),n(k))$,

$$\Phi(\hat{\gamma}) \triangleq \langle \Lambda(u(\gamma,n)|\hat{\gamma})\Lambda^T(u(\gamma,n)|\hat{\gamma}) \; \big| \; \gamma = \hat{\gamma} \rangle \tag{2.59}$$

denoted as the LOE (Fisher) Information Matrix, and

$$\Lambda(u|\hat{\gamma}) \triangleq \begin{cases} \frac{\partial}{\partial\hat{\gamma}} \ln \lambda(u|\hat{\gamma}) & \lambda(u|\hat{\gamma}) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.60}$$

Further, the mean-squared error, Σ, (2.41), of this estimate local to the point $\varepsilon=0$ is

$$\Sigma(o) = \Phi^{-1}(\gamma) \tag{2.61}$$

As indicated above, several variants to the basic MLS receiver design using this approach were studied, differing initially in their definitions of the state and/or parameter vectors, x and γ respectively. Detailed development of the LOE quantities defined above is deferred until the next section of this chapter where the various designs specific to particular state and/or parameter vector formulations will be described. We conclude this discussion of the Scan Data Processor by noting that, in view of (2.58) and (2.61) above, the principle calculations done by the SDP are those of $\Phi(\hat{\gamma})$ and $\Lambda(u|\hat{\gamma})$. These, in fact, are the quantities passed to the Tracking Loop Filter, discussed next.

Tracking Loop Filter

Inputs to the Tracking Loop Filter from the Scan Data Processor are the quantities $\Lambda(u|\hat{\gamma}(k|k-1))$ and $\Phi(\hat{\gamma}(k|k-1))$. If we form the estimate

$\hat{\varepsilon}(k|k)$, as prescribed in (2.58), that is:

$$\hat{\varepsilon}(k|k) = R(\hat{\gamma})\Lambda(u|\hat{\gamma}) \tag{2.62}$$

where $\hat{\gamma} = \hat{\gamma}(k|k-1)$ (2.63)

and $R(\hat{\gamma}) \triangleq \Phi^{-1}(\hat{\gamma}(k|k-1))$ (2.64)

and then tentatively form a "pre-estimate", $\hat{\hat{\gamma}}(k|k)$, in the following manner:

$$\hat{\hat{\gamma}}(k|k) = \hat{\gamma}(k|k-1) + \hat{\varepsilon}(k|k), \tag{2.65}$$

we find that $\hat{\hat{\gamma}}$ can be written

$$\hat{\hat{\gamma}} = \hat{\gamma}(k|k-1) + \varepsilon(k) + \hat{\varepsilon}(k|k) - \varepsilon(k) \tag{2.66}$$

$$= \gamma(k) + \eta(k) \tag{2.67}$$

$$= Hx(k) + \eta(k) \tag{2.68}$$

i.e. the pre-estimate $\hat{\hat{\gamma}}(k|k)$, in a neighborhood of $\varepsilon=0$, is in fact, a "pseudo-observation" which is both <u>linear</u> in x and corrupted <u>additively</u> by the zero mean, white noise $\eta(k)$ with covariance $R(\hat{\gamma}(k|k-1))$, (2.63).

Following conventional Kalman filter theory and forming the innovations process, $\hat{\gamma}(k|k-1) - \hat{\hat{\gamma}}(k|k)$, gives

$$\hat{\gamma}(k|k-1) - \hat{\hat{\gamma}}(k|k) = \hat{\varepsilon}(k|k) \tag{2.69}$$

i.e. the innovations process is the estimate $\hat{\varepsilon}(k|k)$, (2.62), produced (effectively) by the LOE-theory-based Scan Data Processor. The filter state update equation has the form

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\hat{\varepsilon}(k|k) \tag{2.70}$$

where $K(k)$, the Kalman gain, is calculated by cycling through 3 equations for each value of k, k = 1, 2,..., usually as follows:

Extrapolated Error Covariance:

$$P(k|k-1) = F(k-1)P(k-1|k-1)F^T(k-1) + G(k-1)Q(k-1)G^T(k-1) \tag{2.71}$$

Kalman Gain:

$$\kappa(k) = P(k|k-1)H^T[HP(k|k-1)H^T + R(\hat{\gamma})]^{-1} \qquad (2.72)$$

Updated Error Covariance:

$$P(k|k) = (I-\kappa(k)H)P(k|k-1)(I-\kappa(k)H)^T + \kappa(k)R\hat{\gamma})\kappa^T(k) \qquad (2.73)$$

In the present application some simplification is possible, however. Comparing (2.26) and (2.70) above indicates that

$$\xi(k|k) = \kappa(k)\hat{\varepsilon}(k|k) \qquad (2.74)$$

$$= \kappa(k)R(\hat{\gamma})\Lambda(u|\hat{\gamma}) \qquad (2.75)$$

and substituting from (2.72) into the latter gives

$$\xi(k|k) = P(k|k-1)H^T[HP(k|k-1)H^T + R(\hat{\gamma})]^{-1}R(\hat{\gamma})\Lambda(u|\hat{\gamma}) \qquad (2.76)$$

or, after simplifying,

$$\xi(k|k) = \Gamma(k)\Lambda(u|\hat{\gamma}) \qquad (2.77)$$

where

$$\Gamma(k) = P(k|k-1)H^T[I + \Phi(\hat{\gamma})HP(k|k-1)H^T]^{-1} \qquad (2.78)$$

is a new $N_S \times N_G$ gain matrix not requiring the inversion of the matrix $\Phi(\hat{\gamma})$ (produced by the SDP) for its calculation (by (2.64), $R^{-1}(\hat{\gamma})$ appearing in the simplification, was replaced by $\Phi(\hat{\gamma})$). The refined state-estimate update equation, corresponding to (2.70) is the following:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \Gamma(k)\Lambda(u|\hat{\gamma}) \qquad (2.79)$$

Comparing (2.75) and (2.77) indicates that

$$\kappa(k)R(\hat{\gamma}) = \Gamma(k) \qquad (2.80)$$

or that

$$\kappa(k) = \Gamma(k)R^{-1}(\hat{\gamma}) = \Gamma(k)\Phi(\hat{\gamma}) \qquad (2.81)$$

Substituting this into (2.73) and simplifying gives the following:

Updated Error Covariance:

$$P(k|k) = (I-\Gamma(k)\Phi(\hat{\gamma})H)P(k|k-1)(I-\Gamma(k)\Phi(\hat{\gamma})H)^T + \Gamma(k)\Phi(\hat{\gamma})\Gamma^T(k) \quad (2.82)$$

which also does not require the inversion of $\Phi(\gamma)$.

In summary, the MLS receiver design developed, a tracking receiver, will operate as a recursive state estimator and begin the (kth) data processing cycle by extrapolating the prior state estimate $\hat{x}(k-1|k-1)$ to the present, producing the

Extrapolated State Estimate, $\hat{x}(k|k-1)$, (2.27)

and then masking it, giving the

Extrapolated Parameter Estimate, $\hat{\gamma}(k|k-1)$, (2.36)

this, an estimate of parameter vector $\gamma(k)$ with error $\varepsilon(k)$. Next, given $\hat{\gamma}(k|k-1)$ and the vector $u(k)$ of observations, the Scan Data Processor, designed under a criterion of producing an estimate $\hat{\varepsilon}(k|k)$ of error $\varepsilon(k)$ that is locally optimum at $\varepsilon=0$, stops short of this result and instead calculates the following:

Log Likelihood Ratio, $\Lambda(u|\hat{\gamma})$, (2.60)

LOE Information Matrix, $\Phi(\hat{\gamma})$, (2.59)

The Tracking Loop Filter accepts Λ and Φ from the SDP and completes the data processing cycle with following sequence of calculations.

Extrapolated Error Covariance, $P(k|k-1)$, (2.71)

Filter Gain, $\Gamma(k)$, (2.78)

Updated State Estimate, $\hat{x}(k|k)$, (2.79)

Updated Error Covariance, $P(k|k)$, (2.82).

This concludes the derivation at the general level. The results are specialized to particular state- and parameter-vector formulations next.

D. SPECIFIC DESIGNS

Throughout the research program three specializations of the above general design structure received most of the attention. These are characterized by the formulations of their parameter and state <u>estimate</u> vectors, as follows:

i. "Non-adaptive" Design

$$N_G = 2, \quad \hat{\gamma} = (\hat{\alpha}, \hat{\theta})^T \tag{2.83}$$

$$N_s = 3, \quad \hat{x} = (\hat{\alpha}, \hat{\theta}, \hat{\dot{\theta}})^T \tag{2.84}$$

ii. "Optimal" Design (Adaptive)

$$N_G = 5, \quad \hat{\gamma} = (\hat{\alpha}, \hat{\theta}, \hat{\alpha}_R, \hat{\theta}_R, \hat{\beta})^T \tag{2.85}$$

$$N_s = 8, \quad \hat{x} = (\hat{\alpha}, \hat{\theta}, \hat{\dot{\theta}}, \hat{\alpha}_R, \hat{\theta}_R, \hat{\dot{\theta}}_R, \hat{\beta}, \hat{\omega}_{sc})^T \tag{2.86}$$

iii. "Suboptimal" Design (Adaptive)

$$N_G = 4, \quad \hat{\gamma} = (\hat{\alpha}, \hat{\theta}, \hat{\alpha}_R, \hat{\theta}_R,)^T \tag{2.87}$$

$$N = 6, \quad \hat{x} = (\hat{\alpha}, \hat{\theta}, \hat{\dot{\theta}}, \hat{\alpha} , \hat{\theta} , \hat{\dot{\theta}})^T \tag{2.88}$$

The non-adaptive design was identified and studied as a baseline design in accessing the benefits of adaptivity as supplied by the other designs. The optimal design was the principal focus of the study; the definitions (2.85) and (2.86), and the resulting design, are predicated on the assumption that the multipath interference phenomenon, when present (in q, (2.46), e.g.), is one which is fully described by samples taken at the angle function repetition rate, or more concisely, via the sampling theorem,

$$\omega_{sc} < \pi(\text{Rep. Rate}) = \frac{\pi}{T} \tag{2.89}$$

where T is the interval between scans. Under this assumption then, β on the active scan is nearly constant, i.e. in q, (2.46),

$$\beta_j \cong \beta(k), \quad \text{for all } j = 1, 2, ---, J \tag{2.90}$$

(where $\beta(k)$ is the phase difference at the start of the kth scan) and hence $\hat{\omega}_{sc}$ does not appear in the $\hat{\gamma}$-formulation for this design. The relation (2.89) is a restriction that would not always be met in practice, of course, and the suboptimal design represented an effort to formally relax this condition and simultaneously reduce the dimensions of the vectors and hence the complexity of the algorithm. This design was accomplished by

i. Assuming the β_j in q (2.46) were all independent random variables, uniformly distributed on the interval $(-\pi,\pi)$ (corresponding to the assumption that $\omega_{sc} \to \infty$); then

ii. Following (2.57) and taking the average of the likelihood ratio over all the β_j and then using it in the subsequent design.

Both the state and parameter vectors are devoid of both β and ω_{sc} in this third design.

A fourth and very recently conceived design, motivated also by the desire to relax the restriction (2.89), though more complex than the "Optimal" design, is characterized as follows:

"6D LOE" Design (adaptive)

$$N_G = 6, \quad \hat{\gamma} = (\hat{\alpha}, \hat{\theta}, \hat{\alpha}_R, \hat{\theta}_R, \hat{\beta}, \hat{\omega}_{sc})^T \tag{2.91}$$

and

$$N_s = 8, \quad \hat{x} = (\hat{\alpha}, \hat{\theta}, \hat{\dot{\theta}}, \hat{\alpha}_R, \hat{\theta}_R, \hat{\dot{\theta}}_R, \hat{\beta}, \hat{\omega}_{sc})^T \tag{2.92}$$

or

$$N_s = 9, \quad \hat{x} = (\hat{\alpha}, \hat{\theta}, \hat{\dot{\theta}}, \hat{\alpha}_R, \hat{\theta}_R, \hat{\dot{\theta}}_R, \hat{\beta}, \hat{\omega}_{sc}, \hat{\dot{\omega}}_{sc})^T \tag{2.93}$$

The assumption here is that there is sufficient information in the J samples taken within a single scan (J/2 on each semi-scan, centered on the expected centroid of the direct path pulse) to produce an estimate of ω_{sc} using just one scan's data; it appears also the fractional accuracy of such an estimate would improve with increasing ω_{sc}, though it may be somewhat θ-dependent, since the interval between the TO-scan and FRO-scan pulses is both θ-dependent and relevant to the ω_{sc}-estimate produced (or, more specifically, the LOE estimate of the error in $\hat{\omega}_{sc}(k|k-1)$. The "6D LOE" design will be described in Appendix C with any results obtained at the time of grant closure.

Here we focus on the earlier designs, in particular, the details for the optimal and suboptimal designs. The non-adaptive design is clearly imbedded, in a sense, in both of these designs and needn't be treated separately.

Optimal Design

For this case, referencing (2.56), $\lambda(u|\hat{y})$ can be written as follows:

$$\lambda(u|\hat{y}) = \prod_{j=1}^{J} \lambda_j(u_j|\hat{q}_j) \tag{2.94}$$

where

$$\lambda_j(u_j|\hat{q}_j) \triangleq M_o(4\hat{q}_j u_j)\exp(-\hat{q}_j) \tag{2.95}$$

$$> 0, \text{ for all } u_j \text{ and finite } \hat{q}_j \tag{2.96}$$

and

$$\hat{q}_j \triangleq q_j(\hat{y}) \tag{2.97}$$

$$= \hat{\alpha}^2 p^2_j(\hat{\theta}) + 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta}) p_j(\hat{\theta}_R)\cos\hat{\beta} + \hat{\alpha}_R^2 p_j^2(\hat{\theta}_R) \tag{2.98}$$

Hence, via (2.60)

$$\Lambda(u|\hat{\gamma}) \triangleq \frac{\partial \ln}{\partial \hat{\gamma}} \lambda(u|\hat{\gamma}) = \sum_{j=1}^{J} \Lambda_j(u_j|\hat{q}_j) \tag{2.99}$$

where

$$\Lambda_j(u_j|\hat{q}_j) \triangleq \frac{\partial}{\partial \hat{\gamma}} \ln \lambda_j(u_j|\hat{q}_j) = \frac{\frac{\partial \lambda_j}{\partial \hat{\gamma}}(u_j|\hat{q}_j)}{\lambda_j(u_j|\hat{q}_j)}$$

$$= \left(\frac{d\hat{q}_j}{d\hat{\gamma}}\right) \left(\frac{\frac{\partial \lambda_j}{\partial \hat{q}_j}(u_j|\hat{q}_j)}{\lambda_j(u_j|\hat{q}_j)}\right) \tag{2.100}$$

or

$$\Lambda(u|\hat{\gamma}) = D(\hat{\gamma}) w(u|q(\hat{\gamma})) \tag{2.101}$$

in which

$$D(\hat{\gamma}) \triangleq \frac{d\hat{q}}{d\hat{\gamma}} = \begin{pmatrix} \frac{\partial \hat{q}_1(\hat{\gamma})}{\partial \hat{\gamma}_1} & \text{---------} & \frac{\partial \hat{q}_J(\hat{\gamma})}{\partial \hat{\gamma}_1} \\ \text{---------} & \text{---------} & \text{---------} \\ \frac{\partial \hat{q}_1}{\partial \hat{\gamma}_{N_G}} & \text{---------} & \frac{\partial \hat{q}_J(\hat{\gamma})}{\partial \hat{\gamma}_{N_G}} \end{pmatrix}, \; N_G \times J \tag{2.102}$$

$$\begin{pmatrix} \text{---}\; 2\hat{\alpha} p_j^2(\hat{\theta}) + 2\hat{\alpha}_R p_j(\hat{\theta}) p_j(\hat{\theta}_R) \cos\hat{\beta} \;\text{---} \\ \text{---}\; 2\hat{\alpha}^2 p_j(\hat{\theta}) \dot{p}_j(\hat{\theta}) + 2\hat{\alpha}\hat{\alpha}_R \dot{p}_j(\hat{\theta}) p_j(\hat{\theta}_R) \cos\hat{\beta} \;\text{---} \\ \text{---}\; 2\hat{\alpha} p_j(\hat{\theta}) p_j(\hat{\theta}_R) \cos\hat{\beta} + 2\hat{\alpha}_R p_j^2(\hat{\theta}_R) \;\text{---} \\ \text{---}\; 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta}) \dot{p}_j(\hat{\theta}_R) \cos\hat{\beta} + 2\hat{\alpha}_R^2 \dot{p}_j(\hat{\theta}_R) p_j(\hat{\theta}_R) \;\text{---} \\ \text{---}\; (-\, 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta}) p_j(\hat{\theta}_R) \sin\hat{\beta}) \;\text{---} \end{pmatrix} \tag{2.103}$$

where

$$\dot{p}_j(\theta) \triangleq \frac{d}{d\theta_e} p[\theta_e] \Big|_{\theta_e = \theta - \theta_A(\tau_j)} \quad \text{(and similarly for } \dot{p}_j(\theta_R)), \tag{2.104}$$

and

$$w(u|\hat{\gamma}) \triangleq (---,\ w_j(u_j|\hat{\gamma}),\ ---)^T, \quad \text{J-vector} \tag{2.105}$$

where

$$w_j(u_j|\hat{\gamma}) \triangleq \frac{\frac{\partial\lambda_j}{\partial\hat{q}_j}(u_j|\hat{q}_j)}{\lambda_j(u_j|\hat{q}_j)} \tag{2.106}$$

$$= 4u_j \frac{M_1}{M_o}(4q_j(\hat{\gamma})u_j) - 1 \tag{2.107}$$

in which

$$\frac{M_1}{M_o}(\cdot) \triangleq \frac{M_1(\cdot)}{M_o(\cdot)} \tag{2.108}$$

and, for any real z>0,

$$M_1(z) \triangleq \frac{d}{dz} M_o(z) \tag{2.109}$$

where $M_o(\cdot)$ was as defined in (2.54) in relation to $I_o(\cdot)$, as follows for any real x:

$$M_o(x^2) = I_o(x) \tag{2.110}$$

A corresponding relation for $M_1(\ )$ is as follows:

$$M_1(x^2) = \frac{1}{2x} I_1(x) \tag{2.111}$$

where $I_1(\cdot)$ is the modified Bessel function of the first kind, first order. The well-known soft-limiter characteristic of

$$\frac{I_1}{I_o}(\cdot) \qquad \text{(initial slope of } \tfrac{1}{2}\text{, saturation value of 1)}$$

corresponds to the following for $\frac{M_1}{M_o}(x)$, $x \geq 0$:

$$\left.\frac{M_1}{M_o}(x)\right|_{x=0} = \tfrac{1}{4} \tag{2.112}$$

$$\frac{d}{dx}\frac{M_1}{M_o}(x)\Bigg|_{x=0} = -\frac{1}{32} \tag{2.113}$$

$$\frac{M_1}{M_o}(x)\Bigg|_{x \text{ large}} \to \frac{1}{2\sqrt{x}} \tag{2.114}$$

These conditions are satisfied exactly by the approximation ([3], pp. 15-17)

$$\frac{M_1}{M_o}(x) \approx \frac{1}{2\,(4+x)^{\frac{1}{2}}} \tag{2.115}$$

whose error peaks at only 4% around x=30. Substituting this in (2.107) above for $w_j(u_j|\hat{\gamma})$ gives the expression

$$w_j(u_j|\hat{\gamma}) \approx \frac{u_j}{(1 + q_j(\hat{\gamma})u_j)^{\frac{1}{2}}} - 1 \tag{2.116}$$

which was used in this design.

Substituting (2.101) above in the defining equation (2.59) for the LOE Information Matrix Φ gives

$$\Phi(\hat{\gamma}) \triangleq \langle\Lambda(u(\gamma,n)|\hat{\gamma})\Lambda^T(u(\gamma,n)|\hat{\gamma})|\gamma = \hat{\gamma}\rangle \tag{2.117}$$

$$= D(\hat{\gamma})H_w(q(\hat{\gamma}))D^T(\hat{\gamma}) \tag{2.118}$$

where

$$H_w(q(\hat{\gamma})) \triangleq \langle\omega(u(\gamma,n)|\hat{\gamma})w^T(u(\gamma,n)|\hat{\gamma})|\gamma = \hat{\gamma})\rangle \tag{2.119}$$

The criterion for locally optimum estimation (more specifically, locally unbiased estimation) at $\varepsilon=0$ assures that

$$0 = \langle\Lambda(u(\gamma,n)|\hat{\gamma})|\gamma = \hat{\gamma}\rangle = D(\hat{\gamma})\langle\omega(u(\gamma,n)|\hat{\gamma})|\gamma = \hat{\gamma}\rangle \tag{2.120}$$

A simulation study ([4], p.15) of the process $w(u|\hat{\gamma})$, using the approximation (2.116) gave support for (2.120) as well as strong evidence that $w(u|\hat{q})$ is white, i.e.

$$\langle w_i(u_i(\gamma,n_i)\,|\hat{\gamma})w_j(u_j(\gamma,n_j)\,|\hat{\gamma})\,|\gamma=\hat{\gamma}\rangle = \begin{cases} 0, & \text{for } i\neq j \\ h_{w_j}(\hat{\gamma}), & \text{for } i=j \end{cases} \quad (2.121)$$

and on this basis $H_w(\hat{q})$, (2.119), which is the covariance of $w(u|\,\hat{q})$ local to $\gamma-\hat{\gamma}=0$, was taken as diagonal, i.e.

$$H_w(\hat{q}) = \text{Diag}(\ldots,\ h_{w_j}(\hat{\gamma}),\ \ldots),\ J \times J \quad (2.122)$$

where $h_{w_j}(\hat{\gamma})$, it was also found ([4], Appendix A), could be approximated, as follows:

$$h_{w_j}(\hat{\gamma}) \approx \frac{1}{1+2\ q_j(\hat{\gamma})} \quad (2.123)$$

with an error that peaked at about 20% for $\hat{q}_j=2$.

The antenna scanning function, $\theta_A(\tau)$, used was the following

$$\theta_A(\tau_j) \triangleq \begin{cases} \theta_{A_{max}} + \Omega\tau_j &, \quad 0\leq\tau_j<T_s \\ \theta_{A_{min}} &, \quad T_s\leq\tau_j<T_F \\ \theta_{A_{min}} - \Omega(\tau_j-T_F), & \quad T_F\leq\tau_j<T_1 \end{cases} \quad (2.124)$$

where the parameters are defined, as follows: (see Figure 2.1)

$$\Omega \triangleq -\frac{\theta_{A_{max}} - \theta_{A_{min}}}{T_s} \quad (2.125)$$

$$T_F \triangleq T_s + T_R - 2\,\frac{\theta_{A_{min}}}{\Omega} \quad (2.126)$$

$$T_s \triangleq \text{duration of the TO-scan} \quad (2.127)$$

$$T_R \triangleq \text{interval between zero intercepts} \quad (2.128)$$

$$T_1 \triangleq T_s + T_F \quad (2.129)$$

Values for the parameters θ_{Amax}, θ_{Amin}, T_s and T_R are essentially prescribed by the MLS specifications and will be given in the simulation

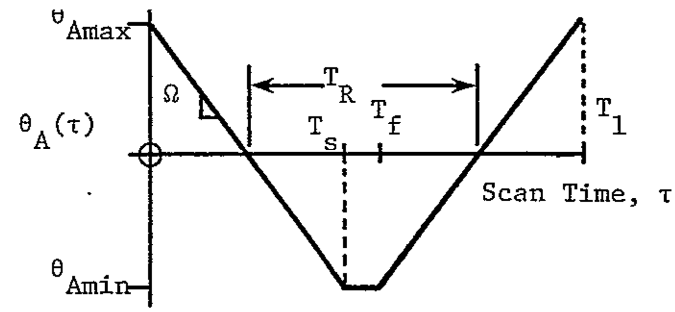


Figure 2.1 MLS Antenna Scanning Function

discussion. The antenna selectivity function, p( ), and its derivative $\dot{p}$( ), are not prescribed in the specifications; plausible functions were chosen for simulation use, however, and these will be described later, also. This completes the description of the scan data processor for the optimal design.

The $\hat{y}$ and $\hat{x}$ vector formulations in (2.85) and (2.86) require matrices F and H in the tracking loop, as follows:

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad (2.130)$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad (2.131)$$

The product matrix GQG^T used was a diagonal one in which the diagonal elements represented generally measures (variances) of the intuitive uncertainty in the elements of x(k), given those of x(k-1). The following values were selected as part of the tracking loop design:

$$GQG^T_{11} = GQG^T_{44} = \text{Max}\ \{(\tfrac{1}{2})^2,\ (\tfrac{\hat{Y}}{10})^2\} \qquad (2.132)$$

representing 10% uncertainty for S/N $\geq$ 14 db (in the direct path signal strength only; the assignment of the same value to the '44' component was arbitrary);

$$GQG^T_{22} = GQG^T_{55} = GQG^T_{77} = 0 \qquad (2.133)$$

representing full reliance in these coordinates on integrations of derivatives;

$$GQG^T_{33} = GQG^T_{66} = \left|\ddot{\theta}_{max}\right|^2 T \Big|_{AZ} = .01T \tag{2.134}$$

where $\left|\ddot{\theta}_{max}\right| = 0.1$ deg/sec^2 in Azimuth, was determined from a study of a representative set of landing patterns ([3], pp. 40ff);

$$GQG^T_{88} = \left(\frac{\Delta\beta}{T}\right)^2 = \frac{0.04}{T^2} \quad \left(<<\left(\frac{\pi}{T}\right)^2\right) \tag{2.135}$$

representing an error $\Delta\beta$ in phase (due to ω_{sc} uncertainty (i.e. error)) of 0.2 radians between scans, well within the limit of π radians associated with the sampling theorem.

Finally, in recognition of the limitations on the true values of the states, imposed by system geometry, modeling ambiguities, etc., the results obtained from the estimation algorithm described thus far were subjected to various additional constraint operations before being designated and subsequently evaluated as "the estimates". Each of these constraints are described pictorially in Figure 2.2 in the conventional format of an input--output graph of a function of one variable. In all cases the abscissa (input) is the result of the estimation update, (2.79) above, and the ordinate is the estimate to be output (or used in the next estimation cycle).

Suboptimal Design

For this case, the β_j, j=1,2, ---, J, are taken as independent random variables, each uniformly distributed on $[-\pi,\pi]$. Conceptually, $\hat{q}_j$ is also random and can be written

$$\hat{q}_j = \hat{q}_{A_j} + \hat{q}_{B_j}\cos\beta_j = q_j(\hat{\gamma},\beta_j) \tag{2.136}$$

where

$$\hat{q}_{A_j} \triangleq \hat{\alpha}^2 p_j^2(\hat{\theta}) + \hat{\alpha}_R^2\, p_j^2(\hat{\theta}_R) = q_{A_j}(\hat{\gamma}), > 0 \tag{2.137}$$

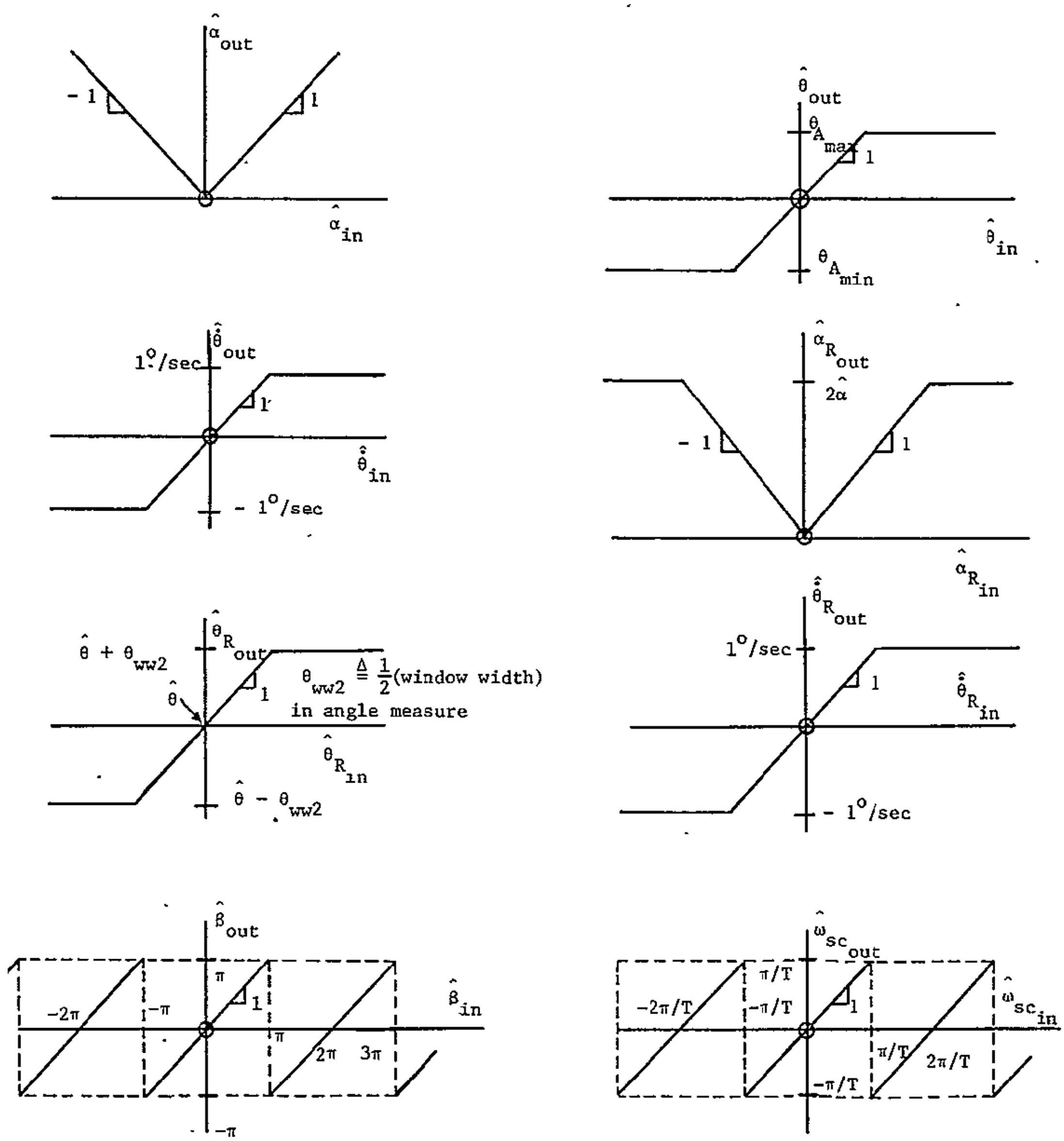


Figure 2.2 Constraints on Estimates

$$\hat{q}_{B_j} \triangleq 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta})p_j(\hat{\theta}_R) = q_{B_j}(\hat{\gamma}), \text{ indefinite} \tag{2.138}$$

and

$$-\hat{q}_{A_j} \leq \hat{q}_{B_j} \leq \hat{q}_{A_j} \tag{2.139}$$

We seek $\lambda(u|\hat{\gamma})$, which, applying (2.57), is given by

$$\lambda(u|\hat{\gamma}) = \langle \prod_{j=1}^{J} \lambda_j(u_j|q_j(\hat{\gamma},\beta_j))|u,\hat{\gamma}\rangle \tag{2.140}$$

in which the averaging is done wrt the β_j, j=1,2, ---, J. Because the β_j are independent this can be written in a form similar to (2.94) as follows:

$$\lambda(u|\hat{\gamma}) = \prod_{j=1}^{J} \lambda_j(u_j|\hat{q}_{A_j},\hat{q}_{B_j}) \tag{2.141}$$

where

$$\lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j}) \triangleq \langle\lambda_j(u_j|q_j(\hat{\gamma},\beta_j))|u_j,\hat{\gamma}\rangle \tag{2.142}$$

$$> 0 \text{ (by virtue of (2.95),(2.96))} \tag{2.143}$$

And, as in (2.99),

$$\Lambda(u|\hat{\gamma}) = \sum_{j=1}^{J} \Lambda_j(u_j|\hat{\gamma}) \tag{2.144}$$

where, here,

$$\Lambda_j(u_j|\hat{\gamma}) \triangleq \frac{\partial}{\partial\hat{\gamma}}\ln \lambda_j(u_j|\hat{q}_{A_j},\hat{q}_{B_j}) \tag{2.145}$$

$$= \frac{\frac{\partial}{\partial\hat{\gamma}} \lambda_j(u_j|\hat{q}_{Aj},\hat{q}_{Bj})}{\lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j})} \tag{2.146}$$

$$= \left(\frac{d\hat{q}_{A_j}}{d\hat{\gamma}}\right)\left(\frac{\frac{\partial}{\partial\hat{q}_{A_j}} \lambda(u_j|\hat{q}_{A_j},\hat{q}_{B_j})}{\lambda(u_j|\hat{q}_{A_j},\hat{q}_{B_j})}\right) + \left(\frac{d\hat{q}_{B_j}}{d\hat{\gamma}}\right)\left(\frac{\frac{\partial}{\partial\hat{q}_{B_j}} \lambda(u_j|\hat{q}_{A_j},\hat{q}_{B_j})}{\lambda(u_j|\hat{q}_{A_j},\hat{q}_{B_j})}\right) \tag{2.147}$$

Consequently, $\Lambda(u|\hat{\gamma})$ in (2.144) above, may be written

$$\Lambda(u|\hat{\gamma}) = D_A(\hat{\gamma})w_A(u|\hat{\gamma}) + D_B(\hat{\gamma})w_B(u|\hat{\gamma}) \tag{2.148}$$

where

$$D_A(\hat{\gamma}) \triangleq \begin{pmatrix} --- & \frac{\partial q_{A_j}(\hat{\gamma})}{\partial \hat{\gamma}_1} & --- \\ \hline --- & \frac{\partial q_{A_j}(\hat{\gamma})}{\partial \hat{\gamma}_{N_G}} & --- \end{pmatrix}, \; N_G \times J \tag{2.149}$$

$$\begin{pmatrix} --- & 2\hat{\alpha}p_j^2(\hat{\theta}) & --- \\ --- & 2\hat{\alpha}^2 p_j(\hat{\theta})\dot{p}_j(\hat{\theta}) & --- \\ --- & 2\hat{\alpha}_R p_j^2(\hat{\theta}_R) & --- \\ --- & 2\hat{\alpha}^2{}_R p_j(\hat{\theta}_R)\dot{p}_j(\hat{\theta}_R) & --- \end{pmatrix}, \tag{2.150}$$

$$D_B(\hat{\gamma}) \triangleq \begin{pmatrix} --- & \frac{\partial q_{B_j}(\hat{\gamma})}{\partial \hat{\gamma}_1} & --- \\ \hline --- & \frac{\partial q_{B_j}(\hat{\gamma})}{\partial \hat{\gamma}_{N_G}} & --- \end{pmatrix}, \; N_G \times J \tag{2.151}$$

$$\begin{pmatrix} --- & 2\hat{\alpha}_R p_j(\hat{\theta}_R)\; p_j(\hat{\theta}) & --- \\ --- & 2\hat{\alpha}\hat{\alpha}_R \dot{p}_j(\hat{\theta})p_j(\hat{\theta}_R) & --- \\ --- & 2\hat{\alpha}p_j(\hat{\theta})p_j(\hat{\theta}_R) & --- \\ --- & 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta})\dot{p}_j(\hat{\theta}_R) & --- \end{pmatrix} \tag{2.152}$$

$$w_A(u|\hat{\gamma}) \triangleq (---w_{A_j}(u_j|\hat{\gamma}) ---)^T, \text{ J-vector} \tag{2.153}$$

$$w_B(u|\hat{\gamma}) \triangleq (---w_{B_j}(u_j|\hat{\gamma})---)^T, \text{ J-vector} \tag{2.154}$$

in which

$$w_{A_j}(u_j|\hat{\gamma}) \triangleq \frac{\frac{\partial}{\partial \hat{q}_{A_j}} \lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j})}{\lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j})} \tag{2.155}$$

$$= \frac{\frac{\partial}{\partial \hat{q}_{A_j}} \lambda_j(u_j|q_j(\hat{\gamma}, \beta_j))|u_j, \hat{\gamma})}{<\lambda_j(u_j|q_j(\hat{\gamma}, \beta_j))|u_j \hat{\gamma}>} \tag{2.156}$$

$$= \frac{<(\frac{\partial \hat{q}_j}{\partial \hat{q}_{A_j}})(\frac{\partial \lambda_j}{\partial \hat{q}_j}(u_j|\hat{q}_j)|u_j, \hat{\gamma})>}{<\lambda_j(u_j|\hat{q}_j)|u_j, \hat{\gamma}>} \tag{2.157}$$

$$= \frac{\frac{\partial \lambda_j}{\partial \hat{q}_j}(u_j|\hat{q}_j)|u_j \hat{\gamma}>}{<\lambda_j(u_j|\hat{q}_j)|u_j, \hat{\gamma}>} \tag{2.158}$$

and, similarly

$$w_{B_j}(u_j|\hat{\gamma}) \triangleq \frac{\frac{\partial}{\partial \hat{q}_{B_j}} \lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j})}{\lambda_j(u_j|\hat{q}_{A_j}, \hat{q}_{B_j})} \tag{2.159}$$

$$= \frac{<(\frac{\partial \hat{q}_j}{\partial \hat{q}_{B_j}})(\frac{\partial \lambda_j}{\partial \hat{q}_j}(u_j|\hat{q}_j)|u_j, \hat{\gamma})>}{<\lambda_j(u_j|\hat{q}_j)|u_j, \hat{\gamma}>} \tag{2.160}$$

$$= \frac{<\cos\beta_j \frac{\partial \lambda_j}{\partial \hat{q}_j}(u_j|\hat{q}_j)|u_j, \hat{\gamma})}{<\lambda_j(u_j|\hat{q}_j)|u_j, \hat{\gamma}>} \tag{2.161}$$

Making use of (2.106) in (2.158) and (2.161) [and noting that $\hat{\gamma}$ in (2.106) included the phase difference parameter] gives

$$w_{A_j}(u_j|\hat{\gamma}) = \frac{\langle w_j(u_j|\hat{\gamma},\beta_j)\lambda_j(u_j|\hat{q}_j)|u_j,\hat{\gamma}\rangle}{\langle\lambda_j(u_j|\hat{q}_j)|u_j,\hat{\gamma}\rangle} \tag{2.162}$$

$$= \langle w_j(u_j|\hat{\gamma},\beta_j)W(u_j|\hat{\gamma},\beta_j)|u_j,\hat{\gamma}\rangle \tag{2.163}$$

$$w_{B_j}(u_j|\hat{\gamma}) = \langle w_j(u_j|\hat{\gamma},\beta_j)\cos\beta_j W(u_j|\hat{\gamma},\beta_j)|u_j,\hat{\gamma}) \tag{2.164}$$

where

$$W(u_j|\hat{\gamma},\beta_j) \triangleq \frac{\lambda_j(u_j|\hat{q}_j)}{\langle\lambda_j(u_j|\hat{q}_j)|u_j,\hat{\gamma}\rangle} > 0 \tag{2.165}$$

is a weighting factor which essentially modifies the (uniform) a priori distribution of β_j, giving an a posteriori one conditioned on u_j and $\hat{\gamma}$, i.e.

$$p(\beta_j|u_j,\hat{\gamma}) = W(u_j|\hat{\gamma},\beta_j)p(\beta_j) \tag{2.166}$$

(Clearly

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} p(\beta_j|u_j,\hat{\gamma})d\beta_j = \langle W(u_j|\hat{\gamma},\beta_j)|u_j,\hat{\gamma}\rangle = 1, \tag{2.167}$$

independent of u_j, $\hat{\gamma}$, as expected.) A more explicit form for w_{A_j} results from the substitution of (2.95) for λ_j and (2.107) for w_j into (2.162);

writing the expectations in integral form and taking advantage of the even symmetry wrt β_j(i.e. as cos β_j in $\hat{q}_j$) gives:

$$w_{A_j}(u_j|\hat{\gamma}) = \frac{\int_o^\pi [4u_j\frac{M_1}{Mo}(4\hat{q}_ju_j)-1]\, M_o(4\hat{q}_ju_j)\exp(-q_j)d\beta_j}{\int_o^\pi M_o(4\hat{q}_ju_j)\exp(-q_j)d\beta_j} \tag{2.168}$$

or, in terms of the more familiar modified Bessel functions of the first kind, I_0, and I_1,

$$w_{A_j}(u_j \mid \hat{\gamma}) = \frac{\int_o^{\pi} [(\frac{u_j}{\hat{q}_j})^{\frac{1}{2}} \frac{I_1}{I_o}(2(\hat{q}_j u_j)^{\frac{1}{2}}) - 1]\, I_0(2(\hat{q}_j u_j)^{\frac{1}{2}}) \exp(-\hat{q}_j) d\beta_j}{\int_o^{\pi} I_o(2(\hat{q}_j u_j)^{\frac{1}{2}}) \exp(-\hat{q}_j) d\beta_j} \quad (2.169)$$

Substituting the approximations (for any real z)

$$\frac{I_1}{I_o}(z) \approx \frac{z}{(4 + z^2)^{\frac{1}{2}}} \quad \text{(good to about 4\%)} \quad (2.170)$$

$$I_o(z) \approx \frac{\exp[(4 + z^2)^{\frac{1}{2}} - \frac{2}{1+z}]}{(1 + 2\pi z)^{\frac{1}{2}}} \quad \text{(good to about 7\%)} \quad (2.171)$$

and the following expansion for $\hat{q}_j$

$$\hat{q}_j = \hat{q}_{A_j}(1 + B_j \cos\beta_j) \quad (2.172)$$

where

$$B_j \triangleq \frac{\hat{q}_{B_j}}{\hat{q}_{A_j}} \quad (2.173)$$

(and, by virtue of (2.139)

$$-1 \le B_j \le +1) \quad (2.174)$$

gives

$$w_{A_j} = \frac{\int_o^{\pi} \omega_j L_j d\beta_j}{\int_o^{\pi} L_j \, d\beta_j} \quad (2.175)$$

$$w_{B_j} = \frac{\int_o^{\pi} \omega_j \cos\beta_j L_j d\beta_j}{\int_o^{\pi} L_j \, d\beta_j} \quad (2.176)$$

where

$$\omega_j \triangleq \frac{u_j}{(1 + \hat{q}_{A_j} u_j (1 + B_j \cos\beta_j))^{\frac{1}{2}}} - 1 \quad (2.177)$$

and

$$L_j \overset{\Delta}{=} \frac{\exp[2((1+\hat{q}_{A_j} u_j(1+B_j\cos\beta_j))^{\frac{1}{2}} - \frac{1}{1 + 2(\hat{q}_{A_j} u_j)^{\frac{1}{2}} (1+B_j\cos\beta_j)^{\frac{1}{2}}}) - q_A (1+B_j\cos\beta_j)]}{(1 + 4\pi(\hat{q}_{A_j} u_j)^{\frac{1}{2}} (1 + B_j\cos\beta_j)^{\frac{1}{2}})^{\frac{1}{2}}} \tag{2.178}$$

These last models, (2.175) and (2.176), were the basis of the calculations below.

Substituting (2.148) for $\Lambda(u|\hat{\gamma})$ into (2.59) for $\Phi(\hat{\gamma})$ gives

$$\Phi(\hat{\gamma}) = D_A(\hat{\gamma})H_{w_A}(\hat{\gamma})D_A^T(\hat{\gamma}) + D_A(\hat{\gamma})H_{w_{AB}}(\hat{\gamma})D_B^T(\hat{\gamma})$$

$$+ (D_A(\hat{\gamma})H_{w_{AB}}(\hat{\gamma})D_B^T(\hat{\gamma}))^T + D_B(\hat{\gamma})H_{w_B}(\hat{\gamma})D_B^T(\hat{\gamma}) \tag{2.179}$$

where, noting that now

$$u = u(\gamma,\beta_u,n), \tag{2.180}$$

the J x J matrices H_{w_A}, H_{w_B}, and $H_{w_{AB}}$ are given by the following:

$$H_{w_A}(\hat{\gamma}) \overset{\Delta}{=} \langle w_A(u(\gamma,\beta_u,n)|\hat{\gamma})w_A^T(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}\rangle \tag{2.181}$$

$$H_{w_B}(\hat{\gamma}) \overset{\Delta}{=} \langle w_B(u(\gamma,\beta_u,n)|\hat{\gamma})w_B^T(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}\rangle \tag{2.182}$$

$$H_{w_{AB}}(\hat{\gamma}) \overset{\Delta}{=} \langle w_A(u(\gamma,\beta_u,n)|\hat{\gamma})w_B^T(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}\rangle \tag{2.183}$$

in which these averages are taken wrt to the noise n and phase difference β_j in u. Assurances given by the LOE theory that

$$0 = \langle\Lambda(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma})\rangle \tag{2.184}$$

$$= D_A(\hat{\gamma})\langle w_A(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}) + D_B(\hat{\gamma})\langle w_B(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\gamma\rangle \tag{2.185}$$

strongly suggest that

$$\langle w_A(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}\rangle = 0 \tag{2.186}$$

$$\langle w_B(u(\gamma,\beta_u,n)|\hat{\gamma})|\gamma=\hat{\gamma}\rangle = 0 \tag{2.187}$$

and a simulation study of the processes $w_A(u|\hat{\gamma})$ and $w_B(u|\hat{\gamma})$, using numerical approximations of (2.175) and (2.176) (discussed below) respectively, gave support for (2.186) and (2.187), as well as strong evidence that

$$<w_{A_i}(u_i(\gamma,\beta_{u_i},n_i)|\hat{\gamma})w_{A_j}(u_j(\gamma,\beta_{u_j},n_j)|\hat{\gamma})|\gamma=\hat{\gamma}> = \begin{cases} 0 & i \neq j \\ h_{w_{A_j}}(\hat{\gamma}), & i=j \end{cases} \quad (2.188)$$

$$<w_{B_i}(u_i(\gamma,\beta_{u_i},n_i)|\hat{\gamma})w_{B_j}(u_j(\gamma,\beta_{u_j},n_j)|\hat{\gamma})|\gamma=\hat{\gamma}> = \begin{cases} 0 & i \neq j \\ h_{w_{B_j}}(\hat{\gamma}), & i=j \end{cases} \quad (2.189)$$

$$<w_{A_i}(u_i(\gamma,\beta_{u_i},n_i)|\hat{\gamma})w_{B_j}(u_j(\gamma,\beta_{u_j},n_j)|\hat{\gamma})|\gamma=\hat{\gamma}> = \begin{cases} 0 & i \neq j \\ h_{w_{AB_j}}(\hat{\gamma}), & i=j \end{cases} \quad (2.190)$$

and, on the basis of these conclusions, the matrices $H_{w_A}(\hat{\gamma})$ and $H_{w_B}(\hat{\gamma})$ which are the covariances of processes $w_A(u|\hat{\gamma})$ and $w_B(u|\hat{\gamma})$ local to $\gamma-\hat{\gamma}=0$, and the matrix $H_{w_{AB}}(\hat{\gamma})$, the cross-variance of the processes w_A and w_B local to $\gamma-\hat{\gamma}=0$, were taken as diagonal, i.e.

$$H_{w_A}(\hat{\gamma}) = \mathrm{Diag}(---, h_{w_{Aj}}(\hat{\gamma}), ---),\ J \times J \quad (2.191)$$

$$H_{w_B}(\hat{\gamma}) = \mathrm{Diag}(---, h_{w_{Bj}}(\hat{\gamma}), ---),\ J \times J \quad (2.192)$$

$$H_{w_{AB}}(\hat{\gamma}) = \mathrm{Diag}(---, h_{w_{AB_j}}(\hat{\gamma}), ---),\ J \times J \quad (2.193)$$

where definitions for $h_{w_{A_j}}(\hat{\gamma})$, $h_{w_{B_j}}(\hat{\gamma})$ and $h_{w_{AB_j}}(\hat{\gamma})$ were taken from equations (2.188), (2.189) and (2.190), respectively, above.

Efforts to use (2.175), (2.176), (2.188), (2.189) and (2.190) and obtain approximations, respectively, for w_{A_j}, w_{B_j}, $h_{w_{A_j}}$, $h_{w_{B_j}}$ and $h_{w_{AB_j}}$, analogous to those in (2.116) and (2.123) for w_j and h_{w_j}, respectively, in the optimal design, were <u>not</u> successful. As indicated above, it was necessary to use numerical procedures to perform the averaging indicated in the calculations of $w_{A_j}(u_j|\hat{\gamma})$, $w_{B_j}(u_j|\hat{\gamma})$ and $\Phi(\hat{\gamma})$). Numerical versions of (2.175) and (2.176) were used to calculate $w_{A_j}(u|\hat{\gamma})$ and $w_{B_j}(u|\gamma)$ in which integrations wrt β were replaced by simulation averages -- i.e. by summations over on index set of LMAX values of β taken uniformly over the interval $[0,\pi]$, with due regard for the dynamic range of the computing machine. The forms used are, as follows, suppressing the "j" subscript temporarily:

$$w_A = \left(\frac{\max_\ell(1, |h_{a_\ell}|)}{\max_\ell (1, h_\ell)}\right) \left(\frac{f_{s_A}}{f_s}\right) \tag{2.194}$$

$$w_B = \left(\frac{\max_\ell (1, |h_{b_\ell}|)}{\max_\ell (1, h_\ell)}\right) \left(\frac{f_{s_B}}{f_s}\right) \tag{2.195}$$

where

$$h_\ell = \frac{1}{(1 + 4\pi \quad (q_A u)^{\frac{1}{2}} \ (1 + B\cos\beta_\ell)^{\frac{1}{2}})^{\frac{1}{2}}} \tag{2.196}$$

$$h_{a_\ell} = \left(-1 + \frac{u}{(1 + q_A u(1 + B\cos\beta_\ell))^{\frac{1}{2}}}\right) h_\ell \tag{2.197}$$

$$h_{b_\ell} = h_{a_\ell} \cos\beta_\ell \tag{2.198}$$

$$f_s = \sum_\ell^{LMAX} h_\ell \exp(g_\ell - g_m) \tag{2.199}$$

$$f_{s_A} = \sum_\ell^{LMAX} h_{a_\ell} \exp(g_\ell - g_{m_a}) \tag{2.200}$$

$$f_{s_B} = \sum_{\ell}^{LMAX} h_{b_\ell} \exp(g_\ell - g_{m_b}) \qquad (2.201)$$

$$g_m = \ln(LMAX) + \ln\ (\max_\ell(1,h_\ell) + \max_\ell(g_\ell) - EXPMAX \qquad (2.202)$$

$$g_{m_a} = \ln(LMAX) + \ln\ (\max_\ell(1, \left|h_{a_\ell}\right|) + \max_\ell\ (g_\ell) - EXPMAX \qquad (2.203)$$

$$g_{m_b} = \ln(LMAX) + \ln\ (\max_\ell(1, \left|h_{b_\ell}\right|) + \max_\ell\ (g_\ell) - EXPMAX \qquad (2.204)$$

and

$$EXPMAX \triangleq \ln(\text{largest REAL variable representable on the computer}) \qquad (2.205)$$

$$(\simeq 88.\ \text{on the PDP-11, } 322.\ \text{on the CDC Cyber 172}) \qquad (2.206)$$

These equations were the basis for the simulation study that led to the conclusions of whiteness, equations (2.186) thru (2.190), and the consequent diagonality of H_{w_A}, H_{w_B}, and $H_{w_{AB}}$, (2.191) thru (2.193). These same equations were used in a numerical study of w_{A_j} and w_{B_j} as functions u_j, q_{A_j} and B_j in an effort to find approximating functions,

$$w_{A_j}(u_j | q_{A_j}, B_j),\ w_{B_j}(u_j | q_{A_j}, B_j)$$

"after the fact", again without success. As a result

i. The numerical computation of w_{A_j}, w_{B_j}, given u_j, q_{A_j}, and B_j, equations (2.194) thru (2.204) had to be programmed as part of the receiver design, to run "on-line" (with a substantial increment in processing time). More will be said about this in the simulation discussion in the next chapter.

ii. Also, the second-order averages, $h_{w_{A_j}}$, $h_{w_{B_j}}$ and $h_{w_{AB_j}}$, associated with the calculation of Φ could not be adequately approximated in closed-form, and numerical averages based on (2.188) thru (2.190) had to be done.

These latter calculations will be summarized at this point in the discussion since they were done "off-line" and hence were not part of the receiver simulation per se, but more a part of the design.

As (2.188) thru (2.190) suggests, the second-order averages had to be taken wrt the β_{u_j}-variable and the noise n_j in the observation sample, u_j, appearing as an argument of the conditional mean processes w_{A_j}, w_{B_j} -- clearly a very lengthy process but one fortunately that could be done off-line. Such an approach was used, but the effort then to find approximating functions "after the fact" was again not successful, and a plan was pursued involving calculating and storing many values of the second-order averages in tables off-line, then using table-lookup procedures indexed by q_{A_j} and B_j and interpolation on-line to calculate $\Phi(\hat{\gamma})$.

To improve the accuracy of these results and more nearly quarantee the non-negativeness of the calculated $\Phi(\hat{\gamma})$, the Φ-matrix was determined element-wise, as follows:

$$\Phi_{\ell i} = \Phi_{i\ell} = \sum_{j=1}^{J} [s_{w_{A_j}} D_{A_{ij}} - s_{w_{B_j}} D_{B_{ij}})(s_{w_{A_j}} D_{A_{\ell j}} - s_{w_{B_j}} D_{B_{\ell j}}) + (1 + R_{AB_j}) s_{w_{A_j}} s_{w_{B_j}} (D_{A_{ij}} D_{B_{\ell j}} + D_{A_{\ell j}} D_{B_{ij}})] \tag{2.207}$$

where

$$s_{w_{A_j}} \triangleq (h_{w_{A_j}})^{\frac{1}{2}}, \qquad s_{w_{A_j}} > 0 \tag{2.208}$$

$$s_{w_{B_j}} \triangleq (h_{w_{B_j}})^{\frac{1}{2}}, \qquad s_{w_{B_j}} > 0 \tag{2.209}$$

$$R_{AB_j} \triangleq \frac{h_{w_{AB_j}}}{s_{w_{A_j}} s_{w_{B_j}}} \quad , \quad -1 \le R_{AB_j} \le +1 \tag{2.210}$$

The averaging associated with these calculations also was done by simulation rather than by using numerical integration of the associated probability integrals. In particular, suppressing the above "j" subscript temporarily and considering (2.136) and (2.172) then clearly $u_{(j)}$ in (2.48) can be written without confusion as

$$u = q(q_A, B, \beta_u) + 2n_c[q(q_A, B, \beta_u)]^{\frac{1}{2}} + n_c^2 + n_s^2 \tag{2.211}$$

$$= u(\beta_u, n \,|\, q_A, B) \tag{2.212}$$

where, of course $q_A = q_A(\gamma)$ and $B=B(\gamma)$. Then $w_{A_{(j)}}$ can be denoted as

$$w_A(u \,|\, \hat{q}_A, \hat{B}) = w_A(u(\beta_u, n \,|\, q_A, B) \,|\, \hat{q}_A, \hat{B}) \tag{2.213}$$

and with this notation the calculation of s_{w_A} is described, as follows

$$s_{w_A} = \left[\frac{1}{L_{max} J_{n_{max}}} \sum_{\ell=1}^{L_{max}} \sum_{i=1}^{J_{n_{max}}} w_A^2(u(\beta_{u_\ell}, n_i \,|\, \hat{q}_A, \hat{B}) \,|\, \hat{q}_A, \hat{B})\right]^{\frac{1}{2}} \tag{2.214}$$

$$= S_{w_A}(\hat{q}_A, \hat{B}) \tag{2.215}$$

and similarly for s_{w_B}, etc. where the components n_{c_i}, n_{s_i} of the $J_{n_{max}}$ noise vectors n_i were drawn from a Gaussian pseudorandom noise generator (with mean zero, variance 0.5, see (2.11)) and the L_{max} values of β_{u_ℓ} were taken uniformly from the interval $[0,\pi]$. Values for L_{max} and $J_{n_{max}}$ used were

$$L_{max} = 11 \tag{2.216}$$

$$J_{n_{max}} = 400 \tag{2.217}$$

for each (q_{A_j}, B_j) point. In building the tables 300 (q_{A_j}, B_j) points were employed distributed generally, as follows

$$0.1 \leq q_{A_j} \leq 10^8 \qquad \text{(25 values)} \tag{2.218}$$

$$0.01 \leq B_j \leq 0.990 \qquad \text{(12 values)} \tag{2.219}$$

and values for $\ln(s_{w_{A_j}})$, $\ln(s_{w_{B_j}})$ and R_{AB_j} were calculated and stored. These off-line calculations were done by PROGRAM WLOGSW, associated subroutines WAVGS and WAWB, and FUNCTION GAUSS, which are all included in Appendix A. Values calculated for $\ln(s_{w_{A_j}})$, $\ln(s_{w_{B_j}})$, and R_{AB_j} are are given in Tables 2.1, 2.2 and 2.3 respectively.

Then, in the receiver, for each $(\hat{q}_{A_j}, \hat{B}_j)$ point associated with the estimate $\hat{y}(k\ k-1)$, the tables were entered and

i. Values $\hat{q}_{A_i}$, $\hat{q}_{A_{i+1}}$, $\hat{B}_i$, $\hat{B}_{i+1}$ from the tables were found, such that

$$\hat{q}_{A_i} \leq \hat{q}_{A_j} < \hat{q}_{A_{i+1}} \tag{2.220}$$

$$\hat{B}_i \leq \hat{B}_j < \hat{B}_{i+1} \tag{2.221}$$

ii. Then, linear interpolation between calculated averages in the table was done, using the general formula

$$f(x_1+\Delta x, y_1+\Delta y) = f(x_1,y_1) + \frac{f(x_2,y_1) - f(x_1,y_1)}{x_2 - x_1}\Delta x + \frac{f(x_1,y_2) - f(x_1,y_1)}{y_2 - y_1}\Delta y + \frac{f(x_1,y_1) + f(x_2,y_2) - f(x_1,y_2) - f(x_2,y_1)}{(x_2 - x_1)(y_2 - y_1)}\Delta x\Delta y \tag{2.222}$$

iii. The interpolated values of $\ln(s_{w_{A_j}})$, $\ln(s_{w_{B_j}})$ were exponentiated),

Table 2.1 Ln $(s_{w_{A_j}}(q_{A_j}, \hat{B}_j))$

B = QA:	.010	.020	.040	.060	.100	.200	.300	.500	.700	.900	.950	.990
.1000	-.11	-.11	-.11	-.11	-.11	-.11	-.11	-.11	-.11	-.11	-.11	-.11
.1778	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.16
.3162	-.23	-.23	-.23	-.23	-.23	-.23	-.23	-.23	-.23	-.24	-.24	-.24
.5623	-.33	-.33	-.33	-.33	-.33	-.33	-.33	-.33	-.34	-.34	-.35	-.35
1.000	-.47	-.47	-.47	-.47	-.47	-.47	-.47	-.48	-.49	-.50	-.50	-.51
1.778	-.65	-.65	-.65	-.65	-.66	-.66	-.66	-.68	-.69	-.71	-.72	-.72
3.162	-.90	-.90	-.91	-.91	-.91	-.91	-.92	-.95	-.97	-.98	-.97	-.97
5.623	-1.2	-1.2	-1.2	-1.2	-1.2	-1.2	-1.2	-1.3	-1.3	-1.2	-1.2	-1.2
10.00	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.6	-1.7	-1.7	-1.5	-1.4	-1.3
17.78	-1.8	-1.8	-1.8	-1.8	-1.8	-1.9	-1.9	-2.1	-2.0	-1.7	-1.6	-1.4
31.62	-2.1	-2.1	-2.1	-2.1	-2.1	-2.2	-2.3	-2.4	-2.4	-2.0	-1.8	-1.6
56.23	-2.3	-2.3	-2.4	-2.4	-2.4	-2.6	-2.7	-2.8	-2.7	-2.4	-2.1	-1.7
100.0	-2.6	-2.6	-2.7	-2.7	-2.8	-3.0	-3.1	-3.1	-3.0	-2.7	-2.4	-1.9
177.8	-2.9	-2.9	-3.0	-3.0	-3.1	-3.3	-3.4	-3.4	-3.4	-3.0	-2.7	-2.0
316.2	-3.2	-3.2	-3.3	-3.3	-3.5	-3.7	-3.8	-3.8	-3.6	-3.2	-2.9	-2.2
562.3	-3.5	-3.5	-3.6	-3.7	-3.9	-4.0	-4.1	-4.1	-3.8	-3.3	-3.1	-2.3
1000.	-3.6	-3.6	-4.0	-4.1	-4.3	-4.4	-4.4	-4.3	-3.9	-3.5	-3.2	-2.6
1778.	-4.1	-4.2	-4.4	-4.5	-4.6	-4.7	-4.7	-4.4	-4.1	-3.7	-3.4	-2.8
3162.	-4.4	-4.5	-4.7	-4.9	-5.0	-5.1	-4.9	-4.5	-4.3	-3.9	-3.7	-3.1
5623.	-4.7	-4.9	-5.1	-5.2	-5.3	-5.3	-5.0	-4.7	-4.6	-4.2	-4.0	-3.4
10000	-5.1	-5.3	-5.5	-5.5	-5.6	-5.4	-5.1	-5.0	-4.8	-4.5	-4.3	-3.7
.1000E+06	-6.6	-6.7	-6.8	-6.6	-6.3	-6.2	-6.1	-6.0	-5.9	-5.6	-5.4	-4.9
.1000E+07	-7.9	-7.7	-7.4	-7.3	-7.3	-7.2	-7.2	-7.2	-7.1	-6.8	-6.6	-6.0
.1000E+08	-8.6	-8.5	-8.4	-8.4	-8.4	-8.4	-8.4	-8.3	-8.2	-7.9	-7.7	-7.2
.1000E+09	-9.6	-9.5	-9.5	-9.5	-9.5	-9.5	-9.5	-9.5	-9.4	-9.1	-8.9	-8.3

Table 2.2 Ln $(s_{w_{\beta_j}}(q_{A_j}, B_j))$

B = QA:	.010	.020	.040	.060	.100	.200	.300	.500	.700	.900	.950	.990
.1000	-7.8	-7.1	-6.4	-6.0	-5.5	-4.8	-4.4	-3.9	-3.5	-3.2	-3.1	-3.1
.1778	-7.4	-6.7	-6.0	-5.6	-5.1	-4.4	-4.0	-3.5	-3.1	-2.8	-2.7	-2.7
.3162	-6.9	-6.2	-5.5	-5.1	-4.6	-3.9	-3.5	-3.0	-2.7	-2.4	-2.3	-2.3
.5623	-6.5	-5.8	-5.1	-4.7	-4.2	-3.5	-3.1	-2.6	-2.2	-2.0	-1.9	-1.9
1.000	-6.1	-5.4	-4.7	-4.3	-3.8	-3.1	-2.7	-2.2	-1.8	-1.6	-1.6	-1.5
1.778	-5.7	-5.1	-4.4	-4.0	-3.4	-2.8	-2.4	-1.9	-1.6	-1.4	-1.3	-1.3
3.162	-5.6	-4.9	-4.2	-3.8	-3.3	-2.6	-2.2	-1.7	-1.5	-1.3	-1.2	-1.2
5.623	-5.5	-4.9	-4.2	-3.8	-3.3	-2.6	-2.2	-1.8	-1.5	-1.3	-1.3	-1.2
10.00	-5.6	-4.9	-4.2	-3.8	-3.3	-2.6	-2.3	-1.9	-1.7	-1.5	-1.4	-1.3
17.78	-5.6	-4.9	-4.2	-3.8	-3.3	-2.7	-2.4	-2.2	-2.0	-1.7	-1.6	-1.4
31.62	-5.6	-4.9	-4.2	-3.8	-3.3	-2.8	-2.6	-2.5	-2.4	-2.0	-1.8	-1.6
56.23	-5.6	-4.9	-4.2	-3.8	-3.4	-2.9	-2.8	-2.8	-2.7	-2.4	-2.1	-1.7
100.0	-5.6	-4.9	-4.2	-3.9	-3.5	-3.1	-3.1	-3.1	-3.0	-2.7	-2.4	-1.9
177.8	-5.6	-4.9	-4.2	-3.9	-3.6	-3.4	-3.4	-3.4	-3.4	-3.0	-2.7	-2.0
316.2	-5.6	-4.9	-4.3	-4.0	-3.8	-3.7	-3.8	-3.8	-3.7	-3.2	-3.0	-2.2
562.3	-5.6	-4.9	-4.4	-4.1	-4.0	-4.0	-4.1	-4.1	-3.9	-3.5	-3.2	-2.4
1000.	-5.6	-5.0	-4.5	-4.3	-4.3	-4.4	-4.4	-4.4	-4.1	-3.7	-3.4	-2.6
1778.	-5.6	-5.0	-4.6	-4.6	-4.6	-4.7	-4.8	-4.6	-4.4	-3.9	-3.5	-2.9
3162.	-5.6	-5.1	-4.9	-4.9	-4.9	-5.1	-5.1	-4.8	-4.6	-4.1	-3.8	-3.2
5623.	-5.7	-5.3	-5.1	-5.2	-5.3	-5.4	-5.3	-5.1	-4.9	-4.3	-4.1	-3.5
10000	-5.6	-5.5	-5.5	-5.5	-5.6	-5.7	-5.5	-5.3	-5.1	-4.6	-4.4	-3.7
.1000E+06	-6.6	-6.7	-6.8	-6.8	-6.7	-6.6	-6.5	-6.3	-6.1	-5.8	-5.5	-4.9
.1000E+07	-7.9	-8.0	-7.8	-7.7	-7.7	-7.5	-7.5	-7.4	-7.3	-6.9	-6.7	-6.0
.1000E+08	-9.0	-8.9	-8.8	-8.7	-8.7	-8.7	-8.7	-8.6	-8.4	-8.1	-7.8	-7.2
.1000E+09	-10.	-9.6	-9.8	-9.8	-9.8	-9.8	-9.8	-9.7	-9.6	-9.2	-9.0	-8.3

Table 2.3 $R_{AB_j}(q_{A_j}, B_j)$

RAB:

B = / QA:	.010	.020	.040	.060	.100	.200	.300	.500	.700	.900	.950	.990
.1000	.14	.14	.14	.14	.14	.14	.14	.16	.18	.21	.21	.20
.1778	-.43E-01	-.43E-01	-.43E-01	-.43E-01	-.43E-01	-.41E-01	-.37E-01	-.26E-01	-.44E-02	.34E-01	.48E-01	.49E-01
.3162	-.16	-.16	-.16	-.16	-.16	-.16	-.16	-.15	-.14	-.12	-.10	-.96E-01
.5623	-.24	-.24	-.24	-.24	-.24	-.24	-.24	-.24	-.25	-.24	-.24	-.23
1.000	-.32	-.32	-.32	-.32	-.32	-.32	-.32	-.33	-.35	-.37	-.37	-.37
1.778	-.37	-.37	-.37	-.37	-.37	-.37	-.38	-.40	-.44	-.48	-.49	-.50
3.162	-.35	-.35	-.35	-.35	-.35	-.36	-.38	-.43	-.50	-.59	-.61	-.63
5.623	-.28	-.28	-.28	-.28	-.28	-.30	-.33	-.42	-.55	-.69	-.73	-.75
10.00	-.22	-.22	-.22	-.22	-.22	-.24	-.28	-.39	-.57	-.77	-.81	-.84
17.78	-.18	-.18	-.18	-.18	-.19	-.21	-.25	-.39	-.59	-.81	-.86	-.90
31.62	-.15	-.15	-.16	-.16	-.16	-.20	-.25	-.41	-.60	-.83	-.89	-.94
56.23	-.14	-.14	-.14	-.14	-.15	-.19	-.26	-.43	-.61	-.84	-.91	-.96
100.0	-.13	-.13	-.13	-.13	-.14	-.20	-.28	-.45	-.63	-.84	-.91	-.97
177.8	-.12	-.12	-.12	-.13	-.14	-.21	-.29	-.46	-.64	-.85	-.91	-.98
316.2	-.11	-.11	-.12	-.12	-.14	-.22	-.30	-.47	-.65	-.85	-.90	-.98
562.3	-.11	-.11	-.11	-.12	-.15	-.23	-.31	-.48	-.66	-.83	-.90	-.98
1000	-.11	-.11	-.11	-.12	-.15	-.23	-.31	-.48	-.66	-.82	-.89	-.97
1778	-.10	-.10	-.11	-.12	-.15	-.24	-.32	-.48	-.62	-.81	-.89	-.97
3162	-.10	-.99E-01	-.11	-.13	-.16	-.24	-.31	-.47	-.57	-.82	-.89	-.97
5623	-.96E-01	-.96E-01	-.11	-.13	-.16	-.23	-.30	-.41	-.55	-.82	-.89	-.96
10000	-.95E-01	-.97E-01	-.11	-.13	-.16	-.22	-.29	-.37	-.59	-.82	-.88	-.96
.1000E+06	-.93E-01	-.10	-.12	-.10	-.10	-.14	-.26	-.42	-.56	-.78	-.86	-.96
.1000E+07	-.93E-01	-.61E-01	-.35E-01	-.43E-01	-.87E-01	-.15	-.22	-.38	-.56	-.78	-.86	-.96
.1000E+08	-.19E-01	-.15E-01	-.36E-01	-.47E-01	-.74E-01	-.15	-.22	-.38	-.56	-.78	-.86	-.96
.1000E+09	-.94E-02	-.15E-01	-.30E-01	-.44E-01	-.74E-01	-.15	-.22	-.38	-.56	-.78	-.86	-.96

and the results with the interpolated value of R_{AB} were used in calculating $\Phi_{\ell i}$ via (2.207).

This completes the calculation of $\Lambda(u|\hat{\gamma})$ and $\Phi(\hat{\gamma})$, hence the description of the Scan Data Processor, for the Suboptimal Design.

The Tracking Loop for the Suboptimal Design is identical with that of the Optimal Design, except for changes due to the lower dimensions of the state and parameter vectors. Generally

i. The last two rows of the state vector estimate and corresponding rows and/or columns of associated matrices were eliminated;

ii. The last row of the parameter-vector (estimate) and corresponding rows and/or columns of associated matrices were eliminated.

This simple adaptation procedure results from the special formulations of the state- and parameter-vectors adopted. It was employed in the simulation easily and without problem.

This same general procedure was used also in obtaining the Non-Adaptive Design (recall (2.83), (2.84)) -- simply by (initializing $\hat{\gamma}_R=0$ and then) pruning the several vectors and matrices back to the appropriate dimensions, starting with either the optimal or suboptimal designs (both give the same results). This too was used in simulation without problem.

This concludes the formal development of the receiver algorithms. We turn next to the simulation studies and discussion of results.

SECTION III

SIMULATION STUDIES

The principal simulation result used in evaluating the performance of the various receiver designs was the calculated root mean square error (RMSE) (sample) statistic. A large number of studies of RMSE versus θ_{sep} were conducted, where

$$\theta_{sep} \triangleq \theta - \theta_R \tag{3.1}$$

These studies were parameterized, in general, by the following:

$$\text{DSNRDB (or S/N)} \triangleq 20 \log_{10} \alpha \tag{3.2}$$

$$\rho \triangleq \alpha_R/\alpha \tag{3.3}$$

β, the phase difference at the beginning of the simulation run (3.4)

$F_{sc} = \omega_{sc}/2\pi$, the scalloping rate (Hz) (3.5)

$B_{MLS} \triangleq$ the 3 db beam width of the MLS transmitting antenna (3.6)

$B_{RCVR} \triangleq$ the presumed 3 db beam width in the receiver of the MLS transmitting antenna (3.7)

Other RMSE studies performed included:

1. RMSE versus (B_{RCVR}/B_{MLS}), parameterized by S/N, ρ, β, F_{sc} and θ_{sep}; this study was deemed important because of the presumption in the Optimal and Suboptimal receivers of a value for the MLS ground (transmitting) antenna beam width, a parameter not currently transmitted in the preamble of the MLS signal.

2. RMSE versus F_{sc}, parameterized by S/N, ρ, β, θ_{sep}, B_{MLS}, B_{RCVR}.

Two series of error time-history studies were conducted also, all runs in both cases being parameterized by the full set, S/N, ρ, β, F_{sc}, θ_{sep}, B_{MLS}, B_{RCVR}, as well as other variants, such as the presence or absence of the constraints imposed on the estimates, etc. One set of error time-history studies, termed Interference Acquisition scenario runs, evaluated the "pull-in" ability of the interference tracker for potential use in interference acquisition. The second set of error time-history studies, termed Crossing Multipath scenario runs, evaluated the tracking performance of the receiver designs in dynamic environments involving multipath interference which is initially out-of-beam, then closes to in-beam, crosses, and finally opens to out-of-beam.

A major portion of the research was devoted to simulation studies -- both program development and receiver performance evaluation. The programs used in the receiver performance studies, totaling about 100 pages of FORTRAN code, are listed in Appendix A and are briefly discussed in the first part of this section. The second part of this section presents a representative selection of data from the various simulation runs, and discusses the results.

A. SIMULATION MODELS

The bulk of the computation performed in the simulation was done in FORTRAN subroutines described as follows:

MLSSUB: Simulation of the environment (via a state-space model with "true"-state, x(k)) and the received (envelope) signal vector, u(k); we note here that within a scan the quantity

$$\beta_j = \beta + \omega_{sc} T_j \qquad (3.7a)$$

is used as a better approximation of the phase difference (rather than simply β) in computing the jth sample, u_j, of the received envelope signal.

RCVR: Computation of the estimate, $\hat{\theta}(k|k)$, of the angular coordinate $\theta(k)$, given the observations vector, $u(k)$;

CONTRL: Conduct of the simulation run, including performing all I/O operations, special initializations and performance evaluation calculations.

The macro-flow-charts in Figure 3.1 show the organizations of these routines and their interrelations.

The simulation main program, MLSSIM, simply establishes COMMON storages and calls MLSSUB. Subroutine MLSSUB calls a library gaussian pseudorandom number generator function GAUSS and the following two MLS functions, in addition to RCVR and CONTRL:

PMLS: The antenna selectivity function, $p_{MLS}(\theta_e)$, of the MLS transmitting antenna, used in constructing the observations vector $u(k)$. The following -23 db sidelobe function was used in the study:

$$p_{MLS}(\theta_e) = \begin{cases} \pi/4, & |z| = 1 \\ \dfrac{\cos \frac{\pi}{2} z}{1 - z^2}, & |z| \neq 1 \end{cases} \qquad (3.8)$$

where $z \triangleq 2.4\theta_e / B_{MLS}$ (3.9)

$\theta_e \triangleq$ angle from beam center (3.10)

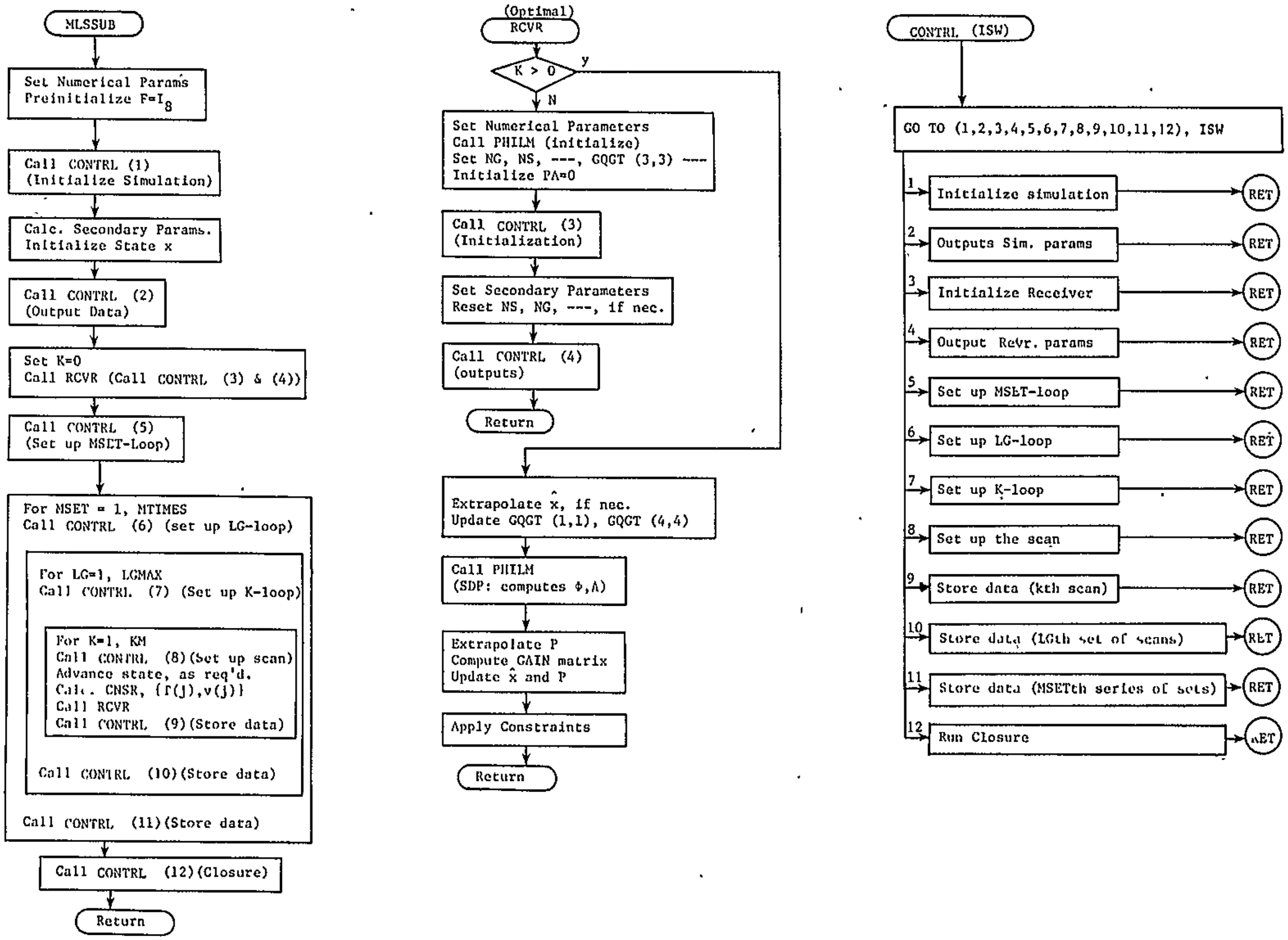


Figure 3.1 Flow Charts for MLSSUB, (OPT) RCVR, and CONTRL

THA: The antenna scanning function, $\theta_A(\tau)$, of the MLS transmitting antenna, used in constructing the vector u(k). The function in (2.124) above was specified in the MLS specifications and used in the study.

The programs MLSSIM, MLSSUB, THA, DFLTR1 (a 1st-order digital filtering subroutine used when the threshold receiver is running) and a BLOCK DATA program, MLS, collectively constituted a software module (or file), denoted MLSSIM. The function PMLS, and BLOCK DATA program PMLSID were put into a separate module, PMLS1, to facilitate changing the $p_{MLS}(\cdot)$ function. The listouts of these and other programs found in Appendix A are grouped into modules.

Two scenarios of subroutine RCVR were used, which are distinguished by module (or file) names suggestive of their natures, as follows:

OPTRVR: The optimal structure, comprising the Scan Data Processor and the Tracking Loop; calls subroutine PHILM (which calculates SDP quantities Φ, Λ, and is described below) as well as matrix arithmetic subroutines MATSM, MATMUL and MATINV.

THDRVR: A design similar to present commercial approaches using thresholding principles [11]. The simulation model, which involved

i) A 3db-below-peak threshold (referred to the linear envelope)

ii) A 300 microsec tracking gate for interference exclusion.

iii) A dwell gate for "loss-of-track" decisions

iv) Input of the log envelope signal, filtered by a 25 KHz low pass filter.

v) Error filtered with a 10 r/s bandwidth low pass filter for evaluation,

was developed to provide baseline data for performance comparisons. See the references [11], [4, pp. 25,26] and the program in Appendix A for further details. The program calls subroutine DFLTR1 to provide the 25 kHz filtering.

Two versions of PHILM were used, distinguished by module names, as follows:

PLOPT: The Scan Data Processor calculations of Φ, Λ for the Optimal design. This uses function THA, (2.124) above, and functions P and PDOT given below. We note here that in the calculation of matrix D in PLOPT, analogous to (3.7a), the quantity

$$\hat{\beta}_j = \hat{\beta} + \hat{\omega}_{sc}\,\tau_j \tag{3.10a}$$

was used for $\hat{\beta}$ in (2.103) in an effort to improve the receiver performance by making use of the $\hat{\omega}_{sc}$-information at this point.

PLSUB: The SDP calculations of Φ, Λ for the Suboptimal design. This calls the same functions THA, P, PDOT used by PLOPT, preceeding, but also calls subroutines WAWBJ (which calculated "on-line" conditional averages

$w_A(u_j\ \gamma)$ and $w_B(u_j\ \gamma)$ using (2.194), (2.195) and succeeding equations) and subroutine SWFCNS (which did the table look-up and necessary interpolation to produce the values $s_{w_{A_j}}$, $s_{w_{B_j}}$ and R_{AB_j} (2.208), (2.209) and (2.210) respectively needed to calculate matrix Φ via (2.207). See the program in Appendix A for further details.

The functions P, PDOT constitute the module POPT1 and are used by both versions of PHILM. They are described as follows:

P: The antenna selectively function $p(\theta_e)$ assumed in the receiver design to be in effect in the received signal vector u(k). The following -23 db sidelobe function was used in the study:

$$p(\theta_e) = \begin{cases} \pi/4, & |z| = 1 \\ \dfrac{\cos \frac{\pi}{2} z}{1 - z^2}, & |z| \neq 1 \end{cases} \tag{3.11}$$

where $z \triangleq 2.4\ \theta_e/B_{RCVR}$ (3.12)

$\theta_e \triangleq$ angle from beam center (3.13)

This is the same function as $p_{MLS}(\theta_e)$, (3.8), but it was programmed twice with distinct names to allow different functions to be used (alternate function were not studied, however). Figure 3.2 shows the function $p(\cdot)$, (3.11) above, centered in the same 65-sample window on the same sampling grid (FSAMP = 160 kHz, OMEGA = 20,000 Deg./sec.) as in the receiver.

```
RUN PPLOT
  BMLS =    1.00000
 OMEGA =    20000.0
 FSAMP =    160000.
DTHETA =  0.125000
  JMAX =       65
THEMIN =  -4.00000
THEMAX =   4.00000
```

ORDINATE AXIS: MIN=-0.695066E-01 MAX= 1.00000
INCREMENT= 0.178251E-01

THETA DEGREES	P
-4.00000	0.887470E-02
-3.87500	0.531045E-02
-3.75000	0.000000
-3.62500	-0.607831E-02
-3.50000	-0.116305E-01
-3.37500	-0.152869E-01
-3.25000	-0.158933E-01
-3.12500	-0.127983E-01
-3.00000	-0.607825E-02
-2.87500	0.335622E-02
-2.75000	0.138107E-01
-2.62500	0.230294E-01
-2.50000	0.285714E-01
-2.37500	0.282949E-01
-2.25000	0.208731E-01
-2.12500	0.625487E-02
-2.00000	-0.140207E-01
-1.87500	-0.367328E-01
-1.75000	-0.571548E-01
-1.62500	-0.695066E-01
-1.50000	-0.676436E-01
-1.37500	-0.459040E-01
-1.25000	-0.468134E-07
-1.12500	0.721765E-01
-1.00000	0.169962
-0.875000	0.289645
-0.750000	0.424579
-0.625000	0.565685
-0.500000	0.702312
-0.375000	0.823339
-0.250000	0.918414
-0.125000	0.979128
0.000000	1.00000
0.125000	0.979128
0.250000	0.918414
0.375000	0.823339
0.500000	0.702311
0.625000	0.565685
0.750000	0.424579
0.875000	0.289645
1.00000	0.169962
1.12500	0.721765E-01
1.25000	-0.936267E-07
1.37500	-0.459040E-01
1.50000	-0.676436E-01
1.62500	-0.695066E-01
1.75000	-0.571548E-01
1.87500	-0.367328E-01
2.00000	-0.140207E-01
2.12500	0.625487E-02
2.25000	0.208731E-01
2.37500	0.282949E-01
2.50000	0.285714E-01
2.62500	0.230294E-01
2.75000	0.138107E-01
2.87500	0.335624E-02
3.00000	-0.607822E-02
3.12500	-0.127983E-01
3.25000	-0.158933E-01
3.37500	-0.152869E-01
3.50000	-0.116305E-01
3.62500	-0.607832E-02
3.75000	0.000000
3.87500	0.531045E-02
4.00000	0.887470E-02

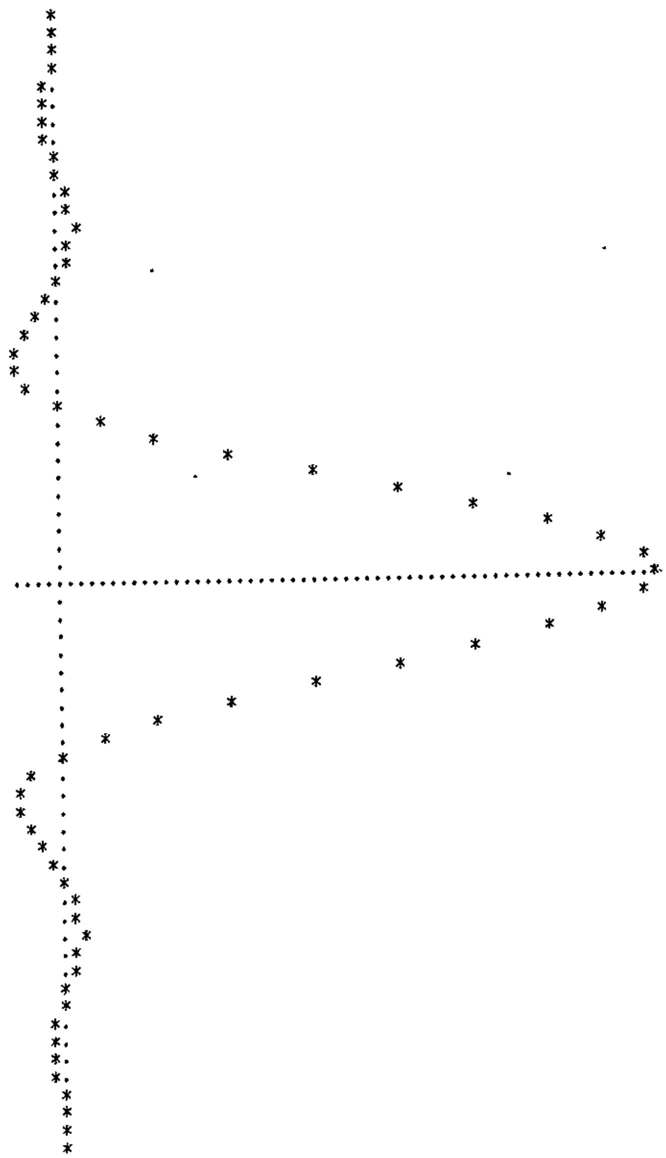


Figure 3.2 Antenna Selectivity function, p(θ), used

PDOT: The function $\frac{dp(\theta_e)}{d\theta_e}$. On the basis of (3.11) thru (3.13) the following was used:

$$\frac{dp(\theta e)}{d\theta e} = \begin{cases} \frac{-0.3\pi}{B_{RCVR}} \text{ signum }(z), & |z| = 1 \\ \frac{0.3\pi^2}{B_{RVCR}} \left\{ \frac{\cos(z+1)\pi/2 - \frac{\sin(z+1)\pi/2}{(z+1)\pi/2}}{(z+1)\pi/2} + \frac{\cos(z-1)\pi/2 - \frac{\sin(z-1)\pi/2}{(z-1)\pi/2}}{(z-1)\pi/2} \right\}, & |z| \neq 1 \end{cases} \quad (3.14)$$

Several versions of subroutine CONTRL, distinguished by module names, as follows, were used, conducting the simulation through various types of scenarios and test runs:

CTLACQN: An interference acquisition scenario, testing the "pull-in" power of the interference tracking algorithm. Initially no interference is present and the interference tracker states are tethered to "idler" values, as follows:

$$\hat{\alpha}_{R_o} = 0.5\ \hat{\alpha} \quad (3.15)$$

$$\hat{\theta}_{R_o} = \hat{\theta} - 1.5 \text{ degrees} \quad (3.16)$$

$$\hat{\dot{\theta}}_{R_o} = 0 \text{ degrees/second} \quad (3.17)$$

$$\hat{\beta}_o = \pi/2 \text{ radians} \quad (3.18)$$

$$\hat{\omega}_{sc_o} = 0 \text{ radians/second} \quad (3.19)$$

A step of interference then occurs with prescribed parameter values, the interference tracker is untethered and the estimate error-time-histories are generated.

CTLCRMP: A crossing multipath scenario, testing the tracking abilities of a receiver design from an out-of-beam interference condition, thru a crossing in-beam condition to a out-of-beam situation again under prescribed conditions; error-time-histories are generated.

CTLMSTH: An RMS error versus θ_{sep} study in which, by a suitable nonzero assignment of ω_{sc}, the effects of β are approximately averaged out as the statistical sample of desired size, 100 scans, evolves. 110 scans are calculated and the first 10 discarded in computing error statistics for each value of θ_{sep}. The program increments θ_{sep} and repeats the calculation for up to 13 values of θ_{sep}.

CTLOE: Another, more expensive, RMS error versus θ_{sep} study in which for each value of θ_{sep} β is stepped through 20 values uniformly spaced on the $(-\pi,\pi)$ interval and, for each value of β, 30 scans are generated, the first 10 being discarded and the latter 20 being used in the statistical calculations.

CTLMSSBB: An RMS error versus B_{ratio} study, where

$$B_{ratio} \triangleq B_{RCVR}/B_{MLS}. \tag{3.20}$$

The study is performed in a manner similar to that in CTLMSTH above, including the use of nonzero ω_{sc}.

7 values of B_{ratio} were used in the range

$$(10)^{-\frac{1}{2}} \leq B_{ratio} \leq (10)^{\frac{1}{2}} \qquad (3.21)$$

with

$$\left.\begin{array}{l} B_{RCVR} = 1° \\ B_{MLS} > 1° \end{array}\right\} \text{, when } B_{ratio} < 1 \qquad (3.22)$$

and

$$\left.\begin{array}{l} B_{RCVR} > 1° \\ B_{MLS} = 1° \end{array}\right\} \text{, when } B_{ratio} > 1 \qquad (3.23)$$

CTLMSFS: An RMS Error versus F_{sc} study. This study is also performed in a manner similar to that in CTLMSTH above, except that ω_{sc} is assigned higher and higher integer multiples of the minimum value (0.135 Hz in AZIMUTH) which would integrate β over a 2π interval during a 100-scan time period.

Block data programs were included in many modules to initialize COMMON storages. A library of general math and utility programs was also used and is included in the program listings in Appendix A.

All simulation runs were made with Azimuth angle function data, though the option for elevation simulation was included in the programs. Parameter values written in storages are, as follows:

	AZ	EL	
$\theta_{A_{max}}$ =	62.666667°	30.666667°	(3.24)
$\theta_{A_{min}}$ =	-62.0°	0.0°	(3.25)
T_s =	6.233333 ms	1.533333 ms	(3.26)
T_R =	6.6 ms	0.4 ms	(3.27)
Rep. Rate = 1/T =	13.5 Hz	40.5 Hz	(3.28)

Except as noted above for B_{ratio} studies, values assigned B_{MLS} and B_{RCVR} were, as follows:

$$B_{MLS} = B_{RCVR} = 1^\circ \tag{3.29}$$

Specification of the intensity of the interference was made using the parameter ρ defined in (3.3). As indicated above in connection with Figure 3.2, 65 samples were taken in each semiscan, i.e.

$$J = 130 \tag{3.30}$$

at the sampling rate

$$F_{samp} = 160 \text{ KHz} \tag{3.31}$$

with the 33rd and 98th samples occuring where the peaks of the direct path pulses were expected, based on $\hat{\theta}(k|k-1)$.

Estimation error was calculated in the expected manner, i.e. x-x, except for components associated with β, $\dot{\omega}_{sc}$ (and $\hat{\omega}_{sc}$ in the 6D LOE model). In these components, the differences in absolute values, e.g. $|\beta| - |\hat{\beta}|$, were used for error evaluation to accomodate the ambiguity in these variables.

B. SIMULATION RUNS AND RESULTS

The results of nearly 50 runs are reported here in 23 plots and 23 tables. Table 3.1 summarizes the runs made--by type of run and CONTRL module, parameter values used, RCVR type used and figure numbers and table numbers in which the results appear. [In <u>these</u> discussions "OPTRVR" implies the use of modules OPTRVR and PLOPT (an abuse of terminology, perhaps), "SUBOPT" implies the use of modules OPTRVR and PLSUB.]

Figure 3.3 shows the computed RMS error (θ component) versus θ_{sep} for several S/N values for the OPTRVR, ρ=0.8. Figure 3.4 presents the

Table 3.1 Summary of Simulation Runs

Type of Run (Config)	S/N	ρ	θ_{SEP}	β	F_{SC}	B_{ALS}	B_{RCVR}	Figure Nos. (3.-) OPT RVR	Figure Nos. (3.-) SUBOPT	Figure Nos. (3.-) THD RVR	Table Nos. (3.-) OPT RVR	Table Nos. (3.-) SUBOPT	Remarks
RMSE (θ_{SEP}) (CTMOTH)	40	0.8	-.375° to 1.575°	180°	0.135 Hz	1°	1°	3			2		
	40	0.5						4			3		
	30	0.8						3,5	5	5	4	10	
	30	0.5						4			5		
	20	0.8						3,6	6,8	6	6	11	Non-Adapt. also
	20	0.5						4,7	7	7	7	12	
	14	0.8						3			8		
	14	0.5						4			9		
RMSE (θ_{SQ}) (CTCNE)	20	0.8	-.375° to 1.625°	180° only	0 Hz	1°	1°		8				
				every 18°, 20 values					8				
				180° only					8				
RMSE (B_{RCVR}/B_{ALS}) (CTTHREE)	20	0.8	1.5°	-180°	0.35 Hz	3.16° to 1°	1° to 3.16°	9			13		
	30	0.8						9			14		
	30	0.5						9			15		
	20	0.5						9			16		
RMSE (F_{SC})	30	0.8	1.5°	0°	0 to 16.75 Hz	1°	1°	10			17		
					47.25 to 60.75			11			18	20	No constraints
								11	11		19	21	with constr.
								11			22		Tethers, N.C.
								11			23		Tethered, W.C.
Interfer Ang. (CTIANG)	40	0.8	0.25°	45°	0.6 Hz	1°	1°	12					
			2°					13					
			3°					14					
			2°		2.5 Hz			15					
	20		0.5°		0.6 Hz.			16	22				
			3°		0.6 Hz			17					
			0.5°		2.5 Hz.			18					
	14		1°		0.6 Hz			19					
	40		1°	-168°	51.3 Hz.			20	23				
	40		0.5	45°	0.6 Hz				21				
Crossing Multipath (CTCRMP)	20	0.8	-2.75° to 2.3°	-168°	51.3 Hz	1°	1°	24	24	24			
	20	0.8	-2.75° to 2.3°	-168°	51.3 Hz.	1°	1°	25	25	25			

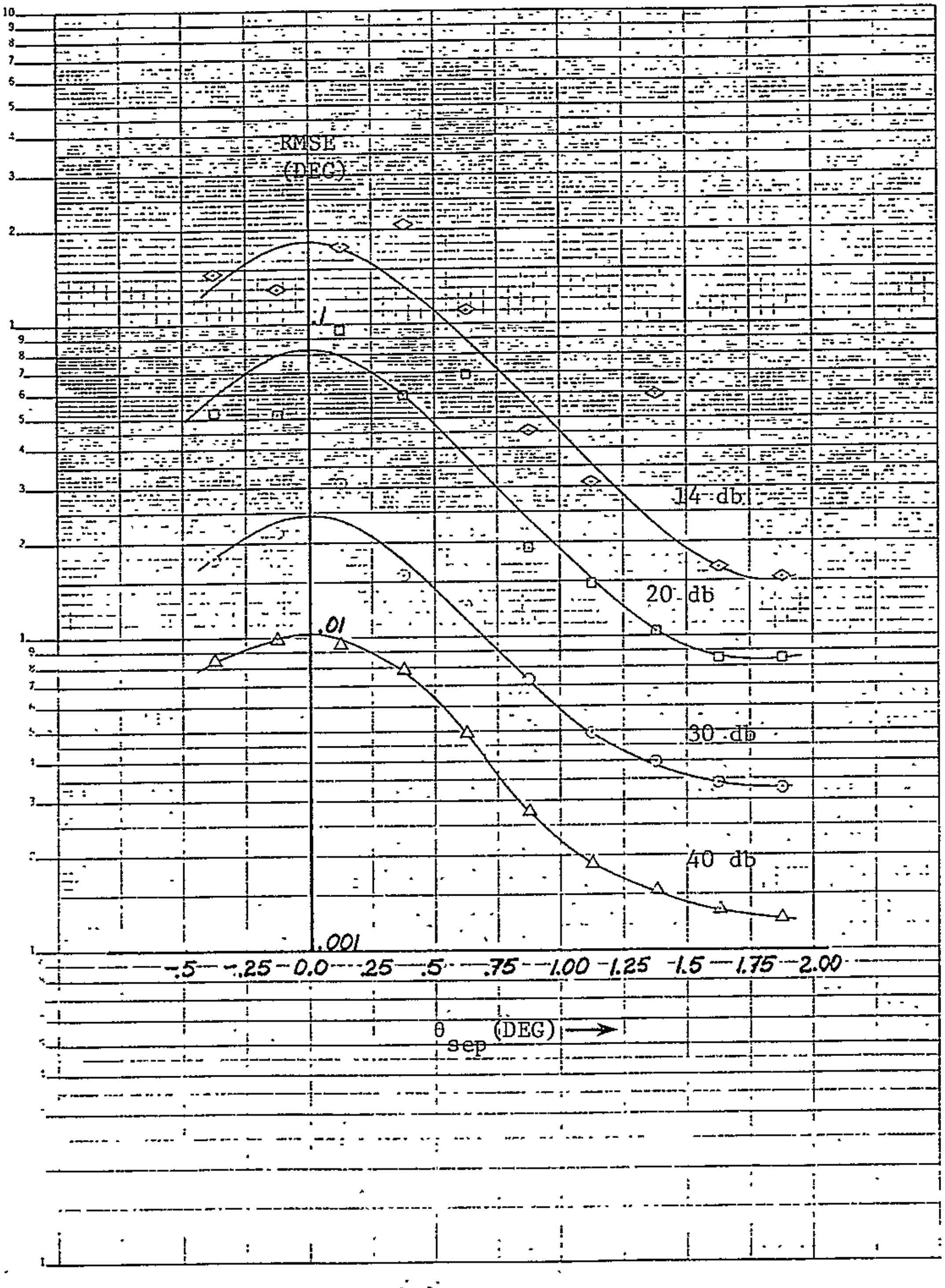


Figure 3.3 RMSE (θ_{sep}), OPTRVR, ρ=0.8

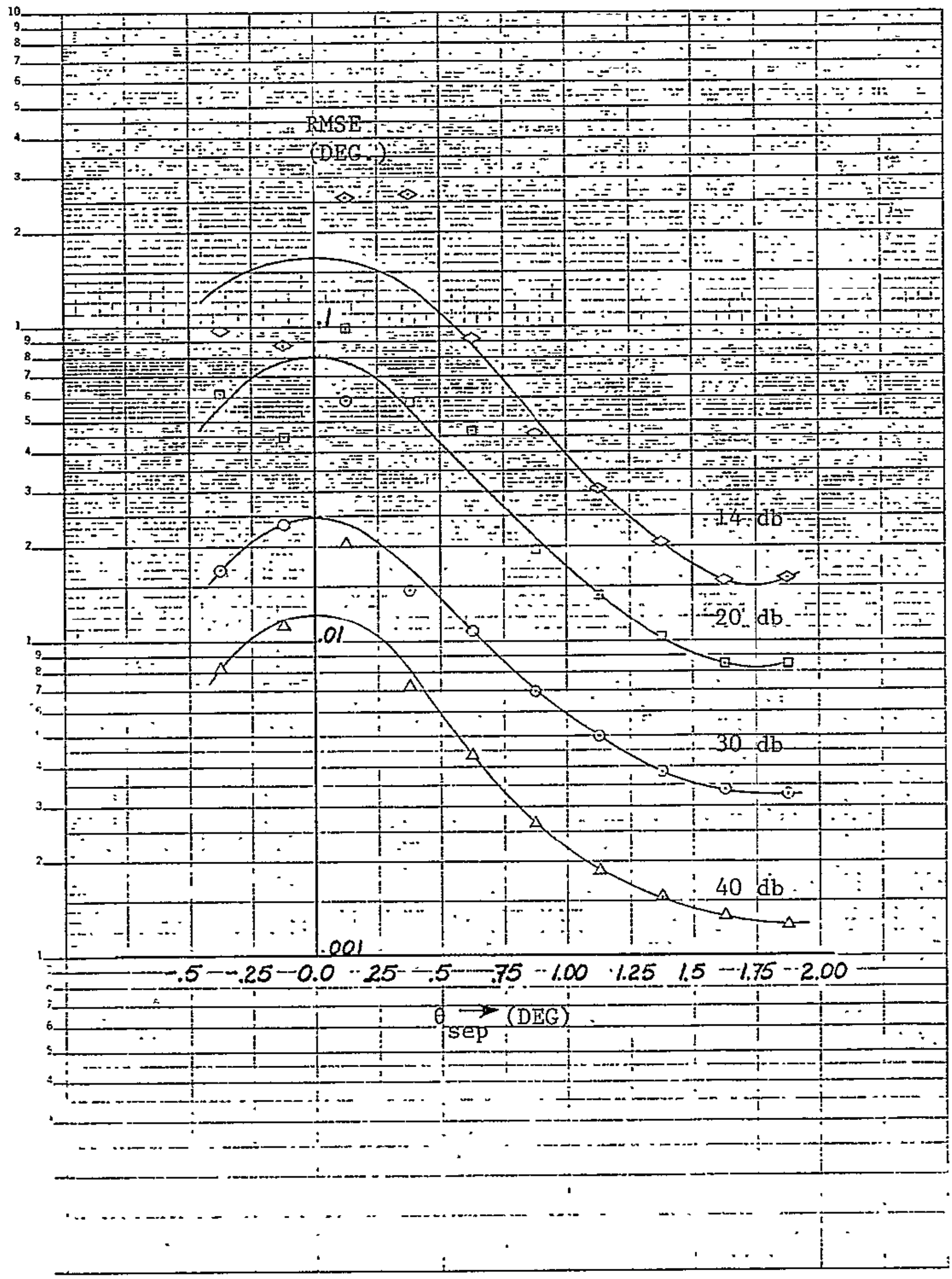


Figure 3.4 RMSE (θ_{sep}), OPTRVR, ρ=0.5

same type of data for ρ=0.5. The scatter in the points for the lower S/N cases is due to noise; the apparent odd symmetry in the scatter about the origin is due to several causes:

i. The lack of symmetry between the TO, FRO scans wrt the noise samples;

ii. The effect of noise on the β-estimate (in one run, not included here, $\hat{\beta}$ was tethered to the true value β and the calculated RMSE (θ_{sep})(θ-component) exhibited more nearly the expected even symmetry wrt θ_{sep}.

iii. The use of the <u>same</u> noise sample function for each data point.

Figures 3.5 thru 3.7 show comparisons of RMSE (θ_{sep}) for the receivers for various values of S/N and ρ. Figure 3.6, for example, shows that with S/N=20 db, ρ=0.8, the optimal design offers improvement by a factor up to about 30 over the threshold receiver. The suboptimal design, without tracking phase difference, shows improvement by a factor up to about 15 over the threshold receiver. The non-adaptive design, basically of optimal <u>structure</u>, but premised on interference-free reception, shows approximately equivalent performance as the threshold receiver.

Tables 3.2 thru 3.12 give the full results of the OPTRVR and SUBOPT simulation runs associated with Figures 3.3 thru 3.7 -- error statistics for each coordinate of the state estimate as a function of θ_{sep}. One observation that can be made from this data is that the mean error at small separation angles becomes a more significant contributor to the RMSE as the S/N diminishes. This probably signals a diminishing validity of the LOE criterion (error being in a neighborhood of zero), due

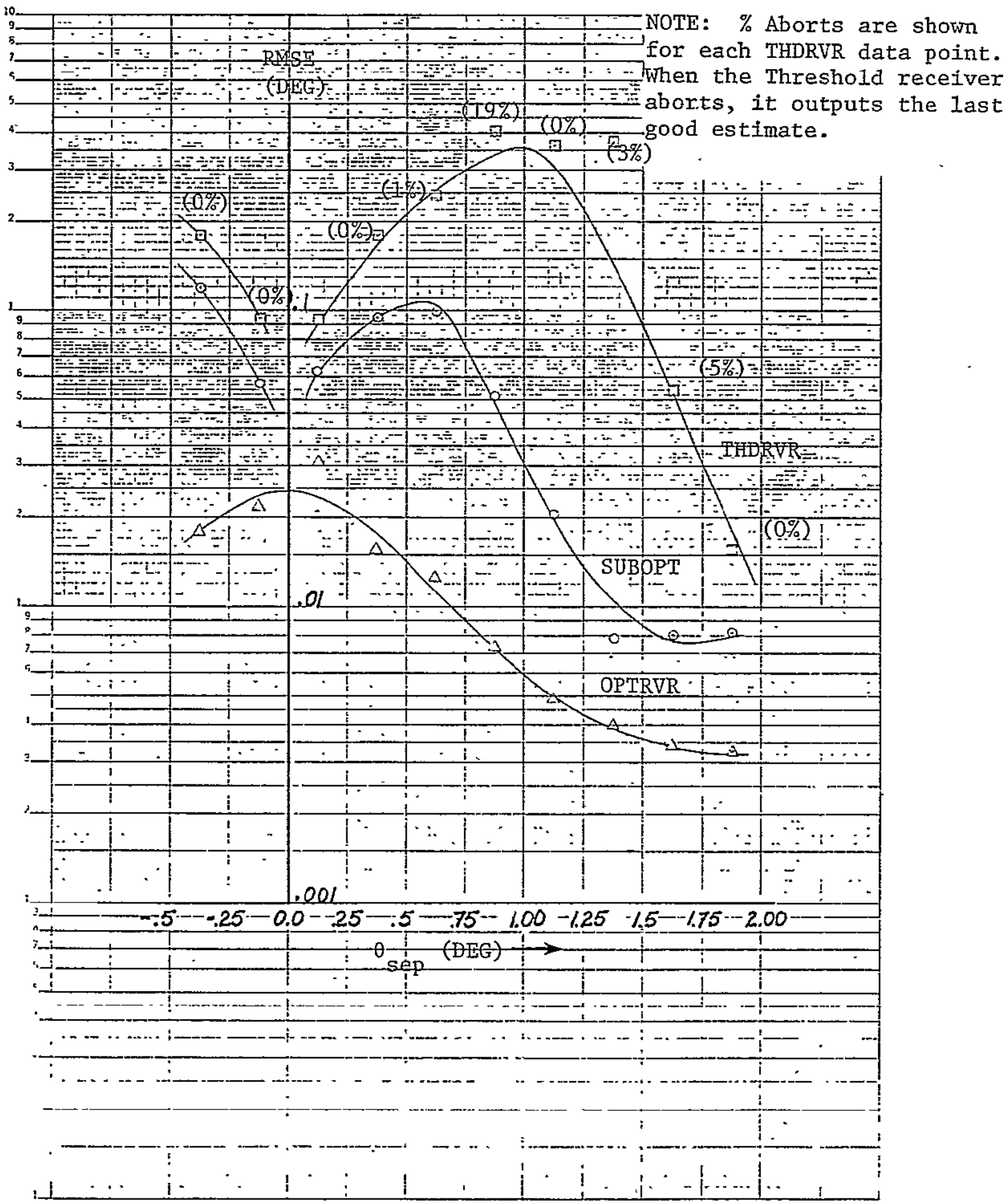


Figure 3.5 RMSE (θ_{sep}), comparison of receivers, S/N=30 db, ρ=0.8

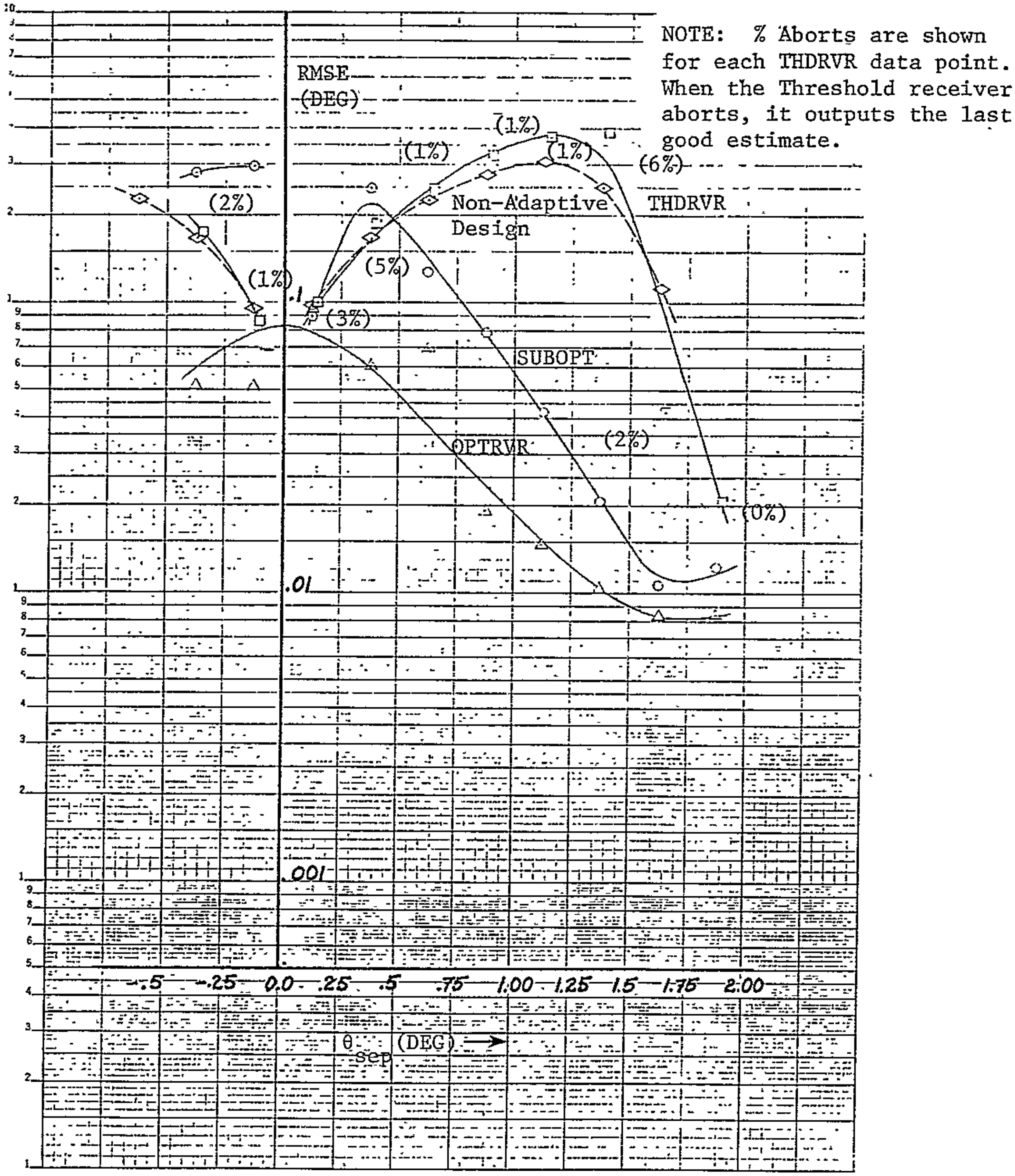


Figure 3.6 RMSE (θ_{sep}), comparison of receivers, S/N=20db, ρ=0.8

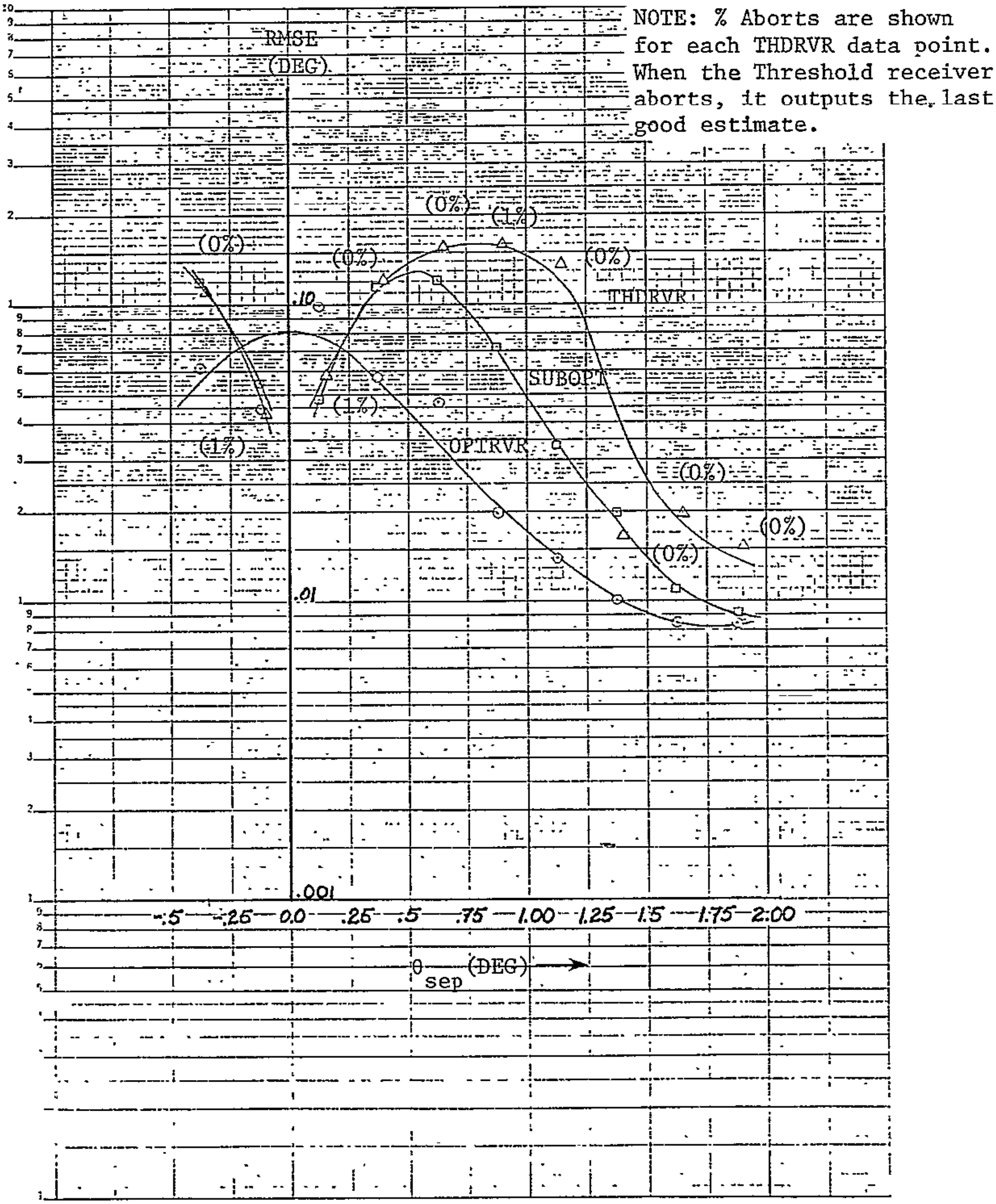


Figure 3.7 RMSE (θ_{sep}), comparison of receivers
S/N = 20db, ρ=0.5

NRUN = 1

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	-1.43	-.286E-02	.146E-02	-1.07	.260E-02	-.163E-02	-.272E-01	-.441E-01
	-.125	-3.82	-.292E-02	.649E-03	12.1	-.164E-01	-.200E-02	.378E-01	-.233
	.125	1.25	-.228E-02	-.102E-02	8.26	.105E-01	.144E-02	.603E-01	-.369
	.375	-.581	.144E-02	.112E-02	-2.27	-.626E-02	-.218E-02	-.208E-01	-.154E-01
	.625	-.280	.149E-02	.990E-03	-.609	-.339E-02	-.171E-02	-.186E-01	-.269E-02
	.875	.257E-01	-.118E-04	.451E-04	-.774E-01	-.731E-03	-.414E-03	-.464E-02	-.176E-02
	1.13	.254E-01	-.137E-03	-.145E-03	.562E-02	-.344E-04	-.929E-04	.231E-03	.190E-02
	1.38	.215E-01	-.138E-03	-.493E-04	.107E-01	.127E-05	-.231E-03	.191E-02	-.566E-01
	1.63	.150E-01	-.734E-04	-.134E-04	.120E-01	-.141E-05	-.310E-03	.123E-02	.930E-02
	1.88	.103E-01	.586E-04	.938E-04	.144E-01	.304E-05	-.312E-03	.583E-03	-.413E-01
ERMS:	-.375	4.12	.846E-02	.236E-01	4.15	.113E-01	.295E-01	.678E-01	.438
	-.125	12.7	.998E-02	.209E-01	15.0	.209E-01	.269E-01	.163	.857
	.125	10.9	.956E-02	.208E-01	13.9	.188E-01	.280E-01	.181	.948
	.375	3.67	.793E-02	.237E-01	3.87	.105E-01	.244E-01	.489E-01	.437
	.625	1.13	.492E-02	.223E-01	1.18	.638E-02	.212E-01	.407E-01	.348
	.875	.441	.276E-02	.172E-01	.456	.358E-02	.193E-01	.224E-01	.241
	1.13	.247	.188E-02	.152E-01	.232	.233E-02	.167E-01	.251E-01	.259
	1.38	.183	.154E-02	.148E-01	.184	.186E-02	.154E-01	.234E-01	.429
	1.63	.161	.132E-02	.142E-01	.174	.162E-02	.157E-01	.313E-01	.357
	1.88	.161	.125E-02	.137E-01	.177	.145E-02	.152E-01	.391E-01	.474
ESTD:	-.375	3.87	.796E-02	.236E-01	4.01	.110E-01	.294E-01	.622E-01	.435
	-.125	12.1	.954E-02	.209E-01	8.79	.130E-01	.268E-01	.158	.825
	.125	10.6	.928E-02	.207E-01	11.2	.157E-01	.280E-01	.171	.873
	.375	3.63	.780E-02	.237E-01	3.13	.840E-02	.243E-01	.443E-01	.437
	.625	1.10	.469E-02	.223E-01	1.01	.540E-02	.212E-01	.362E-01	.348
	.875	.440	.276E-02	.172E-01	.449	.350E-02	.193E-01	.219E-01	.241
	1.13	.246	.188E-02	.152E-01	.232	.232E-02	.167E-01	.251E-01	.259
	1.38	.182	.153E-02	.148E-01	.184	.186E-02	.154E-01	.233E-01	.425
	1.63	.161	.132E-02	.142E-01	.174	.162E-02	.157E-01	.312E-01	.357
	1.88	.161	.124E-02	.137E-01	.177	.145E-02	.152E-01	.391E-01	.473

Table 3.2 Error Statistics vs $|\theta_{sep}|$, OPTRVR, S/N=40db, ρ=0.8

NRUN = 2

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	-.975	-.186E-02	.126E-02	-1.49	.528E-02	-.223E-02	-.319E-01	-.348E-01
	-.125	-5.19	-.391E-02	.170E-02	13.9	-.348E-01	-.584E-02	.521E-03	-.256
	.125	-5.23	.133E-02	-.686E-02	17.5	.555E-01	.163E-01	.592E-01	-1.61
	.375	-.669	.153E-02	.137E-02	-2.22	-.950E-02	-.211E-02	-.274E-01	-.192E-01
	.625	-.117	.829E-03	.894E-03	-.634	-.560E-02	-.223E-02	-.190E-01	.302E-02
	.875	.310E-01	-.188E-04	.660E-04	-.649E-01	-.927E-03	-.723E-03	-.663E-02	-.936E-02
	1.13	.154E-01	-.778E-04	-.563E-04	.934E-02	.208E-03	-.318E-03	.683E-03	-.635E-01
	1.38	.202E-01	-.119E-03	-.163E-04	.706E-02	.578E-04	-.456E-03	.101E-02	-.746E-01
	1.63	.156E-01	-.740E-04	.460E-04	.843E-02	.658E-04	-.503E-03	.107E-01	-.644E-01
	1.88	.106E-01	.526E-04	.557E-04	.102E-01	.508E-04	-.455E-03	.374E-02	-.447E-01
ERMS:	-.375	4.11	.820E-02	.221E-01	4.37	.183E-01	.354E-01	.796E-01	.482
	-.125	14.5	.111E-01	.199E-01	16.6	.415E-01	.346E-01	.174	.818
	.125	21.6	.202E-01	.278E-01	24.4	.831E-01	.669E-01	.513	4.82
	.375	3.44	.721E-02	.218E-01	3.97	.169E-01	.305E-01	.585E-01	.430
	.625	1.09	.434E-02	.199E-01	1.29	.110E-01	.270E-01	.389E-01	.306
	.875	.404	.262E-02	.168E-01	.449	.548E-02	.218E-01	.315E-01	.311
	1.13	.241	.187E-02	.153E-01	.243	.371E-02	.190E-01	.370E-01	.404
	1.38	.180	.152E-02	.147E-01	.187	.281E-02	.179E-01	.292E-01	.512
	1.63	.161	.132E-02	.142E-01	.177	.242E-02	.179E-01	.430E-01	.473
	1.88	.160	.124E-02	.136E-01	.179	.219E-02	.176E-01	.414E-01	.522
ESTD:	-.375	3.99	.799E-02	.221E-01	4.11	.176E-01	.353E-01	.729E-01	.481
	-.125	13.6	.104E-01	.198E-01	9.19	.226E-01	.341E-01	.174	.777
	.125	20.9	.202E-01	.269E-01	16.9	.618E-01	.649E-01	.510	4.54
	.375	3.37	.705E-02	.217E-01	3.29	.139E-01	.304E-01	.516E-01	.429
	.625	1.08	.426E-02	.199E-01	1.12	.951E-02	.269E-01	.339E-01	.306
	.875	.402	.262E-02	.168E-01	.444	.540E-02	.218E-01	.308E-01	.311
	1.13	.240	.186E-02	.153E-01	.243	.370E-02	.190E-01	.370E-01	.399
	1.38	.179	.152E-02	.147E-01	.187	.281E-02	.179E-01	.292E-01	.506
	1.63	.160	.132E-02	.142E-01	.177	.242E-02	.179E-01	.417E-01	.468
	1.88	.160	.123E-02	.136E-01	.178	.219E-02	.176E-01	.413E-01	.520

Table 3.3 Error Statistics vs θ_{sep}, OPTRVR, S/N=40db, ρ=0.5

NRUN = 3

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRODT	B	WSC
EMEAN:	-.375	-1.29	-.740E-02	.943E-03	-.876	.446E-02	-.117E-02	-.695E-01	-.145
	-.125	-4.30	-.855E-02	.388E-02	9.89	-.598E-01	-.931E-02	.441E-01	-.324
	.125	6.01	-.225E-01	-.342E-02	-1.29	-.358E-02	.357E-02	.612E-01	-.214
	.375	-.670	.364E-02	.991E-03	-1.40	-.114E-01	-.873E-03	-.569E-01	-.126
	.625	-.163	.228E-02	.997E-03	-.368	-.534E-02	-.217E-02	-.391E-01	-.902E-02
	.875	.346E-01	-.959E-03	.110E-02	-.302E-01	-.988E-04	-.181E-02	.119E-01	-.640E-01
	1.13	.260E-01	-.932E-03	.499E-03	.264E-02	.407E-03	-.759E-03	.128E-01	-.133
	1.38	.147E-01	-.666E-03	.499E-03	.466E-02	.535E-03	-.743E-03	.827E-02	-.158
	1.63	.599E-02	-.361E-03	.461E-03	.232E-02	.425E-03	-.804E-03	.963E-02	-.222
	1.88	-.263E-03	.727E-04	.204E-03	.111E-02	.191E-03	-.504E-03	.132E-01	-.259
ERMS:	-.375	2.79	.179E-01	.301E-01	2.82	.237E-01	.358E-01	.145	.647
	-.125	9.81	.216E-01	.237E-01	11.3	.720E-01	.316E-01	.234	.864
	.125	7.00	.310E-01	.264E-01	5.70	.224E-01	.291E-01	.205	.715
	.375	2.32	.157E-01	.306E-01	2.60	.214E-01	.309E-01	.127	.675
	.625	.956	.125E-01	.322E-01	1.01	.166E-01	.310E-01	.136	.517
	.875	.378	.726E-02	.231E-01	.371	.845E-02	.241E-01	.633E-01	.510
	1.13	.228	.493E-02	.204E-01	.222	.617E-02	.218E-01	.596E-01	.565
	1.38	.175	.398E-02	.202E-01	.183	.457E-02	.200E-01	.541E-01	.715
	1.63	.159	.340E-02	.201E-01	.174	.411E-02	.207E-01	.740E-01	.776
	1.88	.160	.326E-02	.200E-01	.177	.370E-02	.202E-01	.115	.873
ESTD:	-.375	2.47	.163E-01	.301E-01	2.68	.233E-01	.358E-01	.127	.631
	-.125	8.82	.198E-01	.234E-01	5.55	.402E-01	.302E-01	.230	.861
	.125	3.60	.213E-01	.262E-01	5.55	.221E-01	.289E-01	.196	.683
	.375	2.23	.153E-01	.306E-01	2.19	.181E-01	.308E-01	.114	.663
	.625	.942	.123E-01	.322E-01	.944	.157E-01	.309E-01	.985E-01	.517
	.875	.376	.726E-02	.231E-01	.370	.845E-02	.240E-01	.622E-01	.506
	1.13	.227	.484E-02	.204E-01	.222	.615E-02	.218E-01	.582E-01	.570
	1.38	.174	.393E-02	.202E-01	.183	.453E-02	.200E-01	.535E-01	.697
	1.63	.159	.338E-02	.201E-01	.174	.409E-02	.207E-01	.734E-01	.743
	1.88	.160	.326E-02	.200E-01	.177	.370E-02	.202E-01	.114	.833

Table 3.4 Error Statistics vs θ_{sep}, OPTRVR, S/N=30db, ρ=0.8

NRUN =

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	-.483	-.214E-02	.214E-02	-.428	-.250E-02	-.733E-02	-.638E-01	-.143
	-.125	-1.06	.182E-02	.404E-02	7.02	-.673E-01	-.887E-02	.919E-01	-.453
	.125	5.74	-.388E-01	-.863E-02	-.773	-.522E-01	-.350E-01	.183	-.165
	.375	.375	-.247E-02	.252E-03	-1.43	-.173E-01	.239E-02	-.517E-01	-.186
	.625	.470E-01	-.363E-03	.656E-03	-.263	-.445E-02	-.949E-03	-.192E-01	-.749E-01
	.875	.162E-01	-.500E-03	.964E-03	.139E-01	.213E-02	-.243E-02	.873E-02	-.725E-01
	1.13	.138E-01	-.573E-03	.767E-03	-.239E-02	.112E-02	-.143E-02	.909E-02	-.198
	1.38	.143E-01	-.617E-03	.482E-03	.219E-03	.158E-02	-.109E-02	.932E-02	-.185
	1.63	.641E-02	-.343E-03	.292E-03	-.613E-02	.105E-02	-.109E-02	.150E-01	-.314
	1.88	.711E-03	.383E-04	.128E-03	-.801E-02	.293E-03	-.293E-03	.230E-01	-.300
ERMS:	-.375	2.40	.169E-01	.277E-01	2.68	.420E-01	.474E-01	.174	.757
	-.125	7.58	.235E-01	.302E-01	7.75	.770E-01	.300E-01	.422	1.12
	.125	7.83	.582E-01	.331E-01	5.08	.110	.748E-01	.444	.876
	.375	2.05	.145E-01	.276E-01	2.55	.329E-01	.353E-01	.146	.733
	.625	.892	.108E-01	.290E-01	1.05	.275E-01	.415E-01	.100	.611
	.875	.359	.694E-02	.229E-01	.383	.141E-01	.292E-01	.585E-01	.560
	1.13	.223	.498E-02	.207E-01	.229	.994E-02	.255E-01	.632E-01	.756
	1.38	.173	.383E-02	.198E-01	.186	.697E-02	.228E-01	.619E-01	.747
	1.63	.158	.338E-02	.200E-01	.176	.625E-02	.239E-01	.939E-01	.915
	1.88	.159	.326E-02	.200E-01	.178	.550E-02	.226E-01	.999E-01	.926
ESTD:	-.375	2.35	.168E-01	.276E-01	2.65	.419E-01	.468E-01	.162	.743
	-.125	7.51	.234E-01	.300E-01	3.30	.374E-01	.286E-01	.412	1.03
	.125	5.32	.433E-01	.320E-01	5.02	.963E-01	.661E-01	.405	.860
	.375	2.01	.143E-01	.276E-01	2.11	.280E-01	.353E-01	.137	.709
	.625	.891	.108E-01	.290E-01	1.02	.271E-01	.415E-01	.986E-01	.606
	.875	.359	.692E-02	.229E-01	.383	.140E-01	.291E-01	.578E-01	.555
	1.13	.223	.495E-02	.207E-01	.229	.987E-02	.255E-01	.626E-01	.730
	1.38	.172	.378E-02	.198E-01	.186	.679E-02	.228E-01	.612E-01	.724
	1.63	.158	.337E-02	.200E-01	.176	.616E-02	.239E-01	.927E-01	.860
	1.88	.159	.326E-02	.200E-01	.178	.549E-02	.226E-01	.972E-01	.876

Table 3.5 Error Statistics vs θ_{sep}, OPTRVR, S/N=30db, ρ=0.5

NRUN = 5

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	.514	.269E-01	.108E-01	1.17	-.586E-01	-.197E-01	.727E-01	-.323
	-.125	.256	.247E-01	.140E-01	3.47	-.100	-.200E-01	.220	-.499
	.125	3.20	-.747E-01	-.174E-01	1.09	.738E-02	-.112E-02	.484	-.450
	.375	1.41	-.451E-01	-.131E-01	.499	.262E-01	.177E-01	.119	-.325
	.625	.662	-.459E-01	.204E-02	.841	.786E-01	.142E-02	.389	-.516
	.875	.106	-.963E-02	.606E-03	.420E-01	.867E-02	.292E-03	.330E-01	-.296
	1.13	.259E-01	-.543E-02	.229E-02	-.128E-01	.398E-02	-.231E-02	.332E-01	-.376
	1.38	-.139E-02	-.318E-02	.165E-02	-.155E-01	.439E-02	-.127E-02	.364E-01	-.378
	1.63	-.164E-01	-.150E-02	.102E-02	-.262E-01	.313E-02	-.461E-03	.485E-01	-.523
	1.88	-.256E-01	.324E-03	.706E-03	-.316E-01	.504E-04	-.428E-03	.638E-01	-.610
ERMS:	-.375	1.62	.521E-01	.330E-01	1.70	.783E-01	.385E-01	.312	.964
	-.125	3.55	.515E-01	.345E-01	3.83	.119	.342E-01	.612	1.16
	.125	3.86	.958E-01	.433E-01	2.63	.366E-01	.385E-01	.595	1.09
	.375	1.76	.596E-01	.375E-01	1.51	.603E-01	.416E-01	.245	.835
	.625	1.05	.686E-01	.683E-01	1.25	.113	.964E-01	.647	1.13
	.875	.307	.191E-01	.259E-01	.320	.247E-01	.364E-01	.170	.926
	1.13	.212	.148E-01	.281E-01	.219	.192E-01	.344E-01	.139	.869
	1.38	.165	.102E-01	.256E-01	.181	.126E-01	.260E-01	.118	.821
	1.63	.156	.848E-02	.257E-01	.175	.111E-01	.281E-01	.146	1.12
	1.88	.159	.845E-02	.267E-01	.177	.968E-02	.269E-01	.228	1.27
ESTD:	-.375	1.54	.446E-01	.312E-01	1.23	.519E-01	.331E-01	.303	.908
	-.125	3.54	.451E-01	.315E-01	1.62	.635E-01	.277E-01	.571	1.04
	.125	2.16	.600E-01	.396E-01	2.40	.358E-01	.385E-01	.346	.992
	.375	1.04	.391E-01	.351E-01	1.43	.543E-01	.377E-01	.215	.769
	.625	.812	.510E-01	.683E-01	.924	.805E-01	.964E-01	.517	1.00
	.875	.288	.165E-01	.259E-01	.318	.231E-01	.364E-01	.167	.878
	1.13	.211	.137E-01	.280E-01	.219	.188E-01	.343E-01	.135	.783
	1.38	.165	.974E-02	.255E-01	.181	.118E-01	.260E-01	.112	.729
	1.63	.155	.835E-02	.257E-01	.173	.106E-01	.281E-01	.137	.994
	1.88	.157	.844E-02	.267E-01	.174	.968E-02	.269E-01	.219	1.12

Table 3.6 Error Statistics vs θ_{sep}, OPTRVR, S/N=20db, ρ=0.8

NRUN = 6

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	.746	.373E-01	.173E-01	1.23	-.137	-.478E-01	.175	-.501
	-.125	.777	.291E-01	.107E-01	2.06	-.724E-01	-.285E-02	.323	-.468
	.125	2.80	-.670E-01	-.159E-01	.739	-.143E-01	-.224E-01	.922	-.304
	.375	1.19	-.405E-01	-.588E-02	.164	.194E-01	.189E-01	.182	-.535
	.625	.369E-01	-.148E-01	-.169E-02	1.16	.163	.164E-01	.352	-.586
	.875	.125	-.120E-01	-.143E-02	.133	.327E-01	.440E-02	.131	-.318
	1.13	.203E-01	-.469E-02	.251E-02	-.226E-01	.868E-02	-.277E-02	.379E-01	-.433
	1.38	-.381E-02	-.269E-02	.151E-02	-.332E-01	.744E-02	-.245E-02	.506E-01	-.495
	1.63	-.157E-01	-.146E-02	.247E-03	-.479E-01	.520E-02	-.809E-03	.717E-01	-.659
	1.88	-.242E-01	.179E-03	.698E-03	-.502E-01	.408E-03	-.321E-03	.784E-01	-.667
ERMS:	-.375	1.58	.619E-01	.409E-01	1.79	.191	.665E-01	.423	1.19
	-.125	2.41	.446E-01	.308E-01	2.45	.938E-01	.365E-01	.644	1.04
	.125	4.05	.987E-01	.477E-01	1.30	.864E-01	.621E-01	1.32	.920
	.375	1.68	.577E-01	.430E-01	1.34	.769E-01	.547E-01	.312	1.18
	.625	1.00	.465E-01	.564E-01	1.30	.186	.897E-01	.504	1.19
	.875	.280	.195E-01	.284E-01	.309	.429E-01	.360E-01	.215	.863
	1.13	.201	.140E-01	.267E-01	.221	.315E-01	.442E-01	.148	.908
	1.38	.166	.101E-01	.254E-01	.187	.215E-01	.338E-01	.151	1.07
	1.63	.155	.842E-02	.252E-01	.182	.164E-01	.321E-01	.195	1.33
	1.88	.158	.842E-02	.264E-01	.182	.154E-01	.325E-01	.277	1.27
ESTD:	-.375	1.39	.495E-01	.371E-01	1.30	.133	.462E-01	.385	1.08
	-.125	2.28	.337E-01	.289E-01	1.33	.596E-01	.364E-01	.557	.934
	.125	2.93	.724E-01	.450E-01	1.07	.852E-01	.579E-01	.942	.869
	.375	1.19	.411E-01	.426E-01	1.33	.744E-01	.514E-01	.254	1.05
	.625	1.00	.441E-01	.564E-01	.580	.888E-01	.882E-01	.361	1.03
	.875	.250	.154E-01	.283E-01	.279	.278E-01	.357E-01	.171	.802
	1.13	.200	.132E-01	.266E-01	.220	.303E-01	.441E-01	.144	.798
	1.38	.166	.974E-02	.253E-01	.184	.201E-01	.337E-01	.142	.945
	1.63	.155	.830E-02	.252E-01	.176	.155E-01	.321E-01	.181	1.15
	1.88	.156	.842E-02	.264E-01	.175	.154E-01	.325E-01	.266	1.08

Table 3.7 Error Statistics vs θ_{sep}, OPTRVR, S/N=20db, ρ=0.5

NRUN = 7

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	.935	.112	.292E-01	.716	-.126	-.501E-01	.460	-.757
	-.125	1.23	.937E-01	.300E-01	1.27	-.118	-.329E-01	.735	-.630
	.125	1.03	-.102	-.403E-01	1.73	.637E-01	.130E-01	1.03	-.511
	.375	.957	-.130	-.347E-01	.720	.989E-01	.588E-01	.458	-.862
	.625	.422	-.675E-01	-.601E-02	.540	.103	.173E-01	.414	-.686
	.875	.153	-.305E-01	-.395E-02	.716E-01	.284E-01	.606E-02	.118	-.603
	1.13	-.416E-02	-.929E-02	.357E-02	-.317E-01	.108E-01	-.362E-02	.628E-01	-.661
	1.38	.283E-01	-.170E-01	-.114E-01	.130E-01	.216E-01	.175E-01	-.349E-01	-35.9
	1.63	-.412E-01	-.268E-02	.247E-02	-.548E-01	.744E-02	-.277E-02	.107	-.698
	1.88	-.502E-01	.183E-02	.547E-03	-.590E-01	.567E-03	-.657E-04	.108	-1.05
EPMS:	-.375	1.30	.146	.575E-01	1.17	.178	.844E-01	.658	1.56
	-.125	1.76	.129	.555E-01	1.76	.167	.585E-01	.950	1.56
	.125	2.19	.177	.784E-01	1.93	.849E-01	.407E-01	1.36	1.06
	.375	1.66	.208	.102	1.39	.199	.137	.816	1.66
	.625	.836	.110	.766E-01	.784	.141	.888E-01	.691	1.24
	.875	.293	.451E-01	.389E-01	.263	.482E-01	.468E-01	.263	1.26
	1.13	.201	.310E-01	.416E-01	.203	.373E-01	.479E-01	.253	1.26
	1.38	.202	.590E-01	.644E-01	.204	.820E-01	.845E-01	.794	36.6
	1.63	.154	.164E-01	.321E-01	.176	.218E-01	.357E-01	.286	1.30
	1.88	.157	.152E-01	.327E-01	.176	.175E-01	.335E-01	.419	1.83
ESTD:	-.375	.900	.932E-01	.495E-01	.930	.126	.679E-01	.471	1.37
	-.125	1.27	.889E-01	.466E-01	1.23	.118	.483E-01	.602	1.43
	.125	1.93	.145	.672E-01	.864	.561E-01	.386E-01	.885	.926
	.375	1.35	.163	.963E-01	1.19	.173	.124	.675	1.42
	.625	.722	.864E-01	.784E-01	.568	.963E-01	.871E-01	.554	1.04
	.875	.249	.332E-01	.387E-01	.253	.389E-01	.464E-01	.235	1.11
	1.13	.201	.296E-01	.414E-01	.200	.356E-01	.477E-01	.245	1.08
	1.38	.200	.565E-01	.633E-01	.203	.791E-01	.827E-01	.793	7.06
	1.63	.148	.161E-01	.320E-01	.167	.205E-01	.356E-01	.265	1.10
	1.88	.149	.151E-01	.327E-01	.166	.175E-01	.335E-01	.405	1.50

Table 3.8 Error Statistics vs θ_{sep}, OPTRVR, S/N=14 db, ρ=0.8

NRUN = 8

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	-.375	.674	.756E-01	.226E-01	.444	-.147	-.514E-01	.358	-.807
	-.125	.803	.570E-01	.141E-01	.872	-.813E-01	-.140E-01	.838	-.210
	.125	1.50	-.158	-.577E-01	.248	-.168E-01	-.195E-01	.716	-1.28
	.375	1.64	-.168	-.431E-01	-.149	-.136	-.561E-01	.784	-.863
	.625	.221	-.468E-01	-.112E-01	.611	.179	.440E-01	.477	-.889
	.875	.128	-.293E-01	-.138E-02	.104	.661E-01	.837E-02	.225	-.824
	1.13	.801E-02	-.126E-01	.323E-02	-.198E-01	.280E-01	-.214E-03	.125	-.760
	1.38	-.299E-01	-.493E-02	.305E-02	-.579E-01	.125E-01	-.581E-02	.124	-.666
	1.63	-.429E-01	-.113E-02	.121E-02	-.651E-01	.894E-02	-.294E-02	.108	-.815
	1.88	-.465E-01	.200E-02	.323E-03	-.668E-01	.313E-02	-.696E-03	.124	-1.03
ERMS:	-.375	.966	.975E-01	.506E-01	.906	.228	.864E-01	.630	1.46
	-.125	1.36	.872E-01	.386E-01	1.17	.130	.505E-01	1.21	.848
	.125	2.23	.258	.985E-01	.939	.152	.815E-01	.990	3.48
	.375	2.19	.263	.116	.747	.231	.990E-01	1.21	1.48
	.625	.788	.917E-01	.764E-01	.751	.220	.108	.723	1.51
	.875	.284	.456E-01	.443E-01	.302	.103	.757E-01	.431	1.43
	1.13	.194	.305E-01	.382E-01	.210	.683E-01	.697E-01	.314	1.29
	1.38	.161	.206E-01	.335E-01	.165	.493E-01	.569E-01	.309	1.23
	1.63	.154	.155E-01	.310E-01	.186	.323E-01	.423E-01	.346	1.39
	1.88	.157	.159E-01	.333E-01	.186	.298E-01	.433E-01	.522	1.76
ESTD:	-.375	.692	.615E-01	.453E-01	.789	.175	.695E-01	.518	1.21
	-.125	1.10	.660E-01	.359E-01	.774	.101	.486E-01	.869	.821
	.125	1.65	.203	.798E-01	.905	.151	.791E-01	.684	3.23
	.375	1.46	.201	.108	.732	.187	.816E-01	.915	1.20
	.625	.756	.789E-01	.776E-01	.436	.128	.986E-01	.542	1.22
	.875	.253	.350E-01	.443E-01	.283	.790E-01	.752E-01	.368	1.16
	1.13	.194	.277E-01	.381E-01	.209	.623E-01	.697E-01	.288	1.04
	1.38	.158	.200E-01	.333E-01	.176	.477E-01	.566E-01	.283	1.04
	1.63	.148	.155E-01	.309E-01	.174	.311E-01	.422E-01	.329	1.13
	1.88	.149	.157E-01	.333E-01	.173	.296E-01	.433E-01	.508	1.43

Table 3.9 Error Statistics vs θ_{sep}, OPTRVR, S/N=14db, ρ=0.5

NRUN = 3

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT
EMEAN:	-.375	-11.9	-.101	.234E-02	3.80	-.305E-01	-.918E-02
	-.125	-9.05	-.375E-01	.955E-02	17.1	-.971E-01	-.417E-01
	.125	-9.53	.271E-01	-.175E-01	13.7	.415E-01	.704E-01
	.375	-9.52	.795E-01	-.793E-03	5.77	.531E-01	.447E-02
	.625	-2.01	.192E-01	-.178E-01	.767	.402E-01	.188E-01
	.875	.310	-.332E-02	-.399E-02	.328	.748E-02	.611E-02
	1.13	.222	-.486E-02	-.100E-02	.225	.673E-02	.576E-03
	1.38	.630E-01	-.113E-02	.233E-03	.570E-01	.200E-02	-.632E-03
	1.63	-.299E-01	.753E-03	.201E-03	.375E-02	-.642E-03	-.617E-03
	1.88	-.166E-01	.911E-03	.149E-03	-.529E-02	-.114E-02	-.391E-03
ERMS:	-.375	16.0	.119	.200	5.01	.863E-01	.189
	-.125	15.1	.562E-01	.100E+00	18.3	.201	.246
	.125	16.9	.624E-01	.119	16.4	.153	.255
	.375	11.5	.945E-01	.762E-01	6.66	.102	.146
	.625	6.51	.990E-01	.749E-01	4.97	.116	.108
	.875	1.76	.518E-01	.595E-01	2.23	.789E-01	.861E-01
	1.13	.705	.206E-01	.522E-01	.686	.227E-01	.497E-01
	1.38	.372	.787E-02	.236E-01	.408	.972E-02	.276E-01
	1.63	.271	.801E-02	.193E-01	.310	.116E-01	.271E-01
	1.88	.191	.820E-02	.184E-01	.199	.108E-01	.224E-01
ESTD:	-.375	10.7	.629E-01	.200	3.26	.807E-01	.188
	-.125	12.1	.419E-01	.995E-01	6.64	.176	.242
	.125	14.0	.563E-01	.118	8.95	.147	.245
	.375	6.54	.510E-01	.762E-01	3.32	.873E-01	.146
	.625	6.19	.971E-01	.728E-01	4.91	.109	.107
	.875	1.73	.517E-01	.594E-01	2.20	.785E-01	.859E-01
	1.13	.669	.200E-01	.522E-01	.648	.217E-01	.497E-01
	1.38	.366	.779E-02	.236E-01	.404	.952E-02	.276E-01
	1.63	.269	.798E-02	.193E-01	.310	.116E-01	.271E-01
	1.88	.190	.815E-02	.184E-01	.199	.108E-01	.224E-01

Table 3.10 Error Statistics vs θ_{sep}, SUBOPT, S/N=30db, ρ=0.8

NRUN = 5

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT
EMEAN:	-.375	4.73	.254	.389E-01	-.802	-.332E-01	-.755E-01
	-.125	5.61	-.280	-.464E-01	-.568	.198	.421E-01
	.125	-1.55	-.207E-03	-.301E-01	7.03	.577	.102
	.375	4.06	-.215	-.338E-01	-.105	.488E-01	.734E-01
	.625	-.260	-.182E-01	-.286E-01	2.81	.233	.377E-01
	.875	.432	-.392E-01	-.112E-01	.879	.905E-01	.157E-01
	1.13	.244	-.240E-01	-.259E-02	.228	.326E-01	.416E-02
	1.38	.388E-01	-.686E-02	.618E-03	.300E-01	.106E-01	.114E-03
	1.63	-.291E-01	.990E-03	.129E-02	-.308E-01	.209E-02	-.950E-03
	1.88	-.341E-01	.289E-02	.985E-03	-.429E-01	-.273E-02	-.964E-03
ERMS:	-.375	4.81	.280	.169	3.52	.200	.181
	-.125	5.94	.294	.109	3.75	.210	.749E-01
	.125	5.45	.895E-01	.969E-01	7.07	.640	.171
	.375	4.29	.248	.149	2.94	.188	.178
	.625	2.43	.128	.114	2.89	.249	.106
	.875	.920	.791E-01	.666E-01	1.00	.116	.862E-01
	1.13	.349	.416E-01	.391E-01	.356	.623E-01	.605E-01
	1.38	.178	.202E-01	.305E-01	.195	.273E-01	.369E-01
	1.63	.174	.108E-01	.267E-01	.213	.143E-01	.318E-01
	1.88	.171	.121E-01	.263E-01	.203	.171E-01	.289E-01
ESTD:	-.375	.879	.117	.165	3.43	.197	.164
	-.125	1.93	.908E-01	.986E-01	3.70	.711E-01	.619E-01
	.125	5.23	.895E-01	.921E-01	.762	.276	.137
	.375	1.38	.123	.145	2.94	.182	.163
	.625	2.42	.127	.110	.661	.879E-01	.985E-01
	.875	.813	.687E-01	.657E-01	.478	.732E-01	.847E-01
	1.13	.250	.340E-01	.390E-01	.274	.531E-01	.604E-01
	1.38	.174	.190E-01	.305E-01	.193	.252E-01	.369E-01
	1.63	.171	.107E-01	.267E-01	.211	.141E-01	.318E-01
	1.88	.167	.117E-01	.263E-01	.198	.169E-01	.289E-01

Table 3.11 Error Statistics vs θ_{sep}, SUBOPT, S/N=20db, ρ=0.8

NRUN = 6

QTY	THESEP	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT
EMEAN:	-.375	-.743	-.848E-02	.260E-01	3.90	-.811	-.165
	-.125	-.712	.118E-02	.155E-01	4.43	-.245	-.361E-01
	.125	-.690	.247E-02	-.122E-01	4.48	.380	.781E-01
	.375	-.743	.137E-01	-.245E-01	3.99	.842	.152
	.625	-.547	.118E-01	-.194E-01	2.58	.418	.659E-01
	.875	.150E-01	-.108E-01	-.868E-02	1.00	.178	.308E-01
	1.13	.112	-.135E-01	-.199E-02	.252	.653E-01	.111E-01
	1.38	.179E-01	-.462E-02	.781E-03	.258E-01	.210E-01	.161E-02
	1.63	-.294E-01	.197E-02	.103E-02	-.432E-01	.648E-02	-.168E-02
	1.88	-.292E-01	.290E-02	.953E-03	-.622E-01	-.281E-02	-.302E-02
ERMS:	-.375	3.05	.119	.100	3.93	.868	.271
	-.125	3.43	.553E-01	.552E-01	4.45	.305	.849E-01
	.125	3.44	.480E-01	.491E-01	4.50	.444	.102
	.375	3.05	.114	.963E-01	4.02	.901	.274
	.625	2.28	.121	.100	2.63	.422	.143
	.875	.948	.718E-01	.635E-01	1.03	.201	.123
	1.13	.336	.372E-01	.367E-01	.352	.106	.964E-01
	1.38	.170	.196E-01	.286E-01	.192	.500E-01	.569E-01
	1.63	.154	.112E-01	.264E-01	.215	.210E-01	.369E-01
	1.88	.156	.917E-02	.254E-01	.210	.285E-01	.370E-01
ESTD:	-.375	2.96	.119	.966E-01	.497	.311	.215
	-.125	3.35	.553E-01	.530E-01	.431	.182	.769E-01
	.125	3.37	.479E-01	.476E-01	.409	.229	.656E-01
	.375	2.96	.113	.931E-01	.512	.322	.228
	.625	2.21	.120	.983E-01	.495	.561E-01	.127
	.875	.947	.710E-01	.629E-01	.258	.937E-01	.120
	1.13	.317	.346E-01	.366E-01	.246	.841E-01	.958E-01
	1.38	.169	.191E-01	.286E-01	.190	.454E-01	.568E-01
	1.63	.151	.110E-01	.264E-01	.210	.200E-01	.369E-01
	1.88	.153	.870E-02	.254E-01	.201	.284E-01	.369E-01

Table 3.12 Error Statistics vs θ_{sep}, SUBOPT, S/N=20db, ρ=0.5

in part to the relatively large value of GQGT (8.8) used, $\tilde{\simeq}(2.7r/s)^2$ (in AZ). This value was selected to facilitate tracking a varying ω_{sc} under the condition that it is changing at the maximum rate at which it can be tracked in the present structure, based on angle function rep. rate considerations (recall (3.135)) (i.e. without extracting its derivative and integrating (as in the 6D LOE design)). This was a practical consideration that penalized some the performance in the steady state (when ω_{sc} was not changing).

Figure 3.8 summarizes a study of the SUBOPT receiver; it shows that

1. In the worst cast wrt β, i.e. β=180°, the constraints on the estimate helped for small θ_{sep} and did not influence the performance for larger θ_{sep}; and that
2. Averaging over noise effects and β simultaneously by using a small non-zero $\dot{\beta}$ ($=\omega_{sc}$) to sweep β over a 2π interval seems adequate generally, with some error possibly arising for small θ_{sep}. This method of averaging over β was used generally.

Figure 3.9 and Tables 3.13 thru 3.16 following show the effects of a form of mismatch between the receiver design and its signal environment, specifically a mismatch in the presumed and actual values of the transmitting antenna beamwidth, (given the antenna selectivity functions, p, p_{MLS}, are otherwise identical). Much can be said about the necessity of tuning a high-performance signal processor to its signal environment, but the manifest RMSE sensitivity here to B_{RCVR}/B_{MLS} is nevertheless striking. The tables provide some insight, for example

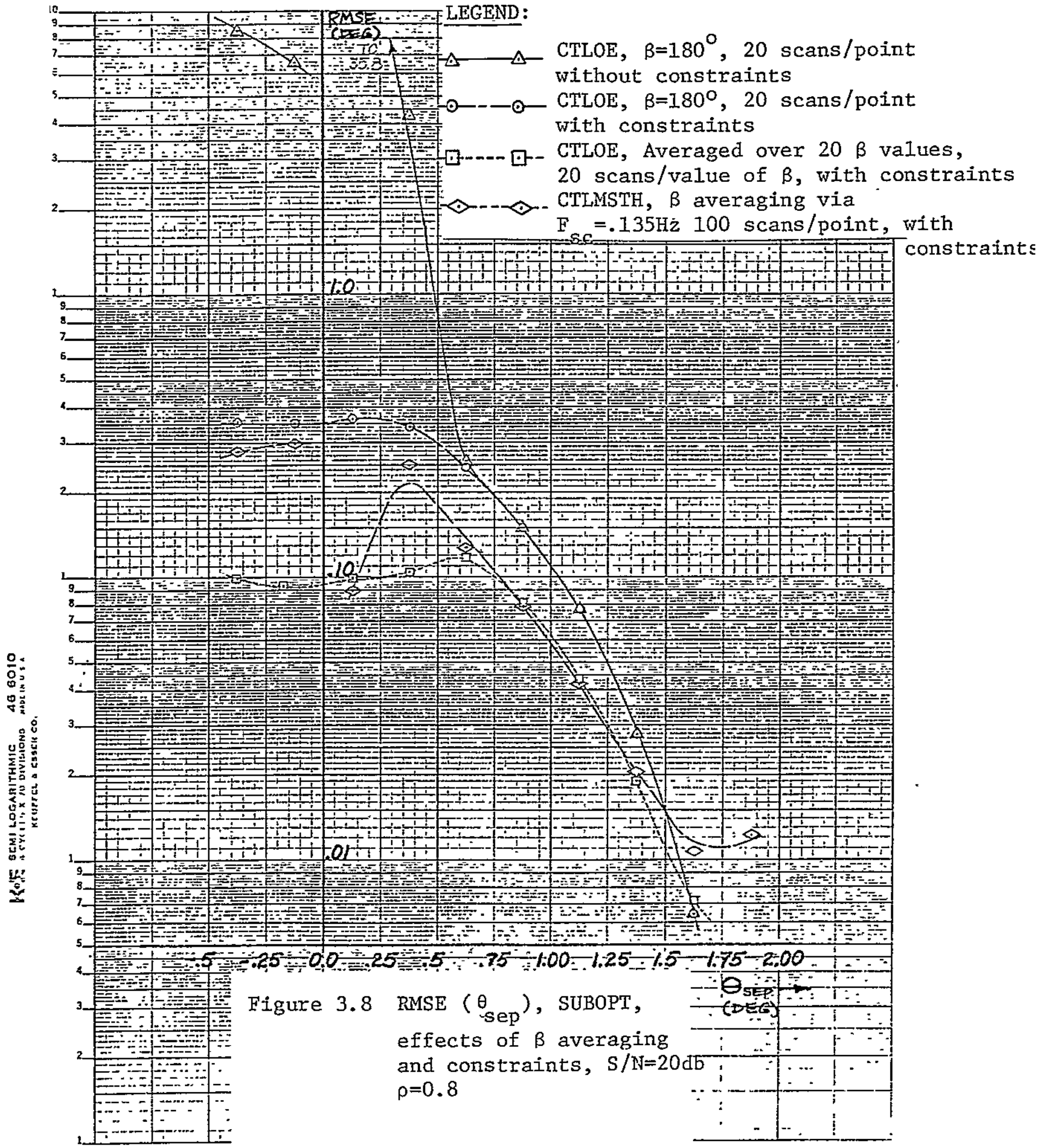


Figure 3.8 RMSE (θ_{sep}), SUBOPT, effects of β averaging and constraints, S/N=20db ρ=0.8

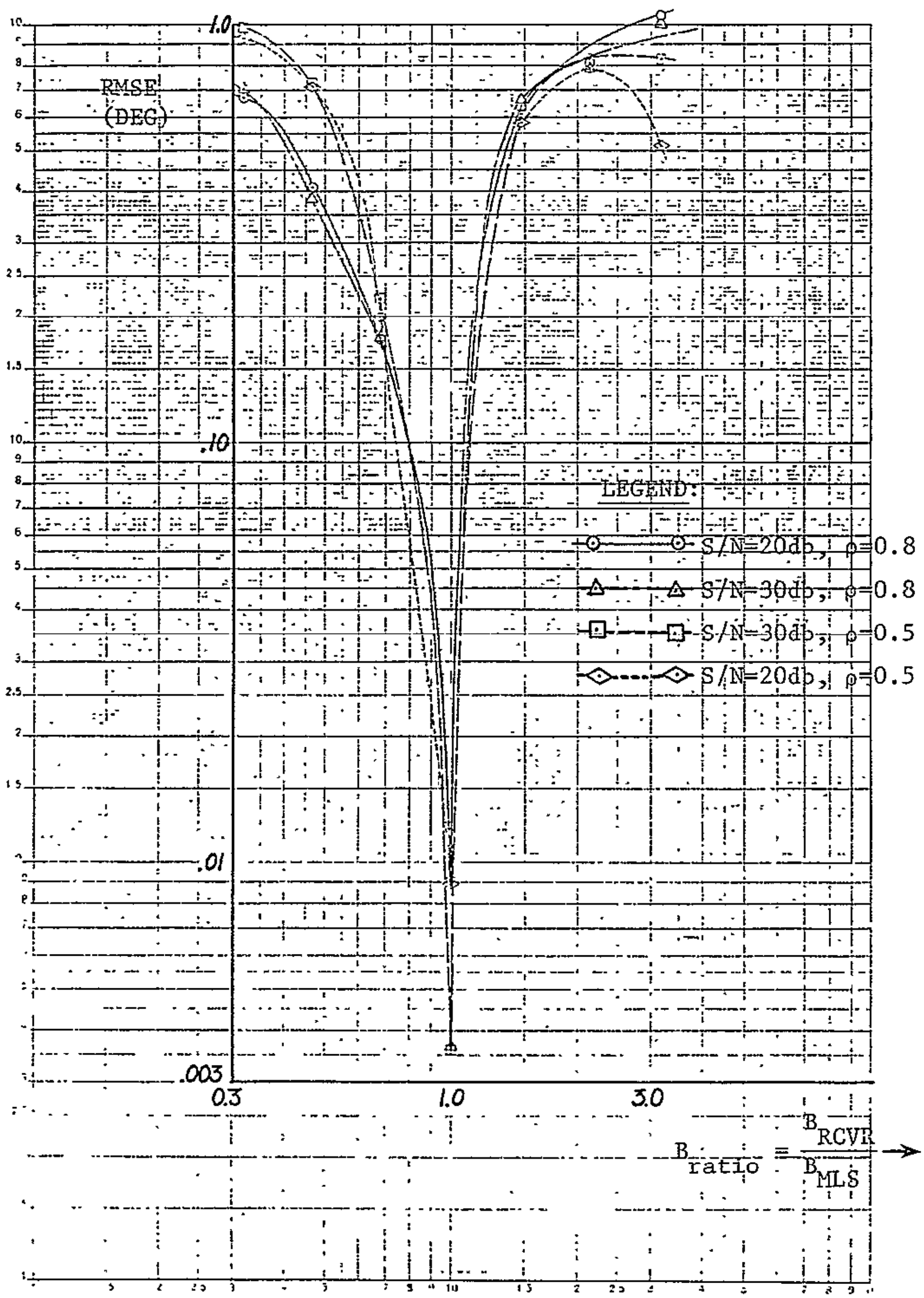


Figure 3.9 RMSE (B_{RCVR}/B_{MLS}), OPTRVR, $\theta_{sep}=1.5^{o}$

NRUN = 1

QTY	BRATIO	ALFA	THETA	THEDOT	ALFAR	THETAR	THPDOT	B	WSC
EMEAN:	.316	-3.75	-.553	-.130	-4.76	.721E-01	.335E-01	.734E-01	-20.6
	.464	-2.47	-.345	-.599E-01	-3.20	.252	.101	.598	-16.6
	.681	-1.27	-.151	-.221E-01	-1.67	.906E-01	.382E-01	.662	-5.05
	1.00	-.961E-02	-.234E-02	.127E-02	-.210E-01	.392E-02	-.110E-02	.390E-01	-.456
	1.47	-49.2	.636	.646E-01	-49.5	-.687	-.748E-01	-1.43	.728
	2.15	-6.92	.798	.774E-01	-1.24	-.422	-.118	-1.11	-.289
	3.16	.587	1.13	.170	4.94	1.11	.189	-1.14	-2.90
ERMS:	.316	5.87	.676	.463	7.39	.208	.352	1.18	24.1
	.464	3.49	.409	.337	4.82	.474	.452	1.08	19.7
	.681	1.48	.177	.114	2.16	.225	.196	1.06	6.29
	1.00	.158	.915E-02	.257E-01	.177	.114E-01	.261E-01	.124	.931
	1.47	55.2	.638	.171	55.5	.689	.186	1.68	.746
	2.15	8.25	.817	.235	4.35	.492	.156	1.81	1.33
	3.16	1.79	1.16	.317	5.15	1.48	.421	1.67	3.51
ESTD:	.316	4.51	.389	.445	5.65	.196	.351	1.18	12.5
	.464	2.47	.218	.331	3.60	.402	.441	.896	10.6
	.681	.768	.922E-01	.112	1.38	.206	.193	.823	3.75
	1.00	.158	.885E-02	.257E-01	.176	.107E-01	.261E-01	.118	.812
	1.47	24.8	.467E-01	.158	25.0	.538E-01	.171	.883	.161
	2.15	4.46	.174	.222	4.17	.252	.101	1.42	1.29
	3.16	1.69	.266	.267	1.43	.979	.376	1.21	1.98

Table 3.13 Error Statistics vs B_{ratio}, OPTRVR, S/N=20db, ρ=0.8, θ_{sep}=1.5^{o}

NRUN = 2

QTY	BRATIO	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	.316	-11.8	-.589	-.557E-01	-14.8	.296	.122	-.126	-18.9
	.464	-7.96	-.331	-.149E-02	-10.4	.232	.922E-02	.631E-01	-20.7
	.681	-4.04	-.158	-.157E-01	-5.24	.103	.181E-01	.774	-9.57
	1.00	.122E-01	-.634E-03	.493E-03	.398E-02	.654E-03	-.904E-03	.108E-01	-.173
	1.47	-294.	.661	.179E-01	-295.	-.715	-.193E-01	-1.46	.787
	2.15	-13.8	.770	.444E-01	6.62	-.546	-.142	-.960	-1.66
	3.16	14.8	-.110	.237	-.672E-01	-4.71	-.327	.150	-17.0
ERMS:	.316	18.5	.703	.510	22.9	.454	.515	1.26	22.1
	.464	11.0	.384	.358	15.0	.442	.437	1.15	24.1
	.681	4.68	.178	.116	6.71	.210	.194	1.08	10.5
	1.00	.164	.368E-02	.203E-01	.177	.440E-02	.206E-01	.608E-01	.712
	1.47	371.	.663	.101	372.	.716	.106	1.70	.793
	2.15	22.8	.781	.240	18.3	.588	.240	1.87	3.08
	3.16	32.5	1.06	.589	36.0	6.41	.626	1.29	22.4
ESTD:	.316	14.2	.384	.507	17.5	.345	.500	1.26	11.6
	.464	7.65	.193	.358	10.9	.377	.437	1.15	12.4
	.681	2.35	.806E-01	.115	4.19	.183	.194	.749	4.39
	1.00	.164	.362E-02	.203E-01	.177	.435E-02	.206E-01	.598E-01	.690
	1.47	226.	.391E-01	.990E-01	227.	.414E-01	.105	.859	.988E-01
	2.15	18.1	.135	.236	17.1	.218	.194	1.61	2.59
	3.16	28.9	1.05	.539	36.0	4.35	.534	1.28	14.5

Table 3.14 Error Statistics vs B_{ratio}, OPTRVR, S/N=30db, ρ=0.8, θ_{sep}=1.5°

NRUN = 3

QTY	BRATIO	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	.316	-7.05	-.856	-.136	-23.0	-.555	-.881E-01	.777E-02	-19.5
	.464	-1.93	-.654	-.917E-01	-16.5	-.561	-.624E-01	.474	-17.6
	.681	-2.33	-.207	-.198E-01	-6.15	-.129E-02	.294E-01	.815	-8.59
	1.00	.107E-01	-.524E-03	.486E-03	-.125E-02	.154E-02	-.130E-02	.142E-01	-.212
	1.47	-349.	.595	.746E-01	-351.	-.741	-.813E-01	-1.46	-.112
	2.15	-128.	.834	.481E-01	-121.	-.535	-.799E-01	-1.55	.373
	3.16	10.7	.772	.442E-01	11.0	1.35	-.286E-01	-.384	-14.1
ERMS:	.316	12.7	.983	.543	24.9	.757	.557	1.23	23.7
	.464	7.70	.729	.429	17.9	.697	.503	1.37	21.2
	.681	2.41	.210	.846E-01	8.05	.301	.301	1.15	9.29
	1.00	.163	.360E-02	.200E-01	.179	.678E-02	.239E-01	.750E-01	.793
	1.47	434.	.608	.210	436.	.763	.243	1.74	4.28
	2.15	146.	.837	.163	141.	.544	.155	1.80	.780
	3.16	11.4	.828	.442	11.5	2.22	.590	1.31	18.3
ESTD:	.316	10.6	.483	.525	9.60	.515	.550	1.23	13.5
	.464	7.45	.322	.419	6.85	.414	.500	1.28	11.9
	.681	.626	.326E-01	.823E-01	5.19	.301	.299	.816	3.55
	1.00	.163	.356E-02	.200E-01	.179	.660E-02	.239E-01	.737E-01	.764
	1.47	258.	.127	.197	259.	.180	.229	.947	4.28
	2.15	71.1	.691E-01	.156	71.6	.984E-01	.133	.917	.685
	3.16	3.94	.300	.440	3.22	1.76	.589	1.25	11.7

Table 3.15 Error Statistics vs B_{ratio}, OPTRVR, S/N=30db, ρ =0.5, $\theta_{sep}=1.5^{o}$

NRUN = 4

QTY	BRATIO	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	.316	-2.26	-.817	-.187	-7.08	-.522	-.167	.205	-18.0
	.464	-.699	-.634	-.162	-5.25	-.577	-.144	.945	-9.36
	.681	-.756	-.197	-.156E-01	-2.03	-.302E-01	.697E-01	.915	-4.57
	1.00	-.108E-01	-.198E-02	.981E-03	-.403E-01	.681E-02	-.163E-02	.575E-01	-.531
	1.47	-38.0	.581	.612E-01	-38.8	-.729	-.892E-01	-1.44	.683
	2.15	-9.21	.774	.889E-01	-7.39	-.368	-.660E-01	-1.42	-.144
	3.16	3.39	.485	.377E-01	4.16	-.965	-.249	-.608E-01	-29.2
ERMS:	.316	4.00	.933	.542	7.75	.717	.566	1.34	21.4
	.464	2.47	.709	.397	5.73	.704	.479	1.37	11.0
	.681	.603	.200	.727E-01	2.63	.312	.254	1.27	6.87
	1.00	.158	.891E-02	.250E-01	.182	.177E-01	.307E-01	.165	1.19
	1.47	41.5	.584	.150	42.3	.733	.204	1.70	.730
	2.15	10.7	.784	.214	9.15	.398	.841E-01	1.75	1.15
	3.16	3.70	.515	.302	4.33	1.09	.291	1.34	29.9
ESTD:	.316	3.30	.451	.508	3.17	.492	.541	1.32	11.6
	.464	2.36	.318	.362	2.29	.404	.456	.996	5.84
	.681	.264	.345E-01	.710E-01	1.68	.311	.245	.884	5.14
	1.00	.158	.869E-02	.250E-01	.177	.163E-01	.307E-01	.155	1.07
	1.47	16.7	.569E-01	.137	16.9	.728E-01	.183	.899	.257
	2.15	5.38	.121	.194	5.40	.150	.521E-01	1.01	1.14
	3.16	1.50	.171	.300	1.22	.500	.152	1.34	6.61

Table 3.16 Error Statistics vs B_{ratio}, OPTRVR, S/N=20db, $\rho=0.5$, $\theta_{sep}=1.5^{o}$

1. An estimation of α and α_R by integration of the signal, such as in done here, may be especially vulnerable to uncertainties in other parameters that affect signal energy, such as beam width; and

2. The errors in the θ and θ_R components are heavily biased, when $\beta_{ratio} \neq 1$, indicating loss of LOE validity, specifically wrt decoupling of estimates of errors.

Perhaps peak detection, maybe in conjunction with an integration approach, would provide more robust estimates of α and α_R. Smaller errors in the θ and θ_R components should then result, but the residual errors might be reduced further by having the nominal beam width at each air terminal coded and transmitted in the MLS preamble. Additional robustness and possibly some simplification advantages might result from use in the receiver of a p(·) function which doesn't model exactly any particular transmitting antenna selectivity function but does produce a best fit in some sense over the class of transmitting antenna selectivity functions to which the receiver is exposed. This is a problem area that needs further study.

Figure 3.10 presents an RMSE (F_{sc}) study for scalloping rates in the first lobe of the rep. rate sampled signal spectrum, i.e.

$$F_{sc} \leq \frac{13.5}{2} \text{ Hz} = 6.75 \text{ Hz} \qquad (3.32)$$

All the values of F_{sc} for which the RMSE was calculated, except 0.0 Hz, cause β to integrate an integral number of times around a 2π interval as the run progresses hence the results should be independent of the initial β. The RMSE value at 0.0 Hz is probably dependent upon the

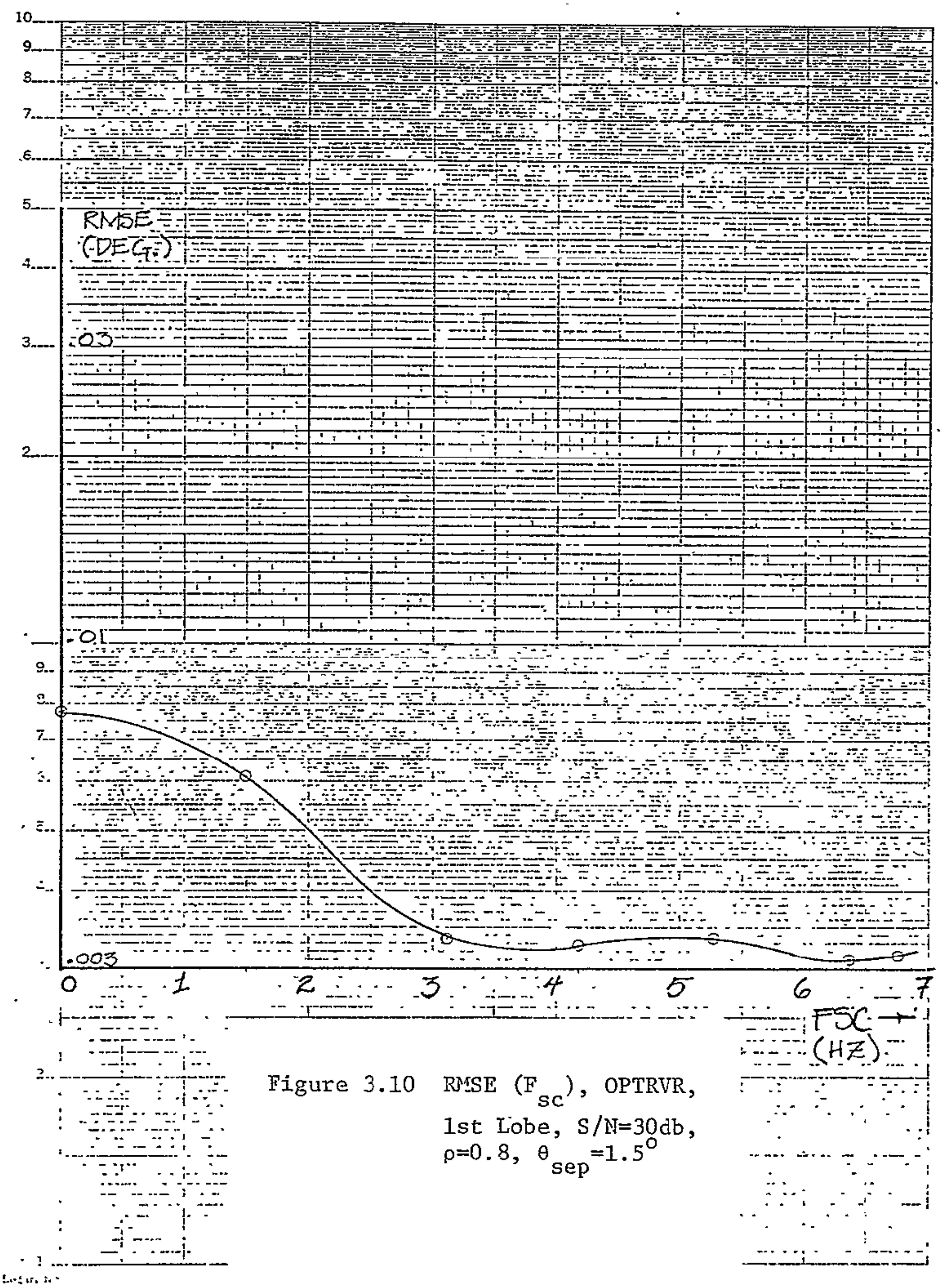


Figure 3.10 RMSE (F_{sc}), OPTRVR, 1st Lobe, S/N=30db, ρ=0.8, θ_{sep}=1.5^{o}

value of β, 0°, during the run. Table 3.17 shows all computed statistics of all error components for this simulation run.

Figure 3.11 and the associated Tables 3.18 thru 3.23 present the results of several statistical error studies versus F_{sc} with the latter ranging on the fifth lobe of the rep. rate sampled signal spectrum, i.e.

$$47.25 \leq F_{sc} \leq 60.75 \text{ Hz.} \qquad (3.23)$$

These studies indicate

1. The OPTRVR without constraints is definitely superior to the same receiver with constraints (Tables 3.18 and 3.19).
2. The SUBOPT performance is the same with and without constraints (Tables 3.20 and 3.21).
3. The OPTRVR with tethered estimates β, $\hat{\omega}_{sc}$ as shown performs the same with and without constraints (Tables 3.22 and 3.23).
4. Items (2) and (3) above suggest it is the constraint on $\hat{\omega}_{sc}$ only that degrades the OPTRVR performance referenced in item (1) above.

Relaxing the constraint on $\hat{\omega}_{sc}$ in OPTRVR may be beneficial here without harming the performance for low F_{sc} cases, but there is no certainty of that without more tests. The benefits of the constraints generally were established in Figure 3.8 but that involved the SUBOPT receiver where there was no ω_{sc} estimate. Clearly, relaxing the $\hat{\omega}_{sc}$ constraint in the OPTRVR will not in any sense enable tracking of ω_{sc} above half the rep. rate, since the 5D LOE does not exploit individually (for ω_{sc}-information) the TO and FRO scan pulses (as the 6D LOE would do). Further sutdy at other values of S/N, ρ and θ_{sep} are needed here.

NRUN = 5

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	0.	.296E-02	-.148E-03	.587E-03	-.272E-02	.123E-03	-.892E-03	-.153	-1.13
	1.49	.118E-01	-.549E-03	-.186E-03	.404E-03	.517E-03	-.133E-02	.133E-01	-.878E-02
	3.11	.132E-01	-.328E-03	-.816E-03	.104E-01	.105E-02	-.972E-03	.128E-01	-.286E-01
	4.19	.147E-01	-.470E-03	-.290E-03	.428E-02	.517E-03	-.364E-03	.378E-02	-.325E-01
	5.27	.932E-02	-.247E-03	-.624E-03	.737E-02	.581E-03	-.453E-03	.183E-01	-.224E-01
	6.35	.810E-02	-.240E-03	-.530E-03	.714E-02	.722E-03	-.588E-03	.865E-02	-.270E-01
	6.75	.749E-03	-.378E-04	.227E-03	.150E-02	.617E-03	-.110E-02	-.114E-01	.825
ERMS:	0.	.167	.377E-02	.210E-01	.179	.423E-02	.207E-01	.186	1.42
	1.49	.161	.361E-02	.203E-01	.174	.412E-02	.195E-01	.588E-01	.834
	3.11	.168	.334E-02	.194E-01	.177	.412E-02	.200E-01	.469E-01	.781
	4.19	.164	.327E-02	.193E-01	.177	.417E-02	.210E-01	.552E-01	.792
	5.27	.167	.337E-02	.197E-01	.181	.414E-02	.207E-01	.523E-01	.813
	6.35	.165	.310E-02	.192E-01	.171	.403E-02	.196E-01	.500E-01	.797
	6.75	.159	.315E-02	.195E-01	.180	.368E-02	.189E-01	.103	1.09
ESTD:	0.	.167	.377E-02	.210E-01	.179	.423E-02	.207E-01	.105	.873
	1.49	.161	.357E-02	.203E-01	.174	.409E-02	.195E-01	.572E-01	.834
	3.11	.168	.333E-02	.194E-01	.177	.399E-02	.200E-01	.451E-01	.781
	4.19	.164	.324E-02	.193E-01	.177	.414E-02	.210E-01	.550E-01	.791
	5.27	.166	.336E-02	.197E-01	.181	.410E-02	.207E-01	.490E-01	.813
	6.35	.165	.309E-02	.192E-01	.170	.397E-02	.196E-01	.492E-01	.796
	6.75	.159	.315E-02	.195E-01	.180	.363E-02	.189E-01	.103	.713

Table 3.17 Error Statistics vs F_{se}, OPTRVR, 1st Lobe S/N=30db, ρ=0.8, θ_{sep}=1.5^o

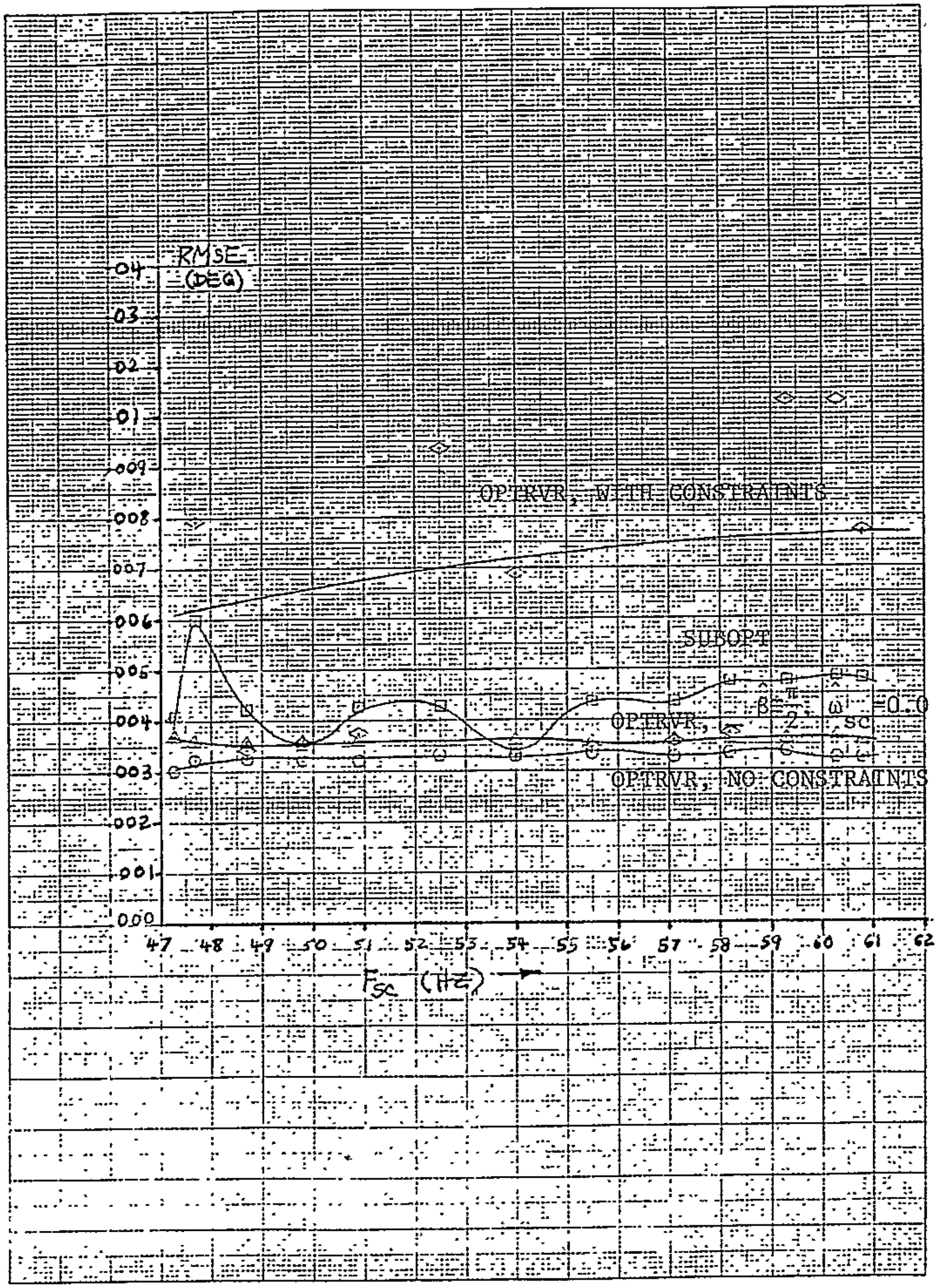


Figure 3.11 RMSE (F_{sc}), comparison of receivers and effects of constraints, 5th lobe, S/N=30db, ρ=0.8, θ_{sep}=1.5^{o}

NRUN = 6

CTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	47.3	.274E-02	.106E-03	.542E-04	.857E-02	.700E-03	-.268E-03	-.739E-02	.201E-01
	47.7	.706E-02	-.453E-04	.168E-03	.235E-03	-.681E-04	-.885E-03	-.245E-02	-.380E-02
	48.7	.334E-02	.118E-03	.203E-03	.561E-02	.414E-03	-.466E-03	-.103E-03	.377E-01
	49.8	.114E-01	-.310E-03	.112E-03	.450E-03	-.489E-04	-.107E-02	-.635E-02	.712E-03
	50.9	.943E-02	-.201E-03	.240E-03	.702E-02	.646E-03	-.103E-02	-.599E-02	-.341E-01
	52.5	.807E-02	-.285E-03	.281E-03	.112E-02	.208E-03	-.748E-03	-.209E-02	.163E-01
	54.0	.540E-02	-.190E-03	.287E-03	.263E-02	.462E-04	-.419E-03	-.315E-01	-.536E-02
	55.5	.733E-02	-.211E-03	.439E-03	.552E-02	.298E-03	-.386E-03	.275E-02	.258E-01
	57.1	.642E-02	-.956E-04	.294E-03	.454E-02	.347E-03	-.218E-03	.360E-03	.335E-02
	58.2	.632E-02	-.554E-04	.606E-03	-.258E-04	-.163E-04	-.143E-02	-.376E-02	.462E-01
	59.3	.532E-02	.140E-03	.195E-03	.637E-03	.233E-04	-.839E-03	-.114E-02	.602E-02
	60.3	.459E-02	.180E-04	.101E-03	.292E-02	.214E-03	.292E-04	.505E-02	-.572E-03
	60.8	.290E-02	.175E-03	.126E-03	.442E-02	.317E-03	-.454E-03	.929E-03	.674E-02
ERMS:	47.3	.165	.301E-02	.192E-01	.176	.368E-02	.194E-01	.647E-01	.935
	47.7	.158	.326E-02	.205E-01	.175	.393E-02	.204E-01	.474E-01	.698
	48.7	.162	.325E-02	.199E-01	.174	.369E-02	.202E-01	.591E-01	.972
	49.8	.163	.322E-02	.198E-01	.174	.379E-02	.205E-01	.528E-01	.885
	50.9	.160	.320E-02	.202E-01	.178	.367E-02	.188E-01	.671E-01	1.01
	52.5	.162	.330E-02	.204E-01	.176	.388E-02	.203E-01	.410E-01	.661
	54.0	.162	.328E-02	.204E-01	.174	.374E-02	.199E-01	.390E-01	.670
	55.5	.161	.333E-02	.203E-01	.179	.372E-02	.194E-01	.470E-01	.753
	57.1	.161	.326E-02	.202E-01	.176	.378E-02	.201E-01	.446E-01	.720
	58.2	.165	.331E-02	.203E-01	.173	.390E-02	.200E-01	.412E-01	.683
	59.3	.160	.337E-02	.205E-01	.176	.375E-02	.201E-01	.411E-01	.703
	60.3	.164	.324E-02	.200E-01	.178	.391E-02	.201E-01	.581E-01	.859
	60.8	.165	.324E-02	.198E-01	.176	.375E-02	.196E-01	.347E-01	.615
ESTD:	47.3	.165	.301E-02	.192E-01	.175	.361E-02	.194E-01	.643E-01	.934
	47.7	.158	.326E-02	.205E-01	.175	.393E-02	.204E-01	.473E-01	.698
	48.7	.162	.325E-02	.199E-01	.174	.367E-02	.202E-01	.591E-01	.971
	49.8	.163	.321E-02	.198E-01	.174	.379E-02	.205E-01	.525E-01	.885
	50.9	.159	.319E-02	.202E-01	.178	.361E-02	.187E-01	.669E-01	1.01
	52.5	.162	.328E-02	.204E-01	.176	.388E-02	.203E-01	.409E-01	.661
	54.0	.162	.327E-02	.204E-01	.174	.374E-02	.199E-01	.230E-01	.670
	55.5	.161	.332E-02	.203E-01	.178	.370E-02	.193E-01	.469E-01	.753
	57.1	.161	.326E-02	.202E-01	.176	.377E-02	.201E-01	.446E-01	.720
	58.2	.165	.331E-02	.203E-01	.173	.390E-02	.200E-01	.410E-01	.681
	59.3	.160	.336E-02	.205E-01	.176	.375E-02	.201E-01	.410E-01	.703
	60.3	.164	.324E-02	.200E-01	.178	.390E-02	.201E-01	.579E-01	.859
	60.8	.165	.324E-02	.198E-01	.178	.374E-02	.196E-01	.347E-01	.615

Table 3.18 Error Statistics vs F_{sc}, OPTRVR, 5th Lobe, S/N=30db, ρ=0.8, θ_{sep}=1.5°, NO CONSTRAINTS

NRUN = 6

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	47.3	.176	-.223E-02	.116E-02	.282	.477E-02	-.169E-02	-.179	295.
	47.7	.144	-.268E-02	.479E-02	.220	.478E-02	-.617E-02	-.118	293.
	48.7	.100	-.710E-03	.918E-03	.163	.199E-02	-.126E-02	-.113	304.
	49.8	.102	-.839E-03	.847E-03	.170	.229E-02	-.819E-03	-.106	311.
	50.9	.110	-.126E-02	.199E-02	.179	.295E-02	-.253E-02	-.104	317.
	52.5	.152	-.490E-02	-.473E-04	.224	.827E-02	-.915E-03	-.560E-02	290.
	54.0	.100	.142E-03	.616E-02	.152	.567E-03	-.777E-02	-1.59	299.
	55.5	.922E-01	-.103E-02	.428E-04	.147	.238E-02	-.318E-03	-.940E-01	347.
	57.1	.115	-.132E-02	.122E-02	.182	.288E-02	-.239E-02	-.109	356.
	58.2	.105	-.792E-03	.758E-03	.174	.219E-02	-.969E-03	-.111	363.
	59.3	.192	-.885E-02	.910E-03	.249	.132E-01	-.201E-02	-.601E-02	333.
	60.3	.202	-.753E-02	.607E-02	.296	.123E-01	-.761E-02	-.111	355.
	60.8	.139	-.531E-02	.644E-02	.192	.789E-02	-.623E-02	-.115	343.
ERMS:	47.3	.242	.404E-02	.196E-01	.334	.642E-02	.208E-01	1.66	295.
	47.7	.239	.795E-02	.424E-01	.304	.108E-01	.464E-01	.973	293.
	48.7	.205	.341E-02	.196E-01	.249	.469E-02	.206E-01	.947	304.
	49.8	.205	.355E-02	.200E-01	.278	.491E-02	.212E-01	.935	311.
	50.9	.210	.370E-02	.210E-01	.279	.513E-02	.205E-01	.927	317.
	52.5	.277	.934E-02	.477E-01	.341	.128E-01	.554E-01	.977	290.
	54.0	.236	.687E-02	.492E-01	.269	.935E-02	.567E-01	1.59	299.
	55.5	.201	.345E-02	.192E-01	.238	.483E-02	.206E-01	.877	347.
	57.1	.206	.359E-02	.196E-01	.267	.500E-02	.204E-01	.851	356.
	58.2	.207	.372E-02	.206E-01	.256	.462E-02	.197E-01	.807	363.
	59.3	.286	.128E-01	.518E-01	.369	.182E-01	.579E-01	.802	333.
	60.3	.329	.126E-01	.613E-01	.387	.170E-01	.647E-01	.938	355.
	60.8	.215	.771E-02	.388E-01	.286	.113E-01	.471E-01	1.45	343.
ESTD:	47.3	.166	.333E-02	.196E-01	.179	.429E-02	.207E-01	1.65	.532
	47.7	.191	.748E-02	.421E-01	.209	.973E-02	.460E-01	.966	9.54
	48.7	.179	.334E-02	.196E-01	.166	.424E-02	.206E-01	.940	1.48
	49.8	.178	.345E-02	.200E-01	.219	.434E-02	.212E-01	.929	1.49
	50.9	.179	.355E-02	.209E-01	.214	.420E-02	.203E-01	.921	1.23
	52.5	.231	.795E-02	.477E-01	.256	.960E-02	.554E-01	.977	2.33
	54.0	.214	.687E-02	.489E-01	.222	.934E-02	.562E-01	.167	1.34
	55.5	.179	.329E-02	.192E-01	.187	.420E-02	.206E-01	.872	.768
	57.1	.174	.333E-02	.195E-01	.195	.409E-02	.202E-01	.844	1.73
	58.2	.178	.364E-02	.206E-01	.188	.407E-02	.197E-01	.799	2.10
	59.3	.215	.918E-02	.517E-01	.272	.126E-01	.578E-01	.802	2.45
	60.3	.260	.101E-01	.610E-01	.249	.118E-01	.643E-01	.932	15.7
	60.8	.164	.560E-02	.363E-01	.212	.811E-02	.467E-01	1.45	3.28

Table 3.19 Error Stastistics vs F_{sc}, OPTRVR, 5th Lobe, S/N=30db, ρ=0.8, θ_{sep}=1.5^{o}, WITH CONSTRAINTS

NRUN = 7

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT
EMEAN:	47.3	-.554E-01	.222E-02	.152E-03	-.437E-01	-.195E-02	-.464E-04
	47.7	.494E-01	-.171E-02	.628E-03	.520E-01	.253E-02	-.129E-02
	48.7	-.217E-01	.547E-03	.167E-03	-.291E-01	.232E-03	-.249E-03
	49.8	-.173E-01	-.144E-03	.529E-03	-.173E-01	.585E-03	-.613E-03
	50.9	-.195E-01	-.126E-03	-.112E-03	-.102E-01	.120E-02	-.378E-03
	52.5	-.249E-02	-.591E-03	.409E-03	.196E-01	.954E-03	-.315E-03
	54.0	-.102E-01	.403E-04	.272E-03	-.117E-01	.460E-03	-.688E-03
	55.5	-.606E-02	-.148E-03	.876E-03	-.132E-01	.789E-03	-.303E-03
	57.1	-.145E-01	-.500E-03	.577E-03	-.346E-01	.139E-03	-.165E-03
	58.2	-.101E-01	-.393E-03	.232E-03	-.454E-01	-.152E-03	-.985E-03
	59.3	-.265E-01	.217E-03	.700E-03	-.469E-02	.115E-03	-.383E-03
	60.3	.234E-01	-.126E-02	.172E-03	.149E-01	.161E-02	-.323E-03
	60.8	.672E-01	-.314E-02	.335E-03	.550E-01	.343E-02	-.711E-03
ERMS:	47.3	.220	.410E-02	.193E-01	.231	.449E-02	.197E-01
	47.7	.224	.588E-02	.253E-01	.215	.665E-02	.253E-01
	48.7	.210	.421E-02	.231E-01	.227	.510E-02	.219E-01
	49.8	.196	.353E-02	.191E-01	.214	.441E-02	.206E-01
	50.9	.195	.429E-02	.228E-01	.225	.543E-02	.243E-01
	52.5	.198	.429E-02	.219E-01	.226	.457E-02	.216E-01
	54.0	.193	.340E-02	.180E-01	.211	.419E-02	.188E-01
	55.5	.220	.437E-02	.239E-01	.221	.503E-02	.225E-01
	57.1	.200	.437E-02	.227E-01	.207	.503E-02	.221E-01
	58.2	.219	.474E-02	.230E-01	.259	.501E-02	.220E-01
	59.3	.205	.476E-02	.230E-01	.267	.552E-02	.260E-01
	60.3	.200	.482E-02	.224E-01	.239	.712E-02	.286E-01
	60.8	.216	.488E-02	.177E-01	.207	.547E-02	.170E-01
ESTD:	47.3	.213	.344E-02	.193E-01	.227	.404E-02	.197E-01
	47.7	.218	.562E-02	.253E-01	.208	.613E-02	.253E-01
	48.7	.209	.418E-02	.231E-01	.225	.509E-02	.219E-01
	49.8	.197	.353E-02	.191E-01	.214	.437E-02	.206E-01
	50.9	.195	.425E-02	.228E-01	.224	.530E-02	.243E-01
	52.5	.198	.425E-02	.219E-01	.225	.447E-02	.216E-01
	54.0	.193	.340E-02	.180E-01	.211	.416E-02	.188E-01
	55.5	.220	.437E-02	.239E-01	.220	.502E-02	.225E-01
	57.1	.199	.434E-02	.227E-01	.204	.503E-02	.221E-01
	58.2	.219	.472E-02	.230E-01	.255	.501E-02	.220E-01
	59.3	.203	.476E-02	.230E-01	.267	.552E-02	.260E-01
	60.3	.207	.455E-02	.224E-01	.238	.693E-02	.286E-01
	60.8	.205	.364E-02	.177E-01	.200	.426E-02	.169E-01

Table 3.20 Error Statistics vs F_{sc}, SUBOPT, 5th Lobe, S/N=30db, $\rho=0.8$, $\theta_{sep}=1.5^{o}$ NO CONSTRAINTS

NRUN = 7

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT
EMEAN:	47.3	-.554E-01	.222E-02	.152E-03	-.437E-01	-.195E-02	-.464E-04
	47.7	.494E-01	-.171E-02	.628E-03	.520E-01	.253E-02	-.129E-02
	48.7	-.217E-01	.547E-03	.167E-03	-.291E-01	.232E-03	-.249E-03
	49.8	-.173E-01	-.144E-03	.529E-03	-.173E-01	.585E-03	-.613E-03
	50.9	-.195E-01	-.126E-03	-.112E-03	-.102E-01	.120E-02	-.378E-03
	52.5	-.249E-02	-.591E-03	.409E-03	.196E-01	.954E-03	-.315E-03
	54.0	-.102E-01	.403E-04	.272E-03	-.117E-01	.460E-03	-.688E-03
	55.5	-.606E-02	-.148E-03	.876E-03	-.132E-01	.789E-03	-.303E-03
	57.1	-.145E-01	-.500E-03	.577E-03	-.346E-01	.139E-03	-.165E-03
	58.2	-.101E-01	-.398E-03	.232E-03	-.454E-01	-.152E-03	-.985E-03
	59.3	-.285E-01	.217E-03	.700E-03	-.469E-02	.115E-03	-.383E-03
	60.3	.234E-01	-.126E-02	.172E-03	.149E-01	.161E-02	-.323E-03
	60.8	.672E-01	-.314E-02	.335E-03	.530E-01	.343E-02	-.711E-03
ERMS:	47.3	.220	.410E-02	.193E-01	.231	.449E-02	.197E-01
	47.7	.224	.588E-02	.253E-01	.215	.663E-02	.253E-01
	48.7	.210	.421E-02	.231E-01	.227	.510E-02	.219E-01
	49.8	.198	.353E-02	.191E-01	.214	.441E-02	.206E-01
	50.9	.196	.425E-02	.228E-01	.225	.543E-02	.243E-01
	52.5	.198	.429E-02	.219E-01	.226	.457E-02	.216E-01
	54.0	.193	.340E-02	.180E-01	.211	.419E-02	.188E-01
	55.5	.220	.437E-02	.239E-01	.221	.508E-02	.225E-01
	57.1	.200	.437E-02	.227E-01	.207	.503E-02	.221E-01
	58.2	.219	.474E-02	.230E-01	.259	.501E-02	.220E-01
	59.3	.205	.476E-02	.230E-01	.267	.552E-02	.260E-01
	60.3	.208	.482E-02	.224E-01	.239	.712E-02	.286E-01
	60.8	.216	.480E-02	.177E-01	.207	.547E-02	.170E-01
ESTD:	47.3	.213	.344E-02	.193E-01	.227	.404E-02	.197E-01
	47.7	.218	.562E-02	.253E-01	.208	.613E-02	.253E-01
	48.7	.209	.418E-02	.231E-01	.225	.509E-02	.219E-01
	49.8	.197	.353E-02	.191E-01	.214	.437E-02	.206E-01
	50.9	.195	.425E-02	.228E-01	.224	.530E-02	.243E-01
	52.5	.198	.425E-02	.219E-01	.225	.447E-02	.216E-01
	54.0	.193	.340E-02	.180E-01	.211	.416E-02	.188E-01
	55.5	.220	.437E-02	.239E-01	.220	.502E-02	.225E-01
	57.1	.199	.434E-02	.227E-01	.204	.503E-02	.221E-01
	58.2	.219	.472E-02	.230E-01	.255	.501E-02	.220E-01
	59.3	.203	.476E-02	.230E-01	.267	.552E-02	.260E-01
	60.3	.207	.465E-02	.224E-01	.238	.693E-02	.286E-01
	60.8	.205	.364E-02	.177E-01	.200	.426E-02	.169E-01

Table 3.21 Error Stastistics vs F_{sc}, SUBOPT, 5th Lobe, S/N=30db, ρ=0.8, θ_{sep}=1.5^{o}, WITH CONSTRAINTS

NRUN = 8

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	47.3	.162	-.142E-02	-.761E-03	.259	.331E-02	.191E-03	-.817E-13	297.
	47.7	.952E-01	-.104E-02	-.300E-03	.146	.213E-02	-.186E-03	.439E-12	299.
	48.7	.941E-01	-.101E-02	-.267E-03	.146	.224E-02	-.226E-03	.108E-13	306.
	49.8	.939E-01	-.100E-02	-.292E-03	.146	.222E-02	-.223E-03	.881E-14	313.
	50.9	.936E-01	-.973E-03	-.278E-03	.146	.216E-02	-.260E-03	.384E-12	320.
	52.5	.936E-01	-.100E-02	-.281E-03	.146	.219E-02	-.217E-03	-.204E-14	330.
	54.0	.118	-.113E-02	-.243E-03	.173	.238E-02	-.272E-03	-1.57	339.
	55.5	.939E-01	-.994E-03	-.265E-03	.146	.218E-02	-.241E-03	.319E-12	349.
	57.1	.949E-01	-.104E-02	-.268E-03	.146	.215E-02	-.203E-03	-.427E-12	359.
	58.2	.945E-01	-.102E-02	-.272E-03	.145	.215E-02	-.251E-03	.182E-12	366.
	59.3	.937E-01	-.992E-03	-.295E-03	.146	.219E-02	-.191E-03	.421E-12	372.
	60.3	.941E-01	-.101E-02	-.280E-03	.146	.218E-02	-.215E-03	-.335E-12	379.
	60.8	.749E-01	-.922E-03	-.124E-03	.113	.184E-02	-.373E-03	-.284E-12	382.
ERMS:	47.3	.232	.369E-02	.199E-01	.314	.542E-02	.210E-01	1.57	297.
	47.7	.213	.359E-02	.201E-01	.257	.498E-02	.217E-01	.907	299.
	48.7	.205	.356E-02	.199E-01	.239	.479E-02	.208E-01	.907	306.
	49.8	.201	.356E-02	.200E-01	.260	.485E-02	.213E-01	.907	313.
	50.9	.200	.356E-02	.200E-01	.255	.475E-02	.209E-01	.907	320.
	52.5	.200	.357E-02	.200E-01	.261	.479E-02	.211E-01	.907	330.
	54.0	.203	.361E-02	.200E-01	.246	.488E-02	.209E-01	1.57	339.
	55.5	.205	.354E-02	.200E-01	.238	.474E-02	.207E-01	.907	349.
	57.1	.199	.357E-02	.199E-01	.239	.474E-02	.208E-01	.907	359.
	58.2	.198	.361E-02	.203E-01	.237	.469E-02	.205E-01	.907	366.
	59.3	.195	.363E-02	.202E-01	.261	.482E-02	.213E-01	.907	372.
	60.3	.193	.364E-02	.203E-01	.249	.473E-02	.210E-01	.907	379.
	60.8	.182	.353E-02	.200E-01	.208	.464E-02	.208E-01	1.57	382.
ESTD:	47.3	.167	.340E-02	.199E-01	.179	.429E-02	.210E-01	1.57	.748E-04
	47.7	.191	.343E-02	.201E-01	.211	.450E-02	.217E-01	.907	.607E-04
	48.7	.182	.341E-02	.199E-01	.189	.423E-02	.208E-01	.907	I
	49.8	.178	.341E-02	.200E-01	.216	.431E-02	.213E-01	.907	.716E-04
	50.9	.177	.343E-02	.200E-01	.210	.423E-02	.209E-01	.907	I
	52.5	.177	.343E-02	.200E-01	.216	.426E-02	.211E-01	.907	.143E-03
	54.0	.166	.342E-02	.200E-01	.175	.426E-02	.209E-01	.559E-06	.131E-03
	55.5	.183	.340E-02	.200E-01	.188	.421E-02	.207E-01	.907	.131E-03
	57.1	.175	.341E-02	.199E-01	.189	.422E-02	.208E-01	.907	I
	58.2	.174	.346E-02	.202E-01	.186	.417E-02	.205E-01	.907	.226E-03
	59.3	.171	.346E-02	.202E-01	.216	.430E-02	.213E-01	.907	.200E-03
	60.3	.168	.350E-02	.203E-01	.202	.419E-02	.210E-01	.907	.143E-03
	60.8	.165	.341E-02	.200E-01	.175	.426E-02	.208E-01	1.57	.205E-03

Table 3.22 Error Stastistics vs F_{sc}, OPTRVR, 5th Lobe, $(\hat{\beta},\omega_{sc})$ tethered to $(\pi/2,0)$
S/N=30db, ρ=0.8, θ_{sep}=1.5^{o}, NO CONSTRAINTS

NRUN = 8

QTY	FSC	ALFA	THETA	THEDOT	ALFAR	THETAR	THRDOT	B	WSC
EMEAN:	47.3	.162	-.142E-02	-.761E-03	.259	.331E-02	.191E-03	-.817E-13	297.
	47.7	.952E-01	-.104E-02	-.300E-03	.146	.213E-02	-.186E-03	.439E-12	299.
	48.7	.941E-01	-.101E-02	-.267E-03	.146	.224E-02	-.226E-03	.106E-13	306.
	49.8	.939E-01	-.100E-02	-.292E-03	.146	.222E-02	-.223E-03	.631E-14	313.
	50.9	.938E-01	-.973E-03	-.278E-03	.146	.216E-02	-.268E-03	.384E-12	320.
	52.5	.938E-01	-.100E-02	-.281E-03	.146	.219E-02	-.217E-03	-.284E-14	330.
	54.0	.118	-.113E-02	-.243E-03	.173	.238E-02	-.272E-03	-1.57	339.
	55.5	.939E-01	-.994E-03	-.260E-03	.146	.218E-02	-.241E-03	.319E-12	349.
	57.1	.949E-01	-.104E-02	-.268E-03	.146	.215E-02	-.203E-03	-.427E-12	359.
	58.2	.945E-01	-.102E-02	-.272E-03	.145	.215E-02	-.251E-03	.182E-12	366.
	59.3	.937E-01	-.992E-03	-.295E-03	.146	.219E-02	-.191E-03	.421E-12	372.
	60.3	.941E-01	-.101E-02	-.280E-03	.146	.218E-02	-.215E-03	-.335E-12	379.
	60.8	.749E-01	-.922E-03	-.124E-03	.113	.184E-02	-.373E-03	-.184E-12	382.
ERMS:	47.3	.232	.369E-02	.199E-01	.314	.542E-02	.210E-01	1.57	297.
	47.7	.213	.359E-02	.201E-01	.257	.498E-02	.217E-01	.907	299.
	48.7	.205	.356E-02	.199E-01	.239	.479E-02	.205E-01	.907	306.
	49.8	.201	.356E-02	.200E-01	.260	.485E-02	.213E-01	.907	313.
	50.9	.200	.356E-02	.200E-01	.255	.475E-02	.209E-01	.907	320.
	52.5	.200	.357E-02	.200E-01	.261	.479E-02	.211E-01	.907	330.
	54.0	.203	.361E-02	.200E-01	.246	.488E-02	.209E-01	1.57	339.
	55.5	.205	.354E-02	.200E-01	.238	.474E-02	.207E-01	.907	349.
	57.1	.199	.357E-02	.199E-01	.239	.474E-02	.208E-01	.907	359.
	58.2	.198	.361E-02	.203E-01	.237	.469E-02	.205E-01	.907	366.
	59.3	.195	.360E-02	.202E-01	.261	.482E-02	.213E-01	.907	372.
	60.3	.193	.364E-02	.203E-01	.249	.473E-02	.210E-01	.907	379.
	60.8	.182	.333E-02	.200E-01	.208	.464E-02	.208E-01	1.57	382.
ESTD:	47.3	.167	.340E-02	.199E-01	.179	.429E-02	.210E-01	1.57	.748E-04
	47.7	.191	.343E-02	.201E-01	.211	.450E-02	.217E-01	.907	.807E-04
	48.7	.182	.341E-02	.199E-01	.209	.423E-02	.205E-01	.907	I
	49.8	.179	.341E-02	.200E-01	.216	.431E-02	.213E-01	.907	.716E-04
	50.9	.177	.343E-02	.200E-01	.210	.423E-02	.209E-01	.907	I
	52.5	.177	.343E-02	.200E-01	.216	.426E-02	.211E-01	.907	.143E-03
	54.0	.166	.342E-02	.200E-01	.175	.426E-02	.209E-01	.559E-06	.131E-03
	55.5	.183	.340E-02	.200E-01	.188	.421E-02	.207E-01	.907	.131E-03
	57.1	.175	.341E-02	.199E-01	.189	.422E-02	.208E-01	.907	I
	58.2	.174	.346E-02	.202E-01	.188	.417E-02	.205E-01	.907	.226E-03
	59.3	.171	.346E-02	.202E-01	.216	.430E-02	.213E-01	.907	.200E-03
	60.3	.168	.350E-02	.203E-01	.202	.419E-02	.210E-01	.907	.143E-03
	60.8	.165	.341E-02	.200E-01	.175	.426E-02	.208E-01	1.57	.205E-03

Table 3.23 Error Stastistics vs F_{sc}, OPTRVR, 5th Lobe, $(\beta, \hat{\omega}_{sc})$ tethered to $(\pi/2, 0)$
S/N=30db, ρ=0.8, $\theta_{\ }$=1.5°, WITH CONSTRAINTS

Figures 3.12 thru 3.23 present a study of the interference acquisition capabilities of the interference trackers in the OPTRVR and SUBOPT designs. The 4 top traces on each figure are error time histories repectively (from the top) of the α, θ, α_R and θ_R estimate components; the bottom tract shows when the step of interference (ρ=0.8) occurs and simultaneously when the interference tracker estimate-vector elements are released from the idler values (equations 3.15 thru 3.19 above). The direct path pulse is being tracked from time zero. All plots have the same time-axis scaling.

Figures 3.11 thru 3.20 for the OPTRVR design (with constraints) show the following:

1. Successful pull-in for small scalloping rates and separation angles as small as the following:

   a. θ_{sep} = 0.25° @ S/N = 40 db (Figure 3.12)

   b. θ_{sep} = 0.5° @ S/N = 20 db (Figure 3.16)

   c. θ_{sep} = 1° @ S/N = 14 db (Figure 3.19)

2. Successful pull-in for small scalloping rates and separation angles probably extending nearly to the window edge (4°).

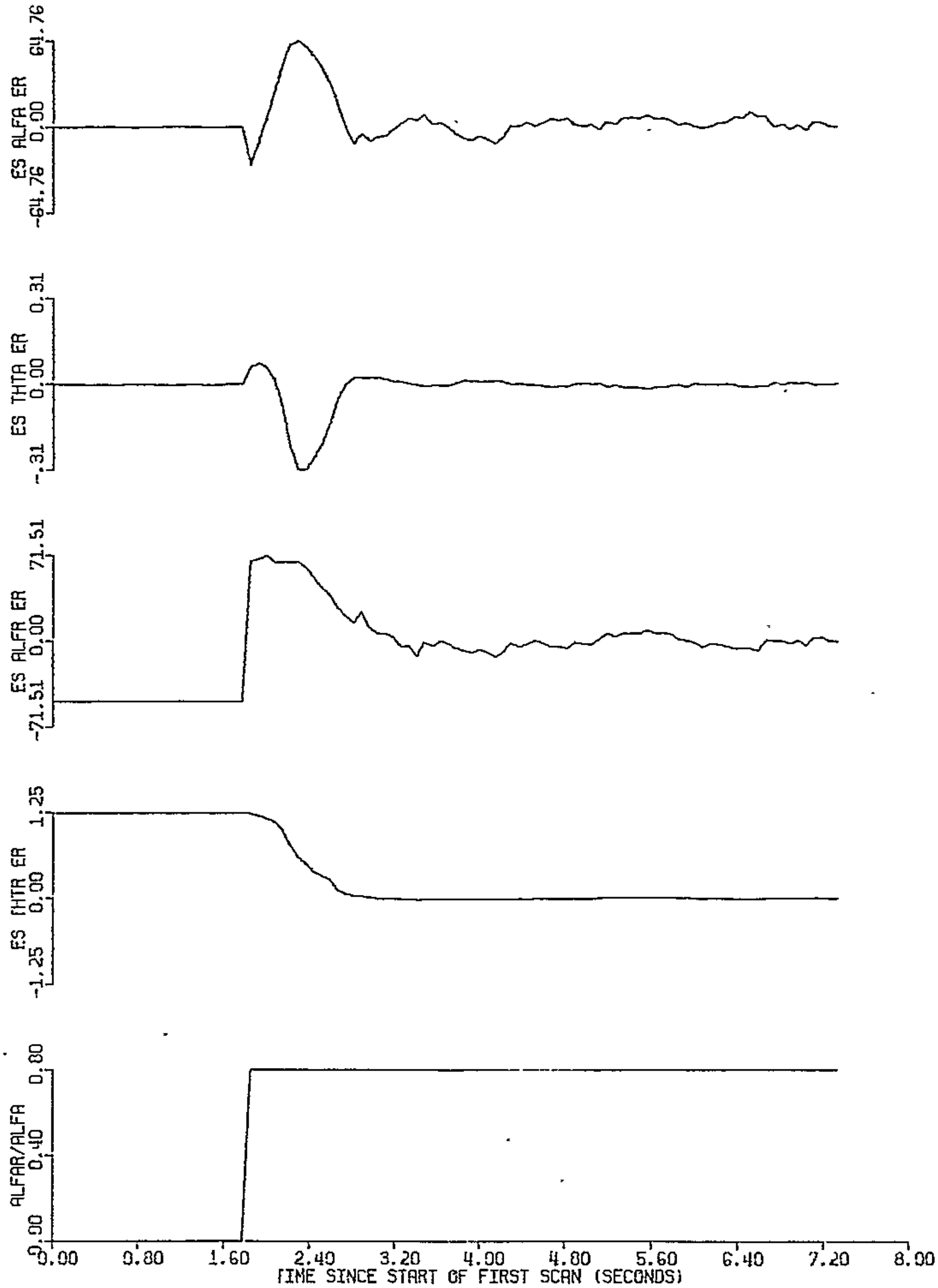
3. Successful pull-in @ S/N 40 db, θ_{sep} = 1° and F_{sc} in the 5th lobe (Figure 3.20). The exact F_{sc} (51.3 Hz) and initial β (-168°) were selected to produce, in the middle of the run, a maximum enhancement of the TO pulse and a maximum cancellation of the FRO pulse.

Figures 3.21 thru 3.23 for the SUBOPT design show the following:

1. Results for θ_{sep} = 0.5° and F_{sc} = 0.6 Hz (low):

SIM. JOB: HUGHOLO FILE NO: 512211U301 DATE: 04/03/79

PLOT JOB: HUGHOS1 PROGRAM: ACQMP1 DATE: 04/04/79

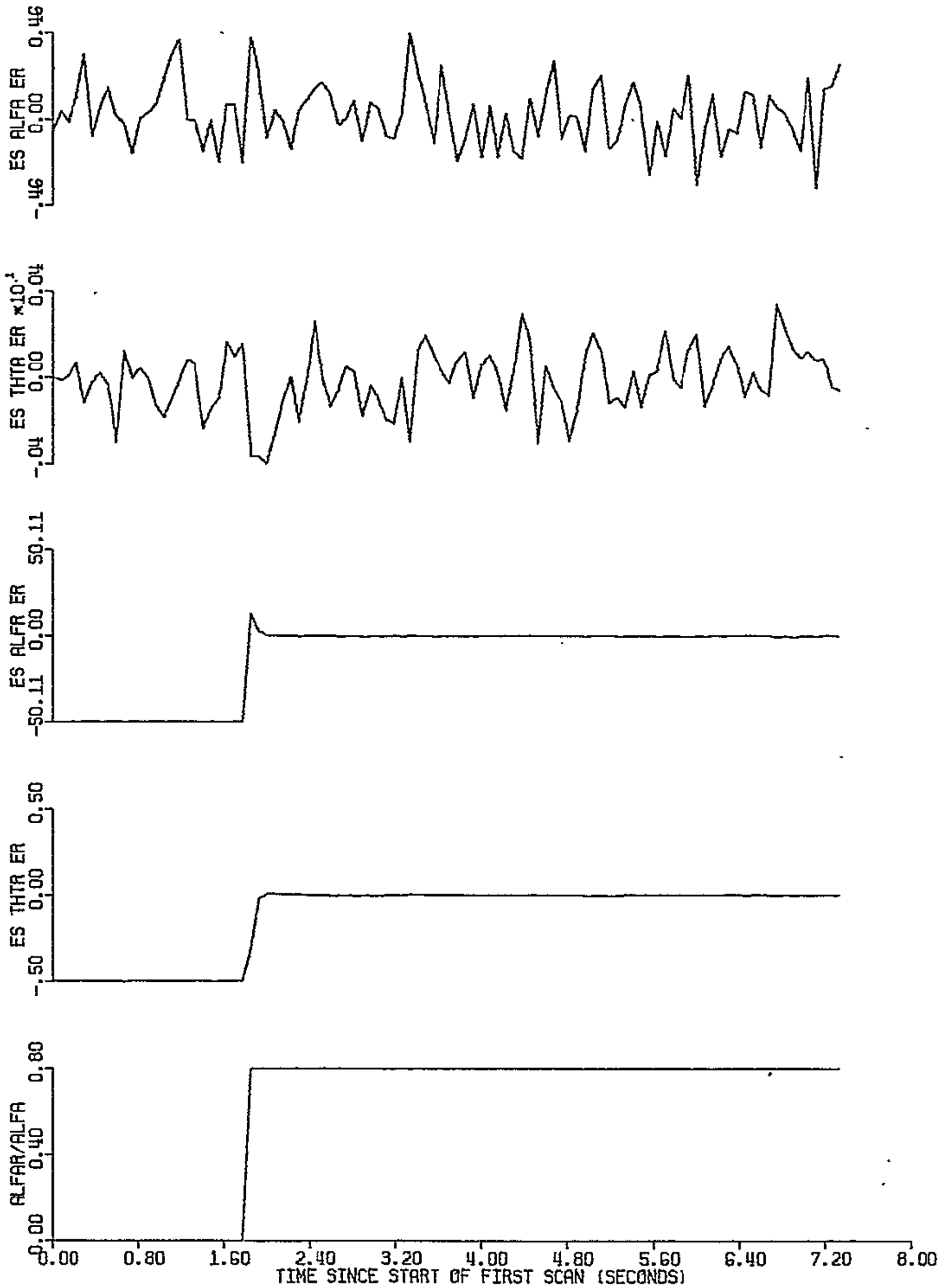


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= .250 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= -.390 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= .241E-01HZ

Figure 3.12 Interference Acquisition, OPTRVR, S/N=30db, ρ=0.8, $\theta_{sep}=0.25^o$, $\beta=45^o$, F_{sc}=0.6Hz

SIM. JOB: HUGHOS1 FILE NO: 512211U302 DATE: 04/03/79

PLOT JOB: HUGHO34 PROGRAM: ACOMP1 DATE: 04/05/79

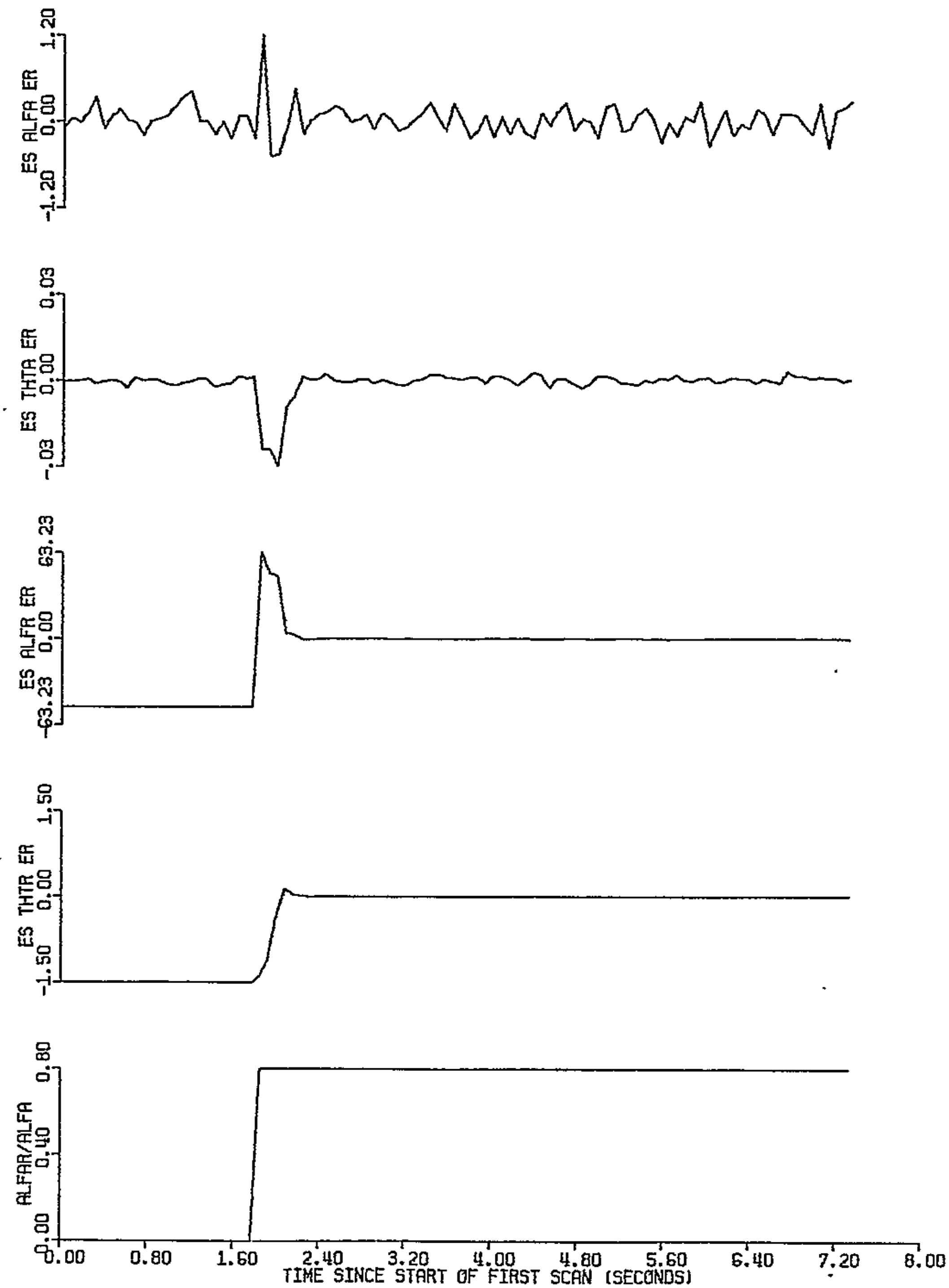


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= 2.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= 5.61 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= -.118 HZ

Figure 3.13 Interference Acquisition, OPTRVR, S/N=40db, ρ=0.8 $\theta_{sep}=2^{o}$, $\beta=45^{o}$, F_{sc}=0.6Hz

SIM. JOB: HUGHOTE FILE NO: 5122110303 DATE: 04/03/79

PLOT JOB: HUGH034 PROGRAM: ACQMP1 DATE: 04/05/79

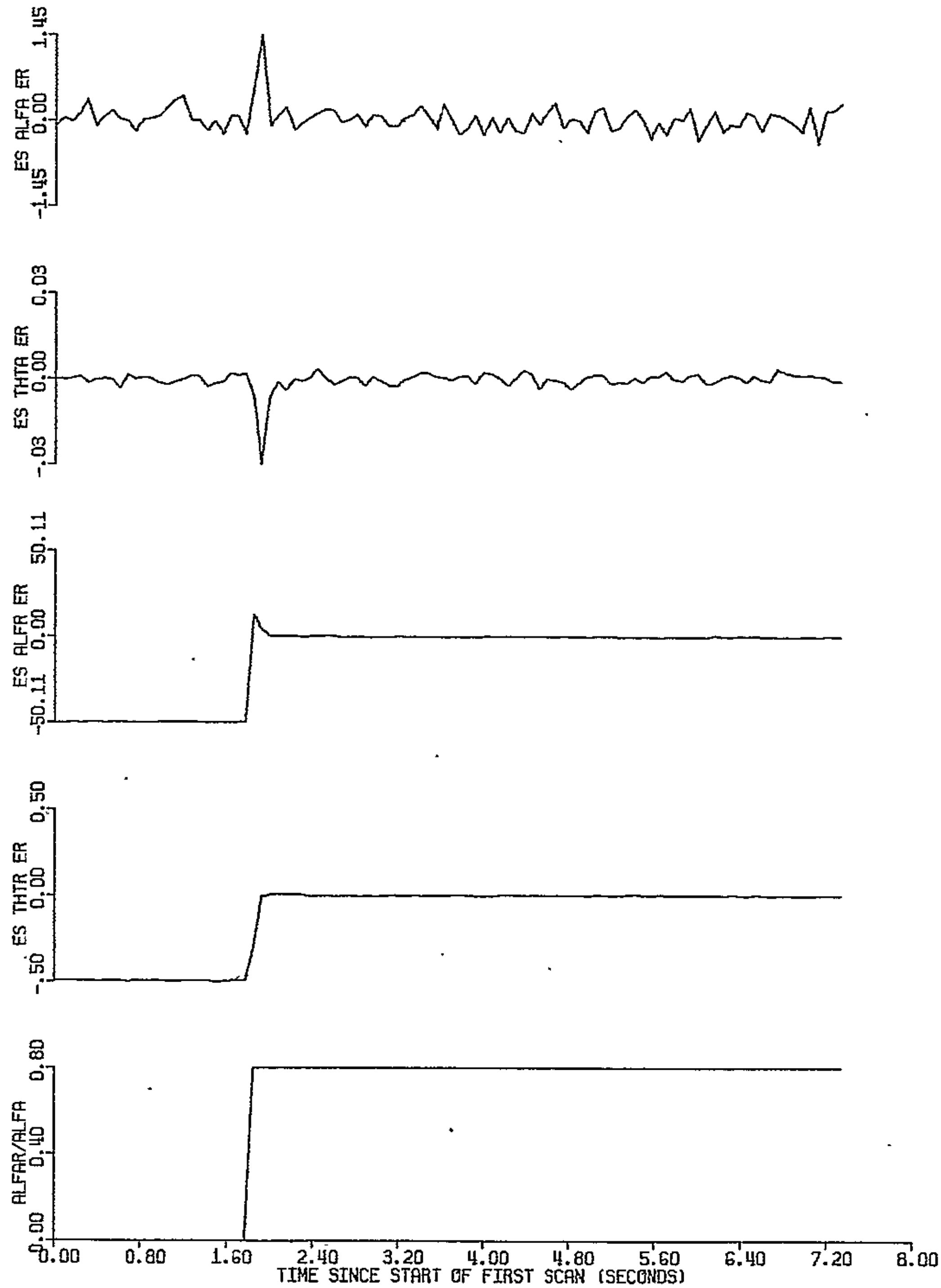


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= 3.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= -2.47 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= .946E-01HZ

Figure 3.14 Interference Acquisition, OPTRVR, S/N=40db, ρ=0.8
$\theta_{sep}=3^o$, $\beta=45^o$, F_{sc}=0.6Hz

SIM. JOB: HUGHOU1 FILE NO: 5122110307 DATE: 04/03/79

PLOT JOB: HUGHOM1 PROGRAM: ACOMP1 DATE: 04/05/79

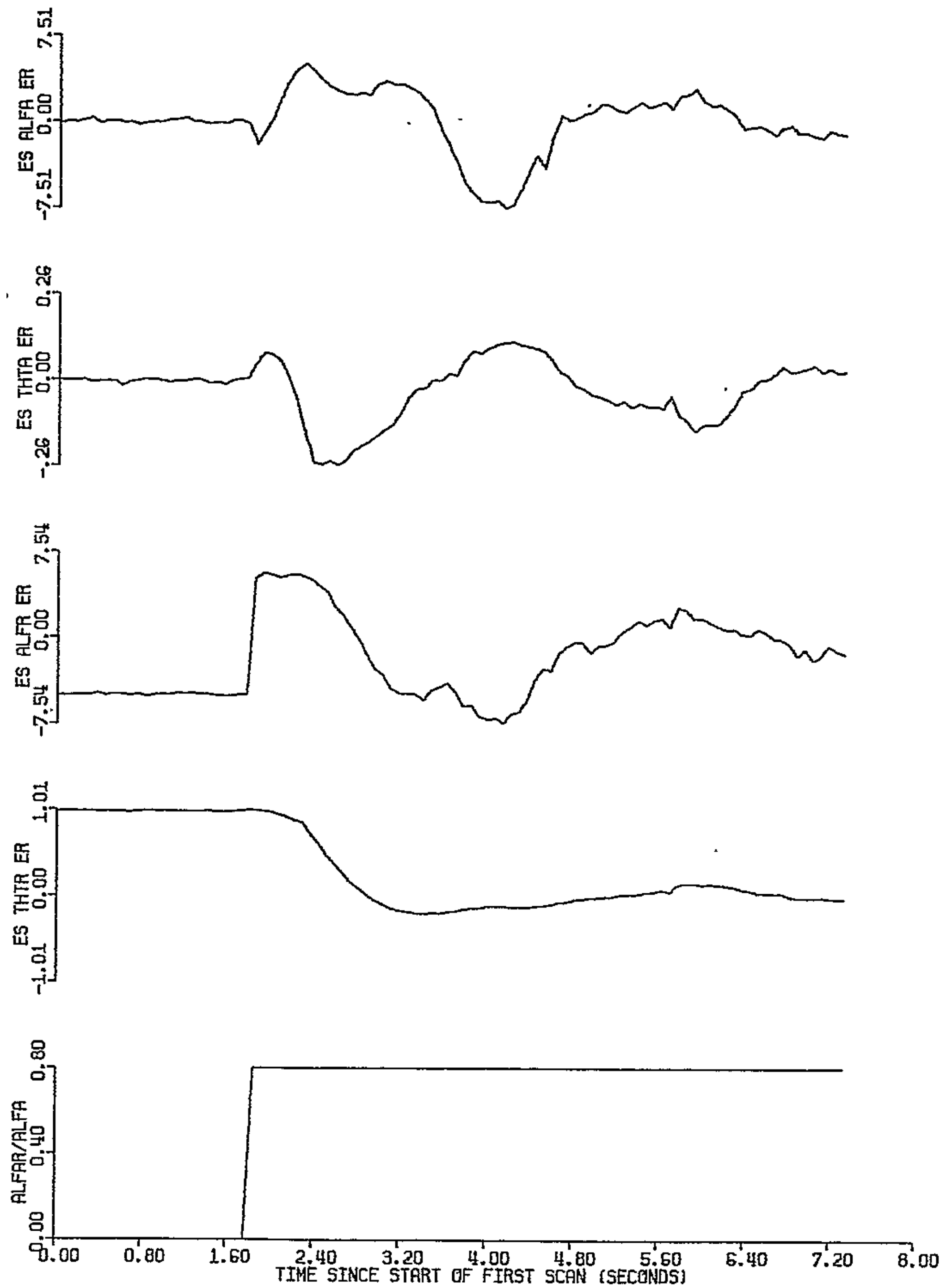


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, 2.500 HZ, THESEP= 2.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, PCPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= 4.97 DEG
FSC ESTIMATE ERROR: INITIAL= 2.50 ,FINAL= .203 HZ

Figure 3.15 Interference Acquisition, OPTRVR, S/N=40db, ρ=0.8, $\theta_{sep}=2^{o}$, $\beta=45^{o}$, $F_{sc}=2.5$Hz

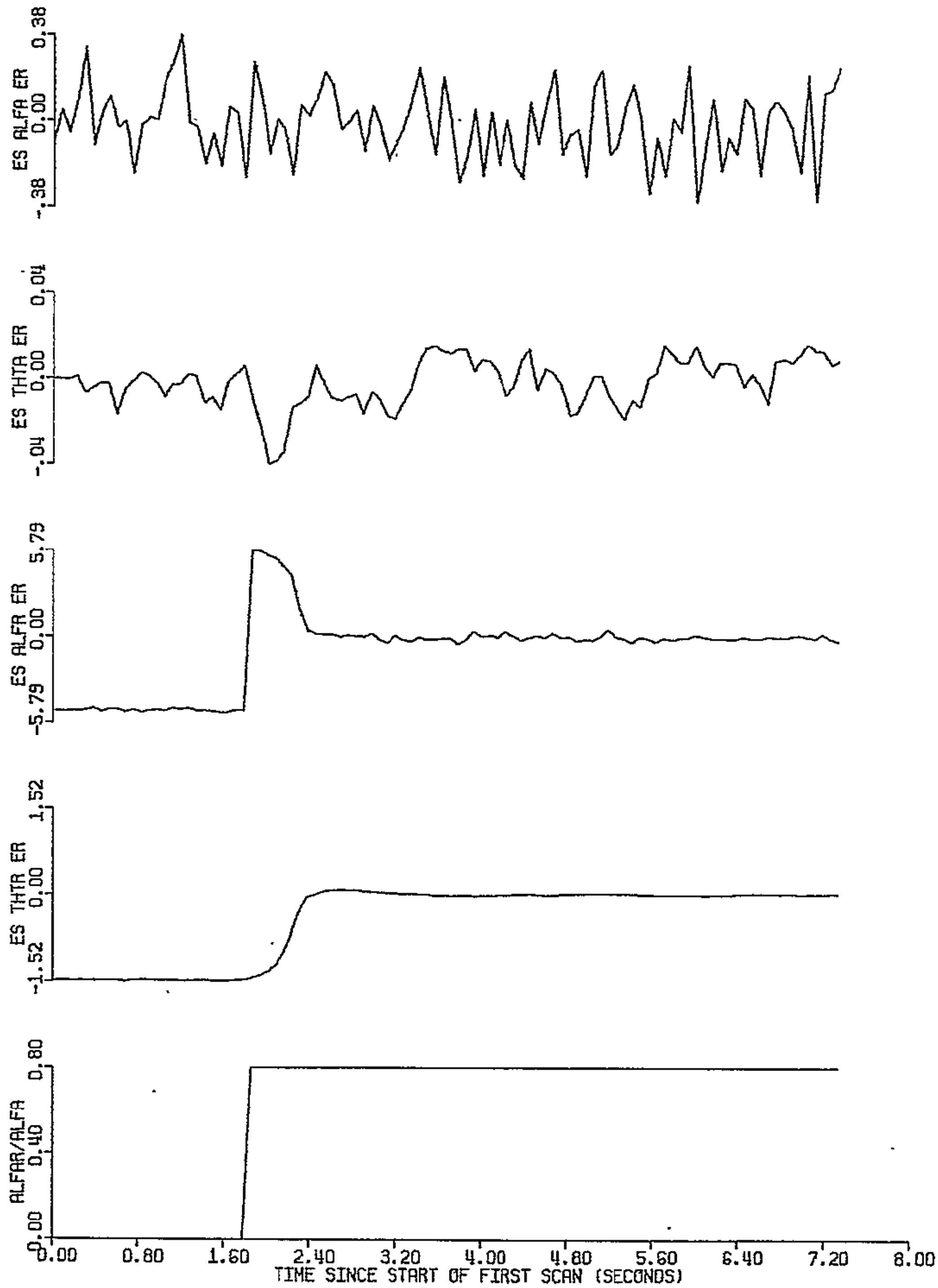
SIM. JOB: HUGHOTR FILE NO: 5122110304 DATE: 04/03/79

PLOT JOB: HUGH034 PROGRAM: ACQMP1 DATE: 04/05/79



SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 20.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= .500 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= -2.98 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= .124 HZ

Figure 3.16 Interference Acquisition, OPTRVR, S/N=20db, ρ=0.8, θ_{sep}=0.5°, β=45°, F_{sc}=0.6Hz

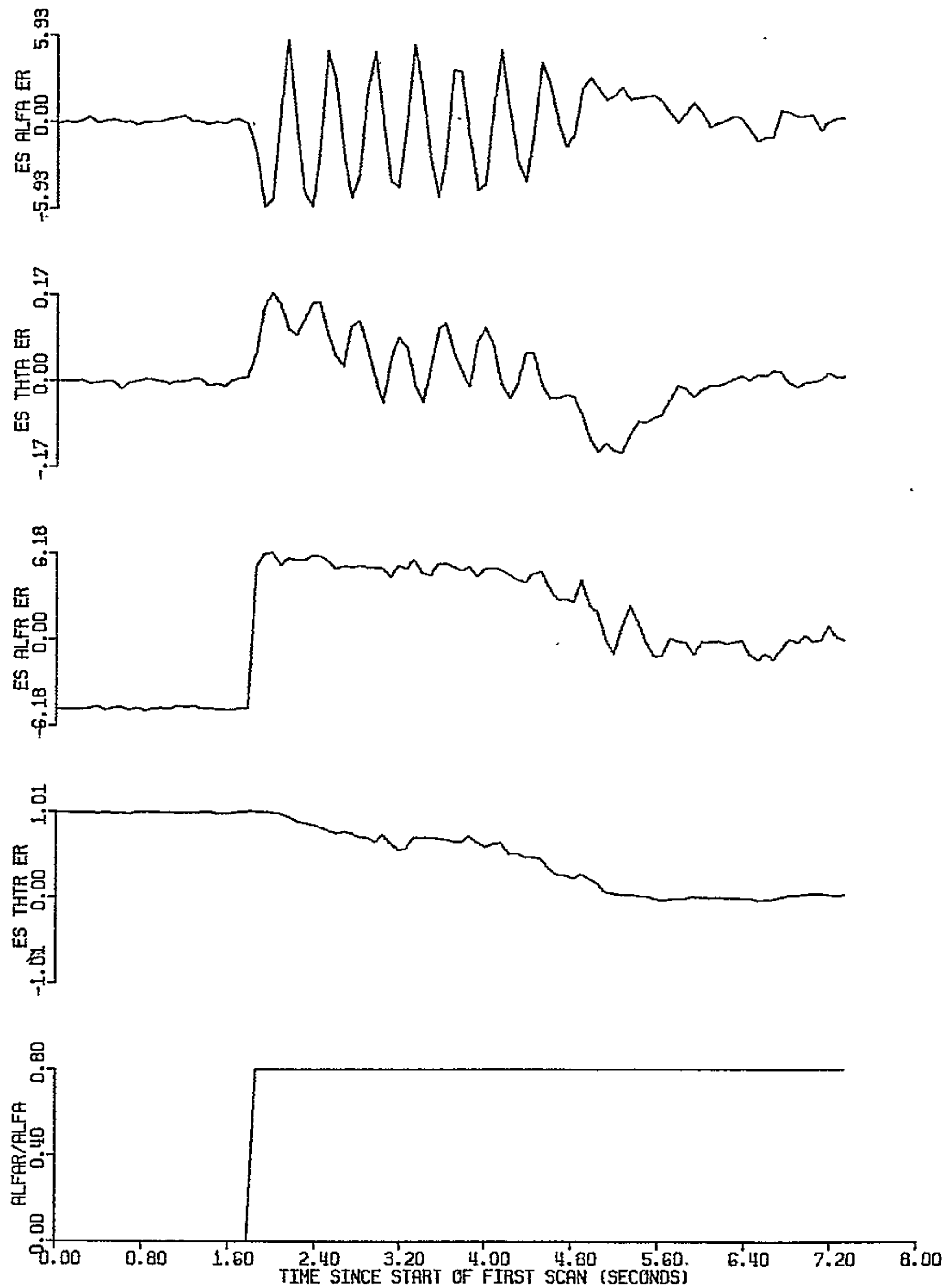


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 20.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= 3.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= 6.88 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= .174 HZ

Figure 3.17 Interference Acquisition, OPTRVR, S/N=20db, $\rho=0.8$, $\theta_{sep}=3^o$, $\beta=45^o$, $F_{sc}=0.6$Hz

SIM. JOB: HUGHOVC FILE NO: 512211U308 DATE: 04/03/79

PLOT JOB: HUGHOM1 PROGRAM: ACQMP1 DATE: 04/05/79

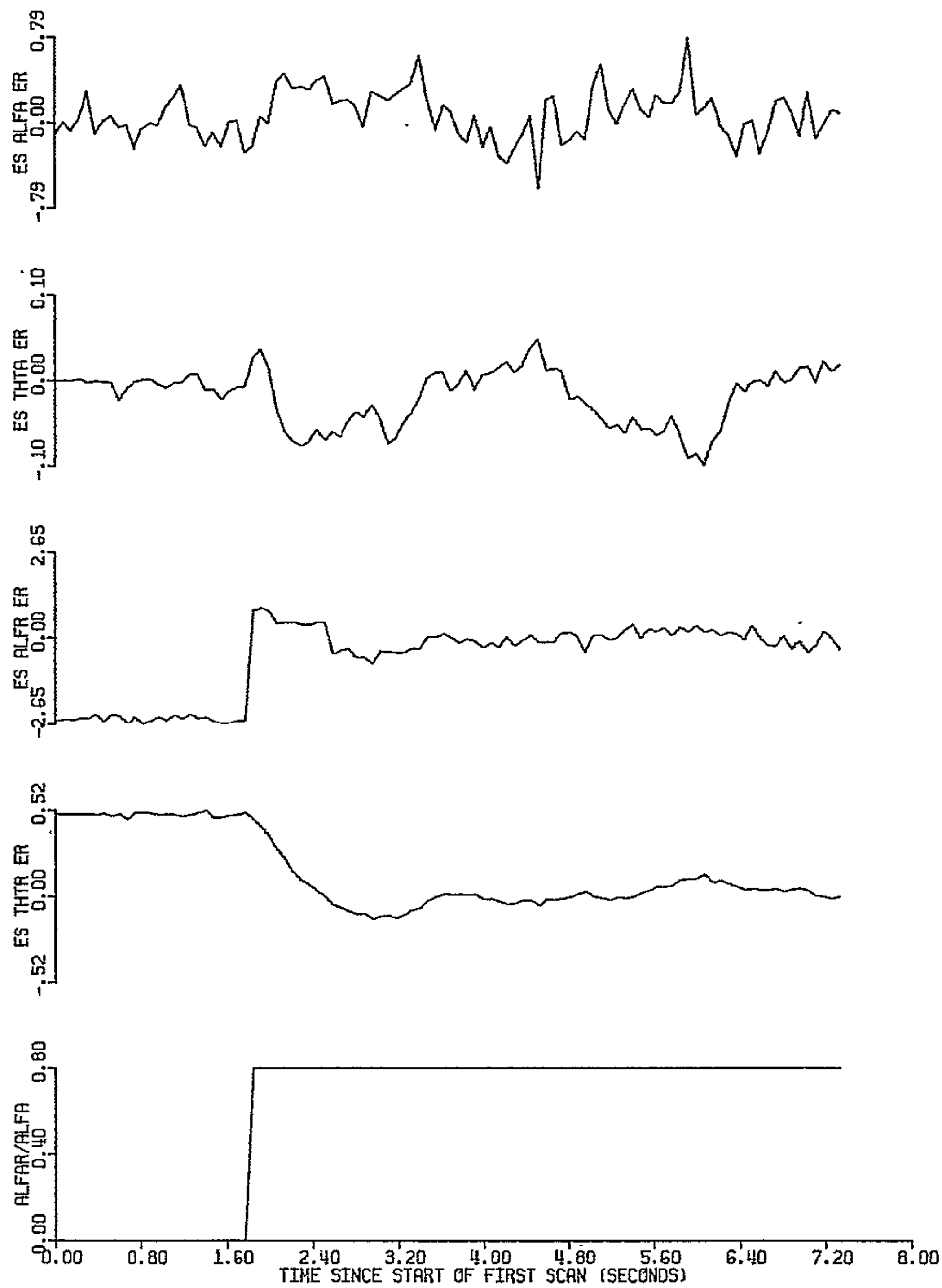


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 20.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, 2.500 HZ, THESEP= .500 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD., POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= -10.4 DEG
FSC ESTIMATE ERROR: INITIAL= 2.50 ,FINAL= -.257 HZ

Figure 3.18 Interference Acquisition, OPTRVR, S/N=20db, ρ=0.8, θ_{sep}=0.5^o, β=45^o, F_{sc}=2.5Hz

SIM. JOB: HUGHOUA FILE NO: 512211U306 DATE: 04/03/79

PLOT JOB: HUGHOM1 PROGRAM: ACQMP1 DATE: 04/05/79

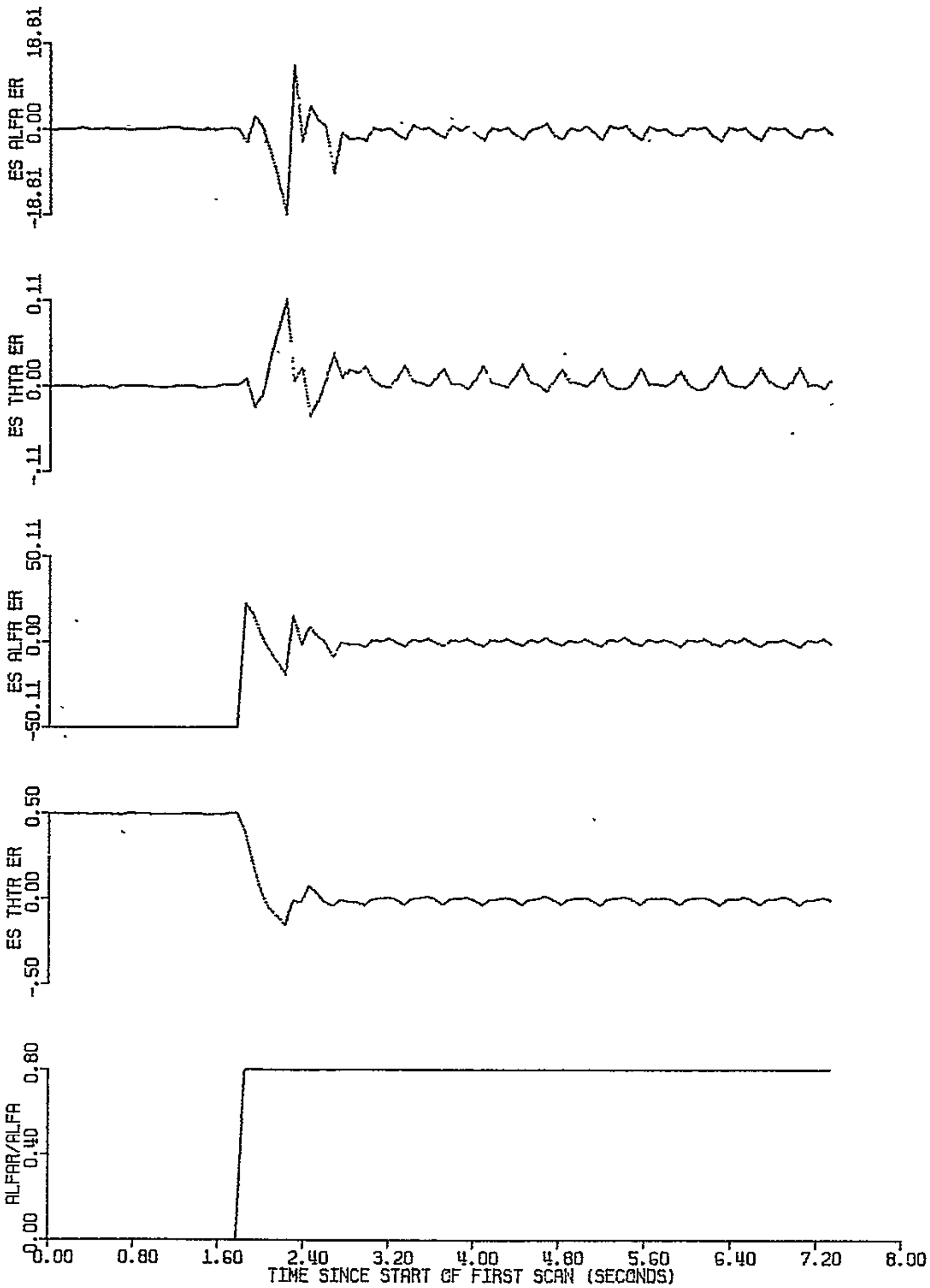


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 14.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= 1.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= -45.0 ,FINAL= -1.64 DEG
FSC ESTIMATE ERROR: INITIAL= .600 ,FINAL= -.482E-01HZ

Figure 3.19 Interference Acquisition, OPTRVR, S/N=14db, ρ=0.8, $\theta_{sep}=1^{\circ}$, $\beta=45^{\circ}$, F_{sc}=0.6Hz

SIM. JOB: HALENPY FILE NO: 5122110309 DATE: 05/02/79

PLOT JOB: HUGHOSY PROGRAM: ACOMP1 DATE: 05/02/79



SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA=-168.0 DEG,FSC=, 51.300 HZ, THESEP= 1.000 DEG
RECEIVER: OPTIML, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG
INTERFERENCE TRACKER IDLER VALUES: .5, 1.5 DEG, 0.0 DEG/SEC, 90.0 DEG, 0.0 HZ
BETA ESTIMATE ERROR: INITIAL= 78.0 ,FINAL= 5.11 DEG
FSC ESTIMATE ERROR: INITIAL= 51.3 ,FINAL= 50.6 HZ

Figure 3.20 Interference Acquisition, OPTRVR, S/N=40db, ρ=0.8
$\theta_{sep}=1^{\circ}$, $\beta= -168^{\circ}$, F_{sc}=51.3Hz

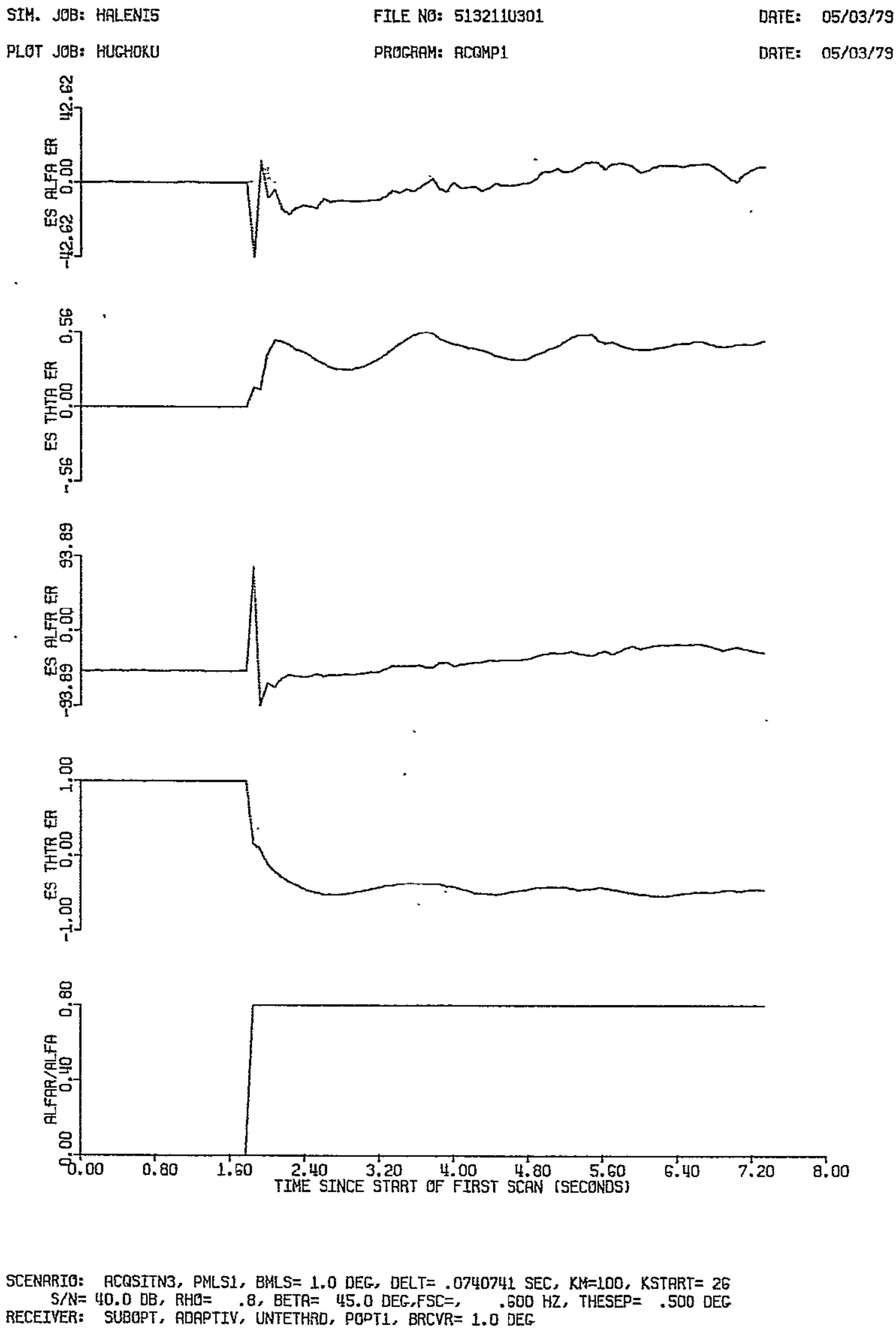
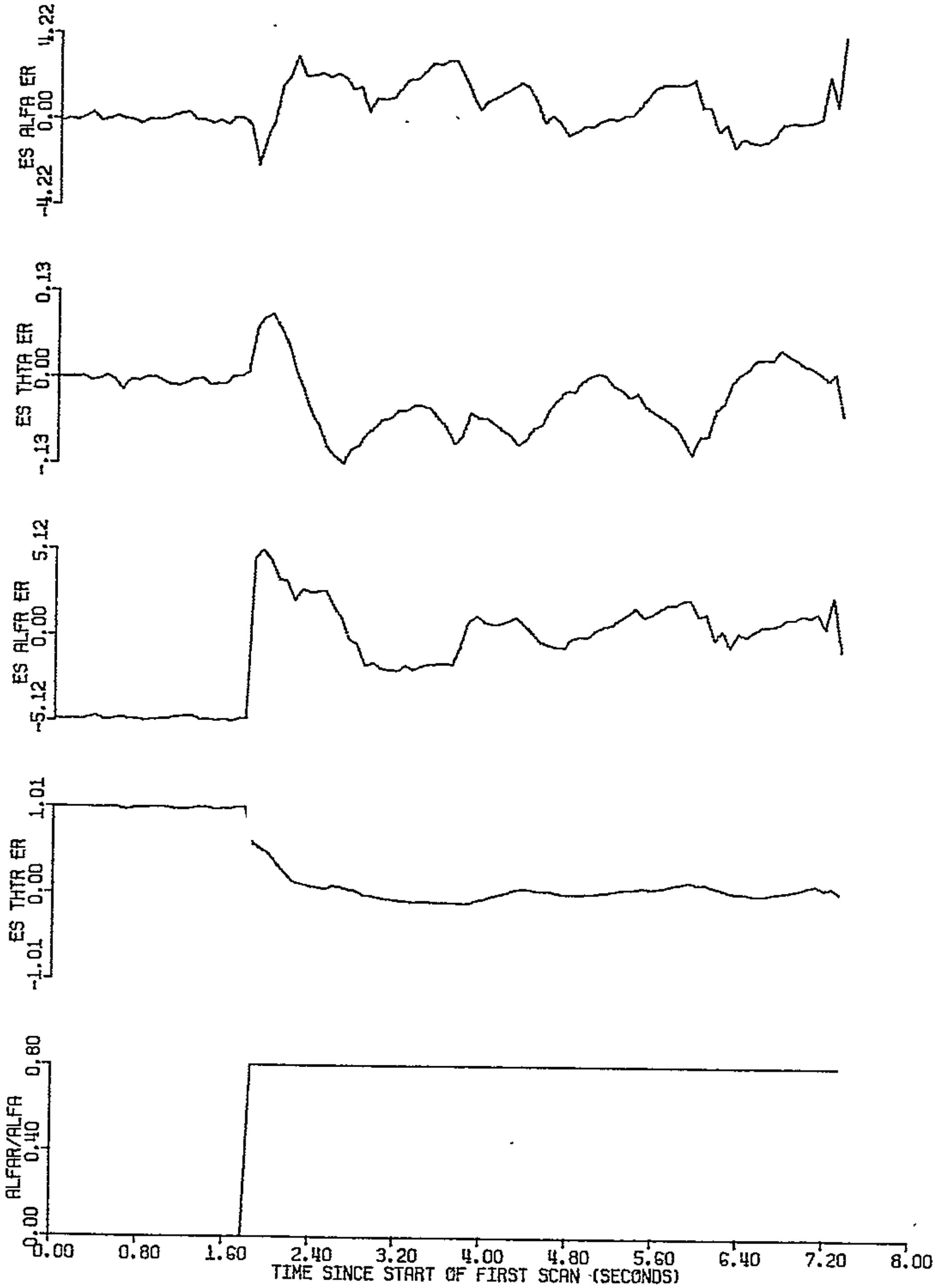


Figure 3.21 Interference Acquisition, SUBOPT, S/N=40db, ρ=0.8, $\theta_{sep}=0.5^{o}$, β=45^{o}, F_{sc}=0.6Hz

SIM. JOB: HALENHO FILE NO: 5132110303 DATE: 05/03/79

PLOT JOB: HUGHOKU PROGRAM: ACQMP1 DATE: 05/03/79

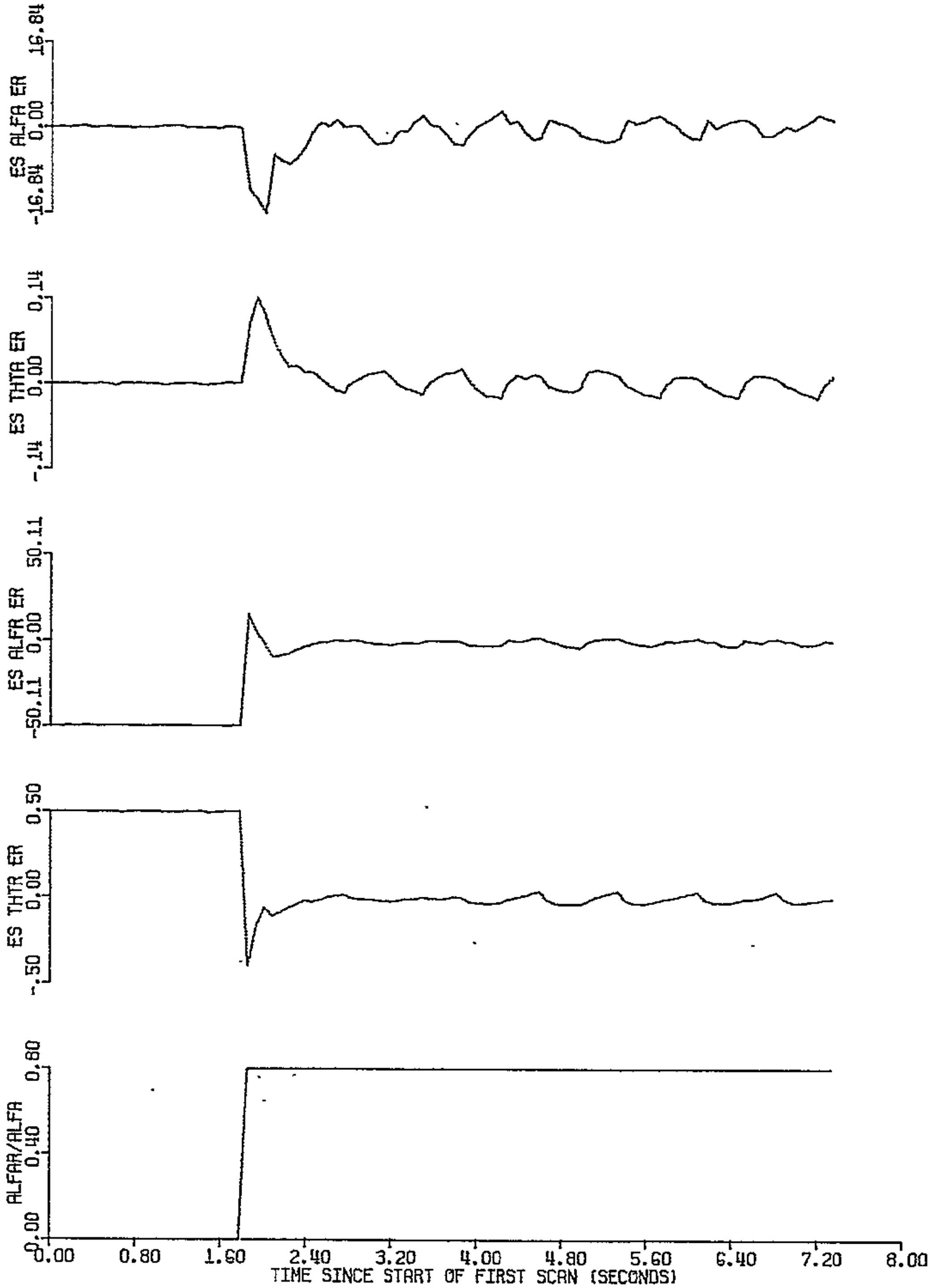


SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 20.0 DB, RHO= .8, BETA= 45.0 DEG,FSC=, .600 HZ, THESEP= .500 DEG
RECEIVER: SUBOPT, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG

Figure 3.22 Interference Acquisition, SUBOPT, S/N=20db, ρ=0.8, θ_{sep} = 0.5°, β=45°, F_{sc} = 0.6 Hz

SIM. JOB: HALENIT FILE NO: 5132110302 DATE: 05/03/79

PLOT JOB: HUGHOKU PROGRAM: ACOMP1 DATE: 05/03/79



SCENARIO: ACQSITN3, PMLS1, BMLS= 1.0 DEG, DELT= .0740741 SEC, KM=100, KSTART= 26
S/N= 40.0 DB, RHO= .8, BETA=-168.0 DEG,FSC=, 51.300 HZ, THESEP= 1.000 DEG
RECEIVER: SUBOPT, ADAPTIV, UNTETHRD, POPT1, BRCVR= 1.0 DEG

Figure 3.23 Interference Acquisition, SUBOPT, S/N=40db, ρ=0.8, $\theta_{sep}=1^{\circ}$, $\beta= -168^{\circ}$, 51.3 Hz

a. S/N = 40 db -- unsuccessful pull-in, in fact loss of track of the direct path pulse (Figure 3.21);

b. S/N = 20 db -- successful pull-in (Figure 3.22);

2. Successful pull-in @ S/N = 40 db, $\theta_{sep} = 1°$ and F_{sc} in the 5th lobe, again maximum enhancement of TO pulse, maximum cancellation of FRO pulse (Figure 3.23).

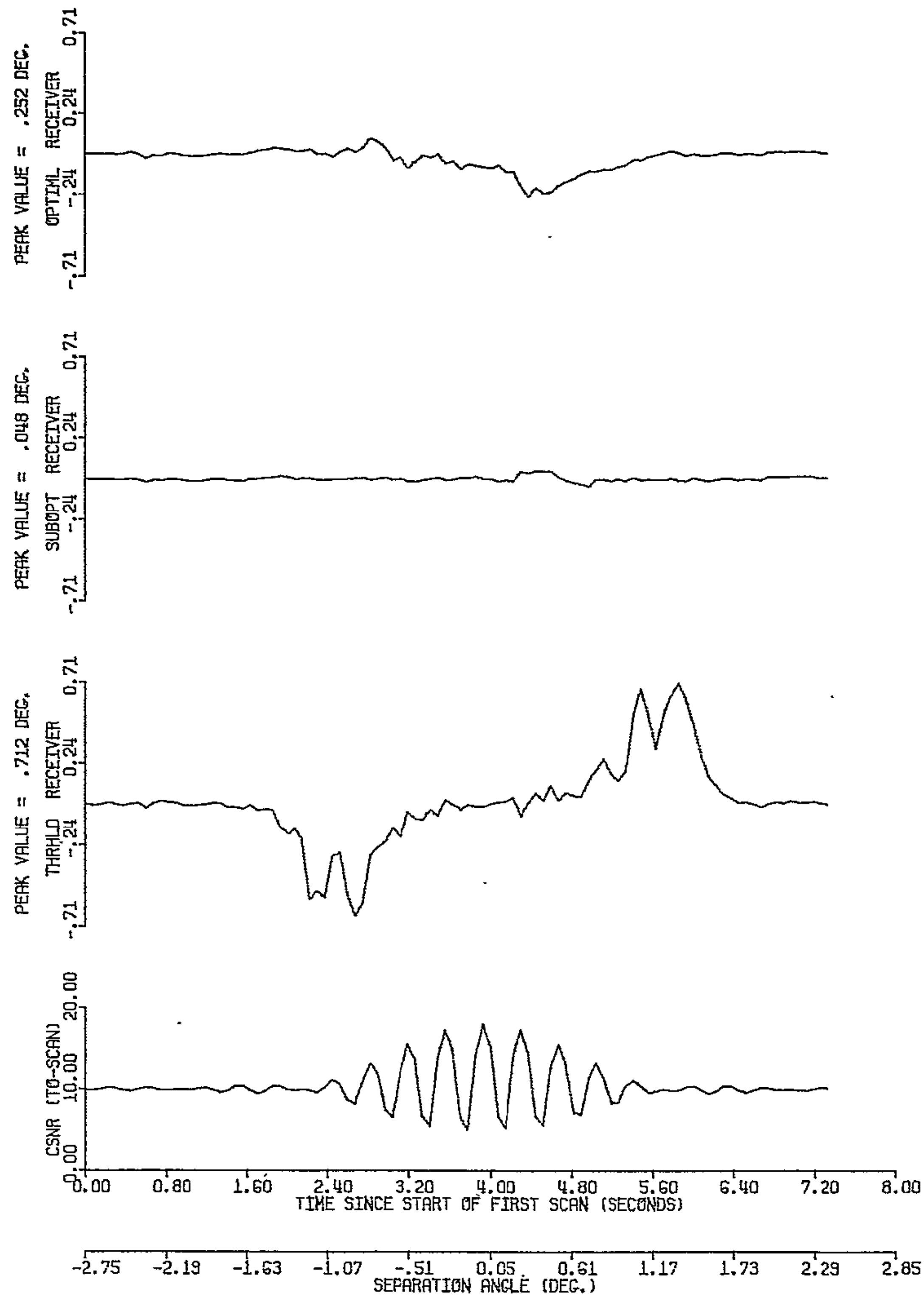
The poor performance shown in Figure 3.21 is probably attributable to the low scalloping rate, the successful pull-in of Figure 3.22 notwithstanding. The SUBOPT design is essentially premised upon on arbitrarily high scalloping rate, and under these circumstances the results of Figure 3.21 are probably more to be expected that those of Figure 3.22. The successful pull-in of Figures 3.22 is probably only a testament to the beneficial effects of noise acting as dither; or worse, it may be sample function dependent. More study here should be done.

Finally, Figures 3.24 and 3.25 provide comparisons of the three receivers OPTRVR, SUBOPT and the threshold receiver, in a crossing multipath scenario, beginning with the receivers in track and the interference initially out-of-beam. The same noise sample function was used to construct the envelope signal applied to the receiver in each case, and the error time histories are all plotted to the same scale. Two significant differences in the figures, one intentional, one not, are, as follows:

1. In Figure 3.24, the OPTRVR and SUBOPT designs include constraints, in Figure 3.25 these are without constraints.

2. In the two figures that OPTRVR and SUBOPT error time history traces are interposed; in Figures 3.24, that for OPTRVR is on top; in Figure 3.25, that for SUBOPT is on top.

SIM. JOB: HALENIA FILE NO: 1122110101 DATE: 05/03/79
HALENIU 1132110101 05/03/79
HALENJ5 1111100101 05/03/79
PLOT JOB: CRPLOLH PROGRAM: PCRMP1 DATE: 05/03/79



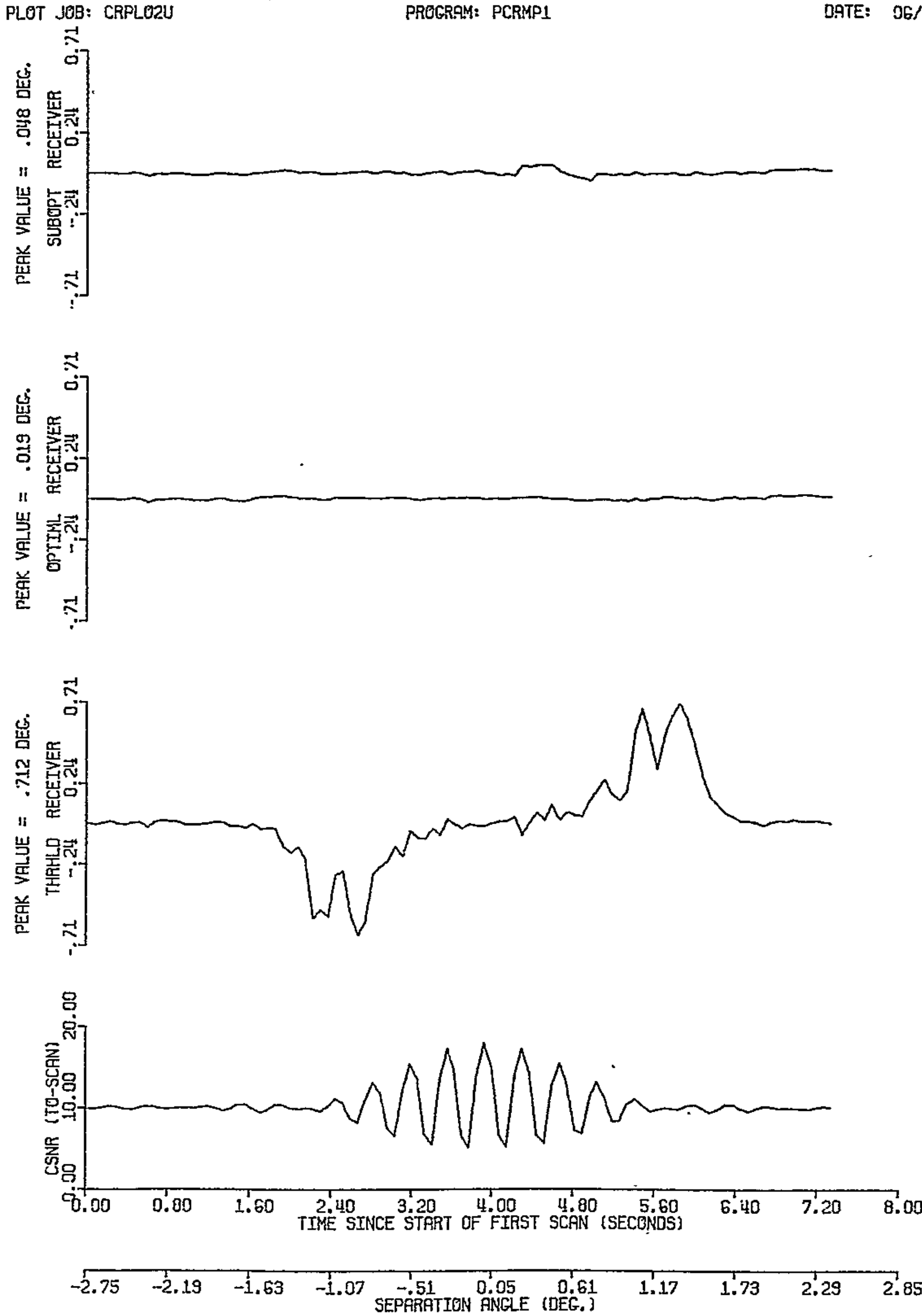
S/N= 20.0 DB, RHO= .80, BETA=-168.0 DEG, FSC= 51.3 HZ, KM= 100 SCANS, BMLS= 1.00, PMLS1

OPTIML: ADAPTIV, UNTETHRD, POPT1, BRCVR=1.00 DEG., ERMS= .690944E-01 DEG.
SUBOPT: ADAPTIV, UNTETHRD, POPT1, BRCVR=1.00 DEG., ERMS= .136560E-01 DEG.
THRHLD: -3 DB , UNTETHRD, 6.00% OF SCANS ABORTED, ERMS= .243510 DEG.

Figure 3.24 Crossing Multipath, OPTRVR and SUBOPT with constraints S/N=20db, ρ=0.8, β= - 168, F_{sc} = 51.3 Hz

SIM. JOB: MLSSUOG FILE NO: 1132110117 DATE: 05/24/78
MLSOPOF 1122110117 05/24/78
MLSTHYX 1111100117 05/23/78

PLOT JOB: CRPL02U PROGRAM: PCRMP1 DATE: 06/22/78



S/N= 20.0 DB, RHO= .80, BETA=-168.0 DEG, FSC= 51.3 HZ, KM= 100 SCANS, BMLS= 1.00, PMLS1

SUBOPT: ADAPTIV, UNTETHRD, POPT1, BRCVR=1.00 DEG., ERMS= .136560E-01 DEG.
OPTIML: ADAPTIV, UNTETHRD, POPT1, BRCVR=1.00 DEG., ERMS= .755801E-02 DEG.
THRHLD: -3 DB , UNTETHRD, 6.00% OF SCANS ABORTED, ERMS= .243510 DEG.

Figure 3.25 Crossing Multipath, OPTRVR and SUBOPT without constraints
$S/N=20db$, $\rho=0.8$, $\beta= - 168^{o}$, $F_{sc} = 51.3$ Hz

Another very significant difference is in the OPTRVR error time histories themselves -- that without constraints peaks at about 1/50°, and that with constraints, at about ¼°, worse than the SUBOPT response, which peaks at about 1/20° with and without constraints. Again, we suspect the $\hat{\omega}_{sc}$ constraint for the deterioration of the OPTRVR performance. Nevertheless, all optimal structure receivers performed better in this demonstration than did the threshold receiver, which peaked at 0.71°. The lowest trace shows the carrier signal-to-noise ratio CSNR, evaluated at the peak of the direct path pulse on the TO-scan, as the simulation evolves. The F_{sc} used here is in the 5th lobe, the exact β, F_{sc}, values used being again that combination producing maximum enhancement of the TO-pulse and maximum cancellation of the FRO-pulse at crossover.

In summary, a simulation study involving principally statistical error studies but also some error time histories has been carried out; a substantial collection of FORTRAN programs was developed. Simulation results show for the OPTRVR design

1. RMSE performance improvement factors wrt the threshold receiver sometimes approaching 30 at 20 db S/N;
2. Full track capability limited to scalloping rates below ½ of rep. rate (below 6.75 Hz in AZ).
3. Some deterioration in performance when constraints are used believed attributable to the $\hat{\omega}_{sc}$ constraint. Removal of this one constraint may restore the generally higher RMSE performance of the unconstrained OPTRVR without reducing the resistance to loss of track apparently characterizing the constrained receiver (see SUBOPT below). Also perhaps a depend-

able angle-tracking capability at higher scalloping rates might result from removal of the $\hat{\omega}_{sc}$-constraint.

The results show for the SUBOPT design

1. Lesser performance than the OPTRVR design, improvement factors wrt the threshold receiver sometimes approaching 15 at 20 db S/N;
2. Performance probably better at higher scalloping rates, deteriorating at lower rates;
3. Very definite improvement in track-holding ability at small separation angles when the receiver constraints are applied. There is no ω_{sc}-estiamte, hence no $\hat{\omega}_{sc}$-constraint in the SUBOPT design.

Based on studies of the OPTRVR, we expect both the OPTRVR and SUBOPT design performance to be quite sensitive to error, in the receiver model, of the transmitting antenna selectivity function, particularly the beam width parameter. Possible approaches to reduce the effects on performance were discussed. Finally, in error time history studies

1. Strong interference pull-in capabilities at various S/N's and F_{sc}'s were demonstrated, which might be used in an interference acquisition scheme; and
2. A high performance in a representative crossing multipath situation was demonstrated, and is to be generally expected, particularly if the removal of the $\hat{\omega}_{sc}$-constraint will have the effects expected and discussed.

SECTION IV

EXPERIMENTAL SYSTEM CONSIDERATIONS

An experimental receiver development project was included in the original research proposal and begun in 1976 in parallel with tracking algorithm development then in progress. The project was short-lived, however, and abandoned the following year, principally because the computational demands of the evolving tracking algorithms simply could not be met with any economically feasible microcomputer that was available at the time. The general design philosophy and the allocation of tasks among

1. the interface hardware
2. the foreground software
3. the background software

were described in [3, pp. 52-60]; the approach to the interface design, involving specifically a state controller, is conveyed in Appendix B of this final report.

At the present time, both the MLS tracking algorithm development and the state-of-the-art in microcomputing are more advanced. It seems safe to conjecture, in conclusion, that one way the requisite computer power might be obtained economically, certainly in the near future, is to make use of a bank, or an array, of now diminishingly expensive microprocessors. One or more microprocessors might serve the executive function, allocating the resources provided by the others to the various computational needs as they arise. Advantages of such an arrangement might be, as follows:

1. It would be able to exploit the large potential for parallel computation in the tracking receiver calculations;
2. It would have high protection against total system failure due to isolated failures;
3. It would easily accomodate the randomized repetition rates in the MLS.

SECTION V

SUMMARY AND CONCLUSIONS

This report has described research performed at the University and concerned with optimal MLS receiver theory, design and simulation evaluation. The program has produced a general receiver structure which, it is believed, gives close to limiting performance; it consists of a Scan Data Processor (SDP) based on the theory of Locally Optimum Estimation enclosed in a Tracking Loop Filter (TLF) based on MMSE recursive state estimation. Three concrete specializations of the general structure were carried out, characterized by both the dimension of the Scan Data Processor and the method of extraction of phase difference (β) and scalloping frequency (ω_{sc}) information, as follows:

5D SDP: Denoted the Optimal design (OPTRVR); SDP extracts β information from each scan; TLF extracts ω_{sc} information from the sequence of scans;

4D SDP: Denoted the Suboptimal design (SUBOPT); Both β and ω_{sc} are suppressed from the model and not estimated,

6D SDP: Denoted the 6D LOE design; SDP extracts β and ω_{sc} information from each scan; TLF may optionally extract $\dot{\omega}_{sc}$ information.

Two of these, the OPTRVR and SUBOPT designs were studied extensively in simulation studies, including:

a. Statistical error studies under various conditions;

b. Studies of interference acquisition capability using "pull-in",

c. Performance evaluations with "crossing" multipath interference.

and comparisons of performances were made with a simulated "threshold" receiver approximately representative of the Phase III design. Finally, a limited effort to implement an experimental receiver was undertaken, but was aborted with little more result than the tentative approach planned and the insights gained.

In the simulation studies of RMSE the OPTRVR and SUBOPT designs showed improvements over the threshold receiver by factors approaching 30 and 15, respectively, at low scalloping rates. Constraints were imposed on the computed estimates, based largely on natural limitations on the true state values, and as a result, improved ability to maintain track was noted in the SUBOPT design at small separation angles and moderate S/N. At higher values of scalloping rate, ω_{sc}, the SUBOPT design was superior to the OPTRVR with constraints. The data suggests, however, that the $\hat{\omega}_{sc}$-constraint in OPTRVR may be hurting performance some, though even with its removal, successful full-state tracking can't be expected in OPTRVR in the strict sense at higher scalloping frequencies. A high sensitivity to error in the receiver model of the MLS transmitting antenna selectivity function, particularly the beam width parameter, was observed in the SUBOPT design (and believed to be characteristic of the OPTRVR design also). Approaches were discussed to reduce this problem. Both the OPTRVR and SUBOPT designs exhibited sufficient multipath "pull-in" capability to merit a "pull-in" approach to interference acquisition. In summary, the receiver designs developed in this research have demonstrated in simulation superior levels of performance for MLS receivers. Some areas of consideration need further study, however, to fine-tune the designs to the signal and state dynamics environment and other aspects of the application. On the basis of available data the SUBOPT design appears to be the more appropriate for the MLS aircraft receiver application, where the scalloping rates experienced are generally greater than one-half the MLS angle function repetition rate, and thus beyond the full-track capability of the OPTRVR design.

From the viewpoint of implementation, it would appear that, on the basis of trends in the microprocessor art,

1. Input of the envelope samples to the computer should be made by direct memory access (DMA), and
2. The comptuer should, perhaps, be a bank of microprocessors performing many operations in parallel (to supply the requisite computing power) and operating as a unified system under an executive function microprocessor.

Finally, an operational receiver should probably be structurally adaptive also with an ability to expand the state vector (by appending a "sentry" tracker channel tethered to "idler" values) as additional interference pulses are recognized and acquired. The "sentry" should move from a position one side of the sampling window to the opposite side and then back on alternate scans to preserve the integrity of the main pulse track. Also when the separation angle for an interference pulse drops below a certain threshold and information needed to distinguish the pulse becomes less available, track of that interference pulse should then be dropped and the state dimension suitably decremented.

PRECEDING PAGE BLANK NOT FILMED.

REFERENCES

1. McAlpine, G. A., Highfill, J. H., III, Irwin, S. H., and Padgett, J. E., "Optimization of MLS Receivers for Multipath Environments," Rpt. #EE-4033-101-75, Research Laboratories for the Engineering Sciences, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, 22901, Dec. 1975.

2. McAlpine, G. A., Highfill, J. H., III, and Irwin, S. H., "Optimization of MLS Receivers for Multipath Environments," Rtp. #EE-4633-102-76, Research Laboratories for the Engineering Sciences, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, 22901, March 1976.

3. McAlpine, G. A., Highfill, J. H., III, and Irwin, S. H., "Optimization of MLS Receivers for Multipath Environments," Rpt. # UVA/528062/EE76/103, Research Laboratories for the Engineering Sciences, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, 22901, Dec. 1976.

4. McAlpine, G. A., Highfill, J. H., III, Irwin, S. H., Nelson, R. A., and Koleyni, G., "Optimization of MLS Receivers for Multipath Environments," Rpt. # UVA/528063/EE77/104, Research Laboratories for the Engineering Science, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia 22901, Nov. 1977.

5. McAlpine, G. A., Highfill, J. H., III, Tzeng, C. P. J., and Koleyni, G., "Optimization of MLS Receivers for Multipath Environments," Rpt. # UVA/528062/EE78/106, Research Laboratories for the Engineering Science, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia 22901, Oct. 1978.

6. MLS Scanning Beam Working Group, "MLS Technique Assessment," Final Report, Prepared for DOT, FAA, December 1974 (Binders 1 & 2 and Appendices).

7. "MLS Functional Requirements Specification," Report No. FAA-ER-700-07A, DOT, FAA, July 25, 1975.

8. "MLS Signal Format Specificatin for the Time Reference Scanning Beam Guidance System," Report No. FAA-ER-700-08A, DOT, FAA, May 30, 1975.

9. Murphy, James T., Jr., "Locally Optimum Estimation with Applications to Communication Theory," Ph.D. Dissert., Univ. of Florida, 1968.

10. Davenport, W. B., Jr., and Root, W. L., An Introduction to the Theory of Random Signals and Noise, McGraw-Hill, New York, 1958.

11. Kelley, R. J., "Time Reference Microwave Landing System Multipath Control Techniques," Navigation, 23:1, Spring 1976, pp. 42-58.

APPENDIX A

COMPUTER PROGRAMS

Programs in this appendix are listed below in the order presented, generally alphabetized by Module name in two groups. Programs within each Module are listed with a brief characterization of the Module. Programs for the 6D LOE design are given in Appendix C.

SIMULATION PROGRAMS

CTLACQN3:	CONTRL, Interference Acquisition Scenario
CTLCRMP:	CONTRL, Crossing Multipath Scenario
CTLMSBB1:	CONTRL, RMSE (B_{ratio})
CTLMSFS1:	CONTRL, RMSE (F_{sc})
CTLMSFS2:	CONTRL, RMSE (F_{sc}), ($\hat{\beta}$, $\hat{\omega}_{sc}$) tethered to ($\pi/2$,0)
CTLMSTH2:	CONTRL, RMSE (θ_{sep}), averaging with $F_{sc}\neq 0$
CTLOE:	CONTRL, RMSE (θ_{sep}), averaging with $F_{sc}=0$
MLSSIM:	MLSSIM, MLSSUB, THA, DFLTR1, MLS
OPTRNC:	RCVR, ORVRID, No constraints, except on $\hat{\beta}$
OPTRVR:	RCVR, ORVRID, with constraints
PLSUB:	PHILM, WAWBJ, SWFCNS, PLUSID, Suboptimal SDP
PLOPT:	PHILM, PLOPID, Optimal SDP
PMLS1:	PMLS, PMLSID, p-function used in generating signal
POPT1:	P, PDOT, POPTID, p-function presumed in receiver
THDRVR:	RCVR, TRVRID, threshold receiver

UTILITY AND SERVICE PROGRAMS

ACQMP1:	ACQMP1, Acquisition plot generator
LABLIB:	CLIP, GAUSS, INTIO, LOG10, MATIN, MATMUL, MATOUT, MATSM, MULPLT, PLOTR, PVALUE, REALIO, RETURN, SATU, library of utility routines
PCRMP1:	PCRMP1, Crossing multipath plot generator
RLOGSW:	RLOGSW, Reads and lists out SWAJ, SWBJ, & RABJ data from file
WLOGSW:	WLOGSW, WAWB, WAVGS, computes SWAJ, SWBJ, & RABJ data and writes files

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C CROSSING MULTIPATH SCENARIO
C
      REAL LAMDA
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER YNAM1(2),YNAM2(2),YNAM3(2),YNAM4(2),YNAM5(2),FUNCTN(2)
      INTEGER YNAM7(2),YNAM8(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV
      LOGICAL FILOUT,FILEIN
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(6,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,ED(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION IDNRS(6),IDASCI(24),IDENTS(6),RMSVAL(8)
      DIMENSION EALFA(100),ETHET(100),EALFR(100),ETHER(100),XFOUR(100)
      DIMENSION          X(8),         SCANNR(100)
      DIMENSION EBETA(100),EFSC(100)
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H50000OF000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7HACQSITN,
     *3*7H       ,7HGENERAL,8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/26/,FILOUT/.FALSE./,
     CFILEIN/.FALSE./
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
      DATA XNAME/8H  SCAN  ,8H NUMBER /,YNAM2/8H  ETHET ,8HDEGREES /
      DATA YNAM1/8H  EALFA ,8H        /,YNAM3/8H EALFAR ,8H        /
      DATA YNAM4/8H  ETHER ,8HDEGREES /,YNAM5/8H  RHO   ,8H        /
      DATA YNAM7/8HEABSBETA,8H DEGREES/
      DATA YNAM8/8HEABSFSC ,8H  HERTZ /
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
    1 FORMAT(1H )
   10 FORMAT(9(/))
   11 FORMAT (1H1)
C
C*********************************************************************
C
```

```
      100 CONTINUE
C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
CTHE FOLLOWING RECEIVER PARAMETERS
CSIGMA,IAC,DSNRDB,RHO,FSC,BETA,JM,BMLS
          IF(NFIRST) GO TO 155
          NOAC=.FALSE.
          NFIRST=.TRUE.
          JBNAM=JOBNAME(I)
          ISIM=5
          IMOD=3
          WRITE (7,110)
      110 FORMAT(34H1INTERFERENCE ACQUISITION SCENARIO/)
          IF (IRCVR.NE.1) GO TO 118
          WRITE (7,115)
      115 FORMAT (35H THIS IS NOT FOR THRESHOLD RECEIVER)
          STOP * ABORTED--NOT FOR THDRVR*
      118 CONTINUE
          WRITE(7,120) (DATIN(I),I=1,2)
      120 FORMAT(26H THIS MAY READ INPUT FILE ,2A10/)
          CALL LOGIO(6HFILEIN,FILEIN,0)
          IF(.NOT.FILEIN) NSTART=0
          CALL INTIO(6HNSTART,NSTART,100,0)
          NSTOP=NSTART
          CALL INTIO(6H NSTOP,NSTOP,100,0)
          CALL LOGIO(6HFILOUT,FILOUT,0)
          CALL LOGIO(6HTETHRD,TETHRD,0)
          CALL LOGIO(6HADAPTV,ADAPTV,0)
          IF(TETHRD) ITETHR=2
          IF(ADAPTV) GO TO 135
          IRSIGN=-1
          IADAP=3
      135 CONTINUE
          DO 145 I=2,6
          IASC=3*(I-1)+IDNRS(I)+6
          IDENTS(I)=IDASCI(IASC)
      145 CONTINUE
          ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
      150 FORMAT(A7,I1)
          WRITE (7,152) IDNRS,IMOD
      152 FORMAT (1H ,6I1,1X,I1/)
          WRITE (7,153) IDENTS
      153 FORMAT (1H ,A8)
          CALL REALIO (6HRHOMAX,RHOMAX,0)
          CALL REALIO(6H  DTHO,DTHO,0)
          CALL REALIO(6H  TORO,TORO,0)
          BETAO=BO*180./PI
          FSCO=WSCO*.5/PI
          CALL REALIO(6H BETAO,BETAO,0)
          CALL REALIO(6H  FSCO,FSCO,0)
          BO=BETAO*PI/180.
          WSCO=FSCO*(2.*PI)
      155 CONTINUE
          IF (FILEIN) GO TO 158
          NRUN=0
          DSNRDB=20.
          RHO=0.5
          BETA=45.
```

```
              FSC=0.
              BMLS=1.
              BRCVR=1.
              THESEP=1.88
              CALL INTIO(6H  NRUN,NRUN,100,0)
              CALL REALIO(6HDSNRDB,DSNRDB,0)
              CALL REALIO(6H   RHO,RHO,0)
              CALL REALIO(6H  BETA,BETA,0)
              CALL REALIO(6H   FSC,FSC,0)
              CALL REALIO(6H  BMLS,BMLS,0)
              CALL REALIO(6H BRCVR,BRCVR,0)
              CALL REALIO(6HTHESEP,THESEP,0)
              CALL INTIO(6HKSTART,KSTART,100,0)
              MORE=.FALSE.
              GO TO 170
        158   CONTINUE
              READ(15,160) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP,KSTART
        160   FORMAT (I5,7G10.3,I5)
              IF(NRUN.LT.NSTART) GO TO 158
              IF(NRUN.GE.NSTOP) MORE=.FALSE.
        170   CONTINUE
              XO(3,IAE)=XO(6,IAE)=0.
              XO(5,IAE)=XO(2,IAE)-THESEP
              EBETAO=ABS(BETA)-ABS(BETAO)
              EFSCO=ABS(FSC)-ABS(FSCO)
              RETURN
        C
        C*************************************************************************
        C
        200   CONTINUE
        C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
        CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
              WRITE (7,209)
        209   FORMAT(5H1NRUN,4X,6HDSNRDB,7X,3HRHO,8X,4HBETA,8X,3HFSC,9X,
             C4HBMLS,8X,5HBRCVR,7X,6HTHESEP,6X,6HKSTART)
              WRITE(7,210) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP,KSTART
        210   FORMAT(1HO,I4,7(2X,G10.3),2X,I4)
              IF(.NOT.FILOUT) GO TO 245
              IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1HO,NRUN
        220   FORMAT(6I1,A1,I1,A1,I1)
              IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
        230   FORMAT(6I1,A1,I1,I2)
              IX=IREQST(6LTAPE17,3L*PF)
              IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
              IF(IX.NE.0) STOP≠OUTPUT FILE REQUEST NOT SATISFACTORY≠
              WRITE(7,240) (DATOUT(I),I=1,2)
        240   FORMAT(25HOTHIS WRITES OUTPUT FILE ,2A10)
        245   CALL DATE(TODAY)
              WRITE (7,250) FUNCTN (IAE),IAE
        250   FORMAT (1HO,A7,15H FUNCTION (IAE=,I1,1H))
              X(4)=XO(4,IAE)=0.
              WRITE (7,260) ((I,X(I)),I=1,8)
        260   FORMAT (15HOINITIAL STATE://(3H X(,I1,4H) = ,G13.6))
              WRITE(7,270) IDENTS(2)
        270   FORMAT(1HO,A7)
              RETURN
```

```
C*******************************************************************
C
300   CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE UNLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
      ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
      IF(IRSIGN.LT.0) IR=-IR
C
      RETURN
C
C*******************************************************************
C
400   CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
      WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
410   FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
     *         4H,NS=,I1,1H))
      WRITE(7,420) (IDENTS(I),I=5,6)
420   FORMAT(1H0,A8/1H0,A7//)
      RETURN
C
C*******************************************************************
C
500   CONTINUE
C THIS SETS-UP THE MSET-LOOP
      RETURN
C
C*******************************************************************
C
600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
      RETURN
C
C*******************************************************************
C
700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
      CALL INTID(6H    KM,KM,1,1)
      WRITE(7,710)
710   FORMAT (4H0  K,          3X,6HCSNRTD,5X,6HCSNRFR,6X,3HQTY,7X,
     *         4HALFA,5X,5HTHETA,6X,6HTHEDOT,5X,5HALFAR,6X,6HTHETAR,5X,
     C         6HTHRDOT,6X,4H B  ,7X,3HWSC/)
      RETURN
C
C*******************************************************************
C
800   CONTINUE
C THIS INITIALIZES THE K-TH SCAN
      IF(K.EQ.KSTART) X(4)=RHO*X(1)
      IF(K.EQ.KSTART) NGM=NG
      RETURN
C
C*******************************************************************
C
```

```
  900 CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
      THERR=X(2)-XS(2)
      DTHE=X(2)-X(5)
      COLB=SQRT(PPDIAG(2))
      DO 910  I=1,NS
      ES(I)=X(I)-XS(I)
      RMSVAL(I)=SQRT(PPDIAG(I))
  910 CONTINUE
      IF (NS.GE.7) ES(7) = ABS(X(7))-ABS(XS(7))
      IF (NS.GE.8) ES(8) = ABS(X(8))-ABS(XS(8))
      SCANNR(K)=K
      EALFA(K)=ES(1)
      ETHET(K)=ES(2)
      EALFR(K)=ES(4)
      ETHER(K)=ES(5)
      XFOUR(K)=X(4)/X(1)
      IF (NS.LE.6) GO TO 915
      EBETA(K)=180.*ES(7)/PI
      EFSC(K)=0.5*ES(8)/PI
  915 CONTINUE
      WRITE (7,920) K,     CSNRT,CSNRF,1HX,(X(I),I=1,8)
  920 FORMAT(1H ,I3,        2(1X,G10.3),8X,A1,1X,8(1X,G10.3))
      WRITE (7,930) 7H      XS,(XS(I),I=1,NS)
      WRITE (7,930) 7H     ERR,(ES(I),I=1,NS)
      WRITE (7,930) 7HSQ(PII),(RMSVAL(I),I=1,NS)
  930 FORMAT (1H ,27X,A7,1X,8(1X,G10.3))
      WRITE (7,1)
      RETURN
C
C*********************************************************************
C
 1000 CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
      WRITE(7,1060) (GQGT(I,I),I=1,NS)
 1060 FORMAT (10(/),35H ON LAST SCAN, DIAGONAL OF GQGT WAS/
     *1H ,G11.4,7(1H,,G11.4))
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL PLOTR(SCANNR, EALFA,KM,XNAME,YNAM1,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL PLOTR(SCANNR, ETHET,KM,XNAME,YNAM2,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL PLOTR(SCANNR, EALFR,KM,XNAME,YNAM3,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL PLOTR(SCANNR, ETHER,KM,XNAME,YNAM4,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
```

```
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL PLOTR(SCANNR, XFOUR,KM,XNAME,YNAM5,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      IF (NS.LE.6) GO TO 1065
      WRITE (7,11)
      YMIN=YMAX=0.
      CALL PLOTR(SCANNR,EBETA,KM,XNAME,YNAM7,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE(7,11)
      CALL PLOTR(SCANNR,EFSC,KM,XNAME,YNAM8,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
 1065 CONTINUE
      IF(.NOT.FILOUT) RETURN
      WRITE(7,11)
      WRITE(7,1070) (DATOUT(I),I=1,2)
 1070 FORMAT(13H OUTPUT FILE ,2A10,1H:/)
      WRITE(7,1072) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
 1072 FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,1074) IDENTS,KSTART
 1074 FORMAT(1H ,6(1X,A8,9X),I3/)
      WRITE(7,1076) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP
 1076 FORMAT(1H ,I2,3X,7(3X,G12.6)/)
      IKM=MIN0(35,KM-9)
      DO 1077 I=1,IKM
 1077 WRITE(7,1078)EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
 1078 FORMAT(1H ,5(G13.6,5X))
      DO 1079 I=1,5
 1079 WRITE(7,1080)
 1080 FORMAT(1H,5(6X,1H.,11X))
      KM3=KM-3
      DO 1081 I=KM3,KM
 1081 WRITE(7,1078)EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
      WRITE(7,1083)RHOMAX,DTHO,TDRO,BETAO,FSCO,EBETAO,EBETA(KM),EFSCO,
     CEFSC(KM)
 1083 FORMAT (1H0,9(G12.6,2X))
      WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      WRITE(17) IDENTS,KSTART
      WRITE(17) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP
      DO 1090 I=1,KM
      WRITE(17) EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
 1090 CONTINUE
      WRITE(17) RHOMAX,DTHO,TDRO,BETAO,FSCO,EBETAO,EBETA(KM),EFSCO,EFSC(
     CKM)
      REWIND 17
      IX=IATTACH(6LTAPE20,DATOUT)
      CALL RETURN(6LTAPE20)
      IF(IX.NE.0) GO TO 1095
      IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
      GO TO 1096
 1095 IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1096 IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP≠OUTPUT FILE CATALOG NOT SATISFACTORY≠
```

```
            CALL RETURN(6LTAPE17)
            WRITE (7,1099) (DATOUT(I),I=1,2)
      1099  FORMAT(13HCOUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
            RETURN
      C
      C**********************************************************************
      C
      1100  CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
      CLGMAX SETS OF KM SCANS
            RETURN
      C
      C**********************************************************************
      C
      1200  CONTINUE
      C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
            WRITE (7,11)
            RETURN
      C**********************************************************************
      C
      C
            END
44030B CM STORAGE USED        6.147 SECONDS
```

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C CROSSING MULTIPATH SCENARIO
C
      REAL LAMDA
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER TITLE1(3),TITLE2(2,2),FUNCTN(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV
      COMMON/IDDATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TORO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION CSNRTO(100),X(8),THESER(100),THESEP(100),ABBORT(100)
      DIMENSION IDNRS(6),IDASCI(18),IDENTS(6),CSNRFR(100)
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H100000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/15/,IRSIGN/1/
      DATA TITLE1/6H X(2) ,6HX2-X5 ,6HXS(2) /
      DATA TITLE2/6HFIL ER,6HX2-XS2,6HX2-XS2,6HSQRP22/
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
      DATA XNAME/8HTHETASEP,8HDEGREES /,YNAME/8HTHES ERR,8HDEGREES /
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
1     FORMAT(1H )
10    FORMAT(9(/))
11    FORMAT (1H1)
C
C********************************************************************
C
100   CONTINUE
C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
CTHE FOLLOWING RECEIVER PARAMETERS
CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JM,BMLS
      IF(NFIRST) GO TO 155
      NFIRST=.TRUE.
      JBNAM=JOBNAME(I)
      ISIM=1
```

ORIGINAL PAGE IS OF POOR QUALITY

```
                 IMOD=1
                 WRITE (7,110)
           110   FORMAT (28H1CROSSING-MULTIPATH SCENARIO/)
                 GO TO 122
           COLD  IX=IATTACH(6LTAPE15,DATIN)
           COLD  IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
           COLD  IF(IX.NE.0) STOP*INPUT FILE ATTACH NOT SATISFACTORY*
           COLD  WRITE(7,120) (DATIN(I),I=1,2)
           120   FORMAT(23H THIS READS INPUT FILE ,2A10/)
           122   CONTINUE
    *      NULL TRANSFER STATEMENT -- TRANSFER IGNORED
                 CALL INTIO(6HNSTART,NSTART,100,0)
                 NSTOP=NSTART
                 CALL INTIO(6H NSTOP,NSTOP,100,0)
                 CALL LOGIO(6HTETHRD,TETHRD,0)
                 IF(IRCVR.EQ.1) GO TO 125
                 CALL LOGIO(6HADAPTV,ADAPTV,0)
           125   CONTINUE
                 IF(TETHRD) ITETHR=2
                 IF(ADAPTV) GO TO 135
                 IRSIGN=-1
                 IADAP=3
           135   CONTINUE
                 DO 145 I=2,6
                 IASC=3*(I-1)+IDNRS(I)
                 IDENTS(I)=IDASCI(IASC)
           145   CONTINUE
                 ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
           150   FORMAT(A7,I1)
                 WRITE (7,152) IDNRS,IMOD
           152   FORMAT (1H ,6I1,1X,I1/)
                 WRITE (7,153) IDENTS
           153   FORMAT (1H ,A8)
           155   CONTINUE
                 READ(10,160) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
           160   FORMAT(I5,6G10.3)
                 IF(NRUN.LT.NSTART) GO TO 155
                 IF(NRUN.GE.NSTOP) MORE=.FALSE.
                 RETURN
           C
           C*********************************************************************
           C
           200   CONTINUE
           C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
           CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
                 WRITE(7,210) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
           210   FORMAT(5H1NRUN,12X,6HDSNRDB,16X,3HRHO,16X,4HBETA,17X,3HFSC,16X,
                *4HBMLS,15X,5HBRCVR/1H0,I4,6(10X,G10.3))
                 IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1H0,NRUN
           220   FORMAT(6I1,A1,I1,A1,I1)
                 IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
           230   FORMAT(6I1,A1,I1,I2)
                 IX=IREQST(6LTAPE17,3L*PF)
                 IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
                 IF(IX.NE.0) STOP*OUTPUT FILE REQUEST NOT SATISFACTORY*
                 WRITE(7,240) (DATOUT(I),I=1,2)
           240   FORMAT(25H0THIS WRITES OUTPUT FILE ,2A10)
```

```
      CALL DATE(TODAY)
      WRITE (7,250) FUNCTN (IAE),IAE
  250 FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
      WRITE (7,260) ((I,X(I)),I=1,8)
  260 FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
      WRITE(7,270) IDENTS(2)
  270 FORMAT(1H0,A7)
      RETURN
C
C*********************************************************************
C
  300 CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
      ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
      IF(IRSIGN.LT.0) IR=-IR
C
      RETURN
C
C*********************************************************************
C
  400 CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
      WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
  410 FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
     *             4H,NS=,I1,1H))
      WRITE(7,420) (IDENTS(I),I=5,6)
  420 FORMAT(1H0,A8/1H0,A7/)
      RETURN
C
C*********************************************************************
C
  500 CONTINUE
C THIS SETS-UP THE MSET-LOOP
      ICOUNT=0
      RETURN
C
C*********************************************************************
C
  600 CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
      RETURN
C
C*********************************************************************
C
  700 CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
      CALL INTIO(6H    KM,KM,1,1)
      WRITE(7,710) (TITLE1(I),I=1,3),(TITLE2(I,ITIT),I=1,2)
  710 FORMAT (4H0  K,6X,5HCSNRT,6X,5HCSNRF,5(5X,A6,4X)/)
      EMEAN=0.
      EMS=0.
      LASTCT=0
      RETURN
```

A-11

ORIGINAL PAGE IS OF POOR QUALITY

```
      C
      C*********************************************************************
      C
      800   CONTINUE
      C THIS INITIALIZES THE K-TH SCAN
            SEP=SPACE
            RETURN
      C
      C*********************************************************************
      C
      900   CONTINUE
      C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
            IF(ICOUNT.NE.LASTCT) SEP=ABORT
            THERR=X(2)-XS(2)
            COL8=THERR
            DTHE=X(2)-X(5)
            IF (IRCVR .NE. 1) COL8=SQRT(PPDIAG(2))
            IF (IRCVR .EQ. 1) CALL DFLTR1(COL8,THERR,FL10)
            WRITE(7,910) K,CSNRT,CSNRF,X(2),DTHE,XS(2),SEP,THERR,COL8
      910   FORMAT (1H ,I3,2(2X,F9.3),3(2X,G13.6),1X,A1,1X,G13.6,2X,G13.6)
            CSNRTO(K)=CSNRT
            CSNRFR(K)=CSNRF
            THESEP(K)=DTHE
            THESER(K)=THERR
            ABBORT(K)=SEP
            LASTCT=ICOUNT
            EMEAN=EMEAN+THERR
            EMS=EMS+THERR**2
            RETURN
      C
      C*********************************************************************
      C
      1000  CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
            EMEAN=EMEAN/KM
            EMS=EMS/KM
            ERMS=SQRT(EMS)
            ESTDEV=SQRT(EMS-EMEAN**2)
            XW=ICOUNT
            TCOUNT=100.*XW/KM
            WRITE(7,1010)
      1010  FORMAT (31HOTHETA-ERROR SAMPLE STATISTICS:)
            IF (IRCVR .EQ. 1) WRITE(7,1020)
      1020  FORMAT (17H (FILTERED ERROR))
            WRITE(7,1030) EMEAN,ERMS,ESTDEV
      1030  FORMAT (10HO EMEAN = ,G13.6/10H   ERMS = ,G13.6/
           *10H ESTDEV = ,G13.6)
            IF (IRCVR .EQ. 1) WRITE(7,1040) TCOUNT
      1040  FORMAT (//1HO,F7.2,22H% OF SCANS ARE ABORTED)
            YMIN=YMAX=0.
            WRITE (7,11)
            CALL PLOTR(THESEP,THESER,KM,XNAME,YNAME,YMIN,YMAX,0)
            WRITE (7,11)
            CALL REALID(6H  YMIN,YMIN,1)
            CALL REALID(6H  YMAX,YMAX,1)
            IF (IRCVR .NE. 1) WRITE(7,1060) (GQGT(I,I),I=1,NS)
      1060  FORMAT (10(/),35H ON LAST SCAN, DIAGONAL OF GQGT WAS/
```

A-12

```
         *1H ,G11.4,7(1H,,G11.4))
          WRITE(7,11)
          WRITE(7,1070) (DATOUT(I),I=1,2)
 1070     FORMAT(13H OUTPUT FILE ,2A10,1H:/)
          WRITE(7,1072) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
 1072     FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
          WRITE(7,1074) IDENTS
 1074     FORMAT(1H ,6(1X,A8,9X)/)
          WRITE(7,1076) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
 1076     FORMAT(1H ,5X,I2,6X,6(5X,G13.6)/)
          IKM=MINO(35,KM-9)
          DO 1077 I=1,IKM
 1077     WRITE(7,1078) CSNRTO(I),CSNRFR(I),THESER(I),ABBORT(I),THESEP(I),I
 1078     FORMAT(1H ,3(G13.6,5X),6X,A1,11X,G13.6,5X,I7)
          DO 1079 I=1,5
 1079     WRITE(7,1080)
 1080     FORMAT(1H,6(6X,1H.,11X))
          KM3=KM-3
          DO 1081 I=KM3,KM
 1081     WRITE(7,1078) CSNRTO(I),CSNRFR(I),THESER(I),ABBORT(I),THESEP(I),I
          WRITE(7,1082) EMEAN,ERMS,ESTDEV,TCOUNT,YMIN,YMAX
 1082     FORMAT(1H0,6(G13.6,5X)/)
          WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
          WRITE(17) IDENTS
          WRITE(17) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
          DO 1090 I=1,KM
          WRITE(17)     CSNRTO(I),CSNRFR(I),THESER(I),ABBORT(I),THESEP(I),I
 1090     CONTINUE
          WRITE(17) EMEAN,ERMS,ESTDEV,TCOUNT,YMIN,YMAX
          REWIND 17
          IX=IATTACH(6LTAPE20,DATOUT)
          CALL RETURN(6LTAPE20)
          IF(IX.NE.0) GO TO 1095
          IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
          GO TO 1096
 1095     IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1096     IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
          IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
          CALL RETURN(6LTAPE17)
          WRITE (7,1099) (DATOUT(I),I=1,2)
 1099     FORMAT(13HOOUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
          RETURN
C
C*********************************************************************
C
 1100     CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
CLGMAX SETS OF KM-SCANS
          RETURN
C
C*********************************************************************
C
 1200     CONTINUE
C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
          RETURN
C
C*********************************************************************
```

```
285          C
                   END
```

430008 CM STORAGE USED 4.098 SECONDS

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C RMSE VS.  BRATIO ( = BRCVR/BMLS).
C
      REAL LAMDA
      INTEGER XNAME1(2),YNAME1(2),YNAM1(2,9)
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER TITLE1(9),FUNCTN(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV,
     CFILEIN,FILOUT
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TORO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION X(8),THESER(115),XDATUM(115),EEME(13),EERM(13)
      DIMENSION IDNRS(6),IDASCI(20),IDENTS(6),EEST(13),TTCO(13),YYMI(13)
      DIMENSION YYMA(13)
      DIMENSION BRATIO(13)
      DIMENSION EEMEA(13,9),EERMS(13,9),EESTD(13,9)
      DIMENSION EMEAN(9),EMS(9),ERMS(9),ESD(9)
      EQUIVALENCE (EEME(1),EEMEA(1,2)),(EERM(1),EERMS(1,2)),
     *(EEST(1),EESTD(1,2))
      EQUIVALENCE(THERR,ES(2))
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA YNAM1/8HALF RMSE,8H        ,8HTHE RMSE,8HDEGREES ,
     *8HTDT RMSE,8HDEG/SEC ,8HALR RMSE,8H        ,8HTHR RMSE,
     *8HDEGREES ,8HTRD RMSE,8HDEG/SEC ,8HBET RMSE,8HDEGREES ,
     *8HFSC RMSE,8H   HZ   ,8HFSCORMSE,8H HZ/SEC /
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H200000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7H       ,
     *8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/11/,BRARAT/10./,FILEIN
     */.FALSE./,FILOUT/.TRUE./
      DATA TITLE1/6H ALFA ,6H THETA,6HTHEDOT,6H ALFAR,6HTHETAR,6HTHRDOT
     C,6H   B  ,6H  WSC ,6HWSCDOT/
      DATA XNAME/8H BETA   ,8HDEGREES /,YNAME/8HTHES ERR,8HDEGREES /
      DATA XNAME1/8H BRATIO ,8H        /
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
```

```
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
    1 FORMAT(1H )
   10 FORMAT(9(/))
   11 FORMAT (1H1)
C
C*****************************************************************
C
  100 CONTINUE
C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
CTHE FOLLOWING RECEIVER PARAMETERS
CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JH,BMLS
      IF(NFIRST) GO TO 155
      NDAC=.TRUE.
      NFIRST=.TRUE.
      JBNAM=JOBNAME(I)
      ISIM=3
      IMOD=1
      WRITE (7,110)
  110 FORMAT(38H1RMSE VS.BRATIO = BRCVR/BMLS SCENARIO/)
      CALL LOGIO(6HFILEIN,FILEIN,0)
      IF (.NOT.FILEIN) NSTART=0
      IF(.NOT.FILEIN) GO TO 122
      IX=IATTACH(6LTAPE15,DATIN)
      IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
      IF(IX.NE.0) STOP*INPUT FILE ATTACH NOT SATISFACTORY*
      WRITE(7,120) (DATIN(I),I=1,2)
  120 FORMAT(23H THIS READS INPUT FILE ,2A10/)
  122 CALL LOGIO(6HFILOUT,FILOUT,0)
      CALL INTIO(6HNSTART,NSTART,100,0)
      NSTOP=NSTART
      CALL INTIO(6H NSTOP,NSTOP,100,0)
      CALL LOGIO(6HTETHRD,TETHRD,0)
      IF(IRCVR.EQ.1) GO TO 125
      CALL LOGIO(6HADAPTV,ADAPTV,0)
  125 CONTINUE
      IF(TETHRD) ITETHR=2
      IF(ADAPTV) GO TO 135
      IRSIGN=-1
      IADAP=3
  135 CONTINUE
      DO 145 I=2,6
      IASC=3*(I-1)+IDNRS(I)+2
      IDENTS(I)=IDASCI(IASC)
  145 CONTINUE
      ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
  150 FORMAT(A7,I1)
      WRITE (7,152) IDNRS,IMOD
  152 FORMAT (1H ,6I1,1X,I1/)
      WRITE (7,153) IDENTS
  153 FORMAT (1H ,A8)
      CALL INTIO(6H LGMAX,LGMAX,13,0)
      CALL INTIO(6H    KM,KM,115,0)
      CALL REALIO(6HBRARAT,BRARAT,0)
      KSTART=MIN0(KM,KSTART)
      BRATIO(1)=1.
      BRATMX=SQRT(BRARAT)
```

```
          ROOT=1./(LGMAX-1)
          IF(LGMAX.LE.1) GO TO 156
          DO 154 I=1,LGMAX
  154     BRATIO(I)=BRATMX**(2.*ROOT*(I-1)-1.)
  156     CONTINUE
          KMNET=KM-KSTART+1
          FSCMIN=1./(DELTAT(IAE)*KMNET)
          NRUN=NSTART-1
          DSNRDB=20.
          RHO=0.5
          BETA=45.
          IFS=1
          TSEP=1.
  155     CONTINUE
          IF(FILEIN) GOTO 158
          NRUN=NRUN+1
          WRITE(7,11)
          CALL INTIO(6H  NRUN,NRUN,1,1)
          CALL REALIO(6HDSNRDB,DSNRDB,0)
          CALL REALIO(6H   RHO,RHO,0)
          CALL REALIO(6H  BETA,BETA,0)
          CALL INTIO(6H   IFS,IFS,100,0)
          CALL REALIO(6H  TSEP,TSEP,0)
          GOTO 167
  158     CONTINUE
          READ(15,160) NRUN,DSNRDB,RHO,BETA,IFS,TSEP
          IF(EOF(15))170,165
  160     FORMAT(I5,3G10.3,5X,I5,2G10.3)
  165     IF(NRUN.LT.NSTART) GO TO 158
  167     IF(NRUN.GE.NSTOP) MORE=.FALSE.
          FSC=IFS*FSCMIN
          XO(5,IAE)=XO(2,IAE)-TSEP
          XO(3,IAE)=0.
          XO(6,IAE)=0.
          RETURN
  170     STOP*EOF REACHED ON INPUT FILE*
C
C*******************************************************************
C
  200     CONTINUE
C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
          WRITE(7,210) NRUN,DSNRDB,RHO,BETA,IFS,TSEP
  210     FORMAT(5HONRUN,12X,6HDSNRDB,16X,3HRHO,16X,4HBETA,16X,4HIFSC,16X,
         *4HTSEP/1HO,I4,3(10X,G10.3),12X,I5,3X,
         *(10X,G10.3))
          IF(.NOT.FILOUT) GOTO 245
          IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1HO,NRUN
  220     FORMAT(6I1,A1,I1,A1,I1)
          IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
  230     FORMAT(6I1,A1,I1,I2)
          IX=IREQST(6LTAPE17,3L*PF)
          IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
          IF(IX.NE.0) STOP*OUTPUT FILE REQUEST NOT SATISFACTORY*
          WRITE(7,240) (DATOUT(I),I=1,2)
  240     FORMAT(25HOTHIS WRITES OUTPUT FILE ,2A10)
  245     CALL DATE(TODAY)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      WRITE (7,250) FUNCTN (IAE),IAE
250   FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
      WRITE (7,260) ((I,X(I)),I=1,8)
260   FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
      WRITE(7,270) IDENTS(2)
270   FORMAT(1H0,A7)
      RETURN
C
C*********************************************************************
C
300   CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRO
      ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
      IR=ISIGN(IR,IRSIGN)
C
      RETURN
C
C*********************************************************************
400   CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
      WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
410   FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
     *        4H,NS=,I1,1H))
      WRITE(7,420) (IDENTS(I),I=5,6)
420   FORMAT(1H0,A8/1H0,A7/)
      RETURN
C
C*********************************************************************
C
500   CONTINUE
C THIS SETS-UP THE MSET-LOOP
      ICOUNT=0
      DO 510 I=1,LGMAX
      YYMI(I)=1.E322
510   YYMA(I)=-1.E322
      RETURN
C
C*********************************************************************
C
600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
      RETURN
C
C*********************************************************************
C
700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
      BMLS=AMAX1(1.,1./BRATIO(LG))
      BBB=2.4/BMLS
      BRCVR=AMAX1(BRATIO(LG),1.)
      BB=2.4/BRCVR
      PDCRIT=PI*BB/8.
      CC=PI*PDCRIT
```

```
            X(7)=XO(7,IAE)
            DUM=GAUSS(1.0)
            WRITE(7,11)
            CALL INTIO(6H  NRUN,NRUN,1,1)
            WRITE(7,705) LG,BRATIO(LG),BHLS,BRCVR
      705   FORMAT(1H ,4X,3HLG=,I2,5X,9HBRATIO = ,G12.5/8H BHLS = ,G12.5,5X,
           *8HBRCVR = ,G12.5)
            CALL INTIO(6H    KM,KM,1,1)
            WRITE(7,710) (TITLE1(I),I=1,NS)
      710   FORMAT(6H0    K,4X,5HCSNRT,4X,5HQTY  ,9(3X,A6,2X))
            DO 712 I=1,NS
            EMEAN(I)=0.
      712   EMS(I)=0.
            LASTCT=0
            ICOUNT=0
            DO 730 IPA=1,NS
            DO 720 JPA=1,NS
            PA(IPA,JPA)=0.
      720   CONTINUE
      730   CONTINUE
            RETURN
      C
      C******************************************************************
      C
      800   CONTINUE
      C THIS INITIALIZES THE K-TH SCAN
            SEP=SPACE
            RETURN
      C
      C******************************************************************
      C
      900   CONTINUE
      C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
            IF(ICOUNT.NE.LASTCT) SEP=ABORT
            DO 905 I=1,NS
      905   ES(I)=X(I)-XS(I)
            IF(NS.GE.7) ES(7)=ABS(X(7))-ABS(XS(7))
            IF(NS.GE.8) ES(8)=ABS(X(8))-ABS(XS(8))
            BETTA=(180./PI)*X(7)
            COL8=THERR
            IF (IRCVR .NE. 1) COL8=SQRT(PPDIAG(2))
            IF (IRCVR .EQ. 1) CALL DFLTR1(COL8,THERR,FL10)
            IF((NRUN.EQ.NSTART.AND.LG.EQ.1).OR.(K.EQ.KM)) GOTO 906
            GO TO 950
      906   WRITE(7,910) K,CSNRT,(X(I),I=1,NS)
      910   FORMAT(3H0  ,I3,2X,G10.3,5H  X= ,2X,9(G10.3,1X))
            WRITE(7,920) (XS(I),I=1,NS)
      920   FORMAT(18X,5H XS= ,2X,9(G10.3,1X))
            WRITE(7,930) (ES(I),I=1,NS)
      930   FORMAT(18X,5H ES= ,2X,9(G10.3,1X))
            IF(IRCVR.EQ.1) WRITE(7,940) COL8
      940   FORMAT(8X,15HUNFILT. ES(2)= ,13X,G10.3)
      950   XDATUM(K)=BETTA
            THESER(K)=THERR
            LASTCT=ICOUNT
            IF(K.LT.KSTART) RETURN
            DO 960 I=1,NS
```

```
          EMEAN(I)=EMEAN(I)+ES(I)
      960 EMS(I)=EMS(I)+ES(I)**2
          RETURN
      C
      C*************************************************************************
      C
     1000 CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
          DO 1005 I=1,NS
          EMEAN(I)=EMEAN(I)/KMNET
          EMS(I)=EMS(I)/KMNET
          ERMS(I)=SQRT(EMS(I))
          ESD(I)=SQRT(EMS(I)-EMEAN(I)**2)
     1005 CONTINUE
          XW=ICOUNT
          TCOUNT=100.*XW/KM
          WRITE(7,1009)
     1009 FORMAT(3(/))
          WRITE(7,1010)KMNET
     1010 FORMAT(5X,26HERROR SAMPLE STATISTICS: (,I3,9H SAMPLES))
          IF(IRCVR.EQ.1) WRITE (7,1015)
     1015 FORMAT(5X,31HTHRESHOLD RCVR (FILTERED ERROR))
          WRITE(7,1020)(TITLE1(I),I=1,NS)
     1020 FORMAT(1H0,20X,4HQTY ,9(3X,A6,2X))
          WRITE(7,1030)(EMEAN(I),I=1,NS)
     1030 FORMAT(1H0,16X,8HEMEAN = ,9(G10.3,1X))
          WRITE(7,1040)(ERMS(I),I=1,NS)
     1040 FORMAT(18X,7HERMS = ,9(G10.3,1X))
          WRITE(7,1050)(ESD(I),I=1,NS)
     1050 FORMAT(19X,6HESD = ,9(G10.3,1X))
          IF(IRCVR.EQ.1)WRITE(7,1060)TCOUNT
     1060 FORMAT(//1H ,4X,F7.2,22H% OF SCANS ARE ABORTED)
          IF(IRCVR.NE.1)WRITE(7,1070)(GQGT(I,I),I=1,NS)
     1070 FORMAT(5(/),4X,35H ON LAST SCAN, DIAGONAL OF GQGT WAS/26X
         C9(G10.3,1X))
          DO 1071 I=1,NS
          EEMEA(LG,I)=EMEAN(I)
          EERMS(LG,I)=ERMS(I)
     1071 EESTD(LG,I)=ESD(I)
          DO 1072 I=1,KM
          YYMI(LG)=AMIN1(THESER(I),YYMI(LG))
     1072 YYMA(LG)=AMAX1(THESER(I),YYMA(LG))
          TTCO(LG)=TCOUNT
          IF(LG.NE.1) RETURN
          YMIN=YMAX=0.
          WRITE(7,11)
          CALL INTIO(6H  NRUN,NRUN,1,1)
          CALL PLOTR(XDATUM,THESER,KM,XNAME,YNAME,YMIN,YMAX,0)
          RETURN
      C
      C*************************************************************************
      C
     1100 CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
      CLGMAX SETS OF KM SCANS
          WRITE(7,11)
          CALL INTIO(6H  NRUN,NRUN,1,1)
```

```
      WRITE(7,1110) 6HBRATIO,(TITLE1(I),I=1,NS)
 1110 FORMAT(1H0,10X,3HQTY  ,9(3X,A6,2X))
      WRITE(7,1120)BRATIO(1),(EEMEA(1,I),I=1,NS)
 1120 FORMAT(1H0, 7X,7HEMEAN: ,9(G10.3,1X))
      DO 1130 J=2,LGMAX
 1130 WRITE(7,1161)BRATIO(J),(EEMEA(J,I),I=1,NS)
      WRITE(7,1140)BRATIO(1),(EERMS(1,I),I=1,NS)
 1140 FORMAT(1H0, 8X,6HERMS: ,9(G10.3,1X))
      DO 1150 J=2,LGMAX
 1150 WRITE(7,1161)BRATIO(J),(EERMS(J,I),I=1,NS)
      WRITE(7,1160)BRATIO(1),(EESTD(1,I),I=1,NS)
 1160 FORMAT(1H0, 8X,6HESTD: ,9(G10.3,1X))
      DO 1165 J=2,LGMAX
 1165 WRITE(7,1161)BRATIO(J),(EESTD(J,I),I=1,NS)
 1161 FORMAT(15X,9(G10.3,1X))
      IF (.NOT.FILOUT) GOTO 1199
      WRITE(7,11)
      WRITE(7,1170) (DATOUT(I),I=1,2)
 1170 FORMAT(13H OUTPUT FILE ,2A10,1H:/)
      WRITE(7,1172) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
 1172 FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,1174) IDENTS,KSTART
 1174 FORMAT(1H ,6(1X,A8,9X),6X,I3/)
      WRITE(7,1176) NRUN,DSNROB,RHO,BETA,FSC,TSEP
 1176 FORMAT(1H ,5X,I2,6X,5(5X,G13.6)/)
      DO 1175 LGN=1,LGMAX
 1175 WRITE(7,1182) EEME(LGN),EERM(LGN),EEST(LGN),TTCD(LGN),YYMI(LGN),
     *YYMA(LGN),BRATIO(LGN),LGN
 1182 FORMAT(1H ,7(G13.6,5X),I5)
      WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      WRITE(17) IDENTS,KSTART
      WRITE(17)NRUN,DSNROB,RHO,BETA,FSC,TSEP
      DO 1190 LGN=1,LGMAX
 1190 WRITE(17) EEME(LGN),EERM(LGN),EEST(LGN),TTCD(LGN),YYMI(LGN),
     *YYMA(LGN),BRATIO(LGN),LGN
      IX=IATTACH(6LTAPE20,DATOUT)
      IF(IX.NE.0) GO TO 1195
      READ(20) DUM,DUM,IDUM,IDUM,IDUM,DUM,IDUM
      CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
      GO TO 1196
 1195 CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1196 IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
      CALL RETURN(6LTAPE17)
      WRITE (7,1198) (DATOUT(I),I=1,2)
 1198 FORMAT(13H0OUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
 1199 CONTINUE
      DO 1103 I=1,NS
      FACTOR=1.
      IF(I.EQ.7) FACTOR=180./PI
      IF(I.EQ.8) FACTOR=.5/PI
      IF(I.EQ.9) FACTOR=.5/PI
      YMIN=0.
```

```
            YMAX=0.
            DO 1101 LG1=1,LGMAX
      1101  EEME(LG1)=EERMS(LG1,I)*FACTOR
            YNAME1(1)=YNAM1(1,I)
            YNAME1(2)=YNAM1(2,I)
            K1=MOD(I,2)
            IF(K1.EQ.1)WRITE(7,11)
            IF(K1.EQ.1) CALL INTIO(6H  NRUN,NRUN,1,1)
               IF(K1.EQ.0) WRITE(7,10)
            CALL PLOTR(BRATIO,EEME,LGMAX,XNAME1,YNAME1,YMIN,YMAX,0)
      1103  CONTINUE
            RETURN
      C
      C*********************************************************************
      C
      1200  CONTINUE
      C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
            RETURN
      C
      C*********************************************************************
      C
            END
```

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C RMSE VS.  FSC  STUDY
C
      REAL LAMDA
      INTEGER XNAME1(2),YNAME1(2),YNAM1(2,9)
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER TITLE1(9),FUNCTN(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV,
     CFILEIN,FILOUT
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GOGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNROB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION X(8),THESER(115),XDATUM(115),EEME(13),EERM(13)
      DIMENSION IDNRS(6),IDASCI(20),IDENTS(6),EEST(13),TTCO(13),YYHI(13)
      DIMENSION YYMA(13)
      DIMENSION FSCA(13),IFSCA(13)
      DIMENSION EEMEA(13,9),EERMS(13,9),EESTD(13,9)
      DIMENSION EMEAN(9),EMS(9),ERMS(9),ESD(9)
      EQUIVALENCE (EEME(1),EEMEA(1,2)),(EERM(1),EERMS(1,2)),
     *(EEST(1),EESTD(1,2))
      EQUIVALENCE(THERR,ES(2))
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA YNAM1/8HALF RMSE,8H        ,8HTHE RMSE,8HDEGREES ,
     *8HTDT RMSE,8HDEG/SEC ,8HALR RMSE,8H        ,8HTHR RMSE,
     *8HDEGREES ,8HTRD RMSE,8HDEG/SEC ,8HBET RMSE,8HDEGREES ,
     *8HFSC RMSE,8H   HZ   ,8HFSCDRMSE,8H HZ/SEC /
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H200000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7H       ,
     *8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/11/,FILEIN
     */.FALSE./,FILOUT/.TRUE./
      DATA TITLE1/6H ALFA ,6H THETA,6HTHEDOT,6H ALFAR,6HTHETAR,6HTHRDOT
     C,6H   B  ,6H  WSC ,6HWSCDOT/
      DATA XNAME/8HSCAN NO.,8H    K   ../,YNAME/8HTHES ERR,8HDEGREES /
      DATA XNAME1/8H   FSC  ,8H   HZ   /
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
```

```
      C
            GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
      1     FORMAT(1H )
      10    FORMAT(9(/))
      11    FORMAT (1H1)
      C
      C*********************************************************************
      C
      100   CONTINUE
      C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
      CTHE FOLLOWING RECEIVER PARAMETERS
      CSIGMA,IAE,DSNROB,RHO,FSC,BETA,JM,BMLS
            IF(NFIRST) GO TO 155
            NOAC=.TRUE.
            NFIRST=.TRUE.
            JBNAM=JOBNAME(I)
            ISIM=4
            IMOD=1
            WRITE (7,110)
      110   FORMAT(19H1RMSE VS. FSC STUDY)
            CALL LOGIO(6HFILEIN,FILEIN,0)
            IF (.NOT.FILEIN) NSTART=0
            IF(.NOT.FILEIN) GO TO 122
            IX=IATTACH(6LTAPE15,DATIN)
            IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
            IF(IX.NE.0) STOP≠INPUT FILE ATTACH NOT SATISFACTORY≠
            WRITE(7,120) (DATIN(I),I=1,2)
      120   FORMAT(23H THIS READS INPUT FILE ,2A10/)
      122   CALL LOGIO(6HFILOUT,FILOUT,0)
            CALL INTIO(6HNSTART,NSTART,100,0)
            NSTOP=NSTART
            CALL INTIO(6H NSTOP,NSTOP,100,0)
            CALL LOGIO(6HTETHRD,TETHRD,0)
            IF(IRCVR.EQ.1) GO TO 125
            CALL LOGIO(6HADAPTV,ADAPTV,0)
      125   CONTINUE
            IF(TETHRD) ITETHR=2
            IF(ADAPTV) GO TO 135
            IRSIGN=-1
            IADAP=3
      135   CONTINUE
            DO 145 I=2,6
            IASC=3*(I-1)+IDNRS(I)+2
            IDENTS(I)=IDASCI(IASC)
      145   CONTINUE
            ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
      150   FORMAT(A7,I1)
            WRITE (7,152) IDNRS,IMOD
      152   FORMAT (1H ,6I1,1X,I1/)
            WRITE (7,153) IDENTS
      153   FORMAT (1H ,A8)
            CALL INTIO(6H    KM,KM,115,0)
            KSTART=MINO(KM,KSTART)
            KMNET=KM-KSTART+1
            FSCMIN=1./(DELTAT(IAE)*KMNET)
            READ(10,105) LGMAX,(IFSCA(I),I=1,LGMAX)
      105   FORMAT(14I5)
```

```
            WRITE(7,106) LGMAX,(IFSCA(I),I=1,LGMAX)
      106   FORMAT(9HOLGMAX = ,I2/1H ,13(I7,2X))
            DO 154 I=1,LGMAX
      154   FSCA(I)=FSCMIN*IFSCA(I)
            WRITE(7,107)(FSCA(I),I=1,LGMAX)
      107   FORMAT(1H 13(F7.3,2X))
            NRUN=NSTART-1
            DSNRDB=20.
            RHO=0.5
            BETA=45.
            BMLS=1.
            BRCVR=1.
            TSEP=1.
      155   CONTINUE
            IF(FILEIN) GOTO 158
            NRUN=NRUN+1
            WRITE(7,11)
            CALL INTIO(6H  NRUN,NRUN,1,1)
            CALL REALIO(6HDSNRDB,DSNRDB,0)
            CALL REALIO(6H   RHO,RHO,0)
            CALL REALIO(6H  BETA,BETA,0)
            CALL REALIO (6H  BMLS,BMLS,0)
            CALL REALIO (6H BRCVR,BRCVR,0)
            CALL REALIO(6H  TSEP,TSEP,0)
            GOTO 167
      158   CONTINUE
            READ(15,160) NRUN,DSNRDB,RHO,BETA,BMLS,BRCVR,TSEP
            IF(EOF(15))170,165
      160   FORMAT(I5,3G10.3,5X,3G10.3)
      165   IF(NRUN.LT.NSTART) GO TO 158
      167   IF(NRUN.GE.NSTOP) MORE=.FALSE.
            FSC=FSCA(1)
            XO(5,IAE)=XO(2,IAE)-TSEP
            XO(3,IAE)=0.
            XO(6,IAE)=0.
            RETURN
      170   STOP≠EOF REACHED ON INPUT FILE≠
      C
      C*************************************************************************
      C
      200   CONTINUE
      C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
      CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
            WRITE(7,210) NRUN,DSNRDB,RHO,BETA,BMLS,BRCVR,TSEP
      210   FORMAT(5HONRUN,5X,6HDSNRDB,12X,3HRHO,12X,4HBETA,12X,4HBMLS,11X,
           *5HBRCVR,12X,4HTSEP/1HO,I4,6(6X,G10.3))
            IF(.NOT.FILOUT) GOTO 245
            IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1HO,NRUN
      220   FORMAT(6I1,A1,I1,A1,I1)
            IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
      230   FORMAT(6I1,A1,I1,I2)
            IX=IREQST(6LTAPE17,3L*PF)
            IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
            IF(IX.NE.0) STOP≠OUTPUT FILE REQUEST NOT SATISFACTORY≠
            WRITE(7,240) (DATOUT(I),I=1,2)
      240   FORMAT(25HOTHIS WRITES OUTPUT FILE ,2A10)
      245   CALL DATE(TODAY)
```

```
      WRITE (7,250) FUNCTN (IAE),IAE
250   FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
      WRITE (7,260) ((I,X(I)),I=1,8)
260   FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
      WRITE(7,270) IDENTS(2)
270   FORMAT(1H0,A7)
      RETURN
C
C*****************************************************************
C
300   CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
      ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
      IR=ISIGN(IR,IRSIGN)
C
      RETURN
C
C*****************************************************************
400   CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
      WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
410   FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
     *            4H,NS=,I1,1H))
      WRITE(7,420) (IDENTS(I),I=5,6)
420   FORMAT(1H0,A8/1H0,A7/)
      RETURN
C
C*****************************************************************
C
500   CONTINUE
C THIS SETS-UP THE MSET-LOOP
      ICOUNT=0
      DO 510 I=1,LGMAX
      YYMI(I)=1.E322
510   YYMA(I)=-1.E322
      RETURN
C
C*****************************************************************
C
600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
      RETURN
C
C*****************************************************************
C
700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
      FSC=FSCA(LG)
      XO(8,IAE)=2.*PI*FSC
      X(8)=XO(8,IAE)
      X(7)=XO(7,IAE)
      DUM=GAUSS(1.0)
      WRITE(7,11)
```

```
      CALL INTIO(6H  NRUN,NRUN,1,1)
      WRITE(7,705) LG,FSC
705   FORMAT(1H ,4X,3HLG=,I2,5X,6HFSC = ,G12.5)
      CALL INTIO(6H    KM,KM,1,1)
      WRITE(7,710) (TITLE1(I),I=1,NS)
710   FORMAT(6H0    K,4X,5HCSNRT,4X,5HQTY   ,9(3X,A6,2X))
      DO 712 I=1,NS
      EMEAN(I)=0.
712   EMS(I)=0.
      LASTCT=0
      ICOUNT=0
      DO 730 IPA=1,NS
      DO 720 JPA=1,NS
      PA(IPA,JPA)=0.
720   CONTINUE
730   CONTINUE
      RETURN
C
C**********************************************************************
C
800   CONTINUE
C THIS INITIALIZES THE K-TH SCAN
      SEP=SPACE
      RETURN
C
C**********************************************************************
C
900   CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
      IF(ICOUNT.NE.LASTCT) SEP=ABORT
      DO 905 I=1,NS
905   ES(I)=X(I)-XS(I)
      IF(NS.GE.7) ES(7)=ABS(X(7))-ABS(XS(7))
      IF(NS.GE.8) ES(8)=ABS(X(8))-ABS(XS(8))
      BETTA=(180./PI)*X(7)
      COL8=THERR
      IF (IRCVR .NE. 1) COL8=SQRT(PPDIAG(2))
      IF (IRCVR .EQ. 1) CALL DFLTR1(COL8,THERR,FL10)
      IF((NRUN.EQ.NSTART.AND.LG.EQ.1).OR.(K.EQ.KM)) GOTO 906
      GO TO 950
906   WRITE(7,910) K,CSNRT,(X(I),I=1,NS)
910   FORMAT(3H0  ,I3,2X,G10.3,5H  X= ,2X,9(G10.3,1X))
      WRITE(7,920) (XS(I),I=1,NS)
920   FORMAT(18X,5H XS= ,2X,9(G10.3,1X))
      WRITE(7,930) (ES(I),I=1,NS)
930   FORMAT(18X,5H ES= ,2X,9(G10.3,1X))
      IF(IRCVR.EQ.1) WRITE(7,940) COL8
940   FORMAT(8X,15HUNFILT. ES(2)= ,13X,G10.3)
950   XDATUM(K)=K
      THESER(K)=THERR
      LASTCT=ICOUNT
      IF(K.LT.KSTART) RETURN
      DO 960 I=1,NS
      EMEAN(I)=EMEAN(I)+ES(I)
960   EMS(I)=EMS(I)+ES(I)**2
      RETURN
C
```

```
C**********************************************************************
C
1000  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
      DO 1005 I=1,NS
      EMEAN(I)=EMEAN(I)/KMNET
      EMS(I)=EMS(I)/KMNET
      ERMS(I)=SQRT(EMS(I))
      ESD(I)=SQRT(EMS(I)-EMEAN(I)**2)
1005  CONTINUE
      XW=ICOUNT
      TCOUNT=100.*XW/KM
      WRITE(7,1009)
1009  FORMAT(3(/))
      WRITE(7,1010)KMNET
1010  FORMAT(5X,26HERROR SAMPLE STATISTICS: (,I3,9H SAMPLES))
      IF(IRCVR.EQ.1) WRITE (7,1015)
1015  FORMAT(5X,31HTHRESHOLD RCVR (FILTERED ERROR))
      WRITE(7,1020)(TITLE1(I),I=1,NS)
1020  FORMAT(1H0,20X,4HQTY ,9(3X,A6,2X))
      WRITE(7,1030)(EMEAN(I),I=1,NS)
1030  FORMAT(1H0,16X,8HEMEAN = ,9(G10.3,1X))
      WRITE(7,1040)(ERMS(I),I=1,NS)
1040  FORMAT(18X,7HERMS = ,9(G10.3,1X))
      WRITE(7,1050)(ESD(I),I=1,NS)
1050  FORMAT(19X,6HESD = ,9(G10.3,1X))
      IF(IRCVR.EQ.1)WRITE(7,1060)TCOUNT
1060  FORMAT(//1H ,4X,F7.2,22H% OF SCANS ARE ABORTED)
      IF(IRCVR.NE.1)WRITE(7,1070)(GQGT(I,I),I=1,NS)
1070  FORMAT(5(/),4X,35H ON LAST SCAN, DIAGONAL OF GQGT WAS/26X
     C9(G10.3,1X))
      DO 1071 I=1,NS
      EEMEA(LG,I)=EMEAN(I)
      EERMS(LG,I)=ERMS(I)
1071  EESTD(LG,I)=ESD(I)
      DO 1072 I=1,KM
      YYMI(LG)=AMIN1(THESER(I),YYMI(LG))
1072  YYMA(LG)=AMAX1(THESER(I),YYMA(LG))
      TTCO(LG)=TCOUNT
      IF(LG.NE.1) RETURN
      YMIN=YMAX=0.
      WRITE(7,11)
      CALL INTIO(6H  NRUN,NRUN,1,1)
      CALL PLOTR(XDATUM,THESER,KM,XNAME,YNAME,YMIN,YMAX,0)
      RETURN
C
C**********************************************************************
C
1100  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
CLGMAX SETS OF KM SCANS
      WRITE(7,11)
      CALL INTIO(6H  NRUN,NRUN,1,1)
      WRITE(7,1110) 6H  FSC ,(TITLE1(I),I=1,NS)
1110  FORMAT(1H0,10X,3HQTY  ,9(3X,A6,2X))
      WRITE(7,1120)FSCA(1),(EEMEA(1,I),I=1,NS)
1120  FORMAT(1H0, 7X,7HEMEAN: ,9(G10.3,1X))
```

```
      DO 1130 J=2,LGMAX
 1130 WRITE(7,1161) FSCA(J),(EEMEA(J,I),I=1,NS)
      WRITE(7,1140) FSCA(1),(EERMS(1,I),I=1,NS)
 1140 FORMAT(1H0, 8X,6HERMS: ,9(G10.3,1X))
      DO 1150 J=2,LGMAX
 1150 WRITE(7,1161) FSCA(J),(EERMS(J,I),I=1,NS)
      WRITE(7,1160) FSCA(1),(EESTD(1,I),I=1,NS)
 1160 FORMAT(1H0, 8X,6HESTD: ,9(G10.3,1X))
      DO 1165 J=2,LGMAX
 1165 WRITE(7,1161) FSCA(J),(EESTD(J,I),I=1,NS)
 1161 FORMAT(15X,9(G10.3,1X))
      DO 1103 I=1,NS
      FACTOR=1.
      IF(I.EQ.7) FACTOR=180./PI
      IF(I.EQ.8) FACTOR=.5/PI
      IF(I.EQ.9) FACTOR=.5/PI
      YMIN=0.
      YMAX=0.
      DO 1101 LG1=1,LGMAX
 1101 EEME(LG1)=EERMS(LG1,I)*FACTOR
      YNAME1(1)=YNAM1(1,I)
      YNAME1(2)=YNAM1(2,I)
      K1=MOD(I,2)
      IF(K1.EQ.1)WRITE(7,11)
      IF(K1.EQ.1) CALL INTIO(6H  NRUN,NRUN,1,1)
         IF(K1.EQ.0) WRITE(7,10)
      CALL PLOTR(FSCA,EEME,LGMAX,XNAME1,YNAME1,YMIN,YMAX,0)
 1103 CONTINUE
      IF(.NOT.FILOUT) RETURN
      WRITE(7,11)
      WRITE(7,1170) (DATOUT(I),I=1,2)
 1170 FORMAT(13H OUTPUT FILE ,2A10,1H:/)
      WRITE(7,1172) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
 1172 FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,1174) IDENTS,KSTART
 1174 FORMAT(1H ,6(1X,A8,9X),6X,I3/)
      WRITE(7,1176) NRUN,DSNRDB,RHO,BETA,FSC,TSEP
 1176 FORMAT(1H ,5X,I2,6X,6(5X,G13.6)/)
      DO 1175 LGN=1,LGMAX
 1175 WRITE(7,1182) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
     *YYMA(LGN),FSCA(LGN),LGN
 1182 FORMAT(1H ,7(G13.6,5X),I5)
      WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      WRITE(17) IDENTS,KSTART
      WRITE(17)NRUN,DSNRDB,RHO,BETA,BHLS,BRCVR,TSEP
      DO 1190 LGN=1,LGMAX
 1190 WRITE(17) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
     *YYMA(LGN),FSCA(LGN),LGN
      IX=IATTACH(6LTAPE20,DATOUT)
      IF(IX.NE.0) GO TO 1195
      READ(20) DUM,DUM,IDUM,IDUM,IDUM,DUM,IDUM
      CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
      GO TO 1196
 1195 CALL RETURN(6LTAPE20)
      REWIND 17
```

```
      IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1196 IF(IX.NE.0) CALL INTID(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
      CALL RETURN(6LTAPE17)
      WRITE (7,1198) (DATOUT(I),I=1,2)
 1198 FORMAT(13H0OUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
      RETURN
C
C************************************************************************
C
 1200 CONTINUE
C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
      WRITE(7,11)
      RETURN
C
C************************************************************************
C
      END
```

45000B CM STORAGE USED 13.054 SECONDS

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C RMSE VS.  FSC  STUDY
C
      REAL LAMDA
      INTEGER XNAME1(2),YNAME1(2),YNAM1(2,9)
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER TITLE1(9),FUNCTN(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV,
     CFILEIN,FILOUT
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRO,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNROB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION X(8),THESER(115),XDATUM(115),EEME(13),EERM(13)
      DIMENSION IDNRS(6),IDASCI(20),IDENTS(6),EEST(13),TTCO(13),YYMI(13)
      DIMENSION YYMA(13)
      DIMENSION FSCA(13),IFSCA(13)
      DIMENSION EEMEA(13,9),EERMS(13,9),EESTD(13,9)
      DIMENSION EMEAN(9),EMS(9),ERMS(9),ESD(9)
      EQUIVALENCE (EEME(1),EEMEA(1,2)),(EERM(1),EERMS(1,2)),
     *(EEST(1),EESTD(1,2))
      EQUIVALENCE(THERR,ES(2))
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA YNAM1/8HALF RMSE,8H        ,8HTHE RMSE,8HDEGREES ,
     *8HTOT RMSE,8HDEG/SEC ,8HALR RMSE,8H        ,8HTHR RMSE,
     *8HDEGREES ,8HTRD RMSE,8HDEG/SEC ,8HBET RMSE,8HDEGREES ,
     *8HFSC RMSE,8H   HZ   ,8HFSCDRMSE,8H HZ/SEC /
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H20000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7H       ,
     *8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/11/,FILEIN
     */.FALSE./,FILOUT/.TRUE./
      DATA TITLE1/6H ALFA ,6H THETA,6HTHEDOT,6H ALFAR,6HTHETAR,6HTHRDOT
     C,6H   B  ,6H  WSC ,6HWSCDOT/
      DATA XNAME/8HSCAN NO.,8H   K    /,YNAME/8HTHES ERR,8HDEGREES /
      DATA XNAME1/8H  FSC   ,8H   HZ   /
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
```

```
      C
            GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
      1     FORMAT(1H )
      10    FORMAT(9(/))
      11    FORMAT (1H1)
      C
      C*******************************************************************
      C
      100   CONTINUE
      C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
      CTHE FOLLOWING RECEIVER PARAMETERS
      CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JM,BMLS
            IF(NFIRST) GO TO 155
            NOAC=.TRUE.
            NFIRST=.TRUE.
            JBNAM=JOBNAME(I)
            ISIM=4
            IMOD=2
            WRITE (7,110)
      110   FORMAT(19H1RMSE VS. FSC STUDY)
            CALL LOGIO(6HFILEIN,FILEIN,0)
            IF (.NOT.FILEIN) NSTART=0
            IF(.NOT.FILEIN) GO TO 122
            IX=IATTACH(6LTAPE15,DATIN)
            IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
            IF(IX.NE.0) STOP≠INPUT FILE ATTACH NOT SATISFACTORY≠
            WRITE(7,120) (DATIN(I),I=1,2)
      120   FORMAT(23H THIS READS INPUT FILE ,2A10/)
      122   CALL LOGIO(6HFILOUT,FILOUT,0)
            CALL INTIO(6HNSTART,NSTART,100,0)
            NSTOP=NSTART
            CALL INTIO(6H NSTOP,NSTOP,100,0)
            CALL LOGIO(6HTETHRD,TETHRD,0)
            IF(IRCVR.EQ.1) GO TO 125
            CALL LOGIO(6HADAPTV,ADAPTV,0)
      125   CONTINUE
            IF(TETHRD) ITETHR=2
            IF(ADAPTV) GO TO 135
            IRSIGN=-1
            IADAP=3
      135   CONTINUE
            DO 145 I=2,6
            IASC=3*(I-1)+IDNRS(I)+2
            IDENTS(I)=IDASCI(IASC)
      145   CONTINUE
            ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
      150   FORMAT(A7,I1)
            WRITE (7,152) IDNRS,IMOD
      152   FORMAT (1H ,6I1,1X,I1/)
            WRITE (7,153) IDENTS
      153   FORMAT (1H ,A8)
            CALL INTIO(6H    KM,KM,115,0)
            KSTART=MIN0(KM,KSTART)
            KMNET=KM-KSTART+1
            FSCMIN=1./(DELTAT(IAE)*KMNET)
            READ(10,105) LGMAX,(IFSCA(I),I=1,LGMAX)
      105   FORMAT(14I5)
```

```
      WRITE(7,106) LGMAX,(IFSCA(I),I=1,LGMAX)
106   FORMAT(9HOLGMAX = ,I2/1H ,13(I7,2X))
      DO 154 I=1,LGMAX
154   FSCA(I)=FSCMIN*IFSCA(I)
      WRITE(7,107)(FSCA(I),I=1,LGMAX)
107   FORMAT(1H 13(F7.3,2X))
      NRUN=NSTART-1
      DSNRDB=20.
      RHO=0.5
      BETA=45.
      BMLS=1.
      BRCVR=1.
      TSEP=1.
155   CONTINUE
      IF(FILEIN) GOTO 158
      NRUN=NRUN+1
      WRITE(7,11)
      CALL INTIO(6H  NRUN,NRUN,1,1)
      CALL REALIO(6HDSNRDB,DSNRDB,0)
      CALL REALIO(6H   RHO,RHO,0)
      CALL REALIO(6H  BETA,BETA,0)
      CALL REALIO (6H  BMLS,BMLS,0)
      CALL REALIO (6H BRCVR,BRCVR,0)
      CALL REALIO(6H  TSEP,TSEP,0)
      GOTO 167
158   CONTINUE
      READ(15,160) NRUN,DSNRDB,RHO,BETA,BMLS,BRCVR,TSEP
      IF(EOF(15))170,165
160   FORMAT(I5,3G10.3,5X,3G10.3)
165   IF(NRUN.LT.NSTART) GO TO 158
167   IF(NRUN.GE.NSTOP) MORE=.FALSE.
      FSC=FSCA(1)
      XO(5,IAE)=XO(2,IAE)-TSEP
      XO(3,IAE)=0.
      XO(6,IAE)=0.
      RETURN
170   STOP*EOF REACHED ON INPUT FILE*
C
C*********************************************************************
C
200   CONTINUE
C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
      WRITE(7,210) NRUN,DSNRDB,RHO,BETA,BMLS,BRCVR,TSEP
210   FORMAT(5HONRUN,5X,6HDSNRDB,12X,3HRHO,12X,4HBETA,12X,4HBMLS,11X,
     *5HBRCVR,12X,4HTSEP/1H0,I4,6(6X,G10.3))
      IF(.NOT.FILOUT) GOTO 245
      IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1H0,NRUN
220   FORMAT(6I1,A1,I1,A1,I1)
      IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
230   FORMAT(6I1,A1,I1,I2)
      IX=IREQST(6LTAPE17,3L*PF)
      IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE REQUEST NOT SATISFACTORY*
      WRITE(7,240) (DATOUT(I),I=1,2)
240   FORMAT(25HOTHIS WRITES OUTPUT FILE ,2A10)
245   CALL DATE(TODAY)
```

```
      WRITE (7,250) FUNCTN (IAE),IAE
 250  FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
      WRITE (7,260) ((I,X(I)),I=1,8)
 260  FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
      WRITE(7,270) IDENTS(2)
 270  FORMAT(1H0,A7)
      RETURN
C
C**********************************************************************
C
 300  CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
       ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
       IR=ISIGN(IR,IRSIGN)
C
       RETURN
C
C**********************************************************************
 400  CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
       WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
 410   FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
      *        4H,NS=,I1,1H))
       WRITE(7,420) (IDENTS(I),I=5,6)
 420   FORMAT(1H0,A8/1H0,A7/)
       RETURN
C
C**********************************************************************
C
 500   CONTINUE
C THIS SETS-UP THE MSET-LOOP
       ICOUNT=0
       DO 510 I=1,LGMAX
       YYMI(I)=1.E322
 510   YYMA(I)=-1.E322
       RETURN
C
C**********************************************************************
C
 600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
       RETURN
C
C**********************************************************************
C
 700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
       FSC=FSCA(LG)
       XO(8,IAE)=2.*PI*FSC
       X(8)=XO(8,IAE)
       X(7)=XO(7,IAE)
       DUM=GAUSS(1.0)
       WRITE(7,11)
```

```
      CALL INTIO(6H  NRUN,NRUN,1,1)
      WRITE(7,705) LG,FSC
  705 FORMAT(1H ,4X,3HLG=,I2,5X,6HFSC = ,G12.5)
      CALL INTIO(6H    KM,KM,1,1)
      WRITE(7,710) (TITLE1(I),I=1,NS)
  710 FORMAT(6H0    K,4X,5HCSNRT,4X,5HQTY  ,9(3X,A6,2X))
      DO 712 I=1,NS
      EMEAN(I)=0.
  712 EMS(I)=0.
      LASTCT=0
      ICOUNT=0
      DO 730 IPA=1,NS
      DO 720 JPA=1,NS
      PA(IPA,JPA)=0.
  720 CONTINUE
  730 CONTINUE
      RETURN
C
C***********************************************************************
C
  800 CONTINUE
C THIS INITIALIZES THE K-TH SCAN
      SEP=SPACE
      RETURN
C
C***********************************************************************
C
  900 CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
      IF(ICOUNT.NE.LASTCT) SEP=ABORT
      XS(7)=0.5*PI
      XS(8)=0.0
      DO 905 I=1,NS
  905 ES(I)=X(I)-XS(I)
      IF(NS.GE.7) ES(7)=ABS(X(7))-ABS(XS(7))
      IF(NS.GE.8) ES(8)=ABS(X(8))-ABS(XS(8))
      BETTA=(180./PI)*X(7)
      COL8=THERR
      IF (IRCVR .NE. 1) COL8=SQRT(PPDIAG(2))
      IF (IRCVR .EQ. 1) CALL DFLTR1(COL8,THERR,FL10)
      IF((NRUN.EQ.NSTART.AND.LG.EQ.1).OR.(K.EQ.KM)) GOTO 906
      GO TO 950
  906 WRITE(7,910) K,CSNRT,(X(I),I=1,NS)
  910 FORMAT(3H0  ,I3,2X,G10.3,5H  X= ,2X,9(G10.3,1X))
      WRITE(7,920) (XS(I),I=1,NS)
  920 FORMAT(18X,5H XS= ,2X,9(G10.3,1X))
      WRITE(7,930) (ES(I),I=1,NS)
  930 FORMAT(18X,5H ES= ,2X,9(G10.3,1X))
      IF(IRCVR.EQ.1) WRITE(7,940) COL8
  940 FORMAT(8X,15HUNFILT. ES(2)= ,13X,G10.3)
  950 XDATUM(K)=K
      THESER(K)=THERR
      LASTCT=ICOUNT
      IF(K.LT.KSTART) RETURN
      DO 960 I=1,NS
      EMEAN(I)=EMEAN(I)+ES(I)
  960 EMS(I)=EMS(I)+ES(I)**2
```

```
      RETURN
C
C*********************************************************************
C
1000  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
      DO 1005 I=1,NS
      EMEAN(I)=EMEAN(I)/KMNET
      EMS(I)=EMS(I)/KMNET
      ERMS(I)=SQRT(EMS(I))
      ESD(I)=SQRT(EMS(I)-EMEAN(I)**2)
1005  CONTINUE
      XW=ICOUNT
      TCOUNT=100.*XW/KM
      WRITE(7,1009)
1009  FORMAT(3(/))
      WRITE(7,1010)KMNET
1010  FORMAT(5X,26HERROR SAMPLE STATISTICS: (,I3,9H SAMPLES))
      IF(IRCVR.EQ.1) WRITE (7,1015)
1015  FORMAT(5X,31HTHRESHOLD RCVR (FILTERED ERROR))
      WRITE(7,1020)(TITLE1(I),I=1,NS)
1020  FORMAT(1H0,20X,4HQTY ,9(3X,A6,2X))
      WRITE(7,1030)(EMEAN(I),I=1,NS)
1030  FORMAT(1H0,16X,8HEMEAN = ,9(G10.3,1X))
      WRITE(7,1040)(ERMS(I),I=1,NS)
1040  FORMAT(18X,7HERMS = ,9(G10.3,1X))
      WRITE(7,1050)(ESD(I),I=1,NS)
1050  FORMAT(19X,6HESD = ,9(G10.3,1X))
      IF(IRCVR.EQ.1)WRITE(7,1060)TCOUNT
1060  FORMAT(//1H ,4X,F7.2,22H% OF SCANS ARE ABORTED)
      IF(IRCVR.NE.1)WRITE(7,1070)(GQGT(I,I),I=1,NS)
1070  FORMAT(5(/),4X,35H ON LAST SCAN, DIAGONAL OF GQGT WAS/26X
     C9(G10.3,1X))
      DO 1071 I=1,NS
      EEMEA(LG,I)=EMEAN(I)
      EERMS(LG,I)=ERMS(I)
1071  EESTD(LG,I)=ESD(I)
      DO 1072 I=1,KM
      YYMI(LG)=AMIN1(THESER(I),YYMI(LG))
1072  YYMA(LG)=AMAX1(THESER(I),YYMA(LG))
      TTCO(LG)=TCOUNT
      IF(LG.NE.1) RETURN
      YMIN=YMAX=0.
      WRITE(7,11)
      CALL INTIO(6H  NRUN,NRUN,1,1)
      CALL PLOTR(XDATUM,THESER,KM,XNAME,YNAME,YMIN,YMAX,0)
      RETURN
C
C*********************************************************************
C
1100  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
CLGMAX SETS OF KM SCANS
      WRITE(7,11)
      CALL INTIO(6H  NRUN,NRUN,1,1)
      WRITE(7,1110) 6H  FSC ,(TITLE1(I),I=1,NS)
1110  FORMAT(1H0,10X,3HQTY  ,9(3X,A6,2X))
```

```
              WRITE(7,1120)FSCA(1),(EEMEA(1,I),I=1,NS)
       1120   FORMAT(1H0, 7X,7HEMEAN: ,9(G10.3,1X))
              DO 1130 J=2,LGMAX
       1130   WRITE(7,1161) FSCA(J),(EEMEA(J,I),I=1,NS)
              WRITE(7,1140) FSCA(1),(EERMS(1,I),I=1,NS)
       1140   FORMAT(1H0, 8X,6HERMS: ,9(G10.3,1X))
              DO 1150 J=2,LGMAX
       1150   WRITE(7,1161) FSCA(J),(EERMS(J,I),I=1,NS)
              WRITE(7,1160) FSCA(1),(EESTD(1,I),I=1,NS)
       1160   FORMAT(1H0, 8X,6HESTD: ,9(G10.3,1X))
              DO 1165 J=2,LGMAX
       1165   WRITE(7,1161) FSCA(J),(EESTD(J,I),I=1,NS)
       1161   FORMAT(15X,9(G10.3,1X))
              DO 1103 I=1,NS
              FACTOR=1.
              IF(I.EQ.7) FACTOR=180./PI
              IF(I.EQ.8) FACTOR=.5/PI
              IF(I.EQ.9) FACTOR=.5/PI
              YMIN=0.
              YMAX=0.
              DO 1101 LG1=1,LGMAX
       1101   EEME(LG1)=EERMS(LG1,I)*FACTOR
              YNAME1(1)=YNAM1(1,I)
              YNAME1(2)=YNAM1(2,I)
              K1=MOD(I,2)
              IF(K1.EQ.1)WRITE(7,11)
              IF(K1.EQ.1) CALL INTIO(6H  NRUN,NRUN,1,1)
               IF(K1.EQ.0) WRITE(7,10)
              CALL PLOTR(FSCA,EEME,LGMAX,XNAME1,YNAME1,YMIN,YMAX,0)
       1103   CONTINUE
              IF(.NOT.FILOUT) RETURN
              WRITE(7,11)
              WRITE(7,1170) (DATOUT(I),I=1,2)
       1170   FORMAT(13H OUTPUT FILE ,2A10,1H:/)
              WRITE(7,1172) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
       1172   FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
              WRITE(7,1174) IDENTS,KSTART
       1174   FORMAT(1H ,6(1X,A8,9X),6X,I3/)
              WRITE(7,1176) NRUN,DSNRDB,RHO,BETA,FSC,TSEP
       1176   FORMAT(1H ,5X,I2,6X,6(5X,G13.6)//)
              DO 1175 LGN=1,LGMAX
       1175   WRITE(7,1182) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
             *YYMA(LGN),FSCA(LGN),LGN
       1182   FORMAT(1H ,7(G13.6,5X),I5)
              WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
              WRITE(17) IDENTS,KSTART
              WRITE(17)NRUN,DSNRDB,RHO,BETA,BMLS,BRCVR,TSEP
              DO 1190 LGN=1,LGMAX
       1190   WRITE(17) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
             *YYMA(LGN),FSCA(LGN),LGN
              IX=IATTACH(6LTAPE20,DATOUT)
              IF(IX.NE.0) GO TO 1195
              READ(20) DUM,DUM,IDUM,IDUM,IDUM,DUM,IDUM
              CALL RETURN(6LTAPE20)
              REWIND 17
              IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
              GO TO 1196
```

```
1195  CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
1196  IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
      CALL RETURN(6LTAPE17)
      WRITE (7,1198) (DATOUT(I),I=1,2)
1198  FORMAT(13HOOUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
      RETURN
C
C*************************************************************************
C
1200  CONTINUE
C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
      WRITE(7,11)
      RETURN
C
C*************************************************************************
C
      END

45000B CM STORAGE USED     15.561 SECONDS
```

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C RMSE VS.  THETA SEP.
C
      REAL LAMDA
      INTEGER XNAME1(2),YNAME1(2),YNAM1(2,9)
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER TITLE1(9),FUNCTN(2)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV,
     CFILEIN,FILOUT
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TORO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION X(8),THESER(115),XDATUM(115),EEME(13),EERM(13)
      DIMENSION IDNRS(6),IDASCI(20),IDENTS(6),EEST(13),TTCO(13),YYMI(13)
      DIMENSION YYMA(13)
      DIMENSION TSEP(13)
      DIMENSION EEMEA(13,9),EERMS(13,9),EESTD(13,9)
      DIMENSION EMEAN(9),EMS(9),ERMS(9),ESD(9)
      EQUIVALENCE (EEME(1),EEMEA(1,2)),(EERM(1),EERMS(1,2)),
     *(EEST(1),EESTD(1,2))
      EQUIVALENCE(THERR,ES(2))
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA YNAM1/8HALF RMSE,8H        ,8HTHE RMSE,8HDEGREES ,
     *8HTDT RMSE,8HDEG/SEC ,8HALR RMSE,8H        ,8HTHR RMSE,
     *8HDEGREES ,8HTRD RMSE,8HDEG/SEC ,8HBET RMSE,8HDEGREES ,
     *8HFSC RMSE,8H   HZ   ,8HFSCDRMSE,8H HZ/SEC /
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H200000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7H       ,
     *8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/11/,TSEPO/-.75/,DTSEP/.25/,FILEIN
     */.FALSE./,FILOUT/.TRUE./
      DATA TITLE1/6H ALFA ,6H THETA,6HTHEDOT,6H ALFAR,6HTHETAR,6HTHRDOT
     C,6H   B  ,6H  WSC ,6HWSCDOT/
      DATA XNAME/8H  BETA  ,8HDEGREES /,YNAME/8HTHES ERR,8HDEGREES /
      DATA XNAME1/8HTHETASEP,8HDEGREES /
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
```

```
      C
            GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
      1     FORMAT(1H )
      10    FORMAT(9(/))
      11    FORMAT (1H1)
      C
      C*****************************************************************
      C
      100   CONTINUE
      C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
      CTHE FOLLOWING RECEIVER PARAMETERS
      CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JH,BMLS
            IF(NFIRST) GO TO 155
            NOAC=.TRUE.
            NFIRST=.TRUE.
            JBNAM=JOBNAME(I)
            ISIM=2
            IMOD=2
            WRITE (7,110)
      110   FORMAT(25H1RMSE VS. THESEP SCENARIO/)
            CALL LOGIO(6HFILEIN,FILEIN,0)
            IF (.NOT.FILEIN) NSTART=0
            IF(.NOT.FILEIN) GO TO 122
            IX=IATTACH(6LTAPE15,DATIN)
            IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
            IF(IX.NE.0) STOP#INPUT FILE ATTACH NOT SATISFACTORY#
            WRITE(7,120) (DATIN(I),I=1,2)
      120   FORMAT(23H THIS READS INPUT FILE ,2A10/)
      122   CALL LOGIO(6HFILOUT,FILOUT,0)
            CALL INTIO(6HNSTART,NSTART,100,0)
            NSTOP=NSTART
            CALL INTIO(6H NSTOP,NSTOP,100,0)
            CALL LOGIO(6HTETHRD,TETHRD,0)
            IF(IRCVR.EQ.1) GO TO 125
            CALL LOGIO(6HADAPTV,ADAPTV,0)
      125   CONTINUE
            IF(TETHRD) ITETHR=2
            IF(ADAPTV) GO TO 135
            IRSIGN=-1
            IADAP=3
      135   CONTINUE
            DO 145 I=2,6
            IASC=3*(I-1)+IDNRS(I)+2
            IDENTS(I)=IDASCI(IASC)
      145   CONTINUE
            ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
      150   FORMAT(A7,I1)
            WRITE (7,152) IDNRS,IMOD
      152   FORMAT (1H ,6I1,1X,I1/)
            WRITE (7,153) IDENTS
      153   FORMAT (1H ,A8)
            CALL INTIO(6H LGMAX,LGMAX,13,0)
            CALL INTIO(6H    KM,KM,115,0)
            CALL REALIO(6H TSEPO,TSEPO,0)
            CALL REALIO(6H DTSEP,DTSEP,0)
            KSTART=MINO(KM,KSTART)
            DO 154 I=1,LGMAX
```

```
        154   TSEP(I)=TSEPO+DTSEP*(I-1)
              KMNET=KM-KSTART+1
              FSCMIN=1./(DELTAT(IAE)*KMNET)
              NRUN=NSTART-1
              DSNRDB=20.
              RHO=0.5
              BETA=45.
              IFS=1
              BMLS=1.
              BRCVR=1.
        155   CONTINUE
              IF(FILEIN) GOTO 158
              NRUN=NRUN+1
              WRITE(7,11)
              CALL INTIO(6H  NRUN,NRUN,1,1)
              CALL REALIO(6HDSNRDB,DSNRDB,0)
              CALL REALIO(6H   RHO,RHO,0)
              CALL REALIO(6H  BETA,BETA,0)
              CALL INTIO(6H   IFS,IFS,100,0)
              CALL REALIO(6H  BMLS,BMLS,0)
              CALL REALIO(6H BRCVR,BRCVR,0)
              GOTO 167
        158   CONTINUE
              READ(15,160) NRUN,DSNRDB,RHO,BETA,IFS,BMLS,BRCVR
              IF(EOF(15))170,165
        160   FORMAT(I5,3G10.3,5X,I5,2G10.3)
        165   IF(NRUN.LT.NSTART) GO TO 158
        167   IF(NRUN.GE.NSTOP) MORE=.FALSE.
              FSC=IFS*FSCMIN
              XO(5,IAE)=XO(2,IAE)-TSEP(1)
              XO(3,IAE)=0.
              XO(6,IAE)=0.
              RETURN
        170   STOP*EOF REACHED ON INPUT FILE*
        C
        C**********************************************************************
        C
        200   CONTINUE
        C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
        CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
              WRITE(7,210) NRUN,DSNRDB,RHO,BETA,IFS,BMLS,BRCVR
        210   FORMAT(5HONRUN,12X,6HDSNRDB,16X,3HRHO,16X,4HBETA,16X,4HIFSC,16X,
             *4HBMLS,15X,5HBRCVR/1HO,I4,3(10X,G10.3),12X,I5,3X,
             *2(10X,G10.3))
              IF(.NOT.FILOUT) GOTO 245
              IF(NRUN.LE.9) ENCODE(10,220,DOUT) IDNRS,1HU,IMOD,1HO,NRUN
        220   FORMAT(6I1,A1,I1,A1,I1)
              IF(NRUN.GE.10) ENCODE(10,230,DOUT) IDNRS,1HU,IMOD,NRUN
        230   FORMAT(6I1,A1,I1,I2)
              IX=IREQST(6LTAPE17,3L*PF)
              IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
              IF(IX.NE.0) STOP*OUTPUT FILE REQUEST NOT SATISFACTORY*
              WRITE(7,240) (DATOUT(I),I=1,2)
        240   FORMAT(25HOTHIS WRITES OUTPUT FILE ,2A10)
        245   CALL DATE(TODAY)
              WRITE (7,250) FUNCTN (IAE),IAE
        250   FORMAT (1HO,A7,15H FUNCTION (IAE=,I1,1H))
```

```
                WRITE (7,260) ((I,X(I)),I=1,8)
          260   FORMAT (15HOINITIAL STATE://(3H X(,I1,4H) = ,G13.6))
                WRITE(7,270) IDENTS(2)
          270   FORMAT(1HO,A7)
                RETURN
          C
          C*********************************************************************
          C
          300   CONTINUE
          C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
          CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
          CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
                ITIT=MINO(IRCVR,2)
          C
          C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
                IR=ISIGN(IR,IRSIGN)
          C
                RETURN
          C
          C*********************************************************************
          C
          400   CONTINUE
          C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
                WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
          410   FORMAT (1HO,A8,4HRCVR/1HO,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
               *                    4H,NS=,I1,1H))
                WRITE(7,420) (IDENTS(I),I=5,6)
          420   FORMAT(1HO,A8/1HO,A7/)
                RETURN
          C
          C*********************************************************************
          C
          500   CONTINUE
          C THIS SETS-UP THE MSET-LOOP
                ICOUNT=0
                DO 510 I=1,LGMAX
                YYMI(I)=1.E322
          510   YYMA(I)=-1.E322
                RETURN
          C
          C*********************************************************************
          C
          600   CONTINUE
          C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
                RETURN
          C
          C*********************************************************************
          C
          700   CONTINUE
          C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
                X(5)=X(2)-TSEP(LG)
                X(7)=XO(7,IAE)
                DUM=GAUSS(1.0)
                CALL INTIO(6H  NRUN,NRUN,1,1)
                WRITE(7,705) LG,TSEP(LG),X(5)
          705   FORMAT(1H1,4X,3HLG=,I2,5X,7HTHESEP=,G13.6,5X,5HX(5)=,G13.6/)
                CALL INTIO(6H    KM,KM,1,1)
```

```
          WRITE(7,710) (TITLE1(I),I=1,NS)
     710  FORMAT(6H0    K,4X,5HCSNRT,4X,5HQTY  ,9(3X,A6,2X))
          DO 712 I=1,NS
          EMEAN(I)=0.
     712  EMS(I)=0.
          LASTCT=0
          ICOUNT=0
          DO 730 IPA=1,NS
          DO 720 JPA=1,NS
          PA(IPA,JPA)=0.
     720  CONTINUE
     730  CONTINUE
          RETURN
C
C**********************************************************************
C
     800  CONTINUE
C THIS INITIALIZES THE K-TH SCAN
          SEP=SPACE
          RETURN
C
C**********************************************************************
C
     900  CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
          IF(ICOUNT.NE.LASTCT) SEP=ABORT
          DO 905 I=1,NS
     905  ES(I)=X(I)-XS(I)
          IF(NS.GE.7) ES(7)=ABS(X(7))-ABS(XS(7))
          IF(NS.GE.8) ES(8)=ABS(X(8))-ABS(XS(8))
          BETTA=(180./PI)*X(7)
          COL8=THERR
          IF (IRCVR .NE. 1) COL8=SQRT(PPDIAG(2))
          IF (IRCVR .EQ. 1) CALL DFLTR1(COL8,THERR,FL10)
          IF((NRUN.EQ.NSTART.AND.LG.EQ.1).OR.(K.EQ.KM)) GOTO 906
          GO TO 950
     906  WRITE(7,910) K,CSNRT,(X(I),I=1,NS)
     910  FORMAT(3H0  ,I3,2X,G10.3,5H   X= ,2X,9(G10.3,1X))
          WRITE(7,920) (XS(I),I=1,NS)
     920  FORMAT(18X,5H XS= ,2X,9(G10.3,1X))
          WRITE(7,930) (ES(I),I=1,NS)
     930  FORMAT(18X,5H ES= ,2X,9(G10.3,1X))
          IF(IRCVR.EQ.1) WRITE(7,940) COL8
     940  FORMAT(8X,15HUNFILT. ES(2)= ,13X,G10.3)
     950  XDATUM(K)=BETTA
          THESER(K)=THERR
          LASTCT=ICOUNT
          IF(K.LT.KSTART) RETURN
          DO 960 I=1,NS
          EMEAN(I)=EMEAN(I)+ES(I)
     960  EMS(I)=EMS(I)+ES(I)**2
          RETURN
C
C**********************************************************************
C
    1000  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
```

```
               DO 1005 I=1,NS
               EMEAN(I)=EMEAN(I)/KMNET
               EMS(I)=EMS(I)/KMNET
               ERMS(I)=SQRT(EMS(I))
               ESD(I)=SQRT(EMS(I)-EMEAN(I)**2)
         1005  CONTINUE
               XW=ICOUNT
               TCOUNT=100.*XW/KM
               WRITE(7,1009)
         1009  FORMAT(3(/))
               WRITE(7,1010)KMNET
         1010  FORMAT(5X,26HERROR SAMPLE STATISTICS: (,I3,9H SAMPLES))
               IF(IRCVR.EQ.1) WRITE (7,1015)
         1015  FORMAT(5X,31HTHRESHOLD RCVR (FILTERED ERROR))
               WRITE(7,1020)(TITLE1(I),I=1,NS)
         1020  FORMAT(1H0,20X,4HQTY ,9(3X,A6,2X))
               WRITE(7,1030)(EMEAN(I),I=1,NS)
         1030  FORMAT(1H0,16X,8HEMEAN = ,9(G10.3,1X))
               WRITE(7,1040)(ERMS(I),I=1,NS)
         1040  FORMAT(18X,7HERMS = ,9(G10.3,1X))
               WRITE(7,1050)(ESD(I),I=1,NS)
         1050  FORMAT(19X,6HESD = ,9(G10.3,1X))
               IF(IRCVR.EQ.1)WRITE(7,1060)TCOUNT
         1060  FORMAT(//1H ,4X,F7.2,22H% OF SCANS ARE ABORTED)
               IF(IRCVR.NE.1)WRITE(7,1070)(GQGT(I,I),I=1,NS)
         1070  FORMAT(5(/),4X,35H ON LAST SCAN, DIAGONAL OF GQGT WAS/26X
              C9(G10.3,1X))
               DO 1071 I=1,NS
               EEMEA(LG,I)=EMEAN(I)
               EERMS(LG,I)=ERMS(I)
         1071  EESTD(LG,I)=ESD(I)
               DO 1072 I=1,KM
               YYMI(LG)=AMIN1(THESER(I),YYMI(LG))
         1072  YYMA(LG)=AMAX1(THESER(I),YYMA(LG))
               TTCO(LG)=TCOUNT
               IF(LG.NE.1) RETURN
               YMIN=YMAX=0.
               WRITE(7,11)
               CALL PLOTR(XDATUM,THESER,KM,XNAME,YNAME,YMIN,YMAX,0)
               RETURN
         C
         C**********************************************************************
         C
         1100  CONTINUE
         C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
         CLGMAX SETS OF KM SCANS
               WRITE(7,11)
               CALL INTIO(6H  NRUN,NRUN,1,1)
               WRITE(7,1110) 6HTHESEP,(TITLE1(I),I=1,NS)
         1110  FORMAT(1H0,10X,3HQTY  ,9(3X,A6,2X))
               WRITE(7,1120)TSEP(1),(EEMEA(1,I),I=1,NS)
         1120  FORMAT(1H0, 7X,7HEMEAN: ,9(G10.3,1X))
               DO 1130 J=2,LGMAX
         1130  WRITE(7,1161) TSEP(J),(EEMEA(J,I),I=1,NS)
               WRITE(7,1140)TSEP(1),(EERMS(1,I),I=1,NS)
         1140  FORMAT(1H0, 8X,6HERMS: ,9(G10.3,1X))
               DO 1150 J=2,LGMAX
```

```
 1150 WRITE(7,1161) TSEP(J),(EERMS(J,I),I=1,NS)
      WRITE(7,1160)TSEP(1),(EESTD(1,I),I=1,NS)
 1160 FORMAT(1HO, 8X,6HESTD: ,9(G10.3,1X))
      DO 1165 J=2,LGMAX
 1165 WRITE(7,1161) TSEP(J),(EESTD(J,I),I=1,NS)
 1161 FORMAT(15X,9(G10.3,1X))
      IF (.NOT.FILOUT) GOTO 1199
      WRITE(7,11)
      WRITE(7,1170) (DATOUT(I),I=1,2)
 1170 FORMAT(13H OUTPUT FILE ,2A10,1H:/)
      WRITE(7,1172) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
 1172 FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,1174) IDENTS,KSTART
 1174 FORMAT(1H ,6(1X,A8,9X),6X,I3/)
      WRITE(7,1176) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
 1176 FORMAT(1H ,5X,I2,6X,6(5X,G13.6)/)
      DO 1175 LGN=1,LGMAX
 1175 WRITE(7,1182) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
     *YYMA(LGN),TSEP(LGN),LGN
 1182 FORMAT(1H ,7(G13.6,5X),I5)
      WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      WRITE(17) IDENTS,KSTART
      WRITE(17) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
      DO 1190 LGN=1,LGMAX
 1190 WRITE(17) EEME(LGN),EERM(LGN),EEST(LGN),TTCO(LGN),YYMI(LGN),
     *YYMA(LGN),TSEP(LGN),LGN
      IX=IATTACH(6LTAPE20,DATOUT)
      IF(IX.NE.0) GO TO 1195
      READ(20) DUM,DUM,IDUM,IDUM,IDUM,DUM,IDUM
      CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
      GO TO 1196
 1195 CALL RETURN(6LTAPE20)
      REWIND 17
      IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1196 IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
      CALL RETURN(6LTAPE17)
      WRITE (7,1198) (DATOUT(I),I=1,2)
 1198 FORMAT(13HOOUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
 1199 CONTINUE
      DO 1103 I=1,NS
      FACTOR=1.
      IF(I.EQ.7) FACTOR=180./PI
      IF(I.EQ.8) FACTOR=.5/PI
      IF(I.EQ.9) FACTOR=.5/PI
      YMIN=0.
      YMAX=0.
      DO 1101 LG1=1,LGMAX
 1101 EEME(LG1)=EERMS(LG1,I)*FACTOR
      YNAME1(1)=YNAM1(1,I)
      YNAME1(2)=YNAM1(2,I)
      K1=MOD(I,2)
      IF(K1.EQ.1)WRITE(7,11)
      IF(K1.EQ.1) CALL INTIO(6H NRUN,NRUN,1,1)
         IF(K1.EQ.0) WRITE(7,10)
```

```
            CALL PLOTR(TSEP,EEME,LGMAX,XNAME1,YNAME1,YMIN,YMAX,0)
      1103  CONTINUE
            RETURN
      C
      C*************************************************************************
      C
      1200  CONTINUE
      C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
            RETURN
      C
      C*************************************************************************
      C
            END
```

45000B CM STORAGE USED 12.547 SECONDS

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C     STUDY OF LOE PERFORMANCE
C     THIS VERSION INVESTIGATES THE ERROR PERFORMANCE
C
      REAL LAMDA
       REAL E(2),TE(2),MSE
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV
      INTEGER FUNCTN(2)
      REAL XNA(2),YNA(2)
      COMMON/IODATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XD(8,2),YD(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION X(8),TSEP(20),BETAE(20),BULK(25,10),BULL(15,10),XE(8)
       DIMENSION DMAT(5,5)
      DIMENSION NAMES(11),NAME5(11),NAME8(5)
      DIMENSION IDNRS(6),IDASCI(24),IDENTS(6)
      DIMENSION AK(50),ATE(50),PABORT(10),PABMAX(10)
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1))
      DATA ABORT/1HA/,SPACE/1H /,ADAPTV/.TRUE./
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7HACQSITN,
     *2*7H       ,7HLOERMSE,7HGENERAL,8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA KSTART/11/,IRSIGN/1/,KMNET/20/
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
       DATA TSEPMN/-0.50/,DTSEP/0.25/
       DATA NAMES/6H E1AVG,6H E2AVG,6H E1MIN,6H E2MIN,
     * 6H E1MAX,6H E2MAX,6H E1RMS,6H E2RMS,6HARIAVG,6HGEOAVG,
     * 6H  RMSE/
       DATA NAME5/3HAVG,3HAVG,3HMIN,3HMIN,3HMAX,3HMAX,3HRMS,3HRMS,
     * 3HAVG,3HAVG,3HAVG/
      DATA NAME8/6H TEAVG,6H TEMIN,6H TEMAX,6H TERMS,2H  /
      DATA BETMIN/0./
      DATA XNA/6H  SCAN  ,8H   K    /
      DATA YNA/8H X2-XS2 ,8H        /
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
    1 FORMAT(1H )
   10 FORMAT(9(/))
   11 FORMAT (1H1)
```

```
C
C*************************************************************************
C
 100  CONTINUE
C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
CTHE FOLLOWING RECEIVER PARAMETERS
CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JM,BMLS
      MORE=.FALSE.
      JBNAM=JOBNAME(I)
      ISIM=6
      IMOD=3
       WRITE (7,110)
  110 FORMAT (22H1LOE PERFORMANCE STUDY)
      CALL INTIO (6H KMNET,KMNET,100,0)
      DELBL=.01
      LGMAX=10
      MTIMES=20
       TETHRD=.TRUE.
      NOLOE=.FALSE.
      NOKLMN=.TRUE.
      CALL REALIO (6HDSNRDB,DSNRDB,0)
      CALL REALIO (6H   RHO,RHO,0)
       CALL REALIO (6HTSEPMN,TSEPMN,0)
       CALL REALIO (6H DTSEP,DTSEP,0)
       CALL INTIO (6H NTSEP,LGMAX,20,0)
       CALL REALIO(6HBETMIN,BETMIN,0)
       CALL INTIO(6HNOBSEP,MTIMES,20,0)
       DBETAE=360./MTIMES
       DO 120 I=1, LGMAX
  120  TSEP(I)=TSEPMN+(I-1)*DTSEP
       DO 126 I=1,MTIMES
  126  BETAE(I)=BETMIN+(I-1)*DBETAE
       XO(3,IAE)=0.
       XO(5,IAE)=XO(2,IAE)-TSEPMN
       XO(6,IAE)=0.
       BETA=BETMIN
       FSC=0.
      CALL LOGIO(6HTETHRD,TETHRD,0)
      IF(IRCVR.EQ.1) GO TO 125
      CALL LOGIO(6H NOLOE,NOLOE,0)
      CALL LOGIO(6HNOKLMN,NOKLMN,0)
      IF (NOKLMN) KSTART=1
      CALL LOGIO(6HADAPTV,ADAPTV,0)
 125  CONTINUE
      KM=KSTART+KMNET-1
      CALL INTIO (6HKSTART,KSTART,1,1)
      CALL INTIO (6H    KM,KM,1,1)
      IF(TETHRD) ITETHR=2
      IF(ADAPTV) GO TO 135
      IRSIGN=-1
      IADAP=3
 135  CONTINUE
      DO 145 I=2,6
      IASC=3*(I-1)+IDNRS(I)+6
      IDENTS(I)=IDASCI(IASC)
 145  CONTINUE
      ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
```

```
  150 FORMAT(A7,I1)
      WRITE (7,152) IDNRS,IMOD
  152 FORMAT (1H ,6I1,1X,I1/)
      WRITE (7,153) IDENTS
  153 FORMAT (1H ,A8)
      RETURN
C
C**********************************************************************
C
  200 CONTINUE
C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
      CALL DATE(TODAY)
      WRITE (7,250) FUNCTN (IAE),IAE
  250 FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
      WRITE (7,260) ((I,X(I)),I=1,8)
  260 FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
      WRITE(7,270) IDENTS(2)
  270 FORMAT(1H0,A7)
      RETURN
C
C**********************************************************************
C
  300 CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
      ITIT=MIN0(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
      IF(IRSIGN.LT.0) IR=-IR
C
      RETURN
C
C**********************************************************************
C
  400 CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
      WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
  410 FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
     *        4H,NS=,I1,1H))
      WRITE(7,420) (IDENTS(I),I=5,6)
  420 FORMAT(1H0,A8/1H0,A7/)
      RETURN
C
C**********************************************************************
C
  500 CONTINUE
C THIS SETS-UP THE MSET-LOOP
      ICOUNT=0
      DO 502 J=1,LGMAX
      BULL(11,J)=0.
      BULL(12,J)=+1.E322
      BULL(13,J)=-1.E322
      BULL(14,J)=0.
       BULK(15,J)=0.
       BULK(16,J)=0.
```

```
      BULK(17,J)=+1.E322
      BULK(18,J)=+1.E322
      BULK(19,J)=-1.E322
      BULK(20,J)=-1.E322
      PABMAX(J)=-1.E322
       BULK(21,J)=0.
       BULK(22,J)=0.
       BULK(23,J)=0.
       BULK(24,J)=0.
       BULK(25,J)=0.
  502  CONTINUE
      CALL INTIO (6HMTIMES,MTIMES,1,1)
       WRITE(7,510)
 510  FORMAT(1H1,40X,21HLOE ERROR PERFORMANCE/)
       WRITE(7,511) (TSEP(I),I=1,LGMAX)
  511  FORMAT(1H0,7HTHESEP=,7X,10(1X,F8.5,2X))
       WRITE(7,512)
 512  FORMAT(1H0,5HBETA:)
      RETURN
C
C***********************************************************************
C
600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
       BETA=BETAE(MSET)
       XO(7,IAE)=PI*BETA/180.
       X(7) = XO (7,IAE)
      IF(MOD(MSET,8).EQ.0) WRITE(7,511)(TSEP(I),I=1,LGMAX)
      RETURN
C
C***********************************************************************
C
700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
       THESEP=TSEP(LG)
       X(5) = X(2)-THESEP
       DUM=GAUSS (1.0)
       LASTCT=0
       ICOUNT=0
       FL10(4)=0.
      THETAV=0.
      THETMN=+1.E322
      THETMX=-1.E322
      THERMS=0.
       E1AVG=0.
       E2AVG=0.
      E1MIN=+1.E322
      E2MIN=+1.E322
      E1MAX=-1.E322
      E2MAX=-1.E322
       E1RMS=0.
       E2RMS=0.
       MSE=0.
      LASTCT=0
      DO 705 I=1,8
      DO 705 J=1,8
705   PA(I,J)=0.
```

```
      RETURN
C
C*********************************************************************
C
800   CONTINUE
C THIS INITIALIZES THE K-TH SCAN
      DO 810 I=1,5
810   RDIAG(I)=1.
      IF(TETHRD) RETURN
      RETURN
C
C*********************************************************************
C
900   CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
C
C     FOLG SHOULD BE IN RCVR
C
      XS(1)=ABS(XS(1))
      XS(4)=ABS(XS(4))
      IF(ICOUNT.NE.LASTCT) SEP=ABORT
      THERR=X(2)-XS(2)
      IF (IRCVR.EQ.1) CALL DFLTR1 (THERR,THERR,FL10)
      LASTCT=ICOUNT
      IF(K.LT.KSTART) RETURN
       TE(1)=X(1)-XS(1)
       TE(2)=THERR
      THETAV=THETAV+TE(2)
      THETMN=AMIN1(THETMN,TE(2))
      THETMX=AMAX1(THETMX,TE(2))
      THERMS=THERMS+(TE(2)**2)
      IF(IRCVR.EQ.1) RETURN
      IF((RDIAG(1).GT.1.E-293).AND.(RDIAG(2).GT.1.E-293)) GOTO 910
       CALL MATOUT(6H PHI =,PHI,NG,NG,5,1)
       CALL MATOUT(6HRMAT =,RMAT,NG,NG,5,1)
       CALL MATMUL(5,NG,NG,5,NG,NG,NG,NG,RMAT,PHI,DMAT,1)
       CALL MATOUT(6HR*PHI=,DMAT,NG,NG,5,1)
      STOP *BAD MATRIX PHI*
 910   CONTINUE
       SIG1=SQRT(RDIAG(1))
       SIG2=SQRT(RDIAG(2))
       E(1)=TE(1)/SIG1
       E(2)=TE(2)/SIG2
       E1AVG=E1AVG+E(1)
       E2AVG=E2AVG+E(2)
       E1MIN=AMIN1(E(1),E1MIN)
       E2MIN=AMIN1(E(2),E2MIN)
       E1MAX=AMAX1(E(1),E1MAX)
       E2MAX=AMAX1(E(2),E2MAX)
       E1RMS=E1RMS+(E(1)**2)
       E2RMS=E2RMS+(E(2)**2)
       DO 920 I=1,2
       DO 920 J=1,2
 920   MSE=MSE+TE(I)*TE(J)*PHI(I,J)
      RETURN
C
C*********************************************************************
```

```
      C
      1000  CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
            BULL(1,LG)=THETAV/KMNET
            BULL(2,LG)=THETMN
            BULL(3,LG)=THETMX
            BULL(4,LG)=SQRT(THERMS/KMNET)
            BULL(11,LG)=BULL(11,LG)+BULL(1,LG)
            BULL(12,LG)=AMIN1(BULL(12,LG),THETMN)
            BULL(13,LG)=AMAX1(BULL(13,LG),THETMX)
            BULL(14,LG)=BULL(14,LG)+(THERMS/KMNET)
            PABORT(LG)=(100.*ICOUNT)/KM
            PABMAX(LG)=AMAX1(PABMAX(LG),PABORT(LG))
            IF (IRCVR.EQ.1) RETURN
            BULK(1,LG)=E1AVG/KMNET
            BULK(2,LG)=E2AVG/KMNET
             BULK(3,LG)=E1MIN
             BULK(4,LG)=E2MIN
             BULK(5,LG)=E1MAX
             BULK(6,LG)=E2MAX
            BULK(7,LG)=SQRT(E1RMS/KMNET)
            BULK(8,LG)=SQRT(E2RMS/KMNET)
             BULK(9,LG)=SQRT((BULK(7,LG)**2+BULK(8,LG)**2)/2)
             BULK(10,LG)=SQRT(BULK(7,LG)*BULK(8,LG))
            BULK(11,LG)=SQRT(MSE/KMNET)
             BULK(15,LG)=BULK(15,LG)+BULK(1,LG)
             BULK(16,LG)=BULK(16,LG)+BULK(2,LG)
             BULK(17,LG)=AMIN1(BULK(17,LG),BULK(3,LG))
             BULK(18,LG)=AMIN1(BULK(18,LG),BULK(4,LG))
             BULK(19,LG)=AMAX1(BULK(19,LG),BULK(5,LG))
             BULK(20,LG)=AMAX1(BULK(20,LG),BULK(6,LG))
             BULK(21,LG)=BULK(21,LG)+BULK(7,LG)**2
             BULK(22,LG)=BULK(22,LG)+BULK(8,LG)**2
             BULK(23,LG)=BULK(23,LG)+BULK(9,LG)**2
             BULK(24,LG)=BULK(24,LG)+BULK(10,LG)**2
             BULK(25,LG)=BULK(25,LG)+BULK(11,LG)**2
            RETURN
      C
      C************************************************************************
      C
      1100  CONTINUE
      C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
      CLGMAX SETS OF KM SCANS
            WRITE(7,1101) BETA
      1101  FORMAT(1H0,F7.2)
            DO 1160 I=1,4
            WRITE(7,1150) NAMEB(I),(BULL(I,J),J=1,LGMAX)
       1150 FORMAT(1H ,7X,A6,1H=,10(1X,G10.3))
       1160 CONTINUE
            IF(IRCVR.EQ.1) WRITE(7,1150) 6H%ABORT,(PABORT(J),J=1,LGMAX)
            IF(IRCVR.EQ.1)  RETURN
            WRITE(7,1103) (BULK(1,J),J=1,LGMAX)
      1103  FORMAT(1H ,7X,7H E1AVG=,10(1X,G10.3))
             DO 1190 I=2,11
             WRITE(7,1115) NAMES(I),(BULK(I,J),J=1,LGMAX)
       1115  FORMAT (1H ,7X,A6,1H=,10(1X,G10.3))
       1190  CONTINUE
```

```
      RETURN
C
C*********************************************************************
C
 1200 CONTINUE
C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
       DO1220 I=1,LGMAX
      BULL(11,I)=BULL(11,I)/MTIMES
      BULL(14,I)=SQRT(BULL(14,I)/MTIMES)
      IF(IRCVR.EQ.1)  GOTO 1220
       BULK(15,I)=BULK(15,I)/MTIMES
       BULK(16,I)=BULK(16,I)/MTIMES
       BULK(21,I)=SQRT(BULK(21,I)/MTIMES)
       BULK(22,I)=SQRT(BULK(22,I)/MTIMES)
       BULK(23,I)=SQRT(BULK(23,I)/MTIMES)
       BULK(24,I)=SQRT(BULK(24,I)/MTIMES)
       BULK(25,I)=SQRT(BULK(25,I)/MTIMES)
 1220  CONTINUE
       WRITE(7,1230)
 1230  FORMAT(1H0,9HWRT BETA:/)
      DO 1233 I=11,14
       I11=I-10
      WRITE(7,1234) NAME5(2*I11),NAME8(I11),(BULL(I,J),J=1,LGMAX)
 1234 FORMAT(1H ,3X,A4,A6,1H=,10(1X,G10.3))
 1233 CONTINUE
      IF(IRCVR.EQ.1)WRITE(7,1234)4HMAX ,6H%ABGRT,(PABMAX(J),J=1,LGMAX)
      IF (IRCVR.EQ.1) GOTO 1248
       DO 1245 I=15,25
       IT=I-14
       WRITE(7,1240) NAME5(IT),NAMES(IT),(BULK(I,J),J=1,LGMAX)
 1240  FORMAT(1H ,3X,A4,A6,1H=,10(1X,G10.3))
 1245  CONTINUE
 1248 CONTINUE
       WRITE(7,511) (TSEP(I),I=1,LGMAX)
       WRITE(7,1250)
 1250  FORMAT(1H0)
      IF(IRCVR.NE.1) WRITE(7,1260)(GQGT(I,I),I=1,NS)
 1260 FORMAT(13H GQGT(I,I) = ,G13.6,7(1X,G13.6))
      RETURN
C
C*********************************************************************
C
      END
```

44000B CM STORAGE USED 7.674 SECONDS

```
      SUBROUTINE MLSSUB
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TORO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
C GENERAL SIMULATION:  OPTIMIZATION OF MLS RECEIVERS FOR
C       MULTIPATH ENVIRONMENTS
      DIMENSION X1(8),INDEX(2),Y(4),X(8)
      EQUIVALENCE (THAMAX,Y(1)),(ALFA,X(1))
C
C  FOLG BEGINS EXECUTABLE STATEMENTS
C
      PI2=2.*PI
      SQ22=SQRT(.5)
C
C     FOLG INITIALIZES THE DIAGONAL OF F(8,8)
C
9     DO 10 I=1,8
      F(I,I)=1.
10    CONTINUE
      CALL CONTRL(1)
      SSQ2=SIGMA*SQRT(2.)
C
C       FOLLOWING COMPLETES INITIALIZATION AND COMPUTES SECONDARY
C       PARAMETERS.
C
      DO 15 I=1,4
      FL10(I)=FL10AE(I,IAE)
15    CONTINUE
      DO 20 I=1,4
      Y(I)=YO(I,IAE)
20    CONTINUE
      OMEGA=-(THAMAX-THAMIN)/TS
      SECPD=1./OMEGA
      TF=TS+TR-2.*THAMIN*SECPD
C
C FOLG ARE CONSTANT PARAMETERS USED BY P
C
      BBB=2.4/BMLS
      DUM=GAUSS(1.0)
      ALPHA=10.**(DSNRDB/20.)
      XO(1,IAE)=ALPHA
      ALPHAR=RHO*ALPHA
```

```
      PROGRAM MLSSIM(INPUT,OUTPUT,TAPE10=INPUT,TAPE12,TAPE15,
     *TAPE7=OUTPUT,TAPE17,TAPE20)
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRO,MORE
      COMMON /IODATA/IDNRS(6)
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSD(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,ED(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),ROIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
C
   10 CONTINUE
      CALL MLSSUB
      IF(MORE) GO TO 10
      STOP
      END
```

410008 CM STORAGE USED   .134 SECONDS

```
      FUNCTION THA(T)
C
C      THIS COMPUTES THE ANTENNA SCAN ANGLE AT LOCAL SCAN TIME T.
C      THE PARAMETERS OF THE SCAN WAVEFORM,HENCE THE IDENTITY OF
C      THE ANGLE FUCTION,ARE PASSED THROUGH COMMON AND ARE DEFINED
C      AS FOLLOWS:
C      TR=TIME BETWEEN TRAVERSAL OF ZERO DEGREES IN A TO-FRO SCAN
C      TS=DURATION OF THE TO SCAN
C      TF=DURATION OF THE TO SCAN+INTERSCAN REST INTERVAL
C THAMAX=ANTENA ANGLE AT BEGINNING OF TO SCAN
C THAMIN=ANTENNA ANGLE AT END OF TO SCAN
C OMEGA=ANTENNA ANGULAR SCAN RATE,DEG/SEC,DURING TO SCAN
C
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
        T1=TS+TF
        IF(T.GE.0.0) GO TO 50
      THA=THAMAX
        RETURN
 50     IF(T.GT.TS)GO TO 100
      THA=THAMAX+OMEGA*T
        RETURN
 100    IF(T.GE.TF) GO TO 200
      THA=THAMIN
        RETURN
 200    IF(T.GT.T1) GO TO 250
      THA=THAMIN-OMEGA*(T-TF)
        RETURN
 250  THA=THAMAX
        RETURN
        END
```

41000B CM STORAGE USED .118 SECONDS

```
          XO(4,IAE)=ALPHAR
          XO(7,IAE)=PI*BETA/180.
          XO(8,IAE)=PI2*FSC
          TSAMP=1./FSAMP
          DELT=DELTAT(IAE)
          F(2,3)=DELT
          F(5,6)=DELT
          F(7,8)=DELT
          JM1=JM+1
          JM2=JM/2
          TWW2=0.5*TSAMP*(JM2+1)
          DO 30 I=1,8
          X(I)=XO(I,IAE)
   30     CONTINUE
          CALL CONTRL(2)
C
C FOLG CALLS RECEIVER FOR INITIALIZATION
C
          K=0
          CALL RCVR
C
C FOLLOWING BEGINS SIMULATION PER SE, LGMAX RUNS OF KM SCANS EACH
C
          CALL CONTRL(5)
          DO 777 MMSET=1,MTIMES
          MSET=MMSET
C
          CALL CONTRL(6)
          DO 1 LGG=1,LGMAX
          LG=LGG
          CALL CONTRL(7)
          DO 2 KK=1,KM
          K=KK
C
          CALL CONTRL(8)
          IF (K .EQ. 1) GO TO 122
C
C   FOLLOWING ADVANCES THE TRUE STATE AND SAVE PRIOR VALUE
C
          DO 100 I=1,8
          X1(I)=X(I)
          X(I)=0.
  100     CONTINUE
          DO 120 I=1,8
          DO 110 J=1,8
          X(I)=X(I)+F(I,J)*X1(J)
  110     CONTINUE
  120     CONTINUE
          B=PVALUE(B,PI,0.)
C
C FOLG SETS THES PRIOR TO COMPUTING SAMPLE TIMES
          THES=XS(2)+DELT*XS(3)
          IF(.NOT.TETHRO) GO TO 124
  122     CONTINUE
          DO 123 I=1,8
          XS1(I)=X(I)-EO(I)
  123     CONTINUE
```

```
      THES=XS1(2)
  124 CONTINUE
C
C FOLG BEGINS COMPUTATION OF SIGNALS AND RELATED QUANTITIES
C     FIRST, CALCULATION OF CONSTANTS ON THE PRESENT SCAN
C
      AL2=ALFA**2
      AR2=ALFAR**2
      AA2=2.*ALFA*ALFAR
      TPST=(THES-THAMAX)*SECPD
      TPSF=TF+(THAMIN-THES)*SECPD
      TZT=TPST-TWW2
      TZF=TPSF+TWW2
C FOLG COMPUTES FOR THIS KTH SCAN THE SIGNAL PEAK TIMES TPKT, TPKF AND
C THE COMPOSITE SIGNAL-TO-NOISE RATIO  CSNR(K)
      TPKT=(X(2)-THAMAX)*SECPD
      TPKF=TF+(THAMIN-X(2))*SECPD
C
C     THAJ=THA(TPKT)=X(2), BY DEFINITION
C PJ=PMLS(THE-THA(TPKT))=1.0,BY DEFINITION
C AND SIMILARLY FOR TPKF
C
      PRJ=PMLS(X(5)-X(2))
      BT=PVALUE(B+WSC*TPKT,PI,0.)
      BF=PVALUE(B+WSC*TPKF,PI,0.)
      QJT=AL2+AA2*PRJ*COS(BT)+AR2*(PRJ**2)
      QJF=AL2+AA2*PRJ*COS(BF)+AR2*(PRJ**2)
      CSNRT=SQRT(QJT)
      CSNRF=SQRT(QJF)
      CSNRMN=AMIN1(CSNRT,CSNRF)
      CSNRMX=AMAX1(CSNRT,CSNRF)
      CSNR=CSNRMN
      DCSNR=CSNRMX-CSNRMN
C
C FOLG INITIATES A J-LOOP TO COMPUTE SAMPLE TIMES AND (LINEAR)
C     ENVELOPE SAMPLE VALUES
C
      DO 130 J=1,JM2
      TINCR=J*TSAMP
      JFR=JM1-J
      INDEX (1)=J
      INDEX (2)=JFR
C FOLG COMPUTES SAMPLE TIMES
      TJ=TZT+TINCR
      T(J)=TJ
      T(JFR)=TZF-TINCR
      THAJ=THA(TJ)
      PJ=PMLS(THE-THAJ)
      PRJ=PMLS(THR-THAJ)
C FOLG COMPUTES ENVELOPE SAMPLES
      DO 127 I=1,2
      JVAL=INDEX(I)
      XNC=SQ22*GAUSS(0.)
      XNS=SQ22*GAUSS(0.)
      BLOCAL=PVALUE(B+WSC*T(JVAL),PI,0.)
      QJ=AL2*(PJ**2)+AA2*PJ*PRJ*COS(BLOCAL)+AR2*(PRJ**2)
      QJ=AMAX1(QJ,0.0)
```

```
              SQRTQ2=2.*SQRT(QJ)
              UJ=QJ+SQRTQ2*XNC+XNC**2+XNS**2
              V(JVAL)=SSQ2*SQRT(UJ)
        127   CONTINUE
        130   CONTINUE
        C
        C      FOLLOWING COMPUTES RECEIVER RESPONSES
        C
              CALL RCVR
        C
        C      FOLLOWING ASSIMILATES DATA FROM THIS SCAN FOR FUTURE EVALUATION
        C
              CALL CONTRL(9)
        2     CONTINUE
        C
        C FOLLOWING ASSIMILATES DATA FROM THIS LG-TH SET OF SCANS
        C
              CALL CONTRL(10)
        1     CONTINUE
        C
        C FOLLOWING ASSIMILATES DATA FROM THIS MSET-TH SERIES OF LGMAX SETS OF KM SCANS
        C
              CALL CONTRL(11)
        777   CONTINUE
        C
        C FOLLOWING CLOSES THE SIMULATION RUN
        C
              CALL CONTRL(12)
              RETURN
              END

1000B CM STORAGE USED    1.399 SECONDS
```

```
      BLOCK DATA MLS
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE
      COMMON/IDDATA/IDNRS(6)
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(8,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DATA PI/3.1415927/,F/64*0./,DSNRDB/20./,RHO/.5/
      DATA DELTAT/.074074074,.024691358/,SIGMA/.7071/
      DATA XO/0.,30.,.35,0.,32.75,-.35,0.,0.,
     *        0.,10.,0.,0.,12.,-.9,0.,2./
      DATA YO/62.666667,-62.,6.233333E-3,6.6E-3,
     *30.666667,0.,1.533333E-3,0.4E-3/
      DATA FL10AE/.5,0.,-.5,0.,.25,0.,-.75,0./
      DATA FL25/.34831,.34831,-.30336,0./
      DATA JM/130/,TETHRD/.FALSE./,KM/100/,FSAMP/1.6E5/
      DATA H/1.,5*0.,1.,10*0.,1.,5*0.,1.,10*0.,1.,5*0./
      DATA NSO/2,8,6,3/,NGO/2,5,4,2/
      DATA DELBL/.01/,EO/8*0./
      DATA GQGT/64*0./,PA/64*0./
      DATA RHOMAX/.8/,DTHO/2.75/,TDRO/0.0/,BO/0.0/,WSCO/0.0/
      DATA NOLOE/.TRUE./,NOKLMN/.FALSE./,NOAC/.TRUE./
      DATA IAE/1/,FSC/51.30/,BETA/-168.0/,LGMAX/1/
      DATA BMLS/1.0/,BRCVR/1.0/,MTIMES/1/,MORE/.TRUE./
      DATA IDNRS(5)/1/
      END

 41000B CM STORAGE USED          .209 SECONDS
```

```
      SUBROUTINE DFLTR1(X,Y,C)
      DIMENSION C(4)
      S=X-C(3)*C(4)
      Y=C(1)*S+C(2)*C(4)
5     C(4)=S
      RETURN
      END
```

41000B CM STORAGE USED .043 SECONDS

```
      SUBROUTINE RCVR
      REAL LAMDA,MJM2
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TORO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),ROIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/X(8)
      DIMENSION TPA1(8,8),PHT(8,5),TPA2(8,8),IVNG(5)
      DIMENSION RVNG(5),GAIN(8,5)
C
      IF (K .GT. 0) GO TO 40
C FOLG IS INITIALIZATION
      PI2=2.*PI
      SS2=.5/(SIGMA**2)
      CALL PHILM
      NS=NSO(IR)
      NG=NGO(IR)
      NGMIN=2
      NGMAX=NGO(IR)
      GQGT33=.01*DELT
      GQGT(3,3)=GQGT33
      GQGT(6,6)=GQGT33
      GQGT(8,8)=.04/(DELT**2)
      DO 20 I=1,NS
      DO 10 J=1,NS
      PA(I,J)=0.
   10 CONTINUE
   20 CONTINUE
      CALL CONTRL(3)
      BB=2.4/BRCVR
      PDCRIT=PI*BB/8.
      CC=PI*PDCRIT
      IF(IR.GT.0) GO TO 30
      NS=NSO(4)
      NG=NGO(4)
   30 CONTINUE
      NGM=NGMIN
      IF(NOAC) NGM=NG
      KALMAN=.NOT. NOKLMN
      LOE=.NOT. NOLOE
      CALL CONTRL(4)
      RETURN
   40 CONTINUE
      IF(K.GT.1) GO TO 50
C
C DIAGNOSTIC OUTPUT OF INPUT DATA FOR K=1 GOES HERE
C
      GO TO 90
   50 CONTINUE
```

```
C FOLG EXTRAPOLATES STATE ESTIMATE, AS REQ,D
      IF(TETHRD) GO TO 80
      DO 70 I=1,NS
      XS1(I)=0.
      DO 60 J=1,NS
      XS1(I)=XS1(I)+F(I,J)*XS(J)
60    CONTINUE
70    CONTINUE
80    CONTINUE
90    CONTINUE
C
      GQGT11=AMAX1(.25,.01*(XS1(1)**2))
      GQGT(1,1)=GQGT11
      GQGT(4,4)=GQGT11
C
C FOLLOWING COMPUTES VECTORS Q(JM),HW(JM),LAMDA(NG),AND MATRICES DT(JM,
C NG),PHI(NG,NS) ALSO SQUARED AMPLITUDE ENVELOPE VECTOR U(JM),AND
C INOVATION;S PROCESS VECTOR W(JM)
C
      CALL PHILM
      IF (NOLOE) GO TO 120
C
C FOLG COMPUTES RMAT=PHI-INVERSE AND THE LOE
      CALL MATINV(RMAT,5,NG,IVNG,RVNG)
      DO 100 I=1,NG
      RDIAG (I)=RMAT (I,I)
100   CONTINUE
      CALL MATMUL (5,NG,NG,5,NG,1,5,1,RMAT,LAMDA,GAMAES,1)
      CALL MATMUL (5,NG,NS,5,NG,1,8,1,H,GAMAES,ESLOE,2)
      DO 110 I=1,NS
      XSLOE(I)=XS1(I)+ESLOE(I)
      IF (NOKLMN) XS(I)=XSLOE(I)
110   CONTINUE
120   CONTINUE
      IF (NOKLMN) GO TO 170
C
C       FOLLOWING EXTRAPOLATES STATE ESTIMATES,ERROR
C       COVARIANCE MATRIX PA(NS,NS)
C
      CALL MATHUL(8,NS,NS,8,NS,NS,8,8,PA,F,TPA1,3)
      CALL MATHUL(8,NS,NS,8,NS,NS,8,8,F,TPA1,PA,1)
      CALL MATSM(PA,PA,GQGT,1,8,8,8,NS,NS,8,NS,NS,0)
      DO 130 I=1,NS
      PMDIAG (I)=PA(I,I)
130   CONTINUE
C
C       FOLLOWING COMPUTES MODIFIED-KALMAN GAIN MATRIX GAIN(NS,NG)
C
      CALL MATMUL(8,NS,NS,5,NG,NS,8,5,PA,H,PHT,3)
      CALL MATMUL(5,NG,NG,5,NG,NS,8,8,PHI,H,TPA1,1)
      CALL MATMUL(8,NG,NS,8,NS,NG,8,8,TPA1,PHT,TPA2,1)
      DO 140 I=1,NG
      TPA2(I,I)=TPA2(I,I)+1.
140   CONTINUE
      CALL MATINV(TPA2,8,NG,IVNG,RVNG)
      CALL MATMUL(8,NS,NG,8,NG,NG,8,5,PHT,TPA2,GAIN,1)
C
```

```
      C     FOLLOWING UPDATES STATE ESTIMATE
            CALL MATMUL (8,NS,NG,5,NG,1,8,1,GAIN,LAMDA,ES,1)
            CALL MATSM (XS,XS1,ES,1,8,1,8,NS,1,8,NS,1,0)
      C
      C     FOLLOWING UPDATES STATE ESTIMATE ,ERROR COVARIANCE MATRIX
      C     P(NS,NS)
      C
            CALL MATMUL(8,NS,NG,8,NG,NS,8,8,GAIN,TPA1,TPA2,1)
            DO 150 I=1,NS
            TPA2(I,I)=TPA2(I,I)-1.
        150 CONTINUE
            CALL MATMUL(8,NS,NS,8,NS,NS,8,8,PA,TPA2,TPA1,3)
            CALL MATMUL(8,NS,NS,8,NS,NS,8,8,TPA2,TPA1,PA,1)
            CALL MATMUL(5,NG,NG,8,NS,NG,8,8,PHI,GAIN,TPA1,3)
            CALL MATMUL(8,NS,NG,8,NG,NS,8,8,GAIN,TPA1,TPA2,1)
            CALL MATSM(PA,PA,TPA2,1,8,8,8,NS,NS,8,NS,NS,0)
            DO 160 I=1,NS
            PPDIAG (I)=PA(I,I)
        160 CONTINUE
        170 CONTINUE
            IF (NS.GE.7) XS(7)=PVALUE(XS(7),PI,0.)
            RETURN
            END
```

410008 CM STORAGE USED 1.082 SECONDS

```
      BLOCK DATA ORVRID
      COMMON /IDDATA/IDNRS(6)
      DATA IDNRS(4)/2/
      END
```

41000B CM STORAGE USED .014 SECONDS

```
      SUBROUTINE RCVR
      REAL LAMDA,MJM2
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      COMMON/MLS000/X(8)
      DIMENSION TPA1(8,8),PHT(8,5),TPA2(8,8),IVNG(5)
      DIMENSION RVNG(5),GAIN(8,5)
C
      IF (K .GT. 0) GO TO 40
C FOLG IS INITIALIZATION
      PI2=2.*PI
      SS2=.5/(SIGMA**2)
      CALL PHILM
      NS=NSO(IR)
      NG=NGO(IR)
      NGMIN=2
      NGMAX=NGO(IR)
      GQGT33=.01*DELT
      GQGT(3,3)=GQGT33
      GQGT(6,6)=GQGT33
      GQGT(8,8)=.04/(DELT**2)
      DO 20 I=1,NS
      DO 10 J=1,NS
      PA(I,J)=0.
   10 CONTINUE
   20 CONTINUE
      CALL CONTRL(3)
      BB=2.4/BRCVR
      PDCRIT=PI*BB/8.
      CC=PI*PDCRIT
      IF(IR.GT.0) GO TO 30
      NS=NSO(4)
      NG=NGO(4)
   30 CONTINUE
      NGM=NGMIN
      IF(NOAC) NGM=NG
      KALMAN=.NOT. NOKLMN
      LOE=.NOT. NOLOE
      CALL CONTRL(4)
      RETURN
   40 CONTINUE
      IF(K.GT.1) GO TO 50
C
C DIAGNOSTIC OUTPUT OF INPUT DATA FOR K=1 GOES HERE
C
      GO TO 90
   50 CONTINUE
```

```
C FOLG EXTRAPOLATES STATE ESTIMATE, AS REQ,D
      IF(TETHRD) GO TO 80
      DO 70 I=1,NS
      XS1(I)=0.
      DO 60 J=1,NS
      XS1(I)=XS1(I)+F(I,J)*XS(J)
60    CONTINUE
70    CONTINUE
80    CONTINUE
90    CONTINUE
C
      GQGT11=AMAX1(.25,.01*(XS1(1)**2))
      GQGT(1,1)=GQGT11
      GQGT(4,4)=GQGT11
C
C FOLLOWING COMPUTES VECTORS Q(JM),HW(JM),LAMDA(NG),AND MATRICES DT(JM,
C NG),PHI(NG,NS) ALSO SQUARED AMPLITUDE ENVELOPE VECTOR U(JM),AND
C INOVATION;S PROCESS VECTOR W(JM)
C
      CALL PHILM
      IF (NOLOE) GO TO 120
C
C FOLG COMPUTES RMAT=PHI-INVERSE AND THE LOE
      CALL MATINV(RMAT,5,NG,IVNG,RVNG)
      DO 100 I=1,NG
      RDIAG (I)=RMAT (I,I)
100   CONTINUE
      CALL MATMUL (5,NG,NG,5,NG,1,5,1,RMAT,LAMDA,GAMAES,1)
      CALL MATMUL (5,NG,NS,5,NG,1,8,1,H,GAMAES,ESLOE,2)
      DO 110 I=1,NS
      XSLOE(I)=XS1(I)+ESLOE(I)
      IF (NOKLMN) XS(I)=XSLOE(I)
110   CONTINUE
120   CONTINUE
      IF (NOKLMN) GO TO 170
C
C      FOLLOWING EXTRAPOLATES STATE ESTIMATES,ERROR
C      COVARIANCE MATRIX PA(NS,NS)
C
      CALL MATMUL(8,NS,NS,8,NS,NS,8,8,PA,F,TPA1,3)
      CALL MATMUL(8,NS,NS,8,NS,NS,8,8,F,TPA1,PA,1)
      CALL MATSM(PA,PA,GQGT,1,8,8,8,NS,NS,8,NS,NS,0)
      DO 130 I=1,NS
      PMDIAG (I)=PA(I,I)
130   CONTINUE
C
C      FOLLOWING COMPUTES MODIFIED-KALMAN GAIN MATRIX GAIN(NS,NG)
C
      CALL MATMUL(8,NS,NS,5,NG,NS,8,5,PA,H,PHT,3)
      CALL MATMUL(5,NG,NG,5,NG,NS,8,8,PHI,H,TPA1,1)
      CALL MATMUL(8,NG,NS,8,NS,NG,8,8,TPA1,PHT,TPA2,1)
      DO 140 I=1,NG
      TPA2(I,I)=TPA2(I,I)+1.
140   CONTINUE
      CALL MATINV(TPA2,8,NG,IVNG,RVNG)
      CALL MATMUL(8,NS,NG,8,NG,NG,8,5,PHT,TPA2,GAIN,1)
C
```

```
C      FOLLOWING UPDATES STATE ESTIMATE
      CALL MATMUL (8,NS,NG,5,NG,1,8,1,GAIN,LAMDA,ES,1)
      CALL MATSM (XS,XS1,ES,1,8,1,8,NS,1,8,NS,1,0)
C
C      FOLLOWING UPDATES STATE ESTIMATE ,ERROR COVARIANCE MATRIX
C      P(NS,NS)
C
      CALL MATMUL(8,NS,NG,8,NG,NS,8,8,GAIN,TPA1,TPA2,1)
      DO 150 I=1,NS
      TPA2(I,I)=TPA2(I,I)-1.
  150 CONTINUE
      CALL MATMUL(8,NS,NS,8,NS,NS,8,8,PA,TPA2,TPA1,3)
      CALL MATMUL(8,NS,NS,8,NS,NS,8,8,TPA2,TPA1,PA,1)
      CALL MATMUL(5,NG,NG,8,NS,NG,8,8,PHI,GAIN,TPA1,3)
      CALL MATMUL(8,NS,NG,8,NG,NS,8,8,GAIN,TPA1,TPA2,1)
      CALL MATSM(PA,PA,TPA2,1,8,8,8,NS,NS,8,NS,NS,0)
      DO 160 I=1,NS
      PPDIAG (I)=PA(I,I)
  160 CONTINUE
  170 CONTINUE
      XS(1)=ABS(XS(1))
      AN=(THAMAX+THAMIN)/2.
      XS(2)=SATU(XS(2)-AN,(THAMAX-THAMIN)/2.)+AN
      IF(NS.GE.3) XS(3)=SATU(XS(3),1.)
      IF(NS.GE.4) XS(4)=SATU(ABS(XS(4)),2*XS(1))
      MJM2=FLOAT(-(JM/2+1))
      THWW2=MJM2*OMEGA/(2.*FSAMP)
      AMX=XS(2)+THWW2
      AMN=XS(2)-THWW2
      AN2=(AMX+AMN)/2.
      IF(NS.GE.5) XS(5)=SATU(XS(5)-AN2,(AMX-AMN)/2.)+AN2
      IF(NS.GE.6) XS(6)=SATU(XS(6),1.)
      IF (NS.GE.7) XS(7)=PVALUE(XS(7),PI,0.)
      IF(NS.GE.8) XS(8)=PVALUE(XS(8),PI/DELT,0.)
      RETURN
      END
```

41000B CM STORAGE USED 1.381 SECONDS

A-68

```
      BLOCK DATA ORVRID
      COMMON /IDDATA/IDNRS(6)
      DATA IDNRS(4)/2/
      END
```

41000B CM STORAGE USED .015 SECONDS

A-69

```
      SUBROUTINE PHILM
C
CTHIS SUBOPTIMAL VERSION OF PHILM IS
C  BASED ON A 4-ELEMENT PARAMETER VECTOR
CA AND PROVIDES A CRUDE SEARCH AND ACQUISITION
CA FUNCTION ACCESSED AS FOLLOWS
CASEARCH MODE:NGM=NGMIN
CAACQUISITION MODE:NGMIN.LT.NGM.LT.NGMAX
CAFULL TRACK MODE:NGM=NGMAX
C
C
C  THIS NEEDS NGMIN.LE.NGM (.LE.NG) .LE.NGMAX
      REAL LAMDA
      LOGICAL NOKLMN,NDLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR10/ALFA,THES,THESDT,ALFAR,THRS,THRSDT,B,WSC,XS(8)
      DIMENSION PH(4,4),RL(4)
      EQUIVALENCE (RL1,RL(1)),(RL2,RL(2)),(RL3,RL(3)),(RL4,RL(4))
      EQUIVALENCE (PH11,PH(1,1)),(PH21,PH(2,1)),(PH22,PH(2,2)),
     *  (PH31,PH(3,1)),(PH32,PH(3,2)),(PH33,PH(3,3)),
     *  (PH41,PH(4,1)),(PH42,PH(4,2)),(PH43,PH(4,3)),(PH44,PH(4,4))
C
C
C  FOLLOWING EFFECTS INITIALIEATION
C
      IF(K.GT.0) GO TO 10
      JM2=JM/2
      JM1=JM+1
      SS2=.5/(SIGMA**2)
      IR=3
      RETURN
10    CONTINUE
CA
CA    FOLLOWING PROGRAMS THE SEARCH
CA
      IF(NGM.GT.NGMIN) GO TO 30
      ALFAR=ALFA*RHOMAX
      THRS=THES-DTHO
      THRSDT=TDRO
      B=BO
      WSC=WSCO
30    CONTINUE
C
C  FOLLOWING COMPUTES CONSTANTS FOR PRESENT SCAN
C
      AL2=2.*ALFA
      AL22=AL2*ALFA
      AR2=2.*ALFAR
      AR22=AR2*ALFAR
      AA2=AL2*ALFAR
      NGDEL=NGM-NGMAX
```

```
      DO 35 I=1,NG
      RL(I)=0.
      DO 34 L=1,I
34    PH(I,L)=0.
35    CONTINUE
C
C  FOLLOWING INITIATES LOOP AND COMPUTES FUNCTIONS
C  FOR EACH J
C
      DO 60 JI=1,JM2
      THAJ=THA(T(JI))
      THEE=THES-THAJ
      PJ=P(THEE)
      PDJ=PDOT(THEE)
      P2J=PJ**2
      THER=THRS-THAJ
      PRJ=P(THER)
      PDRJ=PDOT(THER)
      PR2J=PRJ**2
      PPRJ=PJ*PRJ
      QAJ=.5*AL22*P2J
      BA=0.
      DAJ1=AL2*P2J
      DAJ2=AL22*PJ*PDJ
      DBJ1=0.
      DBJ2=0.
      IF (NGDEL .LT. 0) GO TO 40
      QAJ=QAJ+.5*AR22*PR2J
      QBJ=AA2*PPRJ
      BA=SIGN(0.9899999,QBJ)
      IF (QAJ.GE.1.E-38) BA=SIGN(AMIN1(0.9899999,ABS(QBJ)/QAJ),QBJ)
      DBJ1=AR2*PPRJ
      DBJ2=AA2*PDJ*PRJ
      DAJ3=AR2*PR2J
      DBJ3=AL2*PPRJ
      DAJ4=AR22*PDRJ*PRJ
      DBJ4=AA2*PJ*PDRJ
40    CONTINUE
      QAJ=AMIN1(0.9999999E8,QAJ)
      QAJ=AMAX1(0.1,QAJ)
      U1=SS2*V(JI)**2
      U2=SS2*V(JM1-JI)**2
      CALL WAWBJ (WA1,WB1,QAJ,BA,U1)
      CALL WAWBJ (WA2,WB2,QAJ,BA,U2)
      WA=WA1+WA2
      WB=WB1+WB2
      CALL SWFCNS (SA,SB,RAB,QAJ,BA)
      RAB1=RAB+1.
      RL1=RL1+DAJ1*WA+DBJ1*WB
      RL2=RL2+DAJ2*WA+DBJ2*WB
      SDA1=DAJ1*SA
      SDA2=DAJ2*SA
      SDB1=DBJ1*SB
      SDB2=DBJ2*SB
      DSD1=SDA1-SDB1
      DSD2=SDA2-SDB2
      PH11=PH11+DSD1**2+RAB1*2.*SDA1*SDB1
```

```
      PH21=PH21+DSD1*DSD2+RAB1*(SDA2*SDB1+SDA1*SDB2)
      PH22=PH22+DSD2**2+RAB1*2.*SDA2*SDB2
      IF (NGDEL .LT. 0) GO TO 50
      RL3=RL3+DAJ3*WA+DBJ3*WB
      RL4=RL4+DAJ4*WA+DBJ4*WB
      SDA3=DAJ3*SA
      SDA4=DAJ4*SA
      SDB3=DBJ3*SB
      SDB4=DBJ4*SB
      DSD3=SDA3-SDB3
      DSD4=SDA4-SDB4
      PH31=PH31+DSD1*DSD3+RAB1*(SDA3*SDB1+SDA1*SDB3)
      PH32=PH32+DSD2*DSD3+RAB1*(SDA3*SDB2+SDA2*SDB3)
      PH33=PH33+DSD3**2+RAB1*2.*SDA3*SDB3
      PH41=PH41+DSD1*DSD4+RAB1*(SDA4*SDB1+SDA1*SDB4)
      PH42=PH42+DSD2*DSD4+RAB1*(SDA4*SDB2+SDA2*SDB4)
      PH43=PH43+DSD3*DSD4+RAB1*(SDA4*SDB3+SDA3*SDB4)
      PH44=PH44+DSD4**2+RAB1*2.*SDA4*SDB4
   50 CONTINUE
   60 CONTINUE
C
C  FOLLOWING COMPUTES VECTOR LAMDA(NG) AND MATRICE PHI(NG,NG)
C
      DO 110 I=1,NG
      LAMDA(I)=RL(I)
      DO 100 L=1,I
      PHILI=2.*PH(I,L)
      PHI(L,I)=PHILI
      PHI(I,L)=PHILI
      IF (NOLOE) GO TO 90
      RMAT(L,I)=PHILI
      RMAT(I,L)=PHILI
   90 CONTINUE
  100 CONTINUE
  110 CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED 1.381 SECONDS

```
C
C
C
C
C
C
      SUBROUTINE WAWBJ(WA,WB,QA,B,U)
CTHIS COMPUTES A SAMPLE EACH OF THE PROCESSES WA % WB,
CGIVEN SAMPLES OF QA,B, AND U
      LOGICAL NFIRST
      DIMENSION HA(11),HB(11),COSB(11),G(11),H(11)
      DATA LMAX/10/,NFIRST/.FALSE./
C
      IF(NFIRST) GO TO 10
      LMAX1=LMAX+1
      RLMAX1=LMAX1
      ALNLMX=ALOG(RLMAX1)
      BETA2=3.1415927/LMAX
      DO 5 L=1,LMAX1
      COSB(L)=COS((L-1)*BETA2)
    5 CONTINUE
      NFIRST=.TRUE.
   10 CONTINUE
      GMAX=-1.E+322
      HMAX=1.
      HAMAX=1.
      HBMAX=1.
      DO 20 L=1,LMAX1
      COSBL=COSB(L)
C BCBL1=AMAX1(1.+B*COSBL,0.)
      BCBL1=1.+B*COSBL
      ABC=QA*BCBL1
      UABC=U*ABC
      RAD=2.*SQRT(UABC)
      RAD1=SQRT(1.+UABC)
      GL=2.*(RAD1-1./(1.+RAD))-ABC
      G(L)=GL
      HL=1./SQRT(1.+6.2831846*RAD)
      H(L)=HL
      HAL=(-1.+U/RAD1)*HL
      HA(L)=HAL
      HBL=HAL*COSBL
      HB(L)=HBL
      GMAX=AMAX1(GL,GMAX)
      HMAX=AMAX1(HL,HMAX)
      HAMAX=AMAX1(ABS(HAL),HAMAX)
      HBMAX=AMAX1(ABS(HBL),HBMAX)
   20 CONTINUE
      GZ=GMAX-741.
      FZ=ALOG(HMAX)+GZ
      GM=FZ+ALNLMX
      GZ=GZ+ALNLMX
      GMA=GZ+ALOG(HAMAX)
      GMB=GZ+ALOG(HBMAX)
      FS=0.
      FSA=0.
      FSB=0.
```

```
                      DO 30 L=1,LMAX1
                      GL=G(L)
   60                 GLM=GL-GM
                      GLMA=GL-GMA
                      GLMB=GL-GMB
                      IF(GLM.GE.-674.) FS=FS+EXP(GLM)*H(L)
                      IF(GLMA.GE.-674.) FSA=FSA+EXP(GLMA)*HA(L)
   65                 IF(GLMB.GE.-674.) FSB=FSB+EXP(GLMB)*HB(L)
                30    CONTINUE
                      WA=(HAMAX/HMAX)*(FSA/FS)
                      WB=(HBMAX/HMAX)*(FSB/FS)
                      RETURN
   70                 END
```

41000B CM STORAGE USED .451 SECONDS

A-74

```
      C
      C
      C
      C
      C
      C
            SUBROUTINE SWFCNS(HWAJ,HWBJ,RABJ,QAV,BA)
            INTEGER FNA(4)
            LOGICAL NFIRST
            DIMENSION BULK(12,25,3),QA(25),B(25),DEX(24),DEY(11)
            DIMENSION AA(3),BB(3),CC(3),DD(3),APPVA(3)
            DIMENSION QAB(25,2),QB(2),IQBMAX(2),IQJB(2)
            EQUIVALENCE (QA(1),QAB(1,1)),(B(1),QAB(1,2))
            EQUIVALENCE (IQMAX,IQBMAX(1)),(JBMAX,IQBMAX(2))
            EQUIVALENCE (IQ,IQJB(1)),(IB,IQJB(2)),(BV,QB(2))
            DATA FNA/10HMLSLOGSWCD,1LC,2*0/
            DATA NFIRST/.FALSE./
      C
      C FOLG GENERATES TABLE FROM DATA FILE
      C
            IF(NFIRST) GO TO 10
            NFIRST=.TRUE.
            IX=IATTACH(6LTAPE12,FNA,2LID,6RM3497E)
            IF (IX.EQ.0) GO TO 5
            CALL INTIO(6HIXAT12,IX,1,1)
            STOP*INPUT FILE ATTACH (TAPE12) NOT SATISFACTORY*
        5   CONTINUE
            READ (12) IDUM,IDUM
            READ (12) IQMAX,(QA(I),I=1,IQMAX)
            READ (12) JBMAX,(B(J),J=1,JBMAX)
            READ (12)    (((BULK(J,I,K),J=1,JBMAX),I=1,IQMAX),K=1,3)
            IQM1=IQMAX-1
            DO 3 I=1,IQM1
        3   DEX(I)=QA(I+1)-QA(I)
            JBM1=JBMAX-1
            DO 6 I=1,JBM1
        6   DEY(I)=B(I+1)-B(I)
            IQ=IQMAX/2
            IB=JBMAX/2
            CALL RETURN(6LTAPE12)
       10   CONTINUE
      C
      C FOLG TRUNCATES ARGUMENTS AND HANDLES SMALL BV CASES
      C
            BV=ABS(BA)
            IF (BV .GE. .01) GO TO 20
            HWAJ=1./SQRT(1.+2.*QAV)
            HWBJ=0.
            RABJ=0.
            RETURN
       20   CONTINUE
      C
      C FOLG DETERMINES IQ,IB
            QB(1)=QAV
            DO 1500 L=1,2
            QBL=QB(L)
     1001   IF(QBL.LT.QAB(IQJB(L),L)) GO TO 1101
```

```
                  IQJBP1=IQJB(L)+1
                  IQBMMM=IQBMAX(L)
                  DO 1201 I=IQJBP1,IQBMMM
                  IF(QBL.LT.QAB(I,L)) GO TO 1202
            1201  CONTINUE
            1202  IQJB(L)=I-1
                  GO TO 1500
            1101  IQJBM1=IQJB(L)-1
                  DO 1301 I=1,IQJBM1
                  MMII=IQJBM1-I+1
                  IF(QBL.GE.QAB(MMII,L)) GO TO 1302
            1301  CONTINUE
            1302  IQJB(L)=MMII
            1500  CONTINUE
            C
            C
            C FOLG DETERMINES INTERPOLATED VALUES
            C
                  DIX=QAV-QA(IQ)
                  DIY=BV-B(IB)
                  CX=DIX/DEX(IQ)
                  CY=DIY/DEY(IB)
                  DO 301 K=1,3
                  AA(K)=BULK(IB,IQ,K)
                  BB(K)=BULK(IB+1,IQ,K)
                  CC(K)=BULK(IB,IQ+1,K)
                  DD(K)=BULK(IB+1,IQ+1,K)
                  APPVA(K)=AA(K)+(CC(K)-AA(K))*CX+(BB(K)-AA(K))*CY+CX*CY*(AA(K)
                 *+DD(K)-CC(K)-BB(K))
            301   CONTINUE
                  HWAJ=EXP(APPVA(1))
                  HWBJ=EXP(APPVA(2))
                  RABJ=APPVA(3)
                  IF(BA.LT.0.) RABJ=-RABJ
                  RETURN
                  END
```

41000B CM STORAGE USED .666 SECONDS

```
      BLOCK DATA PLSUID
      COMMON/IDDATA/IDNRS(6)
      DATA IDNRS(3)/3/
      END
```

41000B CM STORAGE USED .014 SECONDS

A-77

```
      SUBROUTINE PHILM
C
C THIS OPTIMAL VERSION OF PHILM IS FOR ALL SCALLOPING RATES
C AND PROVIDES A CRUDE SEARCH-AND-ACQUISITION FUNCTION
C ACCESSED AS FOLLOWS:
C     SEARCH MODE: NGM=NGMIN
C     ACQUISITION MODE: NGMIN.LT.NGM.LT.NGMAX
C     FULL TRACK MODE: NGM=NGMAX
C
C
C      THIS NEEDS NGMIN.LE.NGM (.LE.NG) .LE.NGMAX
C
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,ED(8),GQGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/ALFA,THES,THESDT,ALFAR,THRS,THRSDT,B,WSC,XS(8)
      DIMENSION INDEX(2),DT(130,5),HW(130),W(130)
C
C FOLLOWING EFFECTS INITIALIZATION
C
      IF(K.GT.0) GO TO 10
      JM2=JM/2
      JM1=JM+1
      SS2=.5/(SIGMA**2)
      IR=2
      RETURN
   10 CONTINUE
C
C FOLLOWING PROGRAMS THE SEARCH
C
      IF (NGM.GT.NGMIN) GO TO 30
      ALFAR=ALFA*RHOMAX
      THRSDT=TDRO
      B=BO
      WSC=WSCO
      THRS=THES-DTHO
   30 CONTINUE
C
C      FOLLOWING COMPUTES CONSTANTS FOR PRESENT SCAN
C
      AL2=2.*ALFA
      AL22=AL2*ALFA
      AR2=2.*ALFAR
      AR22=AR2*ALFAR
      AA2=AL2*ALFAR
C
C      FOLLOWING INITIATES LOOP AND COMPUTES FUNCTIONS
C      FOR EACH J
C
```

```
      DO 60 JI=1,JM2
      INDEX(1)=JI
      INDEX(2)=JM1-JI
      THAJ=THA(T(JI))
      THEE=THES-THAJ
      PJ=P(THEE)
      PDJ=PDOT(THEE)
      P2J=PJ**2
      THER=THRS-THAJ
      PRJ=P(THER)
      PDRJ=PDOT(THER)
      PR2J=PRJ**2
      PPRJ=PJ*PRJ
      QNRJ=.5*AL22*P2J
      QRJ=QNRJ+.5*AR22*PR2J
      D1NET=AL2*P2J
      C1=AR2*PPRJ
      D2NET=AL22*PJ*PDJ
      C2=AA2*PDJ*PRJ
      D3J=AR2*PR2J
      C3=AL2*PPRJ
      D4J=AR22*PDRJ*PRJ
      C4=AA2*PJ*PDRJ
      D5J=-AA2*PPRJ
C FOLG COMPLETES CALCULATIONS FOR TO,FRO SCANS, RESP.
      DO 50 I=1,2
      J=INDEX(I)
      BLOCAL=PVALUE(B+WSC*T(J),3.1415927,DELBL)
      SB=SIN(BLOCAL)
      CB=COS(BLOCAL)
      QJ=QNRJ
      QRJALL=QRJ-D5J*CB
      D1J=D1NET
      D2J=D2NET
      IF (NGM.LT.NGMAX) GO TO 40
      QJ=QRJALL
      D1J=D1J+C1*CB
      D2J=D2J+C2*CB
   40 CONTINUE
      DT(J,1)=D1J
      DT(J,2)=D2J
      DT(J,3)=D3J+C3*CB
      DT(J,4)=D4J+C4*CB
      DTJ5=D5J*SB
      DT(J,5)=DTJ5
C  FOLG IS FOR A 6D LOE DESIGN
C     DT(J,6)=DTJ5*T(J)
C FOLLOWING COMPUTES VECTOR HW(JM),AND INNOVATIONS PROCESS VECTOR W(JM)
C
      HW(J)=1./(1.+2.*QJ)
      UJ=V(J)
      UJ=SS2*(UJ**2)
      W(J)=(UJ/SQRT(1.+QJ*UJ))-1.
   50 CONTINUE
   60 CONTINUE
C
C      FOLLOWING COMPUTES VECTOR LAMDA(NG) AND MATRICE PHI(NG,NG)
```

```
C
      DO 110 I=1,NG
      LAMDA(I)=0.0
      DO 70 J=1,JM
      LAMDA(I)=LAMDA(I)+DT(J,I)*W(J)
   70 CONTINUE
      DO 100 L=1,I
      PHILI=0.
      DO 80 J=1,JM
      PHILI=PHILI+HW(J)*DT(J,I)*DT(J,L)
   80 CONTINUE
      PHI(L,I)=PHILI
      PHI(I,L)=PHILI
      IF (NOLOE) GO TO 90
      RMAT(L,I)=PHILI
      RMAT(I,L)=PHILI
   90 CONTINUE
  100 CONTINUE
  110 CONTINUE
      RETURN
      END
```

410008 CM STORAGE USED .805 SECONDS

```
      BLOCK DATA PLOPID
      COMMON /IDDATA/IDNRS(6)
      DATA IDNRS(3)/2/
      END
```

41000B CM STORAGE USED .015 SECONDS

```
      FUNCTION PMLS(THETA)
C     THIS COMPUTES THE TRANSMITTED SIGNAL INTENSITY(RELATIVE TO
C     THAT AT BEAM CENTER  AS A FUNCTION OF THE ANGLE-OFF-BORESIGHT.
C     THE MODEL GIVES FIRST SIDE LOBES 23 DB BELOW THE MAIN LOBE.
C
      LOGICAL MORE
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DATA PCRIT/.7853981635/,AA/1.570796327/
      PMLS=PCRIT
      Z=BBB*THETA
      IF (ABS(ABS(Z)-1.) .LE. 1.E-7) RETURN
      PMLS=(COS(AA*Z))/(1.-Z**2)
      RETURN
      END
```

41000B CM STORAGE USED .071 SECONDS

PMLS1
Page 1 of 2

```
      BLOCK DATA PMLSID
      COMMON/IDDATA/IDNRS(6)
      DATA IDNRS(2)/1/
      END
```

41000B CM STORAGE USED .016 SECONDS

```
      FUNCTION P(THETA)
C      THIS COMPUTES THE TRANSMITTED SIGNAL INTENSITY(RELATIVE TO
C      THAT AT BEAM CENTER  AS A FUNCTION OF THE ANGLE-OFF-BORESIGHT.
C      THE MODEL GIVES FIRST SIDE LOBES 23 DB BELOW THE MAIN LOBE.
C
      COMMON/RCVR09/BRCVR,BB,POCRIT,CC
      DATA PCRIT/.7853981635/,AA/1.570796327/
      P=PCRIT
      Z=BB*THETA
      IF (ABS(ABS(Z)-1.) .LE. 1.E-7) RETURN
      P=(COS(AA*Z))/(1.-Z**2)
      RETURN
      END
```

410008 CM STORAGE USED .068 SECONDS

```
C
      FUNCTION PDOT (THETA)
C      THIS COMPUTES THE DERIVATIVE DP(THETA)/DTHETA WHERE P(THETA)
C      IS THE TRANSMITTED SIGNAL INTENSITY(RELATIVE TO THAT AT BEAM
C      CENTER) AND THETA IS THE ANGLE-OFF-BORESIGHT.THE P-MODEL USED
C      IS THAT GIVING FIRST SIDE LOBES 23 DB BELOW THE MAIN LOBE.
C
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      DATA AA/1.570796327/
C
      PDOT=SIGN(PDCRIT,-THETA)
      Z=BB*THETA
      IF (ABS(ABS(Z)-1.) .LE. 1.E-7) RETURN
      CP=AA*(Z+1.)
      CM=AA*(Z-1.)
      PDOT=CC*((COS(CP)-SIN(CP)/CP)/CP+(COS(CM)-SIN(CM)/CM)/CM)
      RETURN
      END
```

41000B CM STORAGE USED .119 SECONDS

```
      C
            BLOCK DATA POPTID
            COMMON/IDDATA/IDNRS(6)
            DATA IDNRS(6)/1/
    5       END

 41000B CM STORAGE USED     .013 SECONDS
```

```
      SUBROUTINE RCVR
C THRESHOLD RECEIVER
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(8,8),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(8),GOGT(8,8),H(5,8),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMAES(5),RDIAG(5),PMDIAG(8)
      COMMON/RCVR06/PPDIAG(8),RMAT(5,5),PHI(5,5),PA(8,8),LAMDA(5)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(8),ESLOE(8),ES(8)
      COMMON/RCVR09/BRCVR,BB,POCRIT,CC
      COMMON/RCVR10/XS1(8),XS(8)
      DIMENSION VPK(2),VTH(2),N(2),U(130),W(130),TRG(2)
      DATA GX/100./
C
      IF (K .GT. 0) GO TO 50
C FOLG IS INITIALIZATION
      IR=1
      NS=NSO(IR)
      NG=NGO(IR)
      CALL CONTRL (3)
      OMEGA=-20000.
      JM2=JM/2
      JM22=(JM2+1)/2
      TSAMP=1./FSAMP
      JDELAY=IFIX(2.*3.1415927*25000.*TSAMP+.5)
      JM22=JM22+JDELAY
      ICOUNT=0
      XS1(3)=0.0
      XS(3)=0.0
      CALL CONTRL (4)
      RETURN
   50 CONTINUE
C
C FOLG PRODUCES TG1 AND TG2
      TPST=(XS1(2)-THAMAX)/OMEGA
      TPSF=TF+((THAMIN-XS1(2))/OMEGA
C
C FOLG PRODUCES LOG ENVELOPE SIGNAL FILTERED TO 25 KHZ
      DO 70 L=1,2
      VPEAK=0.0
      DO 60 J=1,JM2
      I1=J+(L-1)*JM2
      JM21=JM2+1
      U(I1)=20.0*ALOG10(1.+GX*V(I1))
      IF((I1.EQ.1).OR.(I1.EQ.JM21)) FL25(4)=0.
      CALL DFLTR1(U(I1),W(I1),FL25)
      VPEAK=AMAX1(VPEAK,W(I1))
   60 CONTINUE
      VPK(L)=VPEAK
      VTH(L)=VPEAK-3.
   70 CONTINUE
      IF(K.NE.1) GO TO 80
      NABORT=0
```

```
      NNOM=1+IFIX(2.*5.0E-5/TSAMP+.5)
      N(1)=NNOM
      N(2)=NNOM
   80 CONTINUE
      DO 140 I=1,2
      JJ=JM22-N(I)/2
      N2=(N(I)+1)/2
      DJ=(N2-1)*TSAMP
      TG1=TPST-DJ
      TG2=TPSF-DJ
      TAUSTT=TG1
      IF(I.EQ.2) TAUSTT=TG2
      VT=VTH(I)
      J0=(I-1)*JM2+JJ
      J1=J0+N(I)-2
      L=1
      ISIGN=1
      DO 100 J=J0,J1
      V0=W(J)
      V1=W(J+1)
      TAU=TAUSTT+(J-J0)*TSAMP
      DELTAV=(V1-VT)*ISIGN
      IF(DELTAV.LT.0.0) GO TO 100
      TDWELL=TAU+TSAMP*((VT-V0)/(V1-V0))
      IF(J.NE.J0) GO TO 90
      IF(V0.GT.VT) TDWELL=TAU
   90 IF(L.EQ.2) GO TO 130
      L=2
      ISIGN=-1
      TDWEL1=TDWELL
  100 CONTINUE
      IF(L.EQ.1) GO TO 190
      TDWELL=TAU+TSAMP
  130 CONTINUE
      GWIDTH=TDWELL-TDWEL1
      IF((GWIDTH.LT.15.E-6).OR.(GWIDTH.GT.350.E-6)) GO TO 190
      TRG(I)=2.*GWIDTH
      TC=(TDWELL+TDWEL1)/2.
      IF(I.EQ.2) GO TO 150
      TC1=TC
  140 CONTINUE
  150 DELTET=TC-TC1
      DDD=(.5E-6)*IFIX(DELTET/.5E-6+.5)
      XS1(2)=(-OMEGA/2.)*(DDD-TR)
      XS(2)=XS1(2)
      XS1(1)=(VPK(1)+VPK(2))/2.
      XS1(1)=(10.**(XS1(1)/20.)-1.)/GX
      XS(1)=XS1(1)
      DO 160 I=1,2
      RG=TRG(I)/TSAMP
      N(I)=1+2*IFIX(RG/2.+.5)
  160 CONTINUE
      NABORT=0
      RETURN
  190 ICOUNT=ICOUNT+1
      NABORT=NABORT+1
      IF(NABORT.LT.5) RETURN
```

```
 115          N(1)=NNOM
              N(2)=NNOM
              XS(2)=XS1(2)
              RETURN
              END
```

41000B CM STORAGE USED .822 SECONDS

```
      BLOCK DATA TRVRID
      COMMON/IDDATA/IDNRS(6)
      DATA IDNRS(3),IDNRS(4),IDNRS(6) /1,1,0/
      END
```

41000B CM STORAGE USED .023 SECONDS

```
      PROGRAM ACQMP1(INPUT,OUTPUT,TAPE10=INPUT,TAPE15,TAPE7=OUTPUT,
     *TAPE20)
      INTEGER DATIN(4)
      DIMENSION ERALFA(100),ERTHTA(100),ERALFR(100),ERTHTR(100)
      DIMENSION X4TOX1(100),IDENTS(6),LABELS(5),NAME(10),NAME1(10)
      DIMENSION NAME2(10),NAMTIM(4),TIM(102),TOTAL(100,5),YERROR(5)
      DIMENSION YY(102)
      EQUIVALENCE (RNAME2,NAME2(10)),(RNAME,NAME(10))
      EQUIVALENCE (TOTAL(1,1),ERALFA(1)),(TOTAL(1,2),ERTHTA(1)),
     *(TOTAL(1,3),ERALFR(1)),(TOTAL(1,4),ERTHTR(1)),
     *(TOTAL(1,5),X4TOX1(1))
      DATA NAME1/10HSIM. JOB: ,3*1H ,10H FILE NO: ,3*1H ,10H    DATE: ,
     *1H /
      DATA NAME2/10HPLOT JOB: ,3*1H ,10H PROGRAM: ,10HACQMP1    ,2*1H ,
     *10H    DATE: ,1H /
      DATA ALX/10./,ALY/2./,YERROR/14.,11.,8.,5.,2./,S/.12/,H/.07/
      DATA NAMTIM/10HTIME SINCE,10H START OF ,10HFIRST SCAN,10H (SECONDS
     *)/,IBLANK/2H  /
      DATA DATIN/10HMLSSIMDATA,3*0/,YMMAX/1./
      DATA LABELS/10HES ALFA ER,10HES THTA ER,10HES ALFR ER,
     *10HES THTR ER,10HALFAR/ALFA/
      CALL CALCOMP(20)
      CALL FACTOR(1.)
      CALL PLOT(+2.0,1.5,-3)
C
C PUT JOBNAME AND DATE IN APPROPRIATE ARRAY ELEMENTS.
C
      NAME2(2)=JOBNAME(A)
      RNAME2=DATE(A)
C
C READ NUMBER OF FILES TO BE PLOTTED.
C
      READ(10,10) NPLOTS
   10 FORMAT(I2)
C
C WRITE JOBNAME, DATE, AND NUMBER OF PLOTS TO BE MADE.
C
      WRITE(7,20) NAME2(2),RNAME2,NPLOTS
   20 FORMAT(10H1JOBNAME: ,A10,15X,6HDATE: ,A10//1X,I2,20H FILES TO BE P
     *LOTTED)
C
C******************300: LOOP TO PLOT EACH FILE.
C
      DO 300 N=1,NPLOTS
C
C*******************************************
C*   READ DATA FROM DISK, WRITE ON LPR.   *
C*******************************************
C
C READ CURRENT FILE NAME;WRITE PLOT NUMBER AND FILE NAME.
C
      READ(10,30) DATIN(2)
   30 FORMAT(A10)
      WRITE(7,40) (DATIN(I),I=1,2),N,NPLOTS
   40 FORMAT (13H1INPUT FILE: ,2A10,10X,5HPLOT ,I2,4H OF ,I2/)
      IR=IATTACH(6LTAPE15,DATIN)
      IF(IR.EQ.0) GO TO 60
```

```
      WRITE (7,50)
50    FORMAT (20H FILE DID NOT ATTACH/)
      STOP*INPUT FILE ATTACH NOT SATISFACTORY*
C
C READ DATA FROM DISK AND PLACE IN PROPER LOCATION OR ARRAY.
C
60    READ(15) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      READ(15) (IDENTS(I),I=1,6),KSTART
      READ(15)NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP
C******************70
      DO 70 K=1,KM
      READ(15) ERALFA(K),ERTHTA(K),ERALFR(K),ERTHTR(K),X4TOX1(K)
70    CONTINUE
      READ(15) RHOMAX,DTHO,TDRO,BETAO,FSCO,EBETAO,EBETAM,EFSCO,EFSCM
      CALL RETURN(6LTAPE15)
      DECODE (10,75,DOUT) NN1,IRCVR,NN2,NN3,NN4
75    FORMAT(I2,I1,I3,A1,I3)
C
C WRITE DATA READ FROM DISK ON THE LINEPRINTER.
C
      WRITE(7,80) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
80    FORMAT (1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,90) IDENTS,KSTART
90    FORMAT (1H ,6(1X,A8,9X),3X,I3/)
      WRITE(7,100)  NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR,THESEP
100   FORMAT(1H ,2X,I2,4X,7(2X,G13.6)/)
C
C WRITE FIRST 35 VALUES, 5 PERIODS, THEN THE LAST 3.
C
C******************120
      DO 120 K=1,35
      WRITE(7,110)(TOTAL(K,L),L=1,5)
110   FORMAT(1H ,5(G13.6,5X))
120   CONTINUE
C******************140
      DO 140 K=1,5
      WRITE(7,130)
130   FORMAT(1H ,5(6X,1H.,11X))
140   CONTINUE
      KM3=KM-3
C******************150
      DO 150 K=KM3,KM
      WRITE(7,110) (TOTAL(K,L),L=1,5)
150   CONTINUE
      WRITE(7,151)RHOMAX,DTHO,TDRO,BETAO,FSCO,EBETAO,EBETAM,EFSCO,EFSCM
151   FORMAT(1H0,9(G12.6,2X))
C
C******************************
C*   PLOTTING BEGINS HERE.   *
C******************************
C
C DRAW LEFT EDGE OF THE PAGE
C
      CALL PLOT(-2.0,0.0,3)
      CALL PLOT(-2.0,7.0,2)
C
C WRITE FILE AND PLOT INFORMATION AT TOP OF PAGE.
```

```
      C
            XH=-.5
            YH=8.7-S
            CALL SYMBOL(XH,YH,H,NAME1,0.0,90)
      C******************160
            DO 160 I=1,10
            NAME(I)=IBLANK
      160   CONTINUE
            NAME(2)=JBNAM
            DECODE(10,170,DOUT) NAME(6)
      170   FORMAT(A10)
            RNAME=TODAY
            CALL SYMBOL(XH,YH,H,NAME,0.0,100)
            YH=YH-2.*S
            CALL SYMBOL(XH,YH,H,NAME2,0.0,100)
      C
      C  WRITE PLOT INFORMATION ON BOTTOM OF PAGE
      C
      200   ENCODE(81,210,NAME) (IDENTS(I),I=1,2),BMLS,DELT,KM,KSTART
      210   FORMAT(11HSCENARIO:  ,A8,1H,,A6,7H, BMLS=,F4.1,11H DEG, DELT=,
           *F9.7,9H SEC, KM=,I3,9H, KSTART=,I3)
            YH=S
            CALL SYMBOL(XH,YH,H,NAME,0.0,81)
      220   ENCODE(81,230,NAME) DSNRDB,RHO,BETA,FSC,THESEP
      230   FORMAT(9H     S/N=,F5.1,9H DB, RHO=,F5.1,7H, BETA=,F6.1,
           *10H DEG,FSC=,F8.3,12H HZ, THESEP=,F6.3,4H DEG)
            YH=0.0
            CALL SYMBOL (XH,YH,H,NAME,0.0,81)
      240   ENCODE(59,250,NAME) (IDENTS(I),I=3,6),BRCVR
      250   FORMAT(10HRECEIVER: ,A7,2H, ,A7,2H, ,A8,1H,,A6,8H, BRCVR=,
           *F4.1,4H DEG)
            YH=-S
            CALL SYMBOL (XH,YH,H,NAME,0.0,59)
            IF (IRCVR.NE.2) GO TO 258
            ENCODE(83,255,NAME) RHOMAX,DTHO,TORO,BETAO,FSCO
      255   FORMAT(39H      INTERFERENCE TRACKER IDLER VALUES:,F4.1,1H,,
           CF4.1,5H DEG,,F4.1,9H DEG/SEC,F5.1,5H DEG,,F4.1,3H HZ)
            YH=-2*S
            CALL SYMBOL (XH,YH,H,NAME,0.0,83)
            ENCODE(64,256,NAME) EBETAO,EBETAM
      256   FORMAT(34H      BETA ESTIMATE ERROR: INITIAL=,G10.3,7H,FINAL=,G10.3
           *,3HDEG)
            YH=-3.*S
            CALL SYMBOL (XH,YH,H,NAME,0.0,64)
            ENCODE(63,257,NAME) EFSCO,EFSCM
      257   FORMAT(34H      FSC  ESTIMATE ERROR: INITIAL=,G10.3,7H,FINAL=,G10.3
           *,2HHZ)
            YH=-4.*S
            CALL SYMBOL (XH,YH,H,NAME,0.0,63)
      258   CONTINUE
            XH=0.0
            CALL FACTOR(.5)
      C
      C FILL ARRAY WITH HORIZONTAL AXIS DATA.
      C
      C******************260
            DO 260 K=1,KM
```

```
          TIM(K)=DELT*(K-1)
    260   CONTINUE
          CALL SCALE(TIM,ALX,KM,1)
    C*******************290: LOOP FOR 5 PLOTS.
          DO 290 L=1,5
          AMAX=-1.E+320
          AMIN=1.E+320
    C*******************270
          DO 270 K=1,KM
    C
    C FIND MIN AND MAX OF ERROR DATA.
    C
          AMIN=AMIN1(TOTAL(K,L),AMIN)
          AMAX=AMAX1(TOTAL(K,L),AMAX)
    270   CONTINUE
          YMAX=AMAX1(ABS(AMAX),ABS(AMIN))
          YMIN=-YMAX
          IF(L .EQ. 5) YMIN=0.
          YINC=(YMAX-YMIN)/ALY
    C
    C FILL ARRAY WITH ERROR DATA INCLUDING YMIN AND YINC.
    C
    C*******************280
          DO 280 K=1,KM
          YY(K)=TOTAL(K,L)
    280   CONTINUE
          YY(KM+1)=YMIN
          YY(KM+2)=YINC
    C
    C DRAW Y-AXIS AND WRITE LABEL.
    C
          YH=YERROR(L)
          YHM=-YH
          CALL AXIS(XH,YH,LABELS(L),10,ALY,90.0,YMIN,YINC)
    C
    C MAKE PLOT OF ERROR DATA.
    C
          CALL PLOT(0.0,YH,-3)
          CALL LINE(TIM,YY,KM,1,0,0)
          CALL PLOT(0.0,YHM,-3)
    290   CONTINUE
    C
    C MAKE HORIZONTAL AXIS FOR ALL 5 PLOTS.
    C
          CALL AXIS(XH,YH,NAMTIM,-40,ALX,0.0,TIM(KM+1),TIM(KM+2))
    C
    C 5 PLOTS FOR THE FILE ARE NOW COMPLETED.
    C
          CALL FACTOR(1.)
    C
    C RESET ORIGIN AT  8.5 INCHES ALONG X-AXIS
    C
          CALL PLOT(8.5,0.0,-3)
    300   CONTINUE
    C
    C DRAW RIGHT EDGE OF FINAL PLOT.
    C
```

```
          CALL PLOT(-2.0,0.0,3)
  230     CALL PLOT(-2.0,7.0,2)
          CALL ENDPLT
          STOP
          END
```

42000B CM STORAGE USED 2.817 SECONDS

```
                  SUBROUTINE CLIP(A,KM,AMAX,AMIN,ISW,KC)
                  DIMENSION A(KM)
                  LOGICAL LA,LE
            C
                  KC=KM
                  LA=.FALSE.
                  LE=.FALSE.
                  ISW=0
                  DO 100 KK=1,KM
                  K=KM+1-KK
                  IF(A(K).GE.AMIN) GO TO 10
                  LE=.TRUE.
                  A(K)=AMIN
                  IF(ISW.EQ.0) ISW=-K
                  GO TO 20
               10 IF(LA) GO TO 20
                  IF((ISW.NE.0).AND.(KC.EQ.KM)) KC=K
               20 IF(A(K).LE.AMAX) GO TO 30
                  A(K)=AMAX
                  LA=.TRUE.
                  IF(ISW.EQ.0) ISW=K
                  GO TO 100
               30 IF(LE) GO TO 100
                  IF((ISW.NE.0).AND.(KC.EQ.KM)) KC=K
              100 CONTINUE
                  END
```

41000B CM STORAGE USED .171 SECONDS

```
      FUNCTION GAUSS(DUMMY)
C
C     GAUSE PRODUCES INDEPENDENT PSEUDO-RANDOM NUMBERS DISTRIBUTED
C     NORMALLY (0,1) BY THE DIRECT METHOD DESCRIBED IN THE FOLLOWING
C     REFERENCE:
C     M.ABRAMOWITZ,I.A.STEGUN,HANDBOOK OF MATHEMATICAL FUNCTIONS
C     APPLIED MATHEMATIC SERIES#55,NATIONAL BUREAU OF STANDARD,U.S
C     DEPT OF COMMERCE,NOV.1970,PAGE.953.
C     THIS VERSION OF GAUSS TAKES INDEPENDENT PSEUDO-RANDOM NUMBERS
C     UNIFORMALLY DISRIBUTED ON (0,1).FROM THE CDC-SUPPLIED FUNCTION
C     SUBPROGRAM RANF(X).
      LOGICAL HAS
      DATA HAS/.FALSE./
      IF(DUMMY.EQ.0.0) GO TO 30
      CALL RANSET(0)
      HAS=.FALSE.
   30 CONTINUE
      IF(HAS) GO TO 40
      X=SQRT(-2.*ALOG(RANF(1.0)))
      T=6.283185308*RANF(1.)
      SAVED=X*SIN(T)
      GAUSS=X*COS(T)
      HAS=.TRUE.
      RETURN
   40 GAUSS=SAVED
      HAS=.FALSE.
      RETURN
      END
```

41000B CM STORAGE USED .115 SECONDS

```
      SUBROUTINE INTIO (SYMBOL,IVAL,IVALMX,IZ)
      INTEGER SWITCH,SLASH,SYMBOL
      DATA SLASH/1H//
      IF(IZ)1,2,3
 1    READ(10,10) IVAL
 10   FORMAT(I6)
      WRITE(7,20) IVAL
 20   FORMAT(1H ,I6/)
      GO TO 80
 2    WRITE(7,30) SYMBOL,IVAL
 30   FORMAT(1H ,A6,3H = ,I6)
      READ(10,40) SWITCH,IVAL2
 40   FORMAT(A1,I6)
      IF (SWITCH .EQ. SLASH) GO TO 60
      WRITE(7,50)
 50   FORMAT(1H )
      RETURN
 60   IVAL=IVAL2
      WRITE(7,70) SLASH,IVAL
 70   FORMAT(1H+,15X,A1,I6/)
 80   IF (IVAL .LE. IVALMX) RETURN
      WRITE (7,90) SYMBOL,IVAL,IVALMX,SYMBOL,IVALMX
 90   FORMAT (1H ,A6,2H (,I6,16H) > MAX. VALUE (,I6,1H)/
     +1H ,A6,8H SET TO ,I6/)
      IVAL=IVALMX
      RETURN
 3    WRITE(7,100) SYMBOL,IVAL
 100  FORMAT(1H ,A6,3H = ,I6/)
      RETURN
      END
```

41000B CM STORAGE USED .131 SECONDS

```
      SUBROUTINE LOGIO (SYMBOL,VALUE,IZ)
      LOGICAL VALUE,VAL2
      INTEGER SWITCH,SLASH,SYMBOL
      DATA SLASH/1H//
      IF (IZ) 1,2,3
    1 READ (10,10) VALUE
   10 FORMAT (L5)
      WRITE(7,20) VALUE
   20 FORMAT(1H ,L5/)
      RETURN
    2 WRITE (7,30) SYMBOL,VALUE
   30 FORMAT(1H ,A6,3H = ,L5)
      READ (10,40) SWITCH,VAL2
   40 FORMAT (A1,L5)
      IF (SWITCH .EQ. SLASH) GO TO 60
      WRITE(7,50)
   50 FORMAT(1H )
      RETURN
   60 VALUE=VAL2
      WRITE(7,70) SLASH,VALUE
   70 FORMAT(1H+,14X,A1,L5/)
      RETURN
    3 WRITE (7,80) SYMBOL,VALUE
   80 FORMAT (1H ,A6,3H = ,L5/)
      RETURN
      END
```

41000B CM STORAGE USED .104 SECONDS

A-99

```
      SUBROUTINE MATIN      73/172  TS                         FTN 4.6+452       05/18/79  13.53.08        PAGE     1

                       SUBROUTINE MATIN (SYMBOL,A,M1,N1,M)
                 C  THIS INPUTS A MATRIX
                 C  PARAMETERS AS FOLLOWS
                 C  SYMBOL IS A 6 CHARACTER ALPHANUMERIC,, EG. *     A*
                 C  A IS THE ARRAY NAME
                 C  M1 AND N1 ARE THE DIMENSIONS OF THE DESIRED MATRIX
                 C  M IS THE COLUMN LENGTH OF THE ARRAY STORAGE
                       DIMENSION A(M,N1)
                       INTEGER SYMBOL
                       WRITE (7,5) SYMBOL,M1,N1
                 5     FORMAT(14HOINPUT MATRIX ,A6,2H (,I2,3H X ,I2,10H) BY ROWS:)
                       DO 30 I=1,M1
                       READ (10,10) (A(I,J),J=1,N1)
                 10    FORMAT(10(G13.6))
                       WRITE(7,20) (A(I,J),J=1,N1)
                 20    FORMAT(1H ,10(G13.6))
                 30    CONTINUE
                       RETURN
                       END

  41000B CM STORAGE USED          .118 SECONDS
```

```
      SUBROUTINE MATMUL(NRA,N2RA,NCA,NRB,N2RB,NCB,NRC,
     *NCC,A,B,C,L)
      DIMENSION A(NRA,NCA),B(NRB,NCB),C(NRC,NCC)
      GO TO (101,102,103,104),L
  101 IF(NCA.NE.N2RB) GO TO 1000
      DO 200 I=1,N2RA
      DO 200 J=1,NCB
      C(I,J)=0.
      DO 200 K=1,NCA
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
  200 CONTINUE
      RETURN
  102 IF(N2RA.NE.N2RB) GO TO 1000
      DO 400 I=1,NCA
      DO 400 J=1,NCB
      C(I,J)=0.
      DO 400 K=1,N2RA
      C(I,J)=C(I,J)+A(K,I)*B(K,J)
  400 CONTINUE
      RETURN
  103 IF(NCA.NE.NCB) GO TO 1000
      DO 600 I=1,N2RA
      DO 600 J=1,N2RB
      C(I,J)=0.
      DO 600 K=1,NCA
      C(I,J)=C(I,J)+A(I,K)*B(J,K)
  600 CONTINUE
      RETURN
  104 IF(N2RA.NE.NCB) GO TO 1000
      DO 800 I=1,NCA
      DO 800 J=1,N2RB
      C(I,J)=0.
      DO 800 K=1,N2RA
      C(I,J)=C(I,J)+A(K,I)*B(J,K)
  800 CONTINUE
      RETURN
 1000 WRITE(7,2000)
 2000 FORMAT(5X,*CHECK THE PROGRAM FOR MISTAKES*)
      RETURN
      END
```

41000B CM STORAGE USED .427 SECONDS

```
      SUBROUTINE MATOUT(SYMBOL,A,M1,N1,M,IZ)
CTHE VARIABLES IN SUBROUTINE ≠MATOUT≠ ARE AS FOLLOWS
C≠SYMBOL≠ IS A 6-CHARACTER ALPHANUMERIC EG.*     A*
C≠A≠ IS THE ARRAY NAME
C≠M1≠,AND ≠N1≠ ARE THE DIMENSIONS(VERT,HORIZ) OF THE
CPRINTOUT
C≠M≠ IS THE COLOMN LENGTH OF ≠A≠ AS DIMENSIONED IN THE
CCALLING PROGRAM
C≠IZ=0≠ CAUSES DOUBLE SPACING,OTHERWISE SINGLE SPACING
      DIMENSION A(M,N1)
      INTEGER SYMBOL
      WRITE(7,5) SYMBOL
    5 FORMAT(*0*,A6,/)
      DO 10 I=1,M1
      WRITE(7,20)  (A(I,J),J=1,N1)
   20 FORMAT(10(1X,G12.5))
      IF(IZ.EQ.0) WRITE(7,15)
   15 FORMAT(* *)
   10 CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED .098 SECONDS

```
      SUBROUTINE MATSM(A,B,C,L,MA,NA,MB,M2B,NB,MC,M2C,NC,K)
      DIMENSION A(MA,NA),B(MB,NB),C(MC,NC)
      GO TO (101,102,102,101),L
  101 IF(M2B.NE.M2C.OR.(NB.NE.NC)) GO TO 1000
      GO TO 110
  102 IF(NB.NE.M2C.OR.(M2B.NE.NC)) GO TO 1000
  110 IMAX=M2B
      JMAX=NB
      IF(L.LE.2) GO TO 200
      IMAX=NB
      JMAX=M2B
  200 IF(IMAX.GT.MA.OR.JMAX.GT.NA) GO TO 1000
      DO 600 I=1,IMAX
      DO 600 J=1,JMAX
      A(I,J)=B(I,J)+((-1)**K)*C(I,J)
      IF(L.EQ.2) A(I,J)=B(J,I)+((-1)**K)*C(I,J)
      IF(L.EQ.3) A(I,J)=B(I,J)+((-1)**K)*C(J,I)
      IF(L.EQ.4) A(I,J)=B(J,I)+((-1)**K)*C(J,I)
  600 CONTINUE
      RETURN
 1000 WRITE(7,10000)
10000 FORMAT(5X,*MATRICES NOT COFORMABLE*)
      RETURN
      END
```

41000B CM STORAGE USED .336 SECONDS

```
      SUBROUTINE MULPLT(B,A,NRY,NP,NV,XNAME,YNAME,EYMIN,EYMAX,SYM)
      INTEGER LINE(61),BLANK,DOT,SYM(NP),LO,HI
      LOGICAL MANUAL
      INTEGER XNAME(2),YNAME(2,NV)
      DIMENSION KP1(10),YMAX(10),YMIN(10),KK(10),B(NP),A(NRY,NV)
      DATA BLANK,DOT/1H ,1H./,HI,LO/94,1H≥/
      IF(NRY.LT.NP) STOP≠NRY<NP, WRONG≠
      MANUAL=.TRUE.
      IF(EYMAX.EQ.EYMIN) MANUAL=.FALSE.
      IF(EYMIN.GT.EYMAX) PAUSE≠YOU MADE A MISTAKE INPUTTING LIMITS≠
      DO 10 I=1,NV
      YMIN(I)=1.E322
      YMAX(I)=-1.E322
   10 CONTINUE
      DO 30 J=1,NV
      DO 20 I=1,NP
      IF(A(I,J).GT.YMAX(J)) YMAX(J)=A(I,J)
      IF(A(I,J).LT.YMIN(J)) YMIN(J)=A(I,J)
   20 CONTINUE
   30 CONTINUE
      IF(MANUAL) GO TO 41
      YYMIN=YMIN(1)
      YYMAX=YMAX(1)
      DO 40 I=2,NV
      IF(YMAX(I).GT.YMAX(I-1)) YYMAX=YMAX(I)
      IF(YMIN(I).LT.YMIN(I-1)) YYMIN=YMIN(I)
   40 CONTINUE
      GO TO 45
   41 YYMAX=EYMAX
      YYMIN=EYMIN
   45 RANGE=YYMAX-YYMIN
      KAXIS=60.*(-YYMIN/RANGE)+1.5
      IF(YYMIN.GT.0.0) KAXIS=1
      IF(YYMAX.LT.0.0) KAXIS=61
      DIS=RANGE/60.
      MIN=1
      XMIN=ABS(B(1))
      DO 50 J=2,NP
      IF(ABS(B(J)).GE.XMIN)  GO TO 50
      XMIN=ABS(B(J))
      MIN=J
   50 CONTINUE
   60 CONTINUE
      DO 70 K4=1,3
      WRITE(7,80)
   70 CONTINUE
   80 FORMAT(1H0)
      WRITE(7,90)
   90 FORMAT(2X,7HLEGEND:)
      DO 100 I=1,NV
      WRITE(7,110) SYM(I),YNAME(1,I),YNAME(2,I),YMIN(I),YMAX(I)
  100 CONTINUE
  110 FORMAT(8X,A1,2X,A8,2X,A8,5X,4HMIN=,G13.6,3X,4HMAX=,
     *G13.6)
      WRITE(7,80)
      WRITE(7,80)
      WRITE(7,80)
```

```
      WRITE(7,115) XNAME(1),YYMIN,YYMAX
115   FORMAT(2X,A8,5X,14HORDINATE AXIS:,6H MIN=,G13.6,
     *7H   MAX=,G13.6)
      WRITE(7,120) XNAME(2),DIS
120   FORMAT(3X,A8,30X,10HINCREMENT=,2X,G13.6,/)
      DO 130 J=1,61
      LINE(J)=BLANK
130   CONTINUE
      DO 170 J2=1,NP
      IF(J2.EQ.MIN) GO TO 180
      LINE(KAXIS)=DOT
      DO 140 I=1,NV
      X=60.0*((A(J2,I)-YYMIN)/RANGE)
      KK(I)=X+1.5
      IF(KK(I).GT.61) LINE(61)=HI
      IF(KK(I).LT.1) LINE(1)=LO
      IF(KK(I).LT.1.OR.KK(I).GT.61) GO TO 135
      LINE(KK(I))=SYM(I)
135   IF(KK(I).GT.61) KK(I)=61
      IF(KK(I).LT.1) KK(I)=1
140   CONTINUE
      WRITE(7,150) B(J2),LINE
150   FORMAT(1X,G13.6,6X,61A1)
      DO 160 I=1,NV
      LINE(KK(I))=BLANK
160   CONTINUE
170   CONTINUE
      GO TO 211
180   CONTINUE
      DO 190 J=1,61
      LINE(J)=DOT
190   CONTINUE
      DO 200 I=1,NV
      X=60.*((A(MIN,I)-YYMIN)/RANGE)
      KP1(I)=X+1.5
      IF(KP1(I).LT.1) LINE(1)=LO
      IF(KP1(I).GT.61) LINE(61)=HI
      IF(KP1(I).LT.1.OR.KP1(I).GT.61) GO TO 200
      LINE(KP1(I))=SYM(I)
200   CONTINUE
      WRITE(7,150) B(J2),LINE
      DO 210 J=1,61
      LINE(J)=BLANK
210   CONTINUE
      GO TO 170
211   DO 220 K5=1,3
      WRITE(7,80)
220   CONTINUE
      EYMAX=YYMAX
      EYMIN=YYMIN
230   CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED .816 SECONDS

```
      SUBROUTINE PLOTR(B,A,NP,XNAME,YNAME,EMIN,EMAX,ISC)
      DIMENSION B(NP),A(NP),LINE(61)
       INTEGER HI,LO,LINE,BLANK,DOT,STAR
      LOGICAL MANUAL,SYME
      INTEGER XNAME(2),YNAME(2)
      DATA BLANK,DOT,STAR/1H ,1H.,1H*/,HI/94/,LO/1H≥/
      SYME=.FALSE.
      MANUAL=.TRUE.
      IF(EMAX.EQ.EMIN) MANUAL=.FALSE.
      IF(EMIN.GT.EMAX) STOP≠YOU MADE A MISTAKE INPUTTING EMIN % EMAX≠
      IISC=ISC
      IF(MANUAL) IISC=1
      IF(MANUAL) GO TO 11
      YMIN=1.E38
      YMAX=-1.E38
      DO 10 I=1,NP
      IF(A(I).GT.YMAX) YMAX=A(I)
      IF(A(I).LT.YMIN) YMIN=A(I)
   10 CONTINUE
      GO TO 19
   11 YMAX=EMAX
      YMIN=EMIN
   19 XMIN=ABS(B(1))
      MIN=1
      DO 20 J=2,NP
      IF(ABS(B(J)).GE.XMIN)  GO TO 20
      XMIN=ABS(B(J))
      MIN=J
   20 CONTINUE
      IF(IISC.EQ.0) GO TO 21
      RANGE=YMAX-YMIN
      KAXIS=60.*(-YMIN/RANGE)+1.5
      IF(YMIN.GT.0.0) KAXIS=1
      IF(YMAX.LT.0.0) KAXIS=61
      DIS=RANGE/60.
      GO TO 30
   21 IF(YMIN.GT.0.) GO TO 22
      IF(YMAX.LT.0.) GO TO 23
      KAXIS=31
      SYME=.TRUE.
      ABY=ABS(YMIN)
      RANGE=AMAX1(YMAX,ABY)
      DIS=RANGE/30.
      GO TO 30
   22 KAXIS=1
      RANGE=YMAX
      DIS=RANGE/60.
      GO TO 30
   23 KAXIS=61
      RANGE=-YMIN
      DIS=RANGE/60.
   30 CONTINUE
      DO 40 K4=1,3
      WRITE(7,50)
   40 CONTINUE
   50 FORMAT(1H0)
      WRITE(7,60) XNAME(1),YNAME(1),YMIN,YMAX
```

```
 60   FORMAT(2X,A8,7X,A8,3X,20HORDINATE AXIS:  MIN=,G13.6,3X,4HMAX=,
     *G13.6)
      WRITE(7,70) XNAME(2),YNAME(2),DIS
 70   FORMAT(2X,A8,7X,A8,25X,10HINCREMENT=,G13.6,/)
      DO 80 J=1,61
      LINE(J)=BLANK
 80   CONTINUE
      DO 100 J2=1,NP
      IF(J2.EQ.MIN) GO TO 110
      X=60.0*((A(J2)-YMIN)/RANGE)
      IF(IISC.EQ.0.AND.YMIN.GT.0.) X=60.*(A(J2)/RANGE)
      IF(SYME) X=30.*(A(J2)/RANGE)+30.
      K=X+1.5
      LINE(KAXIS)=DOT
      IF(K.GT.61) LINE(61)=HI
      IF(K.GT.61) KMAX=61
      IF(K.GT.61) GO TO 89
      IF(K.LT.1) LINE(1)=LO
      IF(K.LT.1) KMAX=KAXIS
      IF(K.LT.1) GO TO 89
      LINE(K)=STAR
      KMAX=KAXIS
      IF(K.GT.KMAX) KMAX=K
 89   WRITE(7,90) B(J2),A(J2),(LINE(N4),N4=1,KMAX)
 90   FORMAT(1X,G13.6,2X,G13.6,2X,61A1)
      IF(K.GT.61) LINE(61)=BLANK
      IF(K.GT.61) GO TO 100
      IF(K.LT.1) LINE(1)=BLANK
      IF(K.LT.1) GO TO 100
      LINE(K)=BLANK
 100  CONTINUE
      GO TO 145
 110  CONTINUE
      DO 120 J=1,61
      LINE(J)=DOT
 120  CONTINUE
      X=60.*((A(MIN)-YMIN)/RANGE)
      IF(IISC.EQ.0.AND.YMIN.GT.0.) X=60.*(A(J2)/RANGE)
      IF(SYME) X=30.*(A(J2)/RANGE)+30.
      KP1=X+1.5
      IF(KP1.GT.61) LINE(61)=HI
      IF(KP1.LT.1) LINE(1)=LO
      IF(KP1.LT.1.OR.KP1.GT.61) GO TO 129
      LINE(KP1)=STAR
 129  WRITE(7,130) B(J2),A(J2),LINE
 130  FORMAT(1X,G13.6,2X,G13.6,2X,61A1)
      DO 140 J=1,61
      LINE(J)=BLANK
 140  CONTINUE
      GO TO 100
 145  DO 150 K5=1,3
      WRITE(7,50)
 150  CONTINUE
      EMAX=YMAX
      EMIN=YMIN
 160  CONTINUE
      RETURN
```

```
115            END
```

41000B CM STORAGE USED .817 SECONDS

```
      FUNCTION PVALUE(X,P,DEL)
C
C     THIS ASSUMES X RANGES ON THE REAL LINE, MODULO 2P AND
C REDUCES X TO THE INTERVAL (-P+DEL,P-DEL), EXCLUSIVE OF A SMALL
C INTERVAL ABOUT THE ORIGIN, (-DEL,DEL).
C
      B=X/P
      C=P*(AMOD(ABS(B)+1.,2.)-1.)
      IF (DEL .NE. 0.) C=SIGN(AMAX1(AMIN1(ABS(C),P-DEL),DEL),C)
      IF (B.LT. 0.) C=-C
      PVALUE=C
      RETURN
      END
```

41000B CM STORAGE USED .092 SECONDS

```
      SUBROUTINE REALIO(SYMBOL,VALUE,IZ)
      INTEGER ISW,ISLASH,SYMBOL
      DATA ISLASH/1H//
      IF(IZ) 1,2,3
    1 READ(10,10) VALUE
   10 FORMAT(G13.6)
      WRITE(7,20) VALUE
   20 FORMAT(1H ,G13.6/)
      RETURN
    2 WRITE(7,30) SYMBOL,VALUE
   30 FORMAT(1H ,A6,3H = ,G13.6)
      READ (10,40) ISW,VAL2
   40 FORMAT (A1,G13.6)
      IF (ISW .EQ. ISLASH) GO TO 60
      WRITE(7,50)
   50 FORMAT(1H )
      RETURN
   60 VALUE=VAL2
      WRITE(7,70) ISLASH,VALUE
   70 FORMAT(1H+,22X,A1,G13.6/)
      RETURN
    3 WRITE(7,80) SYMBOL,VALUE
   80 FORMAT(1H ,A6,3H = ,G13.6/)
      RETURN
      END
```

410008 CM STORAGE USED .098 SECONDS

```
      SUBROUTINE RETURN(LFN)
      DIMENSION FIT(1)
      IX=INDXFIT(LFN,FIT)
      CALL STOREF(FIT(IX),2LCF,1LU)
    5 CALL CLOSEM(FIT(IX))
      RETURN
      END
```

41000B CM STORAGE USED .047 SECONDS

A-111

```
      FUNCTION SATU(X,XL)
      IF (ABS(X).LE.XL) GO TO 2
      SATU=SIGN(XL,X)
      RETURN
    2 SATU=X
      RETURN
      END
```

41000B CM STORAGE USED .046 SECONDS

```
      PROGRAM PCRMP1(INPUT,OUTPUT,TAPE10=INPUT,TAPE15,TAPE7=OUTPUT,
     *TAPE20)
      INTEGER DATIN(4)
      DIMENSION TIM(102),YY(102), TODAY(3),PKVAL(3),YPK(3),ALYSNR(3)
      DIMENSION EMEAN(3),ERMS(3),ESTDEV(3),TCOUNT(3),YMIN(3),YMAX(3)
      DIMENSION IDENTS(6,3),CSNRTO(102),THESEP(102),JOBNAM(3),CSNRFR(102)
      DIMENSION DOUT(3),ABBORT(100),THESER(100,3)
      DIMENSION NAME1(10),NAME2(10),NAME(10),YPEAK(3),ALYERR(3)
      DIMENSION NAMSEP(3),NAMSNR(2),NAMTIM(4)
      DIMENSION YORERR(3,3)
      EQUIVALENCE (RNAME2,NAME2(10)),(RNAME,NAME(10))
      DATA YMMAX/1.0/,H/0.07/,SIZE/0.5/,S/.12/
      DATA NAME1/10HSIM. JOB: ,3*1H ,10H FILE NO: ,3*1H ,10H    DATE: ,
     *1H /
      DATA NAME2/10HPLOT JOB: ,3*1H ,10H PROGRAM: ,10HPCRMP1 ,2*1H ,
     *10H    DATE: ,1H /
      DATA NAME/10*2H  /
      DATA DATIN/10HMLSSIMDATA,3*0/
      DATA NAMSNR/10HCSNR (TO-S,10HCAN)      /
      DATA NAMSEP/10H   SEPARAT,10HION ANGLE ,10H(DEG.)    /
      DATA NAMTIM/10HTIME SINCE,10H START OF ,10HFIRST SCAN,10H (SECONDS
     *)/
      DATA ALXTIM/5.0/
      DATA ALYERR/4.0,2.0,1.5/
      DATA ALYSNR/2.0,1.5,1./
      DATA YPK/1.31,.31,.07/
      DATA YORERR/4.0,0.0,0.0,6.0,3.25,0.0,6.5,4.5,2.5/
      DATA YORSNR/1.0/,IBLANK/2H  /
C
C
C
      CALL CALCOMP(20)
      CALL FACTOR (1.)
      CALL PLOT(2.00,1.50,-3)
      NAME2(2)=JOBNAME(A)
      RNAME2=DATE(A)
      READ(10,10) NPLOTS
   10 FORMAT(I2)
      WRITE(7,20) NAME2(2),RNAME2,NPLOTS
   20 FORMAT(11H1JOBNAME:  ,A10,15X,7HDATE:  ,A10//1H ,I3,
     *17H PLOTS TO BE DONE)
C******************320
      DO 320 NP=1,NPLOTS
      READ(10,30) NFILES
   30 FORMAT(I1)
C******************180
      DO 180 N=1,NFILES
      READ(10,40)  DATIN(2)
   40 FORMAT(53X,A10)
      WRITE(7,50) NP,NPLOTS, (DATIN(I),I=1,2),N,NFILES
   50 FORMAT(6H1PLOT ,I2,4H OF ,I2/12H INPUT FILE ,2A10,2H (,I3,
     *4H OF ,I3,2H):/)
      IR=IATTACH(6LTAPE15,DATIN)
      IF(IR.EQ.0) GO TO 70
      WRITE(7,60)
      STOP*INPUT FILE ATTACH NOT SATISFACTORY*
   60 FORMAT(20H FILE DID NOT ATTACH/)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      C
      C READ FILE INTO NTH ELEMENTS OF ARRAYS,WHERE NECESSARY
      C
      70    READ(15) DOUT(N),DELT,MTIMES,LGMAX,KM,TODAY(N),JBNAM(N)
            READ(15) (IDENTS(I,N),I=1,6)
            READ (15) NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
      C******************80
            DO 80 K=1,KM
            READ(15) CSNRTO(K),CSNRFR(K),THESER(K,N),ABBORT(K),THESEP(K)
      80    CONTINUE
            READ(15) EMEAN(N),ERMS(N),ESTDEV(N),TCOUNT(N),YMIN(N),YMAX(N)
            CALL RETURN(6LTAPE15)
      C
      C WRITE DATA ON LINEPRINTER AS IN CONTROL
      C
            WRITE(7,90)  DOUT(N),DELT,MTIMES,LGMAX,KM,TODAY(N),JBNAM(N)
      90    FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
            WRITE(7,100) (IDENTS(I,N),I=1,6)
      100   FORMAT(1H ,6(1X,A8,9X)/)
            WRITE(7,110)  NRUN,DSNRDB,RHO,BETA,FSC,BMLS,BRCVR
      110   FORMAT(1H ,5X,I2,6X,6(5X,G13.6)/)
      C******************130
            DO 130 K=1,35
            WRITE(7,120) CSNRTO(K),CSNRFR(K),THESER(K,N),ABBORT(K),THESEP(K),K
      120   FORMAT(1H ,3(G13.6,5X),6X,A1,11X,G13.6,5X,I7)
      130   CONTINUE
      C******************150
            DO 150 K=1,5
            WRITE(7,140)
      140   FORMAT(1H ,5(6X,1H.,11X))
      150   CONTINUE
            KM3=KM-3
      C******************160
            DO 160 K=KM3,KM
            WRITE(7,120) CSNRTO(K),CSNRFR(K),THESER(K,N),ABBORT(K),THESEP(K),K
      160   CONTINUE
            WRITE(7,170)  EMEAN(N),ERMS(N),ESTDEV(N),TCOUNT(N),YMIN(N),YMAX(N)
      170   FORMAT(1H0,6(G13.6,5X)/)
      180   CONTINUE
      C
      C DRAW LEFT EDGE OF THE PAGE
            CALL PLOT(-2.0,0.0,3)
            CALL PLOT(-2.0,7.0,2)
      C
      C WRITE INFORMATION ON THE TOP OF THE PAGE
      C
            YH=8.8-S
            XH=-0.5
            CALL SYMBOL(XH,YH,H,NAME1,0.0,90)
            Y2H=8.8-(NFILES+1)*S
      C******************190
            DO 190 I=1,10
            NAME(I)=IBLANK
      190   CONTINUE
      C******************200
            DO 200 N=1,NFILES
            NAME(2)=JBNAM(N)
```

```
      DECODE(10,195,DOUT(N)) NAME(6)
  195 FORMAT(A10)
      RNAME=TODAY(N)
      Y1H=8.8-N*S
      CALL SYMBOL(XH,Y1H,H,NAME,0.0,100)
  200 CONTINUE
      CALL SYMBOL(XH,Y2H,H,NAME2,0.0,100)
C
C ORIGINATE TIME DELAY
C
      KM1=KM+1
      KM2=KM+2
C*******************210
      DO 210 K=1,KM
      K1=K-1
      TIM(K)=DELT*K1
C
C FIND MAX,MIN,AND INCREMENT FOR CSNTRO
C
  210 CONTINUE
      CSNRTO(KM1)=0.0
      SNRINC=IFIX(CSNRTO(1)+.5)
      CSNRTO(KM2)=SNRINC
      YMM=YMMAX
      YMIX=-1.E+320
      ALX=ALXTIM/SIZE
      CALL SCALE(TIM,ALX,KM,1)
C*******************220
      DO 220 N=1,NFILES
      YPEAK(N)=AMAX1(-YMIN(N),YMAX(N))
      YMIX=AMAX1(YPEAK(N),YMIX)
  220 CONTINUE
      YMM=AMIN1(YMM,YMIX)
C*******************260
      DO 260 N=1,NFILES
C*******************230
      DO 230 K=1,KM
      YY(K)=THESER(K,N)
      ZS=SIGN(YMM,YY(K))
      IF(ABS(YY(K)).GE.YMM) YY(K)=ZS*YMM
  230 CONTINUE
      ENCODE(23,240,PKVAL) YPEAK(N)
  240 FORMAT(13HPEAK VALUE = ,F5.3,5H DEG.)
      YY(KM1)=-YMM
      YY(KM2)=2.*SIZE*YMM/ALYERR(NFILES)
      CALL FACTOR(1.)
      CALL PLOT(0.0,YORERR(N,NFILES),-3)
      XB=-3.0*S
      YB=YPK(NFILES)
      CALL SYMBOL(XB,YB,H,PKVAL,90.0,23)
      CALL FACTOR(SIZE)
      ALY=ALYERR(NFILES)/SIZE
      ENCODE(17,250,NAMERR) IDENTS(3,N)
  250 FORMAT(A8,9H RECEIVER)
      CALL AXIS(0.0,0.0,NAMERR,17,ALY,90.0,YY(KM1),YY(KM2))
      CALL LINE(TIM,YY,KM,1,0,0)
      CALL FACTOR(1.0)
```

```
          CALL PLOT(0.0,-YORERR(N,NFILES),-3)
  260     CONTINUE
          CALL PLOT(0.0,YORSNR,-3)
          CALL FACTOR(SIZE)
          ALLY=ALYSNR(NFILES)/SIZE
          CALL AXIS(0.0, 0.0,NAMTIM,-40,ALX,0.0,TIM(KM1),TIM(KM2))
          CALL AXIS(0.0,0.0,NAMSNR,14,ALLY,90.0,CSNRTO(KM1),CSNRTO(KM2))
          DTS=THESEP(2)-THESEP(1)
          TSMIN=THESEP(1)
          TSINC=(DTS/DELT)*TIM(KM2)
          EPS=-.5/SIZE
          CALL AXIS(0.0,EPS,NAMSEP,-30,ALX,0.0,TSMIN,TSINC)
          CALL LINE(TIM,CSNRTO,KM,1,0,0)
          CALL FACTOR(1.0)
          CALL PLOT(0.0,-YORSNR,-3)
          XH=-.44
          YH=-S
          ENCODE(93,270,NAME)DSNRDB,RHO,BETA,FSC,KM,BMLS,IDENTS(2,1)
  270     FORMAT(4HS/N=,F5.1,5H DB, ,5H RHO=,F5.2,2H, ,5HBETA=,F6.1,6H DEG,
         *,4HFSC=,F6.1,5H HZ, ,3HKM=,I4,8H SCANS, ,5HBMLS=,F5.2,2H, ,A8)
          CALL SYMBOL(XH,YH,H,NAME,0.0,93)
          YH=YH-S
C******************310
          DO 310 N=1,NFILES
          YH=YH-S
          DECODE(10,280,DOUT(N)) AA,BB,CC
  280     FORMAT(A2,A1,A7)
          DECODE(1,285,BB) IRCVR
  285     FORMAT(I1)
          IF(IRCVR.EQ.1) ENCODE(78,290,NAME) (IDENTS(I,N),I=3,5),TCOUNT(N),
         *ERMS(N)
  290     FORMAT(A7,3H:  ,A7,2H, ,A8,2H, ,F6.2,20H% OF SCANS ABORTED, ,
         *5HERMS=,G13.6,5H DEG.)
          IF(IRCVR.NE.1) ENCODE(78,300,NAME) (IDENTS(I,N),I=3,6),BRCVR,
         *ERMS(N)
  300     FORMAT(A7,3H:  ,A7,2H, ,A8,2H, ,A6,8H, BRCVR=,F4.2,8H DEG., ,
         *5HERMS=,G13.6,5H DEG.)
          CALL SYMBOL(XH,YH,H,NAME,0.0,78)
  310     CONTINUE
          CALL PLOT(8.5,0.0,-3)
  320     CONTINUE
          CALL PLOT(-2.0,0.0,3)
          CALL PLOT(-2.0,7.0,2)
          CALL ENDPLT
          STOP
          END
```

42000B CM STORAGE USED 4.128 SECONDS

```
      PROGRAM RLOGSW (OUTPUT,TAPE10,TAPE7=OUTPUT)
C   THIS READS <SWA>, <SWB>, % <RAB>
      REAL NC(1000),NS(1000)
      INTEGER FNA(4)
      COMMON/DATA1/QA,B,U,WAA,WBA,WA2A,WB2A,WABA,WA,WB
      COMMON/DATA2/LMAX1,COSB,JNMAX,NC,NS,ALNLMX,RIJMAX
      DIMENSION QAI(25),BJ(12),BULK(12,25,3),DATUM(5),COSB(101)
      EQUIVALENCE (WAA,DATUM(1))
      DATA PI/3.1415927/,LMAX/100/,JNMAX/1000/,JBMAX/12/
      DATA FNA/10HMLSLOGSWCD,1LC,2*0/
C
C FOLLOWING READS DATA FROM FILE
      IX=IATTACH(6LTAPE10,FNA)
      READ (10) LMAX1,JNMAX
      READ (10) IQMAX,(QAI(I),I=1,IQMAX)
      READ (10) JBMAX,(BJ(J),J=1,JBMAX)
      READ (10) (((BULK(J,I,K),J=1,JBMAX),I=1,IQMAX),K=1,3)
C
      DO 80 K=1,3
      IF(K.EQ.1) WRITE(7,870)
      IF(K.EQ.2) WRITE(7,871)
      IF(K.EQ.3) WRITE(7,872)
  870 FORMAT (1H1,///,5HOSWA:)
  871 FORMAT (1H1,///,5HOSWB:)
  872 FORMAT (1H1,///,5HORAB:)
      WRITE (7,880) (BJ(J),J=1,JBMAX)
  880 FORMAT (4HOB =,8X,12(2X,F6.3,2X))
      WRITE (7,885)
  885 FORMAT (4H QA:)
      DO 70 I=1,IQMAX
      WRITE (7,890) QAI(I),(BULK(J,I,K),J=1,JBMAX)
  890 FORMAT (1H ,G11.4,12(1X,G9.2))
   70 CONTINUE
   80 CONTINUE
      STOP
      END
```

410008 CM STORAGE USED .300 SECONDS

RLOGSW
Page 1 of 1

```
      PROGRAM WLOGSW(INPUT,OUTPUT,PUNCH,TAPE12,TAPE5=INPUT,TAPE7=OUTPUT,
     1TAPE6=PUNCH)
C     THIS COMPUTES <WA>, <WB>, <SWA>, <SWB>, % <RAB>
      REAL NC(1000),NS(1000)
      COMMON/DATA1/QA,B,U,WAA,WBA,WA2A,WB2A,WABA,WA,WB
      COMMON/DATA2/LMAX1,COSB,JNMAX,NC,NS,ALNLMX,RIJMAX
      DIMENSION QAI(25),BJ(12),DATUM(5)
      DIMENSION DATA3(12,25,3),BULK(12,25,3),COSB(101),DATA4(937)
      EQUIVALENCE (WAA,DATUM(1)),(QAI(1),DATA4(1)),(BJ(1),DATA4(26))
      EQUIVALENCE (DATA4(38),DATA3(1,1,1))
C
      DATA PI/3.1415927/,LMAX/100/
      DATA JNMAX/1000/,IQMAX/25/,JBMAX/12/
      DATA QAI/.1,.1778,.3162,.5623,1.,1.778,3.162,5.623,10.,
     *17.78,31.62,56.23,100.,177.8,316.2,562.3,1000.,1778.,3162.,
     *5623.,1.E4,1.E5,1.E6,1.E7,1.E8/
      DATA BJ/.01,.02,.04,.06,.1,.2,.3,.5,.7,.9,.95,.99/
C
      SQ22=SQRT(.5)
      READ(5,8)LMAX,JNMAX
    8 FORMAT(2I5)
      WRITE(7,9) LMAX,JNMAX
    9 FORMAT(7H LMAX= ,I5/8H JNMAX= ,I5/)
      BETA2=PI/LMAX
      LMAX1=LMAX+1
      DO 10 L=1,LMAX1
      COSB(L)=COS((L-1)*BETA2)
   10 CONTINUE
      DO 15 J=1,JNMAX
      NC(J)=SQ22*GAUSS(0.)
      NS(J)=SQ22*GAUSS(0.)
   15 CONTINUE
      ALMAX1=LMAX1
      ALNLMX=ALOG(ALMAX1)
      RIJMAX=JNMAX*ALMAX1
      RIJMAX=1./RIJMAX
      WRITE (7,800) LMAX1
  800 FORMAT (1H ,I4,25H VALUES OF BETA (0 TO PI))
      WRITE (7,810) LMAX1
  810 FORMAT (1H ,I4,27H VALUES OF BETA-U (0 TO PI))
      WRITE (7,820) JNMAX
  820 FORMAT (1H ,I5,34H VALUES OF NOISE SAMPLES (NCJ,NSJ))
   30 CONTINUE
      WRITE (7,830) QAI(1),QAI(2),QAI(IQMAX),IQMAX
  830 FORMAT (4H QA=,G13.6,2H, ,G13.6,6H,..., ,G13.6,2H (,I2,
     *    8H) VALUES)
      WRITE (7,840) BJ(1),BJ(2),BJ(JBMAX),JBMAX
  840 FORMAT (3H B=,F7.4,2H, ,F7.4,6H,..., ,F7.4,2H (,I2,
     *     8H) VALUES)
      DO 60 I=1,IQMAX
      QA=QAI(I)
      DO 50 J=1,JBMAX
      B=BJ(J)
      CALL WAVGS
      DO 45 K=1,3
      BULK(J,I,K)=DATUM(K+2)
   45 CONTINUE
```

```
 50    CONTINUE
 60    CONTINUE
C     CHANGES TO LOG
       DO 100K=1,2
       DO 100 I=1,IQMAX
       DO 100 J=1,JBMAX
       DATA3(J,I,K)=ALOG(SQRT(BULK(J,I,K)))
100    CONTINUE
       DO 200 I=1,IQMAX
       DO 200 J=1,JBMAX
200    DATA3(J,I,3)=BULK(J,I,3)/SQRT(BULK(J,I,1)*BULK(J,I,2))
C     PUNCHES CARDS
       IIJJ=3*IQMAX*JBMAX+IQMAX+JBMAX
       WRITE(6,900) LMAX1,JNMAX,IQMAX,JBMAX,1
900    FORMAT(4(I5,5X),26X,*MLSLOGSWCDC*,I3)
       WRITE(6,901) ((DATA4(I),I+1),I=1,IIJJ)
901    FORMAT (G23.16,43X,*MLSLOGSWCDC*,I3)
C    WRITES ON DATA FILE LOGSW2.DAT
       WRITE(12) LMAX1,JNMAX
       WRITE (12) IQMAX,(QAI(I),I=1,IQMAX)
       WRITE (12) JBMAX,(BJ(J),J=1,JBMAX)
       WRITE (12) (((DATA3(J,I,K),J=1,JBMAX),I=1,IQMAX),K=1,3)
C    WRITES ON DECWRITER
C
       DO 80 K=1,3
       IF(K.EQ.1) WRITE(7,870)
       IF(K.EQ.2) WRITE(7,871)
       IF(K.EQ.3) WRITE(7,872)
870    FORMAT(/,1H0,4HSWA:)
871    FORMAT(/,1H0,4HSWB:)
872    FORMAT(/,1H0,4HRAB:)
       WRITE (7,880) (BJ(J),J=1,JBMAX)
880    FORMAT (4HOB =,8X,12(2X,F6.3,2X))
       WRITE (7,885)
885    FORMAT (4H QA:)
       DO 70 I=1,IQMAX
       WRITE (7,890) QAI(I),(DATA3(J,I,K),J=1,JBMAX)
890    FORMAT (1H ,G11.4,12(1X,G9.2))
70     CONTINUE
80     CONTINUE
       STOP
       END

410008 CM STORAGE USED       .827 SECONDS
```

```
      SUBROUTINE WAWB
CTHIS COMPUTES A SAMPLE EACH OF THE PROCESSES WA % WB
      REAL NC(1000),NS(1000)
      COMMON/DATA1/QA,B,U,WAA,WBA,WA2A,WB2A,WABA,WA,WB
      COMMON/DATA2/LMAX1,COSB,JNMAX,NC,NS,ALNLMX,RIJMAX
      DIMENSION HA(101),HB(101),COSB(101),G(101),H(101)
C
      GMAX=-1.E+322
      HMAX=1.
      HAMAX=1.
      HBMAX=1.
      DO 20 L=1,LMAX1
      COSBL=COSB(L)
      BCBL1=AMAX1(1.+B*COSBL,0.)
      ABC=QA*BCBL1
      UABC=U*ABC
      RAD=2.*SQRT(UABC)
      RAD1=SQRT(1.+UABC)
      GL=2.*(RAD1-1./(1.+RAD))-ABC
      G(L)=GL
      HL=1./SQRT(1.+6.2831846*RAD)
      H(L)=HL
      HAL=(-1.+U/RAD1)*HL
      HA(L)=HAL
      HBL=HAL*COSBL
      HB(L)=HBL
      GMAX=AMAX1(GL,GMAX)
      HMAX=AMAX1(HL,HMAX)
      HAMAX=AMAX1(ABS(HAL),HAMAX)
      HBMAX=AMAX1(ABS(HBL),HBMAX)
   20 CONTINUE
      GZ=GMAX-741.
      FZ=ALOG(HMAX)+GZ
      GM=FZ+ALNLMX
      GZ=GZ+ALNLMX
      GMA=GZ+ALOG(HAMAX)
      GMB=GZ+ALOG(HBMAX)
      FS=0.
      FSA=0.
      FSB=0.
      DO 30 L=1,LMAX1
      GL=G(L)
      GLM=GL-GM
      GLMA=GL-GMA
      GLMB=GL-GMB
      IF(GLM.GE.-674.) FS=FS+EXP(GLM)*H(L)
      IF(GLMA.GE.-674.) FSA=FSA+EXP(GLMA)*HA(L)
      IF(GLMB.GE.-674.) FSB=FSB+EXP(GLMB)*HB(L)
   30 CONTINUE
      WA=(HAMAX/HMAX)*(FSA/FS)
      WB=(HBMAX/HMAX)*(FSB/FS)
      RETURN
      END
```

41000B CM STORAGE USED   .399 SECONDS

```
      SUBROUTINE WAVGS
C GIVEN QA % B, THIS COMPUTES VARIOUS AVERAGES OF WA % WB OVER U
      REAL NCJ,NSJ,NC(1000),NS(1000)
      DIMENSION DATUM (5)
      COMMON/DATA1/QA,B,U,WAA,WBA,WA2A,WB2A,WABA,WA,WB
      COMMON/DATA2/LMAX1,COSB(101),JNMAX,NC,NS,ALNLMX,RIJMAX
      EQUIVALENCE (WAA,DATUM(1)),(IBUMAX,LMAX1)
C
      DO 10 K=1,5
      DATUM(K)=0.
   10 CONTINUE
      DO 30 I=1,IBUMAX
      COSBUI=COSB(I)
      QU=QA*AMAX1(1.+B*COSBUI,0.)
      SQRQU2=2.*SQRT(QU)
      DO 20 J=1,JNMAX
      NCJ=NC(J)
      NSJ=NS(J)
      U=QU+NCJ*SQRQU2+NCJ*NCJ+NSJ*NSJ
      CALL WAWB
      WAA=WAA+WA
      WBA=WBA+WB
      WA2A=WA2A+WA*WA
      WB2A=WB2A+WB*WB
      WABA=WABA+WA*WB
   20 CONTINUE
   30 CONTINUE
      DO 40 K=1,5
      DATUM(K)=DATUM(K)*RIJMAX
   40 CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED .175 SECONDS

APPENDIX B

EXPERIMENTAL SYSTEM DATA ACQUISITION

PDP-11/03 DRV 11 Parallel Line Unit signals which will be used:

Outputs:
- OUT ØØ through OUT 15*
- NEW DATA READY*
- DATA TRANSMITTED*
- CSRØ*

Inputs:
- IN ØØ through IN 09*
- REQ A*
- REQ B*

Signals which will be generated by the data acquisition hardware:

BIT Ø through BIT 9*
START CONVERT*
STATUS*
CLEAR CONTROLLER*
TO SCAN IN (DECODE 33)*
t_s RECEIVED
COMPARISON MATCH*
COUNTER ENABLE
COUNTER DISABLE
ID RECEIVED ≠ ID REQUESTED
$t_s \cdot$ TO SCAN IN
ID RECEIVED = ID REQUESTED
$t_s \cdot \overline{\text{TO SCAN IN}}$
SØ through S5*

Signals which are expected from the BENDIX receiver:

t_R*
ELevation*
AZimuth*
Log Video(+)* and Log Video(-)*

Signals which will be supplied from front panel switch: Basic Wide (BW) or Basic Narrow (BN)*

Quick Overview of Normal Operation

After initialization, the hardware will ask the PDP 11/Ø3 to select the

Note: * indicates that the signal will be transmitted via the backplane of the VERO rack.

type of scan (Azimuth or Elevation), which should next be sampled. In response the PDP will read out the samples from the previously sampled scan which were stored in a semiconductor memory; then, the PDP will select the next type of scan to be sampled. This selection is referred to as "ID REQUESTED" (ID REQ).

Now, the hardware waits for the t_R signal from the BENDIX receiver which indicates the beginning of the next scan. (Note: t_R must come shortly after Bit 5 of the BARKER code (data Bits 1 through 5); however, scan function identification is associated with data bits 6 through 11. The current design assumes that Azimuth or Elevation scans are the only type being transmitted from the airport. If this is not the case design changes must be made. This problem will be discussed further in relation to the Function ID logic.) After receipt of the t_R pulse, the hardware waits for either the Azimuth or the Elevation line from the BENDIX receiver to be raised. This is referred to as the "ID RECEIVED" (ID REC).

If the ID requested is unequal to the ID received, then the hardware waits until the desired scan is received. When the desired scan is received (ID requested = ID received), the hardware asks the PDP to tell it when to start sampling the TO scan. The PDP sends a 16-bit t_s word to the hardware. When a hardware timer operating at 3.84 MHz counts up to the t_s word, TO scan sampling commences. The t_s word measures time with respect to the t_R reference, but there are some delays which are involved which must be compensated for by adjusting the t_s word sent to the hardware. These adjustments will be discussed later.

The data samples have 10-bit resolution and are stored in a 256 x 12 semiconductor memory. After 33 samples are taken, the hardware asks the PDP to send a t_s word to tell it when to start FRO scan sampling. When the hardware timer counts up to this t_s word, FRO scan sampling commences. Then, 34 samples are taken (the last one will be ignored when data are read out for processing).

Finally, the hardware returns to the state in which it asks the PDP to select the next type of scan to be sampled. The PDP reads out the 66 samples and sends out the next ID requested.

Detailed Description of the Data Acquisition Hardware

The detailed description will come in two parts. First, the hardware will be divided into functional units which will be described as separate entities. Second, a step-by-step description of the operation of the hardware will be given to show how the functional units interact by passing signals through the backplane of the VERO rack. The backplane wiring will be described after all of the functional units have been presented. The descriptions are most meaningful if they are read along with the logic diagrams of the hardware.

Functional Units

The hardware is divided into separate functional units, each of which has its own Vero Finger Board. These boards will be mounted in a Vero rack, and the backplane will provide the necessary interconnections from board-to-board and from the rack to the PDP 11/Ø3 computer. The functional units are:

a. POWER SUPPLY
b. OSCILLATOR (3.84 MHz Clock)
c. A/D and Sample/Hold
d. STATE CONTROLLER and FUNCTION ID LOGIC
e. SCAN TIMER
f. SAMPLE TIMER & BUFFER MEMORY

Power Supply (Figure B-1)

The power supply is a Datel Systems BPM 15/150-D5, which converts +5 volts to ±15 volts with COMMON. The supply has been mounted on a PC board with an aluminum plate for heat dissipation. Note: It is very important in a

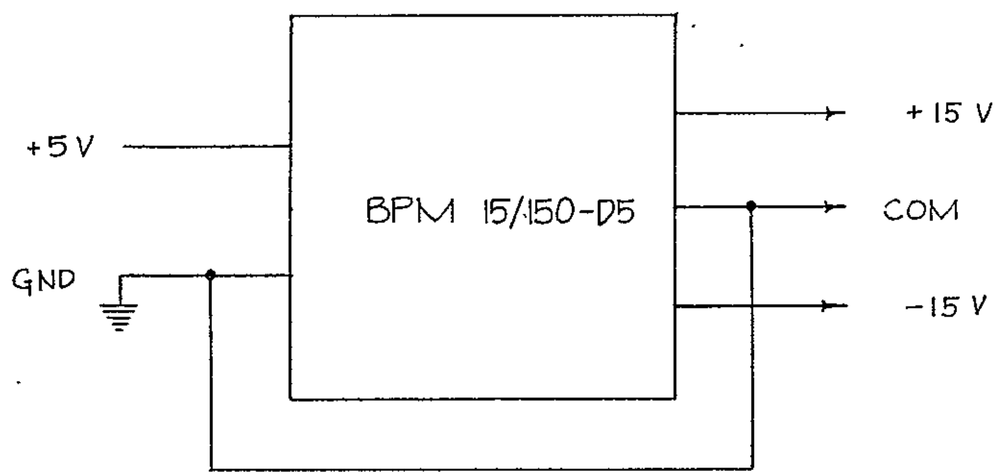


Figure B-1 Power Supply

system with both analog and digital signals present to maintain the correct distinction between analog and digital ground. However, these must ultimately be tied together -- preferably, as close to the power supply as possible. Therefore, GROUND and COMMON have been wired together on the power supply board. [Note: second thoughts suggest the two supplies should be tied together at the A/D Converter (see below); this will be considered further]

Inputs: +5 Volts, GROUND Outputs: +15 Volts, -15 Volts, COMMON

Oscillator

The oscillator is a 3.84 MHz crystal oscillator which has been mounted on a special PC board which provides a ground plane underneath the entire unit.

Inputs: +5 Volts, GROUND Outputs: CLOCK

A/D and Sample/Hold (Figure B-2)

This board will include an AD509J Op Amp and an ADC1109 Analog-to-Digital Converter, both by Analog Devices, plus an SHM-12 Sample and Hold by Datel. The op amp is used in the standard noninverting configuration, with high-frequency compensation, to provide a gain of four. This gain will boost the 0 - 2.5 Volts swing of the Log Video input to 0 - 10 Volts, which is the unipolar range of the A/D converter.

The board will operate as follows. The A/D converter will receive a START CONVERT command, via the backplane, from the SAMPLE TIMER & BUFFER MEMORY board. The converter will then raise its STATUS output signal, which will make the SHM-12 hold the current value of the Log Video input. Approximately four microseconds later, when the conversion is completed, the STATUS line will drop low. The 10 bits of digitized data are then valid. The negative-going edge of the STATUS signal will cause the 10 bits to be written into the buffer memory via the backplane. Note: While the schematic diagram for this board shows the

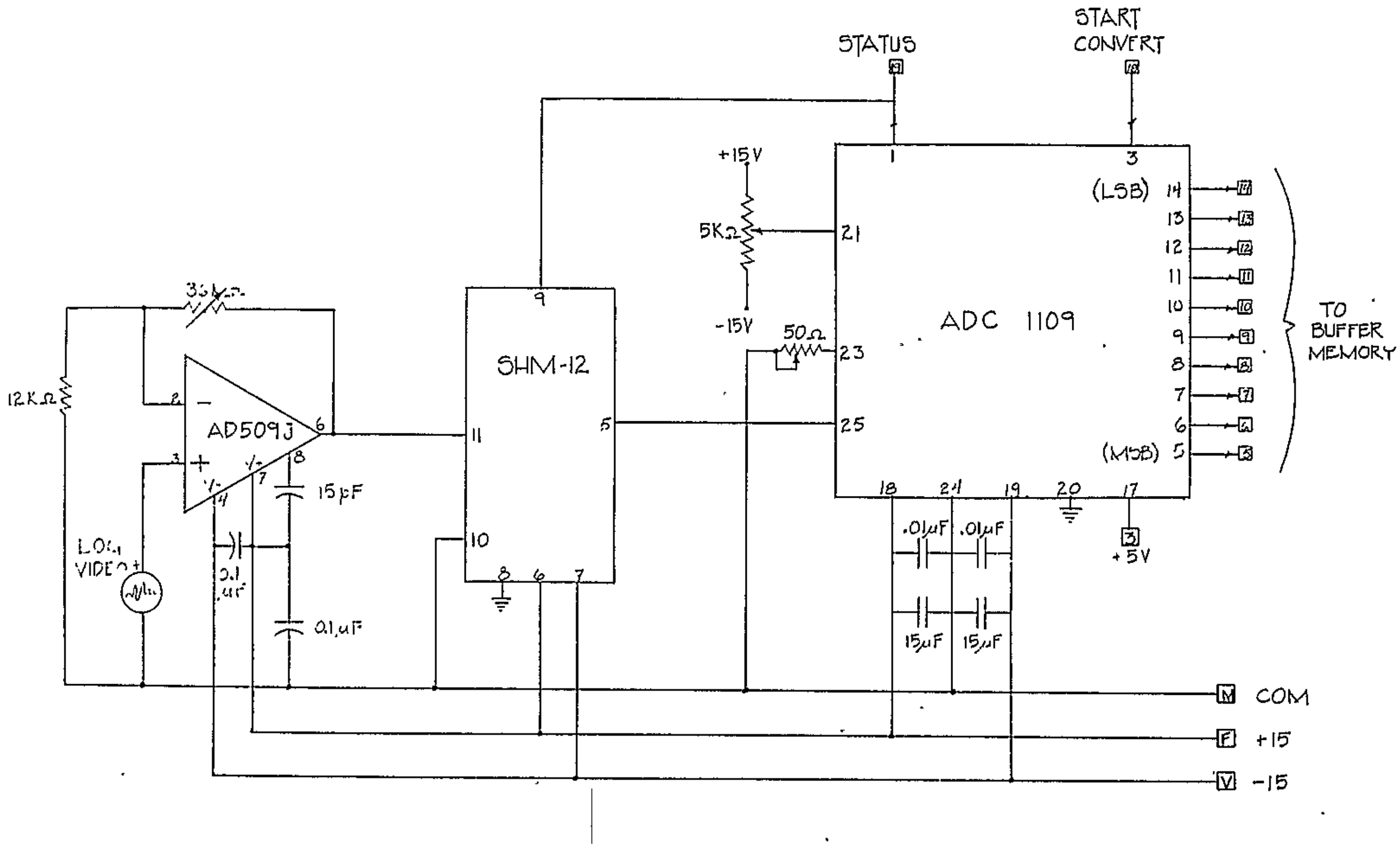


Figure B-2 A/D and Sample/Hold

ADC 1109 with trim pot adjustments for the zero adjust (Pin 21) and the gain adjust (Pin 23), these should be replaced ultimately with precision resistors. Also, it is assumed that Log Video will be brought to the backplane from the BENDIX receiver through a shielded cable.

Inputs: +5 Volts, GROUND, +15 Volts, -15 Volts, COMMON, START CONVERSION, Log Video(+), Log Video(-)

Outputs: STATUS, Bit 1(MSB) through Bit 10(LSB)

State Controller and Function ID Logic (Figures B-3 and B-4)

The state controller design is based on the control-state counter, presented in Section 5.11 of Thomas R. Blakeslee's book, Digital Design with Standard MSI and LSI. The controller consists of two 74151 data multiplexers, a 74193 presettable binary counter, and a 7442 demultiplexer. The inverters on the outputs of the demultiplexer are used for buffering and also establish positive logic; thus, when a line, such as S1, goes high, the controller is in State 1. The operation of the controller is discussed in Blakeslee's book and will not be described here. However, we may summarize its performance with the state diagram on the following page (Figure B-3).

The signals which cause the transitions from one state to another are produced at several different points in the data acquisition system:

Signal	Origin
CLEAR CONTROLLER	SAMPLE TIMER & BUFFER MEMORY LOGIC; the signal goes high at initialization when CSRØ is raised and, also, after FRO scan sampling.
NEW DATA READY	PDP 11/Ø3 DRV 11; this signal is raised when the DRV 11 places data in its output buffer. The data in this case is a code indicating the type of scan which the computer wants to sample next.
	BENDIX receiver.

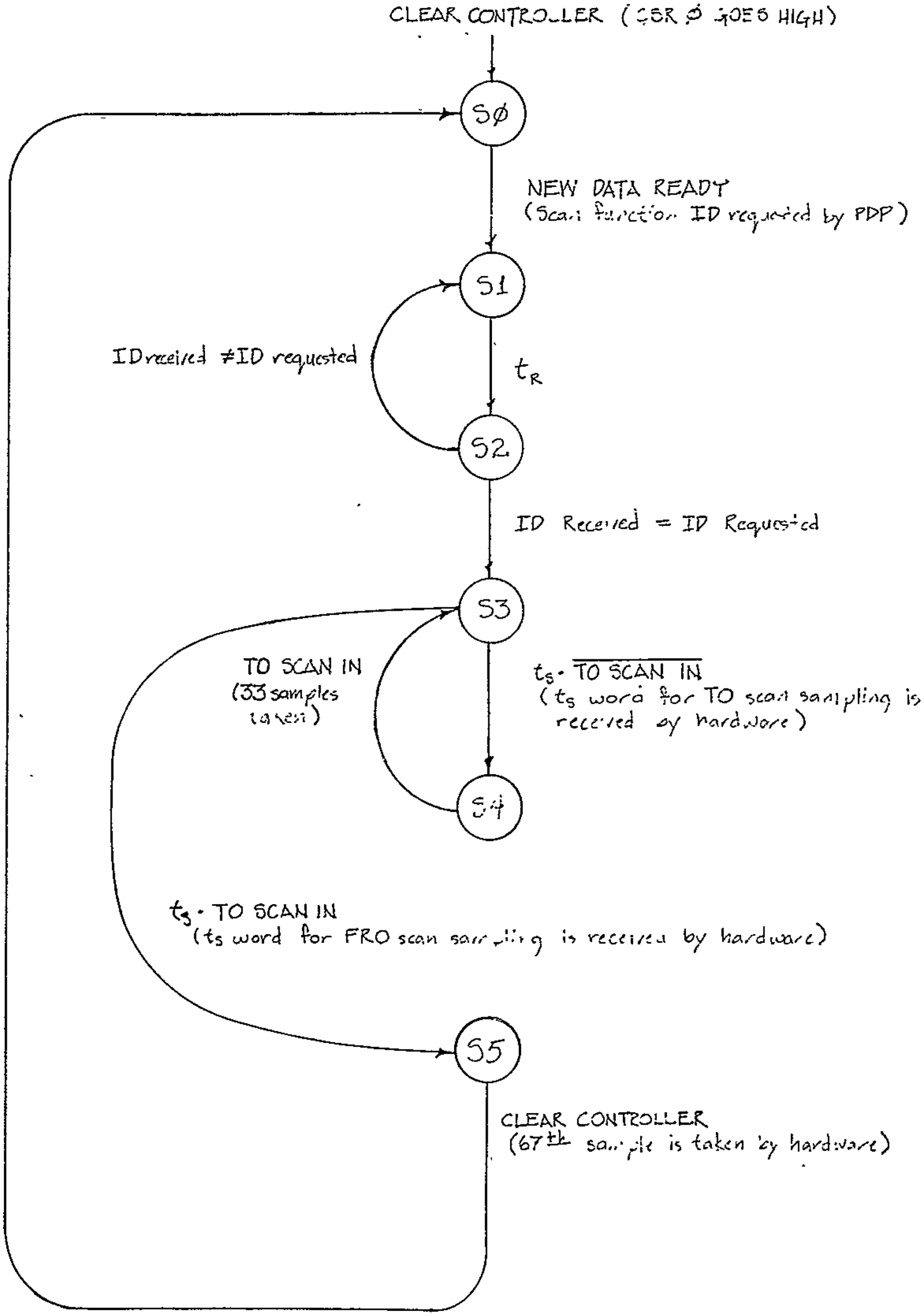


Figure B-3 State Diagram for State Controller

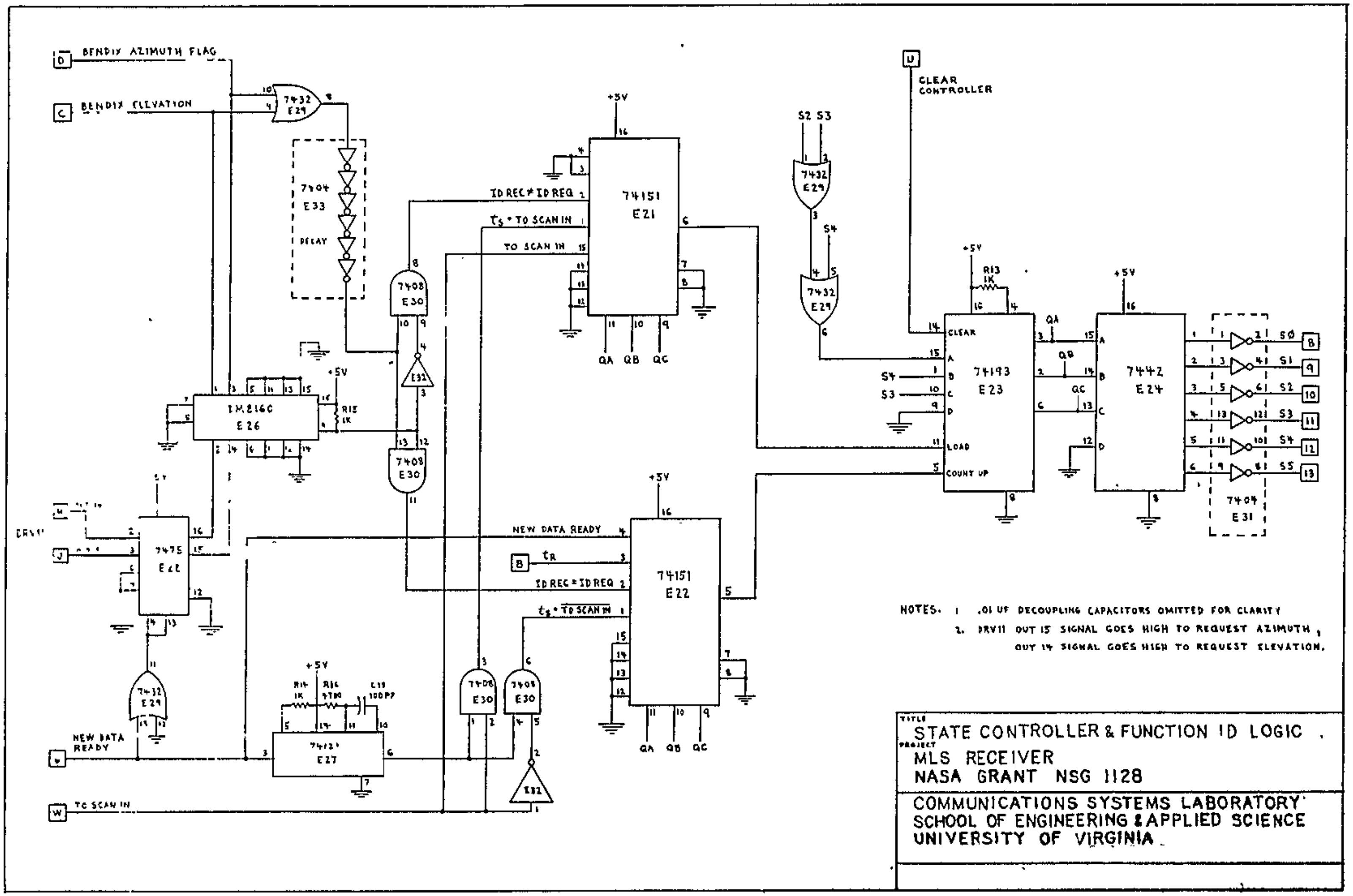


Figure B-4 State Controller and Function ID Logic

Signal	Origin
ID received = ID requested ID received ≠ ID requested	FUNCTION ID LOGIC; these signals are produced on the same board that contains the state controller. The Function ID logic is discussed below.
TO SCAN IN	SAMPLE TIMER & BUFFER MEMORY LOGIC; after 33 samples have been taken, this signal is raised to indicate that TO scan sampling is completed.
$t_s \cdot \overline{\text{TO SCAN IN}}$	STATE CONTROLLER & FUNCTION ID LOGIC; when the PDP sends out the <u>first</u> t_s word, the NEW DATA READY signal is pulsed to indicate that data is in the DRV 11 output buffer. The TO scan samples have not yet been taken, so $\overline{\text{TO SCAN IN}}$ is true. We use the trailing edge of the NEW DATA READY signal to fire a one-shot, whose output is logically ANDed with $\overline{\text{TO SCAN IN}}$ to produce $t_s \cdot \overline{\text{TO SCAN IN}}$. By using the trailing edge, we insure that the data is stabilized in the output buffer before we accept it as valid.
$t_s \cdot$ TO SCAN IN	Similar to above; however, when the PDP sends out the <u>second</u> t_s word, TO scan sampling has already been completed and TO SCAN IN is true. Thus, TO SCAN IN and $\overline{\text{TO SCAN IN}}$ are used by the hardware to differentiate between TO and FRO scans.

<u>Function ID Logic</u> (Figure B-4)

As mentioned in the quick overview, the function ID logic, as currently designed, will work only if Azimuth and Elevation scans are the only scans being transmitted from the airport. The logic works as follows. When making an ID request, the computer will use signals OUT 15 and OUT 14 of the DRV 11. OUT 15 will be raised to request Azimuth; OUT 14 for Elevation. The NEW DATA READY pulse associated with this output will strobe the request into a 7475 buffer latch. Now, when the BENDIX receiver decodes the scan function ID, either the Azimuth flag or the Elevation flag will be raised. A DM8160 comparator will compare the ID received with the ID requested. It takes approximately 20 nsec for the comparator output to become valid. In order to prevent premature generation of invalid ID RECEIVED ≠ ID REQUESTED or ID RECEIVED = ID

REQUESTED signals, the rising edge of either the Azimuth or Elevation flag is passed through a delay line and is used to gate the comparator output and its complement on to the state controller. Thus, both of the signals ID REQUESTED ≠ ID RECEIVED and ID REQUESTED = ID RECEIVED remain low until the comparator output is valid, at which time only one of the signals goes high. This is necessary for the correct operation of the state controller (see Figure B-5).

If the airport were to broadcast more than just Azimuth and Elevation scans, the hardware as currently designed would not work. The BENDIX receiver would have to send out a t_R pulse at the beginning of each scan as it decodes the BARKER code; it has no way of knowing if the scan to follow is Azimuth, Elevation, or otherwise. Upon receipt of the t_R pulse, the state controller would enter STATE 2. Now, however, the controller would be stuck if neither the Azimuth nor Elevation flags were raised, as neither of the signals ID REC = ID REQ or ID REC ≠ ID REQ would be enabled to take the controller to another state. There are two ways around this problem. One solution would be to have the Bendix receiver provide additional flag signals for the other scans and to expand on the current design using some of the extra inputs to the DM8160 comparator. The raising of any of the flag signals could be used to gate the comparator output on to the state controller. A second solution would be to obtain precise information about the delay between the output of the Bendix t_R pulse and the output of the Azimuth or Elevation flags. Since the t_R pulse starts a 3.84 MHz counter, logic could be designed around this counter to provide a pulse after this delay had elapsed. This pulse could be used to gate the comparator output on to the state controller.

Inputs: +5 Volts, GROUND, CLEAR CONTROLLER, TO SCAN IN, CLOCK
BENDIX t_R, Azimuth, Elevation
NEW DATA READY, OUT 14, OUT 15

Outputs: Sϕ, S1, S2, S3, S4, S5

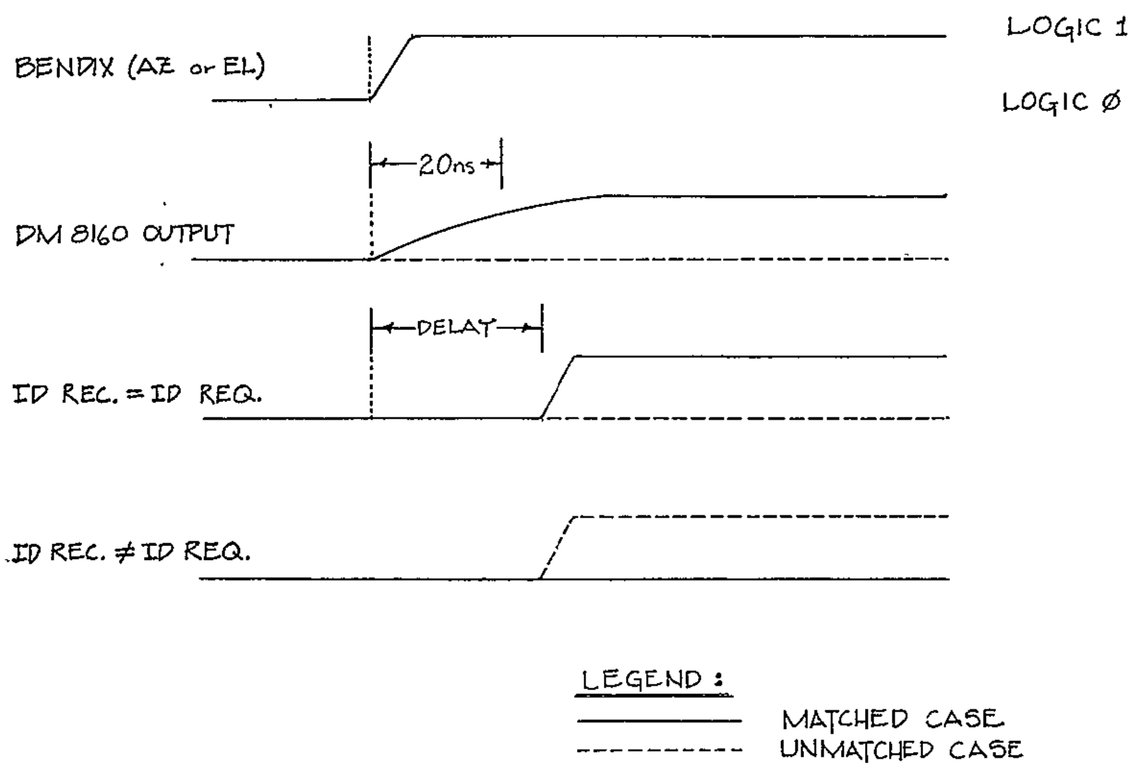


Figure B-5 Timing Diagram for Function ID Logic

Scan Timer (Figure B-6)

The purpose of the scan timer is to initiate sampling of the TO and FRO scans at the times specified by the 16-bit t_s words sent from the PDP. The scan timer uses four 7475 four-input latches to hold the t_s word. Four 7493 binary counters are wired to provide a 16-bit binary counter. The combination of a DM8130 10-bit comparator and a DM 8160 6-bit comparator will be used for a 16-bit comparison between the latches and counters. Cross-coupled NOR gates are used to implement classic Set-Reset flip-flops. Two flip-flops are used and will be referred to as the clock flip-flop and the comparator flip-flop. The clock flip-flop controls the flow of the 3.84 MHz clock signal to the binary counter. The comparator flip-flop controls the passage to the backplane of the ANDed comparator outputs (referred to as COMPARISON MATCH).

The scan timer will operate as follows. When the state controller enters STATE Ø, both flip-flops are reset. This will prevent the counter from being incremented and will disable the COMPARISON MATCH signal from reaching the backplane. When the controller enters STATE 1, the binary counter is cleared. Now, upon receipt of the BENDIX t_R pulse, the state controller enters STATE 2. This sets the clock flip-flop, and the binary counter begins counting at 3.84 MHz. If the ID received ≠ ID requested signal is produced by the function ID logic, the controller re-enters STATE 1; and the counter is again cleared. It will remain cleared as long as the controller sits in STATE 1, even though the 3.84 MHz clock signal still flows to the counter input. On the other hand, if the ID received = ID requested signal is produced, then the controller enters STATE 3. Note that the t_s word has not yet been loaded into the latches; therefore, any matching between the counter and the latches cannot be valid. So STATE 3 is used to reset the comparator flip-flop, preventing the COMPARISON MATCH signal from being raised.

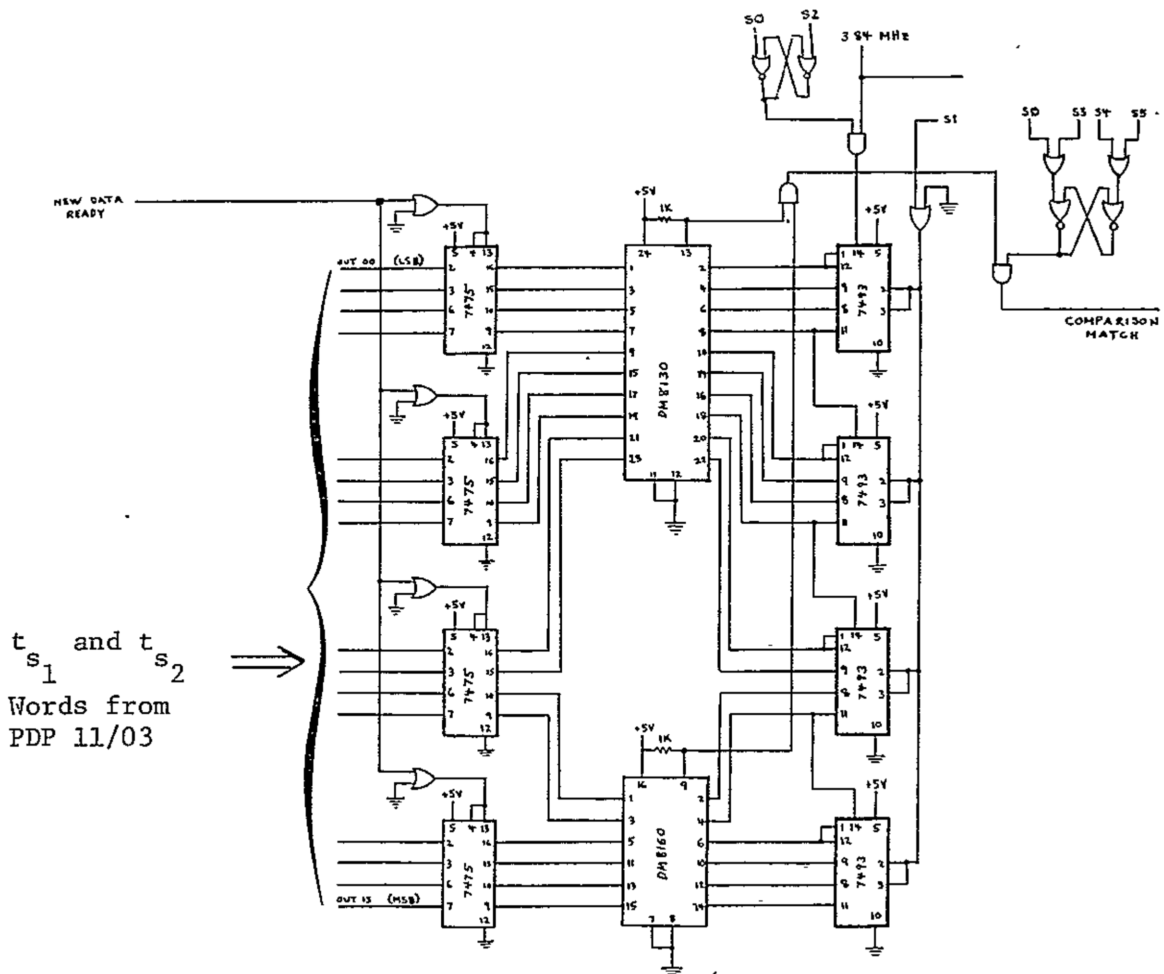


Figure B-6 Scan Timer Logic

STATE 3 is used to signal an interrupt request (REQUEST B) to the PDP. In response, the PDP will output a 16-bit t_s word which indicates when TO scan sampling should start. As mentioned in the quick overview, this t_s word will measure time from the t_R pulse but must be corrected for a constant delay time which will be discussed in the next section. The NEW DATA READY signal associated with the output of the t_s word is used to strobe the 16 bits into the 7475 latches. The trailing edge of the NEW DATA READY signal is used to generate the $t_s \cdot \overline{\text{TO SCAN IN}}$ signal which will make the controller enter STATE 4. At this point, the latches contain a valid t_s word; so STATE 4 is used to set the comparator flip-flop. Now, when the counter counts up to the t_s word, the comparator outputs go high. The comparator flip-flop is set; thus, the COMPARISON MATCH signal is raised. This signal is passed via the backplane to the SAMPLE TIMER & BUFFER MEMORY board and enables sampling to commence.

After 33 samples have been taken, the state controller re-enters STATE 3. The Request B line to the PDP is raised to request the FRO scan t_s word. Since the t_s word has not yet been received, STATE 3 resets the comparator flip-flop to prevent the COMPARISON MATCH signal from going high. When the t_s word is received, the $t_s \cdot$ TO SCAN IN signal will make the controller enter STATE 5. STATE 5 sets the comparator flip-flop to allow the COMPARISON MATCH signal to go high. Meanwhile, the counter continues to count at 3.84 MHz. When it counts up to the FRO scan t_s word, the comparator outputs again go high; and the COMPARISON MATCH signal is raised. This allows FRO scan sampling to commence. After the 67^{th} sample is taken, the controller returns to STATE Ø.

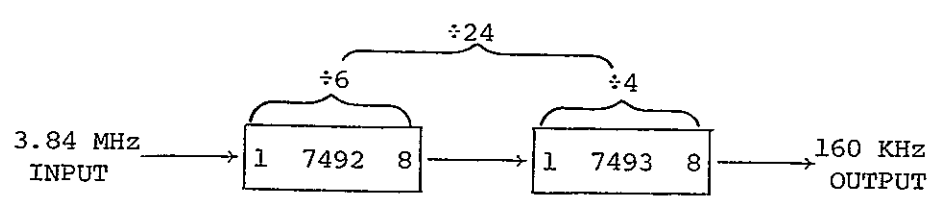
Inputs: +5 Volts, GROUND, OUT ØØ(LSB) through OUT 15(MSB), New Data Ready, CLOCK, SØ through S5

Output: COMPARISON MATCH

Sample Timer and Buffer Memory (Figure B-7)

The purpose of this board is to control the operation of the A/D and Sample/Hold board and to store the samples as they are taken so that they may be read back by the PDP 11/Ø3. Three Intel 2101 256 x 4 Static MOS RAM's provide the memory storage. Although only 66 samples with 10-bit resolution need to be stored, space is at a premium on the VERO boards and the 2101's provide the necessary storage with only three 22-pin I.C. packages. The samples are taken at a rate of 160 KHz. This frequency is obtained from the 3.84 MHz clock by a divide-by-24 counter which consists of the series combination of a 7492 counter and a 7493 counter. The memory address logic for 2101 RAM's is provided by an 8-bit binary counter consisting of two 7493 counters. These will be referred to as the address counters. Additional logic is wired to the outputs of the address counters to provide signals which go high when the counters point to the 33^{rd} and to the 67^{th} locations in memory. This additional logic consists of two 7421 dual 4-input AND gates along with some inverters and 7408 AND gates.

NOTE: As was mentioned previously, the t_s words sent by the computer to the SCAN TIMER must be adjusted for a constant delay factor. This delay is due to two primary sources. One source of delay is the divide-by-24 counter.



A detailed analysis of the operation of this counter, based on its internal logic, yields the following result: the counter provides

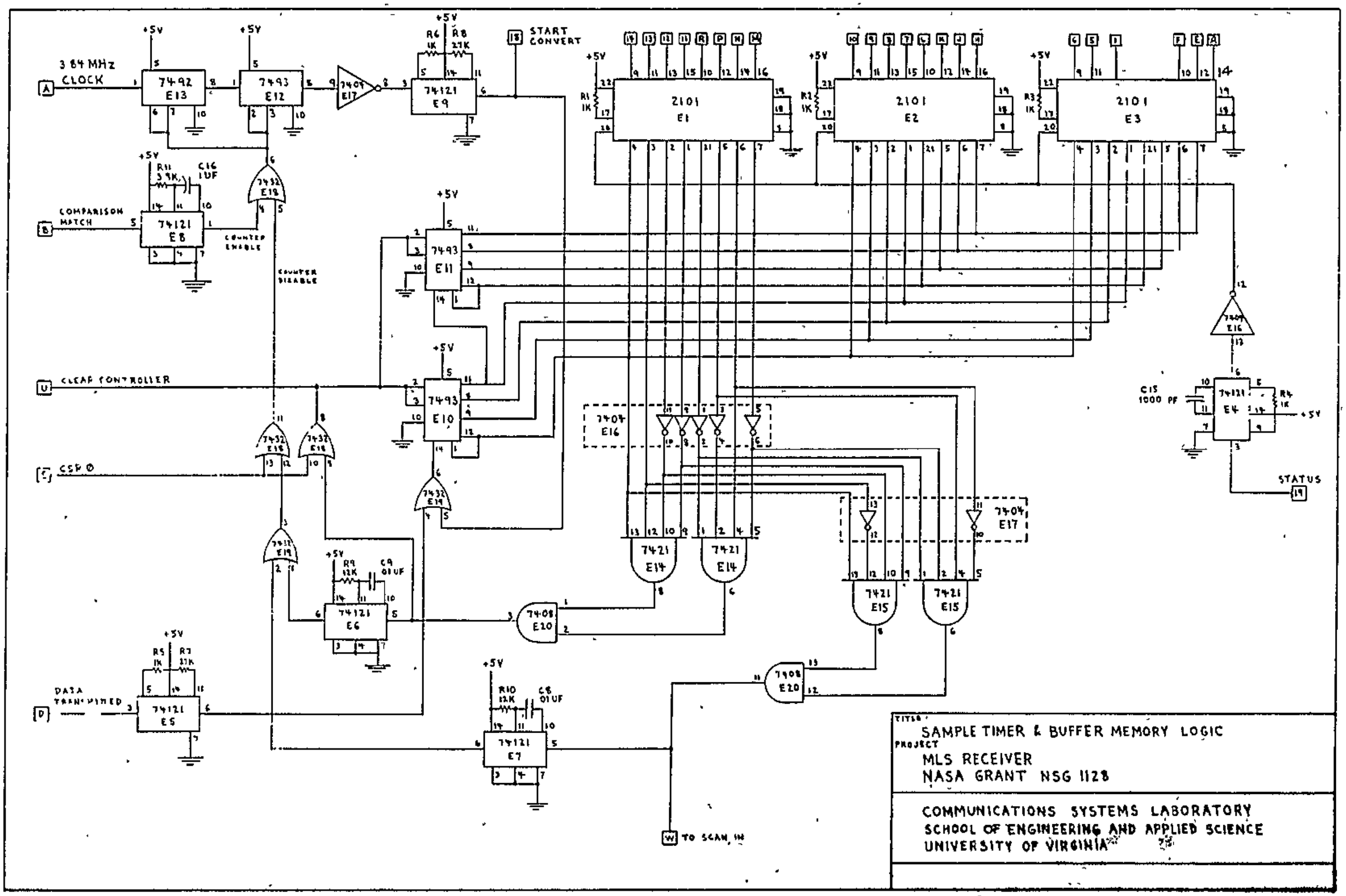


Figure B-7 Sample Timer and Buffer Memory the proper frequency division, but the output is shifted in phase. The first output pulse comes at the end (trailing edge) of nine input pulses. At 3.84 MHz, this delay is approximately 2.34 microseconds.

A second source of delay will be the Bendix receiver. The receiver is to output the leading edge of its t_R pulse no later than 100 microseconds after reception of Bit 5 of the five-bit Barker code. The actual delay will have to be obtained from Bendix. Note that this delay will be compensated for partially or wholly by the delay in transmission of the Log Video envelope.

Other delays associated with gating and signal propagation are measured in nanoseconds and are insignificant when compared to the sources of delay above. Thus, the total delay may be expressed as:

TOTAL DELAY=(COUNTER DELAY)+(BENDIX t_R DELAY)+(BENDIX Log Video DELAY)

This total delay must be subtracted from the t_s words which are calculated by the computer before they are sent to the SCAN TIMER board.

The sample timer and buffer memory board will operate as follows. During the initialization following power-up, the computer will raise and then lower the CSRØ signal of the DRV 11 Parallel Line Unit (DRCSR Bit Ø). This pulsing of CSRØ will do three things: (1) it will raise the CLEAR CONTROLLER signal, thus placing the state controller in STATE Ø; (2) it will clear the divide-by-24 counter; (3) it will clear the address counter so that it points to memory location zero of the buffer memory.

Now, STATE Ø is the state in which the computer reads out the samples from the previously sampled scan. It does this by repeatedly reading the contents of the DRV 11 input buffer, DRINBUF. After each read operation, the DRV 11 automatically pulses the DATA TRANSMITTED signal. This signal will be used by the hardware to increment the address counter so that the next sample may be read. Thus, to read out the 66 samples the computer must do the following. First, it must read DRINBUF once and ignore this sample. This sample, stored in buffer memory location zero, is actually the 67^{th} sample and that is why it is ignored. The 66 samples of interest are stored in Locations 1 through 66. So the computer now does 66 successive reads to obtain the 66 samples. The data is in the lower 10 bits of the 16-bit input buffer, with Bit Ø as the Least Significant Bit.

After the 66^{th} sample has been read (67 read operations), the hardware automatically clears the address counters in preparation for the next set of samples to be taken. The hardware also raises the CLEAR CONTROLLER signal (this has no effect here since the controller is already in STATE Ø) and fires a one-shot (the purpose of the one-shot will be explained later).

No more activity occurs on the board until the COMPARISON MATCH signal from the SCAN TIMER board is raised, signifying that sampling is to begin. The COMPARISON MATCH signal is fed into a one-shot (E8). The output of this one-shot, labeled COUNTER ENABLE, is normally high and is used to <u>prevent</u> the divide-by-24 counter from incrementing. When COMPARISON MATCH goes high, COUNTER ENABLE goes <u>low</u> for an interval somewhat longer than the time necessary to take 33 samples at 160 KHz. Thus, the divide-by-24 counter is allowed to start incrementing. Each output pulse of the counter fires a one-shot whose output is the START CONVERT signal. This signal causes the A/D and Sample/Hold board to take a sample and also increments the address counter so that it addresses the location in the buffer memory where the sample will be stored. When the A/D conversion is completed, the STATUS line goes low. This transition fires a one-shot (E4) which provides a low-true write pulse for the buffer memory. The 10-bit digitized sample is thus written into the buffer memory at the address indicated by the address counters.

When the 33^{rd} sample is taken, logic wired to the outputs of the address counters decodes the 33 and raises the TO SCAN IN signal. A one-shot (E7) is fired which raises the COUNTER DISABLE signal. This signal remains high for the duration of the TO scan and clears and disables the divide-by-24 counter, thus preventing any more samples from being taken during the scan. The raising of TO SCAN IN takes the state controller back to STATE 3.

The board is now inactive until the COMPARISON MATCH signal goes high, signifying that sampling for the FRO scan must begin. The COUNTER ENABLE signal goes low, and samples are taken and stored as before. However, during the FRO scan, 34 samples are taken, the last of which will be ignored. This is done so that the 66 samples of interest will be stored in consecutive memory locations, while still providing a self-clearing address counter. Thus, when the 67^{th} sample is taken, logic wired to the outputs of the address counters decodes the 67 and clears the address counter so that the 67^{th} sample gets written into memory location zero. A one-shot (E6) is fired whose output raises the COUNTER DISABLE line, thus preventing anymore samples from being taken during the FRO scan. The CLEAR CONTROLLER line is also raised, so the state controller is sent back to STATE Ø.

STATE Ø is used to signal the PDP 11/Ø3 with an interrupt (Request A). The PDP responds by reading out 67 samples (ignoring the first) and then by sending out the code for the next type of scan to be sampled.

Note: The Basic Wide/Basic Narrow option will be transmitted to the buffer memory from a front panel switch. The option will be stored in memory along with the data samples and may be checked by looking at Bit 15 of the input buffer after any of the data samples have been read.

Inputs: +5 V, GROUND, Bit Ø(LSB) through Bit 9(MSB), BW/BN, COMPARISON MATCH, CLOCK, STATUS CSR Ø, DATA TRANSMITTED

Outputs: START CONVERT, TO SCAN IN, CLEAR CONTROLLER, IN ØØ(LSB) through IN Ø9(MSB), IN 15

Backplane Wiring

The table on the following page gives terminal pin assignments for each of the VERO FINGER BOARDS which will be used to implement the functional units described above (Figure B-8).

II. Step-by-Step Operation

We may now summarize and clarify all of the previous discussion by giving a step-by-step description of the operation of the data acquisition hardware. The state transitions of the state controller will provide a convenient outline structure for the description.

Power Up

During the initialization following power-up, the computer must pulse the CSR Ø signal. On the sample timer and buffer memory board this pulse will clear the divide-by-24 counter and the address counter and will also raise the CLEAR CONTROLLER signal. The CLEAR CONTROLLER signal is transmitted to the state controller and function ID logic board, where it places the state controller in STATE Ø.

State Ø

The STATE Ø line is connected to the Request A line of the DRV 11; so

Pin	POWER SUPPLY	OSCILLATOR	A/D and Sample/Hold	State Controller & Function ID Logic	Scan Timer	Sample Timer & Buffer Memory
1						RW/RH
2	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts
3	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts
4	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts
5			Bit (MSB) (10 bits out to buffer memory)			Bit 9 (MSB) (10 bits in from A/D)
6			Bit	NEW DATA READY	NEW DATA READY	Bit 8
7			Bit			Bit 7
8			Bit	S0	S0	Bit 6
9			Bit	S1	S1	Bit 5
10			Bit	S2	S2	Bit 4
11			Bit	S3	S3	Bit 3
12			Bit	S4	S4	Bit 2
13			Bit	S5	S5	Bit 1
14			Bit (LSB)			Bit 0 (LSB)
15	GROUND	GROUND	GROUND	GROUND	GROUND	GROUND
16	GROUND	GROUND	GROUND	GROUND	GROUND	GROUND
17	GROUND	GROUND	GROUND	GROUND	GROUND	GROUND
18			START CONVERT			START CONVERT
19			STATUS			STATUS
20						
21		CLOCK		CLOCK	CLOCK	CLOCK
22						
A			Log Video (+)			IN 15 (RW/EH to PDP)
B				BENDIX t_R	COMPARISON MATCH	
C			Log Video (-)	BENDIX ELEVATION	OUT Ø0 (LSB) (t_s word from computer)	
D				BENDIX AZIMUTH	OUT Ø1	DATA TRANSMITTED
E					OUT Ø2	IN Ø9 (LSB) (10 bit sample out to PDP)
F	+15 Volts		+15 Volts		OUT Ø3	IN Ø8
H				OUT 14 (for ID requested)	OUT Ø4	IN Ø7
J				OUT 15 (for ID requested)	OUT Ø5	IN Ø6
K					OUT Ø6	IN Ø5
L					OUT Ø7	IN Ø4
M	COMMON		COMMON		OUT Ø8	IN Ø3
N					OUT Ø9	IN Ø2
P					OUT 10	IN Ø1
R					OUT 11	IN Ø0 (MSB)
S					OUT 12	LSSS
T					OUT 13	
U				CLEAR CONTROLLER	OUT 14	CLEAR CONTROLLER
V	-15 Volts		-15 Volts		OUT 15 (MSB)	
W				TO SCAN IN		TO SCAN IN
X						
Y						
Z						

Figure B-8 Terminal Pin Assignments

when STATE Ø occurs, an interrupt is generated. The STATE Ø signal also is transmitted to the scan timer board, where it resets the clock flip-flop and comparator flip-flop, thus preventing the COMPARISON MATCH signal from being raised.

Response to Interrupt Request A

The computer must respond to the Request A interrupt by reading its input buffer 67 times. The first read will be ignored and is used merely to increment the address counter so that it points to buffer memory location one, which holds the first sample. The DATA TRANSMITTED signal generated by each read operation is transmitted to the buffer memory to increment the address counter.

After reading in the samples, the computer will send out a code for the type of scan that it wants to have sampled. To request Azimuth the OUT 15 signal will be raised; to request Elevation the OUT 14 signal will be raised. The NEW DATA READY signal generated by this output will latch the ID requested on the STATE CONTROLLER & Function ID Logic board and will take the state controller to STATE 1.

State 1

The STATE 1 signal is transmitted to the scan timer board, where it clears a 16-bit binary counter which will operate at 3.84 MHz. Now, the controller waits until the Bendix receiver generates a t_R pulse. This pulse will take the controller to STATE 2.

State 2

The STATE 2 signal is transmitted to the scan timer board, where it sets the clock flip-flop. This makes the 16-bit binary counter begin to count at 3.84 MHz.

Meanwhile, the Bendix receiver will be decoding the function ID of the current scan. It will raise either the Azimuth or Elevation flag. The function ID logic, in turn, will generate either the signal "ID received ≠ ID requested" or the signal "ID received = ID requested." If ID rec ≠ ID req is generated, the state controller goes back to STATE 1, thus stopping and clearing the 16-bit counter. If ID rec = ID req is generated, the state controller goes to STATE 3.

<u>State 3</u>

The STATE 3 line is connected to the Request B line of the DRV 11; so when STATE 3 occurs, an interrupt is generated. The STATE 3 signal is also transmitted to the scan timer board, where it resets the comparator flip-flip, thus preventing the COMPARISON MATCH signal from being raised.

<u>Response to Interrupt Request B</u>

The computer must respond to the <u>first</u> Request B by sending out a 16-bit t_s word (with adjustments as noted before) which tells the hardware when to begin TO scan sampling. In response to the <u>second</u> Request B (STATE 3 is generated twice), the computer must send out a 16-bit t_s word which tells the hardware when to begin FRO scan sampling.

If the TO scan has not been sampled, then the TO SCAN IN signal from the sample timer board will be low. The state controller board will use the trailing edge of the NEW DATA READY signal assocaited with the first t_s word output to generate the signal $t_s \cdot \overline{\text{TO SCAN IN}}$; this signal will take the controller to STATE 4.

On the other hand, if the TO scan has already been sampled, then the state controller board will generate the signal $t_s \cdot$ TO SCAN IN; this signal will take the controller to STATE 5.

State 4

The STATE 4 signal is transmitted to the scan timer board, where it sets the comparator flip-flop, thus allowing the COMPARISON MATCH signal to go high when the 16-bit counter counts up to the TO scan t_s word. When the COMPARISON MATCH signal is raised, sampling begins as described in Section I. After 33 samples have been taken, the TO SCAN IN signal from the sample timer board is raised. This signal takes the state controller back to STATE 3.

State 5

The STATE 5 signal is transmitted to the scan timer board, where it sets the comparator flip-flop, thus allowing the COMPARISON MATCH signal to go high when the 16-bit counter counts up to the FRO scan t_s word. When COMPARISON MATCH is raised, sampling begins as described in Section I. After 34 additional samples have been taken, the sample timer board raises the CLEAR CONTROLLER signal, which takes the state controller back to STATE Ø.

Test Program

The computer program used to test the completed data acquisition system is shown in Figure B-9.

DATA INTERFACE
WITH CONTROLLER
TEST
5/6/76

```
.MAIN.  RT-11 MACRO VM02-12    PAGE 1

        000000          .ASECT
        167770 DRCSR    =       167770
        167772 OUTBUF   =       167772
        167774 INBUF    =       167774
                        .MCALL  .REGDEF
  000000                .REGDEF
        001000 .        =       1000
  001000 000005         RESET                   ;INITIALIZE
  001002 012706         MOV     #1000,SP
         001000
  01006 012767          MOV     #1,DRCSR
         000001
         166754
  01014 012767          MOV     #0,DRCSR
         000000
         166746
  01022 032767 A:       BIT     #200,DRCSR      ;WAIT FOR REQ A
         000200
         166740
  01030 001774          BEQ     A
  01032 016700          MOV     INBUF,R0        ;BRING IN SAMPLES
         166736
  01036 012700          MOV     #BUFFER,R0
         002000
  01042 016720 B:       MOV     INBUF,(R0)+
         166726
  01046 022700          CMP     #BUFEND,R0
         002204
  01052 001373          BNE     B
  01054 000000          HALT
  01056 016767 C:       MOV     FUNC,OUTBUF     ;OUTPUT FUNCTION
         001716
         166706
  01064 032767 D:       BIT     #100000,DRCSR   ;WAIT FOR REQ B
         100000
         166676
  01072 001774          BEQ     D
  01074 016767          MOV     TS1,OUTBUF      ;OUTPUT TS1
         001702
         166670
  01102 032767 E:       BIT     #100000,DRCSR   ;WAIT FOR REQ B
         100000
         166660
  01110 001774          BEQ     E
  01112 016767          MOV     TS2,OUTBUF      ;OUTPUT TS2
         001666
         166652
  01120 000740          BR      A               ;REPEAT
        002000 .        =       2000
  02000        BUFFER:  .BLKW   ^D66            ;DEFINE 66 WORD BLOCK
  02204        BUFEND:
        003000 .        =       3000
  03000 000000 FUNC:    .WORD   0               ;DEFINE PARAMETERS
  03002 000000 TS1:     .WORD   0
  03004 000000 TS2:     .WORD   0
        000001'         .END
```

TS1 - 4 msec ≃ 1544 ϕ_{10} = 3612ϕ_{γ}

TS3 = 11 msec ≃ 4246ϕ_{10} = 12273$|_{\gamma}$

```
SYMBOL TABLE

A       001022          B       001042          BUFEND  002204
BUFFER  002000          C       001056          D       001064
DRCSR = 167770          E       001102          FUNC    003000
INBUF = 167774          OUTBUF= 167772          PC    =%000007
R0    =%000000          R1    =%000001          R2    =%000002
R3    =%000003          R4    =%000004          R5    =%000005
SP    =%000006          TS1     003002          TS2     003004

. ABS.  003006     000
        000000     001
ERRORS DETECTED: 0
FREE CORE: 1707. WORDS

MLSTES,MLSTES=MLSTES

?*EOF*?
*
```

Figure B-9 Data Acquisition System Test Program

APPENDIX C

A 6D LOE OPTIMAL RECEIVER

In this design, a 6th component, $\hat{\omega}_{sc}$, is appended to the definition of $\hat{\gamma}$, requiring basically only two changes:

1. Appending a 6th row to matrix $D(\hat{\gamma})$ so that instead of (2.103) it now reads

$$D(\hat{\gamma}) = \begin{pmatrix} --- & 2\hat{\alpha}p_j^2(\hat{\theta}) + 2\hat{\alpha}_R p_j(\hat{\theta})p_j(\hat{\theta}_R)\cos\hat{\beta}_j & --- \\ --- & 2\hat{\alpha}^2 p_j(\hat{\theta})\dot{p}_j(\hat{\theta}) + 2\hat{\alpha}\hat{\alpha}_R\dot{p}_j(\hat{\theta})p_j(\hat{\theta}_R)\cos\hat{\beta}_j & --- \\ --- & 2\hat{\alpha}p_j(\hat{\theta})p_j(\hat{\theta}_R)\cos\hat{\beta}_j + 2\hat{\alpha}_R p^2(\hat{\theta}_R) & --- \\ --- & 2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta})\dot{p}_j(\hat{\theta}_R)\cos\hat{\beta}_j + 2\hat{\alpha}_R^2\dot{p}_j(\hat{\theta}_R)p_j(\hat{\theta}) & --- \\ --- & (-2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta})p_j(\hat{\theta}_R)\sin\hat{\beta}_j) & --- \\ --- & (-2\hat{\alpha}\hat{\alpha}_R p_j(\hat{\theta})p_j(\hat{\theta}_R)\tau_j\sin\hat{\beta}_j) & --- \end{pmatrix} \quad \text{(C-1)}$$

where, as indicated in (3.10a), the quantity

$$\hat{\beta}_j = \hat{\beta} + \hat{\omega}_{sc}\,\tau_j \quad \text{(C-2)}$$

is one which has already been in use in the SD LOE (see PLOPT, Appendix A)(in spite of the assumption (2.90) used in the derivation of the LOE that all the β_j are equal); and

2. Changing the dimensions of all affected arrays in the programs.

Both changes affect module PLOPT, the latter additionally affects modules MLSSIM, OPTRVR and CTLACQN, the (interference acquisition) version of CONTRL which was selected for this study. Listings of the modified versions of these modules, CTLACQ6, MLSIM6, OPRVR6 and PLOPT6 follow below. Two options of this basic design were provided in the programs.

1. $N_s = 8$, and the state vector x is as defined for the Optimal design;

2. $N_s = 9$, and an additional component, $\dot{\omega}_{sc}$, is appended to the end of the state vector definition.

It was not possible to run any simulation studies involving the 6D LOE, the computing resources having practically been exhausted at this point (including the $4700 University grant); as a debug exercise, however, the programs, by parameter inputs, were specialized to the prior SD LOE OPTRVR design and an earlier interference acquisition run with OPTRVR duplicated.

```
      SUBROUTINE CONTRL(ISW)
C
C  THIS CONDUCTS THE MLS SIMULATION THROUGH A
C INTERFERENCE ACQUISITION SCENARIO,6D LOE
C
      REAL LAMDA
      INTEGER XNAME(2),YNAME(2),DATIN(4),DATOUT(4),DOUT
      INTEGER YNAM1(2),YNAM2(2),YNAM3(2),YNAM4(2),YNAM5(2),FUNCTN(2)
      INTEGER YNAM7(2),YNAM8(2),YNAM9(2),STITLE(9)
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE,NFIRST,ADAPTV
      LOGICAL FILOUT,FILEIN
      COMMON/IDDATA/ISIM,IPMLS,IRCVR,IADAP,ITETHR,IPOPT
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(6),NGO(6)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GQGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(9),ESLOE(9),ES(9)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(9),XS(9)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC,WSCDOT
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(9,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DIMENSION IDNRS(6),IDASCI(24),IDENTS(6),RMSVAL(9)
      DIMENSION EALFA(100),ETHET(100),EALFR(100),ETHER(100),XFOUR(100)
      DIMENSION             X(9),            SCANNR(100)
      DIMENSION EBETA(100),EFSC(100)
      DIMENSION EFSCDT(100)
      EQUIVALENCE (ALFA,X(1)),(ISIM,IDNRS(1)),(DOUT,DATOUT(2))
      DATA ABORT/1HA/,SPACE/1H /,NFIRST/.FALSE./,ADAPTV/.TRUE./
      DATA DATIN/10HMLSSIMDATA,10H500000F000,2*0/
      DATA DATOUT/10HMLSSIMDATA,3*0/
      DATA IDASCI/7HCROSSMP,7HRMSE(T),7HRMSE(B),7HRMSE(F),7HACQSITN,
     *3*7H       ,7HGENERAL,8H PMLS1  ,8H PMLS2  ,
     *8H PMLS3  ,8H THRHLD ,8H OPTIML ,8H SUBOPT ,8H -3 DB  ,8HADAPTIV ,
     *8HNONADAP ,8HUNTETHRD,8HTETHERED,2H  ,8H POPT1  ,8H POPT2  ,
     *8H POPT3  /
      DATA NSTART/3/,IRSIGN/1/,KSTART/26/,FILOUT/.FALSE./,
     CFILEIN/.FALSE./
      DATA FUNCTN/7HAZIMUTH,7HELEVAT./
      DATA GQGT88/7.3/,GQGT99/1.E4/
      DATA STITLE/6H ALFA ,6HTHETA ,6HTHEDOT,6HALFAR ,6HTHETAR,
     C6HTHRDOT,6H  B   ,6H WSC  ,6HWSCDOT/
      DATA XNAME/8H  SCAN  ,8H NUMBER /,YNAM2/8H  ETHET ,8HDEGREES /
      DATA YNAM1/8H  EALFA ,8H        /,YNAM3/8H EALFAR ,8H        /
      DATA YNAM4/8H  ETHER ,8HDEGREES /,YNAM5/8H  RHO   ,8H        /
      DATA YNAM7/8HEABSBETA,8H DEGREES/
      DATA YNAM8/8HEABSFSC ,8H  HERTZ /
      DATA YNAM9/8HEABFSCDT,8H HZ/SEC /
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200),ISW
    1 FORMAT(1H )
```

C-3

```
        10    FORMAT(9(/))
        11    FORMAT (1H1)
        C
        C*********************************************************************
        C
        100   CONTINUE
        C THIS IDENTIFIES THE SCENARIO AND REINITIALIZES, AS NECESSARY,
        CTHE FOLLOWING RECEIVER PARAMETERS
        CSIGMA,IAE,DSNRDB,RHO,FSC,BETA,JM,BMLS
              IF(NFIRST) GO TO 155
              NOAC=.FALSE.
              NFIRST=.TRUE.
              JBNAM=JOBNAME(I)
              ISIM=5
              IMOD=4
              WRITE (7,110)
        110   FORMAT(34H1INTERFERENCE ACQUISITION SCENARIO/7H 6D LOE/)
              IF (IRCVR.NE.1) GO TO 118
              WRITE (7,115)
        115   FORMAT (35H THIS IS NOT FOR THRESHOLD RECEIVER)
              STOP ≠ ABORTED--NOT FOR THDRVR≠
        118   CONTINUE
              CALL LOGIO(6HFILEIN,FILEIN,0)
              IF(.NOT.FILEIN) NSTART=0
              IF(.NOT.FILEIN) GO TO 122
              IX=IATTACH(6LTAPE15,DATIN)
              IF(IX.NE.0) CALL INTIO(6HIX(AT),IX,1,1)
              IF(IX.NE.0) STOP≠INPUT FILE ATTACH NOT SATISFACTORY≠
              WRITE(7,120) (DATIN(I),I=1,2)
        120   FORMAT(23H THIS READS INPUT FILE ,2A10/)
        122   CALL LOGIO(6HFILOUT,FILOUT,0)
              CALL INTIO(6HNSTART,NSTART,100,0)
              NSTOP=NSTART
              CALL INTIO(6H NSTOP,NSTOP,100,0)
              CALL LOGIO(6HTETHRD,TETHRD,0)
              CALL LOGIO(6HADAPTV,ADAPTV,0)
              IF(TETHRD) ITETHR=2
              IF(ADAPTV) GO TO 135
              IRSIGN=-1
              IADAP=3
        135   CONTINUE
              DO 145 I=2,6
              IASC=3*(I-1)+IDNRS(I)+6
              IDENTS(I)=IDASCI(IASC)
        145   CONTINUE
              ENCODE(8,150,IDENTS(1)) IDASCI(ISIM),IMOD
        150   FORMAT(A7,I1)
              WRITE (7,152) IDNRS,IMOD
        152   FORMAT (1H ,6I1,1X,I1/)
              WRITE (7,153) IDENTS
        153   FORMAT (1H ,A8)
              CALL INTIO(6H    IR,IR,6,0)
              CALL REALIO (6HRHOMAX,RHOMAX,0)
              CALL REALIO(6H  DTHO,DTHO,0)
              CALL REALIO(6H  TORO,TORO,0)
              BETAO=BO*180./PI
              FSCO=WSCO*1./(2.*PI)
```

```
 NOTE      *         MULTIPLY BY 1 -- IGNORED
                              CALL REALIO(6H BETAO,BETAO,0)
                              CALL REALIO(6H  FSCO,FSCO,0)
                              IF (IR.GE.5) CALL REALIO(6HGQGT88,GQGT88,0)
                              IF (IR.GE.6) CALL REALIO (6HGQGT99,GQGT99,0)
                              B0=BETAO*PI/180.
                              WSCO=FSCO*2*PI
                              NRUN=NSTART-1
                              DSNRDB=20.
                              RHO=0.5
                              BETA=45.
                              FSC=0.
                              BMLS=1.
                              BRCVR=1.
                              THESEP=1.88
                              FSCDOT=XO(9,IAE)*.5/PI
                     155      CONTINUE
                              IF (FILEIN) GO TO 158
                              NRUN=NRUN+1
                              WRITE(7,11)
                              CALL INTIO (6H  NRUN,NRUN,1,1)
                              CALL REALIO(6HDSNRDB,DSNRDB,0)
                              CALL REALIO(6H   RHO,RHO,0)
                              CALL REALIO(6H  BETA,BETA,0)
                              CALL REALIO(6H   FSC,FSC,0)
                              IF(IR.EQ.6) CALL REALIO(6HFSCDOT,FSCDOT,0)
                              CALL REALIO(6H  BMLS,BMLS,0)
                              CALL REALIO(6H BRCVR,BRCVR,0)
                              CALL REALIO(6HTHESEP,THESEP,0)
                              CALL INTIO(6HKSTART,KSTART,100,0)
                              GO TO 170
                     158      CONTINUE
                              READ(15,160) NRUN,DSNRDB,RHO,BETA,FSC,FSCDOT,BMLS,BRCVR,
                             CTHESEP,KSTART
                              IF(EOF(15)) 180,165
                     160      FORMAT (I5,8G10.3,I5)
                     165      IF(NRUN.LT.NSTART) GO TO 158
                              IF(NRUN.GE.NSTOP) MORE=.FALSE.
                     170      IF(NRUN.GE.NSTOP) MORE=.FALSE.
                              XO(3,IAE)=XO(6,IAE)=0.
                              XO(5,IAE)=XO(2,IAE)-THESEP
                              XO(9,IAE)=FSCDOT*(2.*PI)
                              EBETAO=ABS(BETA)-ABS(BETAO)
                              EFSCO=ABS(FSC)-ABS(FSCO)
                              RETURN
                     180      STOP*EOF REACHED ON INPUT FILE*
                     C
                     C*************************************************************************
                     C
                     200      CONTINUE
                     C THIS OUTPUTS BASIC SIMULATION DATA OF INTEREST, SUCH AS
                     CTHE ANGLE FUNCTION, INITIAL STATE, ETC.
                              WRITE (7,209)
                     209      FORMAT(5H1NRUN,4X,6HDSNRDB,7X,3HRHO,8X,4HBETA,8X,3HFSC,7X,6HFSCDOT
                             C,8X,4HBMLS,8X,5HBRCVR,6X,6HTHESEP,5X,6HKSTART)
                              WRITE(7,210) NRUN,DSNRDB,RHO,BETA,FSC,FSCDOT,BMLS,
                             CBRCVR,THESEP,KSTART
```

```
  210   FORMAT(1H0,I4,8(2X,G10.3),2X,I4)
        IF(.NOT.FILOUT) GO TO 245
        IF(NRUN.LE.9) ENCODE(10,220,DOUT) IONRS,1HU,IMOD,1H0,NRUN
  220   FORMAT(6I1,A1,I1,A1,I1)
        IF(NRUN.GE.10) ENCODE(10,230,DOUT) IONRS,1HU,IMOD,NRUN
  230   FORMAT(6I1,A1,I1,I2)
        IX=IREQST(6LTAPE17,3L*PF)
        IF(IX.NE.0) CALL INTIO(6HIX(RQ),IX,1,1)
        IF(IX.NE.0) STOP*OUTPUT FILE REQUEST NOT SATISFACTORY*
        WRITE(7,240) (DATOUT(I),I=1,2)
  240   FORMAT(25H0THIS WRITES OUTPUT FILE ,2A10)
  245   CALL DATE(TODAY)
        WRITE (7,250) FUNCTN (IAE),IAE
  250   FORMAT (1H0,A7,15H FUNCTION (IAE=,I1,1H))
        X(4)=X0(4,IAE)=0.
        WRITE (7,260) ((I,X(I)),I=1,9)
  260   FORMAT (15H0INITIAL STATE://(3H X(,I1,4H) = ,G13.6))
        WRITE(7,270) IDENTS(2)
  270   FORMAT(1H0,A7)
        RETURN
C
C*********************************************************************
C
  300   CONTINUE
C THIS IDENTIFIES THE BASIC RECEIVER STRUCTURE (THRHLD,OPT,SUBOPT)
CAND REINITIALIZES, AS NECESSARY, THE FOLLOWING RCVR DATA
CIR (NEGATE ONLY), NOAC,NOKLMN,NOLOE,BRCVR,DELBL,TETHRD
        IF(IR.GE.5) GQGT(8,8)=GQGT88
        IF(IR.GE.6) GQGT(9,9)=GQGT99
        ITIT=MINO(IRCVR,2)
C
C NEGATE IR WITH THE FOLLOWING FOR NONADAPTIVE RECEIVER
        IR=ISIGN(IR,IRSIGN)
C
        RETURN
C
C*********************************************************************
C
  400   CONTINUE
C THIS OUTPUTS BASIC RECEIVER DATA OF INTEREST
        WRITE(7,410) (IDENTS(I),I=3,4),IR,NG,NS
  410   FORMAT (1H0,A8,4HRCVR/1H0,A8,6HDESIGN/5H (IR=,I2,4H,NG=,I1,
       *                  4H,NS=,I1,1H))
        WRITE(7,420) (IDENTS(I),I=5,6)
  420   FORMAT(1H0,A8/1H0,A7/)
        RETURN
C
C*********************************************************************
C
  500   CONTINUE
C THIS SETS-UP THE MSET-LOOP
        RETURN
C
C*********************************************************************
C
  600   CONTINUE
C THIS SETS-UP THE LG-LOOP FOR THE (MSET)-TH SERIES OF SETS
```

```
      RETURN
C
C***********************************************************************
C
700   CONTINUE
C THIS SETS-UP THE K-LOOP FOR THE (LG)-TH SET OF KM SCANS
      CALL INTIO(6H    KM,KM,1,1)
      WRITE(7,710) (STITLE(I),I=1,9)
710   FORMAT(4H0 K ,3X,6HCSNRTO,5X,6HCSNRFR,3X,3HQTY,3X,9(2X,A6,3X)/)
      RETURN
C
C***********************************************************************
C
800   CONTINUE
C THIS INITIALIZES THE K-TH SCAN
      IF(K.EQ.KSTART) X(4)=RHO*X(1)
      IF(K.EQ.KSTART) NGM=NG
      RETURN
C
C***********************************************************************
C
900   CONTINUE
C THIS SAVES/PREPROCESSES DATA FROM THE K-TH SCAN
      THERR=X(2)-XS(2)
      DTHE=X(2)-X(5)
      COL8=SQRT(PPDIAG(2))
      DO 910  I=1,NS
      ES(I)=X(I)-XS(I)
      RMSVAL(I)=SQRT(PPDIAG(I))
910   CONTINUE
      IF (NS.GE.7) ES(7) = ABS(X(7))-ABS(XS(7))
      IF (NS.GE.8) ES(8) = ABS(X(8))-ABS(XS(8))
      IF(NS.GE.9) ES(9)=ABS(X(9))-ABS(XS(9))
      SCANNR(K)=K
      EALFA(K)=ES(1)
      ETHET(K)=ES(2)
      EALFR(K)=ES(4)
      ETHER(K)=ES(5)
      XFOUR(K)=X(4)/X(1)
      IF(NS.GE.7) EBETA(K)=180.*ES(7)/PI
      IF(NS.GE.8) EFSC(K)=0.5*ES(8)/PI
      IF(NS.GE.9) EFSCDT(K)=0.5*ES(9)/PI
      IF(NRUN.NE.NSTART.AND.K.NE.KM) RETURN
      WRITE (7,920) K,     CSNRT,CSNRF,1HX,(X(I),I=1,9)
920   FORMAT(1H ,I3,2(1X,G10.3),8X,A1,1X,9(1X,G10.3))
      WRITE (7,930) 7H     XS,(XS(I),I=1,NS)
      WRITE (7,930) 7H    ERR,(ES(I),I=1,NS)
      WRITE (7,930) 7HSQ(PII),(RMSVAL(I),I=1,NS)
930   FORMAT (1H ,27X,A7,1X,9(1X,G10.3))
      WRITE (7,1)
      RETURN
C
C***********************************************************************
C
1000  CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (LG)-TH SET OF KM SCANS
      WRITE(7,1060) (GQGT(I,I),I=1,NS)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
 1060 FORMAT(10(/),18H ON THE LAST SCAN:,7X,9HGQGT(I,I),1X,9(1X,G10.3))
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR, EALFA,KM,XNAME,YNAM1,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR, ETHET,KM,XNAME,YNAM2,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR, EALFR,KM,XNAME,YNAM3,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR, ETHER,KM,XNAME,YNAM4,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR, XFOUR,KM,XNAME,YNAM5,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      IF (NS.LE.6) GO TO 1065
      WRITE (7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      YMIN=YMAX=0.
      CALL PLOTR(SCANNR,EBETA,KM,XNAME,YNAM7,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      YMIN=YMAX=0.
      WRITE(7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR,EFSC,KM,XNAME,YNAM8,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
      IF(NS.LE.8) GO TO 1065
      YMIN=YMAX=0.
      WRITE(7,11)
      CALL INTIO (6H  NRUN,NRUN,1,1)
      CALL PLOTR(SCANNR,EFSCDT,KM,XNAME,YNAM9,YMIN,YMAX,0)
      CALL REALIO(6H  YMIN,YMIN,1)
      CALL REALIO(6H  YMAX,YMAX,1)
 1065 CONTINUE
      IF(.NOT.FILOUT) RETURN
      WRITE(7,11)
      WRITE(7,1070) (DATOUT(I),I=1,2)
 1070 FORMAT(13H OUTPUT FILE ,2A10,1H:/)
      WRITE(7,1072) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
```

```
 1072 FORMAT(1H ,A10,8X,G13.6,5X,3(5X,I3,10X),A10,8X,A10/)
      WRITE(7,1074) IDENTS,KSTART
 1074 FORMAT(1H ,6(1X,A8,9X),I3/)
      WRITE(7,1076) NRUN,DSNRDB,RHO,BETA,FSC,FSCDOT,BMLS,BRCVR,THESEP
 1076 FORMAT(1H ,I2,3X,8(3X,G12.6)/)
      IKM=MINO(35,KM-9)
      DO 1077 I=1,IKM
 1077 WRITE(7,1078)EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
 1078 FORMAT(1H ,5(G13.6,5X))
      DO 1079 I=1,5
 1079 WRITE(7,1080)
 1080 FORMAT(1H,5(6X,1H.,11X))
      KM3=KM-3
      DO 1081 I=KM3,KM
 1081 WRITE(7,1078)EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
      WRITE(7,1083)RHOMAX,OTHO,TDRO,BETAO,FSCO,EBETAO,EBETA(KM),EFSCO,
     CEFSC(KM)
 1083 FORMAT (1HC,9(G12.6,2X))
      WRITE(17) DOUT,DELT,MTIMES,LGMAX,KM,TODAY,JBNAM
      WRITE(17) IDENTS,KSTART
      WRITE(17) NRUN,DSNRDB,RHO,BETA,FSC,FSCDOT,BMLS,BRCVR,THESEP
      DO 1090 I=1,KM
      WRITE(17) EALFA(I),ETHET(I),EALFR(I),ETHER(I),XFOUR(I)
 1090 CONTINUE
      WRITE(17) RHOMAX,OTHO,TDRO,BETAO,FSCO,EBETAO,EBETA(KM),EFSCO,EFSC(
     CKM)
      REWIND 17
      IX=IATTACH(6LTAPE20,DATOUT)
      CALL RETURN(6LTAPE20)
      IF(IX.NE.0) GO TO 1095
      IX=ICATALO(6LTAPE17,DATOUT,2LPW,8RHIGHFILL,2LRP,365)
      GO TO 1096
 1095 IX=ICATALO(6LTAPE17,DATOUT,2LXR,8RHIGHFILL,2LRP,365)
 1096 IF(IX.NE.0) CALL INTIO(6HIX(CA),IX,1,1)
      IF(IX.NE.0) STOP*OUTPUT FILE CATALOG NOT SATISFACTORY*
      CALL RETURN(6LTAPE17)
      WRITE (7,1099) (DATOUT(I),I=1,2)
 1099 FORMAT(13HOOUTPUT FILE ,2A10,33H IS WRITTEN, CATALOGED AND CLOSED)
      RETURN
C
C*********************************************************************
C
 1100 CONTINUE
C THIS SAVES/PROCESSES/OUTPUTS DATA FROM THE (MSET)-TH SERIES OF
CLGMAX SETS OF KM SCANS
      RETURN
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
 1200 CONTINUE
C THIS EFFECTS CLOSURE OF THE SIMULATION RUN
      WRITE (7,11)
      RETURN
C
C*********************************************************************
C
      END
```

45000B CM STORAGE USED 11.349 SECONDS

```
      PROGRAM MLSSIM(INPUT,OUTPUT,TAPE10=INPUT,TAPE12,TAPE15,
     *TAPE7=OUTPUT,TAPE17,TAPE20)
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE
      COMMON /IDDATA/IDNRS(6)
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(6),NGO(6)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHPD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GOGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(9),ESLOE(9),ES(9)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(9),XS(9)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC,WSCDOT
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XD(9,2),YD(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
C
   10 CONTINUE
      CALL MLSSUB
      IF(MORE) GO TO 10
      STOP
      END
```

41000B CM STORAGE USED   .131 SECONDS

C-11

```
      SUBROUTINE MLSSUB
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHG,TDRO,BO,WSCO,NSO(4),NGO(4)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GQGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(9),ESLOE(9),ES(9)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(9),XS(9)
      COMMON/MLS000/ALFA,THE,THEDOT,ALFAR,THR,THRDOT,B,WSC,WSCDOT
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS002/DCSNR,CSNR,LG,TPKT,TPKF
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(9,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
C GENERAL SIMULATION:  OPTIMIZATION OF MLS RECEIVERS FOR
C        MULTIPATH ENVIRONMENTS
      DIMENSION X1(9),INDEX(2),Y(4),X(9)
      EQUIVALENCE (THAMAX,Y(1)),(ALFA,X(1))
C
C  FOLG BEGINS EXECUTABLE STATEMENTS
C
      PI2=2.*PI
      SQ22=SQRT(.5)
C
C     FOLG INITIALIZES THE DIAGONAL OF F(8,8)
C
9     DO 10 I=1,9
      F(I,I)=1.
10    CONTINUE
      CALL CONTRL(1)
      SSQ2=SIGMA*SQRT(2.)
C
C      FOLLOWING  COMPLETES INITIALIZATION AND COMPUTES SECONDARY
C      PARAMETERS.
C
      DO 15 I=1,4
      FL10(I)=FL10AE(I,IAE)
15    CONTINUE
      DO 20 I=1,4
      Y(I)=YO(I,IAE)
20    CONTINUE
      OMEGA=-(THAMAX-THAMIN)/TS
      SECPD=1./OMEGA
      TF=TS+TR-2.*THAMIN*SECPD
C
C FOLG ARE CONSTANT PARAMETERS USED BY P
C
      BBB=2.4/BMLS
      DUM=GAUSS(1.0)
      ALPHA=10.**(DSNRDB/20.)
      XO(1,IAE)=ALPHA
      ALPHAR=RHO*ALPHA
```

```
      XO(4,IAE)=ALPHAR
      XO(7,IAE)=PI*BETA/180.
      XO(8,IAE)=PI2*FSC
      TSAMP=1./FSAMP
      DELT=DELTAT(IAE)
      F(2,3)=DELT
      F(5,6)=DELT
      F(7,8)=DELT
      F(7,9)=0.5*DELT**2
      F(8,9)=DELT
      JM1=JM+1
      JM2=JM/2
      TWW2=0.5*TSAMP*(JM2+1)
      DO 30 I=1,9
      X(I)=XO(I,IAE)
   30 CONTINUE
      CALL CONTRL(2)
C
C FOLG CALLS RECEIVER FOR INITIALIZATION
C
      K=0
      CALL RCVR
C
C FOLLOWING BEGINS SIMULATION PER SE, LGMAX RUNS OF KM SCANS EACH
C
      CALL CONTRL(5)
      DO 777 MMSET=1,MTIMES
      MSET=MMSET
C
      CALL CONTRL(6)
      DO 1 LGG=1,LGMAX
      LG=LGG
      CALL CONTRL(7)
      DO 2 KK=1,KM
      K=KK
C
      CALL CONTRL(8)
      IF (K .EQ. 1) GO TO 122
C
C       FOLLOWING ADVANCES THE TRUE STATE AND SAVE PRIOR VALUE
C
      DO 100 I=1,9
      X1(I)=X(I)
      X(I)=0.
  100 CONTINUE
      DO 120 I=1,9
      DO 110 J=1,9
      X(I)=X(I)+F(I,J)*X1(J)
  110 CONTINUE
  120 CONTINUE
      B=PVALUE(B,PI,0.)
C
C FOLG SETS THES PRIOR TO COMPUTING SAMPLE TIMES
      THES=XS(2)+DELT*XS(3)
      IF(.NOT.TETHRD) GO TO 124
  122 CONTINUE
      DO 123 I=1,9
```

```
           XS1(I)=X(I)-EO(I)
   123     CONTINUE
           THES=XS1(2)
   124     CONTINUE
C
C FOLG BEGINS COMPUTATION OF SIGNALS AND RELATED QUANTITIES
C ... FIRST, CALCULATION OF CONSTANTS ON THE PRESENT SCAN
C
           AL2=ALFA**2
           AR2=ALFAR**2
           AA2=2.*ALFA*ALFAR
           TPST=(THES-THAMAX)*SECPD
           TPSF=TF+(THAMIN-THES)*SECPD
           TZT=TPST-TWW2
           TZF=TPSF+TWW2
C FOLG COMPUTES FOR THIS KTH SCAN THE SIGNAL PEAK TIMES TPKT, TPKF AND
C THE COMPOSITE SIGNAL-TO-NOISE RATIO  CSNR(K) .
           TPKT=(X(2)-THAMAX)*SECPD
           TPKF=TF+(THAMIN-X(2))*SECPD
C
C     THAJ=THA(TPKT)=X(2), BY DEFINITION
C PJ=PMLS(THE-THA(TPKT))=1.0,BY DEFINITION
C AND SIMILARLY FOR TPKF
C
           PRJ=PMLS(X(5)-X(2))
           BT=PVALUE(B+(WSC+0.5*WSCDOT*TPKT)*TPKT,PI,0.)
           BF=PVALUE(B+(WSC+0.5*WSCDOT*TPKT)*TPKT,PI,0.)
           QJT=AL2+AA2*PRJ*COS(BT)+AR2*(PRJ**2)
           QJF=AL2+AA2*PRJ*COS(BF)+AR2*(PRJ**2)
           CSNRT=SQRT(QJT)
           CSNRF=SQRT(QJF)
           CSNRMN=AMIN1(CSNRT,CSNRF)
           CSNRMX=AMAX1(CSNRT,CSNRF)
           CSNR=CSNRMN
           DCSNR=CSNRMX-CSNRMN
C
C FOLG INITIATES A J-LOOP TO COMPUTE SAMPLE TIMES AND (LINEAR)
C     ENVELOPE SAMPLE VALUES
C
           DO 130 J=1,JM2
           TINCR=J*TSAMP
           JFR=JM1-J
           INDEX (1)=J
           INDEX (2)=JFR
C FOLG COMPUTES SAMPLE TIMES
           TJ=TZT+TINCR
           T(J)=TJ
           T(JFR)=TZF-TINCR
           THAJ=THA(TJ)
           PJ=PMLS(THE-THAJ)
           PRJ=PMLS(THR-THAJ)
C FOLG COMPUTES ENVELOPE SAMPLES
           DO 127 I=1,2
           JVAL=INDEX(I)
           TJVAL=T(JVAL)
           XNC=SQ22*GAUSS(0.)
           XNS=SQ22*GAUSS(0.)
```

```
      BLOCAL=PVALUE(B+(WSC+0.5*WSCDOT*TJVAL)*TJVAL,PI,0.)
      QJ=AL2*(PJ**2)+AA2*PJ*PRJ*COS(BLOCAL)+AR2*(PRJ**2)
      QJ=AMAX1(QJ,0.0)
      SQRTQ2=2.*SQRT(QJ)
      UJ=QJ+SQRTQ2*XNC+XNC**2+XNS**2
      V(JVAL)=SSQ2*SQRT(UJ)
  127 CONTINUE
  130 CONTINUE
C
C       FOLLOWING COMPUTES RECEIVER RESPONSES
C
      CALL RCVR
C
C       FOLLOWING ASSIMILATES DATA FROM THIS SCAN FOR FUTURE EVALUATION
C
      CALL CONTRL(9)
    2 CONTINUE
C
C FOLLOWING ASSIMILATES DATA FROM THIS LG-TH SET OF SCANS
C
      CALL CONTRL(10)
    1 CONTINUE
C
C FOLLOWING ASSIMILATES DATA FROM THIS MSET-TH SERIES OF LGMAX SETS OF KM SCANS
C
      CALL CONTRL(11)
  777 CONTINUE
C
C FOLLOWING CLOSES THE SIMULATION RUN
C
      CALL CONTRL(12)
      RETURN
      END
```

41000B CM STORAGE USED 1.465 SECONDS

```
      FUNCTION THA(T)
C
C      THIS COMPUTES THE ANTENNA SCAN ANGLE AT LOCAL SCAN TIME T.
C      THE PARAMETERS OF THE SCAN WAVEFORM,HENCE THE IDENTITY OF
C      THE ANGLE FUCTION,ARE PASSED THROUGH COMMON AND ARE DEFINED
C      AS FOLLOWS:
C      TR=TIME BETWEEN TRAVERSAL OF ZERO DEGREES IN A TO-FRO SCAN
C      TS=DURATION OF THE TO SCAN
C      TF=DURATION OF THE TO SCAN+INTERSCAN REST INTERVAL
C THAMAX=ANTENA ANGLE AT BEGINNING OF TO SCAN
C THAMIN=ANTENNA ANGLE AT END OF TO SCAN
C OMEGA=ANTENNA ANGULAR SCAN RATE,DEG/SEC,DURING TO SCAN
C
      COMMON/RCVROO/THAMAX,THAMIN,TS,TR,OMEGA,TF
        T1=TS+TF
        IF(T.GE.0.0) GO TO 50
      THA=THAMAX
        RETURN
50      IF(T.GT.TS)GO TO 100
      THA=THAMAX+OMEGA*T
        RETURN
100     IF(T.GE.TF) GO TO 200
      THA=THAMIN
        RETURN
200     IF(T.GT.T1) GO TO 250
      THA=THAMIN-OMEGA*(T-TF)
        RETURN
250   THA=THAMAX
        RETURN
        END
```

41000B CM STORAGE USED .114 SECONDS

```
      SUBROUTINE DFLTR1(X,Y,C)
      DIMENSION C(4)
      S=X-C(3)*C(4)
      Y=C(1)*S+C(2)*C(4)
    5 C(4)=S
      RETURN
      END
```

41000B CM STORAGE USED .046 SECONDS

C-17

```
      BLOCK DATA MLS
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD,MORE
      COMMON/IDDATA/IDNRS(6)
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(6),NGO(6)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GQGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/MLS001/CSNRT,CSNRF,DSNRDB,RHO,BETA,FSC,LGMAX
      COMMON/MLS003/FL10AE(4,2),FL10(4),DELTAT(2),XO(9,2),YO(4,2)
      COMMON/MLS004/BMLS,BBB,MTIMES,MSET,MORE
      DATA PI/3.1415927/,F/81*0./,DSNRDB/20./,RHO/.5/
      DATA DELTAT/.074074074,.024691358/,SIGMA/.7071/
      DATA XO/0.,30.,.35,0.,32.75,-.35,0.,0.,0.,
     *        0.,10.,0.,0.,12.,-.9,0.,2.,0./
      DATA YO/62.666667,-62.,6.233333E-3,6.6E-3,
     *30.666667,0.,1.533333E-3,0.4E-3/
      DATA FL10AE/.5,0.,-.5,0.,.25,0.,-.75,0./
      CATA FL25/.34831,.34831,-.30336,0./
      DATA JM/130/,TETHRD/.FALSE./,KM/100/,FSAMP/1.6E5/
      DATA H/1.,6*0.,1.,12*0.,1.,6*0.,1.,12*0.,1.,12*0.,1./
      DATA NSO/2,8,6,3,8,9/,NGO/2,5,4,2,6,6/
      DATA DELBL/.01/,EO/9*0./
      DATA GQGT/81*0./,PA/81*0./
      DATA RHOMAX/.8/,DTHO/2.75/,TDRO/0.0/,BO/0.0/,WSCO/0.0/
      DATA NOLOE/.TRUE./,NOKLMN/.FALSE./,NOAC/.TRUE./
      DATA IAE/1/,FSC/51.30/,BETA/-168.0/,LGMAX/1/
      DATA BMLS/1.0/,BRCVR/1.0/,MTIMES/1/,MORE/.TRUE./
      DATA IDNRS(5)/1/
      END
```

41000B CM STORAGE USED .219 SECONDS

```
      SUBROUTINE RCVR
      REAL LAMDA,MJM2
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR00/THAMAX,THAMIN,TS,TR,OMEGA,TF
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BD,WSCO,NSO(6),NGO(6)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GQGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(9),ESLOE(9),ES(9)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/XS1(9),XS(9)
      DIMENSION TPA1(9,9),PHT(9,6),TPA2(9,9),IVING(6)
      DIMENSION RVNG(6),GAIN(9,6)
C
      IF (K .GT. 0) GO TO 40
C FOLG IS INITIALIZATION
      PI2=2.*PI
      SS2=.5/(SIGMA**2)
      CALL PHILM
      NS=NSO(IR)
      NG=NGO(IR)
      NGMIN=2
      NGMAX=NGO(IR)
      GQGT33=.01*DELT
      GQGT(3,3)=GQGT33
      GQGT(6,6)=GQGT33
      GQGT(8,8)=.04/(DELT**2)
      DO 20 I=1,NS
      DO 10 J=1,NS
      PA(I,J)=0.
   10 CONTINUE
   20 CONTINUE
      CALL CONTRL(3)
      BB=2.4/BRCVR
      PDCRIT=PI*BB/8.
      CC=PI*PDCRIT
      IF(IR.GT.0) GO TO 30
      NS=NSO(4)
      NG=NGO(4)
   30 CONTINUE
      NGM=NGMIN
      IF(NOAC) NGM=NG
      KALMAN=.NOT. NOKLMN
      LOE=.NOT. NOLOE
      CALL CONTRL(4)
      RETURN
   40 CONTINUE
      IF(K.GT.1) GO TO 50
C
C DIAGNOSTIC OUTPUT OF INPUT DATA FOR K=1 GOES HERE
C
      GO TO 90
   50 CONTINUE
C FOLG EXTRAPOLATES STATE ESTIMATE, AS REQ'D
```

```
      IF(IETHRD) GO TO 80
      DO 70 I=1,NS
      XS1(I)=0.
      DO 60 J=1,NS
      XS1(I)=XS1(I)+F(I,J)*XS(J)
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
C
      GQGT11=AMAX1(.25,.01*(XS1(1)**2))
      GQGT(1,1)=GQGT11
      GQGT(4,4)=GQGT11
C
C FOLLOWING COMPUTES VECTORS Q(JM),HW(JM),LAMDA(NG),AND MATRICES DT(JM,
C NG),PHI(NG,NS) ALSO SQUARED AMPLITUDE ENVELOPE VECTOR U(JM),AND
C INOVATION;S PROCESS VECTOR W(JM)
C
      CALL PHILM
      IF (NOLOE) GO TO 120
C
C FOLG COMPUTES RMAT=PHI-INVERSE AND THE LOE
      CALL MATINV(RMAT,6,NG,IVNG,RVNG)
      DO 100 I=1,NG
      RDIAG (I)=RMAT (I,I)
  100 CONTINUE
      CALL MATMUL (6,NG,NG,6,NG,1,6,1,RMAT,LAMDA,GAMAES,1)
      CALL MATMUL (6,NG,NS,6,NG,1,9,1,H,GAMAES,ESLOE,2)
      DO 110 I=1,NS
      XSLOE(I)=XS1(I)+ESLOE(I)
      IF (NOKLMN) XS(I)=XSLOE(I)
  110 CONTINUE
  120 CONTINUE
      IF (NOKLMN) GO TO 170
C
C      FOLLOWING EXTRAPOLATES STATE ESTIMATES,ERROR
C      COVARIANCE MATRIX PA(NS,NS)
C
      CALL MATMUL(9,NS,NS,9,NS,NS,9,9,PA,F,TPA1,3)
      CALL MATMUL(9,NS,NS,9,NS,NS,9,9,F,TPA1,PA,1)
      CALL MATSM(PA,PA,GQGT,1,9,9,9,NS,NS,9,NS,NS,0)
      DO 130 I=1,NS
      PMDIAG (I)=PA(I,I)
  130 CONTINUE
C
C      FOLLOWING COMPUTES MODIFIED-KALMAN GAIN MATRIX GAIN(NS,NG)
C
      CALL MATMUL(9,NS,NS,6,NG,NS,9,6,PA,H,PHT,3)
      CALL MATMUL(6,NG,NG,6,NG,NS,9,9,PHI,H,TPA1,1)
      CALL MATMUL(9,NG,NS,9,NS,NG,9,9,TPA1,PHT,TPA2,1)
      DO 140 I=1,NG
      TPA2(I,I)=TPA2(I,I)+1.
  140 CONTINUE
      CALL MATINV(TPA2,9,NG,IVNG,RVNG)
      CALL MATMUL(9,NS,NG,9,NG,NG,9,6,PHT,TPA2,GAIN,1)
C
C      FOLLOWING UPDATES STATE ESTIMATE
```

```
      CALL MATMUL (9,NS,NG,6,NG,1,9,1,GAIN,LAMDA,ES,1)
      CALL MATSM (XS,XS1,ES,1,9,1,9,NS,1,9,NS,1,0)
C
C      FOLLOWING UPDATES STATE ESTIMATE ,ERROR COVARIANCE MATRIX
C      P(NS,NS).
C
      CALL MATMUL(9,NS,NG,9,NG,NS,9,9,GAIN,TPA1,TPA2,1)
      DO 150 I=1,NS
      TPA2(I,I)=TPA2(I,I)-1.
  150 CONTINUE
      CALL MATMUL(9,NS,NS,9,NS,NS,9,9,PA,TPA2,TPA1,3)
      CALL MATMUL(9,NS,NS,9,NS,NS,9,9,TPA2,TPA1,PA,1)
      CALL MATMUL(6,NG,NG,9,NS,NG,9,9,PHI,GAIN,TPA1,3)
      CALL MATMUL(9,NS,NG,9,NG,NS,9,9,GAIN,TPA1,TPA2,1)
      CALL MATSM(PA,PA,TPA2,1,9,9,9,NS,NS,9,NS,NS,0)
      DO 160 I=1,NS
      PPDIAG (I)=PA(I,I)
  160 CONTINUE
  170 CONTINUE
      XS(1)=ABS(XS(1))
      AN=(THAMAX+THAMIN)/2.
      XS(2)=SATU(XS(2)-AN,(THAMAX-THAMIN)/2.)+AN
      IF(NS.GE.3) XS(3)=SATU(XS(3),1.)
      IF(NS.GE.4) XS(4)=SATU(ABS(XS(4)),2*XS(1))
      MJM2=FLOAT(-(JM/2+1))
      THWW2=MJM2*OMEGA/(2.+FSAMP)
      AMX=XS(2)+THWW2
      AMN=XS(2)-THWW2
      AN2=(AMX+AMN)/2.
      IF(NS.GE.5) XS(5)=SATU(XS(5)-AN2,(AMX-AMN)/2.)+AN2
      IF(NS.GE.6) XS(6)=SATU(XS(6),1.)
      IF (NS.GE.7) XS(7)=PVALUE(XS(7),PI,0.)
      IF (NG.EQ.5) XS(8)=PVALUE(XS(8),PI/DELT,0.)
      RSMAX=8000.
      IF(NS.GE.9) XS(9)=SATU(XS(9),RSMAX)
      RETURN
      END
```

41000B CM STORAGE USED 1.412 SECONDS

```
      BLOCK DATA ORVRID
      COMMON/IDDATA/IDNRS(6)
      DATA IDNRS(4)/2/
      END
```

41000B CM STORAGE USED .012 SECONDS

```
      SUBROUTINE PHILM
C
C THIS OPTIMAL VERSION OF PHILM IS FOR ALL SCALLOPING RATES
C AND PROVIDES A CRUDE SEARCH-AND-ACQUISITION FUNCTION
C ACCESSED AS FOLLOWS:
C     SEARCH MODE: NGM=NGMIN
C     ACQUISITION MODE: NGMIN.LT.NGM.LT.NGMAX
C     FULL TRACK MODE: NGM=NGMAX
C
C
C   THIS NEEDS NGMIN.LE.NGM (.LE.NG) .LE.NGMAX
C
      REAL LAMDA
      LOGICAL NOKLMN,NOLOE,NOAC,KALMAN,LOE,TETHRD
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      COMMON/RCVR02/RHOMAX,DTHO,TDRO,BO,WSCO,NSO(6),NGO(6)
      COMMON/RCVR03/PI,F(9,9),FL25(4),FSAMP,K,KM,TETHRD,NG,NS,JM
      COMMON/RCVR04/DELT,EO(9),GQGT(9,9),H(6,9),ICOUNT,SIGMA
      COMMON/RCVR05/NOLOE,NOKLMN,NOAC,GAMES(6),RDIAG(6),PMDIAG(9)
      COMMON/RCVR06/PPDIAG(9),RMAT(6,6),PHI(6,6),PA(9,9),LAMDA(6)
      COMMON/RCVR07/T(130),V(130)
      COMMON/RCVR08/XSLOE(9),ESLOE(9),ES(9)
      COMMON/RCVR09/BRCVR,BB,PDCRIT,CC
      COMMON/RCVR10/ALFA,THES,THESDT,ALFAR,THRS,THRSDT,B,WSC,
     CWSCDOT,XS(9)
      DIMENSION INDEX(2),DT(130,6),HW(130),W(130)
C
C FOLLOWING EFFECTS INITIALIZATION
C
      IF(K.GT.0) GO TO 10
      JM2=JM/2
      JM1=JM+1
      SS2=.5/(SIGMA**2)
      RETURN
   10 CONTINUE
C
C FOLLOWING PROGRAMS THE SEARCH
C
      IF (NGM.GT.NGMIN) GO TO 30
      ALFAR=ALFA*RHOMAX
      THRSDT=TDRO
      B=BO
      WSC=WSCO
      THRS=THES-DTHO
C FOLG TETHERS WSCDOT
      WSCDOT=0.
   30 CONTINUE
C
C     FOLLOWING COMPUTES CONSTANTS FOR PRESENT SCAN
C
      AL2=2.*ALFA
      AL22=AL2*ALFA
      AR2=2.*ALFAR
      AR22=AR2*ALFAR
      AA2=AL2*ALFAR
C
C     FOLLOWING INITIATES LOOP AND COMPUTES FUNCTIONS
```

```
C      FOR EACH J
C
       DO 60 JI=1,JM2
       INDEX(1)=JI
       INDEX(2)=JM1-JI
       THAJ=THA(T(JI))
       THEE=THES-THAJ
       PJ=P(THEE)
       PDJ=PDOT(THEE)
       P2J=PJ**2
       THER=THRS-THAJ
       PRJ=P(THER)
       PDRJ=PDOT(THER)
       PR2J=PRJ**2
       PPRJ=PJ*PRJ
       QNRJ=.5*AL22*P2J
       QRJ=QNRJ+.5*AR22*PR2J
       D1NET=AL2*P2J
       C1=AR2*PPRJ
       D2NET=AL22*PJ*PDJ
       C2=AA2*PDJ*PRJ
       D3J=AR2*PR2J
       C3=AL2*PPRJ
       D4J=AR22*PDRJ*PRJ
       C4=AA2*PJ*PDRJ
       D5J=-AA2*PPPJ
C FOLG COMPLETES CALCULATIONS FOR TO,FRO SCANS, RESP.
       DO 50 I=1,2
       J=INDEX(I)
       TJ=T(J)
       BLOCAL=PVALUE(B+(WSC+0.5*WSCDOT*TJ)*TJ,3.1415927,DELBL)
       SB=SIN(BLOCAL)
       CB=COS(BLOCAL)
       QJ=QNRJ
       QRJALL=QRJ-D5J*CB
       D1J=D1NET
       D2J=D2NET
       IF (NGM.LT.NGMAX) GO TO 40
       QJ=QRJALL
       D1J=D1J+C1*CB
       D2J=D2J+C2*CB
 40    CONTINUE
       DT(J,1)=D1J
       DT(J,2)=D2J
       DT(J,3)=D3J+C3*CB
       DT(J,4)=D4J+C4*CB
       DTJ5=D5J*SB
       DT(J,5)=DTJ5
C FOLG IS FOR A 6D LOE DESIGN
       DT(J,6)=DTJ5*TJ
C FOLLOWING COMPUTES VECTOR HW(JM),AND INNOVATIONS PROCESS VECTOR W(JM)
C
       HW(J)=1./(1.+2.*QJ)
       UJ=V(J)
       UJ=SS2*(UJ**2)
       W(J)=(UJ/SQRT(1.+QJ*UJ))-1.
 50    CONTINUE
```

```
   60   CONTINUE
C
C        FOLLOWING COMPUTES VECTOR LAMDA(NG) AND MATRICE PHI(NG,NG)
C
        DO 110 I=1,NG
        LAMDA(I)=0.0
        DO 70 J=1,JM
        LAMDA(I)=LAMDA(I)+DT(J,I)*W(J)
   70   CONTINUE
        DO 100 L=1,I
        PHILI=0.
        DO 80 J=1,JM
        PHILI=PHILI+HW(J)*DT(J,I)*DT(J,L)
   80   CONTINUE
        PHI(L,I)=PHILI
        PHI(I,L)=PHILI
        IF (NOLOE) GO TO 90
        RMAT(L,I)=PHILI
        RMAT(I,L)=PHILI
   90   CONTINUE
  100   CONTINUE
  110   CONTINUE
        RETURN
        END
```

41000B CM STORAGE USED .823 SECONDS

```
      BLOCK DATA PLOPID
      COMMON /IDDATA/IDNRS(6)
      COMMON/RCVR01/NGMIN,NGMAX,DELBL,NGM,IR,IAE
      DATA IDNRS(3)/2/
      DATA IR/6/
      END
```

41000B CM STORAGE USED .019 SECONDS

DISTRIBUTION LIST

Copy No.	
1 - 2	NASA Scientific & Technical Information Facility P. O. Box 8757 Baltimore/Washington International Airport Baltimore, MD 21240
3 - 13	Flight Electronics Division National Aeronautics and Space Administration Langley Research Center Hampton, VA 23365
14 - 15	G. A. McAlpine
16 - 18	J. H. Highfill III
19	E. A. Parrish, Jr.
20	I. A. Fischer OSP
21 - 22	E. H. Pancake Clark Hall
23	RLES Files

JO# 9903 JDH

UNIVERSITY OF VIRGINIA

School of Engineering and Applied Science

The University of Virginia's School of Engineering and Applied Science has an undergraduate enrollment of approximately 1,000 students with a graduate enrollment of 350. There are approximately 120 faculty members, a majority of whom conduct research in addition to teaching.

Research is an integral part of the educational program and interests parallel academic specialties. These range from the classical engineering departments of Chemical, Civil, Electrical, and Mechanical to departments of Biomedical Engineering, Engineering Science and Systems, Materials Science, Nuclear Engineering, and Applied Mathematics and Computer Science. In addition to these departments, there are interdepartmental groups in the areas of Automatic Controls and Applied Mechanics. All departments offer the doctorate; the Biomedical and Materials Science Departments grant only graduate degrees.

The School of Engineering and Applied Science is an integral part of the University (approximately 1,400 full-time faculty with a total enrollment of about 14,000 full-time students), which also has professional schools of Architecture, Law, Medicine, Commerce, and Business Administration. In addition, the College of Arts and Sciences houses departments of Mathematics, Physics, Chemistry and others relevant to the engineering research program. This University community provides opportunities for interdisciplinary work in pursuit of the basic goals of education, research, and public service.