4378-1152-F

NASA CR-145303

INVESTIGATION OF NEW TECHNIQUES FOR AIRCRAFT NAVIGATION USING THE OMEGA NAVIGATION SYSTEM

by

Ernest G. Baxa, Jr.

Research Triangle Institute
Research Triangle Park, NC 27709

Contract NAS1-14005

February 1978

NASA

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

ACKNOWLEDGMENTS

This report was prepared by the Research Triangle Institute, Research Triangle Park, North Carolina, under Contract NAS1-14005. The work is being administered by the Flight Instrumentation Division, Langley Research Center.

The report describes results of studies in support of a NASA program to evaluate the performance capabilities of the OMEGA navigation system for use by civil aviation. The Research Triangle Institute has worked closely with Mr. C. D. Lytle, Mr. J. G. Wells, Jr., and Mr. E. M. Bracalente of the Telecommunications Research Branch under the direction of Mr. J. H. Schrader. Software support at the Langley Research Center has been provided by personnel of the LTV Aerospace Corporation under contract to NASA and assigned to this project.

RTI staff members participating in this study are as follows:

Dr. E. G. Baxa, Jr., Project Leader
Dr. C. L. Britt, Jr., Laboratory Supervisor
Ms. S. L. Wheelock, Mathematician
Mr. K. W. Leland, Engineer
Ms. M. E. Gordon, Secretary

Dr. R. W. Stroh, North Carolina State University faculty member participated as a consultant in the development of the microprocessor simulator developed under this contract.

"Page missing from available version"

page IV

TABLE OF CONTENTS

"Page missing from available version"

VII

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

"Page missing from available version"

page XIV

ABSTRACT

This final report under Contract NAS1-14005 documents work done in support of a current NASA program for evaluation of the performance capabilities of the OMEGA navigation system for use by civil aviation. With support from the Research Triangle Insitute, NASA personnel at the Langley Research Center are investigating the implementation of an OMEGA navigation receiver with a microprocessor as the computational component. This support included providing a version of the INTEL 4004 microprocessor macroassembler suitable for use on the CDC-6600 system and development of a FORTRAN IV simulator program for the microprocessor. Supporting studies included development and evaluation of navigation algorithms to generate relative position information from OMEGA VLF phase measurements. Simulation studies were used to evaluate assumptions made in developing a navigation equation in OMEGA Line of Position (LOP) coordinates. Included in the navigation algorithms was a procedure for calculating a position in latitude/longtitude given an OMEGA LOP fix. A comparison of this procedure with a previously published procedure is presented.

Implementation of a digital phase-locked loop (DPLL) was evaluated on the basis of phase response characteristics over a range of input phase variations. Included also is an analytical evaluation on the basis of error probability of an algorithm for automatic time synchronization of the receiver to the OMEGA broadcast format. The use of actual OMEGA phase data and published propagation prediction corrections (PPC) to determine phase velocity estimates is discussed. Algorithms and the necessary computer programming were developed to incorporate phase velocity estimates in the generation of OMEGA charts for use on standard aeronautical topographical maps. A software package written to plot OMEGA LOPs directly onto Lambert projection maps is described.

Finally, the results of several flight tests with the microprocessor based OMEGA receiver are presented. These describe the position estimate errors relative to tracking radar as well as the utility of the various navigation outputs which have been discussed in detail within this report.

Page xvi
is
BLANK

1.0 INTRODUCTION

For several years NASA personnel at the Langley Research Center at Hampton, Virginia, have been investigating utilization of the OMEGA navigation system. An experimental program designed primarily to evaluate accuracies and limitations of various operating modes has been supported by the Research Triangle Institute and is currently continuing from the ground-based phase to an airborne phase. An earlier report (ref. 1) describes the experimental program and presents some preliminary data analysis results.

This report describes work done under Contract NAS1-14005 over the past 20 months in studies to support the NASA program for evaluation of the performance capabilities of the OMEGA navigation system for use by civil aviation. Most of the studies under this contract have been related to providing computer software and analysis support in investigating the implementation of an OMEGA navigation receiver which uses a microprocessor as the computational component. These studies included providing a microprocessor cross-assembler and simulator for use by NASA personnel, developing algorithms to provide for mapping of OMEGA coordinates to aeronautical topographical map coordinates, investigation of use of composite OMEGA for lane determination, skywave correction techniques, and methods for estimating actual phase velocity. Additionally, the Institute provided support to the experimental program (ref. 1) by accumulating, interpolating and reformatting skywave correction values to be used with all of the data gathered during the program.

The major hardware development effort associated with the OMEGA receiver has been done at Langley Research Center by NASA personnel with support provided by RTI and LTV Aerospace Corporation. The software development has been a joint effort as has performance evaluation. In flight tests conducted at Wallops Flight Center in Virginia, NASA personnel there have provided computer reduction of radar tracking tapes. This report is intended to provide documentation of results of this total effort and should not be interpreted as results of work done solely by RTI personnel. Flight test evaluations are continuing at NASA Langley and will be reported on elsewhere.

In this report Chapter 2 describes particular aspects of the microprocessor based OMEGA navigation receiver being implemented by NASA personnel. Chapter 3 provides a discussion of the navigation equations developed for the receiver and algorithms for converting OMEGA coordinates to latitude/longitude. Chapter 4 provides a discussion of phase velocity estimation using a developed VLF propagation model and using published propagation prediction corrections for OMEGA. A procedure for generation of OMEGA chart lattice grids for use directly with aeronautical Lambert projection topographical maps is described. Chapter 5 describes the results of several flight tests which were conducted at the NASA Wallops Flight Center facility with the microprocessor-based OMEGA receiver. Six appendices are included to provide more detail relating to specific tasks within this contract. Appendix A provides a detailed analysis of the Digital Phase-Locked Loop (DPLL) receiver. Appendix B discusses the procedure used to collect, interpolate, and reformat skywave correction data (PPC) for the receiver locations and time associated with the ground-based experimental program conducted at NASA Langley (ref. 1 and 5). Appendix C describes two iterative methods for determining the latitude/longtitude of an OMEGA LOP fix. Appendix D provides software documentation of the OMEGA Chart Lattice Program. Appendix E details the changes made to the INTEL 4004 Macro Assembler for use on the NASA-LRC CDC-6600 computer. Appendix F is a user's manual for the RTI developed INTEL simulator program SIM4.

2.0 MICROPROCESSOR BASED OMEGA NAVIGATION RECEIVER

2.1 General

With the development and refinement of microprocessors and microcomputers the feasibility of a very low-cost, sophisticated receiver for OMEGA navigation use has been enhanced. Personnel at NASA Langley Research Center are involved with development of a feasibility model of a digitally implemented phase-locked loop OMEGA receiver which uses an INTEL 4004 microprocessor as the computational component. The receiver is configured in a small aircraft type chasis requiring 12 VDC power. The receiver includes an analog r.f. front end using mechanical band-pass filters, employs high gain and limiting to provide adequate SNR and OMEGA noise immunity, and accomplishes phase detection using a digitally implemented phase-locked loop. The receiver is designed to be capable of tracking any four OMEGA transmissions at 10.2 kHz and 13.6 kHz and will provide for navigation in either the 10.2 kHz carrier mode or the 3.4 kHz difference frequency mode. An internal crystal controlled clock operating at 2.61120 MHz provides reference phase information. A digital discriminator of the general type described in a previous report (ref. 1) is used in conjunction with other digital circuits and the 4004 microprocessor and associated software to operate on a 0.8 second duty cycle during operator selected OMEGA transmission times. This chapter describes portions of the receiver including an analysis of the synchronization algorithm used to automatically lock to the OMEGA signal format, an analysis of the phase-locked loop implementation in terms of the loop parameter selections, and a discussion of antenna types which are designed for airborne use that could be employed with the receiver.

2.2 Synchronization to the OMEGA Signal Format

The automatic synchronization technique described in this section would normally be implemented at the time the receiver is powered up and could be used at any time synchronization is lost. A keyboard input from

the operator initiates the action. Synchronization is accomplished on the basis of sampled measurements of the output of an envelope detector driven by the receiver limiter output filtered at the 13.6 kHz carrier frequency. A sampling rate of 12.8 samples/sec is implemented so that 128 envelope sample values are accumulated over a 10 sec OMEGA frame time. Each amplitude sample value is thresholded and digitized in binary levels ("0" or "1") and accumulated in one of 128 4-bit registers over a 150 second interval. The result is a set of 128 averaged amplitude samples which have a value in the interval 0, 15 and represent an average measure of the inband signal over the OMEGA 10 second format relative to the starting time of the sampling process. This averaged "waveform" can then be correlated with a stored pattern representing a noiseless version of the OMEGA signal where signal on-times are represented as maximum positive and signal off-times are represented as zero. In practice the stored measurements of signal amplitude are multiplied by two, 15 is subtracted from each value to remove bias, and the stored pattern is adjusted accordingly. In the stored pattern on-times are represented by +15 and off-times are represented by -15 sample values. By correlating the stored pattern with the stored signal samples the time offset between the start of the sampling window can be determined relative to the beginning of the OMEGA signal format and individual transmitter on-times can be identified in terms of a number of sample times ($\frac{1}{12.8}$ sec intervals) from the origin.

In the receiver, since the phase-locked loop discriminator uses a 0.8 sec phase measurement interval and the shortest OMEGA signal on-time is 1.1 sec., it is only necessary to locate a particular transmitter on-time within $\pm$.15 sec or approximately $\pm$ 2 sample intervals to achieve adequate synchronization.

In the actual receiver an early version of the software has not been generalized in that receiver check-out will be accomplished using a stored pattern representing the N. Dakota 13.6 kHz transmission on-time and adjacent off-times only. This is particularly convenient since N. Dakota is distinctly the strongest most stable station received in the Hampton, Va., area and does provide a sufficiently peaked autocorrelation function to

give reliable synchronization of the receiver. This can furthermore demonstrate the feasibility of the method.

2.2.1 Analytical evaluation of the Automatic Synchronizer.— Consider the situation where a stored amplitude pattern is truncated to include on-times of the B, C, and D segments at 13.6 kHz and the adjacent off-times. Figure 2-1 illustrates the 54 sample values to be stored (assuming 12.8 samples per sec) referenced to the start time of the NOR 10.2 kHz transmission. It can be noted that the sample number 65 is synchronous with the

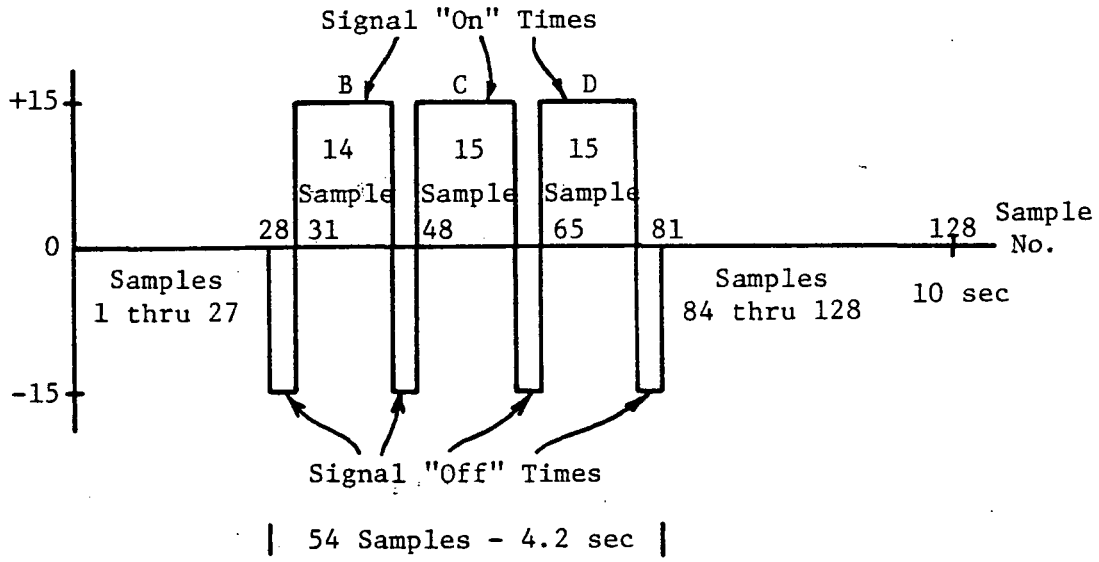


Figure 2-1. Reference Signal Pattern for Correlation Tests.

beginning of the D transmission at 13.6 kHz. On-time sample values are given a weight of +15 and off-time samples are assigned a weight of -15.

Figure 2-2 is the correlation function of the pattern of Figure 2-1. This assumes all times not represented are at the -15 level (no signal). This correlation function is defined in terms of points calculated at discrete sample time ($\Delta\tau = \frac{1}{1.28}$ sec) displacements from $-54\Delta\tau \leq \tau \leq +54\Delta\tau$. This function is quite peaked and has sufficient range to offer good potential for time location of the stored pattern with sufficient accuracy to insure synchronization.

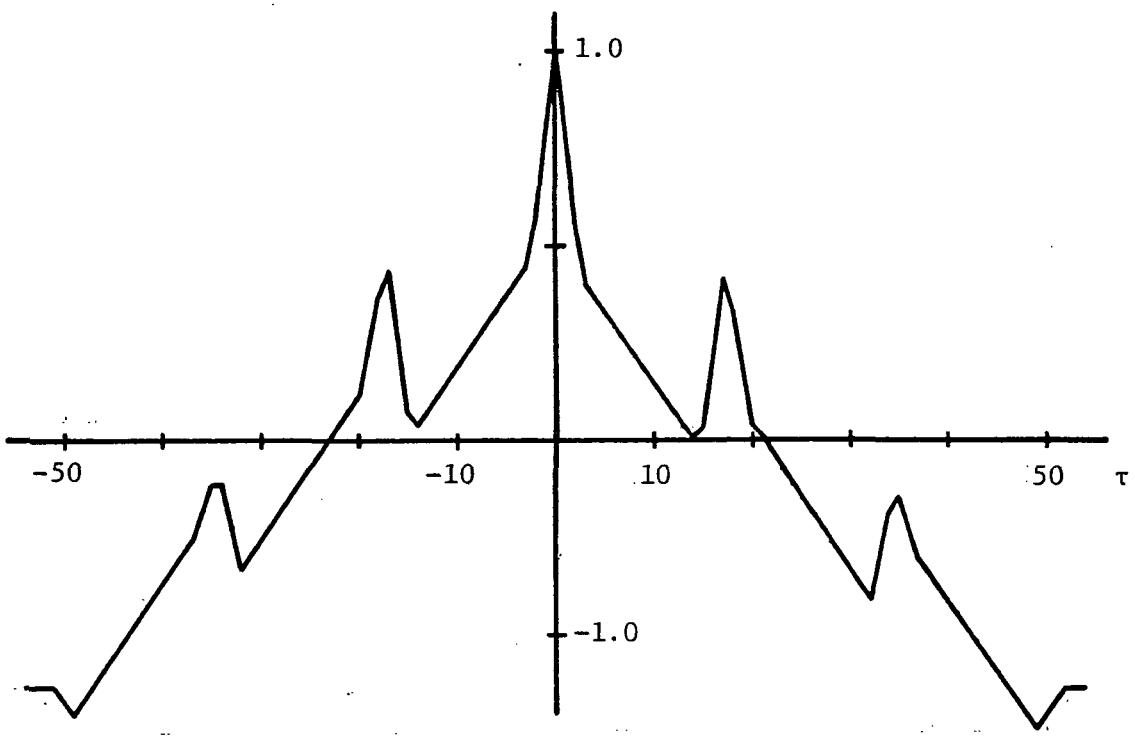


Figure 2-2. Correlation Function of 3 Segment Amplitude Pattern Considering Perfect Reception of Only Three Stations

The probability of not acquiring synchronization is dependent primarily on the choice of the threshold, T, used to quantize each envelope sample in arriving at the average amplitude level at each sample time during the 10 second interval. The choice of T can be made based on the probability of error at any given sample time. The following discussion defines a procedure for choosing T to minimize the probability of error at any sample time.

Figure 2-3 represents the circuit functions to arrive at a quantized envelope level at each sample time. Representing the envelope detector

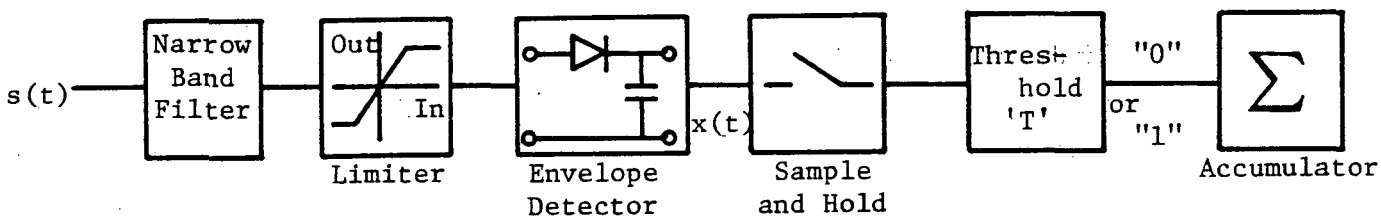


Figure 2-3. Processing in Receiver To Get "Average Envelope" in Each Sample Position.

output as x(t) consider two possible situations stated in the form of exhaustive mutually exclusive hypotheses. The output x(t) is

$$H_0: \quad x(t) = n(t)$$

or

$$H_1: \quad x(t) = A + n(t)$$

where n(t) represents envelope fluctuations and A is the amplitude of the envelope of the OMEGA signal. Here H_0 is the noise alone hypothesis and H_1 is the signal hypothesis. The noise or level fluctuations will be described in terms of a zero mean probability density function $p_n^\circ(\cdot)$ with standard deviation σ. In general σ may vary with time which is representative of a non-stationary process and can be used to model noise containing burst energy. Since x(t) is rectified it can be described statistically in terms of a rectified density function $p_n'(\cdot)$ which is related to $p_n^\circ(\cdot)$ as a folded density with values for $x > 0$ only. Prior to thresholding, x(t) is sampled so that each sample $x_i \equiv x(t_i)$ can be described in terms of the density function $p_n'(\cdot)$ as

$$H_0: \quad f_0(x_i) = p_n'(x) = 2p_n^\circ(x) \quad , x \geq 0$$

and

$$H_1: \quad f_1(x_i) = p_n'(x-A) = p_n^A(x) + p_n^{-A}(x) \quad , x \geq 0$$

where $p_n^A(\cdot)$ is just $p_n^\circ(\cdot)$ with mean A.

Consider the probability of error $Pr(\varepsilon)$ at any sample point in the thresholding process. Assigning a "0" when signal is present or assigning a "1" when no signal is present would constitute an error. Formulating $Pr(\varepsilon)$

$$Pr(\varepsilon) = Pr(H_1) \, [1 - P_{DET}] + Pr(H_0) \, P_{FA}$$

where $Pr(H_1) \triangleq$ probability that signal of amplitude A is present

$Pr(H_0) \triangleq$ probability that no signal is present

$[1-P_{DET}] \triangleq [1-Pr(x > T|H_1)] = Pr(x < T|H_1)$

$P_{FA} \triangleq Pr(x > T|H_0)$.

Therefore

$$Pr(\varepsilon) = Pr(H_1) \int_0^T [p_n^A(r) + p_n^{-A}(r)]dr + Pr(H_0)[1 - \int_0^T 2p_n^\circ(r)dr] \quad (2\text{-}1)$$

It is implicit in this formulation that the limiter threshold is set so as not to limit the signal amplitude of the input OMEGA signal. Reducing (2-1) yields

$$Pr(\varepsilon) = Pr(H_0)[1 - \int_0^T 2p_n^\circ(r)dr] + Pr(H_1)[\int_{-T}^T p_n^A(r)dr] \quad (2\text{-}2)$$

To minimize $Pr(\varepsilon)$, choose T such that $\frac{\partial P_r(\varepsilon)}{\partial T} = 0$. From (2-2)

$$0 = -2Pr(H_0)p_n^\circ(T) + Pr(H_1)\ [p_n^A(T) + p_n^A(-T)]$$

or choose T to satisfy

$$\frac{p_n^A(T) + p_n^A(-T)}{p_n^\circ(T)} = \frac{2Pr(H_0)}{Pr(H_1)} \quad (2\text{-}3)$$

Assuming gaussian statistics where $p_n^\circ(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{r^2}{2\sigma^2}\right\}$

(2-3) can be approximated by

$$\frac{AT}{\sigma^2} - \frac{A^2}{2\sigma^2} = \ell n\ [2Pr(H_0)/Pr(H_1)] \quad (2\text{-}4)$$

Let $T' = \frac{T}{\sigma}$ and $A' = \frac{A}{\sigma}$ and (2-4) reduces to

$$T' = \frac{A'}{2} + \frac{1}{A'} \ell n \, 2Pr(H_0)/Pr(H_1). \qquad (2\text{-}5)$$

Figure 2-4 is a plot of (2-5) indicating the asymptote of $T' = \frac{A'}{2}$ where $Pr(H_0)$ and $Pr(H_1)$ are assigned values based on the proportion of samples which is represented by the stored signal pattern corresponding to Figure 2-1. From (2-5) it is seen that for $A >> \sigma$ it is desirable to use $T = \frac{A}{2}$, i.e., set the threshold at one-half the signal envelope level for minimum probability of error. For greater noise variation the threshold T should be increased. Normally $A \geq 2\sigma$ such that the limit of T may be considered as $T \leq 0.75A$ with the likelihood of signal presence as shown in Figure 2-1. Note that with the assumptions made here $E\{n(t)\} = \sigma \sqrt{\frac{2}{\P}}$.

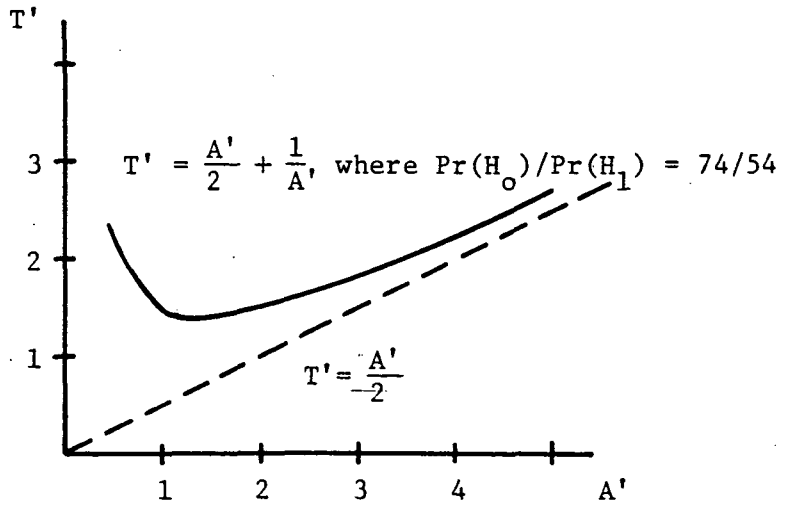


Figure 2-4. Sample Threshold vs. Envelope Signal Amplitude for Minimum Probability of Error.

It can be shown that if A > 0 and if 15 OMEGA intervals are sufficient for each signal on-time and off-time sample accumulator to achieve the average count value that synchronization will be achieved with probability one. This is dependent upon choosing T for minimum probability of error for each sample. Synchronization tests using this procedure have not been made.

2.3 Digital Phase-Locked Loop Analysis

The microprocessor based OMEGA receiver employs a digital phase-locked loop (DPLL) for phase measurement of up to four operator selected OMEGA stations at two carrier frequencies. Implementation of the loop is in hardware and software. An 8-bit RAM register is designated for each transmitter/frequency measurement segment to save the loop phase value at the end of a given measurement interval. With only one loop used, the loop phase is set at the beginning of each measurement interval according to the value stored in RAM after the previous measurement interval of that segment. The loop phase is then an 8-bit significant measure of OMEGA phase relative to the local oscillator.

For each carrier frequency the phase representation is in terms of clock counts relative to the local oscillator with 2.56 counts per centicycle at the respective frequency. Analyzing the performance at 10.2 kHz a pulse train derived from the receiver clock at a 2.61120 MHz rate is used to count relative phase. The discriminator has a 0.8 sec duty cycle every 10 seconds so that an average phase difference between the received phase and the loop phase is determined over each measurement interval. The phase difference in whole clock counts is scaled by four dividers of values 16, 12, 32, and 2. The discriminator characteristic is such that maximum output occurs when the phase difference is 25 cecs (MOD 2π). The maximum output is in clock counts

$$C = \pm \frac{(0.8)(10200)(256)}{2} = \pm 1044480$$

where the factor 2 in the denominator results from the fact that counts are accumulated during the 0.8 sec measurement interval during only one-half cycle each cycle. This is a similar scheme to that described in a previous report (ref. 1). After scaling the discriminator output is in counts

$$C' = \pm \frac{C}{32\times16\times12\times2} = \pm 85 \quad .$$

This yields a discriminator gain of 85/25 = 3.4 counts/cec. The second-order feedback loop has a gain $1/K_3$ in the direct channel and gain $1/K_1$ in the integrator channel. The loop phase is shifted by FIX($\frac{1}{2.56}$ × FEEDBACK OUTPUT) at the end of each measurement interval. Figure 2-5 depicts the DPLL in block diagram form. Appendix A describes this DPLL in terms of analog equivalent transfer functions. The gain constants K_1 and K_3 are free parameters which are set up as powers of 2 and implemented in software. Bit shifting is used to accomplish the scaling operations in the direct and integrated parts of the feedback loop.

Of primary interest is the selection of the constants K_1 and K_3 to provide for DPLL phase response suitable for the environment in which the receiver will operate. Recommended values are discussed in the following analysis based on step-responses, phase ramp responses, and functions of phase with respect to time which would be typical for an aircraft maneuvering while navigating with OMEGA. Except for local oscillator drift which appears as a phase ramp input (in the short term) this analysis does not include response to noisy phase measurements. Therefore the conclusions represent a "best case" situation for the DPLL.

Several loop parameters can be defined in terms of the gains K_1 and K_3 using analog loop analysis (see Appendix A). Referring to Figure 2-5 the loop gain is

$$G = \frac{K_0 K_d}{10K_3} = \frac{.1328}{K_3} \quad ,$$

the damping factor is

$$\delta = .576 \frac{\sqrt{K_1}}{K_3}$$

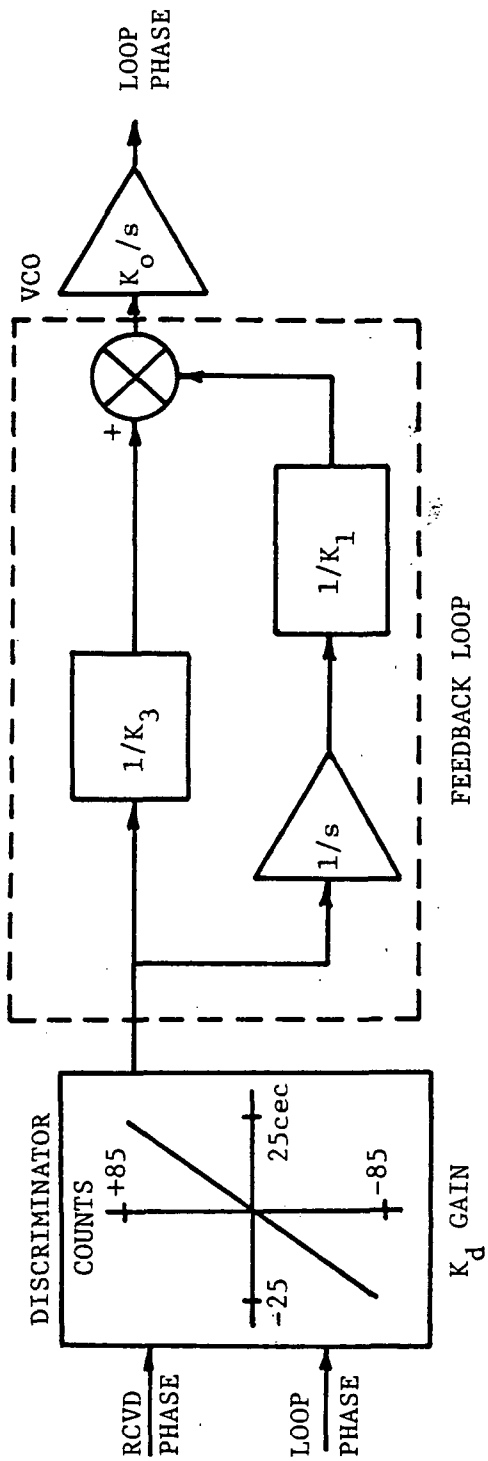


Figure 2-5. Analog Equivalent of DPLL

and the natural frequency is

$$\omega_n = \frac{.115}{\sqrt{K_1}} \text{ rads/sec}.$$

Underdamped response ($\delta \leq 1$) is characteristic for $K_1 < 3.01K_3^2$. Figures 2-6 and 2-7 illustrate the DPLL step response for the situations where $K_1 = 8$, $K_3 = 2$ and $K_1 = 16$, $K_3 = 4$ both of which are underdamped. The abscissa of these plots is the number of phase samples input to the DPLL at 10 second intervals. This scale is therefore in units of time, i.e., the number of 10 second intervals. The phase step input is assumed to begin at $t = 0$. The ordinate is in counts and centicycles. The step input has a magnitude of 20 cec and the DPLL response is shown as the solid curve. The response is presented as a continuous curve but is in fact a sequence of line segments drawn between discrete loop outputs at the 10 sec intervals. It can be seen that both responses exhibit overshoot ($\sim$ 25% and $\sim$ 30%) and that the time constant for the $K_3 = 4$ case is approximately twice that of the $K_3 = 2$ response. The time response for these two situations is less than 20 secs which is probably the upper bound for a maneuvering aircraft to avoid loss of lock and erroneous position estimates. For either case the overshoot can be reduced or eliminated by increasing K_1 thus increasing damping.

For the overdamped situation ($K_1 > 3.01\ K_3^2$) the time constant is not expressable as a simple function of loop parameters. However as the damping factor becomes large with respect to 1, the time constant variation approaches that of the first order loop and can be approximated by $\tau \simeq \frac{K_3}{.1328}$. Conclusions of this analysis are that an overdamped system is desirable provided that the response time is maintained fast enough. Figure 2-8 provides a summary of K_1, K_3 choices based on this step response considering % overshoot and T_{ss} (time to steady state). The time to steady state is

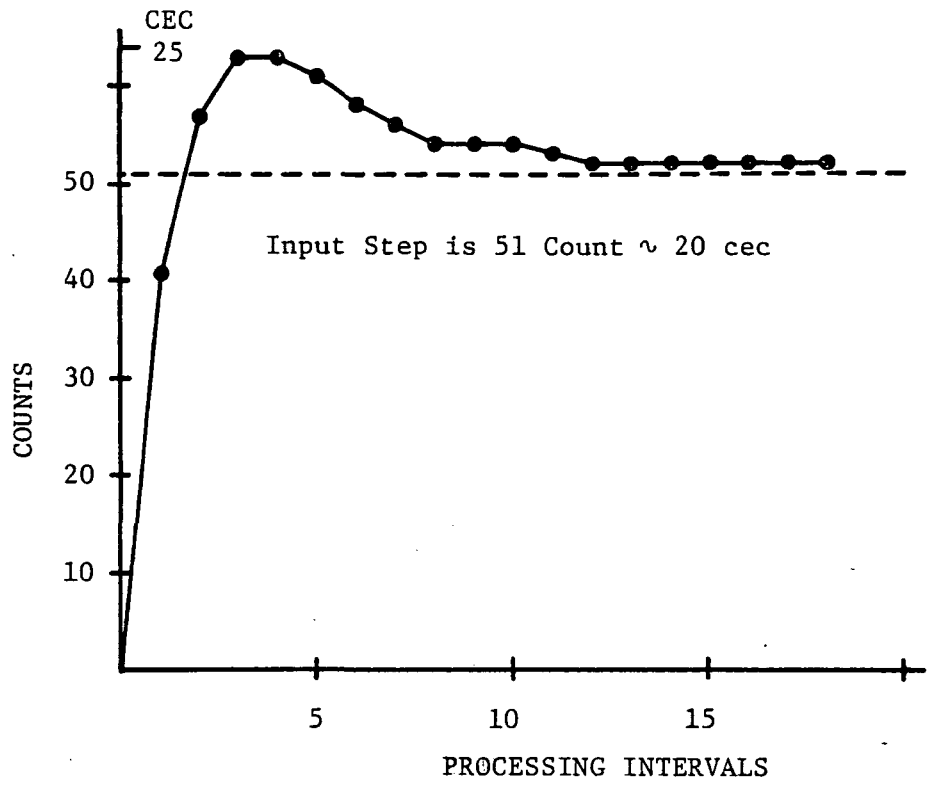


Figure 2-6. DPLL Step Response with K_1 = 8, K_3 = 2.

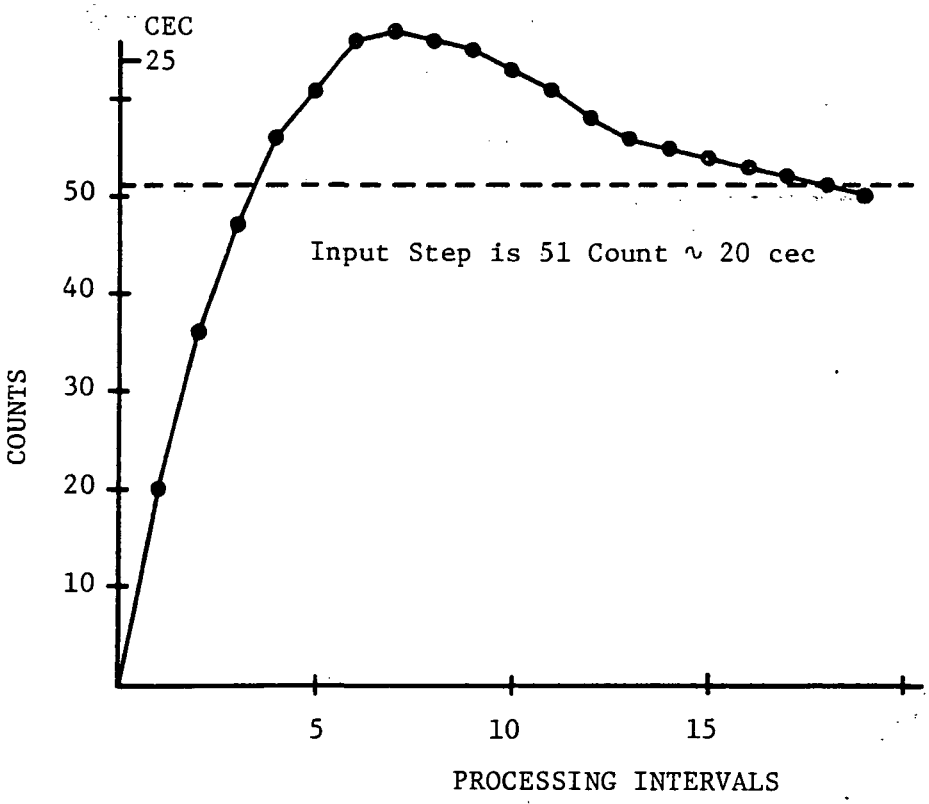


Figure 2-7. DPLL Step Response with K_1=16, K_3=4.

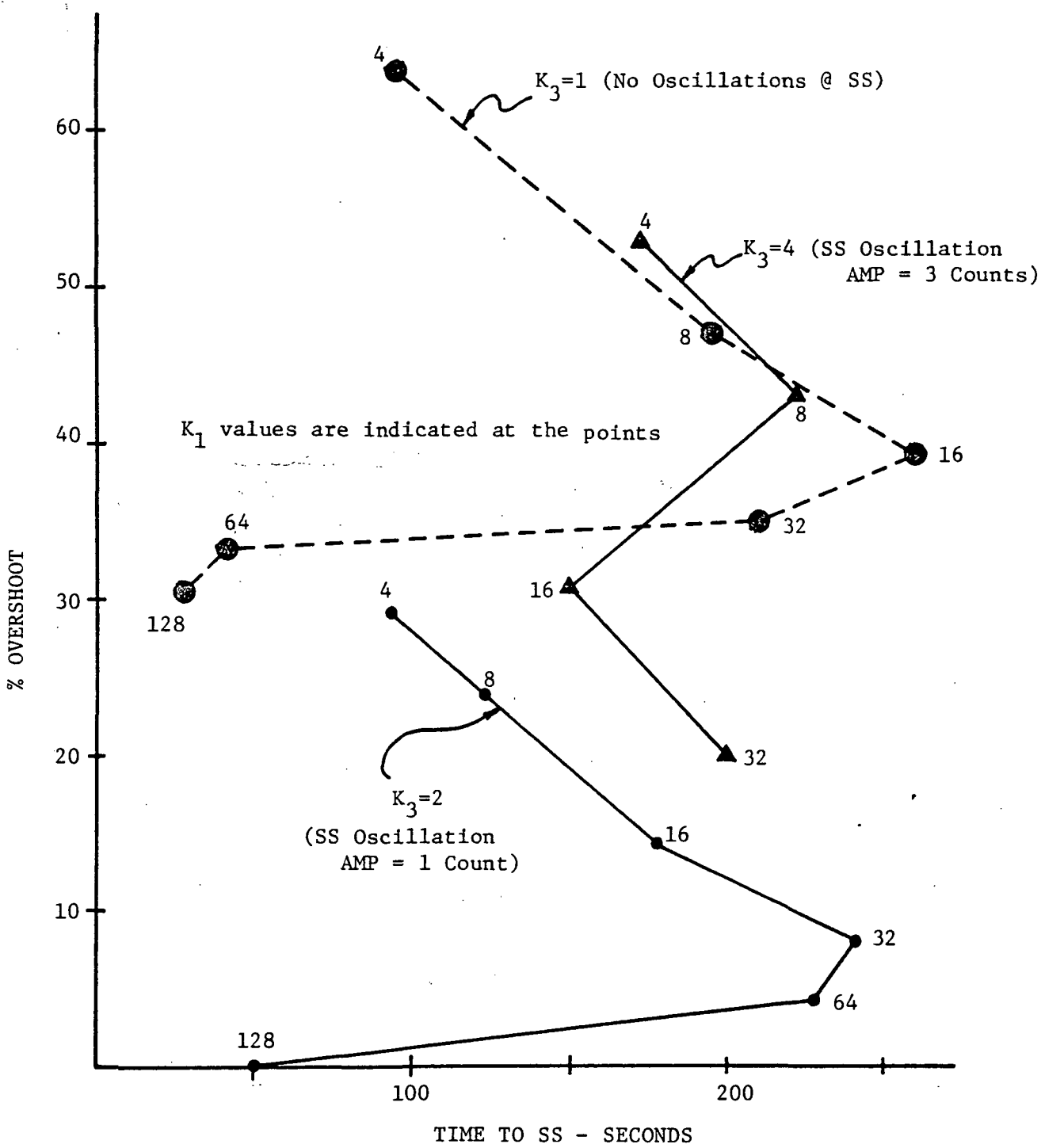


Figure 2-8. Summary of Step Responses.

defined as a "settling time", i.e., the time required for the magnitude of the loop oscillation to remain within one phase count ($\frac{1}{2.56}$ cec which is the resolution of the DPLL). From this summary it appears that a value $K_3 \geq 2$ and $K_1 > 8$ will be necessary to provide an overshoot of less than 20% and a satisfactory response time. It should be noted that in the second order DPLL overshoot will occur even when the analog equivalent damping factor is greater than 1.

Further analysis is based on evaluating DPLL response to input phase changes with time corresponding to receiver movement typical of a maneuvering aircraft. This includes phase ramp inputs which appear if any local oscillator phase drift occurs from one 10 second sample to the next. By definition a second order phase-locked loop will track a phase ramp input with no lag. This is valid for the DPLL in that the average lag is zero. In the digital implementation there is some oscillation in the loop response to a ramp input attributable to quantization. (See Appendix A and ref. 1).

In performing this analysis the receiver is simulated with an input phase corresponding to that which would be observed in a vehicle moving at some velocity VEL across an OMEGA lane until the loop is allowed to lock, making a 180° turn at any selected turn rate and moving back towards the origin phase line at a constant velocity. Figure 2-9 indicates a typical spatial trajectory of movement where the value $\dot{\theta}$ is the turn rate and t_1 is the time at which the turn is initiated. Figure 2-10 represents the corresponding phase plane plot of the received phase which would be observed at the vehicle moving in the trajectory of Figure 2-9. The function $\phi(t)$ is the received phase as a function of time assuming that the phase at time t_o is $\phi(t_o) = \phi_o$ in centicycles at 10.2 kHz. The analysis that follows is all done at the 10.2 kHz OMEGA frequency.

Let

VEL = aircraft velocity

and

$\dot{\theta}$ = aircraft turn rate in degrees/sec.

The received phase ramp input in cecs/sec corresponding to velocity VEL is

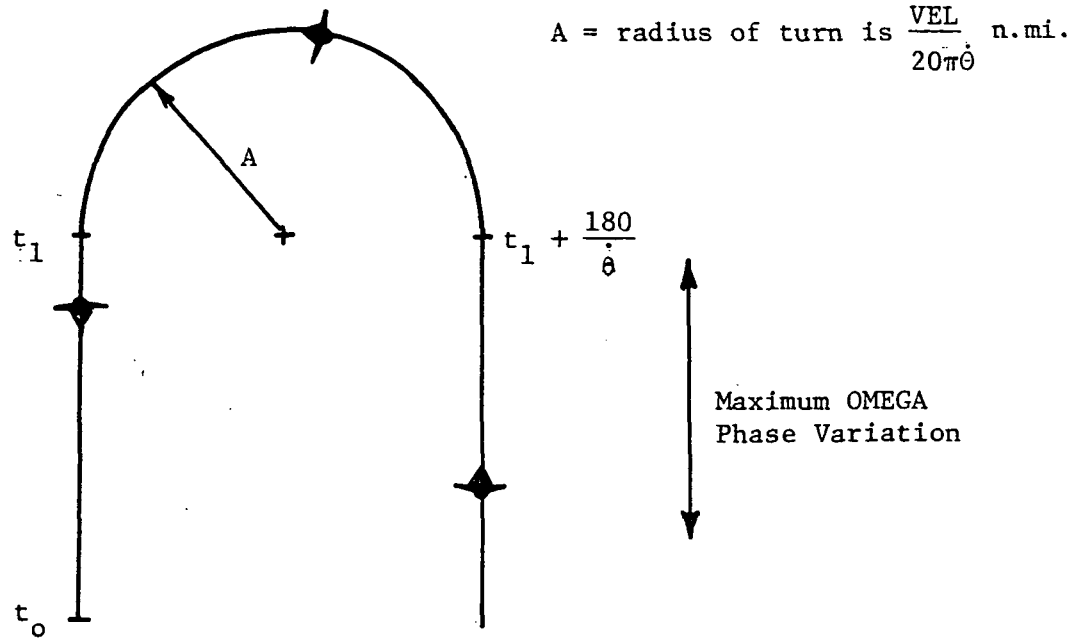


Figure 2-9. Spatial Flight Path Trajectory of Aircraft With Velocity VEL Knots and Turn-rate $\dot{\theta}$ degrees/sec.

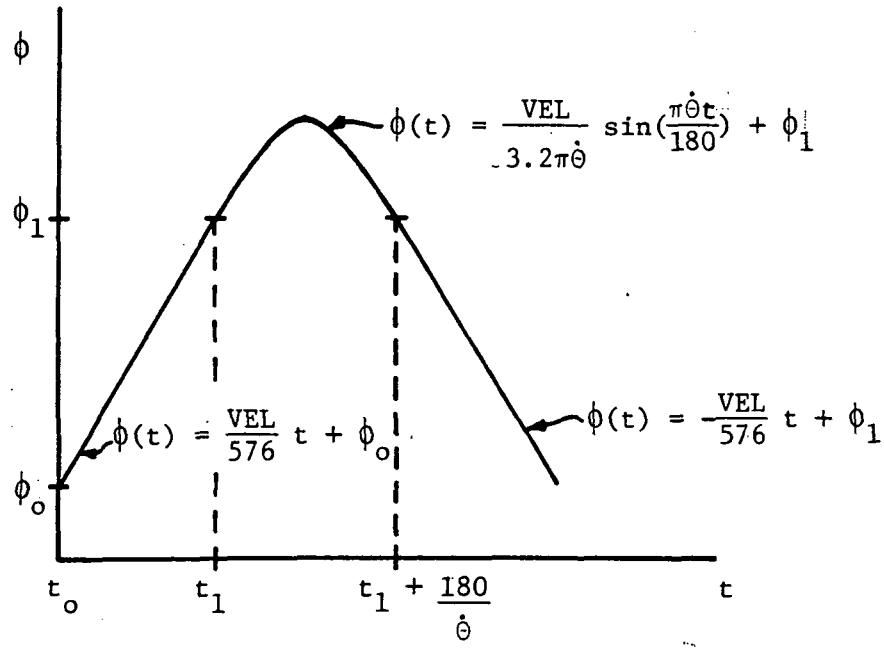


Figure 2-10. Phase Plane Trajectory of Flight Path Which Represents Received OMEGA Phase At Aircraft.

$$\dot{\phi} = \frac{VEL}{576} \quad .$$

At time t_1 the aircraft goes into a turn, thus the received phase at time t_1 is

$$\phi_1 = \dot{\phi} t_1 + \phi_o$$

where ϕ_o is initial received phase. During the turn the phase changes according to a sine function defined in terms of an offset, ϕ_1, an amplitude, A, and a period, T. The period T is defined in terms of the turn rate as

$$T = \frac{360}{\dot{\theta}} \text{ sec.}$$

The amplitude is defined as:

$$A = \frac{VEL \cdot T}{1152\pi} = \frac{VEL}{3.2\pi\dot{\theta}} \text{ cecs.}$$

The offset is just ϕ_1 such that at time t_1 the received phase varies with time according to

$$\phi(t) = A \sin \left(\frac{2\pi}{T} t\right)$$

or

$$\phi(t) = \frac{VEL}{3.2\pi\dot{\theta}} \sin \left(\frac{\pi}{180} \dot{\theta} t\right) + \phi_1$$

where VEL is in knots, T is in seconds and ϕ_1 is in cecs at 10.2 kHz.

At t = T/2 the phase ramp represents the received phase as

$$\phi(t) = -\dot{\phi}t + \phi_1$$

where

$$\phi(t_1) = \phi(t_1 + T/2) = \phi_1 \ .$$

In summary, input VEL in knots, ϕ_o in cecs, and $\dot{\theta}$ in degrees/sec so that

$$\phi = \frac{VEL}{576}$$

$$\phi(t) = \dot{\phi}t + \phi_o \qquad 0 \leq t < t_1$$

$$\phi(t) + \frac{VEL}{3.2\pi\dot{\theta}} \sin\left(\frac{\pi\dot{\theta}t}{180}\right) + \phi_1 \qquad t_1 \leq t < t_1 + T/2$$

$$\phi(t) = -\dot{\phi}t + \phi_1 \qquad t_1 + T/2 \leq t \ .$$

This function is plotted in Figure 2-10.

The DPLL response to this input is provided in Figures 2-11 and 2-12. In these plots the input phase driving function is represented as a sequence of solid line segments between 10 second sample values. All phase values are reduced to the interval (0, 100) cec so that phase discontinuities appear in the plots when phase changes from 100 to 0 cec. Actually 100 cec and 0 cec are the same phase value. Responses are plotted as dots at 10 second intervals. The abscissa is again labeled in units of the number of processing intervals. Results are for the loop tracking a phase change corresponding to a velocity of 576 knots along the half-racetrack (HT) pattern. All turns are at 3 degrees/sec. Initially all loop registers are zero and $\phi_o = 0$ in Figure 2-11 and $\phi_o = 20$ cecs in Figure 2-12. The initial leg of the HT pattern is traversed for 500 seconds (50 processing intervals) to allow the loop to lock to the phase ramp. For the situation in Figure 2-11 the maximum lags during turn and the time until the lags are within 1 cec upon completion

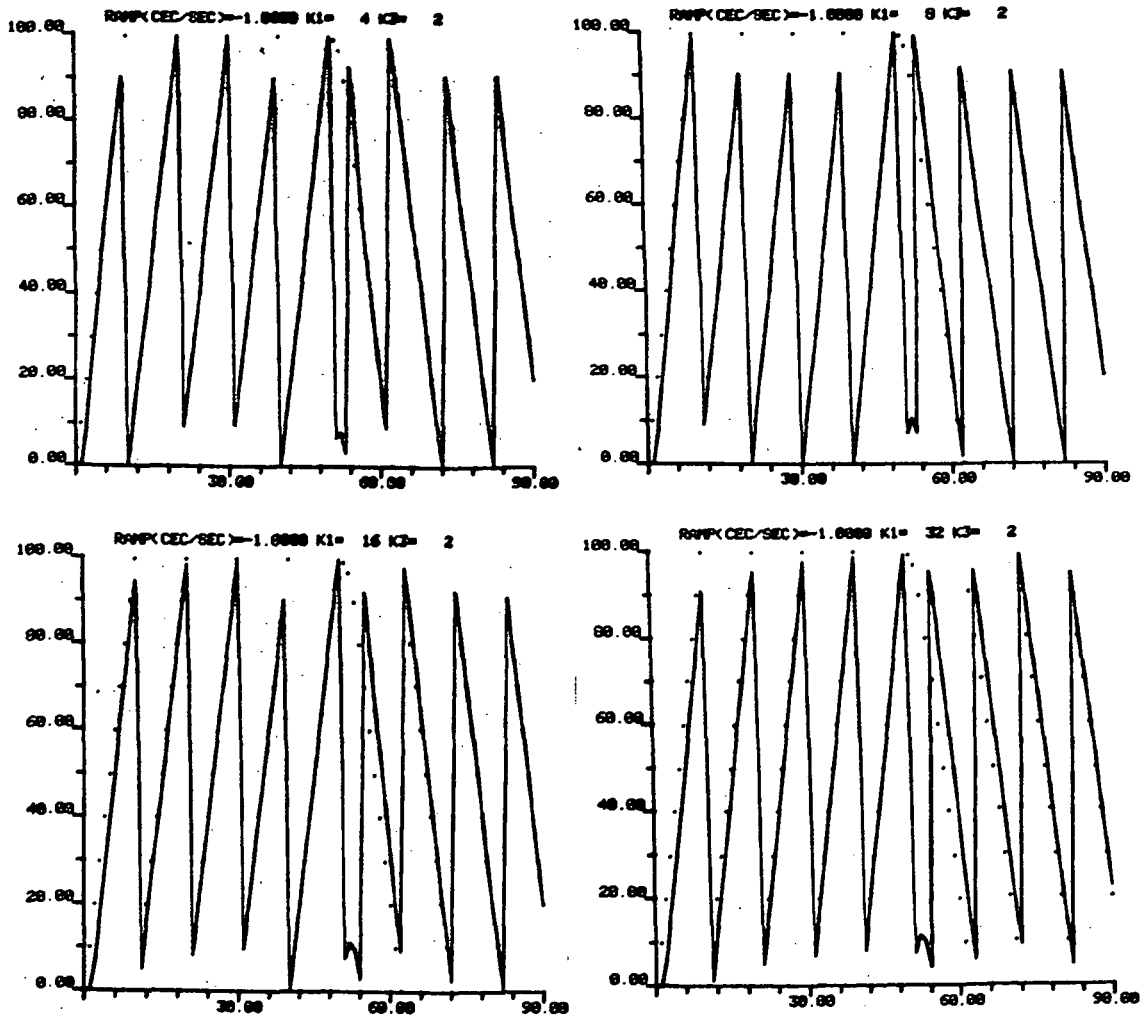


Figure 2-11. Loop Response in HT Pattern, VEL = 576 kts, $\dot{\theta}$ = 3°/sec, ϕ_o = 0.

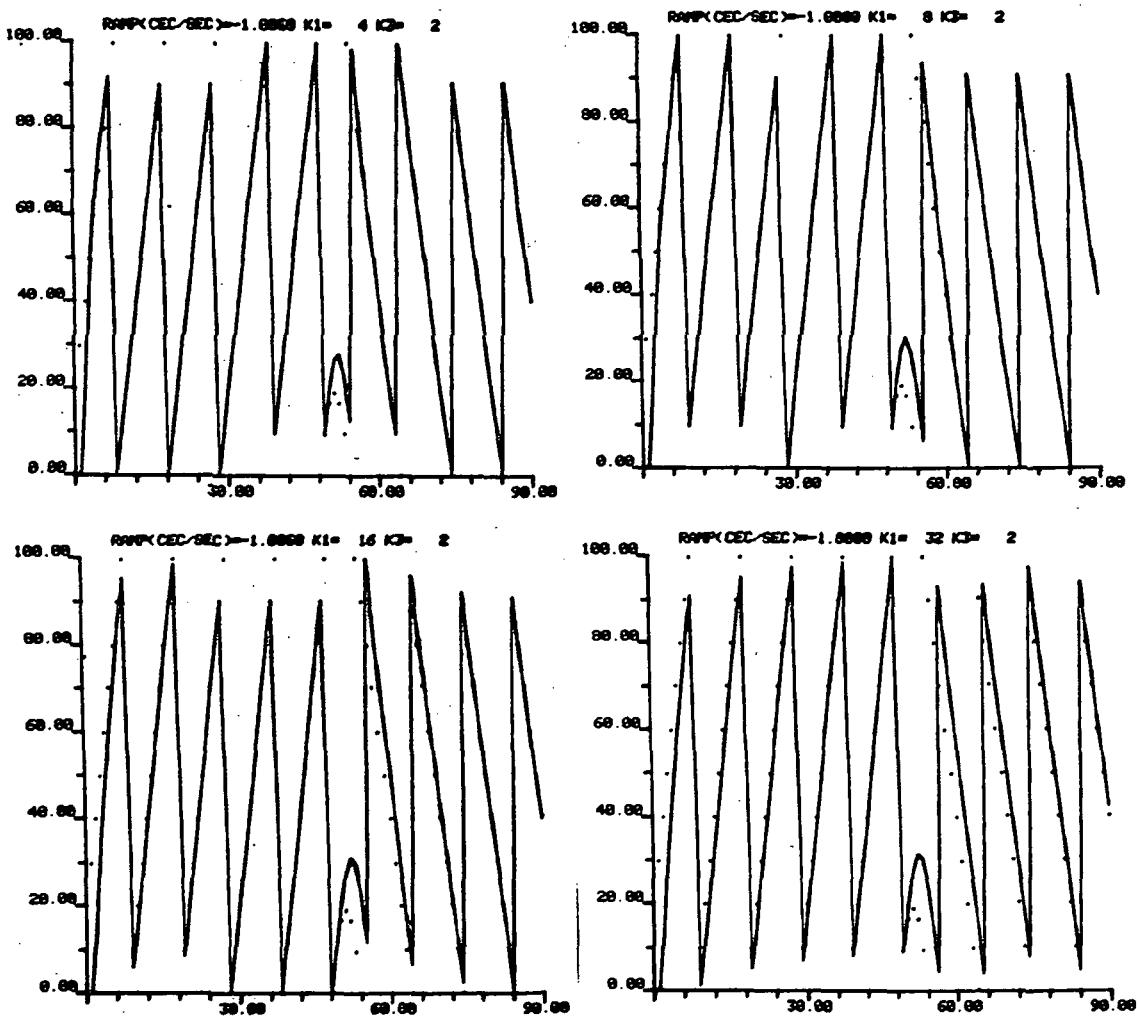


Figure 2-12. Loop Response in HT Pattern, VEL = 576 kts, $\dot{\theta}$ = 3°/sec, ϕ_o = 20 cec.

of the turn as tabulated in Table 2-1 for various values of K1 (4, 8, 16, 32). In Figure 2-12 a value of ϕ_o = 20 cecs is used so that the phase trace during the turn can be illustrated better. The initial lock-up time is somewhat longer for the situation in Figure 2-12 but the loop behavior in the turn and coming out of the turn are essentially the same as for Figure 2-11 as summarized in Table 2-1.

In Figures 2-13 and 2-14 several different velocities are attempted for K_1 values of 4, 8, 16, and 32. Phase ramp values of 0.1 correspond to VEL $\approx$ 58 kts, 0.5 to VEL $\approx$ 288 kts, 1.0 to VEL = 576 kts, 1.2 to VEL = 691 kts, 1.5 to VEL $\approx$ 664 kts, 2.0 to VEL = 1152 kts, and 2.5 to VEL = 1440 kts. Notice that a loop with K_1 = 4 does not lose lock (skip cycles) until velocity is greater than 1152 kts whereas the other two K_1 values skip cycles at lower velocity values.

Based on these analysis a value of K_3 = 2 and K_1 = 16 is recommended for the DPLL of Figure 2-5. These provide an inpulse response time constant of approximately one processing interval, a slightly overdamped response with a 10-15% step response overshoot, a reasonable time to steady state ($\sim$ 200 seconds) with a phase ramp corresponding to an initial instantaneous velicity of 576 knots,and no loss of lock in a 3°/sec turn at this speed.

2.4 Antennas for Airborne Use

The DPLL receiver is designed for aircraft use and the antenna selection must be based on characteristics of the airborne environment. Two basic types of OMEGA antennas are available. These are E-field antennas which include the whip, short stub, and the plate antenna. These may have active matching circuitry or be strictly passive antennas. The other type is the H-field antenna normally implemented in the form of a crossed-loop antenna which has associated with it a steering mechanism to orient the directional sensitivity of the antenna. In the airborne environment the E field antenna normally offers the advantage of providing a better SNR to the receiver and can be much simpler,however,disadvantages include susceptability to precipitation static (P-static) and possible interference from on-board aircraft power (normally 400 HZ). The H-field antenna offers

TABLE 2-1

LAG Phase Values and Times to Lock-up in HT Pattern.

K3 = 2

K1	Time To* Initial Lock (sec)	Max Lag In Turn (cec)	Time To Error < 1 cec (sec after max lag)
4	50	13.7	80
8	90	19.1	110
16	220	23.0	240
32	440	25.0	530

*Initial lock is defined as earliest time loop phase remains less than 1 cec from received phase.

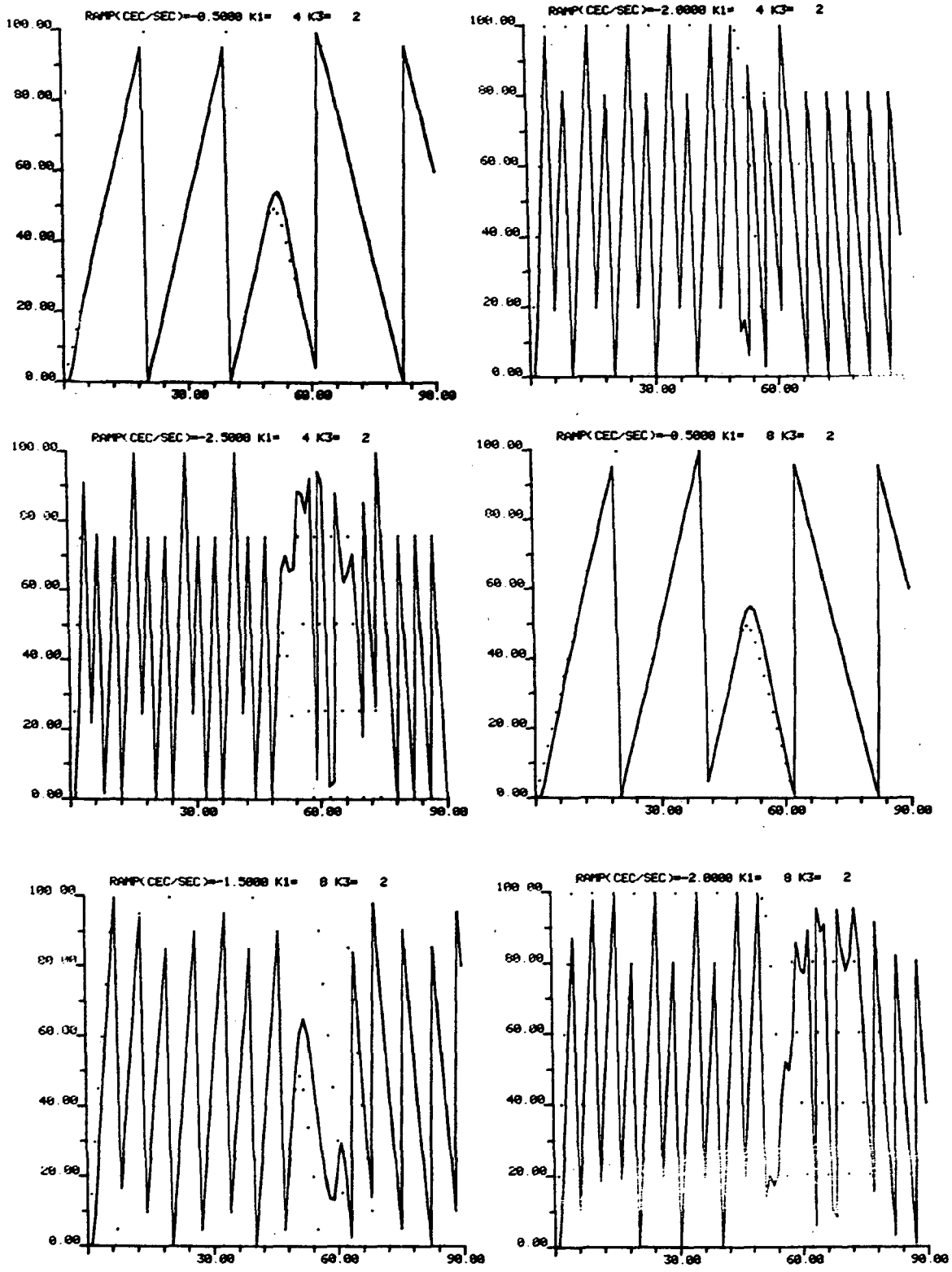


Figure 2-13. Loop Responses in HT Pattern at Selected Velocities for K_1 = 4 and 8.

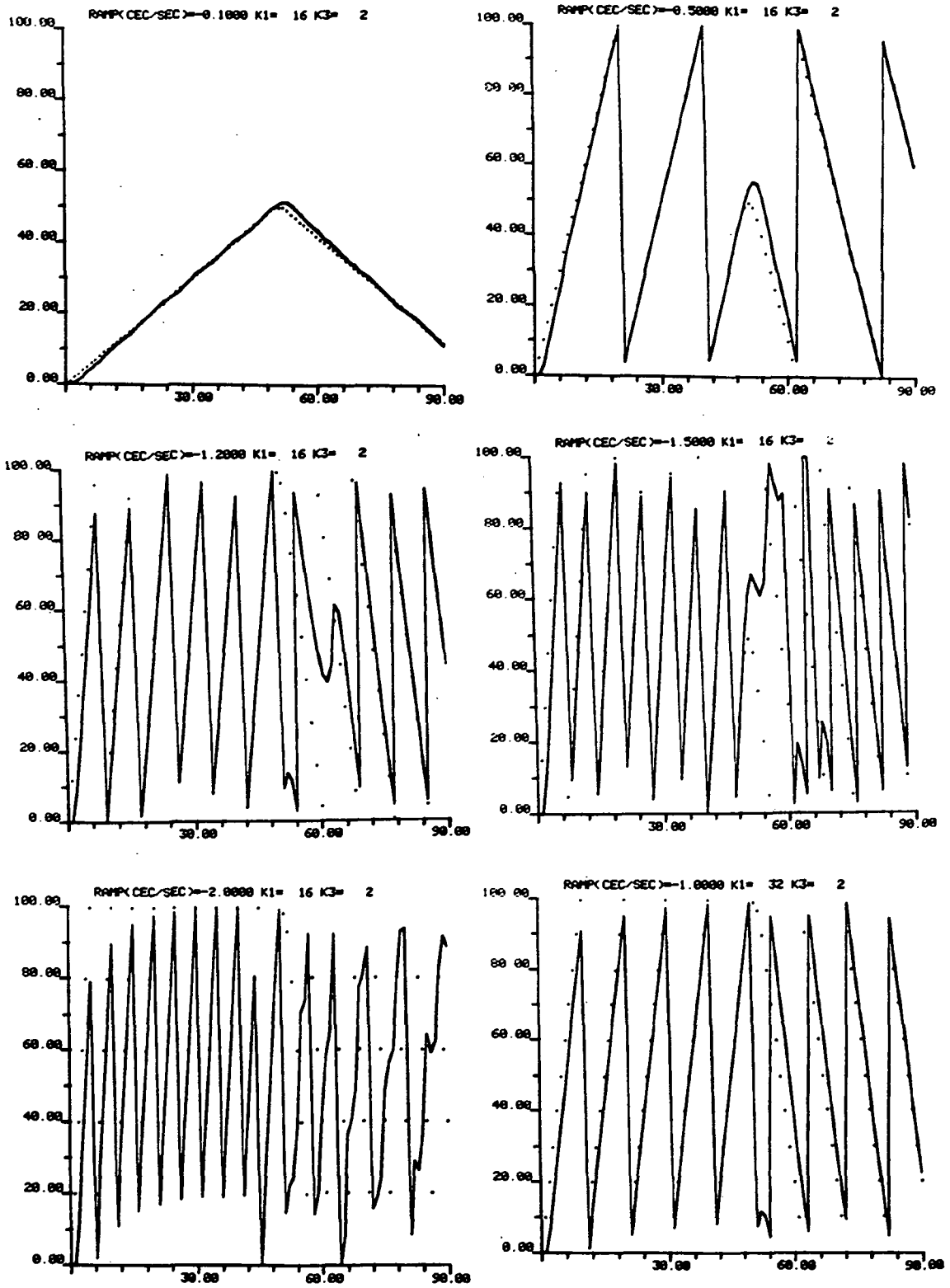


Figure 2-14 Loop Responses in HT Pattern at Selected Velocities for K_1 = 16 and 32.

immunity from the P-static and E-field noise associated with the aircraft power. However, active matching networks are required to provide adequate SNR and some form of stearing is necessary since the loop antenna is directional and must be "steered" to provide omnidirectional capability needed for a navigation receiver. Table 2-2 provides summary characteristics for a representative sample of airborne antennas. The plate antenna or the short stub (blade) seem to offer the least of the disadvantages and should provide adequate signal strengths in an aircraft.

2.5 Use of the Intel 4004 Microprocessor

In developing the feasibility model of the microprocessor based OMEGA receiver two tools were needed to facilitate software development. One was the Intel Macro Assembler (ref. 2), purchased by NASA-LRC, to provide for assembly of microprocessor code on the CDC-6600 computer. Even though the assembler is written as a portable software package for any computer with at least a 32-bit word, several modifications were necessary to provide for use with the CDC-6600. Appendix E details these modifications. To facilitate use of the 4004 microprocessor, RTI personnel wrote a FORTRAN IV simulation program, SIM4. This program provides for a full complement of RAM (read and write memory), ROM (read-only memory) and associated input and/or output ports. SIM4 may be executed in either batch or interactive mode and requires a machine word of at least 32 bits. Appendix F is a user manual for the SIM4 simulator.

Table 2-2 Typical OMEGA Antenna Characteristics

Antenna Type	Frequency Response	Effective Heights	Sensitivity $\mu v/m/\sqrt{Hz}$	Maximum Outputs	Size	Weight	Power Required
Crossed Loop Spears Type 6-7144	10-14kHz	>1m @ 400^{Ω} >0.5m @ 100^{Ω}	<0.5	1v into 400^{Ω} 0.5v into 100^{Ω}	6" x 6" x 1.75"	3 lbs	12v, 50ma
Plate Spears Model 717	10-14kHz	1m @ 400^{Ω} 0.5m @ 100^{Ω}	<2	1v into 400^{Ω} 0.5v into 100^{Ω}	7" x 12" x 0.75"	5 lbs	12v, 40ma
Blade Bayshore UPS-1908	10-14kHz	10cm @ 50^{Ω}	~1	9_V @ 50^{Ω}	8.5" x 2" x 5"	2 lbs	24-30v, 80ma

"Page missing from available version"

page 28

3.0 NAVIGATION ALGORITHMS FOR OMEGA NAVIGATION

3.1 General

An OMEGA receiver with computational capability included can become a sophisticated navigation equipment. It can provide to the navigator output information such as corrected position estimates in any desired coordinate system, estimates of vehicle velocity, deviation from desired courses or routes, and position and time relative to defined destination or origin points. This chapter presents a discussion of receiver navigation algorithms to be included in the software associated with the OMEGA microprocessor-based receiver which is being developed by NASA-LRC personnel. Included in this presentation is documentation of the navigation equations to be implemented in software, an evaluation of some of these algorithms, and specific analysis with respect to computational limitations in the software developed for the chosen microprocessor.

3.2 Navigation Outputs

The OMEGA navigation receiver is designed to provide navigation capability at the 10.2 kHz carrier frequency and at the 3.4 kHz difference frequency. The 3.4 kHz difference frequency phase is generally more stable in terms of diurnal variations but does not offer the precision of the 10.2 kHz phase measurement. It is visualized that the 3.4 kHz navigation will be adequate for enroute flight and the 10.2 kHz carrier navigation will be needed in the vicinity of a runway. The receiver is set up to provide phase measurements from up to four OMEGA transmitters at both the 13.6 and 10.2 kHz carrier frequencies. The navigation algorithm provides LOP measurements at 10.2 and 3.4 kHz continuously so that navigation outputs can be had by the navigator in either mode at any time by using a front panel mode switch. Switching between modes at any time will not cause any loss of lane count. In implementation the navigation algorithm is set up to make all calculations in 10.2 kHz units regardless of mode. Certain inputs are required at the time of initialization of the receiver. These are used with phase measurements to provide outputs of relative position.

The navigation receiver is designed to operate on a point to point basis and employs memory to define current position and up to two waypoints in terms of OMEGA LOP intersections. In operation a front panel rotary switch and a sixteen key pushbutton keyboard is provided to allow user input and display stored and calculated output. A three digit LED display provides keyboard echo display and information display for stored outputs. Further display lights indicate the direction of cross-track error with the left arrow doubling as an algebraic negative sign indicator. Two front panel lights serve to indicate the active waypoint and double as synchronization indicators during the OMEGA format synchronization mode. (See Figure F-6)

Two rotary switch positions and double function keyboard inputs provide for operation input outlined in Table 3-1. Action inputs include synchronization (SYNC) which is used to initially synchronize the receiver to the OMEGA broadcast format and course designation input which defines a desired course in terms of current position information and data associated with the waypoints. From the allowable keystroke sequences given in Table 3-1 there are four possible courses which can be set up: current position to either waypoint; waypoint to waypoint with either as origin.

The navigation algorithm uses calculated LOP phase values for station pairs defined by the operator to calculate cross-track deviation, heading to designated waypoint, distance to destination (designated waypoint), ground track velocity, current heading, difference between current heading and heading to designated waypoint, and estimated time of arrival at designated destination waypoint. Table 3-2 lists these navigation outputs.

In calculating the desired outputs the implicit assumption used in defining the navigation equations is that LOPs are straight lines. All calculations are defined in LOP coordinate space based on position information input by the navigator, average lane widths and propagation prediction corrections input by the navigator, linear geometry, and LOP calculations from phase measurements of designated transmitter pairs. Figure 3-1 defines the coordinate system and the geometry of the calculations. LOP_x and LOP_y are defined in terms of OMEGA transmitter station pairs using the "S" keyboard entry of Table 3-1. Lane widths in n.mi./lane are input using a "Δ" entry, PPC corrections are input using the "D" entry (10.2 corrections are made in the 10.2 kHz MODE and 3.4 corrections are made in the 3.4 kHz MODE), and LOP crossing angle is input using the "ϕ" entry.

TABLE 3-1

Inputs to OMEGA Navigation Receiver
Note: Q/P indicates "Q" or "P"

RSW	KEYSTROKE 1	2	3	4	5	6	FUNCTION
0	SYNCH						To initiate synchronization to OMEGA format
0	Δ	x/y	0-9	0-9	0-9	E+	Enter x or y lane width miles per lane of 10.2 range 00.0 - 99.9
0	ϕ	x/y	0-9	0-9	0-9	E+/E-	Enter LOP x or y angle to gradient from north CW+ $-180 \leq \phi_x, \phi_y \leq +180$
0	S	x/y	1-7	1-8 < key1	E+		Enter station pair for LOP x or y 1=A,2=B,3=C, 4=D,5=E,6=F,7=G,8=H Pairs in numerical order
1	P_0	x/y	0-9	0-9	0-9	E+	LOP x or y whole lane value at origin - multiples of 3
1	p_0	x/y	0-2	0-9	0-9	E+	LOP x or y inner lane value (0,1 or 2) and fractional lane value at origin $0.00 \leq p_{0x}, p_{0y} \leq 2.99$
1	P_1	x/y	0-9	0-9	0-9	E+	Same as P_0 except at waypoint "1"
1	p_1	x/y	0-2	0-9	0-9	E+	Same as p_0 except at waypoint "1"
1	P_2	x/y	0-9	0-9	0-9	E+	Same as P_0 except at waypoint "2"
1	p_2	x/y	0-2	0-9	0-9	E+	Same as p_0 except at waypoint "2"
1	D	x/y	0-2	0-9	0-9	E+/E-	LOP x or y correction If 10.2 mode $-.5 < D_x, D_y \leq +.5$. If 3.4 mode $-1.5 < D_x, D_y \leq +1.5$
1	$P_0/WP_1/WP_2$	WP_1/WP_2	E+				Used to activate course either (current pos) to waypoint "i" or waypoint "j", i,j = 1 or 2, $i \neq j$

TABLE 3-2

Function Switch Related Navigation Outputs

WP = waypoint; all azimuth angles relation to north

RSN	FUNCTION	INFORMATION DISPLAYED
3	DEG. (TO WP)	Integer degrees. CW azimuth from current position to active waypoint
4	N.MI. (TO WP)	Distance from current position to active waypoint (DTD)
5	DEG. (ERROR)	Integer degrees. Difference between current heading and deg. (to WP)
6	N.MI. (ERROR)	Cross-track deviation (CTD)
7	MIN. (ERROR)	Relative estimated time of arrival at active waypoint (ETA)
8	DEG. (TRACK)	Integer degrees. CW bearing of current heading
9	KTS. (TRACK)	Ground track velocity

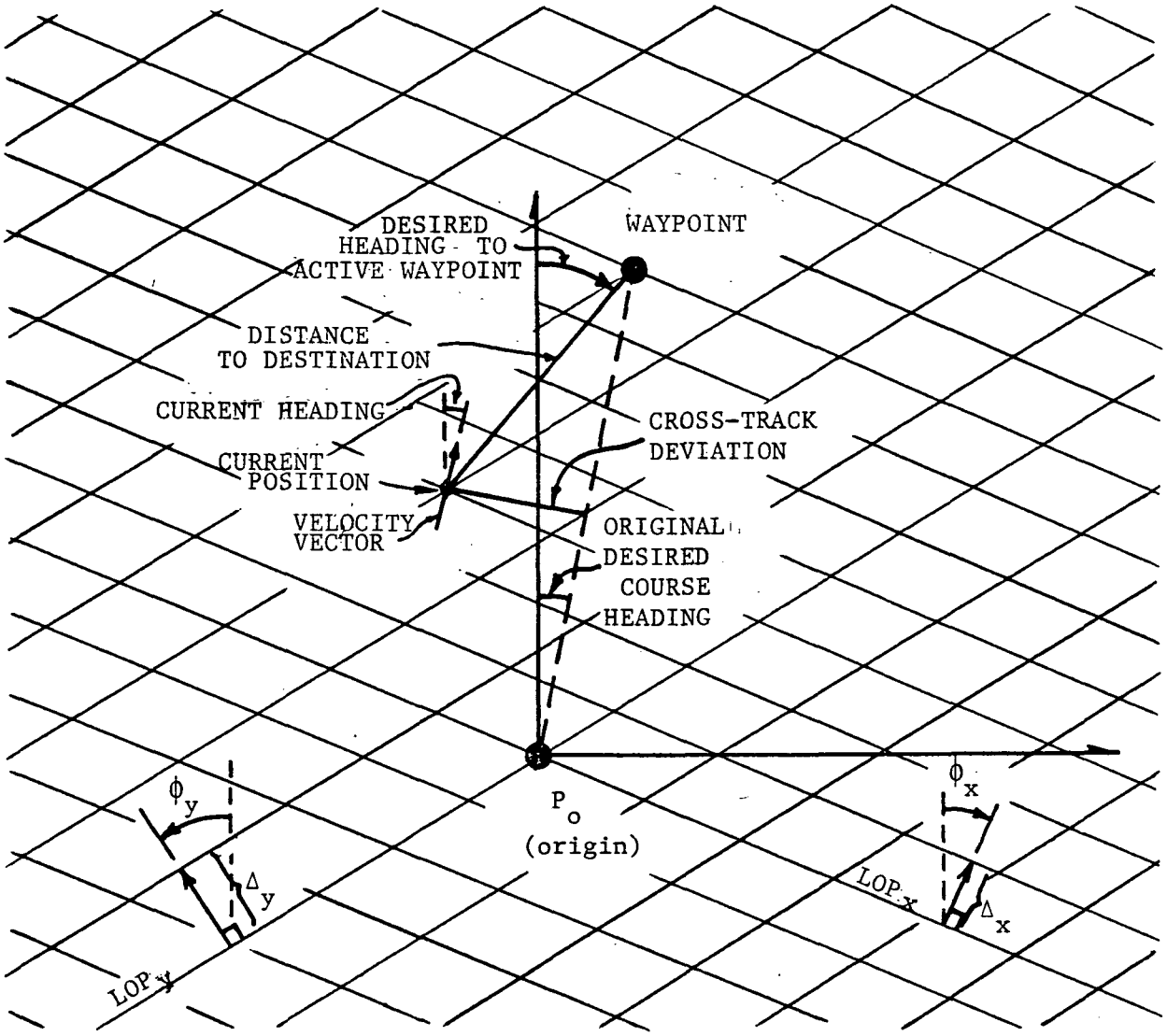


Figure 3-1. Geometrical Definition of OMEGA Receiver Navigation Space.

Other input sequences given in Table 3-1 provide for user definition of current position (this is origin upon initial setup) and waypoint definition in terms of LOP_x and LOP_y coordinate position.

To display input information a separate rotary switch position is designated. Table 3-3 lists the keystroke sequences required to obtain the different readouts. Navigation equation results are displayed using the rotary switch positions given in Table 3-2.

3.2.1 Definition of navigation equations.— The navigation equations consist of those algebraic functions used to determine the various outputs which can be displayed to the navigator during operation of the receiver. Each point in space (horizontal plane) of interest for a given flight path and the information relative to these points was illustrated in Figure 3-1. In making these calculations, differences in LOP phase values are transformed into ground distances in an X,Y (ΔE, ΔN) rectilinear coordinate system so that standard rectilinear and trigonometric relationships can be used to calculate the navigation parameters of interest.

Figure 3-2 defines the relationships between an LOP_x, LOP_y space and a ΔE, ΔN rectilinear space necessary to define the transformation of position coordinates in LOP space to $\Delta E, \Delta N$ space. From Figure 3-2

$$\delta\overline{LOP} = [M]\ \bar{r}$$

where

$$\delta\overline{LOP}^T = [\delta LOP_x\ \ \delta LOP_y]\ , \quad \bar{r}^T = [\Delta E, \Delta N]$$

and

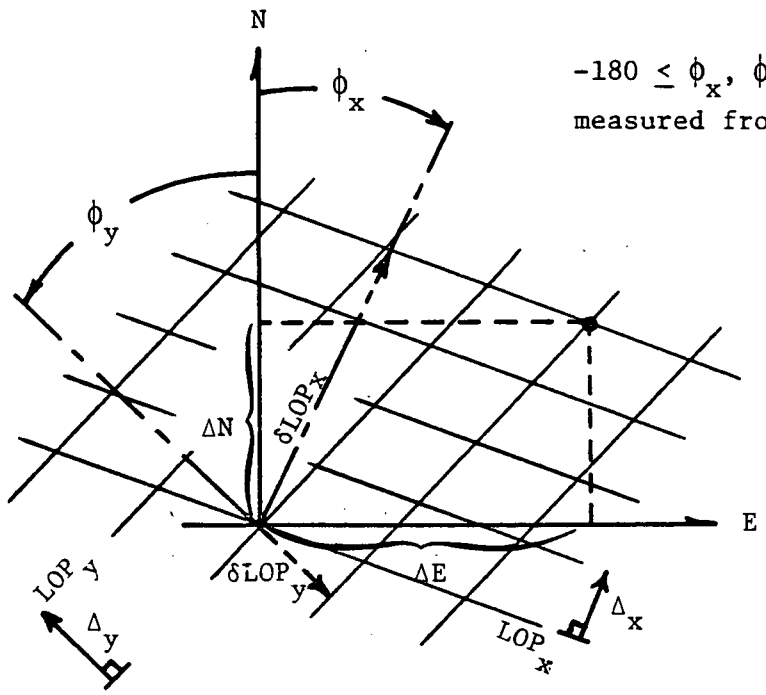
$$[M] = \begin{bmatrix} \Delta_x^{-1} \sin \phi_x & \Delta_x^{-1} \cos \phi_x \\ \Delta_y^{-1} \sin \phi_y & \Delta_y^{-1} \cos \phi_y \end{bmatrix} .$$

Here Δ_x, Δ_y are the input constants defining the lane widths in n.mi. (n.mi./100 cec) for the LOP_x and LOP_y lanes in the geographical area of interest, (ϕ_x, ϕ_y) are the angles measured from north to the gradient vector of the respective LOP's and (δLOP_x, δLOP_y) are the components of

TABLE 3-3

Keystrokes to Read-Back Keyboard Entries (RSW = 2)

KEYSTROKES 1	2	FUNCTION
P_0	x/y	Display lane value of LOPx or LOPy at current position
p_0	x/y	Display inner lane and fractional lane value of LOPx or LOPy at current position
P_1	x/y	Same as P_0 except at waypoint "1"
p_1	x/y	Same as p_0 except at waypoint "1"
P_2	x/y	Same as P_0 except at waypoint "2"
p_2	x/y	Same as p_0 except at waypoint "2"
Δ	x/y	Display lane width value for LOPx or LOPy
D	x/y	Display PPC for LOPx or LOPy. MODE switch determines 10.2 or 3.4
ϕ	x/y	Display angle of gradient for LOPx or LOPy
S	x/y	Display station pair for LOPx or LOPy



$$\delta LOP_x = \frac{+1}{\Delta_x} \sin\phi_x \Delta E + \frac{1}{\Delta_x} \cos\phi_x \Delta N$$

$$\delta LOP_y = \frac{+1}{\Delta_y} \sin\phi_y \Delta E + \frac{1}{\Delta_x} \cos\phi_y \Delta N$$

Figure 3-2. Geometry of Transformation Between Rectilinear Coordinate Space (E, N) and LOP Linear Space (LOP_x, LOP_y)

the vector between two points in the LOP_x, LOP_y coordinate space. The position relative to some arbitrary origin in terms of the rectilinear coordinate components of the vector between two points is defined as $(\Delta E, \Delta N)$.

Given the difference in LOP phase units between two points $\delta\overline{LOP}$, the difference in distance $\overline{r}$ is determined using

$$\overline{r} = [M]^{-1} \delta\overline{LOP}$$

where

$$[M]^{-1} = \begin{bmatrix} K_{xe} & K_{ye} \\ K_{xn} & K_{yn} \end{bmatrix} \tag{3-1}$$

with

$$K_{xe} = \frac{\Delta_x \cos\phi_y}{\sin(\phi_x - \phi_y)}$$

$$K_{xn} = \frac{-\Delta_x \sin\phi_y}{\sin(\phi_x - \phi_y)}$$

$$K_{ye} = \frac{-\Delta_y \cos\phi_x}{\sin(\phi_x - \phi_y)}$$

$$K_{yn} = \frac{\Delta_y \sin\phi_x}{\sin(\phi_x - \phi_y)}$$

where $-180 \leq \phi_x, \phi_y \leq +180$ and the k_{ij} are in units of n.mi./lane when phase is in units of 10.2 kHz cycles.

Upon receiver initialization the origin position is defined in terms of LOP coordinates P_{0x}, p_{0x} and P_{0y}, p_{0y} which is also the initial current position. P_{0x}, P_{0y} are whole lane values while p_{0x}, p_{0y} are fractional

lane values (10.2 units). Similarly waypoints are defined in terms of the P_{1x}, p_{1x}, P_{1y}, p_{1y} set. To obtain desired course heading assuming destination as waypoint "1" the difference in LOP coordinates is calculated as

$$\delta LOP_x = \left(P_{1x}+p_{1x}\right)-\left(P_{0x}+p_{0x}\right)$$

$$\delta LOP_y = \left(P_{1y}+p_{1y}\right)-\left(P_{0y}+p_{0y}\right)$$

Using the transformation $[M]^{-1}$ the relative position vector components $\Delta E_1(0), \Delta N_1(0)$ are calculated to yield the desired course heading $\theta_D(0)$ as

$$\theta_D(0) = \arctan\left\{\frac{\Delta E_1(0)}{\Delta N_1(0)}\right\} \qquad (3\text{-}2)$$

In the algorithm implementation, if $\Delta N_1(0) < \Delta E_1(0)$ then the calculation becomes

$$\theta_D(0) = 90 - \arctan\left(\frac{\Delta N_1(0)}{\Delta E_1(0)}\right) \qquad (3\text{-}3)$$

The distance between the two points is the original distance to destination calculated as

$$DTD(0) = \frac{\Delta N_1(0)}{\cos\theta_D(0)} \qquad \Delta N_1(0) \geq \Delta E_1(0)$$

$$= \frac{\Delta E_1(0)}{\sin\theta_D(0)} \qquad \Delta N_1(0) < \Delta E_1(0)$$

As the vehicle with the receiver progresses on a path from origin to destination the current position is updated using the phase-locked loop measured phase values to form LOP phase differences. To obtain a velocity

estimate it is necessary to estimate the phase velocity $\frac{d\phi}{dt}$ in each loop to form an estimated LOP phase velocity. The loop output phase value differences can be used to form the derivative of LOP phase. Some smoothing algorithm could be used to form a stable estimate; however, the precision of the loop output values is only one part in 256 (~11 kts with minimum 8 n.mi. lane width) which is not adequate. It should be stated that a good velocity estimate using OMEGA, if a vehicle is in a maneuvering situation, is probably not reasonable with ten second updates. Additionally, normal OMEGA noise (spikes of 2cec) will make it necessary to use some form of smoothing (low-pass filtering) with introduced estimate lag. This update rate problem and consideration of the pilot or navigator needs resulted in the implementation of a velocity estimation technique which is only accurate during essentially straight and level flight (i.e., other than maneuvering situations involved significant accelerations).

The implementation of the digital phase-locked loops involves storing in RAM memory the integrator value associated with the second order loop (see Figure 2-5). This integrator accumulates the discriminator output in a 12-bit register with a resolution of one part in 4096 (4096 counts = 1 lane; 0.7 kts with minimum 8 n.mi. lane width). This integrator is a measure of the phase velocity associated with a given loop. This phase velocity measure includes clock drift, clock frequency offset phase drift, noise, and phase change due to receiver movement. Considering the difference of two integrator values in estimating an LOP phase rate, the clock frequency offset phase drift is cancelled out, but, instantaneous noise may increase depending on the degree of correlation between the noise in the two loops associated with each LOP. Clock drift is insignificant so that the LOP integrator difference represents a noisy vehicle velocity estimate in fractional lanes per ten seconds. Using data on flight ONR-12*(September, 1977) conducted by NASA personnel, the effect of using ten second sampled integrator values for velocity estimates has been investigated. The flight involved three straight legs between VOR sites SNOW HILL (Md.), SEA ISLE (N.J.), and KENTON (Del.) (see Washington, D.C., sectional aeronautical chart) north of the NASA Wallops Flight Center facility. On the first leg (SNOW HILL to SEA ISLE) the receiver was not set up properly for the entire leg so the data were unreliable. The

* See Chapter 5 for description of a similar flight ONR-13

second and third leg data were used to form ten second velocity estimates during straight and level flight. Figure 3-3 illustrates $\frac{d\phi}{dt}$ for the HAW-TRI LOP for the second leg (SEA ISLE to KENTON) and Figure 3-4 illustrates $\frac{d\phi}{dt}$ for the same LOP for the third leg (KENTON to SNOW HILL). The overall mean of the ten second samples represents a good estimate of LOP velocity in each situation although on leg three there is some long term fluctuation probably attributed to vehicle flight variations from the intended straight course. The significant problem is the large sample-to-sample variation. With this, any velocity estimate would be unreliable.

A simple smoothing algorithm has been postulated and evaluated using this data. Instead of using the LOP integrator difference value directly, an average, $\overline{INT(t)}$ is formed using the generating function

$$\overline{INT(t)} = k\ INT(t) + (1-k)\ \overline{INT(t-10)}$$

where $\overline{INT(t-10)}$ is the average ten seconds before the current sample INT(t) is read. The constant k<1 is conveniently a power of two ($k = 2^{-n}$ $n = 1, 2, \ldots$). Two weighting values have been investigated corresponding to n = 2 and 3. In Figure 3.3 the 1/4 weighting corresponds to n = 2 with a start at the beginning of the data. The 1/8 weighting is for n = 3. The initial lock-up is not really meaningful. The filter response after 10 - 15 ten second samples represents the "steady state response." Both offer considerable smoothing with 1/8 weighting being better. In Figure 3-4 the 1/8 weighting response is shown. Note that the response tracks the mean of the data well with a lag on the order of 80 seconds. The step response time constant of the 1/8 weight filter is approximately 80 seconds.

Using the average LOP integrator values, the $[M]^{-1}$ transformation of (3-1) is used to form $v_E(t)$ and $v_N(t)$, the east and north components of velocity at sample time t in units of knots,

$$\begin{bmatrix} v_E(t) \\ v_N(t) \end{bmatrix} = 360 \cdot \begin{bmatrix} K_{xe} & K_{ye} \\ K_m & K_{yn} \end{bmatrix} \begin{bmatrix} \overline{INT(t)_x} \\ \overline{INT(t)_y} \end{bmatrix} .$$

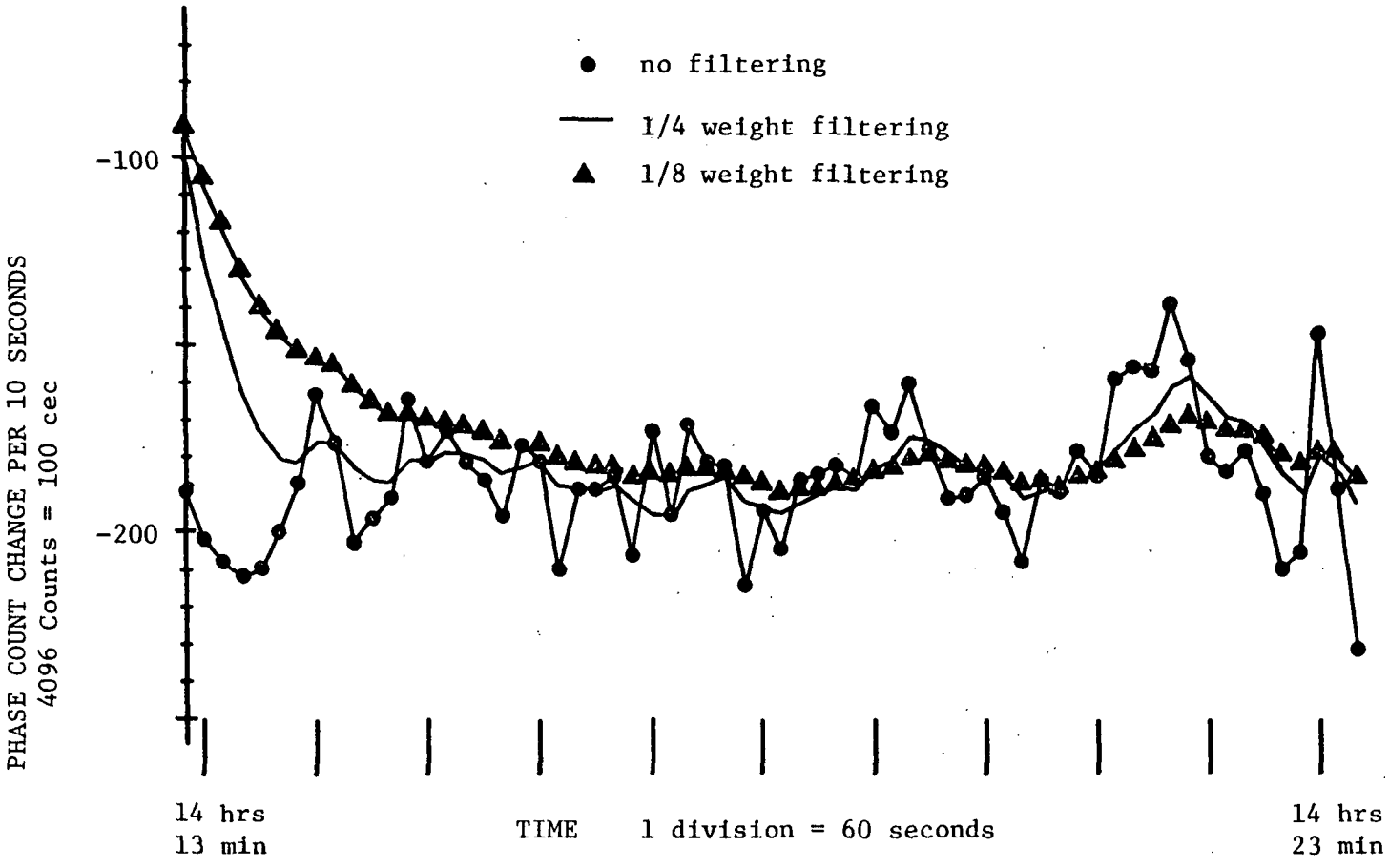


Figure 3-3. HAW-TRI Loop Integrator Difference Values With and Without Filtering (LEG 2, Flight ONR-12, 10.2 kHz)

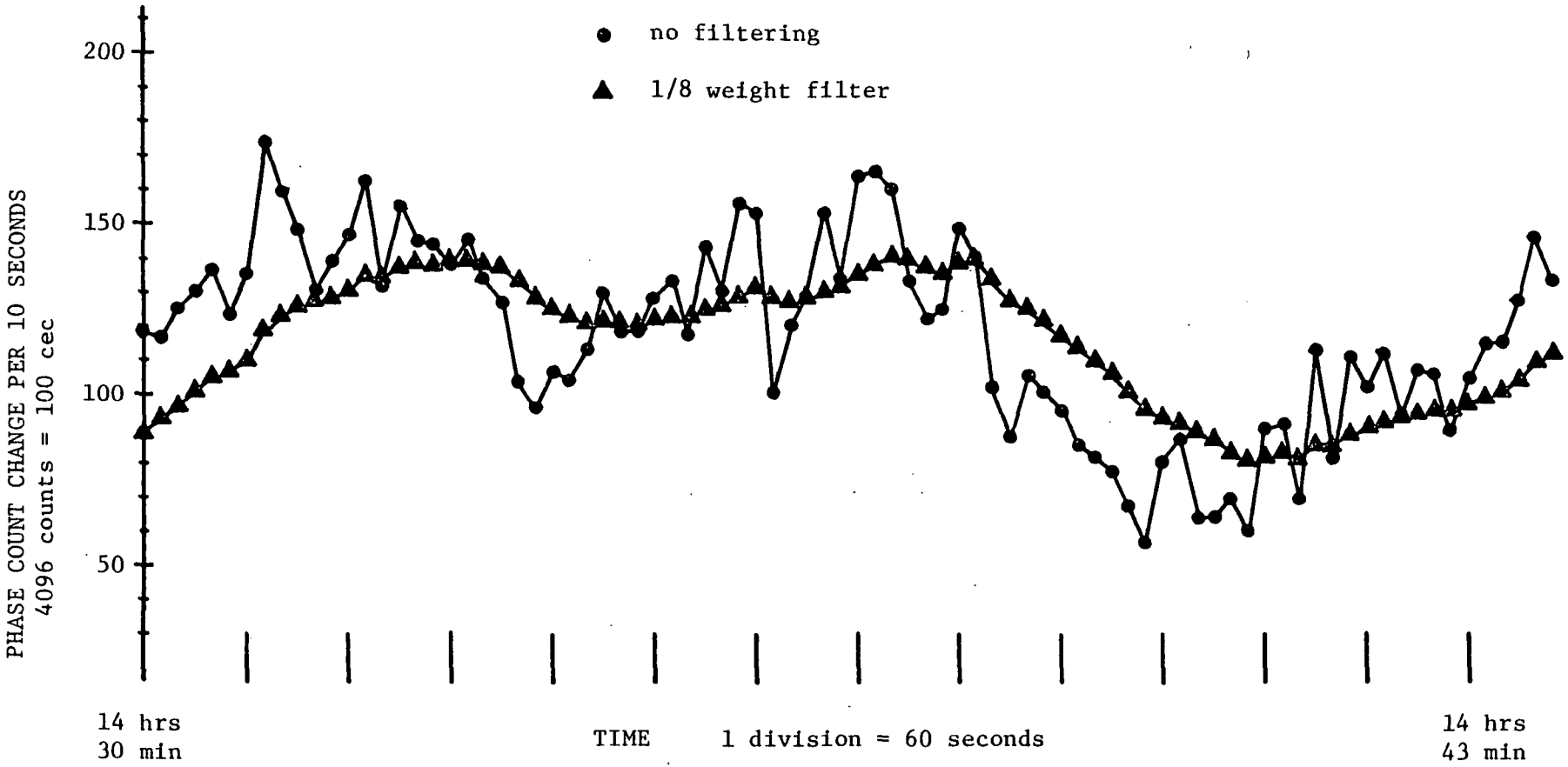


Figure 3-4. HAW-TRI Loop Integrator Difference Values With and Without Filtering (LEG 3, Flight ONR-12, 10.2 kHz)

The current track heading is

$$\theta_H(t) = \arctan\left\{\frac{v_E(t)}{v_N(t)}\right\}, \quad v_N(t) \geq v_E(t)$$

$$= 90 - \arctan\left\{\frac{v_N(t)}{v_E(t)}\right\}, \quad v_N(t) < v_E(t)$$

and the current estimate of velocity, VEL (knots) is

$$VEL(t) = \frac{v_N(t)}{\cos\theta_H(t)}, \quad v_N(t) \geq v_E(t)$$

$$= \frac{v_E(t)}{\sin\theta_H(t)}, \quad v_N(t) < v_E(t)$$

Using the difference between LOP phase measurements at current position and destination DTD is calculated at current time t and the estimated time of arrival (ETA) at destination is

$$ETA(t) = \frac{DTD(t)}{VEL(t)} \cdot 60 \quad \text{(min.)}$$

The output DEG to WP is the heading from current position to the destination waypoint and is calculated using (3-2) and (3-3) where $\Delta E_1(t)$ and $\Delta N_1(t)$ are used as a measure of distance to waypoint. Heading error is then the difference between current heading and heading to waypoint

$$\theta_e = \theta_H(t) - \theta_D(t)$$

Finally, cross-track deviation XTD is calculated using the difference between current heading to waypoint $\theta_D(t)$, original course heading to waypoint $\theta_D(0)$, and DTD:

$$XTD(t) = DTD(t) * \sin[\theta_D(0) - \theta_D(t)]$$

where $XTD(t) < 0$ if $\theta_D(t) > \theta_D(0)$ and $XTD(t) > 0$ if $\theta_D(t) < \theta_D(0)$.

3.2.2 Evaluation of navigation equations.— The algorithm development of the navigation outputs has been based on the assumption that the LOPs are straight lines within a local region and are parallel and uniformly spaced. These assumptions only approximate the actual situation. A simulation of the navigation algorithms was made to evaluate errors resulting from this approximation. A typical flight segment of approximately 66 n.mi. was set up between Hampton, Va., and Wallops Island, Va., with origin and destination defined in terms of the chart LOP intersections assuming OMEGA navigation with transmitters A-Norway, G-Trinidad, C-Hawaii, and LOPs AG, GC. A straight line path in Lambert projection x, y coordinates was defined between origin and destination. At uniformly spaced distance intervals along this path LOP values were calculated. These were assumed to be the result of measured phase values at each point. The situation with errorless phase measures was included to evaluate the effect of assuming LOPs to be straight lines. Other runs included phase measurement error.

In evaluating the navigation equations, heading error and cross-track deviation were calculated using the actual position as given by the Lambert x, y coordinates (a rectilinear system) and the position estimate as determined by the navigation equations which used relative LOP phase measurements. Figure 3-5 depicts the situation as represented in a rectilinear coordinate system. The actual path is illustrated as an actual straight line in the x, y coordinate system which represents the true path in this evaluation. The curved dotted line is an exaggerated representation of an OMEGA LOP coordinate straight line transformed into the rectilinear system. The heading error δ and cross track error XTD are geometric errors. In the simulation the receiver is initially at the origin x_0, y_0 and is moved

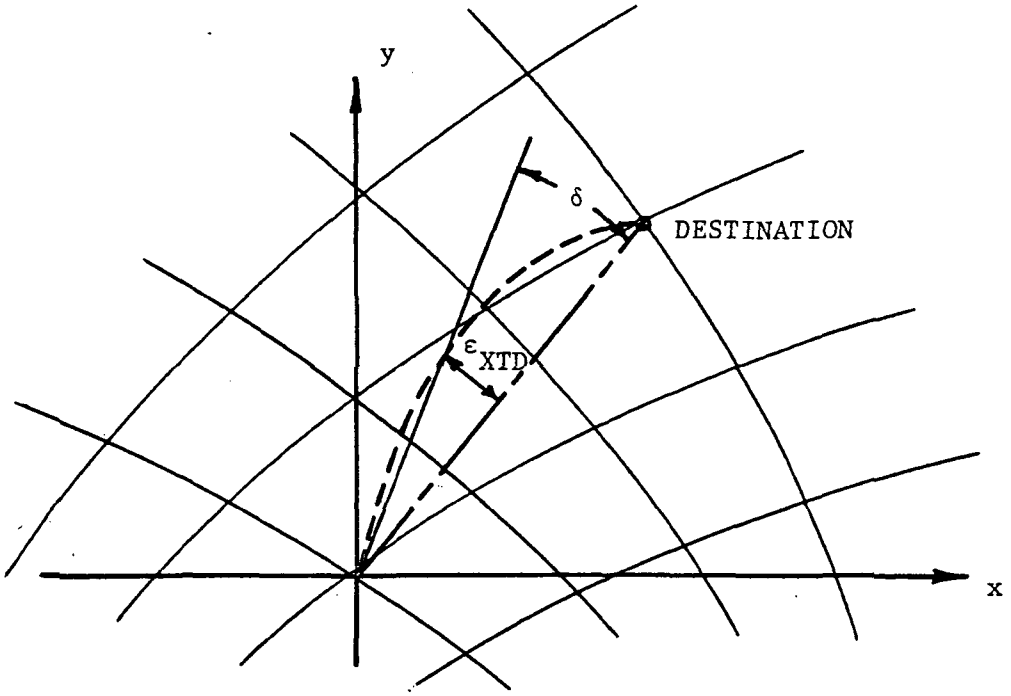


Figure 3-5. Geometry of OMEGA Straight Line in Rectilinear Coordinate System.

along a straight line according to $x_i = x_{i-1} + k_x$, $y_i = y_{i-1} + k_y$ where $\frac{k_x}{k_y}$ is the slope of the line between origin and destination. At each point x_i, y_i the LOP phase values of the two transmitter pairs are calculated. This involves converting Lambert x, y coordinates to latitude and longitude and then calculation of the LOPs at this latitude/longitude using the CHART routine (ref. 1). Using the navigation equations and either this calculated LOP position or this position plus error, an LOP difference is formed and transformed to an east, north position difference between the actual position and the estimated position. This displacement is transformed to a heading error and a cross-track error. Figure 3-6 illustrates the cross track error XTD. Heading error is simply the angle defined from the origin position between the heading along the actual course and the heading that would have resulted in position (x_i', y_i') on the "OMEGA straight line".

With the assumption of no phase measurement error Figure 3-7 illustrates heading error δ and cross-track deviation XTD on a flight between Hampton, Va., and Wallops Island, Va. This simulates operation at 10.2 kHz. As might be expected XTD is greatest near mid path and on this ~66 n.mi. path has a maximum value of approximately 230 m. Heading error is a maximum initially and remains less than 1° for this flight path. These results indicate that the straight lines assumption for flight paths of this magnitude with reasonable LOP crossing angle (~37° in this case) closely approaches the actual situation.

Figures 3-8 and 3-9 provide distributions of cross-track deviation and heading error on the same flight path with a normal phase measurement error of zero mean and 6 cec RMS. Figure 3-8 involves using LOPs NOR-TRI and TRI-HAW while Figure 3-9 is with LOPs TRI-HAW and HAW-NDK which have a crossing angle of ~78°. Heading error in both cases is generally within $\pm$ 3° and cross-track deviation is generally less than 1 km. These results reiterate that geometric error due to the assumption of OMEGA LOPs as evenly spaced straight lines should be relatively insignificant.

3.2.3 Latitude/longitude navigation outputs.— Navigation output from an OMEGA system can include position estimate information in coordinates convenient for the navigator. In this section two methods for converting an LOP intersection to a latitude/longitude position estimate are discussed.

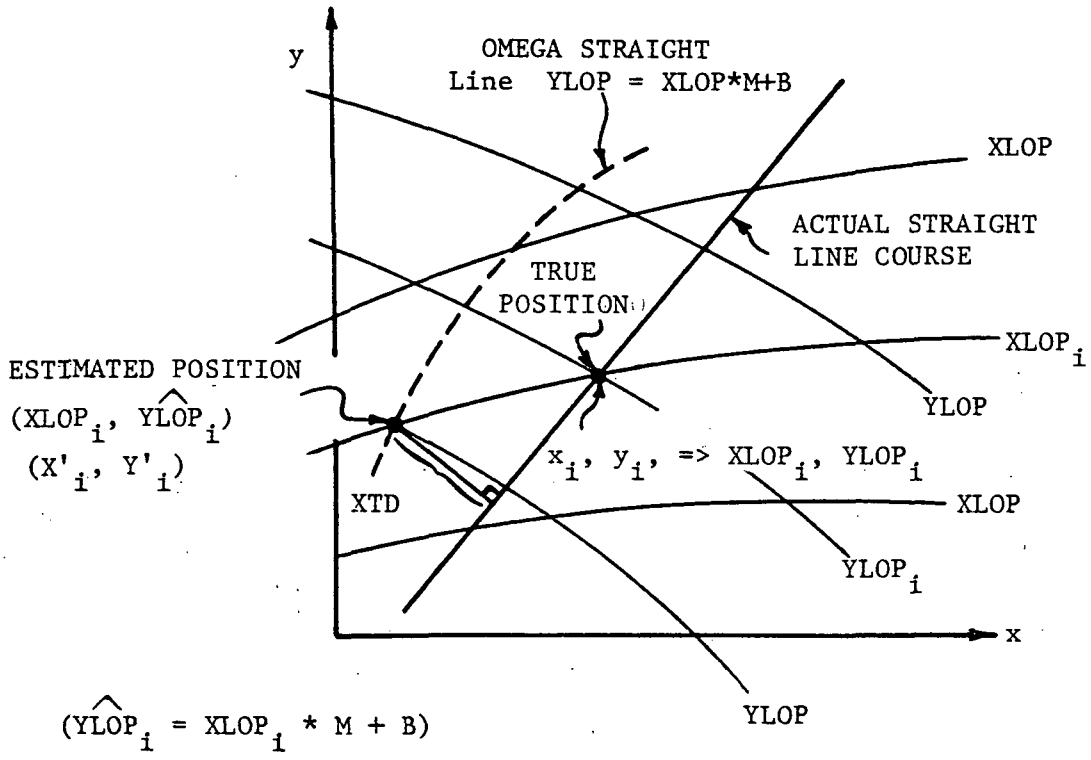


Figure 3-6. Cross-Track Deviation (XTD) Resulting From Assumption of LOPs Being Uniformly Spaced Straight Lines.

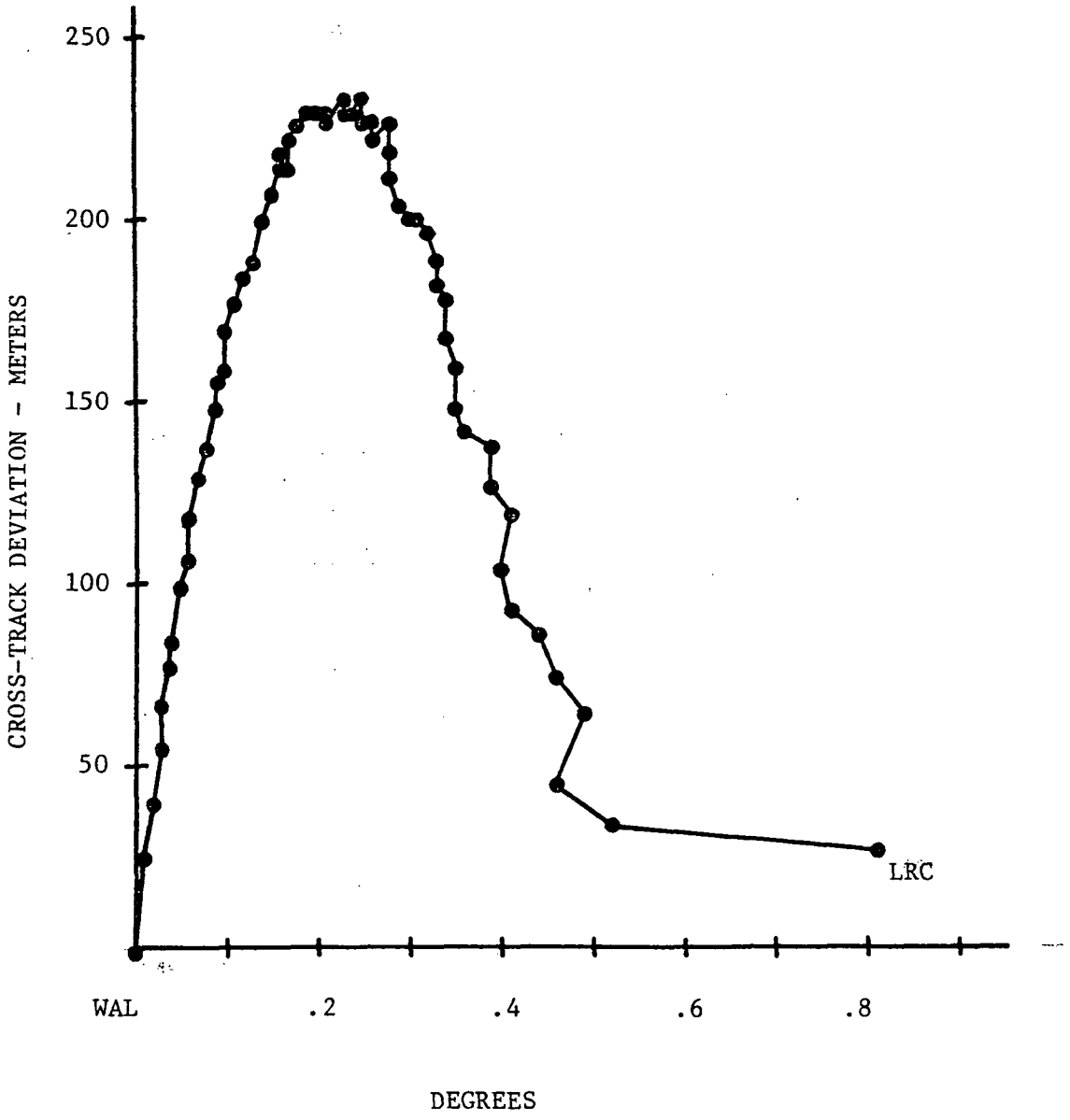
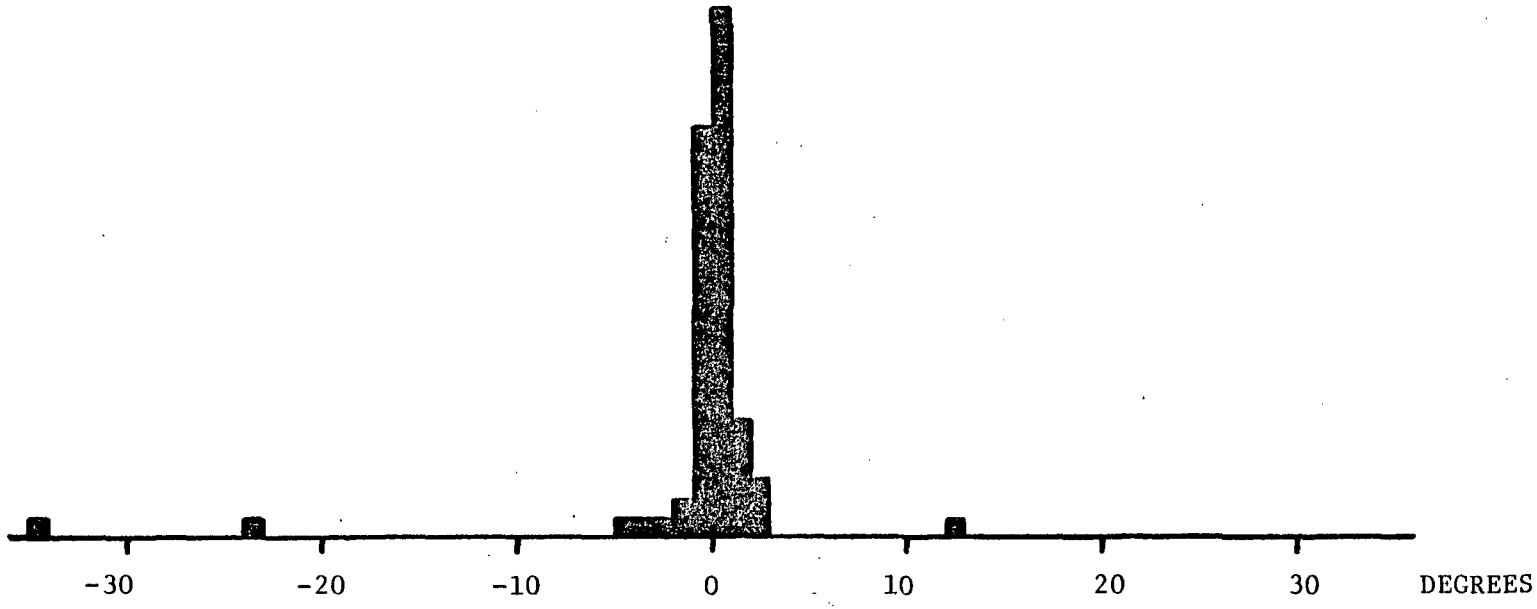
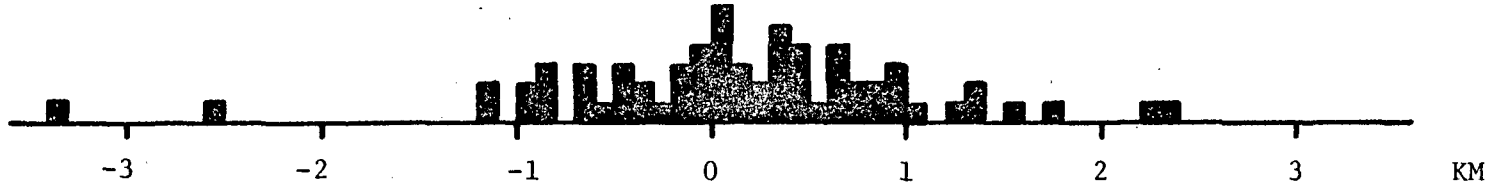


Figure 3-7. Cross-Track Deviation vs. Heading to Destination for Simulated Flight from Hampton to Wallops with no OMEGA Error. LOPs NOR-TRI and TRI-HAW at 10.2 kHz.

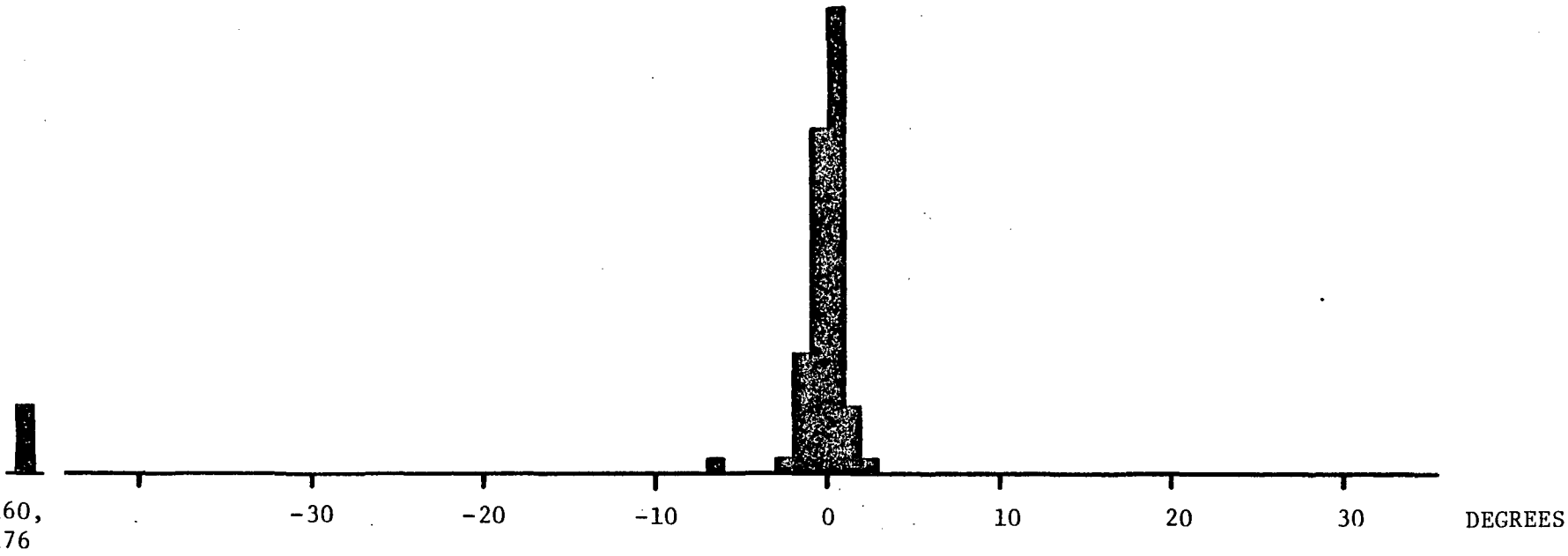


(a) Heading to Destination Error Distribution

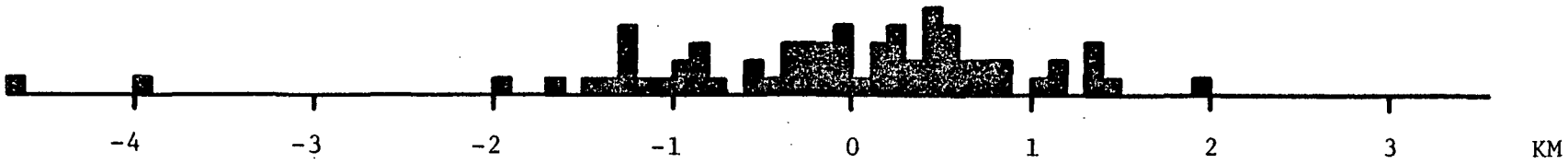


(b) Cross-Track Error Distribution

Figure 3-8. Simulated Flight from Hampton to Wallops Island with OMEGA RMS Error of 6 cec. LOPs NOR-TRI and TRI-HAW at 10.2 kHz with Fixes at 1 n.mi. Intervals.



(a) Heading to Destination Error Distribution



(b) Cross-Track Error Distribution

Figure 3-9. Simulated Flight from Hampton to Wallops as in Figure 3-8 with LOPs TRI-HAW and HAW-NDK.

The first method is tailored to the situation when two LOP estimates (an intersection) are available and an estimate of the latitude and longitude of this intersection is desired. The second method is attributable to Pierce (ref. 3) and is applicable when two or more LOP estimates are available and an estimate of the latitude and longitude is needed. Both methods are iterative and are terminated on the basis of a stopping rule which defines the magnitude of allowable convergence error. The first method is termed the "orthogonal step" method and is continued until the estimated position in latitude/longitude yields LOP values which are within some ε of the input LOP values. The Pierce method is compared on the basis of the same stopping rule, however, Pierce (ref. 3) has suggested other rules to test for convergence which are based on analyzing the change in the magnitude of the step size from one iteration to the next.

Fundamentally, the methods involve using some latitude/longitude point which is a "starting point", determining what the LOP values for the transmitter station pairs are at that point, estimating a step size in longitude and/or latitude to move toward the desired point, recalculating the LOP values at this new point, comparing these values with the measured LOP values,and,on the basis of whether the magnitude of the error is less than or equal to ε as defined in the stopping rule,repeating the process or terminating the process with an estimate of latitude/longitude. The orthogonal step method derives step size in latitude or longitude from the LOP difference between current point and desired ending point and the baseline LOP gradient. The first step is in latitude or longitude depending on which will move the estimate closer to the desired point. Each succeeding point alternates steps in latitude and longitude. The Pierce method is a direct step method in that step direction and step size are based on the geocentric angle between current position defined by phase distance to a set of known positions and desired position based on the LOP measurements. Results presented use a calculation step gain factor which is a function of the iteration number. Appendix C defines both of the algorithms in more detail.

In comparing the methods several typical situations were used. These involved defining some starting point in terms of latitude/longitude and

some ending point in terms of LOP_x/LOP_y, employing the method until convergence, and comparing methods on the basis of the number of steps to convergence. Several assumptions should be stated. In the Pierce method any starting point within the general area of the OMEGA transmitters being used for navigation is acceptable. With the orthogonal step method the starting point should be as close as possible to the desired point to provide for rapid convergence. The exact limitations have not been determined, however, this does not appear to be a significant factor when considering the application of the algorithm. Necessarily in most applications of OMEGA the navigator or navigation receiver will know the receiver location within a few miles. Thus a starting point close to the desired point will be known in virtually every situation. With both methods, either the LOP phase values used are assumed to be corrected to chart so that the chart phase velocity is applicable, or, a good estimate of actual phase velocity is necessary.

Table 3-4 provides a comparison between the two methods for several desired positions displaced from Hampton, Virginia, with Hampton as the starting point. In Table 3-4(a) LOPs NOR-HAW and TRI-HAW are considered. These two LOPs have a crossing angle of $\sim 60^{\circ}$. In Table 3-4(b) LOPs TRI-HAW and TRI-NDK with a crossing angle of $\sim 12^{\circ}$ are used. The stopping rule for each situation was defined in terms of a convergence in both LOPs at termination to be less than or equal to 1 cec at 10.2 kHz. The direction of the desired point from the starting point varied and distances ranged from about 10 n.mi. to 150 n.mi. Figure 3-10 provides a plot of the loci of iterations for the two methods with LOPs NOR-TRI and TRI-HAW. The orthogonal step method appears to be more robust from the standpoint of variation in speed of convergence with respect to LOP crossing angles. However, for good crossing angles the Pierce method provides definite time to convergence advantages. Table 3-4 also includes for the Pierce method the stopping parameter value associated with an alternate stopping rule which is necessary if more than two LOP measurements are available. At each iteration the RMS value of the step size geocentric angle relative to all transmitter paths is calculated and compared with some value which is no larger than the estimated propagation error (say 0.004 degrees). Note that this alternate stopping rule would have been

TABLE 3-4

Comparison of Orthogonal Step and Pierce Direct Step Methods for Lat./Long. Estimation of OMEGA LOP Fix

Note: Starting point for all examples is Hampton, Va., position LAT: 37.0985°N, Long: 76.3851°W ε = 1 cec.

a. LOP's NOR-HAW, TRI-HAW at 10.2 kHz. Crossing Angle $\approx 60°$

	METHOD I ORTHOGONAL STEP			METHOD II PIERCE DIRECT STEP			
DESIRED POSITION	EST POSITION	ERROR SECS OF ARC	# OF ITER	EST POSITION	ERROR SECS OF ARC	# OF ITER	RMS ΔT
36.3022°N (AHO)	36.3052	10.8	15	36.3031	3.2	6	—
77.0275°N	77.0281	2.2		77.0264	4.0		
35.9031°N (RTI)	35.9055	8.6	16	35.9037	2.2	7	.0039
78.8666°W	78.8677	4.0		78.8659	2.5		
37.9282°N (WAL)	37.9272	3.6	15	37.9276	2.2	6	.0059
75.4760°W	75.4783	8.3		75.4770	3.6		
"				37.9281	0.36	7	.0012
				75.4761	0.36		
37.0934°N	37.0928	2.2	12	37.0926	2.9	5	—
75.9715°W (FIS)	75.9731	5.8		75.9728	4.7		
37.1813°N (PGO)	37.1839	9.4	12	37.1816	1.1	6	.0027
77.2133°W	77.2134	0.4		77.2128	1.8		

TABLE 3-4 (CONT'D)

b. LOP's TRI-HAW, TRI-NDK at 10.2 kHz. Crossing Angle ≈ 12°

DESIRED POSITION	METHOD I ORTHOGONAL STEP EST POSITION	ERROR SECS OF ARC	# OF ITER	METHOD II PIERCE DIRECT STEP EST POSITION	ERROR SECS OF ARC	# OF ITER	RMS ΔT
36.3022°N$_{(AHO)}$	36.318	56.88	21	36.3190*	60.48	> 50	—
77.0275°W	77.013	52.20		77.0079	70.56		
37.9282°N$_{(WAL)}$	37.906	79.92	28	37.8643*	230.04	> 50	—
75.4760°W	75.499	82.8		75.5520	273.6		
37.0934°N$_{(FIS)}$	37.078	55.44	22	37.0850	30.24	50	.0033
75.9715°W	75.988	59.4		75.9807	33.12		
37.1813°N$_{(PGO)}$	37.198	60.12	9	37.1926*	40.63	> 50	.00474
77.2133°W	77.197	58.68		77.2029	37.44		

*No convergence at 50 iterations. Values shown in Table are those resulting at the 50th iteration.

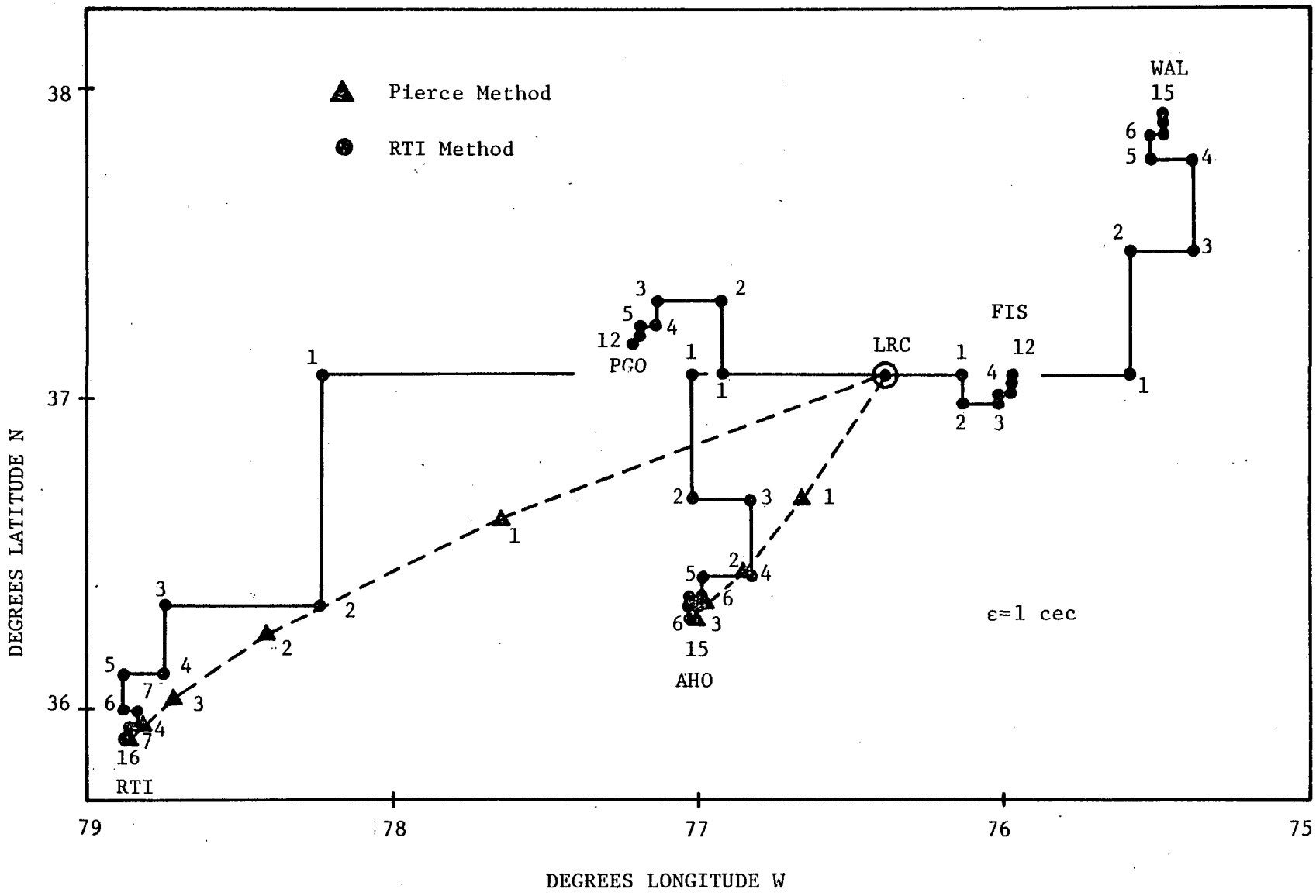


Figure 3-10. Loci of Iterations from Hampton in 3 Station Network Using TRI-HAW and NOR-NDK LOPs at 10.2 kHz.

satisfied in all situations which involved 50 or fewer iterations except one with the stopping parameter set at .004°. In this one case an extra iteration yielded convergence according to this rule.

3.3 Algorithms for Navigation Equation Implementation

This section summarizes the algorithm development relative to implementation of the previously described navigation equations for OMEGA navigation. The equations are implemented in software for the INTEL 4004 microcomputer which is a part of a feasibility model of an aircraft OMEGA navigation receiver. The discussion which follows defines the input/output variables used by the navigator/operator, the algorithms used in the development of navigation outputs, the numerical precision of the various calculations used in arriving at these output quantities, and the associated arithmetic and trigonometric functions implemented in the software.

3.3.1 Navigation input/output variables.— As discussed in Section 3.2, with the microprocessor based OMEGA navigation receiver the navigator can operate in either the carrier mode (10.2 kHz) or the difference frequency mode (3.4 kHz = 13.6-10.2 kHz). The receiver actually calculates the necessary position estimate information for both modes and only uses the estimate based on the operator selected mode for the navigation output calculations. Thus lane counting is continuous in both modes and allows the operator to change the mode at any time without interrupting the receiver operation. Furthermore, all phase unit type inputs and outputs are in 10.2 kHz units for operator convenience, ambiguity resolution, and to facilitate software implementation of the navigation equations for the two modes. Because of this characteristic, an LOP numbering convention is assumed for use of the receiver. The LOP charts are assumed to follow the conventional 10.2 kHz numbering scheme. Terms of reference are as follows. Three 10.2 kHz lanes correspond to one 3.4 kHz lane so that in referring to the LOP numbers, each "whole lane" will mean three 10.2 kHz lanes corresponding to one 3.4 kHz lane. Thus whole lane numbers will be modulo three integers. Associated with the whole lane values will be an "inner-lane" value which is either 0, 1, or 2 depending on which 10.2 kHz lane is referred to

within a "whole lane." Therefore, the actual 10.2 kHz lane number of any inner-lane is simply the sum of the "whole lane" value and the "inner-lane" value. For example, whole lane values might be 900, 903, 906, etc. corresponding to 3.4 kHz lanes 300, 301, 302, etc. The corresponding 10.2 kHz lanes in the first whole lane (900) would be 900, 901, 902 depending on whether the inner lane reference is to 0, 1, or 2 respectively. Charts employed with the receiver would have every third 10.2 kHz lane as a bold line with the appropriate "whole lane" number. Inner lanes would be lighter shaded lines with numbers 1 or 2 with the first "inner-lane" in each whole lane understood to be "0." Figure 3-11 illustrates a "one LOP" chart with this numbering convention.

OMEGA propagation prediction corrections are employed according to a "modified-differential" concept in this initial implementation. The operator actually inputs a 3.4 kHz and a 10.2 kHz correction when receiver operation is initialized. These corrections (in 10.2 kHz units) are signed phase values (in the range ±150 for 3.4 kHz mode and ±50 for 10.2 kHz mode) which can be determined using dead reckoning or using some published correction. Once entered, the values are continuously used in forming position estimates until changed by the navigator/operator. A new value may be entered at any time and will immediately replace the old value. All position estimate outputs are based on corrected phase measurements except when no correction has been input (necessarily zero) which is the situation when the receiver is initially powered up.

The receiver is capable of storing two navigation waypoints identified as waypoint #1 (WP1) and waypoint #2 (WP2) in addition to present position in the form of a two-LOP intersection. Each intersection LOP value is input in terms of a lane value (whole lane) in integer form and an inner-lane/fractional lane value in centicycles (cec of 10.2 kHz in the range 0,299). The operator designates the two LOP's through separate keyboard entries which become designated as LOP_x and LOP_y. Subsequent current position and waypoint identities are then understood to be in terms of an LOP_x and an LOP_y value.

Another input required for navigation is the lane width (10.2 kHz lane) conversion factor for each of the LOP_x and LOP_y lanes. This is the n.mi per 100 cec value which would normally be provided on a navigation chart.

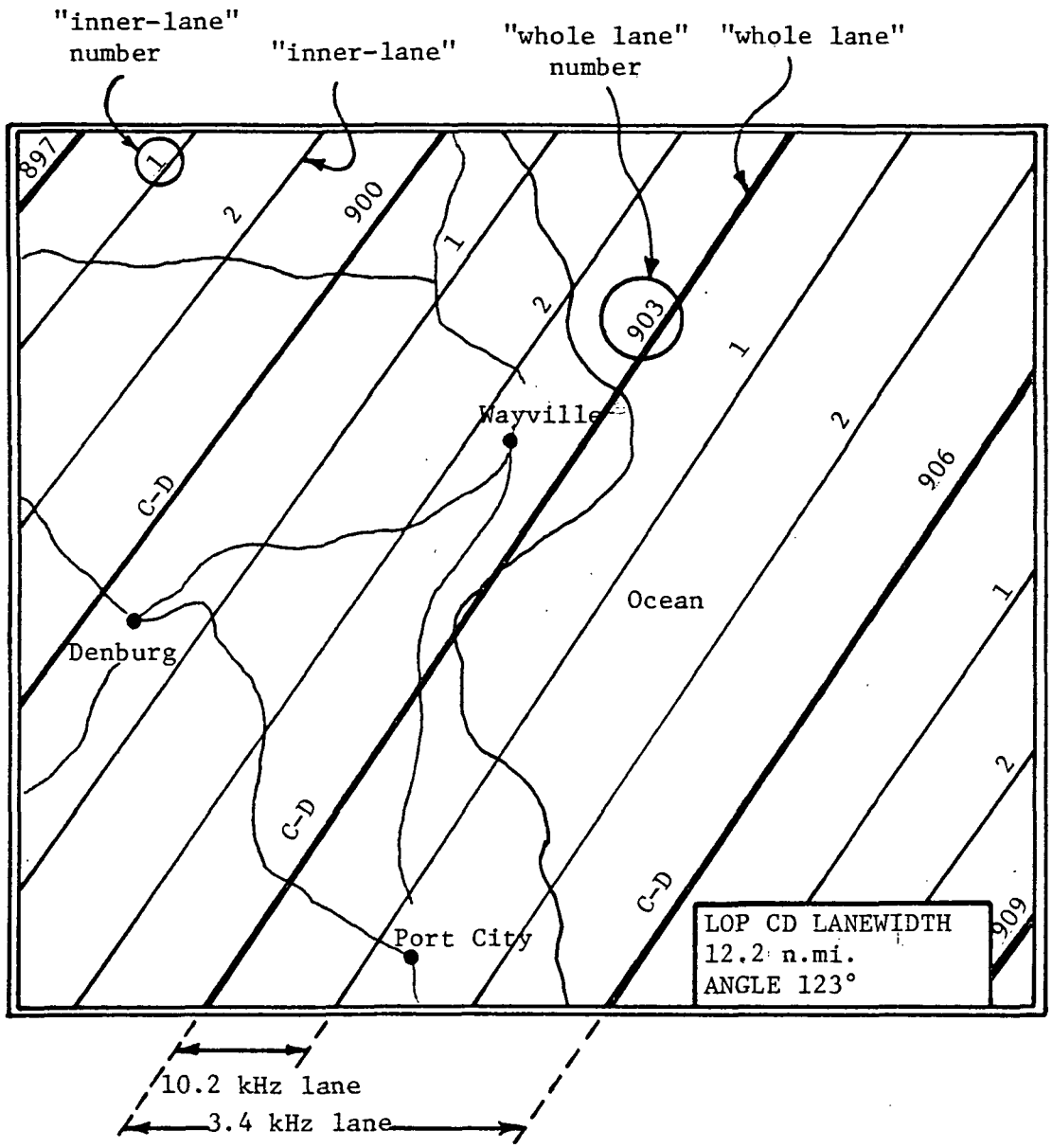


Figure 3-11. OMEGA Chart To Illustrate LOP Numbering Convention Used With Airborne Receiver

In Figure 3-11 a note at the bottom right-hand corner provides this information for the LOP illustrated. This is a nominal value for the entire area of the chart. Associated with the lane width is a lane angle value in degrees measured from grid north to the LOP gradient direction. This value is a nominal value for the chart assuming that the LOP lines are straight lines with a slope which is accurate at the center of the chart. A value is illustrated in the lower right-hand corner of Figure 3-11.

The complete set of navigation type inputs which are used with the receiver is detailed in Table 3-5. Included with each table entry is the keyboard label related variable definition, the number of digits used for the variable, the numerical precision, and the assumed range limits of the variable. The inputs are all decimal numbers and are converted to binary and stored as binary integers by the keyboard interpreter routine as if the numbers were all decimal integers. The decimal point that is associated with some of the variables is accounted for within the navigation equations software as these values are used. At no time are the original input values in the storage area changed by the software so that any input variable can be accessed from the keyboard and will appear on the display as it was entered. It should be noted that internal precision within the navigation software is limited to a maximum of twelve bits so that occasions arise when the variable value which is input cannot be represented precisely within the navigation software. This is basically just a statement of limitations inherent in decimal to binary conversion when fractional decimal numbers are involved and the binary representation is bit limited. The precision of navigation equation calculations will be addressed in a subsection to follow.

Navigation output values consist of those inputs from the navigator which can be examined and will read out exactly as input, as well as, calculated output values. Table 3-6 lists the calculated output variables, the selector switch or keyboard label related to the variable, the number of digits displayed with the implied decimal point, and the range limits over which the value is valid. Outputs are all decimal numbers. All calculated values are stored in the receiver as binary mixed numbers so that some of the precision loss is incurred upon binary to decimal conversion associated with the display routine itself. This is just a result of insufficient significance in the three-digit display.

Table 3-5. Input Variables To OMEGA Navigation Receiver

Input Value	Keyboard Designator	Units	Variable Form	Assumed Range	
				Decimal Input	Internal Hexadecimal
LOP X, LOP Y Station Pairs Designation	SX or SY	N/A	2-digit*	12, 87	N/A
Angle of LOP X, LOP Y Gradient from Grid North (CW+)	ϕX or ϕY	Degrees	3-digit integer	0, 359 principle value of $\lvert\phi_x - \phi_y\rvert \geq 41°$	0, 167
LOP X, LOP Y Lane Width	ΔX or ΔY	tenths of n.mi.	3-digit integer	0.0, 200	0, 0C8
Current Position Lane	P_oX, P_oY	10.2 kHz lanes	3-digit integer modulo 3	0, 999	0, 3E7
Current Position Fractional Lane	p_ox, p_oy	10.2 kHz centilanes	3-digit integer	0, 299	0, 12B
Waypoint Lane	P_1X, P_1Y P_2X, P_2Y	10.2 kHz lanes	3-digit integer modulo 3	0, 999	0, 3E7
Waypoint Fractional Lane	p_1x, p_1y p_2x, p_2y	10.2 kHz centilanes	3-digit integer	0, 299	0, 12B
10.2 kHz Mode Correction	ΔX, ΔY in 10.2 mode	10.2 kHz centilanes	2-digit signed integer	-50, +50	FCE, 032
3.4 kHz Mode Correction	ΔX, ΔY in 3.4 mode	10.2 kHz centilanes	3-digit signed integer	-150, +150	F6A, 096

*The station pair inputs SX, SY are each two digit numbers to define the LOP X, LOP Y OMEGA transmitter pairs. The first digit (10's place) is the first station, and the second digit (1's place) is the second station in a designated pair. The convention A=1, B=2, C=3, D=4, E=5, F=6, G=7, H=8 is followed so that if SX is 25, then LOP X is the BE LOP, etc.

Table 3-6. Output Variables From OMEGA Navigation Receiver

Output Variable	Keyboard or Selector SW Designator	Units	Variable Form	Valid Range of Output	
				Internal Hexadecimal	Decimal Output
Current Position Lane	P_oX, P_oY	10.2 kHz lanes	3-digit integer	0, 7FF	0, 999
Current Position Fractional Lane	p_ox, p_oy	10.2 kHz centilanes	3-digit 2-decimal places	0, 12A	0, 298
Distance to Destination	N.MI. (to WP)	n.mi.	3-digit integer	0, 3FF	0, 999
Angle from Current Position to Active Waypoint Referenced to Grid North	DEG (to WP)	degrees	3-digit integer	0, 167	0, 359
Cross-track Deviation	N.MI. (error)	tenths of n.mi.	3-digit 1-decimal place	0, 7F8	0, 999
Heading Error Between Current Hdg and Course Hdg to WP	DEG (error)	degrees	3-digit integer	0, 167	0, 359
Estimated Time of Arrival	MIN (error)	minutes	3-digit integer	0, 3FF	0, 999
Ground Track Velocity	KNTS (track)	n.mi./hr. knots	3-digit integer	+3C, 3FF	60, 999
Ground Track Velocity Vector Referenced to Current Bearing	DEG (track)	degrees	3-digit integer	0, 167	0, 359

Notes: +To insure that ETA is valid, velocity must be at least 60 knots.
Velocity of 15.9927895 n.mi./min. is actual maximum within calculations.

* Exact within limitations of OMEGA receiver. No computation error will occur..

3.3.2 Navigation Calculations.— In the course of arriving at the various navigation output values that are to be displayed, a number of intermediate calculations are necessary. These calculations involve use of entries from the keyboard as well as phase measurements within the receiver. A set of arithmetic and trigonometric subroutines are also involved in these calculations and will be discussed in more detail in a later subsection.

In the process of numerical calculations, a modified floating point arithmetic is employed. All numbers are represented in a floating point form but no normalization is used. This means that the actual binary point location for numbers varies, and the computer code is detailed according to the location of the binary point, i.e., each calculation must be based on the location of the binary points of the operands. In all arithmetic operations, there are never more than twelve bits of significant. For some computations, this just does not provide enough significance so that numbers must be treated in two parts and bit shifting operations performed to allow for meaningful ranges in the numerical results. As each calculation is discussed, the necessary operations to retain significance will be defined. The resulting limitations on the ranges of particular variables which are imposed will also be stated.

The navigation section of the microprocessor code is entered during each OMEGA frame (10 sec intervals) to update the various navigation output values. Initial entry is set up with an origin to destination keystroke sequence once all the navigation inputs have been entered. An initialize flag is simultaneously set with a course designation sequence entry which is used to provide a branch within the code to set up the initial desired course heading between origin and destination.

The first step in the calculation is to read the appropriate OMEGA transmitter 10.2 kHz phase values which are eight bit representations of the most recent phase-locked-loop phase corresponding to each of the operator selected stations. According to the selected LOP's chosen for navigation, the LOPX and LOPY values are calculated and stored, each with eight bits of significance. These values can be interpreted as the binary representation of the LOP phase in fractional lanes at 10.2 kHz if the

binary point is assumed to be to the left of the most significant bit of the eight bit quantity. Thus an LOP phase value is in the range 0.0 to 0.99609375 lanes (0.0, 99.609375 cec @ 10.2 kHz) with a resolution of .00390625 lanes (.390625 cec). The LOPX, LOPY values are each in turn corrected with the 10.2 kHz correction and used with the last calculated values (last entry into the navigation section of code) to form DLOPX, DLOPY which are the 10 second changes in LOP phase. These changes are dependent upon both phase measurement error (propagation error and receiver clock drift) and any velocity attributable to the receiver or vehicle containing the receiver. Thus

$$\text{DLOPX}(i) = \text{LOPX}(i) - \text{LOPX}(i-1) \quad @\ 10.2\ \text{kHz}$$
$$\text{DLOPY}(i) = \text{LOPY}(i) - \text{LOPY}(i-1) \quad @\ 10.2\ \text{kHz}$$

where i refers to the current 10 second frame and (i-1) refers to the last 10 second frame. The DLOPX, DLOPY values are twelve bit numbers formed from two eight bit numbers which are each right justified in a twelve bit word. For the calculation LOPX and LOPY may be interpreted as scaled fractional lane values with the binary point immediately to the right of the most significant bit of the twelve bit word (standard form) and are necessarily positive since the most significant four bit byte is zero. Let ϕ_x be the mantissa of the LOPX value. Then, with the twelve bit representation of these numbers

$$\delta_x(i) \times 2^{-3} = [\phi_x(i) * 2^{-3}] - [\phi_x(i-1) * 2^{-3}]$$

where it is assumed that the LOP values are corrected using the correction value input by the operator. Here $\delta_x(i)$ is the mantissa of the DLOPX value.

The DLOPX, DLOPY values are assumed to have at most six significant bits restricting the 10 second change in LOP phase to approximately 24.6 cec. With a minimum lane width at 10.2 kHz ($\sim$8 n.mi.), this corresponds to a velocity of 11.8 n.mi./min. which is considered to be adequate. At this point in the code, the DLOPX, DLOPY values are shifted left 1 bit and stored in the most significant two bytes of 3 byte words in processor RAM to be used for lane counting.

In the sequence of calculating DLOPX, DLOPY values, the 10.2 LOPX, LOPY as well as the 3.4 LOPX, LOPY values are calculated in corrected form. In calculating the corrected values, the 10.2 kHz X, Y corrections input in 10.2 kHz phase units (DOX, DOY) are input in the range ±50 cec. These are scaled to a 0,99 range by adding 100 if negative and then dividing by 100×2^{-8} to get into the proper fractional representation. Thus

$$\text{LOPX CORRECTION} = X_{CORR}(10.2) \text{ lanes} * 2^{-3}$$

$$\text{LOPY CORRECTION} = Y_{CORR}(10.2) \text{ lanes} * 2^{-3}$$

and are normalized properly to be added directly to the LOPX, LOPY values calculated from the receiver output. Some loss of precision is possible since the divide operation does in effect round-off the correction to a value represented in eight bits which is not greater than the value input. For example, if a correction of $-15_{(10)}$ is input from the keyboard, this is converted to $FF1_{(H)}$ by the keyboard decimal to binary routine and is scaled by +100 (since negative) to yield $055_{(H)}$. Dividing by 100×2^{-8} yields a lane correction of .84765625 as compared to the desired value of .85 (error = -.00234375 lanes). Maximum error is -.00375 lanes. The 3.4 kHz corrections are input in units of 10.2 kHz in the range ±150 cec. These are scaled to a 0,299 range by adding 300 if negative and then dividing by 300×2^{-9} and shifting the result 1 bit to the right to get the fractional lane value in units of 3.4 kHz. Again, numerical error due to round-off can occur with a maximum error of -.00375 lanes of 3.4 kHz (-.01125 lanes of 10.2).

The mode switch position is checked; and depending on the mode the current position RAM registers, P_oX, p_ox, P_oY, and p_oy are updated in 10.2 kHz units. This update includes a lane counting algorithm so that lane counting is actually accomplished in the selected mode. Basically the same procedure is used regardless of mode, but the implementation is slightly different. In the 10.2 kHz mode the changes in LOPX,LOPY values are determined. If the fractional lane change is small and the most significant bit of the 8 bit fractional lane values are different, which is characteristic of a lane crossing with the phase count representation used, a lane crossing is recorded. If the change is negative to positive,* the inner-lane register (1-four bit byte) is incremented. If a positive to negative change is

*Here the most significant bit of the 8-bit fractional lane value is interpreted as a sign bit. If 0 then positive; if 1 then negative.

observed, the inner lane register is decremented. Before exiting, the inner lane register is checked. Since it must be 0, 1, or 2, if it has been incremented to 3, it must be cleared and the lane register (P_oX or P_oY) incremented. If the inner-lane register was decremented to -1, then the lane register is decremented by 3 and 3 is added to the inner lane register. Thus lane crossings are accounted for in the 10.2 kHz mode. In the 3.4 kHz mode. the LOP phase change is determined in 3.4 kHz units. A sign change is used to increment or decrement the lane register by 3 directly. Then the inner-lane/fractional lane twelve-bit current position register is replaced with the new 3.4 kHz LOP phase measurement after conversion to 10.2 kHz units which automatically sets the inner lane register properly. The procedure uses separate register pairs to contain the current position information corresponding to both modes of operation so that a mode change at any time will not interfere with continuity of operation.

The current position information is scaled for keyboard/display before final storage in RAM memory. It is necessary to convert the inner-lane/fractional lane count register to an integer number of centicycles of 10.2 kHz for proper interface with the display software. This is accomplished through multiplication by 100 represented as $100 * 2^{-7}$. Thus the calculated value of phase is $\phi_{LOP} * 2^{-3}$. Multiplying by $100 * 2^{-7}$ and shifting right one bit yields the integer number of centicycles.[+] A one cec round-off error is possible in this conversion.

Once lane counting and updating of the current position registers has been completed, a sequence of pre-flight computations is entered. Here the input LOP gradient angles Φ_x and Φ_y are differenced to form an angle used to define the LOP to grid coordinate transformation (3-1). Limitations on the ranges of the numbers are again imposed because of the twelve bit method of representing numbers. Since k_x, k_y transformation constants have at most 5 bits to the left of the binary point and the assumed maximum on 10.2 kHz lane widths is 20 n.mi. then the difference $|\Phi_x - \Phi_y|$ must be no less than about 41°. This should not be restrictive since LOP's for navigation should have as close to a 90° crossing angle as possible based on geometrical accuracy considerations.

In calculating k_x and k_y, the input lane widths DELX' and DELY' are stored as integers ten times the actual value since the keyboard decimal to binary conversion assumes the decimal number is an integer. Thus DELX'=10*DELX and DELY' =10*DELY are the number of tenths of a mile per lane in error by $\pm$.05 n.mi.

[+]If multiplication by $100 * 2^{-8}$ is used, no bit shift in the product is required; however, additional round-off error would occur.

To get k_x and k_y, the values DELX', DELY' are divided by 10 times the sine of the difference in the LOP gradient angles ($\Phi_x - \Phi_y$). The error in this difference angle is approximately $\pm 1°$ (round-off error). The sine representation has a resolution of $\sim .0005$ so that 10 times the sine will have a resolution of $\sim .008$. The resulting values for k_x and k_y have a resolution of $\sim .06$ n.mi. with only four bits to the right of the binary point. Considering the precision of the input values, the precision of k_x, k_y is at the very best on the order of $\pm .1$ n.mi./lane. To get the elements of the transformation matrix k_{xn}, k_{yn}, k_{xe}, k_{ye} the values k_x, k_y are multiplied by sine or cosine of either Φ_x or Φ_y. The resulting values for k_{xn}, k_{yn}, k_{xe}, k_{ye} have a precision on the order of $\pm .2$ n.mi./lane since k_x, k_y are shifted left two bits before multiplication. This shift is possible because k_x, k_y are limited in magnitude to 5 bits to the left of the binary point which is immediately to the left of the least significant byte of the twelve bit word. All multiplies and divides are fractional operations based on assuming the binary point is shifted to the right of the most significant bit and input values are treated as unsigned.

Following through the calculations

$$DELX'' = DELX' * 2^{-11}$$

and

$$(\sin \Delta\phi)' = 10.0 \sin\Delta\phi * 2^{-4}$$

$$\text{where } (10.0 \times 2^{-4}) * \sin\Delta\phi = (\sin \Delta\phi)'$$

Thus

$$k_x' = \frac{DELX''}{(\sin \Delta\phi)'} = \frac{10.0\ DELX * 2^{-11}}{10.0 \sin \Delta\phi * 2^{-4}} = \frac{DELX}{\sin \Delta\phi} * 2^{-7}$$

$$\text{or } k_x' = k_x * 2^{-7}$$

This means k_x is of the form $\pm$00X XXXX.XXXX where the "." represents the location of the actual binary point. Following through, k_x is shifted left two places to move the most significant possible bit to the right of the sign bit. Then

$$k_x'' = k_x * 2^{-5} \quad .$$

To form, for example, k_{xn}

$$k_{xn}' = k_x'' * (-\sin \phi_y) = -k_x \sin \phi_y * 2^{-5} = k_{xn} * 2^{-5}$$

so that k_{xn} is of the form $\pm$XXXXX.XXXXXX with a maximum value of 31.984375.

It can be noted that multiple precision operations and storage could be used for the transformation elements to improve the numerical precision in the calculations. The additional code requirements are not considered to be acceptable at the present time. Also it was stated that these constants are recalculated each time the navigation equations are entered. In the present form, the flexibility of responding immediately to a change in a lane width value or lane gradient angle value is provided for.

Upon completion of the "pre-flight" computations, the grid x,y distance from the current position to the designated destination waypoint is computed. Using the contents of the active destination waypoint registers P_iX, P_iY, p_ix, p_iy (i=1 or 2) and the current position registers P_oX, P_oY, p_ox, p_oy, the phase differences between points and the transformation elements kxn, kyn, kxe, kye are used to calculate a ΔX and ΔY (grid coordinates), and a bearing to destination.

In calculating the LOP_x, LOP_y phase differences between current position and destination, two steps are required within the constraints of the twelve bit word size. First the full lane difference is calculated using the contents of the P_iX, P_iY registers. The magnitude of this difference is assumed less than or equal to $31_{(10)}$, thus five bits of significance, which restricts the displacements to a maximum of 248 n.mi. with minimum lane width of 8 n.mi. In the second step the difference in the inner lane/fractional lane count registers is calculated. The inner lane value is then algebraically combined with the full lane difference value and the resulting difference is shifted 6 bits left. The fractional lane count difference is shifted 2 bits right. The two differences are concatenated yielding a difference magnitude (11 bits of significance) of up to 31.984375 lanes with a resolution of $\pm$.0078125 lanes. Maximum error due to truncation is .0117 lanes. The LOP differences expressed as lane difference * 2^{-5} in the 12-bit floating format are transformed using the

kxn, kxe, kyn, kye values previously calculated and stored as n.mi./lane * 2^{-5} to yield displacements in n.mi. * 2^{-10} in the grid north and east directions. The multiplication of k_{ij} and an LOP difference can have at most a 4 n.mi. round-off error. Truncation of the fractional part of the lane counts can cause at most a 0.37 n.mi. error. With the implicit error contained in the transformation k values, the resulting displacement in n.mi. can be in error by 6-7 n.mi. in a distance on the order of 1000 n.mi. The percentage error can be much greater. Consider a phase difference of 0.1 lane with k value of 31 n.mi./lane. The correct value of distance is 3.1 n.mi. The calculated value is $+00000.000110_{(2)}$ multiplied by $+11111.000000_{(2)}$ yielding a value of 2.0 n.mi. when using the floating multiply routine. If the k value is 8 n.mi./lane,a value of 0.5 n.mi. results,instead of an actual value of 0.8 n.mi.

The north and east displacement between current position and destination waypoint is used with an arctangent function to compute bearing to destination. The north and east displacements (ΔN and ΔE) are each shifted left two bits before any subsequent calculations. This assumes a maximum value of 255.5 n.mi. displacement and will provide additional significance in calculations that follow. Upon initial entry into the navigation section of the code, the bearing to destination represents the bearing from point of origination (contents of P_o, p_o registers) to destination waypoint. Thereafter this value represents the course bearing from current position (contents of P_o, p_o registers) to the active destination waypoint until a new course is set up. A new course designation creates a new origination bearing. The arctangent function yields an angle with an error less than 0.7° so that the precision on the bearing calculation is primarily dependent on the accuracy of the north and east displacements.

The distance to destination (DTD) is calculated using either the north or east displacement with the current bearing angle to destination. In order to preserve as much accuracy as possible, the current bearing angle is checked. If it has a primary value of less than 45°, then cosine is used with the ΔN displacement. For a primary bearing of greater than 45°, the ΔE displacement is used with the sine. This allows for the sine/cosine divisor to be as large as possible and provides for using the larger of ΔN or ΔE for the calculation. The DTD value should be within 10 n.mi. of the true value.

Using the original course bearing and the calculated bearing from current position, a bearing error is used to determine course cross-track deviation, XTD. The product of DTD and the sine of the course bearing error yields XTD with very nearly the same precision as DTD.

The cross-track deviation is a signed number which is stored as the number of tenths of n.mi. off the desired course. A positive sign indicates displacement to the "left" such that a right turn would be needed to bring the aircraft back on course. A negative sign indicates a displacement in the opposite sense. Multiplying DTD * 2^{-8} by the sine of the difference between BCUR and BORG yields XTD * 2^{-8}. This is shifted left one bit and multiplied by 10 * 2^{-4} to yield [10 * XTD] * 2^{-11} which is the number of tenths of a n.mi. ready for display.

The velocity vector is calculated using the phase-locked loop integrator values which are stored in RAM. The second order phase-locked loops associated with the receiver accumulate the discriminator phase error output for each loop with updates every ten seconds. The accumulated counts (4096 = 1 lane) represents an instantaneous measure of $\frac{d\phi}{dt}$ for each OMEGA station received phase. The integrator value has large sample-to-sample variations correspnding to noise in the received phase. To provide a smoothed LOP velocity estimate the integrator values are differenced and filtered using the algorithm described in Section 3.1. This required one 12 bit RAM register for each smoothed integrator value which is the average fractional lane change in ten seconds. This assumes that the velocity is less than 0.24976 lanes in 10 seconds. The LOP count difference then has at most 10 significant bits. The differences are scaled left one bit to preserve significance in subsequent calculations This value can then be interpreted as 4 times the fractional lane change in 10 seconds with the binary point to the right of the MSB. A resoluation of 0.703 kts is possible with 8 n.mi. lanes (1.7578 kts with 20 n.mi. lanes). Using the difference between loop integrator values should cancel any phase velocity component due to frequency offset or phase drift within the receiver.

These signed DLOPX' and DLOPY' values obtained from the 10.2 kHz loop integrators are then used with k_{xy}, k_{xn}, k_{xe}, and k_{ye} transformations to get north and east components of the velocity vector. With the arctangent function, an estimate of current heading is calculated and referenced to the current

bearing to destination to be output in degrees as track error. The magnitude of the velocity vector is calculated similarly to DTD, in that the larger of VEL(EAST) and VEL(NORTH) component is used with the current heading angle to determine velocity in n.mi. per minute. Examining one component of this calculation DLOPX', DLOPY' values are stored as four times the number of lanes of LOP phase change per 10 sec. Thus a velocity component is calculated in terms of a lane width factor (LWF) stored as LWF * 2^{-5} as

$$V_i = (LWF * 2^{-5}) * (2 * DLOP') * 2 = VEL_i * 2^{-4}$$

and is of the form $\pm$XXXX.XXXXXXX representing n.mi. per 20 sec. Integer multiply by three, i.e., rotate this number 1 place left (mul. by 2) and adding to the original number, yields n.mi. per minute expressed as VMIN * 2^{-4} with four significant bits to the left of the binary point allowing a maximum value for velocity in n.mi. per minute of 15.992. In each component calculation, the lane width factor has a precision on the order of 0.2 n.mi./lane. Since DLOPX, DLOPY values are limited to 0.246 lanes, the VMIN calculation should be accurate to within about 0.1 n.mi./min including round-off.

Velocity in n.mi. per minute is used to calculate estimated time of arrival (ETA) at the selected destination waypoint. The DTD stored as DIST * 2^{-10} in the floating point format is divided by VMIN in a two step calculation to preserve numerical precision. To insure a valid ETA calculation with the allowable range of DTD, it is necessary that VMIN $\geq$ 1.0 n.mi./min. This is a limitation imposed by the arithmetic divide routine (DIVF) in that all operations require the denominator to be larger in absolute value than the numerator. In calculating ETA, the DTD value is split into two parts, DIST1 and DIST2 where DTD = DIST1 + DIST2. The most significant two bytes of DTD are treated as DIST1. This is shifted right four bits to form DIST1 = DIST * 2^{-12}. When divided by VMIN, this yields the most significant part of ETA as ETA1 = TIME * 2^{-8}. The least significant byte of DTD, DIST2 = DIST * 2^{-8}, is divided by VMIN to yield ETA2 = TIME * 2^{-4}. At this point ETA2 is shifted right one byte to form ETA2 = TIME * 2^{-8}. Then ETA = ETA1 + ETA2 is calculated and shifted right three bits to form the integer estimate of minutes to destination. This is stored for operator display and has a maximum meaningful value of 256 mins on the display. The

round-off error in this calculation is insignificant (<0.5 min) and accumlated round-off should yield an ETA accurate to within 10 percent of the actual value.

The velocity in n.mi./hour or knots is required for display. The VMIN value is multiplied by the conversion constant 60 stored as $60 * 2^{-6}$ (i.e., $+1111_{(2)}$) and the result is shifted right one bit to get integer knots for output display. Round-off error in this calculation is at most 2.0 knots. With an error of 0.1 n.mi./min in VMIN the velocity can be in error on the order of 8 knots.

Before exiting navigation, the DTD value is shifted right three bits to form the integer value of distance to destination for display by the operator. A maximum value of 256 n.mi. is valid.

3.3.3 Arithmetic and trigonometric functions.— The arithmetic package used with the microprocessor includes a number of routines to facilitate 3-byte (12 bit) word operations and bit manipulation operations in addition to various arithmetic functions. For the arithmetic operations, three twelve bit registers termed A, B, and X are used. The functions can be described in terms of operations on numbers stored in these registers.

Arithmetic operations include fractional multiply (MULFF), subtract (SUBT), subtract with sign returned (SUBF), add (ADDF), and divide (DIVF). All of these routines assume that the arguments are stored in registers A and B upon initial entry. The routines ADDF, SUBT, and SUBF assume the binary points are aligned and treat the numbers as twelve bit signed numbers. The routines MULFF and DIVF assume the input value to be unsigned with the binary point to the right of the most significant bit. The sign of these operations must be determined and adjusted using other support routines.

In multiply (MULFF) register A is loaded with the absolute value of the multiplicand, and register B is loaded with the multiplier. The product is returned in A with the arguments destroyed. Multiplication is accomplished through successive left shifts of the multiplier coupled with right shifts of the multiplicand. The product is formed as an accumulation of shifted versions of the multiplicand, one addition for each bit set in the multiplier. The product is accumulated in a twelve bit register so that any bit shifted out of the multiplicand is lost without even the benefit of sum carries into the product register. The resulting round-off error can be significant.

The divide routine (DIVF) uses the value pre-loaded into the A register as the dividend and the B register contents as the divisor. To work correctly,both must be unsigned with contents (A) < contents (B). The dividend is shifted left one bit at a time and added to the complement of the divisor whenever it becomes at least as large as the divisor. A bit is set in the quotient register which is the A register on return for each such subtraction. No round-off error is experienced; however, there is truncation error such that the quotient can have eleven bits of significance and is always less than or equal to the true quotient within the resolution allotted. Using larger intermediate words for this operation would not improve precision.

The add routine adds the contents of registers A and B and returns the sum in register A. The microprocessor accumulator is set to 1 if the sum is not zero and the sign of the sum is set in the carry bit. In the subtract routine (SUBT), the difference in the contents of registers A and B is returned in A. A second subtract routine (SUBF) returns the sign of the difference in the carry bit and sets the accumulator to one if the difference is zero. These routines all provide maximum significance within the twelve bit word size.

Supporting routines include ABS(A) which replaces the contents of register A with its absolute value and sequences to a SGNA routine which returns the original sign of the contents of register A in the carry bit and sets the accumulator to 1 if A is zero. Other special purpose routines are not included in the description.

Word manipulation routines include RROLL to roll the contents of register A to B, B to X, and X to A. The routine SWAP is used to exchange the contents of registers A and B while STORE replaces contents of B with contents of A. Bit manipulation routines include ARS to perform a one bit right shift on the twelve bit contents of register A with repetition of the sign bit, and EARS which shifts the carry bit into the sign bit of A while accomplishing a one bit right shift. The routine ALS does a one-bit left shift of register A with a zero shifted into the least significant bit.

The trigonometric routines include sine, cosine, and arctangent. A supporting routine NRMANG is used to normalize an angle in the 0, 180° range before a cosine or sine operation. The sine routine uses an angle transformation (subtracting 90°) and calls the cosine routine. The cosine routine uses a truncated series approximation

$$\cos x = 1 - .49670x^2 + .03705x^4$$

from Abramowitz (ref.6) which has an error $\geq 9 \times 10^{-4}$. The routines will return the cosine or sine of any angle in the 0, 360° range with the proper sign. The arctangent routine returns an angle in degrees from a ratio z using

$$\text{ANGLE} = 215.625 * \left(\frac{z}{3.75+z^2}\right) \qquad (3\text{-}4)$$

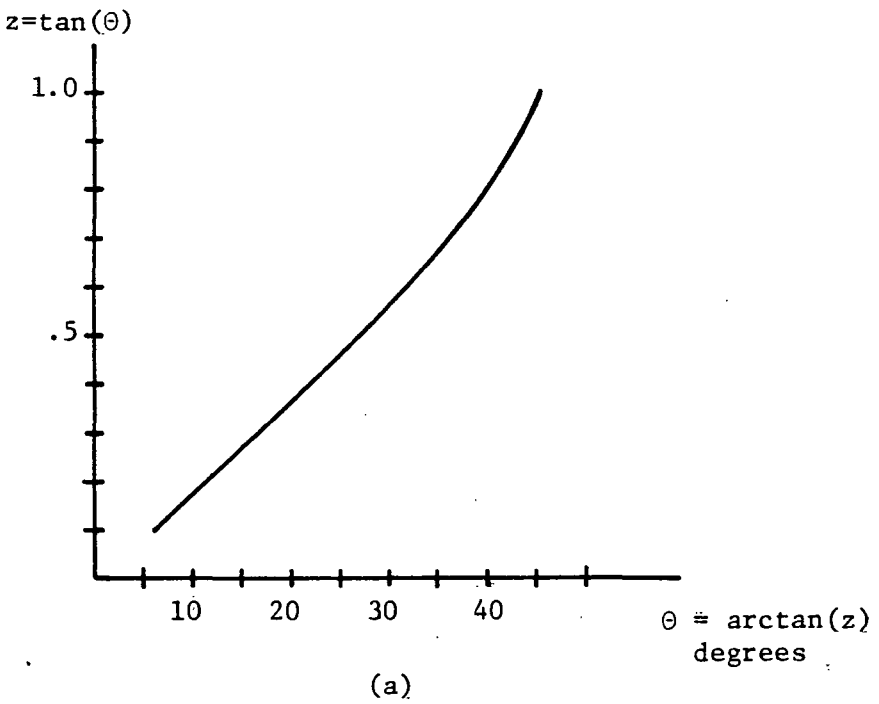
which is an approximation to a transformation in Abramowitz (ref. 6) given as

$$\text{ANGLE} = \frac{z}{1+.28z^2}$$

with ANGLE in radians. Figure 3-12 illustrates the error in degrees using (3-4) to approximate the arctangent (z) over a range 0, 45° since the arctangent function is not used outside this range.

3.4 Summary

This chapter has considered several aspects relating to the interface between the navigation receiver and the navigator. During the continuation of the development of software to implement the navigation function additional effort is needed to further evaluate the algorithms which are used in the receiver. The precision associated with computations for example is an important consideration. Generation of latitude/longitude from OMEGA phase position is important from the aspect of determining the value of this form of output with respect to the cost of implementation and the overall navigation accuracy obtainable. The discussion presented analytically describes the navigation outputs which compose the current concept of the navigation receiver. These will be the subject of flight test evaluation with particular emphasis on determining the receiver capability to provide these outputs with sufficient accuracy to be meaningful and to provide the navigator with adequate positional information.



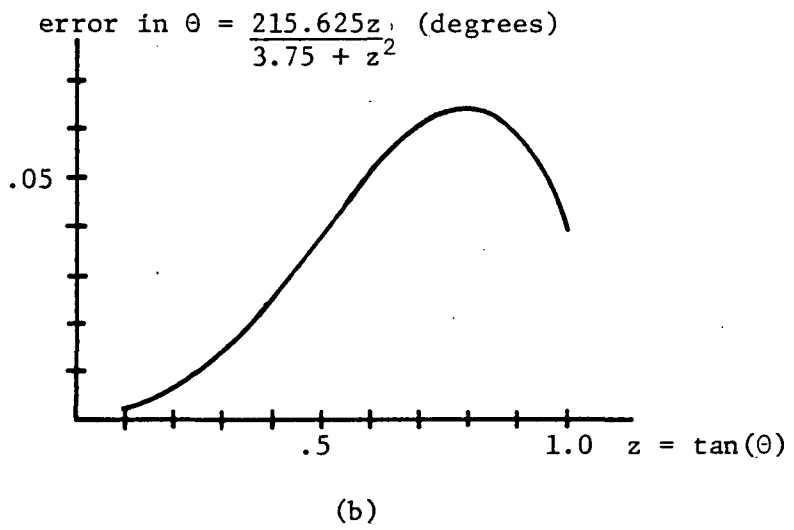


Figure 3-12. Illustration of Arctangent Function Precision Used In OMEGA Receiver. (a) Estimated Angle (degrees) vs. Tangent Ratio, (b) Error in Estimated Angle (degrees) vs. Tangent Ratio.

4.0 THE OMEGA NAVIGATION CHART AND PHASE VELOCITY ESTIMATES

In the context of preceding discussion of the airborne OMEGA receiver development, a product for producing OMEGA charts is important. The navigator must have a navigation chart, or at least chart information available, in order to provide the necessary input information to the receiver. Inherent in chart definition is the need for a suitable VLF phase velocity estimate or set of estimates (for each OMEGA transmitter) to determine chart LOPs in a region of interest. Resulting charts can account for propagation prediction correction, at least on the average, and can also account for modal interference which may be significant (see ref. 4). This chapter provides discussion relative to estimation of OMEGA VLF phase velocity through the use of a VLF propagation model and through the use of model generated propagation predictions (PPC). A procedure for developing OMEGA charts is described.

4.1 Use of VLF Propagation Model to Determine Phase Velocity Estimates

In the course of this investigation and an associated contract (see ref. 5). techniques for modelling VLF propagation have been investigated. A model has been developed which can be used to analyze multimodal propagation and when used in conjunction with real data, such as that obtained by NASA-LRC personnel, can provide a means of estimating phase velocity.

Intrinsic to the waveguide theory of VLF propagation is the representation of the EM fields as a sum of propagating modes. This mode sum is made tractable by introducing a notation to represent the different functional characteristics of the solution as defined by Wait (ref. 7) and subsequently extended by Galejs (ref. 8).

For a Vertical Dipole Source the mode sum becomes

$$E_r(h,d) = \frac{-\eta I d s e^{-i\pi/4}}{h\sqrt{a\lambda \sin(d/a)}} \sum_q S_q^{1.5} \Lambda_q^e G_q^e(h_s) G_q^e(h) \exp\{ik_o S_q d\} \quad (4\text{-}1)$$

where h is the distance above the ground surface; Λ_q^e is the excitation efficiency factor of the qth mode; $G_q^e(h_s)$ and $G_q^e(h)$ are the source and receiver height gain functions respectively; k_o is the freespace wave

number; d is the distance along the earth's surface from transmitter to receiver; Ids is the dipole current moment of the source; a is the earth's radius; λ is the freespace wavelength. The propagation parameter S_q for the qth mode can be considered as the sine of the complex angle of incidence at the ground, which is commonly used in the literature (ref. 7, 8, and 9). The superscript e in (4-1) indicates the vertical polarization of the source antenna. A waveguide mode as formulated in (4-1) has both TM and TE field components, due to the anisotropy of the ionosphere. However, at VLF frequencies, the field components of one type are usually dominant lending to descriptive terms such as quasi-TM and quasi-TE for the individual modes (ref. 10). To characterize the ground level field patterns of a VLF source located on the ground, the excitation factor and propagation parameter must be evaluated for each important mode. The height gain functions for this situation are defined to be unity.

Based on the theoretical development of Galejs, a propagation model was developed using a cylindrical coordinate system in the ionoshpere. The model yields results for phase velocity and attenuation which agree to high accuracy with published results for the NECL Waveguide Model. The propagation parameter S_q, which is determined by the model for each mode, is a complex number that describes the phase velocity V_q and the attenuation X_q in db/Mm for mode q as follows:

$$\frac{C}{V_q} = \mathrm{Re}\{S\}; \quad X_q = 8.686 \times 10^6 \ \mathrm{Im}\{S\} \quad .$$

Excitation factors A_q are also determined and are required when the effects of higher order modes are considered.

For a particular path the received VLF phase is dependent upon the frequency and path length as well as certain parameters that characterize the earth ionoshpere waveguide over the path. The required propagation model input parameters include the ground conductivity, magnetic latitude, magnetic azimuth, the ionospheric height profiles of electron density, and collision frequency.

The phase variation of the OMEGA signal with distance can be described by the phase velocity of the first TM mode when this mode is strongly dominant which is the intended situation for navigation with OMEGA. This phase velocity describes a linear variation of the received phase with distance.

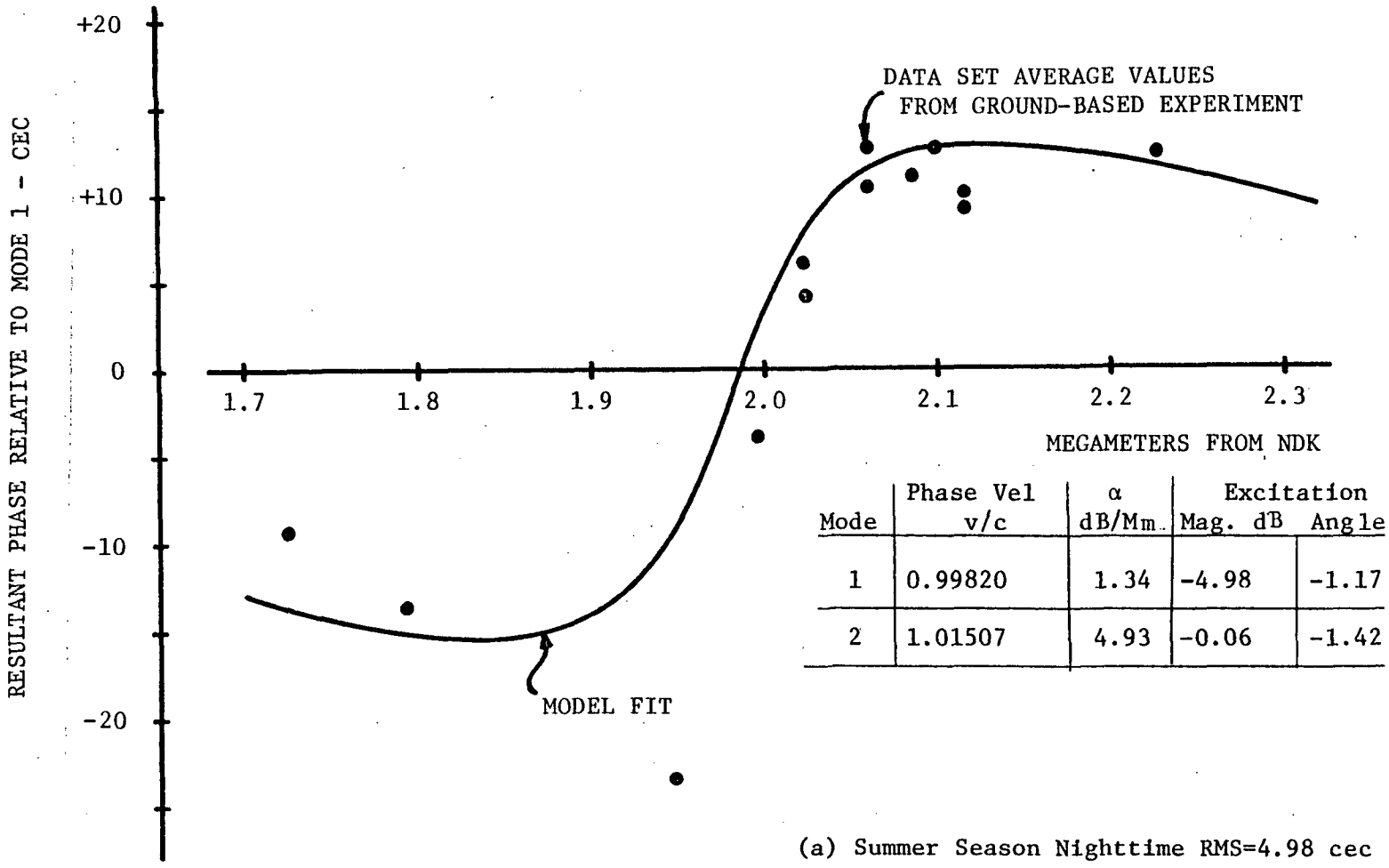
When higher order modes do affect the received phase an apparent phase velocity might be used to account for this effect while not complicating the calculations necessary at the receiving site in, for example, an airborne microprocessor based receiver. In most situations an apparent phase velocity could accurately account for modal interference in intervals of 100 to 200km along the path to the Omega transmitter of the contaminated signal. The apparent phase velocity at the destination could be used in an airborne receiver and provide some improvement in position estimates made from modally contaminated data.

The propagation model can be used to provide estimates of the phase velocity at a geographic location. For locations sufficiently close to an OMEGA transmitter to be affected by modal interference, an apparent phase velocity can be calculated from the phase of the total field predicted by the model if the path is in complete daylight or darkness and otherwise homogeneous. This condition is not very restrictive because of the limited separation of transmitter and receiver in such a case. Accurate phase velocity prediction for a signal contaminated by modal interference requires an accurate characterization of the ionosphere electron density profile. This is established by fitting phase data taken at different receiver ranges to the propagation model for the desired region. Phase contamination of the data by modal interference actually serves to improve the estimation of mode 1 phase velocity. This results from the sensitivity of the location of interference minima to small changes in electron density profile height, which also affects the mode 1 phase velocity. The OMEGA phase data taken by NASA-Langley personnel and analyzed by RTI has been used to fit the ionosphere profiles to agree with observed phase measurements of the North Dakota-Trinidad LOP taken over a period of several years. The fit was established by adjusting the ionosphere electron density profile to minimize the mean square phase difference between model predictions and measured nighttime phase averages at all receiver sites for a given season. A profile was determined for summer and winter and can be expected to be valid near the minimum of the sunspot cycle. The propagation model, using those profiles, yields mode 1 phase velocities which are in good agreement

with the OMEGA Propagation Prediction Corrections based on the assumed dominant first mode. The seasonal effect of ionosphere pattern on the nighttime 13.6 kHz mode 1 phase velocity is small (1-2 cec) and is not found in the PPC corrections available in 1974-1975 for the experimental region. The above seasonal shift in ionosphere reflection height is about half that shown in the profiles of Deeks (ref. 10). However the average heights in both cases predict phase velocities for mode 1 that, in the absence of modal interference, would yield a phase difference less than 1 cec at 8 Mm. The reduced size of the seasonal ionosphere shift can probably be attributed to the lower latitude at which the experimental data for this study were taken.

An application of the method described above for estimating phase velocities for the nighttime North Dakota signal begins with the least squares fit of the model predictions to the measured phase data shown in Figure 4-1. The phase data shown in this figure are nighttime averages of the NDK-TRI LOP and are referenced to the PPC predicted ode 1 phase for convenience in plotting. A plot of the predicted apparent phase velocity for the North Dakota signal, based on the preceding data fit, is shown in Figure 4-2, for a portion of the NDK-Hampton radial.

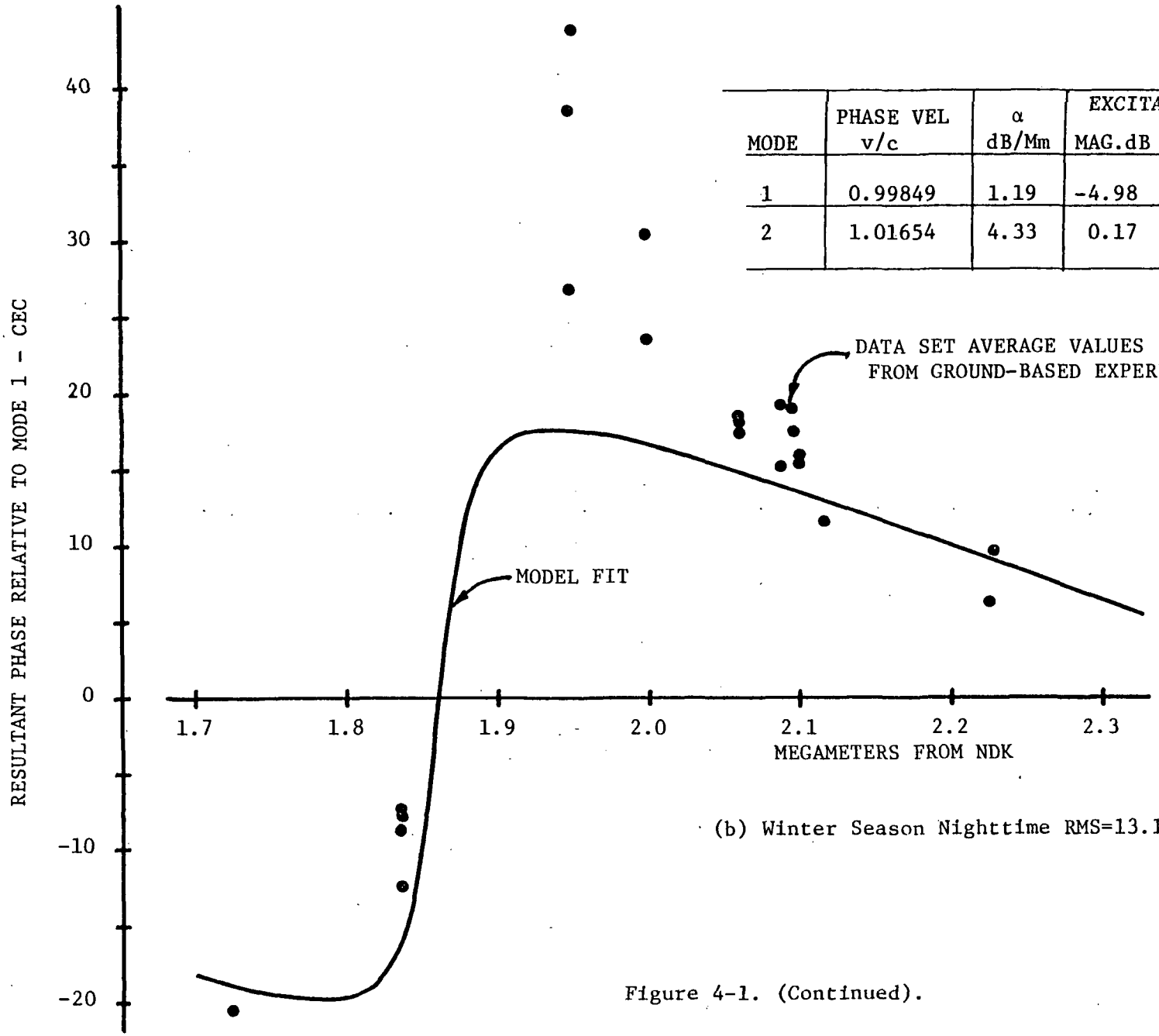
If modal interference is to be predicted for a given station (with present ability) data must be taken for the region of interest corresponding to the appropriate part of the sunspot cycle and season at a series of ranges. The data must be collected over a period of time long enough to allow determination of profiles which will predict the average phase behavior. Set-up times similar to those used in the NASA experimental program, (1-2 weeks) should be sufficient (see Ref. 1). The area accurately covered by the profile so determined can be expected to be reasonably large, for example, the middle Atlantic states might be covered by the data studied in this report. The possibility of correlating seasonal ionosphere profiles with magnetic latitude may increase, in the future, the area accurately covered by a set of radial phase measurements. This method could be used at present to provide improved navigation in key areas where a strong signal, providing good navigation geometry, is somewhat contaminated by modal interference as in the case of the nighttime 13.6 kHz North Dakota signal on the Atlantic east coast.



Mode	Phase Vel v/c	α dB/Mm	Excitation Mag. dB	Angle
1	0.99820	1.34	-4.98	-1.17
2	1.01507	4.93	-0.06	-1.42

(a) Summer Season Nighttime RMS=4.98 cec

Figure 4-1. Propagation Model Fit to Observed 13.6 kHz Phase Data.



MODE	PHASE VEL v/c	α dB/Mm	EXCITATION MAG.dB	Angle
1	0.99849	1.19	-4.98	-1.59
2	1.01654	4.33	0.17	-1.06

(b) Winter Season Nighttime RMS=13.18 cec.

Figure 4-1. (Continued).

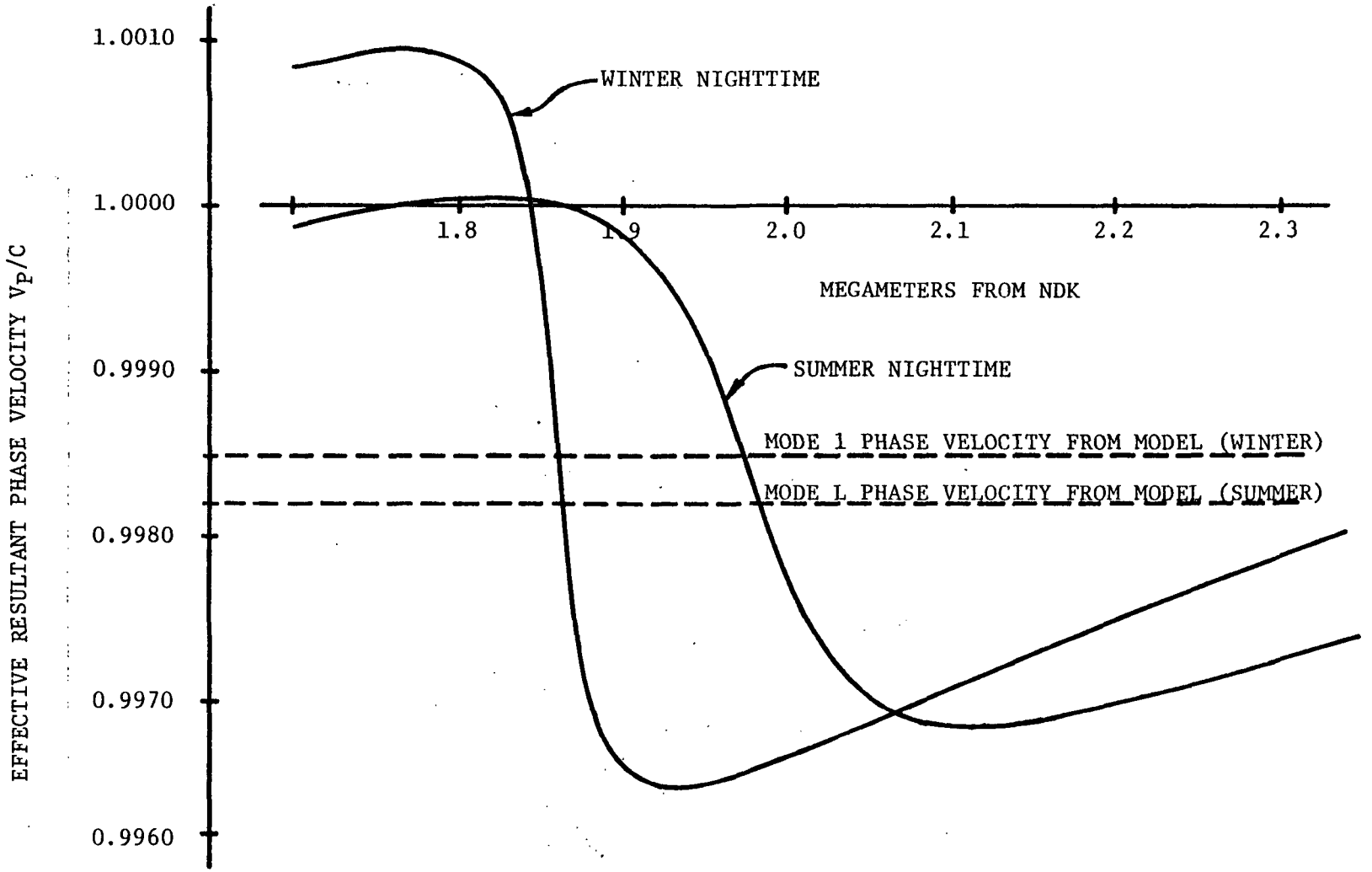


Figure 4-2. Effective Phase Velocity of 13.6 kHz Along NDK-Hampton Radial.

The current knowledge of ionosphere profiles is quite sufficient to provide accurate estimates of phase velocities to predict mode 1 phase alone. A nonhomogeneous path may be segmented in this case since it is assumed here that higher order modes originating at the source and produced by conversion at the path inhomogeneities is negligible at the receiver site due to the higher attenuation associated with these interferring modes. The phase velocity is calculated in each homogeneous segment and the apparent phase velocity at the receiver site is just the average of the segment phase velocities weighted by segment length.

4.2 OMEGA Phase Velocity Estimation Using Published PPC

In navigating with an OMEGA receiver the diurnal phase variation is of course the largest source of error. These errors and other less predictable corrections can be accounted for in various ways to improve navigation accuracy. The receiver can employ some correction model in algorithm form such as a polynomial generating equation (e.g. ref. 12) to generate real-time propagation prediction corrections (PPC) to enable the navigator to operate on the basis of phase measurements corrected to chart. Another method which can be advantageous particularly when operating in the difference frequency mode (e.g. 3.4 kHz) is to adjust the chart itself to account for an average PPC. As observed in a previous evaluation (ref. 13) the difference frequency diurnal phase variation is greatly reduced from that observed at the carrier frequencies. To arrive at a suitable correction, special geographically tailored PPC can be used to estimate an average reciprocal wavelength (phase $\propto$ reciprocal wavelength) which is equivalent to estimating a reciprocal phase velocity.

Considering the use of special PPC values of phase, the effective reciprocal wavelength of an OMEGA transmission can be estimated as a means of defining a perturbation from chart reciprocal wavelength. The predicted phase at frequency j for the transmission from station i is

$$\phi_{ij} = \frac{D_i}{\lambda_{ij}} = \frac{D_i}{\lambda_{cj}} - SWC_{ij} \qquad (4\text{-}2)$$

where D_i is the distance to transmitter i from the receiver and $\left(\frac{1}{\lambda_{cj}}\right)$ is the chart reciprocal wavelength at frequency j. SWC_{ij} is the PPC phase for the signal from transmitter i at frequency j in cycles. Then the average predicted reciprocal wavelength is

$$E_T\left\{\frac{1}{\lambda_{ij}}\right\} = \left[\frac{1}{D_i}\frac{D_i}{\lambda_{cj}} - E_T\left\{SWC_{ij}\right\}\right] \tag{4-2}$$

where $E_T\{\cdot\}$ represents a time average over some defined period. Then the effective reciprocal wavelength $\left(\frac{1}{\lambda_{ij}}\right)$ is defined as the estimate of the mean and from (4-2) is

$$\hat{\lambda}^{-1}_{ij} = \frac{1}{\lambda_{cj}} - \frac{E_T\{SWC_{ij}\}}{D_i} \tag{4-3}$$

Table 4-1 provides tabulated distances from Hampton to transmitters A, G*, C, D, and chart reciprocal wavelength for each OMEGA frequency.

For the difference frequencies

$$\phi_{i\ 3.4} = \frac{D_i}{\lambda_{c\ 13.6}} - SWC_{i\ 13.6} - \left[\frac{D_i}{\lambda_{c\ 10.2}} - SWC_{i\ 10.2}\right]$$

$$= D_i\left[\left(\frac{1}{\lambda_{c\ 13.6}}\right) - \left(\frac{1}{\lambda_{c\ 10.2}}\right)\right]\left[SWC_{i\ 13.6} - SWC_{i\ 10.2}\right]$$

or

$$D_i\left(\frac{1}{\lambda_{i\ 3.4}}\right) = D_i\left[\left(\frac{1}{\lambda_{c\ 13.6}}\right) = \left(\frac{1}{\lambda_{c\ 10.2}}\right)\right] - SWC_{i\ 3.4}$$

and the estimate of the difference frequency reciprocal wavelength is

$$\hat{\lambda}^{-1}_{i\ 3.4} = \left[\frac{1}{\lambda_{c\ 13.6}} - \frac{1}{\lambda_{c\ 10.2}}\right] - \frac{E_T\{SWC_{i\ 3.4}\}}{D_i} \tag{4-4}$$

*G - Trinidad location

TABLE 4-1

Tabulated Distances from Hampton Receiver Site to OMEGA Transmitters and Chart Value Reciprocal Wavelengths

OMEGA TRANSMITTER	DISTANCE TO LRC (m)
A - Norway	6.2709729695×10^6
G - Trinidad	3.2769646908×10^6
C - Hawaii	7.8738322838×10^6
D - North Dakota	2.0860930975×10^6

OMEGA FREQ.	CHART RECIPROCAL WAVELENGTH $_m^{-1}$
10.2 kHz	$(29468.087)^{-1}$
11 1/3	$(26521.279)^{-1}$
13.6	$(22101.066)^{-1}$

TABLE 4-2

Selected Yearly Average SWC Values Using Special Set of SWC for Hampton Obtained from Hydrographic Center

STATION		SWC (YEARLY AVG)(cec)	cec/m	NORMALIZED (YEARLY AVG) SWC
Norway	10.2	- 47.312	-7.54460	10^{-6}
Norway	13.6	- 120.337	-1.91895	10^{-5}
Norway	3.4	- 73.025	-1.16449	10^{-5}
Trinidad	10.2	- 26.077	-7.95767	10^{-6}
Trinidad	13.6	- 67.034	-2.04561	10^{-5}
Trinidad	3.4	- 40.957	-1.24984	10^{-5}
N. Dakota	10.2	- 26.454	-1.26811	10^{-5}
N. Dakota	13.6	- 52.034	-2.49433	10^{-5}
N. Dakota	3.4	- 25.58	-1.22621	10^{-5}

Once an estimate of reciprocal wavelength is obtained then predicted average phase from a given transmitter at a given frequency can be made using

$$\hat{\phi}_{ij} = D_i \left(\hat{\lambda}^{-1}_{ij} \right) \qquad (4\text{-}5)$$

where $\hat{\phi}_{ij}$ is predicted average phase. In composite OMEGA various estimates of a difference frequency phase are possible using different relative weightings of the selected pair of primary frequencies. For a given period the predicted difference frequency phase may be essentially constant such that a special chart for each such period could provide the navigator a means of locating position, or at least locating a primary frequency lane, very accurately without need for skywave corrections. To generate such charts an estimate of reciprocal wavelength is needed. One such estimate can be made using the skywave corrections calculated by the Hydrographic Center. Of course any inherent errors in these corrections will show up in the reciprocal wavelength estimate.

In Table 4-2 selected yearly average skywave corrections are presented which were calculated from the special set of SWC obtained for the Hampton, Va., receiver site from the Hydrographic Center. Using the distances to transmitters given in Table 4-1 normalized average SWC values are also given in Table 4-2. Table 4-3 lists the calculated reciprocal wavelengths and inverse reciprocal wavelengths using Tables 4-1 and 4-2 with (4-3) and (4-4). Also included are estimated chart phase values for each station and station pair at the selected frequency using (4-5). The chart values using the chart reciprocal wavelength are also shown in Table 4-3 for information.

4.3 Development of OMEGA Chart Lattice Grids

In developing the application of the OMEGA navigation system for general aviation use it is desirable that OMEGA lattice information be integrated with conventional nautical charts. In accomplishing this objective, the approach used has been to develop a procedure for overlaying an OMEGA

TABLE 4-3

Calculated Reciprocal Wavelengths

Station*	10.2 kHz Estimated Reciprocal $\lambda(m^{-1})$	10.2 kHz Inverse of Estimate (m)	10.2 kHz Chart λ_c (m)	13.6 kHz Estimated Reciprocal $\lambda(m^{-1})$	13.6 kHz Inverse of Estimate(m)	13.6 kHz Chart λ_c (m)
Norway	3.40105×10^{-5}	29402.717	29468.087	4.54386×10^{-5}	22007.729	22101.066
Trinidad	3.40146×10^{-5}	29399.147	29468.087	4.54512×10^{-5}	22001.596	22101.066
N. Dakota	3.40618×10^{-5}	29358.378	29468.087	4.54961×10^{-5}	21979.897	22101.066
Norway-Trinidad	3.39689×10^{-5}	29438.711	29468.087	4.53317×10^{-5}	22059.625	22101.066
Norway-N. Dakota	3.39849×10^{-5}	29424.870	29468.087	4.54099×10^{-5}	22021.630	22101.066
Trinidad-N. Dakota	3.39319×10^{-5}	29470.836	29468.087	4.53726×10^{-5}	22039.712	22101.066

Station	3.4 kHz Estimated Reciprocal $\lambda(m^{-1})$	3.4 kHz Inverse of Estimate(m)	3.4 kHz Chart λ_c (m)
Norway	1.14281×10^{-5}	87503.447	88404.261
Trinidad	1.14367×10^{-5}	87438.142	88404.261
N. Dakota	1.14343×10^{-5}	87456.212	88404.261
Norway-Trinidad	1.13628×10^{-5}	88006.406	88404.261
Norway-N. Dakota	1.14250×10^{-5}	87527.012	88404.261
Trinidad-N. Dakota	1.14408×10^{-5}	87406.507	88404.261

*All values based on SWC at Hampton receiver site

grid onto conventional topographical maps used by airborne navigators. The standard sectional aeronautical charts used in the United States and approved by DOD, FAA, and DOC are Lambert Conformal Conic Projections with information specifically designed for use in airborne navigation.

The Lambert topographic projection produces a map which defines points on the earth's surface in a rectilinear coordinate system such that distances between points on the surface is preserved. The projection itself is common and is described in various documents (e.g. ref. 14 and 15). Commonly, transformations are defined in terms of conversions between Lambert x, y coordinates and latitude/longitude coordinates. Convenient procedures for conversion of latitude/longitude to OMEGA LOP values have been used previously (ref. 1) and procedures for the inverse transformation are discussed in Chapter 3 and Appendix C.

A computer program has been developed to generate and plot points along OMEGA LOPs for any selected pair of transmitting stations using any appropriate estimate of phase velocity or wavelength. Plots are made in the rectilinear coordinate system of the Lambert projection. Appendix D provides a description of the program FLATBED which was written in a joint effort by RTI personnel and LTV Aerospace contractor personnel at NASA-Langley. The program provides for plotting lines of latitude and longitude as well as LOPs on the Langley Research Center flatbed plotter. An additional feature allows for plotting of an OMEGA derived course on the chart. With slight modification the latitude-longitude lines plot can be suppressed so that the program can be used to plot an OMEGA grid directly onto a Lambert projection topographical map for relatively small areas comparable to aeronautical charts. This program is designed for worldwide use with certain restrictions. Latitude-longitude boundaries for a particular area of interest may encompass the Equator or 0° longitude but no provision has been made for crossing the poles or for a grid encompassing 180° longitude. Figure 4-3 is a sample plot of output in the form of 3.4 kHz OMEGA lanes superimposed on a latitude-longitude grid. The OMEGA lanes have not been numbered but could be easily determined using the same algorithm defined by the Hydrographic Center of the Defense Mapping Agency which publishes the OMEGA charts.

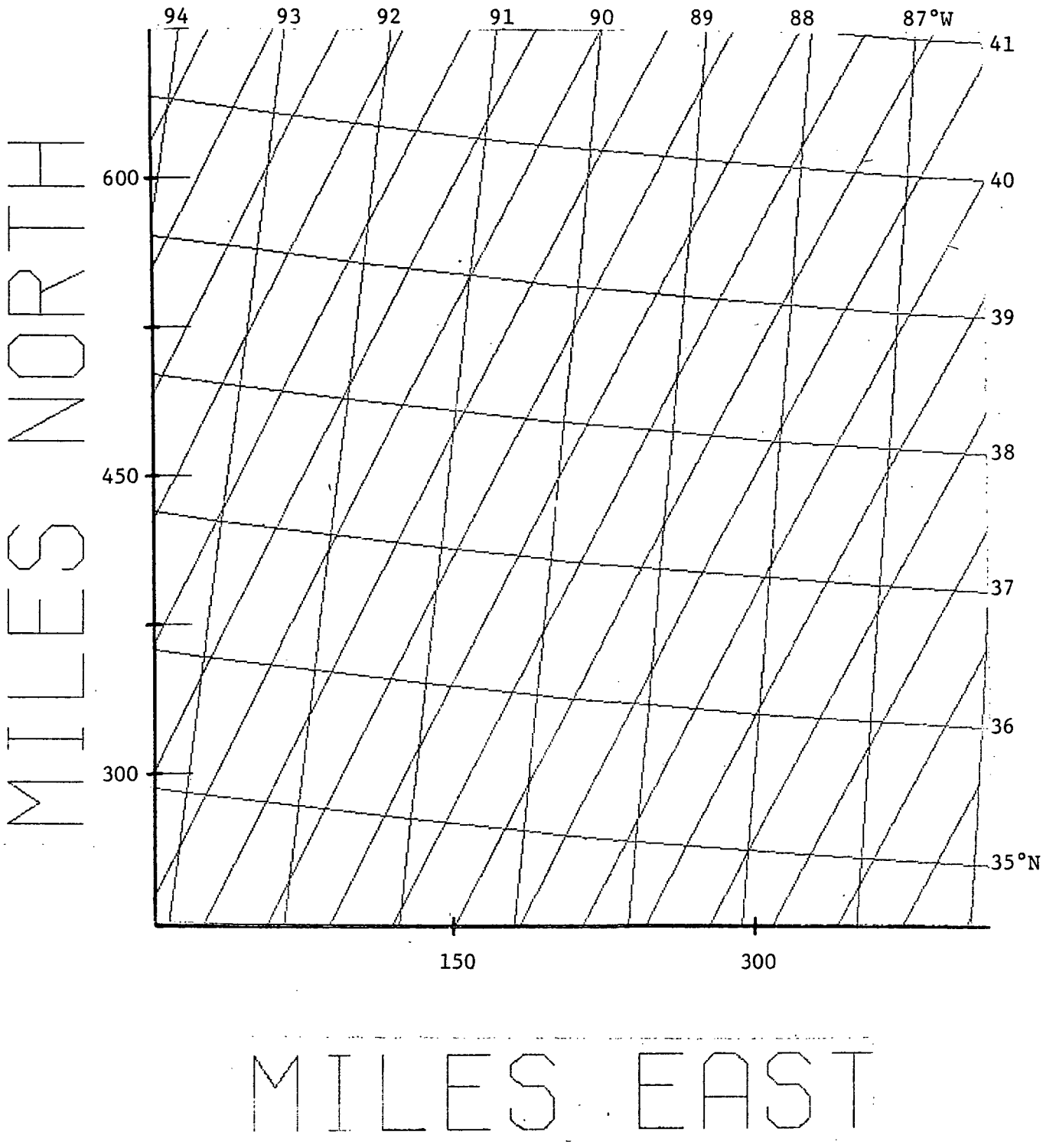


Figure 4-3. 3.4 kHz OMEGA Chart Superimposed on Latitude/Longitude Grid as Plotted by Computer Program on CDC-6600.

5.0 FLIGHT TEST EVALUATION

To evaluate the microprocessor based OMEGA navigation receiver a series of flight tests have been made using the facilities at Wallops Flight Center. A C45 aircraft was equipped with a rack mounted version of the OMEGA receiver and digital tape recording equipment. During flights the NASA Wallops tracking radar was used to provide "true" position information in the evaluation of the OMEGA receiver navigation accuracy. The tests have been designed not only to evaluate receiver accuracy but to analyze the navigation related outputs which are computed, various automatic features of the receiver including initial OMEGA format lock-up and lane-counting, and general pilot or navigator acceptance.

A digital flight recorder was associated with the receiver to record certain RAM memory locations at ten second intervals. The integrator (12 bits) and PAR (8 bits) values associated with each of the four phase-locked loops were recorded for the 10.2 and 13.6 kHz carrier frequencies. The "PAR" value represents the loop phase value at the end of the last measurement interval for the associated OMEGA transmitter station and frequency. The current LOP station pairs as were designated by the operator along with the LOP gradient angles and magnitudes were recorded. Also the LOP skywave corrections which were derived differentially and input by the operator are recorded. Other recorded input values include the lane, innerlane, and fractional lane values for the designated waypoints. Navigation output values recorded on tape include current position lane, innerlane, and fractional lane value at each ten second time, current heading angle to waypoint, distance to destination, velocity, ground track heading,heading error, cross-track deviation, and estimated time of arrival at the designated waypoint. A time code in seconds since the beginning of the day enables comparison with the radar position data recorded separately at the radar.

In evaluating positional accuracy of the OMEGA receiver the ten-second LOP measurements were used in post-flight analysis to calculate latitude/longitude and Lambert projection x,y position. Since the tracking radar elevation, range, and azimuth data were converted to a local x,y system the OMEGA

derived Lambert x,y positions were transformed to the so-called "Wallops x,y system". The Wallops coordinate system is simply a rectilinear coordinate system on a plane tangent to the Wallops runway onto which the radar coordinate positions (polar coordinates) are projected. The "y" axis is oriented north. The transformed OMEGA data yields a standard Lambert conical topographical projection coordinate position. A linear transformation was derived to convert the Lambert coordinates to Wallops' coordinates for position error evaluation. Using translation and rotation a minimum mean square transformation was obtained

$$\bar{X} = [M_1]\ [\bar{X}' - \bar{X}_o]$$

where $[\bar{X}']^T = [x'\ y']$ is the Lambert position, $\bar{X}_o$ is the origin of the Wallops system in the Lambert system (x_o = 178453.8m, yo = 714903.8m), and

$$[M_1] = K\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

is a rotational transformation with $\theta = 1.19646°$, K = 6030/1852 ft/m.

5.1 Flight Test Results

Flight ONR-3 in April, 1977, consisted of five legs. From Wallops Island, VA, a course heading of approximately 115° was flown over the Atlantic Ocean for a distance of approximately 92km (50 n.mi.). A return to Wallops on a heading of ~295° was followed by a leg from Wallops to Cape Charles VOR, a leg from Cape Charles to Harcum VOR, and a leg from Harcom to Wallops. Figure 5-1 is a radar track of this flight. The VOR site locations are approximate and two of the Norway-Hawaii and Trinidad-North Dakota LOPs are illustrated. Note that the radar track did not start until the aircraft was about 28km (~15n.mi.) away from Wallops and that loss of track occurred on the return leg over the ocean. This representation is based on five second radar position estimates obtained from the NASA Wallops facility. The aircraft airspeed is estimated at about 150-160 kts.

The receiver did not have the smoothed velocity estimation algorithm incorporated and the complete data tape of navigation outputs was not available for this early flight. To investigate the sample-to-sample velocity estimate, data from a previous flight (March 23, 1977) on a C54 aircraft flying a similar course over the Atlantic was used. The TRI-NDK LOP data was used on the two

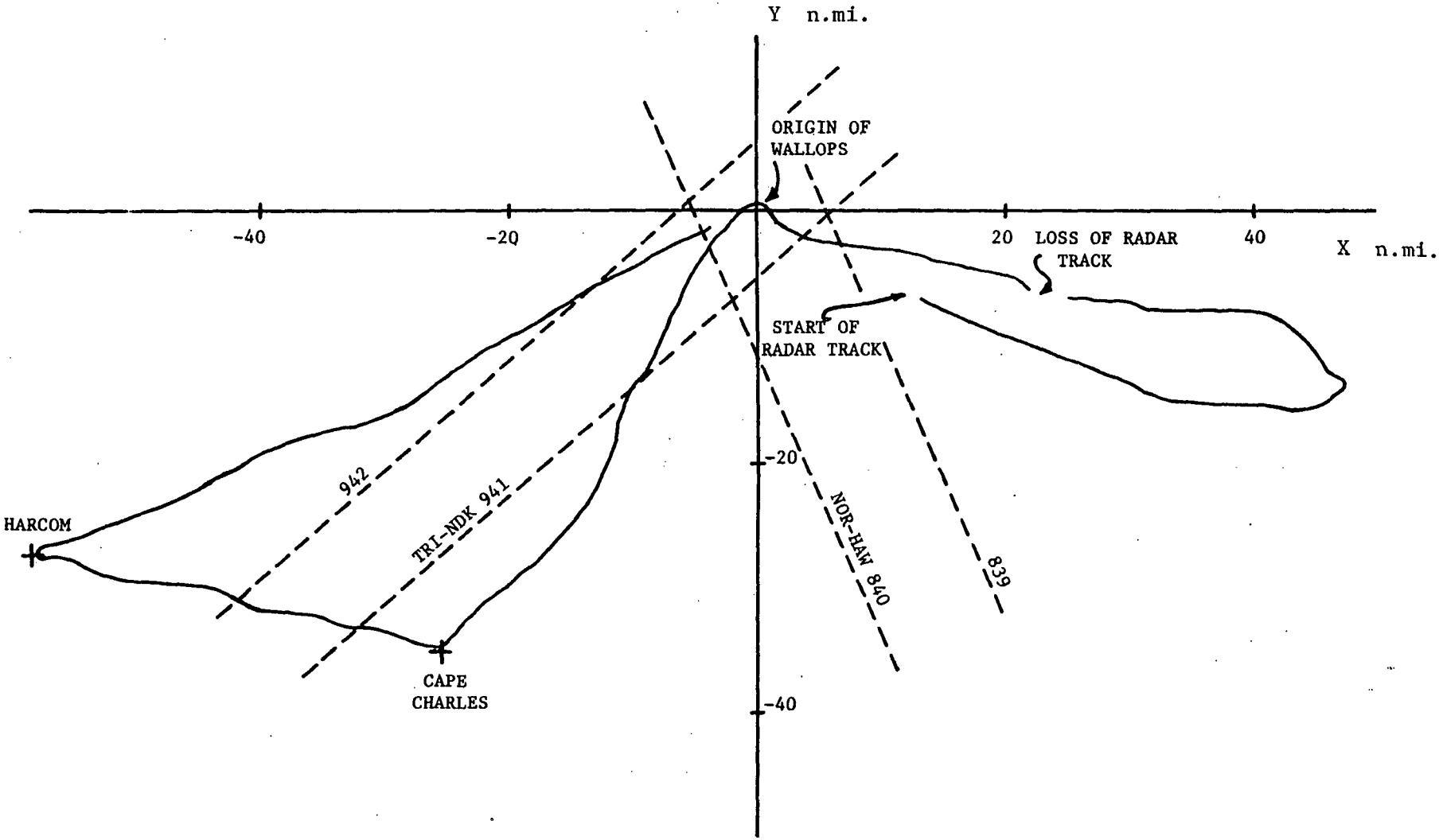


Figure 5-1. Radar Track of Flight ONR-3.

legs over the Atlantic to provide velocity estimates during post-flight analysis. Each ten second LOP phase measurement was differenced to estimate velocity. Two traces are illustrated in Figure 5-2. A ground speed velocity scale is shown to the left of the ordinate and was derived assuming a course heading of 115° out, 295° return, and a 139° gradient direction for the TRI-NDK LOP which has 6.56 cec/km. As can be seen the mean of these estimates should provide a good estimate of ground-track velocity, however, ten-second values are too noisy to be useful. As mentioned in Chapter 3, the estimated time of arrival, ground-track heading, and heading error are dependent on this velocity estimate also.

As the microprocessor based OMEGA navigation receiver was initially set up, the loop phase measurements were recorded on tape once each ten seconds. The recording was made during a gap in the OMEGA measurement format in that only four stations are phase tracked at two frequencies. Thus at the time phase measurements are recorded the time delay associated with the various station phase values varies over the previous ten second interval. Furthermore since the phase-locked loops are second order, each recorded phase value is a prediction for the measurement to follow. Ideally, each phase measurement should be translated in time so that the recorded values represent concurrent phase estimates either at the time of recording or at some known time displaced relative to the time of recording. This problem is really inherent in OMEGA and only is significant when the receiver is operating in a rapidly moving vehicle such as an aircraft. Any clock frequency offset in the receiver will also contribute to this problem since this just adds to the effect of vehicle motion creating a greater phase change with time.

In evaluating the data recorded from the flight tests, radar position information is used as true position. The radar position data has been provided at five second increments in time. To evaluate navigation accuracy it then is necessary to time synchronize the OMEGA derived position with the radar derived position for meaningful analysis. This has been done in post-flight analysis by incorporating a time translation between radar fixes and OMEGA fixes derived from recorded phase measurements. The method assumes some mean time of OMEGA measurements and compares the OMEGA position fix at some time t_i with the radar fix at $t_i - \Delta t$. The information needed to

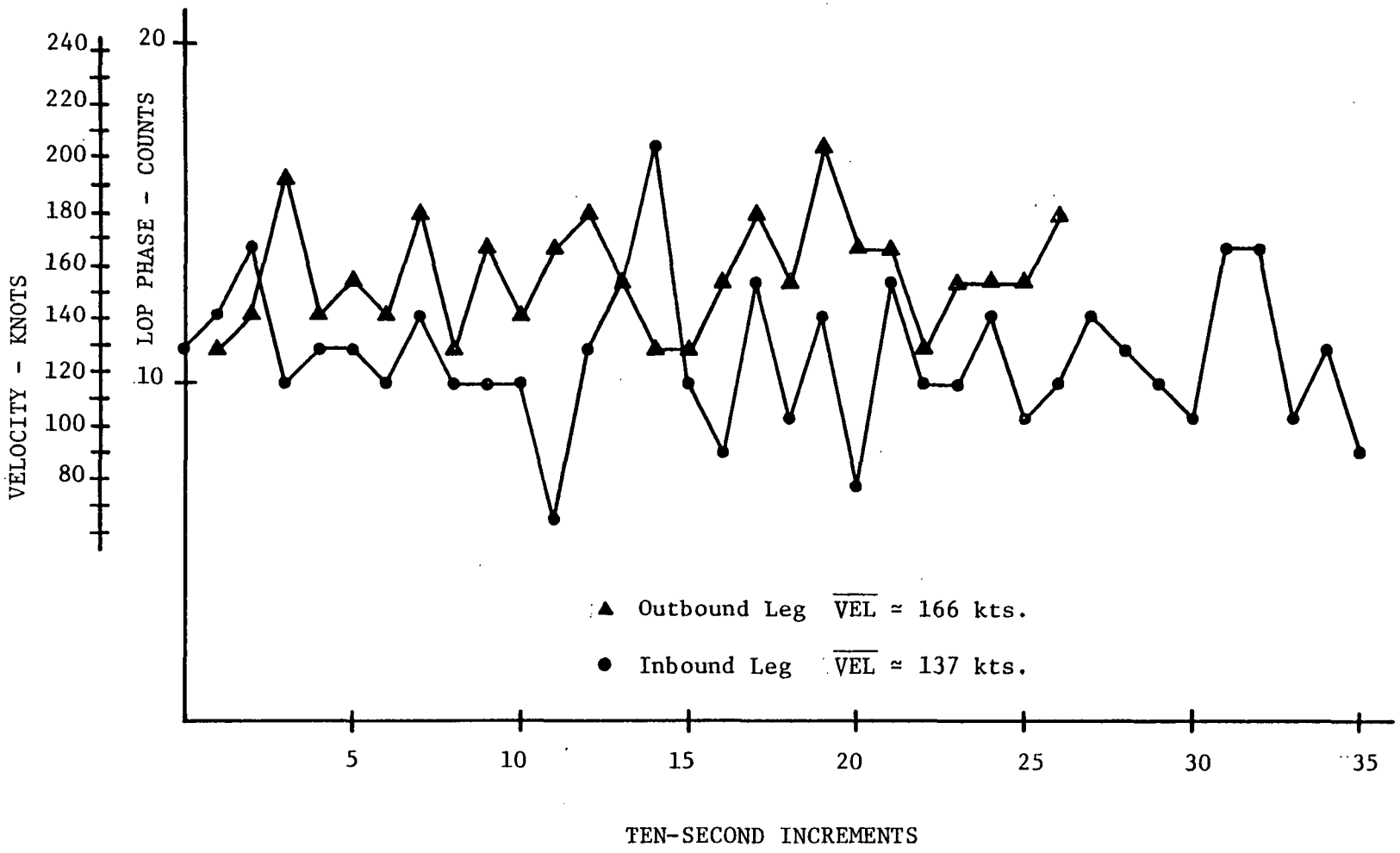


Figure 5-2. LOP TRI-NDK Phase Gradient Based on 10-second Samples.
Flight ONR-3 LEGS 1 & 2

translate each OMEGA station phase to a common point in time was not available. For later flights it is anticipated that this will be an automatic procedure within the receiver.

5.1.1 Flight ONR-3. - Considering data for flight ONR-3 (March 25, 1977) position error analysis was made using a Δt=8 secs (or 9 secs)* between radar position data and OMEGA position error. This means that the LOP value corresponding to a time t_i ($t_i - t_{i-1}$ = 10 seconds) is used to estimate position and compared to the radar position Δt seconds earlier. This choice for Δt may tend to produce an offset in the position error since the LOP position at time t_i is a predicted position ten-seconds ahead. The phase measurement at the end of any particular station measurement interval is the predicted phase for the next measurement interval. Thus, a slight directional offset in position error is conceivable with the error direction depending on the direction of flight. The OMEGA position estimates would tend to be ahead of the radar positions.

Figures 5-3 through 5-10 present results of ONR-3 flight analysis. LOPs AC (Norway-Hawaii) and GD (Trinidad-N. Dakota) were used to derive OMEGA position estimates. Figure 5-3 is a plot of relative position error for the entire flight using 10.2 kHz OMEGA data (see Figure 5-1). The major error spread is definitely in the northeast-southwest direction and can be attributed primarily to noise in the Norway measurement. Superimposed are a set of axes which allow the position error data to be interpreted as skywave corrected. The Δx, Δy displacement from the uncorrected origin is based on the hour 19 published SWC values for the four stations used. The data were taken during hours 18, 19, and 20. With SWC the spread is more nearly centered with respect to Norway but the Trinidad-N. Dakota offset is somewhat worsened. In Figure 5-4, the hour 18 data are shown corrected with the tabulated SWC. This represents data taken during the flight out and back over the Atlantic Ocean.

---

* refers algebraically, to time of OMEGA position based on recorded value phase measurement time minus radar time. A gap in the radar track caused a one second shift in the radar position times in the middle of the flight so that Δt=8 was not possible during the entire flight.

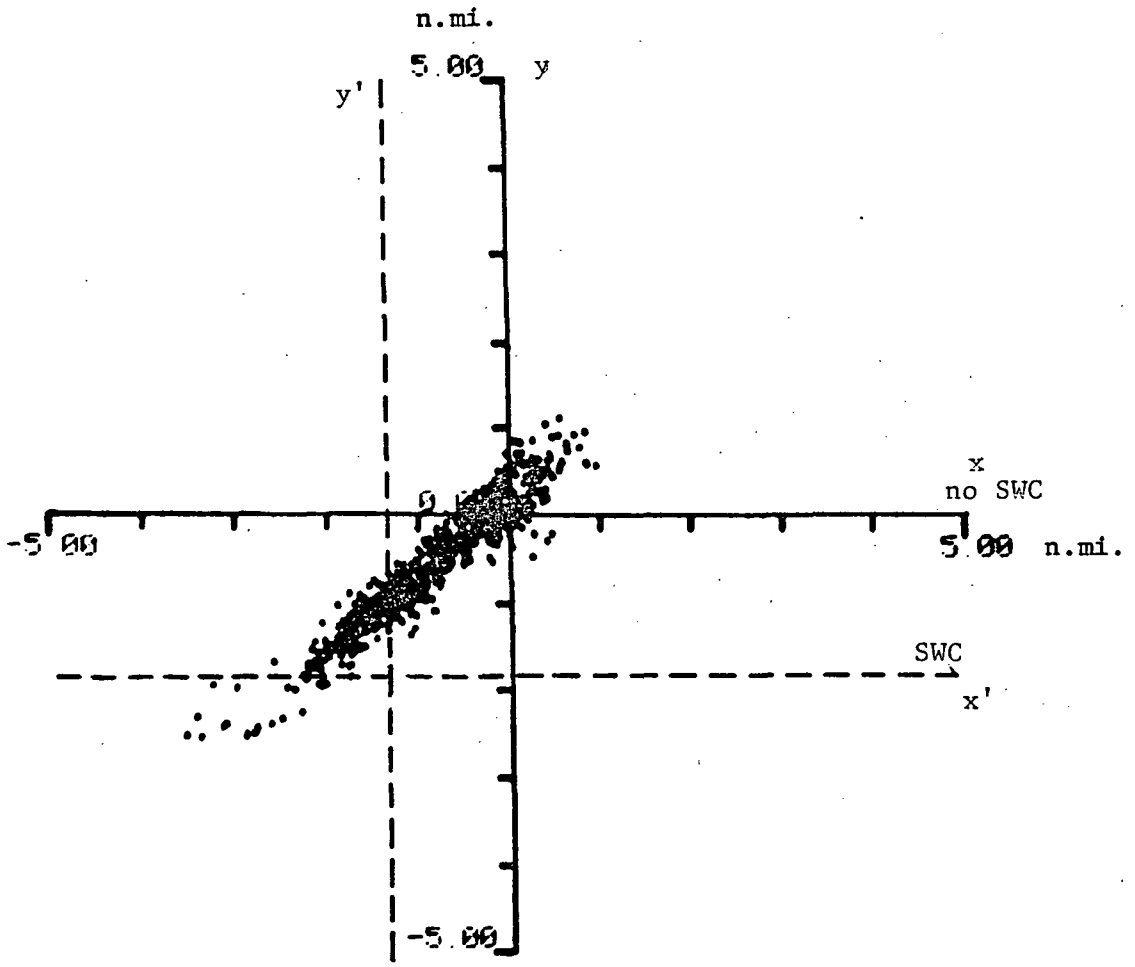


Figure 5-3. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-3 at 10.2 kHz using LOPs NOR-HAW and TRI-HAW. Δt = 8 secs.

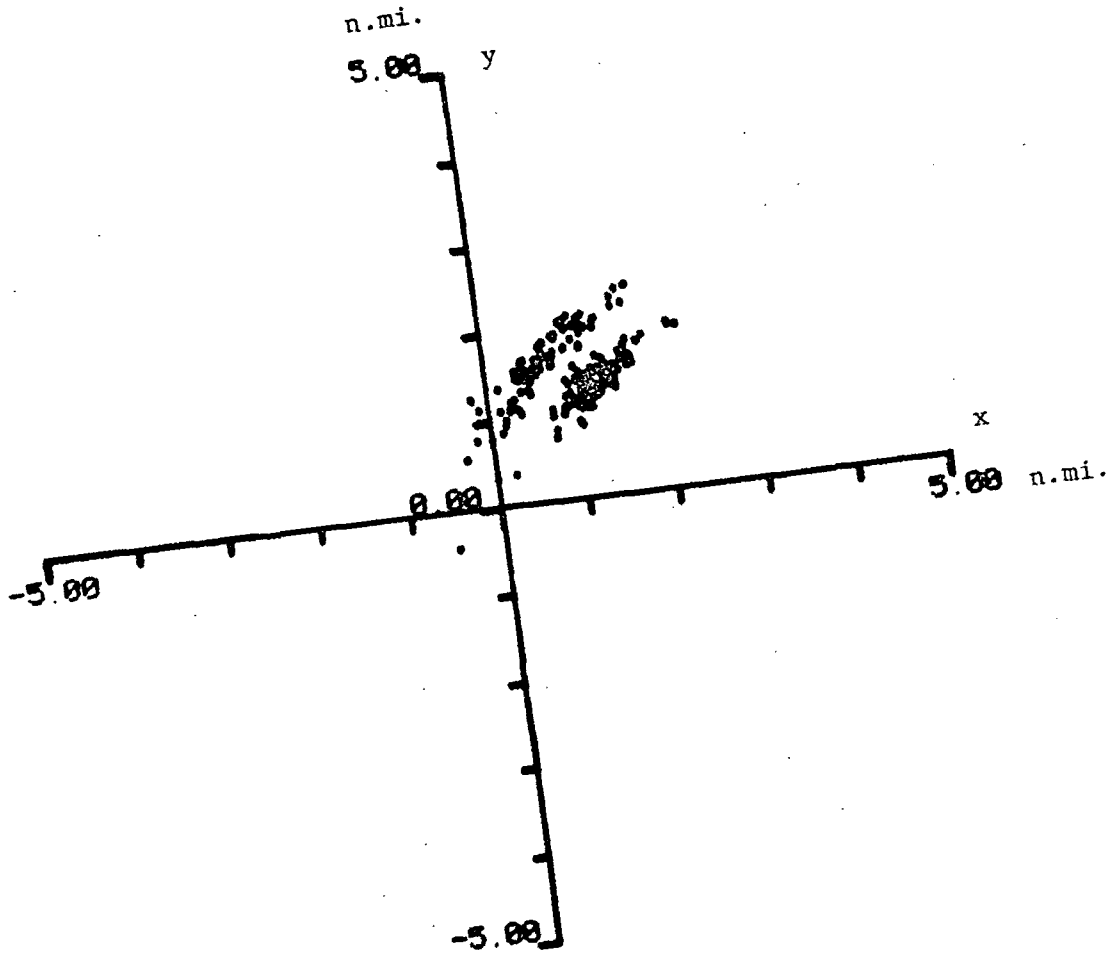


Figure 5-4. Skywave Corrected OMEGA Derived Position Error Relative To Wallops Radar Fixes for Flight ONR-3 at 10.2 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 13 secs.

Here the Δt has been increased to 13 secs. Note there are two distinct groupings which illustrate the effect of the predicting ahead characteristic of the OMEGA receiver. The group of points to the lower left can be attributed to the flight out while the other group can be attributed to the return leg. The greatest error variation is in the Norway direction and some offset error is evident even with corrections. Figure 5-5 is relative position error on the same flight using 13.6 kHz data. Note that the variation in the Norway direction is somewhat reduced and the offset of the corrected data mean is greater than with the 10.2 kHz data. Figure 5-6 is uncorrected OMEGA position error with 3.4kHz difference frequency phase. In Figure 5-7 position error (uncorrected) is plotted relative to aircraft heading. The effect of the OMEGA being a prediction ahead as well as inherent phase lag of the OMEGA loops during turns is clearly evident. Using separate means for the different groups of data points Figure 5-8 illustrates this same data with mean corrections. This illustrates that locally derived differential corrections can be quite useful. Figure 5-9 provides mean corrected position error data for 3.4kHz navigation plotted relative to aircraft heading. In these figures aircraft heading has been derived from pairs of position points of the radar tracking data. In Figure 5-10 the effect of phase lag in a turn is demonstrated. From Figure 5-1 the Harcum VOR is at the end of the middle leg in the triangular course. As the turn is made the OMEGA position estimate continues in a westerly direction and then changes direction rather rapidly as the loops respond. There is a rather large mean error which tends to move away from the aircraft direction of motion indicative of phase measurement lag during the early half of the turn.

5.1.2 Flight ONR-5.— For flight ONR-5 (April 20, 1977) the Δt value was varied from -4 secs (-5)* to 10 secs (11)* to evaluate the effect of phase prediction. Flight ONR-5 consisted of four legs flown in a box pattern in

---

* a gap in radar data inserted a one second time shift in part of the recorded positions.

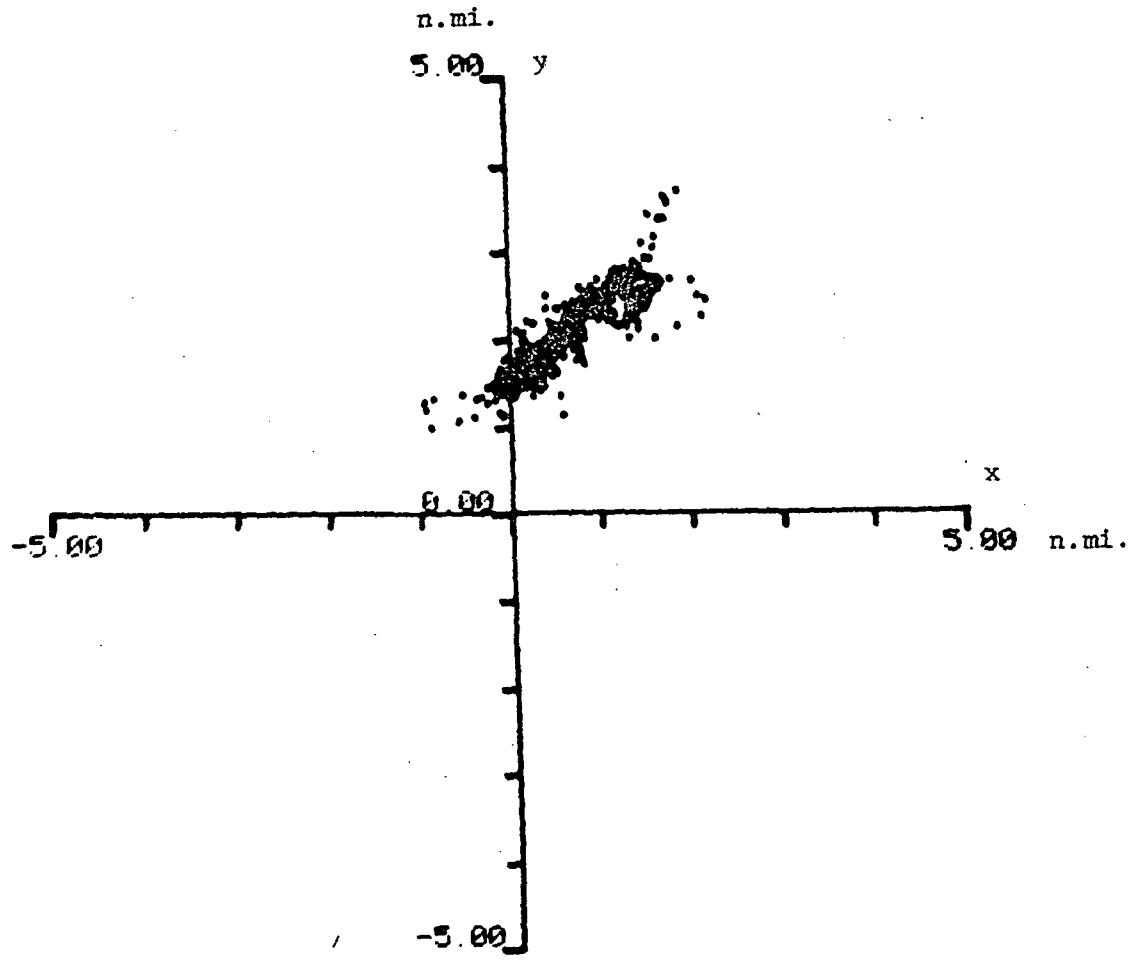


Figure 5-5. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-3 at 13.6 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 8 secs.

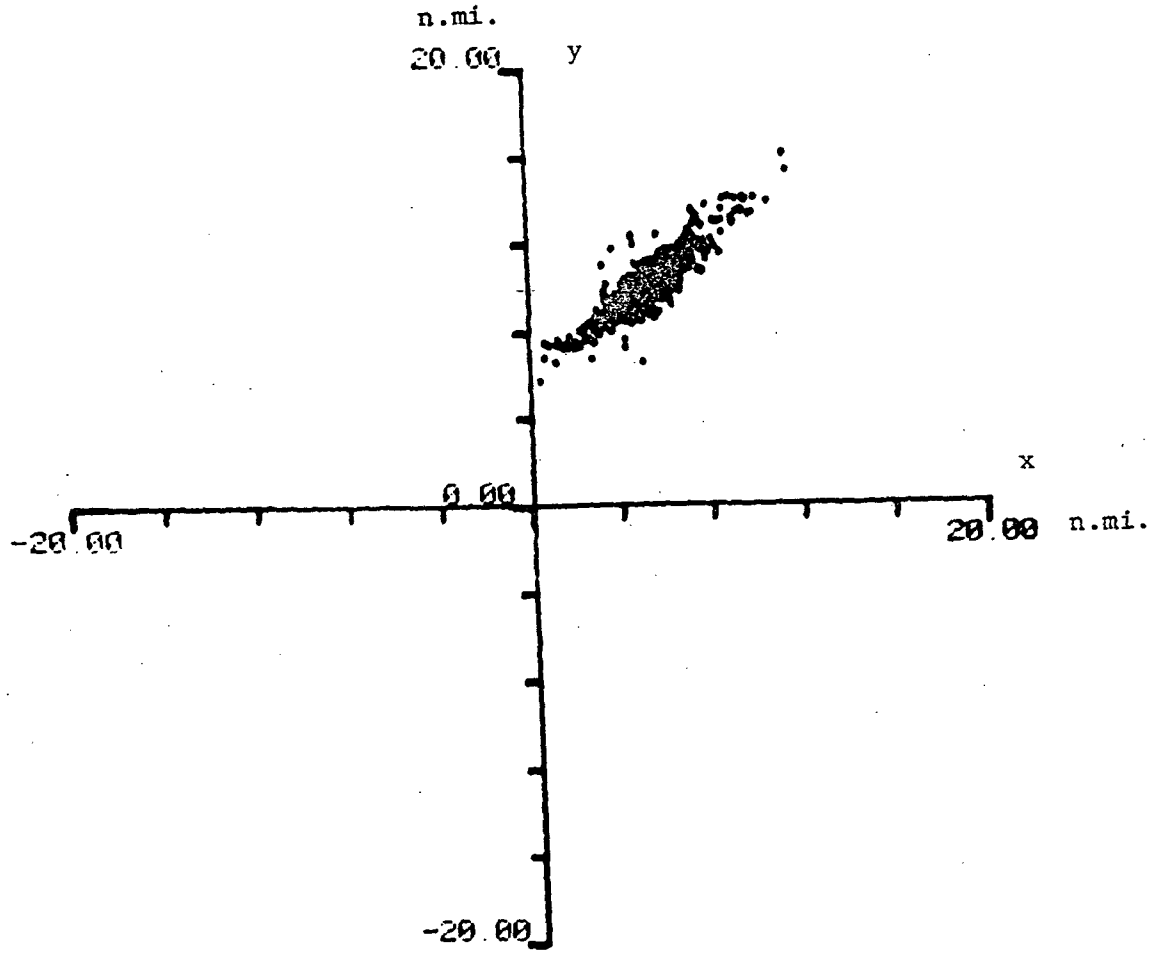


Figure 5-6. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-3 at 3.4 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 8 secs.

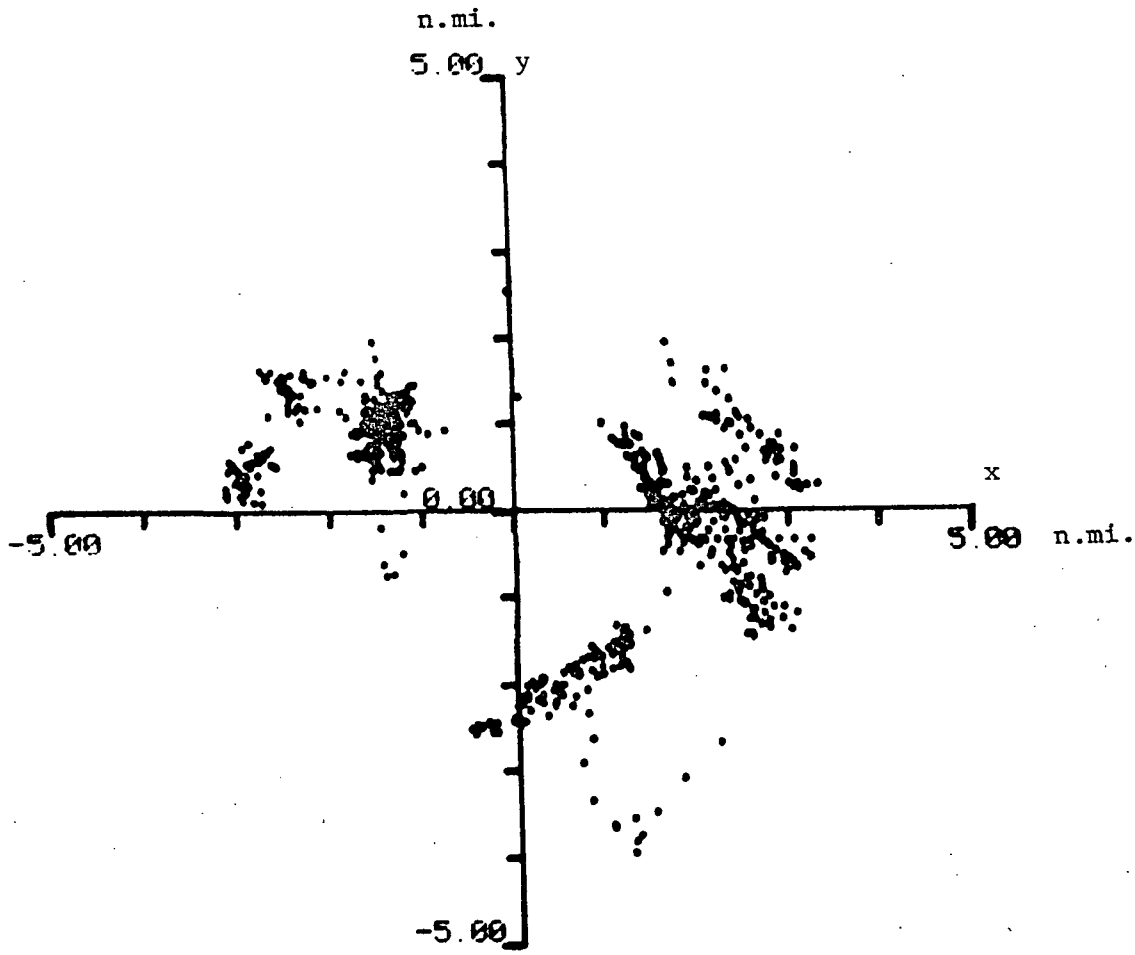


Figure 5-7. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-3 at 13.6 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 8 secs.

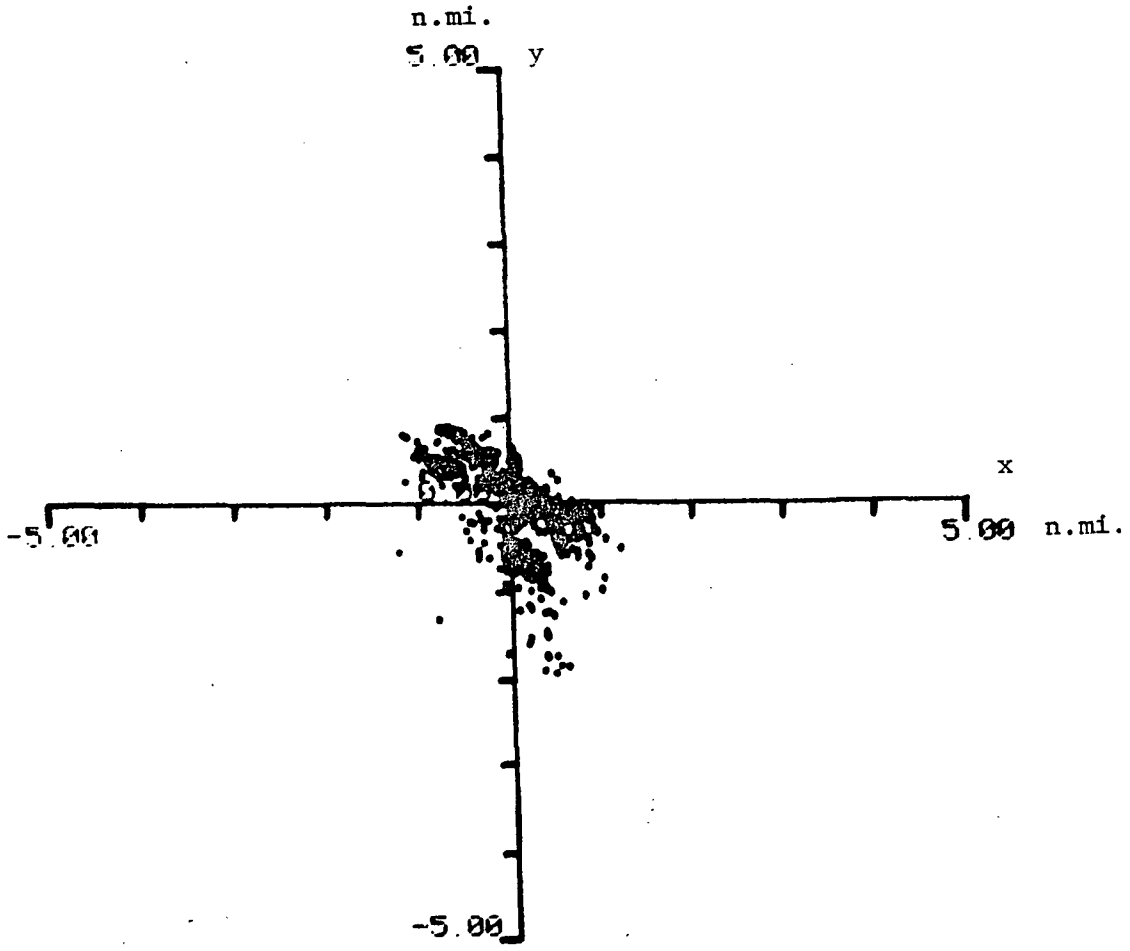


Figure 5-8. Mean Corrected OMEGA Derived Position Error Relative to Aircraft Heading Using Wallops Radar Fixes for Flight ONR-3 at 13.6 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 8 secs.

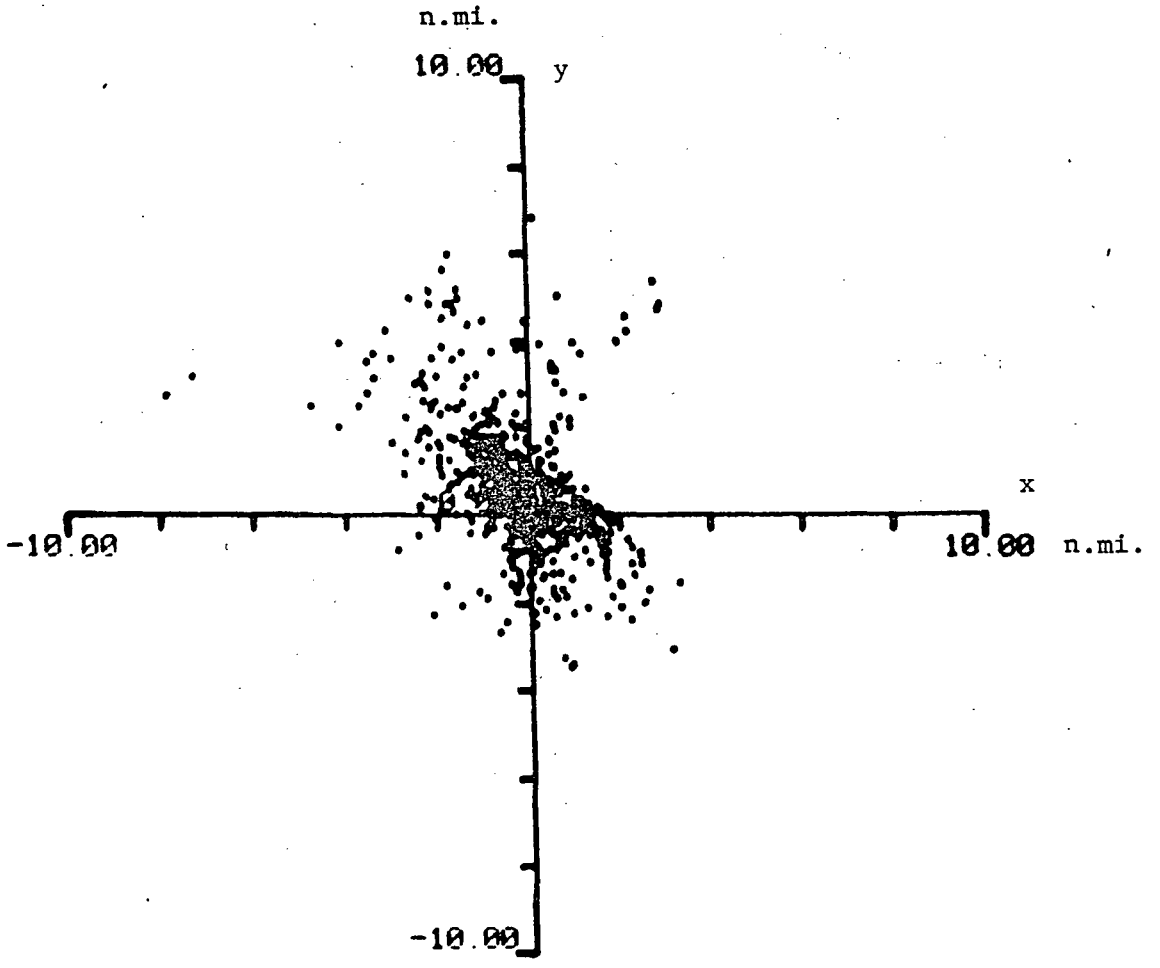


Figure 5-9. Mean Corrected OMEGA Derived Position Error Relative to Aircraft Heading Using Wallops Radar Fixes for Flight ONR-3 at 3.4 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 8 secs.

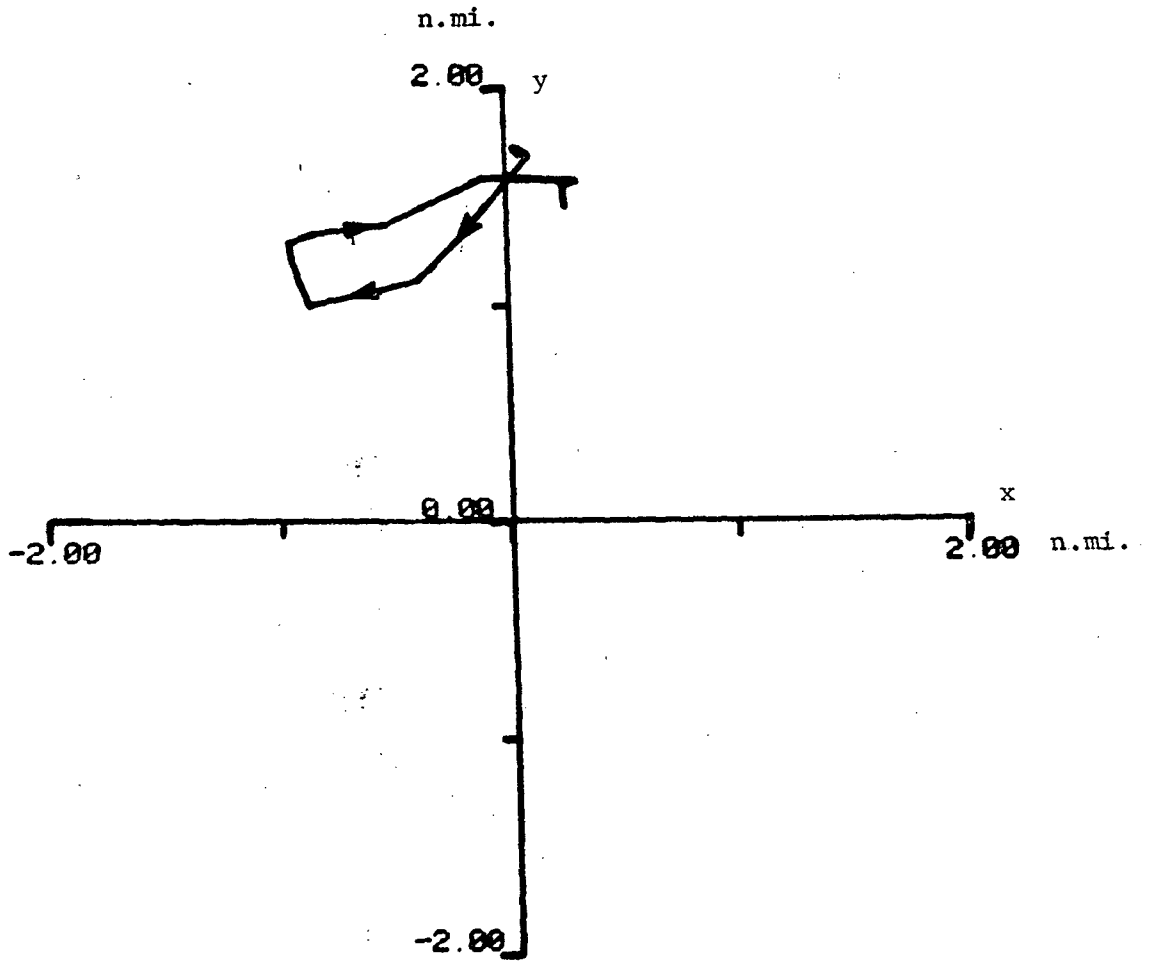


Figure 5-10. Uncorrected OMEGA Derived Position Error During Turn Over Harcum VOR Site Showing Lag Effects During Flight ONR-3 at 13.6 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 9 secs.

the eastern part of Virginia. Figure 5-11 shows a radar track of the flight. To see the effect of Δt in the position error analysis, Figure 5-12 shows position error using 10.2 Hz data for four different Δt values. The effect of OMEGA position prediction is evident for the larger values of Δt. For $\Delta t = -4$ secs the uncorrected position errors are very well centered. Little variation is observed in the general direction of the error. The spread due to time translation of the OMEGA position estimate is again apparent in Figure 5-13. It can be noted that there is a large mean error in the uncorrected 13.6 kHz position data as well. Figure 5-14 shows the 3.4 kHz position error data without corrections. Again the offset is significant and the error variation in the Norway direction is predominant.

5.1.3 Flight ONR-13. Flight ONR-13 took place from Wallops Flight Center on October 12, 1977. This flight involved a triangular course over three VOR sites north of Wallops: Snow Hill, Sea Isle, and Kenton. Figure 5-15 is a reproduced version of the FPS-16 radar track of the aircraft flight around this triangle. Figure 5-16 is a plot of position estimates using the OMEGA receiver data. OMEGA position estimates were derived from NOR-HAW and NDK-TRI LOP measurements at 10.2 kHz. Each LOP fix was converted to a Lambert x-y position and then to the Wallops x-y coordinate system which is plotted in the figure. The OMEGA derived position estimates appear to compare well with the radar track however more rigorous comparison was not possible because the recorded radar position tapes were not available for this flight at the time this report was prepared.

Using the OMEGA data some further analysis was made to determine the effect of truncation in the calculations made within the OMEGA receiver. Figure 5-17 shows cross-track deviation as recorded from the receiver and as calculated post-facto using the recorded position fixes and waypoint data along leg 1 between Snow Hill and Sea Isle. The OMEGA receiver truncation in calculations does impose a bias in the output but does describe the variations adequately. In Figure 5-18 a similar analysis of distance to destination along leg 1 is presented. The effect of truncation is obvious in that the receiver output estimates are consistently too small. The values are meaningful but could be improved upon. In Figures 5-19 and 5-20 the cross-track error and distance to destination values are plotted relative to the desired course along leg 1 for comparison. In Figure 5-19 the receiver output paramenters are used while

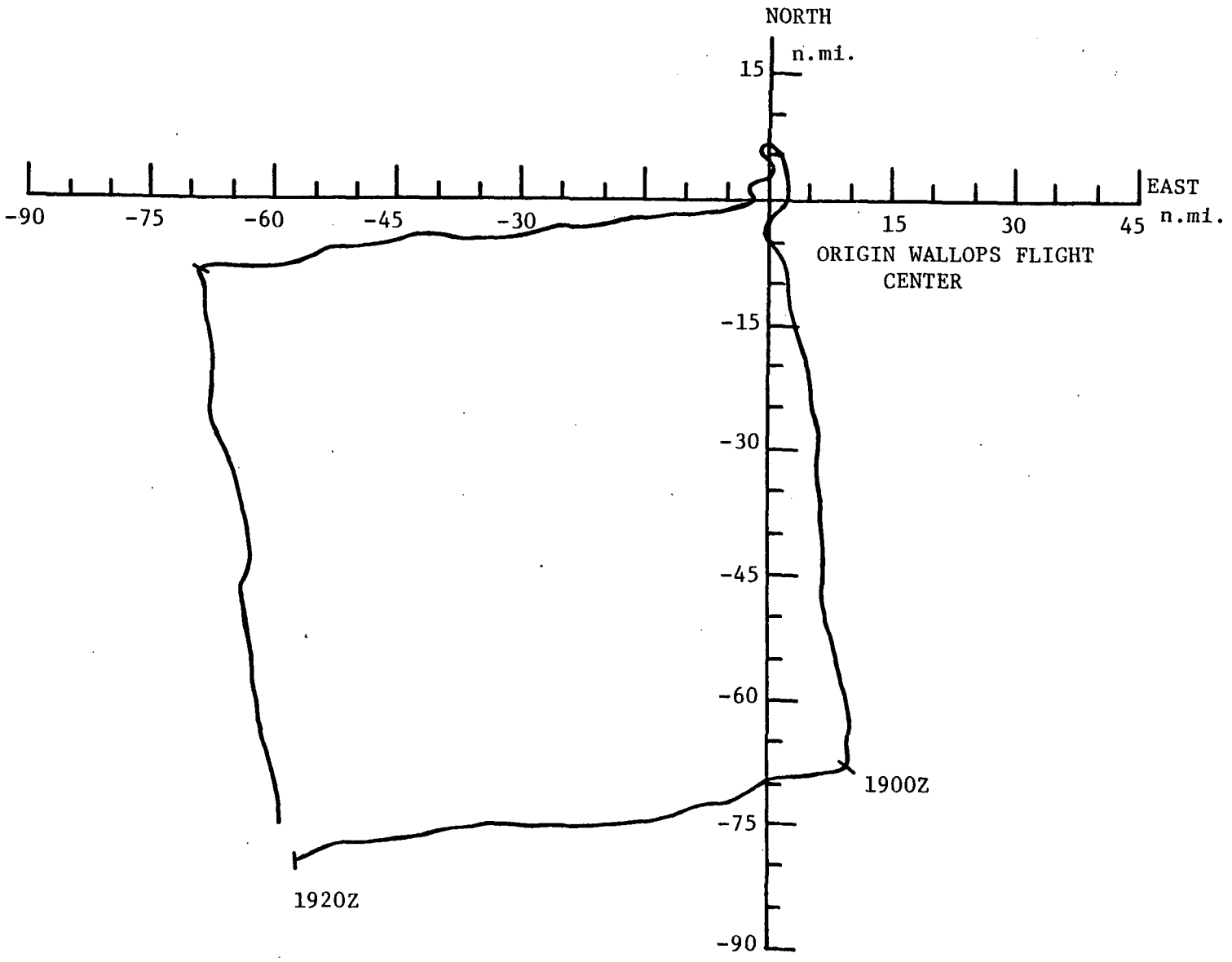
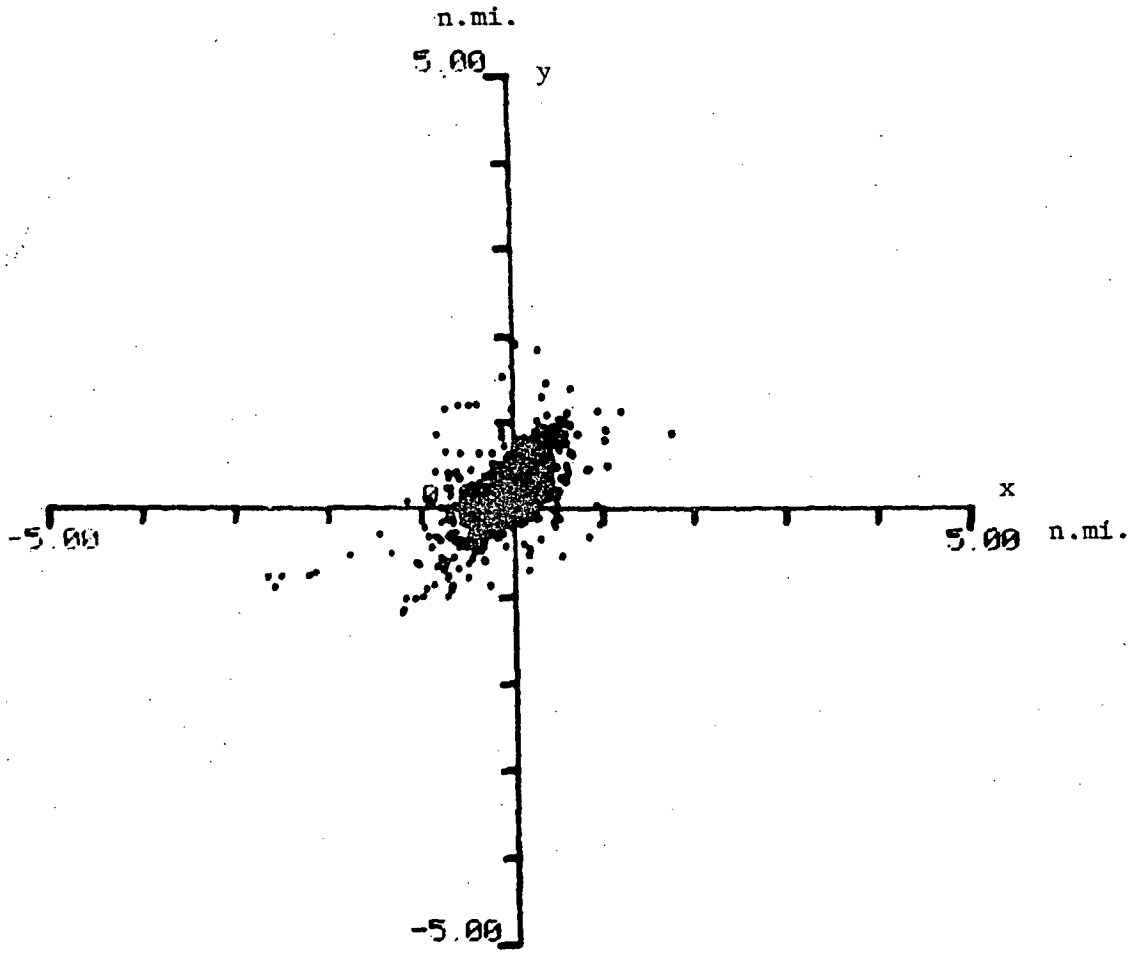
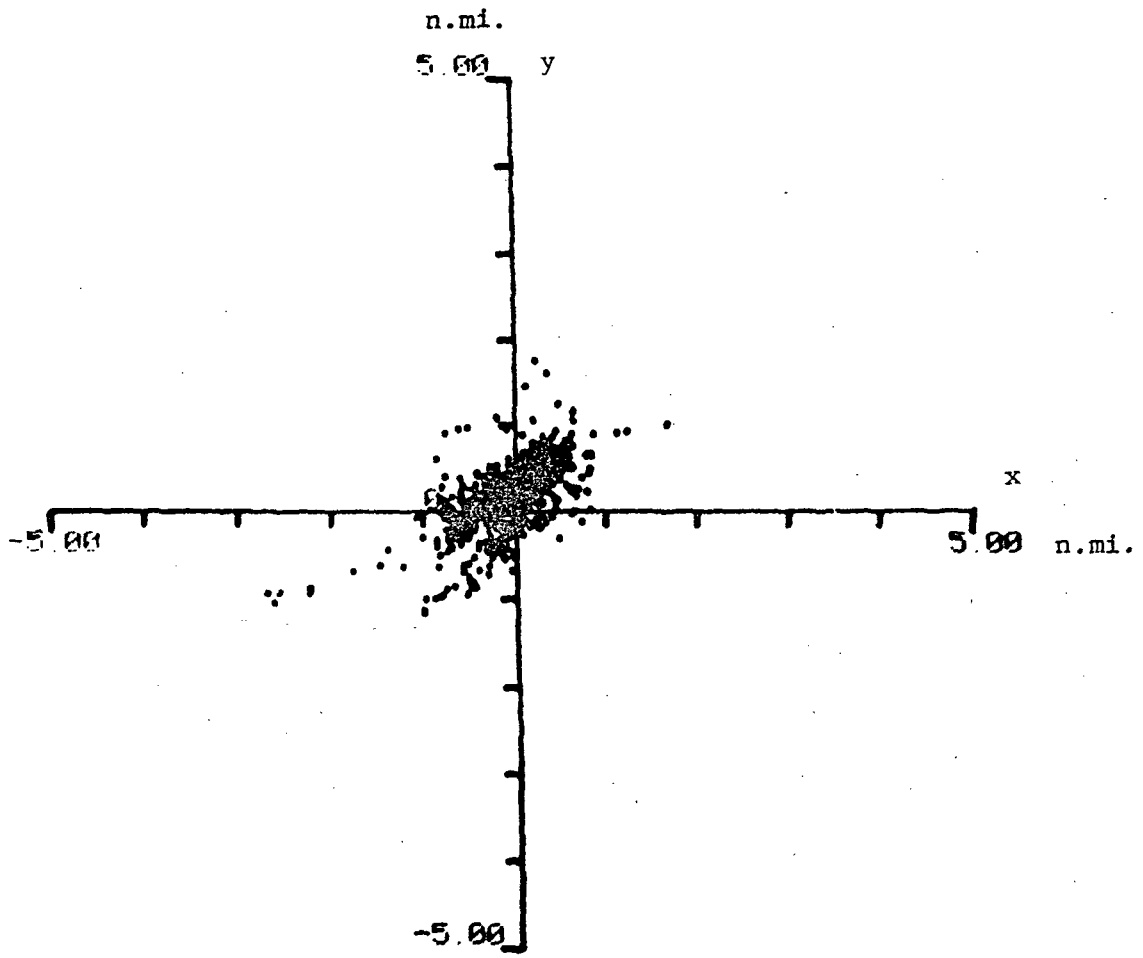


Figure 5-11. Reproduction of Wallops Radar Derived Track for Flight ONR-5.



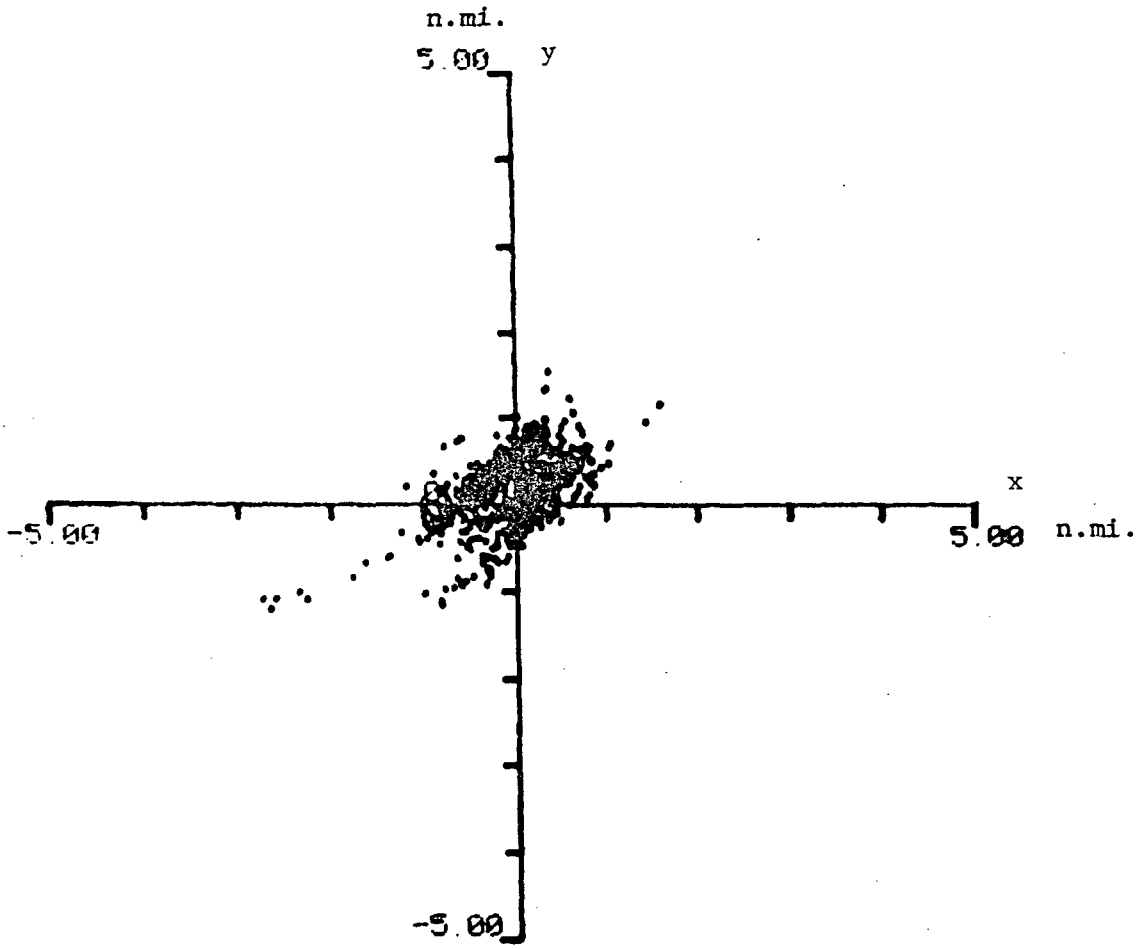
(a)

Figure 5-12. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-5 at 10.2 kHz using LOPs NOR-HAW and TRI-NDK.
(a) Δt = -4, -5 secs, (b) Δt = 0, 1 secs
(c) Δt = 5, 6 secs (d) Δt = 10, 11 secs



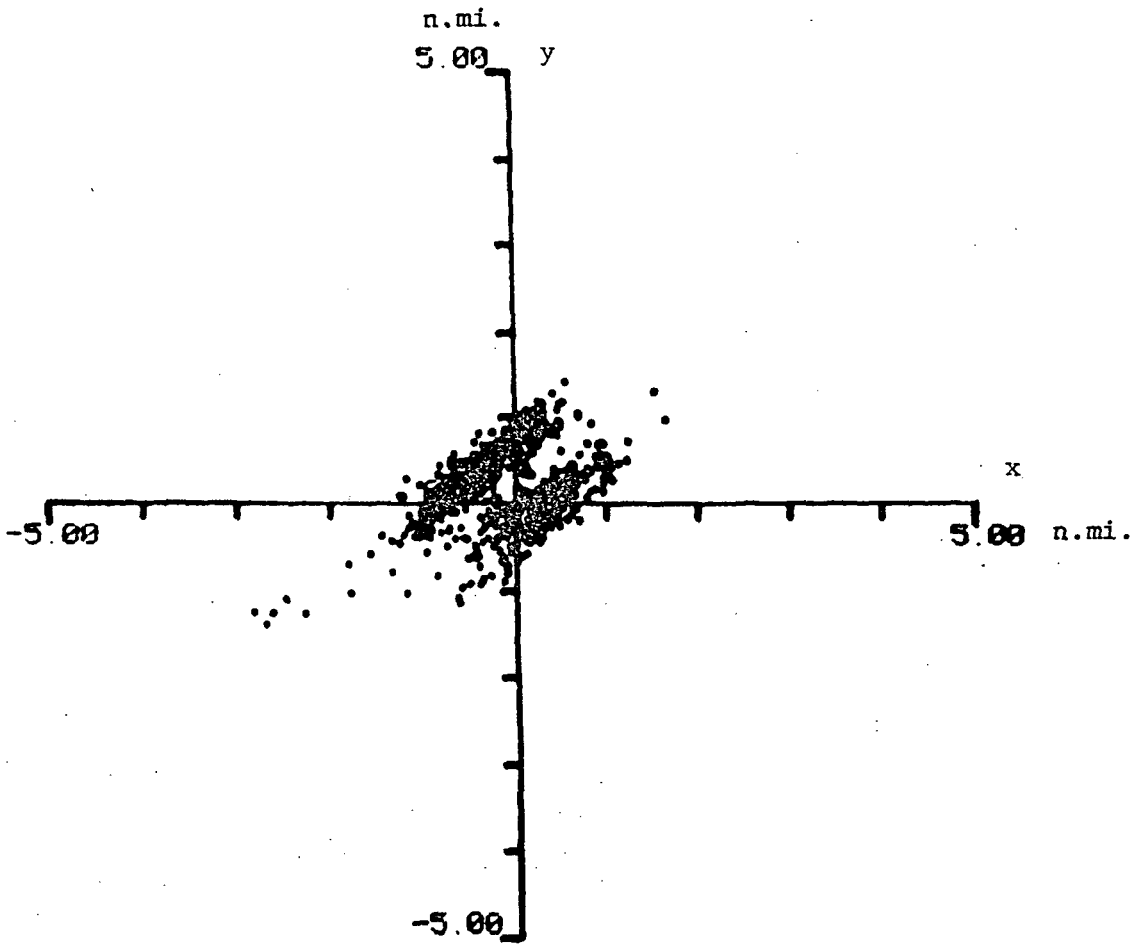
(b)

Figure 5-12. Continued.



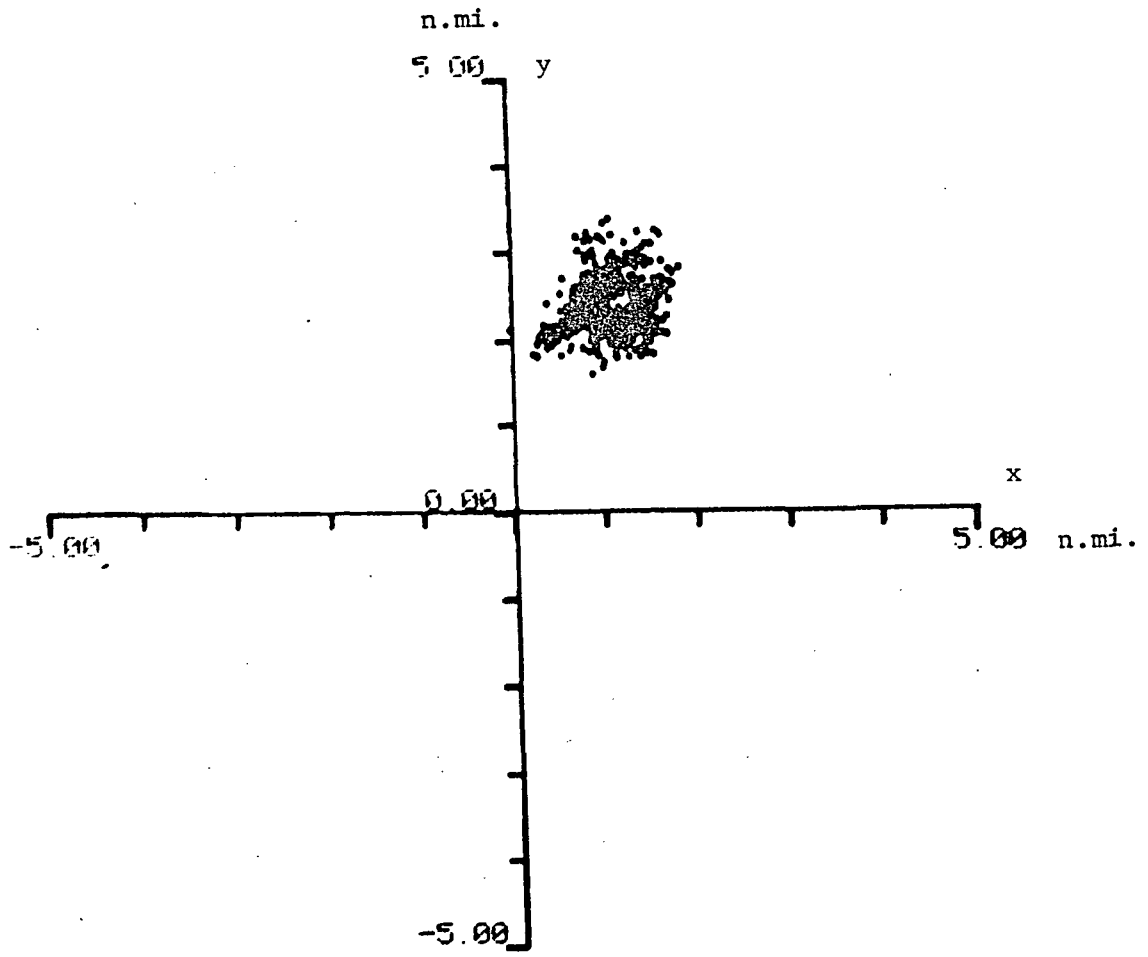
(c)

Figure 5-12. Continued.



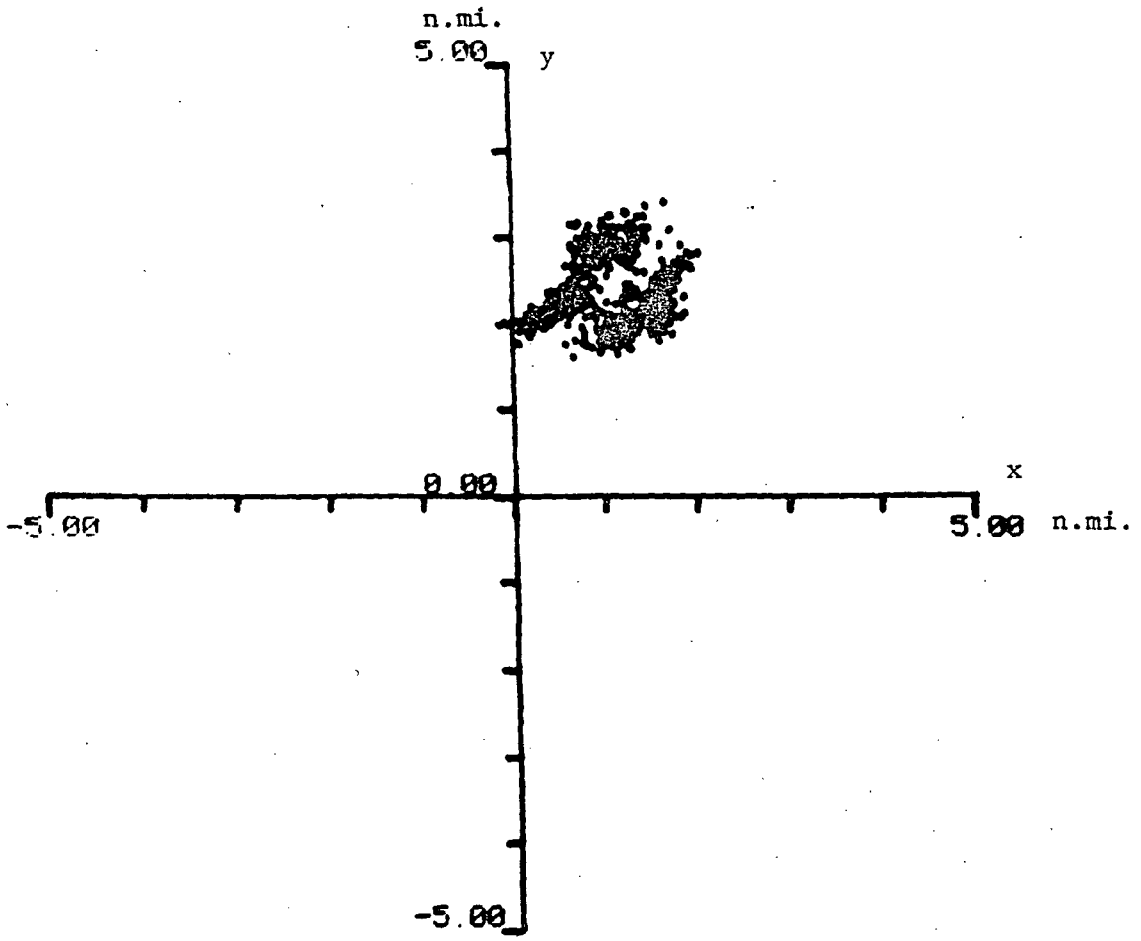
(d)

Figure 5-12. Continued.



(a)

Figure 5-13. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight 0NR-3 at 13.6 kHz using LOPs NOR-HAW and TRI-NDK.
(a) Δt = 5, 6 secs, (b) Δt = 10, 11 secs



(b)

Figure 5-13. Continued.

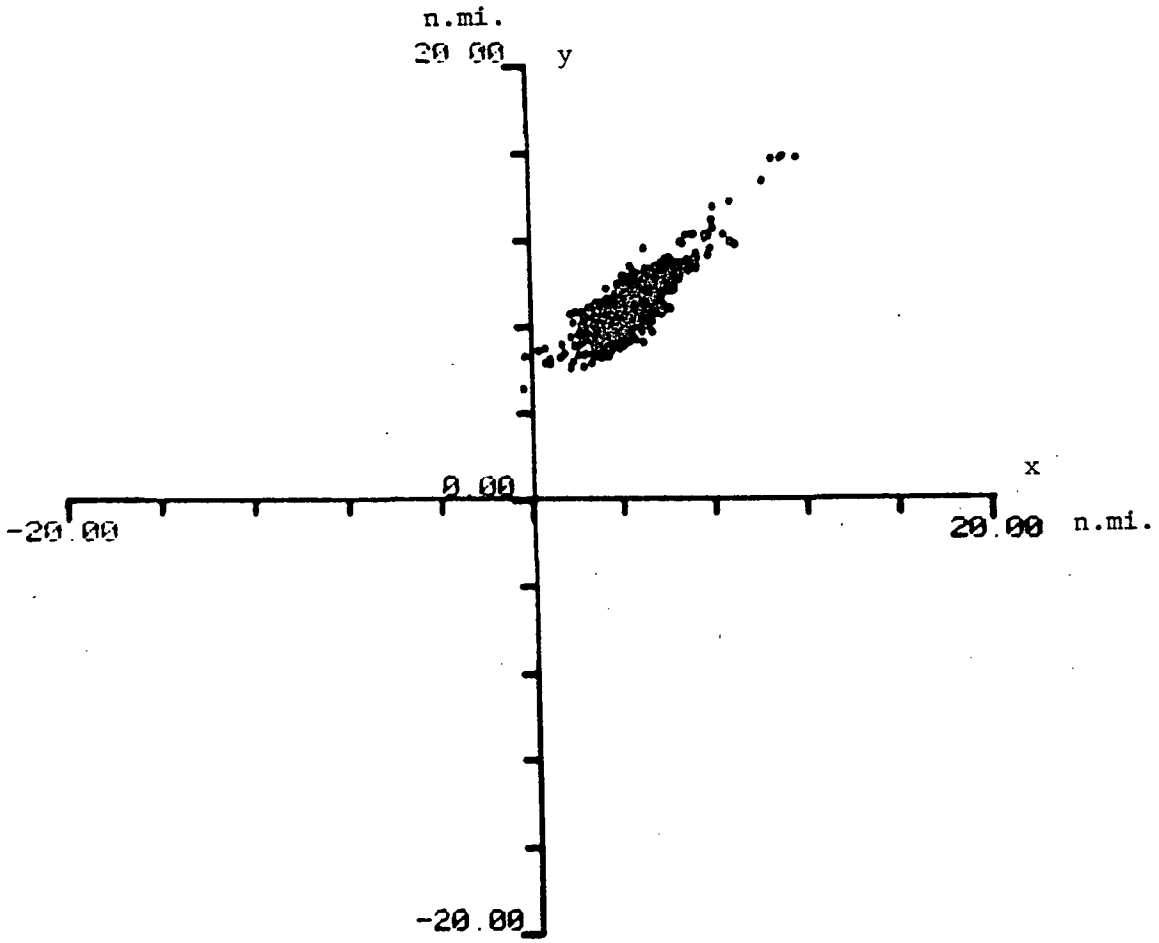


Figure 5-14. Uncorrected OMEGA Derived Position Error Relative to Wallops Radar Fixes for Flight ONR-5 at 3.4 kHz using LOPs NOR-HAW and TRI-NDK. Δt = 5, 6 secs.

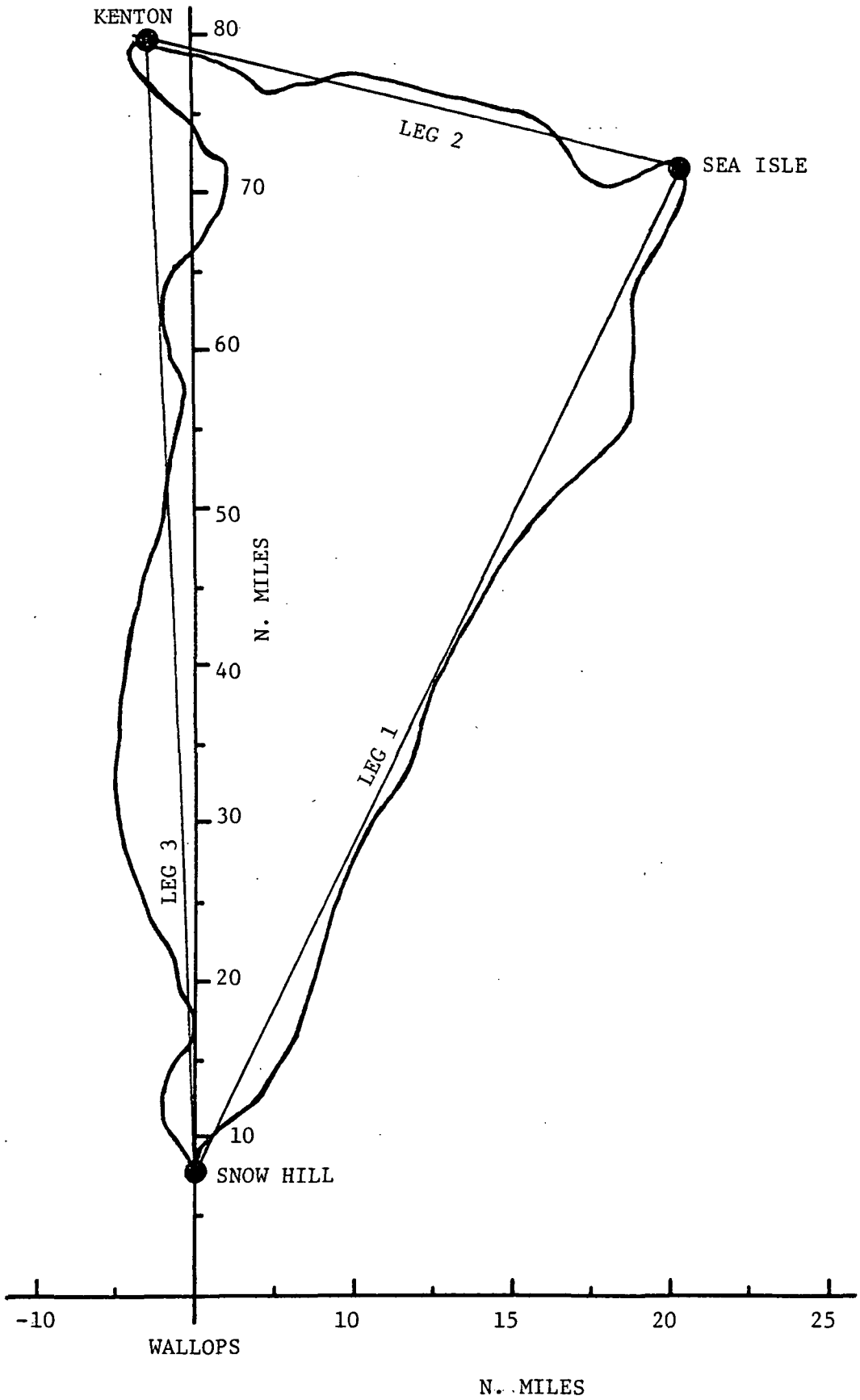


Figure 5-15. Reproduced Radar Track for OMEGA Flight ONR-13

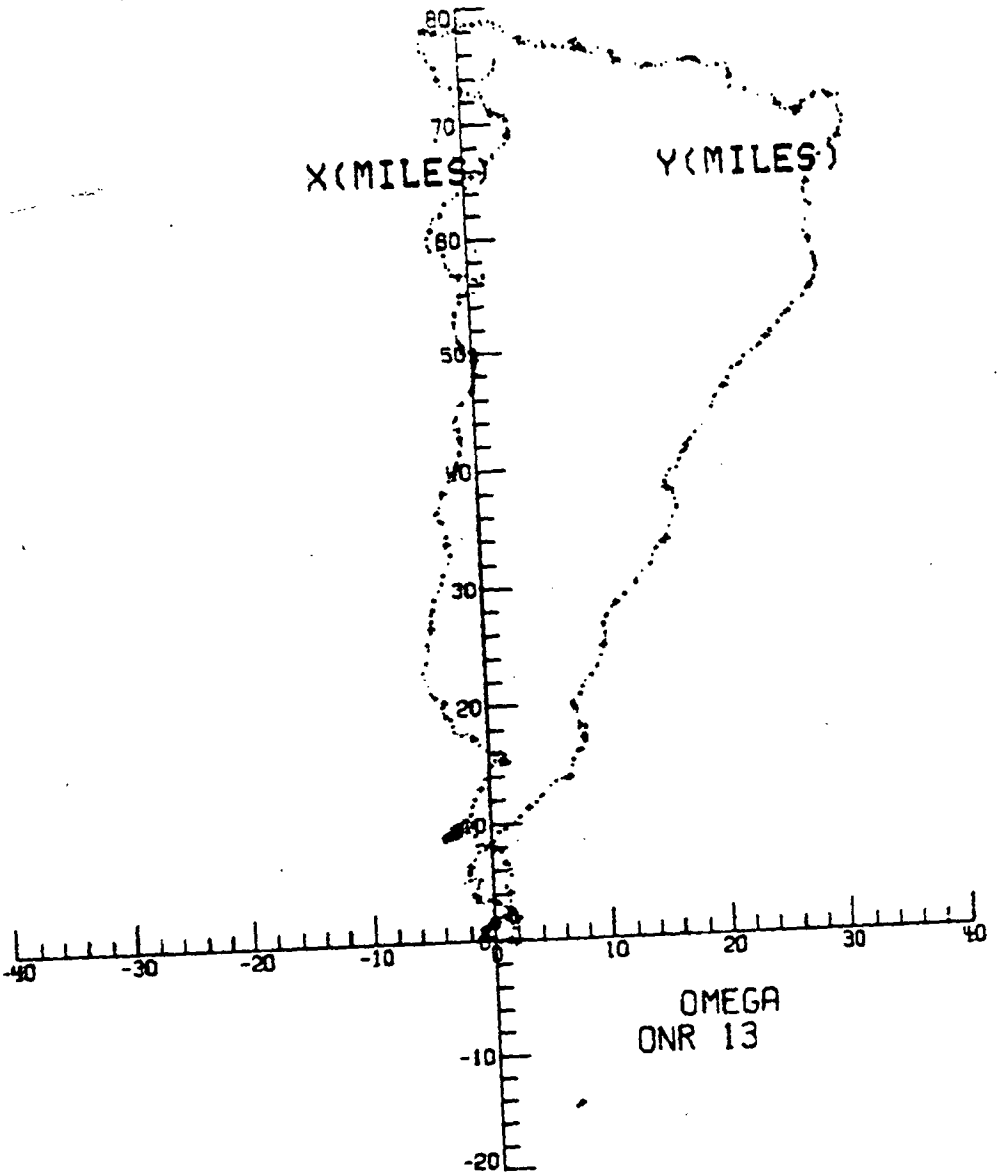


Figure 5-16. Airborne OMEGA Receiver Position Estimates for Flight ONR-13 Using NOR-HAW and NDK-TRI LOPs at 10.2 kHz.

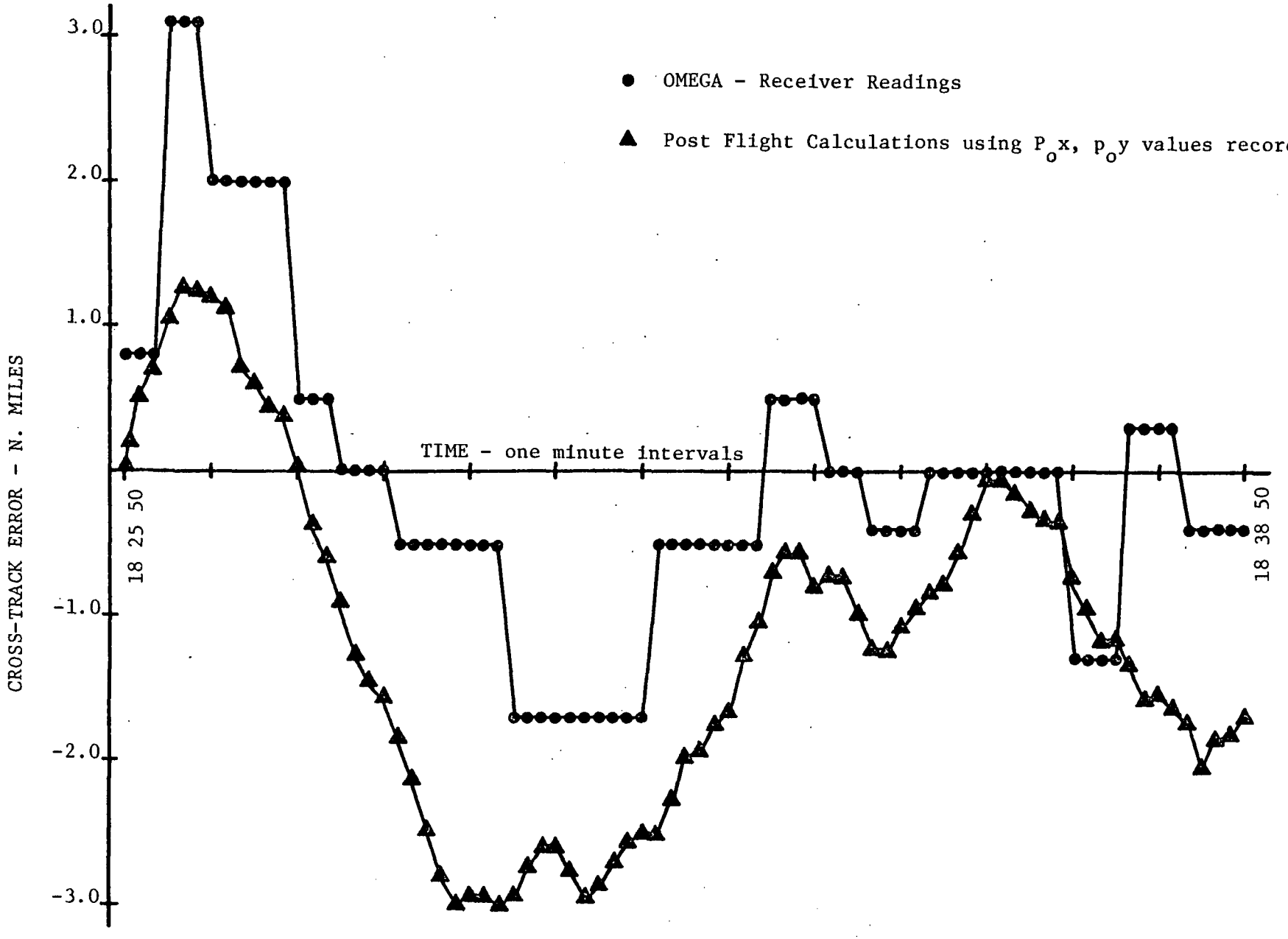


Figure 5-17. Evaluation of OMEGA Receiver Output Cross-track Deviation during LEG 1 of ONR-13

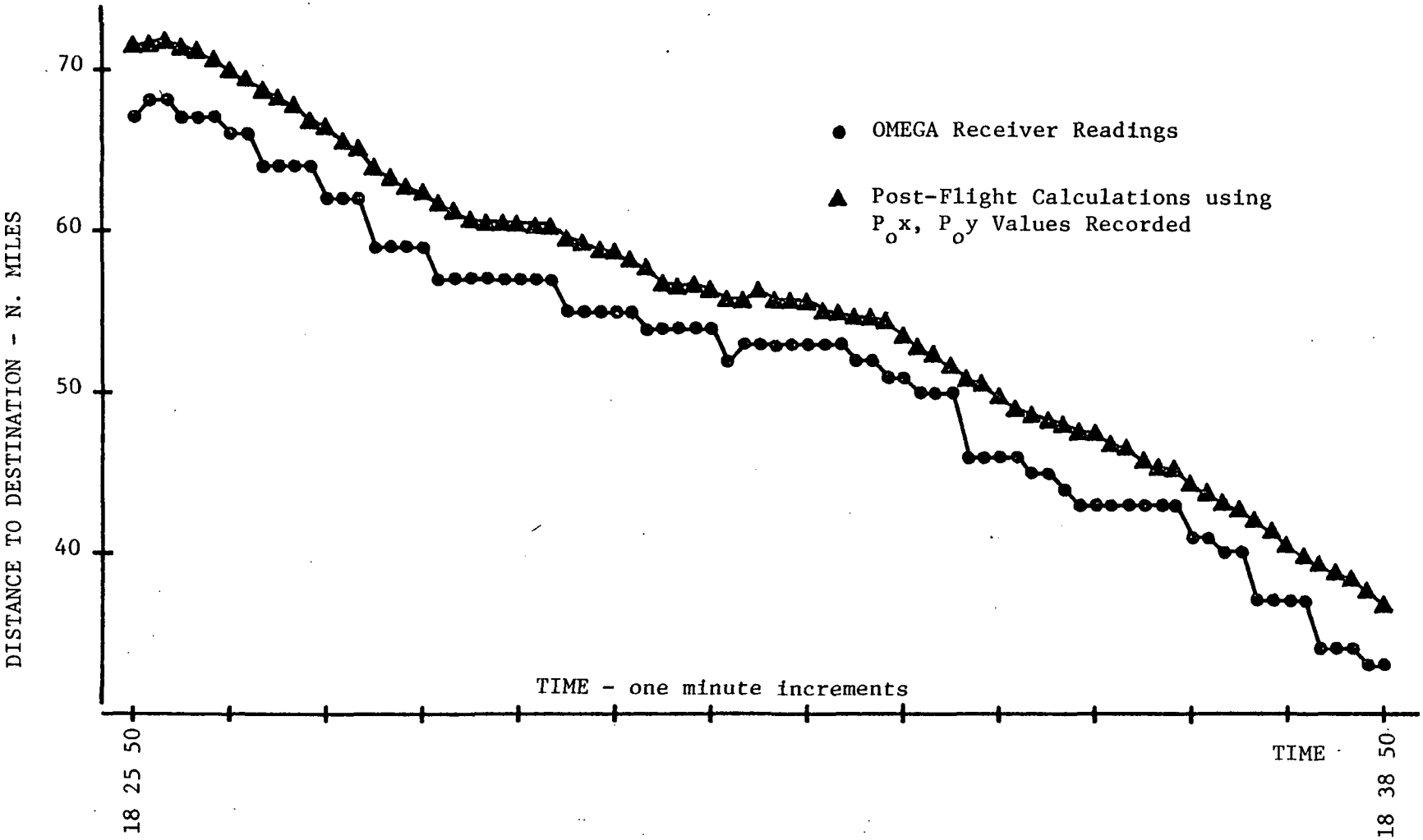


Figure 5-18. Evaluation of OMEGA Receiver Output Distance-to-Destination during LEG 1 of ONR-13

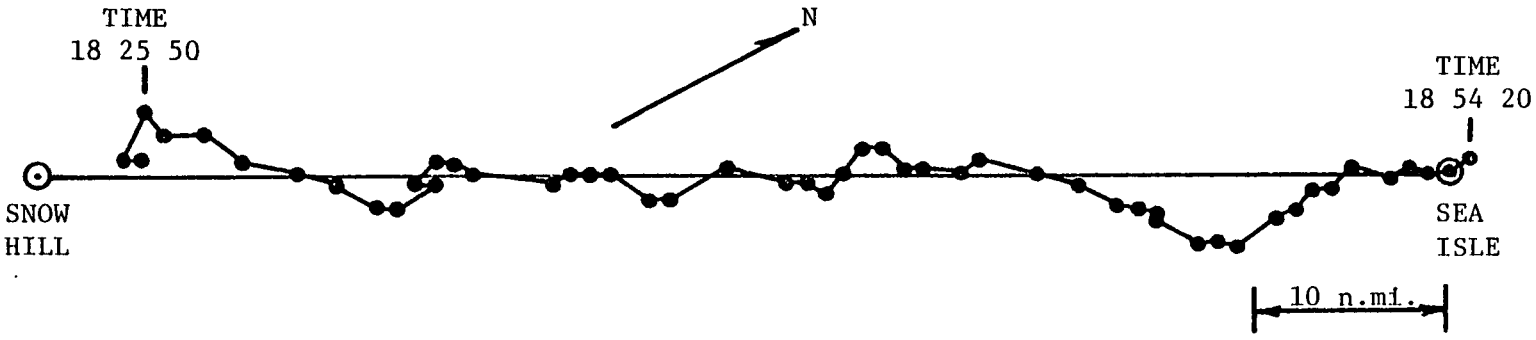


Figure 5-19. OMEGA Determined Position Based On DTD and XTD Readout Values During LEG 1 of ONR-13

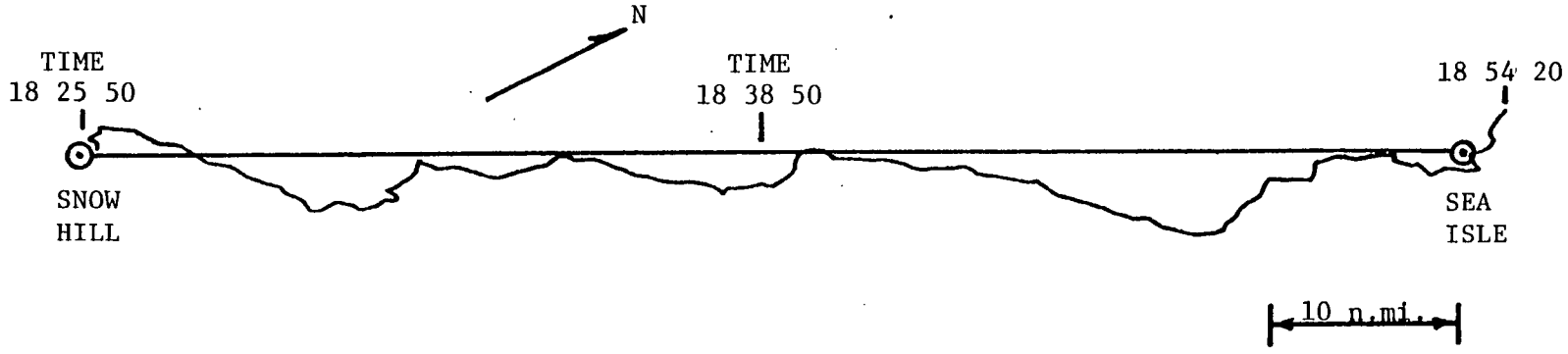


Figure 5-20. Position Relative to Desired Course Using Post-flight Calculated Values of DTD and XTD Based On Recorded NOR-HAW and NDK-TRI LOP Measurements at 10.2 kHz During LEG 1 of ONR-13

in Figure 5-20 the OMEGA LOP fixes are used in post-flight calculations without any significant loss of precision.

In Figure 5-21 the velocity estimates (ground-track) are shown for each of the three flight legs. The noise which appeared on previous flights was smoothed using the new velocity smoothing algorith (see section 3.2.1). Although the velocity does vary over a large range, which can be attibuted to the fact that the aircraft did incur significant velocity changes inflight, the smoothed estimate is an improvement.

5.2 Summary of Test Flights.

Several other flights were made which have not been reported here. The data analyses were not available primarily becuase of delays in getting the recorded radar position data. Four flights during the period October 31 through November 2, 1977 are considered to be the most meaningful. In general the OMEGA derived position error was small enough to make the navigation receiver functional even as an IFR navigation device. On two successive flights from Wallops to Elizabeth City, N.C. on November 2, 1977, IFR conditions were encountered. The pilot was compelled by FAA regulations to fly according to VOR readings. However, using the OMEGA receiver and waypoints along the route the flight was flown according to the OMEGA receiver while continuously monitoring the VOR associated instruments. The OMEGA receiver proved its value and using distance to destination and heading to waypoint readouts the pilot was able to navigate to the runway quite satisfactorily. Limited use was made of differential corrections transferred via radio link from a ground-based receiver at Wallops.

The OMEGA navigation receiver proved to be valuable for airborne use even with the limitations that are inherent because of the microcomputer which was used. Even without the complete rigorous analysis results the following conclusions can be made: (1) OMEGA can serve as a viable navigation means for aviation, (2) Implementation of a low-cost navigation receiver for general aviation use is in fact feasible and should be encouraged, (3) Additional work is desirable to update the microcomputer which was used to reduce the cost further while allowing for increased navigation capability.

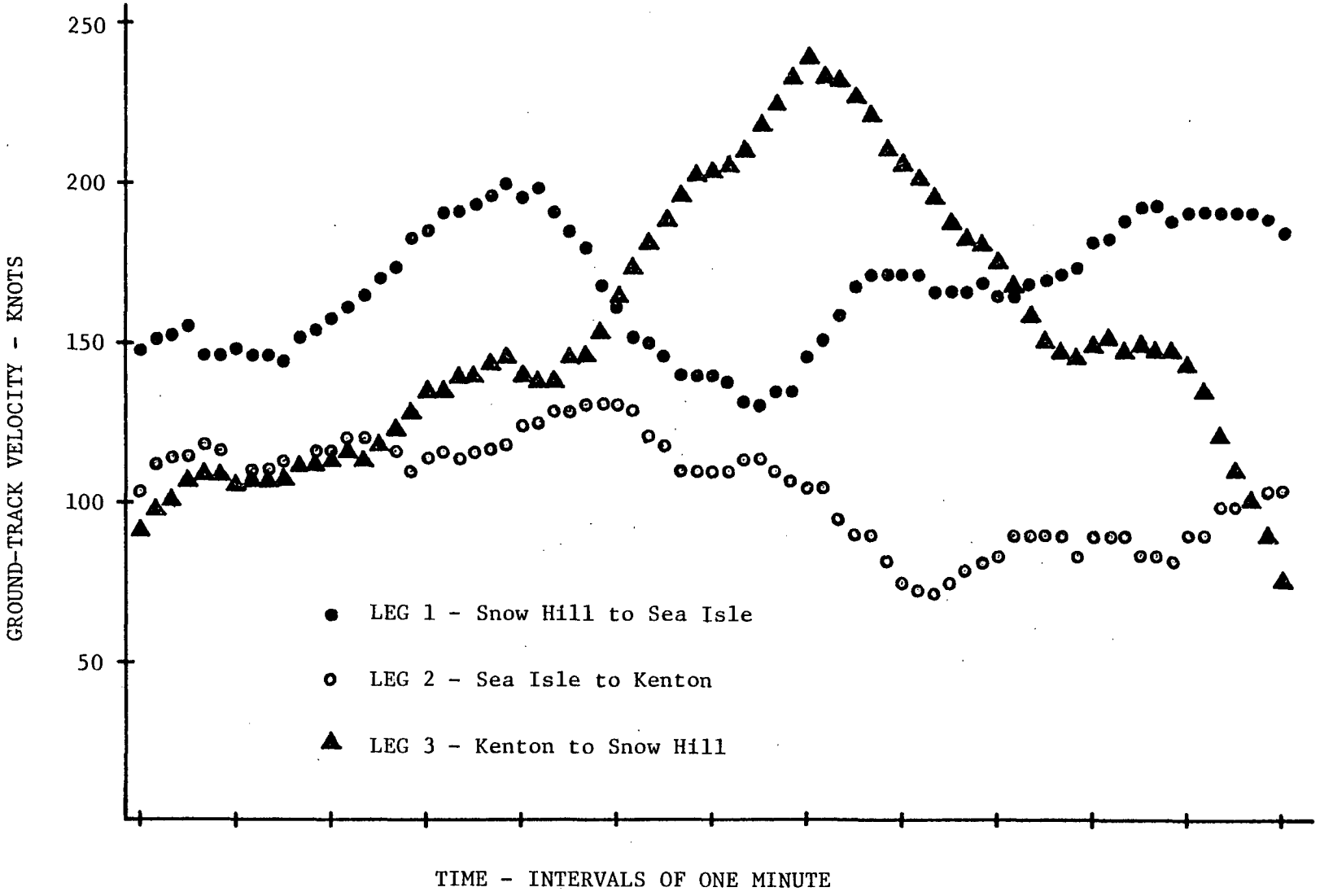


Figure 5-21. Representative Twelve Minute Segments of OMEGA Receiver Velocity Estimates From Each Leg of Flight ONR-13.

"Page missing from available version"

page 120

6.0 SUMMARY AND CONCLUSIONS

The primary objective of this contract has been to provide supportive studies to a Langley Research Center program for evaluation of the performance capabilities of the OMEGA Navigation System for use by civil aviation. One major emphasis has been related to the development of a feasibility model of an OMEGA navigation receiver suitable for general aviation use, which employs a simple microprocessor. In developing the software for this receiver, support effort involved with the use of the microprocessor cross-assembler on the CDC-6600 computer and development of a simulator has provided valuable and necessary tests. With these, software development of the receiver algorithms has proceeded independent of hardware development. The primary effort of RTI personnel in the development of software has been related to navigation algorithms. This has included development and evaluation of methods to enable a receiver to synchronize to the OMEGA broadcast format, derive information from the OMEGA signals, and generate useful and accurate positional type information to the navigator.

The propagation model development work has been directed towards providing more accurate methods for generation of propagation prediction corrections (PPC) for OMEGA navigation use. This coupled with the development of software to generate OMEGA LOP's directly onto topographical maps provides a capability to provide charts to the navigator, which can yield acceptable navigation error without the use of additional corrections, particularly when operating in a difference frequency mode.

Another effort has involved developing a set of standard PPC's for data gathered during the ground-based experimental program. These have been based on corrections obtained from the Hydrographic Center (Defense Mapping Agency) which publishes standard sets of corrections for OMEGA users. Techniques were investigated to use these corrections to provide estimates of phase velocity for use in developing navigation charts which account for PPC's.

Future efforts should involve continuation of the effort in developing navigation algorithms for use with the microprocessor based OMEGA receiver.

This should include investigation of accuracies of the calculations for the particular hardware/software configuration which results from this development effort. Possible use of a more recently developed microprocessor with greater computational capability and higher hardware packaging densities should be considered. The impact of a microprocessor with interrupt capability and a bigger word size can offer significant advantages. Investigation should also evaluate the feasibility of incorporating algorithms to allow input and output data to be in units of latitude/longitude to eliminate the requirement of the navigator to use OMEGA coordinates. Use of velocity aiding techniques to improve the reliability of the navigation receiver outputs should also be investigated.

APPENDICES

APPENDIX A

AIRBORNE RECEIVER DIGITAL PHASE-LOCKED LOOP

The microprocessor based OMEGA receiver being investigated by NASA-LRC employs a digital phase-locked loop (DPLL). Figure A-1 depicts this DPLL in block diagram form. The loop filter gain values K_1 and K_3 are restricted

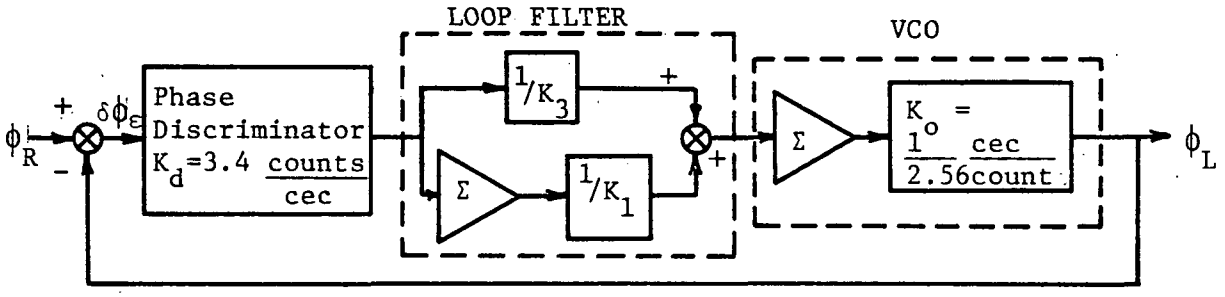


Figure A-1. Digital Phase-Locked Loop Functional Block Diagram

to be integer powers of 2 due to convenience in the software representation of this portion of the DPLL. With K_0 and K_d as indicated in Figure A-1, the loop response to various inputs, $\phi_R(t)$, for a range of possible values for K_1 and K_3 is presented in this appendix.

The representation of the DPLL as given in Figure A-1 can be analyzed as an analog loop by replacing the "Σ" functions with the transfer function representation of an integrator, $\frac{1}{10S}$ where the "10" multiplier is required since ϕ_R is received at 10 second intervals. In the analog situation the overall transfer function becomes

$$\frac{\phi_L(S)}{\phi_R(S)} = H(S) = \frac{\frac{.1328}{K_3}\left(S + \frac{K_3}{10K_1}\right)}{S^2 + \frac{.1328}{K_3}S + \frac{.01328}{K_1}} \qquad \text{(A-1)}$$

The denominator is of the general form $(S^2+2\delta\omega_n S+\omega_n^2)$ so that from (A-1) the natural frequency is

$$\omega_n = \left[\frac{.01328}{K_1}\right]^{1/2}$$

and the "damping factor" is

$$\delta = \frac{.576\sqrt{K_1}}{K_3}$$

(see ref. 15). From the inverse transform of (A-1) the impulse response can be expressed as

$$h(t) = \frac{\delta\left(\omega_2 + \frac{\omega_n}{2\delta}\right)}{\sqrt{\delta^2-1}} e^{\omega_2 t} - \frac{\delta\left(\omega_1 + \frac{\omega_n}{2\delta}\right)}{\sqrt{\delta^2-1}} e^{\omega_1 t} \qquad \text{(A-2)}$$

where

$$\omega_1, \omega_2 = -\delta\omega_n \pm \omega_n\sqrt{\delta^2-1}$$

For the "underdamped" system, $\delta < 1$, and the only real part of the exponent of h(t) is $-\delta\omega_n t$ so that the "time constant" τ is

$$\tau = \frac{1}{\delta\omega_n} = \frac{K_3}{.0664} \qquad \text{(A-3)}$$

which is the time constant associated with the envelope of the response. There are imaginary exponents which contribute to oscillatory response. For the "overdamped" situation, $\delta > 1$, the entire exponent in (A-2) is real and the time constant is a more complex function than that of (A-3). As δ become large the time constant approaches

$$\tau \approx \frac{1}{2\delta\omega_n} = \frac{K_3}{.1328}$$

which is the time constant of the first order loop i.e., that time constant where the integrator channel of the loop filter is not present. It should be noted that overshoot is always present with this loop in digital form because of quantization error and lack of integrator "leakage" inherent to the digital implementation. Even so, with the "overdamped" situation ($\delta > 1$), the overshoot is reduced from the "underdamped" situation which is significant for the application to be considered here. The time constant is significant from the standpoint of considering "settling time," i.e., that time required for the loop to respond to received phase changes and for $(\phi_R - \phi_L) \rightarrow \varepsilon$ for some chosen ε.

In Figures A-2 through A-4 the step response is shown for a range of values of K_1 for three different values of K_3. Note that overshoot decreases with increasing K_1 and settling time first increases and then decreases as K_3 is increased. Figure A-5 summaries these step responses where steady state is defined as the point at which $\phi_R - \phi_L$ remains less that 1 cec.

Next consider the situation when the DPLL receiver is moving at a velocity such that the received phase is changing at a rate $\dot{\phi}_i$. Then assuming that the receiver is moving perpendicularly across an OMEGA LOP at 10.2 kHz (~16 n.mi./100 cecs), the minimum velocity which corresponds to this phase rate is

$$\text{VEL} = \dot{\phi}_i \frac{3600 \text{ sec/hr}}{100 \text{ cecs/lane}} \cdot 16 \text{ n.mi./lane}$$

or

$$\text{VEL} = 576 \, \dot{\phi}_i$$

where $\dot{\phi}_i$ is in cecs/sec. Table A-1 provides some sample values.

With the DPLL receiver moving in a circular pattern the received phase is

$$\phi_R(t) = A \sin(\omega t)$$

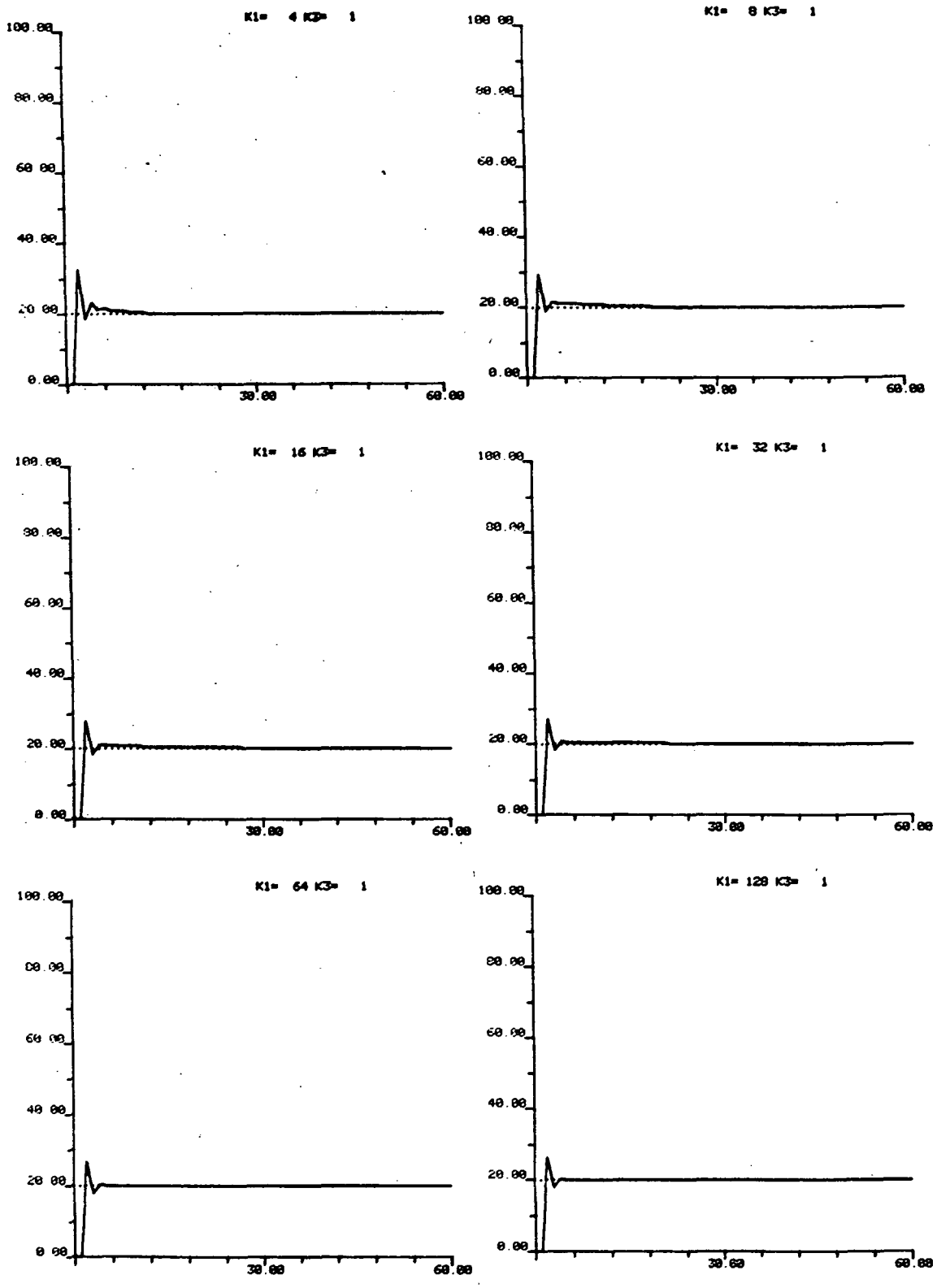


Figure A-2. Step Response of Digital Phase-Lock Loop for Various Values of K_1 With $K_3 = 1$.

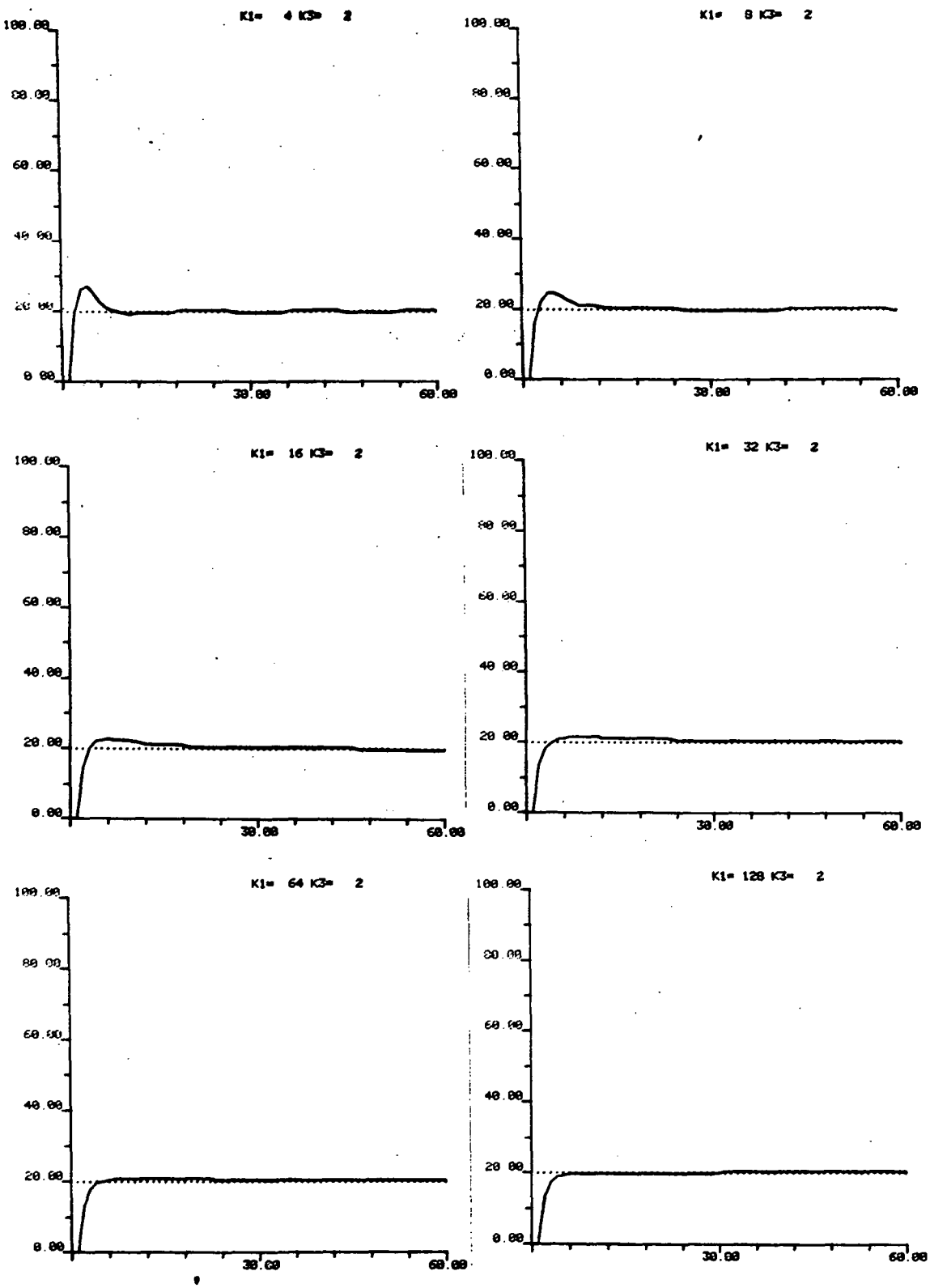


Figure A-3. Step Response of Digital Phase-Lock Loop for Various Values of K_1 With K_3 = 2.

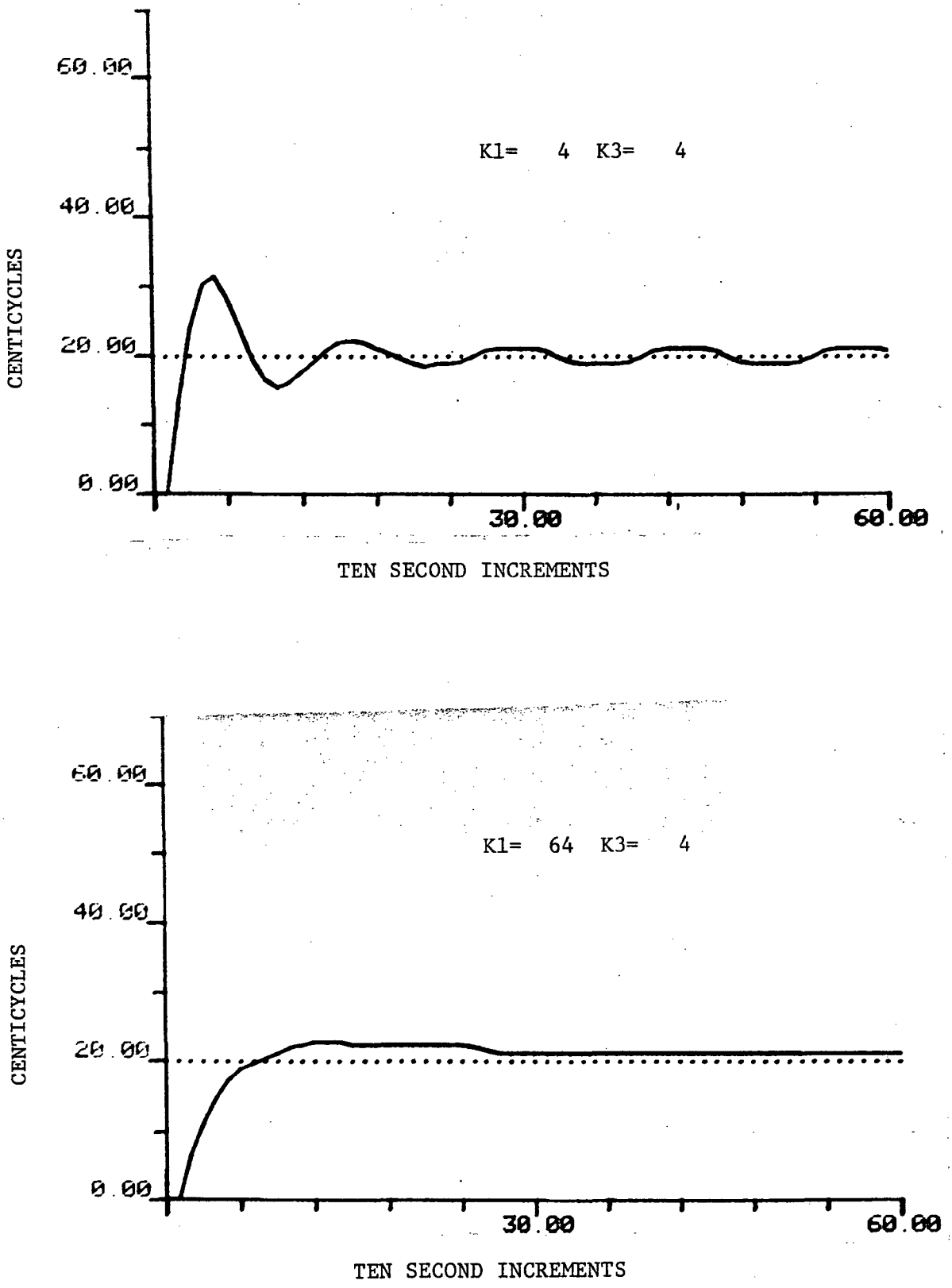


Figure A-4. Step Response of Digital Phase-Lock Loop for Two Values of K_1 With $K_3 = 4$.

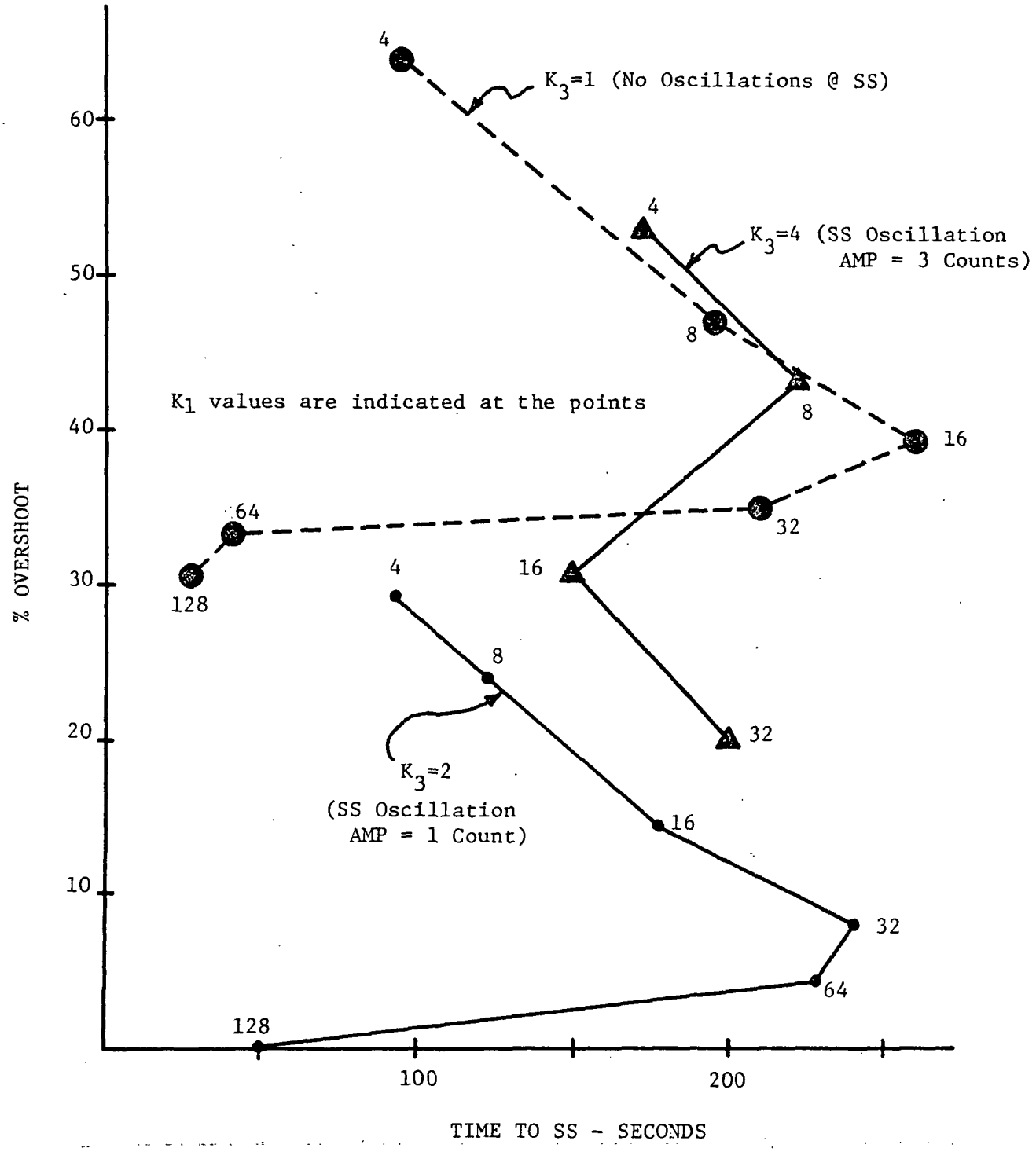


Figure A-5. Summary of Step Responses.

TABLE A-1

MINIMUM VELOCITIES

$\dot{\phi}_i$ cec/sec	VEL(knots)
0.01	5.8
0.1	57.6
1.0	576
2.0	1152

TABLE A-2

VELOCITY IN KNOTS

$\dot{\theta}$ (o/sec)	AMP (cec) / T (sec)	20	40	50
3	120	603	1206	1507
1	360	201	402	503
0.6	600	121	241	302

where $\omega = 2\pi f = \frac{2\pi}{T}$. The turn rate is $\dot{\theta} = \frac{360}{T}$ degrees per second where T is period in secs and the velocity is $VEL = r\dot{\theta} = A\ (\frac{360}{T})$ with $\dot{\theta}$ in radians/sec. Assuming A = AMP of phase variation in cecs and 100 cecs = 16 n.mi., then

$$VEL = \frac{AMP}{100} \cdot 16 \cdot \frac{360}{T} \cdot 3600 \cdot \frac{\pi}{180} \text{ Knots}$$

or

$$VEL = \frac{AMP}{T} \cdot 1152\pi$$

Table A-2 provides a tabulation of velocity for several $\dot{\theta}$ and AMP values.

Figures A-6 through A-8 provide DPLL response for phase ramp inputs with a range of value for K_1 and K_3. Figures A-9 through A-13 provide steady state response of the DPLL receiver for various ramp inputs and K_3 values. Note that K_1 values affect the time to reach the steady state but with the values shown do not affect the steady state response. Figures A-14 through A-15a illustrate DPLL response with an instantaneous velocity change and Figure A-15b illustrates response in a turn of 1°/sec at a velocity of 500 knots with $K_1 = 4$ and $K_3 = 2$. In all of these figures discontinuities occur because phase in cec is reduced to a range (0,100). These discontinuities are shown as nearly vertical straight lines.

Considering a DPLL direct channel gain of 1/2 ($K_3 = 2$) in the loop appears to provide good overall response considering stability within a noisy environment and ability to track a phase during accelerations. Further testing of the DPLL considers only a direct channel gain of 1/2 and various values of integrator channel gain $1/K_1$ (K_1 of 2^n, $n \geq 0$ and integer). In performing this analysis the receiver is simulated with an input phase corresponding to that which would be observed in a vehicle moving at some velocity VEL across an OMEGA lane until the loop is allowed to lock up, making a 180° turn at any selected turn rate and moving back towards the origin phase point at a constant velocity. Figure A-16 indicates a typical spatial trajectory of movement where the value $\dot{\theta}$ is the turn rate and t_1 is the time at which the turn is initiated. Figure A-17

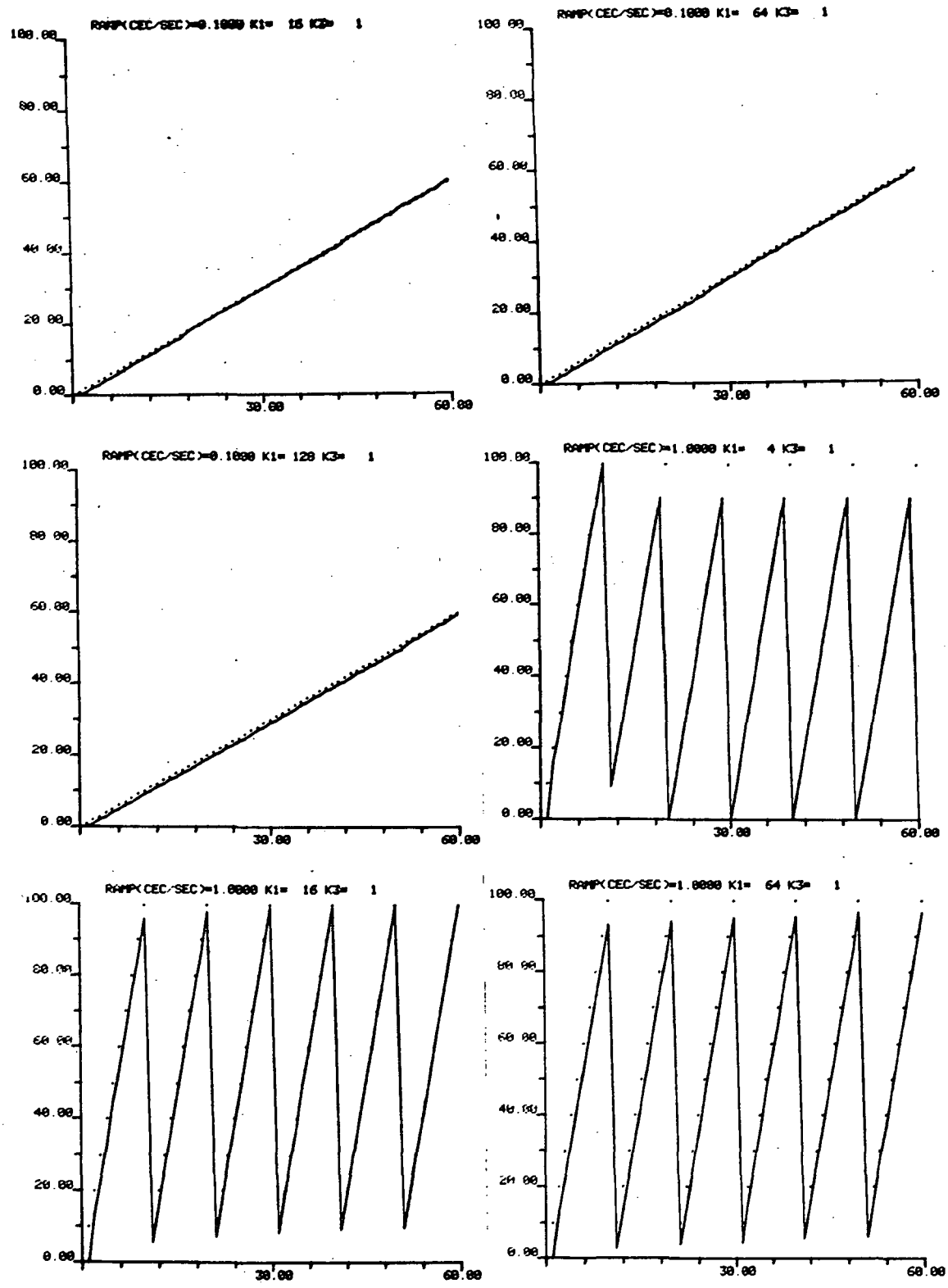


Figure A-6. Ramp Response of Digital Phase-Lock Loop for Various Values of K_1 With K_3 = 1.

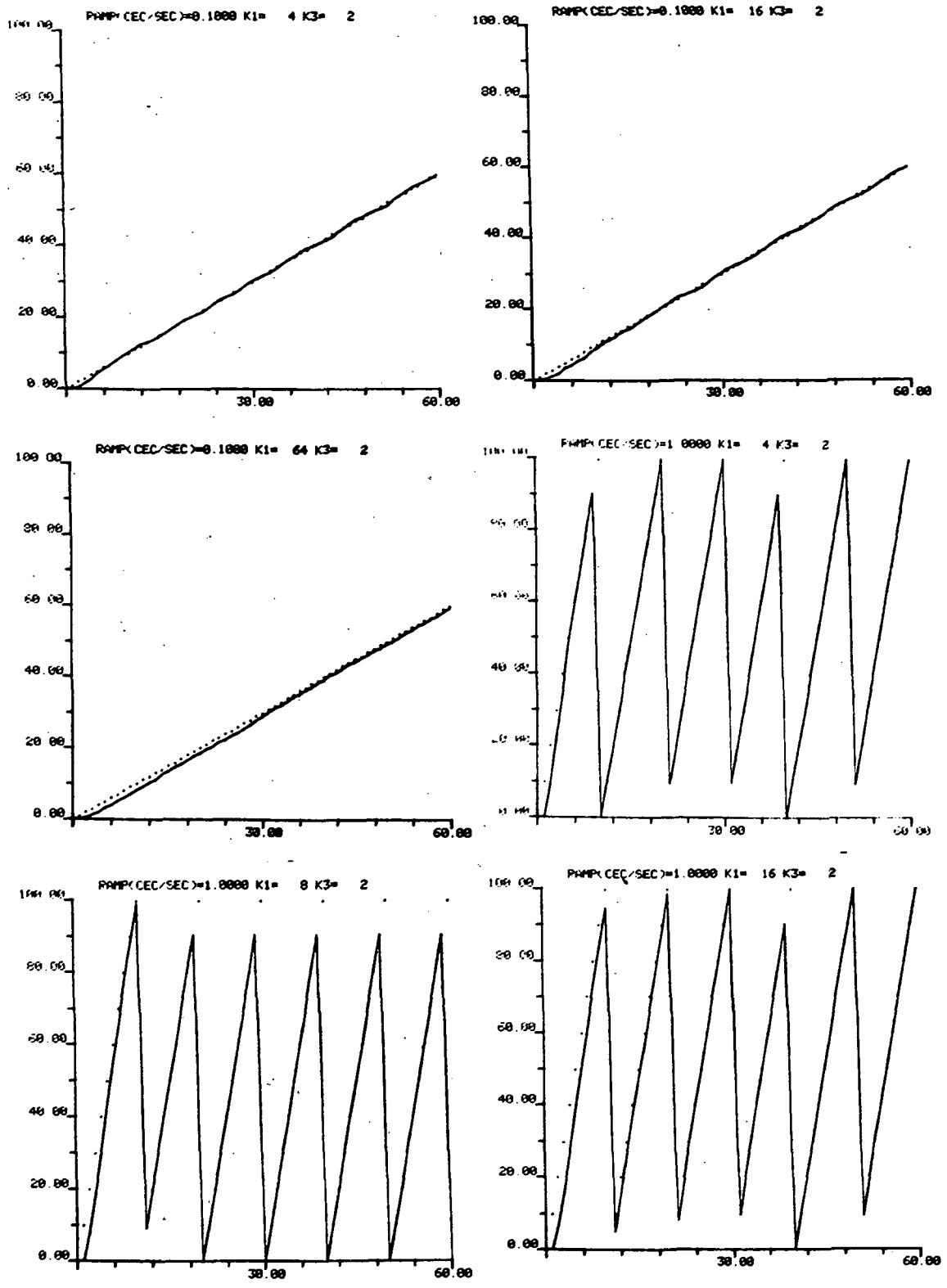


Figure A-7. Ramp Response of Digital Phase-Lock Loop for Various Values of K_1 With $K_3 = 2$.

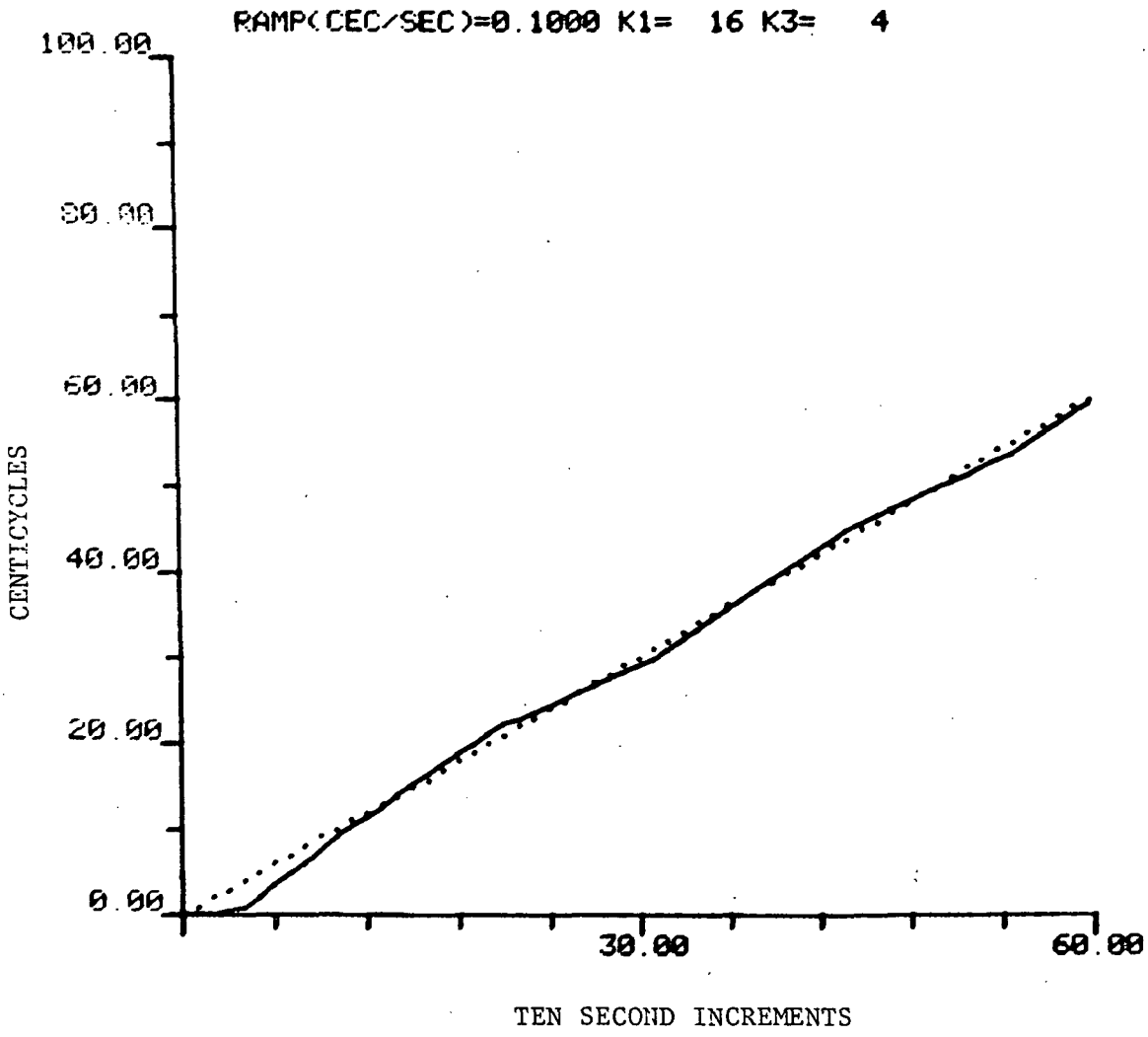


Figure A-8. Ramp Response of Digital Phase-Lock Loops for $K_1 = 16$ and $K_3 = 4$.

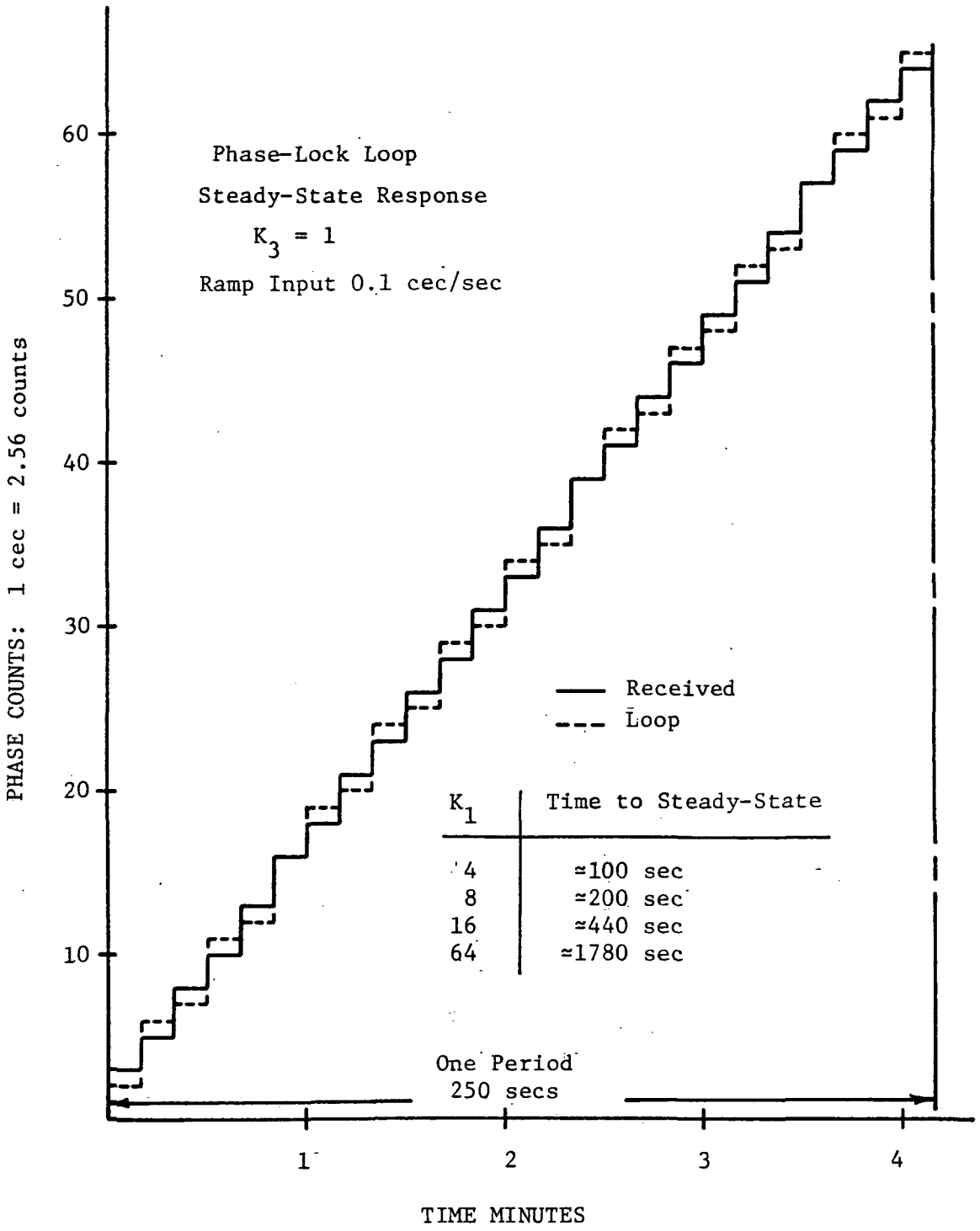


Figure A-9. Steady-State Response of Digital Phase-lock Loop to 0.1 Cec/Sec Ramp Input Over One Period With K_3=1.

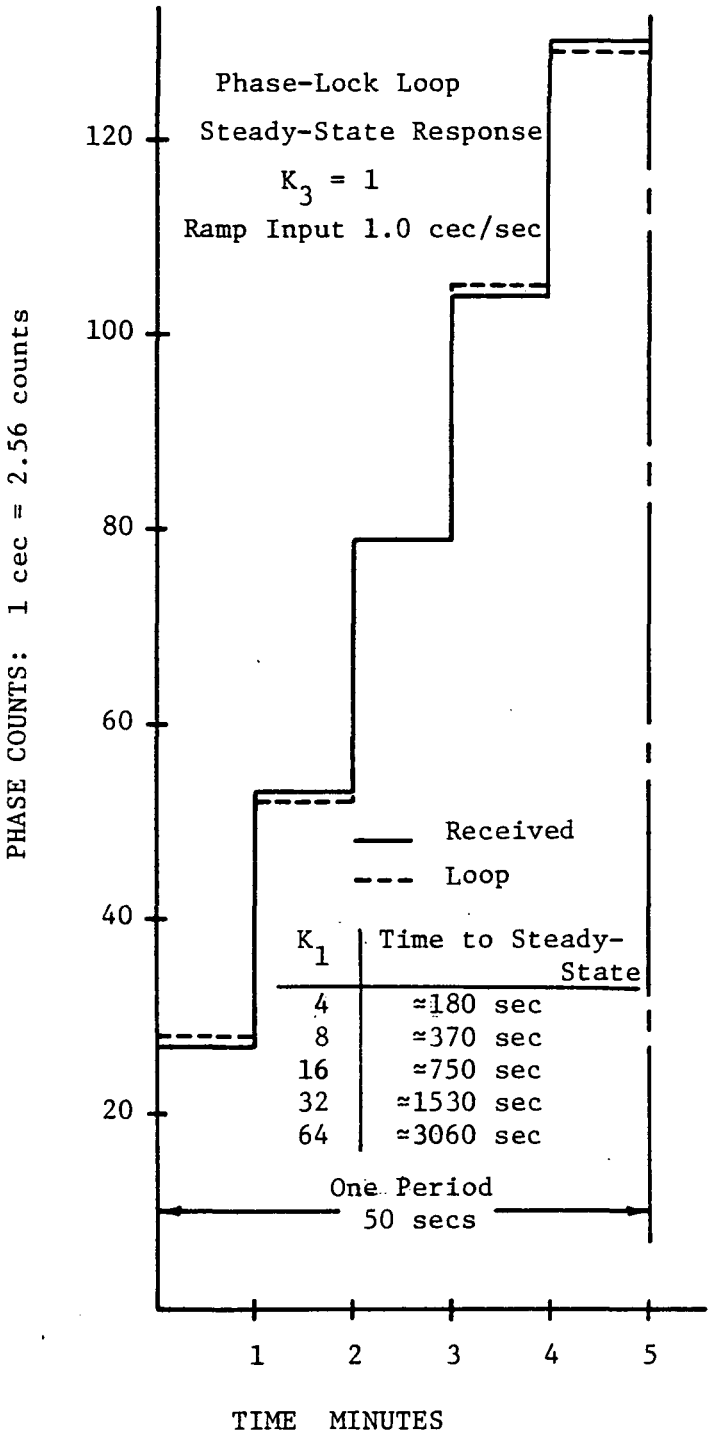


Figure A-10. Steady-State Response of Digital Phase-Lock Loop to 1.0 Cec/Sec Ramp Input Over One Period With K_3 = 1.

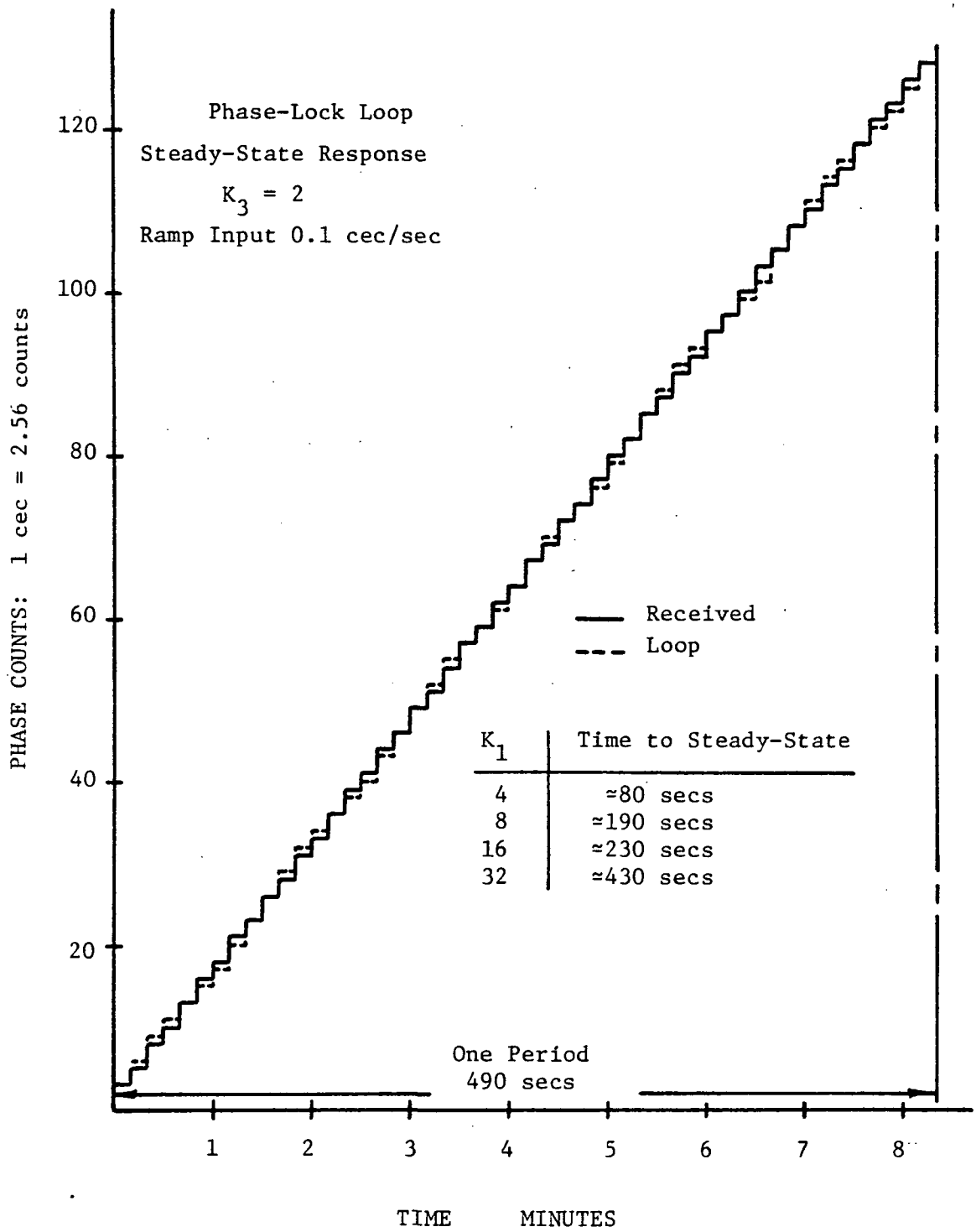


Figure A-11. Steady-State Response of Digital Phase-Lock Loop to 0.1 Cec/Sec Ramp Input Over One Period With $K_3 = 2$.

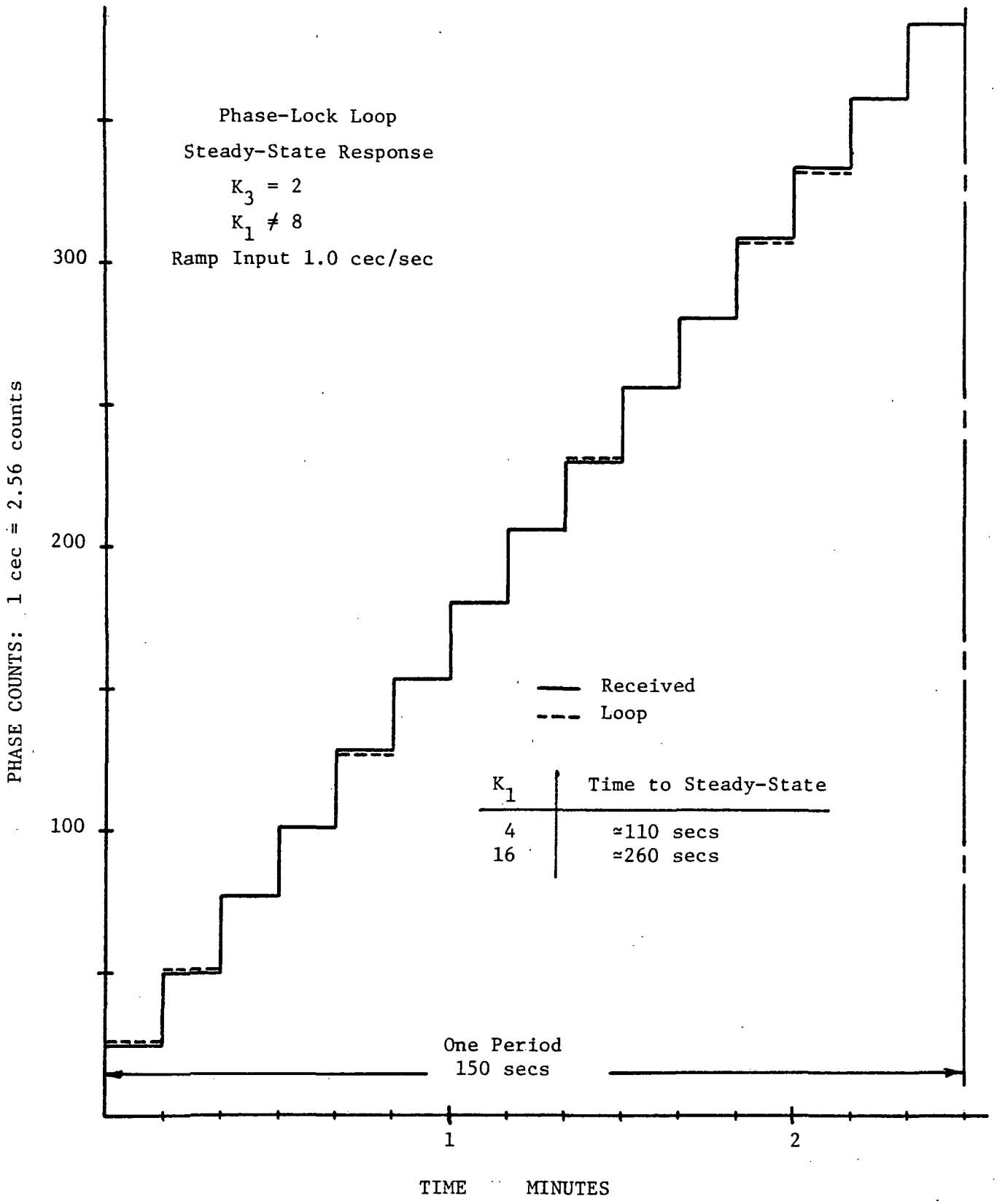


Figure A-12. Steady-State Response of Digital Phase-Lock Loop to 1.0 Cec/Sec Ramp Input Over One Period With $K_3 = 2$ and $K_1 = 4, 16$.

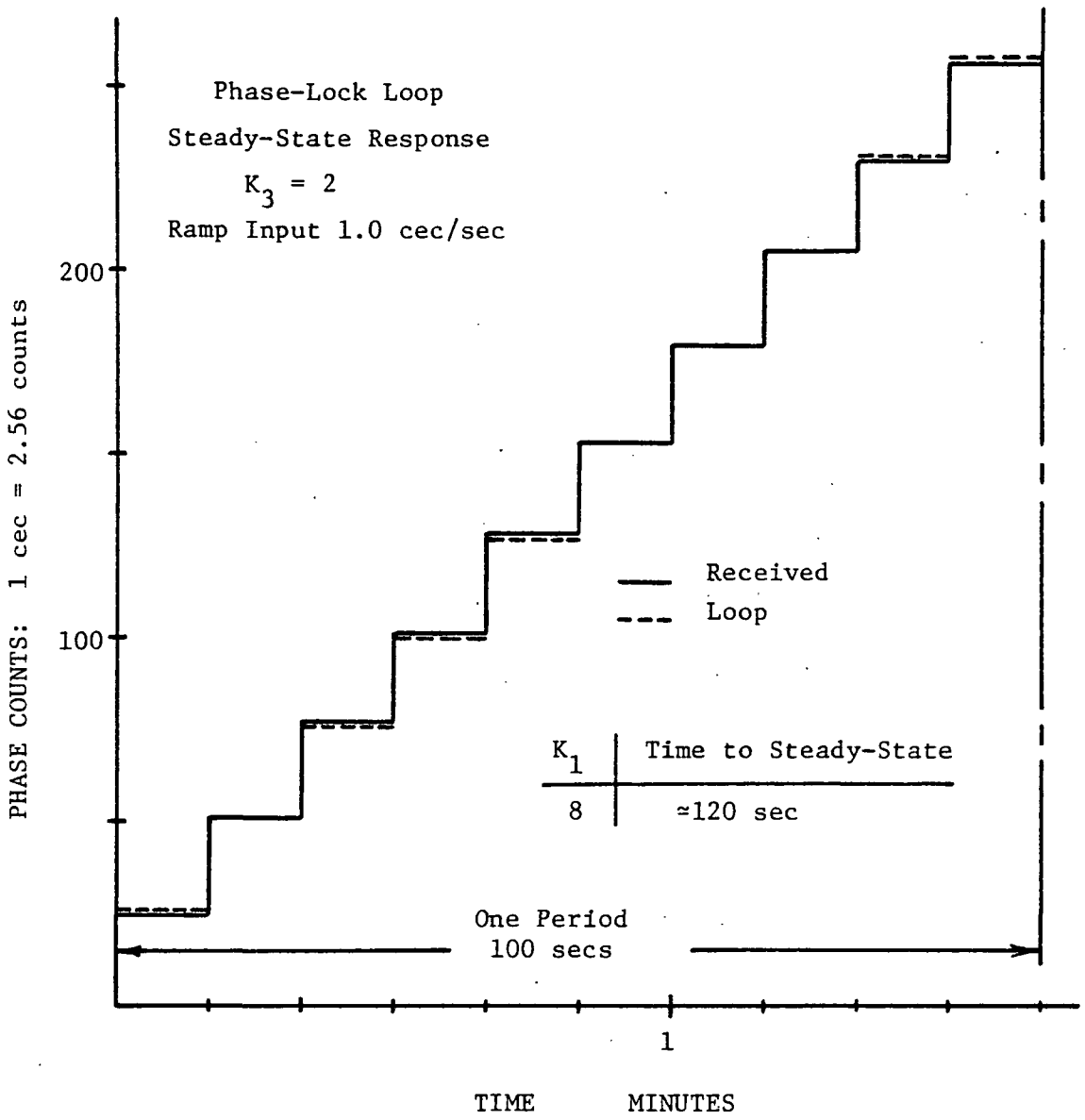


Figure A-13. Steady-State Response of Digital Phase-lock Loop to 1.0 Cec/Sec Ramp Input Over One Period With K_3= 2 and K_1= 8.

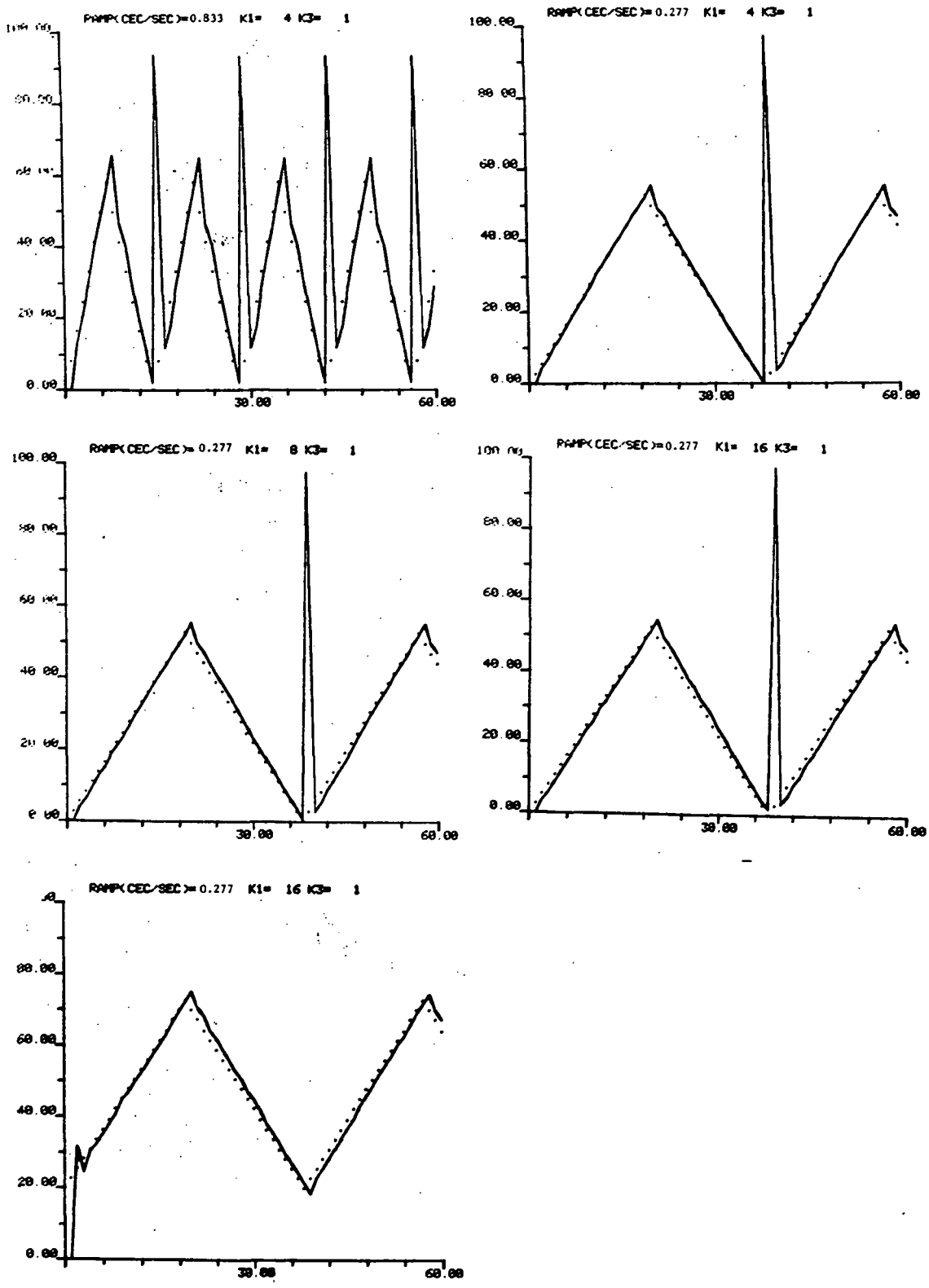


Figure A-14. Response of Digital Phase-Lock Loop With Ramp Input Which Has an Instantaneous Direction Change With $K_3 = 1$ and Various Values of K_1.

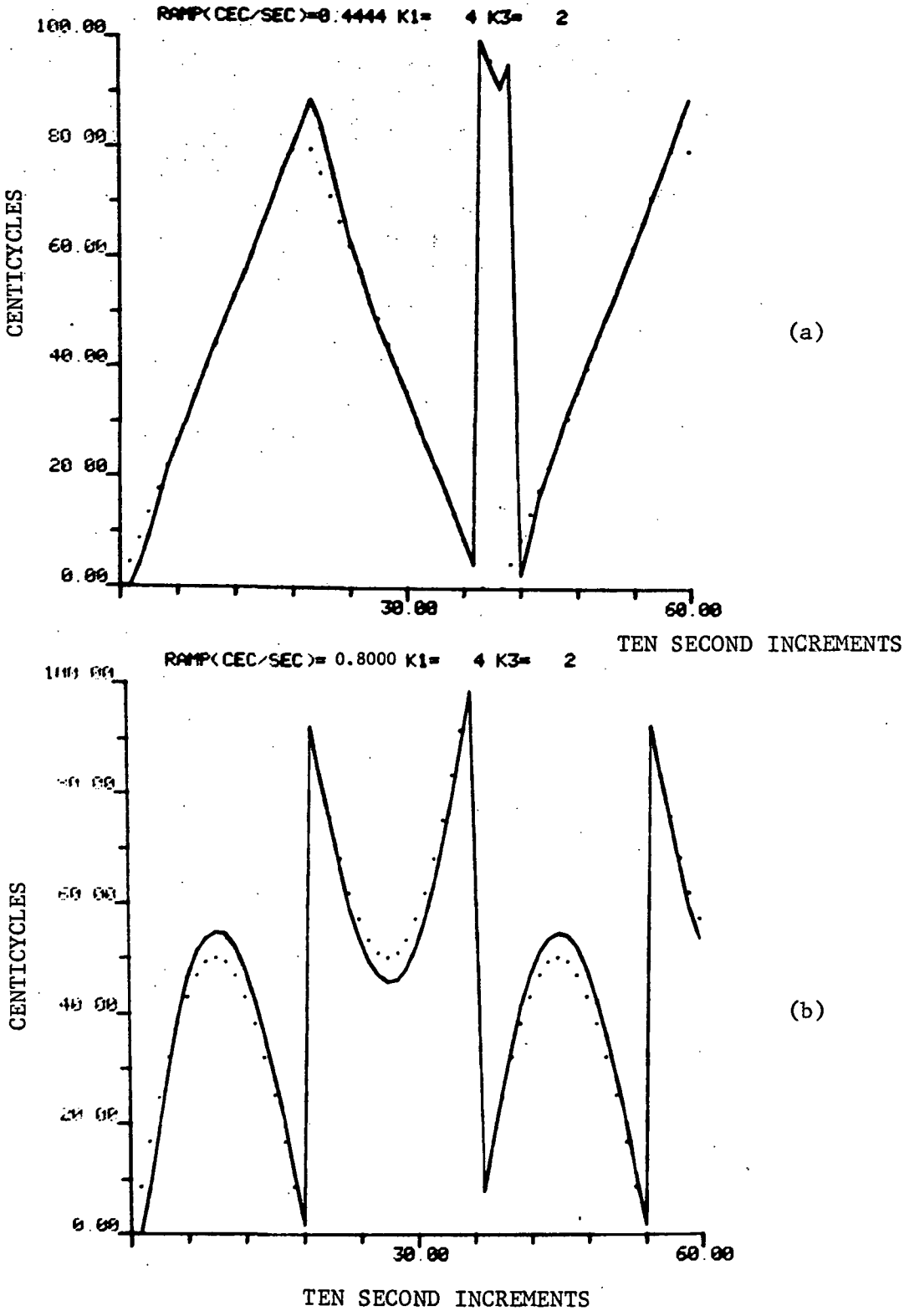


Figure A-15. Response of Digital Phase-Lock Loop With $K_3 = 2$ and $K_1 = 4$. (a) Input is Phase Ramp With Instantaneous Direction Changes, (b) Input is Phase Change Characteristics of a Racetrack Type Holding Pattern Corresponding to 500 Knot Velocity With 1°/Sec Turns.

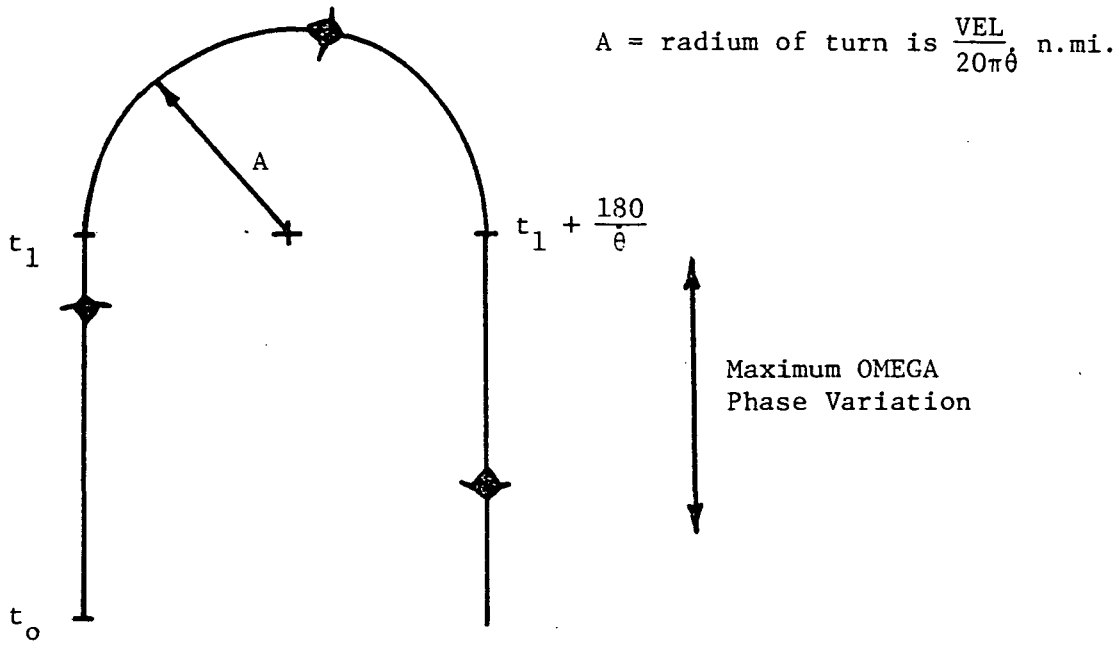


Figure A-16. Spatial Flight Path Trajectory of Aircraft with Velocity VEL Knots and Turn-rate $\dot{\theta}$ degrees/sec.

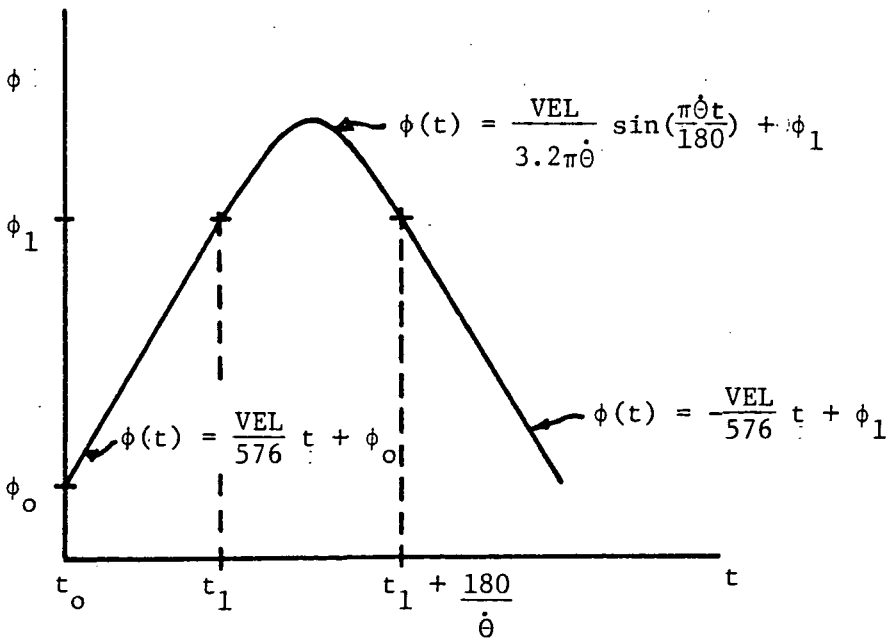


Figure A-17. Phase Plane Trajectory of Flight Path Which Represents Received OMEGA Phase at Aircraft.

represents the corresponding phase plane plot of the received phase which would be observed at the vehicle moving in the trajectory of Figure A-16. The function $\phi_R(t)$ is the received phase as a function of time assuming that the phase at time t_0 is $\phi_R(t_0) = \phi_0$ in centicycles at 10.2 kHz. The analysis that follows is all done at the 10.2 kHz OMEGA frequency.

To investigate the behavior of the DPLL on an aircraft receiver moving at a set velocity, VEL, then making a 180° turn and flying back all at the velocity, VEL, let VEL = a/c velocity, and $\dot{\theta}$ = a/c turn rate in degrees/sec. The phase ramp input in cecs/sec corresponding to velocity VEL is

$$\dot{\phi}_R = \frac{VEL}{576} \quad .$$

At time t_1 the a/c goes into turn, thus the received phase at time t_1 is

$$\phi_{R1} = \dot{\phi}_R t_1 + \phi_0$$

where ϕ_0 is initial received phase. During the turn the phase changes according to a sine function defined in terms of an offset, ϕ_1, an amplitude, A, and a period, T. The period T is defined in terms of the turn rate as

$$T = \frac{360}{\dot{\theta}} \text{ sec.}$$

The amplitude is defined as:

$$A = \frac{VEL \cdot T}{1152\pi} \text{ cecs} = \frac{VEL}{3.2\pi\dot{\theta}} \quad .$$

The offset is just ϕ_1 such that at time t_1 the received phase varies with time according to

$$\phi_R(t) = A \sin \left(\frac{2\pi}{T} t\right)$$

or

$$\phi_R(t) = \frac{VEL}{3.2\pi\dot{\theta}} \sin\left(\frac{\pi}{180}\dot{\theta}t\right) + \phi_1$$

where VEL is in knots, T is in seconds, and ϕ_1 is in cecs at 10.2 kHz. At t = T/2 the phase ramp represents the received phase as

$$\phi_R(t) = -\dot{\phi}_R t + \phi_1$$

where

$$\phi_R(t_1) = \phi_R(t_1 + T/2) = \phi_1 .$$

(In summary input VEL in knots, ϕ_0 in cecs, and $\dot{\theta}$ in degrees/sec.)
Thus

$$\dot{\phi} = \frac{VEL}{576}$$

$$\phi(t) = \dot{\phi}t + \phi_0 \qquad 0 \leq t < t_1$$

$$\phi(t) + \frac{VEL}{3.2\pi\dot{\theta}} \sin\left(\frac{\pi\dot{\theta}t}{180}\right) + \phi_1 \qquad t_1 \leq t < t_1 + T/2$$

$$\phi(t) = -\dot{\phi}t + \phi_1 \qquad t_1 + T/2 \leq t$$

This function is plotted in Figure A-17.

Figures A-18 and A-19 provide results of the loop tracking a phase change corresponding to a velocity of 576 knots along the half-racetrack (HT) pattern. All turns are at 3 degrees/sec. Initially all loop registers are zero and ϕ_0 = 0 in Figure A-18 and ϕ_0 = 20 cecs in Figure A-19. The initial leg of the HT pattern is traversed for 500 seconds (50 processing intervals) to allow the loop to lock to the phase ramp. For the situation in Figure A-18 the maximum lags during turn and the time until the lags are

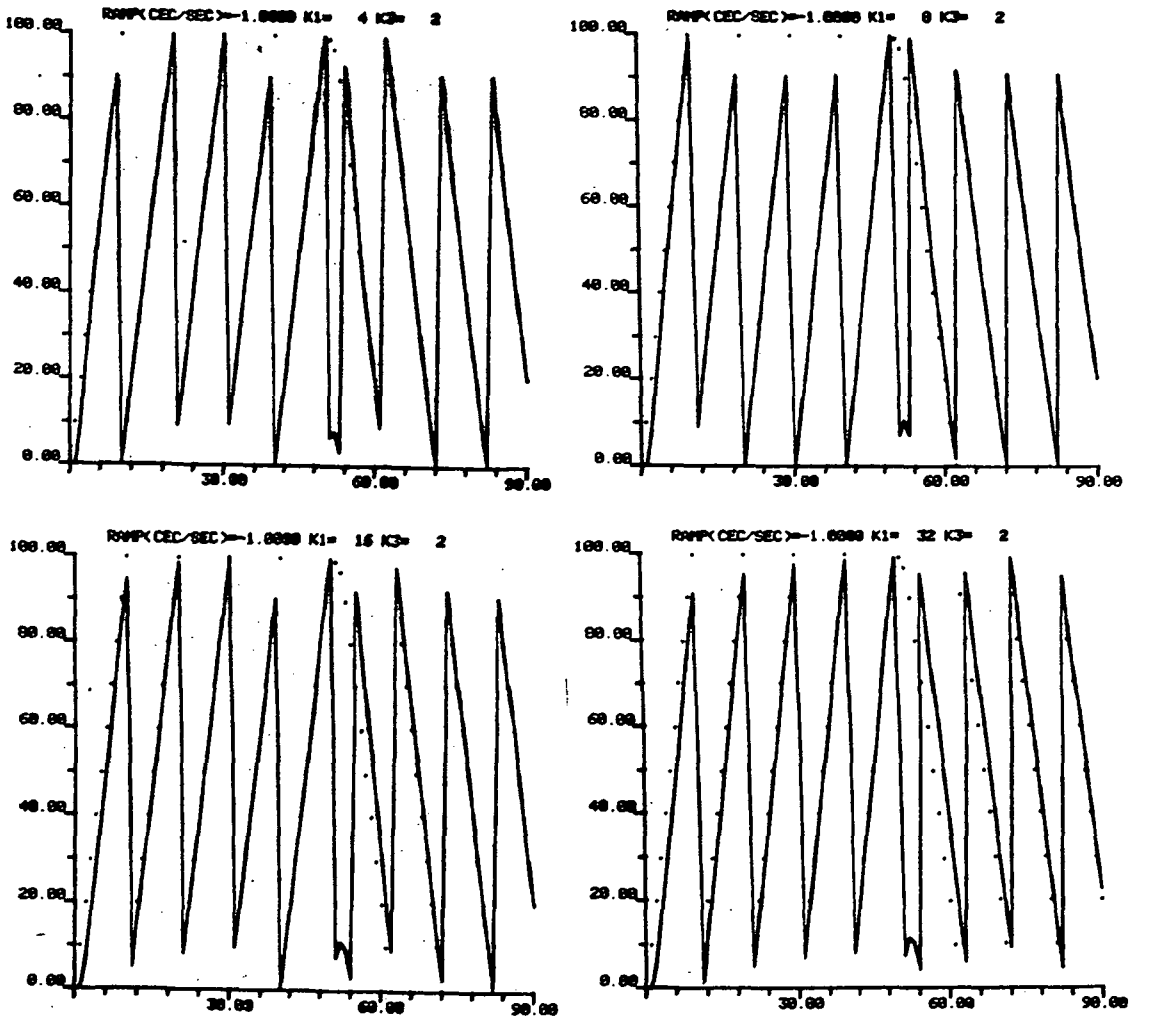


Figure A-18. Loop Response in HT Pattern, VEL = 576 kts, $\dot{\theta}$ = 3°/Sec, ϕ_o = 0.

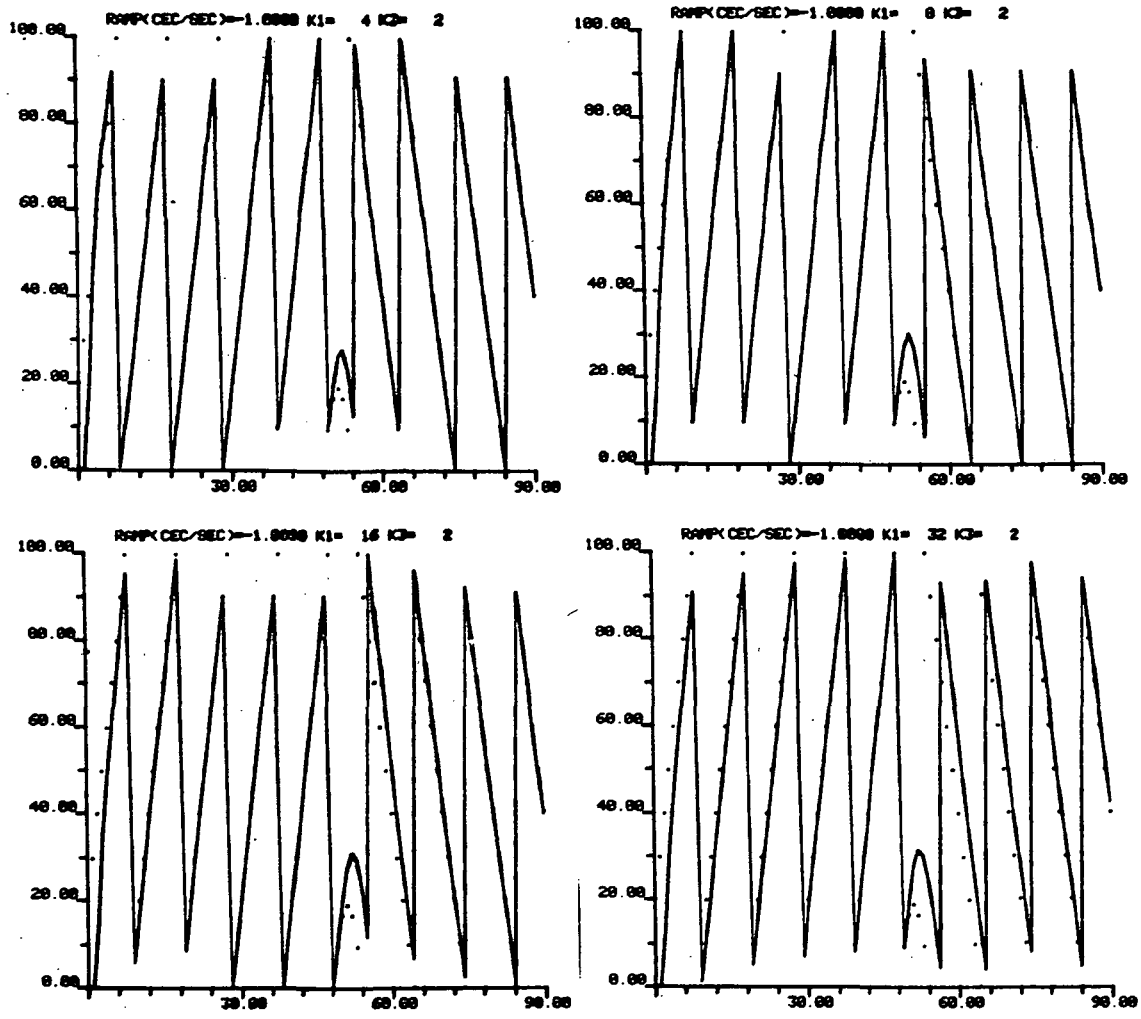


Figure A-19. Loop Response in HT Pattern, VEL = 576 kts, $\dot{\theta}$ = 3°/Sec, ϕ_o = 20 Cec.

within 1 cec upon completion of the turn are tabulated in Table A-3 for various values of K_1 (4, 8, 16, 32). In Figure A-19 a value of ϕ_0 = 20 cecs is used so that the phase trace during the turn can be illustrated better. The initial lock-up time is somewhat longer for the situation in Figure A-19 but the loop behavior in the turn and coming out of the turn are essentially the same as for Figure A-18 as summarized in Table A-3.

In Figures A-20 and A-21 several different velocities are attempted for K_1 values of 4, 8, 16, and 32. A ramp value of 0.1 corresponds to VEL $\approx$ 58 kts, 0.5 to VEL $\approx$ 288 kts, 1.0 to VEL = 576 kts, 1.2 to VEL = 691 kts, 1.5 to VEL $\approx$ 664 kts, 210 to VEL = 1152 kts, and 2.5 to VEL = 1440 kts. It can be noted that a loop with K_1 = 4 does not lose lock (skip cycles) until velocity is greater than 1152 kts whereas the other two K_1 values skip cycles at lower velocity values.

TABLE A-3

LAG Phase Values and Times to Lock-Up in HT Pattern

$K_3 = 2$

K_1	Time To* Initial Lock (sec)	Max Lag In Turn (cec)	Time To Error < 1 cec (sec after max lag)
4	50	13.7	80
8	90	19.1	110
16	220	23.0	240
32	440	25.0	530

*Initial lock is defined as earliest time loop phase remains less than 1 cec from received phase.

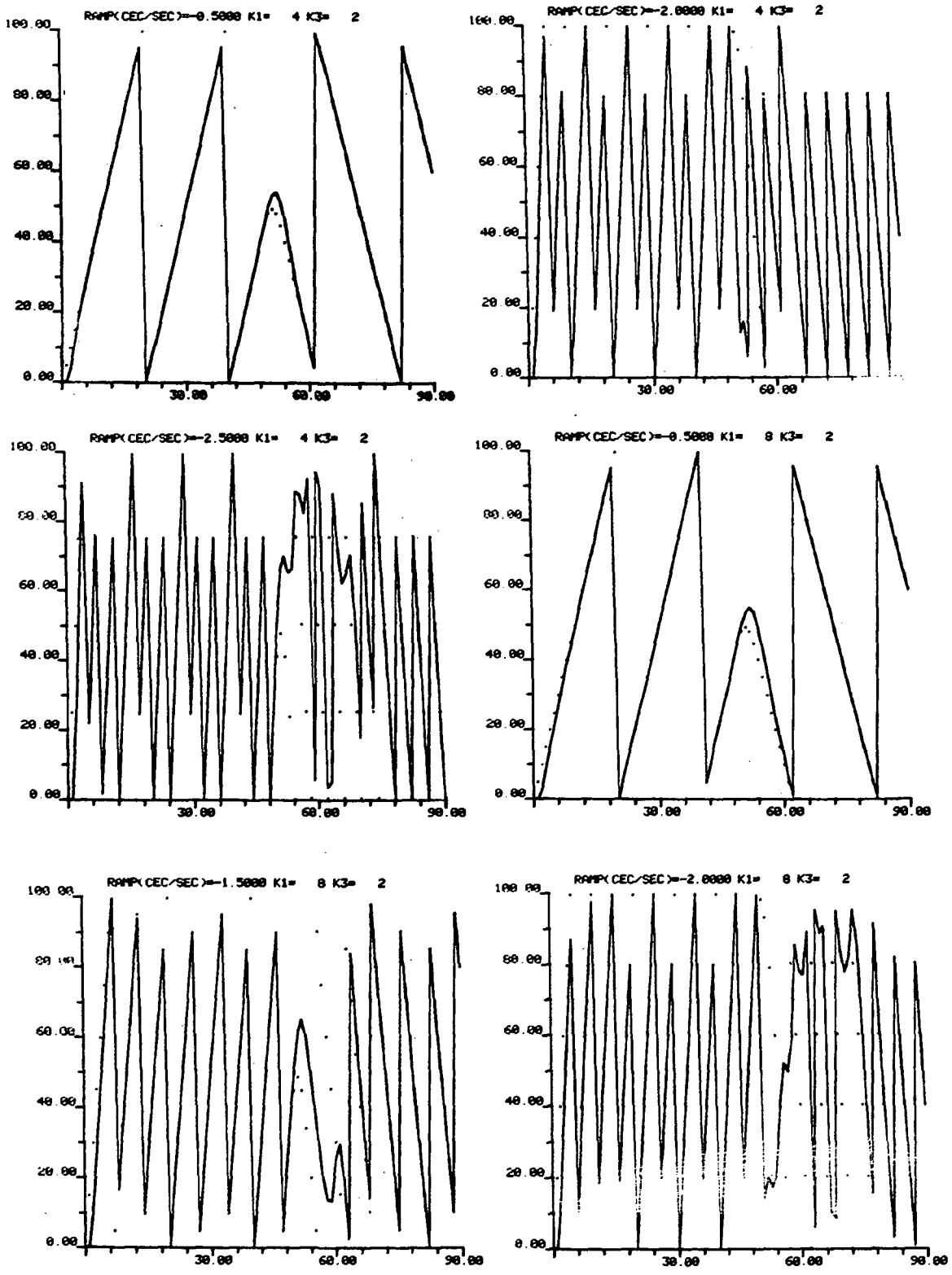


Figure A-20. Loop Responses in HT Pattern at Selected Velocities for K_1 = 4 and 8.

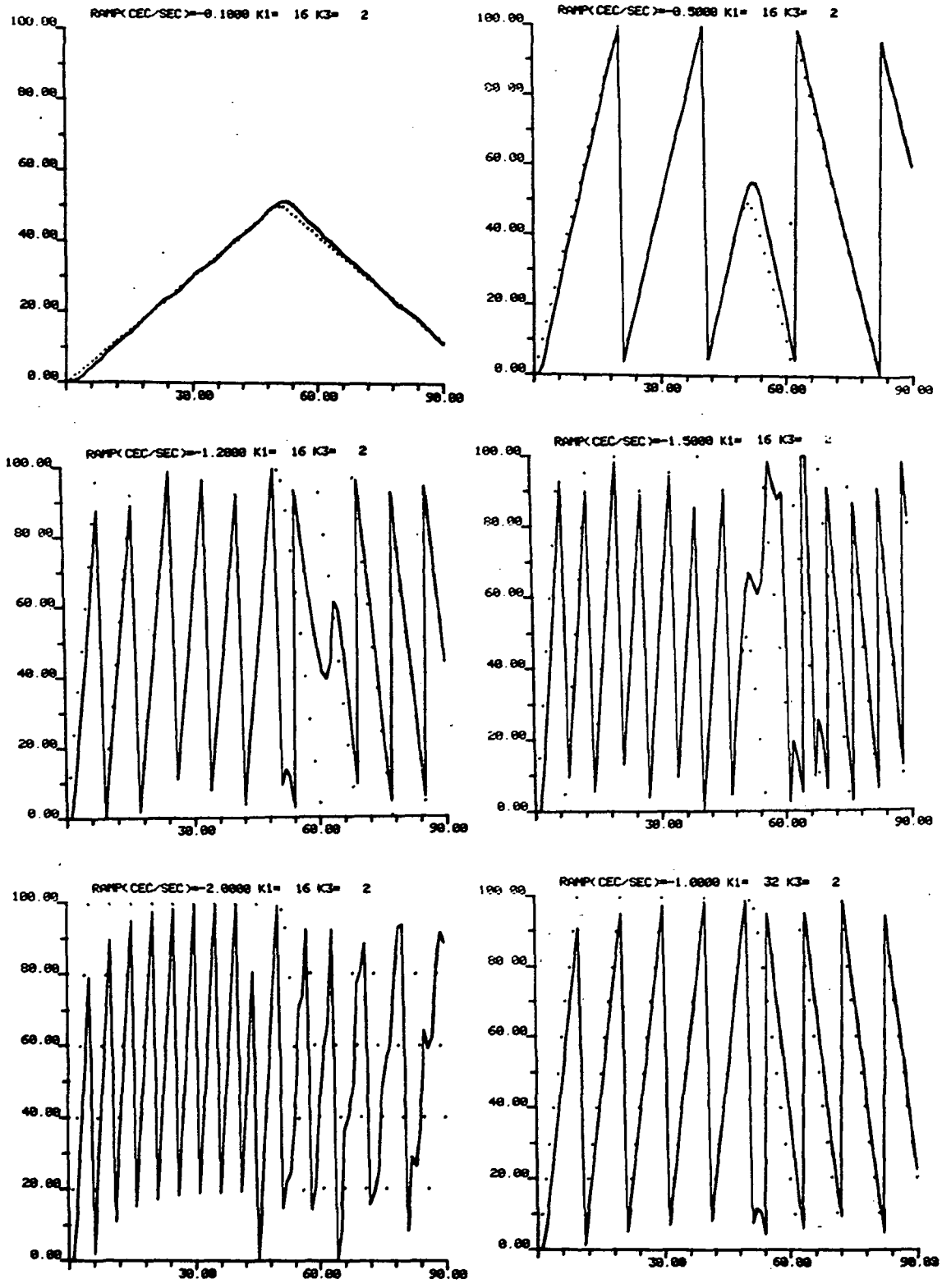


Figure A-21. Loop Responses in HT Pattern at Selected Velocities for K_1 = 16 and 32.

"Page missing from available version"

page 152

APPENDIX B

SPECIALIZED PROPAGATION PREDICTION CORRECTIONS

To complete the projected needs of data associated with the ground-based OMEGA program which was initiated in 1973 (ref. 1), special propagation prediction corrections (PPC) or skywave corrections have been collected and prepared for use. The Hydrographic Center of the Defense Mapping Agency publishes corrections for 4° latitude by 4° longitude geographical grids at hourly intervals. Special corrections were requested by RTI for each specific receiver location used (ref. 1) in the experimental data gathering program. These corrections were obtained at half hour intervals during the times data was collected at each of the receiver sites. Provision was made to further interpolate these corrections to estimate 10 second values for direct sample-by-sample integration with the phase measurements. In a previous report (ref. 1, Appendix G) a description of the restructured magnetic tape of corrections was given along with a description of a FORTRAN IV computer program written to run on the NASA Langley Research Center CDC-6600 which will generate the 10 second corrections and merge them with the data.

Work under this current contract has included requesting, reformatting, and delivering the complete set of PPC values to NASA-LRC on magnetic tape. Additionally two copies each of three bound volumes of hourly PPC values have been delivered to the contract technical monitor as resource material for visual data analysis. Of these volumes, one volume contains individual transmitter corrections (Norway, Trinidad, Hawaii, N. Dakota) for each receiver for each period during which data were collected. A second volume contains the same information in the form of LOP corrections for the six pair combinations of the four transmitters. The third volume contains differential corrections for those time periods when receiver two was remote from the Hampton, Va., location. Differential corrections are for the six LOP's. All correction tables include PPC data at all three OMEGA frequencies. In addition to hourly PPC values for each day during which

data was collected a maximum deviation value is calculated for each 24 hour period. Figure B-1 is a sample of these tables.

LOCATION LRC TAPE NO. 1.33 YEAR 1973 FREQ = 10.2

TRANS B-C

DAY #	MAX DEV	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
282	109	-11	2	15	29	42	48	50	51	51	52	66	89	64	50	36	20	1	2	0	-3	-6	-7	-17	-20
283	109	-10	2	15	29	42	48	50	51	51	52	66	90	64	50	36	20	1	2	0	-3	-6	-8	-17	-19
284	111	-10	2	16	30	42	48	50	51	51	52	66	91	63	50	37	20	1	2	1	-3	-5	-8	-17	-20
285	110	-10	3	16	30	43	48	50	51	51	52	66	90	66	49	37	21	1	2	1	-3	-5	-8	-17	-20
286	110	-10	3	16	30	43	48	50	51	51	52	66	90	67	49	38	21	1	2	1	-2	-5	-8	-17	-20
287	109	-10	3	17	30	43	48	50	51	51	52	66	90	67	51	38	21	1	2	1	-2	-5	-8	-17	-19
288	109	-9	4	17	31	43	48	50	51	51	52	66	90	67	52	38	21	1	2	1	-2	-5	-7	-18	-19

Figure B-1. PPC table of hourly values for 10.2 kHz LOP BC* at Hampton, Virginia, receiver site for days 282-288.

*BC was Trinidad-Hawaii at this time

APPENDIX C

METHODS FOR CONVERSION OF LOP FIX TO LAT/LONG.

I. Orthogonal Step Method

Given: LOP crossing fix $LOPX_d$, $LOPY_d$

To find: Estimated Latitude/Longitude at this fix $\hat{\phi}_d$, $\hat{\lambda}_d$

a. With a starting point latitude ϕ_0, longitude λ_0 calculate $LOPX_0$, $LOPY_0$

b. Calculate LOP differences $\Delta LOPX_i = LOPX_d - LOPX_0$

$$\Delta LOPY_i = LOPY_d - LOPY_0$$

c. Using a degrees lat. or long. and scale β from baseline ratio calculate step size

$$\Delta 1_i = \beta \Delta LOPX_i$$

$$\Delta 2_i = \beta \Delta LOPY_i$$

d. *For largest ΔK_i (K = 1,2) step in longitude or latitude by ΔK_i degrees either 1) $\lambda_i = \lambda_{i-1} + \Delta K_i$

or 2) $\phi_i = \phi_{i-1} + \Delta K_i$

e. Calculate $LOPX_{i+1}$, $LOPY_{i+1}$ at new lat./long. λ_i, ϕ_i

f. Calculate difference between LOP's at (i + 1) iteration and at desired point.

$$\Delta LOPX_{i+1} = LOPX_d - LOPX_{i+1}$$

$$\Delta LOPY_{i+1} = LOPY_d - LOPY_{i+1}$$

g. Check convergence by comparing ΔLOP values with stopping rule criteria

If $|\Delta LOPX_{i+1}| < \varepsilon$ and $|\Delta LOPY_{i+1}| < \varepsilon$ do step h

otherwise reiterate beginning with step c.

*On first iteration a step in longitude is compared to a step in latitude; use step direction with smallest $\Delta LOPX_1 + \Delta LOPY_1$.
On successive iterations step in different direction from previous step, i.e. step in "orthogonal" directions at each successive step.

h. Convergence satisfied: Calculate $\hat{\phi}_d$, $\hat{\lambda}_d$ using $LOPX_{i+1}$ and $LOPY_{i+1}$ calculated at step e.

For the orthogonal step method the scale factor β is calculated using

$$\beta = \frac{\Lambda}{6\times10^3} \text{ degrees/cec}$$

where Λ is wavelength at the OMEGA frequency of operation (n.mi./lane) and an approximate equatorial longitude value of 60 n.mi./degree is used in calculating β. In each situation involving calculation of LOP values at a particular latitude/longitude point (ϕ, λ) the distance on the earth's surface to each transmitter of a pair is used along with the wavelength Λ and the centerline value N_{ij} to yield

$$LOP_{ij} = \left[\frac{d_i - d_j}{\Lambda} + N_{ij}\right] \times 100 \quad \text{cec}$$

A suitable value of $\varepsilon = 1.0$ cec.

II. Pierce Method for Determining Lat./Long. of an LOP fix (ref. 3)

Given: LOP measurements at one OMEGA frequency involving up to 4 transmitters (3 independent LOP's with 4 transmitter stations).

a. Let: S_i represent phase of i^{th} transmitter in cycles relative to mean,

CTR_j represent center lane value (chart) at frequency j,

λ_j represent wavelength at frequency j (chart if using corrections),

$T_{obs}(i)$ in degrees of central sector using data from i^{th} transmitter.

Thus for $\forall i$ calculate

$$T_{obs}(i) = (S_i - \bar{S}) \cdot \frac{180\ \lambda_i}{\pi r_a}$$

where r_a is equatorial earth radius (6378166m)

An example: (4 stations received)

$$S_1 - \bar{S} = \text{A-MEAN} = \frac{1}{4}(3AC - 2BC + BD)$$

$$S_2 - \bar{S} = \text{B-MEAN} = \frac{1}{4}(BD + 2BC - AC)$$

$$S_3 - \bar{S} = \text{C-MEAN} = \frac{1}{4}(BD - 2BC - AC)$$

$$S_4 - \bar{S} = \text{D-MEAN} = \frac{1}{4}(2BC - 3BD - AC)$$

where $AC = AC' - CTR_j$ with CTR_1 (j = 1 corresponds to 10.2 kHz) = 900

$BC = BC' - CTR_j$

$BD = BD' - CTR_j$

and AC', BC', BD' are LOP measurements in cycles.

b. Assume some starting point with latitude ϕ_p and longitude λ_p. This should naturally be as close to the true position as is possible but can be a general estimate such as an approximate mid-point between the OMEGA transmitters used. The procedure to transform the OMEGA fix to a latitude/longitude position is an iterative algorithm. The number of steps to convergence is dependent on how close the starting point is to the actual position.

c. Using the starting point ϕ_p, λ_p form

$$\delta S_i = \frac{r_a - r_b}{4} \left[\frac{3 \sin \sigma_i - \sigma_i}{1 + \cos \sigma_i} \left\{ (1 + \sin\phi_i \sin\phi_p)^2 - (\cos\phi_i \cos\phi_p)^2 \right\} \right.$$

$$\left. - \frac{3 \sin \sigma_i + \sigma_i}{1 + \cos \sigma_i} \left\{ (1 - \sin\phi_i \sin\phi_p)^2 - (\cos\phi_i \cos\phi_p)^2 \right\} \right]$$

where $\cos\sigma_i = \sin\phi_i \sin\phi_p + \cos\phi_i \cos\phi_p \cos(\lambda_i - \lambda_p)$; $[\sigma_i > 0]$. Here ϕ_i, λ_i is the latitude/longitude of OMEGA transmitter station i, r_a and r_b are the equatorial and polar earth radii respectively.

d. Using the correction δS_i (meters) calculate

$$T_i = \left(\sigma_i + \frac{\delta S_i}{r_a} \right) \cdot \frac{180}{\pi} \quad \text{degrees (central sector) to transmitter i}$$

and find

$$\theta_i = \cos^{-1} \left\{ \frac{\sin\phi_i - \sin\phi_p \cos\sigma_i}{\cos\phi_p \sin\sigma_i} \right\}, \quad \text{with the sign of } (\lambda_i - \lambda_p) .$$

e. The average distance to various transmitters is

$$\mu_T = \frac{1}{n} \sum_{i=1}^{n} T_i \qquad \text{(degrees)}$$

f. The step size for each transmitter is

$$\Delta T_i = T_i - \mu_T - T_{obs}(i)$$

g. The latitude/longitude increments are

$$\Delta\phi_i = \Delta T_i \cos\theta_i$$

$$\Delta\lambda_i = \frac{\Delta T_i \sin\theta_i}{\cos\phi_p}$$

h. Form an average latitude/longitude increment as

$$\overline{\Delta\phi} = \frac{1}{n} \sum_{i=1}^{n} \Delta\phi_i$$

$$\overline{\Delta\lambda} = \frac{1}{n} \sum_{i=1}^{n} \Delta\lambda_i$$

i. A new position estimate is formed using*

$$\hat{\phi}_p = \phi_p + G \ \overline{\Delta\phi}$$

$$\hat{\lambda}_p = \lambda_p + G \ \overline{\Delta\lambda}$$

*To determine the step gain G use either G = 1.7 or G = $(1.2 + \log_{10} k)(\cos\phi_p)^{1/4}$ where k is the iteration number.

j. Use $\hat{\phi}_p$ and $\hat{\lambda}_p$ as a new starting point and repeat the algorithm by recalculating δS_i, T_i, and θ_i to get a new position estimate (step c).

k. The procedure is iterated until a satisfactory position estimate is obtained using some defined stopping rule. Two stopping rules might be considered:

1. If two LOP's are used the LOP crossing is a fix and the algorithm to determine the lat./long. of this fix can be repeated until an inverse calculation of the LOP from the estimated lat./long. $(\hat{\phi}_p, \hat{\lambda}_p)$ is within some ε of the measured values (say 1 cec).

2. An alternate stopping rule is necessary if more than two LOP measurements are used since a fix is in general an estimate of position based on three or more LOP pair fixes. This rule may also be applied with a two LOP fix. At each iteration calculate

$$\Delta T_{RMS} = \left\{ \frac{1}{n} \sum_{i=1}^{n} (\Delta T_i - \overline{\Delta T})^2 \right\}^{1/2}$$

and compare to some ε (say .004 degrees) which is smaller than the expected propagational error.

APPENDIX D

OMEGA CHART LATTICE PROGRAM (FLATBED)

This Appendix described the program FLATBED developed for use on the CDC-6600 computer system at the NASA-Langley Research Center facility. The computer program is written in FORTRAN IV and consists of a driver and eleven subroutines. Output is designed for a flat-bed x-y plotter which can be set up with a Lambert projection topographical map. User input consists of latitude/longitude grid boundaries, a latitude and longitude increment, latitude/longitude coordinates of the desired OMEGA transmitter pair, the OMEGA lane counting offset value, appropriate wavelength values (may be separate for each transmitter) and the latitude/longitude measure of map range in inches.

D.1 General Description of the OMEGA Chart Algorithm

Upon definition of the area of interest in terms of latitude/longitude boundaries a matrix of points is defined in terms of LOP value at each point in the map area determined on the basis of the latitude and longitude increments. LOP values are determined by calculating at each grid point the earth's surface distance to each OMEGA transmitter using subroutine DISFUN which is based on the Fifth Inverse Method of Sodano (see ref. 1, Appendix C). Using an estimate for wavelength at the OMEGA frequency of interest an LOP phase value can be calculated. Conventional numbering is provided so that lane values are consistent with standard published OMEGA charts. At this point interpolation between array points is accomplished using subroutine INTERP to determine the nearest latitude/longitutde point to a point in the array through which an integral LOP crosses. Then subroutine SORT employs a simple bubble sort to set up the array of points defining integral LOP crossings in order by latitude in preparation for plotting. Subroutine LAMBRT is then called to convert each point defined on an LOP in latitude/longitude coordinates to Lambert x,y coordinates which are then scaled to plotter pen coordinates.

D.2 Conversion of Latitude/Longitude to Lambert x,y Corrdinate

The coordinate transformation algorithm to convert a latitude/longitude position to a Lambert x,y position is presented in terms of an example involving the Washington Sectional Aeronautical Chart published by the Coast & Geod. Sur., ESSA, Dept. of Commerce. This map covers latitudes 36° to 40° and longitude 72° to 79° and the projection standard parallels are 33°20' and 38°40' (ref. 17). Table D-1 provides a definition of constants used in the transformation. For a given latitude/longitude ϕ,λ calculations proceed as follows:

$$\theta = L_6(\lambda_0 - \lambda)$$

$$r = K * \left[\cot\left(\frac{\pi}{4} + \frac{\phi}{2}\right)\right]^{\ell}\left[\frac{1 + \varepsilon\sin\phi}{1 - \varepsilon\sin\phi}\right]^{\varepsilon\ell/2}$$

$$x = r\sin\theta$$

$$y = L_4 - r\cos\theta$$

The inverse procedure is somewhat more complex. Given a position in Lambert coordinates, x,y

$$\theta = \arctan\frac{x-L_1}{L_4-y}$$

$$\lambda = L_2 - \frac{\theta}{L_6}$$

where θ and λ are in seconds. Then

$$R = \frac{L_4-y}{\cos\theta}$$

and S is calculated according to

$$S_1 = \frac{L_4 - L_3 - y + 2R \sin^2\left(\frac{\theta}{2}\right)}{L_5}$$

$$S_2 = \frac{S_1}{1 + \left(\frac{S_1}{10^8}\right)^2 L_9 - \left(\frac{S_1}{10^8}\right)^3 L_{10} + \left(\frac{S_1}{10^8}\right)^4 L_{11}}$$

$$S_3 = \frac{S_1}{1 + \left(\frac{S_2}{10^8}\right)^2 L_9 - \left(\frac{S_2}{10^8}\right)^3 L_{10} + \left(\frac{S_2}{10^8}\right)^4 L_{11}}$$

$$S = \frac{S_1}{1 + \left(\frac{S_3}{10^8}\right)^2 L_9 - \left(\frac{S_3}{10^8}\right)^3 L_{10} + \left(\frac{S_3}{10^8}\right)^4 L_{11}}$$

Form $\omega' = L_7 - 600$ in minutes

$\omega'' = 36000 + L_8 - S/30.87167045$ in seconds.

Then $\omega = \omega' + \omega''$

and

$\phi' = L_7 - 600$

and $\phi'' = \omega'' + [1047.54046 + (6.211776 + .036448 \cos^2\omega) \cos^2\omega] \sin\omega \cos\omega$

with $\phi = \phi' + \phi''$

resulting in a position ϕ,λ in latitude and longitude.*

*Constants have been calculated for the Washington sectional area.

TABLE D-1

Lambert Projection Constants

CONSTANT	DEFINITION	VALUE
L_1	false easting or x coordinate of central meridian	0.0
L_2	central meridian expressed in seconds	
L_3	map radius of the central parallel (ϕ_0)	8774941.796 m
	ϕ_0 = arcsin ℓ $\quad \ell$ = .58800002 K = 13016802m	
	$R_{\phi_0} = L_4 = K\left[\cot\left(\frac{\pi}{4} + \frac{\phi_0}{2}\right)\right]^{\ell}\left[\frac{1+\varepsilon\sin\phi_0}{1-\varepsilon\sin\phi_0}\right]^{\varepsilon\ell/2}$	
	ε^2 = .006722670022	
L_4	map radius of the lowest parallel of the projection table plus the y value on the central meridian at this parallel. lowest parallel is 31°30' (30' below zone limit of 32°)	9275742.094 m
	$R_6 = L_4 = K\left[\cot\left(\frac{\pi}{4} + \frac{\phi}{2}\right)\right]^{\ell}\left[\frac{1+\varepsilon\sin\phi}{1-\varepsilon\sin\phi}\right]^{\varepsilon\ell/2}$ $\quad \phi = 31°30'$	
	K = 13016802 m	
L_5	scale (m) of the projection along the central parallel (ϕ_0)	.9989218077
	$m_{\phi_0} = L_5 = \frac{\ell - R_{\phi_0}}{N_0\cos\phi_0}$ $\quad R_{\phi_0} = L_3$	
	$N_0 = \frac{a}{(1-\varepsilon^2\sin^2\phi_0)^{1/2}}$	
	a = 6378388	
L_6	is the ℓ computed from the basic equations for the Lambert projection with two standard parallels.	.58800002

TABLE D-1 (Contd)

CONSTANT	DEFINITION	VALUE
L_7	degrees and minutes portion, in minutes, of the rectifying latitude for ϕ_0	2152
	$\omega_0'' = \phi'' - [1052.893882 - (4.483344 - .023520\cos^2\phi)\cos^2\phi]\sin\phi\cos\phi$ (in seconds)	
L_8	remainder of ω_0, i.e. the seconds	35.38512
L_9	$\left(\frac{1}{6R_0N_0}\right) \times 10^{16}$, $R_0 = \frac{a(1-\varepsilon^2)}{(1-\varepsilon^2\sin^2\phi_0)^{3/2}}$	41.05207625/m^2
	$N_0 = \frac{a}{(1-\varepsilon^2\sin^2\phi_0)^{1/2}}$, a = 6378388m (major earth radius)	
L_{10}	$\frac{\tan\phi_0}{24(R_0N_0)^{3/2}} \times 10^{24}$	114.7627691/m^2
L_{11}	$\frac{5+3\tan^2\phi_0}{120\ R_0N_0^3} \times 10^{32}$	3314.761234/m^4

p166

D.3 Description of Fortran IV Program FLATBED

Figure D-1 provides a flowchart of the main driver program and associated support subroutines designed to plot an OMEGA LOP grid superimposed on a latitude-longitude grid on a flatbed plotter. This version has been designed to plot between latitudes 32° and 40° using a parallel of 31°30' to establish the Lambert y=0 point. The central meridian of the map (XCNTR) is used to determine the x=0 point. The calculated Lambert x-y points are all shifted relative to the plotter origin for plotting. Each point on the plot is reduced to pen position x-y values so that the resulting plot is consistent with Lambert projection topographical maps. The program uses plotter associated subroutines PSEUDO, CALPLT, NUMBER, and PNTPLT which have not been included in this description.

The scale is set to 1:500000 which results in approximately 6.86 n.mi/inch for the resulting plot which will match the standard aeronautical sectional maps. This is set in the main program and is used as a means of scaling the pen movement. Input data to the main driver routine includes the number of LOPs and the number of degrees (may be less than 1) between lines of longitude and latitude to be plotted. The subroutine THEGRID is the principle subroutine which controls the plotting of latitude/longitude grid and LOP grid. Upon return to FLATBED the plotter output file is terminated before ending.

The subroutine THEGRID sets the lower left hand corner of the map at 5.0, 3.9 on the plotter and proceeds with the plot in two sections. The grid is done left longitude to center and then center to right longitude in two steps. The four corners of each half are specified by input latitude/longitude pairs. As each half is plotted the corners are marked first to provide registration if it is desired to overlay the grid on a map. Then the logic proceeds to plot the grid. Upon completing the left half the right half is plotted in the same manner. Subroutine BOXIT marks the map corners while subroutine FLAT and FLON are used to plot the latitude/longitude positions. After LOP positions are calculated subroutine LAMPLOT actually carries out the sequence of plot instructions. There is no limit to the number of different LOPs which may be plotted on a given map.

Figure D-2 provides a listing of the Fortran IV program FLATBED and associated subroutines.

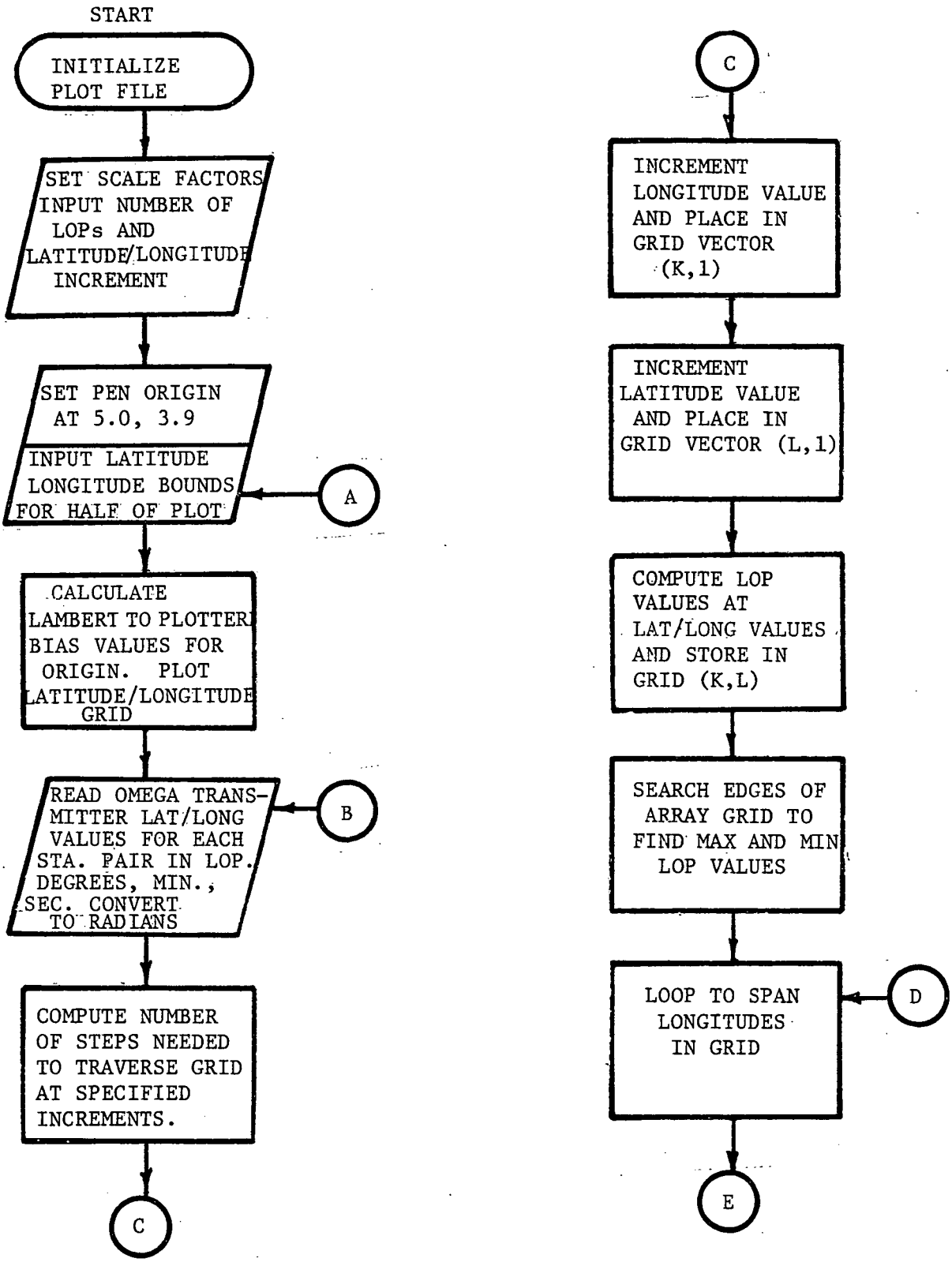


Figure D-1. Flowchart of OMEGA Chart Lattice Computer Program

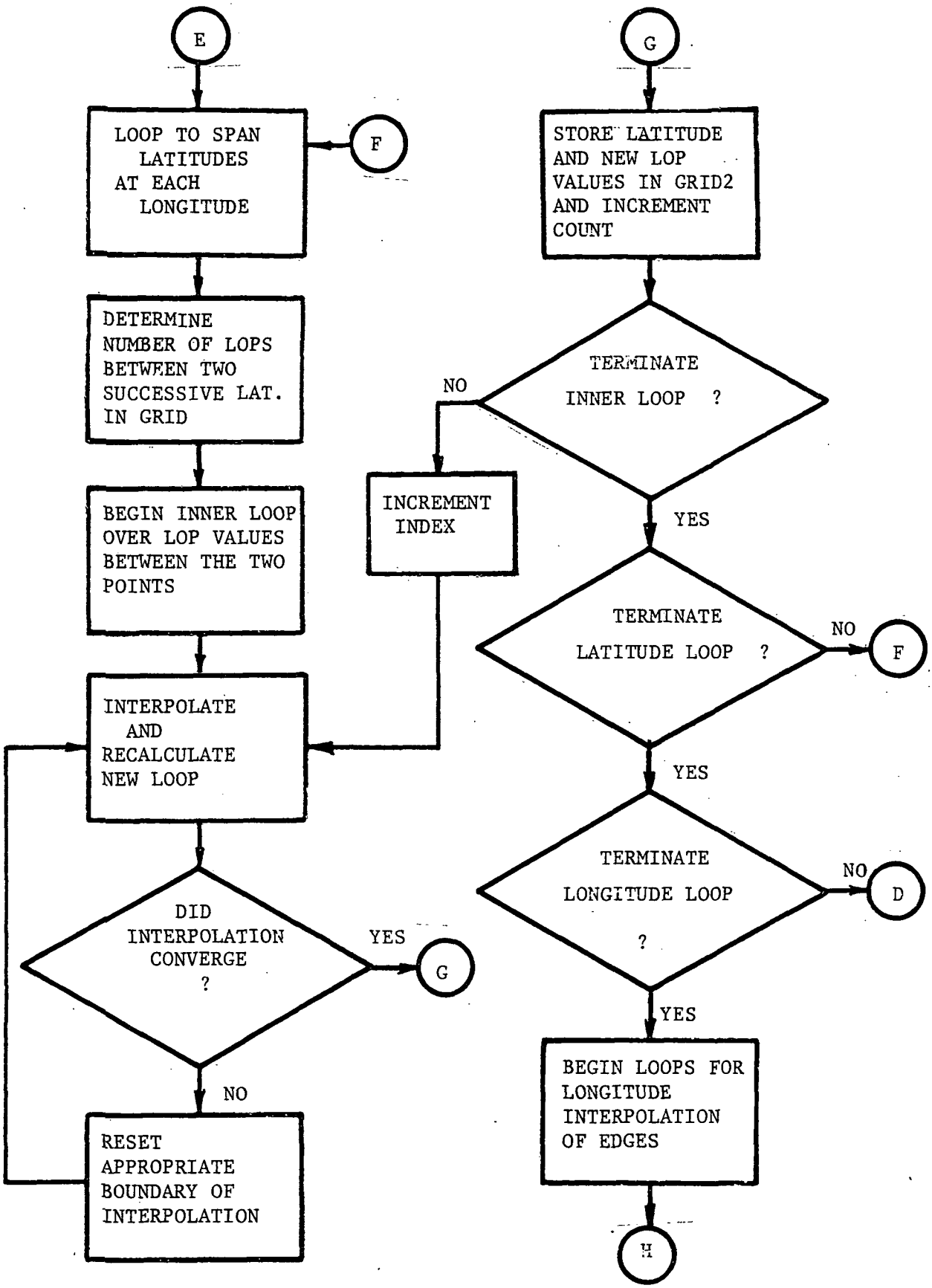


Figure D-1. Continued

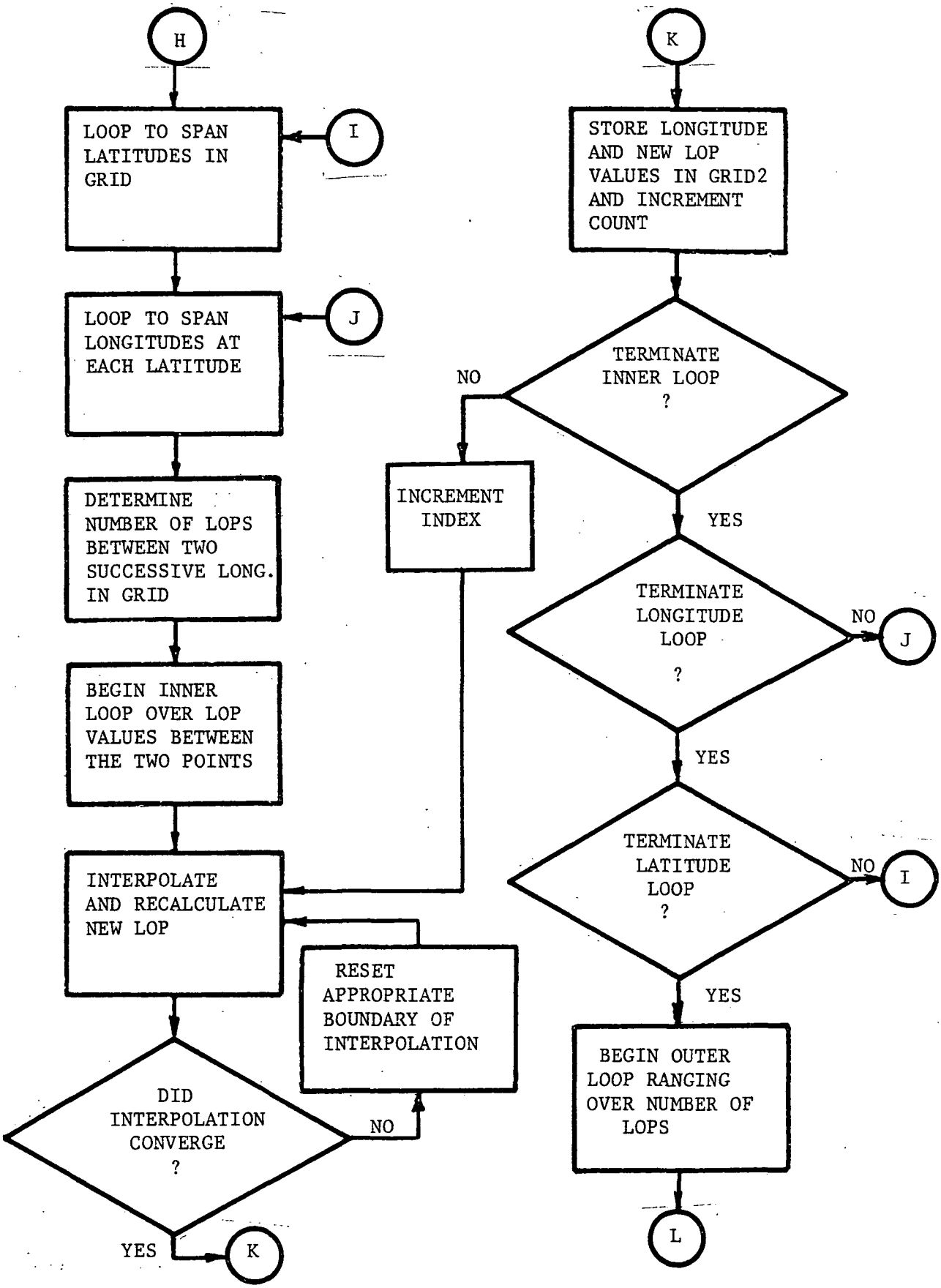


Figure D-1. Continued

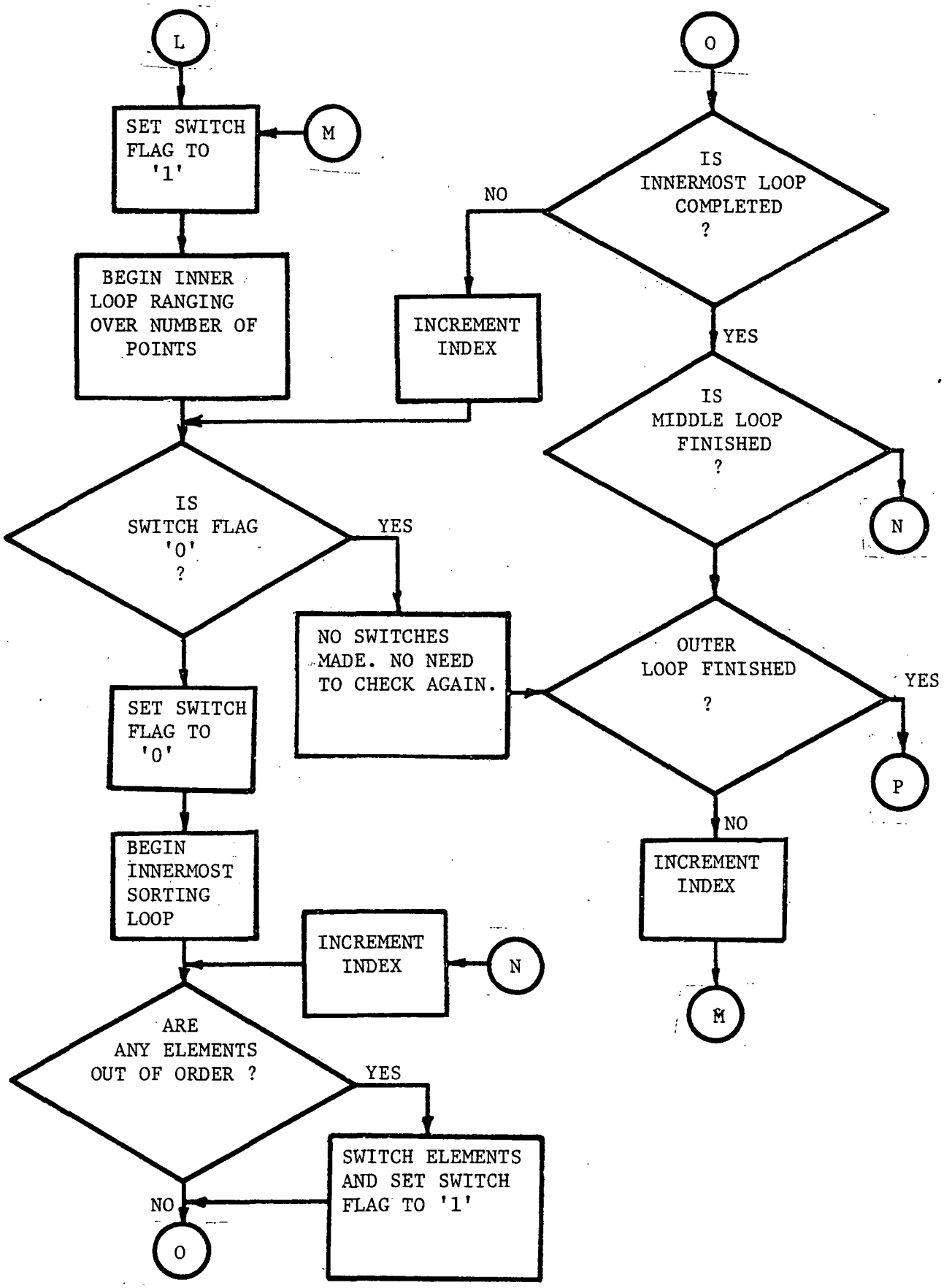


Figure D-1. Continued

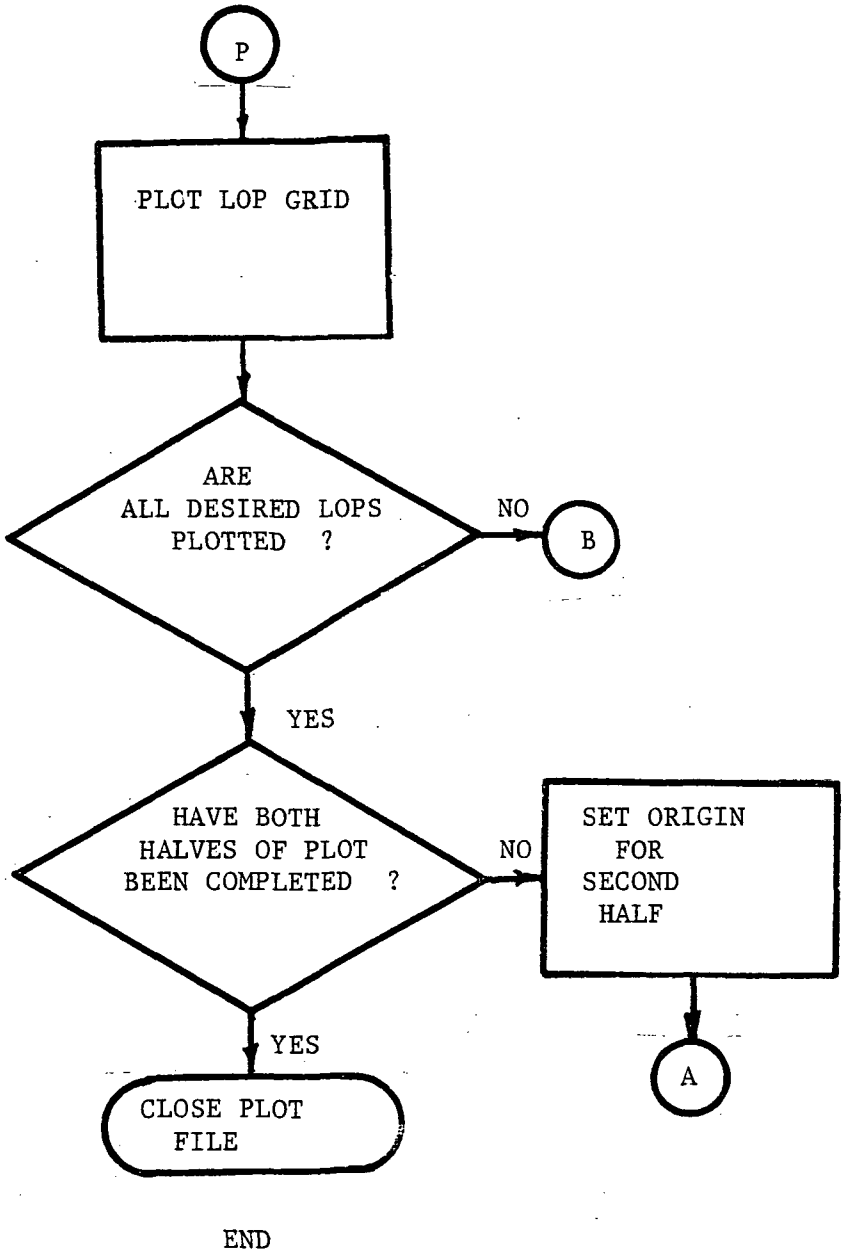


Figure D-1. Continued

```
                PROGRAM FLATBED(TAPE10,INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
          C     FLATBED GENERATES A LAT-LON AND AN LOP GRID IN PARTS SUITABLE FOR USE
          C     ON THE LRC FLATBED PLOTTER.
000003            COMMON/BB/OFFSET,WAVE1,WAVE2
000003            COMMON/BIAS/EAST,YBIAS
000003            COMMON/CC/RLONL,RLONR,RLATL,RLATR,RLTIN,RLOIN,RLT1,RLN1,RLT2,RLN2
000003            COMMON/DD/GRID(20,20)
000003            COMMON/FF/I,J
000003            COMMON/GG/IOFF(20),GRID2(20,20,3),ISUB
000003            COMMON/NOS/ILAT,ILON
000003            COMMON/MATH/ PID2,PI,PI3D2,PI2,DTOR,RTOD
000003            COMMON/NXEBIAS/BIASN,BIASE,IUNIT
000003            COMMON/METRIC/SCX,SCY
000003            COMMON/SCALES/XSCALE,YSCALE
000003            COMMON/TT/NLATS,NLONS
000003            COMMON XCNTR
000003            COMMON/ZZ/A,B,C,D
000003            REAL MSCALE
000003            NAMELIST/INPUTA/XLGTH,YLGTH,ILOP,DEL
          C
          C *** INITIALIZE PLOT FILE
          C
000003            CALL PSEUDO
          C
          C *** SET CONSTANTS CALCULATED OUTSIDE OF THE PROGRAM
          C DEFAULTS FOR BOX OPTION
000004            YLGTH=10.
000006            XLGTH=20.
000007            XSCALE=0.02
000011            YSCALE=0.02
          C DEFAULT FOR NUMBER OF LOP SETS
000015            ILOP=0
          C *** CALCULATE 1:500000 SCALE
000016            RSCALE=41666.667/6076.10333
000020            WRITE(6,105) RSCALE
000025          105 FORMAT(//* MAP SCALES IN VARIOUS UNITS*//
                   1*  1:500,000---NAUTICAL MILES/INCH =*F10.5)
000025            BIASE=0.0
000026            BIASN=0.0
000027            TSCALE=500000./(12.*5280.)
000032            WRITE(6,106) TSCALE
000037          106 FORMAT(*  1:500,000---STATUTE MILES/INCH = *F10.5)
000037            MSCALE=500000./39.37
000041            WRITE(6,107) MSCALE
000047          107 FORMAT(*  1:500,000---METERS/INCH = *F15.5)
000047            XSCALE=1.0/MSCALE
000051            XSCALE=XSCALE*0.5
000053            YSCALE=XSCALE
000054            SCX=XSCALE
000055            SCY=YSCALE
000075            READ INPUTA
000100            PRINT INPUTA
          C *** GENERATE AND PLOT THE GRIDS
000103            CALL THEGRID(DEL,ILOP)
000105            CALL CALPLT(0.,0.,999)
000110            PRINT 665
000112          665 FORMAT(//40X,10(*-*),*MAP(S) COMPLETED*,10(*-*))
000113            STOP 666
000115            END
```

Figure D-2. Listing of Fortran IV Program to Generate OMEGA Chart Lattice.

```
                SUBROUTINE THEGRID(DEL,ILOP)
000005          COMMON/BIAS/EAST,YBIAS
000005          COMMON/CC/REEL(10)
000005          COMMON/GG/IOFF(20),GRID2(20,20,3),ISUB
000005          COMMON/NOS/ILAT,ILON
000005          COMMON/SCALES/XSCALE,YSCALE
000005          COMMON/  ZZ  /VLONL,VLONR,VLATL,VLATR
000005          COMMON XCNTR
          C *** ON FLATBED PLOTTER THE LAT-LON AND LOP GRID MUST BE DRAWN IN TWO
          C     CHUNKS: FROM LEFT TO CENTER  AND FROM CENTER TO RIGHT
000005          PI=2.0*ASIN(1.0)
000010          DTOR=PI/180.
000012          RTOD=180./PI
          C *** MOVE PEN TO LOWER LEFT OF MAP
000013          CALL CALPLT(5.0,3.9,-3)
000015          DO 200 I=1,2
000020          READ(5,10) VLONL,VLONR,VLATL,VLATR
000027       10 FORMAT(4F10.5)
000033          WRITE(6,11) I,VLONL,VLONR,VLATL,VLATR
000051       11 FORMAT(* GRID CHUNK NO.*I2* LON-LAT*4F10.5)
000051          REEL(1)=VLONL*DTOR
000053          REEL(2)=VLONR*DTOR
000055          REEL(3)=VLATL*DTOR
000056          REEL(4)=VLATR*DTOR
          C *** CALCULATE LAMBERT BIASES
000060          CALL LAMBRT(X,YBIAS,VLONL,VLATL)
000063          EAST=2.0*ABS(X)
000066                       CALL LAMBRT(X,Y,VLONR,VLATL)
000071          WRITE(6,12) YBIAS,EAST
000101       12 FORMAT(* BIASES N,E (METERS) *2F15.5)
          C *** TAP OUT THE CORNERS OF THIS PLOTTING REGION
000101          CALL BOXIT
000102          CALL FLAT(DEL,I)
000105          CALL FLON(DEL)
          C *** PLOT THE LOPS
000110          IF(ILOP.EQ.0) GO TO 90
000112          DO 50 JJ=1,ILOP
000114          CALL INLOP
000115          CALL GRID1
000116          CALL INTERP
000117          CALL SORT(JJ)
          C *** PRINT OUT LOPS COMPUTED
000121          PRINT 39,ISUB
000127       39 FORMAT(1H1//*  COMPUTED *I3* LOP VALUES*//)
000127          DO 43 K=1,ISUB
000132          T=GRID2(K,1,1) + 0.5
000134          KK=IFIX(T)+1
000137          PRINT 40
000142       40 FORMAT(1X,40(*-*))
000142          DO 42 L=2,KK
000145          RLE=GRID2(K,L,1)*RTOD
000151          RLA=GRID2(K,L,2)*RTOD
000155          K1=IFIX(RLE)
000156          K2=IFIX(RLA)
000160          RLE=(RLE-K1)*60.
000163          RLA=(RLA-K2)*60.
000165          PRINT 41,K1,RLE,K2,RLA,GRID2(K,L,3),IOFF(K)
000207       41 FORMAT(2(I10,1X,F8.4,1X),F20.5,I10)
000207       42 CONTINUE
000213       43 CONTINUE
000216          CALL LAMPLOT(GRID2,ISUB)
000220       50 CONTINUE
000224       90 CONTINUE
          C *** RESET ORIGIN FOR SECOND PART
000224          X2=EAST/2.0*XSCALE
000227          Y2=(YBIAS-Y)*YSCALE
000232          Y2=-Y2
000233          WRITE(6,22) X2,Y2
000242          IF(I.EQ.1) CALL CALPLT(X2,Y2,-3)
000252      200 CONTINUE
000264          RETURN
000267          END
```

Figure D-2. Continued.

```
            SUBROUTINE LAMBRT(XVAL,YVAL,XLONG,XLAT)
          C INPUT
          C       XLONG     LONGITUDE (DECIMAL DEGREES)
          C       XLAT      LATITUDE (DECIMAL DEGREES)
          C       XCNTR     CENTER MERIDIAN
          C OUTPUT
          C       XVAL      LAMBERT X IN METERS
          C       YVAL      LAMBERT Y IN METERS
000007          COMMON XCNTR
000007          DATA RBASE/9275742.099/,ECCEN/.081991889977/,YKVAL/13016802./,
                1YLVAL/0.58800002/,PI/3.141592654/
000007          XPAR=XLAT*PI/180.
000011          THETA=(YLVAL*(XCNTR-XLONG))*PI/180.
000014          EXP1=ECCEN*YLVAL/2.0
000017          FACT=ECCEN*SIN(XPAR)
000022          TERM1=((1.+FACT)/(1.-FACT))**EXP1
000032          TERM2=(1./TAN(PI/4.+XPAR/2.))**YLVAL
000044          RVAL=YKVAL*TERM2*TERM1
000046          XVAL=RVAL*SIN(THETA)
000057          YVAL=RBASE-RVAL*COS(THETA)
000067          RETURN
000067          END

                SUBROUTINE BOXIT
000002          COMMON XCNTR
000002          COMMON/ZZ/A,B,C,D
000002          COMMON/METRIC/XSC,YSC
000002          COMMON/BIAS/EAST,YBIAS
          C     BOXIT PLOTS THE CORNERS OF THE LATITUDE-LONGITUDE AREA TO
          C     ALLOW THE MAPPERS TO LINE UP FOR OVERLAID FLATBED PLOTTING
000002          Z=EAST/2.0
000004          XSCALE=1.0/XSC
000006          YSCALE=1.0/YSC
000010          WRITE(6,99) A,B,C,D,Z,XCNTR,YBIAS
000031       99 FORMAT(//*  BOXIT PARAMETERS:*/* LONGITITUDES *2F10.5/* LATITUDES
                1*2F10.5/* BIAS EAST (METERS)* F12.5/* CENTER MERIDIAN* F10.5/* BIAS
                2S NORTH(METERS) *F12.5)
000031          CALL LAMBRT(XLL,YLL,A,C)
000034          CALL LAMBRT(XLR,YLR,B,C)
000037          CALL LAMBRT(XUR,YUR,B,D)
000042          CALL LAMBRT(XUL,YUL,A,D)
000045          XLL=(XLL+Z)/XSCALE
000050          XLR=(XLR+Z)/XSCALE
000051          XUR=(XUR+Z)/XSCALE
000053          XUL=(XUL+Z)/XSCALE
000055          YLL=(YLL-YBIAS)/YSCALE
000060          YLR=(YLR-YBIAS)/YSCALE
000062          YUR=(YUR-YBIAS)/YSCALE
000064          YUL=(YUL-YBIAS)/YSCALE
000066          WRITE(6,100)XLL,YLL,XLR,YLR
000102          WRITE(6,100)XUR,YUR,XUL,YUL
000116          CALL CALPLT(0.0,0.0,-3)
000121          CALL PNTPLT(0.0,0.0,1,1)
          C *** LOWER LEFT
000124          CALL PNTPLT(XLL,YLL,1,1)
000127          CALL CALPLT(XLL,YLL,-3)
          C *** LOWER RIGHT
000132          CALL PNTPLT(XLR-XLL,YLR-YLL,1,1)
000141          CALL CALPLT(XLR-XLL,YLR-YLL,-3)
          C *** UPPER RIGHT
000150          CALL PNTPLT(XUR-XLR,YUR-YLR,1,1)
000157          CALL CALPLT(XUR-XLR,YUR-YLR,-3)
          C *** UPPER LEFT
000166          CALL PNTPLT(XUL-XUR,YUL-YUR,1,1)
000175          CALL CALPLT(XUL-XUR,YUL-YUR,-3)
          C *** LOWER LEFT AGAIN
000204          CALL CALPLT(XLL-XUL,YLL-YUL,-3)
          C *** TO THE TRUE ORIGIN
000213          CALL CALPLT(-XLL,-YLL,2)
000221          CALL CALPLT(-XLL,-YLL,-3)
000227          CALL CALPLT(0.0,0.0,-3)
000232       100 FORMAT(*  BIASES IN INCHES)...*4F10.5)
000232          RETURN
000233          END
```

Figure D-2. Continued.

```
                SUBROUTINE FLAT(DEL,II)
              C *** FLAT PLOTS LATITUDES IN WAY NECESSARY FOR THE FLATBED OPTION
              C     DEL       INCREMENT FOR PLOTTED LATITUDE AND LONGITUDE POINTS
              C               A DECIMAL PART OF A DEGREE I.E. .5 MEAN 30 MINUTES
              C     II        FLAG TO HALF OF MAP BEING PLOTTED
              C               II=1 LEFT
              C               II=2 RIGHT
000005                COMMON/ZZ/VLONL,VLONR,VLATL,VLATR
000005                COMMON/NOS/ILAT,ILON
000005                COMMON/METRIC/XSCALE,YSCALE
000005                COMMON/BIAS/EAST,YBIAS
000005                COMMON XCNTR
000005                DIMENSION TEMPLON(40),TEMPLAT(40)
000005                ILON=VLONL-VLONR+1
000011                ILAT=VLATR-VLATL+1
000015                IF(DEL.EQ.0.5) ILON=ILON*2-1
000022                IF(DEL.EQ.0.5) ILAT=ILAT*2-1
000026                ILON=ILON+1
000030                EDGE=EAST/2.0
000031                XLAT=VLATL
000033                DO 100 K=1,ILAT
000035                XLON=VLONL
000037                DO 200 I=1,ILON
000040                CALL LAMBRT(X,Y,XLON,XLAT)
000043                TEMPLON(I)=X+EDGE
000046                TEMPLAT(I)=Y-YBIAS
000050                XLON=XLON-DEL
000053              200 CONTINUE
000056                WRITE(6,250) XLAT,(TEMPLON(I),TEMPLAT(I),I=1,ILON)
000074              250 FORMAT(*  LATITUDE *F10.2/(1X,2F12.4))
              C *** SCALE VALUES
000074                DO 300 I=1,ILON
000077                TEMPLON(I)=TEMPLON(I)*XSCALE
000101                TEMPLAT(I)=TEMPLAT(I)*YSCALE
000103              300 CONTINUE
              C *** PLOT LATITUDE
000105                CALL CALPLT(TEMPLON(1),TEMPLAT(1),3)
000110                DO 400 I=2,ILON
000113                CALL CALPLT(TEMPLON(I),TEMPLAT(I),2)
000116              400 CONTINUE
              C *** LABEL LATITUDE
000122                IF(II.EQ.2)
                     1CALL NUMBER(TEMPLON(ILON)+0.5,TEMPLAT(ILON),0.5,XLAT,0.0,1)
000134                XLAT=XLAT+DEL
000136              100 CONTINUE
000140                RETURN
000141                END

                SUBROUTINE INLOP
              C *** INLOP READS IN DATA WHICH VARIES WITH EACH LOP SET TO BE PLOTTED
000002                COMMON/CC/REEL(10)
000002                COMMON/BB/OFFSET,WAVE1,WAVE2
000002                NAMELIST/INPUT2/OFFSET,WAVE1,WAVE2
000002                DIMENSION INT(10),INT1(10)
000002                PI=2.0*ASIN(1.0)
000006                READ INPUT2
000010                PRINT INPUT2
000013                DO 20 I=5,10
000015                READ 10,INT(I),INT1(I),REEL(I)
000026                PRINT 10,INT(I),INT1(I),REEL(I)
000040               20 CONTINUE
000042               10 FORMAT(I4,1X,I3,1X,F8.0)
              C       CONVERT TO RADIANS FOR LOP WORK
000042                DO 30 I=5,10
000044                REEL(I)=(FLOAT(INT(I))+FLOAT(INT1(I))/60.+REEL(I)/3600. )
000053                REEL(I)=REEL(I)*PI/180.
000056               30 CONTINUE
000060                WRITE(6,1950) (REEL(I),I=1,10)
000071             1950 FORMAT(//*  CONVERTED DATA*/(E15.5))
000071                RETURN
000072                END
```

Figure D-2. Continued.

```
                SUBROUTINE FLON(DEL)
              C     DEL       INCREMENT FOR PLOTTED LATITUDE AND LONGITUDE POINTS
              C               A DECIMAL PART OF A DEGREE I.E. .5 MEAN 30 MINUTES
000003          COMMON/ZZ/VLONL,VLONR,VLATL,VLATR
000003          COMMON/NOS/ILAT,JLON
000003          COMMON/METRIC/XSCALE,YSCALE
000003          COMMON/BIAS/EAST,YBIAS
000003          COMMON XCNTR
000003          DIMENSION TLAT(40),TLON1(40)
000003          XLON=VLONL
              C *** FLATBED USES ALL LONGITUDES IN REDUCED ARAEA
000004          ILON=JLON
000006          DO 100 K=1,ILON
000010          XLAT=VLATL
000012          DO 200 I=1,ILAT
000013          CALL LAMBRT(X,Y,XLON,XLAT)
000016          TLON1(I)=X +EAST/2.0
000022          TLAT(I)=Y-YBIAS
000024          XLAT=XLAT+DEL
000027        200 CONTINUE
000032          WRITE(6,250) XLON,(TLON1(I),TLAT(I),I=1,ILAT)
000050        250 FORMAT( * LONGITUDE *F10.2/(2F12.4))
000050          DO 400 I=1,ILAT
000053          TLAT(I)=TLAT(I)*YSCALE
000055          TLON1(I)=TLON1(I)*XSCALE
000057        400 CONTINUE
              C *** PLOT LONGITUDE
000061          CALL CALPLT(TLON1(1),TLAT(1),3)
000064          DO 500 I=2,ILAT
000067          CALL CALPLT(TLON1(I),TLAT(I),2)
000072        500 CONTINUE
000076          CALL NUMBER(TLON1(ILAT)-0.5,TLAT(ILAT)+0.5,0.5,XLON,0.0,1)
000105          XLON=XLON-DEL
000110        100 CONTINUE
              C *** CENTER MERIDIAN
000112          CENTERX=(EAST/2.0)*XSCALE
000115          CALL CALPLT(CENTERX,0.0,3)
000117          CALL LAMBRT(X,Y,XCNTR,VLATR)
000122          Y=Y-YBIAS
000124          Y=Y*YSCALE
000126          CALL CALPLT(CENTERX,Y,2)
              C *** LABEL CENTER MERIDIAN
000130          CALL NUMBER(CENTERX-0.5,Y+0.5,0.5,XCNTR,0.0,1)
000140          RETURN
000141          END

                SUBROUTINE LAMPLOT(GRID2,ISUB)
              C     LAMBERT TRANSFORM
              C     PLACES LOP IN A GRID RECTILINEAR IN UNITS NORTH X EAST
000005          COMMON/BIAS/EAST,YBIAS
000005          COMMON/METRIC/XSCALE,YSCALE
000005          COMMON XCNTR
000005          DIMENSION GRID2(20,20,3)
000005          PI=2.*ASIN(1.0)
000010          RTOD=180/PI
000012          DO 100 K=1,ISUB
000015          UB=GRID2(K,1,1)+0.5
000020          L=IFIX(UB)+1
000022          IF(L.LT.3) GO TO 100
000024          IPEN=3
000025          DO 100 M=2,L
              C *** CONVERT TO DEGREES
000027          R1=GRID2(K,M,1)*RTOD
000034          R2=GRID2(K,M,2)*RTOD
000037          CALL LAMBRT(X,Y,R2,R1)
              C *** BIAS AND SCALE THE LAMBERT VALUES
000042          X=X+EAST/2.0
000045          Y=Y-YBIAS
000047          X=X*XSCALE
000050          Y=Y*YSCALE
000052          CALL CALPLT(X,Y,IPEN)
000054          IPEN=2
000055        100 CONTINUE
000064          RETURN
000065          END
```

Figure D-2. Continued.

```
                SUBROUTINE GRID1
          C
          C        THIS SUBROUTINE COMPUTES VALUES OF LINES OF POSITION AS PHASE
          C        DIFFERENCES BETWEEN STATIONS PLUS AN OFFSET.  THE LOPES ARE
          C        STORED IN AN ARRAY GRID ALONG WITH A LATITUDE AND LONGITUDE
          C        VECTOR. NUMBER OF ELEMENTS CONTAINED IN GRID IS DETERMINED
          C        BY THE RESPECTIVE LATITUDE OR LONGITUDE RANGES ALONG WITH THEIR
          C        INCREMENT SIZE.
          C
000002            COMMON/EE/RLAT1,RLON1,RLAT2,RLON2,DIST
000002            COMMON/CC/RLONL,RLONR,RLATL,RLATR,RLTIN,RLOIN,RLT1,RLN1,RLT2,RLN2
000002            COMMON/DD/GRID(20,20)
000002            COMMON/BB/OFFSET,WAVE1,WAVE2
000002            COMMON/FF/I,J
          C
          C        COMPUTE NUMBER OF LATITUDE STEPS AND LONGITUDE STEPS
          C        AND SET APPRORIATE SUBSCRIPT LIMITS
          C
000002            RA=ABS(RLATR-RLATL)/RLTIN
000006            RA=RA+.5
000010            J=IFIX(RA)+2
000012            RB=ABS(RLONR-RLONL)/RLOIN
000016            RB=RB+.5
000020            I=IFIX(RB)+2
000022            GRID(1,1)=OFFSET
000023            IF(I.LE.20.AND.J.LE.20)GO TO 10
000034            PRINT 5
000037          5 FORMAT(* TOO MANY GRID DIVISIONS REQUIRED FOR THIS SIZE AT THE*/*
                 1 GIVEN LATITUDE OR LONGITUDE INCREMENT*)
000037            IF(I.GT.20)I=20
000043            IF(J.GT.20)J=20
000046         10 CONTINUE
          C       STEPS ACROSS LONGITUDE AND DOWN LATITUDE WILL BE INCREASINGLY
          C       NEGATIVE
          C
000046            DO 50 K=2,I
000050            RLON1=RLONL-(K-2)*RLOIN
000055            GRID(K,1)=RLON1
000057            DO 50 L=2,J
000060            RLAT1=RLATR-(L-2)*RLTIN
          C THERE WAS AN I
000065            GRID(1,L)=RLAT1
000070            RLAT2=RLT1
000072            RLON2=RLN1
          C
          C        STATION 1
          C
000073            CALL DISFUN
000074            PHA=DIST/WAVE1
          C
000076            RLAT2=RLT2
000100            RLON2=RLN2
          C
          C        STATION 2
          C
000101            CALL DISFUN
000102            PHB=DIST/WAVE2
          C
          C
          C
000104            DEPHE=PHA-PHB+OFFSET
000107            GRID(K,L)=DEPHE
000113       50   CONTINUE
000117            RETURN
000120            END
```

Figure D-2. Continued.

```
      SUBROUTINE INTERP
C
C        THIS ROUTINE TAKES GRID PRODUCED IN SUBROUTINE GRID1 AND
C        PRODUCES A 3-DIMENSIONAL ARRAY OF INTERPOLATED LATITUDES
C        AND LONGITUDES. THE VECTOR IOFF IS AN ARRAY IN WHICH INT-
C        EGER VALUES OF LOP≠S ARE FOUND.  THE SUBSCRIPT OF THIS
C        ARRAY CORRESPONDS TO THE FIRST SUBSCRIPT OF GRID2 TO LABEL
C        THE LOP≠S.  THE SECOND SUBSCRIPT OF  GRID WILL SPAN PAIRS
C        OF INTERPOLATED POINTS FOR THE LOP DESIGNATED BY THE FIRST
C        SUBSCRIPT.  THE THIRD SUBSCRIPT SPANS POINT COORDINATES
C        WITH A SUBSCRIPT OF 1 ACCESSING LATITUDE, 2 ACCESSING LONG-
C        ITUDE AND 3 REFERRING TO A COMPUTED VALUE OF AN LOP AT THE
C        LONGITUDE AND LATITUDE SPECIFIED IN SUBSCRIPTS 1 AND 2
C        ISUB IS A SUBSCRIPT LIMIT ON IOFF
C
C
C        RLAT1,RLON1,RLAT2,RLON2,DIST  ARE PARAMETERS FOR COMMUN-
C        ICATION WITH SUBROUTINE DISFUN
C
000002       COMMON/BB/OFFSET,WAVE1,WAVE2
000002       COMMON/DD/GRID(20,20)
000002       COMMON/GG/IOFF(20),GRID2(20,20,3),ISUB
000002       COMMON/FF/I,J
000002       COMMON/EE/RLAT1,RLON1,RLAT2,RLON2,DIST
000002       COMMON/CC/RLONL,RLONR,RLATL,RLATR,RLTIN,RLOIN,RLT1,RLN1,RLT2,RLN2
C
C        FIND LARGEST AND SMALLEST LOP VALUE IN GRID
C        BY SEARCH OF EDGES
C
000002       ISUB=0
000004       RA=GRID(2,2)
000006       RB=GRID(2,2)
000007       DO 10 K=2,I
000010       IF(GRID(K,2).LT.RB)RB=GRID(K,2)
000014       IF(GRID(K,2).GT.RA)RA=GRID(K,2)
000020       IF(GRID(K,J).LT.RB)RB=GRID(K,J)
000030       IF(GRID(K,J).GT.RA)RA=GRID(K,J)
000040    10 CONTINUE
C
000043       DO 20 K=2,J
000044       IF(GRID(2,K).LT.RB)RB=GRID(2,K)
000053       IF(GRID(2,K).GT.RA)RA=GRID(2,K)
000062       IF(GRID(I,K).GT.RA)RA=GRID(I,K)
000072       IF(GRID(I,K).LT.RB)RB=GRID(I,K)
000102    20 CONTINUE
000105       KA=IFIX(RA)
000106       KB=IFIX(RB)
000110       ISUB=KA-KB
C
C        CHECK FOR TOO MANY LOP VALUES
C
000111       IF(ISUB.LE.20) GO TO 25
000113       PRINT 26
000117    26 FORMAT(* TOO MANY LOP≠≠S IN THIS SPECIFIED GRID*)
000117       ISUB=20
C
C        FILL IOFF ARRAY WITH PROPER LOP NUMBERS
C
000120    25 DO 30 K=1,ISUB
000122       IOFF(K)=KB+K
000124       GRID2(K,1,1)=0.
000125       GRID2(K,1,2)=0.
000126       GRID2(K,1,3)=0.
000127    30 CONTINUE
000131       LOPTOP=KB+ISUB
C
C        NOW FOR LATITUDE INTERPOLATION (ITERATIVE) TO FORCE POINTS
C        TO INTEGER LOP COORDINATES
C
000132       DO 500 K=2,I
000134       DO 500 L=3,J
000135       M=L-1
```

Figure D-2. Continued.

```
000137          KR=IFIX(GRID(K,M))
000143          KS=IFIX(GRID(K,L))
000147          IF(KS-KR)100,500,90
000151       90 KSAV=KR
000153          KR=KS
000154          KS=KSAV
                C
                C
000155      100 KP=KR-KS
000157          DO 150 N=1,KP
000161          K2=KS+N
000163          IF(K2.GT.LOPTOP)GO TO 500
000166          A=GRID(1,M)
000171          B=GRID(K,M)
000175          C=GRID(K,L)
000200          D=GRID(1,L)
000203          RNLON=GRID(K,1)
000205          RLON2=RNLON
000206      105 RNLAT=A+(K2-B)/(C-B)*(D-A)
                C
                C          CHECK THE LATITUDE THAT WAS INTERPOLATED BY RECOMPUTATION
                C
000217          RLAT1=RLT1
000221          RLON1=RLN1
000222          RLAT2=RNLAT
000224          CALL DISFUN
000225          X=DIST
000227          RLAT1=RLT2
000230          RLON1=RLN2
000232          CALL DISFUN
000233          EEP=X/WAVE1-DIST/WAVE2+OFFSET
                C
                C          CHECK FOR CONVERGENCE TO WITHIN DETERMINED CRITERIA-IF NOT SET
                C          ONE OF THE BOUNDARIES TO THE RECENT INTERPOLATION FOR FORCING
                C          CONVERGENCE
                C
                C ***
                C     SET THAT TOLERANCE
                C ***
000240          TOL=.001
000242          TOL=.0001
000243          IF(ABS(FLOAT(K2)-EEP).LT.TOL) GO TO 140
000250          TOL=0.00001
                C
                C          DETERMINE PROPER BOUNDARY SIDE
                C
                C
000251          IF(C-B)110,140,135
000254      110 IF(K2-EEP)130,140,135
000260      120 IF(K2-EEP)135,140,130
000263      130 B=EEP
000265          A=RNLAT
000266          GO TO 105
000267   135    C=EEP
000271          D=RNLAT
000272          GO TO 105
000273      140 LSUM=K2-K8
000275          REEL=GRID2(LSUM,1,1)+.5
000277          ICNT=IFIX(REEL)
000301          ICNT=ICNT+1
                C
                C          CHECK FOR TOO MANY POINTS REQUIRED
                C
000303          IF(ICNT.LT.20) GO TO 142
000305          PRINT 222
000311   222    FORMAT(* TOO MANY POINTS FOR AN LOP, PLOT WILL NOT BE COMPLETE*)
000311          GO TO 150
000312   142    GRID2(LSUM,1,1)=ICNT
000315          GRID2(LSUM,ICNT+1,1)=RNLAT
000320          GRID2(LSUM,ICNT+1,2)=RNLON
000323          GRID2(LSUM,ICNT+1,3)=EEP
000326      150 CONTINUE
000331      500 CONTINUE
```

Figure D-2. Continued.

```
C
C        INSERT EDGES BY LONGITUDE INTERPOLATION SIMILAR TO LATITUDE
C        INTERPOLATION
C
000336         KQ=2
000337         DO 900 IL=1,2
000341         DO 750 L= 3,I
000342         M=L-1
000344         KR=IFIX(GRID(M,KQ))
000350         KS=IFIX(GRID(L,KQ))
000354         IF(KS-KR)600,750,510
000356     510 KSAV=KR
000360         KR=KS
000361         KS=KSAV
000362     600 KP=KR-KS
000364         DO 650 N=1,KP
000366         K2=KS+N
000370         IF(K2.GT.LOPTOP)GO TO 750
000373         A=GRID(M,1)
000375         B=GRID(M,KQ)
000400         C=GRID(L,KQ)
000404         D=GRID(L,1)
000405         RLAT2=GRID(1,KQ)
000410         RNLAT=RLAT2
000411     520 RNLON=A+(K2-B)/(C-B)*(D-A)
000422         RLAT1=RLT1
000424         RLON1=RLN1
000425         RLON2=RNLON
000427         CALL DISFUN
000430         X=DIST
000432         RLAT1=RLT2
000433         RLON1=RLN2
000435         CALL DISFUN
000436         EEP=X/WAVE1-DIST/WAVE2+OFFSET
C ***
C        CHECK ON TOLERANCE CRITERION
C ***
000443         IF(ABS(K2-EEP).LT.TOL) GO TO 550
000450         IF(C-B)530,550,535
000452     530 IF(K2-EEP)540,550,542
000456     535 IF(K2-EEP)542,550,540
000461     540 B=EEP
000463         A=RNLON
000464         GO TO 520
000465     542 C=EEP
000467         D=RNLON
000470         GO TO 520
000471     550 LSUM=K2-K8
000473         REEL=GRID2(LSUM,1,1)+.5
000475         ICNT=IFIX(REEL)
000477         ICNT=ICNT+1
000501         IF(ICNT.LT.20)GO TO 543
000503         PRINT 222
000507         GO TO 650
000510     543 GRID2(LSUM,1,1)=ICNT
000513         GRID2(LSUM,ICNT+1,1)=RNLAT
000516         GRID2(LSUM,ICNT+1,2)=RNLON
000521         GRID2(LSUM,ICNT+1,3)=EEP
000524     650 CONTINUE
000527     750 CONTINUE
000532         KQ=J
000533     900 CONTINUE
000535    1000 RETURN
000536         END
```

Figure D-2. Continued.

```
      SUBROUTINE SORT(II)
C
C        THIS SUBROUTINE SORTS GRID2 PRODUCED IN INTERP FOR INPUT TO A
C        PLOT ROUTINE
C
C
000003      COMMON/GG/IOFF(20),GRID2(20,20,3),ISUB
C
C *** FIND LOP WITH GREATEST NUMBER OF POINTS
C     SET KMAX = NUMBER OF POINTS IN LOP NUMBER KPT
C
000003      KMAX=1
000004      DO 10 I=1,ISUB
000006      K=IFIX(GRID2(I,1,1)+0.5)
000011      IF(K.LE.KMAX) GO TO 10
000013      KMAX=K
000014      KPT=I
000015   10 CONTINUE
C
C *** FIND MIN AND MAX LONGITUDES IN THE LOP WITH MAXIMUM POINTS
C     SET VALMAX AND VALMIN
000020      IMIN=2
000021      IMAX=2
000022      VALMAX=GRID2(KPT,IMAX,2)
000025      VALMIN=VALMAX
000026      DO 20 I=3,KMAX
000030      TEMP=GRID2(KPT,I,2)
000034      IF(TEMP.LE.VALMAX) GO TO 15
000036      VALMAX=TEMP
000037      IMAX=I
000040      GO TO 20
000040   15 IF(TEMP.GE.VALMIN) GO TO 20
000043      VALMIN=TEMP
000043      IMIN=I
000045   20 CONTINUE
C
C *** FIND THE SLOPE OF THE LOP BETWEEN THE MIN AND MAX VALIES
000050      KEY=1
000051      XNUM=GRID2(KPT,IMAX,2)-GRID2(KPT,IMIN,2)
000057      XDEN=GRID2(KPT,IMAX,1)-GRID2(KPT,IMIN,1)
000065      THETA=ATAN(XNUM/XDEN)
000071      IF(THETA.GT.1.0.OR.THETA.LT.-1.0) KEY=2
C *** IF SLOPE WITHIN 37 DEGREES OF A PARALLEL SORT ON LONGITUDE
C     KEY=1  SORT ON LATITUDE
C     KEY=2  SORT ON LONGITUDE
C
000103      DO 60 I=1,ISUB
000105      ISW=1
000106      K=IFIX(GRID2(I,1,1)+.5)
000111      IF(K.LT.2) GOTO 60
000113      MTOP=K+1
000115      DO 50 M=2,MTOP
000116      L=K-M+2
000121      IF(ISW.EQ.0)GOTO 60
000122      ISW=0
000122      DO 50 J=2,L
000124      IF(GRID2(I,J,KEY).LT.GRID2(I,J+1,KEY)) GO TO 50
000135      ISW=1
000136      R1=GRID2(I,J,1)
000140      R2=GRID2(I,J,2)
000141      R3=GRID2(I,J,3)
000143      GRID2(I,J,1)=GRID2(I,J+1,1)
000145      GRID2(I,J,2)=GRID2(I,J+1,2)
000147      GRID2(I,J,3)=GRID2(I,J+1,3)
000151      GRID2(I,J+1,1)=R1
000153      GRID2(I,J+1,2)=R2
000154      GRID2(I,J+1,3)=R3
000156   50 CONTINUE
000163   60 CONTINUE
000166      RETURN
000167      END
```

Figure D-2. Continued.

```
                SUBROUTINE DISFUN
      C
      C     THE FOLLOWING CODE COMPUTES DISTANCES BETWEEN POINTS ON THE
      C     EARTH≠S SURFACE TAKING INTO ACCOUNT CURVATURE AND VARIATION
      C     IN DISTANCES BETWEEN LINES OF LONGITUDE AT DIFFERENT LATITUDES.
      C
000002          COMMON/EE/RLAT1,RLON1,RLAT2,RLON2,DIST
      C
      C     RLAT1 AND RLON1 ARE LATITUDE AND LONGITUDE(RESPECTIVELY) OF ONE
      C     POINT AND RLAT2 AND RLON2 ARE THE LATITUDE AND LONGITUDE FOR THE
      C     SECOND POINT. THESE ARE INPUT IN RADIANS. DIST IS AN OUTPUT VAR-
      C     IABLE WHICH RETURNS THE DISTANCE COMPUTEDIN THIS ROUTINE
      C
000002          COMMON/GT/EQRAD,PORAD,FLAT,FLAT2,F2,F4,F5,F6,F7,F8,PI,TWOPI
      C
      C     THE ABOVE ARE COMPUTED PARAMETERS OR CONSTANTS USED IN THE
      C     COMPUTATIONS
      C
      C     EQRAD AND PORAD ARE THE EARTH≠S EQUATORIAL AND POLAR RADII
      C     RESPECTIVELY. FLAT IS A FLATTENING CONSTANT OF THE EARTH
      C     AND THE OTHER PARAMETERS ARE HAND COMPUTED FROM THE FIRST
      C     THREE AND INITIALIZED IN A BLOCK DATA SUBPROGRAM
      C
      C
000002          BETA1=ATAN((1.-FLAT)*TAN(RLAT1))
000011          SBETA1=SIN(BETA1)
000013          CBETA1=COS(BETA1)
000015          BETA2=ATAN((1.-FLAT)*TAN(RLAT2))
000024          SBETA2=SIN(BETA2)
000026          CBETA2=COS(BETA2)
000030          DELL=RLON1-RLON2
000032          ADELL=ABS(DELL)
000034          IF(ADELL-PI)506,505,505
000037      505 ADELL=TWOPI-ADELL
000041      506 SIDEL=SIN(ADELL)
000043          CODEL=COS(ADELL)
000045          A=SBETA1*SBETA2
000047          B=CBETA1*CBETA2
000051          COPHI=A+B*CODEL
000054          OAUT1=(SIDEL*CBETA2)*(SIDEL*CBETA2)
000056          OAUT2=(SBETA2*CBETA1-SBETA1*CBETA2*CODEL)
000062          OAUT2=OAUT2*OAUT2
000063          SIPHI=SQRT(OAUT1+OAUT2)
000067          C=B*SIDEL/SIPHI
000072          EM=1.-C*C
000074          PHI=ASIN(SIPHI)
000076          IF(COPHI)507,508,508
000100      507 PHI=PI-PHI
000102      508 PHISQ=PHI*PHI
000103          CSPHI=1./SIPHI
000105          CTPHI=COPHI/SIPHI
000107          PYSCO=SIPHI*COPHI
000110          TERM1=F7*PHI
000112          TERM2=A*(F6*SIPHI-F2*PHISQ*CSPHI)
000117          TERM3=EM*(F2*PHISQ*CTPHI-F8*(PHI+PYSCO))
000125          TERM4=A*A*F2*PYSCO
000130          TERM5=EM*EM*(F5*(PHI+PYSCO)-F2*PHISQ*CTPHI-F4*PYSCO*COPHI*COPHI)
000141          TERM6=A*EM*F2*(PHISQ*CSPHI+PYSCO*COPHI)
000147          DIST=PORAD*(TERM1+TERM2+TERM3-TERM4+TERM5+TERM6)
000157          RETURN
000160          END

                BLOCK DATA
      C
      C     PARAMETERS FOR USE BY SUBROUTINE DISFUN ARE INITIALIZED
      C
000002          COMMON/GT/EQRAD,PORAD,FLAT,FLAT2,F2,F4,F5,F6,F7,F8,PI,TWOPI
000002          DATA EQRAD,PORAD,FLAT/6378166.,6356784.283,3.352329869E-03/
000002          DATA FLAT2,F2,F4,F5,F6,F7,F8,PI,TWOPI/1.123811555E-05,
               *5.61905775E-06,1.404764444E-06,7.023822219E-07,3.363567985E-03,
               *1.003363567985,1.681783993E-03,3.141592654,6.283185308/
000002          END
```

Figure D-2. Continued.

page 184

APPENDIX E

ADAPTATION OF INTEL MACRO ASSEMBLER FOR USE ON CDC-6600

The following modifications were made to the MNTEL MACRO ASSEMBLER (ref. 2) in order to execute on the CDC-6600 at Langley Research Center (LRC).

(1) The symbol table size was increased from 1000 to 2000 words with the following changes:

Original Version	Modified Version	Occurrence
SYMTAB (1000)	SYMTAB (2000)	17 places in program
FREMEM (1000)	FREMEM (2000)	BLOCK DATA
SYMSIZ (1000)	SYMSIZ (2000)	BLOCK DATA

(2) The character set defined in IOTRAN array of the BLOCK DATA Subroutine was changed to incorporate only certain CDC BCD characters which are listed below. All other characters in the original array were changed to blanks. Note the semicolon is represented in octal due to lack of that character on the keyboard.

blank					
$	0	:	F	P	Z
(	1	;(775555···5B)	G	Q	
)	2	<	H	R	
*	3	+	I	S	
+	4	>	J	T	
,	5	A	K	U	
-	6	B	L	V	
.	7	C	M	W	
/	8	D	N	X	
	9	E	O	Y	

(3) To eliminate page overflow, a variable, IQLEN, was assigned to limit the number of lines/page on the Assembler output listing. To run at LRC this limit was set to 40.

Old Version	New Version	Occurrence
IF(CONTRL(CC).EQ.51)	IF(CONTRL(CC).EQ.IQLEN)	10637000

where IQLEN is defined in BLOCK DATA as 40.

(4) To execute in batch mode, modifications had to be made to the I/O unit number assignments. Teletype Output (TTYO) had to be assigned a unique unit number TTYO/6/ in BLOCK DATA Subroutine.

(5) An error was found in Subroutine ROMAR. The limit of the DO LOOP which packs code into one word was changed from 20 to 16.

Old Version	New Version	Occurrence
DO 100 I=1,20,4	DO 100 I=1,16,4	13085150

(6) A Program Card is needed to define I/O files.

```
PROGRAM MCS4 (TAPE20, OUTPUT, TAPE23, TAPE22,
.TAPE21, INPUT, TAPE5=INPUT, TAPE6=OUTPUT)
```

APPENDIX F

USERS MANUAL FOR A SIMULATOR PROGRAM WRITTEN FOR THE INTEL 4004 MICROPROCESSOR, SIM4, VERS. 1, REV. B

F.1 INTRODUCTION

SIM4 is a digital computer program written by RTI personnel in FORTRAN to simulate the Intel 4004 Microprocessor. The simulator provides for a full complement of RAM (read and write memory), ROM (read-only memory) and associated input and/or output ports. The user should be familiar with the 4004 hardware configuration (ref. 18). SIM4 may be executed in either batch or interactive mode and requires a machine word of at least 32-bits. Approximately 100K words of core are used.

The following sections describe the use of the SIM 4 program. Section 2 describes program flow and control for use in both the batch and interactive modes. Section 3 describes control statement and object code inputs to SIM4 as well as message type outputs. Section 4 describes input/output over RAM/ROM ports and other inputs associated with simulated execution of a microprocessor program. Section 5 describes data files associated with SIM4 execution. Sections 6 through 8 are included to provide samples of control statements, input object code, and SIM4 dump output. Section 9 describes an application of SIM4 and the associated input/output handlers used during simulation runs. Flowcharts of all simulator routines are provided in Section 10.

One input required to the simulator is object code produced by the Intel Cross-Assembler Version 2.2, MAC4 (ref 19). The user should have knowledge of and access to this assembler program and be familiar with the 4004 Assembler instructions (ref. 20).

This document is primarily for the use of SIM4 on the CDC 6000 series computers at Langley Research Center Computer Complex. The user should be familiar with the method of submitting jobs at LRC Computer Complex. SIM4 and MAC4 are stored on data cell and readily available.

F.2 PROGRAM FLOW AND CONTROL

The relationship between the cross assembler, MAC4, and the 4004 simulator, SIM4, is illustrated in Figure F-1. Inputs to the assembler include 4004 assembly instructions and cross-assembler control statements. Outputs from the assembler include a source list of the microprogram with associated object code and assembly errors, an object file in either hexadecimal or BNF format* and an object file in packed decimal format. The latter file provides object code in a form suitable for use by SIM4. It is, therefore, an input file to the simulator along with SIM4 control statements. Outputs from SIM4 include error messages and dumps.

The cross-assembler and 4004 simulator may execute in either batch or interactive mode. The user must supply the appropriate system control cards and assign physical devices for the data files. Figure F-2 illustrates the program flow in the batch mode. Figure F-3 illustrates program flow in the interactive mode. The device assignments are for illustration and do not imply the best arrangement for all users.

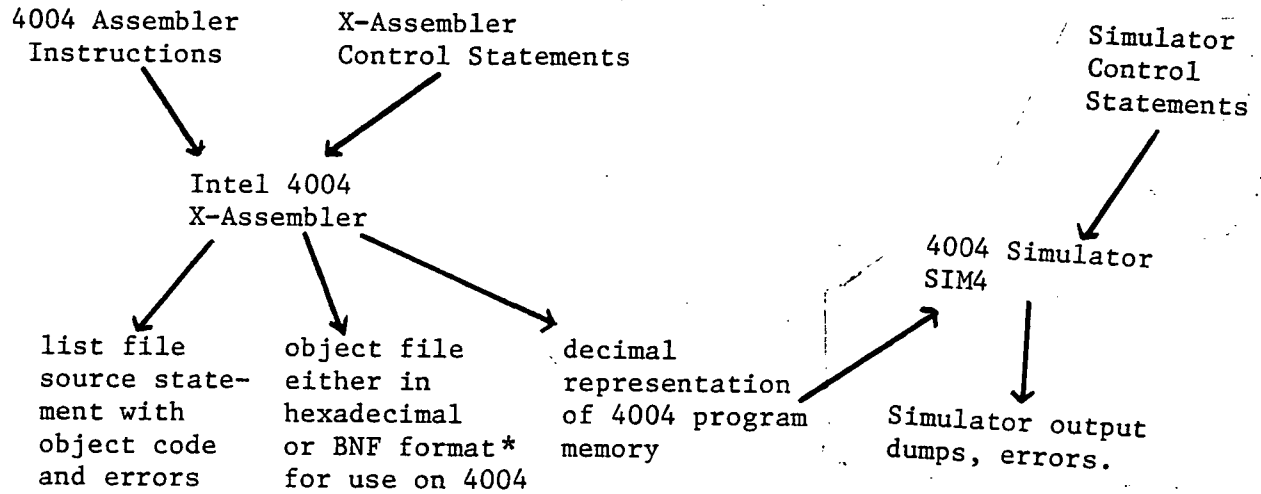
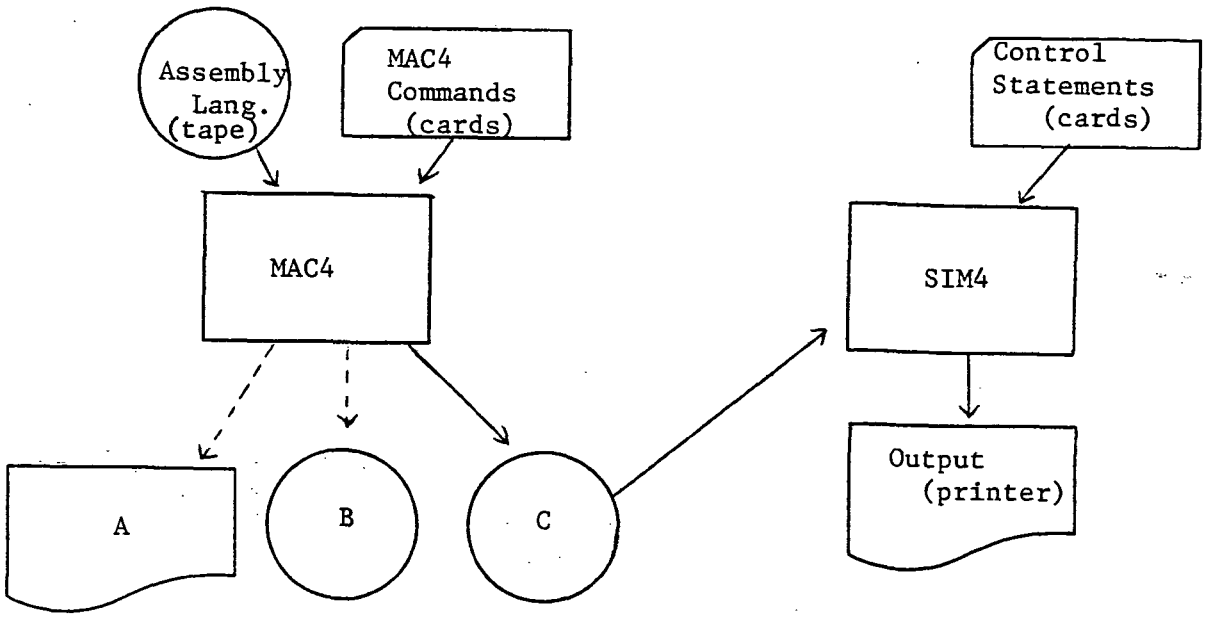


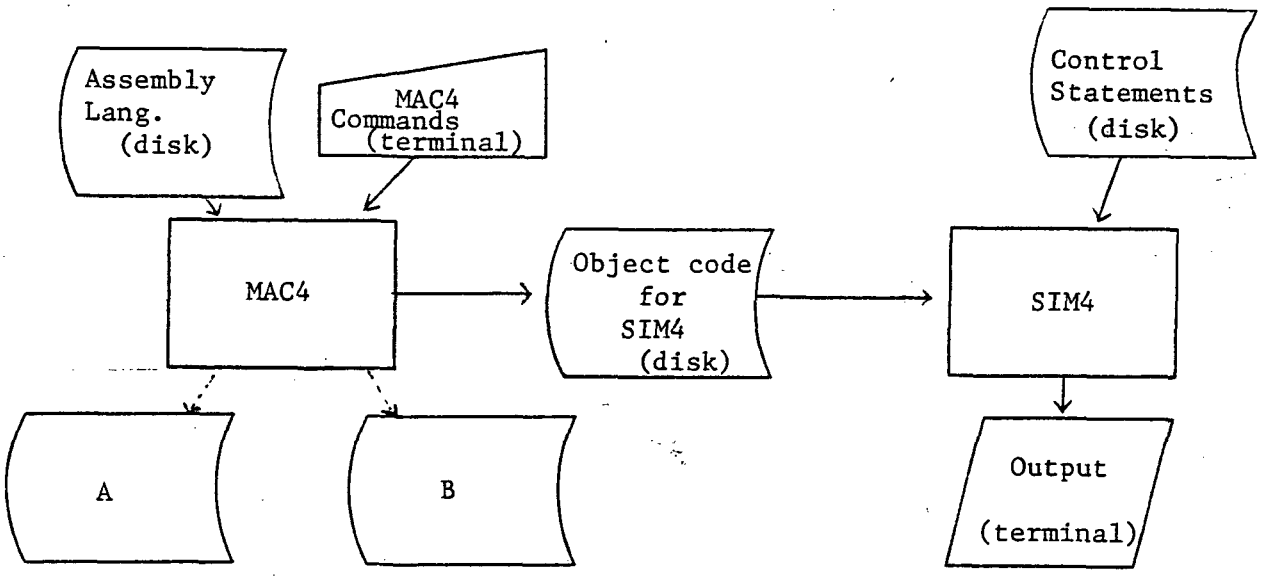
Figure F-1. Data Flow

*BNF format definition see Ref. 2.



A - List file: source w/object code and errors, optional (printer)
B - Object file in hex or BNF format, optional (tape)
C - Object code for SIM4 (tape)

Figure F-2. Program Flow in Batch Mode



A - List file: source w/object code and errors, optional (disk)
B - Object file in hexadecimal or BNF format, optional (disk)

Figure F-3. Program Flow in Interactive Mode

F.3 SIMULATOR I/O

In application of SIM4, certain inputs in the form of control statements with associated parameters are available to the user. These are supplied as needed by the user along with the cross-assembler output which is the object code input to SIM4. Subsection F.3.1 describes these inputs. Outputs include error messages and dumps of RAM, ROM and internal registers and are described in subsection F.3.2.

F.3.1 Inputs

F.3.1.1 Control statements: format (80R1). - SIM4 is controlled by one or more statements of up to 80 characters in length beginning with an alphabetic character in the first column. These statements are interpreted before the simulator starts processing microprocessor instructions. They provide the simulator information such as program (ROM) and data (RAM) memory configuration and start/halt points. They also allow the user to specify a dump of ROM, RAM and/or internal registers, modify program memory and initialize data memory prior to execution. A special control statement is reserved for specific user application and requires the user supplied subroutine, ECOM (see subsection F.4.5). The types of control statements are enumerated below and a sample is given in Section F.6.

F.3.1.1.1 Halt:

COL.	1	2	3	4	5	6	.	.	.	80
	H	a	a	a	←——— anything ———					→

SIM4 will halt after the instruction at the 3 hexadecimal digit address "a a a"is executed. Columns 5 through 80 may contain anything. (If the program is to halt after a 2 byte instruction, the address of its first byte must be used in the "H" statement).

F.3.1.1.2 Set time limit:

COL.	1	2	3	4	5	6	7	80
	T	t	t	. . .	t	b	← anything	→

SIM4 will halt if running time exceeds the decimal number"t t t ... t". The number must begin with a non-blank column 2 and is terminated by a blank column b. The microprocessor program running time is measured by an integer variable TIME which is initialized at 0 and is incremented by 1 for each 10.8 microsecond instruction cycle. This variable is printed out at

the end of the simulation and at any break points. It is also passed as an argument to the user-supplied I/O routines, and is also accessible in the COMMON block TCOM. If no "T" control statement is given, the program stops at a default value of TIME = 1000000 (4004 microprocessor actual running time of 10.8 seconds).

Note: It is advisable to use both "H" and "T" statements to prevent the simulation from using excessive computer time in case of unexpected behavior of the microprocessor program.

F.3.1.1.3 Set program counter:

COL.	1	2	3	4	5	80
	I	a	a	a	← anything →	

SIM4 will begin with the program counter set to the 3 hexadecimal digit address"a a a". If no "I" statement is given, the simulation starts at location 000.

F.3.1.1.4 Stop at first BBL instruction:

COL.	1	2	3	4	80
	S	← anything →			

If this control statement is given, SIM4 will stop at the first BBL instruction encountered (unless the program counter limit or time limits are reached first). This option can be used to check out one microprocessor subroutine.

F.3.1.1.5 Dump internal registers:

COL.	1	2	3	4	80
	R	← anything →			

If this control statement is given, all the internal registers of the microprocessor cpu chip will be dumped at the end of the simulation and any specified break points (see "B" statement below).

F.3.1.1.6 Dump program memory:

COL.	1	2	3	4	80
	P	← anything →			

If this control statement is given, those program memory chips present in the specified configuration (see "Y" statement below) are dumped at the end of the simulation and at any specified break points (see "B" statement below).

F.3.1.1.7 Dump data memory:

COL.	1	2	3	4											80
	D	←	—	—	—	anything	—	—	—	—	—	—	—	—	→

If this control statement is given, those data memory chips present in the specified configuration (see "Z" statement below) are dumped at the end of the simulation and at any specified break points (see "B" statement below).

F.3.1.1.8 Special user command:

COL.	1	2	3	4	5										80
	E	←	—	—	—	user specified	—	—	—	—	—	—	—	—	→

This statement is reserved for user's specific application. It is intended to provide the user a method of establishing a data input file, prior to execution of microcode, to be accessed later with a RDR instruction. If this control statement is utilized, subroutine ECOM must be supplied by the user.

F.3.1.1.9 Change program memory:

COL.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	80
	C	a	a	a	d	d	x	a	a	a	d	d	x		

This control statement allows words in program memory to be changed after the simulator file from MAC4 has been read in. This allows minor errors to be corrected without reassembling the microprocessor program. Columns 2, 3, and 4 contain a three hexadecimal digit address and columns 5 and 6 contain two hexadecimal digits giving its new contents. Column 7 may contain anything. A second address-data pair may begin in column 8 if desired, a third in column 14, etc. The statement is terminated by a blank in the position of the first digit of an address. As many "C" statements may be used as desired. If an address refers to a program memory chip not in the configuration specified by a "Y" statement (assuming the "Y" statement has already been processed), the associated data will be ignored and a warning message given. The remainder of the statement, if any, will still be processed.

F.3.1.1.10 Initialize data memory:

COL.	1	2	3	4	5	6	7	8	9	10	11	12	13	80
	A	a	a	a	d	x	a	a	a	d	x			

This statement allows main data memory characters to be initialized before the simulation begins. (The default contents of data memory before simulation

begins is all zeros.) Columns 2, 3, and 4 contain a 3 hexadecimal digit address and column 5 contains 1 hexadecimal digit which is entered into this data memory location. Column 6 may contain anything. A second address-data pair may begin in column 7 if desired, a third in column 12, etc. The statement is terminated by a blank in the position of the first digit of an address. As many "A" statements may be used as desired. The first digit of an address corresponds to the RAM bank number, which is loaded into the command register by the microprocessor DCL instruction. Only digits between 0 and 7 can be valid. The second and third digits of an address correspond to the byte passed by the microprocessor SRC instruction. Valid addresses are further restricted by the data memory configuration specified by the "Z" statement (assuming the "Z" statement preceeds the "A" statement). If an address is not valid, the associated data will be ignored and a warning message given. The remainder of the statement, if any, will still be processed.

F.3.1.1.11 Initialize data memory status characters:

COL.	1	2	3	4	5	6	7	8	9	10	11	12	80
	X	a	a	a	d	x	a	a	a	d	x		

This statement allows data memory status characters to be initialized before the simulation begins. Its format is identical to that of the "A" statement, with the exception that the third digit of an address specifies the status character number: 0, 1, 2 or 3. The first 2 address digits specify bank, chip and register numbers as in the "A" statement.

F.3.1.1.12 Set break point address:

COL.	1	2	3	4	5	6	7	8	9	10	80
	B	a	a	a	x	a	a	a	x		

This statement sets break point addresses, i.e., addresses at which any dumps called for by "R", "P" or "D" statements will occur in addition to occurrence at the end of the simulation. Dumps will occur after each execution of an instruction whose location is specified as a break point. (If a break point is to be placed at a 2 byte instruction, the address of its first byte must be given in the "B" statement.) Addresses must be given as three hexadecimal digits starting in column 2, with one column containing anything separating addresses. A blank in the position of the first digit of an address terminates the statement. More than one "B" statement may be used, but only the first 20 break points will be accepted. Any more will be ignored.

F.3.1.1.13 Set program memory configuration:

COL.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	80
	Y	d	d	d														d←anything→	

This statement sets the program memory configuration. Each of columns 2 through 17 corresponds to 256 bytes of program memory. A"0"or blank specifies this 256 byte block is writable program memory, a"1"specifies a block of ROM, and a "2" specifies that the corresponding block is absent. If no "Y" statement is given, all 16 possible 256 byte blocks are assumed to be present and writable. If a 4004 microprocessor program attempts to access a block of program memory not in the specified configuration, the simulation is terminated and an error message given. If program memory dumps are requested by a "P" statement, only those 256 byte blocks indicated present are dumped. (The distinction between writable program memory and ROM is made for the purpose of checking the WPM (write program memory) instruction. This instruction has <u>not</u> yet been implemented in SIM4 but can be included at a later date).

F.3.1.1.14 Set data memory configuration:

COL.	1	2	3	4	5	30	31	32	33	34	35	80
	Z	d	d	d	d	d	d	d	d	←—	anything —→	

This statement sets the data memory configuration. Each of columns 2 through 33 corresponds to one 4002 RAM chip. (Columns 2, 3, 4 and 5 correspond to chips 0, 1, 2 and 3 of bank 0; columns 6, 7, 8 and 9 to chips 0, 1, 2 and 3 of bank 1, etc.) A "0" or blank in a given column specifies that the chip is present. Any other character specifies the chip is absent. If no "Z" statement is given, all of the maximum complement of 32-4002 chips are assumed to be present. If a microprocessor program attempts to write, read, or send data to an output port of a nonexistent 4002 chip, the simulation is terminated and an error message given. If data memory dumps are requested by a "D" statement, only those chips indicated present are dumped.

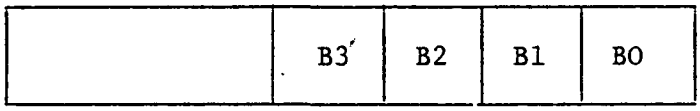
F.3.1.1.15 Go. Begin execution:

COL.	1	2	3	4	80
	G	←————	anything	———→	

This statement starts the simulation and <u>must always be present</u> as the <u>last control statement</u>.

F.3.1.2 Object code: format (1X,4(1H ,I12)).— The cross assembler, MAC4, will produce the object code of a microprocessor program in decimal representation when $Z=1 option is specified. (see Ref. 19) This file contains exactly 256 records with each record, of 4I12 format, containing four decimal integers. Each integer represents four 8-bit microprocessor instructions packed into one word of at least 32 bits.[1]

Example:



Word = B3*256**3 BO represents an 8-bit instruction residing in location
+B2*256**2 Y of memory
+B1*256 B1
+BO B2
B3 represents an 8-bit instruction residing in location
Y+4 of memory

These instructions are read into ROM memory array for later execution. Sample input is provided in Section F.7.

F.3.2 Outputs

F.3.2.1 Errors.— This section is broken down into three categories: control statement errors, informative messages and error messages.

F.3.2.1.1 Control statement error: A control statement error is generated during interpretation of a control statement. It will not terminate execution, however, parts of the statement will be ignored.

Message	Description
INVALID CONTROL STATEMENT XX XX	An undefined command, invalid address, negative time limit, negative program counter, negative memory data or PMCFG is not equal to 1, 2, or 3. Remainder of control statement is ignored.
INVALID ADDRESS STARTING IN COL. XX OF CONTROL STATEMENT XX XX	A reference was made to a non-existent 4001 ROM or 4002 RAM chip. Only that reference is ignored.

[1] If a 32 bit word machine is used, provision must be made in the simulator to deal with 2's complement.

F.3.2.1.2 Informative messages: Informative messages occur during actual execution of the microprocessor instructions. They do not terminate execution.

Message	Description
UNDEFINED INSTRUCTION - INTERPRETED AS NOP	An instruction was encountered that has not been defined -- execution continues
WPM NOT YET IMPREMENTED IN SIMULATOR	Message arises if WPM is encountered--instruction is ignored--execution continues.

F.3.2.1.3 Error messages: The following error messages cause the termination of program execution.

Message	Description
NON-EXISTENT PROGRAM MEMORY ACCESSED	A reference is made to a non-existent 4001 ROM chip
NON-EXISTENT DATA MEMORY ACCESSED	A reference is made to a non-existent 4002 RAM chip
OUTPUT SENT TO NON-EXISTENT RAM PORT	A reference is made to a non-existent RAM 4002 chip
BBL INSTRUCTION ENCOUNTERED	Simulator is terminated on BBL instruction if BBL flag is set

F.3.2.2 Dumps.— A SIM4 control statement of D, P, or R will provide dumps of RAM, ROM and internal registers. Output of dumps will occur at any given break point (see section F.3.1.1.12) and at termination of execution. Examples of each type of dump are given in Section F.8.

F.4 PROGRAM I/O

Users may input data over a ROM input port or output data over a ROM or RAM output port. Since the hardware connected to each port is unique to each system, the I/O routines ROMI, ROMO, RAMO, and UTEST that simulate the I/O hardware must be user supplied.

An additional subroutine, ECOM, not intended to simulate I/O hardware, deciphers the user's E control statement. Each routine must be supplied by the user, even if their use is not required, in order to satisfy linkage. (Dummy routines may be used if a particular I/O function is not used in a simulation.) Descriptions of these subroutines follow:

F.4.1 RAM Port Output

SUBROUTINE RAMO (IPORT, IACC, ITIME). When a WMP instruction is executed, the RAM port number is calculated from the contents of the command and SRC registers. If this port corresponds to a 4002 chip which exists in the specified data memory configuration, the above subroutine is called with the port number (0 through 31), accumulator contents, and integer TIME variable passed as arguments.

F.4.2 ROM Port Output

SUBROUTINE ROMO (IPORT, IACC, ITIME). When a WRR instruction is executed, the ROM port number is calculated from the contents of the SRC register, and the above subroutine is called. The port number (0 through 15), accumulator contents, and integer TIME variable are passed as arguments. (The port number is not checked against the program memory configuration, since I/O ports do not have to be associated with ROM chips when the 4008 - 4009 chips are used.)

F.4.3 ROM Port Input

SUBROUTINE ROMI (IPORT, IACC, ITIME). When an RDR instruction is executed, the ROM port number is calculated from the contents of the SRC register, and the above subroutine is called. The port number (0 through

15) and integer time variable are passed to the subroutine, and the accumulator is loaded with the number passed from the subroutine to the main program as the second variable (IACC). This variable must not fall outside the range of 0 through 15.

F.4.4 Test Input

SUBROUTINE UTEST (TESTF). When a JCN instruction is executed with condition bit C4 set this routine is called to supply the logical variable TESTF to the simulation.

F.4.5 User Specified Input

SUBROUTINE ECOM. When an E control statement is issued, this routine is called to decode the remainder of the statement. It may set up an input file to be accessed later by a RDR instruction.

Section F.9 describes an I/O routine written for simulation of a particular microprocessor application and is included as an illustrative example.

Note: Most major variables of SIM4 are in labeled common. If one of the above user-supplied subroutines requires a variable not passed in the call statement, it may be obtained by including the appropriate common block in the subroutine.

F.5 SIM4 DATA FILES

The following is a list of I/O logical unit numbers used by SIM4 along with suggested physical device assignment. There are three input unit numbers. Each corresponds to a specific input data file. Only one output unit OUN1 is currently utilized by SIM4. However, OUN2 is available if data is to be routed to two physical devices (i.e. printer and disk).

File Description	Symbolic Device Name	Current Logical Unit	Possible Physical Device
control statements	IUN1	20	cards, disk, terminal
object code	IUN2	21	disk, tape
memory port input	IUN3	10	cards, terminal
output file 1	OUN1	25	terminal, printer
output file 2	OUN2	26	disk, tape

F.6 EXAMPLE OF CONTROL STATEMENTS FOR 4004 SIMULATOR

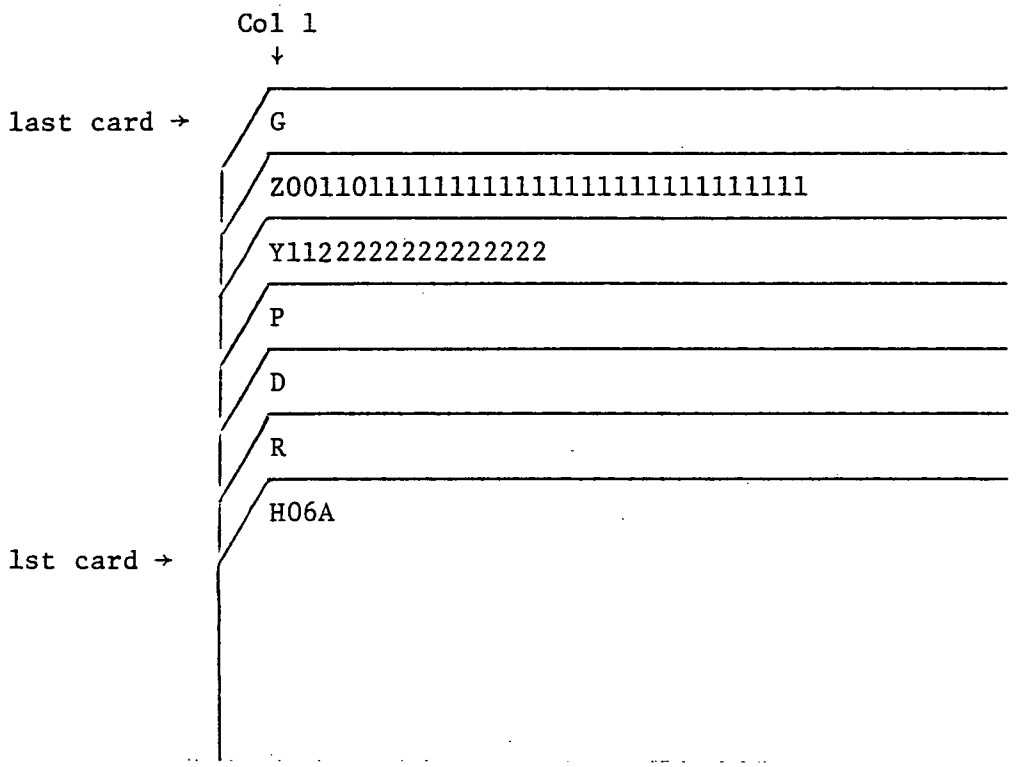


Figure F-4. Example of Control Statements for SIM4.

Figure F-4 illustrates a set of commands for the 4004 Microprocessor Simulator. A brief explanation follows:

H06A	Halt after executing instruction in location 06A.
R	Dump contents of internal registers at the end of each break point and at the end of execution.
D	Dump data memory that is indicated present by the configuration array at each break point and at the end of execution.
P	Dump program memory that is indicated present by the configuration array at each break point and at the end of execution.

Y1122222222222222	Set program memory configuration (16 digits). Each digit refers to one block of memory. 1 = ROM, 2 = Absent, 0 = writable program memory.
Z00110111.....111	Set data memory configuration (32 digits). Each digit corresponds to a 4002 chip. A blank or 0 indicates chip is present. Anything else indicates chip is not present.
G	Start execution

F.7 SAMPLE INPUT OBJECT CODE

The Intel Cross-assembler MAC4 can be used to produce a file suitable for use as input to the SIM4 simulator. This is accomplished by specifying a $Z=1 in the control statements to MAC4 (Ref. 19). Each file will have 256 lines or records with each record having 4 words (1024 words total). If the microprogram code does not require the entire 1024 words, then the remaining words are filled with zeros. Fig. B-1 provides a sample listing of four columns of records which represents a typical SIM4 object file.

578822353	4074775296	551093616	554705536
4227397856	2101104	539234338	602481137
3760585707	494232417	270663712	569450532
636167145	1969447392	47	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
.	.	.	.
.	.	.	.
.	.	.	.

F.8 SAMPLE DUMPS

In application of the 4004 microprocessor simulator, the user may call for dumps of data RAM memory, program ROM memory and/or cpu registers with the appropriate control card commands. The following sections describe these dumps.

F.8.1 RAM Memory Dump

With a D control statement (sec. F.3.1.1.7) to the simulator, a dump of any active data RAM chips will occur at all break points (sec. F.3.1.1.12) and at end of execution. In this example, the active RAM chips are chip 0 and 1 in bank 0 and chip 0 in bank 1. The data RAM configuration must have been set with a Z control statement of the form Z001101111 ....,..... 111 (see sec. F.3.1.1.14).

```
ODATA RAM DUMP
OBANK  0   CHIP  0
 DATA CHARACTERS:
    9   1   0   5   2   6   7   5   3   0   0   0   0   0   0   0
    1   3   5   3   2   7   6   4   0   0   0   0   0   0   0   0
    0   5   5   8   4   3   4   0   4   0   0   0   0   0   0   0
    8   8   4   1   0   9   0   1   3   0   0   0   0   0   0   0

OSTATUS CHARACTERS:
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
OBANK  0   CHIP  1
 DATA CHARACTERS:
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

OSTATUS CHARACTERS:
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
OBANK  1   CHIP  0
 DATA CHARACTERS:
    A   D   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

OSTATUS CHARACTERS:
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

F.8.2 Internal Register Dump

With an R control statement (see sec. F.3.1.1.5) to the simulator, a dump of the microprocessor cpu registers will occur at all break points (see sec. F.3.1.1.12) and at end of execution.*

```
04004 REGISTER DUMP
0     ACC   CARRY  STK(1)   STK(2)   STK(3)   STPTR
        3       1     018      131      18A       2
0      PC     OPR     OPA    COMRG     SRCA
      131       C       3        0       20
0INDEX REGISTERS:
  0:   1   A
  2:   3   0
  4:   0   0
  6:   3   0
  8:   2   0
 10:   B   0
 12:   1   D
 14:   0   2
```

* Explanation of register notations are given below:

ACC	Accumulator
CARRY	Carry bit
STK(1), STK(2), STK(3)	3 stack registers
STPTR	Stack pointer
PC	Program counter
OPR, OPA	Upper and lower instruction register (operator, operand)
COMRG	Command register (selects data RAM bank)
SRCA	Address register loaded by SRC instruction

For further register definitions see ref. 18 and 19.

F.8.3 ROM Memory Dump

With a P control statement (see sec. F.3.1.1.6) to the simulator, a dump of active program memory chips occurs at all break points (see sec. F.3.1.1.12) and at end of execution. In this sample, the active ROM chips are 0 and 1. The ROM configuration must have been set with a Y control statement (see sec. F.3.1.1.13) of the form Y11222 ..... 2.

```
OPROGRAM MEMORY DUMP
OCHIP  0
    2E  00  20  00  22  00  24  00  50  6C  20  10  22  10  24  10
    50  6C  D0  51  5F  D2  51  5F  D4  51  5F  28  30  26  00  29
    E9  27  E4  69  29  E9  27  E5  69  29  E9  27  E6  69  29  E9
    27  E7  27  EC  29  E7  27  ED  29  E6  27  EE  29  E5  27  EF
    29  E4  28  00  29  D8  BD  F0  E2  19  4D  51  9D  F0  E2  11
    53  40  55  51  9D  D1  E2  19  5B  40  5D  51  9D  D1  E2  11
    63  51  9D  00  D5  51  5F  BD  51  9D  40  6A  28  0A  23  EA
    BA  AA  F1  99  12  7E  B1  F8  B1  21  BA  E0  78  6E  21  E9
    25  E0  65  71  7E  A5  14  8D  F0  25  E0  75  89  C0  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
OCHIP  1
    24  00  26  10  22  20  F0  B8  25  E9  27  EB  FB  23  E0  63
    65  67  78  08  51  78  BD  51  9D  C0  24  00  26  10  22  30
    F0  FA  F9  27  E8  F1  25  EB  FB  23  E0  65  67  73  22  51
    88  BD  51  9D  C0  24  00  26  10  28  40  25  E9  B0  27  E9
    B1  51  4B  29  B0  E0  65  67  79  3B  C4  22  0B  D0  B0  F5
    80  73  55  41  5E  F6  B2  B1  F5  B1  F6  82  41  4D  C0  FC
    F2  14  73  F8  20  6E  F1  81  B1  1A  6C  60  30  31  00  6A
    1A  35  6A  D6  BD  51  9D  C0  26  20  28  40  B7  F8  B7  27
    E9  29  E1  A7  1C  7D  D5  BD  51  9D  C2  26  30  28  20  D1
    BC  B7  F1  9C  B7  27  E9  29  E2  A7  1C  92  C3  BD  F4  BD
    D1  FD  2F  AD  E0  F0  FD  BF  F2  BF  F7  14  AE  6E  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
```

F.9 DOCUMENTATION OF SIM4 I/O ROUTINES SIMULATING AN OMEGA NAVIGATION RECEIVER

Microprocessor-based OMEGA VLF Navigation receiver input/output hardware was simulated using subroutines described in this Section. These subroutines were used in conjunction with the SIM4 program to provide a complete simulation of the receiver. All I/O actions in the receiver were initiated by enable codes sent over ROM output ports 0 and 1. These codes may either request information to be placed on the data bus or direct output to a specific device. The enable codes for this particular simulation are given in Table F-1.

Figure F-5 provides a block diagram description of the I/O routines and how they relate to each other. The ECOM subroutine handles any E control statements (see sec. F.3.1.1.8) and establishes a data set that is in common with ROMO. ROMI and ROMO handle any input/output over ROM ports and communicate by a common variable. Subroutine READI converts user input of rotary switch position and keyboard entries into internal numeric code. This allows the use of symbolic notation for front panel entries made by the user.

The receiver's front panel illustrated in Figure F-6, provides rotary switch and keyboard inputs to the microprocessor and a display register for output. The rotary switch on the front panel designates whether data is to be input from the keyboard or output via the display registers. If set to an input position (1, 2 or 3), the receiver will accept a specific sequence of keyboard entries made by the user. If the rotary switch is set to an output position, data will be sent to the display on the front panel. What is displayed depends on the current rotary switch position. For example, if the switch points to RANGE, N.MI., then the value displayed would indicate the users range to destination. Also on the front panel are indicator lights which come on if the user's current position is off course or at destination. Associated with each rotary switch position and keyboard entry are specific codes given in Table F-2.

TABLE F-1. ENABLE CODES

MSB	LSB	FUNCTION		COMMENTS
0	0	LS Byte 10.2		No Action
0	1	MS Byte 10.2		No Action
0	2	Load MS and LS Bytes into Position		No Action
0	3	LS Byte 13.6		No Action
0	4	MS Byte 13.6		No Action
0	5	Load		No Action
0	6	Inactive		
0	7	Inactive		
0	8	Inactive		
0	9	Reset Phase Detector		No Action
0	A	LS Digit	DISPLAY...Only 3 char. are needed; therefore, 0-D may be ignored	
0	B	Digit		
0	C	Digit		
0	D	MS Digit		
0	E	Rotary Switch		Set up to Input Current Rotary Position
1	0	Pos. LSB	10.2 Phase Errors	Read from Table of Phase Errors (Currently all 1)
1	1	Pos. MSB		
1	2	Neg. LSB		
1	3	Neg. MSB		
1	4	Pos. LSB	13.6 Phase Errors	Look up in Table of Errors (Currently all equal 1)
1	5	Pos. MSB		
1	6	Neg. LSB		
1	7	Neg. MSB		
1	8	Inactive		
1	9	Inactive		
1	A	Inactive		
1	B	Inactive		
1	C	KB Ready Bit Reset to 0		
1	D	KB Data Ready?		
1	E	KB Data Enable		Set up to Input Next Keyboard Character

TABLE F-1. CONCLUDED

	MSB	LSB	FUNCTION	COMMENTS
	1	F	Inactive	
*	2	0	PDC Control	Look up in Table Return Value
*	2	1	Left Light	
*	2	2	Right Light	
*	2	3	Destination Light	
	F	X	Disable Code	No Action

*Latched Output

Data output over ROMO PORT#2

MSB of enable code over ROMO PORT #1

LSB of enable code over ROMO PORT #0

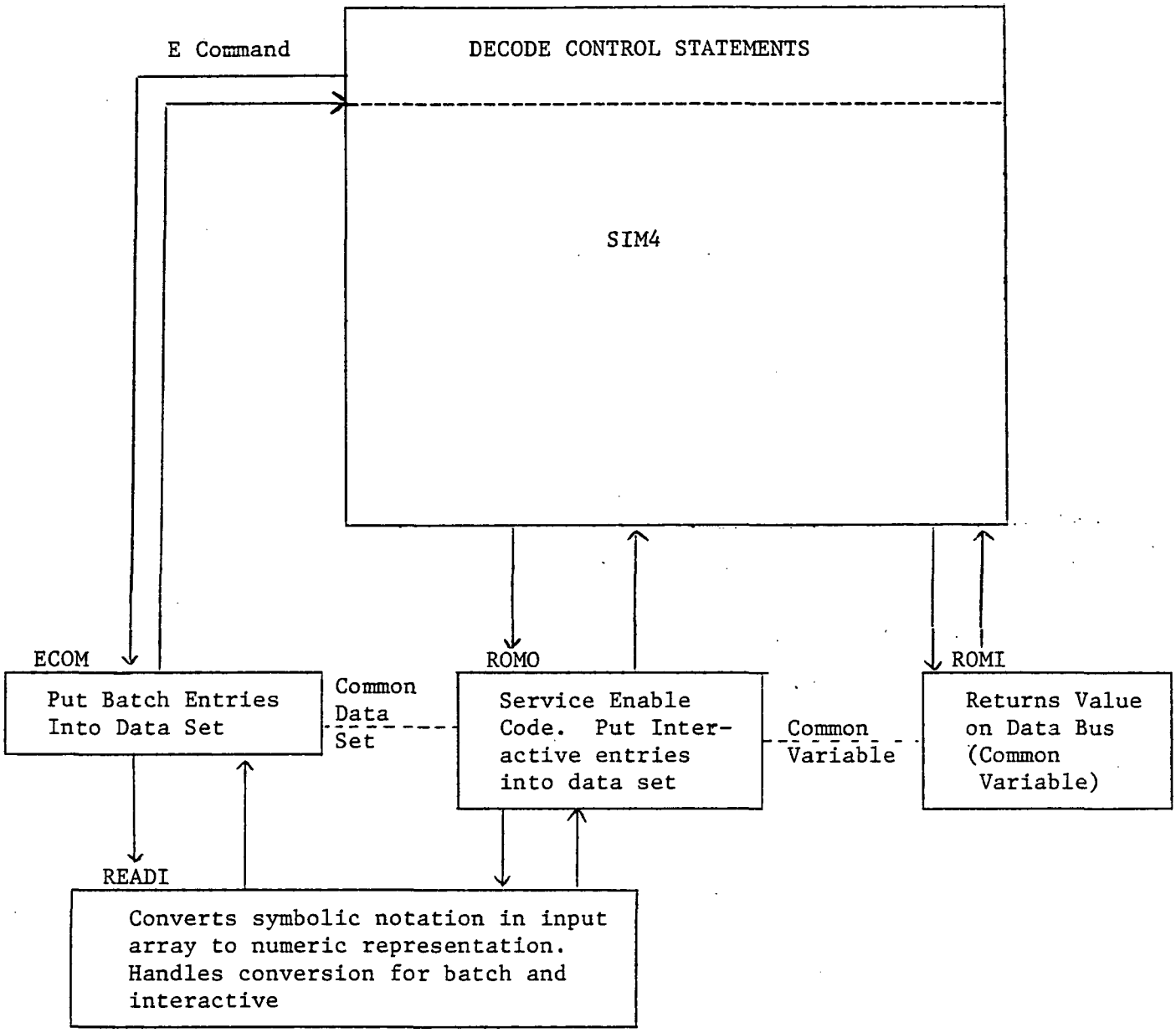


Figure F-5. Block Diagram Description of I/O Routines.

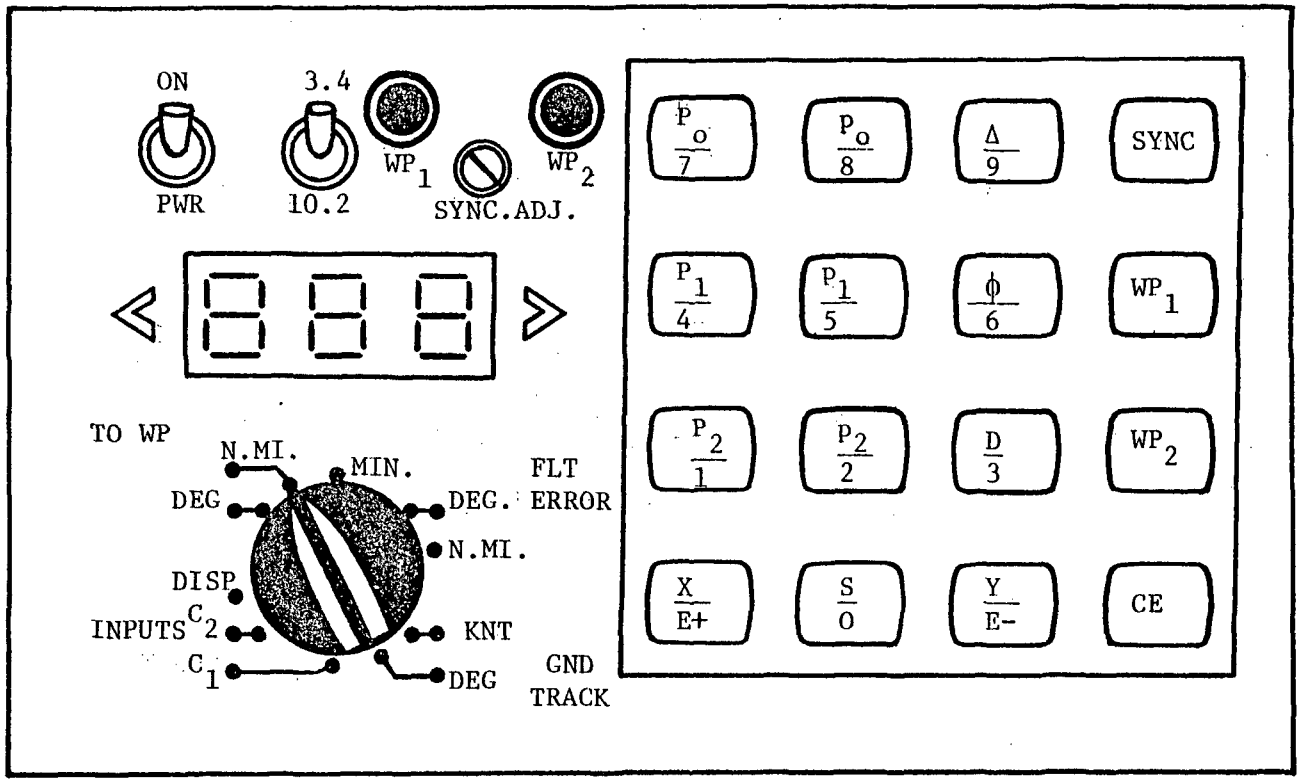


Figure F-6. OMEGA Receiver Front Panel

TABLE F-2. CURRENT FRONT PANEL INPUT CODES

Rotary Switch Codes: (see Figure F-6).			Keyboard Character Codes	
CODE	ROTARY SWITCH POSITION AS SHOWN IN FIGURE F-6	DESCRIPTION	CODE	REPRESENTATION ON KEYBOARD
RS1	Inputs C_1	Enter Δ, ϕ	A	SYNC
RS2	Inputs C_2	Enter P_n, D, S	7	P_o, 7
DSP	DISP	Display C	8	P_o, 8
DTD	N. MI. to WP	Range, N. MI.	9	Δ, 9
BRG	DEG to WP	Bearing, DEG	B	WP_1
ETA	MIN	Time, MIN	4	P_1, 4
ERH	N. MI. FLT error	Error, N. MI.	5	P_1, 5
ERD	DEG FLT error	Error, DEG	6	ϕ, 6
GTV	KNT GND Track	Velocity, Knots	C	WP_2
GTH	DEG GND Track	Heading, DEG	1	P_2, 1
STP*		Stop (Interactive Mode Only)	2	P_2, 2
REG*		Dump Registers	3	D, 3
RAM*		Dump RAM	D	CE
ROM*		Dump ROM	+	X, E+
			0	S, O
			-	Y, E-

* Not on front panel. Included for convenience of user only.

F.9.1 Subroutines for I/O

Subroutines written specifically for the Omega receiver simulation are described below. A simplified view of the relation between the routines is shown in Figure F-5.

F.9.1.1 Subroutine ROMO (IP, IA, ITIME)

This routine accepts data from ROM output port #2, deciphers and services enable codes from ROM output ports 0 and 1, manages the front panel data set and requests additional input from the user if the data set is empty.

IP = ROM output port number
IA = output value
ITIME = time

F.9.1.2 Subroutine ROMI (IP, IA, ITIME)

This routine returns via a designated ROM input port the current value on the data bus. This value was placed in a common data area by the ROMO subroutine.

IP = ROM input port number
IA = input value
ITIME = time

F.9.1.3 Subroutine READI (IBUF, IBUF1, JFLG)

This routine converts external (symbolic) input into internal (numeric) code.

IBUF (80) = array, converted internal character set
IBUF1 (80) = array, symbolic input
JFLG = flag, 0 = interactive input
1 = keyboard input only
2 = batch input

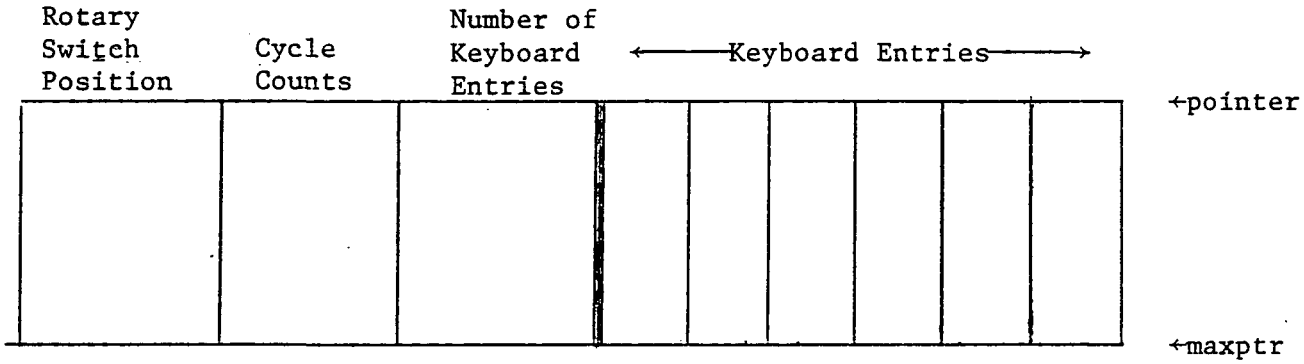
F.9.1.4 Subroutine ECOM

This routine sets up a front panel data set from "batch"[1] entries (E - control statement). This data set is then available to subroutine ROMO during execution of the microprogram.

[1]Use of the term "batch" in this appendix refers to entries made via the E control statement. It does not mean that the program SIM4 was necessarily executed in batch mode.

F.9.2 Front Panel Input

An internal data set may be established prior to execution of the mirco-program using a "batch" input format and/or during execution of the micro-program using an interactive input format. Both methods create a data set with the following format:



"Batch" entries are exhausted before interactive mode requests additional input. Each entry made by the user must have a 3 character rotary switch position code and a 3 decimal digit count (except do not follow STP,REG,ROM, RAM with a count). If the rotary switch requires keyboard entires, then they should follow the count. The count should be as big as the number of keyboard entries since it establishes the number of cycles for each rotary switch position.

F.9.2.1 Batch Input format.-

COL.	a	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	80
	E	a	a	a	d	d	d	c	c	x	a	a	a	d	d	d	x	

One Entry:

- aaa = Rotary Switch Code (3 characters)*
- ddd = Cycle Counts for this Position (3 decimal digits)**
- cc = Keyboard Entries for this Position (Optional)
- X = Blank

* = required

** = required except for STP, RAM, ROM, REG

As many entries per card as possible.

Do not break an entry over 2 cards.

As many E-command cards are allowed as needed, but the total number of entries must not exceed 50.

(Further entries may be made in interactive mode.)

F.9.2.2 Interactive input format.-

COL.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	.....	80
	a	a	a	d	d	d	c	c	x	z	z	z	d	d	d	x		

Format is same as batch except:

1) E is no longer required in COL. 1

2) Entries are accepted until a a a = STP

As in batch mode, the maximum number of entries is 50.

F.9.3 Front Panel Output

Output from the microprocessor appears on the front panel 3-digit display. Each digit may be one of the following characters:

0 - 9

A, blank, C, D, E, F

Output of a B or integer 11 will result in a blank on the display for that character.

Display output format:

LL XXX RL DL

LL, RL, DL = { ON - if left, right or destination light is on
Blank - if left, right or destination light is off

XXX = display lights

F.9.4 Errors

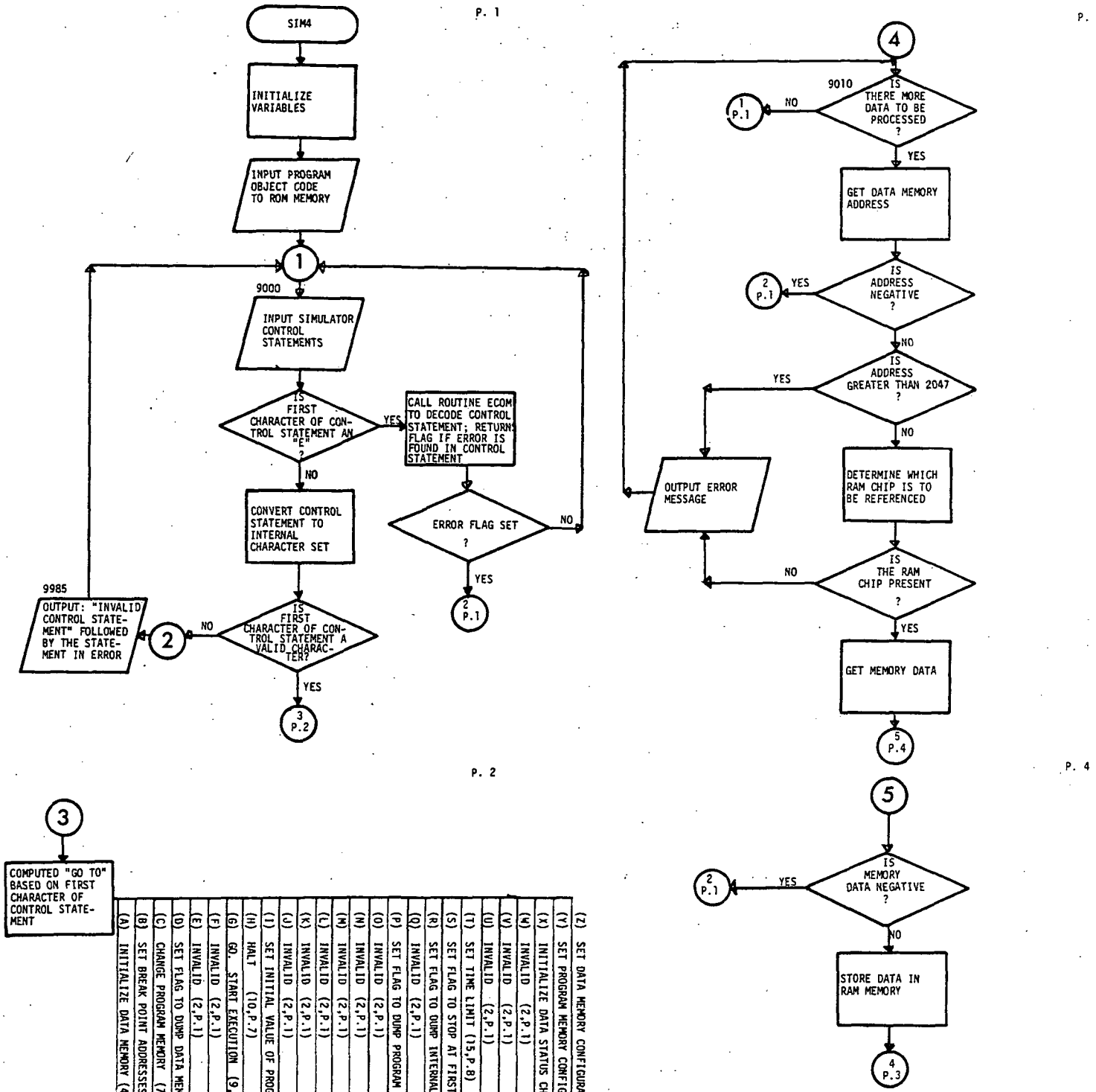
Both batch and interactive modes have limited error checking of input values. An entire input line of up to 80 characters is rejected if one character cannot be defined by a specified input array.

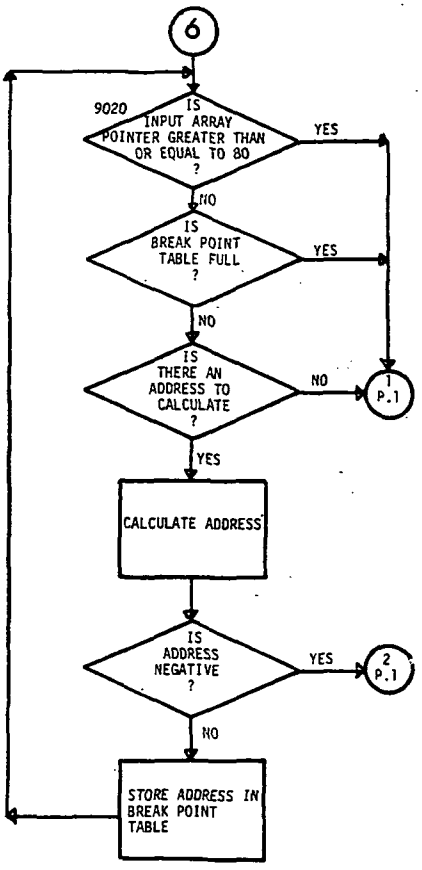
Example BRG005 RS10063G- STP
↑

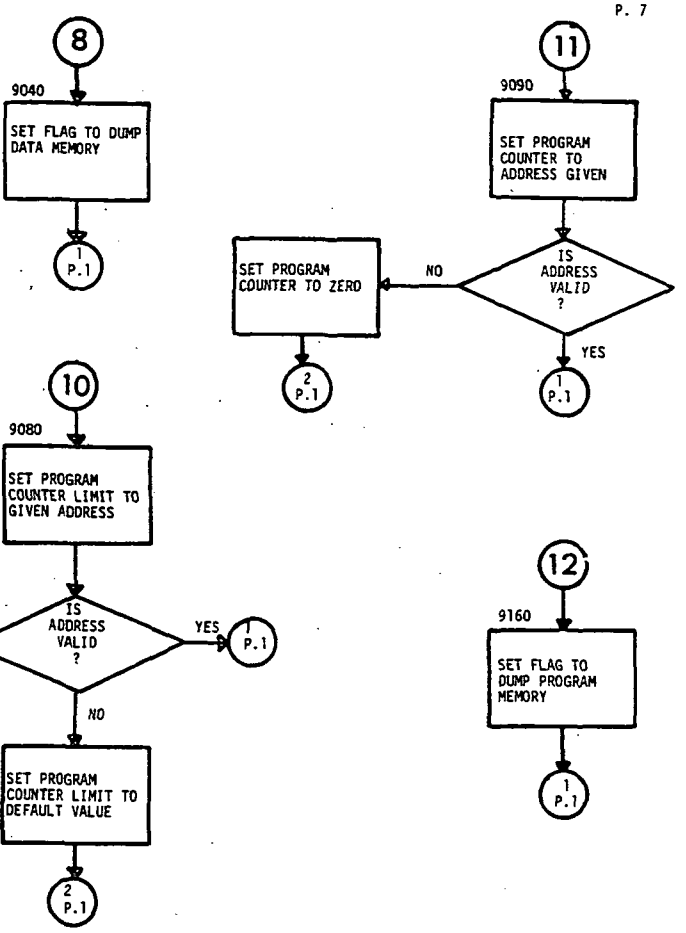
In the above example, G is not a valid keyboard entry and therefore, the entire line is rejected. The user is requested to re-enter the line (interactive) or line is ignored (batch).

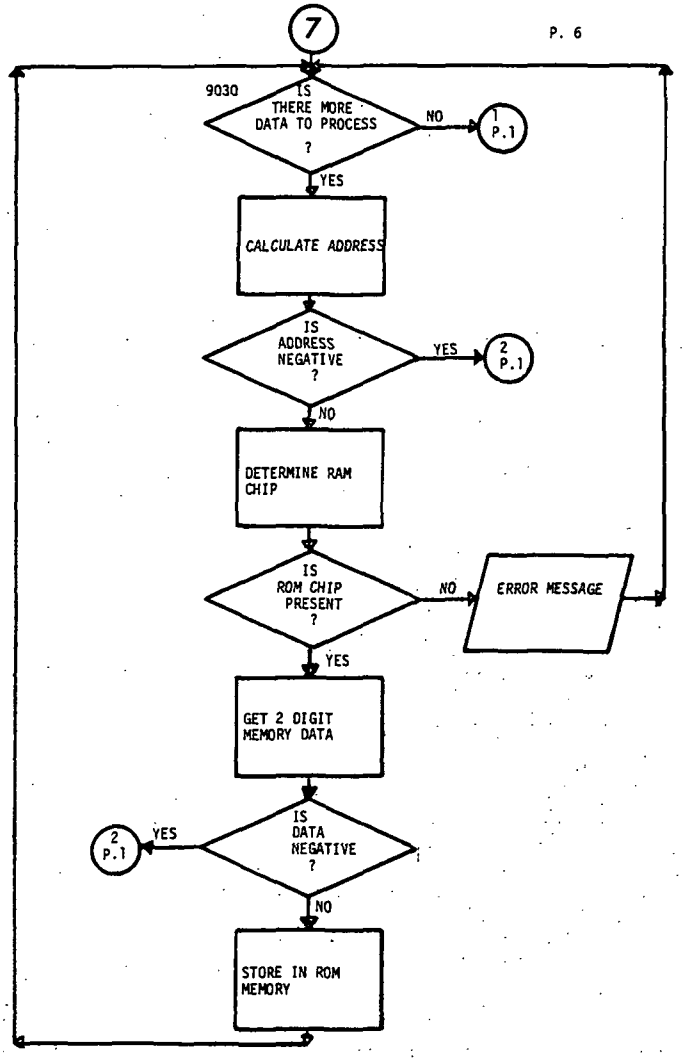
A specific error, E01, returned via the display by the user program, signals the I/O routines that the keyboard sequence for a particular rotary switch position was invalid. A request is made to re-enter just the keyboard values for that rotary switch position.

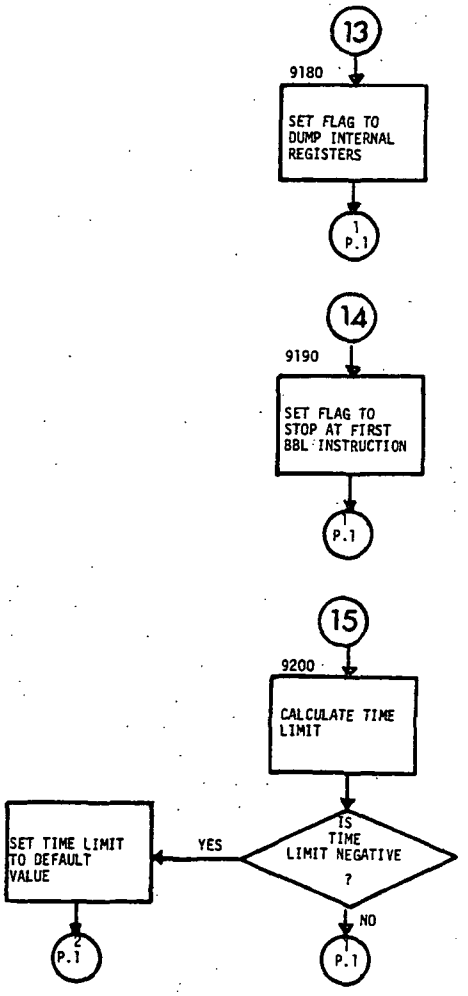
F.10 SIM4 PROGRAM FLOWCHART



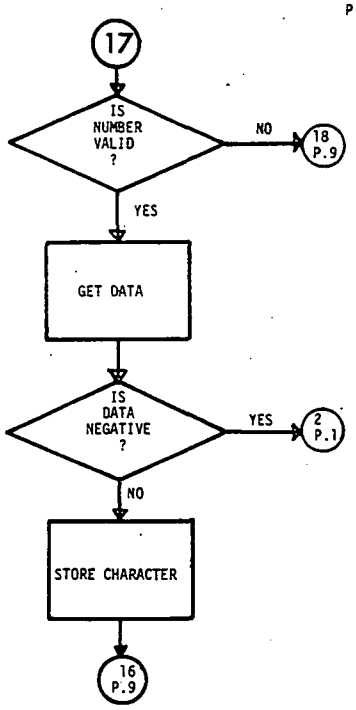

6
9020
IS INPUT ARRAY POINTER GREATER THAN OR EQUAL TO 80 ?
YES
NO
IS BREAK POINT TABLE FULL ?
YES
NO
IS THERE AN ADDRESS TO CALCULATE ?
NO
1 P.1
YES
CALCULATE ADDRESS
IS ADDRESS NEGATIVE ?
YES
2 P.1
NO
STORE ADDRESS IN BREAK POINT TABLE


8
9040
SET FLAG TO DUMP DATA MEMORY
1 P.1
11
9090
SET PROGRAM COUNTER TO ADDRESS GIVEN
IS ADDRESS VALID ?
NO
SET PROGRAM COUNTER TO ZERO
2 P.1
YES
1 P.1
10
9080
SET PROGRAM COUNTER LIMIT TO GIVEN ADDRESS
IS ADDRESS VALID ?
YES
1 P.1
NO
SET PROGRAM COUNTER LIMIT TO DEFAULT VALUE
2 P.1
12
9160
SET FLAG TO DUMP PROGRAM MEMORY
1 P.1

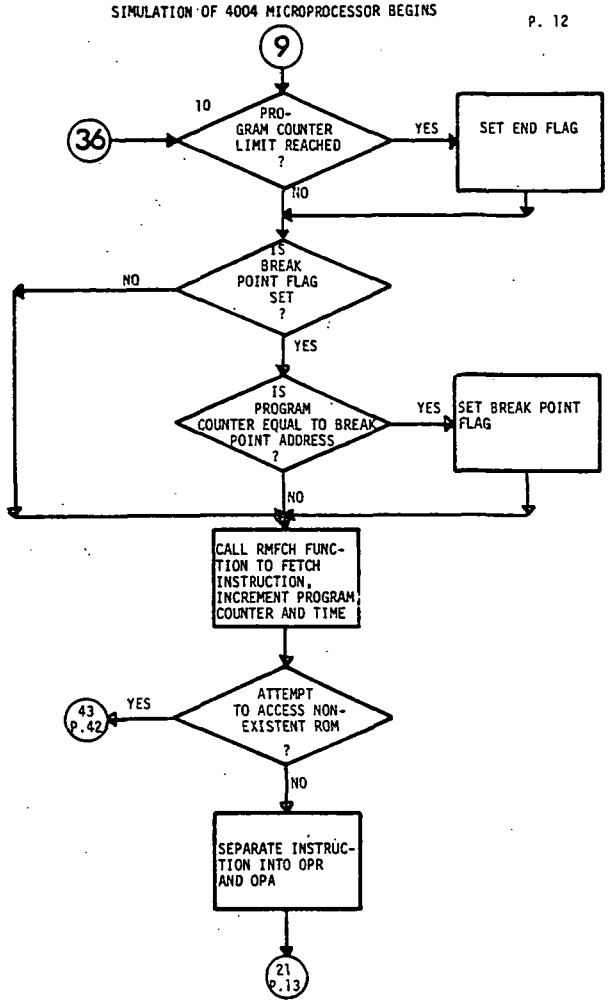

7
9030
IS THERE MORE DATA TO PROCESS ?
NO
1 P.1
YES
CALCULATE ADDRESS
IS ADDRESS NEGATIVE ?
YES
2 P.1
NO
DETERMINE RAM CHIP
IS ROM CHIP PRESENT ?
NO
ERROR MESSAGE
YES
GET 2 DIGIT MEMORY DATA
IS DATA NEGATIVE ?
YES
2 P.1
NO
STORE IN ROM MEMORY


13
9180
SET FLAG TO DUMP INTERNAL REGISTERS
1 P.1
14
9190
SET FLAG TO STOP AT FIRST BBL INSTRUCTION
1 P.1
15
9200
CALCULATE TIME LIMIT
IS TIME LIMIT NEGATIVE ?
YES
SET TIME LIMIT TO DEFAULT VALUE
2 P.1
NO
1 P.1

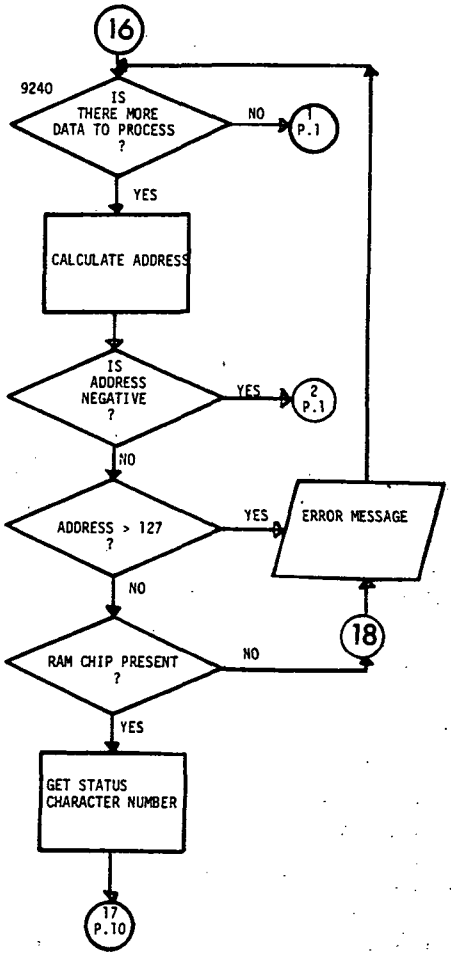
P. 10



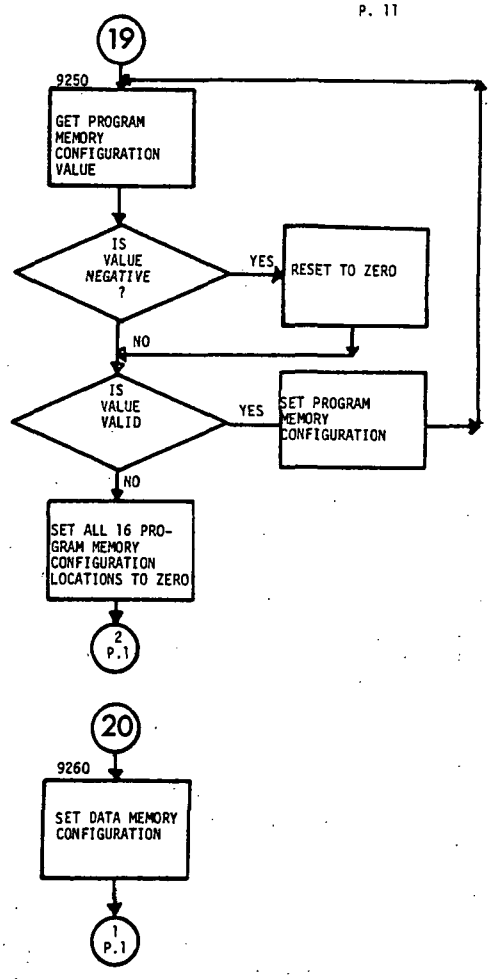
P. 12



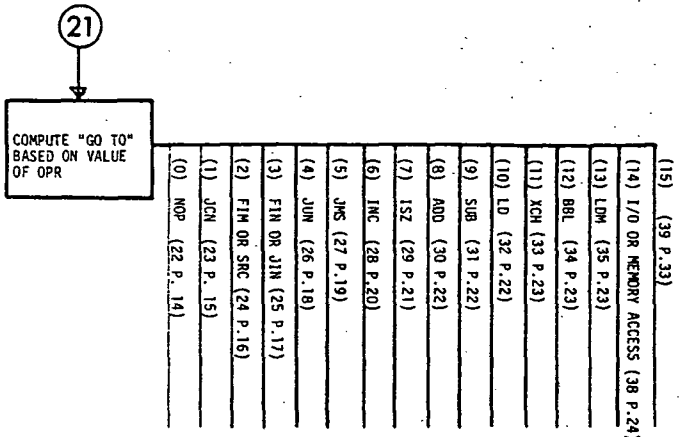
P. 9



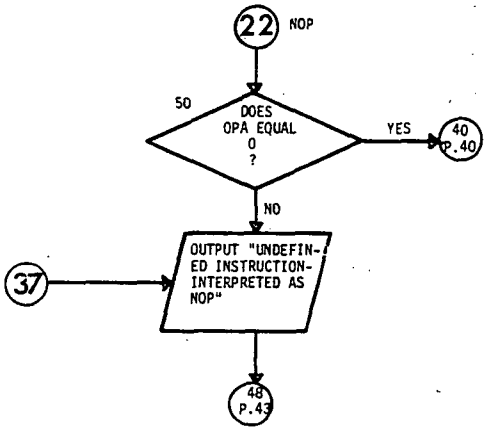
P. 11



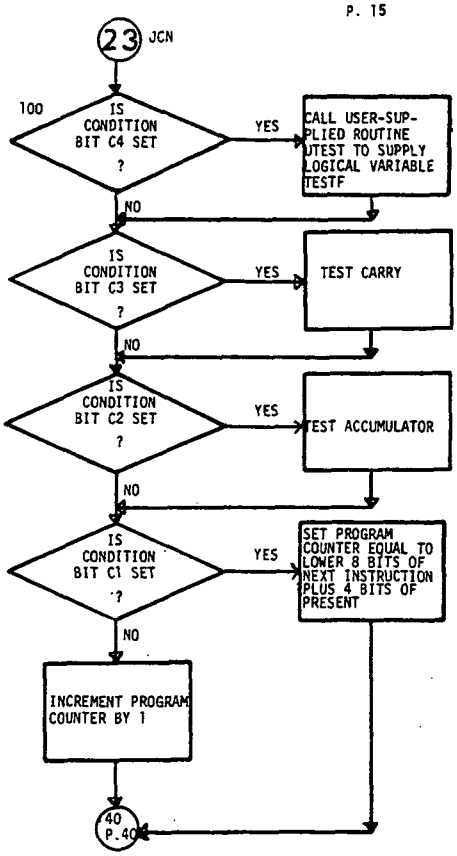
P. 13



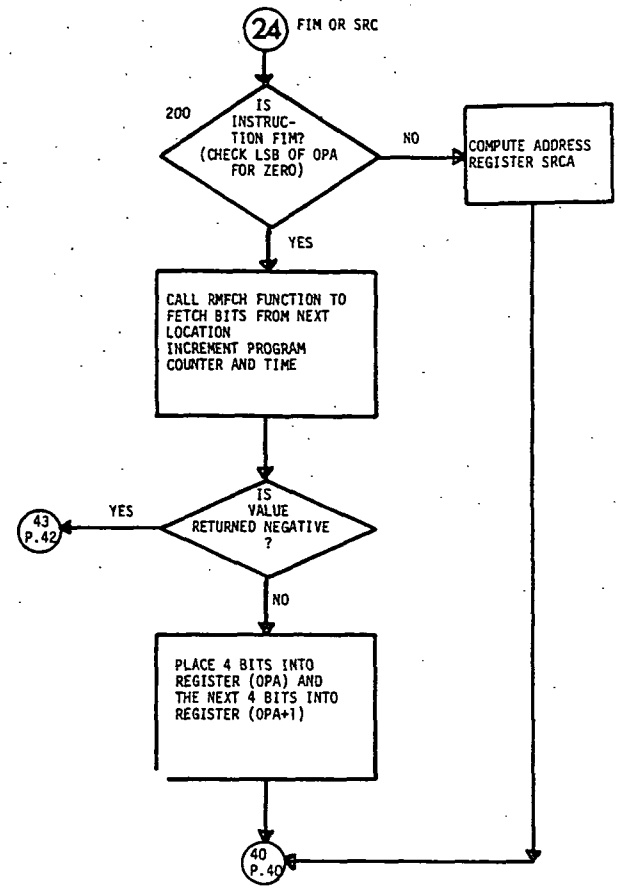
P. 14



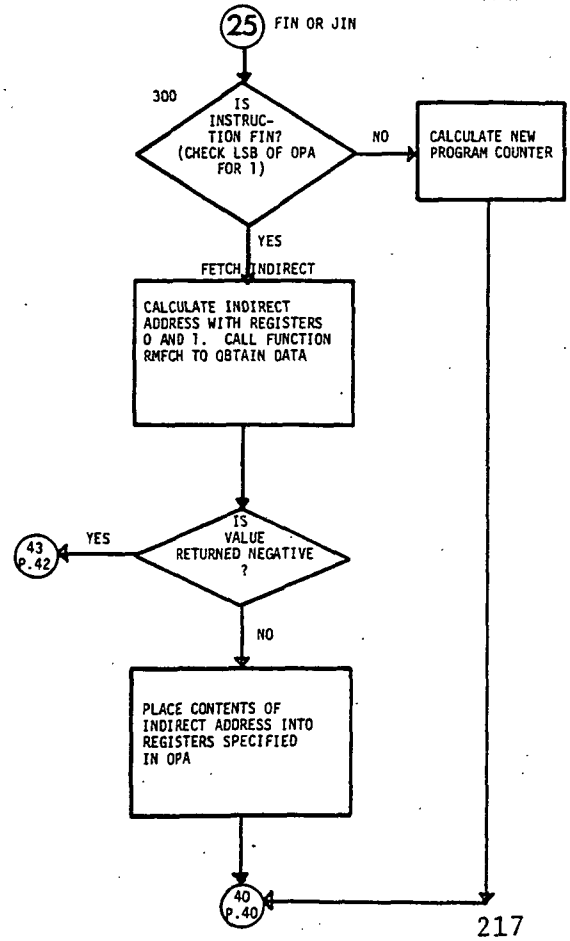
P. 15



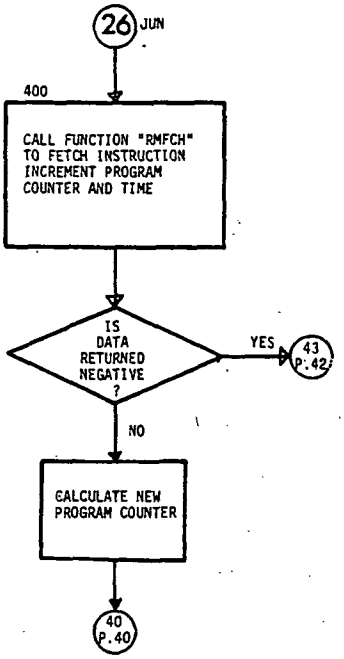
P. 16



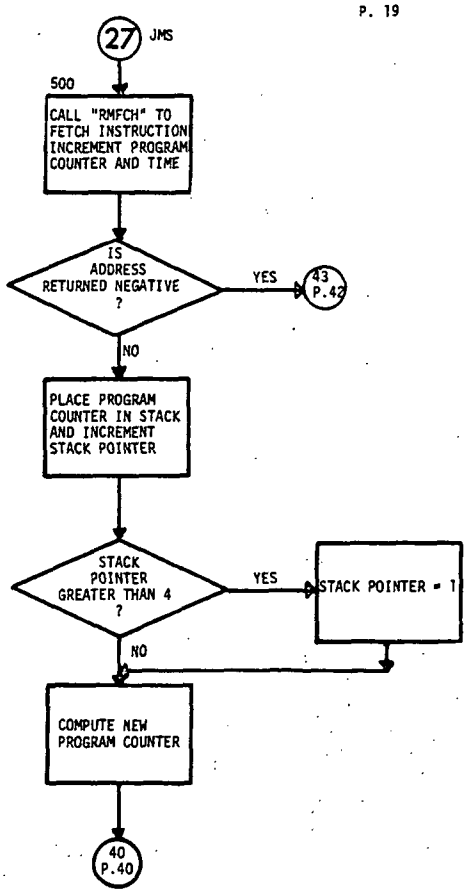
P. 17



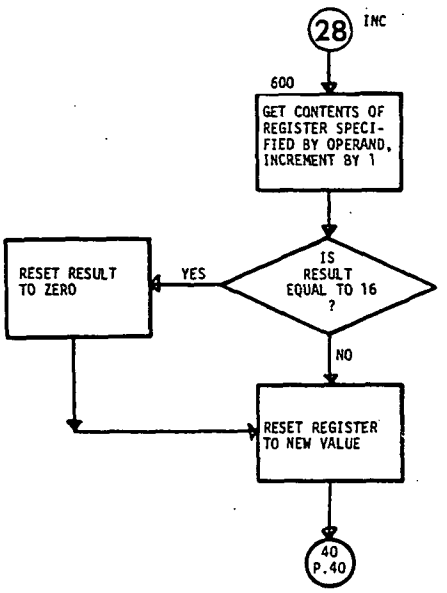
P. 18



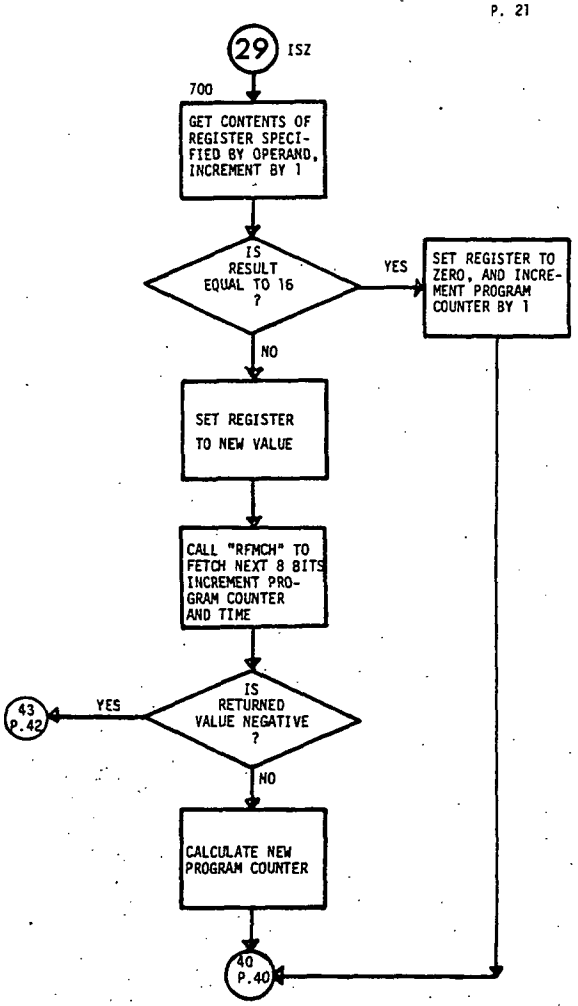
P. 19

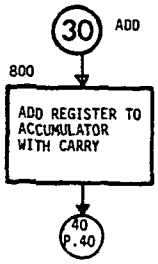


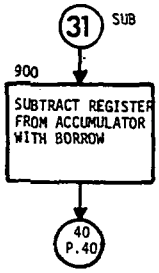
P. 20

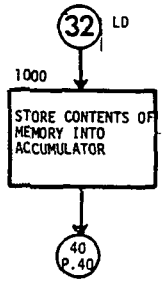


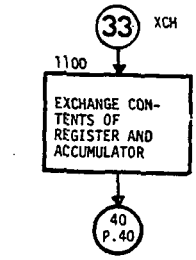
P. 21

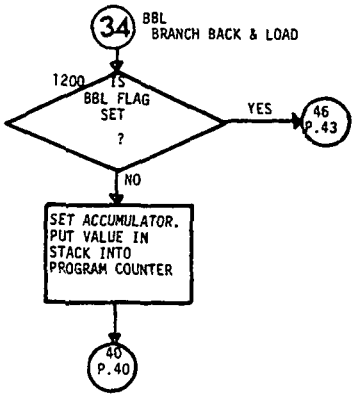


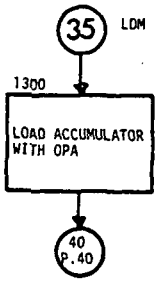

30
ADD
800
ADD REGISTER TO ACCUMULATOR WITH CARRY
40
P.40

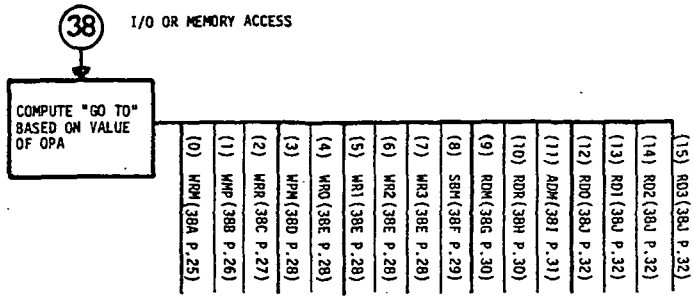

31
SUB
900
SUBTRACT REGISTER FROM ACCUMULATOR WITH BORROW
40
P.40

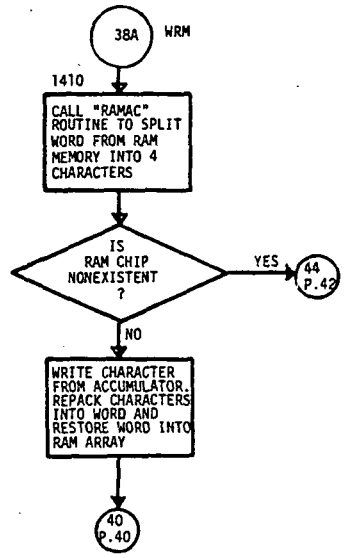

32
LD
1000
STORE CONTENTS OF MEMORY INTO ACCUMULATOR
40
P.40

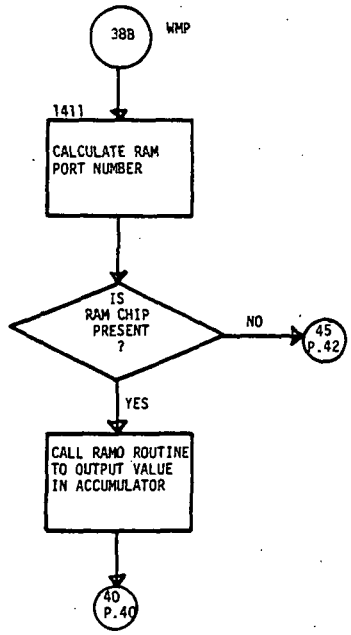

33
XCH
1100
EXCHANGE CONTENTS OF REGISTER AND ACCUMULATOR
40
P.40


34
BBL
BRANCH BACK & LOAD
1200
IS BBL FLAG SET ?
YES
46
P.43
NO
SET ACCUMULATOR. PUT VALUE IN STACK INTO PROGRAM COUNTER
40
P.40

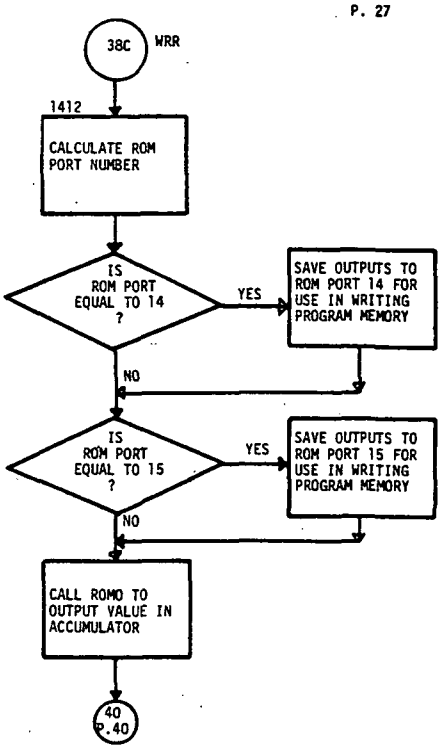

35
LDM
1300
LOAD ACCUMULATOR WITH OPA
40
P.40


38
I/O OR MEMORY ACCESS
COMPUTE "GO TO" BASED ON VALUE OF OPA
(0) WRM(38A P.25)
(1) WMP(38B P.26)
(2) WRR(38C P.27)
(3) WPM(38D P.28)
(4) WR0(38E P.28)
(5) WR1(38E P.28)
(6) WR2(38E P.28)
(7) WR3(38E P.28)
(8) SBM(38F P.29)
(9) RDM(38G P.30)
(10) RDR(38H P.30)
(11) ADM(38I P.31)
(12) RD0(38J P.32)
(13) RD1(38J P.32)
(14) RD2(38J P.32)
(15) RD3(38J P.32)

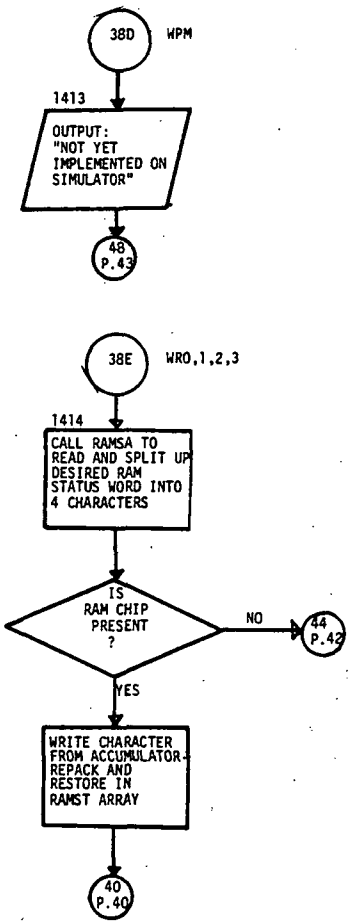

38A
WRM
1410
CALL "RAMAC" ROUTINE TO SPLIT WORD FROM RAM MEMORY INTO 4 CHARACTERS
IS RAM CHIP NONEXISTENT ?
YES
44
P.42
NO
WRITE CHARACTER FROM ACCUMULATOR. REPACK CHARACTERS INTO WORD AND RESTORE WORD INTO RAM ARRAY
40
P.40


38B
WMP
1411
CALCULATE RAM PORT NUMBER
IS RAM CHIP PRESENT ?
NO
45
P.42
YES
CALL RAMO ROUTINE TO OUTPUT VALUE IN ACCUMULATOR
40
P.40

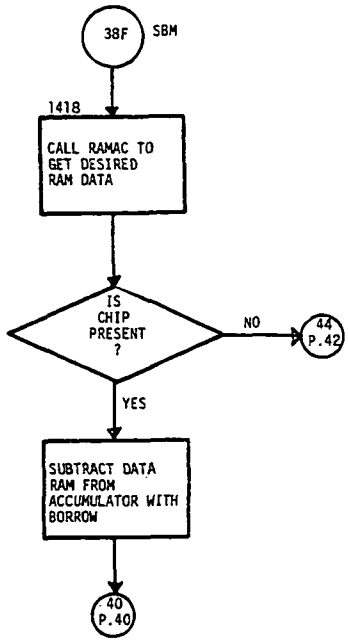
P. 27



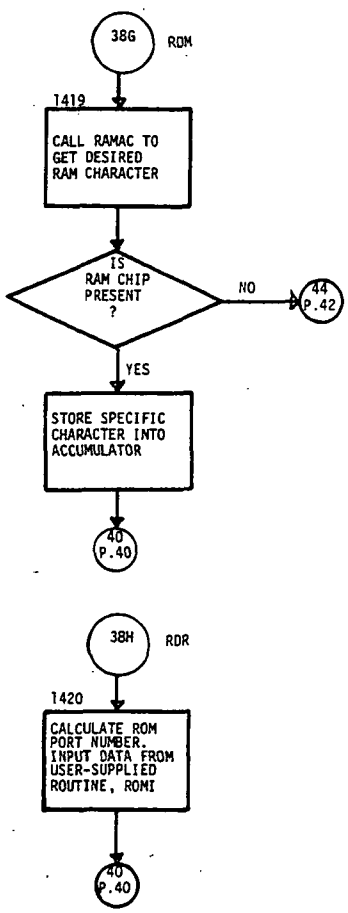
P. 28



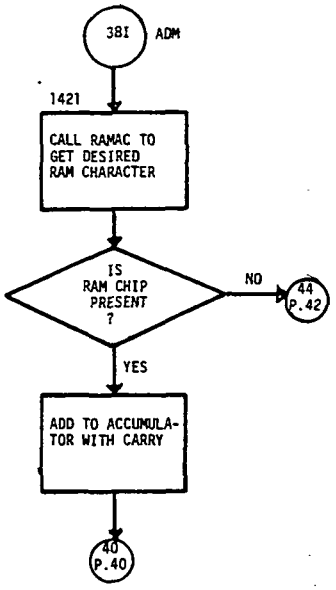
P. 29



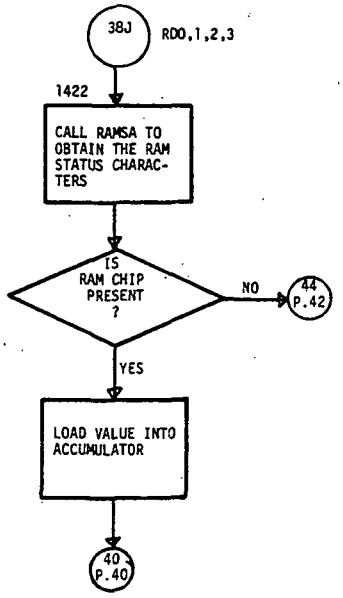
P. 30



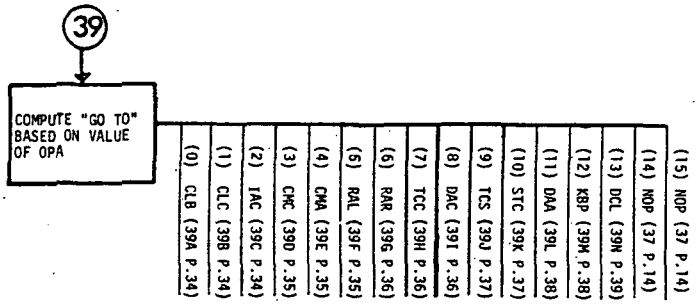
P. 31



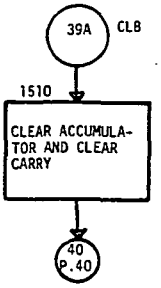
P. 32

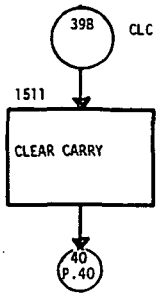


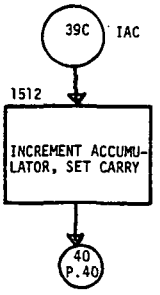
P. 33



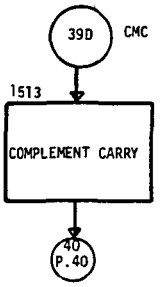
P. 34

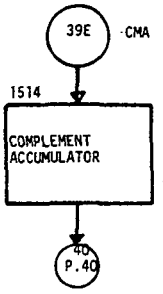


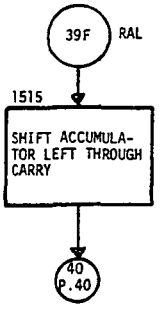




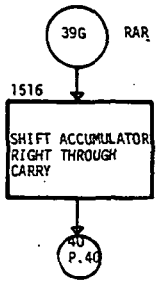
P. 35

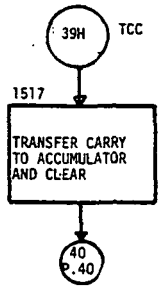


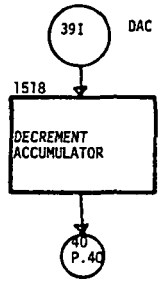




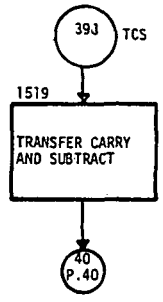
P. 36

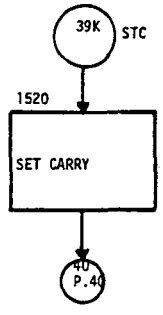




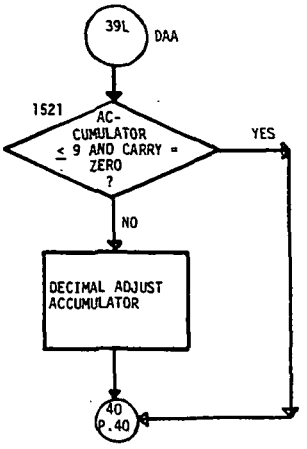


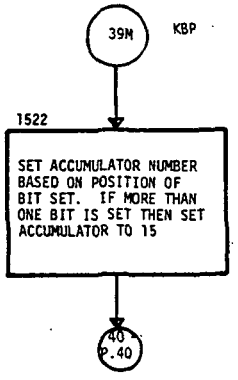
P. 37



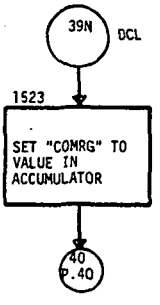


P. 38

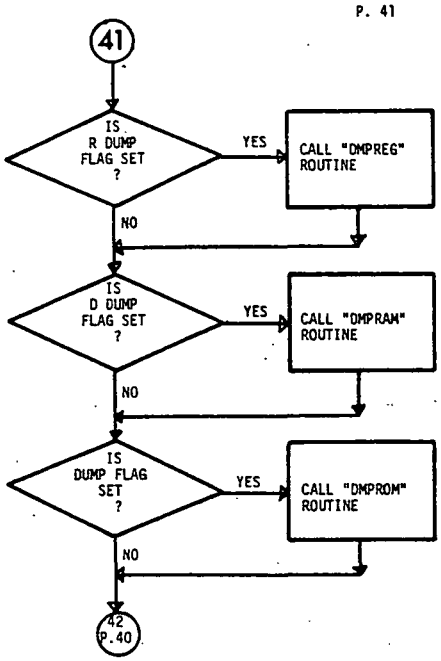




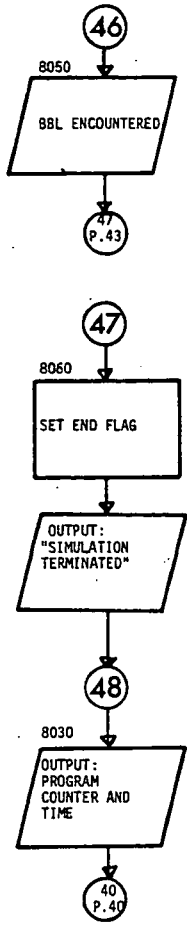
P. 39



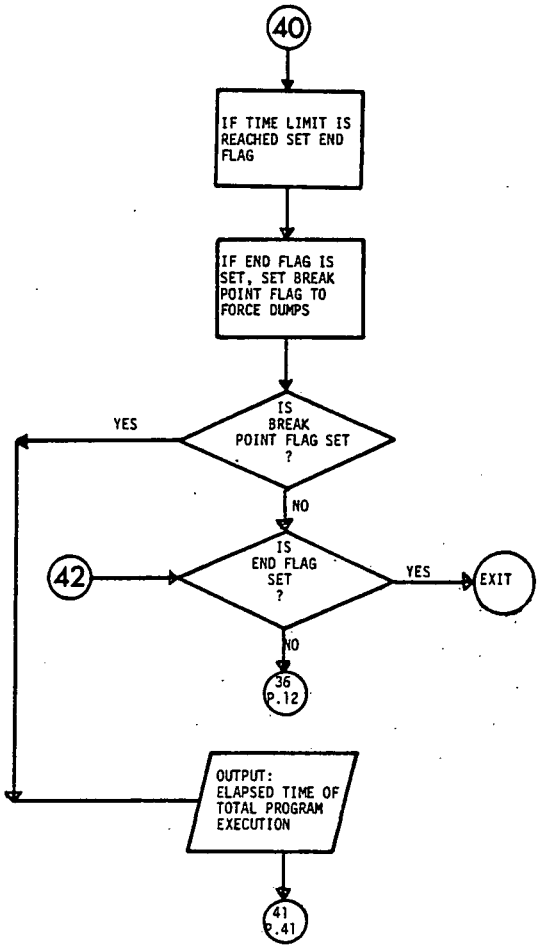
P. 41



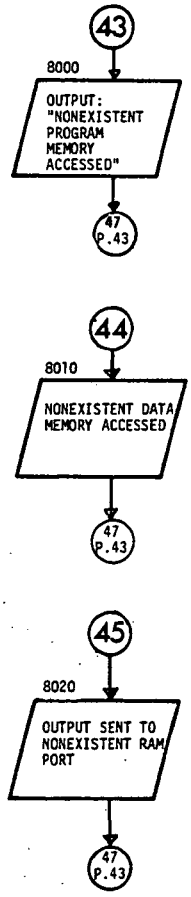
P. 43



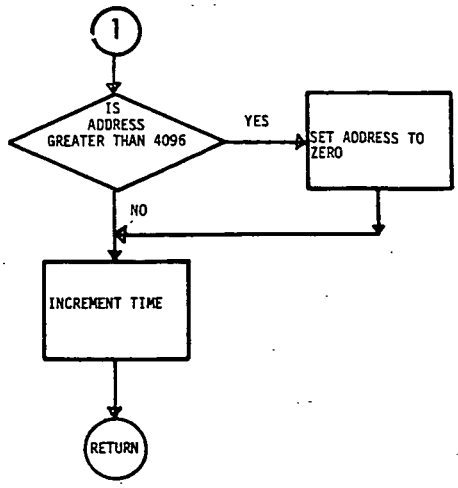
P. 40



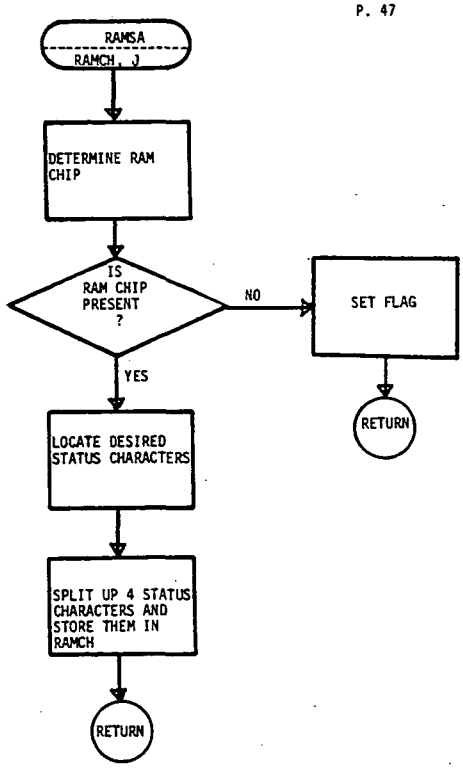
P. 42



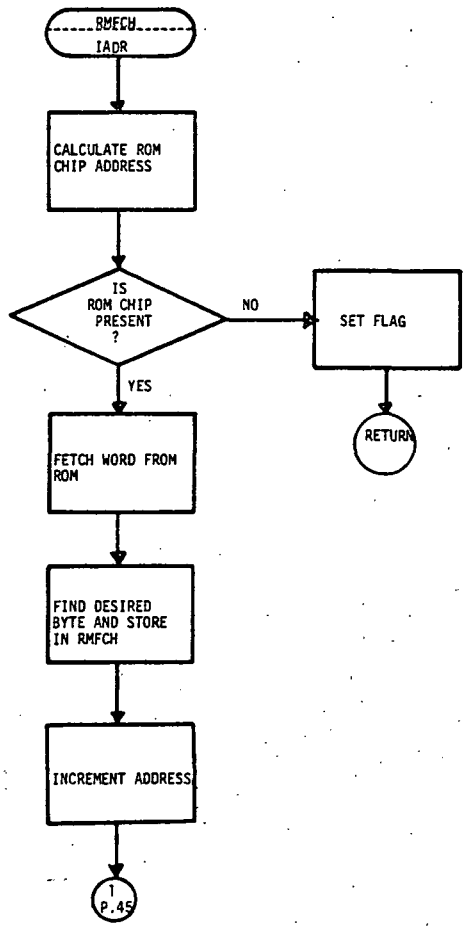
P. 45



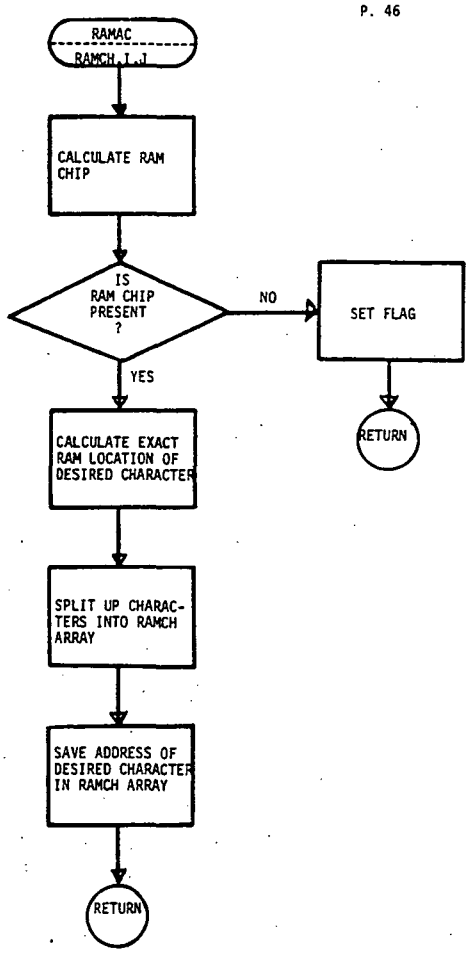
P. 47

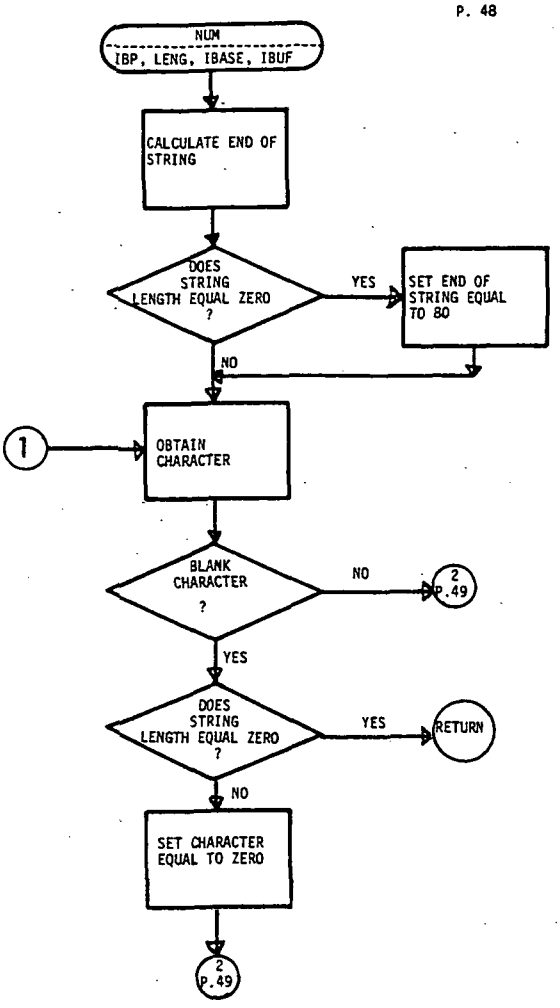


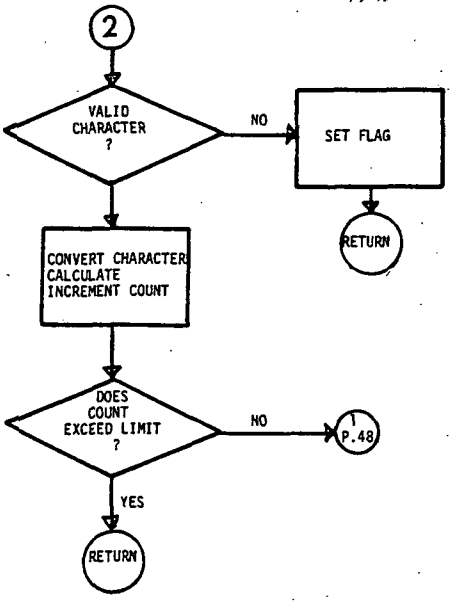
P. 44

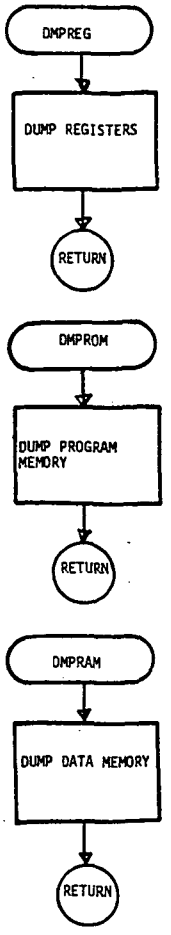


P. 46

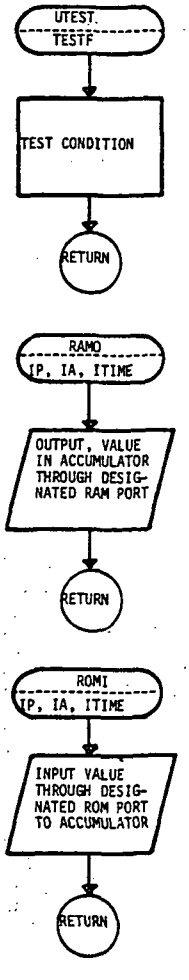


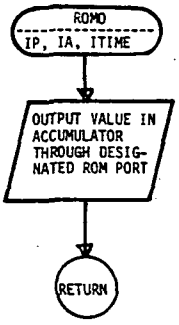


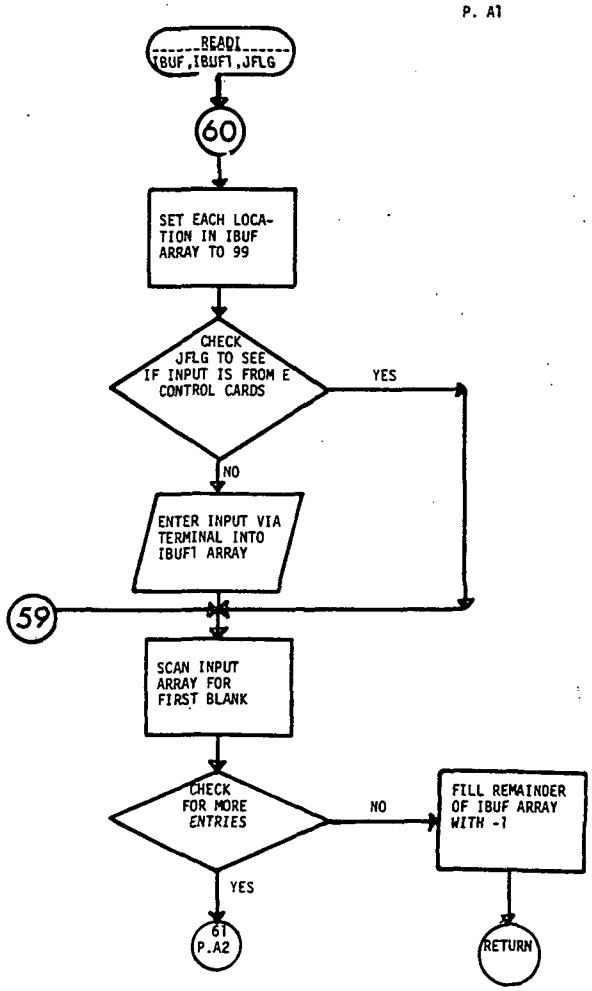


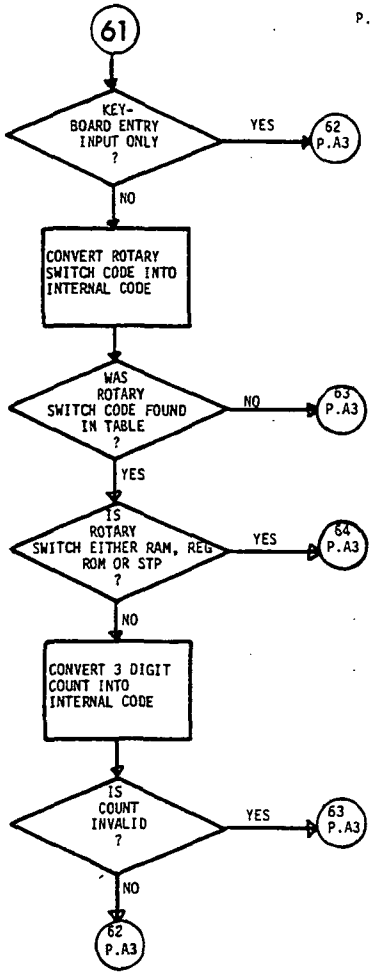


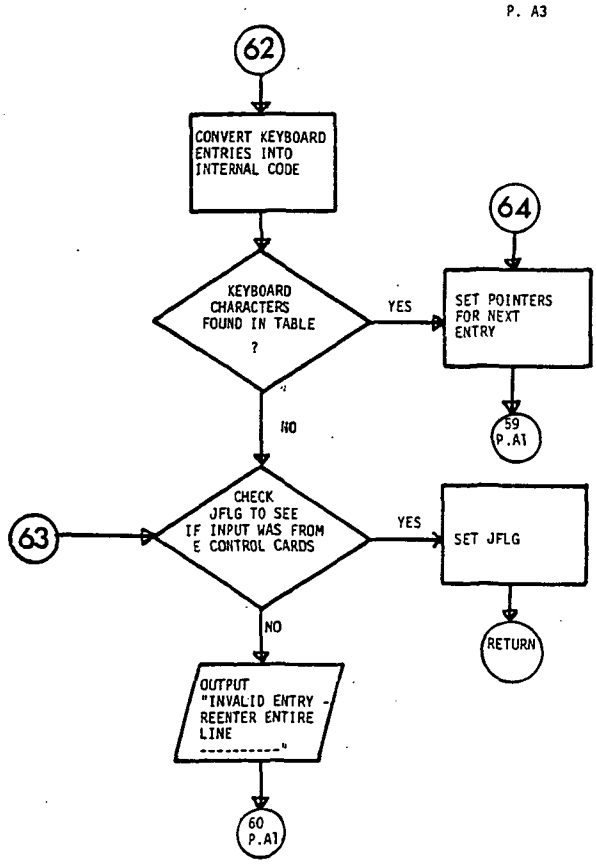
USER SUPPLIED SUBROUTINES

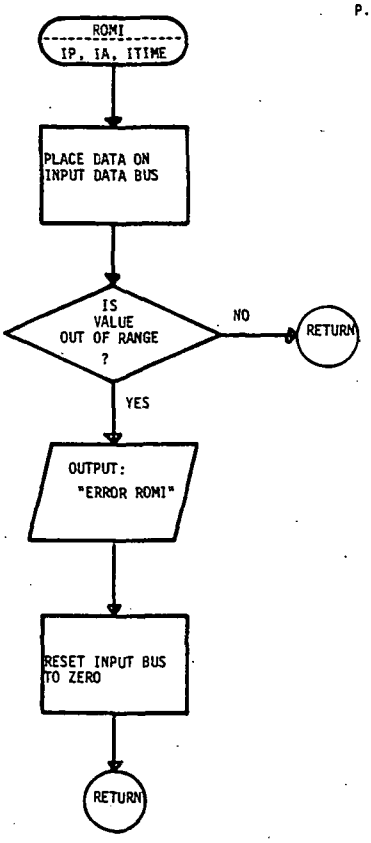


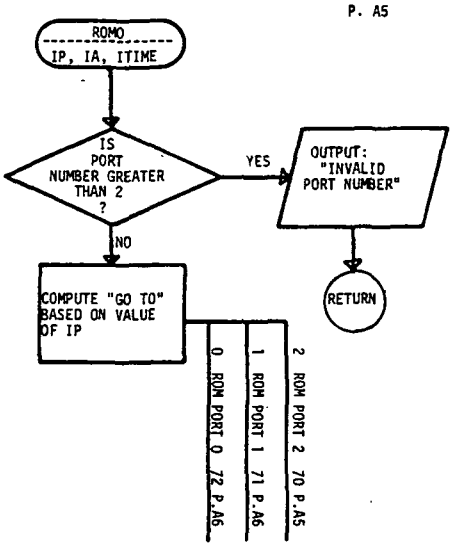

P. 52
ROMO
IP, IA, ITIME
OUTPUT VALUE IN ACCUMULATOR THROUGH DESIG-NATED ROM PORT
RETURN

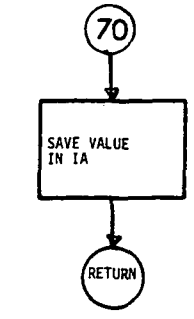

P. A1
READI
IBUF,IBUF1,JFLG
60
SET EACH LOCA-TION IN IBUF ARRAY TO 99
CHECK JFLG TO SEE IF INPUT IS FROM E CONTROL CARDS
YES
NO
ENTER INPUT VIA TERMINAL INTO IBUF1 ARRAY
59
SCAN INPUT ARRAY FOR FIRST BLANK
CHECK FOR MORE ENTRIES
NO
FILL REMAINDER OF IBUF ARRAY WITH -1
YES
61 P.A2
RETURN

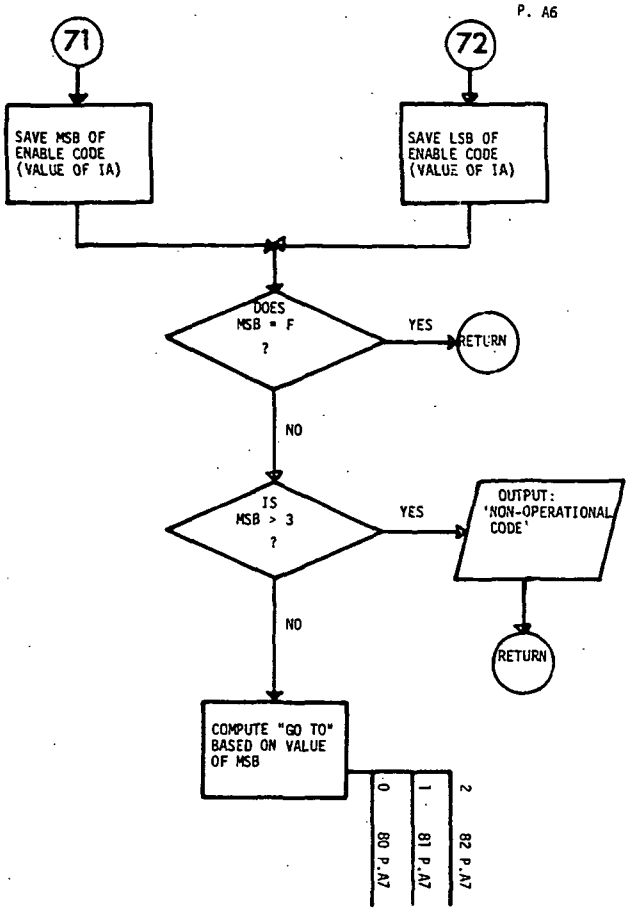

P. A2
61
KEY-BOARD ENTRY INPUT ONLY ?
YES
62 P.A3
NO
CONVERT ROTARY SWITCH CODE INTO INTERNAL CODE
WAS ROTARY SWITCH CODE FOUND IN TABLE ?
NO
63 P.A3
YES
IS ROTARY SWITCH EITHER RAM, REG ROM OR STP ?
YES
64 P.A3
NO
CONVERT 3 DIGIT COUNT INTO INTERNAL CODE
IS COUNT INVALID ?
YES
63 P.A3
NO
62 P.A3

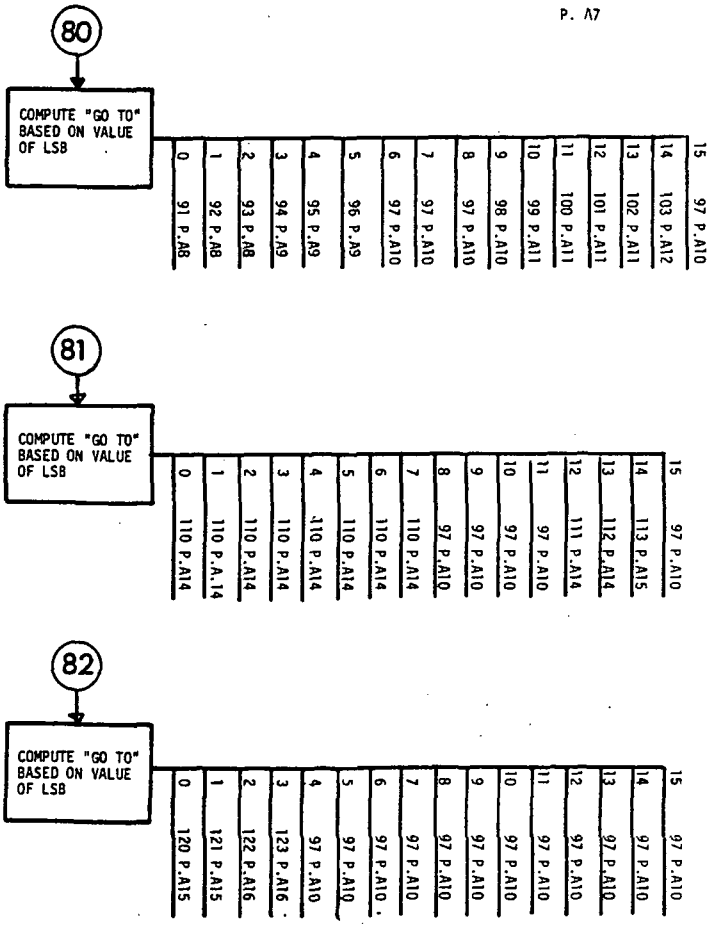

P. A3
62
CONVERT KEYBOARD ENTRIES INTO INTERNAL CODE
64
KEYBOARD CHARACTERS FOUND IN TABLE ?
YES
SET POINTERS FOR NEXT ENTRY
59 P.A1
NO
63
CHECK JFLG TO SEE IF INPUT WAS FROM E CONTROL CARDS
YES
SET JFLG
RETURN
NO
OUTPUT "INVALID ENTRY - REENTER ENTIRE LINE ----------"
60 P.A1

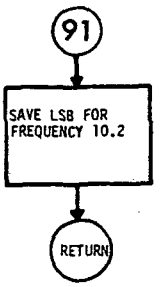

P. A4
ROMI
IP, IA, ITIME
PLACE DATA ON INPUT DATA BUS
IS VALUE OUT OF RANGE ?
NO
RETURN
YES
OUTPUT: "ERROR ROMI"
RESET INPUT BUS TO ZERO
RETURN

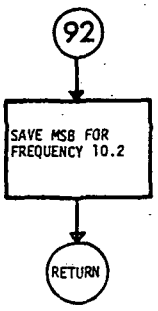

P. A5
ROMO
IP, IA, ITIME
IS PORT NUMBER GREATER THAN 2 ?
YES
OUTPUT: "INVALID PORT NUMBER"
RETURN
NO
COMPUTE "GO TO" BASED ON VALUE OF IP
0 ROM PORT 0 72 P.A6
1 ROM PORT 1 71 P.A6
2 ROM PORT 2 70 P.A5

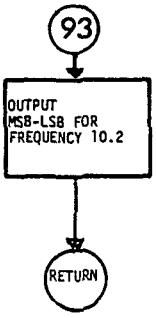

70
SAVE VALUE IN IA
RETURN

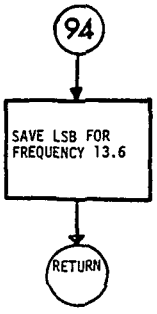

P. A6
71
SAVE MSB OF ENABLE CODE (VALUE OF IA)
72
SAVE LSB OF ENABLE CODE (VALUE OF IA)
DOES MSB = F ?
YES
RETURN
NO
IS MSB > 3 ?
YES
OUTPUT: 'NON-OPERATIONAL CODE'
RETURN
NO
COMPUTE "GO TO" BASED ON VALUE OF MSB
0 80 P.A7
1 81 P.A7
2 82 P.A7


P. A7
80
COMPUTE "GO TO" BASED ON VALUE OF LSB
0 91 P.A8
1 92 P.A8
2 93 P.A8
3 94 P.A9
4 95 P.A9
5 96 P.A9
6 97 P.A10
7 97 P.A10
8 97 P.A10
9 98 P.A10
10 99 P.A11
11 100 P.A11
12 101 P.A11
13 102 P.A11
14 103 P.A12
15 97 P.A10
81
COMPUTE "GO TO" BASED ON VALUE OF LSB
0 110 P.A14
1 110 P.A.14
2 110 P.A14
3 110 P.A14
4 110 P.A14
5 110 P.A14
6 110 P.A14
7 110 P.A14
8 97 P.A10
9 97 P.A10
10 97 P.A10
11 97 P.A10
12 111 P.A14
13 112 P.A14
14 113 P.A15
15 97 P.A10
82
COMPUTE "GO TO" BASED ON VALUE OF LSB
0 120 P.A15
1 121 P.A15
2 122 P.A16
3 123 P.A16
4 97 P.A10
5 97 P.A10
6 97 P.A10
7 97 P.A10
8 97 P.A10
9 97 P.A10
10 97 P.A10
11 97 P.A10
12 97 P.A10
13 97 P.A10
14 97 P.A10
15 97 P.A10

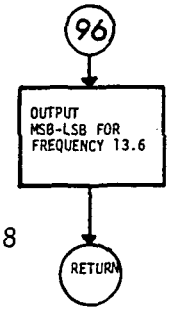

91
SAVE LSB FOR FREQUENCY 10.2
RETURN

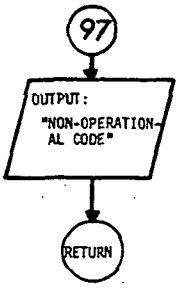

92
SAVE MSB FOR FREQUENCY 10.2
RETURN

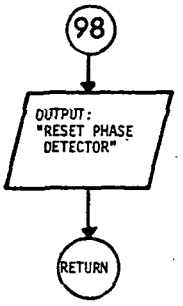

93
OUTPUT MSB-LSB FOR FREQUENCY 10.2
RETURN

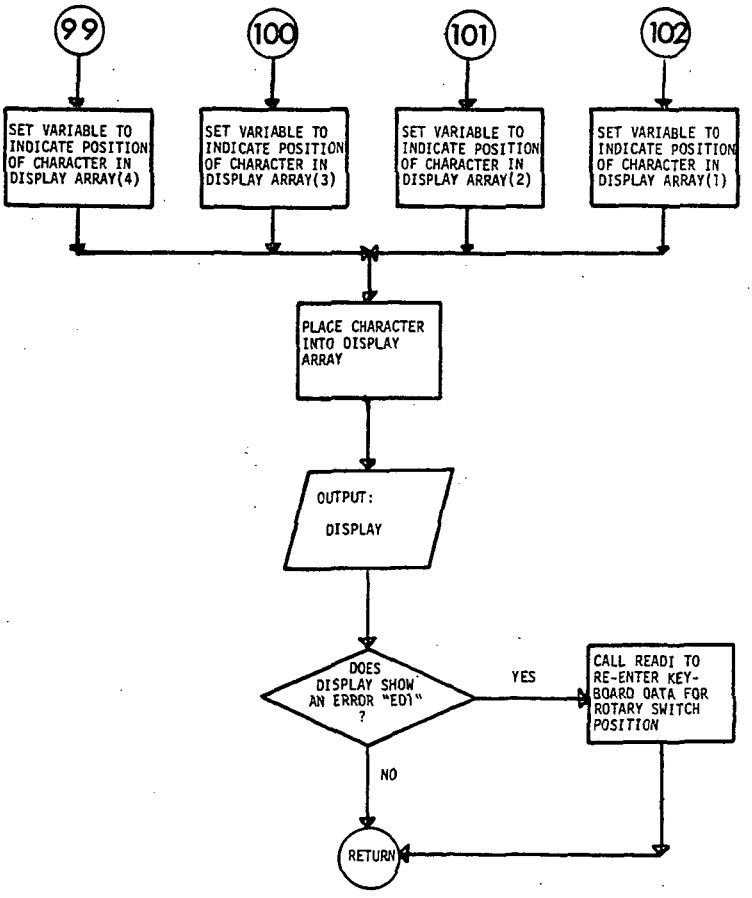

94
SAVE LSB FOR FREQUENCY 13.6
RETURN


95
SAVE MSB FOR FREQUENCY 13.6
RETURN

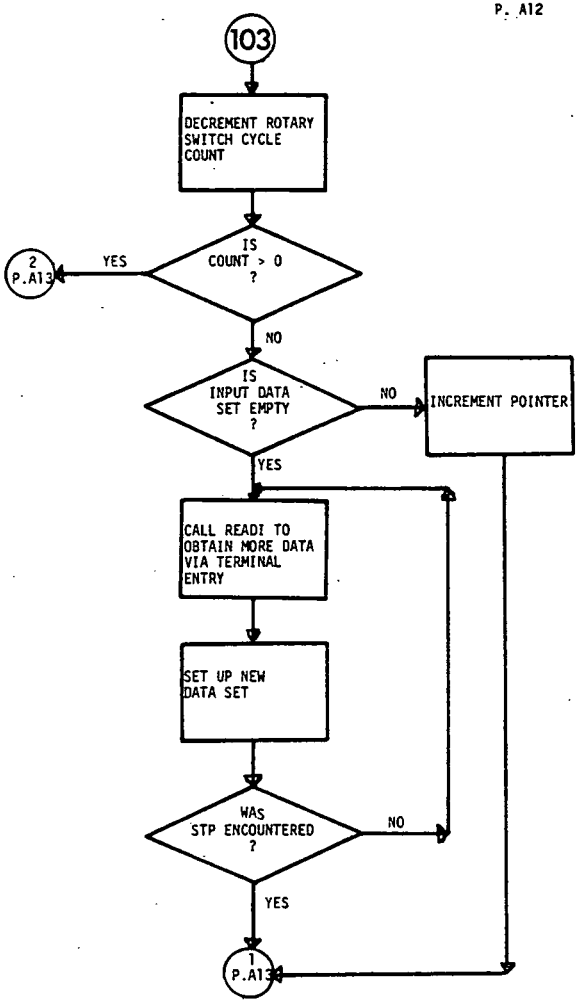

96
OUTPUT MSB-LSB FOR FREQUENCY 13.6
RETURN


97
OUTPUT: "NON-OPERATIONAL CODE"
RETURN

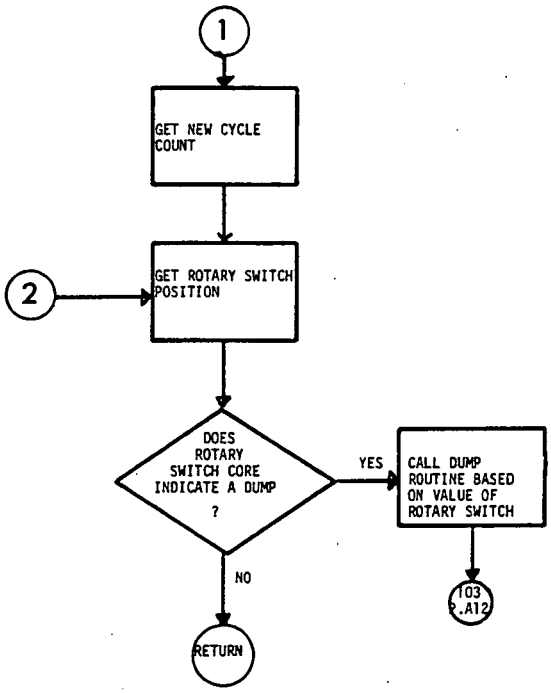

98
OUTPUT: "RESET PHASE DETECTOR"
RETURN


99
SET VARIABLE TO INDICATE POSITION OF CHARACTER IN DISPLAY ARRAY(4)
100
SET VARIABLE TO INDICATE POSITION OF CHARACTER IN DISPLAY ARRAY(3)
101
SET VARIABLE TO INDICATE POSITION OF CHARACTER IN DISPLAY ARRAY(2)
102
SET VARIABLE TO INDICATE POSITION OF CHARACTER IN DISPLAY ARRAY(1)
PLACE CHARACTER INTO DISPLAY ARRAY
OUTPUT: DISPLAY
DOES DISPLAY SHOW AN ERROR "ED1" ?
YES
CALL READI TO RE-ENTER KEYBOARD DATA FOR ROTARY SWITCH POSITION
NO
RETURN

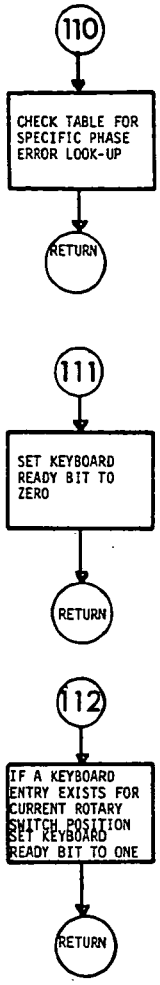
P. A12



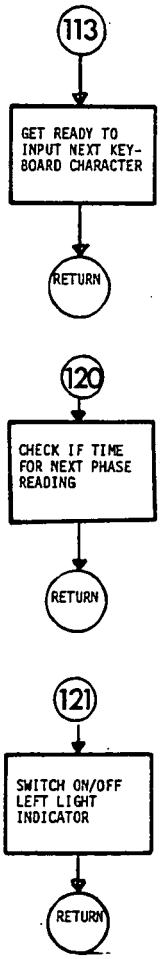
P. A13



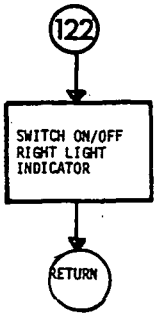
P. A14

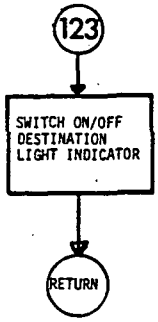


P. A15

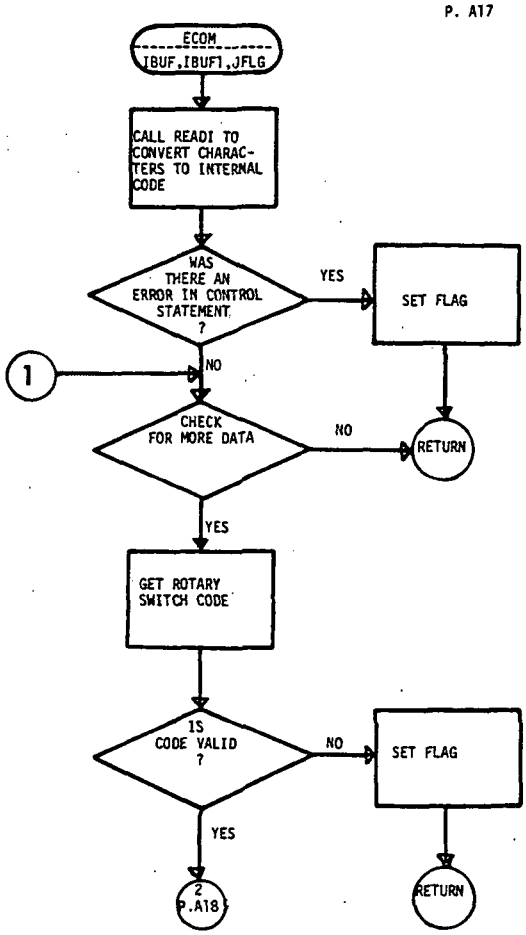


P. A16

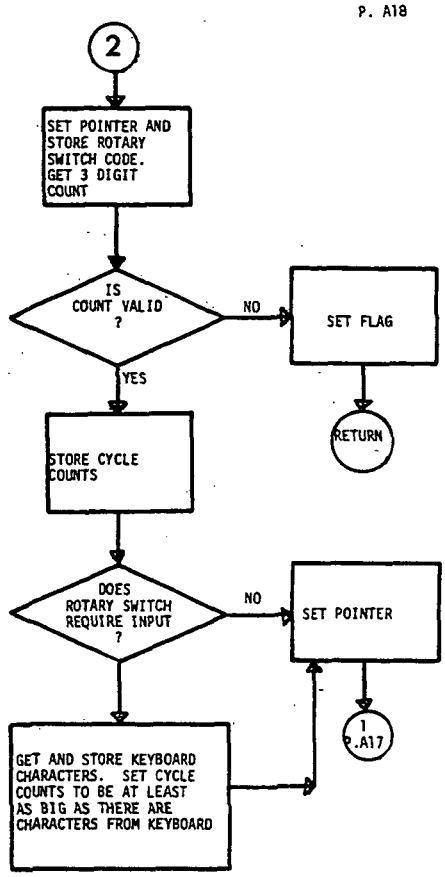




P. A17



P. A18



REFERENCES

Page
Intentionally
Left Blank

REFERENCES

1. Baxa, E. G., Jr., "Implementation of an Experimental Program to Investigate the Performance Characteristics of OMEGA Navigation," NASA CR-132516, Contractor Report under contract NAS1-12043, Research Triangle Institute, Research Triangle Park, North Carolina, September 1974.

2. VERSION 210 4004 MACRO ASSEMBLER, MAC4, INTEL CORP., Santa Clara, California, September 1974.

3. Pierce, J. A., "OMEGA: Facts, Hopes, and Dreams," Technical Report No. 652, Engineering and Applied Physics, Harvard University, June 1974, AD-782 396.

4. Baxa, E. G., Jr., Lytle, C. D., "On Observations of Modal Interference of the North Dakota OMEGA Transmission," Navigation Journal of the Institute of Navigation, Vol. 22, No. 4, Winter 1975-76, pp 309-323.

5. Baxa, E. G., Jr., et al, "Continued Investigation of Potential Application of OMEGA Navigation to Civil Aviation." RTI Technical Report, Contract NAS1-13290, March 1978.

6. Abramowitz, M., Stegun, J. A.: Handbook of Mathematical Functions National Bureau of Standards, December 1965.

7. Wait, J. R., Electromagnetic Waves in Stratified Media, Pergamon Press, 1970.

8. Galejs, Janis, Terrestrial Propagation of Long Electromagnetic Waves, Pergamon Press, 1972.

9. Budden, K. G., Radio Waves in the Ionisphere, Cambridge University Press, 1961.

10. Morfitt, D. G., "Comparison of Waveguide and Wavehop Techniques for VLF Propagation Modeling," NWC TP 4952 (Naval Weapons Center).

11. Deeks, P. G., "D-Region electron distributions in middle latitudes deduced from the reflexion of long radio waves," Proc. Roy. Soc. 291 (1426), pp 413-427.

12. Josephy, N. H. and Kasper, J. F., Jr., "A Polynomial Approximation Technique for Small-Computer Skywave Correction Implementation," Proceedings 1st OMEGA Symposium, ION, Washington, D.C., November 1971.

13. Lytle, C. D., Bradshaw, E. S., Jowers, L. J., "A Comprehensive Experimental Program for Investigation of Various OMEGA Operational Modes with Selected Data Analysis Results," Proceedings of the 2nd OMEGA Symposium, ION, Washington, D.C., November 1974.

14. Thomas, Paul D., Conformal Projections in Geodesy and Cartography, Coast and Geodetic Survey, U.S. Dept. of Commerce, Special Pub No. 251, 1952.

REFERENCES (Continued)

15. Claire, Charles D., State Plane Coordinates by Automatic Data Processing, Coast and Geodetic Survey, ESSA, U. S. Department of Commerce, Pub 62-4, 1973.

16. Veterbi, A. J., Principles of Coherent Communication, McGraw-Hill, 1966.

17. "U.S. Air Force Projection Tables for the Lambert Conformal Conic Projection," Aeronautical Chart and Information Center, USAF, St. Louis, MO, (date unknown).

18. MCS-4 Microcomputer Set Users Manual, Intel Corporation, Santa Clara, California, September 1974, Rev. 5.

19. 4004 Macro Assembler, MAC4, Intel Corporation, Santa Clara, California, Santa Clara, California, 1974.

20. 4004 Assembly Language Programming Manual, Vers. II, Intel Corp. Santa Clara, California, 1974.