General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NASA CR-145228

R-1047

AN ALGORITHM FOR THE AUTOMATIC SYNCHRONIZATION OF OMEGA RECEIVERS

(Final Report for NASA Contract NAS1-14391)

by

William M. Stonestreet
Thomas L. Marzetta

January 1977

(NASA-CR-145228) AN ALGORITHM FOR THE AUTOMATIC SYNCHRONIZATION OF OMEGA RECEIVERS Final Report (Draper (Charles Stark) Lab., Inc.) 35 p HC A03/MF A01 CSCL 17G
N78-17033
Unclas
G3/04 04214

FEB 1978
RECEIVED
NASA STI FACILITY

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

Approved for public release; distribution unlimited.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. NASA CR-145228	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ALGORITHM FOR THE AUTOMATIC SYNCHRONIZATION OF OMEGA RECEIVERS		5. Report Date January 1977
		6. Performing Organization Code
7. Author(s) William M. Stonestreet and Thomas L. Marzetta		8. Performing Organization Report No. R-1047
9. Performing Organization Name and Address The Charles Stark Draper Laboratory, Inc. Cambridge, Massachusetts 02139		10. Work Unit No.
		11. Contract or Grant No. NAS1-14391
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23655		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An algorithm for the automatic synchronization of Omega receivers was previously developed and functionally described. This algorithm was extended, and its performance for various signal-to-noise ratios and station/frequency combinations was determined via numerical simulation. Also, the performance of the algorithm was determined for receiver-design parameters which effect the algorithm. Parameters considered are the tracking-loop input bandwidth, the data-sampling rates, and the length of the data-collecting interval. Values for these parameters which offer the best synchronization performance are suggested.

17. Key Words Suggested by Author Omega synchronization least squares receivers	18. Distribution Statement Approved for public release; distribution unlimited.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	33	

R-1047

AN ALGORITHM FOR THE AUTOMATIC SYNCHRONIZATION OF OMEGA RECEIVERS

(Final Report for NASA Contract NAS1-14391)

by

William M. Stonestreet
Thomas L. Marzetta

January 1977

Approved: William G. Denhard

William G. Denhard, Head
Air Force Programs
Department

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ACKNOWLEDGMENT

This report was prepared by The Charles Stark Draper Laboratory, Inc. under Contract NAS1-14391 with the National Aeronautics and Space Administration.

Publication of this report does not constitute approval by NASA of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1
2	THE OMEGA NAVIGATION SYSTEM	2
3	SYNCHRONIZATION ALGORITHM	5
	3.1 Optimal Signal-Vector Estimate	5
	3.2 Synchronization Algorithms	7
4	SIMULATION	14
5	RESULTS	16
6	SUGGESTED FORM OF SYNCHRONIZATION ALGORITHM AND ASSOCIATED RECEIVER-DESIGN PARAMETERS	22
LIST OF REFERENCES		23
APPENDIX A - FORTRAN LISTING OF THE SYNCHRONIZATION ALGORITHM WITH THE BEST MONTE CARLO RESULTS		24

SECTION 1

INTRODUCTION

An algorithm for the automatic synchronization of Omega receivers was previously developed and functionally described in Reference 1. This algorithm was extended, and its performance for various signal-to-noise ratios and station/frequency combinations was determined via numerical simulation. Also, the performance of the algorithm was determined for receiver-design parameters which effect the algorithm. Parameters considered are the tracking-loop input bandwidth, the data-sampling rates, and the length of the data-collecting interval. Values for these parameters which offer the best synchronization performance are suggested.

This report is divided into five main sections. First, the Omega navigation system and the requirement for receiver synchronization are discussed. Next, the synchronization algorithm is described. Then, the numerical simulation and its associated assumptions are described. The results of the simulation are presented next. Finally, the suggested form of the synchronization algorithm and the suggested receiver-design values are presented. Appendix A is a Fortran listing of the synchronization algorithm used in the simulation.

SECTION 2

THE OMEGA NAVIGATION SYSTEM

Omega is a world-wide hyperbolic radio-navigation system. The system employs eight stations around the world. The stations are located in Norway, Liberia, Hawaii, North Dakota, La ReUnion Island, Argentina, Trinidad (this is a temporary station and will be replaced with a permanent station located somewhere in the South Pacific), and Japan. Each station transmits the primary-navigation frequency of 10.2 kHz. The user measures the received phase from three or more stations to determine his position. The measurement repeats every 360 degrees. Thus, without further information, the determination of position is ambiguous. The 10.2-kHz hyperbolic lines-of-position are separated by approximately 8 nmi. Therefore, to accurately resolve position ambiguities, the user must have an initial knowledge of his position to within ±4 nmi.

The stations also transmit secondary-navigation frequencies of 13.6 and 11.33 kHz. The secondary-navigation frequencies are used to extend the distance between lines-of-position to 24 and 72 nmi for two- and three-frequency receivers, respectively. Thus, to resolve position ambiguities, the user must initially only know his position to within ±12 and ±36 nmi for two- and three-frequency receivers, respectively.

To prevent the transmissions from interfering with each other, the station/frequency transmissions are time-division multiplexed. Figure 1 depicts the Omega transmission format. Each station/frequency transmission lasts for approximately 1 second. There is a 0.2-second quiet period between transmissions. The sequence repeats every 10 seconds.

Clearly, to navigate properly, the receiver tracking loops must be synchronized to the transmitted format. The transmission format was designed such that accurate measurement of the transmission periods of any two stations at a single frequency, or of any two frequencies from a single station, allow the user to determine unambiguously the starting time of the format. When the Omega system is complete, i.e., the South

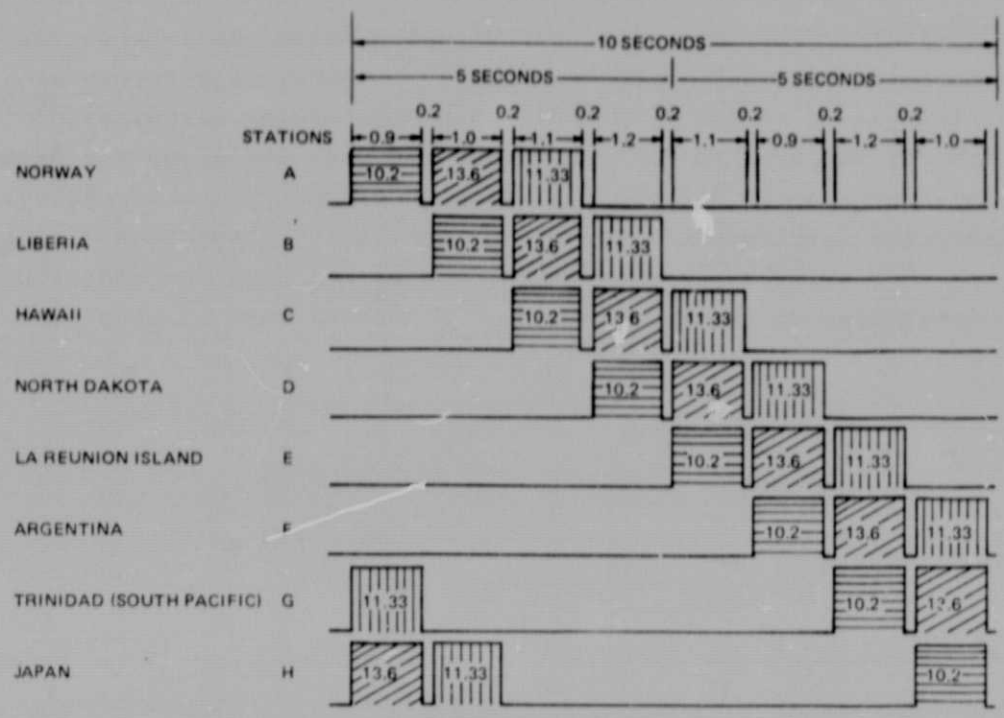


Figure 1. Omega transmission format.

Pacific station is operational and all the stations are transmitting, the starting time of the transmission format can also be unambiguously determined with a priori knowledge of the user's approximate position and the measurement of a transmission period from a single station on a single frequency. Frequently, one or more of the stations is off the air for maintenance. Thus, a synchronization algorithm cannot depend upon a priori knowledge of the stations expected to be received. However, the performance of any synchronization algorithm should improve with accurate a priori knowledge.

One method of synchronization is to sense the amplitude of the received signal and measure the period of time that it exceeds a predetermined threshold. Due to the low signal-to-noise ratios, accurate measurements require very narrow bandwidths in the amplitude-sensing circuitry. In the very low-frequency band, the predominate noise source is spike noise[2,3]* caused by lightning strikes throughout the world.

---

* Superscript numerals refer to similarly numbered references in the List of References.

Proper nonlinear discrimination against spike noise can improve the signal-to-noise ratio by 15 dB or more.[4,5] Thus, many receivers employ limiters or clippers in their phase-processing circuitry.[6,7] Therefore, if the synchronization method requires amplitude measurements, both amplitude-sensing and phase-processing circuitry are necessary. This increases both receiver complexity and cost. A synchronization algorithm that employs phase information does not need the additional amplitude-sensing circuitry. Section 3 discusses such an algorithm.

SECTION 3

SYNCHRONIZATION ALGORITHM

In this section, expressions for the optimal sine and cosine projections of the signal are determined, and two forms of the synchronization algorithm are presented.

3.1 Optimal Signal-Vector Estimate

Assume that the signal to be estimated is of the form A sin $(\omega t + \phi)$, where A and ϕ are constants, and that this signal plus noise is passed through a nonlinear system generating an output B sin $(\omega t + \theta)$, where B is constant and θ varies with time due to the changing amplitude and phase of the noise terms. Figure 2 is a polar representation of two such signals.

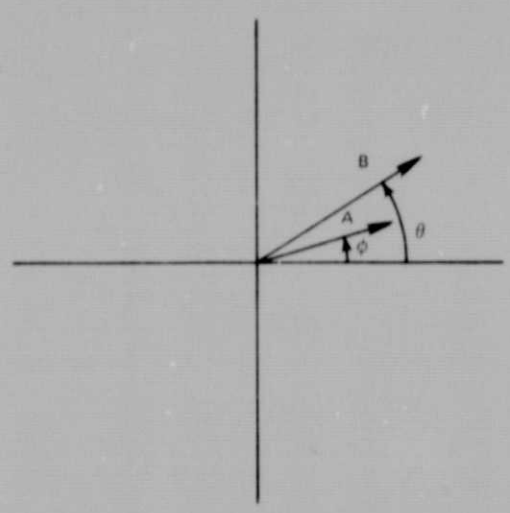


Figure 2. Polar representation of the Omega signal, A sin $(\omega t + \phi)$, and the output of the front end, B sin $(\omega t + \theta)$.

Samples B sin $(\omega t + \theta_i)$ of the output signal are then generated. Several such samples are depicted in polar form in Figure 3. The squared error between each of these samples and the input signal is

$$\epsilon_i^2 = (A \cos \phi - B \cos \theta_i)^2 + (A \sin \phi - B \sin \theta_i)^2 \qquad (1)$$

where θ_i is the phase of each sample. Equation (1) yields

$$\epsilon_i^2 = (A \cos \phi)^2 + (B \cos \theta_i)^2 - 2AB \cos \phi \cos \theta_i$$

$$+ (A \sin \phi)^2 + (B \sin \theta_i)^2 - 2AB \sin \phi \sin \theta_i \qquad (2)$$

The squared error over an interval in which more than one sample is taken is the sum of the individual squared errors, i.e.,

$$\epsilon^2 = \sum_{i=1}^{N} \epsilon_i^2 = NA^2 \cos^2 \phi + \sum_{i=1}^{N} B^2 \cos^2 \theta_i - 2A \cos \phi \sum_{i=1}^{N} B \cos \theta_i$$

$$+ NA^2 \sin^2 \phi + \sum_{i=1}^{N} B^2 \sin^2 \theta_i - 2A \sin \phi \sum_{i=1}^{N} B \sin \theta_i \qquad (3)$$

where N is the number of samples.

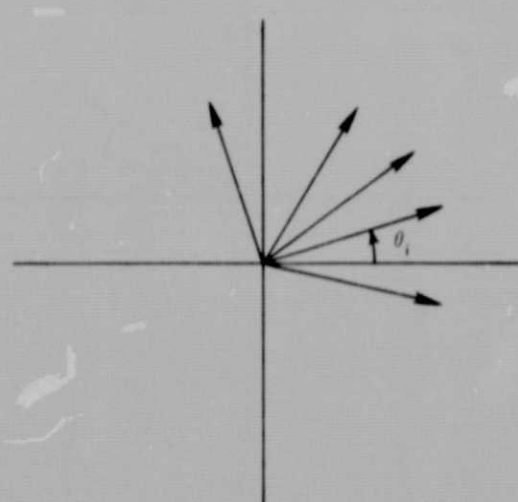


Figure 3. Polar representation of the sampled data from the front end, B sin $(\omega t + \theta_i)$.

Differentiating Eq. (3) with respect to A cos ϕ yields

$$\frac{\partial \varepsilon^2}{\partial (A \cos \phi)} = 2NA \cos \phi - 2 \sum_{i=1}^{N} B \cos \theta_i \tag{4}$$

Setting this expression equal to zero and solving for A cos ϕ gives the optimal minimum-squared estimate of A cos ϕ, which is

$$\widehat{A \cos \phi} = \frac{1}{N} \sum_{i=1}^{N} B \cos \theta_i \tag{5}$$

To determine the least-squares estimate of A sin ϕ, differentiate Eq. (3) with respect to A sin ϕ

$$\frac{\partial \varepsilon^2}{\partial (A \sin \phi)} = 2NA \sin \phi - 2 \sum_{i=1}^{N} B \sin \theta_i \tag{6}$$

Next, equate this expression with zero and solve for A sin ϕ. Thus

$$\widehat{A \sin \phi} = \frac{1}{N} \sum_{i=1}^{N} B \sin \theta_i \tag{7}$$

3.2 Synchronization Algorithms

The Omega format consists of eight transmission bursts and eight quiet periods which are repeated every 10 seconds. Thus, data from consecutive 10-second segments may be used for synchronization. For example, if we call the time at which we start taking data t_o and take 20 seconds of data, the samples taken during the intervals $[t_o, t_o + 0.9]$ and $[t_o + 10, t_o + 10.9]$ may be used to generate the signal-vector estimates for Station A. Two methods of employing data from consecutive 10-second segments are presented.

In the first method, data from consecutive 10-second segments is used to determine ensemble vector estimates of the received signal at each sampling point in the Omega format. Then, a least-square fit of the ensemble signal estimates to the local-reference Omega format is performed, and the squared error is determined by summing the squared errors between

the ensemble estimates and the local reference. The fit is iterated for many possible starting times of the local-reference format, and the time which produces the minimum-squared error is chosen as the correct starting time.

The squared error in fitting the signal-vector estimates to the local-reference format for the kth Omega burst, where k = 1, ... , 8, which correspond to Stations A through H, respectively (see Figure 1), is determined by summing the squared errors between the estimates for the kth burst and the individual sample-point estimates. However, since the expression for each of these squared errors will be of a form similar to that of Eq. (2), it is not necessary to calculate the amplitude and phase estimates, only the projections on the sine and cosine axes. These expressions can be determined by expanding Eq. (5) and (7). It can be shown that assuming B equals 1, the expanded expressions are

$$\widehat{(A \cos \phi)}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} \left[\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right]_k \tag{8}$$

and

$$\widehat{(A \sin \phi)}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} \left[\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right]_k \tag{9}$$

respectively. Where N_k equals the number of sample points in the kth burst, W equals the number of consecutive 10-second segments of data used, and k is the same as before. The expressions in brackets on the right-hand side of Eq. (8) and (9) are the projections of the jth sample-point estimates on the cosine and sine axes, respectively.

Therefore, from Eq. (2), the squared error for the jth sample point in the kth Omega burst is

$$\varepsilon_{jk}^2 = \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right]^2 + \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right]^2$$

$$+ \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k^2 + \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k^2 + \text{(see next page)}$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right] \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right] \left(\frac{1}{W} \sum_{n=1}^{W} \sin \theta_{hj} \right)_k \tag{10}$$

Thus the squared error ε_k^2 for the kth Omega burst is

$$\varepsilon_k^2 = \sum_{j=1}^{N_k} \varepsilon_{jk}^2 = N_k \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right]^2$$

$$+ N_k \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right]^2 + \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{n=1}^{W} \cos \theta_{hj} \right)_k^2$$

$$+ \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k^2$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right] \left[\sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right]$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right] \left[\sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right] \tag{11}$$

which may be written as

$$\boxed{\begin{aligned}\varepsilon_k^2 = & \sum_{j=1}^{N_K}\left(\frac{1}{W}\sum_{h=1}^{W}\sin\theta_{hj}\right)_k^2 + \sum_{j=1}^{N_k}\left(\frac{1}{W}\sum_{h=1}^{W}\cos\theta_{hj}\right)_k^2 \\ & - N_k\left[\frac{1}{N_k}\sum_{j=1}^{N_k}\left(\frac{1}{W}\sum_{h=1}^{W}\sin\theta_{hj}\right)_k\right]^2 \\ & - N_k\left[\frac{1}{N_k}\sum_{j=1}^{N_k}\left(\frac{1}{W}\sum_{h=1}^{N_k}\cos\theta_{hj}\right)_k\right]^2\end{aligned}} \tag{12}$$

where k, N_k, W, j, and h, are the same as before.

The manner in which the squared error for a quiet period is defined is not so obvious. Clearly, one may say that during this period the signal is by definition zero, and therefore the squared error in the mth quiet period, where m = 1, ... 8, is simply proportional to the square of the amplitude estimate, i.e.

$$\boxed{\begin{aligned}\varepsilon_m^2 = & N\left[\frac{1}{N}\sum_{\ell=1}^{N}\left(\frac{1}{W}\sum_{h=1}^{W}\sin\theta_{\ell h}\right)\right]_m^2 \\ & + N\left[\frac{1}{N}\sum_{\ell=1}^{N}\left(\frac{1}{W}\sum_{h=1}^{W}\cos\theta_{\ell h}\right)\right]_m^2\end{aligned}} \tag{13}$$

where N is the number of sample points (which is the same for all quiet periods since they are of equal duration) in the quiet period, and as before, W is the number of consecutive 10-second segments employed in the computation.

However, the degree to which zero can be determined in the receiver is directly dependent upon the number of samples, and therefore the sampling rate. Thus, for a fixed sampling rate there is a minimum nonzero value which will occur during the actual quiet period. Thus, some of the squared

error in the quiet periods, as determined by Eq. (13), is due to inherent receiver inaccuracies. An alternative method is to assume that the squared error in each quiet period is zero, i.e.,

$$\varepsilon_m^2 = 0 \quad \forall m \tag{14}$$

Equations (12) and (13) or (14) may be used to determine the squared error for the entire 10-second format, i.e.

$$\varepsilon^2 = \sum_{k=1}^{8} \varepsilon_k^2 + \sum_{m=1}^{8} \varepsilon_m^2 \tag{15}$$

This process is iterated for many possible starting times of the local-reference format, and the time which generates the minimum-squared error is chosen as the correct starting time, and the local commutator is reset to this starting time.

In the second method, instead of attempting to determine the squared error between the ensemble sample-point estimates and the ensemble station estimates (as is done in the first method), the squared error between the individual sample points and the ensemble station estimates is determined. These errors are summed to produce a total squared error for a particular starting time of the local-reference format. This process is iterated for many possible starting times of the local-reference format, and the time which generates the minimum-squared error is chosen as the correct starting time.

Using Eq. (2), (8), and (9), the squared error for the jth sample point in the kth Omega burst of the hth 10-second segment is

$$\varepsilon_{hjk}^2 = \left[\frac{1}{N_k}\sum_{j=1}^{N_k}\left(\frac{1}{W}\sum_{h=1}^{W}\cos\theta_{hj}\right)_k\right]^2$$

$$+ \left[\frac{1}{N_k}\sum_{j=1}^{N_k}\left(\frac{1}{W}\sum_{h=1}^{W}\sin\theta_{hj}\right)_k\right]^2 \quad + \text{(see next page)}$$

$$+ (\cos \theta_{hj})_k^2 + (\sin \theta_{hj})_k^2$$

$$- 2\left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj}\right)_k\right] \cos \theta_{hjk}$$

$$- 2\left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj}\right)_k\right] \sin \theta_{hjk} \tag{16}$$

where N_k and W are the same as before. In this case, the squared error for the kth Omega burst is

$$\epsilon_k^2 = \sum_{j=1}^{N_k} \sum_{h=1}^{W} \epsilon_{hjk}^2 = N_k W \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj}\right)_k\right]^2$$

$$+ N_k W \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj}\right)_k\right]^2$$

$$+ N_k W (\cos \theta_{hj})_k^2 + N_k W (\sin \theta_{hj})_k^2$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj}\right)_k\right] \left[\sum_{j=1}^{N_k} \sum_{h=1}^{W} \cos \theta_{hjk}\right]$$

$$- 2 \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj}\right)_k\right] \left[\sum_{j=1}^{N_k} \sum_{h=1}^{W} \sin \theta_{hjk}\right] \tag{17}$$

Equation (17) can be rewritten as

$$\boxed{\varepsilon_k^2 = N_k W \left\{ 1 - \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \cos \theta_{hj} \right)_k \right]^2 - \left[\frac{1}{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{W} \sum_{h=1}^{W} \sin \theta_{hj} \right)_k \right]^2 \right\}} \tag{18}$$

Equations (18) and (13) or (14) are used to determine the squared error for the entire format, i.e.

$$\varepsilon^2 = \sum_{k=1}^{8} \varepsilon_k^2 + \sum_{m=1}^{8} \varepsilon_m^2 \tag{19}$$

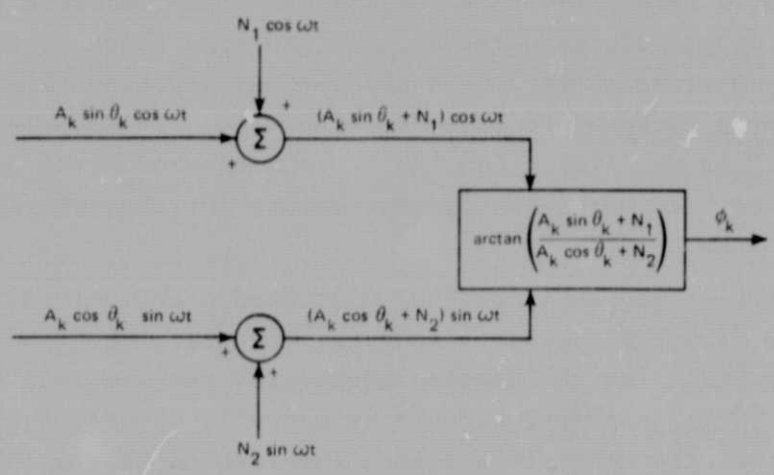
As before, this process is iterated for many possible starting times of the local-reference format, and the time which generates the minimum squared error is chosen as the correct starting time, and the local commutator is reset to this starting time. If Eq. (14) is used to estimate ε_m^2 in Eq. (19), then the minimum squared error will occur at the starting time which has maximum values for the two right-most expressions of Eq. (18).

The synchronization algorithms discussed previously do not employ any type of a priori knowledge. However, the algorithms should have the capability of utilizing information concerning the stations expected to be received. This information could be manually entered into the receiver by the operator. If accurate a priori information is available, the performance of the synchronization algorithm should improve. Both of the synchronization algorithms discussed can be easily modified to incorporate a priori knowledge. Stations whose signals are not expected to be received can be treated in the same manner as a quiet period, and either Eq. (13) or (14) can be used to determine the squared errors for those stations.

SECTION 4

SIMULATION

The performance of the synchronization algorithms presented in Section 3 was determined via Monte Carlo techniques. The model used to simulate the inputs to the synchronization algorithms during periods in which a signal is received is shown in Figure 4. For each transmission



N_1 AND N_2 ARE STATISTICALLY INDEPENDENT GAUSSIAN WHITE-NOISE PROCESSES WITH ZERO MEANS. THEIR VARIANCES ARE OPERATOR SELECTABLE. A_k AND θ_k ARE THE RECEIVED AMPLITUDE AND PHASE FROM THE kth OMEGA STATION. A_k IS OPERATOR SELECTABLE, AND θ_k IS RANDOMLY CHOSEN AT THE START OF EACH RUN.

Figure 4. Model used to simulate the received signal from the kth Omega station.

burst, the model generates the phase of a constant-amplitude signal from a constant-phase constant-amplitude signal and additive white Gaussian channel noise. The signal from the kth burst, $A_k \sin \phi_k$, is broken into sine and cosine components, $A_k \cos \theta_k \sin \omega t$ and $A_k \sin \theta_k \cos \omega t$, respectively. Then, white Gaussian channel-noise processes, $N_1 \cos \omega t$ and $N_2 \sin \omega t$, are added to the appropriate quadrature inputs. The input phase to the synchronization algorithms, θ_k, is determined by taking the arctangent of $(A_k \sin \theta_k + N_1)$ divided by $(A_k \cos \theta_k + N_2)$. This produces an input phase that is a combination of the phase of the received signal and a pseudo-random phase due to the noise processes. The amplitude of the input signal is set to unity, and hence, the power of the input to the synchronization algorithm is fixed, as would be the case if a limiter preceded the synchronization algorithm. The amplitude of the received signal, A_k, is selected by the operator, and may be different for each Omega signal. The phase of the received signal, θ_k, is randomly chosen at the beginning of each Monte Carlo run. θ_k is uniformly distributed between $[-\pi, \pi]$. The input to the synchronization algorithm during the quiet periods, and during transmission bursts when no signal is received, is also uniformily distributed between $[-\pi, \pi]$.

The input to the synchronization algorithms is sampled every ΔT seconds, where ΔT inverse is an integral multiple of 10 Hz. For the Monte Carlo runs, ΔTs of 0.1 and 0.05 second were used. The starting time of the Omega transmission format can be offset from that of the local reference by integral multiples of 0.025 second.

SECTION 5

RESULTS

This section presents the results of Monte Carlo runs of the simulation described in Section 4. The performance of the synchronization algorithms described in Section 3.2 is determined for various station/frequency pairs, signal-to-noise ratios, data-sampling rates, and data-collection intervals. Two station/frequency combinations were used in the comparisons: the A,D 10.2-kHz transmission bursts, and the C,D 10.2-kHz transmission bursts (the results are valid for any combination of station/frequencies transmitting during these time slots). These combinations were chosen for three reasons. First, they can all be received in the continental United States. Second, the A,D combination has a large time separation, 2.7 seconds between transmission bursts, and the difference in the duration of the two bursts, 0.3 second, is a maximum (see Figure 1). Third, the C,D combination has the minimum separation time, 0.2 second, and the minimum difference in burst duration, 0.1 second. Thus, synchronization using data from the C,D combination should be difficult, whereas synchronization using data from the A,D combination should be relatively easy. Two sampling rates, 0.1 second and 0.05 second, were considered.

Table 1 presents the Monte Carlo results for the synchronization algorithm where the squared error is determined from Eq. (12) and (13). Notice that the performance of the algorithm increases significantly with the use of a priori knowledge. Also, there is a significant performance difference between the A,D and the C,D combinations when a priori knowledge is not used. However, at low signal-to-noise ratios the algorithm does not perform very well.

Table 2 presents the Monte Carlo results for the synchronization algorithm employing Eq. (12) and (14). Here again, the performance of the algorithm is increased with the use of a priori knowledge, and there is a significant difference in algorithm performance between the A,D and

Table 1. Monte Carlo results for the synchronization algorithm employing Eq. (12) and (13).

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 50)	With Prior Knowledge (Out of 50)
A,D	0.1	10	0	50	50
A,D	0.1	10	-10	50	50
A,D	0.1	10	-20	17	29
C,D	0.1	10	0	49	50
C,D	0.1	10	-10	49	50
C,D	0.1	10	-20	10	29

Table 2. Monte Carlo results for the synchronization algorithm employing Eq. (12) and (14).

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 50)	With Prior Knowledge (Out of 50)
A,D	0.1	10	0	35	50
A,D	0.1	10	-10	33	49
A,D	0.1	10	-20	6	7
C,D	0.1	10	0	22	50
C,D	0.1	10	-10	19	50
C,D	0.1	10	-20	4	3

C,D combinations without a priori knowledge. Comparing Tables 1 and 2 indicates that the synchronization algorithm used to generate the data in Table 1 performs better than the synchronization algorithm used to generate the data in Table 2. The only difference between these two algorithms is the manner in which they treat quiet periods. The algorithm used in the Monte Carlo runs for Table 1 used the data during quiet periods to produce an error term; the algorithm associated with Table 2 did not.

The Monte Carlo results for the synchronization algorithm employing Eq. (18) and (13) are presented in Table 3. In all cases, the performance of this algorithm is better, or about the same, as the performance of the algorithms associated with Tables 1 and 2. The main difference between these algorithms is the manner in which data from consecutive 10-second segments is used to generate the squared error. In this algorithm, the squared error between the individual sample point and the ensemble station estimates is determined. In the algorithms associated with Tables 1 and 2, the squared error between the ensemble sample-point estimates and the ensemble station estimates is determined.

Table 3. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (13).

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 50)	With Prior Knowledge (Out of 50)
A,D	0.1	10	0	50	50
A,D	0.1	10	-10	50	50
A,D	0.1	10	-20	18	35
C,D	0.1	10	0	50	50
C,D	0.1	10	-10	49	50
C,D	0.1	10	-20	22	36

Table 4 presents the Monte Carlo results for the synchronization algorithm employing Eq. (18) and (14). This algorithm differs from the algorithm associated with Table 3 in that it does not make use of data taken during quiet periods. The performance of this algorithm is better than that of any of the other algorithms.

All of the results presented sample the input signal at a 10-Hz rate. The effects of a 20-Hz sampling rate for the algorithms associated with Table 4 are indicated in Table 5. From Table 4, the probability of synchronization for the A,D station combination with a sampling rate of 10 Hz and a signal-to-noise ratio of -20 dB in a 100-Hz bandwith, is 60%. The corresponding probability of synchronization for the C,D station combination is 46%. For a 20-Hz sampling rate and the same signal-to-noise ratio, the probability of synchronization for the A,D and C,D station combinations was 70% and 60%, respectively. With a priori knowledge, this algorithm was able to correctly identify the starting time of the Omega transmission format every time.

Table 4. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (14).

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 50)	With Prior Knowledge (Out of 50)
A,D	0.1	10	0	50	50
A,D	0.1	10	-10	50	50
A,D	0.1	10	-20	30	46
C,D	0.1	10	0	50	50
C,D	0.1	10	-10	50	50
C,D	0.1	10	-20	23	46

Table 5. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (14).

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 10)	With Prior Knowledge (Out of 10)
A,D	0.05	10	0	10	10
A,D	0.05	10	-10	10	10
A,D	0.05	10	-20	7	10
C,D	0.05	10	0	10	10
C,D	0.05	10	-10	10	10
C,D	0.05	10	-20	6	10

As yet, the effects of the time interval over which data is collected have not been discussed. Table 6 presents the Monte Carlo results of the synchronization algorithm associated with Tables 4 and 5 for three data-collection intervals, 100 seconds (which correspond to 10 consecutive 10-second segments of Omega data), 30 seconds, and 10 seconds. Clearly, for low signal-to-noise ratios, 10 or 30 seconds of data is not adequate, whereas, with 100 seconds of data, the algorithm performed well.

Table 6. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (14) for various data-collection intervals.

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 10)	With Prior Knowledge (Out of 10)
A,D	0.05	10	-17	9	10
A,D	0.05	3	-17	2	7
A,D	0.05	1	-17	1	5
A,D	0.1	10	-17	9	10
A,D	0.1	3	-17	1	6
A,D	0.1	1	-17	0	0

Since the algorithm associated with Tables 4 through 6 does not make use of the data collected during quiet periods, it is possible for a station burst of 1.0-second duration to produce zero error when the data is fit to a local-reference burst with a 0.9-second duration. This type of error can occur when the Omega transmission format is offset from the local-reference format by one-half the sampling period. In the same manner, station bursts of 1.2 and 1.1 second will produce zero error when compared with local-reference bursts at 1.1 and 1.0 second, respectively. However, this effect decreases as the sampling period decreases, and is less pronounced when the stations received are not consecutive transmission bursts in the Omega format. Table 7 presents the Monte Carlo results of this algorithm for a 25-ms offset between the transmitted format and the local-reference format. A comparison of Tables 5 and 7 indicates that with a sampling period of 0.05 second, the effect of the 25-ms offset is negligible. Table 8 presents the Monte Carlo results of the synchronization algorithm that employs Eq. (18) and (13), and thus uses the data in the quiet periods to generate an error. Clearly, the performance of this algorithm is worse than the performance of the algorithm associated with Table 7, which does not make use of data in the quiet periods to generate an error.

Table 7. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (14) with a 25-ms offset between the transmitted format and the local-reference format.

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 10)	With Prior Knowledge (Out of 10)
A,D	0.05	10	0	10	10
A,D	0.05	10	-10	10	10
A,D	0.05	10	-20	7	10
C,D	0.05	10	0	10	10
C,D	0.05	10	-10	10	10
C,D	0.05	10	-20	5	10

Table 8. Monte Carlo results for the synchronization algorithm employing Eq. (18) and (13) with a 25-ms offset between the transmitted format and the local-reference format.

Stations	ΔT (s)	No. of Periods	SNR (dB) BW = 100 Hz	No. of Correct Synchronizations	
				Without Prior Knowledge (Out of 10)	With Prior Knowledge (Out of 10)
A,D	0.05	10	0	10	10
A,D	0.05	10	-10	10	10
A,D	0.05	10	-20	5	8
C,D	0.05	10	0	10	10
C,D	0.05	10	-10	9	10
C,D	0.05	10	-20	2	8

SECTION 6

SUGGESTED FORM OF SYNCHRONIZATION ALGORITHM AND ASSOCIATED RECEIVER-DESIGN PARAMETERS

In Section 5, it was shown that for the cases considered the form of the synchronization algorithm with the best performance attempts to minimize Eq. (19), where ϵ_k^2 is determined by Eq. (18) and ϵ_m^2 equals zero for all m. Equation (18) is minimized when the two right-most expressions are maximized. Thus, in terms of performance, an equivalent implementation of the algorithm is to choose the starting time of the local reference which maximizes the quantity

$$\sum_{k=1}^{8}\left\{\left[\sum_{j=1}^{N}\sum_{h=1}^{W}\cos\theta_{hj}\right]_k^2 + \left[\sum_{j=1}^{N_k}\sum_{h=1}^{W}\sin\theta_{hj}\right]_k^2\right\} \tag{20}$$

where k, N_k, and W are the same as before. This implementation has the advantage of requiring less computer operations, and thus less time, than the implementation that employs Eq. (18). The results reported in Section 5 indicate that a good choice for W, the number of consecutive 10-second segments of data collected, is 10. The results also indicate that the input bandwidth to the synchronization algorithm should be 10 Hz, and the data-sampling period should be 0.05 second. However, if a wider bandwidth is desired in the phase-tracking circuitry, then the sampling period can be decreased, and consecutive data samples can be averaged to generate 0.05-second samples. For example, if a phase-tracking bandwidth of 100 Hz is desired, then the sampling period should be 0.005 second, and 10 consecutive data samples should be averaged to form a 0.05-second sample.

LIST OF REFERENCES

1. Cox, D., W. Stonestreet, et al., Digital Phase Processing for Low-Cost Omega Receivers, Draper Laboratory Report P-190, presented at the Thirtieth Technical Meeting on Medium Accuracy Low-Cost Navigation sponsored by the Avionic Panel at AGARD, 8-12 September 1975.

2. Feldman, J., An Atmospheric Model with Application to Low Frequency Navigation Systems, MIT Sc.D. Thesis, June 1972.

3. Maxwell, E., and D. Stone, ELF and ULF Atmospheric Noise, DECO Electronics Report AD-428273, 10 January 1964.

4. Pierce, J., Omega: A World-Wide Navigational System: System Specification and Implementation, AD-630-900, 1 May 1966.

5. Postema, L., Performance Optimization of Linear Loran Receivers Operating in a Non-Gaussian Atmospheric Noise Environment, presented at The Third Annual Convention of the Wild Goose Association, 4 October 1972.

6. Omega Navigation Set AN/ARN-99 (XN-2), Final Engineering Report, Northrop Corporation Report AD-775-598, December 1973.

7. McTaggart, D., "Design and Performance of CMC Computerized Airborne Omega Receiver", Proceedings of the First Omega Symposium, ION, 9-11 November 1971.

APPENDIX A

FORTRAN LISTING OF THE SYNCHRONIZATION ALGORITHM WITH THE BEST MONTE CARLO RESULTS

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      SUBROUTINE MCODIC
C
C
C
      REAL DB(1000)
      REAL PHI(8)
      REAL A(8)
      REAL X(1000)
      REAL Y(1000)
      REAL Z(1000)
      REAL DATAMP(1000)
      REAL DATPHI(1000)
      INTEGER NW(16)
      INTEGER NB(16)
      COMMON /DATBAS/DB
      COMMON /NCOMUT/NW,NB
      COMMON /IDATGN/J1,J2
      COMMON /IENSAV/J4
      COMMON /ENSDAT/X,Y
      COMMON /IS/ISYNC0,ISYNC1
      COMMON /DATPOL/DATAMP,DATPHI
      COMMON /TH/THETA
      COMMON /IMCDUD/IM
      COMMON /EF/E
      EQUIVALENCE (DB(0201),A(1)),(DB(0202),A(2)),(DB(0203),A(3)),
     1(DB(0204),A(4)),(DB(0205),A(5)),(DB(0206),A(6)),
     2(DB(0207),A(7)),(DB(0208),A(8)),(DB(0209),PHI(1)),
     3(DB(0210),PHI(2)),(DB(0211),PHI(3)),(DB(0212),PHI(4)),
     4(DB(0213),PHI(5)),(DB(0214),PHI(6)),(DB(0215),PHI(7)),
     5(DB(0216),PHI(8)),(DB(0226),RNPER),(DB(0227),RNSAMP),
     6(DB(0228),PHIRAN),(DB(0231),ERROR),(DB(0232),DATAM),
     7(DB(0233),DATPH)
C
C
      NSAMP=IFIX(RNSAMP)
      NPER=IFIX(RNPER)
C     INITIALIZE VARIABLES IN OTHER SUBROUTINES.
      J1=1
      J2=1
      J4=1
      IM=1
      DO 1 J=1,NSAMP
      X(J)=0.
    1 Y(J)=0.
C
C
C     IF PHIRAN=0, THEN A PHASE MUST BE SPECIFIED FOR EACH STATION
C     WITH NON-ZERO AMPLITUDE.  IF PHIRAN=1, THEN A UNIFORM RANDOM
C     PHASE IS GENERATED FOR EACH STATION HAVING NON-ZERO AMPLITUDE.
      IF (PHIRAN .EQ. 0.) GO TO 3
      DO 2 I1=1,8
      IF (A(I1) .EQ. 0.) GO TO 2
      PHI(I1)=(URNDF(0.)-.5)*2.0*3.14159265
    2 CONTINUE
C
C
C     INDICES ARE NOW COMPUTED TO REPRESENT THE OMEGA COMMUTATOR.
    3 NW(1)=(9*NSAMP)/100
      NB(1)=1
      NW(2)=(2*NSAMP)/100
```

```
      NB(2)=NB(1)+NW(1)
      NW(3)=(10*NSAMP)/100
      NB(3)=NB(2)+NW(2)
      NW(4)=NW(2)
      NB(4)=NB(3)+NW(3)
      NW(5)=(11*NSAMP)/100
      NB(5)=NB(4)+NW(4)
      NW(6)=NW(2)
      NB(6)=NB(5)+NW(5)
      NW(7)=(12*NSAMP)/100
      NB(7)=NB(6)+NW(6)
      NW(8)=NW(2)
      NB(8)=NB(7)+NW(7)
      NW(9)=(11*NSAMP)/100
      NB(9)=NB(8)+NW(8)
      NW(10)=NW(2)
      NB(10)=NB(9)+NW(9)
      NW(11)=(9*NSAMP)/100
      NB(11)=NB(10)+NW(10)
      NW(12)=NW(2)
      NB(12)=NB(11)+NW(11)
      NW(13)=(12*NSAMP)/100
      NB(13)=NB(12)+NW(12)
      NW(14)=NW(2)
      NB(14)=NB(13)+NW(13)
      NW(15)=(10*NSAMP)/100
      NB(15)=NB(14)+NW(14)
      NW(16)=NW(2)
      NB(16)=NB(15)+NW(15)
C
C
      DO 5 I2=1,NPER
      DO 4 I3=1,NSAMP
      CALL DATGEN
      CALL ENSAVG
    4 CONTINUE
    5 CONTINUE
      CALL SYNCH
      CALL CRUNCH
C
      ERROR=E(1)
      DATAM=DATAMP(1)
      DATPH=DATPHI(1)
C
C
C
      RETURN
C
      END
      SUBROUTINE DATGEN
C
C
C
      REAL DB(1000)
      REAL PHI(8)
      REAL A(8)
      INTEGER NW(16)
      INTEGER NB(16)
      COMMON /IDATGN/J1,J2
      COMMON /NCOMUT/NW,NB
```

```
      COMMON /TH/THETA
      COMMON /DATBAS/DB
      EQUIVALENCE (DB(0201),A(1)),(DB(0202),A(2)),(DB(0203),A(3)),
     1(DB(0204),A(4)),(DB(0205),A(5)),(DB(0206),A(6)),(DB(0207),A(7)),
     2(DB(0208),A(8)),(DB(0209),PHI(1)),(DB(0210),PHI(2)),
     3(DB(0211),PHI(3)),(DB(0212),PHI(4)),(DB(0213),PHI(5)),
     4(DB(0214),PHI(6)),(DB(0215),PHI(7)),(DB(0216),PHI(8)),
     5(DB(0229),STDEV)
C
C
C     J1, RANGING BETWEEN 1 AND 16, REPRESENTS THE SIGNAL PERIOD
C     CURRENTLY BEING GENERATED.  J2, RANGING BETWEEN 1 AND NW(.),
C     REPRESENTS THE PRESENT SAMPLE WITHIN THE J1 SIGNAL PERIOD.
      J3=(J1+1)/2
      WC=GAUSF(STDEV/SQRT(2.))
      WS=GAUSF(STDEV/SQRT(2.))
      J4=2*J3-J1
      IF (J4 .EQ. 0) GO TO 1
      IF (A(J3) .EQ. 0.) GO TO 1
C
      CTHETA=A(J3)*COS(PHI(J3))+WC
      STHETA=A(J3)*SIN(PHI(J3))+WS
      GO TO 2
C
    1 CTHETA=WC
      STHETA=WS
C
    2 THETA=ATAN2(STHETA,CTHETA)
C
C
      J2=J2+1
      IF (J2 .LE. NW(J1)) GO TO 3
      J2=1
      J1=J1+1
      IF (J1 .GT. 16) J1=1
    3 CONTINUE
C
      RETURN
C
C
C
      END
      SUBROUTINE ENSAVG
C
C
C
      REAL X(1000)
      REAL Y(1000)
      REAL DB(1000)
      COMMON /DATBAS/DB
      COMMON /ENSDAT/X,Y
      COMMON /TH/THETA
      COMMON /IENSAV/J4
      EQUIVALENCE (DB(0226),RNPER),(DB(0227),RNSAMP)
C
C
      NSAMP=IFIX(RNSAMP)
      X(J4)=X(J4)+COS(THETA)/RNPER
      Y(J4)=Y(J4)+SIN(THETA)/RNPER
      J4=J4+1
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS

```
      IF (J4 .GT. NSAMP) J4=1
C
C
      RETURN
C
C
C
      END
      SUBROUTINE SYNCH
C
C
C
      REAL DB(1000)
      REAL X(1000)
      REAL Y(1000)
      REAL E(1000)
      REAL B(9)
      REAL ET(16)
      INTEGER NW(16)
      INTEGER NB(16)
      COMMON /NCOMUT/NW,NB
      COMMON /ENSDAT/X,Y
      COMMON /EE/E
      COMMON /DATBAS/DB
      COMMON /IS/ISYNC0,ISYNC1
      EQUIVALENCE (DB(0217),B(1)),(DB(0218),B(2)),(DB(0219),B(3)),
     1(DB(0220),B(4)),(DB(0221),B(5)),(DB(0222),B(6)),
     2(DB(0223),B(7)),(DB(0224),B(8)),(DB(0225),B(9)),
     3(DB(0227),RNSAMP),(DB(0230),TSYNC0),(DB(0266),TSYNC1),
     4(DB(0267),E0),(DB(0268),E1)
C
      EMAX0=0.0
      EMAX1=0.0
      NSAMP=IFIX(RNSAMP)
C
C     THE WEIGHTED MEAN-SQUARE ERROR IS NOW COMPUTED FOR EACH
C     POSSIBLE STARTING POINT, AND THE STARTING POINT CORRESPONDING
C     TO THE SMALLEST MEAN-SQUARE ERROR IS ASSIGNED TO ISYNCH.
      DO 6 I1=1,NSAMP
C
C     THE MEAN-SQUARE ERROR ET(1), ET(3),...,ET(15) IS NOW
C     COMPUTED FOR EACH OF THE 8 SIGNAL PERIODS.
      DO 3 I2=1,15,2
      I3=(I2+1)/2
      ET(I2)=0.
      IF (B(I3) .EQ. 0.0) GO TO 3
      XAVG=0.
      YAVG=0.
      K1=NW(I2)
C
      DO 1 I4=1,K1
      K2=1+MOD((NB(I2)+I4+I1-3),NSAMP)
      XAVG=XAVG+X(K2)
    1 YAVG=YAVG+Y(K2)
      ET(I2)=(XAVG**2+YAVG**2)/FLOAT(NW(I2))
    3 CONTINUE
C
C
C
C
```

```
      E(I1)=ET(1)+ET(3)+ET(5)+ET(7)+ET(9)+ET(11)+ET(13)+ET(15)
C
      IF (E(I1) .LE. EMAX0) GO TO 7
      ISYNC0=I1
      EMAX0=E(I1)
    7 IF (E(I1) .LE. EMAX1 .OR. E(I1) .GE. EMAX0) GO TO 6
      ISYNC1=I1
      EMAX1=E(I1)
    6 CONTINUE
      TSYNC0=FLOAT((ISYNC0-1)*10)/RNSAMP
      TSYNC1=FLOAT((ISYNC1-1)*10)/RNSAMP
      E0=E(ISYNC0)
      E1=E(ISYNC1)
C
      RETURN
C
C
C
      END
      SUBROUTINE CRUNCH
C
      REAL AEST(16)
      REAL PHIEST(16)
      REAL X(1000)
      REAL Y(1000)
      REAL DATAMP(1000)
      REAL DATPHI(1000)
      REAL DB(1000)
      INTEGER NW(16)
      INTEGER NB(16)
      COMMON /DATBAS/DB
      COMMON /NCOMUT/NW,NB
      COMMON /RNSDAT/X,Y
      COMMON /DATPOL/DATAMP,DATPHI
      COMMON /IS/ISYNC0,ISYNC1
      EQUIVALENCE (DB(0234),AEST(1)),(DB(0235),AEST(2)),
     1(DB(0236),AEST(3)),(DB(0237),AEST(4)),(DB(0238),AEST(5)),
     2(DB(0239),AEST(6)),(DB(0240),AEST(7)),(DB(0241),AEST(8)),
     3(DB(0242),AEST(9)),(DB(0243),AEST(10)),(DB(0244),AEST(11)),
     4(DB(0245),AEST(12)),(DB(0246),AEST(13)),(DB(0247),AEST(14)),
     5(DB(0248),AEST(15)),(DB(0249),AEST(16)),(DB(0250),PHIEST(1)),
     6(DB(0251),PHIEST(2)),(DB(0252),PHIEST(3)),(DB(0253),PHIEST(4)),
     7(DB(0254),PHIEST(5)),(DB(0255),PHIEST(6)),(DB(0256),PHIEST(7)),
     8(DB(0257),PHIEST(8)),(DB(0258),PHIEST(9)),(DB(0259),PHIEST(10)),
     9(DB(0260),PHIEST(11)),(DB(0261),PHIEST(12)),(DB(0262),PHIEST(13))
      EQUIVALENCE (DB(0263),PHIEST(14)),(DB(0264),PHIEST(15)),
     1(DB(0265),PHIEST(16)),(DB(0227),RNSAMP)
C
      NSAMP=IFIX(RNSAMP)
      DO 1 I1=1,NSAMP
      DATAMP(I1)=SQRT(X(I1)**2+Y(I1)**2)
    1 DATPHI(I1)=ATAN2(Y(I1),X(I1))
C
      DO 3 I2=1,16
      XAVG=0.
      YAVG=0.
      K1=NW(I2)
      DO 2 I3=1,K1
      K2=1+MOD((NB(I2)+I3+ISYNC0-3),NSAMP)
      XAVG=XAVG+X(K2)
```

```
    2 YAVG=YAVG+Y(K2)                                                   00003260
      RNW=FLOAT(NW(I2))                                                 00003270
      XAVG=XAVG/RNW                                                     00003280
      YAVG=YAVG/RNW                                                     00003290
      AEST(I2)=SQRT(XAVG**2+YAVG**2)                                    00003300
    3 PHIEST(I2)=ATAN2(YAVG,XAVG)                                       00003310
C                                                                       00003320
      RETURN                                                            00003330
C                                                                       00003340
C                                                                       00003350
C                                                                       00003360
      END                                                               00003370
      SUBROUTINE MCODEV                                                 00003380
C                                                                       00003390
C                                                                       00003400
C                                                                       00003410
      RETURN                                                            00003420
C                                                                       00003430
C                                                                       00003440
C                                                                       00003450
      END                                                               00003460
      SUBROUTINE MCODUD                                                 00003470
C                                                                       00003480
C                                                                       00003490
C                                                                       00003500
      REAL DB(1000)                                                     00003510
      REAL E(1000)                                                      00003520
      REAL DATAMP(1000)                                                 00003530
      REAL DATPHI(1000)                                                 00003540
      COMMON /DATBAS/DB                                                 00003550
      COMMON /EE/E                                                      00003560
      COMMON /DATPOL/DATAMP,DATPHI                                      00003570
      COMMON /IMCDUD/IM                                                 00003580
      EQUIVALENCE (DB(0231),ERROR),(DB(0232),DATAM),(DB(0233),DATPH)    00003590
      IM=IM+1                                                           00003600
      ERROR=E(IM)                                                       00003610
      DATAM=DATAMP(IM)                                                  00003620
      DATPH=DATPHI(IM)                                                  00003630
C                                                                       00003640
      RETURN                                                            00003650
C                                                                       00003660
C                                                                       00003670
C                                                                       00003680
      END                                                               00003690
```