

AD-A077 358

WYOMING UNIV LARAMIE DEPT OF MECHANICAL ENGINEERING F/G 16/1
RESEARCH ON TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS--ETC(U)
OCT 79 K M PELL, J E NYDAHL DAAG29-77-G-0090
UNCLASSIFIED UWME-DR-906-101-1 NL

1 of 1
AD
A077358



END
DATE
FILMED
R-79
DDC

ARO 14855.2-A-E

(12)

AD A 077358

RESEARCH ON
TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS

FINAL REPORT

LEVEL

KYNRIC M. PELL
and
JOHN E. NYDAHL

October, 1979

U.S. ARMY RESEARCH OFFICE

GRANT NO. DAAG 29-77-G-0090

UNIVERSITY OF WYOMING
LARAMIE, WYOMING

DDC
RECEIVED
NOV 28 1979
A

DDC FILE COPY

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

79 11 26 138

3-A-C 2824 09A



THE FINDINGS IN THIS REPORT ARE NOT TO BE
CONSTRUED AS AN OFFICIAL DEPARTMENT OF
THE ARMY POSITION, UNLESS SO DESIGNATED
BY OTHER AUTHORIZED DOCUMENTS.

DDC
RESEARCH
A

DDC LIFE COBY

28 11 58 138

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. 14	3. RECIPIENT'S CATALOG NUMBER UWME-DR-9061011
4. TITLE (and Subtitle) RESEARCH ON TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS.		5. TYPE OF REPORT & PERIOD COVERED 9 Final Report - Feb 77 - Apr 79
7. AUTHOR(s) Kynric M. Pell & John E. Nydahl		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Wyoming Department of Mechanical Engineering Laramie, WY 82071		8. CONTRACT OR GRANT NUMBER(s) 15 DAAG 29-77-G-0090 Feb 77 - April 79
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 60
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 11 31 October 1979
		13. NUMBER OF PAGES 54
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Missiles, rockets, attitude, lasers, instrumentation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A new concept, employing ground-based lasers, for attitude measurement of flight vehicles on test ranges is described. Two ground-based laser stations are required. Each station consists of a standard laser radar unit with the addition of a continuous-wave laser as well as transmitting and detecting optics for the cw laser. Cooperative vehicles incorporating two roof prisms, in addition to standard retroreflecting elements for tracking are required. → next page		

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (Continued)

cont. → The mathematical algorithms and software required to determine vehicle attitude from retroreflected pulse data are presented. Hardware required for application to non-spinning vehicles is described. A single laser-ground station has been installed on Range 1 of Redstone and has been used to determine the space position of missiles, rockets and aircraft. Hardware problems with the attitude subsystem have precluded verification of the attitude measurement concept.



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Table of Contents

	Page
List of Figures	ii
List of Appendixes	iii
Introduction	1
Description of the concept	1
Reformulation of the mathematical description	6
Vehicle position determination	10
Application to non-rolling vehicles	12
Status of the LAMPAMS	14
Conclusions and Recommendations	14
References	15

Accession For	
NTIS GR&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification _____	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or special
A	23 CP

List of Figures

Figure	Title	Page
1	Geometric optics of a roof prism	2
2	Geometry for the development of the mathematical model	2
3	Vehicle equipped with two prisms	4
4	Retroreflection planes referenced to a laser ground station at two different times	6
5	Definition of $\hat{\epsilon}_{\tau i}$	7
6	Definition of α	7
7	Geometric relationship	8
8	LAMPAMS magnetic tape format	11
9	Reworked 2.75 inch fuse assembly	13

List of Appendixes

Appendix	Title	Page
I	ASP11 Program to determine attitude of vehicle	16
II	LTPOS Program to determine vehicle position	41

Introduction

The objective of this study was to investigate techniques for improving missile and rocket test range instrumentation. Specifically, a laser concept for measurement of vehicle position and attitude originally proposed by Conard and Pell (1,2,3) was to be exploited.

Description of the concept.

The concept has been described in detail elsewhere (4,5,6); however, a description will be included here for completeness.

Two ground-based laser tracking stations are required. Typically these stations incorporate a pulsed laser, transmitter, and detector optics all situated on an elevation over azimuth mount. The return signal from the vehicle must generally be enhanced through the use of conventional corner cubes, reflective tape, or paints located on the vehicle. Such devices are current technology, typified by the Precision Automated Tracking System (PATS). In the following presentation it is assumed that the tracker will provide the space position of the vehicle. In order to determine attitude, each of the ground stations is additionally equipped with a continuous wave laser; and two roof prisms are located onboard the test vehicle. For vehicles exhibiting roll rates equal to or greater than the desired attitude data rates, the prisms are simply inlet into the surface of the round at the convenient location (e.g., dummy warhead). In the development which follows, we will assume that this is the case.

Consider the 90-deg roof prism shown in Fig. 1. For this application, the two surfaces which are shown crosshatched are silvered. Two lines emanating from the center of each silvered surface and perpendicular

to their respective surface define a plane, termed here the plane of the retroreflector. A simple ray trace shows that a ray of light incident on one of the silvered surfaces and describing a path parallel to the plane of the retroreflector is generally reflected off the second surface back to the origin. If a rolling vehicle is equipped with a 90-deg roof prism and tracked so that it is continuously illuminated by a cw laser, it is clear that, a return signal of cw radiation will be returned to the laser site each time the plane of the retroreflector passes through this ground station. The signal will be in the form of a pulse, the width of which depends on the beam divergence angle, the range, the optical quality of the roof prism, and the roll rate of the vehicle. The frequency of the pulses is directly related to the roll rate.

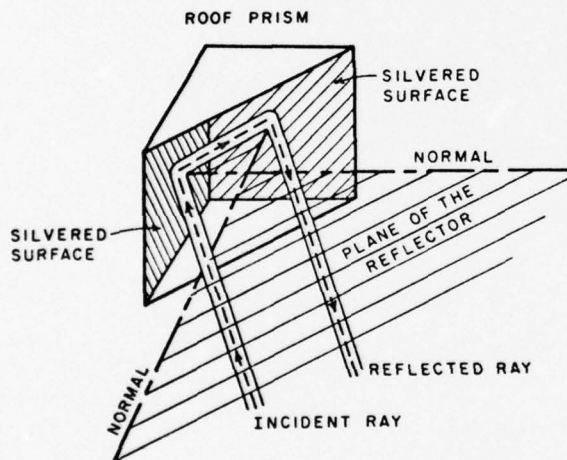


Fig. 1 Geometric optics of a roof prism.

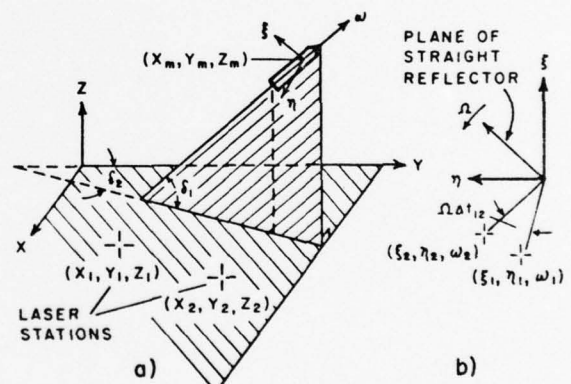


Fig. 2 Geometry for the development of the mathematical model.

A test range equipped with two tracking stations is illustrated in Fig. 2. A relationship between the return pulses received at the two stations will now be derived for a vehicle fixed in space, rotating at a constant angular rate, and equipped with a single roof prism oriented

such that the plane of the reflector contains the roll axis of the vehicle. Two coordinate systems will be utilized as shown in Fig. 2. The Earth-fixed system is defined with the origin located at the launch point; positive Z axis in the vertical upward direction; positive Y axis in the downrange direction; and the X axis in the crossrange direction to provide a right-handed system. A vehicle-centered system is defined with the origin located at the intercept of a plane bisecting the 90-deg prism and the vehicle's roll axis. The ω axis coincides with the roll axis of the vehicle, positive toward the nose; η perpendicular to ω and parallel to the XY plane, positive toward the positive X direction; and ξ perpendicular to ω and η to form a right-hand orthogonal system. The components of the position vectors of the i th ground station and the vehicle in the Earth-fixed system are (X_i, Y_i, Z_i) and (X_m, Y_m, Z_m) , respectively. Using the transformation matrix between the two systems, the components of the i th ground station in the vehicle-centered system are

$$\eta = \cos(\delta_2) (X_i - X_m) - \sin(\delta_2) (Y_i - Y_m) \quad (1a)$$

$$\omega_i = \sin(\delta_2) \cos(\delta_1) (X_i - X_m) + \cos(\delta_2) \cos(\delta_1) (Y_i - Y_m) + \sin(\delta_1) (Z_i - Z_m) \quad (1b)$$

$$\xi_i = -\sin(\delta_2) \sin(\delta_1) (X_i - X_m) - \cos(\delta_2) \sin(\delta_1) (Y_i - Y_m) + \cos(\delta_1) (Z_i - Z_m) \quad (1c)$$

where δ_1 and δ_2 represent the geometric pitch and yaw, respectively, defined as in Fig. 2a. Referring to Fig. 2b, it may be seen that the

time interval between pulses returned to the i and j ground station is

$$\Delta t_{ij} = \frac{1}{\Omega} \left[\arctan\left(\frac{\xi_i}{\eta_i}\right) - \arctan\left(\frac{\xi_j}{\eta_j}\right) \right] \quad (2)$$

Note that substitution for the ξ_i and η_i using Eq. (1) yields an equation in terms of relative position of the stations and the vehicle, and the geometric pitch and yaw. A third ground station could be used to provide a similar relationship yielding two equations and two unknowns (δ_1, δ_2) where it is assumed that the relative positions are obtained from the laser tracker, and the roll rate is inferred from pulses returned to a single station. Unfortunately, simultaneous solution of these two equations is relatively insensitive to yaw variation for reasonable third station locations and suffers the disadvantage of requiring three ground stations.

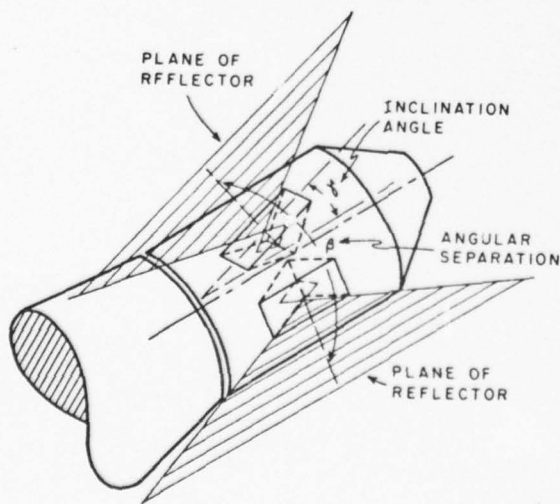


Fig. 3 Vehicle equipped with two prisms.

Consider now the addition of a second roof prism to the vehicle, as indicated in Fig. 3, with an angular separation of β relative to the first reflector and a skew of γ . The geometry of the prisms is shown in more detail in Fig. 4. Assume that the t_1 the plane of the straight retroreflector passes through the i th ground station. As the vehicle continues to roll the plane of the second reflector eventually passes through the same ground station at t_2 . It is apparent from the figure that the vehicle must roll a distance $\beta + \theta$ so that the time interval between the two pulse receptions is $\Delta t_{ii} = (1/\Omega) (\beta + \theta_i)$.

Figure 4 also indicates that $\tan \sigma_i = R/B_i$ and $\tan \gamma = C_i/B_i$. Therefore, $\theta_i = \arcsin (\tan \gamma) / (\tan \sigma_i)$. By noting that

$$\tan \sigma_i = (\eta_i^2 + \xi_i^2)^{1/2} / \omega_i$$

one obtains the result

$$\Delta t_{ii} = \frac{1}{\Omega} \left[\beta + \arcsin \left[\frac{\omega_i \tan \gamma}{(\eta_i^2 + \xi_i^2)^{1/2}} \right] \right] \quad (3)$$

which can be expressed in terms of δ_1 and δ_2 using Eqs. (1).

This equation is very similar to the Yawsonde formulation because both approaches involve the relationship of planes, fixed with respect to the vehicle, relative to a remote point. The Yawsonde approach differs from the concept presented here in as much as it involves on-board detectors and telemetry. Equations (2) and (3) may be solved numerically for δ_1 and δ_2 assuming once again that Ω , $X_i - X_m$, $Y_i - Y_m$, $Z_i - Z_m$, Δt_{12} are known.

Unfortunately, a closed form solution of these two equations could not be found. The problem was reformulated to achieve a closed form solution as explained in the next section.

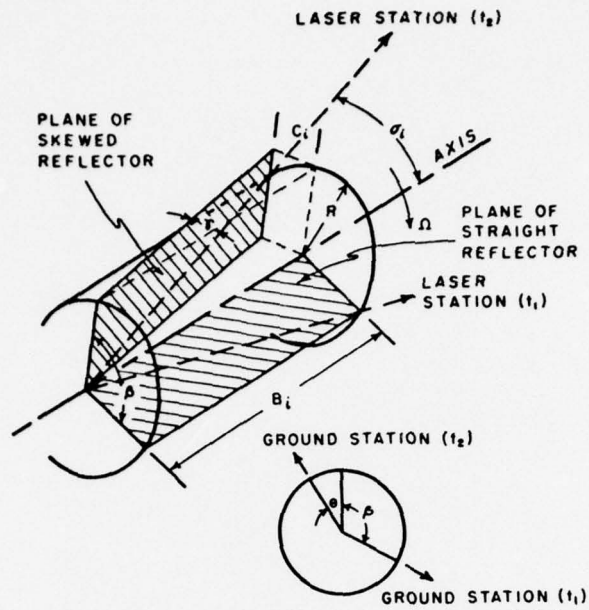


Fig. 4 Retroreflection planes referred to a laser ground station at two different times.

Reformulation of the mathematical description

An alternate expression for the time interval for retroreflection from one retroreflector to the two ground stations can be developed.

Two ground stations are located relative to an earth fixed coordinate system by vectors \vec{r}_i . The position of a missile rotating with a constant angular velocity ω about a roll axis \hat{e}_ω is given by vector \vec{r}_m . From these vectors, we form unit vectors \hat{e}_{ri} describing the direction from the missile to the ground stations:

$$\hat{e}_{ri} \equiv \frac{\vec{r}_i - \vec{r}_m}{|\vec{r}_i - \vec{r}_m|} \quad i = 1, 2 \quad (4)$$

In this equation the carat ($\hat{}$) denotes a vector of unit magnitude and $|\vec{r}_i - \vec{r}_m|$ denotes the magnitude of the vector difference between \vec{r}_i and \vec{r}_m .

A retroreflection plane may be defined as a plane within which a light signal transmitted to the missile is returned to the place of emission. A unit normal which defines the plane of retroreflection currently passing through ground station (i) may be defined as $\hat{\epsilon}_{\tau i}$, where $\hat{\epsilon}_{\tau i}$ is physically interpretable as a unit vector lying along the intersection of two mirror surfaces placed 90° apart.

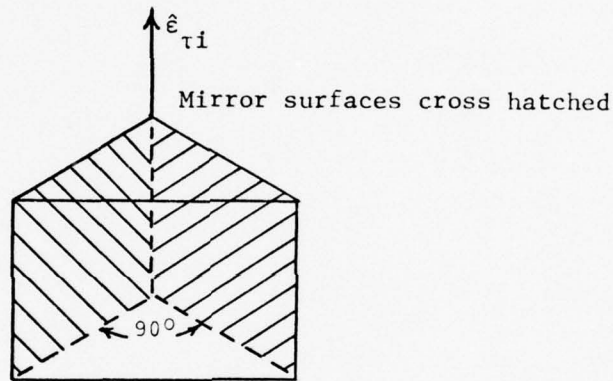


Fig. 5 Definition of $\hat{\epsilon}_{\tau i}$

We define a skewness angle α as the angle between $\hat{\epsilon}_{\tau i}$ and a normal to the roll axis ($\hat{\epsilon}_{ni}$) when $\hat{\epsilon}_{\tau i}$, $\hat{\epsilon}_\omega$, and $\hat{\epsilon}_{ni}$ are all coplanar and which allows for mounting the mirrors skewed relative to the roll axis.

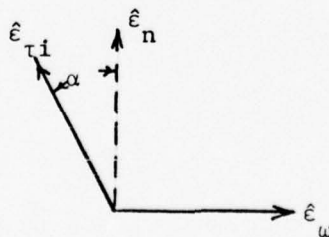


Fig 6. Definition of α

The angle between $\hat{\epsilon}_{r1}$ and $\hat{\epsilon}_\omega$ is the tracking aspect angle σ_1 .

$$\hat{\epsilon}_{r1} \times \hat{\epsilon}_\omega = \sin \sigma_1 \hat{\epsilon}_{n1} \quad (5)$$

The angle between the two ground stations as taken from the missile is

defined as μ .

$$\hat{\epsilon}_{r1} \cdot \hat{\epsilon}_{r2} = \cos \mu_{12} \quad (6)$$

The retroreflection plane is assumed to pass through ground station 1 at time t_1 . It passes through station 2 at time $t_2 = t_1 + t_{12}$. Between these two times, the retroreflection plane has rotated through an angle $\omega \Delta t_{12}$ where ω is the constant roll rate of the missile.

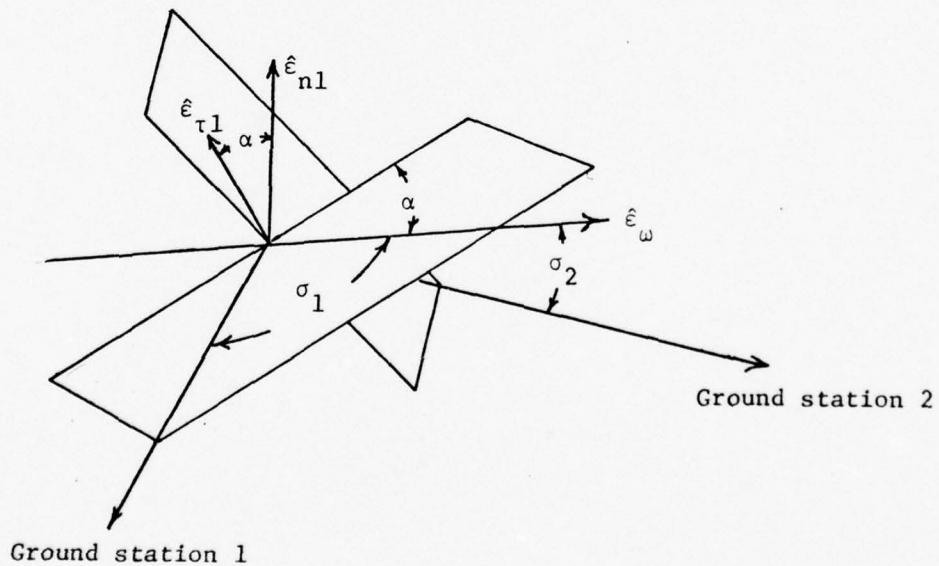


Fig. 7 Geometric relationship

In order to determine the angle of rotation, we must define a unit vector $\hat{\epsilon}_{Ai}$ that lies in the plane of retroreflection with station i, but is perpendicular to the roll axis $\hat{\epsilon}_\omega$. It is necessary that $\hat{\epsilon}_{Ai}$ be a linear combination of vector $\hat{\epsilon}_{ri}$ and $\hat{\epsilon}_\omega$. This $\hat{\epsilon}_{Ai}$ is found using the form

$$\hat{\epsilon}_{Ai} = \frac{\hat{\epsilon}_\omega \times \hat{\epsilon}_{ri}}{|\hat{\epsilon}_\omega \times \hat{\epsilon}_{ri}|} \quad (7)$$

Now that all angles and vectors needed to find the aspect angles σ_i have been found, we may define $\hat{\epsilon}_{\tau i}$.

$$\hat{\epsilon}_{\tau i} = -\sin \alpha \hat{\epsilon}_\omega + \frac{\cos \alpha}{\sin \sigma_i} \hat{\epsilon}_{ri} \times \hat{\epsilon}_\omega \quad (8)$$

Using equations (7) and (8), and the fact that

$$|\hat{\epsilon}_\omega \times \hat{\epsilon}_{\tau i}| = |\sin(90 + \alpha)| = \cos \alpha \quad (9)$$

we arrive at

$$\hat{\epsilon}_{Ai} = \frac{1}{\sin \sigma_i} [\hat{\epsilon}_{ri} - \cos \sigma_i \hat{\epsilon}_\omega] \quad (10)$$

As previously mentioned, the angle of roll that the missile has experience between retroreflecting station 1 and 2 is $\omega \Delta t_{12}$. The vectors $\hat{\epsilon}_{A1}$ and $\hat{\epsilon}_{A2}$ must also have undergone this same angular change so that

$$\hat{\epsilon}_{A1} \cdot \hat{\epsilon}_{A2} = \cos(\omega \Delta t_{12})$$

Using equation (10) twice, we see that

$$\begin{aligned} \cos(\omega \Delta t_{12}) &= \hat{\epsilon}_{A1} \cdot \hat{\epsilon}_{A2} \\ &= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\hat{\epsilon}_{r1} - \cos \sigma_1 \hat{\epsilon}_\omega] \cdot [\hat{\epsilon}_{r2} - \cos \sigma_2 \hat{\epsilon}_\omega] \\ &= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\hat{\epsilon}_{r1} \cdot \hat{\epsilon}_{r2} - \cos \sigma_1 \hat{\epsilon}_\omega \cdot \hat{\epsilon}_{r2} \\ &\quad - \cos \sigma_2 \hat{\epsilon}_{r1} \cdot \hat{\epsilon}_\omega + \cos \sigma_1 \cos \sigma_2 \hat{\epsilon}_\omega \cdot \hat{\epsilon}_\omega] \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\cos \mu_{12} - \cos \sigma_1 \cos \sigma_2 \\
&\quad - \cos \sigma_2 \cos \sigma_1 + \cos \sigma_1 \cos \sigma_2] \\
&= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\cos \mu_{12} - \cos \sigma_1 \cos \sigma_2] \quad (12)
\end{aligned}$$

With equation (3) written in the form

$$t_{if} = \frac{1}{\Omega} [\beta + \arcsin \left(\frac{\tan \gamma}{\tan \sigma} \right)] \quad (13)$$

Equations (12) and (13) can be solved simultaneously for σ_1 and σ_2 in closed form. The two aspect angles can be used to determine the orientation of the longitudinal axis of the vehicle thereby providing vehicle attitude. A solution algorithm has been implemented in program ASP11 which is documented in Appendix I. Input data for ASP11 includes ground station locations, missile position, roll rate and measured time intervals.

In order to obtain an accurate missile position in range coordinates from the data tape generated by LAMPAMS a second program was developed.

Vehicle position determination

Data tapes generated by the LAMPAMS are in binary form, 9 track, 800 bpi. The data arrangement on the tape is shown in Fig. 8. In order to generate accurate position information the azimuth, elevation and range recorded on the tape in binary form are:

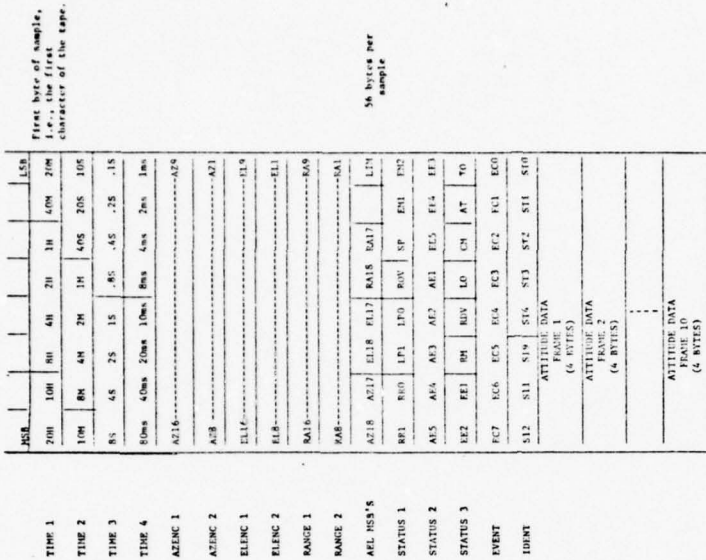
1. Transformed to based 10.
2. Corrected for bias.
3. Smoothed with a running 10 point least squares polynomial.
4. Transformed to range coordinates.

KEY - PAYS DATA (First sixteen bytes in each sample)

Time: (BCD) H, hours
 M, minutes
 S, seconds
 ms, milliseconds
 Encoders: (natural binary, LSB = 360°/2¹⁶) A2, azimuth angle
 E1, elevation angle
 Range: (Binary, LSB = 1.0 foot)
 Status bits: LIN, limit
 RA, range
 RE, recording rate
 R0 = 10 samples per second
 R1 = 20 samples per second
 R2 = 50 samples per second
 R3 = 100 samples per second
 LP, transmitter power (unused)
 RP, range overflow
 SP, signal present
 EP, event marker switches
 AP, azimuth tracking error
 RP, remote data valid flag (unused)
 LP, laser on
 CP, computer track mode
 AT, autotrack mode
 TQ, track quality switch
 EC, event count
 SI, system ID
 ST, station ID

The "EPOCH OFFSET" is encoded as follows:
 0 0 0 0 = -10 milliseconds
 0 0 0 1 = -9
 0 0 1 0 = -8
 0 0 1 1 = -7
 0 1 0 0 = -6
 1 0 0 0 = -5
 1 0 0 1 = -4
 1 0 1 0 = -3
 1 0 1 1 = -2
 1 1 0 0 = -1

Time of Attitude Pulse = EPOCH TIME (time from "PAYS" data in same sample)
 +EPOCH OFFSET + PULSE POSITION



There are 32 samples per record or 1792 bytes per record.

NOTE: ATTITUDE DATA (Last 40 bytes in each sample - 4 bytes per frame, 10 frames per sample)
 NOTE: Attitude data frames of all zeros should be ignored.
 One frame of valid attitude data is as shown below:

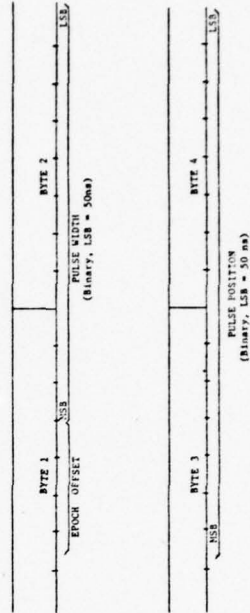


Fig. 8 LAMPAMS magnetic tape format

A program (LTPOS) has been written to accomplish these tasks and a listing is included as Appendix II. This program was initially developed to run on the central computer at the U.S. Army Missile Command (CDC 6600) in anticipation that this is where the data analysis would be done. Subsequently, the program was adapted to run on the Hewlett Packard 1000 System, which was installed on the test range at Redstone Arsenal.

An additional program to determine tracker bias based on measurements of surveyed points equipped with retroreflectors was acquired from Yuma proving ground and adapted to run on the central computer at Redstone. This program has been partially converted to run on the Hewlett Packard system; however, this has not been completed. Because of the length of the program overlay techniques are required and considerable additional software development will be required to accomplish this task.

Application to non-rolling vehicles

The approach evolved for testing non-rolling or slowly rolling vehicles involves spinning a portion of the dummy warhead up to relatively high rates prior to launch and simply allowing it to spin-down during flight. In order to demonstrate the approach, twelve 2.75 inch rocket fuses were reworked as shown in Figure 9 and provided to MICOM for use in flight tests. Design of a pre-spin device was coordinated with MICOM personnel and a unit was fabricated at MICOM.

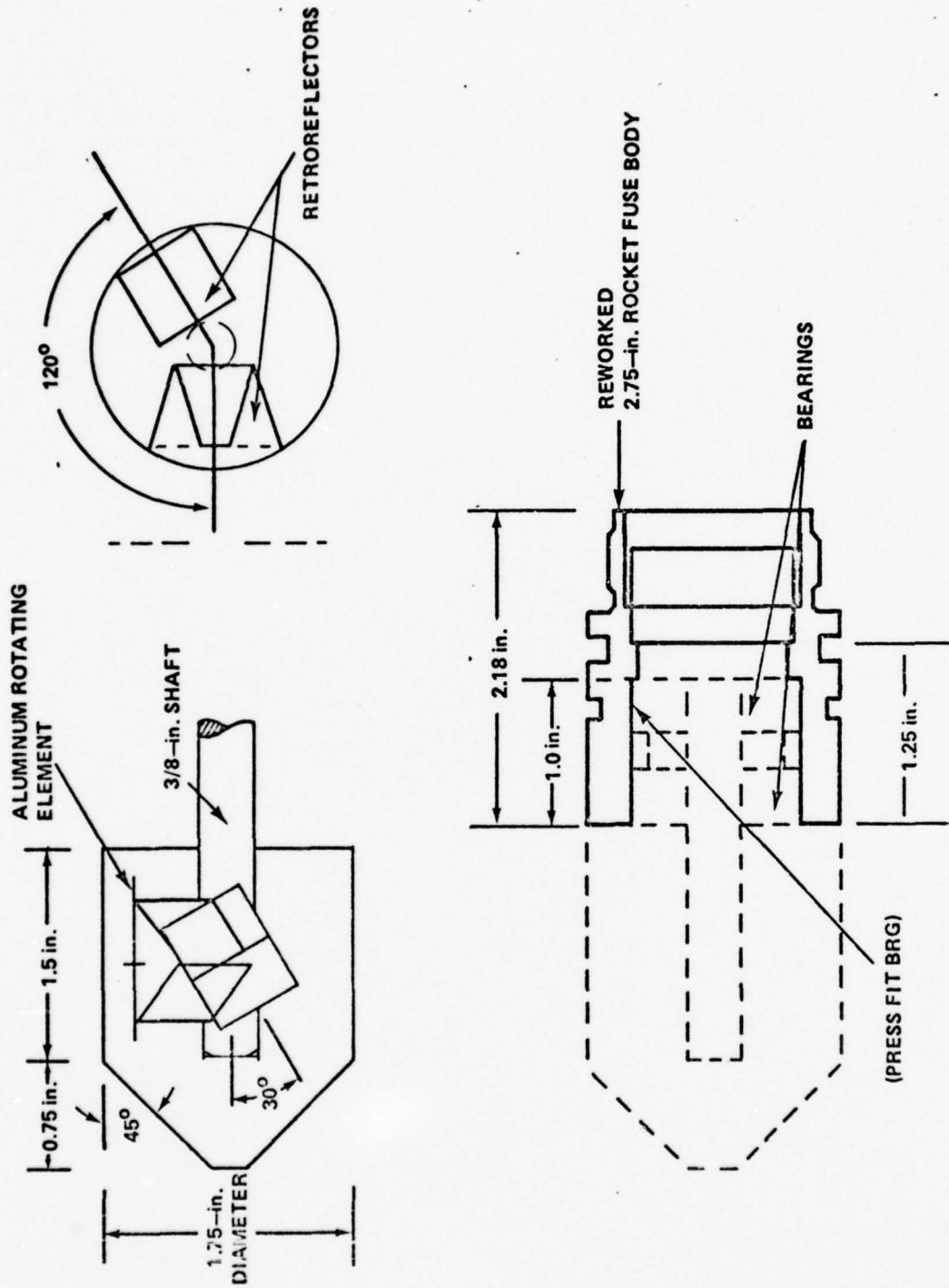


Fig. 9 Reworked 2.75 inch fuse assembly

Status of the LAMPAMS

The LAMPAMS system was installed on Range 1 of Redstone Arsenal in the spring of 1978. Since that time the tracker portion of the system has become the primary instrument for vehicle position determination. The program LTPOS originally provided to MICOM has served as the basis for data reduction. It has been extensively revised by MICOM personnel to include graphical output.

Unfortunately, the CW laser and signal conditioning hardware associated with the attitude subsystem have been subject to repeated failures and as of September 1979 no attitude data has been generated.

Since both the pulsed laser used for tracking and the CW laser on the attitude system operate at the same frequency the data must be separated on the basis of pulse width in software. In addition, the potential for missing a return pulse from the attitude subsystem is higher than that associated with the tracking system. The software package required to determine appropriate time intervals for use in ASP11 has not been completed because of these uncertainties and the lack of data to resolve the areas in question.

Conclusions and Recommendations

Determination of vehicle attitude using the concept described requires two laser ground stations. Presently, only one station is available. The concept can however, be verified using a single ground station. Every effort should be made to get the attitude subsystem of LAMPAMS operational so that the required software can be completed. The efficacy of the concept could then be verified. This should be completed prior to acquisition of a second laser ground station.

References

1. "A Laser System for Determination of Rocket Attitude and Roll Rate," with Mark J. Russell, John E. Nydahl, and William Lindberg. Final Report, grant DAHCO4074-0063/Department of Mechanical Engineering Report UWME-DR-4061051, July 1974.
2. "Measurement of Missile Position and Attitude by Lasers," with R.G. Conard and J.E. Nydahl. MICOM Report TR-RD-76-7, July 1975.
3. "Techniques for Remote Sensing of Missile Position and Attitude," MICOM Report TR-RD-76-11, October 1975.
4. "A Laser System to Measure Rocket Attitude in Flight," with John E. Nydahl et al., Journal of Spacecraft and Rockets, Vol. 15, No. 4, July-August 1978
5. "Preliminary Test Results on a Laser System for Measurement of Rocket Attitude During Flight Test," with J.E. Nydahl et al., paper 78-17, Space Science Meeting, AIAA, Huntsville, AL January 1978
6. "Attitude Determination of Missiles and Rockets Using a Laser System," with J. E. Nydahl, R. G. Conard and C.R. Cooke, Proceedings of the 9th Annual Electro Optics Conference, Anaheim, CA, October 1977

Appendix I

ASP11

Program to determine attitude

of vehicle

(Hewlett Packard System 1000 version)

THE MAIN PROGRAM

The main program exists to call the proper input subroutine for reduction of input values to pitch and yaw angles. It also sets the input/output device numbers, the maximum allowable error, the constant π , and the number of degrees per radian.

Currently the main program calls only one other program segment, SUBROUTINE INPUT.

SUBROUTINE INPUT

The subroutine provides an interactive method of operating the rest of the program. Sixteen commands give versatility and ease to changing any or all of the parameters involved in a pitch and yaw determination. The subroutine prompts the user for required input of the various parameters. As an added check, the values fed in after a prompt are displayed to the user. If any doubt remains as to what variables the program is using, a command is available to display the current values of all variables.

The first time through, the user is prompted to input all the necessary variables for a computation of pitch and yaw. After all variables are input, a prompt for a command is printed. At this point fifteen possible commands are available:

RUN - will determine the pitch and yaw using the current variables

A RETURN, with nothing entered - does the same as RUN

DISPLAY - displays a list of all current variables

? - provides a list of commands that are available and what they do

GS1 - will change the coordinates of Ground Station One

GS2 - changes Ground Station Two coordinates

MISSILE - changes missile coordinates

DT11 - changes the time difference for the two mirrors to retroreflect through ground station one

DT12 - changes the time difference for one mirror to retroreflect through ground station one, and then ground station two

BETA - changes the radial separation between the two mirrors on the missile

SK1 - changes the skewness of mirror set one

SK2 - changes the skewness of mirror set two

OMEGA - changes the roll rate of the missile

RESTART - starts an entirely new case where all variables must be reentered

STOP - halts the execution of the program when you are finished

Any command may be input after the prompt "COMMAND", and in any order.

Omega, the roll rate, may be input in

RAD - radians per second

RPS - revolutions per second

RPM - revolutions per minute

DPS - degrees per second

The program converts the input value into radians per second.

When INPUT encounters a RUN command, or a blank command, subroutine SOLVE is called to determine the pitch and yaw angles.

SUBROUTINE SOLVE

SOLVE is the organizer for the solution of pitch and yaw angles. It calls TUP to calculate the geometric relationships between the ground stations and the missile. Next, SIG11 is called to determine the first aspect angle. SIG12 is subsequently called to find what the other aspect angle is. ESUBW is then called to find the correct roll axis.

SOLVE always returns to INPUT.

SUBROUTINE SETUP

SETUP determines \vec{R}_1 , \vec{R}_2 , \hat{e}_{R1} , and \hat{e}_{R2} . Unit vector \hat{e}_{R1} is parallel to \vec{R}_1 . The magnitudes R1 and R2 are determined, as is COSMU which is the cosine of the angle between \vec{R}_1 and \vec{R}_2 .

SUBROUTINE SIG11

Using the time necessary for both sets of missile mirrors to retro-reflect through ground station one (DT11), and the yawsonde equation, the first aspect angle SIGMA1 is determined.

SUBROUTINE SIG12

The time required for one mirror to retroreflect through ground station one and then ground station two (DT12), when used in

$$\cos(\Omega \Delta t_{12}) = \frac{1}{\sin \sigma_1 \sin \sigma_2} [\cos \mu - \cos \sigma_1 \cos \sigma_2] \quad (1)$$

where: Ω = OMEGA, the missile roll rate in radians per second

$$\Delta t_{12} = DT12$$

$$\cos \mu = \text{COSMU}$$

$$\cos \sigma_1 = \cos(\text{SIGMA1})$$

$$\sin \sigma_1 = \sin(\text{SIGMA1})$$

determines two possible SIGMA2 aspect angles. SIG11 provides SIGMA1 (σ_1), SETUP provides COSMU ($\cos \mu$), so that σ_2 may be determined explicitly.

Two roots become possible for σ_2 . These are SIG2A and SIG2B. To further complicate matters, we only accept angles that are $0 \leq \sigma_2 \leq \pi$. When the arctangent function is used, a negative argument may represent either $-\sigma_2$ or $\pi/2 - \sigma_2$.

These difficulties were circumvented by:

- 1) replacing a $\sigma_2 < 0$ by $\pi/2 - \sigma_2$, hence making σ_2 positive
- 2) testing the resultant σ_{2A} and σ_{2B} to see if equation (1) is satisfied.

When the subroutine returns to SOLVE, ISOL contains the integer number of solutions which satisfy (1). If ISOL = 0, no solution is possible, and SOL1 = SOL2 = 0, an arbitrary default value. If ISOL = 1, the correct solution is in SOL1. If ISOL = 2 the two solutions are in SOL1 and SOL2.

SUBROUTINE ESUBW

When two aspect angles are given, the roll axis, $\hat{\epsilon}_w$ (ESUBW) may be determined by the solution of:

$$\begin{aligned}\hat{\epsilon}_{R1} \cdot \hat{\epsilon}_w &= \cos(\sigma_1) \\ \hat{\epsilon}_{R2} \cdot \hat{\epsilon}_w &= \cos(\sigma_2) \\ \hat{\epsilon}_w \cdot \hat{\epsilon}_w &= 1\end{aligned}\tag{2}$$

Equations (2) are nonlinear, and therefore not easily solved. The method used in this program determines the \hat{i} component of $\hat{\epsilon}_w$, or ϵ_{wx} .

This reduces (2) to:

$$\begin{pmatrix} \epsilon_{R1S} & \epsilon_{R1Y} & \epsilon_{R1Z} \\ \epsilon_{R2X} & \epsilon_{R2Y} & \epsilon_{R2Z} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \epsilon_{wx} \\ \epsilon_{xy} \\ \epsilon_{wz} \end{pmatrix} = \begin{pmatrix} \cos(\sigma_1) \\ \cos(\sigma_2) \\ \epsilon_{wx} \end{pmatrix}\tag{3}$$

Equation (3) is linear, and easily solved with a SIMultaneous eQuation solver, SIMQ. With the roll axis known, pitch and yaw are easily determined by:

$$\begin{aligned}\text{YAW} &= \text{ARCTAN}(\epsilon_{wx}/\epsilon_{xy}) \\ \text{PITCH} &= \text{ARCTAN}[\epsilon_{wx}/(\epsilon_{wx}^2 + \epsilon_{wy}^2)^{1/2}]\end{aligned}$$

The three cases which are impossible to solve using this formulation

are

$$\text{Case I } \epsilon_{R1Y} = \epsilon_{R2Y} = 0$$

$$\text{Case II } \epsilon_{R1Z} = \epsilon_{R2Z} = 0$$

$$\text{Case III } \hat{\epsilon}_{R1} = K \hat{\epsilon}_{R2} \text{ where } K \text{ is a real number}$$

Case III physically corresponds to the missile being located between the two ground stations along a vector connecting the ground stations.

Cases I and II reduce equation (3) to an overdeterminant set which may be solved for ϵ_{WX} and ϵ_{WZ} , and ϵ_{WX} and ϵ_{WY} respectively, as a system of two equations in two unknowns. The remaining component may be found by

$$\epsilon_{WX}^2 + \epsilon_{WY}^2 + \epsilon_{WZ}^2 = 1 \quad (5)$$

The main difficulty would be the determination of ϵ_{WX} . This is done in subroutine ESUBWX.

It is important to note that, given two aspect angles, there are two possible roll axes. To determine the correct roll axis, subroutine TIME is called. When TIME is given a roll axis, it determines the two time differences DELT11 and DELT12 from previous work. If the two computed time differences are within the allowable error limit, then the roll axis is valid, otherwise it is not.

SUBROUTINE ESUBWX

ESUBWX explicitly determines ϵ_{WX} from equation (2). There are generally two solutions ϵ_{WX} . These reside in ROOT1 and ROOT2. Occasionally the two roots are redundant, then ISOL, the number of solutions is one. If $\hat{\epsilon}_{R1}$ and $\hat{\epsilon}_{R2}$ are linearly dependent, then ISOL = 0 as no solution is possible.

In the solution of ϵ_{WX} , a quadratic equation of the form

$$(1 + B^2 + D^2)\epsilon_{WX}^2 + 2(AB + CD)\epsilon_{XW} + (A^2 + C^2 - 1) = 0 \quad (6)$$

is recurrent. The values of A, B, C and D vary depending on the vectors $\hat{\epsilon}_{R1}$, $\hat{\epsilon}_{R2}$, and the values of $\cos(\sigma_1)$ and $\cos(\sigma_2)$. Subroutine QUADRA solves quadratic equation (6) given A, B, C, and D.

SUBROUTINE QUADRA

QUADRA solves equation (6) given A, B, C and D. If the radical resulting is less than zero, ISOL = 0, and no solution is possible because the two roots would be complex. This corresponds to two non-intersecting aspect angle cones. If ISOL = 1, one solution is the same as the other. This corresponds to two tangent cones. Typically ISOL = 2, and two solutions for ϵ_{wx} , ROOT1 and ROOT2, exist.

SUBROUTINE SIMQ

Given an N by N matrix [A], and a vector [B] (N x 1), SIMQ solves the matrix equation

$$[A] [X] = [B] \quad (7)$$

for [X], a vector that is N x 1. During the course of computation, [A] and [B] are destroyed.

A possible alteration to speed computation would be to use Cramer's rule for the solution of [X] instead of the current Gauss elimination procedure.

SUBROUTINE TIME

TIME calculates DELT11 and DELT12 when given a roll axis and all geometric constants.

Pitch and yaw are determined from equations (4). An alternate coordinate system is defined as

A possible alteration to speed computation would be to use Cramer's

rule for the solution of [X] instead of the current Gauss elimination procedure.

SUBROUTINE TIME

TIME calculates DELT11 and DELT12 when given a roll axis and all geometric constants.

Pitch and yaw are determined from equations (4). An alternate coordinate system is defined as

$$\begin{aligned}\eta_1 &= \cos(\text{YAW}) \text{RX}_i - \sin(\text{YAW})\text{RY}_i & (8) \\ \omega_1 &= \sin(\text{YAW}) \cos(\text{PITCH})\text{RX}_i + \cos(\text{PITCH})\text{RY}_i + \sin(\text{PITCH})\text{RZ}_i \\ \epsilon_1 &= \sin(\text{YAW}) \sin(\text{PITCH})\text{RX}_i - \cos(\text{YAW}) \sin(\text{PITCH})\text{RY}_i + \\ &\quad \cos(\text{PITCH})\text{RZ}_i\end{aligned}$$

times are found by

$$\begin{aligned}\text{DELT11} &= \frac{1}{\text{OMEGA}} \{ \text{BETA} + \text{ARCSIN} [\omega_1 \text{TAN}(\text{SKEW1}) / (\epsilon_1^2 + \eta_1^2)^{1/2}] \} & (9) \\ \text{DELT12} &= \frac{1}{\text{OMEGA}} \text{ARCTAN} (\epsilon_1/\eta_1) - \text{ARCTAN} (\epsilon_2/\eta_2)\end{aligned}$$

where OMEGA is the roll rate of the missile in radians per second

BETA is the angular separation of the mirrors in radians

SKEW1 is the skewness of mirror one

If both of the DELT times are within the allowable error tolerance of the initial times given, then the pitch and yaw values are output.

```

0001  FTN4.L
0002  PROGRAM ASPII
0003  DOUBLE PRECISION X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
0004  1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0005  2 YR1, ZR1, XR2, YR2, ZR2, ANSWER, P1, RADIAN,
0006  3 ERROR, EWX, EWY, EWZ
0007  INTEGER OUTPUT
0008  COMMON X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
0009  1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0010  2 YR1, ZR1, XR2, YR2, ZR2, ANSWER(4,6), P1, RADIAN,
0011  3 ERROR, EWX, EWY, EWZ, NUMSOL, NREAD, OUTPUT
0012  DIMENSION IPRAM(5)
0013  CALL RMPAR( IPRAM)
0014  NUMSOL=0
0015  NREAD= IPRAM(1)
0016  OUTPUT= IPRAM(2)
0017  PI=4.D0*DATAN(1.D0)
0018  RADIAN= 180.D0/PI
0019  ERROR= 1.D-5
0020  CALL INPUT
0021  STOP
0022  END

```

```

0023      SUBROUTINE INPUT
0024      DOUBLE PRECISION DT11,DT12
0025      DOUBLE PRECISION BETAIN,SKEW11,SKEW21
0026      DOUBLE PRECISION OMEGA,OMEGIN,PITCH,YAW
0027      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XI,YI,ZI,SKEW1,
0028      1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0029      2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
0030      3 ERROR,EWX,EWY,EWZ
0031      INTEGER OUTPUT
0032      COMMON X1,Y1,Z1,X2,Y2,Z2,XI,YI,ZI,SKEW1,
0033      1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0034      2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
0035      3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0036      INTEGER ICONT(32)
0037      DATA ICONT/2HGS,2H1,2HGS,2H2,2HMI,2HSS,2HDT,2H11,2HDT,
0038      1 2H12,2HBE,2HTA,2HSK,2H1,2HRU,2HN,2H,2H,2HST,2HOP,
0039      2 2HDI,2HSP,2HSK,2H2,2HOM,2HEG,2HRE,2HST,2H?,2H,2HSO,
0040      3 2HLU/
0041      DATA IRAD1,IRPS2,IRPM2,IDPS1/2HRA,2HS,2HM,2HDP/
0042      5 KFLAG=0
0043      10 WRITE(OUTPUT,100)
0044      100 FORMAT(" INPUT GROUND STATION 1 COORDS: X,Y,Z")
0045      READ(NREAD,*)X1,Y1,Z1
0046      WRITE(OUTPUT,101)X1,Y1,Z1
0047      101 FORMAT(1P3(2X,D20.15))
0048      IF(KFLAG.NE.0) GOTO 300
0049      20 WRITE(OUTPUT,102)
0050      102 FORMAT(" INPUT GROUND STATION 2 COORDS: X,Y,Z")
0051      READ(NREAD,*)X2,Y2,Z2
0052      WRITE(OUTPUT,101)X2,Y2,Z2
0053      IF(KFLAG.NE.0) GO TO 300
0054      30 WRITE(OUTPUT,103)
0055      103 FORMAT(" INPUT MISSILE COORDS: X,Y,Z")
0056      READ(NREAD,*)XI,YI,ZI
0057      WRITE(OUTPUT,101)XI,YI,ZI
0058      IF(KFLAG.NE.0) GO TO 300
0059      40 WRITE(OUTPUT,104)
0060      104 FORMAT(" INPUT DELTA T 11, IN SECONDS")
0061      READ(NREAD,*)DT11
0062      WRITE(OUTPUT,105)DT11
0063      IF(KFLAG.NE.0) GO TO 300
0064      105 FORMAT(2X,1PD23.15)
0065      50 WRITE(OUTPUT,106)
0066      106 FORMAT(" INPUT DELTA T 12, IN SECONDS")
0067      READ(NREAD,*)DT12
0068      WRITE(OUTPUT,105)DT12
0069      IF(KFLAG.NE.0) GO TO 300
0070      60 WRITE(OUTPUT,107)
0071      107 FORMAT(" INPUT THE RADIAL SEPARATION OF MIRRORS IN "
0072      1 " DEGREES")
0073      READ(NREAD,*)BETAIN
0074      BETA=BETAIN/RADIAN
0075      WRITE(OUTPUT,105)BETAIN
0076      IF(KFLAG.NE.0) GO TO 300
0077      70 WRITE(OUTPUT,108)
0078      108 FORMAT(" INPUT THE SKEW ANGLE OF MIRROR 1, IN DEGREES")
0079      READ(NREAD,*)SKEW11
0080      SKEW1=SKEW11/RADIAN
0081      WRITE(OUTPUT,105)SKEW11
0082      110 FORMAT("/ WHEN YOU WISH TO CHANGE SOMETHING, OR RUN"/
0083      1 " THE PROG., THEN FOLLOW THE LIST BELOW"/
0084      2" GS1=GROUND STATION ONE COORDINATES"/
0085      2" GS2=GROUND STATION TWO COORDS. "/" MISSILE=MISSILE COORDS."/
0086      4" SK1=SKEW OF MIRROR 1"/" SK2=SKEW OF MIRROR 2"/" RUN=COMPUTE"
0087      5/" NO ENTRY, JUST A RETURN, WILL EXEC THE PROG"/" STOP="
0088      6"HALT OF EXECUTION"/" BETA=RADIAL SEPARATION OF THE MIRRORS"/
0089      7" DISPLAY=CURRENT VALUES OF ALL VARIABLES"/)
0090      IF(KFLAG.NE.0) GO TO 300
0091      71 WRITE(OUTPUT,109)

```

```

0092 109  FORMAT(" INPUT THE SKEW ANGLE OF MIRROR 2, IN DEGREES")
0093      READ(NREAD,*)SKEW21
0094      SKEW2=SKEW21/RADIAN
0095      WRITE(OUTPUT,105)SKEW21
0096      IF(KFLAG.NE.0)GO TO 300
0097      72  WRITE(OUTPUT,115)
0098      READ(NREAD,*)OMEGIN
0099      READ(NREAD,112) IUNIT1,IUNIT2
0100      114  FORMAT(2X,D23.15,2A2)
0101      115  FORMAT(" INPUT THE ROLL RATE, SKIP A LINE, "/
0102      1  " AND THEN RAD FOR RADIANS PER SECOND"/
0103      2  " RPS FOR REVOLUTIONS PER SECOND"/
0104      3  " RPM FOR REVOLUTIONS PER MINUTE"/
0105      4  " DPS FOR DEGREES PER SECOND")
0106      OMEGA=0.D0
0107      IF(IUNIT1.EQ.IRAD1)OMEGA=OMEGIN
0108      IF(IUNIT2.EQ.IRPS2)OMEGA=2.D0*PI*OMEGIN
0109      IF(IUNIT2.EQ.IRPM2)OMEGA=120.D0*PI*OMEGIN
0110      IF(IUNIT1.EQ.IDPS1)OMEGA=OMEGIN/RADIAN
0111      IF(OMEGA.EQ.0.D0) GO TO 91
0112      WRITE(OUTPUT,116)OMEGIN,IUNIT1,IUNIT2,OMEGA
0113      116  FORMAT(" YOUR "1PD16.10,2X,2A2"CONVERTS TO"F16.10" RADIANS PER"
0114      1  " SECOND")
0115      KFLAG=1
0116      IF(KFLAG.EQ.0) WRITE(OUTPUT,110)
0117      111  FORMAT(" COMIAND")
0118      300  WRITE(OUTPUT,111)
0119      READ(NREAD,112) ICOM1,ICOM2
0120      112  FORMAT(2A2)
0121      IFLAG=0
0122      DO 301 I=1,16
0123      IF(ICONT(2*I-1).EQ.ICOM1.AND.ICONT(2*I).EQ.ICOM2)IFLAG=1
0124      301  CONTINUE
0125      IF(IFLAG.EQ.0) WRITE(OUTPUT,113)
0126      IF(IFLAG.EQ.0) GO TO 300
0127      113  FORMAT(" EH?")
0128      GO TO(10,20,30,40,50,60,70,80,80,90,200,71,72,5,73,74),IFLAG
0129      91  WRITE(OUTPUT,113)
0130      GO TO 300
0131      73  WRITE(OUTPUT,110)
0132      GO TO 300
0133      74  WRITE(OUTPUT,117)
0134      DO 119 INT=1,NUMSOL
0135      119  WRITE(OUTPUT,118) INT,ANSWER( INT, 1),ANSWER( INT, 2),ANSWER( INT, 3)
0136      1 ,ANSWER( INT, 4),ANSWER( INT, 5),ANSWER( INT, 6)
0137      117  FORMAT("/" SOLUTION PITCH YAW TIME11 TIME12"
0138      1  " ERR11 ERR12"/)
0139      118  FORMAT(2X,15,2F10.5,2X,2F10.6,2D10.4)
0140      GO TO 300
0141      90  RETURN
0142      80  CALL SOLVE(PITCH,YAW,OMEGA,DT11,DT12)
0143      GO TO 300
0144      200  WRITE(OUTPUT,201)X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,BETA1,SKEW11,
0145      1  SKEW21,DT11,DT12,OMEGA
0146      201  FORMAT(" STATION ONE: "2(D16.9",")D16.9/" STATION TWO: "
0147      12(D16.9",")D16.9/" MISSILE: "2(D16.9",")D16.9/" BETA= "D16.9" DEG"
0148      2  ,"REES"/" SKEW1= "D16.9" DEGREES"/" SKEW2= "D16.9" DEGREES",
0149      3  /" DELTA T 11= "D16.9" SECONDS"/" DELTA T 12= ",
0150      4  D16.9" SECONDS"/" OMEGA= "D16.9" RADIANS PER SECOND"/)
0151      GO TO 300
0152      END

```

```

0153 C SIMQ SOLVES THE MATRIX EQUATION A TIMES X EQUALS B
0154 C FOR THE VECTOR X
0155 C GIVEN ARE THE N BY N MATRIX A, AND THE VECTOR B.
0156 C WHEN KS=0, A VALID SOLUTION IS GIVEN
0157 C WHEN KS=1, THE MATRIX A IS SINGULAR
0158 SUBROUTINE SIMQ(A,B,KS)
0159 DOUBLE PRECISION A(3,3),B(3),DETA,DETB,DETC,DET
0160 KS=0
0161 DET=A(1,1)*(A(2,2)*A(3,3)-A(2,3)*A(3,2))-
0162 1 A(1,2)*(A(2,1)*A(3,3)-A(2,3)*A(3,1))+
0163 2 A(1,3)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
0164 IF(DET.EQ.0.D0) KS=1
0165 IF(DET.EQ.0.D0) RETURN
0166 DETA=B(1)*(A(2,2)*A(3,3)-A(2,3)*A(3,2))-
0167 1 A(1,2)*(B(2)*A(3,3)-A(2,3)*B(3))+
0168 2 A(1,3)*(B(2)*A(3,2)-B(3)*A(2,2))
0169 DETB=A(1,1)*(B(2)*A(3,3)-A(2,3)*B(3))-
0170 1 B(1)*(A(2,1)*A(3,3)-A(2,3)*A(3,1))+
0171 2 A(1,3)*(A(2,1)*B(3)-A(3,1)*B(2))
0172 DETC=A(1,1)*(A(2,2)*B(3)-B(2)*A(3,2))-
0173 1 A(1,2)*(A(2,1)*B(3)-B(2)*A(3,1))+
0174 2 B(1)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
0175 B(1)=DETA/DET
0176 B(2)=DETB/DET
0177 B(3)=DETC/DET
0178 RETURN
0179 END

```

```

0180 C E SUB W IS THE ROLL AXIS OF THE MISSILE. GIVEN THE TWO ASPECT ANGLES
0181 C THIS ROUTINE WILL DETERMINE THE ROLL AXIS UNIT VECTOR
0182 C AS WELL AS THE PITCH AND YAW FOR THE PARTICULAR
0183 C CASE IN QUESTION
0184 SUBROUTINE ESUBW(SIGMA1,SIGMA2,OMEGA,DT11,DT12)
0185 DOUBLE PRECISION DT11,DT12
0186 DOUBLE PRECISION SIGMA1,SIGMA2
0187 DOUBLE PRECISION OMEGA,ROOT1,ROOT2
0188 DOUBLE PRECISION DELT11,DELT12,COSS1,COSS2,DET
0189 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0190 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0191 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
0192 3 ERROR,EWX,EWY,EWZ
0193 INTEGER OUTPUT
0194 COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0195 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0196 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
0197 3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0198 DOUBLE PRECISION AA(3,3),A(3,3),B(3),C(3)
0199 IF(YR1.EQ.YR2.AND.YR1.EQ.0.D0) GO TO 90
0200 IF(ZR1.EQ.ZR2.AND.ZR1.EQ.0.D0) GO TO 95
0201 AA(1,1)=XR1
0202 AA(2,1)=XR2
0203 AA(3,1)=1.D0
0204 AA(1,2)=YR1
0205 AA(2,2)=YR2
0206 AA(3,2)=0.D0
0207 AA(1,3)=ZR1
0208 AA(2,3)=ZR2
0209 AA(3,3)=0.D0
0210 CALL ESWX(SIGMA1,SIGMA2,ROOT1,ROOT2,ISOL,DT11,DT12)
0211 IF(ISOL.GT.0) GO TO 10
0212 WRITE(OUTPUT,20)
0213 RETURN
0214 20 FORMAT(" ***NO SOLUTION AVAILABLE FOR THE ROLL AXIS**"/
0215 1 " **ESWX DID NOT RETURN A VALUE FOR EWX**"/)
0216 10 COSS1=DCOS(SIGMA1)
0217 COSS2=DCOS(SIGMA2)
0218 C(1)=COSS1
0219 C(2)=COSS2
0220 C(3)=ROOT1
0221 DO 1000 I=1,3
0222 B(I)=C(I)
0223 DO 1000 J=1,3
0224 1000 A(I,J)=AA(I,J)
0225 CALL SIMQ(AA,C,KS)
0226 IF(KS.EQ.0) GO TO 30
0227 WRITE(OUTPUT,40)
0228 40 FORMAT(" ***ERROR IN SIMQ. THE MATRIX FOR FINDING THE"/
0229 1 " ROLL AXIS IS SINGULAR*****")
0230 RETURN
0231 30 EWX=C(1)
0232 EWY=C(2)
0233 EWZ=C(3)
0234 CALL TIME(DELT11,DELT12,OMEGA,DT11,DT12)
0235 IF(DABS(DELT11-DELT12).GT.ERROR) GO TO 50
0236 IF(DABS(DELT12-DELT12).GT.ERROR) GO TO 50
0237 RETURN
0238 50 IF(ISOL.EQ.1) RETURN
0239 B(3)=ROOT2
0240 CALL SIMQ(A,B,KS)
0241 IF(KS.EQ.0) GO TO 60
0242 WRITE(OUTPUT,40)
0243 RETURN
0244 60 EWX=B(1)
0245 EWY=B(2)
0246 EWZ=B(3)
0247 CALL TIME(DELT11,DELT12,OMEGA,DT11,DT12)
0248 IF(DABS(DELT11-DELT12).GT.ERROR) GO TO 70
0249 IF(DABS(DELT12-DELT12).GT.ERROR) GO TO 70
0250 RETURN
0251 70 WRITE(OUTPUT,80)
0252 80 FORMAT(" **ESUBW**THERE IS NO SUITABLE ROLL AXIS**")
0253 * RETURN

```

```

0254 C YR1=YR2=0.D0
0255 90 DET=XR1*ZR2-XR2*ZR1
0256 IF(DET.NE.0.D0) GO TO 91
0257 WRITE(OUTPUT,93)
0258 93 FORMAT(" **R1 AND R2 ARE NOT LINEARLY INDEPENDENT**"/
0259 1 " *****NO SOLUTION POSSIBLE*****")
0260 RETURN
0261 91 COSS1=DCOS(SIGMA1)
0262 COSS2=DCOS(SIGMA2)
0263 EWX=(COSS1*ZR2-ZR1*COSS2)/DET
0264 EWZ=(XR1*COSS2-XR2*COSS1)/DET
0265 EWY=DSQRT(1.D0-EWX**2-EWZ**2)
0266 RETURN
0267 95 COSS1=DCOS(SIGMA1)
0268 COSS2=DCOS(SIGMA2)
0269 DET=XR1*YR2-YR1*XR2
0270 IF(DET.NE.0.D0) GO TO 96
0271 WRITE(OUTPUT,93)
0272 RETURN
0273 96 EWX=(COSS1*YR2-YR1*COSS2)/DET
0274 EWY=(XR1*COSS2-XR2*COSS1)/DET
0275 EWZ=DSQRT(1.D0-EWX**2-EWY**2)
0276 RETURN
0277 END

```

```

0278 C SIGMA 1 IS DETERMINED BY THE YAWSONDE EQUATION IN THIS ROUTINE
0279 SUBROUTINE SIG11(OMEGA,SIGMA1,DT11)
0280 DOUBLE PRECISION DT11,DTAN
0281 DOUBLE PRECISION OMEGA,A,ALPHA,SIGMA1
0282 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0283 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0284 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,P1,RADIAN,
0285 3 ERROR,EWX,EWY,EWZ
0286 INTEGER OUTPUT
0287 COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0288 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0289 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),P1,RADIAN,
0290 3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0291 A=DATAN(SKEW2)/DATAN(SKEW1)
0292 ALPHA=OMEGA*DT11-BETA
0293 SIGMA1=DATAN((DTAN(SKEW1)*DSQRT(1.D0+A*A+2.D0*A*DCOS(ALPHA)
0294 1 ))/DSIN(ALPHA))
0295 IF(SIGMA1.LT.0.D0)SIGMA1=SIGMA1+PI
0296 RETURN
0297 END

```

```

0298 C THE FOLLOWING CALCULATES THE SOLUTION OF SIGMA 2 FROM
0299 C THE ONE PRISM, TWO GROUND STATION FORMULATION
0300 SUBROUTINE SIG12(OMEGA,SIGMA1,DT12,SOL1,SOL2,ISOL)
0301 DOUBLE PRECISION DT12,DIFF2A,DIFF2B,DSIG2A,DSIG2B
0302 DOUBLE PRECISION OMEGA,SIGMA1,DTAN
0303 DOUBLE PRECISION A,SOL1,SOL2,SIG2A,SIG2B,B,RAD2,RAD,COSMEG
0304 DOUBLE PRECISION COSS1,COSS2A,COSS2B,SINS1,SINS2A,SINS2B
0305 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0306 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0307 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
0308 3 ERROR,EWX,EWY,EWZ
0309 INTEGER OUTPUT
0310 COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0311 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0312 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
0313 3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0314 A=DTAN(SIGMA1)*DCOS(OMEGA*DT12)
0315 ISOL=0
0316 SOL1=0.D0
0317 SOL2=0.D0
0318 B=COSMU/DCOS(SIGMA1)
0319 RAD2=A*A-B*B+1.D0
0320 IF(RAD2.GT.0.D0) GO TO 10
0321 WRITE(OUTPUT,100)
0322 100 FORMAT(" *****ERROR IN THE SIGMA 12 SUBROUTINE*****")
0323 SIG2A=0.D0
0324 RETURN
0325 10 RAD=DSQRT(RAD2)
0326 COSMEG=DCOS(OMEGA*DT12)
0327 SIG2A=DATAN((A*B-RAD)/(B+A*RAD))
0328 SIG2B=DATAN((A*B+RAD)/(B-A*RAD))
0329 IF(SIG2A.LT.0.D0)SIG2A=SIG2A+PI
0330 IF(SIG2B.LT.0.D0)SIG2B=SIG2B+PI
0331 COSS1=DCOS(SIGMA1)
0332 COSS2A=DCOS(SIG2A)
0333 SINS1=DSIN(SIGMA1)
0334 SINS2A=DSIN(SIG2A)
0335 COSS2B=DCOS(SIG2B)
0336 SINS2B=DSIN(SIG2B)
0337 DIFF2A=DAES(COSMU-COSS1*COSS2A-COSMEG*SINS1*SINS2A)
0338 DIFF2B=DAES(COSMU-COSS1*COSS2B-COSMEG*SINS1*SINS2B)
0339 DSIG2A=SIG2A*RADIAN
0340 DSIG2B=SIG2B*RADIAN
0341 IF(DIFF2A.GT.ERROR)GO TO 20
0342 C SIG2A IS A GOOD ROOT
0343 IF(DIFF2B.LT.ERROR) GO TO 15
0344 C SIG2A GOOD,SIG2B BAD
0345 SOL1=SIG2A
0346 ISOL=1
0347 RETURN
0348 C BOTH ROOTS ARE BAD
0349 12 WRITE(OUTPUT,110)
0350 110 FORMAT(" **THE TWO SOLUTIONS OF SIGMA2 ARE INCORRECT**")
0351 WRITE(OUTPUT,120)DSIG2A,DIFF2A,DSIG2B,DIFF2B
0352 120 FORMAT(" THE FIRST SOLUTION OF SIGMA 2 IS "D19.13/
0353 1 " WITH AN ERROR OF "D19.13/
0354 2 " THE SECOND SOLUTION IS "D19.13/
0355 3 " WITH AN ERROR OF "D19.13)
0356 WRITE(OUTPUT,999) SIG2A,SIG2B,COSS1,SINS1,COSS2A,SINS2A,COSMU
0357 1 ,COSMEG,RAD,SIGMA1,OMEGA,DT12,A,B,COSS2B,SINS2B
0358 999 FORMAT(" SIG2A="1PD23.16" SIG2B="D23.16" COSS1="D23.16/
0359 1 " SINS1="D23.16" COSS2A="D23.16" SINS2A="D23.16/
0360 2 "COSMU="D23.16" COSMEG="D23.16" RAD="D23.16" SIGMA1="D23.16/
0361 3 " OMEGA="D23.16" DT12="D2.16" A="D23.16" B="D23.16
0362 4 /" COSS2A="D23.16" SINS2B="D23.16/)
0363 RETURN
0364 C BOTH ROOTS ARE GOOD
0365 15 ISOL=2
0366 SOL1=SIG2A
0367 SOL2=SIG2B
0368 RETURN
0369 C SIG2A IS BAD, SIG2B IS GOOD
0370 20 IF(DIFF2B.GT.ERROR) GO TO 12
0371 SOL1=SIG2B
0372 ISOL=1
0373 RETURN
0374 END

```

```

C SUBROUTINE SETUP DETERMINES ALL OF THE VARIABLES IN THE
C COMMON BLOCK
  SUBROUTINE SETUP
  DOUBLE PRECISION X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
  1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
  2 YR1, ZR1, XR2, YR2, ZR2, ANSWER, PI, RADIAN,
  3 ERROR, EWX, EWY, EWZ
  INTEGER OUTPUT
  COMMON X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
  1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
  2 YR1, ZR1, XR2, YR2, ZR2, ANSWER(4,6), PI, RADIAN,
  3 ERROR, EWX, EWY, EWZ, NUMSOL, NREAD, OUTPUT
  RX1=X1-XM
  RY1=Y1-YM
  RZ1=Z1-ZM
  R1=DSQRT(RX1**2+RY1**2+RZ1**2)
  RX2=X2-XM
  RY2=Y2-YM
  RZ2=Z2-ZM
  R2=DSQRT(RX2**2+RY2**2+RZ2**2)
  COSMU=(RX1*RX2+RY1*RY2+RZ1*RZ2)/(R1*R2)
C MAKE ALL THE PLANAR COMPONENTS NORMALISED
  XR1=RX1/R1
  YR1=RY1/R1
  ZR1=RZ1/R1
  XR2=RX2/R2
  YR2=RY2/R2
  ZR2=RZ2/R2
C ZERO OUT THE ANSWER ARRAY, AND NUMSOL(NUMBER OF SOLUTIONS)
  DO 10 I=1,4
  DO 10 J=1,6
  10 ANSWER(I,J)=0.D0
  NUMSOL=0
  RETURN
  END

```

```

0410 C SOLVE IS THE MAIN ORGANIZER FOR THE EXPLICIT SOLUTION
0411 C OF PITCH AND YAW ANGLES GIVEN DELTA T 11 AND DELTA T 12
0412 C THIS ROUTINE IS RESPONSIBLE FOR THE CORRECT LINKING OF ALL
0413 C OTHER SUBROUTINES WITH THE EXCEPTION OF INPUT
0414 SUBROUTINE SOLVE(PITCH, YAW, OMEGA, DT11, DT12)
0415 DOUBLE PRECISION DT11, DT12
0416 DOUBLE PRECISION OMEGA, SIGMA1
0417 DOUBLE PRECISION SIG2A, SIG2B
0418 DOUBLE PRECISION PITCH, YAW
0419 DOUBLE PRECISION X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
0420 1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0421 2 YR1, ZR1, XR2, YR2, ZR2, ANSWER, PI, RADIAN,
0422 3 ERROR, EWX, EWY, EWZ
0423 INTEGER OUTPUT
0424 COMMON X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
0425 1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0426 2 YR1, ZR1, XR2, YR2, ZR2, ANSWER(4,6), PI, RADIAN,
0427 3 ERROR, EWX, EWY, EWZ, NUMSOL, NREAD, OUTPUT
0428 DIMENSION ITIME(5)
0429 C CALL BECTIM
0430 CALL EXEC(11, ITIME, IYEAR)
0431 ISTART=1000*ITIME(2)+10*ITIME(1)
0432 CALL SETUP
0433 CALL SIG11(OMEGA, SIGMA1, DT11)
0434 CALL SIG12(OMEGA, SIGMA1, DT12, SIG2A, SIG2B, IGOOD)
0435 IF(IGOOD.EQ.0)RETURN
0436 CALL ESUBW(SIGMA1, SIG2A, OMEGA, DT11, DT12)
0437 IF(IGOOD.EQ.2)CALL ESUBW(SIGMA1, SIG2B, OMEGA, DT11, DT12)
0438 ITIME=0
0439 C CALL MARKTIM(ITIME)
0440 CALL EXEC(11, ITIME, IYEAR)
0441 IFINIS=1000*ITIME(2)+10*ITIME(1)
0442 JTIME=IFINIS-ISTART
0443 WRITE(OUTPUT,1001)JTIME
0444 1001 FORMAT(2X,43(" ")/" THE TIME REQUIRED FOR THIS COMPUTATION WAS"
0445 1 /119" MILLISECONDS"/2X43(" ")/)
0446 RETURN
0447 END

```

```

0448 C QUAD FINDS THE QUADRATIC ROOTS TO A SPECIALIZED EQUATION
0449 SUBROUTINE QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
0450 DOUBLE PRECISION ROOT1,ROOT2,A,B,C,D,AQUAD,BQUAD,CQUAD,RADSQ,RAD
0451 INTEGER OUTPUT
0452 ROOT1=0.D0
0453 RCOT2=0.D0
0454 ISOL=0
0455 AQUAD=1.D0+B*B+D*D
0456 C NOTE THAT AQUAD IS ALWAYS GREATER THAN ONE
0457 BQUAD=2.D0*(A*B+C*D)
0458 CQUAD=A*A+C*C-1.D0
0459 RADSQ=BQUAD**2-4.D0*AQUAD*CQUAD
0460 IF(RADSQ.GE.0.D0) GO TO 10
0461 WRITE(OUTPUT,100)
0462 C THERE IS AN ERROR APPARENT
0463 100 FORMAT(/" ***ERROR IN QUADRATIC SOLUTION***"/
0464 1 " CANNOT TAKE THE SQUARE ROOT OF"/
0465 2 " A NEGATIVE NUMBER"/)
0466 RETURN
0467 10 RAD=DSQRT(RADSQ)
0468 ISOL=2
0469 IF(RAD.EQ.0.D0) ISOL=1
0470 C THE TWO SOLUTIONS ARE IDENTICAL
0471 ROOT1=- (BQUAD+RAD)/(2.D0*AQUAD)
0472 ROOT2=(-BQUAD+RAD)/(2.D0*AQUAD)
0473 RETURN
0474 END

```

```

0475 C ESWX SOLVES THE FOLLOWING SET OF EQUATIONS FOR EWX
0476 C   ER1 DOT EW = COS(SIGMA1)
0477 C   ER2 DOT EW = COS(SIGMA2)
0478 C   EW DOT EW = 1.D0
0479 C ROOT1 IS THE FIRST SOLUTION FOR EWX
0480 C ROOT2 IS THE SECOND SOLUTION
0481 C ISOL IS THE NUMBER OF SOLUTIONS FOR THE SET
0482 SUBROUTINE ESWX(SIGMA1,SIGMA2,ROOT1,ROOT2,ISOL,DT11,DT12)
0483 DOUBLE PRECISION SIGMA1,SIGMA2,DT11,DT12
0484 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0485 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSNU,XR1,
0486 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
0487 3 ERROR,EWX,EWY,EWZ
0488 INTEGER OUTPUT
0489 COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0490 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0491 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
0492 3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0493 DOUBLE PRECISION A,B,C,D,ROOT1,ROOT2,COSS1,COSS2
0494 DOUBLE PRECISION DENOM,DENNEW
0495 C SET UP THE ERROR DEFAULTS
0496   ROOT1=0.D0
0497   ROOT2=0.D0
0498   ISOL=0
0499 C GET THE DIRECTION COSINES FOR THE ASPECT ANGLES
0500   COSS1=BCOS(SIGMA1)
0501   COSS2=BCOS(SIGMA2)
0502 C NOW GO TO THE SOLUTION FOR THE NINE ZILLION CASES
0503   IF((ZR1*ZR2).NE.0.D0)GO TO 110
0504   IF(ZR1.EQ.0.D0) GO TO 10
0505   IF(ZR2.EQ.0.D0) GO TO 5
0506   WRITE(OUTPUT,1000)
0507 1000 FORMAT(" ***ERROR--SOME TYPE OF LOGIC MISTAKE***")
0508   RETURN
0509   5 IF(YR2.EQ.0.D0)GO TO 6
0510   A=COSS2/YR2
0511   B=-XR2/YR2
0512   C=(COSS1-A*YR1)/ZR1
0513   D=- (XR1+B*YR1)/ZR1
0514   CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
0515   RETURN
0516   6 IF(XR2.EQ.0.D0) GO TO 7
0517   ISOL=1
0518   ROOT1=COSS2/XR2
0519   RETURN
0520   7 WRITE(OUTPUT,1010)
0521 1010 FORMAT(" ***VECTOR ER2 IS THE ZERO VECTOR, NO SOLUTION**")
0522   RETURN
0523 C ZR1=0.D0
0524   10 IF(ZR2.EQ.0.D0)GO TO 11
0525 C ZR2.NE.0.D0
0526   IF(YR1.EQ.0.D0) GO TO 12
0527   A=COSS1/YR1
0528   B=-XR1/YR1
0529   C=(COSS1-YR2*A)/ZR2
0530   D=- (B*YR2+XR2)/ZR2
0531   CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
0532   RETURN
0533 C ZR1=YR1=0.D0
0534   12 IF(XR1.EQ.0.D0) GO TO 13
0535   ISOL=1
0536   ROOT1=COSS1/XR1
0537   RETURN
0538   13 WRITE(OUTPUT,1020)
0539 1020 FORMAT(" ***VECTOR ER1 IS THE ZERO VECTOR, NO SOLUTION**")
0540   RETURN

```

```

0541 C ZR1=0
0542   11 DENOM=XR1*YR2-XR2*YR1
0543     IF(DENOM.EQ.0.D0) GO TO 14
0544     ISOL=1
0545     ROOT1=YR2*COSS1-COSS2*YR1
0546     RETURN
0547   14 WRITE(OUTPUT,1040)
0548 C ZR1.NE.0.D0, AND ZR2.NE.0.D0--
0549 C THE MOST COMMON CASE
0550   110 DENOM=ZR2*YR1-YR2*ZR1
0551     IF(DENOM.EQ.0.D0) GO TO 120
0552     A=(ZR2*COSS1-ZR1*COSS2)/DENOM
0553     B=(XR2*ZR1-ZR2*XR1)/DENOM
0554     C=(COSS1-YR1*A)/ZR1
0555     D=- (XR1+B*YR1)/ZR1
0556     CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
0557     RETURN
0558 C DENOM=0.D0
0559   120 DENNEW=ZR2*XR1-XR2*ZR1
0560     IF(DENNEW.EQ.0.D0) GO TO 130
0561     ISOL=1
0562     ROOT1=(ZR2*COSS1-ZR1*COSS2)/DENNEW
0563     RETURN
0564   130 WRITE(OUTPUT,1040)
0565 1040 FORMAT(" **ER1 AND ER2 ARE NOT LINEARLY INDEPENDENT, "
0566           1 "NO SOLUTION")
0567     RETURN
0568     END

```

```

0569 C THIS ROUTINE CALCULATES THE TIME REQUIRED FOR DELT11
0570 C AND DELT12 TO ROLL THROUGH THE RESPECTIVE GROUND STATIONS
0571 SUBROUTINE TIME(DELTA11,DELTA12,OMEGA,DT11,DT12)
0572 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0573 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0574 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,P1,RADIAN,
0575 3 ERROR,EWX,EWY,EWZ
0576 INTEGER OUTPUT
0577 COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
0578 1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
0579 2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),P1,RADIAN,
0580 3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
0581 DOUBLE PRECISION DT11,DT12,DTAN
0582 DOUBLE PRECISION OMEGA,DASIN
0583 DOUBLE PRECISION XYMAG,PITCH,YAW,PITOUT,YAWOUT,SINPIT
0584 DOUBLE PRECISION SINYAW,COSYAW,COSPIT,N1,W1,E1,N2,E2,DELT11
0585 DOUBLE PRECISION DELT12,ERR11,ERR12
0586 C DOUBLE PRECISION DEG11,DEG12
0587 XYMAG=DSQRT(EWX**2+EWY**2)
0588 PITCH=DATN2(EWZ,XYMAG)
0589 YAW=DATN2(EWX,EWY)
0590 PITOUT=PITCH*RADIAN
0591 YAWOUT=YAW*RADIAN
0592 SINYAW=DSIN(YAW)
0593 SINPIT=DSIN(PITCH)
0594 COSYAW=DCOS(YAW)
0595 COSPIT=DCOS(PITCH)
0596 N1=COSYAW*RX1-SINYAW*RY1
0597 W1=SINYAW*COSPIT*RX1+COSYAW*COSPIT*RY1+SINPIT*RZ1
0598 E1=-SINYAW*SINPIT*RX1-COSYAW*SINPIT*RY1+COSPIT*RZ1
0599 N2=COSYAW*RX2-SINYAW*RY2
0600 E2=-SINYAW*SINPIT*RX2-COSYAW*SINPIT*RY2+COSPIT*RZ2
0601 DELT11=(BETA+DASIN(W1*DTAN(SKEW1)/DSQRT(E1*E1+N1*N1)))/OMEGA
0602 DELT12=(DATN2(E1,N1)-DATN2(E2,N2))/OMEGA
0603 C DEG11=OMEGA*DELT11*RADIAN
0604 C DEG12=OMEGA*DELT12*RADIAN
0605 ERR11=DABS(DELT11-DT11)
0606 ERR12=DABS(DELT12-DT12)
0607 NUMSOL=NUMSOL+1
0608 ANSWER(NUMSOL,1)=PITOUT
0609 ANSWER(NUMSOL,2)=YAWOUT
0610 ANSWER(NUMSOL,3)=DELT11
0611 ANSWER(NUMSOL,4)=DELT12
0612 ANSWER(NUMSOL,5)=ERR11
0613 ANSWER(NUMSOL,6)=ERR12
0614 IF(DABS(PITOUT).GT.90.D0.OR.DABS(YAWOUT).GT.90.D0)RETURN
0615 IF(DABS(DELT12-DT12).GT.ERROR)RETURN
0616 WRITE(OUTPUT,82)PITOUT,YAWOUT
0617 WRITE(OUTPUT,999)DELT11,DELT12
0618 999 FORMAT(" DELT11="1PD23.16" DELT12="D23.16)
0619 82 FORMAT(" USING PITCH="2X,F11.6" DEG AND YAW="2X,F11.6" DEG")
0620 C WRITE(OUTPUT,401)DEG11,DEG12
0621 C401 FORMAT("/" DELTA T 11 ROLLS THROUGH"F20.15" DEGREES"/
0622 C 1 " DELTA T 12 ROLLS THROUGH"F20.15" DEGREES")
0623 CALL ERR(DT11,DT12,OMEGA)
0624 RETURN
0625 END

```

```
0626 C THIS IS A ROUTINE TO AUGMENT THE MINI-COMPUTER
0627     DOUBLE PRECISION FUNCTION DASIN(X)
0628     DOUBLE PRECISION X
0629     DASIN=DATN2(X,DSQRT(1.D0-X*X))
0630     RETURN
0631     END
0632 C ANOTHER SUBROUTINE FOR THE EDIFICATION OF A MINI-COMPUTER
0633     DOUBLE PRECISION FUNCTION DTAN(X)
0634     DOUBLE PRECISION X
0635     DTAN=DSIN(X)/DCOS(X)
0636     RETURN
0637     END
```

```

0638 C ERROR CALCULATES THE ERRORS OF SIGMA1,SIGMA2,PITCH, AND YAW.
0639 C THIS ANALYSIS IS BASED ON TOTAL DIFFERENTIALS WITH
0640 C QUADRATIC PERTURBATIONS NEGLECTED, AND WITH SMALL ANGLE APPROXIMATIONS
0641 C SO THAT SIN(DELTA P)=DELTA P, AND COS(DELTA P)=1
0642 SUBROUTINE ERR(DT11,DT12,OMEGA)
0643 DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2, XM, YM, ZM, SKEW1,
0644 1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0645 2 YR1, ZR1, XR2, YR2, ZR2, ANSWER, P1, RADIAN,
0646 3 ERROR, EWX, EWY, EWZ
0647 INTEGER OUTPUT
0648 COMMON X1, Y1, Z1, X2, Y2, Z2, XM, YM, ZM, SKEW1,
0649 1 SKEW2, BETA, RX1, RY1, RZ1, R1, RX2, RY2, RZ2, R2, COSMU, XR1,
0650 2 YR1, ZR1, XR2, YR2, ZR2, ANSWER(4,6), P1, RADIAN,
0651 3 ERROR, EWX, EWY, EWZ, NUMSOL, NREAD, OUTPUT
0652 DOUBLE PRECISION A, ALPHA, ARG, ATANG1, ATANG2, ATERM, A1, A2, BTERM,
0653 1 B1, B2, COSALF, COSG1, COSPIT, COSS1, COSS2, COSW, COSYAW, CP1, CP2, CTERM,
0654 2 CT1, CT2, C1, C2, DALF, DENOM, DG1, DG2, DMU, DPITCH, DP1, DP2, DSIG1, DSIG2,
0655 DTAN, DT1, DT11, DT12, DT2, DWDT12, DYAW, OMEGA, PHI1, PHI2, PITCH, RAD,
0656 4 SIGMA1, SINALF, SINMU, SINPIT, SINS1, SINS2, SINW, SINYAW, SP1, SP2, ST1,
0657 5 ST2, TANG1, TERM1, TERM2, TERM3, THETA1, THETA2, XNUMER, XYMAG, XYTIAG1,
0658 6 XYMAG2, YAW
0659 DOUBLE PRECISION BOUND, DELP, DELY
0660 DOUBLE PRECISION S1OUT, S2OUT, DS1OUT, DS2OUT
0661 DO 1000 I=1,5
0662 BOUND=1.D0**(-I+1)
0663 9 FORMAT(" NOW INSIDE THE ERROR ANALYSIS ROUTINE, WITH ALL"/
0664 1 " PERTURBATIONS SET AS "F10.5)
0665 WRITE(OUTPUT,9) BOUND
0666 DP1=BOUND
0667 DP2=BOUND
0668 DP2=BOUND
0669 DWDT12=BOUND
0670 DALF=BOUND
0671 DG1=BOUND
0672 DG2=BOUND
0673 DT1=BOUND
0674 DT2=BOUND
0675 THETA1=DATN2(XR1, YR1)
0676 THETA2=DATN2(XR2, YR2)
0677 XYMAG1=DSQRT(XR1**2+YR1**2)
0678 XYMAG2=DSQRT(XR2**2+YR2**2)
0679 PHI1=DATN2(ZR1, XYMAG1)
0680 PHI2=DATN2(ZR2, XYMAG2)
0681 CP1=DCOS(PHI1)
0682 CP2=DCOS(PHI2)
0683 SP1=DSIN(PHI1)
0684 SP2=DSIN(PHI2)
0685 ST1=DSIN(THETA1)
0686 ST2=DSIN(THETA2)
0687 CT1=DCOS(THETA1)
0688 CT2=DCOS(THETA2)
0689 YAW=DATN2(EWX, EWY)
0690 XYMAG=DSQRT(EWX**2+EWY**2)
0691 PITCH=DATN2(EWZ, XYMAG)
0692 COSPIT=DCOS(PITCH)
0693 SINPIT=DSIN(PITCH)
0694 COSYAW=DCOS(YAW)
0695 SINYAW=DSIN(YAW)
0696 C CALCULATE MU AND DMU
0697 COSMU=XR1*XR2+YR1*YR2+ZR1*ZR2
0698 SINMU=DSQRT(1.D0-COSMU**2)
0699 XNUMER=SP1*ST1*(SP2*ST2*DP2-CP2*CT2*DT2)+
0700 1 CP2*ST2*(SP1*ST1*DP1-CP1*CT1*DT1)+
0701 2 CP1*CT1*(SP2*CT2*DP2+CP2*ST2*DT2)+
0702 3 CP2*CT2*(SP1*CT1*DP1+CP1*ST1*DT1)-
0703 4 CP1*SP2*DP1-SP1*CP2*DP2
0704 DMU=XNUMER/SINMU

```

```

0705 C CALCULATE DSIG1 THE ERROR OF SIGMA1
0706 ATANG1=DATAN(SKEW1)
0707 ATANG2=DATAN(SKEW2)
0708 ALPHA=OMEGA*DT11-BETA
0709 COSG1=DCOS(SKEW1)
0710 A=ATANG2/ATANG1
0711 COSALF=DCOS(ALPHA)
0712 SINALF=DSIN(ALPHA)
0713 TANG1=DTAN(SKEW1)
0714 RAD=DSQRT(1.D0+A*A+2.D0*A*COSALF)
0715 TERM1=TANG1*(COSALF*RAD/SINALF**2-A/RAD)*DALF
0716 TERM2=(RAD/(SINALF*COSG1*COSG1)-TANG1/(SINALF*RAD)*(A+COSALF))*
0717 1 ATANG2/ATANG1**2*(1.D0/(1.D0+SKEW1**2))*DG1
0718 TERM3=TANG1/SINALF*(A+COSALF)/RAD/ATANG1/(1.D0+SKEW2**2)*DC2
0719 ARC=TANG1/SINALF*RAD
0720 DSIG1=1.D0/(1.D0+ARC**2)*(TERM1+TERM2+TERM3)
0721 SIGMA1=DATAN(ARC)
0722 S1OUT=SIGMA1*RADIAN
0723 DS1OUT=DSIG1*RADIAN
0724 WRITE(OUTPUT,10)SIGMA1,DSIG1,S1OUT,DS1OUT
0725 10 FORMAT(" SIGMA1="1PD23.15" RADIANS, WITH DSIG1="D23.15/
0726 1 " = "0PF23.15" DEGREES = "F23.15" DEGREES")
0727 C CALCULATE DSIG2 FROM IMPLICIT TOTAL DIFFERENTIAL OF THE
0728 C TWO STATION ASPECT ANGLE FORMULA
0729 COSS1=DCOS(SIGMA1)
0730 SINS1=DSIN(SIGMA1)
0731 COSS2=XI*XR1+YM*YR1+ZN*ZR1
0732 SINS2=DSQRT(1.D0-COSS2**2)
0733 COSW=DCOS(OMEGA*DT12)
0734 SINW=DSIN(OMEGA*DT12)
0735 DENOM=SINS1*COSS2*COSW-COSS1*SINS2
0736 ATERM=(SINS1*COSS2-COSS1*SINS2*COSW)*DSIG1
0737 BTERM=(SINS1*SINS2*SINW)*DWD12
0738 CTERM=COSMU*DMU
0739 DSIG2=(ATERM+BTERM-CTERM)/DENOM
0740 SIGMA2=DASIN(SINS2)
0741 S2OUT=SIGMA2*RADIAN
0742 DS2OUT=DS2*RADIAN
0743 WRITE(OUTPUT,20)SIGMA2,DSIG2,S2OUT,DS2OUT
0744 20 FORMAT(" SIGMA2="1PD23.15" RADIANS, WITH DSIG2="D23.15/
0745 1 " = "0PF23.15" DEGREES = "F23.15" DEGREES")
0746 A1=-SINPIT*SINYAW*CP1*ST1-SINPIT*COSYAW*CT1+COSPIT*SP1
0747 B1=COSPIT*COSYAW*CP1*ST1-COSPIT*SINYAW*CP1*CT1
0748 C1=-SINS1*DSIG1+COSPIT*SINYAW*(SP1*ST1*DP1-CP1*CT1*DT1)
0749 1 +COSPIT*COSYAW*(SP1*CT1*DP1+CP1*ST1*DT1)-SINPIT*CP1*DP1
0750 A2=-SINPIT*SINYAW*CP2*ST2-SINPIT*COSYAW*CT2+COSPIT*SP2
0751 B2=COSPIT*COSYAW*CP2*ST2-COSPIT*SINYAW*CP2*CT2
0752 C2=-SINS2*DSIG2+COSPIT*SINYAW*(SP2*ST2*DP2-CP2*CT2*DT2)
0753 1 +COSPIT*COSYAW*(SP2*CT2*DP2+CP2*ST2*DT2)-SINPIT*CP2*DP2
0754 DENOM=A1*B2-A2*B1
0755 DPITCH=(C1*B2-C2*B1)/DENOM
0756 DYAW=(A1*C2-A2*C1)/DENOM
0757 DELP=DPITCH*RADIAN
0758 DELY=DYAW*RADIAN
0759 WRITE(OUTPUT,30)DPITCH,DYAW,DELP,DELY
0760 30 FORMAT(" THE ERROR OF PITCH IS"1PD23.15" YAW IS"D23.15" RADIANS"/
0761 1 " IN DEGREES, THIS IS "0PF23.15" "F23.15)
0762 1000 CONTINUE
0763 RETURN
0764 END
0765 ENDS

```

APPENDIX II

LTPOS

Program to Determine Vehicle Position

(Interdata OS/32 Version)

SBATCH

C PROGRAM LIPOS(3,85)

C

C PROGRAM TO READ LASER TRACKER DATA TAPE, UNPACK DATA,

C SMOOTH AND OUTPUT TO LINE PRINTER IN TABULAR FORM

C

C

COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),

X LIM(32),IROV(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),

X XL,YL,ZL,IH1,IM1,SEC1

X PULST1(320),PULST2(320),EPOCH(320),PULSW(320),PULSP(320)

C

C

C DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1

C X EPOCH,PULSW,PULSP

C INTEGER PULST1,PULST2

INTEGER*4 HR,WW,J,MN,MS

INTEGER*4 IH,IM,IH1,IM1

DIMENSION IP(5),IJ(2)

DIMENSION IST(6)

DIMENSION WW(448),ISEC(32),IMS(32)

EQUIVALENCE (IP(1),LU),(IP(2),LP),(IP(3),NFILE)

EQUIVALENCE (I,IJ(1)),(J,IJ(2)),(IP(4),NREC)

C

C

DATA RADEG/57.29577951/,PI/3.141592654/,IPAGE/1/,IPNT/0/

DATA PI2/6.283185308/,AZ0/90./,ELO/0./

C

C GET USER'S CONSOLE & LPR LU'S

C

CALL CLOSE(8,IST1)

CALL CLOSE(1,IST2)

CALL CLOSE(6,IST3)

CALL OPENW(1,'CON:',4,0,0,IST4)

CALL OPENW(6,'PR:',2,0,0,IST5)

CALL OPENW(8,'MAG0:',0,0,0,IST6)

IF(IST1.NE.0) GO TO 400

125 CONTINUE

IF(IST2.NE.0) GO TO 401

126 CONTINUE

IF(IST3.NE.0) GO TO 402

127 CONTINUE

IF(IST4.NE.0) GO TO 403

128 CONTINUE

IF(IST5.NE.0) GO TO 404

129 CONTINUE

IF(IST6.NE.0) GO TO 405

130 CONTINUE

LU=0

LP=0

NFILE=0

NREC=0

IF(LU.EQ.0) LU = 1

IF(LP.EQ.0) LP = 6

IF(NFILE.EQ.0) GO TO 2

IF(NREC.EQ.0) NREC = 1

C

C POSITION TAPE TO FILE & REC # (DEFAULTS = 1)

C

CALL MTRFL(NFILE,NREC)

C

```

C   READ FIRST TWO WORDS FOR T=0
C
C   2 CONTINUE
C   CALL EXEC(1,110B,IJ,2)
   READ(B)WW
C 120 FORMAT(2A2)
C   SEPARATE INTO H,M,S
C   UNPACK DATA
   K=1
   L=K-1
   HR=0
   IARG1=14*L+1
   J=WW(IARG1)
   HR=HR+(ISHFT(J,-31))*20
   HR=HR+(ISHFT((ISHFT(J,1)),-31))*10
   HR=HR+(ISHFT((ISHFT(J,2)),-28))
   IH1 = HR
   MN=0
   MN=MN+(ISHFT(ISHFT(J,6),-31))*40
   MN=MN+(ISHFT(ISHFT(J,7),-31))*20
   MN=MN+(ISHFT(ISHFT(J,8),-31))*10
   MN=MN+(ISHFT(ISHFT(J,9),-31))*8
   MN=MN+(ISHFT(ISHFT(J,10),-31))*4
   MN=MN+(ISHFT(ISHFT(J,11),-31))*2
   MN=MN+(ISHFT(ISHFT(J,12),-31))*1
   IM1 =MN
   KSEC=0
   KSEC=KSEC+(ISHFT(ISHFT(J,13),-31))*40
   KSEC=KSEC+(ISHFT(ISHFT(J,14),-31))*20
   KSEC=KSEC+(ISHFT(ISHFT(J,15),-31))*10
   KSEC=KSEC+(ISHFT(ISHFT(J,16),-31))*8
   KSEC=KSEC+(ISHFT(ISHFT(J,17),-31))*4
   KSEC=KSEC+(ISHFT(ISHFT(J,18),-31))*2
   KSEC=KSEC+(ISHFT(ISHFT(J,19),-31))*1
C 40  ISEC(K)=KSEC
   MS=0
   MS=MS+(ISHFT(ISHFT(J,20),-31))*800
   MS=MS+(ISHFT(ISHFT(J,21),-31))*400
   MS=MS+(ISHFT(ISHFT(J,22),-31))*200
   MS=MS+(ISHFT(ISHFT(J,23),-31))*100
   MS=MS+(ISHFT(ISHFT(J,24),-31))*80
   MS=MS+(ISHFT(ISHFT(J,25),-31))*40
   MS=MS+(ISHFT(ISHFT(J,26),-31))*20
   MS=MS+(ISHFT(ISHFT(J,27),-31))*10
   MS=MS+(ISHFT(ISHFT(J,28),-31))*8
   MS=MS+(ISHFT(ISHFT(J,29),-31))*4
   MS=MS+(ISHFT(ISHFT(J,30),-31))*2
   MS=MS+(ISHFT(ISHFT(J,31),-31))*1
   IMS(K)=MS
   SEC1=FLOAT(ISEC(K))+FLOAT(IMS(K))/1000.
C   IH1 = ISHFT(IAND(I,140000B),-14)*10 +
C   X      ISHFT(IAND(I,360000B),-10)
C   IM1  = ISHFT(IAND(I,16000B),-7)*10 +
C   X      ISHFT(IAND(I,1700B),-3)
C   SEC1 = FLOAT(IAND(I,78))*10. +
C   X      FLOAT(ISHFT(IAND(J,1700000B),-12)) +
C   X      FLOAT(ISHFT(IAND(J,74000B),-8))/10. +
C   X      FLOAT(ISHFT(IAND(J,3600B),-4))/100. +
C   X      FLOAT(IAND(J,178))/1000.
C   WRITE(6,998)IH1,IM1,SEC1
C998  FORMAT(2X,I20,2X,I20,2X,F15.3)
   CALL READR(60,LE,IE,IH1,IM1,SEC1)
C   BACK UP TO START OF RECORD

```

```

C      CALL EXEC(3,2106)
      REWIND 8
C      WRITE(LU,100)
C 100  FORMAT('ENTER AZIMUTH,ELEVATION OFFSET (DEG)')
C      READ(LU,*)AZO,ELO
      AZO = AZO/RADEG
      ELO = ELO/RADEG
C      GET LAUNCHER COORDS.
C      WRITE(LU,110)
C 110  FORMAT('ENTER LAUNCHER COORDINATES (X,Y,Z)')
C      READ(LU,*)XL,YL,ZL
C
C      READ DATA RECORD
C
C      1 CALL UNPCK(IEOF)
C
C      OFFSET CORRECTIONS
C
C      DO 10 I=1,32
C      AZIMUTH
      AZ(I) = (AZ(I)*360./262144.)/RADEG +AZO
      IF(AZ(I).GT.PI ) AZ(I)=AZ(I)-PI2
      IF(AZ(I).LT.-PI ) AZ(I)=AZ(I)+PI2
C      ELEVATION
      EL(I) = (EL(I)*360./262144.)/RADEG +ELO
      IF(EL(I).GT.PI) EL(I)=EL(I)-PI2
      IF(EL(I).LT.-PI ) EL(I)=EL(I)+PI2
C      TILT
      AZ(I) = ATAN2 ( SIN(AZ(I))* COS(EL(I)) ,  COS(AZ(I))* COS(EL(I)))
      AZ(I) = AZ(I)*RADEG
      EL(I) = EL(I)*RADEG
C      WRITE(6,600)AZ(I),EL(I),RA(I)
C 600  FORMAT(10X,3(3X,F13.7))
      10 CONTINUE
      CALL OUTPI(LP,IEOF,IPNT,IPAGE)
      IF(IEOF.NE.1) GO TO 1
999  REWIND 8
      WRITE(LP,200)
200  FORMAT('1')
      GO TO 99
400  WRITE(1,450) IST1
      GO TO 125
450  FORMAT('DEVICE ASSIGNMENT ERROR***IST1=',I5)
401  WRITE(1,452) IST2
      GO TO 126
402  WRITE(1,453) IST3
      GO TO 127
403  WRITE(1,454) IST4
      GO TO 128
404  WRITE(1,455) IST5
      GO TO 129
405  WRITE(1,456) IST6
      GO TO 130
452  FORMAT('DEVICE ASSIGNMENT ERROR***IST2=',I5)
453  FORMAT('DEVICE ASSIGNMENT ERROR***IST3=',I5)
454  FORMAT('DEVICE ASSIGNMENT ERROR***IST4=',I5)
455  FORMAT('DEVICE ASSIGNMENT ERROR***IST5=',I5)
456  FORMAT('DEVICE ASSIGNMENT ERROR***IST6=',I5)
99  CONTINUE
      END
      SUBROUTINE TREL(NFILE,NREC)
      IFILE = NFILE - 1
      IREC  = NREC  - 1

```

```

C      FIND FILE
      1 CONTINUE
      IF(IFILE.LE.0) GOTO 10
C EXEC CALL REMOVED HERE
      IFILE = IFILE - 1
      GOTO 1
C      FIND RECORD #
      10 CONTINUE
      IF(IREC.LE.0) GOTO 99
C EXEC CALL REMOVED HERE
      IREC = IREC - 1
      GOTO 10
      99 RETURN
      END
      SUBROUTINE HEADR(LU,LP,IH,IM,SEC)
C
C      ROUTINE TO PRINT HEADER SHEET FOR LASER TRACKER
C      DATA LIST
C
      DIMENSION IPROJ(10),IDATE(5)
C
C      DOUBLE PRECISION IH,IM,SEC
      INTEGER*4 IH,IH1,IM,IM1
C
      WRITE(LU,100)
      100 FORMAT('ENTER PROJECT NAME')
      READ(LU,101)(IPROJ(K),K=1,10)
      101 FORMAT(10A2)
      WRITE(LU,110)
      110 FORMAT('ENTER TEST DATE (EX: 01 JAN 78)')
      READ(LU,111)(IDATE(K),K=1,5)
      WRITE(LU,120)
      120 FORMAT('TIME OF TEST')
      READ(LU,121)IH,IM,SEC
      121 FORMAT(2(I2,X),F6.3)
      111 FORMAT(5A2)
      WRITE(LP,200)
      200 FORMAT(1H1,20(/),
X          40X,'US ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND',/,
X          51X,'TECHNOLOGY LABORATORY',/,
X          48X,'SYSTEMS SIMULATION DIRECTORATE',/,
X          53X,'SYSTEMS EVALUATION',/,
X          50X,'LAMPANS - LASER TRACKER',/)
      WRITE(LP,210)(IPROJ(K),K=1,10),(IDATE(K),K=1,5),IH,IM,SEC
      210 FORMAT(15X,10A2,23X,5A2,26X,2(I2,':'),F6.3)
      RETURN
      END
      SUBROUTINE UNPCK(IEOF)
C
C      ROUTINE TO READ LAMPANS DATA TAPE AND UNPACK DATA
C      INTO VARIABLES (HOURS,MINUTES,SEC,AZ,EL,RA; ETC.)
C
C      RETURNS:
C      IEOF = 0 IF MORE DATA ON TAPE
C      IEOF = 1 IF END OF FILE DETECTED WITH READ
C
C
      COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
X          LIM(32),IROY(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),
X          XL,YL,ZL,IH1,IM1,SEC1
C          PULST1(320),PULST2(320),EPOCH(320),PULSA(320),PULSP(320)
C

```

```
DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
DOUBLE PRECISION EPOCH,PULSW,PULSP
```

```
DIMENSION WW(448),ISEC(32),IMS(32)
INTEGER*4 HR,WW,J,MN,MS
INTEGER*4 IH,IM,IH1,IM1
INTEGER PULST1,PULST2
```

```
IEOF = 0
```

```
READ RECORD
```

```
J = 0
```

```
1 READ(8)WW
```

```
CHECK FOR EOF
```

```
CALL EXEC(13,8,IEQ15) *****
IEQ15=IAND(IEQ15,200B) *****
IF(IEQ15.NE.0) IEOF = 1
IF(IEOF.EQ.1) GOTO 99
```

```
UNPACK DATA
DO 125 K=1,32
L=K-1
```

```
0 HR=0
```

```
IARG1=14*L+1
```

```
J=WW(IARG1)
```

```
HR=HR+(ISHFT(J,-31))*20
```

```
HR=HR+(ISHFT((ISHFT(J,1)),-31))*10
```

```
HR=HR+(ISHFT((ISHFT(J,2)),-28))
```

```
IH(K)=FLOAT(HR)-IH1
```

```
MN=0
```

```
MN=MN+(ISHFT(ISHFT(J,6)),-31))*40
```

```
MN=MN+(ISHFT(ISHFT(J,7)),-31))*20
```

```
MN=MN+(ISHFT(ISHFT(J,8)),-31))*10
```

```
MN=MN+(ISHFT(ISHFT(J,9)),-31))*8
```

```
MN=MN+(ISHFT(ISHFT(J,10)),-31))*4
```

```
MN=MN+(ISHFT(ISHFT(J,11)),-31))*2
```

```
MN=MN+(ISHFT(ISHFT(J,12)),-31))*1
```

```
0 IM(K)=MN-IM1
```

```
KSEC=0
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,13)),-31))*40
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,14)),-31))*20
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,15)),-31))*10
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,16)),-31))*8
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,17)),-31))*4
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,18)),-31))*2
```

```
KSEC=KSEC+(ISHFT(ISHFT(J,19)),-31))*1
```

```
0 ISEC(K)=KSEC
```

```
MS=0
```

```
MS=MS+(ISHFT(ISHFT(J,20)),-31))*800
```

```
MS=MS+(ISHFT(ISHFT(J,21)),-31))*400
```

```
MS=MS+(ISHFT(ISHFT(J,22)),-31))*200
```

```
MS=MS+(ISHFT(ISHFT(J,23)),-31))*100
```

```
MS=MS+(ISHFT(ISHFT(J,24)),-31))*80
```

```
MS=MS+(ISHFT(ISHFT(J,25)),-31))*40
```

```
MS=MS+(ISHFT(ISHFT(J,26)),-31))*20
```

```
MS=MS+(ISHFT(ISHFT(J,27)),-31))*10
```

```
MS=MS+(ISHFT(ISHFT(J,28)),-31))*8
```

```
MS=MS+(ISHFT(ISHFT(J,29)),-31))*4
```

```

MS=MS+(ISHFT(ISHFT(J,31),-31))*1
50  IMS(K)=MS
SEC(K)=FLOAT(ISEC(K))+FLOAT(IMS(K))/1000,-SEC1
Iww1=ISHFT(ISHFT(ISHFT(ww(14*L+3),16),-30),16)
Iww2=ISHFT(ww(14*L+2),-16)
60  AZ(K)=FLOAT(IOR(Iww1,Iww2))
Iww1=ISHFT(ISHFT(ww(14*L+2),16),-16)
Iww2=ISHFT(ISHFT(ISHFT(ww(14*L+3),18),-30),16)
70  EL(K)=FLOAT(IOR(Iww1,Iww2))
Iww1=ISHFT(ww(14*L+3),-16)
Iww2=ISHFT(ISHFT(ISHFT(ww(14*L+3),20),-30),16)
80  RA(K)=FLOAT(IOR(Iww1,Iww2))
LIM(K)=ISHFT(ISHFT(ww(14*L+3),23),-31)
C   IANS(K,9)=ISHFT(ISHFT(ww(14*L+3),24),-30)
C   IANS(K,10)=ISHFT(ISHFT(ww(14*L+3),26),-30)
IROV(K)=ISHFT(ISHFT(ww(14*L+3),28),-31)
ISP(K)=ISHFT(ISHFT(ww(14*L+3),29),-31)
C   IANS(K,13)=ISHFT(ISHFT(ww(14*L+3),30),-31)
C   IANS(K,14)=ISHFT(ISHFT(ww(14*L+3),31),-31)
C 90  IANS(K,15)=ISHFT(ww(14*L+4),-27)
C   IANS(K,16)=ISHFT(ISHFT(ww(14*L+4),5),-27)
C   IANS(K,17)=ISHFT(ISHFT(ww(14*L+4),10),-31)
C   IANS(K,18)=ISHFT(ISHFT(ww(14*L+4),11),-31)
LSRON(K)=ISHFT(ISHFT(ww(14*L+4),12),-31)
100  ICMP(K)=ISHFT(ISHFT(ww(14*L+4),13),-31)
IAUTO(K)=ISHFT(ISHFT(ww(14*L+4),14),-31)
C   IANS(K,22)=ISHFT(ISHFT(ww(14*L+4),15),-31)
C   IANS(K,23)=ISHFT(ISHFT(ww(14*L+4),16),-24)
C   IANS(K,24)=ISHFT(ISHFT(ww(14*L+4),24),-29)
C 110  IANS(K,25)=ISHFT(ISHFT(ww(14*L+4),27),-27)
M=0
DO 125 I=1,46,5
J=I-1
M=M+1
IC=M+4+14*L
C   IANS(K,J+26)=ISHFT(ww(IC),-31)
C   IANS(K,J+27)=ISHFT(ISHFT(ww(IC),1)-28)
C   IANS(K,J+28)=(ISHFT(ISHFT(ww(IC),5),-21))*50.0E-9
C   IANS(K,J+29)=ISHFT(ISHFT(ww(IC),16),-31)
C 120  IANS(K,J+30)=(ISHFT(ISHFT(ww(IC),17),-18))*50.0E-9
125  CONTINUE
C
C
C   WRITE(1,101)
C 101  FORMAT(5X,'SUBROUTINE UNPCK DK')
99  RETURN
END
SUBROUTINE OUIPI(LP,IEOF,IPNT,IPAGE)
C
C
COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
X   LIM(32),IROV(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),
X   XL,YL,ZL,IH1,IM1,SEC1
C   X   PULST1(320),PULST2(320),EPOCH(320),PULSW(320),PULSP(320)
C
C   DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
C   INTEGER*4 IH,IM,IH1,IM1
C   X   EPOCH,PULSW,PULSP
C   INTEGER PULST1,PULST2
C
DIMENSION TIME(100),X(100),Y(100),XDD(100),Y(100),YD(100),
X   YDD(100),Z(100),ZD(100),ZDD(100),JLPS(100),JROV(100),
X   JSIG(100),JLSK(100),JCF(100),JAJT(100)

```

```

C      DIMENSION D(100)
C
C      DOUBLE PRECISION TIME
      REAL JK
C
C      LOAD OUTPUT BUFFERS
C      WRITE(1,105)
C105  FORMAT(5X,'MADE IT INTO SUBROUTINE OUTPT')
C
      DO 100 I=1,32
      IPNT = IPNT+1
      X(IPNT) = AZ(I)
      Y(IPNT) = EL(I)
      Z(IPNT) = RA(I)
      JLIM(IPNT)=LIM(I)
      JROV(IPNT)=IROV(I)
      JSIG(IPNT)=ISP(I)
      JLSR(IPNT)=LSRON(I)
      JCMP(IPNT)=ICMP(I)
      JAUTO(IPNT)=IAUTO(I)
      TIME(IPNT)=(IH(I))*3600. +(IM(I))*60.+SEC(I)
C      WRITE(6,500)(EPOCH(K),PULST1(K),PULST2(K),PULSW(K),
C      X      PULSP(K),K=1,320)
C 500  FORMAT(X,F22.11,4(5X,I10),/,)
      IF(IPNT.LT.100) GOTO 100
      1 IF(IPNT.EQ.0) GOTO 99
C      WRITE(6,500)(XDD(K),YDD(K),ZDD(K),K=1,IPNT)
      500  FORMAT(3(5X,E13.7))
C      WRITE(1,106)
C106  FORMAT(2X,'MADE IT UP TO CALL TO FIT')
      CALL FIT(TIME,X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD,IPNT)
C      WRITE(6,500)(XDD(K),YDD(K),ZDD(K),K=1,IPNT)
C      WRITE(1,501)
C 501  FORMAT(' AFTER 'FIT'',/)
C      WRITE(6,500)(X(K),Y(K),Z(K),K=1,IPNT)
      DO 20 J = 1,IPNT
      CALL XYZCV(X(J),XD(J),XDD(J),Y(J),YD(J),YDD(J),Z(J),ZD(J),ZDD(J))
C      WRITE(1,107)
C107  FORMAT(2X,'BACK FROM XYZCV')
      20  CONTINUE
      DO 35 J = 1,IPNT,50
      JK=0.
      XS=0.
      YS=0.
      ZS=0.
      XV=0.
      YV=0.
      ZV=0.
      XA=0.
      YA=0.
      ZA=0.
      IPAGE = IPAGE+1
      CALL LABEL(LP,IPAGE)
      KK=J+49
      IF((IPNT.LT.100).AND.(J.GT.50))KK=IPNT
      IF(IPNT.LT.50) KK=IPNT
      DO 30 K=J,KK
      JK=JK+1.
      XS=XS+X(K)
      YS=YS+Y(K)
      ZS=ZS+Z(K)
      XV=XV+XD(K)
      YV=YV+YD(K)

```

```

ZV=ZV+ZD(K)
XA=XA+XDD(K)
YA=YA+YDD(K)
ZA=ZA+ZDD(K)
WRITE(LP,600) TIME(K),X(K),Y(K),Z(K),XD(K),YD(K),ZD(K),
X XDD(K),YDD(K),ZDD(K),JLIM(K),JRDV(K),JSIG(K),
X JLSR(K),JCMP(K),JAUTO(K)
30 CONTINUE
XS=XS/JK
YS=YS/JK
ZS=ZS/JK
XV=XV/JK
YV=YV/JK
ZV=ZV/JK
XA=XA/JK
YA=YA/JK
ZA=ZA/JK
35 WRITE(LP,601)XS,YS,ZS,XV,YV,ZV,XA,YA,ZA
601 FORMAT(/,' MEANS:',4X,9(X,F10.2))
IPNT=0
C*****
C IEOF=1
C*****
IF(IEOF.EQ.1) GOTO 99
100 CONTINUE
600 FORMAT(X,F7.3,3X,9(X,F10.2),6I3)
IF(IEOF.EQ.0) GOTO 99
IF(IPNT.NE.0) GOTO 1
99 RETURN
END
SUBROUTINE FIT(TIME,X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD,IPNT)
C
C ROUTINE TO SMOOTH LASER TRACKER DATA
C
DOUBLE PRECISION CX(10),CY(10),CZ(10)
DIMENSION X(100),XD(100),XDD(100),Y(100),YD(100),YDD(100)
DIMENSION Z(100),ZD(100),ZDD(100)
DOUBLE PRECISION TIME(100)
C
600 WRITE(6,600)IPNT,(X(I),Y(I),Z(I),I=1,IPNT)
600 FORMAT(3X,15,/,100(X,3(E13.7,5X),/,))
IPNT1=IPNT-5
DO 100 I=1,IPNT1
IF(I.GT.7) GOTO 10
IF(I.GT.1) GOTO 100
CALL CRVFI(TIME(I),X(I),3,9,CX)
CALL CRVFI(TIME(I),Y(I),3,9,CY)
CALL CRVFI(TIME(I),Z(I),3,9,CZ)
DO 50 J=1,7
CALL FUNCI(TIME(J),X(J),XD(J),XDD(J),CX)
CALL FUNCI(TIME(J),Y(J),YD(J),YDD(J),CY)
CALL FUNCI(TIME(J),Z(J),ZD(J),ZDD(J),CZ)
50 CONTINUE
GOTO 100
10 CONTINUE
CALL CRVFI(TIME(I-5),X(I-5),3,9,CX)
CALL CRVFI(TIME(I-5),Y(I-5),3,9,CY)
CALL CRVFI(TIME(I-5),Z(I-5),3,9,CZ)
CALL FUNCI(TIME(I),X(I),XD(I),XDD(I),CX)
CALL FUNCI(TIME(I),Y(I),YD(I),YDD(I),CY)
CALL FUNCI(TIME(I),Z(I),ZD(I),ZDD(I),CZ)
100 CONTINUE
IPNT1=IPNT-5
DO 110 I=IPNT1,IPNT

```

```

      CALL FUNC1(TIME(I),X(I),XD(I),XDD(I),CX)
      CALL FUNC1(TIME(I),Y(I),YD(I),YDD(I),CY)
      CALL FUNC1(TIME(I),Z(I),ZD(I),ZDD(I),CZ)
110  CONTINUE
      RETURN
      END
      SUBROUTINE CRVFT(X,Y,M,N,C)
C
C   LEAST-SQUARES POLYNOMIAL CURVE FITTING ROUTINE
C   SOLVES FOR COEFFICIENTS C(I) GIVEN X-Y PAIRS
C   OF DATA POINTS FOR EQUATIONS OF THE FORM:
C       Y=C(1)+C(2)*X+C(3)*X**2+...+C(M+1)*X**M
C
C   C = COEFFICIENT ARRAY
C   X = INDEPENDANT VARIABLE
C   Y = DEPENDANT VARIABLE
C   M = ORDER OF POLYNOMIAL (10 MAX)
C   N = NO. X-Y PAIRS
C
C
      DIMENSION X(10),Y(10)
      DOUBLE PRECISION A(11,11),B(11),P(20),YY(9),C(10)
      DOUBLE PRECISION FACTOR,SUM,TEMP,X
C
C
C       WRITE(6,661)M,N
661  FORMAT(X,16,5X,16)
      DO 1 I=1,9
C       WRITE(6,666)X(I),Y(I),C(I)
666  FORMAT(X,3(E13.7,5X))
      1  YY(I)=Y(I)
      NN=M+1
      DO 5 I=1,NN
      5  C(I)=0.
C       COMPUTE 'P' ARRAY (POWERS X(I) )
      MX2=M*2
      DO 13 I=1,MX2
      P(I)=0.
      DO 13 J=1,N
      13 P(I)=P(I)+X(J)**I
C       DEVELOP CONSTANT TERMS OF NORMAL EQNS.
      L=M+1
      DO 30 I=1,L
      DO 30 J=1,L
      K=I+J-2
      IF(K)29,29,28
      28 A(I,J)=P(K)
      GOTO 30
      29 A(I,1)=N
      30 CONTINUE
      B(1)=0.
      DO 21 J=1,N
      21 B(1)=B(1)+YY(J)
      DO 22 I=2,L
      B(I) = 0.
      DO 22 J=1,N
      22 B(I)=B(I)+YY(J)*X(J)**(I-1)
C       PIVOTAL CONDENSATION
      NM1=L-1
      DO 300 K=1,NM1
      KP1=K+1
      NK2=K
      DO 400 I=KP1,L

```

```

      IF(DABS(A(I,K))-DABS(A(MX2,K))) 400,400,401
401  MX2=I
400  CONTINUE
      IF(MX2-K)500,500,405
405  DO 410 J=K,L
      TEMP=A(K,J)
      A(K,J)=A(MX2,J)
410  A(MX2,J)=TEMP
      TEMP=B(K)
      B(K)=B(MX2)
      B(MX2)=TEMP
C     ELIMINATION AND BACK SOLUTION
450  DO 300 I=KP1,L
      FACTOR=A(I,K)/A(K,K)
      A(I,K)=0.
      DO 301 J=KP1,L
301  A(I,J)=A(I,J)-FACTOR*A(K,J)
300  B(I)=B(I)-FACTOR*B(K)
      C(L)=B(L)/A(L,L)
      I=NM1
710  IP1=I+1
      SUM=0.
      DO 700 J=IP1,L
700  SUM=SUM+A(I,J)*C(J)
      C(I)=(B(I)-SUM)/A(I,I)
      I=I-1
      IF(I) 800,800,710
800  CONTINUE
      RETURN
      END
      SUBROUTINE FUNCT(TIME,X,XD,XDD,C)
      DOUBLE PRECISION C(4)
      DOUBLE PRECISION TIME
C     WRITE(6,100)(C(K),K=1,4)
100  FORMAT(15X,4(B13.7,3X),'***')
C     X=C(1) +C(2)*TIME +C(3)*TIME**2 +C(4)*TIME**3
      X=((C(4)*TIME+C(3))*TIME+C(2))*TIME+C(1)
C     XD=C(2) +2.*C(3)*TIME +3.*C(4)*TIME**2
      XD=(3.*C(4)*TIME+2.*C(3))*TIME+C(2)
      XDD=2.*C(3) +6.*C(4)*TIME
      RETURN
      END
      SUBROUTINE XYZCV(X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD)
C
C
C     ROUTINE TO CONVERT TO RANGE CO-ORDINATE SYSTEM
C
C     XLT,YLT,ZLT = TRACKER CO-ORDS.
C     XL,YL,ZL = LAUNCHER CO-ORDS.
C
C
COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
X     LIM(32),IROV(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),
X     XL,YL,ZL,IH1,IM1,SEC1
C
C     DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
      INTEGER*4 IH,IM,IH1,IM1
C
C     TRACKER CO-ORDS.
C
      XLT = 1072.125
      YLT = -2359.947

```

```

ZLT = 23.885
C*****
C TRY RANGE COORDS FOR LAUNCHER (XL,YL,ZL)
XL=0.
YL=0.
ZL=0.
C*****
C
C COMPUTE POS. OF ROUND (X,Y,Z)
C
C WRITE(6,600)X,Y,Z
C 600 FORMAT(3(3X,E14.7))
X=X*.01745329
Y=Y*.01754329
XD=XD*.01754329
YD=YD*.01754329
XDD=XDD*.01754329
YDD=YDD*.01754329
XM=XLT-XL+Z* COS(Y)* COS(X)
YM=YL-YL+Z* COS(Y)* SIN(X)
ZM=ZLT-ZL+Z* SIN(Y)
C*****START THEIRS*****
C
C DETERMINE VELOCITIES
C
C VR=ZD
VAZ=Z*XD* COS(Y)
VEL=Z*YD
C
C VELO IN LAUNCHER-CENTERED SYS
C
C VX=VR* COS(X)* COS(Y)-VAZ* SIN(X)-VEL* COS(X)* SIN(Y)
C VY=VR* SIN(X)* COS(Y)+VAZ* COS(X)-VEL* SIN(X)* SIN(Y)
C VZ=VR* SIN(Y)+VEL* COS(Y)
C
C COMPUTE ACCELS.
C
C AR=ZDD-Z*YD**2-Z*XD**2*( COS(Y))**2
AAZ=(Z*XDD+2*XD*ZD)* COS(Y)-
X 2.*Z*XD* SIN(Y)*YD
AEL=Z*YDD+2.*YD*ZD+Z* SIN(Y)* COS(Y)*XD**2
C
C CONVERT TO LAUNCHER CO-ORDS.
C
C AX=AR* COS(X)* COS(Y)-AAZ* SIN(X)-AEL* COS(X)* SIN(Y)
C AY=AR* SIN(X)* COS(Y)+AAZ* COS(X)-AEL* SIN(X)* SIN(Y)
C AZ1=AR* SIN(Y)+AEL* COS(Y)
X=XM
Y=YM
Z=ZM
XD=VX
YD=VY
ZD=VZ
XDD=AX
YDD=AY
ZDD=AZ1
C*****END THEIRS, START MINE*****
C
C VR=ZD
C VAZ=XD
C VEL=YD
C XD=VR* COS(Y)* COS(X)-Z*VEL* SIN(Y)* COS(X)-Z*VAZ* COS(Y)* SIN(X)
C YD=VR* COS(Y)* SIN(X)+Z*VEL* SIN(Y)* SIN(X)+Z*VAZ* COS(Y)* COS(X)
C ZD=VR* SIN(Y)+Z*VEL* COS(Y)

```

```

C      AR=ZDD
C      AAZ=XDD
C      AEL=YDD
C      XDD=(AR-Z*(VEL**2+VAZ**2))*COS(Y)*COS(X)
C      X      +(AR-2.*VR*VEL)*SIN(Y)*COS(X)
C      X      +(Z*AAZ-2.*VR*VAZ)*COS(Y)*SIN(X)
C      X      -2.*Z*VEL*VAZ*SIN(Y)*SIN(X)
C      YDD=(AR-Z*VAZ**2)*COS(Y)*SIN(X)
C      X      -(VR*VEL+2.*Z*VEL*VAZ)*SIN(Y)*COS(X)
C      X      +(2.*VR*VAZ-Z*(VEL**2-AAZ))*COS(Y)*COS(X)
C      X      -(VR*VEL+Z*AEL)*SIN(Y)*SIN(X)
C      ZDD=(AR-Z*VEL**2)*SIN(Y)+(2.*VR*VEL+Z*AEL)*COS(Y)
C      X=XM
C      Y=YM
C      Z=ZM
C*****END MINE*****
C      WRITE(6,601)X,Y,Z
C 601 FORMAT(3(3X,E14.7),/)
      RETURN
      END
      SUBROUTINE LABEL(LP,IPAGE)
C
C
C      ROUTINE TO PRINT DATA PAGE LABEL & PG. #
C
C
      NPAGE = IPAGE -1
      WRITE(LP,100)NPAGE
100  FORMAT(1H1,52X,'LAMPAMS - LASER TRACKER',19X,'PAGE: ',I4,/,
X     3X,'TIME',17X,'POSITION',24X,'VELOCITY',23X,'ACCELERATION',
X     15X,'STATUS',/,
X     3X,'(SEC)',19X,'(FT)',26X,'(FT/SEC)',23X,'(FT/SEC-SEC)',/,
X     17X,'X',10X,'Y',10X,'Z',10X,'X',10X,'Y',10X,'Z',10X,
X     'X',10X,'Y',10X,'Z',5X,'LH OV SP OV CP AT',/,
X     3X,'---- ',9(8X,'---'),3X,6(' --'),/)
      RETURN
      END
$BEND

```

US ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND
 TECHNOLOGY LABORATORY
 SYSTEMS SIMULATION DIRECTORATE
 SYSTEMS EVALUATION

LAMPAMS - LASER TRACKER

TIME (SEC)	POSITION (FT)			VELOCITY (FT/SEC)		
	X	Y	Z	X	Y	Z
----	---	---	---	---	---	---
0.250	1067.57	259.27	25.45	58.11	-47.44	43.47
0.260	1068.51	258.56	26.08	36.92	-24.50	20.18
0.270	1069.06	258.26	26.30	19.13	-6.59	2.82
0.280	1069.29	258.26	26.23	4.72	6.28	-8.54
0.290	1069.27	258.47	25.99	-6.32	14.10	-14.19
0.300	1069.06	258.79	25.70	-13.99	16.87	-13.83
0.310	1068.74	259.12	25.48	-18.30	14.58	-7.57
0.320	1068.27	259.51	25.45	-6.47	5.34	5.12
0.330	1068.36	259.30	25.76	6.08	-9.46	14.32
0.340	1068.76	258.94	26.22	10.66	-7.66	15.49
0.350	1069.09	258.66	26.54	6.14	-4.14	7.96
0.360	1069.11	258.84	26.55	-2.11	-1.78	-3.58
0.370	1068.85	258.85	26.23	-7.17	1.68	-10.53
0.380	1068.53	258.97	25.86	-6.05	4.54	-9.86
0.390	1068.46	259.06	25.72	0.69	4.05	-3.85
0.400	1068.66	259.07	25.82	7.56	11.49	1.68
0.410	1068.98	259.37	26.00	9.91	-5.46	4.25
0.420	1069.22	259.23	26.11	5.00	4.20	2.54
0.430	1069.20	259.08	26.10	-4.05	-7.88	-3.04
0.440	1068.92	259.15	25.90	-11.38	2.78	-7.25
0.450	1068.55	258.95	25.66	-11.85	11.33	-7.16
0.460	1068.32	259.64	25.54	-5.03	2.64	-3.51
0.470	1068.37	259.59	25.57	4.51	-1.34	0.50
0.480	1068.66	259.34	25.67	12.40	-13.07	3.18

Note that target acceleration and LAMPAMS status report have not been presented but are normally included in the output.

1)) *40
1)) *20
1)) *10
1)) *8
31)) *4
41)) *2
31)) *1

3), -31)) *40
4), -31)) *20
5), -31)) *10
6), -31)) *8
7), -31)) *4
8), -31)) *2
9), -31)) *1

31)) *800
31)) *400
31)) *200
31)) *100
31)) *80
31)) *40
31)) *20
31)) *10
31)) *6
31)) *4
31)) *2