NASA TECHNICAL MEMORANDUM

NASA TM X- 62,386

(NASA-TM-X-62386) USER'S MANUAL FOR A PARAMETER IDENTIFICATION TECHNIQUE (NASA) 200 p HC $7.00 CSCL 09B

N75-25623

Unclas

G3/61 25352

NASA TM X- 62,386

USERS' MANUAL FOR A PARAMETER IDENTIFICATION TECHNIQUE

Gerd Kanning

Ames Research Center
Moffett Field, Calif. 94035

March 1975

1. Report No.
TM X-62,386

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
USERS' MANUAL FOR A PARAMETER IDENTIFICATION TECHNIQUE

5. Report Date

6. Performing Organization Code

7. Author(s)
Gerd Kanning

8. Performing Organization Report No.
A-5752

9. Performing Organization Name and Address
NASA Ames Research Center
Moffett Field, Calif. 94035

10. Work Unit No.
760-63-01

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A digital computer program written in Fortran is presented that implements the system identification theory for deterministic systems using input-output measurements. The user supplies programs simulating the mathematical model of the physical plant whose parameters are to be identified. The user may choose any one of three options. The first option allows for a complete model simulation for fixed input forcing functions. The second option identifies up to 36 parameters of the model from wind tunnel or flight measurements. The third option performs a sensitivity analysis for up to 36 parameters. The use of each option is illustrated with an example using input-output measurements for a helicopter rotor tested in a wind tunnel.

17. Key Words (Suggested by Author(s))
Parameter identification
Helicopter parameter identification
Digital computer program for identification
Aircraft parameter identification

18. Distribution Statement
Unclassified - Unlimited

STAR Category - 02, 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	198	$7.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

USERS' MANUAL FOR A PARAMETER IDENTIFIACTION TECHNIQUE

Gerd Kanning

Ames Research Center

SUMMARY

A digital computer program written in Fortran is presented that implements the system identification theory for deterministic systems using input-output measurements. The program contains an algorithm that detects the existance of nonuniqueness of the parameters and reduces the parameter set until a unique set of parameters is obtained. The dynamic model of the plant whose parameters are to be identified can be nonlinear, time varying and periodic.

The programs were categorized into user supplied programs, user modified programs internal programs and libra: programs. To apply the identification programs the user makes the necessary changes in each category to simulate the model behavior. The user has the capability to specify any one of three options. The first option allows for a complete model simulation for fixed input forcing functions. The second option identifies up to 36 parameters of the model from wind tunnel or flight measurements. The third option calculates a sensitivity analysis up to 36 parameters. An example using input-output measurements for a helicopter rotor tested in a wind tunnel is given for each option.

INTRODUCTION

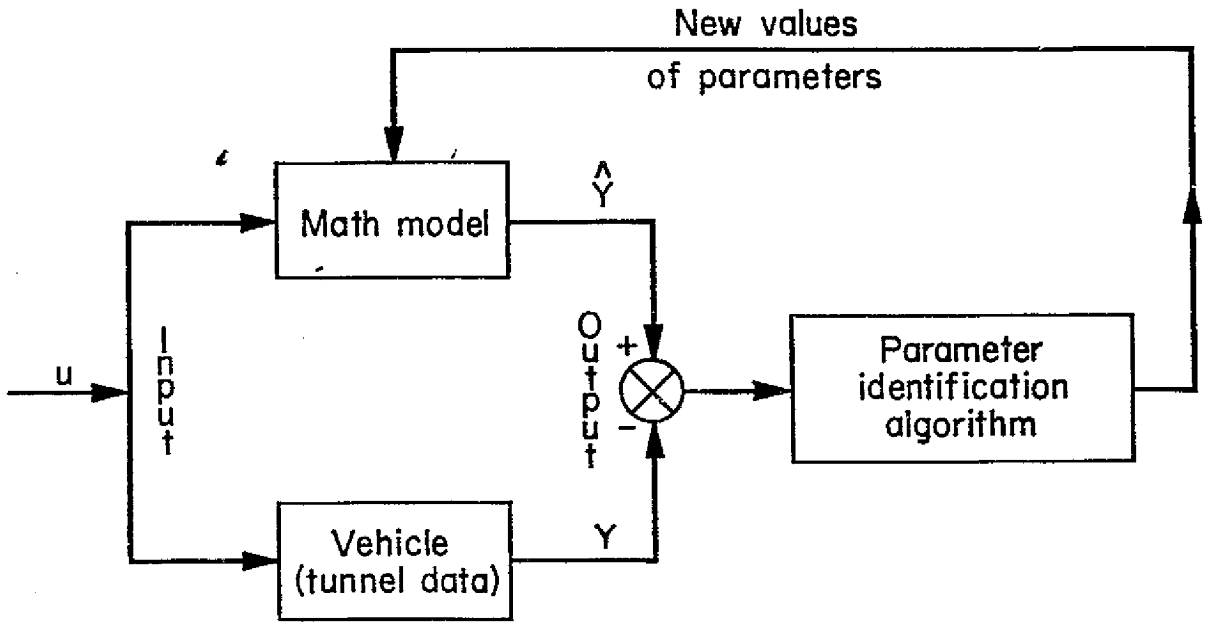
Wind tunnel and flight data have been extensively used for estimating stability and control derivatives for airplanes and more recently also for helicopters. Previously, computer programs used for the estimation have been specialized for each particular application. In this report a computer program is presented which is applicable to a wide range of systems with only minor program modifications. The program has the novel feature of calculating the uniqueness of the parameters estimated. This can be of crucial importance since input-output measurements often lead to nonunique solutions for the parameter values. Nonuniqueness may be due to parameter redundancy or insufficient information in the measurements and computations. An algorithm was presented in reference 1 that detects the existance of nonuniqueness and also determines a unique set of parameters. This report makes available the digital computer programs used in the above reference and gives a description of the programs implementing the system identification theory.

The dynamic model of the plant must be obtained from the analysis of the physical process occurring in the plant. The model can be nonlinear and time varying and must be written in the form $\dot{\bar{x}} = f(\bar{x},t,\bar{p})$. The identification program estimates the parameter vector $\bar{p}$ from input-output measurements.

The digital computer programs were written in Fortran and were programmed on an IBM 360/67 computer. The programs were catagorized into user supplied programs, user modified programs, internal programs and library programs. The user makes the necessary changes in each catagory to simulate the model behavior. The user has the capability to specify any one of three options. The first option allows for a complete model simulation for fixed input forcing functions. The second option identifies up to 36 parameters of the model from wind tunnel or flight measurements. The third option calculates a sensitivity analysis for up to 36 parameters. Each option is specified from external data cards and the results are displayed on an on-line printer.

PROGRAM DESCRIPTION

In reference 1 a theory is presented for identification of nonlinear systems in the presence of nonuniqueness. In this report a description of the computer program is given that represents an implementation of the theory. A brief review of the parameter identification process is first given. Sketch A shows a general outline of the identification process. The same input u is applied to both the math model and the plant ("plant" refers to the physical system from which measured time histories are available and whose parameters are to be estimated). The best values of the parameters for the math model are those that minimize the difference between the model and plant outputs. The identification process is an algorithm that compares the outputs of the math model and plant and adjusts the parameter values (such as inertia, spring constant, damping constant, etc.) until this difference is a minimum. What this minimum should be is established by least squares fit and the identification algorithm iterates until this criterion is reached. It may not always be possible to reach this minimum

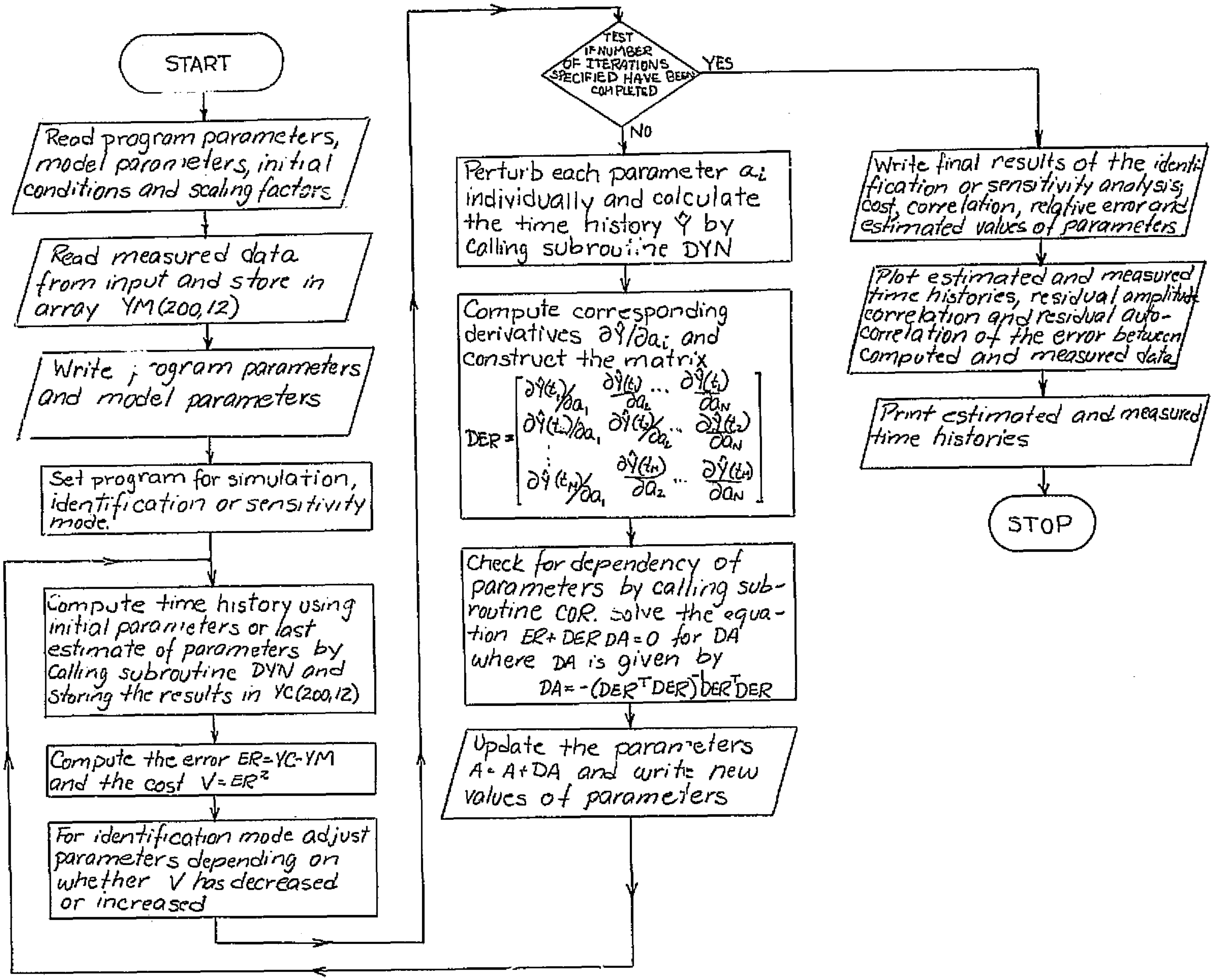


Sketch A.- Block diagram of parameter identification technique.

or even to obtain an improvement in the matching of the model and plant outputs. Problems encountered with the technique and some of their solutions are also given in reference 1. This report deals only with the description of the computer programs implementing the theory of reference 1. A flow diagram of the parameter identification technique as implemented on the digital computer is given in sketch B. This diagram is a simplified sketch of the identification algorithm and only gives those operations that are central to the technique. The equations given in the diagram follow the development of the identification technique as discussed in reference 1.

The differential equations describing the model can be nonlinear and time varying of the form $\dot{x} = f(\bar{x},t)$. The differential equations for the plant should be written in first order form to be easily compatable with the computer programs presented. It should be noted here that the system equations do not necessarily have to be differential but can be of polynomial form. If differential equations are not used to describe the model, then slight modifications in some of the subroutines presented are required. These modifications are pointed out in the description of the individual subroutines.

The order of the differential equations that the programs can handle is not restricted. The identification programs are limited, however, by the total number of parameters to be estimated and the total number of measurements and data points per measurement channel



Sketch B.- Flow diagram for parameter identification technique.

which are used. All of these limitations can be removed if the necessary changes are made in the program. The program as discussed is limited to estimating a maximum of 36 parameters using a maximum of 12 measurements and 200 data points for each measurement. A smaller number of parameters, measurements and data points can be used without altering the basic parameter identification program. Examples for estimating smaller number of parameters are given in a later section.

The capability of the identification program is divided into three modes of operation. Each mode is specified by use of input data cards as is illustrated in the example section. First, the program is run in a simulation mode. In this mode of operation the program calculates the system response to specified forcing functions and system parameters. The results are stored on disk from which a punched card deck is obtained or the results are printed or plotted using the on-line printer. Second, the program is operated in the identification mode. In this mode of operation, measurements from the plant (max. of 12 measurements and 200 data points per measurement) are used to estimate specified parameters (max. of 36) of the model. The identification program is given nominal values of the parameters to start the iteration. It iterates until a best match between the measured response and the model response is obtained. At the end of the iteration sequence the program prints the best estimate of the parameters which produced the smallest error between the measured and calculated response. The measured and calculated responses can be stored on computer disk, printed or plotted by the on-line printer for visual inspection of the results. The autocorrelation of the measured and computed output can also be obtained in this mode of operation. The results can again be stored on computer disk and then printed or plotted. Third, the program is operated in the sensitivity analysis mode. The sensitivity is defined (ref. 1) as a dimensionless ratio of the change in the output due to a change in the parameter. For this mode of operation the user can specify the sensitivity analysis of all the parameters that are to be estimated. This is done in order to expose those parameters that are most sensitive to small variations. The most sensitive parameters are then specified in the identification mode since they are most critical in matching the measured and calculated response. The output for the sensitivity analysis is a table listing of the parameters and their sensitivity. Each of these three options is demonstrated by an example and the meaning of the output results are discussed.

ORGANIZATION OF PROGRAMS

The computer programs are divided into four categories: user supplied programs, user modified programs, internal programs and library programs. Each catagory will be discussed such that the user can apply these programs to his particular problem. Table 1 gives a list of all the programs required for the identification of the system parameters. The type of program as catagorized above is also indicated in Table 1. The user supplied and the user modified programs will be discussed in detail in the following pages, while the discussion of the internal and library programs is given in appendix A. The program discussion will be divided into a section on program "Description" and a section on program "Usage".

In the "Description" section a general overview of the individual function of the program is given while in the "Usage" section the function of each variable in the calling list and variables in the common statements pertinent to the particular program are given.

TABLE 1. LIST OF IDENTIFICATION PROGRAMS

			BYTES OF STORAGE
1.	MAIN2$$	(user supplied program)	5032
2.	NEWP$$	" " "	4404
3.	DERIV$$	" " "	3764
4.	OUTPUT$$	" " "	1792
5.	PRINT1$$	" " "	1564
6.	IDENT1$$	(user modified program)	3436
7.	INOUT$$	" " "	13224
8.	DYN$$	" " "	1532
9.	READIN$$	(internal program)	3652
10.	PARAM$$	" "	5600
11.	ADJUST$$	" "	1856
12.	DYDA$$	" "	11688
13.	COR$$	" "	13704
14.	TRADUC$$	" "	556
15.	ERROR$$	" "	2536
16.	CRUNCH$$	" "	13892
17.	PLOTIN$$	" "	3428
18.	INTS	(library program)	
19.	INTM	" "	
20.	SETTIM	" "	

USER SUPPLIED AND USER MODIFIED PROGRAMS

1. MAIN2$$

Description

MAIN2$$ is the main line program that will call the identification program. The parameters of the mathematical model are read into the program and the same information is printed. Following the execution of these statements, the initial conditions required for the differential equations of the model are defined in the T array (see description of integration routine). For the example programmed here the A matrix defined by the differential equation of the model $\dot{\bar{x}} = A(t)\bar{x} + B(t)\bar{u}$ is initialized using the system parameters. After the initialization of the system equations the subroutine IDENT1 is called. This subroutine calls all the remaining subroutines until the identification of the specified parameters is completed.

Usage

CALL MAIN2$$

At the beginning of the program the common /MAIN1/ must appear with the T array whose size will depend on the order of the dynamic model to be integrated (see description of internal subroutine in appendix A). The remaining variables in common are used in modeling the system dynamics for the helicopter rotor example illustrated. The user can create new commons to describe his particular mathematical model. These common statements should only be present in those subroutines the user must supply (see list of user supplied subroutines). At the end of the MAIN2$$ program the user calls the subroutine IDENT1 which in turn calls the identification programs.

2. NEWP$$

Description

The NEWP subroutine is supplied by the user. The subroutine lists the parameters of the model that are to be identified. The list of parameters given for this particular example are shown at the beginning of the program where they are defined by the P array with the left side of the equation representing the variable name. The identification program adjusts the P's until the best match between the calculated and measured data is reached. Therefore all coefficients in the model that are effected by the P's must be recalculated every time the subroutine is called. Hence the remaining part of the program calculates the coefficients of the mathematical model that are effected as the parameters P change. The coefficients listed in the program were previously initialized in the main program but they must now be updated as the P's are varied.

Usage

CALL NEWP

There are no arguments in the call of the subroutine. The labeled commons /MAIN1/, /DERIV1/, /SENTIV/,/OUTNEW/,/INFLOW/, and /REVSF/ were supplied by the user to define elements of the model equations. The labeled common /RPM/ amd /COM1/ are associated with the identification programs and must be present in the program exactly as shown. The P array defines the parameters to be identified and the N1 variable defines the total number of parameters that can be identified (max. N1=36). Following the common statements, the Fortran statement IF(N1.EQ.O) RETURN must follow in order to by pass the program if no parameters are identified. The next statement must be the computed GØ TØ statement

```
      GØ TØ (1,2,3...36),N1

 36
 35
  .   .   .
  .   .   .
  .   .   .
  2   BLOSS=P(2)
  1   BIY=P(1)
```

If N1 is less than 36, for example N1=21, then the computed GØ TØ and the list of parameters must be defined for 21 parameters. (See example). Note that the variables BIY, BLOSS etc. are variable names the user assigns to the parameters to be identified. The user is free to choose these variable names as long as they are not in conflict with the variables defined in common /RPM/ and /COM1/.

3. DERIV$$

Description

The subroutine is supplied by the user and must be compatible with the library integration routine used to integrate the equations of motion. For the model simulated in this example, the DERIV subroutine evaluates the derivatives of the dynamical equations that are written in the form $\dot{\bar{x}}=A\bar{x}+B\bar{u}$. If the equations are not differential and some other means are used to obtain the calculated behavior of the model, the subroutine is not needed. The model response is then entirely computed in the DYN subroutine (see description of DYN).

Usage

There are no arguments in the calling list. All common statements except the common /STATE/ are supplied by the user to generate the model response. The common /STATE/ must be present in the program as shown. The only variable used from the common statement are defined by the arrays YM(200,12) and SCALE(12) (see READIN subroutine). The YM array contains the measured response and the forcing functions of the system. The SCALE (12) contains the scaling factors required to convert the data in YM to engineering units. For the model programmed in DERIV, channels 7 and 8 contained the forcing functions of the system. These forcing functions are needed to force the mathematical model in the exact same manner as they were used to force the actual system.

4. OUTPUT$$

Description

The subroutine OUTPUT is supplied by the user. The subroutine is called by the DYN subroutine whenever a calculated data point is to be stored. The program calculates the output matrix $\bar{y}=C\bar{x}$ from the state variables $\bar{x}$. The results of $\bar{y}$ are stored in the array YC(200,12). The error between the calculated (YC) and the measured response (YM) is stored in the array ER(2400). Finally the calculated response YC is weighted by the weighting matrix WEIGHT(12) and the result is stored in YW(2400).

Usage

CALL OUTPUT

Three common statements that come from the identification programs must be present in the subroutine. These commons are /STATE/, /RPM/ and /WGHT/. The variables used in the OUTPUT subroutine from these commons are ND, NS, SCALE(12) which are defined as:

ND number of data points in each measured output (read from input data in the INOUT subroutine)

NS number of measured outputs (read from input data in the INOUT subroutine)

SCALE(12) scaling factors for 12 channels that convert data from integer values to engineering units or back into integer values (read from input data cards in the INOUT subroutine).

YC(200,12) the computed response calculated in the OUTPUT subroutine (maximum of 12 time histories containing a maximum of 200 data points each).

YM(200,12) the measured array brought into the OUTPUT subroutine (read from data cards or computer disk by the READIN subroutine).

YW(2400) the computed array YC(200,12) weighted by the weighting matrix WEIGHT(12) calculated in subroutine OUTPUT.

ER(2400) the error between the calculated and measured response obtained in the OUTPUT subroutine.

The remaining common statements and variables are unique to this particular model simulation. The user must supply the required commons to generate the YC(200,12) and YM(200,12) respectively. The measured and computed data is converted to engineering units with the scale factor SCALE(12) before printing. The array PSI1(200) contains the angle measurement corresponding to each measured and computed data point. Other variables of interest to the user may be transferred to the subroutine for printing.

Usage

CALL PRINT1

The PRINT1 subroutine has no arguments in the calling list. All variables are brought into the program through the common /STATE/ and /PRIN/. The variables used in the program for the common /STATE/ are ND, NS, SCALE(12), YC(200,12) and YM(200,12) which have been defined in the discussion of the previous subroutine. All variables in the common /PRIN/ are printed out with the on-line printer. The only variable of particular interest is the array PSI1(200) which contains the angle measurement corresponding to each measured and computed data point stored in the matrics YC(200,12) and YM(200,12). The remaining variables are unique to the particular dynamic system modeled. The user may change the /PRIN/ common in order to transfer variables into the program which he would like to print.

6. IDENT1$$

Description

The subroutine calls a number of subroutines that carry out the identification process. At the beginning of the program the initial per cent variation for each parameter is set to 1% and the weight for all 12 measurements are set to 1. If different weights are to be assigned, they can be read into the program through input data cards (see example). The initial conditions that are read into the main program and stored in the T array are redefined in the matrix XINIR(J). The remaining part of the program primarily deals with calling the input-output subroutine INOUT and the parameter identification routine PARAM. The subroutine ERROR called in IDENT1 calculates the relative error, the correlation, the autocorrelation of the error and variance of the parameters. Three other subroutines called by the program are PRINT1, CRUNCH and PLOTIN. The PRINT1 subroutine prints the results of the identification. The CRUNCH subroutine stores the data while the PLOTIN subroutine plots the data on the "on-line" printer. Any one or all of these options may be specified to be executed by the program from the input data cards (see example of data input).

Usage

CALL IDENT1

The subroutine has no arguments in the calling list of the program. All common statements must remain unchanged except for the common /MAIN1/. This common is defined in the main program and contains a number of variables that define the parameters of the system. The T matrix must be present in this common as has been defined in the main program. The variable form common /MAIN1/ used in the subroutine and its meaning is:

T(123) an array required by the integration routine AL INTS (see description of library subroutines used in the program).

Note that the remaining variables in this common statement are given dummy names since they are not needed in the subroutine. The user must store in the XINIT(J) array the initial conditions of the model that are defined in the T array (see lines 2900 and 3300 of subroutine IDENT1 in appendix B). No further changes in the remaining part of the program are to be made.

7. INOUT$$

Description

The subroutine is called by the IDENT1 subroutine and its primary purpose is to read and write parameters of the model that are to be identified. The measured data is read into the program by the use of the subroutine READIN and is stored in the matrix YM(200,12). The data can either be read from cards or from a computer disk as will be described later in the description of the READIN subroutine. The measured and calculated data is scaled by the scaling factor SCALE(J). This is done to compress the measured data on cards in order to minimize the number of data cards. Further discussion of the format used in reading the measured data cards will be given in the description of the subroutine READIN.

Several options that the user may specify with input data cards are also carried out in the program. (See example for data input). The option for simulating the equations of motion with no identification is performed in the program between lines 24800 and 26400 (see listing of subroutine in appendix B). Control from the program is returned to the IDENT1 subroutine unless an error in reading the data cards is encountered upon which the program stops.

Usage

CALL INOUT(JJ)

The argument in the call is either 1 or 2. For JJ=1 the input cards are read and written out. For JJ=2 the identification has been completed and statements form lines 23100 and 24100 are executed in the INOUT subroutine. The common statement /MAIN1/

are:

T(123) an array required by the integration routine AL INTS, and OMEGA the angular velocity of rotor.

The angle increment in degrees and the equivalant time in seconds between measured data points must be calculated in the program. The step size (KDT in millidegree) between measurements is brought into the program from the READIN subroutine. The increment in degrees (DPSI) and the equivalant time interval (DT) are then calculated between lines 10600 and 10800 of the program.

8. DYN$$

Description

The subroutine DYN is used to call the integration routines that integrate the dynamical equations of motion for the mathematical model. The first program called in the subroutine is the subroutine NEWP. The NEWP subroutine initializes the parameters in the mathematical model that are to be identified. Next the integration routine CALL INTS and CALL INTM are called. These two routines are standard library integration programs that integrate first order differential equations of the form $\dot{\bar{x}}=f(\bar{x},t,)$ where $\dot{\bar{x}}$, and $\bar{x}$ are n-th order vectors and the prime indicates differentiation with respect to the independent variable t. These two routines also require a user supplied subroutine named DERIV. The subroutine DERIV evaluates the derivatives and stores them in the array T(4+N) where N is the number of differential equations. The T array consists of 12N+3 cells if Adams-Moulton option is used or 4N+3 cells if the Runge-Kutta option is used. One additional subroutine supplied by the user and called in DYN is the subroutine OUTPUT. This subroutine is called each time a calculated data point obtained by the integration routine is required to be stored. The complete time history for the mathematical model is thus generated.

If integration of the equations is not necessary then whatever static equations governing the model behavior must be programmed in the DYN subroutine. The integration subroutines and the subroutine DERIV are not required. Although the OUTPUT subroutine must be supplied by the user and the calculated model response must be stored in the appropriate matrices.

Usage

CALL DYN (N1, KT, KMIX)

The arguments in the calling list are:

N1 number of parameters (max 36; this parameter is not used in the program)

KT must be set to 1 in the calling program. KT is used in the program to initialize T(2) and also acts as a counter to keep track of the number of times a calculated data point is stored.

KMIX not used.

The labeled common /STATE/ must be present in the program. This common is used to transfer variables between the identification programs. The labeled common /MAIN1/ and /OUTNEW/ are supplied by the user to obtain the time histories for the model equations programmed. If the integration program is used, the name of the derivative subroutine must appear on an EXTERNAL card at the beginning of the DYN subroutine.

The remaining programs that are given in table 1 will be discussed in appendix A. These programs are not changed by the user unless he is acquainted with the functions of the programs beyond the brief description given in the appendix.

SYSTEM DEPENDENT FEATURES

The only system dependent program in the list of table 1 is the subroutine PLOTIN. This subroutine requires a printer with 125 characters per line and 45 lines per page.

EXAMPLES USING THE IDENTIFICATION PROGRAM

Three examples are given to illustrate the application of the programs to a system of the form $\dot{\bar{x}}=f(\bar{x},t)$. Each example illustrates one unique feature of the program. The mathematical model that is used throughout these examples is developed in reference 2. The mathematical model represents the flapping and feathering dynamics of a three bladed hingeless helicopter rotor in forward flight. The equations of the model are linear but time varying and periodic due to the forward speed of the rotor. The complete equations written in state space form are shown in table 2. These equations are used to illustrate the use of the identification program.

The three modes of operation for the identification program that can be specified by external data cards are simulation, identification and sensitivity. Each of these options is demonstrated below by use of an example. Regardless of which option is used, the user must program the mathematical model equations that simulate the dynamic behavior of the system. The changes required in the computer programs to incorporate the users mathematical model are outlined in the section titled Organization of Programs.

EXAMPLE 1: Simulation of Equations

With the mathematical model incorporated in the identification program, the simulation option of the program can be used. The main program (MAIN2$$) and the subroutine DERIV and NEWP (NEWP is not used for the simulation option but the subroutine must be present

$\dot{x}_1$	0	1	0	0	0	
$\dot{x}_2$	$\frac{-(I_3k_R-I_G)\Omega^2-\frac{3}{2}C_2^2(I_z-I_y)\Omega^2-\frac{3}{2}k_\beta C_2^2-K_s}{I_{GE}}$	$\frac{-\frac{3}{2}c_\beta C_2^2-C_s}{I_{GE}}$	$\frac{\Omega C_s}{I_{GE}}$	$\frac{-(I_3k_R-2I_G)\Omega}{I_{GE}}$	$\frac{C_2\zeta_o k_\beta+C_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}}$	$-C$ …
$\dot{x}_3$	0	0	0	1	0	
$\dot{x}_4$	$\frac{-\Omega C_s}{I_{GE}}$	$\frac{(I_3k_R-2I_G)\Omega}{I_{GE}}$	$\frac{-(I_3k_G-I_G)\Omega^2-\frac{3}{2}C_2^2(I_z-I_y)\Omega^2-\frac{3}{2}k_\beta C_2^2-K_s}{I_{GE}}$	$\frac{-\frac{3}{2}c_\beta C_2^2-C_s}{I_{GE}}$	0	
$\dot{x}_5$	0	0	0	0	0	
$\dot{x}_6$	$\frac{1}{2}C_2\gamma\Omega^2\left[B4+2\mu(\zeta_o\cos\psi+\sin\psi)B3+\mu^2\left(\frac{1}{2}-\frac{1}{2}\cos 2\psi+\zeta_o\sin 2\psi\right)B2\right]+\frac{C_2(I_z-I_y)\zeta_o\Omega^2}{I_y}$	$\frac{C_2(I_x+I_y-I_z)\Omega}{I_y}$	0	0	$-\frac{1}{2}\gamma\Omega^2[\zeta_o B4\mu(\cos\psi+\zeta_o\sin\psi)B3]+\frac{1}{2}\mu^2[\zeta_o+\zeta_o\cos 2\psi+\sin 2\psi)B2]+\frac{1}{2}\gamma\Omega^2\left[B4+2\mu(\zeta_o\cos\psi+\sin\psi)B3+\mu^2\left(\frac{1}{2}-\frac{1}{2}\cos 2\psi+\zeta_o\sin 2\psi\right)B2\right]\tan\delta_3-\frac{k_\beta}{I_y}+\frac{(I_x+I_y\zeta_o^2-I_z)\Omega^2}{I_y}-\frac{\frac{3}{2}e\Omega^2}{R-e}$	$-\frac{1}{2}\gamma\Omega$ … $+\frac{\mu}{\Omega}$ (…
$\dot{x}_7$	0	0	0	0	0	
$\dot{x}_8$	$-\frac{1}{4}C_2\gamma\Omega^2\left\{B4+2\mu[\zeta_o\cos(\psi+120°)+\sin(\psi+120°)]B3+\mu^2\left[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+120°)+\zeta_o\sin 2(\psi+120°)\right]B2\right\}-\frac{\frac{1}{2}C_2(I_z-I_y)\zeta_o\Omega^2}{I_y}$	$\frac{-\frac{1}{2}\Omega C_2(I_x+I_y-I_z)}{I_y}$	$\frac{\sqrt{3}}{4}C_2\gamma\Omega^2\left\{B4+2\mu[\zeta_o\cos(\psi+120°)+\sin(\psi+120°)]B3+\mu^2\left[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+120°)+\zeta_o\sin 2(\psi+120°)\right]B2\right\}+\frac{\frac{\sqrt{3}}{2}C_2(I_z-I_y)\zeta_o\Omega^2}{I_y}$	$\frac{\frac{\sqrt{3}}{2}\Omega C_2(I_x+I_y-I_z)}{I_y}$	0	
$\dot{x}_9$	0	0	0	0	0	
$\dot{x}_{10}$	$-\frac{1}{4}C_2\gamma\Omega^2\left\{B4+2\mu[\zeta_o\cos(\psi+240°)+\sin(\psi+240°)]B3+\mu^2\left[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240°)+\zeta_o\sin 2(\psi+240°)\right]B2\right\}-\frac{\frac{1}{2}C_2(I_z-I_y)\zeta_o\Omega^2}{I_y}$	$\frac{-\frac{1}{2}\Omega C_2(I_x+I_y-I_z)}{I_y}$	$-\frac{\sqrt{3}}{4}C_2\gamma\Omega^2\left\{B4+2\mu[\zeta_o\cos(\psi+240°)+\sin(\psi+240°)]B3+\mu^2\left[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240°)+\zeta_o\sin 2(\psi+240°)\right]B2\right\}-\frac{\frac{\sqrt{3}}{2}C_2(I_z-I_y)\zeta_o\Omega^2}{I_y}$	$\frac{-\frac{\sqrt{3}}{2}\Omega C_2(I_x+I_y-I_z)}{I_y}$	0	

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & +\frac{\sqrt{3}}{2}k_\beta & -\frac{\sqrt{3}}{2}k_\beta \\ -k_\beta & +\frac{1}{2}k_\beta & +\frac{1}{2}k_\beta \end{bmatrix} \begin{bmatrix} x_5 \\ x_7 \\ x_9 \end{bmatrix}$$

ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME

0	0	0	0	0	x_1		
$\frac{(I_x+I_y-I_z)\Omega}{I_{GE}}$	$\frac{-\frac{1}{2}c_2\zeta_o k_\beta-\frac{1}{2}c_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}}$	$\frac{\frac{1}{2}c_2(I_x+I_y-I_z)\Omega}{I_{GE}}$	$\frac{-\frac{1}{2}c_2\zeta_o k_\beta-\frac{1}{2}c_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}}$	$\frac{\frac{1}{2}c_2(I_x+I_y-I_z)\Omega}{I_{GE}}$	x_2		
0	0	0	0	0	x_3		
0	$\frac{\frac{\sqrt{3}}{2}c_2\zeta_o k_\beta+\frac{\sqrt{3}}{2}c_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}}$	$-\frac{\frac{\sqrt{3}}{2}c_2(I_x+I_y-I_z)\Omega}{I_{GE}}$	$\frac{-\frac{\sqrt{3}}{2}c_2\zeta_o k_\beta-\frac{\sqrt{3}}{2}c_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}}$	$\frac{\frac{\sqrt{3}}{2}c_2(I_x+I_y-I_z)\Omega}{I_{GE}}$	x_4		
1	0	0	0	0	x_5		
$\Big[\frac{B4}{\Omega}$ $\cos\psi+\sin\psi)B3\Big]$	0	0	0	0	x_6		$\frac{1}{2}\gamma\Omega^2\Big[$ $+\frac{1}{2}\gamma\Omega$ $+\frac{1}{2}\gamma\Omega$ $+\frac{1}{2}\gamma\Omega$ $+\frac{1}{2}\gamma\Omega$
0	0	1	0	0	x_7		
0	$-\frac{1}{2}\gamma\Omega^2\{\zeta_o B4+\mu[\cos(\psi+120°)+\zeta_o\sin(\psi+120°)]B3$ $+\frac{1}{2}\mu^2[\zeta_o+\zeta_o\cos 2(\psi+120°)+\sin 2(\psi+120°)]B2\}$ $+\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi+120°)+\sin(\psi+120°)]B3$ $+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+120°)+\zeta_o\sin 2(\psi+120°)\Big]B2\Big\}\tan\delta_3$ $-\frac{k_\beta}{I_y}+\frac{(I_x+I_y\zeta_o^2-I_z)\Omega^2}{I_y}-\frac{\frac{3}{2}e\Omega^2}{R-e}$	$-\frac{1}{2}\gamma\Omega^2\Big\{\frac{B4}{\Omega}$ $+\frac{\mu}{\Omega}[\zeta_o\cos(\psi+120°)$ $+\sin(\psi+120°)]B3\Big\}$	0	0	x_8	+	$\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi$ $+\frac{1}{2}\gamma\Omega^2\Big\{B5+2\mu[\zeta_o$ $+\frac{1}{2}\gamma\Omega^2(B3+\mu[\zeta_o\cos$ $+\frac{1}{2}\gamma\Omega^2(B4\cos(\psi+1$ $+\frac{1}{2}\gamma\Omega^2(B4\sin(\psi+1$
0	0	0	0	1	x_9		
0	0	0	$-\frac{1}{2}\gamma\Omega^2\Big\{\zeta_o B4+\mu[\cos(\psi+240°)$ $+\zeta_o\sin(\psi+240°)]B3+\frac{1}{2}\mu^2[\zeta_o+\zeta_o\cos 2(\psi+240°)$ $+\sin 2(\psi+240°)]B2\Big\}$ $+\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi+240°)$ $+\sin(\psi+240°)]B3+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240°)$ $+\zeta_o\sin 2(\psi+240°)\Big]B2\Big\}\tan\delta_3$ $-\frac{k_\beta}{I_y}+\frac{(I_x+I_y\zeta_o^2-I_z)\Omega^2}{I_y}-\frac{\frac{3}{2}e\Omega^2}{R-e}$	$-\frac{1}{2}\gamma\Omega^2\Big\{\frac{B4}{\Omega}$ $+\frac{\mu}{\Omega}[\zeta_o\cos(\psi+240°)$ $+\sin(\psi+240°)]B3\Big\}$	x_{10}		$\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos($ $+\frac{1}{2}\gamma\Omega^2\Big\{B5+2\mu[\zeta_o c$ $+\frac{1}{2}\gamma\Omega^2(B3+\mu[\zeta_o\cos$ $+\frac{1}{2}\gamma\Omega^2(B4\cos(\psi+$ $+\frac{1}{2}\gamma\Omega^2(B4\sin(\psi+$

TABLE 2.- MATHEMATICAL MODEL IN STATE SPACE FORM

$$
\begin{bmatrix}
0 & 0 & 0 \\
\dfrac{\frac{1}{2} C_2(I_x+I_y-I_z)\Omega}{I_{GE}} & \dfrac{-\frac{1}{2} C_2\zeta_o k_\beta - \frac{1}{2} C_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}} & \dfrac{\frac{1}{2} C_2(I_x+I_y-I_z)\Omega}{I_{GE}} \\
0 & 0 & 0 \\
\dfrac{-\frac{\sqrt{3}}{2} C_2(I_x+I_y-I_z)\Omega}{I_{GE}} & \dfrac{-\frac{\sqrt{3}}{2} C_2\zeta_o k_\beta - \frac{\sqrt{3}}{2} C_2(I_z-I_y)\zeta_o\Omega^2}{I_{GE}} & \dfrac{\frac{\sqrt{3}}{2} C_2(I_x+I_y-I_z)\Omega}{I_{GE}} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
\begin{gathered}-\frac{1}{2}\gamma\Omega^2\Big\{\frac{B4}{[illegible]} \\ +\frac{\mu}{\Omega}[\zeta_o\cos(\psi+120^\circ) \\ +\sin(\psi+120^\circ)]B3\Big\}\end{gathered} & 0 & 0 \\
0 & 0 & 1 \\
0 & \begin{gathered}-\frac{1}{2}\gamma\Omega^2\Big\{\zeta_o B4+\mu[\cos(\psi+240^\circ) \\ +\zeta_o\sin(\psi+240^\circ)]B3+\frac{1}{2}\mu^2[\zeta_o+\zeta_o\cos 2(\psi+240^\circ) \\ +\sin 2(\psi+240^\circ)]B2\Big\} \\ +\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi+240^\circ) \\ +\sin(\psi+240^\circ)]B3+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240^\circ) \\ +\zeta_o\sin 2(\psi+240^\circ)\Big]B2\Big\}\tan\delta_3 \\ -\frac{k_\beta}{I_y}+\frac{(I_x+I_y\zeta_o^2-I_z)\Omega^2}{I_y}-\frac{\frac{3}{2}e\Omega^2}{R-e}\end{gathered} & \begin{gathered}-\frac{1}{2}\gamma\Omega^2\Big\{\frac{B4}{\Omega} \\ +\frac{\mu}{\Omega}[\zeta_o\cos(\psi+240^\circ) \\ +\sin(\psi+240^\circ)]B3\Big\}\end{gathered}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}
+
\begin{bmatrix}
0 \\
\frac{\nu_1}{I_{GE}}\sin(\psi+\psi_o)-\frac{\nu_2}{I_{GE}}\cos(\psi+\psi_o) \\
0 \\
\frac{\nu_1}{I_{GE}}\cos(\psi+\psi_o)+\frac{\nu_2}{I_{GE}}\sin(\psi+\psi_o) \\
0 \\
\begin{gathered}\frac{1}{2}\gamma\Omega^2\Big[B4+2\mu(\zeta_o\cos\psi+\sin\psi)B3+\mu^2\Big(\frac{1}{2}-\frac{1}{2}\cos 2\psi+\zeta_o\sin 2\psi\Big)B2\Big]\theta_o \\ +\frac{1}{2}\gamma\Omega^2\Big[B5+2\mu(\zeta_o\cos\psi+\sin\psi)B4+\mu^2\Big(\frac{1}{2}-\frac{1}{2}\cos 2\psi+\zeta_o\sin 2\psi\Big)B3\Big]\theta_1 \\ +\frac{1}{2}\gamma\Omega^2[B3+\mu(\zeta_o\cos\psi+\sin\psi)B2]\lambda_o \\ +\frac{1}{2}\gamma\Omega^2[B4\cos\psi+\mu(\cos\psi\sin\psi+\zeta_o\cos^2\psi)B3]\lambda_x \\ +\frac{1}{2}\gamma\Omega^2[B4\sin\psi+\mu(\sin^2\psi+\zeta_o\cos\psi\sin\psi)B3]\lambda_y+\frac{k_\beta}{I_y}\beta_p\end{gathered} \\
0 \\
\begin{gathered}\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi+120^\circ)+\sin(\psi+120^\circ)]B3+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+120^\circ)+\zeta_o\sin 2(\psi+120^\circ)\Big]B2\Big\}\theta_o \\ +\frac{1}{2}\gamma\Omega^2\Big\{B5+2\mu[\zeta_o\cos(\psi+120^\circ)+\sin(\psi+120^\circ)]B4+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+120^\circ)+\zeta_o\sin 2(\psi+120^\circ)\Big]B3\Big\}\theta_1 \\ +\frac{1}{2}\gamma\Omega^2\{B3+\mu[\zeta_o\cos(\psi+120^\circ)+\sin(\psi+120^\circ)]B2\}\lambda_o \\ +\frac{1}{2}\gamma\Omega^2\{B4\cos(\psi+120^\circ)+\mu[\cos(\psi+120^\circ)\sin(\psi+120^\circ)+\zeta_o\cos^2(\psi+120^\circ)]B3\}\lambda_x \\ +\frac{1}{2}\gamma\Omega^2\{B4\sin(\psi+120^\circ)+\mu[\sin^2(\psi+120^\circ)+\zeta_o\cos(\psi+120^\circ)\sin(\psi+120^\circ)]B3\}\lambda_y+\frac{k_\beta}{I_y}\beta_p\end{gathered} \\
0 \\
\begin{gathered}\frac{1}{2}\gamma\Omega^2\Big\{B4+2\mu[\zeta_o\cos(\psi+240^\circ)+\sin(\psi+240^\circ)]B3+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240^\circ)+\zeta_o\sin 2(\psi+240^\circ)\Big]B2\Big\}\theta_o \\ +\frac{1}{2}\gamma\Omega^2\Big\{B5+2\mu[\zeta_o\cos(\psi+240^\circ)+\sin(\psi+240^\circ)]B4+\mu^2\Big[\frac{1}{2}-\frac{1}{2}\cos 2(\psi+240^\circ)+\zeta_o\sin 2(\psi+240^\circ)\Big]B3\Big\}\theta_1 \\ +\frac{1}{2}\gamma\Omega^2\{B3+\mu[\zeta_o\cos(\psi+240^\circ)+\sin(\psi+240^\circ)]B2\}\lambda_o \\ +\frac{1}{2}\gamma\Omega^2\{B4\cos(\psi+240^\circ)+\mu[\cos(\psi+240^\circ)\sin(\psi+240^\circ)+\zeta_o\cos^2(\psi+240^\circ)]B3\}\lambda_x \\ +\frac{1}{2}\gamma\Omega^2\{B4\sin(\psi+240^\circ)+\mu[\sin^2(\psi+240^\circ)+\zeta_o\cos(\psi+240^\circ)\sin(\psi+240^\circ)]B3\}\lambda_y+\frac{k_\beta}{I_y}\beta_p\end{gathered}
\end{bmatrix}
$$

FOLDOUT FRAME

in the calling sequences of the program) contain the dynamics for the helicopter blade system in this example. The main program reads and writes the input data used to simulate the helicopter blade equations. The user supplies this portion of the program. The amount of data read into the program and written back out will depend on the complexity of the users mathematical model. A sample of the data input deck that is used to obtain the simulation run for the equations of motion is shown in Figure 1(a) and 1(b). Each card is numbered from 1 to 54 along the left margin and is discussed below:

Data Card Input Discussion for Simulation Option

Card 1 The first card is the users logon card for the computer system used.

Card 2 This card specifies the disk storage unit 07 and identifies the data set by the name SIMUL.R0001.

Card 3 The name of the users main program is MAIN2$$. The main program is called at this point and is followed by the data cards.

Card 4-16 These cards represent the input data for the users mathematical model. The user may have more or less data cards describing the mathematical model. The number of cards will depend on the complexity of the mathematical model and on the number of parameters to be read into the program to completely describe the dynamics of the system.

The data cards from this point on are read by the identification programs (most of these data cards are read by subroutine IDENT1 and INOUT and the format as shown here must be followed exactly). The fortran names for these variables are also given below.

Card 17 (format 3A4) For the simulation option the following characters are punched starting in column 1 *SIMULATION.

Card 18 (format 10A8) On this card any comment may be written between columns 1 through 80.

Card 19 (format 7I5, 12I1, I3) The variable names read into the program from this card are:

N1 number of parameters - maximum 36

ND number of data points over which the simulation is performed (ND=199)

NS number of measured outputs (not greater than 12)

KSK the skipping factor for data points to be read (can be set to zero for simulation option since no measured data is read into the program)

NSTP number of integration steps between measurement intervals used in integrating the equations of motion (for the simulation option NSTP is a skipping factor for the storing and printing of the results)

KSTOP	code number indicating the last data point should remain set at 10,000 for the three options illustrated.
LIST	set to zero for simulation option
INIT	set to zero for simulation option
ISTART	set to zero for simulation option

Card 20 (format 4A1) The variable on this card is defined as:

SYMB	four one-character symbols used to plot the time histories on the on-line printer.

Card 21 This card remains blank for the simulation run.

Card 22-25 (format 6E10.3) The format for each data card is 6E10.3 with a C in column 65 if the next card is a continuation of the parameters to be read. On these cards the values of the parameters in the simulation option is entered. Since a maximum of 36 parameters can be identified a maximum of 6 cards can be found here. For this case only 21 parameters are investigated and hence 4 cards are needed with 6 values each on the first three cards and 3 values on the 4th card.

Card 26-29 (format 6E10.3) These cards represent the lower bounds of the parameter values that are entered on cards 22 to 25. For the simulation option these cards are blank. The number of blank cards must be equal to the number of parameter cards (cards 22 to 25). For this case there are 4 parameter cards hence 4 blank cards are required.

Cards 20-33 (format 6E10.3) These cards represent the upper bounds of the parameter values that have been entered on cards 22 to 25. For the simulation option these cards are blank. The number of blank cards must be equal to the number of parameter cards (cards 22 to 25)

Cards 34 & 35 (format 12(A4,1X)) Each card contains the name of the parameter as they are entered on cards 22 to 25. A maximum of 4 characters for each name is allowed. There must be as many cards as needed to define the name of the N1 parameters. For this case N1=21 and with a format of 12(A4,1X) for each card, two cards containing the names of the parameters are required.

Cards 36-39 (format 6E10.3/6A8/6E10.3/6A8) These cards contain the scaling factors and the units for the measured and computed data. The scaling factors (maximum of 12 for the 12 possible measurements) scale the measured and computed data between magnitudes of +10,000 for internal computation. The scaling factors are also used to convert the output data back to engineering units. For the simulation option the scaling factors entered are chosen with some a prior knowledge of the magnitude for the output channels. The scaling factors are then determined by dividing the largest magnitude of each channel by 10,000.

The scale factors are entered on the first card which is followed by a second card with the units of the corresponding channel. The format for the scaling factors is 6E10.3 and the units are entered with format 6A8 on the next card. The ordering of the scaling card followed by the card containing the units must be followed if more than 6 scaling factors are needed. Also a C must be punched in column 65 of the first scaling card if more than 6 scaling factors are entered.

Cards 40-42 (format 10A8) The units of the parameters are defined on these cards. Ten names are punched per card with eight characters for each unit name. As many data cards as needed to define the units of all the parameters must be supplied.

(The following three cards are unique to the simulation option.)

Card 43 For the simulation option the characters DSET05 are entered on this card.

Card 44 The characters RUN are punched in the first three columns and a 1 appears in column 6 with a run identification number between 0 and 99 punched in columns 8 and 9. In columns 16 to 19 the step size to be used by the integration routine in millidegrees is entered. For this case the step size is 5778 millidegrees.

Card 45 The word END appears on this card for the simulation option.

Card 46 (**********) The 10 stars on this card indicate the end of the command sequence that started with data card 43.

Card 47 The first 4 columns will contain the word DSET and the next 2 columns contain the disk number (07 in this example) on which the generated data is stored. In columns 9 and 12 the user can enter run identification numbers of his choice.

Card 48 (*STORAGE) This command must follow the disk number card (card 47) with a 1 in columns 11 to 22 if all computed data is to be stored. For this example four ones appear in columns 11 to 14 to indicate storage of the first 4 output channels.

Card 49 (*GO PLOT) This command will prepare the program to plot the time histories that are specified by the next card.

Card 50 (format A8, I2, 4I5) The first 8 characters contain the command **OUTPUT. In columns 10, 15, 20 and 25 the channel numbers to be plotted are punched, which in this example are channels 1, 2, 3 and 4. If more than 4 channels are required to be plotted, the next card must again start with the command **OUTPUT and with the appropriate channel numbers in columns 10, 15, 20 and 25.

Card 51 (*GO PRINT) This command will print the calculated data that has been stored in the subroutine PRINT1.

Card 52 (**********) The 10 stars indicate the end of the above command sequence that started with card 47.

Card 53 (*RETURN) This card returns command from the subroutine IDENT1 to the users main program.

Card 54 (LOGOFF) This is a job control card that starts in column 3 and terminates the execution of the program.

The exact number of cards as outlined in the example will vary, depending on the problem the user is trying to solve. Therefore the data input card number will vary from problem to problem. However, the order of these data cards must be followed as has been outlined. In the next section the corresponding computer output for the simulation example is discussed.

Computer Program Output Discussion for Simulation Option

Figure 1(c) shows the corresponding output for the simulation run when the data input deck of figures 1(a) and 1(b) is executed. The output on figures 1(c) and 1(d) are produced by the users main program. As was discussed previously the user writes any desired output on the "on-line" printer required to describe the mathematical model. The output on figures 1(e) to 1(k) is produced by the identification program and is briefly discussed. The title printed on the top of figure 1(e) was entered on card 18 of the data input cards. The next 4 lines give a print-out of the data read into the program on card 19. The initial conditions printed here were read into the main program and then transferred through a common statement into the IDENT1 subroutine (see discussion of IDENT1 subroutine). The remaining print-out on this figure follows closely to the information that was read into the program in figure 1(a) starting with card 18. On figure 1(f) the first line identifies the run number that was read into the program on card 44 of figure 1(a). The variable KDT on the first line represents the step size in millidegrees at which the output data was stored. The remaining print-out indicates that the output variables requested to be stored have been stored and that an on-line printer plot of the data will follow. Figures 1(g) through 1(i) illustrate the on-line plotting feature of the computed output. Although the title of the plots are "Measured and Computed Time Histories", note that for the simulation option of the program, only the computed time history is plotted. On the right margin of each plot the corresponding time (azimuth angle in degrees for this example) is printed for each plotted data point. Only a portion of each time history is shown here for illustrative purposes. To evaluate the magnitude for each plotted axis produced by the on-line printer, the following formula is used. The scaling factor of the channel to be plotted is multiplied by the largest of the two Fortran variables (MEAS or COMP) printed at the end of each plot. This gives the maximum magnitude of the y-axis. Since 25 carriage spaces are allowed for the positive and negative y-axis of each plot, $\frac{1}{25}$ th of the maximum magnitude gives the magnitude of each carriage space. Applying the formula to variable

1 of figure 1(g) gives a maximum magnitude of 4000 ft-lbs (SCALE (1)* MEAS = $(4x10^{-1})$ $(1x10^{4})$=4000 ft-lbs) or 160 ft-lbs per carriage space ($\frac{1}{25}$ *4000=160 ft-lbs).

The printed output of the plotted data is given on figures 1(j) and 1(k). The information printed on these figures was stored in the PRINT1 subroutine (see discussion of PRINT1 subroutine). For the simulation example the data given under the title MEASURED VALUES on figure 1(j) is the same as given under the title COMPUTED OUTPUT VALUES, since no measured data is read into the program for the simulation option. Only a portion of the printed data is presented to illustrate the output format of the simulation option. The last few lines on figure 1(k) indicate the end of the simulation run with a print-out of the computation time to execute the option.

Figure 1(ℓ) shows the complete time history for the simulation run for 6 rotor revolutions. These plots are produced off-line from the computed data that was stored on computer disk 07 under the name SIMUL. R0001 (see input data deck figure 1(a) line 2). The upper two traces of figure 1(ℓ) represent the roll and pitch hub moments while the lower two traces give the swashplate roll and pitch angles in rotating coordinates for 6 rotor revolutions.

Example 2: Parameter Identification

As discussed in example 1, the users mathematical model must be incorporated in the identification program. The main program (MAIN2$$) and the subroutine DERIV$$, NEWP$$, OUTPUT$$ and PRINT1$$ are entirely supplied by the user. Some modifications are required of subroutine IDENT1$$, INOUT$$ and DYN$$. A discussion of program changes required to incorporate the users mathematical model is given in the section "Organization of Programs". All information the user prefers to display about the mathematical model should be read into the main program and written back out on the on-line printer. This has been done for the mathematical model treated here, as was seen for the data inputs and print-out of example 1 figure 1(a) and 1(b). A similar discussion will follow using the programs in the parameter identification option. First the data input cards are discussed for this option which are followed by a discussion of the parameter identification output.

A listing of the input data deck for this example is given in figures 2(a) and 2(b). The input cards are numbered from 1 to 68 for the following discussion. The explanation of some of these input cards was given in example 1 where the simulation option was outlined.

<u>Data Card Input Discussion for Identification Option</u>

Card 1 — The first card as well as the last card (card 68) are unique to the users computing system and the user must supply these cards.

Card 2 This card defines a data set on which the output is stored (disk 10 with the name IDENT. R0026). The disk number to be used for storing the data is requested on input card 63. If the user does not have the capability of disk storage, then the user may either write the output information on tape or print the output without storing it.

Card 3 This card defines the data set on which the measured data required for the identification has been stored (disk 20 with the name SFCT01.RUN6). The disk number to be used is specified by card 43. If the user does not have the capability of disk storage, the measured data must be supplied either on tape storage or on a punched card deck. Further discussion using punched card inputs is given on Card 43.

Card 4 The user calls the main program on this card. For the three examples given here, the name of the main program is MAIN2$$. The input cards for the identification run follows the calling of the main program.

Card 5-17 These cards represent the input data the user supplies to his main program. The number of cards will depend on the complexity of the users mathematical model and on the number of parameters of the model the user desires to read into the program.

The input data cards from this point on are read by the identification programs in subroutine IDENT1$$ and INOUT$$. As has been mentioned in the simulation example 1, the format for the input data from this point on must be carefully followed.

Card 18 — On this card any comment may be written in columns 1 to 80. This comment is written at the beginning of the identification run.

Card 19 — format (7I5, 12I1, I3) The variables are defined as:

N1	number of parameters - maximum 36
ND	number of data points in each measured output - equal to or less than 199
NS	number of measured outputs - equal to or less than 12
KSK	the skipping factor for the measured data to be read into the program
NSTP	number of steps between the measurement intervals used in integrating the equations of motion - example: if NSTP=2 and the data is recorded every 10 degrees then the integration step size is 5 degrees
KSTOP	code number indicating the last data point - should remain set at 10,000 for the three options illustrated
LIST	this variable is either 0 or 1. If it is set to 1, then the measured data read from computer disk, tape or cards is written out before the identification program. If LIST is set to 0 no measured data is printed.

INIT 0 for identification option

ISTART defines the first value to be read from the measured data. (example, if ISTART=5, the program will start reading the measured data starting with the fifth value) ISTART is normally set to zero

Card 20 format (4A1, 1X, 6I5, 15X, F10.5, 10X, 2I5) The variables on this card are defined as:

SYMB four one-character symbols used to plot the time histories

KSWTCH set to zero for all options

ITPL specifies the iteration number at which a plot is requested (the on-line printer produces a two axis plot which plots on each axis the measured and computed data)

IPLOT1 channel number to be plotted on the first axis

IPLOT2 channel number to be plotted on the second axis

KSKIP skipping factor for data to be plotted (example; if it is desired to plot the measured and computed data of output channel number 3 and 5 at iteration 2 and at every other data point, then the above variables are set to: ITPL=2, IPLOT1=3, IPLOT2=5, ISKIP=2. If all of these variables are set to zero, a later option (cards 52 to 61 can be used to plot the final results of the identification)

ITMAX maximum number of iterations

THR threshold value for subroutine COR - between 0 and 1 but usually set at about .001. THR is used in the dependent analysis of the subroutine COR. The extent of the dependence between parameters is specified by the threshold value (THR). For linear independence THR=0 and for linear dependence THR=1. For a complete discussion of the dependent analysis and the subroutine COR see reference 1.

MCOR integer between 0 and 6 which specifies the amount of output to be printed from subroutine COR. If MCOR=0, no output is printed from subroutine COR while MCOR=6 gives the maximum printed output from subroutine COR. MCOR is usually set to 6 for the identification run.

ICOR iteration number at which the output of the dependence analysis in subroutine COR is printed

Card 21 (format 36I1) The variable name for this card is MNS(N1). A maximum of 36 one-digit integers associated with the 36 parameters that can be identified by the program. If any one of the 36 integers is set to zero, the corresponding parameter is not identified.

In the example, a 1 has been punched in columns 2,3,4, 15, 16 and 21 which indicates that parameters associated with column 2,3,4,15,16 and 21 are to be identified. Note that in this example a maximum of 21 parameters of the model can be identified since N1=21 on card 19.

INIT	0 for identification option
ISTART	defines the first value to be read from the measured data. (example, if ISTART=5, the program will start reading the measured data starting with the fifth value) ISTART is normally set to zero
Card 20	format (4A1, 1X, 6I5, 15X, F10.5, 10X, 2I5) The variables on this card are defined as:
SYMB	four one-character symbols used to plot the time histories
KSWTCH	set to zero for all options
ITPL	specifies the iteration number at which a plot is requested (the on-line printer produces a two axis plot which plots on each axis the measured and computed data)
IPLOT1	channel number to be plotted on the first axis
IPLOT2	channel number to be plotted on the second axis
KSKIP	skipping factor for data to be plotted (example; if it is desired to plot the measured and computed data of output channel number 3 and 5 at iteration 2 and at every other data point, then the above variables are set to: ITPL=2, IPLOT1=3, IPLOT2=5, ISKIP=2. If all of these variables are set to zero, a later option (cards 52 to 61 can be used to plot the final results of the identification)
ITMAX	maximum number of iterations

Card 22-25 (format 6E10.3) The nominal values of the parameters to be identified are punched on these cards. The names and units for each of these parameters is specified latter on in the data deck. The user determines the ordering of the parameters in the subroutine NEWP$$ (see description of subroutine NEWP$$). Once this order has been established the parameters must be read into the program accordingly. The C in column 65 indicates continuation of the parameters to be read into the program (also see simulation example for this discussion).

Card 26-29 (format 6E10.3) The lower limit the identification program can adjust the nominal values of the parameters are punched on these cards. The same formating is used to read these cards as was used to read in the nominal values (card 22-25).

Card 30-33 (format 6E10.3) The upper limit the identification program can adjust the nominal values of the parameters are punched on these cards.

Card 34-35 (format 12(A4,1X) These cards contain the name of the parameters in the order they are entered on card 22 to 25. A maximum of 4 characters for each name is allowed. There must be as many cards as needed to define the name of the N1 parameters.

Card 36-39 (format 6E10.3/6A8/6E10.3/6A8) These cards contain the scaling factors and the units for the measured and computed data. For the identification example the measured data is read into the program with an integer format (see subroutine READIN). The scaling factors used to convert the integer format to engineering units are entered on these 4 data cards. If only four measurements are read into the program then the corresponding 4 scaling factor must be entered. The scaling factors for the remaining channels can be obtained as has been discussed under the simulation example 1.

Card 40-42 (format 10A8) The units of the parameters are defined on these cards. Ten names can be punched per card with eight characters for each unit name. As many data cards must be supplied to define the units of all parameters.

Card 43 For the identification option this card contains DSETxx where xx represents the disk unit on which the measured data is stored. If the data is not read from a computer disk, then the data can be entered at this point using data cards. The procedure for entering the data cards has been discussed in the simulation example 1(a) under Card 43 to 45. The data cards will follow card 44 and are entered with a format compatible with the subroutine READIN$$.

Card 44 This card contains the command *RELVAR starting in column 1. The command refers to the percentage variation that each parameter is allowed to vary from its nominal value for the first iteration. The cards immediately following the *RELVAR command contain the percentage variation for the parameters. If the percentage variation is set to zero the program starts with a 1% change for each parameter. After the first iteration the program determines the percentage variation internally. At the end of the desired number of iterations the percentage variation for each parameter is printed. This information can be read again into the program at this point if more iterations are required for the same case.

Card 45-48 (format 6E10.3) These cards contain the percentage variation for all the parameters. The same format as was used to read in the nominal values of the parameters is used to read in the percentage variations.

Card 49 This card will contain the command *READWGH. The command refers to the weights that are to be used for the NS values. Following the values of weights, the values of ERRMIN and DMIN are read into the program. The variable ERRMIN refers to the minimum ERROR (error between measured and calculated value) the identification algorithm will stop iterating. The value DMIN refers to the minimum value a parameter is considered irrelevant in effecting the output response.

For this example four measurements are used for the identification of the parameters, hence four weights are given on card 49 followed by the values for ERRMIN and DMIN. It should be noted that the weights for all channels was taken equal to 1. The weights should be set to 1 if the measured data has been scaled to fall between ±10,000 as was done in all cases here. If the measured data had not been scaled, the difference in magnitude between the angle and moment measurements would be very large and a weighting factor would be required.

Card 51 This card contains ten stars (*) indicating the end of the series of data cards beginning with card 44 titled *RELVAR. The starred card must always be present regardless of whether the relative variation data or the weight data is present. The input data cards starting with card 52 and ending with card 65 are referred to as the second set of input commands that must always end with a starred card (card 66). Anyone or all of the options listed can be requested. In this example all options are given to illustrate to the user how these options are used.

Card 52 The command *GO PLOT will prepare the program to plot some of the time histories that are specified by the next card.

Card 53 (format A8, I2, 4I5) The command card **OUTPUT specifies in columns 10, 15, 20 and 25 which channels are to be plotted. If more than four channels are required to be plotted the next card must again be **OUTPUT with the appropriate channel numbers in columns 10, 15, 20 and 25. A maximum of 12 channels can be plotted hence a maximum of 3 output cards may be required. For this example four output channels are plotted and only one output command card is needed.

Card 54-57 (format 8A, I2, 2I5) The command card **RESAMP plots the amplitude and the amplitude difference (residual amplitude) between the measured and calculated data. The first integer on the **RESAMP card refers to the output channel for which the residual amplitude is to be calculated. The second integer refers to the channel number in which the calculated residual amplitude is stored. The computed and measured results are then plotted on the left half of the computer paper, with the residual amplitudes plotted on the right half of the computer paper.

Card 58-61 (format 8A, I2, 2I5) The command card **RESCOR plots the amplitude and the residual autocorrelation function corresponding to the outputs. The first integer on the **RESCOR card refers to the output channel for which the residual amplitude is to be calculated. The second integer refers to the channel number in which the calculated residual autocorrelation is stored. The results are plotted as was discussed under card 54 to 57.

Care must be exercised when using the above two options that adequate number of channels are available to store the calculated residual amplitudes and the residual autocorrelations. For example, if it is required to make the above calculations for 5 of the output channels, then a total of 15 channels are required but only 12 channels are available. In this case not all the residual amplitudes and residual autocorrelation functions can be calculated unless three of the output channels are destroyed.

Card 62 (format 8A) This card contains the command *GO STORE that prepares the program for storing the desired channels specified by the next two cards.

Card 63 (format 4A, I2, 2X, 2I2) This card will contain the command DSET followed by the disk number on which the output data is stored. For this example the disk number is 10 and the name of the data set is given on card 2 as IDENT.R0026. The remaining integers on this card can be left blank or used for identifying the run number.

Card 64 (format 8A, 2X, 12I1) The command card *STORAGE followed by 12 one digit numbers specifies which channels are to be stored. If any one of the 12 channels is not required to be stored then the corresponding digit on this card is set to zero.

Card 65 (format 2A5) The command card *GO PRINT will print the data stored in the PRINT1$$ subroutine.

Card 66 (format 2A5) This card contains 10 stars to indicate to the program that the end of the second set of command cards has been reached (second set of command cards started with card 52). The number of cards between cards 52 and 65 will vary depending on what output the user likes to calculate and display. For example, the user may have been only interested in a plot of the output (command cards 52 and 53) and a computer print out of the results (card 65). In this case only three cards would appear between card 51 and 66 but the starred command (card 66) must be present to end this command sequence.

Card 67 (format 8A) The command *RETURN returns control from the identification programs to the users main program.

Card 68 This is the computer job termination card for the computer system used to generate the data for the identification example discussed here.

The exact number of cards as outlined in this example will vary, depending on the problem the user is trying to solve. However, the ordering of the data cards should be followed as has been outlined in this example. Some variation in the ordering of the option cards *RELVAR, *READWGH, *GO PLOT, **OUTPUT, *GO STORE and *GO PRINT may be carried out, although it is best to maintain the ordering that has been outlined in this example. Any one of these options or all of the options shown in this example may be left out without effecting the execution of the program.

Computer Program Output Discussion for Identification Option

The computer output produced after executing the identification program is shown in figures 2(c) through 2(ii). The printed output on figure 2(c) and the top 2 lines of figure 2(d) are produced by the users main program. The remaining lines on figure 2(d) are written by the subroutine READIN$$ after the measured data is read into the program from disk number 20 (see input data card 2). The information printed indicates that the increment between measured data points (KDT) is 11556 millidegree and that every other data point (SKIP) was read. A total of 64 data points for each of the 9 channels indicated on figure 2(d) have been read and stored in the program.

The printed output on figures 2(e) to 2(ii) is produced by the identification program. All information printed on figure 2(e) was read into the program from data cards 18 to 49. Only the initial conditions were read into the program from data cards 5 to 17 and then were transferred through common into the identification program. These initial conditions are printed on figure 2(e) along with the remaining parameters of the system.

On figure 2(f) the title "ITERATION O" is printed and below the title the values of the parameters used in the model simulation are displayed. These initial parameter values determine the math model's response before the identification algorithm is used. The relative error and the cost calculated from the difference between the model's response and the measured data are also given. The relative error is defined as the square root of the sum of the error squared divided by the area between the model response and the measured data. The cost is defined as the sum of the error (between model and measurement) squared. When scanning the identification output results the relative error and the cost for each iteration should be observed. These quantities should decrease after several iterations if the identification algorithm is improving the match between the model simulation and the measured data.

The results printed on figure 2(g) and those on top of figure 2(h) are calculated in subroutine COR. The subroutine determines the uniqueness of the parameters to be identified. Two examples discussing the output from this subroutine are given in reference 1 on page 51 and 52. A brief discussion of this output will be given here. The first column titled "BASIC PARAMETER" gives a list of the parameters that were specified to be identified by the program in order of decreasing orthogonality. The second column titled "SEPARATION" give a measure of the orthogonality. The third column titled "CRITICAL PARAMETER" gives a list of those parameters that may become dependent if the threshold ε is decreased (The threshold ε is the angle between two basic vectors; if $\varepsilon=0$ the vectors are linear dependent. The threshold value was specified on card 19). The last column gives the separation for the critical parameters. The results are interpreted as follows: the basic parameter chosen was TWST and the next parameter found by the program which was almost orthogonal to the parameter TWST was the parameter BETA. The separation between parameters TWST and BETA was 9.49E-01. (a separation of 1 would indicate complete orthogonality or complete independence between these two parameters). The critical parameter was found to be CLEC with a separation of 1.27E-02. This indicates that if the threshold was increased from the value specified in the program, the next most likely parameter to become dependent will be CLEC. This procedure continues for all the parameters that were specified to be identified with the basis vectors becoming less and less orthogonal. On figure 2(h) the threshold used in the program is printed followed by a line of digits 0 or 2 one for each of the 21 parameters. The zero indicates the corresponding parameter was not requested to be identified. The two indicates the corresponding parameter was specified to be identified and was found to be independent of the other parameters. A one for any one of these digits indicates the corresponding parameters to be dependent on some other parameter in the set. All this information is summarized on figure 2(h) under the titles "INDEPENDENT PARAMETERS", "IRRELAVANT PARAMETERS", "NOT ESTIMATED" and "NOT USED". The row starting with the word STATUS and followed by 21 numbers with the nomonic NU (abbreviation for not used) refers to the 21 parameters of the model and whether the parameter was requested to be identified. The following row starting with the nomonic SENS gives the sensitivity of the parameters that were specified to be identified. The last five rows on figure 2(h) give the results for the first iteration with the relative error and the cost for this iteration printed on the right hand side (note that the relative error and cost have decreased from iteration 0). The rest of the information printed on figure 2(h) gives the values for the identified parameters after the first iteration. The print out on figures 2(i) and 2(j) is similar to what has been discussed previously. On figure 2(k) the final results of the identification are given. First a fit test between the measured and calculated results is given in terms of cost, correlation and relative error which is followed by the values of the estimated parameters. The name of each parameter and its estimated value, error bounds, units, sensitivity and dependency index for each parameter is given. Following this print out the output requested on data cards 52 to 62 is displayed.

The *GO PLOT command initiates the plotting of variables specified on data cards 52 to 61. The computer plot of variables stored in channels 1, 2, 3 and 4 are plotted on figures 2(e) through 2(o). On these figures the comparison of the measured and calculated variables are plotted with the corresponding time (angle measurement) printed on the far right side of the figures. In order to evaluate the magnitude of each plotted axis produced by the on-line printer, the following formula must be used. The scaling factor of the channel to be plotted is multiplied by the largest of the two Fortran variables (MEAS or COMP) printed at the end of each plot. This will give the maximum magnitude of the y-axis. Since 25 carriage spaces are allowed for the positive and negative y-axis of each plot, $\frac{1}{25}$th of the maximum magnitude gives the magnitude of each carriage space. (see example 1 where this calculation has been done for illustrative purposes).

The information printed on figures 2(p) through 2(w) was specified by data cards 54 thru 57 with the option cards **RESAMP. The **RESAMP option calculates the residual amplitude correlation for the measured and calculated output. On the left side of figure 2(p) the measured and calculated output for the first channel is plotted and the resulting amplitude correlation is plotted on the right side of figure 2(p). The results for the next 3 channels are plotted on figures 2(r) to 2(w).

The plot on figure 2(x) gives the residual autocorrelation of the error between computed and measured data for channel 1. The measured and calculated results are plotted on the left while the residual autocorrelation function is plotted on the right side of figure 2(x).

The residual autocorrelation for channel 2, 3 and 4 are given on figures 2(z) and 2(ee).

At the middle of figure 2(ee) the command to store the calculated data on computer disk is printed. This command was given on data input cards 62 to 64. At the beginning of the data deck, disk unit 10 was specified for storing the output data under the name IDENT. R0026. On figure 2(ff) run identification numbers specified by the user, the step size in millidegrees and an identification digit of 0 or 1 (0 indicates the variable is not stored while a 1 indicates the corresponding channel is stored) are printed. This information was read into the program from data cards 63 to 64. The results on figure 2(gg) confirms the storing of each channel as it was stored on disk. Also a scale factor to be used by the user in retrieving the data from disk is given for those channels that require a new scaling factor. The scaling factors for channels not given on figure 2(gg) must use the scaling factors given on data card 36 to 39 of the input deck.

The final output for this identification run is given by the command *GO PRINT. This command prints the channels the user has requested in the subroutine PRINT1$$. For this example measured and computed results of the identification are printed on figures 2(hh) and 2(ii).

The information stored on computer disk (unit 10 under name IDENT.R0026) was plotted more accurately (as compared to the on line printer plot) with a CALCOMP plotter and is shown in figures 2(jj) & 2(kk). The upper traces of each plot show the comparison of the measured and calculated results after two iterations using the identification program. The middle trace is a plot of the residual amplitude correlation while the bottom plot shows the residual autocorrelation of the error between computed and measured data.

This completes the discussion of the input data cards and corresponding computer output for the program in the identification mode. For this example only two iterations had been specified, although the format for the output will be similar if more iterations are required. In some cases the program may stop before the total number of iterations specified have been reached. This will occur if the algorithm can not find further improvements in the time history matching between measured and calculated data.

Example 3: SENSITIVITY ANALYSIS

This example outlines the procedure to be followed to obtain a a sensitivity analysis of the model parameters. With the users mathematical model incorporated into the identification programs as discussed under "Organization of Programs" the sensitivity analysis can be obtained. The sensitivity as calculated in this program is defined as a dimensionless ratio of the change in the output due to a change in the parameter.

A similar procedure used in examples 1 and 2 to obtain the simulation and identification runs is used to obtain the sensitivity analysis. First, the data input deck for the sensitivity analysis is discussed line by line. Second, the corresponding output obtained by executing the input deck is given.

A listing of the input data deck is shown in figures 3(a) and 3(b). The input cards are numbered from 1 to 49 for the following discussion. It should again be pointed out that the user may not have the same number of data cards. This will depend on the complexity of the users mathematical model and on how much data he prefers to read into the program and the number of parameters to be identified. For this example the sensitivity analysis is performed for 21 parameters, although the program has the capability to obtain the sensitivity analysis for 36 parameters.

An explanation of the input data cards shown in figures 3(a) and 3(b) is given below.

Data Card Input Discussion for Sensitivity Option

Card 1	This is the logon procedural card for the computing system used to execute the program.
Card 2	The user calls the main program on this card.
Card 3-15	These cards represent the input data the user supplies to the main program that describe the mathematical model. (Same as in example 1 and 2).

As in the previous two cases, the input cards from here on are read by the identification subroutines.

Card 16	(format 3A4) The command card *SENSITIVITY defines the sensitivity option of the identification programs.
Card 17	(format 10A8) Any comment can be written on this card by the user. This comment is written at the beginning of the computer print-out. The name of the option used is an appropriate comment to be punched on this card.
Card 18	(format 6I5) The variables are defined as:
N1	number of parameters - maximum 36
ND	number of data points over which the sensitivity analysis is to be caluclated (ND=199)
KSK	set equal to 0 or 1 for sensitivity analysis
NSTP	number of integration steps between the measurement interval specified by card 43
KSTP	a code number set to 10000 for the sensitivity option

Card 19	(format 4A1, 46X, F10.5, 10X, 2I5) The variables on this card are defined as:
SYMB	left blank for the sensitivity analysis
THR	threshold value for subroutine COR which can take on values from 0 to 1 but is usually set to .001. (see explanation on Card 20 of example 2)
MCOR	set to 6 for the sensitivity analysis
Card 20	(format 36I1) A maximum of 36 one-digit integers associated with the 36 parameters. For the sensitivity analysis the integers must be set to 2 if the sensitvity of the parameters is to be obtained or 0 if it is not to be calculated.
Card 21-24	(format 6E10.3) The nominal values of the parameters for which the sensitivity analysis is to be calculated. (See discussion under example 2, Card 22-25).
Card 25-28	(format 6E10.3) For the sensitivity analysis these cards are blank. The number of blank cards must be equal to the number of parameter cards (cards 21-24).
Card 29-32	(format 6E10.3) These cards are blank for the sensitivity analysis. The number of of blank cards must be equal to the number of parameter cards.

Card 33-34 (format 12(A4, 1X)) These cards contain the name of the parameters whose nominal values where given on Cards 21 to 24.

Card 35-38 (format 10A8) The units for the parameters given on Cards 21 to 24 are specified on these cards. Ten names can be punched per card with eight characters for each unit name. As many data cards are supplied to define the units of all parameters.

Card 42 (format A3, I2) For the sensitivity option the characters DSET05 are entered on this card.

Card 43 (format A3, 2I3, 6X,I5) For the sensitivity option the work RUN appears in the first three characters of this card. The next two integers can be used by the user for run identification purposes. The last integers represent the measurement interval or the step size if NSTP on card 18 is set to 1. For NSTP>1 the measurement interval will be divided by NSTP.

Card 44 (format A3) This card must contain the word END.

Card 48 This is the computer job termination card for the computing system used to generate the sensitivity analysis.

As in the previous two examples some variation in the number of data cards for the sensitivity run may occur, but this will depend on the problem the user is trying to solve. The order of these data cards will remain the same for the sensitivity option. The computer output obtained after executing the sensitivity example data cards is discussed below.

COMPUTER PROGRAM OUTPUT DISCUSSION FOR SENSITIVITY OPTION

Figure 3(c) shows the computer printout for the sensitivity example. The output on figure 3(c) and part of figure 3(d) are produced by the users main program as has been previously discussed. The bottom two lines on figure 3(d) represent the beginning of the sensitivity analysis output. These two lines were entered into the program from data cards 16 and 17 and are displayed here. On figure 3(e) the identification program displays a message that is printed from the READIN$$ subroutine which has no meaning for this option. The remaining informantion printed on figure 3(e) was specified on data cards 18 and 42. Figure 3(f) is a printout of information that was entered through data cards 16 to 41. This printout is similar to that shown for the simulation and identification example. The last 5 lines on figure 3(f) are a printout of the model parameters used in the model simulation for obtaining the sensitivity analysis.

The information printed on figure 3(g) is the result of the dependent analysis performed in subroutine COR (see reference 1). The first column titled "BASIC PARAMETER: gives a list of the parameters in order of decreasing orthogonality. The second column titled "SEPARATION" gives a measure of the orthogonality. The third column titled "CRITICAL PARAMETER" gives a list of those parameters that may become dependent if the threshold is increased. (The threshold ε is the angle between two basic vectors; if $\epsilon=0$ the vectors are linear dependent. The value for ϵ was read into the program on data card 19). The fourth column titled "SEPARATION" gives the separation for the critical parameter. To interpret the output listed in the four columns, one observes first what the "basic parameters" are for the model and next which parameters are the "critical parameters". As an example, the first parameter printed is GYIP, and the next parameter selected by the program is LAMX with a separation of 1 which indicates orthogonality between these vectors. The critical parameter associated with LAMX is BIX with a separation of 1.06E-01. The next basic parameter found is BETA with a separation of 9.03E-01 and the corresponding critical parameter is SWEP with a separation 7.68E-02. This procedure continues for all the parameters of the model with the basic vectors becoming less and less orthogonal. Following this output parameters of the model not listed in the first column that are dependent, irrelevant or that have been discarded a priori by the user are given. In this example GSPD, GYIP are found to be dependent.

The program next indicates that if GSPD is increased by one unit (absolute variation-symbolized by a prime) then GYIP will decrease by -5.66E-03 units. The same reasoning follows for the remaining parameter listed on figure 3(g). This information is useful in estimating the error in GYIP due to the lack of knowledge of parameter GSPD.

On the top of figure 3(h) the number of parameters and the threshold used in this analysis are printed. Below this line the words "MAGIC NUMBER" is written followed by a number either 0, 1, or 2 for each of the parameters of the model (21 in this example). The numbers have the following meaning: 0 indicates the corresponding parameter is irrelevant or was requested by the user to be held fixed, 1 indicates that the corresponding parameters is a dependent parameter and 2 indicates that the parameter is independent for the mix of parameter chosen for the sensitivity analysis. The results of the dependent analysis are summarized by the printout starting on the third line from the top of figure 3(h). At the bottom of figure 3(h) the parameter increments for each of the parameters for the gradient evaluation are printed. For the parameter identification option, this information can be entered into the program with data cards. (See parameter identification input data example).

The results printed on figure 3(i) give the sensitivity for each of the parameters (21 for this example). The sensitivity is a measure of the model's response to a parameter change. A sensitivity of 1 corresponds to a direct proportionality between the corresponding parameter and the output. If the sensitivity is very small with respect to 1 it indicates that the output is almost independent of the parameter. The parameter names are listed in the first column and their sensitivities are given under the column titled "SENSITIVITY". The dependency of the parameters as has previously been discussed are also given under the column "DEPENDENCEY INDEX".

The remaining printout on figure 3(i) indicates the end of the sensitivity run and the computation time required to per-from this run.

```
  LOGON FSTGK,MAINPR,60  ;FST19 T4947 'GERD KANNING STOP 17 PH 5455'
DDEF FT07F001,VS,SIMUL,R0001
CALL MAIN2SS
 0 0 0 0 1 0 0
+.40    +00+.268   +03+.2684 +03+.165   +02+.117   +01+.97    +00
+.15    +01+.1926  +03+.188   +01+.4566 +01+.00    +00+.12021+06+.145   +01
-.943   +01+.225   +01+.0     +00+.70    +02+.2246  -02
+.0     +00+.0     +00-.2428  +01+.11296+01+.10    -01
+.1912  +01-.4666  +01+.14    +01-.12535+02+.592   +00+.1481  +02+.857   +01
+.3     +00+.15    +00+.60    +02+.00    +00+.10200+05+.87    +00
+.00    +00+.70    +01+.80    +02
+.2652  +01-.2856  +02-.115   +01-.519   +02-.507   +02+.121   +02
+.1768  +01                -.252   +01
+.5778  +01+.00    +00+.00    +00+.2     +01
+.10    +02+.10    +02+.10    -01+.10    -01+.1     -01+.1     -01+.1     -01

*SIMULATION
HELICOPTER ROTOR SIMULATION WITH NOMINAL VALUES FOR THE MODEL PARAMETERS
   21   199    4     0     210000     0000000
0*+.

+.268   +03+.97    +00+.857   +01+.12021+06+.4566 +01+.15    +01     C
+.87    +00-.699   -02-.10    -04+.1     -04+.188   +01+.225   +01     C
+.1     -01+.355   +00+.80    +02+.7     +01+.4     +00+.5296  +02     C
+.3     +00+.15    +00-.943   +01









BIY ,LOSS,CLEC,BETA,LOCK,SWEP,  C2,LAM ,LAMX,LAMY,BRDO,PCON,
  CD,  MU,COMP,FDMP, BIX,GSPD,GYIP,GYID,TWST
+.40    +00+.40    +00+.5     -03+.5     -03+.1     +01+.1     +01     C
 FT-LBS  FT-LBS  DEGREE  DEGREE
+.1     +01+.1     +01

SLUG.FT2(NO DIM)DEG      FTLB/RAD(NO DIM)DEG      (NO DIM)(NO DIM)(NO DIM)(NO DIM)
FT       DEG      (NO DIM)(NO DIM)FTLB/R.SFTLB/R.SSLUG.FT2(NO DIM)SLUG.FT2SLUG.FT2
DEG
DSET 5
RUN  1  6       5/78
END
```

(a) Data input deck.

Figure 1.- Simulation run.

```
**********
DSET07  0106
*STORAGE  1111
*GO PLOT
**OUTPUT 1     2     3     4
*GO PRINT
*********
*RETURN
  LOGOFF
```

(b) Data input deck.

Figure 1.- Continued.

TWO DEGREE OF FREEDOM HELICOPTER BLADE AND GYRO SIMULATION

PROGRAM CONTROL LOGIC
IFLOQ= 0 IHINGE= 0 IROT= 0 INEW= 0 ISIML= 1 IFLOW= 0 IREC= 0

BLADE PARAMETERS
BIX= 0.40 BIY=268.00 BIZ=268.40 RADIUS=16.5 CHORD=1.17 BLADE LOSS=.97

SWEEP= 1.50 RPM= 192.6 HINGE POSITION=1.88 LOCK= 4.566 DELTA3= 0.00 K BETA=0.120210E 06 PI= 1.45

TWIST= -9.43 PRECONE= 2.250 CT= 0.0000 AIR VELOCITY= 70.00 AIR DENSITY=.002246 ALPHA= 0.000 ALPHAS= 0.000

LITTLE A1S= 0.000 LITTLE B1S= 0.000 BIG A1S=-2.428 BIG B1S= 1.130 CD=0.010 LAM=0.000000 LAMX=0.000000 LAMY=0.000000

BLADE INITIAL CONDITIONS
BETA1= 1.912 BETA1 RATE= -4.67 BETA2= 1.400 BETA2 RATE= -12.53 BETA3= 0.592 BETA3 RATE= 14.81 COLLECTIVE= 8.6

GYRO PARAMETERS
IP= 0.300 ID= 0.150 GYRO ANGLE= 60.0 GYRO DAMP L= 0.00 GYRO RPM=10200.0 MECHANICAL ADVANTAGE C2=0.870

FEATHERING RESTRAINT KZETA= 0.00 FEATHERING DAMPING CZETA= 7.00 SWASHPLATE DAMPING CSDAMP= 80.00 PSIOF= 0.00

INITIAL CONDITIONS ON GYRO
DELTA1= 2.652 DELTA1 RATE= -28.56 DELTA2= -1.150 DELTA2 RATE= -51.90 NU1=-50.70 NU2= 12.10

PHI= 1.768 RPHI= 0.00 THET= -2.520 RTHET= 0.00 RMBIAS= 0.00 PMBIAS= 0.00 RABIAS= 0.000 PABIAS= 0.000

INITIAL CONDITIONS FOR STARTING AND STOPPING THE INTEGRATION ROUTINE
DELPSI= 5.78 REVS= 0.00 REVO= 0.00 REVF= 2.00 TRUNC=0.0000

SCALE FACTORS FOR PLOTTING PROGRAM

SCALE(1)	SCALE(2)	SCALE(3)	SCALE(4)	SCALE(5)	SCALE(6)	SCALE(7)
1.000E 01	1.000E 01	1.000E-02	1.000E-02	1.000E-02	1.000E-02	1.000E-02
SCALE(8)	SCALE(9)	SCALE(10)	SCALE(11)	SCALE(12)	SCALE(13)	SCALE(14)
0.000	0.000	0.000	0.000	0.000	0.000	0.000

(c) Output.

Figure 1.- Continued.

```
                                                   PARAMETERS CALULATED FROM INPUT DATA
                                              OMEGA= 20.169 GYSPED= 52.96 MU= 0.355 LAMBDA=-.006999

                              DELTAT=0.5000E-02 TIMES=0.0000     TIMEO=0.0000     TIMEF=0.6231E 00
*SIMULATION
HELICOPTER ROTOR SIMULATION WITH NOMINAL VALUES FOR THE MODEL PARAMETERS

 DATA CARDS LISTING*****RUN NUMBER  1- 6
X AXIS INCREMENT KDT= 11556 SKIP=  2     199 POINTS PER CHANNEL
THE  0 FOLLOWING CHANNELS WERE READ : 0,
```


(d) Output.

Figure 1.- Continued.

```
HELICOPTER ROTOR SIMULATION WITH NOMINAL VALUES FOR THE MODEL PARAMETERS
------------------------------------------------------------------------

 21 PARAMETERS        4 OUTPUTS

199 MEASURED VALUES PER OUTPUT    ( 796 MEASUREMENTS),        2 WAS THE SKIPPING FACTOR USED

    TIME INTERVAL BETWEEN MEASUREMENTS  0,0050 SEC,          2 STEPS OF INTEGRATION IN EACH INTERVAL

  ANGLE INCREMENT BETWEEN MEASUREMENTS      5,78 DEG

INITIAL CONDITIONS
  4,629E-02 -4,985E-01 -2,007E-02 -9,058E-01  3,337E-02 -8,144E-02  2,443E-02 -2,188E-01  1,033E-02  2,585E-01

SCALING FACTORS FOR DATA UNITS
IDTU=  4,000E-01 FT-LBS  /  4,000E-01 FT-LBS  /  5,000E-04 DEGREE  /  5,000E-04 DEGREE  /  1,000E 00       /  1,000E 00       /
  1,000E 00        /  1,000E 00        /

PARAMETERS NAMES
    1) BIY      2) LOSS     3) CLEC     4) BETA     5) LOCK     6) SWEP     7)   C2     8) LAM      9) LAMX    10) LAMY    11) BRDO    12) PCON

  13)   CD    14)   MU    15) COMP    16) ^DMP    17)  BIX    18) GSPD    19) GYIP    20) GYID    21) TWST

INITIAL VALUES OF THE PARAMETERS
  2,680E 02  9,700E-01  8,570E 00  1,202E 05  4,566E 00  1,500E 00  8,700E-01 -6,990E-03 -1,000E-05  1,000E-05  1,880E 00  2,250E 00

  1,000E-02  3,550E-01  8,000E 01  7,000E 00  4,000E-01  5,296E 01  3,000E-01  1,500E-01 -9,430E 00
```

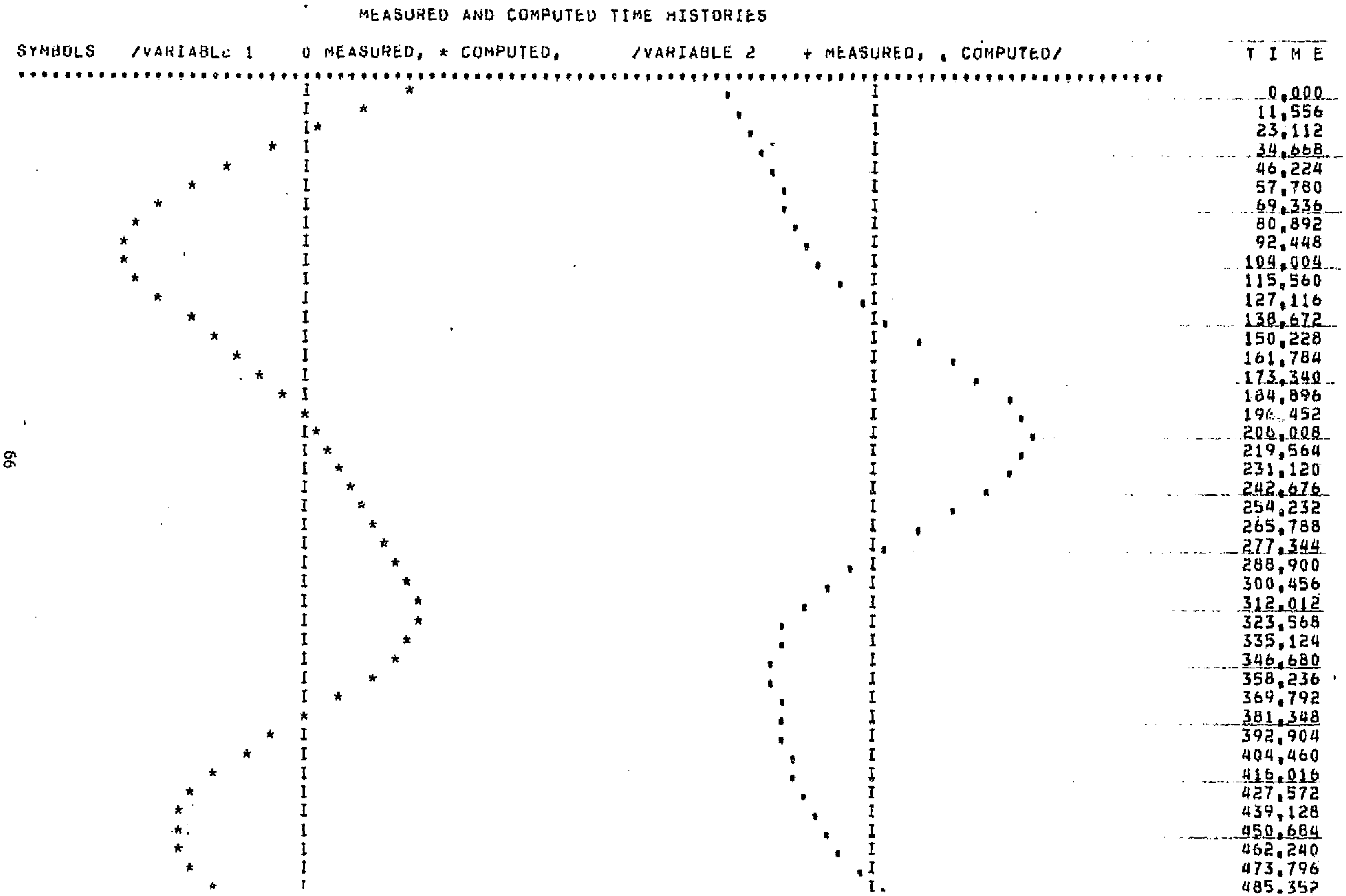
(e) Output.

Figure 1.- Continued.

```
NTAPE 1 NRUN 6      KDT=11556

VARIABLES STORED ARE IDENTIFIED BY A 1 IN COLUMNS 11 TO 22 AFTER THE COMMAND *STORAGE

*STORAGE  111100000000
  VARIABLE 1 IS STORED
  VARIABLE 2 IS STORED
  VARIABLE 3 IS STORED
  VARIABLE 4 IS STORED
END

SIMULATED RUN  1 6 COMPLETED AND STORED ON UNIT  7


 *GO PLOT


 **OUTPUT
```

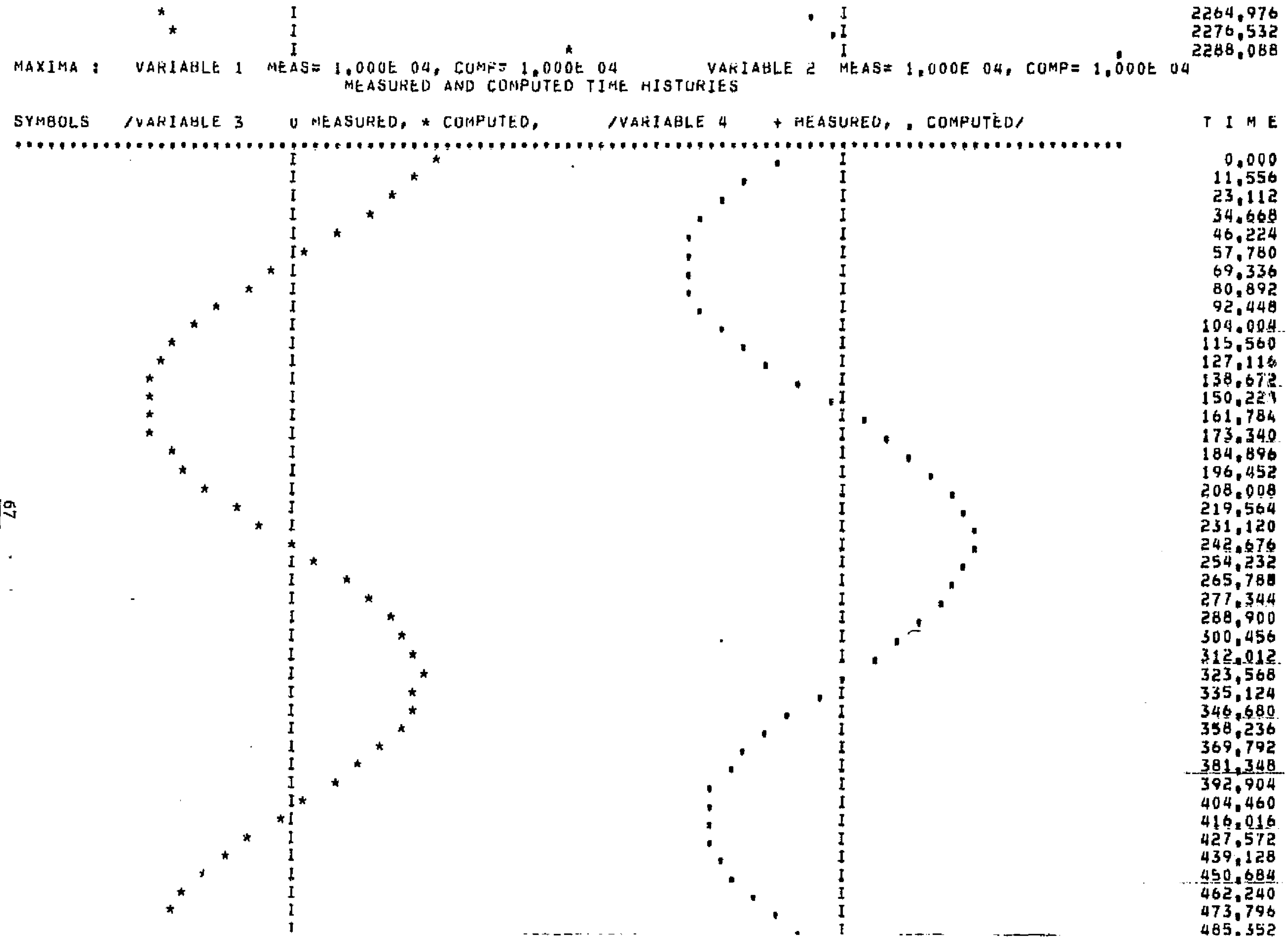
(f) Output.

Figure 1.- Continued.



(g) Output.

Figure 1.- Continued.



(h) Output.

Figure 1.- Continued.

```
          *         I                                 .       I                              2264.976
         *          I                                   .     I                              2276.532
                    I                 *                       I                      .       2288.088
MAXIMA :   VARIABLE 3  MEAS= 1.000E 04, COMP= 1.000E 04       VARIABLE 4  MEAS= 1.000E 04, COMP= 1.000E 04


 *GO PRIN
```


(i) Output.

Figure 1.- Continued.

	COMPUTED OUTPUT VALUES				MEASURED VALUES					COMPUTED VALUES					
PSI	MBETAX	MBETAY	ROLL A	PITCH A	MBETAX	MBETAY	ROLL A	PITCH A	SPMR	SPMP	BETA1	BETA2	BETA3	RDELT1	RDELT2
0.00	1469.2	-1921.8	2.652	-1.150	1469.2	-1921.8	2.652	-1.150	-50.7	12.1	1.912	1.400	0.592	-28.56	-51.90
11.56	917.5	-1780.4	2.319	-1.648	917.5	-1780.4	2.319	-1.648	-50.7	12.1	1.859	1.263	0.758	-37.68	-46.45
34.67	-378.6	-1472.2	1.416	-2.416	-378.6	-1472.2	1.416	-2.416	-50.7	12.1	1.754	0.948	1.156	-51.17	-29.22
46.22	-1019.9	-1345.2	0.885	-2.655	-1019.9	-1345.2	0.885	-2.655	-50.7	12.1	1.717	0.795	1.356	-54.65	-18.41
57.78	-1586.6	-1240.2	0.330	-2.782	-1586.6	-1240.2	0.330	-2.782	-50.7	12.1	1.691	0.663	1.536	-55.88	-6.77
69.34	-2034.2	-1144.4	-0.225	-2.790	-2034.2	-1144.4	-0.225	-2.790	-50.7	12.1	1.669	0.564	1.683	-54.90	5.18
80.89	-2330.9	-1037.0	-0.761	-2.679	-2330.9	-1037.0	-0.761	-2.679	-50.7	12.1	1.642	0.507	1.789	-51.83	16.91
92.45	-2461.2	-894.8	-1.255	-2.454	-2461.2	-894.8	-1.255	-2.454	-50.7	12.1	1.601	0.498	1.851	-46.84	27.85
104.00	-2427.4	-697.5	-1.692	-2.126	-2427.4	-697.5	-1.692	-2.126	-50.7	12.1	1.538	0.538	1.873	-40.15	37.47
115.56	-2248.8	-432.2	-2.054	-1.711	-2248.8	-432.2	-2.054	-1.711	-50.7	12.1	1.449	0.624	1.861	-32.03	45.34
127.12	-1958.9	-96.5	-2.328	-1.227	-1958.9	-96.5	-2.328	-1.227	-50.7	12.1	1.335	0.751	1.828	-22.77	51.09
138.67	-1598.5	299.7	-2.506	-0.697	-1598.5	299.7	-2.506	-0.697	-50.7	12.1	1.203	0.906	1.785	-12.69	54.48
150.23	-1209.6	734.6	-2.581	-0.145	-1209.6	734.6	-2.581	-0.145	-50.7	12.1	1.061	1.079	1.744	-2.13	55.38
161.78	-828.8	1176.1	-2.549	0.403	-828.8	1176.1	-2.549	0.403	-50.7	12.1	0.921	1.254	1.709	8.53	53.81
173.34	-482.6	1585.4	-2.411	0.923	-482.6	1585.4	-2.411	0.923	-50.7	12.1	0.795	1.418	1.683	18.88	49.89
184.90	-185.5	1922.1	-2.173	1.394	-185.5	1922.1	-2.173	1.394	-50.7	12.1	0.693	1.559	1.661	28.50	43.88
196.45	60.0	2149.8	-1.845	1.795	60.0	2149.8	-1.845	1.795	-50.7	12.1	0.627	1.668	1.635	36.98	36.11
208.01	261.0	2242.0	-1.439	2.111	261.0	2242.0	-1.439	2.111	-50.7	12.1	0.600	1.741	1.597	43.93	26.99
219.56	430.7	2186.0	-0.973	2.331	430.7	2186.0	-0.973	2.331	-50.7	12.1	0.618	1.778	1.541	49.01	16.95
231.12	585.4	1985.0	-0.466	2.449	585.4	1985.0	-0.466	2.449	-50.7	12.1	0.677	1.784	1.462	51.97	6.44
242.68	740.2	1658.5	0.059	2.460	740.2	1658.5	0.059	2.460	-50.7	12.1	0.774	1.768	1.361	52.64	-4.11
254.23	905.2	1238.5	0.579	2.368	905.2	1238.5	0.579	2.368	-50.7	12.1	0.900	1.739	1.241	50.94	-14.28
265.79	1083.0	765.1	1.070	2.177	1083.0	765.1	1.070	2.177	-50.7	12.1	1.044	1.707	1.111	46.94	-23.72
277.34	1266.1	280.3	1.511	1.897	1266.1	280.3	1.511	1.897	-50.7	12.1	1.196	1.678	0.982	40.80	-32.07
288.90	1437.8	-176.9	1.880	1.540	1437.8	-176.9	1.880	1.540	-50.7	12.1	1.343	1.654	0.863	32.80	-39.06
300.46	1574.4	-575.0	2.162	1.121	1574.4	-575.0	2.162	1.121	-50.7	12.1	1.473	1.632	0.766	23.35	-44.43
312.01	1649.3	-892.7	2.344	0.658	1649.3	-892.7	2.344	0.658	-50.7	12.1	1.579	1.607	0.700	12.91	-47.99
323.57	1638.3	-1120.2	2.419	0.168	1638.3	-1120.2	2.419	0.168	-50.7	12.1	1.654	1.571	0.670	2.00	-49.60
335.12	1524.7	-1258.7	2.384	-0.328	1524.7	-1258.7	2.384	-0.328	-50.7	12.1	1.698	1.518	0.679	-8.67	-49.17
346.68	1303.0	-1318.8	2.243	-0.809	1303.0	-1318.8	2.243	-0.809	-50.7	12.1	1.714	1.444	0.727	-19.17	-46.71
358.24	980.2	-1318.2	2.004	-1.255	980.2	-1318.2	2.004	-1.255	-50.7	12.1	1.708	1.349	0.810	-28.47	-42.27
369.79	576.4	-1277.4	1.678	-1.648	576.4	-1277.4	1.678	-1.648	-50.7	12.1	1.688	1.238	0.921	-36.37	-36.00
381.35	121.6	-1215.5	1.282	-1.970	121.6	-1215.5	1.282	-1.970	-50.7	12.1	1.663	1.117	1.050	-42.56	-28.12
392.90	-347.5	-1146.1	0.833	-2.206	-347.5	-1146.1	0.833	-2.206	-50.7	12.1	1.637	0.996	1.187	-46.84	-18.92
404.46	-791.6	-1074.3	0.352	-2.345	-791.6	-1074.3	0.352	-2.345	-50.7	12.1	1.614	0.885	1.320	-49.08	-8.79
416.02	-1174.1	-997.0	-0.141	-2.380	-1174.1	-997.0	-0.141	-2.380	-50.7	12.1	1.592	0.794	1.440	-49.24	1.84
427.57	-1464.7	-903.7	-0.626	-2.308	-1464.7	-903.7	-0.626	-2.308	-50.7	12.1	1.566	0.732	1.538	-47.36	12.47
439.13	-1643.3	-780.6	-1.082	-2.132	-1643.3	-780.6	-1.082	-2.132	-50.7	12.1	1.529	0.705	1.609	-43.55	22.60
450.68	-1702.0	-614.6	-1.491	-1.860	-1702.0	-614.6	-1.491	-1.860	-50.7	12.1	1.477	0.716	1.652	-38.00	31.75
462.24	-1645.8	-396.6	-1.837	-1.502	-1645.8	-396.6	-1.837	-1.502	-50.7	12.1	1.404	0.763	1.668	-30.93	39.48
473.79	-1491.6	-124.8	-2.106	-1.076	-1491.6	-124.8	-2.106	-1.076	-50.7	12.1	1.312	0.842	1.663	-22.62	45.42
485.35	-1265.1	193.7	-2.286	-0.601	-1265.1	193.7	-2.286	-0.601	-50.7	12.1	1.204	0.948	1.644	-13.36	49.29
496.91	-996.3	542.8	-2.371	-0.098	-996.3	542.8	-2.371	-0.098	-50.7	12.1	1.086	1.071	1.619	-3.49	50.93
508.46	-714.1	898.3	-2.355	0.410	-714.1	898.3	-2.355	0.410	-50.7	12.1	0.969	1.201	1.594	6.63	50.28

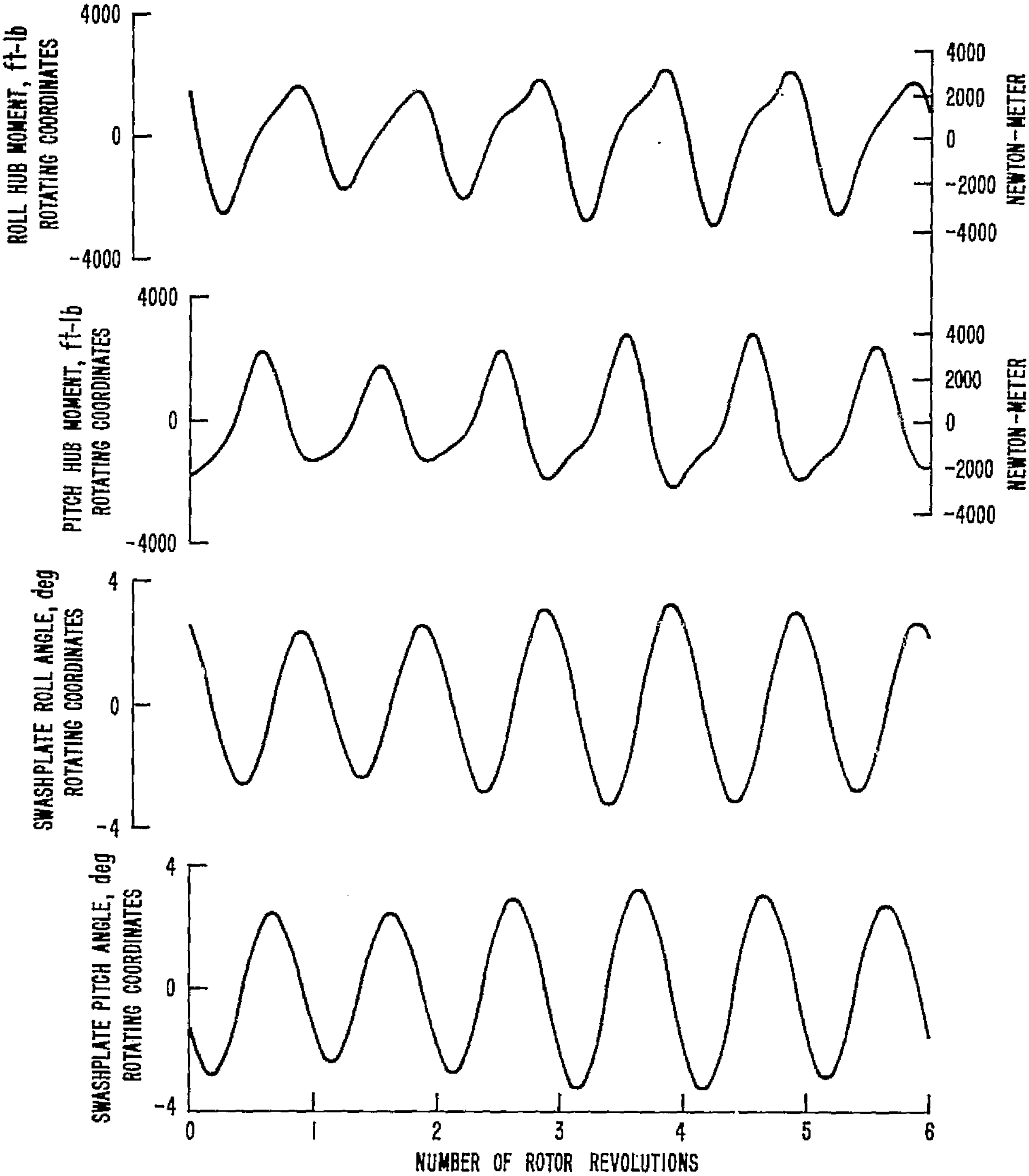
(j) Output.

Figure 1.- Continued.

```
2287.89   4000.0   4000.0   5.000   5.000   4000.0   4000.0   5.000   5.000  -50.7   12.1  1.153  0.931  1.687 -14.30  53.55

 *********

***END OF IDENTIFICATION RUN NUMBER 1 , TOTAL COMPUTATION TIME 44.10 SEC. ***
*RETURN
```


(k) Output.

Figure 1.- Continued.



(1) Math model simulation without parameter identification.

Figure 1.- Concluded.

```
   LOGON FSTGK,MAINPR,D240;FST19 T4947 'GERD KANNING STOP 17 PH 5455'
DDEF FT10F001,VS,IDENT,R0026
DDEF FT20F001,VS,SFCT01,RUN6
CALL MAIN2SS
-1 0 0 0 0 0 0
+.40   +00+.268  +03+.2684 +03+.165  +02+.117  +01+.97   +00
+.15   +01+.1926 +03+.188  +01+.4566 +01+.00   +00+.12021+06+.145  +01
-.943  +01+.225  +01+.0    +00+.70   +02+.2246 -02
+.0    +00+.0    +00-.2428 +01+.11296+01+.10   -01
+.1912 +01-.4666 +01+.14   +01-.12535+02+.592  +00+.1481 +02+.857  +01
+.3    +00+.15   +00+.60   +02+.00   +00+.10200+05+.87   +00
+.00   +00+.70   +01+.80   +02
+.2652 +01-.2856 +02-.115  +01-.519  +02-.507  +02+.121  +02
+.1768 +01          -.252  +01
+.5778 +01+.00   +00+.00   +00+.2    +01
+.10   +02+.10   +02+.10   -01+.10   -01+.1    -01+.1    -01+.1    -01

HELICOPTER ROTOR IDENTIFICATION FROM REAL DATA (TEST 366 RUN 19 RECORD 6)  TSS
   21   64    4    2    210000    0000000
0*+.                             2                         .005              6    1
011100000000001100001
+.268  +03+.97   +00+.857  +01+.12021+06+.4566 +01+.15   +01    C
+.87   +00-.699  -02-.10   -04+.1    -04+.188  +01+.225  +01    C
+.1    -01+.355  +00+.80   +02+.7    +01+.4    +00+.5296 +02    C
+.3    +00+.15   +00-.943  +01
+.26532+03+.95   +00+.797  +01+.63900+05+.4    +01+.1395 +01    C
+.844  +00-.421  -01-.321  -01-.2    +00+.187  +01+.2140 +01    C
+.1    -02+.3479 +00+.24   +02+.2    +01+.32   +00+.5190 +02    C
+.29   +00+.147  +00-.896  +01
+.27068+03+.989  +00+.917  +01+.16829+06+.5023 +01+.1605 +01    C
+.896  +00+.1    +00+.2    +00+.41   -01+.25   +01+.236  +01    C
+.3    +01+.3621 +00+.136  +03+.14   +02+.48   +00+.5402 +02    C
+.31   +00+.153  +00-.990  +01
BIY ,LOSS,CLEC,BETA,LOCK,SWEP,  C2,LAM ,LAMX,LAMY,BRDO,PCON,
  CD,  MU,CDMP,FDMP, BIX,GSPD,GYIP,GYID,TWST
+.1599 +00+.1722 +00+.40000-03+.3959 -03+.1599 +00+.1722 +00    C
 FT-LBS  FT-LBS  DEGREE  DEGREE
+.8797 -02+.4493 -02+.1599 +00+.1722 +00+.4    -03+.3959 -03
 FT-LB   FT-LB
SLUG.FT2(NO DIM)DEG      FTLB/RAD(NO DIM)DEG      (NO DIM)(NO DIM)(NO DIM)(NO DIM)
FT       DEG      (NO DIM)(NO DIM)FTLB/R.SFTLB/R.SSLUG.FT2(NO DIM)SLUG.FT2SLUG.FT2
DEG
DSET20
*RELVAR
+.14   +00+.21   -01+24    -01+.1    +00+.11   +00+.5    +00    C
```

(a) List of input data cards.

Figure 2.- Parameter identification run.

```
+.69     -01+.48    -01+.83    -01+.54    -01+.5     +00+.44    +00     C
+.5      +00+.22    +00+.3     +00+.36    +00+.5     +00+.1     -01     C
+.4      +00+.1     -01+.84    -01
*READWGH
+.10     +01+.10    +01+.10    +01+.1     +01+.1     -03+.1     -06
**********
*GO PLOT
**OUTPUT 1     2     3     4
**RESAMP 1     5
**RESAMP 2     6
**RESAMP 3     7
**RESAMP 4     8
**RESCOR 1     9
**RESCOR 2    10
**RESCOR 3    11
**RESCOR 4    12
*GO STORE
DSET10  0106
*STORAGE  111111111111
*GO PRINT
*********
*RETURN
  LOGOFF
```

(b) List of input data cards.

Figure 2.- Continued.

```
TWO DEGREE OF FREEDOM HELICOPTER BLADE AND GYRO SIMULATION

PROGRAM CONTROL LOGIC
IFLOQ=-1    IHINGE= 0    IROT= 0    INEW= 0 ISIML= 0 IFLOW= 0 IREC= 0

BLADE PARAMETERS
BIX=  0.40 BIY=268.00 BIZ=268.40 RADIUS=16.5 CHORD=1.17 BLADE LOSS=.97

SWEEP=  1.50 RPM= 192.6 HINGE POSITION=1.88 LOCK= 4.566 DELTA3= 0.00 K BETA=0.120210E 06 PI=  1.45

TWIST= -9.43 PRECONE= 2.250  CT= 0.0000 AIR VELOCITY=  70.00 AIR DENSITY=.002246 ALPHA= 0.000 ALPHAS= 0.000

LITTLE A1S= 0.000 LITTLE B1S= 0.000 BIG A1S=-2.428 BIG B1S= 1.130 CD=0.010 LAM=0.000000 LAMX=0.000000 LAMY=0.000000

BLADE INITIAL CONDITIONS
BETA1=  1.912 BETA1 RATE=  -4.67 BETA2=  1.400 BETA2 RATE= -12.53 BETA3=  0.592 BETA3 RATE=  14.81 COLLECTIVE=  8.6

GYRO PARAMETERS
IP= 0.300 ID= 0.150 GYRO ANGLE= 60.0 GYRO DAMP L=  0.00 GYRO RPM=10200.0 MECHANICAL ADVANTAGE C2=0.870

FEATHERING RESTRAINT KZETA=  0.00 FEATHERING DAMPING CZETA=  7.00 SWASHPLA[illegible] DAMPING CSDAMP= 80.00 PSIUF=  0.00

INITIAL CONDITIONS ON GYRO
DELTA1=  2.652 DELTA1 RATE=  -28.56 DELTA2= -1.150 DELTA2 RATE=  -51.90 NU1=-50.70 NU2= 12.10

PHI=  1.768 RPHI=    0.00 THET= -2.520 RTHET=    0.00 RMBIAS=    0.00 PMBIAS=    0.00 RABIAS= 0.000 PABIAS= 0.000

INITIAL CONDITIONS FOR STARTING AND STOPPING THE INTEGRATION ROUTINE
DELPSI= 5.78  REVS= 0.00  REVO= 0.00  REVF= 2.00 TRUNC=0.0000

SCALE FACTORS FOR PLOTTING PROGRAM
SCALE(1)     SCALE(2)     SCALE(3)     SCALE(4)     SCALE(5)     SCALE(6)     SCALE(7)
1.000E 01    1.000E 01    1.000E-02    1.000E-02    1.000E-02    1.000E-02    1.000E-02

SCALE(8)     SCALE(9)     SCALE(10)    SCALE(11)    SCALE(12)    SCALE(13)    SCALE(14)
0.000        0.000        0.000        0.000        0.000        0.000        0.000
```

(c) Output.

Figure 2.- Continued.

PARAMETERS CALULATED FROM INPUT DATA
OMEGA= 20.169 GYSPED= 52.96 MU= 0.355 LAMBDA=-.006999

DELTAT=0.5000E-02 TIMES=0.0000 TIMEO=0.0000 TIMEF=0.6231E 00
HELICOPTER ROTOR IDENTIFICATION FROM REAL DATA (TEST 366 RUN 19 RECORD 6) TSS

DATA CARDS LISTING*****RUN NUMBER 1= 6
X AXIS INCREMENT KDT= 11556 SKIP= 2 64 POINTS PER CHANNEL
THE 9 FOLLOWING CHANNELS WERE READ : 1, 2, 3, 4, 5, 6,11,12,14,

(d) Output.

Figure 2.- Continued.

```
HELICOPTER ROTOR IDENTIFICATION FROM REAL DATA (TEST 366 RUN 19 RECORD 6)  TSS
------------------------------------------------------------------------------

 21 PARAMETERS        4 OUTPUTS

 64 MEASURED VALUES PER OUTPUT    ( 256 MEASUREMENTS),        2 WAS THE SKIPPING FACTOR USED

    TIME INTERVAL BETWEEN MEASUREMENTS  0,0050 SEC,           2 STEPS OF INTEGRATION IN EACH INTERVAL

   ANGLE INCREMENT BETWEEN MEASUREMENTS      5,78 DEG

INITIAL CONDITIONS
  4,629E-02 -4,985E-01 -2,007E-02 -9,058E-01  3,337E-02 -8,144E-02  2,443E-02 -2,188E-01  1,033E-02  2,585E-01

SCALING FACTORS FOR DATA UNITS
1DTU=  1,599E-01 FT-LBS  /  1,722E-01 FT-LBS  /  4,000E-04 DEGREE  /  3,959E-04 DEGREE  /  1,599E-01          /  1,722E-01          /
 8,797E-03 FT-LB   /  4,493E-03 FT-LB   /  1,599E-01          /  1,722E-01          /  4,000E-04          /  3,959E-04          /

PARAMETERS NAMES
    1) BIY     2) LOSS     3) CLEC     4) BETA     5) LOCK     6) SWEP     7)   C2     8) LAM      9) LAMX    10) LAMY    11) BRDO    12) PCON

   13)   CO   14)    MU   15) CDMP    16) FDMP    17)  BIX    18) GSPD    19) GYIP    20) GYID    21) TWST

INITIAL VALUES OF THE PARAMETERS
  2,680E 02  9,700E-01  9,570E 00  1,202E 05  4,566E 00  1,500E 00  8,700E-01 -6,990E-03 -1,000E-05  1,000E-05  1,880E 00  2,250E 00

  1,000E-02  3,550E-01  8,000E 01  7,000E 00  4,000E-01  5,296E 01  3,000E-01  1,500E-01 -9,430E 00
LOWER BOUNDS
  2,653E 02  9,500E-01  7,970E 00  6,390E 04  4,000E 00  1,395E 00  8,440E-01 -4,210E-02 -3,210E-02 -2,000E-01  1,870E 00  2,140E 00

  1,000E-03  3,479E-01  2,400E 01  2,000E 00  3,200E-01  5,190E 01  2,900E-01  1,470E-01 -8,960E 00
UPPER BOUNDS
  2,707E 02  9,890E-01  9,170E 00  1,683E 05  5,023E 00  1,605E 00  8,960E-01  1,000E-01  2,000E-01  4,100E-02  2,500E 00  2,360E 00

  3,000E 00  3,621E-01  1,360E 02  1,400E 01  4,800E-01  5,402E 01  3,100E-01  1,530E-01 -9,900E 00

THE IDENTIFICATION PROCEDURE WILL USE A MAXIMUM OF   2 ITERATIONS AND A SEPARATION THRESHOLD EQUAL TO 0,50E-02

***THE FOLLOWING WEIGHTS ARE USED***
      1,000E 00           1,000E 00           1,000E 00           1,000E 00
ERRMIN =  1,000E-04  ,    DMIN =  0,000

INITIAL PARAMETER INCREMENTS FOR GRADIENT EVALUATION
  1,400E-01  2,100E-02  2,400E 01  1,000E-01  1,100E-01  5,000E-01  6,900E-02  4,800E-02  8,300E-02  5,400E-02  5,000E-01  4,400E-01
  5,000E-01  2,200E-01  3,000E-01  3,600E-01  5,000E-01  1,000E-02  4,000E-01  1,000E-02  8,400E-02
                                  +,10    +01+,10   +01+,10    +01+,1    +01+,1    -03+,1    -06
```

(e) Output.

Figure 2.- Continued.

```
                      **********

ITERATION  0.    MODE WAS99.  KND= 2                                                                  REL.ERROR = 1.31E 00    COST = 2.435E 07
-----------------------------------P  A  R  A  M  E  T  E  R  S        V  A  L  U  E  S---------------------------------------------
  1 13 25     2 14 26     3 15 27     4 16 28     5 17 29     6 18 30     7 19 31     8 20 32     9 21 33    10 22 34    11 23 35    12 24 36
  2,680E 02  9,700E-01  8,570E 00  1,202E 05  4,566E 00  1,500E 00  8,700E-01 -6,990E-03 -1,000E-05  1,000E-05  1,880E 00  2,250E 00

  1,000E-02  3,550E-01  8,000E 01  7,000E 00  4,000E-01  5,296E 01  3,000E-01  1,500E-01 -9,430E 00
INCREMENT IN PARAMETER # 3 REDUCED TO 242,2215 %
INCREMENT IN PARAMETER # 3 REDUCED TO  26,4437 %
INCREMENT IN PARAMETER # 3 REDUCED TO   4,8659 %
```


(f) Output.

Figure 2.- Continued.

BASIC PARAMETER TWST	SEPARATION	CRITICAL PARAMETER	SEPARATION
BETA	9.49E-01	CLEC	1.27E-02
COMP	8.13E-01	CLEC	1.26E-02
LOSS	5.95E-01	CLEC	1.10E-02
FDMP	4.17E-01	CLEC	9.95E-03
CLEC	9.86E-03	CLEC	9.86E-03

(g) Output.

Figure 2.- Continued.

```
***THERE ARE 21 PARAMETERS IN THIS PROBLEM ,         THE SEPARATION THRESHOLD WAS 0.50E-02
MAGIC NUMBER  0 2 2 2 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 2


*  INDEPENDENT PARAMETERS
*  LOSS,CLEC,BETA,CDMP,FDMP,TWST,
*
*  IRRELEVANT PARAMETERS
*  NONE,
*
*  NOT ESTIMATED
*  NONE,
*
*  NOT USED
*  BIY ,LOCK,SWEP,  C2,LAM ,LAMX,LAMY,BRDO,PCON,  CD,  MU, BIX,GSPD,GYIP,GYID,


DET= 0.355861616827987D-05   TIME =    0.535 SEC
STATUS  1NU  2    3    4    5NU  6NU  7NU  8NU  9NU 10NU 11NU 12NU 13NU 14NU 15   16   17NU 18NU 19NU 20NU 21
SENS = 0.00 0.09 0.20 0.09 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.03 0.06 0.00 0.00 0.00 0.00 0.27
START  0,  FIRST 21 (TWST),  LAST  3 (CLEC)      SEP =  0.99E-02                     TIME =   84.50 SEC

ITERATION  1.    MODE WAS 0.  KND= 0                                                  REL.ERROR = 1.11E 00    COST = 1.825E 07
-------------------------------------P  A  R  A  M  E  T  E  R  S      V  A  L  U  E  S------------------------------------------
 1 13 25    2 14 26    3 15 27    4 16 28    5 17 29    6 18 30    7 19 31    8 20 32    9 21 33   10 22 34   11 23 35   12 24 36
 2.680E 02  9.500E-01  9.170E 00  7.810E 04  4.566E 00  1.500E 00  8.700E-01 -6.990E-03 -1.000E-05  1.000E-35  1.880E 00  2.250E 00

 1.000E-02  3.550E-01  1.360E 02  2.000E 00  4.000E-01  5.296E 01  3.000E-01  1.500E-01 -9.900E 00
```

(h) Output.

Figure 2.- Continued.

BASIC PARAMETER TWST	SEPARATION	CRITICAL PARAMETER	SEPARATION
BETA	8,89E-01	CLEC	1,02E-02
CDMP	6,73E-01	CLEC	1,02E-02
LOSS	5,17E-01	CLEC	9,61E-03
FDMP	1,70E-01	CLEC	7,08E-03
CLEC	6,89E-03	CLEC	6,89E-03

(i) Output.

Figure 2.- Continued.

```
***THERE ARE 21 PARAMETERS IN THIS PROBLEM ,          THE SEPARATION THRESHOLD WAS 0.50E-02
MAGIC NUMBER  0 2 2 2 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 2

   *  INDEPENDENT PARAMETERS
   *  LOSS,CLEC,BETA,CDMP,FDMP,TWST,
   *
   *  IRRELEVANT PARAMETERS
   *  NONE,
   *
   *  NOT ESTIMATED
   *  NONE,
   *
   *  NOT USED
   *  BIY ,LOCK,SWEP,  C2,LAM ,LAMX,LAMY,BRDO,PCON,  CD,  MU, BIX,GSPD,GYIP,GYID,


DET= 0.130577732983201D-06   TIME =     0.531 SEC
STATUS  1NU  2    3    4    5NU  6NU  7NU  8NU  9NU 10NU 11NU 12NU 13NU 14NU 15   16   17NU 18NU 19NU 20NU 21
SENS = 0.00 0.16 0.16 0.10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.15 0.04 0.00 0.00 0.00 0.00 0.17
START  0,  FIRST 21 (TWST),  LAST  3 (CLEC)      SEP =  0.69E-02                         TIME =   59.26 SEC

ITERATION  2,     MODE WAS 0,  KND= 0                                                   REL.ERROR = 1.06E 00     COST = 1.713E 07
-------------------------------------P  A  R  A  M  E  T  E  R  S        V  A  L  U  E  S-----------------------------------------
  1 13 25     2 14 26     3 15 27     4 16 28     5 17 29     6 18 30     7 19 31     8 20 32     9 21 33    10 22 34    11 23 35    12 24 36
  2.680E 02  9.500E-01  9.170E 00  6.390E 04  4.566E 00  1.500E 00  8.700E-01 -6.990E-03 -1.000E-05  1.000E-05  1.880E 00  2.250E 00

  1.000E-02  3.550E-01  1.360E 02  2.000E 00  4.000E-01  5.276E 01  3.000E-01  1.500E-01 -9.900E 00


PARAMETER INCREMENTS (DA/A) FOR GRADIENT EVALUATION

  BIY =  1.4E-01  LOSS=  2.3E-02  CLEC=  1.8E-02  BETA=  1.1E-01  LOCK=  1.1E-01  SWEP=  5.0E-01
   C2=  6.9E-02  LAM =  4.8E-02  LAMX=  8.3E-02  LAMY=  5.4E-02  BRDO=  5.0E-01  PCON=  4.4E-01
   CD=  5.0E-01    MU=  2.2E-01  CDMP=  3.6E-01  FDMP=  5.0E-01   BIX=  5.0E-01  GSPD=  1.0E-02
  GYIP=  4.0E-01  GYID=  1.0E-02  TWST=  2.3E-02
```

(j) Output.

Figure 2.- Continued.

FINAL RESULTS OF THIS IDENTIFICATION

FIT TEST ** COST = 0.1713E 08 CORRELATION =-0.12500

RELATIVE ERROR = 1.0607

ESTIMATED VALUES OF THE PARAMETERS

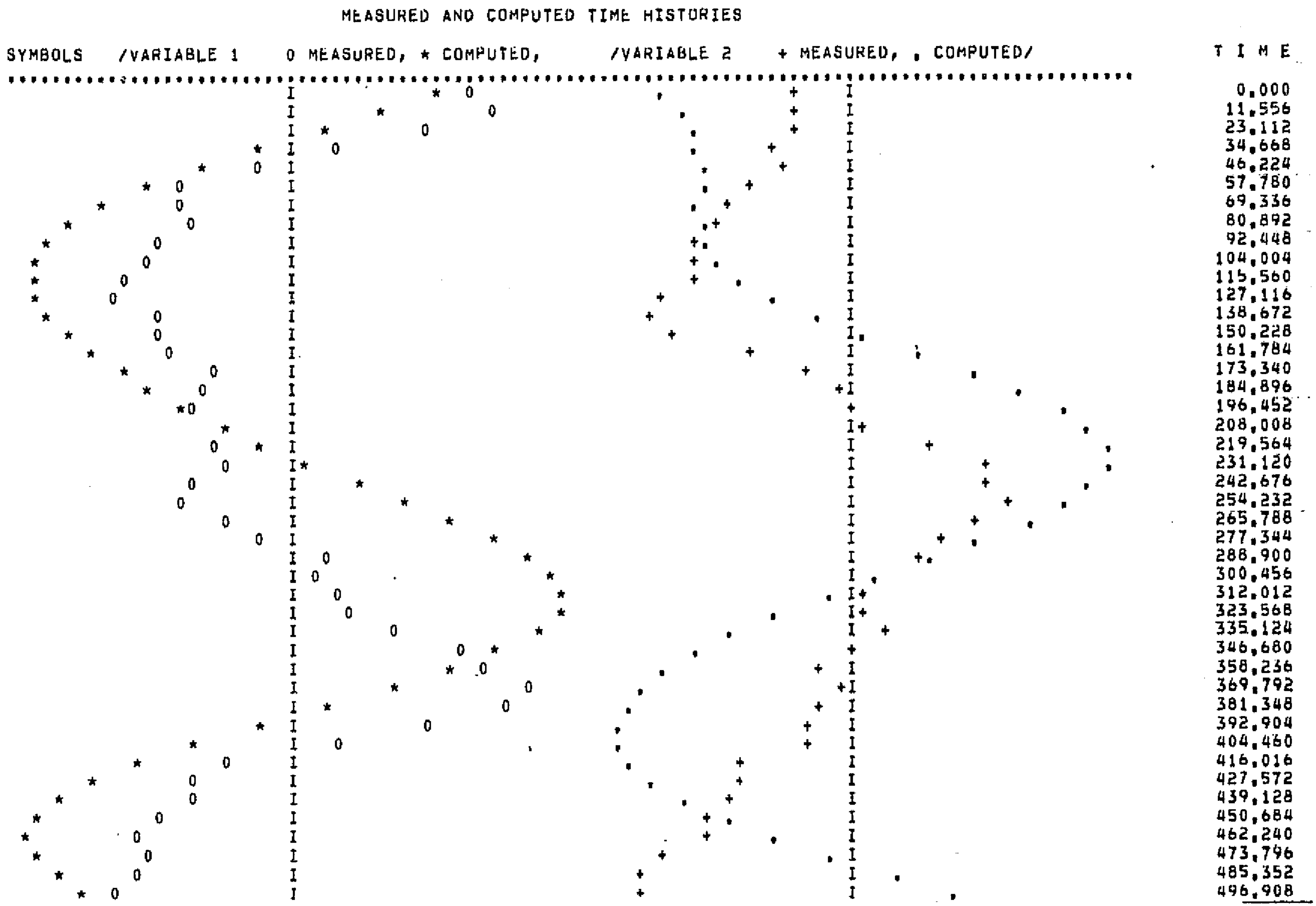
	NAME		EST.VALUE	ERROR BOUNDS		UNITS	SENSITIVITY	DEPENDENCY INDEX FINAL	DEPENDENCY INDEX INITIAL
1	BIY	=	2.6800E 02	+/-	0.0	SLUG.FT2	0.00000	0	0
2	LOSS	=	9.5000E-01	+/-	1.1E-01	(NO DIM)	3.21449	2	1
3	CLEC	=	9.1700E 00	+/-	4.1E 01	DEG	4.26869	2	1
4	BETA	=	6.3900E 04	+/-	3.4E 04	FTLB/RAD	0.33903	2	1
5	LOCK	=	4.5660E 00	+/-	0.0	(NO DIM)	0.00000	0	0
6	SWEP	=	1.5000E 00	+/-	0.0	DEG	0.00000	0	0
7	C2	=	8.7000E-01	+/-	0.0	(NO DIM)	0.00000	0	0
8	LAM	=	-6.9900E-03	+/-	0.0	(NO DIM)	0.00000	0	0
9	LAMX	=	-1.0000E-05	+/-	0.0	(NO DIM)	0.00000	0	0
10	LAMY	=	1.0000E-05	+/-	0.0	(NO DIM)	0.00000	0	0
11	BRDO	=	1.8800E 00	+/-	0.0	FT	0.00000	0	0
12	PCON	=	2.2500E 00	+/-	0.0	DEG	0.00000	0	0
13	CD	=	1.0000E-02	+/-	0.0	(NO DIM)	0.00000	0	0
14	MU	=	3.5500E-01	+/-	0.0	(NO DIM)	0.00000	0	0
15	CDMP	=	1.3600E 02	+/-	5.7E 02	FTLB/R.S	0.19451	2	1
16	FDMP	=	2.0000E 00	+/-	3.1E 01	FTLB/R.S	0.05120	2	1
17	BIX	=	4.0000E-01	+/-	0.0	SLUG.FT2	0.00000	0	0
18	GSPD	=	5.2960E 01	+/-	0.0	(NO DIM)	0.00000	0	0
19	GYIP	=	3.0000E-01	+/-	0.0	SLUG.FT2	0.00000	0	0
20	GYID	=	1.5000E-01	+/-	0.0	SLUG.FT2	0.00000	0	0
21	TWST	=	-9.9000E 00	+/-	5.8E 01	DEG	3.29363	2	1

*GO PLOT

**OUTPUT

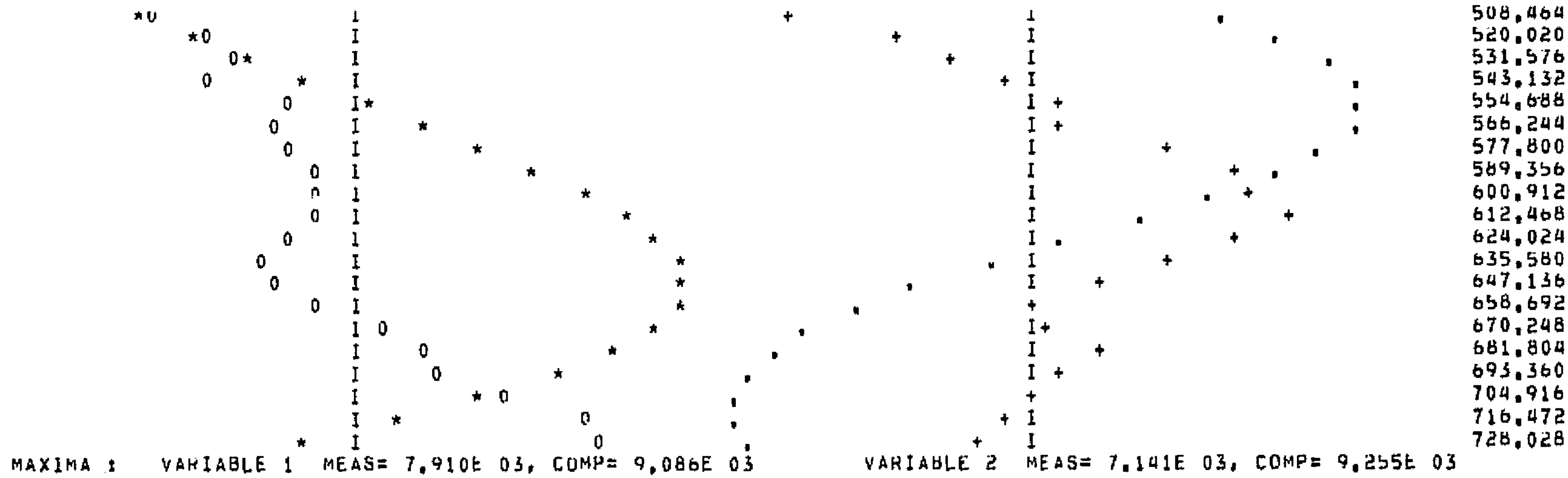
(k) Output.

Figure 2.- Continued.



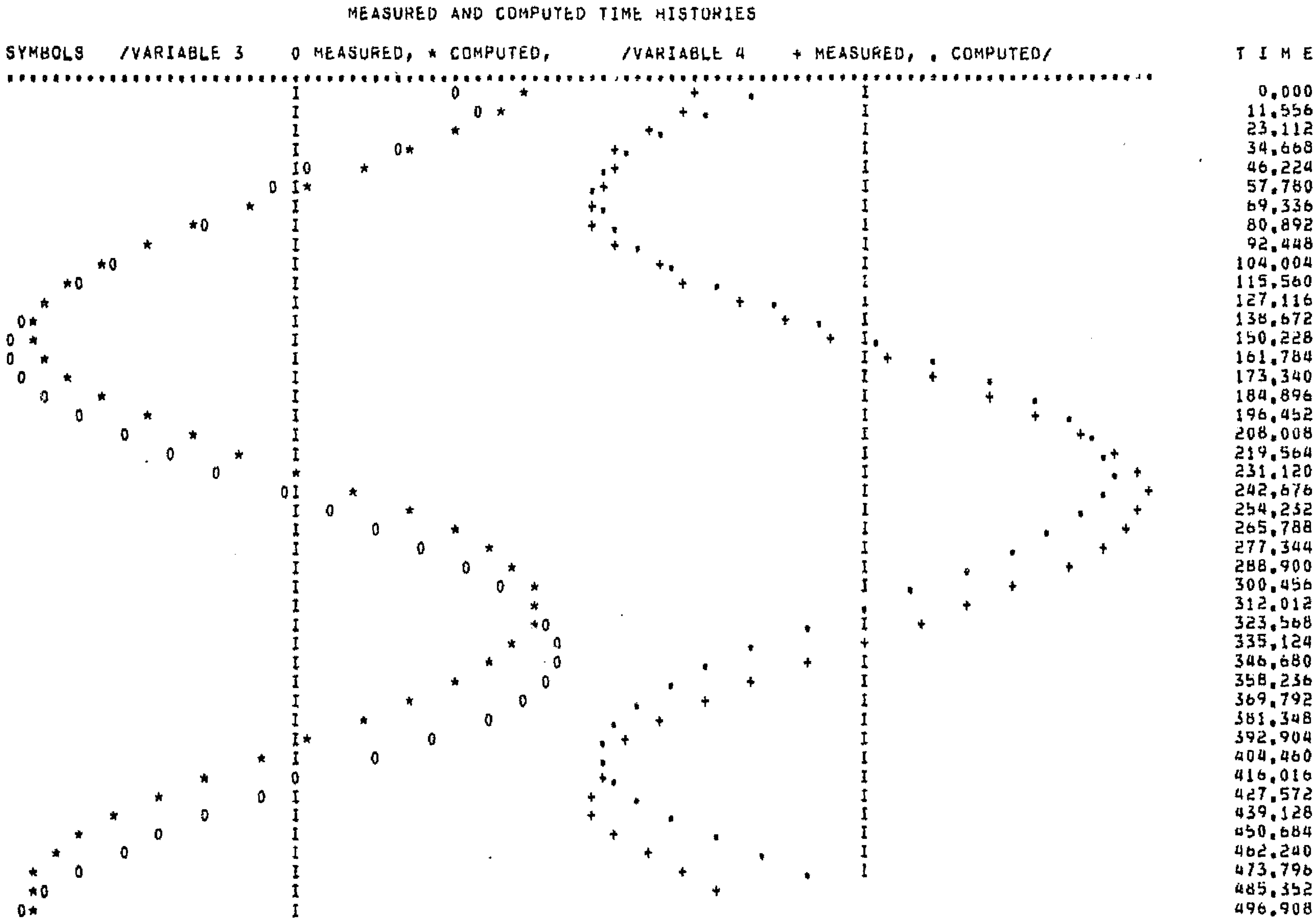
(1) Output.

Figure 2.- Continued.



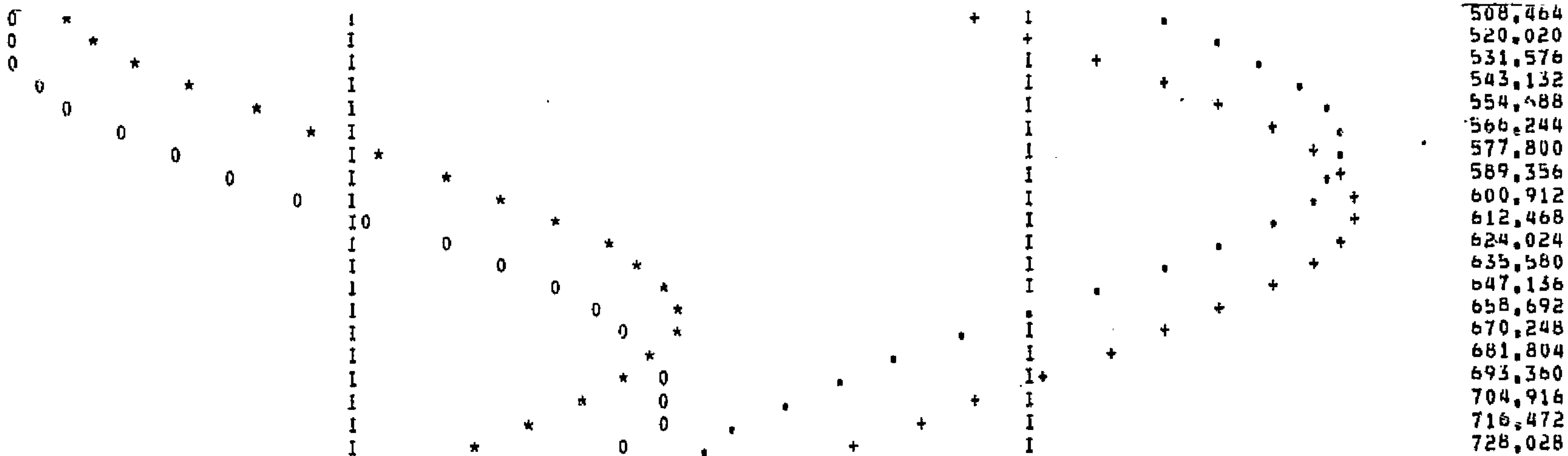
(m) Output.

Figure 2.- Continued.



(n) Output.

Figure 2.- Continued

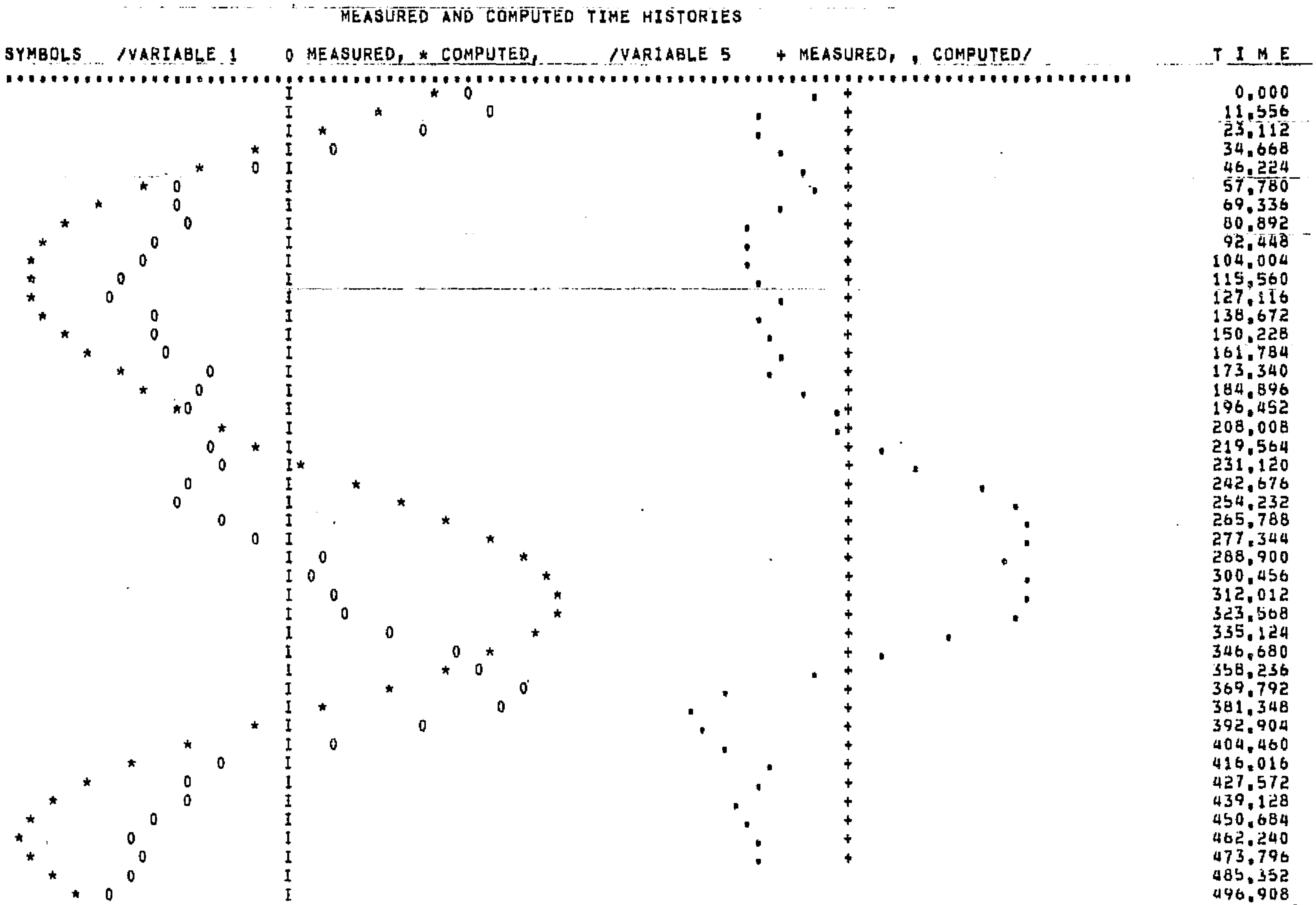


MAXIMA : VARIABLE 3 MEAS= 8,000E 03, COMP= 7,828E 03 VARIABLE 4 MEAS= 7,989E 03, COMP= 7,603E 03

**RESAMP

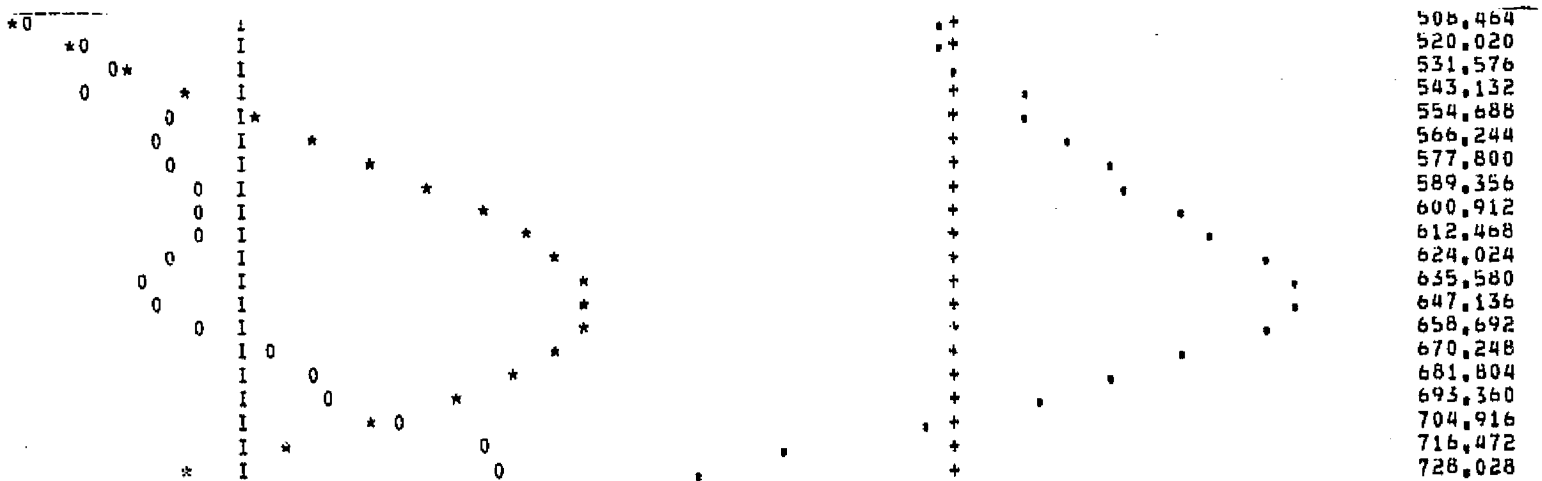
(o) Output.

Figure 2.- Continued.



(p) Output.

Figure 2.- Continued.

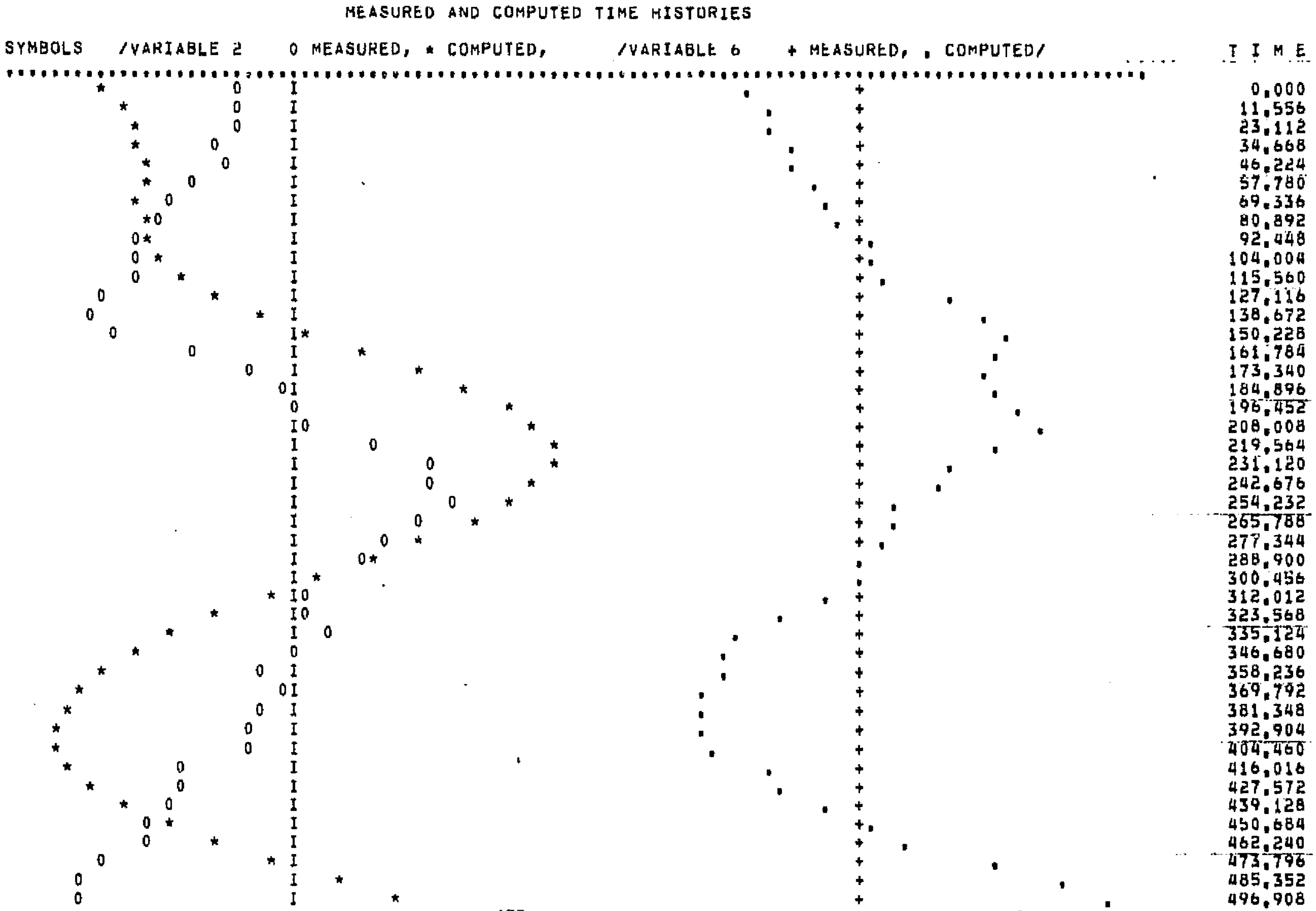


MAXIMA 1   VARIABLE 1 MEAS= 7,910E 03, COMP= 9,086E 03   VARIABLE 5 MEAS= 1,000E-50, COMP= 1,125E 04

**RESAMP

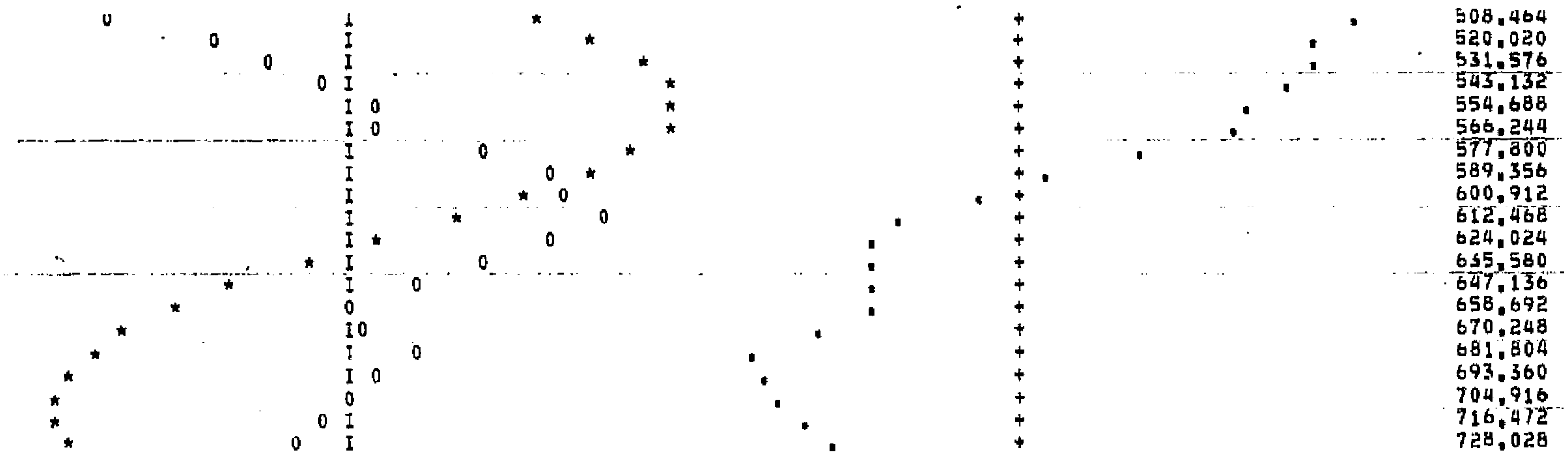
(q) Output.

Figure 2.- Continued.



(r) Output.

Figure 2.- Continued.

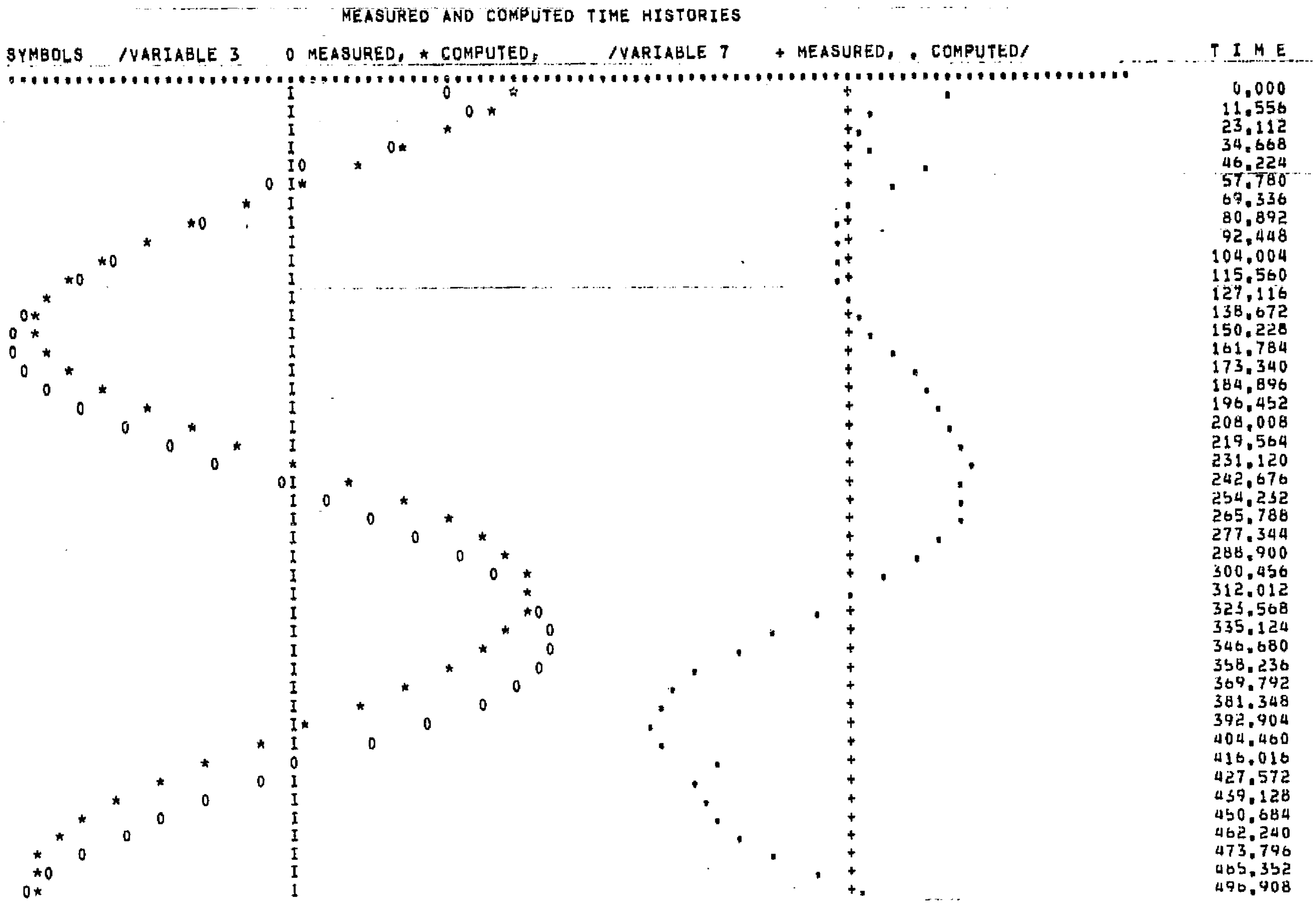


MAXIMA :   VARIABLE 2 MEAS= 7,141E 03, COMP= 9,255E 03   VARIABLE 6 MEAS= 1,000E-50, COMP= 1,172E 04

**RESAMP

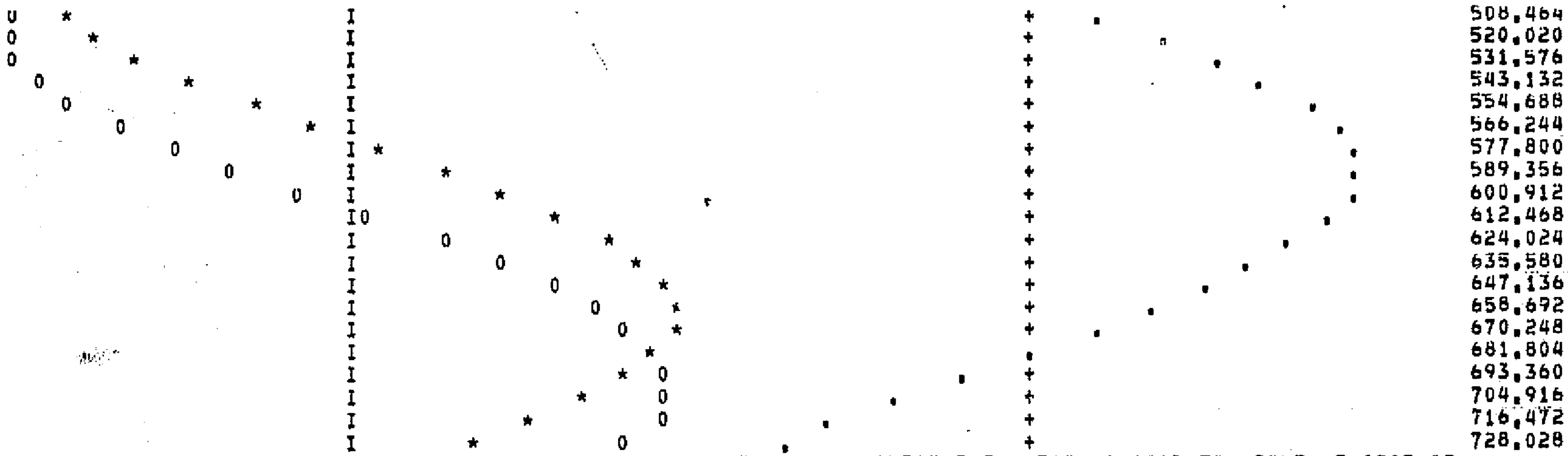
(s) Output.

Figure 2.- Continued.



(t) Output.

Figure 2.- Continued.



MAXIMA : VARIABLE 3 MEAS= 8.000E 03, COMP= 7.828E 03 VARIABLE 7 MEAS= 1.000E-50, COMP= 5.080E 03

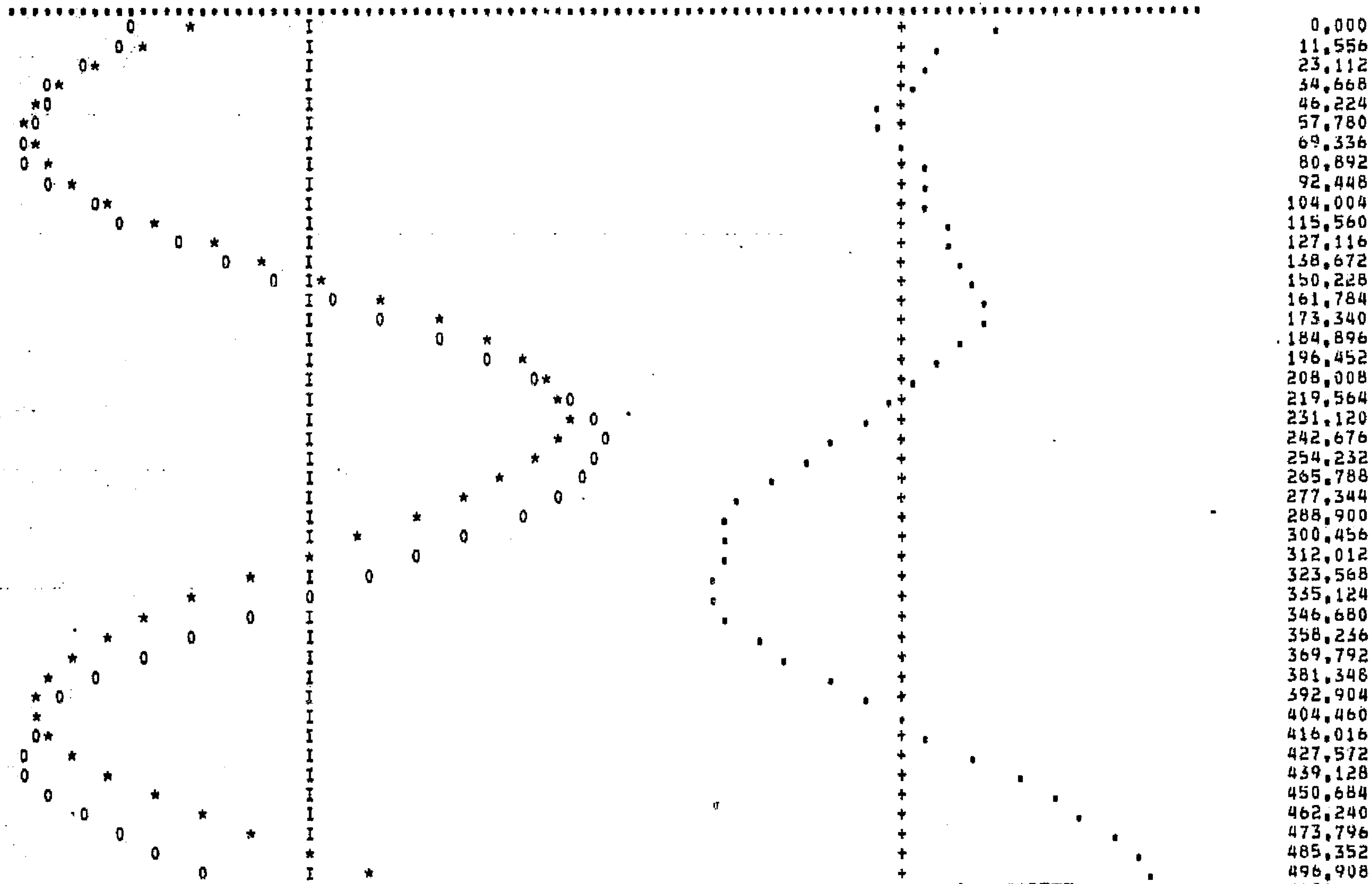
**RESAMP

(u) Output.

Figure 2.- Continued.

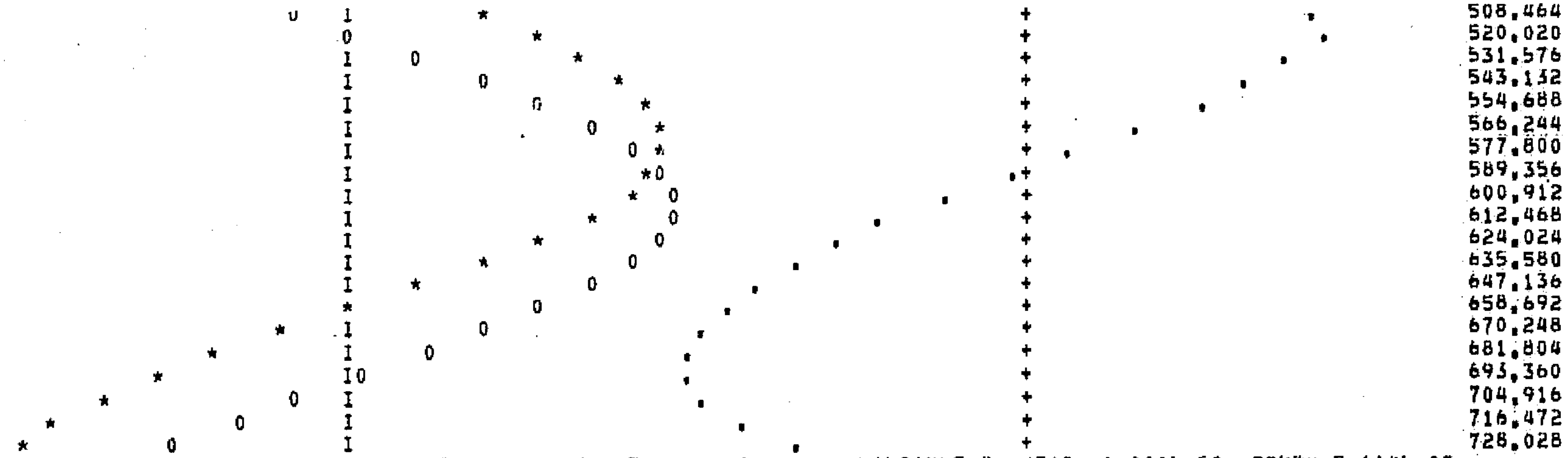
MEASURED AND COMPUTED TIME HISTORIES

SYMBOLS /VARIABLE 4 0 MEASURED, * COMPUTED, /VARIABLE 8 + MEASURED, . COMPUTED/ TIME



(v) Output.

Figure 2.- Continued.

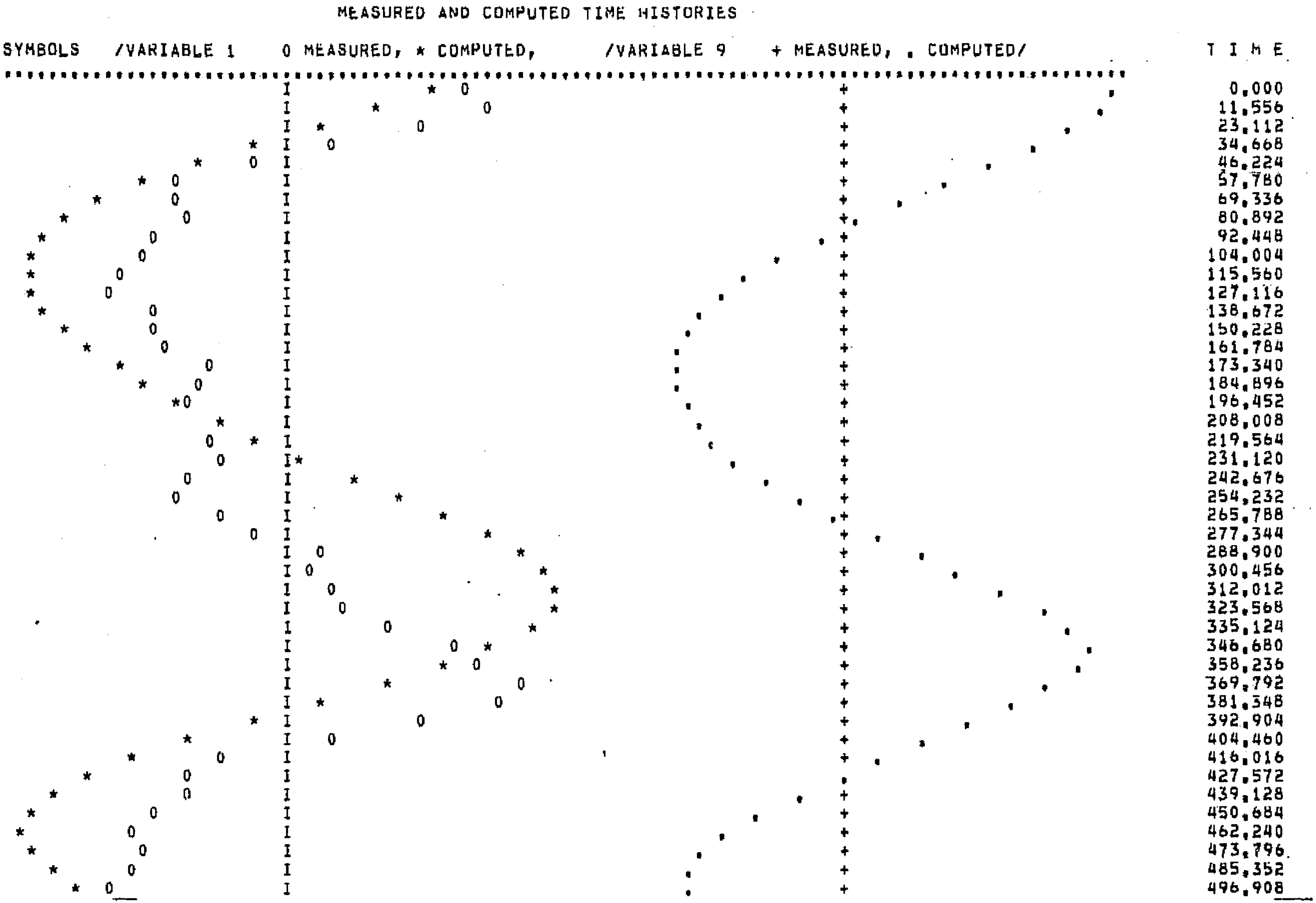


MAXIMA : VARIABLE 4 MEAS= 7.989E 03, COMP= 7.603E 03 VARIABLE 8 MEAS= 1.000E-50, COMP= 5.114E 03

**RESCOR

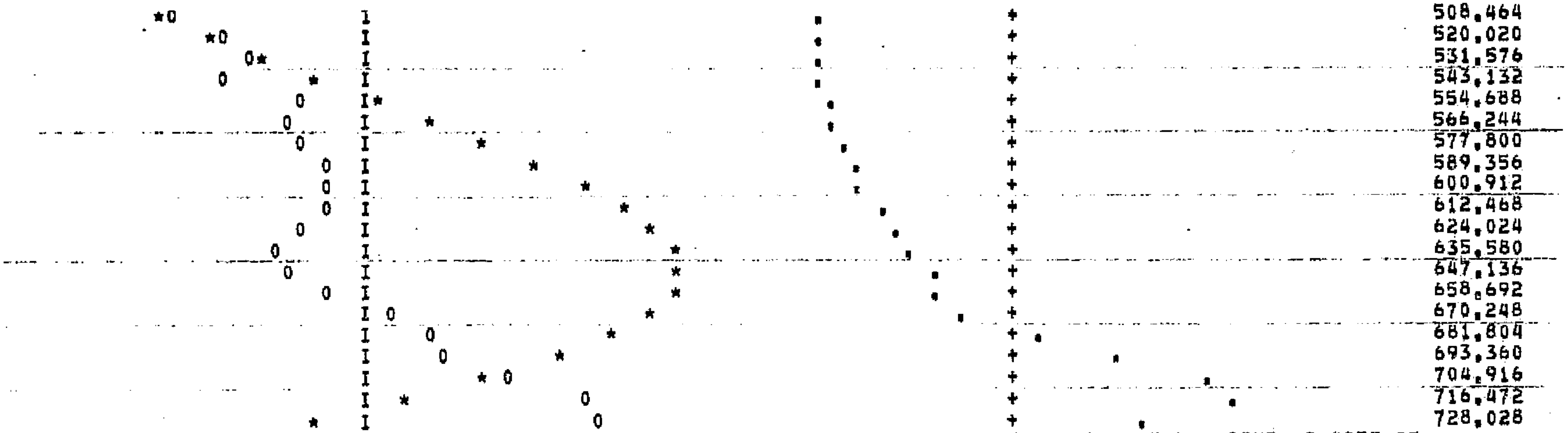
(w) Output.

Figure 2.- Continued.



(x) Output.

Figure 2.- Continued.



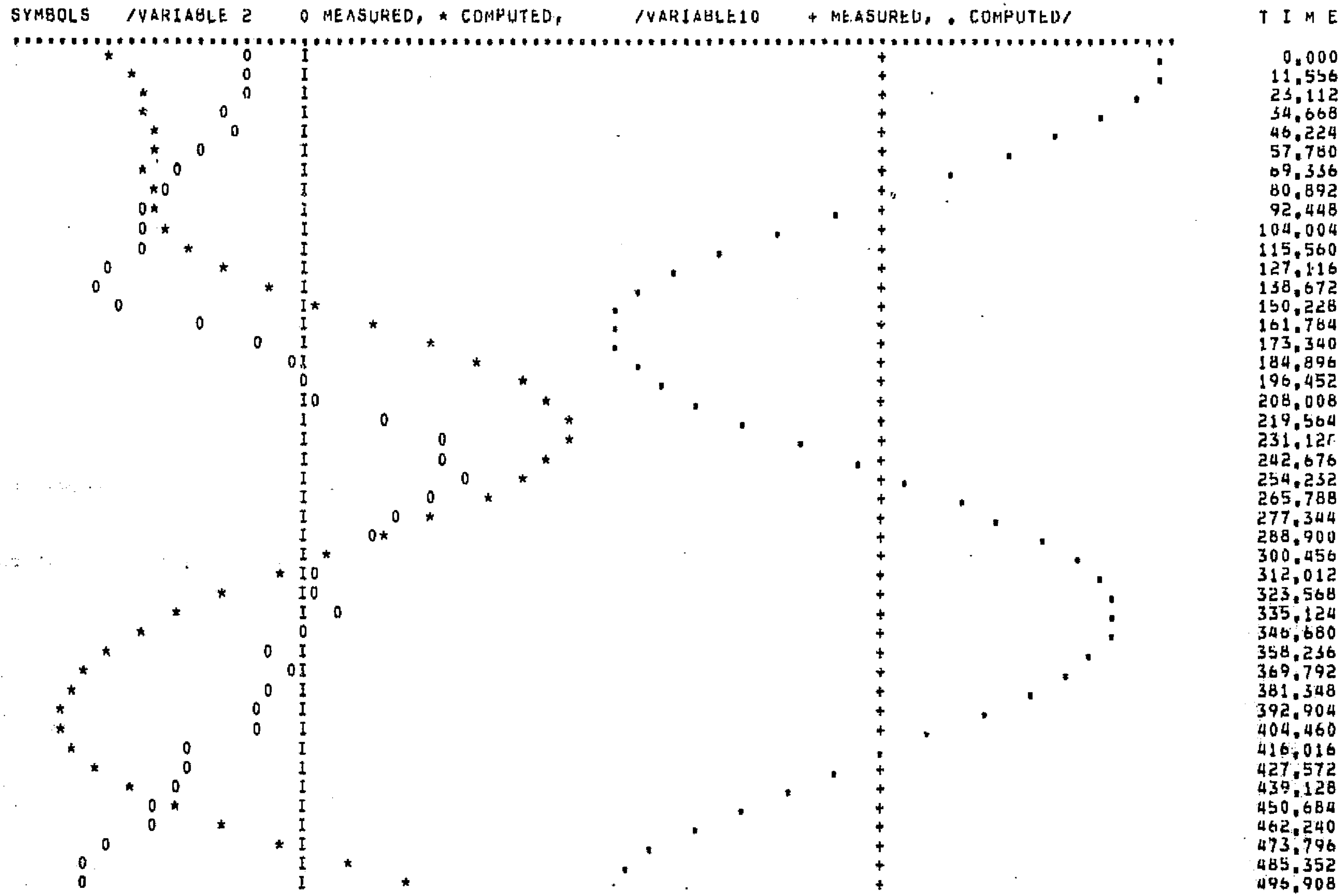
MAXIMA : VARIABLE 1 MEAS= 7.910E 03, COMP= 9.086E 03 VARIABLE 9 MEAS= 1.000E-50, COMP= 2.493E 07

**RESCOR

(y) Output.

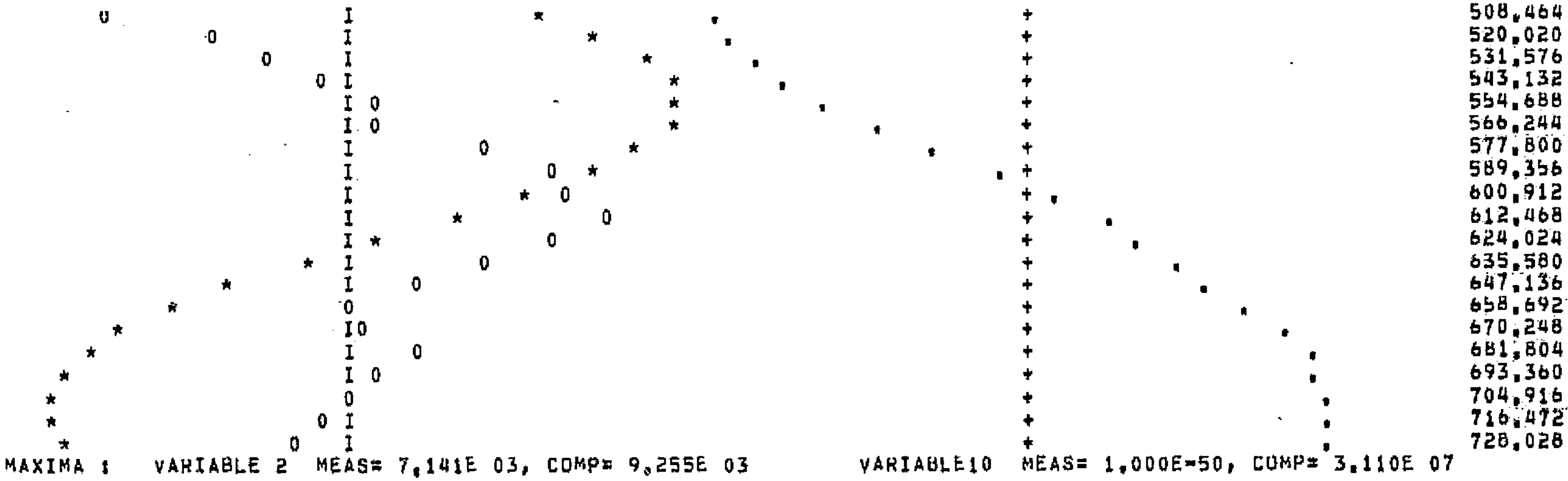
Figure 2.- Continued.

MEASURED AND COMPUTED TIME HISTORIES



(z) Output.

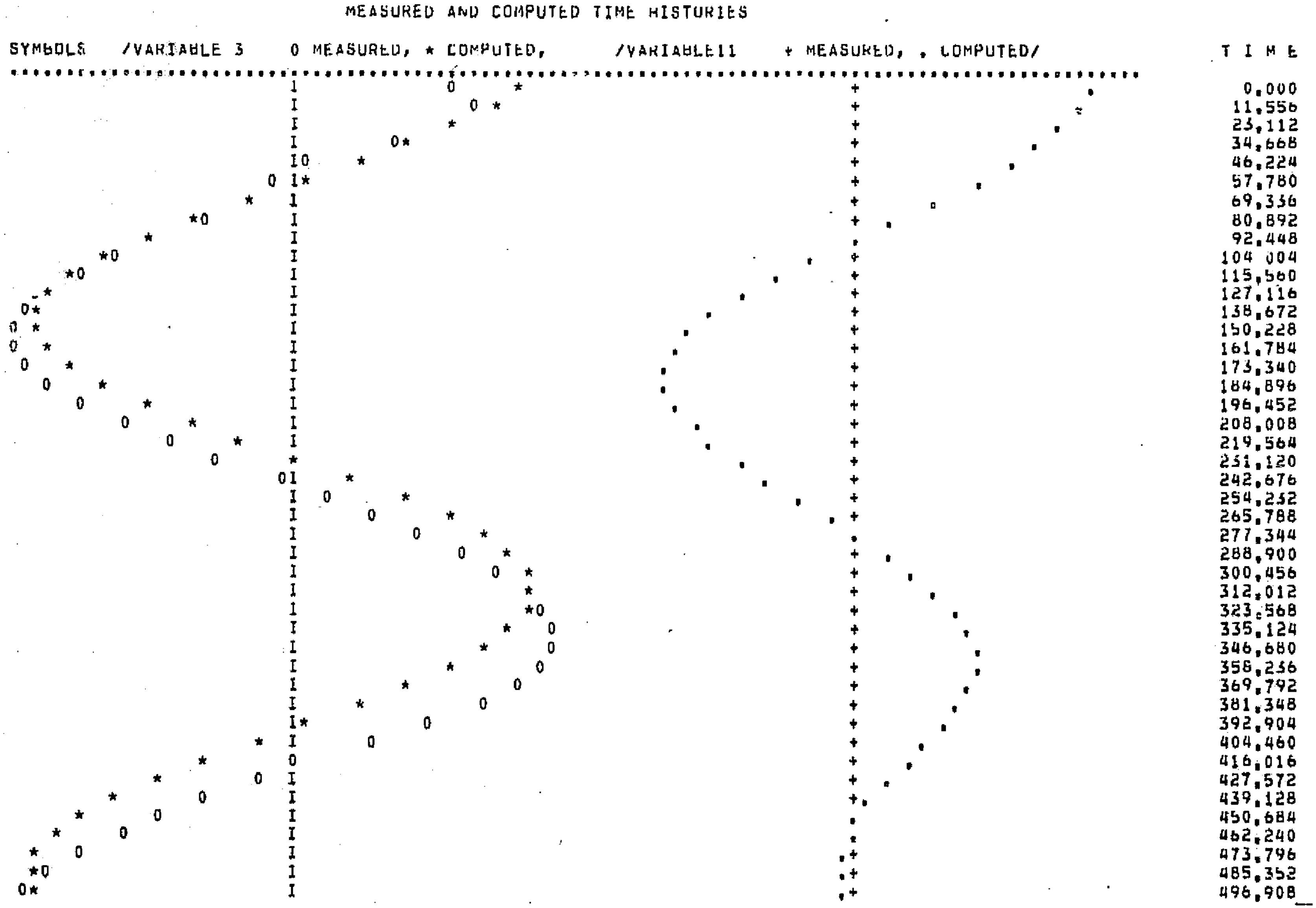
Figure 2.- Continued.



**RESCOR

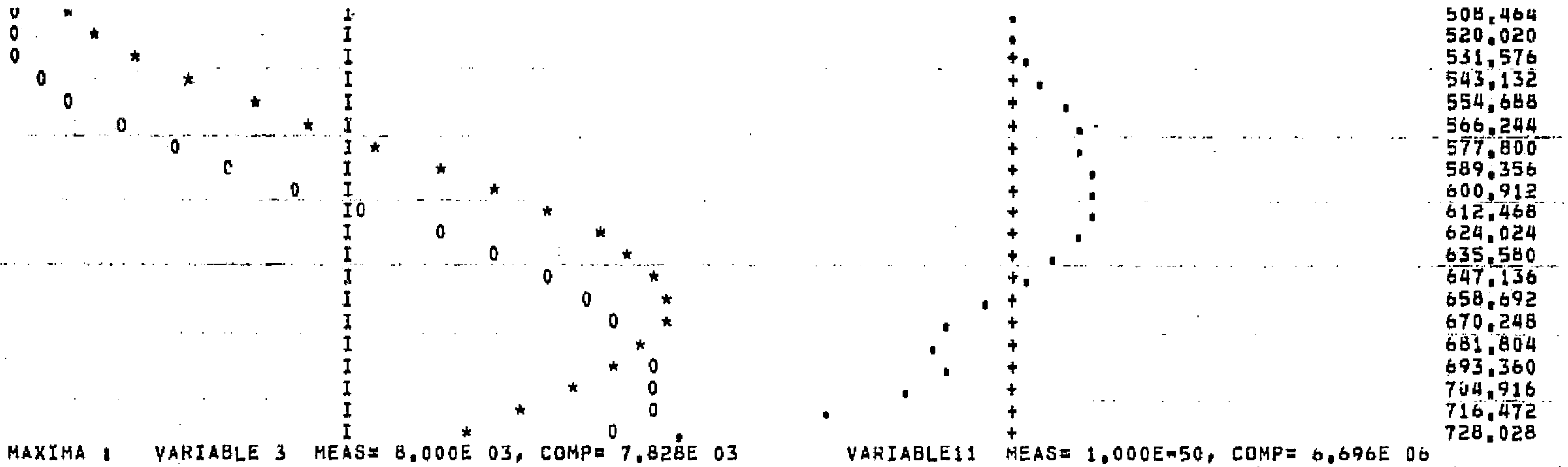
(aa) Output.

Figure 2.- Continued.



(bb) Output.

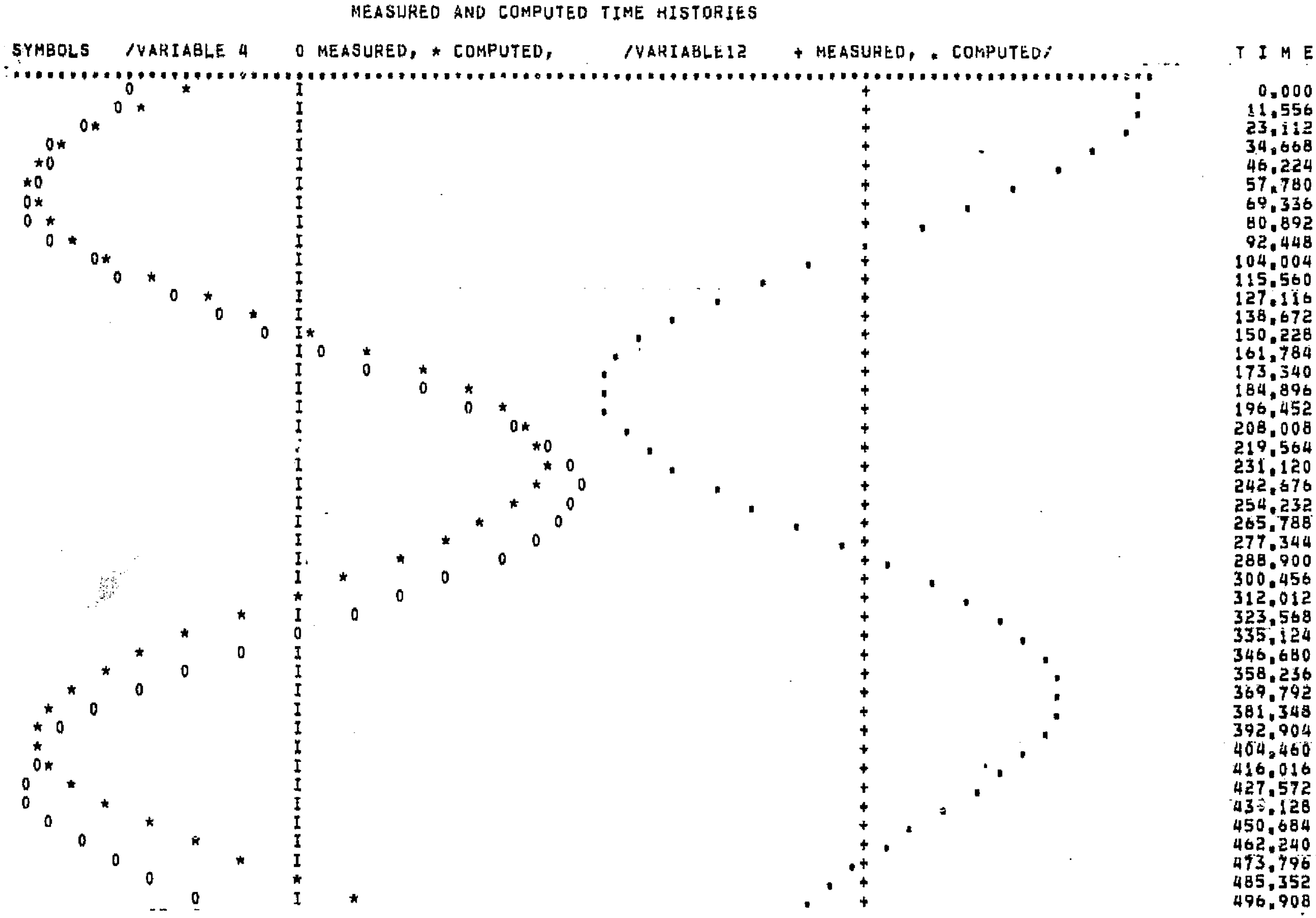
Figure 2.- Continued.



**RESCOR

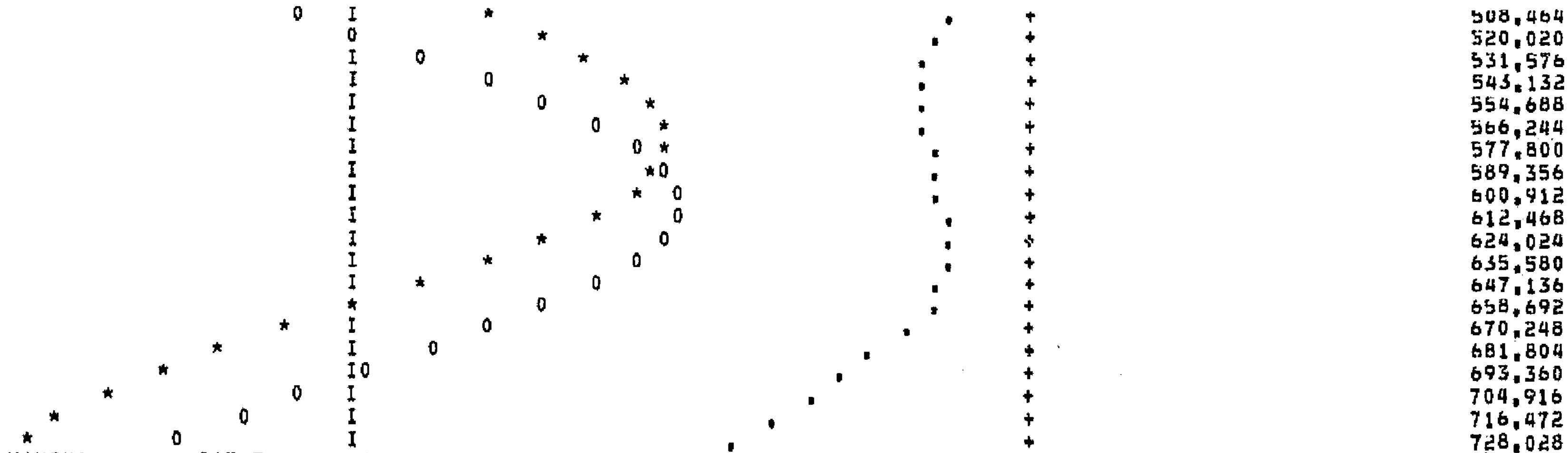
(cc) Output.

Figure 2.- Continued.



(dd) Output.

Figure 2.- Continued.




```
MAXIMA :   VARIABLE 4  MEAS= 7.989E 03, COMP= 7.603E 03        VARIABLE12  MEAS= 1.000E-50, COMP= 6.814E 06
```

```
*GO STOR

THE DATA WILL BE STORED ON UNIT10
```

(ee) Output.

Figure 2.- Continued.

```
NTAPE 1 NRUN 6      KDT=11556

VARIABLES STORED ARE IDENTIFIED BY A 1 IN COLUMNS 11 TO 22 AFTER THE COMMAND *STORAGE
*STORAGE  111111111111
```


(ff) Output.

Figure 2.- Continued.

```
  ****OUTPUT DATA HAD TO BE RESCALED BEFORE STORING
TO OBTAIN ENGR.    VALUES MULTIPLY STORED VALUES BY  :

 2.249E-01 FOR CHANNEL  5
 2.523E-01 FOR CHANNEL  6
 4.982E 02 FOR CHANNEL  9
 6.694E 02 FOR CHANNEL 10
 3.348E-01 FOR CHANNEL 11
 3.372E-01 FOR CHANNEL 12
  VARIABLE 1 IS STORED
  VARIABLE 2 IS STORED
  VARIABLE 3 IS STORED
  VARIABLE 4 IS STORED
  VARIABLE 5 IS STORED
  VARIABLE 6 IS STORED
  VARIABLE 7 IS STORED
  VARIABLE 8 IS STORED
  VARIABLE 9 IS STORED
  VARIABLE10 IS STORED
  VARIABLE11 IS STORED
  VARIABLE12 IS STORED
END


 *GO PRINT
```


(gg) Output.

Figure 2.- Continued.

COMPUTED OUTPUT VALUES					MEASURED VALUES					COMPUTED VALUES					
PSI	MBETAX	MBETAY	ROLL A	PITCH A	MBETAX	MBETAY	ROLL A	PITCH A	SPMR	SPMP	BETA1	BETA2	BETA3	RDELT1	RDELT2
0.00	781.0	-1021.6	2.652	-1.150	984.3	-285.7	1.915	-1.828	-63.8	8.3	1.912	1.400	0.592	-28.56	-51.90
11.56	500.7	-953.6	2.327	-1.707	1054.4	-308.2	2.098	-2.006	-58.9	7.6	1.857	1.261	0.745	-36.80	-51.47
23.11	196.1	-890.9	1.902	-2.181	723.4	-307.7	1.816	-2.363	-55.0	10.6	1.796	1.099	0.896	-47.36	-42.98
34.67	-117.5	-846.8	1.386	-2.552	258.2	-410.2	1.164	-2.664	-49.0	13.3	1.740	0.920	1.041	-55.49	-31.49
46.22	-422.7	-825.9	0.808	-2.796	-158.5	-348.0	0.199	-2.710	-48.1	10.1	1.694	0.735	1.172	-60.26	-18.02
57.78	-701.9	-824.1	0.198	-2.913	-537.6	-519.7	-0.203	-2.807	-57.7	14.3	1.659	0.557	1.283	-61.72	-4.90
69.34	-939.6	-830.1	-0.433	-2.894	-565.7	-642.1	-0.496	-2.961	-59.0	15.5	1.627	0.396	1.369	-62.44	8.55
80.89	-1124.7	-827.1	-1.044	-2.735	-518.7	-702.7	-1.008	-2.929	-61.6	14.1	1.589	0.266	1.429	-59.14	22.66
92.45	-1251.2	-796.8	-1.609	-2.444	-658.5	-887.7	-1.568	-2.662	-55.9	10.6	1.534	0.173	1.467	-52.99	35.38
104.00	-1318.3	-723.1	-2.090	-2.039	-710.9	-846.9	-2.030	-2.246	-44.2	0.2	1.454	0.124	1.488	-43.37	46.08
115.56	-1330.6	-595.2	-2.464	-1.555	-825.4	-835.9	-2.406	-1.936	-46.1	-3.1	1.344	0.122	1.498	-31.62	52.68
127.12	-1296.6	-409.6	-2.731	-1.001	-914.8	-1065.6	-2.761	-1.382	-48.3	2.7	1.204	0.166	1.507	-20.88	57.97
138.67	-1226.0	-170.6	-2.876	-0.394	-691.1	-1094.8	-2.960	-0.827	-43.3	7.8	1.040	0.253	1.521	-8.40	62.13
150.23	-1127.9	110.2	-2.884	0.228	-674.6	-962.8	-3.103	-0.293	-45.8	20.2	0.862	0.377	1.544	5.73	62.28
161.78	-1008.5	414.1	-2.746	0.846	-591.3	-562.4	-3.110	0.259	-41.4	22.9	0.680	0.530	1.573	20.00	60.11
173.34	-870.7	717.8	-2.478	1.409	-390.8	-230.6	-3.008	0.820	-42.9	13.8	0.509	0.703	1.603	32.83	53.02
184.90	-714.5	996.6	-2.111	1.898	-446.8	-19.6	-2.737	1.448	-51.4	12.6	0.361	0.885	1.624	41.71	44.07
196.45	-538.7	1227.5	-1.649	2.302	-502.2	52.5	-2.356	1.929	-54.8	10.9	0.245	1.067	1.625	49.97	35.04
208.01	-341.5	1391.7	-1.113	2.600	-318.7	69.4	-1.896	2.451	-59.8	9.0	0.170	1.241	1.595	56.47	23.77
219.56	-122.9	1476.7	-0.523	2.782	-379.6	464.1	-1.347	2.812	-57.2	12.8	0.140	1.401	1.528	60.68	11.67
231.12	114.8	1476.9	0.095	2.825	-321.6	808.5	-0.814	3.045	-53.6	12.7	0.156	1.540	1.421	62.23	-2.39
242.68	366.1	1394.6	0.705	2.731	-520.3	802.8	-0.120	3.163	-65.0	25.1	0.218	1.657	1.279	59.83	-16.00
254.23	621.2	1238.5	1.301	2.496	-525.4	927.8	0.434	3.137	-58.4	28.5	0.319	1.751	1.108	57.04	-29.37
265.79	867.1	1021.6	1.823	2.134	-334.0	737.9	0.969	3.002	-51.8	25.3	0.453	1.818	0.921	48.15	-41.99
277.34	1087.3	760.1	2.243	1.673	-131.0	561.9	1.520	2.735	-44.7	23.8	0.612	1.856	0.732	36.39	-50.67
288.90	1264.5	470.9	2.538	1.139	204.0	423.1	1.973	2.339	-36.2	9.9	0.786	1.862	0.554	23.28	-56.36
300.46	1382.7	170.7	2.707	0.582	167.3	153.1	2.403	1.745	-47.2	3.3	0.964	1.832	0.402	10.17	-57.13
312.01	1429.6	-125.3	2.769	0.004	271.8	95.7	2.722	1.208	-53.0	6.9	1.136	1.763	0.284	0.05	-57.72
323.57	1397.8	-404.2	2.711	-0.583	298.5	83.0	2.924	0.648	-53.4	14.1	1.294	1.655	0.209	-11.77	-57.79
335.12	1285.0	-656.0	2.522	-1.151	577.2	208.2	3.031	0.115	-45.1	11.0	1.432	1.509	0.179	-24.72	-54.91
346.68	1095.2	-873.2	2.219	-1.656	873.4	28.8	3.029	-0.514	-47.0	9.9	1.546	1.330	0.197	-35.83	-47.23
358.24	838.8	-1050.0	1.818	-2.090	1001.1	-151.5	2.875	-1.147	-47.8	11.4	1.634	1.126	0.258	-44.35	-38.97
369.79	532.0	-1182.8	1.335	-2.432	1264.8	-54.8	2.610	-1.694	-46.2	6.8	1.695	0.909	0.359	-51.49	-29.07
381.35	194.5	-1268.6	0.802	-2.667	1143.9	-164.3	2.180	-2.230	-54.1	11.5	1.729	0.693	0.491	-55.51	-17.66
392.90	-151.5	-1304.8	0.222	-2.791	718.3	-192.5	1.646	-2.589	-48.7	9.7	1.738	0.490	0.647	-58.86	-6.51
404.46	-483.9	-1288.4	-0.363	-2.788	264.3	-250.2	0.938	-2.828	-46.1	5.1	1.720	0.314	0.815	-58.34	6.63
416.02	-782.0	-1217.0	-0.927	-2.665	-342.7	-607.0	0.045	-2.898	-51.3	11.6	1.672	0.176	0.985	-54.87	18.46
427.57	-1028.3	-1089.1	-1.464	-2.419	-509.4	-601.5	-0.381	-2.968	-50.6	12.4	1.594	0.085	1.149	-51.05	29.96
439.13	-1209.8	-904.8	-1.939	-2.064	-521.9	-672.4	-0.935	-2.918	-62.0	12.2	1.484	0.046	1.298	-43.82	40.52
450.68	-1319.0	-667.5	-2.349	-1.614	-680.4	-778.9	-1.443	-2.700	-59.7	11.7	1.342	0.061	1.426	-36.26	48.92
462.24	-1354.0	-383.9	-2.657	-1.087	-791.5	-771.6	-1.911	-2.346	-54.8	7.8	1.172	0.127	1.526	-25.36	56.01
473.79	-1318.3	-65.1	-2.848	-0.511	-744.5	-1069.7	-2.352	-1.979	-60.3	9.0	0.981	0.240	1.604	-12.98	59.69
485.35	-1219.9	273.4	-2.923	0.102	-792.0	-1180.8	-2.700	-1.563	-53.6	9.2	0.778	0.392	1.654	-1.33	61.81
496.91	-1069.6	612.6	-2.866	0.712	-888.2	-1190.6	-2.991	-1.040	-50.3	8.1	0.576	0.572	1.678	12.11	60.51
508.46	-879.2	931.3	-2.683	1.295	-839.5	-1087.1	-3.154	-0.482	-45.0	15.9	0.388	0.768	1.678	24.38	56.21

(hh) Output.

Figure 2.- Continued.

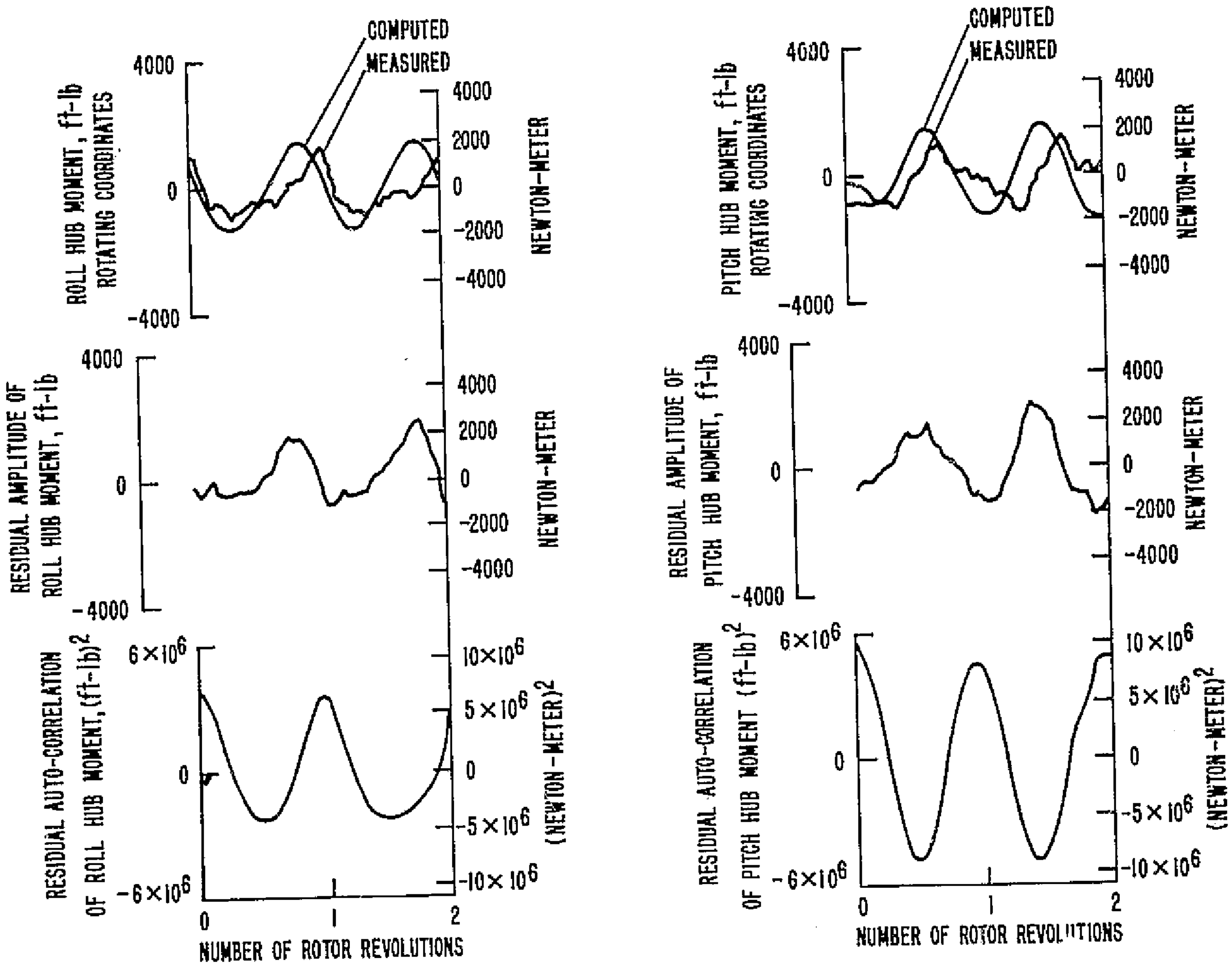
520.02	-660.4	1207.6	-2.370	1.824	-620.6	-617.7	-3.200	0.039	-35.3	19.1	0.227	0.968	1.651	36.70	49.71
531.57	-423.0	1420.7	-1.948	2.263	-483.5	-362.0	-3.119	0.706	-44.1	17.6	0.105	1.160	1.598	46.97	39.43
543.13	-175.5	1553.2	-1.444	2.616	-591.5	-122.4	-2.858	1.314	-47.9	17.5	0.031	1.333	1.515	53.64	29.74
554.69	75.1	1593.8	-0.877	2.861	-289.1	157.0	-2.600	1.806	-46.4	6.4	0.011	1.479	1.401	59.04	18.60
566.24	323.1	1537.8	-0.284	2.990	-312.0	189.6	-2.155	2.336	-53.5	3.9	0.044	1.590	1.256	60.54	6.48
577.80	562.8	1388.8	0.330	3.007	-266.6	638.7	-1.645	2.688	-41.7	2.8	0.130	1.666	1.084	61.37	-4.46
589.35	788.9	1157.4	0.926	2.892	-120.6	962.1	-1.106	2.949	-41.7	3.2	0.262	1.707	0.891	58.76	-17.63
600.91	995.0	861.3	1.492	2.660	-167.9	1081.1	-0.471	3.133	-48.6	11.8	0.430	1.717	0.687	54.28	-28.66
612.47	1173.9	522.3	2.017	2.314	-155.4	1229.7	0.163	3.144	-44.3	16.4	0.623	1.699	0.484	49.12	-39.40
624.02	1316.9	164.5	2.460	1.864	-274.4	990.0	0.896	2.951	-51.9	22.8	0.830	1.659	0.296	39.74	-49.40
635.58	1413.6	-188.7	2.812	1.323	-385.7	648.5	1.506	2.670	-51.0	31.1	1.037	1.599	0.136	29.67	-57.29
647.13	1452.8	-516.4	3.040	0.710	-307.8	329.8	1.952	2.278	-46.2	26.4	1.231	1.519	0.016	16.52	-63.58
658.69	1423.6	-801.9	3.131	0.071	-165.3	28.1	2.370	1.776	-55.0	18.4	1.401	1.418	-0.054	2.03	-64.89
670.25	1318.2	-1033.7	3.101	-0.575	132.2	127.3	2.651	1.314	-47.9	7.1	1.540	1.295	-0.069	-9.96	-63.85
681.80	1133.9	-1205.7	2.943	-1.186	334.4	347.7	2.875	0.838	-44.3	-0.6	1.641	1.146	-0.027	-22.52	-59.43
693.36	874.5	-1316.6	2.669	-1.748	379.6	156.5	3.013	0.210	-48.3	7.0	1.703	0.975	0.070	-33.25	-53.39
704.91	550.6	-1369.4	2.277	-2.259	666.5	13.6	3.032	-0.391	-45.3	10.6	1.728	0.785	0.216	-43.92	-47.47
716.47	179.0	-1370.2	1.786	-2.685	1002.9	-101.1	2.961	-0.988	-45.6	9.2	1.721	0.585	0.400	-53.38	-37.98
728.03	1599.0	1722.0	4.000	3.959	1599.0	1722.0	4.000	3.959	-49.8	13.6	1.688	0.386	0.611	-59.71	-26.91

****

***END OF IDENTIFICATION RUN NUMBER 1 . TOTAL COMPUTATION TIME181.50 SEC. ***
*RETURN

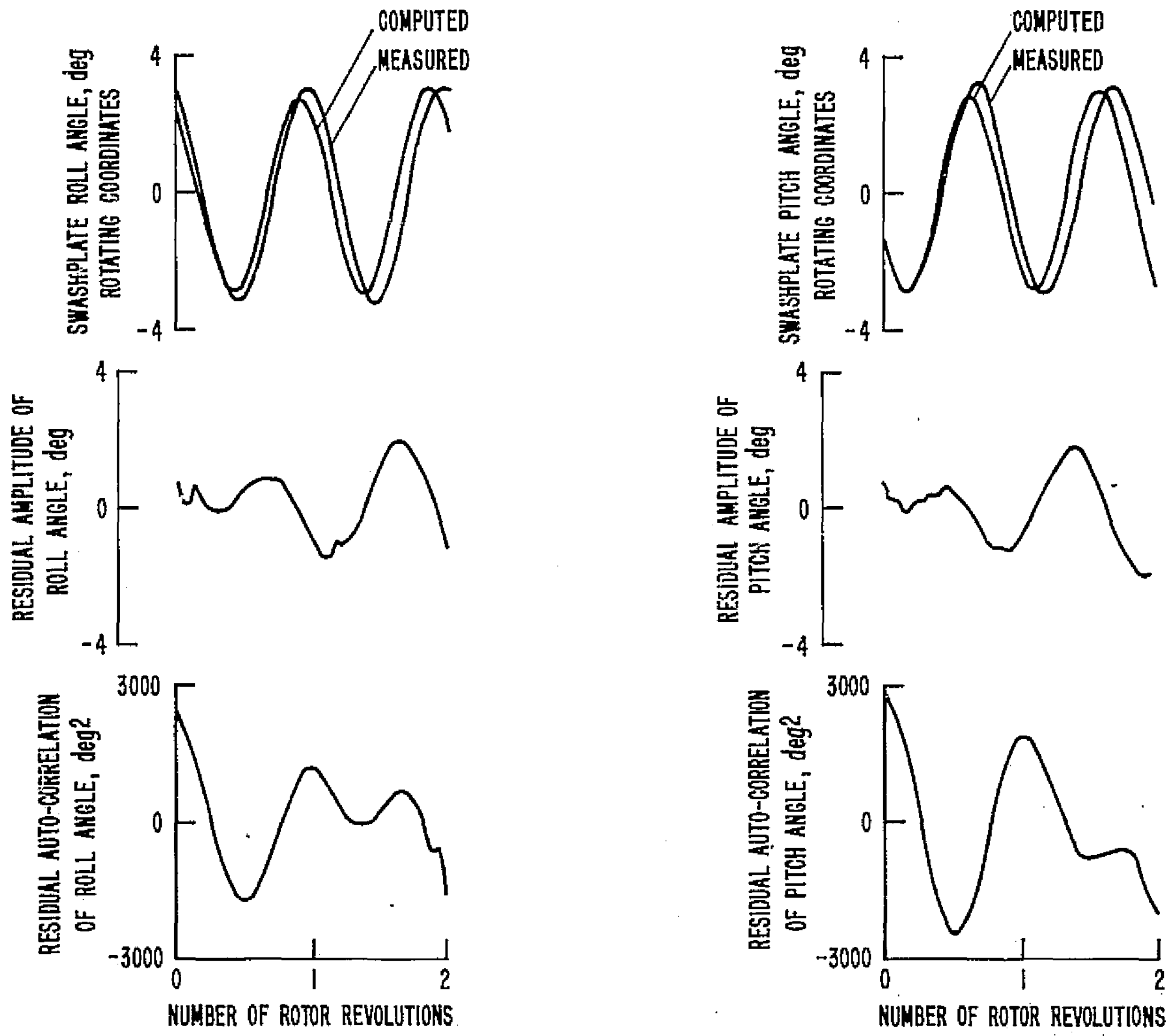
(ii) Output.

Figure 2.- Continued.



(jj) Identification results for two iterations; plot of hub moments and the residual amplitude and autocorrelation of the error between computed and measured hub moments.

Figure 2.- Continued.



(kk) Identification results for two iterations; plot of swashplate angles and the residual amplitude and autocorrelation of the error between computed and measured swashplate angles.

Figure 2.- Concluded.

```
   LOGON FSTGK,MAINPR,D260IFST19 T4947 'GERD KANNING STOP 17 PH 5455'
CALL MAIN2SS
-1 0 0 0 0.0 0
+.40    +00+.268  +03+.2684 +03+.165  +02+.117  +01+.97   +00
+.15    +01+.1926 +03+.188  +01+.4566 +01+.00   +00+.12021+06+.145  +01
-.943   +01+.225  +01+.0    +00+.70   +02+.2246 -02
+.0     +00+.0    +00-.2428 +01+.11296+01+.10   -01
+.1912  +01-.4666 +01+.14   +01-.12535+02+.592  +00+.1481 +02+.857  +01
+.3     +00+.15   +00+.60   +02+.00   +00+.10200+05+.87   +00
+.00    +00+.70   +01+.80   +02
+.2652  +01-.2856 +02-.115  +01-.519  +02-.507  +02+.121  +02
+.1768  +01          -.252  +01
+.5778  +01+.00   +00+.00   +00+.2    +01
+.10    +02+.10   +02+.10   -01+.10   -01+.1    -01+.1    -01+.1    -01

*SENSITIVITY
SENSITIVITY ANALYSIS FOR HELICOPTER ROTOR PARAMETERS--RECORD 6
   21   63    4     1  210000
                                                          .0001            6
2222222222222222222
+.268   +03+.97   +00+.857  +01+.12021+06+.4566 +01+.15   +01    C
+.87    +00-.699  -02-.10   -04+.1    -04+.188  +01+.225  +01    C
+.1     -01+.355  +00+.80   +02+.7    +01+.4    +00+.5296 +02    C
+.3     +00+.15   +00-.943  +01

BIY ,LOSS,CLEC,BETA,LOCK,SWEP,  C2,LAM ,LAMX,LAMY,BRDO,PCON,
  CD,  MU,CDMP,FDMP, BIX,GSPD,GYIP,GYID,TWST
+.1599 +00+.1722 +00+.40000-03+.3959 -03+.3495 -03+.5140 -05    C
 FT-LBS  FT-LBS  DEGREE  DEGREE
+.8797 -02+.4493 -02
 FT-LB   FT-LB
SLUG,FT2(NO DIM)DEG      FTLB/RAD(NO DIM)DEG      (NO DIM)(NO DIM)(NO DIM)(NO DIM)
FT       DEG      (NO DIM)(NO DIM)FTLB/R,SFTLB/R,SSLUG,FT2(NO DIM)SLUG,FT2SLUG,FT2
DEG
OSET 5
RUN  1  6       5778
END
**********.
```

(a) Data input deck.

Figure 3.- Sensitivity run.

```
***********
*RETURN
  LOGOFF
```


(b) Data input deck.

Figure 3.- Continued.

TWO DEGREE OF FREEDOM HELICOPTER BLADE AND GYRO SIMULATION

PROGRAM CONTROL LOGIC

IFLOQ=-1 IHINGE= 0 IROT= 0 INEW= 0 ISIML= 0 IFLOW= 0 IREC= 0

BLADE PARAMETERS

BIX= 0.40 BIY=268.00 BIZ=268.40 RADIUS=16.5 CHORD=1.17 BLADE LOSS=.97

SWEEP= 1.50 RPM= 192.6 HINGE POSITION=1.88 LOCK= 4.566 DELTA3= 0.00 K BETA=0.120210E 06 P1= 1.45

TWIST= -9.43 PRECONE= 2.250 CT= 0.0000 AIR VELOCITY= 70.00 AIR DENSITY=.002246 ALPHA= 0.000 ALPHAS= 0.000

LITTLE A1S= 0.000 LITTLE B1S= 0.000 BIG A1S=-2.428 BIG B1S= 1.130 CD=0.010 LAM=0.000000 LAMX=0.000000 LAMY=0.000000

BLADE INITIAL CONDITIONS

BETA1= 1.912 BETA1 RATE= -4.67 BETA2= 1.400 BETA2 RATE= -12.53 BETA3= 0.592 BETA3 RATE= 14.81 COLLECTIVE= 8.6

GYRO PARAMETERS

IP= 0.300 ID= 0.150 GYRO ANGLE= 60.0 GYRO DAMP L= 0.00 GYRO RPM=10200.0 MECHANICAL ADVANTAGE C2=0.870

FEATHERING RESTRAINT KZETA= 0.00 FEATHERING DAMPING CZETA= 7.00 SWASHPLATE DAMPING CSDAMP= 80.00 PSIOF= 0.00

INITIAL CONDITIONS ON GYRO

DELTA1= 2.652 DELTA1 RATE= -28.56 DELTA2= -1.150 DELTA2 RATE= -51.90 NU1=-50.70 NU2= 12.10

PHI= 1.768 RPHI= 0.00 THET= -2.520 RTHET= 0.00 RMBIAS= 0.00 PMBIAS= 0.00 RABIAS= 0.000 PABIAS= 0.000

INITIAL CONDITIONS FOR STARTING AND STOPPING THE INTEGRATION ROUTINE

DELPSI= 5.78 REVS= 0.00 REVO= 0.00 REVF= 2.00 TRUNC=0.0000

SCALE FACTORS FOR PLOTTING PROGRAM

SCALE(1)	SCALE(2)	SCALE(3)	SCALE(4)	SCALE(5)	SCALE(6)	SCALE(7)
1.000E 01	1.000E 01	1.000E-02	1.000E-02	1.000E-02	1.000E-02	1.000E-02
SCALE(8)	SCALE(9)	SCALE(10)	SCALE(11)	SCALE(12)	SCALE(13)	SCALE(14)
0.000	0.000	0.000	0.000	0.000	0.000	0.000

(c) Output.

Figure 3.- Continued.

PARAMETERS CALULATED FROM INPUT DATA
OMEGA= 20.169 GYSPED= 52.96 MU= 0.355 LAMBDA=-.006999

DELTAT=0.5000E-02 TIMES=0.0000 TIMED=0.0000 TIMEF=0.6231E 00

*SENSITIVITY
SENSITIVITY ANALYSIS FOR HELICOPTER ROTOR PARAMETERS--RECORD 6

(d) Output.

Figure 3.- Continued.

```
 DATA CARDS LISTING*****RUN NUMBER  1= 6
X AXIS INCREMENT KDT=  5778 SKIP=  1      63 POINTS PER CHANNEL
THE  0 FOLLOWING CHANNELS WERE READ : 0,
```


(e) Output.

Figure 3.- Continued.

```
                    SENSITIVITY ANALYSIS FOR HELICOPTER ROTOR PARAMETERS--RECORD 6
                    ----------------------------------------------------------------

 21 PARAMETERS       4 OUTPUTS

 63 MEASURED VALUES PER OUTPUT    ( 252 MEACUREMENTS),        1 WAS THE SKIPPING FACTOR USED

    TIME INTERVAL BETWEEN MEASUREMENTS  0.0050 SEC.          2 STEPS OF INTEGRATION IN EACH INTERVAL

   ANGLE INCREMENT BETWEEN MEASUREMENTS      5.78 DEG


INITIAL CONDITIONS
  4.629E-02 -4.985E-01 -2.007E-02 -9.058E-01  3.337E-02 -8.144E-02  2.443E-02 -2.188E-01  1.033E-02  2.585E-01

SCALING FACTORS FOR DATA UNITS
1DTU=  1.599E-01 FT-LBS  /  1.722E-01 FT-LBS  /  4.000E-04 DEGREE  /  3.959E-04 DEGREE  /  3.495E-04          /  5.140E-06
 8.797E-03 FT-LB    /  4.493E-03 FT-LB    /

PARAMETERS NAMES
    1) BIY      2) LOSS     3) CLEC     4) BETA     5) LOCK     6) SWEP     7)   C2     8) LAM      9) LAMX    10) LAMY    11) BRDO    12) PCON

   13)    CD   14)    MU   15) CDMP    16) FDMP    17)  BIX    18) GSPD    19) GYIP    20) GYID    21) TWST

INITIAL VALUES OF THE PARAMETERS
  2.680E 02  9.700E-01  8.570E 00  1.202E 05  4.566E 00  1.500E 00  8.700E-01 -6.990E-03 -1.000E-05  1.000E-05  1.880E 00  2.250E 00

  1.000E-02  3.550E-01  8.000E 01  7.000E 00  4.000E-01  5.296E 01  3.000E-01  1.500E-01 -9.430E 00

***THE FOLLOWING WEIGHTS ARE USED***
      1.000E 00            1.000E 00            1.000E 00            1.000E 00
ERRMIN =  1.000E-04  ,    DMIN =  0.000

INITIAL PARAMETER INCREMENTS FOR GRADIENT EVALUATION
  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02
  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02  1.000E-02
                             ********

ITERATION  0.    MODE WAS99.  KND= 2                                              REL.ERROR = 0.00         COST = 0.000
---------------------------------------P  A  R  A  M  E  T  E  R  S       V  A  L  U  E  S---------------------------------------
  1 13 25     2 14 26     3 15 27     4 16 28     5 17 29     6 18 30     7 19 31     8 20 32     9 21 33    10 22 34    11 23 35    12 24 36
  2.680E 02  9.700E-01  8.570E 00  1.202E 05  4.566E 00  1.500E 00  8.700E-01 -6.990E-03 -1.000E-05  1.000E-05  1.880E 00  2.250E 00

  1.000E-02  3.550E-01  8.000E 01  7.000E 00  4.000E-01  5.296E 01  3.000E-01  1.500E-01 -9.430E 00
```

(f) Output.

Figure 3.- Continued.

BASIC PARAMETER GYIP	SEPARATION	CRITICAL PARAMETER	SEPARATION
LAMX	1,00E 00	BIX	1,06E-01
BETA	9,13E-01	SWEP	7,68E-02
GYID	8,34E-01	BIX	7,56E-02
CDMP	6,67E-01	BIX	3,23E-02
LAMY	5,12E-01	BIX	1,36E-02
PCON	3,39E-01	BIX	1,19E-02
BIY	3,30E-01	BIX	1,14E-02
MU	3,01E-01	BIX	6,60E-03
LOSS	2,39E-01	BIX	6,13E-03
LAM	1,83E-01	BIX	5,86E-03
CD	6,09E-02	BIX	5,69E-03
C2	5,87E-02	BIX	4,92E-03
BRDD	5,06E-02	BIX	4,86E-03
FDMP	3,62E-02	TWST	4,00E-03
LOCK	5,78E-03	SWEP	3,47E-03
BIX	3,52E-03	TWST	2,10E-03
SWEP	3,39E-03	TWST	2,00E-03
CLEC	1,95E-03	TWST	1,75E-03

```
          DEPENDENT SETS OF PARAMETERS
SEPARATION

 0,92E-07
         GSPD,GYIP,

IF GSPD' = 1, THEN   GYIP' =-5,66E-03
 0,31E-04
         BIY ,LOSS,CLEC,LAM ,PCON,CDMP,GYIP,GYID,TWST,

IF TWST' = 1, THEN   BIY ' =-1,01E-01   LOSS' =-5,63E-05   CLEC' =-6,53E-01   LAM ' =-1,73E-03   PCON' = 5,28E-03   CDMP' =-4,42E-01
                     GYIP' =-1,82E-03   GYID' =-6,85E-03
```

(g) Output.

Figure 3.- Continued.

```
***THERE ARE 21 PARAMETERS IN THIS PROBLEM ,        THE SEPARATION THRESHOLD WAS 0.10E-03
MAGIC NUMBER  1 1 1 2 2 2 2 1 2 2 2 1 2 2 1 2 2 0 1 1 0

***DEPENDENT SET NUMBER  1   ***SEPARATION = 0.92E-07
*
* A TRUE VALUE IS OBTAINED FOR
*        BIY ,LOSS,CLEC,LAM ,PCON,CDMP,GYIP,GYID,
* IF IS KNOWN THE TRUE VALUE OF
*        GSPD,TWST,

*  INDEPENDENT PARAMETERS
*  BETA,LOCK,SWEP,  C2,LAMX,LAMY,BRDO,  CD,  MU,FDMP, BIX,
*
*  IRRELEVANT PARAMETERS
*  NONE,
*
*  NOT ESTIMATED
*  GSPD,TWST,
*
*  NOT USED
*  NONE,


DET= 0.1147030549762990-36   TIME =    3.954 SEC


PARAMETER INCREMENTS (DA/A) FOR GRADIENT EVALUATION

  BIY =  1.0E-01  LOSS=  2.8E-02  CLEC=  2.2E-02  BETA=  1.0E-01  LOCK=  1.1E-01  SWEP=  1.5E-01
    C2=  8.6E-02  LAM =  5.0E-01  LAMX=  5.0E-01  LAMY=  5.0E-01  BRDO=  5.0E-01  PCON=  2.3E-01
    CD=  5.0E-01    MU=  1.0E-01  CDMP=  3.9E-01  FDMP=  5.0E-01   BIX=  5.0E-01  GSPD=  1.4E-01
  GYIP=  1.4E-01  GYID=  5.0E-01  TWST=  2.7E-02
```


ORIGINAL PAGE IS OF POOR QUALITY

(h) Output.

Figure 3.- Continued

FINAL RESULTS OF THIS IDENTIFICATION

FIT TEST ** COST = 0.0000 CORRELATION = 1.00000
RELATIVE ERROR = 0.0000

ESTIMATED VALUES OF THE PARAMETERS

	NAME		EST.VALUE	ERROR BOUNDS		UNITS	SENSITIVITY	DEPENDENCY INDEX FINAL	INITIAL
1	BIY	=	2.6800E 02	+/-	0.0	SLUG.FT2	0.76922	1	2
2	LOSS	=	9.7000E-01	+/-	0.0	(NO DIM)	2.74623	1	2
3	CLEC	=	8.5700E 00	+/-	0.0	DEG	4.07084	1	2
4	BETA	=	1.2021E 05	+/-	0.0	FTLB/RAD	0.74851	2	2
5	LOCK	=	4.5660E 00	+/-	0.0	(NO DIM)	0.87229	2	2
6	SWEP	=	1.5000E 00	+/-	0.0	DEG	0.51594	2	2
7	C2	=	8.7000E-01	+/-	0.0	(NO DIM)	1.10879	2	2
8	LAM	=	-6.9900E-03	+/-	0.0	(NO DIM)	0.14846	1	2
9	LAMX	=	-1.0000E-05	+/-	0.0	(NO DIM)	0.00028	2	2
10	LAMY	=	1.0000E-05	+/-	0.0	(NO DIM)	0.00031	2	2
11	BRDO	=	1.8800E 00	+/-	0.0	FT	0.01868	2	2
12	PCON	=	2.2500E 00	+/-	0.0	DEG	0.31244	1	2
13	CD	=	1.0000E-02	+/-	0.0	(NO DIM)	0.05107	2	2
14	MU	=	3.5500E-01	+/-	0.0	(NO DIM)	0.71696	2	2
15	CDMP	=	8.0000E 01	+/-	0.0	FTLB/R.S	0.13415	1	2
16	FDMP	=	7.0000E 00	+/-	0.0	FTLB/R.S	0.07337	[illegible]	2
17	BIX	=	4.0000E-01	+/-	0.0	SLUG.FT2	0.03063	[illegible]	2
18	GSPD	=	5.2960E 01	+/-	0.0	(NO DIM)	0.56705	[illegible]	2
19	GYIP	=	3.0000E-01	+/-	0.0	SLUG.FT2	0.56705	1	2
20	GYID	=	1.5000E-01	+/-	0.0	SLUG.FT2	0.00098	1	2
21	TWST	=	-9.4300E 00	+/-	0.0	DEG	3.26822	0	2

****

***END OF IDENTIFICATION RUN NUMBER 1 . TOTAL COMPUTATION TIME126.50 SEC. ***
*RETURN

(i) Output.

Figure 3.- Concluded.

APPENDIX A

INTERNAL PROGRAMS

A description of the internal programs and library programs is given in this appendix. The user is not required to change these programs. Although if the users application of these programs requires the estimation of more than 36 parameters some changes in these subroutines is required. The user will then have to become thoroughly acquainted with the programs given in this appendix before altering them. For cases requiring less than 36 parameters the application of these programs has been discussed in the text.

9. READIN$$

Description

The subroutine reads the measured data from either input cards or from the computer disk. The data the program reads must be scaled by the scaling factor SCALE(J). This scaling factor reduces all data to a four digit integer with the appropriate sign. For the data to be read from cards or disk storage a data card submitted with the identification input deck will specify which disk unit is to be read. This card has the format (starting in column 1) DESTxx where xx represents the unit from which data is to be read. The READIN subroutine which is called by the INOUT subroutine will then read the cards from the disk. If DSET05 is specified the READIN subroutine will look for the data cards submitted with the identification data input. (See example 1 for data card input and example 2 for data read from disk). An example of how the data must be organized for card or disk usage is shown in table 1A. All the cards contain, in column 1 to 4, a code name indicating the type of card defined as:

RUN first card of the data set

DATA first card containing data points corresponding to a variable

**** following cards containing data points of the same variable

END last card of the data set

In column 5-6 and 8-9 two integers are found with the following meaning:

-for a 'RUN' card the run number is specified and the time interval between measurements is 17 to 20 (note in table 1A the time interval was 5700 which had the units of milli-degree)

-for a 'DATA' or continuation ('****') card the variable index and card index corresponding to this variable. (Example **** 1 81 is the 81st data card containing the data points of the variable number 1).

In columns 11 to 80, 14 numbers are found with the format I5 representing the successive values of the variable.

Usage

CALL READIN (KSTOP, LIST, MISFIT)

The arguments in the call carry out the following functions:

KSTOP total number of data points that can be read

TABLE 1A.- EXAMPLE OF MEASURED DATA SET READ BY SUBROUTINE READIN$$

```
RUN   1  6         5700
DATA 1   1 -1642-2469-2053-1167-1083 -881-1315    0  381  508 -637  529 -452 -769
**** 1   2  -793-2281-4348-4746-3931-4839-3783-4440-3390-2625-2068-1251  872  875
  :  :   :     :    :    :    :    :    :    :
**** 1 81   1866 1602 1125 1560  647  12310000    0    0    0    0    0    0    0
DATA 2   1  1102 1194 1104  948  760  430  330  173  -88 -244 -234 -143  178  449
**** 2   2   599  646  858 1129 1190 1316 1367 1260 1281 1051  839  382  132 -193
  :  :   :     :    :    :    :    :    :    :
**** 2 81   -936 -865 -966 -962 -889 -60510000    0    0    0    0    0    0    0
DATA12   1 -2960-2972-3030-3116-3113-3275-34[illegible]-3580-3564-3743-3792-3893-4077-4132
****12   2 -4093-4117-4171-4276-4226-4158-4104-4077-4044-4021-3774-3828-3610-3573
  :  :   :     :    :    :
****12 49 -309810000    0    0    0    0    0    0    0    0    0    0    0    0
END
```

LIST	0:	no print-out of data cards after they are read
	1:	data cards are printed-out after they are read

MISFIT data point at which program should start reading data.

The labeled common /STATE/ must appear in the program and in the calling program. The variables in this common statement used in the program are:

ND number of points to be read

KSK skipping factor which determines whether each point (KSK=1) should be read or every other point (KSK=2) etc., should be read.

INIT(12) an array of 12 1-digit integers corresponding to the output channels. If 1, the first data value of the corresponding output is subtracted from each of the other data points. If 0 no subtraction is made and the data is read in as stored.

YM(200,12) the matrix in which the data is stored for a maximum of 12 channels.

Note that the increment between data points is read into the program on the first data card as shown in the example above and is defined by the variable KDT. The increment is defined as milliseconds or millidegrees and is given by the variable DT in seconds or degrees.

10. PARAM$$

Description

The subroutine PARAM carries out the parameter identification of the mathematical model.

The sum of the measured error squared is calculated and stored in the matrix OUT(1). After initializing the subroutine SETTIM is called in order to keep track of the computation time during the parameter identification cycle of the program. Next the subroutine DYN is called. This subroutine carries out the integration of the equations of motion for the nominal values of the model parameters. The following results are brought into the PARAM subroutine; the calculated response stored in the matrix YC, the error between the calculated and measured response ER = YC - YM and the weighted calculated response YW. After the "call" to the DYN subroutine is completed the relative error is calculated in the PARAM subroutine. The relative error is defined as:

$$\text{Relative error} = \sqrt{\frac{\Sigma_i (YC_i - YM_i)^2 / \sum_i YM_i^2}{NS}}$$

where all variables have been defined except for NS which is equal to the number of measured outputs ($NS \leqq 12$). The difference between the present value of the error is calculated next. If the present value of the error has increased by more that 10% of the past value the subroutine ADJUST is called. In the subroutine ADJUST the parameter values are decreased and the subroutine DYN is called to obtain new values for YC, ER and YW as has been previously described. Following the computation the subroutine DYDA is called which calculates the matrix (see ref. 1)

$$\text{DER}(M,N) = \begin{bmatrix} \frac{\partial \hat{y}(t_1)}{\partial a_1} & \frac{\partial \hat{y}(t_1)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_1)}{\partial a_N} \\ \frac{\partial \hat{y}(t_2)}{\partial a_1} & \frac{\partial \hat{y}(t_2)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_2)}{\partial a_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \hat{y}(t_M)}{\partial a_1} & \frac{\partial \hat{y}(t_M)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_M)}{\partial a_N} \end{bmatrix}$$

where

$$\frac{\partial \hat{y}(t_M)}{\partial a_i} = \frac{(y_i - y)}{da_i}.$$

Once the matrix DER has been calculated the subroutine COR is called by the PARAM subroutine. COR calculates the increment required in the parameters to reduce the error to zero between the calculated and measured time histories, (ref. 1). The increment is given in matrix form as

$$\overline{dA} = - (\overline{DER}^T \overline{DER})^{-1} \quad \overline{DER}^T \overline{ER}$$

Subroutine COR also establishes the dependency between each of the parameters. The results of the dependent analysis performed in COR are printed out after returning to the PARAM subroutine. The entire above procedure is repeated for the number of iterations that have been specified. If the error (ER) is less than 1.E-07 before the total number of iterations requested have been completed, the program terminates the identification process.

Usage

CALL PARAM (N1S, N3S, M1, DMINS, ERRMIN)

The arguments in the call serve the following functions:

N1S total number of parameters that can be estimated (max 36)

N3S total number of data points (equal to number of data points per measurement multiplied by the number of measurements)

M1 control variable used for calling the subroutine PARAM. If M1=0 the PARAM subroutine is not called; M1=1 the PARAM subroutine is called.

DMINS value at which a parameter is considered irrelevant; i.e. if the relative change in the output is less than DMINS=1.OE-07 the parameter is irrelevant. DMINS is set in the IDENT1 subroutine.

ERRMIN the minimum error the identification algorithm will stop iterating. ERRMIN=.001 was set in the IDENT1 subroutine.

All common statements in this program should not be altered by the user.

11. ADJUST$$

Description

This subroutine is called by the PARAM subroutine when the difference between the present and past value of the error has not decreased by more than 10%. The increment in the model parameters are decreased in the subroutine and the time histories for the model are again calculated. This procedure is continued until the error test has been satisfied.

Usage

CALL ADJUST

There are no arguments in the calling list of the program. All the common statements present in the program are for passing variables between the identification programs. The user is not required to make any changes or additions to this subroutine. Control from the program is returned to the PARAM subroutine.

12. DYDA

Description

After the time history for the nominal values of the parameters have been calculated the subroutine DYDA is called. This subroutine calculates the change in the output due to a change in the parameter and then stores the result in the matrix

$$DER(M,N) = \begin{bmatrix} \frac{\partial \hat{y}(t_1)}{\partial a_1} & \frac{\partial \hat{y}(t_1)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_1)}{\partial a_N} \\ \frac{\partial \hat{y}(t_2)}{\partial a_1} & \frac{\partial \hat{y}(t_2)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_2)}{\partial a_N} \\ \vdots & \vdots & & \\ \frac{\partial \hat{y}(t_M)}{\partial a_1} & \frac{\partial \hat{y}(t_M)}{\partial a_2} & \cdots & \frac{\partial \hat{y}(t_M)}{\partial a_N} \end{bmatrix}$$

where $\frac{\partial \hat{y}(t_M)}{\partial a_N}$ represents the change in the output at the t_M's time due to a change in the a_N's parameter.

Usage

CALL DYDA

The user is not required to make any changes or additions to this subroutine. All commons in DYDA are used entirely between the identification programs and do not require modification by the user. The subroutine returns control to the PARAM subroutine.

13. COR

Description

The subroutine is called by the PARAM subroutine after the derivative matrix $\overline{DER}$ and the error matrix $\overline{ER}$ have been calculated. The subroutine uses these matrices to obtain the solution of the equation

$$\overline{dA} = - (\overline{DER}^T\overline{DER})^{-1} \overline{DER}^T\overline{ER}$$

where $\overline{dA}$ represents a column vector whose elements are the incremental changes required in the parameters to force the error $\overline{ER}$ to zero. The above equation represents the least squares solution obtained by minimizing the quantity

$$(\overline{ER}+\overline{DER}\ \overline{dA})^2 = 0$$

The other important function the subroutine performance is determining which parameters of the model can be uniquely identified from a given set of measurements. A complete description of the identification technique and the problem of determining a unique set of parameters is discussed in reference 1. In particular appendix E of reference 1 gives a discussion of the subroutine COR.

Usage

CALL COR (N1, N3, THR, MODE1)

The program has four parameters in the calling list. These parameters are:

N1 number of parameters of the model $1 \leq N1 \leq 36$

N3 total number of output measurements $1 \leq N3 \leq 2400$

THR value of the threshold $0 \leq THR \leq 1$ (THR equal to zero defines the classical linear dependence between parameters)

MODE integer controlling the output; may take values from 0 to 6 (0 prints no output, 6 prints out the full information obtained from COR, between 0 and 5 a partial print out of results are given - see example).

The remaining variables needed in the program come through the common statements in the COR subroutine. The user is not required to make changes to this subroutine.

14. TRADUC$$

Description

TRADUC is used to translate logical information into parameter names in order to ease the reading of the comments in the COR subroutine.

Usage

CALL TRADUC (FOUND, N1, K)

The user makes no changes to this subroutine. The subroutine is called only by subroutine COR. The names of the parameters are given through the COMMON/WRITE/ in the array WA. The output of the subroutine is found in the array WR of the COMMON/WRITE/.

15. ERROR$$

Description

The ERROR subroutine is called after the identification of the parameters has been performed. The program calculates the relative error, the correlation, and the autocorrelation of the error and the variance of the parameters.

Usage

CALL ERROR (N1, N3, M1, V, CORR, RELERR)

The ERROR subroutine is called by the IDENT1 subroutine after the identification process is completed. The functions of each variable in the calling list are the following:

N1	number of parameters (supplied to this program)
N3	total number of data points (supplied to this program)
M1	not used in program
V	cost calculated in the subroutine and defined as $V = \Sigma (ER)^2$
CORR	the correlation defined as $CORR = 1 - \frac{\Sigma_i (YC_i - YM_i)^2 / \Sigma_i YM_i^2}{NS}$
RELERR	the relative error defined as

$$RELERR = \sqrt{\frac{\Sigma_i (YC_i - YM_i)^2 / \Sigma_i YM_i^2}{NS}}$$

The user is not required to make changes to this subroutine.

16. CRUNCH$$

Description

The CRUNCH subroutine is used to write the computed data YC on computer disk. Before the data is written on disk, it is converted to four digit numbers by the scaling factor SCALE (J) see subroutines OUTPUT and READIN. This is done to minimize the field length required for each data point when the data is written on cards. However, if the computed data (YC) after scaling is larger than a four digit number a new scale factor is calculated in the CRUNCH subroutine. The new scale factor allows the computed data to be written by a four digit number. The subroutine CRUNCH writes the new scaling factor for each data channel (maximum of 12) that is required to be rescaled. Once the data has been written on computer disk, a punched card deck of the data can be obtained.

Usage

CALL CRUNCH (NTAPE, NRUN, KSTOP, IDSET)

The user is not required to make changes in the subroutine. The arguments in the calling list are:

NTAPE	not used
NRUN	not used
KSTOP	code number indicating the last data point (usually set to 10,000 and specified on the input data cards - see data set example table 1A)

IDSET	disk number the data will be written on - specified by the user through the input data cards - see examples 1 and 2 of text

Common /STATE/ is the only common in this program. The variables used in subroutine CRUNCH from the common statement are ND, KSK, KDT, SCALE (12) and YC (200,12) which are defined as:

ND	total number of data points
KSK	skipping factor for data points
KDT	increment between data points
SCALE(12)	scaling factor - one for each of the 12 channels
YC(200,12)	the matrix containing the computed data (maximum of 200 points for each of 12 channels)

17. PLOTIN$$

Description

The PLOTIN subroutine plots the computed and measured data using the on-line printer. This plot is useful for visual examination of the computed and measured data. Each time the subroutine is called it will produce two plots. As an example, on one axis it can plot the roll moment (measured and computed) and on the other the pitch moment (measured and computed).

The time or angle corresponding to the plotted data point is printed on the far right side of the plot. The data is scaled in order to produce two plots in 101 carriage spaces of printing paper. At the end of the plot, the scaling factors required to convert the data to engineering units are given. The fortran names of these scaling factors are MEAS and COMP. The largest of these factors for each plot must be used to convert the plotted data to engineering units (see example 1 of text)

Usage

CALL PLOTIN (I1, I2, KSKIP)

The arguments in the calling list of the subroutine carry out the following functions:

I1	channel number corresponding to plot number 1 (set in subroutine IDENT1)
I2	channel number corresponding to plot number 2 (set in subroutine IDENT1)
KSKIP	skipping factor for data points (set in subroutine IDENT1)

The two common statements in the program are /STATE/ and /PLOT/. The variables used in PLOTIN from the common /STATE/ are ND, KSK, KDT, YC(200,12), YM(200,12) and from the common /PLOT/ is SYMB (4).

These variables have the following meaning:

ND	total number of data points per channel (max. 200)

KSK — skipping factor for data

KDT — time interval between measured and computed data points

YC(200,12) — matrix of computed data (maximum of 200 points and 12 channels)

YM(200,12) — matrix of measured data (maximum of 200 points and 12 channels)

SYMB(4) — array containing the symbols to be used for the plot. An option in the input data cards can suppress the calling of the plotting program if only the printed output is desired.

LIBRARY PROGRAMS

The following three subroutines are library programs that are called by the programs discussed in the text. The first two subroutines deal with integrating the equations of motion while the third subroutine is used to keep track of the computation time. The user may substitute his own integration program if the simulation of the model equations requires such programs. Likewise the computation time subroutine is not essential to the application of the identification programs and the user can take this program out of the subroutines or substitute a similar program compatible with the users computation facility. More detailed description of these subroutines can be obtained from reference 3.

18. AL INTS

Description

AL INTS uses double-precision arithmetic internally. The purpose of this subroutine is to obtain a numerical solution of the system of N (N .GE. 1) ordinary first-order differential equations Y' = F(x,Y)

where

Y', Y and F and n-th order vectors and the prime indicates differentiation with respect to the independent variable x.

Usage

The subroutine has two entries, on for set-up, one to integrate over an interval h.

A. <u>Set-Up Entry</u>

When this entry is used, the subroutine does all the necessary initialization to start integrating a x = xo. It then enters the derivative routine, which is supplied by the user, to obtain Y'(xo) and returns control to the main program. This entry must be used to restart the integration at any intermediate point x = XI such as a point of discontinuity, a point at which the user wishes to change the interval size h or the parameter K (which is defined below). In these cases xo is replaced by XI.

The CALL statement, with normal conventions for integer and floating-point number designations, is:

CALL INTS (T,N,K,EU,P,A,HMAX,HMIN,BETA,DERIV,$NNNN),

where

T	is any array of 12N + 3 cells if Adams-Moulton option is used or 4N + 3 cells if Runge-Kutta option is used
N	is the number of differential equations
K	is a code where K = 0 for Adams-Moulton variable step-size mode K = 1 for Rouge-Kutta mode K = 2 for Adams-Moulton fixed step-size mode
EU	is the upper bound of E(n + 1) for truncatation error testing done in the variable step-size mode (EU .GT. 0)
P	is used to compute the lower bound of E(n + 1) and, if used, should be P .GT. 0; if P = 0 the routine sets P = 100.0
A	is a constant used to control interval size reduction and should be A .GT. 0; if A = o the routine sets A = 1.0
HMAX	is the maximum value of h beyond which the routine should not increase h (if HMAX = 0 the routine assumes there is no upper limit on h)

HMIN	is the minimum value of h below which the routine should not decrease h (if HMIN = 0 the routine assumes there is no lower limit on h)
BETA	is the number used to increase or decrease the interval size and must be 0.0 .LE. BETA .LE. 1.0 (if BETA = 0.0 then the routine sets BETA = 0.5)
DERIV	is the name of an subroutine (supplied by the user) that evaluates the derivatives and stores them in T(4 + N) through T(4 + 2N - 1); the name of this subroutine must also appear on an EXTERNAL card. This subroutine has no arguments.
$nnnn	is an optional argument which is a statement number in the user's program, preceded by $, to which control is transferred when an error occurs in the derivative subroutine. (See <u>ERROR Return</u> below)

If K = 1 or K = 2 then the quantities EU, P, A, HMAX, HMIN, and BETA may be specified arbitrarily.

The array contains the following information:

T(2) = x, the independent variable

T(3) = h, the value of the interval size

T(4), T(5), ..., T(4+N-1)	contain the N values of the Y vector
T(4+N), ...,T(4=2N-1)	contain the N values of the Y' vector which are supplied by the derivative subroutine (i.e., DERIV)

Prior to the Set-Up entry the user must set T(2) = xo, T(3) = h, and T(4) through T(4+N-1) to the N values of the initial Y vector The array T should appear in common since it is necessarily refered to in both the main program and the subroutine that evaluates the derivatives. The integration subroutine stores N, scaled at 35, in T(1).

19. INTM

Description

This program is used to integrate one step (i.e., from x(j) to x(j+1)). When control is returned to the main program the solutions Y will be located in T(4) through T(4+N-1) and x(j+1) will be in T(2). The derivatives Y' will appear in T(4+N) through T(4+2N-1).

Usage CALL Statement for this entry is:

```
CALL INTM
```

No arguments are required for this statement.

<u>Error Return</u> (optional)

A labeled COMMON statement COMMON/ERINT/IER should be included in the derivative routine. If an error is detected by the user in this subroutine, IER should be set to 2. ALINTS will return control to the statement number specified in the calling statement. If this error return is executed, a new set-up entry must be made before any further integration may take place.

The error return is optional and may be left unused if all the differential equations are well behaved (i.e., contain no singularities, etc.).

20. SETTIM

Description

The subroutine measure time intervals (in milliseconds) of up to seven hours. Eight seperate timers may be used per task, and timing may be done either in task time (CPU time allotted to the task) or real time (elapsed wall-clock time).

Usage

The program is initialize by the statement

CALL SETTIM (NUMBER, ITYPE)

where NUMBER, and integer, is the timer (from 0 to 7) ITYPE, and integer, is the timer type (zero for CPU time, nonzero for real time)

To measure elapsed time since specified timer was initialized use the integer function:

ITIME=INTVAL(NUMBER, ITYPE)

where ITIME is the elapsed time in milliseconds

NUMBER is the timer number (from 0 to 7)

ITYPE is the timer type (must agree with the type specified

SETTIM for the timer, i.e., zero for CPU time, nonzero for real time)

APPENDIX B

COMPUTER PROGRAM LISTINGS

This appendix gives a listing in Fortran of the parameter identification programs. The three library routines that were discussed in appendix A are not listed here since their application will depend on the users parameter identification problem and the computational facility available to the user.

```
C       MAIN PROGRAM
        REAL NU1,NU2,NU,LAU,LAUT,LAM,MU,LILA1S,LILB1S,LAMX,LAMY
        COMMON/MAIN1/ T(123),A(10,10),C(10,10),E(10),BU(10),BIX,BIY,BIZ,
       1              BRAD,BRADD,BCHORD,BLOSS,SWEEP,PERRPM,BHINGE,
       2              OMEGA,BLOCK,DELTA3,BETAK,TWIST,PRECON,ALPHA,
       3              ALPHAS,LILA1S,AIRV,AIRRHO,GYIP,GYID,GPSID,GYROL,
       4              GYSPED,C2,COLECT,NU1,NU2,TIMEF,ICOUNT,PSI,
       5              IFLOQ,JCOUNT
        COMMON/DERIV1/BPREC,AIROC,AIROC1,B2,B3,B4,B5,B6,B7,B8,A25,A27,A29,
       1              A47,A49,A61,A65,A66,A81,A83,A87,A88,A101,A103,
       2              A109,A1010,GYNEW3,GYEQI,FKZETA,FCDAMP,CD,
       3              CSDAMP,MU,LAM,P1
        COMMON/OUT1/  SCALE(14)
        COMMON/SENTIV/BIGB1S,CT,IHINGE
        COMMON/OUTNEW/YMR(200,4),IROT,INEW,ISIML
        COMMON/BIAS/  RMBIAS,PMBIAS,RABIAS,PABIAS
        COMMON/INOUT1/PHI,RPHI,THET,RTHET,IREC
        COMMON/INFLOW/PSIOF,LAMX,LAMY,IFLOW
        COMMON/REVSF/ R2,R3,R4,R5,R6,R7,R8
C
C       READ PROGRAM CONTROL LOGIC
   10   READ(5,110) IFLOQ,IHINGE,IROT,INEW,ISIML,IFLOW,IREC
C
C       FOR IFLOQ= 0 FIXED S/P MOMENT INPUTS AS READ IN BY NU1 AND NU2
C       FOR IFLOQ=+1 FORCED S/P ANGLES AND MOMENTS--LOCKED GYRO MODE
C       FOR IFLOQ=-1 FORCED S/P MOMENTS ONLY
C       IREC=0 LOCKED STEADY-STATE GYRO CASE--USED WITH IFLOQ=1
C       IREC=1 LOCKED TRANSIENT GYRO CASE USED TOGETHER WITH IFLOQ=1
C       IHINGE= 0 NO HINGE OFFSET IN SIMULATION
C       IHINGE= 1 HINGE OFFSET INCLUDED IN SIMULATION
C       IROT= 0 SIMULATION AND IDENTIFICATION IN ROTATING COORDINATES
C       IROT= 1 SIMULATION AND IDENTIFICATION IN STATIONARY COORDINATES
C       INEW= 0 27 PARAMETER NEWP SUBROUTINE
C       ISIML=0 PROGRAM RUN IN IDENTIFICATION MODE
C       ISIML=1 PROGRAM RUN IN SIMULATION MODE
C       INEW= 1 23 PARAMETER NEWP SUBROUTINE
C       INEW=-1 USED TOGETHER WITH IFLOQ=1  FORCED S/P ANGLES WITH S/P
C               MOMENTS READ FROM INPUT DATA CARDS
C       IFLOW=1 INFLOW VALUE READ FROM INPUT DATA CARDS STEADY TERM AND
C               FIRST HARMONIC
C       IFLOW=0 INFLOW VALUE CALCULATED IN PROGRAM--STEADY TERM ONLY
C
C       READ BLADE CONSTANTS
        READ(5,100) BIX,BIY,BIZ,BRAD,BCHORD,BLOSS
        READ(5,100) SWEEP,RPM,BHINGE,BLOCK,DELTA3,BETAK,P1
```

```
      READ(5,100) TWIST,PRECON,CT,AIRV,AIRRHO,ALPHA,ALPHAS
      READ(5,105) LILA1S,LILB1S,BIGA1S,BIGB1S,CD,LAM,LAMX,LAMY
C
C     INITIAL CONDITIONS ON BLADE
      READ(5,100) BETA1,RBETA1,BETA2,RBETA2,BETA3,RBETA3,COLECT
C
C     READ GYRO  CONSTANTS
      READ(5,100) GYIP,GYID,GPSIO,GYROL,GYRPM,C2
      READ(5,100) FKZETA,FCDAMP,CSDAMP,PSIOF
C
C     INITIAL CONDITIONS ON GYRO
      READ(5,100) DELT1,RDELT1,DELT2,RDELT2,NU1,NU2
      READ(5,105) PHI,RPHI,THET,RTHET,RMBIAS,PMBIAS,RABIAS,PABIAS
C
C     INITIAL CONDITIONS FOR INTEGRATION ROUTINE
      READ(5,100) DELPSI,REVS,REVO,REVF,TRUNC
C
C     SCALE FACTORS FOR PLOTTING PROGRAM
      READ(5,100) (SCALE(I),I=1,14)
  100 FORMAT(7E10.5)
  105 FORMAT(8E10.5)
  110 FORMAT(7I2)
C
C     WRITE INPUT QUANTITIES
      WRITE(6,130)
  130 FORMAT(1H1,36X,58HTWO DEGREE OF FREEDOM HELICOPTER BLADE AND GYRO
     1SIMULATION//)
C
      WRITE(6,132)
  132 FORMAT(55X,21HPROGRAM CONTROL LOGIC)
      WRITE(6,134) IFLOQ,IHINGE,IROT,INEW,ISIML,IFLOW,IREC
  134 FORMAT(45X,7H IFLOQ=I2,3X,8H IHINGE=I2,3X,6H IROT=I2,3X,6H INEW=I2
     1,7H ISIML=I2,7H IFLOW=I2,6H IREC=I2//)
C
      WRITE(6,140)
  140 FORMAT(58X,16HBLADE PARAMETERS)
      WRITE(6,150) BIX,BIY,BIZ,BRAD,       BCHORD,BLOSS,SWEEP,   RPM,
     1BHINGE,       BLOCK,DELTA3,BETAK,P1,TWIST,PRECON,CT,AIRV,AIRRHO,
     2ALPHA,ALPHAS,LILA1S,LILB1S,BIGA1S,BIGB1S,CD,LAM,LAMX,LAMY
  150 FORMAT(29X,5H BIX=F6.2, 5H BIY=F6.2, 5H BIZ=F6.2, 8H RADIUS=F4.1,
     1                     7H CHORD=F4.2, 12H BLADE LOSS=F3.2//19X,
     27H SWEEP=F6.2, 5H RPM=F6.1,                16H HINGE POSITION=F4.2,
     3               6H LOCK=F6.3, 8H DELTA3=F5.2, 8H K BETA=E12.6,4H P1=
     4F6.2//24X, 7H TWIST=F6.2, 9H PRECONE=F6.3, 5H  CT=F7.4,
     514H AIR VELOCITY=F7.2,13H AIR DENSITY=F7.6,  7H ALPHA=F6.3,
```

```
     68H ALPHAS=F6.3// 5X,12H LITTLE A1S=F6.3, 12H LITTLE B1S=F6.3,
     79H BIG A1S=F6.3,  9H BIG B1S=F6.3,4H CD=F5.3,5H LAM=F8.6,
     86H LAMX=F8.6,  6H LAMY=F8.6//)
      WRITE(6,160)
  160 FORMAT(54X,24HBLADE INITIAL CONDITIONS)
      WRITE(6,170) BETA1,RBETA1,BETA2,RBETA2,BETA3,RBETA3,COLECT
  170 FORMAT(2X,7H BETA1=F7.3, 12H BETA1 RATE=F7.2, 7H BETA2=F7.3,
     112H BETA2 RATE=F7.2, 7H BETA3=F7.3, 12H BETA3 RATE=F7.2,
     212H COLLECTIVE=F5.1//)
C
      WRITE(6,180)
  180 FORMAT(58X,15HGYRO PARAMETERS)
      WRITE(6,190) GYIP,GYID,GPSIO,GYROL,GYRPM ,C2,FKZETA,FCDAMP,CSDAMP,
     1PSIOF
  190 FORMAT(15X,4H IP=F6.3, 4H ID=F6.3, 12H GYRO ANGLE=F5.1,
     113H GYRO DAMP L=F6.2,  10H GYRO RPM=F7.1, 25H MECHANICAL ADVANTAGE
     2 C2=F5.3//10X,  27HFEATHERING RESTRAINT KZETA=F6.2,26H FEATHERING
     3DAMPING CZETA=F6.2,27H SWASHPLATE DAMPING CSDAMP=F6.2, 7H PSIOF=F6
     3.2//)
C
      WRITE(6,200)
  200 FORMAT(53X,26HINITIAL CONDITIONS ON GYRO)
      WRITE(6,210) DELT1,RDELT1,DELT2,RDELT2,NU1,NU2
  210 FORMAT(22X,8H DELTA1=F7.3, 13H DELTA1 RATE=F8.2, 8H DELTA2=F7.3,
     113H DELTA2 RATE=F8.2,5H NU1=F6.2, 5H NU2=F6.2//)
      WRITE(6,215) PHI,RPHI,THET,RTHET,RMBIAS,PMBIAS,RABIAS,PABIAS
  215 FORMAT( 2X,5H PHI=F7.3,6H RPHI=F8.2,6H THET=F7.3, 7H RTHET=F8.2,
     1 8H RMBIAS=F8.2, 8H PMBIAS=F8.2, 8H RABIAS=F6.3, 8H PABIAS=F6.3//)
C
      WRITE(6,220)
  220 FORMAT(32X,68HINITIAL CONDITIONS FOR STARTING AND STOPPING THE INT
     1EGRATION ROUTINE)
      WRITE(6,230) DELPSI,REVS,REVO,REVF,TRUNC
  230 FORMAT(32X,8H DELPSI=F5.2, 7H  REVS=F5.2,  7H  REVO=F5.2,
     1  7H  REVF=F5.2, 7H TRUNC=E10.4//)
      WRITE(6,240)
  240 FORMAT(49X,34HSCALE FACTORS FOR PLOTTING PROGRAM)
      WRITE(6,241)
  241 FORMAT(20X,87HSCALE(1)     SCALE(2)     SCALE(3)     SCALE(4)
     1SCALE(5)     SCALE(6)     SCALE(7) )
      WRITE(6,247) (SCALE(I),I=1,7)
      WRITE(6,245)
  245 FORMAT(20X,87HSCALE(8)     SCALE(9)     SCALE(10)    SCALE(11)
     1SCALE(12)    SCALE(13)    SCALE(14))
      WRITE(6,247) (SCALE(I),I=8,14)
```

```
  247 FORMAT(19X,7(1PE10.3,3X)/)
C
C     CONVERT ALL ANGLES TO RADIANS
      RADIAN=57.29578
      SWEEP=SWEEP/RADIAN
      DELTA3=DELTA3/RADIAN
      TWIST=TWIST/RADIAN
      PRECON=PRECON/RADIAN
      ALPHA=ALPHA/RADIAN
      ALPHAS=ALPHAS/RADIAN
      BETA1=BETA1/RADIAN
      RBETA1=RBETA1/RADIAN
      BETA2=BETA2/RADIAN
      RBETA2=RBETA2/RADIAN
      BETA3=BETA3/RADIAN
      RBETA3=RBETA3/RADIAN
      COLECT=COLECT/RADIAN
      GPSIO=GPSIO/RADIAN
      LILA1S=LILA1S/RADIAN
      LILB1S=LILB1S/RADIAN
      BIGA1S=BIGA1S/RADIAN
      BIGB1S=BIGB1S/RADIAN
      DELT1=DELT1/RADIAN
      RDELT1=RDELT1/RADIAN
      DELT2=DELT2/RADIAN
      RDELT2=RDELT2/RADIAN
      DELPSI=DELPSI/RADIAN
      PHI=PHI/RADIAN
      RPHI=RPHI/RADIAN
      THET=THET/RADIAN
      RTHET=RTHET/RADIAN
      RABIAS=RABIAS/RADIAN
      PABIAS=PABIAS/RADIAN
C
      OMEGA=6.283185*RPM/60.0
C     CONVERT AIR VELOCITY TO FT/SEC
      AIRV=1.68781*AIRV
      MU=AIRV*COS(ALPHA)/(OMEGA*BRAD)
C     CONVERT STEP SIZE IN DEG. AND NUMBER OF REV. OF THE ROTOR TO SEC.
C     FOR USE IN INTEGRATION ROUTINE
      DELTAT=DELPSI/OMEGA
      TIMES= REVS*6.283185/OMEGA
      TIMEO= REVO*6.283185/OMEGA
      TIMEF= REVF*6.283185/OMEGA
C     SET BLADE RADIUS SUB ZERO TO HINGE OFFSET
```

```
      BRADO=BHINGE
C
C     SET GYSPED EQUAL TO RATIO OF ROTOR SPEED TO GYRO SPEED
      GYSPED=GYRPM/RPM
C
C     CALCULATE INFLOW
C     LAUT=LAMBDA/MU
      IF(MU .EQ. 0.0) GO TO 265
C
C     IF THE INFLOW IS SPECIFIED FROM INPUT DATA CARDS IFLOW MUST BE 1
      IF(IFLOW .EQ. 1) GO TO 268
      ALFC=ALPHAS-BIGB1S
      LAUT=ALFC -.5*CT/MU**2
  250 NU=0.5*CT/(MU*SQRT(1+LAUT**2))
      LAU=ALFC-NU/MU
      ERR=ABS(1.0-LAU/LAUT)
      IF(ERR .LT. .001) GO TO 260
      LAUT=LAU
      GO TO 250
  260 LAM=LAU*MU
      GO TO 268
  265 LAM=0.0
C
C     WRITE CALCULATED PARAMETERS
  268 WRITE(6,270)
  270 FORMAT(1H0,47X,36HPARAMETERS CALULATED FROM INPUT DATA)
      WRITE(6,280) OMEGA,GYSPED,MU,LAM
  280 FORMAT(42X, 7H OMEGA=F7.3, 8H GYSPED=F6.2, 4H MU=F6.3,
     18H LAMBDA=F8.6/)
      WRITE(6,285) DELTAT,TIMES,TIMEO,TIMEF
  285 FORMAT(31X,8H DELTAT=E10.4, 7H TIMES=E10.4, 7H TIMEO=E10.4,
     1 7H TIMEF=E10.4)
C
C     INITIALIZE THE A AND BU MATRICES
      DO 300 I=1,10
      DO 300 J=1,10
  300 A(I,J)=0.0
      DO 340 I=1,10
  340 BU(I)=0.0
C
C     SET INITIAL CONDITIONS ON T-BLOCK
      T(1)=10.
      T(2)=TIMES
      T(3)=DELTAT
      T(4)=DELT1
```

```
      T(5)=RDELT1
      T(6)=DELT2
      T(7)=RDELT2
      T(8)=BETA1
      T(9)=RBETA1
      T(10)=BETA2
      T(11)=RBETA2
      T(12)=BETA3
      T(13)=RBETA3
      ICOUNT=1
C
C     CALCULATE CONSTANTS FOR USE IN DERIVATIVE ROUTINE
      OMEGA2=OMEGA**2
      C22=C2**2
      GYEQI=GYID+1.5*BIX*C22
      BIXYZG=(BIX+BIY-BIZ)/GYEQI
      BIZYG= (BIZ-BIY)/GYEQI
      BIZXG= (BIZ-BIX)/GYEQI
      BIXYZB=(BIX+BIY-BIZ)/BIY
      BIZB  =BIZ/BIY
      GYNEW1=(GYIP*GYSPED-GYID)*OMEGA2/GYEQI
      GYNEW2=(GYIP*GYSPED-2.0*GYID)*OMEGA/GYEQI
      GYNEW3=GYID*OMEGA2/GYEQI
      GYNEW4=1.5*C22*BIZYG*OMEGA2
      GYNEW5=(1.5*C22*FKZETA)/GYEQI
      GYNEW6=(1.5*C22*FCDAMP)/GYEQI
      GYNEW7=C22*SWEEP*BIXYZG*OMEGA
      GYNEW8=C2*SWEEP*BIZYG*OMEGA2
      GYNEW9=C2*BIXYZG*OMEGA
      SWASHD=CSDAMP/GYEQI
      GYROD=+2.0*GYID*GYROL*OMEGA/GYEQI
      GYCOUP=C2*SWEEP*BETAK/GYEQI
      BCOUP1=SWEEP*OMEGA2*C2
      BCOUP2=BIXYZB*OMEGA*C2
      BCOUP3=BETAK/BIY
      BCOUP4=BIZB*OMEGA2
      BCOUP5=BIXYZB*SWEEP*OMEGA
      BPREC=BCOUP3*PRECON
      HCONST=BIY*SWEEP*C2/GYEQI
      IF(IHINGE .EQ. 0) GO TO 360
      HINGOF=+1.5*BRADO*OMEGA2/(BRAD-BRADO)
      GO TO 370
  360 HINGOF=0.0
  370 A(1,2) = 1.0
      A(2,1)=-GYNEW1-GYNEW4-GYNEW5
```

```
      A(2,2)=-GYROD-1.5*GYNEW7-GYNEW6-SWASHD
      A(2,3)=OMEGA*SWASHD
      A(2,4)=-GYNEW2
C
      A25= GYCOUP+2.0*GYNEW8+HCONST*HINGOF
C
      A(2,6)=-GYNEW9
C
      A27=-.5*GYCOUP-GYNEW8-.5*HCONST*HINGOF
C
      A(2,8)=+.5*GYNEW9
C
      A29=-.5*GYCOUP-GYNEW8-.5*HCONST*HINGOF
C
      A(2,10)=+.5*GYNEW9
      A(3,4) =+1.0
      A(4,1)=-OMEGA*SWASHD
      A(4,2)= GYNEW2
      A(4,3)=-GYNEW1-GYNEW4-GYNEW5
      A(4,4)=-GYROD-1.5*GYNEW7-GYNEW6-SWASHD
C
      A47=+.8667*GYCOUP+1.732*GYNEW8+.8667*HCONST*HINGOF
C
      A(4,8)=-.8667*GYNEW9
C
      A49=-.8667*GYCOUP-1.732*GYNEW8-.8667*HCONST*HINGOF
C
      A(4,10)=+.8667*GYNEW9
      A(5,6) = 1.0
C
      A61=-BCOUP1
C
      A(6,2)=+BCOUP2
      A(6,4)=+0.0
C
      A65=-BCOUP3-BCOUP4-HINGOF
      A66=-BCOUP5
C
      A(6,7)=+0.0
      A(6,8)=+0.0
      A(6,9)=+0.0
      A(6,10)=+0.0
      A(7,8) = 1.0
C
      A81=+.5*BCOUP1
```

```
C
      A(8,2)=-.5*BCOUP2
C
      A83=-.8667*BCOUP1
C
      A(8,4)=+.8667*BCOUP2
      A(8,5)=+0.0
      A(8,6)=+0.0
C
      A87=-BCOUP3-BCOUP4-HINGOF
      A88=-BCOUP5
C
      A(8,9)=+0.0
      A(8,10)=+0.0
      A(9,10)=+1.0
C
      A101=+.5*BCOUP1
C
      A(10,2)=-.5*BCOUP2
C
      A103=+.8667*BCOUP1
C
      A(10,4)=-.8667*BCOUP2
      A(10,5)=+0.0
      A(10,6)=+0.0
      A(10,7)=+0.0
      A(10,8)=+0.0
C
      A109=-BCOUP3-BCOUP4-HINGOF
      A1010=-BCOUP5
C
      BLOCK3=(BCHORD*CD*AIRRHO*BRAD**4)/BIX
      AIROC1=BLOCK3*BIX*OMEGA2/2.0
      AIROC =BLOCK*OMEGA2/2.0
      B2=BLOSS**2/2.0-BRADO**2/(2.*BRAD**2)
      B3=BLOSS**3/3.0-BRADO**3/(3.*BRAD**3)
      B4=BLOSS**4/4.0-BRADO**4/(4.*BRAD**4)
      B5=BLOSS**5/5.0-BRADO**5/(5.*BRAD**5)
      B6=0.2500-BRADO**4/(4.0*BRAD**4)
      B7=0.3333-BRADO**3/(3.0*BRAD**3)
      B8=0.5000-BRADO**2/(2.0*BRAD**2)
C
C     CALCULATE B'S FOR REVERSE FLOW REGION (DEFINED AS R'S)
      R2  =(BLOSS**2-MU**2)/2.
      R3  =(BLOSS**3-MU**3)/3.
```

```
      R4   =(BLOSS**4-MU**4)/4.
      R5   =(BLOSS**5-MU**5)/5.
      R6   =(1.-MU**4)/4.0
      R7   =(1.-MU**3)/3.0
      R8   =(1.-MU**2)/2.0
      PSI=OMEGA*T(2)
C
  400 GK=GK
      CALL IDENT1
  500 GK=GK
      STOP
      END
```

```
      SUBROUTINE NEWP
      REAL NU,LAU,LAUT,LAM,MU,LILA1S,LILB1S,LAMX,LAMY
C
      COMMON/PRM/   IDUM1(74),DUM1(36),P(36),DUM2(88908)
      COMMON/COM1/  L0,N1,DUMARY(120)
C
      COMMON/MAIN1/ T(123),A(10,10),C(10,10),E(10),BU(10),BIX,BIY,BIZ,
     1              BRAD,BRADO,BCHORD,BLOSS,SWEEP,PERRPM,BHINGE,
     2              OMEGA,BLOCK,DELTA3,BETAK,TWIST,PRECON,ALPHA,
     3              ALPHAS,LILA1S,AIRV,AIRRHO,GYIP,GYID,GPSIO,GYROL,
     4              GYSPED,C2,COLECT,NU1,NU2,TIMEF,ICOUNT,PSI,
     5              IFLOQ,JCOUNT
      COMMON/DERIV1/BPREC,AIROC,AIROC1,B2,B3,B4,B5,B6,B7,B8,A25,A27,A29,
     1              A47,A49,A61,A65,A66,A81,A83,A87,A88,A101,A103,
     2              A109,A1010,GYNEW3,GYEQI,FKZETA,FCDAMP,CD,
     3              CSDAMP,MU,LAM,P1
      COMMON/SENTIV/BIGB1S,CT,IHINGE
      COMMON/OUTNEW/YMR(200,4),IROT,INEW,ISIML
      COMMON/INFLOW/PSIOF,LAMX,LAMY,IFLOW
      COMMON/REVSF/ R2,R3,R4,R5,R6,R7,R8
C
      IF(ISIML .EQ. 1) RETURN
      IF(N1.EQ.0)    RETURN
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21),N1
   21 TWIST=P(21)/57.29578
   20 GYID=P(20)
   19 GYIP=P(19)
   18 GYSPED=P(18)
   17 BIX=P(17)
   16 FCDAMP=P(16)
   15 CSDAMP=P(15)
   14 MU=P(14)
   13 CD=P(13)
   12 PRECON=P(12)/57.29578
   11 BRADO=P(11)
   10 LAMY=P(10)
    9 LAMX=P(9)
    8 LAM=P(8)
    7 C2=P(7)
    6 SWEEP=P(6)/57.29578
    5 BLOCK=P(5)
    4 BETAK=P(4)
    3 COLECT=P(3)/57.29578
    2 BLOSS=P(2)
    1 BIY=P(1)
```

```
      BIZ=BIY+BIX
C
      AIRV=MU*OMEGA*BRAD/COS(ALPHA)
C
      IF(MU .EQ. 0.0) GO TO 265
      GO TO 268
  265 LAM=0.0
  268 OMEGA2=OMEGA**2
C
C     IF INERTIA, BETAK AND LOCK NUMBER ARE TO BE CALCULATED AS
C     FUNCTIONS OF HINGE POSITON INEW=1
      IF(INEW .EQ. 1) GO TO 270
      GO TO 290
C
  270 BIZ=BIY+0.4
      BETAK=BIY*OMEGA2*(P1**2-1.0-1.5*BRADO/(BRAD-BRADO))
      BLOCK=(BCHORD*AIRRHO*6.28318*BRAD**4)/BIY
      WRITE(6,280) BRADO,BIY,BIZ,BETAK,BLOCK
  280 FORMAT(2X,' BRADO='E12.6,' BIY='F6.2,' BIZ='F6.2,' BETAK='E12.6,
     1' BLOCK='F4.2)
C
  290 C22=C2**2
      GYEQI=GYID+1.5*BIX*C22
      BIXYZG=(BIX+BIY-BIZ)/GYEQI
      BIZYG= (BIZ-BIY)/GYEQI
      BIZXG= (BIZ-BIX)/GYEQI
      BIXYZB=(BIX+BIY-BIZ)/BIY
      BIZB  =BIZ/BIY
      GYNEW1=(GYIP*GYSPED-GYID)*OMEGA2/GYEQI
      GYNEW2=(GYIP*GYSPED-2.0*GYID)*OMEGA/GYEQI
      GYNEW3=GYID*OMEGA2/GYEQI
      GYNEW4=1.5*C22*BIZYG*OMEGA2
      GYNEW5=(1.5*C22*FKZETA)/GYEQI
      GYNEW6=(1.5*C22*FCDAMP)/GYEQI
      GYNEW7=C22*SWEEP*BIXYZG*OMEGA
      GYNEW8=C2*SWEEP*BIZYG*OMEGA2
      GYNEW9=C2*BIXYZG*OMEGA
      SWASHD=CSDAMP/GYEQI
      GYROD=+2.0*GYID*GYROL*OMEGA/GYEQI
      GYCOUP=C2*SWEEP*BETAK/GYEQI
      BCOUP1=SWEEP*OMEGA2*C2
      BCOUP2=BIXYZB*OMEGA*C2
      BCOUP3=BETAK/BIY
      BCOUP4=BIZB*OMEGA2
      BCOUP5=BIXYZB*SWEEP*OMEGA
```

```
      BPREC=BCOUP3*PRECON
      HCONST=BIY*SWEEP*C2/GYEQI
      IF(IHINGE .EQ. 0) GO TO 360
      HINGOF=+1.5*BRADO*OMEGA2/(BRAD-BRADO)
      GO TO 370
  360 HINGOF=0.0
  370 A(1,2) = 1.0
      A(2,1)=-GYNEW1-GYNEW4-GYNEW5
      A(2,2)=-GYROD-1.5*GYNEW7-GYNEW6-SWASHD
      A(2,3)=OMEGA*SWASHD
      A(2,4)=-GYNEW2
C
      A25= GYCOUP+2.0*GYNEW8+HCONST*HINGOF
C
      A(2,6)=-GYNEW9
C
      A27=-.5*GYCOUP-GYNEW8-.5*HCONST*HINGOF
C
      A(2,8)=+.5*GYNEW9
C
      A29=-.5*GYCOUP-GYNEW8-.5*HCONST*HINGOF
C
      A(2,10)=+.5*GYNEW9
      A(3,4) =+1.0
      A(4,1)=-OMEGA*SWASHD
      A(4,2)= GYNEW2
      A(4,3)=-GYNEW1-GYNEW4-GYNEW5
      A(4,4)=-GYROD-1.5*GYNEW7-GYNEW6-SWASHD
C
      A47=+.8667*GYCOUP+1.732*GYNEW8+.8667*HCONST*HINGOF
C
      A(4,8)=-.8667*GYNEW9
C
      A49=-.8667*GYCOUP-1.732*GYNEW8-.8667*HCONST*HINGOF
C
      A(4,10)=+.8667*GYNEW9
      A(5,6) = 1.0
C
      A61=-BCOUP1
C
      A(6,2)=+BCOUP2
      A(6,4)=+0.0
C
      A65=-BCOUP3-BCOUP4-HINGOF
      A66=-BCOUP5
```

```
C
      A(6,7)=+0.0
      A(6,8)=+0.0
      A(6,9)=+0.0
      A(6,10)=+0.0
      A(7,8) = 1.0
C
      A81=+.5*BCOUP1
C
      A(8,2)=-.5*BCOUP2
C
      A83=-.8667*BCOUP1
C
      A(8,4)=+.8667*BCOUP2
      A(8,5)=+0.0
      A(8,6)=+0.0
C
      A87=-BCOUP3-BCOUP4-HINGOF
      A88=-BCOUP5
C
      A(8,9)=+0.0
      A(8,10)=+0.0
      A(9,10)=+1.0
C
      A101=+.5*BCOUP1
C
      A(10,2)=-.5*BCOUP2
C
      A103=+.8667*BCOUP1
C
      A(10,4)=-.8667*BCOUP2
      A(10,5)=+0.0
      A(10,6)=+0.0
      A(10,7)=+0.0
      A(10,8)=+0.0
C
      A109=-BCOUP3-BCOUP4-HINGOF
      A1010=-BCOUP5
      BLOCK3=(BCHORD*CD*AIRRHO*BRAD**4)/BIX
      AIROC1=BLOCK3*BIX*OMEGA2/2.0
      AIROC=BLOCK*OMEGA2/2.0
      AIROC2= 2.25*AIROC*KLAMB*(P1**2-1.)/(BLOCK*BLOSS**2)
      B2=BLOSS**2/2.0-BRADO**2/(2.*BRAD**2)
      B3=BLOSS**3/3.0-BRADO**3/(3.*BRAD**3)
      B4=BLOSS**4/4.0-BRADO**4/(4.*BRAD**4)
```

```
      B5=BLOSS**5/5.0-BRADO**5/(5.*BRAD**5)
      B6=0.2500-BRADO**4/(4.0*BRAD**4)
      B7=0.3333-BRADO**3/(3.0*BRAD**3)
      B8=0.5000-BRADO**2/(2.0*BRAD**2)
C
C     CALCULATE B'S FOR REVERSE FLOW REGION (DEFINED AS R'S)
      R2  =(BLOSS**2-MU**2)/2.
      R3  =(BLOSS**3-MU**3)/3.
      R4  =(BLOSS**4-MU**4)/4.
      R5  =(BLOSS**5-MU**5)/5.
      R6  =(1.-MU**4)/4.0
      R7  =(1.-MU**3)/3.0
      R8  =(1.-MU**2)/2.0
  500 GK=GK
      RETURN
      END
```

```
      SUBROUTINE DERIV
      REAL MU,NU1,NU2,LAM,LAMX,LAMY
      DIMENSION AIR01(3),AIR02(3),AIR03(3),AIR04(3),AIR05(3),AIR06(3),
     1                AS(10),AIR07(3),AIR08(3)
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1                XINIT(24),YC(200,12),YM(200,12),YW(2400)
C
      COMMON/MAIN1/ T(123),A(10,10),C(10,10),E(10),BU(10),BIX,BIY,BIZ,
     1                BRAD,BRADO,BCHORD,BLOSS,SWEEP,PERRPM,BHINGE,
     2                OMEGA,BLOCK,DELTAS,BETAK,TWIST,PRECON,ALPHA,
     3                ALPHAS,LILA1S,AIRV,AIRRHO,GYIP,GYID,GPSIO,GYROL,
     4                GYSPED,C2,COLECT,NU1,NU2,TIMEF,ICOUNT,PSI,
     5                IFLOQ,JCOUNT
      COMMON/DERIV1/BPREC,AIROC,AIROC1,B2,B3,B4,B5,B6,B7,B8,A25,A27,A29,
     1                A47,A49,A61,A65,A66,A81,A83,A87,A88,A101,A103,
     2                A109,A1010,GYNEW3,GYEQI,FKZETA,FCDAMP,CD,
     3                CSDAMP,MU,LAM,P1
      COMMON/OUTNEW/YMR(200,4),IROT,INEW,IGYRO
      COMMON/INFLOW/PSIOF,LAMX,LAMY,IFLOW
      COMMON/REVSF/ R2,R3,R4,R5,R6,R7,R8
C
      PSI=OMEGA*T(2)
C
C     TEST IF FORCED S/P MOMENTS FROM MEASURED DATA ONLY OR FIXED S/P
C     MOMENTS OR FORCED S/P ANGLES AND MOMENTS
      IF(IFLOW) 10,30,25
   10 NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
      GO TO 30
   25 IF(INEW .EQ. -1) GO TO 27
      NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
   27 T(4)=YMR(ICOUNT,1)
      T(6)=YMR(ICOUNT,2)
      T(5)=YMR(ICOUNT,3)
      T(7)=YMR(ICOUNT,4)
C
C     CALCULATE TIME VARYING TERMS FOR A AND BU MATRIX
   30 NBLD = 3
      BLTOBL = 2.*3.141592/NBLD
C
      DO 40 I=1,3
      ARG = PSI + BLTOBL*FLOAT(I-1)
      CS = COS(ARG)
```

```
      SN = SIN(ARG)
C
C     CHECK FOR REVERSE FLOW REGION BETWEEN 225 AND 315 DEGREES
      IF(SN.LT.-0.7) GO TO 38
      AIRO1(I) = ( B4 + MU*B3*(SN+SWEEP*CS))/OMEGA
      AIRO2(I)=B4 + MU*(2.*B3*(SN+SWEEP*CS) + MU*B2*SN*(SN+2.*SWEEP*CS))
      AIRO3(I)=SWEEP*B4 + MU*(B3*(CS+SWEEP*SN) + MU*B2*CS*(SN+SWEEP*CS))
      AIRO4(I)=B5 + MU*(2.*B4*(SN+SWEEP*CS) + MU*B3*SN*(SN+2.*SWEEP*CS))
      AIRO5(I) = B3 + MU*B2*(SN+SWEEP*CS)
      AIRO6(I)=B6 + MU*(2.*B7*(SN+SWEEP*CS) + MU*B8*SN*(SN+2.*SWEEP*CS))
C
C     TERMS FOR NONUNIFORM INFLOW ARE CALCULATED BY AIRO7 AND AIRO8
      AIRO7(I)= B4*CS+MU*B3*(CS*SN+SWEEP*CS*CS)
      AIRO8(I)= B4*SN+MU*B3*(SN*SN+SWEEP*CS*SN)
      GO TO 40
   38 AIRO1(I) = ( R4 + MU*R3*(SN+SWEEP*CS))/OMEGA
      AIRO2(I)=R4 + MU*(2.*R3*(SN+SWEEP*CS) + MU*R2*SN*(SN+2.*SWEEP*CS))
      AIRO3(I)=SWEEP*R4 + MU*(R3*(CS+SWEEP*SN) + MU*R2*CS*(SN+SWEEP*CS))
      AIRO4(I)=R5 + MU*(2.*R4*(SN+SWEEP*CS) + MU*R3*SN*(SN+2.*SWEEP*CS))
      AIRO5(I) = R3 + MU*R2*(SN+SWEEP*CS)
      AIRO6(I)=R6 + MU*(2.*R7*(SN+SWEEP*CS) + MU*R8*SN*(SN+2.*SWEEP*CS))
C
C     TERMS FOR NONUNIFORM INFLOW ARE CALCULATED BY AIRO7 AND AIRO8
      AIRO7(I)= R4*CS+MU*R3*(CS*SN+SWEEP*CS*CS)
      AIRO8(I)= R4*SN+MU*R3*(SN*SN+SWEEP*CS*SN)
   40 CONTINUE
C     DEFINE ELEMENTS OF A-MATRIX
C
      TNDLT = TAN(DELTA3)
      DUMMY = AIROC1*C2/GYEQI
      A(2,5)=A25+DUMMY*AIRO6(1)
      A(2,7)=A27-.50*DUMMY*AIRO6(2)
      A(2,9)=A29-.50*DUMMY*AIRO6(3)
      A(4,7)=A47+.8667*DUMMY*AIRO6(2)
      A(4,9)=A49-.8667*DUMMY*AIRO6(3)
      A(6,1)=+C2*AIROC*AIRO2(1)+A61
      A(6,5)=-AIROC*(AIRO3(1)-AIRO2(1)*TNDLT)+A65
      A(6,6)=-AIROC*AIRO1(1)+A66
      A(8,1)=-.5*C2*AIROC*AIRO2(2)+A81
      A(8,3)=+.8667*C2*AIROC*AIRO2(2)+A83
      A(8,7) = -AIROC*(AIRO3(2)  - AIRO2(2)*TNDLT) + A87
      A(8,8)=-AIROC*AIRO1(2)+A88
      A(10,1)=-.5*C2*AIROC*AIRO2(3)+A101
      A(10,3)=-.8667*C2*AIROC*AIRO2(3)+A103
      A(10,9)= -AIROC*(AIRO3(3)  - AIRO2(3)*TNDLT) + A109
```

```
      A(10,10)=-AIROC*AIR01(3)+A1010
C
C     CONSTRUCT THE AX MATRIX
   90 DO 100 I=1,10
      AS(I)=0.0
      DO 100 J=1,10
  100 AS(I)=AS(I)+A(I,J)*T(J+3)
C     CONSTRUCT THE BU MATRIX
C
C
      PSIG = PSI + GPSIO
      CSPSI = COS(PSIG)
      SNPSI = SIN(PSIG)
      BU(2) = ( NU1*SNPSI - NU2*CSPSI)/GYEQI
      BU(4) = (+NU1*CSPSI + NU2*SNPSI)/GYEQI
      BU(6)=+AIROC*AIR02(1)*COLECT
     1      +AIROC*AIR05(1)*LAM+AIROC*(LAMX*AIR07(1)+LAMY*AIR08(1))
     2      +AIROC*AIR04(1)*TWIST
     3      +BPREC
      BU(8)=+AIROC*AIR02(2)*COLECT
     1      +AIROC*AIR05(2)*LAM+AIROC*(LAMX*AIR07(2)+LAMY*AIR08(2))
     2      +AIROC*AIR04(2)*TWIST
     3      +BPREC
      BU(10)=+AIROC*AIR02(3)*COLECT
     1       +AIROC*AIR05(3)*LAM+AIROC*(LAMX*AIR07(3)+LAMY*AIR08(3))
     2       +AIROC*AIR04(3)*TWIST
     3       +BPREC
      DO 200 I=1,10
  200 T(I+13)=AS(I)+BU(I)
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
      REAL NU1,NU2
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
      COMMON/PRM/   MN(37),MNS(37),SENSIV(36),P(36),AMIN(36),AMAX(36),
     1              DA(36),ER(2400),DER(2400,36)
      COMMON/WGHT/  WEIGHT(12)
C
      COMMON/MAIN1/ T(123),A(10,10),C(10,10),E(10),BU(10),BIX,BIY,BIZ,
     1              BRAD,BRADO,BCHORD,BLOSS,SWEEP,PERRPM,BHINGE,
     2              OMEGA,BLOCK,DELTA3,BETAK,TWIST,PRECON,ALPHA,
     3              ALPHAS,LILA1S,AIRV,AIRRHO,GYIP,GYID,GPSIO,GYROL,
     4              GYSPED,C2,COLECT,NU1,NU2,TIMEF,ICOUNT,PSI,
     5              IFLOQ,JCOUNT
      COMMON/PRIN/  PSI1(200),SPMRP(200,2),BET1(200),BET2(200),BET3(200)
     1              ,RDELT1(200),RDELT2(200)
      COMMON/OUTNEW/YMR(200,4),IROT,INEW,ISIML
      COMMON/BIAS/  RMBIAS,PMBIAS,RABIAS,PABIAS
C
      RADIAN=57.29578
      PSI1(ICOUNT)=PSI*RADIAN
      CPSI=COS(PSI)
      SPSI=SIN(PSI)
      CPSIO=COS(PSI+GPSIO)
      SPSIO=SIN(PSI+GPSIO)
      IF(IROT .EQ. 1) GO TO 15
C
C     CONVERT STATIONARY BIAS TO ROTATING COORDIANTES
      RBIASR=RMBIAS*CPSI+PMBIAS*SPSI
      PBIASR=-RMBIAS*SPSI+PMBIAS*CPSI
      RABIAR=RABIAS*SPSIO-PABIAS*CPSIO
      PABIAR=RABIAS*CPSIO+PABIAS*SPSIO
      YC(ICOUNT,1)=(BETAK*(+.86667*T(10)-.86667*T(12))+RBIASR)/SCALE(1)
      YC(ICOUNT,2)=(BETAK*(-T(8)+0.5*T(10)+0.5*T(12))+PBIASR)/SCALE(2)
      YC(ICOUNT,3)=(T(4)+RABIAR)*RADIAN/SCALE(3)
      YC(ICOUNT,4)=(T(6)+PABIAR)*RADIAN/SCALE(4)
      GO TO 20
   15 YC1         =BETAK*(.86667*T(10)-.86667*T(12))
      YC2         =BETAK*(-T(8)+0.5000*T(10)+0.5000*T(12))
      YC3         =T(4)*RADIAN
      YC4         =T(6)*RADIAN
      YC(ICOUNT,1)=(YC1*CPSI-YC2*SPSI+RMBIAS)/SCALE(1)
      YC(ICOUNT,2)=(YC1*SPSI+YC2*CPSI+PMBIAS)/SCALE(2)
      YC(ICOUNT,3)=(YC3*SPSIO+YC4*CPSIO+RABIAS*RADIAN)/SCALE(3)
```

```
      YC(ICOUNT,4)=(-YC3*CPSIO+YC4*SPSIO+PABIAS*RADIAN)/SCALE(4)
   20 BET1(ICOUNT)=T(8)*RADIAN
      BET2(ICOUNT)=T(10)*RADIAN
      BET3(ICOUNT)=T(12)*RADIAN
      RDELT1(ICOUNT)=T(5)*RADIAN
      RDELT2(ICOUNT)=T(7)*RADIAN
      SPMRP(ICOUNT,1)=NU1
      SPMRP(ICOUNT,2)=NU2
      DO 25 N=1,NS
      J1=(N-1)*ND+ICOUNT
      ER(J1)=(YC(ICOUNT,N)-YM(ICOUNT,N))*WEIGHT(N)
   25 YW(J1)=YC(ICOUNT,N)*WEIGHT(N)
  400 GK=GK
      RETURN
      END
```

```
      SUBROUTINE PRINT1
      DIMENSION Z(12),ZZ(12)
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
C
      COMMON/PRIN/  PSI1(200),SPMRP(200,2),BET1(200),BET2(200),BET3(200)
     1              ,RDELT1(200),RDELT2(200)
C
      WRITE(6,601)
  601 FORMAT(1H1,12X,'COMPUTED OUTPUT VALUES',23X,'MEASURED VALUES',
     123X,'COMPUTED VALUES'/)
      WRITE(6,700)
  700 FORMAT(4X,'PSI   MBETAX   MBETAY   ROLL A  PITCH A MBETAX  MBETAY
     1      ROLL A PITCH A SPMR   SPMP  BETA1  BETA2  BETA3 RDELT1 RDELT
     22')
 1002 DO 2 J=1,ND
 1003 DO 3 I=1,NS
      Z(I) = YC(J,I)*SCALE(I)
    3 ZZ(I)=YM(J,I)*SCALE(I)
    2 WRITE(6,602) PSI1(J),(Z(I),I=1,NS),(ZZ(I),I=1,NS),SPMRP(J,1),
     1SPMRP(J,2),BET1(J),BET2(J),BET3(J),RDELT1(J),RDELT2(J)
  602 FORMAT(1X,F7.2,2F9.1,2F8.3,2F9.1,2F8.3,2F7.1,3F7.3,2F7.2)
      RETURN
      END
```

```
      SUBROUTINE IDENT1
      DOUBLE PRECISION TITEL,SUNIT,AUNIT,END,G,DDA,D,GPL,RSC,RES,RDW,GST
     1,GPR,SIM,AMP,RET
      DIMENSION NVAR(6)
C
      COMMON/COM1/  N3,N1,V,CORR,RELERR,ERM,SUNIT(12),TITEL(10),
     1              AUNIT(36)
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
      COMMON/PRM/   MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)
     1              ,DA(36),ER(2400),DER(2400,36)
      COMMON/WGHT/  WEIGHT(12)
      COMMON/IDENT/ KSWTCH,ITPL,IPLOT1,IPLOT2,ISKIP,ITMAX,THR,MCOR,ICOR
      COMMON/NVRT/  D(36),DDA(36),G(36,36)
      COMMON/PLOT/  SYMB(4)
      COMMON/PARSUB/INT(5),RE(4),PCT(36),HALT
C
      COMMON/MAIN1/ T(123),D1(230),OMEGA,D2(20),ICOUNT,P,ID2(2)
C
      DATA END,GPL,RSC,RES/'********','*GO PLOT','**RESCOR','**RESAMP'/,
     1RDW/'*READWGH'/,GST/'*GO STOR'/,GPR/'*GO PRIN'/,SIM/'*SIMULAT'/
     2,AMP,RET/'**OUTPUT','*RETURN '/,TEN/'1'/
      LOGICAL *1 HALT
C
C     STORE INITIAL CONDITIONS IN MATRIX XINIT(J)
      ISTAT=T(1)
 1004 DO 4 J=1,ISTAT
    4 XINIT(J) = T(J+3)
C
  100 FORMAT(10A8)
  143 FORMAT(///,2X,10A8)
      KRUN = 0
C
C     INITIAL SETTING OF %VARIATION FOR EACH OF THE 36 PARAMETERS
      DO 997 I=1,36
  997 PCT(I) = 0.01
C
C     INITIALIZING OF WEIGHTS FOR ALL 12 MEASUREMENTS
      DO 998 I=1,12
  998 WEIGHT(I) = 1.
C
C     SET MINIMJM ERROR AT WHICH ALGORITHM WILL STOP ITERATING
      ERM = 0.0001
C
C     SET MINIMUM VALUE AT WHICH PARAMETRS WILL BE CONSIDERED IRRELEVANT
```

```
      DMIN = 1.E-07
C
C     START CLOCK TO KEEP TRACK OF COMPUTATION TIME
 1000 CALL SETTIM(0,0)
      V=0.
      KRUN = KRUN + 1
C
C     CALL INOUT TO READ AND WRITE MODEL PARAMETERS AND MEASURED DATA
      CALL INOUT(1)
      IF(TITEL(1).EQ.RET)    RETURN
C
C     CALCULATE AUTOCORRELATION WHEN READPLOT OPTION IS USED
      IF(V.EQ.-1.)  CALL ERROR(0,N3,0,V,CORR,RELERR)
      IF(TITEL(10).EQ.SIM)   GO TO 212
C
C     IDENTIFY PARAMETERS
      M1 = 1
      CALL PARAM(N1,N3,M1,DMIN,ERM)
      IF(M1.EQ.0)            GO TO 212
C
C     CALCULATE RELATIVE ERROR, CORRELATION AND AUTOCORRELATION
      CALL ERROR(N1,N3,M1,V,CORR,RELERR)
C
C     CALL INOUT TO OUTPUT FINAL RESULTS OF IDENTIFICATION
      CALL INOUT(2)
C
C     READ THE OPTION CARDS
  212 READ(5,100) TITEL
  213 WRITE(6,143) TITEL
      IF(TITEL(1).EQ.GPR)    CALL PRINT1
  243 IF(TITEL(1).NE.GST)    GO TO 43
C
C     STORE DATA ON DISK
      READ(5,570) IDSET,ID1,IRUN
  570 FORMAT(4X,I2,2X,I2,I2)
      WRITE(6,701) IDSET
  701 FORMAT(/' THE DATA WILL BE STORED ON UNIT',I2)
      CALL CRUNCH(ID1,IRUN,10000,IDSET)
C
   43 IF(TITEL(1).EQ.GPL)      GO TO 10
      IF(TITEL(1).NE.END) GO TO 212
      GO TO 99
   10 READ(5,500)  TITEL(1),MULT,NVAR,ND1,ISK1
  500 FORMAT(A8,A1,I1,7I5)
      IF(MULT.EQ.TEN)      NVAR(1) = NVAR(1) + 10
```

```
      IF(NVAR(1) .EQ. 0) GO TO 213
      WRITE(6,143)  TITEL
      NDS = ND
C
C     PLOT ANYONE OF THE FOLLOWING OPTIONS: COMPUTED AND MEASURED VALUES
C     RESIDUAL AMPLITUDE OF THE OUTPUTS OR THE RESIDUAL AUTO-CORRELATION
C     FUNCTION OF THE OUTPUTS
 1001 DO 1 I=1,6,2
      IF(NVAR(I).EQ.0)        GO TO 1
      L = NVAR(I)
      K = NVAR(I + 1)
      IF(TITEL(1).NE.RES)     GO TO 2002
 1003 DO 3 J=1,ND
      YC(J,K) = YC(J,L) - YM(J,L)
    3 YM(J,K) = 0.
      GO TO 202
 2002 IF(TITEL(1).NE.RSC)     GO TO 201
 1002 DO 2 J=1,ND
      J1 = J + (L-1)*NDS
      YC(J,K) = YW(J1)
    2 YM(J,K) = 0.
      GO TO 202
  201 IF(TITEL(1).NE.AMP)     GO TO 213
  202 ISK = 1
      IF(ISK1.NE.0)       ISK = ISK1
      IF(ND1.NE.0)       ND = ND1
      CALL PLOTIN(L,K,ISK)
    1 CONTINUE
C
      ND = NDS
      GO TO 10
   99 TIME = INTVAL(0,0)/1000.
      WRITE(6,600) KRUN,TIME
  600 FORMAT(//' ***END OF IDENTIFICATION RUN NUMBER',I2,' , TOTAL COMPU
     1TATION TIME',F6.2,' SEC. ***')
      GO TO 1000
      END
```

```
      SUBROUTINE INOUT(JJ)
C
      DOUBLE PRECISION TITEL,SUNIT,AUNIT,END,G,DDA,D,CODE,RET,RPL,RELVAR
     1 ,RDW
      DIMENSION CODE(5),SKIP(5)
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
      COMMON/PRM/   MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36),
     1              DA(36),ER(2400),DER(2400,36)
      COMMON/COM1/  N3,N1,V,CORR,RELERR,ERM,SUNIT(12),TITEL(10),
     1              AUNIT(36)
      COMMON/IDENT/ KSWTCH,ITPL,IPLOT1,IPLOT2,ISKIP,ITMAX,THR,MCOR,ICOR
      COMMON/PARSUB/N1DYDA,N3DYDA,IDUM(3),DUM(3),DMIN,PCT(36),HALT
      COMMON/NVRT/  D(36),DDA(36),G(36,36)
      COMMON/WRITE/ WR(36),WA(36)
      COMMON/WGHT/  WEIGHT(12)
      COMMON/PLOT/  SYMB(4)
C
      COMMON/MAIN1/ T(123),D1(230),OMEGA,D2(12),GPSIO,D3(7),ICOUNT,P,
     1              IFLOQ,JCOUNT
      COMMON/OUTNEW/YMR(200,4),IROT,INEW,IGYRO
      COMMON/INOUT1/PHI,RPHI,THET,RTHET,IREC
C
      DATA END,RP,CTINUE,SLH/'********','}','C','/'/,ICODE/'DSET'/,
     1CODE/'*SIMULAT','*RERUN  ','*NEWDATA','*CONTINU','*SENSITI'/,
     1 RET,RPL/'*RETURN ','*READPLO'/,RELVAR/'*RELVAR '/,RDW/'*WEIGHTS'/
C
      LOGICAL *1 SKIP,HALT
      ISTAT=T(1)
C     JJ=2 AFTER IDENTIFICATION HAS BEEN COMPLETED
      IF(JJ.EQ.2)           GO TO 208
  143 FORMAT(31X,10A8)
  100 FORMAT(10A8)
  699 FORMAT(1X,10A8)
      DO 9001 I=1,5
 9001 SKIP(I) = .FALSE.
 9002 READ(5,100) TITEL
      WRITE(6,699) TITEL
      IF(TITEL(1).EQ.CODE(4) )   RETURN
      IF(TITEL(1).EQ.RET)     RETURN
      IF(TITEL(1).NE.RPL)     GO TO 9004
      SKIP(5) = .TRUE.
      SKIP(2) = .TRUE.
      SKIP(3) = .TRUE.
```

```
      SKIP(4) = .TRUE.
                                 GO TO 9002
 9004 DO 9003 I=1,5
      IF(TITEL(1).NE.CODE(I) )      GO TO 9003
      SKIP(I) = .TRUE.
                                      GO TO 9002
 9003 CONTINUE
      IF(SKIP(5))       SKIP(1) = .TRUE.
      IF(SKIP(2))                GO TO 1
      IF(SKIP(3))                GO TO 151
    1 READ(5,101) N1,ND,NS,KSK,NSTP,KSTOP,LIST,INIT,ISTART,
     1 SYMB,KSWTCH,ITPL,IPLOT1,IPLOT2,ISKIP,ITMAX,WGF,THR,MCOR,ICOR
  101 FORMAT(7I5,12I1,I3/4A1,1X,6I5,5X,2F10.5,10X,2I5)
C N1 IS THE NUMBER OF PARAMETERS TO BE ESTIMATED
      N2 = N1 + 1
      MN(N2) = 1
C ND IS THE NUMBER OF POINTS OF THE TIME HISTORY WHICH ARE USED
C NS IS THE NUMBER OF  VARIABLES WHICH WERE MEASURED
C KSK IS THE SKIPPING FACTOR
C MNS WILL DETERMINE WHICH PARAMETERS WILL BE ESTIMATED (NOT EST IF 0)
C MNS(N2) IS THE INDEX OF THE PARAMETER STARTING THE DEPENDENCE ANALYSIS (COR)
      ND0 = ND
    2 READ(5,102) MNS,MNS(N2)
  102 FORMAT(37I1,2X,I2)
      IF(SKIP(3))                GO TO 151
      IF(SKIP(2) )               GO TO 7
C READ THE VALUE OF THE PARAMETERS (A), AND THEIR LIMITS (AMIN ,AMAX)
 1003 DO 3 I1=1,31,6
      I2 = I1 + 5
      READ(5,503) (A(I),I=I1,I2),CND
      IF(CND.NE.CTINUE)          GO TO 30
    3 CONTINUE
   30 IF(I2.LT.N1)               GO TO 60
      READ(5,530) (AMIN(I),I=1,I2)
      READ(5,530) (AMAX(I),I=1,I2)
       DO 930 I=1,N1
      SENSIV(I) = 0.
      IF(AMIN(I).EQ.0.)          AMIN(I) = -1.E+50
  930 IF(AMAX(I).EQ.0.)          AMAX(I) = 1.E 50
      READ(5,531) (WA(I),I=1,N1)
  503 FORMAT(6E10.3,4X,A1)
  530 FORMAT(6E10.3)
  531 FORMAT(12(A4,1X) )
    5 READ(5,505) (SCALE(I),I=1,6),CND,(SUNIT(I),I=1,6)
  505 FORMAT(6E10.3,4X,A1/6A8)
```

```
      IF(CND.EQ.CTINUE)READ(5,505) (SCALE(I),I=7,12),J,(SUNIT(I),I=7,12)
      DO 6 NS2 = 1,11
      IF(SCALE(NS2+1).EQ.0.) GO TO 5508
    6 CONTINUE
 5508 READ(5,508) (AUNIT(I),I=1,N1)
  508 FORMAT(10A8)
C
C     READ THE DATA CARDS .(THE FIRST DATA CARD INDICATES ON WHICH UNIT THE DATA
C     IS TO BE READ. IT MUST CONTAINS, STARTING IN COLUMN 1 : DSETNN , WHERE NN
C     IS AN INTEGER DEFINING THE DATA SET.(05 IS DATA CARDS,07 MIGHT INDICATE A
C     DISK). IF THIS CARD DOES NOT APPEAR,THE DEFAULT IS 05 )
  151 MISFIT = MAX0(1,ISTART)
      IF(SKIP(3))       ND = ND0
      CALL READIN(KSTOP,LIST,MISFIT)
C
C    KDT IS THE ANGLE INCREMENT  IN MILLIDEGREES
      DPSI = KDT/1000.
      DT = DPSI/(57.29578*OMEGA)
C
C     FOR FORCED S/P ANGLES IFLOQ=1  STORE S/P ROLL ANGLE IN YMR(I,1)
C     AND S/P PITCH ANGLE IN YMR(I,2) IN ORDER NOT TO INTERFERE WITH
C     YM IN THE OUTPUT SUBROUTINE
      IF(IFLOQ .EQ. 1) GO TO 799
      GO TO 820
  799 IF(IREC .EQ. 1) GO TO 812
      KKK=0
      DO 810 I=1,ND
      PSI=OMEGA*DT*FLOAT(KKK)*KSK
      KKK=KKK+1
      CPSIO=COS(PSI+GPSIO)
      SPSIO=SIN(PSI+GPSIO)
      YMR(I,1)=YM(I,3)*SCALE(3)/57.29578
      YMR(I,2)=YM(I,4)*SCALE(4)/57.29578
      YMR(I,3)=OMEGA*(PHI*CPSIO+THET*SPSIO)
  810 YMR(I,4)=OMEGA*(-PHI*SPSIO+THET*CPSIO)
      GO TO 820
C
C     GYRO DERIVATIVE CALCULATION FOR TRANSIONT LOCKED GYRO CASE
  812 DEL1=OMEGA*DT*KSK
      DEL2=2.*DEL1
      DEL3=3.*DEL1
      DO 813 I=1,ND
      YMR(I,1)=YM(I,3)*SCALE(3)/57.29578
  813 YMR(I,2)=YM(I,4)*SCALE(4)/57.29578
      NDDD=ND-3
```

```
      DO 815 I=1,NDDD
      YMR(I,3)=-OMEGA*((YMR(I,1)-YMR(I+1,1))/DEL1+(YMR(I,1)-YMR(I+2,1))/
     1DEL2+(YMR(I,1)-YMR(I+3,1))/DEL3)/3.0
  815 YMR(I,4)=-OMEGA*((YMR(I,2)-YMR(I+1,2))/DEL1+(YMR(I,2)-YMR(I+2,2))/
     1DEL2+(YMR(I,2)-YMR(I+3,2))/DEL3)/3.0
      YMR(ND-2,3)=-OMEGA*((YMR(ND-2,1)-YMR(ND-1,1))/DEL1+(YMR(ND-2,1)
     1-YMR(ND,1))/DEL2)/2.0
      YMR(ND-2,4)=-OMEGA*((YMR(ND-2,2)-YMR(ND-1,2))/DEL1+(YMR(ND-2,2)
     1-YMR(ND,2))/DEL2)/2.0
      YMR(ND-1,3)=-OMEGA*(YMR(ND-1,1)-YMR(ND,1))/DEL1
      YMR(ND-1,4)=-OMEGA*(YMR(ND-1,2)-YMR(ND,2))/DEL1
      YMR(ND,3)=YMR(ND-1,3)
      YMR(ND,4)=YMR(ND-1,4)
  820 IF(IROT .EQ. 1) GO TO 822
      GO TO 830
C
C     CONVERT MEASUREMENTS TO STATIONARY COORDINATES IF IROT=1
  822 DPSI=KDT/1000.
      DT=DPSI/(57.29578*OMEGA)
      KKK=0
      DO 825 JROT=1,ND
      PSI=OMEGA*DT*FLOAT(KKK)*KSK
      KKK=KKK+1
      CPSIO=COS(PSI+GPSIO)
      SPSIO=SIN(PSI+GPSIO)
      CPSI=COS(PSI)
      SPSI=SIN(PSI)
      YM1=YM(JROT,1)*SCALE(1)
      YM2=YM(JROT,2)*SCALE(2)
      YM3=YM(JROT,3)*SCALE(3)
      YM4=YM(JROT,4)*SCALE(4)
      YM(JROT,1)=(YM1*CPSI-YM2*SPSI)/SCALE(1)
      YM(JROT,2)=(YM1*SPSI+YM2*CPSI)/SCALE(2)
      YM(JROT,3)=(YM3*SPSIO+YM4*CPSIO)/SCALE(3)
  825 YM(JROT,4)=(-YM3*CPSIO+YM4*SPSIO)/SCALE(4)
  830 IF(MISFIT.GT.0) STOP 2
      IF(SKIP(4))                GO TO 42
      IF(SKIP(2))                GO TO 7
      IF(SKIP(3))                GO TO 5100
C N3 IS THE TOTAL NUMBER OF DATA POINTS
    7 N3 = ND*NS
      NDMAX = 199
      ND = MIN0(ND,NDMAX)
      WRITE(6,107) TITEL,N1,NS,ND,N3,KSK,DT,NSTP
  107 FORMAT('1',30X,10A8 /' ',     30X,80('-')///' ',I4,' PARAMETERS',
```

```
      1    7X,I2,' OUTPUTS'//                    ' ',I4,' MEASURED VALUES PER
     1OUTPUT    (',I4,' MEASUREMENTS),      ',I4,' WAS THE SKIPPING FACTO
     1R USED        '//'       TIME INTERVAL BETWEEN MEASUREMENTS',F8.4,'
     1SEC.',6X,I6,' STEPS OF INTEGRATION IN EACH INTERVAL'/)
      WRITE(6,707) DPSI
  707 FORMAT('     ANGLE INCREMENT BETWEEN MEASUREMENTS',F10.2,' DEG'//)
      WRITE(6,607) (XINIT(I),I=1,ISTAT)
  607 FORMAT(' INITIAL CONDITIONS'/1X,1P12E11.3)
      NS2=NS2+1
      WRITE(6,608) (SCALE(I),SUNIT(I),SLH,I=1,NS2)
  608 FORMAT(/' SCALING FACTORS FOR DATA UNITS'/' 1DTU=',6(1PE11.3,A8,1X
     1,A1) )
   40 WRITE(6,640) (I,RP,WA(I),I=1,N1)
      WRITE(6,140) (A(I),I=1,N1)
 5100 READ(5,100) TITEL
      IF(TITEL(1).NE.RELVAR) GO TO 5050
      READ(5,551)  (PCT(I),I=1,N1)
  551 FORMAT(6E10.3)
      DO 5555 I=1,N1
 5555 IF(PCT(I).EQ.0.)        PCT(I) = 0.01
                              GO TO 5100
 5050 IF(TITEL(1).NE.RDW) GO TO 50
      READ(5,550) (WEIGHT(I),I=1,NS),ERM,DMIN
  550 FORMAT(8E10.3)
                              GO TO 5100
   50 IF(SKIP(5))             GO TO 6650
      IF(SKIP(1))             GO TO 42
      WRITE(6,240) (AMIN(I),I=1,N1)
      WRITE(6,340)(AMAX(I),I=1,N1)
      WRITE(6,807)ITMAX,THR
  640 FORMAT(/' PARAMETERS NAMES'/(1X,12(3X,I2,A1,1X,A4)/))
  140 FORMAT(/' INITIAL VALUES OF THE PARAMETERS'/(' ',1P12E11.3/) )
  240 FORMAT(' LOWER BOUNDS'/(1X,1P12E11.3/) )
  340 FORMAT(' UPPER BOUNDS'/(1X,1P12E11.3/) )
  807 FORMAT( /' THE IDENTIFICATION PROCEDURE WILL USE A MAXIMUM OF ',
     1I3,' ITERATIONS AND A SEPARATION THRESHOLD EQUAL TO',E9.2)
 6650 WRITE(6,650) (WEIGHT(I),I=1,NS)
  650 FORMAT(/' ***THE FOLLOWING WEIGHTS ARE USED***'/6X,6(1PE11.3,10X))
      WRITE(6,651) ERM,DMIN
  651 FORMAT(' ERRMIN = ',1PE10.3,'  ,     DMIN = ',E10.3)
      WRITE(6,660)  (PCT(I),I=1,N1)
  660 FORMAT(/' INITIAL PARAMETER INCREMENTS FOR GRADIENT EVALUATION'/
     1(1X,1P12E11.3))
   42 IF(TITEL(1).EQ.END)     GO TO 43
      READ(5,100) TITEL
```

```
      WRITE(6,143) TITEL
                             GO TO 42
   43 IF(SKIP(4))            GO TO 64
      IF(SKIP(5))            GO TO 162
      IF(SKIP(1))            GO TO 62
      RETURN
  208 CONTINUE
    8 WRITE(6,108)V,CORR,RELERR,
     1(I,WA(I),A(I),DA(I),AUNIT(I),SENSIV(I),MN(I),MNS(I),I=1,N1)
  108 FORMAT('1',10X,'FINAL RESULTS OF THIS IDENTIFICATION'/11X,36('=')/
     3' FIT TEST **       COST =',E12.4,10X,'CORRELATION =',F8.5/9X,
     3'RELATIVE ERROR =',F8.4//
     4' ESTIMATED VALUES OF THE PARAMETERS'/61X,'DEPENDENCY INDEX'/4X,
     A' NAME',5X,'EST.VALUE',3X,
     5'ERROR BOUNDS   UNITS',5X,'SENSITIVITY',3X,' FINAL  INITIAL'/
     6(I4,1X,A4,' =',1PE12.4,'  +/- ',E8.1,1X,A8,0PF14.5,6X,I1,7X,I1) )
      RETURN
   60 READ(5,100) TITEL
      WRITE(6,690) I2,N1,TITEL
  690 FORMAT(//' ****SUB.INOUT MESSAGE.... ONLY',I3,' VALUES WERE FOUND
     1 IN THE DATA CARDS FOR THE',I3,' PARAMETERS'//' THE NEXT CARD CONT
     2AINS...',10A8/' *RUN TERMINATED')
      STOP
   62 READ(5,562) I,IDSET,NTAPE,NRUN
  562 FORMAT(A4,I2,2X,I2,I2)
      IF(I.NE.ICODE)        GO TO 63
  162 KT = 1
      CALL DYN(N1,KT,0)
      J1 = 0
 1066 DO 66 I=1,NS
      DO 66 J=1,ND
      J1 = J1 + 1
      ER(J1) = 0.
   66 YM(J,I) = YC(J,I)
      IF(SKIP(5))            GO TO 69
      CALL CRUNCH(NTAPE,NRUN,10000,IDSET)
      WRITE(6,662) NTAPE,NRUN,IDSET
  662 FORMAT(/' SIMULATED RUN ',2I2,' COMPLETED AND STORED ON UNIT ',I2)
   65 TITEL(10) = CODE(1)
      RETURN
   64 N3 = 0
 1067 DO 67 I=1,NS
      SCALE(I) = 1.
      DO 67 J=1,ND
      N3 = N3 + 1
```

```
      ER(N3) = YM(J,I)
   67 YC(J,I) = YM(J,I)
      V=-1.
      GO TO 65
   63 WRITE(6,663) I
  663 FORMAT(' WRONG CODE ''',A4,''', DSET UNDEFINED, RUN NOT STORED')
      TITEL(1) = RET
         RETURN
   69 N1DYDA = N1
      N3DYDA = N3
      CALL DYDA
      ICOR = ITMAX + 1
      RETURN
      END
```

```
      SUBROUTINE DYN(N1,KT,KMIX)
      EXTERNAL DERIV
      REAL NU1,NU2
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
C
      COMMON/MAIN1/ T(123),D1(230),OMEGA,D2(17),NU1,NU2,TIMEF,ICOUNT,
     1              PSI,IFLOQ,JCOUNT
      COMMON/OUTNEW/YMR(200,4),IROT,INEW,IGYRO
C
      CALL NEWP
C
C     RESET THE INITIAL CONDITIONS
 1001 DO 1 J=1,ISTAT
    1 T(J+3) = XINIT(J)
      ICOUNT=1
      IF(IFLOQ) 4,5,8
    4 NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
      GO TO 5
C
    8 T(4)=YMR(ICOUNT,1)
      T(6)=YMR(ICOUNT,2)
      T(5)=YMR(ICOUNT,3)
      T(7)=YMR(ICOUNT,4)
      IF(INEW .EQ. -1) GO TO 5
      NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
C
C     KDT HAS BEEN SUBSTITUED FOR KOPSI
    5 DT=KDT/(57295.78*OMEGA)
      NSK = NSTP*KSK
      T(2) = (KT-1)*DT
      T(3) = DT/NSTP
      PSI=OMEGA*T(2)
   10 GK=GK
      CALL INTS(T,ISTAT,1,0.01,0.,0.,0,0,0.,DERIV)
      PSI=OMEGA*T(2)
 1002 DO 2 I=1,ND
      KT=KT+1
      ICOUNT=I
      IF(IFLOQ) 40,45,30
C
C     FORCED S/P ANGLES AND MOMENTS
```

```
   30 T(4)=YMR(ICOUNT,1)
      T(6)=YMR(ICOUNT,2)
      T(5)=YMR(ICOUNT,3)
      T(7)=YMR(ICOUNT,4)
      IF(INEW .EQ. -1) GO TO 45
      NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
      GO TO 45
C
C
C     FORCED S/P MOMENTS ONLY
   40 NU1=YM(ICOUNT,7)*SCALE(7)
      NU2=YM(ICOUNT,8)*SCALE(8)
C
   45 CALL OUTPUT
   20 GK=GK
 2002 DO 2 J=1,NSK
      CALL INTM
      PSI=OMEGA*T(2)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READIN(KSTOP,LIST,MISFIT)
      DIMENSION KREAD(14),IDCHAN(14)
C
      COMMON/STATE/ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),XINIT(24)STATE
     1,YC(200,12),YM(200,12),YW(2400)                                  STATE
C
      DATA DSET,RUN,DATA,CNTINU,END/'DSET','RUN ','DATA','****','END '/
      LOGICAL *1 START
      START = .FALSE.
      ISTART = MISFIT
      MISFIT = 0
      NCHAN = 0
      NMAX = 200
      LMX = NMAX
      LAST = NMAX
      NREF = 0
      READ(5,500,ERR=7) CODE,NVAR,NCARD,KREAD
  500 FORMAT(A4,I2,1X,I2,1X,14I5)
      IF(CODE.NE.DSET)        GO TO 7
      IDSET = NVAR
                              GO TO 10
  201 LMX = MINO(LMX,LAST)
      IDEND = 1
    1 IF(LIST.EQ. 0  )        GO TO 10
      WRITE(6,101) NVAR,NCARD,KREAD,LMX
  101 FORMAT(1X,I2,'*',I2,'*',14(I5,2X),3X,I4)
   10 READ(IDSET,110,ERR=60) CODE,NVAR,NCARD,KREAD
  110 FORMAT(A4,I2,1X,I2,1X,14I5)
      IF(CODE.EQ.END )        GO TO 50
      IF(CODE.EQ.RUN )        GO TO 30
      IF(.NOT.START)          GO TO 40
      IF(CODE.EQ.CNTINU)      GO TO 20
      IF(CODE.NE.DATA)        GO TO 60
      IF(NCHAN.GE.12)         GO TO 50
      NCHAN = NCHAN + 1
      IDCHAN(NCHAN) = NVAR
      LMX = MINO(LMX,LAST)
      IDEND = 0
      LAST = 0
      K = KSK
      NREF = 0
      IST1 = ISTART-((ISTART-1)/14)*14
      KINIT = INIT(NVAR)*KREAD(IST1)
   20 IF(IDEND.EQ.1)          GO TO 1
      NREF = NREF + 1
```

```
      IF(NCARD.NE.NREF)        GO TO 40
 1002 DO 2 J=1,14
      IF(J+(NREF-1)*14.LT.ISTART)               GO TO 2
      IF(KREAD(J).EQ.KSTOP.OR.LAST.GE.ND)      GO TO 201
      K = K + 1
      IF(K .LT. KSK)           GO TO 2
      LAST = LAST + 1
      YM(LAST,NCHAN) = KREAD(J) - KINIT
      K = 0
    2 CONTINUE
                               GO TO 1
   30 WRITE(6,130) LIST,NVAR,NCARD
  130 FORMAT(I1,' DATA CARDS LISTING*****RUN NUMBER',I3,'-',I2)
      KDT = KREAD(2)
      DT = KDT/1000.
      START = .TRUE.
                               GO TO 10
   40 MISFIT = 1
    5 WRITE(6,105) NREF,NVAR
  105 FORMAT(//' ***ERROR IN DATA CARDS ORDER, CARD NUMBER',I3,' MISSING
     1 IN VARIABLE',I2,' DATA SET'//' *END OF THE RUN ***')
   61 READ(5,161) CODE
  161 FORMAT(A4)
      IF(CODE.NE.END) GO TO 61
      RETURN
   60 MISFIT = 2
      READ(5,560) (YW(I),I=1,20)
  560 FORMAT(20A4)
      WRITE(6,660) CODE,(YW(I),I=1,20)
  660 FORMAT(' ERRONEOUS CODE !! ',A4,' !!'/' NEXT CARD CONTAINS ',20A4)
      RETURN
   50 LMX = MIN0(LMX,LAST)
      ND = MIN0(ND,LMX)
      KDTS = KDT*KSK
      WRITE(6,600) KDTS,KSK,ND,NCHAN,(IDCHAN(I),I=1,NCHAN)
  600 FORMAT(' X AXIS INCREMENT KDT=',I6,' SKIP=',I3,I8,' POINTS PER CHA
     1NNEL'/' THE',I3,' FOLLOWING CHANNELS WERE READ :',14(I2,','))
      RETURN
    7 IDSET = 5
      WRITE(6,607)
  607 FORMAT(' **SUB.READIN MESSAGE. UNDEFINED DATA SET, 5 IS ASSUMED')
      IF(CODE.NE.RUN)          GO TO 60
      GO TO 30
      END
```

```
      SUBROUTINE PARAM(N1S,N3S,M1,DMINS,ERRMIN)
      DOUBLE PRECISION D,DDA,G,G0
      DIMENSION OUT(12)
C
      COMMON/PARSUB/N1,N3,KC,KT,KMODE,V,V1,VS,DMIN,PCT(36),HALT
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)       PRM
     1,DA(36),ER(2400),DER(2400,36)                                      PRM
      COMMON/STATE/ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),XINIT(24)STATE
     1,YC(200,12),YM(200,12),YW(2400)                                    STATE
      COMMON/IDENT/KSWTCH,ITPL,IPLOT1,IPLOT2,ISKIP,ITMAX,THR,MCOR,ICOR
      COMMON/NVRT/D(36),DDA(36),G(36,36)
      COMMON/BASIS/IBASIS(36),PIV(36),G0(36,36)
      COMMON/WRITE/WR(36),WA(36)
C
      DATA BLK,DSC/'    ','*D '/    ,AIRR/'*I '/,UNUSED/'NU '/
      LOGICAL *1 HALT
      DMIN = DMINS
      N1 = N1S
 1100 DO 100 I=1,NS
      OUT(I) = 1.E-60
 2100 DO 100 J=1,ND
  100 OUT(I) = OUT(I) +  (YM(J,I) )**2
      N2 = N1 + 1
      MODE =  99
      KIT = 0
      KR = 0
      KMODE = 1
      KND = 2
C  DON'T ESTIMATE SOME GIVEN PARAMETERS ( FOR WHICH MNS = 0)
 1008 DO 8 I=1,N1
    8 MN(I) = MNS(I)
      VS = 1.E60
      KC = 0
    1 IF(KIT.GE.KSWTCH) KMODE = 0
      CALL SETTIM(1,0)
      KT = 1
      CALL DYN(N1,KT,KMODE)
  201 J = 0
      RELERR = 0.
      V = 0.
 1002 DO 2 I=1,NS
      V2 = 0.
 1020 DO 20 K=1,ND
      J = J + 1
      V2 = V2 + (YC(K,I)-YM(K,I))**2
```

```
  20 V = V + ER(J)**2
   2 RELERR = RELERR + V2/OUT(I)
     RELERR = SQRT(RELERR/NS)
     N33 = J
 260 N3 = J
     V1 = V
     V = V/J
     DV = V - VS
     IF(KIT.EQ.KSWTCH)         GO TO 5
     IF(DV.GT.1.E-02*VS)       GO TO 6
     KC = 0
                               GO TO 5
   6 CALL ADJUST
     IF(KC.GT.0)               GO TO 201
   5 IF(KIT.LT.ITPL)           CALL PLOTIN(IPLOT1,IPLOT2,ISKIP)
  14 WRITE(6,114) KIT,MODE,KND,RELERR,V,(A(I),I=1,N1)
 114 FORMAT(/' ITERATION',I3,'.     MODE WAS',I2,'.  KND=',I2,56X,
    1'REL.ERROR =',1PE9.2,'     COST =',E10.3/1X,38('='),'P  A  R  A  M
    2 E  T  E  R  S         V  A  L  U  E  S',43('=')/2X,' 1 13 25     2 1
    34 26     3 15 27     4 16 28     5 17 29     6 18 30     7 19 31     8 2
    40 32     9 21 33   10 22 34   11 23 35   12 24 36'/(1X,12E11.3/) )
     IF(KC.EQ.-1)              GO TO 214
     IF(KIT.GE.ITMAX)          GO TO 51
     IF(MODE.EQ.1)             GO TO 214
     IF(RELERR.LT.ERRMIN.OR.KND.EQ.3)        GO TO 51
 214 IF(VS.GT.V.OR.KIT.EQ.KSWTCH)      VS = V
     HALT = .TRUE.
     CALL DYDA
     IF(HALT)                  GO TO 50
     KR = KR + 1
     M=0
     IF(KC + ICOR - KIT.EQ.0)        ICOR = ICOR + 1
     IF(KIT+1.EQ.ICOR)         M=MCOR
     CALL COR(N1,N3,THR,M)
2011 M1 = 0
     KND = 3
1011 DO 11 I=1,N1
     WR(I) = DSC
     IF(G(I,I).EQ.0.)        WR(I) = AIRR
     IF(MNS(I).EQ.0)       WR(I) = UNUSED
     IF(MN(I).EQ.0)                GO TO 11
     WR(I) = BLK
     M1 = M1 + 1
     DA(I) = -DDA(I)
     DA(I) = AMAX1((AMIN(I)-A(I)),DA(I))
```

```
      DA(I) = AMIN1((AMAX(I)-A(I)),DA(I))
      IF (ABS(DA(I)).GT.0.0005*ABS(A(I)) ) KND = 0
      A(I) = A(I) + DA(I)
   11 CONTINUE
   17 WRITE(6,617) (I,WR(I),I=1,N1)
  617 FORMAT(' STATUS ',24(I2,A3))
      WRITE(6,618) (SENSIV(I),I=1,N1)
  618 FORMAT(' SENS =',24F5.2)
      TIME = INTVAL(1,0)/1000.
      I1 = IBASIS(1)
      I2 = IBASIS(M1)
      SEP = SQRT(PIV(I2))
      WRITE(6,600) MNS(N2),I1,WA(I1),I2,WA(I2),     SEP,TIME
  600 FORMAT(' START',I3,',  FIRST',I3,' (',A4,'),  LAST',I3,' (',A4,')'
     1,5X,'SEP =',E10.2,20X,'TIME =',F8.2,' SEC')
      KIT = KIT + 1
      MODE = KMODE
      GO TO 1
   50 WRITE(6,650)
  650 FORMAT(//' ALL THE PARAMETERS ARE IRRELEVANT, POSSIBLE MODEL ERROR
     1, IDENTIFICATION PROCESS CANCELLED')
      RETURN
   51 IF(KIT.GE.ICOR)       GO TO 152
 1052 DO 52 N=1,N1
   52 MN(N) = MNS(N)
      CALL COR(N1,N3,THR,MCOR)
  152 V1=0.
 1053 DO 53 J=1,N3
   53 V1 = V1 + YW(J)**2
      WRITE(6,601) (WA(I),PCT(I),I=1,N1)
  601 FORMAT(//' PARAMETER INCREMENTS (DA/A) FOR GRADIENT EVALUATION'//
     1(1X,6(2X,A4,'=',1PE9.1)))
 1054 DO 54 I=1,N1
      IF(MNS(I).EQ.0)         GO TO 54
      SENSIV(I) = ABS(A(I)*SNGL(D(I))/SQRT(V1) )
   54 CONTINUE
      N3S = N3
      M1 = 0
 1055 DO 55 I=1,N1
   55 IF(MN(I).NE.0)       M1 = M1 + 1
      RETURN
      END
```

```
      SUBROUTINE ADJUST
C
      COMMON/PARSUB/N1,N3,KC,KT,KMODE,V,V2,VS,DMIN,PCT(36),HALT
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)
     1,DA(36),ER(2400),DER(2400,36)
C
      LOGICAL *1 HALT
      IF(KC.EQ.0)             GO TO 7
      KC = KC + 2
      IF(KC.GT.3)             GO TO 5
      IF(KC.GT.1)             GO TO 1003
   16 WRITE(6,116) (DA(N),N=1,N1)
  116 FORMAT(' PARAMETERS INCREMENTS'/(1X,1P12E11.3 ))
 1003 DO 3 I=1,N1
      IF(MN(I).EQ.0)          GO TO 3
      A(I) = A(I) - 1.5*DA(I)
    3 CONTINUE
      KT = 1
      CALL DYN(N1,KT,KMODE)
      VB = 0.
 1030 DO 30 J=1,N3
   30 VB = VB + ER(J)**2
      VB = VB/N3
      EPS = 0.5
      IF(VB.LT.VS)            GO TO 1031
      V1 = 2.*V + 4.*VB - 6.*VS
      EPS = (V - 4.*VB + 3.*VS)/(2.*V1)
 1031 DO 31 I=1,N1
      IF(MN(I).EQ.0)          GO TO 31
      AMEMO = A(I)
      A(I) = A(I) - (EPS - 0.5)*DA(I)
      A(I) = AMAX1(A(I),AMIN(I))
      A(I) = AMIN1(A(I),AMAX(I))
      DA(I) = A(I) - AMEMO + 0.5*DA(I)
   31 CONTINUE
      EPS = -EPS
    4 WRITE(6,104) V,VB,EPS
  104 FORMAT(' ***STEP CORRECTION* COST WAS',1PE10.3,'.     BACK STEP
     1COST =',E10.3,'.  STEP IS MULTIPLIED BY ',E10.3)
      IF(VB.LT.VS)            RETURN
      KT = 1
      CALL DYN(N1,KT,KMODE)
      RETURN
    5 KC = 0
      RETURN
```

```
7 KC=-1
  RETURN
  END
```

```
      SUBROUTINE DYDA
      DIMENSION ERS(2400)
C
      COMMON/PARSUB/N1,N3,KC,KT,KMODE,V,V1,VS,DMIN,PCT(36),HALT
      COMMON/STATE/ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),XINIT(24)STATE
     1,YC(200,12),YM(200,12),YW(2400)                                 STATE
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)      PRM
     1,DA(36),ER(2400),DER(2400,36)                                     PRM
C
      LOGICAL *1 HALT,RERUN
      YWMAX = 0.
 1021 DO 21 J=1,N3
      YWMAX = AMAX1(YWMAX,ABS(YW(J)))
   21 ERS(J) = ER(J)
      RERUN = .FALSE.
 1006 DO 6 N = 1,N1
      IF(RERUN.AND.N.NE.IRERUN)   GO TO 6
C RESET MN
      MN(N) = MNS(N)
      IF(MN(N).EQ.0)            GO TO 6
      AMEMO = A(N)
      STEP =  PCT(N)*A(N)
      IF(A(N).EQ.0.0)           STEP = DA(N)*PCT(N)
C FIRST ORDER PROCEDURE TO OBTAIN THE DERIVATIVES
      A(N) = AMEMO + STEP
      STEP = A(N) - AMEMO
      KT = 1
      CALL DYN(N1,KT,KMODE)
      IRR = 0
      DERMAX = 0.
 1001 DO 1 J=1,N3
      DER(J,N) =0.
      DERJN = ER(J) - ERS(J)
      ABDER = ABS(DERJN)
      DERMAX = AMAX1(DERMAX,ABDER)
      IF(ABDER.LE.DMIN*ABS(YW(J)))      GO TO 1
      DER(J,N) = DERJN/STEP
      IRR = 2
    1 CONTINUE
      SENSIV(N) = DERMAX/YWMAX
      A(N) = AMEMO
      MN(N) = IRR
      IF(IRR.EQ.0)        PCT(N) = 3.*PCT(N)
      IF(MN(N).NE.0)            HALT=.FALSE.
    6 CONTINUE
```

```
      IF(HALT)     RETURN
      DO 23 J=1,N3
   23 ER(J) = ERS(J)
      RERUN = .TRUE.
 1022 DO 22 I=1,N1
      IF(MN(I).EQ.0)           GO TO 22
      PCT(I) = PCT(I)*(1. + 0.9/SENSIV(I))/10.
      PCT(I) = AMAX1(1.0E-06,PCT(I))
      IF(SENSIV(I).LE.1.)      GO TO 22
      IRERUN = I
      PCT1 = PCT(IRERUN)*100.
      WRITE(6,600)  IRERUN,PCT1
  600 FORMAT(' INCREMENT IN PARAMETER #',I2,' REDUCED TO',F9.4,' %')
                               GO TO 1006
   22 PCT(I) = AMIN1(PCT(I),0.5)
      RETURN
      END
```

```
      SUBROUTINE COR(N1,N3,THR,MODE1)
      DOUBLE PRECISION AD,DDA,G,C,GRAD,S,SAVE,SMAX,SMIN,DET
      DIMENSION SEPAR(18),SET(36,18),CLEAR(36),CLEARB(36),IR(36)
C
      COMMON/BASIS/IBASIS(36),PIV(36),G(36,36)
      COMMON/NVRT/AD(36),DDA(36),C(36,36)
      COMMON/WRITE/WR(36),WA(36)
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)
     1,DA(36),ER(2400),DER(2400,36)                                     PRM

      DATA PEQ,COMMA/3H! =,1H,/
      LOGICAL *1 CLEAR,CLEARB,SET,IR,INV
      IF(N1.EQ.0)              GO TO 6999
      CALL SETTIM(2,0)
      MODE = MODE1
      NEXT = 1
C *SECTION 1* ------------COMPUTATION OF THE GRAM MATRIX-----------------------
      SMAX = 0.D0
      DO 500 I=1,N1
      AD(I) = 0.D0
      IF(MN(I).EQ.0)           GO TO 2500
 1503 DO 503 K=1,N3
  503 AD(I) = AD(I) + DBLE(DER(K,I))**2
      IF(AD(I).LT.1.D-70)      MN(I) = 0
      AD(I) = DSQRT(AD(I))
 2500 DO 500 J=1,N1
  500 C(I,J) = 0.D0
 1501 DO 501 I=1,N1
      IF(MN(I).EQ.0)           GO TO 501
 1502 DO 502 J=1,I
      IF(I.EQ.J)               GO TO 502
      IF(MN(J).EQ.0)           GO TO 502
 1504 DO 504 K=1,N3
  504 C(I,J) = C(I,J) + DBLE(DER(K,I))*DBLE(DER(K,J))
      C(I,J) = C(I,J)/(AD(I)*AD(J))
      C(J,I) = C(I,J)
C   DEFINE THE FIRST BASIC PARAMETER
      IF(SMAX.GT.DABS(C(I,J)) ) GO TO 502
      SMAX = DABS(C(I,J))
      NEXT = I
  502 CONTINUE
      C(I,I) = 1.D0
      GRAD = 0.D0
 1510 DO 510 K=1,N3
  510 GRAD = GRAD + DBLE(DER(K,I))*DBLE(ER(K))
```

```
      DDA(I) = GRAD/AD(I)
  501 CONTINUE
      N2 = N1 + 1
      IF(MNS(N2).NE.0)        NEXT = MNS(N2)
C *SECTION 2*----------------INITIALISATION--------------------------------------
      IF(MODE.EQ.-1)          RETURN
      INV = MODE1.GE.10
      IF(INV)     MODE = MODE1-10
      MN(N2) = 1
      IF(MODE.EQ.0)           GO TO 507
      IF(N1.GT.15)            GO TO 507
  505 WRITE(6,6505)
 6505 FORMAT(  1H1,10X,'PARAMETERS CORRELATION MATRIX')
 1506 DO 506 I=1,N1
  506 WRITE(6,6506) WA(I),(C(I,J),J=1,I)
 6506 FORMAT(/1X,A4,2X,15F8.4)
  507 IF(MODE.LT.4)           GO TO 1508
      WRITE(6,601) WA(NEXT)
  601 FORMAT('1   BASIC   ',30X,' CRITICAL'/'  PARAMETER',5X,'SEPARATION'
     1,15X,'PARAMETER',5X,'SEPARATION'/4X,A4/)
C-----INITIALISE
      M1 = 0
 1508 DO 508 I=1,N1
      M1 = M1 +MN(I)
 1509 DO 509 J=1,N1
      IF(J.LE.18)         SET(I,J) = .FALSE.
      G(I,J) = C(I,J)
  509 C(I,J) = 0.D0
      C(I,I) = 1.D0
      IR(I) = MN(I).EQ.0
      MN(I) = ((MN(I)+1)/2)*2
      CLEAR(I) = .FALSE.
      IF(.NOT.IR(I))          GO TO 508
      C(I,I) = 0.D0
      DDA(I) = 0.D0
      CLEAR(I) = .TRUE.
  508 MN(N2) = MIN0(MN(N2),MN(I))
      IF(M1.EQ.0)             GO TO 6999
      DET = 1.D0
      S = THR**2
      KG = 0
      NSET = 0
      PIV(NEXT) = 1.
C *SECTION 3*---------ANALYSIS OF THE VECTOR SET--------------------------------
C THE NEW BASIS VECTOR IS NEXT
```

```
    1 CLEAR(NEXT) = .TRUE.
      IF(IR(NEXT))              GO TO 2004
      KG = KG + 1
      IBASIS(KG) =  NEXT
      IF(KG.GE.N1)              GO TO 20
C COMPUTE THE NEW ARRAY OF REMNANT VECTORS
 1002 DO 2 J=1,N1
      IF(CLEAR(J))              GO TO 2
C       TRANSFORM INPUT VECTOR DDA
      DDA(J) = DDA(J) - DDA(NEXT)*G(J,NEXT)
      IF(.NOT.INV)              GO TO 1003
 1203 DO 203 I=1,N1
  203 C(J,I) = C(J,I) - C(NEXT,I)*G(J,NEXT)
 1003 DO 3 I=1,J
      IF(CLEAR(I))              GO TO 3
      G(J,I) = G(J,I) - G(J,NEXT)*G(NEXT,I)
      G(I,J) = G(J,I)
    3 CONTINUE
    2 CONTINUE
C FIND THE DEPENDENT, THE OPTIMAL AND THE CRITICAL VECTORS
 2004 IREMN = 0
      SMAX = 0.D0
      SMIN = 1.D0
 1004 DO 4 I=1,N1
      IF(CLEAR(I))              GO TO 4
      IREMN = IREMN + 1
      SAVE = G(I,I)
      IF(SAVE.GT.SMIN)          GO TO 5
C STORE THE CRITICAL
      SMIN = SAVE
      IOUT = I
    5 IF(SAVE.LE.SMAX)          GO TO 4
C STORE THE OPTIMAL
      SMAX = SAVE
      NEXT = I
    4 CONTINUE
      IF(IREMN.EQ.0)            GO TO 20
C CHECK THE DEPENDENT
      IF(SMIN.LE.S)             GO TO 8
C COMPUTE THE VALUE OF THE DETERMINANT
      PIV(NEXT) = SMAX
      DET = DET*SMAX
C NORMALISE ROW AND COLUMN NEXT BEFORE THE NEW CYCLE
      SAVE = DSQRT(SMAX)
      IF(MODE.LT.4)             GO TO 1007
```

```
      SMIN = DSQRT(SMIN)
      WRITE(6,600) WA(NEXT),SAVE,WA(IOUT),SMIN
  600 FORMAT(4X,A4, 8X,1PE9.2,18X,A4, 8X,E9.2)
 1007 DO 7 I=1,N1
      IF(INV)      C(NEXT,I) = C(NEXT,I)/SMAX
      IF(CLEAR(I))             GO TO 7
      G(NEXT,I) = G(NEXT,I)/SMAX
    7 CONTINUE
      DDA(NEXT) = DDA(NEXT)/SMAX
                               GO TO 1
C *SECTION 4*-----------ANALYSIS OF THE DEPENDENCE WITHIN THE SUB-BASIS----------
    8 NSET = NSET + 1
      DDA(IOUT) =0.D0
      MN(IOUT) = 0
      CLEAR(IOUT) = .TRUE.
      IF(MODE.EQ.0)            GO TO 2004
      SET(IOUT,NSET) = .TRUE.
C FIND THE COMPONENTS OF IOUT ON THE BASIS
 1009 DO 9 I=1,KG
      KBACK = KG+1-I
      K = IBASIS(KBACK)
      IF(I.EQ.1)               GO TO 9
 1010 DO 10 J=2,I
      KBACK1 = KG-I+J
      K1 = IBASIS(KBACK1)
   10 G(K,IOUT) = G(K,IOUT) - G(K,K1)*G(K1,IOUT)
    9 CONTINUE
 1011 DO 11 I=1,KG
      K = IBASIS(I)
      GRAD = SMIN + (G(K,IOUT)**2)*PIV(K)
      IF(DABS(GRAD).LE.S)      GO TO 11
C MEMORISE THE DEPENDENT VECTOR IN THE LOGICAL ARRAY ''SET''
      SET(K,NSET) = .TRUE.
C SET THE MAGIC NUMBER TO 1 FOR DEPENDENCE OF THE NEXT BASIS VECTOR WITH IOUT
      MN(K) = 1
   11 CONTINUE
C MEMORISE THE SEPARATION OF THE DEPENDENT VECTOR IN THE ARRAY ''SEPAR''
      SEPAR(NSET) = DSQRT(DMAX1(0.D0,SMIN) )
                               GO TO 2004
C *SECTION 5* -----SOLVE THE EQUATION G*DDA = D'*ER   ( INVERT G EVENTUALLY)----
   20 IF(KG.EQ.1)              GO TO 1121
 1120 DO 120 I=2,KG
      KBACK = KG+1-I
      K = IBASIS(KBACK)
 2120 DO 120 J=2,I
```

```
      KBACK1 = KG-I+J
      K1 = IBASIS(KBACK1)
      IF(.NOT.INV)           GO TO 120
 1220 DO 220 L1 = 1,KG
      L = IBASIS(L1)
  220 C(K,L) = C(K,L) - G(K,K1)*C(K1,L)
  120 DDA(K) = DDA(K) - G(K,K1)*DDA(K1)
 1121 DO 121 I=1,KG
      K = IBASIS(I)
  121 DDA(K) = DDA(K)/AD(K)
C *SECTION 6* -----------------FIND THE STRUCTURE OF THE SET-------------------
      IF(MODE.EQ.0)          GO TO 9999
      IF(NSET.EQ.0)          GO TO 24
      IF(MODE.LT.5)          GO TO 24
C WRITE THE DEPENDENT SETS
      WRITE(6,621)
  621 FORMAT(/11X,'DEPENDENT SETS OF PARAMETERS'/ 1X,'SEPARATION'/)
 1022 DO 22 J=1,NSET
 1023 DO 23 I=1,N1
      IF(MN(I).EQ.0.AND.SET(I,J) )      IOUT = I
   23 CLEAR(I) = SET(I,J)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,622) SEPAR(J), (WR(I),COMMA,I=1,NW)
  622 FORMAT(   1X,E9.2/(10X,24(A4,A1 )/))
      IF(MODE.LT.6)          GO TO 22
      K = 0
 1012 DO 12 I=1,N1
      IF(MN(I).EQ.0)         GO TO 12
      IF(.NOT.SET(I,J) )  GO TO 12
      K = K + 1
      G(K,K)     = -G(I,IOUT)*AD(IOUT)/AD(I)
      WR(K) = WA(I)
   12 CONTINUE
   14 WRITE(6,614) WA(IOUT),PEQ,(WR(I),PEQ,G(I,I),I=1,K)
  614 FORMAT(/' IF ',A4,A3,' 1, THEN',6(3X,A4,A3,1PE9.2)/
     1(19X,6(3X,A4,A3,E9.2)/))
   22 CONTINUE
   24 WRITE(6,624) N1,THR,(MN(K),K=1,N1)
  624 FORMAT('1 ***THERE ARE',I3,' PARAMETERS IN THIS PROBLEM .
     1THE SEPARATION THRESHOLD WAS',E9.2  /' MAGIC NUMBER ',36I2/)
      IF(NSET.EQ.0)          GO TO 32
      IF(MODE.LT.3)          GO TO 32
C********************** UNION OF THE DEPENDENT SETS ****************************
 1125 DO 125 I=1,N1
  125 CLEAR(I) = .FALSE.
```

```
      NSETC = 0
 1025 DO 25 J=1,NSET
      IF(CLEAR(J))            GO TO 25
      NSETC = NSETC + 1
 1026 DO 26 K=J,NSET
      IF(CLEAR(K))            GO TO 26
 1027 DO 27 I=1,N1
      IF(SET(I,J).AND.SET(I,K))   GO TO 1028
   27 CONTINUE
                              GO TO 26
 1028 DO 28 I=1,N1
   28 SET(I,NSETC) = SET(I,J).OR.SET(I,K)
      SEPAR(NSETC) = AMIN1(SEPAR(J),SEPAR(K))
      CLEAR(K) = .TRUE.
   26 CONTINUE
   25 CONTINUE
C *SECTION 7* ---------PREPARE AND WRITE THE FINAL DIAGNOSTIC-------------------
 1029 DO 29 J=1,NSETC
 1030 DO 30 I=1,N1
      CLEAR(I) = (MN(I).EQ.1).AND.SET(I,J)
   30 CLEARB(I) = (MN(I).EQ.0).AND.SET(I,J)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,631) J,SEPAR(J),(WR(I),COMMA,I=1,NW)
  631 FORMAT(//' ***DEPENDENT SET NUMBER',I3,'    ***SEPARATION =',E9.2/
     1' *'/' * A TRUE VALUE IS OBTAINED FOR  '/(' *',8X,24(A4,A1 )/) )
      CALL TRADUC(CLEARB,N1,NW)
      WRITE(6,629) (WR(I),COMMA,I=1,NW)
  629 FORMAT(' * IF IS KNOWN THE TRUE VALUE OF '/(' *',8X,24(A4,A1 )/) )
   29 CONTINUE
   32 IF(MODE.LT.2) RETURN
CWRITE THE NAME OF INDEPENDENT, IRRELEVANT, DROPPED AND UNUSED PARAMETERS
 1033 DO 33 I=1,N1
      CLEAR(I) = MN(I).EQ.2
   33 CLEARB(I) = IR(I).AND.(MNS(I).NE.0)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,634) (WR(I),COMMA,I=1,NW)
  634 FORMAT(//' *  INDEPENDENT PARAMETERS  '/(' *',2X,24(A4,A1 )/))
      CALL TRADUC(CLEARB,N1,NW)
      WRITE(6,635) (WR(I),COMMA,I=1,NW)
  635 FORMAT(' *'/' *  IRRELEVANT PARAMETERS  '/(' *',2X,24(A4,A1 )/))
 1036 DO 36 I=1,N1
      CLEAR(I) = MN(I).EQ.0.AND.MNS(I).NE.0
   36 CLEARB(I) = MNS(I).EQ.0
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,637) (WR(I),COMMA,I=1,NW)
```

```
 637 FORMAT(' *'/' *  NOT ESTIMATED'/(' *',2X,24(A4,A1 )/))
     CALL TRADUC(CLEARB,N1,NW)
     WRITE(6,638) (WR(I),COMMA,I=1,NW)
 638 FORMAT(' *'/' *  NOT USED'/(' *',2X,24(A4,A1 )/))
     TIME = INTVAL(2,0)/1000.
     WRITE(6,606) DET,TIME
 606 FORMAT(///' DET=',D22.15,'    TIME =',F9.3,' SEC')
9999 RETURN
6999 WRITE(6,999)
 999 FORMAT(///10X,' ***PROBLEM WITHOUT PARAMETER, COR BY-PASSED')
     RETURN
     END
```

```
      SUBROUTINE TRADUC(FOUND,N1,K)
      DIMENSION FOUND(N1)
C
      COMMON/WRITE/WR(36),WA(36)
C
      DATA BLANK,EMPTY/'    ','NONE'/
      LOGICAL *1 FOUND
      K=0
 1001 DO 1 I=1,N1
      WR(I) = BLANK
      IF(.NOT.FOUND(I) )     GO TO 1
      K = K + 1
      WR(K) = WA(I)
    1 CONTINUE
      IF(K.GT.0)       RETURN
      K=1
      WR(1) = EMPTY
      RETURN
      END
```

```
      SUBROUTINE ERROR(N1,N3,M1,V,CORR,RELERR)
      DOUBLE PRECISION PROD,DDA,D
      DIMENSION MNHOLD(36)
C
      COMMON/STATE/ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),XINIT(24)STATE
     1,YC(200,12),YM(200,12),YW(2400)                                   STATE
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)      PRM
     1,DA(36),ER(2400),DER(2400,36)                                     PRM
      COMMON/IDENT/KSWTCH,ITPL,IPLOT1,IPLOT2,ISKIP,ITMAX,THR,MCOR,ICOR
      COMMON/NVRT/D(36),DDA(36),PROD(36,36)
C
      IF(N1.EQ.0)             GO TO 2000
 1001 DO 1 I=1,N1
      MNHOLD(I) = MN(I)
    1 MN(I) = MNS(I)
      CALL COR(N1,N3,THR,10)
      DO 2 I = 1,N1
    2 MN(I) = MNHOLD(I)
      J = 0
      ERMAX = 0.
      V1 = 0.
      RELERR = 0.
      DO 8 I = 1,NS
      V = 0.
      OUT = 0.
      DO 9 K = 1,ND
      OUT = OUT +  (YM(K,I))**2
      J = J + 1
      ERMAX = AMAX1(ERMAX,ER(J)**2)
    9 V = V + (YC(K,I)-YM(K,I))**2
      RELERR = RELERR + V/OUT
      V1 = V1 + V
    8 CONTINUE
      RELERR = RELERR/NS
      V = V1/J
      CORR = 1. - RELERR
      RELERR = SQRT(RELERR)
C AUTOCORRELATION OF THE RESIDUALS
 2000 V2 = 0.
 1022 DO 22 I=1,NS
      L = (I-1)*ND + 1
 1020 DO 20 K=1,ND
      V1= 0.
      L1 = I*ND - K + 1
 1021 DO 21 J=L,L1
```

```
      JPKM1 = J+K-1
   21 V1 = V1 + ER(J)*ER(JPKM1)
      KPLM1 = K+L-1
   20 YW(KPLM1) = V1/(L1-L+1)
   22 V2 = V2 + YW(L)
      V2 = V2/NS
      IF(N1.EQ.0)        RETURN
C ESTIMATION OF PARAMETERS VARIANCE
 1015 DO 15 N = 1,N1
      DA(N) = 0.
      DDA(N) = 0.
      IF(MN(N).EQ.0) GO TO 15
      P = PROD(N,N)*ERMAX
      P1 = PROD(N,N)*V2
      DA(N) = SQRT(P)/D(N)
      DDA(N) = SQRT(P1)/D(N)
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CRUNCH(NTAPE,NRUN,KSTOP,IDSET)
      DIMENSION  IWRCRD(14),ISTOR(12),YCMAX(12),KOVER(12),YCS(200,12)
C
      COMMON/STATE/ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),XINIT(24)STATE
     1,YC(200,12),YM(200,12),YW(2400)                                   STATE
C
      DATA  RUN,DATA,CNTINU,END,STOR/'RUN ','DATA','****','END ','*STO'/
C
      KDTS=KDT*KSK
      WRITE(6,101) NTAPE,NRUN,KDTS
  101 FORMAT(1H1,' NTAPE',I2,' NRUN',I2,6X,'KDT=',I5//)
      READ(5,502) CODE,ISTOR
  502 FORMAT(A4,6X,12I1)
      WRITE(6,110) ISTOR
  110 FORMAT(1X,' VARIABLES STORED ARE IDENTIFIED BY A 1 IN COLUMNS 11 T
     1O 22 AFTER THE COMMAND *STORAGE'//1X,'*STORAGE  '12I1)
      IF(CODE.NE.STOR)         GO TO 60
C WRITE THE FIRST CARD ON UNIT IDSET
   71 WRITE(IDSET,701) NTAPE,NRUN,KDTS
  701 FORMAT('RUN ',I2,'S',I2,6X,I5,60X)
      KVV = 1
 1001 DO 1 I=1,12
      KOVER(I) = 0
      IF(ISTOR(I).EQ.0)       GO TO 1
      YCMAX(I) = 0.
 1002 DO 2 J=1,ND
    2 YCMAX(I) = AMAX1(YCMAX(I),ABS(YC(J,I)))
      IF(YCMAX(I).GT.9999.OR.YCMAX(I).LT.100.)     KOVER(I) = 1
      KVV = KVV*(1 - KOVER(I))
    1 CONTINUE
      IF(KVV.EQ.1)             GO TO 30
      WRITE(6,603)
  603 FORMAT('1  ****OUTPUT DATA HAD TO BE RESCALED BEFORE STORING'/
     1' TO OBTAIN ENGR.    VALUES MULTIPLY STORED VALUES BY  : ' //  )
 1004 DO 4 I = 1,12
      IF(ISTOR(I)*KOVER(I)*YCMAX(I) .EQ. 0)      GO TO 4
      YCMAX(I) = YCMAX(I)/8000.
 1003 DO 3 J = 1,ND
      YCS(J,I)=YC(J,I)
    3 YC(J,I) = YC(J,I)/YCMAX(I)
      YCMAX(I)=YCMAX(I)*SCALE(I)
      WRITE(6,604)   YCMAX(I),I
  604 FORMAT(1X,1PE10.3,' FOR CHANNEL',I3)
    4 CONTINUE
   30 J1 = ND
```

```
 1005 DO 5 I=1,12
      YM(J1,I) = KSTOP
    5 YC(J1,I) = KSTOP
C------------WRITE THE DATA CARDS----------
 1006 DO 6 I=1,12
      IF(ISTOR(I).EQ.0)       GO TO 6
      J=0
      NCARD=0
      CODE = DATA
    7 NCARD = NCARD + 1
 1017 DO 17 K=1,14
      IWRCRD(K) = 0
      KPJ = K+J
      IF(KPJ.GT.J1)    GO TO 17
      IWRCRD(K) = YC(KPJ,I)
   17 CONTINUE
C WRITE THE DATA CARDS ON UNIT IDSET
      WRITE(IDSET,708) CODE,I,NCARD,IWRCRD
  708 FORMAT(A4,I2,1X,I2,1X,14I5)
      J = J + 14
      IF(J1.LT.J) GO TO 16
      CODE = CNTINU
      GO TO 7
   16 WRITE(6,616) I
  616 FORMAT('    VARIABLE',I2,' IS STORED')
    6 CONTINUE
    9 WRITE(6,109) END
  109 FORMAT(1X,A4)
C WRITE THE LAST CARD ON UNIT IDSET
      WRITE(IDSET,709) END
  709 FORMAT(A4,76X)
C
C     SET CALCULATED VALUES WHICH HAD TO BE RESCALED BACK TO THEIR
C     ORIGINAL VALUES FOR PLOT AND PRINT PROGRAMS
      DO 20 I=1,12
      IF(ISTOR(I)*KOVER(I)*YCMAX(I) .EQ. 0) GO TO 20
      DO 18 J=1,ND
   18 YC(J,I)=YCS(J,I)
   20 CONTINUE
      RETURN
   60 WRITE(6,660)
  660 FORMAT(/' ***NO STORAGE PARAMETERS, ALL VARIABLES STORED')
      DO 61 I=1,12
   61 ISTOR(I) = 1
      GO TO 71
```

```
      SUBROUTINE PLOTIN(I1,I2,KSKIP)
      DIMENSION PLOT(101),XM(4),KP(4)
C
      COMMON/STATE/ ND,NS,KSK,KDT,NSTP,ISTAT,INIT(12),SCALE(12),
     1              XINIT(24),YC(200,12),YM(200,12),YW(2400)
      COMMON/PLOT/  SYMB(4)
C
      DATA DOT/'I'/,BLANK/' '/
C
      DT = KDT*KSK/1000.
 1001 DO 1 I = 1,101
    1 PLOT(I) = BLANK
      PLOT(26) = DOT
      PLOT(76) = DOT
 1002 DO 2 J=1,4
      KP(J) = 0
    2 XM(J) = 1.E-50
      IF(I1.EQ.0) GO TO 2005
 1003 DO 3 I=1,ND,KSKIP
      XM(1) =AMAX1(XM(1),ABS(YM(I,I1) ) )
      AEST = YC(I,I1)
    3 XM(2) =AMAX1(XM(2),ABS(AEST))
 2005 IF(I2.EQ.0) GO TO 5
 1004 DO 4 I=1,ND,KSKIP
      XM(3) =AMAX1(XM(3),ABS(YM(I,I2) ) )
      AEST2 = YC(I,I2)
    4 XM(4) =AMAX1(XM(4),ABS(AEST2))
    5 WRITE(6,105) I1,SYMB(1),SYMB(2),I2,SYMB(3),SYMB(4)
  105 FORMAT(1H1,30X,'MEASURED AND COMPUTED TIME HISTORIES'//' SYMBOLS
     1 /VARIABLE',I2,4X,A1,' MEASURED, ',A1,' COMPUTED,       /VARIABLE',
     2I2,4X,A1,' MEASURED, ',A1,' COMPUTED/',16X,'T I M E'/' ',101('.'))
      XM1 =AMAX1(XM(1),XM(2) )
      XM2 =AMAX1(XM(3),XM(4) )
 1007 DO 7 I = 1,ND,KSKIP
      T = (I-1)*DT
      IF(I1.EQ.0) GO TO 6
      KPLOT = (25.*YM(I,I1))/XM1 + 26
      KP(1) = KPLOT
      PLOT(KPLOT) = SYMB(1)
      KPLOT = (25.*YC(I,I1)  )/XM1 + 26
      KP(2) = KPLOT
      PLOT(KPLOT) = SYMB(2)
    6 IF(I2.EQ.0) GO TO 8
      KPLOT = (25.*YM(I,I2))/XM2 + 76
      KP(3) = KPLOT
```

```
      PLOT(KPLOT) = SYMB(3)
      KPLOT = (25.*YC(I,I2)   )/XM2 + 76
      KP(4) = KPLOT
      PLOT(KPLOT) = SYMB(4)
    8 WRITE(6,108) PLOT,T
  108 FORMAT(' ',101A1,4X,F10.3)
 1010 DO 10 J = 1,4
      KPBLK = KP(J)
   10 PLOT(KPBLK) = BLANK
      PLOT(76) = DOT
      PLOT(26) = DOT
    7 CONTINUE
   11 WRITE(6,111) I1,XM(1),XM(2),I2,XM(3),XM(4)
  111 FORMAT(' MAXIMA :    VARIABLE',I2,'  MEAS=',1XE10.3,', COMP=',E10.3
     1,7X,' VARIABLE',I2,'  MEAS=',E10.3,', COMP=',E10.3)
      RETURN
      END
```

REFERENCES

1. Aubrun, Jean-Noel: Nonlinear Systems Identification in Presence of Nonuniqueness. NASA TN D6467, 1971.

2. Kanning, Gerd; and Biggers, James C.: Development and Verification of a Mathematical Model for a Hingeless Helicopter Rotor. NASA TN D-7834, 1974.

3. Ames Computer Science Library, Moffett Field, California, 94035.