https://ntrs.nasa.gov/search.jsp?R=19750024716 2020-03-22T18:52:21+00:00Z

NASA TECHNICAL MEMORANDUM

NASA TM X-3286

NASA TM X-3286

# TRANSFER-FUNCTION-PARAMETER ESTIMATION FROM FREQUENCY RESPONSE DATA - A FORTRAN PROGRAM

*Robert C. Seidel*

*Lewis Research Center*
*Cleveland, Ohio 44135*

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1975

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA TM X-3286 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| TRANSFER-FUNCTION-PARAMETER ESTIMATION FROM FREQUENCY RESPONSE DATA - A FORTRAN PROGRAM | September 1975 |
| | 6. Performing Organization Code |
| 7. Author(s) | 8. Performing Organization Report No. |
| Robert C. Seidel | E-8165 |
| | 10. Work Unit No. |
| 9. Performing Organization Name and Address | 505-05 |
| Lewis Research Center<br>National Aeronautics and Space Administration<br>Cleveland, Ohio 44135 | 11. Contract or Grant No. |
| | 13. Type of Report and Period Covered |
| 12. Sponsoring Agency Name and Address | Technical Memorandum |
| National Aeronautics and Space Administration<br>Washington, D.C. 20546 | 14. Sponsoring Agency Code |

15. Supplementary Notes

16. Abstract

A FORTRAN computer program designed to fit a linear transfer function model to given frequency response magnitude and phase data is presented. A conjugate gradient search is used that minimizes the integral of the absolute value of the error squared between the model and the data. The search is constrained to insure model stability. A scaling of the model parameters by their own magnitude aids search convergence. Efficient computer algorithms result in a small and fast program suitable for a minicomputer. A sample problem with different model structures and parameter estimates is reported.

| 17. Key Words (Suggested by Author(s)) | 18. Distribution Statement |
|---|---|
| Parameter identification<br>Gradient search<br>Frequency domain | Unclassified - unlimited<br>STAR Category 61 |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price* |
|---|---|---|---|
| Unclassified | Unclassified | 32 | $3.75 |

* For sale by the National Technical Information Service, Springfield, Virginia 22161

# TRANSFER-FUNCTION-PARAMETER ESTIMATION FROM FREQUENCY RESPONSE DATA - A FORTRAN PROGRAM

by Robert C. Seidel

Lewis Research Center

## SUMMARY

A FORTRAN computer program designed to fit a linear transfer function model to given frequency response magnitude and phase data is presented. A conjugate gradient search is used that minimizes the integral of the absolute value of the error squared between the model and the data. The search is constrained to insure model stability. Scaling of the model parameters by their own magnitude aids search convergence. Efficient computer algorithms are used for the calculation of the cost function and gradient and execution of the search. This results in rapid calculations and a small program size suitable for running on a minicomputer. Search convergence times for different model structures and parameters estimates for a sample problem derived from experimental data are presented.

## INTRODUCTION

Frequency response techniques have long been a fundamental tool of system design and testing. One problem often faced is that of fitting a transfer function to given amplitude and phase data. One of the most commonly used methods for doing this is the Bode plot graphical technique (ref. 1). Analytical techniques (refs. 2 to 4) that fit rational polynomial transfer functions using the digital computer have also been used. In reference 5 the mathematical framework is presented for a gradient search solution which, not being limited to rational polynomials, can handle dead time parameters, but there are no computer programs or practical results presented.

Reported herein is a FORTRAN computer program designed to fit a transfer function to frequency response data. The program has been used to obtain the transfer function models reported in references 6 and 7 for supersonic mixed-compression inlets. Search convergence times for different model structures and parameter estimates for a

sample problem derived from reference 6 data are presented. A scaling rule is incorporated to improve search convergence, and a constraint is placed in the search to insure model stability during the search. Efficient algorithms used for several different calculations result in a faster program with reduced storage requirements.

The program formulation is first presented, including formulation of the cost function, the model transfer function, the gradient calculation, and the techniques for scaling and constraining the model parameters. The computer program is discussed, and a sample problem illustration is given. In appendices A to D the report symbols, the computer program, the sample problem computer listing, and the proofs for various algorithms are presented, respectively.

## GENERAL PROGRAM FORMULATION

### Cost Function

The desired result is to match a model transfer function to given plant magnitude A(w) and phase $\theta(w)$ frequency response data over a range of frequencies w. Expressed in the form of a complex number $A(w)e^{j\theta(w)}$, the plant frequency response should be equivalent to the model transfer function $G(jw,\underline{b})$ where jw is the Fourier variable and $\underline{b}$ is a vector of model parameters. A cost function $J(\underline{b})$ representing the fit error between the model and the plant data is formulated to be the integral of the absolute value of the error squared between the model and the data.

$$J(\underline{b}) = \int_0^\infty \left| G(jw,\underline{b}) - A(w)e^{j\theta(w)} \right|^2 dw \qquad (1)$$

In the computer program all calculations are in the frequency domain. However, as an interpretation of equation (1), Parseval's theorem can be used to transform it into the time domain. As shown in reference 5, the result is that the cost function is the integral of the error squared between the plant and model impulse responses.

The equation (1) is a continuous integral and is not suitable for evaluation with a digital computer; it is also not suitable for tabulated frequency response data. If the tabulated data are a sufficiently representative sample of the plant, the integral can be approximated by replacing the integration with a piecewise linear summation (trapezoidal rule):

$$J(\underline{b}) = \frac{1}{2}\sum_{i=1}^{N_d} \left| G(jw_i, \underline{b}) - A(w_i)e^{j\theta(w_i)} \right|^2 (w_{i+1} - w_{i-1}) \qquad (2)$$

where $w_0 \equiv w_1$, $w_{N_d + 1} \equiv w_{N_d}$, and $N_d \geq 2$ is the number of data points in the frequency response. Equation (2) is not the usual form of the trapezoidal rule. It has been modified to increase the program speed. The steps in obtaining the equation (2) form from the trapezoidal rule are presented in appendix D.

The choice of the frequencies for the tabulated data can be important. The frequency interval $(w_{i+1} - w_{i-1})$ weights the fit error at each frequency of equation (2). If the frequencies are widely spaced, the value of the error at a particular frequency could have a disproportionate effect. In certain cases it might be desired to favor a close fit between the model and the plant over a portion of the frequency range. The error could be purposely weighted to assign a relative importance to the fit as a function of frequency. This could be done to reflect uncertainties in the data, for example. The computation time increases proportionately with $N_d$. Therefore, the user may wish to select a limited number of data points to represent the plant. This is the likely case for pulse testing followed by a fast Fourier transform of the data, which produces a large amount of closely spaced frequency response data. The measurement noise on actual systems has not been too favorable for pulse testing to date, however. The single frequency sinusoidal testing, by comparison, concentrates power at each frequency, virtually eliminating serious measurement noise problems.

## Model Transfer Function

The model transfer function is assumed to have a linear structure expressed in factored form. This is the general form familiar to Bode plot users. In this form the transfer function $G(s, \underline{b})$, where $s$ is the Laplace variable, consists of the product of a number of possibly different parameter factors.

$$G(s, \underline{b}) = s^{k_1} \prod_{\ell=1}^{m_p - n_q} g_m(s, \underline{b})_\ell \qquad m = 1, 2, 3, 4, 6, 8 \qquad (3a)$$

The $g_m(s, \underline{b})$ of equation (3a) are the following parameter factors:

$$g_1(s,\underline{b}) = \frac{s}{b1_k} + 1 \tag{3b}$$

$$g_2(s,\underline{b}) = b2 \tag{3c}$$

$$g_3(s,\underline{b}) = \left(\frac{s}{b3_k} + 1\right)^{-1} \tag{3d}$$

$$g_4(s,\underline{b}) = \frac{s^2}{b5_k^2} + \frac{2sb4_k}{b5_k} + 1 \tag{3e}$$

$$g_6(s,\underline{b}) = \left(\frac{s^2}{b7_k^2} + \frac{2sb6_k}{b7_k} + 1\right)^{-1} \tag{3f}$$

$$g_8(s,\underline{b}) = e^{-sb8} \tag{3g}$$

The factors of equations (3b), (3d), (3e), and (3f) are assumed to be repeatable in equation (3a). The elements of the $\underline{b}$ vector are the zeros $b1_k$, the gain b2, the poles $b3_k$, the quadratic zero damping ratios $b4_k$, the quadratic zero natural frequencies $b5_k$, the quadratic pole damping ratios $b6_k$, the quadratic pole natural frequencies $b7_k$, and the exponential (dead time) b8. The number of $g_m(s,\underline{b})$ factors in the transfer function is the number of parameters in the $\underline{b}$ vector $m_p$ minus the number of quadratic factors $n_q$. The exponent of the free $s$ is $k_1$. For example, a particular $G(s,\underline{b})$, where

$$\underline{b}^T = (b2, b8, b1_1, b1_2, b6_1, b7_1)$$

is

$$G(s,\underline{b}) = \frac{b2 \cdot e^{-sb8}\left(\frac{s}{b1_1} + 1\right)\left(\frac{s}{b1_2} + 1\right)}{s\left(\frac{s^2}{b7_1^2} + \frac{2sb6_1}{b7_1} + 1\right)} \tag{4}$$

## Gradient Calculation

The gradient of $J(\underline{b})$ with respect to $\underline{b}$ is information required by a gradient search to find parameters $\underline{b}$ that minimize $J(\underline{b})$. The gradient can be expressed analytically by differentiating the equation (2) cost function. The result can be expressed in a form that facilitates rapid computation by invoking the following identiy for a complex function E:

$$(|E|^2)' = (EE^*)' = E(E')^* + E'E^* = 2\mathcal{R}e\ (E(E')^*) \tag{5}$$

Here, the prime indicates a derivative and the asterisk indicates a complex conjugate. A proof of this identity is presented in appendix D. The gradient of J(b) is

$$\nabla J(\underline{b}) = \mathcal{R}e \sum_{i=1}^{N_d} \left[ G(jw_i, \underline{b}) - A(w_i) e^{j\theta(w_i)} \right] \nabla G(jw_i, \underline{b})^* (w_{i+1} - w_{i-1}) \tag{6}$$

Furthermore, the $\nabla G(jw_i, \underline{b})$ factor can be written to include all the possible combinations of factors for $G(jw_i, \underline{b})$ by defining a vector $Z(s, \underline{b})$ such that

$$\nabla G(s, \underline{b}) = G(s, \underline{b}) Z(s, \underline{b}) \tag{7}$$

By using $G(s, \underline{b})$, which is already computed, the expression is computationally efficient. The $Z(s, \underline{b})$ elements are internal to the computer program which assigns to each parameter the proper form. The details of the $Z(s, \underline{b})$ forms are presented in appendix D.

## Parameter Scaling

It was soon apparent from initial trials without parameter scaling that many searches would not converge. Gradient search techniques generally require parameter scaling to obtain efficient search convergence. An ideal scaling rule would obtain nearly spherical cost functions (where the cost is more or less equally sensitive to each parameter of $\underline{b}$) and would tend to result in rapidly converging searches. The scaling law adopted applies to the specific form of the factored transfer function parameters. A rule of thumb scaling law was incorporated such that parameters tend to have equal influence. This law scales each parameter by its own magnitude. To show this, a new

scaled parameter vector $\underline{p}$ with elements $p_i$ is defined

$$p_i \equiv \frac{1}{(b_i)_h} b_i \tag{8}$$

where the $(b_i)_h$ is the magnitude of the $\underline{b}$ vector elements $b_i$ at the start of each line search. The elements of the scaled gradient $J(\underline{p})$ are formed using the chain rule and equation (8):

$$\frac{\partial J}{\partial p_i} = \frac{\partial b_i}{\partial p_i} \frac{\partial J}{\partial b_i} = (b_i)_h \frac{\partial J}{\partial b_i} \tag{9}$$

The gradient is computed only at the start of each line search. At this time $b_i = (b_i)_h$. Thus, the scaled gradient terms are merely the unscaled gradient terms multiplied by their corresponding parameter values.

The $\underline{p}$ vector does not actually appear in the computer program. To avoid a continual multiplication of the scaling factor times its scaled parameter and gradient, the search subroutine was specialized to handle the actual parameters and the scaled gradient.

Normally, a conjugate gradient search is rescaled only after completing a number of line search iterations equal to at least the number of parameters plus one. But rescaling here is done after each iteration and is a nonstandard modification to the conjugate algorithm of reference 8. The advantages for the search program are less logic and storage requirements for scaling parameters to their current value. In tests there were no significant differences found in either the number of iterations or convergence accuracy between rescaling every iteration and after the usual number. This is believed to result from the fact that away from the minimum $J(\underline{b})$ is likely to be a poor approximation to the theoretically required quadratic $\underline{b}$ costs and near the minimum scaling changes are small.

## Parameter Constraints

Ideally, it would not be necessary to constrain the model parameters for stability as the infinite error cost function boundary separating stable and unstable models would be reflected in the cost function $J(\underline{b})$. However, large changes in $\underline{b}$ during the search may enable a jump across a relatively narrow error boundary. The solution to this problem was to force a sharp step size reduction on parameters trying to cross zero.

Zero is critical because it is the point of instability for factored parameters. A modification to the gradient search algorithm constrains the parameters to remain the same sign during the search. This is done by assigning to a parameter about to cross zero a small value on the same side of zero. Thus; an initially stable model must remain stable. This action also prevents changes between right and left hand plane zeroes, and it also prevents changes between dead and lead time exponentials.

## COMPUTER PROGRAM AND SAMPLE PROBLEM

The FORTRAN IV computer program reported here mechanizes the transfer function parameter identification. The usual procedure to identify experimental plant data consists of entering tabulated plant amplitude $A(w_i)$ and phase shift $\theta(w_i)$ frequency response data and a trial factored form transfer function. The user also supplies roughly estimated parameter values and a code number to identify them as time constants, gains, etc. The program can then be directed to iterate on the supplied parameter values until it has optimized the fit to the data. The program output contains the optimized parameter values and a number indicative of the fit error J. Additional output includes the model frequency response. The computer program will not change the signs of the user supplied parameter values nor will it change the given transfer function structure.

The computer program was written primarily for a conversational (time-sharing) computer, but it could also be run on a batch process computer. The effort to keep down program size and increase program speed was to enable its use on a minicomputer. The basic program was run on an SEL 810-B minicomputer in a nearly on-line identification implementation. The intent to obtain a small program package has ruled against writing a program with such conveniences as a plotting package, error detection routines, and alternative search routines. The conjugate gradient search was chosen here because of its simplicity and ease in rescaling and constraining the parameters. Other search routines may converge in fewer iterations. But this could be overbalanced by the resulting increase in computing complexity and scaling and constraining problems. It has not been required to test the program on higher order models (greater than 15 parameters), but generally long convergence times are indicated.

There are provisions in the computer program to make it more general. On occasions it may be desired to not iterate on known parameters in the model, but to leave them fixed. This can be accomplished in the program through the computer variable JD, which is the number of fixed variables. Another option is helpful for reducing higher order analytical models to lower order models. Known plant transfer functions can be entered (rather than tabulated plant frequency response data) by setting the computer variable INST to 5.

The computer program may be used to design a series compensator that will make a plant look like any desired transfer function. For example, the desired transfer function for the plant plus compensator divided by the plant may be entered or built. Then the compensator assumes the role of the model transfer function.

The program package consists of a main program called MODEL and two subroutines called CGFM and CFG. The MODEL program organizes the data and is written primarily for a time-sharing type of computer. The subroutine CGFM conducts the conjugate gradient search. The subroutine CFG does the actual work of computing the cost function and gradient. These programs are described in detail in appendix B.

The problem is to fit transfer function models to given experimental frequency response data using the computer program. The data are the response of an inlet-pressure signal to a bypass-door control area and is taken from reference 6. The data are the points plotted in figure 1. In handling experimental data the user usually does not know beforehand the model structure. The most useful analytical model may be a compromise between lower order and accuracy of fit to the data. In trying to fit the data a user may iteratively work up to more complex models. The dashed curve of figure 1 is the fit of the following three-parameter model to the data:

$$G(s,\underline{b}) = \frac{1}{\left[\frac{s}{17.5(2\pi)}+1\right]\left\{\left[\frac{s}{130(2\pi)}\right]^2+\frac{2s(0.262)}{130(2\pi)}+1\right\}} \tag{10}$$

The solid curve in figure 1 is the fit of a six-parameter model to the data. A computer listing of this solution is presented in appendix C. There it is shown that the initial estimate of

$$G(s,\underline{b}) = \frac{\left[\frac{s}{1(2\pi)}+1\right]}{\left[\frac{s}{1(2\pi)}+1\right]\left[\frac{s}{10(2\pi)}+1\right]\left[\frac{s}{1000(2\pi)}+1\right]\left\{\left[\frac{s}{1000(2\pi)}\right]^2+\frac{2s(1)}{1000(2\pi)}+1\right\}} \tag{11}$$

converged to the optimized model

$$G(s,\underline{b}) = \frac{\left[\frac{s}{(4.91)(2\pi)}+1\right]}{\left[\frac{s}{(3.70)(2\pi)}+1\right]\left[\frac{s}{(27.0)(2\pi)}+1\right]\left[\frac{s}{(309)(2\pi)}+1\right]\left\{\left[\frac{s}{(145)(2\pi)}\right]^2+\frac{2s(0.265)}{(145)(2\pi)}+1\right\}} \tag{12}$$

after 93 iterations. The six-parameter fit is visibly better over the frequency range of interest, but both are optimum for their given structure.

The initial parameter estimates can affect the search convergence time and sometimes result in detecting a local rather than a global minimum. The convergence times and number of iterations for initial parameter estimates that were off by about a factor of 10 are listed in table I for four different model structures. Convergence times ranged from 24 seconds for the five-parameter model to 1.8 seconds for the two-parameter model (CPU time IBM 360-67 TSS). The fit error $J(\underline{b})$ as calculated by the program for various order models are tabulated in table I. The error ranges from 5.13 for the two-parameter model to 0.59 for the six-parameter model. The program discarded zeroes for models with less than four poles by making them very large. Likewise, the program discarded a four-pole model by making the fourth pole very large. Two simple poles converging to the same value indicated that a quadratic pole would probably do better.

## CONCLUSIONS

A FORTRAN computer program is presented which fits a linear factored form transfer function to given frequency response data. The identification process is often an important first step in system control design. The computer program is based on a conjugate gradient search procedure that minimizes the error between the given frequency-response data and the frequency response of a transfer function that is supplied by the user. The user-supplied transfer function consists of a product of terms that can include a gain, an integrator (or differentiator), a dead time, multiple first-order leads and lags, and multiple second-order leads and lags.

The usual procedure for use of the program consists of entering a table of amplitude and phase-shift frequency-response data and a trial, factored form transfer function; the user also supplies roughly estimated parameter values such as time constants, gains, etc. The computer program then iterates on the supplied parameter values until it has optimized the fit to the experimental data; the excellence of this fit is constrained by the form of the user-supplied transfer function. The computer output consists of the optimized parameter values and a number indicative of the fit error. The computer program does not change the signs of the user-supplied parameter values, nor does it change the given transfer-function structure. The user can "build" a transfer function by trying a series of increasingly complex transfer-function structures being guided by the fit errors in a systematic manner as illustrated by the sample problem.

An alternative use is to reduce the complexity of a high-order transfer function that might result from a dynamic analysis. An option is provided in the program for enter-

ing the complex transfer function in factored form to take the place of the normal table of frequency response data. The simpler model can then be "built" by the usual procedure until it is a reasonable fit to the more complex model.

The conjugate gradient search routine was chosen for the error function minimization because of its simplicity and ease in constraining parameters. In the program, scaling the parameters by their own magnitude was demonstrated to result in good search convergence. Constraining the parameters to remain the same sign during the search insures model stability. The use of time and space saving algorithms resulted in a small and fast program. The program as presented is written primarily for a conversational (time sharing) computer, but it could be run on a batch process computer. The basic program has been run on an SEL 810-B minicomputer as part of a nearly on-line transfer function identification implementation.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, June 23, 1975,
505-05.

## APPENDIX A

## SYMBOLS

| | |
|---|---|
| A | system frequency response magnitude |
| $\underline{b}$ | parameter vector, $m_p \times 1$ |
| $b1_k$ | $\underline{b}$ vector $k^{th}$ zero, rad/sec |
| b2 | $\underline{b}$ vector gain parameter, $(rad/sec)^{-k_1}$ |
| $b3_k$ | $\underline{b}$ vector $k^{th}$ pole, rad/sec |
| $b4_k$ | $\underline{b}$ vector $k^{th}$ quadratic zero damping |
| $b5_k$ | $\underline{b}$ vector $k^{th}$ quadratic zero natural frequency, rad/sec |
| $b6_k$ | $\underline{b}$ vector $k^{th}$ quadratic pole damping |
| $b7_k$ | $\underline{b}$ vector $k^{th}$ quadratic pole natural frequency, rad/sec |
| b8 | $\underline{b}$ vector exponential (dead time), sec |
| c | real variable |
| d | real variable |
| E | complex function, $E = c + jd$ |
| f | real function |
| $f_i$ | $f(x_i)$ |
| G | model transfer function |
| g | transfer function factored parameter factor |
| J | cost function |
| j | imaginary number, $\sqrt{-1}$ |
| jw | Fourier transform variable, rad/sec |
| $k_1$ | integer, exponent of free s's in model |
| $m_p$ | integer, number of elements in $\underline{b}$ vector |
| $N_d$ | integer, number of frequency points in numerical integration |
| $n_q$ | integer, number of quadratic factors |
| $\underline{p}$ | transformed parameter vector, $m_p \times 1$ |
| s | Laplace variable, $sec^{-1}$ |

$w$ frequency, rad/sec

$w_i$ frequency at $i^{th}$ data point, rad/sec

$x_i$ variable at $i^{th}$ data point

$Z$ partial product in gradient vector, $m_p \times 1$

$\theta$ system frequency response phase, deg

Superscripts:

$'$ derivative

$*$ complex conjugate

$T$ transpose

Subscripts:

$( )_h$ value at start of iteration

$i$ $i^{th}$ element

$\ell$ $\ell^{th}$ factored parameter factor

$m$ type of factored parameter factor

## APPENDIX B

## COMPUTER PROGRAM

The FORTRAN computer program package for transfer function parameter identification consists of the main program MODEL, subroutine CFG to calculate the cost function and gradient, and subroutine CGFM to conduct the conjugate gradient search. Symbols used in the programs are defined separately for each program. The programs are dimensioned for a $N_d$ frequency point maximum of 50 and $m_p$ model parameter maximum of 15. The vectors dimensioned for $N_d$ are AMP, PHA, and PLANT. The vectors dimensioned for $m_p$ are B, G, ID, and Z. The vector H is dimensioned $2m_p$ and the vector W is dimensioned $N_d + 1$.

### Program MODEL

The program MODEL handles the data and converts system frequency response data to complex numbers. The computer variables in the program are defined in the section Program MODEL Variable List. A flow chart for the program MODEL is presented in figure 2.

The program starts by printing a heading referencing the namelist variables and variable codes. Then the program prompts for NAM3 namelist data. The namelist variables are entered according to the rules for entering FORTRAN namelist data. The NAM3 variables are AMP, ND, PHA, W, and KPR. The program prints the NAM3 variables if KPR = 1 and then prompts for NAM2 namelist data. The NAM2 variables are B, K1, ID, LIMIT, N, and JD. Then the NAM2 variables are printed and a prompt for the INST variable is printed.

The INST variable is entered in I1 format. The INST code values are 1, 2, 3, 4, 5, and 6. Making INST = 1 causes a search for the optimum parameters. After the search results are printed, another INST prompt is issued. Making INST = 2 returns the program to request NAM2. Making INST = 3 returns the program to request NAM3. Making INST = 4 causes the model response to be printed. The frequency is printed under W, the model frequency response amplitude under AMP, and phase under PHA. Then the zero iteration search results are printed and another INST prompt is issued. Making INST = 5 is the same as making INST = 4 except that the PLANT complex numbers are set equal to the model. This option is used to save entering AMP and PHA data and to build an analytical model for the system. Making INST = 6 stops the program.

The frequencies W are input in radians, but they may also be input in hertz if the

B vector is appropriately scaled by $2\pi$ (see B vector in Program MODEL Variable list).

The following is a FORTRAN listing for the program MODEL[1]:

```
C       PROGRAM MODEL FOR TRANSFER FUNCTION IDENTIFICATION
        DIMENSION B(15),G(15),ID(15),AMP(50),PHA(50),W(51),H(30)
        COMPLEX PLANT(50)
        EXTERNAL CFG
        LOGICAL KG
        COMMON/IDTN/PLANT,W,ID,K1,INST,ND,KNT,NP
        COMMON/FMC/KOUNT,KG
        NAMELIST /NAM3/W,AMP,PHA,ND,KPR /NAM2/LIMIT,N,K1,B,ID,JD
        WRITE(6,110)
110     FORMAT(' NAM3=(W,AMP,PHA,ND,KPR)',/-
       1,' NAM2=(B,ID,K1,LIMIT,N,JD)',/-
       2,' ID=(1=Z;2=G;3=P;4,5=CZD,CZW;6,7=CPD,CPW;8=DT)',/-
       3,' INST=(1=SEARCH,2=NAM2,3=NAM3,4=PRINT,5=PLANT=MODEL,6=-
       4STOP)',/,' IER=(0=CONV,1=NOT CONV,2=ERROR)' )
10      WRITE (6,180)
180     FORMAT(' NAM3?')
        READ (5,NAM3)
        IF(KPR.EQ.1) WRITE(6,310)
310     FORMAT(6H PLANT,/,8X,1HW,14X,3HAMP,11X,3HPHA)
120     FORMAT(1P3E15.5)
        DO 20  J=1,ND
        IF(KPR.EQ.1) WRITE(6,120) W(J),AMP(J),PHA(J)
        THETA=PHA(J)*.017453292
20      PLANT(J)=AMP(J)*CEXP(CMPLX(0.,THETA))
        W(ND+1)=W(ND)
30      WRITE (6,181)
181     FORMAT(' NAM2?')
        READ (5,NAM2)
        NP=N+JD
        WRITE(6,130) LIMIT,N,K1,JD,(ID(J),J=1,NP)
130     FORMAT(' LIMIT,N,K1,JD=',4I3,'   ID=',15I3)
        GO TO 87
777     WRITE (6,182)
182     FORMAT(' INST?')
        READ (5,140) INST
140     FORMAT (I1)
        GO TO (40,30,10,50,45,1000),INST
45      WRITE(6,145)
145     FORMAT(' PLANT SET EQUAL TO MODEL')
50      WRITE(6,160)
160     FORMAT(6H MODEL,/,8X,1HW,14X,3HAMP,11X,3HPHA)
40      SIZE=.1
        EPS=1.E-5
        ITER=LIMIT*(1-INST/4)
        KNT=0
        CALL CGFM(CFG,N,B,F,G,SIZE,EPS,ITER,IER,H)
        WRITE(6,150) F,IER,KOUNT,KNT,SIZE
150     FORMAT(' F=',1PE16.7,'  IER=',I2,'  KOUNT,KNT,SIZE=',-
      1 I3,I4,1PE10.3)
87      WRITE (8,151) (B(J),J=1,NP)
151     FORMAT(' B=',1P8E11.4)
```

[1]A dash at end of a line signals continuation card follows.

```
      GO TO 777
1000  STOP
      END
```

Program Model Variable List

| | |
|---|---|
| AMP | system frequency response magnitude A(w), vector (input variable) |
| B | model parameter b, vector (input variable). Damping ratios must immediately precede their natural frequency terms. Constant (JD) parameters must follow last. |
| CFG | subroutine (declared external in model program) which computes the cost function and gradient |
| EPS | parameter change defining search convergence, e.g., $10^{-5}$ |
| F | F = J(b), cost function |
| G | cost function (scaled) gradient, vector |
| H | storage, vector |
| ID | integer vector that identifies corresponding parameter in B as to type (input variable): 1 = zero, 2 = gain, 3 = pole, 4 = complex zero damping, 5 = complex zero natural frequency, 6 = complex pole damping, 7 = complex pole natural frequency, 8 = dead time |
| IER | search convergence parameter: 0 = search convergence (SIZE reduced to EPS or less within LIMIT iterations), 1 = search not converged, 2 = probable error occurred |
| INST | branching instruction parameter (input variable): 1 = search for optimum, 2 = return to NAM2 namelist, 3 = return to NAM3 namelist, 4 = print frequency and model frequency response amplitude and phase, 5 = set converted system PLANT frequency responses equal to model values and print, 6 = stop |
| ITER | set equal to LIMIT except for the INST = 4 or 5 in which case ITER = 0 |
| J | index of element in vector |
| JD | number of noniterative constant parameters in model (input variable) |
| KG | logical variable .TRUE. means compute gradient |
| KNT | count of cost function evaluations |
| KOUNT | count of line search iterations |

| | |
|---|---|
| KPR | if equal to 1 causes NAM3 variables to be printed (input variable) |
| K1 | exponent of free s in model (input variable) |
| LIMIT | maximum number of iterations (input variable) |
| N | number of iterative parameters of B in model (input variable) |
| ND | number of frequency points over which integration is performed $N_d$ |
| NP | total number, iterative plus constant, of model parameters, NP = N + JD |
| PHA | system frequency response phase $\theta(w)$, in deg vector (input variable) |
| PLANT | converted system response to equivalent complex numbers vector |
| SIZE | parameter step size; e.g., set to 0.1 at start of search |
| THETA | scratch variable for radian phase |
| W | radian frequency w, vector (input variable); can be input in hertz instead of radians but then the units of parameters in B may change. (The B parameters with ID numbers of 1, 3, 5, and 7 need to be divided by $2\pi$ to obtain their corresponding value in hertz, a B with an ID = 8 needs to be multiplied by $2\pi$, and B's with ID = 2, 4, and 6 are not affected) |

## Subroutine CFG (N, B, F, G)

The purpose of this subroutine is to compute the cost function and gradient. The gradient G is the scaled gradient $\nabla J(\underline{p})$. The unscaled gradient for other search programs could be obtained by dividing each term in G by the corresponding parameter in B.

The CFG variables are defined in the CFG Variable List. Those variables carried over in the common block and subroutine call are labeled the same as those described in the program MODEL and are not repeated again. Figure 3 is a flow chart of CFG.

## FORTRAN Listing for Subroutine CFG

```
SUBROUTINE CFG(N,B,F,G)
DIMENSION B(1),G(1),ID(15),W(51)
COMPLEX PLANT(50),Z(15),S,ST,GM,E
LOGICAL KG
COMMON/IDTN/PLANT,W,ID,K1,INST,ND,KNT,NP
COMMON/FMC/KOUNT,KG
KNT=KNT+1
DO 30 J=1,N
```

```
30      G(J)=0.
        WS=W(1)
        F=0.
        DO 100 J=1,ND
        S=CMPLX(0.,W(J))
        GM=S**K1
        DO 8 K=1,NP
        IDK=ID(K)
        GO TO (1,3,1,2,8,2,8,4),IDK
1       GM=GM*(S/B(K)+1.)**(2-IDK)
        IF(KG) Z(K)=S/(S+B(K))*(FLOAT(IDK)-2.)
        GO TO 8
2       ST=(S/B(K+1))**2+2.*S*B(K)/B(K+1)+1.
        GM=GM*ST**(5-IDK)
        IF(.NOT.KG) GO TO 8
        Z(K)=2.*S*B(K)/(B(K+1)*ST)*(5.-FLOAT(IDK))
        Z(K+1)=-Z(K)*(1.+S/(B(K)*B(K+1)))
        GO TO 8
3       GM=GM*B(K)
        Z(K)=CMPLX(1.,0.)
        GO TO 8
4       GM=GM*CEXP(-S*B(K))
        Z(K)=-S*B(K)
8       CONTINUE
        E=GM-PLANT(J)
        F=F+(W(J+1)-WS)*REAL(E*CONJG(E))
        IF(.NOT.KG) GO TO 100
        ST=(W(J+1)-WS)*E*CONJG(GM)
        DO 80 K=1,N
80      G(K)=G(K)+REAL(ST*CONJG(Z(K)))
        IF(INST.EQ.1) GO TO 100
        AM=CABS(GM)
        DEG=ATAN2(AIMAG(GM),REAL(GM))*57.295780
        WRITE(6,120) W(J),AM,DEG
120     FORMAT (1P3E15.5 )
        IF(INST.EQ.5) PLANT(J)=GM
100     WS=W(J)
        RETURN
        END
```

Subroutine CFG Variable List

| | |
|---|---|
| AM | $\lvert G(jw_i, \underline{b})\rvert$ |
| DEG | $\angle G(jw_i, \underline{b})$, deg |
| E | error between model and data |
| GM | partial product becomes $G(s, \underline{b})$ |
| IDK | ID(K) value |
| J | frequency index |
| K | parameter index |

| | |
|---|---|
| S | s |
| ST | temporary value |
| WS | saved W value |
| Z | partial product in model gradient, vector |

## Subroutine CGFM (CFG, N, B, F, G, SIZE, EPS, ITER, IER, H)

The purpose of subroutine CGFM is to perform the conjugate gradient search function minimization. Several nonstandard modifications relative to the conjugate gradient search described in reference 8 exist in the CGFM subroutine. The gradient is the scaled gradient $\nabla J(\underline{p})$ and every iteration updated $\underline{b}$ parameters change the scaled coordinate system. Another nonstandard modification to the search is that the signs of the parameters $\underline{b}$ are not allowed to change during the search.

The minimum for each line search is found by either doubling or taking one third of the parameter values. The process is continued until the minimum cost is known to be bracketed. Then the minimum point is obtained from the minimum of a quadratic curve fit to three points. The gradient is not used in this process. To increase the program speed, there is logic to signal whether the gradient is needed; if not, only the cost function value is required of the CFG subroutine.

The CGFM program variables are defined in the CGFM program variable list. Variables carried over in the common block and subroutine call are labeled the same as those described in the main program and are not repeated again. Figure 4 is a flow chart of CGFM.

## FORTRAN Listing of Subroutine CGFM

```
      SUBROUTINE CGFM(CFG,N,B,F,G,SIZE,EPS,ITER,IER,H)
      DIMENSION B(1),H(1),G(1)
      LOGICAL KG
      COMMON/FMC/KOUNT,KG
C  INITIALIZATIONS
      KOUNT=0
      STEP=2.
      IER=-5
    5 BETA=0.
      NCYC=0
   15 K=0
      DO 20 J=1,N
   20 H(J)=B(J)
      KG=.TRUE.
      CALL CFG(N,B,F,G)
```

```
C  TEST FOR STOPPING SEARCH
      IF(KOUNT.GE.ITER) IER=1
      IF(SIZE.LT.EPS) IER=0
   40 IF(IER.GT.-2) RETURN
C  COMPUTE PAST GRADIENT WEIGHTING
      TSQR=0.
      DO 50 J=1,N
   50 TSQR=TSQR+G(J)**2
      IF(NCYC.EQ.0) GO TO 60
      BETA=TSQR/TSAVE
   60 SCALE=0.
      DO 70 J=1,N
      JN=J+N
      H(JN)=-G(J)+BETA*H(JN)
   70 SCALE=SCALE+ABS(H(JN))
      IF(SCALE.GT.0.) GO TO 80
      IER=0
      GO TO 40
   80 SCALE=SIZE/SCALE
      TSAVE=TSQR
      NCYC=NCYC+1
      FSS=F
      L=1
C  UPDATE B'S
  100 DO 110 J=1,N
      JN=J+N
110   B(J)=H(J)*ABS(1.+SCALE*H(JN))
      FS=F
      KG=.FALSE.
      CALL CFG(N,B,F,G)
C  LOGIC CHANGES LINE SEARCH STEP SIZE OR CONCLUDES SEARCH
      IF(K.GT.0) GO TO 120
      IF(F.LT.FS) GO TO 130
      IF(L.GT.1) GO TO 140
      GO TO 150
  120 IF(F.LT.FSS) GO TO 140
      GO TO 160
  130 SCASV = SCALE
      L= L+1
      SCALE= SCALE*STEP
      FSS=FS
      IF(L.LT.15) GO TO 100
      IER=2
      GO TO 40
C  FIT QUADRATIC CURVE TO 3 PTS. BRACKETING LINE SEARCH MIN.
140   DO 148 J=1,N
      JN=J+N
      R1=H(J)
      R2=H(J)*ABS(1.+SCASV*H(JN))
      R3=B(J)
      IF(L.GT.3) R1=R1+(R2-R1)/STEP
      X1=(FSS-FS)*(R1-R3)
      X2=(FSS-F)*(R2-R1)
      IF(ABS(R2-R3).GT.EPS/4.) GO TO 147
      IF(L.GT.1) B(J)=R2
      GO TO 148
147   B(J)=(X1*(R1+R3)+X2*(R1+R2))/((X1+X2)*2.)
148   IF(B(J)*H(J).LE.0.) B(J)=-.1*B(J)+EPS*R3
```

```
C  UPDATE SEARCH VARIABLES
       SIZE=SIZE*(FLOAT(L)+2.)/4.
       KOUNT=KOUNT+1
       IF(NCYC.GT.N) GO TO 5
       GO TO 15
  150  SCASV= SCALE
       SCALE= SCALE/(1.+STEP)
       K= K+1
       SIZE= SIZE/(1.+STEP)
       GO TO 100
  160  SIZE= SIZE/(1.+STEP)
       DO 180 J=1,N
  180  B(J)=H(J)
       GO TO 5
       END
```

CGFM Program Variable List

| | |
|---|---|
| BETA | conjugate direction weighting |
| FS | saved F |
| FSS | saved FS |
| J | parameter index |
| JN | J + N |
| K | indicator for step size reductions |
| L | number step size increases within iteration |
| NCYC | number of iterations before restarting conjugate search |
| R1, R2, R3 | terms in quadratic curve fit |
| SCALE | step size scale factor |
| SCASV | saved SCALE |
| STEP | step size |
| TSAVE | saved TSQR |
| TSQR | squared gradient terms sum |
| X1, X2 | partial product |

## APPENDIX C

## SAMPLE PROBLEM LISTING

The computer listing for the optimization of a six-parameter model is presented:

```
 1 NAM3=(W,AMP,PHA,ND,KPR)
 2 NAM2=(B,ID,K1,LIMIT,N,JD)
 3 ID=(1=Z;2=G;3=P;4,5=CZD,CZW;6,7=CPD,CPW;8=DT)
 4 INST=(1=SEARCH,2=NAM2,3=NAM3,4=PRINT,5=PLANT=MODEL,6=STOP)
 5 IER=(0=CONV,1=NOT CONV,2=ERROR)
 6 NAM3?
 7  &nam3 w=1,3,7,10,15,20,25,30,35,40,50,60,70,80,90,100,110,120,130,140,
 8  amp=1,.95,.77,.7,.67,.63,.6,.53,.48,.44,.35,.31,.33,.35,2*.32,.3,.29,
 9  .27,.26,pha=-2,-13,-24,-31,-35,-44,-57,-62,-71,-75,-87,-110,-92,-105,
10  -119,-128,-145,-156,-166,-172,nd=20,kpr=0 &end
11 NAM2?
12  &nam2 b=1,1,10,1000,1,1000,id=1,3,3,3,6,7,k1=0,limit=200,n=6,jd=0 &end
13 LIMIT,N,K1,JD=200   6   0   0    ID=   1   3   3   3   6   7
14 B= 1.0000E 00 1.0000E 00 1.0000E 01 1.0000E 03 1.0000E 00 1.0000E 03
15 INST?
16 1
17 F=    5.8507025E-01  IER= 0  KOUNT,KNT,SIZE= 93 417 2.868E-06
18 B= 4.9144E 00 3.6998E 00 2.6970E 01 3.0934E 02 2.6459E-01 1.4546E 02
19 INST?
20 6
   CHCRW400      TERMINATED: STOP
```

The listing shows the plant data entered in NAM3 namelist, a model built in NAM2 namelist, and a search for the optimum parameters. The line numbers were added for discussion purposes, and program responses are capital letter while user input are lowercase. A line by line discussion of the listing follows:

Lines 1 to 5 - For convenience the program listing of the NAM3 and NAM2 variables and the ID, INST, and IER variable codes

Line 6 - A program prompt for NAM3 namelist data

Lines 7 to 10 - The user entered system NAM3 frequency response data

(The frequencies of W are entered in hertz rather than in radians. This requires associating a conversion factor of $2\pi$ with certain parameters of B.)

Line 11 - A program prompt for NAM2 namelist data

Line 12 - User entered NAM2 namelist data required to build the iterative six-parameter transfer function model (See eq. (11).)

Lines 13 and 14 - Program verification of the NAM2 namelist data

Line 15 - A program prompt for a branching instruction

Line 16 - A user input 1 directing the program to search for the optimum parameters

Lines 17 and 18 - Program output of search results (Upon convergence the cost function was reduced to 0.585 after 93 iterations and 417 cost function evaluations. The optimized parameters are listed in eq. (12).)

Line 19 - Another program prompt for a branching instruction

Line 20 - A user input 6 to stop the program

In the sample problem parameter optimization the KPR variable was not set to 1; therefore, there was no recording of the plant data. The INST variable was not set to 4; therefore, there was no recording of the model frequency response. The following listing shows the output upon recalling the program for the case KPR = 1 and INST = 4, but with the NAM3 and NAM2 namelist data otherwise unchanged.

```
     Sample problem plant and model frequency response

NAM3=(W,AMP,PHA,ND,KPR)
NAM2=(B,ID,K1,LIMIT,N,JD)
ID=(1=Z;2=G;3=P;4,5=CZD,CZW;6,7=CPD,CPW;8=DT)
INST=(1=SEARCH,2=NAM2,3=NAM3,4=PRINT,5=PLANT=MODEL ,6=STOP)
IER=(0=CONV,1=NOT CONV,2=ERROR)
NAM3?
 &nam3 kpr=1 &end
PLANT
        W                AMP              PHA
    1.00000E 00     1.00000E 00    -2.00000E 00
    3.00000E 00     9.50000E-01    -1.30000E 01
    7.00000E 00     7.70000E-01    -2.40000E 01
    1.00000E 01     7.00000E-01    -3.10000E 01
    1.50000E 01     6.70000E-01    -3.50000E 01
    2.00000E 01     6.30000E-01    -4.40000E 01
    2.50000E 01     6.00000E-01    -5.70000E 01
    3.00000E 01     5.30000E-01    -6.20000E 01
    3.50000E 01     4.80000E-01    -7.10000E 01
    4.00000E 01     4.40000E-01    -7.50000E 01
    5.00000E 01     3.50000E-01    -8.70000E 01
    6.00000E 01     3.10000E-01    -1.10000E 02
    7.00000E 01     3.30000E-01    -9.20000E 01
    8.00000E 01     3.50000E-01    -1.05000E 02
    9.00000E 01     3.20000E-01    -1.10000E 02
    1.00000E 02     3.20000E-01    -1.28000E 02
    1.10000E 02     3.00000E-01    -1.45000E 02
    1.20000E 02     2.90000E-01    -1.56000E 02
    1.30000E 02     2.70000E-01    -1.66000E 02
    1.40000E 02     2.60000E-01    -1.72000E 02
NAM2?
 &nam2  &end
LIMIT,N,K1,JD=200   6   0   0    ID=   1   3   3   3   6   7
B= 4.9144E 00 3.6998E 00 2.6970E 01 3.0934E 02 2.6459E-01 1.4546E 02
INST?
4
```

```
MODEL
          W                AMP              PHA
    1.00000E 00     9.84501E-01    -6.14033E 00
    3.00000E 00     9.04736E-01    -1.51638E 01
    7.00000E 00     7.88533E-01    -2.45215E 01
    1.00000E 01     7.40270E-01    -3.01572E 01
    1.50000E 01     6.77591E-01    -3.92998E 01
    2.00000E 01     6.21103E-01    -4.78250E 01
    2.50000E 01     5.69208E-01    -5.55071E 01
    3.00000E 01     5.22819E-01    -6.23598E 01
    3.50000E 01     4.82319E-01    -6.84946E 01
    4.00000E 01     4.47492E-01    -7.40447E 01
    5.00000E 01     3.92663E-01    -8.38764E 01
    6.00000E 01     3.53632E-01    -9.26646E 01
    7.00000E 01     3.26457E-01    -1.01007E 02
    8.00000E 01     3.08326E-01    -1.09387E 02
    9.00000E 01     2.97201E-01    -1.18260E 02
    1.00000E 02     2.91303E-01    -1.28117E 02
    1.10000E 02     2.88359E-01    -1.39502E 02
    1.20000E 02     2.84628E-01    -1.52927E 02
    1.30000E 02     2.74384E-01    -1.68562E 02
    1.40000E 02     2.52175E-01     1.74280E 02
F=    5.8507025E-01  IER= 1  KOUNT,KNT,SIZE=  0    1 1.000E-01
B= 4.9144E 00 3.6998E 00 2.6970E 01 3.0934E 02 2.6459E-01 1.4546E 02
INST?
```

## APPENDIX D

## TRAPEZOIDAL RULE

The trapezoidal rule for the approximation of an integral is

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx \frac{x_{i+1} - x_i}{2}\left[f(x_{i+1}) + f(x_i)\right]$$

This leads naturally to the series expression for a piecewise linear summation of trapezoidal areas

$$\sum_{i=1}^{N_d-1} \frac{1}{2}(f_{i+1} + f_i)(x_{i+1} - x_i)$$

where the notation $f(x_i)$ is shorted to $f_i$. Note that for each value of i the series requires the two values $f_i$ and $f_{i+1}$. This is reduced to a single value $f_i$ as described next. First, the series is written in expanded form

$$\frac{1}{2}(f_2x_2 + f_1x_2 - f_2x_1 - f_1x_1) + \frac{1}{2}(f_3x_3 + f_2x_3 - f_3x_2 - f_2x_2) + \ldots$$

$$+ \frac{1}{2}\left(f_{N_d}x_{N_d} + f_{N_d-1}x_{N_d} - f_{N_d}x_{N_d-1} - f_{N_d-1}x_{N_d-1}\right)$$

The expansion after cancellation of terms from adjacent lines is

$$\frac{1}{2}(0 + f_1x_2 - f_2x_1 - f_1x_1) + \frac{1}{2}(0 + f_2x_3 - f_3x_2 - 0) + \ldots$$

$$+ \frac{1}{2}\left(f_{N_d}x_{N_d} + f_{N_d-1}x_{N_d} - f_{N_d}x_{N_d-1} - 0\right)$$

A regrouping of the expansion forms the series

$$\frac{1}{2}\left[f_1(x_2 - x_1) + f_2(x_3 - x_1) + f_3(x_4 - x_2) + \ldots + f_{N_d}\left(x_{N_d} - x_{N_d-1}\right)\right]$$

The series is expressed in the desired form by defining $x_0 \equiv x_1$ and $x_{N_d+1} = x_{N_d}$. Thus,

$$\frac{1}{2}\sum_{i=1}^{N_d} f(x_i)(x_{i+1} - x_{i-1})$$

## Identity

If E is assumed to be a complex function of two real variables c and d,

$$E = c + jd$$

it readily follows that

$$(|E|^2)' = \left[(c + jd)(c - jd)\right]' = (c^2 + d^2)' = 2(cc' + dd')$$

For another starting point, namely, $2\,\mathcal{R}e\,(E(E')^*)$, the result is

$$2\,\mathcal{R}e\,(E(E')^*) = 2\,\mathcal{R}e\left[(c + jd)(c' - jd')\right] = 2(cc' + dd')$$

Thus, the two expressions are the same, and

$$(|E|^2)' = 2\,\mathcal{R}e\,(E(E')^*)$$

## Gradient

The cost function gradient (eq. (6)) contains the transfer function gradient. The transfer function gradient in equation (7) is the transfer function times a vector $Z(s, \underline{b})$.

$$\nabla G(s, \underline{b}) = \left[G(s, \underline{b})\right]\left[Z(s, \underline{b})\right] \tag{7}$$

The transfer function $G(s, \underline{b})$ is the product of factored parameter factors $g_m(s, \underline{b})$ (eq. (3a) to (3g)). The elements of $Z(s, \underline{b})$ are the partial derivatives divided by the

corresponding factor. For example, the $Z(s,\underline{b})$ element for a $b1_k$ factor is

$$\frac{\dfrac{\partial\left(\dfrac{s}{b1_k}+1\right)}{\partial b1_k}}{\left(\dfrac{s}{b1_k}+1\right)} = \frac{\dfrac{-s}{b1_k^2}}{\left(\dfrac{s}{b1_k}+1\right)}$$

There are eight structural forms for the $Z(s,\underline{b})$ elements corresponding to the eight possible parameter forms. The terms are simple, and there is only a change in sign between corresponding numerator and denominator factor forms.

$$b1_k: \frac{\dfrac{-s}{b1_k^2}}{\left(\dfrac{s}{b1_k}+1\right)}$$

$$b2: \frac{1}{b2}$$

$$b3_k: \frac{\dfrac{s}{b3_k^2}}{\left(\dfrac{s}{b3_k}+1\right)}$$

$$b4_k: \frac{\dfrac{2s}{b5_k}}{\left(\dfrac{s^2}{b5_k^2}+\dfrac{2s(b4_k)}{b5_k}+1\right)}$$

$$b5_k: \frac{-2s\left(\frac{b4_k}{b5_k^2} + \frac{s^2}{b5_k^3}\right)}{\left(\frac{s^2}{b5_k^2} + \frac{2s(b4_k)}{b5_k} + 1\right)}$$

$$b6_k: \frac{-\frac{2s}{b7_k}}{\left(\frac{s^2}{b7_k^2} + \frac{2s(b6_k)}{b7_k} + 1\right)}$$

$$b7_k: \frac{+2s\left(\frac{b6_k}{b7_k^2} + \frac{s^2}{b7_k^3}\right)}{\left(\frac{s^2}{b7_k^2} + \frac{2s(b6)}{b7_k} + 1\right)}$$

$$b8: -s$$

## REFERENCES


1. D'Azzo, John J.; Houpis, Constantine H.: Feedback Control System Analysis and Synthesis. McGraw-Hill, 1966.

2. Sanathan, C. K.; Koerner, J.: Transfer Function Synthesis as a Ratio of Two Complex Polynomials. IEEE Trans. Auto. Control, vol. AC-8, no. 1, Jan. 1963, pp. 56-58.

3. Strobel, H.: On a New Method of Determining the Transfer Function By Simultaneous Evaluation of the Real and Imaginary Part of the Frequency Response. Paper 1.F, Proc. 3rd IFAC Congress, Session 1, 1966.

4. Jones, N. B.: New Method for Identification and Synthesis of Linear Systems from Frequency Response Data. Proc. Inst. of Elec. Engs., vol. 117, 1970, pp. 1021-1025.

5. Hays, James R.; Clements, William C., Jr.; and Harris, Thomas R.: The Frequency Domain Evaluation of Mathematical Models for Dynamic Systems. Amer. Inst. Chem. Eng. J., vol. 13, No. 2, 1967, pp. 374-378.

6. Baumbick, Robert J.; Neiner, George H.; and Cole, Gary L.: Experimental Dynamic Response of a Two-Dimensional, Mach 2.7, Mixed-Compression Inlet. NASA TN D-6957, 1972.

7. Baumbick, Robert J.; et al.: Dynamic Response of Mach 2.5 Axisymmetric Inlet with 40 Percent Supersonic Internal Area Contraction. NASA TM X-2883, 1973.

8. Fletcher, R.; and Reeves, C. M.: Function Minimization by Conjugate Gradients. Computer J., vol. 7, No. 2, Jul. 1964, pp. 149-154.

TABLE I. - SEARCH RESULTS OF DIFFERENT MODEL STRUCTURES FIT TO SAME SYSTEM DATA

[Optimum parameters listed to three significant figures.]

| Number of parameters, $m_p$ | Fit error, $J(\underline{b})$ | Initial estimate of $G(s,\underline{b})$ | Optimum estimate of $G(s,\underline{b})$ | Number of iterations | CPU time, sec |
|---|---|---|---|---|---|
| 2 | 5.13 | $\left(\frac{s}{2\pi}+1\right)^{-1}\left[\frac{s}{1000(2\pi)}+1\right]^{-1}$ | $\left[\frac{s}{23.3(2\pi)}+1\right]^{-1}\left[\frac{s}{124(2\pi)}+1\right]^{-1}$ | 15 | 1.8 |
| 3 | 1.34 | $\left(\frac{s}{2\pi}+1\right)^{-1}\left\{\left[\frac{s}{1000(2\pi)}\right]^2+\frac{2s}{1000(2\pi)}+1\right\}^{-1}$ | $\left[\frac{s}{17.5(2\pi)}+1\right]^{-1}\left\{\left[\frac{s}{130(2\pi)}\right]^2+\frac{2s(0.262)}{130(2\pi)}+1\right\}^{-1}$ | 19 | 2.3 |
| 5 | .72 | $\frac{\left(\frac{s}{2\pi}+1\right)\left(\frac{s}{2\pi}+1\right)^{-1}}{\left[\frac{s}{100(2\pi)}+1\right]\left\{\left[\frac{s}{100(2\pi)}\right]^2+\frac{2s}{100(2\pi)}+1\right\}}$ | $\frac{\left[\frac{s}{4.43(2\pi)}+1\right]\left[\frac{s}{3.51(2\pi)}+1\right]^{-1}}{\left[\frac{s}{23.6(2\pi)}+1\right]\left\{\left[\frac{s}{129(2\pi)}\right]^2+\frac{2s(0.296)}{129(2\pi)}+1\right\}}$ | 188 | 24 |
| 6 | .59 | $\frac{\left(\frac{s}{2\pi}+1\right)\left(\frac{s}{2\pi}+1\right)^{-1}\left[\frac{s}{1000(2\pi)}+1\right]^{-1}}{\left[\frac{s}{10(2\pi)}+1\right]\left\{\left[\frac{s}{1000(2\pi)}\right]^2+\frac{2s}{1000(2\pi)}+1\right\}}$ | $\frac{\left[\frac{s}{4.91(2\pi)}+1\right]\left[\frac{s}{3.70(2\pi)}+1\right]^{-1}\left[\frac{s}{309(2\pi)}+1\right]^{-1}}{\left[\frac{s}{27.0(2\pi)}+1\right]\left\{\left[\frac{s}{145(2\pi)}\right]^2+\frac{2s(0.265)}{145(2\pi)}+1\right\}}$ | 93 | 20 |

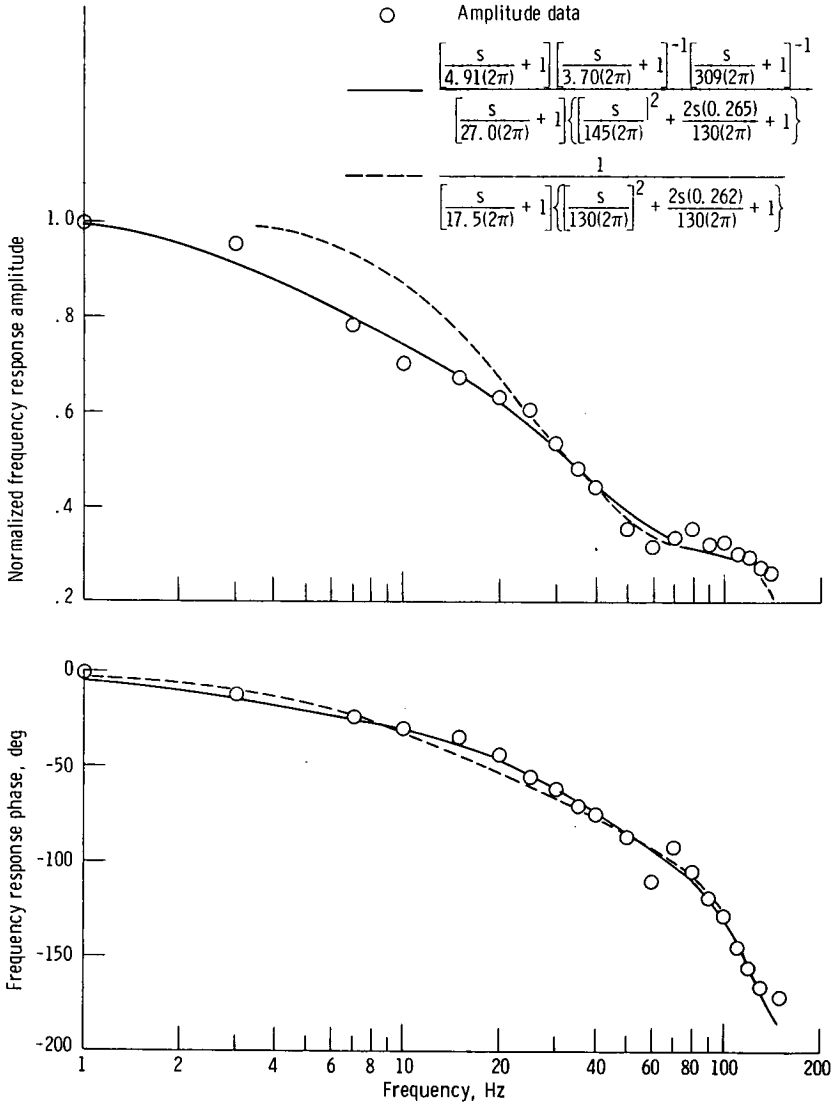


Figure 1. - Frequency response.

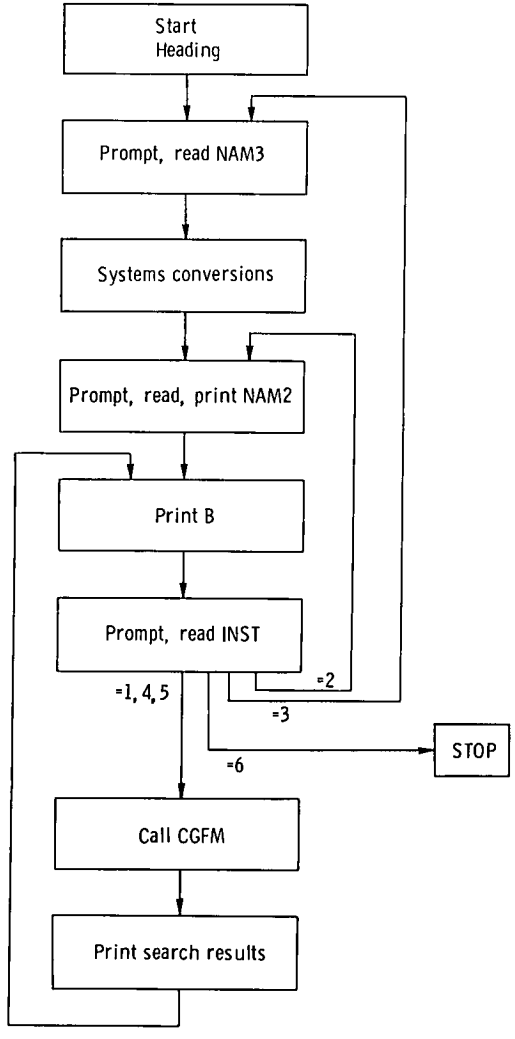


Figure 2. - Flow chart for program MODEL.

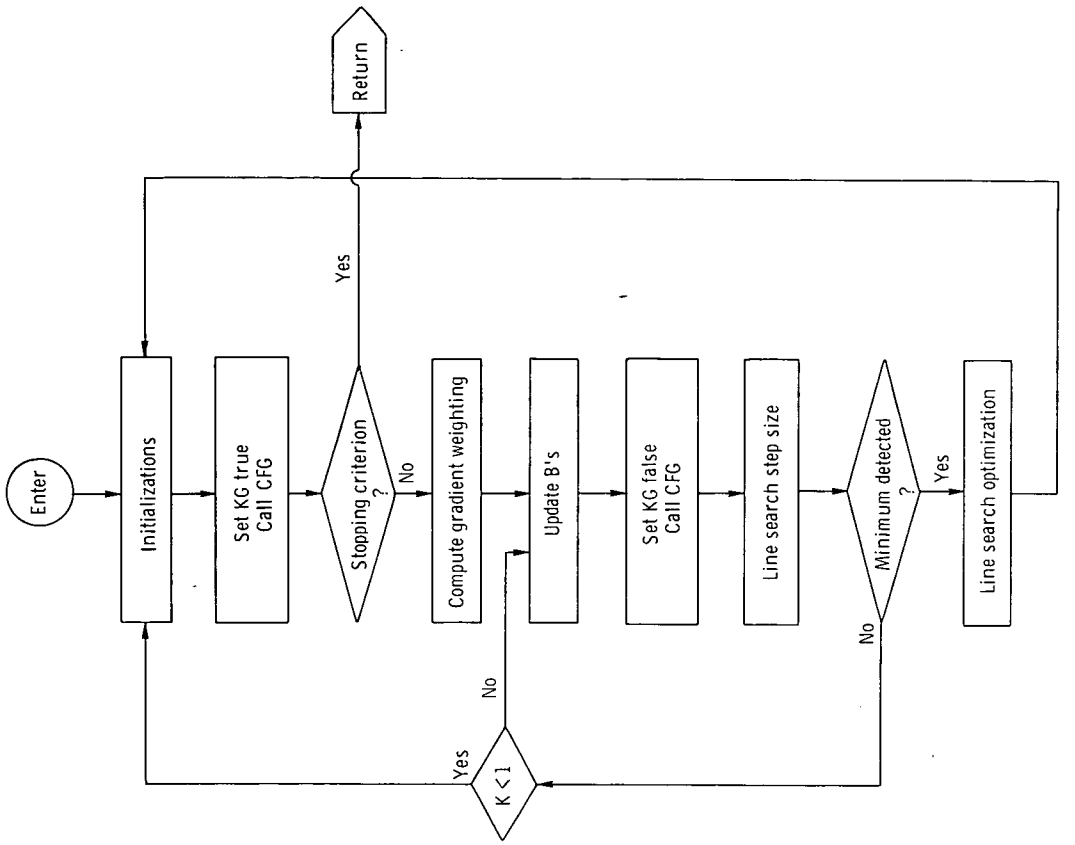


Figure 3. - Flow chart for subroutine CFG.




Figure 4. - Flow chart for subroutine CGFM.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

SPECIAL FOURTH-CLASS RATE
BOOK

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION
451

U.S.MAIL

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

# NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

***Details on the availability of these publications may be obtained from:***

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

**Washington, D.C. 20546**