NASA
Technical
Paper
2311

June 1984

An Algorithm for Maximum Likelihood Estimation Using an Efficient Method for Approximating Sensitivities

Patrick C. Murphy

NASA

An Algorithm for Maximum Likelihood Estimation Using an Efficient Method for Approximating Sensitivities

Patrick C. Murphy

Langley Research Center
Hampton, Virginia

SUMMARY

An algorithm for maximum likelihood (ML) estimation is developed with an efficient method for approximating the sensitivities. The algorithm is applicable to any parameter-estimation problem and is particularly suited for multivariable dynamic systems. The ML algorithm relies on a new optimization method closely related to a modified Newton-Raphson (MNR) technique; the new optimization method is referred to as a modified Newton-Raphson with estimated sensitivities (MNRES).

MNRES determines sensitivities by using slope information from local surface approximations of each output variable in parameter space. The fitted surface allows sensitivity information to be updated at each iteration with a significant reduction in computational effort. MNRES determines the sensitivities with less computational effort than using either a finite-difference method or integrating the analytically determined sensitivity equations as in an MNR procedure. The choice of the type of surface (for example, nth-order polynomial or spline) and the method of fitting the surface (for example, least squares or simply solving simultaneous equations) is made by the user to suit the particular need.

Two surface-fitting methods are discussed and demonstrated, while other possibilities are indicated. Comparisons are made between MNRES and other commonly used optimization methods such as the Nelder-Mead simplex method (a search technique) and the modified Newton-Raphson method (a gradient technique). Several sample problems are solved to compare the techniques. Simple linear systems are used at first, and then nonlinear aircraft estimation problems are solved by using both real and simulated data. MNRES is found to be equally accurate and substantially faster than the commonly used techniques. The reduction in computational effort provided by MNRES is dependent on several factors: the choice of surface-fitting method, the number of unknown parameters, data quality, and accuracy of the sensitivity calculations. MNRES eliminates the need to derive sensitivity equations for each model, thus providing flexibility to use model equations in any format that is convenient and providing a more generally applicable algorithm.

INTRODUCTION

System-identification and parameter-estimation techniques used to determine aircraft stability and control parameters from flight data are well established for the linear flight regimes. In these flight regimes, the aircraft aerodynamic model can be expressed as a linear function of the states and control inputs. In the nonlinear regimes, as in high-angle-of-attack conditions, the aerodynamic model can be a complex function of states and control inputs. Thus, identifying the best mathematical representation (model structure) and estimating model parameters can be very difficult. The need to improve identification and estimation techniques for nonlinear aircraft models has led to further development of the methodology for system identification.

In reference 1 a modified stepwise regression (MSR), along with several testing criteria, is suggested as an efficient method to determine model structure and obtain initial parameter estimates. This was found to be a useful technique for high-angle-of-attack aircraft-identification problems in which nonlinear aerodynamic effects are

present. However, the estimates are asymptotically biased and their standard errors are based on a simplified expression which is valid only for the "classical" linear regression. Therefore, it is beneficial to improve the MSR estimates by using the maximum likelihood (ML) method which has more favorable asymptotic properties (ref. 2). In addition, it is desirable for the ML algorithm to be independent of model structure and sensitivity calculations; this independence provides a more generally applicable algorithm and potentially eliminates some of the computational burden found with integrating sensitivity equations. Also, a very efficient ML estimation algorithm is desirable to reduce the computational effort involved in processing a large number of parameters and candidate models. Reducing computational requirements of the ML method requires careful examination of the optimization methods utilized in the algorithm. Although nonlinear, unconstrained optimization problems have been studied quite extensively (ref. 3), little has been done to improve the optimization techniques as they apply to aircraft estimation problems. The numerical aspects of computing ML estimates for linear dynamic systems in state-vector form was studied in reference 4.

Aircraft estimation problems, which belong to a class of problems involving dynamic systems, require substantial computational effort at each step of the optimization process. At each step the parameters are updated and the equations of motion are integrated to obtain time histories of each response variable. Many ML estimation procedures for aircraft problems use a Newton-Raphson optimization method (ref. 5) that requires solving (integrating) sensitivity equations. This accounts for most of the computational effort since an m state system with n unknown parameters requires mn sensitivity equations plus m state equations to be integrated each iteration. Several states and 30 or 40 parameters are not unrealistic for aircraft estimation problems. This is for only one flight condition; if a model is desired throughout the entire flight envelope, the computational requirements become overwhelming since analysis of various flight conditions may require more than one candidate model.

The objective of this report is to provide an ML method which does not require the analytical form of the sensitivity equations to formulate the algorithm and which provides a method with reduced computational requirements compared with the commonly used algorithms. The report first considers both gradient and search optimization methods: the Nelder-Mead simplex method (SM) (a search technique (ref. 6)) and the commonly used modified Newton-Raphson method (MNR) (a gradient technique). After a comparison of these methods, the MNR is selected as a method for further consideration. As a main objective of the report, an efficient algorithm is subsequently proposed. This algorithm will be referred to as a modified Newton-Raphson with estimated sensitivities (MNRES) method. It determines sensitivities without the analytical form of the sensitivity equations and does it more efficiently than a finite-difference method. Therefore, it is easily used with any model structure determined, for example, by the MSR method. Properties of the improved method are discussed and several test cases are provided. Finally, the MNRES algorithm is applied to a nonlinear aircraft estimation problem by using both real and simulated data. For this study the estimation problem is simplified by assuming that no process noise exists in the system dynamics.

A   system matrix for state equation

a_j jth element of   A   matrix

a_k incoming-parameter vector for MNRES algorithm

a_k^o outgoing-parameter vector for MNRES algorithm

a_y acceleration in y-direction, g units (where   1g = 9.81 m/sec^2)

B   control-distribution matrix for state equation

b   wing span, m

b_j jth element of   B   matrix

C   state-distribution matrix for output equation

C_l rolling-moment coefficient, M_X/qSb

$C_{l,0}, C_{n,0}, C_{Y,0}$ nondimensional aerodynamic force and moments for trimmed flight

C_n yawing-moment coefficient, M_Z/qSb

C_Y lateral-force coefficient, F_Y/qS

D   control-distribution matrix for output equation

E{ }   expectation operator

F_Y force along lateral body axes, N

G_i sensitivity matrix at ith data point

J   cost function

M   Fisher information matrix

M_X rolling moment, N-m

M_Z yawing moment, N-m

m   number of states

N   number of data points

n   number of parameters

n_a number of elements in   A   matrix

n_b number of elements in   B   matrix

P least-squares-parameter covariance matrix divided by σ

p roll rate, rad/sec

q dynamic pressure, $\frac{1}{2} \rho V^2$, Pa

R measurement-noise covariance matrix

r yaw rate, rad/sec

S wing area, m^2

s_k vector of sensitivities for kth element of   Y

$s_{k\ell}$ sensitivity of kth element of   Y   to ℓth element of θ

t time, sec

Δt processing time for Control Data Corporation CYBER 175 digital computer, sec

U input vector

V airspeed, m/sec

v_i vector of measurement noise at ith data point

X vector of states or matrix of parameters in MNRES

Y vector of outputs

y scalar output

Z_i vector of measured outputs at ith data point

z_j jth element of Z_i

α angle of attack, rad

β sideslip angle, rad

Δ incremental value

δ_a aileron deflection, rad

δ_{ij} Kronecker delta

δ_r rudder deflection, rad

θ vector of unknown parameters

θ^* vector of optimal parameters

4

v_i vector of residuals at ith data point

ρ air density, kg/m^3

σ standard error

ϕ roll angle, rad

Subscripts:

a   partial derivative with respect to indicated quantity

H,L   highest and lowest cost, respectively

i,j,k,ℓ general indices

Superscript:

j   index of surface-fitting points

Abbreviations:

CPU   central processing unit

MAX   ML program using MNRES algorithm

MAXLIK   ML program using MNR algorithm

ML   maximum likelihood

MNR   modified Newton-Raphson

MNRES   modified Newton-Raphson with estimated sensitivities

MSR   modified stepwise regression

NR   Newton-Raphson

SM   simplex method

Matrix exponents:

T   transpose matrix

-1   inverse matrix

Mathematical notation:

^   estimated quantity when over symbol

.   derivative with respect to time when over symbol

∇ gradient operator

The following aerodynamic derivatives are referenced to a system of body axes with the origin at the aircraft center of gravity:

$$C_{Y_\beta} = \frac{\partial C_Y}{\partial \beta} \qquad C_{Y_p} = \frac{\partial C_Y}{\partial \frac{pb}{2V}} \qquad C_{Y_r} = \frac{\partial C_Y}{\partial \frac{rb}{2V}}$$

$$C_{Y_{\delta a}} = \frac{\partial C_Y}{\partial \delta_a} \qquad C_{Y_{\delta r}} = \frac{\partial C_Y}{\partial \delta_r} \qquad C_{Y_{\alpha\beta}} = \frac{\partial^2 C_Y}{\partial \alpha\ \partial \beta}$$

$$C_{l_\beta} = \frac{\partial C_l}{\partial \beta} \qquad C_{l_p} = \frac{\partial C_l}{\partial \frac{pb}{2V}} \qquad C_{l_r} = \frac{\partial C_l}{\partial \frac{rb}{2V}}$$

$$C_{l_{\delta a}} = \frac{\partial C_l}{\partial \delta_a} \qquad C_{l_{\delta r}} = \frac{\partial C_l}{\partial \delta_r} \qquad C_{l_{\alpha\beta}} = \frac{\partial^2 C_l}{\partial \alpha\ \partial \beta}$$

$$C_{l_{\alpha p}} = \frac{\partial^2 C_l}{\partial \alpha\ \partial p} \qquad C_{n_\beta} = \frac{\partial C_n}{\partial \beta} \qquad C_{n_p} = \frac{\partial C_n}{\partial \frac{pb}{2V}}$$

$$C_{n_r} = \frac{\partial C_n}{\partial \frac{rb}{2V}} \qquad C_{n_{\delta a}} = \frac{\partial C_n}{\partial \delta_a} \qquad C_{n_{\delta r}} = \frac{\partial C_n}{\partial \delta_r}$$

$$C_{Y_{\beta 3}} = \frac{1}{6} \frac{\partial^3 C_Y}{\partial \beta^3} \qquad C_{n_{\beta 3}} = \frac{1}{6} \frac{\partial^3 C_n}{\partial \beta^3}$$

COMPARISON OF SEARCH AND GRADIENT OPTIMIZATION METHODS

The twofold objective of this report is to develop an optimization method for use in the ML estimation which (1) eliminates the need to solve sensitivity equations and (2) improves, if possible, the speed of the overall estimation algorithm. This should be accomplished while considering practical examples of nonlinear, multivariable dynamic systems such as aircraft systems, which are of primary interest for this research. As a result of the twofold objective for this report, two well-established optimization methods were chosen for consideration.

Two optimization methods (the Nelder-Mead simplex method (SM), also called the flexible polyhedron method, and the modified Newton-Raphson (MNR) method) will be described in this section. More details are provided in references 3 and 6 for the SM and in references 2 and 5 for the MNR. A variation of the MNR will be used in which the derivative information (sensitivities) is computed by using finite differences (refs. 7 and 8). Both forms of the MNR are used in this report.

The Nelder-Mead algorithm was chosen for this study because search methods initially appear to be good candidates for reducing computational demands in aircraft estimation problems. These methods avoid derivative calculations, which is where most of the computation time is spent in the quasi-Newton methods. The search methods are independent of model form and thus are readily applicable to any aerodynamic model. The SM is used at the Langley Research Center and has been found to be advantageous in some aircraft design and control applications (ref. 9).

This report is primarily concerned with nonlinear aircraft estimation problems. Since the MNR approach is commonly used for these problems, it is included as a benchmark. Although it is computationally burdensome to estimate derivatives, this information enables relatively fast convergence of the optimization methods. In fact, Newton's method converges in one pass for cost functions which are quadratic. Hence, Newton-Raphson gradient techniques used for estimation problems of dynamic systems are expected to converge faster when the quadratic approximations for the cost functions are good. Also Newton's method and the quasi-Newton methods provide both step size and direction for each iteration independent of relative parameter scaling. In some problems, however, additional control of step size is needed to ensure convergence. Since removing the requirement to solve sensitivity equations has been proposed, the MNR algorithm in this report will use a simple finite-difference method .except when otherwise noted. This does not turn out to be too costly in terms of computational time (refs. 7 and 8). However, care must be taken to obtain the derivatives as accurately as possible, especially in large problems.

Description of Two Candidate Optimization Methods

The SM represents a class of methods, called search methods, which do not use derivative information to minimize a scalar function of n variables $J(\theta)$. The method uses a flexible polyhedron surface with $n + 1$ vertices where each vertex is defined by a vector θ. The vertex θ_H, producing the highest value of $J(\theta)$, is projected through the centroid of the remaining vertices to define a new vertex. This new vertex, and the remaining ones without θ_H, form a new polyhedron. This operation is called a "reflection." If the new vertex produces a lower cost than θ_L (the vertex producing the smallest $J(\theta)$), then an expansion takes place and a new vertex is located farther out along the same projection. Similarly, if higher costs are found, a contraction takes place. The minimum of the cost function is found by repeatedly deleting the point having the highest value of $J(\theta)$ and adding

new projected points that produce lower $J(\theta)$. The flexible polyhedron is able to adapt to the shape of $J(\theta)$ by stretching down slopes, contracting near minima, and changing direction in curved valleys.

The modified Newton-Raphson optimization method belongs to a class of methods known as variable-metric or quasi-Newton methods. This class has various techniques for approximating the Hessian matrix of the cost function (matrix of second partials with respect to parameters), but they all use first-derivative information for the approximation. The MNR method will be illustrated in an ML estimation problem for a linear system without process noise. The theory behind ML will not be discussed; see reference 2 for details. It is assumed that only the measured outputs are corrupted by noise and that the noise is an uncorrelated, Gaussian white-noise source with zero mean. The problem is to minimize the errors between the computed model outputs and the actual measured outputs. The optimization problem is nonlinear and MNR is usually applied because of its good convergence characteristics even for large numbers of unknown parameters. The system equations are as follows:

State equation:

$$\dot{X} = AX + BU \qquad\qquad X(0) = X_0 \tag{1}$$

Output equation:

$$Y = CX + DU \tag{2}$$

Measurement equation:

$$Z_i = Y_i + v_i \qquad\qquad (i = 1, 2, \ldots, N) \tag{3}$$

where it is assumed that

$$E\{v_i\} = 0 \qquad\qquad E\{v_i(v_j)^T\} = R\delta_{ij} \tag{4}$$

The symbol R represents a diagonal measurement-noise covariance matrix. Without process noise, the cost function to be minimized is

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} \left(z_i - \hat{Y}_i\right)^T R^{-1}\left(z_i - \hat{Y}_i\right) + \frac{N}{2}\ln\left|R^{-1}\right| \tag{5}$$

The unknown R can be determined by minimization of the cost with respect to R. This minimization produces (see ref. 2)

$$\hat{R} = \frac{1}{N}\sum_{i=1}^{N} v_i(v_i)^T \tag{6}$$

8

The cost can then be written as

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{N} \left(Z_i - \hat{Y}_i \right)^T \hat{R}^{-1} \left(Z_i - \hat{Y}_i \right) + \text{Constant} \tag{7}$$

This is the same cost function used in the output error technique from reference 2, except that the covariance of the measurement noise is used as a weighting matrix. The MNR method accomplishes the minimization by expanding Y (the computed output vector) about θ_0 (the initial unknown parameter vector). A Taylor series expansion truncated to first order is computed as

$$\hat{Y}(\theta) = \hat{Y}(\theta_0) + \left. \frac{\partial \hat{Y}}{\partial \theta} \right|_{\theta_0} \Delta\theta \tag{8}$$

where $\Delta\theta = \theta - \theta_0$. Then, by substituting equation (8) into (7), a quadratic approximation of $J(\theta)$ is obtained. The increment $\Delta\theta$ is the unknown. Differentiating J with respect to θ and equating the derivative to 0 to find the minima results in

$$\frac{\partial J}{\partial \theta} = -\sum_{i=1}^{N} (G_i)^T \hat{R}^{-1} v_i + \sum_{i=1}^{N} (G_i)^T \hat{R}^{-1} G_i \, \Delta\theta = 0 \tag{9}$$

where

$$G_i = \left\{ \frac{\partial y_k}{\partial \theta_j} \right\}_i \qquad v_i = Z_i - \hat{Y}_i(\theta_0) \tag{10}$$

and, thus,

$$\Delta\hat{\theta} = \left[\sum_{i=1}^{N} (G_i)^T \hat{R}^{-1} G_i \right]^{-1} \sum_{i=1}^{N} (G_i)^T \hat{R}^{-1} v_i \tag{11}$$

This is often written as

$$\Delta\hat{\theta} = -M^{-1} \left. \frac{\partial J}{\partial \theta} \right|_{\theta_0} \tag{12}$$

emphasizing the Fisher information matrix M and gradient terms. So, for each kth iteration the new estimate $\hat{\theta}_{k+1}$ is given as $\hat{\theta}_{k+1} = \hat{\theta}_k + \Delta\hat{\theta}_{k+1}$. Convergence is

achieved when $\Delta J/J$ and $\Delta\theta/\theta$ are small enough. The sensitivities G_i are determined separately from the aforementioned steps. This can be done numerically with a simple finite-difference method or by integrating the sensitivity equations. The sensitivity equations for the system defined in equations (1) and (2) are

$$\dot{X}_{a_j} = AX_{a_j} + \frac{\partial A}{\partial a_j} X \qquad (j = 1, 2, \ldots, n_a) \qquad (13)$$

$$\dot{X}_{b_j} = AX_{b_j} + \frac{\partial B}{\partial b_j} U \qquad (j = 1, 2, \ldots, n_b) \qquad (14)$$

where each element of A and B is an unknown parameter. Details of the sensitivity-equation approach are discussed in reference 2 and the finite-difference approach is discussed in references 7 and 8.

Performance of Methods on Estimation Problem I

Estimation problem I demonstrates and compares the Nelder-Mead and modified Newton-Raphson optimization methods in a simple ML estimation problem. The MNR method uses a finite-difference technique to compute derivatives which satisfies the first objective of this research, eliminating the need to compute analytical gradients. The MNR method using numerically determined derivatives generally performs with about the same speed as a method using analytically determined derivatives, that is, integrating sensitivity equations (refs. 7 and 8). The purpose of this example is to compare the relative performance of the two approaches on a problem which is characteristic of dynamic systems, such as aircraft estimation problems. The important characteristics of the aircraft estimation problem are the requirement to integrate equations of motion to evaluate the cost function and the requirement to have reasonable initial values for the estimated parameters. These initial values might be given by a least-squares procedure.

The first test problem, referred to as estimation problem I, is a single-input/double-output, linear second-order system with six unknown parameters. The system equations used to generate simulated data are given by equations (1) and (2). The six unknown parameters are the four elements of the 2 × 2 system matrix A and the two elements of the control-input matrix B found in equation (1). Matrices A and B are defined in table I as true values of θ; matrices C and D of equation (2) are the unity and zero matrix, respectively. The data were generated by using a simple Euler integration technique, and the input was selected as U = sin(t). Initial conditions were set to 0. Process noise and measurement noise are excluded for problem I.

For aircraft estimation problems, the bulk of computer time is spent in performing integrations. To prevent any bias in the results due to variations in programming efficiency or integration techniques, the estimation algorithms use the same integration subroutine. For estimation problems I and II a simple Euler integration technique is used and the integration subroutine is specifically designed to integrate matrix second-order linear time-invariant systems. In order to accommodate

TABLE I.- ESTIMATION PROBLEM I

Unknown parameter, θ	True value of θ	Initial value of θ	Final estimated values using method -		
			SM	MNR	MNRES
θ_1	0	0.01	-0.12 E-03	0.89 E-07	0.73 E-06
θ_2	-1.5	-1.6	-1.5	-1.5	-1.5
θ_3	1.0	1.1	1.0	1.0	1.0
θ_4	-.5	-.6	-.5	-.5	-.5
θ_5	.2	.25	.2	.2	.2
θ_6	.1	.15	.1	.1	.1
Cost			0.14 E-08	0.61 E-10	0.11 E-07
Equivalent evaluation ...			715	28	12
Δt, sec			2948	106	47

the MNR algorithm, which requires a fourteenth-order system to be integrated each pass, the system

$$\begin{bmatrix} \dot{X} \\ \dot{X}_{a_j} \\ \dot{X}_{b_j} \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ \partial A/\partial a_j & A & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ X_{a_j} \\ X_{b_j} \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ \partial B/\partial b_j \end{bmatrix} U \qquad (15)$$

is treated as seven, second-order systems: one second-order system for the states (given by eq. (1)), four second-order systems for the sensitivities with respect to A matrix elements (given by eq. (13) with j = 1 through 4) and two-second order systems for the sensitivities with respect to B matrix elements (given by eq. (14) with j = 1 and 2). The results of the first-system integration X(t) are used as the input function for the X_a system. Because the systems are not fully coupled, it is more efficient and just as accurate to integrate them separately (as described) as it is to integrate the complete fourteenth-order system.

The performance of the methods used in this report will be evaluated with the following criteria:

(1) Accuracy of estimates

(2) CPU time to termination

(3) Number of cost-function evaluations

The primary evaluation criteria will be accuracy of estimates and CPU time to termination. Termination is obtained when parameter and cost-function fractional changes are computed to be within a specified precision. Both cost-function change $\Delta J/J$ and parameter change $\Delta\theta/\theta$ are required to be satisfied simultaneously to prevent

premature termination on a plateau where $\Delta J \ll 1$ and $\Delta \theta$ is relatively large, or on a steep slope where $\Delta \theta \ll 1$ and ΔJ is relatively large.

A secondary criterion is the number of cost-function evaluations, which is used only as an indirect guide since each method advances differently with each cost determination. However, the number of cost-function evaluations is important since each method requires the system equations to be integrated at least once for each cost evaluation, with one cost evaluation representing a significant computational effort. To emphasize that the same measure of computational effort is used to compare the different methods, the term "equivalent evaluations" is used. One equivalent evaluation represents the computational effort required to integrate the system equations once. Each method described in this report requires a different number of equivalent evaluations to make one update in the parameter estimates.

Only dynamic systems or aircraft system estimation problems are used in this study, rather than classical test problems such as Rosenbrock's function (ref. 3). Using classical optimization problems, which usually require very little computational time to evaluate the cost function, could lead to different conclusions about the algorithms. The problems in this study range from a simulated linear system with 6 unknown parameters and noise-free measurements (problem I) to a real aircraft problem modeled by a nonlinear system with 20 unknown parameters and measurements corrupted by noise (problem IV).

Figure 1 shows the input and response time histories for problem I, and table I gives the numerical results. This table shows the true value of the six unknown
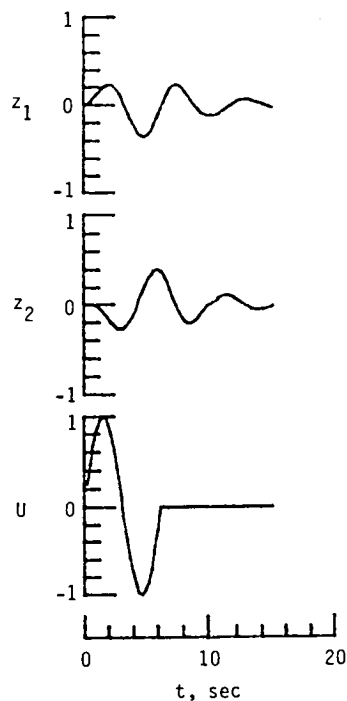

Figure 1.- Time history of input and response variables for problem I.

parameters, their initial values, and the final estimated values. Each method was able to estimate the parameters correctly and satisfy the convergence criterion. The MNR method performed about 30 times faster than the SM. This is apparent when considering the number of equivalent evaluations required by each method, 715 for SM and 28 for MNR.

Table I shows clearly that optimization problems having reasonable starting values and involving time-consuming cost-function evaluations should not be solved with direct search methods, such as SM. Reasonable initial values tend to provide a more quadraticlike cost function for which Newton's method is most effective. If reasonable initial values are not available, the SM may be more attractive. In response to the results of problem I, this study concentrated on improving the MNR method. The next section describes the optimization method developed from MNR.

MODIFIED NEWTON-RAPHSON METHOD WITH ESTIMATED SENSITIVITIES (MNRES)

The MNRES method developed in this paper is essentially an MNR optimization algorithm with an efficient method for estimating sensitivities. The sensitivities are determined with less computational effort than by using either a finite-difference method or analytically determined sensitivity equations. As in the ML/MNR algorithm previously described, the same equations (eqs. (1) through (12)) apply for ML/MNRES; however, the sensitivities G_i are computed by using slope information from local surface approximations of $Y(\theta)$. The fitted surface allows the slope or sensitivity information to be updated at each iteration with a significant reduction in computational effort.

In reference 10, a nonlinear least-squares algorithm is presented which uses a surface-fitting method. This algorithm uses a linear-surface approximation of a scalar-response variable to eliminate derivative calculations altogether. In MNRES, the surface approximation is treated differently by developing an algorithm which retains derivative information in a Newton-Raphson method for multivariable systems. This is done to provide directional information for the convergence process and to provide covariance information.

The sensitivities for MNRES are determined by assuming that they are approximately equal to the slopes of a surface which has been fit to $Y(\theta)$. The surface approximates $Y(\theta)$ near the series expansion point of equation (8). The choice of the type of surface (e.g., nth-order polynomial or spline) and the method of fitting the surface (e.g., least squares or simply solving n simultaneous equations for n unknowns) is made by the user to suit the particular need. The trade-off in choosing a method involves the choice between accuracy of the sensitivities and computational effort. The simplest and computationally least demanding approach is to expand $Y(\theta)$ in an n-term first-degree polynomial in θ and then solve for the coefficients (sensitivities) by solving the n simultaneous equations. Although this approach produces the least accurate sensitivities, this, as well as any other approach, can have the improved accuracy through step-size control, if necessary. Theoretically, in the limit as step size becomes very small, the estimates from the surface fits would equal the estimates from a finite-difference method, which in the limit equals the actual slope. The trade-off for step-size control involves the speed of convergence versus the accuracy of the sensitivities and robustness of the algorithm to noisy data.

The MNRES algorithm is best described by first looking at the computationally least demanding approach of using a linear-surface approximation. Expanding $Y(\theta)$ in a first-degree polynomial in θ for each point in time and at $n + 1$ different points in the n parameter space gives

$$y_{ki}^{j}(\theta^{j}) = s_{k0} + s_{k1}\theta_{1}^{j} + \cdots + s_{kn}\theta_{n}^{j} \tag{16}$$

where i indicates the ith point in time; k indicates the kth element of the output vector $Y(\theta)$; and j indicates one of the $n + 1$ sample points used to fit equation (16) to $Y(\theta)$. Note that $y^{j}(\theta^{j}) = Y(\theta)$ at each of the $n + 1$ points. The sample points are chosen by allowing a small perturbation of each parameter. Alternatively, the perturbation size can be selected to reflect the relative significance of each parameter to the model. This allows for larger perturbations of the less sensitive parameters and smaller perturbations for the very sensitive parameters, thus providing higher quality derivative calculations. This alternative is discussed further in the next section. The slopes s_{k1} to s_{kn} are the desired sensitivities $\{\partial y_{k}/\partial\theta_{j}\}_{i}$, and s_{k0} is the point on $Y(\theta)$ where the sensitivities are desired; it is the series expansion point of equation (8) for the MNR optimization method. Note that because this is a linear surface, the slopes are constant over the surface and need not be evaluated specifically at s_{k0}. If a higher degree polynomial is fit to $Y(\theta)$, the slopes will vary across the fitted surface and, therefore, must be evaluated specifically at s_{k0}. Consider the matrix representation of equation (16) for the first element of Y and for the $n + 1$ sample points:

$$Y_{1i} = XS_{1i} \tag{17}$$

$$\begin{bmatrix} y_{1i}^{0} \\ y_{1i}^{1} \\ \cdot \\ \cdot \\ \cdot \\ y_{1i}^{n} \end{bmatrix} = \begin{bmatrix} 1 & \theta_{1}^{0} & \theta_{2}^{0} & \cdots & \theta_{n}^{0} \\ 1 & \theta_{1}^{1} & \theta_{2}^{1} & \cdots & \theta_{n}^{1} \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ 1 & \theta_{1}^{n} & \theta_{2}^{n} & \cdots & \theta_{n}^{n} \end{bmatrix} \begin{bmatrix} s_{10} \\ s_{11} \\ \cdot \\ \cdot \\ \cdot \\ s_{1n} \end{bmatrix}_{i} \tag{18}$$

Since s_{10} is a known point, equation (18) can be simplified. The first line in equation (18) can be eliminated by subtracting from the other n equations. Thus,

$$\Delta Y_{1i} = \Delta X\, S_{1i} \tag{19}$$

$$\begin{bmatrix} y_{1i}^1 - y_{1i}^0 \\ \\ y_{1i}^2 - y_{1i}^0 \\ \cdot \\ \cdot \\ \cdot \\ y_{1i}^n - y_{1i}^0 \end{bmatrix} = \begin{bmatrix} \Delta\theta_1^1 & \Delta\theta_2^1 & \cdots & \Delta\theta_n^1 \\ \Delta\theta_1^2 & \Delta\theta_2^2 & \cdots & \Delta\theta_n^1 \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \Delta\theta_1^n & \Delta\theta_2^n & \cdots & \Delta\theta_n^n \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \\ \cdot \\ \cdot \\ \cdot \\ s_{1n} \end{bmatrix} \tag{20}$$

where

$$\Delta\theta_k^j = \theta_k^j - \theta_k^0$$

Thus, at time i the sensitivities for the first element in Y are given by

$$S_{1i} = [\Delta X]^{-1} \Delta Y_{1i} \tag{21}$$

Note that the ΔX matrix is independent of time. This enables the sensitivities to be calculated rapidly during each iteration of the algorithm. This is a key factor in reducing the computational effort of the algorithm; in effect, the integration of the mn sensitivity equations has been replaced by a set of m matrix multiplications.

Figure 2 shows, geometrically, two iterations for the case in which θ is of dimension two and a linear surface is used to fit a scalar y. The expansion is simply at time i:

$$\begin{bmatrix} y_i^1 - y_i^0 \\ \\ y_i^2 - y_i^0 \end{bmatrix} = \begin{bmatrix} \Delta\theta_1^1 & \Delta\theta_2^1 \\ \\ \Delta\theta_1^2 & \Delta\theta_2^2 \end{bmatrix} \begin{bmatrix} s_1 \\ \\ s_2 \end{bmatrix}_i \tag{22}$$

During the first iteration, this espansion requires that $y(\theta)$ be evaluated at $n + 1 = 3$ points: y^0, y^1, y^2. Computationally, the first iteration is the most costly phase of the MNRES algorithm. Each evaluation of Y requires that the equations of motion be integrated. The linear surface (indicated by the solid-line triangle in fig. 2) is fit and the slopes (sensitivities) are thereby determined. The algorithm proceeds as in the ordinary MNR method to obtain

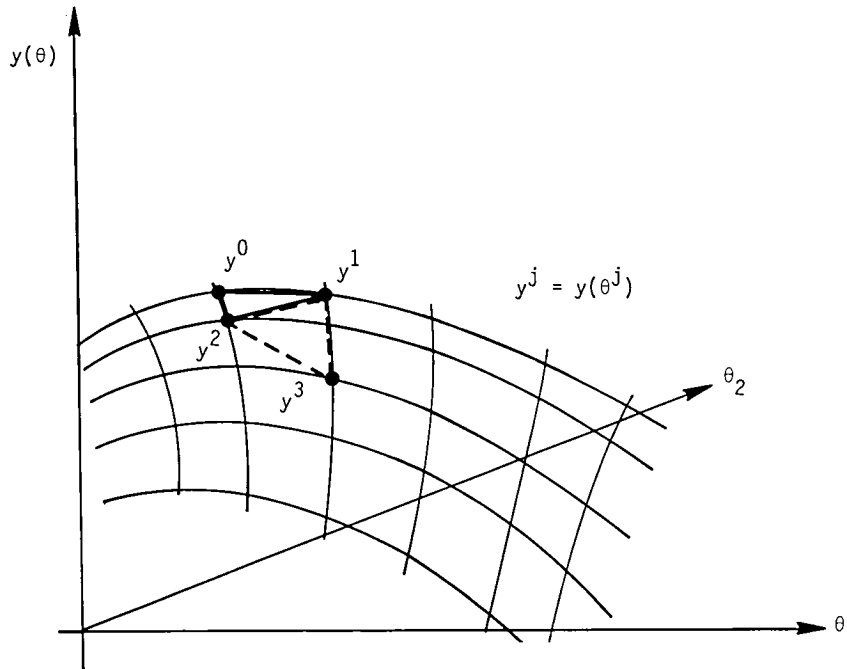$$\hat{\theta}_{k+1} = \hat{\theta}_k + \Delta\hat{\theta}_{k+1}$$

Figure 2.- Linear-surface fit for two iterations of MNRES.

The new Y is evaluated (through integration) at θ_{k+1} to get $y^3(\theta)$. At this point the MNRES algorithm has reduced the sensitivity problem to solving a set of simultaneous equations. This is done by eliminating the θ^j in X, which produced the greatest cost in $J(\theta)$, and replacing that information with the newest estimate of θ. The new surface (indicated by the dotted-line triangle) in figure 2 assumes that y^0 is the high-cost point and thus eliminates it from the fitted surface. The slopes of the new surface provide the sensitivities for the MNRES algorithm to proceed. In this method a check should be made to ensure that the new $Y^j(\theta_{k+1})$ produces a smaller value of $J(\theta)$. In some cases, step-size control or complete restarting may be needed. Note that initialization of the algorithm requires that $n + 1$ integrations be performed for the $n + 1$ trajectories, Y^j. After the start-up, only one integration of the system equations is needed to evaluate the cost $J(\theta)$, outputs $Y(\theta)$, and updated parameter estimates for each iteration.

The least-squares approach to fitting the surface $Y(\theta)$ offers another advantage if an iterative least-squares method is used. The iterative method provides a memory device reducing the storage requirements from $n + 1$ sets of output time histories to just two time histories. One of the two corresponds to the new response predicted by the most recent estimate of θ, and the other corresponds to the outgoing θ that produced the highest cost. The penalty for this advantage is the need to integrate equations of motion twice per iteration; this result still requires substantially less computational effort than that required with the usual MNR method. This algorithm was used in estimation problems I and II and is described subsequently and also in the appendix. It should be noted that because problems I and II did not require a large amount of storage space, all time histories were saved so that only one integration per pass was performed.

When using the iterative least-squares approach, only two changes are made to the MNRES algorithm just described. The first change is in the calculation of the ΔX matrix, and the second change is in the sensitivity calculation. The development

of this formulation will begin with equation (16) in condensed form, which is shown in the following equation. Everything discussed up to this equation in the previous development applies here. Thus,

$$y_{ki}^j = x^j s_{ki}$$ (23)

Simplifying the notation by dropping the ki subscript and writing the matrix form of the equation (which removes the j superscript) gives

$$Y = XS$$ (24)

The least-squares solution for the sensitivity vector is

$$S = [X^T X]^{-1} X^T Y$$ (25)

Now, defining a recursive relation for the $k + 1$ iteration gives

$$P_{k+1} = \left[(X_k)^T X_k \right]^{-1}$$ (26)

and the updating equation is

$$P_{k+1} = \left[P_k^{-1} + (a_k)^T a_k - \left(a_k^o \right)^T a_k^o \right]^{-1}$$ (27)

where a_k is the new set of θ to be included and a_k^o is the outgoing set of θ which produced the high cost in J. The recursive equation for S is

$$S_{k+1} = S_k - P_{k+1} \left[\left(a_k^o \right)^T y_k^o - (a_k)^T y_k + \left[(a_k)^T a_k - \left(a_k^o \right)^T a_k^o \right] S_k \right]$$ (28)

With the new sensitivities determined, the algorithm proceeds as before. The recursive relations are derived in the appendix.

PROPERTIES OF MNRES OPTIMIZATION METHOD

In this section, properties of MNRES which aid convergence are discussed heuristically, and factors which have a significant effect on speed are indicated. Also, similarities and differences with the commonly used MNR method are discussed providing a benchmark for comparison. Finally, to demonstrate the performance of MNRES, two estimation problems with varying noise levels are solved by using MNRES and MNR to obtain ML parameter estimates.

17

The ML parameter estimates are obtained by solving an unconstrained, nonlinear optimization problem; that is, find θ^* which minimizes the cost function $J(\theta)$. The necessary and sufficient conditions for this problem are as follows:

(1) $J(\theta)$ is differentiable at θ^*.

(2) $\nabla J(\theta^*) = 0$.

(3) $\nabla^2 J(\theta^*) > 0$.

The theory for solving the unconstrained optimization problem is often based on the assumption that the cost function $J(\theta)$ is a quadratic function of θ. This provides a much more tractable theory and allows basic properties to be readily established. Corresponding theorems for general nonlinear functions are very difficult to prove. However, techniques developed using the quadratic assumption are still very effective for nonlinear functions. Many techniques for solving nonlinear minimization problems are developed from practical experience.

Comparison of MNR and MNRES Properties

Convergence of NR or MNR algorithms, both with and without finite-difference derivatives, has been well documented (ref. 3). Convergence of MNRES can be shown, at least heuristically, by considering several details. First, the MNRES method is still fundamentally an NR method or, for this study, an MNR method. The only critical difference is that the derivatives are approximate which makes MNRES closer to MNR with numerically determined derivatives. Second, note that fitting a first-degree, n-term polynomial such as

$$y = s_0 + s_1\theta_1 + \ldots + s_n\theta_n \tag{29}$$

to $n + 1$ data points is equivalent to a simple finite-difference method. In effect, as $\Delta\theta^j$ (the distance between points on the fitted surface for MNRES) becomes small enough, the sensitivities become identical to that given by a simple finite-difference method, regardless of the actual functional representation of $Y(\theta)$. The MNRES algorithm simply relaxes the accuracy of the sensitivities in order to reduce substantially the integration requirements; the degree of relaxation varies during the optimization process but can be controlled by limiting step size.

The relaxation of sensitivity accuracy generally appears to be a very beneficial trade-off for Newton-Raphson algorithms. During an MNRES optimization there are two times that the sensitivities are very close to the finite-difference values; these times are during the initialization or start-up of the algorithm and toward the end when $\Delta\theta$ becomes small. During initialization of the algorithm, n different θ are chosen (a perturbation on each element of θ is sufficient) and the actual surface given by $Y(\theta)$ is fitted. The user can choose an initial θ^j such that $\left|(\theta^j - \theta^0)/\theta^0\right| << 1$ for each j. In this study, the same $\Delta\theta^j$ was used in the MNR with finite-difference derivatives as that used in the start-up of the MNRES. This was done for comparison purposes; in practice, the choice of perturbation size for θ may be very different, as discussed later.

18

The computational advantage of MNRES over MNR is determined to a large extent by the number of unknown parameters n. During the first iteration, both MNRES and MNR require n + 1 equivalent evaluations. After the first iteration, MNR continues to use n + 1 equivalent evaluations per iteration but MNRES needs only one per iteration. As the estimation process advances, MNRES continually eliminates values of θ which are far from θ^*, based upon the information in one equivalent evaluation. Thus, at the end, θ is very close to θ^*, producing very small $\Delta\theta^j$ for the derivative estimations. Again, the derivatives obtained by the surface-fitting method approach the values obtained in a finite-difference method.

Two factors aid in preventing divergence during the critical time period, when MNRES is between initialization and the end of the optimization, in which potentially large $\Delta\theta$ can occur. First, the user can always incorporate step-size control logic. Carried to the extreme, MNRES could always be forced to approximate the derivatives the same as a finite-difference method. Of course, convergence would be very slow because of the very small steps. In practice, one would let the algorithm take steps determined by the NR logic (as done in this study); and then if a convergence problem develops, one would begin controlling step size. Secondly, NR, MNR, and MNRES methods advance more quickly as the quadratic approximation of the cost function improves; moreover, the Newton algorithm converges in one step for a quadratic cost function. Since the quadratic approximation of the cost function improves the closer that θ gets to θ^*, and since initial estimates of θ^* are often given by a least-squares procedure or knowledgeable user, θ_0 tends to be close to θ^*. Thus, MNRES starts in a region conducive for convergence and, therefore, the method would not be required to handle very large $\Delta\theta$ for many practical cases.

As mentioned previously, in practice it is beneficial to choose the perturbation size in a different fashion from that used in a simple finite-difference method. Simply using a 1-percent perturbation on each element of θ to obtain the corresponding perturbation in each element of $Y(\theta)$ is not optimum for derivative calculations. Experience has shown that it is beneficial to use perturbation sizes which reflect the importance of the parameter to the model. By computing the sensitivities as $\theta_j^2 M_{jj}$ for each parameter and then letting the perturbation sizes be scaled inversely proportional to the normalized ratios of sensitivities, more accurate derivative information can be obtained. Of course, this applies only when an initialization or "restart" is needed. The fundamental issue is that the less sensitive a parameter, the larger the perturbation necessary to obtain an appropriate size response in the outputs. This approach could also be applied to an MNR method. Theoretically, the same derivative should be obtained for any sufficiently small perturbation in θ; however, because of both the sometimes widely varying sensitivities of the parameters and the numerical-precision limitations, it is beneficial to vary the perturbation size according to the aforementioned rule. The sensitivity defined as $\theta_j^2 M_{jj}$ was introduced in reference 11 and used again in reference 12 as a means of quantifying the significance of a parameter to the model.

There are four factors which have a significant effect on the speed of the MNRES algorithm. The first factor is the degree of precision demanded by the convergence criteria $\Delta J/J$ and $\Delta\theta/\theta$. Naturally, the more precision desired, the greater the computational demand. How well the algorithm meets the demand is determined by the computer capability (word size) and by the information content of the data analyzed.

The second factor is the accuracy of the sensitivities, particularly during initialization of the algorithm. The perturbation size chosen for θ can have a significant effect on the accuracy of the sensitivities. In turn, the accuracy of the sensitivities directly control the speed of the algorithm since G_i determines both the step size and direction in the parameter-update equation. A third important factor is the scaling term used in most ML algorithms to reduce the step size. This term is commonly used to aid convergence since the computed step size is sometimes too large and may cause divergence. The scaling term used in this study was equal to 1. The fourth factor is unique to MNRES and is determined by the number of parameters n to be estimated. This problem is a result of inverting the $n \times n$ matrix ΔX, which may contain very small numbers as the algorithm proceeds. Experience has shown that checking this matrix inversion for numerical difficulties is important in the MNRES method.

Problem I Extended

Estimation problem I is solved again to allow a comparison between MNRES and the two methods previously considered, MNR and SM. Problem I has been described and figure 1 shows the input and response time histories. MNRES uses the same finite-difference method as that of MNR to determine sensitivities during initialization. Also, MNRES uses the iterative least-squares form of the algorithm; however, only one integration per pass is performed. Because of the small storage requirements, all time histories are saved, thus eliminating the need for an additional integration. Table I presents the results of using the three optimization methods in the estimation algorithm. All three algorithms accurately converged to the correct parameter values. However, MNR was 30 times faster than SM; and MNRES was twice as fast as MNR, or 60 times faster than SM. The number of equivalent evaluations had similar ratios, that is, 715:28:12.

The initial values were relatively close to the final solution and thus allowed a good quadratic approximation of the cost function; therefore, the two Newton-Raphson methods were substantially faster. MNRES, however, capitalized more efficiently on the information obtained from each integration of the system equations. Each integration of the system equations provides information which is immediately incorporated into the numerical process when using MNRES. When using MNR, $n + 1$ system integrations (equivalent evaluations) are required before each updating operation; for problem I, $n + 1$ is equal to 7.

Performance of Methods on Estimation Problem II

Estimation problem II is provided to demonstrate the robustness characteristics of MNRES compared with those of the commonly used MNR. The form of MNR and MNRES used in problem I is used again in problem II. The known system from problem I is analyzed again in problem II, except that measurement noise is added and a step input is used to excite the system. Two cases are considered with different levels of measurement noise. The noise is zero mean and Gaussian; the standard error of the noise for each case is 0.0001 and 0.001, respectively. Figure 3 shows time histories of the input and response variables for the two cases, and table II shows the estimation results.

In case 1 both methods produce equally degraded results; however, MNRES still converges to the same precision level more quickly. In case 2, by using a severe

noise level and a limited amount of data, MNRES was unable to converge at all. The results showed that it was oscillating about a solution, unable to find a new parameter vector which would produce a lower cost. The MNRES used on this problem had no special step-size control logic. The solution that was obtained, however, was as accurate as that obtained by MNR, which did converge.
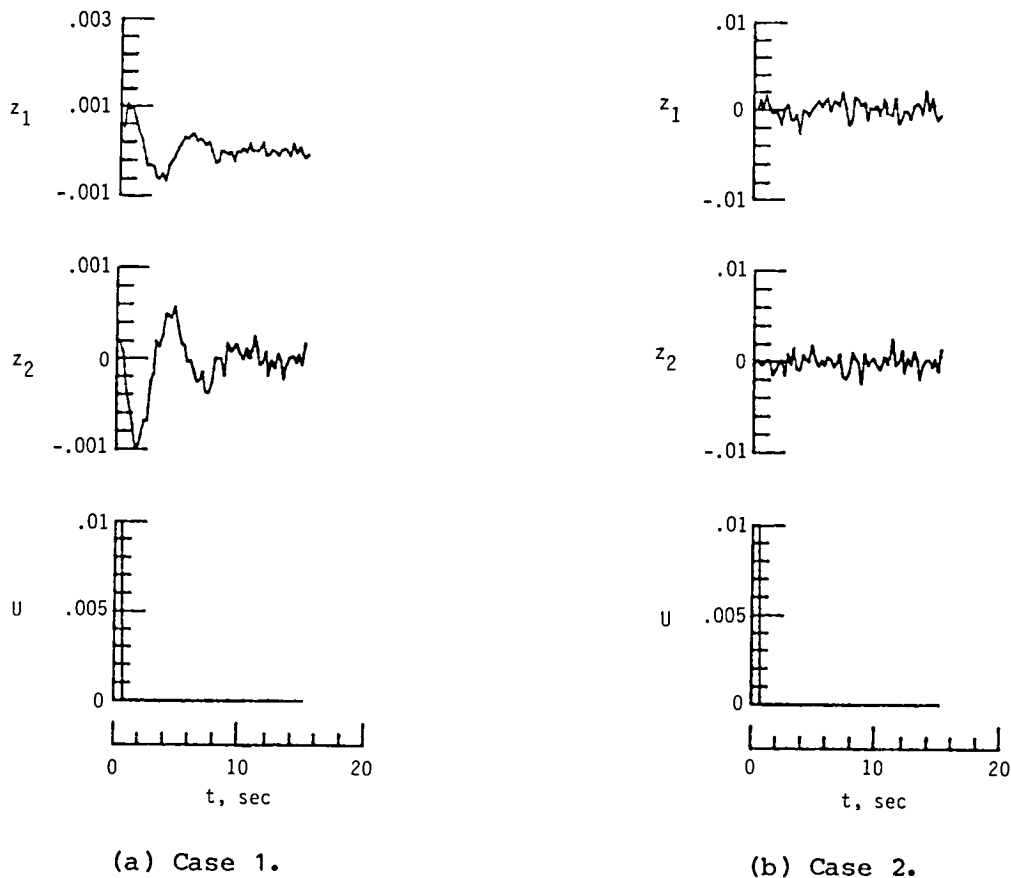

(a) Case 1.   (b) Case 2.

Figure 3.- Time history of input and response variables for problem II.

Meeting convergence requirements does not guarantee accurate results; the error in the estimates ranged from a 5-percent error to a 130-percent error. MNR had both the most accurate and the least accurate estimate. The importance (sensitivity) of a parameter to the model will significantly affect the accuracy of the estimate, particularly under these adverse conditions. Based on these examples, it appears that MNRES performs faster than MNR while providing the same level of precision.

It should be noted that a Euler integration method was used and may have a slightly degraded performance, particularly for the MNR method. Euler integrations tend to have an increasing error with the length of integration. This problem was reduced in part by using relatively few data points. In problem I, only the first 5 sec of data (20 data points) were analyzed. In problem II, for both cases, only 3 sec of data (12 data points) were used. Additionally, step-size control of the optimization algorithm could have been used as another means of improving the convergence capability. This was not used by any algorithm, although SM has the ability to determine whether to extend or contract the polyhedron.

TABLE II.- ESTIMATION PROBLEM II

(a) Case 1

Unknown parameter, θ	True value of θ	Initial value of θ	Final estimated values using method -	
			MNR	MNRES
θ_1	0	0.01	-0.0675	-0.0684
θ_2	-1.5	-1.6	-1.471	-1.471
θ_3	1.0	1.1	1.009	1.010
θ_4	-.5	-.6	-.449	-.448
θ_5	.2	.25	.202	.202
θ_6	.1	.15	.098	.098
Cost			0.105 E-06	0.105 E-06
Equivalent evaluation ...			42	12
Δt, sec			77.54	24.08

(b) Case 2

Unknown parameter, θ	True value of θ	Initial value of θ	Final estimated values using method -	
			MNR	MNRES[1]
θ_1	0	0.01	-0.705	-0.410
θ_2	-1.5	-1.6	-1.228	-1.549
θ_3	1.0	1.1	1.757	.799
θ_4	-.5	-.6	.159	-.238
θ_5	.2	.25	.210	.251
θ_6	.1	.15	.087	.037
Cost			0.104 E-04	0.122 E-04
Equivalent evaluation ...			70	27
Δt, sec			134.44	56.05

[1]MNRES did not converge.

APPLICATION TO NONLINEAR AIRCRAFT ESTIMATION PROBLEMS

In this section the utility of MNRES is demonstrated by application to nonlinear aircraft estimation problems by using both simulated- and real-flight data. For these problems, the computationally least demanding form of MNRES (with eq. (21) instead of (28)) is used to compute sensitivities. The accuracy and robustness of the algorithm are assessed by testing the algorithm on simulated data with known noise levels. Also, performance of the algorithm on real-flight data is checked by comparison with a well-proven ML algorithm. In both cases the problems involve nonlinear lateral models of a general aviation aircraft.

Estimation Using Simulated-Flight Data

In this section ML estimation problem III is considered. This problem simulates real-flight data with varying noise levels. Three cases are considered: case 1 without any measurement noise, case 2 with a representative noise level typical of flight data for the aircraft, and case 3 with twice the noise level of case 2. The standard errors of the simulated measurement noise are shown in table III. In each case the noise is zero mean and Gaussian. The simulated data were created by a

TABLE III.- STANDARD ERRORS OF SIMULATED
MEASUREMENT NOISE

| Output variable	Standard deviations for -		
	Case 1	Case 2	Case 3
β, rad	0	0.010	0.02
p, rad/sec	0	0.010	0.02
r, rad/sec	0	0.010	0.02
ϕ, rad	0	0.005	0.01
a_y, g units ...	0	0.005	0.01

fourth-order Runge-Kutta integration with a step size of 0.05 sec. A maximum likelihood program called MAX is used to solve the problem and it uses the MNRES optimization method. The MNRES method computes sensitivities by using a simple finite-difference technique during initialization and then uses equation (21) thereafter. Program MAX uses two convergence criteria: $\Delta J/J < 0.001$ and $\Delta\theta/\theta < 0.001$. Table IV shows the terms used in the nonlinear aerodynamic model to create the simulation and the parameter estimates obtained through analysis of the simulated data. Time histories are provided for the three cases in figure 4. The control inputs were the same for all three cases and are shown in figure 5. As expected, the estimates of the less easily identified terms are more quickly corrupted as the noise levels increase; however, the estimates are still very reasonable and the time histories are accurately predicted. Table IV shows that the MNRES method can be used effectively in estimating parameters for nonlinear aircraft systems.

Estimation Using Real-Flight Data

In this section ML estimation problem IV is considered. Problem IV uses flight data from a general aviation aircraft operating at an angle of attack of 8°. The problem is solved by using two different ML programs. The first, referred to as program MAX, uses the MNRES algorithm; and the second, referred to as program MAXLIK, uses an MNR algorithm. This MNR algorithm integrates the sensitivity equations to obtain the sensitivities. As used in the simulated-data case, the MNRES method computes sensitivities by using a simple finite-difference technique during initialization and then uses equation (21) thereafter. MAXLIK is a proven code used at the Langley Research Center for aircraft-parameter estimation. MAXLIK was developed from the equations documented in reference 5. For comparison purposes each program uses a

23

TABLE IV.- SIMULATED-DATA ANALYSIS USING MNRES

Unknown parameter, θ	Simulation values	Parameter estimates for –		
		Case 1	Case 2	Case 3
$C_{Y,0}$	0.13	0.1299	0.1298	0.1295
C_{Y_β}	-.411	-.4136	-.4261	-.4401
C_{Y_p}	-.146	-.1524	-.1874	-.2379
C_{Y_r}	.63	.6686	.6070	.5412
$C_{Y_{\delta a}}$	-.053	-.0618	-.0733	-.0872
$C_{Y_{\delta r}}$	.075	.0794	.0775	.0751
$C_{l,0}$	0	.0001	-.0003	-.0005
C_{l_β}	-.123	-.1223	-.1228	-.1240
C_{l_p}	-.397	-.3988	-.4026	-.4094
C_{l_r}	.257	.2573	.2409	.2239
$C_{l_{\delta a}}$	-.182	-.1815	-.1778	-.1755
$C_{l_{\delta r}}$	.077	.0067	.0059	.00497
$C_{l_{\alpha p}}$	2.63	2.6254	2.519	2.4359
$C_{n,0}$	0	-.00005	-.00008	-.0001
C_{n_β}	0	.000003	.0001	.0005
C_{n_p}	-.15	-.1488	-.1524	-.1558
C_{n_r}	-.083	-.0828	-.0861	-.0911
$C_{n_{\delta a}}$	-.0431	-.0425	-.0434	-.0445
$C_{n_{\delta r}}$	1.7	1.7343	1.4419	1.0118
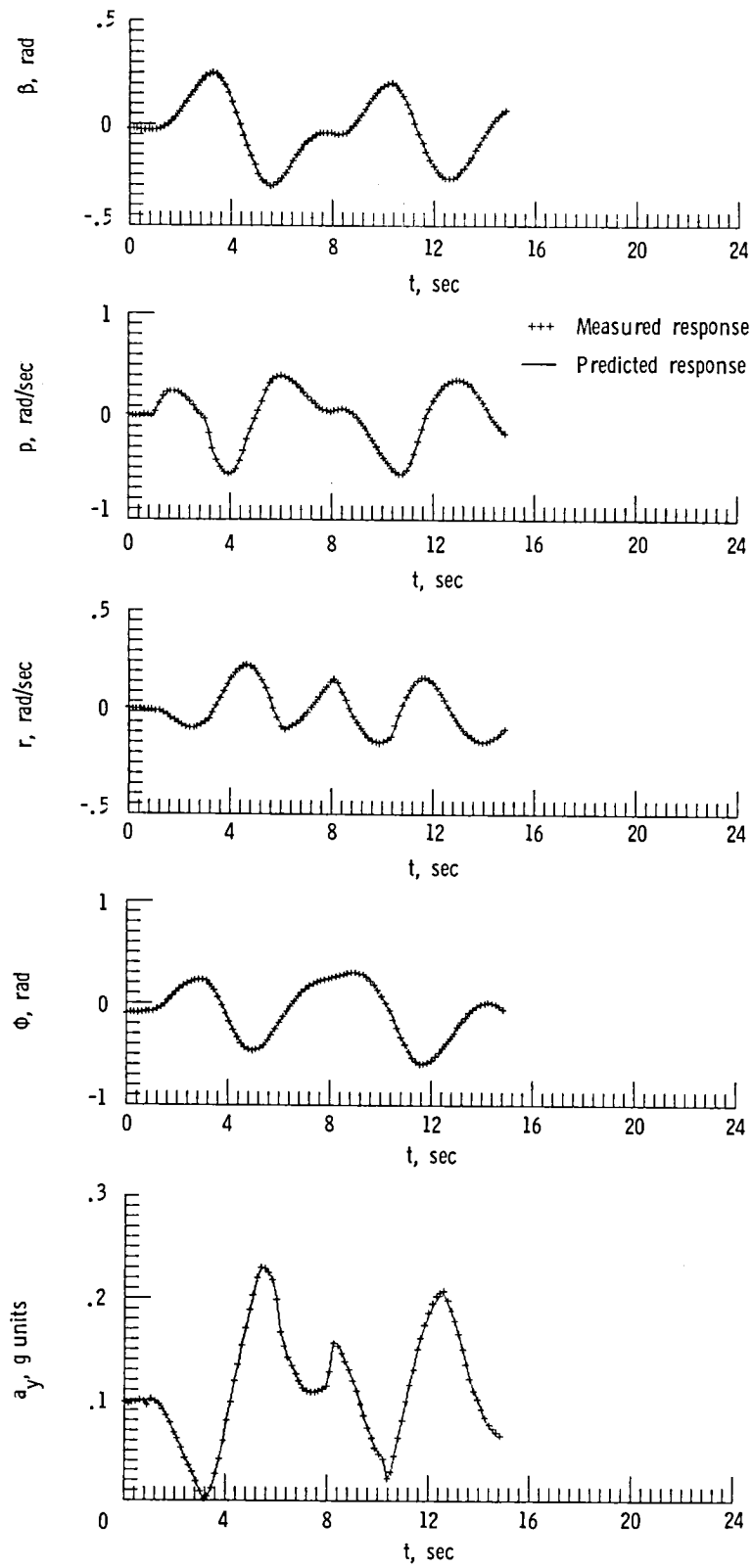
(a) Case 1.

Figure 4.- Measured and predicted responses for lateral simulation.

+++ Measured response
— Predicted response
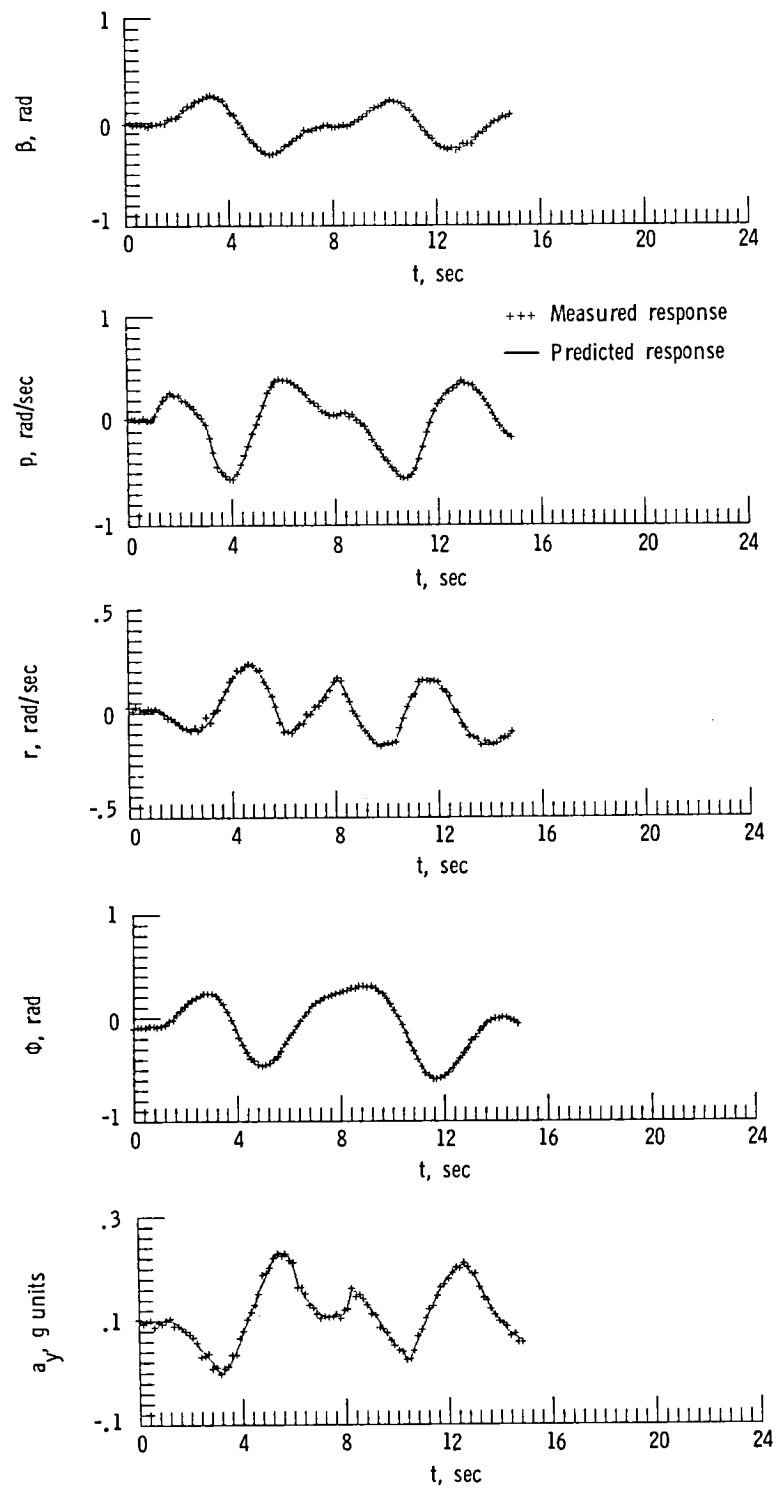
(b) Case 2.

Figure 4.- Continued.

+++ Measured response
— Predicted response
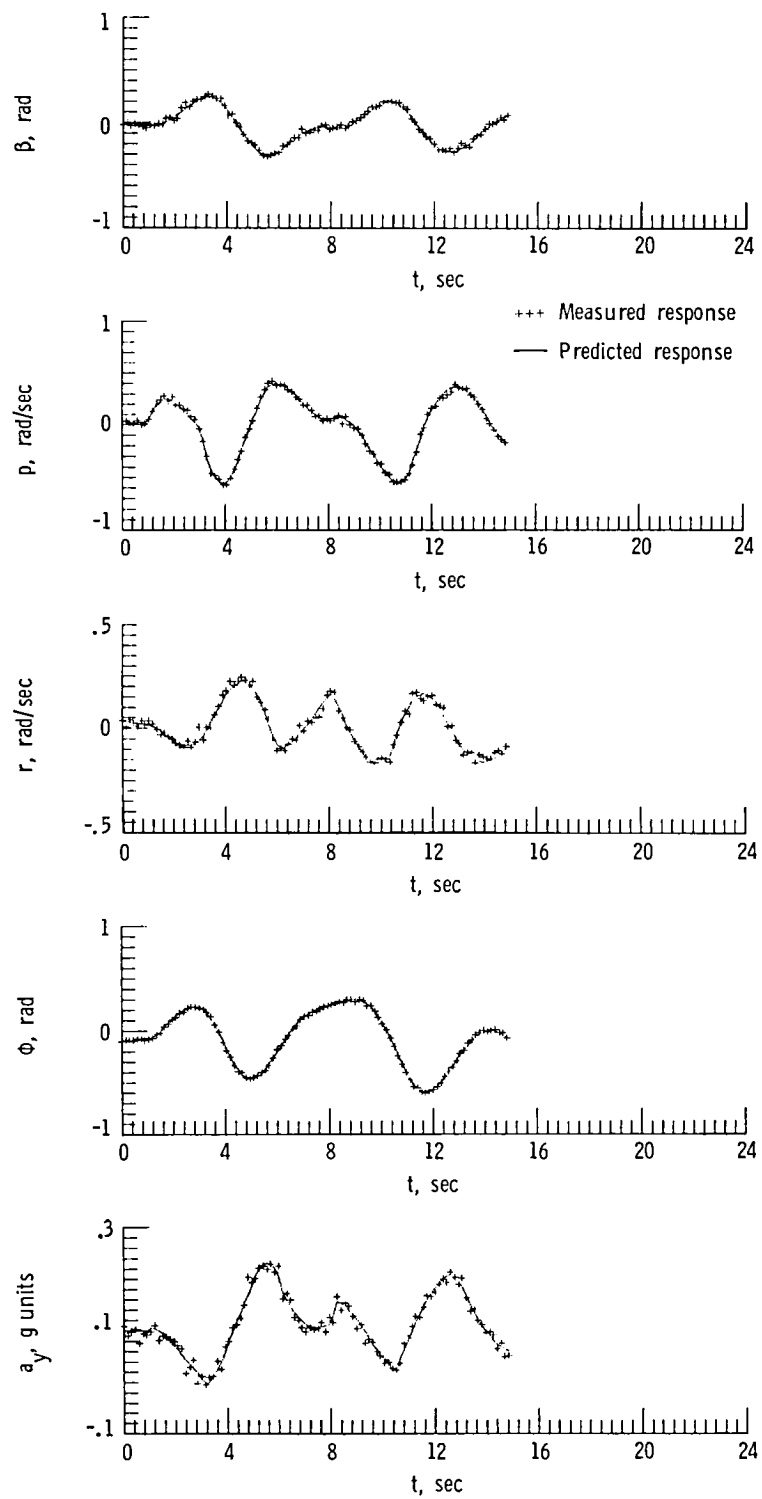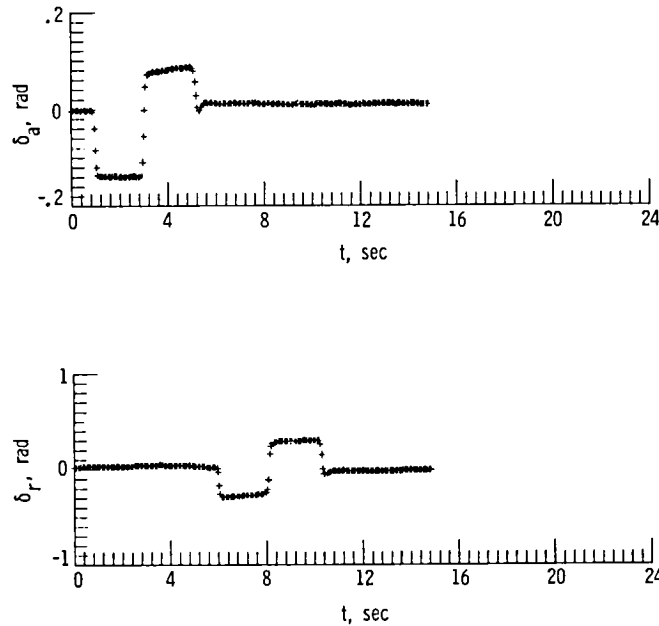
(c) Case 3.

Figure 4.- Concluded.

Figure 5.- Control inputs for
lateral simulation.

fourth-order Runge-Kutta integration method and an integration step size of
0.05 sec. A convergence criterion is set at $\Delta J/J = 0.001$ for both codes. Pro-
gram MAX normally uses an additional criterion restricting the parameter change
$\Delta\theta/\theta$; however, in this problem it is removed to ensure that both programs converge
for the same criterion. Both programs use the same bias and scale-factor corrections
to the flight data. These corrections were determined by using a compatibility
program developed in reference 12.

Estimation problem IV involves a nonlinear lateral model. Table V presents a
comparison between parameter estimates and their standard errors from both MAX and
MAXLIK. It also shows the initial values from the modified stepwise regression
program of references 1 and 13. The sensitivities computed as $\theta_j^2 M_{jj}$ are given in
the last column. Again, there is reasonable agreement between the two approaches.
Standard-error estimates tend to be a little higher for program MAX; this is probably
due to their sensitivity to the derivative information. Repeating the calculations
with program MAX, by allowing the sensitivity ratios to be incorporated into the
initializing derivative calculations, provided a small improvement in the overall
speed of the algorithm. This occurred because only one restart was required during
the optimization process. More improvement would be realized in problems where
restarting occurs several times. Time histories of the measured flight data and
predicted response using the estimated model are shown in figure 6. Execution times
for problem IV indicate program MAX to be about 30 percent faster.

28

TABLE V.- ESTIMATION PROBLEM IV

| Unknown parameter, θ	Initial value of θ	Program MAX		Program MAXLIK		$\theta_j^2 M_{jj}$
		$\hat{\theta}$	$\hat{\sigma}$	$\hat{\theta}$	$\hat{\sigma}$	
$C_{Y,0}$	0.036	0.0061	0.0006	0.0213	0.0005	0.7533 E+03
C_{Y_β}	-.479	-.4603	.0075	-.4608	.0067	.3756 E+05
C_{Y_p}	-.186	-.1378	.0485	-.0604	.0439	.9373 E+03
C_{Y_r}	.522	.6677	.0289	.6209	.0256	.1915 E+04
$C_{Y_{\delta a}}$	-.08	-.0504	.0166	-.0375	.0150	.8129 E+03
$C_{Y_{\delta r}}$	.083	.0814	.0043	.0763	.0037	.9922 E+03
$C_{Y_{\alpha\beta}}$	.45	.4300	.0592	.4512	.0504	.9399 E+03
$C_{l,0}$	0	.0002	.00005	-.0001	.00005	.9618 E+03
C_{l_β}	-.079	-.0872	.0015	-.08	.0013	.1493 E+06
C_{l_p}	-.47	-.5320	.0102	-.4823	.0085	.4529 E+07
C_{l_r}	.187	.1700	.0043	.1543	.0045	.6838 E+04
$C_{l_{\delta a}}$	-.19	-.2035	.0036	-.1852	.0031	.7969 E+05
$C_{l_{\delta r}}$	.01	.00055	.00024	-.0012	.00072	.1340 E+04
$C_{l_{\alpha\beta}}$	-.26	-.2707	.0116	-.2105	.0091	.6327 E+04
$C_{n,0}$	0	-.00063	.00003	-.002	.00002	.1909 E+04
C_{n_β}	.04	.0323	.00045	.0329	.0004	.1515 E+06
C_{n_p}	-.056	-.1043	.0026	-.0916	.0022	.1400 E+06
C_{n_r}	-.15	-.1462	.002	-.1534	.0017	.5039 E+05
$C_{n_{\delta a}}$	0	-.0044	.001	-.0037	.0009	.1780 E+04
$C_{n_{\delta r}}$	-.053	-.0550	.0003	-.0532	.0003	.9048 E+07
[1]$C_{Y_\beta 3}$	-.39	-.39		-.39		
[1]$C_{n_\beta 3}$	.08	.08		.08		

[1]Parameter held fixed.

29
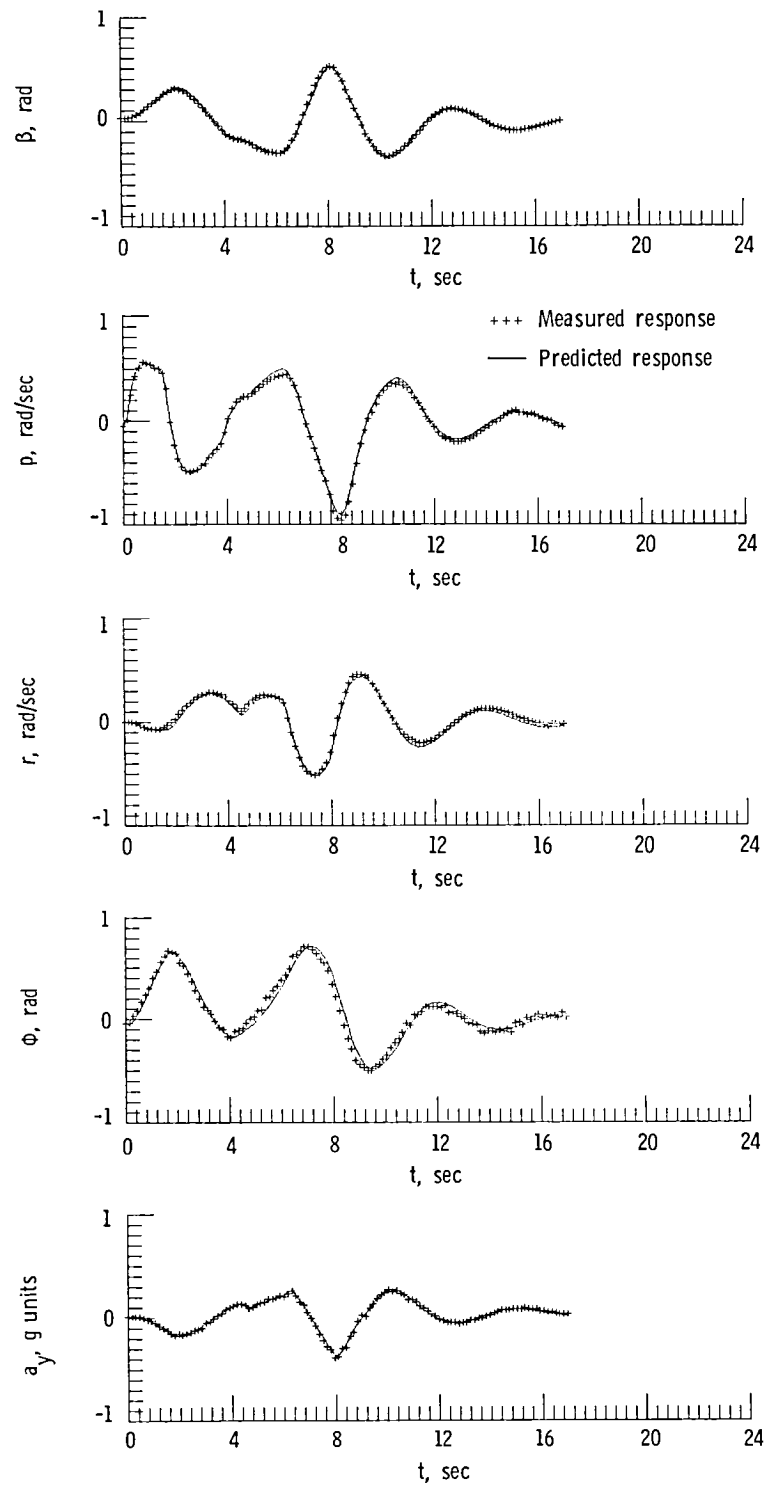
Figure 6.- Real-flight data from measured and predicted
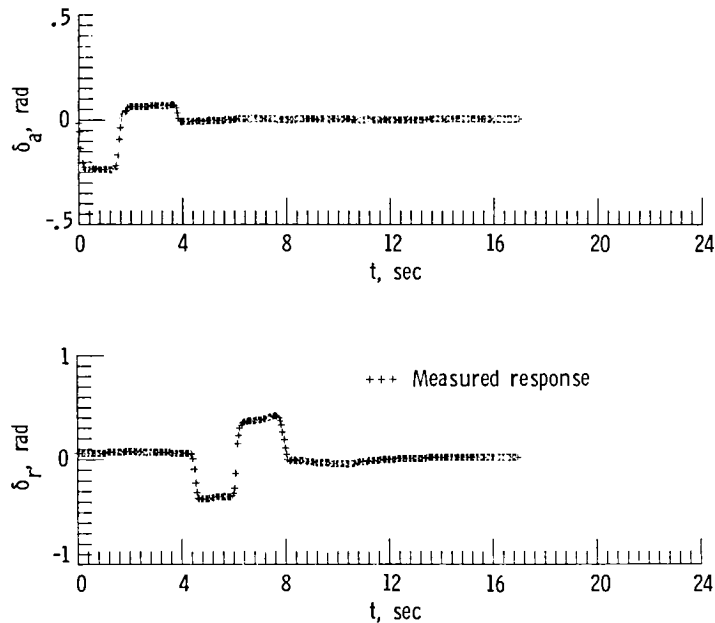responses for lateral simulation.

Figure 6.- Concluded.

CONCLUDING REMARKS

An algorithm for maximum likelihood (ML) estimation using an efficient method for estimating the sensitivities has been developed. The optimization method is referred to as a modified Newton-Raphson with estimated sensitivities (MNRES). The method is applicable to any parameter-estimation problem and is particularly suited for multivariable dynamic systems. In this study nonlinear aircraft systems were analyzed.

The algorithm has three advantages over other commonly used techniques. The first advantage is that the algorithm removes the need to derive sensitivity equations for each new model; this eliminates the computational burden of integrating the sensitivity equations during each iteration of the algorithm. This also provides a lot of flexibility, allowing the model equations to be in any format that is convenient - such as splines, polynomials, or a nonanalytic form. The second advantage is that the algorithm is effective for a variety of methods chosen to fit the output vector surface in parameter space (needed for sensitivity estimation), allowing the user to choose a surface-fitting method best suited to the problem. An approach is discussed which provides a method to reduce storage requirements with little additional computation. The third advantage of the algorithm is that it reduces the computational effort in comparison with the commonly used approach referred to as a modified Newton-Raphson (MNR) method. For small problems (fewer than 15 parameters to be estimated), the reduction can be substantial. For larger nonlinear problems, the reduction is usually more modest; however, improvements may still be significant if data quality, signal compatibility, and sensitivity calculations are good. Based on this study, it appears that the ML/MNRES algorithm generally performs better than the commonly used ML/MNR algorithm.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
May 11, 1984

APPENDIX

ITERATIVE LEAST-SQUARES FORMULATION FOR MNRES

Consider the least-squares problem

$$Y = XS \tag{A1}$$

and the least-squares solution

$$\hat{S} = \left[X^T X\right]^{-1} X^T Y \tag{A2}$$

Define for the $i + 1$ iteration

$$P_{i+1} = \left[(X_i)^T X_i\right]^{-1} \tag{A3}$$

Let a_i^o be one row of X containing old information to be removed from X, and let a_i be a replacement row containing new information to be added to X. Define Z as the common elements of X between two iterations. Partitioning X for the $i - 1$ and ith iterations results in

$$X_{i-1} = \begin{bmatrix} a_i^o \\ z_i \end{bmatrix} \tag{A4}$$

$$X_i = \begin{bmatrix} a_i \\ z_i \end{bmatrix} \tag{A5}$$

By using equations (A4) and (A5), the following relations can be written:

$$(X_{i-1})^T X_{i-1} = \left(a_i^o\right)^T a_i^o + (z_i)^T z_i \tag{A6}$$

$$(X_i)^T X_i = (a_i)^T a_i + (z_i)^T z_i \tag{A7}$$

From equation (A6) obtain

$$(z_i)^T z_i = (X_{i-1})^T X_{i-1} - \left(a_i^o\right)^T a_i^o \tag{A8}$$

32

Substituting equation (A8) into (A7) gives

$$(X_i)^T X_i = (X_{i-1})^T X_{i-1} - \left(a_i^o\right)^T a_i^o + (a_i)^T a_i \qquad \text{(A9)}$$

Substituting equation (A9) into (A3) gives

$$P_{i+1} = \left[(X_{i-1})^T X_{i-1} - \left(a_i^o\right)^T a_i^o + (a_i)^T a_i\right]^{-1} \qquad \text{(A10)}$$

which can also be written as

$$P_{i+1} = \left[P_i^{-1} - \left(a_i^o\right)^T a_i^o + (a_i)^T a_i\right]^{-1} \qquad \text{(A11)}$$

Applying the same development to equation (A2) gives

$$S_{i+1} = P_{i+1}\left[(X_{i-1})^T Y_{i-1} - \left(a_i^o\right)^T y_i^o + (a_i)^T y_i\right] \qquad \text{(A12)}$$

and substituting equations (A2) and (A3) delayed a step into (A12):

$$S_{i+1} = P_{i+1}\left[P_i^{-1} S_i - \left(a_i^o\right)^T y_i^o + (a_i)^T y_i\right] \qquad \text{(A13)}$$

Expanding equation (A13) gives

$$S_{i+1} = P_{i+1} P_i^{-1} S_i - P_{i+1}\left(a_i^o\right)^T y_i^o + P_{i+1}(a_i)^T y_i \qquad \text{(A14)}$$

Noting that

$$S_i - P_{i+1} P_{i+1}^{-1} S_i = 0 \qquad \text{(A15)}$$

and then adding equation (A15) to (A14) gives

$$S_{i+1} = S_i - P_{i+1} P_{i+1}^{-1} S_i + P_{i+1} P_i^{-1} S_i - P_{i+1}\left(a_i^o\right)^T y_i^o + P_{i+1}(a_i)^T y_i \qquad \text{(A16)}$$

$$S_{i+1} = S_i - P_{i+1}\left[P_{i+1}^{-1}\theta_i - P_i^{-1}\theta_i + \left(a_i^o\right)^T y_i^o - (a_i)^T y_i\right] \qquad \text{(A17)}$$

Combining terms gives

$$S_{i+1} = S_i - P_{i+1}\left[(P_{i+1}^{-1} - P_i^{-1})S_i + (a_i^o)^T y_i^o - (a_i)^T y_i\right] \tag{A18}$$

and using equation (A11) yields the desired relation

$$S_{i+1} = S_i - P_{i+1}\left[\left[(a_i)^T a_i - (a_i^o)^T a_i^o\right]S_i + (a_i^o)^T y_i^o - (a_i)^T y_i\right] \tag{A19}$$

REFERENCES

1. Klein, Vladislav; Batterson, James G.; and Murphy, Patrick C.: Determination of Airplane Model Structure From Flight Data by Using Modified Stepwise Regression. NASA TP-1916, 1981.

2. Klein, Vladislav: Identification Evaluation Methods. Parameter Identification, AGARD-LS-104, Nov. 1979, pp. 2-1 - 2-21.

3. Himmelblau, David M.: Applied Nonlinear Programming. McGraw-Hill, Inc., c.1972.

4. Gupta, Narendra K.; and Mehra, Raman K.: Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations. IEEE Trans. Autom. Control, vol. AC-19, no. 6, Dec. 1974, pp. 774-783.

5. Grove, Randall D.; Bowles, Roland L.; and Mayhew, Stanley C.: A Procedure for Estimating Stability and Control Parameters From Flight Test Data by Using a Maximum Likelihood Methods Employing Real-Time Digital System. NASA TN D-6735, 1972.

6. Olsson, Donald M.; and Nelson, Lloyd S.: The Nelder-Mead Simplex Procedure for Function Minimization. Technometrics, vol. 17, no. 1, Feb. 1975, pp. 45-51.

7. Trankle, T. L.; Vincent, J. H.; and Franklin, S. N.: Recent Methods for Nonlinear System Identification. AIAA-81-1865, Aug. 1981.

8. Trankle, T. L.; Vincent, J. H.; and Franklin, S. N.: System Identification of Nonlinear Aerodynamic Models. Advances in the Techniques and Technology of the Application of Nonlinear Filters and Kalman Filters, C. T. Leondes, ed., AGARD-AG-256, Mar. 1982, pp. 7-1 - 7-26.

9. Adams, William M., Jr.; and Tiffany, Sherwood H.: Design of a Candidate Flutter Suppression Control Law for DAST ARW-2. NASA paper presented at 1983 AIAA Conference on Guidance and Control (Gatlinburg, Tenn.), Aug. 15-17, 1983.

10. Ralston, Mary L.; and Jennrich, Robert I.: Dud, A Derivative-Free Algorithm for Nonlinear Least Squares. Technometrics, vol. 20, no. 1, Feb. 1978, pp. 7-13.

11. Klein, V.: On the Adequate Model for Aircraft Parameter Estimation. Report Aero No. 28, Cranfield Inst. Technol., Mar. 1975.

12. Klein, Vladislav; and Schiess, James R.: Compatibility Check of Measured Aircraft Responses Using Kinematic Equations and Extended Kalman Filter. NASA TN D-8514, 1977.

13. Klein, V.; Batterson, J. G.; and Smith, P. L.: On the Determination of Airplane Model Structure From Flight Data. Identification and System Parameter Estimation - Sixth IFAC Symposium, Volume 2, George A. Bekey and George N. Saridis, eds., Pergamon Press, Inc., c.1982, pp. 1034-1039.

1. Report No. NASA TP-2311	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ALGORITHM FOR MAXIMUM LIKELIHOOD ESTIMATION USING AN EFFICIENT METHOD FOR APPROXIMATING SENSITIVITIES		5. Report Date June 1984
		6. Performing Organization Code 505-34-03-02
7. Author(s) Patrick C. Murphy		8. Performing Organization Report No. L-15735
		10. Work Unit No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An algorithm for maximum likelihood (ML) estimation is developed primarily for multi-variable dynamic systems. The algorithm relies on a new optimization method referred to as a modified Newton-Raphson with estimated sensitivities (MNRES). The method determines sensitivities by using slope information from local surface approximations of each output variable in parameter space. The fitted surface allows sensitivity information to be updated at each iteration with a significant reduction in computational effort compared with integrating the analytically determined sensitivity equations or using a finite-difference method. Different surface-fitting methods are discussed and demonstrated. Aircraft estimation problems are solved by using both simulated and real-flight data to compare MNRES with commonly used methods; in these solutions MNRES is found to be equally accurate and substantially faster. MNRES eliminates the need to derive sensitivity equations, thus producing a more generally applicable algorithm.

17. Key Words (Suggested by Author(s)) System identification Maximum likelihood Sensitivities Optimization Parameter estimation	18. Distribution Statement Unclassified – Unlimited Subject Category 08
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified

21. No. of Pages 36	22. Price A03

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

NASA

LIBRARY MATERIAL SLIP		
DO NOT REMOVE SLIP FROM MATERIAL		
Delete your name from this slip when returning material to the library.		
NAME	DATE	MS
E. Moietti	7-03-32	

NASA Langley Form 474 (Rev. Oct. 1999)

If Undeliverable (Section 158
Postal Manual) Do Not Return