NASA Technical Paper 1563

User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation

Richard E. Maine and Kenneth W. Iliff

NOVEMBER 1980

NASA

NASA Technical Paper 1563

User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation

Richard E. Maine and Kenneth W. Iliff
Dryden Flight Research Center
Edwards, California

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1980

CONTENTS

USER'S MANUAL FOR MMLE3,

A GENERAL FORTRAN PROGRAM FOR

MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

Richard E. Maine and Kenneth W. Iliff
Dryden Flight Research Center

INTRODUCTION

This report is a user's manual for the FORTRAN IV computer program MMLE3, a maximum likelihood parameter estimation program capable of handling general bilinear dynamic equations of arbitrary order with or without state noise. Continuous time system equations with discrete sampled observations are used.

Many of the advances in practical application of estimation methodology have been in the area of aircraft stability and control derivatives (ref. 1). The well-defined dynamic model, the high quality automated data acquisition systems (ref. 2), and the presence of analysts oriented towards computer techniques have contributed to this trend. In 1966, NASA Dryden Flight Research Center began investigating (ref. 3) the use of maximum likelihood estimation (ref. 4), based on the theoretical work of Balakrishnan (ref. 5). Taylor and Iliff (ref. 6) refer to the computer program developed as the Newton-Raphson program because it uses a modified Newton-Raphson algorithm (Newton-Balakrishnan) to implement the maximum likelihood estimation. Reference 7 compares the maximum likelihood method to previously used techniques of estimating stability and control derivatives. There have been numerous parallel and subsequent developments (refs. 8 to 10) of techniques identical or similar to the Iliff-Taylor-Balakrishnan approach.

The Newton-Raphson program underwent a gradual evolution based on experience in application to flight data (ref. 11). Reference 12 documents the resulting program released in 1973. This program, named MMLE (modified maximum likelihood estimator), used the same basic algorithm as that of reference 6, but incorporated many features found useful for routinely processing large amounts of flight stability and control data. The program was widely circulated among industry and government agencies.

Extensive, worldwide experience has been obtained using maximum likelihood estimation on actual flight data (refs. 13 to 26). The NASA Dryden Flight Research Center alone has analyzed over 5000 maneuvers from 35 different aircraft with these techniques. Several studies have been made of the factors to be considered in carefully applying maximum likelihood estimation to flight data (refs. 27 to 29). This background of experience and careful study has resulted in maximum likelihood estimation becoming recognized as the leading technique for estimation of aircraft stability and control derivatives (ref. 30). Reference 31 discusses the Dryden Flight Research Center's approach to the gathering and analysis of flight data for maximum likelihood estimation.

The success of maximum likelihood estimation of standard aircraft stability and control derivatives has prompted interest in expanding the scope of its application. Most of the work with actual flight data has been done in stabilized flight at low to moderate angles of attack; a standard linear set of uncoupled equations with two degrees of freedom has been used for longitudinal data, or three degrees of freedom for lateral-directional data. Investigation of unusual aircraft (ref. 32) and extreme flight conditions requires more flexibility in the choice of equations of motion. Application to problems, such as performance estimation (ref. 33) or structural mode identification (ref. 29), also requires this flexibility. Nonaircraft applications have also been considered. The MMLE program, developed primarily for the standard stability and control equations, lacks the versatility to conveniently handle all of the new investigations. Kirsten (ref. 34) notes that MMLE requires the dynamic pressure to remain constant during a maneuver, and Park (ref. 35) says that it is too specialized to analyze structural responses. Neither MMLE nor any of the other widely available programs combine sufficient generality with those features DFRC has found useful in routinely analyzing actual data.

In response to the requirements for a more versatile maximum likelihood estimation program, DFRC began the development of the MMLE3 computer program. MMLE3 was designed to handle a general set of linear or bilinear (sec. 3.1) dynamic equations of arbitrary order. Although greatly refined in application, this estimation problem is solved using the same theoretical tools developed by Balakrishnan in 1966 (ref. 5). However, the final capability added to MMLE3, estimation in the presence of state noise (turbulence), required further theoretical development (ref. 36). Iliff (ref. 37) demonstrated its application to flight data in turbulence. The state noise algorithm is similar in form to the algorithm used without state noise (indeed it reduces to the same algorithm if the state noise power is 0), thus implementation in the same computer program was relatively easy. MMLE3 has the option to use either the state noise or the no-state-noise algorithm.

Iliff's original implementation of the state noise algorithm used the pure continuous time formulation discussed in references 36 and 37. Recent unpublished studies by the authors conclude that a continuous system model with discrete sampled observations is superior in the state noise case (the formulations result in identical algorithms when there is no state noise). MMLE3, therefore, uses a mixed continuous-discrete time formulation.

Section 1 of this report outlines the theory behind the MMLE3 program. Section 2 defines the overall structure and coding philosophy of the program; the most important item discussed is the division of MMLE3 into the basic program and the

user routines. Section 3 documents the use of the basic MMLE3 program, which is a general maximum likelihood estimator for use with any linear system. A set of user routines designed for the standard aircraft stability and control problem is described in section 4. Program listings and reference maps, along with detailed descriptions of the purpose and operation of each subroutine, are found in the Programmer's Manual (ref. 38). Particular attention is paid to the description of the user routines so that the engineer can understand how to program or modify a set of user routines for his specific problem.

SYMBOLS

All data are referenced to fuselage body axes according to right-handed sign conventions.

A	state equation matrix
$\mathscr{A}$	dummy matrix
a	dummy variable
a_n	normal acceleration, g
a_x	longitudinal acceleration, g
a_y	lateral acceleration, g
B	state equation matrix
b	reference span, m (ft)
b	dummy variable
C	observation matrix
C_A	axial force coefficient
C_D	drag coefficient
CG	center of gravity, fraction of chord
C_L	lift coefficient
C_ℓ	rolling moment coefficient
C_m	pitching moment coefficient

C_N	normal force coefficient
C_n	yawing moment coefficient
C_Y	lateral force coefficient
c	reference chord, m (ft)
D	observation matrix
E	observation matrix
$E\{\}$	expected value
FF^*	state noise power spectral density matrix
$f(\ )$	arbitrary function
GG^*	residual covariance matrix
$\mathcal{GG}^*$	measurement noise covariance matrix
g	acceleration of gravity, m/sec^2 (ft/sec^2)
$g(j)$	diagonal element of $(\mathcal{GG}^*)^{-1}$
$g(\ )$	arbitrary function
H	observation matrix
h	altitude, m (ft)
I	identity matrix
I_x	moment of inertia about roll axis, N-m^2 (slug-ft^2)
I_{xe}	engine moment of inertia, N-m^2 (slug-ft^2)
I_{xy}, I_{xz}, I_{yz}	cross products of inertia, N-m^2 (slug-ft^2)
I_y	moment of inertia about pitch axis, N-m^2 (slug-ft^2)
I_z	moment of inertia about yaw axis, N-m^2 (slug-ft^2)

J	cost functional
K	Kalman filter gain matrix
K_α, K_β	flow amplification factors for angle of attack and angle of sideslip
M	Mach number
m	mass, kg (slug)
m	number of observations
N	engine speed positive clockwise viewed from rear, rpm
N	number of time points
n	state noise vector
P	Riccati covariance matrix
p	roll rate, deg/sec
$p(Z/\xi)$	likelihood ratio
q	pitch rate, deg/sec
$\bar{q}$	dynamic pressure, N/m^2 (lbf/ft^2)
R	state equation matrix
$\mathscr{R}$	degrees per radian (57.2958)
r	relaxation factor
r	yaw rate, deg/sec
S	state equation matrix
s	reference area, m^2 (ft^2)
T	thrust, N (lbf)
T	total time, sec
T	transformation matrix
t	time, sec
u	control vector

V	velocity, m/sec (ft/sec)
v	forcing function in state equation
W	*a priori* weighting matrix
w	forcing function in observation equation
x	state vector
x_{a_n}, x_{a_x}, x_{a_y}, x_α, x_β	longitudinal instrument offsets from center of gravity, m (ft)
y_{a_n}, y_{a_x}, y_{a_y}, y_α	lateral instrument offsets from center of gravity, m (ft)
Z	time history of measured response
z	measured observation vector
z_{a_n}, z_{a_x}, z_{a_y}, z_β	vertical instrument offsets from center of gravity, m (ft)
$\prime$	bias vector
α	angle of attack, deg
β	angle of sideslip, deg
Δt	time interval, sec
δ	control deflection, deg
δ_a	aileron deflection, deg
δ_c	extra control deflection
δ_e	elevator deflection, deg
δ_r	rudder deflection, deg
δ_1, δ_2	extra controls
η	measurement noise vector

θ	pitch angle, deg
ξ	vector of unknowns
ξ_0	vector of *a priori* values
Φ	transition matrix
φ	bank angle, deg
ψ	integral of the transition matrix
∇	gradient (row vector)
∇^2	second gradient

Superscripts:

*	transpose
~	predicted estimate
^	corrected estimate

Subscripts:

CG_{flt}, CG_{ref}	at flight or reference center of gravity
c	corrected
i, j, k	general indices
m	measured
p, q, r	rotary derivatives, per rad
α, β	static derivatives, per deg
δ, δ_a, δ_c, δ_e, δ_r, δ_1, δ_2	control derivatives with respect to indicated quantity
ξ	function of ξ
0	bias

Prefix to matrix names:

APR	*a priori* weighting

1.0

Suffixes to matrix names:

I	inverse
L	dimensionalization addition
M	dimensionalization ratio
N	nondimensional
V	variation

1.0 PARAMETER ESTIMATION THEORY

The parameter estimation problem can be defined quite simply in general terms. The system investigated is assumed to be modeled by a set of dynamic equations containing unknown parameters. The system is excited by a suitable input. The input and the actual system response are measured. The values of the unknown parameters are then inferred from the requirement that the model response to the same input should match the actual system response. Formulated in this manner, the identification of the unknown parameters could be easily accomplished by many methods; however, complicating factors arise when considering application to real systems.

The first complication results from the impossibility of obtaining perfect measurements of the response of any real system. The inevitable sensor errors are usually included as additive measurement noise in the dynamic model. Once this noise is introduced, the theoretical nature of the problem changes drastically. It is no longer possible to exactly identify the values of the unknown parameters; rather, the values must be estimated by some statistical criterion. At this point of development, the problem should precisely be referred to as parameter estimation rather than parameter identification, although both terms are often used in the literature. Since the resulting parameter values can no longer be considered perfect, an entire branch of theory (ref. 39) is introduced to answer questions about the accuracy of the estimates. For discrete time observations, the theory of estimation in the presence of measurement noise is relatively straightforward, requiring only basic probability.

The second complication of real systems is the presence of state noise (also referred to as process noise or input noise). State noise is the random excitation of the system from unmeasured sources, the standard example for the aircraft stability and control problem being atmospheric turbulence. If state noise is present, but measurement noise is neglected, the standard line of analysis results in the regression algorithm (ref. 40). The MMLE3 program does not currently include regression.

When both state and measurement noise are considered, the algorithm is more complicated than in the state-noise-only or measurement-noise-only case. However, with reasonable care in the implementation, the computer costs need not be much higher than in the measurement-noise-only case. The MMLE3 program accounts for both state and measurement noise. The measurement-noise-only algorithm can also be used.

The final problem to be faced for real systems is modeling. It has been assumed throughout the above discussion that for some value (called the "true" value) of the unknown-parameter vector, the system is correctly described by the dynamic model. Physical systems are seldom described exactly by simple dynamic models, so the question of modeling error arises.

There is no comprehensive theory of modeling error available. The most common approach taken amounts to ignoring it. Any modeling error is simply treated as state noise and/or measurement noise, in spite of the fact that the modeling error may be deterministic rather than random. The assumed noise statistics can then be adjusted to include the contribution of the modeling error. This procedure has not been rigorously justified, but combined with careful choice of the model, is probably the best approach available. Schweppe (ref. 41) discusses this question in heuristic terms.

Several methods have been advocated which purport to determine the structure of the dynamic model from the data (refs. 42 and 43). This basic concept, however, is fraught with problems for real systems. It is likely that model structure determination will continue to require the engineer's careful consideration of the phenomenology of the physical system to supplement the automatic procedures. MMLE3 does not incorporate automatic model structure determination.

1.1 Maximum Likelihood Estimation

This section describes the maximum likelihood estimation method used by MMLE3. Maximum likelihood estimation has many desirable statistical characteristics; for example, it yields asymptotically unbiased, consistent, and efficient estimates (refs. 4, 5, and 36).

The maximum likelihood estimates are obtained by choosing the vector of unknown parameters, ξ, to maximize the likelihood ratio $p(Z/\xi)$, where Z is the measured response of the system. The crucial element of the theory is the definition of the likelihood ratio. This will be discussed separately for the cases with measurement noise only and with both state and measurement noise.

1.1.1 Measurement Noise Only

When only measurement noise is present, the likelihood ratio is easily computed even for nonlinear systems. The MMLE3 program is written specifically for linear or bilinear (sec. 3.1) systems, but the theory is valid in general. Most nonlinear problems can be reformulated as bilinear systems, so the restriction on MMLE3 is not severe.

The actual system is assumed to be described by

$$\begin{aligned} \dot{x}(t) &= f\left[x(t),\, u(t),\, \xi_{true}\right] \\ z(t_i) &= g\left[x(t_i),\, u(t_i),\, \xi_{true}\right] + \mathcal{G}\eta(t_i) \end{aligned} \tag{1}$$

where

x	state vector
u	input vector
z	measured observation vector
η	measurement noise vector
ξ_{true}	"true" value of vector of unknown parameters

The measurement, z, is obtained at the discrete time points t_i. The vectors $\eta(t_i)$ are assumed to be zero mean, Gaussian, independent random vectors with identity covariance.

For arbitrary ξ, the model (or calculated) system is

$$\dot{\tilde{x}}_\xi(t) = f\left[\tilde{x}_\xi(t),\, u(t),\, \xi\right]$$
$$\tilde{z}_\xi(t_i) = g\left[\tilde{x}_\xi(t_i),\, u(t_i),\, \xi\right] \tag{2}$$

The tilde ($\tilde{}$) notation for the calculated quantities is used for consistency with the state noise and measurement noise case.

The observation, Z, is measured at a finite number of time points. Therefore, it seems natural to maximize $p(\xi/Z)$, the probability of ξ given Z. Unluckily, $p(\xi/Z)$ is difficult to define in general. However, we can take advantage of Bayes' rule to write

$$p(\xi/Z)p(Z) = p(\xi,Z) = p(Z/\xi)p(\xi) \tag{3}$$

and thus

$$p(\xi/Z) = p(Z/\xi)\frac{p(\xi)}{p(Z)} \tag{4}$$

A specific Z has been measured, so $p(Z)$ is a constant, independent of ξ. Furthermore, we assume that there is no *a priori* preference for any particular value of ξ, so $p(\xi)$ is a constant (section 1.3 discusses the case where *a priori* information is available). Thus $p(Z/\xi)$ differs from $p(\xi/Z)$ only by a constant factor. The maximum likelihood method works with $p(Z/\xi)$ instead of $p(\xi/Z)$ because the former is easier to define. Using the Gaussian assumption for the measurement noise, we can write directly

$$p(Z/\xi) = \left[(2\pi)^m |\mathcal{G}\mathcal{G}^*|\right]^{-\frac{1}{2}N} \exp\left\{-\frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_\xi(t_i) - z(t_i)\right]^* (\mathcal{G}\mathcal{G}^*)^{-1}\left[\tilde{z}_\xi(t_i) - z(t_i)\right]\right\} \tag{5}$$

The logarithm of the likelihood ratio is usually used because of its simpler form:

$$\log p(Z/\xi) = -\frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right]^{*} (\mathcal{G}\mathcal{G}^{*})^{-1}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right] - \frac{N}{2}\log|\mathcal{G}\mathcal{G}^{*}| - \frac{N}{2}m\log 2\pi \quad (6)$$

The last term is constant and can be neglected in the minimization. If $\mathcal{G}$ is known, the next to last term is also constant.

Multiplying by -1 to get a minimization, rather than a maximization problem, results in the following simple and intuitively attractive cost function:

$$J(\xi) = \frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right]^{*} (\mathcal{G}\mathcal{G}^{*})^{-1}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right] \quad (7)$$

1.1.2 Measurement Noise and State Noise

When both measurement and state noise are present, there is no known practical method of computing the likelihood ratio for nonlinear systems. Attention is therefore restricted to linear (and bilinear) systems.

The assumed equations of the actual system are

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Fn(t)\\ z(t_i) &= Cx(t_i) + Du(t_i) + \mathcal{G}\eta(t_i)\end{aligned} \quad (8)$$

where n is the state noise vector. The matrices A, B, C, D, and F are functions of ξ. In general, A, B, C, and D can include known functions of time (the time dependence is omitted from the notation for simplicity). The state noise vector, n, is assumed zero mean, Gaussian, and white, with unity spectral density. The measurement noise vectors, $\eta(t_i)$, are assumed to be zero mean, Gaussian, independent random vectors with identity covarience. The measurement and state noise are assumed to be independent.

The log likelihood ratio is then

$$\log p(Z/\xi) = -\frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right]^{*} (GG^{*})^{-1}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right] - \frac{N}{2}\log|GG^{*}| - \frac{N}{2}m\log 2\pi \quad (9)$$

where $\tilde{z}$ is the predicted estimate of z, and GG^{*} is the variance of the filter residuals. Note that this equation is identical to equation (6), except that $\tilde{z}$ is now computed using the Kalman filter (ref. 4) and G replaces $\mathcal{G}$. Naturally, equation (6) can be viewed as a special case of equation (9) with 0 for the Kalman filter gain.

There are two subtle aspects of the differences between equations (6) and (9). First, in equation (9), $\tilde{z}$ is a function of the measured data, z, because of the Kalman filter. For the most part this is of no concern, but it creates some problems in the exact computation of the Cramér-Rao bound (sec. 1.4). Second, G is a complicated function of the coefficients, even if $\mathcal{G}$ is known. MMLE3 avoids this problem by determining G directly instead of $\mathcal{G}$. For known G, the cost functional to be minimized is

$$J(\xi) = \frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_\xi(t_i) - z(t_i)\right]^* (GG^*)^{-1}\left[\tilde{z}_\xi(t_i) - z(t_i)\right] \tag{10}$$

which corresponds to equation (7).

The equation for $\tilde{z}_\xi$ is as follows:

Prediction step—

$$\begin{aligned}\tilde{x}_\xi(t_{i+1}) &= \Phi\hat{x}_\xi(t_i) + \psi Bu\left(t_{i+\frac{1}{2}}\right)\\ \tilde{z}_\xi(t_{i+1}) &= C\tilde{x}_\xi(t_{i+1}) + Du(t_{i+1})\end{aligned} \tag{11}$$

where

$$\begin{aligned}\Phi &= e^{A\Delta t}\\ \psi &= \int_0^{\Delta t} e^{As}\,ds\end{aligned} \tag{12}$$

Correction step—

$$\hat{x}_\xi(t_{i+1}) = \tilde{x}_\xi(t_{i+1}) + K\left[z(t_{i+1}) - \tilde{z}_\xi(t_{i+1})\right] \tag{13}$$

The Kalman filter gain matrix is defined as

$$K = PC^*(GG^*)^{-1} \tag{14}$$

where P is the solution to the discrete time Riccati equation. MMLE3 uses the steady state, continuous time, Riccati equation for an approximate solution for P

$$AP + PA^* - \frac{1}{\Delta t}PC^*(GG^*)^{-1}CP + FF^* = 0 \tag{15}$$

Although this algorithm incorporates a Kalman filter, it should not be confused with the extended Kalman filter algorithm (ref. 44). The filter is not introduced arbitrarily into the formulation; it appears in the implementation of the spectral factorization of the noise. The Kalman filter is used here only for the linear problem of state estimation. The extended Kalman filter algorithm applies the Kalman filter in an ad hoc, linearized manner to the nonlinear problem of combined state and parameter estimation.

The likelihood functional in MMLE3 is expressed in terms of the residual covariance matrix, GG^*, instead of the measurement noise covariance matrix, $\mathcal{G}\mathcal{G}^*$. This results in several simplifications in the algorithm. It does, however, raise one complication which must be addressed. F and $\mathcal{G}$ are unrelated, but there are some combinations of F and G that are not physically meaningful; note that

$$\mathcal{G}\mathcal{G}^* = GG^* - CPC^* \tag{16}$$

The left-hand side must be positive semidefinite to be meaningful, but the right-hand side can be made negative for any GG^* by making F (and thus P) large enough. The constraint

$$GG^* - CPC^* \geq 0 \tag{17}$$

must be satisfied to insure meaningful results. This constraint is equivalent to requiring that the eigenvalues of KC be less than or equal to 1. A necessary condition for this is that the diagonal elements of the matrix KC are less than 1. Because KC generally is nearly diagonal, the diagonal constraint may reasonably approximate the eigenvalue constraint. MMLE3 uses the diagonal constraint because of its simplicity. To insure that impossible constraints are not imposed, MMLE3 constrains only those diagonal elements of KC which correspond to unknown diagonal elements of F.

1.2 Minimization of the Cost Functional

The maximum likelihood method consists of choosing the vector of unknowns, ξ, to maximize the likelihood functional. The likelihood functional has been defined in the previous sections. It remains to describe an algorithm for the maximization. The problem is usually restated as minimizing the negative log likelihood functional of equation (7) or (10) (referred to as the cost functional).

1.2.1 Newton-Balakrishnan Algorithm

The Newton-Raphson algorithm is a well-known technique for iterative minimization of nonlinear functions. An initial estimate, ξ_0, is required. The estimate is then revised iteratively using the equation

$$\xi_{i+1} = \xi_i - \left[\nabla^2_\xi J(\xi_i)\right]^{-1} \nabla^*_\xi J(\xi_i) \tag{18}$$

The properties of this algorithm are well documented in the basic textbooks (ref. 45).

The Newton-Raphson algorithm requires values of the first and second gradients of J. Computation of the first gradient is straightforward—

$$\nabla_\xi J(\xi) = \sum_{i=1}^{N} \left[\tilde{z}_\xi(t_i) - z(t_i)\right]^* (GG^*)^{-1} \nabla_\xi \left[\tilde{z}_\xi(t_i)\right] \tag{19}$$

The second gradient is

$$\nabla_\xi^2 J(\xi) = \sum_{i=1}^{N} \nabla_\xi^* \left[\tilde{z}_\xi(t_i)\right]^* (GG^*)^{-1} \nabla_\xi \left[\tilde{z}_\xi(t_i)\right] + \sum_{i=1}^{N} \left[\tilde{z}_\xi(t_i) - z(t_i)\right]^* (GG^*)^{-1} \nabla_\xi^2 \left[\tilde{z}_\xi(t_i)\right] \tag{20}$$

Computation of the first term of equation (20) is straightforward. The second term requires inordinate amounts of computer time and core; however, it has expected value 0 if ξ is at its "true" value. The second term can therefore be neglected if the algorithm starts near enough to the converged solution. The Newton-Raphson algorithm with this term neglected is referred to as the Newton-Balakrishnan algorithm.

1.2.2 Improvement of Convergence Properties

Since Newton-Raphson type methods converge only if the starting estimates are close enough to the minimizing values, it is common to use special startup algorithms to improve the probability of convergence. For this purpose, MMLE3 contains an option (sec. 3.3.8(19)) to estimate only control derivatives, biases, and initial conditions for the first iteration. The cost functional $J(\xi)$ is quadratic in these unknowns when the other unknowns are constrained to the starting values; thus the Newton-Balakrishnan algorithm attains the constrained minimum in one iteration. In subsequent iterations, all unknowns are estimated.

A program option is available to aid convergence by multiplying the diagonal elements of the second gradient matrix by a constant. With this option invoked, the algorithm becomes similar to the gradient algorithm (ref. 45); convergence becomes more nearly monotone, but requires many more iterations than with Newton-Raphson. The gradient-like algorithm tends to stall before reaching the minimum. Since this stall is easily confused with convergence to the minimum, the final iterations should always be done with the full Newton-Balakrishnan algorithm. Use of the gradient-like algorithm is generally discouraged because of its slow convergence. An alternate approach to initial convergence problems is to use an *a priori* weighting (sec. 1.3) for the first few iterations.

1.2.3 Inequality Constraints

When state noise is present, the minimization is subject to the inequality constraint of equation (17). As mentioned previously, this constraint is approximated by constraining some diagonal elements of KC to be less than or equal to 1. Although simpler than the eigenvalue constraint, this is still a nonlinear inequality constraint.

Minimizing a nonlinear functional subject to a nonlinear inequality constraint is a problem in nonlinear programming. The most common solution technique for such problems is to solve a series of quadratic programming problems that locally approximate the nonlinear problem and converge to the solution of the nonlinear problem (ref. 46). This approach is used by MMLE3.

1.3 Inclusion of *A Priori* Information

Predicted-derivative estimates are often available from previous cases or other sources. It is sometimes desirable to have the estimation algorithm consider this *a priori* information in addition to the new information obtained from the maneuver. This can be heuristically accomplished by adding to the cost functional a quadratic penalty function for departure from the predicted values.

The resulting cost fuctional is

$$J(\xi) = \frac{1}{2}\sum_{i=1}^{N}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right]^{*}\,(GG^{*})^{-1}\left[\tilde{z}_{\xi}(t_i) - z(t_i)\right] + \frac{1}{2}(\xi - \xi_0)^{*}\,W(\xi - \xi_0) \qquad (21)$$

where ξ_0 is the vector of *a priori* values and W is the *a priori* weighting matrix. The maximum likelihood method with the quadratic penalty function can be interpreted as maximizing the unconditional probability density instead of the conditional probability density function.

The *a priori* weighting option can be used to improve the convergence of ill-conditioned cases. The resulting estimates, however, are biased. When *a priori* weighting is used to aid convergence, it is recommended that the *a priori* weighting be removed after initial convergence, and the last few iterations run without *a priori* weighting. The converged values from the algorithm with *a priori* weighting will usually be close enough to the maximum likelihood estimates that the program will rapidly converge to the asymptotically unbiased maximum likelihood estimates if the *a priori* weighting is then removed. When *a priori* weighting is used without this final step, care should always be taken that the resulting bias in the estimates does not lead the user to false conclusions.

1.4 Cramér-Rao Bound

With any parameter estimation method, it is important to have a measure of the accuracy of the estimates. Reference 47 discusses the evaluation of the accuracy of the estimates, including detailed treatment of the subjects briefly mentioned here. In the absence of modeling error or bias, the scatter of the estimates is a reasonable indication of the overall accuracy. If only one or two cases are available at a given condition, however, the scatter cannot be evaluated; an alternate measure of the accuracy is necessary. Even when many cases are used, it is useful to have an indication of which individual estimates are the most reliable. The Cramér-Rao bound provides the best known analytical measure of the accuracy of the estimates.

The Cramér-Rao bound for unbiased estimators (refs. 4, 5, and 39) is

$$\text{Variance}(\xi) \geq \left(E\left\{ \left[\nabla_\xi \log p(Z/\xi) \right]^* \left[\nabla_\xi \log p(Z/\xi) \right] \right\} \right)^{-1} \tag{22}$$

This equation gives only a lower bound for the variance of the estimates. The maximum likelihood estimates, however, are asymptotically efficient; this means that for large time intervals, the variance is approximately equal to the expression in the above inequality, provided that the system and noise are correctly modeled.

For the no-state-noise case, equation (22) can be evaluated as

$$\text{Variance}(\xi) \cong \left\{ \sum_{i=1}^{N} \left[\nabla_\xi^* \tilde{z}_\xi(t_i)^* \right] (GG^*)^{-1} \left[\nabla_\xi \tilde{z}_\xi(t_i) \right] \right\}^{-1} \tag{23}$$

When state noise is present, the exact Cramér-Rao bound is awkward to compute, but Balakrishnan has shown that equation (23) approaches the bound for large time intervals (ref. 36). Equation (23) is seen to be in inverse of the portion of the second gradient matrix used in the Newton-Balakrishnan algorithm (eq. (20)) The program already computes this matrix, so the Cramér-Rao bounds are available with negligible extra computational effort.

The derivation of equation (23) is based on the assumptions of Gaussian white noise. Reference 47 discusses the problems of applying this equation to actual data with band-limited noise. An approximate correction for the effects of band-limited noise is implemented in MMLE3. The residuals $\left(\tilde{z}_\xi - z\right)$ are filtered with a first order filter the break frequency of which can be defined by the user (sec. 3.3.8(23)). The cost functional (eq. (10)) is then evaluated using the filtered residuals in place of the unfiltered residuals and is multiplied by the Nyquist frequency (ref. 47) divided by the filter break frequency. For white residuals, the result should approximately equal the original cost functional value, which should approximately equal the length of the observation vector. MMLE3 multiplies the variances of equation (23) by the adjusted filtered cost functional divided by the length of the observation vector.

The MMLE3 program prints both the Cramér-Rao bounds and the estimated correlations (ref. 47). The Cramér-Rao bounds are the best available indicators of overall accuracy. The smaller the Cramér-Rao bound, the more confidence that can be placed in the estimates. The estimated correlations can sometimes be used to help find the sources of problems indicated by the Cramér-Rao bounds.

1.5 Estimation of Residual Power

The previous discussions have largely neglected the question of estimating G. Theoretically, maximum likelihood estimates of G, as well as the other unknowns, can be obtained by maximizing the likelihood functional shown in equation (9). This would be easy except that $\tilde{z}_\xi$ is a complicated funciton of G. This section describes a two-step estimator that estimates G and the other unknowns separately. When

state noise is not present, this procedure obtains the maximum likelihood estimates. With state noise, the estimates obtained are very close to the maximum likelihood estimates.

The basic G estimator is derived by maximizing equation (9) with respect to G, ignoring the dependence of $\tilde{z}_\xi$ on G. The solution to this maximization problem can be written explicitly

$$GG^* = \frac{1}{N}\sum_{i=1}^{N}\left[\tilde{z}_\xi(t_i) - z(t_i)\right]\left[\tilde{z}_\xi(t_i) - z(t_i)\right]^* \tag{24}$$

To estimate G and the other unknowns, the following two-step algorithm is used each iteration:

1. Use one iteration of the Newton-Balakrishnan algorithm to estimate all of the unknowns except G.

2. Use equation (24) with $\tilde{z}_\xi$ evaluated at the revised ξ to obtain a new estimate of G. (Repeat steps 1 and 2 until convergence.)

The best convergence is usually obtained by doing a few iterations with G fixed, before starting the above algorithm. This is because the sample residual power in the first iteration or two is often quite large and likely to give a worse estimate of G than the starting value.

An overrelaxation algorithm can sometimes speed the convergence of G determination. The MMLE3 program includes an option to use logarithmic overrelaxation on the diagonal elements of $(GG^*)^{-1}$. If the previous estimate of a diagonal element is g_0, and the estimate from equation (24) is g_1, a further revised estimate g_2 is obtained from the equation

$$g_2 = ag_1 \tag{25}$$

where

$$a = \left(\frac{g_1}{g_0 b}\right)^{r-1} \tag{26}$$

where b is the geometric average of the $\frac{g_1}{g_0}$ values for all of the weighted signals, and r is a relaxation factor. Values of r greater than 1 correspond to overrelaxation; values smaller than 1 correspond to underrelaxation. If r equals 1 there is no relaxation. This relaxation is used only for diagonal elements of $(GG^*)^{-1}$. Each off-diagonal element is multiplied by the square roots of the "a" values for the diagonal

elements in the corresponding row and column.

If the state noise algorithm is used, the two-step procedure described for estimating G and the other coefficients results in slow, sometimes erratic, convergence. The reasons for this problem are discussed and a heuristically based fix is described below.

The two-step procedure does not compute correlations between G and the other unknowns. It therefore converges best when such correlations are small. When state noise is not present, G affects the other coefficients only through the relative signal weighting in the cost functional. In general, the coefficients are quite insensitive to changes in the weighting; therefore, the two-step procedure converges very well. When state noise is present, G also affects the Kalman filter gain matrix, K. Most of the coefficients are relatively uncorrelated with K, and thus with G. F, however, enters the cost function only through K; therefore, unknowns in F are strongly affected by changes in G. A heuristic adjustment for F is derived assuming that the optimum K is independent of G; therefore, when G is changed, F is adjusted to minimize the resulting change in K. This is effected, to a first approximation, by multiplying the ith row of F by

$$\frac{\sum_j C(i,j)^2 \, g(j) \sqrt{\dfrac{g_0(j)}{g_2(j)}}}{\sum_j C(i,j)^2 \, g_0(j)} \tag{27}$$

If F was at the optimum for the previous G, this adjustment puts F near enough to the optimum for the new G that convergence to the new optimum is rapid. Only elements of F that are varying are adjusted using this algorithm.

1.6 Solution of the Riccati Equation

This section discusses the methods used for the computation of the Riccati covariance matrix, P, and its gradients. MMLE3 uses only the steady-state solution of the Riccati equation. The discrete time Riccati equation is usually written as a function of the measurement noise covariance matrix, $\mathcal{G}\mathcal{G}^*$. MMLE3, however, works in terms of the residual covariance matrix, GG^*. The discrete time Riccati equation in terms of GG^* is

$$\Phi P \Phi^* - P - \Phi P C^* (GG^*)^{-1} C P \Phi + \Delta t \, \Phi F F^* \Phi^* = 0 \tag{28}$$

Equation (28) is closely approximated by the continuous time Riccati equation

$$AP + PA^* - \frac{1}{\Delta t} P C^* (GG^*)^{-1} C P + FF^* = 0 \tag{29}$$

MMLE3 uses equation (29) because its solution is more convenient than that of equation (28).

Potter's method (ref. 48) is used for the solution of equation (29). The Hamiltonian matrix is defined as

$$\begin{bmatrix} A & -FF^* \\ -\frac{1}{\Delta t} C^* (GG^*)^{-1} C & -A^* \end{bmatrix}$$

The *QR* algorithm (ref. 49) is used to find the eigenvalues and normalized eigenvectors of the Hamiltonian matrix. The matrix of normalized eigenvectors is partitioned into four equal size matrices

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$$

where the eigenvectors corresponding to eigenvalues with positive real parts are in the left partition. Controllability by the state noise and observability is sufficient to guarantee that exactly one-half of the eigenvalues will have positive real parts and that the solution to equation (29) is given by

$$P = -X_{11} X_{21}^{-1} \tag{30}$$

The gradient of P is the solution to a set of Lyapunov equations (one for each partial derivative) of the general form:

$$\mathscr{A}\left(\nabla_\xi P\right)_i + \left(\nabla_\xi P\right)_i \mathscr{A}^* = \mathscr{C} + \mathscr{C}^* \tag{31}$$

The eigenvectors of $\mathscr{A}$ are computed and used to transform $\mathscr{A}$ into $\mathscr{A}'$ as follows:

$$\mathscr{A}' = T^{-1} \mathscr{A} T \tag{32}$$

where $\mathscr{A}'$ is block diagonal, with 1 by 1 blocks for real eigenvalues and 2 by 2 blocks for complex eigenvalues. T is a real transformation matrix, the columns of which are the real eigenvectors and the real and imaginary parts of the complex eigenvector pairs. Then defining

$$\left(\nabla_\xi P\right)_i' = T^{-1} \left(\nabla_\xi P\right)_i T^{*-1} \tag{33}$$

$$(\mathscr{C} + \mathscr{C}^*)' = T^{-1} (\mathscr{C} + \mathscr{C}^*) T^{*-1} \tag{34}$$

we have the block diagonal Lyapunov equation

$$\mathscr{A}' \left(\nabla_\xi P\right)_i' + \left(\nabla_\xi P\right)_i' \mathscr{A}'^* = (\mathscr{C} + \mathscr{C}^*)' \tag{35}$$

Equation (35) is separable into 1 by 1, 2 by 2, and 1 by 2 partitions, the solutions to which can be written explicitly. The inverse of the transformation of equation (33) is then used to obtain $\left(\nabla_{\xi} P\right)_i$.

2.0 MMLE3 PROGRAM STRUCTURE

Because of its generality, MMLE3 sometimes requires a large amount of input to specify a given problem. In addition, some program features that are useful for specific problems have little or no meaning for other problems. To satisfy the conflicting requirements of generality and ease of use, the MMLE3 program is divided into two levels.

The basic level consists of a general maximum likelihood estimation program, applicable to any linear system. The basic program can be run by itself, using input data to completely describe the system to be analyzed and the program options to be used. The use of the basic program is documented in section 3.

At the second level, the basic program is used as the core of a program adapted to a particular application. This adaptation is accomplished by a set of user routines (so called because the user can write or modify them for a particular application). Section 4 describes a particular set of user routines designed for the standard aircraft stability and control problem. When the user routines are written for a particular application, the program input does not have to contain the detailed system specifications; only those items that change from case to case would need to be input. This concept of a modular set of user routines allows the basic program structure to be very general, while retaining the simplicity of input possible in programs designed for specific systems. The use of the user routines to simplify the required input deck is most desirable when large numbers of cases are being processed. The user routines also can be coded to perform computations or operations unique to a specific application. Examples of this function include reading data from special formats, normalizing or correcting the estimates to "standard" conditions, and punching the results in a form suitable to auxiliary programs for purposes such as plotting estimates or updating simulators. The following is a list of the user subroutines and a very brief description of their functions. Naturally, lower level subroutines called from these routines can be defined; the standard aircraft routines include the lower level subroutines INTERP, WTDEF, and WTTRAN.

User Subroutines	Purpose
ONCE	Initialization
WTIN	Predicted-derivative input
USERIN	User input
READTH	Read time histories
THMOD	Modify time history data
AVERAG	Access averages
MATDEF	Matrix defaults
UINIT	Define initial conditions

MAKEL	Define dimensionalization addition (L) matrices
MAKEM	Define dimensionalization ratio (M) matrices
MAKEVW	Define v and w vectors
THOUT	Time history output
OUTPUN	Punched output
TITPLT	Plot titling

The basic program forms the structure within which the user routines fit; therefore, the description of the standard aircraft user routines in section 4 should be regarded as a supplement to the description of the basic program in section 3.

The MMLE3 program is coded to facilitate changing the maximum matrix dimensions. Systems with matrix sizes less than the maximum can be analyzed without changing the code; however, it may prove useful to reduce the maximum dimensions in order to lower core requirements. If the required matrix sizes are larger than the current program dimensions, the code must be changed. Details of an automated system for easily changing the maximum matrix dimensions of the program are described in the Programmer's Manual (ref. 38). The names used in the program and in this report to refer to the maximum matrix dimensions are listed below, along with the current values of the maximum dimensions.

Variable Name	Description	Current Value
LEX	Maximum length of extra signal vector	20
LORD	MAXZ + MAXU + LEX	32
MAXB	Maximum length of bias vector	4
MAXHRD	Maximum number of hard constraints + 1	36
MAXSFT	Maximum number of soft constraints + 1	11
MAXFV	Maximum number of independent unknowns in F	4
MAXKV	Maximum number of independent unknowns affecting K	15
MAXTV	Maximum total number of unknowns (including constraints)	50
MAXU	Maximum length of control vector	4
MAXX	Maximum length of state vector	7
MAXZ	Maximum length of observation vector	8
NI	Maximum number of independent unknowns + 2	35

The maximum number of time points that can be plotted for each maneuver is 1200; plots will automatically be thinned to avoid exceeding this limit. There is no software limit on the number of time points that can be analyzed in a maneuver.

3.0 BASIC PROGRAM

This section describes the basic program and the input required to execute it. In some cases, the input variables of the basic program interact with the user

routines (the most common interaction is for the user routines to change the default values of the basic program). These interactions are described where appropriate. In addition, this section points out where the input cards for the user routines would be placed in the input deck.

3.1 Equations of Motion, Filter, and Gradients

The general bilinear equations of motion used in the MMLE3 program are

$$\begin{aligned} R(t)\dot{x}(t) &= A(t)x(t) + B(t)u(t) + S(t)\iota(t) + v(t) + Fn(t) \\ z(t) &= C(t)x(t) + D(t)u(t) + H(t)\iota(t) + E(t)\dot{x}(t) + w(t) + G\eta(t) \end{aligned} \tag{36}$$

Although equation (36) contains more terms than equation (8) in the theory discussion, the differences do not affect the theory in any important manner. Equation (36) can be recast in the form of equation (8) by appropriate substitutions. The theoretical discussion of section 1 uses the simpler form to keep the details from obscuring the theory. In this section, the exact equations used by MMLE3 are given.

The vector u is the measured control minus a known bias (UOFF), and z is the measured response minus a known bias (YOFF). The biases UOFF and YOFF are closely tied to the treatment of the initial condition. The basic program uses perturbation equations. The state initial condition is defined as 0; UOFF and YOFF are set to the measured control and observation at the first time point. Other treatments of the initial conditions and biases are possible through user routine UINIT. If information is available to define the initial state in absolute (as opposed to perturbation) terms, it would be natural to define UOFF and YOFF as 0.

The vector $\iota(t)$ is used to allow unknown bias terms to be included in the model. The first element of $\iota(t)$ is defined as a constant value of 1 for the entire case in order to allow constant bias terms. The remaining elements of $\iota(t)$ are used to allow the biases for multiple maneuvers (sec. 3.3.8(1)) to be independent. The second element of $\iota(t)$ is defined to be 0 during the first maneuver and 1 for all subsequent maneuvers; the third element is defined to be 0 during the first and second maneuvers and 1 for all subsequent maneuvers, and so on. Consequently, the biases for the first maneuver are the derivatives for the first element of $\iota(t)$, the biases for the second maneuver are the sums of the derivatives for the first and second elements of $\iota(t)$, and so on. The first MAXB maneuvers can therefore have independent biases; the biases for any subsequent maneuvers are equal to the biases of the MAXBth maneuver.

The MMLE3 program allows the system matrices to be time varying (this is equivalent to allowing bilinear equations). The unknown coefficients, however, cannot be treated as time functions. The time-varying system matrices are related to the time-invariant unknowns by equations such as

$$A_{ij}(t) = AM_{ij}(t) \times AN_{ij} + AL_{ij}(t) \tag{37}$$

Note that the operation used is element-by-element multiplication, not matrix multiplication.

The time-invariant unknown coefficients are contained in the AN matrix. The form of this equation is motivated by the aircraft equations of motion (sec. 4), where the unknown nondimensional coefficients are multiplied by known time functions to get the dimensional coefficients. We will refer to AN as a nondimensional matrix, AM and AL as dimensionalization matrices, and A as a dimensional matrix. The use of this terminology, based on the aircraft equations, does not imply any program requirement that the elements of AN actually be nondimensional. AM and AL are arbitrary known functions of time defined by the user routines. Variations of AM and AL with time should be continuous and slow compared with the system response time; these matrices are often time invariant, which results in a considerable saving of computer time. If the user routines are not called, each element of $AM(t)$ is defined as 1, and $AL(t)$ as 0, so that the A and AN matrices are equivalent. In this same manner, the B, S, R, C, D, H, and E matrices are defined in terms of time-invariant "nondimensional" matrices. The F and G matrices do not have nondimensional forms.

In some cases, the same unknown parameter can appear in two or more matrix locations. This results in hard constraints that must be satisfied as the matrices are updated. The handling of such constraints is discussed in section 3.3.11(6). Soft constraints, those implemented by a penalty function approach, are discussed in section 3.3.11(7).

Two general restrictions are necessary: R must be nonsingular and $ER^{-1}F$ must be 0 for all values of the unknown parameters in some neighborhood of the "true" value. The restriction on $ER^{-1}F$ insures that the state noise and measurement noise are independent; it is not needed if the $\dot{x}$ in the $E\dot{x}$ term is considered to be the $\dot{x}$ computed with $F = 0$. The residual covariance matrix (see below) must be nonsingular.

The steady state Riccati covariance matrix, P, is defined by

$$(R^{-1}A)P + P(R^{-1}A)^* - P(ER^{-1}A + C)^* \frac{(GG^*)^{-1}}{\Delta t}(ER^{-1}A + C)P + (R^{-1}F)(R^{-1}F)^* = 0 \qquad (38)$$

with the requirement that P be symmetric. The steady state Kalman filter gain matrix is defined by

$$K = P(ER^{-1}A + C)^*(GG^*)^{-1} \qquad (39)$$

When the system matrices are time varying, P, K and their gradients are computed using average values in the system matrices. All of the other expressions are reevaluated at each time point. Naturally, if $F = 0$ (the no-state-noise case), then $P = 0$ and $K = 0$.

3.1

The filter equations for the state estimation are as follows:

Prediction step—

$$\tilde{x}_\xi(t_{i+1}) = \Phi\hat{x}_\xi(t_i) + \psi R^{-1}Bu(t_{i+\frac{1}{2}}) + \psi R^{-1}\nu + \psi R^{-1}\nu(t_{i+\frac{1}{2}})$$

$$\tilde{z}_\xi(t_{i+1}) = \left(ER^{-1}A + C\right)\tilde{x}_\xi(t_{i+1}) + \left(ER^{-1}B + D\right)u(t_{i+1}) \tag{40}$$

$$+ \left(ER^{-1}S + H\right)\nu + ER^{-1}\nu(t_{i+1}) + w(t_{i+1})$$

where

$$\Phi = e^{R^{-1}A\Delta t} \tag{41}$$

$$\psi = \int_0^{\Delta t} e^{R^{-1}As}\, ds \tag{42}$$

and where throughout this section an index of $i+\frac{1}{2}$ indicates the average of the value at the beginning of a time interval (index i) and the value at the end of a time interval (index $i+1$). One special case must be noted. The predicted state $\tilde{x}$ at the beginning of a time interval is the same as the corrected state $\hat{x}$ at the end of the previous interval. Thus, $\tilde{x}(t_{i+\frac{1}{2}})$ is the average of $\hat{x}(t_i)$ and $\tilde{x}(t_{i+1})$ (not $\tilde{x}(t_i)$ and $\tilde{x}(t_{i+1})$). The same principle applies to $\dot{x}_c$.

Correction step—

$$\hat{x}_\xi(t_{i+1}) = \tilde{x}_\xi(t_{i+1}) + K\left[z(t_{i+1}) - \tilde{z}_\xi(t_{i+1})\right] \tag{43}$$

Gradients of matrices are involved in several of the equations below. The gradient of a matrix is a third order tensor. Formal tensor notation is avoided because the operations involved should be unambiguous without the multiplicity of indices formally needed. For instance, the term $\left(\nabla_\xi B\right)u$ represents the tensor inner product $\left(\nabla_\xi B\right)^i_{jk}u^j$. Those uncomfortable with tensors can substitute partial derivatives for the gradients. Each gradient equation then becomes a set of partial derivative equations involving no tensors. The first gradient of the filter equations is as follows:

Prediction step—

$$\begin{aligned}
\nabla_\xi \tilde{x}_\xi(t_{i+1}) &= \Phi \nabla_\xi \hat{x}_\xi(t_i) + \psi R^{-1}\Big[(\nabla_\xi A)\tilde{x}_\xi(t_{i+\frac{1}{2}}) + (\nabla_\xi B)u(t_{i+\frac{1}{2}}) \\
&\quad + (\nabla_\xi S)\nu - (\nabla_\xi R)\dot{x}_c(t_{i+\frac{1}{2}})\Big] \\
\nabla_\xi \tilde{z}_\xi(t_{i+1}) &= \left(ER^{-1}A + C\right)\nabla_\xi \tilde{x}_\xi(t_{i+1}) + \Big[(\nabla_\xi C)\tilde{x}(t_{i+1}) + (\nabla_\xi D)u(t_{i+1}) + (\nabla_\xi H)\nu + (\nabla_\xi E)\dot{x}_c(t_{i+1})\Big] \\
&\quad + ER^{-1}\Big[(\nabla_\xi A)\tilde{x}_\xi(t_{i+1}) + (\nabla_\xi B)u(t_{i+1}) + (\nabla_\xi S)\nu - (\nabla_\xi R)\dot{x}_c(t_{i+1})\Big]
\end{aligned} \tag{44}$$

where we have defined

$$\dot{x}_c = R^{-1}A\tilde{x}_\xi + R^{-1}Bu + R^{-1}S\nu + R^{-1}v \tag{45}$$

Correction step—

$$\nabla_\xi \hat{x}_\xi(t_{i+1}) = \nabla_\xi \tilde{x}_\xi(t_{i+1}) - K\nabla_\xi \tilde{z}_\xi(t_{i+1}) + (\nabla_\xi K)\Big[z(t_{i+1}) - \tilde{z}_\xi(t_{i+1})\Big] \tag{46}$$

where

$$\begin{aligned}
\nabla_\xi K &= (\nabla_\xi P)\left(ER^{-1}A + C\right)(GG^*)^{-1} \\
&\quad + P\Big[(\nabla_\xi C) + (\nabla_\xi E)R^{-1}A + ER^{-1}(\nabla_\xi A) - ER^{-1}(\nabla_\xi R)R^{-1}A\Big](GG^*)^{-1}
\end{aligned} \tag{47}$$

the gradient $\nabla_\xi P$ is computed as the solution to the set of Lyapunov equations

$$\mathcal{A}(\nabla_\xi P) + (\nabla_\xi P)\mathcal{A}^* = \mathcal{C} + \mathcal{C}^* \tag{48}$$

where

$$\mathcal{A} = R^{-1}A - \frac{1}{\Delta t}K\left(ER^{-1}A + C\right) \tag{49}$$

$$\begin{aligned}
\mathcal{C} &= \left(\frac{1}{\Delta t}KE - I\right)R^{-1}(\nabla_\xi A)P \\
&\quad + R^{-1}(\nabla_\xi R)\, R^{-1}F\,(R^{-1}F)^* - \left(\frac{1}{\Delta t}KE - I\right)R^{-1}(\nabla_\xi R)R^{-1}AP \\
&\quad + \frac{1}{\Delta t}K(\nabla_\xi C)P + \frac{1}{\Delta t}K(\nabla_\xi E)R^{-1}AP - R^{-1}(\nabla_\xi F)(R^{-1}F)^*
\end{aligned} \tag{50}$$

3.2 Files

The MMLE3 program uses several files for input and output. The general usage of these files is discussed in this section. Specific file formats are generally defined by the user and thus are not discussed here. The file formats defined by the standard aircraft routines are discussed in section 4.2.

To facilitate compatibility with different computer systems, the following program variables are used for all file numbers. The Programmer's Manual (ref. 38) describes how the values of these variables can be changed.

Variable Name	Description	Current Value
UCARD	Card reader	1
UPRINT	Line printer	3
UDATA	Input time history data	4
UWT	Predicted-derivative data	10
UPUNCH	Card punch	2
UTHOUT	Output time history data	9
UPLOT	Plot file (on some systems)	13
UT1	Temporary scratch file 1	7
UT2	Temporary scratch file 2	8

3.2.1 Card Reader and Line Printer Files

The card reader and line printer files are assigned FORTRAN device numbers UCARD and UPRINT, respectively. A detailed description of the data for the card reader file is given in section 3.3. The line printer output data are strongly dependent on the input data.

3.2.2 Input Time History File

The FORTRAN file number for the input time history file is UDATA. The format of this file can be defined by the user. READTH is the user routine to read the input time history file; no reference to the file is made in any other routine. READTH can be written to read the time history data from the card reader (sec. 3.3.13) or to compute simulated time history data; in these cases, file UDATA is not used. Various aspects of the data from the input time history file are described in section 3.3.8, items (2) to (9).

3.2.3 Predicted-Derivative File

The FORTRAN device number for the predicted-derivative file is UWT. This is a miscellaneous file which can be used for any special purpose required by the user. This file is not restricted to predicted-derivative tables; it can be used to store any data required by the user routines. The file can be used for input, output, or

scratch storage. Since the purpose of this file can be defined by the user, it can be referenced in any of the user routines. No reference to this file is made outside of the user routines.

3.2.4 Punch File

The FORTRAN device number for the punch file is UPUNCH. The format of this file can be defined by the user. Subroutine OUTPUN is the user routine to write the punch file; no reference to this file is made in any other routine. The information written on the punch file would normally include the final derivative estimates and Cramér-Rao bounds.

3.2.5 Output Time History File

The FORTRAN device number for the output time history file is UTHOUT. The format of this file can be defined by the user. Subroutine THOUT is the user routine to write the output time history file; no reference to this file is made in any other routine. Both the measured and last-iteration-predicted response time histories are available to subroutine THOUT, as well as the last-iteration-corrected state estimates and the control and extra signal time histories. Any of these signals, or data computed from them, can be written to the output time history file.

3.2.6 Plot File

The FORTRAN device number for the plot file is UPLOT. The software on some systems may ignore the device number and create a file with a special name defined by the system. System dependent details of plotting are discussed in the Programmer's Manual (ref. 38).

3.2.7 Internal Scratch Files

FORTRAN file numbers UT1 and UT2 are used for unformatted, internal scratch files. The user generally does not need to be concerned with these files, except possibly to describe their characteristics to the operating system. File UT1 is used during iteration for the measured time history data for a case. Scale factors and biases (sec. 3.3.8(8) and (9)) and any corrections made by subroutine THMOD have been applied to the data on this file. Each record of file UT1 contains LORD + 4 FORTRAN variables. File UT2 is used for measured and estimated time history data for plotting. Each record of file UT2 contains 2 X MAXZ + MAXX + MAXU + LEX FORTRAN variables.

3.3 Input Description

This section describes the card input required to run the basic program. Each subsection describes a group of input cards. The subsections are presented in the order in which the card groups appear in the input deck.

Subsections are included for input to the user routines. The input cards required for these subsections depend on the user routines; therefore, the details of such input are not documented in this section. For the basic program such cards are omitted. If user routines are called (sec. 3.3.2), any input required for the user routines should be placed in the locations indicated by these subsections. The special input requirements for the standard aircraft routines are described in section 4.3.

The syntax check card, user routines control card, user initialization input, predicted-derivative control card, and predicted-derivative input occur only once per program execution. The remaining input card groups are repeated for each new case. Any number of cases can be run in one program execution. The last case is signified by "END" on the endcase card.

The input flow chart on page 29 shows all of the input card groups and the order of their appearance. Each card group is denoted as basic program or user routine. Card groups which may be omitted under some circumstances are labeled as "optional." Of course, all of the user routine card groups are omitted if user routines are not used; furthermore, a specific set of user routines may require no cards in a particular group. A user routine card group is not marked as optional unless the basic program has an option to omit the card group while user routines are being used. Details are given in the appropriate subsections of this section.

3.3.1 Syntax Check Card

MMLE3 has an option to check the syntax of the input cards without executing the estimation. This option can be useful on runs with many cases, which will require large amounts of computer time. In order to activate this option, a card with "SYNTAX " in the first seven columns is added to the front of the deck. After all the syntax errors are found, the syntax check card should be removed, and the job resubmitted for actual execution.

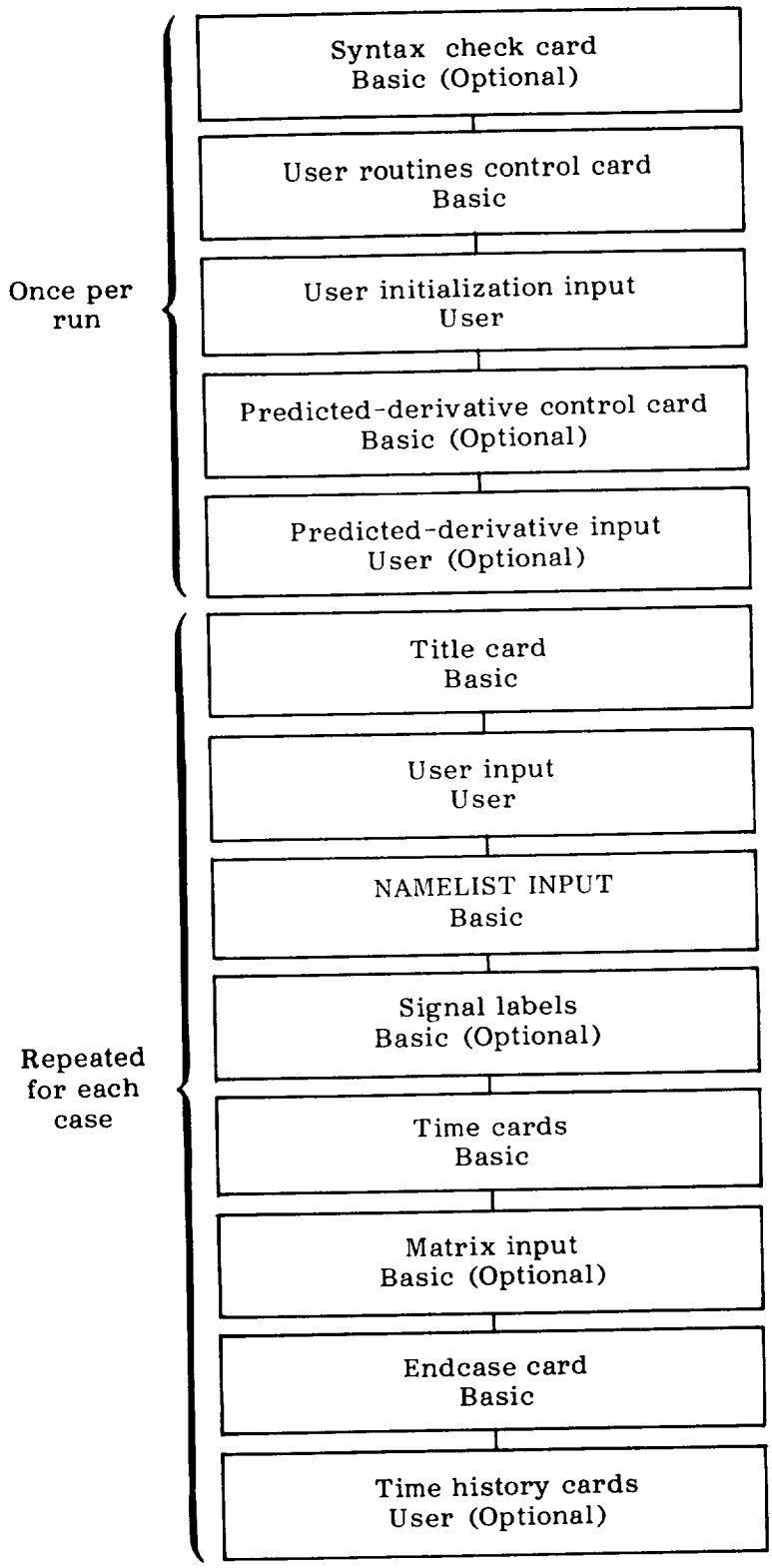
When the syntax check option is used, the input time history file is not used. Therefore, the input time history file (sec. 3.2.2) need not be available, and the time history card input (sec. 3.3.13) should be omitted.

If syntax checking is not desired, the input deck begins with the user routines control card.

3.3.2 User Routines Control Card

This card controls the calls to most of the user routines. Only the first three characters of the card are tested, but the entire card is printed on the output listing, so it may be used for comments on the listing. The card must not have "SYNTAX " in the first seven columns or it will be mistaken for a syntax check card (sec. 3.3.1). If the first three characters are "NO ", the user routines defining the equations of motion are bypassed and the card groups described in sections 4.3.3, 4.3.4, 4.3.5, and 4.3.7 should be omitted. In this case, the equations are defined solely by the matrices in the card input. The default values of nondimensional matrices (N matrices)

Input flow chart:



are all 0, except for RN which defaults to the identity; the dimensionalization ratio matrices (M matrices) are all 1's, and the dimensionalization addition matrices (L matrices) are all 0's; thus, the dimensional matrices are identical to the nondimensional matrices. The time-varying option (sec. 3.3.8 (20)) is meaningless when the user routines are bypassed; it will be overridden if attempted. The subroutines bypassed by this option are ONCE, WTIN, USERIN, AVERAG, MATDEF, MAKEL, MAKEM, and MAKEVW. The remaining six user routines are not affected. Subroutine READTH is always called when a case is executed, since the program must obtain a set of time histories. Subroutines THMOD, THOUT, and TITPLT are also always called, though they perform no functions inherently necessary to the execution of the program. The call to UINIT is controlled by the variable USERIC (sec. 3.3.8(26)) and the call to OUTPUN is controlled by the variable PUNCH (sec. 3.3.8(45)).

The NO option on the user routines control card is useful for simple systems or occasional runs of complicated systems where it is more convenient to read as input all of the relevant matrices than to code routines to define defaults.

If any first three characters other than "NO " are punched on the user routines control card, the user subroutines ONCE, WTIN, USERIN, AVERAG, MATDEF, MAKEL, MAKEM, and MAKEVW are called. The use of this option is suggested when many runs are made with the same or similar models. The user routines would be coded to define most of the features of the models used; input would only be required for those items in the models that change from run to run. For large or complicated models, the use of this option can considerably simplify the input requirements.

3.3.3 User Initialization Input

The user initialization input contains any input required by the user routine ONCE. If user routines are bypassed (sec. 3.3.2), the user initialization input should be omitted. Subroutine ONCE should perform any initialization or input required by the user which is not to be repeated for each case. Predicted-derivative input is not included here (see secs. 3.3.4 and 3.3.5). The input required by the standard aircraft subroutine ONCE is described in section 4.3.1.

3.3.4 Predicted-Derivative Control Card

This card controls the program call to the user routine WTIN. If user routines are bypassed (sec. 3.3.2), this card should be omitted. WTIN reads predicted-derivative data from cards and writes the predicted-derivative file. The first three characters of the predicted-derivative control card must be "NEW" or "OLD" or "NO " or "ONL" (for only); any other first three characters will result in an error message. The entire card is printed on the output listing, so the remaining 77 columns can be used for comments on the listing.

In addition to controlling the call to WTIN, the first three characters of this card determine the value of the logical variable WTFILE. WTFILE is intended to inform the user routines whether or not data are available on the predicted-derivative file.

If the first three characters "OLD" or "NO " are used, subroutine WTIN will not be called. With the NO option, WTFILE will be set to FALSE, indicating no data on the predicted-derivative file. With the OLD option, WTFILE will be set to TRUE, indicating data available on the predicted-derivative file (created by a previous job).

If the first three characters "NEW" are used, WTIN will be called and WTFILE will be set to TRUE. The user subroutine WTIN should create the predicted-derivative file.

If the first three characters "ONL" are used, WTIN will be the last subroutine called. After this call the program will stop. This option is used if WTIN is to create a predicted-derivative file to be saved for later use, but no cases are to be run at present.

3.3.5 Predicted-Derivative Input

The predicted-derivative input contains any input required by the user routine WTIN. The input required by the standard aircraft routine WTIN is described in section 4.3.2. If the user routines control card (sec. 3.3.2) specifies NO , WTIN is not called, so the predicted-derivative input should be omitted. If the user routines control card is anything else, the call to WTIN is controlled by the predicted-derivative control card (sec. 3.3.4). If the predicted-derivative control card specifies NO or OLD, WTIN is not called and the predicted-derivative input should be omitted; if the predicted-derivative control card is NEW or ONL, the input is included.

3.3.6 Title Card

The first card of each case is an 80-column title card. This card is used on page headings, plotted output, and often in the punched output created by user routine OUTPUN.

3.3.7 User Input

Any input required by the user routine USERIN should be inserted at this point in the input deck. The input required by the standard aircraft routine USERIN is described in section 4.3.3. If user routines are bypassed (sec. 3.3.2), USERIN will not be called and the user input should be omitted.

3.3.8 NAMELIST INPUT

The input for this section is in the form of a FORTRAN NAMELIST called INPUT. The FORTRAN manuals for specific computer systems should be consulted for the format required by these systems. The variables in the NAMELIST are listed below. Most of these variables are scalar with type real or integer depending on the standard FORTRAN type convention. A first letter of I, J, K, L, M, or N indicates an integer variable; any other first letter indicates a real variable. Vector variables and exceptions to the standard type convention are indicated parenthetically after the variable

names. Vector lengths are defined by the variables described in section 2. Some closely related variables are listed together under a single item number. The variables are grouped into those relating to the input (items (1) to (10)), the estimation process (items (11) to (27)), and the output (items (28) to (45)).

The default values of all of the variables are also listed. The default values can be altered by the user subroutine USERIN (unless user routines are bypassed (sec. 3.3.2)). The changes in default values made by the standard aircraft routines are listed in section 4.3. All values are reset to the defaults at the beginning of each case, regardless of any values used in previous cases.

(1) NCASE - number of maneuvers to be analyzed as a single case. If two or more maneuvers were performed at disjoint times, they may be analyzed together to obtain a single set of estimates. Program dimensions limit NCASE to 15 or less. One time card (sec. 3.3.10) is required for each maneuver. The use of multiple maneuvers in this manner presumes that the values of the unknown coefficients are the same for all of the maneuvers in the case. The default value of NCASE is 1.

(2) CARD, TAPE (logical) - input source for the time history data. These variables are passed to the user subroutine READTH to allow selection of alternate sources for the input time history data. Either CARD or TAPE may be set to TRUE. A third selection is possible by specifying both CARD and TAPE to be FALSE. Both cannot be TRUE; if CARD is TRUE, TAPE will be forced to FALSE. READTH can assign any data sources to the three available combinations; the sources need not correspond to the variable names. The READTH subroutine supplied reads from cards if CARD is TRUE, or from tape otherwise (see secs. 4.2.1 and 4.3.4 for the data formats). It ignores the TAPE variable. The default values are FALSE for CARD and TRUE for TAPE.

(3) THIN (integer) - thinning factor for input time histories. If THIN is 1, every time point is used. If THIN is 2, every second time point is used, and so forth. The default value is 1.

(4) SPS - sampling rate of the time histories before thinning (samples per second). The sampling rate of the thinned data is obtained by dividing SPS by THIN (item (3)). If SPS is 0, the sampling rate of the thinned data will be determined from the times of the thinned data using the following algorithm. The times of the first two thinned data points are subtracted and the difference is rounded to the nearest 5 milliseconds. The reciprocal of this rounded difference is used for the thinned sampling rate. The rounding process corrects for slippages of up to 2 milliseconds in the recorded times, as long as the actual thinned sample interval is a multiple of 5 milliseconds. If the thinned sample interval is not a multiple of 5 milliseconds, this algorithm will compute an incorrect sampling rate and thus SPS should be specified. The default value of SPS is 0.

(5) MAXREC - maximum number of input records. If user routine READTH is called more than MAXREC times for a single case, the program will terminate with an error message. This feature prevents infinite loops caused by bugs in READTH. The default value is 100,000.

(6) NREC - number of data words exclusive of time words in each input record. This variable is passed to user routine READTH to define the number of data words required at each time point. The maximum value of NREC is limited by program dimensions to 100. The normal default value for NREC is given by the variable LORD (sec. 2), currently 32. This default is changed by the standard aircraft routines.

(7) ZCHAN, UCHAN, EXCHAN (MAXZ, MAXU, and LEX word integer vectors, respectively) - channel numbers of the observations, controls, and extra signals, respectively. ZCHAN, UCHAN, and EXCHAN specify the channels of the signals in the record returned from user routine READTH. The values normally lie between 1 and NREC (item (6)). Values greater than NREC, but less than or equal to the dimension limit of 100, can be used to obtain data values of 0. The normal default values are increasing integers starting from 1 in the vector formed by concatenating ZCHAN, UCHAN, and EXCHAN (thus, the normal default values depend on the lengths of these vectors). The default is changed by the standard aircraft routines.

(8) ZSCALE, USCALE, EXSCAL (MAXZ, MAXU, and LEX word vectors, respectively) - scale factors for observations, controls, and extra signals, respectively. The measured data are multiplied by corresponding elements of ZSCALE, USCALE, or EXSCAL to compensate for known scaling or sign errors. This multiplication is done before any other operations on the data. The default values are all 1.

(9) ZBIAS, UBIAS, EXBIAS (MAXZ, MAXU, and LEX word vectors, respectively) - biases for the observations, controls, and extra signals, respectively. Elements of ZBIAS, UBIAS, or EXBIAS are added to corresponding measured data to compensate for known bias errors. These additions are performed after any scale or sign corrections (item (8)), but before any other operations on the data. The default values are all 0.

(10) RELAB (logical) - option to read labels. If RELAB is set to TRUE, labels for the observations, states, controls, and extra signals will be read as described in section 3.3.9. If RELAB is FALSE, the default labels used are Z*i*, X*i*, U*i*, or EX*i* for the *i*th observation, state, control, and extra signal, respectively. These default labels are changed by the standard aircraft routines. The default value of RELAB is FALSE.

(11) MX - length of the state vector. All matrix dimensions will be forced to conform with MX states. The maximum value for MX is given by the variable MAXX (sec. 2), currently 7. The minimum value for MX is 1. If MX is specified as negative, it will be obtained from the number of rows of the AN matrix (sec. 3.3.11). The standard aircraft routines define a default number of rows of AN (sec. 4.3.5). The default value for MX is -1.

(12) MZ - length of the observation vector. All matrix dimensions will be forced to conform with MZ observations. The maximum value for MZ is given by the variable MAXZ (sec. 2), currently 8. The minimum value for MZ is 1. If MZ is specified as negative, it will be obtained from the number of rows of the GGI matrix (sec. 3.3.11). The standard aircraft routines define a default number of rows of GGI (sec. 4.3.5). The default value for MZ is -1.

(13) MU - length of the control vector. All matrix dimensions will be forced to conform with MU controls. The maximum value for MU is given by the variable MAXU (sec. 2), currently 4. The minimum value of MU is 1. If MU is specified as negative, it will be obtained from the number of columns of the BN matrix (sec. 3.3.11). If this number is 0, MU will be set to the largest column number that contains a nonzero element in BN, DN, BV, or DV. The standard aircraft routines define a default number of columns of BN. The default value for MU is -1.

(14) MB - length of the bias vector. All matrix dimensions will be forced to conform with MB elements in the bias vector (sec. 3.1). The maximum value for MB is given by the variable MAXB (sec. 2), currently 4. The minimum value of MB is 1. If MB is specified as negative, it will be obtained from the number of columns of the SN matrix (sec. 3.3.11). If this number is 0, MB will be set to the minimum of NCASE (item (1)) or MAXB (sec. 2). MB is forced to be less than or equal to NCASE. The default value for MB is -1.

(15) NEAT - number of time reductions in the computation of the transition matrix and its integral. The transition matrix is $e^{\tilde{A}\Delta t}$; where, in general, $\tilde{A} = R^{-1}A$. The direct series evaluation of $e^{\tilde{A}\Delta t}$ may become computationally unstable for low sample rates. In such cases, $e^{\tilde{A}\frac{\Delta t}{2^{NEAT}}}$ and its integral are first evaluated using 10 terms of the power series. The desired transition matrices are then obtained by recursive applications of the formulae

$$e^{\tilde{A}t} = \left(e^{\tilde{A}\frac{t}{2}}\right)^2 \tag{51}$$

$$\int_0^t e^{\tilde{A}s}ds = \left(e^{\tilde{A}\frac{t}{2}} + I\right)\int_0^{\frac{t}{2}} e^{\tilde{A}s}ds \tag{52}$$

This process provides improved computational stability without significantly increasing complexity or computer time (ref. 50). NEAT = 0 implies direct series computation. In general, NEAT should be large enough so that magnitudes of the eigenvalues of $\tilde{A}\frac{\Delta t}{2^{NEAT}}$ are less than 0.1. The default value of NEAT is 0.

(16) NOITER - maximum total number of iterations. The convergence criterion specified by BOUND (item (17)) or the error criterion specified by ERRMAX (item (18)) can cause iteration to terminate early; otherwise, NOITER iterations will be done. If NOITER is 0, the parameter identification step is skipped entirely and the program computes the final time histories using the initial coefficient estimates. Program dimensions limit NOITER to 49 or less. The default value for NOITER is 6.

(17) BOUND - convergence bound. If for any iteration, the error sum changes by less than BOUND times the error sum of the previous iteration, the algorithm is assumed to have converged. One of three actions is then taken, depending on ITAPR (item (22)) and ITG (item (24)) status: *a priori* is turned off, G determination is started, or iteration is stopped. Convergence may occur several times in one case for different *a priori* and G determination conditions. If the FULL1 option (item (19)) is FALSE, the bound will not be checked on the first iteration. BOUND is also not checked on iterations that use DFAC (item (21)). The default value of BOUND is 0.001.

(18) ERRMAX - maximum allowable error sum. If at any time the error sum becomes greater than ERRMAX, it is assumed that the algorithm is not converging properly. Iteration will then terminate and, depending on ERRTH (item (32)), the measured time histories might be printed to provide debugging clues. ERRMAX should be less than the largest single-precision floating-point number defined on the computer divided by the total number of time points in the case. If ERRMAX is less than PLTMAX (item (34)) it will be set equal to PLTMAX. The default value of ERRMAX is 10^{20}.

(19) FULL1 (logical) - full first iteration option. If FULL1 is FALSE, only initial conditions and B, S, D, and H terms are determined in the first iteration. This eliminates convergence problems caused by large initial bias errors. Since these terms affect the response in a linear manner, the Newton-Balakrishnan algorithm converges in one iteration if they are the only unknowns. If FULL1 is true, the first iteration is full normal iteration. The default value of FULL1 is FALSE.

(20) TIMVAR (logical) - time-varying option. If TIMVAR is true, the M and L dimensionalization matrices will be recomputed at each time point. The time-varying option is only partially implemented for the state noise case in that K and its gradients are considered time invariant; time-varying state noise cases should, therefore, be critically examined. If user routines are bypassed (sec. 3.3.2), the time-varying option is meaningless and will be overridden if attempted. USERIC (item (26)) is sometimes required in time-varying cases, depending on the system model. Use of the time-varying option should be avoided if possible because it increases the computer time used by approximately a factor of three. The default for TIMVAR is FALSE.

(21) DFAC, ITDFAC - diagonal convergence factor option. The diagonal elements of the second gradient matrix will be multiplied by DFAC before computing parameter changes for ITDFAC iterations. This option starts when *a priori* is turned off if ITAPR (item (22)) is not 0. If ITAPR is 0, the DFAC option starts on iteration 2 (iteration 1 if FULL1 (item (19)) is TRUE). NOITER (item (16)) should usually be larger than ITDFAC so that the final iterations will be done without the diagonal convergence factor (see sec. 1.2.2). This option sometimes improves the convergence properties of marginally stable maneuvers or cases where the starting estimates are far from the correct values. Convergence using this option tends to be monotone, but very slow. Because of the slow convergence, the use of this option is not recommended unless convergence is not achieved with the standard algorithm. The convergence check (item (17)) is not used on iterations with DFAC. The minimum value of DFAC is 1. Very small increases in DFAC are usually sufficient to affect convergence significantly. Typically used values range between 1.000001 and 1.1. The default values are 1.01 for DFAC, and 0 for ITDFAC.

(22) WAPR, ITAPR - *a priori* control variables. WAPR is the overall weighting factor for *a priori* information. The *a priori* term in the cost functional (sec. 1.3) is proportional to WAPR. A value of 0 for WAPR implies that the *a priori* feature is not used. If WAPR is 0, ITAPR is also forced to 0. ITAPR is the maximum number of iterations for the *a priori* option. If ITAPR is 0, the *a priori* weighting WAPR is used for all iterations. If ITAPR is not 0, *a priori* is used only for ITAPR iteration or until convergence (item (17)). Iteration then continues without *a priori*. The default value of WAPR is 0, and the default value of ITAPR is 0.

(23) FREQCR - break frequency for first order residual filter (Hz). The power of the filtered residual for each observation is printed out each iteration. The sum of the filtered residual powers, weighted by GGI, is used to adjust the Cramér-Rao bounds (sec. 1.4). If FREQCR is 0, the residuals are not filtered and unfiltered residuals are used for the Cramér-Rao bound adjustment. The default value of FREQCR is 0.

(24) ITG, RLXG - *G* determination variables. If ITG is not 0, *G* determination will start after ITG normal iterations or convergence (item (17)), whichever occurs first. Normal iterations are defined as iterations without any iteration-dependent options. Iteration-dependent options include DFAC (item (21)) and the linear-terms-only option (item (19)). If ITAPR (item (22)) is nonzero, then *a priori* is considered an iteration-dependent option. If BOUND (item (17)) indicates convergence after *G* determination has started, iteration terminates. If ITG is 0, *G* determination is not used. RLXG is a relaxation factor used in the *G* determination. The default values are 0 for ITG and 1.0 for RLXG.

(25) DIAGG (logical) - diagonal *G* matrix option. If *G* determination (item (24)) is not used, DIAGG is ignored. If *G* determination is used, DIAGG specifies whether a diagonal or full matrix will be determined. A value of TRUE for DIAGG specifies determination of a diagonal *G* matrix. If a nondiagonal *G* matrix is used for a starting estimate, DIAGG is forced to FALSE (the printout of the options may be incorrect in this case, because the printout precedes the matrix input). The default for DIAGG is TRUE.

(26) USERIC (logical) - user initial condition option. If USERIC is TRUE, user routine UINIT will be called to determine the initial conditions and fixed biases for each maneuver (item (1)). Some implications of the use of UINIT are discussed in section 3.1. UINIT should define the initial state, the initial control, and an observation bias to be added to the computed observations. The program will compute a measured control bias by subtracting the measured initial control from that specified by UINIT. This control bias will be added to the measured controls for each time point. USERIC is sometimes necessary when TIMVAR (item (20)) is used. If USERIC is FALSE, the initial state and control are set to 0 and the observation bias is set to the initial measured observation vector. The program prints out the initial state and control vectors used for each maneuver. The control time histories printed and plotted are the measured controls without any biases added. The default value for USERIC is FALSE. This default is changed by the standard aircraft routines.

(27) VARIC (MAXX word logical vector) - variable initial condition increments for states. For each element of VARIC that is TRUE, an increment to the initial condition of the corresponding state is estimated as one of the unknowns. This increment is added to any initial condition defined by user routine UINIT to obtain the initial condition used to compute the time histories. If USERIC (item (26)) is FALSE, the increments are added to the value 0, since UINIT is not called. If VARIC is used in conjunction with NCASE > 1 (item (1)), the same increment from the values defined by UINIT (or 0) is used for each maneuver. Caution must be taken with the use of VARIC because variable initial conditions are often equivalent to combinations of variables in the SN and HN matrices, resulting in identifiability problems. The default values for VARIC are all FALSE.

(28) TEST (logical) - extra output for debugging. If TEST is TRUE, extra output useful for debugging purposes is printed. Internal location maps are printed below the list of unknowns, and several matrices are printed each iteration. The matrices printed include the following: A, B, S, R, C, D, H, E, P, K, RIA = $R^{-1}A$, RIB = $R^{-1}B$, RIS = $R^{-1}S$, ERIAC = $ER^{-1}A + C$, ERIBD = $ER^{-1}B + D$, ERISH = $ER^{-1}S + H$, PHI = $e^{R^{-1}A\Delta t}$, PSI = ψ, PSIB = $\psi R^{-1}B$, PSIS = $\psi R^{-1}S$, and DK = $\nabla_\xi K$, where $\psi = \int_0^{\Delta t} e^{R^{-1}A}$ sds. The SUM matrix requires special mention. The lower triangular part of SUM contains the second gradient of the cost function. The upper triangular part of this symmetric matrix is not stored. The first gradient is stored and printed as an additional column of the SUM matrix. The gradients printed do not include terms attributable to *a priori* and are not multiplied by DFAC (item (21)). The default value of TEST is FALSE.

(29) PRINTI (logical) - option to print input time history. If PRINTI is TRUE, the measured time histories of the observations, controls, and extra signals are printed. All scale factors (item (8)), biases (item (9)), and modifications by user routine THMOD are applied before the data are printed. The default value of PRINTI is FALSE.

(30) PRINTO (logical) - option to print final output time history. If PRINTO is TRUE, the time histories of the corrected observation estimate are printed in the final iteration. PRINTO is irrelevant if PRINTY (item (31)) is TRUE. The default value of PRINTO is FALSE.

(31) PRINTX, PRINTY (logical) - options to print states and observations for all iterations. If PRINTX or PRINTY is TRUE, time histories of the predicted state estimates or corrected observation estimates, respectively, will be printed each iteration. If both PRINTX and PRINTY are TRUE, the predicted state at each time point will be printed on one line and the corrected observation on the next line. The printout columns of the states and observations are staggered so they can be easily distinguished. The default values of PRINTX and PRINTY are FALSE.

(32) ERRTH (logical) - option for time history printout after errors. If ERRTH is TRUE and a case terminates as a result of exceeding the ERRMAX (item (18)) or

PLTMAX (item (34)) bounds, the measured observations, controls, and extra signals will be printed. This printout provides a check to see if any obvious data problems are present. If PRINTI (item (29)) is TRUE, ERRTH is forced to FALSE since the printout would be redundant in this case. The default value of ERRTH is FALSE.

(33) PLOTEM (logical) - time history plot option. If PLOTEM is TRUE, time history plots comparing measured and predicted time histories are produced unless the ERRMAX (item (18)) or PLTMAX (item (34)) bounds are exceeded. The default value of PLOTEM is TRUE.

(34) PLTMAX - maximum error for plotting. If the final error sum is greater than PLTMAX, plots will not be produced regardless of PLOTEM (item (33)). In this case, the fits will probably be uninformative and may exceed reasonable plotter limits. If ERRTH (item (32)) is TRUE, and PLTMAX is exceeded, measured time histories will be printed as a diagnostic aid. The default value for PLTMAX is 10^5.

(35) XPLOT (MAXX word logical vector) - states to be plotted. For each element of XPLOT that has the value TRUE, the corresponding, corrected state estimate time history will be plotted, provided PLOTEM (item (33)) is TRUE. The default values for XPLOT are all FALSE.

(36) NUPLT - number of controls to be plotted. The first NUPLT control time histories will be plotted if PLOTEM (item (33)) is TRUE. The program limits the maximum value of NUPLT to MAXU (sec. 2), currently 4. The minimum value is 0. The default value for NUPLT is MAXU.

(37) NEXPLT - number of extra signals to be plotted. The first NEXPLT extra signal time histories will be plotted if PLOTEM (item (33)) is TRUE. The program limits the maximum value of NEXPLT to LEX (sec. 2), currently 20. The minimum value is 0. The default value of NEXPLT is 0.

(38) INCH (logical), PLTFAC - plot sizing controls. If PLTFAC is 1 and INCH is TRUE, plots will be sized for inch grid paper. If PLTFAC is 1 and INCH is FALSE, plots are sized for centimeter grid paper. Other values of PLTFAC multiply the plot size proportionally. The default value is 1 for PLTFAC, and FALSE for INCH.

(39) PAPER - paper grid width (centimeters or half inches). Plots will be scaled to fit on paper with a grid width specified by PAPER. The units for PAPER depend on INCH (item (38)). The default value for PAPER is 25 (centimeters) if INCH is FALSE, or 20 (half inches) if INCH is TRUE.

(40) TIMESC - minimum time scale for plots (seconds per centimeter or seconds per half inch). The program automatically increases the time scale above the value TIMESC as necessary to fit the plots on the paper size specified by PAPER (item (39)). The length available for the time axis is PAPER - 2 (centimeters or half inches). The units of TIMESC depend on INCH (item (38)). The default value of TIMESC is 0.5.

(41) ZMIN, ZMAX (MAXZ word vectors) - minimum and maximum ordinate values for observation time history plots. The ordinate axes are 4 centimeters or 4 half inches long, depending on INCH (item (38)). If corresponding elements of

ZMIN and ZMAX are equal for any signal, automatic scaling will be used for that signal. The automatic scaling algorithm used does not force 0 to be included in the scale. If automatic scaling is used for some signal and both the measured and computed time histories of that signal have the same constant value for an entire maneuver (item (1)), the plot for that signal will be omitted for that maneuver. The default values for ZMIN and ZMAX are all 0, implying automatic scaling.

(42) UMIN, UMAX (MAXU word vectors) - minimum and maximum ordinate values for control time history plots. The ordinate axes are 4 centimeters or 4 half inches long, depending on INCH (item (38)). If corresponding elements of UMIN and UMAX are equal for any signal, automatic scaling will be used for that signal. The automatic scaling algorithm used forces 0 to be included in the scale. If automatic scaling is used for some signal and the time history of that signal is 0 for an entire maneuver (item (1)), the plot for that signal will be omitted for that maneuver. The default values for UMIN and UMAX are all 0, implying automatic scaling.

(43) XMIN, XMAX (MAXX word vectors) - minimum and maximum ordinate values for state time history plots. The interpretations of XMIN and XMAX are identical to those of UMIN and UMAX (item (42)). The default values of XMIN and XMAX are all 0, implying automatic scaling.

(44) EXMIN, EXMAX (LEX word vectors) - minimum and maximum ordinate values for extra signal time history plots. The interpretations of EXMIN and EXMAX are identical to those of UMIN and UMAX (item (42)). The default values of EXMIN and EXMAX are all 0, implying automatic scaling.

(45) PUNCH (logical) - punched output option. If PUNCH is TRUE, user routine OUTPUN will be called to write data on the punch file (sec. 3.2.4). If NOITER (item (16)) is 0, PUNCH is forced to FALSE. If the ERRMAX limit (item (18)) is exceeded, OUTPUN will be skipped, regardless of the value of PUNCH. The format of the data written is defined by the user. The default value of PUNCH is FALSE.

3.3.9 Signal Labels

The cards described in this section contain the labels to be used for the observations, states, controls, and extra signals. Signal labels are included only if RELAB (sec. 3.3.8(10)) is set to TRUE. If RELAB is FALSE, the default labels used are Z*i*, X*i*, U*i*, or EX*i* for the *i*th observation, state, control, or extra signal, respectively. These default labels are changed by the standard aircraft routines. The default labels defined by the standard aircraft routines are specified in section 4.3.6.

If RELAB is TRUE, all of the labels must be read; there is no provision for redefining only a portion of the labels. The labels are read, left justified in fields of 10, up to 8 labels per card. Each label is limited to eight characters. The labels for MAXZ observations are read from the first card(s), MAXX states from the next card(s), MAXU controls from the following card(s), and LEX extra signals from the last card(s).

3.3.10 Time Cards

One time card is required for each of the NCASE (sec. 3.3.8(1)) maneuvers to be analyzed. The time cards contain the start and end times for each maneuver expressed as hours, minutes, seconds, and milliseconds in the format (2(3I2, I3, 1X)).

At least one data point with time greater than or equal to the end time must be read from the input file in order for the program to detect the end of the maneuver; end-of-file errors can result from neglecting this fact (the program has no end-of-file checks because such checks are system dependent).

3.3.11 Matrix Input

This section describes the input of the data matrices. Any number (including 0) of the matrices listed below can be read in any order. Each of the matrices has a default value which is used if that matrix is not read in. Any of these defaults can be changed by user routine MATDEF unless user routines are bypassed (sec. 3.3.2). The default values used by the standard aircraft routines are specified in section 4.3.5. All values are reset to the defaults at the beginning of each case, regardless of any values used in previous cases.

The same format is used for all of the matrices, except for the HARD and SOFT matrices (the format for HARD and SOFT is described in item (6)). Each matrix is preceded by a header card with the matrix name, number of rows, and number of columns in (A4, I6, I10) format. The matrix name always starts in column 1 of the card. The body of the matrix follows, one row to a card, in 8F10 format. A diagonal matrix can be input in a more compact form by specifying 0 as the number of columns; the diagonal elements are then read from one card in 8F10 format.

(1) AN, BN, SN, RN, CN, DN, HN, EN - nondimensional starting matrices. The default values of these matrices are all 0, except for RN, which defaults to an identity matrix. The default matrix sizes are all set to 0. The sizes of the AN, BN, and SN input matrices are sometimes used to determine MX, MU, and MB (sec. 3.3.8(11), (13), and (14)). All matrix sizes are adjusted by the program to be consistent with MX, MU, MB, and MZ (sec. 3.3.8(12)). Maximum matrix sizes are discussed in section 2.

(2) F - state noise power spectral density matrix. The default value for F is 0.

(3) AV, BV, SV, RV, CV, DV, HV, EV, FV - variation matrices. These matrices indicate which elements in the nondimensional matrices (item (1)) and F (item (2)) are allowed to vary.

A value of 1 indicates that the corresponding nondimensional or F matrix element is allowed to vary independently; a value of 0 indicates that it is held fixed or is constrained to another element (see HARD, item (6)). The defaults for the variation matrices are all 0.

(4) APRA, APRB, APRS, APRR, APRC, APRD, APRH, APRE, APRF - *a priori* weighting matrices. These matrices contain relative weighting data for the *a priori* feature. The value of each element in the *a priori* weighting matrices should be proportional to the inverse of the *a priori* standard deviation of the corresponding nondimensional coefficient.

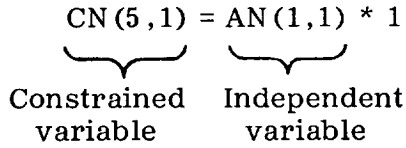
A priori weightings are only implemented for independent unknowns; that is, matrix locations corresponding to values of 1 in the variation matrices (item (3)). *A priori* weightings on dependent unknowns in hard constraints (item (6)) are ignored.

The overall *a priori* weighting is adjusted by WAPR (sec. 3.3.8(22)). The actual *a priori* weighting for each coefficient is WAPR times the square of the corresponding APR element. The default values for the *a priori* weighting matrices are all 0.

(5) GGI - inverse of the residual covariance matrix. The GGI matrix is often diagonal. If the GGI used is diagonal, the program will take advantage of the diagonal condition to reduce computation time. If GGI is not symmetric, an error message will be printed and the upper triangular part will be replaced by the transpose of the lower triangular part to force symmetry. The size of the GGI input matrix is sometimes used to determine MZ (sec. 3.3.8(12)). Maximum sizes are discussed in section 2. The default value for GGI is the 0 matrix.

(6) HARD - hard constraints. Constraint "matrices" are read in the special form described here. The header card contains the "matrix" name (HARD) as usual, and the number of constraints replaces the number of rows. A special provision is made to control whether constraints from cards replace or supplement any default constraints. If any nonzero value is punched for the number of columns on the header card of the HARD matrix, the default constraints will be used in addition to the ones read in. If the number of columns is left blank on the header card of the HARD matrix input, the constraints read in will replace the default constraints.

The constraint instructions follow the header card, one to a card in the format of the sample below:



Valid matrix names here are AN, BN, SN, RN, CN, DN, HN, and EN. The names of the constrained variable and the independent variable start in columns 1 and 11, respectively. Either one or two digits can be used for the row and column numbers. The constraint ratio is in columns 21 to 30. The constraint ratio in this example is 1.

For a hard constraint, the constrained variable should *not* be specified as varying in the variation matrices (item (3)), else the constraint will not work properly—the variation matrices define independently varying elements only. Conversely, the independent variable must be specified as varying in the variation matrices, else the constraint will be ignored as immaterial. Thus, hard constraints may not be chained—*A* constrained to *B*, constrained to *C*—as *B* is not independently varying. However, an equivalent form can be obtained by constraining *A* and *B* to *C*.

It is extremely important to note that only the *change* from the starting value is constrained; the starting values may or may not satisfy the constraint. It is solely the user's responsibility to enter the desired starting value. As a consequence of this formulation, it is valid to constrain one variable to several independent variables. In this case, the changes implied by each constraint are added.

If the constraint ratio is entered as 0, it will be automatically computed as the ratio of the starting values (if the independent variable has a starting value of 0, a warning message is printed and a ratio of 1 is used). The maximum number of hard constraints is MAXHRD - 1 (sec. 2). The default uses no hard constraints.

(7) SOFT - soft constraints. The input format for soft constraints is the same as that for hard constraints, except that the matrix name on the header card is SOFT. For a soft constraint, both variables must be specified as independently varying by the variation matrices (item (3)) or the constraint will be ignored as immaterial (in specific, neither may be a dependent variable in a hard constraint). The soft constraint means simply that *a priori* weighting will penalize the constrained variable for departure from the constraint instead of from its initial condition. Thus, soft constraints are immaterial if *a priori* weighting is not used or if the constrained variable has an *a priori* weighting of 0. Multiple or chained soft constraints are allowed in any combination. The maximum number of soft constraints is MAXSFT - 1 (sec. 2). The default uses no soft constraints.

3.3.12 Endcase Card

The end of the matrix input section is signaled by a card with "END" or "ENDCASE" starting in column 1. "ENDCASE" indicates that more cases follow. "END" is used for the last case to be processed.

3.3.13 Time History Cards

If the user routine READTH reads the time history data from cards, these cards should be placed in this location. Various aspects of the time history data are described in section 3.3.8, items (2) to (9). The card format required by the standard aircraft routine READTH is described in section 4.3.4. All of the time history data cards should be read by READTH if more cases follow; otherwise, the first unread card will be interpreted as the title card for the next case.

4.0 STANDARD AIRCRAFT ROUTINES

This section describes a particular set of user routines referred to as the standard aircraft routines. These routines are intended for the aircraft longitudinal or lateral-directional stability and control problem with no turbulence.

4.1 Equations of Motion

4.1.1 Nonlinear Six-Degree-of-Freedom Equations

This section presents the nonlinear aircraft equations of motion from which the linearized equations used in MMLE3 are derived. The coupled, nonlinear equations are presented first. These equations assume a rigid vehicle and a flat, nonrotating earth. The time rate of change of mass is assumed negligible, and fuel-sloshing effects are ignored. No small angle approximations are used, but the absolute values of β and θ must be less than 90°. The aircraft velocity must not be 0. No symmetry assumptions are made. Engine inertia and thrust terms are included, assuming the engine alignment and thrust vector are along the X-axis. The equations are written in body axes referenced to the center of gravity. More complete forms of the equations of motion and transformations are found in reference 51. All angles are in degrees. The $\dot{V}$ equation is not included.

$$\dot{\alpha} = q - \tan\beta\,(p\cos\alpha + r\sin\alpha) - \frac{\bar{q}s}{mV\cos\beta}\mathscr{R}C_L + \frac{g}{V\cos\beta}\mathscr{R}\left(\cos\theta\cos\varphi\cos\alpha + \sin\theta\sin\alpha - \frac{T}{mg}\sin\alpha\right)$$

$$\dot{\beta} = p\sin\alpha - r\cos\alpha + \mathscr{R}\cos\beta\left(\frac{\bar{q}s}{mV}C_Y + \frac{g}{V}\cos\theta\sin\varphi\right) + \mathscr{R}\sin\beta\left[\frac{\bar{q}s}{mV}C_D - \frac{g}{V}\left(\cos\theta\cos\varphi\sin\alpha - \sin\theta\cos\alpha + \frac{T}{mg}\cos\alpha\right)\right]$$

$$\dot{p}I_x - \dot{q}I_{xy} - \dot{r}I_{xz} = \bar{q}sb\mathscr{R}C_\ell + \frac{1}{\mathscr{R}}\left[qr\left(I_y - I_z\right) + \left(q^2 - r^2\right)I_{yz} + pqI_{xz} - rpI_{xy}\right] \tag{53}$$

$$\dot{q}I_y - \dot{r}I_{yz} - \dot{p}I_{xy} = \bar{q}sc\mathscr{R}C_m + \frac{1}{\mathscr{R}}\left[rp\left(I_z - I_x\right) + \left(r^2 - p^2\right)I_{xz} + qrI_{xy} - pqI_{yz} + 6NrI_{xe}\right]$$

$$\dot{r}I_z - \dot{p}I_{xz} - \dot{q}I_{yz} = \bar{q}sb\mathscr{R}C_n + \frac{1}{\mathscr{R}}\left[pq\left(I_x - I_y\right) + \left(p^2 - q^2\right)I_{xy} + prI_{yz} - qrI_{xz} - 6NqI_{xe}\right]$$

$$\dot{\theta} = q\cos\varphi - r\sin\varphi$$

$$\dot{\varphi} = p + r\cos\varphi\tan\theta + q\sin\varphi\tan\theta$$

4.1.2 Uncoupled Linearized Equations

In order to divide the equations into longitudinal and lateral-directional sets, small angle approximations are needed for β. The sin β term in the $\dot{\beta}$ equation is ignored, and cos β is replaced by 1 in the $\dot{\alpha}$ and $\dot{\beta}$ equations. The tan β in the $\dot{\alpha}$ equation is replaced by $\beta/\mathscr{R}$ (it is not ignored because it is multiplied by p, which can be quite large). Symmetry about the XZ-plane is assumed, so I_{yz} and I_{xy} are 0. The remaining cross-coupling terms are computed using measured data; the assumptions inherent in this usage are discussed in reference 26. Whenever measured α and β are used, they are corrected for upwash and center of gravity offset.

Measured data are also used to linearize the other nonlinear terms in the equations. There is some leeway for engineering judgment in the use of the measured data. Strictly speaking, the linearized longitudinal equations should be obtained by a first-order expansion about the measured longitudinal data—similarly for lateral-directional equations. However, many of the nonlinear terms are satisfactory when simply evaluated at the measured data (that is, using a zero-order expansion). A value judgment must be made for each nonlinear term to decide whether the additional fidelity of the first-order expansion justifies the additional complexity. The implementation in the standard aircraft routines represents the authors' judgment of these tradeoffs; modification for other choices is relatively simple.

4.1.2.1 Longitudinal. - The longitudinal state, control, observation, and extra signal vectors, respectively, are

$$\begin{aligned} x &= (\alpha \;\; q \;\; \theta)^* \\ u &= \left(\delta_e \;\; \delta_c \;\; \delta_1 \;\; \delta_2\right)^* \\ z &= \left(\alpha_m \;\; q_m \;\; \theta_m \;\; a_{n_m} \;\; a_{x_m} \;\; \dot{q}_m\right)^* \end{aligned} \tag{54}$$

$$\text{extra} = (\bar{q} \;\; \beta \;\; p \;\; r \;\; \varphi \;\; M \;\; h \;\; V \;\; \dot{p} \;\; \dot{r} \;\; N \;\; T)^*$$

The nonlinear longitudinal state equations are

$$\begin{aligned} \dot{\alpha} &= -\frac{\bar{q}s}{mV}\mathscr{R}\left(C_L + \dot{\alpha}_0\right) + q + \frac{g}{V}\mathscr{R}(\cos\theta\cos\varphi\cos\alpha + \sin\alpha\sin\theta) - \frac{\beta}{\mathscr{R}}(p\cos\alpha + r\sin\alpha) - \frac{T}{mV}\mathscr{R}\sin\alpha \\ I_y\dot{q} &= \bar{q}sc\mathscr{R}C_m + \frac{rp}{\mathscr{R}}\left(I_z - I_x\right) + \left(r^2 - p^2\right)\frac{I_{xz}}{\mathscr{R}} + \frac{6Nr}{\mathscr{R}}I_{xe} \\ \dot{\theta} &= q\cos\varphi - r\sin\varphi + \dot{\theta}_0 \end{aligned} \tag{55}$$

The $\dot{\alpha}_0$ and $\dot{\theta}_0$ are included to allow for instrument biases. The $\dot{q}$ equation already has the freedom to correct such biases using C_{m_0}; the C_{L_0} in the $\dot{\alpha}$ equation does not allow similar freedom because it is related to the C_N and C_A terms in the observation equations below. Measured data are used for β, p, r, and φ in these equations. Measured α is also used wherever α appears explicitly above. The gravity term in the $\dot{\alpha}$ equation is normally linearized about the measured θ; however, it can be evaluated at the measured θ if it is not desired to integrate the θ equation.

The longitudinal observation equations are

$$
\begin{aligned}
\alpha_m &= K_\alpha\left(\alpha - \frac{x_\alpha}{V}q + \frac{y_\alpha}{V}p\right) \\
q_m &= q \\
\theta_m &= \theta \\
a_{n_m} &= \frac{\bar{q}s}{mg}C_N + \frac{x_{a_n}}{g\mathscr{R}}\dot{q} + \frac{z_{a_n}}{\mathscr{R}^2 g}\left(q^2 + p^2\right) - \frac{y_{a_n}}{g\mathscr{R}}\dot{p} \\
a_{x_m} &= -\frac{\bar{q}s}{mg}C_A + \frac{z_{a_x}}{g\mathscr{R}}\dot{q} - \frac{x_{a_x}}{\mathscr{R}^2 g}\left(q^2 + r^2\right) - \frac{y_{a_x}}{g\mathscr{R}}\dot{r} + \frac{T}{mg} \\
\dot{q}_m &= \dot{q} + \dot{q}_0
\end{aligned}
\tag{56}
$$

The $\dot{q}_0$ is the instrument bias on $\dot{q}$. Biases on a_n and a_x are included in C_{n_0} and C_{A_0}; those in α, q, and θ were handled in the state equations. Measured data are used for p, r, $\dot{p}$, and $\dot{r}$ in these equations. The terms involving q^2 are also evaluated with measured data to eliminate the nonlinearity.

The expansions of the longitudinal force and moment coefficients are

$$
\begin{aligned}
C_N &= C_{N_\alpha}\alpha + C_{N_q}\frac{qc}{2V\mathscr{R}} + C_{N_\delta}\delta + C_{N_0} \\
C_m &= C_{m_\alpha}\alpha + C_{m_q}\frac{qc}{2V\mathscr{R}} + C_{m_\delta}\delta + C_{m_0} + C_{m_{\dot{\alpha}}}\frac{\dot{\alpha}c}{2V\mathscr{R}} \\
C_A &= C_{A_\alpha}\alpha + C_{A_q}\frac{qc}{2V\mathscr{R}} + C_{A_\delta}\delta + C_{A_0} \\
C_L &= C_N \cos\alpha - C_A \sin\alpha
\end{aligned}
\tag{57}
$$

where the δ term is summed over all controls. The equation for C_L is linearized about the average measured α. If MZ (sec. 3.3.8(12)) is 4, the approximation $C_L = C_N$ is used. This approximation is good for low angles of attack.

4.1.2.2 Lateral-directional. - The lateral-directional state, control, observation, and extra signal vectors, respectively, are

$$
\begin{aligned}
x &= (\beta \;\; p \;\; r \;\; \varphi)^* \\
u &= \left(\delta_a \;\; \delta_r \;\; \delta_1 \;\; \delta_2\right)^* \\
z &= \left(\beta_m \;\; p_m \;\; r_m \;\; \varphi_m \;\; a_{y_m} \;\; \dot{p}_m \;\; \dot{r}_m\right)^* \\
\text{extra} &= \left(\bar{q} \;\; \alpha \;\; q \;\; a_n \;\; \theta \;\; M \;\; h \;\; V \;\; \dot{q} \;\; a_x \;\; N \;\; T\right)^*
\end{aligned}
\tag{58}
$$

The nonlinear lateral-directional state equations are

$$\dot{\beta} = \frac{\bar{q}s}{mV}\mathscr{R}\left(C_Y + \dot{\beta}_0\right) + \frac{g}{V}\mathscr{R}\cos\theta \sin\varphi + p \sin\alpha - r\cos\alpha$$

$$\dot{p}I_x - \dot{r}I_{xz} = \bar{q}sb\mathscr{R}C_\ell + \frac{qr}{\mathscr{R}}\left(I_y - I_z\right) + pq\frac{I_{xz}}{\mathscr{R}}$$

$$\dot{r}I_z - \dot{p}I_{xz} = \bar{q}sbC_n\mathscr{R} + \frac{pq}{\mathscr{R}}\left(I_x - I_y\right) - qr\frac{I_{xz}}{\mathscr{R}} - \frac{6Nq}{\mathscr{R}}I_{xe}$$

$$\dot{\varphi} = p + r\cos\varphi\tan\theta + q\sin\varphi\tan\theta + \dot{\varphi}_0$$

The $\dot{\beta}_0$ and $\dot{\varphi}_0$ are included to allow for instrument biases. The $\dot{p}$ and $\dot{r}$ equations already have the freedom to correct for such biases using C_{ℓ_0} and C_{n_0}; the C_{Y_0} in the $\dot{\beta}$ equation does not allow similar freedom because it also appears in the a_y equation below. Measured data are used for α, q, and θ in these equations. Measured φ is used for the trigonometric terms in the $\dot{\varphi}$ equation to eliminate the nonlinearity. The gravity term in the $\dot{\beta}$ equation is normally linearized about the measured φ; however, it can be evaluated at the measured φ if it is not desired to integrate the $\dot{\varphi}$ equation.

The observation equations are

$$\beta_m = K_\beta\left(\beta - \frac{z_\beta}{V}p + \frac{x_\beta}{V}r\right)$$

$$p_m = p$$

$$r_m = r$$

$$\varphi_m = \varphi \tag{60}$$

$$a_{y_m} = \frac{\bar{q}s}{mg}C_Y - \frac{z_{a_y}}{gR}\dot{p} + \frac{x_{a_y}}{g\mathscr{R}}\dot{r} - \frac{y_{a_y}}{\mathscr{R}^2 g}\left(p^2 + r^2\right)$$

$$\dot{p}_m = \dot{p} + \dot{p}_0$$

$$\dot{r}_m = \dot{r} + \dot{r}_0$$

The $\dot{p}_0$ and $\dot{r}_0$ are instrument biases on $\dot{p}$ and $\dot{r}$. Biases on a_y are included in C_{Y_0}; those on β, p, r, and φ were included in the state equations. Measured data are used for p^2 and r^2 in the a_y equation to eliminate the nonlinearity. The β observation equation does not include the division by cos α, which is needed if β is measured by a fixed vane.

The expansions of the force and moment coefficients are

$$C_Y = C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V\mathscr{R}} + C_{Y_r}\frac{rb}{2V\mathscr{R}} + C_{Y_\delta}\delta + C_{Y_0}$$

$$C_\ell = C_{\ell_\beta}\beta + C_{\ell_p}\frac{pb}{2V\mathscr{R}} + C_{\ell_r}\frac{rb}{2V\mathscr{R}} + C_{\ell_\delta}\delta + C_{\ell_0} + C_{\ell_{\dot\beta}}\frac{\dot\beta b}{2V\mathscr{R}} \tag{61}$$

$$C_n = C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V\mathscr{R}} + C_{n_r}\frac{rb}{2V\mathscr{R}} + C_{n_\delta}\delta + C_{n_0} + C_{n_{\dot\beta}}\frac{\dot\beta b}{2V\mathscr{R}}$$

where the δ term is summed over all controls.

4.1.3 Matrix Equations

This section describes the linearized equations of motion in the matrix form exactly as used by the standard aircraft routines. The variables TIMVAR (sec. 3.3.8(20)), and MZ (sec. 3.3.8(12)) affect the exact forms of the linearized equations used.

The general form of the equations is

$$\begin{aligned} R(t)\dot{x}(t) &= A(t)x(t) + B(t)u(t) + S(t)\prime(t) + \nu(t) + Fn(t) \\ z(t) &= C(t)x(t) + D(t)u(t) + H(t)\prime(t) + E(t)\dot{x}(t) + w(t) + G\eta(t) \end{aligned} \tag{36}$$

where A, B, S, R, C, D, H, and E are defined by relationships such as:

$$A_{ij}(t) = AM_{ij}(t) \times AN_{ij} + AL_{ij}(t) \tag{37}$$

Further discussion of the general equations and gradients is found in section 3.1. The N matrices for the standard aircraft routines are further discussed in sections 4.3.2.6 and 4.3.5. The M and L matrices are defined by user routines MAKEM and MAKEL, respectively. The ν and w vectors are defined by user routine MAKEVW.

In the following matrix definitions, the standard aircraft routines use average measured quantities unless TIMVAR (sec. 3.3.8(20)) is TRUE, in which case the matrices are redefined at each time point with the current measured values. The α_c and β_c used are obtained from α_m and β_m by the equations

$$\begin{aligned} \alpha_c &= \frac{\alpha_m}{\text{KALF}} + \frac{q(\text{XALF} + \text{DCGFT})}{V} - \frac{p\text{YALF}}{V} \\ \beta_c &= \frac{\beta_m}{\text{KB}} + \frac{p\text{ZB}}{V} - \frac{r(\text{XB} + \text{DCGFT})}{V} \end{aligned} \tag{62}$$

The variables KALF, XALF, YALF, KB, XB, and ZB are defined in section 4.3.3. If these quantities are allowed to vary, only the starting values are used to compute α_c and β_c. The β_c computation does not include the multiplication by cos α_c, which is needed if β is measured by a fixed vane.

If SHIFT (sec. 4.3.3(2)) is TRUE, DCGFT is defined as the flight center of gravity position (sec. 4.3.3(10)) minus the wind-tunnel reference center of gravity position (sec. 4.3.2(5)) times CHORD (sec. 4.3.3(6)). If SHIFT is FALSE, DCGFT is 0.

User routine UINIT defines the initial conditions and biases. The initial condition is set equal to the measured observation for all states except α and β. The α and β initial conditions are defined using equation (62). The biases UOFF and YOFF (sec. 3.1) are defined as 0.

4.1.3.1 Longitudinal. - The nondimensional longitudinal matrices are:

$AN =$

$$\begin{bmatrix} C_{L_\alpha} & C_{L_q} & 0 \\ C_{m_\alpha} & C_{m_q} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$BN =$

$$\begin{bmatrix} C_{L_{\delta_e}} \\ C_{m_{\delta_e}} \\ 0 \end{bmatrix}$$

$SN =$

$$\begin{bmatrix} C_{L_0} + \dot{\alpha}_0 \\ C_{m_0} \\ \dot{\theta}_0 \end{bmatrix}$$

$RN =$

$$\begin{bmatrix} 1 & 0 & 0 \\ C_{m_{\dot{\alpha}}} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$CN =$

$$\begin{bmatrix} \text{KALF} & \text{KALF} \times (\text{XALF} + \text{DCGFT}) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ C_{N_\alpha} & C_{N_q} & 0 \\ C_{A_\alpha} & C_{A_q} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$DN =$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N_{\delta_e}} \\ C_{A_{\delta_e}} \end{bmatrix}$$

$HN =$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N_0} \\ C_{A_0} \end{bmatrix}$$

$EN =$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \text{XAN} + \text{DCGFT} & 0 \\ 0 & \text{ZAX} & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The longitudinal dimensionalization matrices defined by the user routines are:

AM —

$$\begin{bmatrix} -\frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\frac{c}{2V} & 1 \\ \frac{\bar{q}sc}{I_y}\mathscr{R} & \frac{\bar{q}sc}{I_y}\frac{c}{2V} & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

BM —

$$\begin{bmatrix} -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} \\ -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

SM —

$$\begin{bmatrix} -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} & -\frac{\bar{q}s}{mV}\mathscr{R} \\ -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} & -\frac{\bar{q}sc}{I_y}\mathscr{R} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

RM —

$$\begin{bmatrix} 1 & 1 & 1 \\ -\frac{qsc}{I_y}\frac{c}{2V} & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

CM —

$$\begin{bmatrix} 1 & -\frac{1}{V} & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg}\frac{c}{2V\mathscr{R}} & 1 \\ -\frac{\bar{q}s}{mg} & -\frac{\bar{q}s}{mg}\frac{c}{2V\mathscr{R}} & 1 \end{bmatrix}$$

DM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{qs}{mg} & \frac{qs}{mg} & \frac{qs}{mg} & \frac{qs}{mg} \\ -\frac{qs}{mg} & -\frac{qs}{mg} & -\frac{qs}{mg} & -\frac{qs}{mg} \end{bmatrix}$$

EM —

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & \frac{1}{g\mathcal{R}} & 1 \\ 1 & \frac{1}{g\mathcal{R}} & 1 \end{bmatrix}$$

HM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{qs}{mg} & \frac{qs}{mg} & \frac{qs}{mg} & \frac{qs}{mg} \\ -\frac{qs}{mg} & -\frac{qs}{mg} & -\frac{qs}{mg} & -\frac{qs}{mg} \end{bmatrix}$$

The FM matrix is filled with 1's.

AL —

$$\begin{bmatrix} 0 & 1 & \frac{q}{V}\left(-\cos\varphi \sin\theta \cos\alpha_c + \cos\theta \sin\alpha_c\right) \\ 0 & 0 & 0 \\ 0 & \cos\varphi & 0 \end{bmatrix}$$

The BL, SL, CL, DL, HL, and EL matrices are filled with 0's.

The remaining quantities to be defined for the longitudinal equations are the known forcing functions, ν and w. All of the terms in ν and w are computed using only measured data. The definition of V depends on TIMVAR (sec. 3.3.8(20)) and MX (sec. 3.3.8(11)). The average V is used if TIMVAR is FALSE; otherwise, V is redefined at each time point.

$$\nu(1) = \frac{g}{V}\mathcal{R}\left(\cos\theta \cos\varphi \cos\alpha_c + \sin\theta \sin\alpha_c\right) - \beta_c\left(p \cos\alpha_c + r \sin\alpha_c\right) - AL_{1,3}\left(\theta - \mathrm{YOFF}_3\right) - \frac{\mathcal{R}T}{mV} \sin\alpha$$

where YOFF is the vector of biases in the measurement equation. If USERIC (sec. 3.3.8(26)) is TRUE, YOFF is a 0 vector; otherwise, YOFF is the vector of initial measurements. If MX is less than 3, the $\theta - \mathrm{YOFF}_3$ term is not included. This implies that the program uses measured θ alone, rather than linearizing about it.

$$v(2) = pr\frac{I_z - I_x}{I_y\mathscr{R}} + \left(r^2 - p^2\right)\frac{I_{xz}}{I_y\mathscr{R}} + \frac{6N}{\mathscr{R}}\,r\frac{I_{xe}}{I_y}$$

$$v(3) = -r\sin\varphi$$

$$w(1) = \mathrm{YALF}\,\frac{p}{V} \tag{63}$$

$$w(4) = -\frac{\mathrm{YAN}}{\mathscr{R}g}\dot{p} + \frac{\mathrm{ZAN}}{\mathscr{R}^2 g}\left(q^2 + p^2\right)$$

$$w(5) = -\frac{\mathrm{XAX} + \mathrm{DCGFT}}{\mathscr{R}^2 g}\left(p^2 + r^2\right) - \frac{\mathrm{YAX}}{\mathscr{R}g}\dot{r} + \frac{T}{mg}$$

4.1.3.2 Lateral-directional. - The nondimensional lateral-directional matrices are:

AN —

$$\begin{bmatrix} C_{Y_\beta} & C_{Y_p} & C_{Y_r} & 0 \\ C_{\ell_\beta} & C_{\ell_p} & C_{\ell_r} & 0 \\ C_{n_\beta} & C_{n_p} & C_{n_r} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

BN —

$$\begin{bmatrix} C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \\ C_{\ell_{\delta_a}} & C_{\ell_{\delta_r}} \\ C_{n_{\delta_a}} & C_{n_{\delta_r}} \\ 0 & 0 \end{bmatrix}$$

SN —

$$\begin{bmatrix} C_{Y_0} + \dot{\beta}_0 \\ C_{\ell_0} \\ C_{n_0} \\ \dot{\varphi}_0 \end{bmatrix}$$

RN —

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ C_{\ell_{\dot{\beta}}} & 1 & -\frac{I_{xz}}{I_x} & 0 \\ C_{n_{\dot{\beta}}} & -\frac{I_{xz}}{I_z} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

CN —

$$\begin{bmatrix} \text{KB} & \text{KB} \times \text{ZB} & \text{KB} \times (\text{XB} + \text{DCGFT}) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ C_{Y_\beta} & C_{Y_p} & C_{Y_r} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

DN —

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \end{bmatrix}$$

HN —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ C_{Y_0} \end{bmatrix}$$

EN —

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \text{ZAY} & \text{XAY} + \text{DCGFT} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

4.1.3.2

The dimensionalization matrices defined by the standard aircraft routines for a lateral-directional case are:

AM —

$$\begin{bmatrix} \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\frac{b}{2V} & \frac{\bar{q}s}{mV}\frac{b}{2V} & 1 \\ \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\frac{b}{2V} & \frac{\bar{q}sb}{I_x}\frac{b}{2V} & 1 \\ \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\frac{b}{2V} & \frac{\bar{q}sb}{I_z}\frac{b}{2V} & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

BM —

$$\begin{bmatrix} \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} \\ \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} \\ \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

SM —

$$\begin{bmatrix} \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} & \frac{\bar{q}s}{mV}\mathscr{R} \\ \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} & \frac{\bar{q}sb}{I_x}\mathscr{R} \\ \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} & \frac{\bar{q}sb}{I_z}\mathscr{R} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

RM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{\bar{q}sb}{I_x}\frac{b}{2V} & 1 & 1 & 1 \\ -\frac{\bar{q}sb}{I_x}\frac{b}{2V} & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

CM —

$$\begin{bmatrix} 1 & -\frac{1}{V} & \frac{1}{V} & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg}\frac{b}{2V\mathscr{R}} & \frac{\bar{q}s}{mg}\frac{b}{2V\mathscr{R}} & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

DM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

HM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} & \frac{\bar{q}s}{mg} \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

EM —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -\frac{1}{g\mathscr{R}} & \frac{1}{g\mathscr{R}} & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The *FM* matrix is filled with 1's.

AL —

$$\begin{bmatrix} 0 & \sin\alpha & -\cos\alpha & \cos\varphi\cos\theta\,\frac{g}{V} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & \cos\varphi\tan\theta & 0 \end{bmatrix}$$

The *BL*, *SL*, *RL*, *CL*, *DL*, *HL*, *EL*, and *FL* matrices are filled with 0's.

The vectors v and w in the lateral-directional equations depend on TIMVAR (sec. 3.3.8(20)) and MX (sec. 3.3.8(11)). All of the terms in v and w are computed using only measured data. The average measured V is used in the $v(1)$ equation if TIMVAR is FALSE; otherwise, V is redefined at each time point.

$$v(1) = \sin\varphi\cos\theta\,\frac{g}{V}\mathscr{R} - AL_{1,4}\left(\varphi - \mathrm{YOFF}_4\right)$$

where YOFF is the vector of biases in the measurement equation. If USERIC (sec. 3.3.8(26)) is TRUE, YOFF is a 0 vector; otherwise, YOFF is the vector of initial measurements. If MX is less than 4, the $\varphi - \mathrm{YOFF}_4$ term is omitted. This implies that the program uses the measured φ alone, rather than linearizing about it.

$$
\begin{aligned}
\nu(2) &= rq\frac{I_y - I_z}{I_x\mathscr{R}} + pq\frac{I_{xz}}{I_x\mathscr{R}} \\
\nu(3) &= pq\frac{I_x - I_y}{I_z\mathscr{R}} - rq\frac{I_{xz}}{I_z\mathscr{R}} - \frac{6Nq}{\mathscr{R}}\frac{I_{xe}}{I_y} \\
\nu(4) &= q \sin\varphi \tan\theta \\
w(5) &= -\frac{YAY}{\mathscr{R}^2 g}\left(p^2 + r^2\right)
\end{aligned}
\tag{64}
$$

4.2 Files

This section describes the file formats used by the standard aircraft routines. Only details specific to the standard aircraft routines are mentioned here. General descriptions of the files are found in section 3.2.

4.2.1 Input Time History File

The standard aircraft routine READTH reads time history data from file number UDATA only if CARD (sec. 3.3.8(2)) is FALSE; otherwise, time history data are read from cards and file number UDATA is ignored. Several details about the time history data are described in section 3.3.8, items (3) to (9), and in section 4.3.6.

The input time history file is a standard FORTRAN unformatted file, with one record for each time point. The first four words of each record contain the time as integer hours, minutes, seconds, and milliseconds. The rest of the record contains the data words as real FORTRAN variables.

The default order of the channels is α q V θ a_n $\dot{q}$ a_x δ_e δ_c $\delta_{1_{\text{longitudinal}}}$ $\delta_{2_{\text{longitudinal}}}$ φ h M $\bar{q}$ β p r a_y $\dot{p}$ $\dot{r}$ δ_a δ_r $\delta_{1_{\text{lateral}}}$ $\delta_{2_{\text{lateral}}}$ N T. This order can be changed with ZCHAN, UCHAN, and EXCHAN (sec. 3.3.8(7)).

The last record on the file should have a time greater than or equal to the maneuver stop time (sec. 3.3.10); otherwise, an end-of-file error will occur. Cases need not be run in the order that they appear on the file.

4.2.2. Predicted-Derivative File

FORTRAN file number UWT is used by the standard aircraft routines for predicted derivatives and related data. If the predicted-derivative control card (sec. 3.3.4) specifies NO , this file is not used; if it specifies OLD, an existing file is read. If the predicted-derivative control card specifies NEW or ONL, data are read from cards (sec. 4.3.2) to create a predicted-derivative file. If the ONL option was used, the program then stops; if the NEW option was used, execution continues using the

predicted-derivative file just created.

The predicted-derivative file is composed of 80-column card images. The contents are described below. The units must be consistent with METRIC (sec. 4.3.3(3)).

4.2.2.1 Header information. -

Card 1. 80-column title card identifying the data set.

Card 2. Reference area, reference span, reference chord, reference center of gravity, and axis system. The axis system is either "STAB" or "BODY". The data format is (3F10.3, F10.5, A4).

Card 3. Angle of attack and angle of sideslip flow amplification factors, KALF and KB, respectively (sec. 4.3.3(11)), in (2F10.3) format.

Card 4. Instrument locations XALF, XB, XAN, XAX, and XAY (sec. 4.3.3(12)) in (5F10.3) format.

Card 5. Instrument location YALF, YB, YAN, YAX, and YAY (sec. 4.3.3(13)) in (5F10.3) format.

Card 6. Instrument locations ZALF, ZB, ZAN, ZAX, and ZAY (sec. 4.3.3(14)) in (5F10.3) format.

Card 7. Number of angle of attack, Mach number, and extra parameter break points in the predicted-derivative data tables. The format of this card is (3I10).

Card 8. Angle of attack break points in (8G10.3) format. The break points are continued to further cards in the same format if required.

Card 9. Mach number break points in (8G10.3) format. The break points are continued to further cards in the same format if required.

Card 10. Extra parameter break points in (8G10.3) format. The break points are continued to further cards in the same format if required.

4.2.2.2 Predicted-derivative data. - This section describes the format of the predicted data for one derivative. The format is repeated as many times as there are derivatives. No particular order of the derivatives is required. For each derivative there is a derivative header card, followed by a derivative data table.

The header card contains the derivative name, type, and location in format (A4, 6X, A4, 6X, A2, 1X, I2, 1X, I2). The name is not used by the program, but is solely for ease of user identification. The type should be either "LONG" or "LATR" to indicate a longitudinal or lateral-directional derivative; if anything else is specified for type, the derivative will effectively be ignored. The

derivative location is specified by matrix name (AN, BN, SN, RN, CN, DN, HN, or EN), row, and column.

The derivative data table contains the derivative values at the angle of attack break points on one card in (8G10.3) format (further cards in the same format are added if there are more than eight angle of attack break points). This card (or cards) is repeated for each Mach number break point. Finally, all of the above cards are repeated for each parameter break point. This data table corresponds to that described in section 4.3.2 for AMP functional dependence. The other compressed and reordered forms described in section 4.3.2 cannot be used on the predicted-derivative file (subroutine WTIN expands and reorders the data before writing them to the file).

4.2.2.3 End card. - The end of the predicted-derivative file is indicated by a card with the characters "END" starting in column 1. This card is necessary because FORTRAN end-of-file checks are machine specific.

4.2.3 Punch File

Subroutine OUTPUN is called to punch the estimates if PUNCH (sec. 3.3.8(45)) is TRUE. The first card punched is the title card (sec. 3.3.6). The second card contains the case type, Mach number (sec. 4.3.3(20)), angle of attack (sec. 4.3.3(17)), extra parameter (sec. 4.3.3(21)), and center of gravity (sec. 4.3.3(10)) in format (A4, 6X, 4G10.3). The case type is either "LONG" or "LATR" (sec. 4.3.3(1)). If the date and time are available to the program (see the Programmer's Manual (ref. 38)), they will be punched in the last two fields of 10 columns on the second card. The third card contains the value MAXZ (sec. 2), followed by MAXZ channel averages for the measured observations. This card is in format ("Z AVG ", I3, 7G10.3). If MAXZ is greater than 7, continuation cards in format (8G10.3) are used. The fourth card contains "U AVG ", the value MAXU, and the MAXU channel averages for the controls. The fifth card contains "EX AVG ", the value LEX, and the LEX channel averages for the extra signals. The formats of the fourth and fifth cards are the same as for the third card, including the possibility of continuation cards.

The standard deviations of the measured observations, controls, and extra signals follow. The format is the same as for the averages, except that "SIG" replaces "AVG" in the labels. The signal minima follow in the same format, labeled "MIN"; then the maxima are punched, labeled "MAX".

Following the above header information, the relevant nondimensional matrices and Cramér-Rao bound matrices are punched in standard matrix format (sec. 3.3.11). Any nondimensional matrices that contain no independent unknowns are omitted, along with the corresponding Cramér-Rao bound matrices.

The last card punched contains "ENDCASE" in the first eight columns.

4.2.4 Output Time History File

The standard aircraft routine THOUT is intended for use in creating a simulated-data file. An unformatted FORTRAN file is written which can be used as an input time history file in a later run. The file has one record per time point, with time in integer hours, minutes, seconds, and milliseconds, in the first four words of the record. The rest of the record contains the final iteration predicted observation time history, the controls, and the extra signals. The complete dimensioned lengths of the observations, controls, and extra signals are written. These lengths are MAXZ, MAXU, and LEX, respectively (sec. 2).

4.3 Input Description

This section describes the card input required by the standard aircraft routines. The input description of the basic program given in section 3.3 still applies when the standard aircraft routines are used. The subsections below supplement section 3.3 with details applicable to the standard aircraft routines. In several cases, the standard routines affect the card input of the basic routines; for instance, they change default values—such effects are also described here.

4.3.1 User Initialization Input

The standard aircraft user initialization input consists of default GGI and F matrices. The matrices are read in standard matrix format (sec. 3.3.11), except that the matrix names are five and seven characters long. The matrix names used are "LONGGGI," "LATRGGI," "LONGF," and "LATRF" for the longitudinal and lateral-directional matrices. If not read in, the defaults for LONGF and LATRF are 0 matrices, and the defaults for GGI are diagonal matrices of size 5 by 5 with the following values:

LONGGGI—

```
10  60  30  200  200
```

LATRGGI—

```
150  0.5  300  10  5000
```

The last card of the user initialization input should have "END" starting in column 1. If no F or GGI matrices are to be read in, this can be the only card.

4.3.2 Predicted-Derivative Input

The standard aircraft version of the predicted-derivative input (sec. 3.3.5) is described herein. The predicted-derivative control card (sec. 3.3.4) should specify NO if predicted derivatives are not used, or OLD if the predicted-derivative file is already available; in either of these cases, the predicted-derivative input is omitted. This input is included if either NEW or ONL is specified on the predicted-

derivative control card.

4.3.2.1 Title card. - The first card of the predicted-derivative input is an 80-column title card. It is suggested that this card indicate the aircraft name, data source, initials of person preparing the data, date prepared, and any other identifying information necessary.

4.3.2.2 NAMELIST WIND. - The input described in this section is in the form of a FORTRAN NAMELIST called WIND. NAMELIST format is briefly discussed in section 3.3.8. (For further details, see the FORTRAN reference manuals for specific computer systems.) The parameters in NAMELIST WIND are described below.

(1) STAB (logical) - stability axes. If STAB is TRUE, the longitudinal derivatives are in stability axes; otherwise, they are in body axes. The lateral-directional data must be in body axes, regardless of STAB. The default value of STAB is FALSE.

(2) NABP - number of angle of attack break points in the predicted data. NABP cannot be greater than 20 or less than 1. The default value is 1.

(3) NMBP - number of Mach break points in the predicted data. NMBP cannot be greater than 20 or less than 1. The default value is 1.

(4) NPBP - number of extra parameter (param) break points in the predicted data. Param is used to represent any parameter, other than angle of attack or Mach number, that is used to distinguish derivative estimates. Possible distinctions include aircraft configuration, altitude, and power setting. NPBP cannot be greater than 20 or less than 1. The default value is 1.

(5) CG - reference center of gravity in fraction of chord. The default value is 0.25.

(6) PRINT (logical) - option to print predicted data. If PRINT is TRUE, all of the predicted data will be printed; otherwise, only the title card, reference center of gravity, axis system, break points, and header cards will be printed. The default value of PRINT is FALSE.

(7) AREA, SPAN, CHORD - reference area, span, and chord, respectively (ft^2 and ft or m^2 and m). The default values are all 0.

(8) KALF, KB (real) - flow amplification factors for angle of attack and angle of sideslip. The default values are both 1.

(9) XALF, XB, XAN, XAX, XAY - distances of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively, forward of the center of gravity (ft or m). The center of gravity location referenced is the wind-tunnel reference center of gravity if SHIFT (sec. 4.3.3(2)) is TRUE, or the actual flight center of gravity if SHIFT is FALSE. The default values are 0.

(10) YALF, YB, YAN, YAX, YAY - distances of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively, right of the center of gravity (ft or m). The default values are 0.

(11) ZALF, ZB, ZAN, ZAX, ZAY - distances of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively, below the center of gravity (ft or m). The default values are 0.

4.3.2.3 Angle of attack break points. - The angle of attack break points are listed on one to three cards in (8F10) format.

4.3.2.4 Mach number break points. - The Mach number break points are listed on one to three cards in (8F10) format.

4.3.2.5 Param break points. - The break points for the extra distinguishing parameter, param, are listed on one to three cards in (8F10) format.

4.3.2.6 Predicted-derivative data. - This section describes the predicted data input format for one derivative. The format is repeated for as many derivatives as are desired; no particular order of the derivatives is required. For each derivative, there is a derivative header card, followed by a derivative data table.

The header card contains the derivative name, type, location, and functional dependence. The derivative name is four characters starting in column 1; the name is not used by the program, but is solely for ease of user identification. The type should be either "LONG" or "LATR", starting in column 11, to indicate a longitudinal or lateral-directional derivative; if anything else is specified for type, the derivative will effectively be ignored.

The derivative location is specified by matrix name, row, and column. The matrix name can be AN, BN, SN, RN, CN, DN, HN, or EN, starting in card column 21. A left parenthesis separates the matrix name and row number, a comma separates the row and column numbers, and a right parenthesis follows the column number. The row and column numbers can be either one or two digits. No blanks are allowed before the closing parenthesis, except as the first character of a two-digit row or column number. Examples of permissible forms of the derivative location are: AN(1,1), AN(01,1), and AN(1, 1).

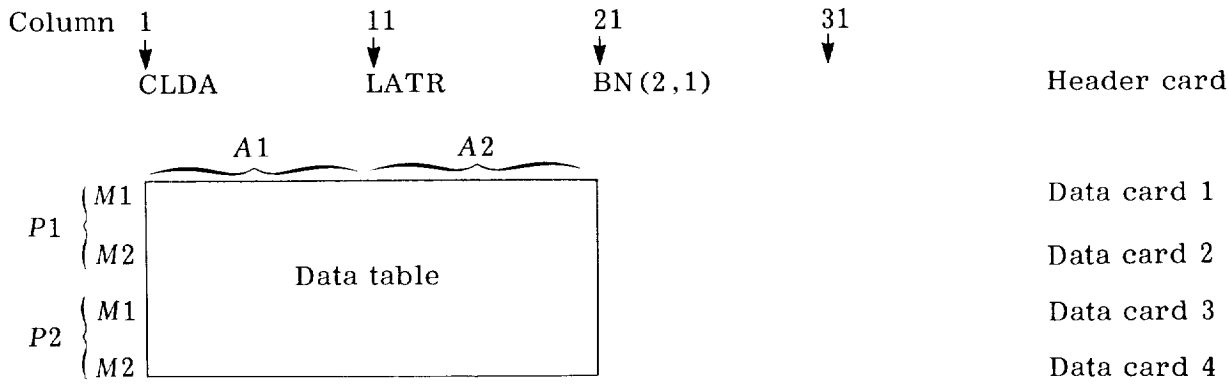
The functional dependence for each derivative is specified by one, two, or three of the characters "A" and "M" and "P" in any order, starting in column 31. The character "A" stands for angle of attack dependence, "M" for Mach number dependence, and "P" for param dependence. Blanks in columns 31 to 33 are equivalent to AMP; any character other than "A" or "M" or "P" or a blank in column 31 indicates the derivative is a constant independent of all three variables.

The form of the data table depends on the functional dependence indicated on the header card. If a constant derivative was indicated, the data table is a single card with the constant value in (F10) format. If the derivative depends on one parameter, the derivative values at the break points for that parameter are listed on

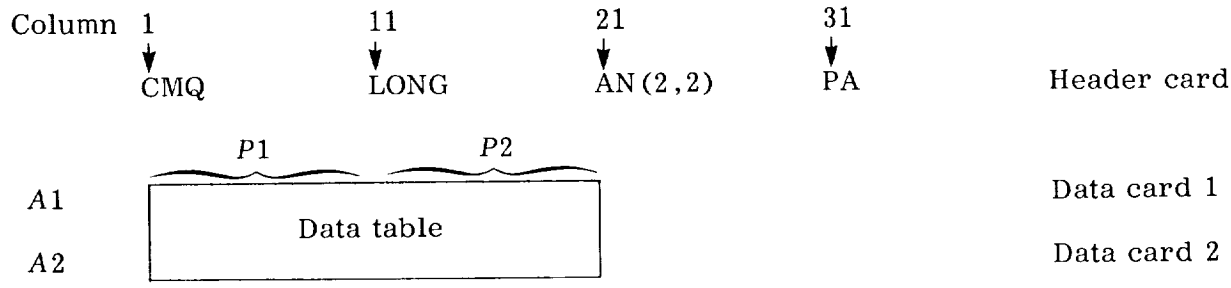
one to three cards in (8F10) format. If the derivative depends on two parameters, the one-parameter form is repeated for each break point of the second parameter. Finally, if the derivative depends on all three parameters, the two-parameter form is repeated for each break point of the third parameter.

Two examples should aid understanding of the data table organization. For both examples, we assume two angle of attack break points (denoted by $A1$ and $A2$), two Mach number break points (denoted by $M1$ and $M2$), and two param break points (denoted by $P1$ and $P2$).

Example 1:



Example 2:



Although the predicted-derivative file can contain values for any of the non-dimensional matrix locations, the following coefficients are those normally used. For a longitudinal case:

$$\text{AN} - \begin{bmatrix} 0 & 0 & 0 \\ C_{m_\alpha} & C_{m_q} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{RN} - \begin{bmatrix} 0 & 0 & 0 \\ C_{m_{\dot{\alpha}}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{BN} - \begin{bmatrix} 0 \\ C_{m_{\delta_e}} \\ 0 \end{bmatrix} \qquad \text{SN} - \begin{bmatrix} 0 \\ C_{m_0} \\ 0 \end{bmatrix}$$

CN —

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ C_{N_\alpha} & C_{N_q} & 0 \\ C_{A_\alpha} & C_{A_q} & 0 \end{bmatrix}$$

DN —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N_{\delta_e}} \\ C_{A_{\delta_e}} \end{bmatrix}$$

HN —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_N \\ C_A \end{bmatrix}$$

Note that total trimmed C_N and C_A are required in the HN matrix. See section 4.3.5(1) for further discussion of this point.

If the predicted longitudinal derivatives are in stability axes (secs. 4.3.2.2(1) and 4.2.2), then C_L and C_D derivatives should be placed in the fourth and fifth rows of the CN, DN, and HN matrices instead of C_N and C_A derivatives. Subroutine WTTRAN will convert the C_L and C_D derivatives to the required C_N and C_A derivatives. Note that the C_L derivatives should *not* be placed directly in the AN, BN, and SN matrices; the C_L derivatives in these matrices are computed by MATDEF, regardless of the axis system of the predicted data.

For a lateral-directional case, the following matrix locations are those normally used:

AN —

$$\begin{bmatrix} C_{Y_\beta} & C_{Y_p} & C_{Y_r} & 0 \\ C_{\ell_\beta} & C_{\ell_p} & C_{\ell_r} & 0 \\ C_{n_\beta} & C_{n_p} & C_{n_r} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

RN —

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ C_{\ell_{\dot\beta}} & 0 & 0 & 0 \\ C_{n_{\dot\beta}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

BN —

$$\begin{bmatrix} C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \\ C_{\ell_{\delta_a}} & C_{\ell_{\delta_r}} \\ C_{n_{\delta_a}} & C_{n_{\delta_r}} \\ 0 & 0 \end{bmatrix}$$

4.3.2.7 End card. - The end of the predicted-derivative file is indicated by a card with the characters "END" starting in column 1.

4.3.3 User Input

User input (sec. 3.3.7) for the standard aircraft routines consists entirely of a NAMELIST called USER. A brief discussion of NAMELIST format is contained in section 3.3.8; for complete details consult the FORTRAN reference manuals for specific computer systems. All values are reset to the defaults at the beginning of each case, regardless of any values used in previous cases. If user routines are bypassed (sec. 3.3.2), user input is not included in the deck. The following variables are included in NAMELIST USER.

(1) LONG, LATR (logical) - type of aerodynamic mode to be analyzed. Either LONG or LATR, but not both, should be set to TRUE to indicate a longitudinal or lateral-directional case, respectively. The default case type is lateral-directional.

(2) SHIFT (logical) - option to shift center of gravity reference from wind tunnel to flight. If SHIFT is TRUE, all predicted-moment derivatives are corrected by the program for the longitudinal distance between the wind-tunnel reference center of gravity (sec. 4.3.2.2(5)) and the flight center of gravity location, CG (item (10)).

The equations used for the moment corrections are

$$\begin{aligned} C_{m_{\delta_{CG_{flt}}}} &= C_{m_{\delta_{CG_{ref}}}} + \left(CG_{flt} - CG_{ref}\right)C_{N_\delta} \\ C_{n_{\delta_{CG_{flt}}}} &= C_{n_{\delta_{CG_{ref}}}} + \left(CG_{flt} - CG_{ref}\right)\frac{\text{CHORD}}{\text{SPAN}}C_{Y_\delta} \end{aligned} \tag{65}$$

where δ represents any α, β, control, or bias derivative. In addition, all longitudinal instrument locations (item (12)) are assumed to be given as offsets from the wind-tunnel reference center of gravity; the program adds the center of gravity difference to these offsets to obtain offsets from the flight center of gravity location. If SHIFT is FALSE, all predicted data are assumed to be referenced to the actual flight center of gravity location. No corrections are made to the moment derivatives or instrument locations, regardless of any values specified for the wind-tunnel reference or flight centers of gravity. No corrections for vertical or lateral center of gravity shifts are included in the program, whether SHIFT is TRUE or not. If the predicted-derivative control card (sec. 3.3.3) specifies NO , SHIFT is forced to FALSE, since no wind-tunnel reference is defined. The default value of SHIFT is TRUE.

(3) METRIC (logical) - metric units option. If METRIC is TRUE, SI (MKS) units are used for all data; otherwise, U.S. Customary (EGS) units are used. The default value of METRIC is FALSE.

(4) AREA - reference wing area (ft^2 or m^2). The units of AREA depend on METRIC (item (3)). The default value is obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(7)), if present; otherwise, the default is 0.

(5) SPAN - reference span (ft or m). The units of SPAN depend on METRIC (item (3)). The default value is obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(7)), if present; otherwise the default is 0.

(6) CHORD - reference chord (ft or m). The units of CHORD depend on METRIC (item (3)). The default value is obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(7)), if present; otherwise, the default is 0.

(7) W - aircraft gross weight (lb or N). The units of W depend on METRIC (item (3)). The default value is 0.

(8) IX, IY, IZ, IXZ (real) - aircraft moments of inertia (slug-ft^2 or kg-m^2). The units depend on METRIC (item (3)). The default values are all 0.

(9) IXE (real) - X-axis moment of inertia of rotating mass of the engine. The units depend on METRIC (item (3)). The default value is 0.

(10) CG - aircraft center of gravity in fraction of the reference chord. The center of gravity is used to adjust the longitudinal instrument offsets (item (12)) and the predicted-moment derivatives if SHIFT (item (2)) is TRUE. The default value of CG is 0.25.

(11) KALF, KB (real) - flow amplification factors for angle of attack and angle of sideslip, respectively. The computed observations for these angles are multiplied by KALF or KB. These multiplication factors are used to linearly model local flow and upwash effects. The default values of KALF and KB are obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(8)), if present. Otherwise, the defaults are 1.

(12) XALF, XB, XAN, XAX, XAY - longitudinal locations of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively (ft or m). These locations are all given as offsets from the center of gravity, positive for sensors forward of the center of gravity. The center of gravity location referenced is the wind-tunnel reference center of gravity if SHIFT (item (2)) is TRUE, or the actual flight center of gravity if SHIFT is FALSE. The units used for the sensor locations depend on METRIC (item (3)). The default values are obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(9)), if present; otherwise, the defaults are 0.

(13) YALF, YB, YAN, YAX, YAY - lateral locations of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively (ft or m). These locations are all given as offsets from the actual flight center of gravity, positive for sensors right of the center of gravity. The units used for these offsets depend on METRIC (item (3)). The defaults are obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(10)), if present; otherwise, the defaults are 0.

(14) ZALF, ZB, ZAN, ZAX, ZAY - vertical locations of the angle of attack, angle of sideslip, normal acceleration, longitudinal acceleration, and lateral acceleration sensors, respectively (ft or m). These locations are all given as offsets from the actual flight center of gravity, positive for sensors below the center of gravity. The units used for these offsets depend on METRIC (item (3)). The default values are obtained from the predicted-derivative file (secs. 4.2.2 and 4.3.2.2(11)), if present; otherwise, the defaults are 0.

(15) QBAR - average dynamic pressure (lb/ft^2 or N/m^2). The units of QBAR depend upon METRIC (item (3)). If QBAR is set to 0, a value is obtained from the average of the measured dynamic pressure time history. The default value of QBAR is 0.

(16) V - average velocity (ft/sec or m/sec). The units of V depend on METRIC (item (3)). If V is set to 0, a value is obtained from the average of the measured velocity time history. The default value of V is 0.

(17) ALPHA - average angle of attack (deg). If ALPHA is set to 999, a value is obtained from the average of the corrected, measured angle of attack time history. The default value of ALPHA is 999.

(18) THETA - average pitch angle (deg). If THETA is set to 999, a value is obtained from the average of the measured pitch angle time history. The default value of THETA is 999.

(19) PHI - average bank angle (deg). If PHI is set to 999, a value is obtained from the average of the measured bank angle time history. The default value of PHI is 999.

(20) MACH (real) - average Mach number. If MACH is set to 0, a value is obtained from the average of the measured Mach number time history. The default value of MACH is 0.

(21) PARAM - extra identifying parameter. This parameter is used in the predicted-derivative table lookup (sec. 4.3.2.6); it is also included on the punched output (sec. 4.2.4). The default value of PARAM is 0.

(22) UVAR (MAXU word integer vector) - vector of control flags. UVAR is a convenient means of defining the BV and DV matrices for the standard aircraft routines, without reading in the entire matrices. UVAR affects the default values of the BV and DV matrices; therefore, if BV or DV matrices are read from cards, the matrices read in will override those defined by UVAR.

For each element of UVAR that is set to 1, the standard derivatives for the corresponding control are allowed to vary; for the elements of UVAR that are 0, the control derivatives are fixed. The C_Y, C_ℓ, and C_n control derivatives are standard for lateral cases; the C_N, C_A, and C_m derivatives are standard for longitudinal cases. The BV and DV matrix defaults defined by UVAR are explicitly shown in section 4.3.5(2). For a longitudinal case, UVAR defaults to 1, 0, 0, 0 . . . (δ_e

derivatives will be determined). For a lateral-directional case, UVAR defaults to 1, 1, 0, 0 . . . (aileron and rudder derivatives will be determined).

4.3.4 Time History Card Input

This section describes the time history card input data (sec. 3.3.13) required by the standard aircraft routines. The time history data are read from cards only if CARD (sec. 3.3.8(2)) is TRUE; otherwise, time history data are read from tape or disk files and this card input should be omitted. Other details of the time history data are described in sections 3.3.8 (items (3) to (9)), 4.1, and 4.3.6.

The format of the first data card for each time point is (3I2, I3, 1X, 7F10). The time in hours, minutes, seconds, and milliseconds is read in the integer format; the last seven fields of the first card contain data for the first seven data channels. Subsequent cards contain the data for the remaining channels in (8F10) format.

The default order of the channels is α q V θ a_n $\dot{q}$ a_x δ_e δ_c $\delta_{1_{\text{longitudinal}}}$ $\delta_{2_{\text{longitudinal}}}$ φ h M $\bar{q}$ β p r a_y $\dot{p}$ $\dot{r}$ δ_a δ_r $\delta_{1_{\text{lateral}}}$ $\delta_{2_{\text{lateral}}}$ N T. This order can be changed with ZCHAN, UCHAN, and EXCHAN (sec. 3.3.8(7)).

The last record of data on the cards should have the first time greater than or equal to the maneuver end time (sec. 3.3.10). If further time history data cards are present, one will be read as the title card for the next case. If too few cards are present, a read error or end-of-file error will occur.

4.3.5 Matrix Defaults

As previously mentioned, all of the input matrix defaults can be changed by the user routines. This section describes the input matrix defaults for the standard aircraft routines. Although in some cases these defaults are unchanged from the basic program defaults described in section 3.3.11, they are repeated here for completeness; descriptions of input format and use of the matrices are not repeated. The dimensionalization matrices used to define the equations of motion are not described in this section, as they are defined internally in routines MAKEM and MAKEL, rather than being read in. The descriptions of these matrices for the standard aircraft routines are found in section 4.1.3.

In general, the matrix sizes are determined by the variables MX, MZ, MU, and MB (sec. 3.3.8(11) to (14)). This section describes the default values of the matrix elements, but does not discuss how large a partition of each matrix will actually be used. Any locations not shown herein are assumed to default to 0.

Many of the matrix defaults are affected by LONG and LATR (sec. 4.3.3(1)). A longitudinal default will refer to a default used if LONG is TRUE, and conversely, the lateral-directional defaults are used if LATR is TRUE; where neither type is specified, the default applies to all cases.

(1) AN, BN, SN, RN, CN, DN, HN, EN - nondimensional starting matrices. There are two types of default values for these matrices: First, the data on the predicted-derivative file (sec. 4.2.2); second, the specific values set in routine MATDEF. The values set in MATDEF will override the values from the predicted-derivative file if they refer to the same matrix element.

Data from the predicted-derivative file are used only if the predicted-derivative control card (sec. 3.3.4) specifies OLD or NEW. These data can contain tables for any location in AN, BN, SN, RN, CN, DN, HN, or EN. The table values are interplotted using ALPHA, MACH, and PARAM (sec. 4.3.3(17), (20), and (21)). Only tables with a type of "LONG" will be used for longitudinal cases, or "LATR" for lateral-directional cases.

Details of the predicted-derivative data are given in sections 4.2.2 and 4.3.2. The remainder of this section discusses the computations made by MATDEF and shows the resulting default matrices. It is assumed that the predicted-derivative data specify the derivatives and matrix locations shown in section 4.3.2.6.

If SHIFT (sec. 4.3.3(2)) is TRUE, the moment derivatives from the predicted-derivative file are corrected from the reference to the flight center of gravity location.

The default longitudinal AN is:

$$\begin{bmatrix} C_{L_\alpha} & C_{L_q} & 0 \\ C_{m_\alpha} & C_{m_q} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where MATDEF computes

$$AN(1,2) = C_{L_q} = C_{N_q} \cos\alpha - C_{A_q} \sin\alpha$$

$$AN(1,1) = C_{L_\alpha} = C_{N_\alpha}\left(\cos\alpha - \frac{\alpha}{\mathcal{R}} \sin\alpha\right) + C_{A_\alpha}\left(-\sin\alpha - \frac{\alpha}{\mathcal{R}} \cos\alpha\right) \qquad (66)$$

$$- \frac{\sin\alpha}{\mathcal{R}} C_{N_0} - \frac{\cos\alpha}{\mathcal{R}} C_{A_0} - \frac{\sin\alpha}{\mathcal{R}} C_{N_q} \frac{qc}{2V\mathcal{R}} - \frac{\cos\alpha}{\mathcal{R}} C_{A_q} \frac{qc}{2V\mathcal{R}} - \frac{\sin\alpha}{\mathcal{R}} C_{N_\delta}\delta - \frac{\cos\alpha}{\mathcal{R}} C_{A_\delta}\delta$$

where the δ terms are summed over all controls. The average values of α, q, and δ are used. If MZ (sec. 3.3.8(12)) is 4, the above computation is replaced by the low α approximations $C_{L_q} = C_{N_q}$ and $C_{L_\alpha} = C_{N_\alpha}$.

The lateral-directional AN default remains:

$$\begin{bmatrix} C_{Y_\beta} & C_{Y_p} & C_{Y_r} & 0 \\ C_{\ell_\beta} & C_{\ell_p} & C_{\ell_r} & 0 \\ C_{n_\beta} & C_{n_p} & C_{n_r} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The default size of AN is set to 3 by 3 for a longitudinal case or 4 by 4 for a lateral-directional case.

The longitudinal BN default is:

$$\begin{bmatrix} C_{L_{\delta_e}} \\ C_{m_{\delta_e}} \\ 0 \end{bmatrix}$$

where MATDEF computes $C_{L_{\delta_e}} = C_{N_{\delta_e}} \cos \alpha - C_{A_{\delta_e}} \sin \alpha$. If MZ (sec. 3.3.8(12)) is 4, the approximation $C_{L_{\delta_e}} = C_{N_{\delta_e}}$ is used.

The lateral-directional BN default remains:

$$\begin{bmatrix} C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \\ C_{\ell_{\delta_a}} & C_{\ell_{\delta_r}} \\ C_{n_{\delta_a}} & C_{n_{\delta_r}} \\ 0 & 0 \end{bmatrix}$$

4.3.5

The longitudinal SN default is:

$$\begin{bmatrix} C_{L_0} \\ C_{m_0} \\ 0 \end{bmatrix}$$

where MATDEF computes $C_{L_0} = C_{N_0} \cos \alpha - C_{A_0} \sin \alpha$. If MZ (sec. 3.3.8(12)) is 4, the approximation $C_{L_0} = C_{N_0}$ is used. The default size of the longitudinal SN is set to 3 by 1.

The lateral-directional SN default remains:

$$\begin{bmatrix} C_{Y_0} \\ C_{\ell_0} \\ C_{n_0} \\ 0 \end{bmatrix}$$

The default size of the lateral-directional SN is set to 4 by 1.

The RN matrix defaults are:

Longitudinal —

$$\begin{bmatrix} 1 & 0 & 0 \\ C_{m_{\dot{\alpha}}} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ C_{\ell_{\dot{\beta}}} & 1 & -\frac{IXZ}{IX} & 0 \\ C_{n_{\dot{\beta}}} & -\frac{IXZ}{IZ} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where IX, IZ, and IXZ are described in section 4.3.3(8).

The CN matrix defaults are:

Longitudinal —

$$\begin{bmatrix} \text{KALF} & \text{KALF} \times (\text{XALF} + \text{DCGFT}) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ C_{N_\alpha} & C_{N_q} & 0 \\ C_{A_\alpha} & C_{A_q} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} \text{KB} & \text{KB} \times \text{ZB} & \text{KB} \times (\text{XB} + \text{DCGFT}) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ C_{Y_\beta} & C_{Y_p} & C_{Y_r} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where C_{Y_β}, C_{Y_p}, and C_{Y_r} are obtained from $AN_{1,1}$, $AN_{1,2}$, and $AN_{1,3}$. KALF and KB are described in section 4.3.3(11), XALF and XB in 4.3.3(12), and ZB in 4.3.3(14). DCGFT is discussed in section 4.1.3.

The defaults for DN are:

Longitudinal —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N_{\delta_e}} \\ C_{A_{\delta_e}} \\ 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \end{bmatrix}$$

The longitudinal values in DN are taken directly from the predicted-derivative file. The lateral-directional values are obtained from $C_{Y_{\delta_a}} = BN_{1,1}$ and $C_{Y_{\delta_r}} = BN_{1,2}$.

The defaults for HN are:

Longitudinal —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N_0} \\ C_{A_0} \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ C_{Y_0} \end{bmatrix}$$

Although the total trimmed C_N and C_A should be placed in HN from the predicted-derivative file, the equations of motion require C_{N_0} and C_{A_0} in these locations. Subroutine WTDEF computes

$$\begin{aligned} C_{N_0} &= C_N - C_{N_\alpha}\alpha - C_{N_\delta}\delta \\ C_{A_0} &= C_A - C_{A_\alpha}\alpha - C_{A_\delta}\delta \end{aligned} \tag{67}$$

and replaces C_N and C_A by C_{N_0} and C_{A_0} in the HN matrix.

The defaults for EN are:

Longitudinal —

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \text{XAN + DCGFT} & 0 \\ 0 & \text{ZAX} & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \text{ZAY} & \text{XAY + DCGFT} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

XAN and XAY are described in section 4.3.3(12), and ZAX and ZAY in 4.3.3(14). DCGFT is discussed in section 4.1.3.

(2) F - state noise spectral density. The default for F is specified by the user initialization input (sec. 4.3.1). If not otherwise read in, the default for F is 0.

(3) AV, BV, SV, RV, CV, DV, HV, EV, FV - variation matrices. The defaults for AV are:

Longitudinal —

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The defaults for BV depend on UVAR (sec. 4.3.3(22)). For the default UVAR, the BV defaults are:

Longitudinal —

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$$

If UVAR is changed, the BV defaults contain columns like those shown for each element of UVAR that is 1. For elements of UVAR that are 0, the corresponding BV column is 0.

The defaults for SV are:

Longitudinal —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

If the weighting on θ (GGI(3,3)) is 0, the default of the third row of the longitudinal SV is 0. Similarly, if the weighting on φ (GGI(4,4)) is 0, the default of the fourth row of the lateral-directional SV is 0.

The defaults for RV are 0.

The default for CV is 0 in a lateral-directional case. In a longitudinal case, the CV default is:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The defaults for DV depend on UVAR (sec. 4.3.3(22)). For the default UVAR, the DV defaults are:

Longitudinal —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

If UVAR is changed, the longitudinal DV default contains columns like that shown for each element of UVAR that is 1. For elements of UVAR that are 0, the corresponding column is 0.

The defaults for HV are:

Longitudinal —

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

If the weighting (diagonal GGI element) for any signal is 0, the elements in the corresponding row of HV are forced to 0; this is true whether HV was read in or defaulted.

The defaults for EV and FV are 0.

(4) APRA, APRB, APRS, APRR, APRC, APRD, APRH, APRE, APRF - *a priori* weighting matrices.

The defaults for APRA are:

Longitudinal —

$$\begin{bmatrix} 100 & 0 & 0 \\ 300 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 300 & 0 & 0 & 0 \\ 1000 & 10 & 10 & 0 \\ 1000 & 10 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The defaults for APRB are:

Longitudinal —

$$\begin{bmatrix} 300 \\ 300 \\ 0 \end{bmatrix}$$

Lateral-directional —

$$\begin{bmatrix} 300 & 300 \\ 1000 & 1000 \\ 1000 & 1000 \\ 0 & 0 \end{bmatrix}$$

The default for APRC is 0 in a lateral-directional case; in a longitudinal case the default is:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 100 & 0 & 0 \\ 100 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The default for APRD in a lateral-directional case is 0; in a longitudinal case the default is:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 300 \\ 300 \\ 0 \end{bmatrix}$$

The defaults for APRS, APRR, APRH, APRE, and APRF are 0.

(5) GGI - inverse of GG^*. The GGI default is specified in the user initialization input (sec. 4.3.1). If not otherwise read in, the default is a diagonal matrix of size 5 by 5. The diagonal values are:

Longitudinal —

10 60 30 200 200

Lateral-directional —

150 .5 300 10 5000

(6) HARD - hard constraints.

For a lateral-directional case, the default hard constraints are:

$$CN(5,i) = AN(1,i) * 1 \qquad i = 1, 2, 3$$

$$DN(5,i) = BN(1,i) * 1 \qquad i = 1 \ldots MU$$

For a longitudinal case, the default hard constraints are:

$$BN(1,i) = DN(4,i) * (\cos\alpha) \qquad i = 1 \ldots MU$$

$$BN(1,i) = DN(5,i) * (-\sin\alpha) \qquad i = 1 \ldots MU$$

$$AN(1,1) = CN(4,1) * \left(\cos\alpha - \frac{\alpha}{\mathscr{R}}\sin\alpha\right)$$

$$AN(1,1) = CN(5,1) * \left(-\sin\alpha - \frac{\alpha}{\mathscr{R}}\cos\alpha\right)$$

$$AN(1,1) = HN(4,1) * \left(-\frac{1}{\mathscr{R}}\sin\alpha\right)$$

$$AN(1,1) = HN(5,1) * \left(-\frac{1}{\mathscr{R}}\cos\alpha\right)$$

$$AN(1,1) = DN(4,i) * \left(-\frac{1}{\mathscr{R}}\sin\alpha\,\delta_i\right) \qquad i = 1 \ldots MU$$

$$AN(1,1) = DN(5,i) * \left(-\frac{1}{\mathscr{R}}\cos\alpha\,\delta_i\right) \qquad i = 1 \ldots MU$$

$$AN(1,1) = CN(4,2) * \left(-\frac{1}{\mathscr{R}}\sin\alpha\,\frac{qc}{2V\mathscr{R}}\right)$$

$$AN(1,1) = CN(5,2) * \left(-\frac{1}{\mathscr{R}}\cos\alpha\,\frac{qc}{2V\mathscr{R}}\right)$$

$$AN(1,2) = CN(4,2) * (\cos\alpha)$$

$$AN(1,2) = CN(5,2) * (-\sin\alpha)$$

MU is described in section 3.3.8(13). Average values of α, q, and δ are used in the above constraints. If MZ (sec. 3.3.8(12)) is 4, $\alpha = 0$ is used in all of the longitudinal constraints. The only operative constraints are then:

$$BN(1,i) = DN(4,i) * 1 \qquad i = 1 \ldots MU$$

$$AN(1,1) = CN(4,1) * 1$$

$$AN(1,2) = CN(4,2) * 1$$

Section 3.3.11(6) describes how the user can control whether constraints read from cards supplement or replace the default constraints.

(7) SOFT - soft constraints. The default includes no soft constraints.

4.3.6 Altered Basic Program Defaults

This section summarizes the changes made by the standard aircraft routines to the default values of the basic program (sec. 3.3.8); unchanged variables are not included. The matrix default changes are described in section 4.3.5, and are not repeated here. Many of these defaults depend on the variables LONG and LATR (sec. 4.3.3(1)). If LONG is TRUE, the case is longitudinal; if LATR is TRUE, the case is lateral-directional.

The default value of NREC (sec. 3.3.8(6)) is changed to 25.

The defaults for ZCHAN, UCHAN, and EXCHAN (sec. 3.3.8(7)) are:

Longitudinal —

ZCHAN	1 2 4 5 7 6
UCHAN	8 9 10 11
EXCHAN	15 16 17 18 12 14 13 3 20 21
	26 27 28 29 30 31 32 33 34 35

Lateral-directional —

ZCHAN	16 17 18 12 19 20 21
UCHAN	22 23 24 25
EXCHAN	15 1 2 5 4 14 13 3 6 7
	26 27 28 29 30 31 32 33 34 35

The default value of RELAB (sec. 3.3.8(10)) remains FALSE, but the default signal labels are changed as follows:

Longitudinal —

OBSERVATIONS	ALPHA Q THETA AN AX Q-DOT
STATES	ALPHA Q THETA
CONTROLS	DELTA-E DELTA-C DELTA-C1 DELTA-C2
EXTRA SIGNALS	Q-BAR BETA P R PHI MACH ALT
	V P-DOT R-DOT RPM THRUST

Lateral-directional —

OBSERVATIONS	BETA P R PHI AY P-DOT R-DOT
STATES	BETA P R PHI
CONTROLS	DELTA-A DELTA-R DELTA-C1 DELTA-C2
EXTRA SIGNALS	Q-BAR ALPHA Q AN THETA MACH
	ALT V Q-DOT AX RPM THRUST

The default value of USERIC (sec. 3.3.8(26)) is changed to TRUE. Furthermore, USERIC should *not* be set to FALSE, or the equations of motion of the standard aircraft routines may be incorrect.

5.0 CONCLUDING REMARKS

A digital computer program written in FORTRAN IV is available for maximum likelihood parameter estimation. This program is capable of handling general linear or bilinear equations of arbitrary order with or without state noise. The basic program is quite general and, therefore, applicable to a wide variety of problems. The basic program can interact with a set of user-written problem-specific routines to simplify the use of the program on specific systems. A set of user routines for the standard aircraft stability and control problem is provided with the program. The program incorporates features that have been found useful for analysis of actual flight data at the NASA Dryden Flight Research Center, based on experience with over 5000 maneuvers from 35 different aircraft.

Dryden Flight Research Center
National Aeronautics and Space Administration
Edwards, Calif., August 24, 1979

REFERENCES

1. Parameter Estimation Techniques and Applications in Aircraft Flight Testing. NASA TN D-7647, 1974.

2. Gerlach, O. H.: High-Accuracy Instrumentation Techniques for Nonsteady Flight Measurements. Flight Test Instrumentation, Vol. 3, M. A. Perry, ed., Pergamon Press, 1964, pp. 77-100.

3. Taylor, Lawrence W., Jr.; and Iliff, Kenneth W.: A Modified Newton-Raphson Method for Determining Stability Derivatives From Flight Data. Paper presented at 2nd International Conference on Computing Methods in Optimization Problems (Sanremo, Italy), Sept. 1968.

4. Goodwin, Graham C.; and Payne, Robert L.: Dynamic System Identification: Experiment Design and Data Analysis. Academic Press, 1977.

5. Balakrishnan, A. V.: Communication Theory. McGraw-Hill Book Co., Inc., c.1968.

6. Taylor, Lawrence W., Jr.; and Iliff, Kenneth W.: Systems Identification Using a Modified Newton-Raphson Method—A FORTRAN Program. NASA TN D-6734, 1972.

7. Iliff, Kenneth W.; and Taylor, Lawrence W., Jr.: Determination of Stability Derivatives From Flight Data Using a Newton-Raphson Minimization Technique. NASA TN D-6579, 1972.

8. Larson, Duane B.: Identification of Parameters by the Method of Quasilinearization. CAL Rep. 164, Cornell Aeronautical Lab., Inc., May 1968.

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

10. Ross, A. Jean; and Foster, G. W.: Fortran Programs for the Determination of Aerodynamic Derivatives From Transient Longitudinal or Lateral Responses of Aircraft. C.P. No. 1344, British A.R.C., 1976.

11. Iliff, Kenneth W.; and Maine, Richard E.: Practical Aspects of Using a Maximum Likelihood Estimation Method to Extract Stability and Control Derivatives From Flight Data. NASA TN D-8209, 1976.

12. Maine, Richard E.; and Iliff, Kenneth W.: A FORTRAN Program for Determining Aircraft Stability and Control Derivatives From Flight Data. NASA TN D-7831, 1975.

13. Suit, William T.: Aerodynamic Parameters of the Navion Airplane Extracted From Flight Data. NASA TN D-6643, 1972.

14. McBrinn, D. E.; and Brassell, B. B.: Aerodynamic Parameter Identification for the A-7 Airplane at High Angles of Attack. Paper presented at AIAA Atmospheric Flight Mechanics Conference, Arlington, Tex., June 7-9, 1976.

15. Iliff, Kenneth W.; Maine, Richard E.; and Shafer, Mary F.: Subsonic Stability and Control Derivatives for an Unpowered, Remotely Piloted 3/8-Scale F-15 Airplane Model Obtained From Flight Test. NASA TN D-8136, 1976.

16. Frei, D. R.: Practical Applications of Parameter Identification. AIAA Paper 77-1136, Aug. 1977.

17. Iliff, Kenneth W.; Maine, Richard E.; and Steers, Sandra Thornberry: Flight-Determined Stability and Control Coefficients of the F-111A Airplane. NASA TM-72851, 1978.

18. Ross, A. Jean: Determination of Aerodynamic Derivatives From Transient Responses in Manoeuvring Flight. Methods for Aircraft State and Parameter Identification, AGARD-CP-172, May 1975, pp. 14-1 to 14-10.

19. Gould, D. G.; Hindson, W. S.: Estimates of the Stability Derivatives of a Helicopter and a V/STOL Aircraft From Flight Data. Methods for Aircraft State and Parameter Identification, AGARD-CP-172, May 1975, pp. 23-1 to 23-9.

20. Klein, V.: Longitudinal Aerodynamic Derivatives of a Slender Delta-Wing Research Aircraft Extracted From Flight Data. Report CIT-FI-74-023, Cranfield Inst. Technol., June 1974.

21. Schuetz, A. J.: Low Angle-of-Attack Longitudinal Aerodynamic Parameters of Navy T-2 Trainer Aircraft Extracted From Flight Data: A Comparison of Identification Techniques, Volume I: Data Acquisition and Modified Newton-Raphson Analysis. NADC-74181-30, Naval Air Development Center, Warminster, June 1975.

22. Marchand, M.; and Koehler, R.: Determination of Aircraft Derivatives by Automatic Parameter Adjustment and Frequency Response Methods. Methods for Aircraft State and Parameter Identification, AGARD-CP-172, May 1975, pp. 17-1 to 17-18.

23. Smith, Harriet J.: Flight-Determined Stability and Control Derivatives for an Executive Jet Transport. NASA TM X-56034, 1975.

24. Sim, Alex G.: A Correlation Between Flight-Determined Derivatives and Wind-Tunnel Data for the X-24B Research Aircraft. NASA SX-3371, 1976.

25. Powers, Bruce G.: Phugoid Characteristics of a YF-12 Airplane With Variable-Geometry Inlets Obtained in Flight Tests at a Mach Number of 2.9. NASA TP-1107, 1977.

26. Yazawa, Kenji: Identification of Aircraft Stability and Control Derivatives in the Presence of Turbulence. AIAA Paper 77-1134, Aug. 1977.

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

28. Steers, Sandra Thornberry; and Iliff, Kenneth W.: Effects of Time-Shifted Data on Flight-Determined Stability and Control Derivatives. NASA TN D-7830, 1975.

29. Iliff, Kenneth W.; and Maine, Richard E.: Further Observations on Maximum Likelihood Estimates of Stability and Control Characteristics Obtained From Flight Data. AIAA Paper 77-1133, Aug. 1977.

30. Park, Gary D.: Parameter Identification Technology Used in Determining In-Flight Airload Parameters. J. Aircraft, vol. 14, no. 3, Mar. 1977, pp. 251-256.

31. Iliff, Kenneth W.; Maine, Richard E.; and Montgomery, T. D.: Important Factors in the Maximum Likelihood Analysis of Flight Test Maneuvers. NASA TP-1459, 1979.

32. Maine, Richard E.: Aerodynamic Derivatives for an Oblique Wing Aircraft Estimated From Flight Data by Using a Maximum Likelihood Technique. NASA TP-1336, 1978.

33. Iliff, Kenneth W.: Maximum Likelihood Estimates of Lift and Drag Characteristics Obtained From Dynamic Aircraft Maneuvers. Proceedings of AIAA 3rd Atmospheric Flight Mechanics Conference, c.1976, pp. 137-150.

34. Kirsten, Paul W.; and Ash, Lawrence G.: A Comparison and Evaluation of Two Methods of Extracting Stability Derivatives From Flight Test Data. AFFTC-TD-73-5, Air Force Flight Test Center, Edwards Air Force Base, May 1974.

35. Park, Gary D.: Determination of Tail-Off Aircraft Parameters Using Systems Identification. Proceedings of AIAA 3rd Atmospheric Flight Mechanics Conference, c.1976, pp. 128-136.

36. Balakrishnan, A. V.: Stochastic Differential Systems I. Filtering and Control—A Function Space Approach. Lecture Notes in Economics and Mathematical Systems, 84, M. Beckmann, G. Goos, and H. P. Künzi, eds., Springer-Verlag (Berlin), 1973.

37. Iliff, K. W.: Identification and Stochastic Control With Application to Flight Control in Turbulence. UCLA-ENG-7340, School of Engineering and Applied Science, Univ. Calif., Los Angeles, Calif., May 1973.

38. Maine, Richard E.: Programmer's Manual for MMLE3, A General FORTRAN Program for Maximum Likelihood Parameter Estimation. NASA TP-1690, 1981.

39. Cramer, Harald: Mathematical Methods of Statistics. Princeton Univ. Press, 1946.

40. Klein, V.: Parameter Identification Applied to Aircraft. Cranfield Report Aero No. 26, Cranfield Inst. Technol., Oct. 1973.

41. Schweppe, Fred C.: Uncertain Dynamic Systems. Prentice-Hall, Inc., 1977.

42. Taylor, Lawrence W., Jr.: Application of a New Criterion for Modeling Systems. Methods for Aircraft State and Parameter Identification, AGARD-CP-172, May 1975, pp. 4-1 to 4-9.

43. Gupta, Narendra K.; Hall, W. Earl, Jr.; and Trankle, Thomas L.: Advanced Methods of Model Structure Determination From Test Data. AIAA Paper 77-1170, Aug. 1977.

44. Jazwinski, Andrew H.: Stochastic Processes and Filtering Theory. Academic Press, 1970.

45. Luenberger, David G.: Optimization by Vector Space Methods. Wiley, 1969.

46. Dixon, L. C. W.: Nonlinear Optimisation. Crane, Russak and Co., Inc., 1972.

47. Maine, Richard E.; and Iliff, Kenneth W.: Estimation of the Accuracy of Dynamic Flight-Determined Coefficients. AIAA Paper 80-0171, Jan. 1980.

48. Potter, James E.: Matrix Quadratic Solutions. J. SIAM Appl. Math., vol. 14, May 1966, pp. 496-501.

49. Wilkinson, J. H.: The Algebraic Eigenvalue Problem. Clarendon Press (London), 1978.

50. Moler, Cleve; and Van Loan, Charles: Nineteen Dubious Ways to Compute the Exponential of a Matrix. SIAM Review, vol. 20, no. 4, Oct. 1978, pp. 801-836.

51. Gainer, Thomas G; and Hoffman, Sherwood: Summary of Transformation Equations and Equations of Motion Used in Free-Flight and Wind-Tunnel Data Reduction and Analysis. NASA SP-3070, 1972.

INDEX OF NAMELIST VARIABLES

The variables in the three NAMELISTS of MMLE3 are listed alphabetically below. The numbers in parentheses refer to the sections and item numbers where definitions are found.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-1563		

4. Title and Subtitle	5. Report Date
USER'S MANUAL FOR MMLE3, A GENERAL FORTRAN PROGRAM FOR MAXIMUM LIKELIHOOD PARAMETER ESTIMATION	November 1980
	6. Performing Organization Code
7. Author(s) Richard E. Maine and Kenneth W. Iliff	8. Performing Organization Report No. H-1084
	10. Work Unit No. 505-36-24
9. Performing Organization Name and Address NASA Dryden Flight Research Center P.O. Box 273 Edwards, California 93523	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, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report is a user's manual for the FORTRAN IV computer program MMLE3, a maximum likelihood parameter estimation program capable of handling general bilinear dynamic equations of arbitrary order with measurement noise and/or state noise (process noise). This volume describes the theory and use of the program. The program listings and discussion of the inner structure and coding of the program are found in the Programmer's Manual for MMLE3, A General FORTRAN Program for Maximum Likelihood Parameter Estimation by Richard E. Maine (NASA TP-1690). The basic MMLE3 program is quite general and, therefore, applicable to a wide variety of problems. The basic program can interact with a set of user-written problem-specific routines to simplify the use of the program on specific systems. A set of user routines for the aircraft stability and control derivative estimation problem is provided with the program.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Maximum likelihood Parameter estimation Aircraft stability and control Computer programs System identification Aircraft flight testing	Unclassified—Unlimited STAR Category 59

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	89	$8.00

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

NASA-Langley, 1980