D RA

ADVANCED CONTROL CONCEPTS

FINAL REPORT

Contract NAS 8-29192 November 1973

By:

Maurice F Hutton And Bernard Friedland

Prepared For

GEORGE C. MARSHALL SPACE FLIGHT CENTER

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

(NASA-CR-120370) ADVANCED CONTROL CONCEPTS Final Report (Singer Co., Little Falls, N.J.) 190 p HC $12.50
N74-32284
CSCL 22A
Unclas
G3/31 16776

SINGER
AEROSPACE & MARINE SYSTEMS

THE SINGER COMPANY • KEARFOTT DIVISION • 1150 MC BRIDE AVENUE • LITTLE FALLS, N. J. 07424

Final Report

ADVANCED CONTROL CONCEPTS

By Maurice F. Hutton and Bernard Friedland

November 1973

Prepared under Contract NAS 8-29192

by

THE SINGER COMPANY, KEARFOTT DIVISION
LITTLE FALLS, NEW JERSEY

for

GEORGE C. MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TABLE OF CONTENTS

1. INTRODUCTION AND SUMMARY

Because of the possible launch configurations required to boost a space shuttle into orbit, it is anticipated that a large number of control effectors, including both aerodynamic surfaces and gimballed rocket engines, will be required to control the vehicle during ascent through the atmosphere. One objective in controlling the vehicle is to determine the deflection angle settings of the control effectors required to trim the vehicle for headwind and sidewind disturbances, and for bias torques due to solid rocket motor misalignments. Because of the launch configuration and the large number of controls, the control engineer is faced with two challenging problems. First, to compute the trim solution may entail solving a system of coupled, nonlinear equations. Second, if the number of control variables exceeds the number of independent trim equations to be satisfied, the trim solution is not unique.

To solve the uniqueness problem, additional constraints must be imposed. A logical choice for the additional constraints is the minimization of a performance criterion that penalizes the degradation in vehicle performance caused by large trim deflection angles. The performance criterion used in this investigation penalizes the following effects:

- Thrust loss (gain) by gimballing the engines away from their nominal condition.
- Thrust loss due to drag caused by deflecting aerodynamic surfaces.
- Excessive hinge moments on aerodynamic surfaces.
- Large movement of the actuators for trim which hampers the flexibility needed for dynamic response.

The inclusion of a performance criterion in the problem formulation results in an optimization problem with equality constraints to be solved for the trim solution. This formulation eliminates the uniqueness problem but the control engineer is still faced with the problem of explicitly solving the equations for the trim solution. Furthermore, the control engineer is likely to want to perform the trim computations many times in order to consider changes in the following:

- Flight regime (dynamic pressure)
- Desired trim conditions
- Launch vehicle configuration
- Set of control effectors
- Steady-state wind disturbances
- Performance criterion.

To serve this need, a computer program entitled TRIMS was developed to solve the trim problem numerically. The equations for the trim solution are based on the method of Lagrange multipliers and in general are nonlinear. Two standard numerical methods, steepest-descent and Newton-Raphson, are available for solving the nonlinear equations. Application of these methods yields a pair of iterative algorithms for computing the trim solution that are included in the TRIMS program. The program user canselect the desired method at the time of program execution. The Newton-Raphson method is more efficient for linear or nearly-linear equations, but may fail to converge in severely nonlinear problems unless started near the optimum solution. If the trim equations are linear and the performance criterion is quadratic, then the trim problem can be solved explicitly. For this case the Newton-Raphson method converges to the exact solution in one iteration. The current version of the TRIMS program for a Space Shuttle during ascent (described in Appendix C) solves the lateral trim problem. The lateral-directional dynamics are in the program and the required data (stability derivatives, moments of inertia, and etc.) supplied by MSFC are stored internally in the block data subroutine. The program permits multiple-case runs and the cost of computing a trim solution is minimal. The program is in a modular form that facilitates changes in the data and/or the equations defining the trim problem.

In computing the trim solution, the control engineer must specify the particular performance criterion to be used. There is no rule or theory for determining a unique performance criterion. The usual procedure is to vary the performance criterion and examine the different trim solutions that result. In the TRIMS program there are fourteen relative weighting factors in the input data that can be varied in the performance criterion. By varying these a family of acceptable trim solutions can be obtained for more detailed examination.

Two methods for determining which of the acceptable trim solutions is preferable were considered.

One possible method for selecting from among several trim solutions is based on controllability. If the trim problem is nonlinear, then the controllability of the linear vehicle dynamics about trim will depend on the particular trim solution. In this case, the trim solution that results in the most controllable system could be used. The notion of a controllability index is developed. This index provides a criterion for ranking the trim solutions according to the degree of controllability. The controllability index is computed from a symmetric, positive semi-definite controllability matrix and is defined as the ratio of the maximum eigenvalue to the minimum eigenvalue of the matrix. This ratio has a minimum value of unity for an orthogonal matrix. For an uncontrollable system, the controllability matrix is singular and the value of the controllability index is infinite. A difficulty with using the controllability index is that the controllability matrix is not unique and the value of the controllability index varies with the choice of this matrix. The controllability Grammain [1] is one possible choice for this matrix. Other controllability matrices are also considered in the development of the controllability index. A second method for selecting a trim solution is based on comparing the trim solution to the maximum allowable deflections. The rocket engines and aerodynamic control surfaces can only rotate a certain maximum angle. For particular flight times, a maximum hinge moment requirement can reduce the maximum deflection angle of an aerodynamic control surface below its physical limit. Obviously, the trim solution must be within these deflection limits. Moreover, to permit freedom of movement, a control deflection should not be too close to its angular limit. Hence, the requirement that all deflection angles be within their limits by a specified margin could be used to select the trim solution. For linear trim equations, a quadratic performance criterion with a diagonal weighting matrix can always be found for which the trim solution meets this requirement if such a solution exists at all. (This property of a diagonal weighting matrix might extend to nonlinear trim equations, but the more general case has not been studied.) The search for a trim solution satisfying this requirement can be accomplished by varying the diagonal elements of the weighting matrix using the penalty function method discussed in Section 3.2.

For the lateral trim problem of the Space Shuttle, there are two aerodynamic control surfaces (aileron and rudder) and five rocket engines (three orbiter engines and two solid rocket motors). The physical or hard limits on the aileron and rudder deflection are

$$\text{aileron} \quad \begin{cases} +15^\circ \\ -40^\circ \end{cases}$$

$$\text{rudder} \qquad \pm 30^\circ$$

and, as noted earlier, maximum allowable deflections can be less than these limits due to hinge moment restrictions which vary with flight time. The physical limits on the rocket engine deflections were assumed to be ± 30°. The maximum control deflections computed by the TRIMS program for the lateral trim problem exceeded the limits when the solid rocket motors are not gimballed. In numerous instances, a deflection angle exceeded 100 degrees. When the trim constraint of zero net side force is removed, the maximum deflection angles decrease by an order of magnitude and are within the limits.

In addition to the trim problem, the capability of the control system to damp out perturbations about trim must be considered. This can be identified as the dynamic response problem. In order to solve the dynamic response problem, it must be determined if the vehicle has sufficient dynamic control authority after trim conditions have been achieved. An approach to this problem based on the controllability Grammain used in defining the controllability index mention previously is studied. The controllability Grammain is used to compute the energy expended by each control in damping out errors from the trim conditions. This approach is limited, however, since it does not directly examine the peak deflection angles nor does it consider the realization of the feedback control system.

It is advantageous to have a single method for solving both the trim problem and the dynamic response problem. The method should minimize the total control deflection required both to trim the vehicle and to damp out initial errors and random disturbances. If the vehicle dynamics are linear, optimum control theory provides the desired method. In Section 3.4 the equations for the solution of the optimum control problem are derived for the case of bias inputs (trim problem) and random inputs (dynamic response problem). In Section 4.3, this theory is used to design an optimum feedback system for the lateral control of the Space Shuttle, and the closed-loop performance is simulated for a step change in side-slip angle. The computations for this example of the optimum control approach were performed with the aid of the Linear Systems Design (LSD) program developed at Singer-Kearfott under its Independent Research and Development program concurrent with this investigation.

It is recommended that further investigation of the trim problem for the Space Shuttle be performed with the aid of the TRIMS program for different combinations of controls (i.e., gimbal solid rocket motors), performance criterion, and trim constraints. In addition, a more extensive design effort using the optimum control approach would merit further consideration.

2. PROBLEM DESCRIPTION

The objective of this study is to determine how the control effectors for the Space Shuttle can be optimally used to achieve trim and dynamic control in the presence of wind disturbances and bias torques due to misalignment of rocket engines. Launch vehicles have in the past been primarily controlled by gimballing the rocket engines. Various Space Shuttle configurations now under investigation indicate that engine gimballing will not provide sufficient control to trim the vehicle for headwind and sidewind disturbances. Consequently, it may be necessary to use aerodynamic surfaces in conjunction with engine gimballing to achieve trim. Because of the severe cross-coupling problems encountered in the launch configurations, it appears that a large number of control effectors may be used. If the number of control effectors exceeds the number of quantities to be controlled, then the set of deflection angles to achieve trim is not unique. Thus, the control engineer in this case has a family (most likely an infinite set) of possible trim solutions to choose from. However, different trim solutions will result in different levels of performance and dynamic control. Consequently, the objective of the control engineer is to select the trim solution that provides the highest level of performance and dynamic control. To achieve this a performance criterion, which ranks the trim solutions according to level of performance and dynamic control, is defined. The problem then becomes, "What is the unique trim solution which optimizes the performance criterion?"

The algebraic equations for computing the trim solution are derived from the differential equations describing the motion of the vehicle by substituting the desired trim conditions. If the number of control variables exceeds the number of (independent) algebraic equations, then the trim solution is not unique. By addition of the performance criterion mentioned above, a meaningful optimization problem which can be solved for a unique trim solution, is obtained. This section develops the general problem in greater detail showing how the trim equations are derived from the equations of motion and the mathematical form of the performance criterion. The general equations for studying the dynamic response about trim are also derived.

2.1 Trim Problem

In general the motion of the vehicle is governed by a set of nonlinear, time-varying, differential equations of the form

$$\dot{x} = a(x,t) + b(\delta, x, t) + c(x, z, t) + v(t) \qquad (2.1)$$

where $x(t)$ = $n \times 1$ vector defining state of the vehicle motion at time t

$\delta(t)$ = $m \times 1$ vector of control deflections

$z(t)$ = $\ell \times 1$ vector of bias disturbances

$v(t)$ = $n \times 1$ vector of random disturbances

$a(x,t)$ = $n \times 1$ vector function of x and t

$b(\delta,x,t)$ = $n \times 1$ vector function of $\delta, x,$ and t

$c(x,z,t)$ = $n \times 1$ vector disturbance function of x, z, and t.

The trim problem is to find the set of control deflections δ_d that yield the desired steady state trim conditions x_d in the presence of bias disturbances z_d. The bias disturbances model the effects of a steady wind and misalignment torques. The trim problem ignores the random disturbances, i.e., $v(t) = 0$ is assumed. Therefore, the trim solution must satisfy

$$\dot{x}_d = 0 = a(x_d,t) + b(\delta_d,x_d,t) + c(x_d,z_d,t) \qquad (2.2)$$

Let

$$0 = \tilde{a}(x_d,t) + \tilde{b}(\delta_d,x_d,t) + \tilde{c}(x_d, z_d,t) \qquad (2.3)$$

represent the subset of (2) required to calculate the trim deflections δ_d where $\tilde{a}$, $\tilde{b}$, and $\tilde{c}$ are $\tilde{n} \times 1$ vector functions with $\tilde{n} \leq n$. In orther words, in obtaining the algebraic equations in (2.2) from the differential equations in (2.1), it is possible that some of the equations in (2.2) are satisfied by x_d independent of δ_d. These equations, although used in (2.1) for computing the dynamic response, are not used in computing δ_d and may be eliminated from (2.2). This elimination which results in (2.3) replacing (2.2) will be illustrated by the lateral control of the Space Shuttle in Section 4.

If $m < \tilde{n}$ then no set of control deflections δ_d exists that satisfy (2.3). If $m > \tilde{n}$ then a solution δ_d exists but is not unique. If $m = n$, there exist a unique solution, but unless b is a linear function of δ_d (i.e., $b(\delta_d, x_d, t) = B(x_d, t)\,\delta_d$ it may be difficult to find.

For the case of infinitely many possible trim solutions $(m > \tilde{n})$, certain solutions are preferable over others. An example of the latter is a solution in which each deflection angle is smaller in magnitude than for another trim solution. Trim solutions in which any of the deflection angles exceed the maximum allowable deflection should be excluded since such solutions cannot be realized. Suppose additional constraints in the form of a performance criterion are included in the problem formulation. The solution that satisfies the trim conditions (2.3) and minimizes the performance criterion is unique. For this approach the trim design problem reduces to the appropriate selection of the performance criterion.

The performance criterion denoted by r is a scalar function of the control

$$r = r(\delta_d) \tag{2.4}$$

In the case $m \geq \tilde{n}$ the trim problem is to find the set of control deflections

$$\delta'_d = [\delta_1 , \delta_2 , \ldots , \delta_m]$$

that satisfy (2.3) and minimize the performance criterion (2.4). In general, (2.3) and (2.4) are nonlinear functions of δ_d and the resulting optimization problem with equality constraints can not be solved analytically. Numerical methods for solving the nonlinear trim problem are developed in Section 3.1.

If the trim equation (2.3) is a linear function of δ_d

$$0 = \tilde{a}(x_d, t) + \tilde{B}(x_d, t)\,\delta_d + \tilde{c}(x_d, z, t) \tag{2.5}$$

where $\tilde{B}$ is a $\tilde{n}$ by m matrix and if the performance criterion is a quadratic form

$$r(\delta_d) = 1/2(\delta_d - \delta_o)' R(\delta_d - \delta_o) \tag{2.6}$$

where r is a positive definite matrix and where δ_o is the desired trim solution (in most instances $\delta_o = 0$) then the trim problem is said to be linear. The linear trim problem can be solved analytically and the equations are derived in Section 3.2.

2.2 GENERAL CONTROL PROBLEM

The trim problem is only part of the vehicle control problem. In addition to bias disturbances, the control system must be able to damp out sudden deviations from trim and to sustain proper vehicle motion in presence of fluctuating disturbances. An example is a sudden change or rapid fluctuation in the side wind velocity or equivalently the sideslip angle β. The capability of the control system to handle rapid fluctuations in β, for example, is commonly determined by simulating the performance for a step change, impulsive change, or random noise with a specified frequency spectrum. The control system must be designed to maintain the control deflections within the physical limits and to return the vehicle to trim within an acceptable setting time. This problem can be identified as the problem of dynamic response about trim. The first step in studying the dynamic response of the vehicle is to linearize the equations of motion about trim. Let Δx , $\Delta \delta$, Δz denote deviations of the state, control deflections, and bias disturbances, respectively, from trim.

$$\Delta x = x - x_d \qquad (2.7)$$

$$\Delta \delta = \delta - \delta_d$$

$$\Delta z = z - z_d$$

Expanding in a Taylor series the nonlinear functions a, b, and c in (2.1) about trim conditions results in the approximations

$$a(x,t) \cong a(x_d,t) + [\partial a/\partial x]\Delta x$$

$$b(x,t) \cong b(x_d,t) + [\partial b/\partial x]\Delta x + [\partial b/\partial \delta]\Delta \delta \qquad (2.8)$$

$$c(x,t) \cong c(x_d,t) + [\partial c/\partial x]\Delta x + [\partial c/\partial z]\Delta z$$

Substracting (2.2) and (2.1) and substituting (2.7) and (2.8) yields the linearized equations of motion

$$\Delta\dot{x} = A\Delta x + B\Delta \delta + C\,\Delta z + v$$

where

$$A = \partial a/\partial x + \partial b/\partial x + \partial c/\partial x \qquad (2.9)$$

$$B = \partial b/\partial \delta \qquad (2.9)$$

$$C = \partial c/\partial z$$

Note that the partial derivatives are evaluated about the trim conditions and that for particular values of δ_d , x_d , z_d , t the matrices A, B, and $\mathbb{C}$ are constant.

If the total motion (trim + dynamic response) is governed by linear differential equations then (2.1) becomes

$$\dot{x} = Ax + B\delta + Cz + v \qquad (2.10)$$

which has the same form as (2.9). The matrices A, B, C in (2.10) are in general a function of time t . By considering only a number of fixed points along the trajectory the problem reduces to a set of matrix equations of the form (2.10) with constant coefficients.

There are two general approaches for studying the general control problem including trim and dynamic response.

Approach 1: First solve the trim problem for a set of acceptable trim solutions by varying the performance criterion (2.4). From this set select the particular trim solution that leads to the best dynamic response. Methods for determining the particular trim solution are developed in Section 3.3.

Approach 2: Formulate a single performance criterion for the general control problem and solve for the optimum combination of trim solution plus dynamic response. This differs from the first approach in that two performance criteria are used in the former—one for the trim problem and one for the dynamic response problem.

Consider all possible combinations of forces and moments that can be generated by the controls of the Space Shuttle. This set defines the control authority of the vehicle. The restrictions on the control authority are of the form of bounds on the deflection angle, i.e.,

$$\delta_{\ell\min} \le \delta_\ell \le \delta_{\ell\max} \qquad \ell = 1, \ldots, m \tag{2.11}$$

For most of the controls the maximum deflection and is the same in either direction

$$\| \delta_\ell \| \le \delta_{\ell\max} \qquad \ell = 1, \ldots, m$$

The primary problem is to find a control solution that satisfies the restrictions (2.11). The restrictions (2.11) are in terms of the total deflection angles resulting from both trim and dynamic response requirements. Hence, the second approach is preferable to the first approach. However, the second approach in general presents more difficult computation problems. If the equations governing the total vehicle motion are nonlinear then it may be necessary to use the first approach; the second may lead to an intractable problem. If, on the other hand, the equations for the total motion are linear, as is the case of the space shuttle dynamics in Section 4, then a design method in the category of the second approach results from the application of optimum control theory. The use of optimum control theory to solve the general control problem with both random and bias input disturbances is developed in Section 3.4 and the application to the lateral control of the Space Shuttle is described in Section 4.3.

Even when the control design is to be performed using optimum control theory, there are advantages to first solving the trim problem. The trim solution is much easier to compute, and sufficient control authority must exist to handle at least the trim problem. Furthermore, the solution to the trim problem can aid in the formulation of the optimum control problem. The correlation between the trim solution and the optimum control solution is considered in Section 3.4.1.

3. ANALYTICAL METHODS

3.1 ITERATIVE SOLUTION OF NONLINEAR TRIM PROBLEM

3.1.1 Lagrange Multipliers

From (2.3) and (2.4) in Section 2 it was shown that the computation of the control deflections required to trim the vehicle for bias disturbances can be modeled as a problem of the following form:

Find the vector δ of dimension m which minimizes a scalar function of δ

$$\min_{\delta} \; r(\delta) \tag{3.1}$$

subject to a set of n equality constraints

$$0 = a + b(\delta) \tag{3.2}$$

For simplicity, the subscript "d" has been dropped from δ_d and (2.3) has been rewritten as (3.2) where

$$a \equiv \tilde{a}(x_d , t) + \tilde{c}(x_d , t)$$

$$b \equiv \tilde{b}(\delta, x_d , t)$$

$$n \equiv \tilde{n}$$

For a particular point in time t along the trajectory and for a particular set of desired trim conditions x_d and bias disturbances z , the vector a in (3.2) is a constant and the vector b ia a function of δ only.

In order to achieve a well-defined optimization problem the performance criterion $r(\delta)$ is assumed to have the following properties: assume that r is differentiable and let δ^* be the value of δ that minimizes $r(\delta)$. (Here the subscript d has been dropped from δ_d since just the properties of the performance criterion r are of interest irrespective of the trim equation (3.2).)

$$r(\delta) \geq r(\delta^*) \geq 0 \tag{3.3}$$

Then in some neighborhoods of δ^* , the performance citerion has the property that the gradient satisfies

$$\partial r / \partial \delta \begin{cases} = 0 & \text{for } \delta = \delta^* \\ \neq 0 & \text{for } \delta \neq \delta^* \end{cases} \tag{3.4}$$

where $\partial r / \partial \delta \equiv [\partial r / \partial \delta_1 , \ldots , \partial r / \partial \delta_m]$. Furthermore the second partial derivative of the performance criterion or Hessian matrix satisfies †

$$\partial^2 r / \partial \delta^2 \begin{cases} > 0 & \text{for } \delta = \delta^* \\ \geq 0 & \text{for } \delta \neq \delta^* \end{cases} \tag{3.5}$$

where $(\partial^2 r / \partial \delta^2)_{ij} \equiv \partial^2 r / \partial \delta_i \partial \delta_j$

The basic approach for solving the nonlinear trim problem given by (3.1) and (3.2) is to apply the well-known method of Lagrange multipliers.

Define a new scalar function h (the Hamiltonian) by

$$h(\delta , \lambda) = r(\delta) + \lambda' (a + b(\delta)) \tag{3.6}$$

where λ is a vector of n unknown parameters, commonly referred to as the "Lagrange multipliers". The fundamental idea underlying the method of Lagrange multipliers is that if δ^* , λ^* is the solution that minimizes h then δ^* is the solution that minimizes r and satisfies (3.2).

Assuming that the functions $r(\delta)$ and $b(\delta)$ are differentiable, the equations for the minimal solution δ^* , λ^* can be obtained by differentiating h and setting the derivatives to zero. This gives

$$\partial r / \partial \delta + \lambda' \partial b / \partial \delta = 0 \tag{3.7}$$

$$a + b(\delta) = 0 \tag{3.8}$$

---

† The notation " > 0" means the matrix is positive definite and " ≥0" means the matrix is positive semi-definite. For reference purposes see Appendix A for a discussion of differentiation by a vector.

This is a system of m+n equations in m+n unknown δ and λ. Only in special cases can (3.7) and (3.8) be solved explicitly. In general, numerical methods must be used to solve (3.7) and (3.8). Iterative numerical methods for determining the solution δ^*, λ^* that minimizes h given by (3.6), start with an initial guess δ_o, λ_o and then proceed to compute a sequence of solutions

$$\delta_1 , \delta_2 , \ldots , \delta_k , \ldots$$

$$\lambda_1 , \lambda_2 , \ldots , \lambda_k , \ldots$$

which converge to the exact solution δ^*, λ^*

$$\delta_k \rightarrow \delta^*$$

$$\lambda_k \rightarrow \lambda^*$$

Two such numerical methods are described in Sections 3.1.2 and 3.1.3.

3.1.2 Numerical Solution by Steepest Descent Method

One numerical method, in common use for many years, for finding the minimum of a function is that of "steepest descent". The steepest descent method is a 1st order gradient method and uses an iterative algorithm for improving the estimate of the solution so as to come closer to satisfying the zero slope conditions

$$\partial h / \partial \delta = 0 \text{ and } \partial h / \partial \lambda = 0$$

The method computes δ_{k+1}, λ_{k+1} from δ_k; the value of λ_k is not used to continue the iteration. The method partitions the vector δ according to

$$\delta' = [\underbrace{x}_{n} \mid \underbrace{u}_{m-n}]$$

where the subvectors x and u are computed separately.

Application of the steepest descent method gives the following steps for computing x_{k+1}, u_{k+1} from x_k, u_k.

1) From x_k , u_k compute the column vector $b(\delta)$.

2) From x_k , u_k compute the matrices $\partial b / \partial x$, $\partial b / \partial u$.

3) Compute the new estimate of subvector x according to

$$\Delta x_k = -(\partial b/\partial x)^{-1}(a + b(\delta))$$

$$x_{k+1} = x_k + \Delta x_k$$

4) From x_{k+1} , u_k compute the row vectors $\partial r/\partial x$, $\partial r / \partial u$.

5) Compute the vector of Lagrange multipliers according to

$$\lambda'_{k+1} = -(\partial r/\partial x)(\partial b/\partial x)^{-1}$$

6) Compute the gradient of h with respect to u using

$$\partial h / \partial u = \partial r / \partial u + \lambda'_{k+1}(\partial b/\partial u)$$

7) Compute the new estimate of subvector u according to

$$\Delta u_k = -\sigma(\partial h/\partial u)'$$

$$u_{k+1} = u_k + \Delta u_k$$

8) Repeat steps 1) through 8) with the updated solution x_{k+1} , u_{k+1} until the total error is very small

$$\| \Delta x_k \|^2 + \| \Delta u_k \|^2 < \epsilon$$

where the norms are given by

$$\| \Delta x_k \|^2 = \Delta x'_k \Delta x_k$$

$$\| \Delta u_k \|^2 = \Delta u'_k \Delta u_k$$

A flow chart showing the basic steps required to implement the steepest descent method for solving the trim control problem on the computer is given in Figure 3.1. A graphical interpretation of first order gradient methods is given on p. 20 of [2].

First order gradient methods usually show substantial improvements in the first few iterations but have poor convergence characteristics as the optimal solution is approached. A second-order gradient method, which uses the "curvature" as well as the "slope" at the nominal point, is discussed in the next section. Second order gradient methods have excellent convergence characteristics as the optimal solution is approached but unless the initial guess is in the region of convergence then the method may not converge or may converge to the wrong solution.

3.1.3 Numerical Solution by Newton-Raphson Method

Newton-Raphson method (or second-order gradient method) for locating the minimum point of a function uses both the first and second derivative at the nominal point to extrapolate a new estimate of the solution. A detailed description of the Newton-Raphson method is given in [1].

Using the Newton-Raphson method to find the minimum solution of $h(\delta , \lambda)$ given by (3.6) yields an iterative algorithm for computing the trim solution. To obtain the equations for computing $\delta_{k+1} , \lambda_{k+1}$ from δ_k , λ_k , first expand $h(\delta , \lambda)$ in a Taylor series about δ_k , λ_k .

$$h(\delta , \lambda) = h(\delta_k , \lambda_k) + \left[\begin{array}{c|c} h_\delta & h_\lambda \end{array}\right] \left[\begin{array}{c} \Delta\delta \\ \hline \Delta\lambda \end{array}\right] + \frac{1}{2} \left[\begin{array}{c} \Delta\delta \\ \hline \Delta\lambda \end{array}\right]' \left[\begin{array}{c|c} h_{\delta\delta} & h_{\delta\lambda} \\ \hline h_{\lambda\delta} & h_{\lambda\lambda} \end{array}\right] \left[\begin{array}{c} \Delta\delta \\ \hline \Delta\lambda \end{array}\right] + \ldots \tag{3.9}$$

where

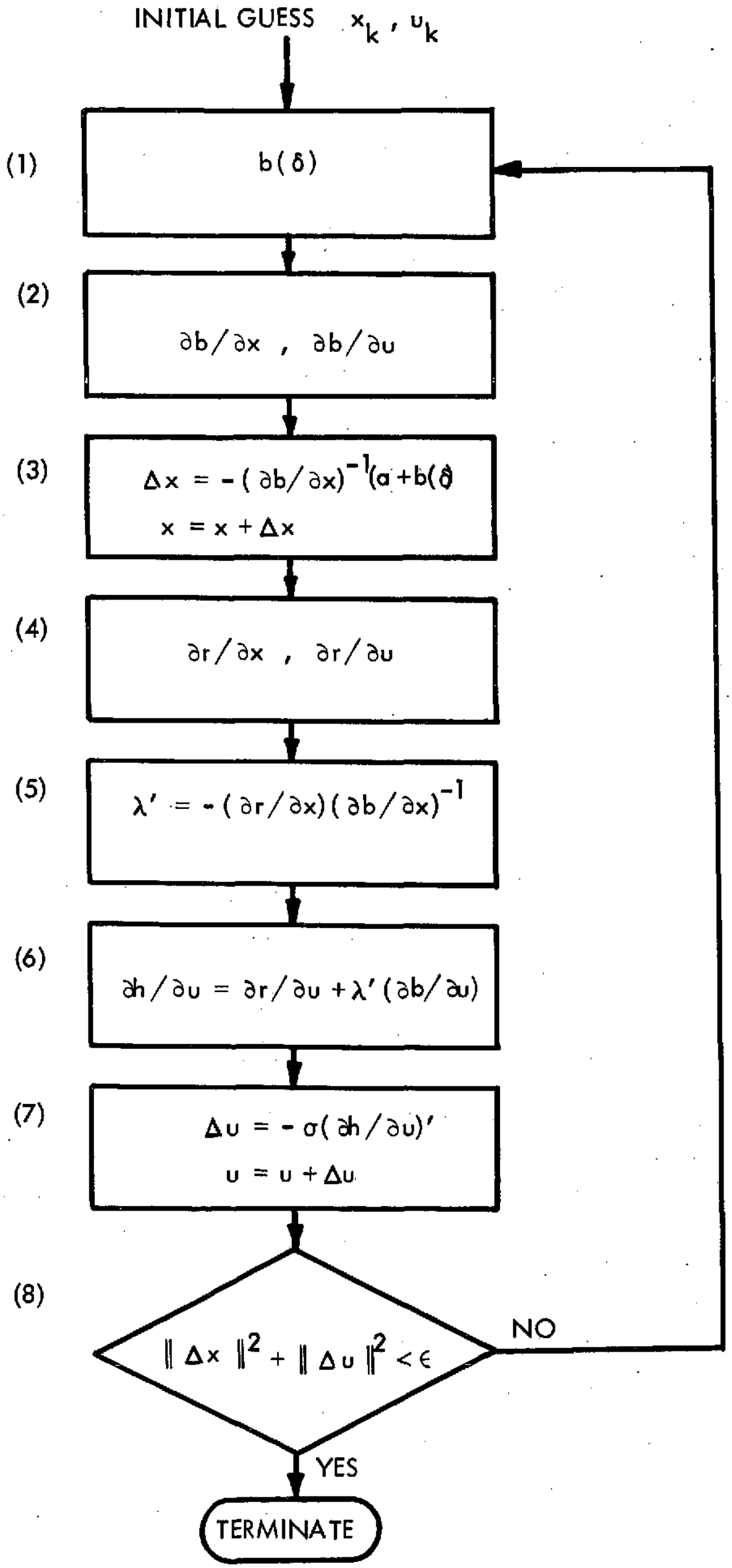
$$\Delta\delta = \delta - \delta_k \qquad \Delta\lambda = \lambda - \lambda_k \tag{3.10}$$

Differentiating (3.6) gives the following set of equations for evaluating the derivatives in (3.9)

$$h_\delta = \partial h/\partial\delta = \partial r/\partial\delta + \lambda'(\partial b/\partial\delta)$$

$$h_\lambda = \partial h/\partial\lambda = a' + b'(\delta) \tag{3.11}$$

Figure 3.1 Flowchart of Steepest Descent Method for Solving the Trim Control Problem



$$
\begin{aligned}
h_{\delta\delta} &= \partial^2 h / \partial \delta^2 = \partial^2 r / \partial \delta^2 + \lambda' (\partial^2 b / \partial \delta^2) \\
h_{\delta\lambda} &= \partial^2 h / \partial \delta \partial \lambda = (\partial b / \partial \delta)' \\
h_{\lambda\partial} &= \partial^2 h / \partial \lambda \partial \delta = \partial b / \partial \delta \\
h_{\lambda\lambda} &= \partial^2 h / \partial \lambda^2 = 0
\end{aligned}
\tag{3.12}
$$

From (3.9) the equations for computing the new estimate of the solution are:

$$
\begin{aligned}
\delta_{k+1} &= \delta_k + \Delta \delta_k \\
\lambda_{k+1} &= \lambda_k + \Delta \lambda_k
\end{aligned}
\tag{3.13}
$$

where the incremental corrections $\Delta \delta_k$, $\Delta \lambda_k$ are the solution of a system of linear equations

$$
\left[\begin{array}{c|c} h_{\delta\delta} & h_{\delta\lambda} \\ \hdashline h_{\lambda\delta} & h_{\lambda\lambda} \end{array} \right]
\left[\begin{array}{c} \Delta \delta_k \\ \hdashline \Delta \lambda_k \end{array} \right]
= - \left[\begin{array}{c} h'_\delta \\ \hdashline h'_\lambda \end{array} \right]
\tag{3.14}
$$

Note that the derivatives are evaluated about the nominal point δ_k , λ_k .

To summarize, the steps in the Newton-Raphson method for computing δ_{k+1} , λ_{k+1} from δ_k , λ_k are as follows:

1) From δ_k compute the column vector $b(\delta)$.
2) From δ_k compute the matrix $\partial b / \partial \delta$.
3) From δ_k compute the tensor $\partial^2 b / \partial \delta^2$.
4) From δ_k compute the row vector $\partial r / \partial \delta$.
5) From δ_k compute the symmetric matrix $\partial^2 r / \partial \delta^2$.
6) Compute the first-order gradient terms h_δ, h_λ according to (3.11).
7) Compute the second-order gradient terms $h_{\delta\delta}$, $h_{\lambda\lambda}$, $h_{\lambda\delta}$ according to (3.12). (Note that $h_{\lambda\lambda} = 0$.)
8) Compute the incremental correction to the solution by solving (3.14) which gives

$$\Delta\delta_k = -[R^{-1} - R^{-1}B'(BR^{-1}B')^{-1}BR^{-1}]h'_\delta - [R^{-1}B'(BR^{-1}B')^{-1}]h'_\lambda$$

$$\Delta\lambda_k = -[(BR^{-1}B')^{-1}BR^{-1}]h'_\delta + [(BR^{-1}B')^{-1}]h'_\lambda \tag{3.15}$$

where the matrices R and B are defined by

$$R = h_{\delta\delta}$$

$$B = h_{\lambda\delta} = h'_{\delta\lambda} \tag{3.16}$$

9) Update the solution according to (3.13).

10) Estimate the error in the solution by computing the norms

$$\| \Delta\delta_k \|^2 = \Delta\delta'_k \Delta\delta_k$$

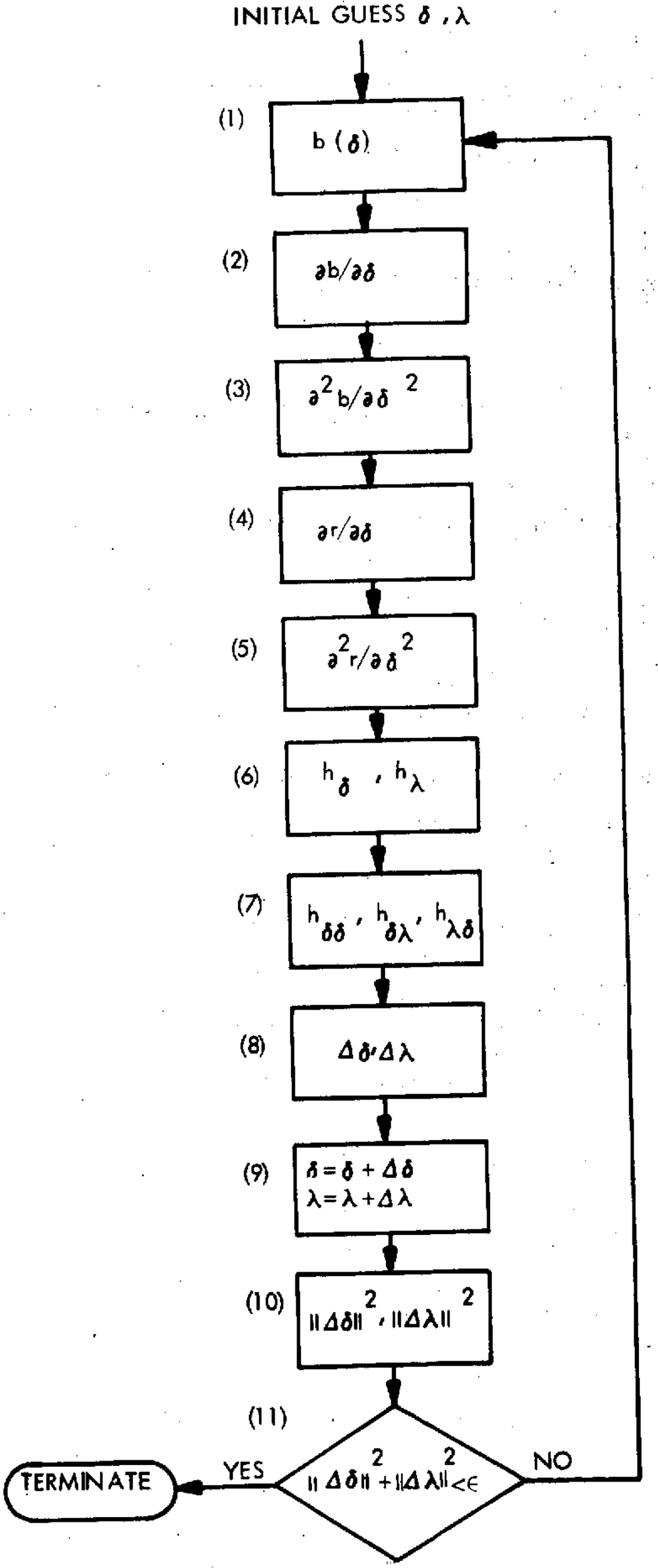
$$\| \Delta\lambda_k \|^2 = \Delta\lambda'_k \Delta\lambda_k$$

11) Repeat steps 1) through 11) with the updated solution δ_{k+1}, λ_{k+1} until the sum of the norms is very small as given by

$$\| \Delta\delta_k \|^2 + \| \Delta\lambda_k \|^2 < \epsilon$$

A flowchart showing the basic steps required to implement the Newton-Raphson method on the computer is given in Figure 3.2.

Figure 3.2 Flow Chart of Newton-Raphson Method for Solving the Trim Control Problem



3.2 SOLUTION OF LINEAR TRIM PROBLEM

3.2.1 Explicit Formulas

In the previous section the general nonlinear trim problem defined by (3.1) and (3.2) was discussed. The case of a linear trim equation

$$a + B\delta = 0 \tag{3.17}$$

and a quadratic performance criterion

$$r(\delta) = 1/2(\delta - \delta_o)'R(\delta - \delta_o) \tag{3.18}$$

is referred to as the linear trim problem and can be solved explicitly. The scalar Hamiltonian function corresponding to (3.17) and (3.18) is

$$h(\delta,\lambda) = 1/2(\delta - \delta_o)'R(\delta - \delta_o) + \lambda'(a+B\delta) \tag{3.19}$$

The vectors and matrices in the right-hand side of (3.19) are defined below

δ = m - vector of control deflections.

δ_o = m - vector of desired control deflections.

λ = vector of Lagrange multipliers of dimension (m-n).

a = constant vector of dimension n .

B = constant matrix of dimension n x m .

R = constant positive definite matrix of dimension m x m .

The trim solution is computed by determining the values of δ and λ that minimize the scalar function h . Differentiating (3.19) and setting the derivatives to zero gives

$$\begin{aligned}(\partial h/\partial \delta)' &= R(\delta - \delta_o) + B'\lambda = 0 \\ (\partial h/\partial \lambda)' &= a + B\,\delta\end{aligned} \tag{3.20}$$

The vector-matrix form of (3.20) is

$$\begin{bmatrix} R & B' \\ B & 0 \end{bmatrix}\begin{bmatrix} \delta \\ \lambda \end{bmatrix} = \begin{bmatrix} \delta_o \\ -a \end{bmatrix} \tag{3.21}$$

Premultiplying both sides of (3.21) by the inverse of the square matrix on the right hand side of (3.21) gives that the optimum trim solution is

$$\delta = [\mathcal{I} - B^{\#}B]\delta_o - B^{\#}a \tag{3.22}$$

where

$$B^{\#} = R^{-1}B'(BR^{-1}B')^{-1} \tag{3.23}$$

Note that $B^{\#}$ is a right inverse of B (i.e., $BB^{\#} = \mathcal{I}$) . Substituting (3.22) and (3.23) into (3.18) and (3.19) gives that the minimum values of performance criterion and Hamiltonian function are

$$h = r = 1/2\,(a+B\,\delta_o)'(BR^{-1}B')^{-1}(a+B\,\delta_o)$$

Consider the example of triming sidewind induced roll and yaw moments using aileron, rudder, and the yaw deflection of a single rocket engine. Setting the rolling and yawing moment coefficients to zero ($C_\ell = C_n = 0$) gives in vector form

$$\begin{bmatrix} C_{\ell Y} \\ C_{nY} \end{bmatrix} \delta_{EY} + \begin{bmatrix} C_{\ell R} \\ C_{nR} \end{bmatrix} \delta_R + \begin{bmatrix} C_{\ell A} \\ C_{nA} \end{bmatrix} \delta_A = \begin{bmatrix} C_{\ell \beta} \\ C_{n\beta} \end{bmatrix} \beta \tag{3.24}$$

or in slightly different form

$$\begin{bmatrix} C_{\ell Y} & C_{\ell R} & C_{\ell A} \\ C_{nY} & C_{nR} & C_{nA} \end{bmatrix} \begin{bmatrix} \delta_{EY} \\ \delta_R \\ \delta_A \end{bmatrix} = \begin{bmatrix} C_{\ell \beta} \\ C_{n\beta} \end{bmatrix} \beta \tag{3.25}$$

The trim equations given by (3.24) or (3.25) are a set of 2 linear equations in three unknowns δ_{EY}, δ_R, δ_A . Since there is one more unknown than equations, (3.24) has an infinite family of possible trim solutions.

A graphical representation of the possible trim solutions can be seen by depicting (3.24) in the yaw-roll moment coefficient plane as shown in Figure 3.3. The four vectors formed by the stability derivatives are represented by solid arrows where the following numerical values were chosen for the example

$$\begin{bmatrix} C_{\ell Y} \\ C_{nY} \end{bmatrix} = \begin{bmatrix} -0.20 \\ 0.80 \end{bmatrix} \qquad \begin{bmatrix} C_{\ell R} \\ C_{nR} \end{bmatrix} = \begin{bmatrix} 0.10 \\ -0.10 \end{bmatrix} \qquad \begin{bmatrix} C_{\ell A} \\ C_{nA} \end{bmatrix} = \begin{bmatrix} 0.07 \\ 0.08 \end{bmatrix} \qquad \begin{bmatrix} C_{\ell \beta} \\ C_{n\beta} \end{bmatrix} = \begin{bmatrix} -0.67 \\ 0.35 \end{bmatrix}$$

The dotted curve in Figure 3.3 represents one of the trim solutions for the case, $\beta = 1°$, and is the vector diagram corresponding to the left hand side of (3.24). The values of the deflection angles are

$$\delta_{EY} = -0.5 \qquad \delta_R = 2.807 \qquad \delta_A = 4.133$$

and are equal to the lengths of the dotted arrows divided by the lengths of the corresponding (parallel) solid arrows. The sign of the deflection angle is positive if the dotted arrow and the corresponding solid arrow point in the same direction, and the sign is negative if the directions are opposite.

The addition of a performance criterion to be minimized will yield a unique solution for (3.24). For illustration, one possible choice might be

$$r(\delta) = (\delta_{EY}/15°)^2 + (\delta_R/20°)^2 + (\delta_A/10°)^2 \qquad (3.26)$$

where $15°$, $20°$, and $10°$ are the corresponding maximum deflections. From (3.25) it follows that for $\beta = 1°$

$$B = \begin{bmatrix} -0.20 & 0.10 & 0.07 \\ 0.80 & -0.10 & 0.08 \end{bmatrix} \qquad a = \begin{bmatrix} -0.67 \\ 0.35 \end{bmatrix} \qquad (3.27)$$

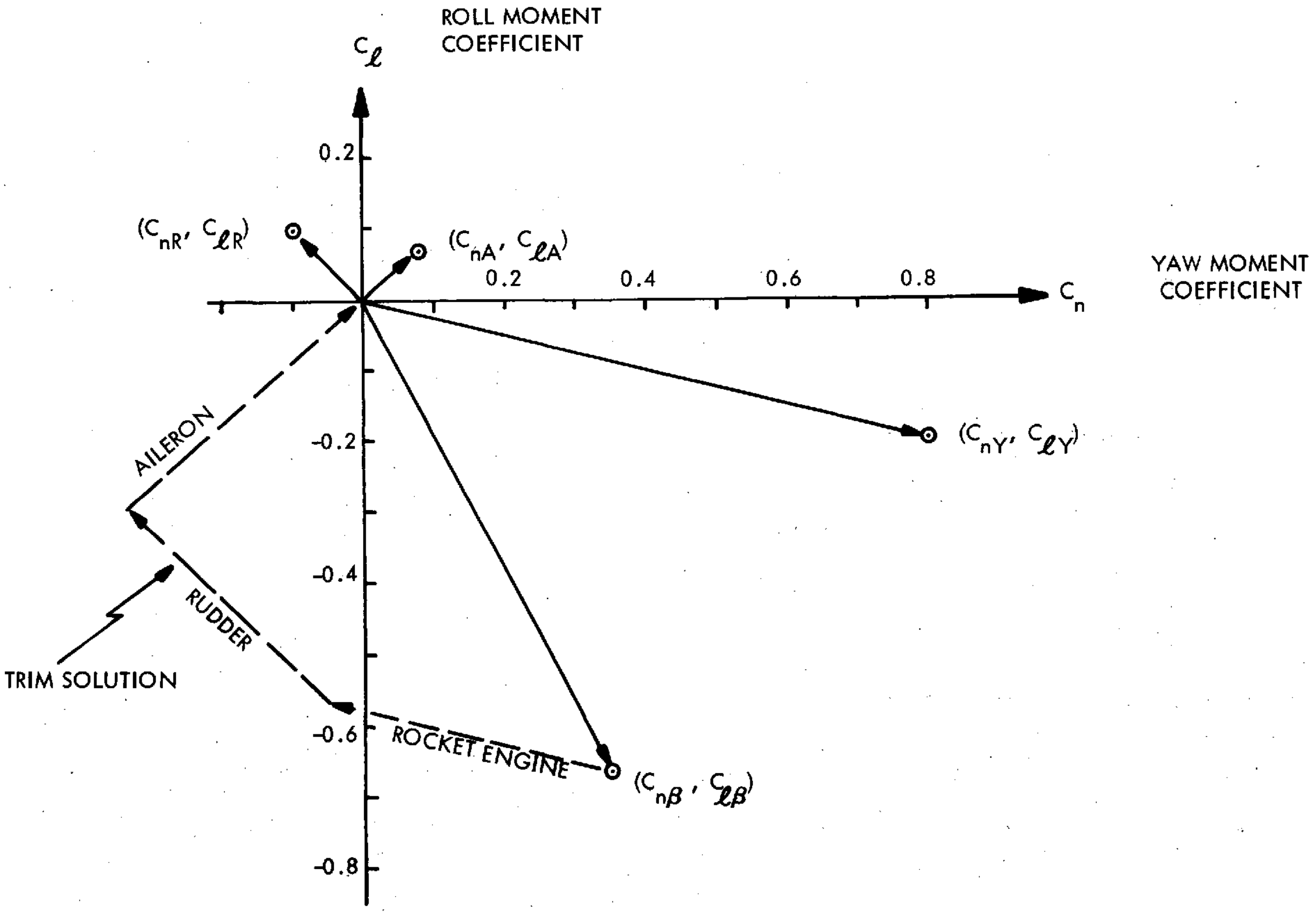
and from (3.26)

$$R = \begin{bmatrix} 1/225 & 0 & 0 \\ 0 & 1/400 & 0 \\ 0 & 0 & 1/100 \end{bmatrix} \qquad \delta_o = 0 \qquad (3.28)$$

Substituting (3.27) and (3.28) into (3.22) and (3.23) gives the solution

$$\delta = \begin{bmatrix} \delta_{EY} \\ \delta_R \\ \delta_A \end{bmatrix} = \begin{bmatrix} 0.10° \\ 5.70° \\ 1.72° \end{bmatrix} \qquad (3.29)$$

Figure 3.3 Yaw/Roll Coupling Characteristics



As an example of how changes in the performance criterion effect the minimal solution suppose in place of (3.26)

$$r(\delta) = (\delta_{EY}/20°)^2 + (\delta_R/20°) + (\delta_A/20°)^2$$

then

$$\delta = \begin{bmatrix} \delta_{EY} \\ \delta_R \\ \delta_A \end{bmatrix} = \begin{bmatrix} 0.10° \\ 5.70° \\ 1.72° \end{bmatrix} \tag{3.30}$$

3.2.2 Performance Criterion Selection

When infinitely many trim solutions are possible, certain solutions definitely require more control authority than other solutions and should not be used. In particular, given a trim solution δ , if it is possible to find another trim solution δ^* such that for each control

$$| \delta^*_\ell | \le | \delta_\ell | \qquad \ell = 1 , 2 , \ldots , m \tag{3.31}$$

where the strict inequality holds for some controls then δ should not be used. Property (3.31) partitions the possible trim solutions into two disjoint sets. If δ satisfies (3.31) it will be referred to as an unfavorable trim solution and if δ does not satisfy (3.31) it will be referred to as a favorable trim solution. The problem of selecting a form of the performance criterion that guarantees a favorable trim solution has been solved.

At this point a simple example is helpful in studying the properties of the trim problem. Suppose there is a single trim equation

$$0 = 6 - 2\delta_1 + \delta_2 \tag{3.32}$$

with two controls δ_1 and δ_2 . The general form of the performance criterion for the case of two controls is

$$r = 1/2 r_1 \delta_1^2 + 1/2 r_2 \delta_2^2 + r_3 \delta_1 \delta_2 \tag{3.33}$$

where

$$R = \begin{bmatrix} r_1 & r_3 \\ r_3 & r_2 \end{bmatrix}$$

and

$$r_1 > 0 \quad , \quad r_2 > 0 \quad , \quad r_3^2 < r_1 r_2$$

One approach for graphically representing the trim problem is to consider $\delta = [\delta_1 , \ldots , \delta_m]'$ as defining the coordinates of a point in an m-dimensional space which shall be referred to as the solution space. This approach is different from the graphical representation in Figure 3.3 where each coordinate corresponds to one of the scalar trim equations and hence might be referred to as the equation space representation. For this example the loci of possible trim solutions in the solution space is the straight line defined by (3.32) and shown in Figure 3.4. The segment of the straight line between points P and Q defines the set of favorable trim solutions and the remaining two segments on either side of P and Q define the set of unfavorable trim solutions.

For each fixed value of the performance criterion, there corresponds a closed contour curve in the solution space. For (3.33), r = constant defines an ellipse centered at the origin of the solution space. By parametrically increasing the value of r a family of concentric ellipses of increasing size is generated. One of these ellipses will be tangent to the straight line passing through P and Q . The point of tangency is the optimum solution. For example suppose $r_1 = r_2 = 1$ and $r_3 = 0$ then the loci of constant performance are circles as illustrated in Figure 3.4(a) for $r = 0.5$ and $r = 3.6$. The circle with $r = 3.6$ intersects the straight line at the single point $\delta_1 = 2.4$ and $\delta_2 = -1.2$. This is also the optimum solution obtained using the formulas (3.22) and (3.23). For the case $r_1 = 4$, $r_2 = 1$, and $r_3 = 0$ the optimum ellipse is

$$18 = 4\,\delta_1^2 + \delta_2^2$$

and is tangent to the straight line PQ at $\delta_1 = 1.5$ and $\delta_2 = -3.0$.

The above example illustrates how varying the weighting matrix R in the performance criterion leads to different trim solutions. However, there are more ways of varying R (degrees of freedom) than necessary. This means different choices of the R matrix can lead to the same optimum trim solution.

The redundancy in the selection of R suggests that R can be restricted to a diagonal matrix without disregarding a favorable trim solution. This assumption simplifies the selection of R . For the example illustrated in Figure 3.4, the principle axes of the ellipse will coincide with the coordinate axes in the solution space when and only when R is diagonal (i.e., $r_3 = 0$) . As a consequence, the optimum trim solution for a diagonal R matrix will always lie on the line segment PQ (region of favorable trim solutions). The optimum solution point in Figure 3.4 will move from point P to point Q as the ratio of the diagonal elements r_1 / r_2 increases from 0 to ∞ . Thus increasing the weighting on δ_1 relative to the weighting on δ_2 causes $|\delta_1|$ to decrease and $|\delta_2|$ to increase.

This example illustrates the following general properties of the weighting matrix in the performance criterion:

Property 1: The optimum trim solution for a diagonal R matrix is always a favorable trim solution.

Property 2: Any favorable trim solution is the optimum solution for some diagonal R matrix.

The general proof of the first property is not difficult. Let δ be the optimum trim solution for

$$R = \text{Diag}\,[r_1, \ldots, r_m]$$

then the minimum value of the performance criterion is

$$r = 1/2(r_1 \delta_1^2 + \ldots + r_m \delta_m^2) \tag{3.34}$$

Suppose δ is an unfavorable trim solution, then there exists another trim solution δ^* satisfying (3.31) for which the value of the performance criterion is

$$r^* = 1/2(r_1 \delta_1^{*2} + \ldots + r_m \delta_m^{*2}) \tag{3.35}$$

Comparing (3.34) to (3.35) term by term, it follows from (3.31) and $r_i > 0$ that

$$r^* < r$$

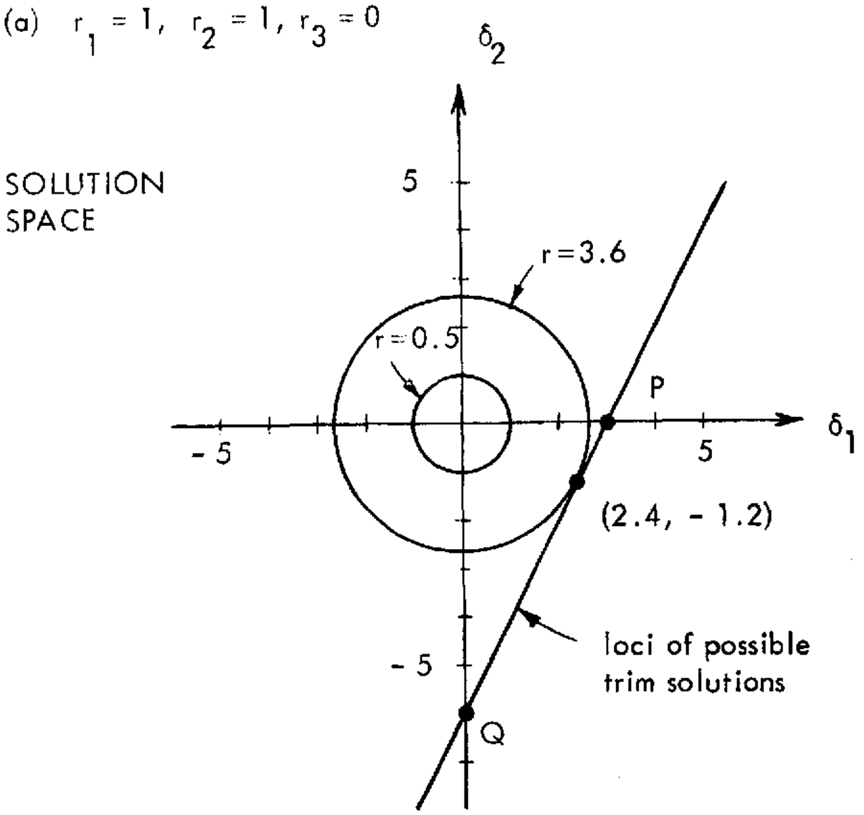
But r is the minimum value and hence a contradiction! Therefore, δ cannot be an unfavorable trim solution.

Given a favorable trim solution δ it should be possible to find a diagonal R matrix in the performance criterion for which the optimum solution is δ. A general method for constructing such an R matrix or equivalently, a general proof of the second property has not yet been found.

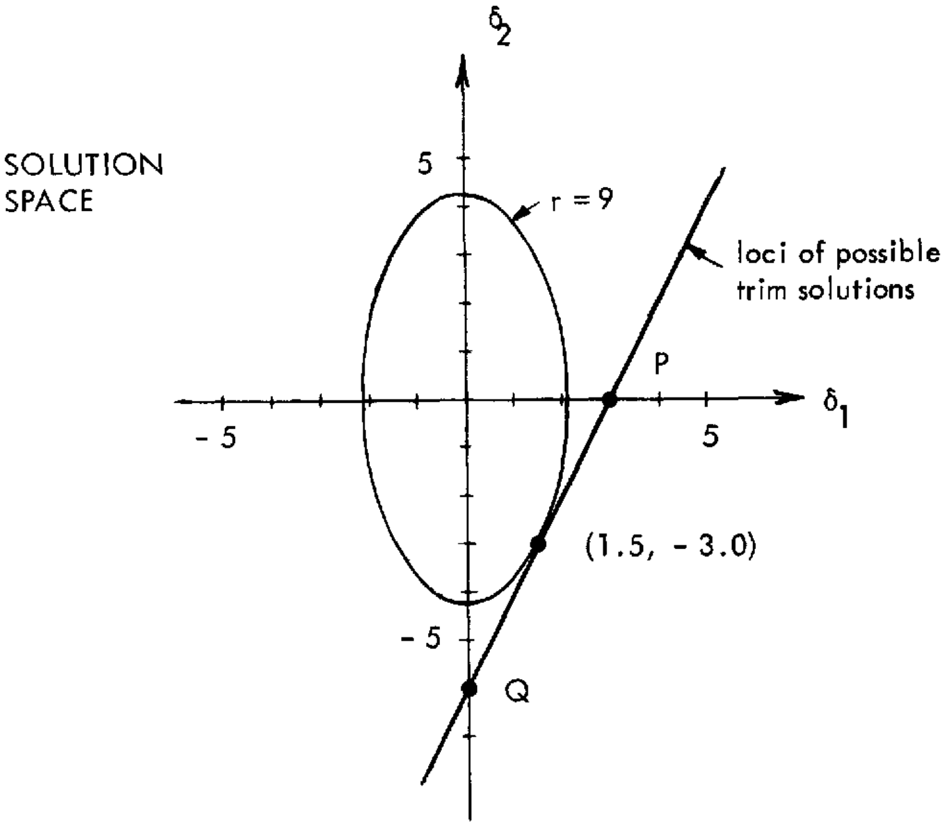
The formulation of trim control problem given by(3.17) and (3.18) is an optimization problem with equality constraints. However, as pointed out in Section 2, inequality constraints also exist due to the physical limitation on the control deflections. For a symmetric control, these will have the form

Figure 3.4 Example of Trim Problem and Solution Space Representation

(a) $r_1 = 1, \ r_2 = 1, \ r_3 = 0$



(b) $r_1 = 4, \ r_2 = 1, \ r_3 = 0$



$$| \delta_{\ell} | \leq \delta_{\ell\,\max} \qquad \ell = 1, \ldots, m \tag{3.36}$$

$\delta_{\ell\,\max}$ = maximum allowable deflection of the ℓth control

The inequality constraints are not included explicitly in the problem formulation since an optimization problem with both equality and inequality constraints is difficult to solve. Instead, the inequality constraints are handled by the penalty function method.

The basic idea of the penalty function method is to repeat the computation of the optimum trim folution for different R matrices in the performance criterion until each ratio $| \delta_{\ell} | / \delta_{\ell\,\max}$ is less than one and the difference $\delta_{\ell\,\max} - | \delta_{\ell} |$ is sufficiently large to provide the additional control required to solve the dynamic response problem. The procedure for varying the elements of R is simplified if R is restricted to be a diagonal matrix. From the properties of a diagonal R matrix discussed previously, this restriction does not exclude any favorable trim solutions but does exclude all unfavorable trim solutions. As an illustration of how to vary the diagonal elements of R, suppose the optimum solution for

$$R = \mathrm{Diag}\,[r_1, \ldots, r_m]$$

results in one of the deflections δ_{ℓ} exceeding its limits. The next step is to increase the corresponding weighting factor r_{ℓ} and solve the problem again. Repeat this procedure until δ_{ℓ} is smaller than the maximum deflection. An increase in the weighting factor r_{ℓ} will cause the magnitude of δ_{ℓ} to decrease at the expense of increasing the magnitude of other deflection angles. If no adjustment of the weighting factors results in all the control deflections being within their corresponding limits then the launch configuration does not possess sufficient control authority. If the limits are exceeded for every control then from Properties 1 and 2, mentioned earlier, no acceptable trim solution exists.

The modification of the performance criterion to produce a more desirable trim solution can be facilitated by realizing that for small perturbations the change in the optimum trim solution is proportional to the change in the weighting factors of the performance criterion. Computing the differentials of (3.22) and (3.23) for the case $\delta_o = 0$ gives

$$d\,\delta = -\,dB^{\#} \cdot a \tag{3.37}$$

$$-\,dB^{\#} = (I - B^{\#}B)R^{-1} \cdot dR \cdot B^{\#} \tag{3.38}$$

The derivation of (3.38) makes use of the identity

$$d(R^{-1}) = -R^{-1} \cdot dR \cdot R^{-1}$$

From (3.23) and (3.38) it can be shown that

$$B \cdot d\delta = 0$$

which also follows from computing the differential of (3.17). Equations (3.37) and (3.38) showthat for small perturbations $d\delta$ varies linearly with δR . Let $d\delta_{\ell}$ denote the change in the trim solution due to dR_{ℓ} , i.e.,

$$dR_{\ell} \rightarrow d\delta_{\ell}$$

Substituting

$$dR = \Sigma w_{\ell} dR_{\ell}$$

into (a 24) where w_{ℓ} is an arbitrary scalar results in

$$d\delta = \Sigma w_{\ell} d\delta_{\ell}$$

Thus, replacing R by R + dR causes the optimum trim solution to become $\delta + d\delta$.

3.3 CONTROLLABILITY AND DYNAMIC RESPONSE

3.3.1 Controllability Grammian

The use of the controllability Grammian for studying dynamic response about trim is developed below. The trim solution uses part of the control authority. If the vehicle deviates from trim due to random disturbance or a sudden wind gust then it must be determined if the control effectors have sufficient authority in reserve to return the vehicle to trim. By using a different trim solution, better dynamic response performance could possibly be achieved with respect to the control limits. The problem of determining which controls are most effective in zeroing out deviations is also of interest. If there are more control effectors available than required it may be possible to disregard those controls whose effectiveness is small.

Basic Theory

In vector-matrix notation the linearized equations of motion about trim have the general form

$$\dot{x} = Ax + Bu \tag{3.39}$$

where

x = state vector of dimension n

u = control vector of dimension m

The equation for the solution is

$$x(t) = \Phi(t)\, x(0) + \int_0^t \Phi(t-\tau)\, Bu(\tau)\, d\tau \tag{3.40}$$

where the transition matrix is

$$\Phi(t) = e^{At} \tag{3.41}$$

The control signal that will drive the error to zero at time T is

$$u(t) = - B'\Phi'(-t)\, W^{-1} x(0) \tag{3.43}$$

where

$$W \equiv W(T) = \int_0^T \Phi(-t)\, BB'\Phi'(-t) dt \tag{3.43}$$

The matrix function W(t) is referred to as the "controllability Grammian" [1]. Substituting (3.42) into (3.40) and using (3.43), it can be shown that $x(T) = 0$.

A useful criterion for indicating the amount of control effort is given by the integral

$$E = \int_0^T u'u \, dt \tag{3.44}$$

which may be viewed as proportional to the total "energy" expended by the control effectors in returning the vehicle to trim. Substituting (3.42) into (3.44) and using (3.43) yields the result

$$E = x'(0) W^{-1} x(0) \tag{3.45}$$

Thus the controllability Grammian W(t) provides a means for computing E .

Let E_ℓ denote the "energy" expended by the ℓth control effector, then

$$E_\ell = \int_0^T u_\ell^2 dt \qquad \ell = 1, \ldots, m \tag{3.46}$$

where

$$u_\ell(t) = B_\ell' \Phi'(-t) W^{-1} x(0) \tag{3.47}$$

and B_ℓ is the ℓth column of the B matrix. Substituting (3.47) into (3.46) results in

$$E_\ell = x'(0) W^{-1} W_\ell W^{-1} x(0) \tag{3.48}$$

with

$$W_\ell = \int_0^T \Phi(-t) B_\ell B_\ell' \Phi'(-t) \, dt \tag{3.49}$$

The ratio E_ℓ / E is a convenient measure for determining the relative effectiveness of the ℓth control effector. Upon substituting

$$BB' = B_1 B_1' + B_2 B_2' + \ldots + B_m B_m' \qquad 3.50)$$

into (3.43), it follows from (3.45), (3.48), and (3.49) that

$$W = W_1 + W_2 + \ldots + W_m \tag{3.51}$$

and

$$E = E_1 + E_2 + \ldots + E_n \tag{3.52}$$

Another approach for computing E and E_ℓ is obtained by rewriting (3.44) as

$$E = \text{trace}\left\{\int_0^T uu'dt\right\} \tag{3.53}$$

Substituting (3.42) into (3.53) gives

$$E = \text{trace}\ \{B'MB\} \tag{3.54}$$

where

$$M = \int_0^T \Phi'(-t)W^{-1}x(0)x'(0)W^{-1}\Phi(-t)\,dt \tag{3.55}$$

Repeating this approach for (3.46) and (3.47) leads to

$$E_\ell = B_\ell' M B_\ell \qquad \ell = 1, 2, \ldots, m \tag{3.56}$$

The advantage of using (3.56) in place of (3.49) is that instead of computing $W_1, W_2, \ldots, W_m$ only have to compute M . The disadvantage is that if the initial state vector x(0) changes then M must be recomputed where as the matrices W_ℓ are not a function of x(0) and hence do not change.

Computation of Controllability Grammian

Several methods for computing the matrix $W \equiv W(T)$ defined by (3.43) are discussed below.

Eigenvector Transformation

Suppose a new set of state variables q(t) are introduced that are related to x(t) by

$$q = Qx \tag{3.57}$$

where by assumption Q is a nonsingular matrix.
Substituting (19) into (1) gives

$$\dot{q} = \tilde{A}q + \tilde{B}u \tag{3.58}$$

where

$$\tilde{A} = QAQ^{-1}$$

$$\tilde{B} = QB$$

Let $\tilde{W}(t)$ denote the controllability Grammian computed from (3.58) then defining $\tilde{W} \equiv \tilde{W}(T)$ and applying the definition (3.43) to (3.58) results in

$$\tilde{W} = QWQ' \quad \text{or} \quad W = Q^{-1}\tilde{W}Q^{-1'} \tag{3.59}$$

If

$$\tilde{A} = \text{Diag}\,[\lambda_1, \lambda_2, \ldots, \lambda_n]$$

where λ_i are the eigenvalues of A then the columns of Q^{-1} form the corresponding set of eigenvectors. In this development it is assumed that the eigenvalues are real and distinct. The method can still be applied to the complex and the multiple eigenvalue case but the computations are more complicated. This method will not be generalized because it is intended only for illustration purposes and as a means for checking the other methods. If $\tilde{A}$ is a diagonal matrix then the transition matrix is a diagonal matrix with diagonal elements.

$$\tilde{\Phi}_{ii}(t) = e^{\lambda_i t} \qquad i = 1, \ldots, n$$

which upon substitution into the definition of the controllability Grammian (3.43) gives that the element of matrix $\tilde{W}$ in row i and column j is

$$\tilde{W}_{ij} = \tilde{b}_i'\tilde{b}_j \int_0^T e^{-(\lambda_i + \lambda_j)t} dt \tag{3.60}$$

where $\tilde{b}_i$ is the vector of dimension m formed by the ith row of $\tilde{B}$, i.e.,

$$\tilde{b}_i' = [\,\tilde{b}_{i1}, \ldots, \tilde{b}_{im}]$$

Integrating (3.60) gives

$$\tilde{W}_{ij} = \tilde{b}_i' \tilde{b}_j \left[e^{-(\lambda_i + \lambda_j)T} - 1 \right] / -(\lambda_i + \lambda_j) \qquad i, j = 1, \ldots, n \tag{3.61}$$

Combining (3.59) and (3.61) defines the eigenvector transformation method for computing W .

To illustrate, consider the example

$$A = \begin{bmatrix} -1 & 2 & 0 \\ 0 & -3 & 0 \\ 1 & 2 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1.5 & -1 \\ -4 & 2 \\ -10.5 & -25 \end{bmatrix} \qquad \begin{aligned} T &= 0.5 \\ m &= 2 \\ n &= 3 \end{aligned} \tag{3.62}$$

If the transformation matrix and its inverse are

$$Q = \begin{bmatrix} -2 & -2 & 0 \\ 0.25 & 0 & -0.25 \\ 0 & 1 & 0 \end{bmatrix} \qquad Q = \begin{bmatrix} -0.5 & 0 & -1 \\ 0 & 0 & 1 \\ -0.5 & -4 & -1 \end{bmatrix}$$

then

$$\tilde{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} 5 & -2 \\ 3 & 6 \\ -4 & 2 \end{bmatrix}$$

Substituting into (3.61)

$$\tilde{W} = \begin{bmatrix} 24.92 & 3.48 & -38.33 \\ 3.48 & 71.88 & 0.0 \\ -38.33 & 0.0 & 63.62 \end{bmatrix}$$

and next substituting $\tilde{W}$ into (3.59)

$$W = \begin{bmatrix} 31.51 & -44.45 & 38.48 \\ -44.45 & 63.62 & -44.45 \\ 38.48 & -44.45 & 1195.47 \end{bmatrix} \tag{3.63}$$

Numerical Integration

Let F(t) represent the integrand of (3.43), i.e.,

$$W = \int_0^T F\,dt \tag{3.64}$$

Differentiating F results in the matrix differential equation

$$-\dot{F} = AF + FA'$$

Integrating both sides of the above equation from 0 to t gives the following linear matrix differential equation for computing the controllability Grammian W(t)

$$-\dot{W} = AW + WA' - BB' \quad , \quad W(0) = 0 \tag{3.65}$$

The solution to (3.65) at $t = T$ is the value of the integral (3.43). Similarly, the linear matrix differential equation for computing $W_\ell(t)$ is

$$-\dot{W}_\ell = AW_\ell + W_\ell A' - B_\ell B_\ell' \quad , \quad W_\ell(0) = 0 \tag{3.66}$$
$$\ell = 1, \ldots, m$$

A computer program for calculating the controllability Grammian by numerically integrating (3.65) was developed. The output form the program for the example (3.62) is shown below and required 0.43 seconds of cpu on the IBM 370.

```
MATRIX A

     -0.100000E 01    0.200000E 01    0.0
      0.0            -0.300000E 01    0.0
      0.100000E 01    0.200000E 01   -0.200000E 01


MATRIX B

      0.150000E 01   -0.100000E 01
     -0.400000E 01    0.200000E 01
     -0.105000E 02   -0.250000E 02

MATRIX W

      0.315126E 02   -0.444507E 02    0.384759E 02
     -0.444507E 02    0.636177E 02   -0.444507E 02
      0.384759E 02   -0.444507E 02    0.119545E 04
```

The computer solution of W agrees with the solution (3.63) calculated by hand.

Recursive Algorithm

Suppose the objective is to compute the controllability Grammian for $t = T, 2T, 3T, \ldots, NT$. Let $W(n)$ denote solution of (3.65) for $t + nT$ and define

$$\Omega = \Phi(-T)$$

A recursive algorithm for computing $W(n), n = 2, 3, \ldots, N$ from $W(1)$ and Ω is developed below. From (3.64)

$$W(n+1) = \int_0^{nT+T} F\,dt = \int_{nT}^{nT+T} F\,dt + \int_0^{nT} F\,dt \tag{3.67}$$

Let $\xi = t - nT$ and from

$$\Phi(-t) = \Phi(-\xi - nT) = \Omega^n \Phi(-\xi)$$

it can be shown that

$$\int_{nT}^{nT+T} F(t)\,dt = \int_0^T F(\xi + nt)\,d\xi = \Omega^n \int_0^T F(\xi)\,d\xi\,\Omega^{n'} \tag{3.68}$$

Substituting (3.68) into (3.67) and using the definition (3.43) results in

$$W(n+1) = \Omega^n W(1) \Omega^{n'} + W(n) \tag{3.69}$$

From (3.69) it can be shown by repeated substitution that

$$W(n+1) = \Omega^n W(1) \Omega^{n'} + \ldots + \Omega W(1) \Omega' + W(1) \tag{3.70}$$

From (3.70) it can be readily proven that

$$W(n+1) = \Omega W(n) \Omega' + W(1) \tag{3.71}$$

Formula (3.71) can be used to reduce the amount of numerical integration. To compute $W(t)$ at $t + NT$ instead of numerically integrating (3.65) from 0 to NT , only integrate (3.65) and

$$-\Phi = A\Phi \quad , \quad \Phi(0) = I$$

from 0 to T and then use (3.71)

3.3.2 Index of Controllability

When the general vehicle dynamics are nonlinear, then the linear equations (2.9) for the dynamic response about the trim solution δ_d are a function of δ_d. Hence, the controllability of the linear system (2.9) varies with the choice of the trim solution. Quantification of controllability provides a measure for determining the trim solution that results in the most controllable linear system. In the previous section, the controllability Grammain $W(t)$ at $t = T$ is used evaluate the integral (3.44) for the scalar E which may be viewed as the energy expended by the control effectors in returning the vehicle to trim during a time span of T seconds. One possible means of quantification is the use of E to indicate the degree of controllability. In this section another means of quantification is developed. An index of controllability is defined as the ratio of maximum to minimum eigenvalues of $W(T)$ or some other controllability matrix.

The time-invariant linear system (3.39) is said to be controllable, if it is possible to find an input u which reduces an arbitrary initial state to zero in finite time T. A necessary and sufficient condition for the system to be controllable is that the controllability Grammain $W(t)$ defined by (3.43) be nonsingular for some finite t. If $W \equiv W(T)$ is nonsingular, then (3.42) defines one of the many possible inputs u that satisfy the definition of controllability. Another matrix often used to study controllability is

$$P(t) = \int_0^t \Phi(\tau)BB'\Phi(\tau)dt \qquad (3.72)$$

where $\Phi(t)$ is the transition matrix (3.41). The matrix $P(t)$ is related to $W(t)$ by

$$P(t) = \Phi(t)W(t)\Phi'(t) \qquad (3.73)$$

and can be identified as the covariance matrix of the state $x(t)$ when $u(t)$ is white noise having a spectral density of unity. It follows from (3.73) that the system is controllable if and only if $P(t)$ is nonsingular for some finite t. If the system (3.39) is stable, then the integral for $P(t)$ exists as $t \to \infty$ and the asymptotic value

$$P = \lim_{t \to \infty} P(t) \qquad (3.74)$$

is the solution to the algebraic equation

$$AP + PA' + BB' = 0 \tag{3.75}$$

It is well known that $P(t)$ or $W(t)$ is nonsingular if and only if the matrix

$$K = [B, AB, \ldots, A^{k+1}B] \tag{3.76}$$

has rank k = order of the system. The rank of K is equal to the rank of $k \times k$ symmetric matrix

$$Q = KK' = BB' + ABB'A' + \ldots + A^{k+1}BB'(A')^{k-1} \tag{3.77}$$

which is more convenient than K for testing controllability.

Indices of Controllability

The necessary and sufficient conditions for controllability of a time-invariant system is that a certain matrix be nonsingular. Possible choices of the test matrix that are symmetric, positive-semidefinite include $W(t)$, P, and Q. This controllability is a property that a given system theoretically either possesses or does not possess. In practical applications, however, there may be instances in which a system may be nearly uncontrollable in the sense that certain initial states may be much harder to reduce to zero than others. Evidence of such situations is that the matrices tested for controllability are nearly singular, i.e., poorly-conditioned. It is thus appropriate to use conditioning of a relevant matrix as an index of controllability. A useful measure [4, 5] of the conditioning of a matrix F is

$$k(F) = \|F\| \cdot \|F^{-1}\| \tag{3.78}$$

where $\|F\|$ denotes the norm of the matrix F defined by

$$\|F\| = \sup_{\|x\| = 1} \|Fx\|$$

where $\|x\|$ is a suitable vector norm. When the Euclidian norm, i.e.

$$\|x\| = \sqrt{x'x}$$

is used, then, for a symmetric matrix F,

$$k(F) = | \lambda_{max} / \lambda_{min} | \tag{3.79}$$

where λ_{max} and λ_{min} are the eigenvalues of largest and smallest magnitude, respectively.

Clearly $k(F) \geq 1$ and reaches the lower limit only when $|\lambda_{max}| = |\lambda_{min}|$, i.e. when all eigenvalues are equal in magnitude. The condition $k(F) \geq 1$ also holds for other norms, as shown in [5].

The quantification of controllability (and/or observability) was considered earlier by several investigators. Kalman, Ho and Narendra [5] considered using the trace or the determinant of the inverse of the controllability matrix as indices of controllability, and Johnson [7] considered the determinant as an index of controllability in greater detail.

The shortcoming of the earlier indices of controllability is that they depend on the scale of the variables used in the problem. For example, multiplying earch control variable by a constant say c, is equivalent to multiplying the B matrix by the same constant and hence the controllability matrix Q as defined by (3.77) or P as defined by (3.74) is multiplied by c^2. Hence the trace of P^{-1} or Q^{-1} is multiplied by c^{-2k}. On the other hand the conditioning number is obviously independent of a scale change, either of the control variables or of the state variables. The conditioning number, however, does depend on the choice of state variables, as the following examples indicate.

Example - Consider the system having the transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{s + a}{(s + 1)(s + 2)}$$

It is clear that if $a = 1$ or 2 the system is either not observable or not controllable or both. The objective of this example is to show the behavior of the controllability index as $a \rightarrow 1$ or 2.

In order to examine the controllability and observability of the system it is necessary to define a suitable set of state variables. In this example the state variables are defined as those of two canonical forms. The Jordan normal form and the companion form.

Jordan Form - The Jordan form can be obtained by expanding H(s) in partial fractions:

$$H(s) = \frac{a-1}{s+1} - \frac{a-2}{s+2}$$

Two block diagram representations of H(s) are given in Figure 3.5. For Figure 3.5(a), the state and output equations are

$$\dot{x}_1 = -x_1 + u \qquad A = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\dot{x}_2 = -2x_2 + u$$

$$y = (a-1)x_1 - (a-2)x_2 \qquad C = [a-1, -(a-2)]$$

For Figure 3.5(b) the state and output equations are

$$\dot{x}_1 = -x_1 + (a-1)u \qquad A = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \qquad B = \begin{bmatrix} a-1 \\ -(a-2) \end{bmatrix}$$

$$\dot{x}_2 = -2x_2 - (a-2)u$$

$$y = x_1 + x_2 \qquad C = [\,1 \quad 1\,]$$

It is thus seen that the A matrix of both representations are identical and also $A = A'$. Moreover, B' of Figure 3.5(a) equals C of Figure 3.5(b) and C' of Figure 3.5(a) equals B of Figure 3.5(b). Hence it follows that observability of Figure 3.5(a) corresponds to controllability of Figure 3.5(b), and vice-versa. Accordingly, examining the controllability of Figure 3.5(b) is equivalent to examining the observability of Figure 3.5(a).

The controllability matrix K of the system of Figure 3.5(b) is

$$K = \begin{bmatrix} a-1 & -(a-1) \\ -(a-2) & 2(a-2) \end{bmatrix}$$

Hence

$$Q = KK' = \begin{bmatrix} 2(a-1)^2 & -3(a-1)(a-2) \\ -3(a-1)(a-2) & 5(a-2)^2 \end{bmatrix}$$

The characteristic equation of Q is

$$\lambda^2 - \lambda(\mathrm{tr}\,A) + |A| = 0$$

where

$$\mathrm{tr}\,A = 2(a-1)^2 + 5(a-2)^2$$
$$|A| = (a-1)^2(a-2)^2$$

There is a characteristic root at $\lambda = 0$, for $a = 1$ or $a = 2$, and these are the values of a for which the system is not controllable, as expected. The condition number of Q, as defined above, is

$$k(Q) = \frac{\mathrm{tr}Q + \sqrt{(\mathrm{tr}Q)^2 - 4|Q|}}{\mathrm{tr}Q - \sqrt{(\mathrm{tr}Q)^2 - 4|Q|}}$$

A curve showing the behavior of $k(Q)$ vs the parameter a is shown in Figure (3.6). It is observed that $k(Q)$ tends to infinity as $a \to 1$ or as $a \to 2$. It is interesting to note, however, that $k(Q)$ reaches (local) minima of 37.9 at $a = 1.61$ and $a = 3.72$. This would suggest that if a were adjustable, the controllability (or observability) can be optimized, in the sense of minimizing $k(Q)$ by using $a = 1.61$ or $a = 3.72$.

Instead of $k(Q)$ we can determine $k(P)$ after solving for P by use of (3.77) The solution of the latter is

$$P = \begin{bmatrix} \frac{(a-1)^2}{2} & -\frac{(a-1)(a-2)}{3} \\ -\frac{(a-1)(a-2)}{3} & \frac{(a-2)^2}{4} \end{bmatrix}$$

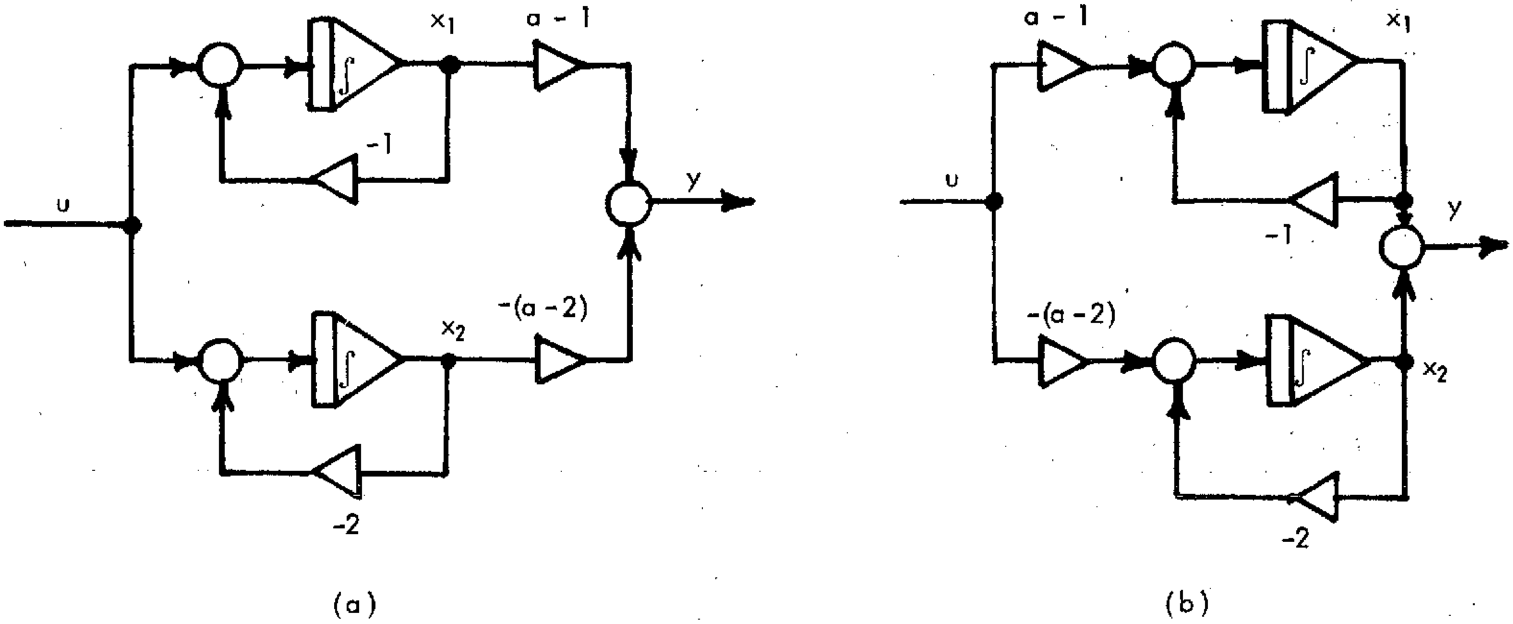


FIGURE 3.5: JORDAN CANONICAL FORMS OF TRANSFER FUNCTION IN EXAMPLE

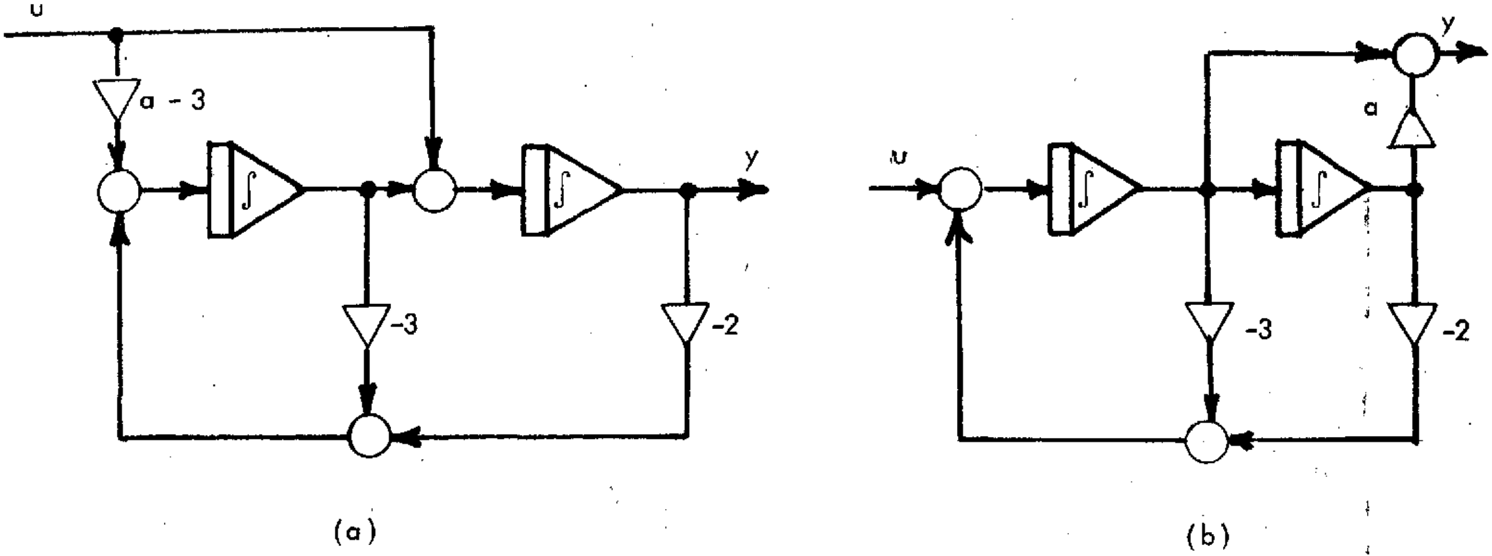


FIGURE 3.8: COMPANION FORM OF TRANSFER FUNCTIONS

whence $\mathrm{tr}P = \frac{(a-1)^2}{2} + \frac{(a-2)^2}{4}$

$$|P| = \frac{1}{72}(a-1)^2(a-2)^2$$

The resulting curve for $k(P)$ is also shown in Fig. 3.6. It is observed that $k(P)$ attains minima of about 34.0 at $a \approx -1.5$ and $a \approx 1.4$.

It is noted that the minimum value of the conditioning number is almost equal for P and Q and one minimum occurs (as expected) between $a = 1$ and $a = 2$. The locations of the other minima are quite different, but the general shapes of the curves are remarkably similar.

It is of interest to examine the effect of adding another independent input on the controllability of the system. Suppose, for example, another input say u_2 was added to the first state, resulting in the equations

$$\dot{x}_1 = -x_1 + (a-1)u_1 + u_2$$

$$\dot{x}_2 = -2x_2 - (a-2)u_1$$

The corresponding B matrix is now

$$B = \begin{bmatrix} a-1 & 1 \\ -(a-2) & 0 \end{bmatrix}$$

The controllability matrix is now

$$K = \begin{bmatrix} a-1 & -1 & -(a-1) & -1 \\ -(a-2) & 0 & 2(a-2) & 0 \end{bmatrix}$$

and

$$Q = KK' = \begin{bmatrix} 2(a-1)^2+2 & -3(a-1)(a-2) \\ -3(a-1)(a-2) & 5(a-2)^2 \end{bmatrix}$$

likewise

$$P = \begin{bmatrix} \frac{(a-1)^2+1}{2} & \frac{-(a-1)(a-2)}{3} \\ -\frac{(a-1)(a-2)}{3} & \frac{(a-2)^2}{4} \end{bmatrix}$$

P and Q are now singular for only one value of a, namely $a = 2$; obviously x_2 is not controllable for $a = 2$.

The curves of $k(P)$ and $k(Q)$ are shown in Fig. 3.7. It is noted that the addition of input u_2 has the effect of reducing the conditioning number for all values of a, as would be expected.

Companion Form - Two alternate companion forms that realize the transfer function H(s) are shown in Fig. 3.8 (a) and (b). The corresponding matrices are as follows

Figure 3.8(a) $$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ a-3 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Figure 3.8(b) $$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} a & 1 \end{bmatrix}$$

Since the C matrix of Fig. 3.8(a) is independent of a it is natural to examine the behavior of this realization for controllability. Likewise, it is natural to examine the realization of Fig. 3.8 (b) for observability.

For the system of Fig. 3.8 (a) it is found that

$$K = \begin{bmatrix} 1 & a-3 \\ a-3 & -3a+7 \end{bmatrix}$$

whence $$Q = KK' = \begin{bmatrix} a^2-6a+10 & -3a^2+17a-24 \\ -3a^2+17a-24 & 10a^2-48a\ 58 \end{bmatrix}$$

and hence $\operatorname{tr} Q = 11a^2 - 54a + 68$

$$|Q| = (a-1)^2 (a-2)^2$$

Solution of (3.75) for P gives

$$P = \begin{bmatrix} \frac{a^2+2}{12} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{a^2-6a+11}{6} \end{bmatrix}$$

with $\operatorname{tr} P = \frac{1}{4}(a^2 - 4a + 8)$

$$|P| = \frac{1}{72}(a-1)^2 (a-2)^2$$

Curves showing $k(P)$ and $k(Q)$ as functions of a are given in Figure 3.9. It is noted that although local minima occur for both $k(P)$ and $k(Q)$ for $1 \leq a \leq 2$, the minima attained exceed 1000 and hence would indicate that operation with a in this interval is undesirable. A very sharp local minimum in $k(Q)$ of about 1.4 occurs at $a \approx 2.7$, and would indicate that operation at this value of a is, in a sense, optimum; $k(P)$ on the other hand does not have any other minimum, but tends to unity as $a \to \pm \infty$. This corresponds to the case in which the "feedforward" gain (to x_1) is negligible in comparison to the direct gain $(a-3)$.

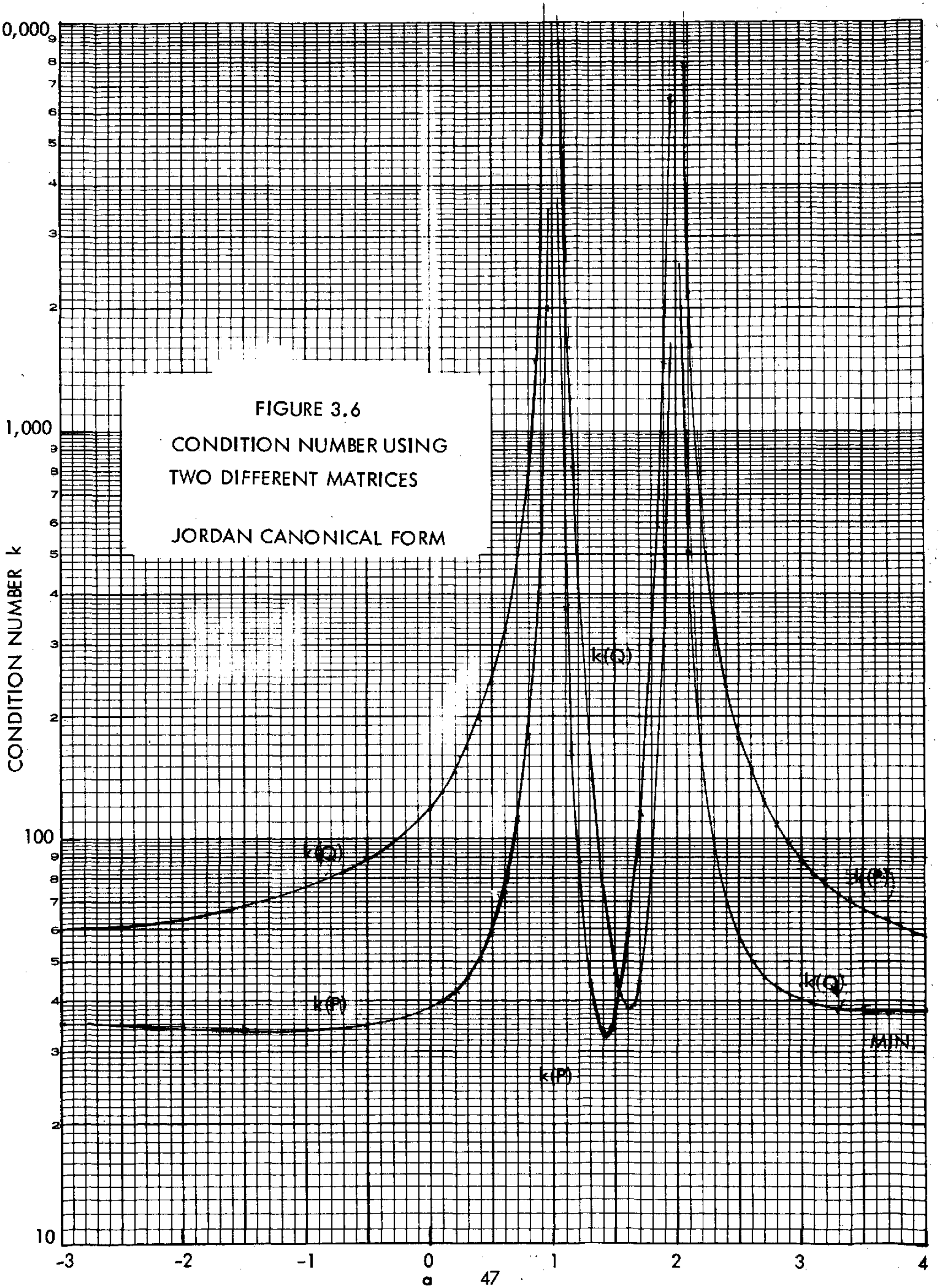


FIGURE 3.6

CONDITION NUMBER USING TWO DIFFERENT MATRICES

JORDAN CANONICAL FORM

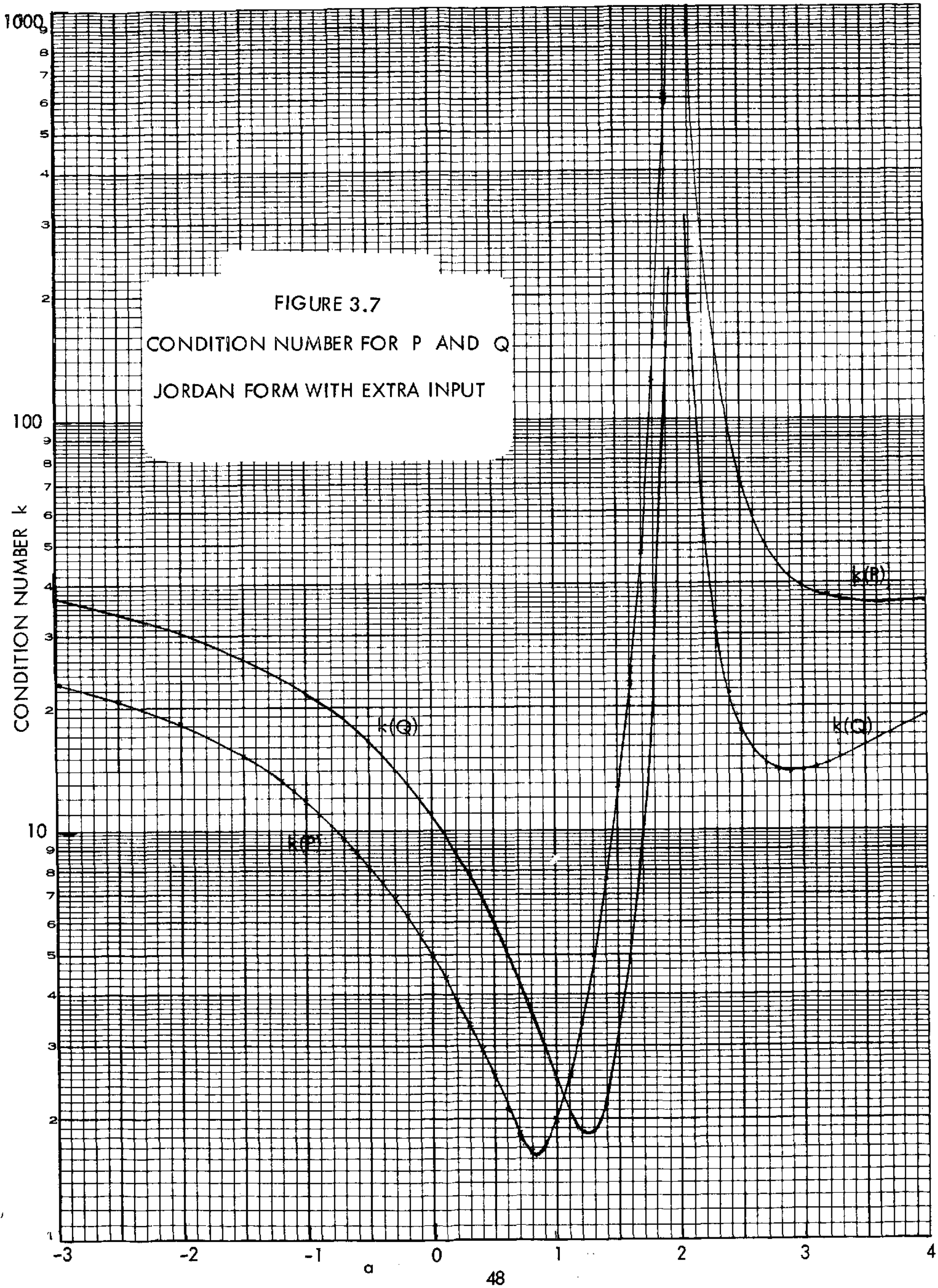


FIGURE 3.7

CONDITION NUMBER FOR P AND Q

JORDAN FORM WITH EXTRA INPUT

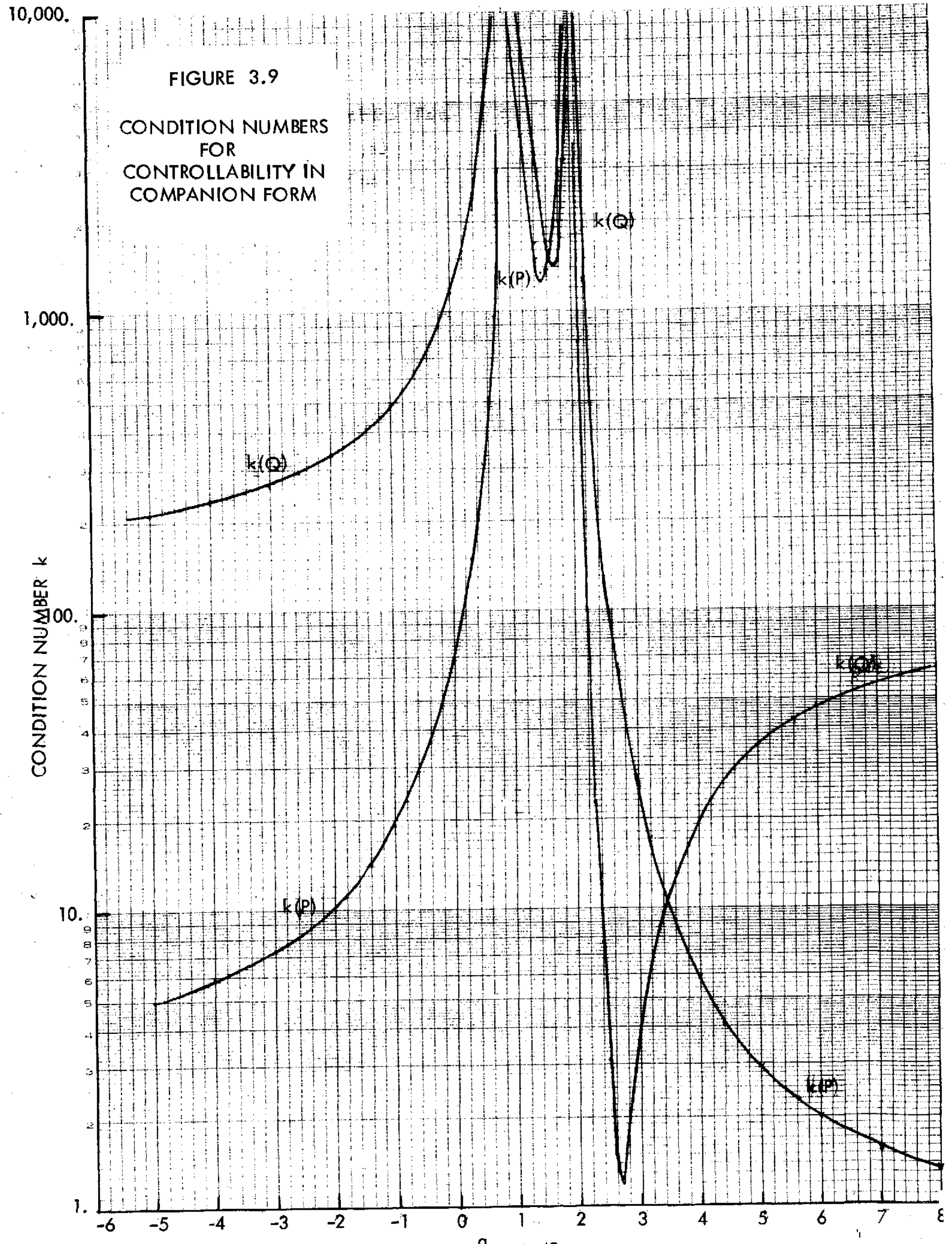


FIGURE 3.9

CONDITION NUMBERS FOR CONTROLLABILITY IN COMPANION FORM

3.4 OPTIMUM CONTROL APPROACH

3.4.1 Optimum Control Computation

If the general control problem described in Section 2.1 can be formulated as an optimum stochastic control problem for a linear process with a quadratic performance criterion than a linear feedback system can be designed to solve both the trim problem and the dynamic response problem. The theory to compute such a feedback system is developed in this section and will be applied in Section 4.3 to the lateral control of the Space Shuttle.

The linear stochastic optimum control problem with bias inputs is defined by the following equations in vector-matrix notation

Process Dynamics:

$$\dot{x} = Ax + Bu + Cz + v \qquad z = \text{constant} \tag{3.80}$$

$$E\{v\} = 0 \qquad E\{vv'\} = V$$

Observation Equation:

$$\begin{aligned} y &= Hx + w \\ E\{w\} &= 0 \qquad E\{ww'\} = W \end{aligned} \tag{3.81}$$

Performance Criterion:

$$J(u) = E\{ \int_t^\infty (x'Qx + \sigma^2 u'Ru)ds \mid y(\tau) \text{ for } \tau \leq t\} \tag{3.82}$$

$$\sigma = \text{scalar parameter}$$

where

x = state vector

u = control vector

y = output vector

z = bias vector

v = input noise vector to process dynamics

w = sensor noise vector

Equation (3.80) is identical to (2.10) except the vector of deflection angles is denoted by u instead of δ . The stochastic optimum control solution is denoted by u in order to distinguish it from the control solution δ obtained by solving the trim problem. This distinction is helpful in the next section when the correlation between u and δ is developed. In the usual problem formulation, the scalar parameter σ is not present in (3.82) since it can be incorporated in the R matrix. In this case, however, the scalar parameter σ is useful in deriving the correlation between u and δ .

If the "bias term" Cz was not present, then the optimum control problem defined by (3.80) - (3.82) would be in the standard form. By defining z as part of the state vector, (3.80) - (3.82) may be rewritten in the standard form. The resulting augmented dynamics are

$$\dot{\overline{x}} = \overline{A}\,\overline{x} + \overline{B}\,u + \overline{v} \tag{3.83}$$

$$y = \overline{H}\,\overline{x} + w \tag{3.84}$$

$$J = E \int_t^{\infty} (\overline{x}'\,\overline{Q}\,\overline{x} + \sigma^2 u'Ru)ds \;\Big|\; y(\tau) \text{ for } \tau \le t\} \tag{3.85}$$

where

$$\overline{x} = \begin{bmatrix} x \\ z \end{bmatrix} \quad \overline{A} = \begin{bmatrix} A & C \\ 0 & E \end{bmatrix} \quad \overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \overline{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

$$\overline{v} = \begin{bmatrix} v \\ f \end{bmatrix} \quad \overline{V} = \begin{bmatrix} V & 0 \\ 0 & \mathcal{F} \end{bmatrix} \quad \overline{H} = \begin{bmatrix} H & 0 \end{bmatrix}$$

The solution to the optimum control problem defined by (3.83) - (3.85) is given by the equations

<u>Deterministic Quadratic Optimum Control:</u>

$$u(t) = -F\hat{\overline{x}}(t) \tag{3.86}$$

where $$F = (1/\sigma^2)R^{-1}\bar{B}'M \tag{3.87}$$

$$M\bar{A} + \bar{A}'M - (1/\sigma^2)M\bar{B}R^{-1}B'M + \bar{Q} = 0 \tag{3.88}$$

Kalman Filter:

$$\dot{\hat{\bar{x}}} = \bar{A}\hat{\bar{x}} + \bar{B}u + K(y-\bar{H}\hat{\bar{x}}) \tag{3.89}$$

where

$$K = P\bar{H}'W^{-1} \tag{3.90}$$

$$0 = \bar{A}P + P\bar{A}' - P\bar{H}'W^{-1}\bar{H}P + \bar{V} \tag{3.91}$$

The feedback control system defined by (3.86) - (3.91) is divided into two parts in tandem. First, a Kalman filter computes the optimum estimate of the augmented state $\hat{\bar{x}}$ from the sensor measurements y . Next, feedback gains multiply the estimated state $\hat{\bar{x}}$ to yield the control signal. In the event that the augmented state vector $\bar{x}$ can be measured perfectly, i.e.,

$$y \equiv \bar{x}$$

then, the Kalman filter is not required. In this case the control system is defined by (3.86) - (3.88) where $\hat{\bar{x}} = \bar{x}$.

Partitioning the augmented state vector into x and z simplifies the equations (3.86) - (3.91) for the control design. The deterministic quadratic optimum control is considered first.

By partitioning the matrix M according to

$$M = \begin{bmatrix} M_1 & M_2 \\ M_2 & M_3 \end{bmatrix}$$

the optimum control solution (3.86) can be rewritten as

$$u(t) = u_x(t) + u_z(t) \tag{3.93}$$

where

$$u_x(t) = - 1/\sigma^2 R^{-1}B'M_1x(t) = - F_x\hat{x}(t)$$

$$u_z(t) = - 1/\sigma^2 R^{-1}B'M_2 z = - F_z\hat{z}(t)$$

The symmetric matrix M_1 is the positive definite solution of

$$M_1A + A'M_1 - 1/\sigma^2 M_1BR^{-1}B'M_1 + Q = 0 \tag{3.94}$$

and the matrix M_2 is computed from M_1 according to

$$M_2 = -(A' - 1/\sigma^2 M_1BR^{-1}B')^{-1}M_1C \tag{3.95}$$

In the derivation of (3.94) and (3.95) it is assumed that $E = 0$ in $\overline{A}$ which corresponds to the assumption $z = \text{constant}$.

Similarly, by partitioning the matrix P according to

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2' & P_3 \end{bmatrix}$$

the equations (3.89) - (3.91) for the Kalman filter become

$$\dot{\hat{x}} = A\hat{x} + Bu + C\hat{z} + K_x(y-Hx) \tag{3.96}$$

$$\dot{\hat{z}} = E\hat{z} + K_z(y-Hx)$$

and

$$\begin{aligned} K_x &= P_1H'W^{-1} \\ K_z &= P_2'H'W^{-1} \end{aligned} \tag{3.97}$$

The partitioning of the P matrix does not simplify the computation of the submatrices P_1 and P_2 as in case of the matrix M . Hence, P_1 and P_2 are computed by solving (3.91) for the positive definite covariance matrix P . In the computation of P it is assumed $E \neq 0$ and $\mathfrak{J} \neq 0$. If $\mathfrak{J} = 0$ then $P_2 = P_3 = 0$. This implies that the bias disturbances z can be determined perfectly which is not realistic. A small amount of damping ($E \neq 0$) is included in the noise model of bias disturbances in order to yield a finite value of P_3 .

3.4.2 Correlation Between Trim Solution and Optimum Control Solution

There is a relationship between the optimum control approach and the trim control approach. This relationship relates the optimum steady state control value $u(\infty)$ to the trim solution δ for the case when the control weighting matrix R in the performance criterion (3.82) of the optimum control approach and in the performance criterion (3.19) of the trim control approach are the same.

The derivation given below is for the case of complete state feedback for which (3.92) holds. It appears that the proof extends to the more general case in which the optimum control system includes the Kalman filter to estimate the state. A detailed proof, however, has not been developed for the more general case.

Substituting (3.93) into (3.80) yields for the case of complete state feedback the closed loop dynamics

$$\dot{x} = \tilde{A}x + \tilde{C}z \tag{3.98}$$

where

$$\tilde{A} = A - 1/\sigma^2 BR^{-1}B'M_1$$

$$\tilde{C} = C - 1/\sigma^2 BR^{-1}B'M_2$$

Since the matrix $\tilde{A}$ is asymptotically stable, setting $\dot{x} = 0$ in (3.98) results in the formula

$$x(\infty) = -\tilde{A}^{-1}\tilde{C}z \tag{3.99}$$

for computing the steady state value of the state vector. In turn, substituting (3.99) and (3.95) into (3.93) gives that the steady state value of the control vector is

$$u(\infty) = u_x(\infty) + u_z(\infty) \tag{3.100}$$

where

$$u_x(\infty) = R^{-1}B'M_1(\sigma^2\tilde{A})^{-1}\tilde{C}z \tag{3.101}$$

$$u_z(\infty) = R^{-1}B'(\sigma^2\tilde{A}')^{-1}M_1Cz \tag{3.102}$$

Next we will consider how $u(\infty)$ varies with the scalar parameter σ in the performance criterion (3.82). In particular what is the limiting value of $u(\infty)$ as σ approaches zero. In determining the limiting solution, we must take into account the variation of the matrices M_1 and M_2 with σ. A solution of (3.94) is sought in the form of a series in ascending power of σ :

$$M_1 = N_0 + \sigma N_1 + \sigma^2 N_2 + \ldots \tag{3.103}$$

In papers by Friedland [8] and Hutton [9], it is shown that the following equations:

$$N_0 B = 0 \tag{3.104}$$

$$\begin{aligned} N_0 A + A' N_0 + Q - N_1 BR^{-1} B' N_1 &= 0 \\ N_1 A + A' N_1 - N_2 BR^{-1} B' N_1 - N_1 BR^{-1} B' N_2 &= 0 \\ \ldots \end{aligned} \tag{3.105}$$

must be satisfied if (3.103) is a solution to (3.94). The above equations are formed by substituting (3.103) into (3.94) and equating matrix coefficients of like powers of σ . By matrix manipulations of (3.104) and (3.105), it is shown in [8] that N_0 is the positive semi-definite solution of

$$\begin{aligned} 0 = {} & N_0 A [I - B(B'QB)^{-1} B'Q] + [I - QB(B'QB)^{-1} B'] A' N_0 + Q - QB(B'QB)^{-1} B'Q \\ & - N_0 AB(B'QB)^{-1} B'A' N_0 \end{aligned} \tag{3.106}$$

After solving (3.106) for N_0 , we can solve (3.105) for the positive semi-definite matrix N_1 .

Consider the asymptotic value of

$$(\sigma^2 \widetilde{A})^{-1} = (\sigma^2 A - BR^{-1} B' M_1)^{-1} \tag{3.107}$$

as σ approaches zero. For all nonzero σ , the matrix M_1 is positive definite. From (3.104), the matrix M_1 is positive semi-definite at $\sigma = 0$. However, if the first term $\sigma^2 A$ in (3.107) decays to zero more rapidly than the second term, then

$$(\sigma^2 \widetilde{A})^{-1} \to - M_1^{-1}(BR^{-1}B')^{-1} \text{ as } \sigma \to 0 \tag{3.108}$$

provided $BR^{-1}B'$ is positive definite. Substituting (3.103) into (3.107) and using (3.104) gives that

$$[\sigma^2 \widetilde{A}]^{-1} = [\sigma^2 A - BR^{-1}B'(\sigma N_1 + \sigma^2 N_2 + \ldots)]^{-1} \tag{3.109}$$

The dominant term in (3.109) is $BR^{-1}B'N_1$ which is derived from the second term in (3.107) and indicates that (3.108) is valid.

Substituting (3.108) into (3.95) and (3.98) gives

$$\lim_{\sigma \to 0} M_2 = - (BR^{-1}B')^{-1}C \tag{3.110}$$

$$\lim_{\sigma \to 0} \widetilde{C} = 0 \tag{3.111}$$

Further substituting (3.108) into (3.101) and (3.102) and using (3.111) yields the results

$$\lim_{\sigma \to 0} u(\infty) = \lim_{\sigma \to 0} u_z(\infty) = - R^{-1}B'(BR^{-1}B')^{-1}Cz \tag{3.112}$$

$$\lim_{\sigma \to 0} u_x(\infty) = 0 \tag{3.113}$$

The trim control problem is to find the set of controls δ satisfying

$$0 = Cz + B\delta \tag{3.114}$$

and minimizing the performance index

$$J = 1/2\, \delta' \overline{R} \delta \tag{3.115}$$

The solution to (3.114) and (3.115) is

$$\delta = -\overline{R}^{-1}B(B\overline{R}^{-1}B')^{-1}Cz \tag{3.116}$$

Comparing (3.116) to (3.112) provides the fundamental result that

$$\delta = \lim_{\sigma \to 0} u(\infty) \quad \text{if} \quad \overline{R} = k^2 R \tag{3.117}$$

where k is an arbitrary scalar. Thus the steady state value of the optimum control solution in the case of unlimited control authority (control weighting matrix R in the performance criterion goes to zero) is equal to the trim solution provided the relative control weighting matrices are the same in both cases. This provides a correlation between the optimum control solution and the trim solution.

4. SPACE SHUTTLE CONTROL

Control of the Space Shuttle is studied during ascent when more control effectors are available than required. The analytical methods developed in Section 3 are applied to the lateral control problem. An illustration of the Space Shuttle configuration, given in Figure 4.1, shows the two aerodynamic surfaces and five rocket engines available for control. For purposes of later reference these controls are identified as follows:

1) top orbiter rocket engine
2) right orbiter rocket engine
3) left orbiter rocket engine
4) right solid rocket motor
5) left solid rocket motor
6) aileron
7) rudder

By varying the angular position of these controls, seven independent means of lateral control are achieved. But only three independent controls are required, leaving four redundant controls. If the solid rocket motors (SRM) are not gimballed then the number as independent controls is reduced to five, leaving two redundant controls. The results in this report are for the latter case. However, the equations and computer programs used to perform the calculation of the control deflections include the possibility of gimballing the SRM.

4.1 Space Shuttle Dynamics

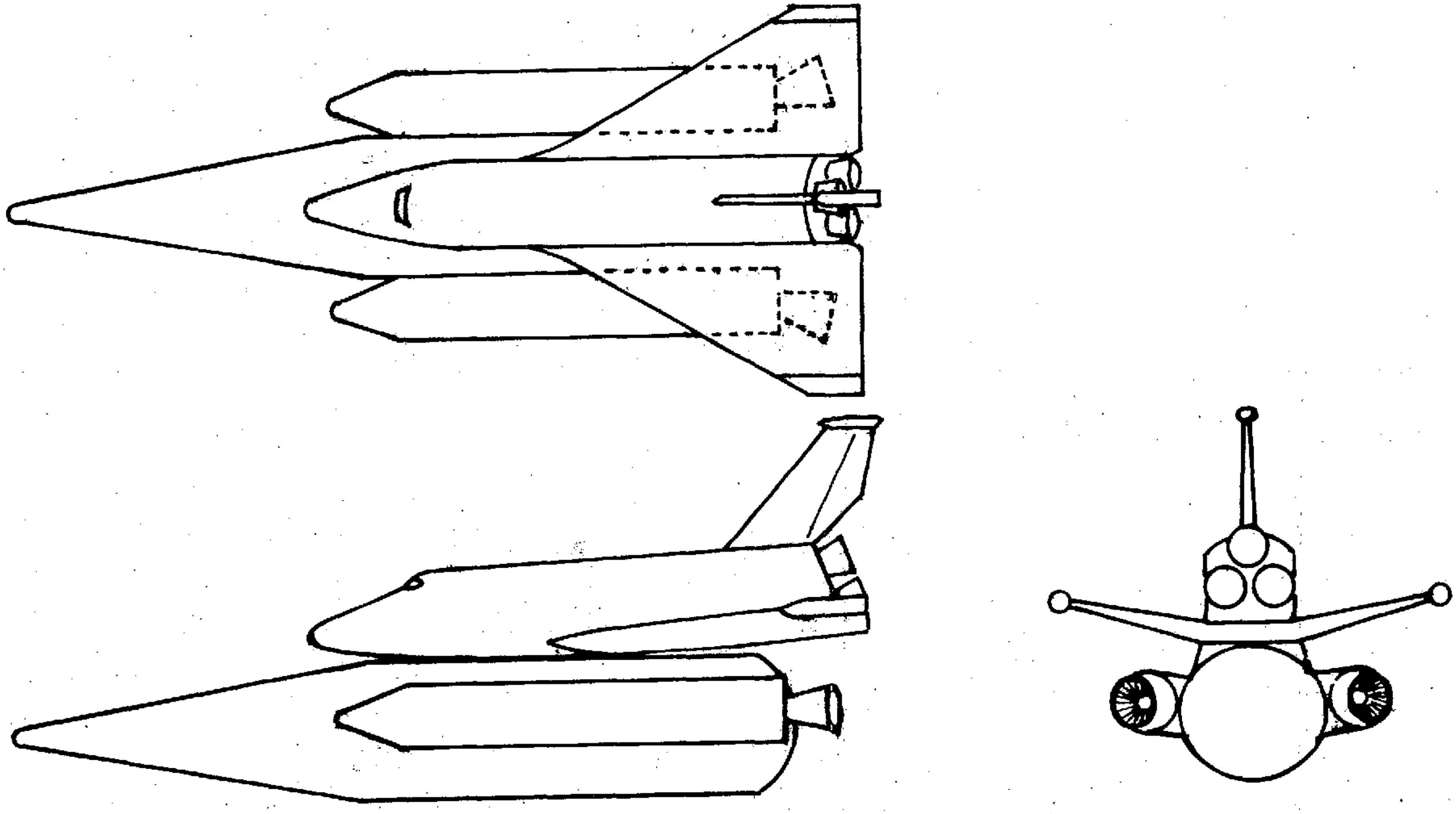
A mathematical model describing the lateral motion of the Space Shuttle is given in this section. This description entails an extensive number of the parameters defined in Appendix B together with a tabulation of their numerical values.

The set of differential equations describing the translational and rotational motion of the vehicle are based on summing the forces and moments along the body axes of the vehicle *. The body axes are defined as a Cartesian coordinate system fixed to the vehicle and whose origin is located at the center of mass as shown in Figure 4.2. The attitude and rotational rate

---

* The notation and definitions used for the aerodynamic terms in the report are in accordance with [1].

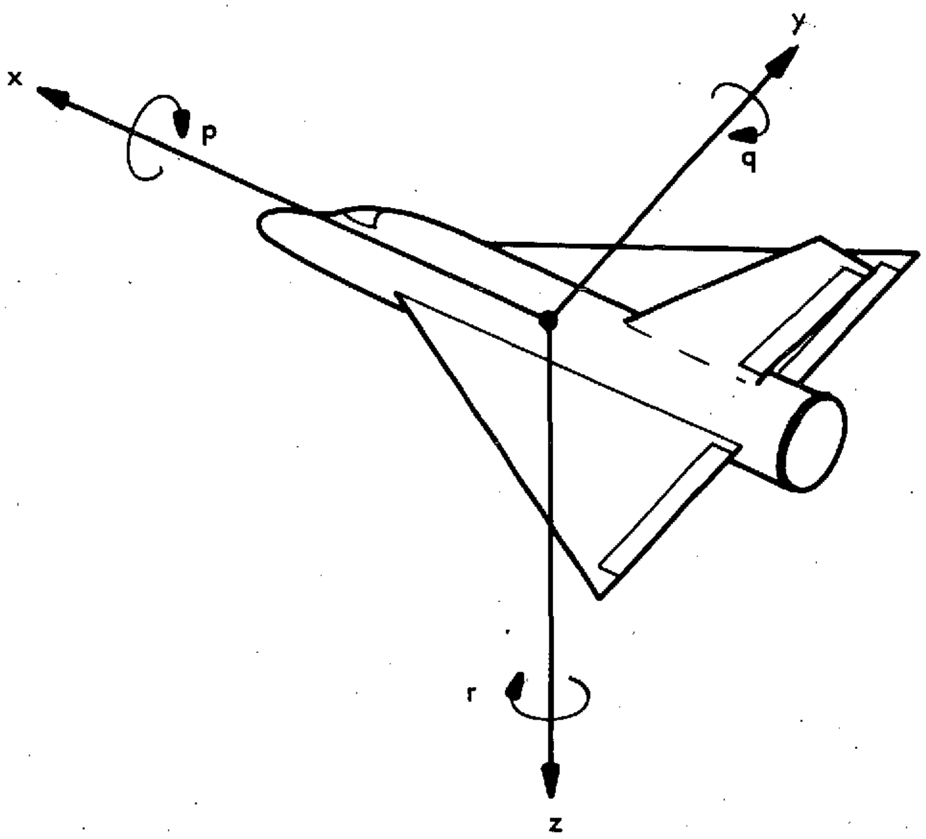
Figure 4.1 Ascent Control Configuration of Space Shuttle



of the vehicle are defined by the Euler angles and the components of the angular velocity vector along the three body axes. Specifically

φ = roll angle
θ = pitch angle
ψ = yaw angle
p = roll rate
q = pitch rate
r = yaw rate

Figure 4.2 Body Axes and Notation



The positive directions are as shown in Figure 4.2. Upper case letters denote the total motion (Nominal and Perturbation) of the vehicle. The velocities, forces, and moments about the three body axes are defined as follows:

U, X = forward velocity and force
V, Y = side velocity and force
W, Z = downward velocity and force
L = rolling moment
M = pitching moment
N = yawing moment

The kinematic and dynamic equations describing the lateral motion of the vechile are

$$Y = m[V + RU - PW - g\cos\Theta\ \sin\Phi]$$

$$L = I_x\dot{P} - I_{xz}\dot{R} + QR(I_z - I_y) - I_{xz}PQ \qquad (4.1)$$

$$N = -I_{xz}\dot{P} + I_z\dot{R} + PQ(I_y - I_x) + I_{xz}QR$$

and

$$\dot{\Phi} = P + Q\sin\Phi\tan\theta + R\cos\Phi\tan\Theta$$

$$\qquad (4.2)$$

$$\dot{\Psi} = (Q\sin\Phi + R\cos\Phi)\sec\Theta$$

where the moments of inertia are defined by

$$I_x = \int(y^2+z^2)\,dm \qquad I_y = \int(x^2+z^2)\,dm$$

$$\qquad (4.3)$$

$$I_z = \int(x^2+y^2)\,dm \qquad I_{xz} = \int xz\,dm$$

For this investigation no data was available on I_{xz}, thus the approximation $I_{xz} = 0$ is used. The (total) vehicle motion modeled by (4.1) and (4.2) can be partitioned into nominal plus perturbation motion by substituting

$$U = U_o + u \qquad P = P_o + p \qquad \Theta = \Theta_o + \theta$$

$$V = V_o + v \qquad Q = Q_o + q \qquad \Phi = \Phi_o + \varphi \qquad (4.4)$$

$$W = W_o + w \qquad R = R_o + r \qquad \Psi = \Psi_o + \psi$$

where the capital letters with subscript "o" denote the nominal motion and the lower case letters denote the perturbation motion. For the nominal motion along the trajectory it is assumed that

$$\begin{array}{lll} U_o \neq 0 & P_o = 0 & \Theta_o \neq 0 \\ V_o = 0 & R_o = 0 & \Phi_o = 0 \\ W_o \neq 0 & Q_o \neq 0 & \Psi_o = 0 \end{array} \qquad (4.5)$$

The nonzero values are tabulated in Appendix B for each of the twelve flight times along the ascent trajectory for which the perturbation motion is to be studied. Substituting (4.4) and (4.5) into (4.1) and (4.2) results in the following linearized equations of motion for small perturbations from the nominal trajectory:

$$\begin{aligned} Y &= m[\dot{v}+U_o r - g\cos\Theta_o \varphi] \\ L &= \mathcal{I}_x \dot{p} + (\mathcal{I}_z - \mathcal{I}_y) Q_o r \\ N &= \mathcal{I}_z \dot{r} + (\mathcal{I}_y - \mathcal{I}_x) Q_o p \\ \dot{\varphi} &= p + Q_o \tan\Theta_o \varphi + \tan\Theta_o r \\ \dot{\psi} &= (Q_o \varphi + r)\sec\Theta_o \end{aligned} \qquad (4.6)$$

Adding the equation $\dot{y} = v$ to (4.6) and reqriting in the state space formulation results in the vector-matrix equation

$$\dot{x} = \bar{A}x + \bar{B}f \qquad (4.7)$$

where the state and forcing vectors are

$$x = [y, \varphi, \psi, v, p, r]'$$

and

$$f = [Y, L, N]'$$

The constant matrix $\bar{B}$ has the form

$$\bar{B} = \begin{array}{cc} \overset{3}{\longleftrightarrow} & \\ \left[\begin{array}{c} 0 \\ \text{- - -} \\ \Lambda^{-1} \end{array}\right] & \begin{array}{l} \updownarrow 3 \\ \\ \updownarrow 3 \end{array} \end{array}$$

where

$$\Lambda = \text{Diag}\,[\,m\,,\,I_x\,,\,I_z\,] \tag{4.8}$$

The forcing vector f represents the lateral forces and moments acting on the vehicle and can be modeled by

$$f \equiv \begin{bmatrix} Y \\ L \\ N \end{bmatrix} = \underset{\text{aerodynamic forces and moments}}{\begin{bmatrix} Y_v & Y_p & Y_r \\ L_v & L_p & L_r \\ N_v & N_p & N_r \end{bmatrix}\begin{bmatrix} v \\ p \\ r \end{bmatrix}} + \underset{\substack{\uparrow \\ \text{control} \\ \text{forces} \\ \text{and} \\ \text{moments}}}{\tilde{B}\delta} + \underset{\substack{\uparrow \\ \text{bias disturbance} \\ \text{forces and moments}}}{\tilde{C}z} \tag{4.9}$$

Substituting (4.9) into (4.7) gives the desired vector-matrix equation for the dynamics of the vehicle

$$\dot{x} = Ax + B\delta + Cz \tag{4.10}$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 1 & a_{26} \\ 0 & a_{32} & 0 & 0 & 0 & a_{36} \\ 0 & a_{42} & 0 & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} \end{bmatrix} \tag{4.11}$$

$a_{22} = Q_o \tan \Theta_o$ $\qquad$ $a_{26} = \tan \Theta_o$

$a_{32} = Q_o \sec \Theta_o$ $\qquad$ $a_{36} = \sec \Theta_o$

$a_{42} = g \cos \Theta_o$

$a_{44} = Y_v$ $\qquad$ $a_{45} = Y_p$ $\qquad$ $a_{46} = Y_r - U_o$

$a_{54} = L_v$ $\qquad$ $a_{55} = L_p$ $\qquad$ $a_{56} = L_r + Q_o(I_y - I_z)/I_x$

$a_{64} = N_v$ $\qquad$ $a_{65} = N_p + Q_o(I_x - I_y)/I_z$ $\qquad$ $a_{66} = N_r$

and

$$B = \begin{array}{cc} \xleftrightarrow{\;3\;} & \\ \left[\begin{array}{c} 0 \\ \hdashline \Lambda^{-1}\tilde{B} \end{array}\right] & \begin{array}{l} 3 \\ 3 \end{array} \end{array} \qquad C = \begin{array}{cc} \xleftrightarrow{\;3\;} & \\ \left[\begin{array}{c} 0 \\ \hdashline \Lambda^{-1}\tilde{C} \end{array}\right] & \begin{array}{l} 3 \\ 3 \end{array} \end{array} \tag{4.12}$$

In the remainder of this section, the formulas for computing the matrix elements in (4.9), which are required for (4.10), are developed.

The formulas for the matrix elements corresponding to the aerodynamic forces and moments are

$$\begin{array}{lll} Y_v = Q\,C_{y\beta}/U_o & Y_p = 0 & Y_r = Q\bar{c}\,C_{yr}/2\,U_o \\ L_v = Q_x C_{\ell\rho}/U_o & L_p = Q_x bC_{\ell p}/2U_o & L_r = Q_x bC_{\ell r}/2U_o \\ N_v = Q_z C_{n\beta}/U_o & N_p = Q_z bC_{rp}/2U_o & N_r = Q_z bC_{nr}/2U_o \end{array} \tag{4.13}$$

where

$$Q = qS/m \qquad Q_x = qSb/\mathcal{I}_x \qquad Q_z = qSb/\mathcal{I}_z$$

q = dynamic pressure

U_o = nominal velocity in x-direction

S = reference area

b = reference length

$\bar{c}$ = length of mean aerodynamic cord

Next the expressions for the forces and moments generated by gimballing the rocket engines are derived. The location and nominal direction of each rocket engine with respect to the Cartesian coordianate system fixed to the vehicle is shown in Figure 4.3 *. The rocket engines are numbered 1 through 5 as indicated in Figure 4.3 and in agreement with the list of controls at the beginning of Section 4. Let x_ℓ, y_ℓ, z_ℓ denote the coordinates of the

---

* The location of SRM was not included in the information received from MSFC. This data was not required since it was assumed the SRM could not be gimballed. However, the equations and corresponding computer programs include the posibility of gimballing the SRM.

vehicle center of gravity where $y_{cg} = 0$. The (position) vector from the center of gravity to the ℓth rocket engine is, therefore,

$$[x_{cg} + x_{\ell}\,,\ y_{\ell}\,,\ z_{cg} + z_{\ell}] \tag{4.14}$$

The thrust vector with magnitude F_{ℓ} has the components

$$\begin{aligned}
&\text{(forward)} && X_{\ell} = F_{\ell}(\cos\theta_{\ell}\cos\psi_{\ell} - \cos\theta_{\ell}\sin\psi_{\ell}\delta_{ey\ell} - \sin\theta_{\ell}\cos\psi_{\ell}\delta_{ep\ell}) \\
&\text{(sideward)} && Y_{\ell} = F_{\ell}(\cos\theta_{\ell}\sin\psi_{\ell} + \cos\theta_{\ell}\cos\psi_{\ell}\delta_{ey\ell} - \sin\theta_{\ell}\sin\psi_{\ell}\delta_{ep\ell}) \\
&\text{(downward)} && Z_{\ell} = F_{\ell}(\sin\theta_{\ell} + \cos\theta_{\ell}\delta_{ep\ell})
\end{aligned} \tag{4.15}$$

where the angles defining the direction of the thrust vector are

θ_{ℓ} = nominal pitch angle of the ℓth rocket engine.

ψ_{ℓ} = nominal yaw angle of the ℓth rocket engine.

$\delta_{ep\ell}$ = pitch deflection of the ℓth rocket engine.

$\delta_{ey\ell}$ = yaw deflection of the ℓth rocket engine.

as shown in Figure 4.4. The arrows in Figure 4.4 indicate the directions of positive angles. The nominal directions of the rocket engines are shown in Figure 4.3 and listed in Table 4.1. The derivation of (4.15) assumes that the deflection angles are small.

Table 4.1 Nominal Directions of the Rocket Engines

		θ_{ℓ}	ψ_{ℓ}
	1	- 18°	0
	2	- 12°	- 3.5°
index ℓ	3	- 12°	3.5°
	4	0	- 15°
	5	0	15°

Figure 4.3 Location and Nominal Direction of Rocket Engines

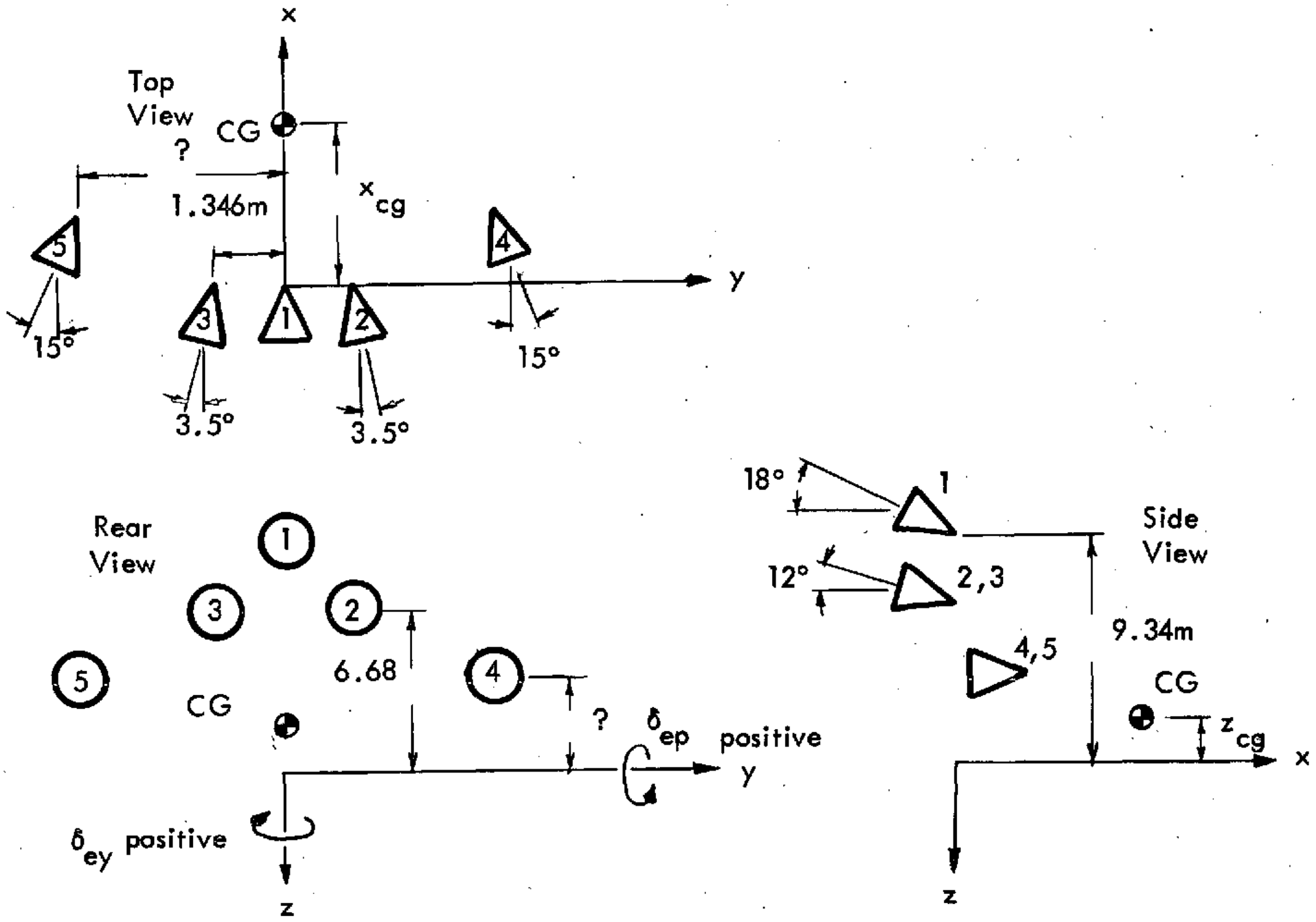
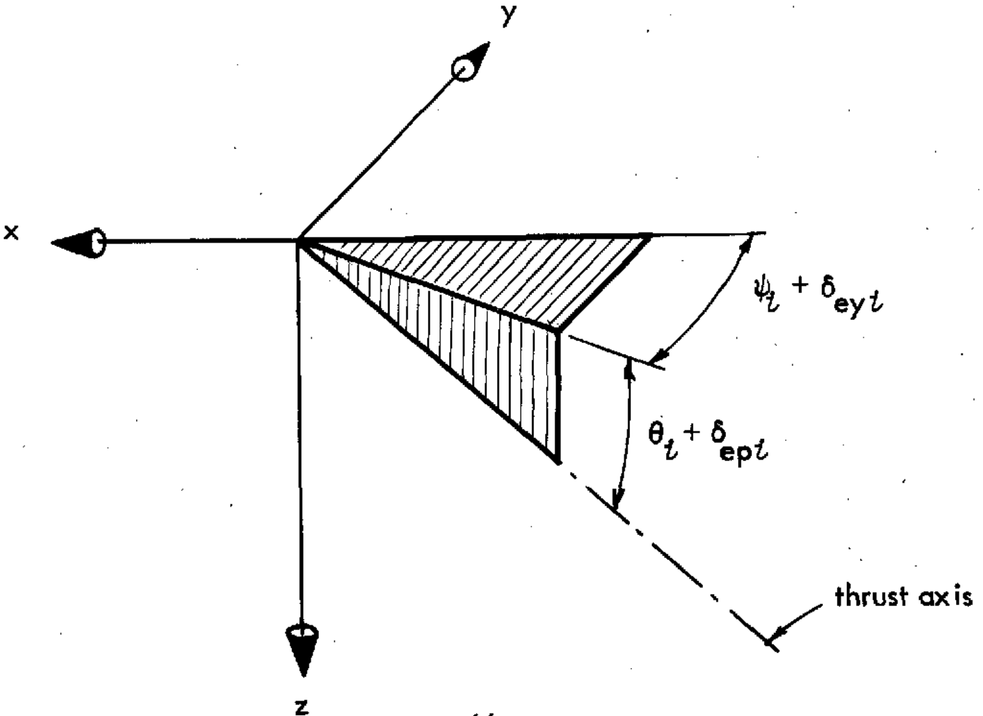


Figure 4.4 Angular Direction of Thrust Axis



The moments induced by the ith rocket engine are given by the cross product of the position vector (4.14) with the thrust vector which results in

$$\begin{aligned} &\text{(roll)} && L_i = y_i Z_i - (z_{cg}+z_i) Y_i \\ &\text{(pitch)} && M_i = (z_{cg}+z_i) X_i - (x_{cg}+x_i) Z_i \\ &\text{(yaw)} && N_i = (x_{cg}+x_i) Y_i - y_i X_i \end{aligned} \tag{4.16}$$

Substituting (4.15) into (4.16) expresses the moments as a linear function of the deflection angles.

Having derived the general equations for modeling the rocket engines, the next step is to derive the equations corresponding to the term $\tilde{B}\delta$ in (4.9).

The elements of the control vector are

$$\begin{aligned} \delta_1 &= \delta_{ey1} \\ \delta_2 &= \delta'_{ey2} = \frac{1}{2}(\delta_{ey3}+\delta_{ey2}) \\ \delta_3 &= \delta'_{ep3} = \frac{1}{2}(\delta_{ep3}-\delta_{ep2}) \\ \delta_4 &= \delta'_{ey4} = \frac{1}{2}(\delta_{ey5}+\delta_{ey4}) \\ \delta_5 &= \delta'_{ep5} = \frac{1}{2}(\delta_{ep5}-\delta_{ep4}) \\ \delta_6 &= \delta_a \\ \delta_7 &= \delta_r \end{aligned} \tag{4.17}$$

where the deflection angles are defined as follows:

δ_{ey1} = yaw angle of top orbiter engine

δ_{ey2} = yaw angle of right orbiter engine

δ_{ep2} = pitch angle of right orbiter engine

δ_{ey3} = yaw angle of left orbiter engine

δ_{ep3} = pitch angle of left orbiter engine

δ_{ey4} = yaw angle of right SRM

δ_{ep4} = pitch angle of right SRM

δ_{ey5} = yaw angle of left SRM

δ_{ep5} = pitch angle of left SRM

The elements of the constant 7 x 3 matrix

$$\widetilde{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} \end{bmatrix} \qquad (4.18)$$

are computed from the following set of formulas:

$$b_{11} = F \cos 18°$$

$$b_{21} = - F(z_1 - z_{cg}) \cos 18°$$

$$b_{31} = F(x_1 - x_{cg}) \cos 18°$$

$$b_{12} = 2F \cos 12° \cos 3.5°$$

$$b_{22} = - 2F(z_2 - z_{cg}) \cos 12° \cos 3.5°$$

$$b_{32} = 2F [(x_2 - x_{cg}) \cos 3.5° - y_2 \sin 3.5°] \cos 12°$$

$$b_{13} = 2F \sin 12° \cos 3.5°$$

$$b_{23} = 2F [y_2 \cos 12° - (z_2 - z_{cg}) \sin 12° \sin 3.5°]$$

$$b_{33} = 2F [y_2 \cos 3.5° + (x_2 - x_{cg}) \sin 3.5°] \sin 12°$$

(4.19)

$$b_{14} = 2F_{SRM}\cos 15°$$
$$b_{24} = 2F_{SRM}(z_4 - z_{cg})\cos 15°$$
$$b_{34} = 2F_{SRM}[(x_4 - x_{cg})\cos 15° - y_4 \sin 15°]$$

$$b_{15} = 0$$
$$b_{25} = 2F_{SRM}y_4$$
$$b_{35} = 0$$

$$b_{16} = qSC_{y\delta_a}$$
$$b_{26} = qSb_{ref}(C_{\ell\delta_a})_{cg}$$
$$b_{36} = qSb_{ref}(C_{n\delta_a})_{cg}$$

(4.19 continued)

$$b_{17} = qSC_{y\delta_r}$$
$$b_{27} = qSb_{ref}(C_{\ell\delta_r})_{cg}$$
$$b_{37} = qSb_{ref}(C_{n\delta_r})_{cg}$$

$$(C_{\ell\delta_a})_{cg} = C_{\ell\delta_a} + C_{y\delta_a}(z_{cg} - z_{mrp})/b_{ref}$$
$$(C_{n\delta_a})_{cg} = C_{n\delta_a} - C_{y\delta_a}(x_{cg} - x_{mrp})/b_{ref}$$
$$(C_{\ell\delta_r})_{cg} = C_{\ell\delta_r} + C_{y\delta_r}(z_{cg} - z_{mrp})/b_{ref} \qquad (4.20)$$
$$(C_{n\delta_r})_{cg} = C_{n\delta_r} - C_{y\delta_r}(x_{cg} - x_{mrp})/b_{ref}$$

The formulas in (4.19) are grouped by column. The ith column of the $\tilde{B}$ matrix in (4.9) defines the values of Y, L, N corresponding to δ_i. The formulas for the first five columns are derived from (4.14) - (4.17). The last two columns corresponding to the aileron and rudder, respectively, are computed using the standard formulas for aerodynamic control surfaces. The data for the stability derivatives received from MSFC were with respect to the moment reference point located at x_{mrp}, y_{mrp}, z_{mrp} where $y_{mrp} = 0$. The translation of data from the moment reference point to the center of gravity is given by (4.20).

The force and moments in (4.9) due to the bias disturbances is modeled by the term $\tilde{C}z$. The elements of the vector z or bias inputs are

$z_1 = \beta$ = side slip angle due to a steady side wind

$z_2 = T_{r_b}$ = roll bias torque due to SRM misalignment

$z_3 = T_{y_b}$ = yaw bias torque due to SRM misalignment

The constant 3x3 matrix $\tilde{C}$ has the form

$$\tilde{C} = \begin{bmatrix} C_{11} & 0 & 0 \\ C_{21} & 1 & 0 \\ C_{31} & 0 & 1 \end{bmatrix} \tag{4.21}$$

where the elements in the first column are computed from

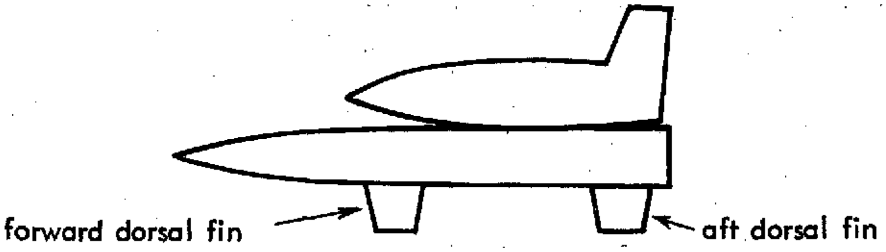
$$C_{11} = qSC^*_{y\beta}\beta$$

$$C_{21} = qSb(C^*_{\ell\beta})_{cg}$$

$$C_{31} = qSb(C^*_{n\beta})_{cg}$$

$$\begin{aligned} (C^*_{\ell\beta})_{cg} &= C^*_{\ell\beta} + C^*_{y\beta}(z_{cg} - z_{mrp})/b \\ (C^*_{n\beta})_{cg} &= C^*_{n\beta} - C^*_{y\beta}(x_{cg} - x_{mrp})/b \\ C^*_{y\beta} &= C_{y\beta} + \Delta C_{y\beta} \\ C^*_{\ell\beta} &= C_{\ell\beta} + \Delta C_{\ell\beta} \\ C^*_{n\beta} &= C_{n\beta} + (\Delta C_{n\beta})_{AFT} + (\Delta C_{n\beta})_{FORWARD} \end{aligned} \tag{4.22}$$

The last three equations in (4.22)account for the change in the stability derivatives due to the addition of a pair of dorsal fins to the Space Shuttle configuration as indicated in the sketch below.



To summarize, the lateral dynamics of the space shuttle is governed by the vector matrix equation (4.10). The coefficient matrices A , B , C in (4.10) are computed using (4.8), (4.9), (4.11) - (4.13), (4.18)- (4.22). The values of the parameters required by these equations are given in Appendix B.

4.2 TRIM PROBLEM AND SOLUTION

When bias disturbances generate forces and moments that cause the vehicle to deviate from the nominal tajectory, the rocket and aerodynamic controls must be deflected in such a way as to counterbalance these forces and moments. For the lateral trim problem of the space shuttle, the bias disturbances are due primarily to steady side winds and SRM misalignments. The state vector x in (4.7) defines the deviation of the vehicle from the nominal trajectory in the lateral-direction. Hence, the trim condition is to maintain $x = 0$. On substituting $x = 0$ into (4.10) one finds that the trim solution δ must satisfy the matrix linear equation

$$0 = B\delta + Cz \tag{4.23}$$

For a given value of the bias vector z , (4.23) represents six equations in seven unknowns. However, the equations are not all linearly independent. From (4.12) the first three equations are identically zero independent of δ and the last three equations have the form

$$0 = \Lambda^{-1}\tilde{B}\delta + \Lambda^{-1}\tilde{C}z \tag{4.24}$$

where Λ is the diagonal matrix defined by (4.8). Premultiply (4.24) by Λ gives

$$0 = \tilde{B}\delta + \tilde{C}z \tag{4.25}$$

which is equivalent to setting $Y = L = N = 0$ in (4.9). In other words, (4.25) states that the trim control must provide zero net side force, rolling moment, and yawing moment in the presence of a steady side wind and SRM misalignments. Replacing (4.23) by (4.25) has reduced the number of trim equations from six to three. In terms of the notation introduced in Section 2.1, the dimensions of the trim problem are

$m = 7$: number of controls
$n = 6$: number of state variables
$\tilde{n} = 3$: number of linearly independent trim equations

In order to determine the optimum trim solution, a performance criterion of the following form was selected:

$$r(\delta) = \frac{1}{2} \sum_{i=1}^{7} W_{1i}^2 (\delta_i / \delta_{i\,max})^2 + \sum_{i=1}^{5} W_{2i}^2 (1 - \cos \delta_i) + \frac{1}{2} \sum_{i=6}^{7} W_{2i}^2 (q S_i C_{Di} \delta_i)^2 \tag{4.26}$$

where

$\delta_{i\,max}$	Maximum deflection angle allowed for the ith control because of physical limitations or excessive hinge moments.
q	Dynamic pressure
S_i	Reference area corresponding to the drag induced by the ith control
C_{Di}	Coefficient of drag corresponding to the ith control.

The numerical values of the above parameters is given in Appendix B.

The seven components of the vector δ of control deflections are defined according to (4.17). The first term in (4.26) penalizes the movement of the actuators for trim in order to leave maximum flexibility for dynamic response. The second term in (4.26) penalizes the thrust loss(gain) caused by gimballing the rocket engines away from their nominal position. The third term in (4.26) penalizes the thrust loss due to drag caused by deflecting aerodynamic surfaces.

Substituting the approximation

$$1 - \cos \delta_i \approx \frac{1}{2} \delta_i^2$$

into (4.26), the performance criterion can be written as the quadratic form

$$r(\delta) = \frac{1}{2}\delta' R \delta \qquad (4.27)$$

where R is a diagonal matrix whose elements are given by

$$R_{ii} = W_{1i}^2 / \delta_{i\,max}^2 + W_{2i}^2 \qquad i = 1, \ldots, 5 \qquad (4.28)$$

$$R_{ii} = W_{1i}^2 / \delta_{i\,max}^2 + W_{2i}^2 (q S_i C_{Di})^2 \qquad i = 6, 7 \qquad (4.29)$$

The fourteen (relative) weighting factors W_{1i} and W_{2i} are selected by the user to achieve the best performance within the restriction imposed by the problem. This best performance is a judgement evaluation unless additional criteria are used.

The lateral trim deflection angles are the solution to the optimization problem defined by (4.25) and (4.27). The objective is to solve the trim problem for the maximum expected values of sideslip angle and for different combinations of roll and yaw misalignment torques that encompass the worse case situation. The sideslip angle is computed from the mean side wind velocity and the vehicle velocity according to

$$\beta = \sin^{-1}(V_y/V)$$

The values of V_y and V for each of the twelve trajectory points are listed in Appendix B and result in the values of sideslip angle listed in Table 4.2. Plotting the values of β as a function of flight time yields the sideslip profile shown in Figure 4.5. Eight different combinations of yaw and roll bias torques due to SRM misalignments were provided by MSFC for studying the trim problem and these are listed in Table 4.3.

A computer program entitled TRIMS for computing lateral trim of the Space Shuttle was developed. The TRIMS program solves the trim problem given by (4.25) and (4.27) using the numerical methods described in Section 3.1. The program user can select either the steepest descent method or the Newton-Raphson method at execution time. Although the trim problem given by (4.25) and (4.26) is linear, these numerical methods have the capability to solve the nonlinear problem. The TRIMS program is coded to facilitate changes in the trim problem including the replacement of the linear trim problem by a nonlinear trim problem.

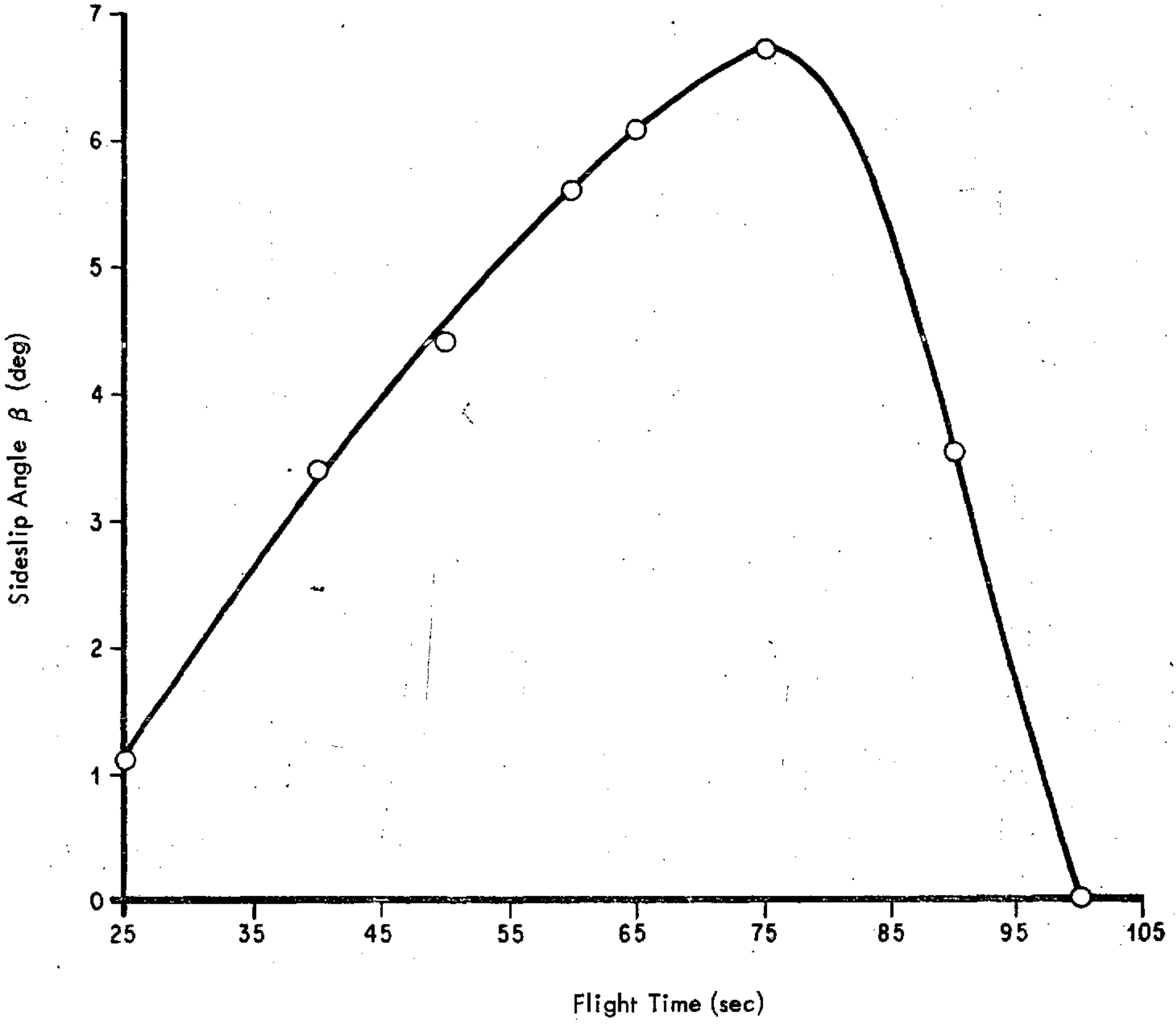
Table 4.2 Sideslip Angle for Different Flight Times

flight time (sec)	β (rad)	β (deg)
25	.02096	1.201
40	.05996	3.436
50	.07887	4.519
60	.09942	5.696
65	.10642	6.097
70	.11124	6.374
75	.11635	6.667
80	.11404	6.534
90	.06169	3.535
100	0	0
110	0	0
140	0	0

Table 4.3 Bias Torques Caused by SRM Misalignment

CASE	YAW BIAS (New. -m) x 10^6	ROLL BIAS (New. -m) x 10^6
1	3.02	0.
2	2.50	0.70
3	0.	0.87
4	-2.50	0.70
5	-3.07	0.
6	-2.50	-0.70
7	0.	-0.87
8	2.50	-0.70

Figure 4.5 Sideslip Angle vs Flight Time Due to Mean Wind Disturbance



The formulas developed in the previous section for computing the matrix elements in (4.25) are coded into the TRIMS program. The numerical data required by these formulas and tabulated in Appendix B is also stored internally in the program. Similarly the formulas (4.28) and (4.29) used to compute the performance criterion (4.27) are coded into the program together with the required numerical data. Only those input parameters with values that are likely to vary from run to run are entered as input data at execution time. These are the fourteen weighting factors w_{1t} and w_{2t} in the performance criterion and the values of the roll and yaw bias torques, z_2 and z_3, respectively. A more detailed description of the TRIMS program including flowcharts, listing, instructions showing how to use the program is given in Appendix C.

The trim angles for the eight different combinations of yaw and roll SRM bias torques in Table 4.3 were computed in a single run of the TRIMS program. Each case entailed computing the trim angles for the twelve trajectory points or flight times which totals to 96 trim solutions. The total cpu time was 5.29 seconds on the IBM 370/165 computer which averages to 0.055 second per trim solution. For this run the second order gradient method and the weighting factors in the performance criterion were chosen to be

$$w_{1\ell} = \begin{cases} 3000 \text{ for } & \ell \neq 6 \\ 4000 \text{ for } & \ell = 6 \end{cases}$$

$$w_{2\ell} = 0 \qquad\qquad \ell = 1, \ldots . 7$$

The lateral trim solutions for the eight cases in Table 4.3 are shown in Figure 4.6 where the trim angles are in degrees. The trim angles δ_4 and δ_5 for the SRM engines are always zero since in the current TRIMS program the SRM engines are not gimballed. However, the provision for gimballing the SRM engines has been included in the development of the development of the TRIMS program.

For most of the trajectory points in Figure 4.6, especially those with high dynamic pressure, some of the deflection angles exceed the allowable limits by an order of magnitude. This indicates that the Space Shuttle configuration does not have sufficient control authority to meet the trim conditions $Y = L = M = 0$ when the SRM engines are not gimballed.

A check of the TRIMS program against a lateral trim solution computed at MSFC was made. The MSFC solution is for the case of zero net rolling and yawing moments, but, unlike in the TRIMS program the requirement of a zero net side force (i.e., $Y = 0$) is <u>not</u> imposed. Also the MSFC solution does not consider the deflection of the aileron. A special modification of the TRIMS program for including or disregarding the trim condition $Y = 0$ and/or the aileron deflection was made and is described in Appendix C. Although the actual modification of the TRIMS program to eliminate the constraint $Y = 0$ is minor, it is based on a novel procedure derived in Appendix D. A comparison of the trim solutions computed by each program for (supposedly) the same trim problem showed that the deflection angles have about the same magnitude but are not equal. A more detailed discussion of the comparison including plots of the trim solutions is given in Appendix E.

Figure 4.6 Lateral Trim of Space Shuttle

```
          CASE  1
          ----

      SYSTEM DYNAMICS PARAMETERS
      --------------------------
                                          YAW BIAS TORQUE =  3020000.0
                                         ROLL BIAS TORQUE =        0.0


      PERFORMANCE CRITERION PARAMETERS
      --------------------------------
                                                 W11 =3000.00            W21 =   0.0
                                                 W12 =3000.00            W22 =   0.0
                                  WEIGHTING      W13 =3000.00            W23 =   0.0
                                  FACTORS        W14 =3000.00            W24 =   0.0
                                                 W15 =3000.00            W25 =   0.0
                                                 W16 =4000.00            W26 =   0.0
                                                 W17 =3000.00            W27 =   0.0


      TRIM DEFLECTION ANGLES
      ----------------------
                                 TRAJ.  FLIGHT                               DELTA
                                  PT.    TIME     (1)      (2)      (3)      (4)      (5)      (6)      (7)
                              ..........................................................................
                              .                                                                          .
                              .    1    25.0   -11.59     4.08   -19.35     0.0      0.0     -7.55    13.03  .
                              .    2    40.0    14.77     4.99    16.13     0.0      0.0      4.28   -13.35  .
                              .    3    50.0    46.99     3.32    61.74     0.0      0.0     31.07   -24.96  .
                              .    4    60.0   100.03    -4.25   146.84     0.0      0.0     81.55   -39.35  .
                              .    5    65.0   128.39   -13.30   229.11     0.0      0.0    121.16   -43.78  .
                              .    6    70.0    95.05    22.88   141.57     0.0      0.0     18.55   -68.60  .
                              .    7    75.0   129.30     8.95   239.63     0.0      0.0     20.78   -70.80  .
                              .    8    80.0   136.46    -2.99   314.14     0.0      0.0     35.18   -80.27  .
                              .    9    90.0    11.56    22.91   201.30     0.0      0.0     52.10   -67.49  .
                              .   10   100.0     5.74    -5.80   -37.39     0.0      0.0     -5.75    27.59  .
                              .   11   110.0    12.93    -9.93   -41.76     0.0      0.0      6.74    60.32  .
                              .   12   140.0   118.73   -57.42  -119.78     0.0      0.0     58.39   110.65  .
                              .                                                                          .
                              ..........................................................................
                                                  TOP      YAW    PITCH     YAW    PITCH  AILERON  RUDDER
                                                <----   ORBITER    ---->< ---- SRM   ---->




          CASE  2
          ----

      SYSTEM DYNAMICS PARAMETERS
      --------------------------
                                          YAW BIAS TORQUE =  2500000.0
                                         ROLL BIAS TORQUE =   700000.0


      PERFORMANCE CRITERION PARAMETERS
      --------------------------------
                                                 W11 =3000.00            W21 =   0.0
                                                 W12 =3000.00            W22 =   0.0
                                  WEIGHTING      W13 =3000.00            W23 =   0.0
                                  FACTORS        W14 =3000.00            W24 =   0.0
                                                 W15 =3000.00            W25 =   0.0
                                                 W16 =4000.00            W26 =   0.0
                                                 W17 =3000.00            W27 =   0.0


      TRIM DEFLECTION ANGLES
      ----------------------
                                 TRAJ.  FLIGHT                               DELTA
                                  PT.    TIME     (1)      (2)      (3)      (4)      (5)      (6)      (7)
                              ..........................................................................
                              .                                                                          .
                              .    1    25.0   -11.49     5.19   -19.19     0.0      0.0     -5.80     8.18  .
                              .    2    40.0    14.29     5.93    15.09     0.0      0.0      4.79   -15.18  .
                              .    3    50.0    47.25     4.11    61.77     0.0      0.0     32.15   -26.64  .
                              .    4    60.0   100.54    -3.79   147.57     0.0      0.0     83.79   -40.56  .
                              .    5    65.0   128.98   -13.09   230.79     0.0      0.0    123.92   -44.77  .
                              .    6    70.0    94.54    23.77   141.11     0.0      0.0     19.24   -69.80  .
                              .    7    75.0   128.96     9.73   240.02     0.0      0.0     21.24   -71.94  .
                              .    8    80.0   136.29    -2.36   315.70     0.0      0.0     35.80   -81.62  .
                              .    9    90.0     9.69    24.38   205.51     0.0      0.0     54.09   -70.14  .
                              .   10   100.0     1.68    -3.15   -34.26     0.0      0.0     -4.23    21.97  .
                              .   11   110.0     7.48    -6.47   -37.58     0.0      0.0      6.29    47.90  .
                              .   12   140.0    91.36   -44.12   -99.66     0.0      0.0     46.67    87.91  .
                              .                                                                          .
                              ..........................................................................
                                                  TOP      YAW    PITCH     YAW    PITCH  AILERON  RUDDER
                                                <----   ORBITER    ----><---- SRM   ---->
```

Figure 4.6 (Continued)

```
        CASE   3
        ----

SYSTEM DYNAMICS PARAMETERS
--------------------------
                                  YAW BIAS TORQUE =          0.0
                                 ROLL BIAS TORQUE =    870000.0

PERFORMANCE CRITERION PARAMETERS
--------------------------------
                                         W11 =3000.00                W21 =   0.0
                                         W12 =3000.00                W22 =   0.0
                          WEIGHTING      W13 =3000.00                W23 =   0.0
                          FACTORS        W14 =3000.00                W24 =   0.0
                                         W15 =3000.00                W25 =   0.0
                                         W16 =4000.00                W26 =   0.0
                                         W17 =3000.00                W27 =   0.0


TRIM DEFLECTION ANGLES
----------------------
                        TRAJ.  FLIGHT                               DELTA
                         PT.    TIME     (1)       (2)      (3)      (4)      (5)      (6)     (7)
                     ...................................................................................
                     .                                                                                 .
                     .    1     25.0     2.46      2.59     3.16     0.0      0.0      4.32  -10.51    .
                     .    2     40.0    25.63      2.99    31.20     0.0      0.0      6.88  -22.68    .
                     .    3     50.0    60.27      0.63    81.35     0.0      0.0     39.90  -32.28    .
                     .    4     60.0   114.42     -8.03   169.95     0.0      0.0     93.27  -45.32    .
                     .    5     65.0   142.74    -17.98   256.87     0.0      0.0    134.91  -48.87    .
                     .    6     70.0   103.77     21.98   156.45     0.0      0.0     20.14  -75.25    .
                     .    7     75.0   139.00      7.19   259.74     0.0      0.0     22.50  -76.72    .
                     .    8     80.0   146.77     -5.50   340.59     0.0      0.0     38.18  -87.15    .
                     .    9     90.0    12.67     24.91   240.96     0.0      0.0     62.35  -80.79    .
                     .   10    100.0    -3.82      2.05    -4.11     0.0      0.0      0.66   -1.08    .
                     .   11    110.0    -4.01      2.18    -3.75     0.0      0.0      0.88   -2.53    .
                     .   12    140.0    -8.61      4.25    -0.63     0.0      0.0     -2.07   -4.59    .
                     .                                                                                 .
                     ...................................................................................
                                         TOP       YAW     PITCH     YAW     PITCH  AILERON  RUDDER
                                        <----    ORBITER     ----><---- SRM  ---->



        CASE   4
        ----

SYSTEM DYNAMICS PARAMETERS
--------------------------
                                  YAW BIAS TORQUE = -2500000.0
                                 ROLL BIAS TORQUE =    700000.0

PERFORMANCE CRITERION PARAMETERS
--------------------------------
                                         W11 =3000.00                W21 =   0.0
                                         W12 =3000.00                W22 =   0.0
                          WEIGHTING      W13 =3000.00                W23 =   0.0
                          FACTORS        W14 =3000.00                W24 =   0.0
                                         W15 =3000.00                W25 =   0.0
                                         W16 =4000.00                W26 =   0.0
                                         W17 =3000.00                W27 =   0.0


TRIM DEFLECTION ANGLES
----------------------
                        TRAJ.  FLIGHT                               DELTA
                         PT.    TIME     (1)       (2)      (3)      (4)      (5)      (6)     (7)
                     ...................................................................................
                     .                                                                                 .
                     .    1     25.0    17.83     -0.86    27.80     0.0      0.0     14.62  -28.71    .
                     .    2     40.0    38.42     -0.75    49.56     0.0      0.0      8.92  -30.03    .
                     .    3     50.0    74.53     -3.62   102.99     0.0      0.0     47.92  -37.67    .
                     .    4     60.0   129.51    -12.95   194.34     0.0      0.0    102.62  -49.97    .
                     .    5     65.0   157.66    -23.50   284.87     0.0      0.0    145.67  -52.90    .
                     .    6     70.0   114.23     19.56   173.65     0.0      0.0     20.77  -80.67    .
                     .    7     75.0   150.29      3.98   281.35     0.0      0.0     23.65  -81.44    .
                     .    8     80.0   158.46     -9.28   367.35     0.0      0.0     40.49  -92.57    .
                     .    9     90.0    16.93     24.76   278.03     0.0      0.0     70.48  -91.22    .
                     .   10    100.0    -7.82      6.45    27.65     0.0      0.0      5.29  -23.70    .
                     .   11    110.0   -13.93      9.97    31.55     0.0      0.0     -4.88  -51.97    .
                     .   12    140.0  -105.21     50.95    98.65     0.0      0.0    -50.00  -95.29    .
                     .                                                                                 .
                     ...................................................................................
                                         TOP       YAW     PITCH     YAW     PITCH  AILERON  RUDDER
                                        <----    ORBITER     ----><---- SRM  ---->
```

Figure 4.6 (Continued)

```
     CASE   5
     ----

SYSTEM DYNAMICS PARAMETERS
--------------------------              YAW BIAS TORQUE = -3070000.0
                                       ROLL BIAS TORQUE =        0.0


PERFORMANCE CRITERION PARAMETERS
--------------------------------                W11 =3000.00          W21 =   0.0
                                                W12 =3000.00          W22 =   0.0
                                  WEIGHTING     W13 =3000.00          W23 =   0.0
                                  FACTORS       W14 =3000.00          W24 =   0.0
                                                W15 =3000.00          W25 =   0.0
                                                W16 =4000.00          W26 =   0.0
                                                W17 =3000.00          W27 =   0.0


TRIM DEFLECTION ANGLES
----------------------           TRAJ.  FLIGHT                            DELTA
                                  PT.    TIME     (1)      (2)      (3)     (4)      (5)      (6)      (7)
                             .........................................................................................
                             .                                                                                       .
                             .     1     25.0    24.12    -3.29    37.88    0.0      0.0     17.32   -31.90          .
                             .     2     40.0    44.17    -3.14    58.11    0.0      0.0      9.31   -31.42          .
                             .     3     50.0    80.21    -6.10   111.94    0.0      0.0     50.27   -38.40          .
                             .     4     60.0   135.32   -15.40   203.81    0.0      0.0    104.48   -50.81          .
                             .     5     65.0   163.33   -25.98   294.99    0.0      0.0    147.65   -53.68          .
                             .     6     70.0   119.03    17.75   181.20    0.0      0.0     20.41   -81.85          .
                             .     7     75.0   155.27     1.94   289.98    0.0      0.0     23.71   -82.37          .
                             .     8     80.0   163.46   -11.42   377.05    0.0      0.0     40.89   -93.61          .
                             .     9     90.0    20.37    23.38   289.63    0.0      0.0     72.07   -93.16          .
                             .    10    100.0    -5.84     5.90    38.01    0.0      0.0      5.85   -28.04          .
                             .    11    110.0   -13.14    10.09    42.45    0.0      0.0     -6.85   -61.32          .
                             .    12    140.0  -120.70    58.37   121.76    0.0      0.0    -59.36  -112.48          .
                             .                                                                                       .
                             .........................................................................................
                                                   TOP      YAW     PITCH     YAW     PITCH  AILERON  RUDDER
                                                  <----   ORBITER    ----><----- SRM  ---->
```

```
     CASE   6
     ----

SYSTEM DYNAMICS PARAMETERS
--------------------------              YAW BIAS TORQUE = -2500000.0
                                       ROLL BIAS TORQUE =  -700000.0

PERFORMANCE CRITERION PARAMETERS
--------------------------------                W11 =3000.00          W21 =   0.0
                                                W12 =3000.00          W22 =   0.0
                                  WEIGHTING     W13 =3000.00          W23 =   0.0
                                  FACTORS       W14 =3000.00          W24 =   0.0
                                                W15 =3000.00          W25 =   0.0
                                                W16 =4000.00          W26 =   0.0
                                                W17 =3000.00          W27 =   0.0


TRIM DEFLECTION ANGLES
----------------------           TRAJ.  FLIGHT                            DELTA
                                  PT.    TIME     (1)      (2)      (3)     (4)      (5)      (6)      (7)
                             .........................................................................................
                             .                                                                                       .
                             .     1     25.0    23.73    -4.34    37.25    0.0      0.0     15.35   -26.68          .
                             .     2     40.0    44.42    -4.01    58.81    0.0      0.0      8.75   -29.43          .
                             .     3     50.0    79.67    -6.82   111.49    0.0      0.0     49.02   -36.61          .
                             .     4     60.0   134.52   -15.77   202.62    0.0      0.0    102.05   -49.50          .
                             .     5     65.0   162.46   -26.09   292.77    0.0      0.0    144.68   -52.61          .
                             .     6     70.0   119.33    16.90   181.34    0.0      0.0     19.70   -80.54          .
                             .     7     75.0   155.40     1.22   289.18    0.0      0.0     23.22   -81.13          .
                             .     8     80.0   163.42   -11.97   374.97    0.0      0.0     40.22   -92.14          .
                             .     9     90.0    22.17    21.91   284.70    0.0      0.0     69.92   -90.29          .
                             .    10    100.0    -1.66     3.15    34.26    0.0      0.0      4.23   -21.97          .
                             .    11    110.0    -7.48     6.47    37.58    0.0      0.0     -6.29   -47.40          .
                             .    12    140.0   -91.36    44.12    99.66    0.0      0.0    -46.67   -87.91          .
                             .                                                                                       .
                             .........................................................................................
                                                   TOP      YAW     PITCH     YAW     PITCH  AILERON  RUDDER
                                                  <----   ORBITER    ----><----- SRM  ---->
```

Figure 4.6 (Continued)

```
CASE  7
----

SYSTEM DYNAMICS PARAMETERS
--------------------------
                                    YAW BIAS TORQUE =          0.0
                                   ROLL BIAS TORQUE =   -870000.0

PERFORMANCE CRITERION PARAMETERS
--------------------------------
                                          W11 =3000.00                 W21 =    0.0
                                          W12 =3000.00                 W22 =    0.0
                              WEIGHTING   W13 =3000.00                 W23 =    0.0
                              FACTORS     W14 =3000.00                 W24 =    0.0
                                          W15 =3000.00                 W25 =    0.0
                                          W16 =4000.00                 W26 =    0.0
                                          W17 =3000.00                 W27 =    0.0

TRIM DEFLECTION ANGLES
----------------------
                              TRAJ.  FLIGHT                             DELTA
                               PT.    TIME     (1)       (2)       (3)       (4)       (5)       (6)       (7)
                             ********************************************************************************
                             *                                                                              *
                             *    1   25.0      9.78     -1.74     14.90      0.0       0.0       5.23     -7.99    *
                             *    2   40.0     33.08     -1.06     42.70      0.0       0.0       6.67    -21.94    *
                             *    3   50.0     66.66     -3.34     91.91      0.0       0.0      41.27    -30.97    *
                             *    4   60.0    120.64    -11.54    180.24      0.0       0.0      92.57    -44.74    *
                             *    5   65.0    148.70    -21.20    266.69      0.0       0.0     133.68    -48.51    *
                             *    6   70.0    110.11     18.68    166.00      0.0       0.0      18.80    -75.08    *
                             *    7   75.0    145.36      3.76    269.46      0.0       0.0      21.97    -76.35    *
                             *    8   80.0    152.93     -8.84    350.07      0.0       0.0      37.84    -86.62    *
                             *    9   90.0     19.19     21.37    249.25      0.0       0.0      61.65    -79.64    *
                             *   10  100.0      3.82     -2.05      4.11      0.0       0.0      -0.66      1.08    *
                             *   11  110.0      4.01     -2.18      3.75      0.0       0.0      -0.88      2.53    *
                             *   12  140.0      8.61     -4.25      0.63      0.0       0.0       2.07      4.59    *
                             *                                                                              *
                             ********************************************************************************
                                               TOP       YAW     PITCH     YAW     PITCH   AILERON   RUDDER
                                             <----    ORBITER     ----><---- SRM   ---->


CASE  8
----

SYSTEM DYNAMICS PARAMETERS
--------------------------
                                    YAW BIAS TORQUE =  2500000.0
                                   ROLL BIAS TORQUE =   -700000.0

PERFORMANCE CRITERION PARAMETERS
--------------------------------
                                          W11 =3000.00                 W21 =    0.0
                                          W12 =3000.00                 W22 =    0.0
                              WEIGHTING   W13 =3000.00                 W23 =    0.0
                              FACTORS     W14 =3000.00                 W24 =    0.0
                                          W15 =3000.00                 W25 =    0.0
                                          W16 =4000.00                 W26 =    0.0
                                          W17 =3000.00                 W27 =    0.0

TRIM DEFLECTION ANGLES
----------------------
                              TRAJ.  FLIGHT                             DELTA
                               PT.    TIME     (1)       (2)       (3)       (4)       (5)       (6)       (7)
                             ********************************************************************************
                             *                                                                              *
                             *    1   25.0     -5.59      1.71     -9.74      0.0       0.0      -5.06     10.21    *
                             *    2   40.0     20.28      2.67     24.34      0.0       0.0       4.62    -14.59    *
                             *    3   50.0     52.40      0.91     70.27      0.0       0.0      33.26    -25.58    *
                             *    4   60.0    105.55     -6.62    155.85      0.0       0.0      83.23    -40.10    *
                             *    5   65.0    133.78    -15.68    238.69      0.0       0.0     122.93    -44.49    *
                             *    6   70.0     99.64     21.11    148.80      0.0       0.0      18.17    -69.66    *
                             *    7   75.0    134.07      6.97    247.85      0.0       0.0      20.82    -71.64    *
                             *    8   80.0    141.24     -5.05    323.32      0.0       0.0      35.53    -81.20    *
                             *    9   90.0     14.93     21.52    212.18      0.0       0.0      53.52    -69.22    *
                             *   10  100.0      7.82     -6.45    -27.65      0.0       0.0      -5.29     23.70    *
                             *   11  110.0     13.93     -9.97    -31.55      0.0       0.0       4.88     51.97    *
                             *   12  140.0    105.21    -50.95    -98.65      0.0       0.0      50.00     95.29    *
                             *                                                                              *
                             ********************************************************************************
                                               TOP       YAW     PITCH     YAW     PITCH   AILERON   RUDDER
                                             <----    ORBITER     ----><---- SRM   ---->
```

4.3 OPTIMUM FEEDBACK CONTROL AND PERFORMANCE

The vector-matrix equations defining the linear stochastic optimum control problem and its solution are given in Section 3.4. These equations entail computation of the matrices F , M , K , P from the matrices A, B , C , E , G , H , V , W , Q , R defining the optimum control problem. In order to simplify the feedback design, the matrices F , M , K , P are partitioned as follows:

$$F = \begin{array}{cc} \left[\begin{array}{cc} F_x & F_z \end{array}\right] \\ \begin{array}{cc} \xleftrightarrow{\;6\;} & \xleftrightarrow{\;3\;} \end{array} \end{array} \qquad M = \begin{array}{cc} \begin{bmatrix} M_1 & M_2 \\ M_2' & M_3 \end{bmatrix} & \begin{array}{c} \updownarrow 6 \\ \updownarrow 3 \end{array} \\ \begin{array}{cc} \xleftrightarrow{\;6\;} & \xleftrightarrow{\;3\;} \end{array} & \end{array}$$

$$K = \begin{bmatrix} K_x \\ K_z \end{bmatrix} \begin{array}{c} \updownarrow 6 \\ \updownarrow 3 \end{array} \qquad P = \begin{array}{cc} \begin{array}{cc} \xleftrightarrow{\;\;\;} & \xleftrightarrow{\;\;\;} \end{array} & \\ \begin{bmatrix} P_1 & P_2 \\ P_2' & P_3 \end{bmatrix} & \begin{array}{c} \updownarrow 6 \\ \updownarrow 3 \end{array} \end{array}$$

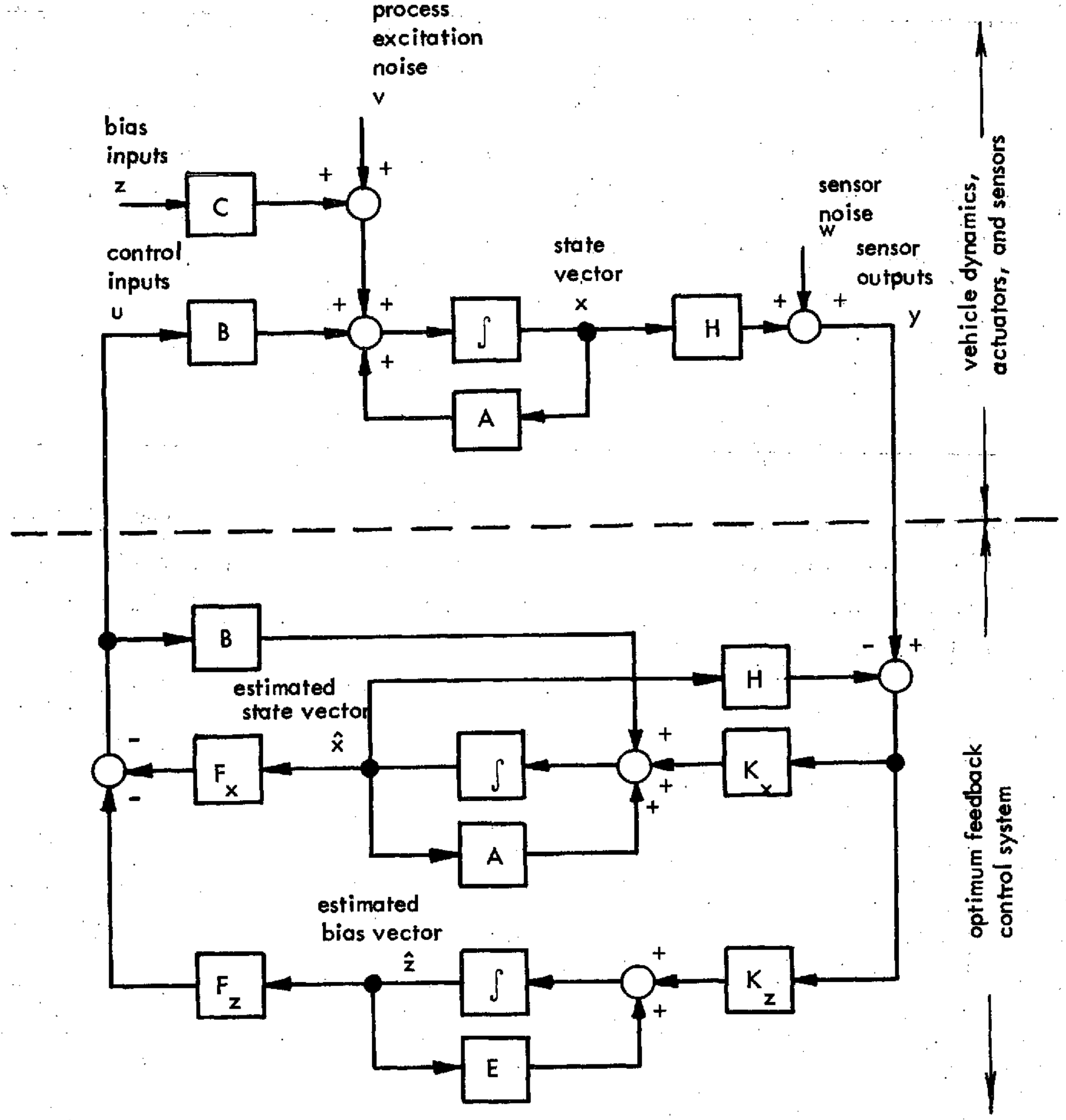
A block diagram of the complete closed loop system in terms of the matrices listed above is given in Figure 4.7. The lower half of the block diagram depicts the optimum feedback control system.

For the lateral control of the Space Shuttle the state vector x , control vector u , bias vector z , and observation vector y are defined to be

$$x = \begin{bmatrix} y \\ \varphi \\ \psi \\ v \\ p \\ r \end{bmatrix} \begin{array}{l} \text{side displacement} \\ \text{roll angle} \\ \text{yaw angle} \\ \text{side velocity} \\ \text{roll rate} \\ \text{yaw rate} \end{array} \qquad u = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \text{top} \\ \text{yaw} \\ \text{pitch} \end{array}\right\} \text{orbiter} \\ \left.\begin{array}{l} \text{yaw} \\ \text{pitch} \end{array}\right\} \text{SRM} \\ \text{aileron} \\ \text{rudder} \end{array}$$

$$z = \begin{bmatrix} \beta \\ T_{y_b} \\ T_{r_b} \end{bmatrix} \begin{array}{l} \text{side slip} \\ \text{yaw bias torque} \\ \text{roll bias torque} \end{array} \qquad y = \begin{bmatrix} y \\ \varphi \\ \psi \\ p \\ r \end{bmatrix}$$

Figure 4.7 Block Diagram of Closed Loop System with Optimum Feedback Control



The vehicle dynamics used in the optimum control design are for the first trajectory point (flight time = 25 sec). The elements of the matrices A , B , C were computed from the data in Appendix B using the equations in Section 4.1. It should be noted that the results in this section are for the case of no dorsal fins and the results in Section 4.2 include the effect of the dorsal fins. For the first trajectory point the difference in the two cases is minor. Since y is assumed to be a subvector of x , the elements of the observation matrix H are either 0 or 1 where a value of 1 in column i indicates that x_i is one of the measured quantities. The relative weighting matrices Q and R in the performance criterion (3.82) are selected by the control designer with the goal of optimizing the closed loop performance. The particular approach adopted for this problem is to select Q and R of the form

$$Q = \text{Diag} \{ y_{max}^{-2} , \varphi_{max}^{-2} , \psi_{max}^{-2} , v_{max}^{-2} , p_{max}^{-2} , r_{max}^{-2} \}$$

$$R = \text{Diag} \{ u_{1\,max}^{-2} , \ . \ . \ . \ . \ , u_{7\,max}^{-2} \}$$

The parameter σ in the performance criterion (3.82) is varied until an acceptable "trade off" is achieved between the closed loop performance and the level of control effort. For this example $\sigma = 1$ and the maximum values of the state variables and control deflections used in the performance criterion are listed below.

$y_{max} = 10\,m$

$\varphi_{max} = 0.01\,rad$

$\psi_{max} = 0.01\,rad$

$v_{max} = 5\,m/sec$

$p_{max} = 0.1\,rad/sec$

$r_{max} = 0.1\,rad/sec$

$u_{j\,max} = 0.2\,rad$

$j = 1 , \ . \ . \ . \ . \ , 7$

The matrix V defining the state excitation noise spectral density is assumed to be diagonal with

$$V_{\ell\ell} = 0 \qquad \ell = 1, 2, 3$$

$$V_{44} = (2v_{max})^2$$

$$V_{55} = (0.1p_{max})^2$$

$$V_{66} = (0.1r_{max})^2$$

For this example the matrices G and E defining the noise model associated with the bias inputs are

$$G = \mathrm{Diag}\,[0.1\ ,\ 0.1 \times 10^{14}\ ,\ 0.1 \times 10^{14}]$$

$$E = \mathrm{Diag}\,[-0.01\ ,\ -0.01\ ,\ 0.01]$$

The ℓth diagonal element of G is roughly equal to the maximum value of z_ℓ squared. The negative diagonal elements in E provide a small amount of damping in the noise model which is required in order that the covariance matrix P_3 corresponding to the bias vector z does not become infinite. The observation noise spectral density matrix is assumed to have the form

$$W = \mathrm{Diag}\,[\sigma_y^2\ ,\ \sigma_\varphi^2\ ,\ \sigma_\psi^2\ ,\ \sigma_p^2\ ,\ \sigma_r^2]$$

The standard deviation σ_y defines the level of noise associated with the measurement of y and the other standard deviations are similarly defined. By varying the standard deviations of the sensor noise as part of the design procedure,different Kalman filter designs are obtained. For the final Kalman filter design in this example

$$\sigma_y = 0.1\,y_{max}$$

$$\sigma_\varphi = 0.1\,\varphi_{max}$$

$$\sigma_\psi = 0.1\,\psi_{max}$$

$$\sigma_p = 0.01\,p_{max}$$

$$\sigma_r = 0.01\,r_{max}$$

The numerical values of these matrices defining the optimum control problem are given in Figure 4.8(a).

A computer program entitled Linear Systems Design (LSD) was used to design the optimum feedback system. The LSD program solves the equations for the optimum control solution given in Section 3.4.1. The resulting matrices M_1 , M_2 , F_x , F_z used to design the deterministic quadratic optimum control are shown in Figure 4.8(b). Similarly, the resulting matrices P_1 , P_2 , P_3 , K_x , K_z used to design the Kalman filter are shown in Figure 4.8(c).

The performance achieved by the feedback control system was simulated for the different designs. The control deflections as a function of time were plotted and are shown Figures 4.9 and 4.10. Note the SRM deflections δ_4 and δ_5 are not plotted since in the current investigation it is assumed that the SRM are not gimballed. The dynamic response in Figures 4.9 and 4.10 is for the case where the vehicle starts from the trim condition for

$$\beta = 1.20°$$

$$T_{r_b} = 3.02 \times 10^6 \text{ N-m}$$

$$T_{y_b} = 0. \text{ N-m}$$

to which correspond the deflection angles

$$\delta_1 = -6.63°$$

$$\delta_2 = 7.35°$$

$$\delta_3 = -11.27°$$

$$\delta_6 = 2.65°$$

$$\delta_7 = -12.11°$$

The initial trim solution shown above is indicated by the straight lines in Figures 4.9 and 4.10. The effect of a 2° step change in the sideslip angle causing an increase from $\beta = 1.20°$ to $\beta = 3.20°$ was simulated. The transient response curves show the performance of the control system in achieving the new trim solution. The curves in Figure 4.9 (a) are for the case of complete state feedback which assumes that the state of the process can be estimated perfectly (i.e., $H = \mathcal{I}$, $V = W = 0$, $\hat{x} = x$). This is not realistic but provides an upper bound on the performance as the estimation capability of the Kalman filter improves.

Observe in this case that the control deflections change discontinuaously due to a step change in β . This does not occur when the Kalman filter is included. The curves for the remaining cases show the performance when different Kalman filter designs are used. The different designs correspond to different values of the W matrix as shown below

$$W = \text{Diag}\,[\sigma_y^2, \sigma_\varphi^2, \sigma_\psi^2, \sigma_p^2, \sigma_r^2]$$

Figure	σ_y	σ_φ	σ_ψ	σ_p	σ_r
4.10b	10	.01	.01	.1	.1
4.10c	10	.01	.01	.01	.01
4.11	1	.001	.001	.001	.001

The value of W in Figure 4.8 (a) and the matrices in Figure 4.8 (c) correspond to the Kalman filter design used in Figure 4.9 .

In Section 3.4.1 a convergence property relating the optimum control approach and the trim control approach is given by (3.117). A demonstration of this property for the lateral control problem of the Space Shuttle is given below. Trajectory point number 1 occurring at 25 seconds after launch is shown in which the roll and yaw bias torques due to misalignment of the solid rocket motors are assumed to be

roll bias torque = 3.02×10^6 (N-m)

yaw bias torque = 0

The control vectors $u_x(\infty)$, $u_z(\infty)$, and $u(\infty)$ obtained by the optimum control approach have been computed in this case for three different values of σ (1., 0.1, 0.01) and are listed in Table 4.4. The computations were performed according to (3.94), (3.95), (3.98)-(3.102) where the control weighting matrix R was chosen to be and where I denotes the identity matrix.

$$R = 25I$$

The trim solution or limiting solution for $\sigma = 0$ was computed using the TRIMS program and is also listed in Table 4.4. An examination of Table 4.4 illustrates that the steady state control level $u(\infty)$ for the optimum control solution approaches the trim solution as σ approaches zero.

Figure 4.8 Numerical Value of Matrices Used in Optimum Control Design

(a) Definition of Linear Stochastic Optimum Control Problem

$$A = \begin{bmatrix} 0. & 0. & 0. & .10000E+01 & 0. & 0. \\ 0. & .16920E+01 & 0. & 0. & .10000E+01 & -.83330E+02 \\ 0. & .16920E+01 & 0. & 0. & 0. & -.83330E+02 \\ 0. & -.11760E+00 & 0. & -.12300E-01 & 0. & -.95400E+02 \\ 0. & 0. & 0. & -.13040E-02 & -.67560E-02 & .15330E-01 \\ 0. & 0. & 0. & .31340E-03 & .14470E-01 & -.11980E-02 \end{bmatrix}$$

$$B = \begin{bmatrix} 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ .71984E+00 & .14779E+01 & .19214E-01 & 0. & 0. & 0. & .35406E+00 \\ .12778E+00 & .17242E+00 & .47832E-01 & 0. & 0. & -.19571E-01 & .12336E+00 \\ -.61986E-01 & -.12772E+00 & -.94834E-04 & 0. & 0. & .33613E-02 & -.39708E-01 \end{bmatrix}$$

$$C = \begin{bmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ -.11739E+01 & 0. & 0. \\ -.12444E+00 & .10493E-07 & 0. \\ .29896E-01 & 0. & .16921E-08 \end{bmatrix} \qquad E = \begin{bmatrix} -.10000E-01 & 0. & 0. \\ 0. & -.10000E-01 & 0. \\ 0. & 0. & -.10000E-01 \end{bmatrix}$$

$$H = \begin{bmatrix} .10000E+01 & 0. & 0. & 0. & 0. & 0. \\ 0. & .10000E+01 & 0. & 0. & 0. & 0. \\ 0. & 0. & .10000E+01 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & .10000E+01 & 0. \\ 0. & 0. & 0. & 0. & 0. & .10000E+01 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.100000E-01 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.100000E\ 05 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.100000E\ 05 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.400000E-01 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.100000E\ 03 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.100000E\ 03 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.250000E\ 02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.250000E\ 02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.250000E\ 02 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.250000E\ 02 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.250000E\ 02 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.250000E\ 02 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.250000E\ 02 \end{bmatrix}$$

$$V = \begin{bmatrix} 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & .10000E+03 & 0. & 0. \\ 0. & 0. & 0. & 0. & .10000E-03 & 0. \\ 0. & 0. & 0. & 0. & 0. & .10000E-03 \end{bmatrix}$$

$$G = \begin{bmatrix} .10000E+00 & 0. & 0. \\ 0. & .10000E+14 & 0. \\ 0. & 0. & .10000E+14 \end{bmatrix}$$

$$W = \begin{bmatrix} .10000E+01 & 0. & 0. & 0. & 0. \\ 0. & .10000E-05 & 0. & 0. & 0. \\ 0. & 0. & .10000E-05 & 0. & 0. \\ 0. & 0. & 0. & .10000E-05 & 0. \\ 0. & 0. & 0. & 0. & .10000E-05 \end{bmatrix}$$

Figure 4.8 Numerical Value of Matrices Used in Optimum Control Design, (Continued)

(b) Deterministic Quadratic Optimum Control Design (Complete State Feedback)

$$M_1 = \begin{bmatrix} 0.278338E\ 00 & -0.563500E\ 01 & 0.762735E\ 00 & 0.378119E\ 01 & -0.158639E\ 00 & 0.445554E\ 02 \\ -0.563500E\ 01 & 0.681854E\ 04 & -0.585306E\ 04 & -0.138048E\ 03 & 0.388273E\ 04 & 0.157787E\ 04 \\ 0.762735E\ 00 & -0.585306E\ 04 & 0.676011E\ 04 & 0.211353E\ 02 & -0.392709E\ 04 & -0.848510E\ 04 \\ 0.378119E\ 01 & -0.138048E\ 03 & 0.211353E\ 02 & 0.908992E\ 02 & -0.800357E\ 01 & 0.106545E\ 04 \\ -0.158639E\ 00 & 0.388273E\ 04 & -0.392709E\ 04 & -0.800357E\ 01 & 0.502995E\ 04 & 0.783438E\ 04 \\ 0.445554E\ 02 & 0.157787E\ 04 & -0.848510E\ 04 & 0.106545E\ 04 & 0.783438E\ 04 & 0.574432E\ 05 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -0.127343E\ 02 & 0.136957E-06 & 0.305701E-06 \\ 0.193046E\ 04 & -0.175135E-04 & -0.462173E-04 \\ -0.332043E\ 03 & 0.344119E-05 & 0.802122E-05 \\ -0.124515E\ 04 & 0.104665E-04 & 0.297547E-04 \\ -0.141713E\ 03 & 0.323558E-04 & 0.621067E-05 \\ -0.149686E\ 05 & 0.182085E-03 & 0.365003E-03 \end{bmatrix}$$

$$F_x = \begin{bmatrix} -0.242246E-02 & 0.119613E\ 02 & 0.157240E\ 01 & -0.656341E-01 & 0.605653E\ 01 & -0.717104E\ 02 \\ -0.513893E-02 & 0.105541E\ 02 & 0.175102E\ 02 & -0.123552E\ 00 & -0.580465E\ 01 & -0.176404E\ 03 \\ 0.243295E-02 & 0.731637E\ 01 & -0.746488E\ 01 & 0.504930E-01 & 0.958743E\ 01 & 0.155895E\ 02 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.611420E-02 & -0.282727E\ 01 & 0.193339E\ 01 & 0.149504E\ 00 & -0.288419E\ 01 & 0.158991E\ 01 \\ -0.179980E-01 & 0.147035E\ 02 & -0.560701E\ 01 & -0.444370E\ 00 & 0.122703E\ 02 & -0.374812E\ 02 \end{bmatrix}$$

$$F_z = \begin{bmatrix} 0.541855E\ 00 & 0.296517E-07 & -0.166160E-07 \\ 0.187014E\ 01 & -0.586262E-07 & -0.625121E-07 \\ -0.117112E\ 01 & 0.696390E-07 & 0.333618E-07 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ -0.190147E\ 01 & -0.848574E-09 & 0.442091E-07 \\ 0.544064E\ 01 & 0.258135E-07 & -0.127670E-06 \end{bmatrix}$$

(c) Kalman Filter Design

$$P_1 = \begin{bmatrix} .44244E+01 & -.78062E-04 & -.77003E-04 & .98272E+01 & -.25592E-03 & .45855E-04 \\ -.78062E-04 & .24450E-04 & .23735E-04 & -.42155E-03 & .93807E-06 & -.67372E-05 \\ -.77003E-04 & .23735E-04 & .24434E-04 & -.41794E-03 & -.53273E-07 & -.67546E-05 \\ .98272E+01 & -.42155E-03 & -.41794E-03 & .43977E+02 & -.17384E-02 & .27358E-03 \\ -.25592E-03 & .93807E-06 & -.53273E-07 & -.17384E-02 & .14250E-04 & -.17866E-06 \\ .45855E-04 & -.67372E-05 & -.67546E-05 & .27358E-03 & -.17866E-06 & .40175E-05 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} -.12735E+00 & -.58636E+06 & .11444E+07 \\ -.11593E-03 & -.88900E+03 & -.13900E+04 \\ -.11413E-03 & -.92330E+03 & -.14021E+04 \\ -.49321E+00 & -.30518E+07 & .53063E+07 \\ -.19722E-03 & .24128E+04 & .92674E+02 \\ .69539E-04 & .55174E+03 & .84891E+03 \end{bmatrix} \qquad P_3 = \begin{bmatrix} .67931E+00 & .77186E+07 & -.10809E+08 \\ .77186E+07 & .94366E+14 & -.12756E+15 \\ -.10809E+08 & -.12756E+15 & .20316E+15 \end{bmatrix}$$

$$K_x = \begin{bmatrix} .44244E+01 & -.78062E+02 & -.77003E+02 & -.25592E+03 & .45855E+02 \\ -.78062E-04 & .24450E+02 & .23735E+02 & .93807E+00 & -.67372E+01 \\ -.77003E-04 & .23735E+02 & .24434E+02 & -.53273E-01 & -.67546E+01 \\ .98272E+01 & -.42155E+03 & -.41794E+03 & -.17384E+04 & .27358E+03 \\ -.25592E-03 & .93807E+00 & -.53273E-01 & .14250E+02 & -.17866E+00 \\ .45855E-04 & -.67372E+01 & -.67546E+01 & -.17866E+00 & .40175E+01 \end{bmatrix}$$

$$K_z = \begin{bmatrix} -.12735E+00 & -.11593E+03 & -.11413E+03 & -.19722E+03 & .69539E+02 \\ -.58636E+06 & -.88900E+09 & -.92330E+09 & .24128E+10 & .55174E+09 \\ .11444E+07 & -.13900E+10 & -.14021E+10 & .92674E+08 & .84891E+09 \end{bmatrix}$$

Figure 4.9 Dynamic Response for a 2° Step in Sideslip Angle

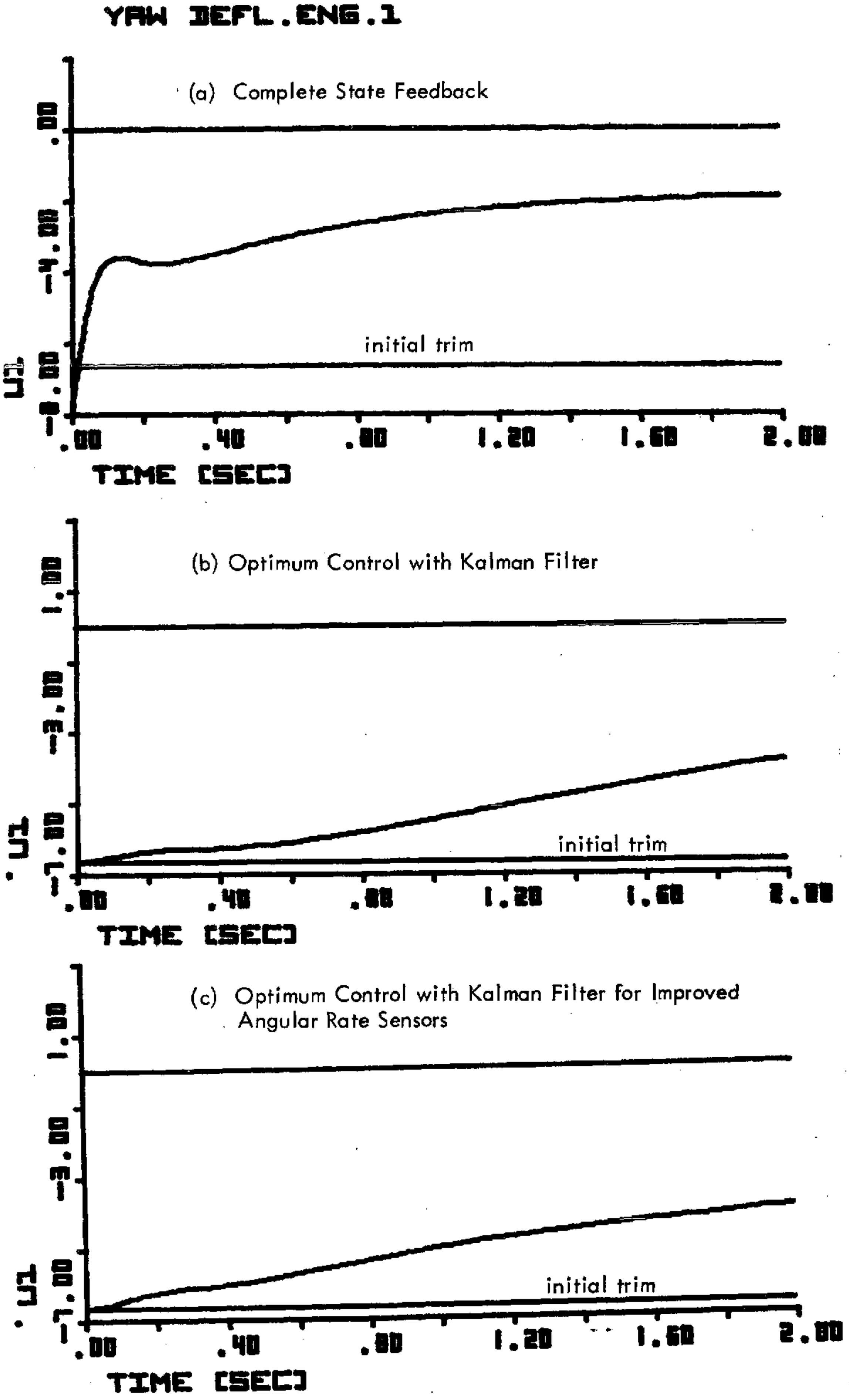


Figure 4.9, continued - 2 -

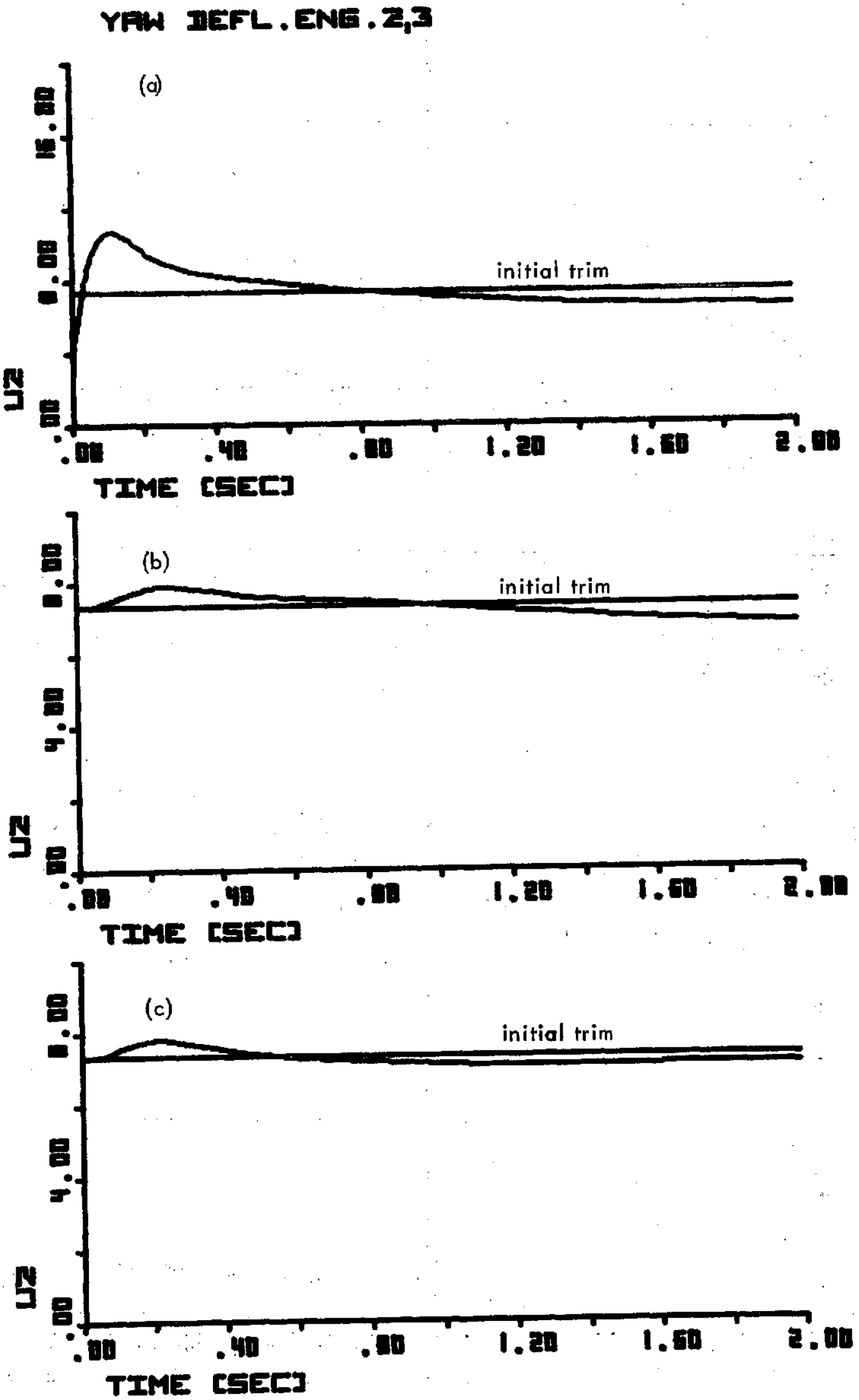


Figure 4.9, continued - 3 -

PITCH DEFL.ENG.2,3

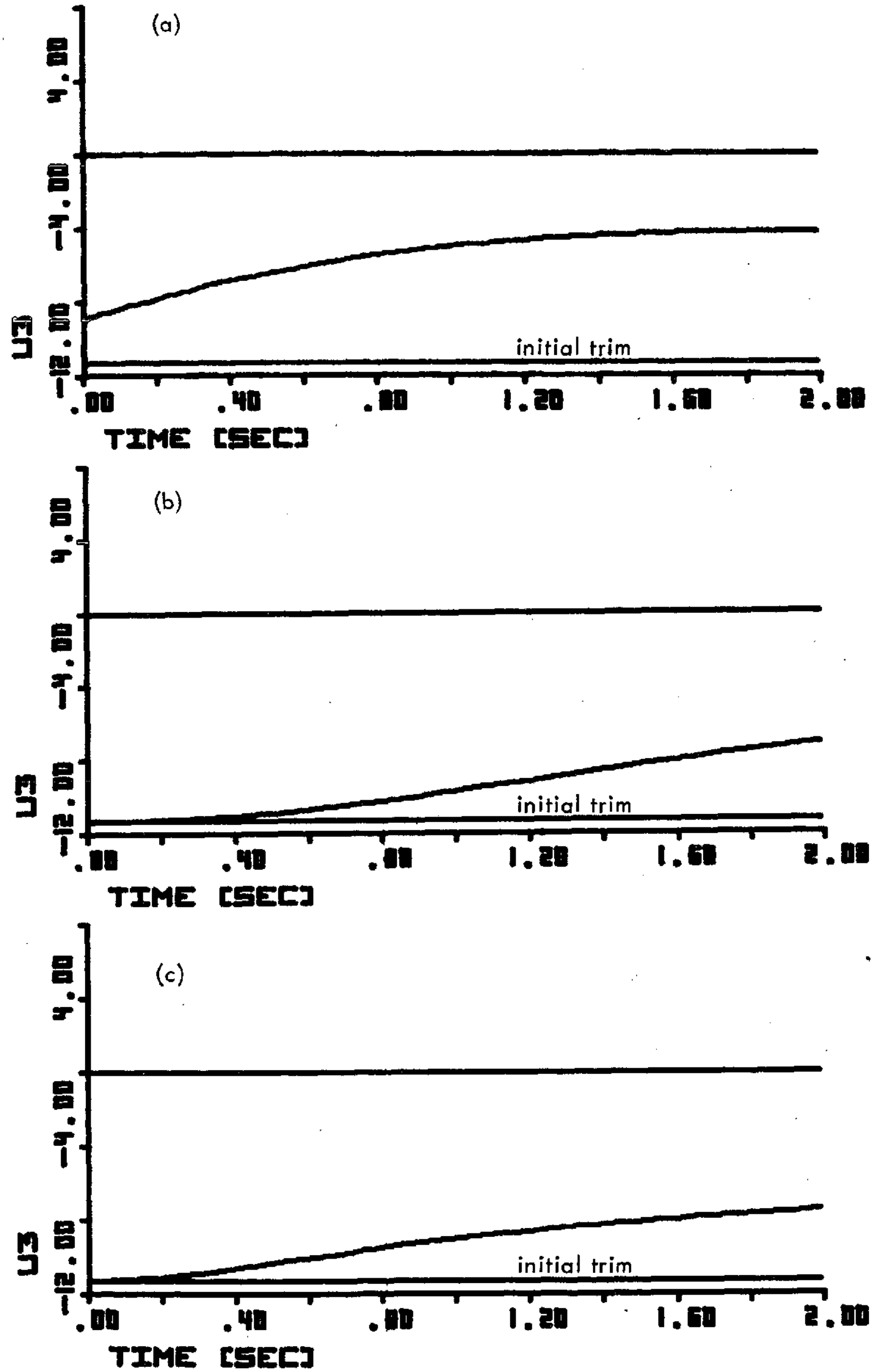


Figure 4.9, continued - 4 -

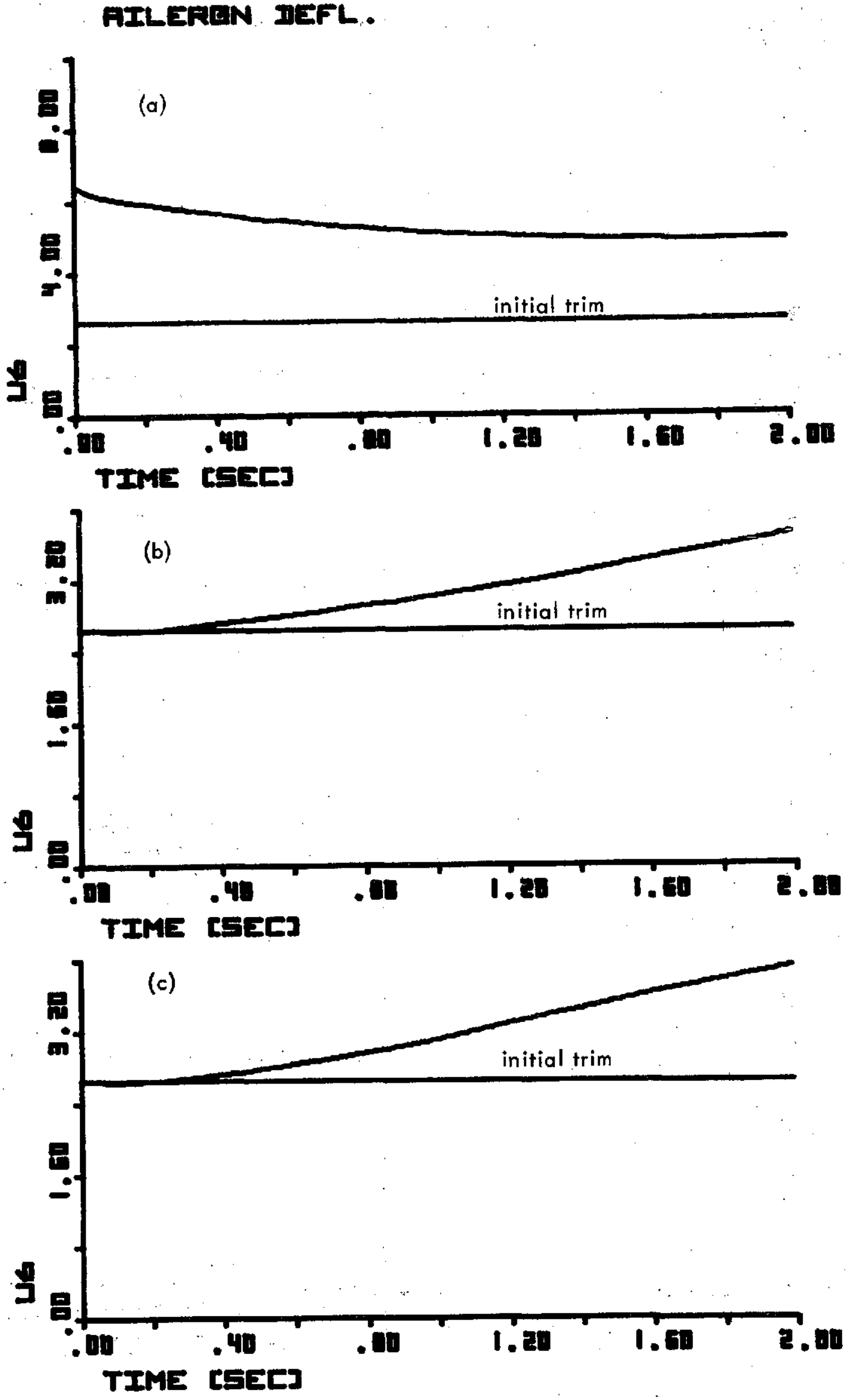


Figure 4.9, continued - 5 -

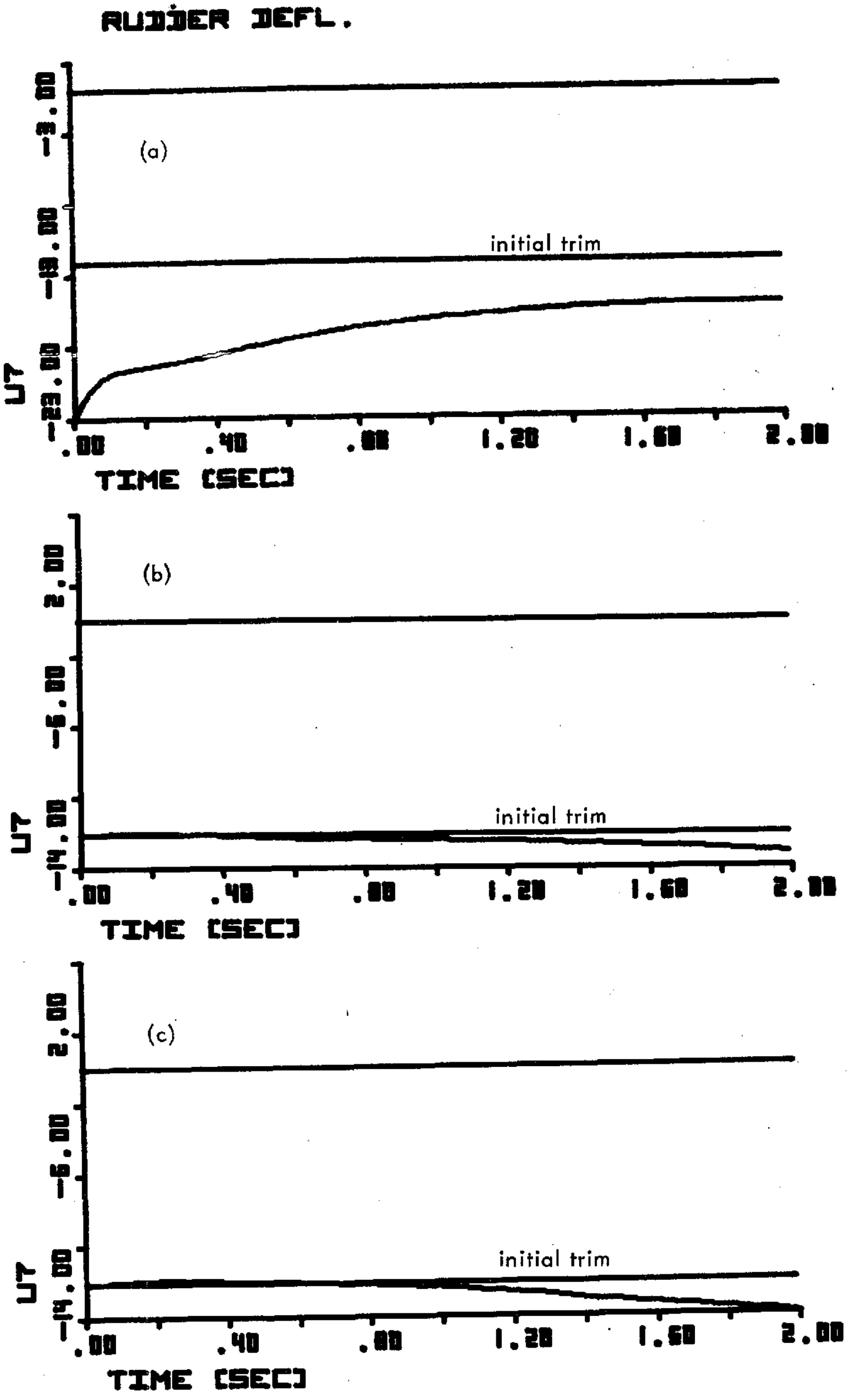
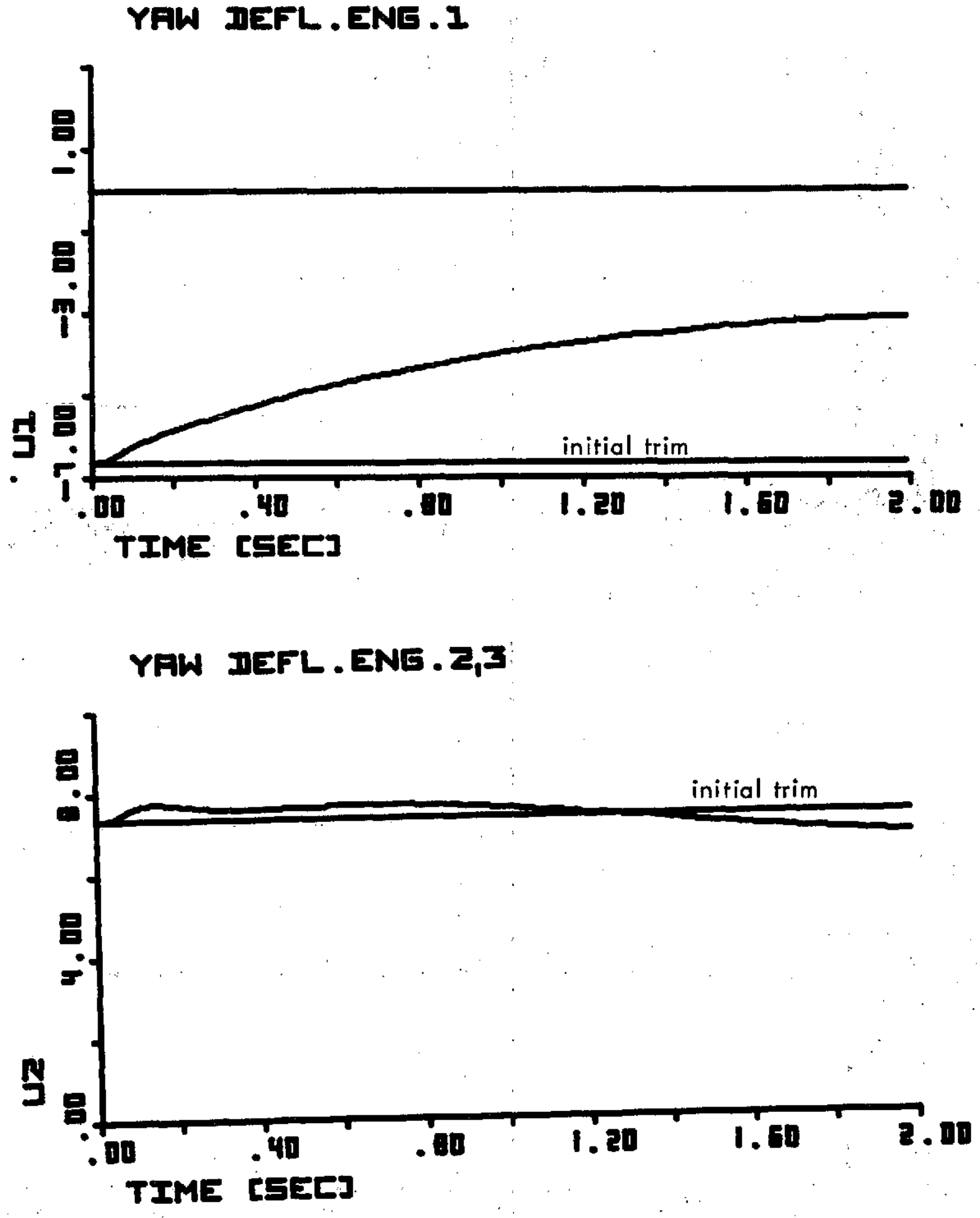
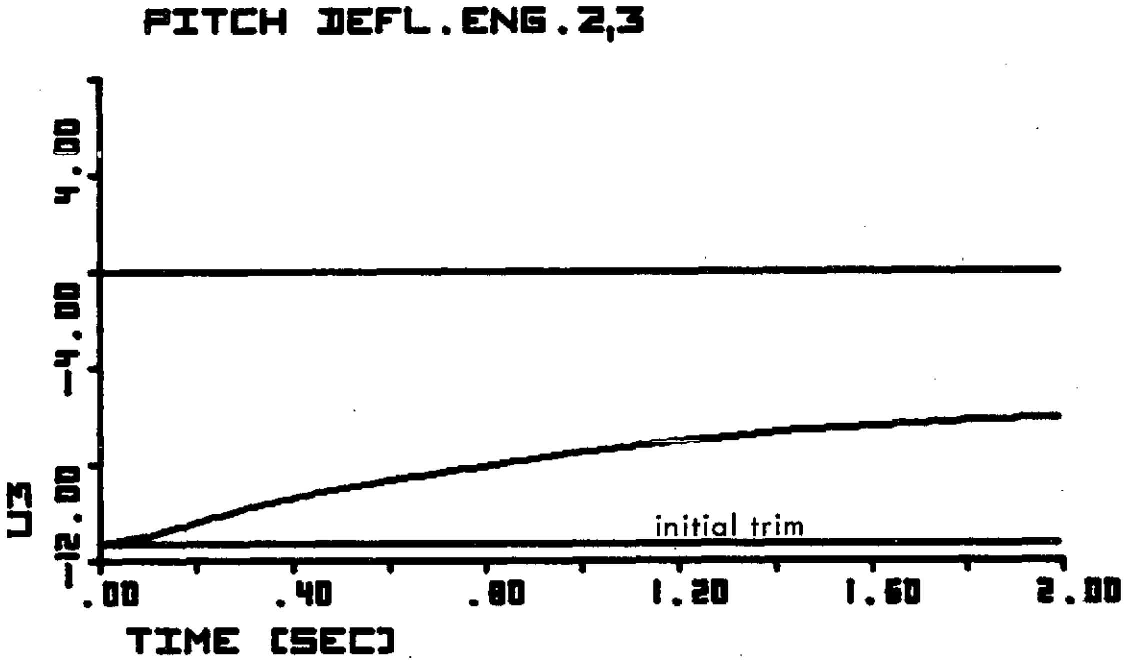


Figure 4.10 Dynamic Response of Optimum Control with Improved Kalman Filter for a 2° Step in Sideslip Angle



C-2

Figure 4.10, continued - 2 -


PITCH DEFL.ENG.2,3

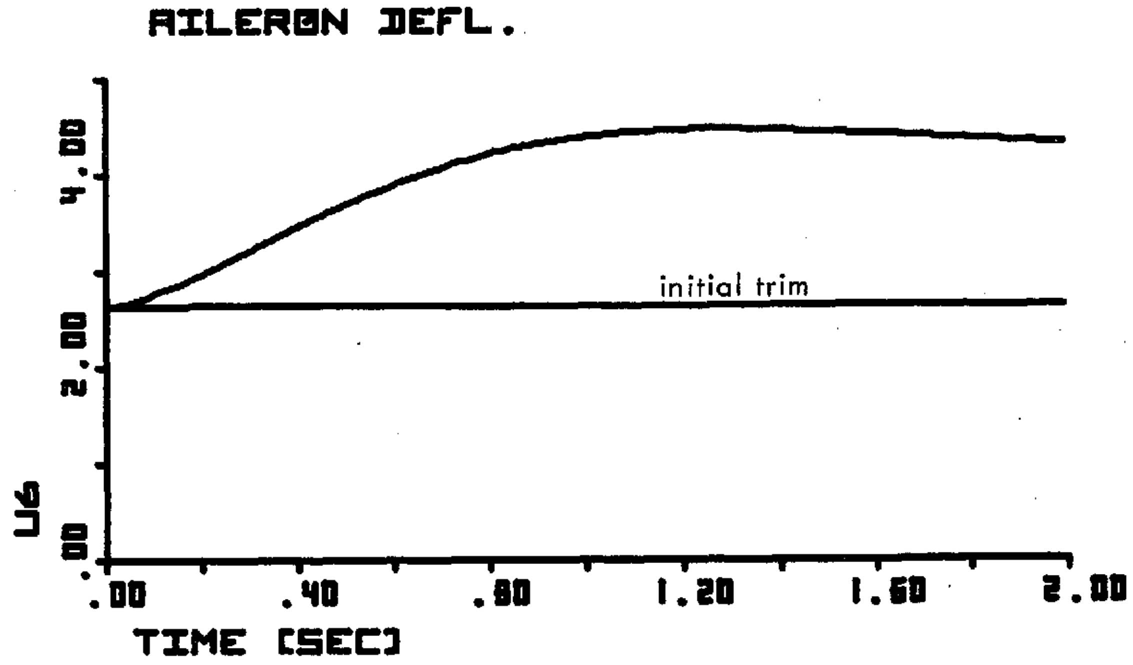

AILERON DEFL.

Figure 4.10, continued - 3 -

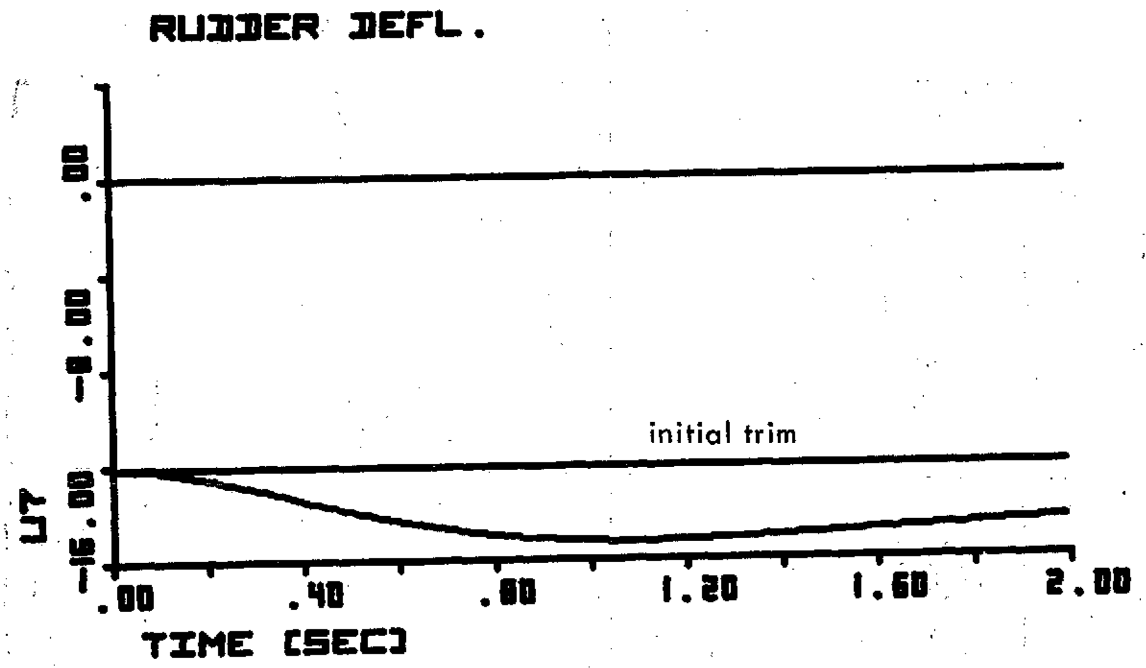


Table 4.6 Convergence of Optimum Steady State Control Level to Trim Control as Control Weighting Decreases

$$\begin{array}{cccccccc}
 & \sigma = 1 & & \sigma = 0.1 & & \sigma = 0.01 & & \begin{array}{c}\text{(trim solution)}\\ \sigma = 0\end{array} \\
u_x(\infty) = & \begin{bmatrix} -0.85 \\ -0.56 \\ -0.61 \\ 0 \\ 0 \\ 0.22 \\ -0.11 \end{bmatrix} & \rightarrow & \begin{bmatrix} -0.20 \\ -0.13 \\ -0.14 \\ 0 \\ 0 \\ 0.05 \\ -0.25 \end{bmatrix} & \rightarrow & \begin{bmatrix} -0.02 \\ -0.01 \\ -0.01 \\ 0 \\ 0 \\ 0.01 \\ -0.03 \end{bmatrix} & \rightarrow & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
u_z(\infty) = & \begin{bmatrix} -5.78 \\ 7.90 \\ -10.64 \\ 0 \\ 0 \\ 2.43 \\ -11.00 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.17 \\ 7.46 \\ -12.24 \\ 0 \\ 0 \\ 2.03 \\ -10.76 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.36 \\ 7.35 \\ -12.39 \\ 0 \\ 0 \\ 2.06 \\ -10.96 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.39 \\ -7.34 \\ -12.42 \\ 0 \\ 0 \\ 2.06 \\ -10.96 \end{bmatrix} \\
u(\infty) = & \begin{bmatrix} -6.63 \\ 7.35 \\ -11.27 \\ 0 \\ 0 \\ 2.65 \\ -12.11 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.37 \\ 7.33 \\ -12.38 \\ 0 \\ 0 \\ 2.08 \\ -11.01 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.38 \\ 7.33 \\ -12.40 \\ 0 \\ 0 \\ 2.07 \\ -10.99 \end{bmatrix} & \rightarrow & \begin{bmatrix} -7.39 \\ 7.34 \\ -12.42 \\ 0 \\ 0 \\ 2.06 \\ -10.96 \end{bmatrix}
\end{array}$$

5. CONCLUSIONS AND FUTURE WORK

Solutions to the trim problem can be efficiently calculated by the TRIMS program using the numerical methods described in Section 3.1. The results of this investigation indicate that numerical solution by the Newton-Raphson method is preferable to the steepest descent method because it yields faster convergence and does not require the user to specify an iteration step size σ . If the initial guess of the solution used to start the Newton-Raphson method is not in the region of convergence then the method may not converge or may converge to the wrong solution. In this case the steepest descent method should be used for the first few iterations to generate a good starting solution to the Newton-Raphson method. This hybrid method could be implemented in the TRIMS program with minor modifications. However, it appears that for most practical trim problems an initial guess of $\delta = 0$ is always in the region of convergence.

For the linear trim problem, a diagonal weighting matrix R in the quadratic performance criterion is sufficient in finding the "best" trim solution with respect to the limits on the deflection angles. Introducing nonzero values for the off-diagonal elements of R complicates the selection of the performance criterion and does not lead to a better trim solution than could be obtained by use of a diagonal matrix. Starting from the trim solution for a given diagonal R matrix, consider the problem of searching for a more desirable trim solution. The penalty function method for varying the diagonal elements of R is a viable approach for improving the trim solution that is easy to use. The penalty function method would be considerably facilitated if the computer computation of the trim solution is performance in a conversational mode of operation rather than a batch mode. In the former case, the user can examine the trim solution and then immediately try a new R matrix. The process can be repeated in a single sitting as many times as is necessary.

The lateral trim solution in Figure 4.6 indicates that the Space Shuttle configuration does not have sufficient control authority when the SRM engines are not gimballed. (The improvement in the trim solution obtained by gimballing the SRM engines is an area for future study which can be performed by the TRIMS program with minor modifications to the block data subroutine.) If the constraint of zero net side force (i.e., $Y = 0$) is eliminated and the vehicle is only trimmed in roll and yaw, the maximum control deflections decrease by roughly an order-of-magnitude. In this case the trim solution is within the deflection limits. Hence, the control requirements increase significantly with the addition of the trim requirement $Y = 0$. Maintaining $Y = 0$ is not as critical as zero net roll and yaw torques because angular errors are multiplied by the vehicle velocity in computing the displacement from the nominal trajectory. This suggests removing the trim condition $Y = 0$ entirely or replacing it by $|Y| < \epsilon$. The value of ϵ depends on how much side displacement error ia acceptable. By varying the weighting matrix R in the performance criterion with flight time rather than holding it constant, significant improvement in the trim solution might be achieved. The problem of realizing a trim solution for a time-varying R matrix must also be considered.

Only the steady-state performance of the control system for bias inputs is considered in the trim calculation. The dynamic or transient response of the controls for fluctuating inputs must also be considered in the overall system design. For the nonlinear trim problem, the index of controllability defined in Section 3.3.2 is a quantitative measure for selecting the trim solution that results in the most controllable system with respect to the dynamic response problem. An integral E proportional to the control energy is defined in Section 3.3.1 and is computed using the controllability Grammian W. Another measure for selecting the trim solution is given by the value of E. If the trim problem is linear, then the value of the controllability index or E does not vary with the trim solution.

The basic question in studying the dynamic response problem is: "What is the maximum deflection of each control for the possible fluctations in the disturbance inputs?" One possible approach to the dynamic response problem is to examine the values of E_i, where E_i denotes the energy expended by the ith control to return the vehicle to trim. The values E_i can be readily computed from the controllability Grammain W. Although this approach has potential in gaining insight into dynamic response problem, it possesses two major limitations. First, there is no simple relationship between the energy E_i and the maximum value of the transient response curve showing the variation in the control deflection angle with time. Second, the control signal corresponding to E_i cannot be realized by a linear feedback control system. The trim problem concerns only the static performance and can be studied without considering the detailed design of the feedback control system. The dynamic response problem, however, concerns the closed-loop transient response and is strongly dependent on the design of the feedback control system.

The most realistic method and possibly the only practical method for studying the dynamic response problem is to design the control system and simulate the closed-loop performance. The application of optimum control theory provides a method for the design of a linear feedback control system that can solve both the trim problem and the dynamic response problem. The correlation between the trim solution and the optimum control solution derived in Section 3.4.1 indicates how the solution to the trim problem can be used to select the proper control weighting R in performance criterion of the optimum control approach. This saves design time since the trim problem is easier to solve. A computer program entitled Linear System Design (LSD) was developed at Singer-Kearfott that is capable of computing the optimum feedback system and simulating the closed-loop performance. Since LSD is a conversational program with an automated plotting capability, many different designs can be studied efficiently. An example illustrating the use of LSD to design an optimum feedback system for the lateral control of the Space Shuttle during ascent is described in Section 4.3. It is recommended a more extensive design effort be pursued using the optimum control approach.

APPENDIX A VECTOR NOTATION AND DIFFERENTIATION

In this appendix the notation used for handling differentiation with respect to vector quantities is reviewed for reference purposes. This notation is useful in describing the solution to the trim control problem.

Let x and y denote an n dimensional and an m dimensional (column) vector, respectively. Further, let α denote a scalar function of x and y and let f denote a vector function of x and y where the dimension of f is p .

$$x = \begin{bmatrix} x_i \\ \vdots \\ x_n \end{bmatrix} \qquad v = \begin{bmatrix} y_i \\ \vdots \\ y_m \end{bmatrix} \qquad f = f(x,y) \qquad \alpha = \alpha(x,y)$$

Differentiation of a vector by a scalar results in a (column) vector defined by

$$\dot{x} = dx/dt = \begin{bmatrix} dx_i/dt \\ \vdots \\ dx_n/dt \end{bmatrix}$$

On the other hand, differentiation of a scalar by a vector results in a row vector defined by

$$\partial\alpha/\partial x = [\,\partial\alpha/\partial x_1,\ \partial\alpha/\partial x_2,\ .\ .\ .\ ,\ \partial\alpha/\partial x_n]$$

The second partial of the scalar α with respect to x and y

$$\partial^2\alpha/\partial x\,\partial y = \partial/\partial y(\partial\alpha/\partial x)'$$

is an n by m matrix whose ijth element is defined by

$$(\partial^2\alpha/\partial x\partial y)_{ij} = \partial^2\alpha/\partial x_i\partial y_i$$

Differentiation of the vector function f with respect to the vector x is a p by n matrix whose ijth element is defined by

$$(\partial f/\partial x)_{ij} = \partial f_i/\partial x_j$$

Consider the scalar formed by the inner product of f and a constant vector λ of dimension p . The second partial of this scalar with respect to x and y

$$\partial^2(\lambda' f)/\partial x\,\partial y = \lambda'(\partial f/\partial x\,\partial y)$$

is an n by m matrix whose ℓjth element is given by

$$\left[\lambda'\left(\frac{\partial f}{\partial x\,\partial y}\right)\right]_{\ell j} = \sum_{k=1}^{p} \lambda_k \left(\frac{\partial f}{\partial x\,\partial y}\right)_{k\ell j}$$

The quantity of $\partial f/\partial x\,\partial y$ is a tensor whose $k\ell j$th element is defined by

$$(\partial f/\partial x\,\partial y)_{k\ell j} = \partial f_k/\partial x_\ell \partial y_\ell$$

APPENDIX B PARAMETERS OF SPACE SHUTTLE DYNAMICS

The equations defining the lateral-direction dynamics of the Space Shuttle during ascent through the atmosphere were derived in Section 4.1. The parameters required to compute the matrix coefficients in the linear equations of motion (4.10) are given in this appendix. The list of parameters appearing below indicates the parameter symbol, value, units, and a brief description. The data is given for twelve different points or flight times along the ascent trajectory and was furnished by Dr. S. Winder of MSFC.

In the column labeled VALUE, there appears either the numerical value or the word "table" or is left blank. The word "table" denotes that the numerical value varies with flight time and the twelve different values are listed in the tables at the end of this appendix. A blank denotes that the value of the parameter has not been specified. The unspecified parameters are the location and thrust of the SRM engines and the stability derivative $C_{y_{\delta a}}$. Most likely $C_{y_{\delta a}}$ is small and is assumed to be zero in this investigation. It is further assumed that the SRM engines are not gimballed but the provision for including the SRM engine deflections is incorporated into the equations.

The stability derivatives $C_{\ell p}$, C_{np}, C_{yr}, $C_{\ell r}$, C_{nr} were not included in the data furnished by MSFC. Their values listed below are rough estimates based on the vehicle configuration. These stability derivatives are not used in computing the trim solution but are required for the study of dynamic response.

LATERAL TRIM PARAMETERS

SYMBOL	VALUE	UNITS	DESCRIPTION
x_1	0	m	x, y, z positions of (top, orbiter) engine 1
y_1	0	m	
z_1	0	m	
x_2	0	m	x, y, z positions of (right orbiter) engine 2
y_2	1.346	m	
z_2	- 6.68	m	
x_3	0	m	x, y, z positions of (left orbiter) engine 3
y_3	- 1.346	m	
z_3	- 6.68		
x_4		m	x, y, z, positions of (right SRM) engine 4
y_4		m	
z_4		m	
x_5		m	x, y, z position of (left SRM) engine 5
y_5		m	
z_5		m	
x_{cg}	table	m	x, y, z position of center of gravity
y_{cg}	0	m	
z_{cg}	table	m	
x_{mrp}	21.6	m	x, y, z position of moment reference point
y_{mrp}	0	m	
z_{mrp}	- 1.47	m	

q	table	new./m^2	dynamic pressure
S	317.73	m^2	reference area
b	28.322	m	reference length
V	table	m/sec	velocity of the vehicle relative to the air
V_y	table	m/sec	side component of V (side wind velocity)
F	table	New.	thrust per orbiter engine
F_{SRM}		New.	thrust per SRM engine
$C_{y\beta}$	table	-	stability derivative
$C_{\ell\beta}$	table	-	stability derivative
$C_{n\beta}$	table	-	stability derivative
$\Delta C_{y\beta}$	table	-	change in $C_{y\beta}$ due to dorsal fins
$\Delta C_{\ell\beta}$	table	-	change in $C_{\ell\beta}$ due to dorsal fins
$(\Delta C_{n\beta})_{AFT}$	table	-	change in $C_{n\beta}$ due to aft dorsal fin
$(\Delta C_{n\beta})_{FORWARD}$	table	-	change in $C_{n\beta}$ due to forward dorsal fin
$C_{y_{\delta a}}$		-	stability derivative
$C_{\ell_{\delta a}}$	table	-	stability derivative
$C_{n_{\delta a}}$	table	-	stability derivative
$C_{y_{\delta r}}$	table	-	stability derivative
$C_{\ell_{\delta r}}$	table	-	stability derivative
$C_{n_{\delta r}}$	table	-	stability derivative
$C_{\ell p}$	-.01	-	stability derivative
C_{np}	- 0.03	-	stability derivative
C_{yr}	0.	-	stability derivative

$C_{\ell r}$	0.022	-	stability derivative
C_{nr}	- 0.11	-	stability derivative
$\bar{c}$	20.	m	length of mean aerodynamic cord
m	table	Kg	vehicle mass
I_x	table	Kg-m^2	vehicle moment of inertia about x axis
I_y	table	Kg-m^2	vehicle moment of inertia about y axis
I_z	table	Kg-m^2	vehicle moment of inertia about z axis
g	table	m/sec^2	acceleration of gravity
$\cos\theta_o$	table	-	cosine of nominal pitch angle
$\sin\theta_o$	table	-	sine of nominal pitch angle
Q_o	table	rad/sec	nominal pitch rate
U_o	table	m/sec	nominal velocity along x axis
W_o	table	m/sec	nominal velocity along z axis
$\delta_{i\,max}$	30	deg	maximum allowable rocket engine deflection (i=1, . . . ,5)
$\delta_{6\,max}$	table	deg	maximum allowable aileron deflection
$\delta_{7\,max}$	table	deg	maximum allowable rudder deflection
δ_6	0	m^2	reference area for drag induced by aileron control
δ_7	0	m^2	reference area for drag induced by rudder control
C_{D6}	0	-	drag coefficient for aileron control
C_{D7}	0	-	drag coefficient for rudder control
$C_{\ell p}$	- 0.1	-	stability derivative
C_{np}	- 0.03	-	stability derivative
C_{yr}	0.0	-	stability derivative
$C_{\ell r}$	0.022	-	stability derivative
C_{nr}	- 0.11	-	stability derivative

DATA: Stability Derivatives

(all data/radian)

flight time (sec)	$C_{y_{\delta a}}$	$C_{\ell_{\delta a}}$	$C_{n_{\delta a}}$	$C_{y_{\delta r}}$	$C_{\ell_{\delta r}}$	$C_{n_{\delta r}}$	$C_{y\beta}$	$C_{\ell\beta}$	$C_{n\beta}$
25	.0	-.0430	.0458	.504	.273	-.510	-1.66	-.283	.302
40	.0	-.0458	.0444	.408	.265	-.489	-1.68	-.285	.315
50	.0	-.0487	.0430	.462	.259	-.473	-1.70	-.286	.325
60	.0	-.0544	.0358	.394	.215	-.388	-1.83	-.291	.404
65	.0	-.0630	.0344	.319	.181	-.310	-1.99	-.298	.468
70	.0	-.0630	.0301	.300	.173	-.345	-2.05	-.326	.460
75	.0	-.0544	.0258	.292	.206	-.340	-1.97	-.384	.344
80	.0	-.0458	.0244	.217	.186	-.254	-1.92	-.356	.266
90	.0	-.0286	.0172	.132	.105	-.137	-1.93	-.299	.238
100	.0	-.0215	.00286	.0961	.055	-.105	-2.03	-.246	.269
110	.0	-.0158	-.00286	.0749	.0406	-.077	-1.98	-.196	.207
140	.0	-.00859	-.0114	.0573	.0286	-.061	-1.60	-.122	-.0284

DATA: Vehicle Parameters

flight time (sec)	m (Kg)	I_x (Kg-m^2)	I_y (Kg-m^2)	I_z (Kg-m^2)	x_{cg} (m)	z_{cg} (m)	F (New.)
25	.218E+7	.953E+8	.526E+9	.591E+9	23.345	- 1.58	1.650E+6
40	.201E+7	.856E+8	.490E+9	.547E+9	23.42	- 1.5847	1.760E+6
50	.190E+7	.794E+8	.468E+9	.519E+9	23.47	- 1.5914	1.825E+6
60	.179E+7	.733E+8	.445E+9	.491E+9	23.52	- 1.5953	1.885E+6
65	.174E+7	.702E+8	.434E+9	.478E+9	23.545	- 1.5979	1.920E+6
70	.169E+7	.671E+8	.423E+9	.464E+9	23.57	- 1.60	1.940E+6
75	.160E+7	.629E+8	.383E+9	.420E+9	24.13	- 1.4626	1.970E+6
80	.154E+7	.606E+8	.372E+9	.405E+9	24.18	- 1.455	1.980E+6
90	.144E+7	.559E+8	.348E+9	.375E+9	24.33	- 1.440	2.025E+6
100	.133E+7	.512E+8	.326E+9	.346E+9	24.535	- 1.4327	2.040E+6
110	.122E+7	.466E+8	.303E+9	.317E+9	24.74	- 1.4255	2.060E+6
140	.914E+6	.329E+8	.234E+9	.228E+9	25.62	- 1.400	2.070E+6

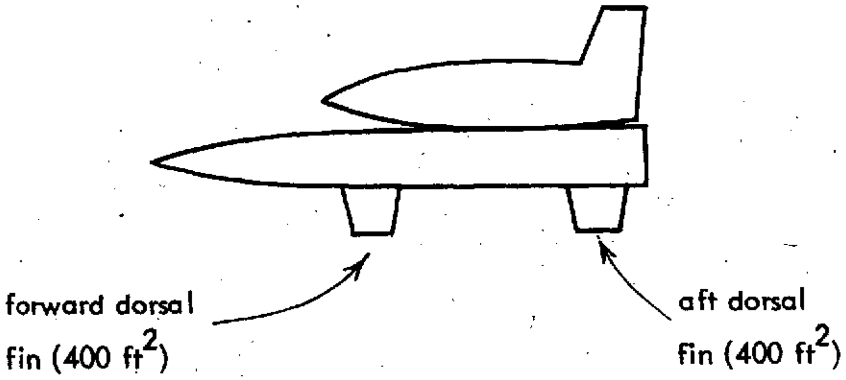
DATA: Trajectory Parameters

flight time (sec)	V (m/sec)	V_y (m/sec)	g (m/sec^2)	$\cos\theta_o$	$\sin\theta_o$	Q_o (rad/sec)	U_o (m/sec)	W_o (m/sec)	q (New/m^2)
25	95.4	2.0	9.8	- .012	1.0	- .203E-1	95.4	.279E+0	.482E+4
40	150.	9.0	9.8	.050	.999	- .512E-2	149.	.149E+3	.987E+4
50	190.	15.0	9.79	.125	.992	.450E-2	186.	.186E+3	.134E+5
60	241.	24.0	9.78	.222	.975	.112E-2	232.	.232E+3	.174E+5
65	272.	29.0	9.78	.274	.962	- .158E-1	257.	.257E+3	.194E+5
70	305.	34.0	9.78	.329	.944	.687E-1	283.	.283E+3	.212E+5
75	343.	40.0	9.77	.384	.923	- .103E+0	310.	.310E+3	.226E+5
80	385.	44.0	9.77	.449	.893	.412E-1	337.	.337E+3	.233E+5
90	486.	30.0	9.76	.566	.824	- .227E-2	392.	.392E+3	.217E+5
100	612.	0.0	9.74	.664	.748	.301E-3	445.	.445E+3	.165E+5
110	768.	0.0	9.73	.681	.732	- .851E-2	498.	.498E+3	.117E+5
140	1520.	0.0	9.68	.874	.486	- .100E-2	673.	.673E+3	.231E+4

DATA: Deflection Limits for Aerodynamic Surface Controls and Change in Stability Derivatives Due to Dorsal Fins

			(all data / degrees)			
flight time (sec)	$\delta_{7\,max}$ rudder hinge moment limit (deg)	$\delta_{6\,max}$ aileron hinge moment limit (deg)	$\Delta C_{y\beta}$	$\Delta C_{\ell\beta}$	$(\Delta C_{n\beta})_{AFT}$	$(\Delta C_{n\beta})$ FORWARD
25	no hinge limit	no hinge limit	- .011	.0031	.0064	- .004
40	42.0	71.8	- .012	.0032	.0067	- .0044
50	30.8	52.6	- .013	.0033	.0074	- .0048
60	23.5	40.0	- .015	.0036	.0085	- .0056
65	14.7	25.1	- .016	.0038	.0094	- .006
70	8.19	14.1	- .017	.0042	.0104	- .006
75	5.54	9.47	- .0165	.0042	.01	- .0058
80	5.23	8.91	- .014	.0035	.0088	- .005
90	6.27	10.69	- .0105	.0027	.0075	- .0044
100	10.23	17.5	- .008	.0017	.005	- .0028
110	19.67	33.64	- .006	.0014	.004	- .0022
140	no hinge limit	no hinge limit	- .004	.001	.0028	- .0015
* hard limits	± 30	40 up-15 down				

* Hard deflection angle limit is used when less than hinge moment limit.



APPENDIX C TRIMS COMPUTER PROGRAM

1. PROGRAM USAGE

Input

The input data to the TRIMS program consists of punched cards. The data deck is divided into cases where for example each case computes the trim solution for different values of roll bias torque. There are seven punched cards per case with the first card containing the case title and the last card indicating whether another case follows or whether this is the last case to be run. A description of the information and format for punching these seven data cards per case is given in Table 1. A sample of an input data deck for a single case run is shown in Figure 1.

Output

The computer printout from the TRIMS program is a single page per case. The printout resulting from the data deck in Figure 1 is shown in Figure 2. The first part of the printout lists the information contained on the data cards and used to compute the trim solution. The trim solution is printed in a convenient tabular form with each row listing the seven trim angles in degrees for a particular flight time. The number of iterations required to compute the trim solution at each trajectory point is also indicated.

Options

Special options have been added to the program since the original development date of February, 1973. The purpose of these options is described in Table 2 including the modifications to the input data required to exercise these options.

TABLE 1: TRIMS PROGRAM INPUT DATA

CARD	COLUMNS	VARIABLE	FORMAT	DESCRIPTION
TITLE CARD				
1	1-72	LINE	72A1	Descriptive case title.
CONTROL CARD				
1	1-5	IGRAD	I5	= 1, use 1st order gradient method; = 2, use 2nd order gradient method.
1	11-20	EPS	E10.3	Upper bound used in the convergence criterion.
1	21-30	STEP	E10.3	Step size used in the 1st order gradient method; leave blank if 2nd order gradient method is used.
TRAJECTORY CARD				
1	1-60	JPT	12I5	If trajectory point no. k, k=1, ..., 12, is to be used then punch a 1 in column 5k ; otherwise punch a 0 in column 5k.
CARD CONTAINING BIAS TORQUES				
1	1-10	YBT	E10.0	Yaw bias torque.
1	11-20	RBT	E10.0	Roll bias torque.
CARDS CONTAINING WEIGHTING FACTORS				
1	1-70	W1	7E10.0	Seven weighting factors in performance criterion for adjusting maximum deflection angles.
2	1-70	W2	7E10.0	Seven weighting factor in performance criterion for adjusting aerodynamic (drag) and thrust losses due to trim.
CASE PARTITION CARD				
1	1	IGØ	I5	= 1, another case follows = 2, last case.

FIGURE 1: EXAMPLE OF INPUT TO TRIMS COMPUTER PROGRAM

```
STUDY OF LATERAL TRIM FOR SPACE SHUTTLE     NO SRM BIAS
    2        0.0001      0.
    1    1    1    1    1    1    1    1    1    1    1    1
 0.         0.
3000.      3000.      3000.      3000.      3000.      3000.      3000.
 0.         0.         0.         0.         0.         0.         0.
2
```

FIGURE 2: EXAMPLE OF OUTPUT FROM TRIMS COMPUTER PROGRAM

```
    CASE  1        STUDY OF LATERAL TRIM FOR SPACE SHUTTLE    NO SRM BIAS
    ----

COMPUTATION CONTROL PARAMETERS
------------------------------            USE  2ND  ORDER GRADIENT METHOD
                                                ---
                                                          UPPER BOUND USED IN THE CONVERGENCE CRITERION =  0.100E-03
TRAJECTORY POINTS
-----------------                   1    1    1    1    1    1    1    1    1    1    1    1


SYSTEM DYNAMICS PARAMETERS
--------------------------             YAW BIAS TORQUE =        0.0
                                      ROLL BIAS TORQUE =        0.0


PERFORMANCE CRITERION PARAMETERS
--------------------------------                W11 =3000.00            W21 =    0.0
                                                W12 =3000.00            W22 =    0.0
                                 WEIGHTING      W13 =3000.00            W23 =    0.0
                                 FACTORS        W14 =3000.00            W24 =    0.0
                                                W15 =3000.00            W25 =    0.0
                                                W16 =3000.00            W26 =    0.0
                                                W17 =3000.00            W27 =    0.0


TRIM DEFLECTION ANGLES
----------------------
                                  TRAJ.  FLIGHT                                DELTA                                          NO. OF
                                   PT.    TIME    (1)       (2)       (3)       (4)       (5)       (6)       (7)               ITERATIONS
                                .............................................................................
                                .                                                                           .
                                .    1    25.0    0.09     -0.00      0.13      0.0       0.0       0.10     -0.10        .          1
                                .    2    40.0    0.45     -0.04      0.59      0.0       0.0       0.14     -0.26        .          1
                                .    3    50.0    0.87     -0.13      1.24      0.0       0.0       0.81     -0.26        .          1
                                .    4    60.0    1.58     -0.31      2.40      0.0       0.0       1.43     -0.36        .          1
                                .    5    65.0    2.04     -0.48      3.50      0.0       0.0       1.88     -0.39        .          1
                                .    6    70.0    1.71      0.02      2.57      0.0       0.0       0.25     -0.88        .          1
                                .    7    75.0    2.26     -0.24      4.03      0.0       0.0       0.39     -0.89        .          1
                                .    8    80.0    2.42     -0.39      5.15      0.0       0.0       0.78     -1.03        .          1
                                .    9    90.0    0.54      0.12      3.48      0.0       0.0       1.31     -0.94        .          1
                                .   10   100.0   -0.00     -0.00      0.00      0.0       0.0       0.00     -0.00        .          1
                                .   11   110.0    0.00      0.00     -0.00      0.0       0.0       0.00      0.00        .          0
                                .   12   140.0    0.00      0.00     -0.00      0.0       0.0       0.00      0.00        .          0
                                .                                                                           .
                                .............................................................................
                                                  TOP       YAW     PITCH      YAW     PITCH  AILERON   RUDDER
                                                <----    ORBITER    ----><---- SRM  ---->
```

TABLE 2: PROGRAM OPTIONS

Option 1 - The program has the capability of disregarding the first trim equality constraint. This equation corresponds to the trim condition of zero net force in the y-direction. To exercise this option change the nonzero values of JPT on the trajectory data card from positive numbers to negative numbers.

Option 2 - The program has the capability of computing the trim solution for the case where the aileron is not used. To exercise this option change the nonzero values of JPT on the trajectory data-card from a magnitude of 1 to a magnitude of 2 (i.e., replace 1 by 2 and replace - 1 by - 2).

Option 3 - The program has the capability of replacing the performance criterion stored internally in the program with the quadratic performance criterion

$$r(\delta) = (\delta_1/c_1)^2 + \ldots + (\delta_7/c_7)^2$$

where $c_1, \ldots, c_7$ are seven constants specified by the user at execution time. To exercise this option replace the fourteen weighting factors in the input data with the values

$$W1(\ell) = -c_\ell \qquad \ell = 1, \ldots, 7$$

$$W2(\ell) = 0.$$

2. PROGRAM DESCRIPTION

TRIMS is a FORTRAN IV computer program composed of a single main or executive routine and many subroutines. The program subroutines may be viewed as divided into two main groups. The first group is comprised of the main routine, entitled TRIMS, plus seven basic subroutines which form the heart of the program. These are listed in Table 3 together with a brief description of their function. The second group contains the utility subroutines which perform a specific matrix operation such as invert a matrix or print out a matrix. There are thirteen of these subroutines which are listed in Table 4. With the exception of GMSYMM, all of the utility subroutines are found in the IBM Scientific Subroutine Package *.

In addition to the calling lists, the transfer of information into and out from the subroutines is achieved by means of five named CØMMØNS. Their names are listed in Table 5 together with a brief functional description. The innerconnection between the main routine, the seven basic subroutines, and the five named CØMMØNS summarizing where each is used is shown in Table 6. The variables in each of the named CØMMØNS are listed and defined in Table 7. The other variables in the program not in a named CØMMØN are listed in Table 8.

In the following pages the FORTRAN source listing of each subroutine is given. The beginning of each listing contains comment cards describing the subroutine which includes the purpose, input variables, output variables, and the subroutines called. Flow diagrams are also given for each of the subroutines with the exception of the IBM SSP subroutines.

---

* System/360 Scientific Subroutine Package, Version *III*, Programmer's Manual, IBM publication GH20-0205-4, Fifthe edition, August 1970.

TABLE 3 : MAIN ROUTINE AND BASIC SUBROUTINES

TRIM	- main routine controlling the basic computational steps.
BLOCK	- block data subroutine for storing data internally in the program.
INPUT	- subroutine used to read in and print out the input data.
OUTPUT	- subroutine used to print out the results of the program.
GRAD1	- subroutine for computing the deflection angles using the 1st order gradient method.
GRAD2	- subroutine for computing the deflection angles using the 2nd order gradient method.
SYSTEM	- subroutine containing the equations defining the system dynamics and the corresponding equations for evaluating the derivatives required by the gradient methods.
COST	- subroutine containing the equations defining the performance criterion and the corresponding derivatives.

TABLE4 : UTILITY SUBROUTINES

GYSYMM	- symmetrize a matrix
MCPY	- matrix copy
MSTR	- storage conversion of a matrix
L0C	- location in compressed-stored matrix
GMSUB	- subtract two general matrices
GMPRD	- product of two general matrices
GMTRA	- transpose of a general matrix
MPRD	- matrix product
CCUT	- partition a matrix by column
MINV	- matrix inversion
SINV	- invert a symmetric positive definite matrix
MFSD	- triangular factorization of a symmetric positive definite matrix
MX0UT	- print a matrix

TABLE 5 : NAMED CØMMØNS

/CØN/	dimension and accuracy parameters
/ARRAY/	values of trim equation, performance criterion, and their derivatives
/TRAJ/	trajectory information
/SYST/	data derived from the space shuttle configuration for computing the system dynamics and trim equation
/PERF/	data used to compute the performance criterion

TABLE 6: INNERCONNECTION OF SUBROUTINES AND NAMED COMMONS

where used requires	TRIMS	BLØCK	INPUT	ØUTPUT	GRAD1	GRAD2	SYSTEM	CØST	/CØN/	/ARRAY/	/TRAJ/	/SYST/	/PERF/
TRIMS			X	X	X	X			X				
BLOCK									X		X	X	X
INPUT									X		X	X	X
ØUTPUT													
GRAD1							X	X	X	X	X		
GRAD2							X	X	X	X	X		
SYSTEM										X		X	
CØST										X			X

TABLE 7: Variables in Named CØMMØN

Program Symbol	Dimension	Symbol	Explanation
/CØN/			
M	. . .	m	Number of trim angles.
NS	. . .	$\tilde{n}$	Number of trim equations.
KMAX	. . .	K_{max}	Maximum number of iterations allowed.
EPSO	. . .	ϵ_0	Relative tolerance used in subroutine SINV.
MPT	. . .	. . .	Maximum number of trajectory points allowed.
/ARRAY/			
AV	6	a	Constant terms in trim equations.
BV	6	$\underline{\bar{b}}(\delta)$	Terms in trim equations varying with trim angles.
BM	60	$\partial \underline{b} / \partial \underline{\delta}$	First derivatives of trim equations.
BT	6,60	$\partial^2 \underline{b} / \partial \underline{\delta}^2$	Second derivative of trim equations.
RS	. . .	r	Performance criterion.
RV	10	$\partial r / \partial \underline{\delta}$	First derivative of performance criterion.
RM	100	$\partial^2 r / \partial \underline{\delta}^2$	Second derivative of performance criterion.
/TRAJ/			
JPT	12	. . .	Index vector determining which trajectory points to use (see program input data).
TF	12	. . .	Flight times corresponding to the different possible trajectory points.
/SYST/			
YBT	. . .	. . .	Yaw bias torque (see program input data).
RBT	. . .	. . .	Roll bias torque (see program input data).
S	. . .	S	Reference area.
BREF	. . .	b_{ref}	Reference length.
X1,Y1,Z1	. . .	x_1, y_1, z_1	Coordinates of (top orbiter) engine 1.
X2,Y2,Z2	. . .	x_2, y_2, z_2	Coordinates of (right orbiter) engine 2.
X3,Y3,Z3	. . .	x_3, y_3, z_3	Coordinates of (left orbiter) engine 3.
X4,Y4,Z4	. . .	x_4, y_4, z_4	Coordinates of (right SRM) engine 4.
X5,Y5,Z5	. . .	x_5, y_5, z_5	Coordinates of (left SRM) engine 5.
XMRP	. . .	x_{mrp}	Coordinates of moment reference point.
YMRP	. . .	y_{mrp}	
ZMRP	. . .	z_{mrp}	
XCG	12	x_{cg}	Coordinates of center of gravity ($Y_{cg} = 0$).
ZCG	12	z_{cg}	

TABLE7: Variables in Named CØMMØN, Continued

Program Symbol	Dimension	Symbol	Explanation
/SYST/, Continued			
Q	12	q	Dynamic pressure.
V	12	V	Vehicle velocity relative to air.
VY	12	V_y	Side wind velocity.
F	12	F	Thrust per orbiter engine.
FSRM	12	F_{SRM}	Thrust per SRM engine.
CYB	12	$C_{y\beta}$	Stability derivative.
CLB	12	$C_{\ell\beta}$	Stability derivative.
CNB	12	$C_{n\beta}$	Stability derivative.
DCYB	12	$\Delta C_{y\beta}$	Change in $C_{y\beta}$ due to dorsal fins.
DCLB	12	$\Delta C_{\ell\beta}$	Change in $C_{\ell\beta}$ due to dorsal fins.
DCNBA	12	$(\Delta C_{n\beta})AFT$	Change in $C_{n\beta}$ due to aft dorsal fin.
DCNBF	12	$(\Delta C_{n\beta})FORWARD$	Change in $C_{n\beta}$ due to forward dorsal fin.
CYA	12	$C_{y\delta a}$	Stability derivative.
CLA	12	$C_{\ell\delta a}$	Stability derivative.
CNA	12	$C_{n\delta a}$	Stability derivative.
CYR	12	$C_{y\delta r}$	Stability derivative.
CLR	12	$C_{\ell\delta r}$	Stability derivative
CNR	12	$C_{n\delta r}$	Stability derivative.
/PERF/			
W1	7	W_1	Vector of relative weighting factors (see program input data).
W2	7	W_2	Vector of relative weighting factors (see program input data).
DAMAX	12	$\delta_{a\,max}$	Maximum deflection angle allowed for aileron.
DRMAX	12	$\delta_{r\,max}$	Maximum deflection angle allowed for rudder.
QQ	12	q	Dynamic pressure.
DMAX	. . .	. . .	Maximum deflection angle allowed for orbiter rocket engines.
SA	. . .	S_a	Reference area corresponding to the drag induced by the aileron.
SR	. . .	S_r	Reference area corresponding to the drag induced by the rudder.
CDA	. . .	C_{Da}	Coefficient of drag corresponding to the aileron.
CDR	. . .	C_{Dr}	Coefficient of drag corresponding to the rudder.

TABLE 8 : Variables not in Named CØMMØN

Program Symbol	Dimension	Symbol	Explanation
I CASE	. . .	. . .	Number of the current case.
I GO	. . .	. . .	Index controlling sequence of cases.
I	. . .	. . .	Do loop index.
L	. . .	. . .	Number of the trajectory point.
I GRAD	. . .	. . .	Order of the gradient method to be used.
K	. . .	k	Number of the iteration.
EPS	. . .	ϵ	Convergence bound in gradient methods.
TIME	. . .	t	Flight time of the current trajectory point.
STEP	. . .	σ	Iteration step size used in first order gradient method.
DELTA	10	$\underline{\delta}$	Vector of trim angles.
LAMDA	6	$\underline{\lambda}$	Vector of Lagrange multipliers.
J	. . .	. . .	Do loop index.
DET	. . .	. . .	Determinant of a matrix
MNS	. . .	. . .	Difference between number of trim angles and trim equations
NØRM	. . .	. . .	Quantity for determining trim solution accuracy.
RU	10	r_u	Subvector of $\partial r/\partial \underline{\delta}$.
RX	10	r_x	Subvector of $\partial r/\partial \underline{\delta}$.
X	10	$\underline{x}$	Subvector of $\underline{\delta}$ (subroutine GRAD1).
X	10	. . .	Dummy vector (subroutine GRAD2).
BX	60	B_x	Square nonsingular submatrix of $\partial \underline{b}/\partial \underline{\delta}$.
DU	10	$\Delta \underline{u}$	Correction to subvector $\underline{u}$ of $\underline{\delta}$.
BU	60	B_u	Submatrix of $\partial \underline{b}/\partial \underline{\delta}$.
J*I*	. . .	. . .	Matrix element index
M2	. . .	. . .	$= m(m+1)/2$.
*I*ER	. . .	. . .	Index used to indicate errors in inverting a positive definite matrix.
HL	10	h_{λ}	Derivative of hamiltonian with respect to $\underline{\lambda}$.
R	100	$h_{\delta\delta}$	Second derivative of hamiltonian with respect to $\underline{\delta}$.

TABLE 8 : Variables not in Named CØMMØN (Continued)

Program Symbol	Dimension	Symbol	Explanation
Y	10	. . .	Dummy vector
DEL	10	$\Delta\underline{\delta}$	Correction to $\underline{\delta}$.
BR	60	BR^{-1}	Matrix product.
LAM	6	$\Delta\underline{\lambda}$	Correction to $\underline{\lambda}$.
BRB	36	$BR^{-1}B'$	Matrix product.
HD	10	h_δ	Derivative of hamiltonian with respect to $\underline{\delta}$.
D	60	. . .	Dummy matrix.
B	60	$h_{\lambda\delta}$	Mixed second derivative of hamiltonian.
CYBCG	. . .	. . .	Stability derivative C_{y_β} about cg .
CNRCG	. . .	. . .	Stability derivative C_{n_r} about cg .
CLBCG	. . .	. . .	Stability derivative C_{ℓ_β} about cg .
CNBCG	. . .	. . .	Stability derivative C_{n_β} about cg .
CLACG	. . .	. . .	Stability derivative C_{ℓ_a} about cg .
CNACG	. . .	. . .	Stability derivative C_{n_a} about cg .
CLRCG	. . .	. . .	stability derivative C_{ℓ_r} about cg .
C1	. . .	. . .	Cos 18°.
C2	. . .	. . .	Cos 12°.
C3	. . .	. . .	Cos 3.5°.
C4	. . .	. . .	Cos 15°.
S1	. . .	. . .	Sin 18°.
S2	. . .	. . .	Sin 12°.
S3	. . .	. . .	Sin 3.5°.
S4	. . .	. . .	Sin 15°.

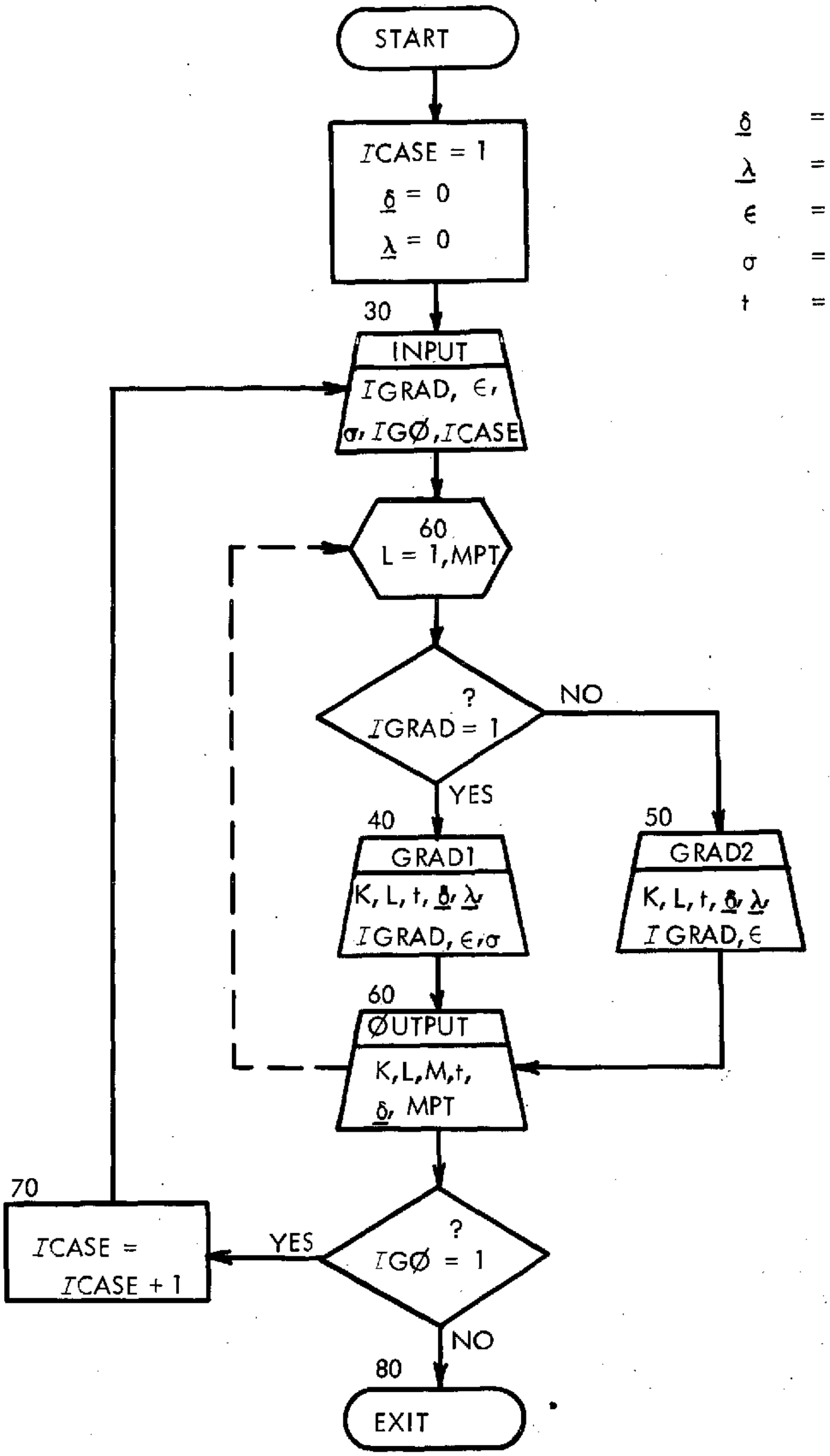
TABLE 8 : Variables not in Named CØMMØN (Continued)

Program Symbol	Dimension	Symbol	Explanation
IJ	...	...	Matrix element index.
QS	...	...	Product qS .
QSB	...	...	Product qSb_{ref} .
RAD	...	...	Conversion factor from radians to degrees.
BETA	...	β	Side slip angle.
II	...	...	Vector element index.
C	...	...	Dummy vector.

```
C     ------------------------------------------------------------------
C
C                 ***  TRIMS  COMPUTER  PROGRAM  ***
C
C        DEVELOPED BY:  M.HUTTON    THE SINGER CO.    FEBRUARY 1973
C
C     ------------------------------------------------------------------
C
C     PURPOSE  MAIN ROUTINE FOR EXECUTION OF COMPUTIONS OF LATERAL
C     -------  TRIM ANGLES FOR SPACE SHUTTLE.
C
C     INPUTS     (SEE SUBROUTINE INPUT).
C     ------
C
C     OUTPUTS    (SEE SUBROUTINE OUTPUT).
C     -------
C
C     SUBROUTINES CALLED      INPUT , GRAD1 , GRAD2 , OUTPUT .
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      REAL   LAMDA
      DIMENSION   DELTA(10) , LAMDA(6)
      COMMON /CON/ M , NS , KMAX , EPS0 , MPT
C
C
C *** INITIALIZATION
      ICASE = 1
      DO 10 I=1,M
   10 DELTA(I) = 0.
      DO 20 I=1,NS
   20 LAMDA(I) = 0.
C
C *** ENTER INPUT DATA
   30 CALL  INPUT(IGRAD,EPS,STEP,IGO,ICASE)
C
C *** COMPUTE TRIM SOLUTION FOR EACH OF THE SELECTED POINTS ALONG
C     THE TRAJECTORY
      DO 60 L=1,MPT
C
C *** DETERMINE COMPUTATIONAL METHOD TO BE USED
      IF(IGRAD-1) 50,40,50
C
C *** COMPUTE TRIM SOLUTION USING 1ST ORDER GRADIENT
   40 CALL  GRAD1(K,L,TIME,DELTA,LAMDA,IGRAD,EPS,STEP)
      GO TO 60
C
C *** COMPUTE TRIM SOLUTION USING 2ND ORDER GRADIENT
   50 CALL  GRAD2(K,L,TIME,DELTA,LAMDA,IGRAD,EPS)
C
C *** PRINT RESULTS
   60 CALL  OUTPUT(K,L,M,TIME,DELTA,MPT)
C
C *** TEST IF END OF COMPUTER RUN
      IF(IGO-1) 80,70,80
C
```

```
C *** GO TO THE NEXT CASE                                               TRM 0600
   70 ICASE = ICASE + 1                                                 TRM 0610
      GO TO 30                                                          TRM 0620
C                                                                       TRM 0630
   80 CALL  EXIT                                                        TRM 0640
      END                                                               TRM 0650
```

TRIMS Flow Diagram



δ = DELTA
λ = LAMDA
ϵ = EPS
σ = STEP
t = TIME

```
C     ------------------------------------------------------------------BLK 0010
C                                                                       BLK 0020
C     SURPROGRAM  BLOCK                                                 BLK 0030
C                                                                       BLK 0040
C     ------------------------------------------------------------------BLK 0050
C                                                                       BLK 0060
C     PURPOSE  BLOCK DATA SUBROUTINE FOR STORING DATA INTERNALLY IN THE BLK 0070
C     -------  PROGRAM. THIS IS THE AERODYNAMIC DATA REQUIRED TO COMPUTEBLK 0080
C              COEFFICIENTS OF LATERAL TRIM EQUATIONS.                  BLK 0090
C                                                                       BLK 0100
C     COMMON BLOCKS    /CON/ , /TRAJ/ , /SYST/ , /PERF/                 BLK 0110
C     ------ ------                                                     BLK 0120
C                                                                       BLK 0130
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BLK 0140
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BLK 0150
C                                                                       BLK 0160
      BLOCK DATA                                                        BLK 0170
C                                                                       BLK 0180
C                                                                       BLK 0190
      COMMON /CON/ M , NS , KMAX , EPSO , MPT                           BLK 0200
      COMMON /TRAJ/  JPT(12) , TF(12)                                   BLK 0210
      COMMON /SYST/  YBT, RBT,     X1, X2, X3, X4, X5, XMRP,            BLK 0220
     1               S, BREF,      Y1, Y2, Y3, Y4, Y5, YMRP,            BLK 0230
     2                             Z1, Z2, Z3, Z4, Z5, ZMRP,            BLK 0240
     3     XCG(12),     ZCG(12),        Q(12),        V(12),     VY(12), BLK 0250
     4       F(12),    FSRM(12),      CYB(12),      CLB(12),    CNB(12), BLK 0260
     5    DCYB(12),    DCLB(12),    DCNBA(12),    DCNBF(12),    CYA(12), BLK 0270
     6     CLA(12),     CNA(12),      CYR(12),      CLR(12),    CNR(12)  BLK 0280
      COMMON /PERF/  W1(7), W2(7), DAMAX(12),DRMAX(12), QQ(12),         BLK 0290
     1               DMAX, SA, SR, CDA, CDR                             BLK 0300
C                                                                       BLK 0310
C                                                                       BLK 0320
      DATA  M,NS,KMAX,EPSO,MPT   / 7, 3,  3, 0.00001, 12  /             BLK 0330
C                                                                       BLK 0340
      DATA  TF      /  25.0   , 40.0   , 50.0   , 60.0   , 65.0  , 70.0  ,  BLK 0350
     1                 75.0   , 80.0   , 90.0   , 100.0  , 110.0 , 140.0  / BLK 0360
C                                                                       BLK 0370
      DATA  X1,Y1,Z1  /  0.,  0.    , -9.34  /                          BLK 0380
      DATA  X2,Y2,Z2  /  0.,  1.346, -6.68   /                          BLK 0390
      DATA  X3,Y3,Z3  /  0., -1.346, -6.68   /                          BLK 0400
      DATA  X4,Y4,Z4  /  0.,  0.    ,  0.    /                          BLK 0410
      DATA  X5,Y5,Z5  /  0.,  0.    ,  0.    /                          BLK 0420
C                                                                       BLK 0430
      DATA  XCG     /  23.345, 23.42 , 23.47 , 23.52 , 23.545, 23.57 ,  BLK 0440
     1                 24.13 , 24.18 , 24.33 , 24.535, 24.74 , 25.62  / BLK 0450
C                                                                       BLK 0460
      DATA  ZCG     / -1.58  ,-1.5847,-1.5914,-1.5953,-1.5979,-1.60   ,  BLK 0470
     1                -1.4626,-1.455 ,-1.440 ,-1.4327,-1.4255,-1.400   / BLK 0480
C                                                                       BLK 0490
      DATA  XMRP,YMRP,ZMRP  / 21.6, 0., -1.47  /                        BLK 0500
C                                                                       BLK 0510
      DATA  Q       /  .482E+4    , .987E+4    , .134E+5    , .174E+5    ,  BLK 0520
     1                 .194E+5    , .212E+5    , .226E+5    , .233E+5    ,  BLK 0530
     2                 .217E+5    , .165E+5    , .117E+5    , .231E+4    /  BLK 0540
C                                                                       BLK 0550
      DATA  S , BREF  /  317.73 , 28.322  /                             BLK 0560
C                                                                       BLK 0570
      DATA  V       /  95.4  , 150.   , 190.   , 241.   , 272.   , 305.   ,  BLK 0580
     1                 343.  , 385.   , 486.   , 612.   , 768.   , 1520.  /  BLK 0590
```

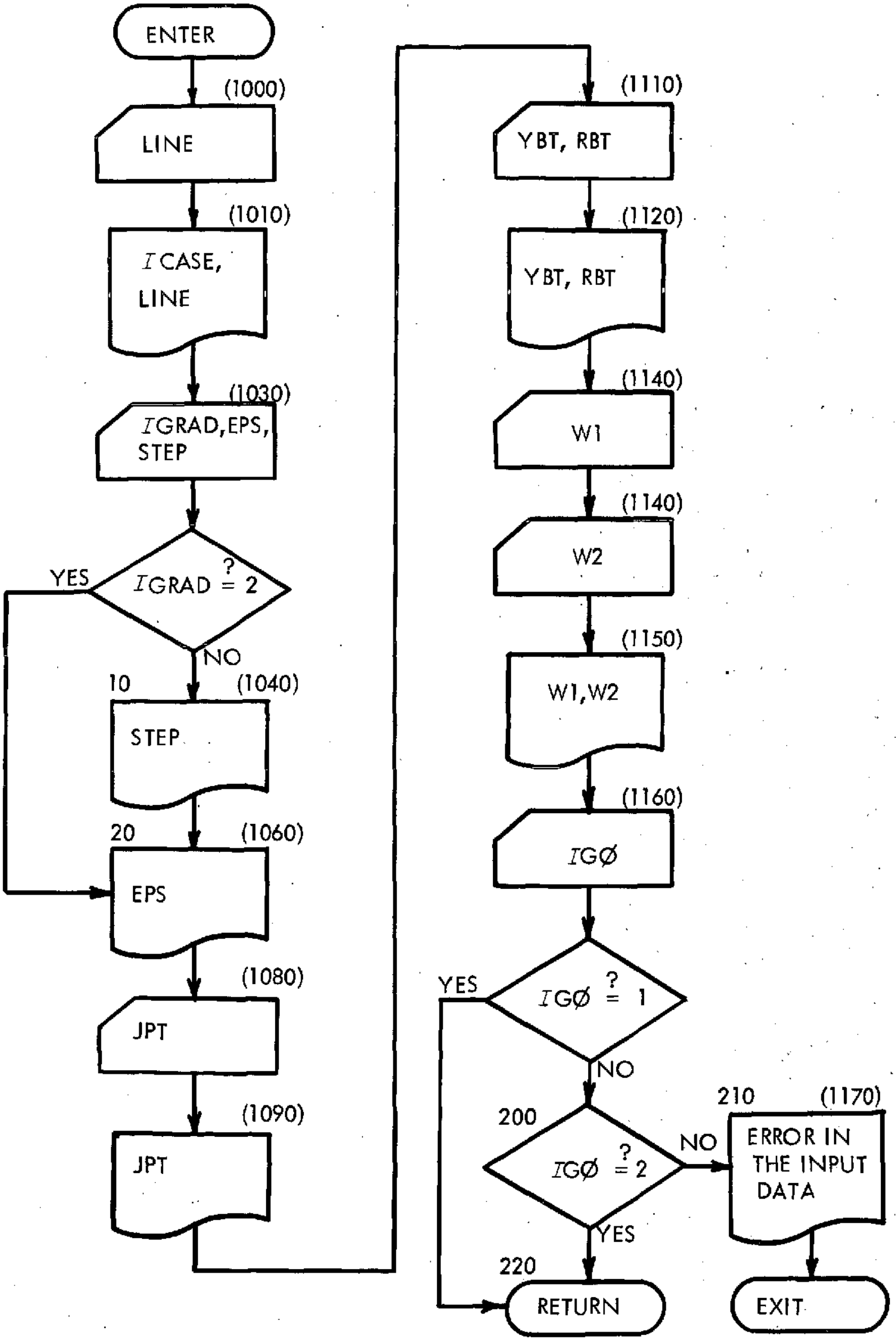
```
C                                                                      BLK 0600
      DATA  VY     /  2.     , 9.     , 15.    , 24.    , 29.    , 34.    ,  BLK 0610
     1                40.    , 44.    , 30.    , 0.     , 0.     , 0.     /  BLK 0620
C                                                                      BLK 0630
      DATA  F      /  1.65E+6   , 1.76E+6   , 1.825E+6  , 1.885E+6  ,  BLK 0640
     1                1.92F+6   , 1.94E+6   , 1.97E+6   , 1.98E+6   ,  BLK 0650
     2                2.025E+6  , 2.04E+6   , 2.06E+6   , 2.07E+6   /  BLK 0660
C                                                                      BLK 0670
      DATA  FSRM   /  12*0.  /                                         BLK 0680
C                                                                      BLK 0690
      DATA  CYB    / -1.66   ,-1.68   ,-1.70   ,-1.83   ,-1.99   ,-2.05   ,  BLK 0700
     1               -1.97   ,-1.92   ,-1.93   ,-2.03   ,-1.98   ,-1.60   /  BLK 0710
C                                                                      BLK 0720
      DATA  CLB    / -.283   ,-.285   ,-.286   ,-.291   ,-.298   ,-.326   ,  BLK 0730
     1               -.384   ,-.356   ,-.299   ,-.246   ,-.196   ,-.122   /  BLK 0740
C                                                                      BLK 0750
      DATA  CNB    /  .302   , .315   , .325   , .404   , .468   , .460   ,  BLK 0760
     1                .344   , .266   , .238   , .269   , .207   ,-.0284  /  BLK 0770
C                                                                      BLK 0780
      DATA  DCYB   / -.011   ,-.012   ,-.013   ,-.015   ,-.016   ,-.017   ,  BLK 0790
     1               -.0165  ,-.014   ,-.0105  ,-.008   ,-.006   ,-.004   /  BLK 0800
C                                                                      BLK 0810
      DATA  DCLB   /  .0031  , .0032  , .0033  , .0036  , .0038  , .0042  ,  BLK 0820
     1                .0042  , .0035  , .0027  , .0017  , .0014  , .001   /  BLK 0830
C                                                                      BLK 0840
      DATA  DCNBA  /  .0064  , .0067  , .0074  , .0085  , .0094  , .0104  ,  BLK 0850
     1                .01    , .0088  , .0075  , .005   , .004   , .0028  /  BLK 0860
C                                                                      BLK 0870
      DATA  DCNBF  / -.004   ,-.0044  ,-.0048  ,-.0056  ,-.006   ,-.006   ,  BLK 0880
     1               -.0058  ,-.005   ,-.0044  ,-.0028  ,-.0022  ,-.0015  /  BLK 0890
C                                                                      BLK 0900
      DATA  CYA    /  12*0.  /                                         BLK 0910
C                                                                      BLK 0920
      DATA  CLA    / -.0430  ,-.0458  ,-.0487  ,-.0544  ,-.0630  ,-.0630  ,  BLK 0930
     1               -.0544  ,-.0458  ,-.0286  ,-.0215  ,-.0158  ,-.00859 /  BLK 0940
C                                                                      BLK 0950
      DATA  CNA    /  .0458  , .0444  , .043   , .0358  , .0344  , .0301  ,  BLK 0960
     1                .0258  , .0244  , .0172  , .00286 ,-.00286 ,-.0114  /  BLK 0970
C                                                                      BLK 0980
      DATA  CYR    /  .504   , .408   , .462   , .394   , .319   , .300   ,  BLK 0990
     1                .292   , .217   , .132   , .0961  , .0749  , .0573  /  BLK 1000
C                                                                      BLK 1010
      DATA  CLR    /  .273   , .265   , .259   , .215   , .181   , .173   ,  BLK 1020
     1                .206   , .186   , .105   , .055   , .0406  , .0286  /  BLK 1030
C                                                                      BLK 1040
      DATA  CNR    / -.510   ,-.489   ,-.473   ,-.388   ,-.310   ,-.345   ,  BLK 1050
     1               -.340   ,-.254   ,-.137   ,-.105   ,-.077   ,-.061   /  BLK 1060
C                                                                      BLK 1070
      DATA  DAMAX  /  40.    , 40.    , 40.    , 40.    , 25.1   , 14.1   ,  BLK 1080
     1                9.47   , 8.91   , 10.69  , 17.5   , 33.64  , 40.    /  BLK 1090
C                                                                      BLK 1100
C     DATA  DAMAX  /  15.    , 15.    , 15.    , 15.    , 15.    , 14.1   ,  BLK 1110
C    1                9.47   , 8.91   , 10.69  , 15.    , 15.    , 15.    /  BLK 1120
C                                                                      BLK 1130
      DATA  DRMAX  /  30.    , 30.    , 23.5   , 14.7   , 8.19   , 8.19   ,  BLK 1140
     1                5.54   , 5.23   , 6.27   , 10.23  , 19.67  , 30.0   /  BLK 1150
C                                                                      BLK 1160
      DATA  QQ     /  .482E+4   , .987E+4   , .134E+5   , .174E+5   ,  BLK 1170
     1                .194E+5   , .212E+5   , .226E+5   , .233E+5   ,  BLK 1180
     2                .217E+5   , .165E+5   , .117E+5   , .231E+4   /  BLK 1190
```

```
C
      DATA  DMAX, SA, SR, CDA, CDR  /  30. , 4*0.  /
C
      END
```

```
C     ------------------------------------------------------------------
C
C     SUBROUTINE  INPUT(IGRAD,EPS,STEP,IGO,ICASE)
C
C     ------------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE USED TO READ IN AND PRINT OUT THE INPUT DATA.
C     -------
C
C     INPUTS     ICASE  = NO. OF CURRENT CASE.
C     ------
C
C     OUTPUTS    IGRAD  = ORDER OF GRADIENT METHOD TO BE USED.
C     -------    EPS    = CONVERGENCE BOUND.
C                STEP   = STEP SIZE IF 1ST ORDER GRADIENT METHOD USED.
C                IGO    = INDEX CONTROLING SEQUENCE OF CASES.
C
C     SUBROUTINES CALLED    NONE
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  INPUT(IGRAD,EPS,STEP,IGO,ICASE)
C
C
 1000 FORMAT(72A1)
 1010 FORMAT(1H1,9X,4HCASE,I3,8X,72A1 / 10X,4H---- )
 1020 FORMAT(//5X,30HCOMPUTATION CONTROL PARAMETERS)
 1030 FORMAT(I5,5X,2E10.3)
 1040 FORMAT(5X,30H------------------------------,10X,61HUSE  1ST  ORDER
     1 GRADIENT METHOD        ITERATION STEP SIZE = ,E10.3 / 50X,3H---)
 1050 FORMAT(5X,30H------------------------------,10X,31HUSE  2ND  ORDER
     1 GRADIENT METHOD / 50X,3H---)
 1060 FORMAT(58X,48HUPPER BOUND USED IN THE CONVERGENCE CRITERION = ,
     1  F10.3)
 1070 FORMAT(5X,17HTRAJECTORY POINTS )
 1080 FORMAT(12I5)
 1090 FORMAT(5X,17H-----------------,13X,12I5 )
 1100 FORMAT(//5X,26HSYSTEM DYNAMICS PARAMETERS)
 1110 FORMAT(2F10.0)
 1120 FORMAT(5X,26H--------------------------,9X,
     1        18H YAW BIAS TORQUE =,F11.1 /
     2   40X,18HROLL BIAS TORQUE =,F11.1 )
 1130 FORMAT(//5X,32HPERFORMANCE CRITERION PARAMETERS)
 1140 FORMAT(7F10.0)
 1150 FORMAT(5X,32H--------------------------------,
     1                    13X,5HW11 =,F7.2,15X,5HW21 =,F7.2 /
     2                    50X,5HW12 =,F7.2,15X,5HW22 =,F7.2 /
     3 35X,9HWEIGHTING,6X,5HW13 =,F7.2,15X,5HW23 =,F7.2 /
     4 35X,9HFACTORS  ,6X,5HW14 =,F7.2,15X,5HW24 =,F7.2 /
     5                    50X,5HW15 =,F7.2,15X,5HW25 =,F7.2 /
     6                    50X,5HW16 =,F7.2,15X,5HW26 =,F7.2 /
     7                    50X,5HW17 =,F7.2,15X,5HW27 =,F7.2 )
 1160 FORMAT(I1)
 1170 FORMAT(//5X,66H* * *  ERROR IN THE INPUT DATA  --  COMPUTER RUN TE
     1RMINATED  * * * )
C
      DIMENSION   IDP(50) , ICP(50) , LINE(72)
```

```
      COMMON /CON/ M , NS , KMAX , EPS0 , MPT
      COMMON /TRAJ/  JPT(12) , TF(12)
      COMMON /SYST/  YBT, RBT
      COMMON /PERF/  W1(7), W2(7)
C
C
C *** ENTER CASE IDENTIFICATION TITLE
      READ(5,1000)   (LINE(I),I=1,72)
      WRITE(6,1010) ICASE , (LINE(I),I=1,72)
C
C *** ENTER COMPUTATIONAL CONTROL PARAMETERS
      WRITE(6,1020)
      READ(5,1030)   IGRAD , EPS , STEP
      IF(IGRAD-2) 10,20,10
   10 WRITE(6,1040)   STEP
      GO TO 30
   20 WRITE(6,1050)
   30 WRITE(6,1060)   EPS
C
C *** ENTER POINTS ALONG TRAJECTORY FOR COMPUTING TRIM
      WRITE(6,1070)
      READ(5,1080)      (JPT(I),I=1,MPT)
      WRITE(6,1090)     (JPT(I),I=1,MPT)
C
C *** ENTER SYSTEM DYNAMICS PARAMETERS
      WRITE(6,1100)
      READ(5,1110)    YBT , RBT
      WRITE(6,1120)   YBT , RBT
C
C *** ENTER PERFORMANCE CRITERION PARAMETERS
      WRITE(6,1130)
      READ(5,1140)   (W1(I),I=1,M)
      READ(5,1140)   (W2(I),I=1,M)
      WRITE(6,1150)    W1(1)  , W2(1)  , W1(2)  , W2(2)  , W1(3)  , W2(3)  ,
     1                 W1(4)  , W2(4)  , W1(5)  , W2(5)  , W1(6)  , W2(6)  ,
     2                 W1(7)  , W2(7)
C
C *** ENTER END OF CASE CARD
      READ(5,1160)   IGO
      IF(IGO-1) 200,220,200
  200 IF(IGO-2) 210,220,210
  210 WRITE(6,1170)
      CALL  EXIT
  220 CONTINUE
C
      RETURN
      END
```

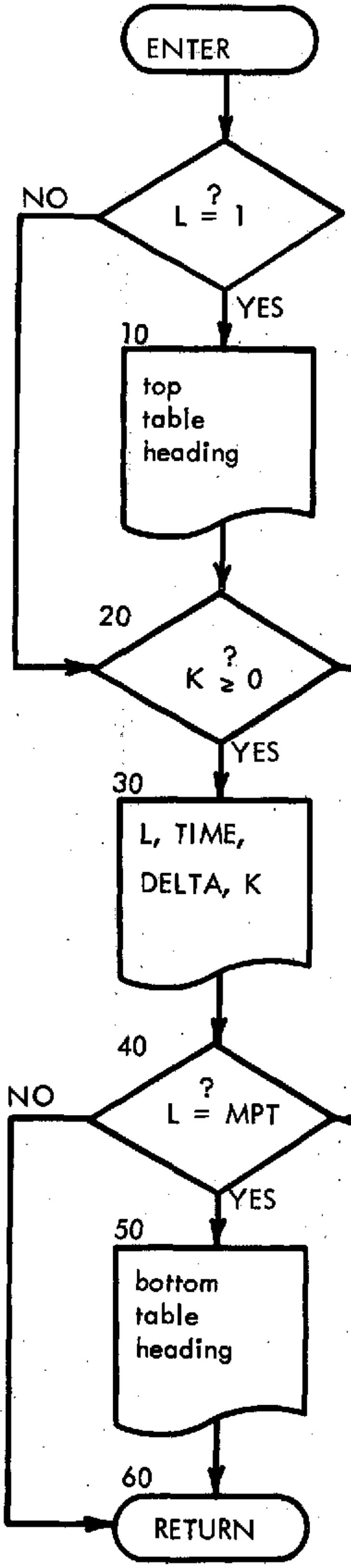
INPUT Flow Diagram



```
C     ------------------------------------------------------------------
C
C     SUBROUTINE  OUTPUT(K,L,M,TIME,DELTA,MPT)
C
C     ------------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE USED TO PRINT OUT THE RESULTS OF THE PROGRAM.
C     -------
C
C     INPUTS     K      = NO. OF ITERATIONS.
C     ------     L      = NO. OF THE TRAJECTORY POINT.
C                M      = NO. OF TRIM ANGLES.
C                TIME   = FLIGHT TIME OF THE TRAJECTORY POINT.
C                DELTA  = VECTOR OF TRIM ANGLES.
C                MPT    = INDEX USED TO DETERMINE LAST TRAJECTORY POINT.
C
C     OUTPUTS    NONE
C     -------
C
C     SUBROUTINES CALLED    NONE
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  OUTPUT(K,L,M,TIME,DELTA,MPT)
C
C
 1010 FORMAT(////5X,22HTRIM DEFLECTION ANGLES
     1          /5X,22H----------------------,7X,13HTRAJ.  FLIGHT,26X,
     2 5HDELTA,32X,6HNO. OF / 35X,85HPT.    TIME    (1)     (2)     (3)
     3    (4)     (5)     (6)     (7)         ITERATIONS  )
C1020 FORMAT(35X,I3,2X,F6.1,1X,7F8.2,7X,I5)
 1020 FORMAT(31X,1H.,3X,I3,2X,F6.1,1X,7F8.2,4X,1H.,2X,I5)
 1030 FORMAT( 48X,55H  TOP     YAW    PITCH    YAW    PITCH  AILERON  RU
     1DDER , /48X,40H<----   ORBITER     ----><---- SRM  ----> )
 1040 FORMAT(31X,77H....................................................
     1......................... )
 1050 FORMAT(31X,1H.,75X,1H.)
C
      DIMENSION   DELTA(1) , ANGLE(10)
      DATA  RAD  / 57.2957795 /
C
C
      IF(L-1) 20,10,20
   10 WRITE(6,1010)
      WRITE(6,1040)
      WRITE(6,1050)
```

```
20 IF(K) 40,30,30                                                      OUT 0480
30 DO 35 I=1,M                                                         OUT 0485
35 ANGLE(I) = RAD * DELTA(I)                                           OUT 0486
   WRITE(6,1020)  L , TIME , (ANGLE(I),I=1,M) , K                      OUT 0490
40 IF(L-MPT) 60,50,60                                                  OUT 0500
50 WRITE(6,1050)                                                       OUT 0510
   WRITE(6,1040)                                                       OUT 0520
   WRITE(6,1030)                                                       OUT 0530
60 RETURN                                                              OUT 0540
   END                                                                 OUT 0550
```

ØUTPUT Flow Diagram

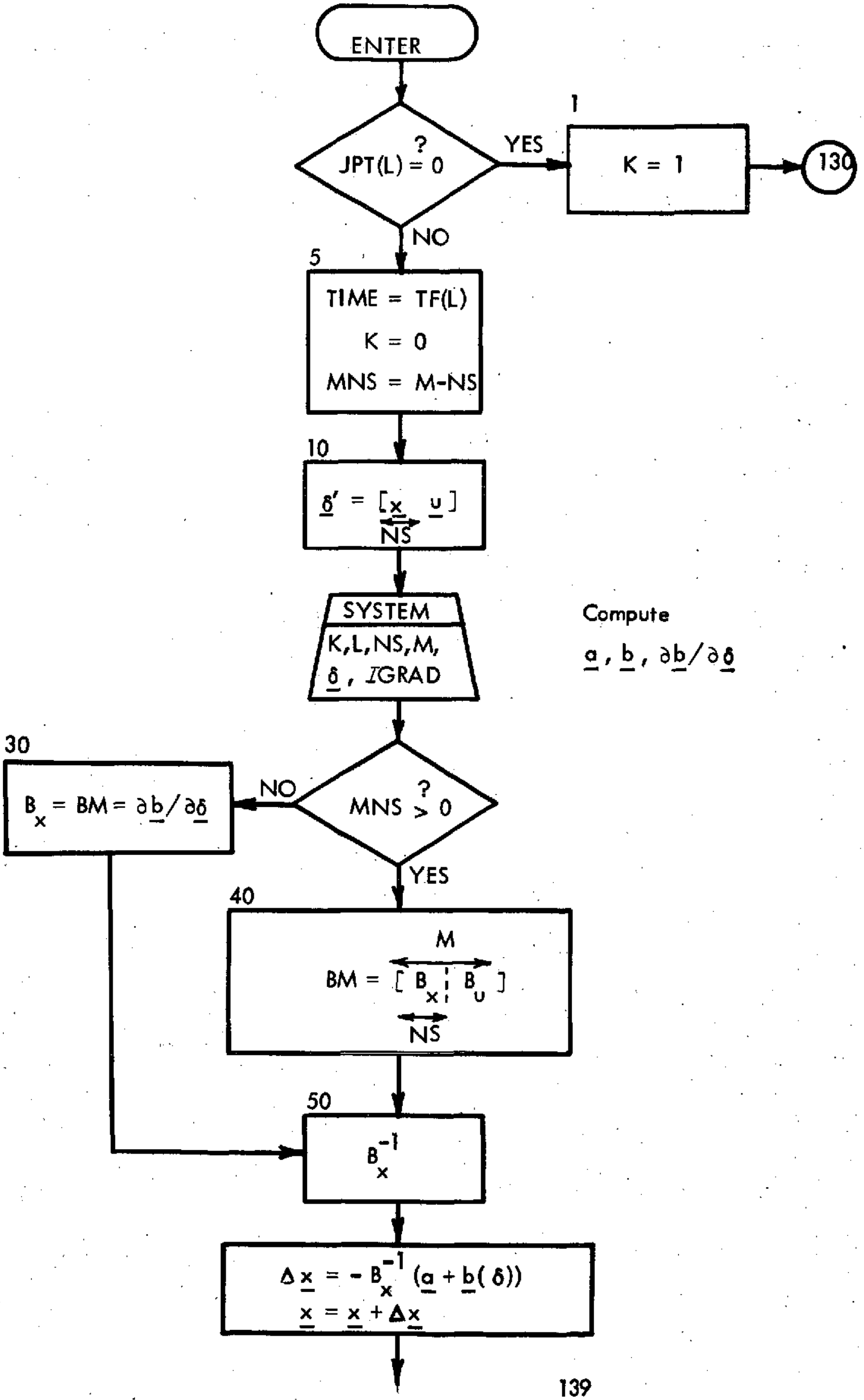


```
C     ----------------------------------------------------------------------
C
C     SUBROUTINE  GRAD1(K,L,TIME,DELTA,LAMDA,IGRAD,EPS,STEP)
C
C     ----------------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE FOR COMPUTING THE DEFLECTION ANGLES USING THE
C     -------  1ST ORDER GRADIENT METHOD.
C
C     INPUTS      K       = NO. OF ITERATIONS.
C     ------      L       = NO. OF THE TRAJECTORY POINT.
C                 TIME    = FLIGHT TIME OF THE TRAJECTORY POINT.
C                 DELTA   = INITIAL GUESS OF TRIM ANGLES.
C                 LAMDA   = INITIAL GUESS OF LAGRANGE MULTPLIERS.
C                 IGRAD   = 1
C                 EPS     = CONVERGENCE BOUND.
C                 STEP    = STEP SIZE.
C
C     OUTPUTS     DELTA   = VECTOR OF TRIM ANGLES.
C     -------     LAMDA   = VECTOR OF LAGRANGE MULTIPLIERS.
C
C     SUBROUTINES CALLED     SYSTEM , COST , MCPY , CCUT , MINV , GMPRD .
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  GRAD1(K,L,TIME,DELTA,LAMDA,IGRAD,EPS,STEP)
C
C
 1000 FORMAT(//5X,75H** WARNING **  1ST ORDER GRADIENT ALGORITHM USED TH
     1E MAX. NO. OF ITERATIONS,I4 /20X,6HNORM =,E10.3,10X,5HEPS =,E10.3)
C
C     . . . . . . . . .     TYPE AND STORAGE ALLOCATION     . . . . . . . . . .
      REAL   LAMDA , NORM
      DIMENSION   DELTA(10), LAMDA(6), BX(60), BU(60), RX(10), RU(10),
     1            X(10), DX(10), DU(10), LB(10), MB(10)
      COMMON /ARRAY/ AV(6), BV(6), BM(60), BT(6,60), RS, RV(10), RM(100)
      COMMON /CON/ M , NS , KMAX , EPS0 , MPT
      COMMON /TRAJ/  JPT(12) , TF(12)
C
C
C *** TEST WHETHER THIS TRAJECTORY POINT IS TO BE USED
      IF(JPT(L)) 5,1,5
    1 K = -1
      GO TO 130
C
C *** COMPUTE THE TIME OF FLIGHT
```

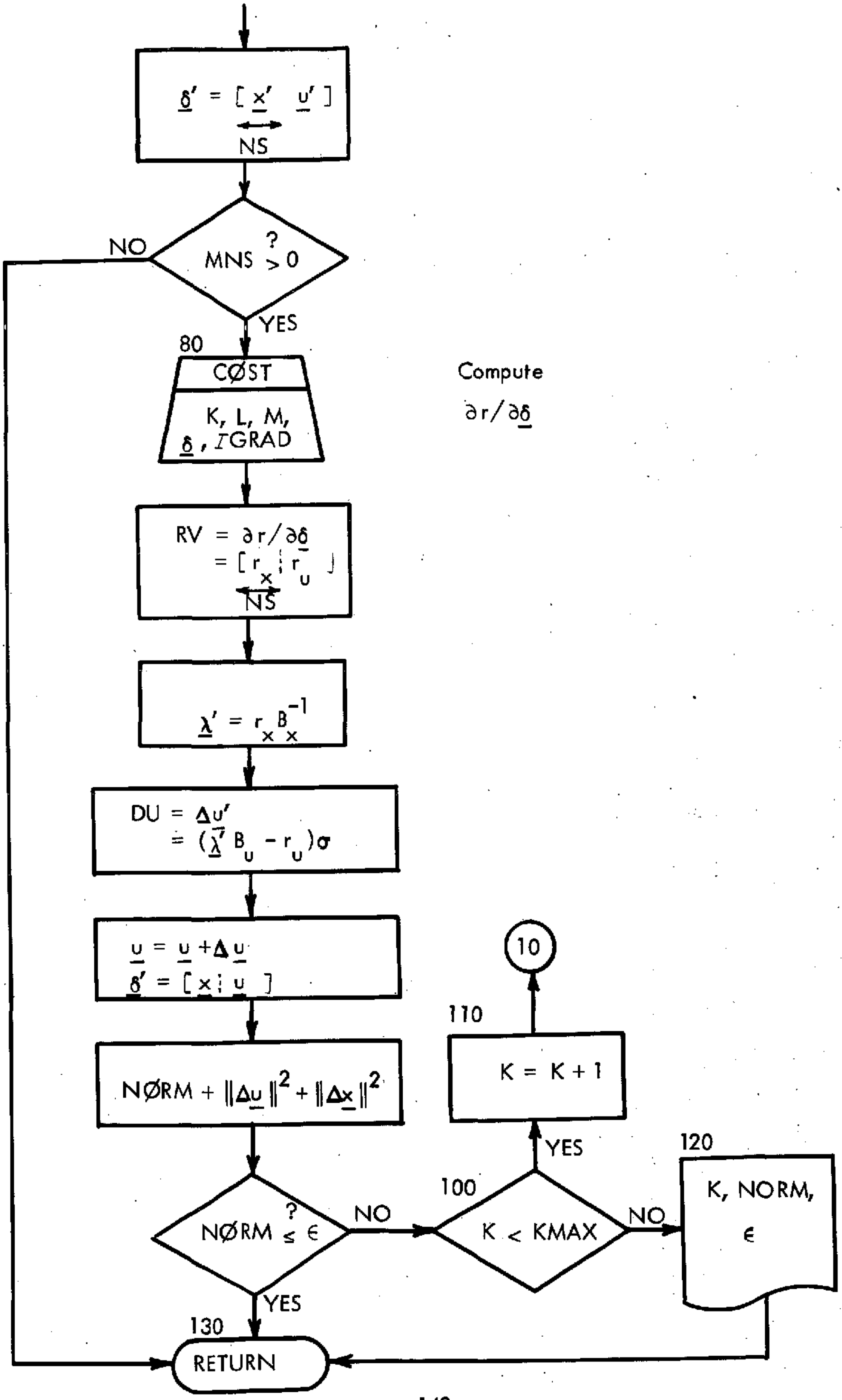
```
    5 TIME = TF(L)
C
C *** START INITIAL ITERATION
      K = 0
      MNS = M - NS
C
   10 CONTINUE
C
C *** COMPUTE GRADIENT TERMS CORRESPONDING TO SYSTEM DYNAMICS
      CALL  SYSTEM(K,L,NS,M,DELTA,IGRAD)
C
C *** PARTITION THE MATRIX  BM  INTO MATRICES  BX  AND  BU
      IF(MNS) 30,30,40
   30 CALL  MCPY(BM,BX,NS,M,0)
      GO TO 50
   40 J = NS + 1
      CALL  CCUT(BM,J,BX,BU,NS,M,0)
C
C *** COMPUTE THE INVERSE OF THE MATRIX  BX
   50 CALL  MINV(BX,NS,DET,LB,MB)
C
C *** COMPUTE VECTOR  X
      DO 60 I=1,NS
   60 DU(I) = - AV(I) - BV(I)
      CALL  GMPRD(BX,DU,DX,NS,NS,1)
      DO 65 I=1,NS
   65 X(I) = X(I) + DX(I)
C
C *** COMPUTE GRADIENT TERMS CORRESPONDING TO PERFORMANCE CRITERION
      DO 70 I=1,NS
   70 DELTA(I) = X(I)
      IF(MNS) 130,130,80
   80 CALL  COST(K,L,M,DELTA,IGRAD)
C
C *** PARTITION THE VECTOR  RV  INTO VECTORS  RX  AND  RU
      J = NS + 1
      CALL  CCUT(RV,J,RX,RU,1,M,0)
C
C *** COMPUTE THE VECTOR  LAMDA
      CALL  GMPRD(RX,BX,LAMDA,1,NS,NS)
C
C *** COMPUTE THE NEW ESTIMATE OF DELTA
      CALL  GMPRD(LAMDA,BU,DU,1,NS,MNS)
      NORM = 0.
      DO 90 I=1,MNS
      DU(I) = ( DU(I) - RU(I) ) * STEP
      NORM = NORM + DU(I)**2
   90 DELTA(NS+I) = DELTA(NS+I) + DU(I)
```

```
      DO 95 I=1,NS
   95 NORM = NORM + DX(I)**2
C
C *** TEST IF THE NEW ESTIMATES ARE SUFFICIENTLY ACCURATE
      IF(NORM-EPS) 130,130,100
C
C *** CHECK FOR EXCESSIVE NUMBER OF ITERATIONS
  100 IF(K-KMAX) 110,120,120
C
C *** PERFORM ANOTHER ITERATION
  110 K = K + 1
      GO TO 10
C
  120 WRITE(6,1000)  K, NORM, EPS
  130 RETURN
      END
```

GRAD1 Flow Diagram



GRAD 1 Flow Diagram (Continued)



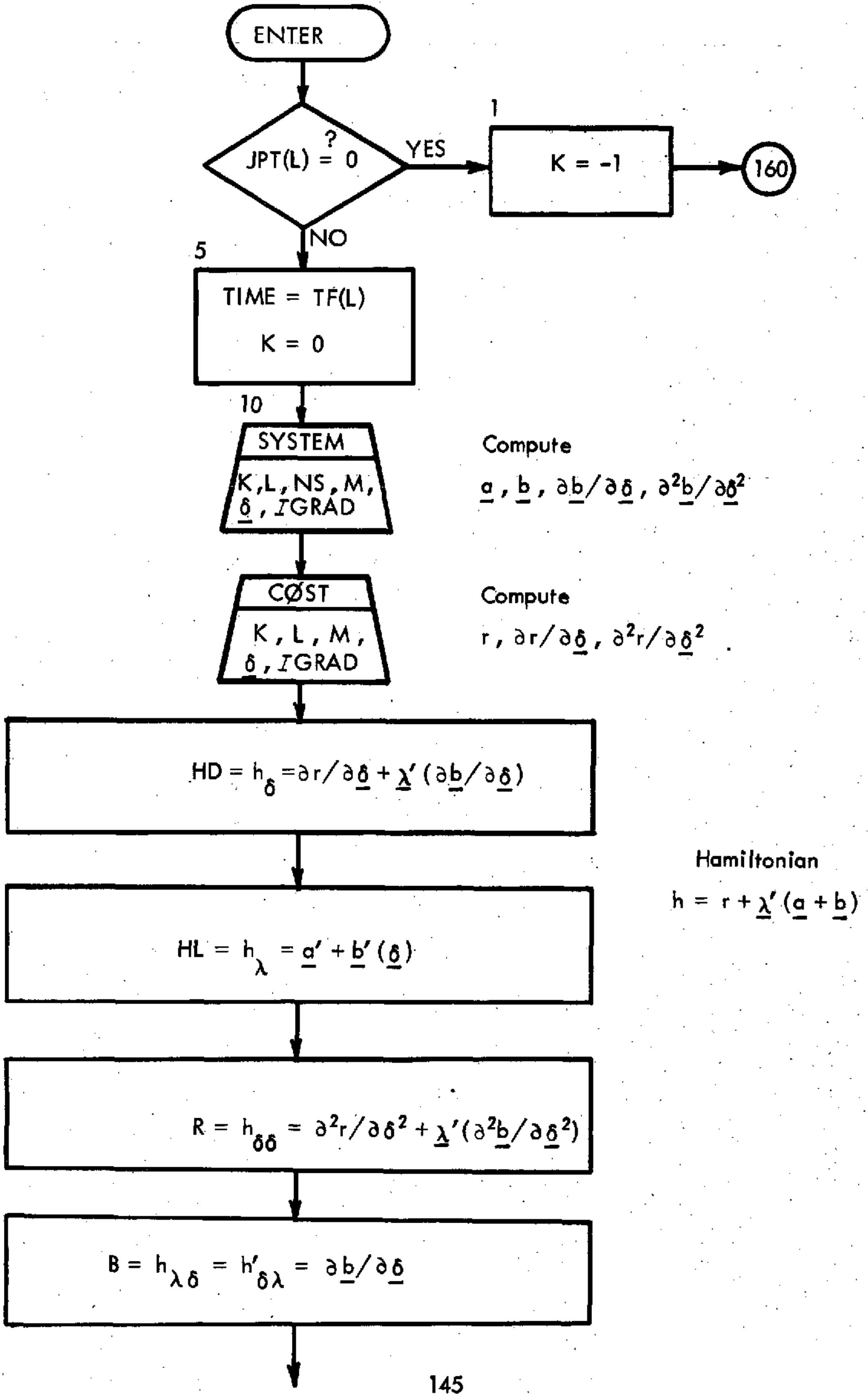
```
C     ----------------------------------------------------------------
C
C     SUBROUTINE  GRAD2(K,L,TIME,DELTA,LAMDA,IGRAD,EPS)
C
C     ----------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE FOR COMPUTING THE DEFLECTION ANGLES USING THE
C     -------  2ND ORDER GRADIENT METHOD.
C
C     INPUTS      K       = NO. OF ITERATIONS.
C     ------      L       = NO. OF THE TRAJECTORY POINT.
C                 TIME    = FLIGHT TIME OF THE TRAJECTORY POINT.
C                 DELTA   = INITIAL GUESS OF TRIM ANGLES.
C                 LAMDA   = INITIAL GUESS OF LAGRANGE MULTPLIERS.
C                 IGRAD   = 2
C                 EPS     = CONVERGENCE BOUND.
C
C     OUTPUTS     DELTA   = VECTOR OF TRIM ANGLES.
C     -------     LAMDA   = VECTOR OF LAGRANGE MULTIPLIERS.
C
C     SUBROUTINES CALLED    SYSTEM , COST , SINV , MXOUT , MPRD ,
C     ----------- ------    GMTRA , GMPRD , GMSYMM , MSTR , GMSUB .
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  GRAD2(K,L,TIME,DELTA,LAMDA,IGRAD,EPS)
C
C
 1000 FORMAT(//5X,55H** ERROR  **  MATRIX R IS NOT POSITIVE DEFINITE
     1 K =,I3,5X,5HEPS =,E12.3 /)
 1010 FORMAT(//5X,65H** WARNING **  LOSS OF SIGNIFICANCE IN INVERTING MA
     1TRIX R      K =,I3,5X,5HEPS =,E12.3 /)
 1020 FORMAT(//5X,57H** ERROR  **  MATRIX BRB IS NOT POSITIVE DEFINITE
     1    K =,I3,5X,5HEPS =,E12.3 /)
 1030 FORMAT(//5X,67H** WARNING **  LOSS OF SIGNIFICANCE IN INVERTING MA
     1TRIX BRB     K =,I3,5X,5HEPS =,E12.3 /)
 1040 FORMAT(//5X,68H** WARNING **  2ND ORDER GRADIENT METHOD USED MAX.
     1NO. OF ITERATIONS,I3,5X,5HEPS =,E12.5,5X,6HNORM =,E12.5 /)
 1050 FORMAT(/10X,9HMATRIX  R )
 1060 FORMAT(/10X,20HMATRIX  R  (INVERSE))
 1070 FORMAT(/10X,9HMATRIX  B )
 1080 FORMAT(/10X,11HMATRIX  BRB )
C
      REAL   LAMDA , LAM , NORM
      DIMENSION   DELTA(10), LAMDA(6), DEL(10), LAM(6),  HD(10), HL(10),
     1            R(100), B(60), BR(60), BRB(36), D(60), X(10), Y(10)
      COMMON /ARRAY/ AV(6), BV(6), BM(60), BT(6,60), RS, RV(10), RM(100)
```

```
      COMMON /CON/ M , NS , KMAX , EPS0 , MPT                            TWO 0490
      COMMON /TRAJ/  JPT(12) , TF(12)                                    TWO 0500
      EQUIVALENCE   (B(1),BM(1))                                         TWO 0510
C                                                                        TWO 0520
C                                                                        TWO 0530
C *** TEST WHETHER THIS TRAJECTORY POINT IS TO BE USED                   TWO 0540
      IF(JPT(L)) 5,1,5                                                   TWO 0550
    1 K = -1                                                             TWO 0560
      GO TO 160                                                          TWO 0570
C                                                                        TWO 0580
C *** COMPUTE THE TIME OF FLIGHT                                         TWO 0590
    5 TIME = TF(L)                                                       TWO 0600
C                                                                        TWO 0610
      K = 0                                                              TWO 0620
C                                                                        TWO*0621
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - TWO*0622
C *** OPTION FOR DISREGARDING AILERON                                    TWO*0623
      IF(JPT(L)+2) 10,6,10                                               TWO*0624
    6 IGRAD = - IGRAD                                                    TWO*0625
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - TWO*0626
C                                                                        TWO*0627
   10 CALL  SYSTEM(K,L,NS,M,DELTA,IGRAD)                                 TWO 0630
      CALL  COST(K,L,M,DELTA,IGRAD)                                      TWO 0640
C                                                                        TWO 0650
C *** COMPUTE THE DERIVATIVE OF THE HAMILTONIAN WITH RESPECT TO DELTA    TWO 0660
      DO 20 I=1,M                                                        TWO 0670
      HD(I) = RV(I)                                                      TWO 0680
      DO 20 J=1,NS                                                       TWO 0690
      JI = J + (I-1)*NS                                                  TWO 0700
   20 HD(I) = HD(I) + LAMDA(J)*BM(JI)                                    TWO 0710
C                                                                        TWO 0720
C *** COMPUTE THE DERIVATIVE OF THE HAMILTONIAN WITH RESPECT TO LAMDA    TWO 0730
      DO 30 J=1,NS                                                       TWO 0740
   30 HL(J) = AV(J) + BV(J)                                              TWO 0750
C                                                                        TWO 0760
C *** COMPUTE THE 2ND DERIVATIVE OF THE HAMILTONIAN   R = HDD            TWO 0770
      M2 = M*(M+1)/2                                                     TWO 0780
      DO 40 I=1,M2                                                       TWO 0790
      R(I) = RM(I)                                                       TWO 0800
      DO 40 J=1,NS                                                       TWO 0810
   40 R(I) = R(I) + LAMDA(J)*BT(J,I)                                     TWO 0820
C                                                                        TWO 0830
C *** COMPUTE THE 2ND DERIVATIVE OF THE HAMILTONIAN   B = HLD            TWO 0840
C                                                                        TWO 0850
C       ( SEE EQUIVALENCE STATEMENT )                                    TWO 0860
C                                                                        TWO 0870
C *** COMPUTE INVERSE OF MATRIX  R                                       TWO 0880
      CALL  SINV(R,M,EPS0,IER)                                           TWO 0890
```

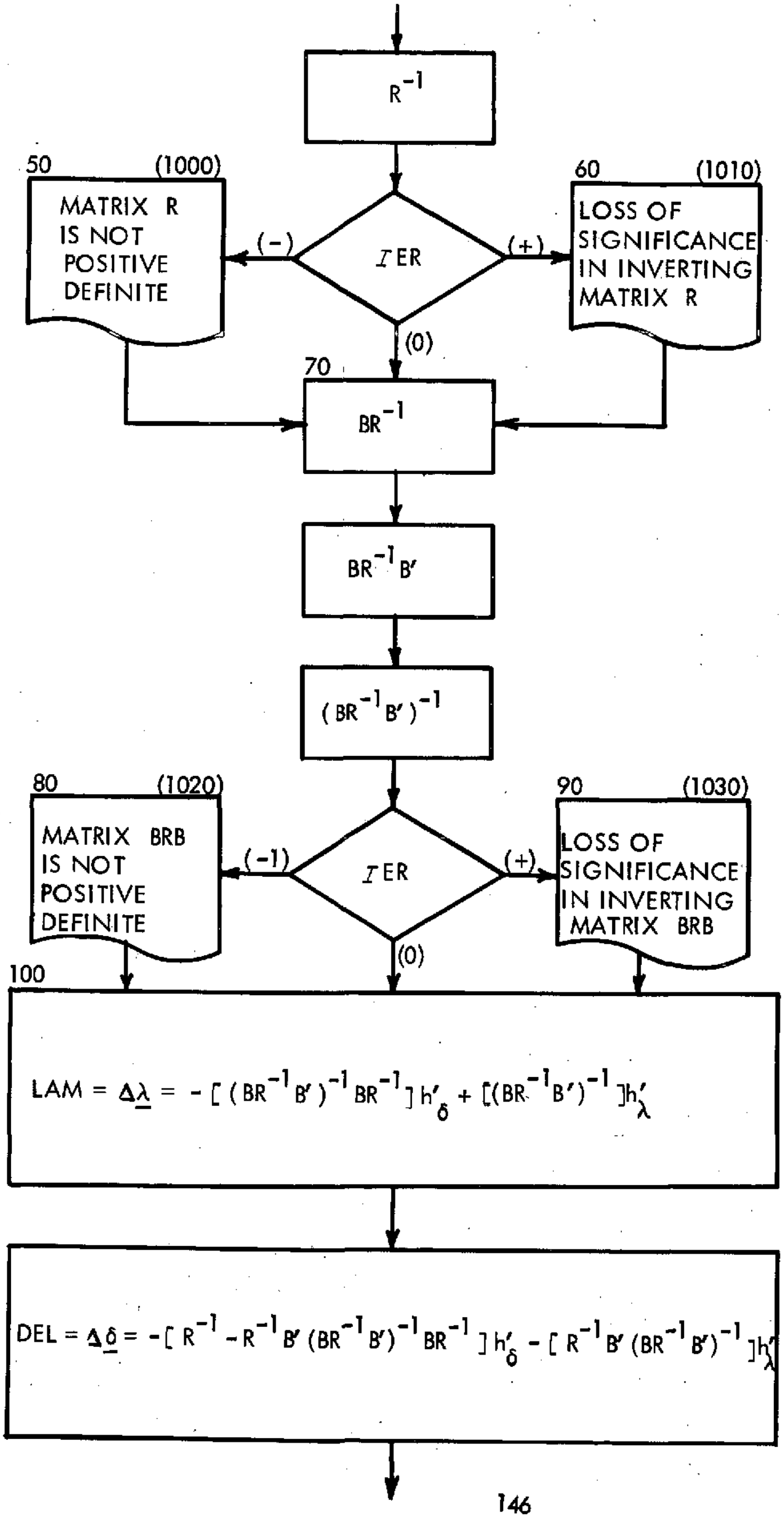
```
      IF(IER) 50,70,60
   50 WRITE(6,1000)  K , EPS0
      WRITE(6,1050)
      CALL  MXOUT(1,R,M,M,1,60,132,1)
      CALL  EXIT
   60 WRITE(6,1010)  K , EPS0
      WRITE(6,1050)
      CALL  MXOUT(1,R,M,M,1,60,132,1)
C
C *** COMPUTE MATRIX  BR
   70 CALL  MPRD(B,R,BR,NS,M,0,1,M)
C
C
C *** COMPUTE MATRIX  BRB
      CALL  GMTRA(B,D,NS,M)
      CALL  GMPRD(BR,D,BRB,NS,M,NS)
      CALL  GMSYMM(BRB,D,NS)
      CALL  MSTR(D,BRB,NS,0,1)
C
C *** COMPUTE INVERSE OF MATRIX  BRB
      CALL  SINV(BRB,NS,EPS0,IER)
      IF(IER) 80,100,90
   80 WRITE(6,1020)  K , EPS0
      WRITE(6,1060)
      CALL  MXOUT(1,R,M,M,1,60,132,1)
      WRITE(6,1070)
      CALL  MXOUT(1,B,NS,M,0,60,132,1)
      WRITE(6,1080)
      CALL  MXOUT(1,BRB,NS,NS,1,60,132,1)
      CALL  EXIT
   90 WRITE(6,1030)  K , EPS0
      WRITE(6,1060)
      CALL  MXOUT(1,R,M,M,1,60,132,1)
      WRITE(6,1070)
      CALL  MXOUT(1,B,NS,M,0,60,132,1)
      WRITE(6,1080)
      CALL  MXOUT(1,BRB,NS,NS,1,60,132,1)
C
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
C *** OPTION FOR DISREGARDING 1ST TRIM EQUALITY CONSTRAINT  --
C *** EQUATION REQUIRING ZERO NET FORCE IN Y-DIRECTION
  100 IF(JPT(L)) 95,96,96
   95 AV(1) = BRB(2)*AV(2) + BRB(4)*AV(3)
      HL(1) = BV(1) - AV(1)/BRB(1)
   96 CONTINUE
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
C
C *** COMPUTE CORRECTION TO LAMDA
```

```
      CALL  GMPRD(BR,HD,X,NS,M,1)                                       TWO 1290
      CALL  MPRD(BRB,X,Y,NS,NS,1,0,1)                                   TWO 1300
      CALL  MPRD(BRB,HL,X,NS,NS,1,0,1)                                  TWO 1310
      CALL  GMSUB(X,Y,LAM,NS,1)                                         TWO 1320
C                                                                       TWO 1330
C *** COMPUTE CORRECTION TO DELTA                                       TWO 1340
      CALL  GMPRD(Y,BR,DEL,1,NS,M)                                      TWO 1350
      CALL  MPRD(R,HD,Y,M,M,1,0,1)                                      TWO 1360
      CALL  GMSUB(DEL,Y,DEL,M,1)                                        TWO 1370
      CALL  GMPRD(X,BR,Y,1,NS,M)                                        TWO 1380
      CALL  GMSUB(DEL,Y,DEL,M,1)                                        TWO 1390
C                                                                       TWO 1400
C *** COMPUTE NEW ESTIMATE OF DELTA                                     TWO 1410
      NORM = 0.                                                         TWO 1420
      DO 110 I=1,M                                                      TWO 1430
      NORM = NORM + DEL(I)**2                                           TWO 1440
  110 DELTA(I) = DELTA(I) + DEL(I)                                      TWO 1450
C                                                                       TWO 1460
C *** COMPUTE NEW ESTIMATE OF LAMDA                                     TWO 1470
      DO 120 J=1,NS                                                     TWO 1480
      NORM = NORM + LAM(J)**2                                           TWO 1490
  120 LAMDA(J) = LAMDA(J) + LAM(J)                                      TWO 1500
      IF(NORM-EPS) 160,160,130                                          TWO 1510
  130 IF(K-KMAX)    140,150,150                                         TWO 1520
  140 K = K + 1                                                         TWO 1530
      GO TO 10                                                          TWO 1540
  150 WRITE(6,1040)  K , EPS0 , NORM                                    TWO 1550
  160 RETURN                                                            TWO 1560
      END                                                               TWO 1570
```

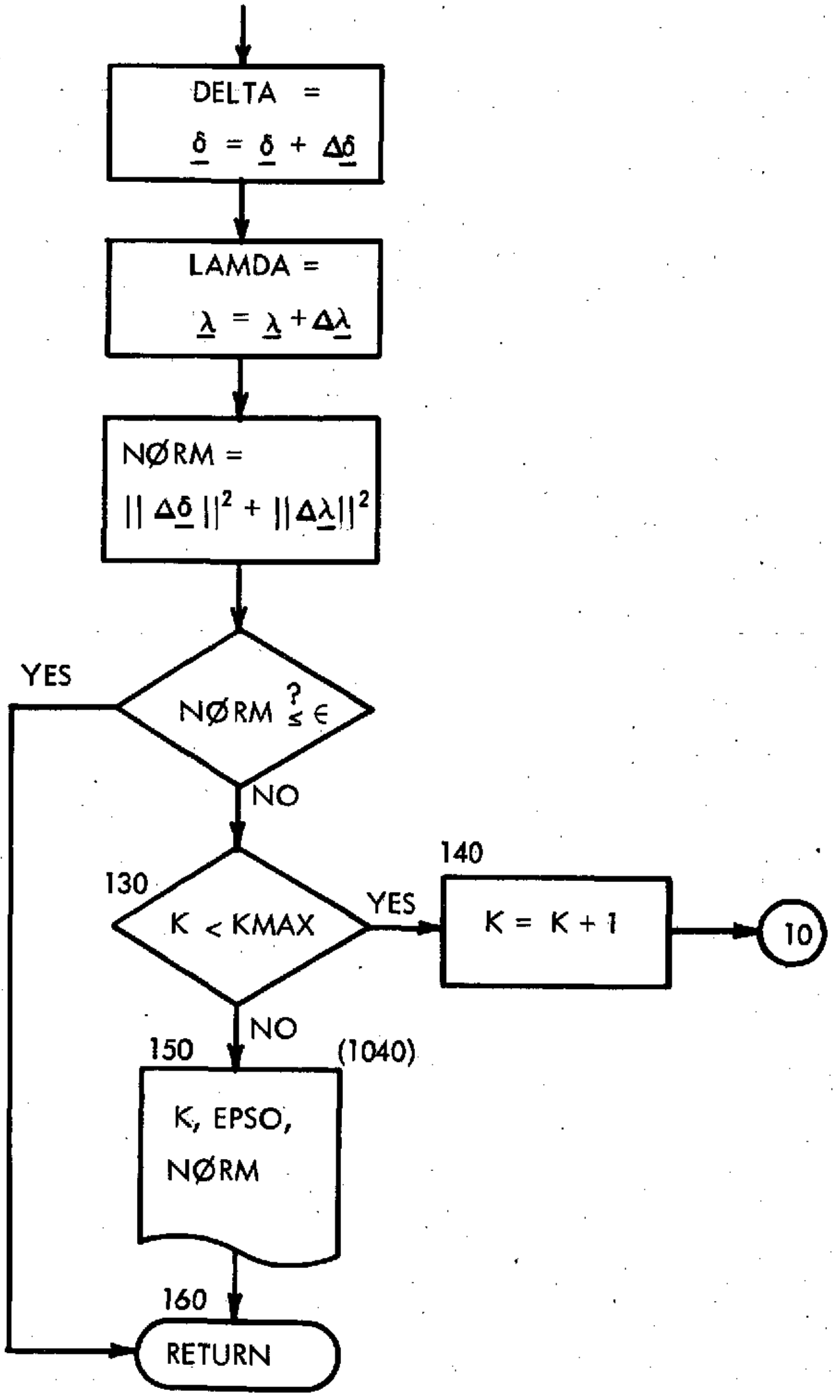
GRAD 2 Flow Diagram



GRAD2 Flow Diagram (Continued)



GRAD2 Flow Diagram (Continued)



```
C     ----------------------------------------------------------------
C
C     SUBROUTINE  SYSTEM(K,L,NS,M,DELTA,IGRAD)
C
C     ----------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE FOR COMPUTING THE COEFFICIENTS IN THE
C     -------  EQUATIONS OF THE LATERAL DYNAMICS DEFINING TRIM.
C              ALSO EVALUATES THE CORRESPONDING DERIVATIVES REQUIRED
C              BY THE GRADIENT METHODS.
C
C     INPUTS     K      = NO. OF ITERATIONS.
C     ------     L      = NO. OF THE TRAJECTORY POINT.
C                NS     = NO. OF TRIM EQUATIONS.
C                M      = NO. OF TRIM ANGLES.
C                DELTA  = VECTOR OF TRIM ANGLES.
C                IGRAD  = ORDER OF GRADIENT METHOD TO BE USED.
C
C     SUBROUTINES CALLED    NONE
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  SYSTEM(K,L,NS,M,DELTA,IGRAD)
C
C
      DIMENSION  DELTA(1)
      COMMON /ARRAY/ AV(6), BV(6), BM(60), BT(6,60), RS, RV(10), RM(100)
      COMMON /SYST/  YBT, RBT,     X1, X2, X3, X4, X5, XMRP,
     1               S, BREF,      Y1, Y2, Y3, Y4, Y5, YMRP,
     2                             Z1, Z2, Z3, Z4, Z5, ZMRP,
     3      XCG(12),      ZCG(12),       Q(12),       V(12),      VY(12),
     4        F(12),     FSRM(12),     CYB(12),     CLB(12),     CNB(12),
     5     DCYB(12),     DCLB(12),   DCNBA(12),   DCNBF(12),     CYA(12),
     6      CLA(12),      CNA(12),     CYR(12),     CLR(12),     CNR(12)
      DATA  RAD  / 57.2957795 /
C
C
      IF(K) 300,100,300
C *** COMPUTE VECTOR  A
  100 CONTINUE
      QS   = Q(L) * S
      QSB  = QS * BREF
      BETA    = ARSIN(VY(L)/V(L))
C
      CYBCG   = CYB(L)  +(DCYB(L))*RAD
      CLBCG   = CLB(L)  +(DCLB(L))*RAD
```

```
      CNBCG  = CNB(L) +(DCNBA(L) + DCNBF(L))*RAD
C
      CLBCG  =  CLBCG + CYBCG*(ZCG(L)-ZMRP)/BREF
      CNBCG  =  CNBCG - CYBCG*(XCG(L)-XMRP)/BREF
C
      AV(1)  = QS  * CYBCG * BETA
      AV(2)  = QSB * CLBCG * BETA  +  RBT
      AV(3)  = QSB * CNBCG * BETA  +  YBT
C
C *** COMPUTE COEFFICIENTS IN VECTOR  B
C
      RAD = 57.2957795
      C1 = COS(18./RAD)
      S1 = SIN(18./RAD)
      C2 = COS(12./RAD)
      S2 = SIN(12./RAD)
      C3 = COS(3.5/RAD)
      S3 = SIN(3.5/RAD)
      C4 = COS(15./RAD)
      S4 = SIN(15./RAD)
C
      CLACG  = CLA(L) + CYA(L)*(ZCG(L)-ZMRP)/BREF
      CNACG  = CNA(L) - CYA(L)*(XCG(L)-XMRP)/BREF
      CLRCG  = CLR(L) + CYR(L)*(ZCG(L)-ZMRP)/BREF
      CNRCG  = CNR(L) - CYR(L)*(XCG(L)-XMRP)/BREF
C
      BM(1)  = F(L) * C1
      BM(2)  =-F(L) * C1 * (Z1 - ZCG(L))
      BM(3)  = F(L) * C1 * (X1 - XCG(L))
      BM(4)  = 2. * F(L) * C2 * C3
      BM(5)  =-2. * F(L) * C2 * C3 * (Z2 - ZCG(L))
      BM(6)  = 2. * F(L) * ((X2-XCG(L))*C3 - Y2*S3) * C2
      BM(7)  = 2. * F(L) * S2 * S3
      BM(8)  = 2. * F(L) *  (Y2*C2 - (Z2-ZCG(L))*S2*S3)
      BM(9)  = 2. * F(L) * ((Y2*C3 + (X2-XCG(L))*S3) * S2)
      BM(10) = 2. * FSRM(L) * C4
      BM(11) =-2. * FSRM(L) * C4 * (Z4 - ZCG(L))
      BM(12) = 2. * FSRM(L) * ((X4-XCG(L))*C4 - Y4*S4)
      BM(13) = 0.
      BM(14) = 2. * FSRM(L) * Y4
      BM(15) = 0.
      BM(16) = QS  * CYA(L)
      BM(17) = QSB * CLACG
      BM(18) = QSB * CNACG
      BM(19) = QS  * CYR(L)
      BM(20) = QSB * CLRCG
      BM(21) = QSB * CNRCG
C
```

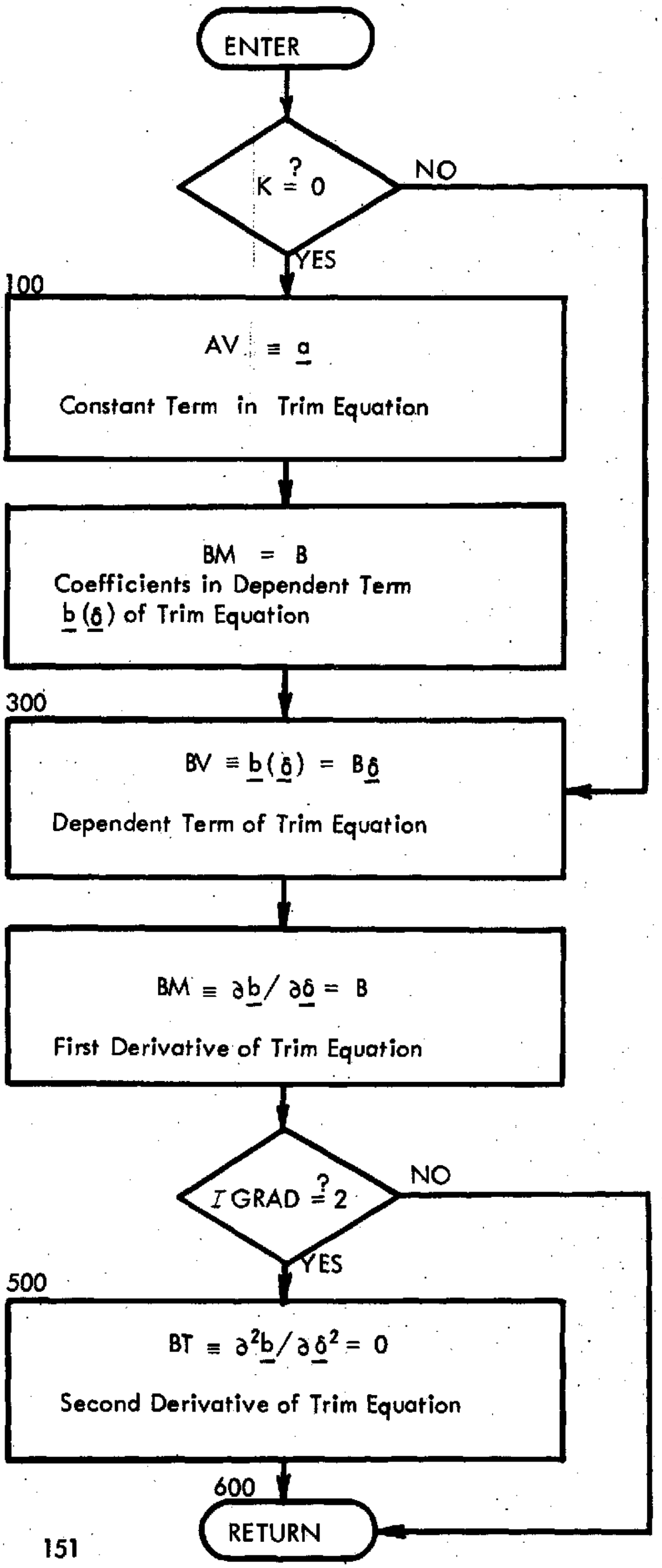
```
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  SYS*0951
C *** OPTION FOR DISREGARDING AILERON                                        SYS*0952
      IF(IGRAD) 290,300,300                                                  SYS*0953
  290 BM(16) = 0.                                                            SYS*0954
      BM(17) = 0.                                                            SYS*0955
      BM(18) = 0.                                                            SYS*0956
      IGRAD = - IGRAD                                                        SYS*0957
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  SYS*0958
C                                                                            SYS*0959
C *** COMPUTE VECTOR  B                                                      SYS 0960
  300 CONTINUE                                                               SYS 0970
      DO 310 I=1,NS                                                          SYS 0980
      BV(I) = 0.                                                             SYS 0990
      DO 310 J=1,M                                                           SYS 1000
      IJ = I + (J-1)*NS                                                      SYS 1010
  310 BV(I) = BV(I) + BM(IJ)*DELTA(J)                                        SYS 1020
C                                                                            SYS 1030
C *** COMPUTE THE 1ST DERIVATIVE OF VECTOR  B                                SYS 1040
C                                                                            SYS 1050
C         (--- CONSTANT MATRIX COMPUTED ABOVE ---)                           SYS 1060
C                                                                            SYS 1070
C                                                                            SYS 1080
      IF(IGRAD-2) 600,500,600                                                SYS 1090
C                                                                            SYS 1100
C *** COMPUTE THE 2ND DERIVATIVE OF VECTOR  B                                SYS 1110
  500 CONTINUE                                                               SYS 1120
      M2 = M*(M+1)/2                                                         SYS 1130
      DO 510 J=1,NS                                                          SYS 1140
      DO 510 I=1,M2                                                          SYS 1150
  510 BT(J,I) = 0.                                                           SYS 1160
  600 RETURN                                                                 SYS 1170
      END                                                                    SYS 1180
```

SYSTEM Flow Diagram

Trim Equation

$\underline{a} + \underline{b}(\underline{\delta}) = 0$

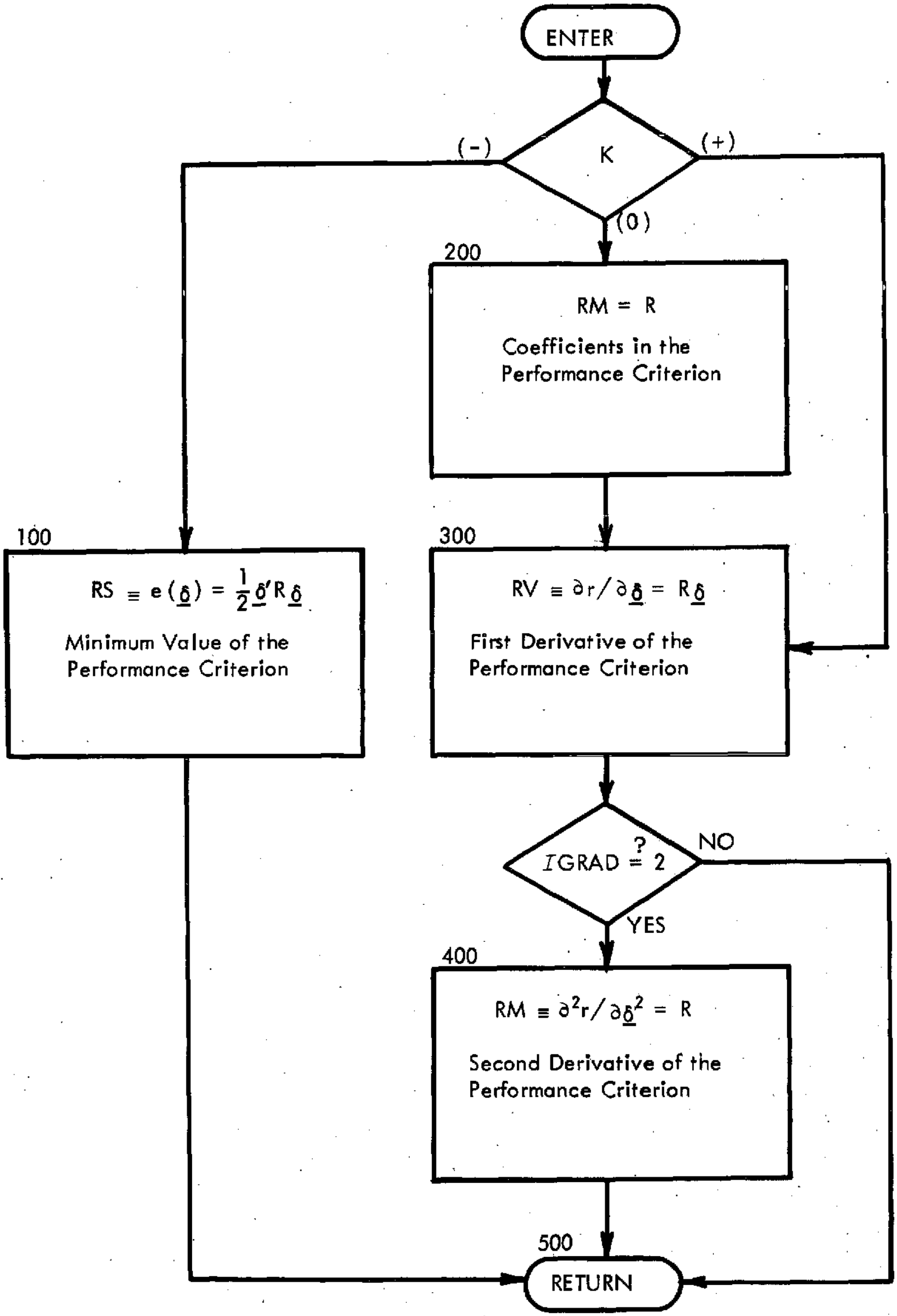
$\underline{b}(\underline{\delta}) = B\underline{\delta}$



```
C     ------------------------------------------------------------------
C
C     SUBROUTINE  COST(K,L,M,DELTA,IGRAD)
C
C     ------------------------------------------------------------------
C
C     PURPOSE  SUBROUTINE FOR COMPUTING THE COEFFICIENTS IN THE
C     -------  PERFORMANCE CRITERION.
C              ALSO EVALUATES THE CORRESPONDING DERIVATIVES REQUIRED
C              BY THE GRADIENT METHODS.
C
C     INPUTS      K       = NO. OF ITERATIONS.
C     ------      L       = NO. OF THE TRAJECTORY POINT.
C                 M       = NO. OF TRIM ANGLES.
C                 DELTA   = VECTOR OF TRIM ANGLES.
C                 IGRAD   = ORDER OF GRADIENT METHOD TO BE USED.
C
C     SUBROUTINES CALLED    NONE
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE  COST(K,L,M,DELTA,IGRAD)
C
C
      DIMENSION  C(7) , DELTA(1)
      COMMON /ARRAY/ AV(6), BV(6), BM(60), BT(6,60), RS, RV(10), RM(100)
      COMMON /PERF/  W1(7), W2(7), DAMAX(12),DRMAX(12), Q(12),
     1               DMAX, SA, SR, CDA, CDR
C
C
      IF(K) 100,200,300
C
C *** COMPUTE MINIMUM VALUE OF THE PERFORMANCE CRITERION
  100 CONTINUE
      RS = 0.
      DO 110 I=1,M
  110 RS = RS + C(I) * DELTA(I)**2
      RS = RS / 2.
      GO TO 500
C
C *** COMPUTE THE COEFFICIENTS IN THE PERFORMANCE CRITERION
  200 CONTINUE
      DO 210 I=1,5
  210 C(I) = (W1(I)/DMAX)**2 + W2(I)**2
      C(6) = (W1(6)/DAMAX(L))**2 + (W2(6)*Q(L)*SA*CDA)**2
      C(7) = (W1(7)/DRMAX(L))**2 + (W2(7)*Q(L)*SR*CDR)**2
```

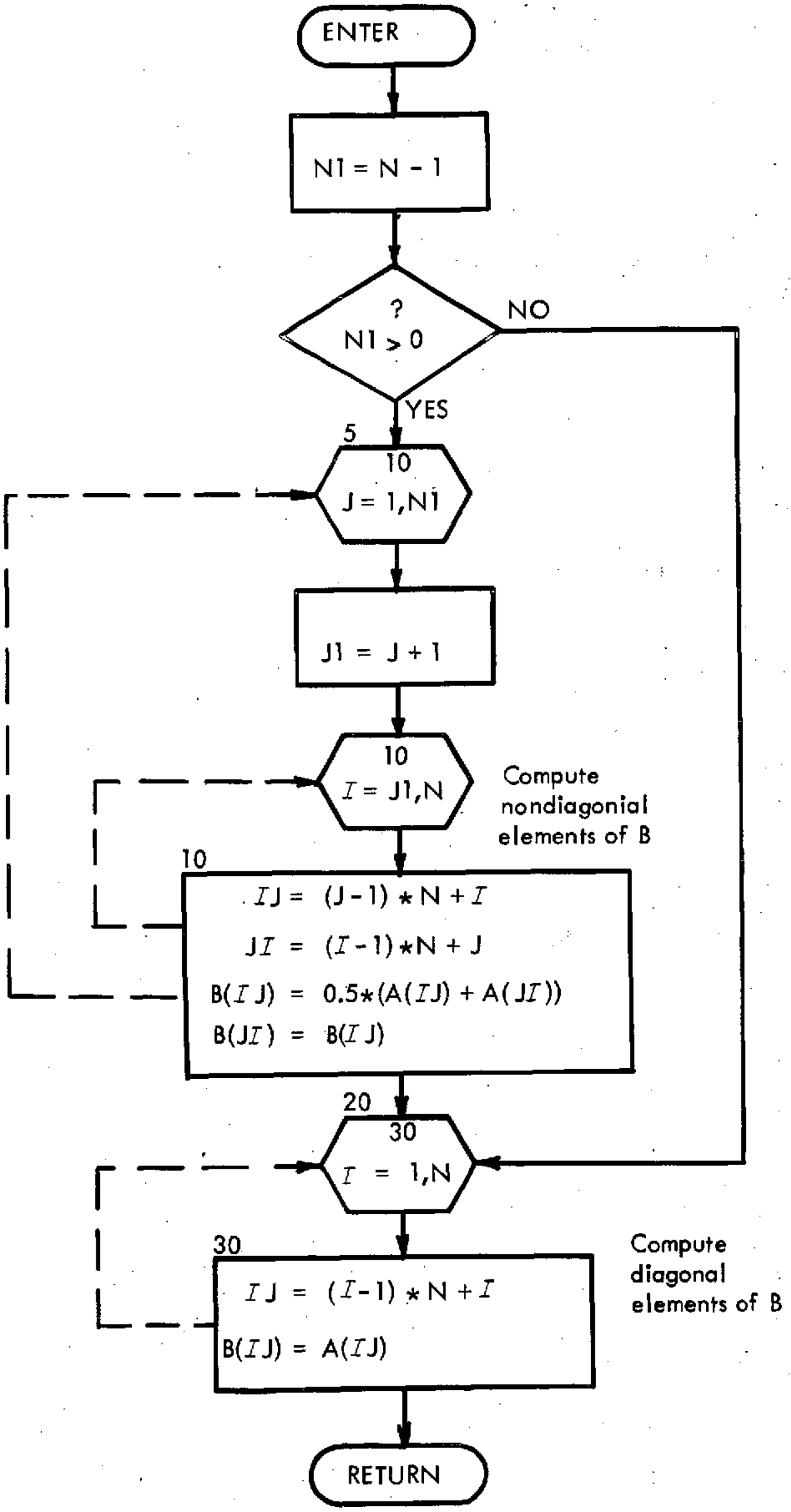
```
      DO 212 I=1,M                                                      CST 0481
      IF(W1(I)) 211,212,212                                             CST 0482
  211 C(I) = 1. / W1(I)**2                                              CST 0483
  212 CONTINUE                                                          CST 0484
      M2 = M*(M+1)/2                                                    CST 0490
      DO 220 I=1,M2                                                     CST 0500
  220 RM(I) = 0.                                                        CST 0510
      DO 230 I=1,M                                                      CST 0520
      II = I*(I+1)/2                                                    CST 0530
  230 RM(II) = C(I)                                                     CST 0540
C                                                                       CST 0550
C *** COMPUTE THE 1ST DERIVATIVE OF THE PERFORMANCE CRITERION           CST 0560
  300 CONTINUE                                                          CST 0570
      DO 310 I=1,M                                                      CST 0580
  310 RV(I) = C(I) * DELTA(I)                                           CST 0590
C                                                                       CST 0600
      IF(IGRAD-2) 500,400,500                                           CST 0610
C                                                                       CST 0620
C *** COMPUTE THE 2ND DERIVATIVE OF THE PERFORMANCE CRITERION           CST 0630
  400 CONTINUE                                                          CST 0640
C                                                                       CST 0650
C         (--- CONSTANT MATRIX COMPUTED ABOVE ---)                      CST 0660
C                                                                       CST 0670
  500 RETURN                                                            CST 0680
      END                                                               CST 0690
```

CØST Flow Diagram



```
C     --------------------------------------------------------------------
C
      SUBROUTINE  GMSYMM(A,B,N)
C
C     --------------------------------------------------------------------
C
C     PURPOSE   COMPUTES A SYMMETRIC MATRIX  B  FROM A SQUARE MATRIX  A
C     -------   ACCORDING TO
C                                B = ( A + A') / 2
C
C     INPUTS       A       = SQUARE MATRIX  (STORAGE MODE = 0).
C     ------       N       = NO. OF ROWS AND COLS. IN A AND B.
C
C     OUTPUTS      B       = SYMMETRIC MATRIX FORMED FROM A  (STORAGE
C     -------                MODE = 0).
C
C     SUBROUTINES CALLED     NONE
C     ----------- ------
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBROUTINE  GMSYMM(A,B,N)
C
C
      DIMENSION  A(1) , B(1)
C
C
      N1 = N - 1
      IF(N1) 20,20,5
    5 DO 10 J=1,N1
      J1 = J + 1
      DO 10 I=J1,N
      IJ = (J-1)*N + I
      JI = (I-1)*N + J
      B(IJ) = 0.5 * (A(IJ) + A(JI))
   10 B(JI) = B(IJ)
   20 DO 30 I=1,N
      IJ = (I-1)*N + I
   30 B(IJ) = A(IJ)
      RETURN
      END
```

GMSYMM Flow Diagram



```
C
C     ..................................................................
C
C        SUBROUTINE MCPY
C
C        PURPOSE
C           COPY ENTIRE MATRIX
C
C        USAGE
C           CALL MCPY (A,R,N,M,MS)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A OR R
C           M - NUMBER OF COLUMNS IN A OR R
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R)
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           LOC
C
C        METHOD
C           EACH ELEMENT OF MATRIX A IS MOVED TO THE CORRESPONDING
C           ELEMENT OF MATRIX R
C
C     ..................................................................
C
      SUBROUTINE MCPY(A,R,N,M,MS)
      DIMENSION A(1),R(1)
C
C        COMPUTE VECTOR LENGTH, IT
C
      CALL LOC(N,M,IT,N,M,MS)
C
C        COPY MATRIX
C
      DO 1 I=1,IT
    1 R(I)=A(I)
      RETURN
      END
```

```
C                                                                       MSTR  10
C     ..................................................................MSTR  20
C                                                                       MSTR  30
C        SUBROUTINE MSTR                                                MSTR  40
C                                                                       MSTR  50
C        PURPOSE                                                        MSTR  60
C           CHANGE STORAGE MODE OF A MATRIX                             MSTR  70
C                                                                       MSTR  80
C        USAGE                                                          MSTR  90
C           CALL MSTR(A,R,N,MSA,MSR)                                    MSTR 100
C                                                                       MSTR 110
C        DESCRIPTION OF PARAMETERS                                      MSTR 120
C           A - NAME OF INPUT MATRIX                                    MSTR 130
C           R - NAME OF OUTPUT MATRIX                                   MSTR 140
C           N - NUMBER OF ROWS AND COLUMNS IN A AND R                   MSTR 150
C           MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A         MSTR 160
C                  0 - GENERAL                                          MSTR 170
C                  1 - SYMMETRIC                                        MSTR 180
C                  2 - DIAGONAL                                         MSTR 190
C           MSR - SAME AS MSA EXCEPT FOR MATRIX R                       MSTR 200
C                                                                       MSTR 210
C        REMARKS                                                        MSTR 220
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A         MSTR 230
C           MATRIX A MUST BE A SQUARE MATRIX                            MSTR 240
C                                                                       MSTR 250
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MSTR 260
C           LOC                                                         MSTR 270
C                                                                       MSTR 280
C        METHOD                                                         MSTR 290
C           MATRIX A IS RESTRUCTURED TO FORM MATRIX R.                  MSTR 300
C            MSA MSR                                                    MSTR 310
C             0   0  MATRIX A IS MOVED TO MATRIX R                      MSTR 320
C             0   1  THE UPPER TRIANGLE ELEMENTS OF A GENERAL MATRIX    MSTR 330
C                    ARE USED TO FORM A SYMMETRIC MATRIX                MSTR 340
C             0   2  THE DIAGONAL ELEMENTS OF A GENERAL MATRIX ARE USED MSTR 350
C                    TO FORM A DIAGONAL MATRIX                          MSTR 360
C             1   0  A SYMMETRIC MATRIX IS EXPANDED TO FORM A GENERAL   MSTR 370
C                    MATRIX                                             MSTR 380
C             1   1  MATRIX A IS MOVED TO MATRIX R                      MSTR 390
C             1   2  THE DIAGONAL ELEMENTS OF A SYMMETRIC MATRIX ARE    MSTR 400
C                    USED TO FORM A DIAGONAL MATRIX                     MSTR 410
C             2   0  A DIAGONAL MATRIX IS EXPANDED BY INSERTING MISSING MSTR 420
C                    ZERO ELEMENTS TO FORM A GENERAL MATRIX             MSTR 430
C             2   1  A DIAGONAL MATRIX IS EXPANDED BY INSERTING MISSING MSTR 440
C                    ZERO ELEMENTS TO FORM A SYMMETRIC MATRIX           MSTR 450
C             2   2  MATRIX A IS MOVED TO MATRIX R                      MSTR 460
C                                                                       MSTR 470
C     ..................................................................MSTR 480
```

```
C
      SUBROUTINE MSTR(A,R,N,MSA,MSR)
      DIMENSION A(1),R(1)
C
      DO 20 I=1,N
      DO 20 J=1,N
C
C        IF R IS GENERAL, FORM ELEMENT
C
      IF(MSR) 5,10,5
C
C        IF IN LOWER TRIANGLE OF SYMMETRIC OR DIAGONAL R, BYPASS
C
    5 IF(I-J) 10,10,20
   10 CALL LOC(I,J,IR,N,N,MSR)
C
C        IF IN UPPER AND OFF DIAGONAL  OF DIAGONAL R, BYPASS
C
      IF(IR) 20,20,15
C
C        OTHERWISE, FORM R(I,J)
C
   15 R(IR)=0.0
      CALL LOC(I,J,IA,N,N,MSA)
C
C        IF THERE IS NO A(I,J), LEAVE R(I,J) AT 0.0
C
      IF(IA) 20,20,18
   18 R(IR)=A(IA)
   20 CONTINUE
      RETURN
      END
```

```
C
C     ...................................................................
C
C        SUBROUTINE LOC
C
C        PURPOSE
C           COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIX OF
C           SPECIFIED STORAGE MODE
C
C        USAGE
C           CALL LOC (I,J,IR,N,M,MS)
C
C        DESCRIPTION OF PARAMETERS
C           I   - ROW NUMBER OF ELEMENT
C           J   - COLUMN NUMBER  OF ELEMENT
C           IR  - RESULTANT VECTOR SUBSCRIPT
C           N   - NUMBER OF ROWS IN MATRIX
C           M   - NUMBER OF COLUMNS IN MATRIX
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           MS=0   SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*M ELEMENTS
C                  IN STORAGE (GENERAL MATRIX)
C           MS=1   SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*(N+1)/2 IN
C                  STORAGE (UPPER TRIANGLE OF SYMMETRIC MATRIX). IF
C                  ELEMENT IS IN LOWER TRIANGULAR PORTION, SUBSCRIPT IS
C                  CORRESPONDING ELEMENT IN UPPER TRIANGLE.
C           MS=2   SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N ELEMENTS
C                  IN STORAGE (DIAGONAL ELEMENTS OF DIAGONAL MATRIX).
C                  IF ELEMENT IS NOT ON DIAGONAL (AND THEREFORE NOT IN
C                  STORAGE), IR IS SET TO ZERO.
C
C     ...................................................................
C
      SUBROUTINE LOC(I,J,IR,N,M,MS)
C
      IX=I
      JX=J
      IF(MS-1) 10,20,30
   10 IRX=N*(JX-1)+IX
      GO TO 36
   20 IF(IX-JX) 22,24,24
   22 IRX=IX+(JX*JX-JX)/2
      GO TO 36
   24 IRX=JX+(IX*IX-IX)/2
      GO TO 36
   30 IRX=0
      IF(IX-JX) 36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END
```

```
C                                                                       GMSU  10
C     ..................................................................GMSU  20
C                                                                       GMSU  30
C        SUBROUTINE GMSUB                                               GMSU  40
C                                                                       GMSU  50
C        PURPOSE                                                        GMSU  60
C           SUBTRACT ONE GENERAL MATRIX FROM ANOTHER TO FORM RESULTANT  GMSU  70
C           MATRIX                                                      GMSU  80
C                                                                       GMSU  90
C        USAGE                                                          GMSU 100
C           CALL GMSUB(A,B,R,N,M)                                       GMSU 110
C                                                                       GMSU 120
C        DESCRIPTION OF PARAMETERS                                      GMSU 130
C           A - NAME OF FIRST INPUT MATRIX                              GMSU 140
C           B - NAME OF SECOND INPUT MATRIX                             GMSU 150
C           R - NAME OF OUTPUT MATRIX                                   GMSU 160
C           N - NUMBER OF ROWS IN A,B,R                                 GMSU 170
C           M - NUMBER OF COLUMNS IN A,B,R                              GMSU 180
C                                                                       GMSU 190
C        REMARKS                                                        GMSU 200
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES             GMSU 210
C                                                                       GMSU 220
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  GMSU 230
C           NONE                                                        GMSU 240
C                                                                       GMSU 250
C        METHOD                                                         GMSU 260
C           MATRIX B ELEMENTS ARE SUBTRACTED FROM CORRESPONDING MATRIX AGMSU 270
C           ELEMENTS                                                    GMSU 280
C                                                                       GMSU 290
C     ..................................................................GMSU 300
C                                                                       GMSU 310
      SUBROUTINE GMSUB(A,B,R,N,M)                                       GMSU 320
      DIMENSION A(1),B(1),R(1)                                          GMSU 330
C                                                                       GMSU 340
C        CALCULATE NUMBER OF ELEMENTS                                   GMSU 350
C                                                                       GMSU 360
      NM=N*M                                                            GMSU 370
C                                                                       GMSU 380
C        SUBTRACT MATRICES                                              GMSU 390
C                                                                       GMSU 400
      DO 10 I=1,NM                                                      GMSU 410
   10 R(I)=A(I)-B(I)                                                    GMSU 420
      RETURN                                                            GMSU 430
      END                                                               GMSU 440
```

```
C
C     ..................................................................
C
C        SUBROUTINE GMPRD
C
C        PURPOSE
C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
C           MATRIX
C
C        USAGE
C           CALL GMPRD(A,B,R,N,M,L)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF FIRST INPUT MATRIX
C           B - NAME OF SECOND INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A
C           M - NUMBER OF COLUMNS IN A AND ROWS IN B
C           L - NUMBER OF COLUMNS IN B
C
C        REMARKS
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
C           OF MATRIX B
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
C           AND THE RESULT IS STORED IN THE N BY L MATRIX R.
C
C     ..................................................................
C
      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
```

```
C
C     ..................................................................
C
C        SUBROUTINE GMTRA
C
C        PURPOSE
C           TRANSPOSE A GENERAL MATRIX
C
C        USAGE
C           CALL GMTRA(A,R,N,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF MATRIX TO BE TRANSPOSED
C           R - NAME OF RESULTANT MATRIX
C           N - NUMBER OF ROWS OF A AND COLUMNS OF R
C           M - NUMBER OF COLUMNS OF A AND ROWS OF R
C
C        REMARKS
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C           MATRICES A AND R MUST BE STORED AS GENERAL MATRICES
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           TRANSPOSE N BY M MATRIX A TO FORM M BY N MATRIX R
C
C     ..................................................................
C
      SUBROUTINE GMTRA(A,R,N,M)
      DIMENSION A(1),R(1)
C
      IR=0
      DO 10 I=1,N
      IJ=I-N
      DO 10 J=1,M
      IJ=IJ+N
      IR=IR+1
   10 R(IR)=A(IJ)
      RETURN
      END
```

```
      DO 10 I=1,M                                                       GMPR 490
      JI=JI+N                                                           GMPR 500
      IB=IB+1                                                           GMPR 510
   10 R(IR)=R(IR)+A(JI)*B(IB)                                           GMPR 520
      RETURN                                                            GMPR 530
      END                                                               GMPR 540
```

```
C                                                                       MPRD  10
C     ..................................................................MPRD  20
C                                                                       MPRD  30
C        SUBROUTINE MPRD                                                MPRD  40
C                                                                       MPRD  50
C        PURPOSE                                                        MPRD  60
C           MULTIPLY TWO MATRICES TO FORM A RESULTANT MATRIX            MPRD  70
C                                                                       MPRD  80
C        USAGE                                                          MPRD  90
C           CALL MPRD(A,B,R,N,M,MSA,MSB,L)                              MPRD 100
C                                                                       MPRD 110
C        DESCRIPTION OF PARAMETERS                                      MPRD 120
C           A - NAME OF FIRST INPUT MATRIX                              MPRD 130
C           B - NAME OF SECOND INPUT MATRIX                             MPRD 140
C           R - NAME OF OUTPUT MATRIX                                   MPRD 150
C           N - NUMBER OF ROWS IN A AND R                               MPRD 160
C           M - NUMBER OF COLUMNS IN A AND ROWS IN B                    MPRD 170
C           MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A         MPRD 180
C                  0 - GENERAL                                          MPRD 190
C                  1 - SYMMETRIC                                        MPRD 200
C                  2 - DIAGONAL                                         MPRD 210
C           MSB - SAME AS MSA EXCEPT FOR MATRIX B                       MPRD 220
C           L - NUMBER OF COLUMNS IN B AND R                            MPRD 230
C                                                                       MPRD 240
C        REMARKS                                                        MPRD 250
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRICES A OR B  MPRD 260
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROWMPRD 270
C           OF MATRIX B                                                 MPRD 280
C                                                                       MPRD 290
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MPRD 300
C           LOC                                                         MPRD 310
C                                                                       MPRD 320
C        METHOD                                                         MPRD 330
C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A MPRD 340
C           AND THE RESULT IS STORED IN THE N BY L MATRIX R. THIS IS A  MPRD 350
C           ROW INTO COLUMN PRODUCT.                                    MPRD 360
C           THE FOLLOWING TABLE SHOWS THE STORAGE MODE OF THE OUTPUT    MPRD 370
C           MATRIX FOR ALL COMBINATIONS OF INPUT MATRICES               MPRD 380
C                         A                B                R           MPRD 390
C                      GENERAL          GENERAL          GENERAL        MPRD 400
C                      GENERAL          SYMMETRIC        GENERAL        MPRD 410
C                      GENERAL          DIAGONAL         GENERAL        MPRD 420
C                      SYMMETRIC        GENERAL          GENERAL        MPRD 430
C                      SYMMETRIC        SYMMETRIC        GENERAL        MPRD 440
C                      SYMMETRIC        DIAGONAL         GENERAL        MPRD 450
C                      DIAGONAL         GENERAL          GENERAL        MPRD 460
C                      DIAGONAL         SYMMETRIC        GENERAL        MPRD 470
C                      DIAGONAL         DIAGONAL         DIAGONAL       MPRD 480
```

```
C                                                                       MPRD 490
C     ..................................................................MPRD 500
C                                                                       MPRD 510
      SUBROUTINE MPRD(A,B,R,N,M,MSA,MSB,L)                              MPRD 520
      DIMENSION A(1),B(1),R(1)                                          MPRD 530
C                                                                       MPRD 540
C        SPECIAL CASE FOR DIAGONAL BY DIAGONAL                          MPRD 550
C                                                                       MPRD 560
      MS=MSA*10+MSB                                                     MPRD 570
      IF(MS-22) 30,10,30                                                MPRD 580
   10 DO 20 I=1,N                                                       MPRD 590
   20 R(I)=A(I)*B(I)                                                    MPRD 600
      RETURN                                                            MPRD 610
C                                                                       MPRD 620
C        ALL OTHER CASES                                                MPRD 630
C                                                                       MPRD 640
   30 IR=1                                                              MPRD 650
      DO 90 K=1,L                                                       MPRD 660
      DO 90 J=1,N                                                       MPRD 670
      R(IR)=0                                                           MPRD 680
      DO 80 I=1,M                                                       MPRD 690
      IF(MS) 40,60,40                                                   MPRD 700
   40 CALL LOC(J,I,IA,N,M,MSA)                                          MPRD 710
      CALL LOC(I,K,IB,M,L,MSB)                                          MPRD 720
      IF(IA) 50,80,50                                                   MPRD 730
   50 IF(IB) 70,80,70                                                   MPRD 740
   60 IA=N*(I-1)+J                                                      MPRD 750
      IB=M*(K-1)+I                                                      MPRD 760
   70 R(IR)=R(IR)+A(IA)*B(IB)                                           MPRD 770
   80 CONTINUE                                                          MPRD 780
   90 IR=IR+1                                                           MPRD 790
      RETURN                                                            MPRD 800
      END                                                               MPRD 810
```

```
      READ SUBROUTINE CCUT
C
C     ..................................................................
C
C        SUBROUTINE CCUT
C
C        PURPOSE
C           PARTITION A MATRIX BETWEEN SPECIFIED COLUMNS TO FORM TWO
C           RESULTANT MATRICES
C
C        USAGE
C           CALL CCUT (A,L,R,S,N,M,MS)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF INPUT MATRIX
C           L - COLUMN OF A TO THE LEFT OF WHICH PARTITIONING TAKES
C               PLACE
C           R - NAME OF MATRIX TO BE FORMED FROM LEFT PORTION OF A
C           S - NAME OF MATRIX TO BE FORMED FROM RIGHT PORTION OF A
C           N - NUMBER OF ROWS IN A
C           M - NUMBER OF COLUMNS IN A
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C
C        REMARKS
C           MATRIX R CANNOT BE IN SAME LOCATION AS MATRIX A
C           MATRIX S CANNOT BE IN SAME LOCATION AS MATRIX A
C           MATRIX R CANNOT BE IN SAME LOCATION AS MATRIX S
C           MATRIX R AND MATRIX S ARE ALWAYS GENERAL MATRICES
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           LOC
C
C        METHOD
C           ELEMENTS OF MATRIX A TO THE LEFT OF COLUMN L ARE MOVED TO
C           FORM MATRIX R OF N ROWS AND L-1 COLUMNS. ELEMENTS OF
C           MATRIX A IN COLUMN L AND TO THE RIGHT OF L ARE MOVED TO FORM
C           MATRIX S OF N ROWS AND M-L+1 COLUMNS.
C
C     ..................................................................
C
      SUBROUTINE CCUT(A,L,R,S,N,M,MS)
      DIMENSION A(1),R(1),S(1)
C
      IR=0
```

```
      IS=0                                                              CCUT 470
      DO 70 J=1,M                                                       CCUT 480
      DO 70 I=1,N                                                       CCUT 490
C                                                                       CCUT 500
C        FIND LOCATION IN OUTPUT MATRIX AND SET TO ZERO                 CCUT 510
C                                                                       CCUT 520
      IF(J-L) 20,10,10                                                  CCUT 530
   10 IS=IS+1                                                           CCUT 540
      S(IS)=0.0                                                         CCUT 550
      GO TO 30                                                          CCUT 560
   20 IR=IR+1                                                           CCUT 570
      R(IR)=0.0                                                         CCUT 580
C                                                                       CCUT 590
C        LOCATE ELEMENT FOR ANY MATRIX STORAGE MODE                     CCUT 600
C                                                                       CCUT 610
   30 CALL LOC(I,J,IJ,N,M,MS)                                           CCUT 620
C                                                                       CCUT 630
C        TEST FOR ZERO ELEMENT IN DIAGONAL MATRIX                       CCUT 640
C                                                                       CCUT 650
      IF(IJ) 40,70,40                                                   CCUT 660
C                                                                       CCUT 670
C        DETERMINE WHETHER RIGHT OR LEFT OF L                           CCUT 680
C                                                                       CCUT 690
   40 IF(J-L) 60,50,50                                                  CCUT 700
   50 S(IS)=A(IJ)                                                       CCUT 710
      GO TO 70                                                          CCUT 720
   60 R(IR)=A(IJ)                                                       CCUT 730
   70 CONTINUE                                                          CCUT 740
      RETURN                                                            CCUT 750
      END                                                               CCUT 760
```

```
C                                                                       MINV  10
C     ..................................................................MINV  20
C                                                                       MINV  30
C        SUBROUTINE MINV                                                MINV  40
C                                                                       MINV  50
C        PURPOSE                                                        MINV  60
C           INVERT A MATRIX                                             MINV  70
C                                                                       MINV  80
C        USAGE                                                          MINV  90
C           CALL MINV(A,N,D,L,M)                                        MINV 100
C                                                                       MINV 110
C        DESCRIPTION OF PARAMETERS                                      MINV 120
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY  MINV 130
C               RESULTANT INVERSE.                                      MINV 140
C           N - ORDER OF MATRIX A                                       MINV 150
C           D - RESULTANT DETERMINANT                                   MINV 160
C           L - WORK VECTOR OF LENGTH N                                 MINV 170
C           M - WORK VECTOR OF LENGTH N                                 MINV 180
C                                                                       MINV 190
C        REMARKS                                                        MINV 200
C           MATRIX A MUST BE A GENERAL MATRIX                           MINV 210
C                                                                       MINV 220
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MINV 230
C           NONE                                                        MINV 240
C                                                                       MINV 250
C        METHOD                                                         MINV 260
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   MINV 270
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    MINV 280
C           THE MATRIX IS SINGULAR.                                     MINV 290
C                                                                       MINV 300
C     ..................................................................MINV 310
C                                                                       MINV 320
      SUBROUTINE MINV(A,N,D,L,M)                                        MINV 330
      DIMENSION A(1),L(1),M(1)                                          MINV 340
C                                                                       MINV 350
C        ...............................................................MINV 360
C                                                                       MINV 370
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  MINV 380
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION      MINV 390
C        STATEMENT WHICH FOLLOWS.                                       MINV 400
C                                                                       MINV 410
C     DOUBLE PRECISION A,D,BIGA,HOLD                                    MINV 420
C                                                                       MINV 430
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS    MINV 440
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS      MINV 450
C        ROUTINE.                                                       MINV 460
C                                                                       MINV 470
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO      MINV 480
```

```
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ...............................................................
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
```

```
C                                                                       MINV 970
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS        MINV 980
C        CONTAINED IN BIGA)                                             MINV 990
C                                                                       MINV1000
   45 IF(BIGA) 48,46,48                                                 MINV1010
   46 D=0.0                                                             MINV1020
      RETURN                                                            MINV1030
   48 DO 55 I=1,N                                                       MINV1040
      IF(I-K) 50,55,50                                                  MINV1050
   50 IK=NK+I                                                           MINV1060
      A(IK)=A(IK)/(-BIGA)                                               MINV1070
   55 CONTINUE                                                          MINV1080
C                                                                       MINV1090
C        REDUCE MATRIX                                                  MINV1100
C                                                                       MINV1110
      DO 65 I=1,N                                                       MINV1120
      IK=NK+I                                                           MINV1130
      HOLD=A(IK)                                                        MINV1140
      IJ=I-N                                                            MINV1150
      DO 65 J=1,N                                                       MINV1160
      IJ=IJ+N                                                           MINV1170
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
   65 CONTINUE                                                          MINV1220
C                                                                       MINV1230
C        DIVIDE ROW BY PIVOT                                            MINV1240
C                                                                       MINV1250
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
C                                                                       MINV1320
C        PRODUCT OF PIVOTS                                              MINV1330
C                                                                       MINV1340
      D=D*BIGA                                                          MINV1350
C                                                                       MINV1360
C        REPLACE PIVOT BY RECIPROCAL                                    MINV1370
C                                                                       MINV1380
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
C                                                                       MINV1410
C        FINAL ROW AND COLUMN INTERCHANGE                               MINV1420
C                                                                       MINV1430
      K=N                                                               MINV1440
```

```
100 K=(K-1)                                                             MINV1450
    IF(K) 150,150,105                                                   MINV1460
105 I=L(K)                                                              MINV1470
    IF(I-K) 120,120,108                                                 MINV1480
108 JQ=N*(K-1)                                                          MINV1490
    JR=N*(I-1)                                                          MINV1500
    DO 110 J=1,N                                                        MINV1510
    JK=JQ+J                                                             MINV1520
    HOLD=A(JK)                                                          MINV1530
    JI=JR+J                                                             MINV1540
    A(JK)=-A(JI)                                                        MINV1550
110 A(JI) =HOLD                                                         MINV1560
120 J=M(K)                                                              MINV1570
    IF(J-K) 100,100,125                                                 MINV1580
125 KI=K-N                                                              MINV1590
    DO 130 I=1,N                                                        MINV1600
    KI=KI+N                                                             MINV1610
    HOLD=A(KI)                                                          MINV1620
    JI=KI-K+J                                                           MINV1630
    A(KI)=-A(JI)                                                        MINV1640
130 A(JI) =HOLD                                                         MINV1650
    GO TO 100                                                           MINV1660
150 RETURN                                                              MINV1670
    END                                                                 MINV1680
```

```
C
C     ..................................................................
C
C        SUBROUTINE SINV
C
C        PURPOSE
C           INVERT A GIVEN SYMMETRIC POSITIVE DEFINITE MATRIX
C
C        USAGE
C           CALL SINV(A,N,EPS,IER)
C
C        DESCRIPTION OF PARAMETERS
C           A      - UPPER TRIANGULAR PART OF THE GIVEN SYMMETRIC
C                    POSITIVE DEFINITE N BY N COEFFICIENT MATRIX.
C                    ON RETURN A CONTAINS THE RESULTANT UPPER
C                    TRIANGULAR MATRIX.
C           N      - THE NUMBER OF ROWS (COLUMNS) IN GIVEN MATRIX.
C           EPS    - AN INPUT CONSTANT WHICH IS USED AS RELATIVE
C                    TOLERANCE FOR TEST ON LOSS OF SIGNIFICANCE.
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0  - NO ERROR
C                    IER=-1 - NO RESULT BECAUSE OF WRONG INPUT PARAME-
C                             TER N OR BECAUSE SOME RADICAND IS NON-
C                             POSITIVE (MATRIX A IS NOT POSITIVE
C                             DEFINITE, POSSIBLY DUE TO LOSS OF SIGNI-
C                             FICANCE)
C                    IER=K  - WARNING WHICH INDICATES LOSS OF SIGNIFI-
C                             CANCE. THE RADICAND FORMED AT FACTORIZA-
C                             TION STEP K+1 WAS STILL POSITIVE BUT NO
C                             LONGER GREATER THAN ABS(EPS*A(K+1,K+1)).
C
C        REMARKS
C           THE UPPER TRIANGULAR PART OF GIVEN MATRIX IS ASSUMED TO BE
C           STORED COLUMNWISE IN N*(N+1)/2 SUCCESSIVE STORAGE LOCATIONS.
C           IN THE SAME STORAGE LOCATIONS THE RESULTING UPPER TRIANGU-
C           LAR MATRIX IS STORED COLUMNWISE TOO.
C           THE PROCEDURE GIVES RESULTS IF N IS GREATER THAN 0 AND ALL
C           CALCULATED RADICANDS ARE POSITIVE.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           MFSD
C
C        METHOD
C           SOLUTION IS DONE USING THE FACTORIZATION BY SUBROUTINE MFSD.
C
C     ..................................................................
C
      SUBROUTINE SINV(A,N,EPS,IER)
```

```
C
C
      DIMENSION A(1)
      DOUBLE PRECISION DIN,WORK
C
C        FACTORIZE GIVEN MATRIX BY MEANS OF SUBROUTINE MFSD
C        A = TRANSPOSE(T) * T
      CALL MFSD(A,N,EPS,IER)
      IF(IER) 9,1,1
C
C        INVERT UPPER TRIANGULAR MATRIX T
C        PREPARE INVERSION-LOOP
    1 IPIV=N*(N+1)/2
      IND=IPIV
C
C        INITIALIZE INVERSION-LOOP
      DO 6 I=1,N
      DIN=1.D0/DBLE(A(IPIV))
      A(IPIV)=DIN
      MIN=N
      KEND=I-1
      LANF=N-KEND
      IF(KEND) 5,5,2
    2 J=IND
C
C        INITIALIZE ROW-LOOP
      DO 4 K=1,KEND
      WORK=0.D0
      MIN=MIN-1
      LHOR=IPIV
```

```
C
C     ..................................................................
C
C        SUBROUTINE MFSD
C
C        PURPOSE
C           FACTOR A GIVEN SYMMETRIC POSITIVE DEFINITE MATRIX
C
C        USAGE
C           CALL MFSD(A,N,EPS,IER)
C
C        DESCRIPTION OF PARAMETERS
C           A      - UPPER TRIANGULAR PART OF THE GIVEN SYMMETRIC
C                    POSITIVE DEFINITE N BY N COEFFICIENT MATRIX.
C                    ON RETURN A CONTAINS THE RESULTANT UPPER
C                    TRIANGULAR MATRIX.
C           N      - THE NUMBER OF ROWS (COLUMNS) IN GIVEN MATRIX.
C           EPS    - AN INPUT CONSTANT WHICH IS USED AS RELATIVE
C                    TOLERANCE FOR TEST ON LOSS OF SIGNIFICANCE.
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0  - NO ERROR
C                    IER=-1 - NO RESULT BECAUSE OF WRONG INPUT PARAME-
C                             TER N OR BECAUSE SOME RADICAND IS NON-
C                             POSITIVE (MATRIX A IS NOT POSITIVE
C                             DEFINITE, POSSIBLY DUE TO LOSS OF SIGNI-
C                             FICANCE)
C                    IER=K  - WARNING WHICH INDICATES LOSS OF SIGNIFI-
C                             CANCE. THE RADICAND FORMED AT FACTORIZA-
C                             TION STEP K+1 WAS STILL POSITIVE BUT NO
C                             LONGER GREATER THAN ABS(EPS*A(K+1,K+1)).
C
C        REMARKS
C           THE UPPER TRIANGULAR PART OF GIVEN MATRIX IS ASSUMED TO BE
C           STORED COLUMNWISE IN N*(N+1)/2 SUCCESSIVE STORAGE LOCATIONS.
C           IN THE SAME STORAGE LOCATIONS THE RESULTING UPPER TRIANGU-
C           LAR MATRIX IS STORED COLUMNWISE TOO.
C           THE PROCEDURE GIVES RESULTS IF N IS GREATER THAN 0 AND ALL
C           CALCULATED RADICANDS ARE POSITIVE.
C           THE PRODUCT OF RETURNED DIAGONAL TERMS IS EQUAL TO THE
C           SQUARE-ROOT OF THE DETERMINANT OF THE GIVEN MATRIX.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           SOLUTION IS DONE USING THE SQUARE-ROOT METHOD OF CHOLESKY.
C           THE GIVEN MATRIX IS REPRESENTED AS PRODUCT OF TWO TRIANGULAR
C           MATRICES, WHERE THE LEFT HAND FACTOR IS THE TRANSPOSE OF
C           THE RETURNED RIGHT HAND FACTOR.
C
C     ..................................................................
C
      SUBROUTINE MFSD(A,N,EPS,IER)
C
C
      DIMENSION A(1)
      DOUBLE PRECISION DPIV,DSUM
C
C        TEST ON WRONG INPUT PARAMETER N
      IF(N-1) 12,1,1
```

```
    1 IER=0                                                             MFSD 610
C                                                                       MFSD 620
C        INITIALIZE DIAGONAL-LOOP                                       MFSD 630
      KPIV=0                                                            MFSD 640
      DO 11 K=1,N                                                       MFSD 650
      KPIV=KPIV+K                                                       MFSD 660
      IND=KPIV                                                          MFSD 670
      LEND=K-1                                                          MFSD 680
C                                                                       MFSD 690
C        CALCULATE TOLERANCE                                            MFSD 700
      TOL=ABS(EPS*A(KPIV))                                              MFSD 710
C                                                                       MFSD 720
C        START FACTORIZATION-LOOP OVER K-TH ROW                         MFSD 730
      DO 11 I=K,N                                                       MFSD 740
      DSUM=0.D0                                                         MFSD 750
      IF(LEND) 2,4,2                                                    MFSD 760
C                                                                       MFSD 770
C        START INNER LOOP                                               MFSD 780
    2 DO 3 L=1,LEND                                                     MFSD 790
      LANF=KPIV-L                                                       MFSD 800
      LIND=IND-L                                                        MFSD 810
    3 DSUM=DSUM+DBLE(A(LANF)*A(LIND))                                   MFSD 820
C        END OF INNER LOOP                                              MFSD 830
C                                                                       MFSD 840
C        TRANSFORM ELEMENT A(IND)                                       MFSD 850
    4 DSUM=DBLE(A(IND))-DSUM                                            MFSD 860
      IF(I-K) 10,5,10                                                   MFSD 870
C                                                                       MFSD 880
C        TEST FOR NEGATIVE PIVOT ELEMENT AND FOR LOSS OF SIGNIFICANCE   MFSD 890
    5 IF(SNGL(DSUM)-TOL) 6,6,9                                          MFSD 900
    6 IF(DSUM) 12,12,7                                                  MFSD 910
    7 IF(IER) 8,8,9                                                     MFSD 920
    8 IER=K-1                                                           MFSD 930
C                                                                       MFSD 940
C        COMPUTE PIVOT ELEMENT                                          MFSD 950
    9 DPIV=DSQRT(DSUM)                                                  MFSD 960
      A(KPIV)=DPIV                                                      MFSD 970
      DPIV=1.D0/DPIV                                                    MFSD 980
      GO TO 11                                                          MFSD 990
C                                                                       MFSD1000
C        CALCULATE TERMS IN ROW                                         MFSD1010
   10 A(IND)=DSUM*DPIV                                                  MFSD1020
   11 IND=IND+I                                                         MFSD1030
C                                                                       MFSD1040
C        END OF DIAGONAL-LOOP                                           MFSD1050
      RETURN                                                            MFSD1060
   12 IER=-1                                                            MFSD1070
      RETURN                                                            MFSD1080
      END                                                               MFSD1090
```

```
C                                                                       MXOU  10
C     ..................................................................MXOU  20
C                                                                       MXOU  30
C        SUBROUTINE MXOUT                                               MXOU  40
C                                                                       MXOU  50
C        PURPOSE                                                        MXOU  60
C           PRODUCES AN OUTPUT LISTING OF ANY SIZED ARRAY ON            MXOU  70
C           LOGICAL UNIT 6                                              MXOU  80
C                                                                       MXOU  90
C        USAGE                                                          MXOU 100
C           CALL MXOUT(ICODE,A,N,M,MS,LINS,IPOS,ISP)                    MXOU 110
C                                                                       MXOU 120
C        DESCRIPTION OF PARAMETERS                                      MXOU 130
C           ICODE- INPUT CODE NUMBER TO BE PRINTED ON EACH OUTPUT PAGE  MXOU 140
C           A-NAME OF OUTPUT MATRIX                                     MXOU 150
C           N-NUMBER OF ROWS IN A                                       MXOU 160
C           M-NUMBER OF COLUMNS IN A                                    MXOU 170
C           MS-STORAGE MODE OF A WHERE MS=                              MXOU 180
C                   0-GENERAL                                           MXOU 190
C                   1-SYMMETRIC                                         MXOU 200
C                   2-DIAGONAL                                          MXOU 210
C           LINS-NUMBER OF PRINT LINES ON THE PAGE (USUALLY 60)         MXOU 220
C           IPOS-NUMBER OF PRINT POSITIONS ACROSS THE PAGE (USUALLY 132)MXOU 230
C           ISP-LINE SPACING CODE, 1 FOR SINGLE SPACE, 2 FOR DOUBLE     MXOU 240
C               SPACE                                                   MXOU 250
C                                                                       MXOU 260
C        REMARKS                                                        MXOU 270
C           THIS SUBROUTINE HAS BEEN MODIFIED BY M.HUTTON ON 11/27/71   MXOU 280
C           TO REDUCE THE AMOUNT OF EXTRA PRINTOUT. TO RETURN THE       MXOU 281
C           SUBROUTINE TO ITS ORIGINAL FORM MODIFY CARDS 280,480,590,   MXOU 282
C           650,900,920 ACCORDING TO THE SSP MANUAL AND REMOVE CARDS    MXOU 283
C           281-284,591.                                                MXOU 284
C                                                                       MXOU 290
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MXOU 300
C           LOC                                                         MXOU 310
C                                                                       MXOU 320
C        METHOD                                                         MXOU 330
C           THIS SUBROUTINE CREATES A STANDARD OUTPUT LISTING OF ANY    MXOU 340
C           SIZED ARRAY WITH ANY STORAGE MODE. EACH PAGE IS HEADED WITH MXOU 350
C           THE CODE NUMBER,DIMENSIONS AND STORAGE MODE OF THE ARRAY.   MXOU 360
C           EACH COLUMN AND ROW IS ALSO HEADED WITH ITS RESPECTIVE      MXOU 370
C           NUMBER.                                                     MXOU 380
C                                                                       MXOU 390
C     ..................................................................MXOU 400
C                                                                       MXOU 410
      SUBROUTINE MXOUT (ICODE,A,N,M,MS,LINS,IPOS,ISP)                   MXOU 420
      DIMENSION A(1),B(8)                                               MXOU 430
    1 FORMAT(1H1,5X, 7HMATRIX ,I5,6X,I3,5H ROWS,6X,I3,8H COLUMNS,       MXOU 440
     18X,13HSTORAGE MODE ,I1,8X,5HPAGE ,I2,/)                           MXOU 450
    2 FORMAT(12X,8HCOLUMN  ,7(3X,I3,10X))                               MXOU 460
    3 FORMAT(1H )                                                       MXOU 470
    4 FORMAT(1H ,4X,7(E16.6))                                           MXOU 480
    5 FORMAT(1H0,4X,7(E16.6))                                           MXOU 490
C   4 FORMAT(1H ,14X,6(E18.8))                                          MXOU 480
C   5 FORMAT(1H0,14X,6(E18.8))                                          MXOU 490
C                                                                       MXOU 500
      J=1                                                               MXOU 510
C                                                                       MXOU 520
C        WRITE HEADING                                                  MXOU 530
C                                                                       MXOU 540
```

```
      NEND=IPOS/16-1                                                    MXOU 550
      LEND=(LINS/ISP)-2                                                 MXOU 560
      IPAGE=1                                                           MXOU 570
   10 LSTRT=1                                                           MXOU 580
C  20 WRITE(6,1)ICODE,N,M,MS,IPAGE                                      MXOU 590
   20 CONTINUE                                                          MXOU 591
      JNT=J+NEND-1                                                      MXOU 600
      IPAGE=IPAGE+1                                                     MXOU 610
   31 IF(JNT-M)33,33,32                                                 MXOU 620
   32 JNT=M                                                             MXOU 630
   33 CONTINUE                                                          MXOU 640
C     WRITE(6,2)(JCUR,JCUR=J,JNT)                                       MXOU 650
      IF(ISP-1) 35,35,40                                                MXOU 660
   35 WRITE(6,3)                                                        MXOU 670
   40 LTEND=LSTRT+LEND-1                                                MXOU 680
      DO 80 L=LSTRT,LTEND                                               MXOU 690
C                                                                       MXOU 700
C        FORM OUTPUT ROW LINE                                           MXOU 710
C                                                                       MXOU 720
      DO 55 K=1,NEND                                                    MXOU 730
      KK=K                                                              MXOU 740
      JT = J+K-1                                                        MXOU 750
      CALL LOC(L,JT,IJNT,N,M,MS)                                        MXOU 760
      B(K)=0.0                                                          MXOU 770
      IF(IJNT)50,50,45                                                  MXOU 780
   45 B(K)=A(IJNT)                                                      MXOU 790
   50 CONTINUE                                                          MXOU 800
C                                                                       MXOU 810
C        CHECK IF LAST COLUMN.  IF YES GO TO 60                         MXOU 820
C                                                                       MXOU 830
      IF(JT-M) 55,60,60                                                 MXOU 840
   55 CONTINUE                                                          MXOU 850
C                                                                       MXOU 860
C        END OF LINE, NOW WRITE                                         MXOU 870
C                                                                       MXOU 880
   60 IF(ISP-1)65,65,70                                                 MXOU 890
   65 WRITE(6,4)  (B(JW),JW=1,KK)                                       MXOU 900
      GO TO 75                                                          MXOU 910
   70 WRITE(6,5)  (B(JW),JW=1,KK)                                       MXOU 920
C                                                                       MXOU 930
C        IF END OF ROWS,GO CHECK COLUMNS                                MXOU 940
C                                                                       MXOU 950
   75 IF(N-L)85,85,80                                                   MXOU 960
   80 CONTINUE                                                          MXOU 970
C                                                                       MXOU 980
C        END OF PAGE, NOW CHECK FOR MORE OUTPUT                         MXOU 990
C                                                                       MXOU1000
      LSTRT=LSTRT+LEND                                                  MXOU1010
      GO TO 20                                                          MXOU1020
C                                                                       MXOU1030
C        END OF COLUMNS, THEN RETURN                                    MXOU1040
C                                                                       MXOU1050
   85 IF(JT-M)90,95,95                                                  MXOU1060
   90 J=JT+1                                                            MXOU1070
      GO TO 10                                                          MXOU1080
   95 RETURN                                                            MXOU1090
      END                                                               MXOU1100
```

APPENDIX D PROCEDURE FOR ELIMINATING CONSTRAINT EQUATIONS IN TRIM PROBLEM

For linear dynamics and a quadratic performance criterion the trim problem can be written in the form

$$0 = a + B\delta \qquad (1)$$

$$r = 1/2\ \delta' R \delta \qquad (2)$$

with

a = constant vector of dimension n
δ = control vector of dimension $m \geq n$
B = $n \times m$ coefficient matrix
R = $m \times m$ positive definite weighting matrix

The objective is to find the set of control angles δ that satisfy (1) and minimize (2). The trim solution is given by

$$\delta = -B^{\#} a \qquad (3)$$

$$B^{\#} = R^{-1} B' (BR^{-1} B')^{-1} \qquad (4)$$

The $m \times n$ matrix $B^{\#}$ is a right inverse of B, i.e., $BB^{\#} = I$.

Consider the new trim problem that results from eliminating k of the n equality constraints. Suppose that the first k constraint equations in (1) are to be disregarded. The problem can always be written in this form by reordering the equations if necessary. Partitioning (1) gives that the new trim problem is

$$0 = a_2 + B_2 \delta_n \qquad (5)$$

$$r = 1/2\ \delta_n' R \delta_n \qquad (6)$$

where

$$a = \begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix} \begin{matrix} \updownarrow k \\ \updownarrow n-k \end{matrix} \qquad B = \begin{bmatrix} B_1 \\ \hline B_2 \end{bmatrix} \begin{matrix} \updownarrow k \\ \updownarrow n-k \end{matrix} \qquad (7)$$

The solution to the new trim problem is

$$\delta_n = -R^{-1}B_2'(B_2R^{-1}B_2')^{-1}a \tag{8}$$

The following question is of interest: Without starting the problem over again, is it possible to compute δ_n using the solution for δ ? The answer is affirmative and a procedure for computing δ_n is developed below.

From the partitioning (7) of the B matrix

$$R^{-1}B' = \left[\begin{array}{c|c} R^{-1}B_1' & R^{-1}B_2' \end{array}\right] \tag{9}$$

$$BR^{-1}B' = \left[\begin{array}{c|c} B_1R^{-1}B_1' & B_1R^{-1}B_2' \\ \hline B_2R^{-1}B_1' & B_2R^{-1}B_2' \end{array}\right] \tag{10}$$

Taking the inverse of (10) results in

$$(BR^{-1}B')^{-1} = \begin{array}{cc} \longleftarrow k \longrightarrow & \longleftarrow n-k \longrightarrow & \\ \left[\begin{array}{c|c} Q_1 & Q_2 \\ \hline Q_2' & Q_3 \end{array}\right] & & \begin{array}{c} k \\ n-k \end{array} \end{array} \tag{11}$$

where

$$Q_1 = E^{-1}$$

$$Q_2 = -E^{-1}B_1R^{-1}B_2'(B_2R^{-1}B_2')^{-1}$$

$$Q_3 = (B_2R^{-1}B_2')^{-1}B_2R^{-1}B_1'E^{-1}B_1R^{-1}B_2'(B_2R^{-1}B_2')^{-1} + (B_2R^{-1}B_2')^{-1}$$

and

$$E = B_1R^{-1}B_1' - B_1R^{-1}B_2'(B_2R^{-1}B_2')^{-1}B_2R^{-1}B_1'$$

Premultiplying (11) by (9) yields the right inverse of the B matrix in partitioned form

$$B^{\#} = \left[B_1^{\#} \;\vdots\; B^{\#}{}_2 \right]$$

where

$$B_1^{\#} = \left[I - R^{-1} B_2' (B_2 R^{-1} B_2)^{-1} B_2 \right] R^{-1} B_1' E^{-1} \tag{12}$$

$$B_2^{\#} = -\left[I - R^{-1} B_2' (B_2 R^{-1} B_2')^{-1} B_2 \right] R^{-1} B_1' E^{-1} B_1 R^{-1} B_2' (B_2 R^{-1} B_2')^{-1} + R^{-1} B_2' (B_2 R^{-1} B_2')^{-1}$$

Substituting (12) into (3) and using (7) gives that

$$\delta = -B_1^{\#} a_1 - B_2^{\#} a_2 \tag{13}$$

If we substitute

$$a_1 = B_1 R^{-1} B_2' (B_2 R^{-1} B_2')^{-1} a_2 \tag{14}$$

into (13) then from (8) and (12) it follows that

$$\delta = \delta_n \tag{15}$$

This result states that if the first k elements in the vector a are replaced by the values computed from (14) then the solution to the original trim problem becomes the solution to the new trim problem created by eliminating the first k constraint equations.

It is apparent from comparing (11) to (14) that (14) can be replaced by

$$a_1 = -Q_1^{-1} Q_2 a_2 \tag{16}$$

This is a more useful equation for computing the new value of a_1 since Q_1 and Q_2 are submatrices of a matrix computed in the solution of the original problem.

To summarize, the steps for computing δ_n are as follows:

1) Start the computation of δ using (3) and (4) in the usual way.
2) After computing $(BR^{-1}B')^{-1}$ form the submatrices Q_1 and Q_2 according to (11).
3) Replace subvector a_1 in a by the value computed from (16).
4) Continue the computation of δ in the usual way. The result will be $\delta = \delta_n$.

The above procedure for computing δ_n does not offer any particular advantage over using (8) if the calculations are to be done by hand. If a computer program, on the other hand, has been developed to compute δ then the above procedure minimizes the amount of program modification required to compute δ_n .

APPENDIX E VERIFICATION OF TRIMS PROGRAM

Lateral trim of the Space Shuttle is an example of the linear trim problem. The linear trim problem is to find the control deflections δ satisfying the equality constraints

$$a + B\delta = 0$$

and minimizing

$$J = 1/2\, \delta' R \delta$$

The solution is

$$\delta = - B^{\#} a$$

$$B^{\#} = - R^{-1} B' (B R^{-1} B')^{-1}$$

The problem of Space Shuttle trim in roll and yaw (two constraint equations) using the following four control deflections:

- yaw deflection of orbiter engine 1
- yaw deflection of orbiter engines 2 and 3
- pitch deflection of orbiter engine 2
  (negative of the pitch deflection of orbiter engine 3)
- rudder deflection

was solved at MSFC. The control deflection angles vs flight time for the case when the R matrix is

$$R = \text{Diag}\,[0.49,\ 0.49,\ 0.49,\ 1.00\,]$$

and the bias torques due to misalignments are

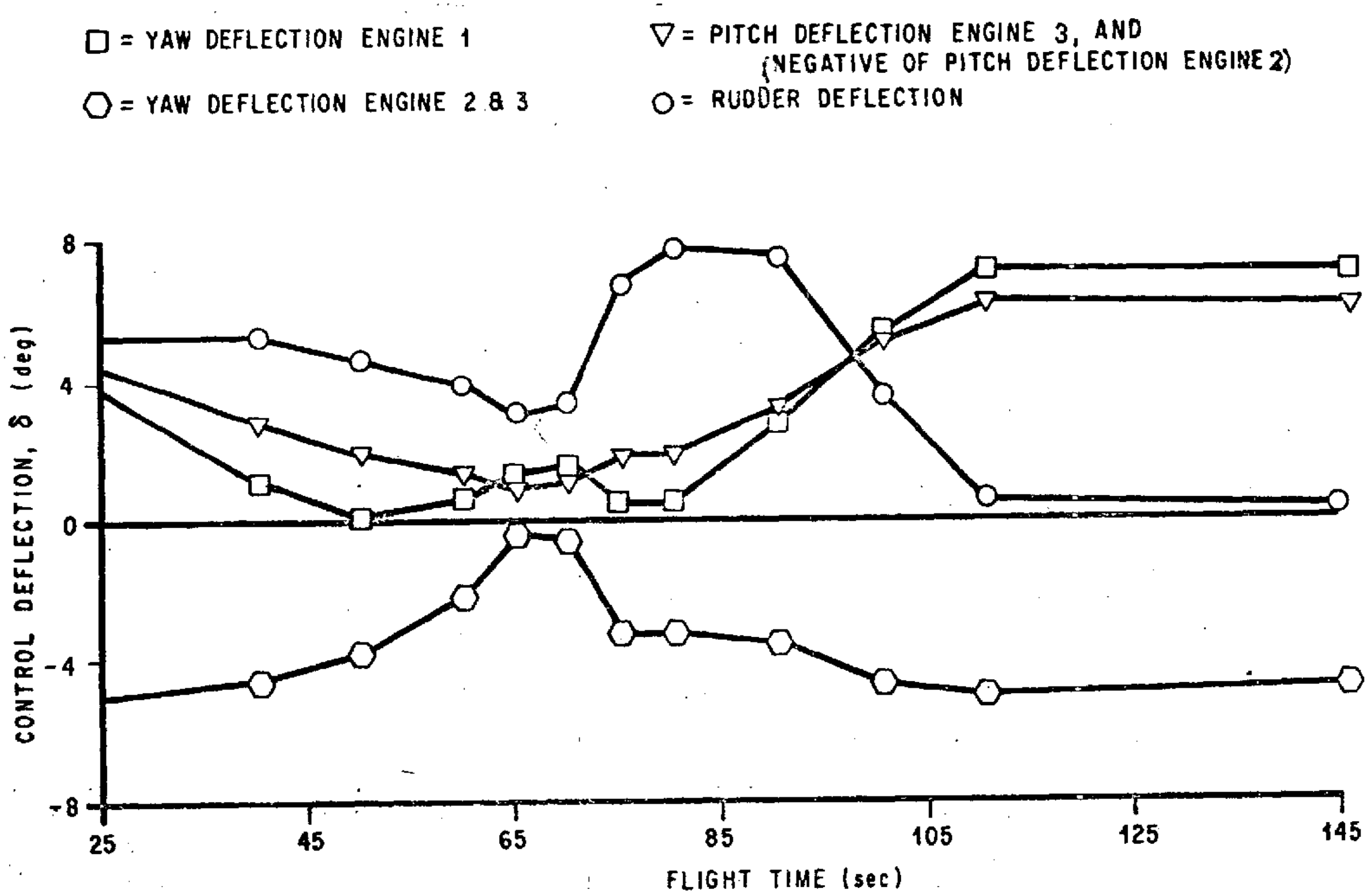
roll torque = 0.87×10^6 N·m
yaw torque = 3.02×10^6 N·m

are plotted in Figure E1.

The solution to (supposedly) the same trim problem was also computed using the TRIMS program as a check of the program. The resulting plot of control deflection angles vs flight time is shown in Figure E2. The TRIMS computation was repeated except without the dorsal fins and the trim solution is plotted in Figure E3.

The results in Figures E2 and E3 computed by TRIMS do not agree with the results in Figure E1 obtained by MSFC. A comparison of the results does not indicate the reason for the difference. The computation of δ and $B^{\#}$ from a , B , and R in the TRIMS program was checked against hand calculations. Most likely, the area of difficulty is in the computation of the vector a and matrix B from the equations of motion.

Figure E 1 Trim Soltuion Computed at MSFC



BIAS TORQUES
DUE TO MISALIGNMENTS

ROLL .87 X 10^6 N·m YAW 3.02 X 10^6 N·m

Figure E 2 Control Deflections vs Flight Time for Space Shuttle Trim in Roll and Yaw with Addition of Dorsal Fins

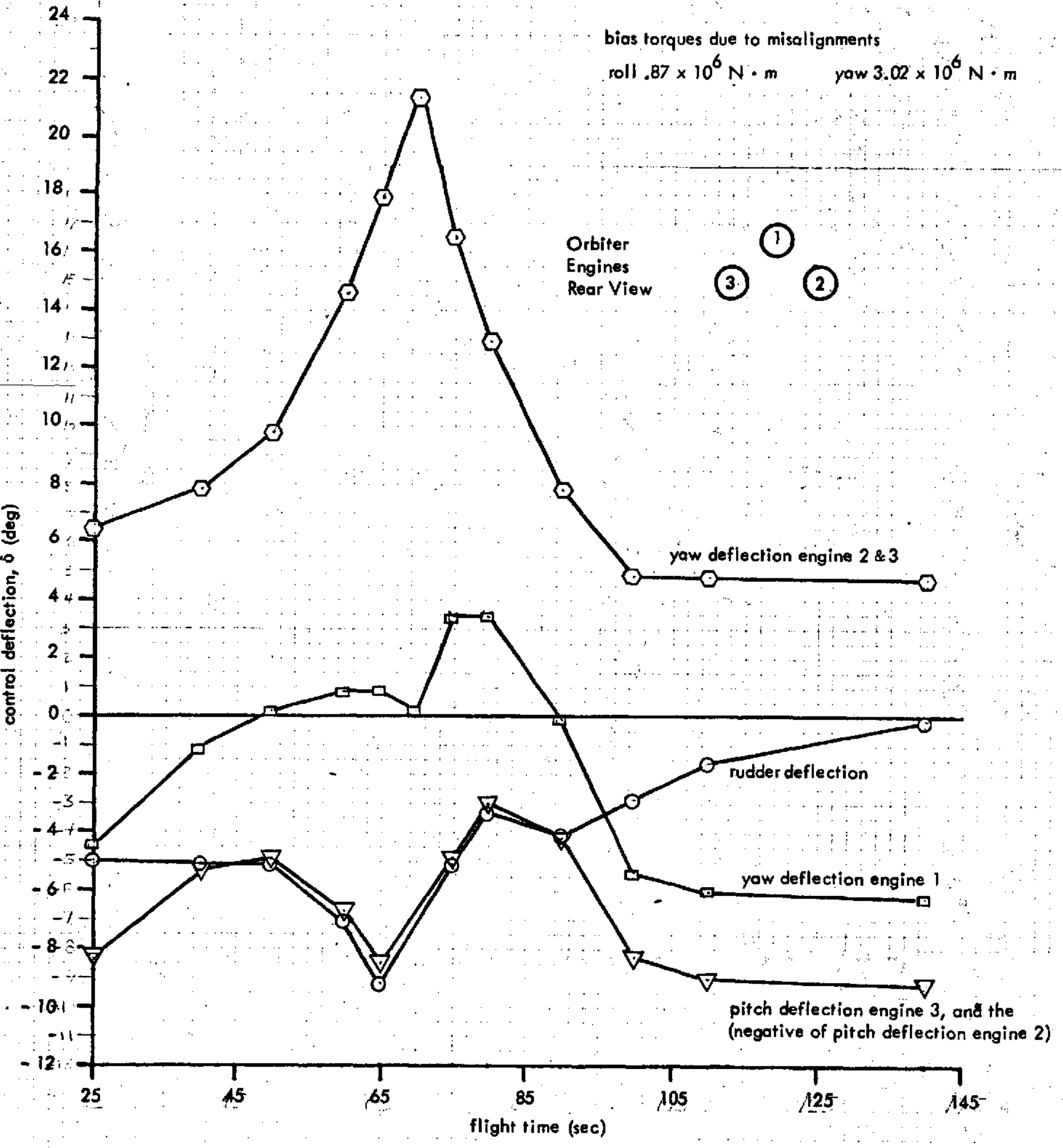
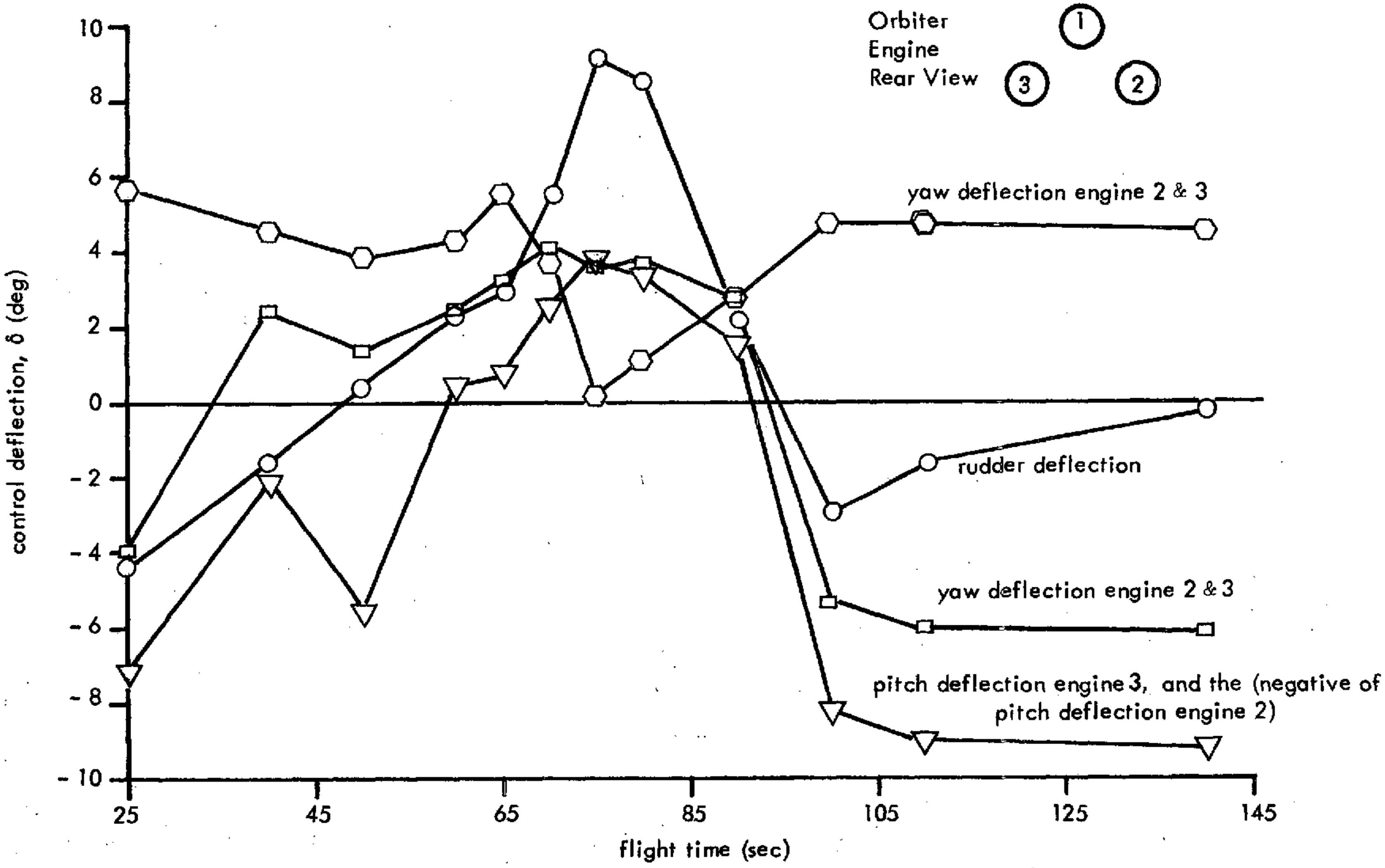


Figure E 3 Control Deflections vs Flight Time for Space Schuttle Trim in Roll and Yaw



bias torques due to misalignments roll .87 x 10^6 N · m yaw 3.02 x 10^6N · m

REFERENCES

1. Brockett, R., Finite Dimensional Linear Systems, John Wiley and Sons, Inc., New York, 1970.

2. Bryson, A. and Ho. Y , Applied Optimal Control, Blaisdell Publishing Co., Waltham, Mass., 1969.

3. Etkin, B., Dynamics of Flight-Stability and Control, John Wiley & Sons, Inc., New York, 1959.

4. Faddeev, D. K. and Faddeeva, V. N., Computional Methods of Linear Algebra, W. H. Freeman & Co., San Francisco, 1963.

5. Householder, A. S., The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Co., New York, 1964.

6. Kalman, R. F., Ho, Y. C., and Narendra, K. S., "Controllability of Linear Dynamical Systems," Contributions to Differential Equations, vol. 1, No. 2, 1961, pp. 182-213.

7. Johnson, C. D., "Optimization of a Certain Quality of Complete Controllability and Observability for Linear Dynamic Systems," J. Basic Engineering, Trans. ASME, ser. D, vol. 91, June 1969, pp. 228-238.

8. Friedland, B., "Limiting Forms of Optimum Stochastic Linear Regulators," Trans. ASME, Journal of Dynamic Systems, Measurement, and Control, September 1971, pp. 134-141.

8. Hutton, M. F., "Solutions of the Singular Stochastic Regulator Problem," to be published in the Journal of Dynamic Systems, Measurement, and Control.