TR-250-1246
DECEMBER 1973

ADVANCED CONTROL CONCEPTS

FINAL REPORT

(NASA-CR-120217) ADVANCED CONTROL CONCEPTS Final Report (Northrop Services, Inc., Huntsville, Ala.) ~~79~~ 92 p HC $7.00 CSCL 22B

N74-23385

Unclas
G3/31 16576

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Aero-Astrodynamics Laboratory

UNDER CONTRACT NAS8-29193

NORTHROP SERVICES, INC.
P. O. BOX 1484
HUNTSVILLE, ALABAMA 35807
TELEPHONE (205)837-0580

TR-250-1246

ADVANCED CONTROL CONCEPTS

FINAL REPORT

by

J. B. Sharp
J. M. Coppey

December 1973

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
AERO-ASTRODYNAMICS LABORATORY

Under Contract NAS8–29193

REVIEWED AND APPROVED BY:

M. A. Sloan, Jr.

M. A. Sloan, Jr., Manager
Dynamics and Control Section

NORTHROP SERVICES, INC.
HUNTSVILLE, ALABAMA

FOREWORD

This report was prepared in fulfillment of the requirements of NASA contract NAS8-29193. George C. Marshall Space Flight Center was the contracting agency. Technical Coordination was maintained through Dr. Steve Winder, Chief, Statistical Dynamics Section, S&E-AERO-DDD.

ABSTRACT

An investigation was made of the problems of excess control devices and insufficient trim control capability on shuttle ascent vehicles. The trim problem is solved at all time points of interest using Lagrangian multipliers and a Simplex based iterative algorithm developed as a result of the study. This algorithm has the capability to solve any bounded linear problem with physically realizable constraints and minimize any piecewise differentiable cost function. Both solution methods also automatically distribute the command torques to the control devices.

It is shown that trim requirements are unrealizable if only the orbiter engines and the aerodynamic surfaces are used. On the other hand if the solid engines are controllable there is ample control margin and realistic cost functions can be minimized to optimize the solution.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DEFINITION OF SYMBOLS AND ABBREVIATIONS

049	- Shuttle configuration 049
$\{T_c\}$	- Torque command profile
δ_j	- Control setting of j^{th} controller
$\omega(\delta_j)$	- Objective function
$\phi(\delta_j)$	- Equality constraints, i.e., yaw/roll moments = 0.0
$\delta_{j_{min}}$, $\delta_{j_{max}}$	- Boundary values for trim devices
λ_i	- Lagrange multipler for i^{th} constraint
$[\Omega]$	- Penalty matrix inverse
δ_a	- Aileron angle
δ_R	- Rudder angle
$\delta_{e_{i_y}}$	- Engine i yaw angle
$\delta_{e_{i_p}}$	- Engine i pitch angle
β	- Wind induced sideslip angle
$C_{\ell_{\delta_i}}$	- Roll moment due to δ_i
$C_{n_{\delta_i}}$	- Yaw moment due to δ_i
α,γ	- Constants defined in the text for convenience
Z_{cg}	- cg - Z location
Z_{e_i}	- Engine-i Z location
P_{e_i}	- Engine i pitch angle
Y_{e_i}	- Engine i yaw angle
SRM	- Solid rocket motor(s)
TVC	- Thrust vector control

DEFINITION OF SYMBOLS AND ABBREVIATIONS (Concluded)

Xj	- Independent variables of linear programming problem
a_j^o	- j^{th} coefficient of objective function of linear programming problem
Z	- Objective function of linear programming problem
a_j^i	- j^{th} coefficient of i^{th} constraint equation of linear programming problem
b^i	- Right hand side of i^{th} constraint equation of linear programming problem
$\bar{q}$	- Dynamic pressure, nts/m^2
S	- Reference area, m^2
b	- Reference length, m
F	- $\omega + \sum_{i=1}^{m} \lambda_i \phi_i$
Δ	- Range percentage used for iterative Simplex solution
NAR	- North American Rockwell

Section I
INTRODUCTION

This report presents the results of a study to develop a technique to solve the trim problem of an aerospace vehicle when the number of control variables exceeds the number of variables to control trim. Distribution of the trim commands to the various control variables in some "optimum" manner is a further goal of the study. The goal of the study is not a computer program as much as it is the development a practical method of understanding and solving the trim problem.

The importance of solving this type problem became evident as the Space Shuttle vehicle requirements were established. In particular, the ascent phase of flight is one area in which these problems arise. In ascent the mated configuration has only one plane of symmetry causing the inherent problem of zero lift and zero moment occurring at different angles-of-attack. Cross-coupling problems are also severe in some candidate designs.

Several candidate configurations have insufficient control authority to control the vehicle in the presence of headwinds or crosswinds with orbiter engines only. When the aerodynamic surfaces on the orbiter are used, the shuttle may or may not be controllable, but in any case the number of control effectors becomes greater than the number of states to control, creating an infinity of trim solutions for any constant disturbance. If controllability problems require vectoring the solid-rocket motors, the number of trim combinations proliferate.

The specific goal of this study was to solve the problem of distributing the trim commands to the various controllers, simultaneously using the extra control choice latitude to optimize some phase of the ascent flight. Further goals were that the solution to the trim problem be simple and easily implemented and that the solution process give the engineer a "feel" for the relative difficulty of meeting the trim requirements as compared to the trim power available.

Of equal importance to solving this problem is the delineation of the concepts involved in its solution. Since the sample problems chosen are realistic, a computer program was necessary to handle the data efficiently, but the ideas behind the program receive the emphasis throughout this report.

The problems that arise in trimming the Shuttle ascent vehicle are primarily in the roll/yaw planes. The shuttle lateral plane trim problem can be outlined in the following manner.

- There are three vehicle state variables: roll, yaw, and sideslip angle.
- There are eight control devices with effectiveness in the lateral plane: three orbiter engines, two solid-booster engines, right and left aileron and a rudder. Each engine has two degrees of freedom (pitch and yaw); however, the top orbiter engine is on the vertical centerline of the vehicle (049 data). Therefore, it has no effect on the lateral dynamics when it is pitched. The net result is 12 control states in the lateral plane.
- All control devices are limited by control stops or hinge moments or both.
- The problem is to find the range of control effector settings that satisfy the trim requirements on the states in some optimum manner.

The reason for selecting the trim problem to optimize the control devices is that Shuttle command torques and forces were currently unknown. Therefore, trimming an aerodynamic disturbance, such as a mean wind, is a convenient substitute for a torque command profile. The results of this study will give not only a trim solution, optimum in some manner, but a mechanization of the distribution of commands to the control devices to achieve a given flight state.

Figure 1-1 illustrates the approach described in the previous paragraph. The T_c quantities represent the torque command profile. The optimization scheme should be independent of the torque commands so that the guidance design functions and control design functions can proceed somewhat independently of each other until system integration time.

The general problem can be stated formally as follows:

minimize

$$\omega(\delta_j)$$

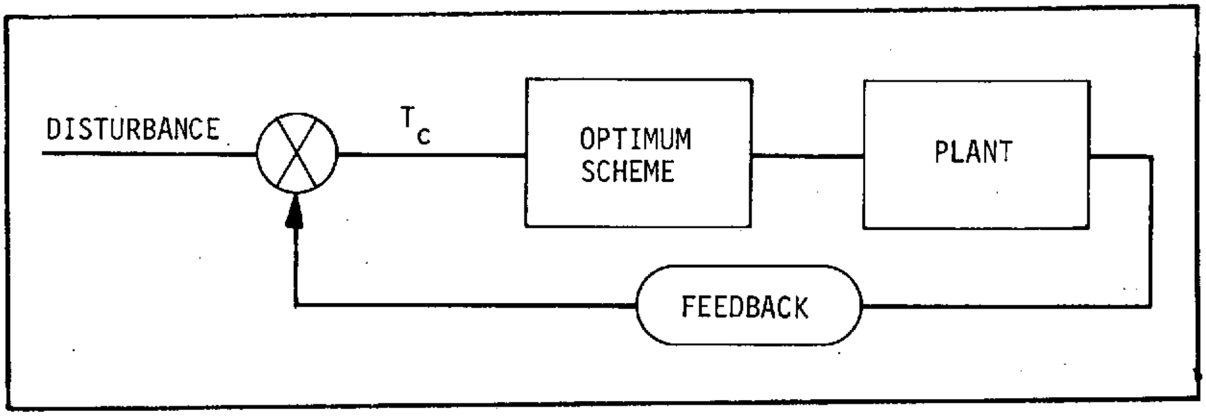


Figure 1-1. FEEDBACK DIAGRAM SHOWING OPTIMUM SCHEME

subject to the trim constraints

$$\phi_i(\delta_j) - T_{c_i} = 0$$

$$i = 1, 2, \ldots, m$$
$$j = 1, 2, \ldots, n$$

and inequality constraints

$$\delta_{j_{min}} \leq \delta_j \leq \delta_{j_{max}}$$

where the T_c represent a torque command of a disturbance vector.

In the Shuttle trim problem considered, the $\phi_i(\delta_j)$ can be represented by a set of i linear equations and the T_{c_i} are just constants. This means that the equation involving $\phi_i(\delta_j)$ can be solved for i δ_j's if j-i δ_j's are chosen beforehand. The boundaries of the problem can be explored by setting j-i control settings on the bounds and calculating the remaining i control variables.

Data for a Shuttle type vehicle were furnished by the COR. These data are presented in Appendix A. The configuration is similar to the 049 configuration with a 400-ft^2 forward fin to reduce the lateral stability. No drag data for the aerodynamic surfaces were included, but the rudder drag coefficient was estimated from wind tunnel data on later configurations. The aileron drag coefficient was assumed to be equal to the rudder for the purposes of this study. No aileron side force coefficient was included in the data either, but this turned out to be unnecessary for the problems solved.

The position limits for the rudder and aileron are shown in the data as well as the hinge moment limits for these surfaces at various flight times. The engine gimbal stop limits were assumed to be 10 degrees circular displacement for the orbiter engines and 10 degrees diagonally up or down for the solid-rocket motors. No data on engine hinge moment limits were available so they were not considered in the problem solution. However, the method of solution used would handle this type of restriction with a simple data change. The mean cross-wind was used as a disturbance for much of the study (Appendix A).

Figure 1-2 shows the mean crosswind included roll/yaw torques (normalized by dividing by $\bar{q}$ sb) at times 25-90 seconds. As can be seen, the yaw torque peaks at 60-65 seconds and the roll torque peaks at about 70 seconds. However, dynamic pressure peaks at 80 seconds so the maximum controllability problem occurs around 75-80 seconds. It is impossible to predict the exact time in advance, since the control problem also depends on the individual control effectiveness and their proportional roll/yaw effectiveness. These parameters are functions of thrust, dynamic pressure and Mach number.

The approach and results section presents in somewhat chronological order the approaches taken and the results achieved in the study. Specifics involving data, some of the mathematical developments, and the computer program developed are included in the Appendices.

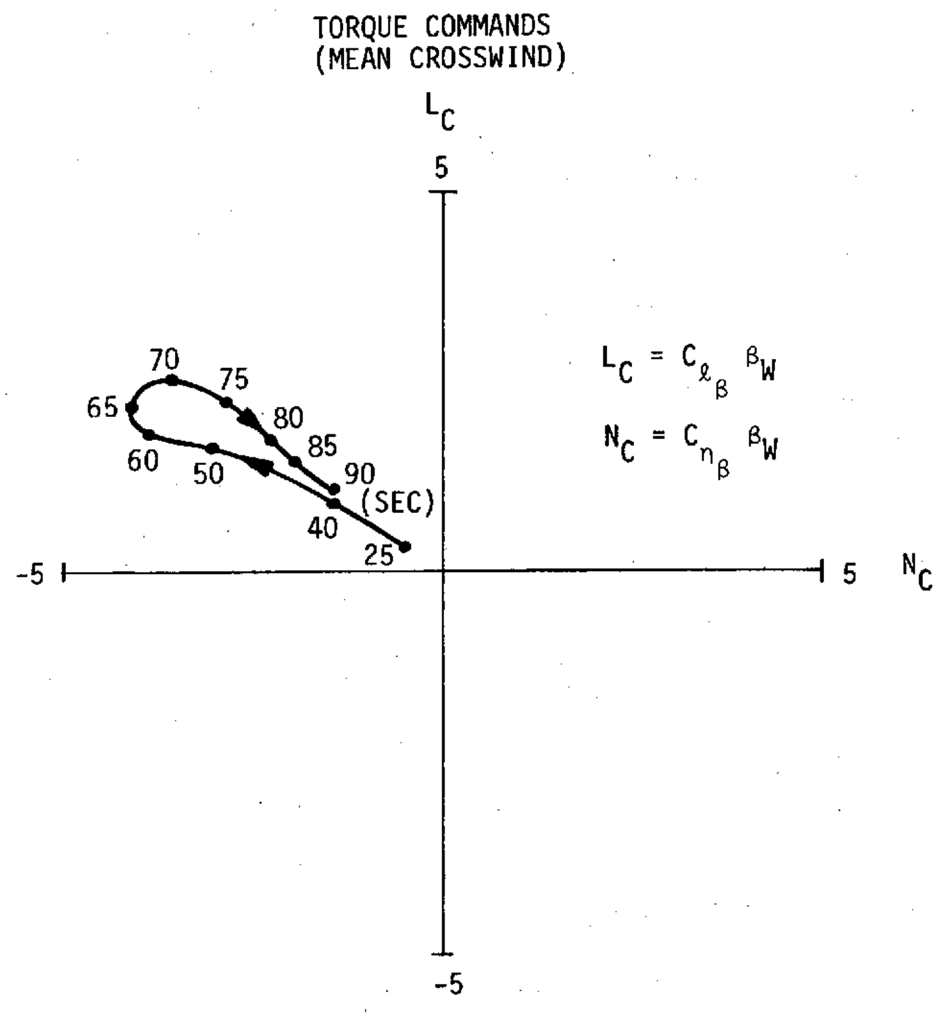


Figure 1-2. MEAN CROSSWIND INDUCED ROLL/YAW TORQUES FROM 25-90 SECONDS

Section II

APPROACH AND RESULTS

The trim problem was approached by first formulating the constraints and limits for the control effectors in a form suitable for analysis. Then, to give the investigator a feel for the problem, several optimized trim solutions were calculated for special cases. After sufficient feel was obtained, the optimization problem was solved using a quadratic cost function. This problem has been solved many times and it can be shown that any cost functional that can be represented by a quadratic cost function can be solved by Lagrangian multipliers, the technique used in this part of the study. Several sample problems were reduced by this method with the objective of giving additional insight into the problem. Since the Lagrangian multiplier technique cannot directly handle inequality constraints, this approach does not guarantee a realizable solution even if one is available. Therefore, the next step in the approach was to develop a means of solving the trim problem with inequality constraints and with a general form cost functional.

Several methods were available to solve bounded problems with nonquadratic cost functions. All of them, except in special cases, require iteration to reach their solution. It was decided that a solution algorithm based on the simplex algorithm would be conceptually the simplest to implement. The advantage of simplex in the solution of the trim problem is that it can be treated as a black box which simultaneously solves the optimization problem with more variables than equations and keeps the solution within the variable bounds.

Several sample problems were solved by this method and the results evaluated. Finally, a dynamic survey was performed to give a feel for the interaction of the Shuttle dynamics with the control trim minimization scheme.

2.1 EVALUATION OF TRIM BOUNDARIES

The solution to the trim problem (zero torques and forces) is an indefinite set of control trim placements. In fact there are an infinity of solutions, since there are fewer equations than control variables. However, the unknown

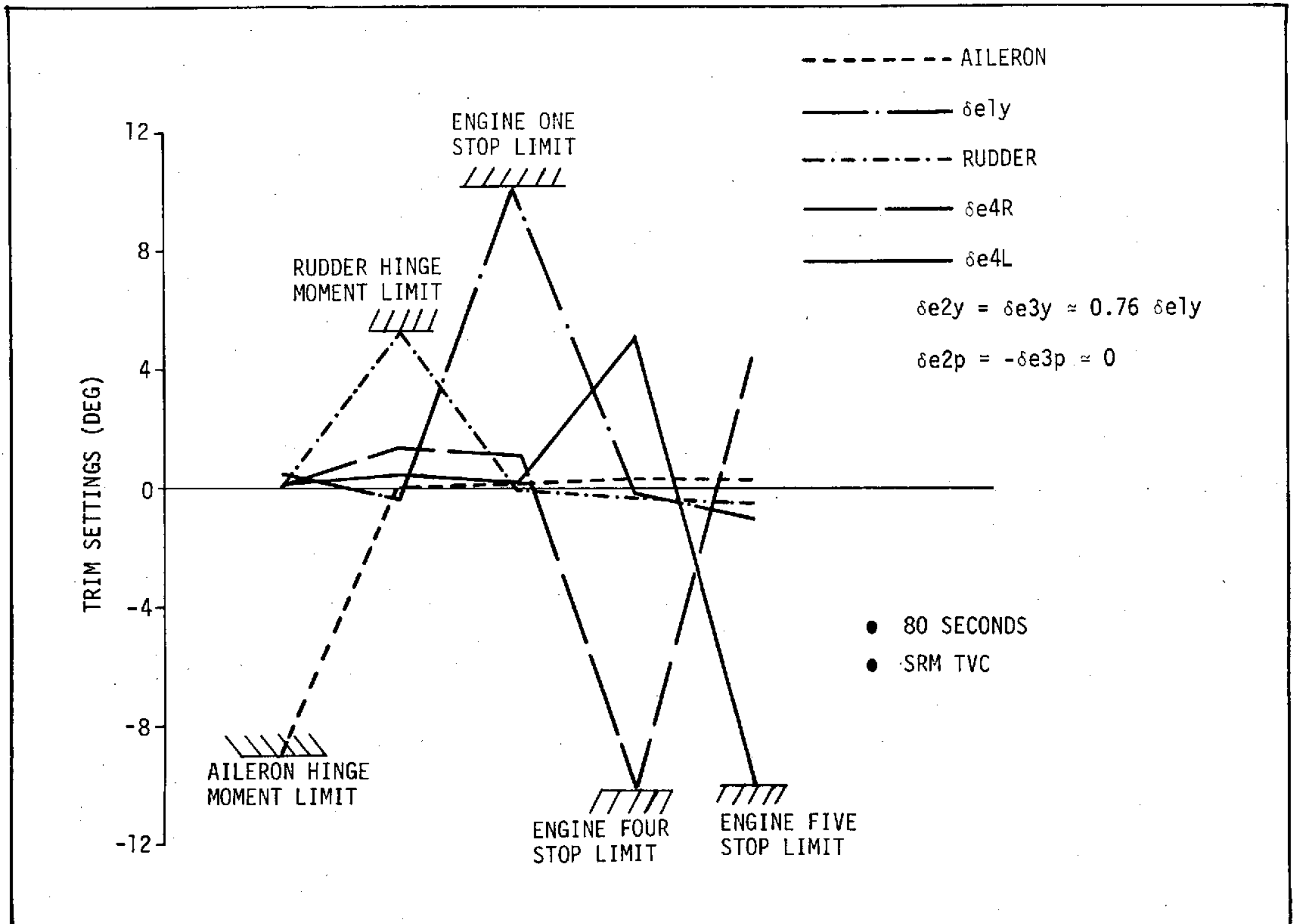


Figure 2-1. CONTROL DEVICES INDIVIDUALLY PLACED ON THEIR LIMITS

control variables are limited in their displacements so the set of possible solutions can be regarded as a bounded, infinite set.

Figure 2-1 shows some of the boundaries of the trim solution as calculated for 80 seconds flight time. This figure gives an indication of the relative trim torques available at 80 seconds flight time. As can be seen, each control device was placed individually on its limit and the other control trim settings were computed. Since the possible trim solutions of 11 control devices form a volume in 11-space it is impossible to show graphically; but the points shown in Figure 2-1 represent points on the boundary of the 11 volume. Since engines two and three do not pitch in the solutions and since they yaw about 75 percent of engine one they are not shown on their limits.

Figure 2-2 shows the minimized control trim settings when each control device was shut out individually. This shows that the vehicle has sufficient control power for trim with any one controller out as long as the SRMs have TVC. It also shows that the aileron has little effect on the trim requirements since deletion of the aileron does not appreciably change.

2.2 TRIM PROBLEMS WITH QUADRATIC COST FUNCTIONS

A quadratic cost functional is a function made up of the sum of the squares of the variables in the function. The partial derivatives of a quadratic cost function are linear making the minimization of the cost a linear problem. Several techniques can be used to solve such problems exactly but the most convenient one for this problem was Lagrangian multipliers.

The Lagrangian multiplier technique will minimize quadratic cost functions, giving an exact solution to the trim problem optimized with respect to the cost. Penalty functions can be devised to minimize such things as drag due to zero surface deflection, thrust gimbal angle, and off-nominal translational accelerations.

The penalty functions can be accounted for in the cost functional. Definition of the cost functional depends upon the particular problem being solved. In general, this functional is a summation of various cost terms which is to

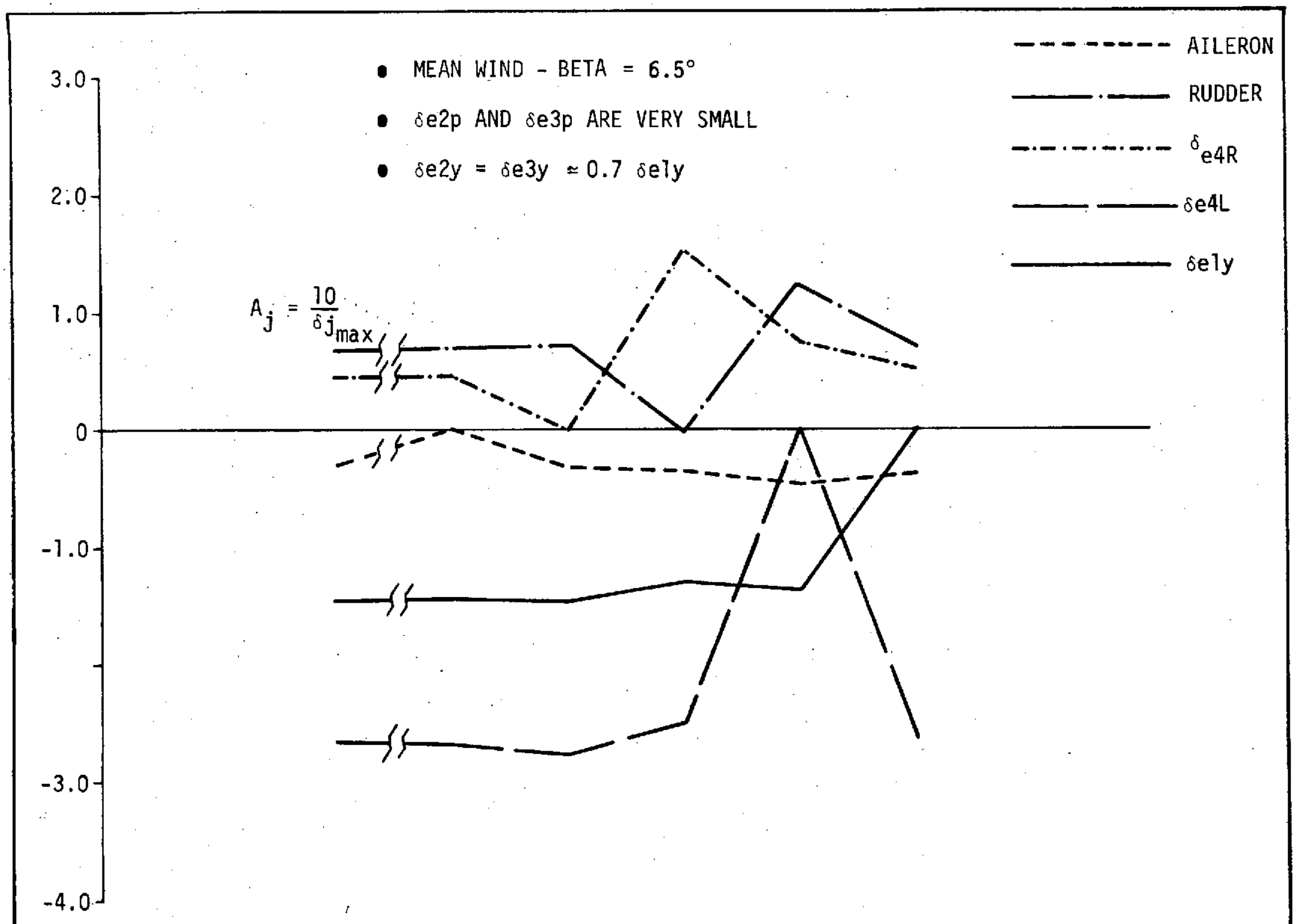


Figure 2-2. CONTROLLABILITY WITH CONTROL DEVICES ZEROED INDIVIDUALLY

be minimized subject to the constraints. The individual terms differ from each other qualitatively and must be weighted in accordance with the requirements of the investigation. An example cost functional, ω, is defined below:

$$\omega = K_1 x(\text{DRAG DUE TO ELEVON DEFLECTION})^2 + K_2 x(\text{THRUST GIMBAL ANGLE})^2 + K_3 x(\text{RUDDER HINGE MOMENT})^2 + K_4 x(\text{VARIATION OF CONTROLLER FROM MEDIAN VALUE})^2 + K_5 x(\text{OFF-NOMINAL TRANSLATIONAL ACCELERATIONS})^2,$$

where the K_i scale factors are selected so that the various terms have the proper relative costs. The K_1 and K_2 terms are included so that the "optimum" trim solution will not produce excess ΔV losses. The K_3 term prevents a trim solution which requires an excessively large hydraulic system for rudder deflections. The K_4 term prevents a trim solution where a control variable value is close to its maximum limit. The K_5 term prevents a solution where the resultant translational accelerations are greatly different from the nominal values. However, the investigator must bear in mind that regardless of the quadratic cost functions devised, it will always reduce to the form

$$\omega = \sum_{j=1}^{m} (a_j \delta_j)^2$$

because the K terms can always be summed with each other if the i^{th} control variable appears in the cost functional more than once. In effect, any study of quadratic cost functionals, no matter how exotic they may be, will always amount to a variation of the a_j coefficients, which are just constants.

Since it was desired to develop a generalized technique for solving trim minimization problems and not to perform a sensitivity study on quadratic cost functions, it was decided to set up and solve a few sample problems to gain further insight into the general trim problem. No attempt was made to represent all possible penalty functions in the cost functional.

The problem was reformulated and Lagrangian multipliers were used to solve in the following manner:

minimize

$$\omega(\delta_j) = \sum_{j=1}^{m} (a_j \delta_j)^2$$

subject to

$$\phi_i = \sum_{j=1}^{n} C_{i_j} \delta_j - T_i = 0$$

$$i = 1, 2, \ldots, n \quad .$$

Let $\quad F = \omega + \sum_{i=1}^{m} \lambda_i \phi_i \quad .$

To minimize F, set

$$\frac{\partial F}{\partial \delta_j} = 0$$

$$j = 1, m \quad .$$

This will yield m equations and m + n unknowns since the λ_i are not explicitly known. Switching to matrix notations:

$$\begin{Bmatrix} \frac{\partial F}{\partial \delta 1} \\ \vdots \\ \frac{\partial F}{\partial \delta_n} \end{Bmatrix} = 2 \begin{bmatrix} a_1^2 & & 0 \\ & \ddots & \\ 0 & & a_n^2 \end{bmatrix} \begin{Bmatrix} \delta 1 \\ \vdots \\ \delta_n \end{Bmatrix} \begin{bmatrix} C_{11} & C_{12} \cdots & C_{1m} \\ C_{21} & C_{22} \ddots & \\ \vdots & & \\ C_{n1} & & C_{nm} \end{bmatrix} \begin{Bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{Bmatrix} \tag{1}$$

define Ω such that

$$\Omega^{-1} = \begin{bmatrix} a_1^2 & 0 \\ 0 & a_m^2 \end{bmatrix}$$

let

$$C = \begin{bmatrix} C_{11} & C_{12} \cdots & C_{1m} \\ C_{21} & & \\ \vdots & & \\ C_{n1} & & C_{nm} \end{bmatrix} \qquad \{\delta\} = \begin{Bmatrix} \delta_i \\ \vdots \\ \delta_n \end{Bmatrix}$$

$$T_c = \begin{Bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{Bmatrix} \qquad \lambda = \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{Bmatrix} \quad \text{and} \quad \phi = \begin{Bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{Bmatrix}$$

returning to Equation (1) we can solve for δ

$$\delta = -\frac{1}{2}\,\Omega^{-1}\,C^{T}\,\lambda \tag{2}$$

also

$$\phi = C\,\delta - T_c = 0 \tag{3}$$

eliminating λ

$$\{\delta\} = [\Omega]\;[C]^T\;\{[C]\;[\Omega]\;[C]^T\}^{-1}\;\{T_c\} \tag{4}$$

We have solved for δ in terms of the C matrix, the penalty matrix Ω, and the torque commands T_c.

An immediate result of using three constraint equations (yaw moment = roll moment = side force = 0) is that no practical minimum solution was obtainable. That is, the control settings are greater than the bounds. The zero side force requirement was the cause of this. When the requirement was dropped the problem reduced to the following:

minimize

$$\omega = (a_1\delta_a)^2+(a_2\delta_R)^2+(a_3\delta_{eiy})^2+(a_4\delta_{e2p})^2+(a_5\delta_{e2y})^2 + (a_6\delta_{e3p})^2+(a_7\delta_{e3y})^2 \tag{5}$$

subject to

$$\phi_1 = C_{\ell_{\delta a}}\,\delta_a + C_{\ell_{\delta R}}\,\delta R + C_{\ell_{\delta e1y}}\,\delta_{e1y} + C_{\ell_{\delta e2p}}\,\delta_{e2p} + C_{\ell_{\delta e2y}}\,\delta_{e2y} + C_{\ell_{\delta e3p}}\,\delta_{e3p} + C_{\ell_{\delta e3y}}\,\delta_{e3y} - (-C_{\ell_\beta}\,\beta) = 0 \tag{6}$$

$$\phi_2 = C_{n_{\delta a}} a + C_{n_{\delta R}} \delta R + C_{n_{\delta e1y}} \delta e1y + C_{n_{\delta e2p}} \delta e2p + C_{n_{\delta e2y}} \delta e2y + C_{n_{\delta e3p}} \delta e3p + C_{n_{\delta e3y}} \delta e\ y - (-C_{n_\beta}\ \beta) = 0 \qquad (7)$$

where β comes from the mean wind. The a_j coefficients can be set to any value so that varying them will represent objective functions that can minimize control deflections, hinge moments, thrust losses due to drag and engine displacements. This technique cannot automatically handle the limits on the control devices so an iteration procedure was devised to solve for δ and then check for excessive values. If they occurred the penalty coefficients were systematically varied to see if the solution could be driven within the physical bounds.

Using the 049 data, the overriding problem near max Q (75-80) seconds is controllability. By choosing the penalty coefficients according to the following formula a solution that minimizes control deflections can be obtained.

$$a_j = 10/\text{minimum } (|\delta_{j_{max}}| \text{ or } |\delta_{j_{min}}|) \ .$$

The philosophy here is that the control devices that are most severely limited will be most heavily weighted. Unfortunately, the solution to this problem, in the neighborhood of max Q, yielded a rudder setting approximately two times as large as the control power can furnish (hinge moment limited). The aileron was also at or near the limit at this time. Systematically varying the penalty coefficients to drive these two devices within the control limits resulted in exceeding the control limits on all of the controllers. While this is not formal proof that there is no solution at these points it is very strong evidence for this conclusion.

Additional insight into the problem can be gained by showing that (for the 049 data) engines 2 and 3 are not independent of each other if there is no reason to weigh one's displacement more heavily than the other.

From the data the following holds:

$$C_{\ell_{\delta e2p}} = C_{\ell_{\delta e3p}}$$

$$C_{n_{\delta e2p}} = C_{n_{\delta e3p}}$$

$$C_{\ell_{\delta e2y}} = C_{\ell_{\delta e3y}}$$

$$C_{n_{\delta e2y}} = C_{n_{\delta e3y}}$$

$$\begin{Bmatrix} \delta_a \\ \delta_R \\ \delta_{e1y} \\ \delta_{e2p} \\ \delta_{e2y} \\ \delta_{e3p} \\ \delta_{e3y} \end{Bmatrix} = \begin{bmatrix} \Omega \end{bmatrix} \begin{bmatrix} C_{\ell_{\delta a}} & C_{n_{\delta a}} \\ C_{\ell_{\delta R}} & C_{n_{\delta R}} \\ C_{\ell_{\delta e1y}} & C_{n_{\delta e1y}} \\ C_{\ell_{\delta_{e2p}}} & C_{n_{\delta_{e2p}}} \\ C_{\ell_{\delta e2y}} & C_{n_{\delta e2y}} \\ C_{\ell_{\delta e2p}} & C_{n_{\delta e2p}} \\ C_{\ell_{\delta e2y}} & C_{n_{\delta e2y}} \end{bmatrix} \begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix}$$

where $A = \left(\begin{bmatrix} C \end{bmatrix} \begin{bmatrix} \Omega \end{bmatrix} \begin{bmatrix} C \end{bmatrix}^t \right)^{-1}$

$$\begin{bmatrix} C \end{bmatrix} \begin{bmatrix} \Omega \end{bmatrix} \begin{bmatrix} C \end{bmatrix}^t = 2 \begin{bmatrix} \Sigma\Omega_{jj} C_{\ell_{\delta j}}^{\;2} & \Sigma\Omega_{jj} C_{\ell_{\delta j}} C_{n_{\delta j}} \\ \Sigma\Omega_{jj} C_{\ell_{\delta j}} C_{m_{\delta j}} & \Sigma\Omega_{jj} C_{n_{\delta j}}^{\;2} \end{bmatrix}$$

$$\begin{bmatrix} A \end{bmatrix} = \left\{ \begin{bmatrix} C \end{bmatrix} \begin{bmatrix} \Omega \end{bmatrix} \begin{bmatrix} C \end{bmatrix}^t \right\}^{-1} = \frac{\begin{bmatrix} \Sigma_j \Omega_{jj} C_{\ell_{\delta j}} & - \Sigma\Omega_{jj} C_{\ell_{\delta j}} C_{n_{\delta j}} \\ - \Sigma\Omega_{jj} C_{\ell_{\delta_j}} C_{n_{\delta j}} & \Sigma\Omega_{jj} C_{n_{\delta j}}^{\;2} \end{bmatrix}}{2 \left(\Sigma\Omega_{jj}^{\;2} C_{\ell_{\delta j}} \Sigma C_{n_{\delta j}}^{\;2} - \left(\Sigma\Omega_{jj} C_{\ell_{\delta j}} C_{n_{\delta j}} \right)^2 \right)}$$

$$[A]\begin{Bmatrix} T_{c_1} \\ T_{c_2} \end{Bmatrix} = \begin{Bmatrix} \left(A_{11}\, T_{c_1} + A_{12}\, T_{c_2}\right) \\ \left(A_{12}\, T_{c_1} + A_{11}\, T_{c_2}\right) \end{Bmatrix} = \begin{Bmatrix} \alpha \\ \gamma \end{Bmatrix}$$

and if Ω_{44} and Ω_{66} and $\Omega_{55} = \Omega_{77}$ (equal weights on engines 2 and 3).

Equation (4) can be written as

$$\begin{Bmatrix} \delta_a \\ \delta_R \\ \delta_{e1y} \\ \delta_{e2p} \\ \delta_{e2y} \\ \delta_{e3p} \\ \delta_{e3y} \end{Bmatrix} = [\Omega][C]^T\,[A]\begin{Bmatrix} T_{c_1} \\ T_{c_2} \end{Bmatrix} = \begin{Bmatrix} \Omega_{11}\left(C_{\ell_{\delta a}}\alpha + C_{n_{\delta a}}\gamma\right) \\ \Omega_{22}\left(C_{\ell_{\delta R}}\alpha + C_{n_{\delta R}}\gamma\right) \\ \Omega_{33}\left(C_{\ell_{\delta e1y}}\alpha + C_{n_{\delta e1y}}\gamma\right) \\ \Omega_{44}\left(C_{\ell_{\delta e2p}}\alpha + C_{n_{\delta e2p}}\gamma\right) \\ \Omega_{55}\left(C_{\ell_{\delta e2y}}\alpha + C_{n_{\delta e2y}}\gamma\right) \\ \Omega_{44}\left(-C_{\ell_{\delta e2p}}\alpha - C_{n_{\delta e2p}}\gamma\right) \\ \Omega_{55}\left(C_{\ell_{\delta e2y}}\alpha + C_{n_{\delta e2y}}\gamma\right) \end{Bmatrix} \qquad (8)$$

$$\delta e2p = -\delta e3p = \Omega_{44}\left(C_{\ell_{\delta e2p}}\alpha + C_{n_{\delta e2p}}\gamma\right)$$

$$\delta e2y = \delta e3y = \Omega_{55}\left(C_{\ell_{\delta e2y}}\alpha + C_{n_{\delta e2y}}\gamma\right)$$

By a similar approach it can be shown that engine 1 is not independent of engines 2 and 3 if its penalty coefficient is the same as 2 and 3. The following ratios hold:

$$C_{\ell e1y} = C_{\ell e2y}\,\frac{(Z_{e_1}+Z_{c_g})\cos P_{e_1}}{(Z_{e_2}+Z_{c_g})\cos P_{e_2}\cos Y_{e_2}} \qquad (9)$$

$$C_{n_{\delta e1y}} = C_{n_{\delta e2y}} \frac{\cos P_{e1}}{\cos P_{e2} \cos Y_{e2}} \tag{10}$$

where Z_{e1} is the Z location of engine one, P_{e1} is the pitch angle of engine one and Y_{e2} is the yaw angle of engine 2. The result is:

$$\delta_{e1y} = \left[\frac{\frac{Z_{e1} Z_{cg}}{(Z_{e2}+Z_{cg})} \frac{\cos P_{e1}}{\cos P_{e2} \cos Y_{e2}} \alpha C_{\ell_{\delta e2y}} + \frac{\cos P_{e1}}{\cos P_{e2} \cos Y_{e2}} \gamma C_{n_{\delta e2y}}}{C_{\ell e2y} + C_{n_{\delta e2y}}} \right] \delta e2y \tag{11}$$

The relationship between δe1y and δe2y is not a direct ratio because Δ and γ are functions of T_1 and T_2, which are the disturbance torques. If the relative value of the disturbance torques change with time, or a different kind of disturbance is used, the ratio between δe1y and δe2y will also change.

Similarly it can be shown that the left and right ailerons operate as mirror images of each other. Since it is difficult to imagine a case where it would be advantageous to operate them independently they have not been separated in any of the work done. However, the orbiter engines have all been operated both in pairs and independently in optimization procedure to date and the results have verified the developments in the previous paragraphs.

Figure 2-3 shows the results of minimizing the control trim displacements at selected time intervals between the flight times of 25 and 90 seconds. As can be seen, the trim settings exceeded the hinge moment limits on the rudder and, to a lesser degree, on the aileron near maximum dynamic pressure. The severity of the controllability problem is illustrated in Figure 2-4. Since the rudder and the aileron exceed the hinge moment limits at 75 seconds, this time point was chosen as a test of the penalty coefficients iteration scheme. If a control device, say δ_j, exceeded its maximum absolute value then its new penalty coefficient (a_j') was reset by the following formula:

$$a_j' = a_j \sqrt{\frac{|\delta j|}{\min[|\delta j_{max}, |\delta j_{min}|]}} \tag{12}$$

where min [ ] means the minimum of the quantity in brackets.

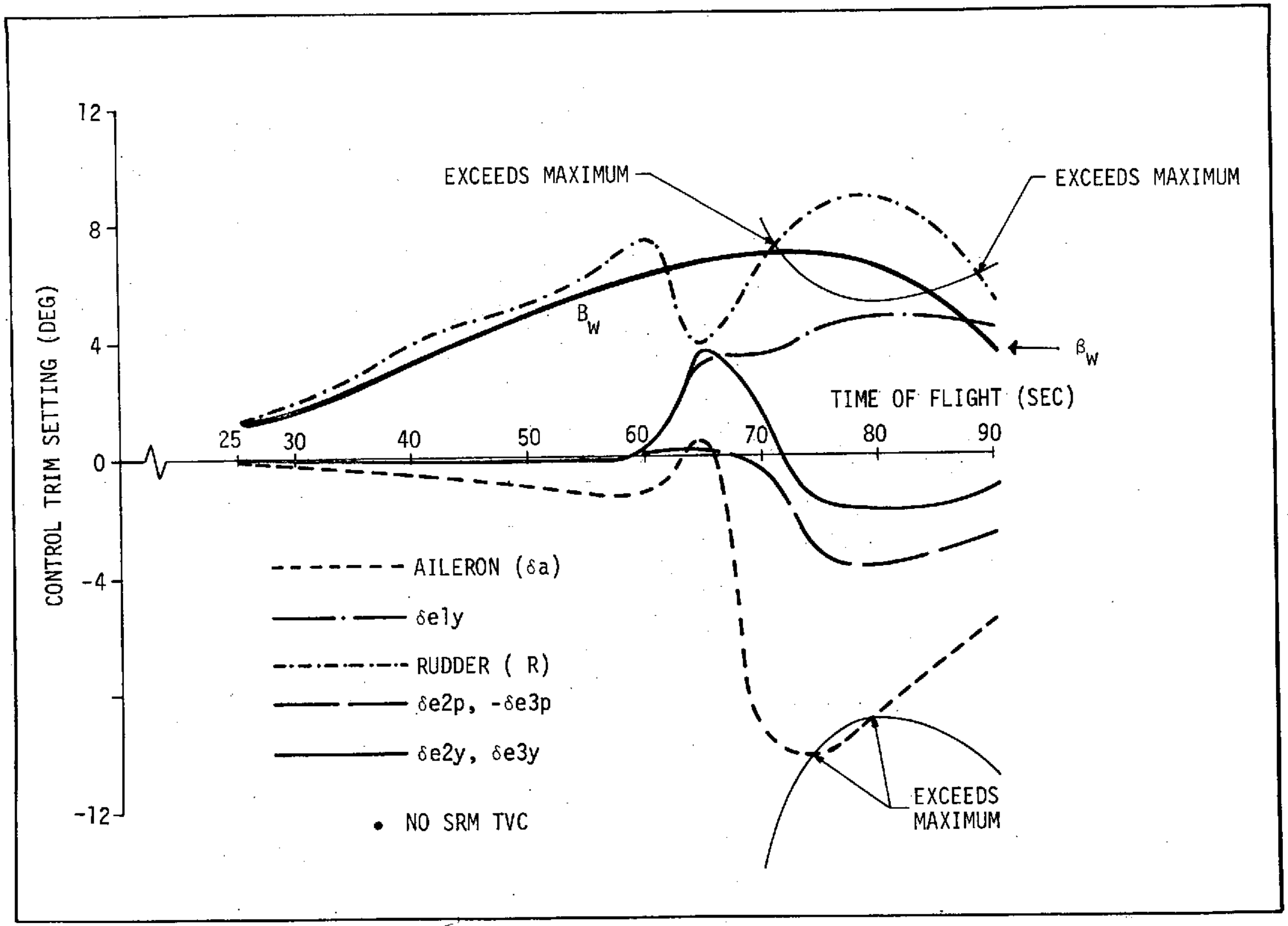


Figure 2-3. QUADRATIC COST TRIM SOLUTIONS VERSUS TIME

Figure 2-4 is a plot of the iteration steps the computer program followed at 75 seconds flight time. Iteration forced the rudder down near its maximum but also forced the engines to exceed their limits. The program failed to find a realizable solution, indicating the magnitude of the controllability problem at this time point (with only a mean wind disturbing function). As previously stated, this does not prove that the trim torques are nonrealizable but points strongly to the possibility.

To alleviate the controllability problem the solid rocket motors (SRM) were allowed to gimbal (TVC). Again the engines acted as a pair as long as the penalty coefficients on the engines were equal. As Figure 2-5 shows, the solid engines relieved the controllability problem. The peak rudder deflection of 3.0 degrees was only about 13 percent of the maximum rudder deflection (hinge moment limited) at that time. The peak solid engine deflection was about 27 percent of the maximum (10 degrees) at 70 and 80 seconds.

The conclusions drawn from the Lagrangian approach to trim optimization are as follows:

- Controllability is a problem on the 049 even with the aerodynamic surfaces, if SRM TVC is not used.
- If SRB TVC is used controllability is no longer a problem.
- The aileron is relatively ineffective in trimming the vehicle.
- If there are no reasons to weight them differently, engines two and three yaw equally and pitch oppositely. The solid engines pair also with the right engine displaced right and down and left engine right and up.
- Trimming the side force is probably impractical.

In summary, the Lagrangian approach gives insight into the problem and an exact solution to the quadratic cost problem. However, if the cost can not be adequately represented by a quadratic the solution will be useless. Furthermore, the procedure does not guarantee a solution within the variable bounds nor does it indicate if a feasible solution is possible.

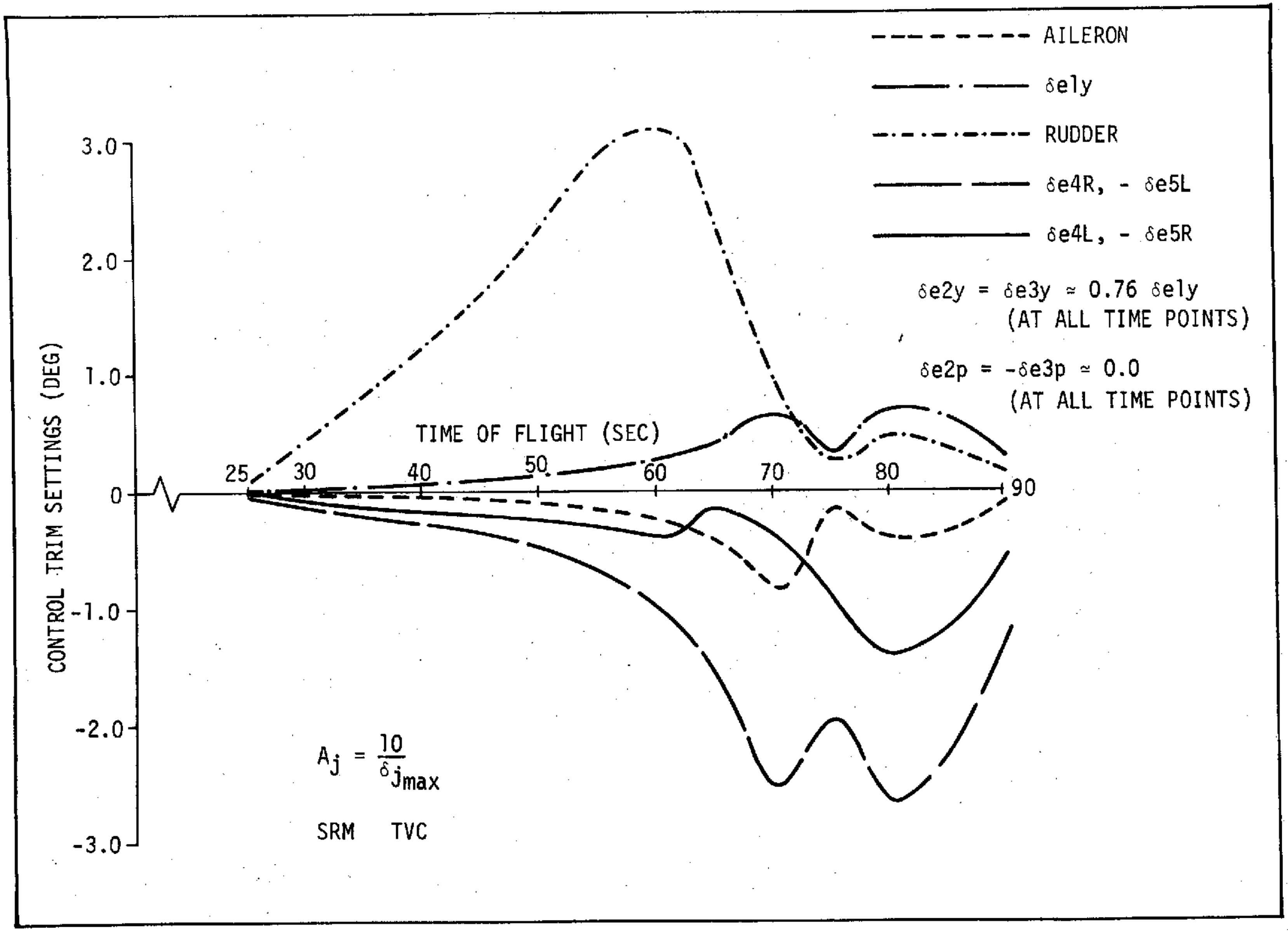


Figure 2-5. QUADRATIC COST TRIM SOLUTIONS VERSUS TIME - SRM TVC

2.3 BOUNDED TRIM PROBLEMS WITH GENERAL COST FUNCTIONS

The trim problems investigated in this study had control variable bounds that typically limited their solutions. In the first part of this study it was necessary to obtain a solution and then check to see if the solution was physically realizable, that is within the variable bounds. Furthermore if the actual cost function was not quadratic then the Lagrangian multiplier technique could not be used at all.

A solution process analogous to the Lagrangian multiplier technique can be used to solve this class of problems. However the details of the solution are involved and the solution process does not provide insight into the problem. A brief description of this approach is included here for clarity.

If the cost functional is differentiable, but the derivatives are non-linear, setting the partials of the cost with respect to the control variables to zero yields a set of nonlinear homogeneous equations which can be solved by iteration (if the partials can be written analytically). If the partials can only be represented by graphs or tables the mechanics of the solution are much stickier.

Handling the inequality constraints adds more complexity to the problem. This requires a transformation of variable and definition of n additional variables linearly related to the original n variables.

Since one of the goals of this study was to develop solution concepts, it was felt this brute force approach would not totally fulfill the work statement. Therefore, the following linear programming technique utilizing simplex was devised.

The idea behind this approach was that the simplex algorithm can be used to force an iteration procedure to converge in the vicinity of the lowest cost. During the convergence procedure simplex will also require the controller bounds to be observed. The cost functional in a simplex problem may be recognized as a summation of partial derivatives. Conceptually it is very simple to place any form of partials in the simplex cost row and the simplex solution that results will be as good as the values of the partials over which the iteration ranges. If this range is held small enough the solution will approximate the minimum cost on that range.

The primary difference between a linear programming approach to the problem and the Lagrangian multipler technique is that the control limits are considered in the solution by linear programming. In other words linear programming will produce a feasible solution in proximity to the optimal solution. If the control limits are high enough to not impact the solution both methods will give approximately the same answer.

As stated previously the ultimate goal of this study was the development of concepts and techniques to solve the trim problem. The solution to the general trim problem is actually an algorithm developed around the simplex technique. Therefore this algorithm is the primary result of the study. The central philosophy of this algorithm is to use the simplex procedure (ref. 1) to solve the problem of minimizing a general cost function, ω, over a small range of the variables' permitted values.

2.3.1 Description of Solution Algorithm

The mathematical basis for the solution algorithm rests on linearizing the cost function, $\omega(\delta_i)$, about a feasible solution, $\{\delta_{i_o}\}$. The cost can be written as follows:

$$\omega(\delta_i) = \omega(\delta_{i_o}) + \sum_{i=1}^{n} \frac{\partial\omega}{\partial\delta_i} (\delta_i - \delta_{i_o}) \quad .$$

The problem is to find a way to minimize $\omega(\delta_i)$ (which is always positive of course). This problem is the same as making

$$\sum_{i=1}^{n} \frac{\partial\omega}{\partial\delta_i} (\delta_i - \delta_{i_o})$$

the largest negative value possible. As will be shown later, Simplex will solve the latter problem, but an iteration algorithm must be used to find the minimum $\omega(\delta_i)$ because it is nonlinear in the general case.

The algorithm developed to solve the trim problem can be described in simple terms. The simplex algorithm itself is described in Appendix B. After the problem is defined the variables are transformed so that they are all non-negative, which is a requirement of simplex. Then, if the neighborhood of the solution is known beforehand, an approximate solution is used as a starting point. If nothing is known about the solution the entire range of the variables can be allowed in the starting solution. In either case a feasible (meets the trim requirements and variable bounds) solution is calculated to start the procedure. Since simplex utilizes a cost row made up of linear costs, this row must be filled with the partial derivatives of the cost function with respect to the appropriate variables. The simplex procedure will solve for the minimum cost solution based on the linearized costs and keep the solution within the bounds of the variable limits. Of course the solution is only as good as the linear approximations to the cost partials. To converge on the correct solution temporary bounds, which are always within the real bounds, are created about the feasible solution. These bounds are set using a range percentage, delta, about the feasible solution. After the simplex solution is calculated, using the temporary bounds, the actual total cost, ω, is computed. As long as the new total cost improves the range percentage remains the same. The problem is set up and solved again, using the last computed solution as a starting point about which the new bounds are set and at which the new linearized partial costs are computed. If the total cost does not improve delta is halved and the new solution is calculated. When the range percentage becomes small enough the simplex solution will be approximately the exact solution. At some improvement criteria the process is terminated. A simplified flow chart of the procedure is shown in Figure 2-6.

2.3.2 Formulation of Simplex Problem

The classical linear programming problem is as follows:

Find $X^j \geq 0$ such that (13)

$$Z = \sum_{j=1}^{n} a_j^o x^j \text{ is minimum and} \tag{14}$$

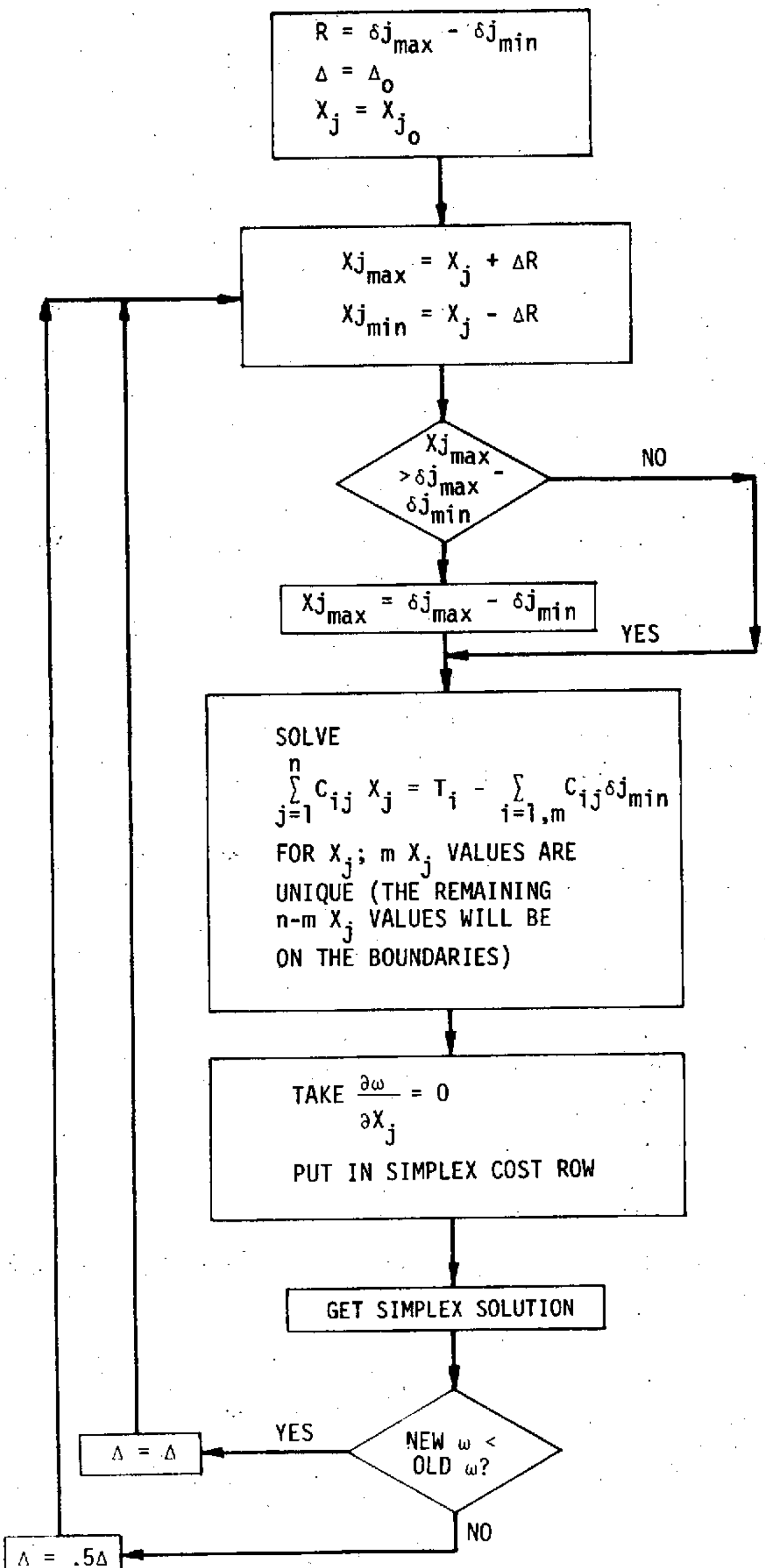


Figure 2-6. FLOW CHART OF SOLUTION ALGORITHM

$$\sum_{j=1}^{n} a_j^i X^j \leq b^i \quad (i = 1, \ldots, m; m < n) \tag{15}$$

The x^j are our control trim settings, Z is the objective function and Equation (15) represents the zero torque requirements and the control limits.

The formal statement of the linear programming problem indicates that only positive values of the problem variables are allowed. Furthermore, the cost function, Z, is a linear summation of the products of the problem variables and their constant multipliers. Since any practical control trim problem may require some negative control settings, a transformation of variables was performed on the problem variables.

If we choose $X_j = \delta_j - \delta_{j_{min}}$

then since $\delta_{j_{min}} \leq \delta_j \leq \delta_{j_{max}}$

$$0 \leq X_j \leq \delta_{j_{max}} - \delta_{j_{min}}$$

and since $\delta_{j_{max}} - \delta_{j_{min}} \geq 0$ (16)

the requirement on X_j is consistent.

In the notation of Equations (13) through (15) the problem has m constraints and n variables. From experience the algorithm will solve the problem in m to 3m steps (ref. 2). However, in the trim optimization problem simplex has to be applied iteratively until the cost reaches a near minimum. If the original solution was allowed to range over the whole boundary, the investigator's experience showed simplex would be applied as many as 20 times. If the solution is approximately known beforehand, the number of iterations was reduced sharply. One way to predict the approximate solution is to use the solution from the previous nearby time steps.

The cost functional can be any piecewise differentiable function. A description of the computer program used to solve the trim problem is in

Appendix C. The cost functional is represented by two simple subroutines so that if it needs to be changed in form (not just coefficients) it can be readily reprogrammed.

For purposes of testing the validity of this approach as well as obtaining useful information, a cost function was defined to represent the drag losses due to aerodynamic surface deflections and the thrust losses due to vectoring the engines off the centerline. The drag losses are represented as typical linear aerodynamic terms which are functions of the absolute aero surface displacement. The thrust losses are calculated by subtracting the cosine of each engines' circular displacement from unity and multiplying by its thrust. Since each engine is represented in the control equations by a pitch and a yaw, the circular displacement is the square root of the sum of the squares of the pitch and yaw displacements. The linear programming procedure will handle any piecewise differentiable cost function but this cost function in Equation (16) was chosen in the bulk of the study

$$\omega = QS_{ref} \sum_{i=1}^{2} C_{D_{\delta_i}} |\delta_i| + \sum_{i=3,5,7,9...n} T_i \left[1 - \cos(\delta_i^2 + \delta_{i+1}^2)\right]^{1/2} \quad (16)$$

The simplex costs are defined as

$$Z = \left.\frac{\partial \omega}{\partial \delta_i}\right|_{\delta_{i\,solution}}$$

or

$$Z = \left[qS_{ref}\, C_{D_{\delta_i}} \frac{\delta_i}{|\delta_i|}\right]_{i=1,2} + \left[T_i \frac{\sin\sqrt{\delta_i^2 + \delta_{i+1}^2}\ \delta_i}{\delta_i^2 + \delta_{i+1}^2}\right]_{i=3,5,7,9\ ...}$$

$$+ T_i \frac{\sin\sqrt{\delta_i^2 + \delta_{i-1}^2}}{\delta_i^2 + \delta_{i-1}^2}\ \delta_i \qquad i = 4,6,8,10,\ ... \quad (17)$$

The rudder drag coefficient was set to 0.7×10^{-3} nt/deg, a value that was estimated from looking at Shuttle rudder data for a similar configuration. For convenience, since there was no aileron drag data available, the aileron drag coefficient was set to that value also.

2.3.3 Minimum Thrust/Drag Losses Withoug SRM TVC

Figure 2-7 shows the minimum thrust, minimum drag results for 50 to 90 seconds for the 049 configuration. The disturbance is the same as that used before, the mean wind. This plot shows the trim results when the solid engines are not used. The "infeasible" region is the time between 74 and 85 seconds. During that period trim requirements cannot be fulfilled with only the orbiter engines and aerodynamic surfaces.

At around 67 seconds the aileron began to deflect. It has not been visible up to that time because its cost was high relative to its control effectiveness. It was visible from that time on because it was necessary to use it to meet the trim constraints, regardless of cost. The limitations on the trim power were caused primarily by the rudder hinge moment limits, which are shown on the graph. It can be seen that the rudder reached its hinge moment bound about three seconds before the control limits were exceeded. At that time the other controls began to increase rapidly toward their bounds as the rudder was squeezed by its hinge moment limit. The aileron reached its hinge moment limit about one second before the infeasible region began. The yaw deflection of engine one increased toward its bound and probably reached it in the infeasible region. On the other side of the infeasible region (85 seconds) something similar occurred. The aileron followed its negative hinge moment bound for a second and then dipped rapidly to zero. The rudder remained exactly on the positive hinge moment limit while engine one yaw deflection decreased.

In the region between 60 and 70 seconds Mach 1 occurs, causing sharp variations in the aerodynamic coefficients. In that time span the rudder decreased while the yaw displacement of engines two and three increased. This appears to be the only consequence of Mach 1 on the control settings.

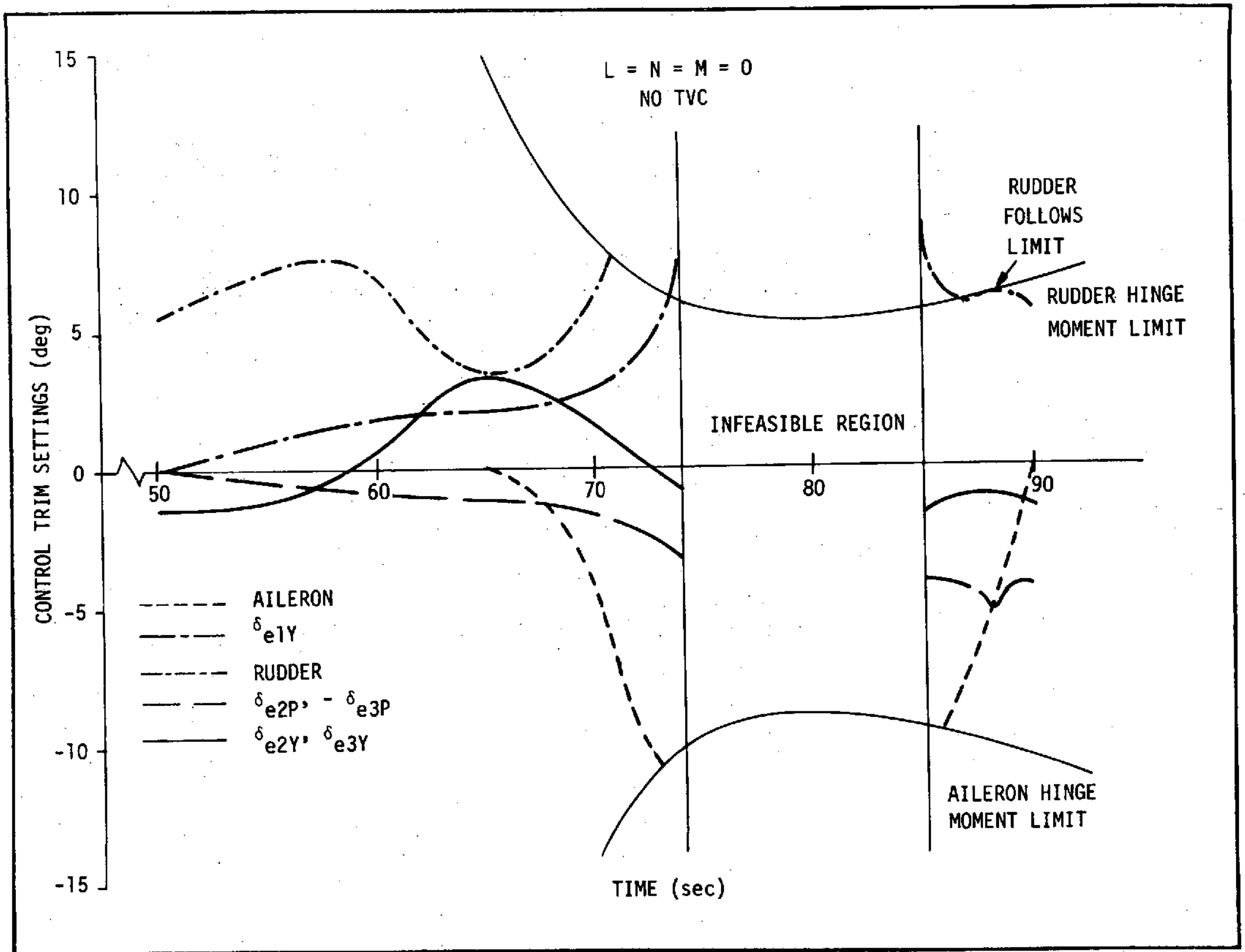


Figure 2-7. MINIMUM THRUST/DRAG LOSSES SOLUTION - NO TVC

2.3.4 Control Variable Proximity Penalty Functions

To show the feasibility of using penalty functions to prevent placing the control devices on their limits, a penalty function was devised as shown below:

$$\omega = \bar{q}S_{ref} \sum_{i=1}^{2} C_{D_{\delta_i}} |\delta_i| + \sum_{i=3,5,7,9,\ldots n} T_i\left[1 - \cos(\delta_i^2 + \delta_{i+1}^2)\right] + \sum_{i=1}^{n} \left(\frac{2\delta_i - \delta_{i_{max}} - \delta_{i_{min}}}{(\delta_i - \delta_{i_{max}})(\delta_i - \delta_{i_{min}})} \right) \tag{18}$$

The last term in the cost function represents a penalty for approaching the variable bounds. This was modified in the "COST" and "SCOST" subroutines (Appendix B) and the solution was obtained for the no SRM TVC case at 71 seconds. The table below compares these results with the pure thrust/drag minimum results from Figure 2-7.

	δa	δR	δe1	δe3p	δe2y	(Nt) Loss	
Thrust/Drag Minimum	-6.9	7.5+	3.14	2.1	.66	74000	δe2p = -δe3p
Proximity Penalty	-6.2	6.7	4.75	2.1	.86	74500	δe3y = δe2y

+means on boundary (hinge moment limited)

The penalty function forces the rudder off its limit, but consequently it increases displacement of the engines (δe1y and δe2y). It is clear that the altered penalty function should cause more thrust/drag loss and the table shows that it does. Or course a proximity penalty function does not increase the vehicle's trim power, but it does keep the control devices off their bounds until the boundary value(s) is absolutely required for trim.

2.3.5 Minimum Thrust/Drag Losses With Added Control Power

Figure 2-8 represents the results of minimizing the thrust and drag losses (subject to the mean wind) with SRM TVC in the control loop. As can be seen, the aileron did not deflect at all and the rudder only deflected between 60 and 70 seconds. This seems to contradict the results on the previous graph since on that plot the rudder dipped at Mach 1 and the engines increased. However,

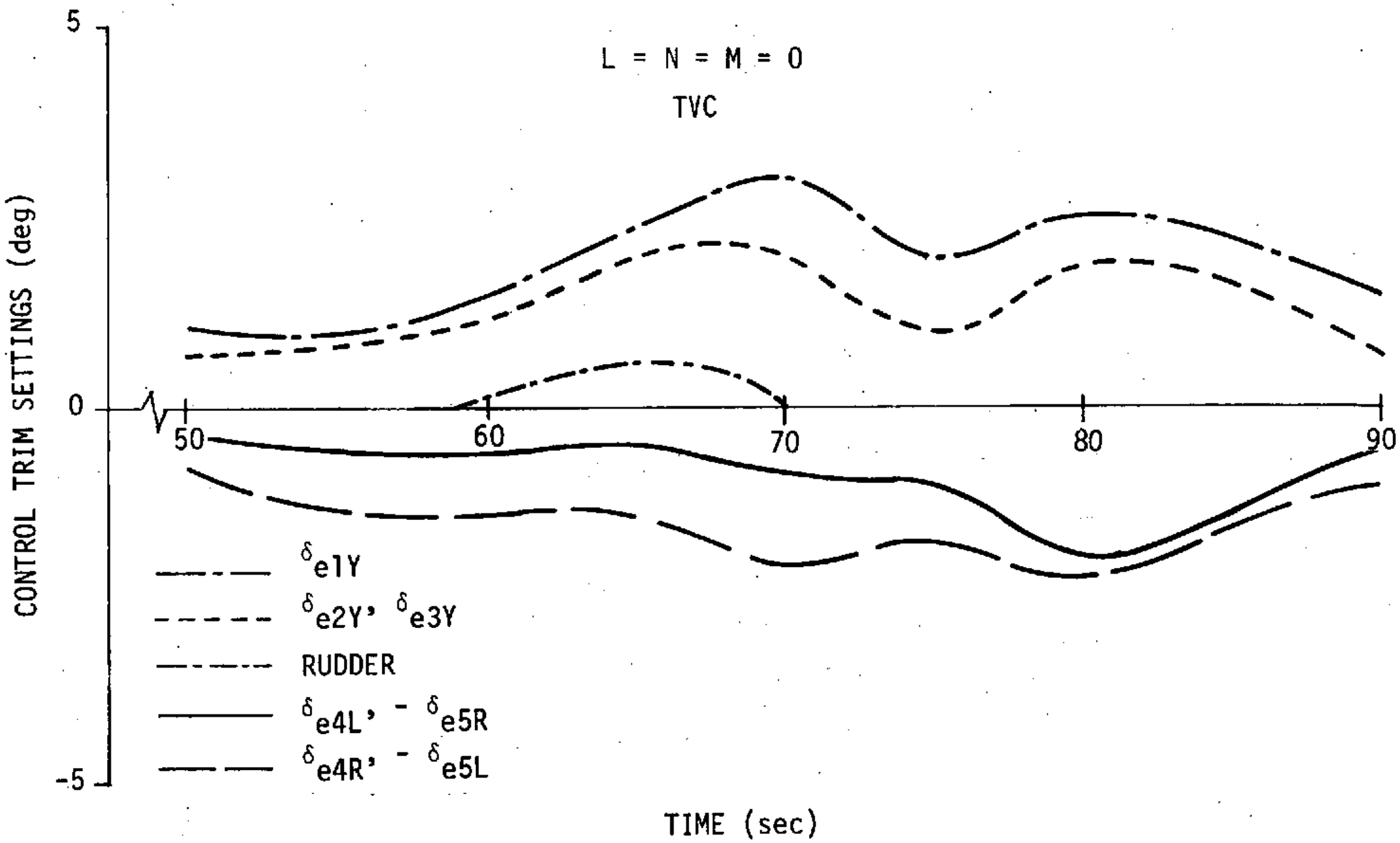


Figure 2-8. MINIMUM THRUST-DRAG LOSSES SOLUTION WITH SRM TVC

the costs of the rudder and engines are dissimilar functions. In fact, the derivatives of the engine costs are functions of the engine settings so that in one region the partial with respect to rudder may be larger than the engine partials, and in another region the partials with respect to engines may predominate. The solid engines are utilized at around 1-2 degrees deflection, a region in which their thrust loss is very small. So simplex selected the rudder to use in place of increasing the solid engines and causing a larger thrust loss.

Figure 2-9 shows the time history of the vehicle with beefed up hydraulics. (The aileron and rudder hinge moment limits removed) as can be seen, the rudder peaked at 12.8 degrees, at 80 seconds. At that time the hinge moment limit is actually 5.23 degrees. By allowing the rudder to reach the values shown on the graph, the aileron was not required to exceed its nominal hinge moment limit. Therefore, the aileron's hydraulic power would not require beefing up if the rudder's hydraulic power were unlimited. The character of the curve up to 90 seconds is the same as Figure 2-7, since the hinge moment limits do not impact the solution up to that point.

Figure 2-10 is a plot of the thrust-drag losses versus time for the configuration with and without TVC and with beefed up aero surface hydraulics to give unlimited hinge moments on the aileron and rudder (stop limits remain unchanged). The thrust-drag losses were lowest with TVC and highest with the current aero surface hydraulics and no TVC. Of course this plot does not reflect the weight increase required to hydraulically control the solid engines. Nor does the chart show the weight cost of beefing up the aero surface hydraulics. As might be expected, the losses were highest when the control power approached its limits, on the no TVC case. It is easy to see that when the control settings are dictated almost entirely by the trim requirements, little can be done to minimize the cost function. Specifically, the high costs occurred because the rudder reached its hinge moment limit and the orbiter engines (primarily δ_{e1Y}) were forced to take up the time slack. As long as the orbiter engines were not required to vector more than two or three degrees their thrust loss was very small (see Equation (16)). When they were forced outside this range their thrust loss became significant. Furthermore the

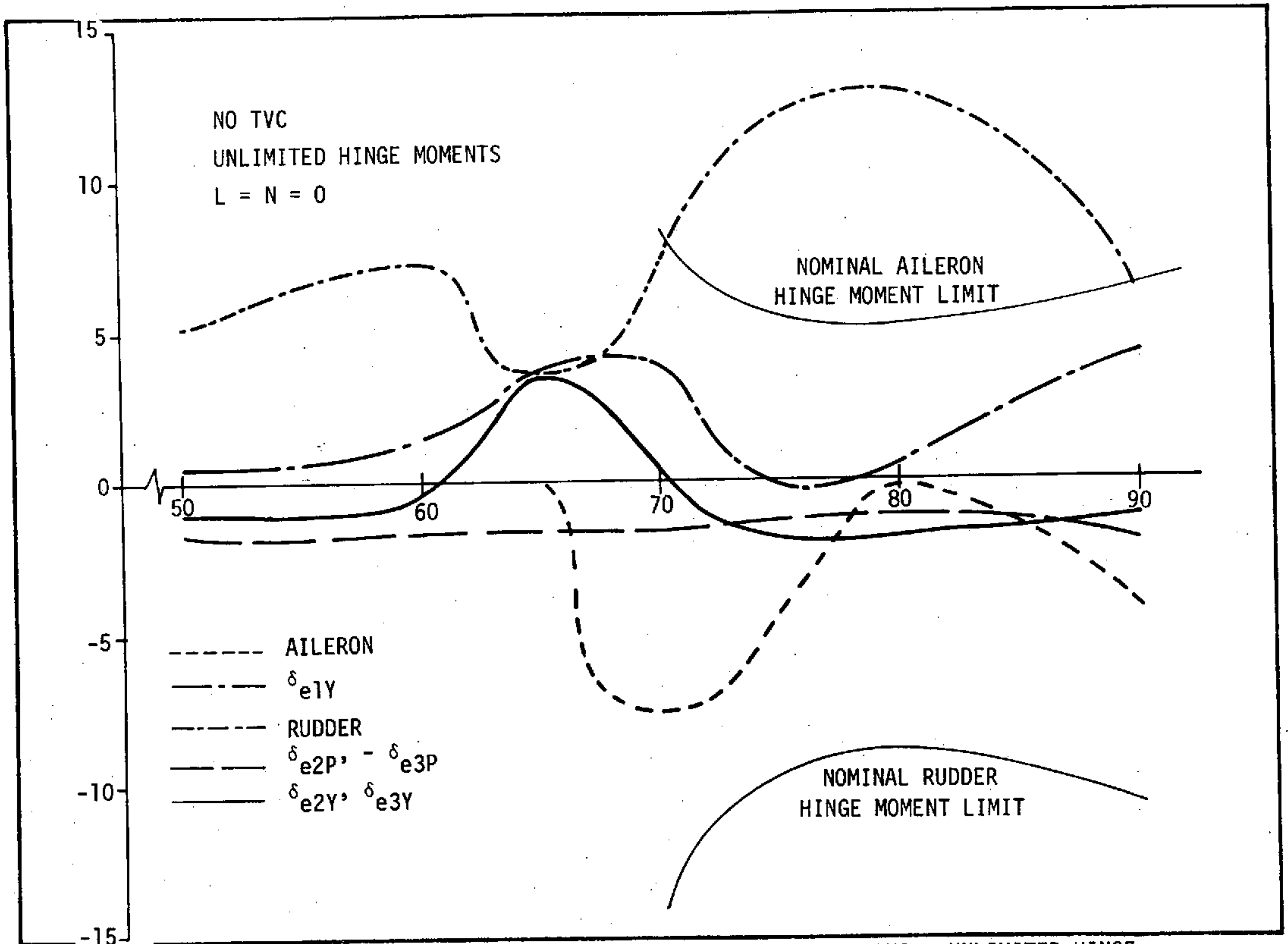


Figure 2-9. MINIMUM THRUST-DRAG LOSS SOLUTIONS - NO SRM TVC - UNLIMITED HINGE MOMENTS

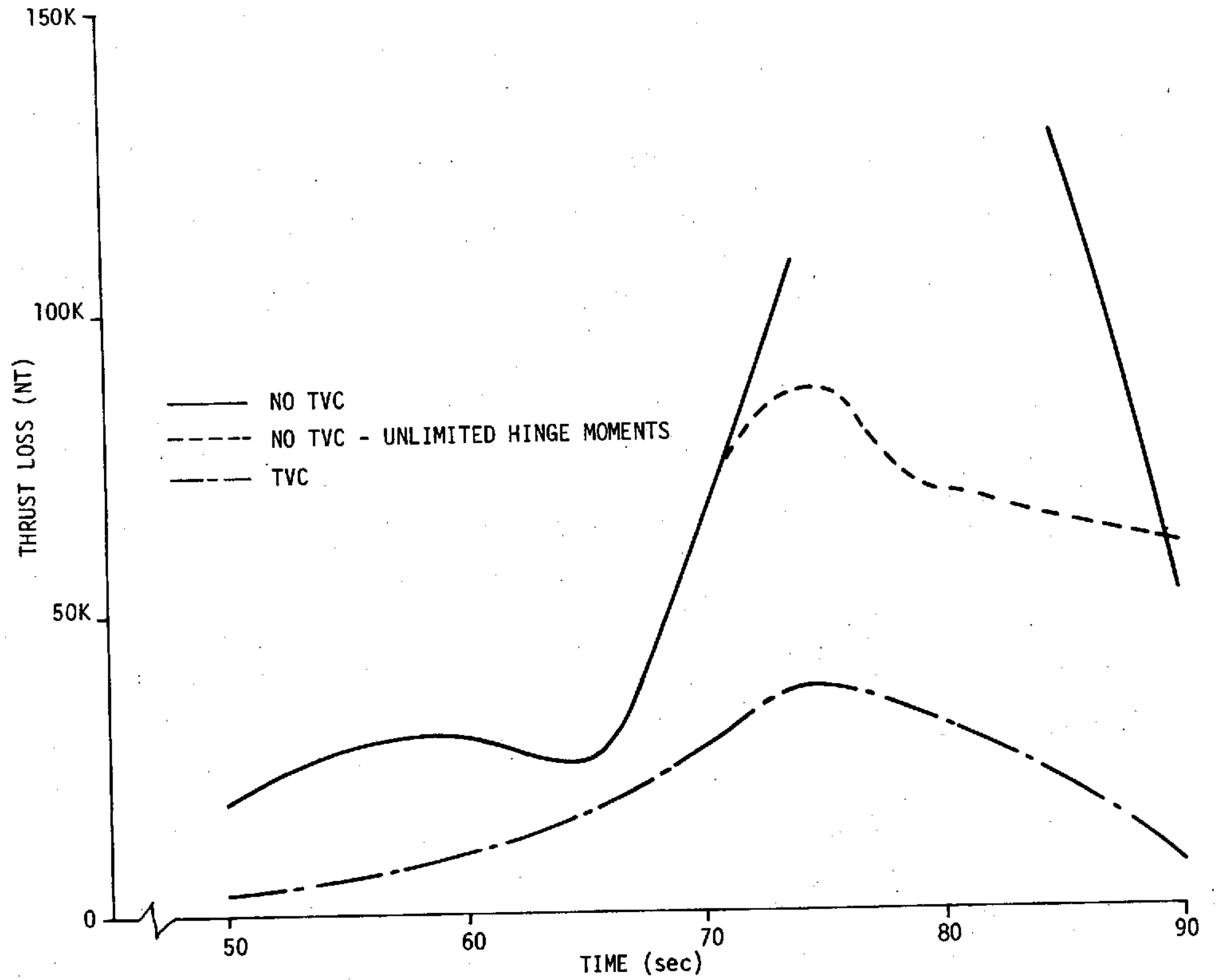


Figure 2-10. THRUST-DRAG LOSSES VERSUS TIME

aileron was forced into the trim solution to counteract roll torques and the aero surface drag loss was also increased.

2.3.6 Use of Control Devices to Eliminate Side Force

It was further desired to assess the ability of the control devices to function as force canceling devices. The usual philosophy behind control selection and placement is moment cancellation. Almost all of the report up to this point has dealt with nulling torque disturbances while minimizing a cost function (usually thrust/drag loss). This section imposes the requirement of zeroing side force as well as torques. Adding this requirement will theoretically cause the vehicle to fly the exact preprogrammed trajectory regardless of the disturbance.

It was stated in subsection 2.2 that zeroing side force was probably impractical. However, it is impossible to tell the exact control limitations using Lagrangian multiplier techniques. Since the Simplex algorithm will give a solution if one is feasible, this problem was solved again using the new technique.

Figure 2-11 shows the results of this study, using the mean crosswind for the disturbance. As can be seen, the controls met the trim requirements until around 65-70 seconds; at that time the control devices exceeded their limits. The limiting devices are primarily the rudder and the solid engines. When δ_{24R} and δ_{e5L} approached their 10-degree maximums they caused δ_{e4L} and δ_{e5L} to rapidly increase also. At some time between 65 and 70 seconds the full control trim capacity was reached. At about that time the rudder also reached its hinge moment limit. The limitations on one of these variables cause the trim capacity to be exceeded. The aileron and the orbiter engines do not limit the solution to the extent the rudder and solid engines do. The orbiter engines appear to be well within their hard limits up to the time the rudder and solid engines reach their boundaries. The aileron is increasing quickly but it appears to be well within its hinge moment limit at 66-68 seconds.

It is interesting to note that the rudder deflection is negative for this case. The rudder deflection was positive when only roll and yaw torques were

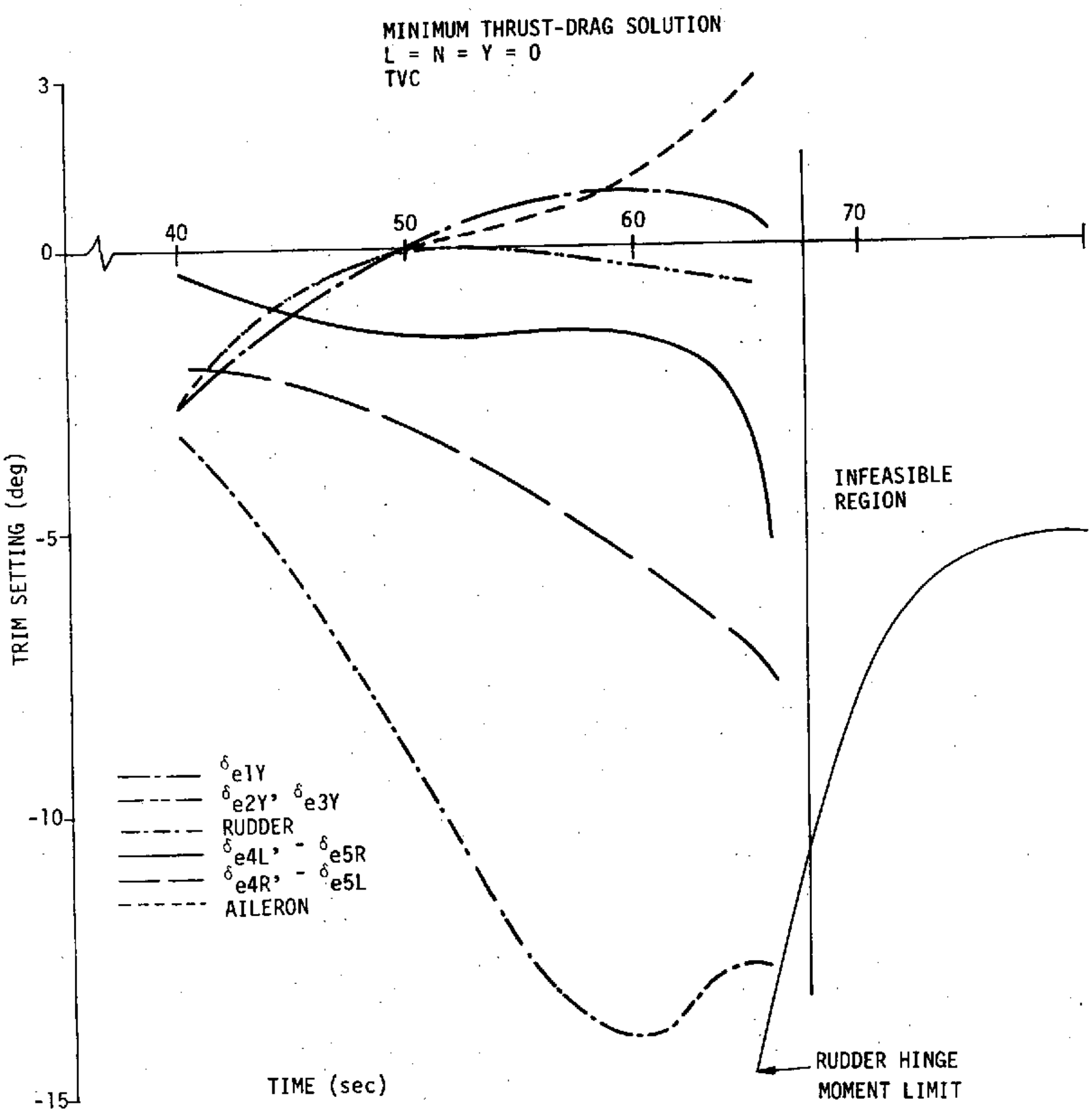


Figure 2-11. MINIMUM THRUST-DRAG SOLUTION L = N = Y = 0 TVC

being cancelled. This happens because the side force requirement dominates the solution. The biggest need is for a negative side force, caused by the wind, and the rudder furnishes part of that. On the other hand, the aileron is positive instead of negative which it was in the previous cases. Apparently the aileron, which has no effect on side force, is being used to cancel some of the torques created by the deflections of the other devices and by the wind torques.

Judging from the control trim limitations alone, it is evident that using the control power to cancel side force is impractical. Other information obtained from the study shows that this requirement is impractical from the standpoint of thrust and drag losses as well. Figure 2-12 shows the losses incurred by trimming the side force are very large compared to those incurred from trimming the roll/yaw torques only (Figure 2-10). Note that the curve seems to be increasing asymptotically as it nears the infeasible region. This is caused by the lack of surplus control authority which limits the flexibility of the minimization scheme. As it nears the limits of feasibility this curve has the same appearance as the curve on Figure 2-10 which represents no TVC. The two curves behave similarly for the same reason. When the trim requirements are very stringent, the minimization program has no room to adjust the controls for minimum loss.

If the aerodynamic hinge moment were increased it would be possible to trim the vehicle's side force. Although only one time point was examined (75 seconds) it was found that the solution at that point required about 17 degrees of negative rudder and five degrees of SRM displacement. The orbiter engines and the aileron were negligibly displaced compared to the rudder and solids. Experience gained from the study indicates that all other time points will trim if 75 seconds has a significant trim capability margin. Nevertheless, the thrust-drag loss was around 200,000 newtons, which would seem to be unacceptably high.

2.4 DYNAMIC SURVEY

In the study to this point the disturbance has been the mean crosswind. In order to assess the interaction of the dynamics with the control system

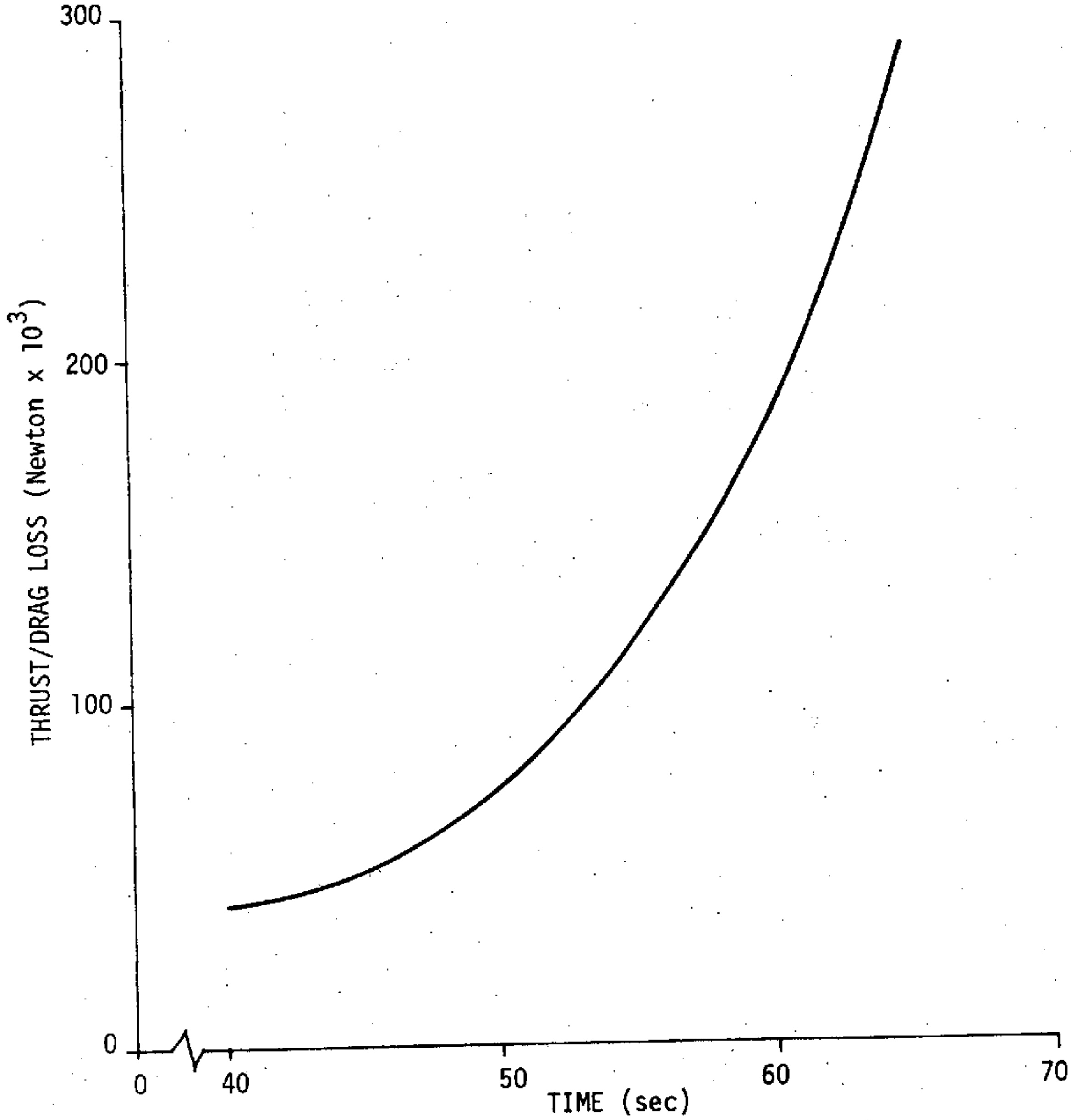


Figure 2-12. MINIMUM THRUST DRAG LOSS VERSUS TIME ZERO SIDE FORCE MEAN CROSSWIND DISTURBANCE

command distribution a quick-look dynamic response was generated for the NAR configuration. This configuration was chosen because the simulation of it was readily available. The data for this configuration are presented in Appendix D. Also in this appendix is a block diagram of the dynamics model and NAR control scheme. The disturbance was a synthetic crosswind with a gust of 7.5 m/sec superimposed on a 75 m/sec wind.

The results of the NAR simulation are shown in Figure 2-13. The commanded control settings are the same as the actual control settings if actuator dynamics are neglected. As can be seen from Appendix D, the rudder, δ_{e1} and δ_{e23D} are used solely to control roll. Yaw is controlled by δ_{e23L} and the solids (δ_{eL4} and δ_{eL5}) are controlled by feedback from both yaw and roll. The torque outputs from these control settings were split into roll and yaw to generate a time history of roll/yaw torque commands.

These torque commands were input to the simplex program to generate a set of minimized trim solutions. The results of the minimum thrust-drag solution process are shown in Figure 2-14. The rudder is more active than the NAR control scheme allows (no hinge moment limit was known or used). The top orbiter engine (δ_{e1}) receives a larger displacement and is yawed completely to its limit by five seconds. The solid engines and δ_{e23D} are slightly less active than the NAR control scheme calls for. There appears to be no significant differences in the deflection of δ_{e23L}.

The most striking result of this graph is the similarity in results between the NAR control system and the Simplex method of torque command distribution. It is probable that NAR performed some similar sort of optimization to determine their signal-splitting system.

Figure 2-15 shows the thrust-drag losses for the two control schemes. The simplex scheme yields about 10 percent less thrust-drag loss. However, this is offset by the fact that the simplex scheme places the top orbiter engine on its limit at the time of peak disturbance.

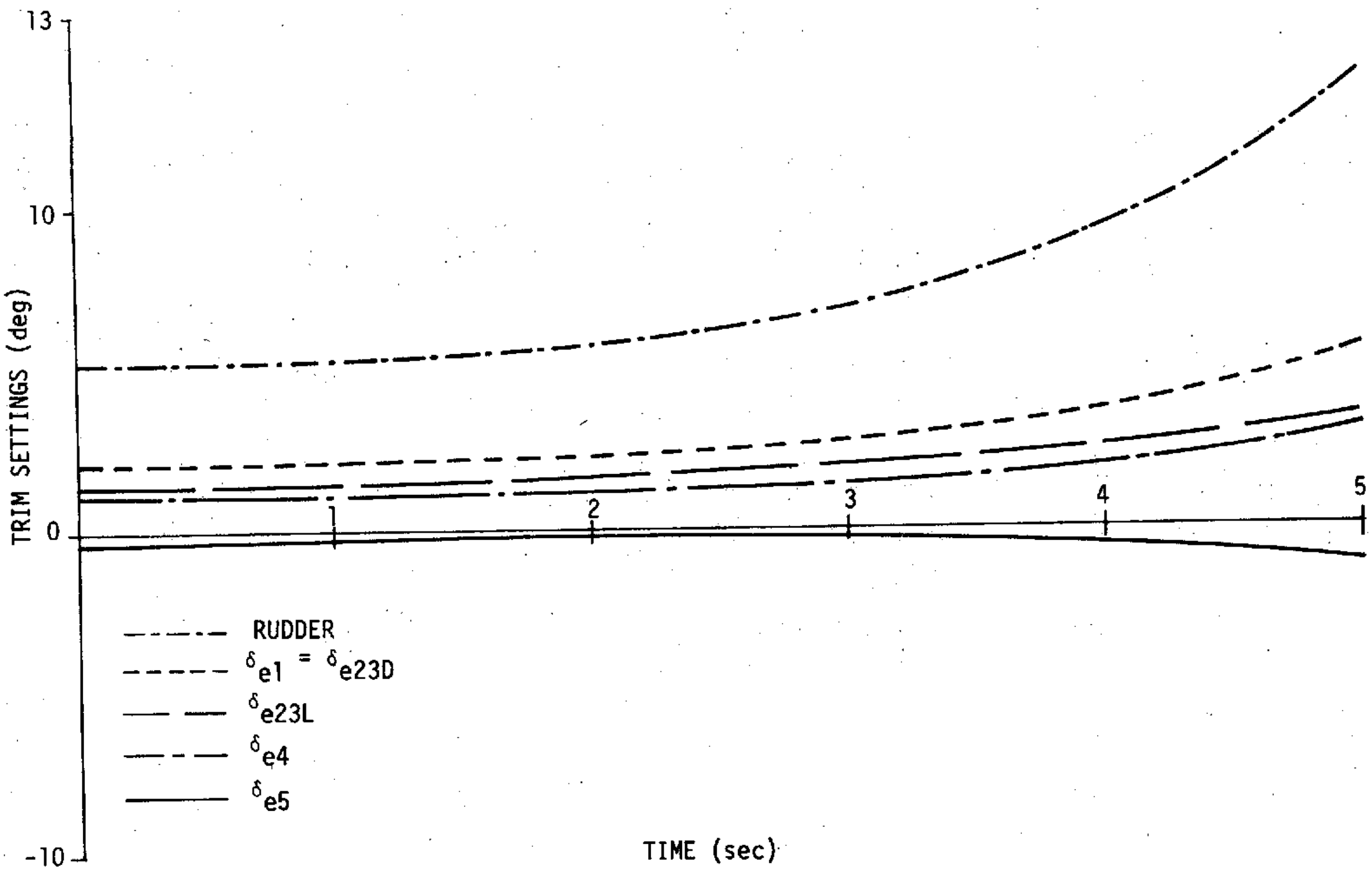


Figure 2-13. SYNTHETIC WIND DISTURBANCE DYNAMIC RESPONSE NAR CONTROL SCHEME

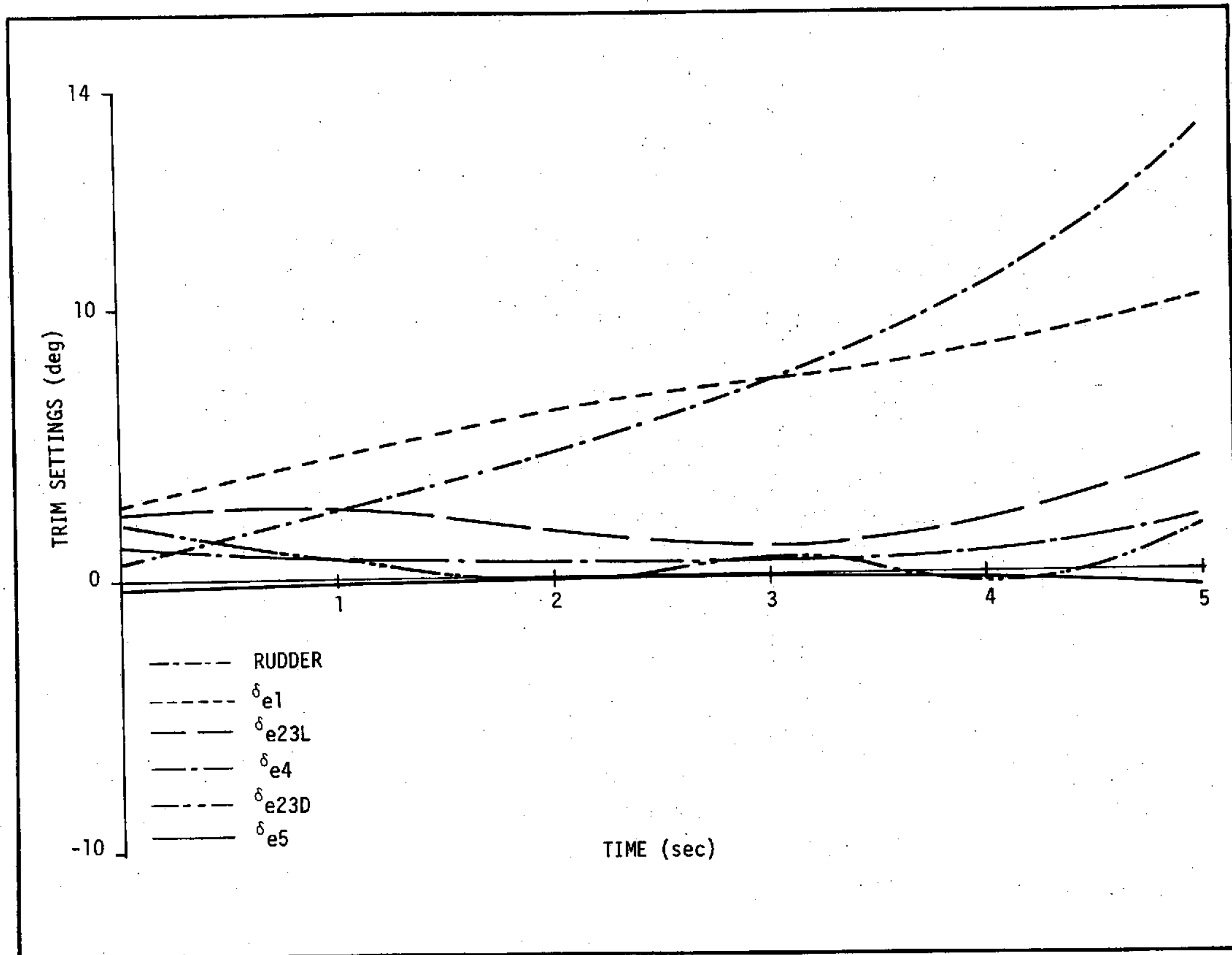


Figure 2-14. SYNTHETIC WIND DISTURBANCE DYNAMIC RESPONSE SIMPLEX COMMAND DISTRIBUTION

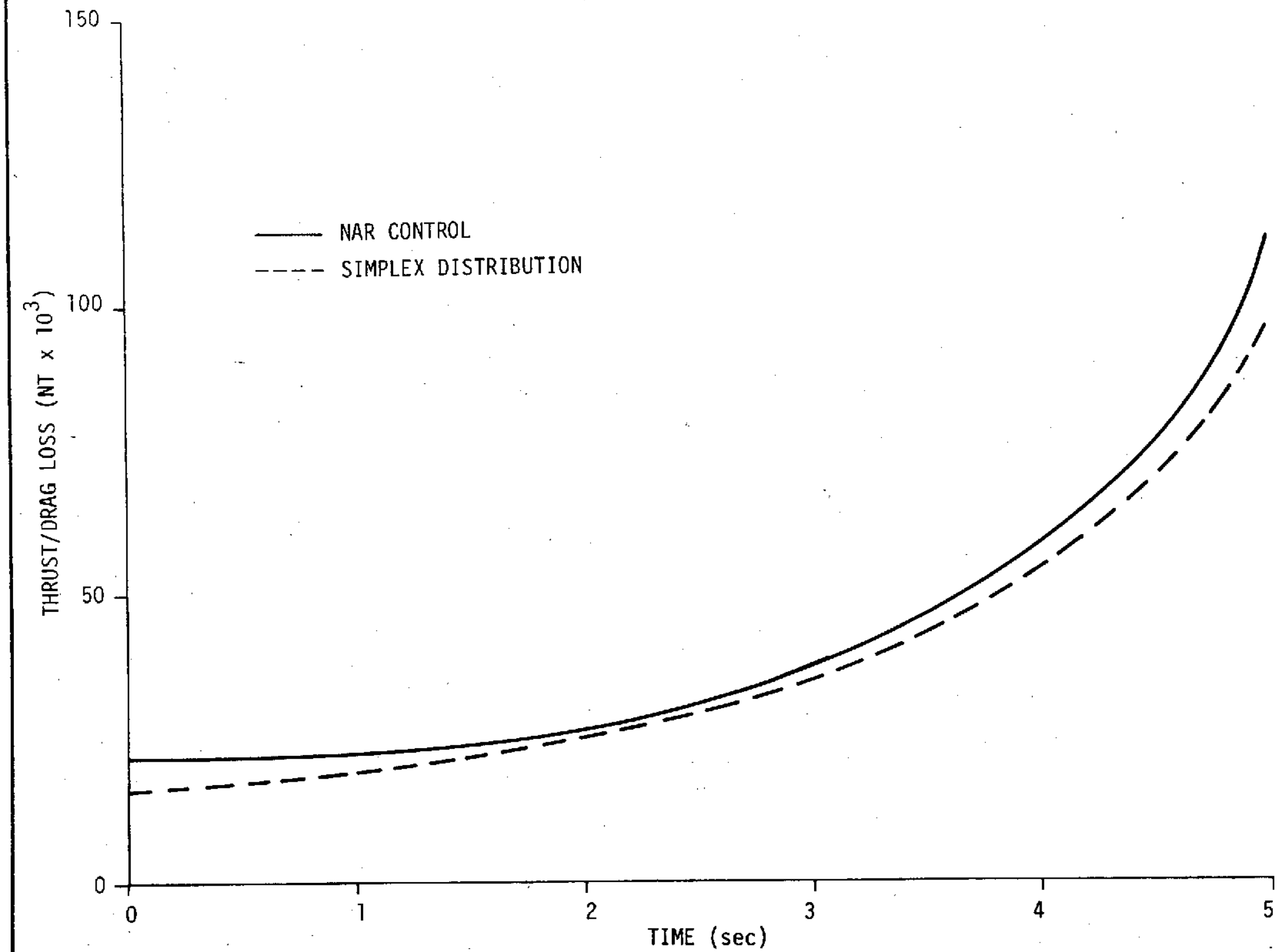


Figure 2-15. SYNTHETIC WIND DISTURBANCE THRUST/DRAG LOSSES

The penalty coefficient on orbiter engine one could be adjusted to move it off its limit. Alternatively the cost functional could be modified to penalize proximity to the control limits. This type of penalty function was discussed earlier in the report.

2.5 FURTHER APPLICATIONS

There are several possible additional uses for this algorithm. On the current NAR configuration there are still payload and load problems during ascent. These problems may require changing the control system to resolve them. The solution algorithm should give an indication of the way to proceed in changing gains and signal splitting. If soft limits are desired, the algorithm could be used to generate position dependent control gains to achieve the same purpose. Ultimately it may be possible to implement the system on-board. The problem that must be solved here is computer time. It requires three to nine matrix row or column operations to get from one flight state to the next if the time change is small enough. This system would automatically solve an engine malfunction problem, minimizing the deterioration in mission capability.

The algorithm has several uses applicable to new configurations. If a configuration does not have the control system defined, a typical torque command profile, such as the mean wind, may be used with the algorithm to compute the settings of the control devices. This will give a preliminary control assessment to determine if controllability problems exist and how to proceed with signal splitting. If they are not already set, the algorithm can be used to set engine cant angles at their most efficient values. Use of the algorithm on problems such as these will add to the background and feel of the designer.

Section III

CONCLUSIONS

- Lagrangian multiplier techniques may be used to solve quadratic cost functions only. They also provide a method for blending the commands to the control devices. However, the trim problems must be unbounded for a solution to always be available.
- The Simplex based algorithm works for general problems. The cost function may be of any form as long as it is piecewise differentiable. The algorithm considers the variables' bounds explicitly in the solution.
- The algorithm is most effective where there is adequate surplus trim power to give flexibility to the solution.
- The algorithm can be used to solve problems such as "optimally" trimming the shuttle ascent vehicle or the orbiter in reentry. ("Optimal" trim means approximately the optimal trim solution with respect to a particular cost function.) It is also useful in obtaining a feel for the trim range and an insight into the basis of signal splitting schemes on the conventional Shuttle ascent control systems. The algorithm may be used in flight if sufficient digital computer speed is available. On-board it would solve the engine malfunction problem automatically.
- The 049 ascent vehicle has controllability problems if the solid engines are not vectorable. If the solid engines are used the controllability problem vanishes. Without SRM TVC simplex shows that there is no feasible roll, yaw trim solution between 74 and 85 seconds of flight. With beefed up rudder hydraulics there is a solution.
- The dynamic survey performed on the NAR configuration showed the simplex control scheme to be close to the results obtained from the NAR control scheme. The simplex scheme saved about 10 percent of the thrust-drag losses compared with the NAR control scheme.

Section IV

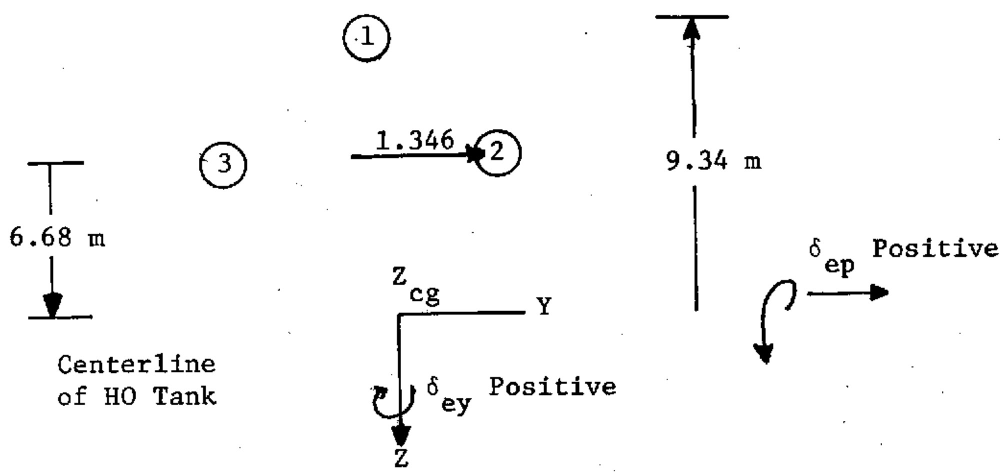
REFERENCES

1. Ackoff, R. L. and Sasieni, M. W., Fundamentals of Operations Research John Wiley and Sons, Inc., 1968.

2. W. Orchard-Hays, Matrices, Elimination and the Simplex Method, 1961.

Appendix A
SHUTTLE 049 DATA

ENGINE FORCES AND MOMENTS



Engine 1

$F \cos 18° \; \delta_{ey_1}$ force in Y

$(9.34 + Z_{cg}) F \cos 18° \; \delta_{ey_1}$ roll

$-X_{cg} \; F \cos 18° \; \delta_{ey_1}$ yaw

where X_{cg}, Z_{cg} > lengths

Engine 2.

$F\{\cos 12° \cos 3.5° \; \delta_{ey_2} + \sin 12° \sin 3.5° \; \delta_{ep_2} + \cos 12° \sin 3.5°\}$

$(6.68 + Z_{cg})\{Y \text{ force}\} - 1.345F\{\sin 12° - \cos 12° \; \delta_{ep_2}\}$

$- X_{cg} \; F \; \{\sin 12° \sin 3.5° \; \delta_{ep_2} + \cos 12° \cos 3.5° \; \delta_{2ey}\}\{\text{yaw}\}$

Engine 3.

$$F\{\cos 12° \cos 3.5° \delta_{ey3} - \sin 12° \sin 3.5° \delta_{ep3} - \cos 12° \sin 3.5°\} \text{ (Y force)}$$

$$(6.68 + Z_{cg}) \text{ (Y Force)} + 1.346F\{\sin 12° - \cos 12° \delta_{ep3}\} \quad \text{roll}$$

$$- X_{cg} F \{-\sin 12° \sin 3.5° \delta_{ep3} + \cos 12° \cos 3.5° \delta_{ey3}\} \quad \text{yaw}$$

A ventral fin was added to the configuration to improve the aerodynamic properties in the region of max q. It significantly reduces the yaw and roll moments due to side slip. This effect is accounted for by direct addition of ΔC_{N_ρ}, ΔC_{y_β} and ΔC_{L_β} to their respective terms in the equations.

DATA

FLIGHT TIME SEC	NEWT THRUST/ ENGINE	ZCG m	XCG m	NEWT/m^2 DYNAMIC PRESSURE	Aref m^2	bref m	NEWT THRUST/SOL.
25	1.65×10^6	-1.58	23.345	$.482 \times 10^4$	3.177×10^2	2.832×10^2	1.28×10^7
40	1.76×10^6	-1.5847	23.42	$.987 \times 10^4$			1.12×10^7
50	1.825×10^6	-1.5914	23.47	$.134 \times 10^5$			1.137
60	1.885×10^6	-1.5953	23.52	$.174 \times 10^5$			1.157
65	1.92×10^6	-1.5979	23.545	$.194 \times 10^5$			1.166
70	1.94×10^6	-1.60	23.57	$.212 \times 10^5$			1.175
75	1.97×10^6	-1.4626	24.13	$.226 \times 10^5$			1.183
80	1.98×10^6	-1.455	24.18	$.233 \times 10^5$			1.192
90	2.025×10^6	-1.440	24.33	$.217 \times 10^5$			1.205
100	2.04×10^6	-1.4327	24.535	$.165 \times 10^5$			1.213
110	2.06×10^6	-1.4255	24.74	$.117 \times 10^5$			1.216
145	2.07×10^6	-1.400	25.62	$.231 \times 10^4$			1.0

DATA

400 FT^2 - FIN		DATA IS PER/DEG		
FLIGHT TIME	AFT $\Delta Cm\beta$	$\Delta C_{1\beta}$	FORWARD $\Delta Cm\beta$	$\Delta Cy\beta$
25	.0064	.0031	-.004	-.011
40	.0067	.0032	-.0044	-.012
50	.0074	.0033	-.0048	-.013
60	.0085	.0036	-.0056	-.015
65	.0094	.0038	-.006	-.016
70	.014	.0042	-.006	-.017
75	.01	.0042	-.0058	-.0165
80	.0088	.0035	-.005	-.014
90	.0075	.0027	-.0044	-.0105
100	.005	.0017	-.0028	-.008
110	.004	.0014	-.0022	-.006
145	.0028	.001	-.0015	-.004

DATA

β_W	FLIGHT TIME	TOTAL VELOCITY m/sec VR	Uo m/sec	Wo m/sec	Cos θ_o	Sin θ_o	GRAVITY g m/sec^2	MACH #	WIND VELOCITY m/sec	(DEG) C_{D8}
1.2°	25	$.954 \times 10^2$	$.954 \times 10^2$	.279	-.012	1.0	9.8	.28	2.0	.0007
3.4°	40	$.150 \times 10^3$	$.149 \times 10^3$	$.165 \times 10^2$	.050	.999	9.8	.44	9.0	
4.5 (4.6)	50	$.190 \times 10^3$	$.186 \times 10^3$	$.340 \times 10^2$	.125	.992	9.79	.56	15.0	
5.7 (5.9)	60	$.241 \times 10^3$	$.232 \times 10^3$	$.654 \times 10^2$	.222	.975	9.78	.707	24.0	
6.1 (6.5)	65	$.272 \times 10^3$	$.257 \times 10^3$	$.875 \times 10^2$	.274	.962		.8	29.0	
6.4 (6.9)	70	$.305 \times 10^3$	$.283 \times 10^3$	$.115 \times 10^3$	.329	.944		.9	34.0	
6.7	75	$.343 \times 10^3$	$.310 \times 10^3$	$.148 \times 10^3$	.384	.923	9.77	1.01	40.0	
6.5	80	$.385 \times 10^3$	$.337 \times 10^3$	$.187 \times 10^3$	.449	.893		1.13	44.0	
3.53	90	$.486 \times 10^3$	$.392 \times 10^3$	$.288 \times 10^3$	.566	.824	9.76	1.43	30.0	.007
0	100	$.612 \times 10^3$	$.445 \times 10^3$	$.420 \times 10^3$	.664	.748	9.74	1.8	0	
	110	$.768 \times 10^3$	$.498 \times 10^3$	$.584 \times 10^3$	.681	.732	9.73			
	140	$.152 \times 10^4$	$.673 \times 10^3$	$.126 \times 10^4$	.874	.486	9.68			

DATA

FLIGHT TIME	$C_{\ell_{Sr}}$	$C_{y_{Sr}}$	$C_{n_{Sr}}$	$C_{1\beta}$	$C_{m\beta}$	$C_{y\beta}$	$q_o\theta_o$
25	.273	.504	-.510	-.283	.302	-1.66	$-.203 \times 10^{1}$
40	.265	.408	-.489	-.285	.315	-1.68	$-.512 \times 10^{-2}$
50	.259	.462	-.473	-.286	.325	-1.70	$.450 \times 10^{-2}$
60	.215	.394	-.388	-.291	.404	-1.83	$.112 \times 10^{-2}$
65	.181	.319	-.310	-.298	.468	-1.99	$-.158 \times 10^{-1}$
70	.173	.300	-.345	-.326	.460	-2.05	$.687 \times 10^{-1}$
75	.206	.292	-.340	-.384	.344	-1.97	-.103
80	.186	.217	-.254	-.356	.266	-1.92	$.412 \times 10^{-1}$
90	.105	.132	-.137	-.299	.238	-1.93	$-.277 \times 10^{-2}$
100	.055	$.961 \times 10^{-1}$	-.105	-.246	.269	-2.03	$.301 \times 10^{-3}$
110	.0406	$.749 \times 10^{-1}$	-.077	-.196	.207	-1.98	$-.851 \times 10^{-2}$
145	.0286	$.573 \times 10^{-1}$	-.061	-.122	-.0284	-1.60	$-.100 \times 10^{-2}$

DATA

FLIGHT TIME	$C_{m_{\delta a}}$	$C_{l_{\delta a}}$	MASS Kg	Ix $Kg - m^2$	Iz $Kg - m^2$
25	.0458	-.0430	$.218 \times 10^7$	$.953 \times 10^8$	$.591 \times 10^9$
40	.0444	-.0458	$.201 \times 10^7$	$.856 \times 10^8$	$.547 \times 10^9$
50	.043	-.0487	$.190 \times 10^7$	$.794 \times 10^8$	$.519 \times 10^9$
60	.0358	-.0544	$.179 \times 10^7$	$.733 \times 10^8$	$.491 \times 10^9$
65	.0344	-.0630	$.174 \times 10^7$	$.702 \times 10^8$	$.478 \times 10^9$
70	.0301	-.0630	$.169 \times 10^7$	$.671 \times 10^8$	$.464 \times 10^9$
75	.0258	-.0544	$.160 \times 10^7$	$.629 \times 10^8$	$.420 \times 10^9$
80	.0244	-.0458	$.154 \times 10^7$	$.606 \times 10^8$	$.405 \times 10^9$
90	.0172	-.0286	$.144 \times 10^7$	$.559 \times 10^8$	$.375 \times 10^9$
100	.00286	-.0215	$.133 \times 10^7$	$.512 \times 10^8$	$.346 \times 10^9$
110	-.00286	-.0158	$.122 \times 10^7$	$.466 \times 10^8$	$.317 \times 10^9$
145	-.0114	-.00859	$.914 \times 10^6$	$.329 \times 10^8$	$.228 \times 10^9$

DATA

FLIGHT TIME	RUDDER HINGE MOMENT LIMIT DEGREES	δa (INCLUDED ANGLE) AILERON HINGE MOMENT LIMIT DEGREES
25	No Hinge Moment Limit (30)	No Aileron Limit
40	42 (30)	71.8
50	30.8 (30)	52.6
60	23.5	40.0
65	14.7	25.1
70	8.19	14.1
75	5.54	9.47
80	5.23	8.91
90	6.27	10.69
100	10.23	17.5
110	19.67	33.64
140	No Hinge Limit	No Hinge Limit
	Hard Limits	Hard Limits
	(±30°)	(40° up - 15° down)

The hinge moment limits are a function of dynamic pressure and as such would change with trajectory variation. However, this data assumes a nominal ascent. In addition, I have included the hard deflection limits which govern when the hinge moments do not. I do not have any data for it yet, but there should be a yaw introduced by unequal aileron deflection.

Appendix B

SIMPLEX ALGORITHMS

The simplex algorithm is a well known algorithm for solving linear, bounded problems with linear cost functionals. It is uniquely suited to the trim optimization problem since simplex solves the problem of more variables than constraints and also handles explicit inequality constraints. The original simplex algorithm is as follows:

1. Examine the a_j^o, $j = 1, \ldots, n-m$.

   a. If all $a_j^o > 0$, the solution is optimal.

   b. If some $a_j^o < 0$, choose $j = s$ by $a_s^o = \min a_j^o < 0$.

2. Examine the a_s^i, $i = 1, \ldots, m$.

   a. If all $a_s^i \leq 0$, the value of Z is unbounded in the class of solutions given by (7).

   b. If some $a_s^i > 0$, choose $i = r$ by

   $$\min_{a_s^i > 0} \frac{b_i}{a_s} = \frac{b^r}{a_s^r} > 0 \quad (b^r > 0)$$

3. Using a_s^r for a pivot, perform one elimination to obtain the new canonical form. The elimination is applied to the right hand side to give the new values of the new basic variables and an extra E.R.T. applied to row zero to give the new relative cost factors. The new value of Z (with sign changed) appears in row zero of the right side.

4. Return to 1.

It can be shown (ref. 2) that if a feasible solution exists, we can obtain a basic feasible solution. Applying the simplex algorithm to this solution, assuming Z is bounded below, we either prove the solution optimal or obtain an improved basic feasible solution, provided all $b^i > 0$ for $a_s^i > 0$. Thus we can never return to a previous solution since each one has a lower value of Z than the preceding one. But there are only a finite number of basic solutions (not exceeding the number of combinations on n things taken m at a time) so that we must eventually arrive at some best basic feasible solution. But this must be an optimal solution or we could repeat the algorithm. Hence there exists a basic feasible solution which is optimal. This proves that if any feasible solution exists and Z has a finite minimum then there exists a basic optimal solution. Furthermore it shows that the algorithm is a finite iterative procedure, i.e., we arrive at a solution in a finite number of steps. This is also clearly true if Z is unbounded below.

Finding A Basic Feasible Solution - The number of linearly independent rows of a restraint matrix is called the rank of the system. If the rank p is less than m, then either there are m-p redundant equations or else there is no solution at all, feasible or otherwise. If the rank is m, then there is some basic solution but there may be no feasible solution. Hence, given a problem there is, in general, no way of knowing whether any solution exists.

To overcome these difficulties, a very simple device exists which either finds a feasible solution or determines that none exists. It also induces rank m in case $p < m$. The idea is to embed the given problem in a larger system and solve a preliminary LP problem before attempting to minimize Z. This process is called Phase I of the simplex method, the optimization of Z being referred to as Phase II.

We suppose that all $b^i \geq 0$. (If not, the entire equation for $b^t < 0$ can be multiplied by -1.)

We now define m artificial variables

$$x^{n+1} \geq 0,\ x^{n+2} \geq 0,\ \ldots,\ x^{n+m} \geq 0$$

and x^{n+m+1} which will eventually be restricted to be non-negative. The coefficients of these variables will be new vectors arbitrarily adjoined to the restraint matrix. The coefficient of x^{n+i} will be unity in row i and in an added row w, zero elsewhere. Letting $m + 1 = w$ the variable x^{n+w} will have a coefficient only in row w. The system of restraints is now

$$\sum_{i=1}^{w} x^{n+i} = 0 \tag{B-1}$$

$$\sum_{j=1}^{n} a_j^i \, x^j + x^{n+1} = b^i \quad (i = 1, \ldots, m) \tag{B-2}$$

Since the x^{n+1} are artificial, they must not appear in the final basic solution, except perhaps at zero level. If $x^{n+w} \geq 0$, then (B-1) guarantees that they are all zero since a sum of non-negative numbers equal to zero implies each is zero.

If the $x^{n+i} = 0$, then clearly (B-2) is the same as the original restraints. Note that the rank of the system (B-1), (B-2) is m+1. This is clearly seen by displaying the entire expanded tableau:

$$\begin{array}{c}
\begin{array}{ccccccccc}
x^1 & x^2 & \cdots & x^n & x^{n+1} & x^{n+2} & \cdots & x^{n+m} & x^{n+w}
\end{array} \\
\left[\begin{array}{ccccccccc}
a_1^o & a_2^o & \cdots & a_n^o & 0 & 0 & \cdots & 0 & \\
a_1^1 & a_2^1 & \cdots & a_n^1 & 1 & & \cdots & 0 & \\
\vdots & \vdots & & \vdots & \vdots & \ddots & & & \\
a_1^m & a_2^m & \cdots & a_n^m & 0 & 0 & \cdots & 1 & \\
0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 1
\end{array}\right]
\end{array}
= \left[\begin{array}{c} Z \\ b^1 \\ \vdots \\ b^m \\ 0 \end{array}\right] \tag{B-3}$$

Although (B-3) is not quite in canonical form, there is an obvious starting solution if we allow x^{n+w} to be negative, namely:

$$x^j = 0, \; j = 1, \ldots, n$$

$$x^{n+w} = -\sum_{i=1}^{m} b^i$$

$$x^{n+i} = b^i, \; i = 1, \ldots, m. \tag{B-4}$$

The solution (albeit infeasible) will remain valid of we apply the following E.R.T:

$$a_j^w = a_j^w - a_j^1 \quad (j = 1, \ldots, n+m), \; b^w = b^w - b^1$$

i.e., if we subtract Equation 1 from Equation w. We can then subtract equations 2, 3, ..., m in turn from Equations w. If we redefine a_j^w and b^w as

$$a_j^w = \sum_{i=1}^{m} a_j^i \; (j = 1, \ldots, n+m)$$

$$b^w = -\sum_{i=1}^{m} b^i$$

the result is the following tableau:

$$\begin{array}{cccccccccc}
x^1 & x^2 & \cdots & x^n & x^{n+1} & x^{n+2} & \cdots & x^{n+w}
\end{array}$$

$$\begin{bmatrix}
a_1^o & a_2^o & \cdots & a_n^o & 0 & 0 & \cdots & 0 \\
a_1^1 & a_2^1 & \cdots & a_n^1 & 1 & 0 & \cdots & 0 \\
a_1^2 & a_2^2 & \cdots & a_n^2 & 0 & 1 & \cdots & 0 \\
\cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
a_1^w & a_2^w & \cdots & a_n^w & 0 & 0 & \cdots & 1
\end{bmatrix} = \begin{bmatrix} Z \\ b^1 \\ b^2 \\ \cdot \\ \cdot \\ \cdot \\ b^w \end{bmatrix} \tag{B-5}$$

which is in canonical form except that $b^w < 0$, and hence $x^{n+w} < 0$. The first objective will therefore be to maximize x^{n+w}. Zero is clearly an upper bound for x^{n+w} by (13). On the other hand, any feasible solution to the original problem would leave all $x^{n+i} = 0$. Therefore if max $x^{n+w} < 0$, there is no solution to the given problem.

From Equation w in (B-5), it is clear that maximizing x^{n+w} is the same as minimizing $F = \sum_{j=1}^{n} a_j^w x^j$ since their sum is a constant,

$$x^{n+w} + F = b^w.$$

Hence we simply use the a_j^w for relative cost factors in Phase I, carrying along a_j^o to be ready for Phase II if and when $x^{n+w} = 0$.

During Phase I, maximizing x^{n+w}, its column of coefficients must never be a candidate for elimination.

It is not possible to eliminate all the artificial variables from any basic solution even in Phase II. For even if the m original restraint rows a_j^i ($i = 1, \ldots, m$; $j = 1, \ldots, m$) were linearly independent, row w is a linear combination of them and hence the $m + 1$ rows for $i = 1, \ldots, w$ and $j = 1, \ldots, n$ are linearly dependent. Hence at least one artificial column must always remain and we have insisted that column $n + w$ stay in during Phase I. If, however, the original m rows had rank $p < m$, then $m - p + 1$ artificial columns will remain at the end of Phase I. If $x^{n+w} = 0$, the solution is a valid one; if $x^{n+w} < 0$, there is no solution.

Degeneracy may also cause x^{n+w} to reach zero before the other artificial columns are removed from the basis. In this case, some of the a_j^w may still be negative. Thus Phase I should be terminated when either $x^{n+w} = 0$ or when all $a_j^w \leq 0$, whichever happens first.

It is possible that x^{n+w} will be eliminated in Phase II if some other artificial column is still in the basis. Whenever any artificial column is eliminated from the basis, it is simply dropped from further consideration.

If the original restraints contained some legitimate unit vectors, then the corresponding artificial columns need not be added and the corresponding rows are not subtracted from row w.

Appendix C
PROGRAM SIMPLEX

This appendix is a listing and description of the computer program developed to solve the problems in this study. Figure C-1 is a flowchart of the calling sequence for program SMPLX.

DATRD is the program that reads the data and calculates the control devices' stability derivatives. This program also reads in such things as penalty coefficients and initial solution and starting range. The program can be started with a zero vector for the starting solution and 100 percent range to insure a feasible solution if one is available.

Then the program enters SIMTM which calls all the subroutines that calculate the solution. For the first pass the partial costs are set (subroutine "cost" calculates $\frac{\partial Z}{\partial \delta}$) at the initial solution, the temporary bounds are set up about that solution and the simplex tableau is set up and solved. This gives the minimum cost sum (subroutine SCOST) on the temporary range if the costs are linear enough on that range.

On subsequent passes CVARB calculates the actual variables from the simplex transformed variables, TORQ calculates the sum of the variables times their coefficients and the disturbance, and SCOST calculates the actual total cost so that actual improvement can be discerned. Then the new cost is compared to the old cost and if there is no improvement the bounds are squeezed by half (SQZ). If there is improvement the new partial costs (SCOST) are calculated at the new solution. Then the new temporary bounds are set. Subroutine SETUP puts the problem in simplex form and SOLTN solves it. The improvement criteria is then checked and if it is fulfilled, the program loops again.

The only things that require change to solve a different problem are COST and SCOST, and then only if the cost function changes form from Equation (16). SCOST represents the total cost and if the form is changed, then the cards from "TCOST = 0.0" to "RETURN" must be altered according to the actual cost equations. Then in subroutine COST the partial derivatives of the total cost equation, evaluated at the current solution, are calculated. To change COST, simply remove the cards from "18 CONTINUE" to "DO 22 I = 1, NVAR".

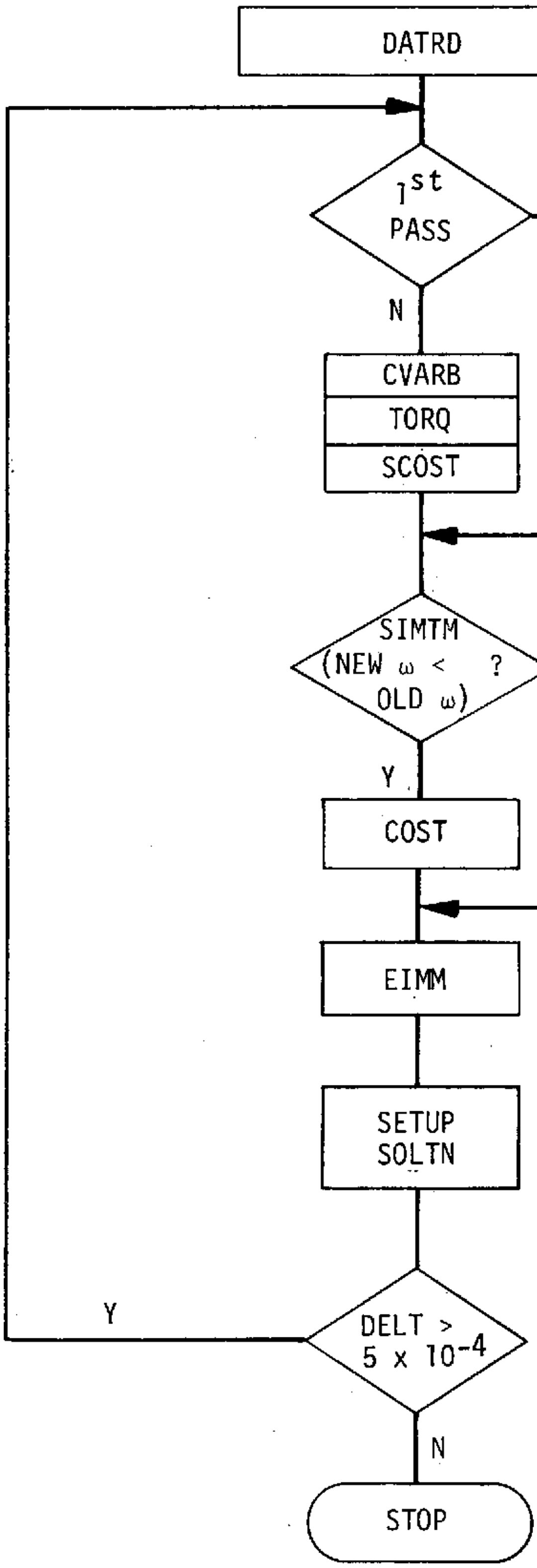


Figure C-1. CALLING SEQUENCE FLOWCHART OF PROGRAM SIMPLEX

The number of variables in the problem is represented by Fortran mnemonic "NVAR". The number of equality constraints is called "NEQ". These must be changed if the problem size changes. The program is dimensioned for a maximum of three equations and 12 variables.

An input description is listed below:

CARD				
	FORMAT(5E13.4)			
1	YE1	ZE1	PAE1	YAE1
2	YE2	ZE2	PAE2	YAE2
3	YE3	ZE3	PAE3	YAE3
4	YE4	ZE4	PAE4	YAE4
5	YE5	ZE5	PAE5	YAE5

(YE1 is y position relative to cg of engine 1, ZE2 is z position engine 2, PAE3 is pitch angle engine 3, YAE4 is yaw angle engine 4)

FORMAT I3

6 TIME (time of flight)

FORMAT (5E13.0)

7 THR, ZCG, XCG, AREF, BREF

(orbiter thrust per engine (nt), ZCG 9m), XCG (m), reference area (m^2), reference length (m))

8 QBAR, 6, MASS, IX, IZ

(dynamic pressure (nt/m^2), gravity (m/sec^2) mass (kg), x axis inertia (kg m^2), z axis inertia (kg m^2)).

9 THRS, DELT

(solid engine thrust per engine (nt), initial solution range, rates of full range 0 - 1.0)

10 (D1MAX(I), DIMIN(I), I=1,NVAR)

(maxima and minima for each variable one at a time)

11 DCCB, DCNB, DCVB

(Δ_{ℓ_β} ΔC_{n_β}, ΔC_{y_β})

12 CLB, CLDA, CLDR

(C_{ℓ_β}, $C_{\ell_{\delta_a}}$, $C_{\ell_{\delta_R}}$)

CARD

13 CNB, CNDA, CNDR

(C_{n_β}, $C_{n_{\delta_a}}$, $C_{n_{\delta_R}}$)

14 CYB, CYDA, CYDR

(C_{Y_β}, $C_{Y_{\delta_a}}$, $C_{Y_{\delta_R}}$)

15 (TC(I), I=1,NEQ)

(disturbance torques and forces aside from wind)

16 BETA

(wind velocity divided by vehicle total velocity)

17 (PC(I)), I=1,NVAR)

(penalty coefficients)

18 (A(I,KKK), I=1,NVAR)

(initial solution vector (temporary location))

```
// *SHARP X368 9250
// FOR
*ONE WORD INTEGERS
*IOCS(1132PRINTER,CARD)
C      PROGRAM    DATRD(C,3C,NR,NW,NEQ,CNV      ,DELT,NPASS)
       INTEGER TIME
       REAL MASS,IX,IZ
       COMMON A(17,26),W(17),L(17),III,JJ,II,IW
       COMMON    DIMAX(12),DIMIN(12)    ,PC(12) ,DD(12)
       COMMON EIMAX(12),EIMIN(12),C(3,13),TC(3),DELT,TCOST,COST1,COST2
       DATA NR/2/,NW/3/,NEQ/3/,CNV/57.2957795/,NVAR/12/
       COST1=1000000.
       COST2=1000000.
       NVAR1 =NVAR+1
       L(10)=50
       READ (NR,200) YE1,ZE1,PAE1,YAE1
       READ (NR,200) YE2,ZE2,PAE2,YAE2
       READ (NR,200) YE3,ZE3,PAE3,YAE3
       READ (NR,200) YE4,ZE4,PAE4,YAE4
       READ (NR,200) YE5,ZE5,PAE5,YAE5
       READ(NR,100) TIME
       READ (NR,201) THR,ZCG,XCG,AREF,BREF
       READ (NR,201) QBAR,G,MASS,IX,IZ
       READ(NR,201) THRS,DELT
       READ(NR,201)  (DIMAX(I),DIMIN(I) ,I=1,NVAR)
       READ (NR,201) DCLB,DCNB,DCYB
       READ(NR,201) CLB,CLDA,CLDR
       READ(NR,201) CNB,CNDA,CNDR
       READ(NR,201) CYB,CYDA,CYDR
       READ(NR,201) (TC(I),I=1,NEQ)
       READ(NR,201) BETA
       FZCG   = (ZCG+1.47)/BREF
       FXCG   = (XCG-21.6)/BREF
       QS     = QBAR*AREF
       QSB    = QS*BREF
C
C         COMPUTES INDIVIDUAL ENGINE DERIVATIVES
C         ENGINES 1,2 AND 3 ARE LIQUIDS (ORBITER)
C         ENGINES 4 AND 5 ARE SOLIDS (SRM)
C         ENGINE 1 PITCH DOES NOT CONTRIBUTE TO LATERAL FORCES AND MOMENTS
C
       TEMP = PAE1/CNV
       CPE1  = COS(TEMP)
       TEMP = PAE2/CNV
       CPE2  = COS(TEMP)
       SPE2  = SIN(TEMP)
       TEMP = PAE3/CNV
       CPE3  = COS(TEMP)
       SPE3  = SIN(TEMP)
       TEMP = YAE2/CNV
       CYE2  = COS(TEMP)
       SYE2  = SIN(TEMP)
       TEMP = YAE3/CNV
       CYE3  = COS(TEMP)
       SYE3  = SIN(TEMP)
       TEMP = YAE4/CNV
       CYE4 = COS(TEMP)
```

```
      TEMP = YAE5/CNV
      CYE5 = COS(TEMP)
      CME1P = -1.0*XCG*THR*CPE1/QSB
      CLE1Y = (ZE1+ZCG)*THR*CPE1/QSB
      CNE1Y = -1.0*XCG*THR*CPE1/QSB
      CLE2P = -(YE2*CPE2 + (ZCG+ZE2)*SPE2*SYE2)*THR/QSB
      CNE2P = -1.0*XCG*THR*SYE2*SPE2/QSB
      CME2P = -1.0*XCG*THR*CPE2*CYE2/QSB
      CLE2Y = (ZE2+ZCG)*THR*CPE2*CYE2/QSB
      CNE2Y = -1.0*XCG*THR*CPE2*CYE2/QSB
      CME3P = -1.0*XCG*THR*CPE3*CYE3/QSB
      CLE3P = -(YE3*CPE3 + (ZCG+ZE3)*SPE3*SYE3)*THR/QSB
      CNE3P = -1.0*XCG*THR*SYE3*SPE3/QSB
      CLE3Y = (ZE3+ZCG)*THR*CPE3*CYE3/QSB
      CNE3Y = -1.0*XCG*THR*CPE3*CYE3/QSB
      CLE4R = 0.707*THRS*(-YE4-(ZE4 +ZCG)*CYE4)/QSB
      CLE4L = 0.707*THRS*(-YE4+(ZE4 +ZCG)*CYE4)/QSB
      CME4R = -0.707*XCG*THRS*CYE4/QSB
      CLE5R = 0.707*THRS*(-YE5-(ZE5 +ZCG)*CYE5)/QSB
      CLE5L = 0.707*THRS*(-YE5+(ZE5 +ZCG)*CYE5)/QSB
      CME4L = CME4R
      CNE4R = 0.707*XCG*CYE4*THRS/QSB
      CNE4L =-0.707*XCG*CYE4*THRS/QSB
      CME5R = -0.707*XCG*THRS*CYE5/QSB
      CNE5R = 0.707*XCG*CYE5*THRS/QSB
      CNE5L =-0.707*XCG*CYE5*THRS/QSB
      CME5L = CME5R
C
C
C         COMPUTES MOMENT COEFFICIENT RELATIVE TO MOMENT REFERENCE POINT
C
C
C         LOADS ENGINE DERIVATIVES INTO C MATRIX
C
      C(1,1) = CLB + DCLB + CYB*FZCG
      C(1,2) = CLDA
      C(1,3) = CLDR + CYDR*FZCG
      C(1,4) = 0.0
      C(1,5) = CLE1Y
      C(1,6) = CLE2P
      C(1,7) = CLE2Y
      C(1,8) = CLE3P
      C(1,9) = CLE3Y
      C(1,10)= CLE4R
      C(1,11)= CLE4L
      C(1,12)= CLE5R
      C(1,13)= CLE5L
C
      C(2,1) = CNB + DCNB - CYB*FXCG
      C(2,2) = CNDA
      C(2,3) = CNDR - CYDR*FXCG
      C(2,4) = 0.0
      C(2,5) = CNE1Y
      C(2,6) = CNE2P
      C(2,7) = CNE2Y
      C(2,8) = CNE3P
      C(2,9) = CNE3Y
```

```
      C(2,10)=CNE4R
      C(2,11)=CNE4L
      C(2,12)=CNE5R
      C(2,13)=CNE5L
C
      C(3,1) = 0.0
      C(3,2) = 0.0
      C(3,3) = 0.0
      C(3,4) = CME1P
      C(3,5) = 0.0
      C(3,6) = CME2P
      C(3,7) = 0.0
      C(3,8) = CME3P
      C(3,9) = 0.0
      C(3,10)=CME4R
      C(3,11)= CME4L
      C(3,12)=CME5R
      C(3,13)= CME5L
C
C
      DO 4 K = 1,NEQ
      TC(K) = TC(K) -  C(K,1)*BETA
    4 CONTINUE
      READ(NR,201) (PC(I),I=1,NVAR)
      READ(NR,201)(A(I,25),I=1,NVAR)
      WRITE(NW,500) TIME
      WRITE(NW,203) DIMAX
      WRITE(NW,203) DIMIN
      WRITE(NW,99)
      WRITE(NW,501)
      WRITE(NW,203)(C(1,I),I=1,8)
      WRITE (NW,522)
      WRITE (NW,203)(C(1,I),I=9,NVAR1)
      WRITE(NW,502)
      WRITE(NW,203)(C(2,I),I=1,8)
      WRITE (NW,523)
      WRITE (NW,203)(C(2,I),I=9,NVAR1)
      WRITE(NW,503)
      WRITE(NW,203)(C(3,I),I=1,8)
      WRITE (NW,524)
      WRITE (NW,203)(C(3,I),I=9,NVAR1)
      WRITE(NW,202) (PC(I),I=1,NVAR)
      WRITE(NW,200) (TC(I),I=1,NEQ)
      CALL LINK(SIMTM)
   99 FORMAT(2X,//,54H LATERAL NON-DIMENSIONAL STABILITY DERIVATIVES *1/
     1RAD* )
  100 FORMAT(I3)
  200 FORMAT(5E13.4)
  201 FORMAT(5F13.0)
  202 FORMAT(8F13.3)
  203 FORMAT(8F13.3)
  500 FORMAT(1H1,/,10X,8H TIME = ,I3,5H SEC ,///)
  501 FORMAT(// ,5X,5H CLB ,8X,5H CLDA,8X,5H CLDR,3X,
     16H CLE1P,7X,6H CLE1Y,7X,6H CLE2P,7X,6H CLE2Y,7X,6H CLE3P)
  502 FORMAT(// ,5X,5H CNB ,8X,5H CNDA,8X,5H CNDR,3X,
     16H CNE1P,7X,6H CNE1Y,7X,6H CNE2P,7X,6H CNE2Y,7X,6H CNE3P)
  503 FORMAT(// ,5X,5HCMALP,8X,5H CMDA,8X,5H CMDR,3X,
```

```
      16H CME1P,7X,6H CME1Y,7X,6H CME2P,7X,6H CME2Y,7X,6H CME3P)
522   FORMAT(//,5X,5HCLE3Y,8X,5HCLE4R,8X,5HCLE4L,8X,5HCLE5R,8X,5HCLE5L)
523   FORMAT(//,5X,5HCNE3Y,8X,5HCNE4R,8X,5HCNE4L,8X,5HCNE5R,8X,5HCNE5L)
524   FORMAT(//,5X,5HCME3Y,8X,5HCME4R,8X,5HCME4L,8X,5HCME5R,8X,5HCME5L)
      END
```

```
// JOB T                                                       3865

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0002        0002        0000

V2 M10  ACTUAL  8K  CONFIG  8K

// *SHARP X368 9250

// FOR
*IOCS(1132PRINTER,CARD)
*ONE WORD INTEGERS
*ARITHMETIC TRACE
*LIST SOURCE PROGRAM
C     PROGRAM SIMTM
C     LINEAR PROGRAMMING FOR 1130  --DATA LOADER PHASE
C        C     (3,13)     =   COEFFICIENT MATRIX
C        TC    (3)        =   MOMENT COMMANDS
C        PC    (12)       =   PENALTY COEFFICIENTS
      DIMENSION D(12)
      COMMON A(17,26),W(17),L(17),III,JJ,II,IW
      COMMON     DIMAX(12),DIMIN(12)     ,PC(12) ,DD(12)
      COMMON EIMAX(12),EIMIN(12),C(3,13),TC(3),DELT,TCOST,COST1,COST2
      DATA       NW/3/,NVAR/12/
      DATA CNV/57.29578/
      NVAR1=2*NVAR +1
      IW = NW
C
C        COMPUTES VALUES OF CONTROL VARIABLE
      IF(L(10)-50) 70,71,70
   71 CONTINUE
      CALL    COST(  DD,DIMAX,DIMIN,A,PC)
      DO 52 I=1,NVAR
      D(I)=A(I,NVAR1)
      DD(I)=D(I)
   52 CONTINUE
      TCOST = 100000000000000.
      COST1 =10000000000000.
      GO TO 72
   70 CONTINUE
      MN=0
      CALL        CVARB(L,NVAR,A,D,EIMIN,EIMAX,NVAR1)
      CALL        TORQ(NVAR,C,D,TC,ROLLT,YAWT,PITT)
      CALL SCOST(COST2,COST1,TCOST,D,PC,CNV,DIMAX,DIMIN,NVAR)
   72 CONTINUE
C
      IF(COST1-TCOST)85,85,86
   85 CONTINUE
      WRITE(NW,202) TCOST
      MN=1
      TCOST = COST1
      CALL SOZ       (DELT,TCOST,COST1)
      GO TO 88
   86 CONTINUE
      DO 87 I=1,NVAR
      DD(I)=D(I)
```

```
 87   CONTINUE
      CALL     COST(  DD,DIMAX,DIMIN,A,PC)
 88   CONTINUE
C
C     COMPUTES EIMAX,EIMIN  FOR NEXT PASS
      CALL          EIMM(EIMAX,EIMIN,NVAR,DD,DELT,DIMAX,DIMIN)
      IF (MN) 60,60,61
 60   CONTINUE
      WRITE(NW,504)
      WRITE(NW,202) DD
      WRITE(NW,202)ROLLT,YAWT,PITT,TCOST,COST1,COST2,DELT
 61   CONTINUE
      CALL            SETUP(EIMAX,EIMIN,NVAR,C,TC,A,W,L,NVAR1,NW,II,JJ,III)
      CALL SOLTN  (NVAR1,NVAR,D)
      IF (DELT-0.0005) 70,70,73
 73   CALL EXIT
202   FORMAT(8E13.5)
504   FORMAT(1H ,9X,21HCONTROL TRIM SETTINGS/,9X,62HAILERON,RUDDER,ENG 1
     1YAW,ENGS 2/3 PITCH/YAW,ENGS 4/5 RIGHT/LEFT)
      END

FEATURES SUPPORTED
 ARITHMETIC TRACE
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON   1176  VARIABLES     40   PROGRAM    322

END OF COMPILATION

// DUP
```

```
// DUP
*STORECI        WS   UA   SIMTM
*DUMP           UA   CD   SIMTM
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SQZ(DELT,TCOST,TEMP)
      A=ABS(TEMP)
      B=ABS(TCOST)
      A=(A-B)/A
       A=ABS(A)
      IF(A-0.01) 10,10,15
   15 IF(A-DELT) 10,10,20
   10 CONTINUE
      DELT=0.5*DELT
   20 CONTINUE
      RETURN
      END
```

```
ACE 008
// DUP
*STORE      WS  UA  SQZ
// FOR
*ONE WORD INTEGERS
      SUBROUTINE   SETUP(EIMAX,EIMIN,NVAR,C,TC,A,W,L,NVAR1,NW,II,JJ,III)
      DIMENSION EIMAX(12),EIMIN(12),C(3,13),TC(3),A(17,26),W(17),L(17)
      DIMENSION RHS(3)
      RHS(1) =0.0
      RHS(2) =0.0
      RHS(3)=0.0
      II=16
      JJ=25
      DO 62 I=1,NVAR
      J=I+1
      RHS(1) = RHS(1)-C(1,J)*EIMIN(I)
      RHS(3)=RHS(3)-C(3,J)*EIMIN(I)
62    RHS(2) = RHS(2)-C(2,J)*EIMIN(I)
      RHS(1)= TC(1)+RHS(1)
      RHS(2)= TC(2)+RHS(2)
      RHS(3)=TC(3)+RHS(3)
      SIGN1 =     RHS(1)/ABS(RHS(1))
      SIGN2 =     RHS(2)/ABS(RHS(2))
      SIGN3=RHS(3)/ABS(RHS(3))
      A(1,NVAR1) = 0.0
      III=II+1                                                      13LPROG
      DO 1010 I=1,III                                               14LPROG
      W(I)=0.0                                                      15LPROG
      L(I)=0                                                        16LPROG
1010  CONTINUE
      DO 1009 K=2,III
      DO 1009 J=1,JJ
      A(K,J)=0.0
1009  CONTINUE
      DO 47 I=1,NVAR
      K=I+1
      A(2,I)        =  C(1,K   )*SIGN1
      A(3,I)        =  C(2,K   )*SIGN2
      A(4,I)=C(3,K)*SIGN3
47    CONTINUE
      A(2,NVAR1)=    RHS(1)*SIGN1
      A(3,NVAR1)=   RHS(2)*SIGN2
      A(4,NVAR1)=RHS(3)*SIGN3
C
C        RELATION OF TRANSFORMED VARIABLES ,SLACKS
C        AND CONTROL LIMITS
C
      DO 48 K=1,NVAR
      KI=K+4
      KJ= K+NVAR
      L(KI) = KJ
      A(KI,K) =1.0
48    A(KI,KJ)=1.0
      DO 46 I=1,NVAR
      J=I+4
46    A(J,NVAR1) = EIMAX(I)-EIMIN(I)
      WRITE(NW,1004)
1004  FORMAT(12H DATA LOADED)                                       11LPROG
```

```
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   SETUP
// FOR
*ONE WORD INTEGERS
*ARITHMETIC TRACE
      SUBROUTINE    SCOST(COST2,COST1,TCOST,D,PC,CNV,DIMAX,DIMIN,NVAR)
      DIMENSION D(12),PC(12),DIMAX(12),DIMIN(12)
      COST2=COST1
      COST1=TCOST
      TCOST=0.0
      DO 25 I=1,2
   25 TCOST = TCOST+PC(I)*ABS(D(I))
      NVAM=NVAR-1
      DO 26 I=3,NVAM,2
      J=I+1
      S=SQRT(D(I)*D(I)+D(J)*D(J))/CNV
   26 TCOST = TCOST+PC(I)     *(1.0-COS(S))
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  CVARB
// FOR
*ONE WORD INTEGERS
*ARITHMETIC TRACE
      SUBROUTINE COST(D,DIMAX,DIMIN,A,PC)
      DIMENSION D(12),DIMAX(12),DIMIN(12),A(17,26),PC(12)
      DATA NVAR/12/
      DATA CNV/57.29578/
      IR=2*NVAR
      DO 18 I=1,NVAR
      J=I+NVAR
      A(1,J)=0.0
   18 CONTINUE
      DO 19 IT=1,2
   19 A(1,IT) =      PC(IT)*D(IT)/ABS(D(IT))
      NVAM=NVAR-1
      DO 20 IT=3,NVAM,2
      IT1=IT+1
      S = SQRT(D(IT)*D(IT)+D(IT1)*D(IT1))/CNV
      S = SIN(S     )/(S*CNV)
      A(1,IT1) = PC(IT1)*D(IT1)*S
   20 A(1,IT)=PC(IT)*D(IT)*S
      DO 22 I=1,NVAR
      K= I+NVAR
   22 A(1,K)=-A(1,I)
      RETURN
      END
```

```
// DUP
*STORE        WS  UA  SCOST
// FOR
*ONE WORD INTEGERS
      SUBROUTINE TORQ(NVAR,C,D,TC,ROLLT,YAWT,PITT)
      DIMENSION C(3,13),D(12),TC(3)
      ROLLT=0.0
      YAWT=0.0
      PITT=0.0
      DO 73 I=1,NVAR
      J=I+1
      ROLLT=ROLLT+C(1,J)*D(I)
      PITT=PITT+C(3,J)*D(I)
   73 YAWT=YAWT+C(2,J)*D(I)
      ROLLT=ROLLT-TC(1)
      YAWT=YAWT-TC(2)
      PITT=PITT-TC(3)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  TORQ
// FOR
*ONE WORD INTEGERS
      SUBROUTINE EIMM(EIMAX,EIMIN,NVAR,DD,DELT,DIMAX,DIMIN)
      DIMENSION EIMAX(12),EIMIN(12),DD(12),DIMAX(12),DIMIN(12)
      DO 57 I=1,NVAR
      EIMAX(I) = DD(I)+DELT*(DIMAX(I)-DIMIN(I))
      EIMIN(I) = DD(I)-DELT*(DIMAX(I)-DIMIN(I))
57    CONTINUE
C
      DO 51 I=1,NVAR
      TEMP =        EIMAX(I)-DIMAX(I)
      IF(TEMP)58,58,54
 54   EIMAX(I) = DIMAX(I)
 58   TEMP =        EIMIN(I)-DIMIN(I)
      IF(TEMP)59,51,51
 59   EIMIN(I) = DIMIN(I)
 51   CONTINUE
      DO 62 I=4,8    ,2
      J=I-1
      A = SQRT(EIMAX(I)*EIMAX(I)+EIMAX(J)*EIMAX(J))
      IF(A-DIMAX(I))56,56,53
 53   CONTINUE
      EIMAX(I) = EIMAX(I)*DIMAX(I)/A
      EIMAX(J) = EIMAX(J)*DIMAX(J)/A
  56  A = SQRT(EIMIN(I)*EIMIN(I)+EIMIN(J)*EIMIN(J))
      B = ABS(DIMIN(I))
      IF(A-B)52,52,55
 55   EIMIN(I)=EIMIN(I)*B/A
      EIMIN(J)=EIMIN(J)*B/A
 52   A = SQRT(EIMIN(J)*EIMIN(J)+EIMAX(I)*EIMAX(I))
      IF(A-B)60,60,61
 61   EIMIN(J) = EIMIN(J)*B/A
      EIMAX(I) = EIMAX(I)*B/A
 60   A = SQRT(EIMIN(I)*EIMIN(I)+EIMAX(J)*EIMAX(J))
      IF(A-B) 62,62,63
 63   EIMIN(I) = EIMIN(I)*B/A
      EIMAX(J) = EIMAX(J)*B/A
 62   CONTINUE
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  EIMM
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CVARP(L,NVAR,A,D,EIMIN,EIMAX,NVAR1)
      DIMENSION L(17),A(17,26),D(12),EIMIN(12),EIMAX(12)
      I=NVAR+3
      DO 55 J=1,I
      K=J+1
      M=L(K)
      IF(M-NVAR)56,56,74
 56   D(M) = A(K,NVAR1)+EIMIN(M)
      GO TO 55
 74   N=M-NVAR
      D(N) = EIMAX(N)-A(K,NVAR1)
  55  CONTINUE
      RETURN
      END
```

```
// DUP
*STORE        WS  UA  COST
// FOR
*ONE WORD INTEGERS                                                      43LPROG
*ARITHMETIC TRACE
*NAME SOLTN                                                             44LPROG
** SOLUTION PHASE                                                       45LPROG
      SUBROUTINE SOLTN(NVAR1,NVAR,D)
C     LINEAR PROGRAMMING FOR THE 1130  --SOLUTION PHASE                 46LPROG
      DIMENSION D(12)
      COMMON A(17,26),W(17),L(17),III,JJ,II,IW
      COMMON     DIMAX(12),DIMIN(12)     ,PC(12) ,DD(12)
      COMMON EIMAX(12),EIMIN(12),C(3,13),TC(3),DELT,TCOST,COST1,COST2
    2 FORMAT(2I2,F8.4)                                                  48LPROG
    5 FORMAT(I4,F20.8)                                                  49LPROG
    6 FORMAT(11H INFEASIBLE)                                            50LPROG
    7 FORMAT(24H VARIABLE           VALUE)                              51LPROG
    8 FORMAT(11H FUNCTIONAL,F20.8,//)                                   52LPROG
    9 FORMAT(24H VARIABLE      SHAD. COST)                              53LPROG
  100 FORMAT(1H )                                                       54LPROG
  101 FORMAT(35H ITER.      FUNCTIONAL          VAR. IN)                55LPROG
  103 FORMAT(9H FEASIBLE)                                               56LPROG
  105 FORMAT(I4,F15.2,10X,I4)                                           57LPROG
  130 FORMAT(10H UNBOUNDED)                                             58LPROG
      CALL DATSW (7,MISW)
      GO TO (700,701),MISW
 701  CONTINUE
      WRITE(IW,101)                                                     59LPROG
 700  CONTINUE
      KKK=0                                                             60LPROG
      CALL DATSW(1,ISW)                                                 61LPROG
      GO TO (40,22),ISW                                                 62LPROG
   22 I=1                                                               63LPROG
   23 I=I+1                                                             64LPROG
      IF(I-III)24,40,40                                                 65LPROG
   24 IF(L(I))23,25,23                                                  66LPROG
   25 DO 27 J=1,JJ                                                      67LPROG
      IF(A(I,J))26,27,26                                                68LPROG
   26 A(III,J)=A(III,J)-A(I,J)                                          69LPROG
   27 CONTINUE                                                          70LPROG
      GO TO 23                                                          71LPROG
   40 K=III                                                             72LPROG
   44 J=0                                                               73LPROG
      W(K)=0.0                                                          74LPROG
      L(K)=0                                                            75LPROG
   42 J=J+1                                                             76LPROG
      IF(J-JJ)41,45,45                                                  77LPROG
   41 IF(A(K,J))43,42,42                                                78LPROG
   43 IF(W(K)-A(K,J))42,42,47                                           79LPROG
   47 W(K)=A(K,J)                                                       80LPROG
      L(K)=J                                                            81LPROG
      GO TO 42                                                          82LPROG
   45 IF(L(K))46,62,46                                                  83LPROG
   46 KJ=L(K)                                                           84LPROG
      DO 120 I=2,II                                                     85LPROG
      IF(A(I,KJ))120,120,121                                            86LPROG
  120 CONTINUE                                                          87LPRWG
```

```
      WRITE(IW,130)
      STOP
  121 I=1
      JK=0
   50 I=I+1
      IF(I-II)52,52,56
   52 IF(A(I,KJ))50,50,51
   51 X=A(I,JJ)/A(I,KJ)
      IF(JK)55,53,55
   55 IF(X-XMIN)53,50,50
   53 XMIN=X
      JK=I
      GO TO 50
   56 X=A(JK,KJ)
      L(JK)=KJ
      DO 57 I=1,III
   57 W(I)=A(I,KJ)
      IJ=JK-1
      DO 59 I=1,IJ
      DO 59 J=1,JJ
      IF(A(JK,J))58,59,58
   58 IF(W(I))580,59,580
  580 A(I,J)=A(I,J)-W(I)*(A(JK,J)/X)
   59 CONTINUE
      IJ=JK+1
      DO 61 I=IJ,III
      DO 61 J=1,JJ
      IF(A(JK,J))60,61,60
   60 IF(W(I))600,61,600
  600 A(I,J)=A(I,J)-W(I)*(A(JK,J)/X)
   61 CONTINUE
      DO 205 J=1,JJ
  205 A(JK,J)=A(JK,J)/X
      KKK=KKK+1
      GO TO (702,703),MISW
  703 CONTINUE
      WRITE(IW,105)KKK,A(K,JJ),L(JK)
  702 CONTINUE
      CALL DATSW(4,ISW)
      GO TO (70,44),ISW
   62 IF(K-1)70,70,63
   63 IJ=JJ-1
      DO 65 J=1,IJ
      IF(A(K,J)-.0001)65,65,66
   65 CONTINUE
      WRITE(IW,103)
      DO 131 J=1,JJ
  131 A(III,J)=0.0
      K=1
      KKK=0
      GO TO 44
   66 WRITE(IW,6)
      PAUSE
   70 WRITE(IW,8)A(1,JJ)
      GO TO (704,705),MISW
  705 CONTINUE
      WRITE(IW,7)
```

```
      DO 71 I=2,II
   71 WRITE(IW,5)L(I),A(I,JJ)
  704 CONTINUE
      CALL DATSW(3,ISW)
      GO TO (72,75),ISW
   72 WRITE(IW,9)
      IJ=JJ-1
      DO 74 J=1,IJ
      IF(A(1,J))73,74,73
   73 WRITE(IW,5)J,A(1,J)
   74 CONTINUE
   75 CALL DATSW(2,ISW)
      GO TO (76,80),ISW
   76 IJ=JJ-1
C     DUMMY READ TO FEED BLANK CARD TO THE PUNCHING STATION OF THE 1442
      READ(2,5)
      DO 77 J=1,IJ
      DO 77 I=1,III
      IF(A(I,J))78,77,78
   78 WRITE(2,2)I,J,A(I,J)
   77 CONTINUE
      WRITE(2,100)
      DO 90 I=1,III
   90 WRITE(2,2)I,L(I),A(I,JJ)
80    CALL DATSW(5,ISW)
      GO TO (81,82),ISW
81    RETURN
82    CALL DATSW(6,ISW)
      GO TO (83,84),ISW
83    CALL LINK(RIGHT)
84    CALL EXIT
      END
```

Appendix D

SHUTTLE NAR DATA

SHUTTLE ASCENT DATA MAX Q

$m = 1,313,800.$

$I_{XX} = 30,812,000.$

$I_{XZ} = 12,682,000.$

$I_{ZZ} = 316,950,000.$

$S_{ref} = 249.91$

$b_{ref} = 32.8$

$\bar{g} = 31,233$

$g = 9.77$

$\theta_o = 53.2$

$U_o = 398.$

$W_o = 7.55.$

$V_T = 398.$

$C_{Y\beta} = -2.567$

$C_{Y\delta r} = .1092$

$C_{Y\delta e1} = .2781$

$C_{Y\delta e23L} = .5562$

$C_{Y\delta e4L} = 1.505$

$C_{Y\delta e5L} = 1.505$

$C_{Yr} = 0.$

$C_{YP} = 0.$

$C_{\ell\beta} = -.4257$

$C_{\ell\delta r} = .04217$

$C_{\ell\delta e1} = .07479$

$C_{\ell\delta e23L} = .1063$

$C_{\ell\delta e23D} = .02295$

$C_{\ell\delta e4L} = .2502$

$C_{\ell\delta e3L} = -.3165$

$C_{\ell r} = 0.$

$C_{\ell P} = 0.$

$C_{n\beta} = 1.249$

$C_{n\delta r} = -.1019$

$C_{n\delta e1} = -.1831$

$C_{n\delta e23L} = -.377$

$C_{n\delta e4L} = -1.158$

$C_{n\delta e5L} = -1.158$

$C_{nr} = 0.$

$C_{np} = 0.$

$C_{D_{\delta_R}} = .00035$ nt/deg

NAR CONTROL SCHEME MAX Q

- Torque Command

$$\hat{\phi} = A_{1P}\,P + A_{o\phi}\,\phi$$

$$\hat{\psi} = A_{1r}\,r + A_{o\psi}\,\psi + A_{KY}\,\ddot{Y}$$

- Signal Distribution Scheme

$$\delta e1 = \delta e23D = 2\hat{\phi}$$

$$\delta R = 5\hat{\phi}$$

$$\delta e23L = \hat{\psi}$$

$$\delta e4L = \sqrt{2}\,(\hat{\psi} - \tfrac{1}{2}\,\hat{\phi})$$

$$\delta e5L = \sqrt{2}\,(\hat{\psi} + \tfrac{1}{2}\,\hat{\phi})$$