AUTOMATED DESIGN AND OPTIMIZATION OF FLEXIBLE BOOSTER AUTOPILOTS VIA LINEAR PROGRAMMING

Authors:

Francis D. Hauser
Glen D. Szollosi
W. Steven Lakins

Prepared for:

George C. Marshall Space Flight Center
Huntsville, Alabama 35812

Contract NAS8-28482
DCN 1-2-75-20071(IF)

MARTIN MARIETTA CORPORATION
P. O. Box 179
Denver, Colorado 80201

AUTOMATED DESIGN AND OPTIMIZATION OF FLEXIBLE BOOSTER AUTOPILOTS VIA LINEAR PROGRAMMING

Authors:

Francis D. Hauser
Glen D. Szollosi
W. Steven Lakins

Prepared for:

George C. Marshall Space Flight Center
Huntsville, Alabama 35812

Approved:

Francis D. Hauser

Francis D. Hauser
Program Manager

Contract NAS8-28482
DCN 1-2-75-20071(IF)

MARTIN MARIETTA CORPORATION
P. O. Box 179
Denver, Colorado 80201

FOREWORD

This report, Volume II of 2 volumes, was jointly prepared by the Guidance and Controls Section and the Scientific Programming Section of Martin Marietta Corporation, Denver Division, under Contract NAS8-28482. Volume I contains the philosophy and the mathematical basis of the non-linear programming algorithm that led to the development of the COEBRA program. This volume is the User's Manual for the COEBRA program. The purpose of the contract was to convert the COEBRA program from the CDC 6400/6500 digital computer system to the UNIVAC 1108 at the George C. Marshall Space Flight Center, and to provide a manual and instruction on the use of the program. This contract was performed from March 1972 to December 1972, and was administered by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, under the direction of Mr. D. K. Mowery, Dynamics and Control Division, Aeroastrodynamics Laboratory.

TABLE OF CONTENTS

TABLE OF CONTENTS (Con't)

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES (Con't)

ABSTRACT

This report, Volume II of 2 volumes, is the User's Manual for the COEBRA program. Volume I contains the historical background, the philosophy, and the mathematical basis of the nonlinear programming algorithm that led to the development of the COEBRA program. COEBRA is an acronym for the Computerized Optimization of Elastic Booster Autopilots. This volume is written assuming that Volume I has been read.

COEBRA is an automatic autopilot design program. The bulk of the design criteria is in the form of minimum allowed gain/phase stability margins. COEBRA has two optimization phases: (1) a phase to maximize stability margins; and (2) a phase to optimize structural bending moment load relief capability in the presence of minimum requirements on gain/phase stability margins. The following is an outline of this report.

Chapter 1 defines the design criteria (stability margins, closed-loop roots, structural bending moment loads, trajectory drift) that COEBRA is capable of considering in its design procedure. Chapter 2 discusses the constraint equations that COEBRA places on the design criteria. Chapter 3 defines the cost function that is used in the optimization phase that maximizes stability margins. The cost function for the phase that optimizes bending moment load relief capability is presented in Chapter 4. Chapter 5 shows the constraint equations that are placed on the value of each autopilot variable. Chapters 6 and 7 illustrate how the user defines the initial or "first guess" autopilot to the COEBRA program. Chapter 8 illustrates the constraints the user can place on the autopilot parameters to yield the the drift minimum condition. Chapter 9 defines additional constraints that

can be used in an attempt to keep COEBRA from designing an autopilot that is sensitive to tolerances on the airframe/autopilot parameters. Chapter 10 summarizes the COEBRA input data, and Chapter 11 contains sample listings of the input data. Appendix A shows a simple example problem that demonstrates the mechanics of the Simplex Algorithm.

CHAPTER 1. THE MARGIN ARRAY

Section 1.1 Introduction

This chapter defines the design criteria that COEBRA is capable of considering. This criteria includes stability margins, rigid-body rotational closed-loop roots, structural bending moment loads, and trajectory drift. This criteria is listed in Table 1.1, the so-called Margin Array Table. This table, even though it contains things other than stability margins, will constantly be referred to throughout this report as the Margin Array. From this table, COEBRA sets up the cost functions (maximize margins <u>or</u> optimize load relief) and the so-called Stability Margin Constraint equations. The Margin Counter and the Figure-of-merit are also calculated from the elements in the Margin Array. Figure 1.1 of Volume I, which illustrates a typical gain/phase frequency response plot, is repeated in this volume to facilitate defining most of the elements of the Margin Array Table. Table 1.2 accompanies Figure 1.1, and contains the definitions of the gain and phase margins.

The SCALF array shown in the Margin Array Table will be defined in Section 1.3 of this chapter. The SPEC array shown in the Margin Array Table, contains the <u>minimum requirements</u> on the gain/phase stability margins. These minimum requirements are used in the so-called Stability Margin Constraint Equations (defined in Chapter 2). The COSPEC Array contains the design <u>objectives</u> for the gain/phase stability margins. These objectives are used in the so-called "Maximize Margins" cost function (defined in Chapter 3). The preset values indicated in the Margin Array Table are to be overridden by the user if he so desires.

Before defining each element of the Margin Array, this section will conclude with a brief overview of how the elements are used. Elements 1 to

TABLE 1.1 THE MARGIN ARRAY

Margin Array Element	Definition in Figure 1.1	Margin Array Element Number	SCALF Array (Preset Values)	SPEC Array		COSPEC Array	
				Preset Values	Scaled Values Corresponding to Preset SCALF	Preset Values	Scaled Values Corresponding to Preset SCALF
<u>Rigid Body</u>:							
Aerodynamic Gain Margin	$\|OA\|$	1	1 db/db	6.0(db)	1.995	6.0(db)	1.995
Rigid Body Gain Margin	$\|OE\|$	2	1 db/db	6.0(db)	1.995	6.0(db)	1.995
Rigid Body Phase Margin	$\measuredangle OB$	3	1 db/5 deg.	30.0(deg.)	1.995	30.0(deg.)	1.995
<u>"First" First Structural Mode:</u>							
Peak Gain	$\|G\|$	4	1 db/db	20.0(db)	0.1	20.0(db)	0.1
Peak Phase	$\measuredangle G$	5	1 db/9.009 deg.	----	----	----	----
Frontside Phase Margin	$\measuredangle OF$	6	1 db/5 deg.	45.0(deg.)	2.818	45.0(deg.)	2.818
Backside Phase Margin	$\measuredangle OH$	7	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
Closest Approach Margin (1 db = 4 deg.)	$\|OI\|$	8	1 db/db	10.0(db)	3.162	10.0(db)	3.162

TABLE 1.1 THE MARGIN ARRAY (Continued)

Margin Array Element	Definition in Figure 1.1	Margin Array Element Number	SCALF Array (Preset Values)	SPEC Array: Preset Values	SPEC Array: Scaled Values Corresponding to Preset SCALF	COSPEC Array: Preset Values	COSPEC Array: Scaled Values Corresponding to Preset SCALF
"Second" First Structural Mode:							
Peak Gain	$\lvert G \rvert$	9	1 db/db	20.0(db)	0.1	20.0(db)	0.1
Peak Phase	$\measuredangle G$	10	1 db/9.009 deg.	---	---	---	---
Frontside Phase Margin	$\measuredangle OF$	11	1 db/5 deg.	45.0(deg.)	2.818	45.0(deg.)	2.818
Backside Phase Margin	$\measuredangle OH$	12	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
Closest Approach Margin (1 db = 4 deg.)	$\lvert OI \rvert$	13	1 db/db	10.0(db)	3.162	10.0(db)	3.162
"First" Second Structural Mode:							
Peak Gain	$\lvert L \rvert$	14	1 db/db	20.0(db)	0.1	20.0(db)	0.1
Peak Phase	$\measuredangle L$	15	1 db/9.009 deg.	---	---	---	---
Frontside Phase Margin	$\measuredangle OK$	16	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
Backside Phase Margin	$\measuredangle OM$	17	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
180° Crossover Margin #1	$\lvert OJ \rvert$	18	1 db/db	10.0(db)	3.162	10.0(db)	3.162
180° Crossover Margin #2	$\lvert OP \rvert$	19	1 db/db	10.0(db)	3.162	10.0(db)	3.162
Closest Approach Margin (1 db = 4 deg.)	$\lvert ON \rvert$	20	1 db/db	10.0(db)	3.162	10.0(db)	3.162

TABLE 1.1 THE MARGIN ARRAY (Continued)

Margin Array Element	Definition in Figure 1.1	Margin Array Element Number	SCALF Array (Preset Values)	SPEC Array		COSPEC Array	
				Preset Values	Scaled Values Corresponding to Preset SCALF	Preset Values	Scaled Values Corresponding to Preset SCALF
<u>"Second" Second Structural Mode:</u>							
Peak Gain	$\|L\|$	21	1 db/db	20.0(db)	0.1	20.0(db)	0.1
Peak Phase	$\measuredangle L$	22	1 db/9.009 deg.	----	----	-----	----
Frontside Phase Margin	$\measuredangle OK$	23	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
Backside Phase Margin	$\measuredangle OM$	24	1 db/5 deg.	60.0(deg.)	3.981	60.0(deg.)	3.981
180° Crossover Margin #1	$\|OJ\|$	25	1 db/db	10.0(db)	3.162	10.0(db)	3.162
180° Crossover Margin #2	$\|OP\|$	26	1 db/db	10.0(db)	3.162	10.0(db)	3.162
Closest Approach Margin (1 db = 4 deg.)	$\|ON\|$	27	1 db/db	10.0(db)	3.162	10.0(db)	3.162
<u>Higher Structural Modes</u>:							
3rd Mode Peak Gain	$\|Q\|$	28	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162
3rd Mode Peak Phase	$\measuredangle Q$	29	1 db/9.009 deg.	----	----	----	----
4th Mode Peak Gain	$\|Q\|$	30	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162
4th Mode Peak Phase	$\measuredangle Q$	31	1 db/9.009 deg.	----	----	----	----
5th Mode Peak Gain	$\|Q\|$	32	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162
5th Mode Peak Phase	$\measuredangle Q$	33	1 db/9.009 deg.	----	----	----	----
6th Mode Peak Gain	$\|Q\|$	34	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162

TABLE 1.1 THE MARGIN ARRAY (Continued)

Margin Array Element	Definition in Figure 1.1	Margin Array Element Number	SCALF Array (Preset Values)	SPEC Array		COSPEC Array	
				Preset Values	Scaled Values Corresponding to Preset SCALF	Preset Values	Scaled Values Corresponding to Preset SCALF
Higher Structural Modes (cont'd.)							
6th Mode Peak Phase	$\measuredangle Q$	35	1 db/9.009 deg.	----	----	----	----
7th Mode Peak Gain	$\lvert Q \rvert$	36	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162
7th Mode Peak Phase	$\measuredangle Q$	37	1 db/9.009 deg.	----	----	----	----
8th Mode Peak Gain	$\lvert Q \rvert$	38	1 db/db	-10.0(db)	3.162	-10.0(db)	3.162
8th Mode Peak Phase	$\measuredangle Q$	39	1 db/9.009 deg.	----	----	----	----
Fuel Slosh Mode Backside Phase Margins:							
1st Slosh Mode	$\measuredangle OD$	40	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
2nd Slosh Mode	$\measuredangle OD$	41	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
3rd Slosh Mode	$\measuredangle OD$	42	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
4th Slosh Mode	$\measuredangle OD$	43	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
5th Slosh Mode	$\measuredangle OD$	44	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
6th Slosh Mode	$\measuredangle OD$	45	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
7th Slosh Mode	$\measuredangle OD$	46	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
8th Slosh Mode	$\measuredangle OD$	47	1 db/5 deg.	20.0(deg.)	1.584	30.0(deg.)	1.995
Rigid-body Phase Margin Frequency (rad/sec)	-	48	-	0.0	0.0	0.0	0.0

TABLE 1.1 THE MARGIN ARRAY (Continued)

MARGIN ARRAY ELEMENT	X	Y	MARGIN ARRAY ELEMENT NUMBER
Peak Value of Angle of Sideslip (rad)			49
Peak Value of Yaw Control Device (rad)			50
Peak Value of Roll Control Device (rad)			51
Amplitude of Case X Autopilot Feedback Loop Evaluated at the Frequency of the Y^{th} Element of the MARVEC Array:			
	7	1, 2, 3	52, 53, 54
	8	1, 2, 3	55, 56, 57
	9	1, 2, 3	58, 59, 60
	10	1, 2, 3	61, 62, 63
	11	1, 2, 3	64, 65, 66
	12	1, 2, 3	67, 68, 69

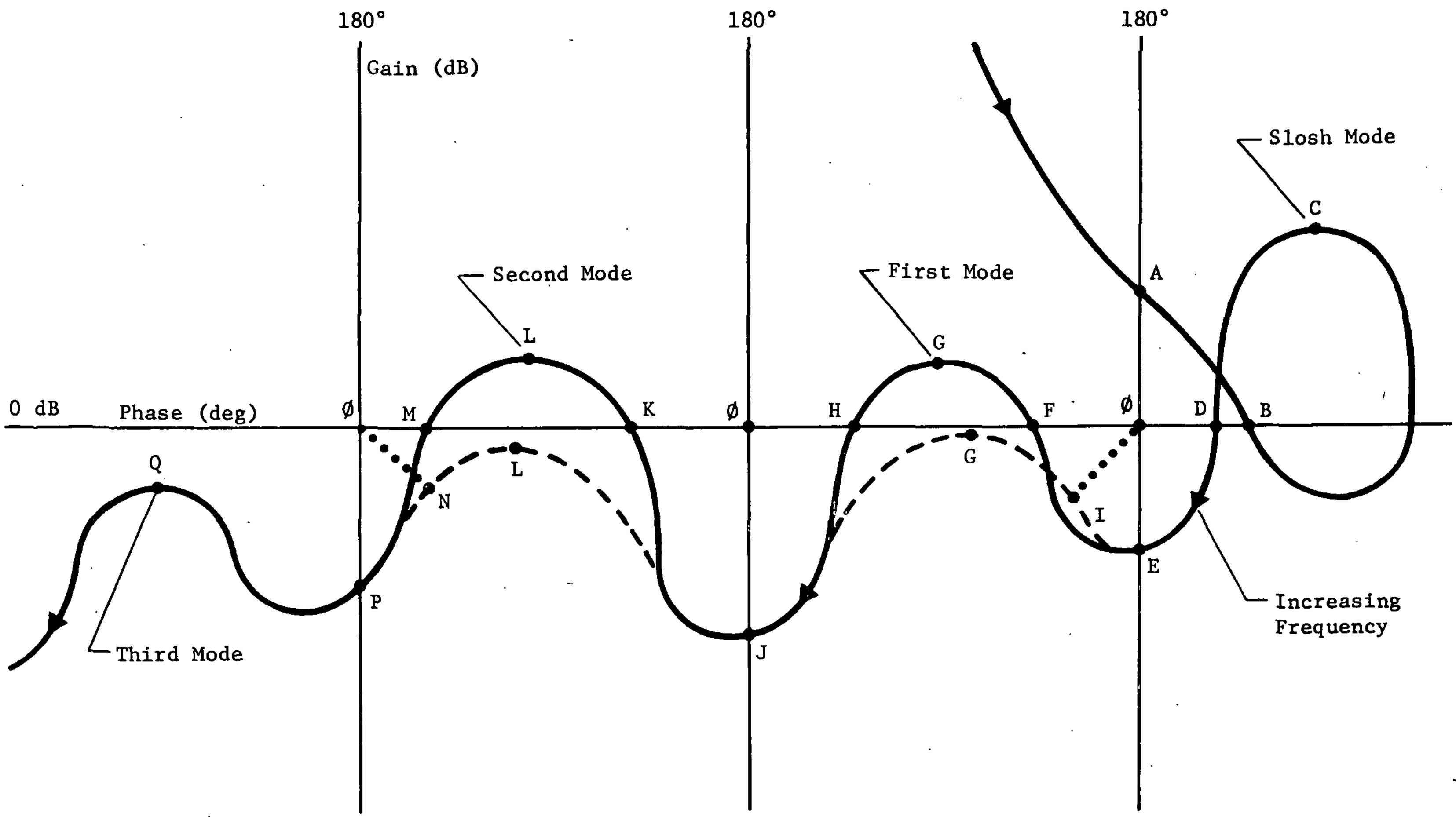


Figure 1.1. Typical Gain/Phase Frequency Response Plot

Table 1.2. Definitions of Gain and Phase Margins

Frequency Range	Reference in Figure 1.1	Typical Design Requirement	Typical Design Objective
Aerodynamic Gain Margin	$\|\phi A\|$	6 dB	6 dB
Rigid-Body Phase Margin	$\measuredangle \phi B$	30°	30°
Slosh Mode Backside Phase Margin	$\measuredangle \phi D$	20°	30°
Rigid-Body Gain Margin	$\|\phi E\|$	6 dB	6 dB
First Mode Frontside Phase Margin	$\measuredangle \phi F$	45°	45°
First Mode Peak Gain	$\|G\|$	None	20 dB
First Mode Peak Phase	$\measuredangle G$	None	0°
First Mode Backside Phase Margin	$\measuredangle \phi H$	60°	60°
First Mode Closest Approach Margin (If First Mode Peak < 0 dB)	$\|\phi I\|$	10 dB (letting 4° ≡ 1 dB)	10 dB
Second Mode 180° Crossover Gain Margin	$\|\phi J\|$	10 dB	10 dB
Second Mode Frontside Phase Margin	$\measuredangle \phi K$	60°	60°
Second Mode Peak Gain	$\|L\|$	None	20 dB
Second Mode Peak Phase	$\measuredangle L$	None	0°
Second Mode Backside Phase Margin	$\measuredangle \phi M$	60°	60°
Second Mode Closest Approach Margin (If Second Mode Peak < 0 dB)	$\|\phi N\|$	10 dB (letting 4° ≡ 1 dB)	10 dB
Second Mode 180° Crossover Gain Margin	$\|\phi P\|$	10 dB	10 dB
Third and Higher Mode Peak Gain	$\|Q\|$	-10 dB	-10 dB
Third and Higher Mode Peak Phase	$\measuredangle Q$	None	0°

48 are, at the user's option, candidates for the "Maximize Margins" cost function, where the objective is to maximize all stability margins and attempt to force all structural bending modes to resonate near zero degrees phase. Elements 49, 50, 51 are candidates for the so-called "Optimize Load Relief" cost function, where the objective is to minimize structural bending moment loads, in the presence of minimum requirements on the gain/phase stability margins. Except for the structural bending mode peak phases and elements 49, 50, and 51, <u>all</u> elements of the Margin Array are candidates for the Stability Margin Constraint Equations, where minimum allowed requirements are put on each.

<u>Section 1.2 Definitions of the Margin Array Elements</u>

(1) <u>Aerodynamic Gain Margin</u> (denoted $|OA|$ in Figure 1.1)

If the phase angle of the total open-loop frequency response at ω = SLOW is less than 180° (where SLOW is an input parameter in rad/sec), the aero gain margin will be the first crossing of the 180° axis as ω increases from SLOW. The frequency search interval is:

$$\text{SLOW} < \omega < \text{W2}$$

where W2 is the minimum of the following three frequencies.

(a) UPPER (which is an input parameter)

(b) frequency at the peak of the lowest frequency 1st structural mode

(c) frequency at the peak of the highest frequency structural mode.

If the phase angle at SLOW is less than 180°, this margin must exist or COEBRA will terminate (unless the NOTERM option of the COBIN namelist is used).

(2) Rigid-body Gain Margin (|OE|)

This will be the lowest frequency 180° crossover encountered in the following frequency interval.

$$W1 < \omega < W2$$

where $W1 = \max \left\{ \text{SLOW, aero gain margin frequency} \right\}$. If fuel slosh modes are included, the rigid-body gain margin will not be considered by COEBRA. Except for fuel slosh and the NOTERM option, COEBRA will terminate if the rigid-body gain margin does not exist.

(3) Rigid-body Phase Margin ($\measuredangle$ OB)

When searching from SLOW, the phase margin will be the first zero db crossing encountered. The following search increment is used:

$$\text{SLOW} < \omega < \text{UPPER}$$

Unless the user exercises the NOTERM option, COEBRA will terminate if the rigid-body phase margin does not exist.

(4) Structural Bending and Fuel Slosh Modes

COEBRA may be run with or without structural bending and/or fuel slosh modes. When modes are included, COEBRA can handle a maximum of eight of them. There are 4 basic types of modes, and each type is constrained differently:

(a) 1st structural bending modes;

(b) 2nd structural bending modes;

(c) 3rd and higher structural bending modes; and

(d) fuel slosh modes.

The user declares the "type" that is to be associated with each mode. He will declare the "type" depending on how he wants to constrain each mode. The modes need not be ordered according to frequency, e.g., a 3rd mode may be lower in frequency than a 1st mode, etc. Also, bending and slosh modes may be intermixed frequency-wise. There are a few rules that must be obeyed, and these will be explained where appropriate.

(4.1) First Structural Bending Modes

Each 1st structural bending mode is defined by the 5 following items:

(a) modal peak gain ($|G|$)

(b) modal peak phase ($\measuredangle G$)

(c) frontside phase margin ($\measuredangle$ OF)

(d) backside phase margin ($\measuredangle$OH)

(e) closest-approach margin $|OI|$

Stability margin constraints on the 1st mode are handled as follows. When the mode peaks below zero db, the closest approach margin will always be constrained. The mode peak gain will not be constrained and obviously, the front and backside phase margins don't even exist. When the mode peaks above zero db, and the input parameter KAPCH = 0, the closest-approach margin will not be constrained. The mode will be constrained by its peak gain, and its front and backside phase margins. When the mode peaks above zero db and KAPCH = 1, the closest-approach margin, the peak gain, and the front and backside phase margins will all be constrained.

The peak phase of the 1st mode is not constrained. It is a candidate only for the "maximize margins" cost function, where the objective is to force the mode to resonate near zero degrees phase. The modal peak gain, the front and backside phase margins and the closest approach margin are also candidates for the "maximize margins" cost function.

COEBRA has slots for 2 first structural bending modes. Both 1st modes must resonate after the rigid-body gain margin if it exists and is constrained.

This discussion will conclude with the frequency search intervals that are used for <u>all</u> structural and slosh mode peaks and for <u>all</u> modal front and backside phase margins. To find each modal <u>peak</u>, COEBRA searches between

$$0.9\omega_D \leq \omega \leq 1.1\omega_D$$

where $\omega_D = \omega_n \sqrt{1-\zeta^2}$ where ζ and ω_n are the input values for the damping ratio and undamped natural frequency of each mode. To find each front and backside phase margin, COEBRA searches between

$$0.1\,\omega_p \leq \omega \leq 10.\omega_p$$

where ω_p is the frequency at each modal peak.

(4.2) <u>Second Structural Bending Modes</u>

Each 2nd structural bending mode is defined by the 7 following items:

(a) modal peak gain ($|L|$)

(b) modal peak phase ($\measuredangle L$)

(c) frontside phase margin ($\measuredangle$ OK)

(d) backside phase margin ($\measuredangle$ OM)

(e) 180° crossover gain Margin #1 ($|OJ|$)

(f) 180° crossover gain Margin #2 (|OP|).

(g) closest-approach margin (|ON|)

Except for the 180° crossover gain margins, all elements of the 2nd structural modes are handled exactly like they are for the first structural modes. Because of the 180° crossover margins, 2nd structural modes cannot be included unless the following modes are also included:

(a) something identified as a first mode that is lower than the 2nd modes in frequency; and

(b) something identified as a 3rd or higher mode that is higher than the 2nd modes in frequency.

COEBRA allows for eight 180° crossover margins to exist between the highest frequency 1st mode and the lowest frequency higher mode. Constraints will be placed on the smallest two of them. Note that COEBRA allows for two 2nd structural modes.

(4.3) Third and Higher Structural Bending Modes

Each of these modes is defined by its:

(a) modal peak gain ($|Q|$); and

(b) modal peak phase ($\measuredangle Q$).

Each modal peak gain is a candidate for the Stability Margin Constraint Equations, and each modal peak gain and phase is a candidate for the "maximize margins" cost function. COEBRA has slots for 6 of these modes, i.e., for a single 3rd mode, a single 4th mode, . . . , a single 8th mode.

An advantage of identifying them in this way is as follows. The 3rd mode at time point #1 can have a different requirement than the 3rd mode at time point #2, by declaring, for example, that the 3rd mode at time point #2 is a 4th mode. Along this same line, the first mode at time point #2 can be declared one of the two 2nd modes, so that its requirement can be different than the first mode at time point #1.

Finally, the only restriction on 3rd and higher modes, is that they occur after the rigid-body phase margin. They may occur before the rigid-body gain margin.

(4.4) <u>Fuel Slosh Modes</u>

For each fuel slosh mode that resonates above zero db, COEBRA finds its backside phase margin ($\measuredangle$ OD). This backside phase margin is a candidate for both the Stability Margin Constraint Equations and the Maximize Margins cost function. COEBRA ignores each slosh mode that resonates below zero db. This backside phase margin is the 1st zero db crossover that is encountered when searching with increasing frequency from the slosh mode peak. It is possible for 2 or more slosh modes to have the same backside phase margin. Slosh modes must peak after the rigid-body phase margin, but slosh modes may be intermixed with the structural modes in any manner according to frequency.

As previously stated when defining the rigid body gain margin, the rigid body gain margin and fuel slosh are mutually exclusive. However, when designing without slosh, the user can bias the rigid body gain margin <u>requirement</u> in order to compensate for the effective rigid body inertia change that will result when slosh is included. But, there is a better technique for designing the gain margin with slosh. The user can input a certain time point twice with identical airframe transfer functions. The user can tell COEBRA to ignore the slosh modes at the first time point so as to design the rigid body gain margin. He can then tell COEBRA to consider the slosh modes at the 2nd time point.

The following is a closely related item, and is included here because it is an important design feature and advantage of the COEBRA algorithm. This technique of entering the same time point twice can also be used to handle severe tolerance conditions. Instead of identical transfer functions, the user can input the nominal airframe transfer functions as the 1st vehicle state. He can input the "toleranced" airframe transfer functions as the 2nd vehicle state. By treating the nominal and the toleranced airframe together, a single autopilot can be designed that will handle both conditions. Note that the "toleranced" airframe can even include malfunction conditions like actuator and sensor failures, etc.

(5) Rigid-body Phase Margin Frequency

Chapter 4, Section 4.3, of Volume I includes a discussion of how COEBRA treats the requirement on the rigid-body rotational roots. This discussion will not be repeated here, but it can be simply stated that COEBRA treats these roots by putting a requirement on the frequency as well as the magnitude of the rigid-body phase margin. The phase margin frequency is a candidate for both the Stability Margin Constraint Equations and the Maximize Margins cost function.

(6) Elements 49, 50 and 51

These elements are only used to form the so-called "optimize load relief" cost function. They are completely defined in Chapter 4, and this definition need not be repeated here.

(7) Elements 52 thru 69

These elements are used in the so-called Vector Constraint Equations that are completely defined in Chapter 9. Basically, these constraint equations are used in an attempt to keep COEBRA from designing an autopilot that is sensitive to tolerances on the airframe/autopilot parameters.

As a final note in this section, the subroutine that "finds and identifies" the margins is not completely general. If COEBRA "behaves strangely", the user should check the margin array that is printed out. The first and second structural modes are the major problem areas. The rigid-body margins are more well-behaved. The following is a tip on how the user can circumvent a rigid-body problem that can occur on an aerodynamically stable vehicle. Figure 1.2 shows a frequency response plot that might result from a vehicle that is "bare airframe" stable.

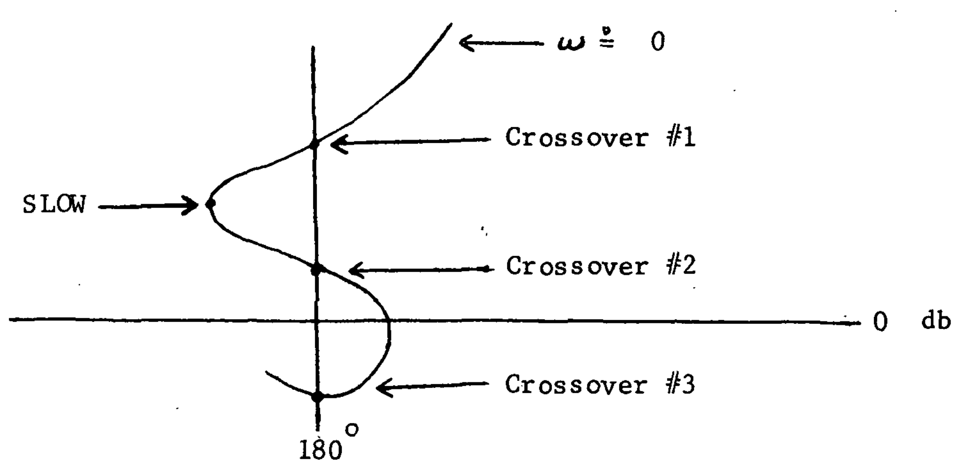


FIGURE 1.2 FREQUENCY RESPONSE FOR AERO STABLE VEHICLE

To keep COEBRA from identifying crossover #1 as the rigid-body gain margin, the user can input SLOW at the frequency indicated. In this way, crossover #2 will be called the aero gain margin, and crossover #3 will be called the rigid-body gain margin.

Section 1.3 The SCALF Array

The primary reason for the SCALF array is to scale gain and phase margins and modal peak gains and phases so that they can be equitably treated together in the maximize margins cost function. The SCALF array puts everything on a decibel scale, and from there, COEBRA converts everything to the real axis scale for the Taylor Series Expansion. In other words, all partial derivatives are computed after the conversion to the real axis scale.

For gain and phase margins, the conversion from decibels and degrees to the real axis is rather straightforward since these margins represent distances from the critical point. This includes the rigid-body gain margin where, a 180° crossover at -6 db is really a +6db margin of stability. If SCALF is 1db/db, this would convert to 1.995 on the real axis. Note that a zero db margin of stability converts to unity and a negative stability margin converts to a number less than unity on the real axis scale.

Since all margins are to be maximized, and since all mode peaks are to be "minimized", it was decided to "invert" the scale on the peak gains of all structural bending modes. In this way they can be "maximized" also. For example, if a mode resonates at +6db, and if SCALE is 1db/db, the modal peak gain converts to 0.5011 on the real axis scale.

The scale used for structural mode peak phases is also a special case. If the mode resonates at + 90° phase, and if SCALF = 1db/9 deg., the phase angle would convert to + 10db and then to +3.162. If the mode resonates at -90° phase and if SCALF = 1db/9 deg., the phase angle would still convert to +10db, but would then convert to -3.162 on the real axis scale. If the mode resonates at 0° phase, this would convert to unity on the real axis. Hence, 0° phase logically serves as the reference. This scaling was necessitated since the modal peak phase angle is not a stability margin and it is generally no more desirable for the mode to resonate at +10° than it is for the mode to resonate at -10°. Also, by scaling this way, the partial derivative of the modal peak phase is always positive in the phase lead direction.

Note that another result of the SCALF array and the corresponding conversion to the real axis, is that all Stability Margin Constraint equations for elements 1 to 48 are "greater than or equal to" type constraints. The SCALF is not applicable to the phase margin crossover frequency (element 48), since it is a real axis quantity already. Finally, the preset values indicated for SCALF in Table 1.1 are to be overridden by the user if he so desires.

Chapter 2. The Margin Constraint Equations

This chapter defines the so-called Margin Constraint Equations. Strictly speaking, the Margin Constraint Equations include constraints on items that are not exactly stability margins. For example, besides the items that can directly be classified as stability margins, these equations include constraints on:

1) The peak gains of structural bending modes;

2) The so-called Drift Minimum Autopilot condition; and

3) The autopilot vector constraints that are defined in Chapter 9.

The phrase "Margin Constraint Equations" is used simply to differentiate this type of constraint from the type that is applied to the magnitudes of the autopilot gains and filter parameters.

There are three types of so-called Margin Constraint Equations: The type that is used for (1) elements 1 to 48 of the Margin Array (defined in Chapter 1); (2) the "drift minimum" condition (defined in Chapter 8); and (3) the so-called autopilot vector constraints (defined in Chapter 9). This chapter will only define the type that is used for elements 1 to 48 of the Margin Array. The other two types (defined in their respective chapters) are formed using the same philosophy, and hence need not be repeated here.

The following discussion defines the Margin Constraint Equations that are used for elements 1 to 48 of the Margin Array. All elements 1 to 48 are constrained in this way, except those elements that are modal peak phases. Modal peak phases are not "constrained" and they are used only in the "Maximize Margins" cost function (defined in Chapter 3). The equation for the i^{th} element at the t^{th} time point, M(i), is obtained from a first order Taylor Series expansion about $M(i)_o$, which is the scaled nominal (initial) value for each major iteration. For simplicity, time notation is not used and is merely implied.

Since it is required that M(i) be greater than or equal to some specified requirement, $M(i)_s$, the margin constraint equation can be written:

$$M(i)_o + \sum_{j=1}^{n} \left[\frac{\partial M(i)}{\partial X(j)}\right]_o * \Delta X(j) \geqslant M(i)_s$$

$$(\text{for } i = 1, \ldots, m)$$

Where:

1) $\Delta X(j) = X(j) - X(j)_o$ Where: X(j) is the j^{th} autopilot variable whose value COEBRA is to optimize, and $X(j)_o$ is the nominal or initial value of X(j) at the beginning of the major iteration;

2) $\left[\partial M(i)/\partial X(j)\right]_o$ is the partial derivative of M(i) with respect to X(j) evaluated at $X(j)_o$. Note that the method of finite differences is used to calculate this partial derivative. Note also that this derivative is computed after the elements of M(i) have been scaled or converted to the "real number" axis.

There are 2 cases for $M(i)_s$:

Case 1: If the requirement on M(i) is already met, then $M(i)_s$ = SPEC (i) where SPEC (i) is the scaled requirement for M(i). For this case, the constraint is "loose";

Case 2: If the requirement on M(i) is not yet met, then $M(i)_s$ = $M(i)_o$ + STEP* [SPEC(i) - $M(i)_o$] . For this case, the constraint is "tight".

As discussed in Chapter 3, Section 3.5, of Volume I, STEP is maximized in the Inner Loop. At the beginning of each "inner" iteration, STEP begins with its input value. If no feasible solution can be obtained (i.e., if the feasible region defined by the Margin Constraints does not overlap the feasible region defined by the Autopilot Variable Constraints), STEP is reduced via

$$\text{STEP} \longleftarrow \text{STEP-DSTEP}$$

until a feasible solution is obtained. DSTEP is an input parameter. The minimum value for STEP is zero. Note that when STEP = 0, a feasible solution is virtually guaranteed since the nominal autopilot should always satisfy this condition which requires no improvement. Note also that STEP has the same value for all the "not-met" Margin Constraint Equations.

This chapter will conclude with 3 notes.

Note #1 The Margin Constraint Equations are comprised of constraints from all the time points that are being designed together.

Note #2 Any structural bending mode whose peak value is less than AMPOUT, is not included in the constraint equations for elements 1 to 48. (AMPOUT is a COBIN namelist parameter).

Note #3 For each 1st and 2nd structural mode, the constraints on the front and backside phase margins, on the modal peak gain, and on the closest-approach margin, are handled as follows:

1) If the mode peaks below zero db, the closest-approach margin will always be constrained. The modal peak will not be constrained, and

obviously, the phase margins don't even exist;

2) When KAPCH = 1, and the modal peak gain is greater than zero db, constraints will be put on the front and backside phase margins, on the modal peak gain, <u>and</u> on the closest-approach margin;

3) When KAPCH = 0, and the modal peak gain is greater than zero db, constraints will be put on the front and backside phase margins and on the modal peak gain, but <u>not</u> on the closest-approach margin.

CHAPTER 3. THE STABILITY MARGIN COST FUNCTION

This chapter contains the definition of the so-called Stability Margin cost function. The objective of this cost function is to attempt to: (1) maximize all stability margins; (2) optimize the location of the rigid-body closed-loop rotational roots; and (3) force all structural bending modes to resonate near zero degrees phase. This cost function (Y_1) is a weighted linear combination of the variable portion of the first order terms in the Taylor series expansion of each element of the Margin Array that is to be "maximized". Y_1, which is given by the following expression, is to be maximized.

$$Y_1 = \sum_{j=1}^{n} \sum_{t=1}^{m} \sum_{i=1}^{48} \mathrm{WTFAC}(i,t)*W(i,t)*U(i,t)*\left[\frac{\partial M(i,t)}{\partial X(j)}\right]_0 *X(j)$$

where:

(1) j is the index of the autopilot variables (total of n).

(2) t is the index of the time points or vehicle states (total of m).

(3) i is the index of the elements of the Margin Array (i = 1 to 48).

(4) X(j) is the j^{th} autopilot variable whose value is to be determined by COEBRA.

(5) $\left[\frac{\partial M(i,t)}{\partial X(j)}\right]_0$ is the partial derivative of the i^{th} element of the <u>scaled</u> Margin Array at time t, with respect to X(j), evaluated at $X(j)_0$.

(6) U(i,t): The purpose of this variable is to account for the following definitions on the signs of the partial derivatives. For all structural mode peak phases, the derivative is positive in the phase lead direction. For all the stability margins

including the root requirement, the derivative is positive when the scaled margin increases. Therefore: U(i,t) is unity for all elements of the Margin Array except for the structural mode peak phases. For the structural mode peak phases:

(a) U(i,t) = +1.0 if the mode peaks between 0^o and -180^o.

(b) U(i,t) = -1.0 if the mode peaks between 0^o and $+180^o$.

This is in keeping with the philosophy to seek to have all structural modes resonate near zero degrees phase. NOTE: For all structural modes, COEBRA will attempt to push each towards its nearest zero degree point.

(7) W(i,t): This is a weighting factor that is computed by COEBRA. Generally, this weighting factor is the ratio of the desired "margins" over the actual "margins". When WTYPE is unity, all the W(i,t)'s are set to unity. When WTYPE is zero, then

(a) for all elements of the Margin Array except for the structural mode peak phases:

$$W(i,t) = \frac{COSPEC(i)}{M(i,t)_o}$$

where: (1) COSPEC(i) is the scaled objective of the i^{th} margin.

(2) $M(i,t)_o$ is the scaled nominal value of the i^{th} margin at the t^{th} time point.

(b) for the structural mode peak phases:

$$W(i,t) = \frac{\text{antilog}\left\{\frac{1}{20} * SCALF(i) * \left|UM(i,t)_o\right|\right\} - 1.0}{\text{antilog}\left\{\frac{1}{20} * SCALF(i) * 90\text{ degrees}\right\} - 1.0}$$

where: (1) $|UM(i,t)_o|$ is the absolute value in degrees, of the nominal unscaled structural mode peak phase at time t, and has a value between 0^o and $+180^o$.

(2) SCALF(i) is the scale factor or multiplying factor that converts $|UM(i,t)_o|$ to decibels. The main purpose of SCALF is to scale the elements of the Margin Array so that gain margins and phase margins can be treated together.

Note that W(i,t) for the structural mode peak phases is derived from a real number scale that is referenced to 90 degrees. That is, if the mode peaks between 0^o and $+90^o$ or between 0^o and -90^o, then

$$0.0 \leq W(i,t) < 1.0$$

If the mode peaks at $+90^o$ or -90^o,

$$W(i,t) = 1.0$$

If the mode peaks between $+90^o$ and $+180^o$ or between -90^o and -180^o,

$$1.0 < W(i,t) \leq \text{(a value that depends on SCALF(i))}$$

(8) WTFAC(i,t) is a weighting factor that is input by the user, that will allow him to emphasize or de-emphasize certain elements of Y_1 relative to others. This array is an additional multiplicative weighting factor to that which is provided by the WTYPE option. This array also allows the user to "build-his-own" cost function if the "built in" cost function will not solve his problem (See Note 1 below).

The following are a series of important notes dealing with the formation of the Stability Margin cost function.

NOTE 1 The "built in" cost function is the result of the preset values for WTFAC(i,t). These values, given next, are to be overridden by the user if he so desires.

(a) 1.0 for the first three rigid body margins.

(b) 1.0 for all structural mode peak gains.

(c) 1.0 for all fuel slosh backside phase margins.

(d) 0.0 for all the other elements of the Margin Array from $i = 1$ to 48.

In other words, if the user does not exercise the WTFAC option, only the rigid body margins, the fuel slosh margins, and the structural mode peak gains will get into the optimize-margins cost function, with weighting factors given by W(i,t).

NOTE 2 WTFAC(i,t) not only allows the user to build his own cost function, and emphasize or de-emphasize certain elements, but it also allows him to correct the following situation. When a time point is entered into the problem two times, (e.g. once for fuel slosh, and once for the rigid body gain margin), WTFAC(i,t) can be used to keep identical margins from being entered into the cost function twice, which has the effect of doubling their weights.

NOTE 3 The user must be warned not to include conflicting elements in the cost function when he exercises the WTFAC option. An example of this is a dual second mode, where the first peak resonates at greater than 180° phase, and the second peak at less than 180° phase, and it is not possible to "push" them apart. NOTE: This note is premature, but will be very important on the second reading.

The IDMODE array (defined in COBDE namelist) might be used to solve the conflict in the above example, by telling COEBRA to simply ignore one of the dual second modes.

NOTE 4 The following is another example of using the WTFAC(i,t) array to keep identical margins from being entered into the cost function two or more times. It is possible for two or more fuel slosh modes to have the same backside phase margin. Hence, the WTFAC(i,t) elements for all of these slosh modes except one, can be set to zero.

Chapter 4. The Load Relief Cost Function

Section 4.1 of this chapter presents the philosophy and the formula that COEBRA uses for the cost function when optimizing the load relief autopilot. Section 4.2 shows the airframe/autopilot equations that COEBRA uses when setting up this cost function. Section 4.3 discusses the so-called wind forcing function that COEBRA uses as the input to the airframe/autopilot equations.

Section 4.1 Formula for the Load Relief Cost Function

As stated earlier, COEBRA has two phases: (1) the so-called "Maximize Margins" phase; and (2) the so-called "Load Relief Optimization" phase. The only difference between these two phases is the cost function. The matrix of constraint equations on the stability margins, on the closed-loop roots and on the autopilot variables, remains the same for either of the two phases.

Philosophy of the load relief cost function is as follows: When the objective is maximize structural bending moment load relief capability, the cost function is comprised of the response of the angle of attack (β) and the control deflections (δ) due to a wind forcing function (β_ω). When the cost function is maximized, the peak values of β and δ are minimized, thereby minimizing total structural bending moment loads.

A separate linear transient response routine is used to calculate the peak values of angle of attack (β_ρ) and control deflections (δ_ρ) due to β_ω. In order to save computer time, a linear transient response approach was used, rather than a nonlinear time-varying trajectory simulation. For illustration, if the user wants to optimize the load relief autopilot during the portion of flight from "load-relief switch-in" to "max-$\bar{q}$", he might input airframe/autopilot data at the following two time points: (1) load relief switch-in; and (2) max-$\bar{q}$. COEBRA will then optimize the autopilot based on a transient response at each of these two time points. That is, COEBRA will calculate the rate of change of β_ρ and δ_ρ with respect to each autopilot variable at the time point of load relief switch-in and at the time point of max-$\bar{q}$. COEBRA will then form the cost function from these sensitivities at these two time points. Via this method, hopefully load relief optimization will also occur at all the intermediate time points. Since COEBRA uses a linear transient response routine, the actual values of β_ρ and δ_ρ do not represent the values that would result from a nonlinear time-varying simulation. However, the actual values are not really germane. Only the trend or sensitivity with respect to the autopilot variables is important.

As will be shown later, the load relief cost function will be formed from a "weighted" linear combination of the sensitivities of β_ρ and δ_ρ. This is a reasonable definition, since it makes the cost function an equation very similar to that used to compute total bending moment loads [Harris, 4]:

Total Bending Moment Load = $K_1 \ \beta\ (t) + K_2 \ \delta \ (t) + f$ (gusts, buffets, etc.)

By a judicious choice of the wind profile that is used to force the linearized equations at each flight time, a load relief autopilot can be designed that possesses a good response characteristic to both low frequency and high frequency winds.

When the user desires to optimize load relief capability, he sets the flag LOADOP to unity and he supplies the necessary airframe/autopilot data needed to form the load relief function. As with the stability margin cost function, the load relief cost function (Y_2) is a weighted linear combination of the variable portion of the first order terms in the Taylor Series.

Y_2 is given as follows:

$$Y_2 = \sum_j \sum_t \sum_{i=49}^{51} \left\{ \text{WTFAC } (i,t) * \left[\frac{\partial M(i,t)}{\partial X(j)}\right]_o * X(j) \right\}$$

In the above expression:

(1) j refers to the summation over all the autopilot <u>variables</u> denoted X(j). (Note that COEBRA will optimize the value of X(j).)

(2) t refers to the summation over all the time points or vehicle states.

(3) i refers to the element of the so-called Margin Array table that was defined in Chapter 1. When i = 49, M (i,t) is the peak value of the angle of attack (β_ρ) at the t^{th} time point. Similarly, when i = 50, M (50,t) is the peak value of the yaw (or pitch) control device deflection ($\delta_{\psi\rho}$). Finally, M (51,t) is the peak value of the roll control device deflection ($\delta_{\phi\rho}$) if the yaw/roll coupled airframe/autopilot equations are used when setting up the cost function.

(4) WTFAC (i,t) refers to arbitrary weighting factors that are input by the user. (Note: WTFAC (i,t) for i = 49, 50 and 51 has a preset value of unity to be overridden by the user).

(5) $\left[\frac{\partial M\ (i,t)}{\partial X\ (j)}\right]_o$ is the partial derivative of M (i,t) with respect to the j^{th} autopilot <u>variable</u>, evaluated at X $(j)_o$, the present nominal value of the j^{th} autopilot <u>variable</u>. As with stability margins, the method of finite differences is used to compute the partial derivatives of β_ρ, $\delta_{\psi\rho}$ and $\delta_{\phi\rho}$ with respect to each autopilot variable.

In conclusion, when maximizing load relief capability, the COEBRA design algorithm will maximize the negative of Y_2 in the presence of the constraint equations on the minimum allowed gain/phase stability margins and closed-loop root locations, and on the allowed ranges of the individual autopilot variables. Note that multiple time point design is handled just as it is when maximizing stability margins.

Section 4.2 Load Relief Airframe/Autopilot Equations

Figure 4.1 defines the linearized airframe equations that are used to calculate β_ρ, $\delta_{\psi\rho}$, and $\delta_{\phi\rho}$. The five equations of Figure 4.1 are:

(1) The normal force equation, where $\beta(s)$ is the Laplace Transform of $\beta(t)$ and $\beta_w(s)$ is the Laplace Transform of the wind that will be defined in Section 4.3.

(2) The yaw (or pitch) moment equation where Ψ_b is the perturbated yaw attitude.

(3) The accelerometer equation which defines the output ($\dot{V}_{acc}$) of a lateral body-mounted accelerometer located a distance L_{AY} from the center of gravity.

(4) The yaw (or pitch) control law which is really the load relief autopilot control law. The parameters denoted $A_1(s)$ and $A_2(s)$ contain the gains and filters that are used in the load relief autopilot. COEBRA optimizes load relief by varying the gains and filters in this control law. Chapter 7 will discuss how the user inputs the initial values for these gains and filters. It is not necessary to use all of the parameters in this control law. Chapter 7 will also explain how additional filtering may be used in this control law. This additional filtering (e.g. filters in the attitude loop (K_D) and in the rate loops (K_{R1}, K_{R2})) can be used to calculate stability margins, but it is assumed that this additional filtering will not affect load relief capability. Hence, this additional filtering will not be used to calculate β_ρ, $\delta_{\psi\rho}$ and $\delta_{\phi\rho}$. This additional filtering will only be used to "meet" stability margin requirements.

$$
\begin{array}{l}
\text{1. NORMAL FORCE} \\
\text{2. YAW MOMENT} \\
\text{3. ACCELEROMETER} \\
\text{4. YAW CONTROL LAW} \\
\text{5. ROLL MOMENT} \\
\text{6. ROLL CONTROL LAW}
\end{array}
\begin{bmatrix}
(BS + Y_{\beta}) & (US - G_{R1}) & Y_{\delta_\psi} & 0 & (-WS - G_{R2}) & 0 \\
N'_{\beta} & S^2 & -N'_{\delta_\psi} & 0 & 0 & N'_{\delta_\phi} \\
Y_{\beta} & -L_{AY}S^2 & Y_{\delta_\psi} & 1 & 0 & 0 \\
0 & A_1(S) & 1 & A_2(S) & 0 & 0 \\
L'_{\beta} & 0 & L'_{\delta_\psi} & 0 & S^2 & -L'_{\delta_\phi} \\
0 & 0 & 0 & 0 & A_3(S) & 1
\end{bmatrix}
\times
\begin{bmatrix}
\beta(S) \\ \psi_b(S) \\ \delta_\psi(S) \\ \dot{V}_{acc}(S) \\ \phi_b(S) \\ \delta_\phi(S)
\end{bmatrix}
=
\begin{bmatrix}
BS \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
\beta_w(S)
$$

NOTE: All parameters are assumed to be in a body-axis reference system. Aerodynamic moment derivatives are referenced to the c.g. and are non-dimensionalized.

FIGURE 4-1.

LINEARIZED EQUATIONS FOR LOAD RELIEF OPTIMIZATION

YAW CONTROL LAW

$$A_1(S) = \left[K_D + K_R S - \frac{K_{DA} S^2}{(1 + \tau_1 S)(1 + \tau_2 S)(1 + \tau_3 S)} \right] K_{TVC}$$

$$A_2(S) = \left[\frac{K_{A1}}{(1 + A_1 S)(1 + A_2 S)(1 + A_3 S)} + \frac{K_{A2}(1 + A_4 S)}{(1 + A_5 S)(1 + A_6 S)(1 + A_7 S)(1 + A_8 S)} \right] \cdot K_{TVC}$$

ROLL CONTROL LAW

$$A_3(S) = \left[K_{DR} + K_{RR} S \right] K_{TVC}$$

where K_{TVC} is the actuator gain

Figure 4-1 (Continued)

(5) The roll moment equation, where ϕ_b is the linearized roll attitude. This equation can be used when yaw/roll coupling is not negligible and is needed when designing the yaw load relief autopilot.

(6) The roll control law which is also used when yaw/roll coupling is important. $A_3(s)$ contains the roll attitude gain (K_{DR}) and the roll rate gain (K_{RR}) that are used in the assumed roll control law. These gains are treated as constants since COEBRA only uses this equation to define yaw/roll coupling. In other words, K_{DR} and K_{RR} are not considered a part of the load relief autopilot.

The equations of Figure 4.1 ignore structural bending and fuel slosh modes, since the rigid-body angle of attack and control deflections are the principal factors in determining structural bending moment loads. Another reason for using only rigid-body equations and, in fact, for using only these specific airframe/autopilot equations, was to minimize computer time. The COEBRA program does not invert the matrix of Figure 4.1. The matrix was inverted by hand, and this "manual inversion" was programmed in such a manner that all COEBRA does is compute S-plane polynomial coefficients from the input data, factor these numerator and denominator polynomials, compute residues, and insert values of time in order to calculate β_P, $\delta_{\psi P}$ and $\delta_{\phi P}$. If the equations of Figure 4.1 were generalized so that COEBRA would have to invert the matrix, this would result in a marked increase in computer time, and is left as a possible future item in the development of COEBRA. In fact, the reason for optimizing the load relief autopilot via a linear transient response approach that uses rather restricted and fixed airframe/autopilot equations, is to minimize computer time. The examples contained in Volume I of this report, demonstrate that this rather restricted approach is effective for at least a certain class of vehicles like Martin Marietta Corporation's Titan booster.

The equations of Figure 4.1 also assume a single control torque source for yaw (δ_ψ) and a single control torque source for roll (δ_ϕ). By appropriate input data, these sources can be either: (1) thrust vector control via gimballed engines, secondary injection, or jet vanes; or (2) control via aerodynamic surfaces. Also, the equations of Figure 4.1 include yaw/roll coupling from inertial and aerodynamic effects, as well as from control effects.

The equations of Figure 4.1 can be used for pitch only, by simply using the indicated yaw plane notation and sign convention, and by not inputting roll data. Obviously, these equations can also be used for yaw only, by simply not inputting the roll data.

As stated earlier, Chapter 7 explains how the user inputs the initial values for the yaw (or pitch) control law gains and filters. The airframe and the roll autopilot parameters that are needed in Figure 4.1, are input via the COBDE namelist. Since the equations of Figure 4.1 are to be used for the pitch plane as well as for the yaw/roll plane, the following sign convention is used in order to avoid confusion between pitch and yaw.

The usual sign convention for forces and moments is:

(1) a positive X-axis force points "out the nose";

(2) a positive sideforce is "out the right wing";

(3) a positive normal force is "down";

(4) a positive pitching moment is "nose up";

(5) a positive yawing moment is "nose right"; and

(6) a positive rolling moment is clockwise looking forward.

With this in mind, all COEBRA data is input as positive values for an aerodynamically unstable vehicle (negative $C_{n\beta}$ or positive $C_{m\alpha}$)with a positive dihedral effect (negative $C_{\ell\beta}$), and with "normal" control torque sources. By "normal" control torque sources is meant:

(1) a positive yaw deflection produces a negative sideforce, a positive yawing moment, and a negative rolling moment; (2) a positive pitch deflection produces a positive normal force, a positive pitching moment, and zero rolling moment; and (3) a positive roll deflection produces a negligible sideforce, a negative yawing moment, and a positive rolling moment. If any of these conditions are not satisfied, the user must compensate by making the appropriate changes in the signs of the input data. For example, if the vehicle is aerodynamically stable (positive $C_{n\beta}$ or negative $C_{m\alpha}$), the corresponding COBDE namelist parameter, i.e., CNB, is input with a negative sign. Table 4.1 defines the coefficients in Figure 4.1, explains the sign convention that is used, and explains how these coefficients are related to the data that is input via the COBDE namelist. A complete summary (with definitions) of these parameters is also contained in Chapter 10.

From the equations of Figure 4.1 and the wind forcing function that will be defined in Section 4.3, COEBRA computes the 3 peak values: β_ρ, $\delta_{\psi\rho}$, and $\delta_{\phi\rho}$. The autopilot sensitivities of β_ρ, $\delta_{\psi\rho}$ and $\delta_{\phi\rho}$ from each time point are then used to form the load relief cost function. Via the so-called "LRCASE" option, the user does have control over which combination of the 3 peak values COEBRA is to compute. LRCASE is a COBDE namelist parameter, and if LOADOP =1, and:

(1) LRCASE (1) $\neq$ 0, COEBRA will compute β_ρ ;

(2) LRCASE (2) $\neq$ 0, COEBRA will compute $\delta_{\psi\rho}$; and

(3) LRCASE (3) $\neq$ 0, COEBRA will compute $\delta_{\phi\rho}$.

In addition, the user may direct COEBRA to compute the peak values, but need not allow all of them to get into the cost function, simply by exercising the WTFAC (i,t) array.

TABLE 4-1

LOAD RELIEF MATRIX COEFFICIENTS

Airframe Parameter	Definition	COEBRA Notation
B	$B = V_{rw}$ for yaw/roll and $\beta \leq 15^{\circ}$ $B = V^2_{rw}/U_r$ for pitch and $\alpha \leq 90^{\circ}$	B
U	Linear velocity along the X body axis	U
W	Linear velocity along the Z body axis	W
L_{AY}	Accelerometer moment arm $L_{AY} = X_{cg} - X_{acc}$ Sign convention: LAY is positive if the accelerometer is forward of the c.g.	LAY
G_{R1}	Gravity component $G_{R1} = g \sin(\theta_o)$ Where g is gravitational acceleration (9.807 m/sec^2) Define θ_o = vehicle attitude relative to local horizontal	GR1 = G*SIN(THETA)
G_{R2}	Gravity component $G_{R2} = g \cos(\theta_o) \cos(\phi_o)$ Define ϕ_o = euler roll angle	GR2 = G*COS(THETA)* COS(PHI)
Y_β	Yaw aerodynamic sideforce coefficient $Y_\beta = \dfrac{\bar{q}\ S_F\ C_{Y\beta}}{M}$	YB = $\dfrac{\text{QBAR*SF*CYB}}{\text{MASS}}$ CYB is input as positive if positive β produces negative sideforce or positive α produces negative normal force.

TABLE 4-1 (CONTINUED)

LOAD RELIEF MATRIX COEFFICIENTS

Airframe Parameter	Definition	COEBRA Notation
N'_β	Yaw aerodynamic torque coefficient (primed coefficient) $$N_\beta = \frac{\bar{q}\, S_F\, b\, C_{n\beta}}{I_{zz}}$$ **Note 1**: The "primed coefficients" are defined by $$N'_i = \frac{N_i + (I_{xz}/I_{zz})\, L_i}{(1 - I_{xz}^2/I_{xx}\, I_{zz})} \quad \text{YAW COEFFICIENT}$$ $$L'_i = \frac{L_i + (I_{xz}/I_{xx})\, N_i}{(1 - I_{xz}^2/I_{xx}\, I_{zz})} \quad \text{ROLL COEFFICIENT}$$ Obviously for the uncoupled case ($I_{xz} = 0$) $$N'_i = N_i \text{ and } L'_i = L_i$$	NB = <u>QBAR*SF*D*CNB</u> / IZZ (If IXZ is input as non-zero, NB is corrected to the primed coefficient form.) CNB is input as positive if vehicle is aerodynamically unstable. CNB is defined in the body axis system and is referenced about the c.g. of the vehicle.
L'_β	Roll aerodynamic torque coefficient (primed coefficient) $$L_\beta = \frac{\bar{q}\, S_F\, b\, C_{\ell\beta}}{I_{xx}}$$ and if yaw/roll inertial coupling is input, L'_β is computed as per Note 1. $C_{\ell\beta}$ is assumed to be in the body axes about the c.g.	LB = <u>QBAR*SF*D*CLB</u> / IXX (If IXZ is input as non-zero, LB is corrected to the primed derivative form) CLB is input as positive if dihedral effect is positive.

TABLE 4-1 (CONTINUED)

LOAD RELIEF MATRIX COEFFICIENTS

Airframe Parameter	Definition	COEBRA Notation
$Y_{\delta\psi}$	Yaw sideforce due to yaw control deflection. For TVC: $Y_{\delta\psi} = \frac{T_Y}{M}$ For Aerodynamic Surface: $Y_{\delta\psi} = \frac{\bar{q}\, S_F\, C_{Y_{\delta_R}}}{M}$ NOTE 2: If CYDR is input as non-zero, COEBRA assumes YDψ to be from an aero surface. This rule applies to <u>all</u> such yaw and roll matrix coefficients: $Y_{\delta\psi}$, $N_{\delta\psi}$, $N_{\delta\phi}$, $L_{\delta\psi}$ and $L_{\delta\phi}$.	NOTE: $Y_{\delta\psi}$ <u>cannot</u> be a combination of TVC and aero surface. The user must choose one or the other. TVC: YDψ = $\frac{\text{TY}}{\text{MASS}}$ Aero Surface: YDψ = $\frac{\text{QBAR*SF*CYDR}}{\text{MASS}}$ CYDR is input as positive if positive yaw deflection produces negative sideforce, or positive pitch deflection produces positive normal force.

TABLE 4-1 (CONTINUED)

LOAD RELIEF MATRIX COEFFICIENTS

Airframe Parameter	Definition	COEBRA Notation
$N'_{\delta\psi}$	Yaw torque due to yaw control deflection. For TVC: $N_{\delta\psi} = \frac{T_Y L_{GY}}{I_{zz}}$ For Aerodynamic Surface: $N_{\delta\psi} = \frac{\bar{q} S_F b C_{n_{\delta_r}}}{I_{zz}}$ The "primed coefficient" is then computed as in Note 1.	TVC: NDψ = TY*LGY / IZZ Aero Surface: NDψ = QBAR*SF*D*CNDR / IZZ CNDR (or LGY) is input as positive for pitch and yaw. See Note 2.
$L'_{\delta\psi}$	Roll torque due to yaw control deflection. For TVC: $L_{\delta\psi} = \frac{T_Y Z_{OFF}}{I_{xx}}$ For Aerodynamic Surface: $L_{\delta\psi} = \frac{\bar{q} S_F b C_{\ell\delta_r}}{I_{xx}}$ The "primed coefficient" is then computed as in Note 1.	TVC: LDψ = TY*ZOFF / IXX Aero Surface: LDψ = QBAR*SF*D*CLDR / IXX CLDR (or ZOFF) is input as positive if positive yaw deflection produces negative rolling moment. See Note 2.

TABLE 4-1 (CONTINUED)

LOAD RELIEF MATRIX COEFFICIENTS

Airframe Parameter	Definition	COEBRA Notation
$N'_{\delta\phi}$	Yaw torque due to roll control deflection. For TVC: $N_{\delta\phi} = 0$ For Aerodynamic Surface: $N_{\delta\phi} = \frac{\bar{q}\ S_F\ b\ C_{n\delta_a}}{I_{zz}}$ The "primed coefficient" is then computed as in Note 1.	TVC: NDϕ = 0 Aero Surface: NDϕ = QBAR*SF*D*CNDA / IZZ CNDA is input as positive if positive aileron deflection produces negative yawing moment. See Note 2.
$L'_{\delta\phi}$	Roll torque due to roll control deflection. For TVC: $L_{\delta\phi} = \frac{T_R\ L_{GR}}{I_{xx}}$ For Aerodynamic Surface: $L_{\delta\phi} = \frac{\bar{q}\ S_F\ b\ C_{\ell\delta_a}}{I_{xx}}$ The "primed coefficient" is then computed as in Note 1.	TVC: LDϕ = TR*LGR / IXX Aero Surface: LDϕ = QBAR*SF*D*CLDA / IXX CLDA (or LGR) is input as positive. See Note 2.

This section will conclude with the following note. The 3 peak values (β_ρ , $\delta_{\psi\rho}$ and $\delta_{\phi\rho}$) are computed via the S-plane equations of Figure 4.1. When designing an analog autopilot, stability margins, etc., are also computed via the S-plane. But with the digital autopilot, margins, etc., are computed via the W-plane. Therefore, with the digital system, autopilot filters must be transformed between the S and W-planes for compatibility between the subroutine that calculates the W-plane frequency response and the subroutine that calculates the time response via S-plane equations.

Because of the simplified model, only the filters in the accelerometer and in the derived attitude acceleration feedback loops require transformation via the following equation:

$$(\text{Time constant in S-plane}) = \frac{T_s}{2} * (\text{Time constant in W-plane})$$

where T_s is the sampling period. This simplified formula is valid since the filters in the load relief loop and in the derived attitude acceleration loop always have extremely low break frequencies. The gains in the autopilot require no transformation.

Section 4.3 The Wind Forcing Function

The wind forcing function is a four segment piecewise-linear function of time. A typical shape for β_ω (t) is shown in Figure 4.2, where the user inputs the slopes denoted as U1, U2, U3, and U4, and the break-times denoted as TW1, TW2 and TW3.

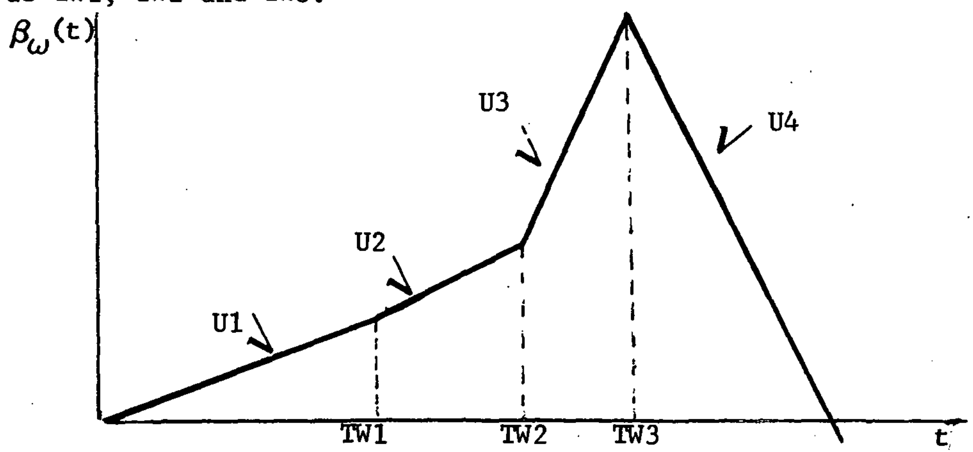


Figure 4.2 The Wind Forcing Function

The general shape of β_ω(t) is fixed, but the user has control over such things as the "size" or "weighting" of the wind shear relative to the total wind profile. As can be seen, β_ω (t) is a series of steps and/or ramps that can be made to approximate the shape of the commonly used synthetic wind profile [Harris, 4]. As stated earlier, by a judicious choice of this wind profile, a load relief autopilot can be designed that possesses a good response characteristic to both low frequency and high frequency winds.

By careful selection of the slopes and break times, the user may choose any relative effective frequency content that he wishes. The following values represent the typical wind profile that is preset in COEBRA, to be overridden by the user.

(a) U1 = .0021 rad/sec.

(b) U2 = .01 rad/sec.

(c) U3 = .0228 rad/sec.

(d) U4 = -.0038 rad/sec.

(e) TW1 = 40. sec.

(f) TW2 = 52.5 sec.

(g) TW3 = 55. sec.

To minimize machine time used in computing the time responses, a full response is computed only for the nominal case. This "full response" is computed from TSTART to TSTOP. When computing the peaks for the partial derivatives, COEBRA searches only a very small time region: The time at which the nominal case peak occurred is stored and a search is done in a small region about this time for the disturbed case. The two input parameters of interest here are DELT which is the computational time increment, and DINCRE which is the number of increments on either side of the time of the nominal peaks over which COEBRA searches for the disturbed peak. The preset values (to be overridden by the user) are:

TSTART = 30 sec

TSTOP = 70 sec

DELT = .05 sec

DINCRE = 10

All nominal case time responses are plotted for each time point. Also, the peaks of β, δ_ψ and δ_ϕ need not occur at the same time.

Chapter 5. Autopilot Variable Constraint Equations

Section 5.1 The Constraint Equation

For each autopilot variable, X(j), the following is the detailed expression of the autopilot variable constraint equation.

$$\mathrm{MAX}\left\{ (1+P)^{-1} * X(j)_o,\ \ XMIN(j)\right\} \leqslant X(j) \leqslant \mathrm{MIN}\left\{ (1+P)*X(j)_o\ ,\ XMAX(j)\right\}$$

for j = 1, ..., n

Where:

1) X(j) is the j^{th} autopilot variable, whose value is to be optimized by COEBRA. $X(j)_o$ refers to the value of X(j) at the beginning of the so-called Major Iteration. Note that the point defined by $X(j)_o$ for all j, is the point about which the partial derivatives are computed, and the Taylor Series is expanded.

2) P refers to the autopilot variable step-size, and has the same value for all the autopilot variables.

3) The XMIN(j) and XMAX(j) arrays refer to the minimum and maximum values ever allowed for X(j).

In words, if XMIN and XMAX are not encountered on a particular iteration, the above constraint equation says that X(j) is allowed to vary no more than about $\pm$P% from $X(j)_o$ on any iteration. Since it is desirable to maximize the step-size on each iteration, thereby getting the maximum "mileage" out of each set of partial derivatives, it is desirable to have a Minor Loop that increases the size of P until improvement in that "search direction" is no longer possible. In other words, the Minor Loop serves to maximize the the autopilot variable step-size. In maximizing P, the Minor Loop uses two "indicators": (1) a counter that keeps track of the number of stability margins that are already met, and (2) a figure-of-merit that is a linear combination of the actual margins. The so-called Margin Counter and Figure-of-Merit are used as follows:

1) If the value of the Margin Counter increases, the value of P can be increased regardless of the value of the Figure-of-Merit.

2) If the value of the Margin Counter does not change, the Figure-of-Merit is used to decide whether P should be increased or decreased.

3) If the value of the Margin Counter decreases, P must be reduced.

The procedure for changing the value of P, is either to double it or to average the present P with the "best-so-far" P.

In the optimization of the value of P, the input parameter PSMALL serves as a convergence criteria on two items:

1) PSMALL is the smallest value allowed for P. COEBRA will terminate when P becomes less than PSMALL, since this means that the present nominal autopilot (the autopilot at the beginning of this major iteration) is the best autopilot that can be obtained.

2) PSMALL is the smallest value allowed for ΔP which is defined as $\left| P(\text{present}) - P(\text{last}) \right|$. When ΔP becomes less than PSMALL, COEBRA will begin the next major iteration.

By inputting PSMALL with the same value as the initial value of P, the minor loop will be bypassed. The initial value of P will then be used unchanged on each <u>major</u> iteration.

As discussed in Chapter 3, Section 3.5, of Volume I, a major benefit of the minor loop is that it allows the algorithm to converge steadily to an "interior" optimum. This is explained as follows. Since the solution to the linear programming problem always lies at a vertex of the feasible region defined by the constraint equations, it is the Minor Loop that allows the algorithm to converge to a local optimum that is interior to the stability margin constraint equations.

Section 5.2 Definitions of the Margin Counter and Figure-of-Merit

Referring to the Margin Array of Chapter 1, the only elements of that array that can be included in the Margin Counter and the Figure-of-Merit are:

1) The 3 rigid-body margins and the phase margin crossover frequency;
2) For the 1st and 2nd structural modes, the front and backside phase margins, and the closest-approach;
3) For the 3rd to the 8th structural modes, the modal peak gain; and
4) For the fuel slosh modes, the backside phase margins.

Section 5.2.1 The Margin Counter

The i^{th} candidate element of the Margin Array, M(i), is counted by the Margin Counter if: (1) M(i), exceeds its "toleranced" SPEC value; and 2) the i^{th} element of the WTMARG array is not equal to zero. The so-called "toleranced" SPEC value is defined as

$$(1\text{-TOLMAC}) * \text{SPEC}(i)$$

The tolerance is applied to SPEC in order to avoid oscillations in the Margin Counter.

NOTE: WTMARG and TOLMAC are input parameters.

Table 5.1 shows the so-called weighting values given to each element in the Margin Counter. For example, if the 3 rigid-body margins and the frontside phase margin of a stable 1st structural mode were the only elements that exceeded their "toleranced" SPEC values, the Margin Counter would equal 7. This assumes that the corresponding values of the WTMARG array were not equal to zero.

Table 5.1 Weighting Values in the Margin Counter

Weighting Value	Candidate Element of Margin Array
2	for each of the 3 rigid-body margins
1	for each front and backside phase margin of the 1st and 2nd structural modes if the modes are stable.*
2	for each closest-approach margin of the 1st and 2nd structural modes if the modes are stable.*
2	for each peak gain of structural modes 3 to 8.
2	for each backside phase margin of the fuel slosh modes.
2	for the rigid-body phase margin crossover frequency.

*If a first or second structural mode is unstable, nothing is counted for that mode. Also, the front and backside phase margins are mutually exclusive with the closest-approach margin. If the mode resonates above zero db, only the front and backside phase margins are eligible to be counted. If the mode resonates below zero db, obviously only the closest-approach margin is eligible.

Section 5.2.2 The Figure-of-Merit

Whereas there is only one Margin Counter, there are two Figures-of-Merit: (1) a figure-of-merit (FM_1) for the phase of COEBRA where stability margins are to be optimized; and (2) a different figure-of-merit (FM_2) for the phase of COEBRA where the objective is to maximize structural bending moment load relief capability. For the "optimize margins" phase:

$$FM_1 = \sum WTMARG(i)*MIN \left\{ COSPEC(i),\ M(i) \right\}$$

where:

1) The summation is carried out over all time points, and over all the "candidate" elements of the Margin Array for i = 1 to 48. These "candidate" elements are summarized in Table 5.2;

2) WTMARG is an array of weighting factors that is input by the user; and

3) COSPEC is the array of desired objectives, that is also included in the Cost Function (Chapter 3).

In other words, as discussed in Volume I, the figure-of-merit (FM_1) is "rewarded" for each value of i, only up to a certain value (given by COSPEC). Note also that COSPEC and M(i), are scaled when they are included in the figure-of-merit.

For the "load relief optimization" phase:

$$FM_2 = \sum WTMARG(i)*M(i)$$

Where:

1) The summation is carried out over all time points, and for i = 49, 50, and 51 (depending on the LRCOST array).

2) M(49) is of course the peak value of the angle of sideslip. M(50) and M(51) are the peak values of the control deflections in yaw and roll respectively.

Table 5.2 Candidate Elements for FM_1

(1)	Each of the 3 rigid-body margins.
(2)	For <u>stable</u> 1st or 2nd structural modes, the closest approach margin.
(3)	For <u>unstable</u> 1st or 2nd structural modes, the unstable front <u>or</u> backside phase margin.
(4)	The peak gain of structural modes 3 to 8.
(5)	The backside phase margin of each fuel slosh mode.
(6)	The rigid-body phase margin crossover frequency.

Section 5.3 Termination

This section discusses the 4 ways in which the optimization process can be terminated.

1) The first way might be referred to as self-termination. This occurs when P becomes less than PSMALL due to the following two situations:

   a) when the Margin Counter always decreases for any value of P that is greater than PSMALL; and

   b) when the figure-of-merit (FM) is used to "break ties" and the following relationship is not satisfied:

   FM (present) $\geq$ (1+TOLFGM)*FM(last)

   where TOLFGM is an input parameter.

   This relationship will not be satisfied when all M(i) exceed their COSPEC values (which means that no further improvement is desired), or when all M(i) do not increase sufficiently (which means that no further improvement is possible). For this latter case, if all margins do not yet meet their requirements, the user must then either (1) try another initial condition for the autopilot variables, (2) add more complexity to the autopilot, and/or (3) relax some of the design requirements and/or alter some of the design objectives.

2) The second method of termination is when NITER major iterations have been completed. (NITER is an input parameter.)

3) The third way is when the user's estimated computer run time is exceeded.

4) The fourth way is due to various error messages like:

a) shutdown because the user's first guess autopilot does not lie within the range of XMIN to XMAX; and

b) shutdown due to the nearly impossible situation where STEP (defined in Chapter 2) goes to zero and no feasible solution can be achieved.

Chapter 6. The QD030 Phase

The user inputs the initial autopilot for COEBRA via the INDATA namelist of the QD030 phase of the program. However, the QD030 phase may be run by itself. This chapter describes the QD030 phase.

Section 6.1 The QD030 Block Diagram

Figure 6.1 is the block diagram that is used by the QD030 phase of the program. COEBRA assumes that the autopilot gains and filters are located in Blocks A thru N. When running COEBRA, the airframe transfer functions (including the actuator/engine) may appear anywhere in Blocks A thru U. But normally, they will be input in Blocks 0 thru U, with the denominator input into Block 0, and the numerators input into Blocks P thru U. When running the QD030 by itself, airframe and autopilot transfer functions may, of course, appear anywhere.

Obviously, all transfer functions may be defined in either the S-plane or the W-plane. COEBRA assumes everything in Blocks A thru U is input in Bode form, i.e., time constants, effective damping ratios, and undamped natural frequencies. However, when running QD030 by itself, only Block 0 thru U need be in Bode form. For the QD030 phase by itself, Blocks A thru N may contain data in polynominal form.

COEBRA optimizes the total open-loop frequency response. The QD030 by itself can compute 13 different open and closed-loop transfer functions, and run frequency and transient responses on any of them. The closed-loop transfer functions assume that all feedback loops are closed, and that the system is forced by θi. The open-loop transfer functions assume that $\theta i = 0$, and that the open-loop forcing function is δ_i. It is emphasized again that

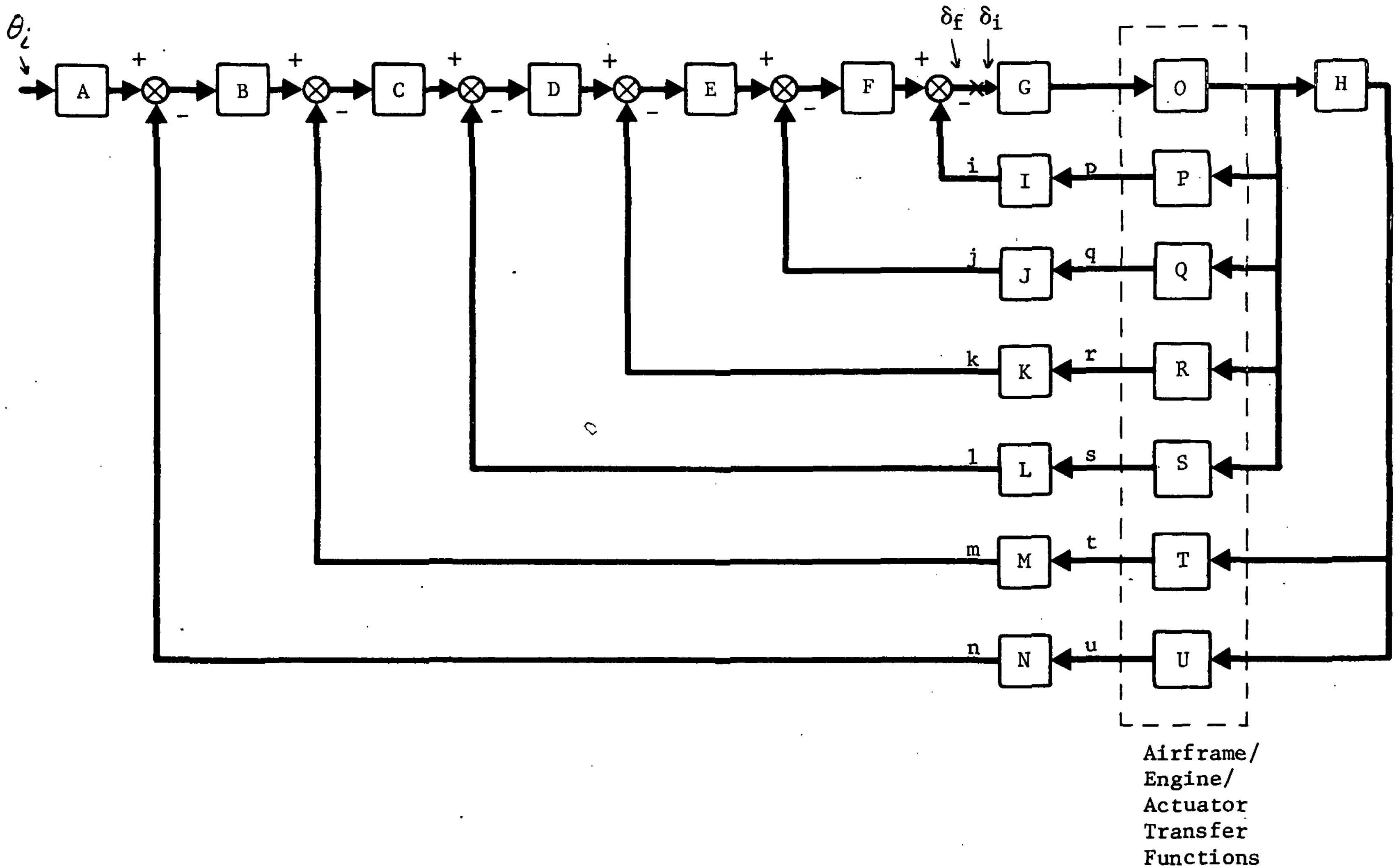


Figure 6.1 ALLOWED AUTOPILOT CONFIGURATION (QD030 BLOCK DIAGRAM FORMAT)

COEBRA deals only with the total open-loop transfer function defined by δ_f / δ_i. Table 6.1 summarizes the 13 open and closed-loop transfer functions that the QD030 phase can compute.

As mentioned earlier, COEBRA assumes everything is input in Bode form. Table 6.2 lists the INDATA namelist names that are associated with Bode form input to Blocks A thru U. Following the INDATA namelist name, the data in each block is input in the following order:

1) No. of real roots in the numerator
2) No. of complex pairs in the numerator
3) No. of free S's (or free W's) in the numerator
4) No. of real roots in the denominator
5) No. of complex pairs in the denominator
6) No. of free S's (or free W's) in the denominator
7) Bode gain
8) Thru . . . .

numerator real roots (τ)

numerator complex pairs (ζ, ω)

denominator real roots (τ)

denominator complex pairs. (ζ, ω)

For example, assume the following Bode form transfer function is put into Block C:

$$C = (3.0) \frac{(1+0.4S)\ (1+0.3S)\ (S)\ (S)}{(1+0.2S)\ (1+ \frac{2(0.1)S}{20.} + \frac{(S)}{(20.)} \frac{(S)}{(20.)})}$$

Associated input:

RC = 2., 0., 2., 1., 1., 0., 3., .4, .3, .2, .1, 20.,

Table 6.1 QD030 Transfer Functions

Case Number	Closed-Loop Cases
1	p/θ_i
2	q/θ_i
3	r/θ_i
4	s/θ_i
5	t/θ_i
6	u/θ_i
Case Number	**Open-Loop Cases**
7	i = GOPI
8	j = FGOQJ
9	k = EFGORK
10	1 = DEFGOSL
11	m = CDEFGOHTM
12	n = BCDEFGOHUN
13	δ_f/δ_i = i+j+k+1+m+n

Table 6.2 INDATA Namelist Names for the Bode Form

INDATA Namelist Name	Block
RA	A
RB	B
RC	C
:	:
RO*	O
:	:
RT	T
RU	U

* Block O is restricted to denominator roots only.

Table 6.3 lists the INDATA namelist names that are associated with so-called polynominal form input that can be used in Blocks A thru N when running the QD030 by itself. Following the INDATA namelist name, the data in each block is input in the following order:

1) The degree of the numerator
2) The degree of the denominator
3) The gain factor
4) Thru . . . .

   All coefficients of the numerator in ascending orders of powers

   All coefficients of the denominator in ascending orders of powers

For example, assume the following polynomial form transfer function is put into Block C:

$$C = (6.0) \quad \frac{(5.0S^3 + 3.0S^2 - 1.0)}{(4.0S^2 - 2.0S)}$$

associated input:

PC = 3., 2., 6., -1., 0., 3., 5., 0., -2., 4.,

Section 6.2 Additional INDATA Namelist Variables

This section will define the additional INDATA namelist variables besides those contained in Tables 6.2 and 6.3. It is noted that in the optimization phase, COEBRA uses only the variables in Table 6.2 and the variables defined below as SLOW and UPPER. Of course, all of the INDATA namelist variables are applicable when running the QD030 phase by itself. If preset values are indicated, they can be overridden by the user if he so desires.

Table 6.3 INDATA Namelist Names for Polynomial Form

INDATA Namelist Name	Block
PA	A
PB	B
PC	C
•	•
•	•
•	•
PM	M
PN	N

6.2.1 SLOW, Floating point, preset value = .01

All frequency responses will be calculated using SLOW (radians per second) as the lower limit on the frequency.

6.2.2 UPPER, Floating point, preset value = 1000.

All frequency responses will be calculated between the limits SLOW and UPPER (radians per second).

6.2.3 NCASE, Fixed point

The transfer functions (or cases) to be computed. Refer to Table 6.1.

6.2.4 NFREQ, Fixed point

Having computed the transfer functions (via NCASE), NFREQ specifies those transfer functions on which a frequency response is to be calculated. If plots are desired, the negative of the transfer function number must be input.

6.2.5 NTRAN, Fixed point

Having computed the transfer functions (via NCASE), NTRAN specifies those transfer functions on which a transient response is to be calculated. If plots are desired, the negative of the transfer function number must be input.

6.2.6 BT, Floating point

The starting time for the transient response.

6.2.7 DT, Floating point

The time increment for the transient response.

6.2.8 FT, Floating point

The final time for the transient response.

6.2.9 NUMZ, Fixed point

Number of free S's (or W's) to be added to the numerator before computing the transient response.

6.2.10 KDENZ, Fixed point

Number of free S's (or W's) to be added to the denominator before computing the transient response.

6.2.11 TOL, Floating point

Tolerance by which roots in the numerator and denominator are cancelled before computing the transient response.

6.2.12 DAMP, Floating point

Any underdamped quadratic roots in the denominator of the total open loop transfer function (CASE 13) that have an effective damping ratio with an absolute value less than DAMP, are considered to be bending or fuel slosh modes. The namelist variable called the IDMODE array is then used to identify the modal type (e.g., 1st mode, etc.) to be associated with each quadratic pair that is selected by DAMP. DAMP serves to eliminate from consideration as modes, things such as sensor dynamics, autopilot filters and prefilters, and even certain modes including perhaps the engine.

6.2.13 AMPOUT, Floating point

Modes that are selected by DAMP whose peaks resonate below AMPOUT, are not considered when calculating stability margins via the INDATA variables called KASK, KVECT, and KVECPL. Units on AMPOUT are in decibels, i.e., if AMPOUT = -20., modes that resonate below -20. decibels will not be considered.

6.2.14 IDMODE (k), Integer Array (8 dimensional)

Referring to the underdamped quadratic roots in the denominator of the total open loop transfer function (CASE 13), any root-pair whose damping ratio is less than DAMP is considered to be from a structural bending or fuel slosh mode. The program lists these bending and fuel slosh modes according to frequency, beginning with the lowest frequency. The IDMODE (k) array is then used to identify these modes according to this ordered list, beginning with the lowest frequency.

IDMODE (k) = 1, if the k^{th} mode is a first structural mode.

IDMODE (k) = 2, if the k^{th} mode is a second structural mode.

etc., until

IDMODE (k) = 8, if the k^{th} mode is an eighth structural mode.

IDMODE (k) = 10 or higher, if the k^{th} mode is a fuel slosh mode.

IDMODE (k) = 0, if the k^{th} mode is to be ignored.

NOTE: When identifying these modes, they need not be ordered according to frequency, e.g., the 1st mode may be higher in frequency than the 3rd mode, or the 4th mode may be lower in frequency than the slosh modes, etc. The exception to this is the 2nd mode, which <u>must</u> be run with something identified as a 1st mode that is <u>lower</u> in frequency, and something identified as a higher mode that is higher in frequency.

<u>6.2.15 KASK</u>, Fixed point, preset value = 1

If KASK = 1, this enables printout at the end of each CASE 13 frequency response of the following stability margins:

1) Aerodynamic gain margin
2) Rigid-body gain margin
3) Rigid-body phase margin
4) The following modal margins, where IDMODE and DAMP identify whether the modes are structural or fuel slosh.
    a) Amplitude of all structural mode peaks

b) Front and backside phase margins of the 1st and 2nd structural modes.

c) Backside phase margins of the slosh modes.

If KASK = 0, this flag is disabled.

6.2.16 KVECT, Fixed point, preset value (disable value) = 0

If KVECT = 1:

1) Polar coordinate plots of the open loop cases asked for by NFREQ are generated at the frequency of each of the stability margins generated by KASK. Since this option works in conjunction with KASK, KASK must be set to unity.

2) Polar vector plots will be generated at up to 10 user specified frequencies supplied by FQVECT in radians per second.

3) In addition to the above, a summary of the plotted vectors is printed out.

6.2.17 FQVECT, Floating point array (10 dimensional)

FQVECT is an array of frequencies (in radians per second) that is input by the user. At each of these frequencies, the program will compute and plot all the vectors that are called for by NFREQ.

6.2.18 KVECPL, Fixed point, preset value (disable value) = 1

If KVECPL = 0, the plots generated by the KVECT option will be suppressed. However, the vector plot summary will still be printed out.

6.2.19 NFLAGG, Fixed point

If NFLAGG = 0, this implies that immediately following this QD030 case will be another QD030 case that will use the same data that is input into Blocks 0 thru U, but different data for Blocks A thru N. Hence, Blocks A thru H will be reset to unity, and Blocks I thru N will be reset to zero.

If NFLAGG = 1, all QD030 Blocks will be reset to their initial values, i.e., Blocks A thru H will be reset to unity, Blocks I thru N will be reset to zero, Block 0 will be reset to unity, and Blocks P thru U will be reset to zero.

6.2.20 In conclusion, the options NCASE, NFREQ, NTRAN, KASK, etc., are not executed during COEBRA's optimization phase, but can be executed when the optimization process terminates. In other words, when optimization is terminated, the program will return to the INDATA namelist and execute all the options that are called for.

Section 6.3 QD030 Input Instructions

1) If QD030 is run without the COEBRA optimization phase, the first card must be:

Col.	1	2	3	4	5	6
		Q	D	0	3	0

2) The second card is a title card and will be printed at the top of each page of output. There must not be a $ as the first character of the title card in column two.

3) The title card is followed by the input data as defined in Sections 6.1 and 6.2.

The input data must have a $ as the first character in the first card. The dollar sign is placed in column two and is immediately followed by the word INDATA. $INDATA must be followed by at least one blank. The data is then placed in columns 10 thru 80. The data items must be separated by commas: however, the use of a comma after the last item is optional.

The end of the data group is indicated by $END beginning in column 2 of a new card.

If more than one card is needed for input data, the last item of each card <u>must</u> be constant followed by a comma. Blanks are permissible between the comma and the next data value. Blanks are prohibited between digits of a data value. The succeeding card must not have a $ in column two and may start in any column except column one. Each successive card will continue with the data elements of input. The data on a continuation card may end in any column.

The form data may take is either as its input variable name or its equivalent array element name. The form is variable name = constant. Subscripts must be integer constants. When the input is in an array in sequentially ascending steps, the form is variable name = constant, constant, ...

4) Transfer function input to Blocks 0 thru U must be in Bode form. Block 0 is used for the common denominator of Blocks P thru U as is suggested by their relative positions in the block diagram. Therefore, <u>only</u> denominator dynamics are allowed in Block 0. If only a portion of the common denominator of Blocks P thru U is input to Block 0, the program will extract the remaining common roots and store them in Block 0.

5) When running QD030 by itself, transfer function input to Blocks A thru N may be in polynomial <u>or</u> Bode form.

   NOTE: When running COEBRA's optimization phase, only Bode form may be used.

6) The program initializes all forward loop blocks (A thru H and 0) to a unity transfer function, and all feedback blocks (I thru N and P thru U) to a zero transfer function.

Section 6.4 QD030 Output Data

The transfer functions that are computed by the QD030 are output in three (3) forms:

1) Form A, Polynomial Form:

$$G = \frac{(Gain)S^k\ (1+a_1S+a_2S^2+. . . a_\ell S^\ell)}{S^m(1+b_1S+. . .b_nS^n)}$$

2) Form B, Bode form:

First lines: time constants and free S's (or W's) (denoted by zeros)

Last lines: pairs of damping ratios and natural frequencies.

3) Form C, Root Locus Form:

First lines: frequencies of the real roots

Last lines: pairs of real parts of the roots and imaginary parts of the roots.

The following is a list of the output in the order in which it appears:

1) Title and page number
2) Input data for Blocks A thru U
3) Transient responses
   a) Case number
   b) Transfer function in Forms A, B and C
   c) Partial Fraction Expansion
   d) Transient Response

4) Frequency Responses

a) Case number

b) Transfer function in Forms A, B and C

c) Frequency Response

5) Stability Margin Summary and Printout of Vector Plots

Section 6.5 Error Comments

This section briefly defines some error comments that may appear.

xxxxUSER INPUT ERRORxxxx This is the heading that will be output if the program finds an error in the input. Following this heading the user will find one of the following messages. These messages are explained below:

1) Element xx is equal to zero.

2) In the xx element, the absolute value of the effective damping ratio of the complex pair is LE 0 or GE 1.

3) Gain is equal to zero.

4) Highest degree of polynomial is equal to zero.

5) Inconsistency between NTYPE array and root input.

6) Inconsistency in IDMODE array input.

The messages 1) and 2) are generated by subroutine CKROOT. This routine checks the user's root input for proper form. These messages will be printed directly below the input if they are applicable to that case.

The messages 3) and 4) are generated by subroutine CKPOLY. This routine checks the user's polynominal input for proper form. These follow the output as above.

Message 5) is generated by subroutine CKLPRV. This routine checks for consistency between NTYPE array input and the root input.

Message 6) is generated by subroutine CKIDEN. This routine checks the IDMODE array for proper form.

All of the user's input data is checked, and if any of these messages appear, the program is terminated and user should correct all input errors before resubmittal.

CHAPTER 7. ALLOWED AUTOPILOT CONFIGURATION

This chapter is written assuming that the reader has read Chapter 6, and understands how to use the INDATA namelist. The INDATA namelist is used to input the initial autopilot. Section 7.1 discusses how the user defines this initial autopilot to the COEBRA optimization routines. Section 7.2 contains additional items that are required in order to define the configuration of the load relief and/or drift minimum autopilots. Section 7.3 contains a discussion of the so-called ROOTCH subroutine that allows COEBRA to change a quadratic filter into 2 single break filters, or vice versa.

Section 7.1 Defining the Initial Autopilot

This section discusses how the user defines the initial autopilot to COEBRA. The items in this section are applicable to both phases of COEBRA, namely, stability margin optimization and so-called load relief optimization. Section 7.2 will discuss the additional items that are needed to define the load relief autopilot to COEBRA.

The user inputs the autopilot parameters via the INDATA namelist, into any of the QD030 blocks A through N (see Figure 6.1 in Chapter 6). COEBRA deals with the autopilot parameters in Bode form, i.e., time constants for single break filters, and effective damping ratios and undamped natural frequencies for quadratic break filters.

In order to allow for multiple time point design, three types of designations are assigned to the autopilot parameters (assigned by the NTYPE(j) array defined later). Each autopilot parameter (Bode gain, time constant, damping ratio, or natural frequency) is one of the following:

(1) Constant, whose value is not to be altered by COEBRA.

(2) Stand-alone variable, whose value, when altered by COEBRA, need not be the same as the parameter that appears in the same block and element location at the immediately preceeding time point.

(3) Common variable, whose value, when altered by COEBRA, must be the same as the variable that appears in the same QD030 block and element location at the preceeding time point.

The user may input parameters (variables and constants) and free S's into any of the QD030 blocks A through N (see Figure 6.1). The following discussion shows the general configuration allowed for the autopilot variables. In each block, a maximum of 41 variables is allowed, and these may be configured as follows:

$$K \frac{(10 \text{ - 1st order functions}) \quad (5\text{-2nd order functions})}{(10 \text{ - 1st order functions}) \quad (5\text{-2nd order functions})}$$

Note that this is the maximum configuration allowed for the variables (stand-alones and commons). Any parameter that appears outside the allowed field must be designated as a constant. Also, free S's may appear, but are not assigned a parameter type. The first order functions are specified by a single parameter, the time constant (τ), and the second order functions by two parameters, effective damping ratio (ζ) and undamped natural frequency (ω).

For example, if, in the numerator of a block, it is desired to have 10 variable single break filters, 1 constant single break filter, 1 variable quadratic pair of roots, 1 constant quadratic pair of roots, and a free S, the numerator will be configured as follows:

$$(\tau_1)\ (\tau_2)\ (\tau_3)\ (\tau_4)\ (\tau_5)\ (\tau_6)\ (\tau_7)\ (\tau_8)\ (\tau_9)(\tau_{10})(\tau_{11})\ (\zeta_1,\omega_1)(\zeta_2,\omega_2)S$$

Where: (1) τ_1 through τ_{10} are the variable single break filters.

(2) τ_{11} must be the constant single break filter.

(3) (ζ_1,ω_1) may be designated as either the variable or the constant quadratic pair.

(4) (ζ_2,ω_2) may also be designated as either the variable or the constant quadratic pair.

NOTE #1: ζ and ω of a quadratic pair need not have the same type designation. For example, ζ may be a constant and ω may be a variable or vice versa. Obviously, ζ and ω may both be constants or may both be variables.

NOTE #2: COEBRA is dimensioned for a maximum of 60 parameters (not including free S's) per time point, including a maximum of 50 variables (stand-alones and commons).

NOTE #3: All of the gains and filters that are denoted as variables must have positive values. Constants may have negative values. If a variable gain must have a negative value, the minus sign must be accounted for in another QD030 block where the minus gain can be designated as a constant. For example, if the variable gain in Block K must be negative, the user must put minus-one gains (designated as constants) in blocks D and E.

The following is a discussion of how the parameter "type designations" are assigned.

In the input namelist called COBDE, which is input for each time point:

(1) NTOTAL is the total number of parameters (variables and constants) for this time point;

(2) NTYPE(j) is a compressed array that identifies the type of each parameter at this time point. This array corresponds

to the parameters in the order in which they are input in the QD030 (INDATA namelist). This array must identify even those constants that lie outside the allowed field.

NTYPE(j) = 0 if the j^{th} parameter is a constant.

= 1 if the j^{th} parameter is a common variable.

= -1 if the j^{th} parameter is a stand-alone variable.

Note that for the first time point entered, only 0's and -1's can be used in the NTYPE(j) array.

NTOTAL and NTYPE(j) are used to identify gains, time constants, damping ratios and natural frequencies, but are not used to identify free S's. Finally, at the beginning of each run, QD030 blocks A through H are each initialized with unity gain and no roots. QD030 blocks I through N are each initialized with zero gain and no roots. Hence:

(a) if in blocks A through H, any block contains roots (including free S's) and/or a gain not equal to unity, then NTOTAL and NTYPE(j) must identify the parameters in that block. If the user desires a block (A through H) to contain simply a variable gain whose initial value is unity, he must enter the gain with a value of one-plus-epsilon.

(b) if in blocks I through N, any block contains a nonzero gain, the parameters in that block must be identified by NTOTAL and NTYPE(j) . NTOTAL and NTYPE(j) must not identify blocks that do not fit into categories (a) and (b) above. Obviously, NTOTAL and NTYPE(j) assign the "types" whether the user is optimizing margins or optimizing load relief.

Section 7.2 Defining the Load Relief (and Drift Minimum) Autopilot Configuration

The section discusses the additional items that are needed to define to COEBRA, the configuration of the load relief autopilot and/or the drift minimum autopilot. These items are in addition to those presented in Section 7.1.

As discussed in Chapter 4, COEBRA uses the control law of Equation 7.1 to optimize load relief capability. (Chapter 8 will show that the same control law is assumed when designing a drift minimum autopilot.)

$$\Delta\delta_{\psi} = (K_D + K_{R1}S + K_{R2}S)(-\Delta\psi_b) + \left[\frac{K_{DA}S^2}{(1+\tau_1 S)(1+\tau_2 S)(1+\tau_3 S)}\right](\Delta\psi_b) + \left[\frac{K_{A1}}{(1+A_1 S)(1+A_2 S)(1+A_3 S)} + \frac{K_{A2}\,(1+A_4 S)}{(1+A_5 S)(1+A_6 S)(1+A_7 S)(1+A_8 S)}\right](-\Delta\dot{V}_{ACC})$$

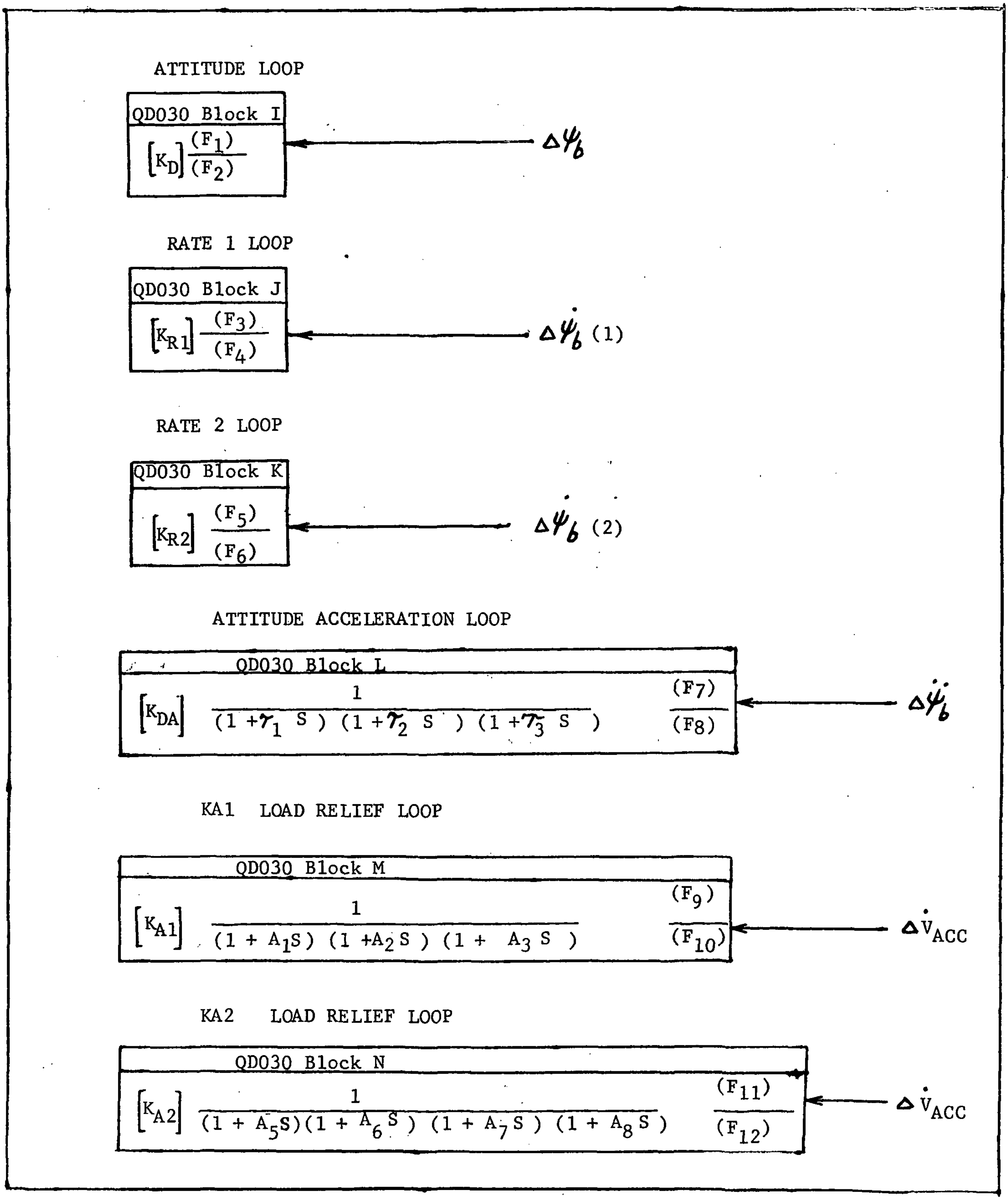
(Equation 7.1)

Equation 7.1 uses yaw plane notation, but it is also applicable to pitch provided that yaw plane notation and sign convention are used. All gains in Equation 7.1 have positive signs.

The COEBRA program must be told how to equate the autopilot parameters in the above control law with the autopilot parameters in the INDATA namelist. This will be done via the following example.

Suppose that the autopilot is input into the QD030 block diagram as shown in Figure 7.1. The filter networks that are denoted F_i do not get into the load relief cost function, but they do contribute to the open-loop frequency response. Hence the F_i that are declared to be variables (by the NTYPE(j) array) will get into the constraint matrix. Note that the F_i might be used for other things besides bending and fuel slosh filter networks. They might also be used for such things as sensor dynamics, noise prefilters, etc.

FIGURE 7.1 LOAD RELIEF AUTOPILOT EXAMPLE #1



The COBDE namelist name called LKD specifies in Hollerith form, the QD030 location of the parameter K_D. For Figure 7.1, the user would input

LKD = 5HRI(7)

Before continuing with the definition of the parameters in Figure 7.1, the COBDE namelist names associated with the remaining load relief control law parameters are given in Table 7.1.

TABLE 7.1

COBDE NAMELIST NAMES FOR THE LOAD RELIEF PARAMETERS

LOAD RELIEF CONTROL LAW PARAMETER (EQUATION 7.1)	COBDE NAMELIST NAME ASSOCIATED WITH ITS QD030 LOCATION
K_D	LKD
K_{R1}	LKR1
K_{R2}	LKR2
K_{DA}	LKDA
τ_1	LT1
τ_2	LT2
τ_3	LT3
K_{A1}	LKA1
A_1	LA1
A_2	LA2
A_3	LA3
K_{A2}	LKA2
A_4	LA4
A_5	LA5
A_6	LA6
A_7	LA7
A_8	LA8

Table 7.2 will now summarize how the user will equate the autopilot parameters in Figure 7.1 with the parameters in the load relief (and drift minimum) control law of Equation 7.1. The following assumptions are made in forming Table 7.2: (1) The filter networks denoted as F_7, F_9 and F_{11} are assumed to be first order filters and hence need only one parameter (namely, the time constant) to define them; and (2) the filter networks denoted as F_8, F_{10}, and F_{12} are assumed to appear as the last elements in the denominators of their respective blocks.

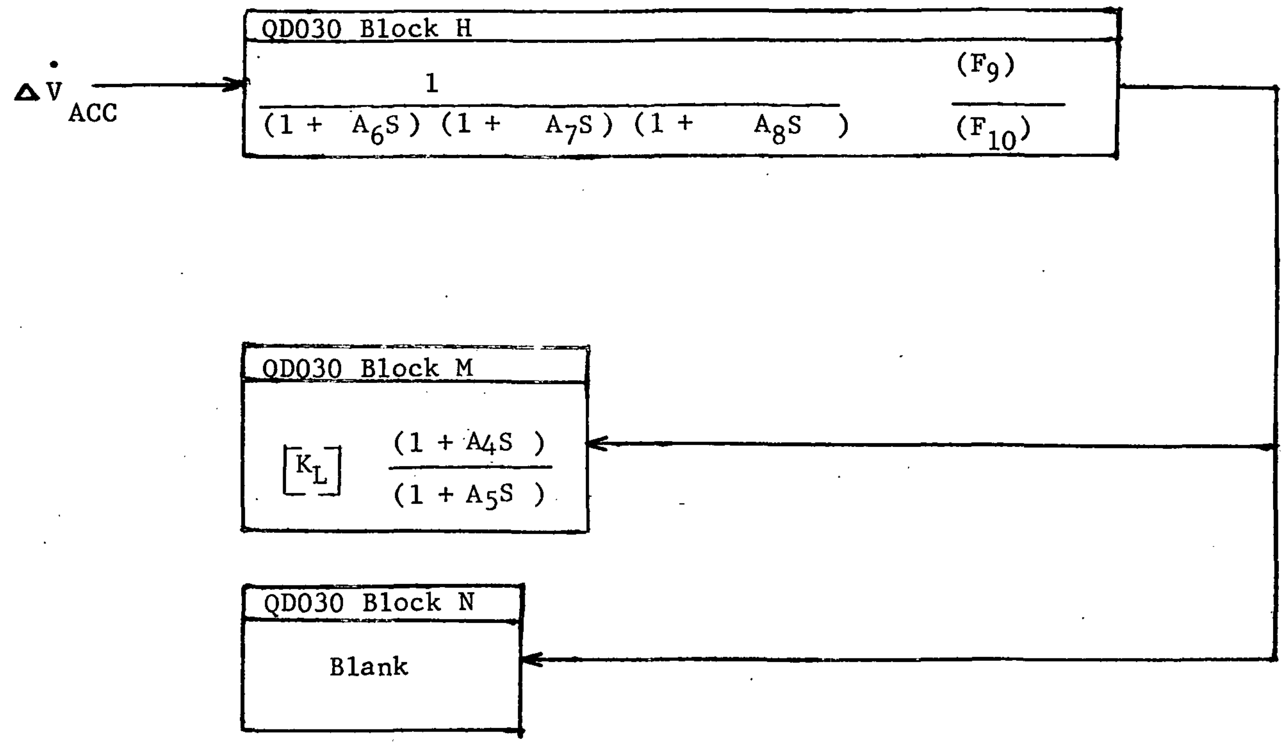
For this example, the time constant A_4 was not used and hence need not be declared. A_4 is needed when the KA1 and the KA2 feedback loops share the same filters. To illustrate this condition, suppose the QD030 blocks M and N of Figure 7.1 are replaced with QD030 blocks H, M and N of Figure 7.2. Table 7.3 summarizes how the user equates the parameters of Figure 7.2 with the parameters of Equation 7.1. In Table 7.3, it is assumed that F_9 is a first order filter, and F_{10} appears as the last element in the denominator of Block H.

TABLE 7.2

LOAD RELIEF AUTOPILOT EXAMPLE #1

LKD	=	5HRI(7)
LKR1	=	5HRJ(7)
LKR2	=	5HRK(7)
LKDA	=	5HRL(7)
LT1	=	5HRL(9)
LT2	=	6HRL(10)
LT3	=	6HRL(11)
LKA1	=	5HRM(7)
LA1	=	5HRM(9)
LA2	=	6HRM(10)
LA3	=	6HRM(11)
LKA2	=	5HRN(7)
LA4	is not used and hence is undeclared	
LA5	=	5HRN(9)
LA6	=	6HRN(10)
LA7	=	6HRN(11)
LA8	=	6HRN(12)

FIGURE 7.2 LOAD RELIEF AUTOPILOT EXAMPLE #2



where, using the notation of Figure 7.1:

$$K_L = K_{A1} + K_{A2}$$

$$A_4 = (K_{A1})(A_5) / K_L$$

TABLE 7.3

LOAD RELIEF AUTOPILOT EXAMPLE #2

LKA1		
LA1		
LA2		not used and hence not declared
LA3		
LKA2	=	5HRM(7)
LA4	=	5HRM(8)
LA5	=	5HRM(9)
LA6	=	5HRH(9)
LA7	=	6HRH(10)
LA8	=	6HRH(11)

The user must exercise care when specifying the autopilot configuration of Figure 7.2, since the combining of the K_{A1} and K_{A2} loops into a single channel may result in undesirable load relief optimization by COEBRA. This is explained as follows. The K_{A2} feedback loop is a pseudo-integration loop on the accelerometer signal. That is,

$$K_{A2} \quad \frac{1}{(1 + A_5 S)} \equiv (K_v \tau) \frac{1}{(1 + \tau S)}$$

where τ is typically a 7.5 sec. time constant, and K_v is the loop gain.

Since $K_L = K_{A1} + K_{A2} \equiv K_a + K_v \tau$

and $A_4 \equiv \tau_L = (K_{A1}) (A_5) / K_L \equiv K_a \tau / K_L$

then if the autopilot parameters K_L and τ_L are disturbed by a factor of P, the perturbated equations are (assuming that τ of the psuedo integration circuit is a constant):

$$(K_L + P K_L) = (K_a + \Delta K_a) + (K_V + \Delta K_V) \tau$$

so that

$$\Delta K_a = P K_L - (\Delta K_V) \tau$$

and

$$(\tau_L + P \tau_L) = \frac{(K_a + \Delta K_a) \tau}{K_L + \Delta K_a + (\Delta K_V) \tau}$$

which gives

$$\left[\begin{array}{l} \Delta K_V = \dfrac{P}{\tau} \left[K_V \tau - K_a (1 + P) \right] \\[2ex] \Delta K_a = P (2 + P) K_a \end{array}\right.$$

We see that K_a increases by more than P and K_V can effectively increase <u>or</u> decrease depending on the nominal or initial values of K_a and $K_V \tau$.

As a final note, each of the parameters in Equation 7.1 (or Table 7.1) can be designated as a common or stand-alone variable, or as a constant. This "type designation" is done by the NTYPE(j) array which declares the parameters in the order in which they appear in QD030 blocks A through N. Obviously, not all of the parameters in Equation 7.1 need be used. Also, the time constants in Equation 7.1 need not be the first elements in the numerators and denominators of the respective QD030 blocks.

Section 7.3 The ROOTCH Subroutine

There is a subroutine in COEBRA called ROOTCH that allows the filters to change planes. For example, a quadratic filter can become 2 real break filters, if COEBRA seeks to drive its damping ratio greater than unity. Also, 2 real break filters can become a quadratic if COEBRA drives them toward one another. When using ROOTCH (by setting the flag KROOT = 1), the allowed autopilot configuration is not as general as it is when not using ROOTCH. When using ROOTCH, filters should only be designated as commons or constants (except for the first time point when commons are entered as stand-alones). Also, the commons must be entered before the constants.

Obviously, when a filter changes planes, COEBRA adjusts the XMAX(j) and XMIN(j) arrays accordingly. As discussed in Chapter 5, the user inputs to COEBRA the minimum (XMIN(j)) and maximum (XMAX(j)) values that are allowed for each stand-alone autopilot variable. When a quadratic filter becomes 2 real break filters, COEBRA will use for the min-max values of the real-filter time constants, the inverted min-max values of the quadratic filter break frequency. When 2 real filters become complex, the inverted min-max values of the time constants are applied to the frequency of the quadratic filter. The min-max values for the damping ratio (ζ) of the quadratic filter will be set to 1.0E-4 and 0.9999 respectively.

CHAPTER 8. THE DRIFT MINIMUM CONSTRAINT EQUATIONS

The purpose of this chapter is to (1) illustrate the derivation of the so-called "Drift Minimum" principle and (2) show in detail how COEBRA constrains the autopilot gains and filters to meet this Drift Minimum criteria.

Section 8.1

Hoelker [5] and Greensite [3] derive the Drift Minimum Principle using pitch plane notation. This section will outline this derivation using yaw plane notation. The five linearized or perturbated yaw plane equations are as follows:

1. The equation for the linear acceleration of the center of gravity of the vehicle perpendicular to the trajectory or standard path:

$$\ddot{Y} = (g_{*}\sin\theta_L)\psi_b + (Y_\beta)\beta + (Y_\delta)\delta$$

2. The moment equation about the vehicle's center of gravity:

$$\ddot{\psi}_b = (N_\beta)\beta + (N_\delta)\delta$$

3. The kinematic equation relating β and ψ_b to the inertial angle $(\dot{Y}/V_I)$ and the angle of sideslip of the wind (β_W):

$$\beta \approx -\psi_b + (\dot{Y}/V_I) + \beta_W$$

4. The equation for the output of a lateral body-mounted accelerometer located a distance L_A forward of the vehicle's center of gravity:

$$\dot{V}(X) = (Y_\beta)\beta + (L_A)\ddot{\psi}_b + (Y_\delta)\delta$$

5. The equation defining the autopilot control law:

$$\delta = K_{TVC}\left[K_D(\psi_c - \psi_b) - K_R\dot{\psi}_b + K_{\ddot{\psi}}\ddot{\psi}_b - \left(K_a + \frac{K_v\tau}{1+\tau s}\right)\dot{V}(X)\right]$$

Note that in Section 7.2 of Chapter 7, the following notation was used:

$$K_{DA} \equiv K_{\ddot{\psi}}$$

$$K_{A1} \equiv K_a$$

$$K_{A2} \equiv K_v\tau$$

$$A_5 \equiv \tau$$

In this equation, all of the gains have positive values.

As shown by Hoelker [5], these five equations can be manipulated into the following form:

$$(K_4)\dddot{Y} + \ddot{Y} = (K_3)\dddot{\psi}_b + (K_2)\ddot{\psi}_b + (K_1)\dot{\psi}_b + (K_o)\psi_b$$

(Equation 8.1)

where K_0, K_1, K_2, K_3, and K_4 are constants and functions of the airframe and autopilot parameters.

Equation 8.1 can be written:

$$\ddot{Y} = \frac{K_3}{1+K_4 s}\dddot{\psi}_b + \frac{K_2}{1+K_4 s}\ddot{\psi}_b + \frac{K_1}{1+K_4 s}\dot{\psi}_b + \frac{K_0}{1+K_4 s}\psi_b$$

(Equation 8.2)

Now the response of the vehicle to any disturbance can be thought of as being made up of two kinds of motion, rotation and translation. The frequency of the "rotational mode" is almost always five to ten times higher than the frequency region of the "translational modes". Hence, assuming that changes in the wind velocity and in the lateral linear velocity ($\dot{Y}$) are negligible during a cycle of the rotational mode, and assuming that the rotational mode is relatively well damped, a so-called quasi-steady state condition will be reached soon after any disturbance. During this quasi-steady state period, $\dddot{\psi}_b$, $\ddot{\psi}_b$ and $\dot{\psi}_b$ will be negligible, and Equation 8.2 then becomes:

$$\ddot{Y} = \frac{K_0}{1+K_4 s}\psi_b$$

If $K_0 = 0$, then $\ddot{Y}$ will also equal zero during this quasi-steady state period. The condition, $K_0 = 0$, is referred to as the Drift Minimum condition since the vehicle theoretically will not be accelerating away from the trajectory following a wind disturbance. Writing K_0 in terms of airframe and autopilot parameters, the Drift Minimum condition is:

$$K_0 = -(K_D)\left(\frac{K_{TVC}*A}{V_I}\right) + (K_a+K_v\tau)\left(\frac{K_{TVC}*A}{V_I}\right)(g*\sin\theta_L) - \left(\frac{N_\beta * g*\sin\theta_L}{V_I}\right) = 0$$

(Equation 8.3)

In equation 8.3, $A = Y_\beta N_\zeta - N_\beta Y_\zeta$

Equation 8.3 is a linear relationship between K_D, K_a, and the gain referred to as $(K_v\tau)$. Note that this linear relationship is independent of the magnitude of the wind velocity. The Drift Minimum phenomenon can be looked on as a condition in which ψ_b, β, and ζ are blended or combined via the autopilot feedback loops in such a way that the forces normal to the trajectory sum to zero. In other words, the forces due to gravity, aerodynamics and control, cancel one another when summed normal to the trajectory.

In general, the characteristic equation (Δ) of this system factors as follows:

$$\Delta = (s^2 + 2\zeta\omega s + \omega^2)\ (s+\omega_1)\ (s+\omega_2)$$

The quadratic pair corresponds to the so-called dominant rigid-body closed-loop roots (the dutch-roll mode), and ω_1 and ω_2 are usually real roots that lie near the origin. Even though all four roots contribute to both rotational and translational motion, usually the quadratic roots of the dutch roll mode represent most of the rotary motion, and as shown by Hoelker [5], the roots given by ω_1 and ω_2 represent most of the path motion. This being the case, Δ can be approximated by the following polynomial:

$$\Delta \doteq s^4 + (2\zeta\omega)s^3 + (\omega^2)s^2 + (\omega^2(\omega_1+\omega_2))\ s + \omega^2\omega_1\omega_2$$

As might be expected, K_0 is related to the constant term $(\omega^2\omega_1\omega_2)$.

In fact,

$$\frac{K_0}{\tau} \doteq \omega^2\omega_1\omega_2$$

(Equation 8.4)

Hence, setting K_o to zero is the same as putting one of the roots that represents path motion right at the origin. This discussion leads to another insight into the drift minimum principle. When $K_o = 0$, let's refer to the root that goes to the origin as ω_2. Hoelker [5] shows that if ω_2 is in the right half plane (unstable), the vehicle will accelerate into the wind. If ω_2 is in the left half plane (stable), the vehicle will accelerate with the wind.

Section 8.2

This section will show in detail how COEBRA constrains the autopilot gains and filters to meet the Drift Minimum criteria.

Section 8.2.1

This subsection contains the derivation of COEBRA's Drift Minimum Constraint Equations. As shown by Equation 8.3, the Drift Minimum criterion is:

$$K_o = a(K_D) + b(K_a) + b(K_v\tau) + c = 0$$

where a, b and c are constants and functions of the airframe parameters, and the gains(K_D, K_a, and $K_v\tau$) all have positive values. Now, at the beginning of each iteration of COEBRA, $K_D = K_{Do}$

$$K_a = K_{ao}$$

and, $(K_v\tau) = (K_v\tau)_o$

Hence, $(K_o)_o = a\,K_{Do} + b\;K_{ao} + b\,(K_v\tau)_o + c$

Now, expanding K_o in a first order Taylor Series, it is desired that

$$K_o + \Delta K_o = 0 \pm \epsilon$$

where $\Delta K_o = \dfrac{\partial K_o}{\partial K_D}\Delta K_D + \dfrac{\partial K_o}{\partial K_a}\Delta K_a + \dfrac{\partial K_o}{\partial(K_v\tau)}\Delta(K_v\tau)$

$= a(K_D - K_{Do}) + b\,(K_a - K_{ao}) + b\,((K_v\tau) - (K_v\tau)_o)$

and ϵ is some tolerance allowed on the Drift Minimum Condition (ϵ is input by the user).

At the beginning of any COEBRA iteration, there are 3 possible conditions on K_o.

Condition 1:

The drift minimum criterion is already met and

$$-\epsilon \leq K_o \leq +\epsilon$$

Condition 2:

The drift minimum criterion is not met and

$$-\epsilon \leq K_o > +\epsilon$$

Condition 3:

The drift minimum criterion is not met and

$$-\epsilon > K_o \leq +\epsilon$$

For each condition on K_o, COEBRA will put 2 constraint equations on Drift Minimum.

Condition 1: If $-\epsilon \leq K_o \leq +\epsilon$

the 2 drift minimum constraint equations will be:

Eq. 1 $\quad K_o + \Delta K_o \leq +\epsilon$

$$\text{or} \quad aK_D + bK_a + b(K_v\tau) \leq +\epsilon - C$$

Eq. 2 $\quad K_o + \Delta K_o \geq -\epsilon$

$$\text{or} \quad aK_D + bK_a + b(K_v\tau) \geq -\epsilon - C$$

Condition 2: If $-\epsilon \leq K_o > +\epsilon$

the 2 drift minimum constraint equations will be:

Eq. 1 $K_o + \Delta K_o \leq K_o + STEP * (\epsilon - K_o)$

or $aK_D + bK_a + b(K_v\tau) \leq K_o + STEP * (\epsilon - K_o) - C$

Eq. 2 $K_o + \Delta K_o \geq -\epsilon$

or $aK_D + bK_a + b(K_v\tau) \geq -\epsilon - C$

NOTE: STEP is a COBIN namelist variable that is defined in Chapter 2 and again in Chapter 10. STEP specifies the percent "improvement" required in the "not-met" constraint equations on each COEBRA iteration. Recall that STEP is adjusted in the so-called Inner Loop.

Condition 3: If $-\epsilon > K_o \leq \epsilon$

the 2 drift minimum constraint equations will be:

Eq. 1 $K_o + \Delta K_o \leq +\epsilon$

or $aK_D + bK_a + b(K_v\tau) \leq +\epsilon - C$

Eq. 2 $K_o + \Delta K_o \geq K_o + STEP * (-\epsilon - K_o)$

or $aK_D + bK_a + b(K_v\tau) \geq K_o + STEP * (-\epsilon - K_o) - C$

Section 8.2.2

This subsection defines how COEBRA calculates the airframe / actuator coefficients in the Drift Minimum Constraint equations. These coefficients, denoted in Section 8.2.1 as a, b, and c, are calculated from the same COBDE namelist data that COEBRA uses to calculate $\beta_P, \zeta_{\psi P}$, and $\zeta_{\phi P}$ for the Load Relief Cost Function. In keeping with the philosophy used for the Load Relief Cost Function, all of the airframe and actuator parameters in the COBDE namelist that are used to calculate a, b, and c, are input with positive values for an aerodynamically unstable vehicle. The reason for defining the sign convention in this way is to avoid confusion between pitch and yaw. When running pitch, the user must use yaw plane notation and the indicated sign convention (i.e., all data positive for unstable vehicle). If the vehicle is stable (i.e., center of pressure aft of the center of gravity), the only parameter that should change is N_β (yawing aero moment due to β), and it should then be input as a negative number.

Using yaw plane notation and the sign convention that all data is positive for an unstable vehicle:

$$a = (K_{TVC}/B)\ (N_\beta Y_{\delta r} + Y_\beta N_{\delta r})$$

$$b = -(K_{TVC}/B)\ (N_\beta Y_{\delta r} + Y_\beta N_{\delta r})\ (g*\sin\theta_L)$$

$$c = (1/B)\ (N_\beta)\ (g*\sin\theta_L)$$

Table 8.1 lists the COBDE namelist variables that are required to calculate a,b&c. Chapter 10 which summarizes all of the variables that are used in the COBDE namelist, contains the detailed definitions of each

TABLE 8.1: COBDE NAMELIST VARIABLES THAT ARE REQUIRED FOR THE DRIFT MINIMUM CONSTRAINTS

COBDE NAMELIST NAME	YAW PLANE SYMBOL	PITCH PLANE EQUIVALENT
CNB	$C_{n\beta}$	$C_{m\alpha}$
CYB	$C_{y\beta}$	$C_{z\alpha}$
CNDR[1.]	$C_{n\delta r}$	$C_{m\delta e}$
CYDR[1.]	$C_{y\delta r}$	$C_{z\delta e}$
QBAR[2.]	$\bar{q}$	$\bar{q}$
RHO[2.]	ρ	ρ
SF	S	S
D	b	c
TY[1.]	T_y	T_p
LGY[1.]	L_{gy}	L_{gp}
IZZ	I_{zz}	I_{yy}
MASS	M	M
G	g	g
THETA	θ_L	θ_L
B	V_{rw}	V^2_{rw}/U_r
KTVC	K_{TVC}	K_{TVC}

1. COEBRA assumes a single control torque source, either TVC or an aero surface. Hence, CNDR and CYDR must be zero if TVC is used, and TY and LGY must be zero if aero surfaces are used.

2. If QBAR is input as zero, COEBRA calculates QBAR from (.5*RHO*B*B).

of these parameters. Table 8.1 merely lists these parameters and shows the yaw plane symbol usually used for each as well as the pitch plane equivalent. Table 8.2 illustrates how COEBRA combines these variables to calculate a, b, and c.

TABLE 8 .2: CALCULATING THE DRIFT MINIMUM COEFFICIENTS

	COEFFICIENT[1]	COEBRA NOTATION IN COBDE NAMELIST	PITCH PLANE EQUIVALENT
N_{β}[2]	$\frac{\bar{q} S b C_{n\beta}}{I_{ZZ}}$	QBAR*SF*D*CNB / IZZ	$M_{\alpha} = \frac{\bar{q} S c C_{m\alpha}}{I_{YY}}$
Y_{β}	$\frac{\bar{q} S C_{y\beta}}{M}$	QBAR*SF*CYB / MASS	$Z_{\alpha} = \frac{\bar{q} S C_{z\alpha}}{M}$
N_{δ_r}[2] from an aero surface	$\frac{\bar{q} S b C_{n\delta r}}{I_{ZZ}}$	QBAR*SF*D*CNDR / IZZ	$M_{\delta e} = \frac{\bar{q} S c C_{m\delta e}}{I_{YY}}$
N_{δ_r} from TVC	$\frac{T_y L_{gy}}{I_{ZZ}}$	TY*LGY / IZZ	$M_{\delta e} = \frac{T_p L_{gp}}{I_{YY}}$
Y_{δ_r} from an aero surface	$\frac{\bar{q} S C_{y\delta r}}{M}$	QBAR*SF*CYDR / MASS	$Z_{\delta e} = \frac{\bar{q} S C_{z\delta e}}{M}$
$Y_{\delta r}$ from TVC	$\frac{T_y}{M}$	TY / MASS	$Z_{\delta e} = \frac{T_p}{M}$

1. By definition, all of these parameters are positive for an aerodynamically unstable vehicle.

2. Note that N_{β} and N_{δ_r} are unprimed aero stability coefficients.

Section 8.2.3

The purpose of this subsection is to discuss the parameter called ϵ . This parameter is input by the user, and is a tolerance allowed on the drift minimum condition. In order to define the units on ϵ , Equation 8.4 is repeated:

$$\frac{K_o}{\tau} \doteq \omega^2 \omega_1 \omega_2 \qquad \text{(Equation 8.4)}$$

Now, ω^2 is usually around unity in magnitude, and $\omega_1 \doteq (1/\tau)$ where τ is the time constant of the psuedo-integrator in the $(K_v\tau)$ feedback loop. Hence, Equation 8.4 becomes

$$\frac{K_o}{\tau} \doteq \frac{\omega_2}{\tau}$$

and K_o is seen to be approximately equal to the frequency of the real root that is right at the origin for "perfect" drift minimum. Since it is desired that

$$-\epsilon \leq K_o \leq +\epsilon$$

then ϵ is a tolerance on the frequency of the "drift minimum" root. For example, if $\epsilon =$ .1, this requires that the root be within $\pm$ .1 rad/sec of the origin. This allowed tolerance (ϵ) is input by the variable called "DRFTOL" in the COBIN namelist. "DRFTOL" is applicable to all the time points.

Section 8.2.4

The purpose of this subsection is simply to state that COEBRA identifies the autopilot parameters (K_D,K_a, and the gain $K_v \tau$) in the drift minimum constraint equations using the same method that is required for "load relief optimization". This method is explained in detail in Chapter 7, and briefly repeated here.

1. The COBDE namelist name corresponding to K_D is LKD.

   LKD = (location in Hollerith of K_D in the QD030 block diagram). For example, if K_D is the gain in Block J, then LKD = 5HRJ(7).

2. K_a is identified by LKA1 which is set equal to the location of K_a in the QD030 block diagram.

3. ($K_v \tau$) is similarly identified via LKA2.

Further, K_D, K_a and ($K_v \tau$) must individually be declared either:

1. A constant;
2. A "stand-alone" variable;
3. A "common" variable; or
4. They may individually have a value of zero, in which case, they will not be declared, and will not even appear in the QD030 Block Diagram. As an example, K_a may be a stand-alone variable, K_D may be a constant, and ($K_v \tau$) may not even be used.

Finally, note that K_D, K_a and ($K_v \tau$) all have positive values when used in the Drift Minimum Constraint Equations.

Section 8.2.5

Additional Notes:

Note 1. The user will select those time points at which the Drift Minimum Constraints are to be applied. He will do this via the COBDE namelist variable called "MINDRF".

If MINDRF = 0, the drift minimum constraints are not to be used at this time point.

If MINDRF = 1, the drift minimum constraints are to be used at this time point.

Note 2. "Drift Minimum" and "Load Relief Optimization" are not linked together by COEBRA. In other words, the user may ask for the drift minimum constraints at any or all of the time points, completely independent of whether LOADOP = 0 (no load relief optimization) or LOADOP = 1 (load relief optimization).

Note 3. The Drift Minimum Constraints can be used whether or not the airframe includes structural bending and/or fuel slosh modes.

Note 4. The Drift Minimum option allows COEBRA to design a "drift minimum" autopilot that meets the stability margin requirements, and if desired, at the same time has the maximum amount of load relief capability. Note that a "high frequency" drift minimum autopilot yields the "least" residual drift rate and hence least trajectory drift. A "high load relief" drift minimum autopilot results in larger drift rates and hence more drift.

Chapter 9. The Vector Constraints

9.1 Introduction

The purpose of the so-called Vector Constraints is to keep the individual autopilot vectors (e.g., the attitude error vector, the rate vector, the accelerometer loop vector, etc.) from getting very much larger than the total resultant vector at all frequencies. When the individual vectors do get much larger than the resultant, this means that vector cancellation exists, and this can lead to problems when tolerances on the airframe/autopilot parameters are considered.

Figure 9.1 illustrates the situation that might exist at the resonant frequency or frequency at the peak of the 1st structural bending mode. The figure shows that the resultant (peak of the 1st structural bending mode) is comprised of the following vectors or feedback loops: (1) attitude error; (2) rate; and (3) accelerometer. Referring to Figure 6.1 and Table 6.1 on the QD030 block diagram, suppose that the attitude error feedback loop was defined by the Case 7 open-loop transfer function, the rate feedback loop by the Case 8 transfer function, and the accelerometer feedback loop by the Case 9 transfer function. Then, Case 7 + Case 8 + Case 9 = Case 13 (Resultant). Now, it is desired to constrain the magnitude of each vector at the frequency of the 1st structural bending mode peak. For example, it is desired that:

$$\left| \text{Case 7} \right| \leqslant C* \quad \left| \text{Case 13} \right|$$

$$\left| \text{Case 8} \right| \leqslant C* \quad \left| \text{Case 13} \right|$$

$$\left| \text{Case 9} \right| \leqslant C* \quad \left| \text{Case 13} \right|$$

where C is an arbitrary constant.

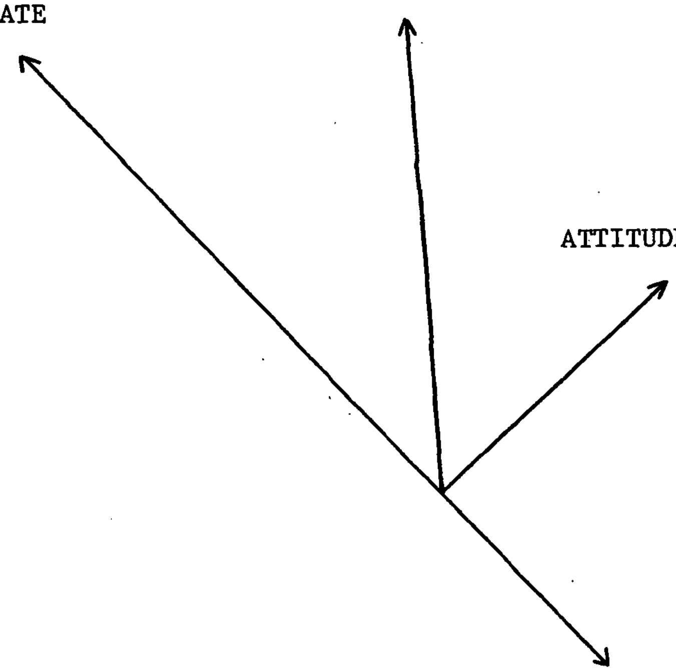


Figure 9.1 ILLUSTRATING THE VECTOR CONSTRAINTS

9.2 The Vector Constraint Equations

At each frequency, the resultant is comprised of K vectors, V(k). In turn, each vector is comprised of L autopilot variables, $X(\ell)$. Expanding each V(k) in a first order Taylor Series about $V(k)_o$ (nominal value of V(k) evaluated at the i^{th} frequency), it is desired that,

$$\left|V(k)_o\right| + \sum_{\ell=1}^{L} \left[\frac{\partial |V(k)|}{\partial X(\ell)}\right]_o * \Delta X(\ell) \leqslant V(k)_s$$

(for k = 1, ..., K)

If the requirement on V(k) is already satisfied, then

$$V(k)_s = SPEC\ (k)$$

If not, then

$$V(k)_s = \left|V(k)_o\right| + STEP* \left[SPEC(k) - \left|V(k)_o\right| \right]$$

Where, at the user's option:

(1) SPEC (k) = PERVEC(k)* | Case 13 |

or (2) SPEC(k) = AMPVEC(k)

where, PERVEC and AMPVEC are input parameters.

9.3 Vector Constraint Input Parameters

This section defines the four input parameters that deal with the so-called vector constraints. COEBRA allows for a total of 18 vector constraints per time point. This is because the user can request that vector constraints be applied to a maximum of 3 elements of the Margin Array, and because each element of the Margin Array can be comprised of a maximum of 6 vectors or feedback loops.

The following 4 input parameters are input via the COBDE namelist.

1) MARVEC(k), Integer array (3 dimensional), preset value = 3*0

This array specifies which elements of the Margin Array are to have the so-called vector constraints. A maximum of three margins per time point can be constrained in this way, and any of the first 47 elements of the Margin Array is a valid choice (except, of course, the structural mode peak phases). For example, if MARVEC = 4, 14, 28, then vector constraints are to be applied to the peaks of the 1st, 2nd and 3rd structural bending modes.

2) KVEC(k), Integer array (6 dimensional), preset value = 6*0

This array specifies which of the six QD030 feedback loops are to be constrained for the margins flagged by MARVEC. These six QD030 feedback loops are identified by the NCASE option (of the INDATA namelist) for NCASE = 7 through 12. For example, if all the QD030 feedback loops are to be constrained, then KVEC = 7, 8, 9, 10, 11, 12.

3) AMPVEC(k), Floating point array (3 dimensional), preset value = 3*0

This is "vector constraint" option #1. At the frequency of each margin specified by MARVEC, AMPVEC(k) is the maximum absolute real value that any QD030 feedback vector (specified by KVEC) may have. There is one value of AMPVEC for each margin called out in MARVEC. For example, if AMPVEC(2) = .5, then each of the QD030 feedback vectors that are called out by KVEC, will be constrained to an amplitude of less than or equal to 0.5, when evaluated at the frequency of the 2nd element of the MARVEC array.

4) PERVEC(k), Floating point array (3 dimensional), preset value = 3*0

This is "vector constraint" option #2. At the frequency of each margin specified by MARVEC, none of the magnitudes of the QD030 feedback vectors (specified by KVEC) may exceed the product:

PERVEC(k)*(magnitude of the k^{th} element of MARVEC)

This option allows the user to constrain each feedback vector to have a magnitude less than or equal to some percentage of the magnitude of the total or resultant vector. For example, suppose MARVEC(3) = 14, and suppose that element #14 (peak of the 2nd structural bending mode) has a value of 0.4. If PERVEC(3) = 1.5, then, at the frequency of Margin array element #14, the magnitude of each QD030 feedback vector that is called out by KVEC, will be constrained to be less than or equal to (1.5)*(0.4) or 0.6.

Chapter 10. COEBRA Input Data Summary

This chapter summarizes and defines the data that is input to COEBRA via the COBDE and COBIN namelists. The COBDE namelist is required for each time point or vehicle state, and the COBIN namelist is input once and is applicable to all time points.

Note that the data in these namelists may appear anywhere in columns 2 through 80. Also, the preset values indicated for these variables are to be overridden by the user if he so desires.

Part A of this chapter defines the COBDE namelist variables. For convenience, the following is a list of these variables, along with the aspect of COEBRA each is concerned with.

(1) Section 1, The Margin Array

- 1.1 IOMEGA
- 1.2 IDMODE
- 1.3 KAPCH
- 1.4 MARVEC
- 1.5 KVEC
- 1.6 AMPVEC
- 1.7 PERVEC

(2) Section 2, The Cost Functions

- 2.1 WTFAC

(3) Section 3, The Autopilot Parameters

- 3.1 NTOTAL
- 3.2 NTYPE

(4) Section 4, The Figures-of-merit

- WTMARG

(5) Section 5, Load Relief Optimization and Drift Minimum Constraints

5.1 The airframe/actuator data that is needed for uncoupled yaw.

CNB	LGY
CYB	IZZ
CNDR	MASS
CYDR	G
QBAR	THETA
RHO	B
SF	KTVC
D	LAY
TY	U

The additional airframe/autopilot data needed for yaw/roll coupling.

CLB	IXX
CLDR	IXZ
CLDA	PHI
CNDA	W
TR	KDR
LGR	KRR
ZOFF	

5.2 Autopilot Parameters

5.2.1	LKD	LKDA	LKA1	LKA2
	LKR1	LT1	LA1	LA4
	LKR2	LT2	LA2	LA5
		LT3	LA3	LA6
				LA7
				LA8

5.2.2 TS

(5) Section 5 - (Continued)

5.3 Wind and Time Search Parameters

U1	TW3
U2	TSTART
U3	TSTOP
U4	DELT
TW1	DINCRE
TW2	

5.4 Miscellaneous Parameters

MINDRF

LRCOST

ROOTOL

<u>Part A, Section 1</u> - The seven COBDE namelist variables that are used when forming the so-called Margin Array (Chapter 1) are discussed in this section.

<u>1.1 IOMEGA</u>, Integer, preset value = 0

IOMEGA = 0, if the rigid-body phase margin frequency requirement <u>is not</u> to be used.

IOMEGA = 1, if the requirement <u>is</u> to be used.

<u>1.2 IDMODE (k)</u>, Integer array (8 dimensional), preset value = 1, 2, 3, 4 5, 6, 7, 8, corresponding to i = 1 to 8

Referring to the underdamped quadratic roots in the denominator of the total open loop transfer function (CASE 13), any root-pair whose damping ratio is less than DAMP (COBIN namelist variable) is considered to be from a structural bending or fuel slosh mode. COEBRA lists these bending and füel slosh modes according to frequency, beginning with the lowest frequency. The IDMODE (k) array is then used to identify these modes according to this ordered list, beginning with the lowest frequency.

IDMODE (k) = 1, if the k^{th} mode is a first structural mode.

IDMODE (k) = 2, if the k^{th} mode is a second structural mode.

etc., until

IDMODE (k) = 8, if the k^{th} mode is an eighth structural mode.

IDMODE (k) = 10 or higher, if the k^{th} mode is a fuel slosh mode.

IDMODE (k) = 0, if the k^{th} mode is to be ignored.

NOTE: COEBRA may be run with or without slosh and/or bending modes. When running with modes, the modes need not be ordered according to frequency; e.g., the 1st mode may be higher in frequency than the 3rd mode, or the 4th mode may be lower in frequency than the slosh modes, etc. The exception to this is the

NOTE: (Continued)

2nd mode, which must be run with something identified as a 1st mode that is lower in frequency and something identified as a higher mode that is higher in frequency.

1.3 KAPCH, Integer, preset value = 0

This option is applicable to all first and second structural modes. When KAPCH = 0 and the modal peak gain is greater than zero db, the closest-approach margin of the mode will not be constrained. The mode will be constrained by its peak gain, peak phase, and front and backside phase margins.

When KAPCH = 1 and the modal peak gain is greater than zero db, the closest-approach margin of the mode will be constrained, along with its peak gain, peak phase, and front and backside phase margins. When the mode peaks below zero db, the closest-approach margin will always be constrained. For this case, the mode peak will not be constrained, but the peak phase can still enter the cost function at the user's option.

1.4 MARVEC (k), Integer array (3 dimensional), preset value = 3*0

This array specifies which elements of the Margin Array are to have the so-called vector constraints (Chapter 9). A maximum of three margins can be constrained in this way, and any of the first 47 elements of the Margin Array is a valid choice (except, of course, the structural mode peak phases). For example, if MARVEC = 4, 14, 28, then vector constraints are to be applied to the peaks of the 1st, 2nd and 3rd structural bending modes.

1.5 KVEC (k), Integer array (6 dimensional), preset value = 6*0

This array is associated with the vector constraints (Chapter 9). It specifies which of the six QD030 feedback loops are to be constrained for the margins flagged by MARVEC (k). These six QD030 feedback loops are identified by the NCASE option (of the INDATA namelist) for NCASE = 7 through 12. If all the QD030 feedback loops contain autopilot variables, and are to be constrained, the KVEC = 7, 8, 9, 10, 11, 12.

1.6 AMPVEC (k), Floating point array (3 dimensional), preset value = 3*0

This is "vector constraint" option #1. At the frequency of each margin specified by MARVEC, AMPVEC (k) is the maximum absolute real value that any QD030 feedback vector (specified by KVEC) may have. There is one value of AMPVEC for each margin called out in MARVEC. For example, if AMPVEC (2) = .5, then each of the QD030 feedback vectors that are called out by KVEC, will be constrained to an amplitude of less than or equal to 0.5, when evaluated at the frequency of the 2nd element of the MARVEC array.

1.7 PERVEC (k), Floating point array (3 dimensional), preset value = 3*0

This is "vector constraint" option #2. At the frequency of each margin specified by MARVEC, none of the magnitudes of the QD030 feedback vectors (specified by KVEC) may exceed the product:

PERVEC (k) * (magnitude of the k^{th} element of MARVEC)

This option allows the user to constrain each feedback vector to have a magnitude less than or equal to some percentage of the magnitude of the total or resultant vector. For example, suppose MARVEC (3) = 14, and suppose that Margin Array element #14 (peak of the 2nd structural bending mode) has a value of 0.4. If PERVEC (3) = 1.5, then, at the frequency of Margin Array element #14, the magnitude of each QD030 feedback vector that is called out by KVEC, will be constrained to be less than or equal to (1.5) * (0.4) or 0.6.

Part A, Section 2 - This section is used to define the COBDE namelist variable called WTFAC which is an arbitrary weighting factor in both cost functions: (1) "Maximize margins" in Chapter 3; and (2) "Maximize Load Relief" in Chapter 4.

2.1 WTFAC (i), Floating point array (51 dimensional). Since the index i corresponds to the elements of the Margin Array Table (Chapter 1), the following table lists the preset values for WTFAC (i).

Preset values for i=1 to 51	Corresponding Margin Array Elements
3 * 1,	rigid body margins
1, 0, 3 * 0, 1, 0, 3 * 0,	both first modes
1, 0, 5 * 0, 1, 0, 5 * 0,	both second modes
1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,	3rd to 8th mode
1, 1, 1, 1, 1, 1, 1, 1,	eighth fuel slosh modes
0,	phase margin crossover frequency
1, 1, 1	$\beta_p, \delta_{\psi p}, \delta_{\phi p}$

This is an array of arbitrary cost function weighting factors that is input by the user. This array is used for both types of cost functions. For the "maximize margins" cost function, WTFAC (i) serves as an additional multiplicative weighting factor to the one that is provided by the WTYPE option. The WTYPE option either provides a "unity" or "desired over nominal" type weighting factor. WTFAC (i) allows the user to emphasize or de-emphasize certain elements of the cost function relative to others. This array also serves another function. Since there is a standard preset cost function, WTFAC (i) allows the user to add or delete elements from this preset cost function. In other words, WTFAC (i) also allows the user to "build his own" cost function, if the "built-in" cost function will not solve his problem.

Part A, Section 3 - This section defines the two COBDE namelist variables that are used to define the "parameter-type" that is to be associated with each autopilot parameter.

3.1 NTOTAL, Integer, Preset value - none, problem dependent.

NTOTAL is the total number of parameters (Bode gains, time constants, damping ratios and natural frequencies) that appear in the QD030 blocks A through N for this time point. NOTE: Free S's are not considered as parameters, though they may be used in any of the QD030 blocks.

3.2 NTYPE (j), Integer array (50 dimensional), preset values - none, problem dependent.

This is a compressed array that corresponds to the parameters in the order in which they are input in the QD030 blocks A through N, and identifies their "type." By order is meant, by ordered QD030 block and by ordered element number within each block.

(a) NTYPE (j) = 0 if the j^{th} parameter is a constant whose value is not to be altered by COEBRA.

(b) NTYPE (j) = -1 if the j^{th} parameter is a stand-alone variable whose value, when altered by COEBRA, need not be the same as the parameter that appears in the same QD030 block and element location at the immediately preceeding time point.

(c) NTYPE (j) = 1 if the j^{th} parameter is a common variable whose value, when altered by COEBRA, must be the same as the variable that appears in the same QD030 block and element location at the preceding timepoint.

The following three notes are very important.

(1) For each time point, before reading in any data, COEBRA initializes the QD030 block diagram as follows:

(a) QD030 blocks A through H are initialized with a gain of unity and no roots including free S's;

(b) QD030 blocks I through N are completely zeroed out. After reading the autopilot parameters into the QD030 block diagram (via the INDATA namelist), COEBRA checks to see how blocks A through N have been changed from their initial values. NTOTAL and NTYPE (j) must only include those parameters in those QD030 blocks A through N.that have been changed from their initial values. If a variable gain whose initial value is unity, must be used in any of the blocks A through H, this this variable gain must be initialized with a value of one-plus-epsilon.

(2) NTOTAL and NTYPE (j) must identify even those constants that lie outside the allowed field as defined in Chapter 7 on the Allowed Autopilot Configuration.

(3) For the first time point entered, only 0's and -1's are to be used in the NTYPE (j) array.

Part A, Section 4

This section is used to define the COBDE namelist parameter WTMARG (i). This parameter is a weighting factor in both of the figures-of-merit (Chapter 5). It is a 51 dimensional integer array. The index i corresponds to the elements of the Margin Array (Chapter 1), and if WTMARG (i) $\neq 0$, the i^{th} element of the Margin Array will be included in the figure-of-merit.

For the "maximize margins" figure of merit, only the following elements of the Margin Array can <u>ever</u> be included. All of the other elements are not applicable.

1. The three rigid body margins. (i = 1, 2, and 3).
2. If the 1^{st} and/or 2^{nd} modes are stable, only their "closest approach" margins can be included. (i = 8, 13, 20, and 27).
3. If the 1^{st} and/or 2^{nd} modes are unstable, only their "unstable" front or backside phase margins can be included. (i = 6 or 7, 11 or 12, 16 or 17, and 23 or 24).
4. For the 3^{rd} to the 8^{th} mode, only their peak gains can be included. (i = 28, 30, 32, 34, 36 and 38).
5. For all 8 slosh modes, their backside phase margins can be included. (i = 40 through 47).
6. The rigid body phase margin frequency (i=48).

WTMARG (48) is preset to zero. The rest of the above elements are preset to unity to be overriden by the user.

For the "optimize load relief" figure-of-merit, β_ρ , $\delta_{\psi\rho}$ and $\delta_{\phi\rho}$ can all be included. (i = 49, 50 and 51). WTMARG (i) for i = 49, 50 and 51 is preset to unity to be overriden by the user.

Part A, Section 5 - This section defines those COBDE namelist parameters that are input only when load relief optimization and/or the drift minimum constraint equations are used. When optimizing load relief and/or constraining drift minimum on a multiple time point run, the user must not enter any of the data of this section, at those time points at which he does not want to consider load relief optimization and/or the drift minimum constraints.

5.1 Airframe, actuator and roll autopilot data

Table 10.1 defines the airframe and actuator data needed for uncoupled load-relief optimization and for the drift minimum constraint equations. Table 10.2 defines the additional airframe and roll autopilot data that is needed for yaw/roll coupled load relief optimization. The following notes are applicable to the data in Tables 10.1 and 10.2.

Note #1: All the data is input in floating point, and all the data is preset to zero.

Note #2: Yaw and roll plane notation is used. A pitch plane design is performed using the indicated yaw plane notation and sign convention, and by not inputting roll data. Obviously, uncoupled yaw is designed by simply not inputting the roll data. Since this same data is to be used for the pitch plane as well as for the yaw/roll plane, the following sign convention is used in order to avoid confusion between pitch and yaw.

The usual sign convention for forces and moments is:

(1) a positive X-axis force points "out the nose";

(2) a positive sideforce is "out the right wing";

(3) a positive normal force is "down";

(4) a positive pitching moment is "nose up";

(5) a positive yawing moment is "nose right"; and

(6) a positive rolling moment is clockwise looking forward.

Table 10.1 DATA FOR DRIFT MINIMUM AND UNCOUPLED LOAD RELIEF

Name	Definition	Pitch Plane Equivalent
CNB	$C_{n\beta}$ - aero moment due to angle of sideslip	$C_{m\alpha}$
CYB	$C_{y\beta}$ - aero sideforce due to angle of sideslip	$C_{z\alpha}$
CNDR	$C_{n\delta r}$ - aero moment due to rudder deflection	$C_{m\delta e}$
CYDR	$C_{y\delta r}$ - aero sideforce due to rudder deflection	$C_{z\delta e}$
QBAR	$\bar{q}$ - dynamic pressure	--
RHO	ρ - air density. Note: if QBAR is zero, COEBRA calculates QBAR from (.5 * RHO * B * B)	--
SF	S_F - aero reference area	--
D	b - aero reference length	C
TY	Controlled thrust	--
LGY	Controlled thrust moment arm (positive number)	--
IZZ	I_{zz} - body axis moment of inertia about the center of gravity	I_{yy}
MASS	m - total vehicle mass	--
G	g - gravitational acceleration	--
THETA	θ_L - vehicle pitch attitude relative to local horizontal (degrees)	--
B	For yaw, $B=V_{rw}$ = total velocity relative to the wind For pitch, $B=V_{rw} * V_{rw}/U_r$ where U_r is body X-axis relative velocity.	--
KTVC	Effective actuator/control-device gain	--
LAY	Lateral body mounted accelerometer moment arm (positive if forward of center of gravity). LAY is not used for Drift Minimum.	--
U	Velocity along the body X-axis. U is not used for Drift Minimum	--

Table 10.2 DATA FOR YAW/ROLL COUPLED LOAD RELIEF

NAME	DEFINITION
CLB	$C_{\ell\beta}$ - Aero roll moment due to angle of sideslip
CLDR	$C_{\ell\delta r}$ - Aero roll moment due to rudder deflection
CLDA	$C_{\ell\delta a}$ - Aero roll moment due to aileron deflection
CNDA	$C_{n\delta a}$ - Aero yaw moment due to aileron deflection
TR	Controlled thrust in roll
LGR	Controlled thrust moment arm in roll (positive number)
ZOFF	Moment arm through which yaw controlled thrust produces a rolling moment. Positive, if positive yaw deflection produces counterclockwise roll.
IXX	Body axis roll moment of inertia about the center of gravity
IXZ	Body axis yaw-roll crossproduct of inertia about the center of gravity.
PHI	ϕ_o - Euler roll angle used to orient the gravity vector with respect to the vehicle's yaw plane (degrees).
W	Velocity in the body Z-axis direction
KDR	Attitude error gain used in the roll autopilot equation.
KRR	Attitude rate gain used in the roll autopilot equation.

With this in mind, all COEBRA data is input as positive values for an aerodynamically unstable vehicle (negative $C_{n\beta}$ or positive $C_{m\alpha}$) with a positive dihedral effect (negative $C_{\ell\beta}$), and with "normal" control torque sources. By "normal" control torque sources is meant: (1) A positive yaw deflection produces a negative sideforce, a positive yawing moment and a negative rolling moment; (2) A positive pitch deflection produces a positive normal force, a positive pitching moment, and zero rolling moment; and (3) A positive roll deflection produces a negligible sideforce, a negative yawing moment, and a positive rolling moment. If any of these conditions are not satisfied, the user must compensate by making the appropriate changes in the signs of the input data. For example, if the vehicle is aerodynamically stable (positive $C_{n\beta}$ or negative $C_{m\alpha}$), the corresponding COBDE namelist parameter in Table 10.1, i.e., CNB, is input with a negative sign.

Note #3: In optimizing load relief, a single control torque source is assumed for yaw (or pitch), and a single control torque source is assumed for roll. By appropriate input data, these sources can be either:

(1) Thrust vector control via gimballed engines, secondary injection, or jet vanes; or

(2) Control via aerodynamic surfaces.

If aero control surface data is not input, thrust vector control is assumed.

Note #4: All aerodynamic stability derivatives are input in the body axis with units of per radian. All aero moment coefficients are assumed about the center of gravity.

Note #5: The angles THETA and PHI must be input in degrees. All other units are arbitrary, but must be consistent.

5.2 Autopilot Parameters

This section defines the COBDE namelist variables that are associated with the autopilot parameters when optimizing load relief and/or constraining drift minimum.

5.2.1 Identifying Each Parameter

As discussed in Chapter 7, the user must tell COEBRA how to equate the autopilot parameters in the QD030 Block Diagram with the parameters that are used in the load relief control law (Equation 7.1 in Chapter 7). Table 10.3 lists the Namelist names that are used to identify the locations in the QD030 Block Diagram of the load relief control law parameters of Equation 7.1. Each namelist name is equated via Hollerith form, to the location of the respective parameter. Obviously, the parameters of Table 10.3 are preset with blank locations.

Table 10.3 LOCATIONS OF THE PARAMETERS OF EQUATION 7.1

Namelist Name	Corresponding QD030 location, in Hollerith, of . . .
LKD	K_D
LKR1	K_{R1}
LKR2	K_{R2}
LKDA	K_{DA}
LT1	τ_1
LT2	τ_2
LT3	τ_3
LKA1	K_{A1}
LA1	A_1
LA2	A_2
LA3	A_3
LKA2	K_{A2}
LA4	A_4
LA5	A_5
LA6	A_6
LA7	A_7
LA8	A_8

5.2.2 TS, Floating point, preset value = 0

TS is used when designing a digital load relief autopilot, and is the sampling period in seconds. Since the user inputs the initial values of the digital autopilot into the QD030 block diagram in W-plane form (because the frequency response is calculated in the W-plane), and since β_p , $\delta_{\psi p}$ and $\delta_{\phi p}$ are calculated via S-plane equations, TS is required to transform the filters in Equation 7.1 back and forth between the S and W-planes. TS is only required for the following filters: LT1, LT2, LT3, LA1, LA2, LA3, LA4, LA5, LA6, LA7, and LA8. Since these are always very low break frequency filters, the following approximate formula is sufficient.

(S-plane time constant) = 0.5 * TS * (W-plane time constant)

If TS $\neq$ 0, COEBRA assumes a digital autopilot, and will use the above equation.

5.3 Wind Parameters

This section defines the COBDE namelist variables that are used to define the wind profile when optimizing load relief capability. The wind profile is completely defined in Chapter 4, but Table 10.4 summarizes the parameters that determine the shape of the wind profile. Table 10.5 summarizes the parameters that are used when searching the time response for β_p, $\delta_{\psi p}$ and $\delta_{\phi p}$.

TABLE 10.4 SHAPE OF THE WIND PROFILE

Name	Definition	Preset Floating Point Value
U1	Slope of β_ω from 0 to TW1 sec. (rad/sec)	.0021
U2	Slope of β_ω from TW1 to TW2 sec. (rad/sec)	.0100
U3	Slope of β_ω from TW2 to TW3 sec. (rad/sec)	.0228
U4	Slope of β_ω from TW3 to TSTOP sec. (rad/sec)	-.0038
TW1	Wind slope break time #1 (sec.)	40.0
TW2	Wind slope break time #2 (sec.)	52.5
TW3	Wind slope break time #3 (sec.)	55.0

TABLE 10.5 DEFINING THE TIME RESPONSE SEARCH

Name	Definition	Preset Value
TSTART	Starting time for calculating and plotting the nominal time response (sec.)	30. (floating point)
TSTOP	Stop time for calculating and plotting the nominal time response (sec.)	70. (floating point)
DELT	Time increment used in calculating the time response (sec.). Because of storage limitations, the following relationship must be satisfied: (TSTART - TSTOP) / (DELT) $\leq$ 2000.	.05 (floating point)
DINCRE	When determining the "disturbed" peak values for the partial derivatives, COEBRA computes the response for only $\pm$ DINCRE time increments on either side of the time at which <u>each</u> (β_ρ $\delta_{\Psi\rho}$ or $\delta_{\Phi\rho}$) nominal peak occurred. Note that β_ρ $\delta_{\Psi\rho}$ and $\delta_{\Phi\rho}$ need not occur at the same time.	10 (Integer)

5.4 Miscellaneous

Table 10.6 contains the definitions of 3 miscellaneous COBDE namelist variables that are used when optimizing load relief and/or constraining drift minimum.

TABLE 10.6 MISCELLANEOUS PARAMETERS

Name	Definition	Preset Value
MINDRF	= 1, the drift minimum constraints will be used. = 0, they will not be used	0 (Integer)
LRCOST (k)	(a) if LRCOST (1) $\neq$ 0, β_ρ will be computed (b) if LRCOST (2) $\neq$ 0, $\delta_{\psi\rho}$ will be computed (c) if LRCOST (3) $\neq$ 0, $\delta_{\phi\rho}$ will be computed Via the WTFAC option, any combination of these peaks can be included in the load relief cost function. For example, even though LRCOST = 3 * 1, if WTFAC (49) and WTFAC (50) are unity and WTFAC (51) is zero, then only β_ρ and $\delta_{\psi\rho}$ will enter the load relief cost function.	3 * 0 (3 dimensional integer array)
ROOTOL	Tolerance used for root cancellation in the three "load relief" transfer functions. If a real root in the numerator and denominator have a frequency such that $\lvert W\ (num) - W(den) \rvert /$ W(den) < ROOTOL, the roots will automatically be cancelled. Also, for complex roots, the magnitude of the roots replaces the frequency in the above formula.	.0001 (Floating point)

Part B of this chapter defines the COBIN namelist variables. The following is a list of these variables.

(1) Section 1, the Margin Array:

1.1 DAMP

1.2 AMPOUT

1.3 SCALF

1.4 MAROPT

1.5 NOTERM

(2) Section 2, the Cost Functions:

2.1 COSPEC

2.2 WTYPE

2.3 LOADOP

(3) Section 3, the Autopilot Variable Constraints:

3.1 XMIN

3.2 XMAX

3.3 P

3.4 PSMALL

(4) Section 4, the Figures-of-Merit and the Margin Counter:

4.1 TOLMAC

4.2 TOLFGM

(5) Section 5, the Stability Margin and Drift Minimum Constraints:

5.1	SPEC
5.2	STEP
5.3	DSTEP
5.4	DRFTOL

(6) Section 6, Miscellaneous:

6.1	NITER
6.2	MFRESP
6.3	DELPAR
6.4	PRINT
6.5	NPRINT
6.6	KROOT

Part B, Section 1

This section defines the COBIN namelist variables that are used to form the Margin Array.

1.1 DAMP, Floating point, preset value = 0.05

Any underdamped quadratic roots in the denominator of the total open loop transfer function (CASE 13) that have an effective damping ratio with an absolute value less than DAMP, are considered to be bending or fuel slosh modes. The IDMODE array is then used to identify the modal type (e.g., 1st mode, etc.) to be associated with each quadratic pair that is selected by DAMP. DAMP serves to eliminate from consideration as modes, things such as sensor dynamics, autopilot filters and prefilters, and even certain modes including perhaps the engine.

1.2 AMPOUT, Floating point, preset value = -20.0

Modes that are selected by DAMP whose peaks resonate below AMPOUT, are not considered when forming the constraint equations or cost function. Units on AMPOUT are in decibels. However, all modes that are selected by DAMP do enter the "maximize margins" figure-of-merit and the Margin Counter.

1.3 SCALF (i), Floating point array (51 dimensional), preset values are given in Table 10.7, where the index i corresponds to the first 51 elements of the Margin Array.

TABLE 10.7 SCALF

1., 1.,.2,	rigid body margins
1.,.111, .2, .2, 1., 1., .111, .2, .2, 1.,	1st structural modes
1., .111, 2 * .2, 3 * 1., 1., .111, 2 * .2, 3 * 1.,	2nd structural modes
1., .111, 1., .111, 1., .111, 1., .111, 1., .111, 1.,.111,	3rd to 8th structural modes
8 * .2,	all 8 fuel slosh modes
1.,	crossover frequency
3 * 1.,	load relief variables

SCALF is an array of multiplying factors that are used to scale the elements of the Margin Array, and has units of (1) db/db, (2) db/deg, or (3) nondimensional.

<u>1.4 MAROPT</u>, Integer, preset value = 0

MAROPT is used to distinquish between the 1st mode of Figure 10.1 and the 1st mode of Figure 10.2. In both cases, the "frontside" phase margin is between 180° and 360°, and the "backside" phase margin is between 0° and 180°. If MAROPT = 1, COEBRA assumes Figure 10.1, and (a) sets the frontside phase margin to a negative margin, (b) sets the backside margin to a positive margin, and (c) attempts to "lag" the mode in the cost function.

1.4 (Continued)

If MAROPT = 0, COEBRA assumes Figure 10.2, and (a) sets the frontside to a positive margin, (b) sets the backside to a negative margin, and (c) "leads" the mode.

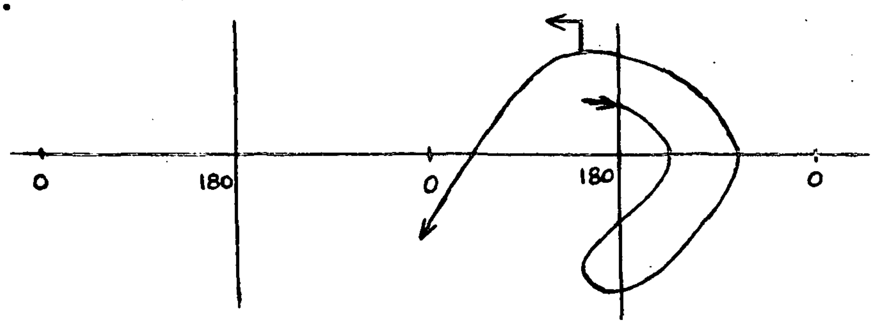


Figure 10.1 MAROPT = 1

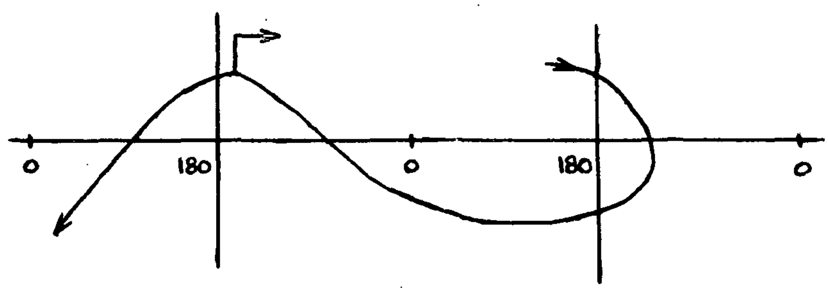


Figure 10.2 MAROPT = 0

1.5 NOTERM, Integer, Preset value = 0

If NOTERM = 0, the first 3 rigid body margins must exist, or COEBRA will terminate. Exceptions: (1) the aerodynamic gain margin if it does not exist at all, and (2) the rigid-body gain margin if fuel slosh modes are included.

If NOTERM = 1, the first 3 rigid body margins do not have to exist.

This section defines the COBIN namelist variables that are used to form the Cost Functions.

2.1 COSPEC (i), Floating point array (48 dimensional), preset values are given in Table 10.8, where the index i corresponds to the first 48 elements of the Margin Array.

TABLE 10.8 COSPEC

6., 6., 30.,	rigid body margins
20., 0., 45., 60., 10., 20., 0., 45., 60., 10.,	1st structural modes
20., 0., 60., 60., 3 * 10., 20., 0., 60., 60., 3 * 10.,	2nd structural modes
-10., 0., -10., 0., -10., 0., -10., 0., -10., 0., -10., 0.,	3rd to 8th structural modes
8 * 30.,	all 8 fuel slosh modes
0.,	crossover frequency

The COSPEC array contains the stability margin objectives that are used to form the "maximize margins" cost function. The COSPEC array is input in decibels, degrees, and rad/sec.

2.2 WTYPE Floating point, preset value = 0.0

The "maximize margins" cost function has two weighting factors: (1) W that is computed by the program; and (2) WTFAC that is input by the user.

If WTYPE = 1., all the elements of W will be unity. If WTYPE = 0., all the elements of W will be the "desired over nominal" type as defined in Chapter 3.

2.3 LOADOP, Integer, preset value = 0

If LOADOP = 0, COEBRA will optimize stability margins.

If LOADOP = 1, COEBRA will optimize load relief.

This section defines the COBIN namelist variables that are used to form the autopilot variable constraint equations.

3.1 XMIN (j), Floating point array (65 dimensional), preset values - none, problem dependent.

This array specifies the minimum values allowed for the autopilot variables. The index j corresponds to the stand-alone autopilot variables in the order in which they are input in the QD030 namelists for all time points. In other words, each "-1" value in all the NTYPE arrays, must have its corresponding XMIN value.

3.2 XMAX (j), Floating point array (65 dimensional), preset values - none, problem dependent

This array specifies the maximum values allowed for the autopilot variables. Each value in the XMIN array must have a corresponding value in the XMAX array.

3.3 P, Floating point, preset value = 0.1

This variable is the initial value to be used for P at the beginning of each major iteration. P is the autopilot variable step-size, and on any iteration, has the same value for all the autopilot variables. P is optimized on what are referred to as minor iterations.

3.4 PSMALL, Floating point, preset value - 0.1

In the optimization of the value of P, PSMALL serves as a convergence criteria on two items:

(1) PSMALL is the smallest value allowed for P. COEBRA will terminate when P becomes less than PSMALL, since this means that the present nominal autopilot (the autopilot at the beginning of this major iteration) is the best autopilot that can be obtained.

(2) PSMALL is the smallest value allowed for Δ P which is defined as $\left| P \text{ (present)} - P \text{ (last)} \right|$. When Δ P becomes less than PSMALL, COEBRA will begin the next major iteration.

By inputting PSMALL with the same value as P, the minor loop will be bypassed. The initial value of P will then be used unchanged on each major iteration.

Part B, Section 4

This section defines the COBIN namelist variables that are associated with the 2 Figures-of-merit and the Margin Counter.

4.1 TOLMAC, Floating point, preset value = 0.03

TOLMAC is a one-sided tolerance on the decisions made in forming the Margin Counter. A margin is counted as "met" if its scaled value exceeds the quantity:

(1.0 - TOLMAC) * (the corresponding scaled SPEC value).

This tolerance is needed to avoid the situation where a margin is met on one iteration, and then, due to nonlinearities, is only slightly not met on the next. Table 10.9 shows the effect of TOLMAC = 0.03.

4.2 TOLFGM, Floating point, preset value = 0.01

TOLFGM is a tolerance on the decisions in the minor loop that are determined by the Figure-of-merit. If the Figure-of-merit from the present iteration, exceeds the quantity.

(1.0 + TOLFGM) * (the Figure-of-merit from the last iteration),

the present iteration is considered to be an improvement over the last. Otherwise, the last is assumed better than the present.

TABLE 10.9 TOLMAC = 0.03

Margin	SPEC value	Scaled SPEC value*	Margin will be counted as met if it exceeds this . . . Scaled Value	Unscaled value*
(1) aero gain	6. db	1.995	1.935	5.75 db
(2) rigid-body gain	6. db	1.995	1.935	5.75 db
(3) rigid-body phase	30. deg.	1.995	1.935	28.75 deg.
(4) 1st mode front-side phase	45. deg.	2.818	2.733	43.75 deg.
(5) 1st mode back-side phase	60. deg.	3.981	3.862	58.75 deg.
(6) 1st mode closest approach	10. db	3.162	3.067	9.75 db
(7) 2nd mode front-side phase	60. deg.	3.981	3.862	58.75 deg.
(8) 2nd mode back-side phase	60. deg.	3.981	3.862	58.75 deg.
(9) 2nd mode closest approach	10. db	3.162	3.067	9.75 db
(10) 3rd to 8th mode peak gain	10. db	3.162	3.067	9.75 db
(11) Fuel slosh back-side phase	20. deg.	1.584	1.536	18.75 deg.
(12) Crossover frequency	2. rps	2.	1.94	1.94 rps

*This is computed using the preset values of the SCALF array.

Part B, Section 5

This section defines the COBIN namelist variables that are used when setting up the Stability Margin and Drift Minimum Constraint Equations.

5.1 SPEC (i), Floating point array (48 dimensional), preset values are given in Table 10.10, when the index i corresponds to the elements in the Margin Array. The units on the SPEC array are in decibels, degrees, and rad/sec.

TABLE 10.10 SPEC

6., 6., 30.,	the rigid-body margins
20., 0., 45., 60., 10., 20., 0., 45., 60., 10.,	1st structural modes
20., 0., 60., 60., 3 * 10., 20., 0., 60., 60., 3 * 10.,	2nd structural modes
-10., 0., -10., 0., -10., 0., -10., 0., -10., 0.,-10., 0.,	3rd to 8th structural modes
8 * 20.,	fuel slosh modes
0.,	crossover frequency

The SPEC array specifies the minimum requirements that are put on the stability margins and closed-loop root locations. The SPEC array is used when forming the Stability Margin constraint equations.

5.2 STEP, Floating point, preset value = 1.0

This is the initial value to be used for STEP at the beginning of each minor iteration. STEP is the "percent improvement" required in each "margin" relative to its SPEC value or required value. On each minor iteration (each value of P), COEBRA maximizes the value of STEP. This is done in the inner loop. STEP applies to all of the so-called "not-met Margin Constraint Equations" on stability margins, root locations, drift minimum, and the autopilot vector constraints.

5.3 DSTEP, Floating point, preset value = 0.2

In the inner loop, STEP is maximized as follows. STEP begins with its initial value (input as STEP). If a feasible solution does not exist, STEP is reduced by DSTEP using the following equation

$$\text{STEP} \longleftarrow \text{STEP} - \text{DSTEP}$$

until a feasible solution is obtained. The minimum value for STEP is zero, for which a feasible solution is virtually guaranteed.

5.4 DRFTOL, Floating point, preset value = 0.0

DRFTOL is the tolerance used in the Drift Minimum Constraint Equations. This tolerance is referred to as ϵ in Chapter 8. As shown in Chapter 8, this is approximately the tolerance that is allowed on the frequency of the drift root that would be exactly at the origin for "perfect" drift minimum. For example, if DRFTOL = .1, this requires that the drift root be within $\pm$ 0.1 rad/sec of the origin.

This section defines miscellaneous COBIN namelist variables.

6.1 NITER, Integer, preset value - none, problem dependent

COEBRA will terminate after NITER major iterations.

6.2 MFRESP, Integer, preset value = 0

If MFRESP $\neq$ 0, the frequency response resulting from each minor iteration (each value of P), will be printed out and plotted.

6.3 DELPAR, Floating point, preset value = 0.02

DELPAR is the percent disturbance that is made on each autopilot variable when computing all the partial derivatives. The partial derivative of the i^{th} element of the Margin Array with respect to the j^{th} autopilot variable is defined as:

$$\frac{\text{Disturbed } M(i) - \text{Nominal } M(i)}{\text{DELPAR} * \text{Nominal } X\ (j)}$$

Sometimes when disturbing an autopilot variable in order to compute partial derivatives, the "disturbed" Margin Array will differ from the "nominal" array because one or more elements have been added or taken away due to the disturbance. When this occurs, the sign on DELPAR is reversed. If an element again has been added or taken away, the derivative of that element with respect to that variable is set to zero. The solution of reversing the sign on DELPAR is also used when a disturbance causes the damping ratio of an autopilot quadratic filter to become greater than or equal to unity.

6.3 (Continued)

When computing the partial derivatives, the elements of the Margin Array are always refound. In other words, the derivative is not calculated at a certain frequency.

Obviously, the partial derivatives are calculated after the elements of the Margin Array have been scaled, and are calculated only once per major iteration.

6.4 PRINT, Integer, preset value = 0

PRINT is a flag that can be used to obtain additional printout for checkout and debugging purposes.

If PRINT = 1, (a) Case 13 numerator roots are printed out for all calls to the polynomial factoring routine, and (b) various computer dumps are printed out.

IF PRINT = 0, this flag is disabled.

6.5 NPRINT, Integer, preset value = 0

NPRINT is a flag that can be used to obtain additional printout for checkout and debugging purposes.

If NPRINT = 1, the initial tableau for each entry to the Simplex Algorithm is printed out.

If NPRINT = 2, all tableaus of the Simplex are printed out.

If NPRINT = 0, this flag is disabled.

6.6 KROOT, Integer, preset value = 0

If KROOT $\neq$ 0, the ROOTCH subroutine (Chapter 7) is to be used. The ROOTCH subroutine allows autopilot quadratic roots to become real roots if the effective damping ratio becomes unity, or vice-versa for real roots to become complex. The ROOTCH subroutine will not be used when optimizing load relief since the filters in the load relief control law must be real roots.

CHAPTER 11. SAMPLE LISTINGS

11.1 COEBRA Deck Setup

Table 11.1 summarizes the deck setup for a COEBRA run.

TABLE 11.1 COEBRA DECK SETUP

	Explanation
COEBRA	COEBRA caller card (columns 2 to 7)
Title Card	Title card for time point #1* (columns 2 to 43)
$INDATA	INDATA namelist data for time point #1 ($ in column 2)
:	
$END	
$COBDE	COBDE namelist data for time point #1 ($ in column 2)
:	
$END	
Title Card	Title card for time point #2*
$INDATA	INDATA namelist data for time point #2
:	
$END	
$COBDE	COBDE namelist data for time point #2
:	
$END	
:	
	Etc. for remaining time points or vehicle states.
:	
ENDBLK	ENDBLK card (columns 2 to 7)
$COBIN	COBIN namelist data
:	
$END	
ENDRUN	ENDRUN card (columns 2 to 7)

* Title card must not have a $ in column 2.

11.2 QD030 Deck Setup

Table 11.2 summarizes the deck setup for a QD030 run.

TABLE 11.2 QD030 DECK SETUP

	Explanation
QD030	QD030 caller card (columns 2 to 6)
Title Card	Title card for first case* (columns 2 to 43)
$INDATA	INDATA namelist data for first case.**
	($ in column 2)
:	
$END	
Title Card	Title card for second case*
$INDATA	INDATA namelist data for second case.**
:	
$END	
:	
:	Etc. for remaining cases.
ENDRUN	ENDRUN card (columns 2 to 7)

* Title card must not have a $ in column 2.

**Note the function of the variable called NFLAGG for multiple case runs.

11.3 Sample Listings

This section contains the listings of the input data used to generate Examples 1 to 6 of Volume I of this report. This section also includes a sample listing of a QD030 test case. The following is a summary of these listings.

(1) Example #1 (Run 1) of Volume I

(2) Example #2 (Run 2) of Volume I

(3) Example #3 of Volume I

(4) Example #4 of Volume I

(5) Example #5 of Volume I

(6) Example #6 of Volume I

(7) QD030 Sample Listing

EXAMPLE #1 (RUN 1) OF VOLUME I

```
COEBRA
  EXAMPLE 1  RUN 1  VOLUME 1
$INDATA
 NCASE=13.,  NFREQ=-13.,
 RF=1.,0.,0.,1.,0.,0.,1.,20.,20.,
 RG=3.,0.,0.,2.,0.,1.,.75,-1.,.9,-.134,1.,2.17,
 RN(7)=1.,
 RO(7)=1.,RU(7)=1.,
 NFLAGC=1
$END
$COBDE
 NTOTAL=10,  NTYPE=0,-1,-1,7*0,
 IOMEGA=1,  WTFAC(48)=0.
$END
ENDBLK
$COBIN  NITER=20, P=.2,  PSMALL=.05,
  XMAX=2*1.E6,  XMIN=2*1.E-6,
 STEP =1.,  DSTEP =.2,
 SPEC=6.,12.,50.,  SPEC(48)=.2,  COSPEC=6.,12.,50.,  COSPEC(48)=.2,
   TOLFGM=.01,
 MFRESP=1,  TOLMAC=.03
$END
ENDRUN
```

EXAMPLE #2 (RUN 2) OF VOLUME I

```
COEBRA
  EXAMPLE 2  RUN 2  VOLUME 1
$INDATA
 NCASE=13.,  NFREQ=-13.,
    RF=2.,0.,0.,2.,0.,0.,1.,10.,.2,10.,.2,
 RG=0.,0.,0.,2.,0.,1.,100.,.125,.05,
 RN(7)=1.,
 RO(7)=1.,RU(7)=1.,
 NFLAGG=1
$END
$COBDE
  NTOTAL=9,  NTYPE=0,4*-1,4*0,
 TOMEGA=1,  WTFAC(48)=1.
$END
ENDBLK
$COBIN  NITER=10, P=.2,  PSMALL=.05,
  XMAX=4*1.E6,  XMIN=4*1.E-6,
 STEP =1.,  DSTEP =.2,
 SPEC=6.,10.,45.,   SPEC(48)=10.,
 COSPEC=6.,10.,45.,  COSPEC(48)=10.,
    TOLFGM=.01,
 MFRESP=1,  TOLMAC=.03
$END
ENDRUN
```

EXAMPLE #3 OF VOLUME I

```
COEBRA
 EXAMPLE 3      VOLUME 1
$INDATA
 RO= 0, 0, 0, 3, 8, 1,  1.00000000E+00,
   -1.36262989E+00,     1.23207033E+00,     -1.86078583E+01,
    1.49987917E-02,     8.64379519E+01,      1.49889949E-02,
    6.10546338E+01,     1.49810447E-02,      4.36472340E+01,
    1.49923601E-02,     4.10520076E+01,      1.49752393E-02,
    3.65225962E+01,     9.99467577E-03,      3.31305402E+01,
    7.92148654E-03,     6.21674075E-00,      1.49872592E-02,
    1.12915627E+02,
 RS= 3, 7, 1, 0, 0, 0,  4.46372564E+00,
    2.66287273E-02,    -2.63300666E-02,      2.50870999E+01,
   -2.48449628E-01,     8.55145836E+01,      2.76766317E-01,
    8.55150711E+01,    -6.79886314E-02,      4.25055846E+01,
    9.75573872E-02,     4.25261830E+01,     -1.46242991E-01,
    3.17005789E+01,     1.71655895E-01,      3.12444246E+01,
    1.34779036E-02,     1.01408910E+01,
 RT= 1, 8, 1, 0, 0, 0,  4.55489160E+00,
    2.42102938E+01,     1.48004139E-02,      8.95946461E+01,
    5.36334728E-01,     7.92111798E+01,     -5.08212644E-01,
    8.02812150E+01,     2.25337946E-01,      4.38882125E+01,
    1.49831127E-02,     4.12408766E+01,     -1.92225544E-01,
    4.38669150E+01,     7.98200171E-03,      3.16898786E+01,
   6.39219883E-03,    5.31199855E-00,
 RU= 3, 7, 1, 0, 0, 0,  4.46372564E+00,
    2.66287273E-02,    -2.63300666E-02,      2.50870999E+01,
   -2.48449628E-01,     8.55145836E+01,      2.76786317E-01,
    8.55150711E+01,    -6.79886314E-02,      4.25055846E+01,
    9.75573872E-02,     4.25261830E+01,     -1.46242991E-01,
    3.17005789E+01,     1.71655895E-01,      3.12444246E+01,
    1.34779036E-02,     1.01408910E+01,
```

EXAMPLE #3 OF VOLUME I (Cont'd.)

```
NCASE=4,10,11,12,13,
NFREQ=10,11,12,-13,
  KVECT=1,
RG(4)=2.0,0.0,0.0,1.0,.01,.01,
RG(7)=.490,
RL(4)=2.,0.,0.,1.,.03333,.03333,
RL(7)=4.105,
RM(3)=1.,2.,0.,0.,1.,.2,.2,
RM(7)=2.392,
RN(3)=1.,2.,0.,0.,1.,.2,.2,
RN(7)=.01,
  RL(7)=4.,  RM(7)=1.6,   RN(7)=.8,
 NFLAGG=1
$END
$COBDE  NTOTAL=12,  NTYPE=3*0,9*-1,  IDMODE=1,2,3,4,5,6,7,8,
  WTFAC(4)=0.,1.,  WTFAC(14)=0.,1.,
  WTMARG(34)=0., WTMARG(36)=0., WTMARG(38)=0.,
  IOMEGA=1,  WTFAC(48)=0.
$END
ENDBLK
$COBIN  XMAX=9*100.,  XMIN=9*.01,  MFRESP=1,NOTERM=1,  TOLMAC=.03,
   P=.2,  PSMALL=.05,  STEP=1.,  DSTEP=.4,
  COSPEC(6)=100.,  COSPEC(7)=100.,  COSPEC(16)=100.,  COSPEC(17)=100.,
  COSPEC(8)=20.,  COSPEC(20)=20.,
  SPEC(48)=2.,  COSPEC(48)=2.,
    TOLFGM=.005,
 NITER=5
$END
ENDRUN
```

EXAMPLE #4 OF VOLUME I

```
COEBRA
 EXAMPLE 4     VOLUME 1
$INDATA
 RO= 0, 0, 0, 3, 8, 1,   1.00000000E+00,
   -1.36262989E+00,      1.23207033E+00,     -1.86078583E+01,
    1.49987917E-02,      8.64379519E+01,      1.49809949E-02,
    6.10546338E+01,      1.49910447E-02,      4.36472340E+01,
    1.49923601E-02,      4.10520076E+01,      1.49752393E-02,
    3.05225962E+01,      9.99467577E-03,      3.31305402E+01,
    7.92148654E-03,      6.21674075E-00,      1.49872592E-02,
    1.12915627E+02,
 RS= 3, 7, 1, 0, 0, 0,   4.46372564E+00,
    2.66287273E-02,     -2.63300666E-02,      2.50870999E+01,
   -2.48449628E-01,      8.55145836E+01,      2.76766317E-01,
    8.55150711E+01,     -6.79886314E-02,      4.25055846E+01,
    9.75573872E-02,      4.25261830E+01,     -1.46242991E-01,
    3.17005789E+01,      1.71655895E-01,      3.12444246E+01,
    1.34779036E-02,      1.01408910E+01,
 RT= 1, 8, 1, 0, 0, 0,   4.55489160E+00,
    2.42102938E+01,      1.48004139E-02,      8.95946461E+01,
    5.36334728E-01,      7.92111798E+01,     -5.08212644E-01,
    8.02812150E+01,      2.25337946E-01,      4.38882125E+01,
    1.49831127E-02,      4.12408766E+01,     -1.92225544E-01,
    4.38669150E+01,      7.98200171E-03,      3.16898786E+01,
  6.39219883E-03,    5.31199855E-00,
 RU= 3, 7, 1, 0, 0, 0,   4.46372564E+00,
    2.66287273E-02,     -2.63300666E-02,      2.50870999E+01,
   -2.48449628E-01,      8.55145836E+01,      2.76786317E-01,
    8.55150711E+01,     -6.79886314E-02,      4.25055846E+01,
    9.75573872E-02,      4.25261830E+01,     -1.46242991E-01,
    3.17005789E+01,      1.71655895E-01,      3.12444246E+01,
    1.34779036E-02,      1.01408910E+01,
```

EXAMPLE #4 OF VOLUME I (Cont'd.)

```
NCASE=4,10,11,12,13,
NFREQ=10,11,12,-13,
RO(4)=2.0,0.0,0.0,1.0,.01,.01,
RG(7)=.490,
RL=0.,0.,0.,2.,0.,0.,.6,.03333,.03333,
RM=0.,0.,1.,2.,0.,0.,.3,.03333,.03333,
RN=0.,0.,1.,2.,0.,0.,.5,.03333,.03333,
 NFLAGG=1
$END
$COBDE  NTOTAL=12,  NTYPE=3*0,9*-1,  IDMODE=1,2,3,4,5,6,7,8,
  WTFAC(4)=0.,1.,  WTFAC(14)=0.,1.,
   WTMARG(16)=0.,  WTMARG(17)=0.,
  WTMARG(36)=0.,  WTMARG(38)=0.,
  TOMEGA=1,  WTFAC(48)=1.
$END
ENDBLK
$CORIN  XMAX=9*100.,  XMIN=9*.01,  MFRESP=1,NOTERM=1,  TOUMAC=.03,
   P=.2,  PSMALL=.05,  STEP=1.,  DSTEP=.4,
  COSPEC(6)=100.,  COSPEC(7)=100.,  COSPEC(16)=100.,  COSPEC(17)=100.,
  SPEC(48)=2.,  COSPEC(48)=2.,
  COSPEC(8)=20.,  COSPEC(20)=20.,
    TOLFGM=.01,
    TOLFGM=.005,
  NITER=5,
$END
ENDRUN
```

EXAMPLE #5 OF VOLUME I

```
COEBPA
 EXAMPLE 5  VOLUME I  T=30AGC
$INDATA
NCASE=4,13, NFREQ=-13,
 RK=1.,0.,0.,3.,0.,0.,1.,80.,374.,16.7,20.,
 RK(7)=.00707,
 RL=4*0.,1.,0.,1.,.6,.135,
 RL(7)=.822,
 RM=0.,0.,0.,1.,1.,0.,1.,1.19,.4,.35,
 RM(7)=.262,
 RN=0.,0.,0.,1.,1.,0.,1.,1.19,.5,.3,
 RN(7)=.0594,
 RO=0.,0.,0.,6.,3.,0.,1.,-88.70598,80.97426,
 -11.3614E2,  1.035811,  1.038851,  1.,  8.220445E-3,
 .2136856,  .01101264,  .3981987,  .01187258,  .5442695,
 RR=6.,4.,0.,0.,1.,0.,-1.75596259E3,
   -1.15117389E+03,     9.98767955E-01,      9.34622403E-01,
   -7.88286993E-01,     3.84944887E-01,     -1.00000000E-00,
    5.06454657E-02,     1.38523307E-02,      1.11189067E-02,
    2.65355323E-01,     9.67332786E-03,      3.77965995E-01,
    1.04464851E-02,     4.96366685E-01
 .5953986,  .6378408,
 RS=8.,2.,0.,0.,0.,0.,4.76909913,
    2.98892800E+03,     9.98600210E-01,      9.23983165E-01,
   -3.68945503E-00,     3.75854172E-00,     -8.45142839E-01,
    2.47812090E-01,    -1.00000000E-00,      1.17167581E-02,
    5.14546021E-01,     9.87638644E-03,      3.05082069E-01,
 RT=7.,4.,1.,0.,2.,0.,24.1625184E+1,
    2.93470770E+03,     9.99972316E-01,      9.27140897E-01,
   -9.71796744E-01,    -2.18829965E-01,      5.76871537E-01,
   -1.00000000E-00,     7.38243757E-03,      1.95112459E-01,
    1.05117631E-02,     3.83116955E-01,      1.09332693E-02,
    4.99957739E-01,     9.99939556E-01,      1.01301196E+00,
 .999896,  1.017521,  .8908525,  1.36203,
```

EXAMPLE #5 OF VOLUME I (Cont'd.)

```
RU=9.,3.,1.,0.,2.,0.,2.38913388E+2,
   2.98099843E+03,      2.87633795E-00,      9.99972044E-01,
  -2.81125906E-00,      9.24695562E-01,     -9.69932268E-01,
 -0.2087,  .05703439,-1.,.01011324,
  .3134832,  .01170666,  .512918,  .9999376,
 1.012925,  .999896,  1.017521,  .8908525,  1.36203,
SLOW=.001,UPPER=5.,
NFLAGG=1,
$END
$COBDE  U=8.46E3,  B=8.46E3,  QBAR=2.96786,  CYB=7.26,  TY=2.0671E6,
  CNB=21.3,  IZZ=2.446E8,  LGY=478.9,  LAY=463.1,
  KIVC=1.075,  MASS=2931.,  THETA=73.31,  G=386.,
       SF=1.13E4,     D=120.,   U1=.0021,      U2=.0228,     U3=-.0038,
  TW1=52.5,  TW2=55.,  TW3=80.,  TS=.04,  LRCOST=1,0,0,
 LKD=5HRL(7),
 LKR1=5HRM(7),
  LKR2=5HRN(7),
  LKA2=5HRK(7),
 LA4=5HRK(8),
  LA5=5HRK(9),
  LA6=5HRK(10),
  LA7=5HRK(11),
 NTOTAL=16,
  NTYPE=-1,-1,0,2*-1,-1,2*-1,-1,3*-1,-1,3*-1,
 IDMODE=1,2,3,
  WTFAC(49)=1.,0.,0.,
  WTMARG(49)=1.,0.,0.,
$END
 EXAMPLE 5  VOLUME I  T=60
$INDATA
NCASE=4,13, NFREQ=-13,
 RK=1.,0.,0.,3.,0.,0.,1.,80.,374.,16.7,20.,
 RK(7)=.005487,
 RL=4*0.,1.,0.,1.,.6,.135,
 RL(7)=.9211,
```

EXAMPLE #5 OF VOLUME I (Cont'd.)

```
RM=0.,0.,0.,1.,1.,0.,1.,1.19,.4,.35,
RM(7)=.2836,
RN=0.,0.,0.,1.,1.,0.,1.,1.19,.5,.3,
RN(7)=.0648,
RO=0.,0.,0.,6.,3.,0.,1.,-48.16893,46.81498,-30.05966E2,
1.035811,  1.038851,  1.,  .01123759,  .4406548,
 8.234303E-3,  .2240668,  .01259356,  .6471467,
RR=6.,4.,0.,0.,1.,0.,-1.95349645E3,
  -3.00910695E+03,     9.98767664E-01,     9.34597184E-01,
  -7.88153915E-01,     3.84590404E-01,    -1.00000000E-00,
   2.24812948E-02,     2.50360527E-02,     1.20631219E-02,
   2.89856098E-01,     8.95818925E-03,     4.02176668E-01,
.01151462,  .6052149,  .5953986,  .6378408,
RS=8.,2.,0.,0.,0.,0.,6.44391765,
   2.10680258E+03,     3.83357417E-00,    -3.76883490E-00,
   9.98599185E-01,     9.23851054E-01,    -8.44785850E-01,
   2.47241897E-01,    -1.00000000E-00,     1.02450285E-02,
   3.49419931E-01,     1.24259669E-02,     6.15143308E-01,
RT=7.,4.,1.,0.,2.,0.,3.25925849E+2,
   2.04828151E+03,     9.99972318E-01,     9.27159364E-01,
  -9.71814448E-01,    -2.18857635E-01,     5.76879072E-01,
  -1.00000000E-00,     7.04523835E-03,     1.96523682E-01,
   1.09041241E-02,     4.28549614E-01,     1.18088561E-02,
   6.10120527E-01,     9.99939572E-01,     1.01301270E+00,
.999896,  1.017521,  .8908525,  1.36203,
RU=9.,3.,1.,0.,2.,0.,3.22667063E+2,
   2.09933528E+03,     3.08466583E-00,     9.99972061E-01,
  -3.01833943E-00,     9.24855482E-01,    -9.70077085E-01,
  -2.09250706E-01,     5.70646257E-01,    -1.00000000E-00,
   1.04398213E-02,     3.59480486E-01,     1.23752562E-02,
   6.18269993E-01,     9.99937724E-01,     1.01293102E+00,
.999896,  1.017521,  .8908525,  1.36203,
SLOW=.001,UPPER=5.,
NFLAGG=1,
$END
```

EXAMPLE #5 OF VOLUME I (Cont'd.)

```
$COBDE  U=1.90015E4,  B=1.90015E4,  QBAR=5.91634,  CYB=8.745,  TY=1.849804E6,
   CNB=28.354,  IZZ=2.10317E8,  LGY=490.066,  LAY=451.938,
  KTVC=1.142,  MASS=2322.132, THETA=52.,  G=386.,
        SF=1.13E4,      D=120.,    U1=.0021,      U2=.0228,      U3=-.0038,
  TW1=52.5,  TW2=55.,  TW3=80.,  TS=.04,  LRCOST=1.0.0,
 LKD=5HRL(7),
 LKR1=5HRM(7),
  LKR2=5HRN(7),
  LKA2=5HRK(7),
 LA4=5HRK(8),
  LA5=5HRK(9),
  LA6=6HRK(10),
  LA7=6HRK(11),
 NTOTAU=16,
  NTYPE=-1, 1,0,2* 1,-1,2* 1,-1,3* 1,-1,3* 1,
 IPMODE=1,2,3,
  WTFAC(49)=1.,0.,0.,
  WTMARG(49)=1.,0.,0.,
$END
 EXAMPLE 5  VOLUME I  T=80BGC
$INDATA
NCASE=4,13, NFREQ=-13,
 RK=1.,0.,0.,3.,0.,0.,1.,80.,374.,16.7,20.,
 RK(7)=.0053,
 RL=4*0.,1.,0.,1.,.6,.135,
 RL(7)=.564,
 RM=0.,0.,0.,1.,1.,0.,1.,1.19,.4,.35,
 RM(7)=.178,
 RN=0.,0.,0.,1.,1.,0.,1.,1.19,.5,.3,
 RN(7)=.0445,
```

EXAMPLE #5 OF VOLUME I (Cont'd.)

```
RO=0.,0.,0.,6.,3.,0.,1.,-61.01003,  59.9327,  -62.83569E2,
1.035811,  1.038851,  1.,  .01136683,  .4630344,
8.236139E-3,  .229044,  .01669305,  .1096366E1,
RR=8.,3.,0.,0.,1.,0.,-2.66565847E3,
  -6.28572141E+03,     9.98999948E-01,     9.52802728E-01,
  -9.01145142E-01,     6.58869697E-01,    -3.02460694E-01,
1.58816375E-1,        -1.00000000E-00,     1.20297447E-02,
   2.22946956E-02,     1.00862593E-02,     2.83893621E-01,
   1.03386251E-02,     4.26044788E-01,
 .5953986,  .6378408,
RS=10.,1.,0.,0.,0.,0., 11.3551204,
   4.57762182E+03,     2.86813702E-00,    -6.98001603E-01,
  -2.78608616E-00,     1.12200077E-00,     9.98100797E-01,
  -1.13578850E-00,     8.73664210E-01,     1.82185418E-01,
  -1.00000000E-00,     1.08068706E-02,     4.02539284E-01,
RT=7.,4.,1.,0.,2.,0.,5.71640157E+2,
   4.50381746E+03,     9.99972248E-01,     5.74302747E-01,
  -9.71458892E-01,    -2.14698647E-01,     9.26568611E-01,
  -1.00000000E-00,     6.97531912E-03,     1.96853952E-01,
   1.11495739E-02,     4.54284946E-01,     9.99939119E-01,
   1.01299230E+00,     1.43288925E-02,     9.95152883E-01,
.999896,  1.017521,  .8908525,  1.36203,
RU=9.,3.,1.,0.,2.,0.,5.68110582E+2,
   4.57084269E+03,     2.57985616E-00,     9.99972208E-01,
  -2.51380429E-00,     9.26039042E-01,    -2.16877332E-01,
  -9.70703174E-01,     5.76083637E-01,    -1.00000000E-00,
   1.09109598E-02,     4.06317388E-01,     9.99938606E-01,
   1.01296925E+00,     1.94164280E-02,     1.24786584E+00,
 .999896,  1.017521,  .8908525,  1.36203,
SLOW=.001,UPPER=5.,
NFLAGG=1,
$END
```

EXAMPLE #5 OF VOLUME I (Cont'd.)

```
$COBDE
 NTOTAL=16,
  NTYPE=-1, 1,0,2* 1,-1,2* 1,-1,3* 1,-1,3* 1,
 IDMODE=1,2,3,
  LRCOST=0,0,0,
$END
ENDBLK
$COBIN  DAMP=.02,  LOADOP=1,  P=.05,  PSMALL=.05,  STEP=0.,  DSTEP=0.,
   NPRINT=1,
 XMIN=23*1.E-5,  XMAX=23*1.E5,  NITER=1
$END
ENDRUN
```

EXAMPLE #6 OF VOLUME I

```
COBRA
  EXAMPLE 6  VOLUME I  T=27AGC
$INDATA
NCASE=4,13,  NFREQ=-13,
RG=0.,0.,0.,2.,0.,0.,1.,.01,.01,
RO(7)=.702,
RK=1.,0.,0.,4.,0.,0.,.00504,3.04,.5,.25,.25,7.5,
RL=0.,0.,0.,2.,0.,0.,1.,.1,.1,
RL(7)=.655,
RM=0.,0.,1.,3.,1.,0.,1.,.0714,.0714,.001471,.76,147.5,
RM(7)=1.0,
RN=0.,0.,1.,2.,0.,0.,1.,.0833,.0714,
RN(7)=.135,
 RO=0   ,0   ,0   ,3.0,3.0,1.0,  1.00000000E+00,
   -1.63958849E+00,     1.49811473E+00,    -2.07721237E+01,
    9.99006447E-03,     1.65572944E+01,     7.96848670E-03,
    9.62832509E-00,     1.00007861E-02,     2.26196234E+01,
 RR=1.0,4.0,1.0,0   ,0   ,0   , -1.45211115E+03,
   -2.11387064E+01,     9.13289708E-03,     2.13389421E+01,
   -6.51269248E-02,     1.50241911E+01,     8.66551815E-02,
    1.49309498E+01,     6.39449900E-02,     5.68120405E-01,
 RS=3.0,2.0,1.0,0   ,0   ,0   ,  3.83658536E+00,
    4.53690771E-02,    -4.37018578E-02,     5.49242862E+01,
    1.00019467E-02,     2.26083941E+01,     9.90118311E-03,
    1.52586588E+01,
 RT=1.0,3.0,1.0,0   ,0   ,0   ,  3.89601433E+00,
    5.37675663E+01,     9.84746578E-03,     2.22473583E+01,
    9.92026512E-03,     1.64173310E+01,     7.16734721E-03,
    8.76949880E-00,
 RU=3.0,2.0,1.0,0   ,0   ,0   ,  3.83658536E+00,
    4.53690771E-02,    -4.37018578E-02,     5.49242862E+01,
    1.00019467E-02,     2.26083941E+01,     9.90118311E-03,
    1.52586588E+01,
 NFLAGC=1
$END
```

EXAMPLE #6 OF VOLUME I (Cont'd.)

```
$CORDE   U=8009.,  B=8009.,  QBAR=2.8447, CYB=6.88,  TY=2.151098E6,
 CNB=26.13,  IZZ=2.46E8,  LGY=470.6,  LAY=620.4,
  KIVC=.702,  MASS=2921.656,  THETA=82.4716,G=386.27,
       SF=1.1304E4,    D=120.,    U1=.0021,       U2=.0228,      U3=-.C038,
  TW1=52.5,  TW2=55.,  TW3=80.,               LRCOST=1.0,0,
 LKD=5HRL(7),
 LKR1=5HRM(7),
  LKP2=5HRN(7),
  LKA2=5HRK(7),
 LA4=5HRK(8),
  LA5=5HRK(9),
  LA6=6HRK(10),
  LA7=6HRK(11),
  LA8=6HRK(12),
 NTOTAU=21,
 NTYPE=3*0,12*-1,3*0,3*-1,
   NTYPE(9)=0,
 IDMODE=1,2,3,
  WTFAQ(49)=1.,0.,0.,
  WTMARG(49)=1.,0.,0.,
$END
  EXAMPLE 6  VOLUME I  T=53
$INDATA
NCASE=4,13,  NFREQ=-13,
RO=0.,0.,0.,2.,0.,0.,1.,.01,.01,
RG(7)=.678,
RK=1.,0.,0.,4.,0.,0.,.00504,3.04,.5,.25,.25,7.5,
RL=0.,0.,0.,2.,0.,0.,1.,.1,.1,
RL(7)=.655,
RM=0.,0.,1.,3.,1.,0.,1.,.0714,.0714,.001471,.76,147.5,
RM(7)=1.0,
RN=0.,0.,1.,2.,0.,0.,1.,.0833,.0714,
RN(7)=.135,
 RO=0   ,0   ,0   ,3.0,4.0,1.0,  1.00000000E+00,
   -8.88991154E-01,     8.62623278E-01,    -4.54730151E+01,
    9.99410198E-03,     1.73215744E+01,     7.95729722E-03,
    9.88552627E-00,     9.99986051E-03,     2.43799650E+01,
    1.49970159E-02,     2.44644397E+01,
```

EXAMPLE #6 OF VOLUME I (Cont'd.)

```
RR=1.0,5.0,1.0,0   ,0   ,0   , -1.56726113E+03,
   -4.95539236E+01,      1.53577303E-02,      3.09891265E+01,
    1.37103381E-02,      2.44409562E+01,      5.34197945E-03,
    1.56826376E+01,      1.55046855E-02,      1.47759095E+01,
    2.81936308E-02,      1.03447811E-00,
RS=3.0,3.0,1.0,0   ,0   ,0   ,  4.46115368E+00,
   -4.85364255E-02,      5.01959466E-02,      3.70900584E+01,
    1.40365111E-02,      2.44430515E+01,      1.09437450E-02,
    2.32951635E+01,      9.96676533E-03,      1.65772393E+01,
RT=1.0,4.0,1.0,0   ,0   ,0   ,  4.52557052E+00,
    3.58576923E+01,      1.34445600E-02,      2.44418884E+01,
    1.19020492E-02,      2.51796085E+01,      9.97580356E-03,
    1.72683056E+01,      6.88790708E-03,      8.72567613E-00,
RU=3.0,3.0,1.0,0   ,0   ,0   ,  4.46115368E+00,
   -4.85364255E-02,      5.01959466E-02,      3.70900584E+01,
    1.40365111E-02,      2.44430515E+01,      1.09437450E-02,
    2.32951635E+01,      9.96676533E-03,      1.65772393E+01,
NFLAGG=1
$END
$COBDE  U=1.7612E4,  B=1.7612E4,  QBAR=6.033,  CYB=8.69,  TY=1.886937E6,
  CNB=33.4,  IZZ=2.095E8,  LGY=477.4,  LAY=613.6,
  KTVC=.678,  MASS=2374.546,  THETA=67.4028,G=386.27,
       SF=1.1304E4,   D=120.,   U1=.0021,      U2=.0228,       U3=-.0038,
  TW1=52.5,  TW2=55.,  TW3=80.,             LRCOST=1,0,0,
 LKD=5HRL(7),
 LKR1=5HRM(7),
  LKP2=5HRN(7),
  LKA2=5HRK(7),
 LA4=5HRK(8),
  LA5=5HRK(9),
  LA6=6HRK(10),
  LA7=6HRK(11),
  LA8=6HRK(12),
 NTOTAL=21,
```

EXAMPLE #6 OF VOLUME I (Cont'd.)

```
 NTYPE=3*0,12* 1,3*0,3* 1,
   NTYPE(9)=0,
 IDMODE=1,2,3,4,
  WTFAC(49)=1.,0.,0.,
  WTMAPG(49)=1.,0.,0.,
$END
  EXAMPLE 6  VOLUME I  T=75BGC
$INDATA
NCASE=4,13,  NFREQ=-13,
RG=0.,0.,0.,2.,0.,0.,1.,.01,.01,
RG(7)=.641,
RK=1.,0.,0.,4.,0.,0.,.00504,3.04,.5,.25,.25,7.5,
RL=0.,0.,0.,2.,0.,0.,1.,.1,.1,
RL(7)=.655,
RM=0.,0.,1.,3.,1.,0.,1.,.0714,.0714,.001471,.76,147.5,
RM(7)=1.0,
RN=0.,0.,1.,2.,0.,0.,1.,.0833,.0714,
RN(7)=.135,
 RO=0   ,0   ,0   ,3.0,4.0,1.0,  1.00000000E+00,
   -1.07899963E+00,     1.05769414E+00,     -9.07280062E+01,
    7.94472336E-03,     1.01126072E+01,      9.99803462E-03,
    1.79635522E+01,     9.99295811E-03,      2.60680978E+01,
    1.49973368E-02,     3.08596988E+01,
 RR=3.0,4.0,1.0,0   ,0   ,0   , -2.01811288E+03,
    2.51520279E-02,    -2.41762230E-02,     -9.07834082E+01,
    1.25867380E-02,     2.85753184E+01,      9.83915773E-03,
    1.76061888E+01,     9.87021535E-03,      1.35457027E+01,
    1.59888632E-02,     9.36710312E-01,
 RS=3.0,3.0,1.0,0   ,0   ,0   ,  6.44680604E+00,
    6.63720594E-02,    -6.49663916E-02,      7.61543319E+01,
    1.22125139E-02,     2.78729717E+01,      1.07473337E-02,
    2.03368553E+01,     9.99465555E-03,      1.75132772E+01,
```

EXAMPLE #6 OF VOLUME I (Cont'd.)

```
 RT=1.0,4.0,1.0,0   ,0   ,0   ,   6.52397993E+00,
    7.41800380E+01,     7.49959656E-02,     2.68733070E+01,
   -5.16490396E-02,     2.69080234E+01,     9.98879690E-03,
    1.79372224E+01,     6.73775530E-03,     8.68368353E-00,
 RU=3.0,3.0,1.0,0   ,0   ,0   ,   6.44680604E+00,
    6.63720594E-02,    -6.49663916E-02,     7.61543319E+01,
    1.22125139E-02,     2.78729717E+01,     1.07473337E-02,
    2.03368553E+01,     9.99465555E-03,     1.75132772E+01,
 NFLAGG=1
$END
$COBDE
 NTOTAL=21,
 NTYPE=3*0,12* 1,3*0,3* 1,
   NTYPE(9)=0,
 IDMODE=1,2,3,4,
  LRCOST=0,0,0,
$END
ENDBLK
$COBIN  DAMP=.02,  LOADOP=1,  P=.05,  PSMALL=.05,  STEP=1.,  DSTEP=.2,
 PRINT=0,   NPRINT=1,
 XMIN=16*1.E-5,  XMAX=16*1.E5,   NITER=1
$END
ENDRUN
```

QD030 SAMPLE LISTING

```
QD030
TEST CASE QD030
$INDATA
  NTRAN=-4,KDENZ=2,BT=0.,DT=.1C,FT=30.,
 KVECT=1,
 FOVECT=6.7844,  16.67686,
 NCASE=4,8,9,10,11,12,13,
 NFREQ=8,9,10,11,12,-13,
 RD(7)=-1., RF(7)=-1.,
 RJ=0.,0.,2.,3.,0.,0.,3.2768E-3,1.24416,1.24416,1.24416,
 RK=1.,0.,0.,4.,0.,0.,1.3271E-4,6.63981,7.5,.24576,.24576,.24576,
 RL=0.,0.,0.,2.,0.,0.,1.15947,.0881906,.116122,
 RM=0.,0.,1.,2.,0.,0.,.74,.0606948,.0426265,
 RN=0.,0.,1.,2.,0.,0.,.74,.0606948,.0426265,
RJ(7)=0.15,3*0.3333,
RK(7)=2.48E-3,2.96,7.5,3*0.3333,
RL(7)=1.4,0.110592,0.110592,
RM(7)=0.413108,2*0.08333,
RN(7)=0.833973,2*0.083333,
 RJ(7)=.095,
 RK(7)=1.562E-3,
 RL(7)=.882,
 RM(7)=.26,
 RN(7)=.525,
 RJ(7)=.0191232,.424704,.12595,.18893,
 RK(7)=13.4E-4,6.15,7.5,.0751834,.1728,.1728,
 RL(7)=1.59526,.100786,.20736,
 RM(7)=1.65888,.0629319,.04608,
 RN(7)=.397672,.0302956,.0262533,
```

QD030 SAMPLE LISTING (Cont'd)

```
R0=1.,8.,1.,3.,6.,1.,5.847943E+0,
    8.255668E+0,-3.836911E-3,2.744002E+0,9.141338E-3,
    1.482371E+1,1.000005E-2,3.371616E+1,1.491065E-2,
    3.625614E+1,2.070125E-2,4.997287E+1,2.177123E-3,
    4.962004E+1,
                1.429914E-2,6.484203E+1,1.500438E-2,
    1.041760E+2,
    -1.195374E+0,1.068842E+0,-4.101781E+1,1.499989E-2,
    1.043408E+2,1.499671E-2,6.283382E+1,1.500014E-2,
    5.026652E+1,1.500205E-2,3.635507E+1,9.999627E-3,
    3.371723E+1,9.956440E-3,1.667851E+1,
RP=5.,7.,1.,3.,6.,1.,1.949492E+3,
    -4.119582E+1,-1.997668E-2,2.244661E-2,9.564108E-3,
    -1.014387E-2,-5.067620E-1,2.787626E+0,6.396601E-1,
    2.693740E+0,1.007363E-2,3.453623E+1,1.491425E-2,
    3.647673E+1,1.498724E-2,6.201540E+1,-4.815873E-1,
    1.017128E+2,5.032200E-1,1.014321E+2,
    -1.195374E+0,1.068842E+0,-4.101781E+1,1.499989E-2,
    1.043408E+2,1.499671E-2,6.283382E+1,1.500014E-2,
    5.026652E+1,1.500205E-2,3.635507E+1,9.999627E-3,
    3.371723E+1,9.956440E-3,1.667851E+1,
RS=3.,7.,1.,3.,6.,1.,5.658251E+0,
    -2.368039E-1,9.904319E+0,2.419247E-1,-8.522766E-1,
    4.073887E+1,1.105701E-2,3.452912E+1,1.438844E-2,
    4.056765E+1,8.649669E-1,4.010764E+1,1.498306E-2,
    6.188272E+1,-2.720169E-1,8.611615E+1,2.996011E-1,
    8.619704E+1,
    -1.195374E+0,1.068842E+0,-4.101781E+1,1.499989E-2,
    1.043408E+2,1.499671E-2,6.283382E+1,1.500014E-2,
    5.026652E+1,1.500205E-2,3.635507E+1,9.999627E-3,
    3.371723E+1,9.956440E-3,1.667851E+1,
```

QD030 SAMPLE LISTING (Cont'd)

```
RT=5.,6.,1.,3.,6.,1.,5.690258E+0,
    9.617756E+0,1.743145E-1,-1.698473E-1,-2.173811E-2,
    2.213556E-2,1.139867E-2,2.556252E+1,1.109599E-2,
    3.433350E+1,1.457154E-2,4.386076E+1,1.491852E-2,
    5.967876E+1,1.031832E-1,8.327270E+1,-7.366104E-2,
    8.315975E+1,
    -1.195374E+0,1.068842E+0,-4.101781E+1,1.499589E-2,
    1.043408E+2,1.499671E-2,6.283382E+1,1.500014E-2,
    5.026652E+1,1.500205E-2,3.635507E+1,9.999627E-3,
    3.371723E+1,9.956440E-3,1.667851E+1,
RU=1.,8.,1.,3.,6.,1.,5.847943E+0,
    8.255660E+0,-3.836911E-3,2.744002E+0,9.141338E-3,
    1.482371E+1,1.000005E-2,3.371616E+1,1.491065E-2,
    3.625614E+1,2.070125E-2,4.997287E+1,2.177123E-3,
    4.962004E+1,
              1.428914E-2,6.484203E+1,1.500438E-2,
    1.041760E+2,
    -1.195374E+0,1.068842E+0,-4.101781E+1,1.499989E-2,
    1.043408E+2,1.499671E-2,6.283382E+1,1.500014E-2,
    5.026652E+1,1.500205E-2,3.635507E+1,9.999627E-3,
    3.371723E+1,9.956440E-3,1.667851E+1,
RO=0.,0.,0.,0.,2.,0.,1.,
                                          7.587335E-3,
    6.784791E+0,1.495375E-2,1.2484445E+2,
NFLACG=1
$END
ENDRUN
```

APPENDIX A. THE SIMPLEX ALGORITHM

The purpose of this appendix is to illustrate the mechanics of the Simplex Algorithm. This will be done in Section 1 via an example problem. Dantzig [1] is the author of the Simplex Algorithm, which is a method for solving Linear Programming problems in a finite number of steps. Ficken [2] and Wilde and Beightler [6] also contain very good illustrations and definitions of the mechanics of the Simplex Algorithm. Section 2 of this appendix shows the settup of the initial tableau of the Simplex Algorithm as it is mechanized in COEBRA.

Section 1. Simplex Algorithm Example Problem

The problem is to maximize the cost function

$$Y = 3X_1 + 2X_2$$

subject to the following constraints:

Constraint #1: $2X_1 + X_2 \leqslant 8$

Constraint #2: $X_1 + 3X_2 \leqslant 15$

and: $X_1, X_2 \geqslant 0$

The feasible region defined by these constraints is illustrated in Figure A.1.

Step #1

The first step begins with denoting X_1 and X_2 as so-called decision variables. Next, so-called slack variables (X_3 and X_4) are introduced to turn the inequality constraints #1 and #2 into equality constraints.

Constraint #1: $2X_1 + X_2 + X_3 = 8$ (Eq. A.1)

Constraint #2: $X_1 + 3X_2 + X_4 = 15$ (Eq. A.2)

In addition, it is now required that X_1, X_2, X_3 and X_4 all be greater than zero. Next, instead of

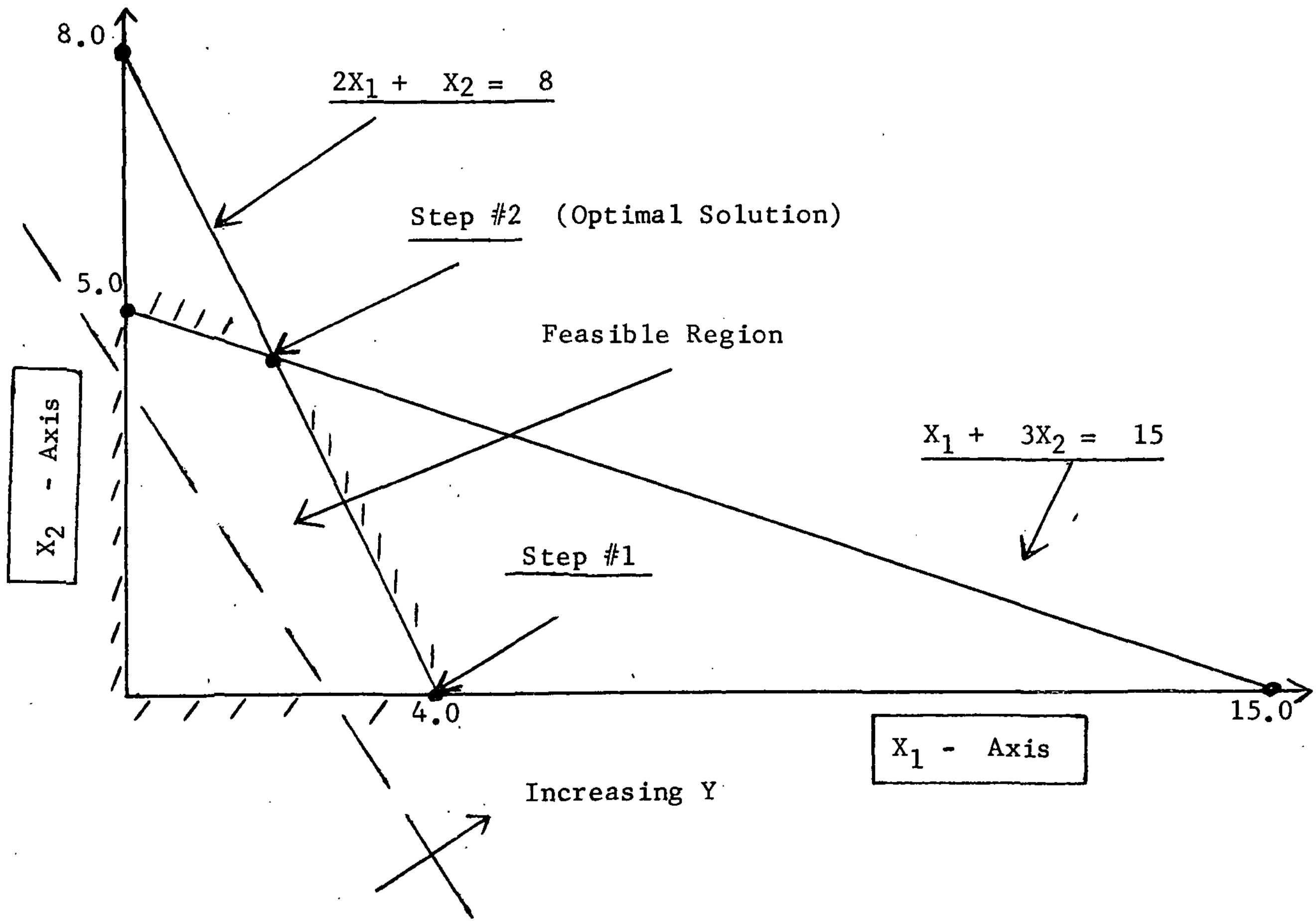


FIGURE A.1 THE EXAMPLE PROBLEM

maximizing Y, change the cost function to one of minimizing Z which will be defined as the negative of Y. Hence,

$$Z = -3X_1 - 2X_2$$

At this point, let us call X_3 and X_4 so-called state variables. Note that Z is not a function of these state variables.

As a first feasible solution, choose $X_1 = X_2 = 0$. From equations A.1 and A.2 (constraints #1 and #2), it is seen that the first value for X_3 is 8, and the first value for X_4 is 15. Also, initially $Y = 0$. Now,

$$\frac{\partial Z}{\partial X_1} = -3$$

which says that Z decreases as X_1 increases. This is desirable. Also,

$$\frac{\partial Z}{\partial X_2} = -2$$

which says that Z decreases as X_2 increases. At each step, the rule is to choose the decision variable that yields the largest payoff. Since X_1 yields the fastest rate of decrease, it will be increased from zero. With X_2 remaining at zero, constraint #1 says that X_1 can be increased to a value of 4 before X_3 is driven to zero. Also, with $X_2 = 0$, constraint #2 says that X_1 can be increased to 15 before X_4 is driven to zero. Since all the variables must remain greater than or equal to zero, we can only increase X_1 to 4. With $X_2 = 0$, and $X_1 = 4$, $Y = 12$, and Step 1 is complete. Step 1 is illustrated in Figure A.1.

Step #2

Since X_3 is now zero, we must "control" it to keep it greater than or equal to zero. Hence, we interchange the role of X_1, and X_3, and now call X_3 a decision variable and X_1, a state variable. Another of the basic rules of the Simplex Algorithm is to keep the cost funtion (Z), a function of only the decision variables. Solving equation A.1 (constraint #1) for X_1 yields:

Constraint #1: $X_1 = 4-0.5X_2 -0.5X_3$ (Eq. A.3)

Substituting X_1 into Z yields:

$$Z = -12 -0.5X_2 + 1.5X_3$$

Substituting X_1 into equation A.2 (constraint #2) yields:

Constraint #2: $2.5X_2 -0.5X_3 + X_4 = 11$ (Eq. A.4)

Summarizing, the decision variables are now:

$$X_2 = 0$$

$$X_3 = 0$$

The state variables are now:

$$X_1 = 4$$

$$X_4 = 11$$

Now:

$$\frac{\partial Z}{\partial X_2} = -0.5$$

$$\frac{\partial Z}{\partial X_3} = 1.5$$

At this point, in order to further reduce Z, we must hold X_3 to zero, and increase X_2.

From equation A.3 (constraint #1) we can increase X_2 to 8 before driving X_1 to zero. From equation A.4 (constraint #2), we can increase X_2 to 4.4 before driving X_4 to zero. Hence, we can increase X_2 only to 4.4. Step 2 is now complete, with $X_2 = 4.4$, $X_4 = 0$, $X_3 = 0$, and $Z = -14.2$. Step 2 is illustrated in Figure A.1.

Step #3

With $X_4 = 0$, we must now interchange the roles of X_4 and X_2 by now calling X_4 a decision variable, and X_2 a state variable. In order to keep Z a function of the decision variables, we must solve equation A.4 (constraint #2) for X_2:

Constraint #2: $X_2 = 4.4 + 0.2X_3 - 0.4X_4$ (Eq. A.5)

Substituting X_2 into Z yields:

$$Z = -14.2 + 1.4X_3 + 0.2X_4$$

Substituting X_2 into equation A.3 (constraint #1) yields:

Constraint #1: $X_1 = 1.8 - 0.6X_3 + 0.2X_4$ (Eq. A.6)

Now:
$$\frac{\partial Z}{\partial X_3} = 1.4$$
$$\frac{\partial Z}{\partial X_4} = 0.2$$

At this point, since X_3 and X_4 are both zero, Z cannot be further reduced, and the optimal solution has been obtained. From equations A.5 and A.6 (constraints #1 and #2), the optimal solution is:

$$X_1 = 1.8$$
$$X_2 = 4.4$$

and $Z = -14.2$

or $Y = 14.2$

Figure A.1 verifies that this is indeed the optimal solution.

Section 2. Initial Tableau of the Simplex Algorithm

The Simplex Algorithm solves the following problem. Find the maximum value of the linear cost function:

$$Y = \sum_{j=1}^{n} a_j X_j$$

subject to a matrix of m linear constraint equations

$$\sum_{j=1}^{n} b_{ij} X_j \begin{pmatrix} \geqslant \\ \leqslant \\ = \end{pmatrix} C_i \qquad (i = 1, \ldots m)$$

and all $X_j \geqslant 0$.

In COEBRA, X_j is the j^{th} stand-alone autopilot variable. Also in COEBRA, the constraint equations are a mixture of " $\leqslant$ " and " $\geqslant$ " type constraints. As shown in Wilde and Beightler [6], the Simplex Algorithm uses slack and artificial variables in addition to the original variables (X_j). Slack variables are used for the " $\leqslant$ " constraints. Slack and artificial variables are used for the " $\geqslant$ " constraints. In the final solution, the artificial variables must have zero value in order for the solution to be feasible.

Table A.1 shows the settup used in COEBRA for the initial tableau of the Simplex Algorithm. The first row is for the cost function coefficients. The next n rows are for the constants (C_i) and the coefficients (b_{ij}) in the " $\leqslant$ " autopilot variable constraint equations. The next K rows are for the " $\leqslant$ " so-called margin constraint equations. Following these, are M rows for the " $\geqslant$ " so-called margin constraint equations. The last n rows are for the " $\geqslant$ " autopilot variable constraint equations. If NPRINT = 1, this tableau will be printed out for each entry into the routine that solves the Simplex Algorithm.

TABLE A.1 INITIAL TABLEAU OF THE SIMPLEX ALGORITHM

	1st Column	n Columns	(n + K) Columns	(n + M) Columns	(n + M) Columns
	Constraint Equation Constants	Autopilot Variables (X_j) for all time points in the order in which they are input	Slack variables for the " $\leqslant$ " constraint equations.	Artificial variables for the " $\geqslant$ " constraint equations	Slack variables for the " $\geqslant$ " constraint equations
1st Row	---	Cost function coefficients (a_j)			
n Rows	(C_i)	Coefficients for the " $\leqslant$ " autopilot variable constraint equations (b_{ij})			
K Rows	(C_i)	Coefficients for the " $\leqslant$ " so-called margin constraint equations (b_{ij})			
M Rows	(C_i)	Coefficients for the " $\geqslant$ " so-called margin constraint equations (b_{ij})			
n Rows	(C_i)	Coefficients for the " $\geqslant$ " autopilot variable constraint equations (b_{ij})			

In the notation used when the tableau is printed out: L = total number of constraint equations

$$= 2n + K + M$$

$$K1 = n + K$$

$$K2 = 0$$

$$K3 = n + M$$

APPENDIX B. BIBLIOGRAPHY

[1.] Dantzig, G. B., "Maximization of a Linear Function of Variables Subject to Linear Inequalities", Activity Analysis of Production and Allocation, New York, N. Y., Wiley, 1951.

[2.] Ficken, F. A., The Simplex Method of Linear Programming, New York, N. Y., Holt, Rinehart and Winston, 1961.

[3.] Greensite, A. L., "Analysis and Design of Space Vehicle Flight Control Systems, Vol. VII, Attitude Control During Launch", NASA CR-826, July 1967.

[4.] Harris, R. D., "Analysis and Design of Space Vehicle Flight Control Systems, Vol. XIV, Load Relief", NASA CR-833, August 1967.

[5.] Hoelker, R. F., "Theory of Artificial Stabilization of Missiles and Space Vehicles With Exposition of Four Control Principles", George C. Marshall Space Flight Center, NASA TN D-555, June 1961.

[6.] Wilde, D. J., and C. S. Beightler, Foundations of Optimization, Englewood Cliffs, New Jersey, Prentice-Hall, 1967.