General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

MCR-75-322
NAS8-30808

Volume I

Technical Manual

August 1975

Computerized Optimization of Elastic Booster Autopilots

(NASA-CR-144026) COMPUTERIZED OPTIMIZATION OF ELASTIC BOOSTER AUTOPILOTS. VOLUME 1: TECHNICAL MANUAL (Martin Marietta Aerospace, Denver, Colo.) 174 p HC $6.75 CSCL 22B

N76-11221

Unclas

G3/18 03077

NOV 1975
RECEIVED
NASA STI FACILITY
INPUT BRANCH

Prepared for:

George C. Marshall Space Flight Center
Huntsville, Alabama

MARTIN MARIETTA

FOREWORD

This report, Volume I of 2 volumes, was jointly prepared by the Guidance and Controls Section and the Scientific Programming Section of Martin Marietta Aerospace, Denver Division, under Contract NAS8-30808. This volume contains the philosophy and the mathematical basis of the non-linear programming algorithm underlying the development of the COEBRA program. Volume II 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 September 1974 to August 1975, 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

CHAPTER		PAGE
1.	THE DESIGN PROBLEM	
	1.1 Introduction	1
	1.2 Analysis in the Frequency Domain	2
	1.3 Analysis in the Time Domain	5
	1.4 The Autopilot Problem	10
	1.5 A Simple Example	14
2.	THE DESIGN ALGORITHMS	
	2.1 Linear Programming	23
	2.2 Linear Programming and the Autopilot Problem	27
	2.3 The Objective Function	29
	2.4 The Load Relief Cost Function	31
	2.5 General Flow Chart	33
	2.6 Selection of Iterative Step Size	35
	2.7 The Inner Loop	37
	2.8 Graphical Illustration of Step-Size Optimization	38 38
	2.9 Convergence to an Exterior Optimum	44
	2.10 Termination	48
	2.11 Overall Flow Chart	49
	2.12 Step-Size Optimization Flow Chart	51
3.	SUMMARY	
	3.1 Summary and Conclusions	54
	3.2 Projected Applications	56

PAGE

BIBLIOGRAPHY 58

APPENDIX A. Computational Aids for Paper and Pencil Design . . . . . .

APPENDIX B. Sample Applications of COEBRA . . . . . .

APPENDIX C. Random Forcing Functions and COEBRA . . .

APPENDIX D. The Input Formats . . . . . .

APPENDIX E. The Simplex Algorithm . . . . . .

CHAPTER 1

THE DESIGN PROBLEM

1.1 Introduction

The detailed design of the autopilot for a large elastic booster is at best a tedious procedure. The function of the autopilot is to insure satisfactory flight from liftoff to orbital entry or touchdown, depending on the specific mission involved. In general, the autopilot works in conjunction with, but independent of, the booster guidance system. It is primarily a stability controller functioning to stabilize the vehicle in order that (1) the vehicle response to guidance commands is not compromised, (2) trajectory dispersions resulting from both internal and external disturbances be minimal, (3) structural integrity of the vehicle be maintained.

Typically, the autopilot designer is provided with a mission profile including nominal trajectory data and vehicle/payload dynamics. A critical element of the trajectory data will include specific flight conditions that must be met at specified points along the trajectory. These points are usually points at which a change in autopilot parameters, filter gains and/or compensation time constants is required, or at which either critical vehicle dynamics are encountere or trajectory maneuvers are scheduled to occur. They are identified in this volume as trajectory events or trajectory time points.

The control conditions to be met at these critical trajectory points can be specified in terms of stability criteria based on an open loop frequency response analysis or a closed loop time response and are often specified as a mixture of both.

The common stability "performance measures" are defined in either the real frequency domain or in the time domain and are derived from the basic properties of the governing differential equations or the transfer functions

(Laplace transforms of the input-output differential equations) for the system under consideration.

1.2 Analysis in the Frequency Domain

Without going into great detail, it is assumed that the reader has covered the ground before, recall that the general solution of a linear differential equation(s) can be written as an exponential function or a linear combination of exponential functions with exponents $\lambda_i t$. The λ_i can be real, complex, or imaginary, depending on the nature of the specific equation(s) being solved. If the real part of any one or more of the λ_i is positive, the system governed by the differential equation is said to be unstable. If the real part of any one or more of the λ_i is identically zero, the system is said to be oscillatory. If the real part of all the λ_i are negative, the system is said to be stable.

The numerical values of the λ_i are derived from the characteristic equation of the system, and are the poles of the system transfer function. We are thus led to a consideration of the location of the system poles and a characterization thereof that will yield the greatest amount of information regarding the overall system behavior. The Nyquist stability criteria provides such a characterization and is typically used in the analysis of space vehicle autopilots.

Let us consider a simple control system as a preliminary. Figure 1-1 exhibits such a system. In this system we relate G_1 to the control elements, G_2 to the controlled system or "plant," H to the feedback elements, and K_1 and K_2 to the forward and feedback path gains. To simplify what follows, we let $G = G_1 G_2$ and the closed loop transfer function becomes

$$\frac{C}{R}(s) = \frac{K_1 G(s)}{1 + K_1 K_2 G(s) H(s)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 1\text{-}1$$

where G(s) and H(s) are the transfer functions relating output to input derived from the Laplace transform of the governing differential equation.

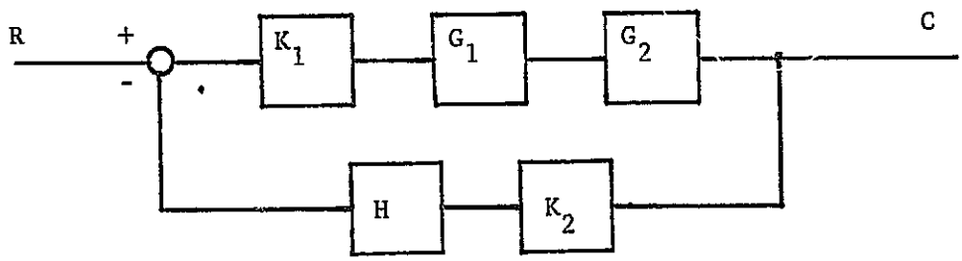


Figure 1-1 Typical Control System

In the application of Nyquist criteria, we treat the complex variable $s = \sigma + j\omega$ as a pure imaginary, or $\sigma = 0$ and examine the characteristic equation

$$1 + K_1 K_2 G(s) H(s) = 0 \quad \ldots \quad 1\text{-}2$$

Rather than treat the real frequency behavior of the closed loop system, we treat the open loop function, $K_1 K_2 G H$, relative to the complex quantity, $-1 = 1 e^{j\pi}$. The Nyquist criteria states that, if in the complex frequency domain, we map

$$F(s) = K_1 K_2 G(s) H(s) \quad \ldots \quad 1\text{-}3$$

as s traverses the contour of Figure 1-2a in the s-plane the mapping of F(s) does not encircle or pass through the -1 point in the F(s) plane, none of the roots of F(s) lie in the right half s-plane or along the imaginary axis of this plane. Thus, in the time domain, the real part of all the λ_i are all less than zero and the system is stable.

Typically, along the contour T_2 of Figure 2a, the quantity F(s) vanishes and we are left only with F(s) mapped as s traverses Γ_3, Γ_4 and Γ_1, usually in that order. We further observe the generally $F(j\omega)$ for positive ω is the mirror

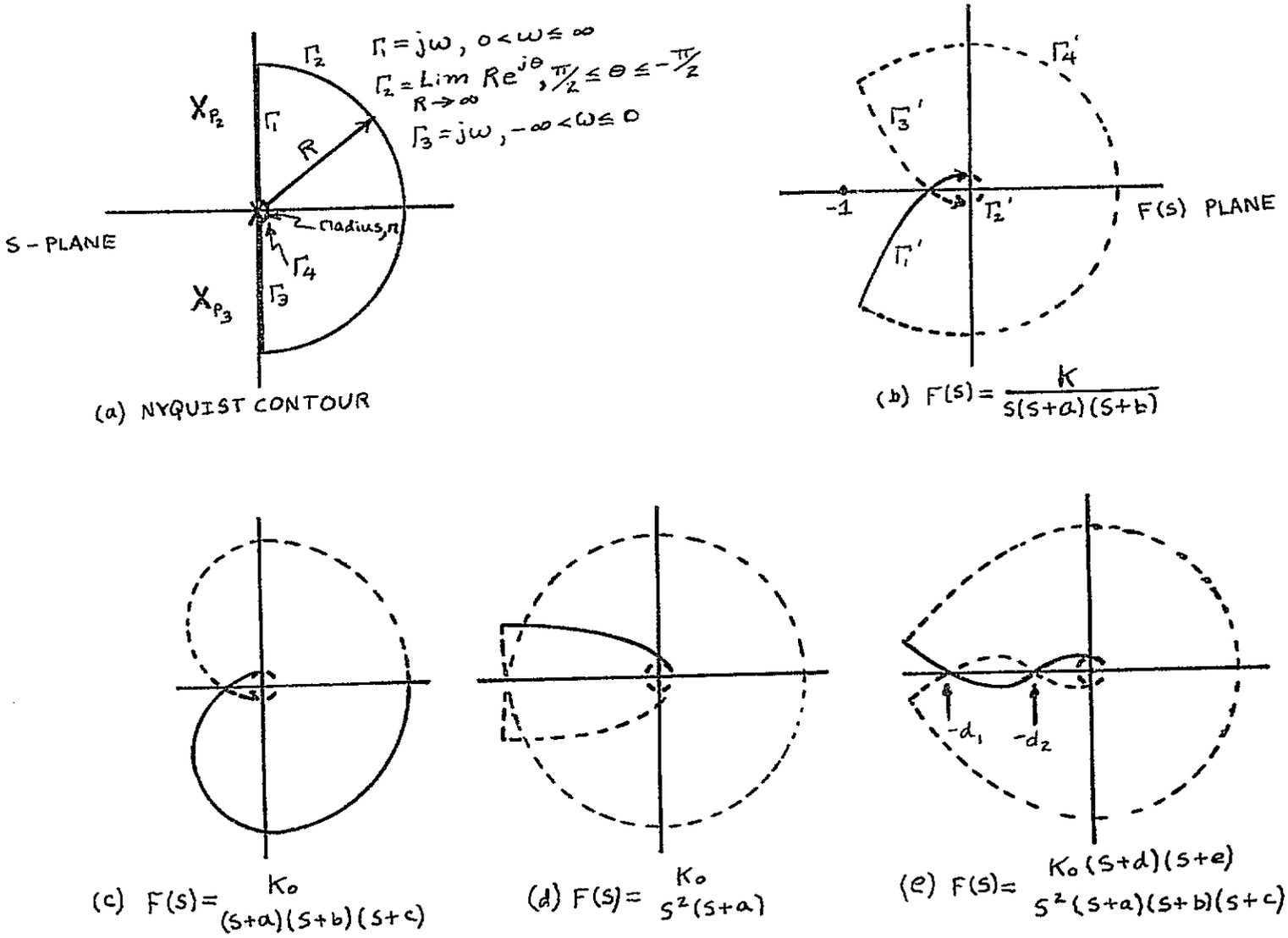


Figure 2-2. The Nyquist Contour and Typical Mappings of F(S) for Simple Systems

image of $F(-j\omega)$ and thus we need only map $F(j\omega)$ for $\omega \geqslant 0$, the real frequency domain. Thus, we can actually observe not only the absolute stability but also the relative stability through examination of the real frequency response.

Consider $F(j\omega) = \mathrm{Re}\ (K_1K_2GH) + j\ I_m\ (K_1K_2GH)$ 1-4

The relative stability can be exhibited by the gain margin,

$$M_K = \left|K_1K_2GH\right|_{\omega_\pi}^{-1} - 1 \quad \ldots\ldots\ldots\ldots \quad 1\text{-}5^*$$

where ω_π is that frequency for which the argument of $F(j\omega)$ or the phase angle of K_1K_2GH is π radians or 180 degrees, and the phase margin

$$M_\Phi = \left|\pi - \arg F(j\omega)\right|_{\omega_c} \text{ radians} \quad \ldots\ldots\ldots\ldots \quad 1\text{-}6a$$

$$= \left|180 - \arg F(j\omega)\right|_{\omega_c} \text{ degrees} \quad \ldots\ldots\ldots\ldots \quad 1\text{-}6b$$

where ω_c, the crossover frequency, is that frequency at which $\left|F(j\omega)\right| = 1$. Additional useful information regarding equivalent damping and the location of the least stable poles of $F(j\omega)$ are discussed in the open literature.[45]** Figure 1-3 illustrates these margins in the conventional manner.

The advantage of the use of the Nyquist diagram in the real frequency domain in the design of elastic booster autopilots lies in the fact that the rigid body, aerodynamic and flexible body bending mode frequency responses are relatively well separated and thus lend themselves to definitions of separate gain and phase margins, as shown in Figure 1-4, an unfolded Nyquist diagram with frequency as a running parameter and magnitude and phase as coordinates.

---

*Gain margin is defined as the gain increase required to drive the system into *instability or $K\ \left|F(j\omega_\pi)\right| = 1$ and $K' = 1/\left|F(j\omega)\right|$.
Thus, in a vector sense, the margin is given by 1-5.

**Figures in brackets refer to references in the Bibliography.

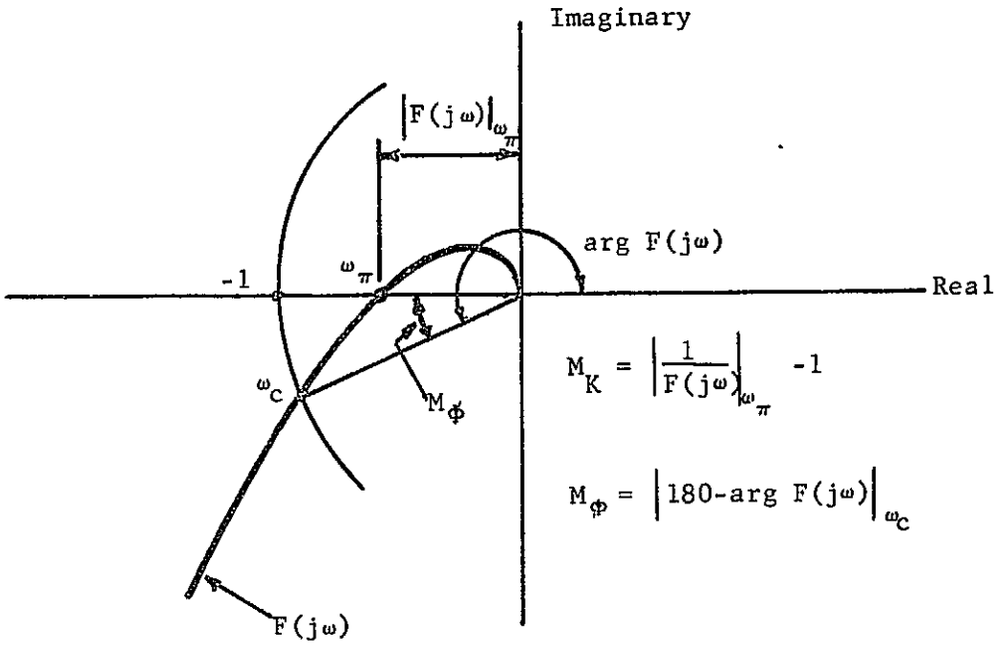
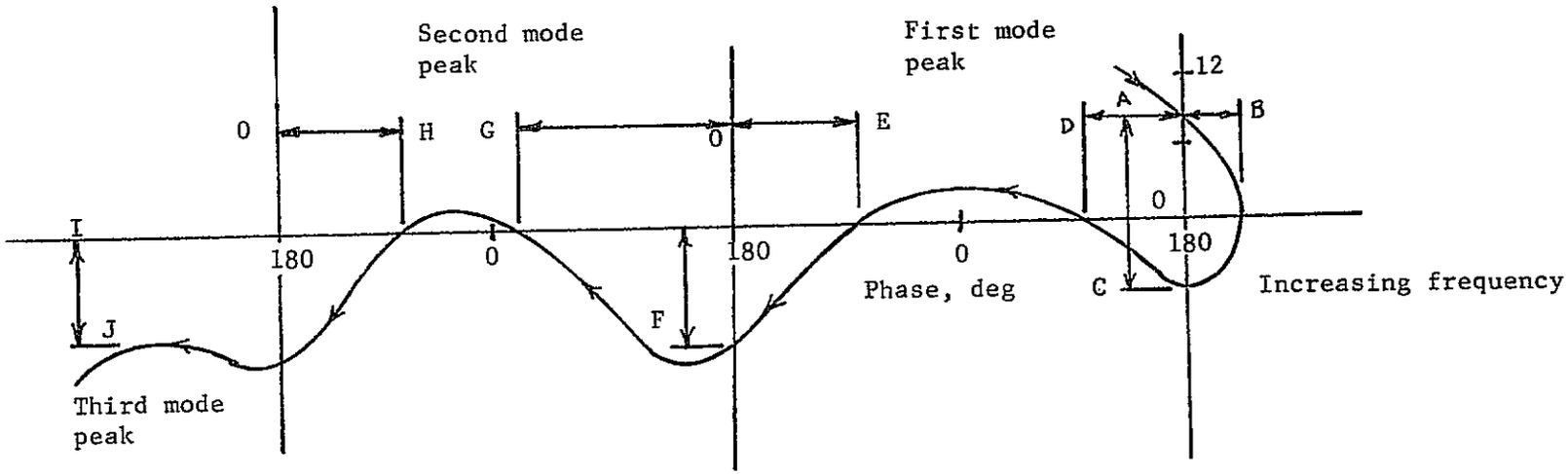


Figure 1-3 Definition of M_K and M_Φ in terms of Nyquist plot

1.3 Analysis in the Time Domain

In the time domain the performance measures are usually defined in terms of the system response to a step function input. The generally accepted performance or stability measures, based on a closed loop system, are the rise time, T_r, settling time, T_s, time to peak value, T_p, time delay, T_d, percent overshoot and steady-state error. The rise time is commonly defined as the time required for the system to go from 0.1 to 0.9 of the final value. The time to peak, T_p, is the time to reach the first peak of overshoot and represents one half cycle of the frequency of the damped oscillation typical of systems. The settling time, T_s, represents the longest exponential decay of the system and is defined as the time required for the system to settle within a band 0.95 to 1.05 of the desired final value. The delay time, T_d, is the time required for the system to reach 0.5 of the final value. The overshoot is the amount the first half cycle of the damped oscillation exceeds the desired final value and the steady state error is the difference between the final value of the system



Legend:

OA	Aerodynamic gain margin	OF	Gain margin between first and second bending modes
OB	Rigid body phase margin	OG	Phase margin second structural bending mode (front side)
OC	Rigid body gain margin	OH	Phase margin second structural bending mode (back side)
OD	Phase margin front side of first bending mode	IJ	Gain margin third and higher structural bending mode peaks
OE	Phase margin back side of first bending mode		

FIGURE 1-4 Open Loop Gain/Phase Plot - Unfolded Nyquist Diagram in Rectangular or Nichols Chart Coordinates

response and the desired value (typically 1). Figure 1-5 illustrates these parameters.

It is of interest to note that many high order systems can be approximated by an equivalent second order system. In any case, the time response is governed by what are known as dominant poles. To better appreciate what is meant by a dominant pole, consider a closed loop system function of the general form

$$C(s) = \frac{K \prod_{i=1}^{\ell} (s^2 + 2\xi_i \omega_i s + \omega_i^2) \prod_{j=1}^{m} (s + \tau_j)}{s^n \prod_{K=1}^{p} (s^2 + 2\xi_K \omega_K s + \omega_K^2) \prod_{q=1}^{r} (s + \tau_q)} \qquad 1\text{-}7$$

where $2\ell + m \leqslant n + 2p + r$, or simply there is an excess of poles over zeros. This enables us to form a partial fraction expansion of C(s) in elementary terms of the form

$$C(s) = \sum_{i=1}^{n} \frac{\alpha_i}{s^i} + \sum_{K=1}^{p} \frac{(\alpha_K s + \beta_K)}{s^2 + 2\xi_K \omega_K s + \omega_K^2} + \sum_{q=1}^{r} \frac{\gamma_q}{s + \tau_q} \qquad 1\text{-}8$$

This, in turns, inverts to:

$$c(t) = \sum_{i=1}^{n} \alpha_i t^i \mu(t) + \sum_{K=1}^{p} \frac{\left[\alpha_K^2 (1 + \xi_K^2 \omega_K^2) + \beta_K (\beta_K - 2\alpha_K \xi_K \omega_K)\right]^{1/2}}{\omega_K^2 (1 - \xi_K^2)} e^{-\xi_K \omega_K t}$$

$$\cdot \cos\left(\omega_K \sqrt{1 - \xi_K^2}\, t - \tan^{-1} \frac{\beta_K - \alpha_K \xi_K \omega_K}{\alpha_K \omega_K \sqrt{1 - \xi_K^2}}\right)$$

$$+ \sum_{q=1}^{r} \gamma_q e^{-\gamma_q t}$$

1-9

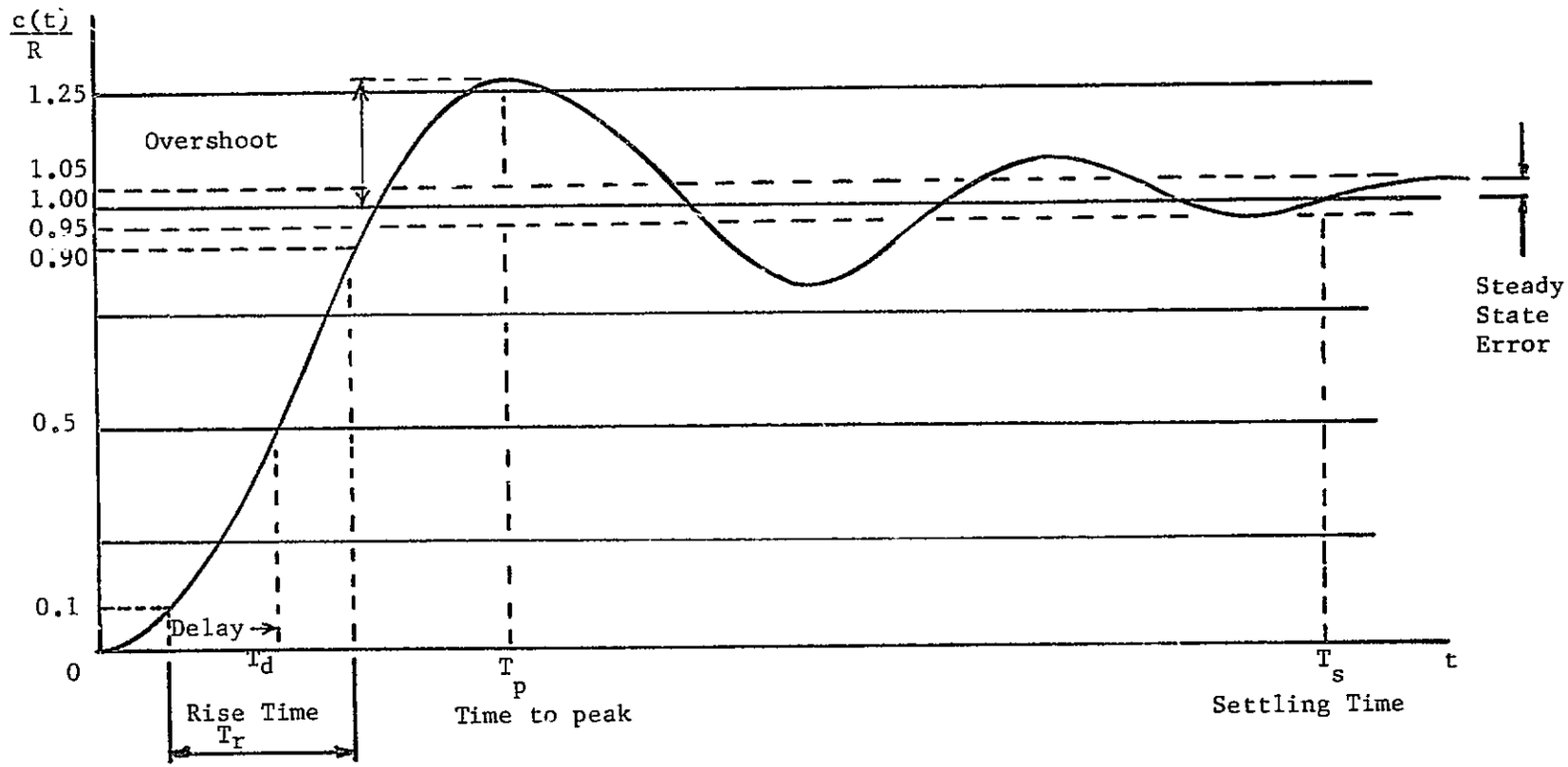


Figure 1-5 Transient Response to Unit Step Function

Observe now the coefficients of t in both the exponential and cosine terms. These are derived from or are identical with the system poles with a step function, $\mathcal{U}$ (t), input. Those terms of equation 1-9 that give rise to the significant portions of c(t) stem from the dominant poles. That is, those terms dominate the performance of the system in the time domain. Their locations are clearly functions of the system parameters and are closely related to the gain margin, phase margin, rise time, settling time, etc.[46] The importance of these poles is discussed further in Appendix B in conjunction with examples.

1.4 The Autopilot Problem

The foregoing indicates that the design of a control system is a relatively straight-forward engineering problem. However, when the numerous inputs and/or disturbances, multiple loops and/or multiple control effectors, or a combination of these are included, as in a typical flight vehicle autopilot, the problem becomes both increasingly difficult and tedious, with results somewhat less than immediately visible. A typical autopilot may have as many as 6 distinct feedback paths with up to 50 constants that must be established in a reliable and economical manner. A mechanization of the iterative design procedure becomes highly advantageous.

Consider a typical autopilot that is to be both gain and phase stabilized. The vehicle flight plan is designed to minimize aerodynamic loading. The vehicle is launched vertically to eliminate or minimize (lateral stabilizing load requirements) on the autopilot. Shortly after lift-off the vehicle is rotated (pitched over) to assume a zero-lift (zero angle of attack) flight condition. The pitch rate is then maintained at the flight path turn rate to obtain a zero angle of attack.

The autopilot designer is supplied with this nominal trajectory as well as nominal vehicle dynamics including rigid body, bending mode, and fuel slosh

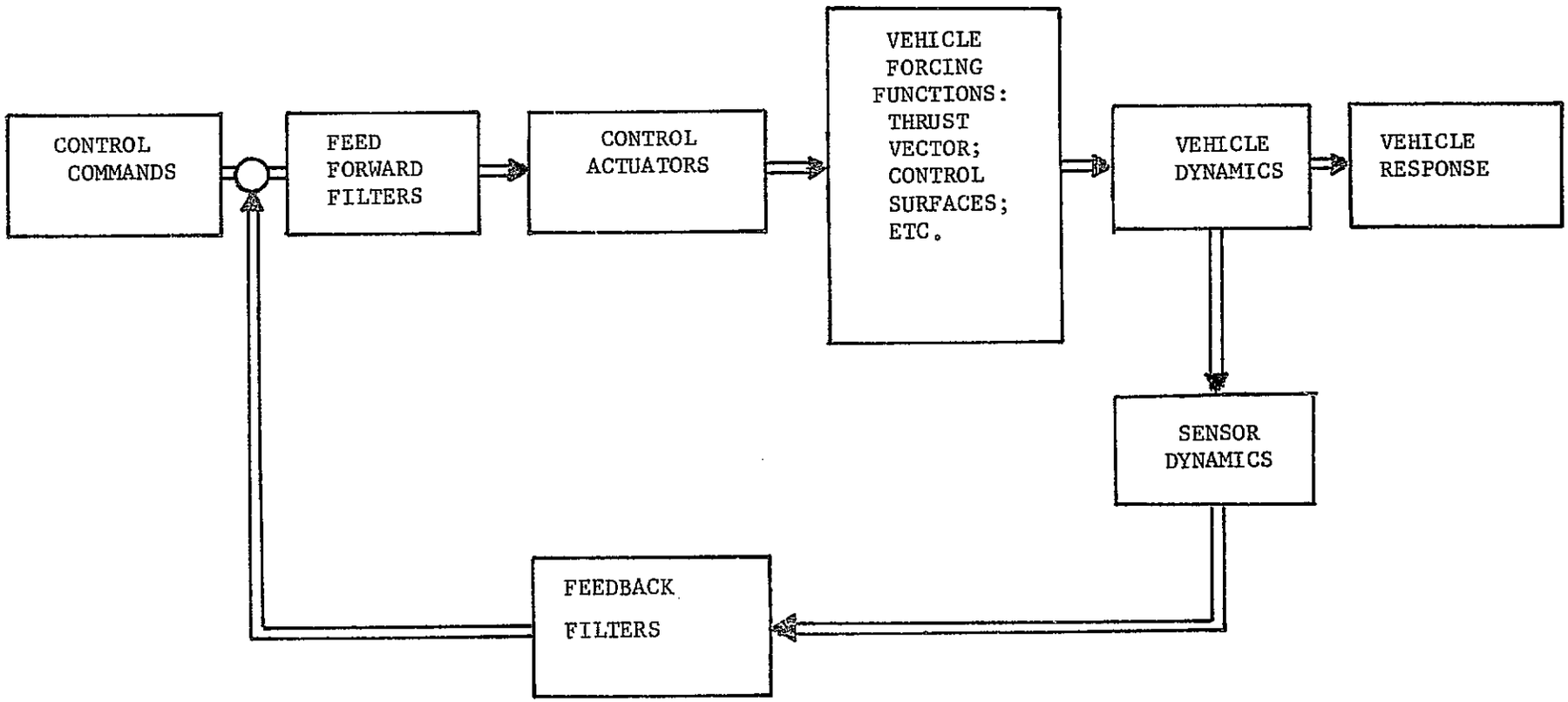


Figure 1-6 Typical Autopilot Configuration

characteristics. Assuming that a standard autopilot configuration has already been established, the usual case today, the first step is to initialize the parameters used within this standard configuration. The initial gains for compensation are derived from the thrust to inertia plot for the specific vehicle, based on a gain margin of about 6 db. The initial time constants are derived from the structural bending mode frequencies. Figure 1-6 illustrates the commonly accepted configuration used. The nominal trajectory and vehicle dynamics (mass and inertia properties) are generally available as the results of a computer run and thus can be provided as a deck of punched cards, magnetic tape or other computer compatible format. Using these data, and the standard configuration with the initial gain and time constants, the frequency response is plotted. A standard computer routine based on a block diagram input format according to the block diagram of Figure 1-7 is used where each block represents an elementary transfer function of the general form

$$T(s) = \frac{s^m \prod_{i=1}^{l} (s+\tau_i) \prod_{k=1}^{j} (s^2 + 2\xi_k \omega_k s + \omega_k^2)}{s^n \prod_{p=1}^{q} (s+\tau_q) \prod_{\mu=1}^{v} (s^2 + 2\xi_\mu \omega_\mu s + \omega_\mu^2)} \qquad 1-10$$

It is to be noted here that T(s) of equation 1-10 is identically the form of G(s) H(s) of the total open loop function of equation 1-3 where for the elementary blocks, m, l, j, n, q and v are of lower magnitude than for the total open loop function. The resulting frequency response is typical of that of Figure 1-4. The response curve is examined for gain and phase margins. The designer then selects new gain constants and/or time constants and repeats the computer run until a satisfactory response satisfying all gain and phase margin constraints is obtained.

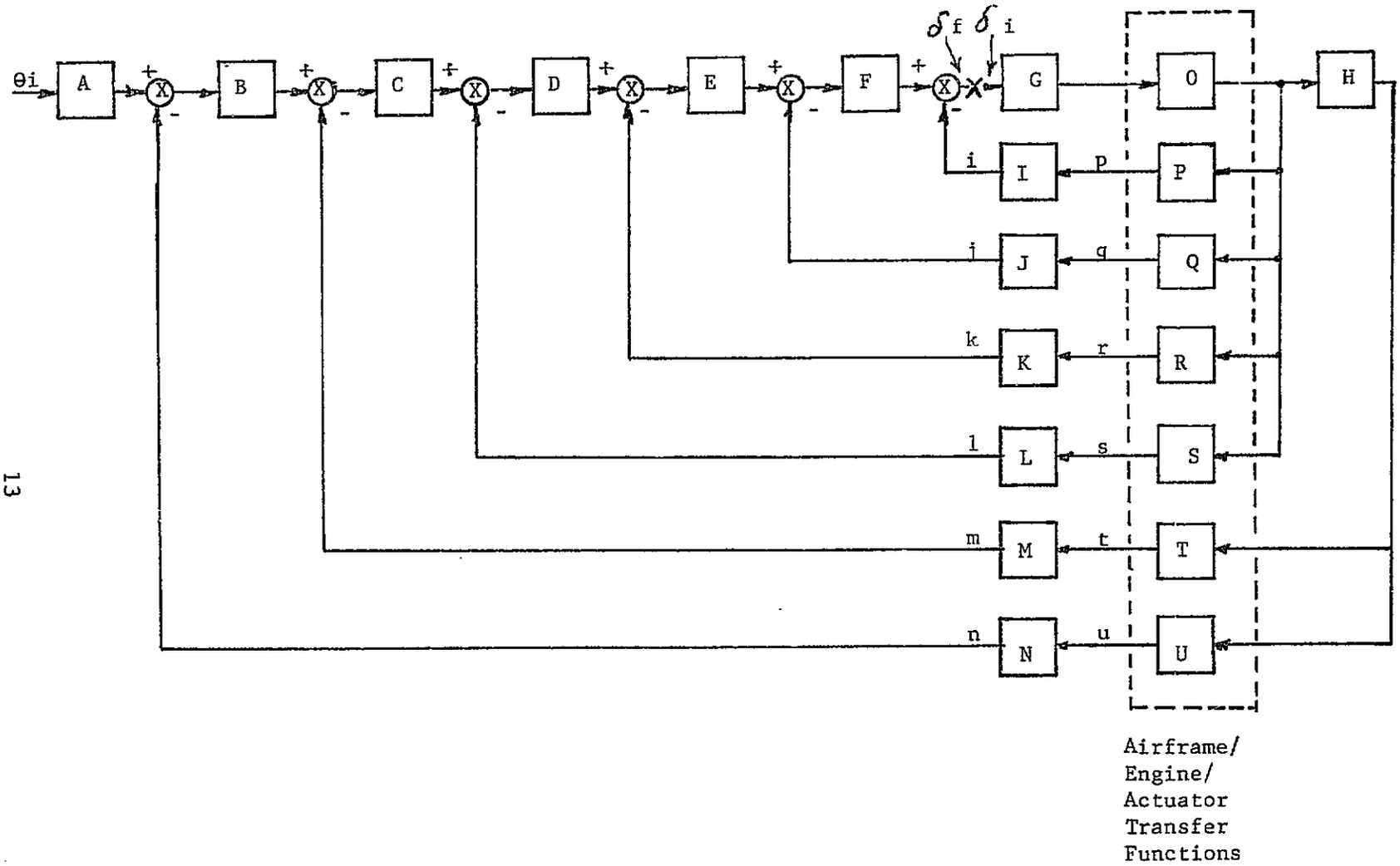


Figure 1-7 Autopilot Configuration for Frequency Response Computation

1.5 A Simple Example

As an example, we treat the system of Figure 1-8 with a specification that the velocity constant, K_v, be 100, the crossover frequency, $\omega_c \geqslant 10$ rad/sec, the gain margin, $M_k \geqslant 10$ db and the phase margin, $M_\Phi \geqslant 45^\circ$. A further constraint is that the system be a unity feedback configuration.

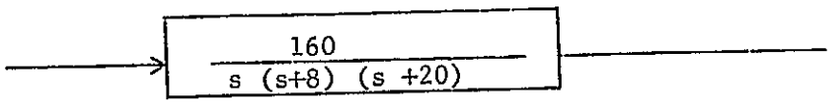


Figure 1-8 Plant of Example

Recall that the velocity constant, K_v, is defined as the ratio of input rate, R', to the steady state error, E, and it can be shown that

$$K_v = \lim_{s \to 0} s \frac{C(s)}{E(s)} = \lim_{s \to 0} s\, G(s)$$

Plotting the conventional Bode diagram and the commonly used gain-phase diagram as in Figures 1-9 and 1-10, we observe that the 3rd order plant, in a unity feedback configuration is unstable. The gain crossover frequency, ω_c, is about 23 radians per second, at which frequency the gain, K, is 1 and the phase is approximately 210 degrees. The Nyquist diagram, of which Figure 1-10 represents a small segment, has the general shape shown in Figure 1-11.

The system is clearly unstable and series compensation in the forward loop is indicated. After some consideration, a lag-lead compensation network is arrived at, resulting in the following configuration (Figure 1-12).

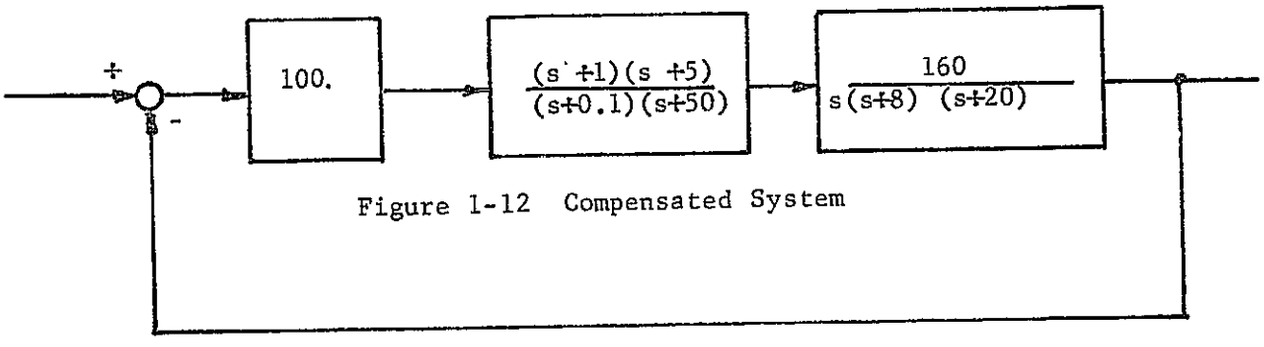


Figure 1-12 Compensated System

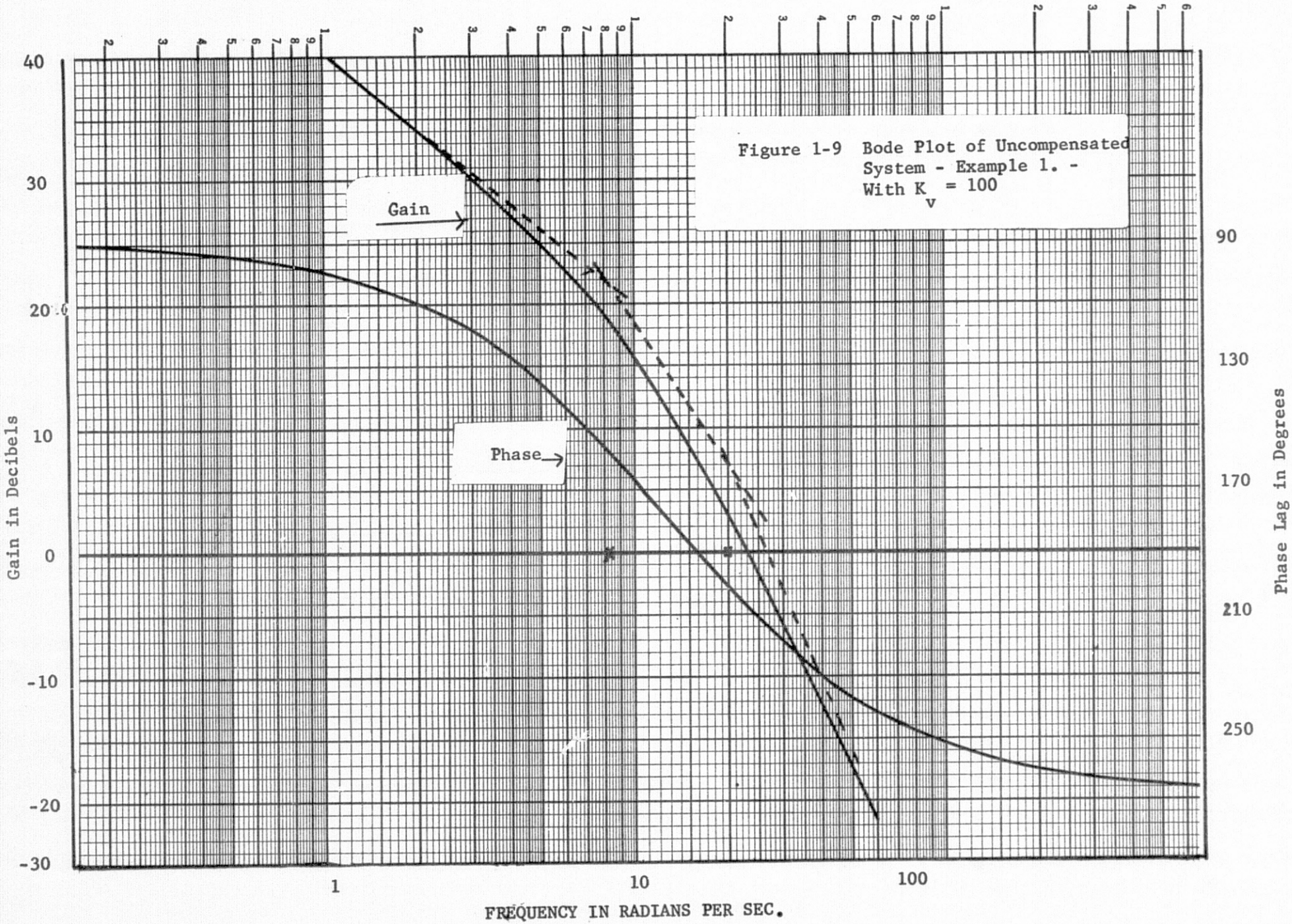


Figure 1-9 Bode Plot of Uncompensated System - Example 1. - With $K_v = 100$

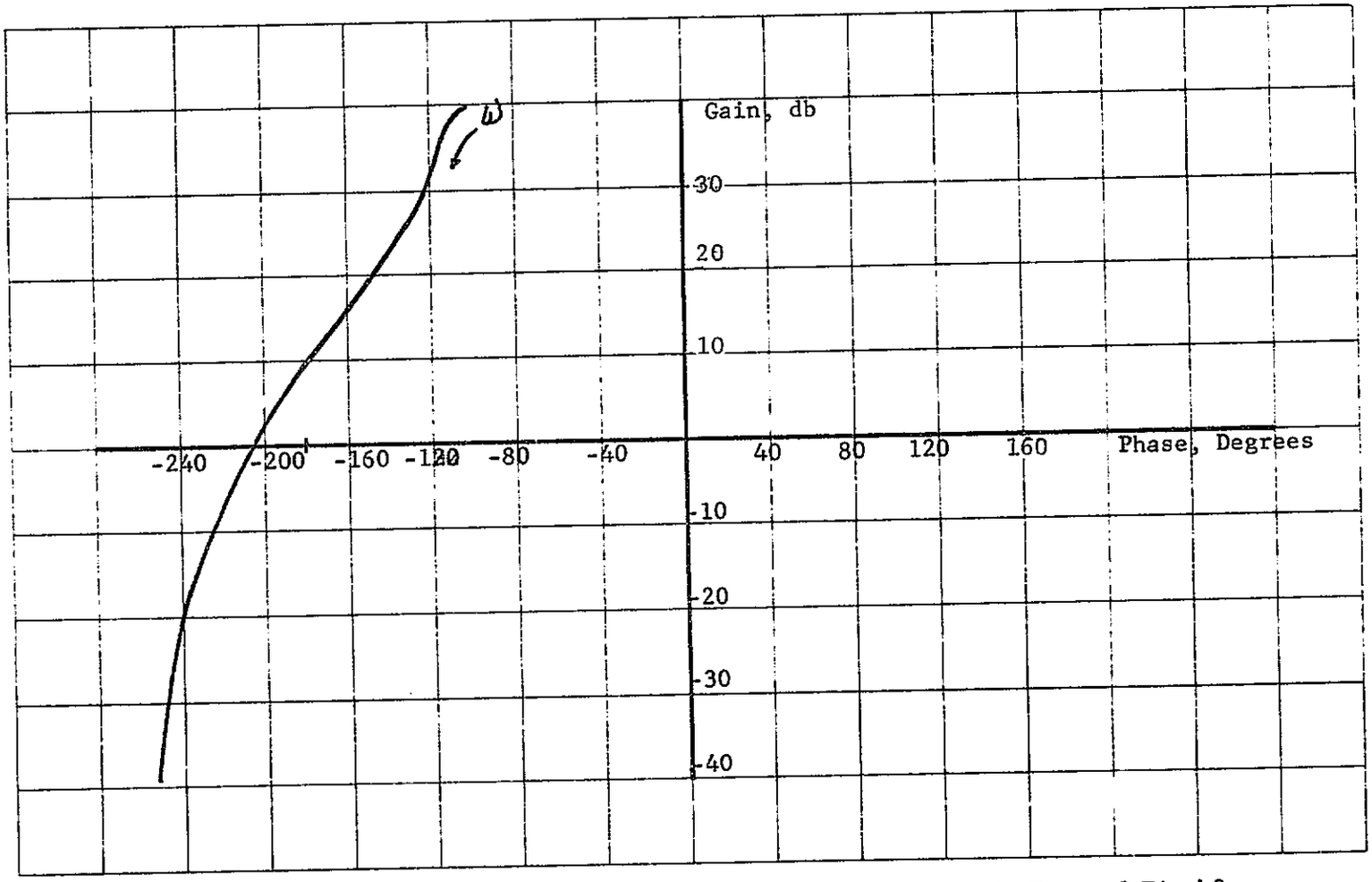


Figure 1-10 Gain-Phase Plot as a Function of Frequency for Bode Plot of Fig. 1-9

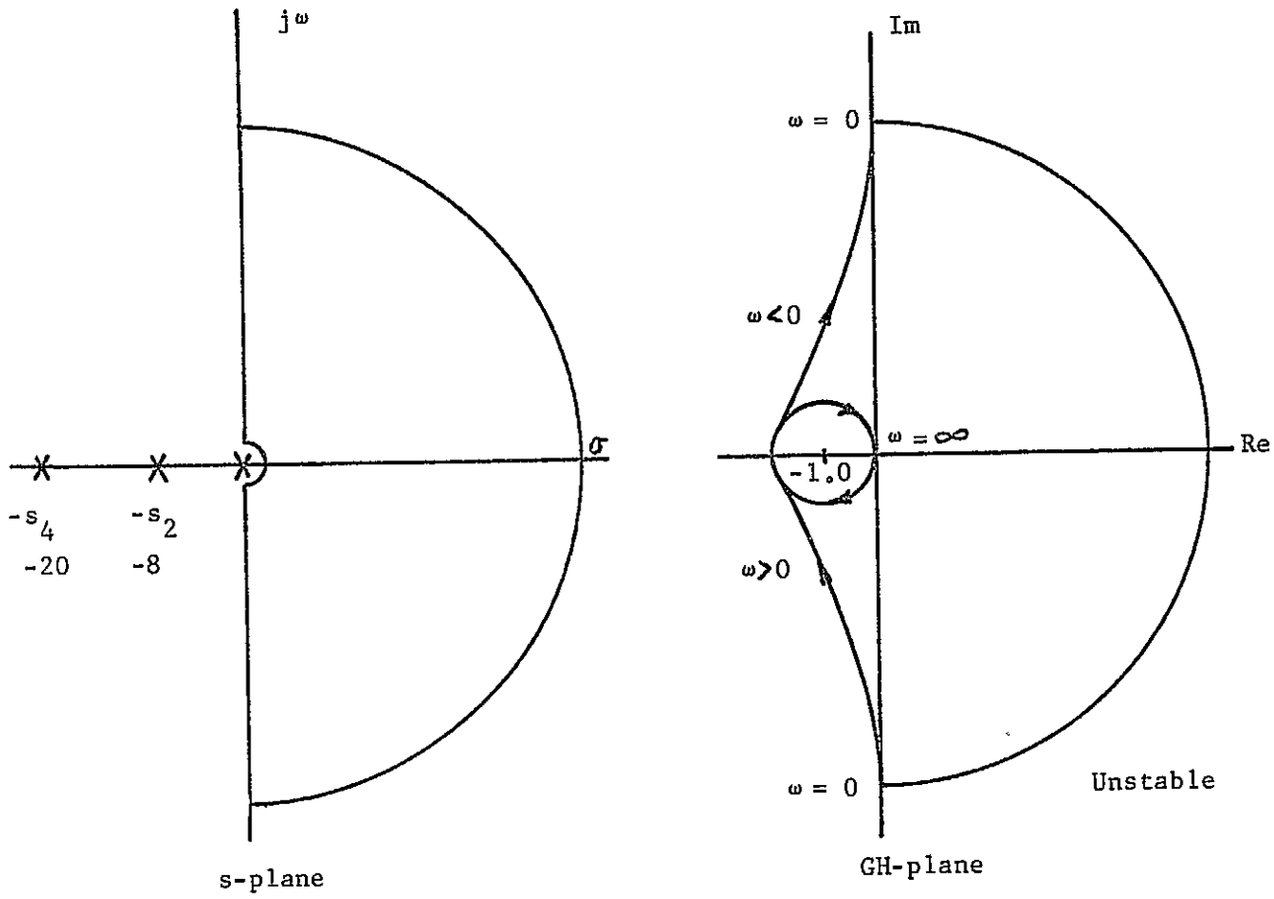


Figure 1-11 Nyquist Plot

Table 1-1 Computation Worksheet - Uncompensated System

ω	$\omega/8$	$<\omega/8$ Degrees	$\omega/20$	$<\omega/20$ Degrees	$<\omega$	ϕ_T Degrees	$\lvert F(j\omega)\rvert$ db
1	0.125	7.15	0.05	2.9	90	100	40
1.5	0.188	9.95	0.075	4.3	90	104	37
2	0.25	14	0.1	5.7	90	110	33.5
4	0.5	26.6	0.2	11.3	90	128	27
8	1	45	0.4	21.8	90	157	17
10	1.25	51.35	0.5	26.6	90	168	15
15	1.875	61.3	0.75	36.9	90	188	8
20	2.5	68.2	1.0	45	90	203	·2
25	3.13	72.3	1.25	51.35	90	214	-3
30	3.75	74.7	1.5	56.3	90	221	-7
40	5.0	78.7	2.0	63.5	90	232	-13
80	10.0	84.29	4.0	76.0	90	250	-30
100	12.5	84.4	5.0	78.7	90	253	-40

Note: Standard paper and pencil computational aids of Appendix 1 used freely.

Figures 1-13 and 1-14 show the Bode and Gain-Phase (Nyquist) plots for the compensated system. The phase margin is about 54° and the gain margin is about 15 db. Clearly, the advantage of a computerized solution to this simple system is doubtful. However, the number of parameters, feedback loops and variables in the autopilot problem evidences this advantage to a marked degree. Recall also that only the starting point and a usable end point were demonstrated for this simple example. Furthermore, we observe that the solution is not optimal in that the gain margin exceeds requirements by about 5 db and the phase margin is about 9° more than required.

We are thus led to develop a computer program that will not only produce the system frequency response in an efficient manner but also will enable an automatic search for near optimum gains and time constants based on a gain margin and/or phase margin constraint or alternatively, the time response and its related performance measures.

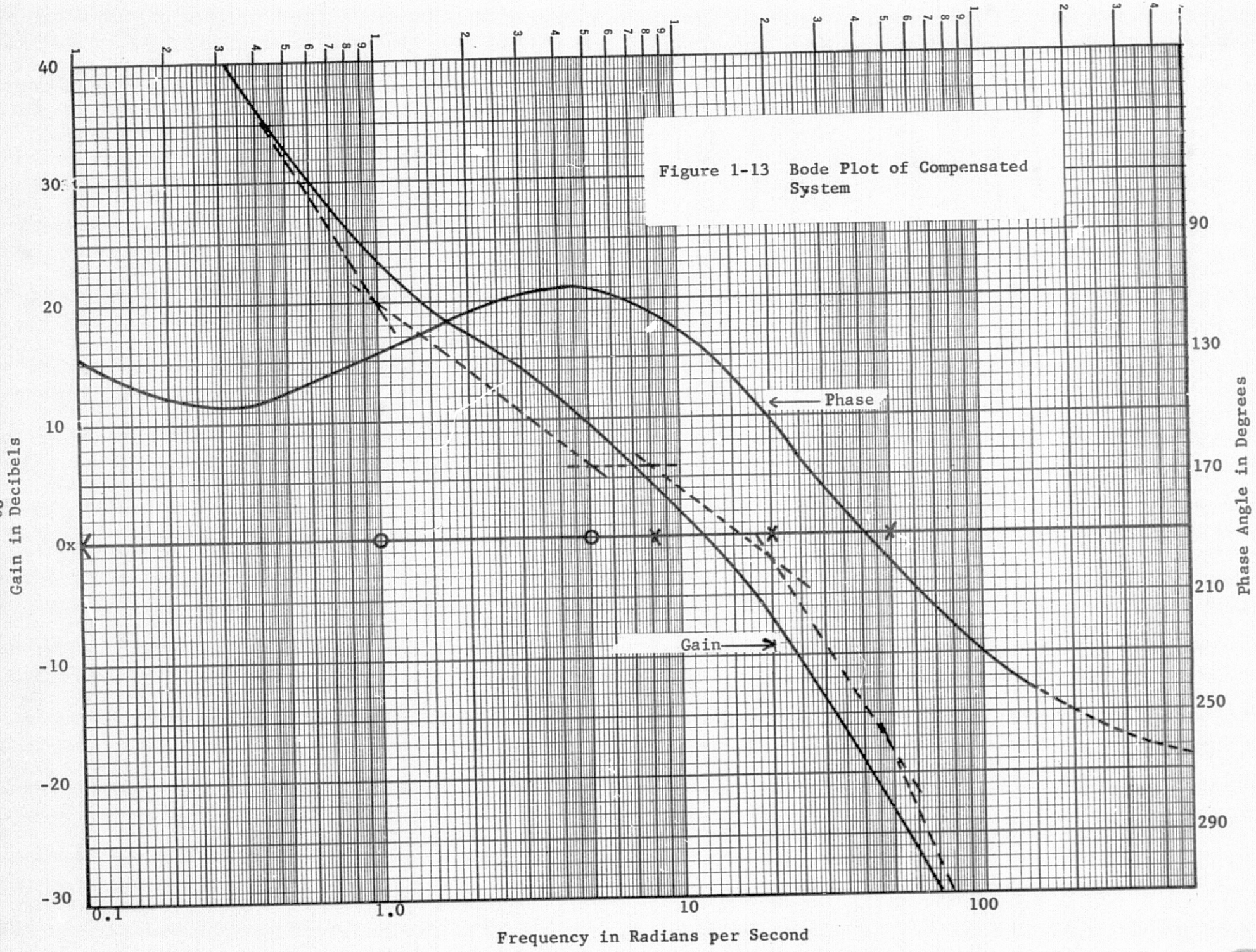


Figure 1-13 Bode Plot of Compensated System

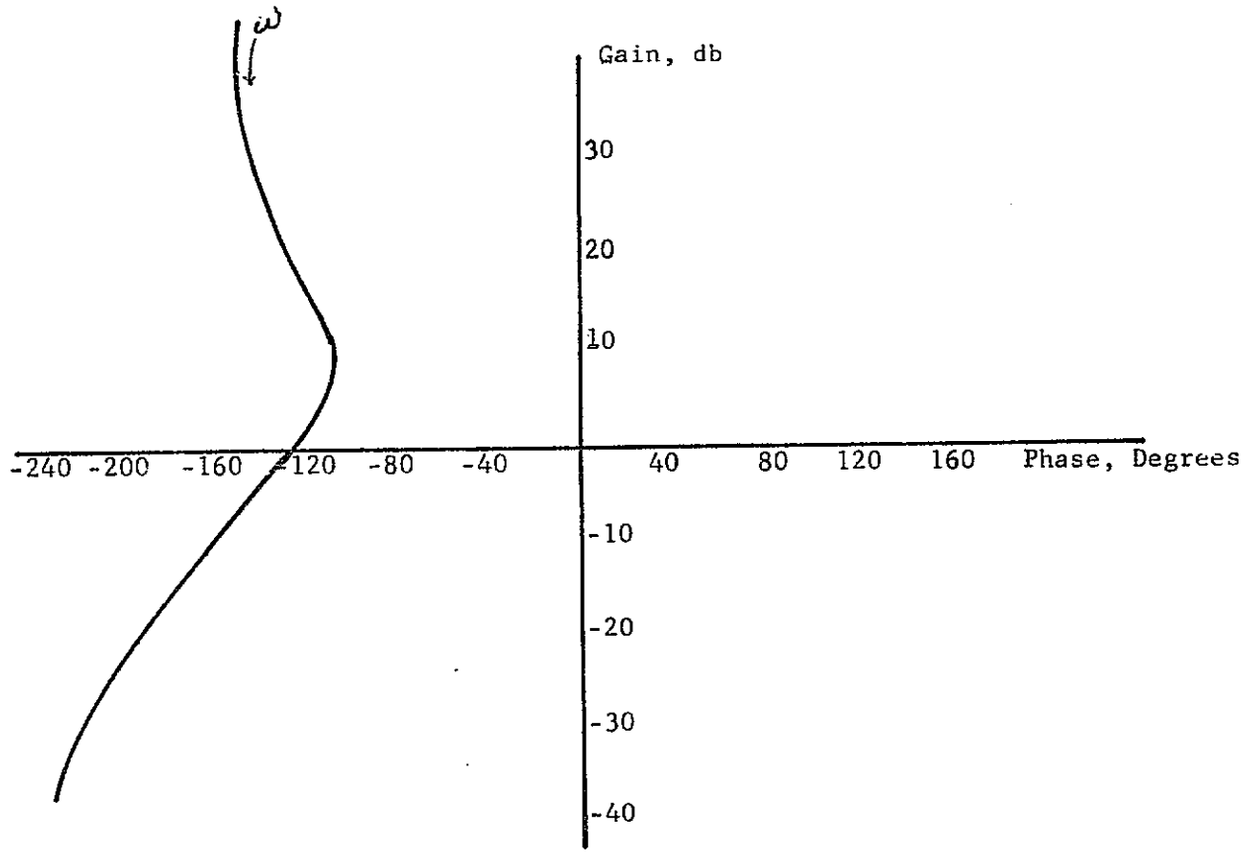


Figure 1-14 Gain-Phase Plot as a Function of Frequency for Bode Plot of Fig. 1-13

Table II - Computation Worksheet - Compensated System

ω	$\omega/0.1$	θ_1 deg	$\omega/8$	θ_2 deg	$\omega/20$	θ_3 deg	$\omega/50$	θ_4 deg	$\omega/1$	θ_5 deg	$\omega/5$	θ_6 deg	θ_T deg	$\lvert F(j\omega)\rvert$ db
0.1	1	45	0.0125	0.7	0.005	--	0.002	--	0.1	5.7	0.02	1.2	128	60
0.2	2	63.5	0.025	1.4	0.01	0.6	0.004	--	0.2	11.3	0.04	2.4	142	48.5
0.4	4	76	0.05	2.9	0.02	1.2	0.008	0.5	0.4	21.8	0.08	4.4	144	36.5
0.6	6	80.6	0.075	4.3	0.03	1.8	0.012	0.7	0.6	31	0.12	6.9	140	30
0.8	8	82.9	0.1	5.7	0.04	2.4	0.016	0.9	0.8	38.7	0.16	9.1	134	26
1	10	84.3	0.125	7.2	0.05	2.9	0.02	1.2	1	45	0.20	11.3	129	23.5
2	20	87.2	0.25	14	0.1	5.7	0.04	2.4	2	63.5	0.4	21.8	114	17
4	40	88.6	0.5	26.6	0.2	11.3	0.08	4.4	4	76	0.8	38.7	106	11.5
8	80	89.0	1.0	45	0.4	21.8	0.16	8.8	8	82.9	1.6	58.0	114	6
10	100	89.4	1.25	51.4	0.5	26.6	0.20	11.3	10	84.3	2.0	63.5	121	2
20	200	90	2.5	68.2	1.0	45	0.4	21.8	20	87.2	4.0	76.	152	-7
40	400	90	5	78.7	2.0	63.5	0.8	38.7	40	88.6	8.0	82.9	189	-19
80	800	90	10	84	4.0	76	1.6	58.0	80	89.2	16.0	86.4	223	-32
100	1000	90	12.5	84.5	5.0	78.7	2.0	63.5	100	89.4	20.0	87.1	230	-37

CHAPTER 2

THE DESIGN ALGORITHMS

2.1 Linear Programming

With gain and phase margins, or any other performance measure, as the autopilot design criteria, the selection of filter time constants and gain factors can be realized through the use of the linear programming technique. With this thought in mind, we examine briefly the linear programming approach before we apply it directly to the autopilot problem.

We treat the linear programming technique in two dimensions in order to clarify the approach. Linear programming addresses the solution of a set of m linear equations in n unknowns subject to constraints on the unknowns and a minimizing functional. Consider

$$\underline{\underline{A}}\ \underline{X} = \underline{B} \qquad \text{2-1}$$

where $\underline{\underline{A}}$ is an m x n coefficient matrix, $\underline{X}$ an n vector of unknowns and $\underline{B}$ is an n vector of constants. We impose the following constraints:

$$x_j \geqslant 0\ , \quad j = 1, 2, \ldots, n \qquad \text{2-2}$$

and

$$\sum_{j=1}^{n} c_j\, x_j = \text{minimum} \qquad \text{2-3}$$

The details of the linear programming method are set forth in the open literature and will not be treated fully here; rather, an example will serve our purposes. Given the set of inequalities:

$$\left.\begin{aligned} -x_1 + 3x_2 &\leqslant 10 \\ x_1 + x_2 &\leqslant 6 \\ x_1 - x_2 &\geqslant 2 \\ x_1 + 3x_2 &\geqslant 6 \end{aligned}\right\} \qquad \text{2-4}$$

with x_1 and x_2 non-negative, minimize the function $-x_1 - 2x_2$. First we alter our minimal function constraint to maximize the function $x_1 + 2x_2$. The first step is to bound the required solution. This is done by changing all the inequalities to equalities and plotting each equation of the set 2-4, recognizing that the equalities represent either upper or lower bounds on the inequalities. The region below the lines $-x_1 + 3x_2 = 10$ and $x_1 + x_2 = 6$ contain admissable values of x_1 & x_2. The entire first quadrant satisfies the requirement that x_1 and x_2 be non-negative. The region above the lines $x_1 - x_2 = 2$ and $x_1 + 3x_2 = 6$ may contain admissable values. Thus, the region bounded by these lines contains an admissable value of both x_1 and x_2 as indicated in Figure 2-1. Overplotting the function to be maximized we write

$$x_1 + 2x_2 = \text{Constant} \quad \ldots \quad 2\text{-}5$$

and plot a series of lines overlaying the region of admissable solution to obtain the admissable solution to the set 2-4 that maximizes the function $x_1 + 2x_2$. Inspection of figure 2-2 indicates that the macimum admissable value of $x_1 + 2x_2$ is 10, from which we see the desired solution to the set 2-4 is $x_1 = 2$, $x_2 = 4$.

This simple two-dimensional (two variable) problem is almost trivial. However, as the dimension n increases, the simple 2-space geometry can no longer be used. The concept, for an n-dimensional space, is still the same. Using a notational scheme that is not dimensionally limited and compacts the statement of the linear programming problem, we write: Determine the column vector $\underline{X}$, a solution of

$$\underline{A}\,\underline{X} = \underline{B} \quad \ldots \quad 2\text{-}5$$

which maximizes

$$Y = \underline{C}^T\,\underline{X} \quad \ldots \quad 2\text{-}6$$

where $\underline{X} \geqslant 0 \quad \ldots \quad 2\text{-}7$

We denote 2-6, where $\underline{C}^T$ is the transpose of the column vector $\underline{C}$, the objective function and 2-5 the constraints. The non-negative property, 2-7, is fundamental to the linear programming algorithm.

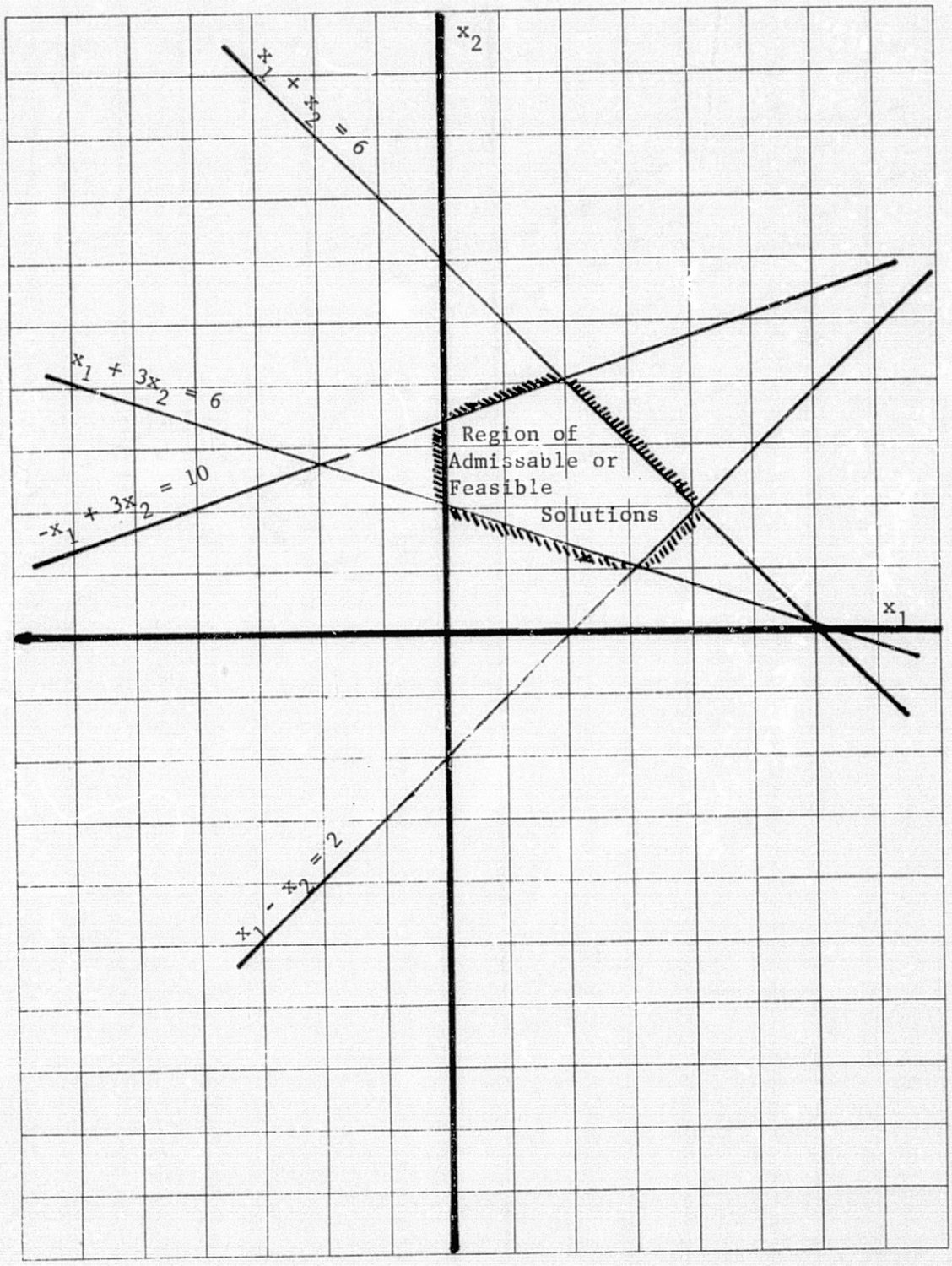


Figure 2-1 Geometry of Linear Programming Example

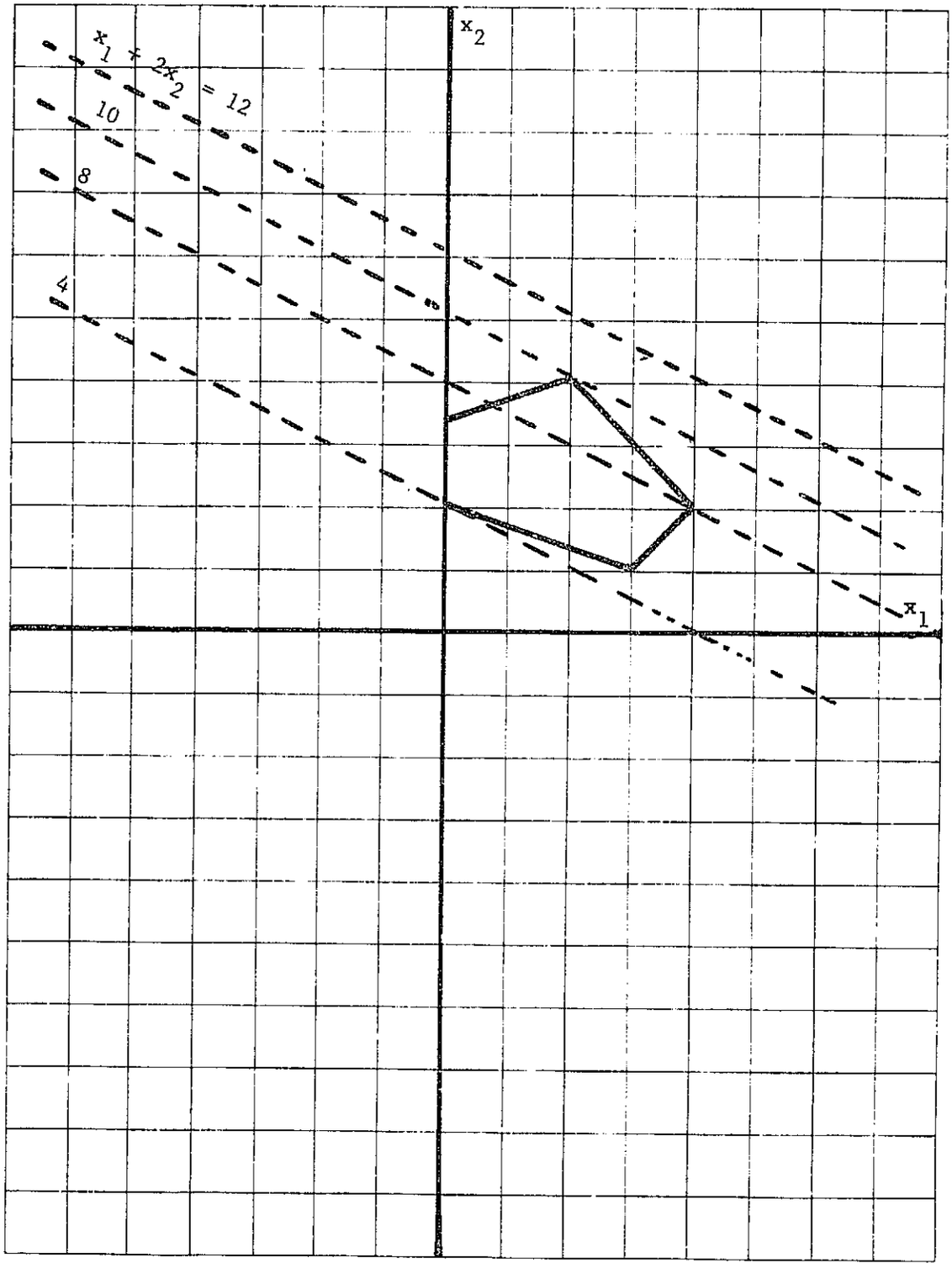


Figure 2-2 Solution of Linear Programming Example

2.2 Linear Programming and the Autopilot Problem

How do we apply this procedure to the autopilot problem? We consider the gain margin, phase margin, rise time, settling time, maximum overshoot, etc. as performance measures or objective functions, and constrain their values to an acceptable range. These acceptable values usually will err on the conservative side in order that unforeseen parameter variations will not allow excursions outside of "safe" values for the performance measures.

In the frequency domain, it is a relatively simple matter to identify the frequencies at which the performance measures are obtained. For example, a gain margin exists at those frequencies for which the phase of the system is 180 degrees, and a phase margin exists at frequencies for which the gain is zero decibels (See Figure 1-4). The frequencies at which pertinent performance measures are found are generally well separated, and in sequential order with increasing frequency, enabling a simple search routine to identify each measure in sequence.

In the time domain, a similar search technique permits identification of rise time, settling time, overshoot and the maximum deviation from final value.*

We thus treat these performance measures here in a general way to illustrate the design technique that has evolved into the COEBRA design algorithms.

Let the vector $\underline{X}$ represent that set of autopilot parameters whose values:

1) Determine directly or indirectly the value of one or more performance measures,
2) Can be varied in order to achieve an "optimum" performance measure in some reasonable sense.
3) Can be changed, periodically, during flight to maintain satisfactory vehicle stability.

---

* Due to structural reasonances, it is possible to have a time response in which the maximum excursion and the first half cycle overshoot are not coincident in time.

Thus we identify each performance measure as a function of this parameter set, assign a nominal value to them, and use these values to initialize an autopilot design. A frequency response or time response is then calculated and the set of performance measures examined. Assuming that a specific performance measure specification is not made, a single element of the parameter vector is changed and the frequency response or time response is repeated. The method of finite difference is then utilized to approximate the partial derivative of the performance measures with respect to this parameter to implement the performance measure constraint equation

$$M_i(\underline{X}_o + \Delta \underline{X}) = M_{io}(\underline{X}_o) + \sum_{j=1}^{n} \left(\frac{\partial M_i}{\partial x_j}\right)_o \Delta x_j \geq M_{is} \qquad i = 1, 2, \ldots, m \qquad \text{2-8*}$$

where M_i is the i-th performance measure, $\overline{X}_o$ the initial value of the parameter vector, M_{is} the specified performance measure minimum and $\Delta x_j = x_j - x_{jo}$. Rewriting 2-8 with the substitution for Δxj

$$\sum_{j=1}^{n} \left(\frac{\partial M_i}{\partial x_j}\right)_o \cdot x_j \geq M_{is} - M_{io}(\underline{X}_o) + \sum_{j=1}^{n} \left(\frac{\partial M_i}{\partial x_j}\right)_o \cdot x_{jo} \qquad i = 1, 2, \ldots m \qquad \text{2-9}$$

it is possible to invoke more than one mission flight condition (trajectory time event) in evaluating the specific performance measures, M_i, under consideration.

A fundamental restriction on this design algorithm is that the user must specify: 1) the autopilot configuration within the limits identified by

*2-8 is actually a truncated Taylor series expansion of the performance measure, M_i, about its initial value for each interation cycle.

the block diagram of Figure 1-7 or in a particular matrix format (see Appendix D); 2) the number of gains and filter time constants (elements of the vector $\overline{X}$) which are to be treated as variables and; 3) those autopilot parameters which are to be treated as constants for each iteration cycle. The algorithm then operates on an individual element of the vector $\underline{X}$ under the constraint

$$x_{j1} \leq x_j \leq x_{jn}, \quad j = 1, 2, \ldots, n \qquad 2\text{-}10$$

where the individual x_j may include several different flight conditions (trajectory times or vehicle states) or any one or more x_j may remain constant over several flight conditions or trajectory time events.

2.3 The Objective Function

We now formulate the objective or cost function in such a way that as the performance measures improve for one flight condition, the trend is for them to improve for all flight conditions, and at the same time each structural bending mode resonance will be forced to occur in the neighborhood of zero degrees phase. This is done by forcing the partial derivatives in the constraint equation to increase. In the format of equation 2-6, we thus write

$$y_1 = \sum_j \sum_t \sum_i W_1 (i, t) \cdot W_2 (i, t) \cdot S (i) \cdot \frac{\partial U (i, t)}{\partial x_j} \cdot x_j \qquad 2\text{-}10$$

where:

1) $\sum_j$ is the summation over all autopilot variables
2) $\sum_t$ is the summation over all flight conditions (trajectory time points or vehicle states).
3) $\sum_i$ is the summation over all performance measures at all flight conditions.
4) W_1 (i, t) refers to a weighting factor. For each performance measure, it is simply a ratio of the <u>desired</u> measure over the <u>actual</u> measure. Hence, if a performance measure requirement is not met, W_1 (i, t)

4) (Continued)

will be greater than unity. It becomes less than unity when a margin exceeds its desired objective. It is noted at this time that in the expression for y_1, i also indexes the phase angle at which each structural bending mode reasonates. For these values of i, the partial derivative indicates the rate of change of each modal peak phase with respect to each autopilot variable, and W_1 (i, t) is written so that the algorithm will attempt to force each mode to resonate near zero degrees phase. W_1 (i, t), will be large for modes that resonate near 180 degrees, and zero for modes that resonate at zero degrees. For some arbitrary angle like 90 degrees, W_1 (i, t) can equal unity.

5) W_2 (i, t) refers to a weighting factor that might be selected by the user. This would give the user the capability to eliminate certain performance measures from the optimization process or to emphasize other measures.*

6) S(i) refers to a scale factor. It serves to scale the margins and modal peak phases so that phase margins and gain margins can be optimized together. For example, it might be desired to equate a five degree increase in the rigid-body phase margin with a one decibel (12.2%) increase in the rigid body gain margin. Similarly, in the time domain, a correlation between rise time and overshoot may require scaling.

In summary, y_1 is a "weighted" linear combination of the "positive" changes in each performance measure. Note that this linear combination can

*W_1 (i,t) and W_2 (i,t) are initially preprogrammed. After an initial run of COEBRA, the user can, at his discretion, input new values for these weighting functions.

incorporate performance measures from all of the vehicle states (flight conditions) that are being designed together. The design algorithm will maximize y_1 (and hence seek to maximize all stability margins and/or time domain performance measures, and seek to force all modes to resonate near zero degrees phase) in the presence of the constraint matrix which includes constraints on each individual margin and each autopilot variable at each time point. The advantage of this algorithm lies in the fact that the constraint equations can specify the minimum requirements for each performance measure while the cost function seeks to maximize the performance measure.

2.4 Load Relief Cost Function

Structural bending moment loads on a launch vehicle are largely due to axial acceleration, aerodynamic loading, and control device deflections [Harris, 15]. Obviously, the booster autopilot can do little to affect axial acceleration, and therefore the main objective of a so-called load relief autopilot is to reduce aerodynamic loading due to angle of attack and to keep control device deflections to a minimum.

Hence, for this design algorithm, when the objective is to 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 the wind dorcing function (β_w). When the cost function is maximized, the peak values of β and δ are minimized.

A separate transient response routine is used to calculate the peak value of angle of attack (β_p) and control deflection (δ_p) due to β_w. As with stability margins, the method of finite differences is used to compute the first partial derivatives of β_p and δ_p with respect to the autopilot variables. The cost function is then formed from the first order terms of the Taylor Series expansions of β_p and δ_p about their nominal values.

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, or

$$y_2 = \sum_j \sum_t \left[W_1(t) * \left(\frac{\partial \beta_p(t)}{\partial X_j} \right)_o + W_2(t) * \left(\frac{\partial \delta_p(t)}{\partial X_j} \right)_o \right] * X_j \qquad 2\text{-}12$$

In the above expression:

(1) $\sum_j$ refers to the summation over all the autopilot variables;

(2) $\sum_t$ refers to the summation over all the vehicle states;

(3) $W_1(t)$ and $W_2(t)$ refer to weighting factors that may be specified by the user.

When maximizing load relief capability, the 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 on the allowed ranges of the individual autopilot variables. Note that multiple time point design is handled just as it is when maximizing other performance measures. Some final notes on the load relief cost function are now listed.

Since the so-called "rigid-body" (as opposed to flexible-body) angle of attack (β) and control deflection (δ) are the principal factors in determining structural bending moment loads, it is felt that only the rigid-body airframe equations of motion [Harris, 15] need to be used in the transient response routine that is used to calculate angle of attack and control deflection. Note also that these rigid-body airframe equations can include planar coupling (e.g., between the yaw and the roll planes), and hence the cost function can include control deflections from several planes (e.g., the yaw plane control deflections (δ_ψ) and the roll plane control deflections (δ_ϕ)).

The wind forcing function can be a series of steps and/or ramps that approximate the commonly used synthetic wind profile [15]. The wind forcing function can also be stochastic, and the design algorithm then minimizes the rms values of β and δ. This is done via Wiener's theorem and the filtering property of power spectral density functions [see Appendix C].

2.5 General Flow Chart

Figure 2-3 is a general flow chart summarizing the main steps involved in the algorithm. It shows the general flow from the initial autopilot for each iteration through the following routines:

(1) The routine that generates the frequency response and finds the stability margins, and the routine that generates the transient response and finds peak β and δ, rise time, settling time, overshoot, etc.

(2) The routine that computes partial derivatives;

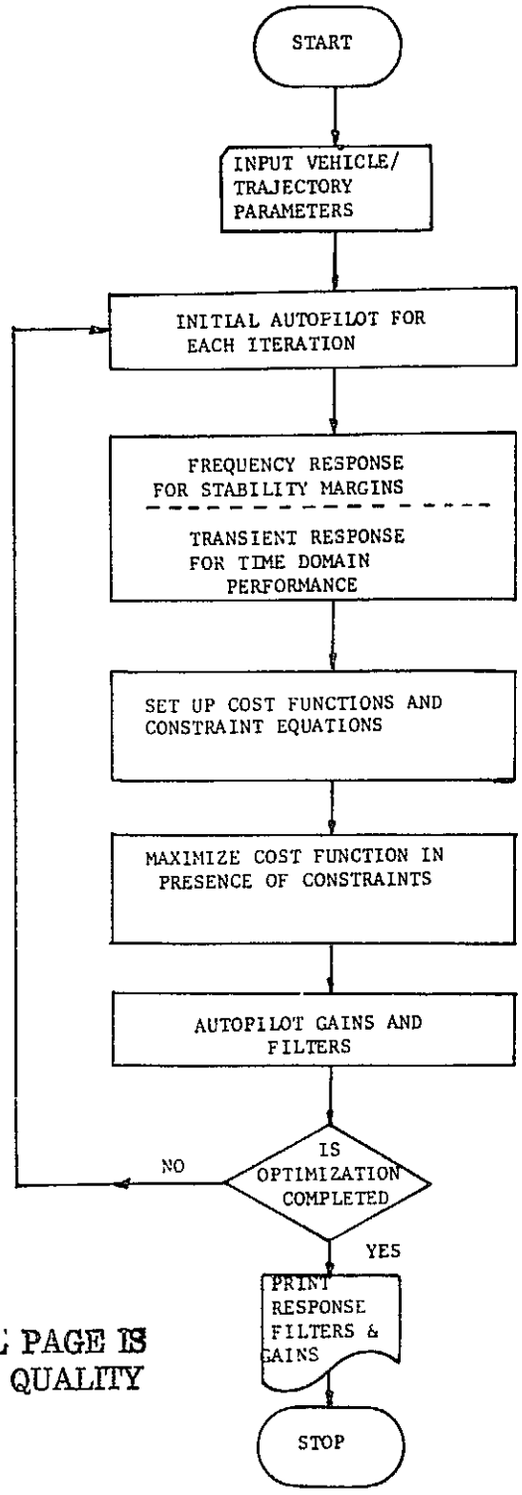
(3) The routines that set up the linear programming problem and solve it; and

(4) The routines that determine whether the design is complete.

If the design is not complete, another major iteration is begun with the best answer obtained in the previous iteration. In other words, the problem is relinearized about the best answer of the previous major loop, and another cycle through the major loop is performed. This iterative process continues until the local optimum is found.

Note that this design process satisfies the five main elements of the design criteria

(1) The method directly treats stability margin requirements and objectives, and structural bending moment load reduction and/or the time domain performance measures.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2-3 General Flow Chart of the Design Algorithm

(2) The method directly handles the user-selected autopilot configuration.

(3) The method directly handles the multiple time point (flight condition) design pr 'em.

(4) The method is not limited by the order of the system. Note that since this is a parameter optimization routine, the order of the autopilot does not necessarily increase with an increase in the order of the fixed parts of the system.

(5) The method can design either a digital autopilot or an analog autopilot.*

2.6 Selection of Iterative Step Size

Referring now to equation 2-3 and the sample problem, equation 2-5, we observe that to establish an optimum admissable solution, the function to be maximized was allowed to take on successive "constant" values. The speed with which the algorithm converges to a solution is dependent on the magnitude of the change in this constant value as the iterative process is carried out. Re-casting equation 2-10, to reflect the step-change in each autopilot variable, we obtain

$$\mathrm{MAX}\left\{(1+p)^{-1} \cdot X_{jo},\ X_{jmin}\right\} \leqslant X_j \leqslant \mathrm{MIN}\left\{(1+p) \cdot X_{jo},\ X_{jmax}\right\}$$

$$j = 1, 2, \ldots, n \qquad 2\text{-}13$$

where:

(1) X_{jmin} and X_{jmax} refer to the minimum and maximum values ever allowed for X_j.

---

*Under this submittal, the formal digital algorithm is not included. However, if the bi-linear transformation is applied to the pulse transfer function, the resulting w-plane description is directly analogous to the s-plane description contained herein. Thus, with the w-plane equations, COEBRA can be used directly to solve the digital autopilot problem.

(2) X_{jo} refers to the initial value of X_j on each iteration. Note that X_{jo} is the point about which the partial derivatives are computed, and about which the Taylor Series is expanded.

(3) P refers to the autopilot variable step-size for each iteration.

In words, if X_{jmin} and X_{jmax} 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_{jo} 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 autopilot variable step-size. In maximizing P, the Minor Loop uses two "indicators": (1) a counter that keeps track of the number of performance measures that are already met, and (2) a figure-of-merit that is a linear combination of their actual values. If the number of "met measures" increases, obviously the value of P can be increased. If the number of "met measures" does not change, the figure-of-merit is used to decide whether P can be further increased. In other words, the measure counter is used to reward those steps that result in an increase in the number of "met measures." Conversely, the counter prohibits those steps that result in a loss in the number of "met measures." Finally, the figure-of-merit is used to break ties when the measure counter does not change from one step to another.

The Minor Loop serves to either keep the problem linear on each major iteration, or to take advantage of the neglected nonlinearities when they might be helpful. In other words, the Minor Loop serves to keep the nonlinearities from "hurting" the steady convergence to a local optimum.

A major benefit of the Minor loop is that it allows the algorithm to

converge steadily to an "interior" optimum.

Since the solution to the linear programming problem always lies at a vertex of the admissable 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 performance measures constraint equations.

2.7 The Inner Loop

The second part of the step-size optimization routine can be illustrated by the following detailed expansion of a particular performance measure constraint equation

$$M_{io}(\bar{X}_o) + \sum_{j=1}^{n} \left(\frac{\partial M_i}{\partial X_j}\right)_o * (X_j - X_{jo}) \geqslant M_{is} \qquad (i = 1, \ldots, m) \qquad 2\text{-}14$$

In the above expression, there are two cases for M_{is}.

(1) If the particular performance measure is already met, then for the next iteration,

M_{is} = SPEC(i)

where SPEC(i) is the minimum allowed value for the ith performance measure

(2) If the particular performance measure is not yet met,

$$M_{is} = M_{io}(\bar{X}_o) + \text{STEP} * \left[\text{SPEC}(i) - M_{io}(\bar{X}_o)\right]$$

Before defining the purpose of the equations for M_{is}, note that in the second equation, (a) if STEP = 1, M_{is} = SPEC(i), and (b) if STEP = 0, $M_{is} = M_{io}(\bar{X}_o)$.

For a given value of P (autopilot variable step-size), there may not be a feasible solution to the linear programming problem if the present autopilot does not meet all of the performance measure constraints. In other words, the feasible region defined by the measure constraint equations may not overlap

the feasible region defined by P. By automatically reducing the value of STEP, the measure constraints are "loosened," until a feasible solution is possible for a given value of P. In this way, the value of P can be increased in a steady and rational manner, and the algorithm will be allowed to converge to a solution in a progressively improving manner.

2.8 Graphical Illustration of Step-Size Optimization

Figures 2-4 through 2-9 graphically illustrate the mechanics and the interaction of the two step-size optimization routines (the Minor Loop and the Inner Loop).

Figure 2-4 shows a hypothetical two dimensional condition that might exist for a rigid-body autopilot design problem. Figure 2-4 is a plot of the attitude error gain (K_D) versus the attitude rate gain (K_R). Plotted on the figure are three nonlinear stability margin constraint equations: (1) the aerodynamic gain margin; (2) the rigid-body phase margin; and, (3) the rigid-body gain margin. Figure 2-4 also shows where the "true" local optimum condition might be, where the objective is to maximize stability margins, and where all three stability margins are equally weighted. Obviously, the "true" optimum for this hypothetical case lies inside the feasible region where all three margin requirements are satisfied. The figure also shows what might be the "first guess" or initial condition on K_D and K_R.

Figure 2-5 shows what the constraints might look like when they are linearized about the initial condition. The figure also shows the linearized interior optimum, where again, all margins have been equally weighted. Note that the linearized optimum is not the same as the nonlinear or "true" optimum for this initial condition. Finally, Figure 2-5 shows the slope of the linearized cost function (y) and the direction in which it increases.

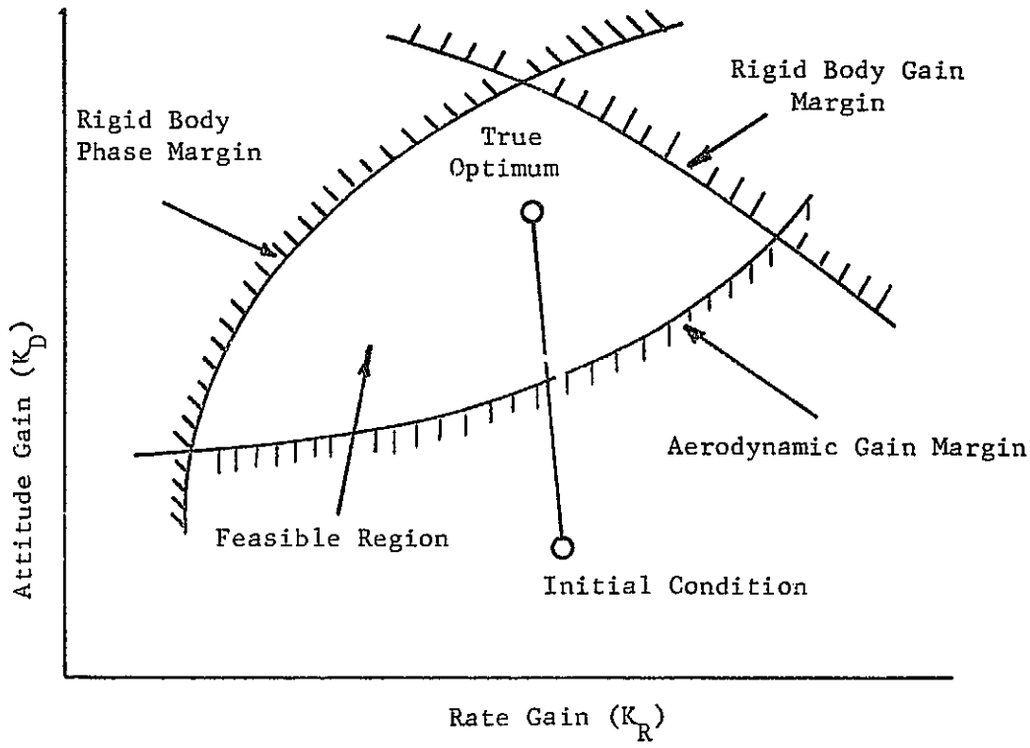


Figure 2-4 Nonlinear Constraints

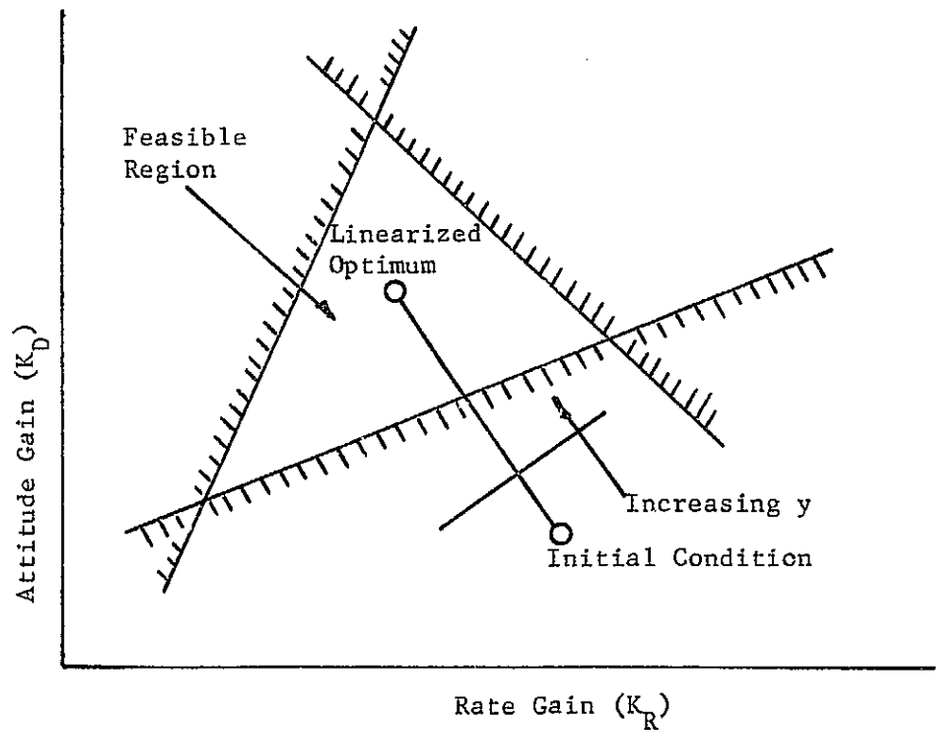


Figure 2-5 Linearized Constraints

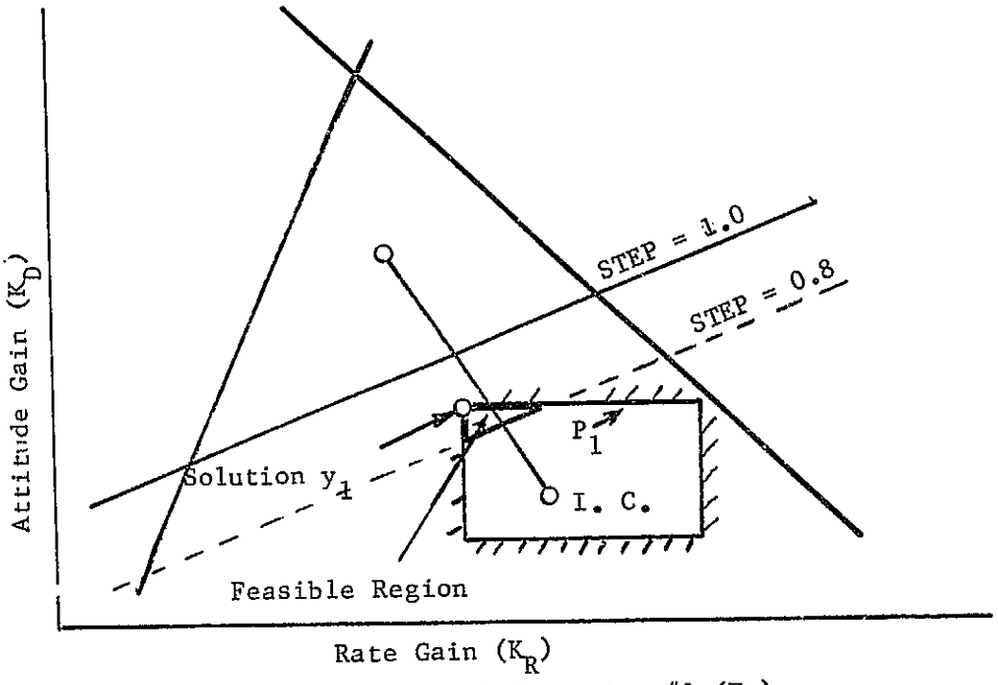


Figure 2-6 Minor Loop Step-size #1 (P_1)

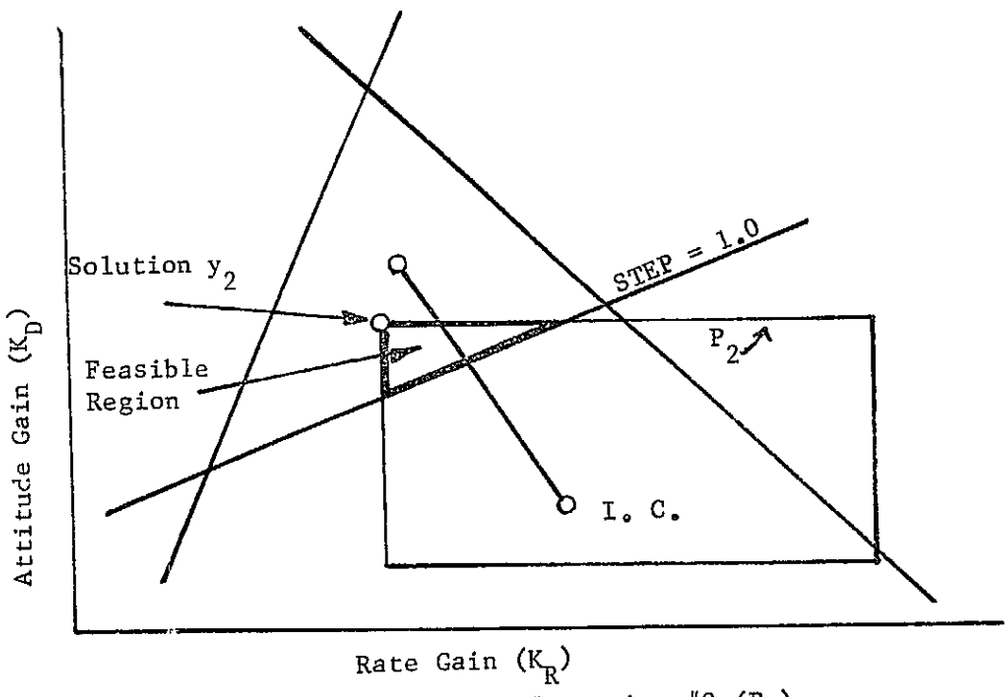


Figure 2-7 Minor Loop Step-size #2 (P_2)

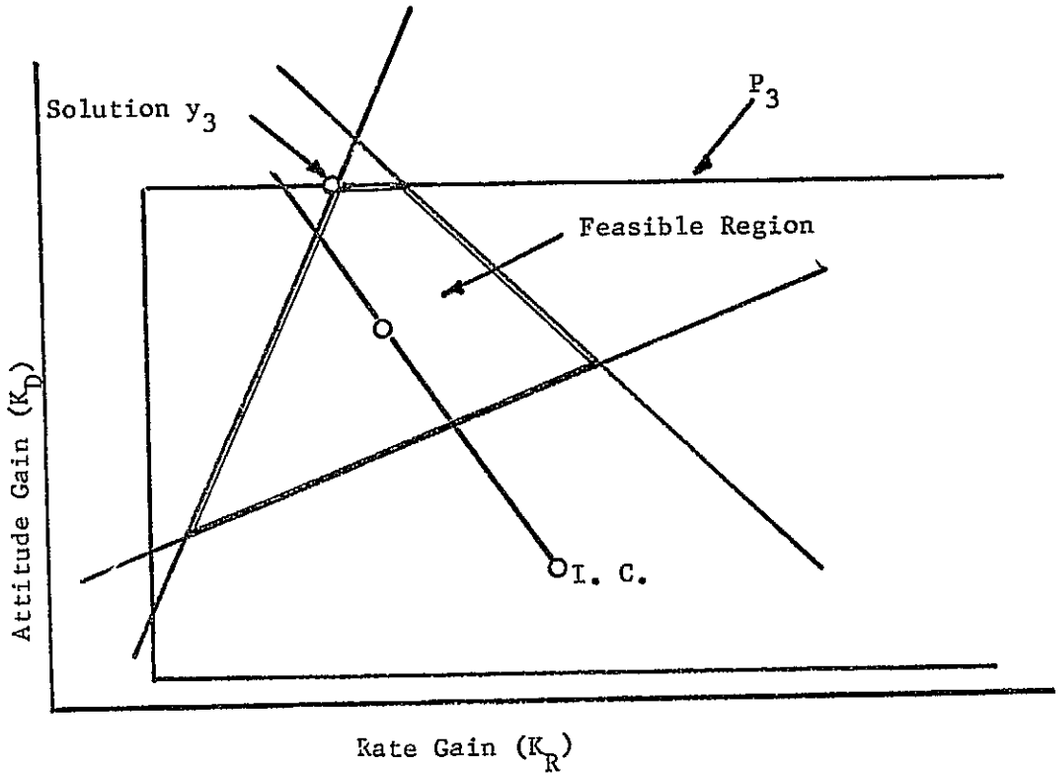


Figure 2-8 Minor Loop Step-size #3 (P_3)

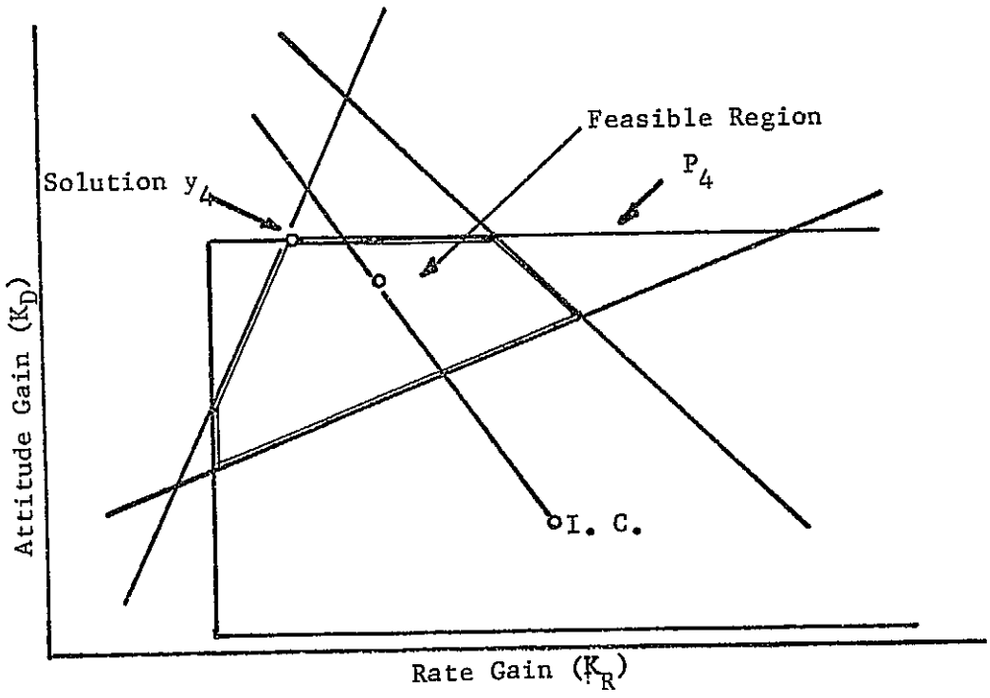


Figure 2-9 Minor Loop Step-size #4 (P_4)

Figure 2-6 illustrates the feasible region defined by the autopilot variable constraint equations on K_D and K_R for step-size #1 (denoted P_1). This feasible region does not overlap the feasible region defined by the stability margin constraint equations. With the initial condition, the rigid body phase and gain margin constraints are satisfied, but the aerodynamic gain margin is not. Hence, the design method enters the inner loop, and relxes the aerodynamic gain margin constraint until a feasible region exists for both the margin constraints and the autopilot variable constraints. This relaxation is accomplished by reducing the parameter denoted as STEP.

When STEP is unity, no relaxation exists. When STEP is reduced to 0.8, the aerodynamic gain margin constraint is relaxed enough so that a feasible region exists. When STEP is 0.8, this means that an "80% improvement" is required for the margin that is not yet satisfied. This so-called "required margin improvement" becomes very important when the optimum is exterior to the feasible region. This is the case most of the time for launch vehicle autopilot design.

As indicated on Figure 2-6, the optimum solution for the first step of the minor loop exists at y_1. Comparing y_1 with the nonlinear cost function and constraints shown on Figure 2-4, it is seen that stability at y_1 is better than at the initial condition. This "improved stability" is indicated by the figure-of-merit which, as discussed previously, is a linear combination of the stability measures. Note that the so-called "met-measure counter" indicates that at y_1, there are still only two margins that are satisfied. Note that the measure counter and the figure-of-merit are formed from an actual evaluation of the frequency response. In other words, they are not computed from the linearized cost function and the linearized constraint equations.

Hence, since y_1 is better than the initial condition, the design process

advances, using the same set of partial derivatives, and hence the same linearized cost function and margin constraint equations that were calculated at the initial condition.

As shown in Figure 2-7, the design method now increases P from P_1 to P_2. Figure 2-7 shows the feasible region defined by the autopilot variable constraint equations for P_2. An overlap exists between the feasible regions defined by the margin and autopilot constraint equations, and hence the inner loop need not be used. For P_2 the optimum solution exists at y_2. By comparing y_2 with the nonlinear constraint equations of Figure 2-4, the margin counter indicates that there are now three margins that are satisfied. Since improved stability has again been achieved, P is further increased from the original initial condition.

As shown in Figure 2-8, P is now increased to P_3. For this step, overlap also exists, and the optimum solution is at y_3. By comparing y_3 to the nonlinear constraints of Figure 2-4, it is seen that the rigid-body phase margin requirement is no longer satisfied. The margin counter indicates that only two margins are satisfied by y_3. Hence, y_3 is not as good as y_2 and P must be reduced.

Figure 2-9 shows the autopilot constraint equations for P_4, where $P_2 < P_4 < P_3$. Again, the inner loop is not needed, and the optimum solution exists at y_4. By comparing y_4 to Figure 2-4, the margin counter shows that there are three margins that are met. But, the figure-of-merit shows that y_2 is better than y_4. Postulating that the difference between P_2 and P_4 is less than some convergence criterion, the algorithm stops this so-called major iteration at y_2. The values of K_D and K_R at y_2 become the initial condition for the next major iteration. At y_2, the problem is relinearized. A new set of partial derivatives is computed, and a new cost function and new constraint equations are formed. As the design progresses, the linearized optimum gets closer and closer to the nonlinear or "true" optimum. It will be observed in the following that the

convergence criteria can be used to terminate this iterative design process.

Table 2-1 summarizes the results of Figures 2-4 through 2-9. These figures have been used to demonstrate steady convergence to a local interior optimum. We will next consider convergence to a local exterior optimum.

As a final note, Figures 2-4 through 2-9 demonstrate that this algorithm does not require that the initial condition lie within the feasible region.

2.9 Convergence to an Exterior Optimum

We now illustrate how the algorithm converges to an exterior or "constrained" optimum. Figure 2-10 shows a case that might exist when optimizing load relief capability since for this phase of design, the optimum solution almost always is exterior to the feasible region defined by the margin constraint equations. Figure 2-10 is a hypothetical two-dimensional case where (1) the nonlinear margin constraint might represent the so-called aerodynamic gain margin, (2) X_1 might represent the attitude error gain, and (3) X_2 might represent the so-called load relief loop gain. Figure 2-10 also shows the nonlinear constrained optimum.

In Figure 2-10 the initial condition on X_1 and X_2 is outside the feasible region. Figure 2-11 shows what the margin constraint and the nonlinear optimum might look like when the problem is linearized about the initial condition. The first step of the algorithm is to "get feasible" and Figure 2-11 will show that in so doing, the algorithm still attempts to approach the optimum.

Referring to Figure 2-11, after a series of iterations through the minor and the inner loops, the solution is shown to exist at y_1. With the linearized margin and cost function as shown, this is the best solution this major iteration can achieve without violating the nonlinear margin constraint.

At y_1, the problem is relinearized as shown in Figure 2-12. This figure shows that any step in the direction of the gradient to the linearized

Table 2-1 Summary of Figures 2-4 Through 2-9

Figure	Minor Loop Step Size	Constraint Relaxation?	Solution	Numbers of Margins Met	Criterion on Fig.-of-Merit	Best Solution So Far
2-5	Initial Condition	---	y_o	2	--	I. C.
2-6	P_1	Yes (STEP=.8)	y_1	2	$y_1 > y_o$	y_1
2-7	P_2 where $P_2 > P_1$	No	y_2	3	--	y_2
2-8	P_3 where $P_3 > P_2$	No	y_3	2	--	y_2
2-9	P_4 where $P_2 < P_4 < P_3$	No	y_4	3	$y_2 > y_4$	y_2

Y_2 serves as initial condition for next major iteration since P_4-P_2 satisfies convergence criteria.

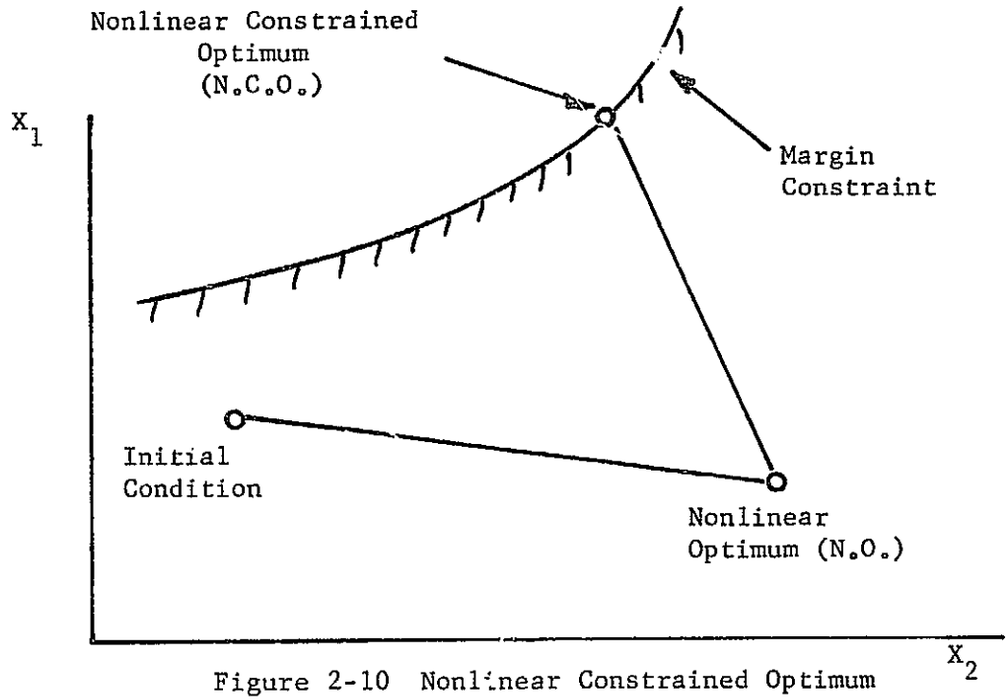


Figure 2-10 Nonlinear Constrained Optimum

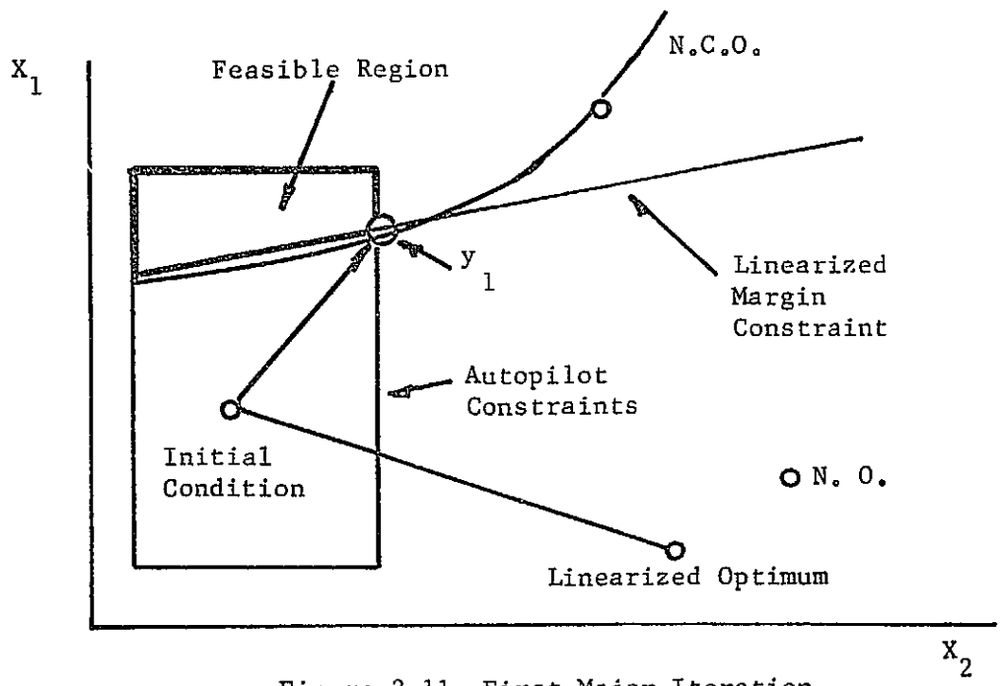


Figure 2-11 First Major Iteration

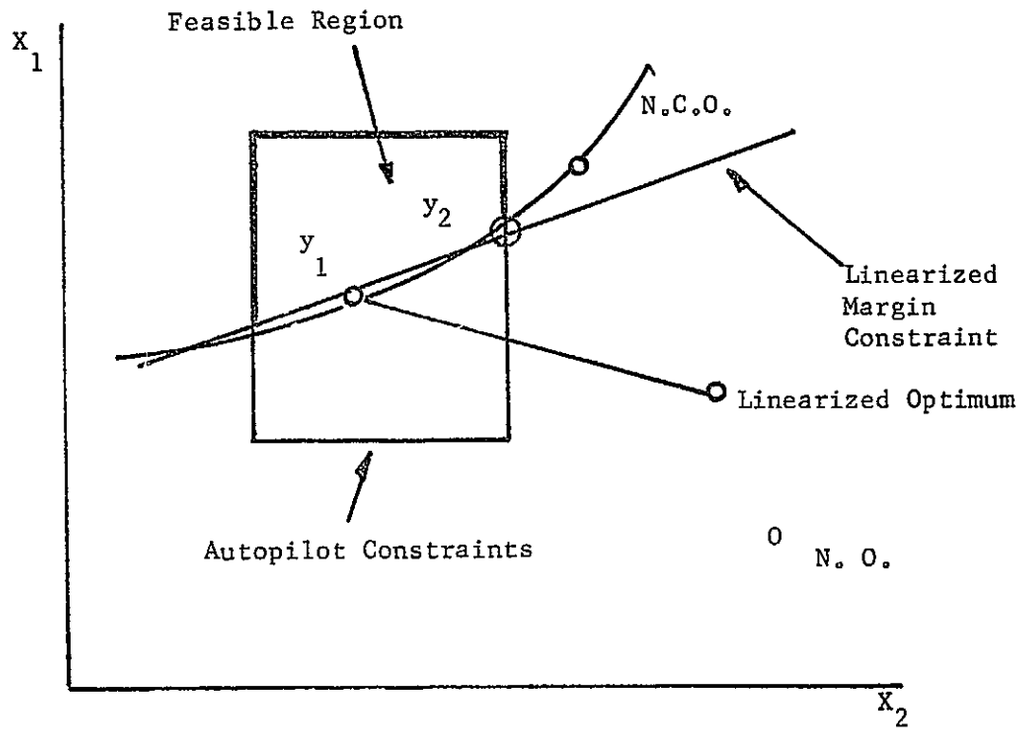


Figure 2-12 Second Major Iteration

optimum would yield an unfeasible solution. Referring to Figure 2-12, again after a series of iterations through the minor and the inner loops, the solution exists at Y_2. With the linearized margin and cost function as shown, this is the best solution this major iteration can achieve without violating the nonlinear margin constraint. With each major iteration, the linearized constrained optimum approaches the nonlinear constrained optimum.

We have illustrated how the algorithm first "gets feasible," and then moves along or parallel to a constraint for the case of a constrained optimum. Example 6 of Appendix B will dramatically demonstrate this situation.

Because of the weighting factors in the cost function and figure-of-merit, situations with an exterior optimum can also exist when optimizing performance measures.

2.10 Termination

We now discuss the two ways in which the design process can be terminated.

The first way might be referred to as self-termination, where the algorithm finds a local optimum and can achieve no improvement over the initial autopilot for a given iteration. The so-called measure counter and the figure-of-merit define this "improvement." Recall that the counter "counts" the number of "met measures," and the figure-of-merit is a linear combination of the actual performance measures. The counter never allows the algorithm to lose a measure, and an iteration is considered better if the counter increases. The figure-of-merit is used to break ties in the counter. An iteration is considered no good if the figure-of-merit decreases. An iteration is considered better only if the figure-of-merit increases by a certain percentage (as specified by the user).

The user may wish to "reward" a certain performance measure only up to a certain <u>desired</u> value. In other words, up to a desired value, the figure-of-merit will include the actual value of the measure. When the measure exceeds

this desired value, the figure-of-merit will only include this desired value. If this happens for all performance measures, the figure-of-merit will not change at all from one iteration to the next, and the algorithm will have found an optimum that not only meets the requirements, but also satisfies the desired objectives. Examples 1 and 2 of Appendix B will illustrate cases where this happened.

The case most likely to be encountered is when not all performance measures exceed their desired values, and the figure-of-merit improves only slightly from one iteration to the next. The percent improvement is less than that required by the user and the design process terminates. For this case, the local optimum may yield a solution that satisfies all the design requirements, in which case the problem is solved. However, if the local optimum does not satisfy the 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.

The second way that the design may be terminated is directly by the user. He may specify termination after a certain number of major iterations or after a certain amount of computer time.

2.11 Overall Flow Chart

Figure 2-13 is a detailed overall flow chart of this nonlinear programming algorithm as it is applied to the problem of launch vehicle autopilot design. It shows the flow of information from the initial autopilot, all the way through the step-size optimization routines, through the termination routine, and back to the initial autopilot for the next major iteration.

In the block showing where the partial derivatives are calculated, reference is made to closed-loop roots. This illustrates the possibility of

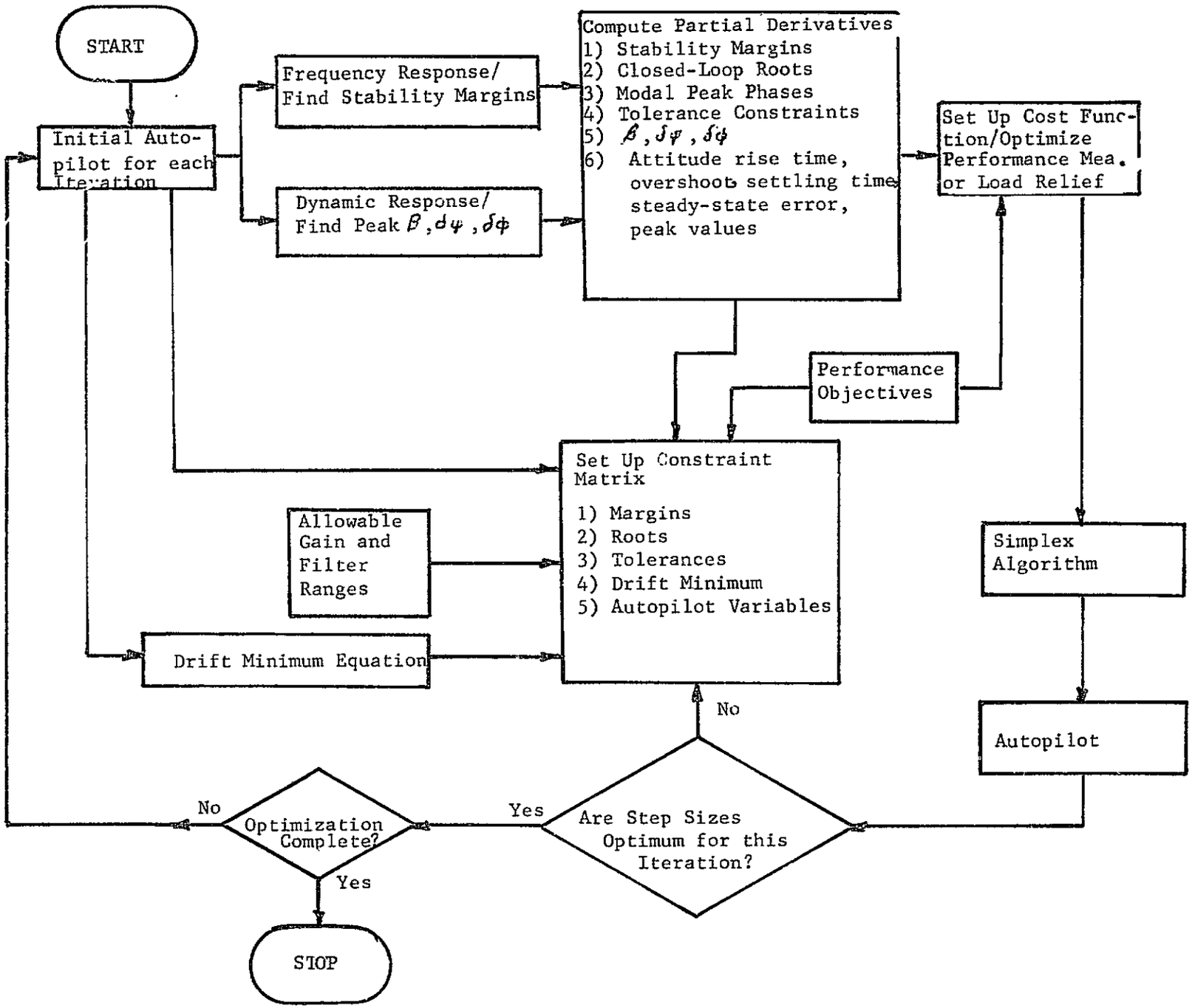


Figure 2-13 Overall Flow Chart of the Design Algorithm

not only optimizing stability margins, but also of optimizing locations of at least the so-called dominant closed-loop roots. In addition, a tolerance constraints routine is indicated. This routine is utilized to prevent the attitude error vector, the rate vectors, the accelerometer loop vector, etc. from getting larger than the total resultant vector at all frequencies. When the individual vectors become much larger than the resultant, vector cancellation may result, which can lead to problems when the airframe parameter tolerances are considered.

In the block showing the constraint matrix set up, reference is made to the "Drift Minimum" condition, (Greensite [14] and Hoelker [17]) which basically is a steady-state relationship between the autopilot gains. This steady-state relationship results in a mix between the forces due to gravity, aerodynamics, and control deflections, that yields a zero net force perpendicular to the vehicle's velocity vector. This illustrates that it is possible not only to put constraints on margins and root locations and on min-max values of the autopilot variables, but also to constrain the autopilot variables to satisfy other conditions like the Drift Minimum condition.

Figure 2-13 also shows a block labeled "Simplex Algorithm." This is in reference to Dantzig's method [9] for solving linear programming problems.*

The block labeled "Performance Objectives" refers to the fact that design objectives are used to form the cost function, while design requirements are used to form the constraint matrix.

2.12 Step-Size Optimization Flow Charts

Figure 2-14 illustrates the step-size optimization routines (namely, the Minor Loop and the Inner Loop) and their relationship to the other routines shown in Figure 2-13.

*See Appendix E

The Inner Loop is used to relax constraints in order to yield a feasible solution to the linear programming problem. The Minor Loop is used to maximize the autopilot variable step-size in order to get the "maximum mileage" out of each set of partial derivatives, and in order to allow the algorithm to steadily converge to a local optimum, particularly to an interior optimum.

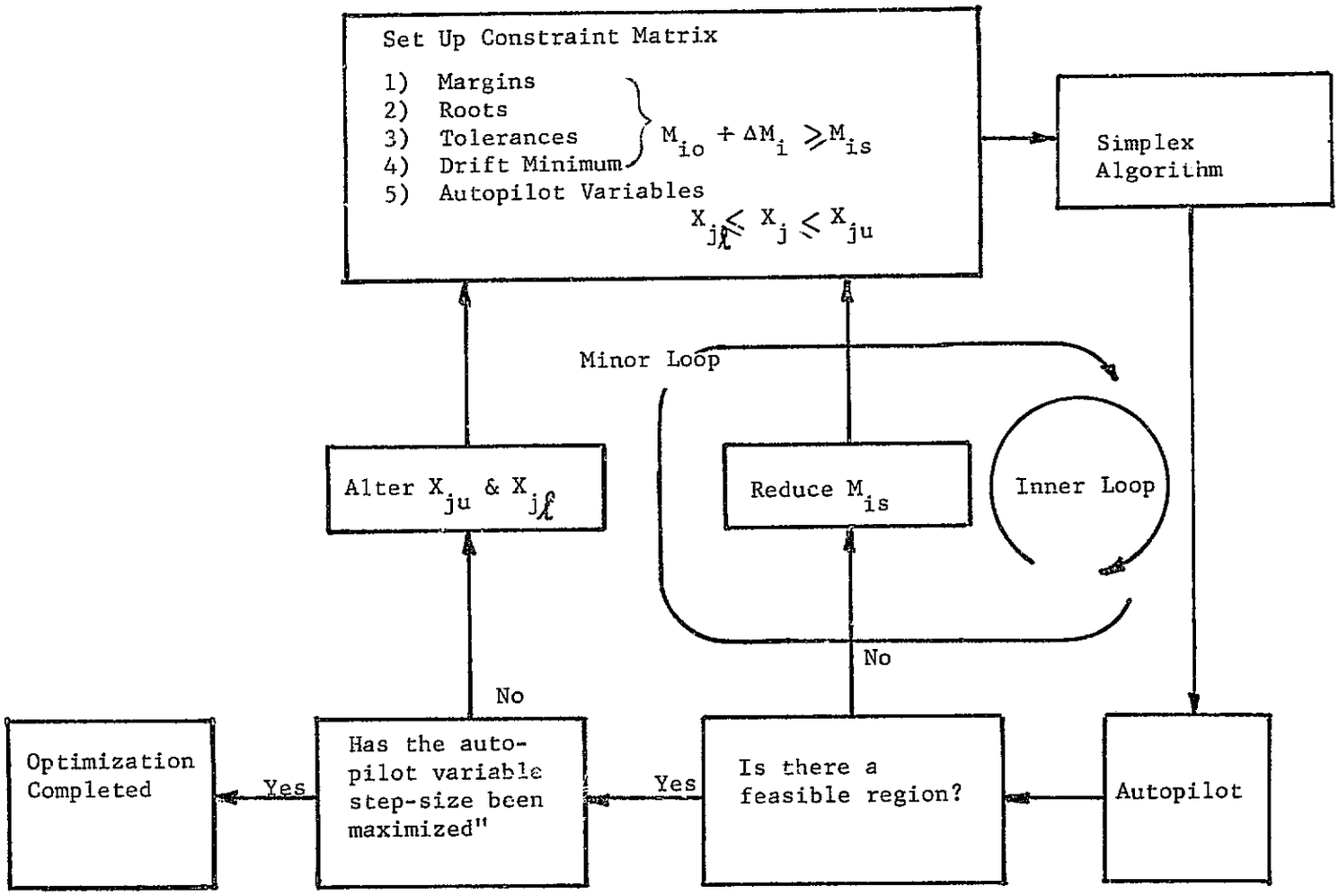


Figure 2-14 Step-Size Optimization Flow Chart

CHAPTER 3

SUMMARY

3.1 Summary and Conclusions

As shown in the examples of Appendix B, the COEBRA program clearly demonstrates that this algorithm successfully solves the problem of automating practical launch vehicle autopilot design and optimization. Perhaps the primary reason for the success of this algorithm is that its approach to design is much the same as the engineer's approach.

Via this algorithm, the COEBRA program satisfies the five basic design requirements:

(1-a) The COEBRA program deals directly with the "optimization" of performance measures in both the frequency domain and the time domain as well as to constrain the location of the rigid body dominant roots. COEBRA imposes an inequality constraint in each individual performance measure and each pair of dominant poles at all flight conditions being considered simultaneously. With the optimum performance measure cost function, COEBRA not only realizes the minimum performance measure requirement, but also seeks to "optimize" all performance measures. With a cost function distinct from the constraint matrix, the cost function can be formed from the performance measure <u>objectives</u>, while the constraint matrix can be formed from the performance measure <u>requirements</u>;

(1-b) With the "optimize load reduction" cost function (formed via a time domain transient response routine), COEBRA seeks to minimize structural bending moment loads (β, δ_ψ, and δ_ϕ due to winds)

while meeting the minimum performance measure requirements.

(1-c) COEBRA can constrain the autopilot parameters to the so-called Drift Minimum condition [14], [17], thereby minimizing trajectory dispersions. In fact, COEBRA can design a Drift Minimum autopilot that has the maximum amount of load relief capability and that meets the minimum stability margin requirements;

(2) COEBRA designs with a user-selected autopilot configuration. From the outset, only practical controllers are considered since the user selects the number and types of feedback loops and the number of gains and filters. COEBRA optimizes the values of the parameters within this feedback structure and constrains the minimum and maximum allowed values on each parameter;

(3) COEBRA handles the problem of multiple time point design by forming the cost function and matrix of constraint equations from performance measures and wind responses at several time points or vehicle states for both deterministic and random forcing functions. In this manner, all vehicle states are optimized simultaneously. Autopilot parameters can be shared between the vehicle states.

A novel feature of the COEBRA program and this design algorithm is that multiple time points are handled by considering a separate airframe for each time point. It is obvious that these "separate airframes" can come from the same flight condition or trajectory time point. For example, the "first airframe" can be the nominal airframe at time t_1 while the "second airframe" can be the airframe at time t_1 with a tolerance on one or more of the

vehicle parameters. In this way, COEBRA can treat both the nominal and the toleranced airframe together to yield a single autopilot that will handle both conditions;

(4) COEBRA can handle a very high order system (30th and greater with up to eight bending and slosh modes per time point). With a user selected feedback configuration, the complexity of the autopilot does not necessarily increase with an increase in the order of the fixed parts of the system. Parameters like sensor and actuator dynamics are included in a very straightforward manner and their inclusion only increases the required computations;

(5) COEBRA designs analog autopilots via the S-plane frequency response, and digital autopilots via the W-plane frequency response.

Examples in Appendix B show that this algorithm can handle both interior and exterior optima. The examples also show that the initial conditions on the controller parameters need not yield feasible solutions, i.e., solutions that meet the constraint requirements. In fact, the examples of this appendix demonstrate that the initial condition on the autopilot parameters need not even yield a stable system.

3.2 Projected Applications

Even though the class of problems this algorithm can handle has not been established, it would appear that it can handle a large variety of engineering-type problems.

For example, it would appear it can handle the problem of designing an

airplane flight control system with the so-called flying qualities design criteria[2]. These criteria include: (1) the longitudinal plane requirements on phugoid stability, flight path stability, and short period response; (2) the lateral-directional flying qualities criteria on the responses of the dutch-roll mode, the spiral mode and the roll mode; and (3) miscellaneous requirements on capability to perform crosswind landings, coordinated turns, etc. These criteria could simply be added to the flexible-body stability margin design requirements that are already included in COEBRA.

Another problem that this algorithm can handle is the design of a reaction control system. This type of control system uses discrete control. This algorithm can be used to optimize phase plane switching logic like the so-called "near-minimum-fuel" switching logic developed by Carney and Conover[5]. Their phase plane logic was developed for a digital attitude control system that requires no rate gyros.

Another problem that this algorithm can surely handle is the design of the autopilot for an interplanetary spacecraft like the Mariner [Kopf, 22]. In order to handle the Mariner autopilot design problem, COEBRA uses a transient response routine that puts requirements such as rise time, overshoot, settling time and steady-state error on the vehicle's attitude response due to guidance commands.

To conclude this discussion, a paper written by Robinson[28] is noted. In this paper, Robinson states that the COEBRA algorithm should prove fruitful in the optimal control of distributed parameter systems.

BIBLIOGRAPHY

1 Anderson, B. and J. Moore, Linear Optimal Control, Englewood Cliffs, New Jersey, Prentice-Hall, 1971.

2 Anonymous, Military Specification - Flying Qualities of Piloted Airplanes, MIL-F-8785B(ASG), August, 1969.

3 Blackburn, T. R. and D. R. Vaughan, "Application of Linear Optimal Control and Filtering Theory to the Saturn V Launch Vehicle," Proceedings of the 19th Congress of the International Astronautical Federation, October, 1968.

4 Burris, P. M. and M. A. Bender, "Aircraft Load Alleviation and Mode Stabilization (LAMS)," AFFDL-TR-68-161, November, 1969

5 Carney, R. and T. Conover, "Digital Attitude Control System" Trans. IEEE on Aerospace and Navigational Electronics, March, 1963.

6 Chang, S. S. L., Synthesis of Optimum Control Systems, New York, N. Y., McGraw-Hill, 1961.

7 Coffee, T. C., "The Automatic Frequency-Domain Synthesis of Multiloop Control Systems," Proceedings AIAA Aerospace Computer System Conference, September 1969.

8 Currie, M. G., "Experience With Modern Control Theory on Flight Control Design," McDonnell Douglas Astronautics Company, Western Division, WD1315, February 1970.

9 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.

10 Davidon, W. C., "Variable Metric Method for Minimization," A. E. C. Research and Development, ANL-5990, December 1959.

11 DiStefano, J. J., et. al., "Schaum's Outline of Theory and Problems of Feedback and Control Systems," Schaum's Outline Series, New York, N. Y., McGraw-Hill, 1967.

12 Edinger, L. D., et. al., "Design of a Load Relief Control System," NASA CR-61169, April 1967.

13 Fletcher, R. and M. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer Journal, 6, 2 (1963).

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

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

16 Hendricks, T. and H. D'Angelo, "An Optimal Fixed Control Structure Design with Minimal Sensitivity for a Large Elastic Booster," Proceedings 5th Annual Allerton Conference on Circuit and Systems Theory, Urbana, Illinois, October 1967.

17 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.

18 Hofmann, L. G., "Topics on Practical Application of Optimal Control to Single and Multiple Control-Point Flight Control Problems," AFFDL-TR-70-52, February 1971.

19 Hooke, R. and T. Jeeves, "Direct Search Solution of Numercial and Statistical Problems," JACM, Vol. 8, No. 2, April 1961.

20 Kalman, R. E., "When is a Linear-Control System Optimal?," Journal of Basic Engineering, March 1964.

21 Klingman, W. R. and D. M. Himmelblau, "Nonlinear Programming with the Aid of a Multiple-Gradient Summation Technique," JACM, Vol. II, No. 4 (October 1964).

22 Kopf, E. H., "A Mariner Orbiter Autopilot Design," Jet Propulsion Laboratory, Pasadena, California, NASA Report No. 32-1349, January 1969.

23 Kuo, B. C., Analysis and Synthesis of Sampled-Data Control Systems, Englewood Cliffs, New Jersey, Prentice-Hall, 1963.

24 Lorenzetti, R. C. and G. L. Nelson, "Direct Lift Control For the LAMS B-52," AFFDL-TR-68-134, October 1968.

25 _____, "Computerized Design of Optimal Direct Lift Controller," J. Aircraft, Vol. 6, No. 2, March-April 1969.

26 _____, "Direct Lift Control for Approach and Landing," J. Aircraft, Vol. 6, No. 3, May-June 1969.

27 Ogata, K., State Space Analysis of Control Systems, Englewood Cliffs, New Jersey, Prentice-Hall, 1967.

28 Robinson, A. C., "A Survey of Optimal Control of Distributed-Parameter Systems," Wright-Patterson Air Force Base, ARL 69-0177, November 1969.

29 Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints," J. Soc. Indust. Appl. Math, Vol. 8, No. 1, March 1960.

30 _____, "The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints," J. Soc. Indust. Appl. Math, Vol. 9, No. 4, December 1961.

31 Rynaski, E. G., et. al., "Optimal Control of a Flexible Launch Vehicle," Cornell Aeronautical Laboratory, NASA CR-80772, July 1966.

32 _____, "Sensitivity Considerations in the Optimal Control of a Flexible Launch Vehicle," Cornell Aeronautical Laboratory, NASA CR-89568, June 1967.

33 Shah, B. V., et. al., "The Method of Parallel Tangents (PAR-TAN) For Finding an Optimum," Office of Naval Research, NR-042-207 (No. 2), 1961.

34 Stapleford, R. L., et. al., "A Practical Optimization Design Procedure for Stability Augmentation Systems," AFFDL-TR-70-11, October 1970.

35 Stear, E. B. and C. P. Lefkowitz, "Automated Design of Space Booster Control Systems," Proceedings Joint Conference on Mathematical and Computer Aids to Design, Anaheim, California, 1969.

36 Stein, G. and A. H. Henke, "A Design Procedure and Handling-Quality Criteria for Lateral-Directional Flight Control Systems," AFFDL-TR-70-152, May 1971.

37 Vandierendonck, A. J., "Design Method for Fully Augmented Systems for Variable Flight Conditions," AFFDL-TR-71-152, January 1972.

38 Whitbeck, R. F., "A Frequency Domain Approach to Linear Optimal Control," J. Aircraft, Vol. 5, No. 4, July-August 1968.

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

40 Newton, G. C. Jr., L. A. Gould and [illegible]. Kaiser, Analytical Design of Linear Feedback Controls, New York, [illegible] Wiley & Sons, Inc., 1951.

41 Wiener, W., Extrapolation, Interpolation and Smoothing of Stationary Time Series, John Wiley & Sons, MIT Technology Press, 1957.

42 Lanning, J. H., Jr. and R. H. Battin, Random Processes in Automatic Control, New York, McGraw-Hill Book Company, 1956.

43 Davenport, W. and W. Root, An Introduction to the Theory of Random Signals and Noise, New York, McGraw-Hill Book Company, 1958.

44 Hauser, F., "Computer Optimization of Elastic Booster Autopilots," PhD Thesis. Department of Electrical Engineering, University of Denver, December 1972.

45 Wilts, C. H., Principles of Feedback Control, Reading, Mass., Addison-Wesley Publishing Company, 1960.

46 Horowitz, I. M., Synthesis of Feedback Systems, New York, Academic Press, 1963.

47 Frager, R. A., W. J. Duncan and A. R. Collar, Elementary Matrices, New York, MacMillan, 1947.

48 Horowitz, I. C. and V. Shakel, "Superiority of Transfer Function Over State-Variable Methods in Linear Time-Invariant Feedback System Design," IEEE Transactions on Automatic Control, Vol. AC-20, No. 1, February 1975, pp. 84-97.

49 Hadley, G., Linear Programming, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1962.

50 Gass, S. I., Linear Programming, Methods and Applications, Mc-Graw Hill Book Company, New York 1964.

51 Orden, A. "Application of the Simplex Method to a Variety of Matrix Problems" in "Symposium of Linear Inequalities and Programming", Directorate of Management Analysis, DCS/Comptroller, Headquarters, U. S. Air Force, Washington, D. C., 1954.

APPENDIX A

COMPUTATIONAL AIDS FOR PAPER AND PENCIL DESIGN

PRECEDING PAGE BLANK NOT FILMED

This appendix provides a few simple computational aids for a preliminary paper and pencil design. A table of rectangular to polar coordinate transformations (p A-2 to A-4 greatly facilitates the construction of the phase/frequency plots, from which, with the corrected asymptotic gain-frequency plots (Bode plots) obtained with the assistance of the standard frequency response curves of pages A-5 through A-16, the gain-phase (Nyquist) diagrams can be constructed. Representative gain phase curves for $-\infty < \omega < \infty$ are shown on pages A-17 through A-20.

For completeness, pp. A-21 through A-24 illustrate representative root loci for closed-loop systems.

PRECEDING PAGE BLANK NOT FILMED

$1+j\omega t=AE^{j\theta}$

WT	A	20 log A	θ-degrees
0.05	1.000	0.000	2.9
0.10	1.005	0.043	5.7
0.15	1.010	0.086	8.6
0.20	1.020	0.172	11.3
0.25	1.035	0.299	14.0
0.30	1.045	0.384	16.7
0.35	1.060	0.506	19.3
0.40	1.080	0.666	21.8
0.45	1.095	0.708	24.5
0.50	1.120	0.984	26.6
0.55	1.140	1.138	28.8
0.60	1.165	1.360	30.95
0.65	1.191	1.511	33.0
0.70	1.220	1.727	35.0
0.75	1.250	1.938	36.9
0.80	1.280	2.144	38.7
0.85	1.310	2.345	40.35
0.90	1.343	2.562	42.0
0.95	1.378	2.785	43.5
1.00	1.414	3.010	45.0
1.05	1.453	3.125	45.8
1.10	1.485	3.435	47.7
1.15	1.522	3.666	49.0
1.20	1.560	3.862	50.2
1.25	1.600	3.032	51.35
1.30	1.640	4.297	52.45
1.35	1.680	4.506	53.5
1.40	1.720	4.711	54.5
1.45	1.760	4.910	55.4
1.50	1.800	5.105	56.3

ORIGINAL PAGE IS
OF POOR QUALITY

$$1 + j\omega T = Ae^{j\theta}$$

ωT	A	20 log A	θ-degrees
1.55	1.840	5.296	57.1
1.60	1.885	5.506	58.0
1.65	1.930	5.711	58.8
1.70	1.975	5.911	59.6
1.75	2.020	6.107	60.3
1.80	2.060	6.277	61.0
1.85	2.100	6.444	61.65
1.90	2.150	6.649	62.3
1.95	2.190	6.809	62.9
2.00	2.235	6.987	63.5
2.05	2.280	7.159	64.05
2.10	2.330	7.347	64.6
2.15	2.375	7.504	65.1
2.20	2.420	7.676	65.6
2.25	2.465	7.836	66.1
2.30	2.525	8.035	66.5
2.35	2.555	8.148	67.0
2.40	2.600	8.299	67.4
2.45	2.645	8.448	67.8
2.50	2.690	8.595	68.2
2.55	2.730	8.723	68.6
2.60	2.780	8.881	69.0
2.65	2.830	9.036	69.35
2.70	2.875	9.173	69.7
2.75	2.925	9.323	70.05
2.80	2.970	9.455	70.4
2.85	3.010	9.571	70.7
2.90	3.060	9.714	71.0
2.95	3.120	9.883	71.3
3.0	3.16	9.994	71.6

$$1 + j\omega T = Ae^{j\theta}$$

ωT	A	20 log A	θ-degrees
3.1	3.26	10.264	72.15
3.2	3.35	10.501	72.65
3.3	3.45	10.756	73.15
3.4	3.54	10.980	73.62
3.5	3.64	11.222	74.08
3.6	3.74	11.457	74.50
3.7	3.83	11.664	74.90
3.8	3.92	11.866	75.28
3.9	4.02	12.085	75.62
4.0	4.13	12.319	76.00
4.1	4.22	12.506	76.30
4.2	4.31	12.690	76.60
4.3	4.41	12.889	76.90
4.4	4.51	13.084	77.20
4.5	4.61	13.274	77.48
4.6	4.71	13.460	77.72
4.7	4.81	13.643	78.00
4.8	4.90	13.804	78.23
4.9	5.0	13.979	78.47
5.0	5.1	14.151	78.70
5.1	5.20	14.320	78.90
5.2	5.29	14.469	79.10
5.3	5.39	14.632	79.30
5.4	5.49	14.791	79.50
5.5	5.59	14.948	79.70
5.6	5.68	15.087	79.88
5.7	5.78	15.239	80.05
5.8	5.38	15.388	80.22
5.9	5.98	15.534	80.38
6.0	6.09	15.692	80.55

$1+jwt=AEj\theta$

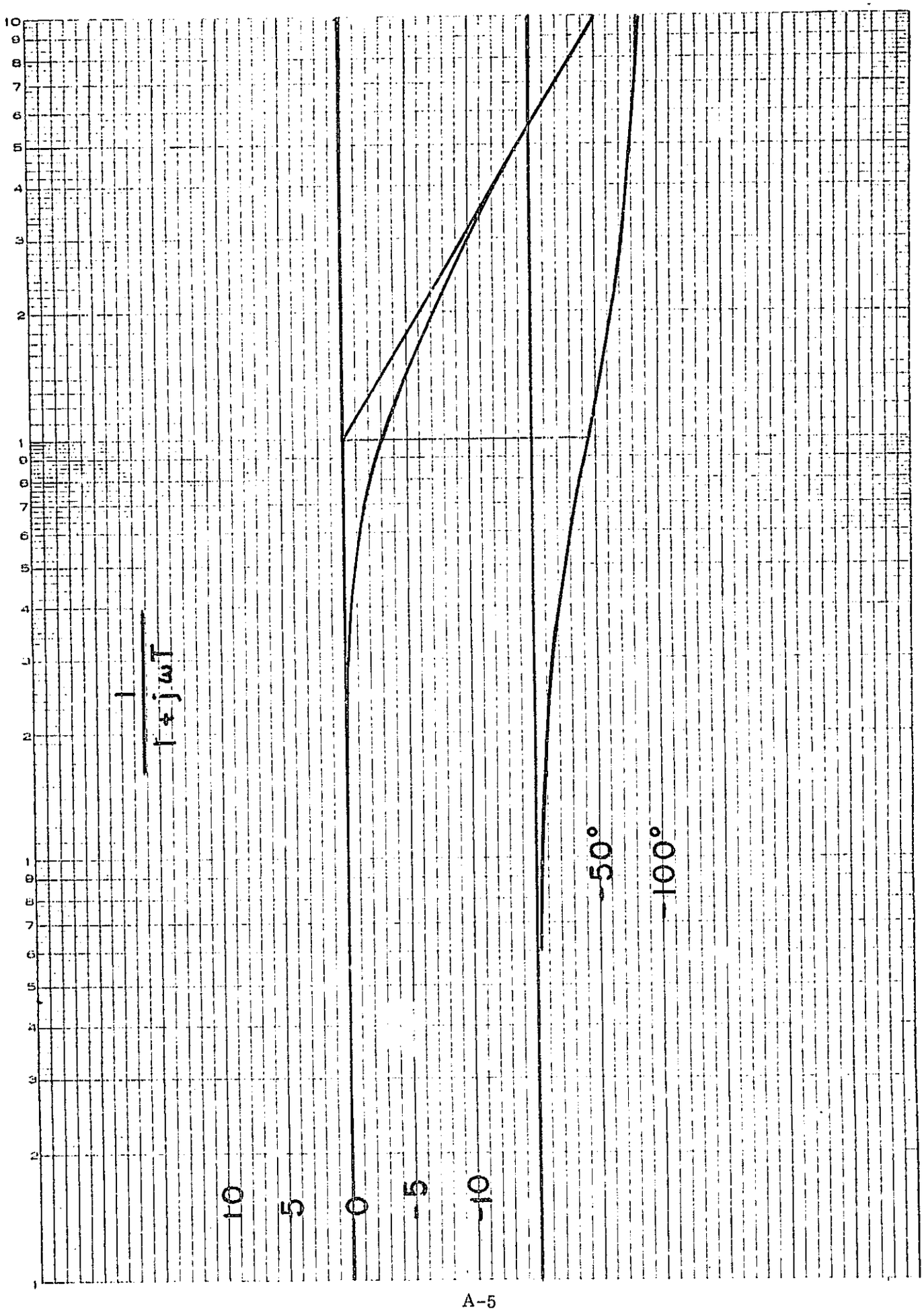
WT	S	20 log A	θ-degrees
6.1	6.19	15.834	80.7
6.2	6.30	15.987	80.88
6.3	6.39	16.110	81.0
6.4	6.48	16.232	81.13
6.5	6.57	16.351	81.25
6.6	6.68	16.496	81.39
6.7	6.77	16.612	81.51
6.8	6.87	16.739	81.64
6.9	6.98	16.852	81.75
7.0	7.07	16.988	81.87
7.1	7.17	17.110	81.99
7.2	7.27	17.231	82.1
7.3	7.38	17.361	82.2
7.4	7.46	17.455	82.31
7.5	7.56	17.570	82.41
7.6	7.66	17.685	82.51
7.7	7.76	17.797	82.6
7.8	7.86	17.908	82.7
7.9	7.96	18.018	82.79
8.0	8.06	18.127	82.88
8.1	8.16	18.234	82.96
8.2	8.26	18.340	83.05
8.3	8.37	18.455	83.14
8.4	8.45	18.537	83.21
8.5	8.55	18.639	83.29
8.6	8.65	18.740	83.37
8.7	8.76	18.850	83.45
8.8	8.85	18.939	83.52
8.9	8.95	19.036	83.59
9.0	9.05	19.133	83.66

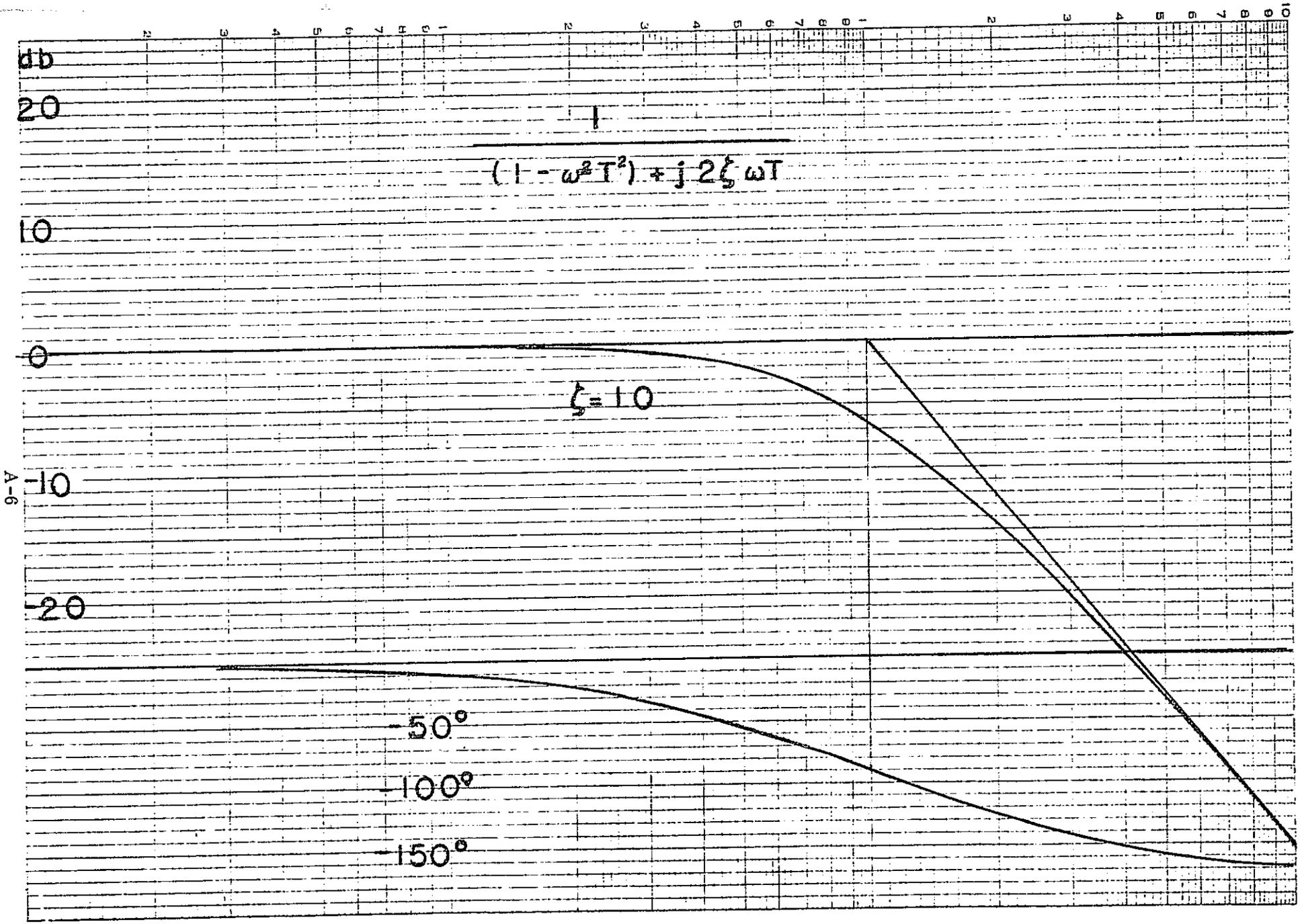
$1+WT=AEj\theta$

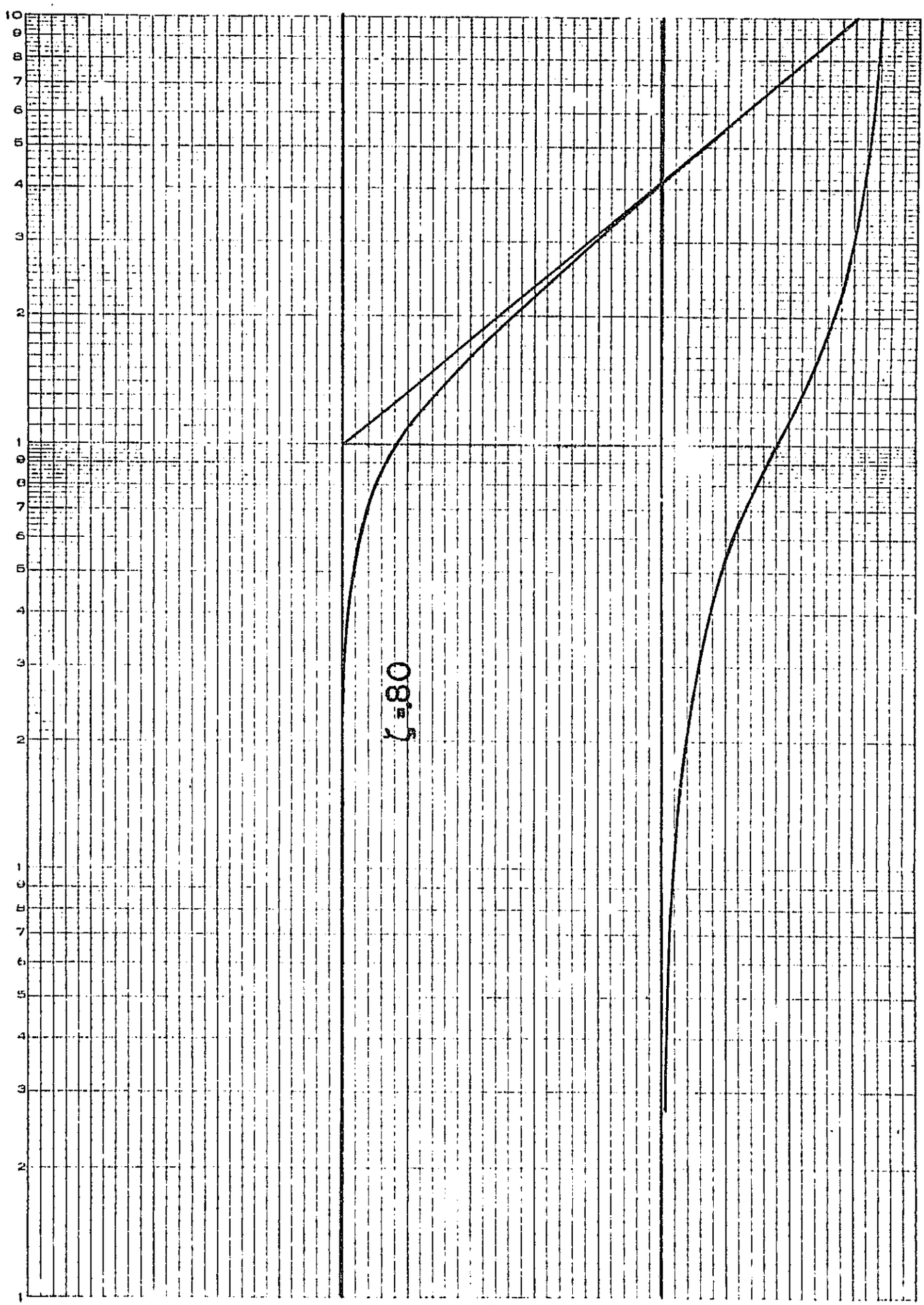
WT	A	20 log A	θ-degrees
9.1	9.15	19.094	83.73
9.2	9.26	19.332	83.8
9.3	9.35	19.416	83.86
9.4	9.45	19.509	83.93
9.5	9.56	19.609	84.0
9.6	9.65	19.691	84.05
9.7	9.74	19.771	84.11
9.8	9.85	19.869	84.17
9.9	9.97	19.974	84.24
10.0	10.05	20.04	84.29
12	12.04	21.584	84.23
14	14.04	22.923	85.83
16	16.03	24.082	86.42
18	18.03	25.105	86.81
20	20.02	26.021	87.16
25	25.02	27.959	87.71
30	30.02	29.542	88.69
35	35	30.881	88.36
40	40	32.041	88.58
45	45	33.064	88.73
50	50	33.979	88.85
00	100	40	89.4

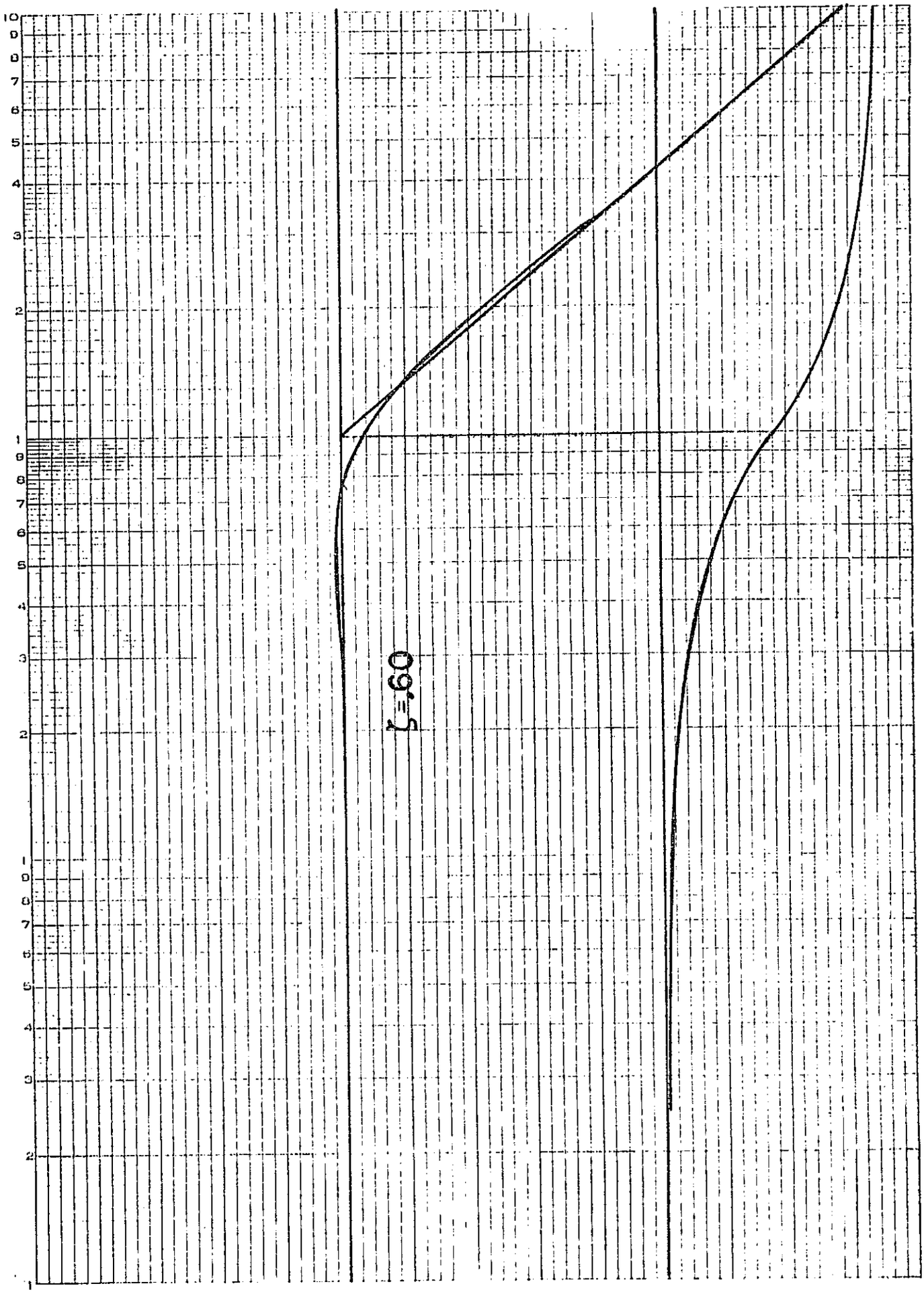
NOTE: $\frac{1}{1+jwt} = \frac{1}{A}E^{-j\theta}$

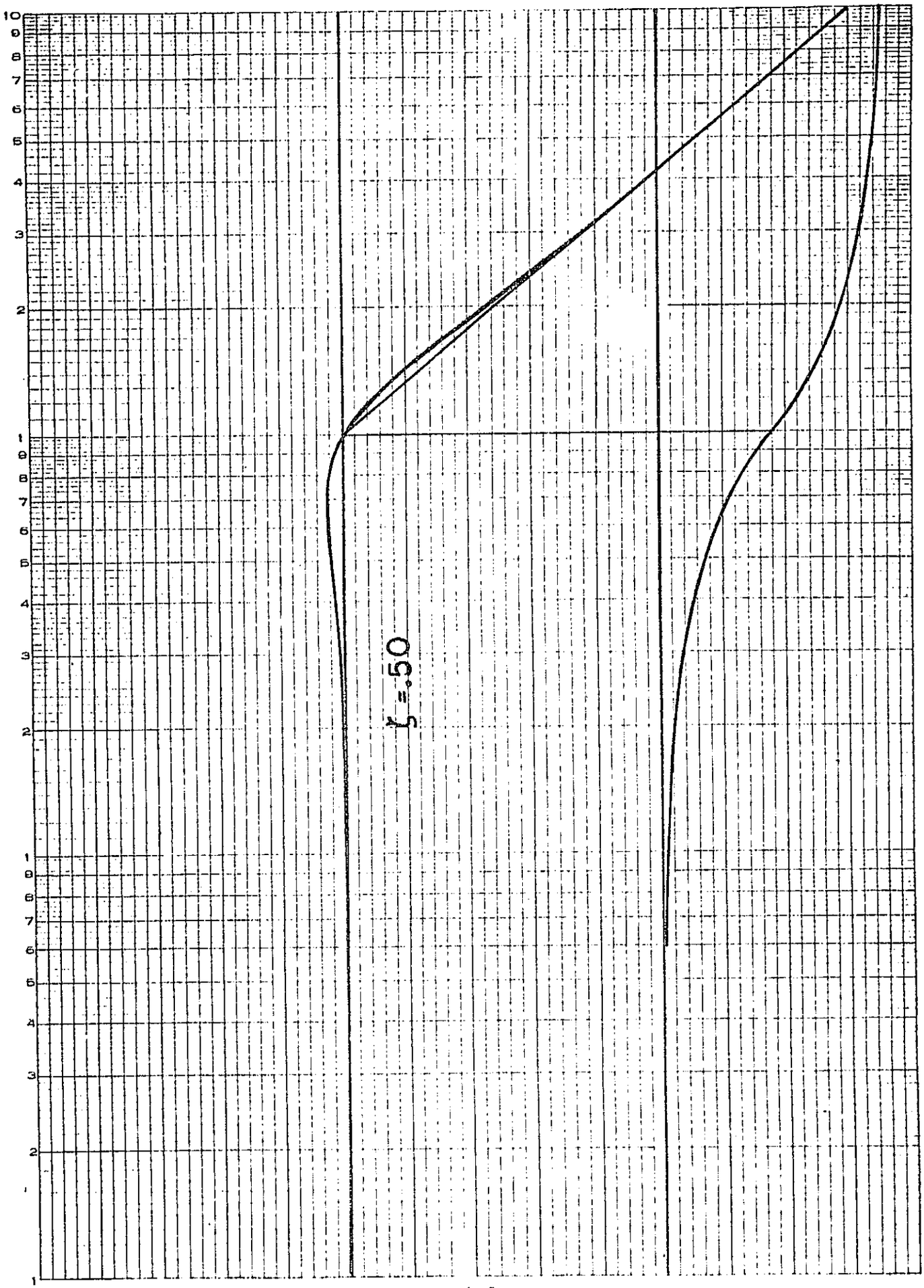
$20 \log \frac{1}{A} = -20 \log A$

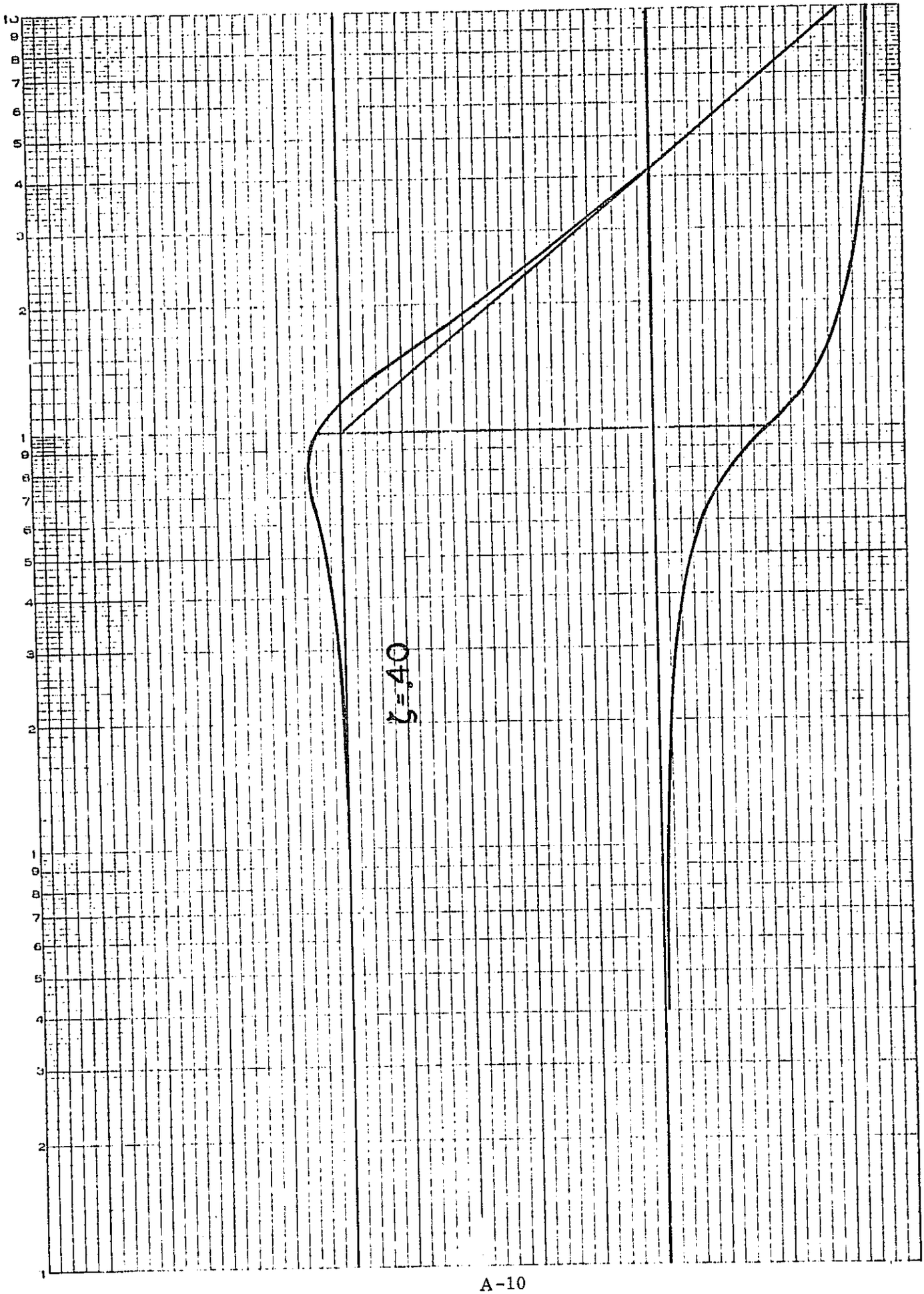

$\frac{1}{1+j\omega T}$
10
5
0
-5
-10
-50°
-100°

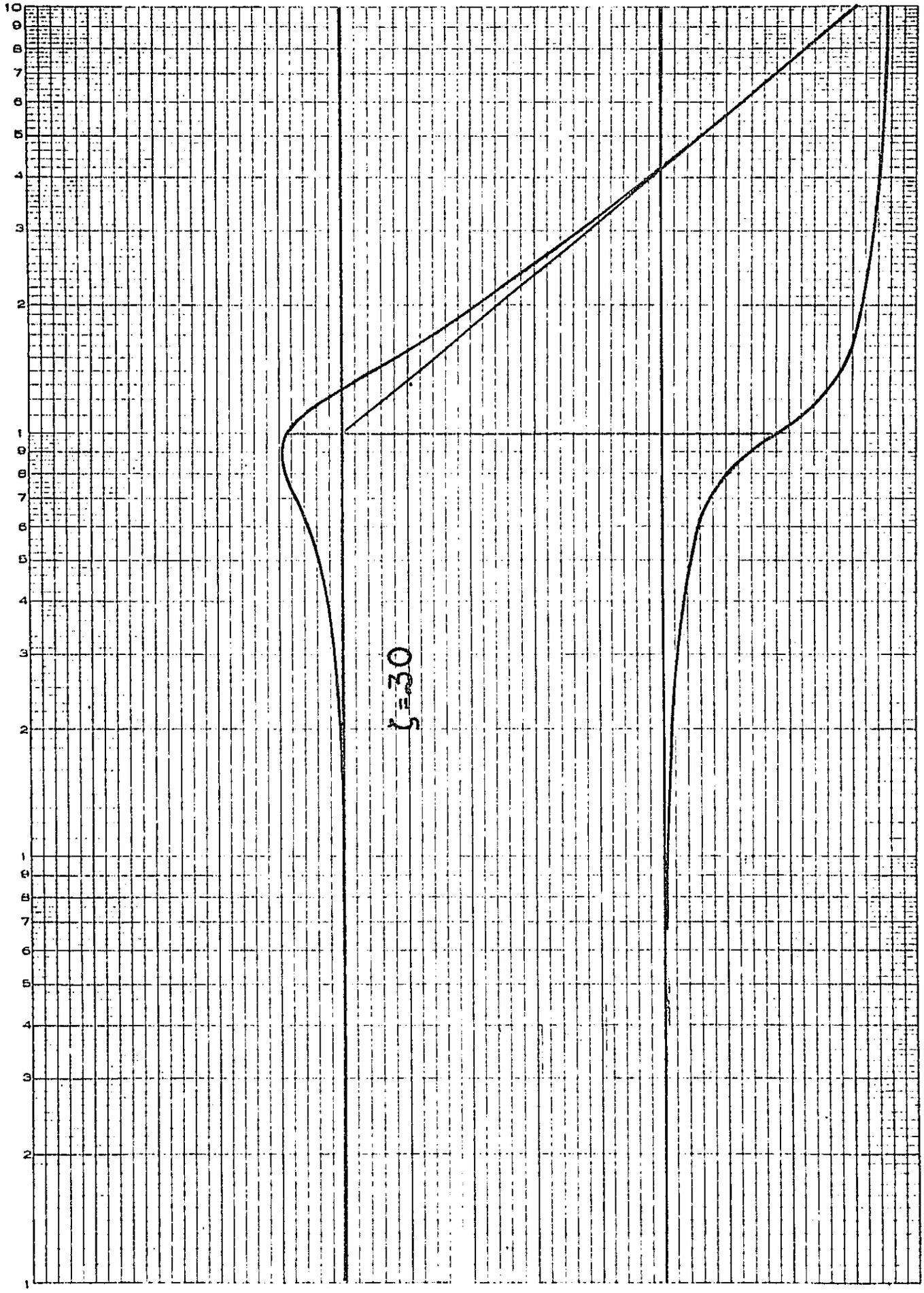

db
20
10
0
-10
-20
$\frac{1}{(1-\omega^2T^2)+j2\zeta\omega T}$
$\zeta = 10$
-50°
-100°
-150°

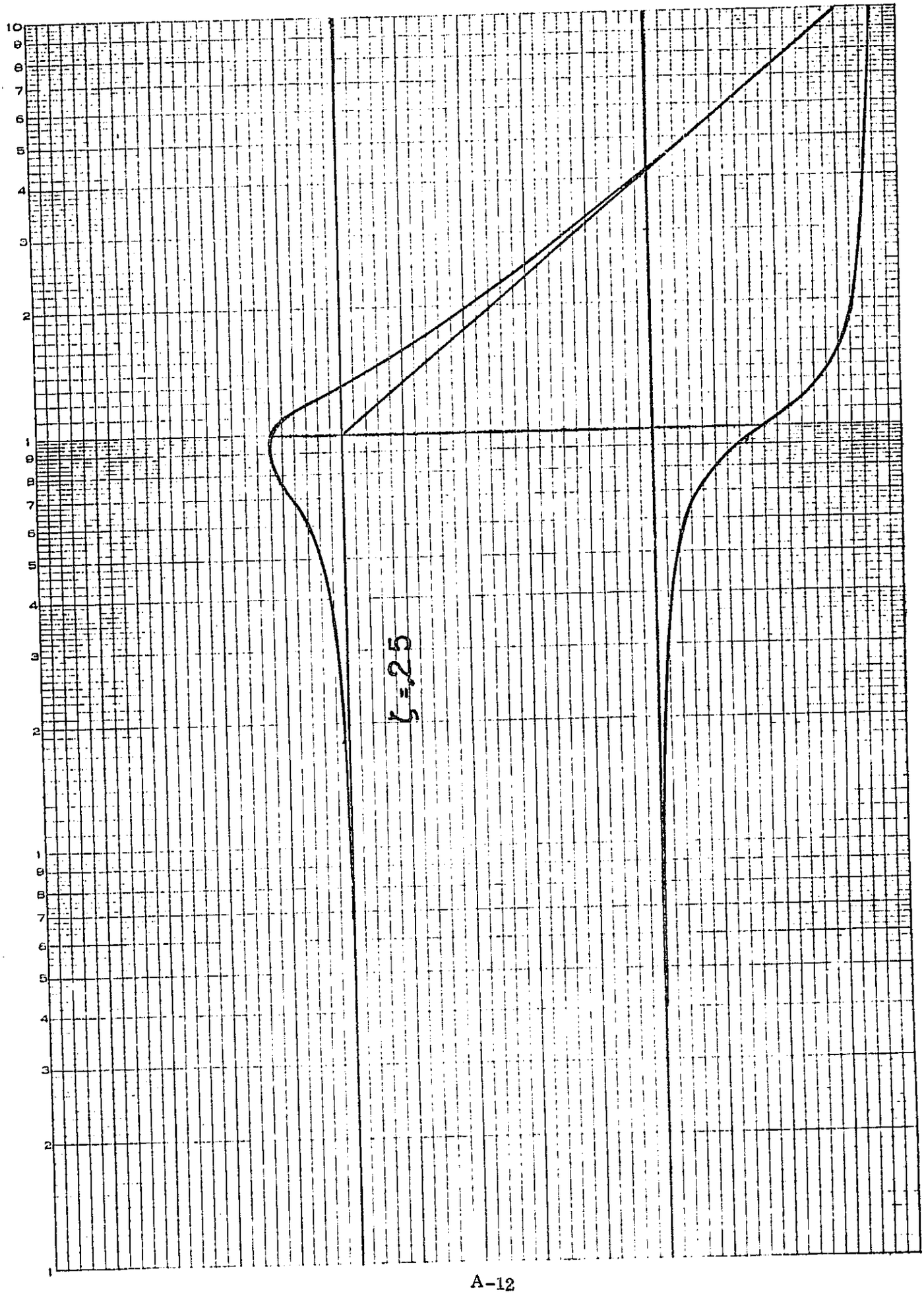

$\zeta_s = 8.0$

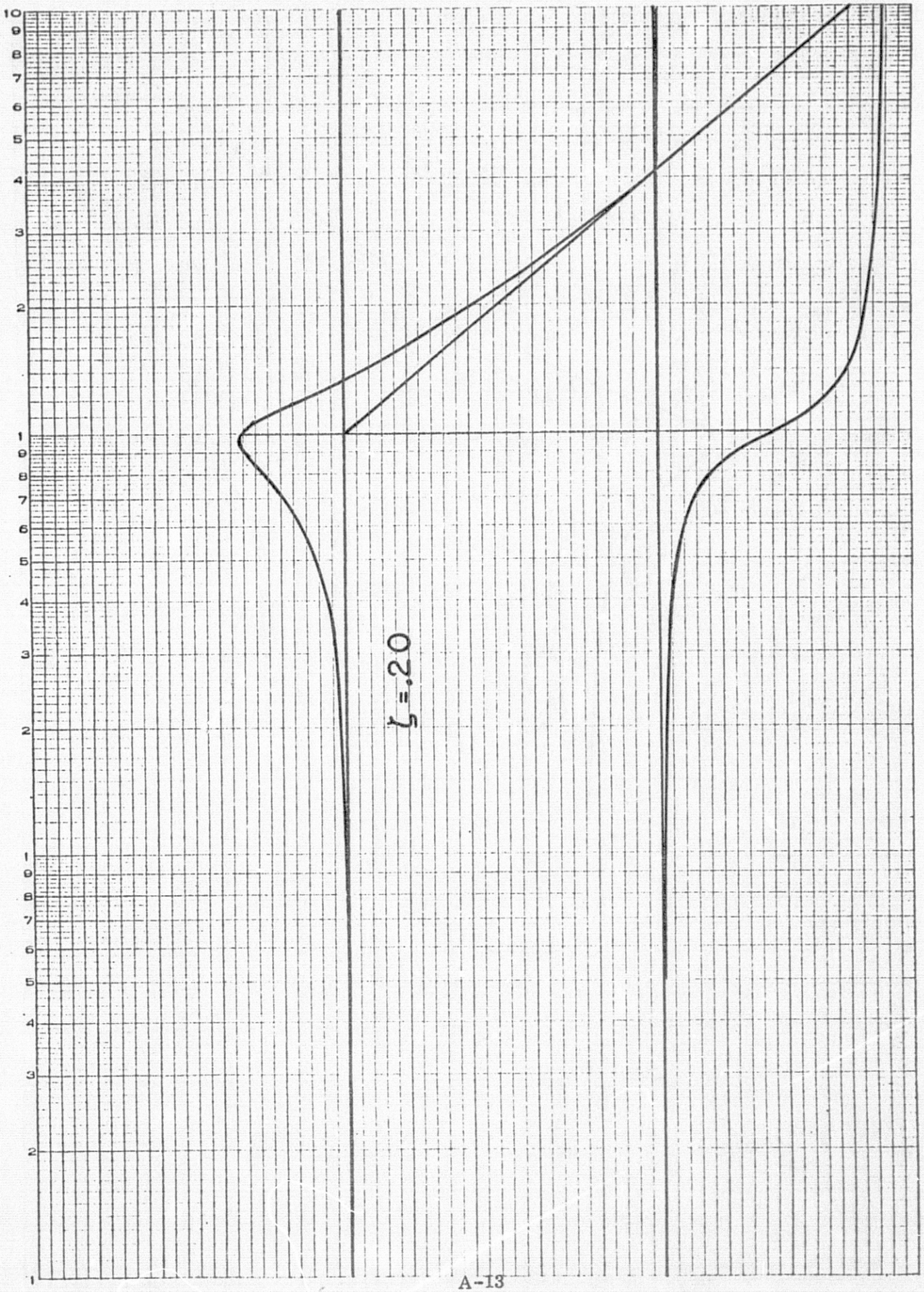

ζ=60

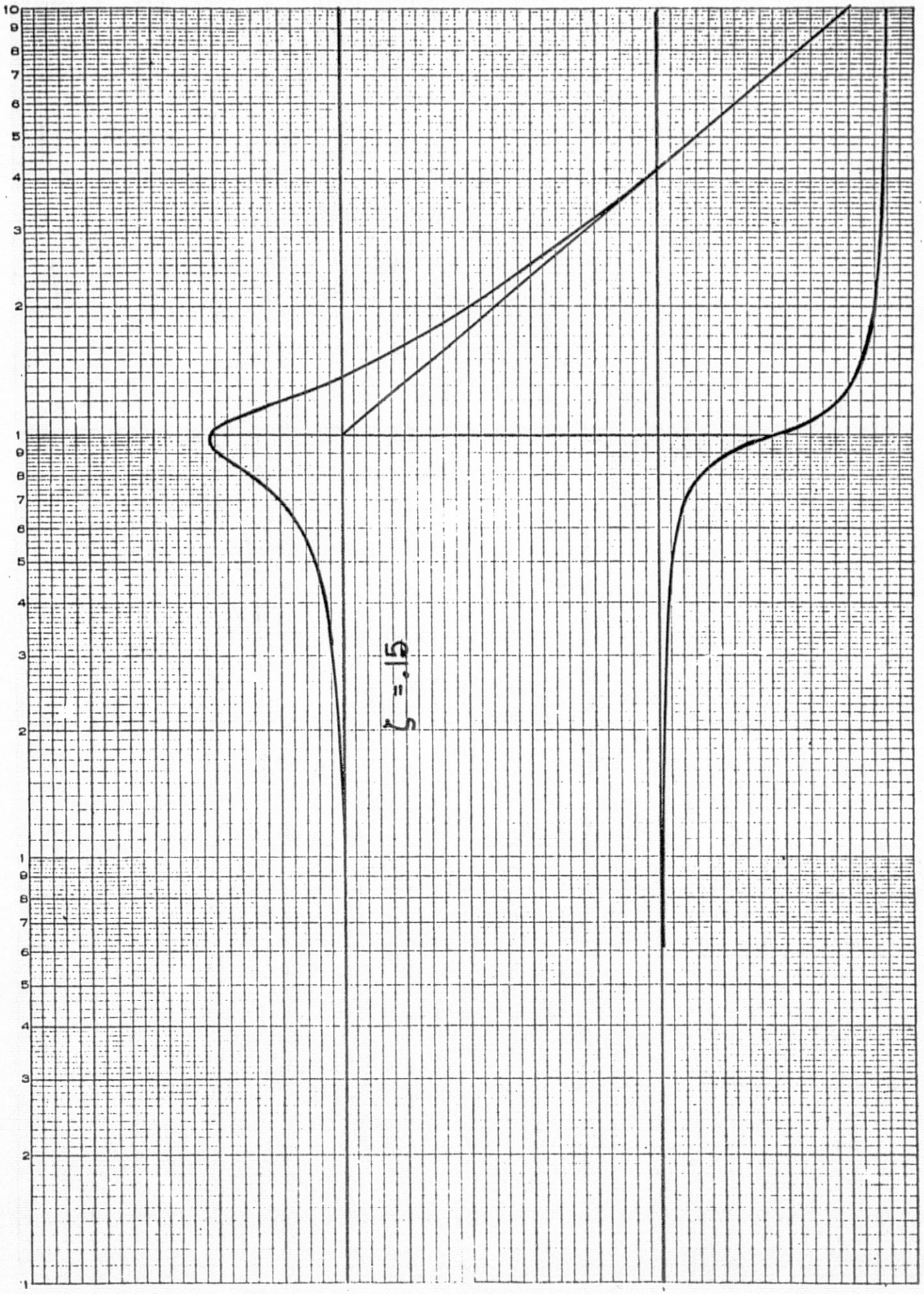

ζ = .50

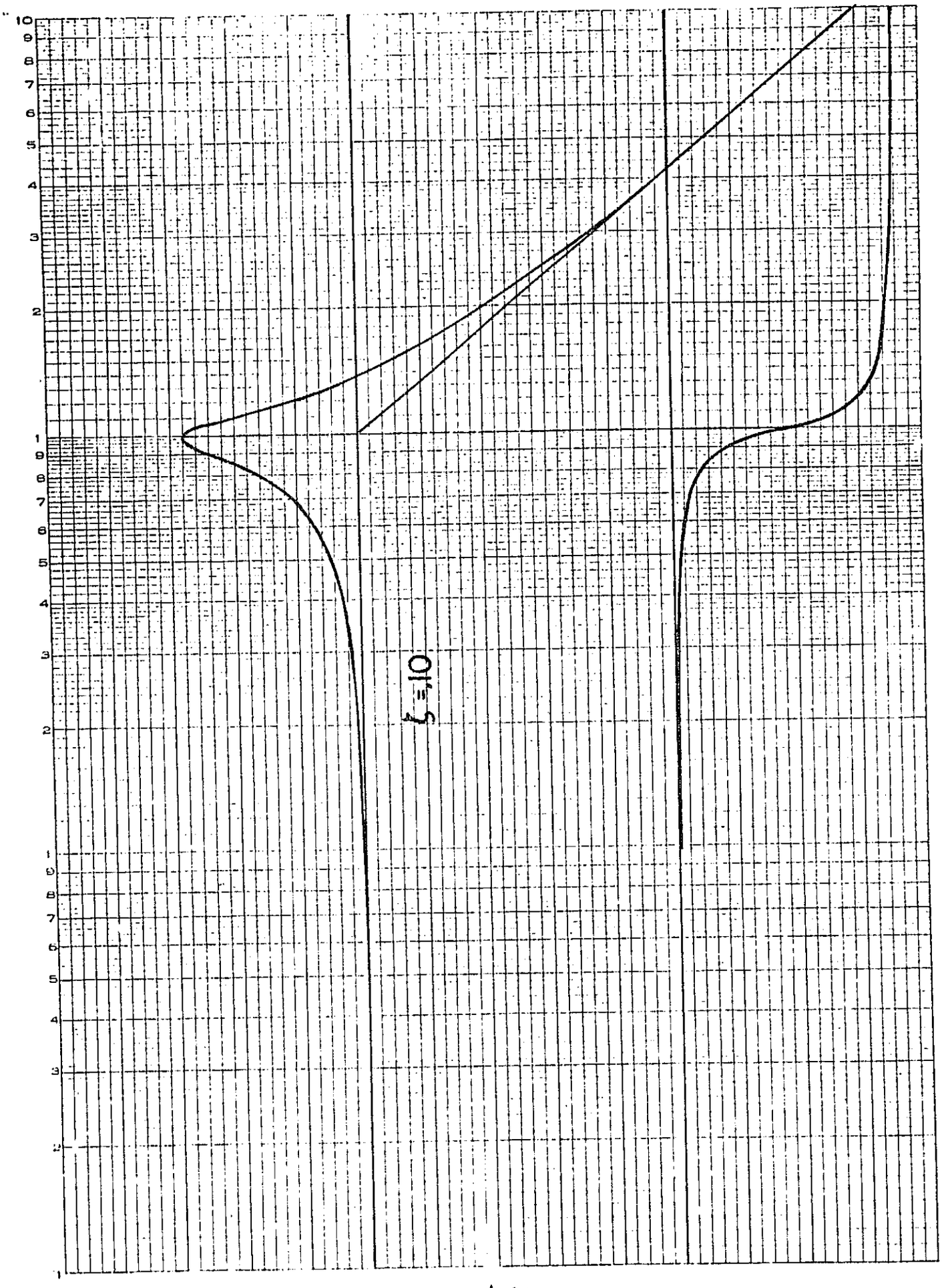

ζ=.40

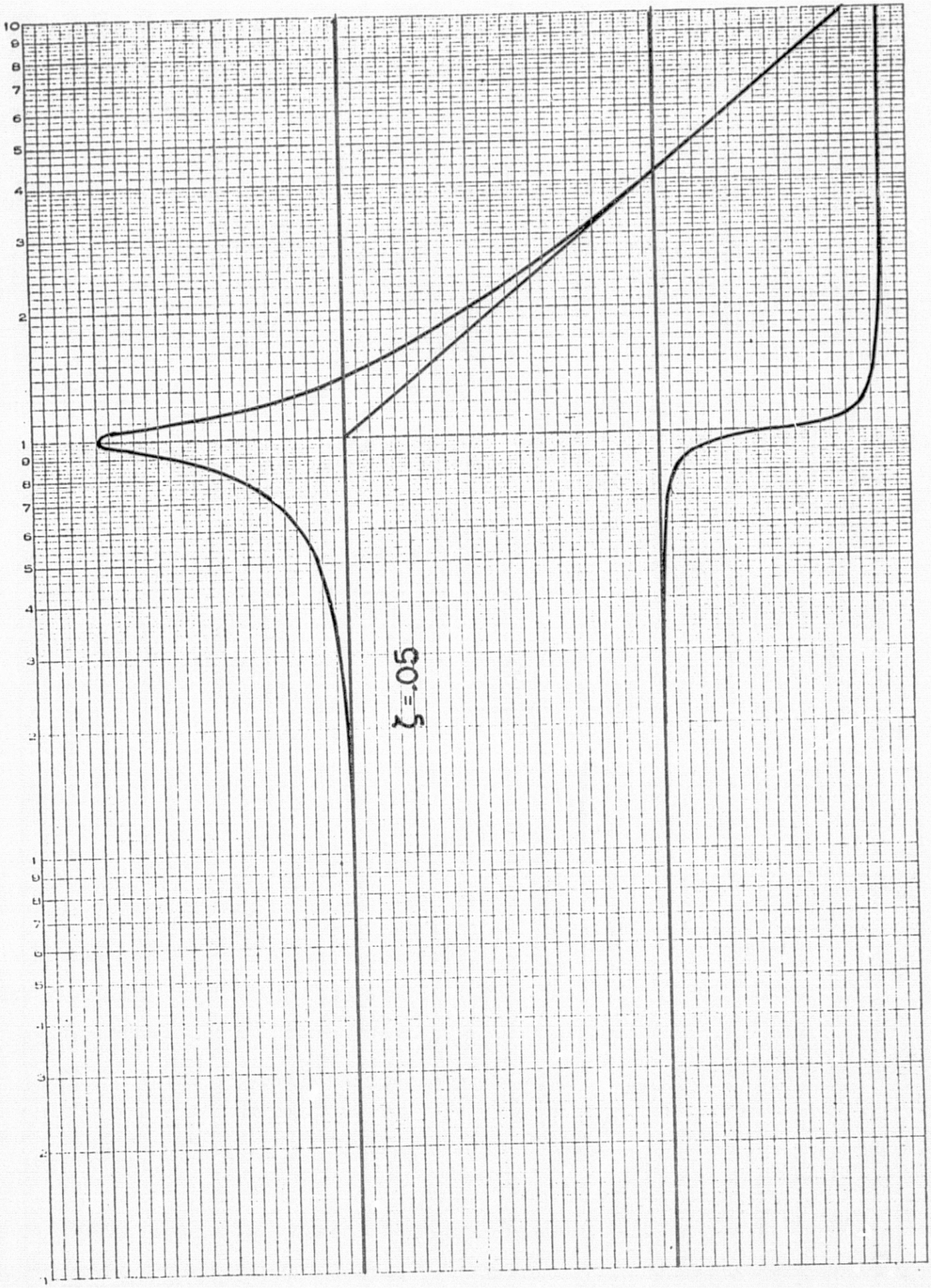

10
9
8
7
6
5
4
3
2
1
9
8
7
6
5
4
3
2
1
9
8
7
6
5
4
3
2
1
ζ=30


$\zeta = .25$

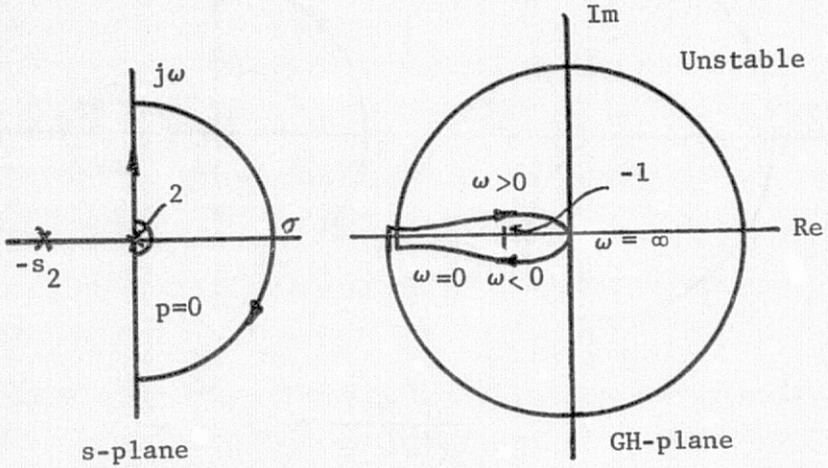

10
9
8
7
6
5
4
3
2
1
9
8
7
6
5
4
3
2
1
9
8
7
6
5
4
3
2
1
ζ=.20


$\zeta = .15$

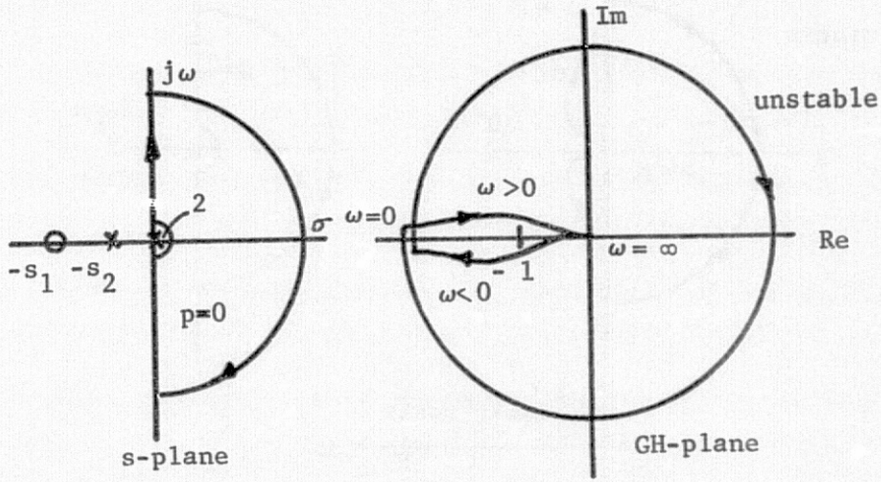

ζ = .10


ζ = .05

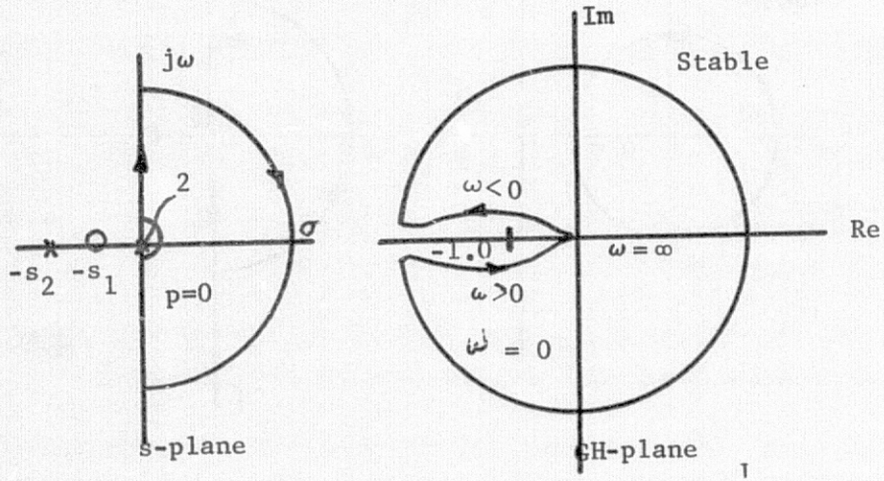
POLAR PLOTS OF SOME COMMON OPEN-LOOP TRANSFER FUNCTIONS



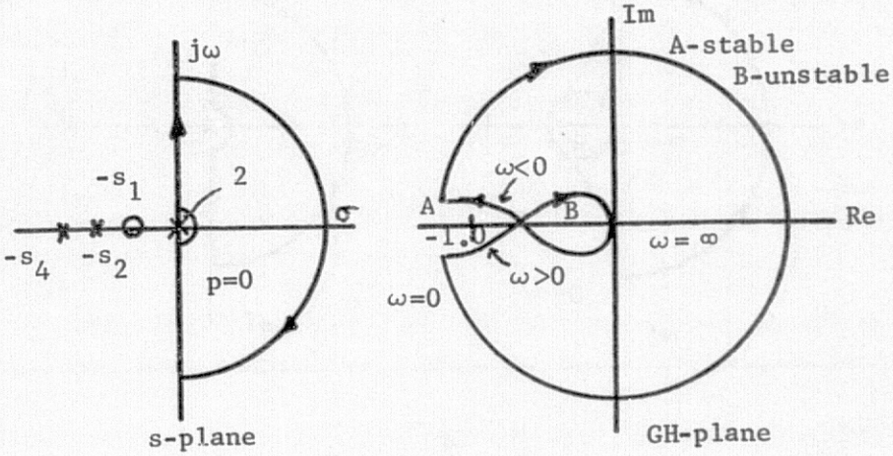
$$GH = \frac{K}{s^2(s+s_2)}$$



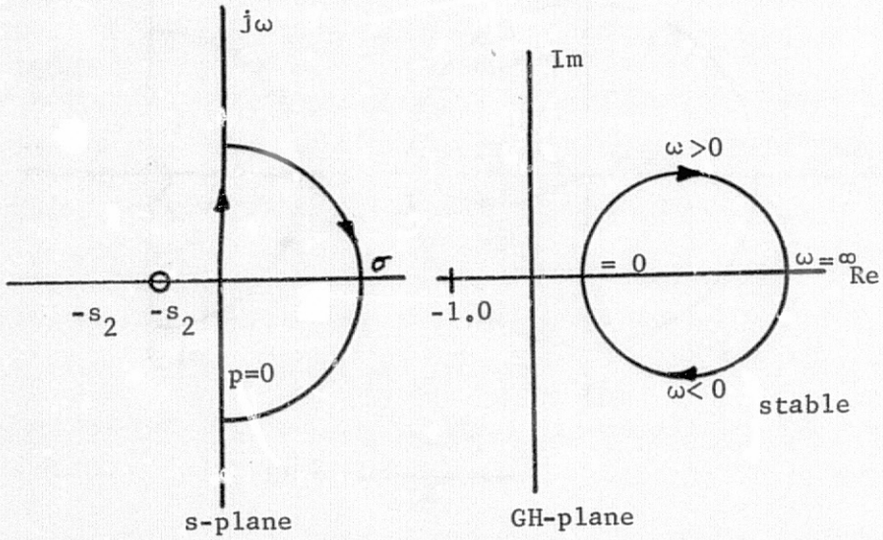
$$GH = \frac{K(s+s_1)}{s^2(s+s_2)}$$



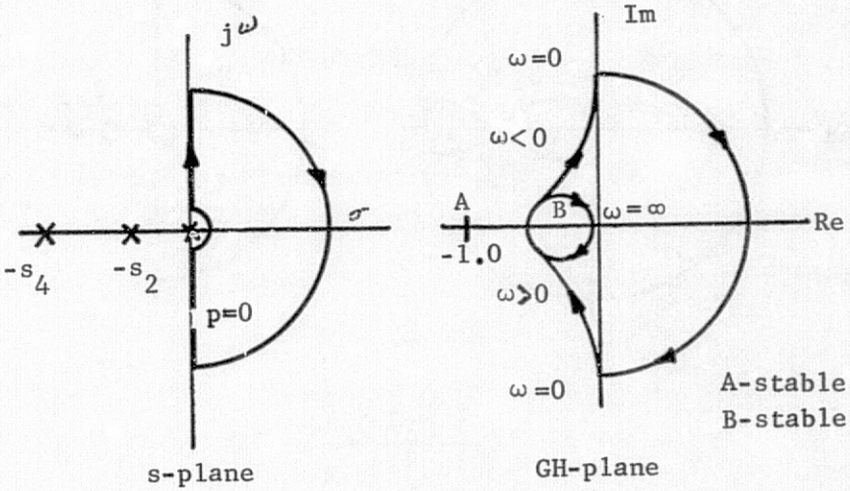
$$GH = \frac{K(s+s_1)}{s^2(s+s_2)}$$



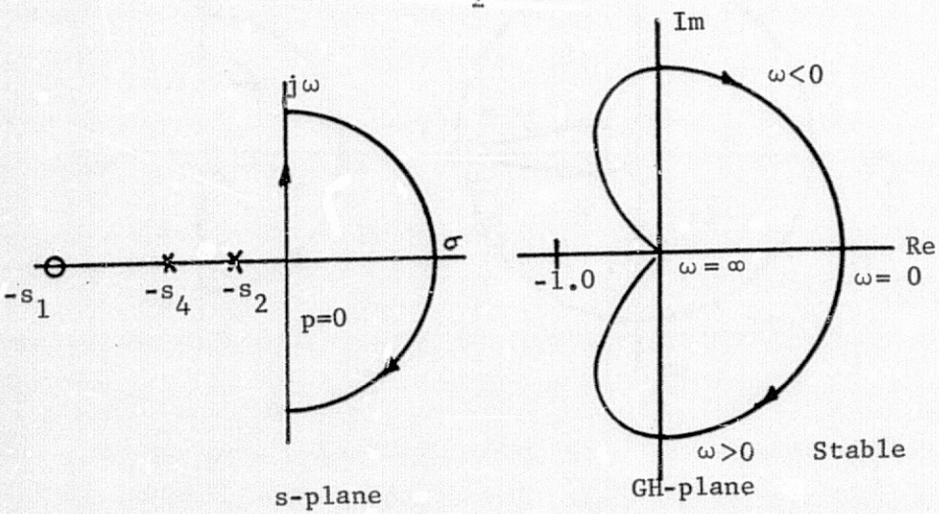
$$GH = \frac{K(s+s_1)}{s^2(s+s_2)(s+s_4)}$$



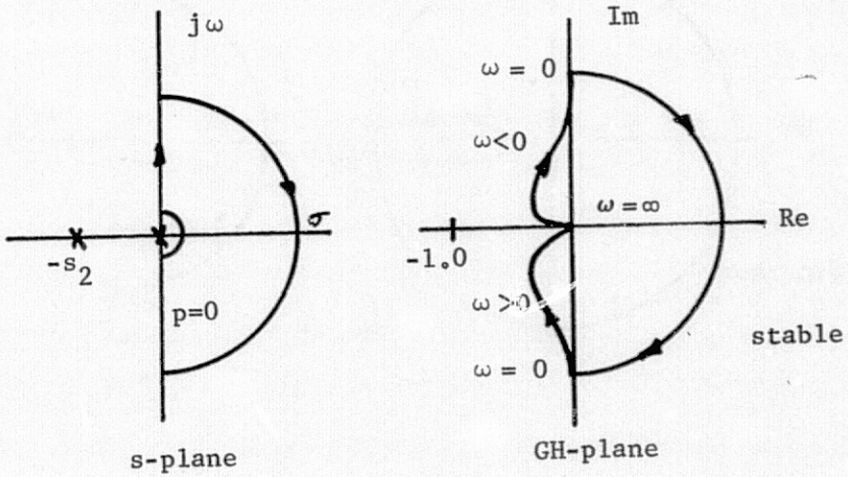
$$GH = \frac{K(s+s_1)}{(s+s_2)}$$



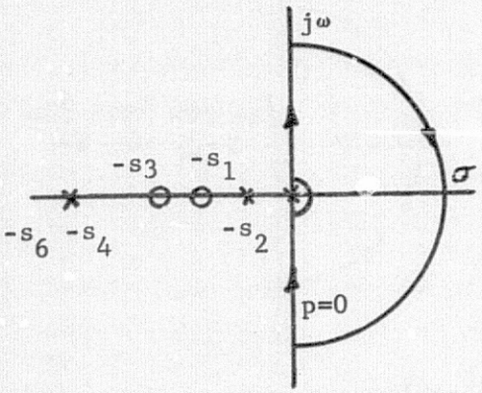
$$GH = \frac{K}{s(s+s_2)(s+s_4)}$$



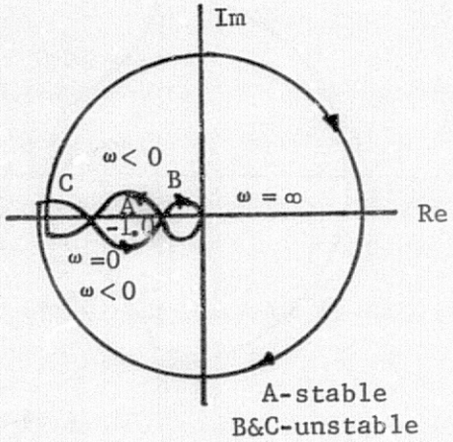
$$GH = \frac{K(s+s_1)}{(s+s_2)(s+s_4)}$$



$$GH = \frac{K}{s(s+s_2)}$$

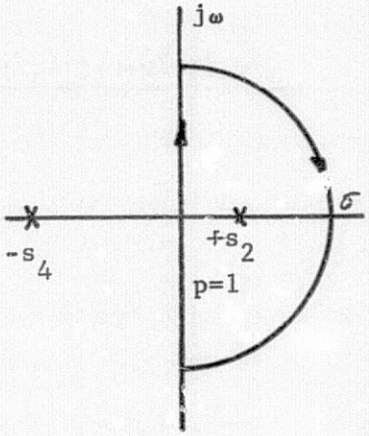


s-plane

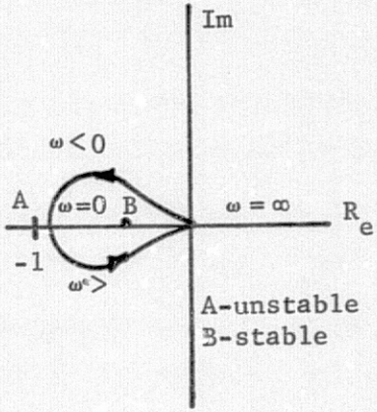


GH-plane

$$GH = \frac{K(s+s_1)(s+s_3)}{s^2(s+s_2)(s+s_4)(s+s_6)}$$

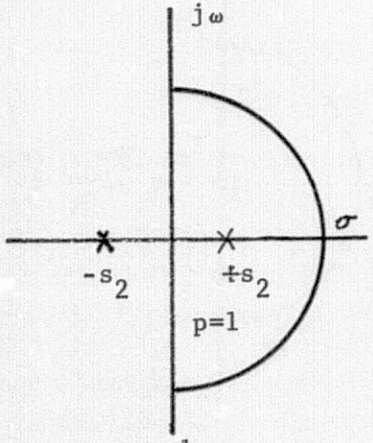


s-plane

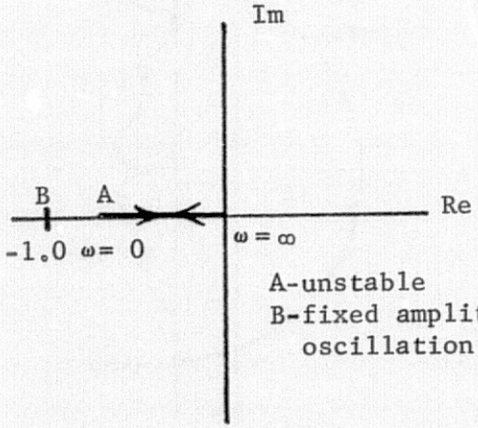


GH-plane

$$GH = \frac{K}{(s+s_2)(s+s_4)}$$

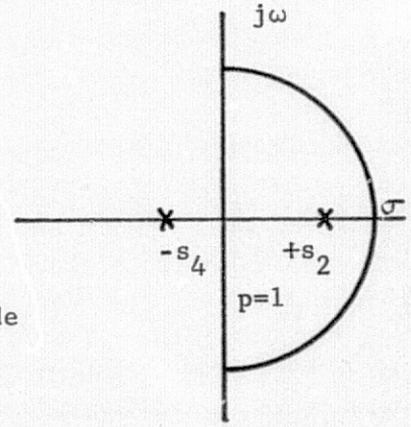


s-plane

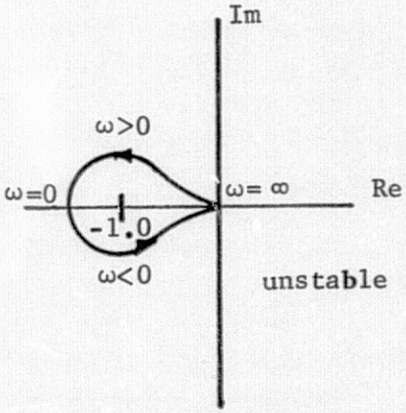


GH-plane

$$GH \frac{K}{(s+s_2)(s-s_2)}$$

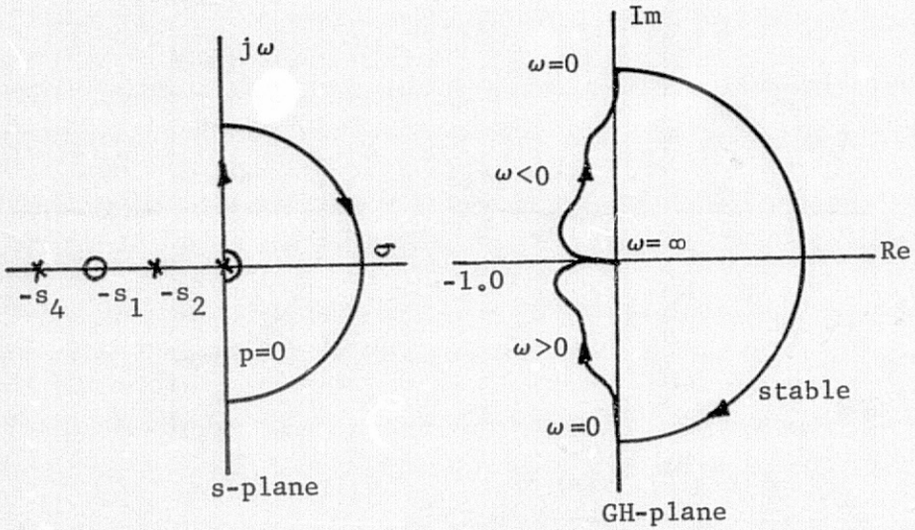


s-plane

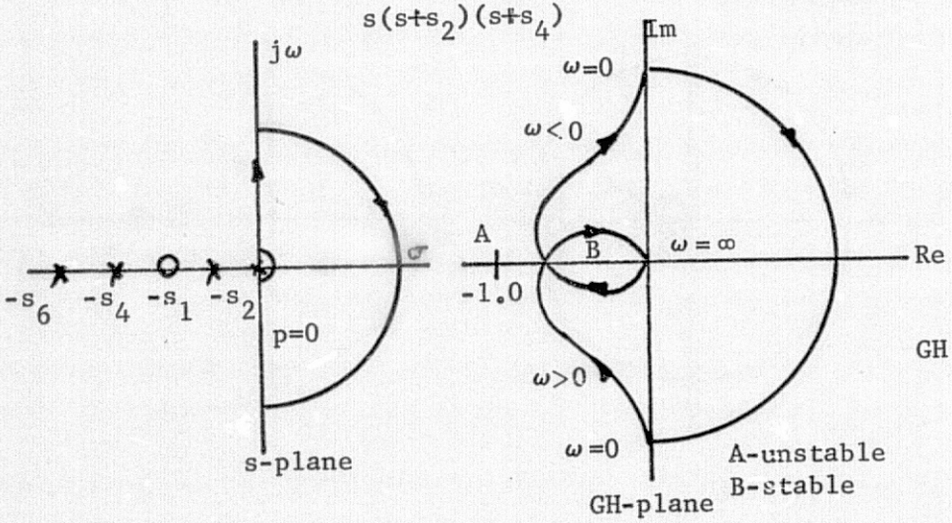


GH-plane

$$GH = \frac{K}{(s+s_2)(s+s_4)}$$



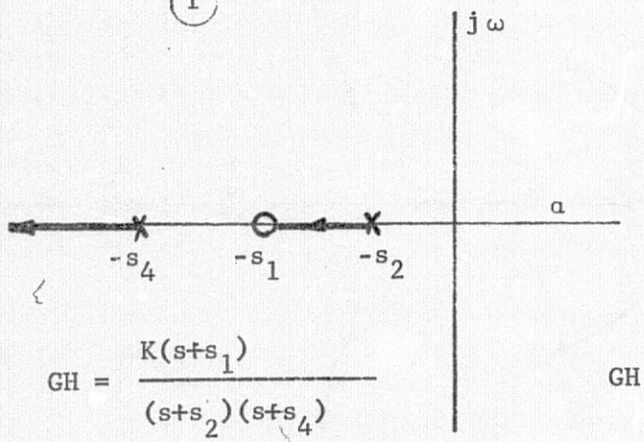
$$GH = \frac{K(s+s_1)}{s(s+s_2)(s+s_4)}$$



$$GH = \frac{K(s+s_1)}{s(s+s_2)(s+s_4)(s+s_6)}$$

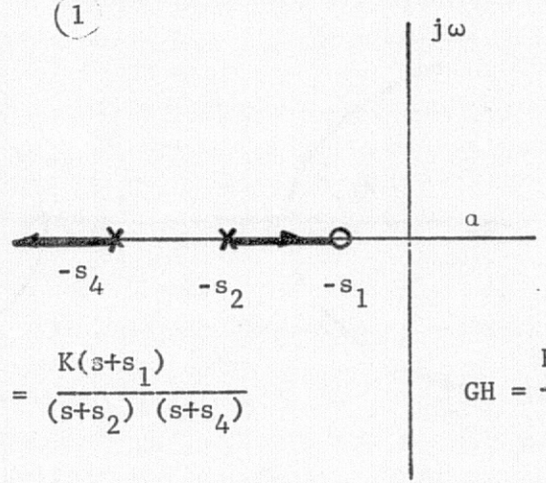
PLOTS OF COMMON ROOT LOCI

1



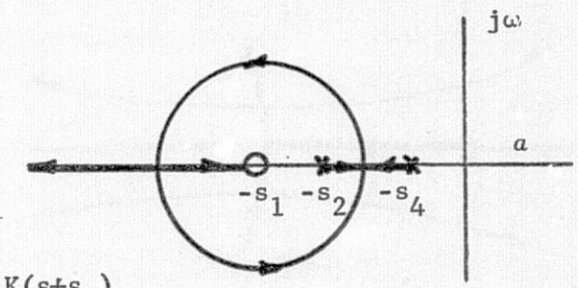
$$GH = \frac{K(s+s_1)}{(s+s_2)(s+s_4)}$$

1



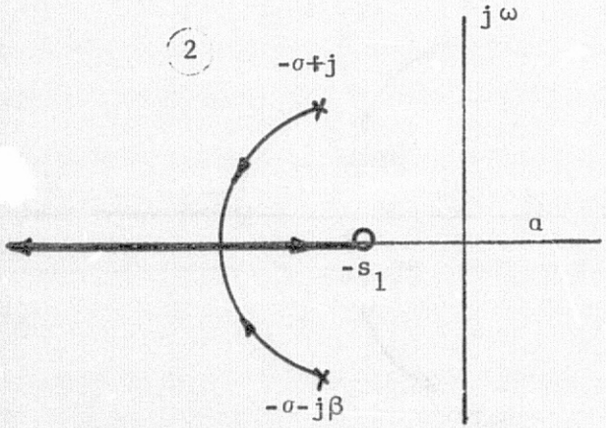
$$GH = \frac{K(s+s_1)}{(s+s_2)\ (s+s_4)}$$

1



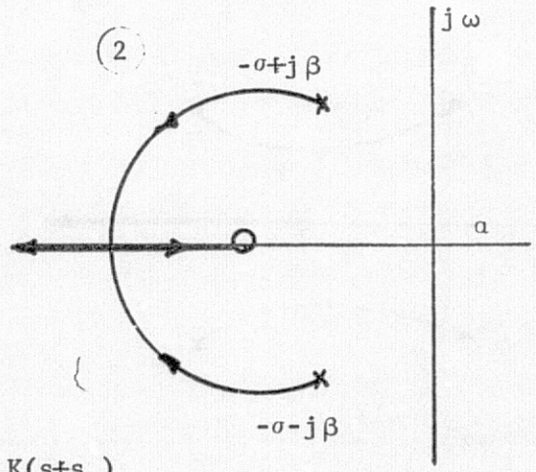
$$GH = \frac{K(s+s_1)}{(s+s_2)(s+s_4)}$$

2



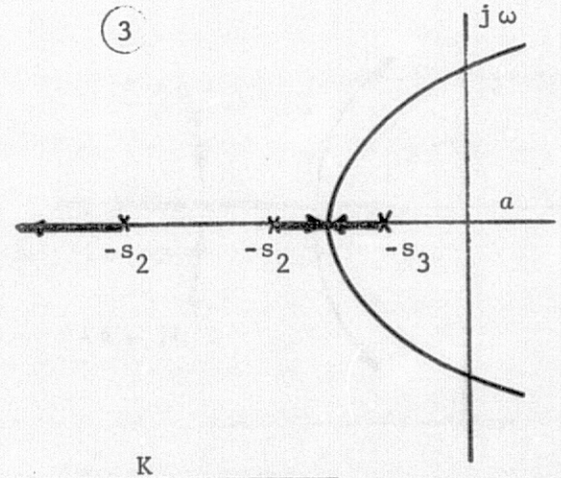
$$GH = \frac{K(s+s_1)}{(s+\sigma+j\beta\ (s+\sigma-j\beta)}$$

2



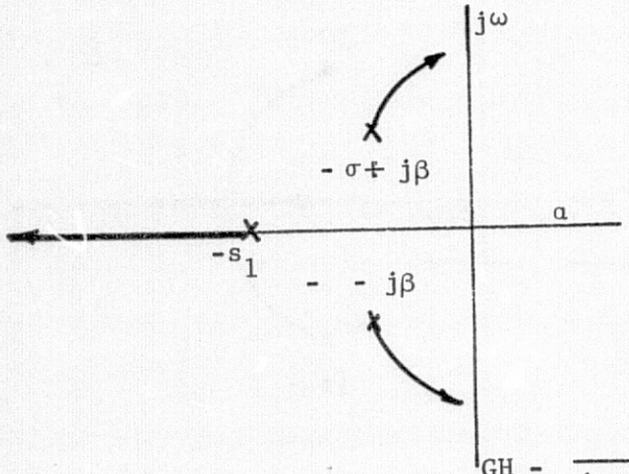
$$GH = \frac{K(s+s_1)}{(s+\sigma+j\beta)(s+\sigma-j\beta)}$$

3

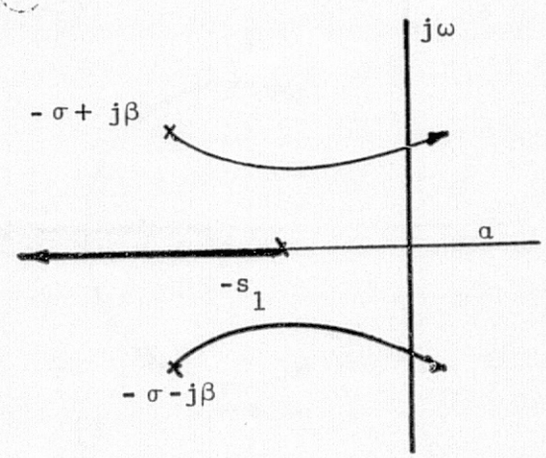


$$GH = \frac{K}{(s+s_1)(s+s_2)(s+s_3)}$$

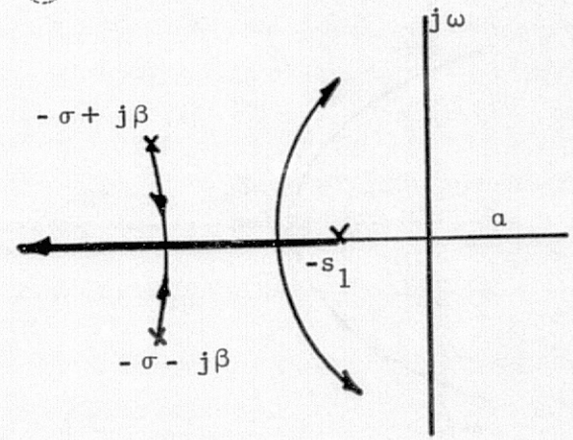
4



4

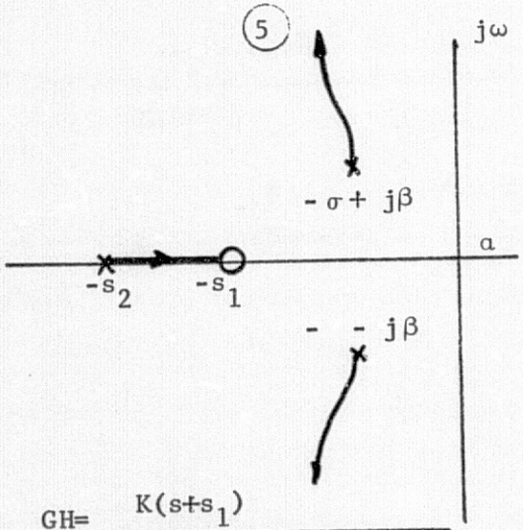


4

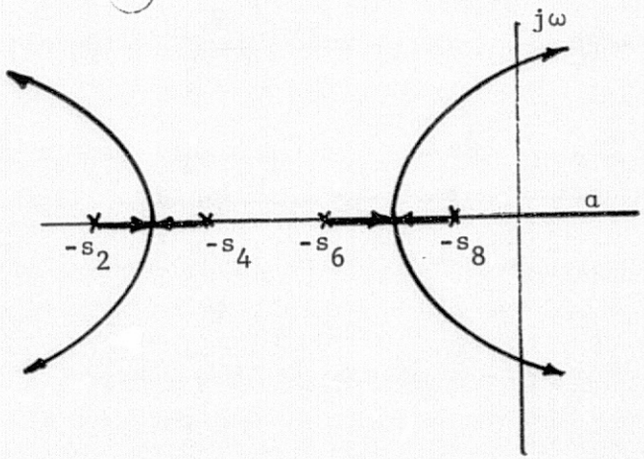


$$GH - \frac{K}{(s+s_1)(s+\quad + j\ )\ (s+\quad - j\ )}$$

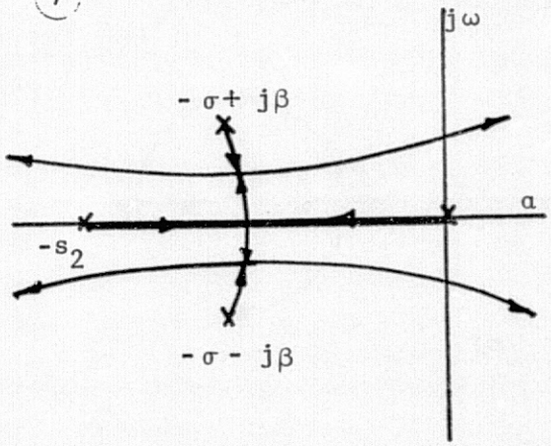
5



6



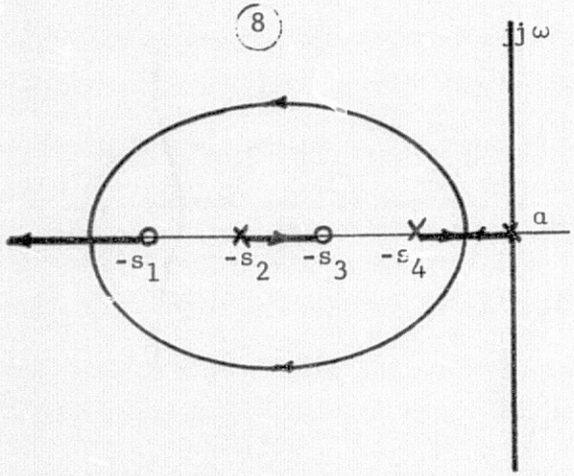
7

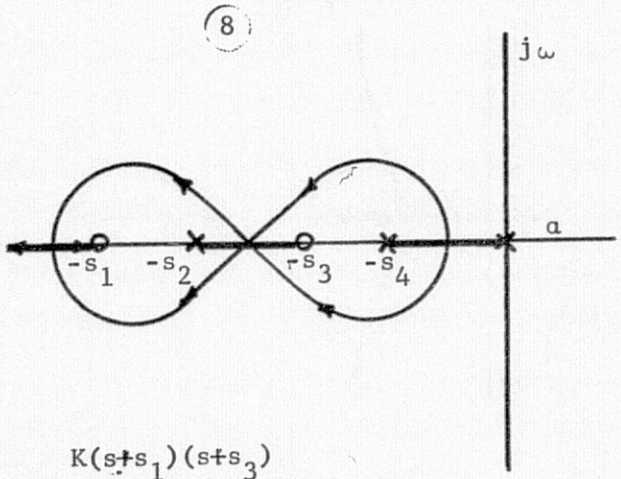


$$GH = \frac{K(s+s_1)}{(s+s_2)(s+\sigma+ j\beta)(s+\sigma-j\beta)}$$

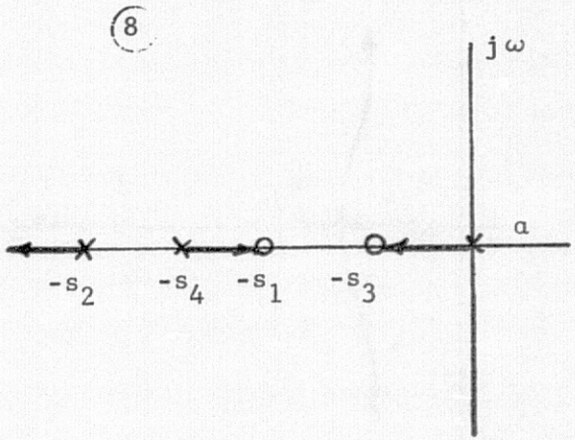
$$GH = \frac{K}{(s+s_2)(s+s_4)(s+s_6)(s+s_8)}$$

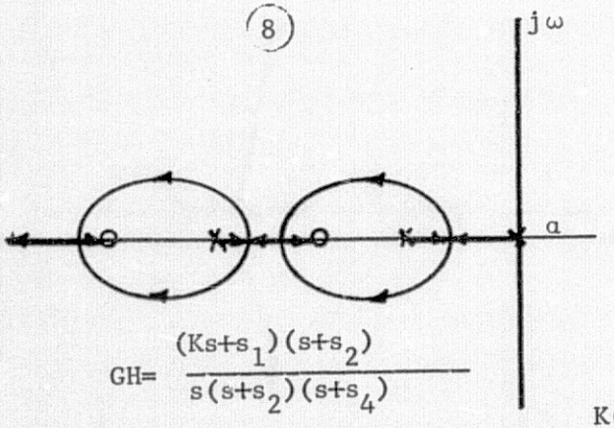
$$GH = \frac{K}{s(s+s_2)(s+\sigma j\beta)(s+\sigma j\ \beta)}$$



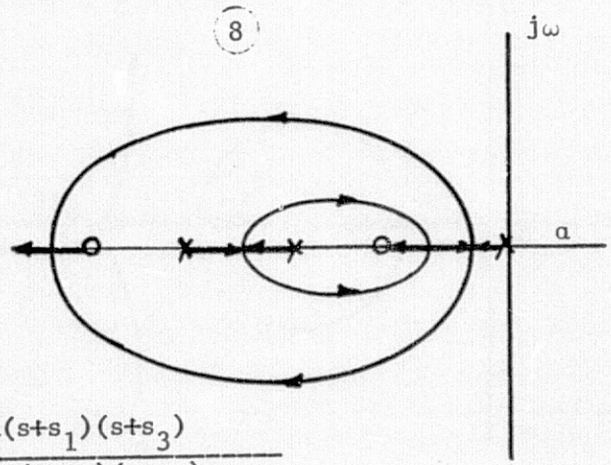


$$GH = \frac{K(s+s_1)(s+s_3)}{s(s+s_2)(s+s_4)}$$

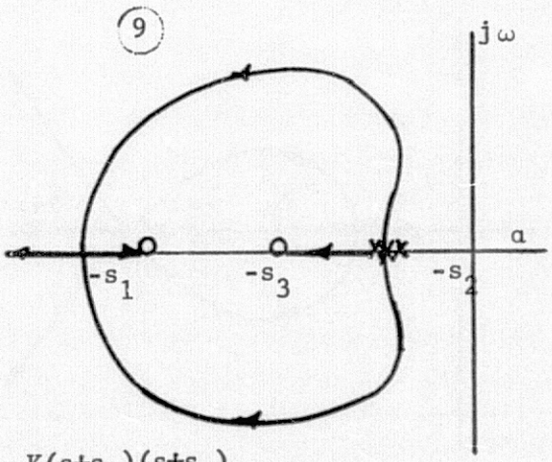




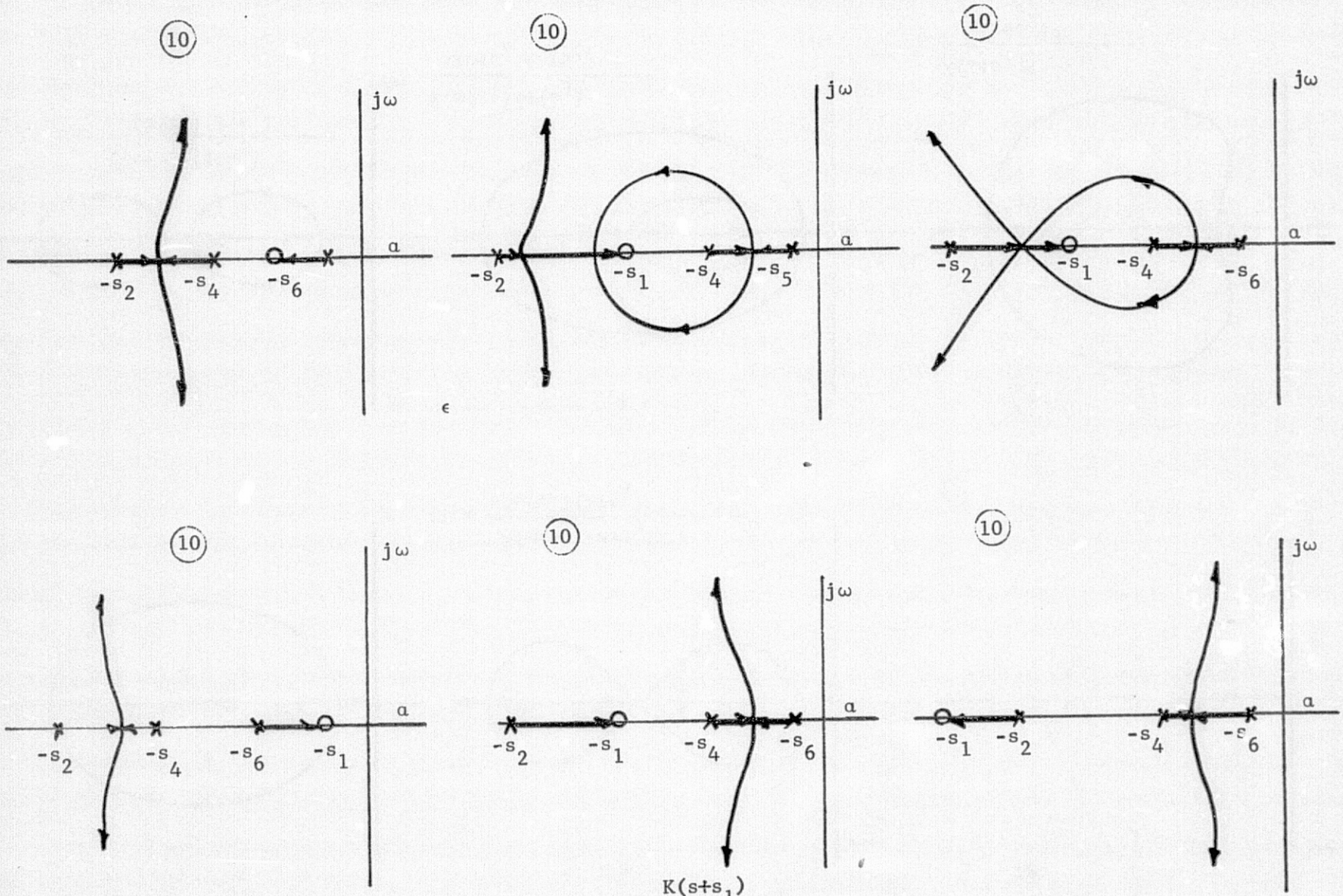
$$GH = \frac{(Ks+s_1)(s+s_2)}{s(s+s_2)(s+s_4)}$$



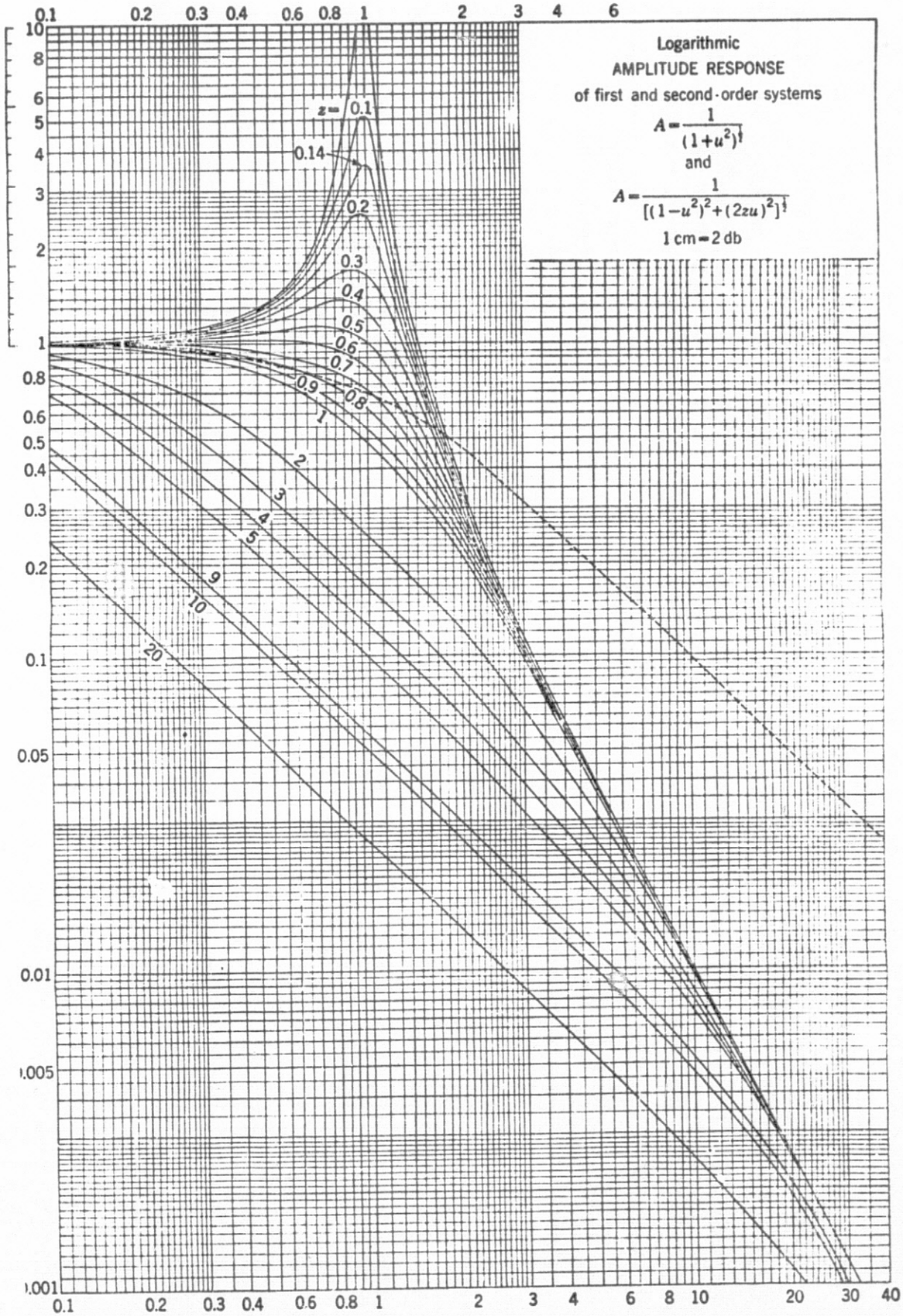
$$GH = \frac{K(s+s_1)(s+s_3)}{s(s+s_2)(s+s_4)}$$

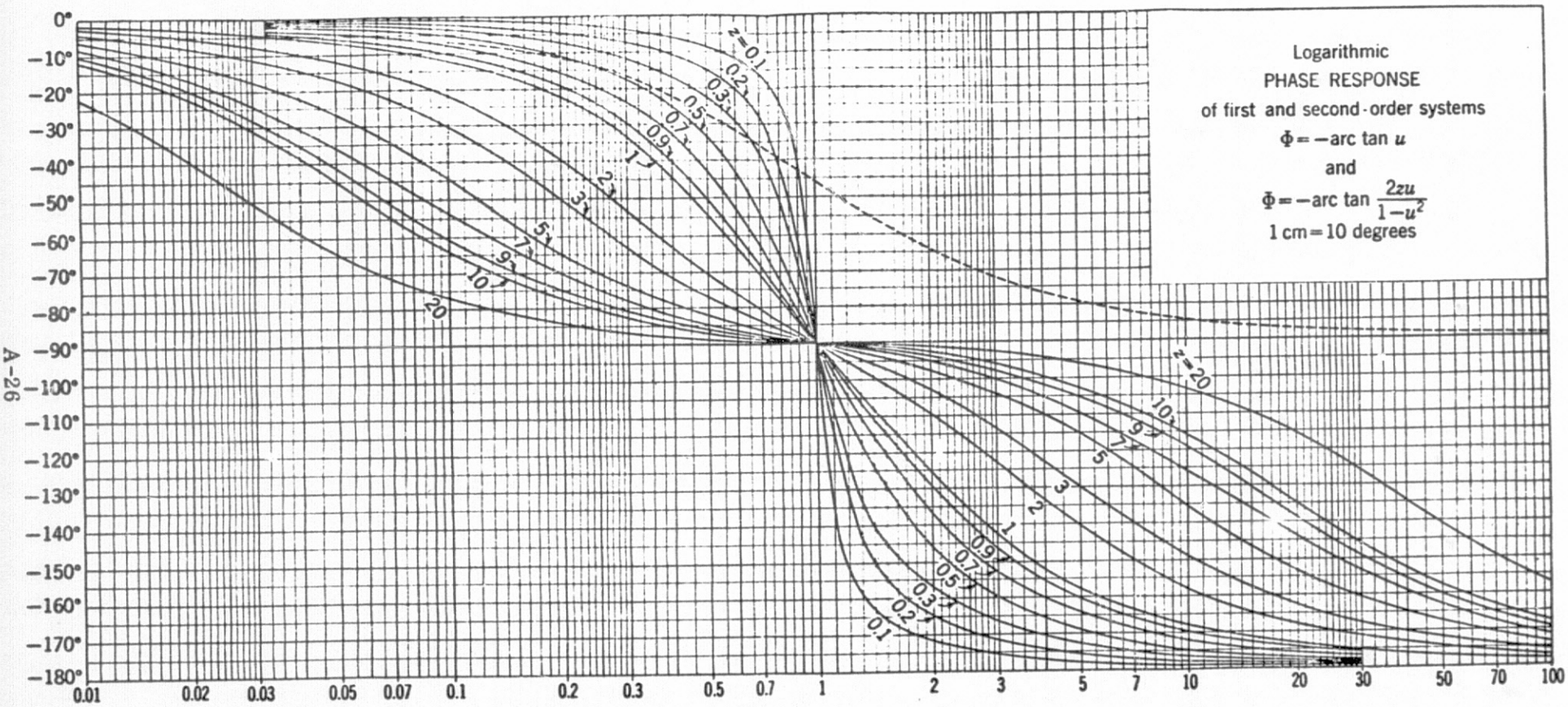


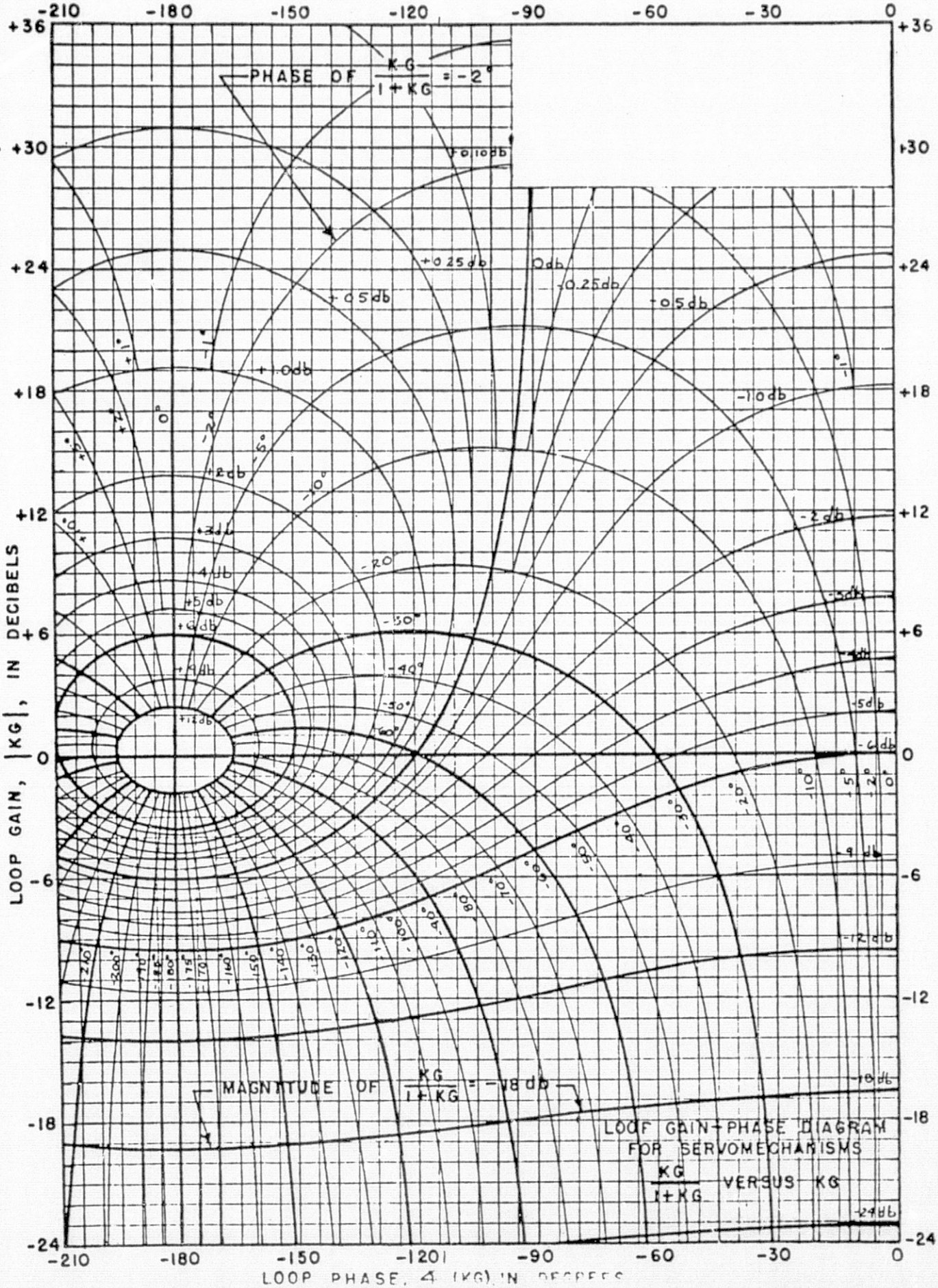
$$GH = \frac{K(s+s_1)(s+s_3)}{(s+s_2)^3}$$



$$GH = \frac{K(s+s_1)}{(s+s_2)(s+s_4)(s+s_6)}$$


Logarithmic
AMPLITUDE RESPONSE
of first and second-order systems
$A = \frac{1}{(1+u^2)^{\frac{1}{2}}}$
and
$A = \frac{1}{[(1-u^2)^2+(2zu)^2]^{\frac{1}{2}}}$
1 cm = 2 db
z = 0.1
0.14
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
2
3
4
5
9
10
20
0.1 0.2 0.3 0.4 0.6 0.8 1 2 3 4 6 8 10 20 30 40
10 8 6 5 4 3 2 1 0.8 0.6 0.5 0.4 0.3 0.2 0.1 0.05 0.01 .005 .001


Logarithmic
PHASE RESPONSE
of first and second-order systems
$\Phi = -\text{arc tan } u$
and
$\Phi = -\text{arc tan } \frac{2zu}{1-u^2}$
1 cm = 10 degrees
0°
−10°
−20°
−30°
−40°
−50°
−60°
−70°
−80°
−90°
−100°
−110°
−120°
−130°
−140°
−150°
−160°
−170°
−180°
0.01
0.02
0.03
0.05
0.07
0.1
0.2
0.3
0.5
0.7
1
2
3
5
7
10
20
30
50
70
100
z=0.1
0.2
0.3
0.5
0.7
0.9
1
2
3
5
7
9
10
20
z=20
10
9
7
5
3
2
1
0.9
0.7
0.5
0.3
0.2
0.1


PHASE OF KG/(1+KG) = -2°
MAGNITUDE OF KG/(1+KG) = -18 db
LOOP GAIN-PHASE DIAGRAM
FOR SERVOMECHANISMS
KG/(1+KG) VERSUS KG
LOOP GAIN, |KG|, IN DECIBELS
LOOP PHASE, ∠ (KG) IN DEGREES

APPENDIX B

SAMPLE APPLICATIONS OF COEBRA

This appendix presents some of the results obtained using the COEBRA program. The first two examples are classical testbook examples and the last five demonstrate the application of COEBRA to booster autopilot design.

Throughout this appendix the following shorthand notation, typical of COEBRA formatting, will be used to represent system transfer functions in either the s-plane or the w-plane. The transfer function given by

$$\frac{K\,(1 + \tau_1 S)\,(1 + \frac{2\zeta_1}{\omega_1} S + \frac{S^2}{\omega_1^2})}{(1 + \tau_2 S)\,(1 + \tau_3 S)\,(1 + \frac{2\zeta_2}{\omega_2} S + \frac{S^2}{\omega_2^2})} \qquad \text{B-1}$$

will be abbreviated as follows:

$$\frac{K\,(T1,\ \zeta 1,\ \omega 1)}{(T2,\ T3,\ \zeta 2,\ \omega 2)} \qquad \text{B-2}$$

Example 1.

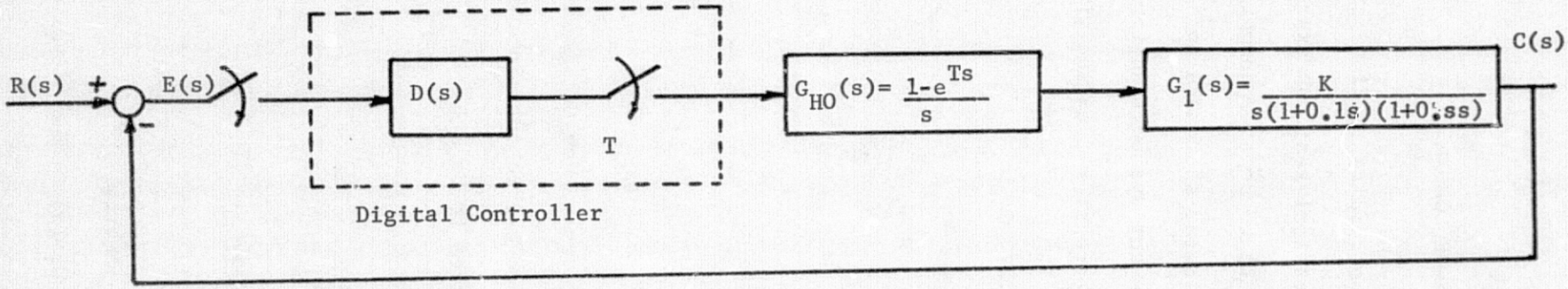
This ecample was selected to illustrate how COEBRA can be utilized to solve the digital system problem. Figure B-1 is the block diagram of the system, together with performance measures to be achieved. The first step is to convert the z-fransforms to the w-plane. Thus,

$$G_{HO}\,G_1(w) = \frac{0.13K\,(z + 1.31)\,(z + 0.054)}{z\,(z-1)\,(z-0.368)} \qquad z = \frac{1 + w}{1 - w}\ ,\ K=3$$

$$= \frac{0.77\,(1-w)\,(1-0.134w)\,(1+0.8925w)}{W(1 + w)\,(1 + 2.165\,w)} \qquad \text{B-3}$$

where K=3 is determined from the value theorem

$$\lim_{u \to \infty} G_{HO}\ G_1(uT) = \lim_{z \to 1} \frac{(z-1)}{z}\, G_{HO}\, G_1\,(z) \qquad 1.5 \qquad \text{B-4}$$



$T = 0.5$ sec.

$K_y \geqslant 1.5$

$M_\psi \geqslant 50$ degrees

$M_p \leqslant 1.3$

Figure B-1 Sampled Data System, Example 9-3, p. 291, Analysis and Synthesis of Sampled Data Control Systems, B. C. Kuu, Prentice-Hall, 1963.

The digital controller configuration selected was of the form

$$D(w) = \frac{1 + a\tau w}{1 + \tau w} \quad \ldots\ldots\ldots\ldots \quad \text{B-5}$$

Conventional paper and pencil design procedures, outlined in Figure B-2 yield a D(w) as

$$D(w) = \frac{1 + 50w}{1 + 100\ w} \quad \ldots\ldots\ldots\ldots \quad \text{B-6}$$

and an M_p of 1.2, M of 50 degrees, M_K of 12 db. converting to the z-domain

$$D(w) = \frac{1 + 50\ w}{1 + 100\ w}\ z = \frac{z-1}{z+1} = \frac{0.25\ (z - 0.96)}{z-0.99} \quad \ldots\ldots \quad \text{B-7}$$

and in the s domain, the compensation network corresponding to D(z) is evaluated from

$$z \quad \frac{Gc(s)}{s} = \frac{1}{1-z^{-1}}\ D(z)$$

or

$$Gc(s) = \frac{1 + 12.5s}{1 + 50\ s}$$

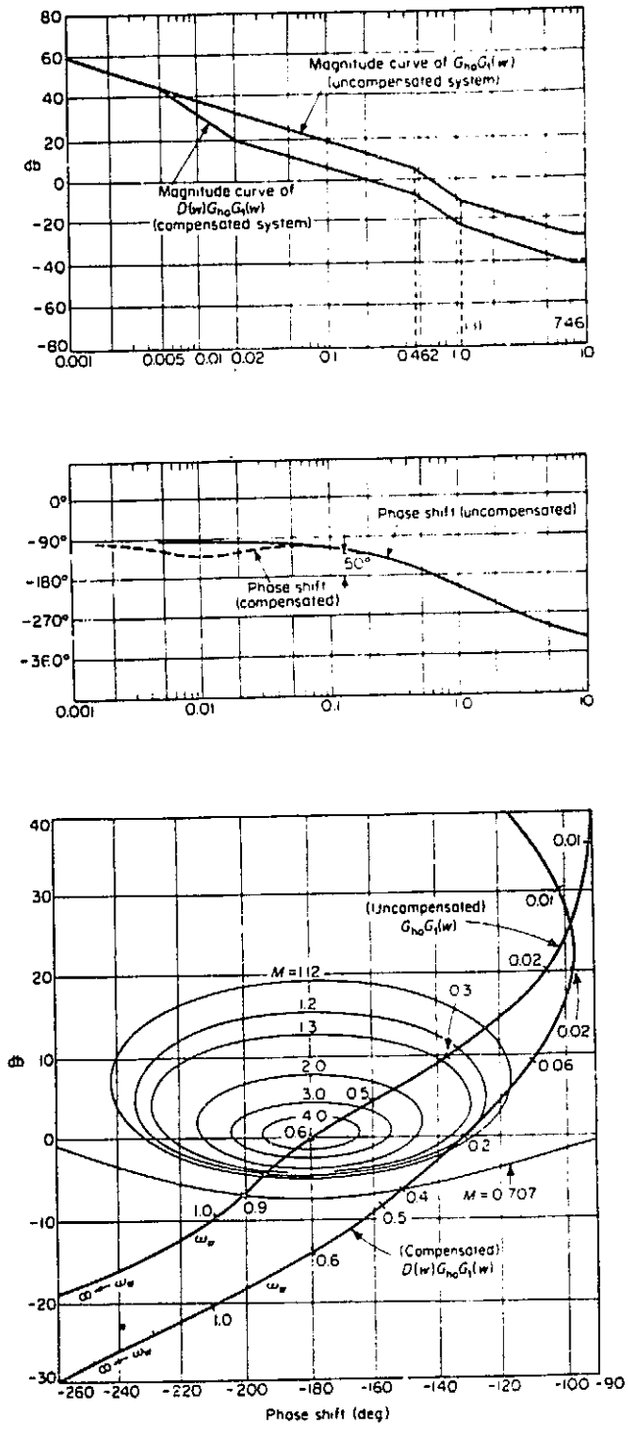


Figure B-2 Conventional Design Plots for Example 1. (a) Asymptotic Gain/Frequency. (b) Phase/Frequency (c) Nichols Chart to Obtain

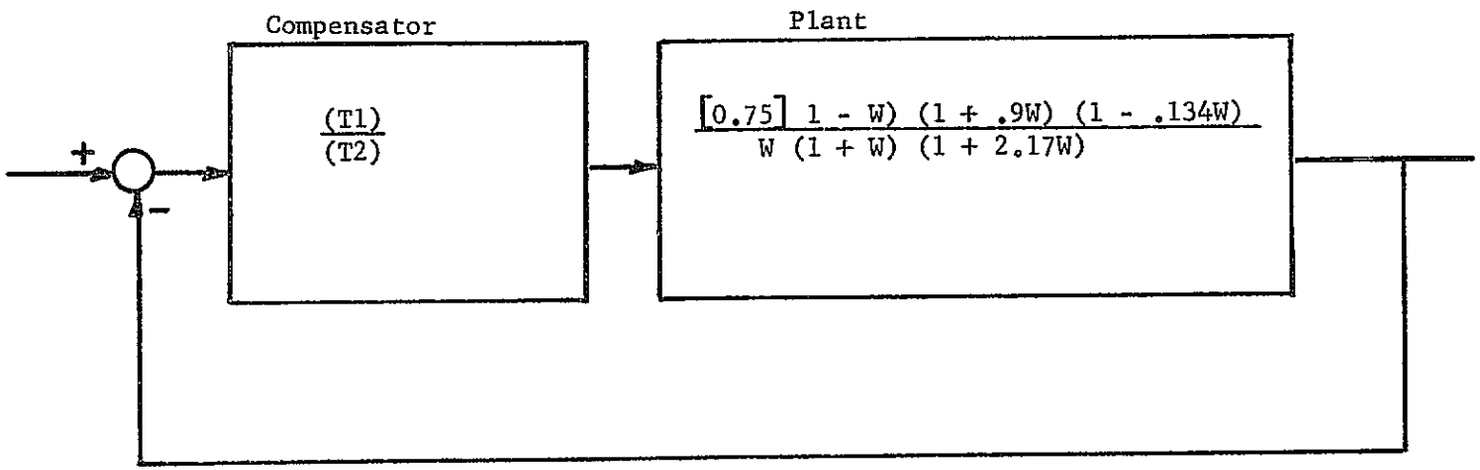


Figure B-3 W-Plane Block Diagram for Example #1

Figure B-3 illustrates the w-plane problem as submitted to COEBRA. Figure B-4 compares the results obtained from COEBRA with that obtained using a classical pencil and paper approach for the first COEBRA run. For this COEBRA run, both T1 and T2 were initialized with a value of 20, and hence the initial response shown in Figure B-4 is identical to the response of the plant only. Note that ω_c was 0.542 for the initial response. Figure B-4 shows the results of the first and sixth (final) major iterations of COEBRA. The sixth iteration was the final once since COEBRA was not "rewarded" for doing better than the classical design. In other words, recalling the discussion on termination, the figure-of-merit was not allowed to increase once the classical design results were matched. The results of the fifth and sixth iterations were identical, since it took COEBRA one iteration to decide that improvement was no longer possible or "permitted."

Table B-1 summarizes the initial and final compensators, as well as the final stability margins obtained from COEBRA run #1. Note that as with the classical approach, the final answer from Run 1 was a phase-lag compensator. Due to the circumstances of the problem, this minimum complexity (first order) compensator had to be a phase-lag model. In other words, phase-lead compensation would be ineffective.

Additional COEBRA runs were made in an attempt to "map the hill," or in other words, to see what COEBRA would do with different initial compensators. As shown in Table B-1, Runs 2, 3, and 4 achieved essentially the same results as were obtained from Run 1 and the classical approach. With the initial compensator of Run 5, COEBRA climbed a local optimum that did not satisfy the design requirements. Run 5 automatically terminated after 14 major iterations when the margin counter and the figure-of-merit essentially ceased to increase. Note that the final answer from Run 5 was not a phase-lag compensator. Run 6 was made with the denominator time constant (T2) of Run 5 changed to a value of 4., so that the

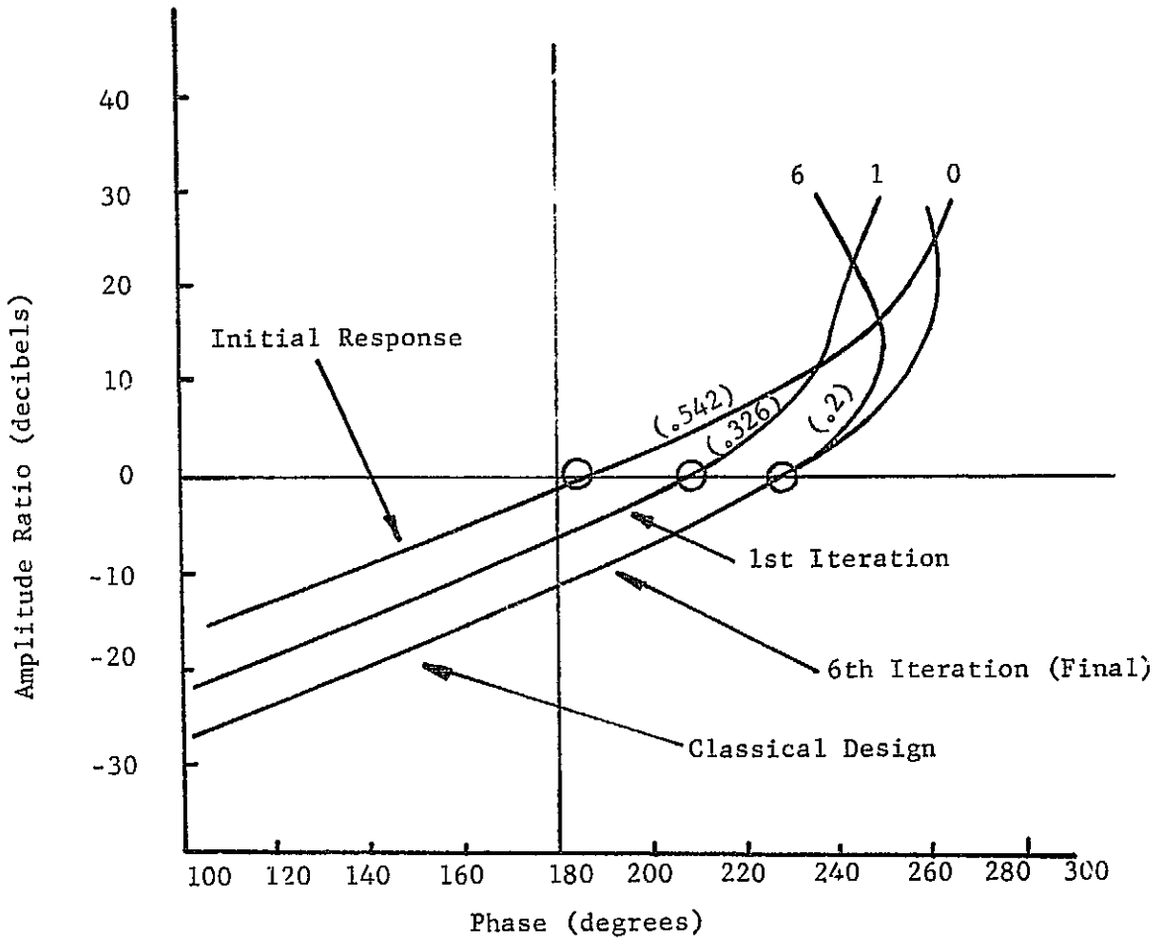


Figure B-4 Example #1, Gain-Phase Frequency Response Plot Comparing the Classical Design with COEBRA Run #1

Table B-1 Example #1 Summary of Results

Run	Initial Compensator		Final Compensator		Final Margins		
	T1	T2	T1	T2	Gain Margin (db)	Phase Margin (deg)	Phase Margin Frequency
Classical	--	--	50.	100.	12.	50.	0.2
1	20.	20.	80.5	283.	12.	50.	0.2
2	50.	50.	60.4	212.	12.	50.	0.2
3	100.	100.	76.3	268.	12.	50.	0.2
4	5.	5.	67.3	237.	12.	50.	0.2
5	2.	2.	1.02	0.0245	7.	30.	0.61
6	2.	4.	83.	291.	12.	50.	0.2

initial compensator was a phase-lag compensator. Table B-1 shows that Run 6 achieved essentially the same result as obtained using the classical approach.

Since the first order phase-lag compensator is the minimum-complexity compensator that can solve this problem, it is not difficult to understand why COEBRA could not converge to a final solution from every initial condition. This points out that the difficulty of any problem is dictated more by the degrees of freedom in the compensator than by the complexity or order of the plant.

B-2 Example #2

This example, easily carried out by classical methods as shown in Chapter 1, is included here to illustrate the use of COEBRA when the compensation is of higher order than that of example 1. Figure B-5 shows the system in COEBRA format. The details of the classical approach are given in Chapter 1. The problem given to the COEBRA program was to adjust the four compensator time constants until the above three design requirements were satisfied.

Table B-3 compares the classical results with those obtained from the first COEBRA run. With all the time constants in the compensator initialized to unity, COEBRA, in six major iterations, climbed to a local optimum that did not meet the design requirements. Conventional design procedures showed that the minimum-complexity compensator that is required to solve this problem, is a lag-lead compensator, and the final answer for the unsuccessful Run 1 is not a lag-lead compensator. COEBRA terminated after six iterations when the margin counter and the figure-of-merit ceased to improve.

COEBRA was reinitialized to the compensator shown for Run 2 in Table B-3. As can be seen from the table, Run 2 achieved all the design objectives. It did so with a lag-lead compensator.

Figure B-6 compares the classical results with those obtained from Run 2.

Table B-2. Example #1 Computer Time

Run	Number of Iterations		Computer Time (sec)
	Major	Minor	
1	6	39	118
2	5	27	89
3	5	21	73
4	7	51	154
5	14	72	252
6	8	53	159

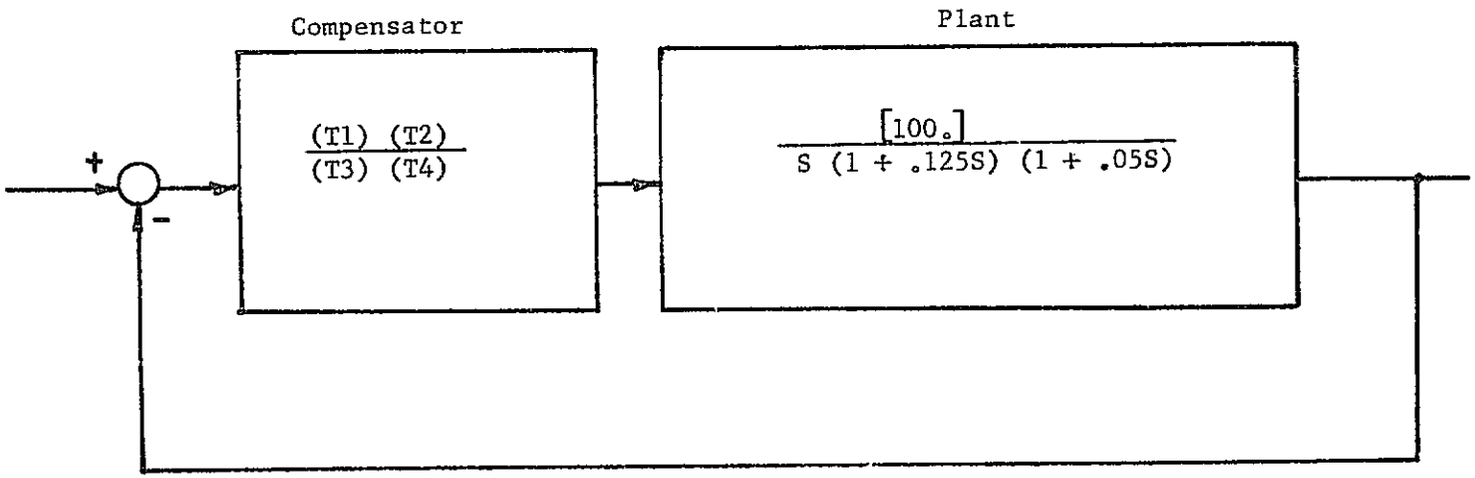


Figure B-5 System Block Diagram for Example #2

Table B-3. Example #2 Summary of Results

Run	Initial Compensator				Final Compensator				Final Margins		
	T1	T2	T3	T4	T1	T2	T3	T4	Gain Margin (db)	Phase Margin (deg)	Phase Margin Freq.(rps)
Classical	--	--	--	--	1.	.2	10.	.02	14.	52.	12.
1	1.	1.	1.	1.	9.91	7.86	20.9	20.9	4.	11.4	10.
2	10.	.2	10.	.2	1.48	1.35	74.	.027	12.	57.	14.2
3	1.5	.7	1.5	.7	1.055	.718	3.02	1.41	1.7	4.6	10.
4	5.	.4	5.	.4	.83	.82	30.6	.0654	11.	53.	10.4

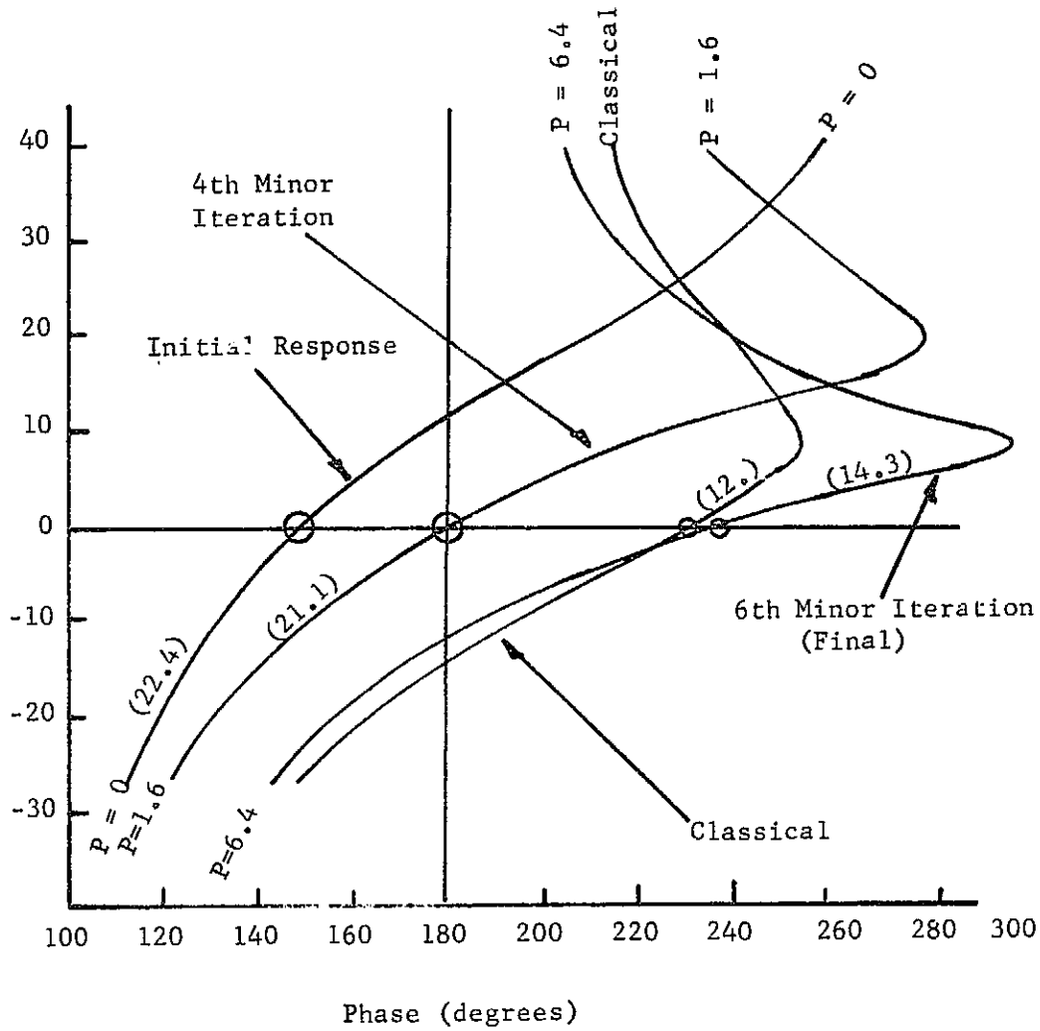


Figure B-6 Example #2, Gain-Phase Frequency Response Plot Comparing Classical Results with COEBRA RRun #2

Since the numerator and denominator of the initial compensator for Run 2 are identical, the inital response shown in Figure B-6 is identical to the response of the plant only. As can be seen, the system with a unity compensator (COEBRA's initial compensator) is unstable. Note that ω_c for the initial response is 22.4 rad/sec. Figure shows the results of the fourth and sixth minor iterations in the first major iteration of Run 2. The results of the sixth minor iteration in the first major iteration satisfy all the design requirements, and these results were the best COEBRA was "allowed" to achieve. As with Example #1, the reason for this was that COEBRA was only "rewarded" up to the design requirements. In other words, recalling the discussion on Termination, the figure-of-merit was not allowed to increase once the classical results were matched. Run 2 ran for two major iterations since it took COEBRA one iteration to decide that further improvement was not allowed.

Two more COEBRA runs were made on this problem. Their initial compensators were chosen to "lie between" the initial compensators of Run 1 and Run 2. From Table B-3, it is seen that the initial compensator for Run 3 was "close" to that for Run 1. As with Run 1, Run 3 climbed a local optimum that did not yield a feasible solution. Run 3 went four major iterations before the counter and figure-of-merit indicated that no further improvement was possible. The final answer from Run 3 was not a lag-leg compensator.

Run 4 was made with an initial compensator that was "between" the initial compensators for Run 2 and 3. As can be seen from Table B-3, Run 4 achieved satisfactory results with a lag-lead compensator.

Since a second order lag-lead compensator is the minimum-complexity compensator that can solve this problem, it is not difficult to understand why COEBRA could not converge to a final solution from every initial condition. As with Example #1, this points out again that the difficulty of any problem is

dictated more by the order of the compensator than by the order of the plant. In most cases, the "optimum hill" boradens and smoothes out as the order or complexity of the compensator increases.

Table B-4 summarizes the computer time required to make COEBRA runs 1 through 4.

B-3 Example #3

Example #3 illustrates the application of COEBRA to a single-time-point autopilot design problem where the initial autopilot was so poor that it resulted in a rigid-body instability. The objective of the COEBRA run was not only to stabilize the system, but also to optimize all stability margins.

The airframe (or system to be controlled) included rigid-body dynamics and eight structural bending modes. The block diagram of the airframe/autopilot system is shown in Figure B-7. Since this is a so-called analog autopilot, the design is performed in the S-plane.

Figure B-7 shows the attitude loop with a gain and two filters, and two rate loops (for bending mode stability [Harris, 15]), each with a gain and two filters. The problem given to COEBRA was to adjust these nine autopilot parameters until all stability margins were optimized.

Figure B-8 shows the open-loop frequency response resulting from the initial autopilot. This figure shows that the initial autopilot did result in a rigid-body instability. The resonances of the eight structural bending modes are indicated by Figure B-8, which also shows that the rigid-body phase margin frequency (ω_c) was 4.62 rad. per second.

Arrows around the critical point in Figure B-8 illustrate the required rigid-body and first mode stability margins. It was also required that ω_c be greater than 2. rad/sec and that modes 2 through 8 be gain stabilized with their peaks resonating below "-10" decibels.

Table B-4. Example #2 Computer Time

Run	Number of Iterations		Computer Time (sec)
	Major	Minor	
1	6	37	148
2	2	17	63
3	4	17	81
4	3	21	83

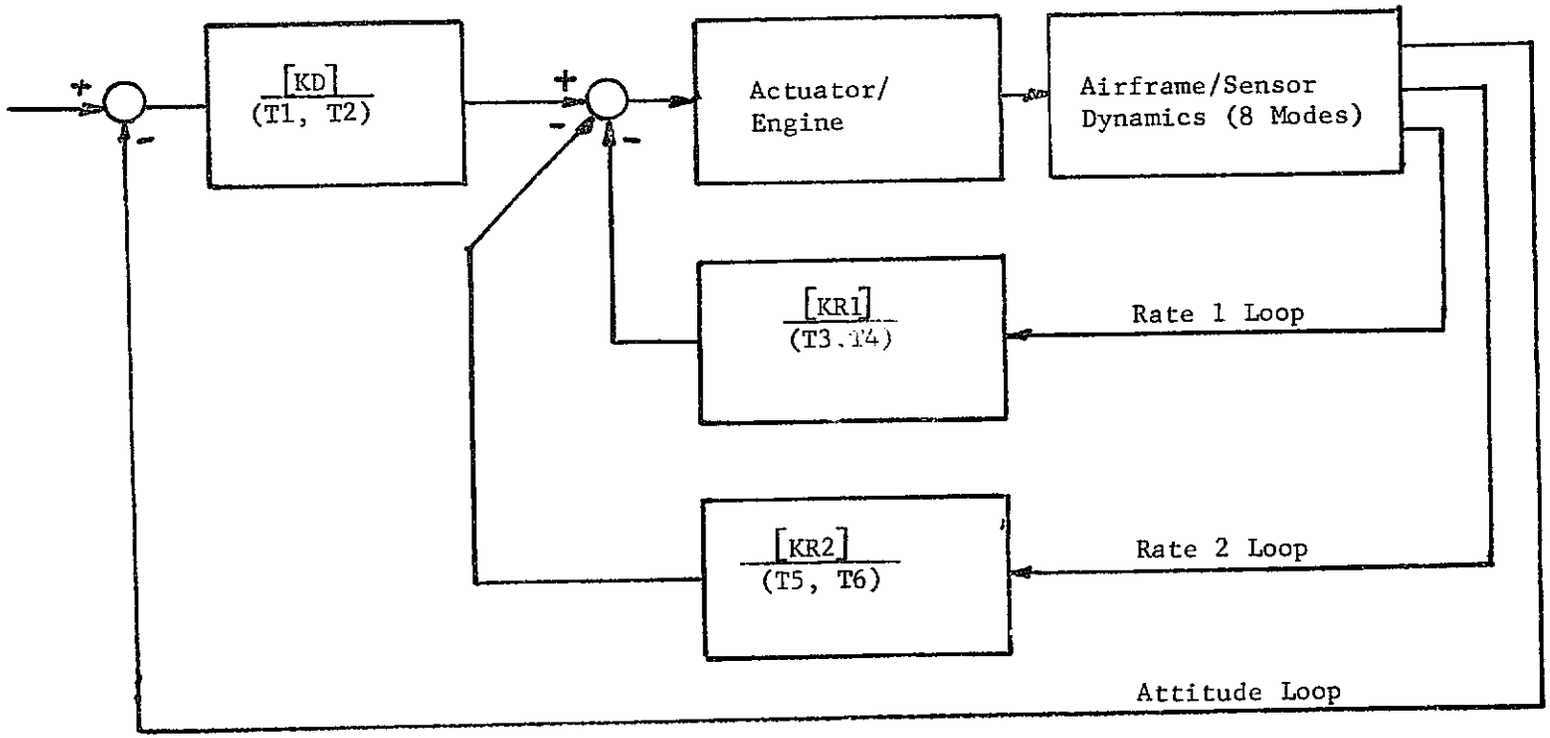


Figure B-7 System Block Diagram for Example #3

Figure B-9 shows the frequency response after the first major iteration. The system is now stable, with ω_c = 2.07 rad/sec.

Figure B-10 shows the frequency response that resulted from the third and final iteration. COEBRA self-terminated after all design requirements were met, and after the margin counter and figure-of-merit ceased to significantly improve following the second major iteration. In other words, the results of the second and third iterations were identical since it took COEBRA one iteration to determine that design improvement was no longer possible. Further improvement would have been rewarded, but COEBRA was unable to achieve it. The margins that prohibited further improvement were ω_c and the phase margin on the "backside" of the first mode.

Table B-5 summarizes the results of this example. It shows the values of all nine parameters for both the initial and the final autopilots. Table B-5 shows that a satisfactory design was achieved in 493 seconds of computer time.

The following is a discussion of how COEBRA presently treats the requirement on the dominant rotational rigid-body closed-loop roots. Up to the present, the time domain response due to guidance commands of a large aerodynamically unstable flexible launch vehicle, has not been too critical. The major concern has been with stability under tolerances and with structural bending moment loads. The main reasons for specifying dominant closed-loop root locations have been to (1)keep the autopilot frequencies sufficiently separated from the guidance loop frequencies for stability purposes, and (2) merely provide "somewhat adequate" response to guidance commands. Historically, it has been found that if the rigid-body phase margin is greater than a certain value, then the rigid-body rotational closed-loop roots will be sufficiently damped at a high enough frequency. For example, on launch vehicles like the one represented in this example problem, if the phase margin is greater than 30 degrees, with a frequency greater than 2.0

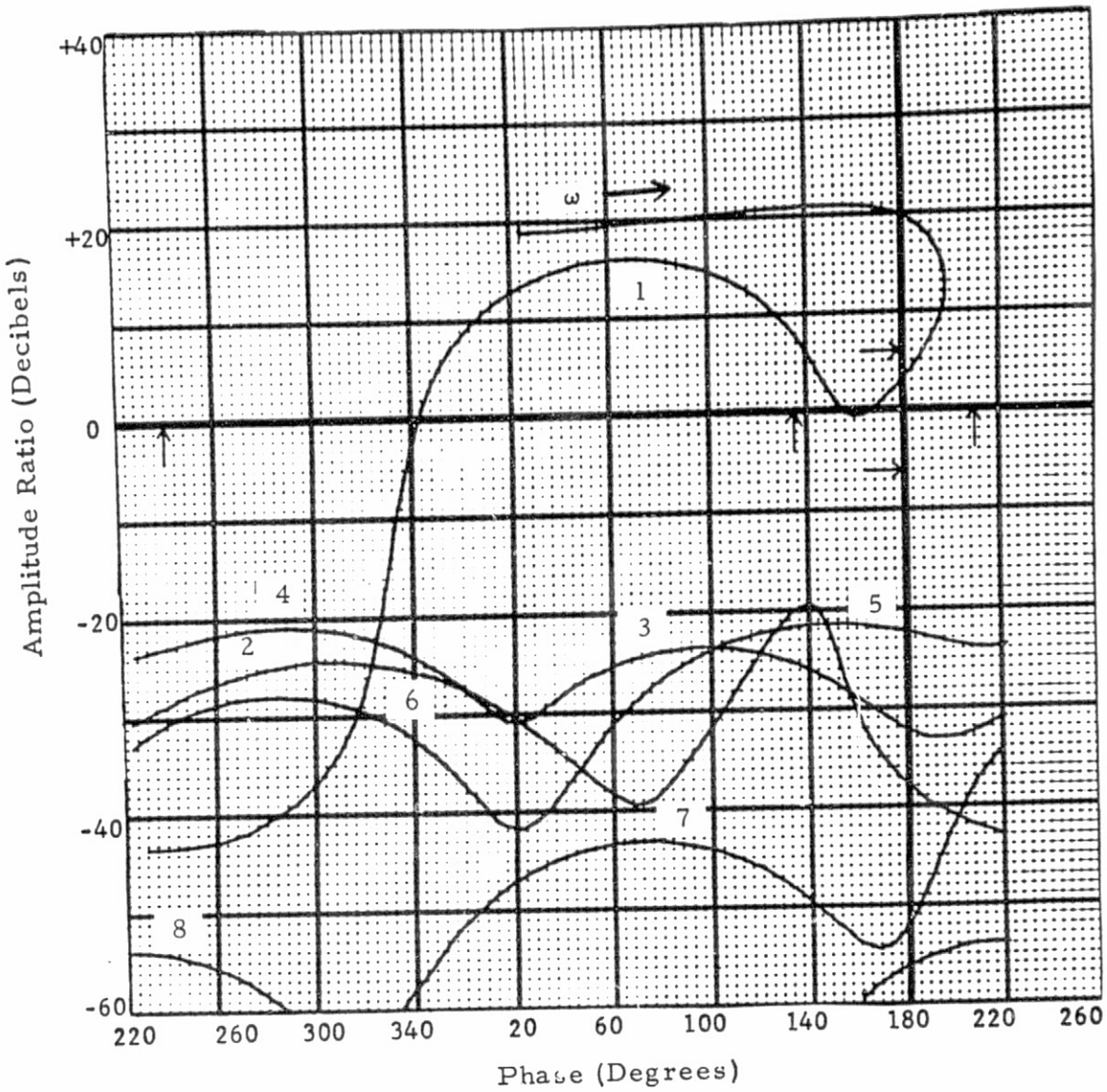


Figure B-8 Example #3, Gain-Phase Frequency Response Plot Resulting from Initial Autopilot (ω_c = 4.62 rps)

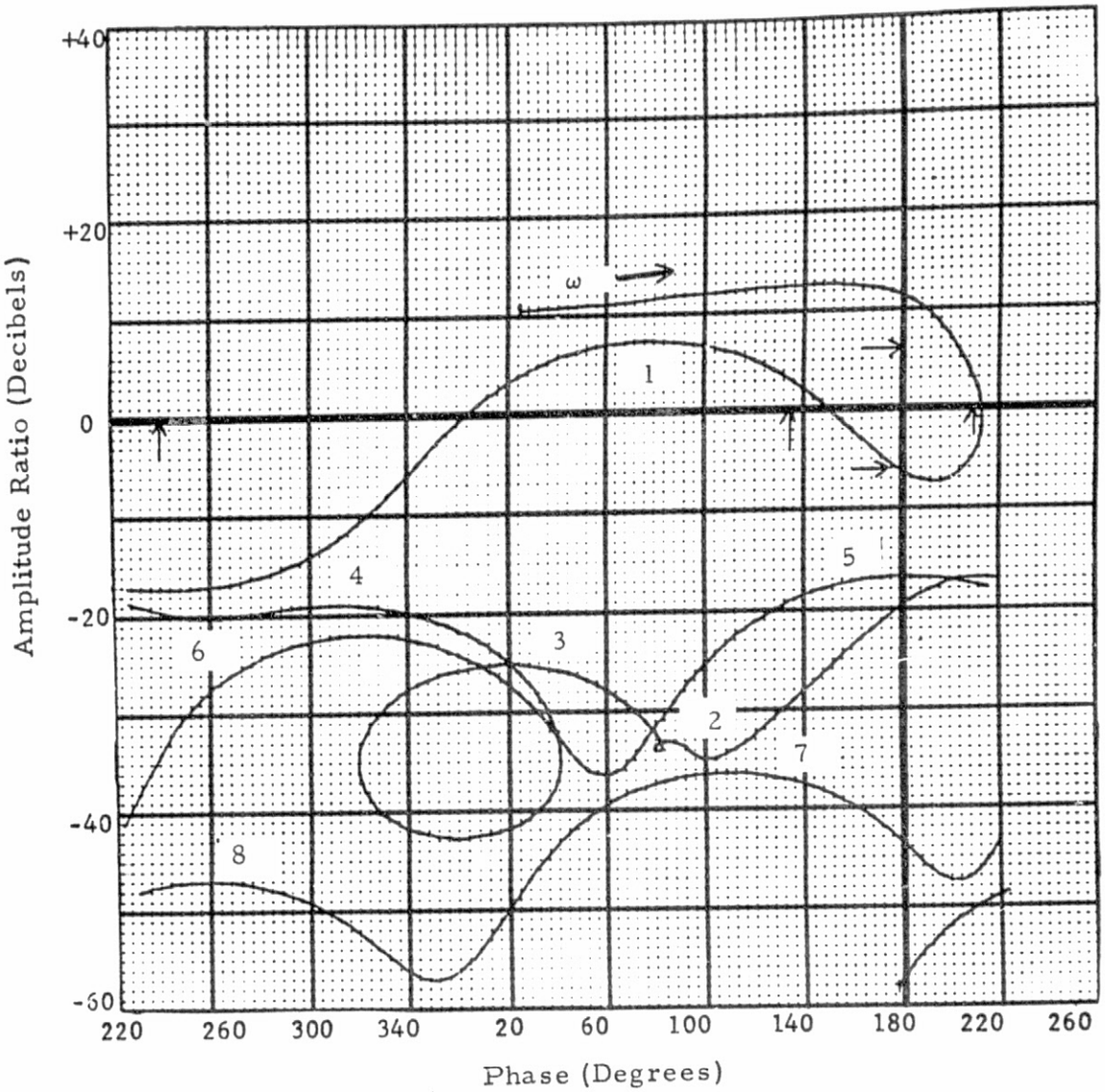


Figure B-9 Example #3, Gain-Phase Frequency Response Plot Resulting from First Major Iteration (ω_c = 2.07 rps)

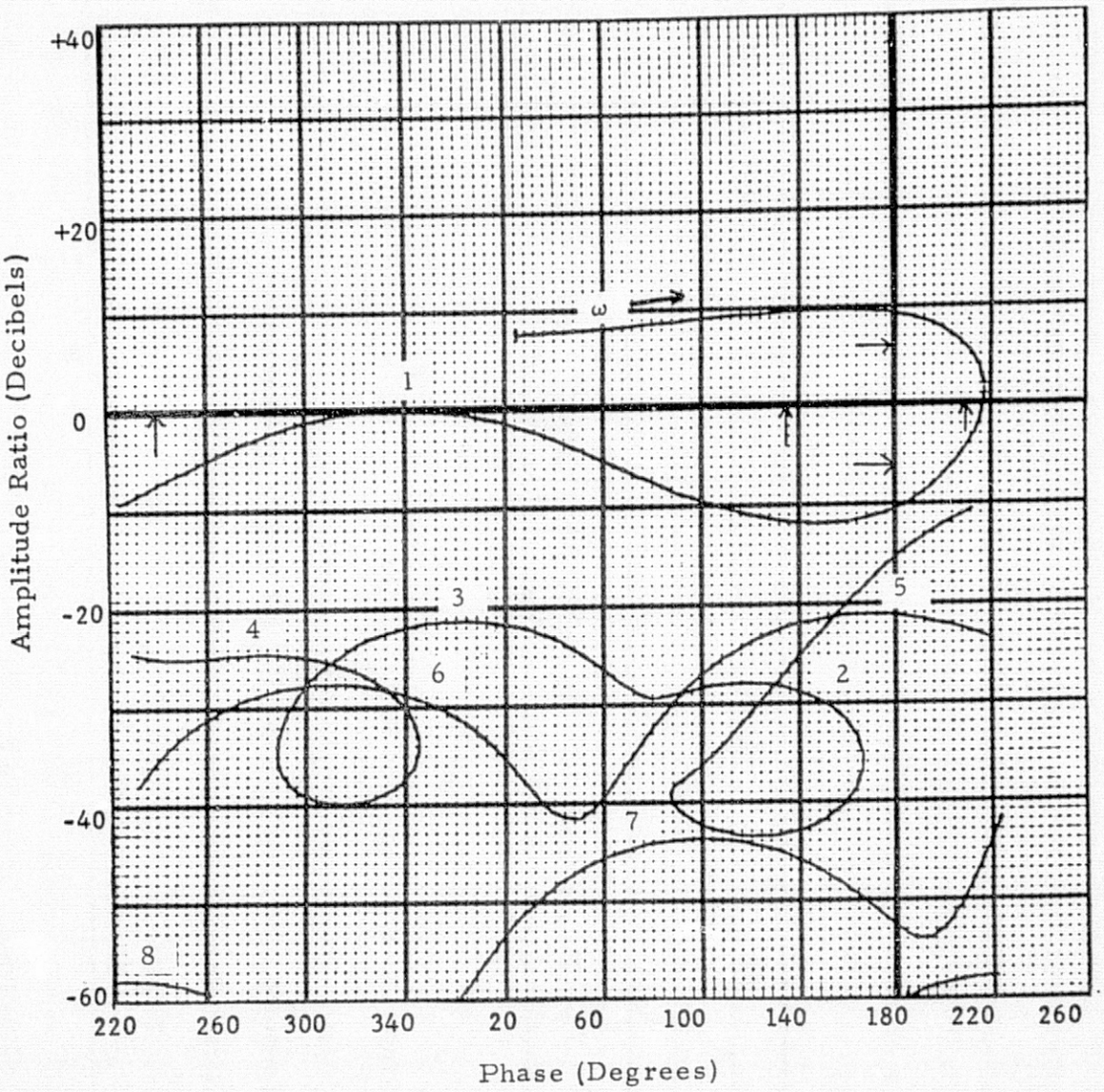


Figure B-10 Example #3, Gain-Phase Frequency Response Plot Resulting From Third (Final) Major Iteration (ω_c = 2.03 rps)

Table B-5. Example #3 Summary of Results

	KD	T1	T2	KR1	T3	T4	KR2	T5	T6
Initial Autopilot	4.	.0333	.0333	1.6	.2	.8	.2	.2	.2
Final Autopilot	1.1	.018	.018	.86	.108	.22	.108	.108	.108

- Objective: Maximum Margins
- Single Time Point, 27th Order System
- 3 Major Iterations and 19 Minor Iterations
- Computer Time: 493 sec or 164 sec per Major Iteration
- Note: Initial Autopilot Yielded Rigid-Body Instability

radians per second, then it is almost certain that the rotational closed-loop roots will have a frequency greater than 1.5 radians per second and a damping ratio greater than 0.30. The M circles for unity feedback systems tend to indicate they this has been.* Hence, rather than finding the actual roots, COEBRA treats the requirement on the rotational closed-loop roots by putting minimum allowed values on the rigid-body phase margin and its frequency (ω_c). This approach was taken in order to avoid the computer time required to find the actual closed-loop roots.

It is recognized that the rigid-body response of a launch vehicle is comprised of a so-called first-order drift root as well as the second-order rotational roots [Greensite, 14, and Harris, 15]. Hence, since the rigid-body response is third-order, the location of the rotational roots alone is not sufficient to ensure adequate response to guidance commands. However, since most launch vehicles are aerodynamically unstable, the instability of the vehicle generall serves to keep the attitude gain high enough, and the flexibility of the vehicle generally serves to keep the rate gain low enough, so that the rotational roots dominate the drift roots. In this way, the location of the rotational roots themselves can be used to indicate response to guidance commands. For this example problem, where the final autopilot yielded a phase margin of 38. degrees at a frequency of 2.03 radians per second, the closed-loop rotational roots had an effective damping ratio of 0.68 and an undamped natural frequency of 2.9 radians per second. This satisfied the design requirements, and a transient response showed that these roots dominated the drift root which had a time constant of 11.5 seconds.

---

*Nichols Chart of Appendix A

B-4 Example #4

Example #4 illustrates the application of COEBRA to the same airframe that was used in Example #3, but this time the initial autopilot was so poor that it resulted in a first-mode instability. As with Example #3, the objective of the COEBRA run was not only to stabilize the system, but also to optimize all stability margins.

The block diagram of the airframe/autopilot system is the same as that of Example #3, and is shown in Figure B-7 . The problem given to COEBRA was to adjust the nine S-plane autopilot parameters until all stability margins were optimized.

Figure B-11 shows the open-loop frequency response resulting from the initial autopilot. This figure shows that the initial autopilot did result in a first-mode instability. The resonances of the eight structural bending modes are indicated in Figure B-11, which also shows that the rigid-body phase margin frequency (ω_c) was 1.17 rad. per second.

As with Example #3, the arrows around the critical point in Figure B-11 illustrate the required rigid-body and first mode stability margins. It was also required that ω_c be greater than 2.0 rad/sec, and that modes 2 through 8 be gain stabilized with their peak resonating below "-10" decibels. Figure B-11 shows that with the initial autopilot, the fourth and fifth modes exceed this requirement.

Figure B-12 shows the frequency response after the second major iteration. The system is now stable, with ω_c = 1.3 rad/sec.

Figure B-13 shows the frequency response that resulted from the fifth and final major iteration. COEBRA self-terminated after all design requirements were met, and after the margin counter and figure-of-merit ceased to significantly improve following the fourth major iteration. In other words, the results of the

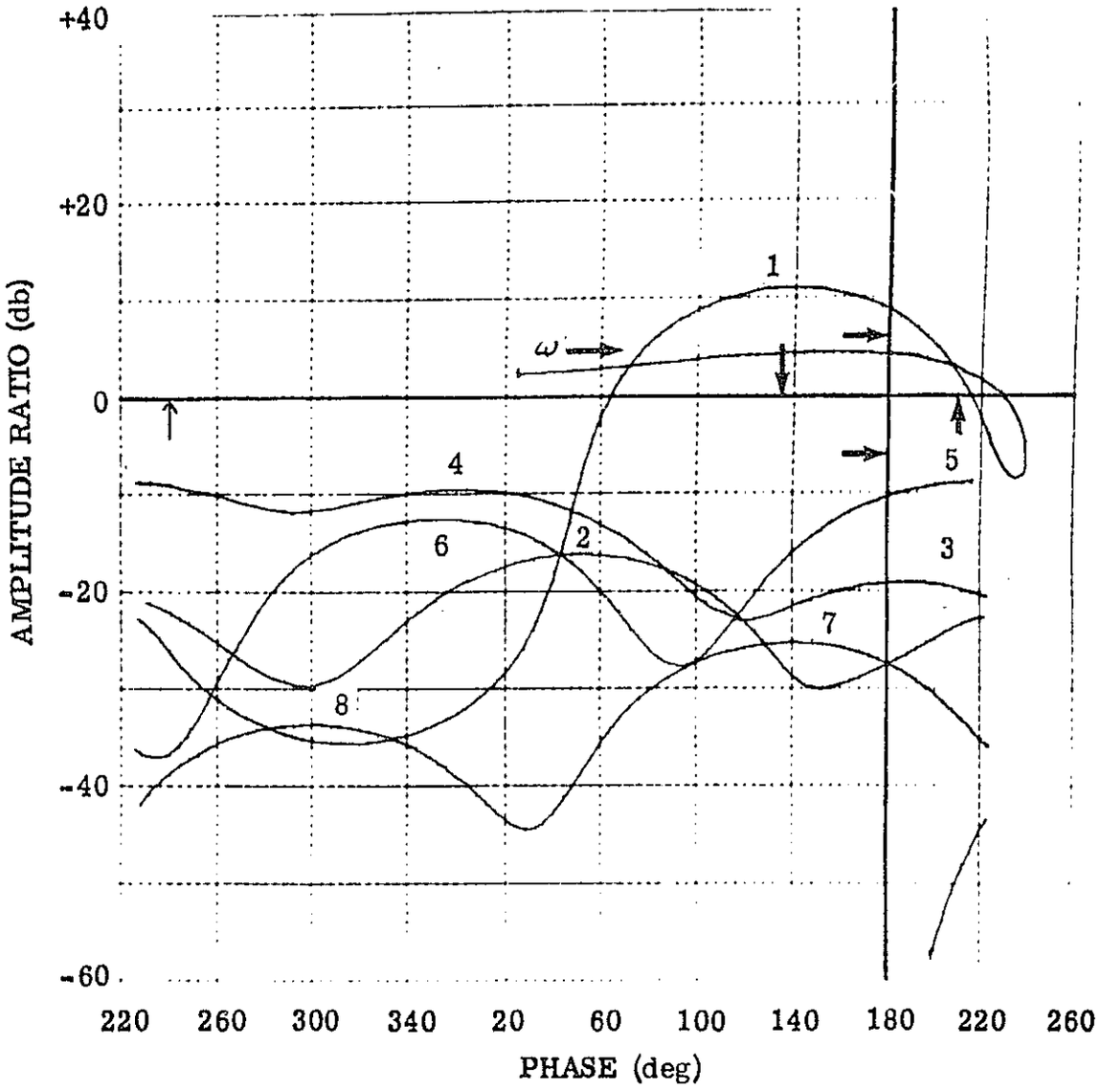


Figure B-11. Example #4, Gain-Phase Frequency Response Plot Resulting From Initial Autopilot (ω_c = 1.7 rps)

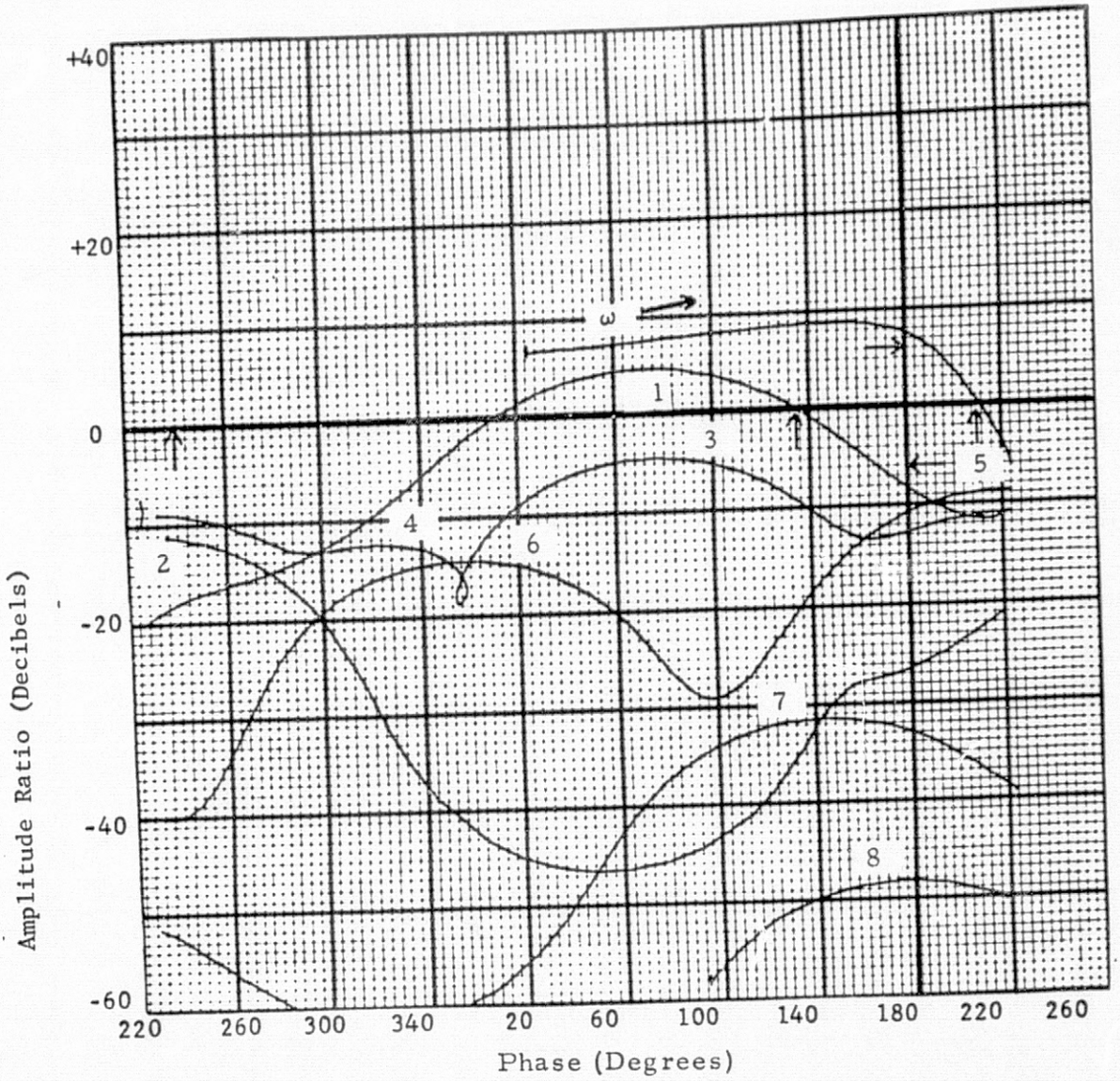


Figure B-12. Example #4, Gain-Phase Frequency Response Plot Resulting From Second Major Iteration (ω_c = 1.3 rps)

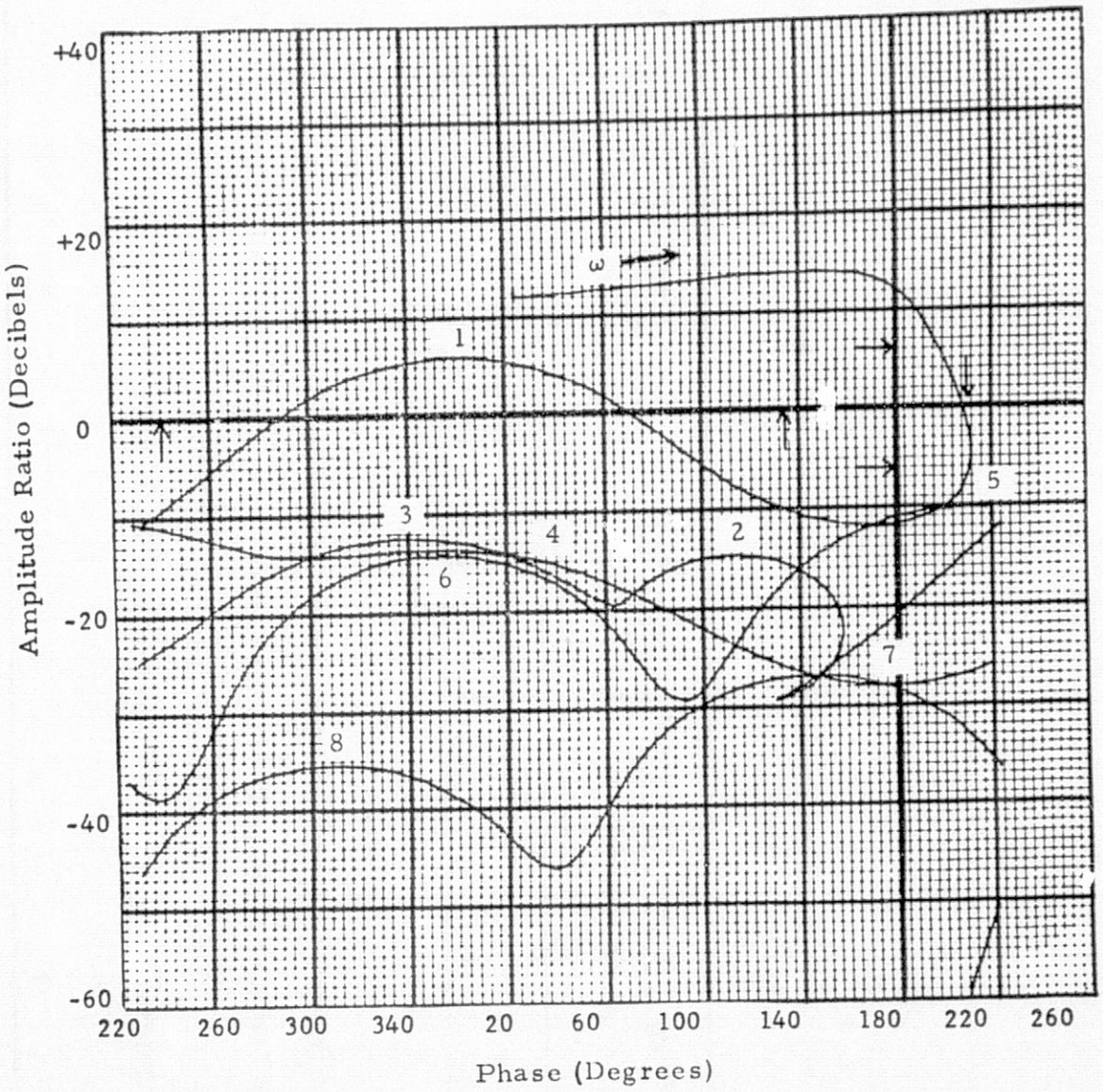


Figure B-13. Example #4, Gain-Phase Frequency Response Plot Resulting From Fifth (Final) Major Iteration (ω_c - 2.0 rps)

Table B-6. Example #4 Summary of Results

	KD	T1	T2	KR1	T3	T4	KR2	T5	T6
Initial Autopilot	.6	.0333	.0333	.3	.0333	.0333	.5	.0333	.0333
Final Autopilot	1.83	.093	.061	.7	.053	.053	.32	.039	.021

- Objective: Maximize Margins
- Single Time Point, Same 27th Order System as Example #3
- 5 Major Iterations and 29 Minor Iterations
- Computer Time: 975 sec or 193 sec per Major Iteration
- Note: Initial Autopilot Yielded First Mode Instability

fourth and fifth iterations were identical since it took COEBRA one iteration to determine that design improvement was no longer possible. Further improvement would have been rewarded, but COEBRA was unable to achieve it. The margins that prohibited further improvement were ω_c, the phase margin on the "backside" of the first mode, and the modal peaks of the third and fifth modes.

Table B-6 summarizes the results of this example. It shows the values of all nine parameters for both the initial and the final autopilots. Comparing these values with those obtained from Example #3, it is seen that COEBRA "climbed" to a different local optimum than it did for Example #3. However, some features of the two results are similar. For both, the attitude gain (KD) is about 1.5, the total rate gain (KR1 + KR2) is about unity, and KR1 is greater than KR2 in order to "center" the first mode around zero degrees.

Table B-6 shows that a satisfactory design was achieved in 975 seconds of computer time.

B-5 Example #5

Example #5 illustrates the application of COEBRA to a three-time-point autopilot design problem, where the objective was to optimize structural bending moment load relief capability. COEBRA was initialized with an autopilot that had previously been designed by engineers. The reason for this COEBRA run was to determine if design improvement could be achieved. Design improvement was defined as an autopilot that had more load relief capability, but still met satisfactory stability margins.

The example deals with the max-$\bar{q}$ portion of flight where aerodynamic loads are critical. The three vehicle states that are designed together are: (1) the time at which the load relief loop (the accelerometer feedback loop) is switched in; (2) the max-$\bar{q}$ time point; and (3) the time at which the load

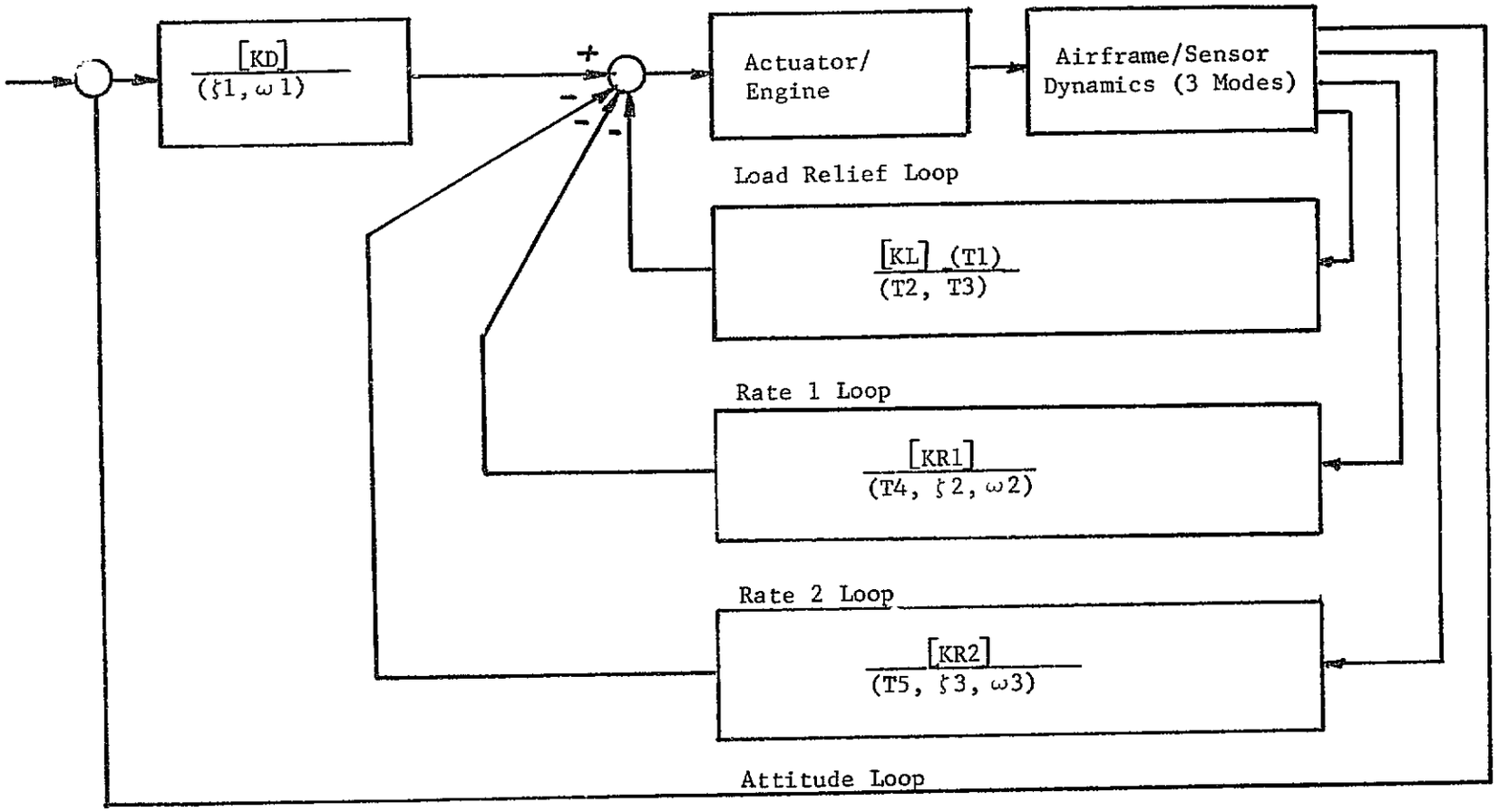


Figure B-14 System Block Diagram for Example #5

relief loop is switched out. The airframe consists of rigid-body dynamics plus three bending modes at each time point.

Figure B-14 shows the block diagram of the airframe/autopilot system. In addition to the attitude loop and two rate loops, Figure B-14 shows the so-called load relief loop. This is a feedback loop on a lateral body-mounted accelerometer signal, and Harris [15] discusses how this loop is used to reduce the angle of attack and control deflections (hence bending moment loads) in the presence of the wind.

COEBRA is allowed to vary the gains and filters shown in Figure B-14. This is a digital autopilot design problem, and hence, these gains and filters are defined in the W-plane. Of course, when the design is complete, these gains and filters will be transformed to the Z-plane where they will be mechanized as coefficients in different equations.

Figure B-14 shows that, at each time point, 15 autopilot parameters can be varied. Since this is a digital autopilot, the four gains (KD, KL, KR1 and KR2) can be different at each of the three time points. The 11 filter network values, though they may be varied, must have the same values at all three time points.

Figure B-15 shows the open-loop frequency response plot that results at the max-$\bar{q}$ time point from the engineer's final autopilot (initial autopilot for the COEBRA run). This figure, as well as the frequency responses at the other two time points (not shown), show that all margin requirements are satisifed. When a six-degree-of-freedom (6 DOF) trajectory was run using this "final" autopilot, the load relief indicator (which is a product of the dynamic pressure times the angle of attack and is indicated as $q\alpha$) was 4908 pounds per square foot.

Figure B-16 shows the frequency response plot that resulted at the max-$\bar{q}$

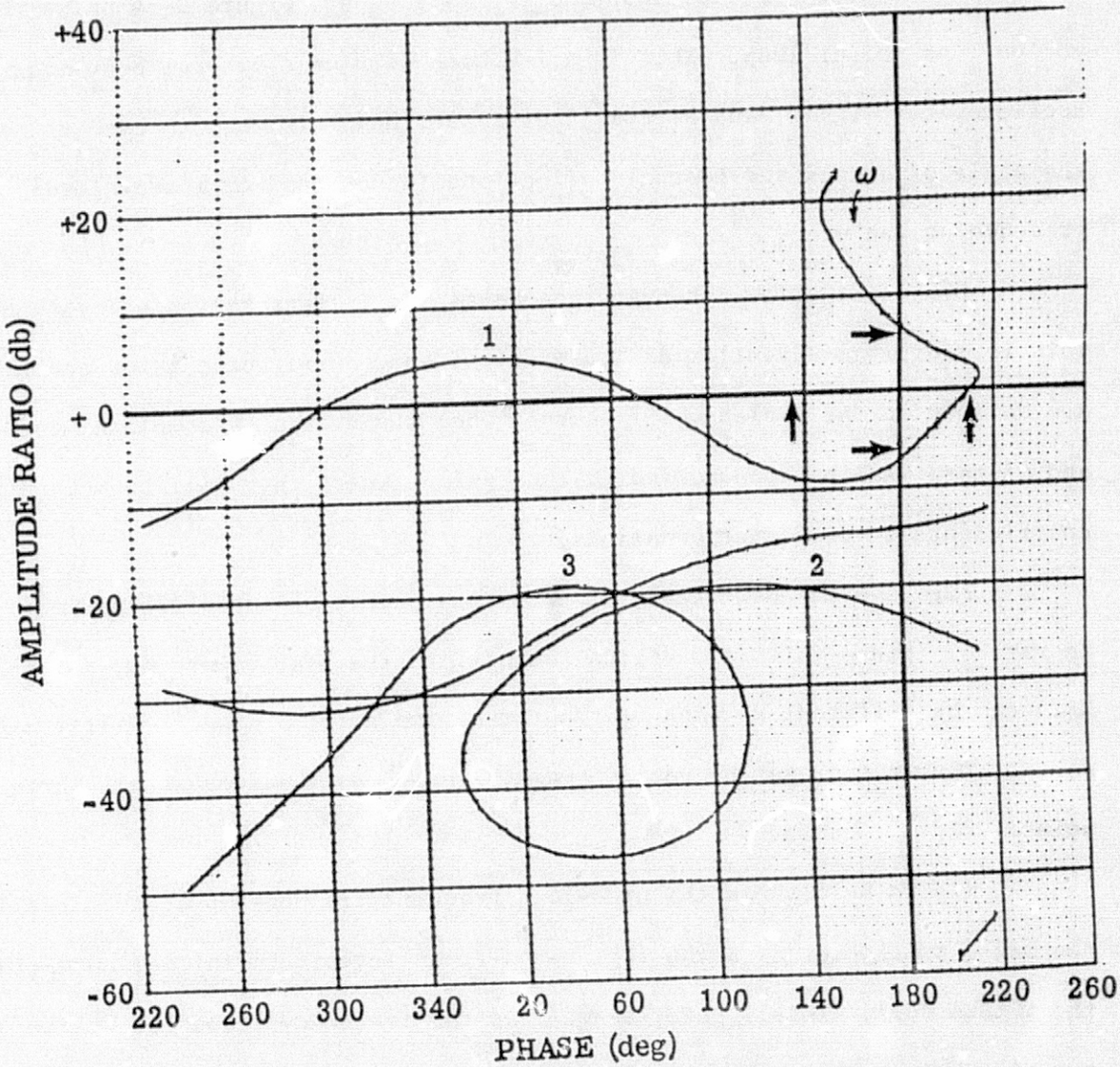


Figure B-15 Example #5, Gain-Phase Frequency Response Plot Resulting from Initial (Engineer's Final) Autopilot at Max-$\bar{q}$

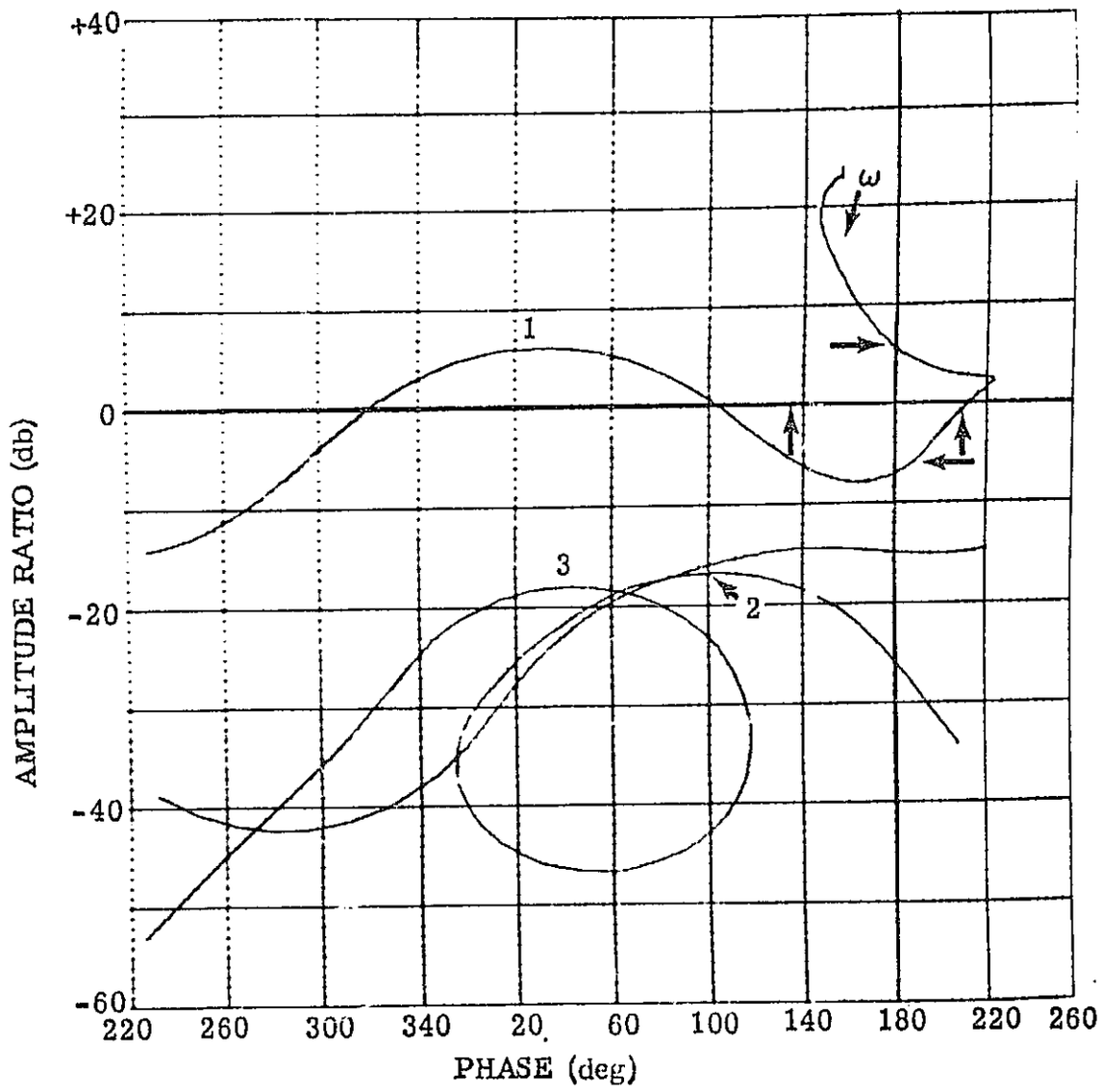


Figure B-16 Example #5, Gain-Phase Frequency Response Plot Resulting from Third (Final) Iteration at Max-q

Table B-7. Example #5 Summary of Results

- Objective: Maximize Load Relief Capability

-

Autopilot	Stability Margins	$q\alpha$
COEBRA Initial (Engineer's Final)	Satisfactory	4908
COEBRA Final	Satisfactory	4765

- System Order:
  - 28th Order at Each of the Three Time Points
  - 15 Autopilot Variables at Each Time Point (4 gains can have different values at each, 11 filters must have same values at each).

- 3 Major Iterations

- Computer Time: 21.3 Minutes or 7.1 Minuters per Iteration

time point from the third and final iteration of the COEBRA run. Stability margin requirements ar met at this time point, as well as at the other two time points. For this final COEBRA autopilot, a 6 DOF trajectory simulation showed that q had been reduced to 4765 pounds per square foot.

Figure B-16 shows that the stability margins from COEBRA's final autopilot, though satisfactory, are less than those from the engineer's final autopilot (Figure B-15). This demonstrates the tradeoff that does exist between stability and load relief.

The conclusion of this example is that, starting from the engineer's final autopilot, COEBRA was able to achieve an improved design by adjusting the values of gains and filters within an engineer's established configuration. It is noted that the COEBRA improvement in load relief did not result because the engineer was incapable, but rather because he was not required to obtain more load reduction.

Table B-7 summarizes the results obtained from this example. It shows that a satisfactory result was obtained after 21.3 minutes of computer time, or 7.1 minutes per iteration.

One final note is mentioned at this time. Another COEBRA run was made on this problem, beginning with the same initial autopilot, but with the objective changed to maximizing stability margins instead of load relief capability. The result of this second run was a design with improved stability margins, but with reduced load relief capability. Computer time for this run was 2.3 minutes per iteration.

B-6 Example #6

Example #5 demonstrated the effectiveness of the structural load relief optimization phase of COEBRA when the initial autopilot met all of the margin requirements. Example #6 was run to see how the load relief phase performs when

the initial autopilot does not meet the margin requirements.

The initial autopilot for this COEBRA run was obtained as follows. In several booster autopilots, there is a feedback loop that is used soley for high frequency stabilization. This loop is "washed out" at frequencies below the rigid-body phase margin, and serves to compensate for the load relief loop gain at high frequencies so that the load relief loop gain can be increase. So for Example #6, a previously designed autopilot case was chosen, and this "high frequency" loop was zeroed out. This resulted in unacceptable stability margins. The design objective for the COEBRA run was to not only return to the condition where all margins are met, but also to achieve at least the same amount of load relief that was achieved with the engineer's original autopilot that used this high frequency feedback loop.

For example #6, three flight conditions were designed together: (1) load relief switch-in; (2) max-$\bar{q}$; and (3) load relief loop switch-out. The airframe included four bending modes at two of the time points, and three modes at the third.

Figure B-17 is a block diagram of the airframe/autopilot system. This is an analog autopilot design problem, and therefore the design is performed in the S-plane. There are 14 autopilot variables, but since this is an analog autopilot, each of these variables must have the same value at all three time points.

Figure B-18 is the open-loop frequency response plot that resulted from COEBRA's initial autopilot at the max-$\bar{q}$ flight condition. It shows that not all margin requirements are met. This same situation exists at the other two flight conditions (not shown).

The load relief indicator for the engineer's original autopilot that used the so-called high frequency feedback loop was 4490 pounds per square foot. This result was obtained from a 6 DOF trajectory simulation. COEBRA's first step

was to "get feasible," but in so doing, it had to give up load relief capability. COEBRA met all margins after three iterations, but $q\alpha$ (from a 6 DOF simulation) increased to 4580 pounds per square foot. However, from the 4th to the 8th iteration, all margin requirements remained satisfied, and $q\alpha$ began decreasing, until on the 8th and final iteration, it had decreased to 3975 pounds per square foot. Again, this was obtained from a 6 DOF simulation, and this $q\alpha$ was 12% less than that of the original autopilot with the high frequency feedback loop. Figure B-19 does show that all margins are met at the max $\bar{q}$ time point with the results of the 8th and final COEBRA iteration. The same situation existed at the other two flight times.

The following discussion refers to Section 2.9 of Chapter 2 on Convergence to an Exterior Optimum. This example has dramatically demonstrated how the COEBRA algorithm converges to a constrained optimum from an unfeasible initial point. The first three iterations were required in order to reach a feasible solution. In "getting feasible," load relief capability was reduced. This did not necessarily have to happen, since the algorithm does try to optimize while "getting feasible." Once the feasible region was reached, the algorithm moved along or parallel to the constraint boundaries until the constrained optimum was reached. The fact that this actually occurred is known because load relief capability steadily increased from the 4th to the 8th iteration, while several stability margins remained "tight against" their requirements. Two of these "tight margins" can be seen in Figure B-19. These two margins are called the rigid-body phase margin, and the phase margin on the "backside" of the first structural bending mode. These margins are indicated by arrows in Figure B-19. These margins were tight after three iterations, and remained tight from the 4th to the 8th and last iteration.

Table B-8 summarizes the results of Example #6. This table shows that

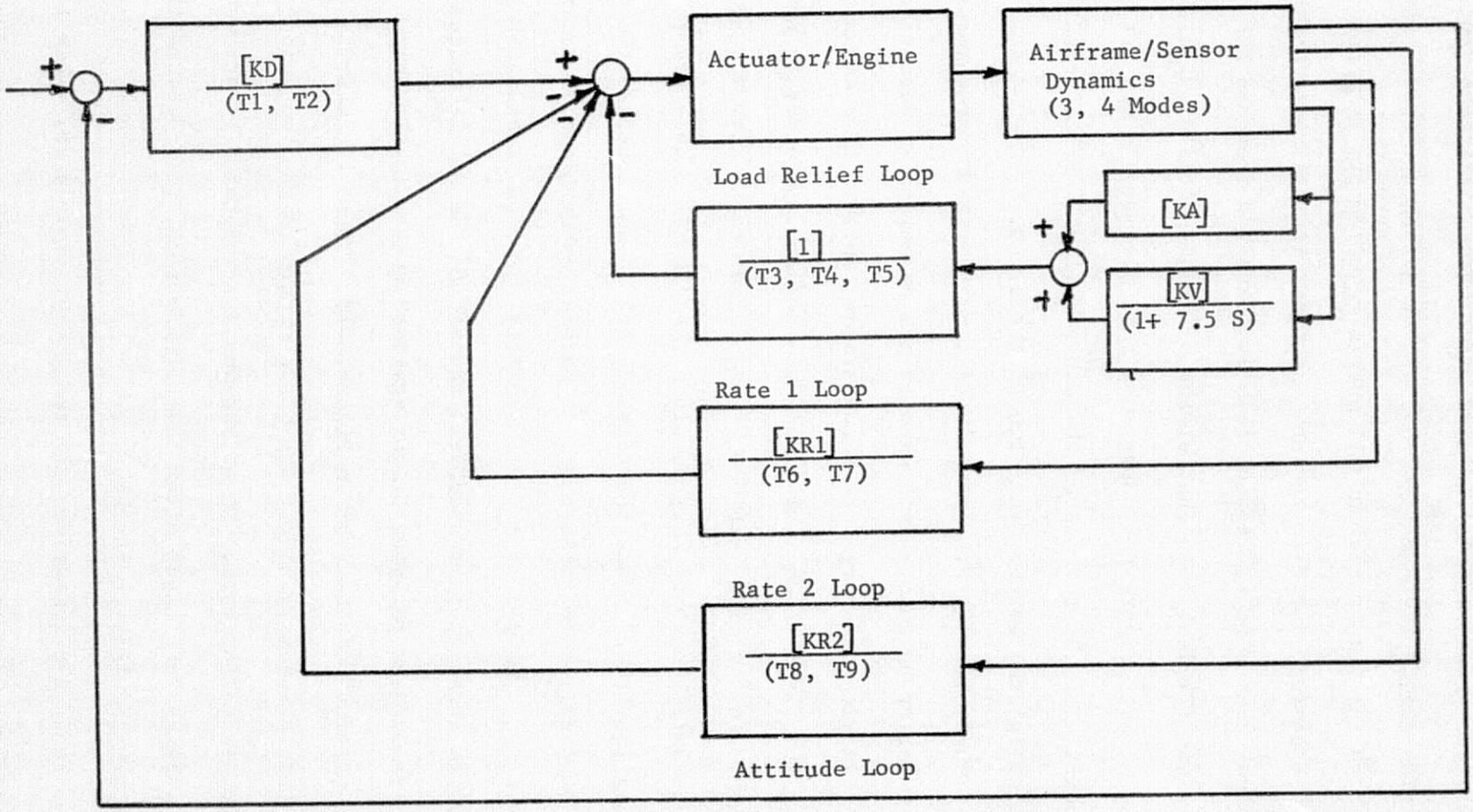


Figure B-17. System Block Diagram for Example #6

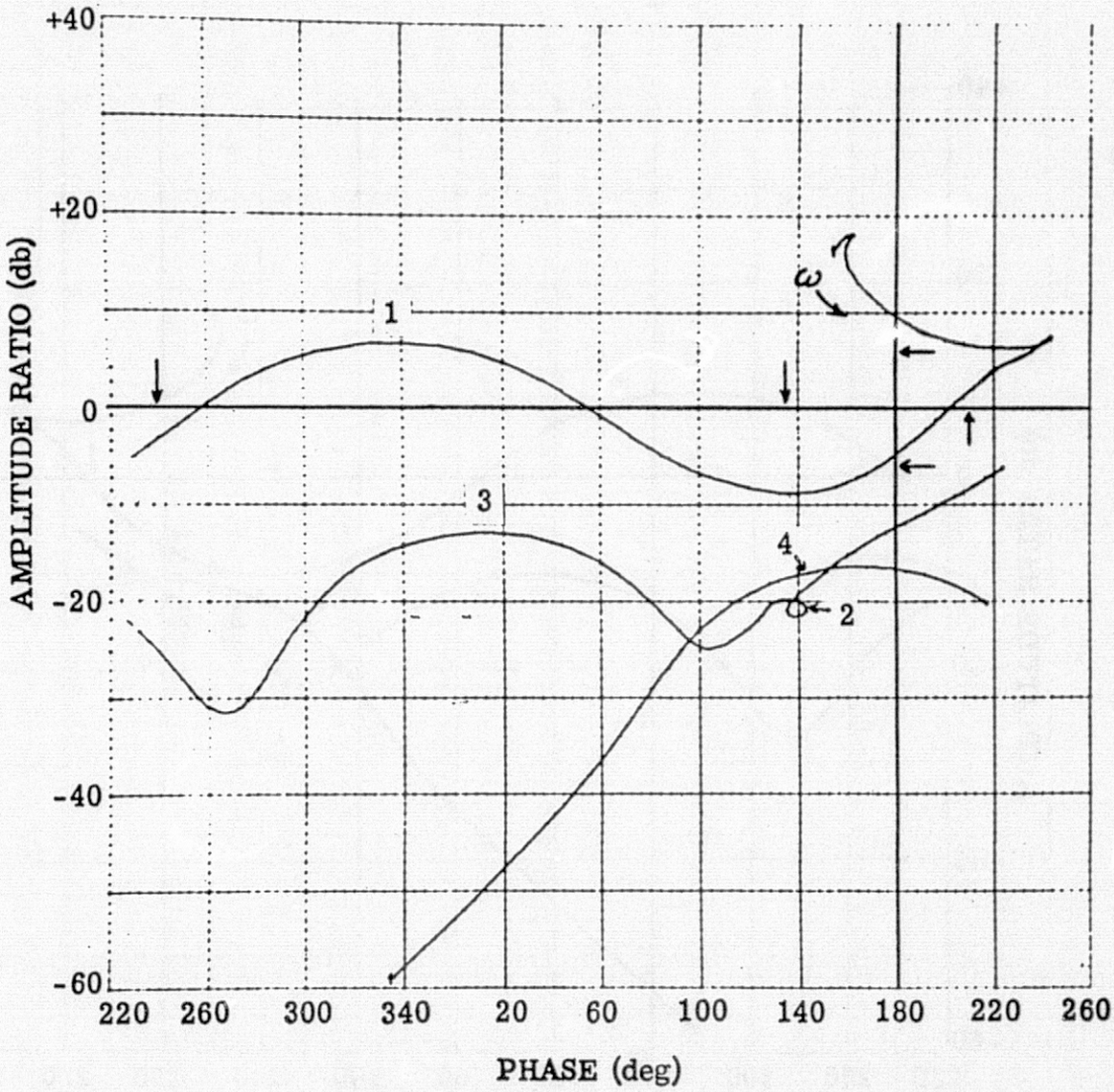


Figure B-18 Example #6, Gain-Phase Frequency Response Plot Resulting from Initial Autopilot at Max-$\bar{q}$

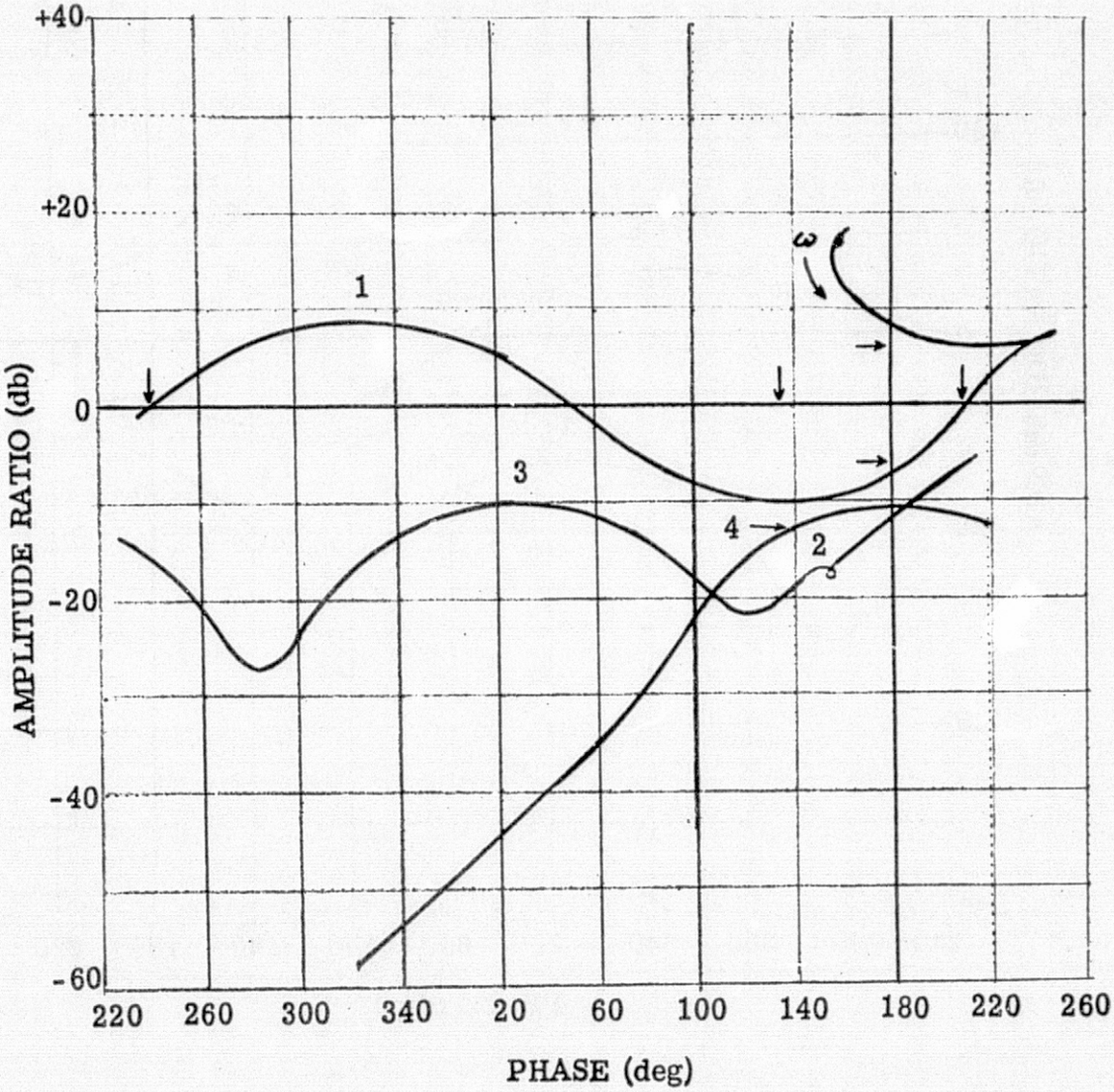


Figure B-19. Example #6, Gain-Phase Frequency Response Plot Resulting From Eighth (Final) Iteration at Max-q

Table B-8. Example #6 Summary of Results

- Objective: Maximize Load Relief Capability

Autopilot	Stability Margins	q_a
Engineer's Final	Satisfactory	4490
COEBRA Initial	Unsatisfactory	4490
Third Iteration	Satisfactory	4580
Eighth (Final) Iteration	Satisfactory	3975

- System Order:
  - o 25th Order at 2 Time Points
  - o 23rd Order at 1 Time Point
  - o 14 Autopilot Variables Which Must Have Same Value at Each Time Point
- 8 Major Iterations
- Computer Time: 59.2 Minutes or 7.4 Minutes per Iteration

the computer time for this example was 59.2 minutes, or 7.4 minutes per iteration.

B-7 Example #7

Example #7 illustrates using COEBRA to design a load relief autopilot in two phases: (1) the initial phase being to first meet all margin requirements; (2) the second phase being to optimize load relief capability. For this example, this approach was considered essential because the "first guess" or initial autopilot was very poor.

This example is taken from a recent effort to design an autopilot for a space shuttle booster configuration. COEBRA was used to design the autopilot for all three channels (pitch, yaw, and roll) at all the critical flight conditions during the first two minutes of ascent. At all the flight times, the airframe included from seven to eight structural bending modes.

The flight condition for this example is the yaw channel during the max-$\bar{q}$ portion of flight. Three time points were designed together: (1) load relief switch-in; (2) max-$\bar{q}$; and (3) load relief switch-out. The airframe included seven modes at each of the time points. While all the results obtained from this design effort are worth noting, this example was selected since it illustrates the two phased approach to load relief autopilot design.

Figure B-20 is the airframe/autopilot block diagram for this example. It shows the attitude loop, a rate loop, and the load relief loop. In addition, it shows an attitude acceleration loop. This is the so-called high frequency loop that was referred to in Example #6. Figure B-18 shows that there are 20 autopilot variables at each time point. The four gains can have different values at each time point, but the 16 filter parameters must have the same value for all the time points.

Figures B-21, 22 and B-23 show the frequency response plots for the

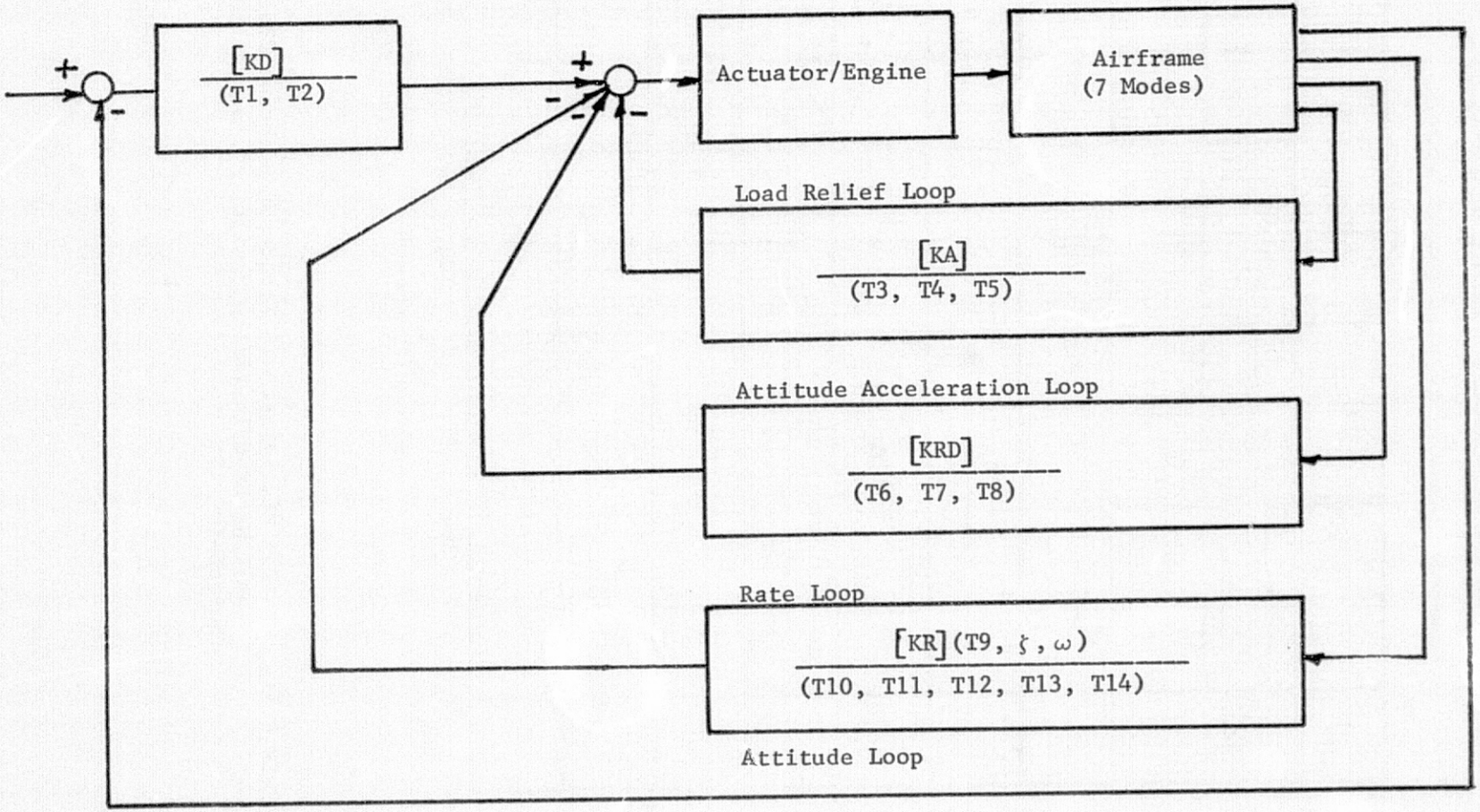


Figure B-20 System Block Diagram for Example #7

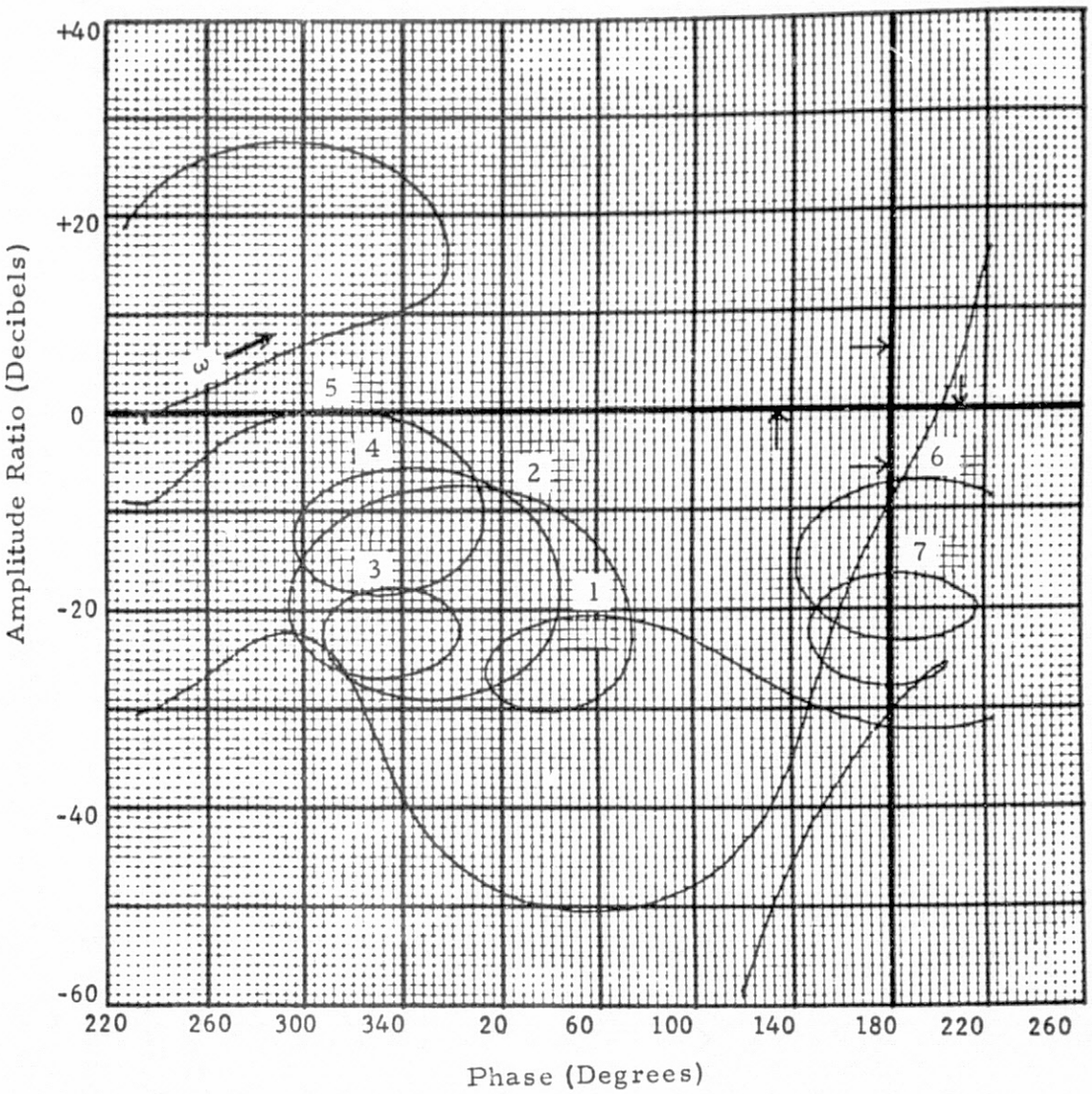


Figure B-21. Example #7, Gain-Phase Frequency Response Plot Resulting From Initial Autopilot at Load Relief Switch-In

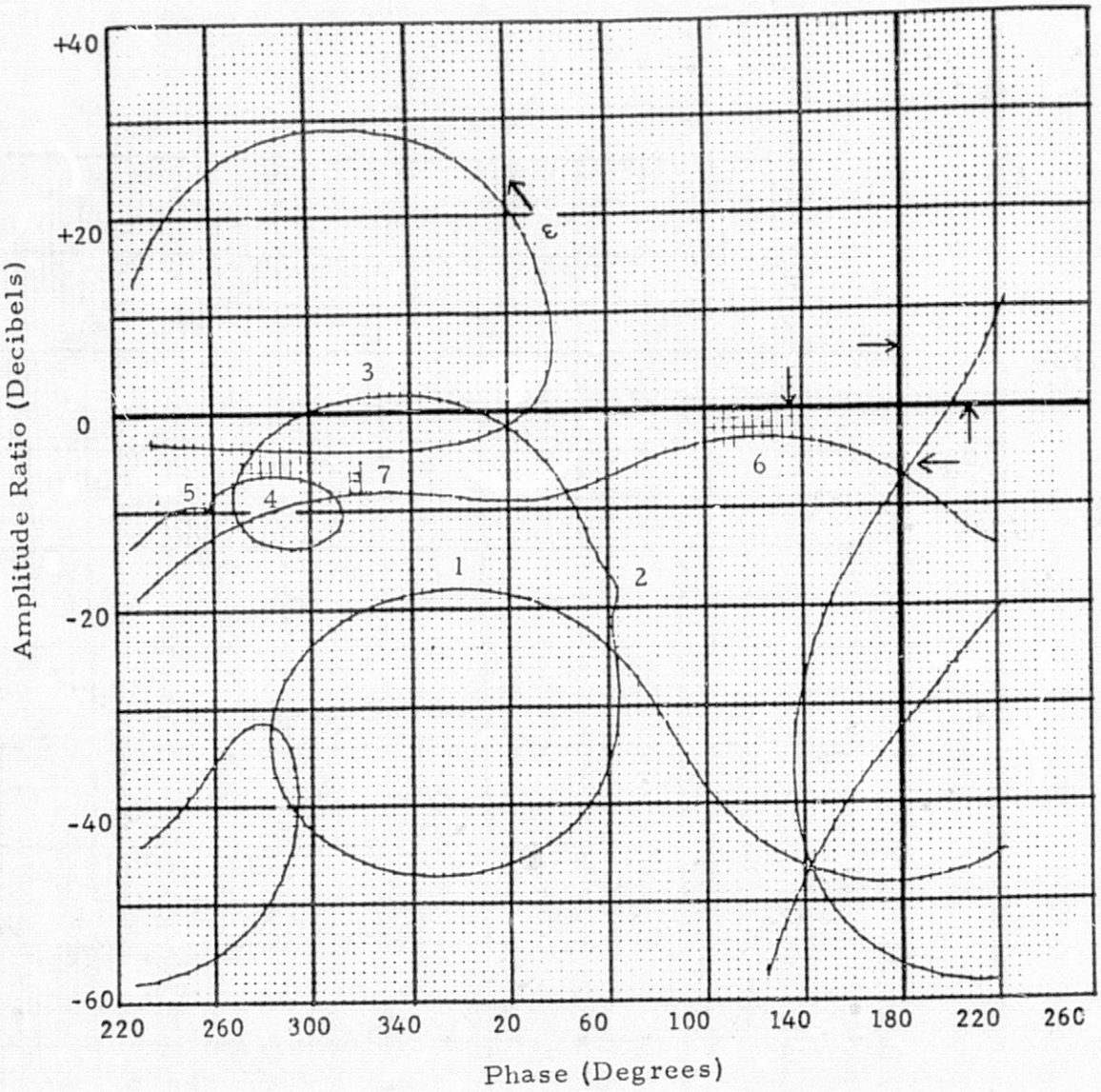


Figure B-22. Example #7, Gain-Phase Frequency Response Plot Resulting From Initial Autopilot at Max-$\overline{q}$

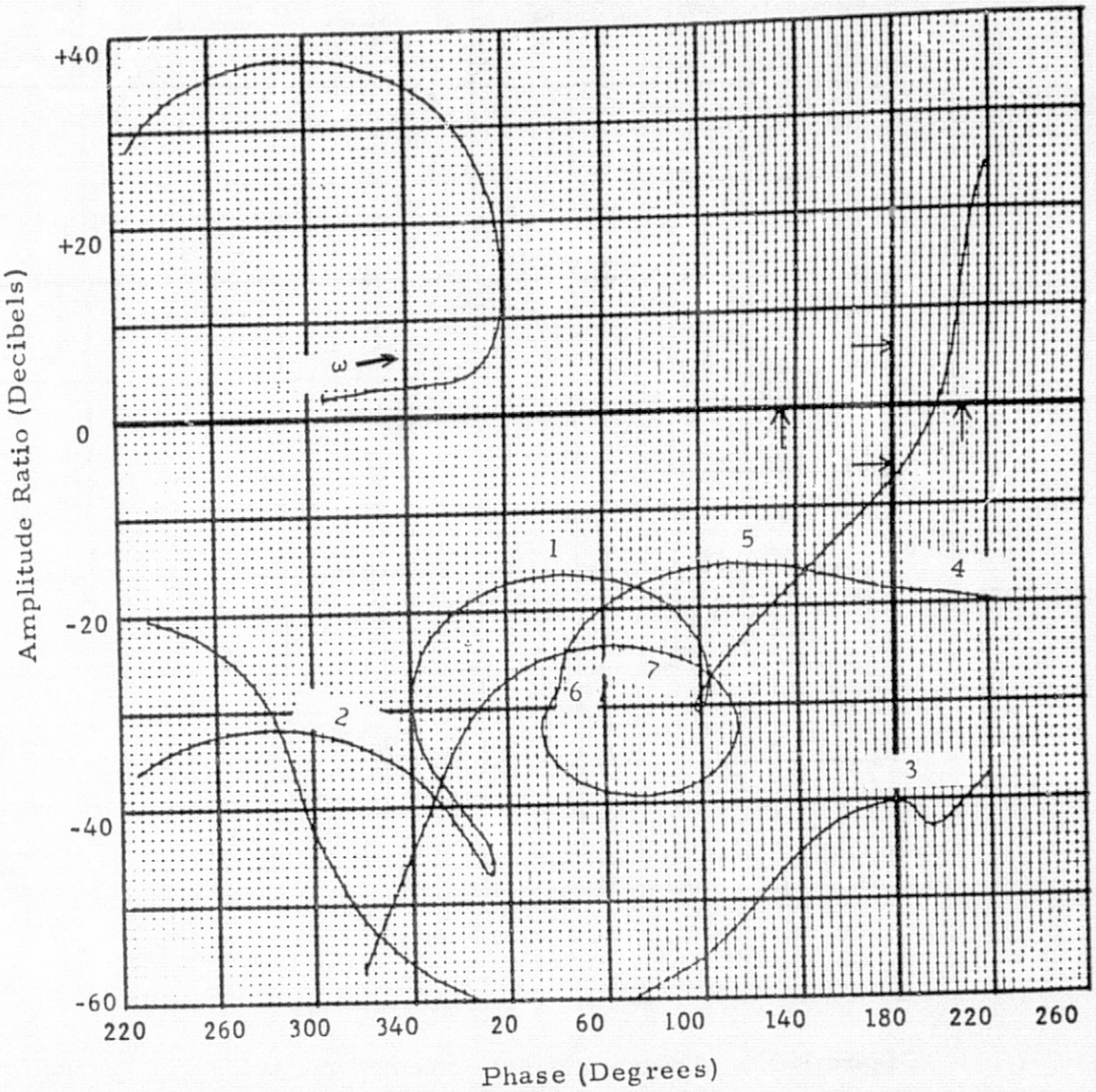


Figure B-23. Example #7, Gain-Phase Frequency Response Plot Resulting From Initial Autopilot at Load Relief Switch-out.

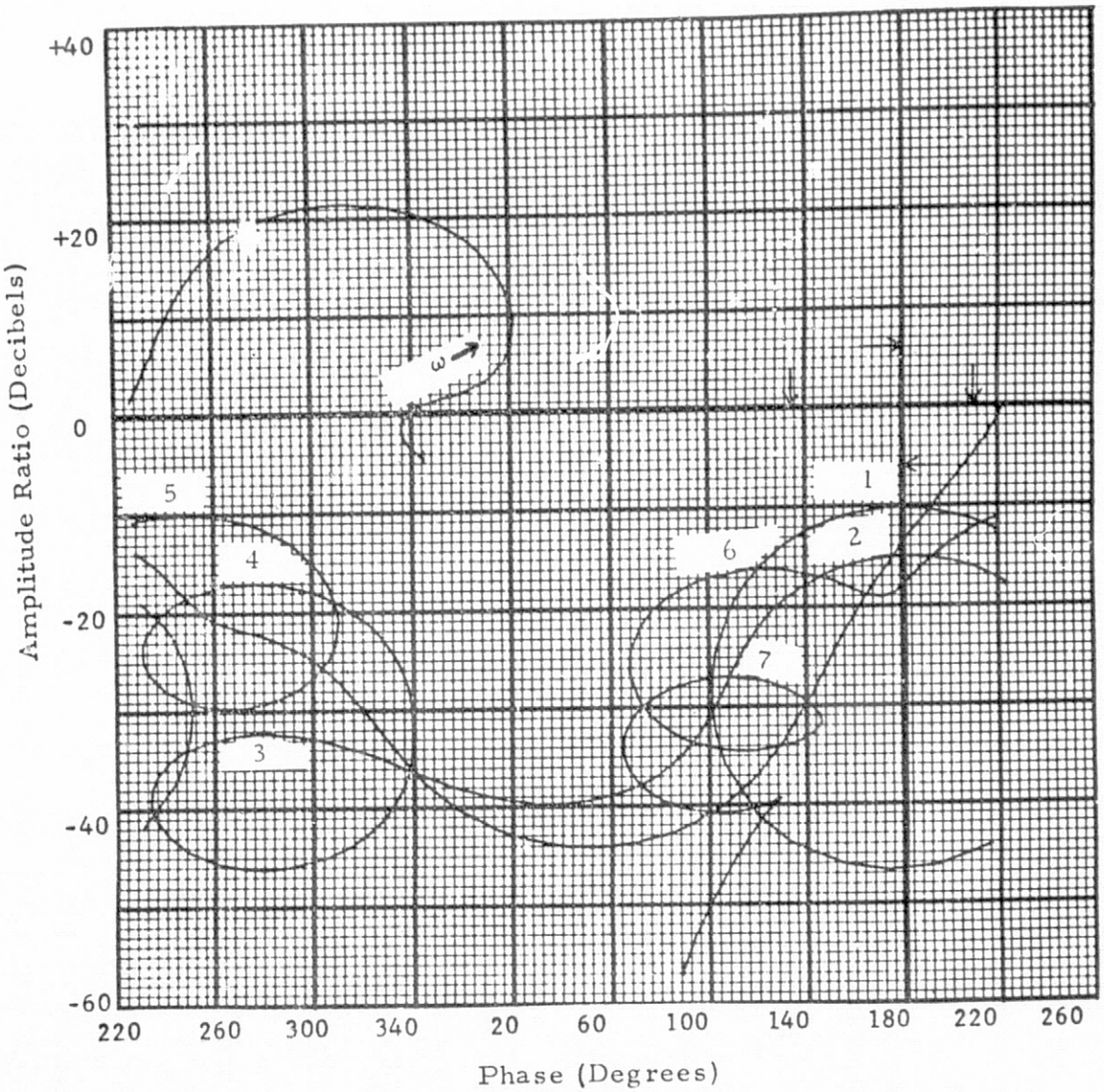


Figure B-24. Example #7, Gain-Phase Frequency Response Plot Resulting From Seventh (Final) Iteration at Load Relief Siwtch-in

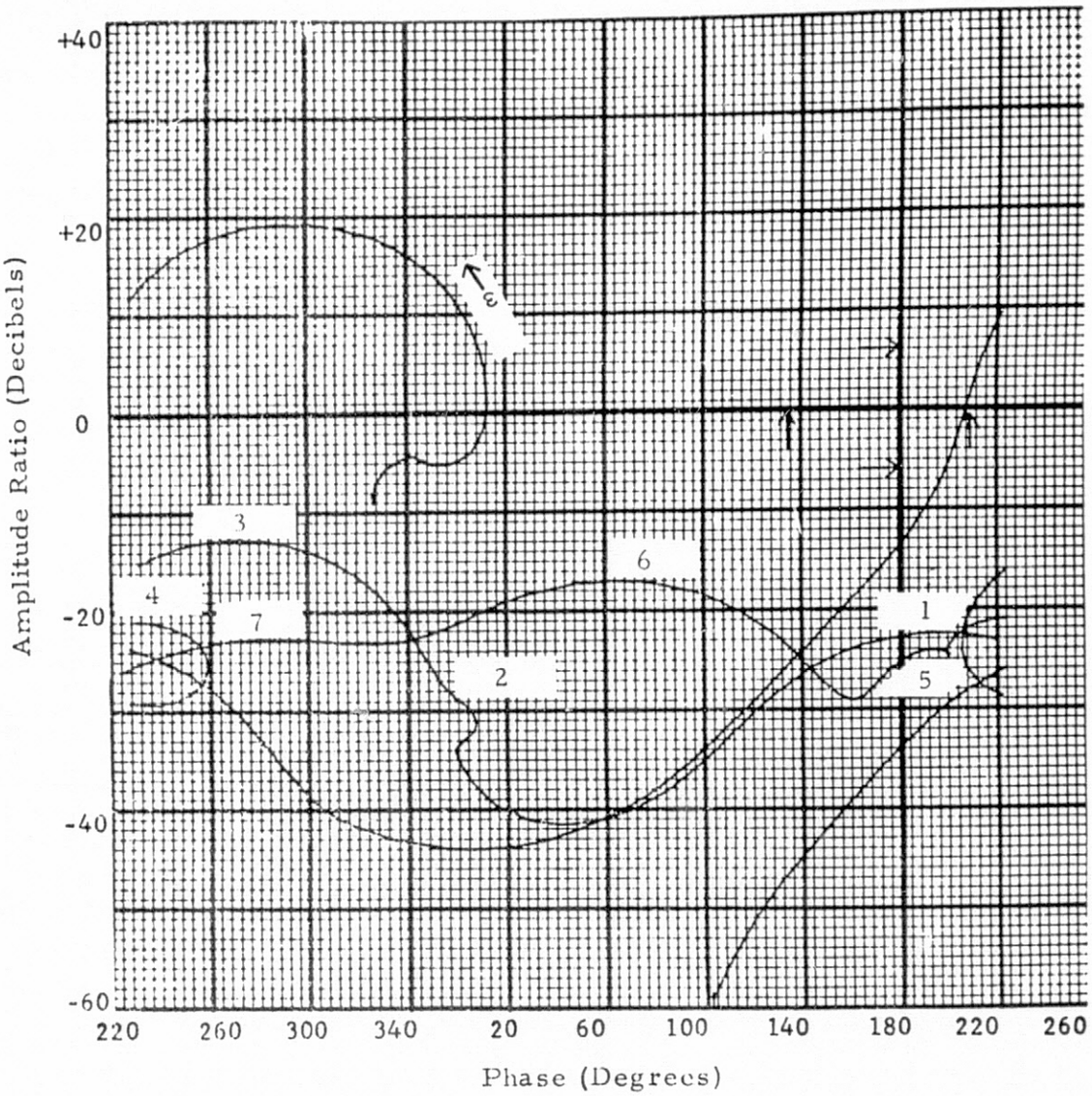


Figure B-25 Example #7, Gain-Phase Frequency Response Plot Resulting From Seventh (Final) Iteration at Max-$\bar{q}$

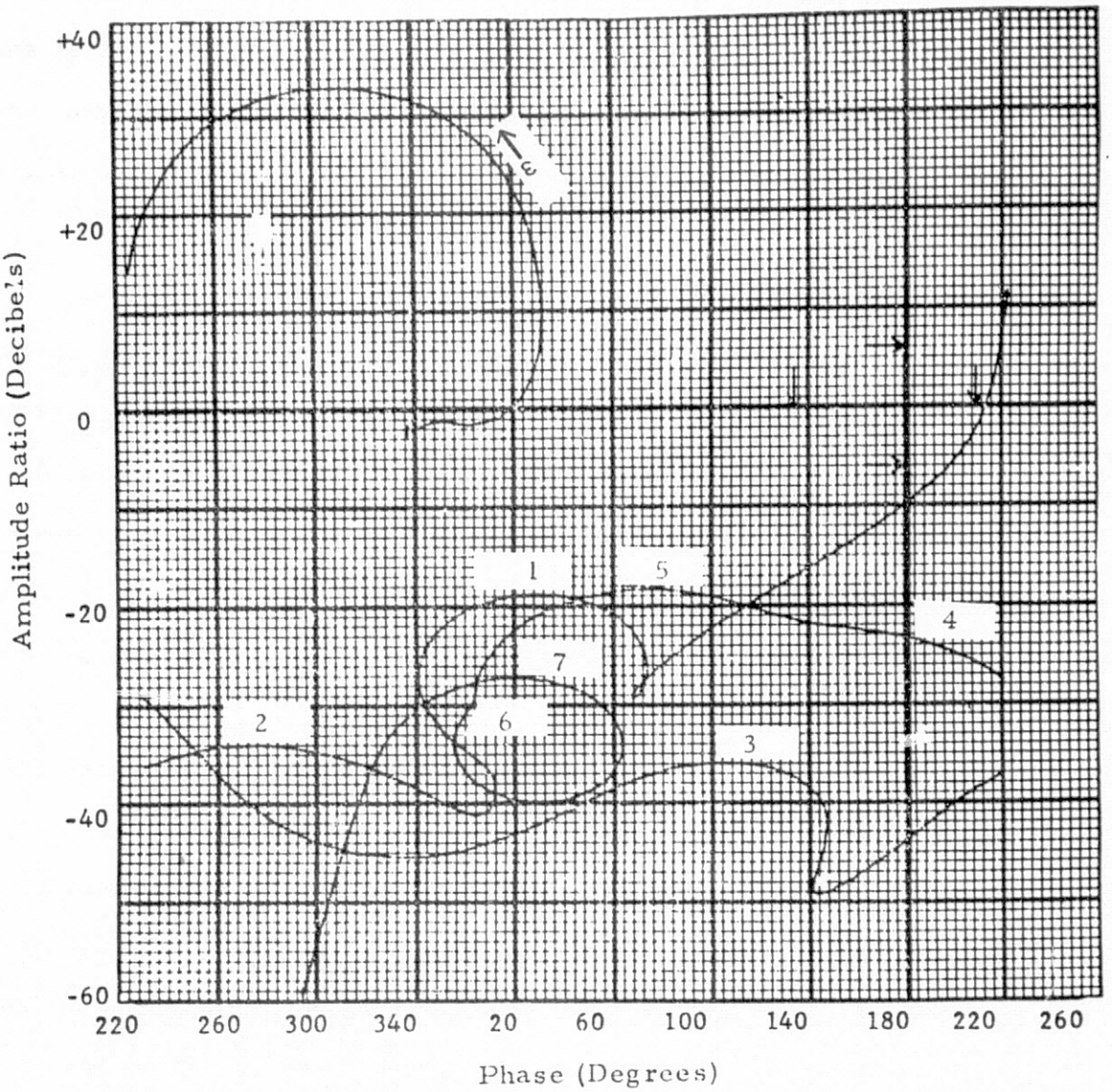


Figure B-26. Example #7, Gain-Phase Frequency Response Plot Resulting From Seventh (Final) Iteration at Load Relief Switch-out

initial autopilot. The system is stable, but the initial autopilot is very poor. The basic margin requirement is that modes 3 through 7 be gain stabilized with a peak amplitude below "-10" decibels. Only the first and second modes can be phase stabilized, but if they are gain stabilized, their so-called "closest approach" distance to the "-1" point must be equivalent to 10 decibels.

The first COEBRA run was made to optimize stability margins. After one iteration, all margins were met. The next COEBRA runs were made to optimize load relief. After six more iterations, an autopilot resulted that yielded the plots of Figure B-24, 25 and B-26. All margin requirements are met. A 6 DOF trajectory simulation was not made, but estimates based on linear transient response results indicate that bending moment loads were reduced 25% from the initial to the final autopilot. Computer time required to do this job was 98 minutes, or 14 minutes per iteration.

This example points to another way in which the COEBRA algorithm can be used. By observing the progress it is able to make from iteration to iteration, it can be used to design a minimum-complexity autopilot. For this example, the fact that COEBRA was able to satisfy all margin requirements in only one iteration, tends to indicate that some of the degrees of freedom in the autopilot could probably be eliminated.

B-8 Conclusions

The results presented in this appendix clearly demonstrate COEBRA's ability to successfully design autopilots for large flexible launch vehicles. Experience with the program shows that while it generally does not save computer time, it does save manpower and the time required to design an autopilot.

APPENDIX C

RANDOM FORCING FUNCTIONS

In the foregoing analysis, though not explicitly stated, it has been assumed that the vehicle forcing functions were deterministic in nature. For example, the wind forcing function has been, in general, implemented by means of a pre-programmed wind profile based on a deterministic combination of steps and ramps. A more general characterization of the wind profile would be as a stochastic or random process. The vehicle behavior under the influence of such a forcing function can significantly affect the performance measures. Furthermore, many deterministic signals have random disturbances or noise superimposed on them.

COEBRA has incorporated in its repertoire the capability to examine the effects of a random or noise forcing function. The prime occurrence in our case is that of a random wind profile. Consider, for example, the structural bending moment load, $l(t)$, existing at a specific sensor location on the vehicle. This load can be expressed as

$$l(t) = K_\alpha \alpha(t) + \sum_i K_i \delta_i(t) \qquad \text{C-1}$$

where α is the vehicle angle of attack and δ_i is the control actuator function. The approach adopted by COEBRA is to constrain or optimize the mean square value of $l(t)$ resulting from the random forcing function.

Once again we call upon the open literature for a detailed exposition of the random process problem [40, 41, 42, 42]. The fundamental mathematical tools and the two-sided Laplace transform and the Fourier transform. Following the pattern of Newton, Gould & Kaiser,[40] we make no formal distinction between them other than that $s = j\omega$ in the following. Let the vehicle or system function be

$$H(s) = \mathcal{L}\left(\frac{l(t)}{f(t)}\right) \qquad \text{C-2}$$

where $f(t)$ is the random forcing function and $H(s)$ thus contains the control

equations and vehicle dynamics or simply those system parameters to be considered or optimized.

Associated with the system represented by H(s) and, of course, the random forcing function f(t) and the system output, l(t). The convolution integral yields

$$l(t) = \int_{-\infty}^{\infty} h(\tau)\, f(t-\tau)\, d\tau \qquad \text{C-3}$$

Introducing the concept of correlation functions, ensembles and the hypothesis as fundamental properties of systems subject to random forcing functions, we treat first the concept of an ensemble. To do so, we follow the influence of Newton, Gould and Kaiser [40] to first make a subtle distinction between a random process and a stochastic process as used herein. A random process is one describable only in terms of its statistical properties. For our purpose, a stochastic process or signal is one that exhibits a degree of randomness but is not exclusively random; i.e., perhaps a sinusoid accompanied by noise. An "ensemble" is considered to be a set of stochastic signals, each generated by an identical process. Viewed from the time domain point of view, the appearance of each element of an ensemble over the same time span would be different but the properties of the ensemble can be identified. For example, the average value of the product of a signal at a specified time, b_1, multiplied by the value of the same signal at a different time, $b_1 + \tau$, is known as the ensemble average or auto correlation function, defined by

$$\varphi_{vv}(t,\tau) \triangleq \overline{v(t_1)v(t_1+\tau)} \qquad \text{C-4}$$

If we denote the ensemble average of the stochastic signal, $v(t)$ as

$$\overline{v(t_1)} \triangleq \lim_{N\to\infty} \frac{1}{N}\sum_{j=1}^{N} v_{ij} \qquad \text{C-5}$$

where v_{ij} is one element of the ensemble $v(t)$, and the time average of the function

$$\bar{v}(t) = \lim_{T\to\infty} \frac{1}{2T}\int_{-\infty}^{\infty} v(t)\, dt \qquad \text{C-6}$$

we observe that if

$$\widetilde{v(t_1)} = \overline{v(t)}$$

the signal $v(t)$ can be considered ergodic, a usual assumption when considering engineering systems. Using this ergodic hypothesis we can rewrite equation C-4 as

$$\varphi_{vv}(\tau) = \overline{v(t)v(t+\tau)} = \lim_{T\to\infty} \frac{1}{2T}\int_{-\infty}^{\infty} v(t)v(t+\tau)\,dt \qquad \text{C-7}$$

Returning now to our autopilot problem, defined by H(s), we note that

$$\varphi_{ll}(\tau) = \int_{-\infty}^{\infty} h(t_1)\int_{-\infty}^{\infty} h(t_2)\,\varphi_{ff}(\tau+t_1-t_2)\,dt_2\,dt_1 \qquad \text{C-8}$$

where φ_{ff} is the autocorrelation function of the stochastic forcing function f(t). We now consider a frequency function related to the correlation function,

$$\Phi_{ll}(s) \triangleq \frac{1}{2\pi}\int_{-\infty}^{\infty} \varphi_{ll}(\tau)\,e^{-st}\,dt \qquad \text{C-9}$$

Recall now, equation C-8, multiplying both sides by e^{-st} and integrating on t from $-\infty$ to ∞ we have

$$\int_{-\infty}^{\infty} \varphi_{ll}(\tau)\,e^{-s\tau}\,d\tau = \int_{-\infty}^{\infty} e^{s\tau}\,d\tau \int_{-\infty}^{\infty} h(t_1)\,dt_1 \int_{-\infty}^{\infty} h(t_2)\,\varphi_{ff}(\tau+t_1-t_2)\,dt_2 \qquad \text{C-10}$$

Now changing the order of integration on the right of C-10 so that we integrate with respect to τ first, and adjusting the argument of the exponential to agree with that of the correlation function, we obtain

$$\int_{-\infty}^{\infty} \varphi_{ll}(\tau)\,e^{-s\tau}\,d\tau = \int_{-\infty}^{\infty} h(t_1)\,dt_1 \int_{-\infty}^{\infty} e^{-st_2}h(t_2)\,dt_2 \int_{-\infty}^{\infty} e^{-s(\tau+t_1-t_2)}\,\varphi_{ff}(\tau+t_1-t_2)\,d\tau \qquad \text{C-11}$$

But this is just

$$\Phi_{\ell\ell}(s) = H(s)H(-s)\Phi_{ff}(s) \qquad \text{C-12}$$

since

$$H(s) = \int_{-\infty}^{\infty} h(t)e^{-st}dt$$

Now, since $\Phi_{\ell\ell}(s)$ given by C-9 is in truth the Fourier transform of $\phi_{\ell\ell}(\tau)$, we can write

$$\varphi_{\ell\ell}(\tau) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \Phi_{\ell\ell}(s)e^{s\tau}ds \qquad \text{C-13}$$

The integration is along the imaginary axis of the s-plane enabling us to write the inverse transform in terms of the real frequency ω or

$$\varphi_{\ell\ell}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_{\ell\ell}(j\omega)e^{j\omega\tau}d\omega \qquad \text{C-14}$$

Letting $\tau = 0$, we observe

$$\varphi_{\ell\ell}(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_{\ell\ell}(j\omega)d\omega \qquad \text{C-15}$$

But this is, from C-7, simply the mean-square value of l(t), the structural bending moment load.

Thus, using C-12, C-15 and the knowledge of the frequency function $\Phi_{ff}(s)$ we are able to make use of a constraint on the mean-square value, as the case arises, to assist in the design of the autopilot.

A word or two about the frequency function $\Phi_{ff}(s)$. Consider, for a moment, that f(t) is a stochastic voltage impressed on a 1 ohm resistor. The mean-square value of f(t) then represents the average power dissipated in the

resistor and $\Phi_{ff}(j\omega)$ represents the power-density spectrum of the signal f(t). Common practice refers to any frequency function obtained through the Fourier transform of an auto correlation function as a power-density spectrum.

Using the COEBRA program, we can determine, as before, the gain constants and filter parameters such that $1^2(t)$, or a similar random response function is constrained and/or minimized. In other words, with Φ_{ff} specified by the user COEBRA will compute $I^2(t)$ and H(s) from the total dynamics.

APPENDIX D

INPUT FORMATS

There are several fundamental restrictions on the COEBRA design algorithm. If analysis is to be carried out in the frequency domain with block diagram format, the autopilot configuration must meet that of Figure 1-7 (the QD030 subroutine). Within limits simple block diagram transformulations should lead to an equivalent structure of this form as illustrated in Figures D-1 and D-2. In the event that the initial configuration is such that the block diagram transformations appear to be unwieldly or tend to obscure the effects of certain autopilot parameters an alternate procedure has been provided. This procedure is closely related to the state variable description of a control system. However, one should <u>not</u> construe, in what follows, that state variable techniques are being applied. Referring to Figure D-1 the reader will note the identification of input and output signals, x_i, to the individual blocks or transfer functions. This identification is the first step in the formulation at the so-called matrix input description for COEBRA. Before going any further, we observe the restrictions on the elements of the matrix to be developed. Each element is to be limited to a second order polynomial ratio in S. The matrix for a single flight condition can have only <u>one</u> active forcing function.

If analysis is to be carried out in the time domain, the matrix format must be used. This leads to a somewhat restrictive definition of a flight condition, as illustrated in Figure D-3. A new COEBRA input must be supplied for each flight condition.

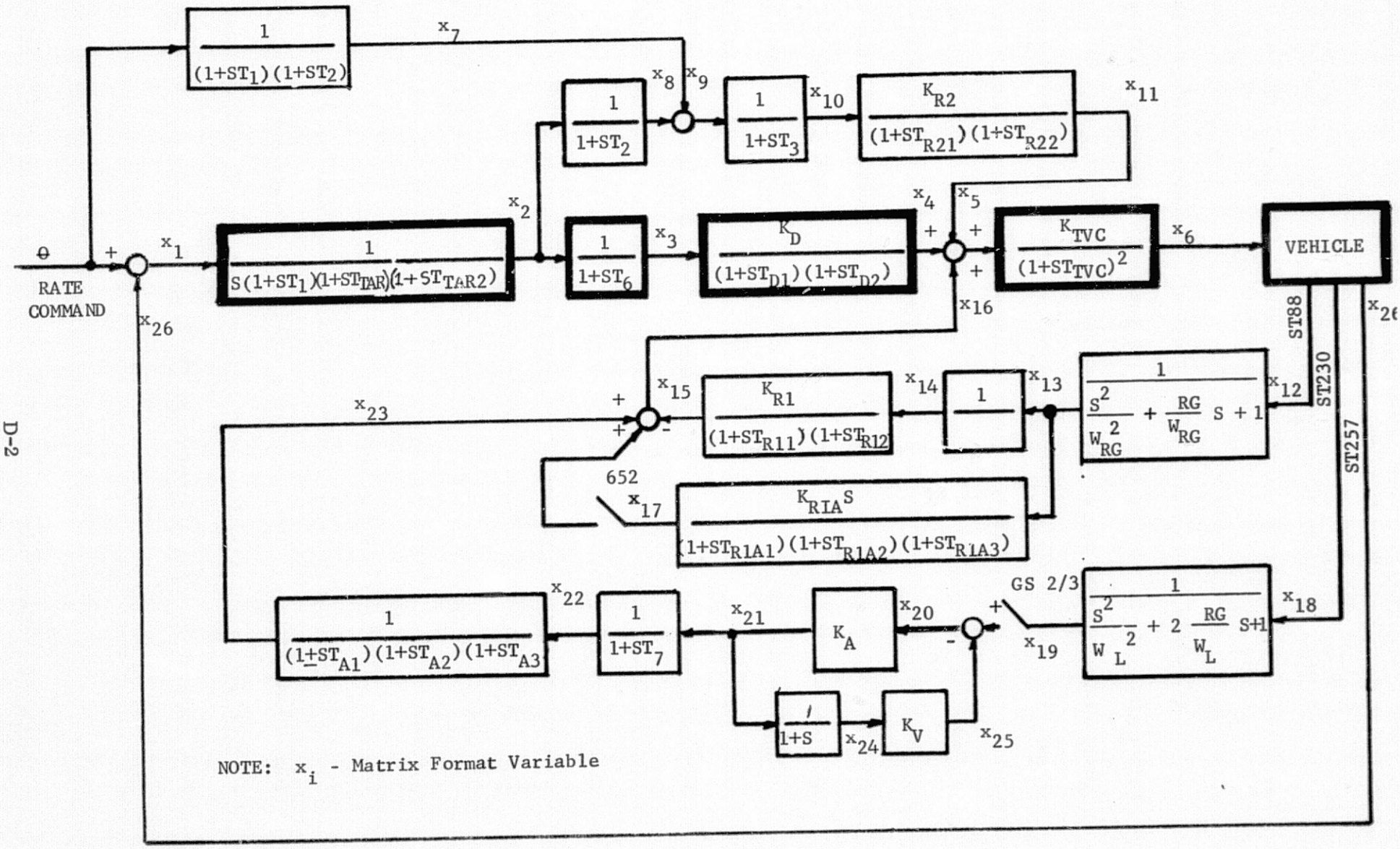
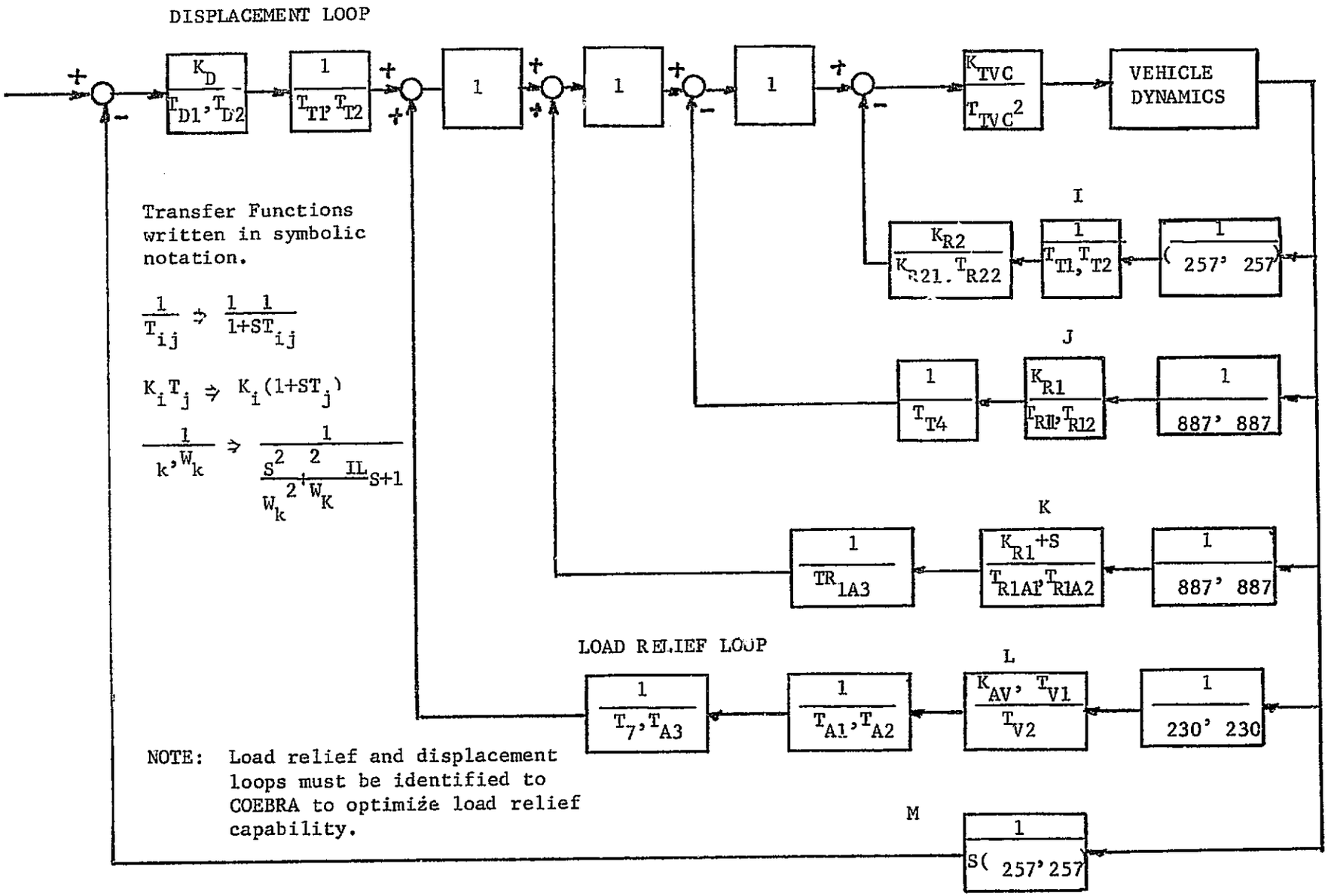


FIGURE D-1 TYPICAL AUTOPILOT FOR BOOSTER PITCH CONTROL



Transfer Functions written in symbolic notation.

$$\frac{1}{T_{ij}} \Rightarrow \frac{1}{1+ST_{ij}}$$

$$K_i T_j \Rightarrow K_i(1+ST_j)$$

$$\frac{1}{k, W_k} \Rightarrow \frac{1}{\frac{S^2}{W_k^2} + \frac{2}{W_K} IL S+1}$$

NOTE: Load relief and displacement loops must be identified to COEBRA to optimize load relief capability.

FIGURE D-2 TYPICAL AUTOPILOT IN COEBRA BLOCK DIAGRAM FORMAT

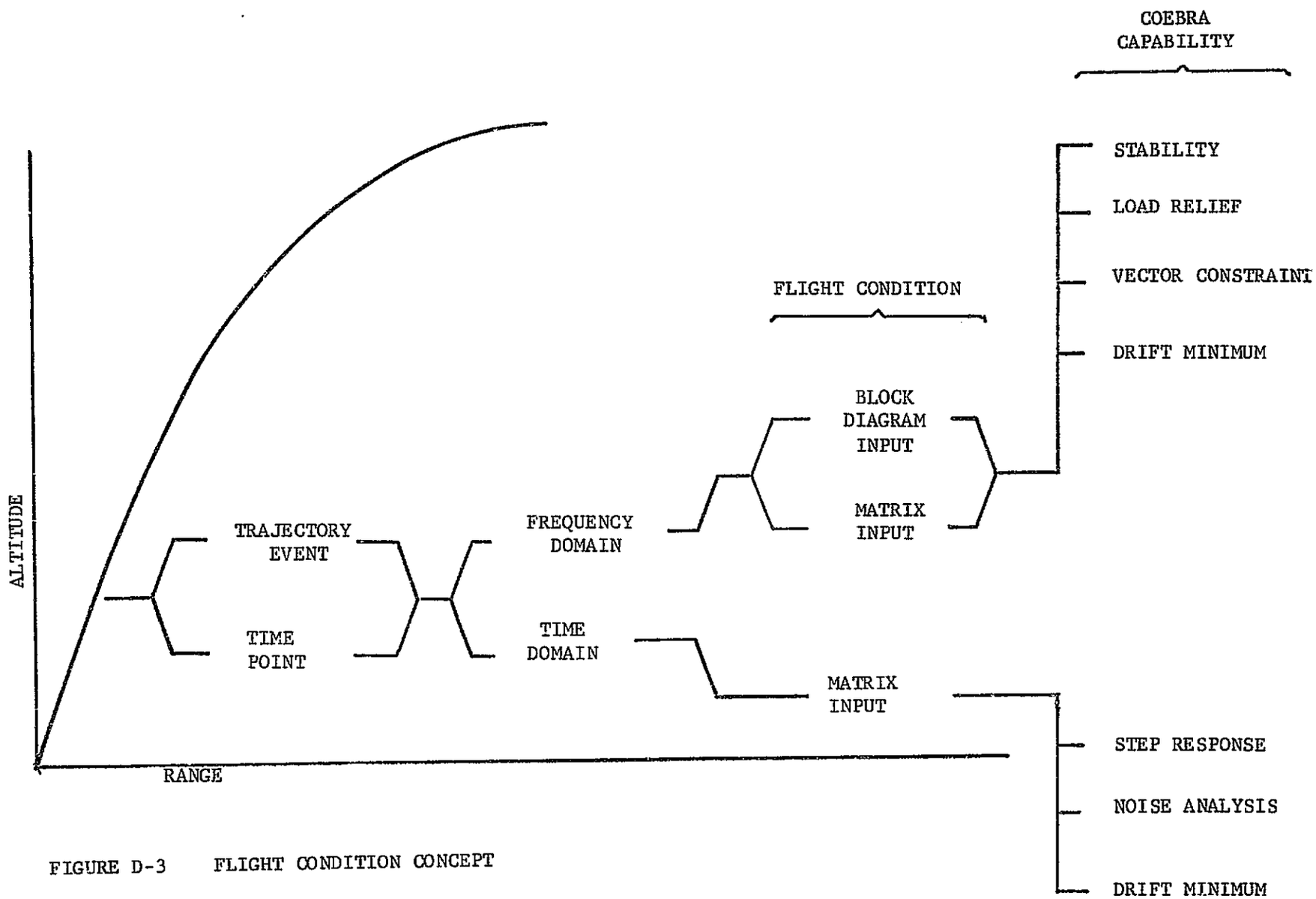


FIGURE D-3 FLIGHT CONDITION CONCEPT

Examining Figure D-1 we can write the following set of equations involving only second order functions in s by the introduction of slack variables and formatted in the matrix notation us.

$$Ax = B\dot{\theta} \quad \text{D-1}$$

where the vector B is comprised of a column of zeros exclusive of a single forcing function element. The solution of this equation yields the time response of the autopilot/vehicle system.

$$x_1 = \dot{\theta} - x_{26}$$

$$x_{2A} = \frac{1}{S(1+ST_1)} x_1$$

$$x_2 = \frac{1}{(1+ST_{TAR1})(1+ST_{TAR2})}$$

$$x_3 = \frac{1}{1+ST_6} x_2$$

$$x_4 = \frac{K_D}{(1+ST_{D1})(1+ST_{D2})} x_3$$

$$x_5 = x_4 + x_{11} + x_{16}$$

$$x_6 = \frac{K_{TVC}}{(1+ST_{TVC})^2}$$

$$x_7 = \frac{1}{(1+ST_1)(1+ST_2)} \dot{\theta}$$

$$x_8 = \frac{1}{1+ST_2} x_2$$

$$x_9 = x_8 - x_7$$

$$x_{10} = \frac{1}{1 + ST_3} x_9$$

$$x_{11} = \frac{K_{R2}}{(1+ST_{R21})(1+ST_{R22})} x_{10}$$

$$x_{12} = \frac{1}{\frac{S^2}{W_{88}^2} + \frac{2\xi_{88}}{W_{88}} S + 1} \underline{T}_{88} \cdot x_6$$

$$x_{13} = \frac{1}{\frac{S^2}{W_{RG}^2} + \frac{2\xi_{RG}}{W_{RG}} S + 1} x_{12}$$

$$x_{14} = \frac{1}{1 + ST_4} x_{13}$$

$$x_{15} = \frac{K_{R1}}{(1 + ST_{R11})(1 + ST_{R12})} x_{14}$$

$$x_{16} = -x_{15} + x_{17} + x_{23}$$

$$x_{17A} = \frac{K_{R1A} S}{(1 + ST_{R1A1})(1 + ST_{R1A2})} x_{13}$$

$$x_{17} = \frac{S}{1 + ST_{R1A3}} x_{17A}$$

$$x_{18} = \frac{S}{\frac{S^2}{W_{230}^2} + \frac{2\xi_{230}}{W_{1\,230}} S + 1} \underline{T}_{230} \cdot x_6$$

$$x_{19} = \frac{1}{\frac{S^2}{W_L^2} + 2\frac{\xi_L}{W_L} S + 1} x_{18}$$

$$x_{20} = x_{19} - x_{25}$$

$$x_{21} = K_A x_{20}$$

$$x_{22} = \frac{1}{1 + ST_7} x_{21}$$

$$x_{23A} = \frac{1}{(1 + ST_{A1})(1 + ST_{A2})} x_{22}$$

$$x_{23} = \frac{1}{1 + ST_{A3}} x_{23A}$$

$$x_{24} = \frac{\tau}{1 + S\tau} x_{21}$$

$$x_{25} = K_V x_{24}$$

$$x_{26} = \frac{1}{\frac{S^2}{W_{251}^2} + 2 \frac{257}{W_{257}} S + 1} \overset{T}{=} 257 x_6$$

In matrix notation this becomes

$$
\begin{bmatrix}
a_{1} & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
a_{21} & a_{22} & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & a_{32} & a_{33} & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & a_{43} & a_{44} & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & a_{54} & a_{55} & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & a_{65} & a_{66} & & & & a_{6,12} & & & & & a_{6,17} & & & & & & & & & & & & & & & \\
 & & & & & a_{76} & a_{77} & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & 0 & a_{88} & & & & & & & & & & & & & & & & & & & & & & \\
 & & a_{93} & & & & & & a_{99} & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & a_{10,8} & a_{10,9} & a_{10,10} & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & a_{11,10} & a_{11,11} & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & a_{12,11} & a_{12,12} & & & & & & & & & & & & & & & & & & \\
 & & & & & & a_{13,7} & & & & & & a_{13,13} & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & a_{14,13} & a_{14,14} & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & a_{15,14} & a_{15,15} & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & a_{16,15} & a_{16,16} & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & a_{17,16} & a_{17,17} & a_{17,18} & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & a_{18,14} & & & & a_{18,18} & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & a_{19,18} & a_{19,19} & & & & & & & & & & & & & \\
 & & & & & & a_{20,7} & & & & & & & & & & & & & a_{20,20} & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & a_{21,20} & a_{21,21} & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & a_{22,21} & a_{22,22} & & & & & & & & & a_{22,28} & \\
 & & & & & & & & & & & & & & & & & & & & & a_{23,22} & a_{23,23} & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & a_{24,23} & a_{24,24} & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & a_{25,24} & a_{25,25} & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & & a_{26,25} & a_{26,26} & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & a_{27,23} & & & a_{27,27} & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & & & a_{28,27} & a_{28,28} & & & & \\
 & & & & & & a_{29,7} & & & & & & & & & & & & & & & & & & & & & a_{29,29} & & & & \\
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_{2A} \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17A} \\ x_{17} \\ x_{18} \\ x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23A} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}
$$

where

$$a_{1,1} = 1$$

$$a_{1,2} = 1$$

$$a_{2,1} = \frac{1}{S(1 + ST_1)}$$

$$a_{2,2} = -1$$

$$a_{3,2} = \frac{1}{(1 + ST_{TAR1})(1 + ST_{TAR2})}$$

$$a_{3,3} = -1$$

$$a_{4,3} = \frac{1}{1 + ST_6}$$

$$a_{4,4} = -1$$

$$a_{5,4} = \frac{K_D}{(1 + ST_{D1})(1 + ST_{D2})}$$

$$a_{5,5} = -1$$

$$a_{6,5} = 1$$

$$a_{6,6} = -1$$

$$a_{6,12} = 1$$

$$a_{6,17} = 1$$

$$a_{7,6} = \frac{K_{TVC}}{(1 + ST_{TVC})^2}$$

$$a_{7,7} = -1$$

$$a_{8,8} = (1 + ST_1)\ (1 + ST_2)$$

$$a_{9,3} = \frac{1}{1 + ST_2}$$

$$a_{9,9} = -1$$

$$a_{10,8} = -1$$

$$a_{10,9} = 1$$

$$a_{10,10} = -1$$

$$a_{11,10} = \frac{1}{1 + ST_3}$$

$$a_{11,11} = -1$$

$$a_{12,11} = \frac{K_{R2}}{(1 + ST_R)(1 + ST_{R22})}$$

$$a_{12,12} = -1$$

$$a_{13,6} = \frac{1}{\frac{S^2}{W_{88}^{\ 2}} + \frac{2\ \ 88}{W_{88}} S + 1} \underline{\underline{T}}_{88}$$

$$a_{13,13} = -1$$

$$a_{14,13} = \frac{1}{\frac{S^2}{W_{R6}^{\ 2}} + \frac{2\ \xi_{RG}}{W_{R6}} S + 1}$$

$$a_{14,14} = -1$$

$$a_{15,14} = \frac{1}{1 + ST_4}$$

$$a_{15,15} = -1$$

$$a_{16,15} = \frac{K_{R1}}{(1 + ST_{R11})(1 = ST_{R12})}$$

$$a_{16,16} = -1$$

$$a_{17,16} = -1$$

$$a_{17,17} = -1$$

$$a_{17,18} = 1$$

$$a_{17,25} = 1$$

$$a_{18,14} = \frac{K_{R1A}S}{(1 + ST_{R1A1})(1 + ST_{R1A2})}$$

$$a_{18,18} = -1$$

$$a_{19,18} = \frac{1}{(1 + ST_{R1A3})}$$

$$a_{19,19} = -1$$

$$a_{20,6} = \frac{S}{\frac{S^2}{W_{230}^2} + \frac{2\xi_{230}}{W_{230}} S + 1} \underline{\underline{T}}_{230}$$

$$a_{20,20} = -1$$

$$a_{21,20} = \frac{1}{\frac{S^2}{W_L^2} + 2\frac{\xi_L}{W_L} S + 1}$$

$$a_{21,21} = -1$$

$$a_{22,21} = 1$$

$$a_{22,22} = -1$$

$$a_{22,28} = -1$$

$$a_{23,28} = -1$$

$$a_{23,22} = K_A$$

$$a_{23,23} = -1$$

$$a_{24,23} = \frac{1}{1 + ST_7}$$

$$a_{24,24} = -1$$

$$a_{25,24} = \frac{1}{(1 + ST_{A1})\ (1 + ST_{A2})}$$

$$a_{25,25} = -1$$

$$a_{26,25} = \frac{1}{1 + ST_{A3}}$$

$$a_{26,26} = -1$$

$$a_{27,23} = \frac{\tau}{1 + S\tau}$$

$$a_{27,27} = -1$$

$$a_{28,27} = K_V$$

$$a_{28,28} = -1$$

$$a_{29,6} = \frac{1}{\frac{S^2}{W_{257}^2} + 2\frac{\zeta_{257}}{W_{257}} S + 1} \underline{\underline{T}}_{257}$$

$$a_{29,29} = -1$$

$$b_1 = 1$$

It should be noted that this example involves a matrix solution for a matrix of order nxn where n is dependent on elements $a_{13,6}$; $a_{26,6}$, $a_{29,6}$ which involve the transformation $\underline{\underline{T}}$, the vehicle dynamics from input x_6 to the individual station sensors, in this case at stations 88, 230 and 257 for example.

A further caution must be observed. The slack variables x_{2A}, x_{17A} and x_{23A} are introduced since the COEBRA program was designed to handle only linear and quadratic terms in s.

In the frequency domain the matrix for mat is somewhat more complicated but also as versatile. The format is fundamentally the same with the vehicle dynamics shown in a more explicit fashion. The usual procedure is to open the autopilot loop, typically at x_6 as shown in Figure D-4.

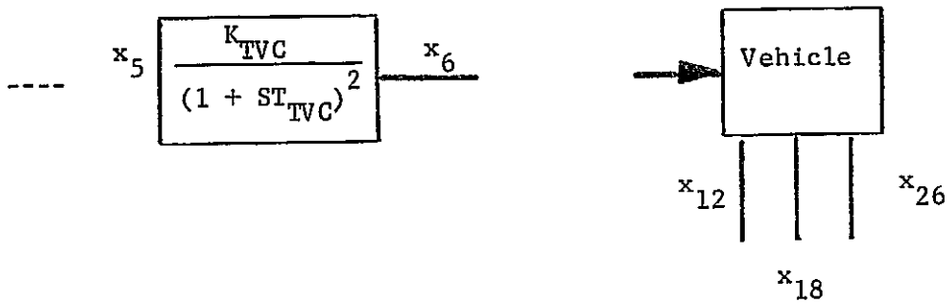


Figure D-4. Opening the Autopilot Loop

The individual vehicle parameters now become, one-at-a time, the system forcing functions. The equation

$$\underline{\underline{A}}\,\underline{X} = \underline{\underline{B}}\,\underline{U} \qquad \text{D-3}$$

now appears as in equation D-4.

$$\left[\begin{array}{c|c} \underline{\underline{T}}_{11} & \underline{\underline{T}}_{12} \quad \underline{\underline{0}} \\ \hline \underline{\underline{0}} & \underline{\underline{T}}_{22} \end{array}\right] \left[\begin{array}{c} \underline{X}_{11} \\ \hline \underline{X}_{12} \\ \hline \underline{X}_{2} \end{array}\right] = \left[\begin{array}{c} \underline{b}_1 \\ \hline \underline{0} \\ \hline \underline{0} \end{array}\right] \bullet u_i \qquad \text{D-4}$$

Where $\underline{\underline{T}}_{11}$ represents the coefficients of the vehicle dynamical equations, $\underline{\underline{T}}_{12}$ the vehicle control equation coefficients for those control effectors not used as a forcing function and $\underline{\underline{T}}_{22}$ the coefficients of the autopilot equations. $\underline{\underline{X}}_{11}$ the vehicle response variable resulting from the active forcing function, $\underline{x}_{12}$ the control effectors not active as a forcing function, $\underline{x}_2$ the autopilot variables as in equation D-2; $\underline{b}_1$ is the three element vector as coefficients of the active control effector under study. $\underline{\underline{T}}_{11}$, the vehicle submatrix will include 2 rigid body equations involving the lateral and rotational motions of the vehicle and n bending modes, thus $\underline{\underline{T}}_{11}$ is an (n+2) x (n+2) matrix; $\underline{\underline{T}}_{12}$ is m x m where m is the number of control effector loops that remain closed (in active control effectors); $\underline{\underline{T}}_{22}$ is, at least for the example of Figure D-1, 29 x 29 and is dependent not only on the actual autopilot configuration but also on the number of slack variables required.

In formatting for input to COEBRA the $\underline{B}$ vector is used to augment the $\underline{\underline{A}}$ matrix. If $\underline{\underline{A}}$ is n x n, the column vector $\underline{B}$ is added to $\underline{\underline{A}}$ as the n + 1st column as an input to COEBRA. The precise details are left to Volume II, the COEBRA Users Manual.

APPENDIX E

THE SIMPLEX ALGORITHM

The simplex algorithm is basically a simple procedure for solving a linear programming problem. The basis for this appendix is derived from references 49, and 50.

The general linear programming problem can be stated as follows: determine the set of n nonnegative variables, x_n, that satisfy m linear inequalities or equalities (the contraints).

$$\underline{\underline{A}}_{m \times n} \underline{X}_n \left\{ \leq = \geq \right\} \underline{B}_m, \; m \quad n \quad \ldots\ldots\ldots\ldots \quad \text{E-1}$$

where one and only one of the signs in the brace, $\leq$, = or $\geq$ holds for each individual constraint but can vary from one constraint to another within the set of m constraints defined by E-1. In addition, these n nonnegative variables are to extremize (maximize or minimize) a linear form (objective or cost function)

$$\underline{z} = \underline{c}^T \underline{x} \quad \ldots\ldots\ldots\ldots \quad \text{E-2.}$$

Any set $\underline{x}$ which satisfies E-1 is called a solution. Any nonnegative solution set $\underline{x} > \underline{o}$ is called feasible (admissable) and any such set that extremizes E-2 is called an optimum feasible (admissable) soulution.

Generally, it is easier to work with equations than inequalities so we convert the inequalities through the introduction of slack variables. For example, given $b_i < o$

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{1n}x_n \left\{ > = < \right\} b_i < o. \quad \ldots\ldots\ldots\ldots \quad \text{E-3}$$

multiplying through the inequality by -1 yields

$$-\sum_{j=1}^{n} a_{in}x_i > -b_i = b_i > o \quad \ldots\ldots\ldots\ldots \quad \text{E-4}$$

where multiplication by -1 has reversed the inequality. For example, $-7 < -2$ but $(-7)(-1) > (-2)(-1)$. It is thus simple to convert all constraints with each $b_i \geq o$.

Converting the inequality constraints into equalities we observe the convention in ordering the constraints as follows:

1. All constraints with $\leq$ signs
2. All constraints with $\geq$ signs
3. All equality constraints

Examining the type 1 constraints first we observe for the i-th constraint

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{. . . . . .} \quad \text{E-5}$$

Introducing a slack variable $x_{n+k} \geq o$ we have

$$x_{n+k} = b_i - \sum_{j=1}^{n} a_{ij} x_j \geq o \quad \text{. . . . . .} \quad \text{E-6}$$

Re-arranging E-6 we can write

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+k} = b_i \quad \text{. . . . . .} \quad \text{E-7}$$

For a type 2 constraint

$$\sum_{p=1}^{n} a_{h_p} x_p \geq b_h \quad \text{. . . . . .} \quad \text{E-8}$$

We write

$$x_{n+h} = \sum_{p=1}^{n} a_{h_p} x_p - b_h \quad \text{. . . . . .} \quad \text{E-9}$$

To obtain

$$\sum_{p=1}^{n} a_{h_p} x_p - x_{n+h} = b_h \quad \text{. . . . . .} \quad \text{E-10}$$

Thus, the original constraints are rewritten, with the introduction of slack variables, as

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+k} = b_i \quad \text{E-11a}$$

$$\sum_{p=1}^{n} a_{h_p} x_p - x_{n+h} = b_h \quad \text{E-11b}$$

$$\sum_{k=1}^{n} a_{pk} x_k = b_q \quad \text{E-11c}$$

It is thus possible to recast the linear programming problem in the form

$$\underline{\underline{A}}_{mxm} \underline{X}_n = \underline{B}_m \quad \text{E-12a}$$

$$x_i \geq o, \ (i = 1, \ldots, n) \quad \text{E-12b}$$

$$\max \underline{z} = \underline{c}^T \underline{x} \quad \text{E-12c}$$

We assume in what follows that we have a basic feasible solution at hand and proceed to develop the algorithm to generate the "optimum" basic solution and then treat the problem of determining the first basic solution.

Let

$$\underline{x}_o = \text{col} \left[x_{10} \ x_{20} \ \ldots \ x_{mo} \ o \ o \ \ldots \ \underline{o} \right] \quad \text{E-13}$$

be a basic feasible solution to the linear programming problem and the set of linearly independent basis vectors be $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_m$, whence

$$\sum_{j=1}^{m} x_{io} \underline{P}_i \underline{P}_i = \underline{P}_o \quad \text{E-14}$$

and

$$\sum_{j=1}^{m} x_{io} c_i = z_o \quad \text{E-15}$$

Where all $x_{io} > o$, the c_i are the coefficients from E-12c and z_o the corresponding value of the objective or cost function. Since the $\underline{p}_i$ are linearly independent any of the vectors $\underline{p}_1, \underline{p}_2 \ldots, \underline{p}_n$ can be written in terms of $\underline{p}_1, \underline{p}_2 \ldots, \underline{p}_m$. If we let $\underline{p}_j$ be defined by

$$\underline{p}_j = \sum_{i=1}^{m} x_{ij} \underline{p}_i, \; j = 1, \ldots, n \qquad \text{E-16}$$

and define

$$z_j = \sum_{i=1}^{m} x_{ij} c_i, \; j = 1, \ldots, n \qquad \text{E-17}$$

where the c_i are the coefficients corresponding to the $\underline{p}_i$, we have $z_j - c_j > o$, evidencing a feasible solution.

Denoting the set P_i as

$$\underline{\underline{\beta}} = [P_1 : P_2 : \ldots : \underline{p}_m] \qquad \text{E-18}$$

we observe that $\underline{P}_0$

$$\underline{\underline{\beta}}\, \underline{x}_o = \underline{p}_o \qquad \text{E-19}$$

Inverting

$$\underline{x}_o = \underline{\underline{\beta}}^{-1} \underline{p}_o \qquad \text{E-20}$$

and

$$\underline{x}_j = \underline{\underline{\beta}}^{-1} \underline{p}_j \qquad \text{E-21}$$

where

$$\underline{x}_o = \text{col}\,[x_{10}, x_{20}, \ldots, x_{mo}]\,, \quad x_{io} > o \quad \ldots\ldots\ldots \qquad \text{E-22}$$

and

$$\underline{x}_j = \text{col}\,[x_{1j}, x_{2j}, \ldots, x_{mj}] \ldots\ldots\ldots\ldots\ldots \qquad \text{E-23}$$

each of which is a column vector.

We group the vectors $\underline{P}_i$ in the form

$$[\underline{P}_o : \underline{P}_1 : \underline{P}_2 \ldots : \underline{P}_m : \underline{P}_{m+1} \ldots \underline{P}_n]$$

$$= [\underline{P}_o : \underline{\beta} : \underline{P}_{m+1} : \ldots : \underline{P}_n] \ldots\ldots\ldots\ldots\ldots \qquad \text{E-24}$$

Pre-multiplying by $\underline{\beta}^{-1}$ we obtain

$$\underline{\beta}^{-1}\,[\underline{P}_o : \underline{\beta} : \underline{P}_{m+1} : \ldots : \underline{P}_n]$$

$$= [\underline{x}_o : I_{mxm} : \underline{x}_{m+1} \ldots \underline{x}_n] \ldots\ldots\ldots\ldots\ldots \qquad \text{E-25}$$

Knowing the c_j we form $z_j - c_j$ and examine for all j, to find $z_j - c_j > o$.

If, for all j, $z_j - c_j > o$, we multiply E-16 by some constant θ and subtract from E-14; similarly multiply E-17 and subtracting from E-15 we have

$$\sum_{i=1}^{m} (x_{io} - \theta x_{ij})\,\underline{P}_i + \theta \underline{P}_j = \underline{P}_o \ldots\ldots\ldots\ldots\ldots \qquad \text{E-26}$$

and

$$\sum_{i=1}^{m} (x_{io} - \theta x_{ij})\,c_i + \theta c_j = z_o - \theta(z_j - c_j) \ldots\ldots \qquad \text{E-27}$$

where θc_j has been added to both sides of E-27. If, in E-26, all the $(x_{io} - \theta x_{ij}) > 0$ we have obtained a new feasible solution whose objective function value is, from E-27, $z = z_o - \theta(z_j - c_j) < z_o$, where

$$\theta_o = \min_i \frac{x_{io}}{x_{ij}} > 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \text{E-28.}$$

If, in initially forming $z_j - c_j$, we find for some j, $z_j - c_j < o$ we already have the minimum feasible solution.

If the set of n vectors $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_n$ contains m unit vectors that can be regrouped to form an m x m unit vector we write E-18 as

$$\underline{\underline{\beta}} = [\underline{P}_1\ \underline{P}_2 \ldots \underline{P}_m] = \underline{\underline{I}}_m \quad \text{E-29}$$

Under this condition equation E-20 becomes

$$\underline{x}_o = \underline{P}_o \quad \text{E-30}$$

$$\underline{x}_j = \underline{P}_j \quad \text{E-31}$$

and we have an initial solution.

To start the simplex procedure, in this case, we have $x_{io} = b_i$, and $x_{ij} = a_{ij}$. To obtain the z_j for j = 0, 1, . . . n, we form the inner product of the j-th vector with the given $\underline{c}$, or

$$z_o = \sum_{i=1}^{m} c_i x_{io} \quad \text{E-32}$$

and

$$z_j = \sum_{i=1}^{m} c_i x_{ij}, \ j = 1, 2, \ldots n \quad \text{E-33}$$

It is convenient to arrange the elements of the problem in a tableau as shown in Figure E-1.

Referring to Figure E-1 we enter the elements z_o and $z_j - c_j$ into the m + 1st row of their respective columns. If all the $z_j - c_j < o$, j = 1, 2 . . . n, the solution $\underline{x}_o = \text{col}\,[x_{10}, x_{20}, \ldots x_{mo}] = \text{col}\,[b_1, b_2, \ldots, b_m]$ is a minimum feasible solution and the corresponding cost function is z_o. If not, we compute a new feasible solution whose basis contains m - 1 vectors of the original basis $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_m$. An efficient procedure is to select a vector $\underline{P}_j$ with its $z_j - c_j > o$ which yields the largest decrease in the value of z. This vector, $\underline{P}_j$, should be the

i	Basis	$\underline{c}$	$\underline{P}_o$	c_1	c_2	.	c_1	.	c_m	c_{m+1}	.	c_j	.	c_k	.	c_n
				$\underline{P}_1$	$\underline{P}_2$	.	$\underline{P}_1$	.	$\underline{P}_m$	$\underline{P}_{m+1}$	.	$\underline{P}_j$	.	$\underline{P}_k$	.	$\underline{P}_n$
1	$\underline{P}_1$	c_1	x_{10}	1	o	.	o	.	o	$x_{1,\ m+1}$	.	x_{1j}	.	x_{1k}	.	x_{1n}
2	$\underline{P}_2$	c_2	x_{12}	o	1	.	o	.	o	$x_{2,\ m+1}$	.	x_{2j}	.	x_{2k}	.	x_{2n}
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
1	$\underline{P}_1$	c_1	x_{11}	o	o	.	1	.	o	$x_{1,\ m+1}$	.	x_{1j}	.	x_{1k}	.	x_{1n}
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
m	$\underline{P}_m$	c_m	x_{1m}	o	o	.	o	.	1	$x_{m,\ m+1}$	.	x_{mj}	.	x_{mj}	.	x_{mn}
m+1			z_o	o	o	o	o	o	o	$z_{m+1} - c_{m+1}$	.	$z_j - c_j$	.	$z_k - c_k$	.	$z_n - c_n$

FIGURE E-1 SIMPLEX TABLEAU FOR INITIAL STEP IN COMPUTATIONAL PROCEDURE

one corresponding to the $\max_j \theta_o (z_j - c_j)$ where θ_o is defined by equation E-28. An alternate choice, often used, is to select that $\underline{P}_j$ corresponding to max $(z_j - c_j)$. This is the approach we take here. Thus, we let

$$\max_j (z_j - c_j) = z_k - c_k > o \quad \text{E-34}$$

whence the vector $\underline{P}_k$ is to be introduced into the basis set. Computing

$$\theta_o = \min_i \frac{x_{io}}{x_{ik}}, \quad x_{ik} > o \quad \text{E-35}$$

where x_{ik} belongs to the x_i in the basic solution. For example

$$\theta_o = \frac{x_{1o}}{x_{1k}}$$

and the vector $\underline{P}_\ell$ of Figure E-1 is the one to be eliminated from the basis set. The new feasible solution has as its basis $\underline{P}_1, \underline{P}_2, \ldots \underline{P}_{1-1}, \underline{P}_{1+1}, \ldots, \underline{P}_m, \underline{P}_k$.

Since our initial basis is $[\underline{P}_1, \underline{P}_2 \ldots \underline{P}_m] = \underline{I}_m$

we can express all of the vectors $\underline{P}_j$ in terms of this set, or

$$\underline{P}_o = x_{1o}\underline{P}_1 + \ldots + x_{\ell o}\,\underline{P}_1 + \ldots + x_{mo}\,\underline{P}_m \quad \text{E-36a}$$

$$\underline{P}_k = x_{1k}\,\underline{P}_1 + \ldots + x_{\ell k}\,\underline{P}_1 + \ldots + x_{mk}\underline{P}_m \quad \text{E-36b}$$

$$\underline{P}_j = x_{ij}\,\underline{P}_1 + \ldots + x_{\ell j}\,\underline{P}_1 + \ldots + x_{mk}\underline{P}_m \quad \text{E-36c}$$

Solving E-36b for $\underline{P}_1$ we have

$$\underline{P}_\ell = \frac{1}{x_{\ell k}} (\underline{P}_k - x_{1k}P_1 - \ldots - x_{mk}\,\underline{P}_m) \quad \text{E-37}$$

Substitution of this value in E-36a yields

$$\underline{P}_o = x_{1o}\,\underline{P}_1 + \ldots + x_{\ell o}\left[\frac{1}{x_{\ell k}} (\underline{P}_k - x_{1k}\,\underline{P}_1 - \ldots - x_{mk}\,\underline{P})\right] + x_{1+\ell,o}\,\underline{P}_{\ell+1} + \ldots + x_{mo}\,\underline{P}_m$$

or

$$\underline{P}_o = \left(x_{10} - \frac{x_{\ell o}}{x_{\ell k}} x_{1k}\right) \underline{P}_1 + \ldots + \frac{x_{\ell o}}{x_{\ell k}} + \ldots$$

$$+ \left(x_{mo} - \frac{x_{\ell o}}{x_{\ell k}} x_{mk}\right) \underline{P}_m \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \text{E-38}$$

which is equivalent to E-26 with $j = k$ and $\theta = \frac{x_{\ell o}}{x_{\ell k}} = \theta_o$.

The new feasible solution

$$\underline{x}_o' = \text{col}\left[x_{10}', x_{20}', \ldots, x_{ko}', \ldots x_{mo}'\right]$$

$$x_i' \geq o$$

is given by

$$\underline{P}_o = \sum_{\substack{i = 1 \\ i \neq 1}}^{m} x_{io}' \underline{P}_i \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \text{E-39}$$

where

$$\left.\begin{aligned} x_{io}' &= x_{io} - \frac{x_{\ell o}}{x_{\ell k}} x_{ik}, \quad i = 1, 2 \ldots, \ell-1, \ell+1, \ldots m \\ x_{ko}' &= \frac{x_{\ell o}}{x_{\ell k}} \end{aligned}\right\} \quad \text{E-40}$$

By substituting E-37 into E-36c we obtain each $\underline{P}_j$ not in the new basis set as

$$\underline{P}_j = \sum_{\substack{i = 1 \\ i \neq 1}}^{m} x_{ij}' \underline{P}_i \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \text{E-41}$$

where

$$x'_{ij} = x_{ij} - \frac{x_{\ell j}}{x_{\ell k}} x_{ik}, \; i \neq 1 \quad \text{E-42}$$

$$x'_{kj} = \frac{x_{\ell j}}{x_{\ell k}} \quad \text{E-43}$$

and $x_{\ell k}$ is denoted the pivot element.

Now

$$z'_j - c_j = \sum_{i=1}^{m} x'_{ij} \; c_i - c_j \quad \text{E-44}$$

which, by substitution of E-40 for x' in

$$z'_o = \sum_{\substack{1 = 1 \\ i \neq \ell}}^{m} c_i \, x_{io},$$

gives

$$z'_o = z_o - \frac{x_{\ell o}}{x_{\ell k}} (z_k - c_k) \quad \text{E-45}$$

To obtain the new solution, $\underline{x}'_o$, the new vectors $\underline{x}'_j$ and the corresponding $z'_j - c_j$ we transform every element in the tableau of Figure E-1 by equation E-42 for x'_{ij} and $x^i_{\ell j} = \frac{x_{\ell j}}{xx_{\ell k}}$. The new cost function is

$$z'_o = x'_{m+1,0} \quad \text{E-46}$$

and

$$z'_j - c_j = x'_{m+1,\,j} \quad \text{E-47}$$

Once an initial computational tableau has been constructed, the simplex procedure calls for the successive application (i.e., an iteration) of:

1. The testing of the $z_j - c_j$ elements to determine whether a minimum solution has been found, i.e., whether $z_j - c_j < 0$ for all j.
2. The selection of the vector to be introduced into the basis if some $z_j - c_j > 0$, i.e., selection of the vector with maximum $z_j - c_j$.
3. The selection of the vector to be eliminated from the basis to ensure feasibility of the new solution, i.e., the vector with min (x_{i0}/x_{ik}) for those $x_{ik} > 0$, where k corresponds to the vector selected in Step 2. If all $x_{ik} < 0$, then the solution is unbounded.
4. The transformation of the tableau by the complete elimination procedure to obtain the new solution and associated elements.

Each such iteration produces a new feasible solution and eventually yields the minimal or optimum solution or reveals an unbounded one.

A simple example will serve to clarify the foregoing. We wish to minimize the cost function

$$z = \underline{c}^T \underline{x} = \begin{bmatrix} 0 & 1 & - & 3 & 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

subject to the constraints.

i	Basis	c	$\underline{P}_o$	c_1	c_2		c_1	.	c_m	$c_m + 1$	.	c_j	.	c_k	.	c_n
				$\underline{P}_1$	$\underline{P}_2$	.	$\underline{P}_1$	.	$\underline{P}_m$	$\underline{P}_m + 1$	.	$\underline{P}_j$	.	$\underline{P}_k$	.	$\underline{P}_n$
1 $\underline{P}$	$\underline{P}_1$	c_1	x'_{10}	1	0		$x'_{1\ell}$		o	x'_{ℓ}, m + 1		x'_{1j}		o		x'_{1n}
2	$\underline{P}_2$	c_2	x'_{20}	0	1		$x'_{2\ell}$		o	x'_{2}, m + 1		x'_{2j}		o		x'_{2n}
ℓ	$\underline{P}_k$	c_k	x'_{ko}				$x'_{\ell\ell}$		o	x'_{ℓ}, m + 1		$x'_{\ell j}$		1		$x'_{\ell n}$
m	P_m	c_m	x'_{mo}				$x'_{m\ell}$		1	x'_{m}, m + 1		x'_{mj}		o		x'_{mn}
m +1			z'_o	o	o		$z'_\ell - c_\ell$	.	o	$z'_{m+1} - c_{m+1}$	.	$z'_j - c_j$				$z'_n - c_n$

FIGURE E-2 TABLEAU FOR SECOND COMPUTATIONAL STEP

$$\underline{\underline{A}}\ \underline{X} = \underline{B}$$

or

$$\begin{bmatrix} 1 & 3 & -1 & 0 & 2 & 0 \\ 0 & -2 & 4 & 1 & 0 & 0 \\ 0 & -4 & 3 & 0 & 8 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 7 \\ 12 \\ 10 \end{bmatrix}$$

We select as our initial basis set the vectors $\underline{P}_1$, $\underline{P}_4$ and $\underline{P}_6$ since c_1, c_4 and c_6 are all zero. The initial tableau is then that of Figure E-3. The initial value of the cost function, z_o, is zero. Computing the z_i we find

$$z_o = \begin{vmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{vmatrix} \begin{bmatrix} 7 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \\ 10 \end{bmatrix} = 0$$

$$z_1 = \begin{vmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{vmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

i	Basis	$\underline{c}$	$\underline{P}_0$	c_1	c_2	c_3	c_4	c_5	c_6
				$\underline{P}_1$	$\underline{P}_2$	$\underline{P}_3$	$\underline{P}_4$	$\underline{P}_5$	$\underline{P}_6$
1	$\underline{P}_1$	c_1	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
2	$\underline{P}_2$	c_2	x_{20}	x_{21}	x_{22}	.	.	.	.
3	$\underline{P}_3$	c_3	x_{30}	x_{31}	x_{32}	.	.	.	.
4	$\underline{P}_4$	c_4	x_{40}	x_{41}	x_{42}	.	.	.	.
5	$\underline{P}_5$	c_5	x_{50}	x_{51}	x_{52}	.	.	.	.
6	$\underline{P}_6$	c_6	x_{60}	x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}
7			z_o	z_1-c_1	z_2-c_2	z_3-c_3	z_4-c_4	z_5-x_5	z_6-x_6

a) The Basic Tableau

i	Basis	$\underline{c}$	$\underline{P}_0$	c_1 $_0$	c_2 $_1$	c_3 $_{-3}$	c_4 $_0$	c_5 $_2$	c_6 $_0$
				$\underline{P}_1$	$\underline{P}_2$	$\underline{P}_3$	$\underline{P}_4$	$\underline{P}_5$	$\underline{P}_6$
1	$\underline{P}_1$	0	7	1	3	-1	0	2	0
2	$\underline{P}_4$	0	12	0	-2	4	1	0	0
3	$\underline{P}_6$	0	10	0	-4	3	0	8	1
4			z_o	z_1-c_1	z_2-c_2	z_3-c_3	z_4-c_4	z_5-c_5	z_6-c_6

b) STEP 1A

i			$\underline{P}_0$	c_1	c_2	c_3	c_4	c_5	c_6
1									
2									
3									
4			0	0	-1	3	0	-2	0

c) STEP 1b

FIGURE E-3 THE INITIAL TABLEAU OF THE EXAMPLE

$$z_1 - c_1 = 0$$

$$z_2 = \begin{bmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 0 \\ -2 \\ 0 \\ -4 \end{bmatrix} = 0$$

$$z_2 - c_2 = -1$$

$$z_3 = \begin{bmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 0 \\ 4 \\ 0 \\ 5 \end{bmatrix} = 0$$

$$z_3 - c_3 = 3$$

$$z_4 = \begin{bmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = 0$$

$$z_4 - c_4 = 0$$

$$z_5 = \begin{bmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 8 \end{bmatrix} = 0$$

$$z_5 - c_5 = -2$$

$$z_6 = \begin{bmatrix} 0 & 1 & -3 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

$$z_6 - c_6 = 0$$

Examining the $z_i - c_i$ for a maximum we find $z_3 - c_3 = 3 > 0$, so we select $\underline{P}_3$ to enter the basis.

Now θ_o is the minimum of $\frac{x_{i0}}{x_{i3}}$ for $x_{i3} > 0$. Examining each

$$\frac{x_{20}}{x_{23}} = \frac{12}{4} = 3 \quad 0$$

$$\frac{x_{30}}{x_{33}} = \frac{10}{3} = 3\ 1/3 \quad 0$$

noting that $x_{13} = -1 < 0$, we find $\theta_o = 3$ and $\underline{P}_4$, corresponding to the pivot element, x_{23}, is to be eliminated. Transforming the tableau

of Figure E-3 we obtain that of Figure E-4, whence $\underline{x}_0' = \text{col}\,[10, 3, 1]$ and $30' = -9$ as seen from:

k	Basis	$\underline{C}$	$\underline{P}_0$	$\underline{P}_1$	$\underline{P}_2$	$\underline{P}_3$	$\underline{P}_4$	$\underline{P}_5$	$\underline{P}_6$
1	$\underline{P}_1$	0	10	1	5/2	0	1/4	2	0
2	$\underline{P}_3$	-3	3	0	-1/2	1	1/4	0	0
3	$\underline{P}_6$	0	1	0	-5/2	0	-3/4	8	1
4			- 9	0	1/2	0	-3/4	-2	0

Figure E-4 The T Array for Step .

$$x_{10}' = x_{10} - \frac{x_{20}}{x_{23}} x_{13} = 7 - \frac{12}{4}(-1) = 10$$

$$x_{20}' = \frac{x_{20}}{x_{23}} = \frac{12}{3} = 3$$

$$x_{30}' = \underline{x_{30}} - \frac{x_{20}}{x_{23}} \cdot x_{33} = 10 - \frac{12}{4} .3 = 1$$

and

$$z_0' = z_0 - \frac{x_{\ell o}}{x_{\ell k}} (z_k - c_k) = 0 - \frac{12}{4}(3) = -9$$

In forming the set $\underline{P}_i$ of the second step tableau of Figure E-4 we observe

$$x'_{ij} = x_{ij} - \frac{x_{2j}}{x_{23}} \quad xi3, \; i \neq 2$$

Thus

$$x'_{11} = 1 - 0 = 1$$

$$x'_{12} = 3 - \frac{(-2)}{4}(-1) = 3 - 1/2 = 2$$

$$x'_{13} = -1 - \frac{4}{4}(-1) = 0$$

$$x'_{14} = 0 - 1/4\ (-1) = 1/4$$

$$x'_{15} = 2 - 0 = 2$$

$$x'_{16} = 0 - 0 = 0$$

etc.

The $z'_i - c_i$ are formed as before and the max $(z'_j - c_j) = z'_2 - c_2 = 1/2 > 0$ is observed. Thus $\theta'_o = \min\left(\frac{x_{io}}{x_{i2}}\right) = \min\left(\frac{10}{5/2}, \frac{3}{-1/2}, \frac{1}{-5/2}\right) \quad 4.$

since we require $x_{i2} > 0$.

$\underline{P}_2$ is now the vector to be introduced into the basis set and $\underline{P}_1$ the one to be eliminated to obtain the tableau of Figure E-5.

i	Basis	$\underline{c}$	$\underline{P}_0$	0	1	-3	0	2	0
				$\underline{P}_1$	$\underline{P}_2$	$\underline{P}_3$	$\underline{P}_4$	$\underline{P}_5$	$\underline{P}_6$
1	$\underline{P}_2$	1	4	2/5	1	0	1/10	4/5	0
2	$\underline{P}_3$	-3	5	1/5	0	1	3/10	2/5	0
3	$\underline{P}_6$	0	11	1	0	0	-1/2	10	1
4			-11	-1/5	0	0	-4/5	-12/5	0

FIGURE E-5 STEP 3 IN THE COMPUTATIONAL SCHEME

Since the $\max_j (z''_j - c_j) = 0$ this is an optimum solution. If the given linear programming problem does not contain a unit matrix, as was assumed in the foregoing the method of artificial basis detailed in references [50 and 51] is referred to and will not be detailed here.

It is the simple^x algorithm that COEBRA utilizes, as an efficient method of solving the autopilot problem.