

~~11180~~
N70-11180
NASA CR-102303



DEPARTMENT
OF
ELECTRICAL ENGINEERING

CASE FILE
COPY

A STUDY OF THE EFFECTS OF VARYING THE SAMPLING FOR A

SIMPLIFIED VERSION OF THE SATURN V/S-1C

FINAL REPORT

Contract No. NAS8-21377

Covering Period May 15, 1968 - August 31, 1969

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MISSISSIPPI STATE UNIVERSITY
STATE COLLEGE, MISSISSIPPI

MISSISSIPPI STATE UNIVERSITY
State College, Mississippi

Final Report

Contract No. NAS8-21377

Covering Period May 15, 1968 - August 31, 1969

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

A STUDY OF THE EFFECTS OF VARYING THE SAMPLING
FOR A SIMPLIFIED
VERSION OF THE SATURN V/S-1C

By

Jerrel R. Mitchell

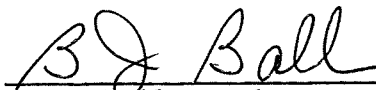
Submitted on behalf of

Willie L. McDaniel, Jr.

Professor of Electrical Engineering



Willie L. McDaniel, Jr.
Principal Investigator



B. J. Ball, Head
Department of Electrical Engineering

TABLE OF CONTENTS

	Page
ABSTRACT	iii
LIST OF FIGURES	iv
DEFINITIONS OF SYMBOLS	ix
INTRODUCTION	1
OPEN LOOP STUDIES	3
CLOSED LOOP STUDIES	11
COMPENSATION	19
CONCLUSION	23
APPENDICES	95
Appendix A	96
Appendix B	109
Appendix C	111
Appendix D	116
Appendix E	133
Appendix F	137
REFERENCES	144

ABSTRACT

The bending frequencies of a large space vehicle, such as the Saturn V/S-1C, can be detrimental to the stability of the system. A scheme for filtering the bending frequencies from the control information is to keep the same relative stable position between the bending mode z-plane poles and corresponding compensation zeros.

The effects of varying the sampling period on a reduced order Saturn V/S-1C is presented for several flight times. Included are open and closed loop studies when sampling period is varied and a typical compensation of the system. Conclusions regarding these are also presented.

LIST OF FIGURES

FIGURE	PAGE
1. Simplified Block Diagram of Saturn V/S-1C	26
2. Reduced Simplified Block Diagram of Saturn V/S-1C	27
3. A Block Diagram for Redefining the Blocks of Figure 4	28
4. Block Diagram Which Results from Figure 3	29
5. Simplified Block Diagram of Saturn V/S-1C Stage After Several Block Diagram Reductions	30
6. Block Diagram of the Open Loop Transfer Function, G_ϕ	31
7. S-plane Loci of the First and Second Bending Modes and Second Slosh Mode as a Function of Flight Time	32
8. S-plane Loci of the Drift and Rigid Body Poles as a Function of Flight Time	33
9. Plot of a Pole of the Second Bending Mode as a Function of Sampling Period for TF = 40 Seconds	34
10. Plot of a Pole of the First Bending Mode as a Function of Sampling Period for TF = 40 Seconds	35
11. Plot of the First Slosh Mode Pole as a Function Sampling Period for TF = 40 Seconds	36
12. Loci of Drift Poles as a Function of Sampling Period for TF = 40 Seconds	37
13. Movement of a Rigid Body Pole as a Function Sampling Period	38
14. Indication of the Movement of a Second Bending Mode Pole as a Function of Sampling Period for TF = 80.11 Seconds	39
15. Plot of a First Bending Mode Pole as the Sampling Period if Varied for TF = 80.11 Seconds	40
16. Plot of the First Slosh Mode Pole as a Function Sampling Period for TF = 80.11 Seconds	41
17. Locus of a Drift Pole as Sampling Period is Varied for TF = 80.11 Seconds	42

LIST OF FIGURES (Continued)

FIGURE	PAGE
18. Plot of a Drift Pole as a Function Sampling Period for TF = 80.11 Seconds	43
19. Rigid Body Locus as Sampling Period is Varied for TF = 80.11 Seconds	44
20. Movement of a Second Bending Mode Pole as the Sampling is Varied for TF = 120 Seconds	45
21. Plot of a First Bending Mode Pole as a Function of Sampling Period for TF = 120 Seconds	46
22. Locus of a First Slosh Mode Pole as a Function Sampling Period for TF = 120 Seconds	47
23. Movement of a Drift Pole as a Function Sampling Period for TF = 120 Seconds	48
24. Movement of a Drift Pole as a Function Sampling Period for TF = 120 Seconds	49
25. Plot of a Rigid Body Pole as a Function Sampling Period for TF = 120 Seconds	50
26. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 20 Seconds and for an Open Loop Gain of 1.0	51
27. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 20 Seconds and for an Open Loop gain of 1.0	52
28. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 40 Seconds and for an Open Loop Gain of 1.0	53
29. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 40 Seconds and for an Open Loop Gain of 1.0	54
30. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 60 Seconds and for an Open Loop Gain of 1.0	55
31. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 60 Seconds and for an Open Loop Gain of 1.0	56

LIST OF FIGURES (Continued)

FIGURE	PAGE
32. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 80.11 Seconds and for an Open Loop Gain of 1.0	57
33. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 80.11 Seconds and for an Open Loop Gain of 1.0	58
34. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 100 Seconds and for an Open Loop Gain of 1.0	59
35. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 100 Seconds and for an Open Loop Gain of 1.0	60
36. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 120 Seconds and for an Open Loop Gain of 1.0	61
37. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 120 Seconds and for an Open Loop Gain of 1.0	62
38. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling period for TF = 140 Seconds and for an Open Loop Gain of 1.0	63
39. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 140 Seconds and for an Open Loop Gain of 1.0	64
40. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 20 Seconds and Open Loop Gain = -1.0	65
41. Plot of Drift and Rigid Body Poles as a Function of Sampling Period when TF = 20 Seconds and Open Loop Gain = -1.0	66
42. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 40 Seconds and Open Loop Gain = -1.0	67

LIST OF FIGURES (Continued)

FIGURE	PAGE
43. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 40 Seconds and Open Loop Gain = -1.0	68
44. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 60 Seconds and Open Loop Gain = -1.0	69
45. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 60 Seconds and Open Loop Gain = -1.0	70
46. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 80.1 Seconds and Open Loop Gain = -1.0	71
47. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 80.1 Seconds and Open Loop Gain = -1.0	72
48. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 100 Seconds and Open Loop Gain = -1.0	73
49. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 100 Seconds and Open Loop Gain = -1.0	74
50. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling period When TF = 120 Seconds and Open Loop Gain = -1.0	75
51. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 120 Seconds and Open Loop Gain = -1.0	76
52. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When TF = 140 Seconds and Open Loop Gain = -1.0	77
53. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 140 Seconds and Open Loop Gain = -1.0	78
54. Nyquist Plot of the Sampled-Data System for T = 0.79 Seconds, TF = 40 Seconds and Open Loop Gain = 1.0	79
55. Phase Margin and Gain Margin vs Sampling Period for TF = 20 Seconds and Open Loop Gain = 1.0	80

LIST OF FIGURES (Continued)

FIGURE	PAGE
56. Phase Margin and Gain Margin vs Sampling Period for TF = 40 Seconds and Open Loop Gain = 1.0	81
57. Phase Margin and Gain Margin vs Sampling Period for TF = 60 Seconds and Open Loop Gain = 1.0	82
58. Phase Margin and Gain Margin vs Sampling Period for TF = 80.11 Seconds and Open Loop Gain = 1.0	83
59. Phase Margin and Gain Margin vs Sampling Period for TF = 100 Seconds and Open Loop Gain = 1.0	84
60. Phase Margin and Gain Margin vs Sampling Period for TF = 120 Seconds and Open Loop Gain = 1.0	85
61. Regular Sampled-Data Root Loci of the Poles of Prime Importance of the Sat V/S-1C When T = 0.37 Seconds and TF = 40 Seconds	86
62. Nyquist Plot of the Sampled-Data Open Loop Transfer Function of the Sat V/S-1C for TF = 40 Seconds, T = 0.37 Seconds, and Open Loop Gain = 1.0	87
63. Uncompensated Bode Plot of the Pulse Open Loop Transfer Function of the Sat V/S-1C for TF = 40 Seconds, T = 0.37 Seconds, and Open Loop Gain = 1.0	88
64. Frequency Response Plots of the Chosen Compensator of Sat V/S-1C	89
65. Nyquist Plot of the Compensated System for TF = 40 Seconds, T = 0.37 Seconds, and Open Loop Gain = 1.0	90
66. Gain and Phase Margins vs Sampling Period for TF = 40 Seconds and Open Loop Gain = 1.0	91
67. Gain and Phase Margins vs Sampling Period for TF = 80.11 Seconds and Open Loop Gain = 1.0	92
68. Gain and Phase Margins vs Sampling Period for TF = 120 Seconds and Open Loop Gain = 1.0	93
69. A pictorial Representation of the Two Surfaces F_1 and F_2 , With Indicated Gradients and Cross Products	94

DEFINITIONS OF SYMBOLS

TF	Time of flight (seconds)
C1 - (c_1)	Aerodynamic restoring coefficient ($1/\text{Sec}^2$)
C2 - (c_2)	Control engine restoring coefficient ($1/\text{Sec}^2$)
K4 - (k_4)	Acceleration normal to vehicle per degree β (Meters/Deg- Sec^2)
K7 - (k_7)	Acceleration normal to vehicle per degree α (Meters/Deg- Sec^2)
K8 - (k_8)	Acceleration at control accelerometer due to rigid body angular accel. ($\frac{\text{M}/\text{Sec}^2}{\text{Deg}/\text{Sec}^2}$)
RP - (R')	Thrust of control engines (Kg)
SIG - (Σ_e)	First mass moment of control engines about gimbals point (Kg- Sec^2)
THE - (θ_e)	Moment of inertia of control engines about the gimbals point (Kg-M- Sec^2)
MOI - (I)	Moment of inertia (Kg-M- Sec^2)
GBAR - (\bar{g})	Vehicle longitudinal acceleration (M/Sec^2)
M - (m)	Total mass of vehicle (Kg- $\text{Sec}^2/\text{Meter}$)
FBN - (f_{b_i})	Natural elastic body bending frequency of i^{th} mode (cps)
ZM - (m_i)	Generalized mass of i^{th} bending mode (Kg- $\text{Sec}^2/\text{Meter}$)
ZB - (ζ_{b_i})	Damping ratio of i^{th} bending mode (Dimensionless)
YB - (Y_{β_i})	Deflection at the gimbals station due to i^{th} bending mode (Meter/Meter)
YPB - ($Y'\beta_i$)	Slope at the gimbals station due to the i^{th} bending mode (1/Meter)

DEFINITIONS OF SYMBOLS (Continued)

- YPP - $(Y'\phi_i)$ Slope at the stabilized platform station due to i^{th}
bending mode (1/Meter)
- YPRG - $(Y'\dot{\phi}_i)$ Slope at the attitude rate sensor station due to i^{th}
bending mode (1/Meter)
- FS - (f_{s_i}) Slosh frequency of i^{th} mode (cps)
- ZS - (ζ_{s_i}) Slosh damping ratio of i^{th} mode (Dimensionless)
- MS - (m_{s_i}) Equivalent slosh mass of i^{th} mode (Kg-Sec²/Meter)
- XBS - (\bar{X}_{s_i}) Distance from center of gravity of vehicle to center of
gravity of equivalent slosh mass - aft is positive (Meters)
- WSS - Servo-system

INTRODUCTION

In recent years the design and analysis techniques applicable to the complex systems resulting from space technology have undergone many changes. However, many of the classical methods have proved basically to be sound and useful, even though the systems have increased in complexity. The digital computer has greatly aided the pragmatic engineer in his quest for upgrading his analysis tools.

The size of the systems, especially the Saturn V, has led the analyst to consider techniques for the improvement of the system response characteristics that were not thought feasible in the past. One of the areas of concern in the Saturn V/S-1C has been that of suppressing the bending frequencies. This report gives results pertinent to the study of bending frequency filtering through adaptive sampling.

The system under study is a simplified version of the Saturn V/S-1C. Two bending modes and one slosh mode have been included in the system dynamics leading to a thirteenth order system. The open loop transfer function of the system has been determined. A partial fraction expansion led to a corresponding z-transform equivalent. The movement of the poles of the open loop transfer function has been determined as a function of vehicle flight time and sampling period. A study of the effects of sampling period variation on the poles of the closed loop system has been completed for sampling periods of $0 < T < 1.0$. In this study two techniques were used--the modified root locus and the frequency response methods. Finally, a typical compensation for the

system was designed. For the chosen compensator the sensitivity of the system with respect to flight time and sampling period was checked.

In this report complete studies of the behavior of the system to sampling period variation are given for $0 \leq T \leq 1.0$. In completing these studies several analysis techniques were developed. Furthermore, it was necessary to redefine or define several terms. These are indicated in the text of the material.

OPEN LOOP STUDIES

A simplified set of equations for the Saturn V/S-1C Stage was furnished by NASA. A block diagram of these equations is shown in Figure 1. In order to aid in the study of the performance of the system as a function of sampling period, it is necessary to reduce the block diagram. First, the portion in the dotted lines in Figure 1 is attacked. Through the repeated use of the techniques of block diagram algebra, the dotted portion in Figure 1 is reduced to Figure 2. For avoiding the cumbersomeness of the expressions in the blocks in Figure 2, the expressions are redefined by the following sequence:

$$B_6 = \left[K_5 C_1 - \frac{57.3 m_{s_1} (\bar{x}_{s_1}^2 s^2 + \bar{g}) s}{s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2} \right] \cdot \left[\frac{s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2}{(s + K_5 K_7)(s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2) - \frac{m_{s_1}}{M} s^3} \right] \quad (1)$$

$$B_7 = K_7 + K_3 - \frac{m_{s_1} \left(\frac{\bar{x}_{s_1}}{57.3} s^2 + K_3 \right) s^2}{M(s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2)} \quad (2)$$

$$B_{8A} = \frac{57.3 m_{s_1} (\bar{x}_{s_1} s^2 + \bar{g}) \left(\frac{\bar{x}_{s_1}}{57.3} s^2 + K_3 \right)}{(s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2)} \quad (3)$$

Using these equalities, Figure 2 becomes Figure 3, which is easily reduced to Figure 4 when

$$G_1(s) = -C_2 + K_4 B_6 \quad (4)$$

and

$$B_{10} = B_8 + B_6 B_7 - C_1 \quad (5)$$

To decrease the complexity of the preceding expressions many of the common terms in (1), (2), and (3) are redefined as follows:

$$X = \frac{57.3M_{s_1}}{I} \quad (6a)$$

$$A = \bar{x}_{s_1} s^2 + \bar{g} \quad (6b)$$

$$B = \frac{x_{s_1}}{57.3} s^2 + K_3 \quad (6c)$$

$$E = s + K_5 K_7 \quad (6d)$$

$$D = s^2 + 2\delta_{s_1} \omega_{s_1} s + \omega_{s_1}^2 \quad (6e)$$

$$Y = \frac{m_{s_1}}{M} \quad (6f)$$

Using the expressions in (6), (4) becomes

$$G_1(s) = -C_2 + \frac{K_4 C_1 D - K_4 X A s}{E D - Y s^3} \quad (7a)$$

or

$$G_1(s) = \frac{-C_2 E D + C_2 Y s^3 + K_4 C_1 D - K_4 X A s}{(E D - Y s^3)} \quad (7b)$$

Similarly, (5) evolves as

$$B_{10} = \frac{XAB - C_1 D}{D} + \frac{K_5 C_1 (K_7 + K_3) D^2 - K_5 C_1 Y B D s^2 - XA(K_7 + K_3) D s + X Y A B s^3}{D(ED - Y s^3)} \quad (8a)$$

or

$$B_{10} = \frac{XABE - C_1 ED + C_1 Y s^3 + K_5 C_1 (K_7 + K_3) D + K_5 C_1 Y B s^2 - XA(K_7 + K_3) s}{(ED - Y s^3)} \quad (8b)$$

Eliminating the feedback loop in Figure (4) yields

$$G_{\phi_R}^*(s) = s G_1(s) G_2(s) \quad (9)$$

and

$$G_{\phi_R}(s) = G_1(s) G_2(s) \quad (10)$$

where

$$G_2(s) = \frac{B_{10D}}{s^2 B_{10D} - B_{10N}} \quad (11)$$

in which the rational polynomial, B_{10} , is defined as

$$B_{10} = \frac{B_{10N}}{B_{10D}} \quad (12)$$

The product, $G_1(s) G_2(s)$, is denoted as

$$G_T(s) = G_1(s) G_2(s) \quad (13)$$

Substituting for X, A, B, E, D, and Y from (6), the numerator, G_{TN} ,

and the denominator, G_{TD} , become

$$\begin{aligned}
G_{TN} = & \left[-C_2 \left(1 - \frac{m_{S_1}}{M}\right) - K_4 \frac{57.3m_{S_1} x_{S_1}}{I} \right] s^3 \\
& + \left[-C_2(2\delta_{S_1} \omega_{S_1} + K_5 K_7) + K_4 C_1 \right] s^2 \\
& + \left[-C_2(\omega_{S_1}^2 + 2\delta_{S_1} \omega_{S_1} K_5 K_7) + K_4 C_1 2\delta_{S_1} \omega_{S_1} - \right. \\
& \left. K_4 \frac{57.3m_{S_1} \bar{g}}{I} \right] s + \left[-C_2 K_5 K_7 \omega_{S_1}^2 + K_4 C_1 \omega_{S_1}^2 \right] \quad (14a)
\end{aligned}$$

and

$$\begin{aligned}
G_{TD} = & \left[1 - \frac{m_{S_1}}{M} - \frac{m_{S_1} x_{S_1}^2}{I} \right] s^5 + \left[2\delta_{S_1} \omega_{S_1} + K_5 K_7 - \right. \\
& \left. \frac{x_{S_1}^2 m_{S_1} K_5 K_7}{I} + \frac{K_5 C_1 m_{S_1} x_{S_1}}{M 57.3} \right] s^4 + \left[\omega_{S_1}^2 + 2\delta_{S_1} \omega_{S_1} K_5 K_7 \right. \\
& \left. - 57.3m_{S_1} \frac{x_{S_1}}{I} \left(\frac{\bar{g}}{57.3} - K_7 \right) + C_1 \left(1 - \frac{m_{S_1}}{M}\right) \right] s^3 + \\
& \left[K_5 K_7 \omega_{S_1}^2 - \frac{57.3m_{S_1} x_{S_1}}{I} \left(K_3 + \frac{\bar{g}}{57.3} \right) K_5 K_7 + \right. \\
& \left. C_1 (2\delta_{S_1} \omega_{S_1} - K_5 K_3) + \frac{K_5 C_1 m_{S_1} K_3}{M} \right] s^2 + \\
& \left[C_1 (\omega_{S_1}^2 - 2\delta_{S_1} \omega_{S_1} K_5 K_3) + \frac{57.3m_{S_1} K_7 \bar{g}}{I} \right] s + \\
& \left[-\frac{57.3m_{S_1} K_5 K_7 K_3 \bar{g}}{I} - C_1 K_5 K_3 \omega_{S_1}^2 \right] \quad (14b)
\end{aligned}$$

With the aid of (9), (10), and (13) and additional block diagram reduction the total block diagram of the system, Figure 1, is reduced

to Figure 5 with

$$G_3 = \frac{R'Y_{1B} + (\Sigma_E Y_{1B} - \theta_E Y'_{1B}) s^2}{57.3M_{1B}(s^2 + 2\delta_{1B}\omega_{1B}s + \omega_{1B}^2)} \quad , \quad (15)$$

$$G_4 = \frac{R'Y_{2B} + (\Sigma_E Y_{2B} - \theta_E Y'_{2B}) s^2}{57.3M_{2B}(s^2 + 2\delta_{2B}\omega_{2B}s + \omega_{2B}^2)} \quad , \quad (16)$$

$$K_8 = (Y'_{1\phi})(57.3) \quad , \quad (17)$$

$$K_9 = (Y'_{2\phi})(57.3) \quad , \quad (18)$$

$$K_{10} = (Y'_{1\phi RG})(57.3) \quad , \quad (19)$$

and

$$K_{11} = (Y'_{2\phi RG})(57.3) \quad . \quad (20)$$

With no compensation network and with samplers and hold devices, H_o , inserted in the ϕ_T and $\dot{\phi}_T$ paths, it is easily seen that if the loop was broken at ϕ_T the diagram pictured in Figure 6 represents the open loop transfer function. The relation between the sampled input and output for this block diagram is

$$\frac{C_{\phi}(z)}{R_{\phi}(z)} = \frac{\mathcal{Z} [H_o W_{ss} (G + K_8 G_3 + K_9 G_4)]}{1 - \mathcal{Z} [H_o W_{ss} s (G_T + K_{10} G_3 + K_{11} G_4)]} \quad . \quad (21)$$

Similarly, it can be shown that

$$\frac{C_{\dot{\phi}}(z)}{R_{\dot{\phi}}(z)} = \frac{\mathcal{Z} [H_o W_{ss} s (G_T + K_{10} G_3 + K_{11} G_4)]}{1 - \mathcal{Z} [H_o W_{ss} (G + K_8 G_3 + K_9 G_4)]} \quad . \quad (22)$$

From (21) and (22) the characteristic equation related to ϕ and $\dot{\phi}$ is

$$1 - \mathcal{Z}\{H_O W_{SS} [(1+s)G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]\} = 0 \quad (23)$$

To facilitate in finding the z-transform of

$$H_O W_{SS} [(s + 1) G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]$$

the partial fraction expansion of it is needed. Because of the complexity of this expression, finding the partial fraction expansion by hand calculations is very impractical. However, by employing the digital computer program in Appendix A much of this task is absorbed. This program yields the poles and partial fraction coefficients of $H_O W_{SS} [(s + 1) G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]$ for several values of flight times.

As an indication to the way the continuous system is affected by flight time, the loci of the poles of $H_O W_{SS} [(s + 1)G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]$ of prime importance as a function of flight time are shown in Figures 7 - 8.* From these figures it is seen that the imaginary parts of the poles of the first and second bending modes increase in a linear fashion with respect to flight time; whereas, the real parts remain almost constant for all flight times. This indicates that in the z-plane the paths of the loci of these poles will be circular in nature. The second sloshing mode loci have similar characteristics except that the real part of the poles changes much more than those of the bending mode poles. This is an indication of a spiraling type motion in the

*The poles of prime importance for the simplified version of the Saturn V/S-1C are the poles of the first and second bending mode, the second slosh mode, the drift and rigid body.

z-plane. As for the other poles, they possess somewhat different motions. For example, the rigid body poles move down the imaginary axis to the right until a flight time of about 90 seconds is achieved. Then it partially retraces the path in the other direction. This pole will have a similar motion in the z-plane. The drift poles have a similar action but in the opposite direction. They will also possess similar paths in the z-plane.

With the information from the algorithm in Appendix A, it is easy to obtain the z-transform of $H_0 W_{SS} [(s + 1)G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]$. This is seen in the following sequence. The partial fraction expansion of $H_0 W_{SS} [(s + 1)G_T + (K_8 + K_{10}s)G_3 + (K_9 + K_{11}s)G_4]$ is of the form

$$G_0(s) = \frac{K_1}{s + a_1} + \frac{K_2}{s + a_2} + \frac{K_3}{s + a_3} + \dots ; \quad (24)$$

the z-transform is thus

$$G_0(z) = \frac{K_1 z}{z - e^{-a_1 T}} + \frac{K_2 z}{z - e^{-a_2 T}} + \frac{K_3 z}{z - e^{-a_3 T}} + \dots \quad (25)$$

where the K's are the partial fraction coefficients and the a's are the poles of the open loop transfer function. The next step is to determine the movement of the poles of (25) as sampling period is varied. The computer program in Appendix B is used for obtaining this information. Plots of the movements of the poles of prime interest for several flight times are seen in Figures 9 - 25.

From these plots one immediately sees that the first and second bending mode poles move in very nearly circular paths. A first conclusion is that, since the first and second bending mode poles move in

approximate circular paths in a nearly linear fashion as flight time is varied, the opposite effect can be instigated by changing the sampling period. That is the bending poles can be maintained at approximately constant positions in the z-plane as flight time increases by changing the sampling period in the proper manner.¹ Another conclusion from these plots is that the bending mode poles are most affected by a variation in sampling period. This is primarily due to the fact that the imaginary part of the bending mode s-plane poles is much greater than any of the other poles. Thus, a feasible idea is that all the poles can be maintained in approximately a fixed position by varying the sampling period. However, whether this idea is workable will depend greatly on the effects on the system of the movement of the drift and rigid body poles with respect to flight time, since they move in such a way that varying the sampling period will not duplicate their previous positions. Another drawback to this idea is that if higher modes of bending and sloshing were included, they might be affected most by a variation in sampling. Their effect on the response of the system would depend greatly on the magnitude of their damping ratio as compared to those of the first and second bending modes and second slosh mode. If their damping ratio is large, compared to the damping ratios of the first and second bending modes and second slosh mode, their loci in the z-plane will spiral very quickly into a neighborhood near the origin while the other poles will still be lingering near the unit circle. In terms of frequency response of the system the movement of the higher modes in this manner may be interpreted as having little effect on the magnitude response but giving a significant contribution to the phase response.

CLOSED LOOP STUDIES

When a sampler and hold device are placed in the loop of a continuous feedback control system, the responses of the system are not the same as they were in the continuous system. In fact, these responses can be modified to some extent by a variation in the sampling rate. The insertion of the sampler does not alter the original s-plane poles of the open loop system. However, the closed loop poles can be in many cases affected greatly. This means that for a constant open loop gain there exists a manifold of closed loop poles for each open loop pole. Corresponding to each pole on the manifold there is a particular sampling period or sampling periods. A manifold of a pole can contain both stable and unstable members. In order for the closed loop system to be stable each closed loop pole must simultaneously be a member of the stable part of its manifold. If a pole's manifold does not possess a stable portion then it is impossible for the system to be stable unless it is modified. In some cases a system can be modified by changing the open loop gain. In other cases an additional network must be placed in the system. This network can either be a continuous network or a digital network.

Although there can be a set of sampling periods for which a system is stable, this does not necessarily mean that the system is relative stable.* Thus the set of sampling periods for which desirable characteristics of a system can be achieved is constrained even more. In order to determine if a system possesses desirable behavior patterns, investigation

*A system is relative stable if the desired stability margins have been achieved.

of many sampling periods is necessary. Several techniques for studying the behavior of a system to changes in sampling period have been considered. The two methods which were found to be the most useful are the modified root locus method and the frequency response method. The modified root locus method is a synthesis method in that it permits the study of the movement of the closed loop poles as sampling period is varied. On the other hand, the frequency response method is more applicable to design because it immediately presents the necessary information for compensating and determining relative stability. Both of these methods are incorporated into the study of stability as a function sampling for the simplified version of the Saturn V/S-IC.

A study of the stability of the system as a function of sampling period is instigated for sampling periods from 0 to 1.0 seconds (sampling frequencies from ∞ to 1.0 hz). This study is done in a two part procedure. First, the stability of the closed loop system for sampling periods from 0.0 to 0.49 seconds is determined by using the "modified root locus technique", which is implemented by the algorithm in Appendix D.* The results of this study for an open loop gain of 1.0 are shown in Figures 26-39; whereas, the results for an open loop gain of -1.0 are shown in Figures 40-53. Again only the poles of prime importance are shown.

An investigation of the plots will show that the loci actually do not begin at $T = 0$ but start at slightly above this value. Because of computational error it is practically impossible to obtain accurately the loci when T is very near 0. However, this does not mean that it

*A modified root locus is a locus as a function of sampling period.²

is impossible to determine the proximity of the poles of the closed loop system when $T = 0$. In fact, for an error sampled unity feedback control system which has an s-plane open loop transfer function that has n poles and $n - 2$ or less zeros, there are n z-plane poles of the closed loop transfer function at $z = 1$ when $T = 0$. The proof of this is as follows: Given an s-plane function of the form

$$G(s) = \frac{A_1 s^{n-2} + A_2 s^{n-3} + \dots + A_{n-1}}{s^n + B_2 s^{n-1} + \dots + B_{n+1}} \quad (26)$$

The z-plane closed loop transfer function of an error-sampled-unity feedback control system with the above s-plane open loop transfer function is of the form

$$\frac{C(z)}{R(z)} = \frac{\mathcal{Z}[G(s)]}{1 \pm \mathcal{Z}[G(s)]} \quad (27)$$

The characteristic equation becomes

$$1 \pm \mathcal{Z} \left[\frac{K_1}{s+a_1} + \frac{K_2}{s+a_2} + \dots + \frac{K_n}{s+a_n} \right] = 0 \quad (28)$$

where (24) has been expanded by partial fraction expansion. Taking the z-transform, (26) becomes

$$1 \pm \left[\frac{K_1 z}{z-e^{-a_1 T}} + \frac{K_2 z}{z-e^{-a_2 T}} + \dots + \frac{K_n z}{z-e^{-a_n T}} \right] = 0 \quad (29)$$

Letting $T \rightarrow 0$ and finding a least common denominator, (27) evolves as

$$(z-1)^n \pm [K_1 (z-1)^{n-1} z + K_2 (z-1)^{n-1} z + \dots + K_n (z-1)^{n-1} z] = 0 \quad (30)$$

or

$$(z-1)^{n-1} [(z-1) \pm z \sum_{i=1}^n K_i] = 0 \quad (31)$$

However, the sum of the partial fraction coefficients for a rational polynomial is zero if the denominator order is two or more greater than the numerator. Therefore (29) becomes

$$(z-1)^n = 0 \quad (32)$$

Thus when $T = 0$, there are n roots of the characteristic equation at $z = 1$.

Observing the figures of the system with an open loop gain of 1.0 (Figures 26-39), it is seen that the loci have similar shapes for all flight times.* One major difference of the loci is the rapidity of movement of certain poles for different flight times. The reason for this is the change in the poles as flight time changes. (These changes can be seen in Figures 7-8 which are the loci of the poles as a function of flight time.) These figures also demonstrate that a positive effect of closing the loop is that the open loop unstable drift poles moved into the unit circle, thus becoming stable. The negative effect of closing the loop with a gain of 1.0 is that for every sampling period in the set $(0, 0.49)$ there is at least one pole of the first or second bending modes which is beyond the unit circle. As for the slosh and rigid body poles, they were affected only slightly by closing the loop. The overall result is that for an open loop gain of 1.0 the system is not stable for any flight times when $T \in (0, 0.49)$.

On the other hand, the figures of the modified root locus for an open loop gain of -1 (Figures 40-53) indicate an opposite behavior in the movement of the drift and second bending mode poles. Instead of

*When an open loop gain of 1.0 is used, the result is a negative feedback system; whereas a -1.0 open loop gain gives a positive feedback system.

the drift poles moving into the unit circle, one moves farther away from the unit circle while one stays in a vicinity of $z = 1.0$. The first and second bending mode poles loci are confined to the interior of the unit circle for most sampling periods. As in the case of the open loop gain of 1.0, the second slosh mode and rigid body poles are affected very little, and thus they remain very closed to their open loop loci for most sampling periods. The conclusions drawn from these figures are that the system is more unstable for a gain of -1.0 than for a gain of 1.0. Furthermore, it is conceived that it would be more difficult to stabilize the system for a gain of -1.0.

The second part of the investigation is a study of sampling periods from 0.5 to 1.0. A different approach for studying the stability as sampling period is varied is used. The approach is the frequency response method. Using this approach in conjunction with the Nyquist stability criterion for a sampled-data system, stabilitywise, these studies produce parallel results to those for sampling periods between 0 and 0.49, that is for most flight times the system is unstable.³ However, there is a set of sampling periods in which the system exhibits the nearest stable characteristics of any sampling periods that are studied. This neighborhood of sampling periods is centered at approximate 0.74 seconds. The most interesting property of this neighborhood is that the system is stable for some flight times. A Nyquist plot for a sampling period of 0.79 seconds and flight time of 40 seconds is shown in Figure 54. Studying this diagram and giving consideration to the negative frequency portion, which is not shown, a total of two counterclockwise encirclements of the (0 dB, 180°) point is counted. Using the Nyquist stability criterion for a sampled-data system it is deduced, since the open loop

system has two poles outside the unit circle, that the system is stable. However, this is not the case for all flight times. Because the open loop rigid body pole moves near the origin in the manner indicated in Figure 8 (along with the open loop zero movements with respect to flight time), the low frequency characteristics of the system vary to some extent. Some of this variation of response at low frequencies causes the high frequency magnitude to be offset by a constant which depends upon the flight time. This causes either or both of the 180° crossing of the Nyquist path in the proximity of the point $(0, 180^\circ)$ to cross at a crossing greater than 0 dB for some of the other flight times. This results in system instability. In order to give a simplified complete picture of the relative stability of the system for sampling periods in the neighborhood of 0.74 seconds, plots of gain margin and phase margin as a function sampling period are presented for several flight times.

Before considering the information conveyed by these plots, a discussion of gain and phase margins of systems that are open loop unstable is in order. The gain and phase margins of a system that is open loop unstable are not as well defined as a system that is open loop stable. For instance, using the Nyquist criterion an open loop system is stable if there is no crossing of the 180° line with a gain greater than 0 dB; but for an open loop unstable system this is not necessarily true. In fact, there must be crossing of the 180° line with gains greater than 0 dB. This is because if the system is to be stable there must be encirclements of the point $(0, 180^\circ)$. Hence, there can be multiple crossings of the 0 dB circle (which are used for determining phase margins), and there can be multiple crossings of

180° line (which are used for determining gain margins). The question is which pair of crossings should be used for yielding the correct stability margins. The answer to this question depends greatly on the system under consideration. However, for most systems the following general procedure can be used:

1. By studying the Nyquist diagram, find the critical parts of the Nyquist path. This is determine what parts of the Nyquist diagram that when least affected will either result in instability or result in stability of the system.
2. From the critical parts of the Nyquist diagram the phase and gain margins should be determined. If there are several gain or phase margins, the ones which cause greatest or nearest instability should be chosen.

Note: It is possible that this will give a gain margin greater than 1.0 or a negative phase margin and the system be stable.* When this is the case the reciprocal of the gain margin should be used and the negative sign on the phase margin should be disregarded; otherwise, they should assume their normal form. An example of this occurrence for phase margins can be deduced from Figure 54 for the part of the path between $\omega = 0.87$ and $\omega = 5.81$. If the phase of the system had been 30° more in the counterclockwise direction, the phase margin of the system would have been negative, but the system would have still been stable. Similarly, a gain margin example can be deduced from Figure 54 if the 180° crossing between $\omega = 0.13$ and $\omega = 0.58$ had been very close

*Some times positive gain margins are denoted as lag phase margins and negative phase margins are denoted as lead phase margins.

but still to the left of the 0 dB point; then, the gain margin would be less than 1.0 but the system would still be stable.

Now that the method for determining gain and phase margins has been presented, it is possible to analyze the stability margin plots in Figures 55-60. From these graphs it is seen that for lower flight times (20 - 60 seconds) the system is stable and as flight time increases the margins of stability decrease until the system is unstable at higher flight times (80 - 120 seconds). As has been previously stated this is caused by the effects of the rigid body pole and the open loop zeros on the high frequency response of the system. Another interesting point about these curves is that the point of maximum stability decreases as flight time increases.* In fact, it decreases from a sampling period of 0.81 seconds at TF = 20 seconds to about 0.738 seconds at TF = 120 seconds. This indicates the possibility of maintaining stability margins by changing sampling period. The failure or success of such a scheme would depend greatly on the compensator to be employed.

*Maximum stability is defined as the sampling period where the system is nearest to being stable or has its greatest stability.

COMPENSATION OF SYSTEM

First, the possibility of using the root locus for compensating will be investigated. Figure 61 is a regular root loci of the first bending mode, second bending mode, second slosh mode, and the drift poles of the simplified version of the Saturn V/S-1C for a flight time of 20 seconds and sampling period of 0.37 seconds.* From this figure, it is observed that breakaway angles of the first and second bending modes are such that these poles move outside the unit circle as open loop gain is increased. It is also seen that the open loop unstable drift poles have moved into the interior of the unit circle. By considering the change in these poles as being linear in open loop gain and extrapolating points between the extremities of the curve, one can deduce that the system can be stabilized at this flight time by reducing open loop gain. However, this will not produce relative stability.** Another possible method of stabilizing the system without reducing open loop gain is to place a compensator with poles and zeros located so that the break-away angles of the first and second bending mode poles break directly into the interior of the unit circle. At the same time this compensator should not significantly adversely affect the loci of the second slosh mode and drift poles. With the addition of the open loop zeros to the given figure, a desirable compensator can be achieved. Obtaining a compensator in this manner is unnecessarily

*A regular root loci is a root loci as a function of open loop gain.

**Relative Stability is stability with stability margins that are as good as or better than the desirable margins. Desired stability margins in this report are greater than 35° phase margin and a gain margin of 2.0 or greater.

laborious. An alternate and less cumbersome procedure is the frequency domain approach.

The design of a compensator in many cases is made easier by applying the techniques of two frequency response methods, Nyquist and Bode. First, the Nyquist criterion can be used to determine if the system is stable, and what changes in the response must be made in order to stabilize the system. Closed loop stability of a system which is open loop unstable cannot be obtained from the Bode plots as easily as it can be obtained from the Nyquist plots. However, in most cases the Bode plots are very helpful in designing compensators. When using the Bode criterion it is advantageous to make the plots as functions of the imaginary frequency ω_w so that the usefulness of the straight line asymptotes will be profitable.

Since it is necessary to compensate the system for stability purposes, it was decided to compensate it at $T = 0.37$ seconds. The reason for this choice is that at this sampling period the open loop z -plane poles of the first and second bending modes are very near each other. Thus, it is conjectured that the effects of both might be reduced with the same filter. In order to determine what type of filter is needed at this sampling period, a thorough study of the Nyquist plot of the unstable system is made (See Figure 62). By observing the given Nyquist plot and using the Nyquist stability criterion given in Kuo³ it is seen that there are 0 encirclements of the $(0, 180^\circ)$ point. Since there are two open loop poles beyond the unit circle, this indicates that two closed loop poles are outside the unit circle. If stabilization of the system is to be achieved, two clockwise encirclements of the $(0, 180^\circ)$ point are needed. Several observations of the Nyquist plot reveal that this might be

accomplished by a single pole compensator if it was placed so that the 180° crossing between $\omega = 2.5$ and $\omega = 5.7$ would occur at a point less than 0 dB. System stabilization could be achieved with this simple compensator; however, the stability margins could not be met. Additional compensation is needed. For aiding in determining the complete compensation function, the Bode plot as a function of the imaginary frequency ω_w (See Figure 63) was made. From the Bode plot the following w-domain compensator is chosen:

$$G_c(j\omega) = \frac{0.316 \frac{j\omega}{0.1} + 1}{\frac{j\omega}{0.2} + 1 \frac{j\omega}{1} + 1} \quad (33)$$

The w-domain frequency response of this compensator is plotted in Figure 64. In the z-domain the compensator is

$$G_c(z) = \frac{0.079(1.1z^2 + 0.2z - 0.9)}{(1.2z^2 - 0.8z)} \quad (34)$$

After cascading the compensator with the open loop transfer function the Nyquist plot information for the compensated system for TF = 40 seconds and T = 0.37 seconds is obtained and plotted in Figure 65. For the chosen flight time and sampling period the system is seen to exhibit desirable stability margins.

However, this study is only applicable for one flight time and one sampling period. Using the above compensator, a study of the sensitivity of the stability margins as a function flight time and sampling period is initiated. Plots of these studies are made in Figures 66-68. For the three flight times it is observed that for the chosen set of sampling periods the curves have similar general characteristics. A major

difference that is observed is a magnification factor. The difference in magnification factor results in relative instability (stability margins were not met) for $T_F = 120$ seconds. The conclusions drawn is that for the particular compensator a different magnification factor for each flight would be needed in order to obtain optimal relative stability. The sampling period can remain constant at the value for which the compensator was designed, $T = 0.37$ seconds. The result is that a relatively simple compensator can reduce the effects of two bending modes simultaneously.

CONCLUSION

In this report the control system of the Saturn V/S-1C has been studied to determine the effects of varying the sampling period. In accomplishing this task two analysis procedures were employed. These two techniques are the modified root locus method and the frequency response method. The modified root locus shows the exact location of the the closed loop poles as sampling period. From the modified root locus plots the adverse conditions, caused by closing the loop, on the poles of the system were observed directly. Although this method certainly produces a measure of relative stability, it does not easily allow for the determination of the classical stability margins. For this reason a more applicable method was sought and found among the frequency response methods. The Nyquist plot was found to furnish sufficient stability information, and by combining it with the Bode diagram, designing procedures became less laborious.

The studies were conducted for sampling periods of $0 \leq T \leq 1.0$. They showed that the system is not stable for all flight times for any sampling periods in this interval. The system was stable for lower flight times when the sampling period was approximately 0.74 seconds. At higher flight times the system became unstable because of the effect at low frequency that rigid body pole and open loop zeros had on the higher frequency response of the system. Phase margins and gain margins were plotted as a function of sampling periods in the neighborhood of 0.74 seconds. These curves indicated that maximum stability decreased as flight time increased. Thus the conclusion drawn was that if the

system was compensated and operated here that similar characteristics could be achieved by reducing the sampling period as flight time increased.

Similar studies were instigated for sampling periods around 0.37 seconds. However, these indicated that the system's characteristics changed very little with respect to flight time. An explanation for this was deduced from the fact that the s-plane poles are mapped into the z-plane through the transformation $z = e^{-pt}$, where p is a typical s-plane pole. Thus, it was seen that if $T = T_1$ and p was varied by Δp that Δz would result; if $T = T_2$ where $T_2 > T_1$ and p was varied by Δp there would result a Δz_2 which was larger than Δz_1 . In other words, for larger sampling periods the system became much more sensitive to a change in the s-plane poles. Not only did the preceding explain the sensitivity of characteristics to sampling period, it also justified that the use of small sampling rates could be dangerous. When using a small sampling rate, a slight variation in the sampling rate or a slight variation in the system parameters could cause a great change in the overall characteristics of the system. This could very possibly result in system instability. Thus, the possibility of such great changes should be avoided by using larger sampling rates.

As was mentioned previously no sampling periods produced stability for all flight times. Stability could only be achieved by compensation. A typical compensator was designed for $T = 0.37$ seconds. The sensitivity of the stability margins were studied as a function of sampling period and flight time in order to determine the applicability of the compensator. For lower flight times and a wide range of sampling periods the compensator was found to meet stability requirements. However, for

flight times beyond 120 seconds relative stability was not obtained for any sampling periods. Again, the cause for this was the rigid body pole and open loop zeros effect on higher frequency responses.

In this report it has been shown that under certain conditions the system becomes very sensitive to variations in sampling period and to s-plane open loop pole variations. Since the open loop poles depend upon many parameters of the continuous systems, it has been decided that future studies should determine the sensitivity of the s-plane poles to parameter variations. The justification of this is due to the fact that the present modeling of the control system is probably incorrect. Thus, it should be known what could be expected if the parameters were too much in error or varied in a certain direction. Furthermore, it is conceived that disturbances in the system could be treated as parameter variations. It would be almost impossible to handle them otherwise.

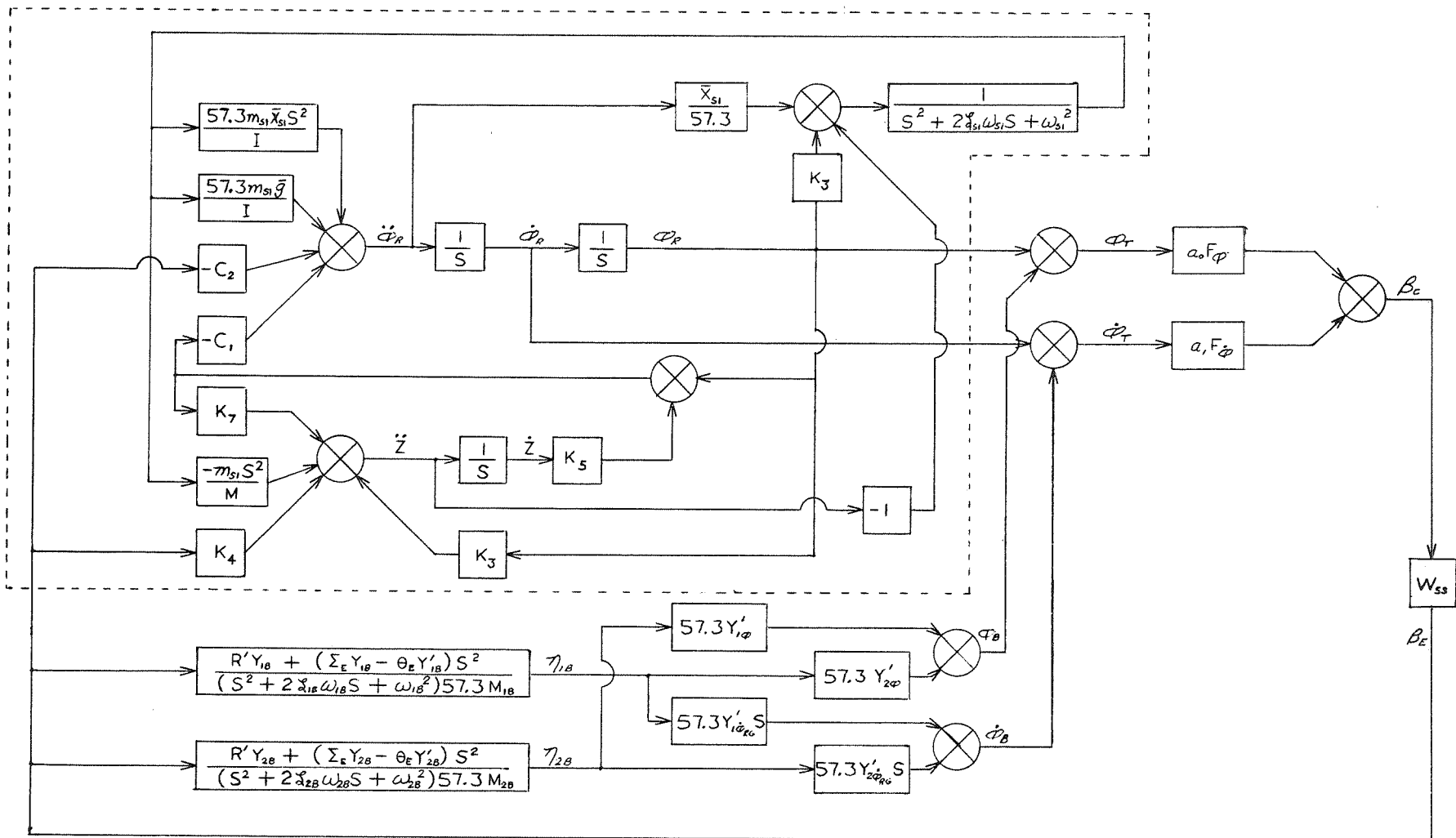


Figure 1. Simplified Block Diagram of Saturn V/S-1C.

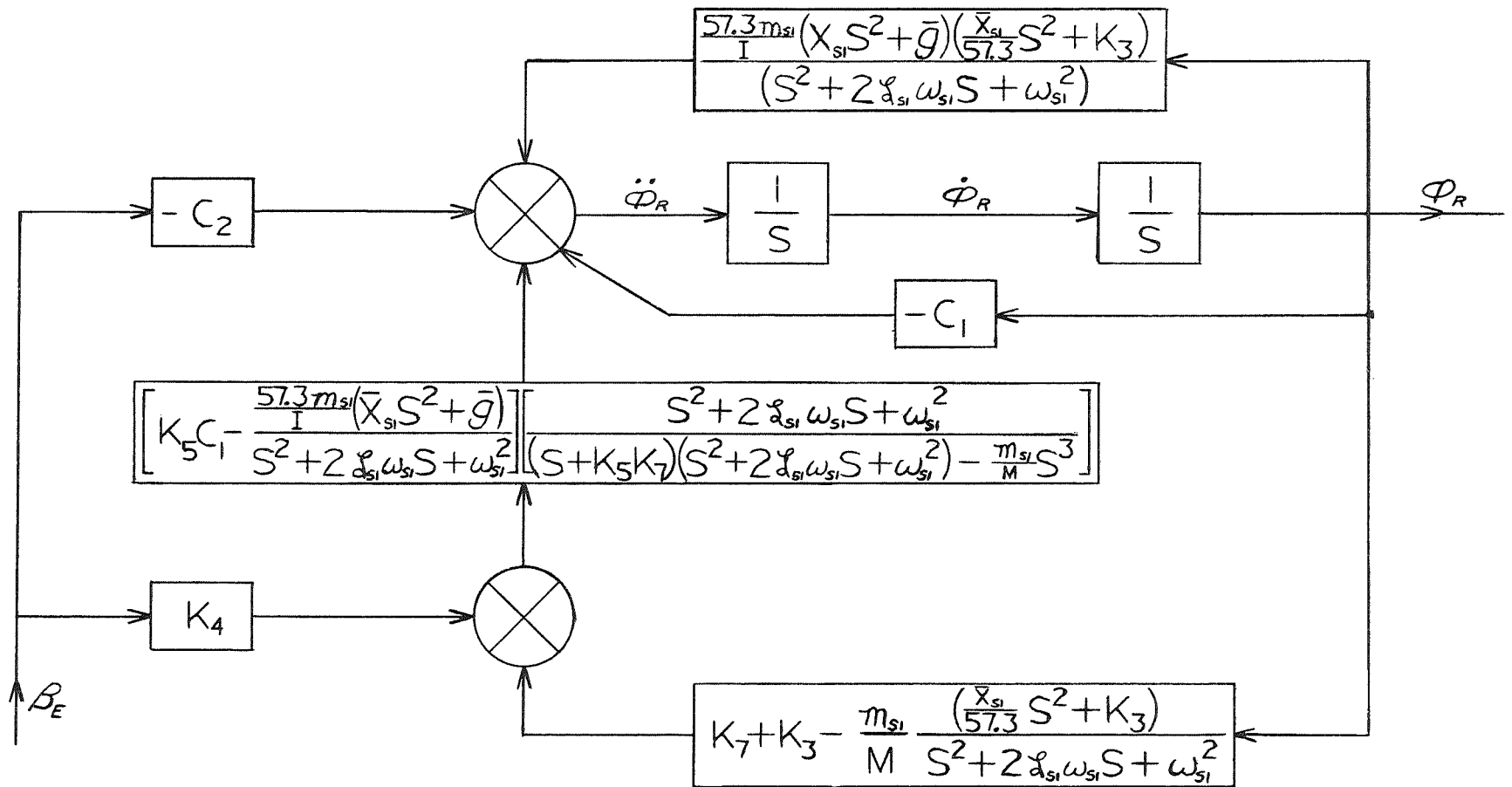


Figure 2. Reduced Simplified Block Diagram of Saturn V/S-1C.

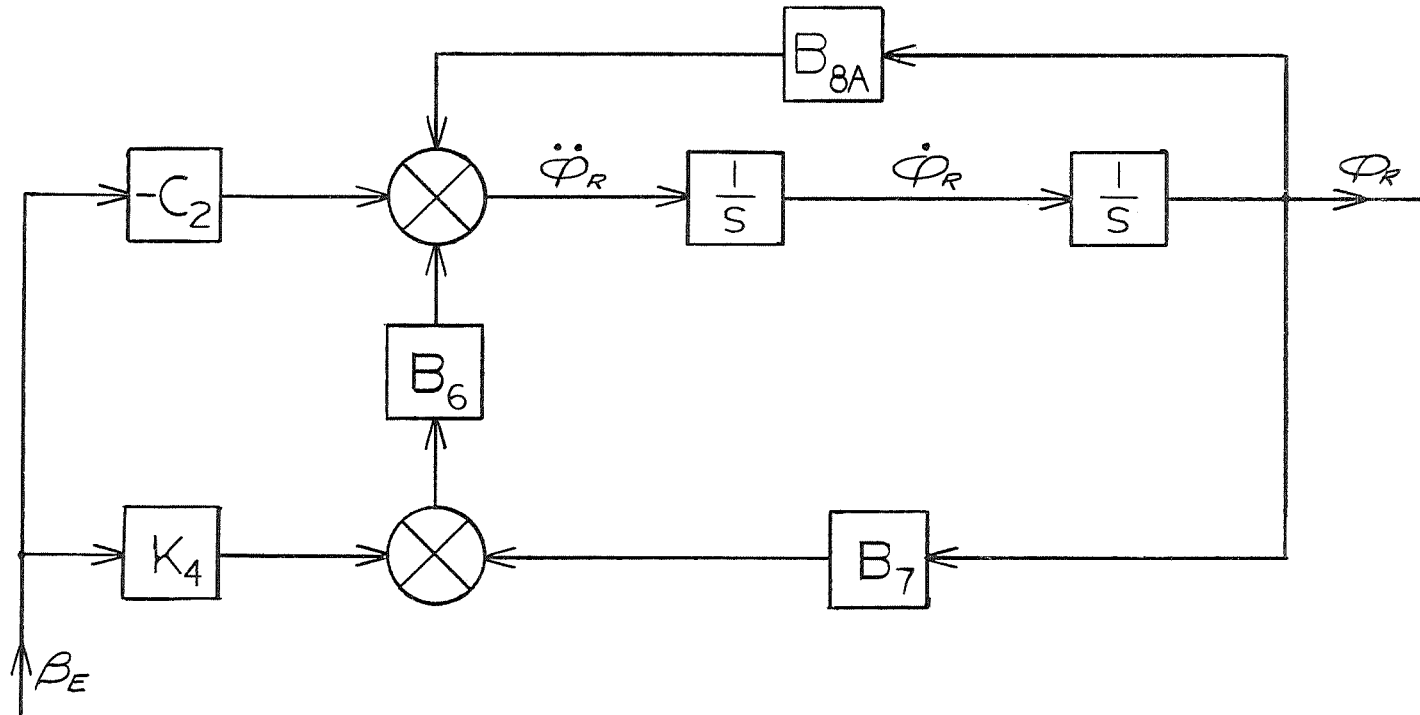


Figure 3. A Block Diagram for Redefining the Blocks of Figure 4.

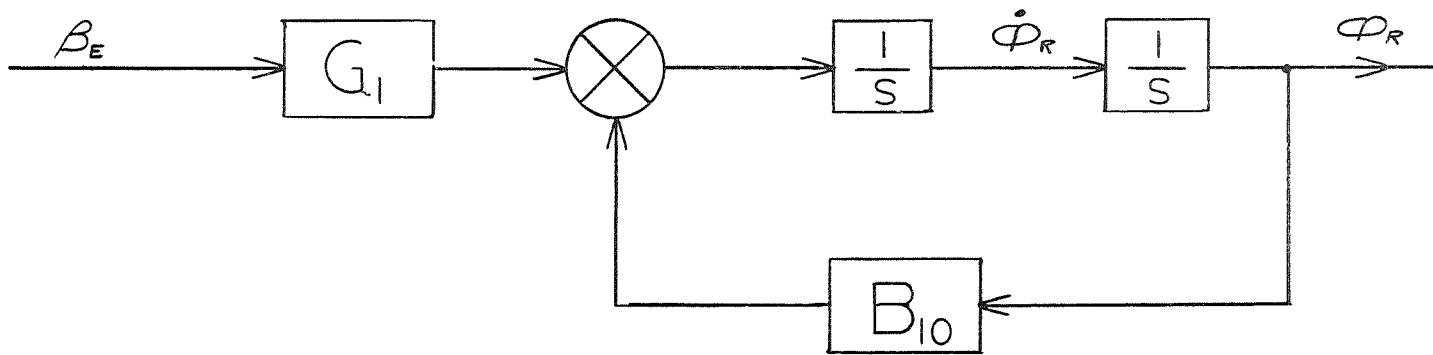


Figure 4. Block Diagram Which Results from Figure 3.

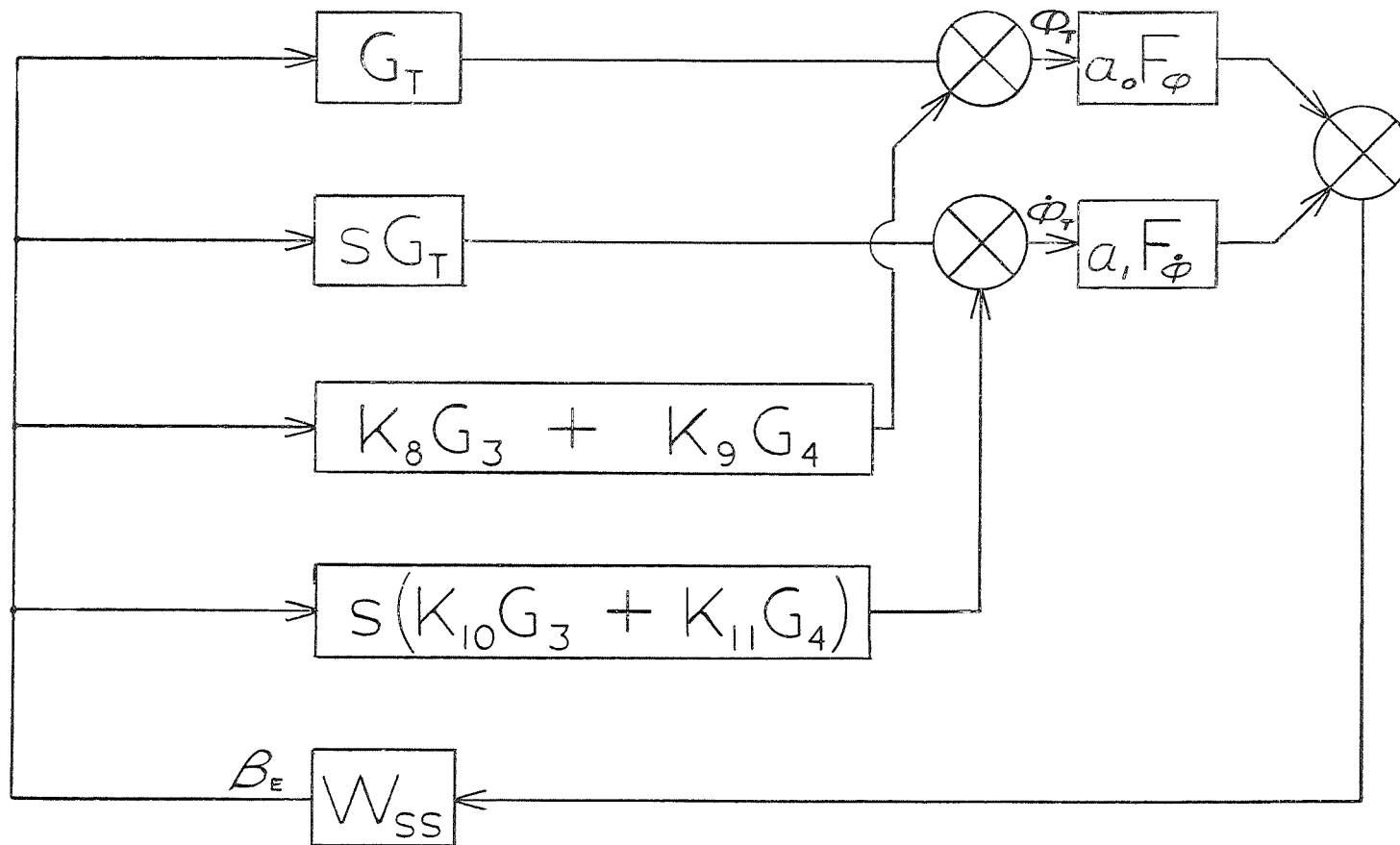


Figure 5. Simplified Block Diagram of Saturn V/S-1C Stage After Several Block Diagram Reductions.

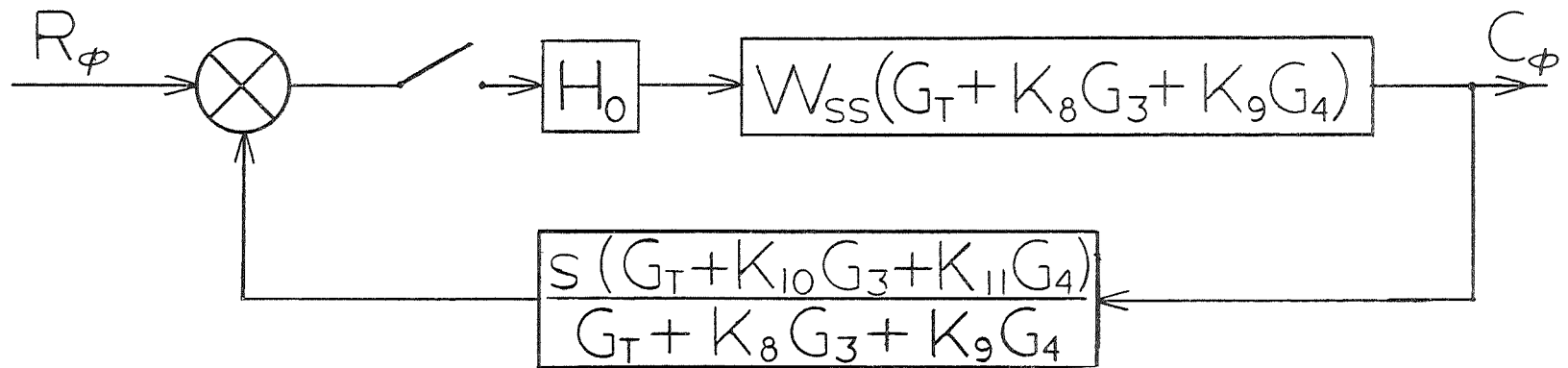
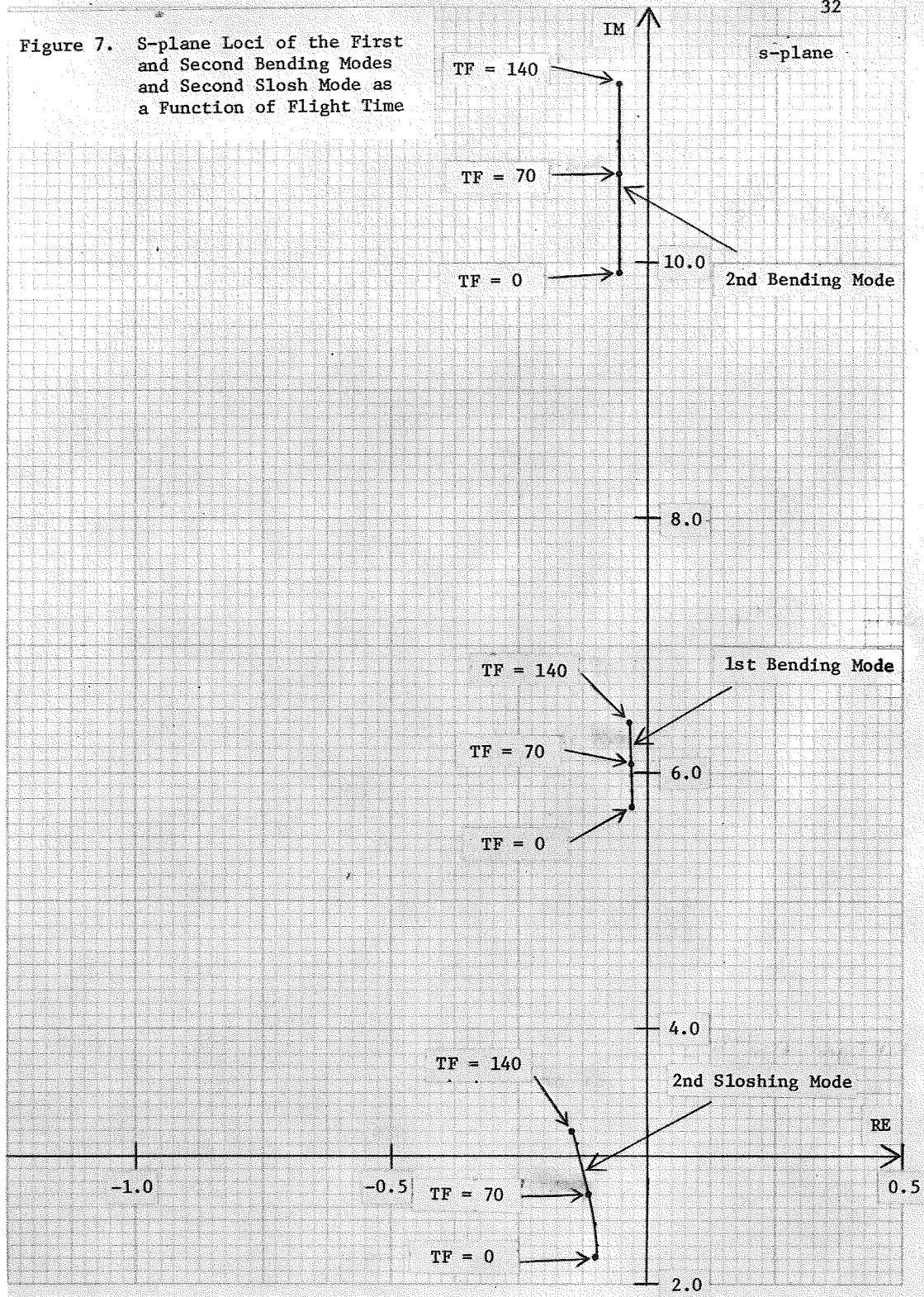


Figure 6. Block Diagram of the Open Loop Transfer Function, G_ϕ .

Figure 7. S-plane Loci of the First and Second Bending Modes and Second Slosh Mode as a Function of Flight Time



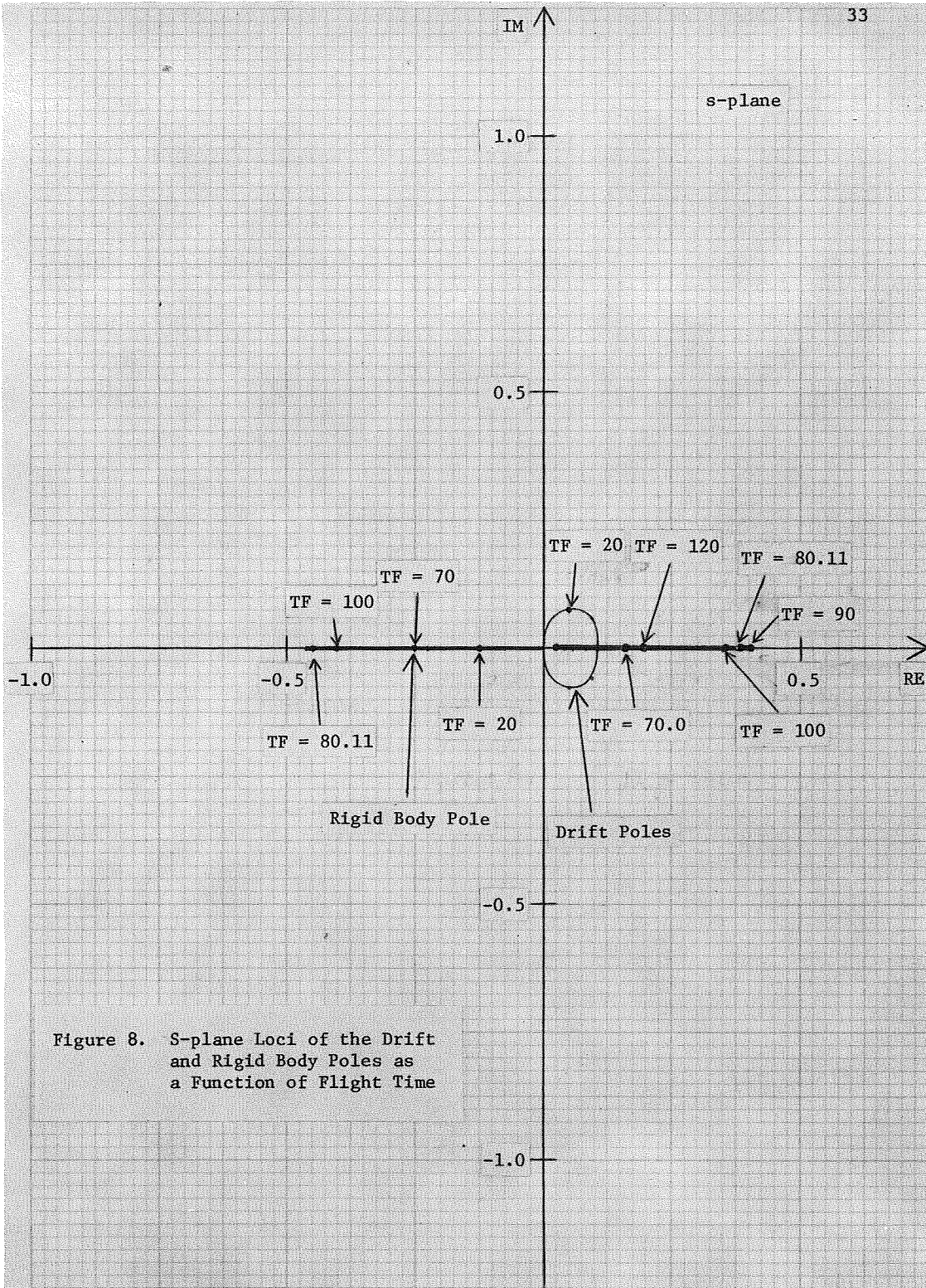


Figure 8. S-plane Loci of the Drift and Rigid Body Poles as a Function of Flight Time

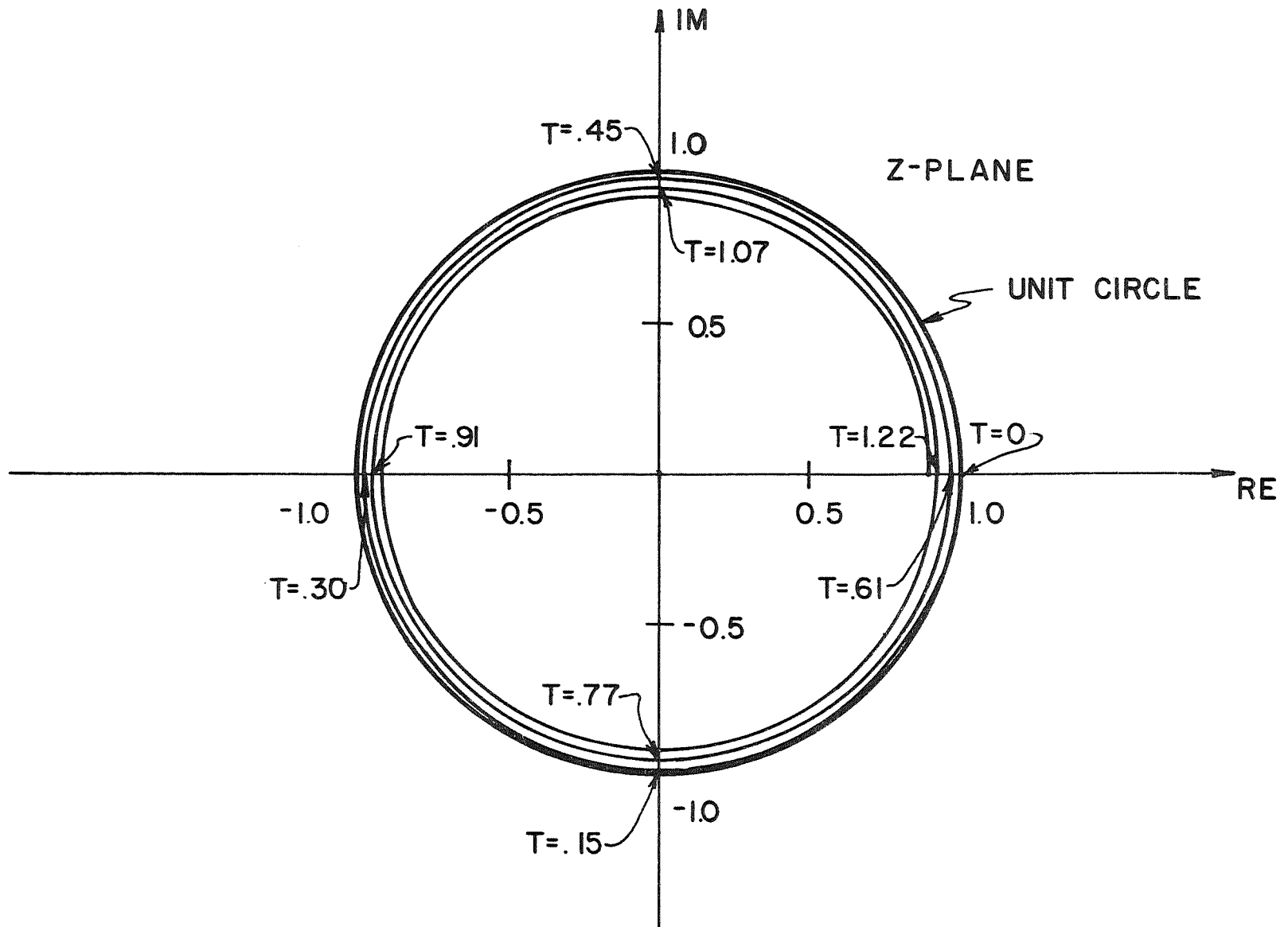


Figure 9. Plot of a Pole of the Second Bending Mode as a Function of Sampling Period for $TF = 40$ Sec.

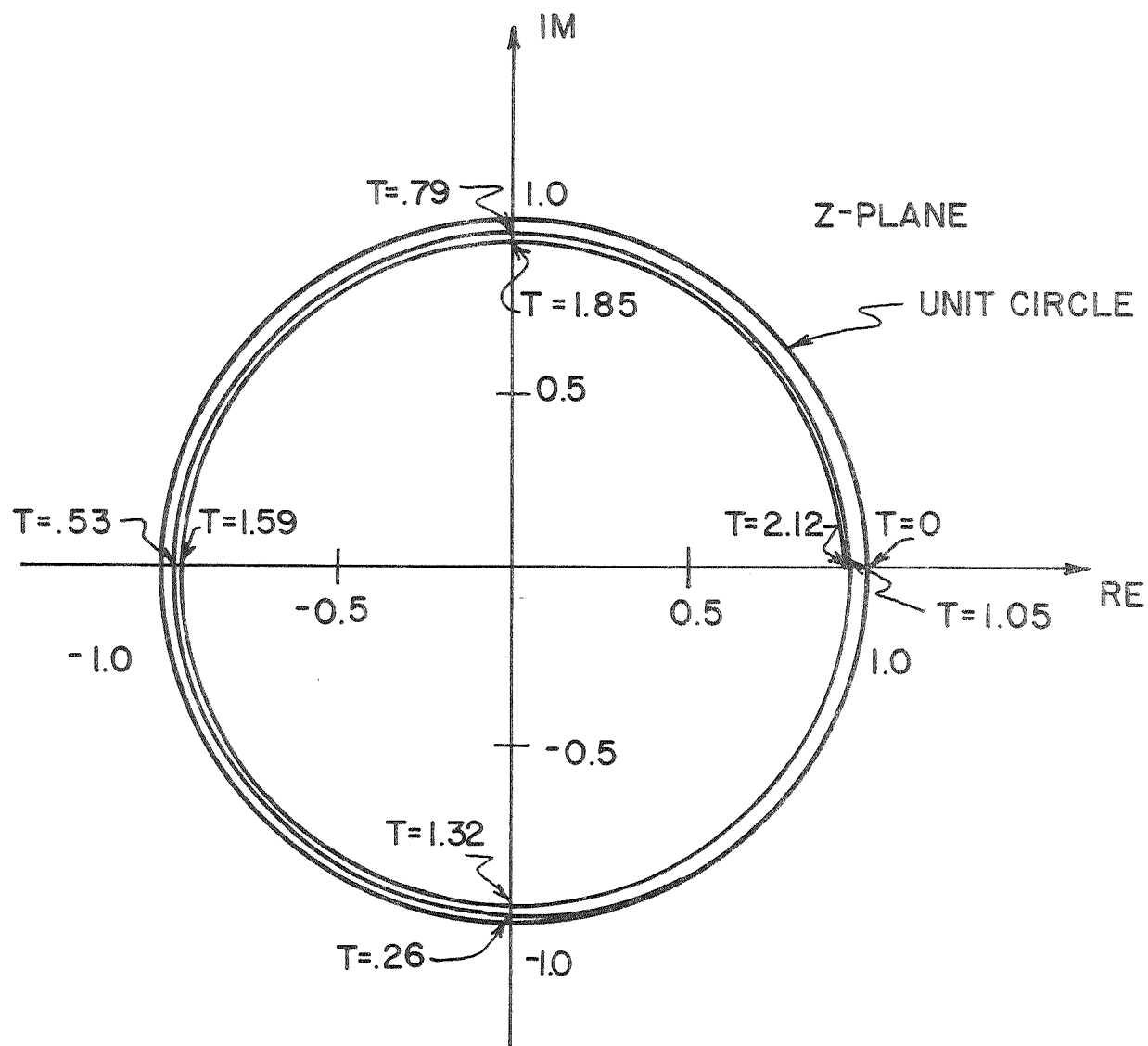


Figure 10. Plot of a Pole of the First Bending Mode as a Function of Sampling Period for TF = 40 Sec.

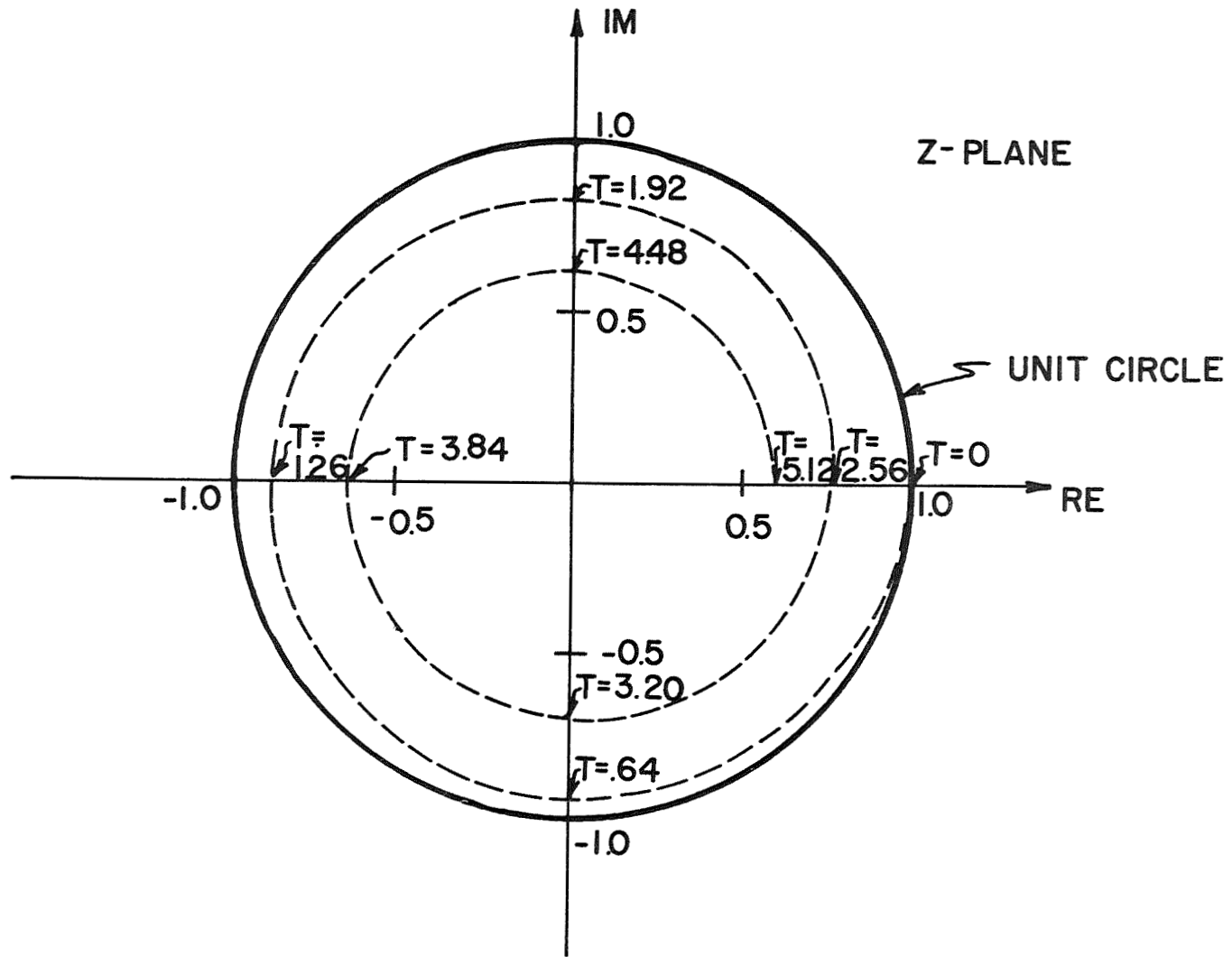


Figure 11. Plot of the First Slosch Mode Pole as a Function Sampling Period for TF = 40 Sec.

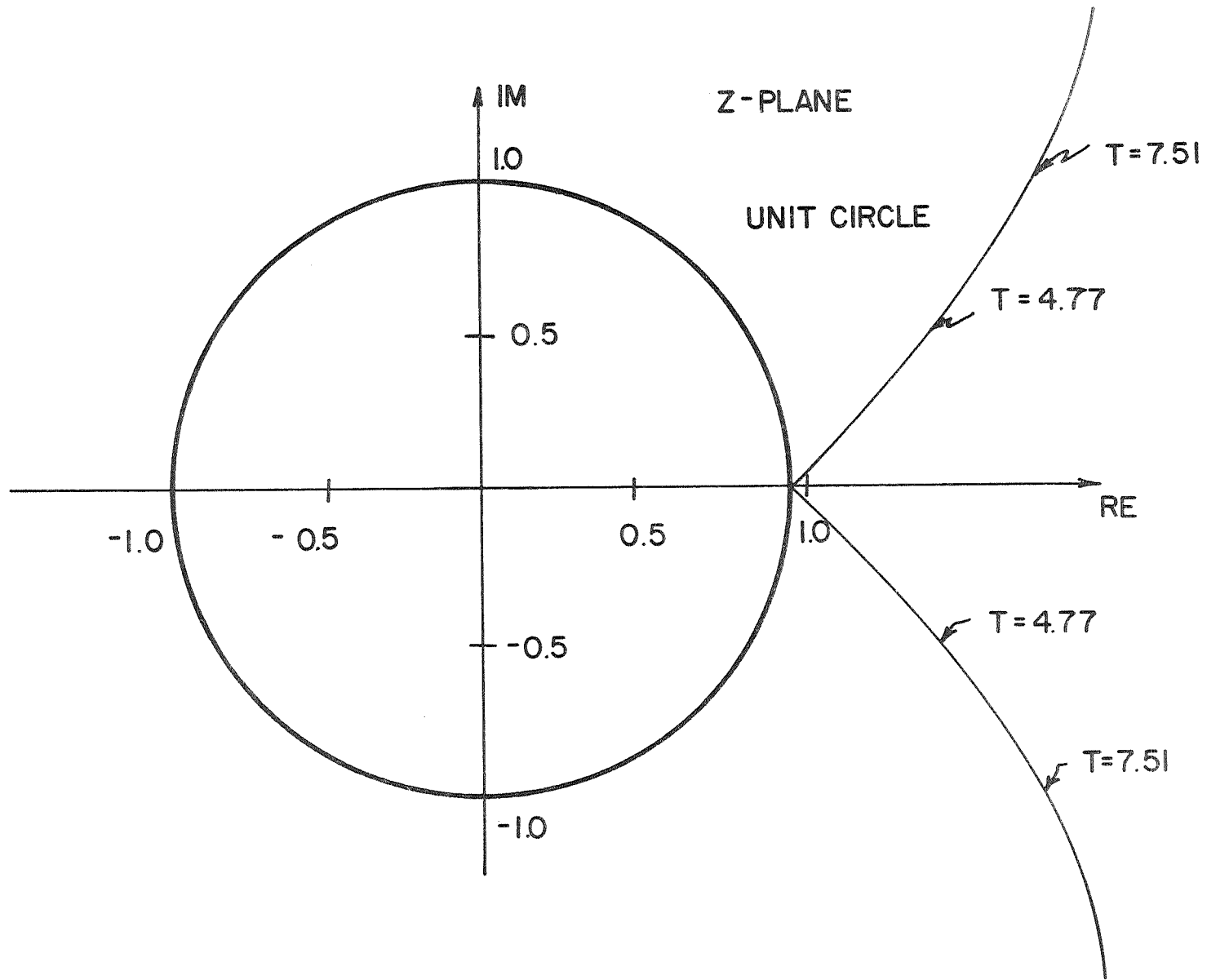


Figure 12. Loci of Drift Poles as a Function of Sampling Period for
 TF = 40 Sec.

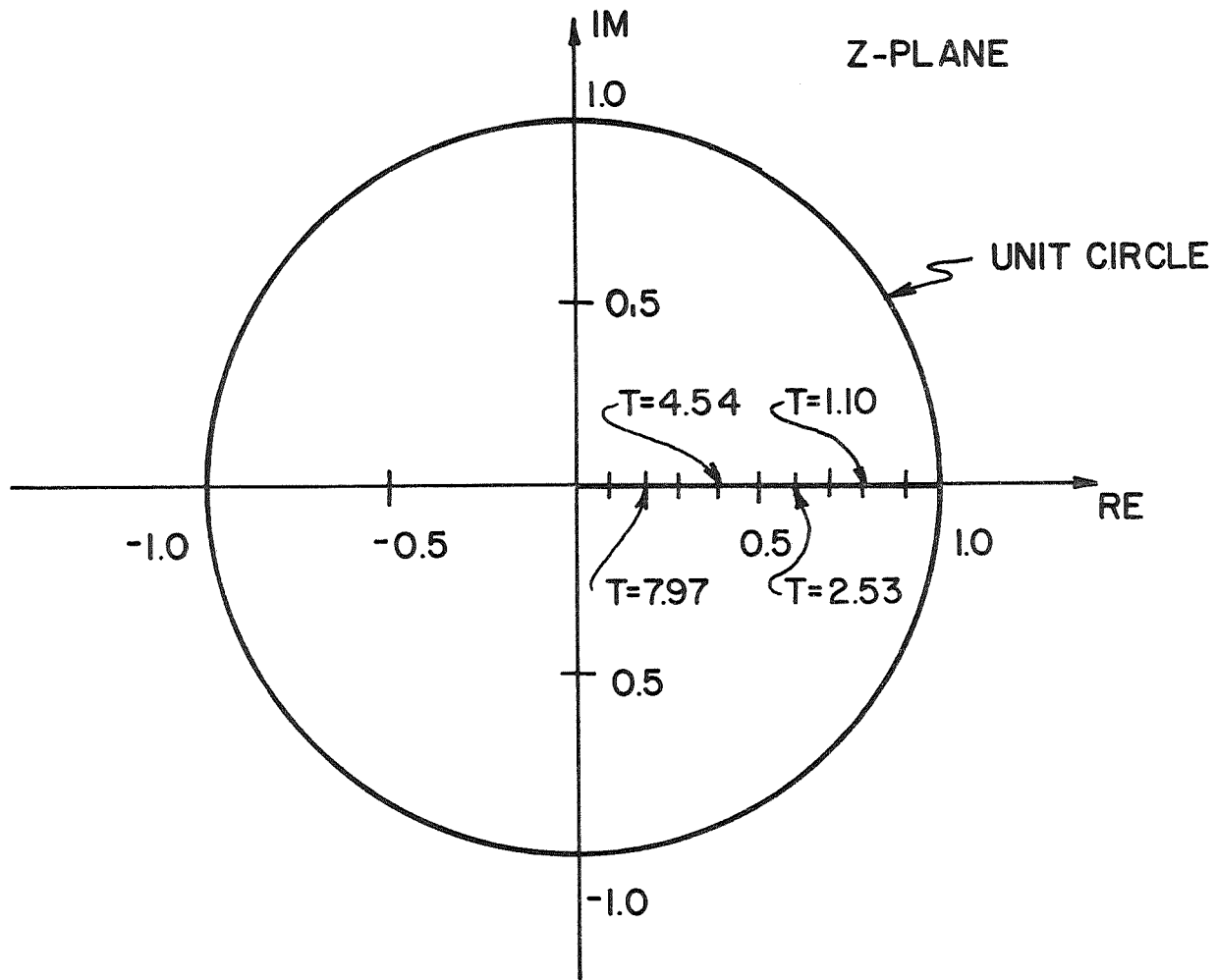


Figure 13. Movement of a Rigid Body Pole as a Function Sampling period for $TF = 40$ Sec.

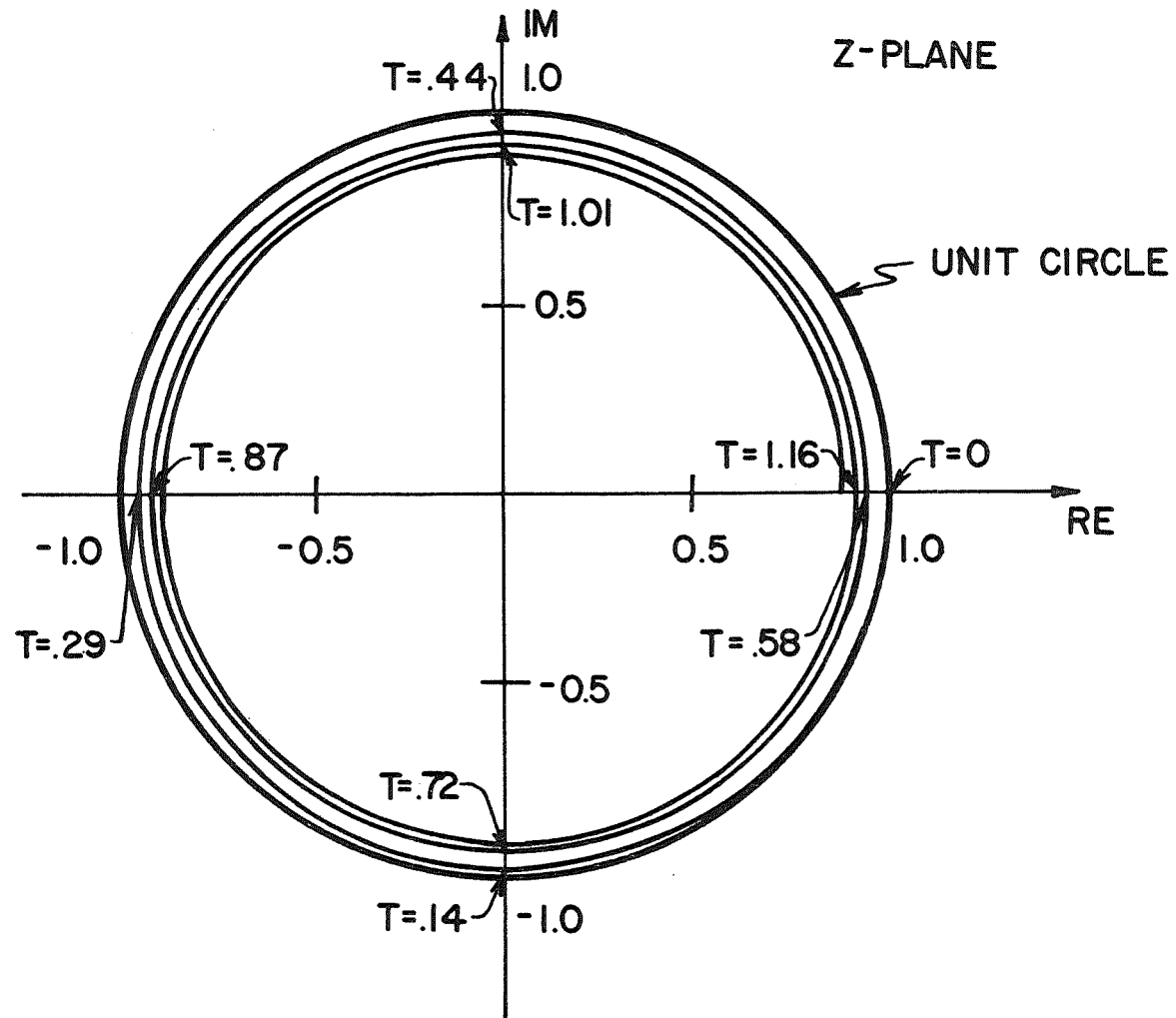


Figure 14. Indication of the Movement of a Second Bending Mode Pole as a Function of Sampling Period for $TF = 80.11$ Sec.

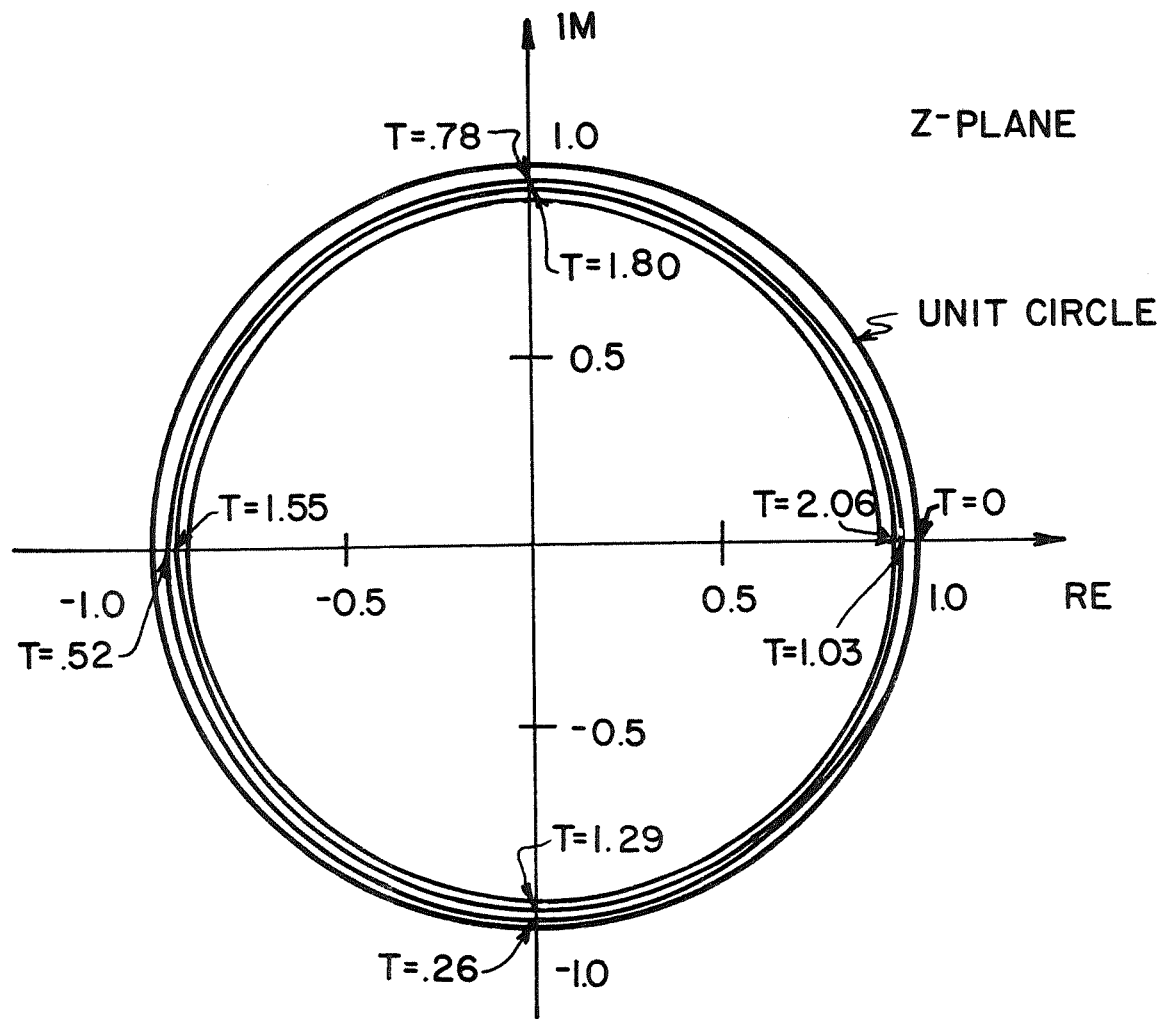


Figure 15. Plot of a First Bending Mode Pole as the Sampling Period is Varied for $TF = 80.11$ Sec.

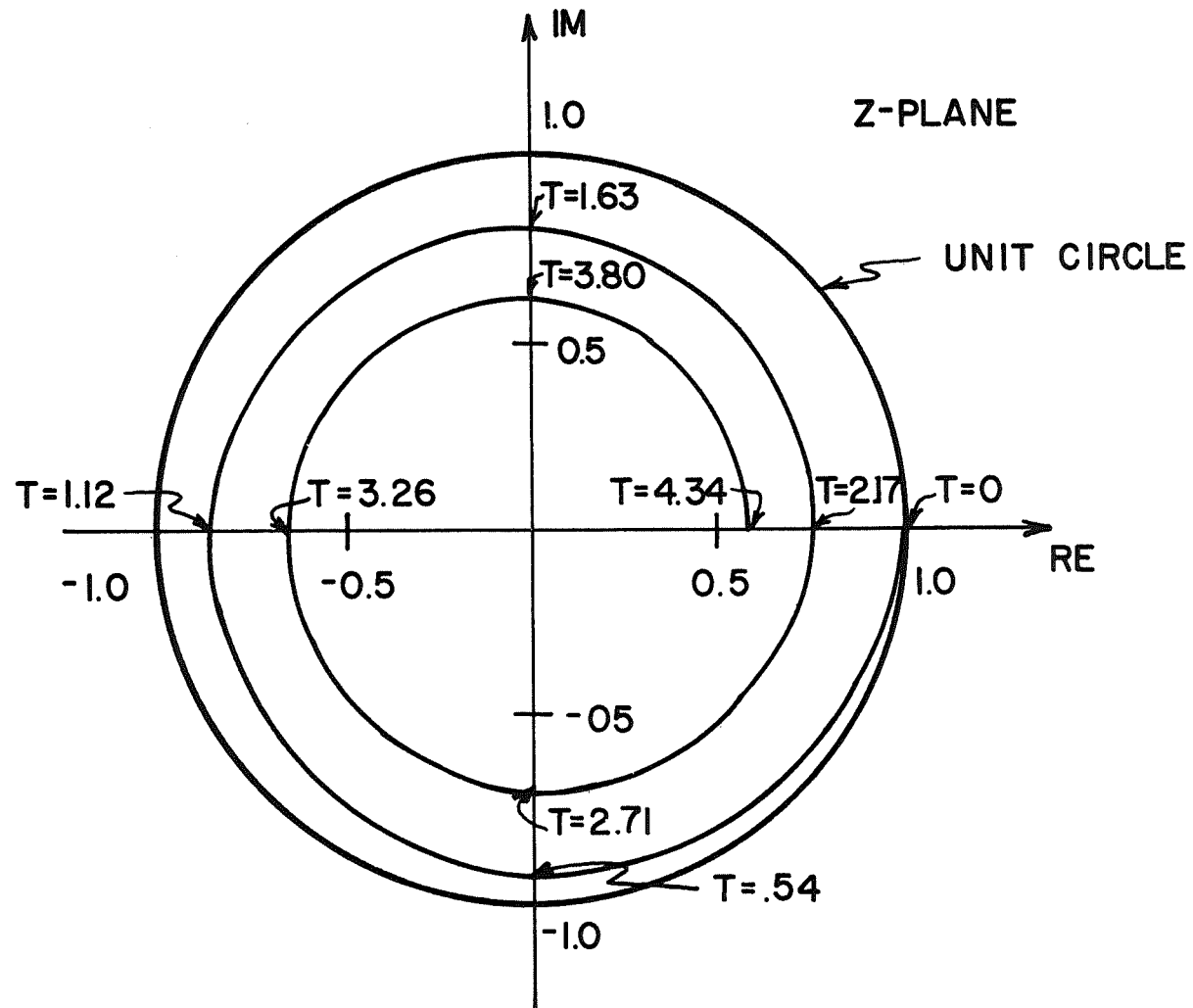


Figure 16. Plot of the First Slosch Mode Pole as a Function Sampling Period for TF = 80.11 Sec.

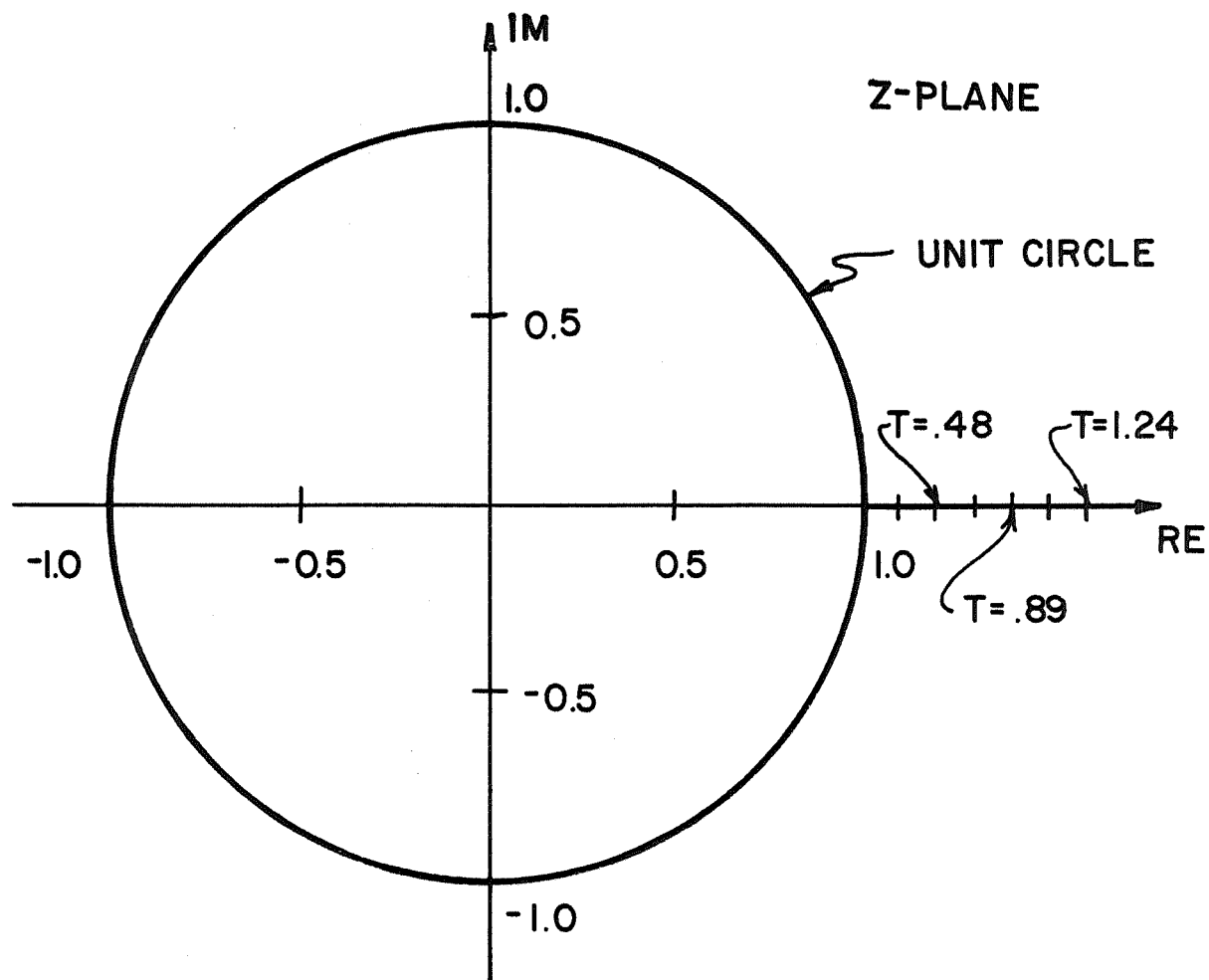


Figure 17. Locus of a Drift Pole as Sampling Period is Varied for $T_F = 80.11$ Sec.

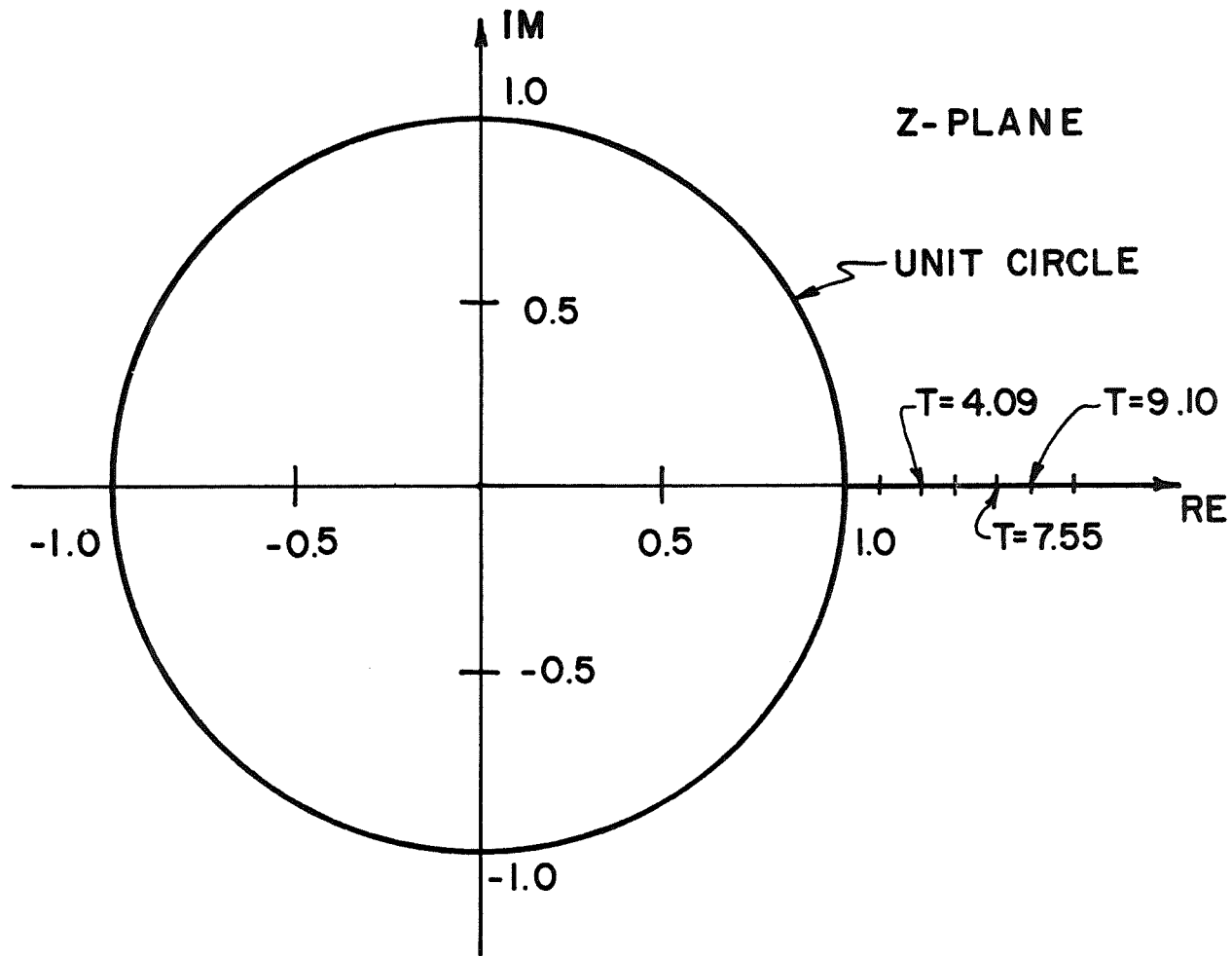


Figure 18. Plot of a Drift Pole as a Function Sampling Period for $TF = 80.11$ Sec.

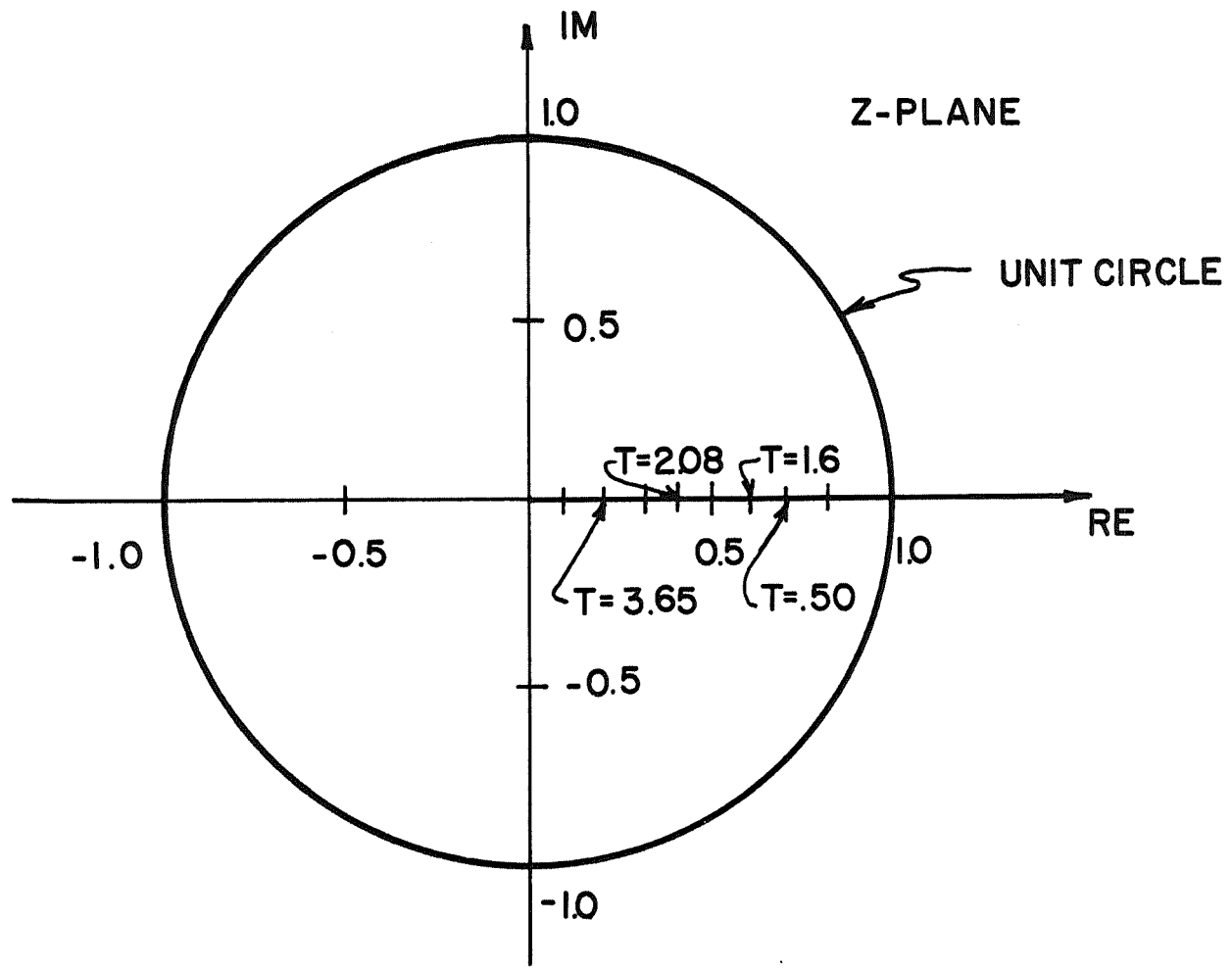


Figure 19. Rigid Body Locus as Sampling Period is Varied for $T_F = 80.11$ Sec.

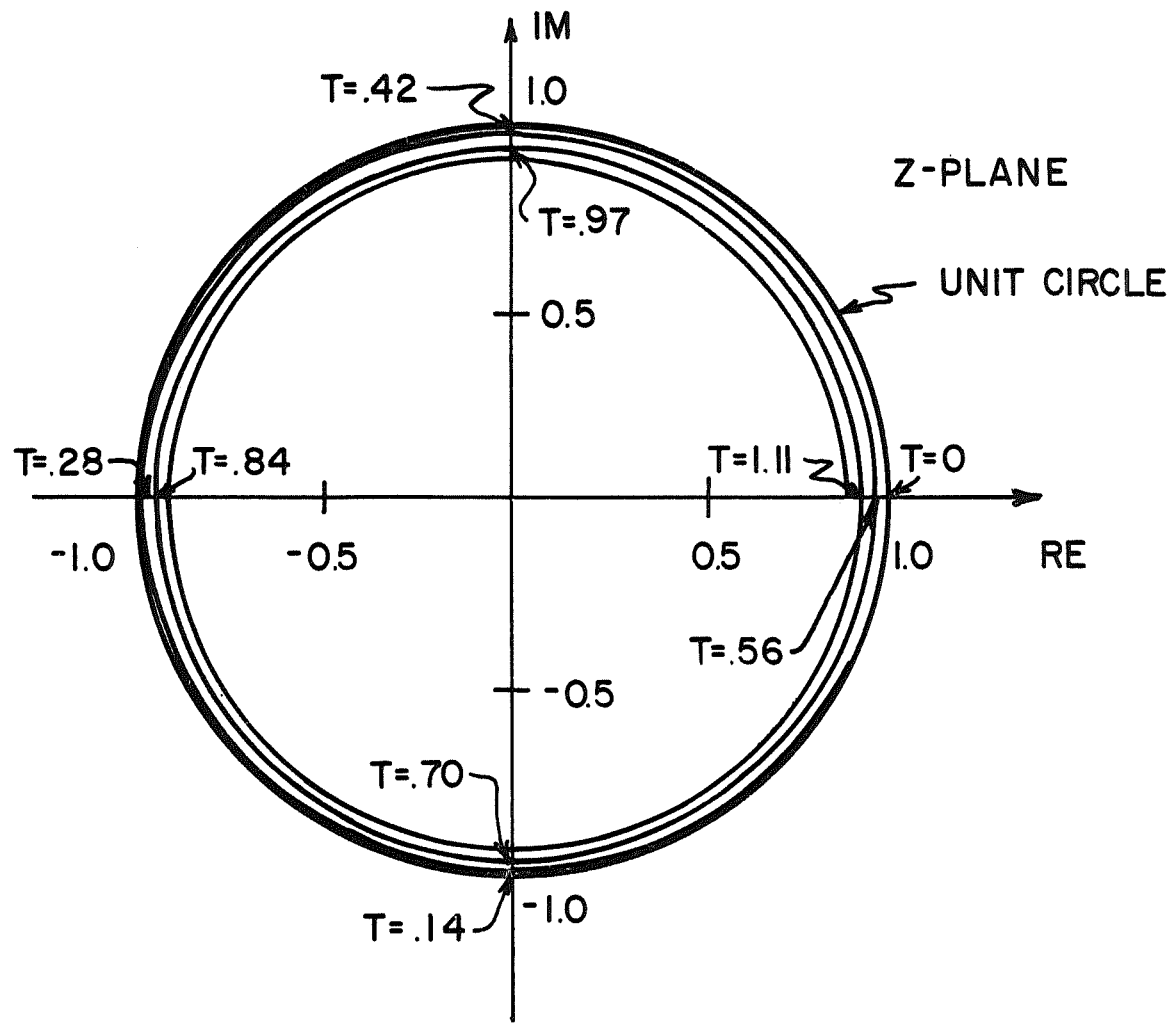


Figure 20. Movement of a Second Bending Mode Pole as the Sampling is Varied for $T_F = 120$ Sec.

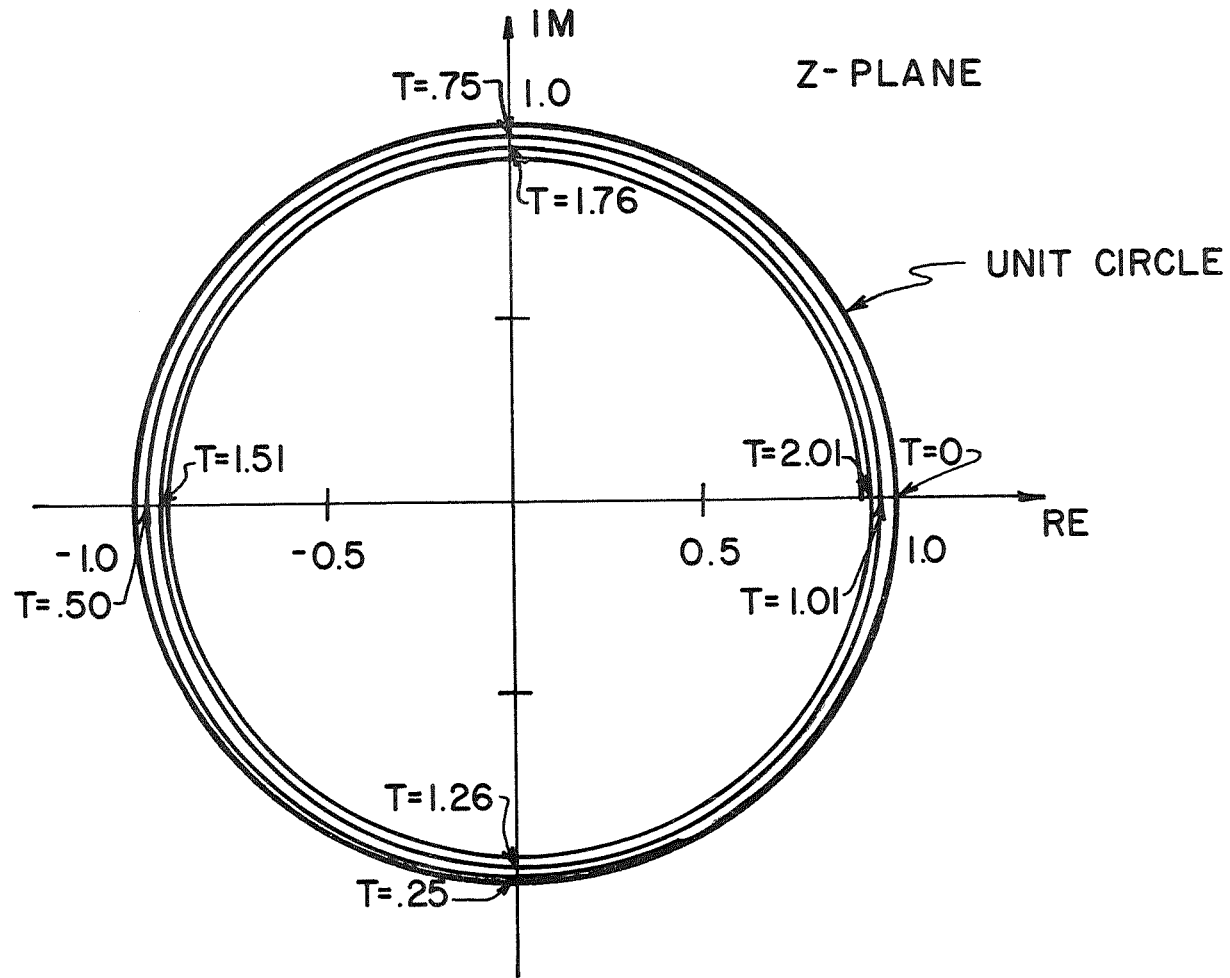


Figure 21. Plot of a First Bending Mode Pole as a Function of Sampling Period for TF = 120 Sec.

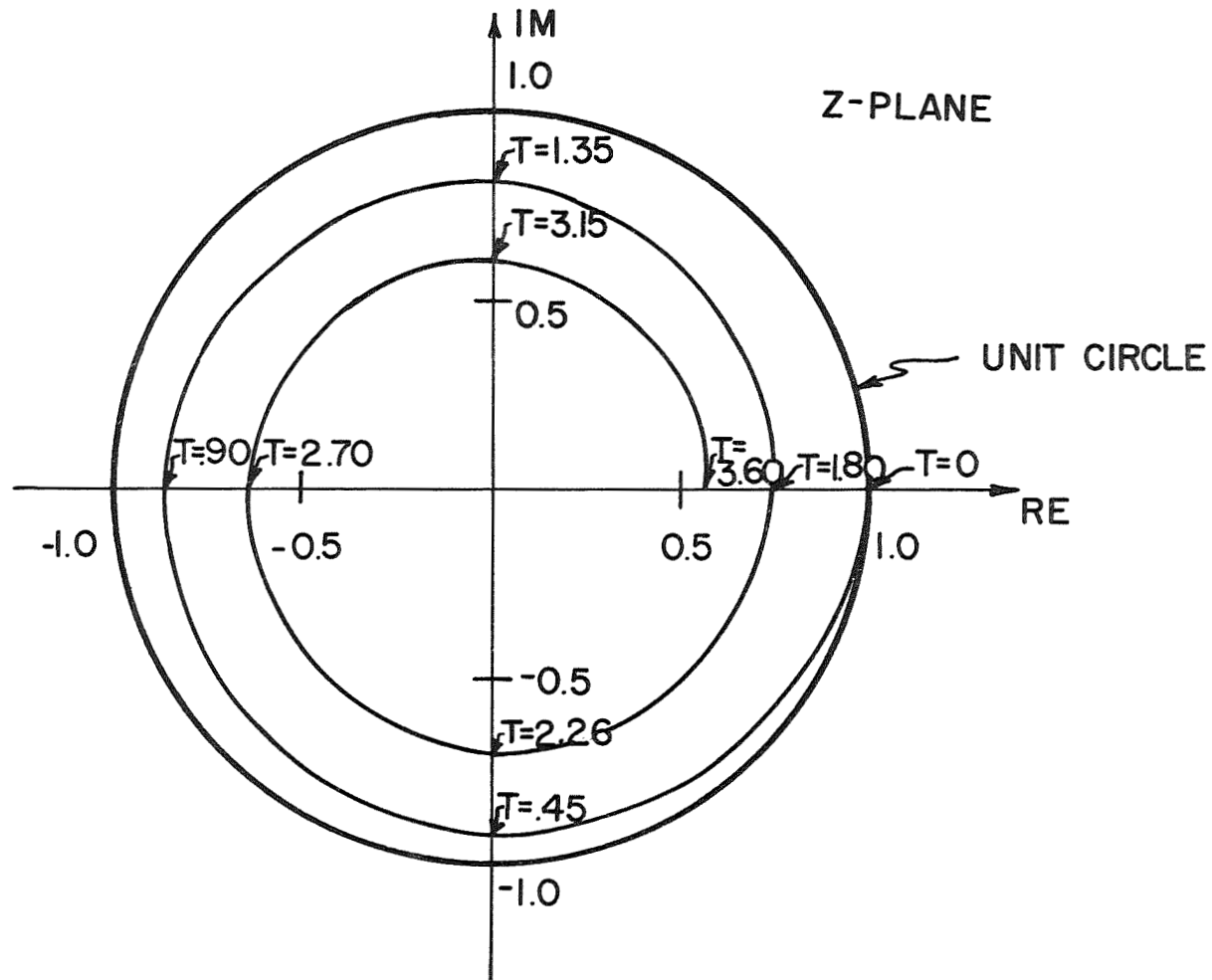


Figure 22. Locus of a First Sloss Mode Pole as a Function Sampling Period for $TF = 120$ Sec.

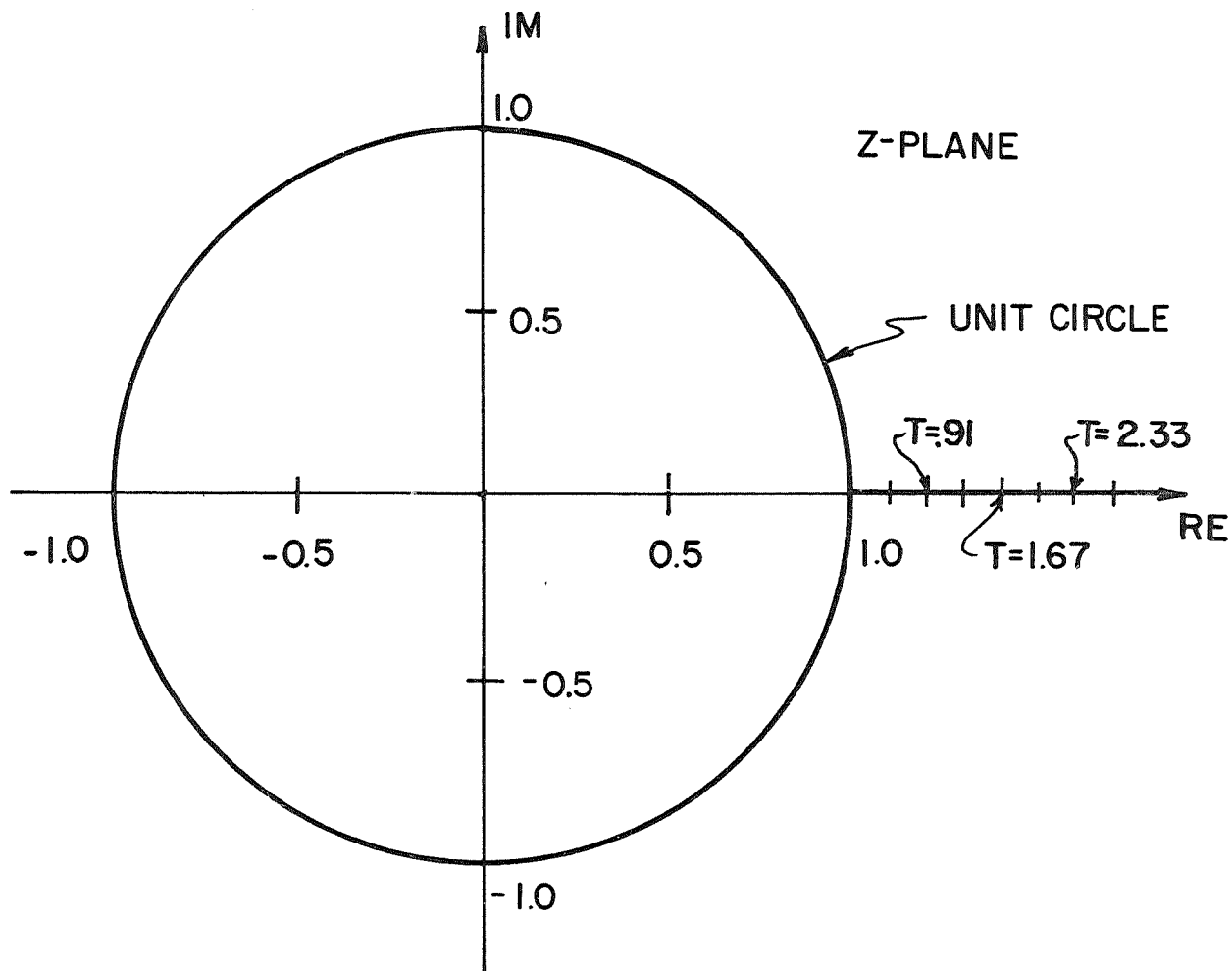


Figure 23. Movement of a Drift Pole as a Function Sampling Period for $TF = 120$ Sec.

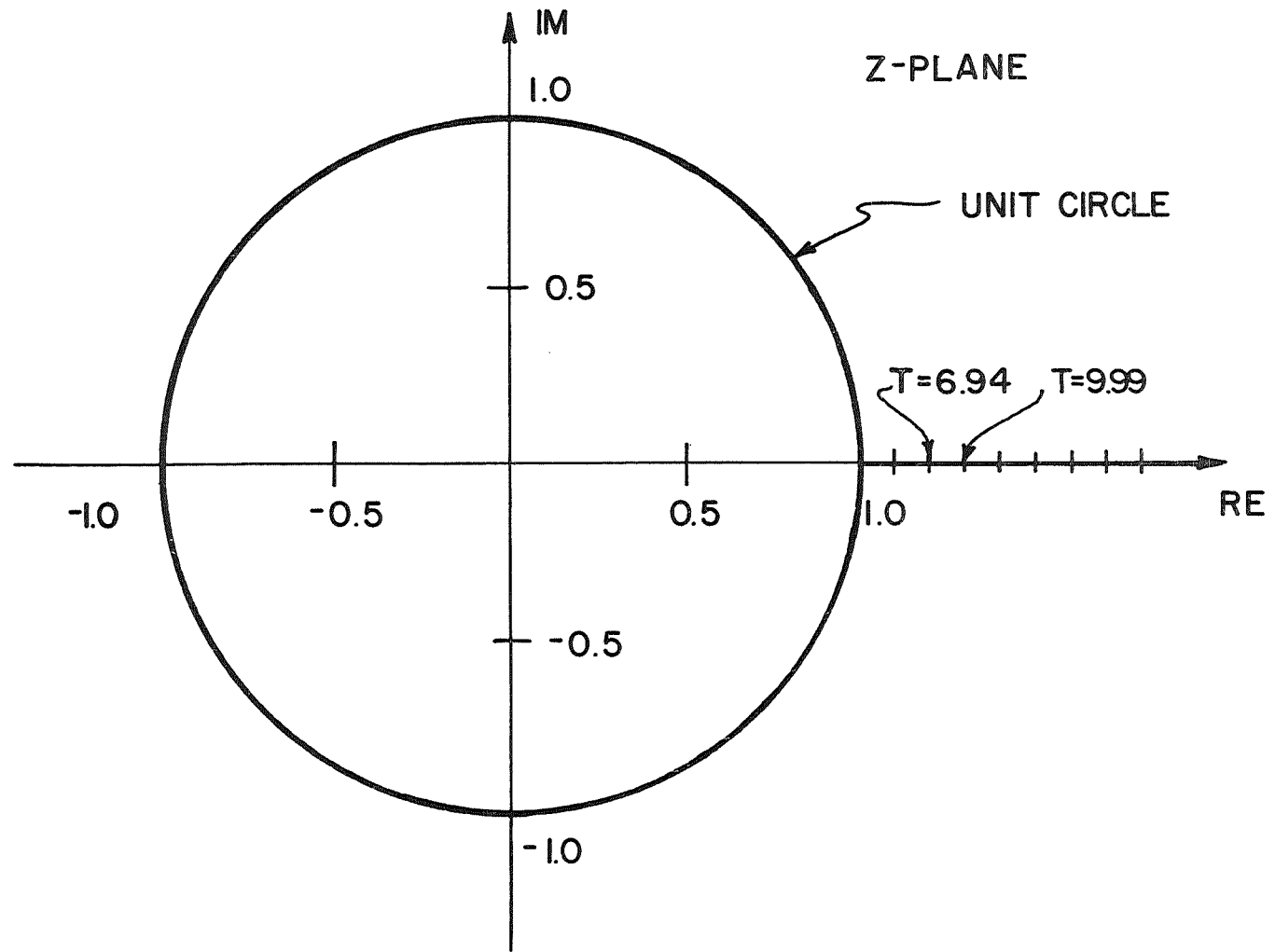


Figure 24. Movement of a Drift Pole as a Function Sampling Period for $TF = 120$ Sec.

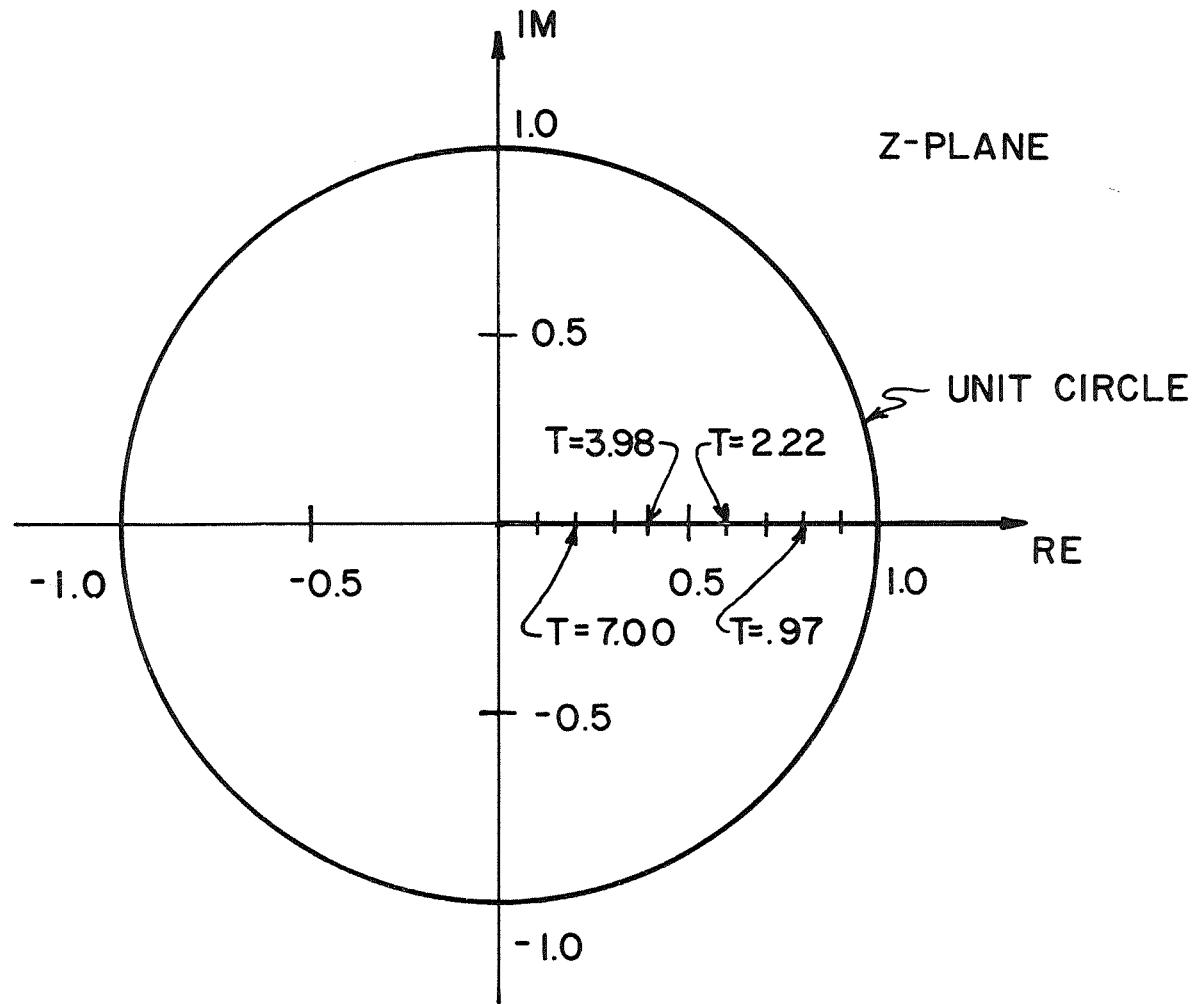


Figure 25. Plot of a Rigid Body Pole as a Function Sampling Period for $T_F = 120$ Sec.

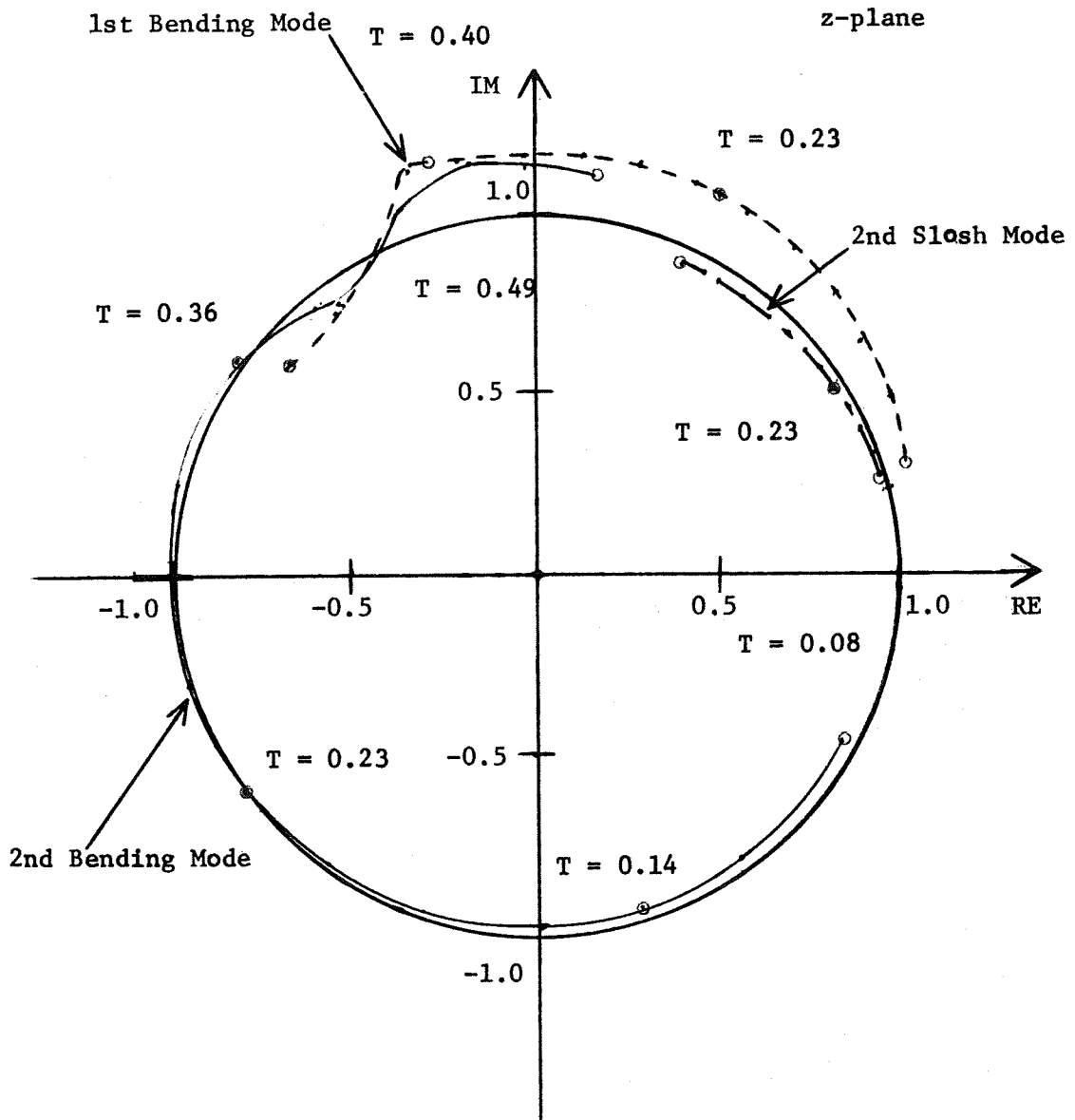


Figure 26. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for $T_F = 20$ Seconds and for an Open Loop Gain of 1.0

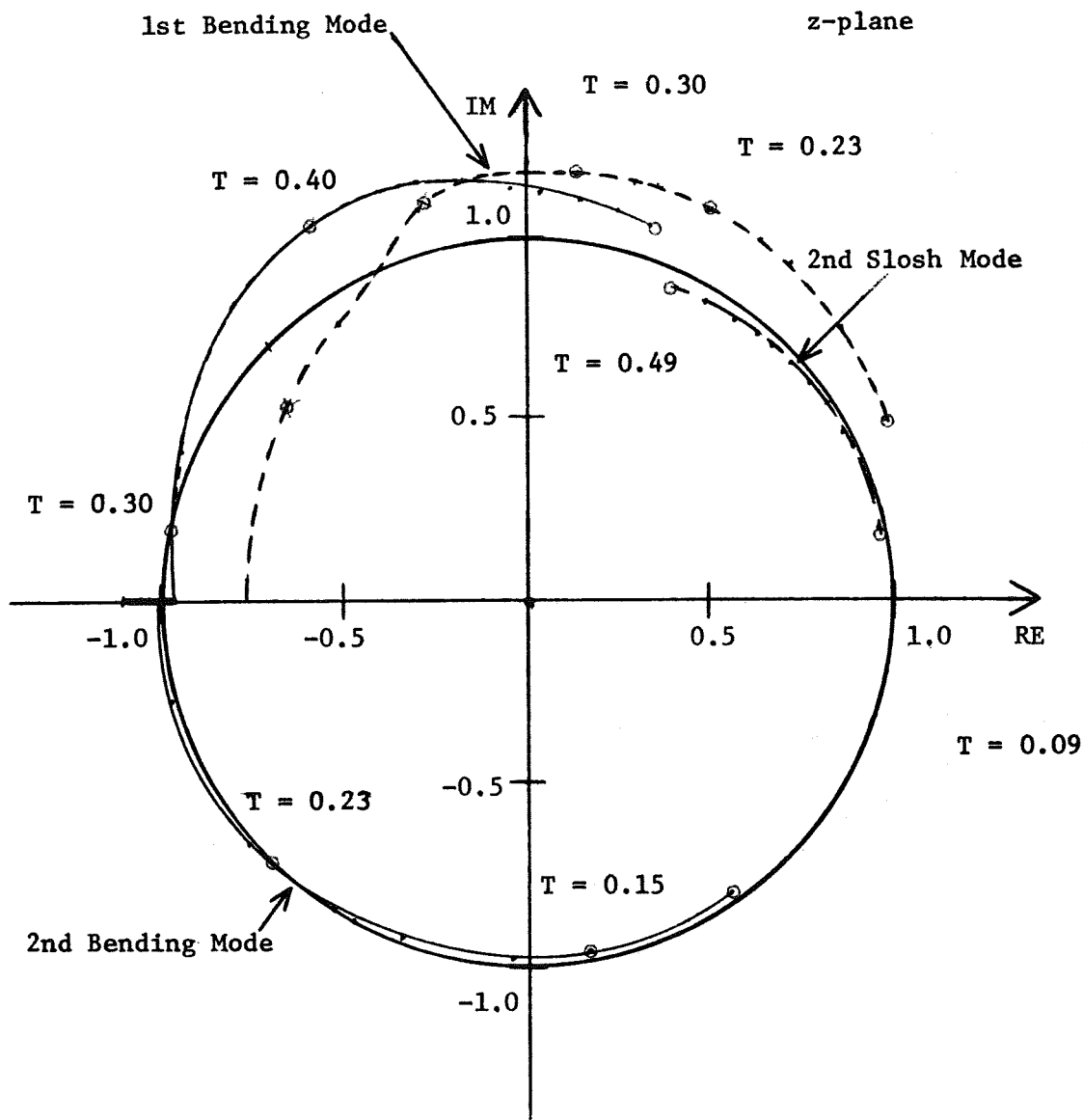


Figure 28. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for $TF = 40$ Seconds and for an Open Loop Gain of 1.0

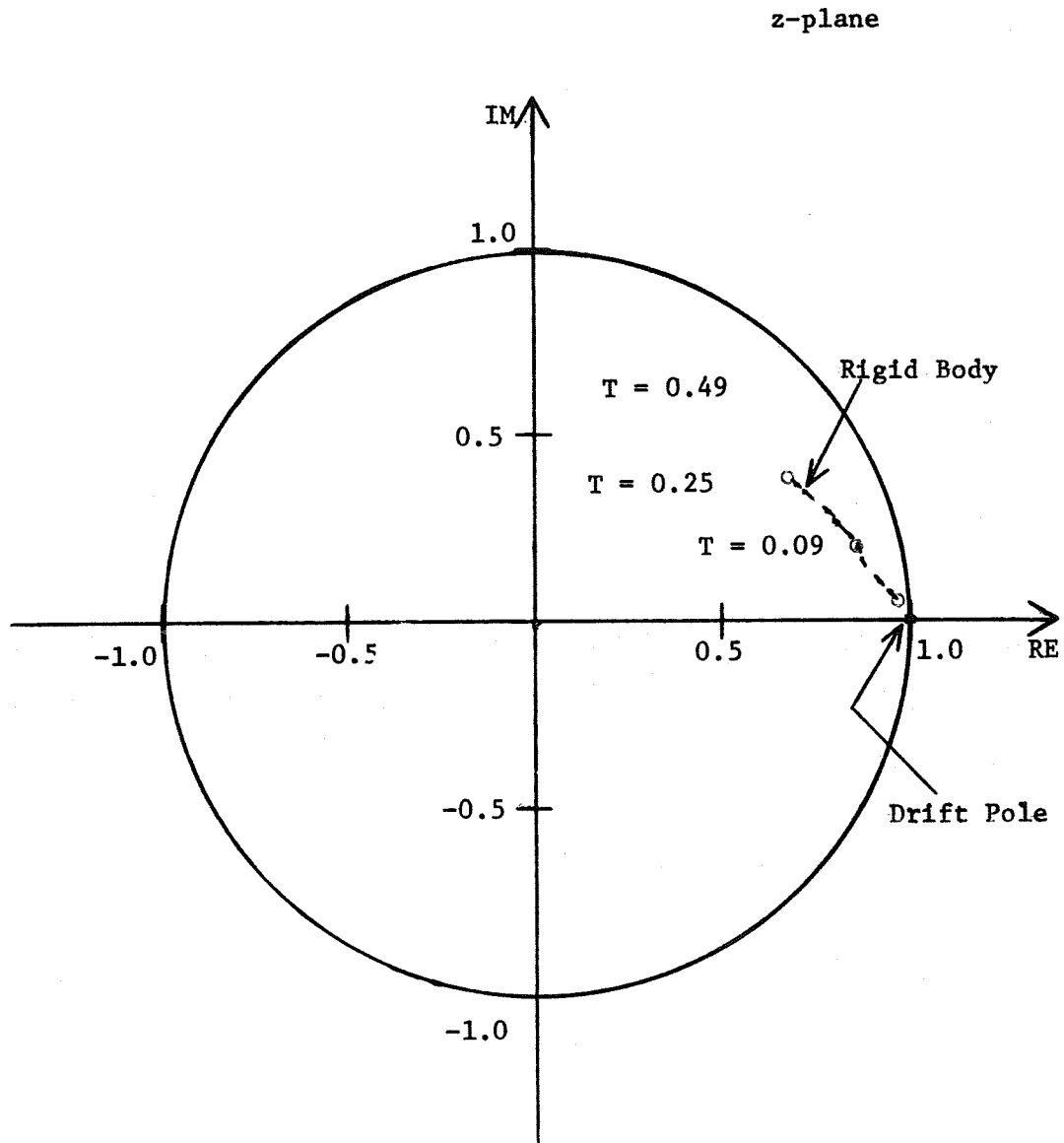


Figure 29. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 40 Seconds and for an Open Loop Gain of 1.0

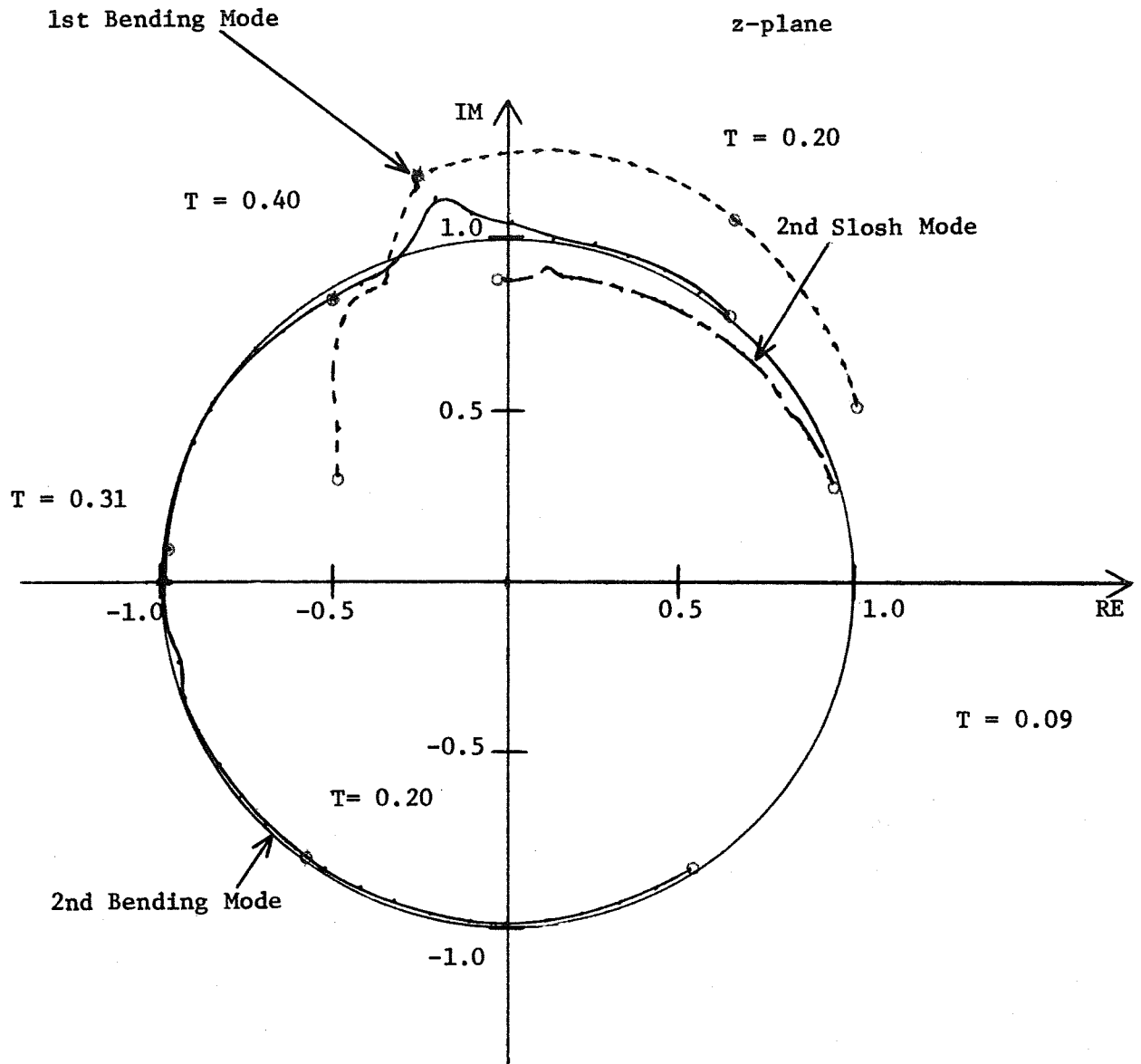


Figure 30. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling Period for TF = 60 Seconds and for an Open Loop Gain of 1.0

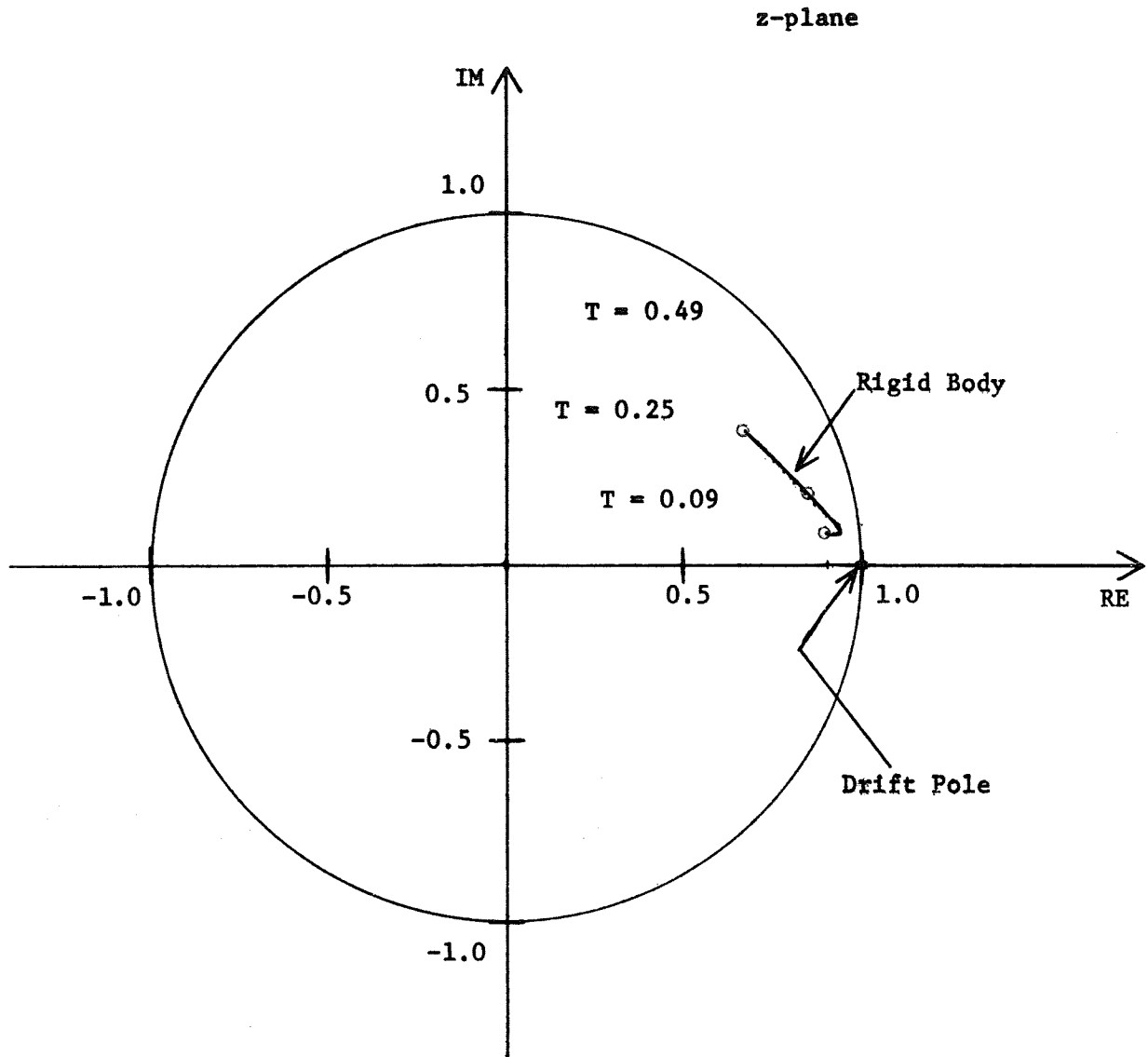


Figure 31. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for TF = 60 Seconds and for an Open Loop Gain of 1.0

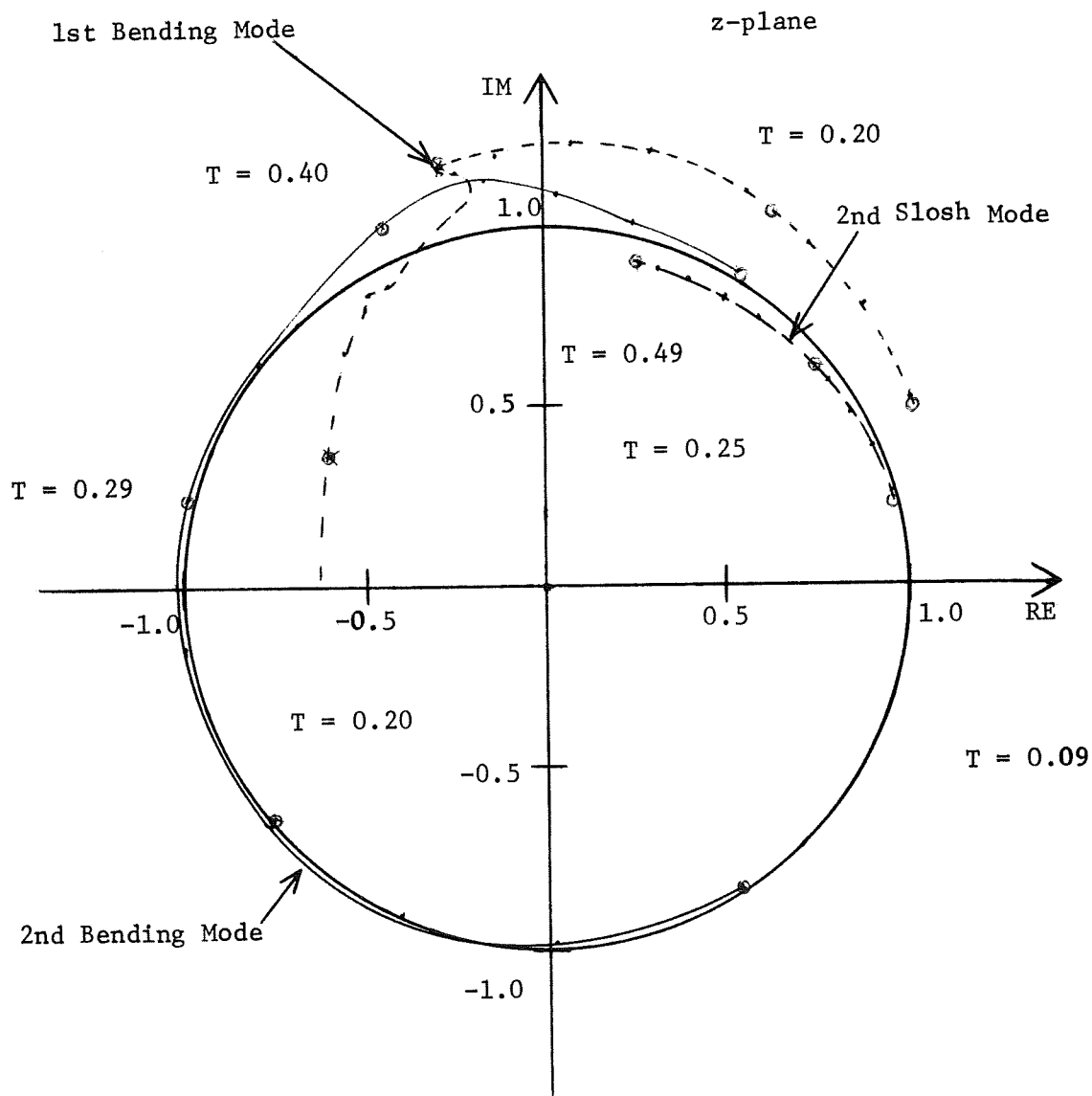


Figure 32. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling period for $TF = 80.11$ Seconds and for an Open Loop Gain of 1.0

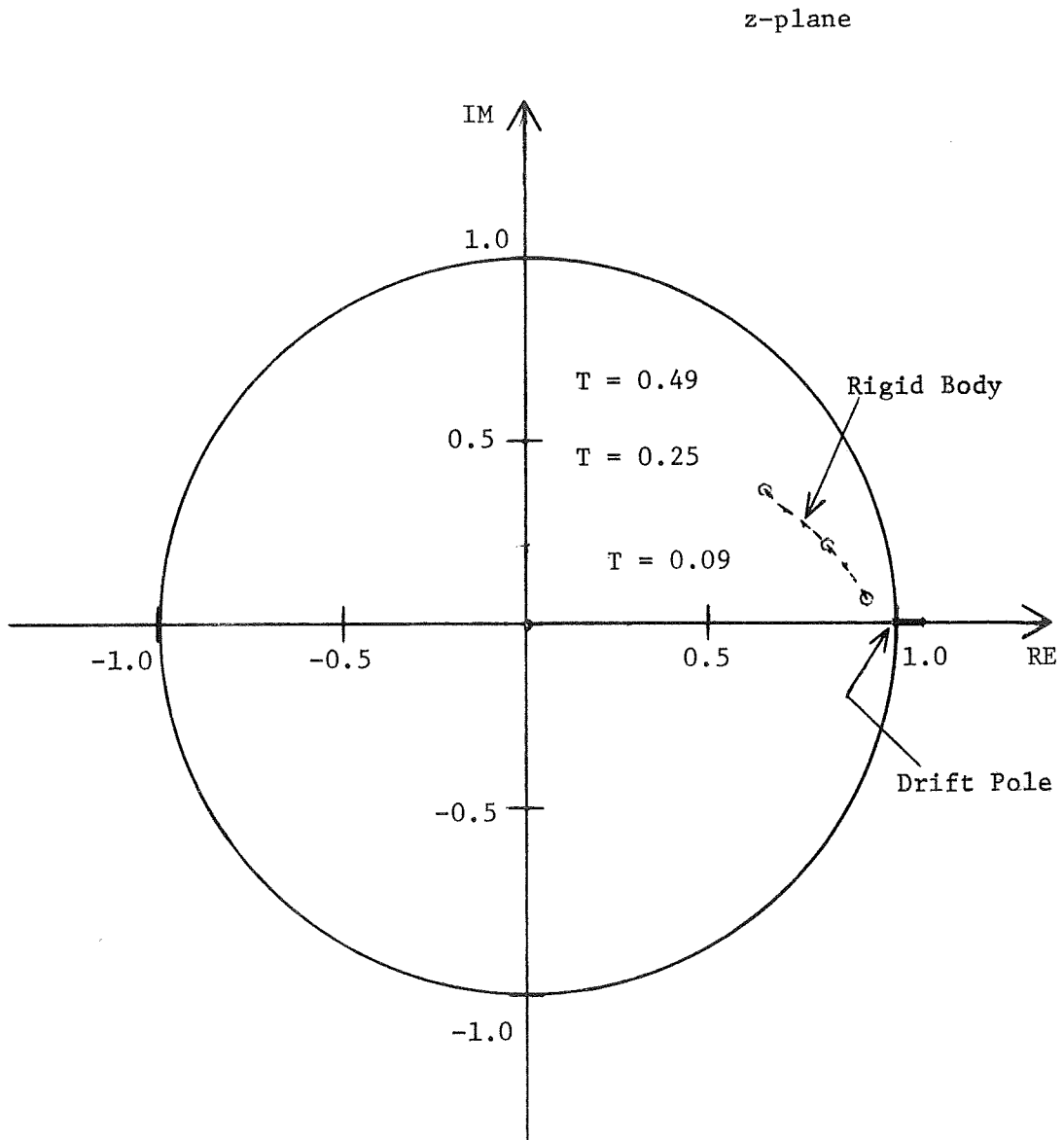


Figure 33. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for $TF = 80.11$ Seconds and for an Open Loop Gain of 1.0

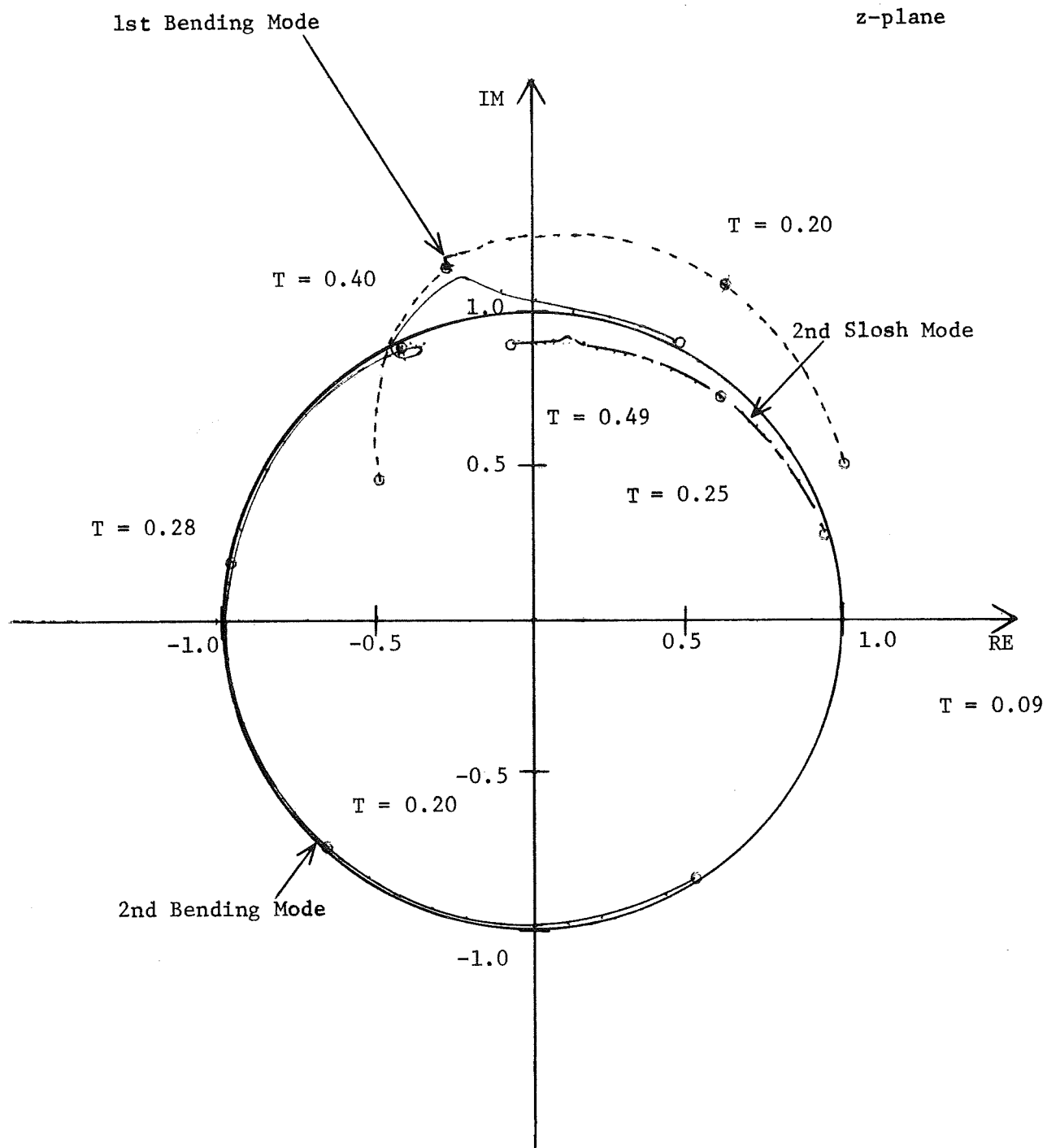


Figure 34. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling period for $T_F = 100$ Seconds and for an Open Loop Gain of 1.0

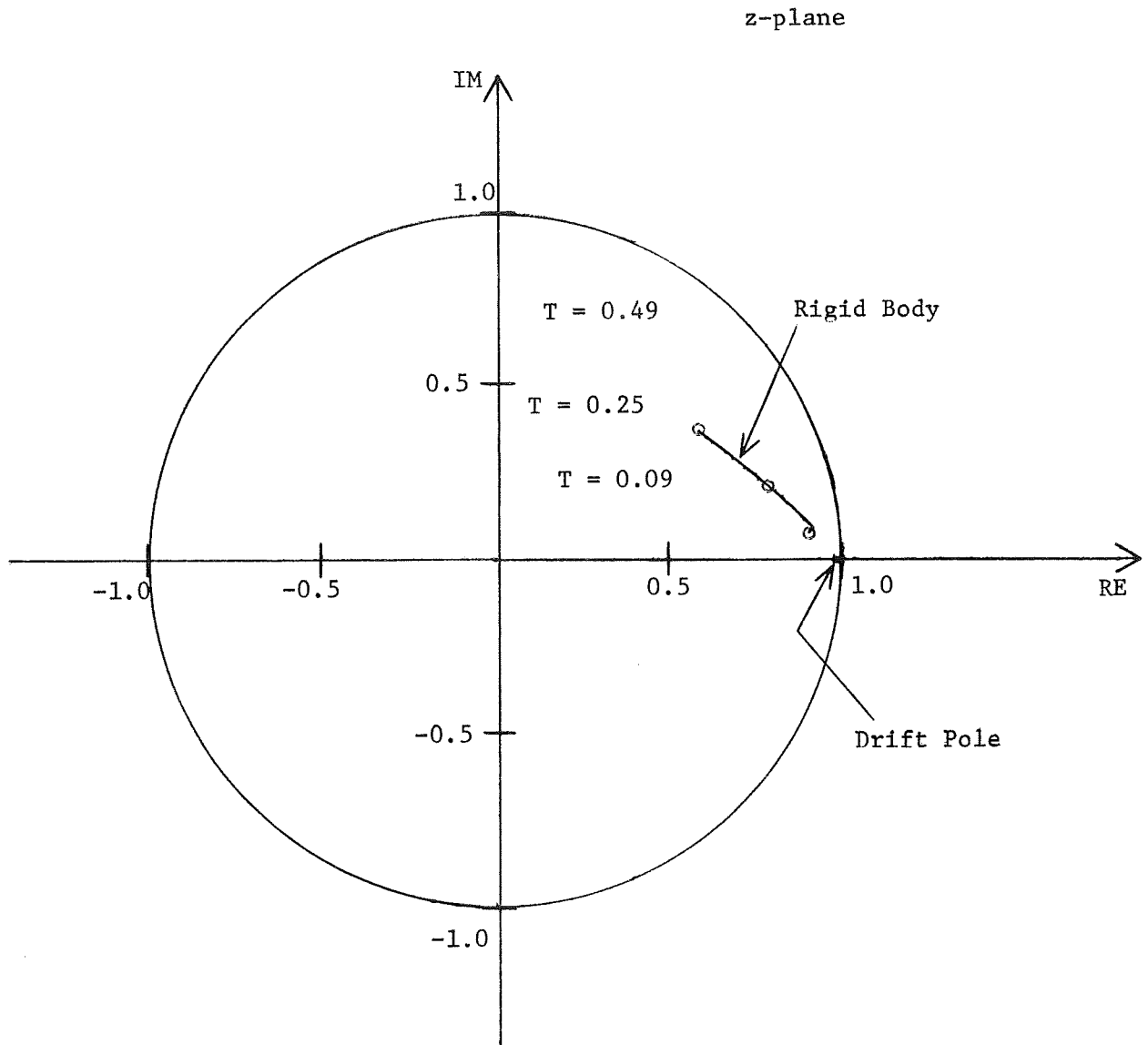


Figure 35. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for $TF = 100$ Seconds and for an Open Loop Gain of 1.0

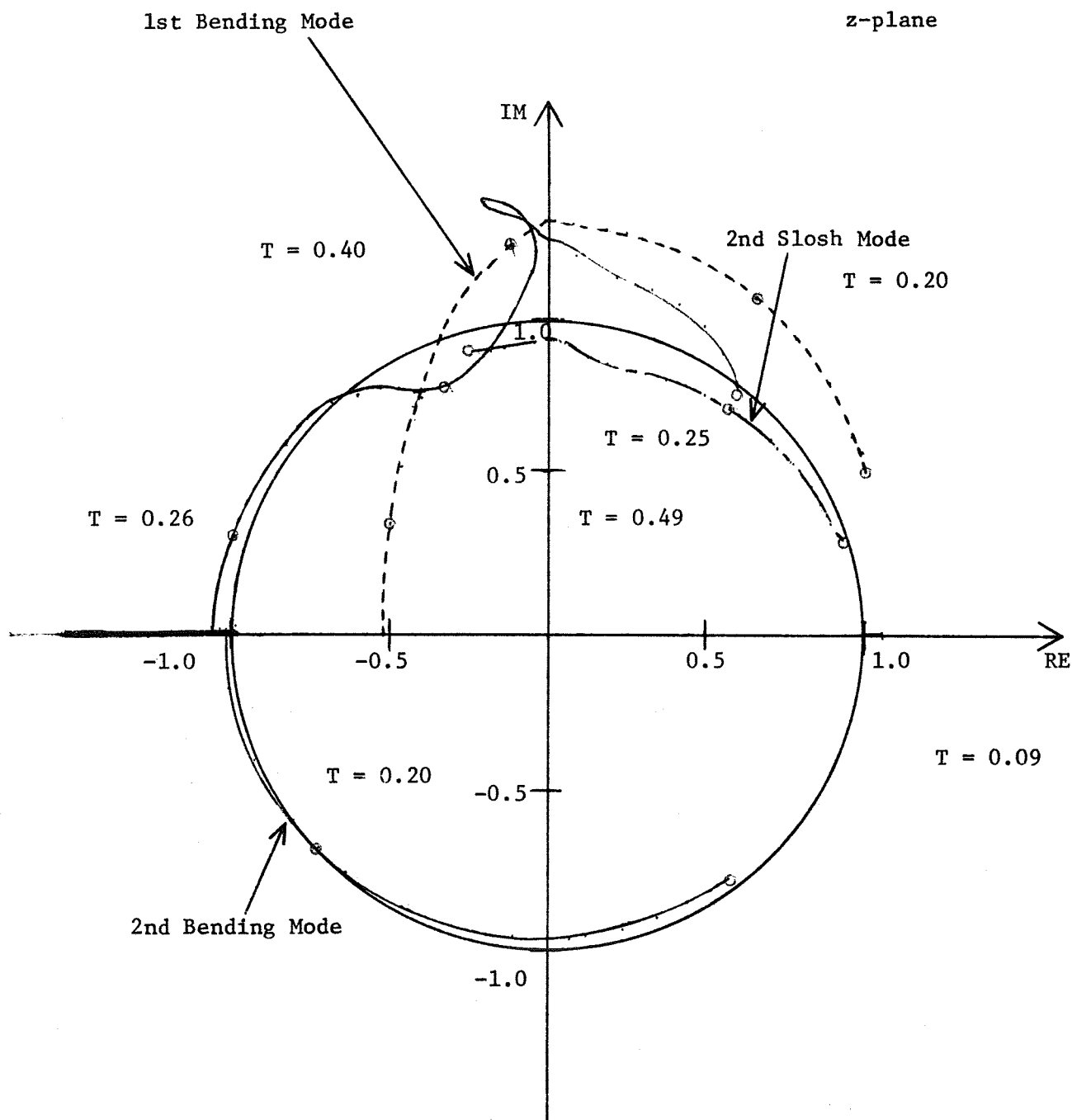


Figure 36. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling period for $TF = 120$ Seconds and for an Open Loop Gain of 1.0

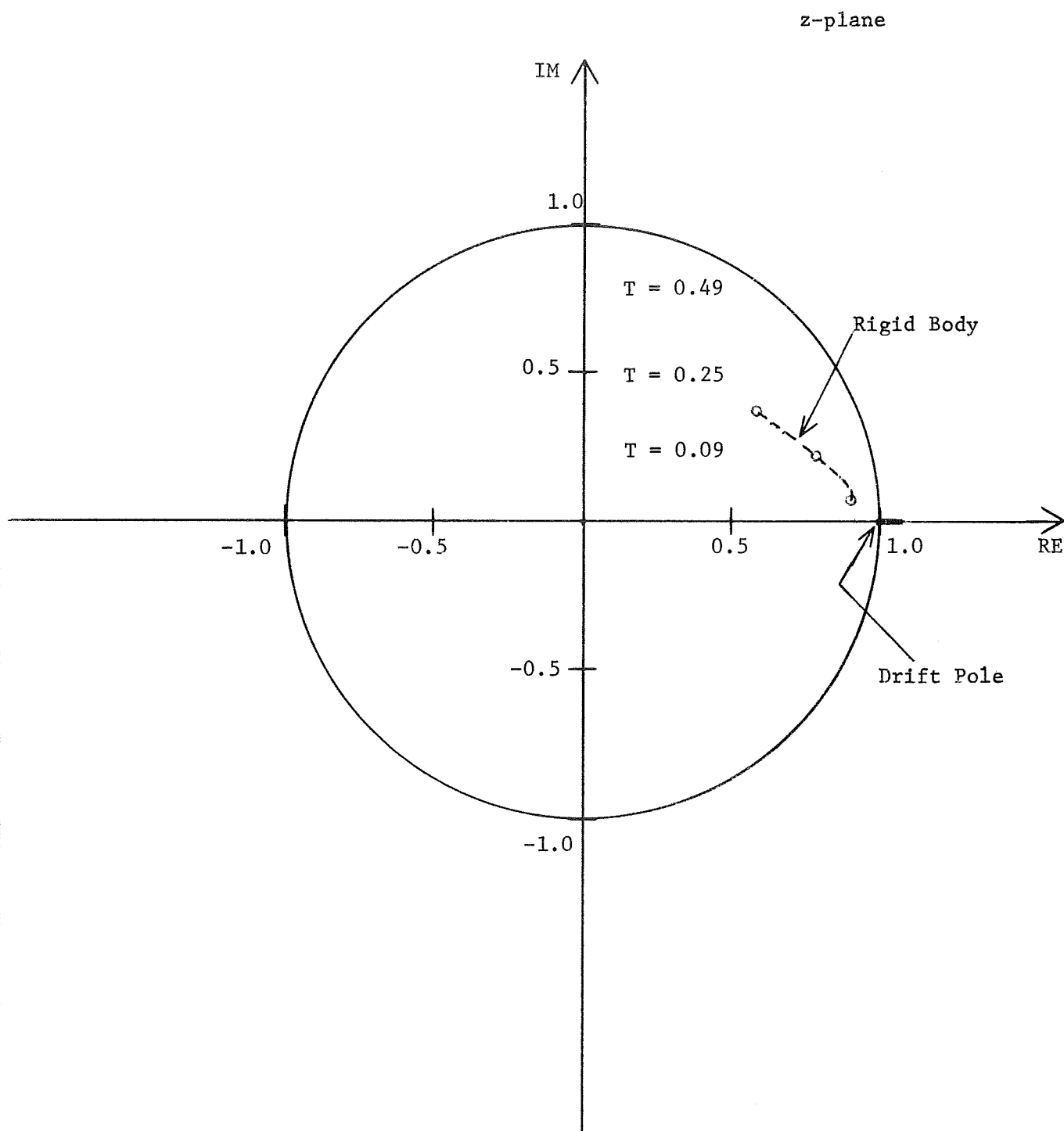


Figure 37. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for $T_F = 120$ Seconds and for an Open Loop Gain of 1.0

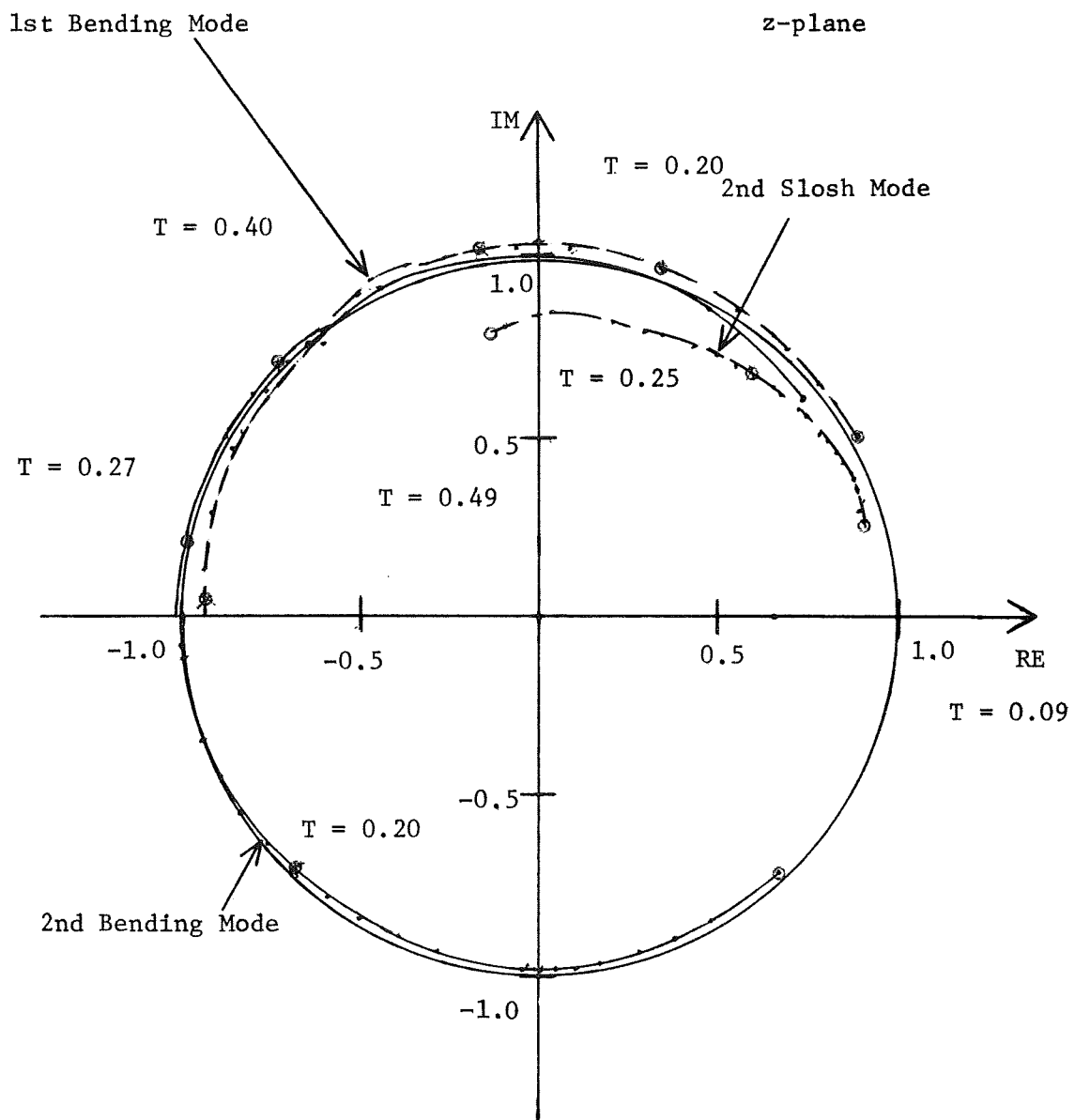


Figure 38. Closed Loop z-plane Loci of the First and Second Bending Modes and Slosh Mode Poles as a Function of Sampling period for TF = 140 Seconds and for an Open Loop Gain of 1.0

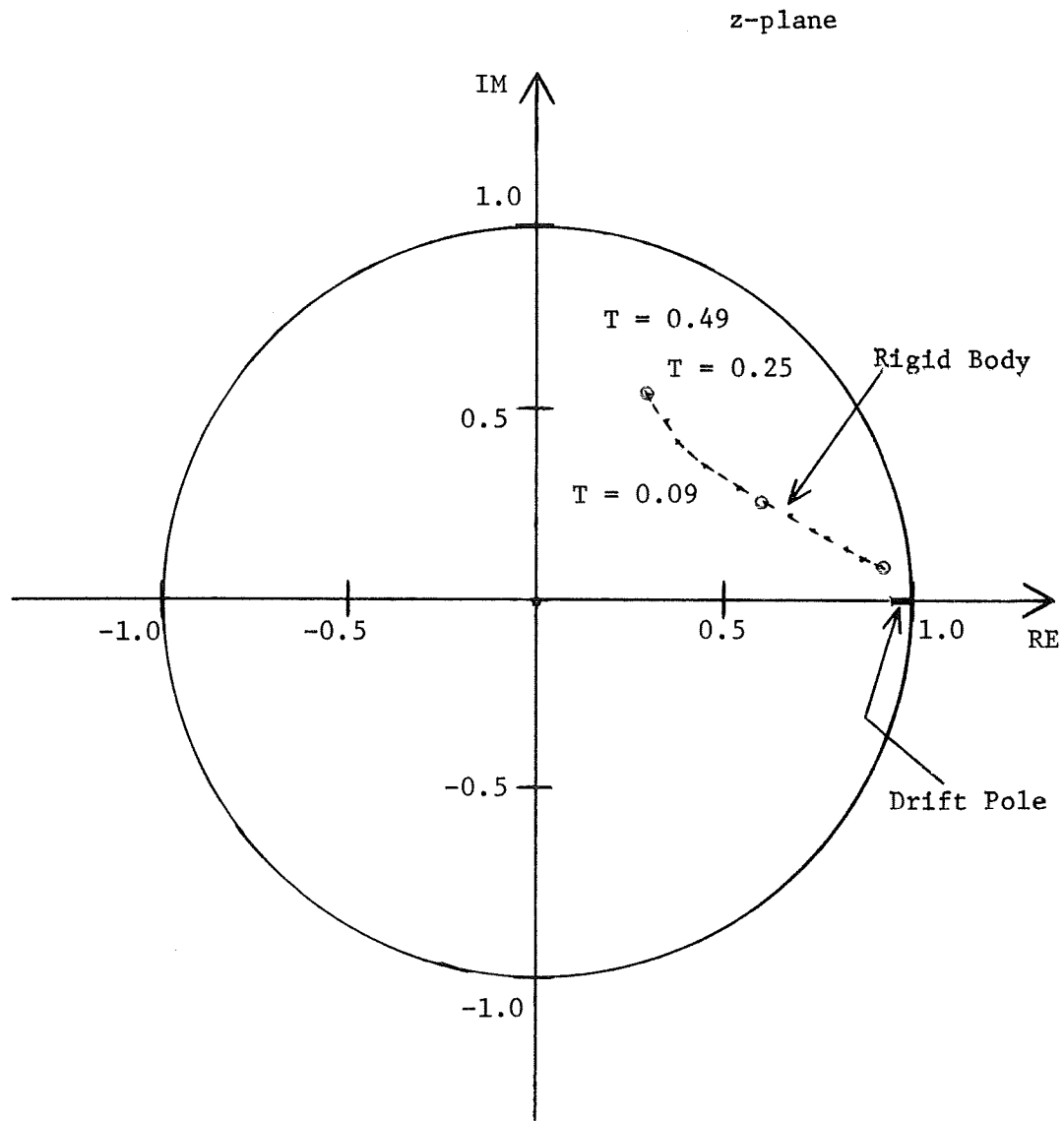


Figure 39. Closed Loop z-plane Loci of the Drift and Rigid Body Poles as a Function of Sampling Period for $TF = 140$ Seconds and for an Open Loop Gain of 1.0

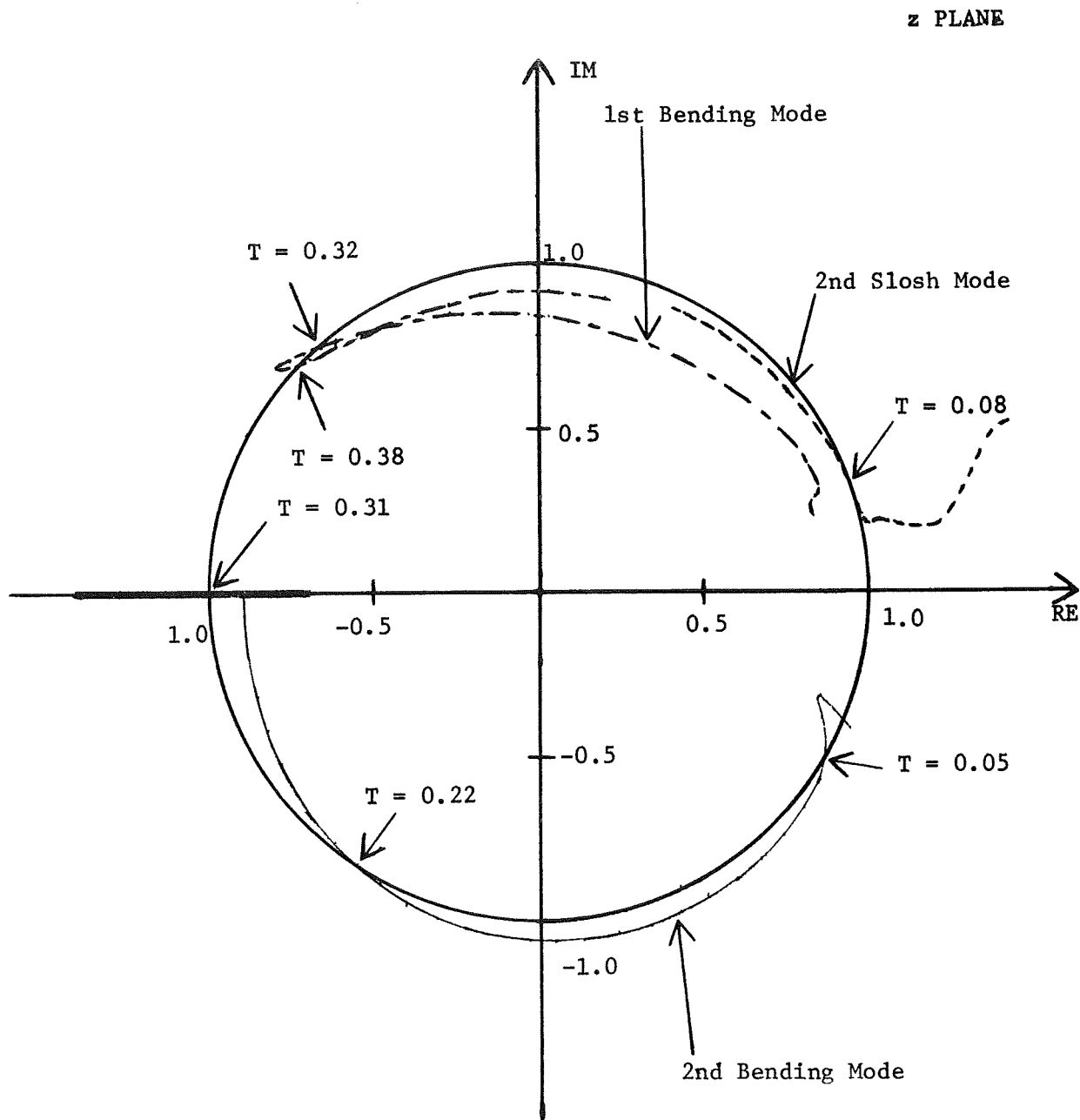


Figure 40. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 20$ Seconds and Open Loop Gain = -1.0

z - PLANE

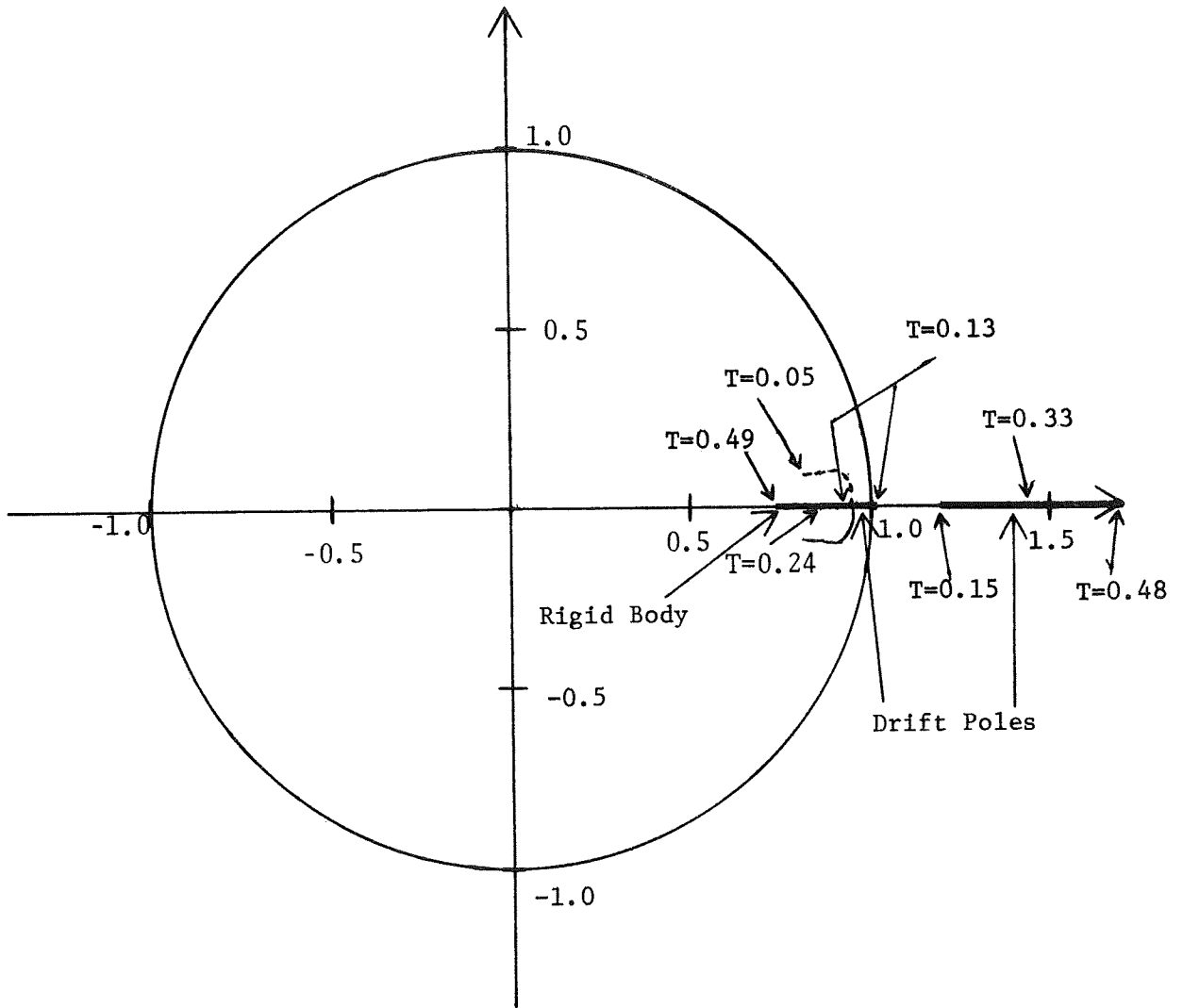


Figure 41. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $T_F = 20$ Seconds and Open Loop Gain = -1.0

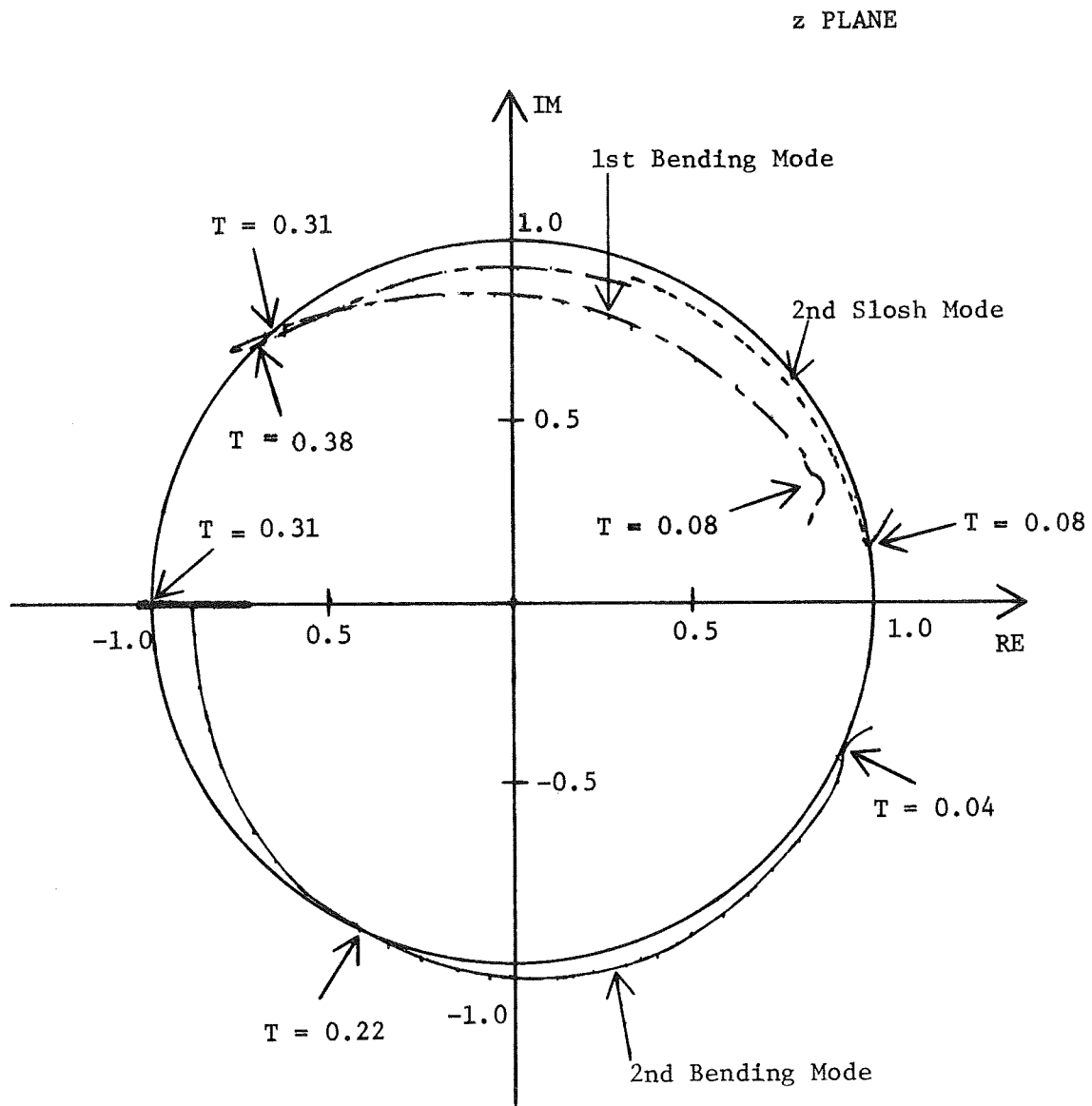


Figure 42. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $T_F = 40$ Seconds and Open Loop Gain = -1.0

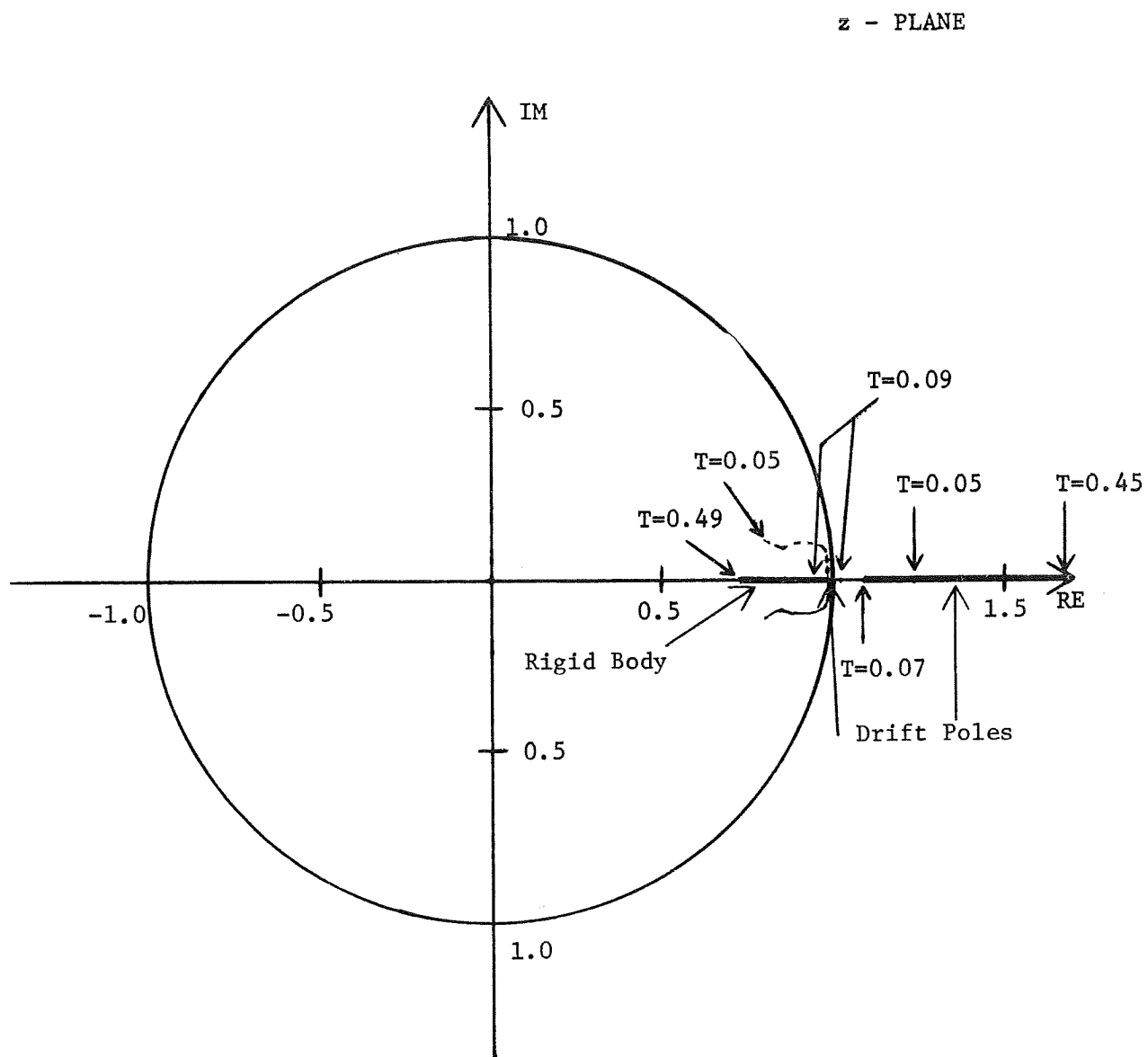


Figure 43. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $T_F = 40$ Seconds and Open Loop Gain = -1.0

z PLANE

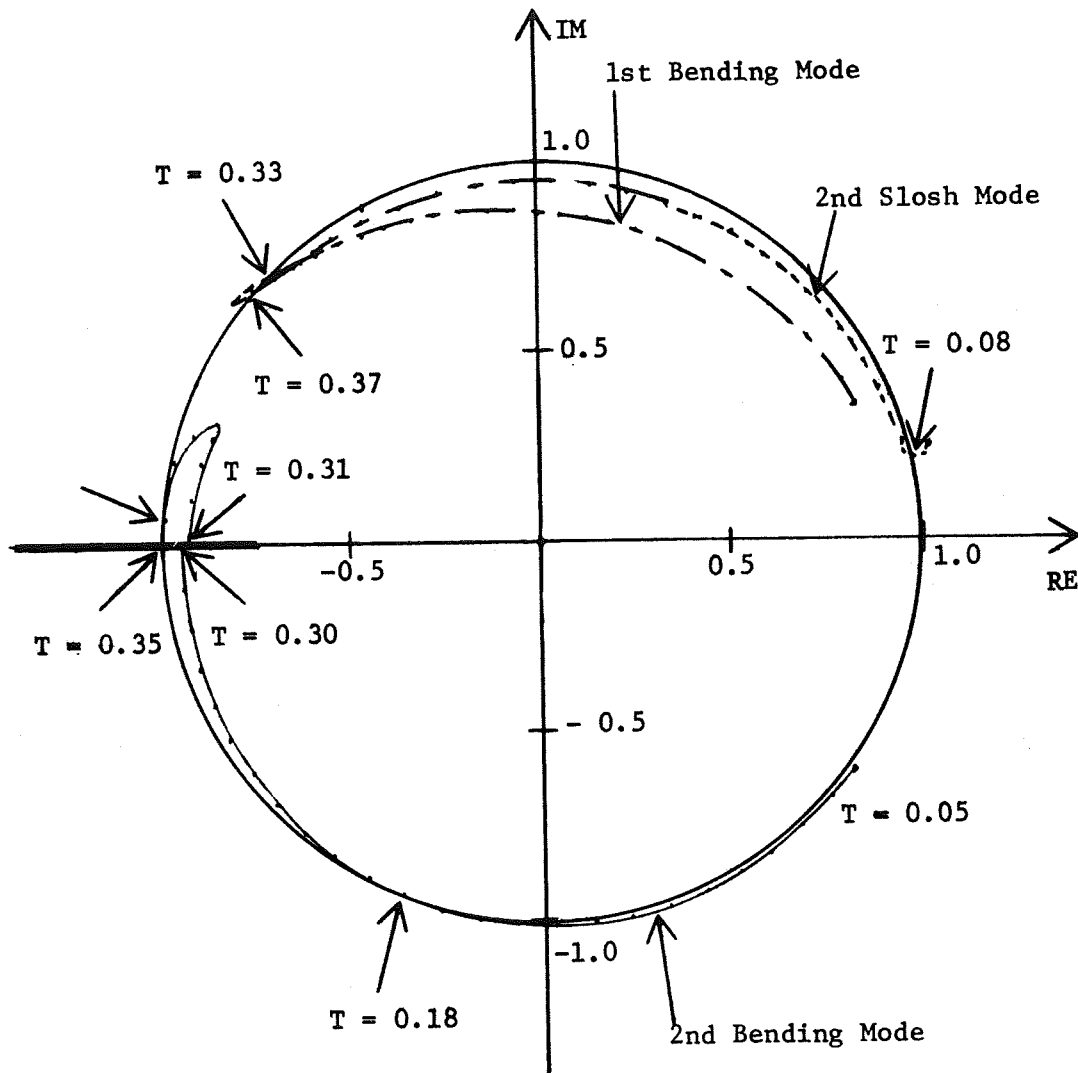


Figure 44. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 60$ Seconds and Open Loop Gain = -1.0

z - PLANE

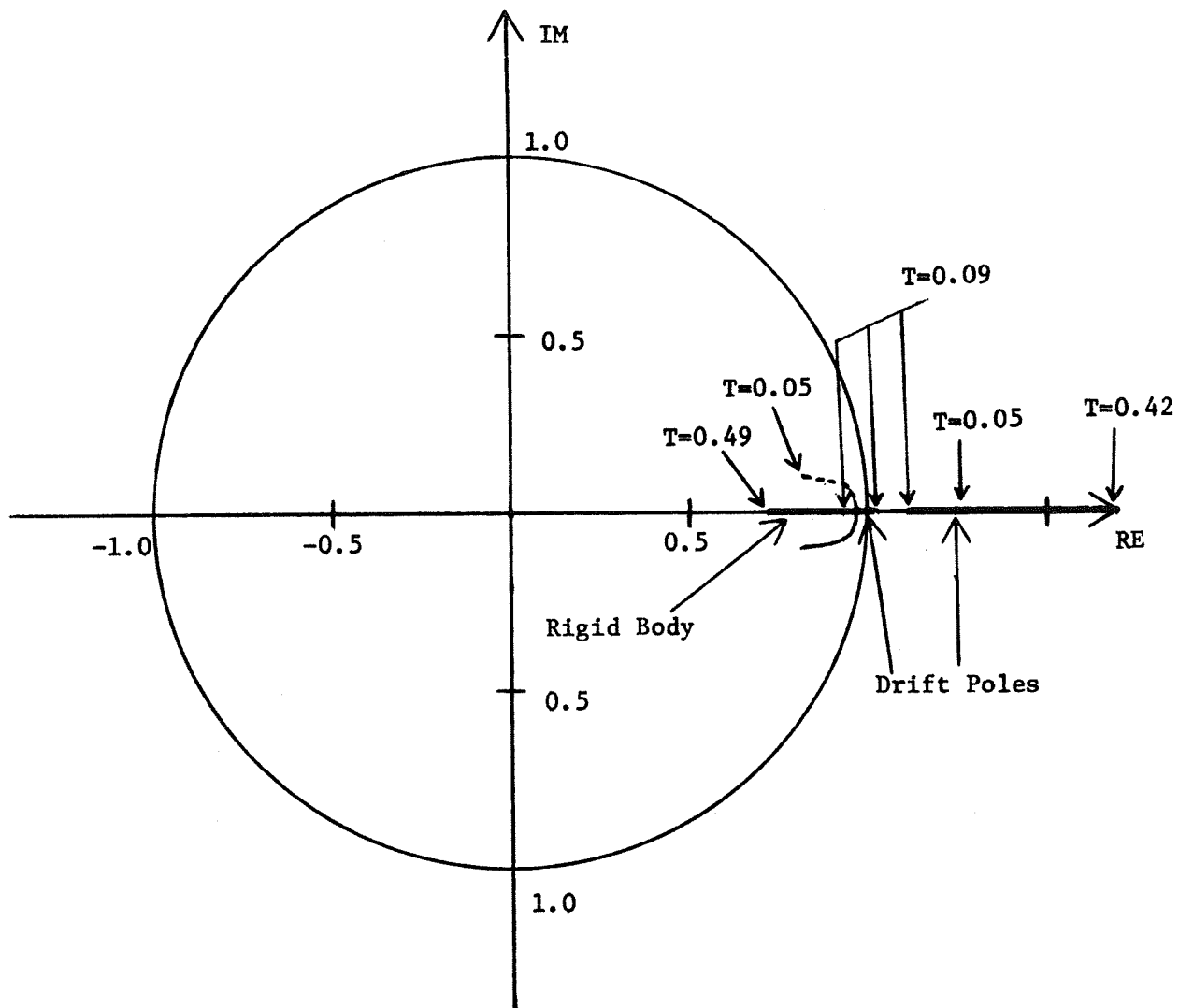


Figure 45. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $T_F = 60$ Seconds and Open Loop Gain = -1.0

z PLANE

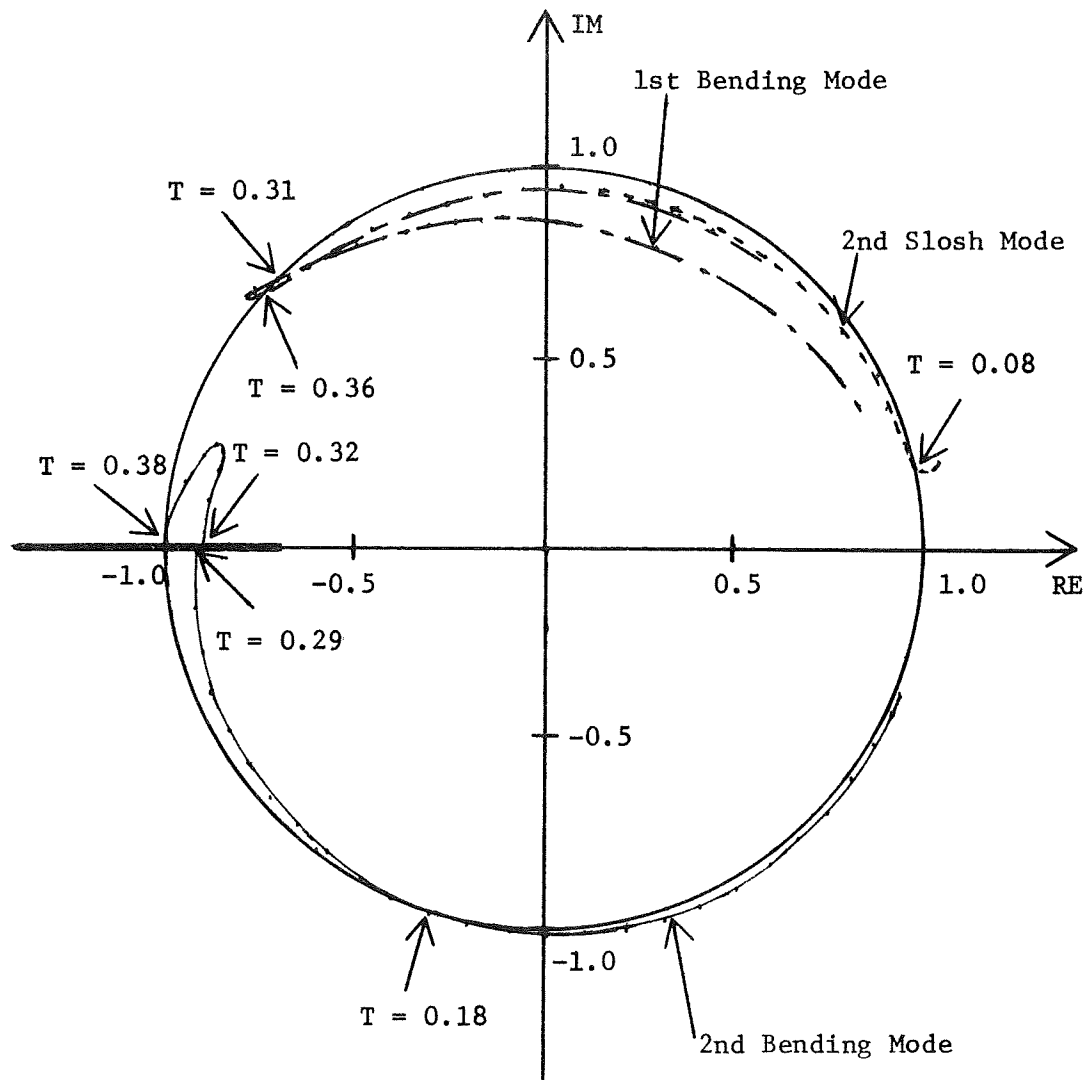


Figure 46. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 80.1$ Seconds and Open Loop Gain = -1.0

z - PLANE

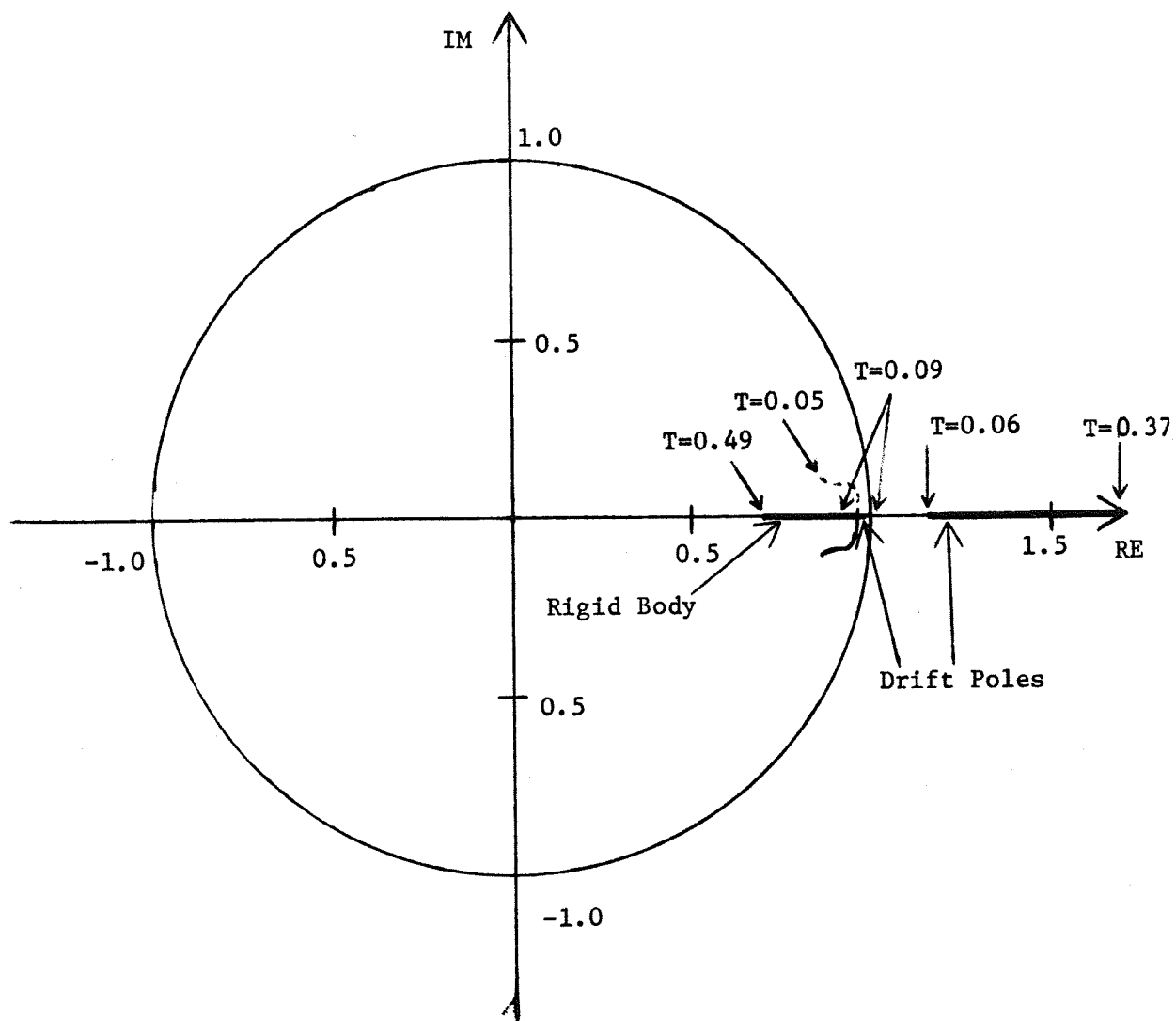


Figure 47. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $TF = 80.1$ Seconds and Open Loop Gain = -1.0

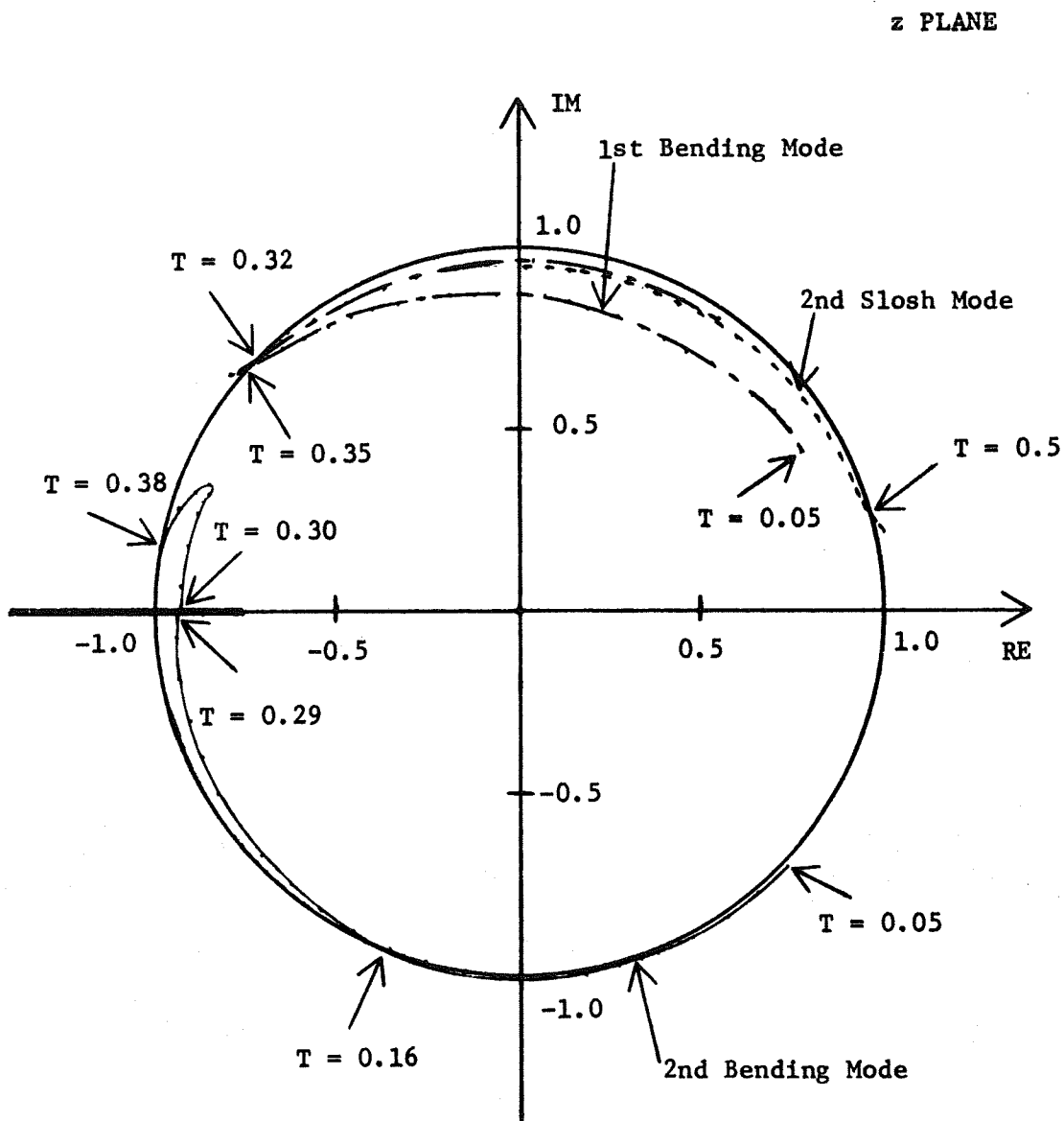


Figure 48. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 100$ Seconds and Open Loop Gain = -1.0

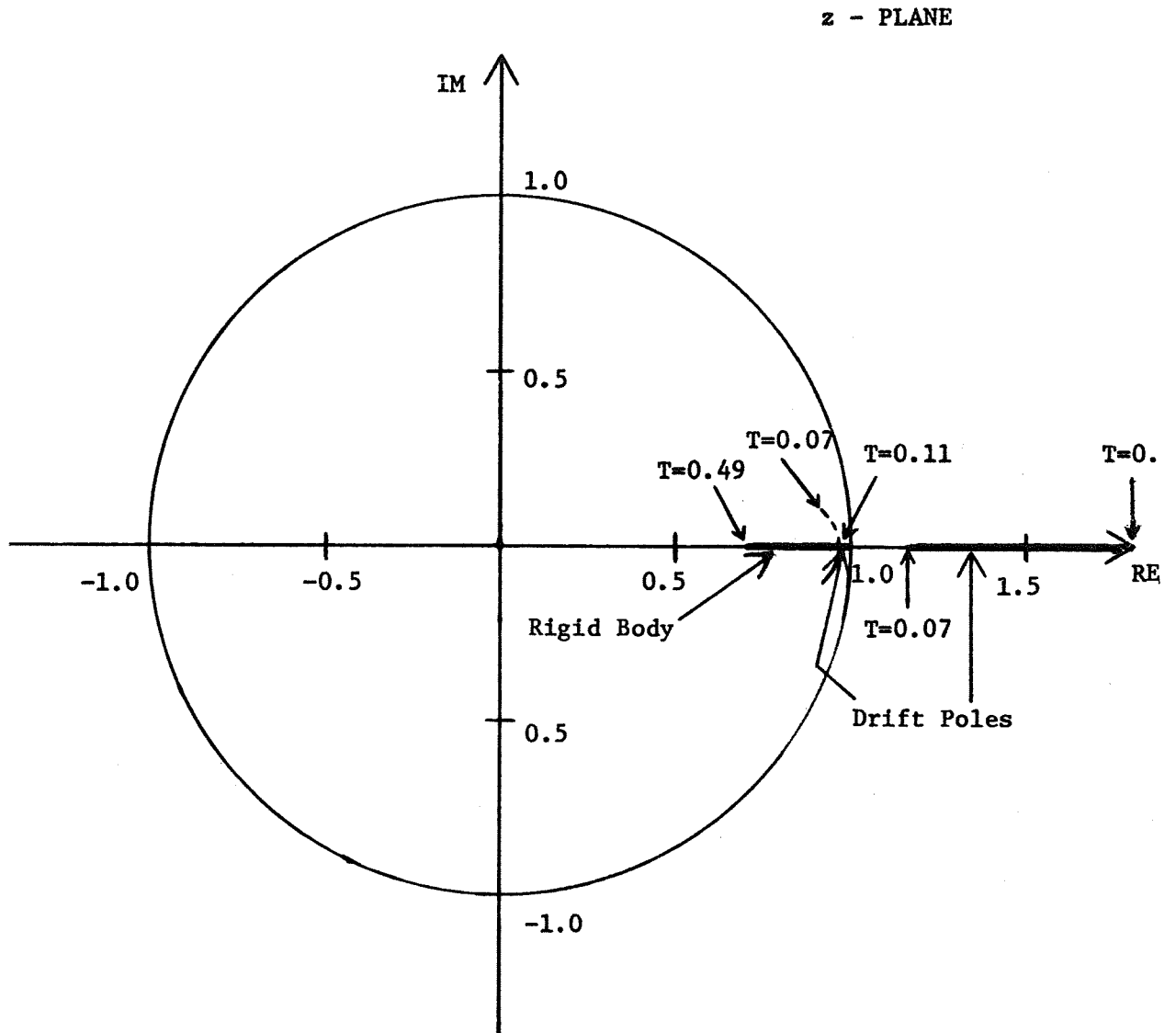


Figure 49. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When TF = 100 Seconds and Open Loop Gain = -1.0

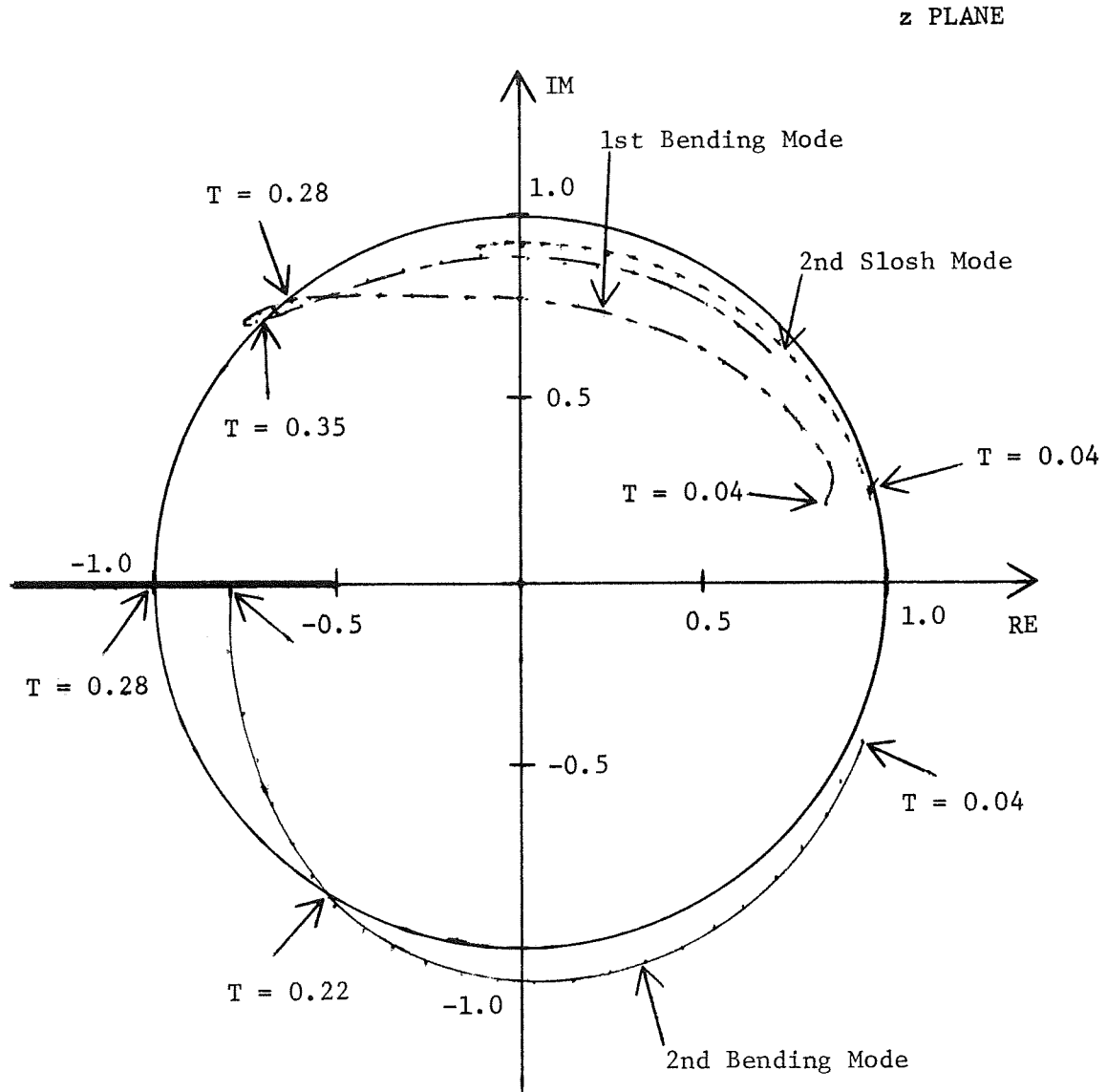


Figure 50. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 120$ Seconds and Open Loop Gain = -1.0

z - PLANE

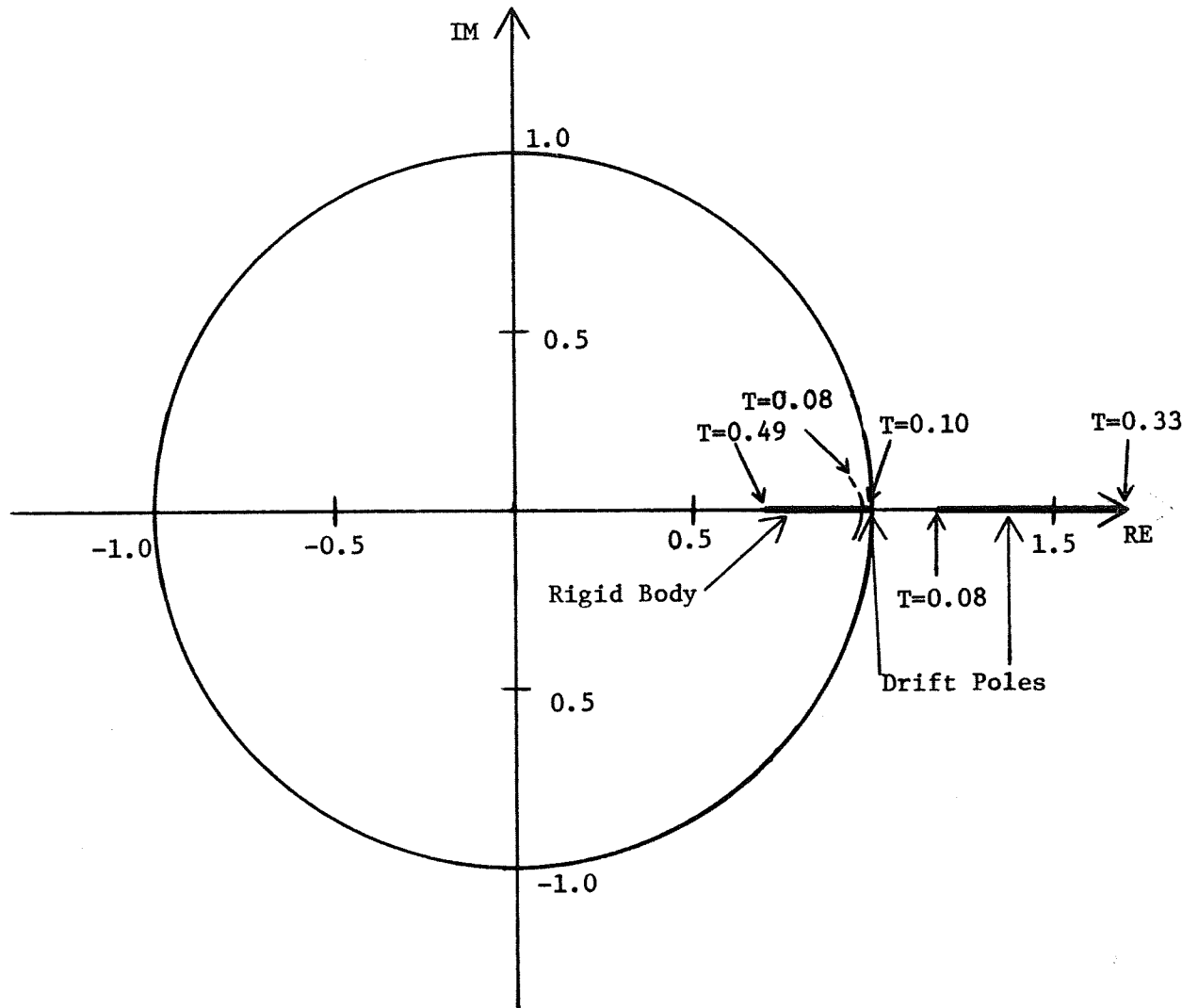


Figure 51. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $TF = 120$ Seconds and Open Loop Gain = -1.0

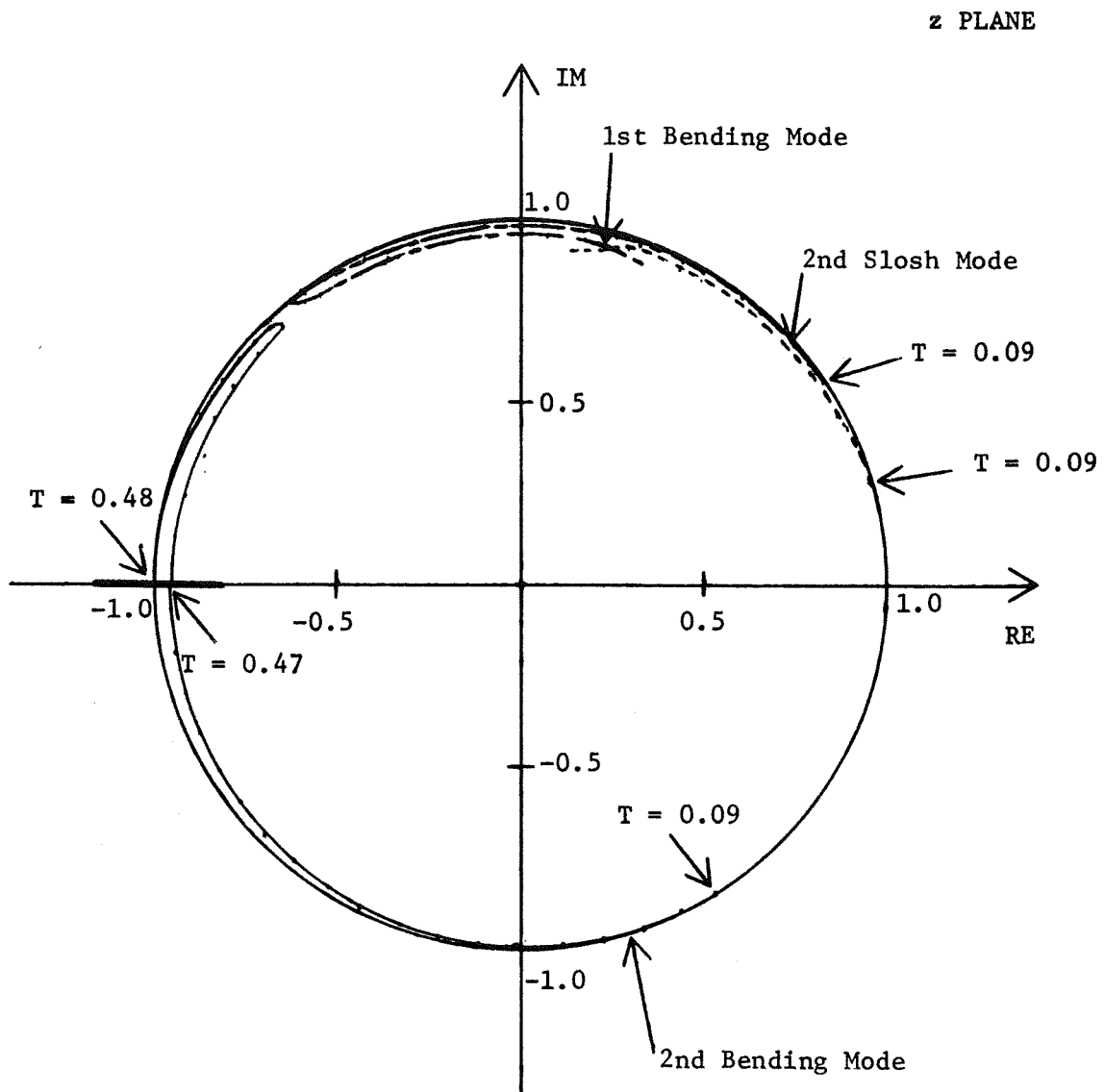


Figure 52. Plot of First Bending Mode, Second Bending Mode, and Second Slosh Mode Poles as a Function of Sampling Period When $TF = 140$ Seconds and Open Loop Gain = -1.0

z - PLANE

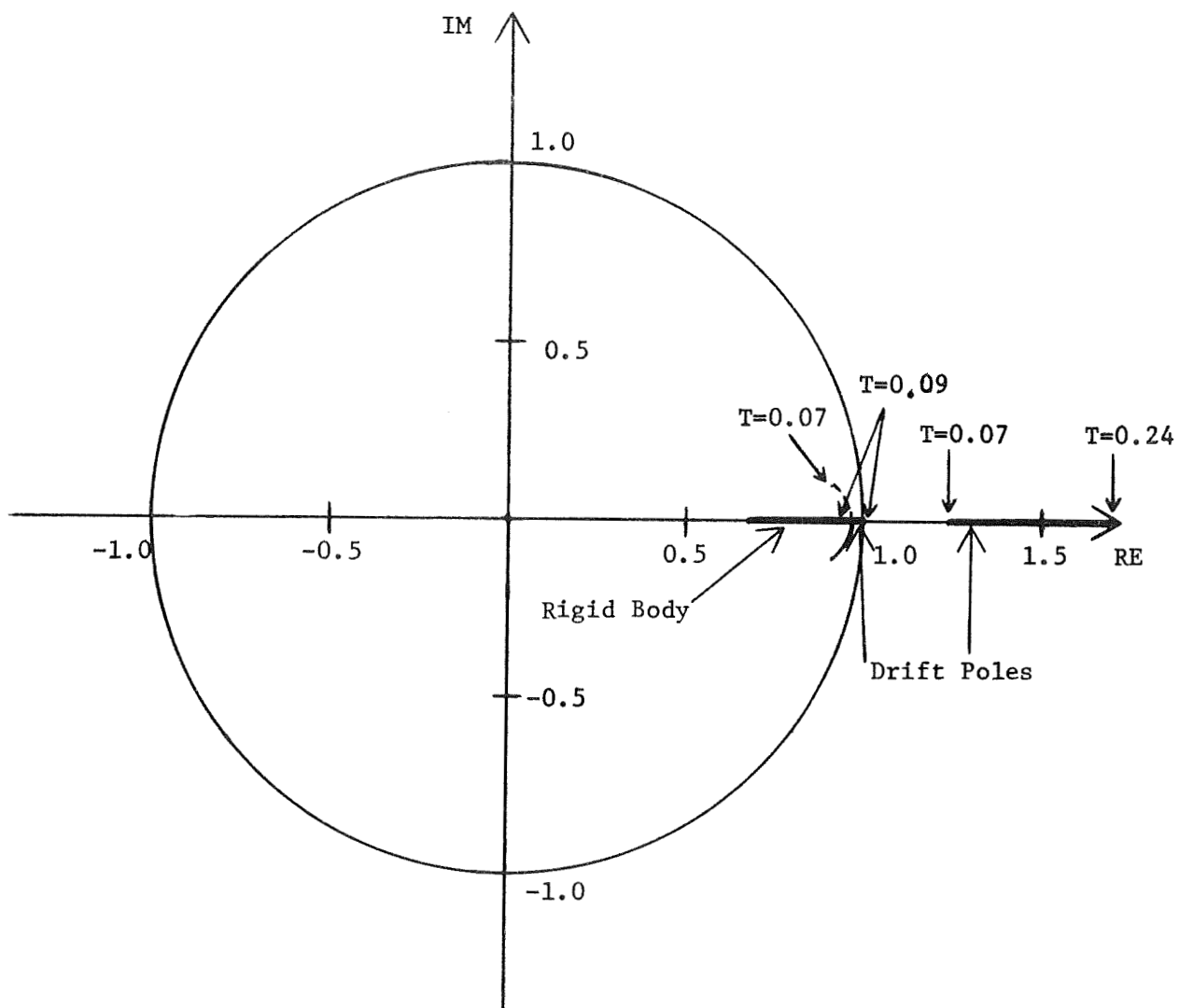
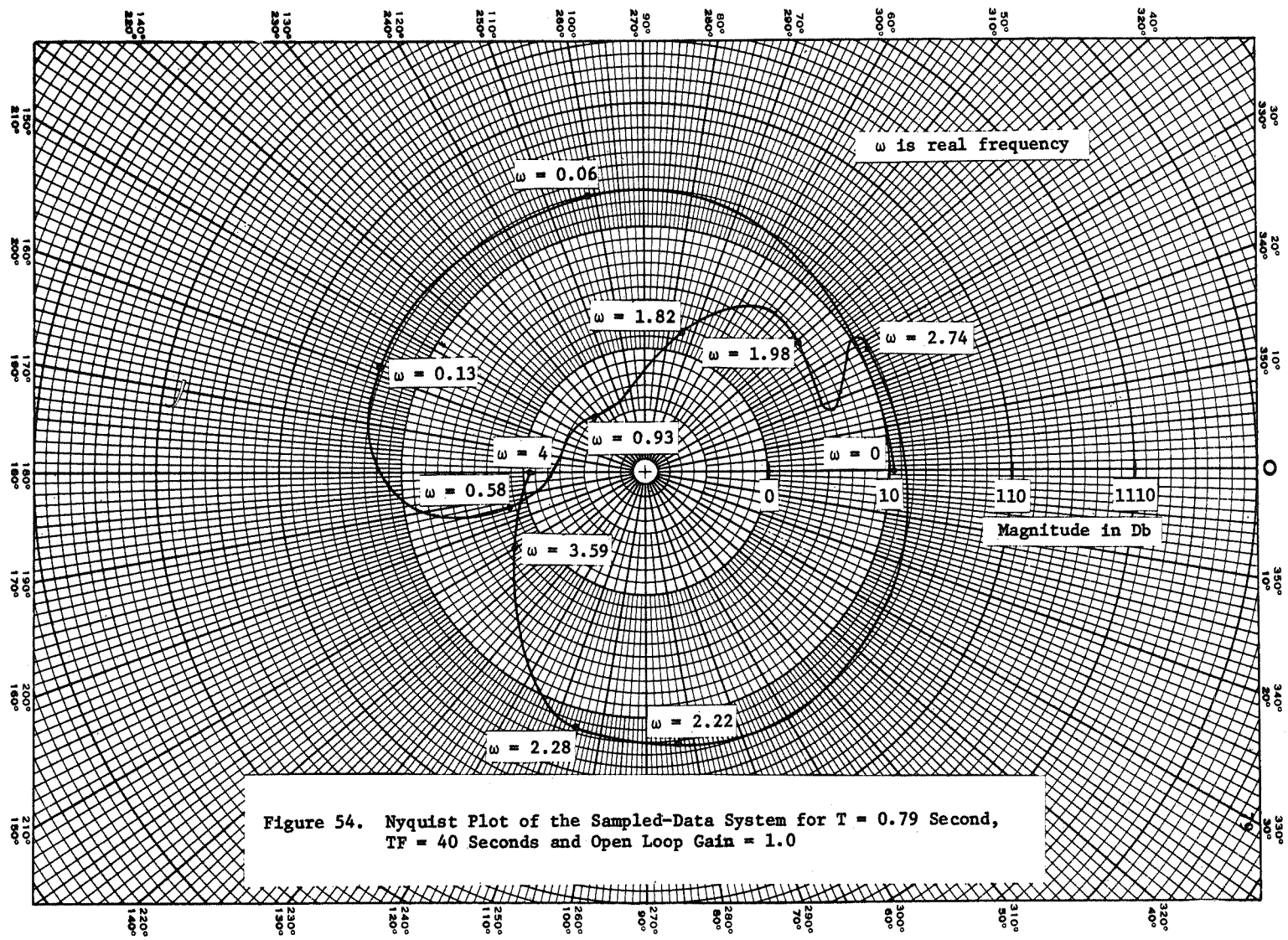


Figure 53. Plot of Drift and Rigid Body Poles as a Function of Sampling Period When $TF = 140$ Seconds and Open Loop Gain = -1.0



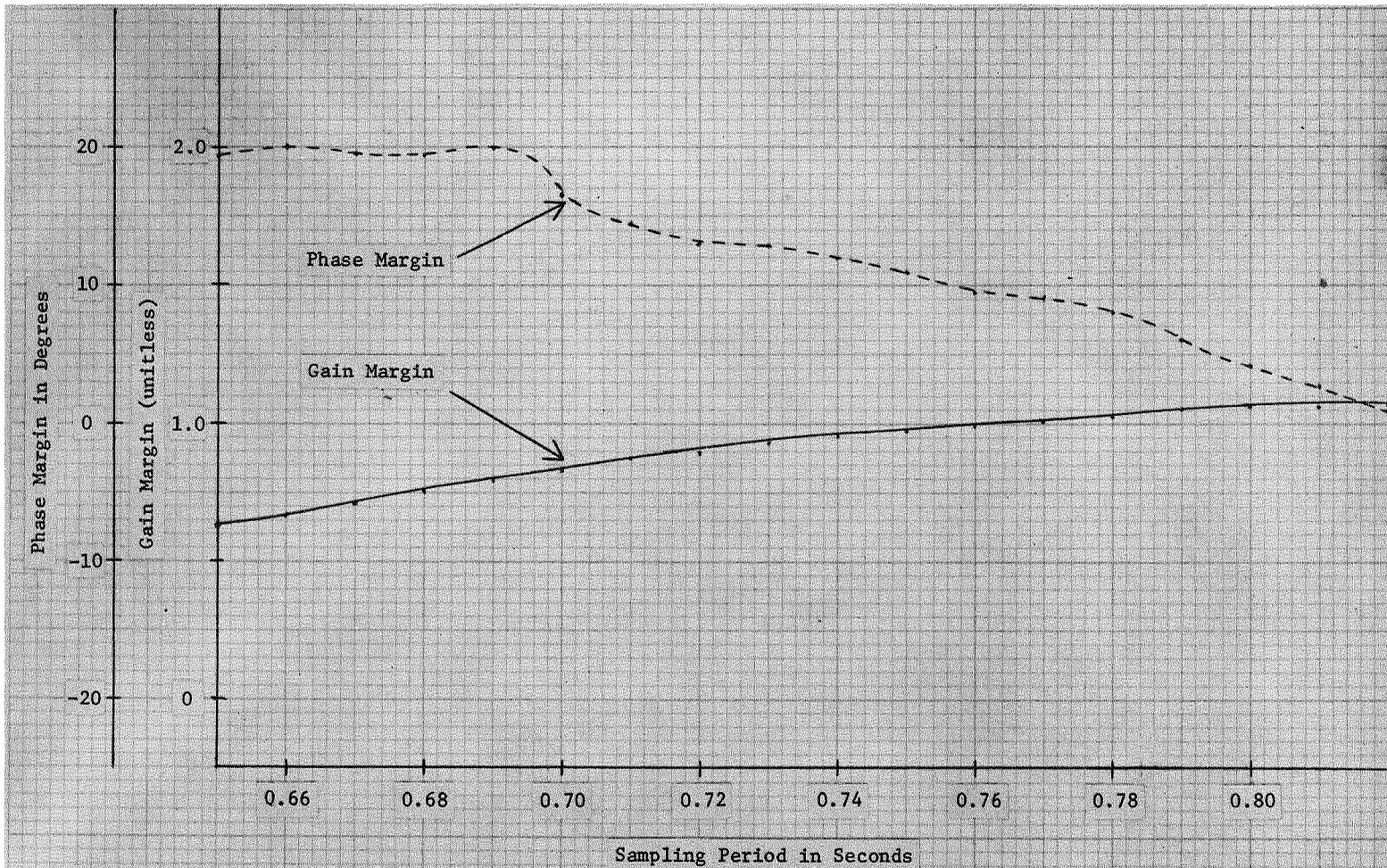


Figure 55. Phase Margin and Gain Margin vs Sampling Period for TF = 20 Seconds and Open Loop Gain = 1.0

Figure 56. Phase Margin and Gain Margin vs Sampling Period for $T_F = 40$ Seconds and Open Loop Gain = 1.0

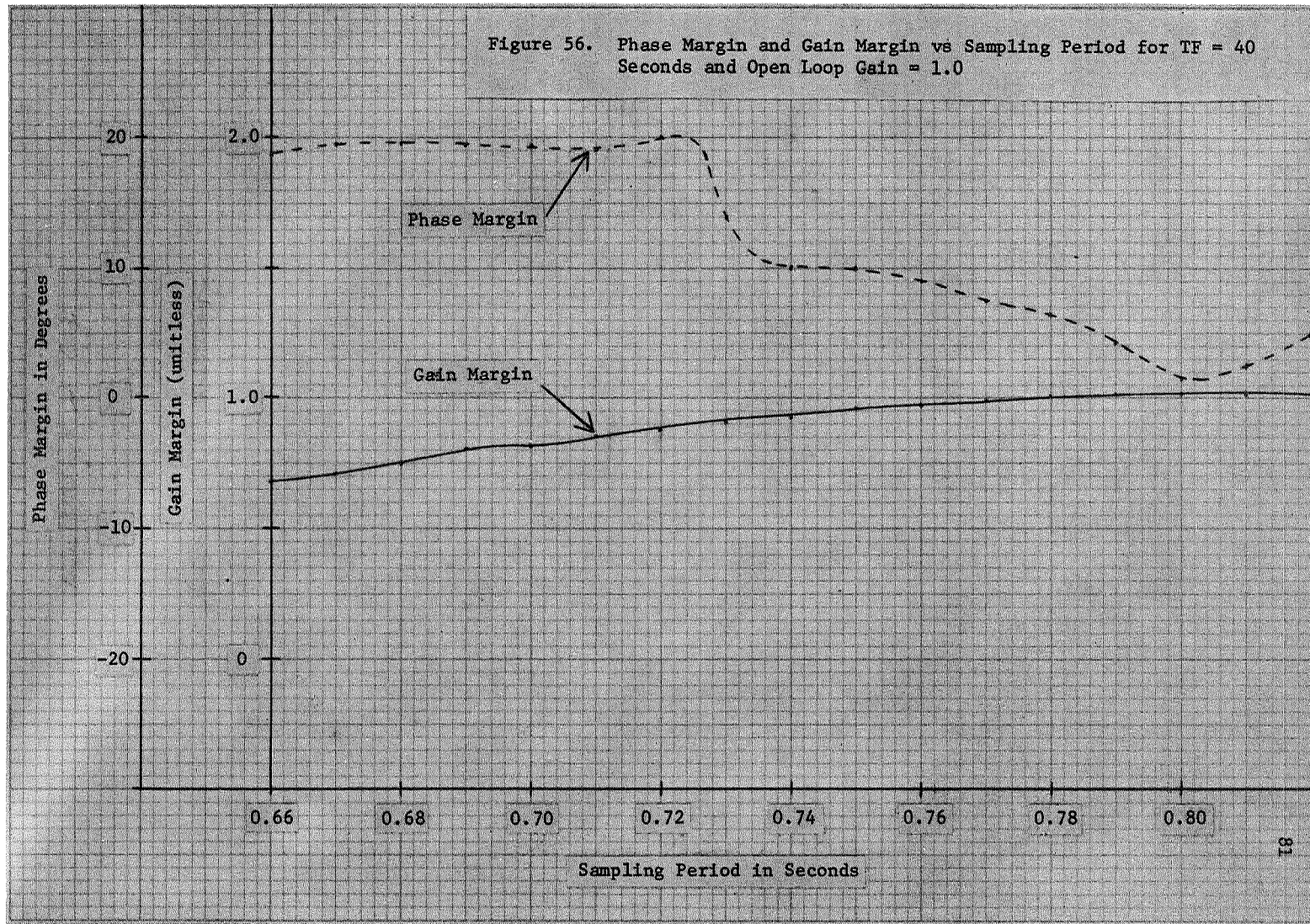


Figure 57. Phase Margin and Gain Margin vs Sampling Period for TF = 60 Seconds and Open Loop Gain = 1.0

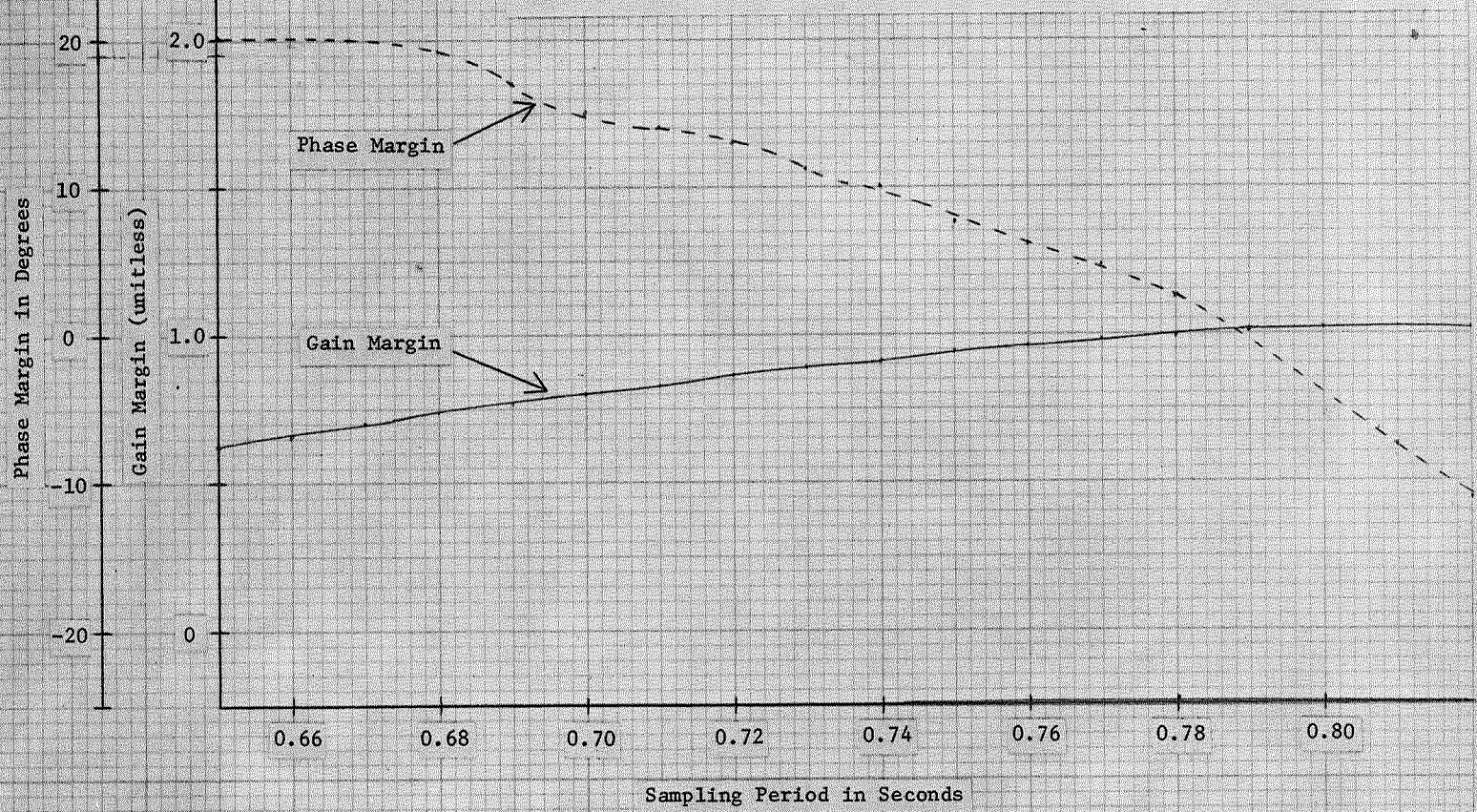


Figure 58. Phase Margin and Gain Margin vs Sampling Period for TF = 80.11 Seconds and Open Loop Gain = 1.0

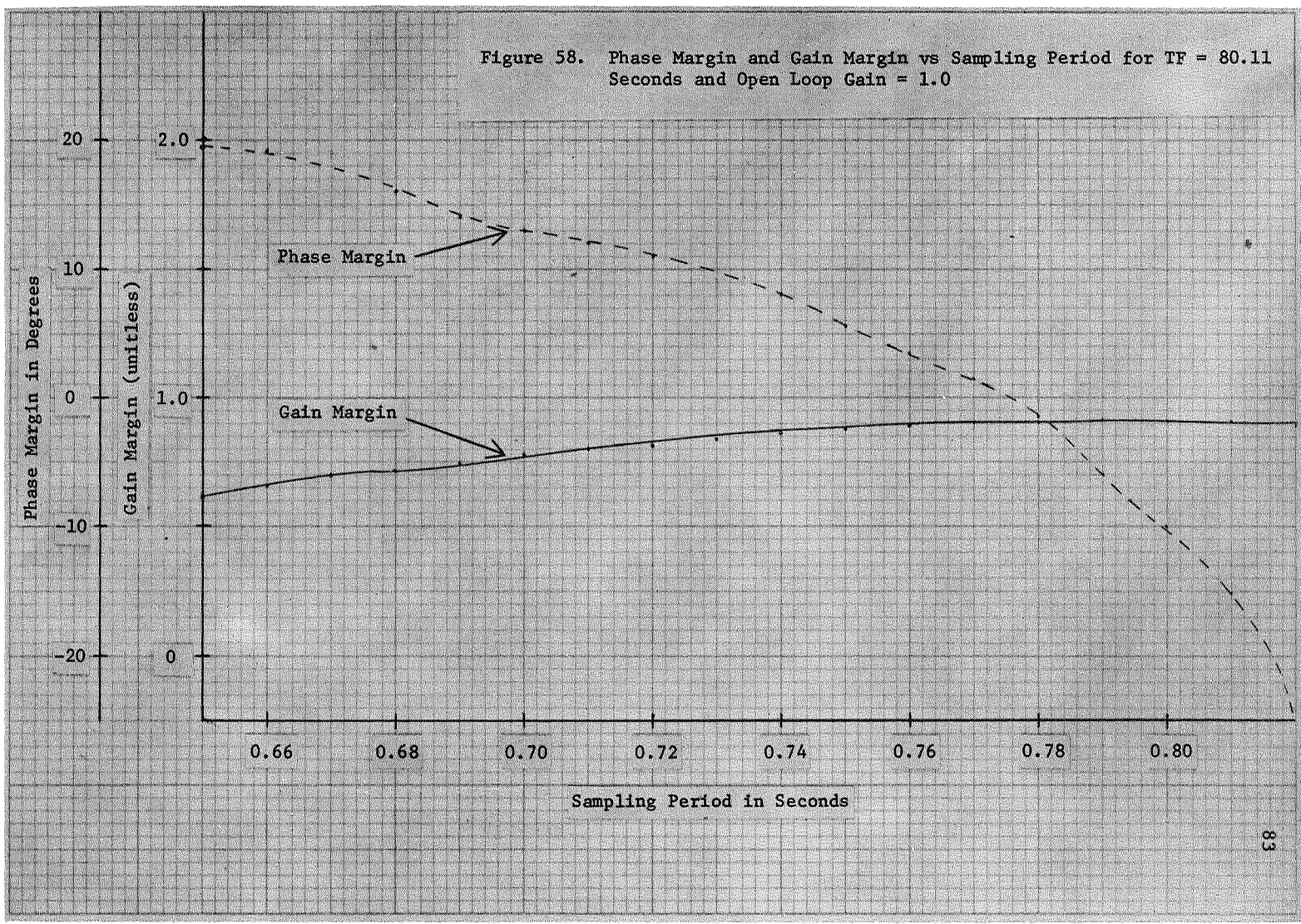


Figure 59. Phase Margin and Gain Margin vs Sampling Period for TF = 100 Seconds and Open Loop Gain = 1.0

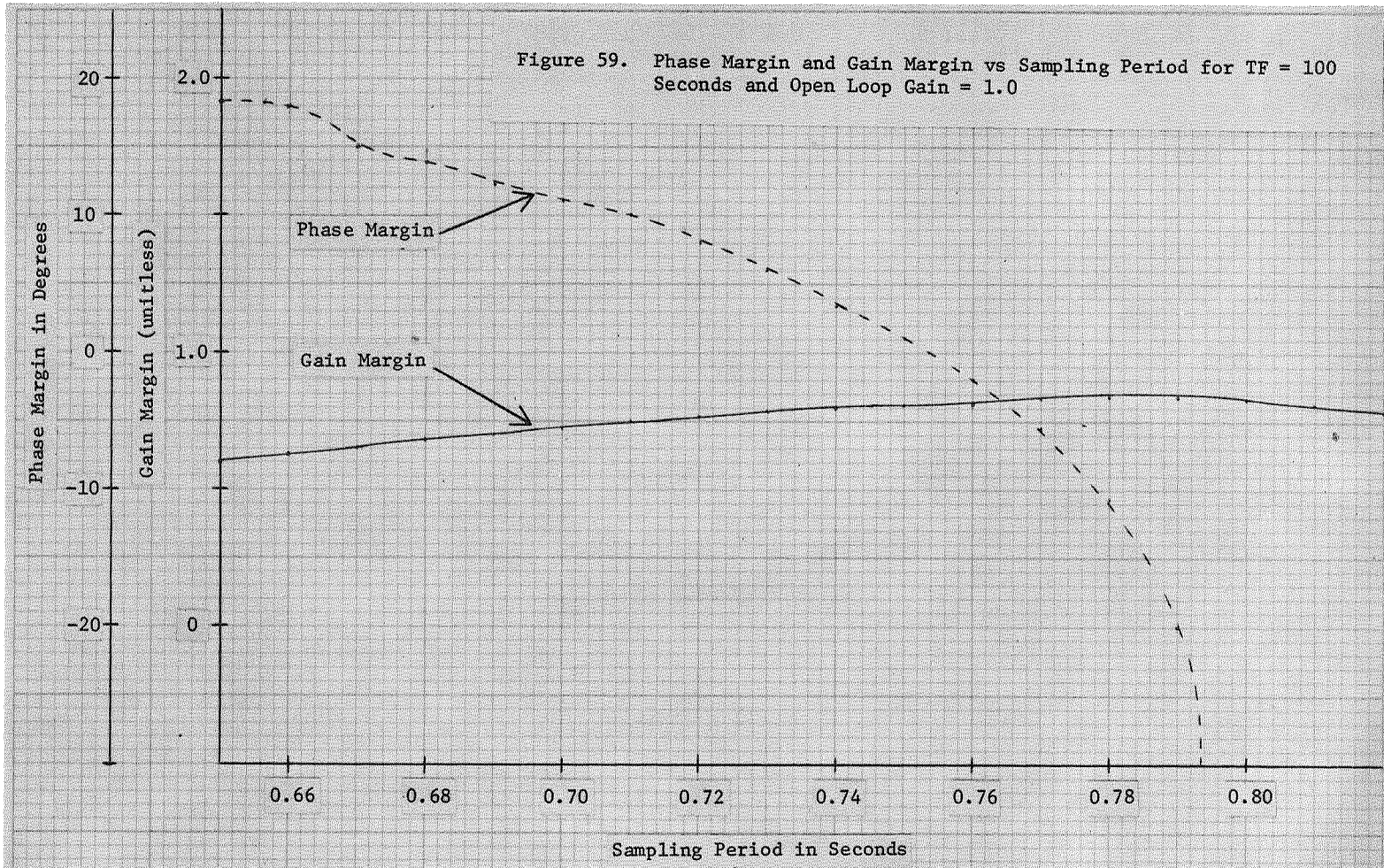
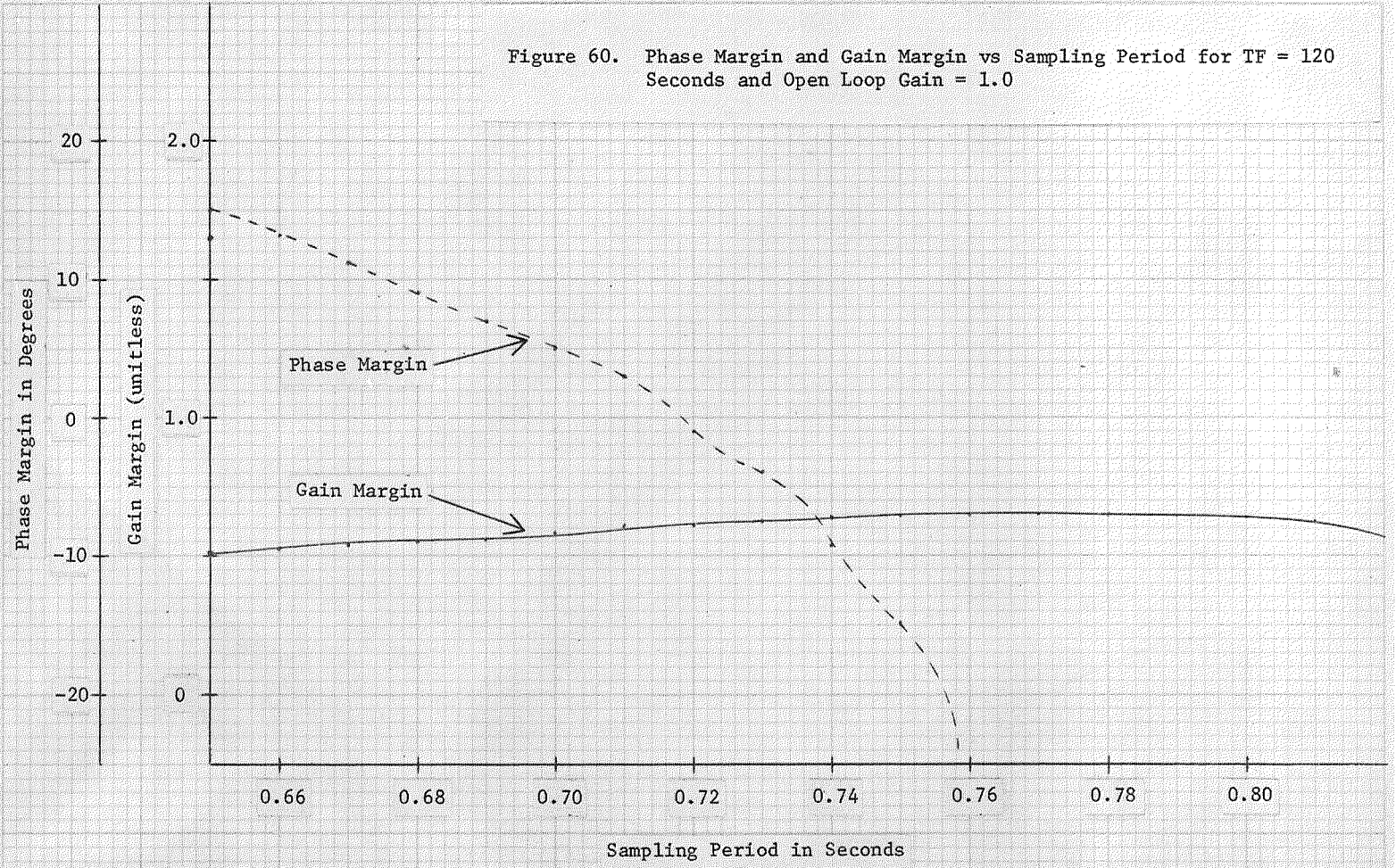


Figure 60. Phase Margin and Gain Margin vs Sampling Period for TF = 120 Seconds and Open Loop Gain = 1.0



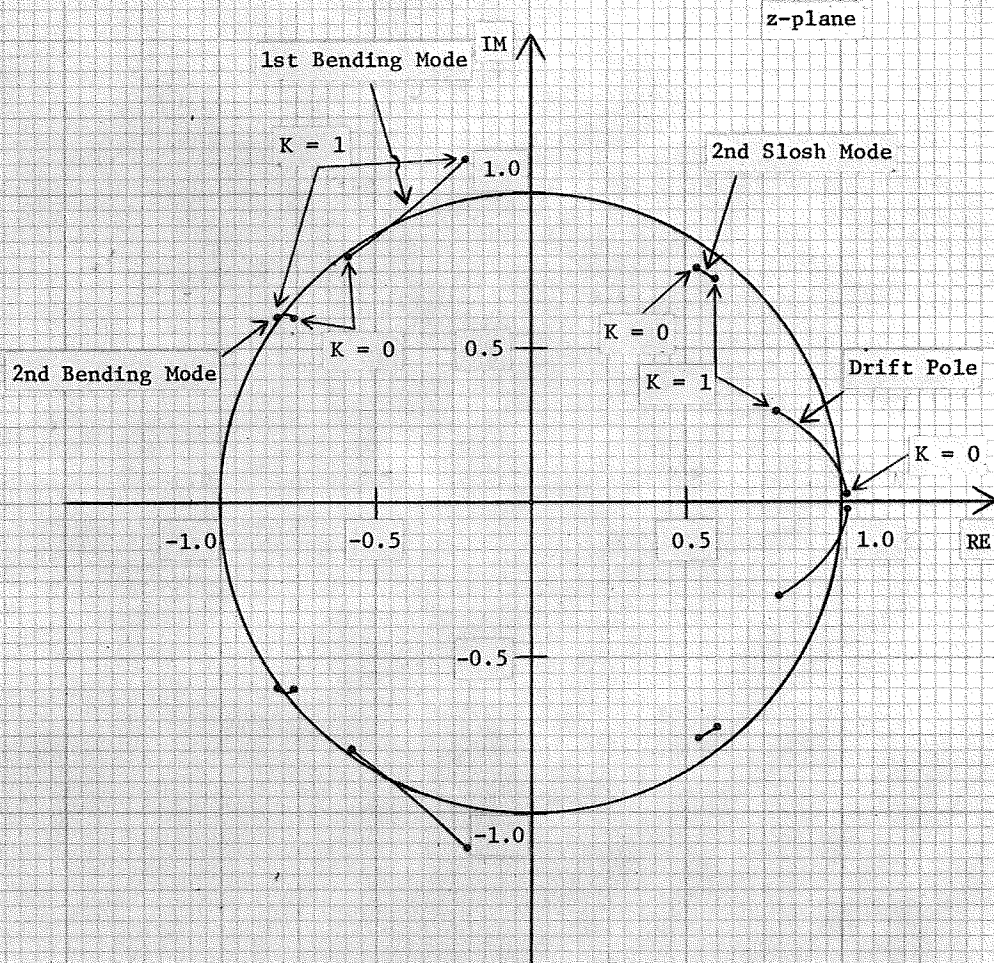
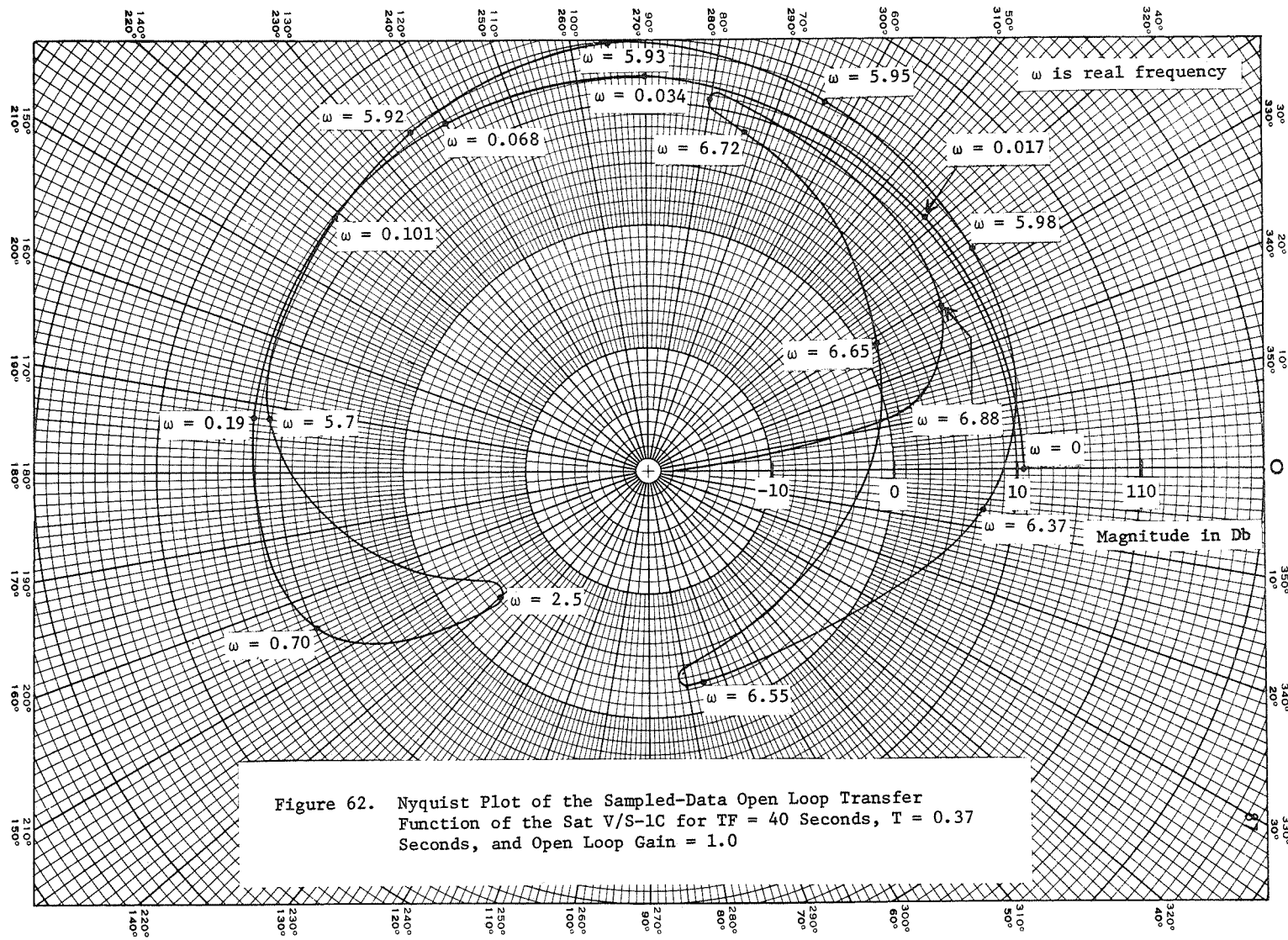


Figure 61. Regular Sampled-Data Root Loci of the Poles of Prime Importance of the Sat V/S-1C When $T = 0.37$ Seconds and $T_F = 40$ Seconds



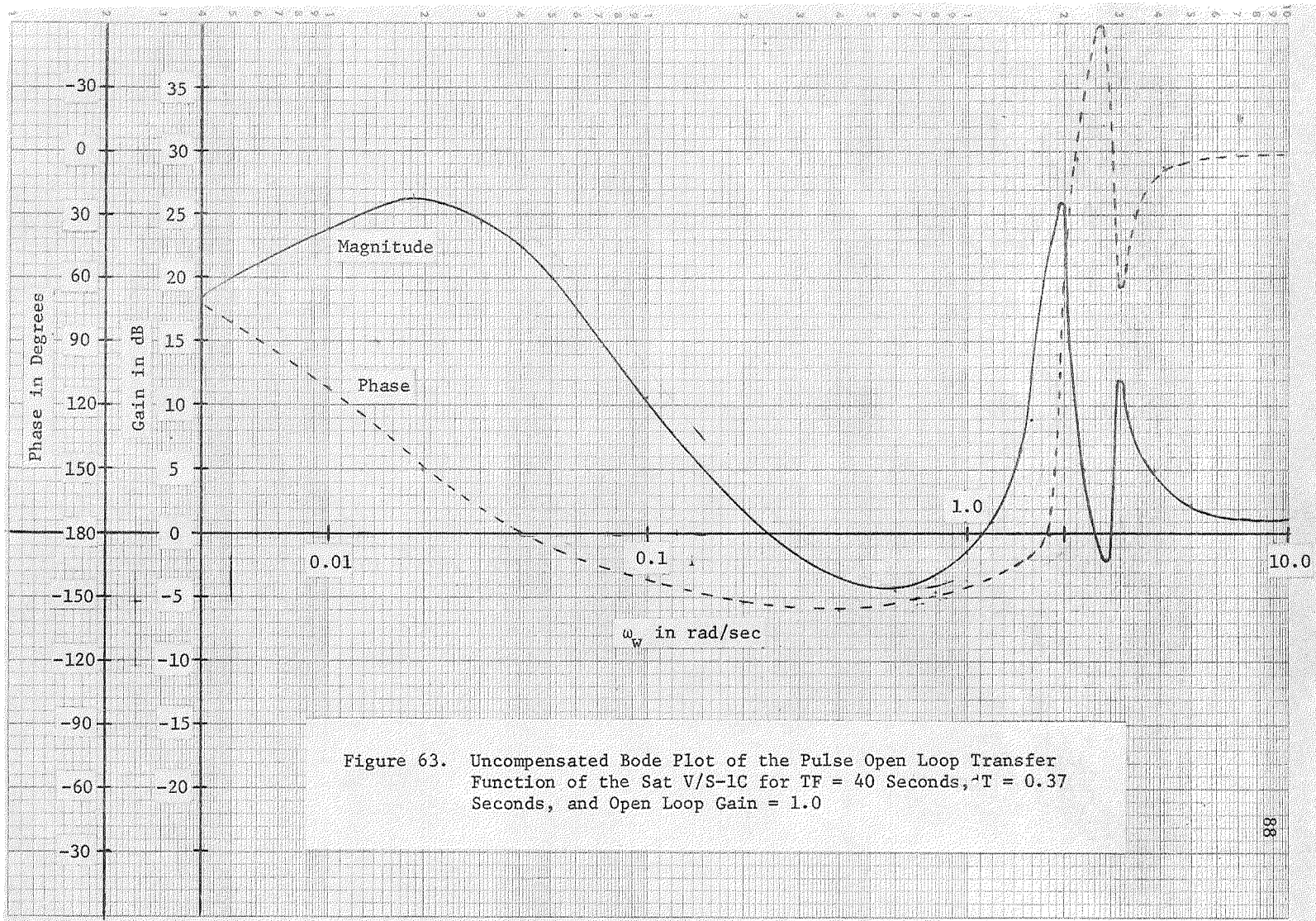
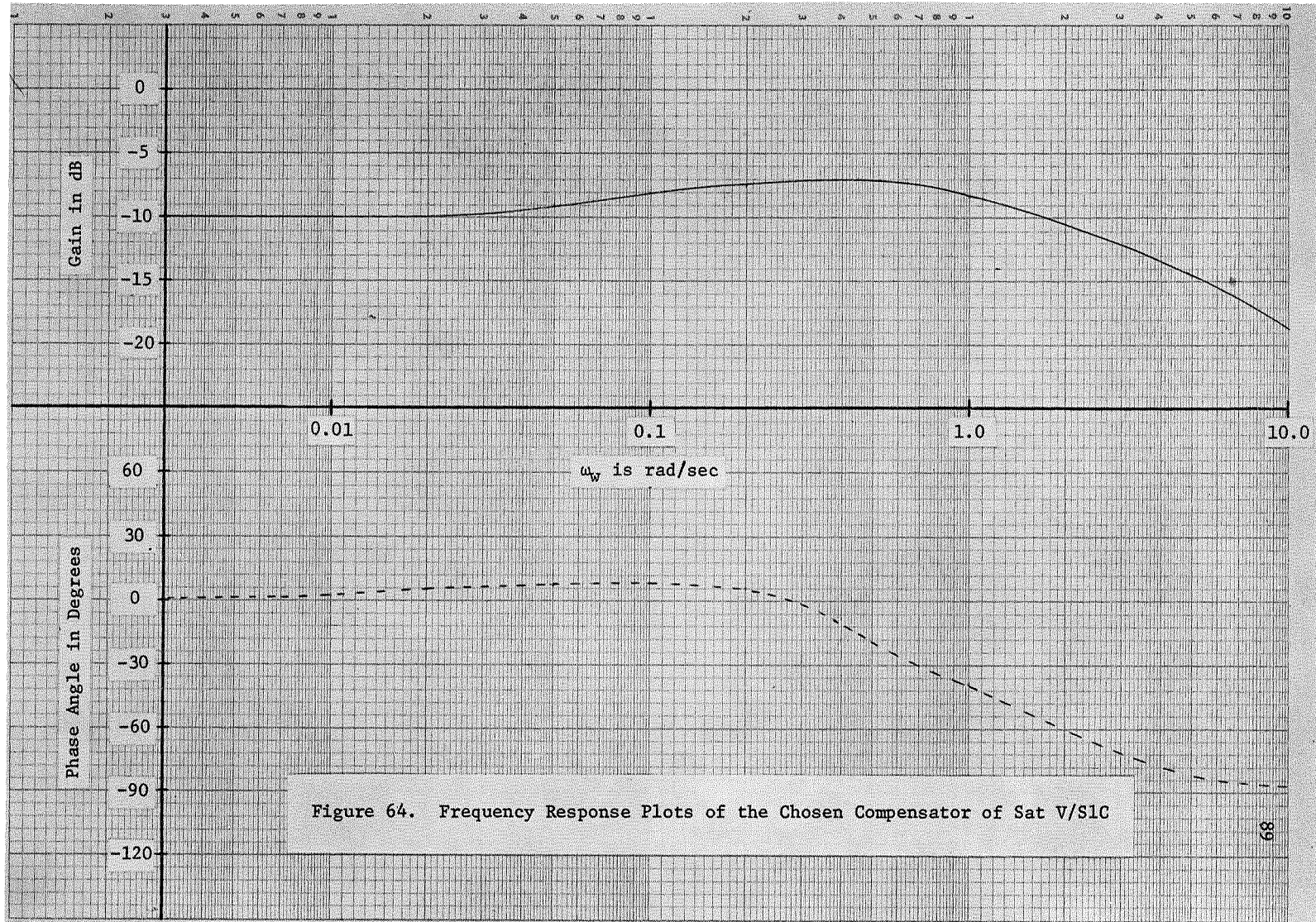


Figure 63. Uncompensated Bode Plot of the Pulse Open Loop Transfer Function of the Sat V/S-1C for TF = 40 Seconds, $T = 0.37$ Seconds, and Open Loop Gain = 1.0



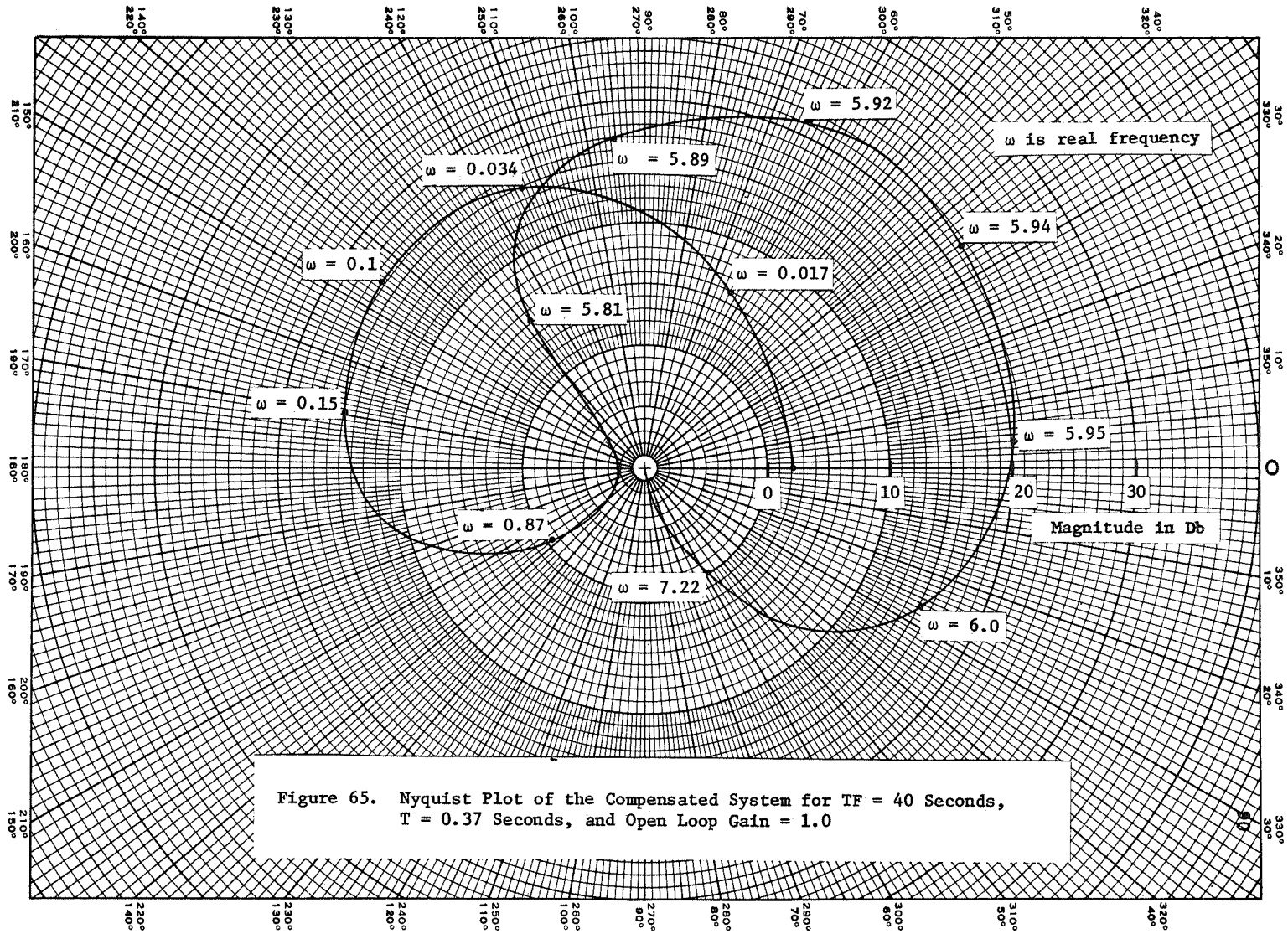


Figure 65. Nyquist Plot of the Compensated System for TF = 40 Seconds, T = 0.37 Seconds, and Open Loop Gain = 1.0

Figure 66. Gain and Phase Margins vs Sampling Period for TF = 40 Seconds and Open Loop Gain = 1.0

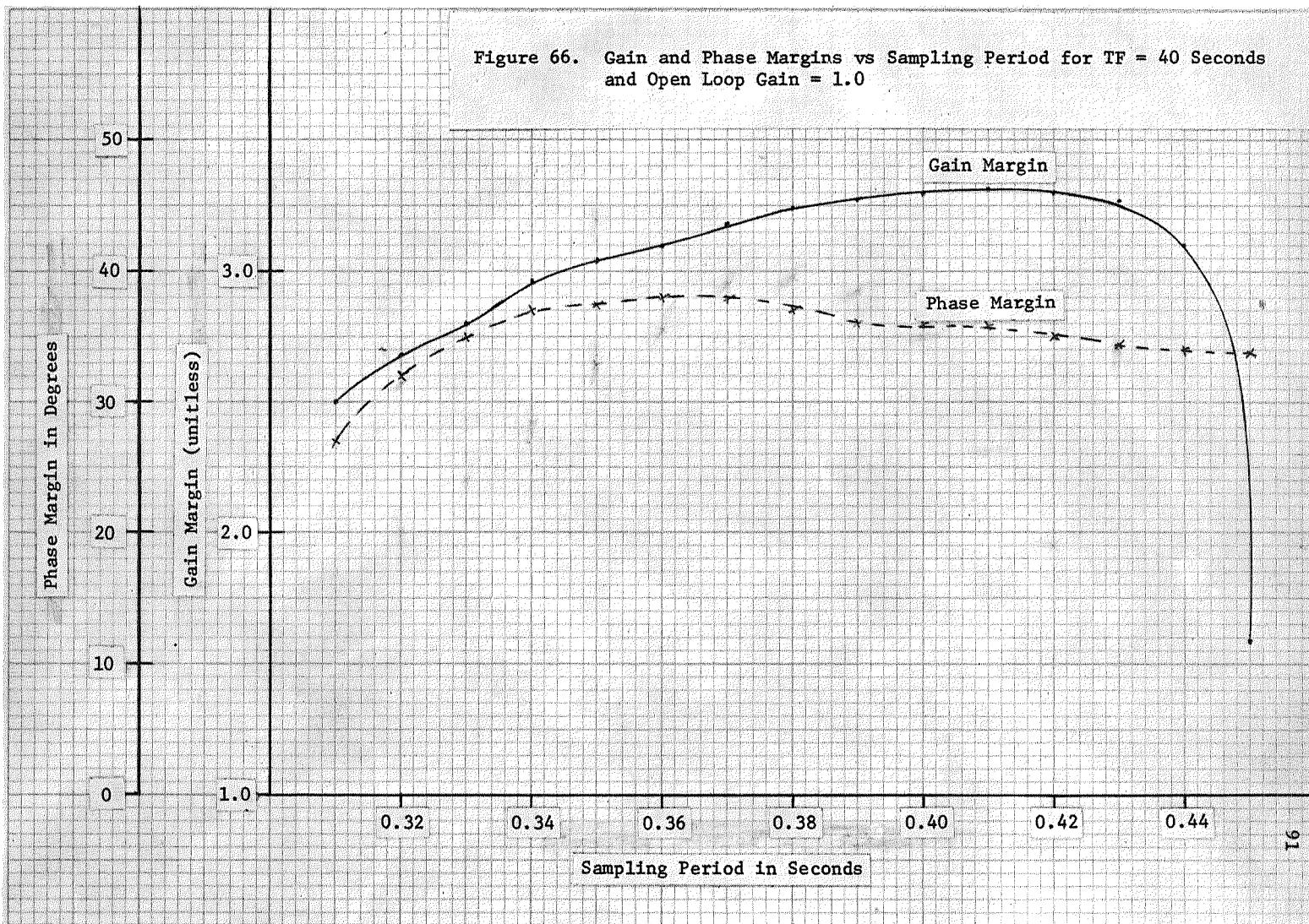


Figure 67. Gain and Phase Margins vs Sampling Period for TF = 80.11 Seconds and Open Loop Gain = 1.0

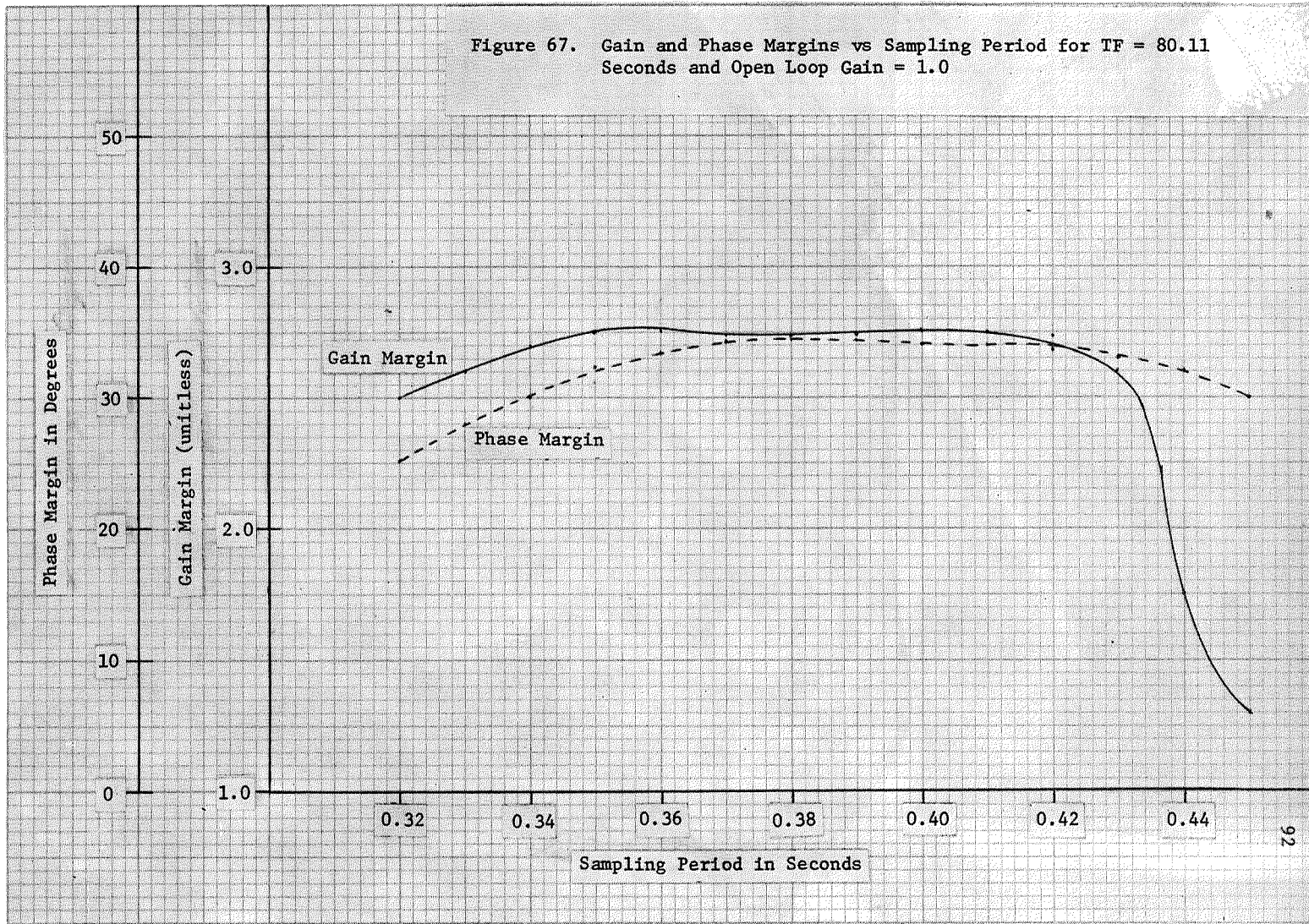
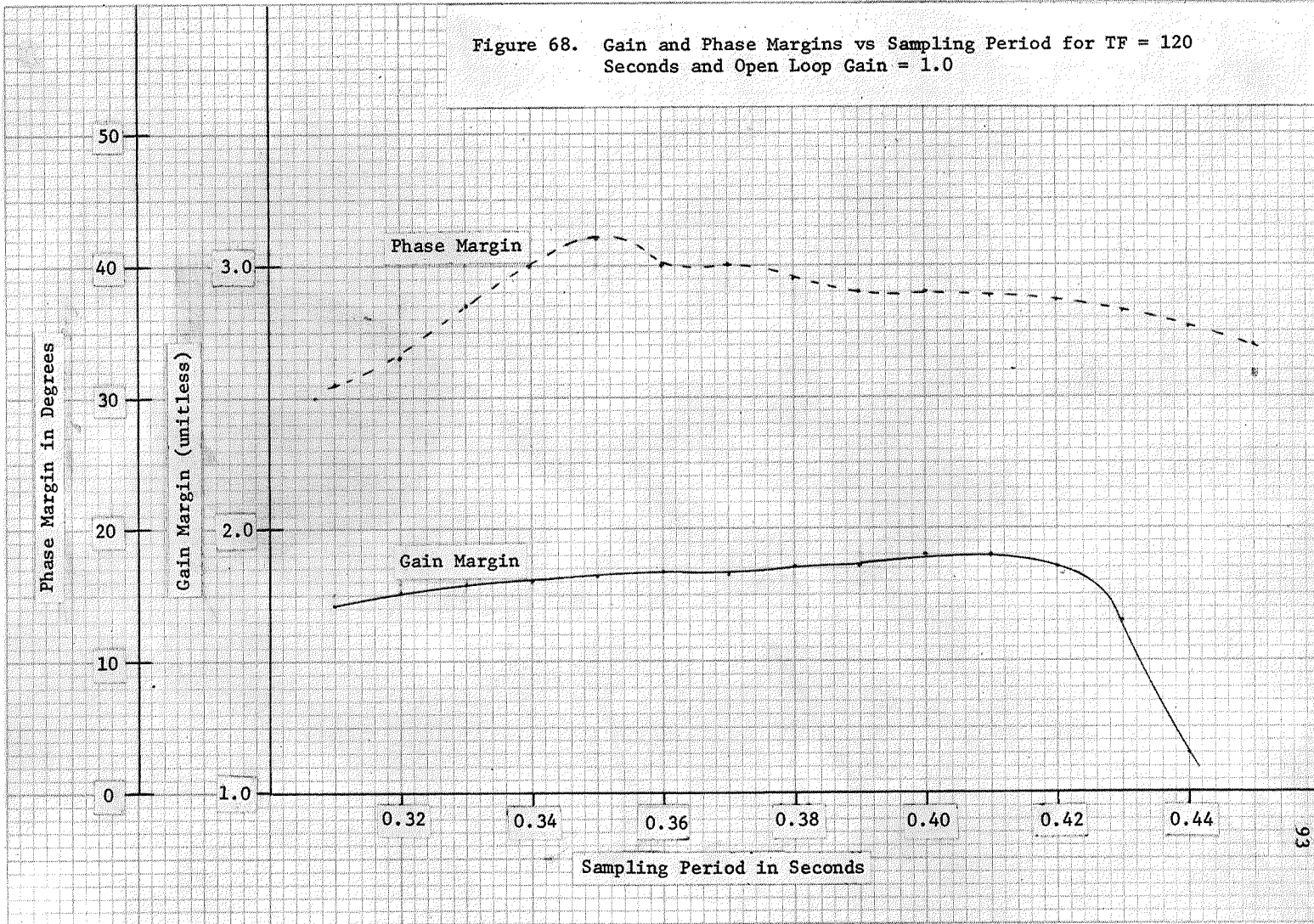


Figure 68. Gain and Phase Margins vs Sampling Period for TF = 120 Seconds and Open Loop Gain = 1.0



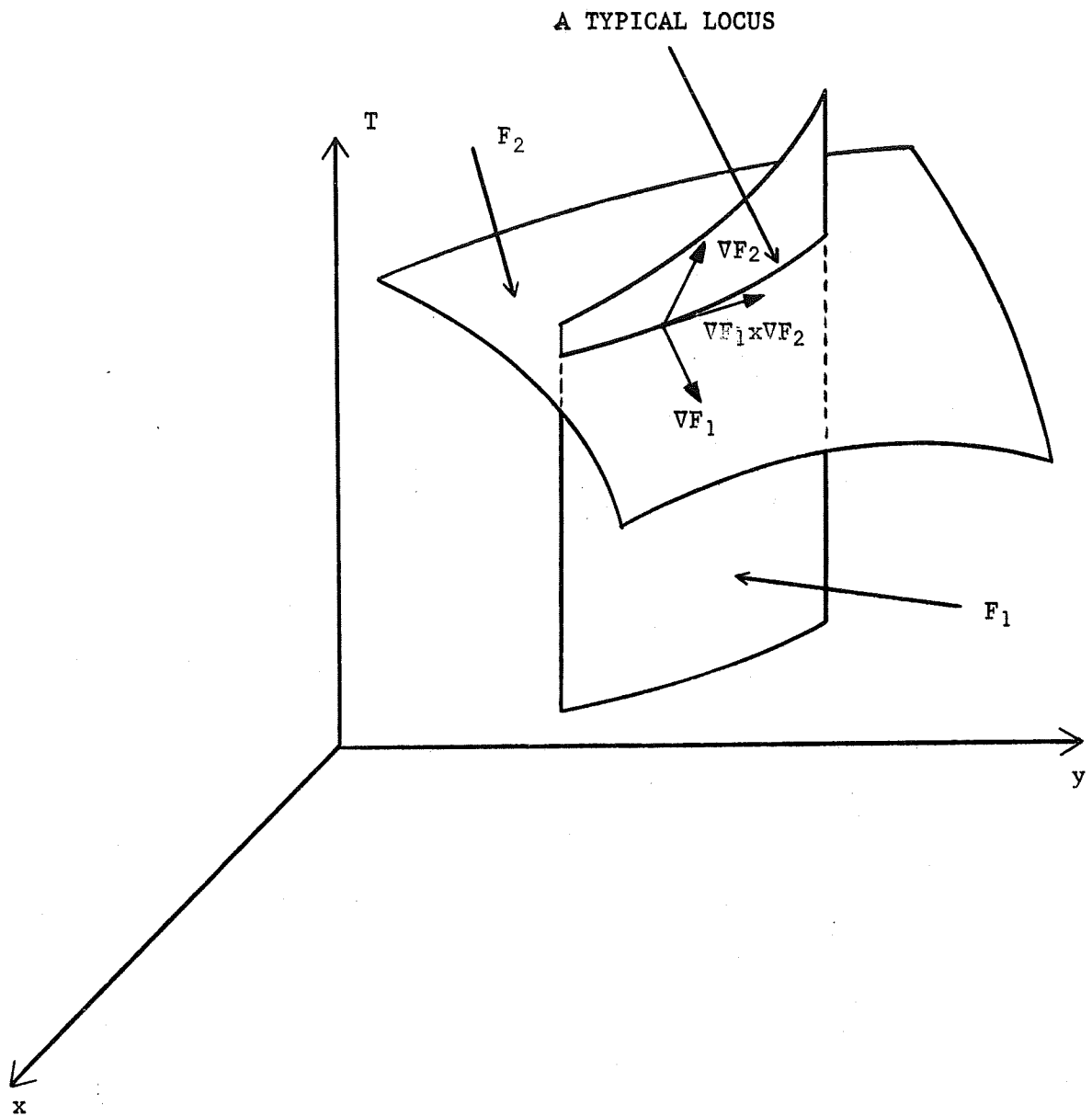


Figure 69. A Pictorial Representation of the Two Surfaces F_1 and F_2 , with indicated gradients and cross product.

APPENDICES

APPENDIX A

S-PLANE BLOCK DIAGRAM REDUCTION AND PARTIAL FRACTION EXPANSION PROGRAM

The following computer program with the associated sub-programs is an aid in finding the necessary s-plane information to facilitate the sampled-data studies. It possesses the following properties:

1. For a particular flight time, it reduces the block diagram beginning with Figure 2.
2. For a particular flight time it finds the roots of the s-plane open loop transfer functions.
3. It finds the partial fraction expansion of the s-plane open loop transfer function.

In obtaining the s-plane transfer function the utilization of polynomial multiplication and addition is used throughout the program. Also, polynomial division is used at one point to factor out common factors which were known to exist in both the numerator and denominator of a certain rational polynomial. After the s-plane open loop transfer function is obtained, the program finds the roots and the partial fraction expansion coefficients.

Many of the symbols of input data used in the program are defined in the List of Symbols. Several of the rational polynomial and constants are defined in Figure 6. The program uses two vectors in defining a rational polynomial--one for its numerator coefficients and one for its denominator coefficients. The order of the coefficients is

normal with respect to the powers of x .^{*} The variables used for the numerator and denominator vectors of a rational polynomial in normal order are exactly the same except for the final digit. The last digit differentiates the vectors of numerators from the vectors of the denominators. An N indicates the vector is a numerator vector; whereas, a D indicates a denominator vector. An R on the end of a vector connotes the vector is in reverse order from normal order. The variables which are not defined in the symbol list or in Figure 6 are simply working variables (variables for internal use only, not for input or output) of the program. All output data is defined in the print outs.

```

C
C   SATURN-V TRANSFER FUNCTION AND ROOT PROGRAM
C
  DIMENSION DIVPOL(50)
  DIMENSION G3N(30),G3D(30),G4N(30),G4D(30),B8N(30),B8D(30),B7N(30),
1  B7D(30),B6N(30),B6D(30),B9N(30),B9D(30),B10AN(30),B10BN(30),
2  B10N(30), B10D(10), G1N(10), G1D(10), G2N(10), G2D(10)
  DIMENSION G1DR(25),G2DR(25),G3DR(25),G4DR(25),ROOTR(25),ROOTI(25),
1  COF(25)
  DIMENSION RTRG1(10), RTRG2(10), RTIG2(10), RTRG3(10),
1  RTIG3(10),RTRG4(10), RTIG4(10), WSSN(5), WSSD(10),
2  AK(25), GTAAN(15), GTAN(15), GTBAN(15), GTBN(15), GTCAN(15),
3  GTCN(15), GTN(15), GTD(15), GWD(20), ROOTC(20)
  DIMENSION RTRWSS(10), RTIWSS(10), WSSDR(10), GTAD(15)
  REAL K4, MI, MASS, K3, K7, MS1, K5
  DOUBLE PRECISION B8N, B8D, B7N, B7D, B6N, B6D, B9N, B9D, B10AN,
1  B10BN, B10N, B10D, G1N, G1D, G2N, G2D, DIVPOL, G3N,
2  G3D, G4N, G4D
  DOUBLE PRECISION WSSN, WSSD, GTAAN, GTAN, GTBAN, GTBN, GTCAN, GTCN,
1  GTD, GTAD, GWD, GTN
  COMPLEX*16 ROOTC, AK1(25), AK2(25)
  READ(1,1) DS1, ZB1, ZB2, SIG, THE
1  FORMAT(5F14.5)
  WRITE(3,2)
2  FORMAT('1',9X,'DS1',12X,'ZB1',11X,'ZB2',11X,'SIG',12X,'THE')
  WRITE(3,3) DS1, ZB1,ZB2,SIG,THE
3  FORMAT(3X,5F14.5)
  DO 50 IX=1,10

```

^{*}Normal order means the coefficient corresponding the higher power of s is first, the coefficient corresponding to the next highest power of s is second, etc.

```

C
C
    READ(1,1) TIME
    WRITE(3,4) TIME
C
4  FORMAT('0',30X,'TIME = ',F6.2)
C
    READ(1,5) MI, MASS, RP, ZM1, ZM2, MS1
5  FORMAT(6F12.2)
    WRITE(3,6)
6  FORMAT('0',
1  5X,'MI',9X,'MASS',9X,'RP',10X,'ZM1',9X,'ZM2',9X,'MS1')
    WRITE(3,18)MI, MASS, RP, ZM1, ZM2, MS1
18 FORMAT(2X,6F12.2)
C
C
    READ(1,7) GBAR,C1, K3, K5, K7, FB1, FB2
7  FORMAT (7F10.6)
    WRITE(3,8)
8  FORMAT('0',3X,'GBAR',7X,'C1',8X,'K3',8X,'K5',8X,'K7',8X,'FB1',
1  7X,'FB2')
    WRITE(3,17)GBAR, C1, K3, K5, K7, FB1, FB2
17 FORMAT(2X,7F10.6)
C
C
    READ(1,7) YPB1, YPB2, YB1, YB2, YP1P1, YP1P2
    WRITE(3,9)
9  FORMAT('0',3X, 'YPB1',6X,'YPB2',7X,'YB1',7X,'YB2',6X,'YP1P1',
1  5X,YP1P2')
    WRITE(3,17) YPB1, YPB2, YB1, YB2, YP1P1, YP1P2
C
C
    READ(1,7) YPRG1, YPRG2, FS1, XS1, K4, C2
    WRITE(3,100)
100 FORMAT('0',3X,'YPRG1',5X,'YPRG2',6X,'FS1',7X,'XS1',8X,'K4',8X,
1  'C2')
    WRITE(3,17)YPRG1, YPRG2, FS1, XS1, K4, C2
    WS1= 2.0 * FS1*3.1416
    WB1= 2.0 * FB1*3.1416
    WB2= 2.0 * FB2*3.1416
C
C
    COEFFICIENTS OF B6
C
    B6N(1)= (-57.3 * MS1 * XS1)/MI
    B6N(2)= K5 * C1
    B6N(3)= K5 * C1 * 2.0 * DS1 * WS1 -(57.3 * MS1 * GBAR) / MI
    B6N(4)= K5 * C1 * WS1**2
    B6D(1)= 1.0 - MS1/ MASS
    B6D(2)= K5 * K7 + 2.0 * DS1 * WS1
    B6D(3)= WS1 ** 2 + 2.0 * DS1 * WS1 * K5 * K7
    B6D(4)= K5 * K7 * WS1**2
    B6D1=B6D(1)
    DO 89 IK=1,4
    B6N(IK)= B6N(IK)/ B6D1
89 B6D(IK)= B6D(IK)/ B6D1

```

```

WRITE(3,900)
900 FORMAT('0',28X,'COEFFICIENTS OF B6')
WRITE(3,300)
WRITE(3,400) (B6N(I), I=1,4)
WRITE(3,500)
WRITE(3,400) (B6D(I), I=1,4)

```

C
C
C

COEFFICIENTS OF B7

```

B7N(1)= - ( MS1 * XS1 ) / (57.3 * MASS)
B7N(2)= 0.0
B7N(3)= K7 +K3 - (MS1 * K3) / MASS
B7N(4)= 2.0 *DS1 * WS1 * (K7 + K3)
B7N(5)= (WS1 **2) * ( K7 + K3)
B7D(1)= 1.0
B7D(2)= 2.0*DS1 *WS1
B7D(3)= WS1**2

```

```

800 FORMAT('0',28X,'COEFFICIENTS OF B7')
WRITE(3,300)
WRITE(3,400) (B7N(I), I=1,5)
WRITE(3,500)
WRITE(3,400) (B7D(I), I=1,3)

```

C
C
C

COEFFICIENTS OF B8

```

B8N(1)= (MS1 * XS1**2)/ MI
B8N(2)= 0.0
B8N(3)=(57.3 * MS1 * XS1 * (K3 + GBAR/57.3))/ MI - C1
B8N(4)= -2.0 * C1 * DS1 * WS1
B8N(5)= ((57.3 * MS1 * K3 * GBAR)/ MI ) - C1 * WS1**2
B8D(1)=1.0
B8D(2)= 2.0*DS1 *WS1
B8D(3)= WS1**2

```

```

700 FORMAT('0',28X,'COEFFICIENTS OF B8')
WRITE(3,300)
WRITE(3,400) (B8N(I), I=1,5)
WRITE(3,500)
WRITE(3,400) (B8D(I), I=1,3)

```

C
C
C

CALCULATION OF B9 COEFFICIENTS

```

WRITE(3,11)
11 FORMAT('0',28X,'COEFFICIENTS OF THE B9')
N=3
N=2
CALL POLMUL(B6D,B7D,N,M,B9D)
KL=N+M+1
N=3
M=4
CALL POLMUL(B6N,B7N,N,M,B9N)
L=N+M+1

```

```

      B9D1= B9D(1)
      DO 85 I=1,KL
85  B9D(I)=B9D(I)/ B9D1
      DO 86 I=1,L
86  B9N(I)= B9N(I)/ B9D1
      WRITE(3,300)
      WRITE(3,10) (B9N(I), I=1,L)
10  FORMAT(2X,6D12.5)
      WRITE(3,500)
      WRITE(3,10) (B9D(I), I=1,KL)
C
C  CALCULATIONS OF THE COEFFICIENTS FOR B10
C
      N=4
      M=5
      CALL POLMUL(B8N,B9D,N,M,B10AN)
      N=7
      M=2
      CALL POLMUL(B9N,B8D,N,M,B10BN)
      WRITE(3,169)
169  FORMAT('0',24X,'COEFFICIENTS OF B10AN AND B10BN')
      WRITE(3,16) (B10AN(I), I=1,10)
      WRITE(3,16) (B10BN(I), I=1,10)
      DO 12 KX=1,10
12  B10N(KX)= B10AN(KX) + B10BN(KX)
      M=5
      N=2
      CALL POLMUL(B8D,B9D,N,M,B10D)
C
C  FACTOR LIKE TERMS OUT OF B10N AND B10D
C
      N=2
      M=9
      CALL POLDIV(B7D,B10N,N,M,DIVPOL)
      N=2
      M=7
      CALL POLDIV(B7D,DIVPOL,N,M,B10N)
      WRITE(3,101) (DIVPOL(I),I=1,N)
      N=2
      M=7
      CALL POLDIV(B7D,B10D,N,M,DIVPOL)
      N=2
      M=5
      CALL POLDIV(B7D,DIVPOL,N,M,B10D)
      WRITE(3,101) (DIVPOL(I),I=1,N)
101  FORMAT('0',4D12.3)
      WRITE(3,175)
175  FORMAT('0',28X,'COEFFICIENTS OF B10')
      WRITE(3,300)
      WRITE(3,10) (B10N(I),I=1,6)
      WRITE(3,500)
      WRITE(3,10) (B10D(I),I=1,4)

```

```

C
C   COEFFICIENTS OF G1
C
      DO 13 KY=1,4
      G1N(KY)= -C2 * B6D(KY) + K4 * B6N(KY)
13  G1D(KY)= B6D(KY)
      WRITE(3,14)
14  FORMAT('0',27X,'COEFFICIENTS OF G1')
      WRITE(3,15)
15  FORMAT('0',31X,'NUMERATOR')
      WRITE(3,16) (G1N(I), I=1,4)
16  FORMAT('0',4D18.5)
      WRITE(3,24)
      WRITE(3,16) (G1D(I), I=1,4)

C
C   COEFFICIENTS OF G2
C
      DO 20 IZ=1,6
20  G2N(IZ)= B10D(IZ)
      DO 21 IZ=5,6
21  B10D(IZ)=0.0
      DO 22 IZ=1,6
22  G2D(IZ)= B10D(IZ) - B10N(IZ)
      WRITE(3,23)
23  FORMAT('0',27X,'COEFFICIENTS OF G2')
      WRITE(3,15)
      WRITE(3,16) (G2N(I), I=1,4)
      WRITE(3,24)
24  FORMAT('0',30X,'DENOMINATOR')
      WRITE(3,16) (G2D(I), I=1,6)

C
C   CALCULATION OF COEFFICIENT FOR DENOMINATOR AND NUMERATOR FOR G3
C   AND G4
C
      G3K=1.0
      G3N(1)=G3K * (SIG * YB1 - THE * YPB1)
      G3N(2) = 0.0
      G3N(3) = G3K * RP * YB1
      G3D(1) = 1.0
      G3D(2)= 2.0 * ZB1 *WB1
      G3D(3) = WB1**2
      WRITE(3,200)
200 FORMAT('0',28X,'COEFFICIENTS OF G3')
      WRITE(3,300)
300 FORMAT('0',31X,'NUMERATOR')
      WRITE(3,400) (G3N(I), I=1,3)
400 FORMAT(2X,4D18.8)
      WRITE(3,500)
500 FORMAT('0',30X,'DENOMINATOR')
      WRITE(3,400) (G3D(I), I=1,3)

C
      G4K=1.0
      G4N(1)= G4K * (SIG * YB2 - THE * YPB2)

```



```

G4N(2) = 0.0
G4N(3) = G4K * RP * YB2
G4D(1) = 1.0
G4D(2) = 2.0 * ZB2 * WB2
G4D(3) = WB2 **2
WRITE(3,600)
600 FORMAT('0',28X,'COEFFICIENTS OF G4')
WRITE(3,300)
WRITE(3,400) (G4N(I), I=1,3)
WRITE(3,500)
WRITE(3,400) (G4D(I), I=1,3)

C
C   COEFFICIENTS OF WSS
C
WSSN(1)=1000000.0/ 0.1189
WSSD(1)= 1.0
WSSD(2)= (0.1544D-04) * WSSN(1)
WSSD(3)= (0.1338D-02) * WSSN(1)
WSSD(4)= (0.0393) * WSSN(1)
WSSD(5)= WSSN(1)
WRITE(3,255)
255 FORMAT('0',28X,'COEFFICIENTS OF WSS')
WRITE(3,300)
WRITE(3,400) WSSN(1)
WRITE(3,500)
WRITE(3,400) (WSSD(I), I=1,5)

C
C   POLES OF SERVO
C
DO 256 I=1,5
L=6-I
256 WSSDR(I)= WSSD(L)
M=4
CALL POLRT(WSSDR,COF,M,RTRWSS,RTIWSS,IER)
IF(IER)278,278,257
257 WRITE(3,73) IER
GO TO 50
278 CONTINUE
WRITE(3,258)
258 FORMAT('0', 30X,'ROOTS OF WSSD')
WRITE(3,76)(RTRWSS(I), RTIWSS(I), I=1,4)

C
C   DETERMINING ROOTS OF G1D, G2D, G3D, AND G4D.
C
C   FOR G1D
DO 71 J=1,4
KA=5-J
71 G1DR(J)= G1D(KA)
M=3
CALL POLRT(G1DR,COF,M,ROOTR,ROOTI,IER)
IF(IER)74,74,72

```

```

72 WRITE(3,73) IER
73 FORMAT(20X,'IER = ',I2)
   GO TO 90
74 WRITE(3,75)
75 FORMAT('0',      30X,'ROOTS OF G1D')
   WRITE(3,76) (ROOTR(K),ROOTI(K),K=1,3)
76 FORMAT('0',15X,2D15.5)
C
   DO 251 J=1,3
   RTRG1(J)= ROOTR(J)
251 RTIG1(J)= ROOTI(J)
C
C   FOR G2D
C
90 CONTINUE
   DO 77 J=1,6
   KB= 7-J
77 G2DR(J)=G2D(KB)
   M=5
   CALL POLRT(G2DR,COF,M,ROOTR,ROOTI,IER)
   IF(IER)79,79,78
78 WRITE(3,73) IER
   GO TO 91
79 WRITE(3,80)
80 FORMAT('0',      30X,'ROOTS OF G2D')
   WRITE(3,76) (ROOTR(K),ROOTI(K),K=1,5)
   DO 252 J=1,5
   RTRG2(J)= ROOTR(J)
252 RTIG2(J)= ROOTI(J)
C
C   FOR G3D
C
91 CONTINUE
   DO 81 J=1,3
   KJ=4-J
81 G3DR(J)=G3D(KJ)
   M=2
   CALL POLRT(G3DR,COF,M,ROOTR,ROOTI,IER)
   IF(IER)83,83,82
82 WRITE(3,73) IER
   GO TO 92
83 WRITE(3,84)
84 FORMAT('0',      30X,'ROOTS OF G3D')
   WRITE(3,76) (ROOTR(K),ROOTI(K),K=1,2)
   DO 253 J=1,2
   RTRG3(J)= ROOTR(J)
253 RTIG3(J)= ROOTI(J)
C
C   FOR G4D
C
92 CONTINUE
   DO 97 J=1,3
   JX=4-J
97 G4DR(J)=G4D(JX)

```

```

M=2
CALL POLRT(G4DR,COF,M,ROOTR,ROOTI,IER)
IF(IER)95,95,94
94 WRITE(3,73) IER
GO TO 25
95 WRITE(3,96)
96 FORMAT('0' ,30X,'ROOTS OF G4D')
WRITE(3,76) (ROOTR(K),ROOTI(K),K=1,2)
DO 254 J=1,2
RTRG4(J)= ROOTR(J)
254 RTIG4(J)= ROOTI(J)
C
C PARTIAL FRACTION EXPANSION OF WSS*G5/S
C
C GT= G1*G2= GIN / G2D
C
C
DO 556 JR=1,3
G3N(JR)=(YP1P1 /ZM1)*G3N(JR)
556 G4N(JR)=(YP1P2 /ZM2)*G4N(JR)
N=5
557 CONTINUE
WSSD(6)= 0.0
RTRWSS(5)= 0.0
RTIWSS(5)= 0.0
C
CALL POLMUL(G3D,G4D,2,2,GTAAN)
CALL POLMUL(G1N,GTAAN,3,4,GTAN)
CALL POLMUL(G2D,G4D,5,2,GTBAN)
CALL POLMUL(GTBAN,G3N,7,2,GTBN)
CALL POLMUL(G2D,G3D,5,2,GTCAN)
CALL POLMUL(GTCAN,G4N,7,2,GTCN)
C
CALL POLMUL(G2D,G3D,5,2,GTAD)
CALL POLMUL(GTAD,G4D,7,2,GTD)
C
C
DO 263 I=1,8
K= 11 - I
J= 9 - I
263 GTAN(K)= GTAN(J)
GTAN(1)= 0.0
GTAN(2)= 0.0
DO 264 I=1,10
GTN(I)= GTAN(I) + GTBN(I) + GTCN(I)
264 GTN(I)= GTN(I) * WSSN(1)
267 CALL POLMUL(WSSD,GTD,N,9,GWD)
DO 265 I=1,2
ROOTR(I)= RTRG3(I)
265 ROOTI(I)= RTIG3(I)
DO 266 I=3,4
L=I-2
ROOTR(I)= RTRG4(L)

```

```

266 ROOTI(I)= RTIG4(L)
DO 401 I=5,9
M= I -4
ROOTR(I)= RTRG2(M)
401 ROOTI(I)= RTIG2(M)
K= 9 + N
DO 268 I=10,K
M= I - 9
ROOTR(I)= RTRWSS(M)
268 ROOTI(I)= RTIWSS(M)
C
L= N + 9
IF(N.EQ.5)GO TO 1000
CALL HEAVY(GTN,GWD,9,L,ROOTR,ROOTI,ROOTC,AK2)
GO TO 1005
1000 CALL HEAVY(GTN,GWD,9,L,ROOTR,ROOTI,ROOTC,AK1)
GO TO 1010
1005 CONTINUE
WRITE(3,269)
269 FORMAT('0',13X,'ROOTS',29X,'WSS*G5S/S CONSTANTS',23X,'WSS*G5 CONSTA
INTS')
AK2(14)= 0.0
WRITE(3,270) (ROOTC(I),AK1(I),AK2(I),I=1,14)
270 FORMAT(' ',2E16.8,8X,2D16.8,8X,2D16.8)
WRITE(2,1030) TIME
1030 FORMAT(' FLIGHT TIME = ',F6.4)
DO 1009 I=1,14
WRITE(2,1007) ROOTC(I), AK1(I)
WRITE(2,1008) AK2(I)
1007 FORMAT(2D16.8,2D24.16)
1008 FORMAT(2D24.16)
1009 CONTINUE
GO TO 273
1010 CONTINUE
C
C PARTIAL FRACTION EXPANSION OF WSS * G5
C
N=N-1
DO 558 JR=1,3
G3N(JR)= (YPRG1/YP1P1)*G3N(JR)
558 G4N(JR)= (YPRG2/YP1P2)*G4N(JR)
GO TO 557
273 CONTINUE
25 CONTINUE
50 CONTINUE
STOP
END

SUBROUTINE POLMUL(CON,COM,N,M,XCOF)
DIMENSION CON(1), COM(1), XCOF(1), CONA(50), COMRA(50)
DOUBLE PRECISION CON, COM, XCOF, CONA, COMRA
C
C THE VECTOR CON IS A VECTOR OF THE COEFFICIENT OF A POLYNOMIAL
OF ORDER N.

```

C THE VECTOR COM IS A VECTOR OF THE COEFFICIENTS OF A POLYNOMIAL OF
 C ORDER M.
 C THE VECTOR XCOF IS A VECTOR OF THE COEFFICIENTS OF THE PRODUCT OF
 C A POLYNOMIAL OF ORDER N AND A POLYNOMIAL OF ORDER M. THE
 C POLYNOMIAL OF WHICH THE COEFFICIENTS ARE THE VECTOR XCOF HAS AN
 C ORDER OF M + N.

```

C
C      DO 1 I=1,M
1 CONA(I)=0.0
  NX=N+1
  DO 2 I=1,NX
  LX=M+1
2 CONA(LX)=CON(I)
  MX=M+1
  DO 3 I=1,MX
  MY=M+2-I
3 COMRA(I)=COM(MY)
  DO 4 I=1,N
  NX=M+1+I
4 COMRA(NX)=0.0
  KY=M+N+1
  KX=KY
  DO 7 K=1,KY
  XCOF(K)=0.0
  DO 5 L=1,KX
5 XCOF(K)= CONA(L) * COMRA(L)+XCOF(K)
  KX=KX-1
  DO 6 J=1,KX
6 CONA(J)=CONA(J+1)
7 CONTINUE
  RETURN
  END

```

```

SUBROUTINE POLDIV(BX,AX,N,M,RPC)
DIMENSION AX(1), BX(1), RPC(1), DX(50)
DOUBLE PRECISION AX, BX, RPC, DX

```

C
 C POLYNOMIAL DIVISION
 C
 C MTH ORDER POLYNOMIAL WHOSE VECTOR OF COEFFICIENTS IS AX IS DIVIDED
 C BY AN NTH ORDER POLYNOMIAL WHOSE VECTOR OF COEFFICIENTS IS BX.
 C
 C ASSUMPTION-M IS GREATER THAN N.
 C

```

  BXA= BX(1)
  NX= N + 1
  MX= M + 1
  DO 1 I=1,NX
1 BX(I)=BX(I)/BXA
  DO 2 I=1,MX
2 AX(I)=AX(I)/BXA

```

C
 C AUGMENT BX
 LM=N + 2

```

      DO 3 I=LM,MX
3  BX(I)= 0.0
C
C
      JX=M-N + 1
      DO 5 I=1,JX
      RPC(I)=AX(I)
      DO 4 J=1,MX
4  DX(J)= AX(J) - BX(J) * RPC(I)
      MX= MX-1
      DO 5 L=1,MX
      LK= L+1
5  AX(L)= DX(LK)
      RETURN
      END

      SUBROUTINE HEAVY(NUM,DEN,M,N,ROOTR,ROOTI,ROOTC,CONST)
      REAL NUMR*8(50), NUM*8(1)
      DIMENSION DEN(1), ROOTR(1), ROOTI(1), CONST(L), ROOTC(L), DENR(50)
1  ,CONN(50), COND(50)
      DOUBLE PRECISION DENR, DEN,AE
      COMPLEX*16 CONST, ROOTC, CMPLX*8, CONN, COND
      COMPLEX STAND, BYE
C
C      FOR EXPANDING A RATIONAL POLYNOMIAL BY PARTIAL FRACTION EXPANSION
C
C      LIMITATIONS
C      1. ROOTS OF THE DENOMINATOR OF THE POLYNOMIAL MUST BE KNOWN.
C      2. IF ALL ROOTS ARE NOT DISTINCT, THEN A SMALL NUMBER IS ADDED
C          TO THE ROOTS NOT DISTINCT TO MAKE THEM DISTINCT.
C
C      DESCRIPTION OF VARIABLES
C
C      1. NUM- VECTOR OF COEFFICIENTS OF NUMERATOR OF POLY.
C      2. DEN- VECTOR OF DENOMINATOR COEFFICIENTS OF POLY.
C      3. ROOTR AND ROOTI -VECTORS OF REAL AND IMAGINARY PARTS OF ROOTS
C          OF THE DEN.
C      4. CONST- PARTIAL FRACTION EXPANSION CONSTANTS
C      5. M- ORDER OF NUM
C      6. N- ORDER OF DEN
C
      MA=M+1
      DO 10 I=1,MA
      MB=M+2-I
10  NUMR(I)=NUM(MB)
      NA=N+1
      DO 20 I=1,NA
      NB=N+2-I
20  DENR(I)=DEN(NB)
C
      DO 1 I=1,N
1  ROOTC(I)=CMPLX(ROOTR(I),ROOTI(I))

```

```
K=M+1
L=N+1
DO 6 I=1,N
  CONN(I)=0.0
  COND(I)=0.0
12 CONTINUE
  BYE= ROOTC(I)
  IF(REAL(BYE).EQ.0.0)GO TO 8
  GO TO 9
8 IF(AIMAG(BYE).NE.0.0)GO TO 9
  CONN(I)=NUMR(1)
  COND(I)=DENR(2)
  GO TO 11
9 CONTINUE
  DO 2 J=1,K
2 CONN(I)= CONN(I) + NUMR(J) * ROOTC(I)**(J-1)
  DO 3 J=2,L
  AE= J-1
3 COND(I)= COND(I)+ DENR(J) * AE * ROOTC(I) **(J-2)
11 STAND= COND(I)
  IF(REAL(STAND).EQ.0.0)GO TO 7
  GO TO 4
7 IF(AIMAG(STAND).NE.0.0)GO TO 4
  ROOTC(I)= ROOTC(I) + 0.10E-05
  GO TO 12
4 CONST(I)= CONN(I)/ COND(I)
  GO TO 6
5 IER= 8
6 CONTINUE
  RETURN
  END
```

APPENDIX B

OPEN LOOP POLE MOVEMENT PROGRAM

The following algorithm is used in order to determine the movement of the first and second bending modes, the second slosh mode, the drift, and the rigid body poles in the z-plane when the sampling period is varied. This program uses the root information from the s-plane program in Appendix A. Using the expression $z=e^{-sT}$ the s-plane poles can be mapped directly into the z-plane for a particular sampling period. (It should be understood that this cannot be done for s-plane zeros.) This simple program performs this task for the given poles.

```

DIMENSION TM(10)
COMPLEX SPOLE(2),ZPOLE(20), SPOLEX(20)
KX=13
TM(1)= 40.0
TM(2)= 80.11
TM(3)= 120.0
DO 12 M=1,3
IF(M.GT.1)KX=9
READ(1,1)(SPOLE(I),I=1,KX)
1 FORMAT(6F10.6)
WRITE(3,20) TM(M)
20 FORMAT('1',20X,'FLIGHT TIME = ',F5.2)
WRITE (3,4)
T=0.0
IF(M.GT.1)GO TO 21
DO 5K=1,100
CALL ZTRAN(SPOLE,T,4,ZPOLE)
WRITE(3,3) T,(ZPOLE(J),J=1,4)
3 FORMAT(1X,F5.2,8F10.5)
4 FORMAT('0',34X,'SERVO Z-PLANE POLES')
T=T+0.01
5 CONTINUE
21 CONTINUE
T=0.0
WRITE(3,6)
6 FORMAT('0',48X,'SLOSHING Z-PLANE POLES')
DO 9 K=1,1000
L=0
IF(M.EQ.1)L=4
DO 7 J=1,5

```



```
7 SPOLEX(J)= SPOLE(J+L)
  CALL ZTRAN(SPOLEX,T,5,ZPOLE)
  WRITE(3,8) T,(ZPOLE(I),I=1,5)
8 FORMAT(1X,F5.2,10E11.4)
  T=T+0.01
9 CONTINUE
  T=0.0
  WRITE(3,11)
  DO 12 J=1,1000
  L2=5
  IF(M.EQ.1)L2=9
  DO 10 I=1,4
10 SPOLEX(I)= SPOLE(I+L2)
  CALL ZTRAN(SPOLEX,T,4,ZPOLE)
  WRITE(3,3) T,(ZPOLE(I),I=1,4)
11 FORMAT('0',33X,'BENDING Z-PLANE POLES')
  T=T+0.01
12 CONTINUE
  STOP
  END
  SUBROUTINE ZTRAN(SPOLEX,T,N,ZPOLE)
  COMPLEX ZPOLE(20),CEXP, SPOLEX(20)
  DO 1 I=1,N
1 ZPOLE(I)= CEXP(SPOLEX(I) * T)
  RETURN
  END
```

APPENDIX C

OPEN LOOP Z-PLANE TRANSFER FUNCTION PROGRAM

The following computer program with the aid of the subroutines finds the z-plane open loop transfer function from information containing the roots and partial fractions expansion coefficients of the open loop s-plane transfer function. This algorithm takes into account a zero order hold when the $1/s$ term of the hold device has been incorporated in the partial fraction expansion information (that is, the partial fraction of $G(s)/s$ was found, rather than of $G(s)$, where $G(s)$ is the s-plane open loop transfer function). With the root and partial fraction information for a particular flight time, a z-plane open loop transfer function for any sampling period can be obtained when this program is employed.

Definitions of input and output variables of this routine are:

- A. 1. REXD - Vector of the real parts of the s-plane poles.
- 2. AEXD - Vector of corresponding imaginary parts of s-plane poles.
- 3. RECON1 and AICON1 - The real and imaginary parts respectively of the partial fraction expansion coefficients.

These must be in the order as their corresponding poles.

- 4. TF - Flight time*

B. Output Variables

- 1. ANUMR - A vector of the numerator coefficients of the z-plane open loop transfer function.

*This is both an input and an output variable.

2. DENOM - A vector of the denominator coefficients of the z-plane open loop transfer functions.**

```

DIMENSION REXD(20), AIXD(20), RECON1(20), RECON2(20), A(20),
2 B(20), C(20), D(20), ANC(20,20), ADC(20,20), TCN(20,20)
3 , XDB(20), CHART(20), ANUMR(20), DENOM(20), AICON1(20),
4 AICON2(20)
DOUBLE PRECISION REXD, AIXD, RECON1, RECON2, A, B, C, D, ANC, ADC,
2 TCN, XDB, CHART, AICON1, AICON2
GAIN= 1.0
DO 10 J0=1,8
T=0.0
READ(1,5) TF
5 FORMAT(13X,F10.5)
WRITE(3,20) TF
WRITE(2,500) TF
20 FORMAT('1', 10X, 'FLIGHT TIME = ', F10.5, 'SECONDS')
500 FORMAT('FLIGHT TIME = ', F10.5, ' SECONDS')
DO 2 I=1,14
READ(1,1) REXD(I), AIXD(I), RECON1(I), AICON1(I)
READ(1,3) RECON2(I), AICON2(I)
1 FORMAT(2D16.8,2D24.16)
3 FORMAT(2D24.16)
2 CONTINUE
DO 4 I=1,14
A(I)= RECON1(I) .+ RECON2(I)
4 B(I)= AICON1(I) + AICON2(I)
WRITE(3,401) (REXD(I),AIXD(I),A(I),B(I),I=1,14)
WRITE(2,502) (REXD(I),AIXD(I),A(I),B(I),I=1,14)
501 FORMAT('0', 2D16.8,2D24.16)
502 FORMAT(2D16.8,2D24.16)
DO 10 J=1,100
WRITE(3,503) T
WRITE(2,504) T
503 FORMAT('0', 20X, 'SAMPLING PERIOD = ', F6.4)
504 FORMAT('SAMPLING PERIOD = ', F6.4)
DO 120 I=1,14
C(I)= -DEXP(REXD(I)*T) * DCOS(AIXD(I)*T)
120 D(I)= -DEXP(REXD(I)*T) * DSIN(AIXD(I)*T)
IQ= 14
I=1
JR=0
150 CONTINUE
JR= JR + 1
IF(AIXD(I))155,160,155
155 ANC(JR,1)= 2.0 * A(I)

```

** ANUMR and DENOM are ordered normally as defined on page 97.

```

    ANC(JR,2)= 2.0 * (A(I) * C(I) + B(I) * D(I))
    ADC(JR,1)= 1.0
    ADC(JR,2)= 2.0
    ADC(JR,3)= C(I) * C(I) + D(I) * D(I)
    I= I + 2
    IF(I-IQ)150,150,180
160 ANC(JR,1)= A(I)
    ANC(JR,2)= 0.0
    ADC(JR,1)= 1.0
    ADC(JR,2)= C(I)
    ADC(JR,3)= 0.0
    I= I + 1
    IF(I-IQ)150,150,180
180 CONTINUE
    CALL HELP(ADC,ANC,JR,TCN,XDB,K)
    DO 6 IZ=1,14
    CHART(IZ)= 0.0
    DO 6 KZ=1,JR
    6 CHART(IZ)= CHART(IZ) + TCN(KZ,IZ) * GAIN
    DO 99 I=1,14
    DENOM(I)= XDB(I)
    99 ANUMR(I)= CHART(I)
    WRITE(3,101) (ANUMR(I),I=1,14)
    WRITE(3,101) (DENOM(I),I=1,14)
    WRITE(2,100) (ANUMR(I),I=1,14)
    WRITE(2,1000) (DENOM(I),I=1,14)
101 FORMAT('0',5E16.8)
100 FORMAT(5E16.8)
    T= T + 0.01
    10 CONTINUE
    STOP
    END

SUBROUTINE HELP(XD,XN,LA,TCN,XDB,K)
DIMENSION XDB(20), XDA(20), XD(20,20), COM(20), XN(20,20)
1 CON(20), TCN(20,20), XCOF(20)
DOUBLE PRECISION XDB, XDA, XD, COM, XN, CON, TCN, XCOF
XDB(1)= XD(2,1)
XDB(2)= XD(2,2)
XDB(3)= XD(2,3)
XDA(1)= XD(1,1)
XDA(2)= XD(1,2)
XDA(3)= XD(1,3)
DO 23 M=1,LA
K=1
N=1
DO 20 I=1,2
20 COM(I)= XN(M,I)
DO 21 J=1,LA
DO 13 I=1,3
13 CON(I)=XD(J,I)
IF(J-M)15,21,15
15 N=1
IF(CON(3))16,17,16
16 N=2

```

```

17 CALL POLMC(CON,COM,N,K,XCOF)
   K=K+N
   KZ=K+1
   DO 30 I=1,KZ
30 COM(I)= XCOF(I)
21 CONTINUE
   DO 22 I=1,KZ
   TCN(M,I)= COM(I)
22 CONTINUE
23 CONTINUE
   L=1
   IF(XDB(3))40,41,40
40 L=2
41 IF(LA)43,26,43
43 LM=LA-2
   DO 25 I=1,LM
   N=1
   IF(XDA(3))44,45,44
44 N=2
45 CALL POLMC(XDA,XDB,N,L,XCOF)
   L=L+N
   JX=L+1
   DO 31 NL=1,JX
31 XDB(NL)= XCOF(NL)
   IF(I-LM) 33,25,25
33 KX= I + 2
   DO 24 NX=1,3
24 XDA(NX)= XD(KX,NX)
25 CONTINUE
26 RETURN
   END

```

```

SUBROUTINE POLMC(CON,COM,N,M,XCOF)
DIMENSION CON(1), COM(1), XCOF(1), CONA(50), COMRA(50)
DOUBLE PRECISION CON, COM, XCOF, CONA, COMRA

```

```

C
C THE VECTOR CON IS A VECTOR OF THE COEFFICIENT OF A POLYNOMIAL
C OF ORDER N.
C THE VECTOR COM IS A VECTOR OF THE COEFFICIENTS OF A POLYNOMIAL OF
C ORDER M.
C THE VECTOR XCOF IS A VECTOR OF THE COEFFICIENTS OF THE PRODUCT OF
C A POLYNOMIAL OF ORDER N AND A POLYNOMIAL OF ORDER M. THE
C POLYNOMIAL OF WHICH THE COEFFICIENTS ARE THE VECTOR XCOF HAS AN
C ORDER OF M + N
C

```

```

   DO 1 I=1,M
1 CONA(I)=0.0
   NX=N+1
   DO 2 I=1,NX
   LX=M+I
2 CONA(LX)=CON(I)

```

```
MX=M+1
DO 3 I=1,MX
MY=M+2-I
3 COMRA(I)=COM(MY)
DO 4 I=1,N
NX=M+1+I
4 COMRA(NX)=0.0
KY=M+N+1
KX=KY
DO 7 K=1,KY
XCOF(K)=0.0
DO 5 L=1,KX
5 XCOF(K)=CONA(L) * COMRA(L)+XCOF(K)
KX=KX-1
DO 6 J=1,KX
6 CONA(J)=CONA(J+1)
7 CONTINUE
RETURN
END
```

APPENDIX D

MODIFIED ROOT LOCUS PROGRAM AND THEORY

The following computer program, which is comprised of a main program and two subroutines, is the program used for obtaining the roots for the modified root locus method. The purpose of the main program is simply input, output, and control. The input and output variables are interpreted as follows:

1. TF - Flight time
2. YD - The complex s-plane roots
3. CONST - The complex partial fraction coefficients of the open loop s-plane transfer function
4. T - Sampling period
5. GAIN - Open loop gain of the system
6. ANUMR - Vector of open loop, numerator, z-plane coefficients in normal order.
7. DENOM - Vector of open loop, denominator z-plane coefficients in normal order.
8. STARTP - Vector of predicted starting points for finding roots
9. ROOT - Vectors of roots of closed loop transfer function
10. ASRT - The absolute value of a root for determining if it is outside the unit circle.

After the roots of the closed loop, z-plane, characteristic equation have been found this program detects if any are outside the unit circle; if none are outside the unit circle, it indicates that the system is stable. If some are outside the unit circle, it denotes the system as being unstable.

The subroutine PRDZ is sub-program for predicting another root of a modified root locus from knowledge of a previous root on the locus. The theoretical background for this program is presented in the following discussion.

The characteristic equation of a typical feedback control system is of the form

$$1 \pm G(s,T) = 0 \quad (\text{A-1})$$

where $G(s,T)$ is the open loop transfer function and T is the parameter for which a locus is desired. The necessary and sufficient conditions for a point, s_o , to be a point on the locus are

$$|G(s_o,T)| = 1 \quad (\text{A-2})$$

and

$$\angle G(s_o,T) = n\pi \quad (\text{A-3})$$

where

$$n = 1,3,5 \dots \text{ for negative feedback}$$

$$n = 0,2,4, \dots \text{ for positive feedback.}$$

The open loop transfer function can be written as

$$G(x,y,T) = G_R(x,y,T) + jG_I(x,y,T) \quad (\text{A-4})$$

in which $G_R(x,y,T)$ and $G_I(x,y,T)$ are the real and imaginary parts of $G(x,y,T)$ respectively, and x and y are the real and imaginary parts of the complex variable s . The magnitude and angle relations of (A-4) are

$$\sqrt{[G_R(x,y,T)]^2 + [G_I(x,y,T)]^2} = |G(x,y,T)| \quad (\text{A-5})$$

and

$$\frac{G_I(x,y,T)}{G_R(x,y,T)} = \tan \theta \quad (\text{A-6})$$

Letting $F_1(x,y,T) = \tan \theta$, (A-6) is rewritten as

$$F_1(x,y,T) = \frac{G_I(x,y,T)}{G_R(x,y,T)} \quad (\text{A-7})$$

With $F_2(x,y,T) = |G(x,y,T)|^2$ then

$$F_2(x,y,T) = [G_R(x,y,T)]^2 + [G_I(x,y,T)]^2 \quad (\text{A-8})$$

Equations (A-7) and (A-8) represent two hypersurfaces which are functions of the three variables x , y , and T . The manifold of these hypersurfaces is composed of several lines, namely the three dimensional lines of the root loci. A point on a locus, is a point common to both surfaces. It is desired to find a line which is tangent to the line of intersection of these two surfaces. This is done by first finding the gradients of the two surfaces. These are

$$\nabla F_1 = \frac{\partial F_1}{\partial x} \vec{i} + \frac{\partial F_1}{\partial y} \vec{j} + \frac{\partial F_1}{\partial T} \vec{k} \quad (\text{A-9})$$

and

$$\nabla F_2 = \frac{\partial F_2}{\partial x} \vec{i} + \frac{\partial F_2}{\partial y} \vec{j} + \frac{\partial F_2}{\partial T} \vec{k} \quad (\text{A-10})$$

where \vec{i} , \vec{j} , and \vec{k} are unit vectors.

Taking the cross product of the two gradients results in

$$\begin{aligned} \nabla F_1 \times \nabla F_2 = & \left[\frac{\partial F_1}{\partial y} \frac{\partial F_2}{\partial T} - \frac{\partial F_2}{\partial y} \frac{\partial F_1}{\partial T} \right] \vec{i} \\ & + \left[\frac{\partial F_2}{\partial x} \frac{\partial F_1}{\partial T} - \frac{\partial F_1}{\partial x} \frac{\partial F_2}{\partial T} \right] \vec{j} + \left[\frac{\partial F_1}{\partial x} \frac{\partial F_2}{\partial y} - \frac{\partial F_2}{\partial y} \frac{\partial F_1}{\partial x} \right] \vec{k} \quad (\text{A-11}) \end{aligned}$$

Equation (A-11) is a vector which lies in the tangent plane of the manifold. From (A-11) a straight line tangent to the intersection of the surfaces can be obtained at any point on the manifold. A geometrical

interpretation of the preceding derivation can be gathered from Figure 69.

Now, further information regarding the partial derivatives will be obtained. First, consideration will be given to the three partials of $F_1(x,y,T)$. The partial of $F_1(x,y,T)$ with respect to x is

$$\frac{\partial F_1(x,y,T)}{\partial x} = \frac{G_R(x,y,T) \frac{\partial G_I(x,y,T)}{\partial x} - G_I(x,y,T) \frac{\partial G_R(x,y,T)}{\partial x}}{[G_R(x,y,T)]^2} \quad (A-12)$$

If this is evaluated at a point on the locus (x_1, y_1, T_1) , it reduces to

$$\left. \frac{\partial F_1(x,y,T)}{\partial x} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = \left. \frac{\frac{\partial G_I(x,y,T)}{\partial x}}{G_R(x,y,T)} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} \quad (A-13)$$

Similarly, the partials with respect to y and T evaluated at a point on the locus are

$$\left. \frac{\partial F_1}{\partial y} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = \left. \frac{\frac{\partial G_I}{\partial y}}{G_R} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} \quad (A-14)$$

and

$$\left. \frac{\partial F_1}{\partial T} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = \left. \frac{\frac{\partial G_I}{\partial T}}{G_R} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} \quad (A-15)$$

where it should be understood that G_R , G_I , and F_1 are functions of x , y , and T . By a similar analysis the partials of F_2 are derived as

$$\left. \frac{\partial F_2}{\partial x} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = 2G_R \left. \frac{\partial G_R}{\partial x} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}}, \quad (\text{A-16})$$

$$\left. \frac{\partial F_2}{\partial y} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = 2G_R \left. \frac{\partial G_R}{\partial y} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}}, \quad (\text{A-17})$$

and

$$\left. \frac{\partial F_2}{\partial T} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}} = 2G_R \left. \frac{\partial G_R}{\partial T} \right|_{\substack{x=x_1 \\ y=y_1 \\ T=T_1}}. \quad (\text{A-18})$$

Again, it should be understood that F_2 and G_R are functions of x , y , and T . Furthermore, from here on it should be understood that the partials and functions are evaluated at a point on the locus. Therefore, for convenience, the evaluation notation will be dropped.

Although the criterion for determining a straight line in the direction of the intersection of the surfaces was derived using the gradients of the surfaces, it is not really necessary to have the gradients. All that is necessary is to have vectors in the direction of the gradients. For this reason the factor $1/G_R$ can be dropped from the partials of F_1 , and the factor $2G_R$ can be dropped from the partials of F_2 . Since the result is no longer the true gradients but scaled versions, a prime will be used in the gradient notation. Thus, the scaled gradients are

$$\nabla F_1' = \frac{\partial G_I}{\partial x} \mathbf{i} + \frac{\partial G_I}{\partial y} \mathbf{j} + \frac{\partial G_I}{\partial T} \mathbf{k} \quad (\text{A-19})$$

and

$$\nabla F_2' = \frac{\partial G_R}{\partial x} \vec{i} + \frac{\partial G_R}{\partial y} \vec{j} + \frac{\partial G_R}{\partial t} \vec{k} . \quad (\text{A-20})$$

From (A-19) and (A-20) one can observe that the problem reduces to considering the intersection of the two surfaces formed by the real and imaginary parts of $G(s,T)$. In fact, from the beginning one would intuitively think that this would be the result. However, in order to maintain as much mathematical rigor as possible the magnitude and angle approach was chosen as the initial point of attack.

Under certain circumstances, it is necessary to still think of the gradients as being those of the magnitude and angle relation. For example, in a regular root locus the angle relation is completely independent of the open loop gain. Thus the partial of the angle function with respect to open loop gain is zero. This result is not obvious from (A-19) and (A-20), but is easily deduced from the magnitude and angle functions (A-7) and (A-8).

Returning to (A-19) and (A-20) it is now possible to obtain a straight line which is tangent to a locus. This is easily done, and the result is

$$\frac{\Delta x}{\frac{\partial G_I}{\partial y} \frac{\partial G_R}{\partial T} - \frac{\partial G_R}{\partial y} \frac{\partial G_I}{\partial T}} = \frac{\Delta y}{\frac{\partial G_R}{\partial x} \frac{\partial G_I}{\partial T} - \frac{\partial G_I}{\partial x} \frac{\partial G_R}{\partial T}} = \frac{\Delta T}{\frac{\partial G_I}{\partial x} \frac{\partial G_R}{\partial y} - \frac{\partial G_I}{\partial y} \frac{\partial G_R}{\partial x}} \quad (\text{A-21})$$

where Δx , Δy , and ΔT are $(x - x_1)$, $(y - y_1)$, and $(T - T_1)$ respectively and x_1 , y_1 , and T_1 are points on a locus.

From (A-21) it is possible to predict a point $(x + jy)$ on a locus for some increment, ΔT , of the parameter for which the locus is being made. For small values of ΔT the predicted point will be very accurate.⁴

The preceding method for predicting points on a locus is only applicable if the partial derivatives can be found. For a sampled-data

system this is an easy task if the open loop transfer function is in the partial fraction expansion form. If this is the case, then the open loop transfer function is of the form

$$G(z) = K \left[\frac{C_1 z}{z - e^{p_1 T}} + \frac{C_2 z}{z - e^{p_2 T}} + \dots \right], \quad (\text{A-22})$$

where the C's are the partial fraction coefficients, the p's are the poles of the s-plane open loop transfer function, and K is the open loop gain.* Letting $G_n(z)$ be the nth term inside the parentheses of (A-22) and letting the complex variable $z = x + jy$

$$G_n(z) = \frac{C_n(x + jy)}{(x + jy) - (e^{\sigma_n T} \cos \omega_n T + j e^{\sigma_n T} \sin \omega_n T)} \quad (\text{A-23})$$

where σ and ω_n are the real and imaginary parts respectively of the p_n in (A-22). After rationalizing (A-23), it can be separated into its real and imaginary parts. This allows for the real and imaginary parts of (A-22) to be written respectively as the sum of the real and imaginary parts of the individual terms of (A-22). Thus,

$$G_R(z) = K \sum_{i=1}^M GR_i, \quad (\text{A-24})$$

and

$$G_I(z) = K \sum_{i=1}^M GI_i, \quad (\text{A-25})$$

in which GR_i is the real part of the ith term of (A-22), GI_i is the imaginary part of the ith terms of (A-22), and M is the number of terms in (A-22). Since a partial of a sum is equal to the sum of the partials, it is obvious how the necessary partials derivatives can be obtained from (A-24) and (A-25).

*If a zero order hold has been included, then the z in the numerator must be changed to a z + 1.

The subroutine POLSER is an algorithm for finding the roots of a polynomial. The steps it uses in computation are as follows:

1. Set a starting point
2. Using the steepest-descent method, find a good approximation to a root
3. Taking the approximation, use the Newton-Raphson technique to refine the approximation until it is within some preassigned ϵ of the exact value.
4. If the root is complex, set the next root equal to the complex conjugate of this root.
5. Find a reduced polynomial by factoring out the term involving the root or roots.
6. With the reduced polynomial return to step 1 and continue the process until all roots have been found.

As was mentioned in the procedure of computation, the steepest-descent method is used for finding approximations to the roots. This method is well known for finding maxima and minima of a function, but it has had little application for finding roots of an equation. Applying this method to root extraction for a polynomial of a complex variable is very simple. First, find the absolute value squared of the polynomial. This new function will be a real function that is a function of two real variables, x and y , which are the real and imaginary parts respectively of the independent variable of the original polynomial. If the original polynomial was of order n , this function will have n absolute minima which will occur at the x and y corresponding to the roots of the polynomial. These minima can easily be found using the steepest-descent

method if they are the only minima of the function, and, in fact, they are the only minima. A proof of this is stated and proved in the following theorem.

THEOREM - The absolute value squared of an n th order polynomial has n absolute minima, and they are located at the real and imaginary parts of the roots of the polynomial. All other extremals are saddle points.

PROOF - Let $s = x + jy$ where x and y are real variables. Suppose that F is an n th order polynomial in s such that

$$F(s) = s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_0 \quad (\text{A-26})$$

Separating F into its real and imaginary parts (A-26) becomes

$$F(x,y) = R(x,y) + j I(x,y) \quad (\text{A-27})$$

where R is the real part of F and I is the imaginary part of F . Since a polynomial is an analytic function, the Cauchy-Rieman equations state that

$$\frac{\partial R}{\partial x} = \frac{\partial I}{\partial y}$$

and

$$\frac{\partial I}{\partial x} = -\frac{\partial R}{\partial y} \quad , \quad (\text{A-28a,b})$$

where for brevity the function notation has been dropped and for convenience will be dropped in all remaining equations of this section. The absolute value squared of (A-27) is

$$f = R^2 + I^2 \quad . \quad (\text{A-29})$$

The necessary conditions for an extremal of (A-29) to occur is for $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$. Taking these partials and setting them equal to zero, the results are

$$R \frac{\partial R}{\partial x} = -I \frac{\partial I}{\partial x}$$

and

$$R \frac{\partial R}{\partial y} = -I \frac{\partial I}{\partial y} \quad . \quad (A-30a,b)$$

An obvious solution to these equations is the x and y so that $R = 0$ and $I = 0$. The x and y for which this occurs is simply the real and imaginary parts of the roots of the original polynomial. Using the Cauchy-Riemann equation, the other solutions of (A-30a,b) would occur for the x and y so that

$$\frac{\partial R}{\partial y} = \frac{\partial I}{\partial x} = 0$$

or

$$\frac{\partial I}{\partial y} = \frac{\partial R}{\partial x} = 0 \quad . \quad (A-31a,b)$$

Equations (A-31a,b) could occur when $R = 0$ and $I = 0$. In fact, this is one case. However, (A-31a,b) can also occur when $R \neq 0$ and $I \neq 0$. The type of these extremals (maximum, minimum, or saddle point) must be determined. In order to find if these extremals are maxima, minima, or saddle points all the second partials of f must be obtained so that sign of $f_{xy}^2 - f_{xx} f_{yy}$ can be investigated. The symbols f_{xy} , f_{xx} , and f_{yy} are respectively

$$\frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial^2 f}{\partial x^2}, \quad \text{and} \quad \frac{\partial^2 f}{\partial y^2} \quad .$$

These second partials evaluated at $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$ are

$$\frac{\partial^2 f}{\partial x^2} = R \frac{\partial^2 R}{\partial x^2} + I \frac{\partial^2 I}{\partial x^2} \quad ,$$

$$\frac{\partial^2 f}{\partial y^2} = R \frac{\partial^2 R}{\partial y^2} + I \frac{\partial^2 I}{\partial y^2} \quad ,$$

and

$$\frac{\partial^2 f}{\partial x \partial y} = R \frac{\partial^2 R}{\partial x \partial y} + I \frac{\partial^2 I}{\partial x \partial y} \quad . \quad (A-32a,b,c)$$

Therefore,

$$\begin{aligned}
 f_{xy}^2 - f_{xx} f_{yy} &= R^2 \frac{\partial^2 R}{\partial x \partial y}^2 + 2RI \frac{\partial^2 R}{\partial x \partial y} \frac{\partial^2 I}{\partial x \partial y} + \\
 I^2 \frac{\partial^2 I}{\partial x \partial y} &- \left[R^2 \frac{\partial^2 R}{\partial x^2} \frac{\partial^2 R}{\partial y^2} + RI \frac{\partial^2 R}{\partial x^2} \frac{\partial^2 I}{\partial y^2} + \right. \\
 &\left. RI \frac{\partial^2 I}{\partial x^2} \frac{\partial^2 R}{\partial y^2} + I^2 \frac{\partial^2 I}{\partial x^2} \frac{\partial^2 I}{\partial y^2} \right] . \quad (A-33)
 \end{aligned}$$

Using the Cauchy-Rieman equations and letting

$$A = \frac{\partial^2 R}{\partial x \partial y}$$

and

$$B = \frac{\partial^2 I}{\partial x \partial y} \quad (A-34a,b)$$

(A-33) becomes

$$R^2 A^2 + I^2 B^2 + R^2 B^2 + I^2 A^2 = (R^2 + I^2)(A^2 + B^2) \quad (A-35)$$

or

$$(R^2 + I^2)(A^2 + B^2) > 0 \quad (A-36)$$

if A or B is non-zero. For this case all the extremals are saddle points. By considering the higher derivatives and the geometry of the problem the same result can be extended to include the case where

$A = B = 0$.

```

DIMENSION CHEQ(20), ANUMR(20), DENOM(20)
COMPLEX XD(20,20), CONST*16(20,20), YD*16(20,20), STARTP(20),
1 ROOT(20), ZNEW, ZOLD, CMLPX, CABS, CONJG
COMPLEX ZSTART(20)
CALL INTERD
DO 100 JO=1,8
READ(1,1) TF
WRITE(3,5) TF
1 FORMAT(14X,F10.5)
5 FORMAT('1',35X,'FLIGHT TIME =',F10.5)
DO 6 I=1,13
ROOT(I)= 1.0

```

```

6  STARTP(I)= 1.0
   DO 3 I=1,14
     READ(1,2) YD(I,2), CONST(I,1)
2  FORMAT(2D16.8,2D24.16)
     XD(I,1)= 1.0
     YD(I,1)= 1.0
     CONST(I,2)= 0.0
3  CONTINUE
   DO 4 I=1,14
4  XD(I,2)= YD(I,2)
     DO 100 J=1,50
       LP=13
       IER= 0
       DELT= 0.01
       READ(1,10) T
       READ(1,12) (ANUMR(I),I=1,14)
       READ(1,12) (DENOM(I),I=1,14)
10  FORMAT(17X,F6.4)
12  FORMAT(5E16.8)
13  CONTINUE
     GAIN=-1.0
     DO 15 I=1,14
15  CHEQ(I)= DENOM*I)+ GAIN * ANUMR(I)
       DO 19 I=1,LP
         CHECK= ABS(CHEQ(15-I)/CHEQ(14-I))
         IF(CHECK-1.0E-04)17,17,21
17  CHEQ(15-I)=0.0
         LP=LP - 1
         ROOT(14-I)= 0.0
         STARTP(14-I)= 0.0
19  CONTINUE
21  CONTINUE
       I=1
16  CONTINUE
       ZOLD= ROOT(I)
       CALL PRDZ(ZOLD,ZNEW,14,T,IER,DELT,GAIN,CONST,XD,UI,UJ,UK)
       IF(IER.NE.0)ZNEW= ROOT(I)
       STARTP(I)= ZNEW
       IF(AIMAG(ROOT(I)))20,18,20
18  I=I+1
       IF(I-LP)16,16,25
20  STARTP(I+1);CONJG(ZNEW)
       I=I+2
       IF(I-LP)16,16,25
25  CONTINUE
       CALL POLSER(CHEQ,ROOT,IER,IEND,LP,STARTP)
       IF(IER.LT.98)GO TO 35
       WRITE(3,28) IER
28  FORMAT('0', 10X,' IER = ',I5)
       GO TO 100
35  IF(IEND.LT.98)GO TO 40
       WRITE(3,38) IEND

```

```

38 FORMAT('0',10X,'IEND =',15)
   GO TO 100
40 CONTINUE
   WRITE(3,45) T
45 FORMAT('0',12X,'SAMPLING PERIOD = ',F6.4)
   WRITE(3,50)
50 FORMAT('0',2X,'PREDICTED ROOTS',21X,'ROOTS',14X,'MAGNITUDES OF ROD
1 TS')
   KS=0
   DO 65 I=1,13
     ABSRT= CABS(ROOT(I))
     WRITE(3,55) STARTP(I), ROOT(I), ABSRT
55 FORMAT('0',2F10.5,10X,2F10.5,10X,F11.8)
     IF(ABSRT.GE.1.0)KS=1
65 CONTINUE
   IF(KS.LT.1)GO TO 80
   WRITE(3,70)
70 FORMAT('0',30X,'*** SYSTEM UNSTABLE ***')
   GO TO 95
80 WRITE(3,90)
90 FORMAT('0',31X,'*** SYSTEM STABLE ***')
95 CONTINUE
100 CONTINUE
110 CONTINUE
   STOP END

```

```

SUBROUTINE PRDZ(ROOT,ZPRED,N,T,IER,DT,K,CONST,XD,UI,UJ,UK)

```

C
C
C

```

PREDICT Z

```

```

COMPLEX CABS,CEXP,CONST*16(20,20),XD(20,20),SON
COMPLEX*16 ROOT*8,ZPRED*8,ZX,CDABS,ZY,ZT,CMLPX*8,SUM
DOUBLE PRECISION PRNX,PRNY,PRNT,PINX,PINY,PINT,PDNX,PDNY,PDNT,
1 DN,RN,AIN,PRIX,PRIY,PRIT,PIIX,PIIY,PIIT,PGX,PGY,PGT,PIGX,
2 PIGY,PIGT
REAL K
IER=0
PGX=0.0
PGY=0.0
PGT=0.0
PIGX=0.0
PIGY=0.0
PIGT=0.0
X= REAL(ROOT)
Y= AIMAG(ROOT)
DO 3 I=1,N
SON= CONST(I,1)
D= REAL(SON)
E= AIMAG(SON)
E= E* K
D= D * K
OMEGA= -AIMAG(XD(I,2))
A= - REAL(XD(I,2))

```

```

ACOS= COS(+OMEGA * T)
ASIN= SIN(+OMEGA * T)
AEXP= EXP(-A * T)
4 PRNX= 2.0 * X - AEXP * ACOS - 1.0
  PRNY= 2.0 * Y + AEXP * ASIN
  PRNT= A * AEXP * ((X - 1.0) * ACOS - Y * ASIN) + OMEGA * AEXP *
1 ((X - 1.0) * ASIN + Y * ACOS)
  PINX=-AEXP * ASIN
  PINY=-AEXP * ACOS + 1.0
  PINT= A * AEXP * (Y * ACOS + (X - 1.0) * ASIN) - OMEGA * AEXP *
1 ((X - 1.0) * ACOS - Y * ASIN)
  PDNX= 2.0 * X - 2.0 * AEXP * ACOS
  PDNY= 2.0 * Y + 2.0 * AEXP * ASIN
  PDNT= 2.0 * A * AEXP * (X * ACOS - Y * ASIN) + 2.0 * OMEGA * AEXP *
1 (X * ASIN + Y * ACOS) - 2.0 * A * AEXP * AEXP
  DN=X*X + Y*Y - 2.0 * AEXP * (X * ACOS - Y * ASIN) + AEXP * AEXP
  IF(DN)7,6,7
6 Y= Y + 1.0E-06
  GO TO 4
7 RN= X*(X - 1.0) + Y*Y - AEXP * ((X - 1.0) * ACOS - Y * ASIN)
  AIN= Y - AEXP * ((X - 1.0) * ASIN + Y * ACOS)
  PRIX= (DN * PRNX - RN * PDNX)/(DN * DN)
  PRIY= (DN * PRNY - RN * PDNY)/(DN * DN)
  PRIT= (DN * PRNT - RN * PDNT)/(DN * DN)
  PIIX= (DN * PINX - AIN * PDNX)/(DN * DN)
  PIYI= (DN * PINY - AIN * PDNY)/(DN * DN)
  PIIT= (DN * PINT - AIN * PDNT)/(DN * DN)
  PGX = PGX + PRIX * D - E * PIIX
  PGY = PGY + PRIY * D - E * PIYI
  PGT = PGT + PRIT * D - E * PIIT
  PIGX= PIGX + PIIX * D + E * PRIX
  PIGY= PIGY + PIYI * D + E * PRIY
3 PIGT= PIGT + PIIT * D + E * PRIT
  ROOTR= REAL(ROOT)
  ROOTI= AIMAG(ROOT)
  UI= PGY * PIGT - PIGY * PGT
  UJ= PIGX * PGT - PGX * PIGT
  UK= PGX * PIGY - PIGX * PGY
  IF(UK.EQ.0.0)IER=1
  IF(IER.GT.0)GO TO 10
5 CONTINUE
  X= (UT/UK) * DT + ROOTR
  Y= (UJ/UK) * DT + ROOTI
  ZPRED= CMLX(X,Y)
10 CONTINUE
  RETURN
  END

```

```

SUBROUTINE POLSER(CONST,ROOT,IER,IEND,M,STARTP)
DIMENSION CONST(1), COF(20), YCONS(20)
DIMENSION RCOF(20)
DOUBLE PRECISION COF, YCONS, ALPHA,SUMSQ, RCOF
DOUBLE PRECISION FR,FI,PFRX,PFIX,PFRY,PFIY,FRO,FIO,PFRXO,
1 PFRYO,PFIXO,PFIYO,FR1,FI1,PFRX1,PFRY1,PFIY1,PFIX1,PFY,PFY
COMPLEX*16 SO,FUN, DERV, CDABS
COMPLEX S1,CMPLX, ROOT(20),CABS, STARTP(20), CONJG
N=M+1
DO 10 I=1,N
10 COF(I)= CONST(I)
J=1
5 IEND= 0
DO 6 I=1,N
6 YCONS(I)=COF(I)
X1= REAL(STARTP(J))
Y1= AIMAG(STARTP(J)) + 1.0E-04
ISKIP= 0
15 IEND= IEND + 1
X= X1
Y= Y1
DO 18 I=1,N
KJ=N + 1 - I
18 RCOF(I)= COF(KJ)
FR= RCOF(1)
FI= 0.0
PFRX= 0.0
PFIX= 0.0
PFRY= 0.0
PFIY= 0.0
FRO= X
FIO= Y
PFRXO= 1.0
PFIXO= 0.0
PFRYO= 0.0
PFIYO= 1.0
DO 25 I=2,N
PFRX= PFRX + RCOF(I) * PFRXO
PFIX= PFIX + RCOF(I) * PFIXO
PFRY= PFRY + RCOF(I) * PFRYO
PFIY= PFIY + RCOF(I) * PFIYO
FR= FR + RCOF(I) * FRO
FI= FI + RCOF(I) * FIO
FR1= X * FRO - Y * FIO
FI1= Y * FRO + X * FIO
PFRX1= FRO + X * PFRXO - Y * PFIXO
PFIX1= Y * PFRXO + FIO + X * PFIXO
PFRY1= X * PFRYO - FIO - Y * PFIYO
PFIY1= FRO + Y * PFRYO + X * PFIYO
PFRXO= PFRX1
PFIXO= PFIX1
PFRYO= PFRY1
PFIYO= PFIY1

```

```

      FRO= FR1
      FIO= FI1
25  CONTINUE
      PFX= 2.0 * ( FR * PFRX + FI * PFI )
      PFY= 2.0 * ( FR * PFRY + FI * PFI )
      ZAP=DSQRT( FR * FR + FI * FI )
      IER=0
      IF(ISKIP.EQ.0)ZAP= ZAP + 100000.0
      IF(ISKIP.EQ.0)GO TO 35
      IF(ZIP + ZIP/ZOP.GE.1.0)DELX= 0.30
      IF(ZIP + ZIP/ZOP.LT.1.00 )DELX= 0.20
      IF(ZIP + ZIP/ZOP.LT.0.15 )DELX= 0.10
      IF(ZIP + ZIP/ZOP.LT.0.05 )DELX= 0.05
      GO TO 27
26  DELX= DELX/2.0
27  DELX= ABS(DELX)
      IF(PFX.GT.0.0)DELX= -DELX
      DX= DELX
28  X1= X + DX
      IF(PFX.NE.0.0)GO TO 30
      Y1= Y - PFY * DX / DABS(PFY)
      X1= X1 - DX
      GO TO 35
30  Y1= (PFY/PFX) * DX + Y
      IF(ABS(Y1-Y) + ABS( DX ).LT.SQRT(2.0*DELX*DELX))GO TO 35
      DX=DX/2.0
      GO TO 28
35  CONTINUE
      ISKIP= 1
36  SO= CMLX(X1,Y1)
      IF(IER.GT.98)GO TO 50
      IER= IER + 1
      MONK=0
      JX= N-1
      NEND= 0
37  FUN= YCONS(1)
      DERV= 0.0
      NXX=JX+1
      DO 40 I=1,JX
      P= NXX - I
      DERV= DERV * SO + YCONS(I) * P
40  FUN= FUN * SO + YCONS(I+1)
      ZIP= CDABS(FUN)
      ZOP= CDABS(DERV)
      NEND= NEND + 1
      IF(NEND.GT.50)IEND=200
      IF(MONK.GT.0)GO TO 42
      IF(ZIP.GT.ZAP)GO TO 26
      IF(CDABS(FUN/DERV).GT.0.025)GO TO 15
42  IF(IEND.GT.98)GO TO 50
      IF(CDABS(FUN/DERV)+ ZIP.LT.1.0E-04)GO TO 43
      SO= SO - FUN/DERV
      GO TO 37

```

```
43 IF(MONK.NE.0)GO TO 45
   MONK= MONK+1
   IF(MONK.GT.5)IEND=100
   MX=M+1
   DO 44 I=1,MX
44 YCONS(I)= CONST(I)
   JX= M
   GO TO 37
45 ROOT(J)= S0
   S1= S0
   IF(ABS(AIMAG(ROOT(J))/REAL(ROOT(J))).LT.1.0E-04)ROOT(J)=CMPLX(
* REAL(S1),0.0)
   IF(AIMAG(ROOT(J)).EQ.0.0)GO TO 60
   ROOT(J+1)= CONJG(ROOT(J))
   J=J+2
   ALPHA= 2 * REAL(S1)
   SUMSQ= REAL(S1) * REAL(S1) + AIMAG(S1) * AIMAG(S1)
   N=N-2
   GO TO 140
60 J= J + 1
   SUMSQ= 0.0
   ALPHA= REAL(S1)
   N=N-1
140 COF(2)= COF(2) + ALPHA * COF(1)
   NX= N-1
145 DO 150 L=2,NX
150 COF(L+1)= COF(L+1) + ALPHA * COF(L) - SUMSQ * COF(L-1)
   IF(N-1)50,50,5
50 RETURN
   END
```

APPENDIX E

OPEN LOOP SAMPLED-DATA FREQUENCY RESPONSE PROGRAM

The ensuing digital routine is an open loop frequency response program for the simplified version of the sampled-data control system of the Saturn V/S-1C. The input variables are defined in the following list:

1. TF - Flight time
2. YD - The open loop s-plane roots
3. CONS1 - The partial fraction expansion coefficients of $sH_O W_{SS} (G_T + K_{10} G_3 + K_{11} G_4)$
4. CONS2 - The partial fraction expansion coefficients of $H_O W_{SS} (G_T + K_8 G_3 + K_9 G_4)$ which is also obtained from Figure 6.

All the preceding information is simply the output information of the program in Appendix A. The output information in this program is well defined on its print-outs.

The method of obtaining the frequency response is to take the z-transform form of the open loop transfer function and then plug in values of z that are on the unit circle. The general incrementation of the z's is accomplished by picking fifty evenly spaced points on the unit circle between $z = 1$ and $z = -1$. The program takes additional points if poles of the open loop transfer function are within some preassigned δ of the unit circle. The reason for this is to keep the printed out phase angles and magnitudes from changing abruptly.


```

COMPLEX GSIR
COMPLEX XD(20,20), CONST*16(20,20), YD*16(20,20)
COMPLEX*16 A(20),GSAM, Z
COMPLEX GSUM, CEXP, CMLX, ARGU, CABS, WROOT
COMPLEX*16 CONS1(20,20), CONS2(20,20)
DIMENSION WCRT(20)
COMPLEX GCOMPS

C
C FOR NO HOLD DEVICE H SHOULD EQUAL 0.0
C FOR A ZERO ORDER HOLD TO BE INCLUDED H SHOULD EQUAL I.
C
H= 1.0

C
C DECLARE ORDER OF SYSTEM(IF A HOLD DEVICE IS USED, IT MUST BE
C INCLUDED).
LP=14
DO 65 LTF=1,8

C
C SET STARTING VALUE OF T.
C
T=0.30

C
C SET OPEN LOOP GAIN
C
GAIN=-1.0
READ(1,1) TF
WRITE(3,5) IF
1 FORMAT(14X,F10.5)
5 FORMAT('1',30X,'FLIGHT TIME = ',F10.5)
DO 3 I=1,LP
READ(1,2) YD(I,2), CONS1(I,1)
READ(1,4) CONS2(I,1)
2 FORMAT(2D16.8,2D24.16)
4 FORMAT(2D24.16)
CONST(I,1)= CONS1(I,1) + CONS2(I,1)
XD(I,1)= 1.0
YD(I,1)= 1.0
CONST(I,2)= 0.0
XD(I,2)= YD(I,2)
3 CONTINUE
DO 60 KO=1,15
T= T + 0.01
WRITE(3,7) T
7 FORMAT('0',30X,'SAMPLING PERIOD ='F6.4)
DO 30 I=1,LP
30 A(I)=CEXP(XD(I,2) * T)
WRITE(3,32)
32 FORMAT('0',30X,'OPEN LOOP POLE LOCATIONS')
WRITE(3,34) (A(I),I=1,19)
34 FORMAT(' ',6F13.5)
WRITE(3,10)
10 FORMAT('0',4X,'FREQUENCY',4X,'FREQUENCY',4X,'MAGNITUDE',4X,
1 'MAGNITUDE',3X,'PHASE ANGLE',4X,'OMEGA-W'/7X,'IN HZ',6X,
2 'IN RAD/SEC',7X,'DB',8X,'UNITLESS',6X,'IN DEG',5X,'IN RAD/SEC'

```

```

3 )
C
C   DETERMINE FREQUENCY INCREMENT.
C
DELW= 3,1416/(50.0*T)
OMEGA= 0.0
OMEGAL=0.0
LX=0
C
C   CHECK TO SEE IF ANY OPEN LOOP POLES ARE IN A DELW DISTANCE OF
C   THE UNIT CIRCLE.
C
DO 22 I=1,LP
WROOT = CMLX(0.0,AIMAG(XD(I,2)))
IF(CABS(CEXP(XD(I,2)*T)-CEXP(WROOT*T)).GT.3.6/57.3 ) GO TO 22
IF(CABS(XD(I,2)).EQ.0.0)GO TO 22
LX=LX+1
WCRT(LX)= AIMAG(XD(I,2))
22 CONTINUE
C
C   DEFINE SUB-FREQUENCY INCREMENT.
C
WINK= DELW/10.0
DO 50 N=1,50
C
C   SET FREQUENCY.
C
OMEGA= OMEGA + DELW
MUL=0
KSTOP=0
C
C   DETERMINE IF THE CHOSEN POINT ON THE UNIT CIRCLE IS WITHIN A DELW
C   DISTANCE FROM A OPEN LOOP POLE. IF IT IS, SUB-INCREMENT THE
C   FREQUENCY IN A DELW INTERVAL.
C
IF(LX.EQ.0)GO TO 26
DO 24 I=1,LX
IF(ABS(WCRT(I)-OMEGA).GT.DELW)GO TO 24
MUL=10
OMEGA= WCRT(I) - DELW/2.0
IF(OMEGA-OMEGAL.LT.0.0)OMEGA=OMEGAL
24 CONTINUE
IF(MUL.EQ.0)GO TO 26
25 OMEGA=OMEGA + WINK
26 CONTINUE
C
C   CALCULATE THE MAGNITUDE AND PHASE OF THE OPEN LOOP TRANSFER
C   FUNCTION AT THE CHOSEN FREQUENCY.
C
GSAM=0.0
ARGU= CMLX(0.0,OMEGA * T)
Z= CEXP(ARGU)

```

```

C
C   EVALUATE COMPENSATOR AT CHOSEN FREQUENCY.
C
GCOMPS= (1.1*Z*Z+0.2*Z-0.9) / (2.4*Z*Z-1.6*Z)
DO 40 I=1,LP
40 GSAM= GSAM + GAIN * CONST(I,1) * (Z- H)/(Z-A(I))
GSIR= GSAM
GCANCL= 57.3 * ATAN2(AIMAG(GSIR),REAL(GSIR))
COMPCG= 20.0 * ALOG10(CABS(GSIR))
GSUM= GSAM * GCOMPS
GMAG= CABS(GSUM)
B= AIMAG(GSUM)
C= REAL(GSUM)
ANGLE= 57.3 * ATAN2(B,C)
FRHZ= OMEGA/6.2832
GMAGDB= 20.0 * ALOG10(GMAG)
IF(ABS((OMEGA*T/2.0)-1.5708).LT.0.0005)OMEGA=OMEGA+DELW/2.0
OMEGAW= TAN(OMEGA * T / 2.0)
WRITE(3,45) FRHZ, OMEGA, GMAGDB,GMAG, ANGLE, OMEGAW, COMPCG,
1 GCANCL
45 FORMAT(' ',8F13.5)
OMEGAL= OMEGA

C
C   DETERMINE IF OMEGA * T IS GREATER THAN PI. IF IT IS, TERMINATE
C   FOR THE SAMPLING THAT IS BEING USED.
C
IF(OMEGA.GT.3.1416/T)GO TO 53
KSTOP= KSTOP + 1
IF(KSTOP.LE.MUL)GO TO 25
50 CONTINUE
53 CONTINUE
WRITE (3,55)
55 FORMAT('1')
60 CONTINUE
65 CONTINUE
STOP
END

```

APPENDIX F

REGULAR ROOT LOCI PROGRAM

The succeeding computer program is a variation of the computer program in Appendix D. The input and output variables are the same. The difference is that this program is used for finding root locations so that a regular root locus can be made. Slight modifications have been made in the main program and in the subroutine PRDZ. The modifications in the main program are that the sampling is held constant while the gain is varied; whereas, in the program in Appendix D the gain is held constant and the sampling period is varied. The changes in the subroutine PRDZ were so that another root would be predicted when a change in gain occurs, rather than a change in sampling period as in Appendix D.

```
DIMENSION CHEQ(20), ANUMR(20), DENOM(20)
COMPLEX XD(20), CONST*16(20,20), YD*16(20,20), STARTP(20),
1 ROOT(20), ZNEW, ZOLD, CMLPX, CABS, CONJG
COMPLEX ZSTART(20)
CALL INTERD
DO 100 JO=1.8
READ(1,1) TF
WRITE(3,5) TF
1 FORMAT(14X,F10.5)
5 FORMAT('1',35X,'FLIGHT TIME = ',F10.5)
DO 6 I=1,13
ROOT(I)= 0.0
6 STARTP(I)= 0.0
DO 3 I=1,14
READ(1,2) YD(I,2), CONST(I,1)
2 FORMAT(2D16.8,2D24.16)
XD(I,1)= 1.0
YD(I,1)= 1.0
CONST(I,2)= 0.0
3 CONTINUE
DO 4 I=1,14
4 XD(I,2)= YD(I,2)
```

```

IER= 0
DELT=-0.1
DO 100 LOX= 1,3
READ(1,10) T
WRITE(3,9) T
9 FORMAT('0',12X,'SAMPLING PERIOD =',F6.4)
READ(1,12) (ANUMR(I),I=1,14)
READ(1,12) (DENOM(I),I=1,14)
10 FORMAT(17X,F6.4)
12 FORMAT(5E16.8)
GAIN=-0.1
DO 100 J=1,11
LP=13
GAIN=GAIN + 0.1
IF(TF.EQ.100.0)GAIN=GAIN-0.2
13 CONTINUE
DO 15 I=1,14
15 CHEQ(I)= DENOM(I)+ GAIN * ANUMR(I)
DO 19 I=1,LP
CHECK= ABS(CHEQ(15-I)/CHEQ(14-I))
IF(CHECK-1.0E-04)17,17,21
17 CHEQ(15-I)=0.0
LP=LP - 1
ROOT(14-I)= 0.0
STARTP(14-I)= 0.0
19 CONTINUE
21 CONTINUE
I=1
16 CONTINUE
ZOLD= ROOT(I)
CALL PRDZ(ZOLD,ZNEW,14,T,IER,DELT,GAIN,CONST,XD,UI,UJ,UK)
IF(IER.NE.0)ZNEW= ROOT(I)
STARTP(I)= ZNEW
IF(AIMAG(ROOT(I)))20,18,20
18 I=I+1
IF(I-LP)16,16,25
20 STARP(I+1)= CONJG(ZNEW)
I=I+2
IF(I-LP)16,16,25
25 CONTINUE
CALL POLSER(CHEQ,ROOT,IER,IEND,LP,STARTP)
IF(IER.LT.98)GO TO 35
WRITE(3,28) IER
28 FORMAT('0',10X,'IER =',I5)
GO TO 100
35 IF(IEND.LT.98)GO TO 40
WRITE(3,38) IEND
38 FORMAT('0',10X,'IEND = ',I5)
GO TO 100
40 CONTINUE
WRITE(3,45) GAIN
45 FORMAT('0',12X,'OPEN LOOP GAIN =',F6.4)
WRITE(3,50)

```

```

50 FORMAT('0',2X,'PREDICTED ROOTS',21X,'ROOTS',14X,'MAGNITUDES OF
1  ROOTS')
   KS=0
   DO 65 I=1,13
     ABSRT= CABS(ROOT(I))
     WRITE(3,55) STARTP(I), ROOT(I), ABSRT
55  FORMAT('0',2F10.5,10X,2F10.5,10X,F11.8)
     IF(ABSRT.GE.1.0)KS=1
65  CONTINUE
     IF(KS.LT.1)GO TO 80
     WRITE(3,70)
70  FORMAT('0',30X,'*** SYSTEM UNSTABLE ***')
     GO TO 95
80  WRITE(3,90)
90  FORMAT('0',31X,'*** SYSTEM STABLE ***')
95  CONTINUE
100 CONTINUE
110 CONTINUE
     STOP
     END

```

```

SUBROUTINE PRDZ(ROOT,ZPRED,N,T,IER,DT,K,CONST,XD,UI,UJ,UK)

```

C
C
C

```

PREDICT Z

```

```

COMPLEX CABS,CEXP,CONST*16(20,20),XD(20,20),SON
COMPLEX*16 ROOT*8,ZPRED*8,ZX,CDABS,ZY,ZT,CMLPX*8,SUM
DOUBLE PRECISION PRNX,PRNY,PRNT,PINX,PINY,PINT,PDNX,PDNY,PDNT,
1  DN,RN,AIN,PRIX,PRIY,PRIT,PIIX,PIIY,PIIT,PGX,PGY,PGT,PIGX,
2  PIGY,PIGT
REAL K
IER=0
PGX=0.0
PGY=0.0
PGT=0.0
PIGX=0.0
PIGY=0.0
PIGT=0.0
X=REAL(ROOT)
Y=AIMAG(ROOT)
DO 3 I=1,N
SON= CONST(I,1)
D= REAL(SON)
E= AIMAG(SON)
R=D
S=E
E= E* K
D=D * K
OMEGA= -AIMAG(XD(I,2))
A= - REAL(XD(I,2))
ACOS=COS(+OMEGA * T)
ASIN=SIN(+OMEGA * T)
AEXP=EXP(-A * T)

```

```

4 PRNX= 2.0 * X - AEXP * ACOS - 1.0
  PRNY= 2.0 * Y + AEXP * ASIN
  PINX=-AEXP * ASIN
  PINY=-AEXP * ACOS + 1.0
  PDNX= 2.0 * X - 2.0 * AEXP * ACOS
  PDNY= 2.0 * Y + 2.0 * AEXP * ASIN
  DN=X*X + Y*Y - 2.0 * AEXP * (X * ACOS- Y * ASIN) + AEXP * AEXP
  IF(DN)7,6,7
6 Y= Y + 1.0E-06
  GO TO 4
7 RN= X*(X - 1.0) + Y*Y - AEXP * ((X - 1.0) * ACOS - Y * ASIN)
  AIN= Y - AEXP * ((X - 1.0) * ASIN + Y * ACOS)
  PRIX= (DN * PRNX - RN * PDNX)/(DN * DN)
  PRIY= (DN * PRNY - RN * PDNY)/(DN * DN)
  PRIT= RN/DN
  PIIX=( DN * PINX - AIN * PDNX)/(DN * DN)
  PIYI=( DN * PINY - AIN * PDNX)/(DN * DN)
  PIIT= AIN/DN
  PGX= PGX + PRIX * D - E * PIIX
  PGY= PGY + PRIY * D - E * PIYI
  PGT= PGT + PRIT * R - S * PIIT
  PIGX= PIGX + PIIX * D + E * PRIX
  PIGY= PIGY + PIYI * D + E * PRIY
3 PIGT= PIGT + PIIT * R + S * PRIT
  ROOTR= REAL(ROOT)
  ROOTI= AIMAG(ROOT)
  UI= PGY * PIGT - PIGY * PGT
  UJ= PIGX * PGT - PGX * PIGT
  UK= PGX * PIGY - PIGX * PGY
  IF(UK.EQ.0.0)IER=1
  IF(IER.GT.0)GO TO 10
5 CONTINUE
  X= (UI/UK) * DT + ROOTR
  Y= (UJ/UK) * DT + ROOTI
  ZPRED= CMLPX(X,Y)
10 CONTINUE
  RETURN
  END

SUBROUTINE POLSER(CONST,ROOT,IER,IEND,M,STARTP)
  DIMENSION CONST(1), COF(20), YCONS(20)
  DIMENSION RCOF(20)
  DOUBLE PRECISION COF, YCONS, ALPHA,SUMQ, RCOF
  DOUBLE PRECISION FR,FI,PFRX,PFLX,PFRY,PFIY,FRO,FIO,PERXO,
1  PFRYO,PFIYO,FR1,FI1,PFRX1,PFRY1,PFIY1,PFLX1,PFY,PFY
  COMPLEX*16 SO, FUN, DERV, CDABS
  COMPLEX S1,CMLPX, ROOT(20),CABS, STARTP(20), CONJG, SINT(20)
  N=M+1
  DO 10 I=1,N
10 COF(I)= CONST(I)
  J=1
  5 IEND= 0
  DO 6 I=1,N

```

```

6 YCONS(I)= COF(I)
  X1= REAL(STARTP(J))
  Y1= AIMAG(STARTP(J)) + 1.0E-04
  ISKIP= 0
15 IEND= IEND + 1
  X= X1
  Y= Y1
  DO 18 I=1,N
  KJ= N + 1 - I
18 RCOF(I)= COF(KJ)
  FR= RCOF(1)
  FI= 0.0
  PFRX= 0.0
  PFIX= 0.0
  PFRY= 0.0
  PFIY= 0.0
  FRO= X
  FIO= Y
  PFRXO= 1.0
  PFIYO= 1.0
  PFIYO= 1.0
  PFRYO= 0.0
  PFIYO= 1.0
  DO 25 I=2,N
  PFRX= PFRX + RCOF(I) * PFRXO
  PFIX= PFIX + RCOF(I) * PFIYO
  PFRY= PFRY + RCOF(I) * PFRYO
  PFIY= PFIY + RCOF(I) * PFIYO
  FR= FR + RCOF(I) * FRO
  FI= FI + RCOF(I) * FIO
  FR1= X * FRO - Y * FIO
  FI1= Y * FRO + X * FIO
  PFRX1= FRO + X * PFRXO - Y * PFIYO
  PFI1= Y * PFRXO + FIO + X * PFIYO
  PFRY1= X * PFRYO - FIO - Y * PFIYO
  PFIY1= FRO + Y * PFRYO + X * PFIYO
  PFRXO= PFRX1
  PFIYO= PFI1
  PFRYO= PFRY1
  PFIYO= PFIY1
  FRO= FR1
  FIO= FI1
25 CONTINUE
  PFX= 2.0 * ( FR * PFRX + FI * PFIY )
  PFY= 2.0 * ( FR * PFRY + FI * PFIY )
  ZAP=DSQRT( FR * FR + FI * FI)
  IER=0
  IF(ISKIP.EQ.0)ZAP= ZAP + 100000.0
  IF(ISKIP.EQ.0)GO TO 35
  IF(ZIP + ZIP/ZOP.GE.1.0)DELX= 0.30
  IF(ZIP + ZIP/ZOP.LT.1.00) DELX= 0.20
  IF(ZIP + ZIP/ZOP.LT.0.15) DELX= 0.10
  IF(ZIP + ZIP/ZOP.LT.0.05) DELX= 0.05
  GO TO 27

```



```

26 DELX= DELX/2.0
27 DELX= ABS(DELX)
   IF(PFX.GT.0.0)DELX= -DELX
   DX= DELX
28 X1= X + DX
   IF(PFX.NE.0.0)GO TO 30
   Y1= Y - PFY * DX / DABS(PFY)
   X1= X1 -DX
   GO TO 35
30 Y1= (PFY/PFX) * DX + Y
   IF(ABS(Y1-Y) + ABS( DX ).LT.SQRT(2.0*DELX*DELX))GO TO 35
   DX=DX/2.0
   GO TO 28
35 CONTINUE
   ISKIP= 1
36 SO= CMPLX(X1,Y1)
   IF(IER.GT.98)GO TO 50
   IER= IER + 1
   MONK=0
   JX=N-1
   NEND= 0
37 FUN= YCONS(1)
   DERV= 0.0
   NXX=JX+1
   DO 40 I=1,JX
   P= NXX - I
   DERV= DERV * SO + YCONS(I) * P
40 FUN= FUN * SO + YCONS(I+1)
   ZIP= CDABS(FUN)
   ZOP= CDABS(DERV)
   NEND= NEND + 1
   IF(NEND.GT.50)IEND=200
   IF(MONK.GT.0)GO TO 42
   IF(ZIP.GT.ZAP)GO TO 26
   IF(CDABS(FUN/DERV).GT.0.025)GO TO 15
42 IF(IEND.GT.98)GO TO 50
   IF(CDABS(FUN/DERV)+ZIP.LT.1.0E-04)GO TO 43
   SO= SO - FUN/DERV
   GO TO 37
43 IF(MONK.NE.0)GO TO 45
   MONK= MONK+1
   IF(MONK.GT.5)IEND=100
   MX=M+1
   DO 44 I=1,MX
44 YCONS(I)= CONST(I)
   JX= M
   GO TO 37
45 ROOT(J)= SO
   S1= SO
   IF(ABS(AIMAG(ROOT(J))/REAL(ROOT(J))).LT.1.0E-04)ROOT(J)=CMPLX(
*   REAL(S1),0.0)
   IF(AIMAG(ROOT(J)).EQ.0.0)GO TO 60

```

```
    ROOT(J+1)= CONJG(ROOT(J))
    J=J+2
    ALPHA= 2 * REAL(S1)
    SUMSQ= REAL(S1) * REAL(S1) + AIMAG(S1) * AIMAG(S1)
    N=N-2
    GO TO 140
60  J= J + 1
    SUMSQ= 0.0
    ALPHA= REAL(S1)
    N=N-1
140 COF(2) + ALPHA * COF(1)
    NX= N-1
145 DO 150 L=2,NX
150 COF(L+1)= COF(L+1) + ALPHA * COF(L) - SUMSQ * COF(L-1)
    IF(N-1)50,50,5
50  CONTINUE
    RETURN
    END
```

REFERENCES

1. R. M. Willett, "Sampled - Data Adaptive Control Utilizing Variable Sampling Rates," Ph.D. Dissertation, Iowa State University, 1966.
2. W. L. McDaniel and J. W. Dalton, "A Modified Root Locus Technique for a Class of Sampled-Data Systems," Proceedings of the Asilomar Conference on Circuits and Systems, U. S. Naval Post Graduate School, Monterey, California, November 1967.
3. B. C. Kuo, Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Inc. Englewood Cliffs, N. J., 1963, pp. 216-239.
4. J. R. Mitchell and W. L. McDaniel, Jr., "A Generalized Root Locus Following Technique," Proceedings of the Eighth Adaptive Processes Symposium, The Pennsylvania State University, November 1969.

