T-6 DIGITAL AUTOPILOT DATA PROCESSING ANALYSIS AND SPECIFICATION

C. Frank Asquity

Army Missile Research, Development and Engineering Laboratory

5 March 1975

TECHNICAL REPORT RG—75—36

T-6 DIGITAL AUTOPILOT DATA PROCESSING ANALYSIS AND SPECIFICATION

C. Frank Asquith
Guidance and Control Directorate
US Army Missile Research, Development and Engineering Laboratory
US Army Missile Command
Redstone Arsenal, Alabama 35809

5 March 1975

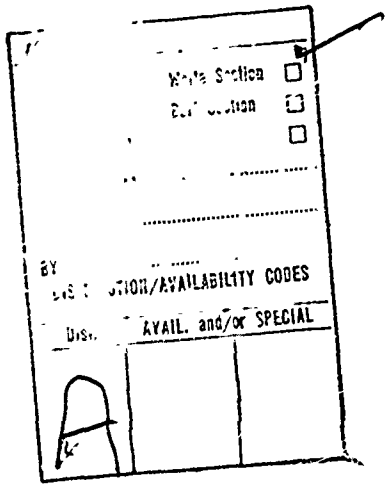
U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama

DISPOSITION INSTRUCTIONS

Destroy this report when it is no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.



TRADE NAMES

Use of trade names or manufacturers in this report does not constitute an official indorsement or approval of the use of such commericial hardware or software.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
REPORT NUMBER RG-75-36	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD-A01[illegible]
4. TITLE (and Subtitle) T-6 DIGITAL AUTOPILOT DATA PROCESSING ANALYSIS AND SPECIFICATION		5 TYPE OF REPORT & PERIOD COVERED Technical Report 6 PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) C. Frank Asquith		8. CONTRACT OR GRANT NUMBER(s)
9 PERFORMING ORGANIZATION NAME AND ADDRESS Commander US Army Missile Command Attn: AMSMI-RG Redstone Arsenal, Alabama 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (DA) 1M362303A214 AMCMSC 632303.11.21402.60
11 CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 5 March 1975 13. NUMBER OF PAGES 65
14 MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15 SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital microprocessor
Homing missile
Analog system
State variable system

PRICES SUBJECT TO CHANGE

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the analysis required to specify the computer data processing for a digital autopilot for a small homing missile. The problem of conversion of an analog system to a digital system by state variable methods is examined. Modification of the resulting digital system to maintain system gain and phase margins is accomplished. Data rate and word length effects are studied and the decomposed state variable type formulation is found to be superior to the z-transform based recursive feedback form.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

1. Introduction and System Requirements

The work described in this report is part of a larger effort to demonstrate the feasibility of using a digital microprocessor to guide and control a small homing missile. The objective of this portion of the project is to define the baseline processing requirements to be imposed on the microprocessor. For this purpose, the microprocessor is defined as a small stored program digital computer with a word length of 16 bits or less, memory capacity of about 1000 words or less, and a memory access time on the order of 2 microseconds. The reason for establishing these guidelines is that the processor must eventually go on board the missile and must be small and inexpensive. Hopefully, the microprocessor will replace present analog circuitry which is alleged to be less reliable, less flexible, bulkier, and more costly. These hardware limitations influence the choice of algorithms to be implemented in the available memory and, more importantly, computation speed is severely bounded.

The missile chosen for this feasibility demonstration is of the HELLFIRE class and the project has been named the Modular Missile Digital Autopilot. It was decided that a good comparison between the present analog system and the proposed digital system could be obtained by converting the analog equations to difference equations which can be programmed. The performance of the analog system will provide a standard against which the performance of the digital system can be judged. If the difference equations can be executed at rates and word lengths such that the results obtained from both systems agree within certain established boundaries, then a satisfactory set of processing requirements for the microprocessor will have been derived. It is recognized that a simple conversion of an analog system to a digital system is not necessarily the best approach to the design of a digital system. A better and more effective system for microprocessor implementation can probably be developed in the digital domain starting from basic missile system performance requirements. However, such a development is a lengthy task that would require extensive analysis and simulation work. It is believed that the analog to digital approach offers a quicker and more predictable chance of success and its performance and processing requirements will offer standards against which any more sophisticated algorithms developed in the future may be judged.

Certain guidelines were used in accepting or rejecting candidate digital implementations. It seemed reasonable that the system phase margin should not be degraded by more than 5 degrees and the gain margin by more than 1 dB. Digital system performance should not deteriorate below the loop crossover frequencies and the data rates should be high enough so that some high frequency roll-off is provided above crossover. Furthermore, it was obvious from the beginning that the primary processing load would be multiplications and a reasonable estimate for this operation in a microprocessor is 50 microseconds. Doubling this time to allow for all other types of processing set a limit

of 10,000 multiplications per second as a maximum allowable load. These figures were used in rejecting or retaining certain approaches to the digital system design.

2. The Analog System

The original analog guidance and control system is shown in Figure 1. The system contains loops of two basic types: a 3-axis inner gyro loop and a 2-axis outer seeker loop. The gyro loop response is shaped by the control compensators $F_3(s)$ in pitch and yaw and $F_4(s)$ in roll. The compensator double zero furnishes adequate lead angle to stabilize the loop and the poles contribute roll-off as well as easing the realizability problem with standard circuit components. The seeker related functions, $F_1(s)$ and $F_2(s)$, provide integration of the angular rate error signals, a low frequency lead zero, and signal filtering by way of a triple pole. The microprocessor is supposed to perform all of the functions represented by $F_1(s)$, $F_2(s)$, $F_3(s)$, and $F_4(s)$.

3. Possible Approaches

The objective of this task now becomes the derivation of a set of programmable difference equations which can be executed at iteration rates such that the performance of the digital system approximates that of the analog system within the guidelines mentioned under Section 1. There exists a variety of potential solutions to this problem many of which might be satisfactory. Several of these solutions were examined.

a. State Variable Formulation

The state variable approach is quite general and consists of the following procedure:

1) Decompose the analog system transfer functions into a system of simple integrators, gains, and summing points.

2) Write down the system vector-matrix first order differential equation of the form

$$\dot{[X]} = [F][X] + [G][U] \tag{3-1}$$

using the integrator outputs as state variables.

3) Obtain the corresponding difference equation

$$[X]_N = [\Phi][X]_{N-1} + [\Gamma][U]_{N-1} \tag{3-2}$$

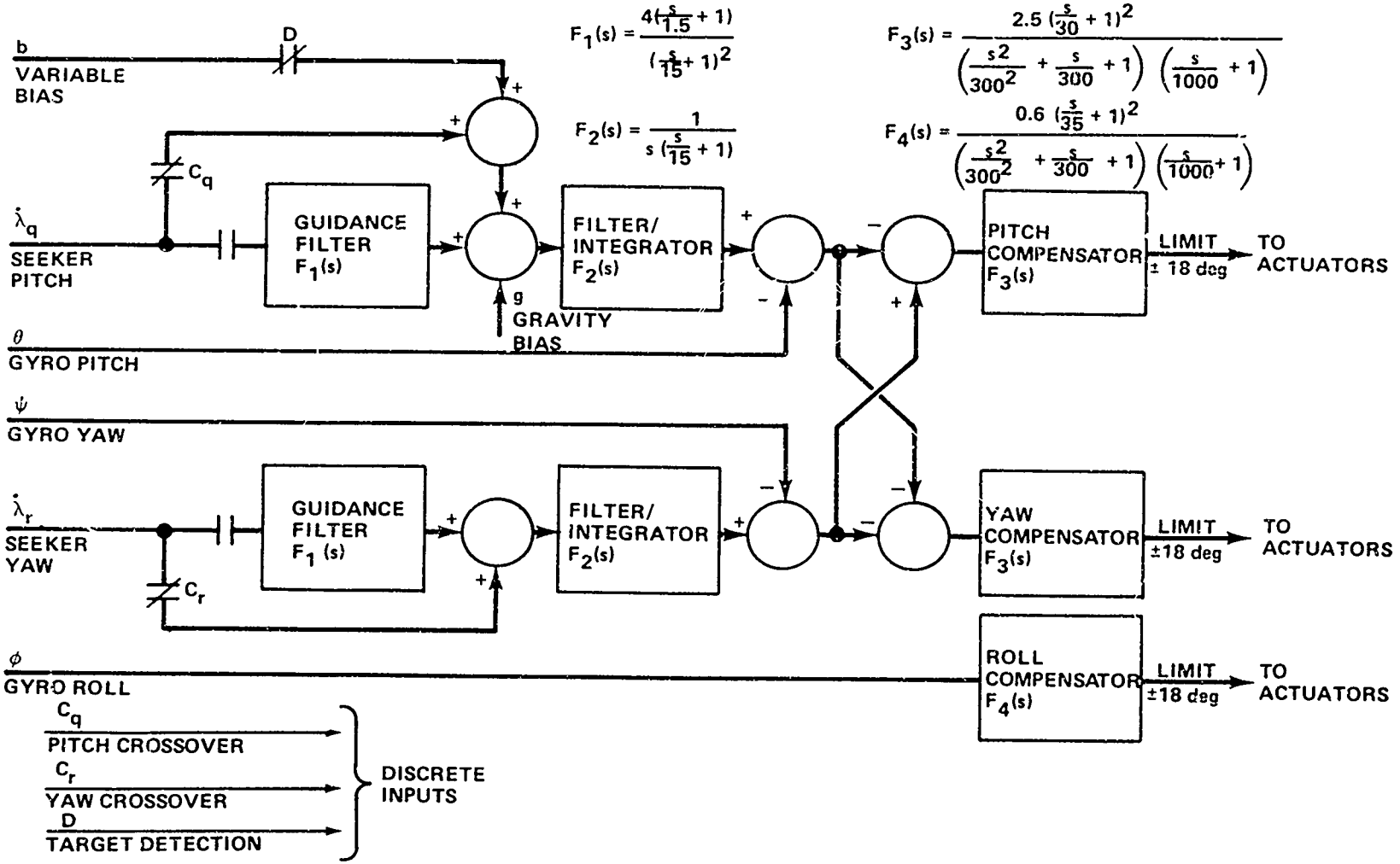


Figure 1. Analog functions to be implemented digitally.

using any of several methods for computing the Φ and Γ matrixes. The series expressions

$$[\Phi] = \sum_{i=0}^{\infty} \frac{[F]^i T^i}{i!} \quad [\Gamma] = \sum_{i=0}^{\infty} \frac{[F]^i T^{(i+1)}}{(i+1)!} [G] \tag{3-3}$$

are especially useful if the job is done on a digital computer. Usually, the system output may be obtained satisfactorily using the analog observation equation gains

$$[Y]_N = [H][X]_N \quad . \tag{3-4}$$

The system characteristics can be examined for different F and G matrixes that result from various decompositions of the analog system and the effect of the update interval T can be studied. Alternative equations that are occasionally useful are

$$[\Phi(T)] = \mathcal{L}^{-1} \{s[I] - [F]\}^{-1} \Big|_{t=T} \tag{3-5}$$

and

$$[\Gamma(T)] = \int_0^T [\Phi(\tau)] \, d\tau \, [G] \quad . \tag{3-6}$$

These equations were used in checking some of the work done with the simpler functions.

b. z-Transform Method

The z-transform approach as used herein generates a frequency domain transfer function such as

$$\frac{y(z)}{u(z)} = \frac{A_1 z^{-1} + A_2 z^{-2} + A_3 z^{-3} + \text{---}}{1 + B_1 z^{-1} + B_2 z^{-2} + B_3 z^{-3} + \text{---}} \quad . \tag{3-7}$$

Cross multiplication and conversion of the resulting equation to the time domain result in an expression such as

$$\begin{aligned} y_N = {} & A_1 u_{N-1} + A_2 u_{N-2} + A_3 u_{N-3} + \text{---} \\ & - (B_1 y_{N-1} + B_2 y_{N-2} + B_3 y_{N-3} + \text{---}) \end{aligned} \tag{3-8}$$

which gives the output y in terms of past outputs and inputs. The

transfer function $y(z)/u(z)$ may require considerable labor in its derivation. Very simple analog functions can be found in tables. Complex functions have to be decomposed into simpler terms that can be handled separately and then recombined to yield a single recursive feedback form such as Equation (3-7).

c. Pole-Zero Design

Pole-zero design in the digital domain can save a lot of time depending on how the problem is stated. If the objective is to duplicate the performance of an analog transfer function as best as is possible, then a logical thing to do is to duplicate its poles and zeros in the digital domain. This is certainly reasonable if the analog system was designed in the frequency domain in the first place, which is nearly always the case for filters and compensators. A transfer function such as

$$F(s) = \frac{s + a}{(s + b)(s^2 + 2\,\delta\omega\, s + \omega^2)} \tag{3-9}$$

would appear as

$$F(z) = \frac{z - e^{-aT}}{(z - e^{-at})(z^2 - 2z\, e^{-\delta\omega T} \cos \omega T \sqrt{1 - \delta^2} + e^{-2\delta\omega T})} \tag{3-10}$$

The functions F(s) and F(z) have poles and zeros at identical locations in the s-plane. However, F(z) has an infinite number of additional poles and zeros outside the primary strip at intervals separated by $2\pi/T$ along lines parallel to the $j\omega$-axis through the original poles and zeros. If frequency response matching is established as a goal for the analog and digital systems, this simplistic approach is about as good as any. It can be argued that this approach is the same as that of the z-transform described previously, but the concept is a little different and more direct. The proper gain has to be inserted so that F(z) and F(s) behave similarly. In all the methods described some zero order hold compensation may be necessary. In the end, pole-zero matching was used in the control compensator.

4. Recursive Feedback Formulation

Early in the task, the recursive feedback formulation exemplified by Equation (3-8) appeared to be a very desirable approach to the problem. The reason for this is the small number of computations required relative to those of algorithms derived by state variable methods. This type of formulation, however, turned out to be highly susceptible to word length effects. Originally, this was blamed on the z-transform but the fault, of course, lies in the form of the final

result and not in the technique used. Difference equations derived by state variables can also be combined into this form and would suffer from the same malady. Conversely, a z-transform derived digital system might work at acceptably short word lengths if left in a decomposed form. However, the total number of computations required will increase and the advantage of this approach over the state variable method will be lost. Obviously, a pole-zero design might also suffer from word length problems unless it is decomposed.

The performance of the recursive feedback formulation derived from a z-transform transfer function suffered spectacular degradation as the data rate approached that rate required for a satisfactory approximation of the analog system. This was very bad since it was believed from the beginning that high execution rates and short word lengths with predominantly single precision arithmetic would be required of the microprocessor. The work that follows is an example of the problem that was encountered.

The version of the filter/integrator transform to be discussed was formulated by letting

$$F(z) = (1 - z^{-1})\, Z\left[\frac{\frac{s}{1.5} + 1}{s^2\left(\frac{s}{15} + 1\right)^3}\right] \quad . \tag{4-1}$$

This function was taken directly from Figure 1 by preceding the function

$$F_1F_2(s) = \frac{\frac{s}{1.5} + 1}{s\left(\frac{s}{15} + 1\right)^3} \tag{4-2}$$

by a zero order hold whose transfer function is

$$F_{ZOH}(s) = \frac{1 - e^{-sT}}{s} \quad . \tag{4-3}$$

The reason for the fictitious hold circuit was that the z-transform derived version of the filter/integrator was being used as a check on the state variable version. The function in Equation (4-1) was decomposed as

$$\frac{\frac{s}{1.5} + 1}{s^2\left(\frac{s}{15} + 1\right)^3} = \frac{7/15}{s} + \frac{1}{s^2} - \frac{7/15}{s + 15} - \frac{8}{(s + 15)^2} - \frac{135}{(s + 15)^3} \quad . \tag{4-4}$$

The transforms of the individual terms were found in tables and terms recombined to give

$$F(z) = \frac{A_1 z^3 + A_2 z^2 + A_3 z + A_4}{z^4 + B_1 z^3 + B_2 z^2 + B_3 z + B_4} \tag{4-5}$$

in which

$$A_1 = \left[-\frac{7}{15} - 8T - 135\frac{T^2}{2}\right] e^{-15T} + \frac{7}{15} + T \tag{4-6}$$

$$A_2 = \left[\frac{14}{15} + 8T - 135\frac{T^2}{2}\right] e^{-30T} + \left[-\frac{7}{15} + 13T + 270\frac{T^2}{2}\right] e^{-15T} - \frac{7}{15} \tag{4-7}$$

$$A_3 = -\frac{7}{15} e^{-45T} + \left[-\frac{7}{15} - 13T + 270\frac{T^2}{2}\right] e^{-30T} \tag{4-8}$$

$$+ \left[\frac{14}{15} - 8T - 135\frac{T^2}{2}\right] e^{-15T}$$

$$A_4 = \left[\frac{7}{15} - T\right] e^{-45T} + \left[-\frac{7}{15} + 8T - 135\frac{T^2}{2}\right] e^{-30T} \tag{4-9}$$

$$B_1 = -(1 + 3 e^{-15T}) \tag{4-10}$$

$$B_2 = 3 e^{-15T} (1 + e^{-15T}) \tag{4-11}$$

$$B_3 = -e^{-30T} (3 + e^{-15T}) \tag{4-12}$$

$$B_4 = e^{-45T} \quad . \tag{4-13}$$

The frequency response of Equation (4-5) can be studied by letting $z = e^{j\omega T}$ in which T is the data interval. The time response can be gotten from the corresponding difference equation

$$y_N = A_1 u_{N-1} + A_2 u_{N-2} + A_3 u_{N-3} + A_4 u_{N-4} \tag{4-14}$$

$$- B_1 y_{N-1} - B_2 y_{N-2} - B_3 y_{N-3} - B_4 y_{N-4} \quad .$$

Equation (4-14) is what would be programmed in the microprocessor for the function $F_1 F_2(s)$. There are only 8 multiplications and 7 additions required in the execution of this equation. The response is identical, theoretically, to that of the state variable equations to be presented later. The state variable method will generally be found to require a few more operations in its execution. After signal cross over in the q and r channels of Figure 1, two equations such as Equations (4-14) are all that are required for seeker error signal integration and filtering.

Frequency response data were taken using a Hewlett/Packard 9830 programmable calculator. The results are summarized in Figures 2 and 3. The response of the original analog system represented by Equation (4-2) is shown for comparison. The response is shown as a function of the data interval, T, and the number of significant decimal digits in the numerator and denominator coefficients of Equation (4-5).

For the purpose of drawing conclusions from Figures 2 and 3, recall that only the coefficients have been rounded off. In practice, the variables would not only be rounded but also scaled. It is believed that maintaining four decimal digits of precision in a 16-bit word length processor is difficult if not impossible. Five or six place precision is out of the question. Furthermore, a data rate of 50 samples per second or greater is highly desirable in the seeker loop to prevent deterioration of the system phase margin (this will be elaborated on later in this report). What one would like to see is an algorithm whose response agrees with that of the analog system up to at least 20 radians per second to within 1 dB in gain and 2 or 3 degrees in phase margin not counting the zero order hold delay. Most of the latter problem can be compensated for if the data rate is high enough. It would be encouraging if this agreement was possible with as few as three significant decimal digits for the coefficients. The 3-digit data taken for this algorithm were so bad that it was not plotted. The 4-digit data are unsatisfactory at 20 samples per second and 5-digit data show deterioration at 50 samples per second. It was concluded from this information that the recursive feedback formulation is unsatisfactory for use in the microprocessor.

Step response data were taken using Equation (4-14) and the conclusions regarding performance degradation with word length agreed with those from the frequency response data. The specific cause of the poor short word length performance of the recursive feedback form lies in the magnitudes of the individual numerator and denominator terms relative to their algebraic sums. The dominator coefficients, in particular, remain on the order of unity as the data rate increases but the sum of the terms becomes very small. Leading digits are lost when the sum is formed.

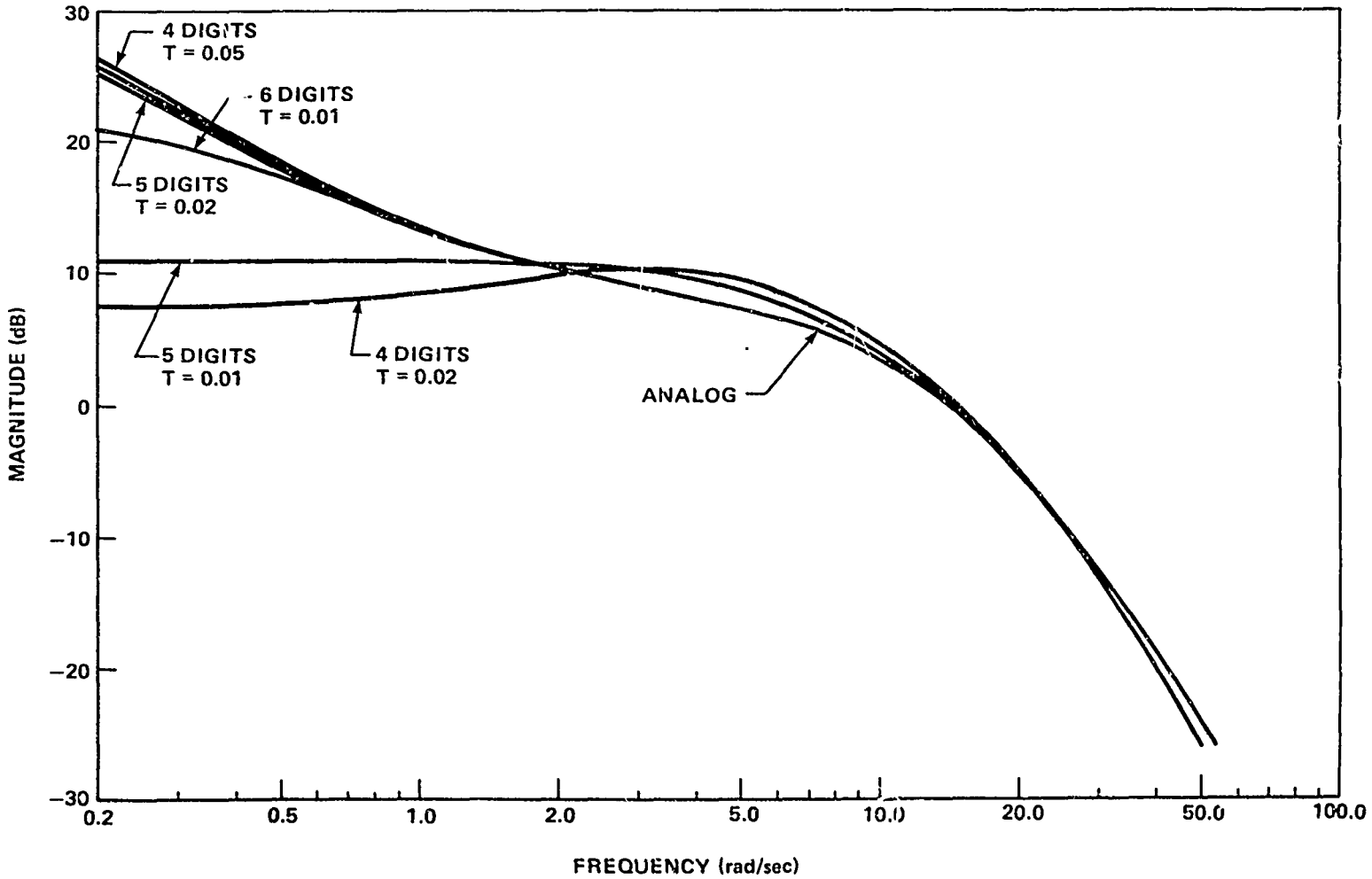


Figure 2. Seeker filter/integrator magnitude response recursive feedback form.

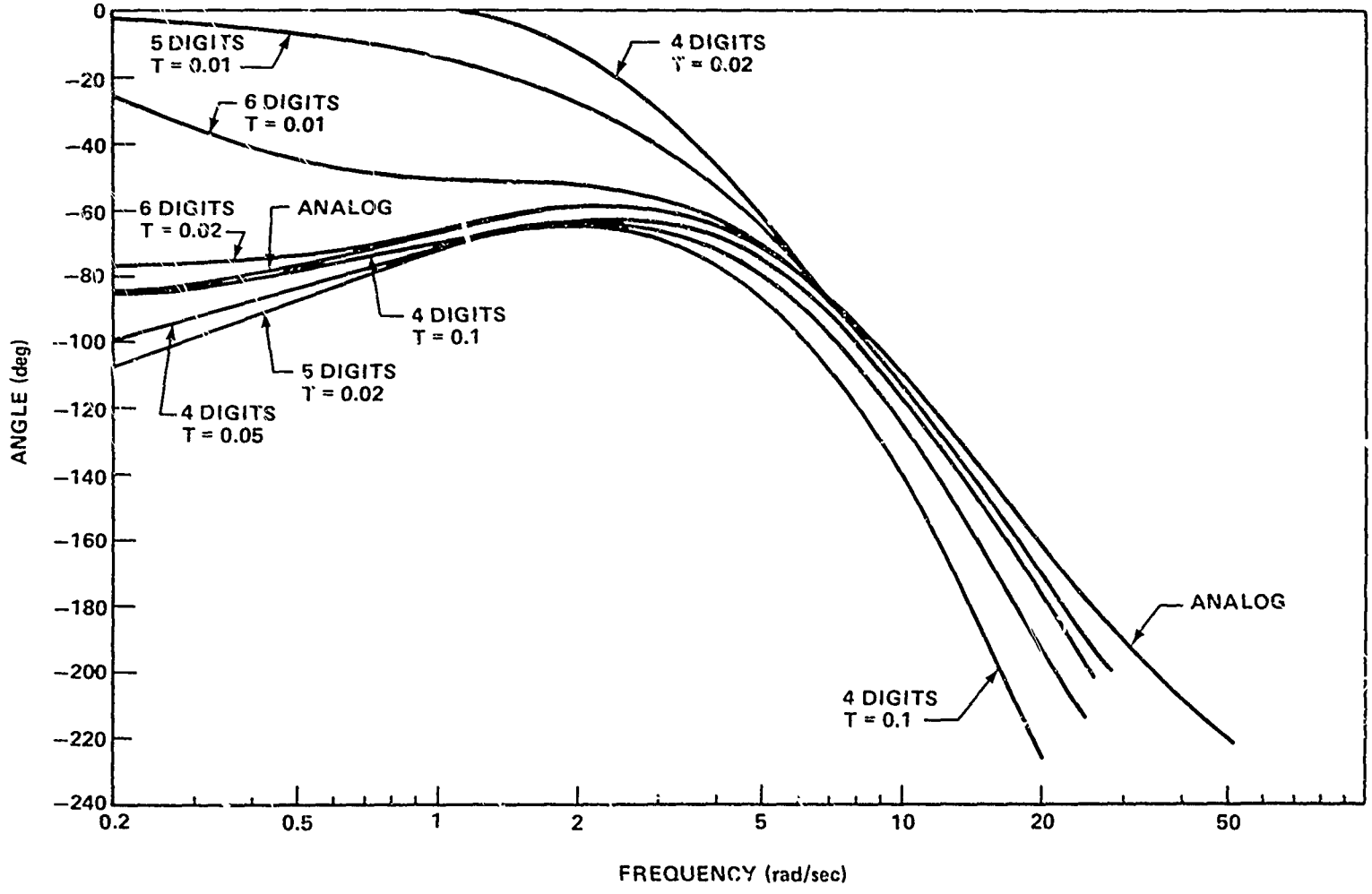


Figure 3. Seeker filter integrator phase response recursive feedback form.

Actually, the problem with the z-transform based recursive feedback formulation was first noticed before word length was ever considered. At very high data rates (2000 samples per second or greater) it was seen that the digital system output got worse but had consistently improved as the rate increased up to that point. Word length is not normally a problem in the H/P 9830 since it uses four 16-bit words for variable storage and maintains 12 significant decimal digits of accuracy between 10^{-99} to 10^{99}. This is better than some large general purpose computers in single precision. These considerations led to an early investigation of word length effects in this algorithm and caused its subsequent abandonment. The z-transform (recursive feedback form) word length matter is mentioned in several references but its severity was not appreciated prior to this investigation.

5. The State Variable System

The analog system shown in Figure 1 can be represented by 17 state variables. Three of these are in the roll channel leaving 14 in the pitch/yaw channels. The basic inputs to the system are the seeker error rate measurements, $\dot{\lambda}_q$ and $\dot{\lambda}_r$ and the gyro pitch, yaw, and roll angles designated θ, ψ, and ϕ. The outputs are z_1, z_2, and z_3 so that the summing junctions preceding the actuators remain analog. This arrangement will minimize the number of digital to analog converters required.

Several potential approaches to the state variable conversion of the analog system to an equivalent digital system were examined. Only the final one will be discussed in detail. In the end, a parallel oriented decomposition of the analog system was used. The system was separated into two q/r channel guidance filter/integrator blocks and three pitch/yaw/roll gyro compensator blocks. This means that a fictitious zero order hold was inserted at the summing junctions at the inputs to the pitch/yaw compensators. The reasons for choosing this arrangement are:

a) A parallel oriented system is computationally efficient in that coupling among the states is minimized.

b) The compensators require higher execution rates than the filter/integrators and it is easier to fulfill this requirement if the compensator states are not coupled back into the filter/integrators.

c) The gyro inputs have to come through a real zero order hold in the digital system so it should be acceptable to assume a hold on the filter integrator outputs with which they are summed.

d) The fact that the filter/integrator outputs go through the compensators is incidental anyway. The seeker error signals could be processed and added to the compensator outputs at the actuator junction.

The q/r channel filter integrator after signal crossover has the form

$$F_1F_2(s) = \frac{4\left(\frac{s}{a}+1\right)}{s\left(\frac{s}{b}+1\right)^3} \qquad \begin{matrix} a = 1.5 \\ b = 15 \end{matrix}$$

$$= \frac{4}{s} - \frac{\frac{4}{b}}{\frac{s}{b}+1} - \frac{\frac{4}{b}}{\left(\frac{s}{b}+1\right)^2} + \frac{4\left(\frac{1}{a}-\frac{1}{b}\right)}{\left(\frac{s}{b}+1\right)^3} \qquad (5\text{-}1)$$

$$= \frac{K_2}{s} - K_3\left[\frac{1}{\frac{s}{b}+1} + \frac{1}{\left(\frac{s}{b}+1\right)^2}\right] + \frac{K_4}{\left(\frac{s}{b}+1\right)^3} \quad .$$

The flow diagrams for this system are shown in Figure 4 and the state equations are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -b & 0 & 0 \\ 0 & b & -b & 0 \\ 0 & 0 & b & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ b \\ 0 \\ 0 \end{bmatrix} [\dot{\lambda}_q] \qquad (5\text{-}2)$$

with the output given by

$$y_q = K_2x_1 - K_3\,(x_2 + x_3) + K_4\,x_4 \quad . \qquad (5\text{-}3)$$

Before crossover, $F_1(s)$ is by-passed and the equations are

$$F_2(s) = \frac{1}{s\left(\frac{s}{b}+1\right)} = \frac{1}{s} - \frac{1/b}{\frac{s}{b}+1} \quad . \qquad (5\text{-}4)$$

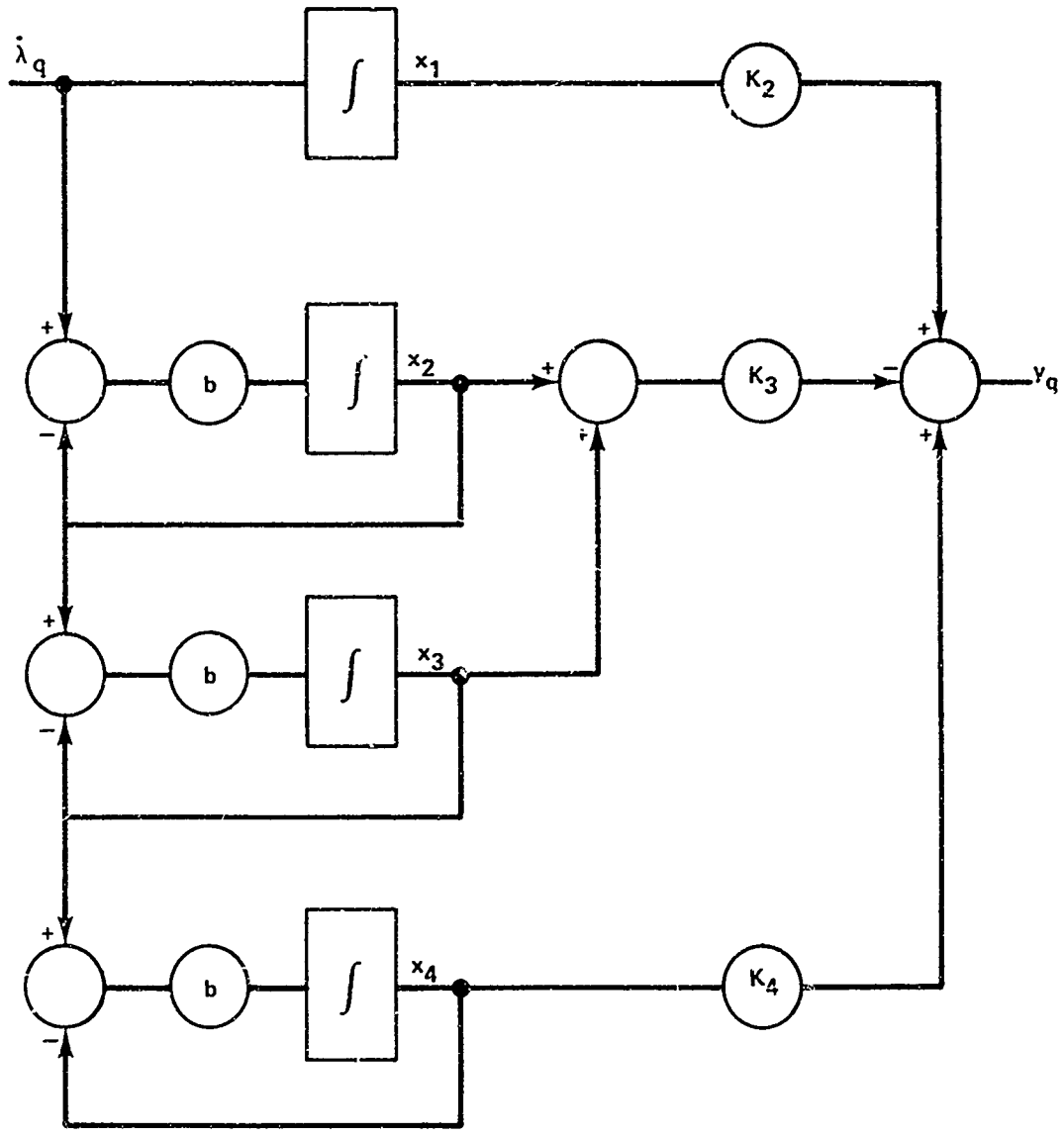


Figure 4. Seeker filter/integrator flow diagram.

The state equations are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ b \end{bmatrix} [\dot{\lambda}_q] \qquad (5\text{-}5)$$

and the output is

$$y_q = x_1 - \frac{1}{b} x_2 = x_1 - K_1 x_2 \quad . \qquad (5\text{-}6)$$

Notice that at crossover, x_4 is initialized to x_2 and $\dot{x}_2 = 0$. Also, the gravity and turning rate biases have been neglected. They will be inserted in the final equations. The equations were handled this way so that the software is simplified. Using the F and G matrixes in Equation (5-2) in the expressions for Φ and Γ Equations (3-3) give the following matrixes.

$$[\Phi] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 \\ 0 & A_{42} & A_{43} & A_{44} \end{bmatrix} \qquad [\Gamma] = \begin{bmatrix} B_{11} \\ B_{21} \\ B_{31} \\ B_{41} \end{bmatrix} \tag{5-7}$$

In these matrixes, it was observed that

$$A_{22} = 1 - B_{21} = 1 - A_{32} - B_{31} \tag{5-8}$$

$$= 1 - A_{42} - A_{32} - B_{41} \quad .$$

Also, $B_{11} = T$ where T is the update time interval. These equations can be used to reduce the number of multiplications required in executing the final set of difference equations.

The frequency response of the digital system was examined from the matrix equation

$$[X]_N = [\Phi][X]_{N-1} + [\Gamma][U]_{N-1} \tag{5-9}$$

from which

$$[X(z)] = \{z[I] - [\Phi]\}^{-1}[\Gamma][U(z)] \quad . \tag{5-10}$$

Since U is a scaler, the individual state variable responses are calculated from

$$\frac{1}{u}[X(j\omega)] = \{e^{j\omega T}[I] - [\Phi]\}^{-1}[\Gamma] \quad . \tag{5-11}$$

The output response is

$$\frac{y}{u}(j\omega) = K_2 \frac{x_1}{u}(j\omega) - K_3 \left\{ \frac{x_2}{u}(j\omega) + \frac{x_3}{u}(j\omega) \right\} + K_4 \frac{x_4}{u}(j\omega) \tag{5-12}$$

after crossover. Before crossover

$$\frac{y}{u}(j\omega) = \frac{x_1}{u}(j\omega) - K_1 \frac{x_2}{u}(j\omega) \quad . \tag{5-13}$$

Equation (5-12) was evaluated with

$$e^{j\omega T} = \cos \omega T + j \sin \omega T \tag{5-14}$$

for various update intervals and the output magnitude and phase angle were plotted. The results are shown in Figures 5 and 6. The magnitude and phase angle plots for the analog system are also shown.

The pitch/yaw control compensator was decomposed as follows:

$$F_3(s) = \frac{2.5\left(\frac{s}{\alpha} + 1\right)^2}{\left(\frac{s}{\beta} + 1\right)\left(\frac{s^2}{\gamma^2} + \frac{s}{\gamma} + 1\right)} \qquad \begin{array}{l}\alpha = 30 \\ \beta = 1000 \\ \gamma = 300\end{array}$$

$$= \frac{2.5\,A}{\frac{s}{\beta} + 1} + \frac{2.5\,(BS + C)}{\frac{s^2}{\gamma^2} + \frac{s}{\gamma} + 1} \tag{5-15}$$

in which

$$A = \frac{\gamma^2}{\alpha^2}\left\{\frac{(\alpha - \beta)^2}{\gamma^2 + \beta^2 - \gamma\beta}\right\} \tag{5-16}$$

$$B = \frac{\beta}{\alpha^2}\left\{\frac{\gamma^2 - \alpha^2 - \gamma\beta + 2\alpha\beta}{\gamma^2 + \beta^2 - \gamma\beta}\right\} \tag{5-17}$$

$$C = 1 - A \quad . \tag{5-18}$$

These equations are diagrammed in Figure 7. From this diagram the state equations are

$$\begin{bmatrix}\dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11}\end{bmatrix} = \begin{bmatrix}\beta & 0 & 0 \\ 0 & -\gamma & -\gamma \\ 0 & \gamma & 0\end{bmatrix}\begin{bmatrix}x_9 \\ x_{10} \\ x_{11}\end{bmatrix} + \begin{bmatrix}-\beta \\ \gamma \\ 0\end{bmatrix}\begin{bmatrix}u_1\end{bmatrix} \tag{5-19}$$

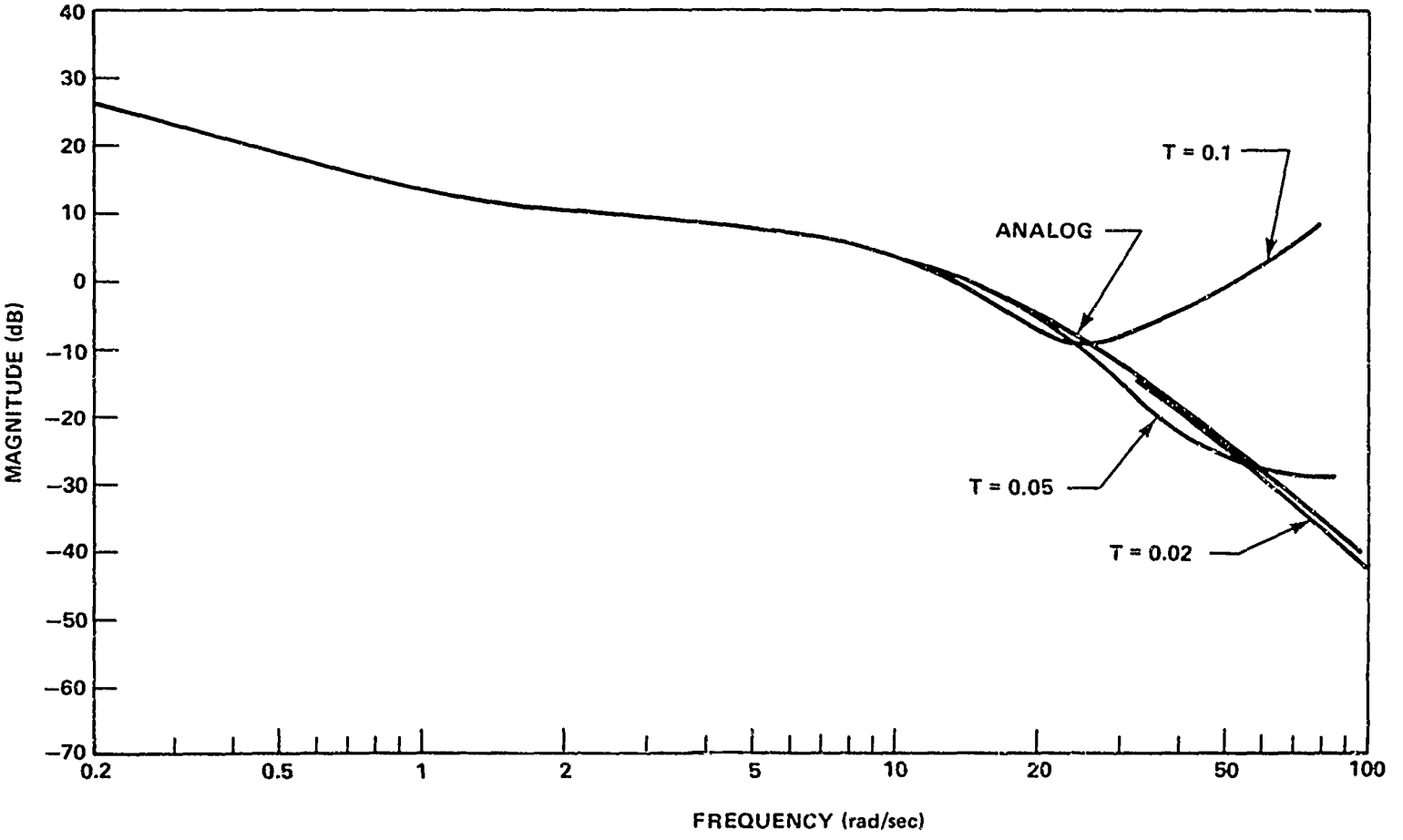


Figure 5. Seeker filter/integrator magnitude response state variable form.

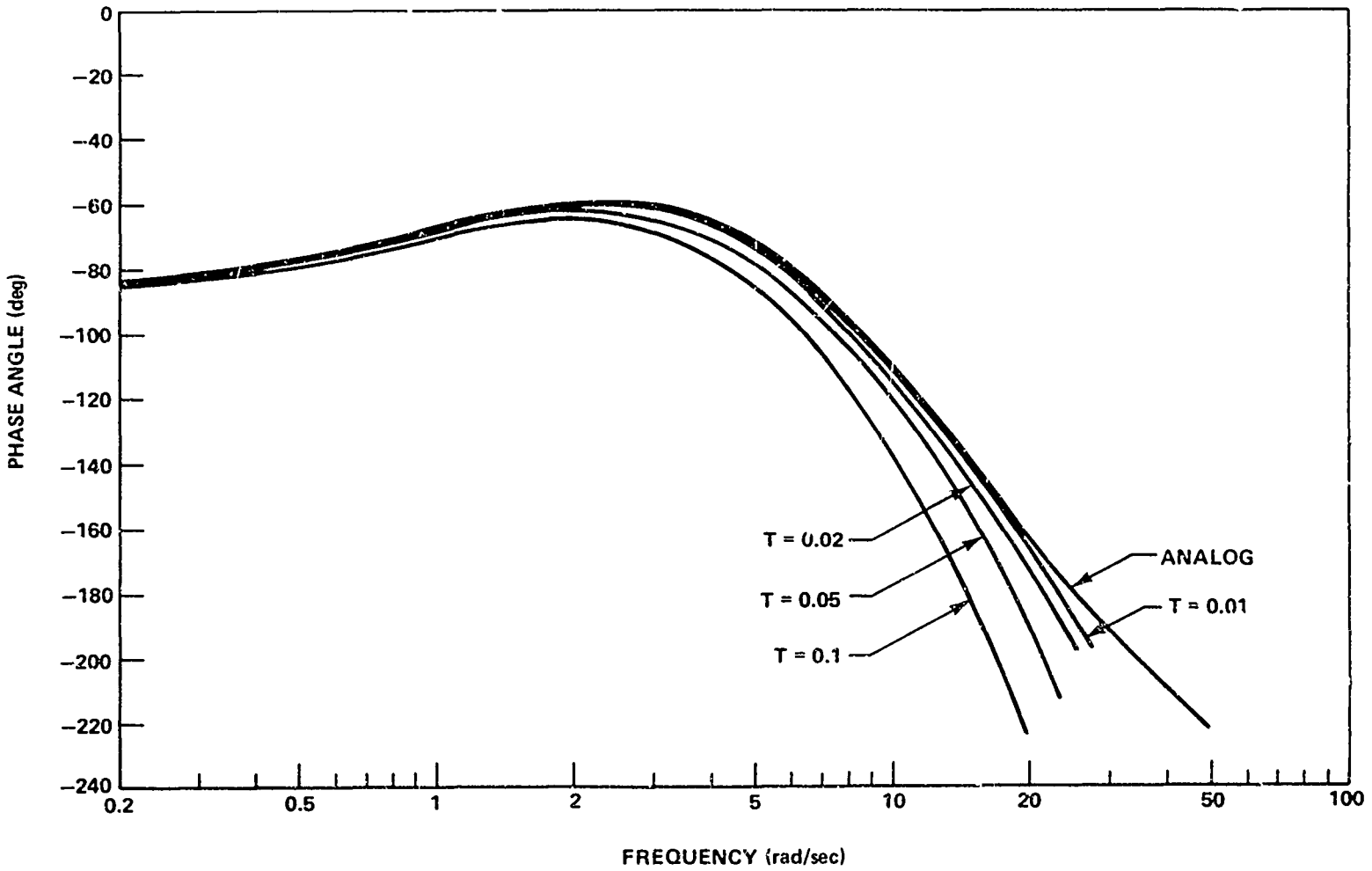


Figure 6. Seeker filter/integrator phase angle response state variable form.

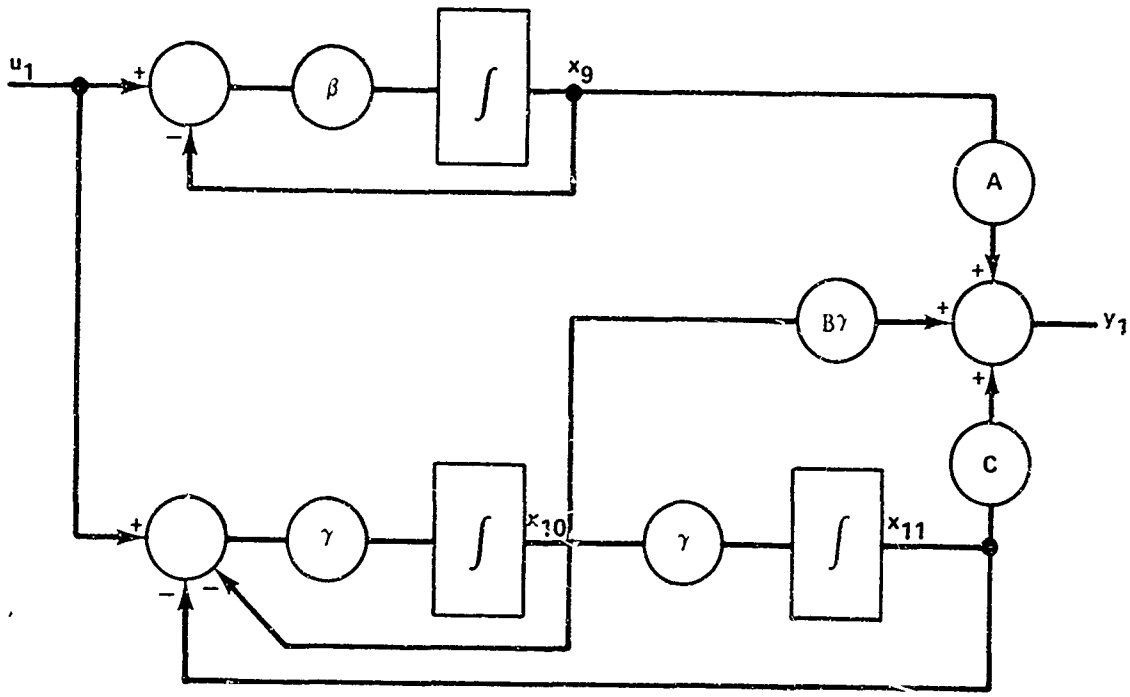


Figure 7. Gyro compensator flow diagram.

and the output is

$$y_1 = Ax_9 + B\gamma\, x_{10} + Cx_{11} \quad . \tag{5-20}$$

Again using Equations (5-2) the matrixes for the difference equations were found to be

$$[\Phi] \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & C_{23} \\ 0 & C_{32} & C_{33} \end{bmatrix} \qquad [\Gamma] = \begin{bmatrix} D_{11} \\ D_{21} \\ D_{31} \end{bmatrix} \quad . \tag{5-21}$$

The following relationships exist among the matrix elements and the gains were redefined as

$$\begin{aligned} D_{11} &= 1 - C_{11} & C &= 1 - A \\ C_{32} &= -C_{23} = D_{21} & K_5 &= A \\ C_{33} &= 1 - D_{31} & K_6 &= B\gamma \quad . \\ & & K_7 &= 2.5 \end{aligned} \tag{5-22}$$

Values for the analog system gains found from the partial fraction expansion are

$$\begin{aligned} A &= 119.1012658 \\ B\gamma &= -53.67088608 \\ C &= -118.1012658 \quad . \end{aligned} \tag{5-23}$$

Again the frequency response of the digital system was found by solving the matrix equation

$$\frac{1}{u}\,[X\ (j\omega)] = \left\{e^{j\omega t}\ [I] - [\Phi]\right\}^{-1}\ [\Gamma] \tag{5-24}$$

with the output

$$\frac{Y}{u}\ (j\omega) = A\ \frac{x_9}{u}\ (j\omega) + B\gamma\ \frac{x_{10}}{u}\ (j\omega) + C\ \frac{x_{11}}{u}\ (j\omega) \quad . \tag{5-25}$$

This was done for various data intervals T. The results are shown in Figures 8 through 10 and should be compared with the analog system response which is also plotted.

The roll compensator whose transfer function is

$$F_4(s) = \frac{0.6\ \left(\frac{s}{35} + 1\right)^2}{\left(\frac{s}{1000} + 1\right)\left(\frac{s^2}{300^2} + \frac{s}{300} + 1\right)} \tag{5-26}$$

is very similar to the pitch/yaw compensator. The Φ and Γ matrixes are, in fact, identical. The gains are replaced by

$$\begin{aligned} A &= 86.6032033 \\ B\gamma &= -43.77938517 \\ C &= -85.6032033 \quad . \end{aligned} \tag{5-27}$$

Basically, the work in this section defines the difference equations to be used for the control compensator and filter/integrator software. The r-channel equations are the same as for the q-channel except that the q-channel has gravity and turning rate bias added. The final equations are shown in Appendix A which was previously published as Internal Technical Note ITN-RGG-1-75, Digital Autopilot Software System Specification (T-6 Version), January 1975.

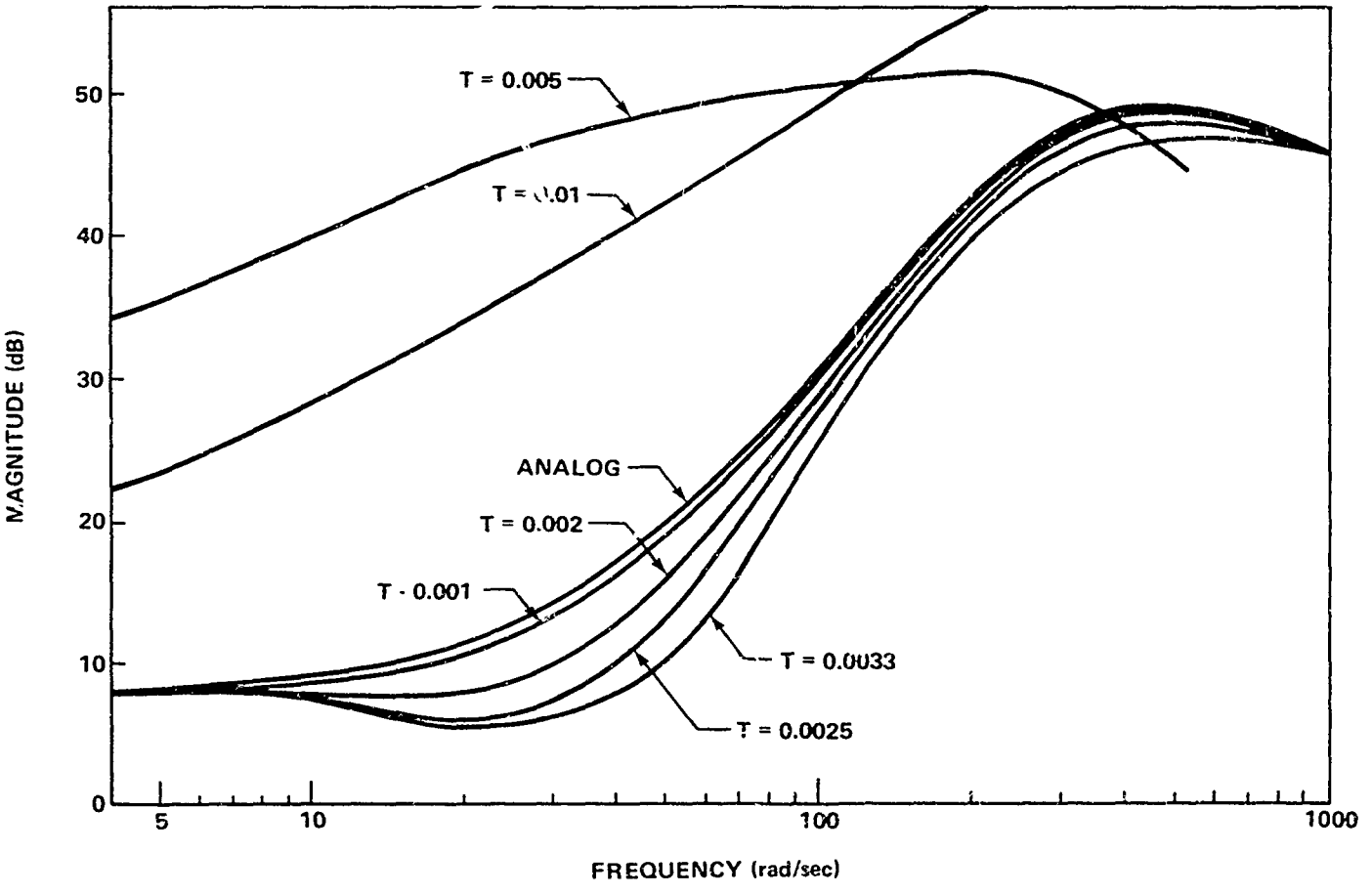


Figure 8. Pitch gyro compensator magnitude response state variable form.

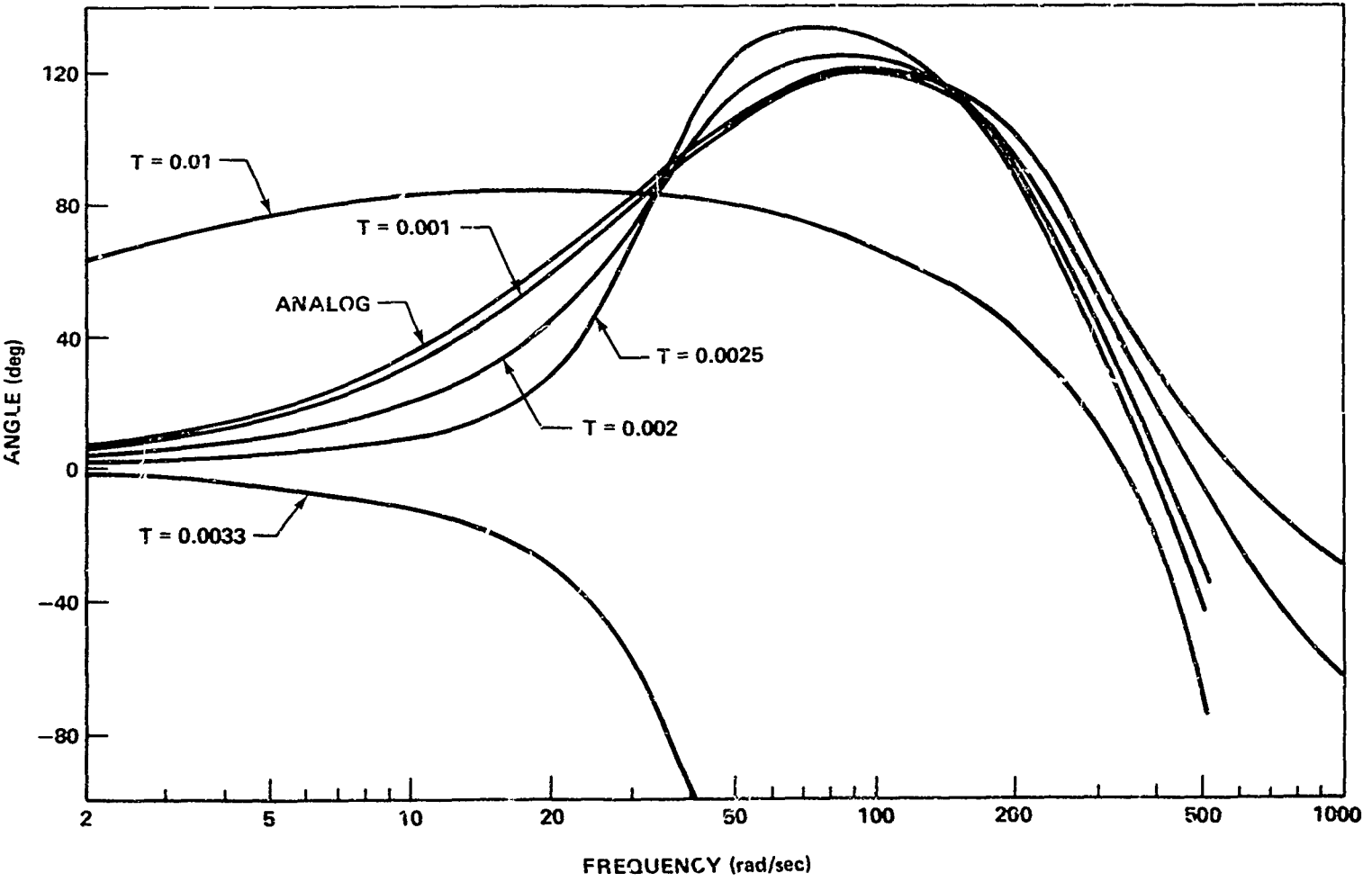


Figure 9. Gyro compensator phase angle response state variable form.

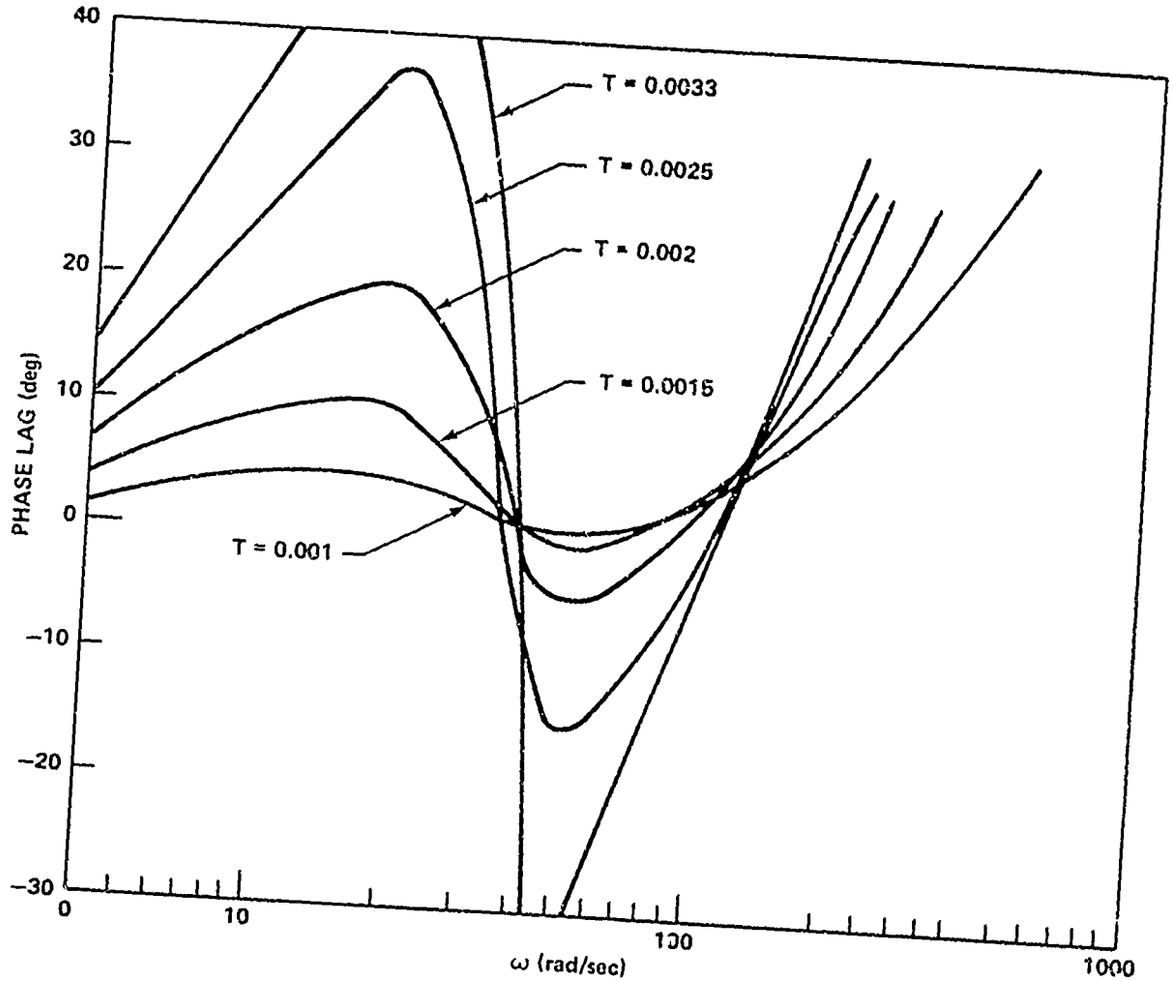


Figure 10. Gyro compensate phase lag state variable form.

Among other state variable based systems which were examined briefly, a series oriented system whose F matrix had very few off-diagonal non-zero elements was examined. However, the Φ matrix for this type of system has a relatively large number of off-diagonal non-zero elements caused by the fact that state variables toward the output are coupled all the way back to the input. If these off-diagonal elements cannot be neglected, the computational load for this system is much greater than for a parallel based one.

6. Modification of the State Variable System

It can now be seen that the difference equations represented by (5-3), (5-7), and (5-9) for the filter/integrators and (5-20), (5-21), and (5-9) for the gyro compensators are not entirely satisfactory. Several factors lead one to this conclusion among which are the number of multiplications in the equations, the anticipated speed of a state of the art microprocessor, the loop crossover frequency data given in Appendix B, and the information in the frequency response curves. Recall that an upper limit of 10,000 multiplications per second was predicted for the microprocessor. From the crossover frequency data and frequency response curves, it appears that performance balance can be maintained among the various algorithms by making the roll update rate highest, the pitch/yaw rates about 1/2 the roll rate and the q/r channel rates about 1/4 to 1/2 the pitch/yaw rates. Using these rate ratios weighted by the number of multiplications in each set of equations showed that the phase margin decrease of 5 degrees could not be met in any loop even at 20,000 multiplications per second. It was decided that either the rates had to be reduced or the number of operations in the equations decreased if the digital autopilot were to be successful.

The primary problem in the digital system is a failure to faithfully reproduce the analog system's phase angle. Magnitude is generally not much of a problem. Phase angle problems are caused by:

a) Lag caused by zero order holds

b) Failure of the digital system to adequately realize the analog system zeros

c) Poles and zeros in the digital system outside the primary strip in the s-plane

d) Quantization effects due to word length.

Zero order hold lag can be compensated for by a prediction. This could be done using the transition matrix but this can increase the number of computations to the extent that not much is gained by the accompanying allowable decrease in data rate. The prediction method should be very simple, preferably in this case requiring only a

multiplication or two. It was assumed that the system timing would be such that a full zero order hold would appear at the microprocessor input in addition to the one required at the output. These zero order holds were compensated for by a simple linear prediction on the output. This prediction is given by

$$z_N = 2\ y_N - y_{N-1} \tag{6-1}$$

which essentially consists of three additions. The compensated output is z_N and y_N and y_{N-1} are the last two uncompensated outputs. The magnitude and phase characteristics of this equation are shown in Figure 11. The response for two zero order holds for which this equation compensates is shown also. In passing, the characteristics of the compensating equation for a single zero order hold

$$z_N = 3/2\ y_N - 1/2\ y_{N-1} \tag{6-2}$$

is shown in Figure 11. It can be seen that the full interval predictor removes all but about 0.5 degree or less of the zero order hold lag up to $\omega T = 0.2$ at a gain cost of 1/3 dB or less. At $\omega T = 0.4$, the remaining phase lag is 3 degrees or less at a gain cost of 1.2 dB or less. The full interval compensation given by Equation (6-1) was added to each of the equation sets in the q-channel, r-channel, pitch, yaw, and roll. The manner in which the equations are specified cause a slight over compensation of 1/2 the pitch/yaw gyro interval in the q/r channels. This causes no significant effect in the seeker loop and is slightly beneficial.

The zero shift problem was primarily troublesome in the gyro compensators. The main purpose of each compensator is to provide enough lead angle near the zero dB crossover frequency by way of a double real zero to stabilize each gyro loop. In pitch and yaw, this zero was placed at 30 radians per second and in roll at 35 radians per second. If the output expression for y_N uses the analog system gains A, B_γ, and C it is found that the location of this double zero in the primary strip in the s-plane depends on the update rate in the digital system. Although the magnitude of the zero remained about right, it breaks into a complex pair. The locus of the comples pair is shown in Figure 12 for pitch and yaw as a function of the data interval. The difference in phase lead for the double real zero and the complex pair is shown in Figure 13. Unfortunately, the lag due to the fact that the zero is complex was close to a maximum at about the system crossover frequency at what was thought to be the maximum acceptable data rate and the lag amounted to 15 to 20 degrees.

This led to a decision to recompute the system gains A, B_γ, and C such that the digital system has a double real zero at 30 radians per second in pitch and yaw. A similar thing was done in roll for the zero at 35 radians per second. This procedure may appear formidable

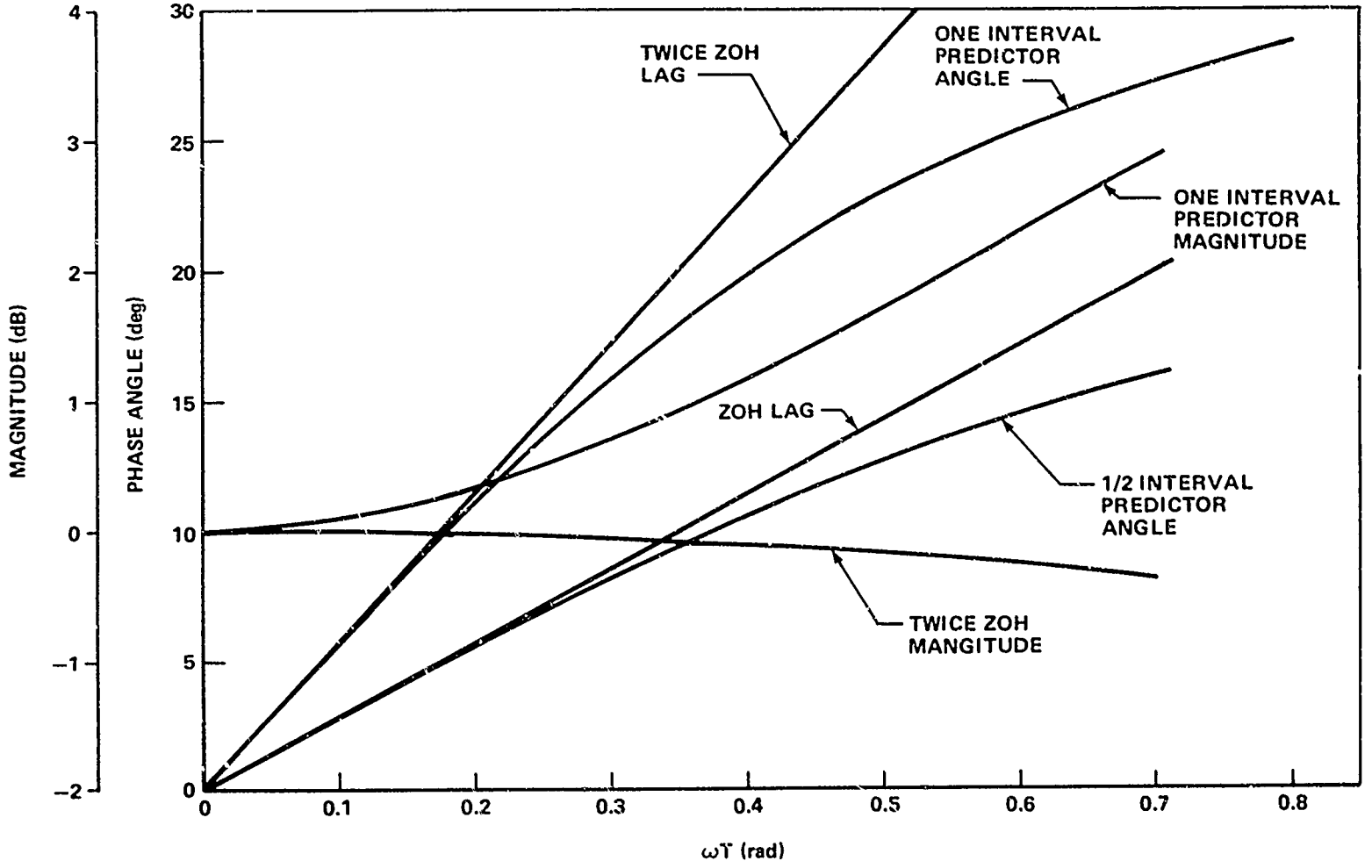


Figure 11. Zero order hold and compensating predictor responses.

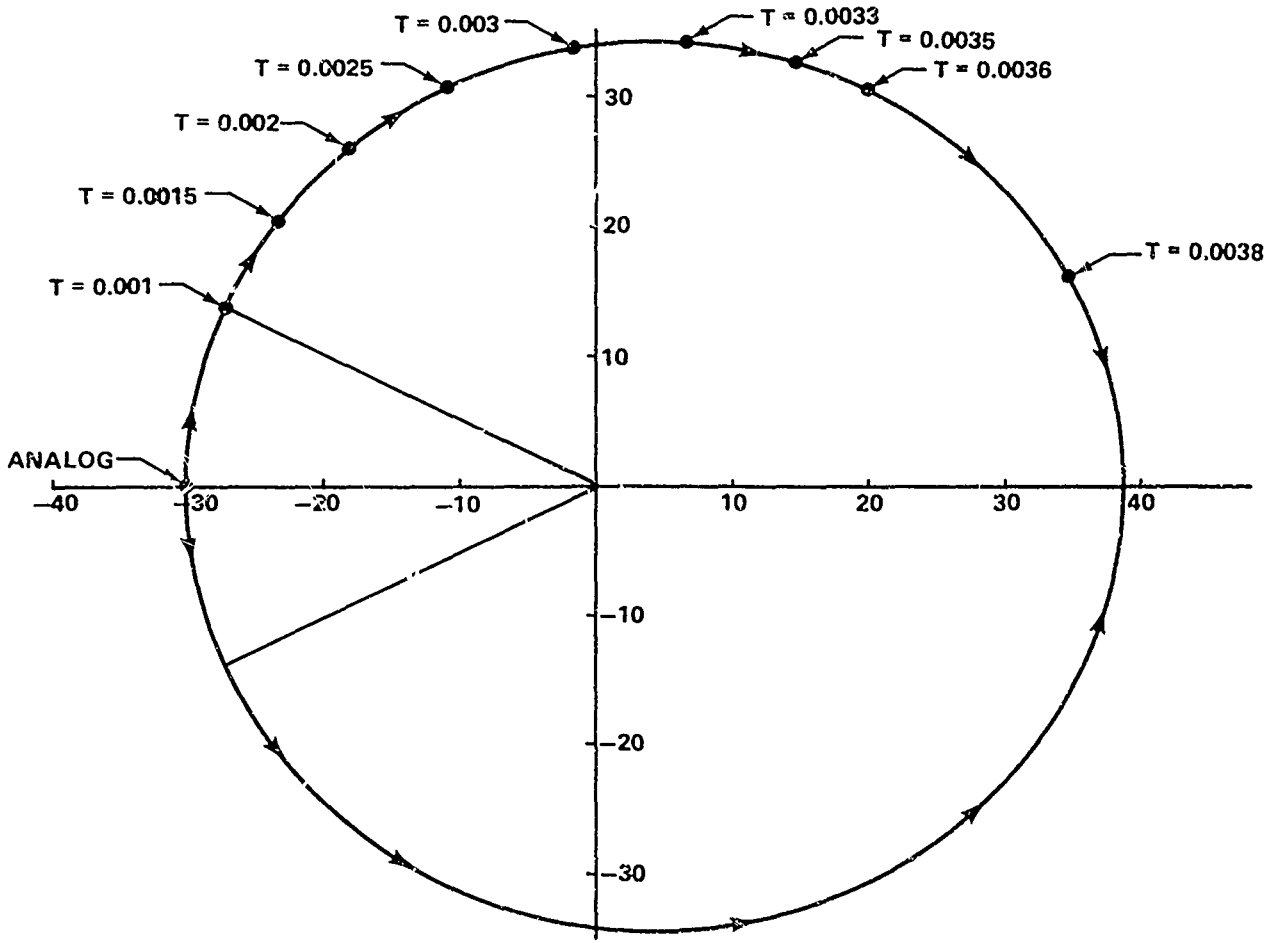


Figure 12. S-plane compensator zero locus.

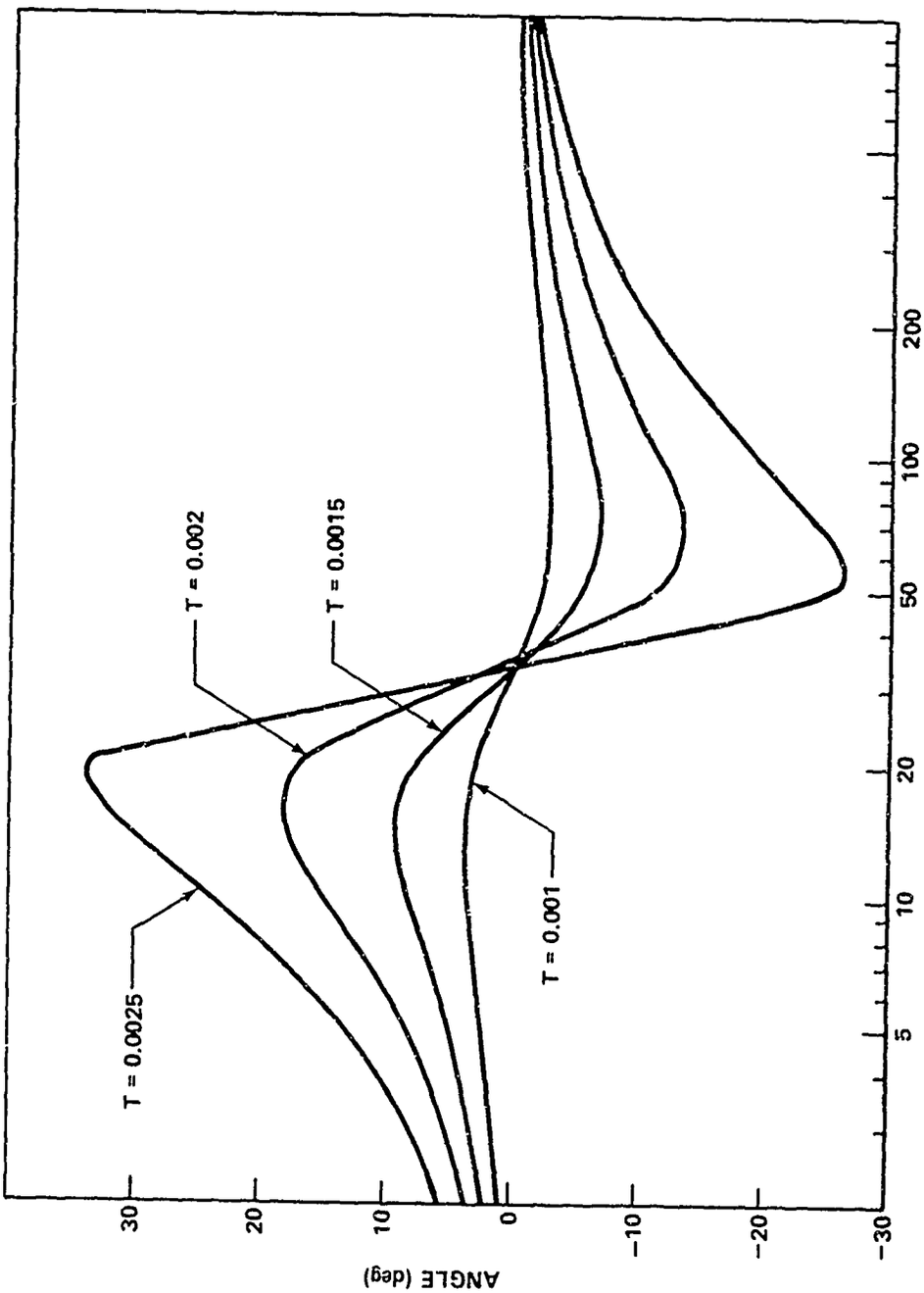


Figure 13. Phase angle difference between double real zero and complex pair.

but it required only a few instructions added to the frequency response program on the 9830 calculator. Conceptually, the matrix equation

$$[X(z)] = \{z[I] - [\Phi]\}^{-1} [\Gamma] [U(z)] \tag{6-3}$$

was solved for the states and the output equation

$$y(z) = A\, x_9(z) + B_Y\, x_{10}(z) + C\, x_{11}(z) \tag{6-4}$$

was forced to have a numerator term equal to

$$(z - e^{-\alpha T})^2 \tag{6-5}$$

by equating coefficients and solving for the required gains A, B_Y, and C. These recomputed gains are the ones specified in Appendix A for the given data rates in pitch, yaw, and roll.

No attempt was made to compensate for pole and zero effects outside the primary strip and, in general, nothing can be done about them. Word length is discussed in the next section.

7. Word Length

Word length selection for the digital autopilot was not considered a firm goal for this part of the project. Final word length will be a natural result obtained from the feasibility demonstration. A 16-bit specification was tentatively set from the beginning for the microprocessor based on prior experience and the desire to have a flexibility not available at shorter word lengths. This investigation was undertaken to insure that single precision 16-bit arithmetic was indeed adequate, to get data on A/D quantization levels, and to see if a shorter word length might be allowable.

The HP 9830 is limited in word length studies since it appears to be a decimal machine to the operator. There is no provision for machine language accessibility and therefore no bit manipulations. Word length effects were studied by truncating decimal digits in system gains and matrix elements and, to a limited extent, by octal digit truncation of the same constants. The results of this truncation shown in Figures 14 through 16 for the seeker filter/integrator and the gyro compensator indicate that the state variable derived formulation is relatively insensitive to short word length quantization errors as compared with the recursive feedback formulation discussed earlier. The two-digit data for the gyro compensator varied too radically to be plotted. Errors for the four-digit curves were too small to plot. On the basis of these curves, it was concluded that three significant octal digits were all that are needed for the system gains and matrix elements.

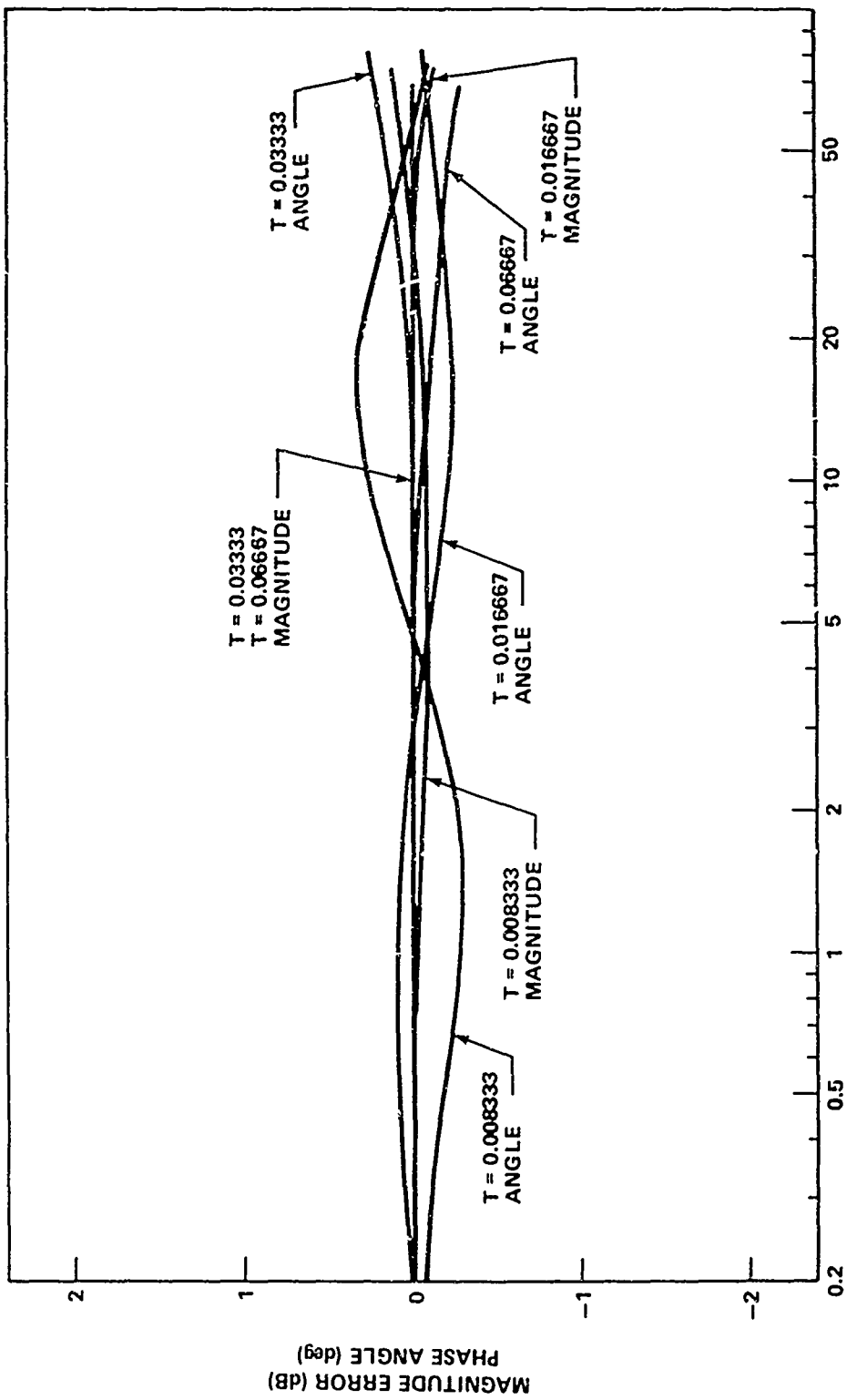


Figure 14. Seeker filter/integrator word length error state variable form 3 decimal digits.

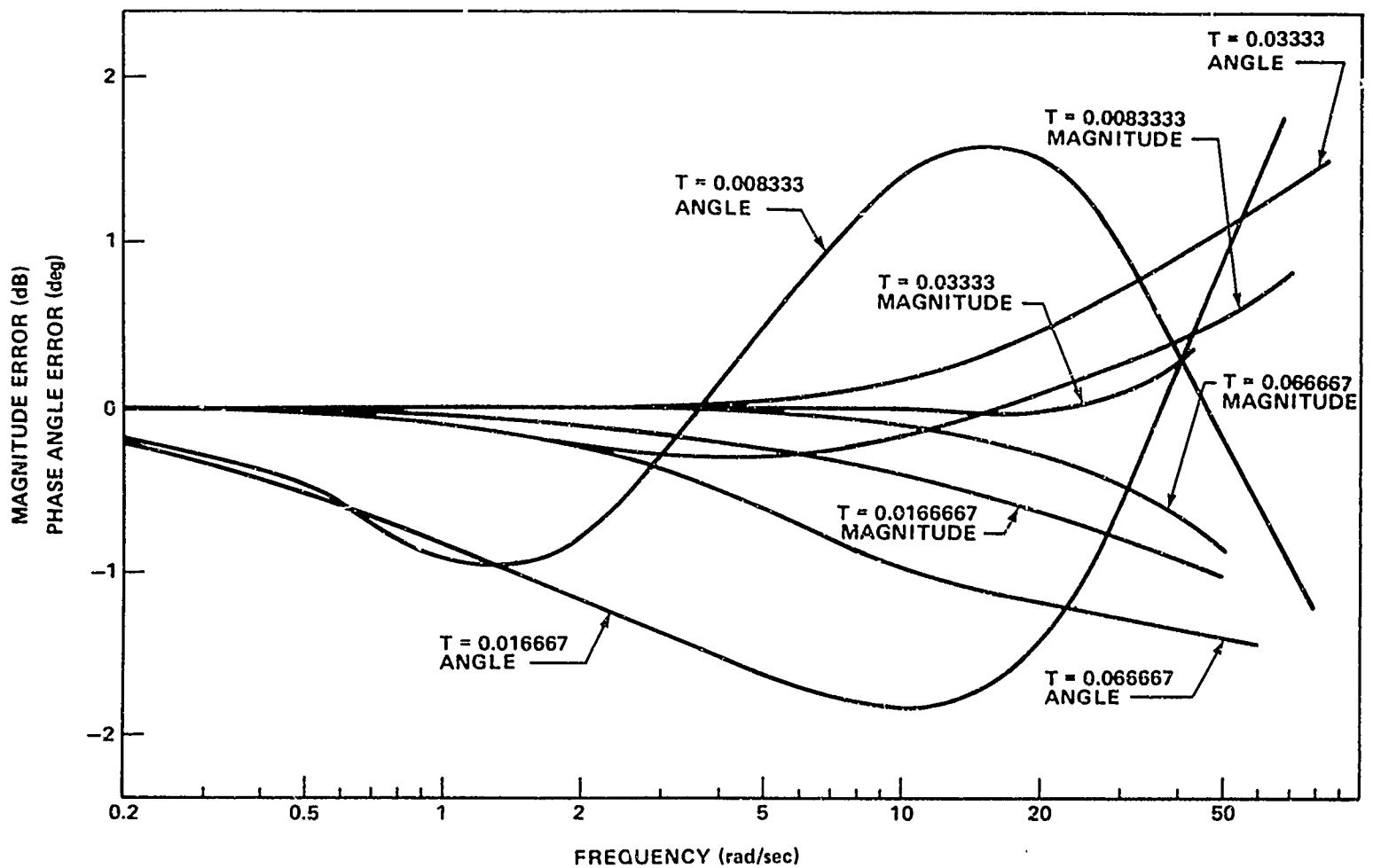


Figure 15. Seeker filter/integrator word length error state variable form 2 decimal digits.

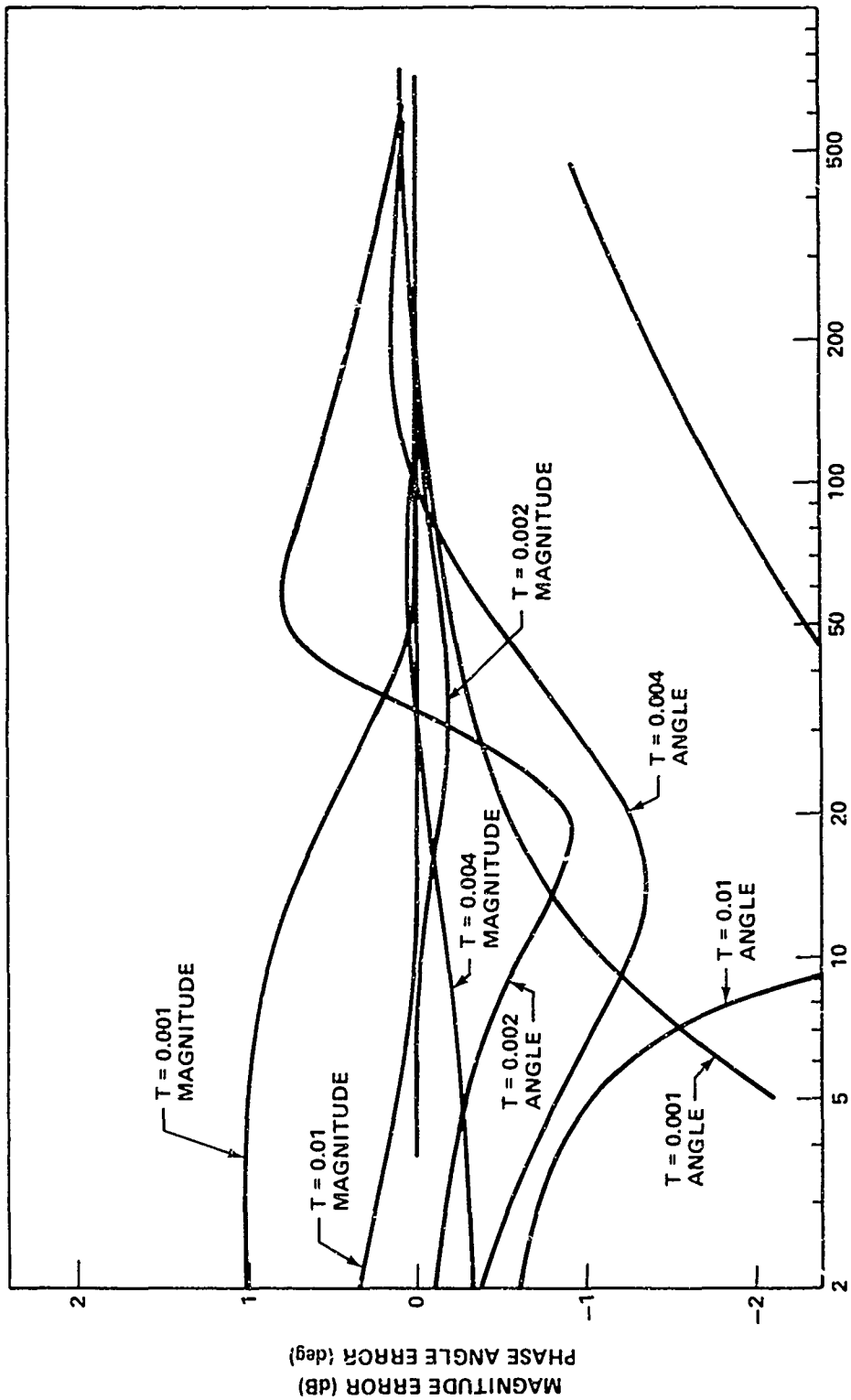


Figure 16. Gyro compensator word length error state variable form 3 decimal digits.

Furthermore, this same quantization level should be adequate for the system variables and, therefore, 10-bit A/D conversion will be adequate. Actually, the only known computations that require full 16-bit precision are the gyro output equations for y_1, y_2, and y_3 in which a small algebraic sum is formed after multiplication of individual variables by high gains. The next critical computation is the integration accuracy in indirect fire in which the missile must be turned so as to bring the target within the seekers' field of view. A 12-bit processor could yield precision close to one part in 1000, however, which should be more than adequate for this computation.

8. Software Analysis Summary

As noted previously, the software system specification based on the analyses discussed in this report is included as Appendix A. Before a final decision could be made on data rates, information was needed concerning loop crossover frequencies and the internal seeker characteristics. The modeling and analyses relative to this work are detailed in Appendix B. A software flow diagram is shown in Appendix A, Figure A-2. In summary, the software consists of two nearly identical sets of difference equations for the q and r channel filter/integrators, two identical sets of equations for the pitch and yaw compensators, and a set of equations for the roll compensator. These equations are based on the state variable equations discussed in Section 5. These equations were modified for zero order hold delay and corrected for unsatisfactory zero locations in the gyro compensators. With these modifications to the basic state variable derived equations, frequency response error curves are shown in Figures 17 through 21. From these curves, rates were chosen to be 500 cycles per second in roll, 250 cycles per second in pitch and yaw, and 71.4 cycles per second for the q/r channel software. The 71.4 cycles per second rate is one update every 7 roll cycles and takes into account the fact that it is desirable to place this rate between expected sidebands from the 20 Hz sampler in the seeker. This problem is discussed further in Appendix B.

The equations could be slightly simplified by neglecting terms that are not significant. The gyro compensator pole, for example, at 1000 radians per second is unnecessary in the digital system and could have been omitted. A few other terms in the Φ and Γ matrixes could also be neglected. In fact, a first order approximation to these matrices was studied and only a small difference in performance compared to the original system was observed. None of these simplifications were incorporated in the design.

The specified difference equations updated at the designated rates on a 16-bit microprocessor will not degrade the system gain margin more than 1 dB or the phase margin more than 5 degrees. Low frequency or "dc" response is immeasurably affected. The data rates are high

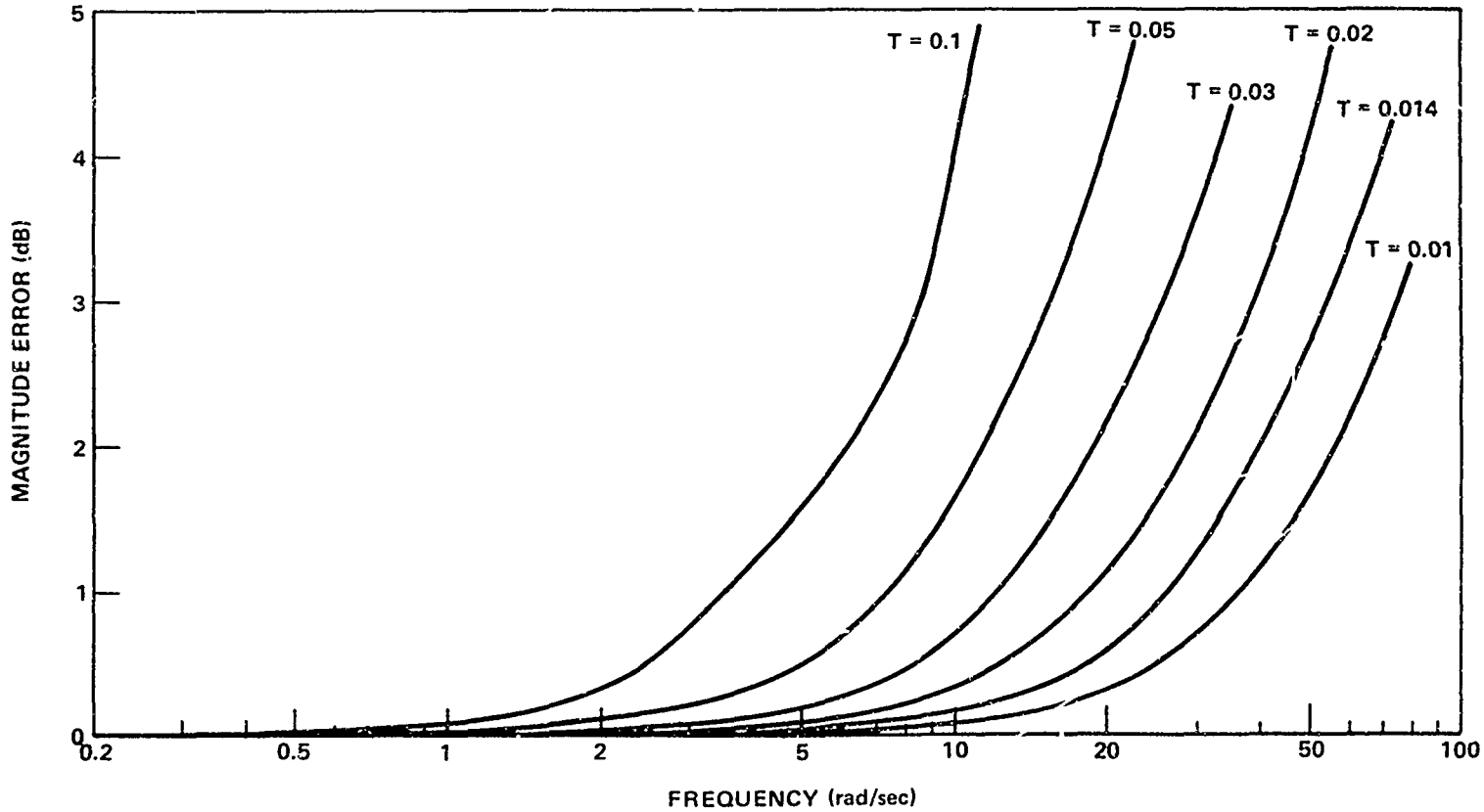


Figure 17. Seeker filter/integrator magnitude error final design.

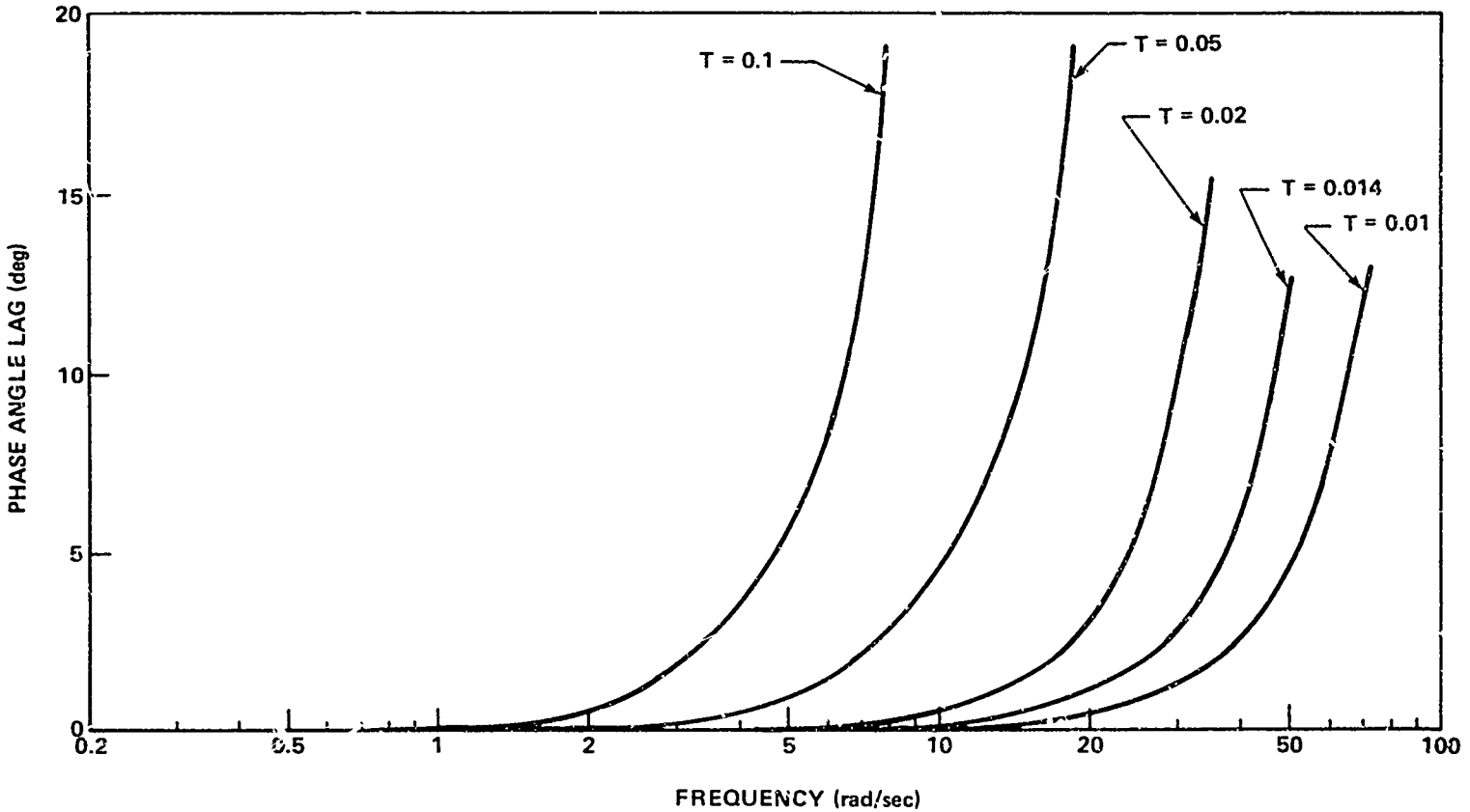


Figure 18. Seeker filter/integrator phase error final design.

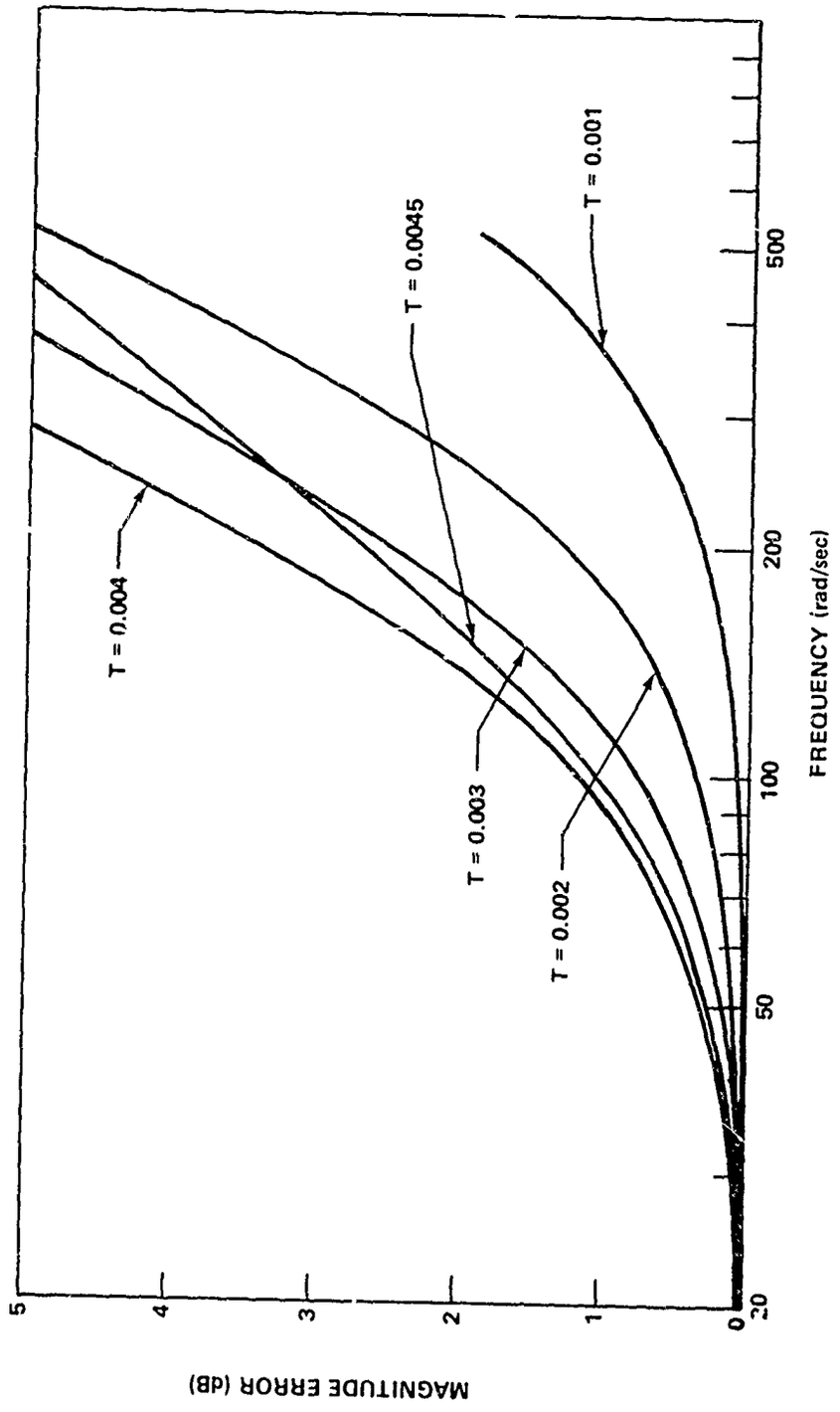


Figure 19. Gyro compensator magnitude error final design.

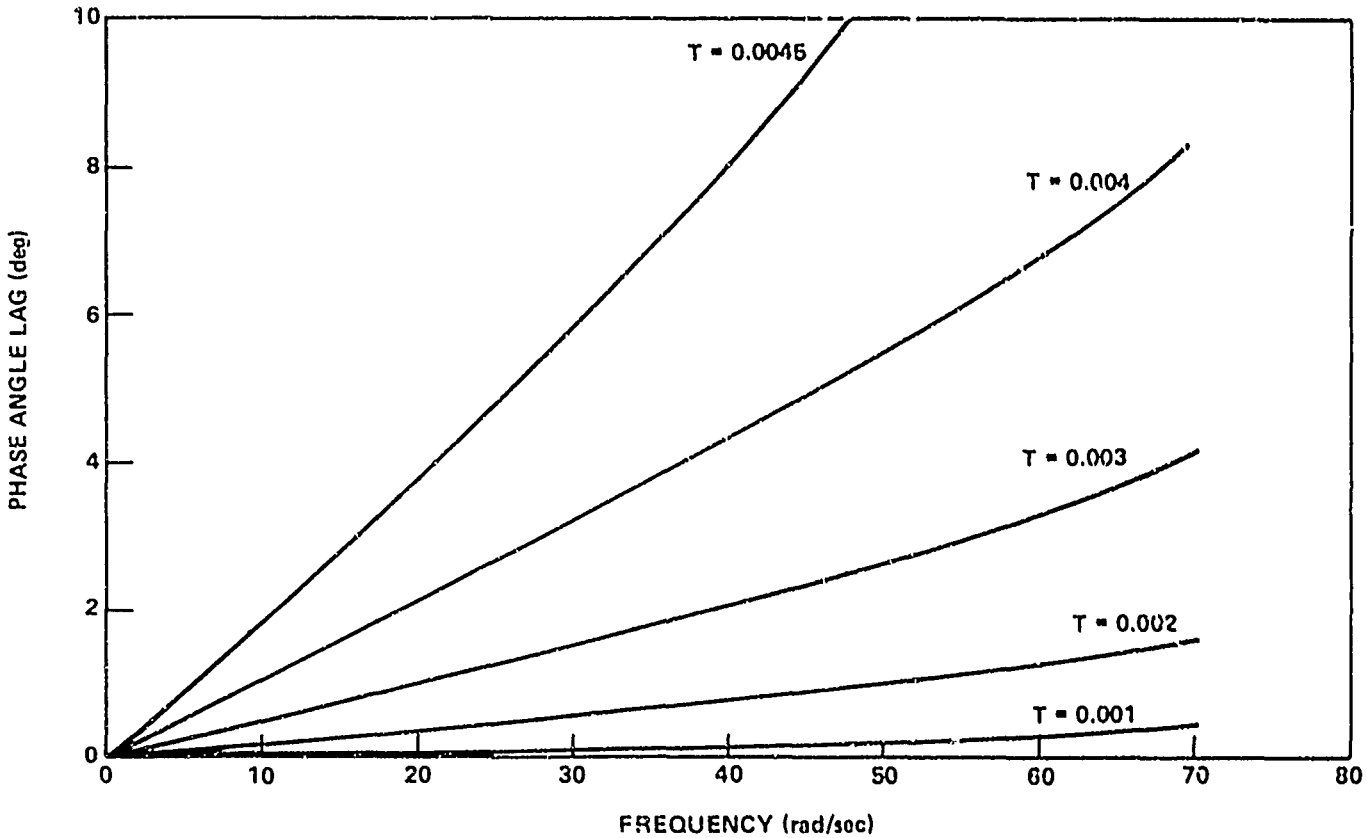


Figure 20. Gyro compensator phase error final design.

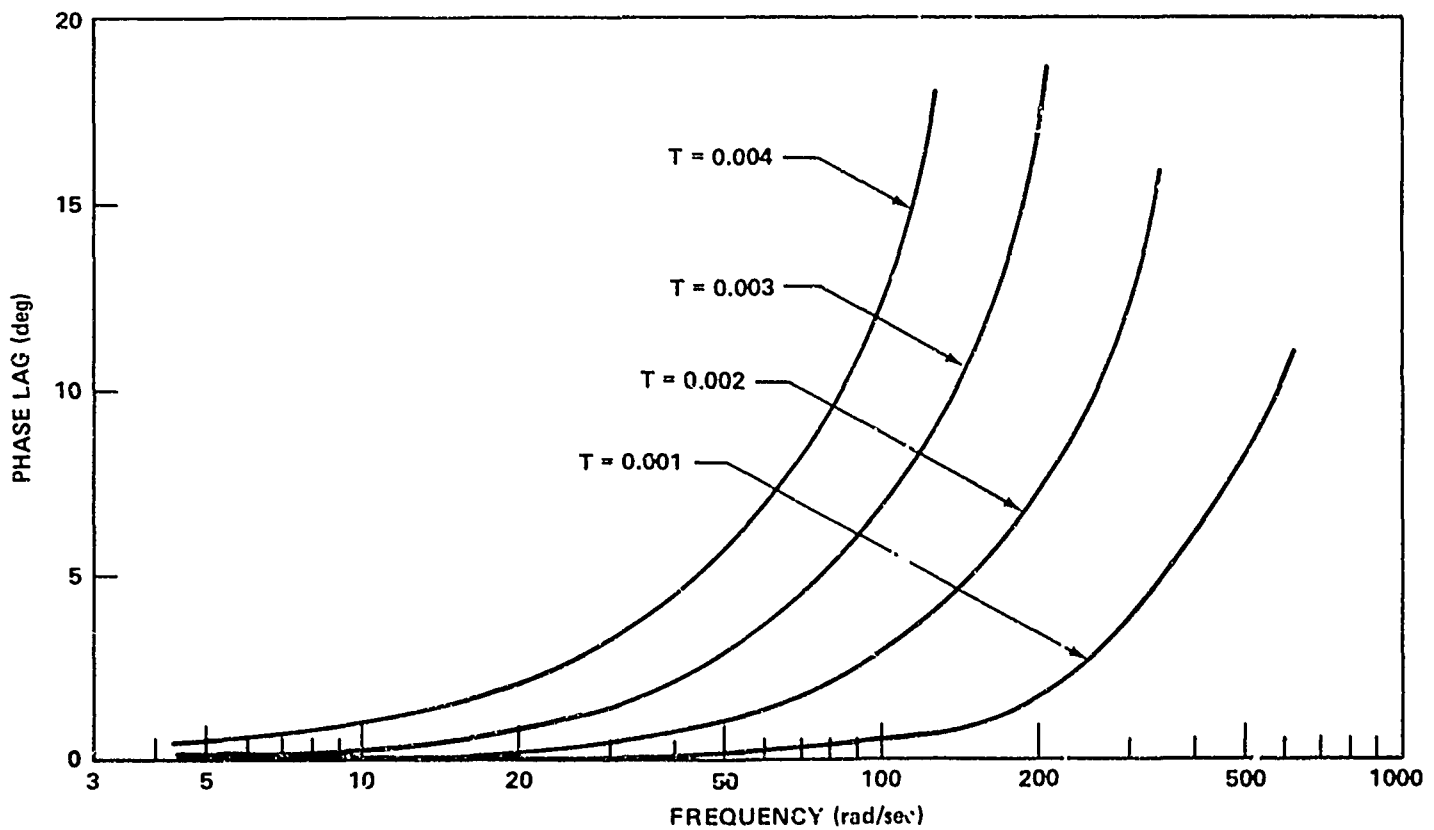


Figure 21. Gyro compensator phase error final design.

enough that noise problems caused by a lack of high frequency roll-off should not be encountered. The execution rate required of the microprocessor is high and might be eased by a less conservative software design but it is believed that a well-designed microprocessor using available technology can meet the challenge.

Appendix A. DIGITAL AUTOPILOT SOFTWARE SYSTEM SPECIFICATION (T–6 VERSION)

I. System Requirements

This document is intended to serve as a combined software system requirement and specification for the modular missile digital autopilot. The software specified herein is a digital implementation of the analog functions shown in Figure A-1. The primary reference sources for the analog system were North American Rockwell document 772-SA-20-72, "Scaled Down THFTY Autopilot Requirements" and Army Missile Command Technical Report RG-73-18. "T6 Missile Description." A general goal to be met is that the software is to be implemented on a state-of-the-microprocessor which is envisioned as a small digital machine with memory access time around 2 μsec, word length no greater than 16 bits, and about a 1K memory. Another goal was arbitrarily established as a performance measure, i.e., the system phase margins and gain margins should not be degraded more than 5 degrees and 1 dB respectively and performance should improve over these figures below the crossover frequencies. The specified software will meet these goals. The justification for the software described in this document will be given in an Army Missile Command Technical Report to be published.

II. Comptational Flow Diagram

The digital system gross computational flow diagram is shown in Figure A-2. The correlation between blocks in Figures A-1 and A-2 is almost one to one with the main exception being an interchange of the filter/integrator and guidance filter blocks. The block update rates are shown in Figure A-2. The 71.4/second rate is approximate and corresponds to a 14-millisecond time interval or one update every 7 roll cycles. Pitch and yaw compensators are updated every other roll cycle.

The inputs to the digital system are:

a) $\dot{\lambda}_q, \dot{\lambda}_r$ - seeker pitch and yaw angular error rates

b) C_q, C_r - seeker signal crossover discretes in pitch and yaw

c) θ, ψ, Φ - gyro pitch, yaw, and roll angles.

The outputs from the digital system are z_1, z_2, z_3 - pitch, yaw, and roll actuator signals.

In addition, a target detection discrete, D, and a variable bias, b, are required in the indirect fire mode.

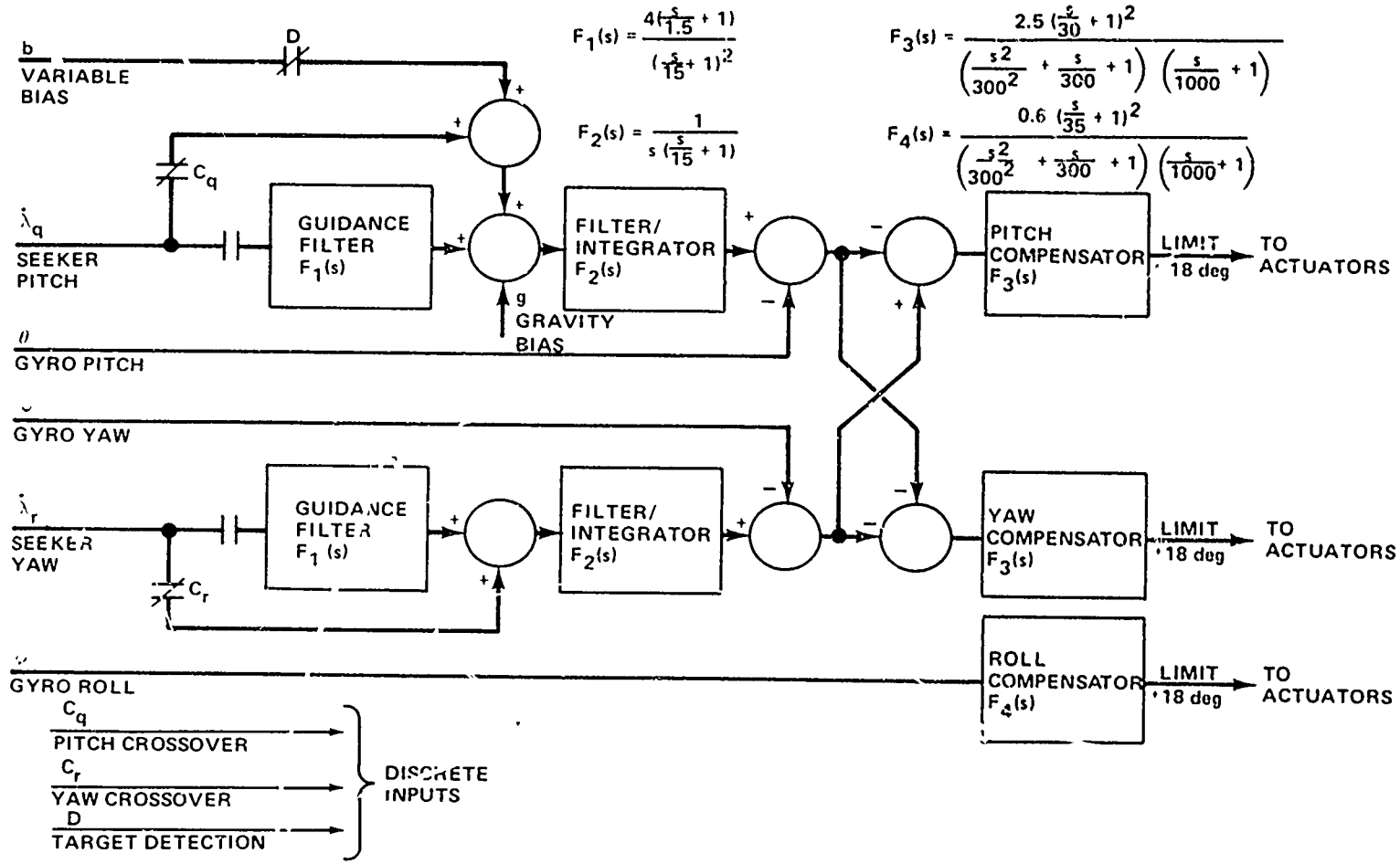


Figure A-1. Analog functions to be implemented digitally.

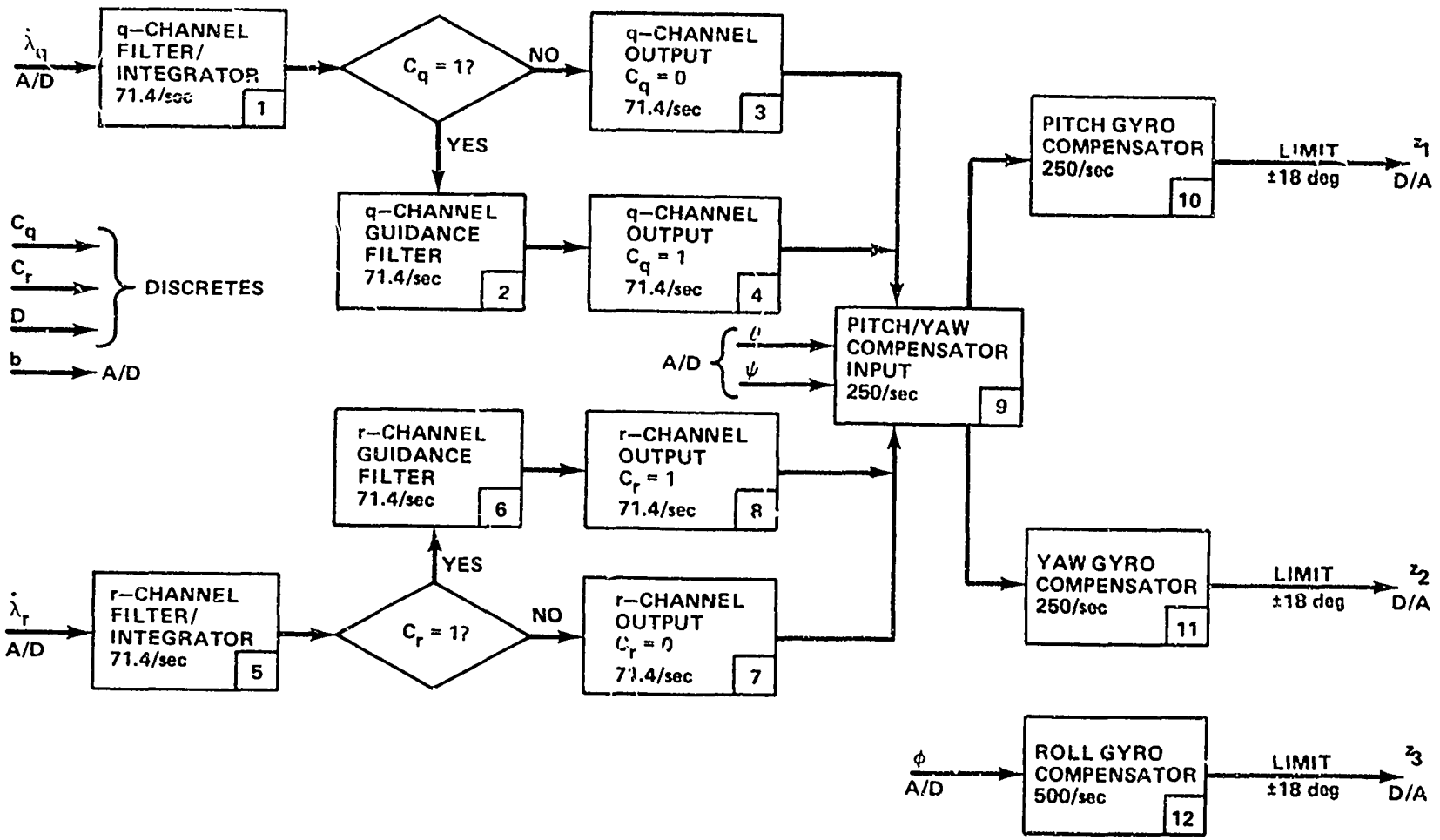


Figure A-2. Digital system flow diagram.

III. Difference Equations

The difference equations to be executed by each block in Figure A-2 are shown in the following numbered paragraphs. The paragraph numbers correspond to the numbers in the lower right hand corners of the blocks in Figure A-2. The symbols in the equations are identified in the glossary of this appendix. In general, the subscript "N" indicates the new computed value of a variable and "L" indicates the value of a variable or measurement saved from the previous cycle for the variable being updated. These difference equations were derived by standard state variable methods and modified to account for an incorrect zero location and zero order hold delays.

1. q-Channel Filter/Integrator Update Time Interval: 0.014 second

Equations:

$$x_{1N} = x_{1L} + B_{11}(\dot{\lambda}_{qL} + g)$$
$$x_{2N} = x_{2L} + B_{21}(\dot{\lambda}_{qL} - x_{2L}) \qquad \text{(A-1)}$$

Input: $\dot{\lambda}_{qL}$ from seeker A/D

Output: x_{1N}, x_{2N} to storage and q-channel output

Storage: x_{1L}, x_{2L}

Constants: g, B_{11}, B_{21}

2. q-Channel Guidance Filter Update Time Interval: 0.014 second (Executed only with $C_q = 1$)

Equations:

$$x_{3N} = x_{3L} + A_{32}\,(x_{2L} - x_{3L}) + B_{31}\,(\dot{\lambda}_{qL} - x_{3L})$$

$$x_{4N} = x_{4L} + A_{42}\,(x_{2L} - x_{4L}) + A_{32}\,(x_{3L} - x_{4L}) \qquad \text{(A-2)}$$
$$+ B_{41}\,(\dot{\lambda}_{qL} - x_{4L})$$

Inputs: $\dot{\lambda}_{qL}$ from seeker A/D and x_{2L} from q-channel filter/integrator

Outputs: x_{3N}, x_{4N} to storage and q-channel output ($C_q = 1$)

Storage: x_{3L}, x_{4L}

Constants: B_{31}, A_{42}, A_{32}, B_{41}

3. q-Channel Output ($C_q = 0$) Update Time Interval: 0.014 second

Equations:

$$y_{qN} = x_{1N} - K_1 x_{2N}$$
$$z_{qN} = 2y_{qN} - y_{qL} \qquad \text{(A-3)}$$

Inputs: x_{1N}, x_{2N} from q-channel filter/integrator

Output: z_{qN} to pitch/yaw compensator input and y_{qN} to storage

Storage: y_{qL}

Constant: K_1

4. q-Channel Output ($C_q = 1$) Update Time Interval: 0.014 second

Equations:

$$y_{qN} = K_2 x_{1N} - K_3 (x_{2N} + x_{3N}) + K_4 x_{4N}$$
$$z_{qN} = 2y_{qN} - y_{qL} \qquad \text{(A-4)}$$

Inputs: x_{1N}, x_{2N}, x_{3N}, x_{4N} from q-channel filter/integrator and guidance filter

Output: z_{qN} to pitch/yaw compensator and y_{qN} to storage

Storage: y_{qL}

Constants: K_2, K_3, K_4

5. r-Channel Filter/Integrator Update Time Interval: 0.014 second

Equations:

$$x_{5N} = x_{5L} + B_{11} \dot{\lambda}_{rL}$$
$$x_{6N} = x_{6L} + B_{21} (\dot{\lambda}_{rL} - x_{6L}) \qquad \text{(A-5)}$$

Input: $\dot{\lambda}_{rL}$ from seeker A/D

Outputs: x_{5N}, x_{6N} to storage and r-channel output

Storage: x_{5L}, x_{6L}

Constants: B_{11}, B_{21}

6. r-Channel Guidance Filter Update Time Interval: 0.014 second (Executed only with $C_r = 1$)

Equations:

$$x_{7N} = x_{7L} + A_{32}\,(x_{6L} - x_{7L}) + B_{31}\,(\dot{\lambda}_{rL} - x_{7L})$$

$$x_{8N} = x_{8L} + A_{42}\,(x_{6L} - x_{8L}) + A_{32}\,(x_{7L} - x_{8L}) + B_{41}\,(\dot{\lambda}_{rL} - x_{8L}) \qquad \text{(A-6)}$$

Inputs: $\dot{\lambda}_{rL}$ from seeker A/D and x_{6L} from r-channel filter/integrator

Outputs: x_{7N}, x_{8N} to storage and r-channel output ($C_r = 1$)

Storage: x_{7L}, x_{8L}

Constants: B_{31}, A_{42}, A_{32}, B_{41}

7. r-Channel Output ($C_r = 0$) Update Time Interval: 0.014 second

Equations:

$$y_{rN} = x_{5N} - K_1\,x_{6N}$$

$$z_{rN} = 2y_{rN} - y_{rL} \qquad \text{(A-7)}$$

Inputs: x_{5N}, x_{6N} from r-channel filter integrator, y_{rN} to storage

Output: z_{rN} to pitch/yaw compensators and y_{rN} to storage

Storage: y_{rL}

Constant: K_1

8. r-Channel Output ($C_1 = 1$) Update Time Interval: 0.014 second

Equations:

$$y_{rN} = K_2\,x_{5N} - K_3\,(x_{6N} + x_{7N}) + K_4\,x_{8N}$$

$$z_{rN} = 2y_{rN} - y_{rL} \qquad \text{(A-8)}$$

Inputs: x_{5N}, x_{6N}, y_{7N}, x_{8N} from r-channel filter/integrator and guidance filter

Outputs: z_{rN} to pitch/yaw compensator input and y_{rN} to storage

Storage. y_{rL}

Constants: K_2, K_3, K_4

9. Pitch/Yaw Compensator Input Update Time Interval: 0.004 second

Equations:

$$\mu_1 = - z_{qL} + \theta_L + z_{rL} - \Psi_L$$
$$\mu_2 = - z_{qL} + \theta_L - z_{rL} + \Psi_L \qquad \text{(A-9)}$$

Inputs: z_{qL} and z_{rL} from q/r channel outputs. θ_L and Ψ_L from pitch/yaw gyro A/D's.

Outputs: μ_1, μ_2 to pitch/yaw compensators

10. Pitch Gyro Compensator Update Time Interval: 0.004 second

Equations:

$$x_{9N} = \mu_1 + C_{11} (x_{9L} - \mu_1)$$
$$x_{10N} = C_{22} x_{10L} + C_{32} (\mu_1 - x_{11L})$$
$$x_{11N} = x_{11L} + C_{32} x_{10L} + D_{31} (\mu_1 - x_{11L}) \qquad \text{(A-10)}$$
$$y_{1N} = x_{11N} + K_5 (x_{9N} - x_{11N}) - K_6 x_{10N}$$
$$z_{1N} = K_7 (2y_{1N} - y_{1L})$$

Inputs: μ_i from pitch yaw compensator input

Output: z_{1N} to actuator D/A. x_{9N}, x_{10N}, x_{11N}, y_{1N} to storage

Storage: x_{9N}, x_{10L}, x_{11L}, y_{1L}

Constants: C_{11}, C_{22}, C_{32}, D_{31}, K_5, K_6, K_7

11. Yaw Gyro Compensator Update Time Interval: 0.004 second

Equations:

$$x_{12N} = \mu_2 + C_{11} (x_{12L} - \mu_2)$$

$$x_{13N} = C_{22} x_{13L} + C_{32} (\mu_2 - x_{14L})$$

$$x_{14N} = x_{14L} + C_{32} x_{13L} + D_{31} (\mu_2 - x_{14L}) \qquad \text{(A-11)}$$

$$y_{2N} = x_{14N} + K_5 (x_{12N} - x_{14N}) - K_6 x_{13N}$$

$$z_{2N} = K_7 (2 y_{2N} - y_{2L})$$

Inputs: μ_2 from pitch yaw compensator input

Outputs: z_{2N} to actuator D/A. x_{12N}, x_{13N}, x_{14N}, y_{1N} to storage

Storage: x_{12L}, x_{13L}, x_{14L}, y_{2L}

Constants: C_{11}, C_{22}, C_{32}, D_{31}, K_5, K_6, K_7

12. Roll Gyro Compensator Update Time Interval: 0.002 second

Equations:

$$x_{15N} = \phi_L + E_{11} (x_{15L} - \phi_L)$$

$$x_{16N} = E_{22} x_{16L} + E_{32} (\phi_L - x_{17L})$$

$$x_{17N} = x_{17L} + E_{32} x_{16L} + F_{31} (\phi_L - x_{17L}) \qquad \text{(A-12)}$$

$$y_{3N} = x_{17N} + K_8 (x_{15N} - x_{17N}) - K_9 x_{17N}$$

$$z_{3N} = K_{10}(2y_{3N} - y_{3L})$$

Input: ϕ_L from roll gyro A/D

Output: z_{3N} to actuator D/A. x_{15N}, x_{16N}, x_{17N}, y_{3N} to storage

Storage: x_{15L}, x_{16L}, x_{17L}, y_{3L}

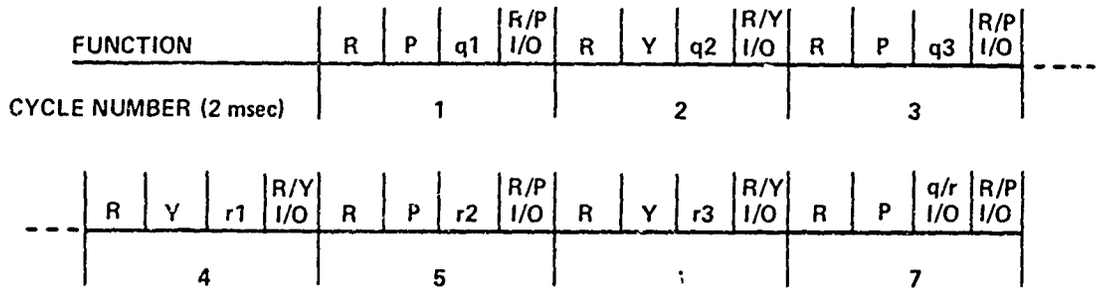
Constants: E_{11}, E_{22}, E_{32}, F_{31}, K_8, K_9, K_{10}

IV. System Initialization

The variables in the difference equations are all initialized to zero. At crossover, set $x_{4L} = x_{2L}$, $x_{8L} = x_{6L}$, $x_{2L} = x_{6L} = 0$.

V. Timing

A typical timing diagram is shown in Figure A-3. In this diagram, a basic timing unit is the 2 msec roll cycle. The roll compensator is shown being executed every cycle. Pitch and yaw compensator executions are alternated. The q/r channel software is assumed to be divided about equally among the seven cycles allocated to it. The next seven cycles have P and Y interchanged. It is of some importance that I/O operations be scheduled as shown in relation to program execution because of effects on the system phase margins and assumptions made in the difference equation derivation. Variances in the timing pattern can be handled but might require modifications in the output equations. Notice that there are really no q/r channel analog outputs.



R,P,Y – ROLL/PITCH/YAW COMPENSATOR EXECUTION
R/P/Y I/O – ROLL/PITCH/YAW INPUT–OUTPUT
q1,q2,q3,r1,r2,r3 – q/r CHANNEL EXECUTION
q/r I/O – q/r CHANNEL INPUT–OUTPUT

Figure A-3. Timing diagram.

VI. Word Length

The recommended processor word length is 16 bits. All computations can be handled satisfactorily with 16-bit single precision arithmetic. It has been established that all constants may be expressed in 3 octal digits not counting leading zeros. The critical computations for word length specification are the compensator output equations in which precision is lost in the process of multiplying terms by high gains with subsequent subtraction. If these computations were done in double-precision, a 12-bit processor would handle this job. However, the 16-bit device is recommended for the purpose of this feasibility demonstration.

VII. Memory Requirements

The total memory required for program storage, constant storage, and intermediate variable storage should not exceed 500 words.

VIII. Instruction Set

The instruction set may be very simple since the only operations are simple additions, subtractions, multiplies, and decisions. A divide is not required in the algorithms.

IX. Computational Summary

A computational summary is shown in Table A-1. Since the dominant load on the microprocessor load is expected to be the multiplications, it can be seen that one of these operations must be executed in about 100 microseconds or less.

TABLE A-1. COMPUTATIONAL SUMMARY

	Multiplies	Adds	Decisions	Analog Inputs	Analog Outputs	Rate (per sec)
q-channel	10	20	1	1	0	71.43
r-channel	10	19	1	1	0	71.43
Pitch Comp	8	16	1	1	1	250
Yaw Comp	8	16	1	1	1	250
Roll Comp	8	13	1	1	1	500
Total (per sec)	9429	17,286	1143	1143	1000	

X. List of Constants

A list of constants is included in Table A-2. These constants are applicable only for the specified data rates. Any change in data rates will require a recomputation of these constants.

XI. Indirect Fire Mode

The indirect fire mode has been somewhat ignored since the primary purpose of this task is to see if a microprocessor can stabilize and guide the missile. Its implementation would require a direct/indirect fire decision, a test for target detection, and provision for the turning rate bias, b.

TABLE A-2. LIST OF CONSTANTS

A_{32} = Decimal	Octal
A_{32} = 0.17022	0.12712
A_{42} = 0.01787	0.01112
B_{11} = 0.01400	0.00713
B_{21} = 0.18942	0.14077
B_{31} = 0.01919	0.01165
B_{41} = 0.00132	0.00053
C_{11} = 0.09672	0.06141
C_{22} = 0.00505	0.00245
C_{32} = 0.54627	0.42754
D_{31} = 0.44868	0.34556
E_{11} = 0.13538	0.10524
E_{22} = 0.43066	0.33438
E_{32} = 0.42476	0.33136
F_{31} = 0.14458	0.11202
g = 1.0000	1.0
K_1 = 0.06667	0.04211
K_2 = 4.0000	4.0
K_3 = 0.26667	0.21042
K_4 = 2.4000	2.3146
K_5 = 141.67	215.53
K_6 = 22.399	26.314
K_7 = 2.5000	2.4
K_8 = 92.908	134.72
K_9 = 40.323	50.645
K_{10} = 0.60000	0.46315

XII. Glossary

$x_{1N} - x_{17N}$	Updated state variable
$x_{1L} - x_{17L}$	Stored state variable
$B_{11}, B_{21}, B_{31}, B_{41}, A_{32}, A_{42}$	q/r channel transition/control matrix
$\dot{\lambda}_{qL}\ \dot{\lambda}_{rL}$	Inputs from seeker
g	Gravity bias constant
y_{qN}, y_{rN}	Updated q/r channel variables
y_{qL}, y_{rL}	Stored q/r channel variables
z_{qN}, z_{rN}	q/r channel outputs
K_1, K_2, K_3, K_4	q/r channel gain constants
μ_1, μ_2	Pitch/yaw compensator inputs
θ_L, ψ_L, ϕ_L	Pitch/yaw/roll gyro measurements
$C_N, C_{22}, C_{32}, D_{31}$	Pitch/yaw compensator transition/control matrix elements
K_5, K_6, K_7	Pitch/yaw channel gain constants
y_{1N}, y_{2N}	Updated pitch/yaw variables
y_{1L}, y_{2L}	Stored pitch/yaw variables
z_{1N}, z_{2N}	Pitch/yaw compensator outputs
$E_{11}, E_{22}, E_{32}, F_{31}$	Roll compensator transition/control matrix elements
y_{3N}	Updated roll compensator variable
y_{3L}	Stored roll compensator variable
z_{3N}	Roll compensator output
C_q, C_r	Seeker q/r signal crossover discretes
D	Indirect fire target detection discrete

Appendix B. LOOP MODELLING AND ANALYSIS

The factors of interest in the modelling and analysis work are the crossover frequencies for the seeker loops and gyro loops, the analog system frequency response roll-off beyond crossover, and the frequency characteristics of the internal seeker electronics. A single channel seeker loop block diagram is shown in Figure B-1 and a similar diagram is shown in Figure B-2 for a gyro loop. Models used for the various blocks are explained in the following paragraphs.

1. Guidance Filter

The guidance filter model is

$$F_1(s) = \frac{s\left(\frac{s}{1.5} + 1\right)}{\left(\frac{s}{15} + 1\right)^2} \quad . \tag{B-1}$$

2. Seeker Filter Integrator

The filter integrator model is

$$F_2(s) = \frac{1}{s\left(\frac{s}{15} + 1\right)} \quad . \tag{B-2}$$

3. Pitch Yaw Gyro Compensator

The pitch yaw compensator model is

$$F_3(s) = \frac{\left(\frac{s}{30} + 1\right)^2}{\left(\frac{s}{1000} + 1\right)\left(\frac{s^2}{300^2} + \frac{s}{300} + 1\right)} \quad . \tag{B-3}$$

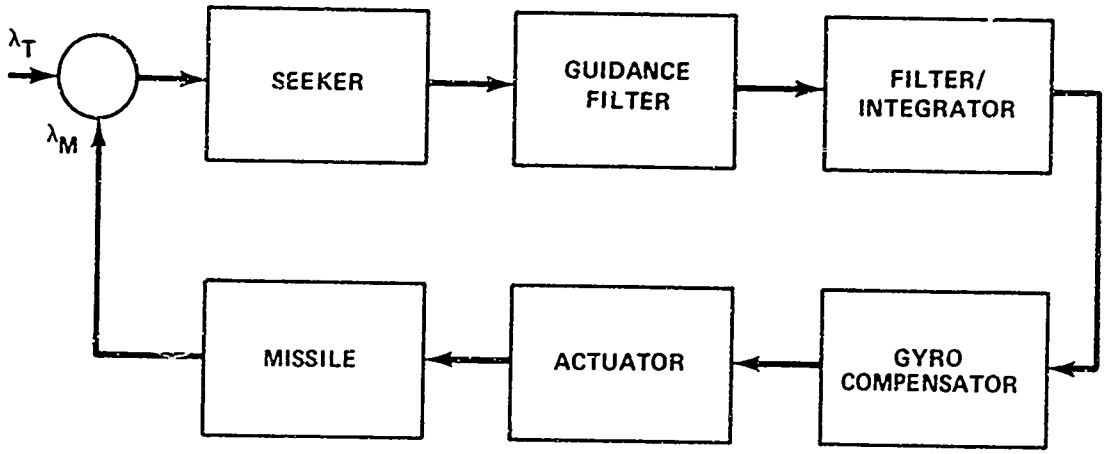


Figure B-1. Seeker loop.

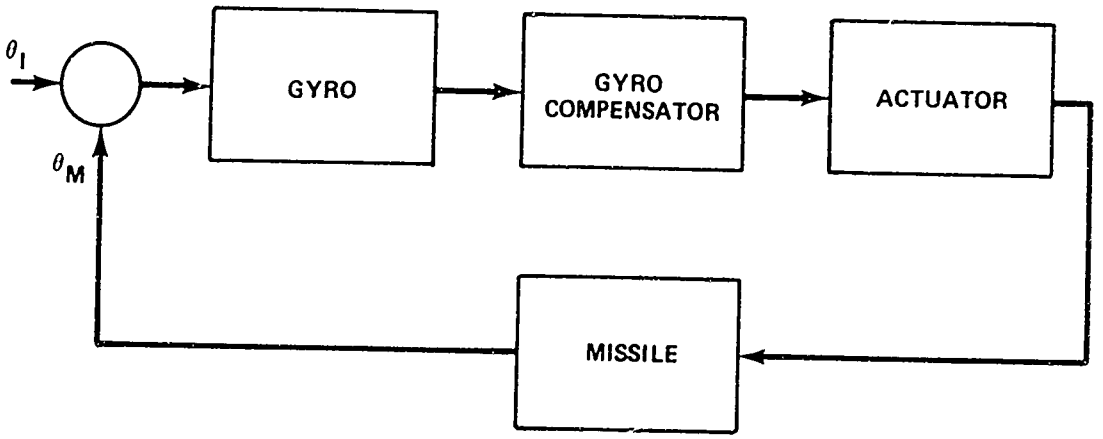


Figure B-2. Gyro loop.

4. Roll Gyro Compensator

The roll gyro compensator model is

$$F_4(s) = \frac{\left(\frac{s}{35} + 1\right)^2}{\left(\frac{s}{1000} + 1\right)\left(\frac{s^2}{300^2} + \frac{s}{300} + 1\right)} \quad . \qquad \text{(B-4)}$$

5. Actuator Model

The actuator model was taken from North American Rockwell document 772-SA-20-72, "Scaled Down THFTV Autopilot Requirements." The transfer function is

$$F_A(s) = \frac{1}{\left(\frac{s}{\omega_A} + 1\right)\left(\frac{s^2}{200^2} + \frac{s}{200} + 1\right)} \quad . \qquad \text{(B-5)}$$

The frequency ω_A was given as a function of dynamic pressure which was converted to Mach number (Table B-1).

TABLE B-1. ACTUAL MODEL DATA

Dynamic Pressure (lb/ft^2)	ω_A (rad/sec)	Mach No.
237	30	0.4
1200	26	0.9
1795	24	1.1
2900	20	1.4
4230	16	1.7

The ω_A versus Mach number tabulation was later used to get approximate values of ω_A for Mach numbers at which aerodynamic data were available.

6. Missile Pitch/Yaw Model

The missile response was modelled by the function

$$\frac{K_p}{s\left(\frac{s}{\omega_p} + 1\right)} \tag{B-6}$$

values for K_p and ω_p were calculated from

$$K_p = \frac{2V\ C_{\ell\delta}}{d\ C_{\ell p}} \qquad \omega_p = -\frac{qsd^2 C_{\ell p}}{2I_p\ V} \tag{B-7}$$

for which information on aerodynamic coefficients, effective area, effective length, and moment of inertia was taken from MICOM Technical Report RG-73-18 "T6 Missile Description." Resulting values are shown in Table B-2.

TABLE B-2. MISSILE PITCH YAW DATA

Mach No.	ω_A	K_p	ω_p
0.5	29	253	0.31
0.9	26	467	0.57
1.3	21	619	0.81

7. Missile Roll Model

The missile roll response was modelled by the function

$$F_R = \frac{K_R}{s\left(\frac{s}{\omega_R} + 1\right)} \quad . \tag{B-8}$$

Values calculated for K_R and ω_R are shown in Table B-3.

TABLE B-3. MISSILE ROLL DATA

Mach No.	ω_A	K_R	ω_R
0.5	29	1286	0.62
0.9	26	2527	1.15
1.3	21	4183	1.62

8. Gyro Model

The gyro transfer function was taken to be unity since no better information was available (See North American document 772-SA-20-72).

9. Seeker Model

The sampled data seeker model was based on the equation

$$y(s) = \frac{G(s)\ U^*(s)}{1 + GH^*(s)} \tag{B-9}$$

in which

$$u^*(s) = \frac{1}{T} \sum_{-\infty}^{\infty} u\ (s + jm\omega_s) \quad . \tag{B-10}$$

From document 772-SA-20-72

$$G(s) = \left(\frac{1 - e^{-sT}}{s}\right)\left(0.5 + \frac{5}{s}\right) \tag{B-11}$$

$$GH^*(s) = \left\{\frac{1}{s}\ G(s)\right\}^* \quad . \tag{B-12}$$

From available tables

$$GH^*(s) = (1 - z^{-1})\left\{\frac{0.5T\ z}{(z - 1)^2} + \frac{\frac{5}{2}\ T^2\ z(z + 1)}{(z - 1)^3}\right\} \tag{B-13}$$

in which $z = e^{sT}$. The fundamental response was computed from

$$F_s(j\omega) = \frac{y}{u}(j\omega) = \frac{1}{T}\frac{G(j\omega)}{1 + GH^*(j\omega)} \quad , \tag{B-14}$$

an approximate analog model ignoring the sampler is

$$\tilde{F}_s(s) = \frac{s\left(\frac{s}{10} + 1\right)}{\frac{s^2}{5} + \frac{s}{10} + 1} \quad . \tag{B-15}$$

Open loop break-point frequency response plots are shown in Figures B-3 and B-4 using data at Mach 0.9 and Equation (B-15) for the seeker. Actual response data obtained by programming the model equations including Equation (B-14) for the seeker yielded crossover frequency information in Table B-4.

TABLE B-4. CROSSOVER FREQUENCY DATA

Mach No. / Mach No.		ω_C (rad/sec)	Phase Margin (deg)	Gain Margin (dB)
	0.5	15	19	23
P/Y	0.9	29	26	13
	1.3	41	24	9
	0.5	23	18	17
Roll	0.9	51	21	7
	1.3	88	10	1
	0.5	13	13	7
Seeker	0.9	18	-8	-4
	1.3	21	-20	-12

According to these data, none of the system phase margins are very reassuring. The seeker loop appears to be unstable at the upper Mach numbers. A contributing factor to this condition is the zero order delay in the internal seeker electronics which contributes 20 to 30 degrees of lag at the indicated crossover frequencies. Some of this delay could be compensated for by the microprocessor software. A time varying gain program in the microprocessor would also help to improve system performance and stability.

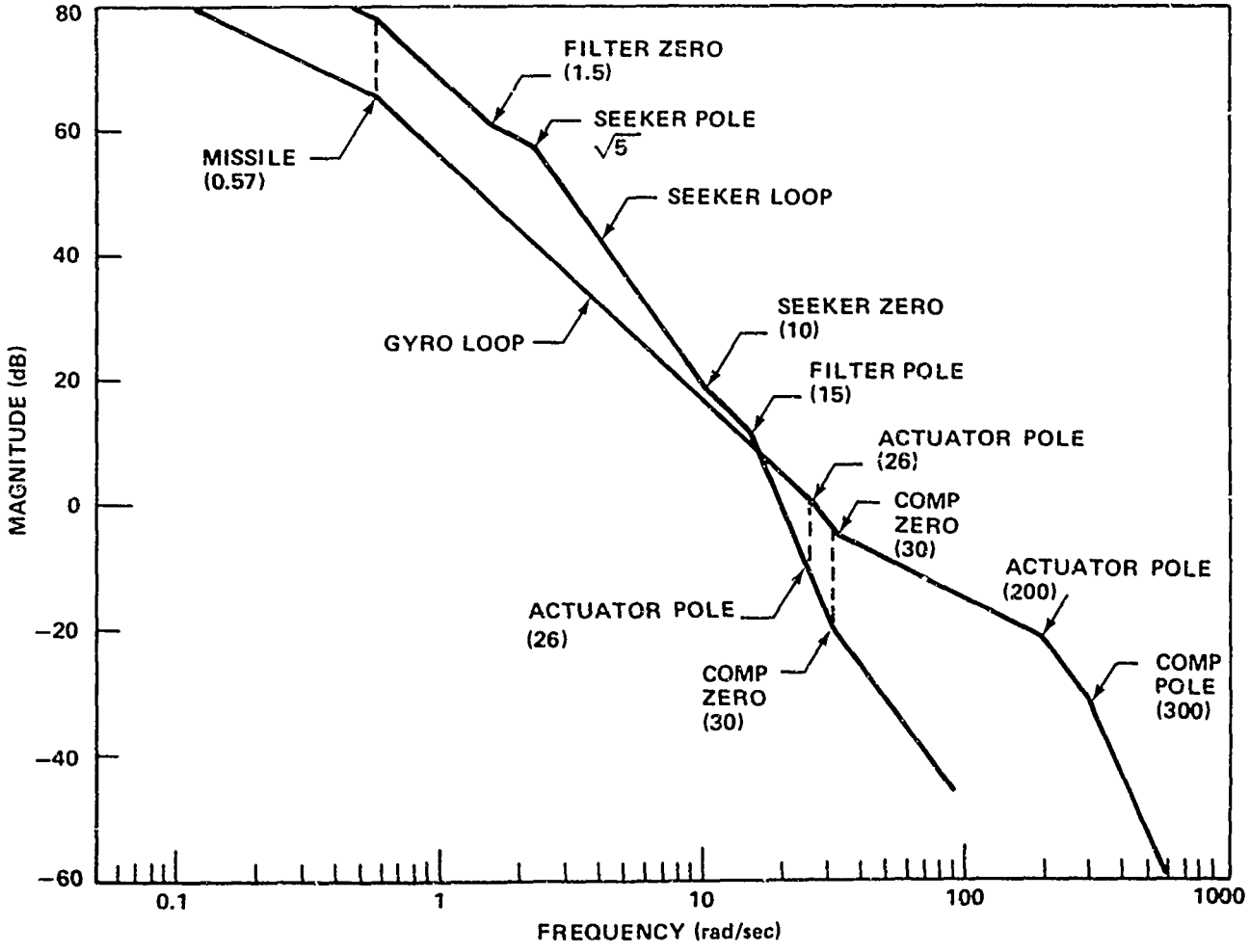


Figure B-3. Pitch loop responses, Mach 0.9, $K_p = 467$, $\omega_p = 0.57$, $\omega_A = 26$.

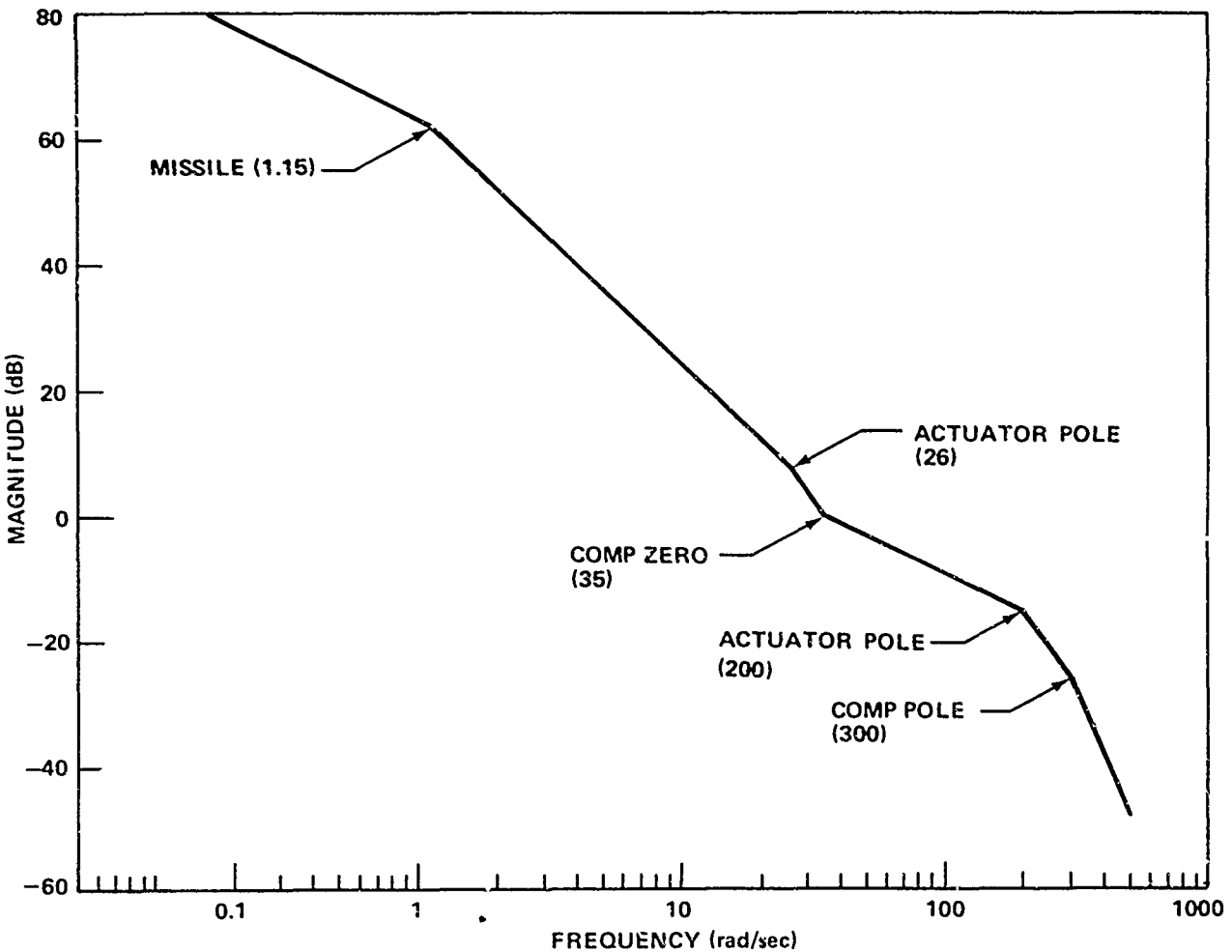


Figure B-4. Roll loop response, Mach 0.9, K_R = 2527, ω_R = 1.15, and ω_A = 26.

It can be seen from gyro compensator data that the digital system response is fairly effective up to the complex actuator pole break at 200 rad/sec. Beyond this, loop response roll-off is so rapid that it does not matter what the digital system does.

The potential seeker sampler sideband problem is caused by the fact that inputs at the digital system's data rate or multiples thereof will cause a bias build-up on the digital integrator. The internal seeker response, calculated using Equations (B-9) and (B-10), is shown in Figure B-5. It is believed that the best that can be done to choose the integrator update interval somewhere between the sets of sidebands from the seeker and hope that nothing exciting happens. A convenient interval from a computer software viewpoint is to use some multiple of the shortest update interval which is that for roll. The interval chosen is 7 roll cycles or 14 milliseconds which corresponds to about 71.4 Hz. This is about midway between the sidebands at 60 to 80 Hz. The seeker gain at this frequency is -27 dB or 0.045 which means that bias build-up at the rate of 0.045 degree per second of flight time per degree of seeker input amplitude at 71.4 Hz could occur. With a loop gain of about -100 dB, however, disturbances at 71.4 Hz should not be a problem. There is not much computation time to be gained by splitting the seeker filter/integrator software into more roll cycles than seven.

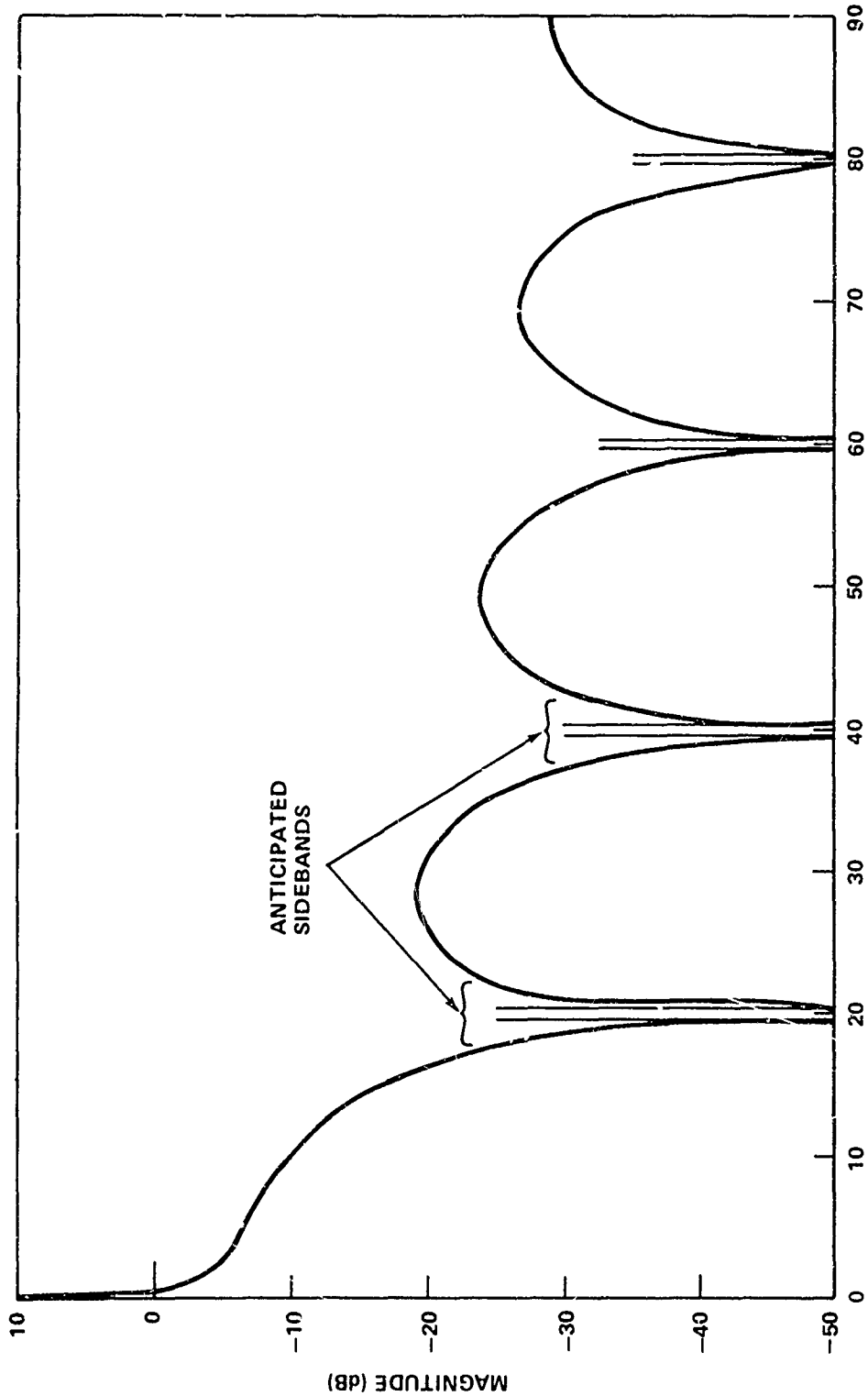


Figure B-5. Internal seeker response.