JOINT INSTITUTE FOR AERONAUTICS AND ACOUSTICS
STANFORD UNIVERSITY

Final Report

MULTI-CALCULATION RATE SIMULATIONS

Submitted to

NASA AMES RESEARCH CENTER
Simulation Sciences Division
Moffett Field, Ca 95035

By

JOINT INSTITUTE FOR AERONAUTICS AND ACOUSTICS

Guidance and Control Laboratory
Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305

J. David Powell, Principal Investigator
Mahboob Akhter, Research Assistant

NSG-2250

November 1977

Final Report

MULTI-CALCULATION RATE SIMULATIONS

Submitted to
NASA Ames Research Center
Simulation Sciences Division
Moffett Field, Ca. 95035

By
Guidance and Control Laboratory
Joint Institute for Aeronautics and Acoustics
Department of Aeronautics and Astronautics

J. David Powell, Principal Investigator
Mahboob Akhter, Research Assistant

NSG-2250

November 1977

TABLE OF CONTENTS

I. INTRODUCTION

Real time simulations of aerospace systems have been developed at several NASA centers for the purpose of providing a testbed for numerous studies involving use of cockpit mockups, visual displays, pilot/astronauts, and vehicle motion. It is common in real-time simulations of large systems to separate the high and low frequency subsystems within the simulation and perform the integrations of the subsystems at different calculation rates. This is done to strike a balance between accuracy of calculation and capacity of the digital computer. Questions arise as to the accuracy of this structure compared to single calculation rates and if any interactions arise that cause errors that are worse than those expected from an analysis of the subsystem above.

This report describes a study that was done on a linear aircraft model that investigates the questions above. Since actual simulations are much more complex with many nonlinearities, these results cannot be applied directly; however, the study does show where the problems are and gives guidelines for selection of sample rates for multiple rate simulations.

II. BACKGROUND

The particular system simulations in question typically have a fast mode (aircraft short period, dutch roll) and a slow mode (phugoid, roll divergence) which can differ by an order of magnitude in the respective natural frequencies. These kind of problems are referred to as "stiff" differential equations in numerical analysis circles although special techniques to solve stiff equations are not absolutely necessary until frequency multiples on the order of 100 or more exist.

Many different techniques have been reported [1,2,3,4,5] but none use different calculation rates for the different modes of the system. The reason for the lack of literature (and lack of interest by numerical analysts) on multi-calculation rate techniques appears to be due to the difficulty in separating systems into fast and slow modes. Numerical integration procedures are never limited to linear systems which are really the only ones that can be cleanly separated.

In aircraft simulations, system descriptions are sufficiently close to linear so as to allow separation based on our knowledge of the approximate linear version. Furthermore, since the differential equations arise from known physical phenomenon which are similar for all vehicles and flight conditions, once the separation has been determined for one case, it is applied successfully for most others.

On the other hand, the general problem of analyzing a linear discrete system with multiple sample rates has been studied extensively [6,7,8,9,10]. Since any numerical integration procedure can be reduced to a set of difference equations, and will be linear if the differential equations being integrated are linear, these methods are applicable. Unfortunately, the methods are very tedious to apply and require large amounts of algebra before going to a computer. Application of these methods to the aircraft simulation were studied for simple integration procedures but judged to be beyond the scope of the study for the more realistic and complex integration procedures.

III. METHOD OF ANALYSIS

To provide a common yardstick for comparing the various algorithms, it was decided to use the frequency response of the aircraft simulations. In particular, the transfer function of the longitudinal mode of a DC-8, from elevator command to vehicle attitude was selected for study. Two methods were employed;

1) discrete analysis using z-transforms of the single calculation rate cases, and
2) numerical simulation of the multi-calculation rate cases.

A. The Selected Example

A DC-8 in approach configuration was selected for study. The transfer function between elevator and attitude for this case is [11]:

$$\frac{\theta(s)}{\delta_e(s)} = -1.338 \frac{(s+0.0605)(s+0.535)}{(s^2+1.69s+2.67)(s^2+0.0198s+0.0267)} \quad (1)$$

which results in the short period and phugoid characteristics as shown in Table I. Note the 10:1 difference between the frequencies of the fast and slow modes.

Table I: EXAMPLE CHARACTERISTICS

	Natural Frequency	Damping
Short Period	1.62 r/sec(.258Hz)	0.522
Phugoid	0.164 r/sec(.0261Hz)	0.0606

The magnitude and phase of (1) was determined analytically and has been included in all the following graphs for comparison (labeled "continuous system"). Since this represents the response of an actual aircraft with varying frequency of input commands, the goal of all digital approximations of this aircraft is to match the continuous response as closely as possible.

B. Discrete Analysis of the Single Rate Case

The transfer function in (1) can be written as a set of differential equations:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + K \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} (\ddot{\delta}_e + b_1\dot{\delta}_e + b_0\delta_e) \tag{2}$$

where for this example:

$$\begin{aligned} a_3 &= 1.7.00522 & \qquad y_1 &= \theta \\ a_2 &= 2.6813872 & y_2 &= \dot{\theta} \\ a_1 &= 0.09712526 & y_3 &= \ddot{\theta} \\ a_0 &= 0.07006953 & y_4 &= \dddot{\theta} \\ K &= -1.338 \\ b_1 &= 0.5955 \\ b_0 &= 0.0323675 \end{aligned}$$

1. Euler's Integration:

Euler's integration [12] can be simply stated by:

$$y(n+1) = y(n) + T[\dot{y}(n)] \tag{3}$$

Combining (2) and (3) and using first differences to generate $\dot{\delta}_e$ and $\ddot{\delta}_e$ yields:

$$\frac{\theta(z)}{\delta_e(z)} = KT^2 \frac{z^2 + n_1 z + n_0}{z^4 + m_3 z^3 + m_2 z^2 + m_1 z + m_0} \tag{4}$$

where

$$\begin{aligned} T &= \text{sample time} \\ K &= -1.338 \\ n_0 &= b_0T^2 - b_1T + 1 \\ n_1 &= b_1T - 2 \\ m_0 &= a_3T^4 - a_2T^3 + a_1T^2 - a_0T + 1 \\ m_1 &= a_2T^3 - 2a_1T^2 + 3a_0T - 4 \\ m_2 &= a_1T^2 - 3a_0T + 6 \\ m_3 &= a_0T - 4 \end{aligned}$$

The frequency response of this discrete transfer function can be determined by evaluating (4) with z taking on values around the unit circle. The computer code for doing this is contained in Appendix A and the results are contained in the following section for T's ranging from 0.05 sec to 0.5 sec.

2. First Order Adams Integration [13]:

The algorithm is:

$$y(n+1) = y(n) + \frac{T}{2}[3f(n) - f(n-1)] \tag{5}$$

where $f(n) = \dot{y}(n)$ from (2).

This algorithm makes use of one past value of the derivative function and therefore increases the order of the discrete system.

The discrete transfer function of (5) is:

$$\frac{\theta(z)}{\delta_e(z)} = K\frac{B_8z^8 + B_7z^7 + B_6z^6 + B_5z^5 + B_4z^4 + B_3z^3 + B_2z^2 + B_1z + B_0}{C_{10}z^{10} + C_9z^9 + C_8z^8 + C_7z^7 + C_6z^6 + C_5z^5 + C_4z^4 + C_3z^3 + C_2z^2 + C_1z + C_0} \tag{6}$$

where the coefficients are defined in Appendix B along with the computer code to evaluate the frequency response for (6).

3. Second Order Adams Integration [13]:

The algorithm is:

$$y(n+1) = y(n) + \frac{T}{12}[23f(n) - 16f(n-1) + 5f(n-2)] \tag{7}$$

which yields:

$$\frac{\theta(z)}{\delta_e(z)} = K\frac{B_{14}z^{14} + B_{13}z^{13} + \ldots + B_1z + B_0}{C_{16}z^{16} + C_{15}z^{15} + \ldots + C_1z + C_0} \tag{8}$$

where the coefficients are defined in Appendix C.

C. Simulation of the Multi-Rate Case

An alternate way of expressing (1) and (2) is also given by Teper [11]:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -.0291 & .0629 & 0 & -32.2 \\ -.251 & -.628 & 243 & 0 \\ -7.7\backslash^{-6} & -8.7\backslash^{-3} & -.792 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ -10.2 \\ -1.35 \\ 0 \end{bmatrix} \delta_e \tag{9}$$

The separation of these equations into fast and slow modes can be done in several ways. The ideal manner would be to transform the equations into their normal modes which would then produce two coupled 2nd order systems, one with pure short period characteristics and one with pure phugoid characteristics. Each could be integrated at sample rates suitable to that mode. In practice, this is difficult due to the non-linear terms in the equations. Furthermore, transforming to and from another state definition takes cpu time and may result in higher cpu loading than the more straightforward methods described next.

1. The 1×3 Separation

If a normal mode analysis of an aircraft is performed, we find that the short period consists primarily of α, q, and θ motion with insignificant effect on u. The phugoid consists of u, q, and θ with little effect on α. Therefore, since u is the only state that does not involve "fast" behavior, it is the only state that can be safely calculated at the slow calculation rate. The "1×3" separation recognizes this fact and partitions accordingly. The equations are:

Fast Loop -

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.628 & 243.5 & 0 \\ -0.0087 & -.792 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -10.2 \\ -.135 \\ 0 \end{bmatrix} \delta_e + \begin{bmatrix} -.251 \\ -.0000077 \\ 0 \end{bmatrix} u \tag{10}$$

Slow Loop -

$$\dot{u} = -0.0291U + [0.0629 \quad -32.2]\begin{bmatrix} w \\ \theta \end{bmatrix} \tag{11}$$

2. The 2×2 Separation

Another natural separation is based on fast calculation of orientation, q and θ, and slow calculation of translation, u and w. It is attractive because a larger portion of the calculations are done at a slower rate, hence more cpu time savings appear achievable. The equations are as follows.

Fast Loop -

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -.792 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} q \\ \theta \end{bmatrix} + \begin{bmatrix} -1.35 \\ 0 \end{bmatrix}\delta_e + \begin{bmatrix} -.77 \times 10^{-5} & -.0087 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} u \\ w \end{bmatrix} \tag{12}$$

Slow Loop -

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -.0291 & .0629 \\ -.251 & -.628 \end{bmatrix}\begin{bmatrix} u \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ -10.2 \end{bmatrix}\delta_e + \begin{bmatrix} 0 & -32.2 \\ 243.5 & 0 \end{bmatrix}\begin{bmatrix} q \\ \theta \end{bmatrix} \tag{13}$$

3. Simulation Procedures

The frequency response of each separation was determined using Euler's Integration (3) and the 1st order Adams Integration (5). It was evaluated at calculation rate ratios (IR) varying between 1 and 20. Since IR = 1 is the single rate case, these calculations could be checked by comparing with the analytical evaluations described in III-B.

The frequency response was determined by evaluating equations (10) and (11) or (12) and (13) using the integration formulas with δ_e equal to a sine wave of magnitude = 1. After an initial transient settling delay the resulting sine wave magnitude and phase was assumed to be the desired frequency response. The short period portion of the transient response was quite short; however, the phugoid transient response was unduly long to wait for settling. Therefore, the procedure calculated the phugoid transient response based on the continuous system

(1) using inverse Laplace transforms and subtracted this from the numerical evaluations before determining amplitude and phase. Appendix D contains a listing of the computer code.

IV. RESULTS

Table II contains a summary of the figures which represent the results of this study. In the simulations, all IR's between 1 and 20 were evaluated. Those cases not shown in the figures were found to be unstable.

TABLE II: SUMMARY OF FIGURES

Figure	Analytical	Simulated	Integration Algorithm	T_{fast}	IR	Separation
1	X		E	.05-.5	1	---
2	X		A1	.05-.5	1	---
3	X		A2	.05-.5	1	---
4	X		E,A1,A2	.05	1	---
5	X		E,A1,A2	.1	1	---
6	X		E,A1,A2	.2	1	---
7		X	E	.05	1-20	1 × 3
8		X	E	.1	1-10	1 × 3
9		X	E	.2	1-5	1 × 3
10		X	E	.05	1-10	2 × 2
11		X	E	.1	1-5	2 × 2
12		X	E	.2	1-3	2 × 2
13		X	A1	.05	1-20	1 × 3
14		X	A1	.1	1-10	1 × 3
15		X	A1	.1	1-5	1 × 3
16		X	A1	.05	1-10	2 × 2
17		X	A1	.1	1-5	2 × 2
18		X	A1	.2	1-3	2 × 2

The most significant result is the difference between the two separations. This can be seen by comparing the deviations from the continuous curves in Figs. 7, 8 and 9 with those in 10, 11 and 12 respectively for Euler's Integration and similarly Figs. 13, 14 and 15 with 16, 17 and 18. For both integration methods, the 1 × 3 separation is decidedly superior. This is no doubt due to the fact that the 2 × 2 separation solves the w(or α) equation at the <u>slow</u> rate while this state is important to the short period dynamics.

The first order Adams integration appears to be the best choice of integration methods. Its advantage over Euler's method and small disadvantage compared to a second order Adams is best illustrated in the Fig. 6b phase plot; however, examination of the magnitude in Fig. 6a shows the Euler method's error arising at a lower frequency but remaining smaller at the higher frequencies. The same kind of behavior is exhibited at the faster sample rates (Figs. 4 and 5) but is more difficult to see due to the greater accuracy.

The sample rate requirement for aircraft simulation with a first order Adams integration is dependent on the desired input frequency to be adequately simulated. Examination of Figs. 13, 14 and 15 indicate that one should select the fast sample rate at approximately 10 times the input frequency to be followed and that a slow rate at one-tenth this rate yields no degradation. In other words, to follow a 2 Hz input, one should solve the short period equations at 20 Hz and the phugoid at 2 Hz.

V. CONCLUSIONS

For a linear model of longitudinal aircraft motion, separation of the equations of motion into slow and fast calculation rate groups is best accomplished by performing u integration at the slow rate and w, q, θ at the fast rate. A separation with u and w as the slow variables and q and θ as the fast gave substantially less accuracy.

A first order Adams integration procedure appeared to be a good choice for real time aircraft simulations.

For the example used ($\omega_{\text{short period}} \cong 0.25$ Hz, $\omega_{\text{phugoid}} = 0.025$ Hz), the fast sample rate should be selected at approximately 10 times the maximum input frequency for which accurate aircraft simulated response is desired. A slow rate of one-tenth the fast rate yielded no degradation over the single rate case.

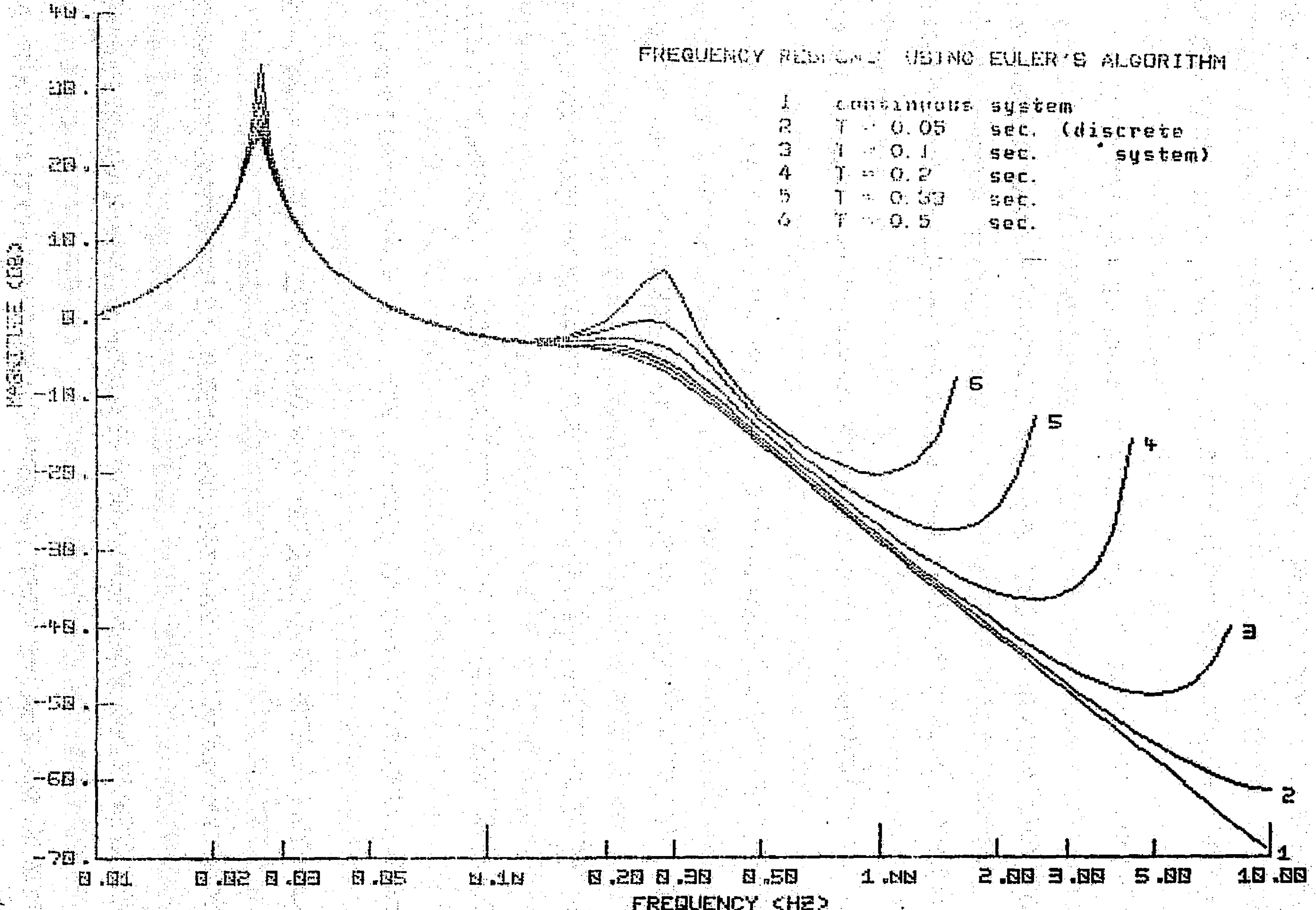


FIGURE 1a

ORIGINAL PAGE IS
OF POOR QUALITY

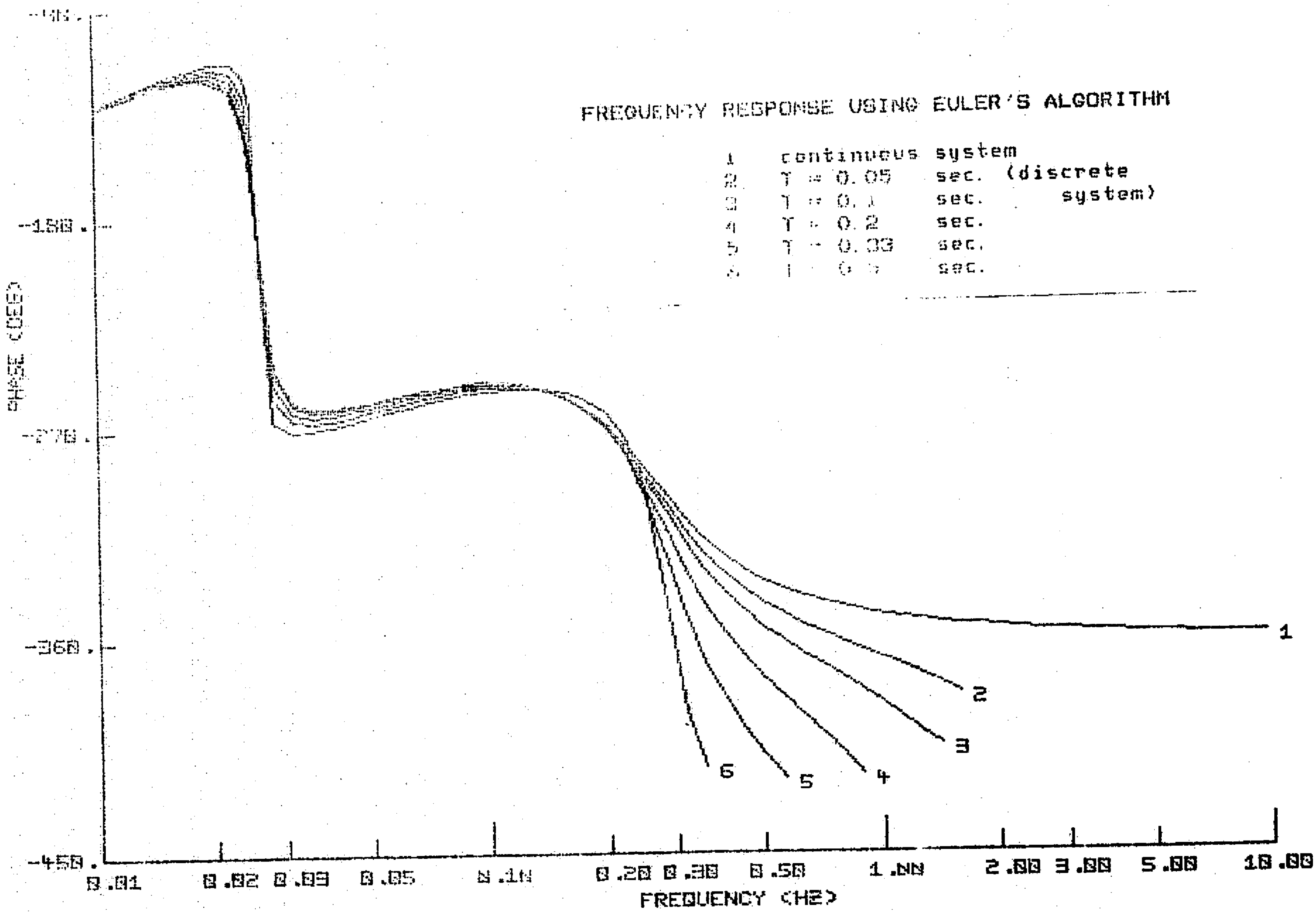

FREQUENCY RESPONSE USING EULER'S ALGORITHM
1 continuous system
2 T = 0.05 sec. (discrete system)
3 T = 0.1 sec.
4 T = 0.2 sec.
5 T = 0.33 sec.
6 sec.
PHASE (DEG)
-180.
-270.
-360.
-450.
0.01 0.02 0.03 0.05 0.10 0.20 0.30 0.50 1.00 2.00 3.00 5.00 10.00
FREQUENCY (HZ)
1
2
3
4
5
6

FIGURE 1b

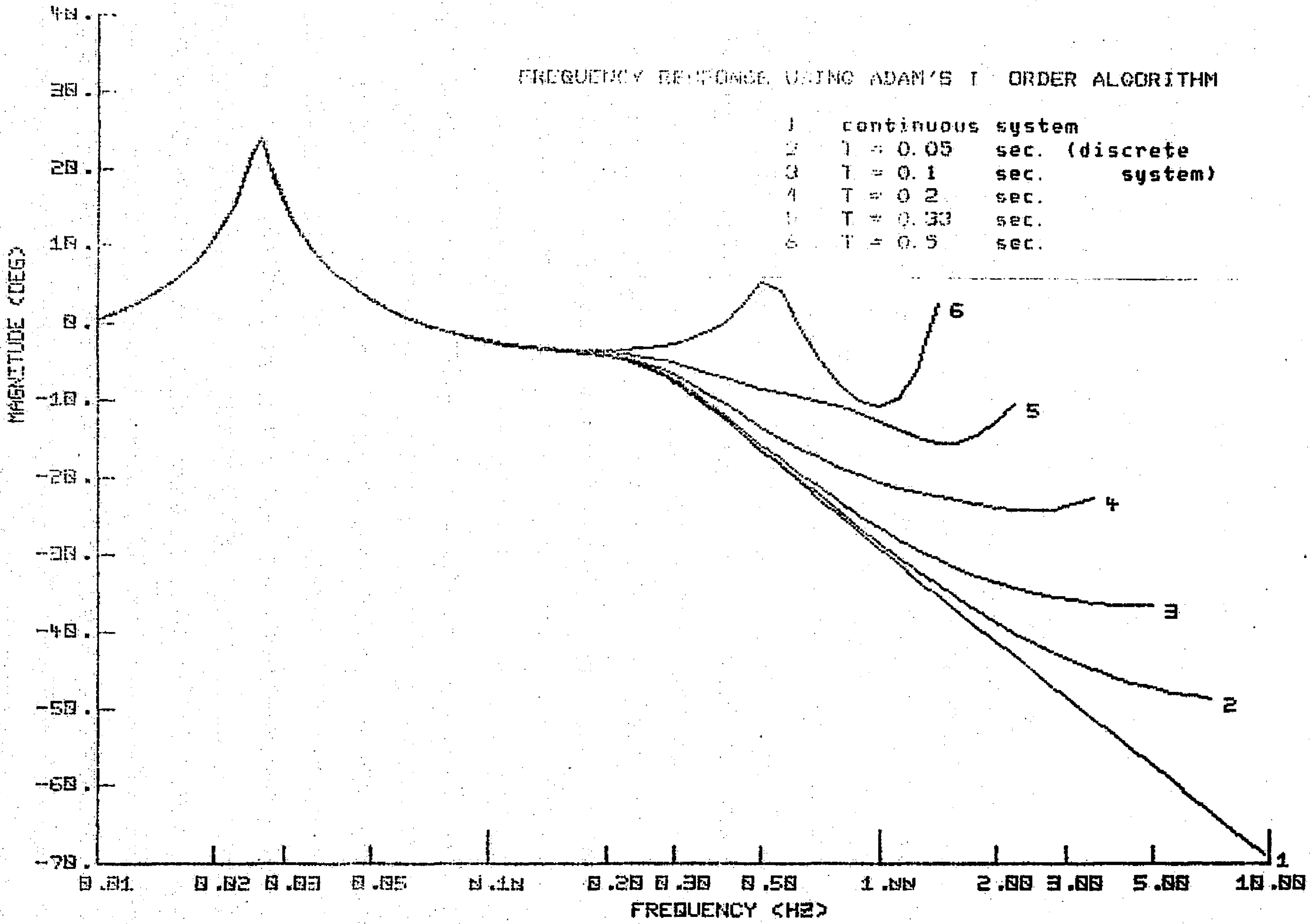

FREQUENCY RESPONSE USING ADAM'S I ORDER ALGORITHM
1 continuous system
2 T = 0.05 sec. (discrete
3 T = 0.1 sec. system)
4 T = 0.2 sec.
5 T = 0.33 sec.
6 T = 0.5 sec.
MAGNITUDE (DEG)
FREQUENCY (HZ)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
1
2
3
4
5
6

FIGURE 2a

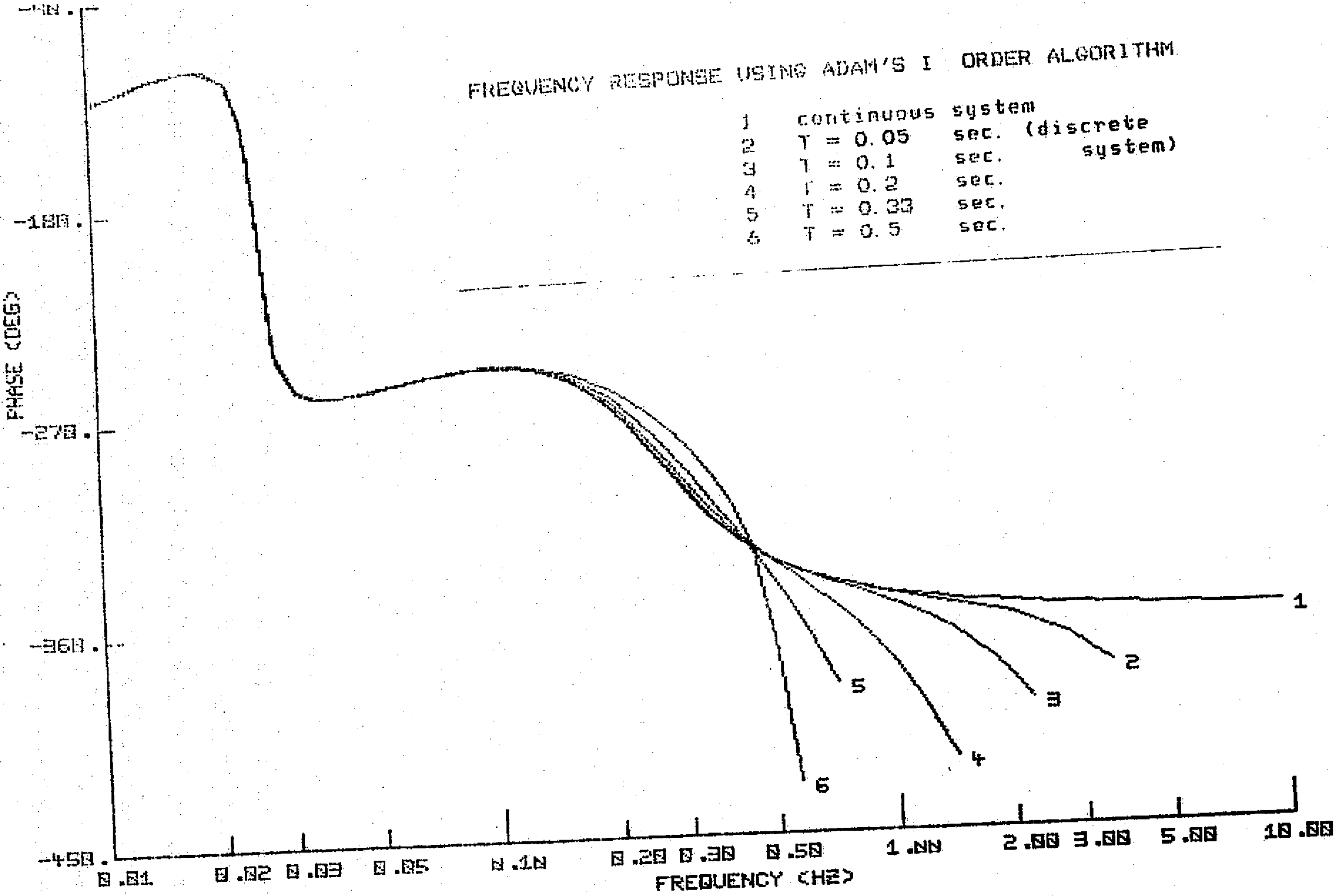

FREQUENCY RESPONSE USING ADAM'S I ORDER ALGORITHM
1 continuous system
2 T = 0.05 sec. (discrete system)
3 T = 0.1 sec.
4 T = 0.2 sec.
5 T = 0.33 sec.
6 T = 0.5 sec.
PHASE (DEG)
-180.
-270.
-360.
-450.
0.01
0.02 0.03
0.05
0.10
0.20 0.30
0.50
1.00
2.00 3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4
5
6

FIGURE 2b

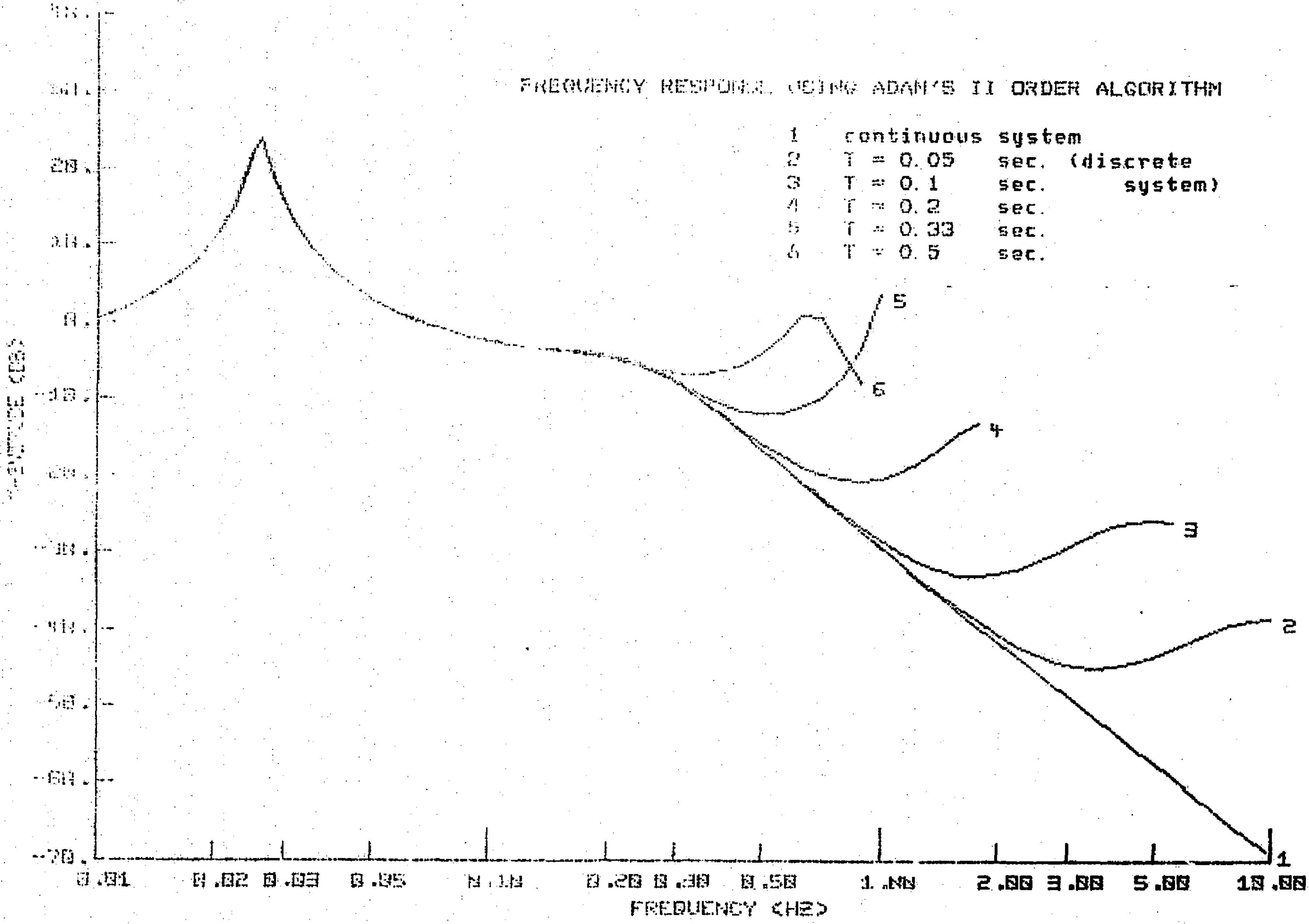

FREQUENCY RESPONSE USING ADAM'S II ORDER ALGORITHM
1 continuous system
2 T = 0.05 sec. (discrete system)
3 T = 0.1 sec.
4 T = 0.2 sec.
5 T = 0.33 sec.
6 T = 0.5 sec.
20.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.02 0.03
0.05
0.20 0.30
0.50
2.00 3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 3a

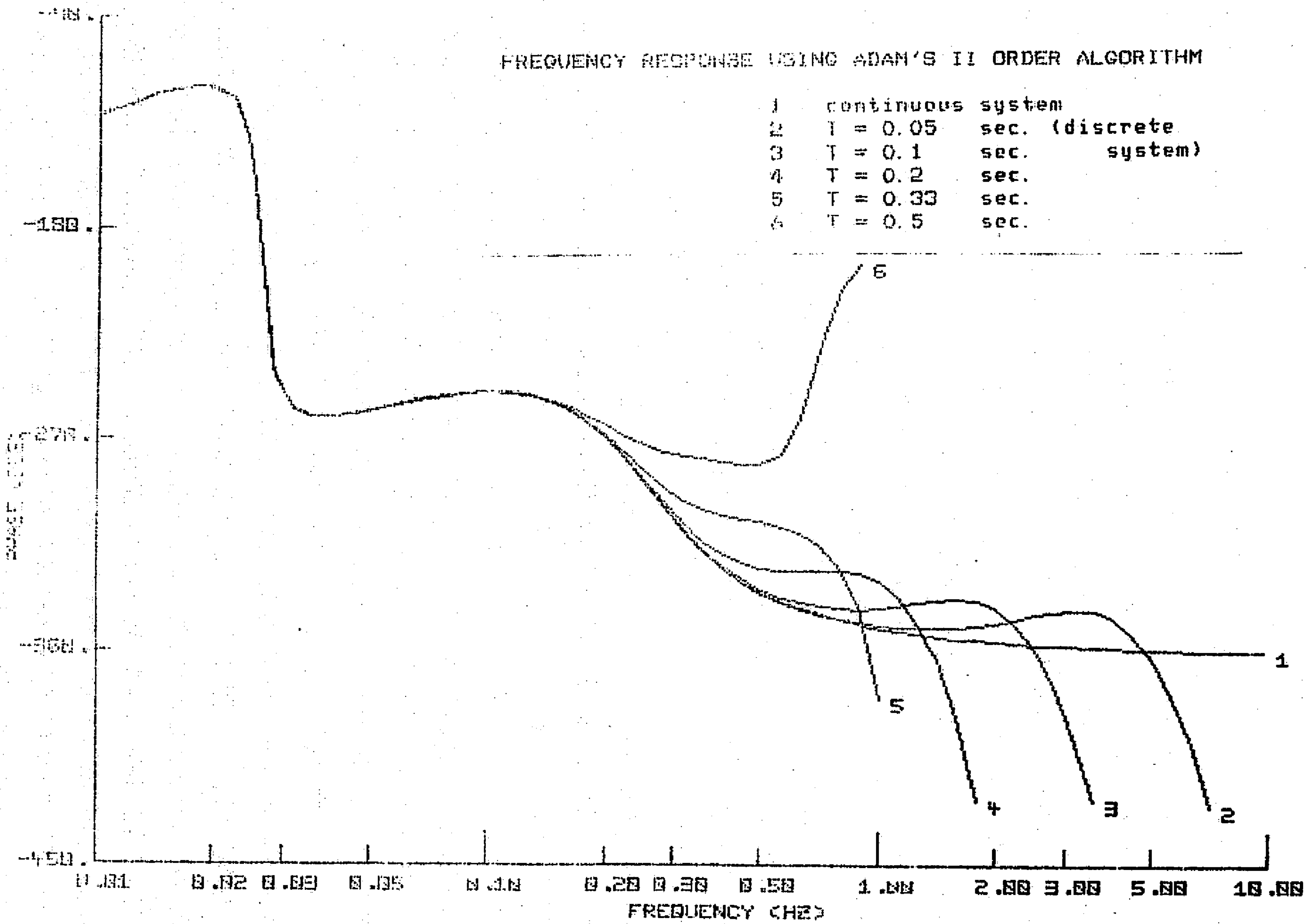
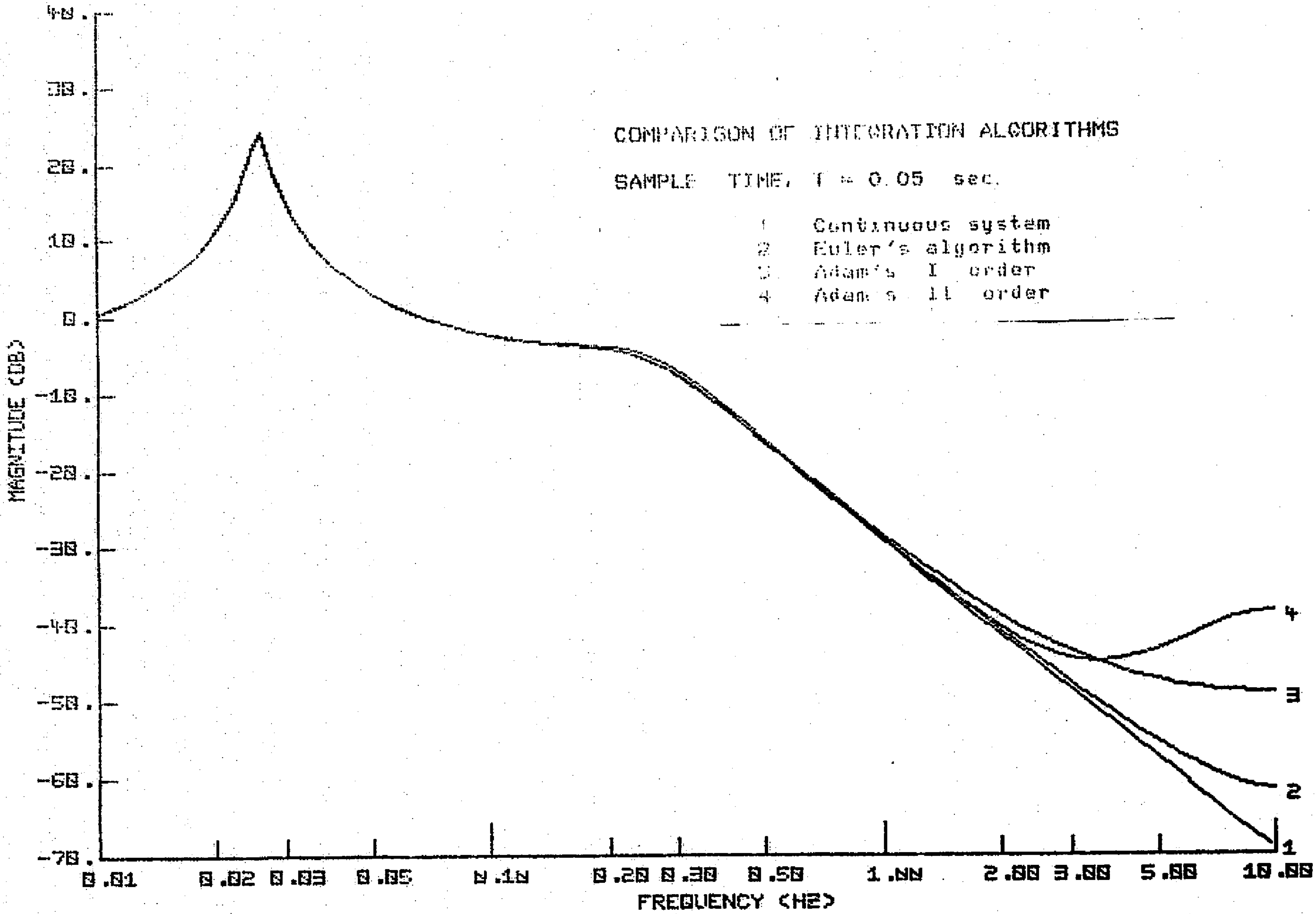
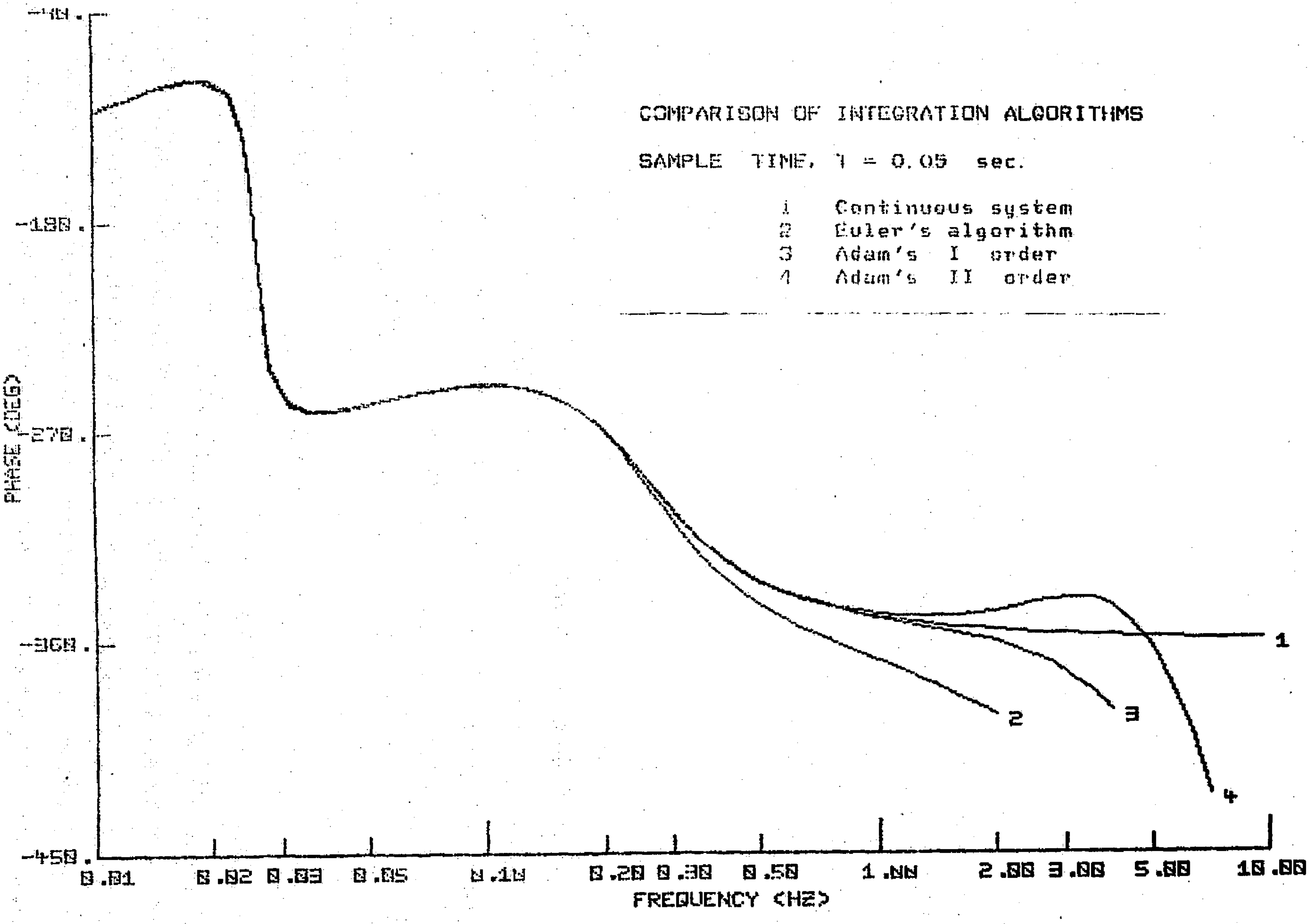

FREQUENCY RESPONSE USING ADAM'S II ORDER ALGORITHM
1 continuous system
2 T = 0.05 sec. (discrete system)
3 T = 0.1 sec.
4 T = 0.2 sec.
5 T = 0.33 sec.
6 T = 0.5 sec.
-180.
-450.
0.02 0.03 0.05
0.20 0.30 0.50
1.00 2.00 3.00 5.00 10.00
FREQUENCY (HZ)
1
2
3
4
5
6

FIGURE 3b



COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME, T = 0.05 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4

FIGURE 4a



COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME, T = 0.05 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
-180.
-270.
-360.
-450.
PHASE (DEG)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4

FIGURE 4b

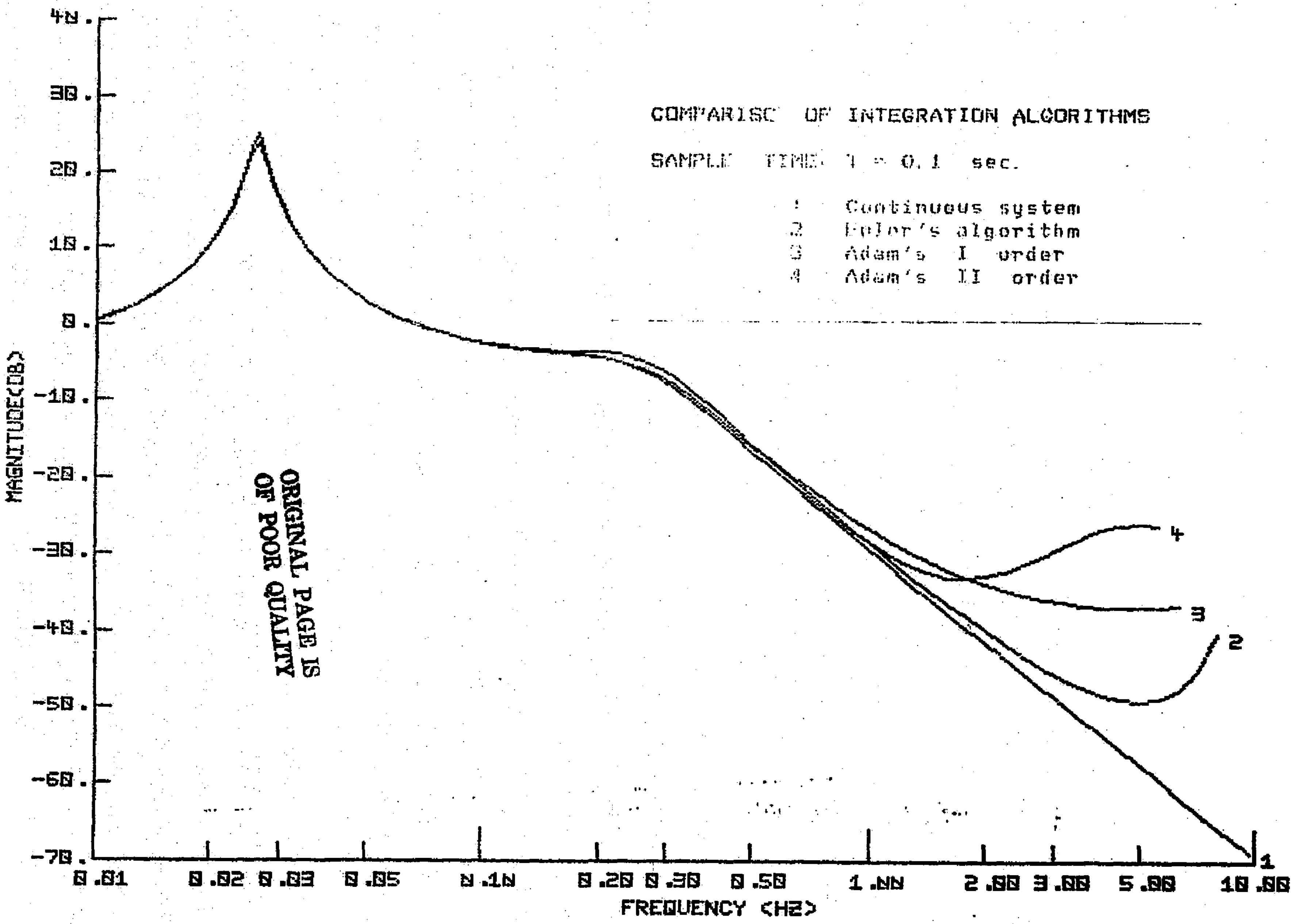

COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME T = 0.1 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
MAGNITUDE(DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4

FIGURE 5a

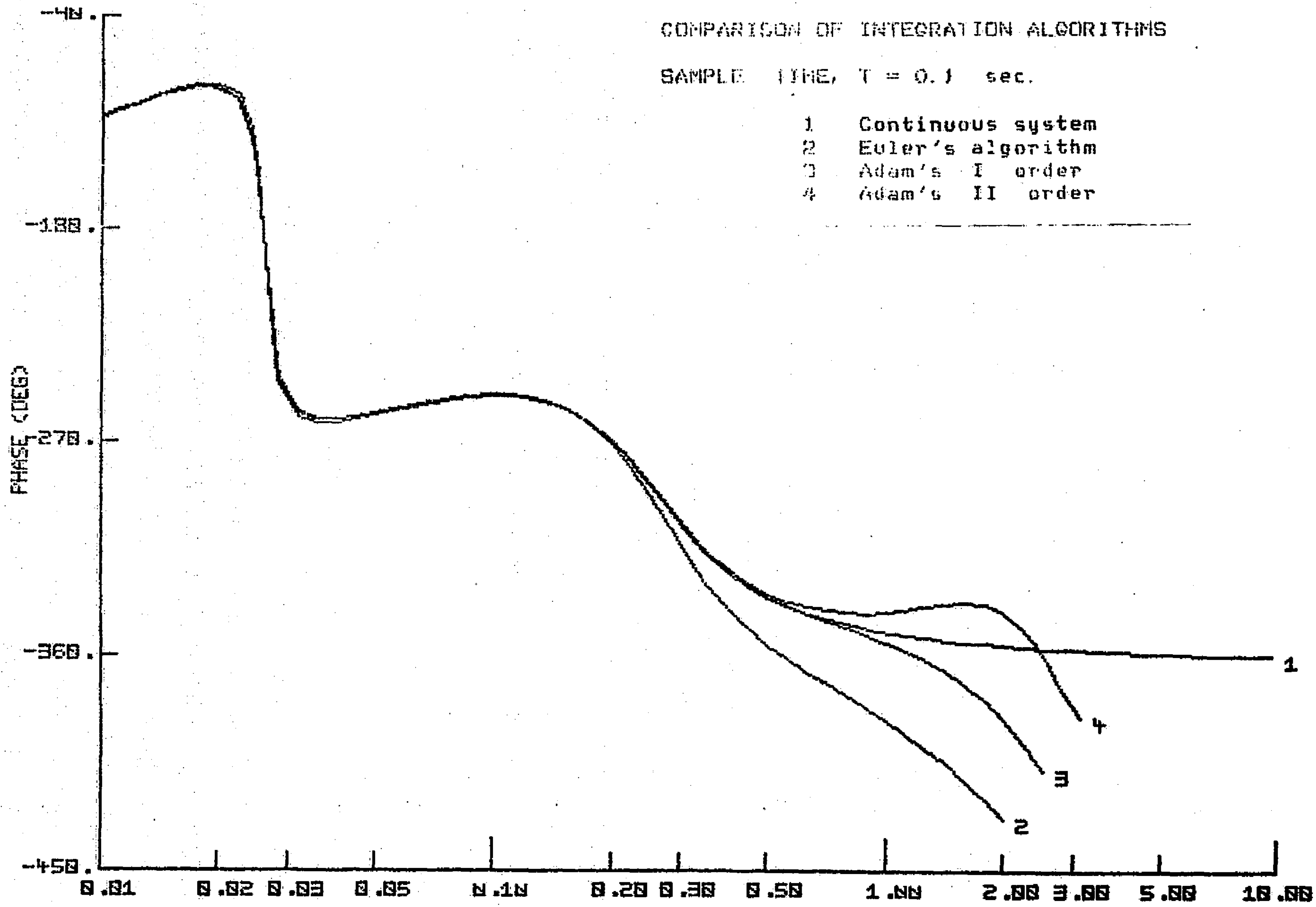

COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME, T = 0.1 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
PHASE (DEG)
-40.
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4

FIGURE 5b

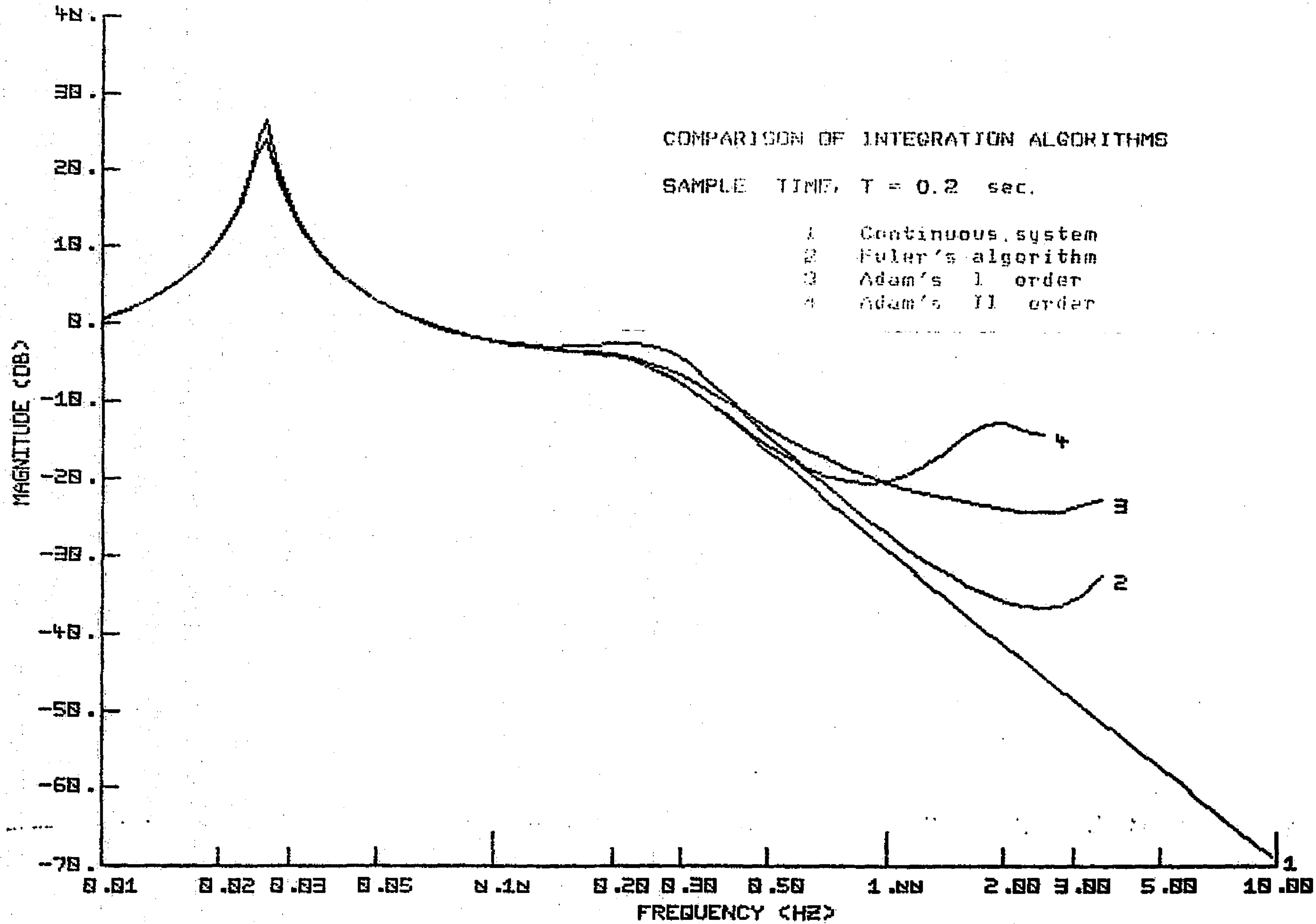

COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME, T = 0.2 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
FREQUENCY (HZ)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
1
2
3
4

FIGURE 6a

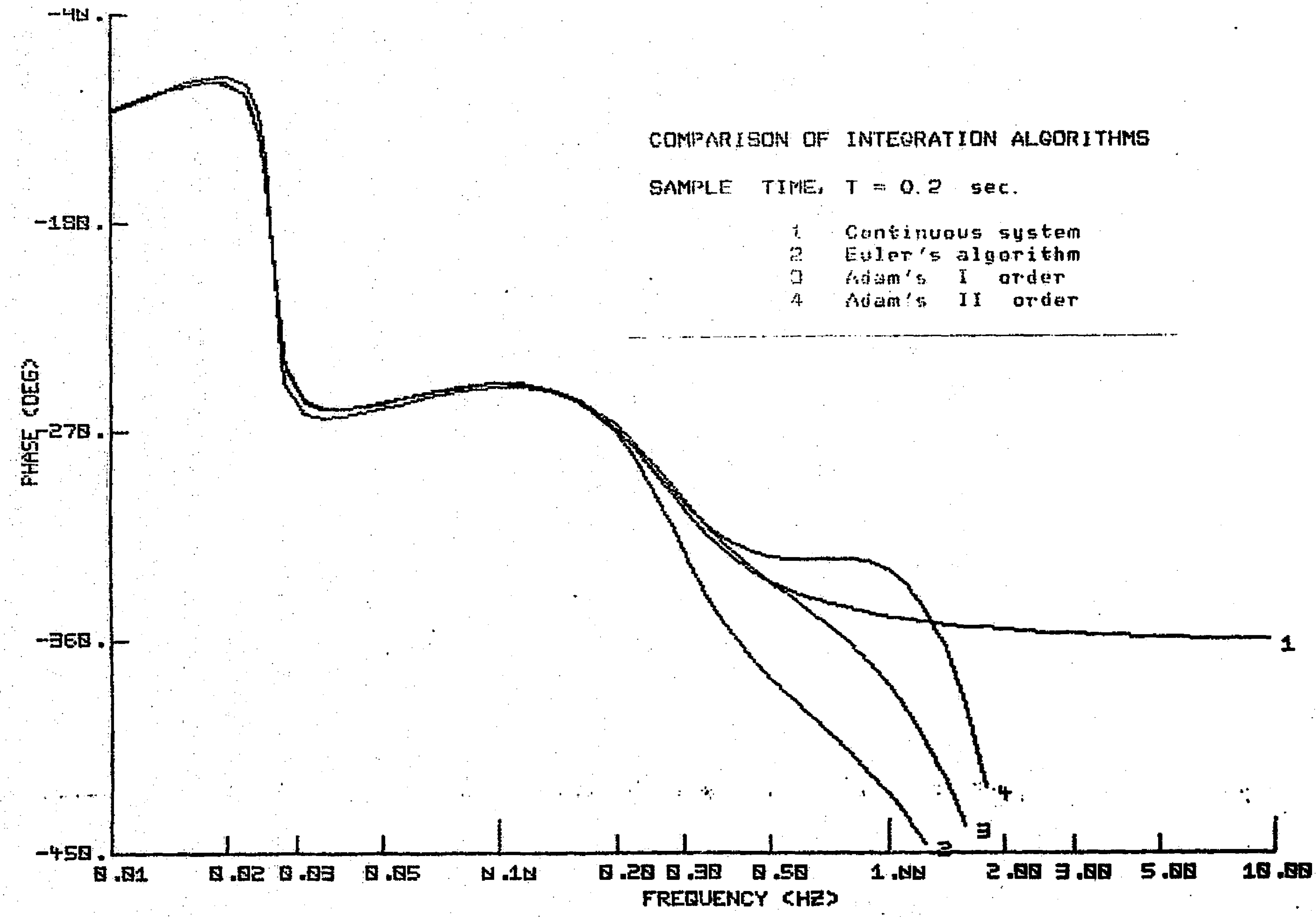

COMPARISON OF INTEGRATION ALGORITHMS
SAMPLE TIME, T = 0.2 sec.
1 Continuous system
2 Euler's algorithm
3 Adam's I order
4 Adam's II order
PHASE (DEG)
-90.
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2
3
4

FIGURE 6b

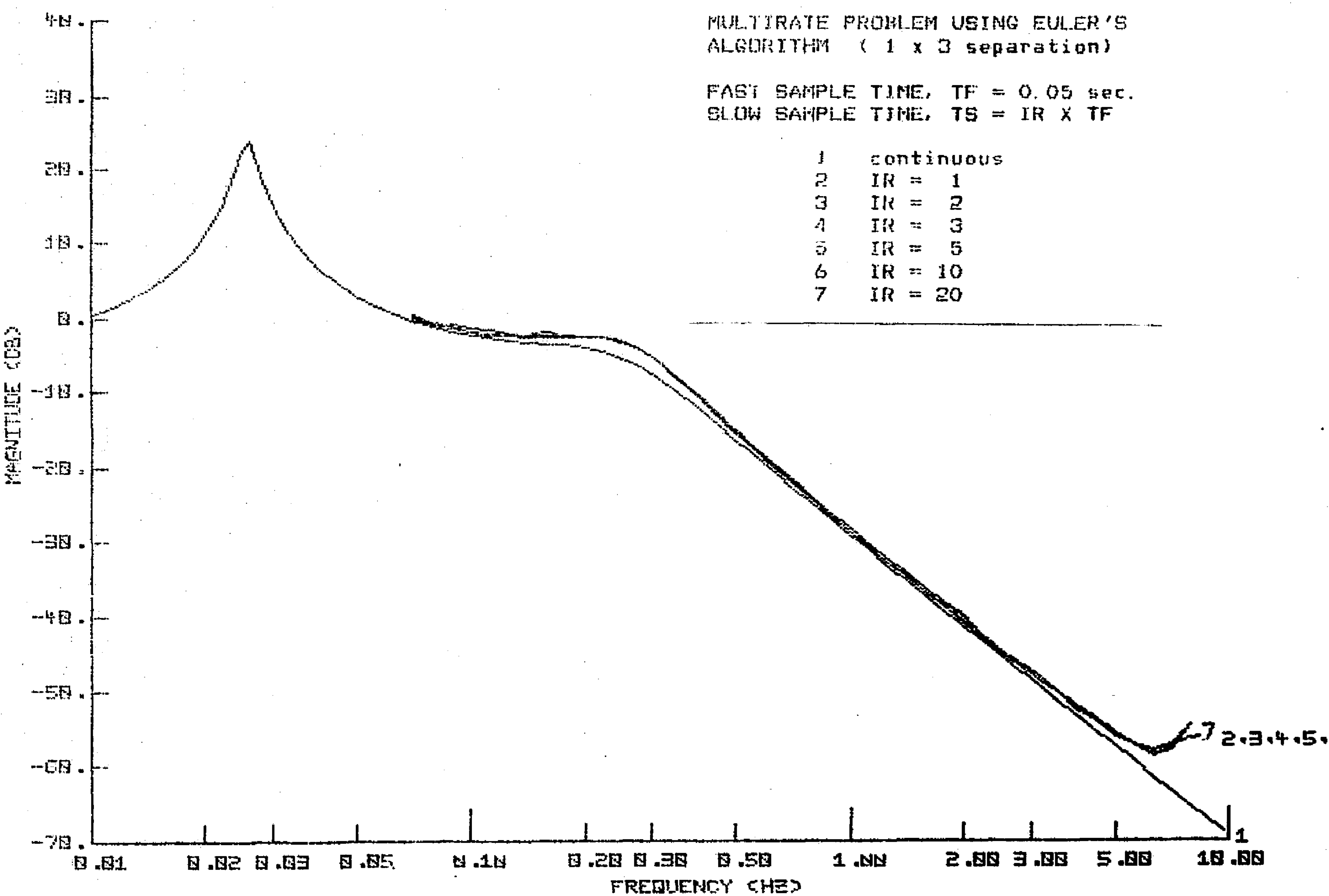

MULTIRATE PROBLEM USING EULER'S
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
7 IR = 20
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02 0.03
0.05
0.10
0.20 0.30
0.50
1.00
2.00 3.00
5.00
10.00
FREQUENCY (HZ)
7
2,3,4,5,6
1

FIGURE 7a

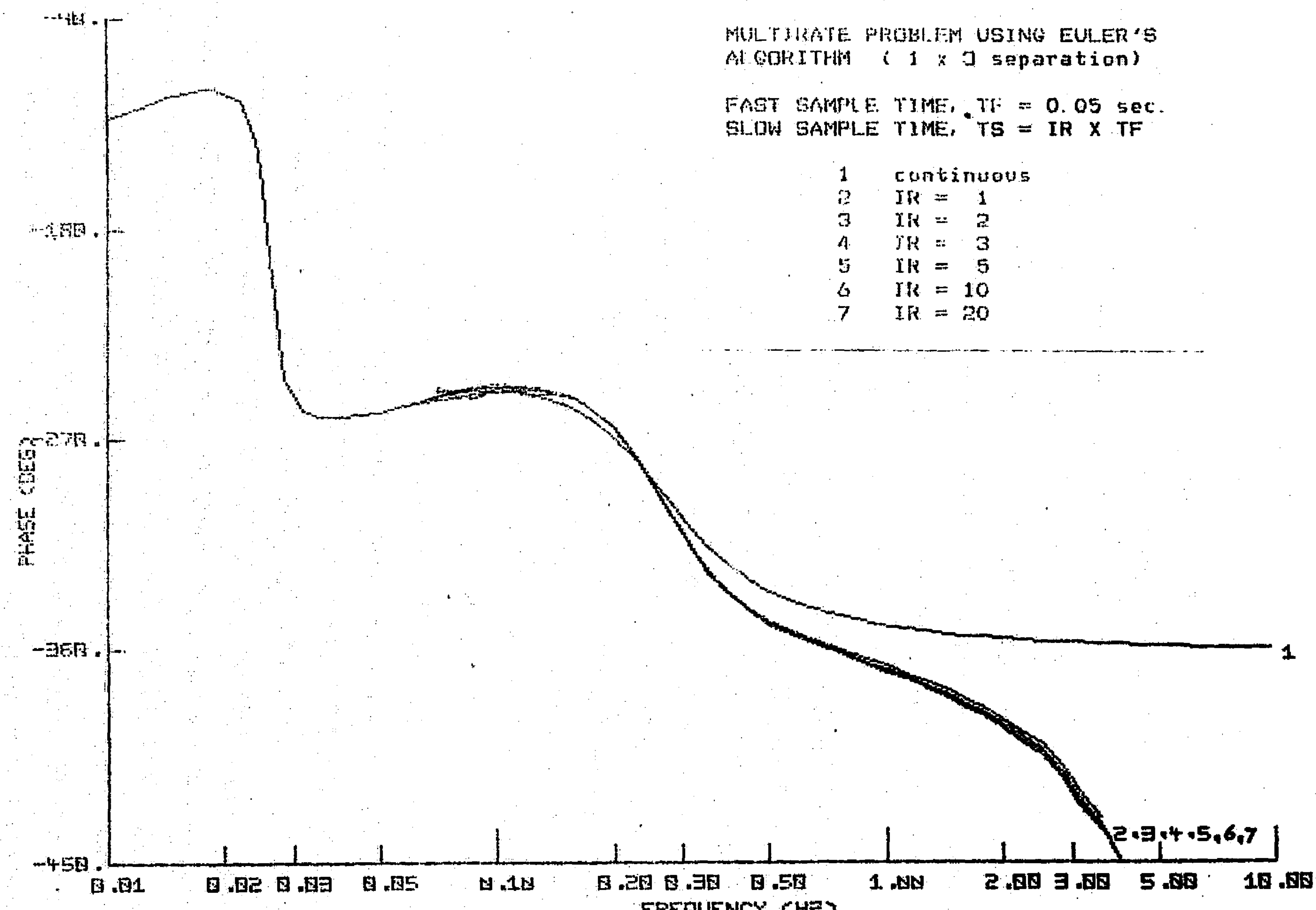

MULTIRATE PROBLEM USING EULER'S ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
7 IR = 20
PHASE (DEG)
-90.
-180.
-270.
-360.
-450.
FREQUENCY (HZ)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
1
2,3,4,5,6,7

FIGURE 7b

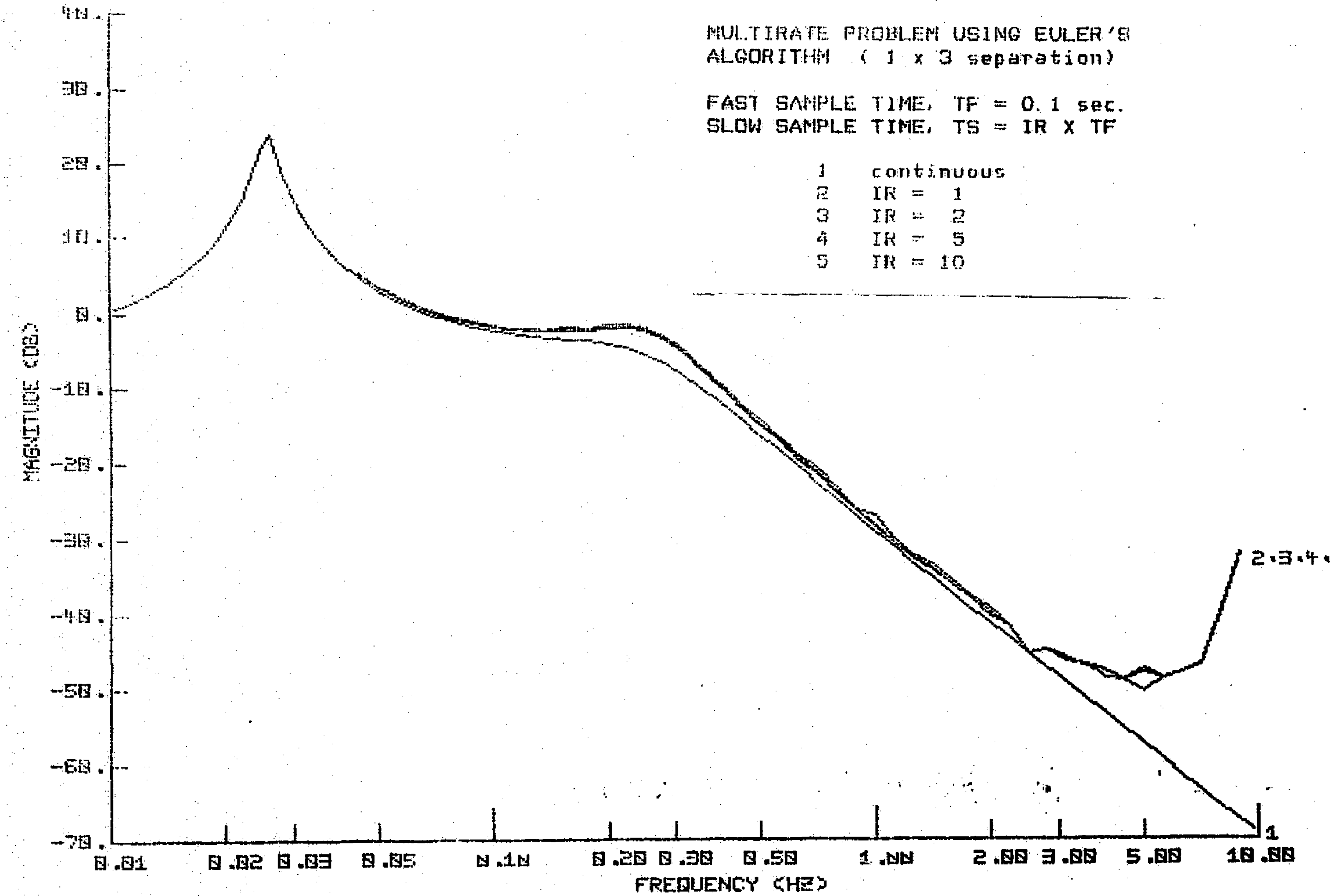

MULTIRATE PROBLEM USING EULER'S
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
5 IR = 10
MAGNITUDE (DB)
FREQUENCY (HZ)
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
2,3,4,
1

FIGURE 8a

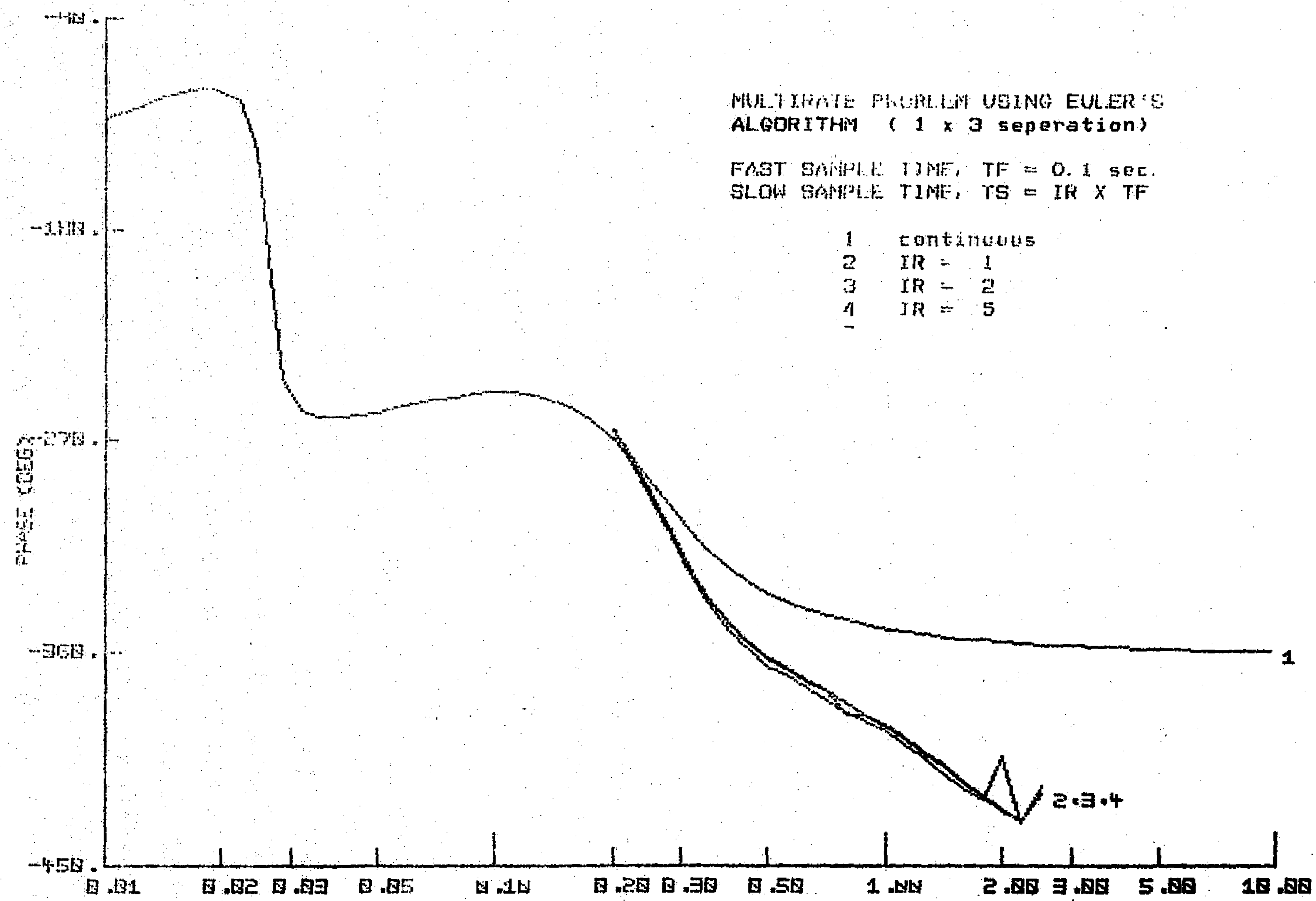

MULTIRATE PROBLEM USING EULER'S ALGORITHM (1 x 3 seperation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
-40.
-180.
-270.
-360.
-450.
PHASE (DEG)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2,3,4

FIGURE 8b

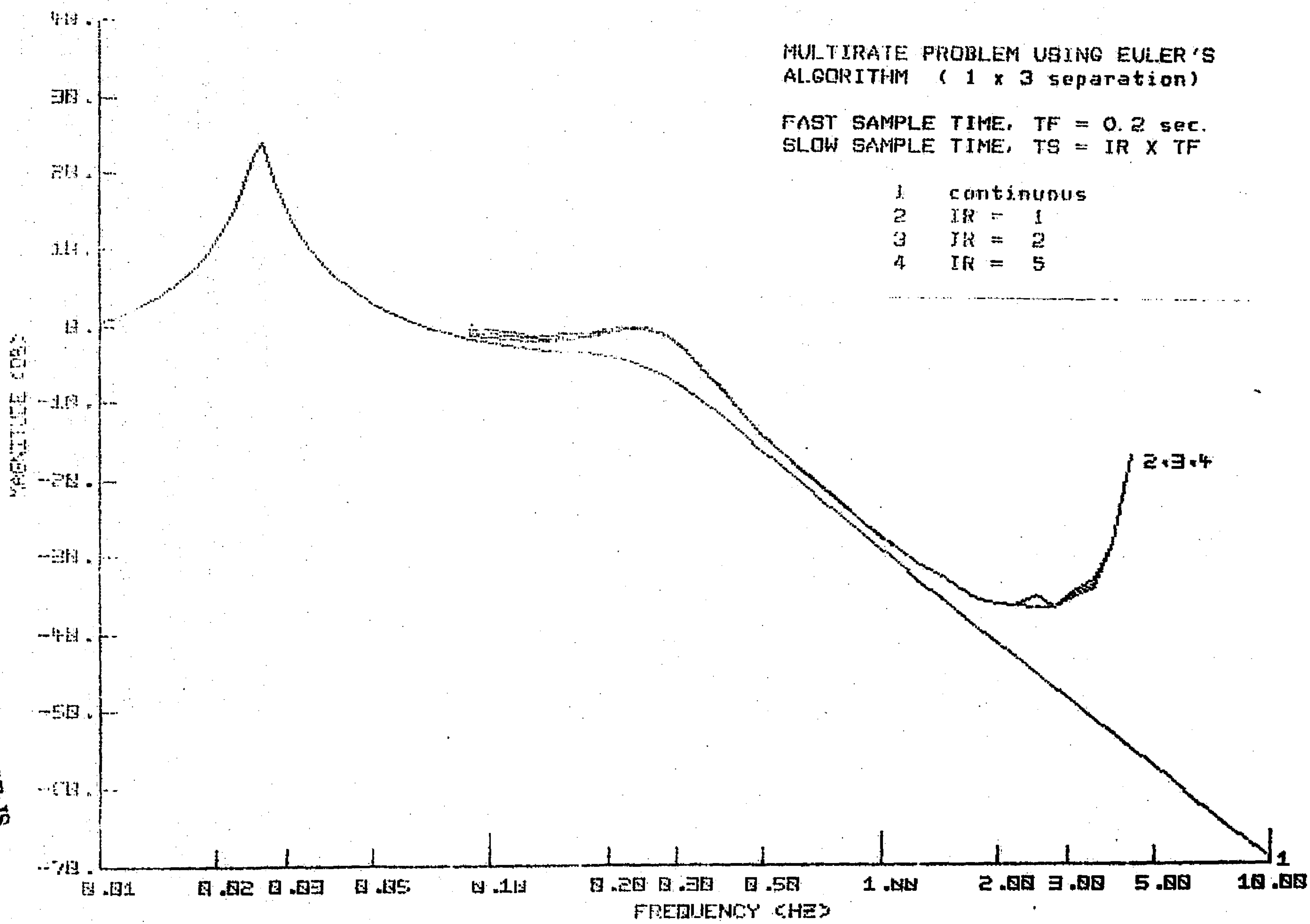
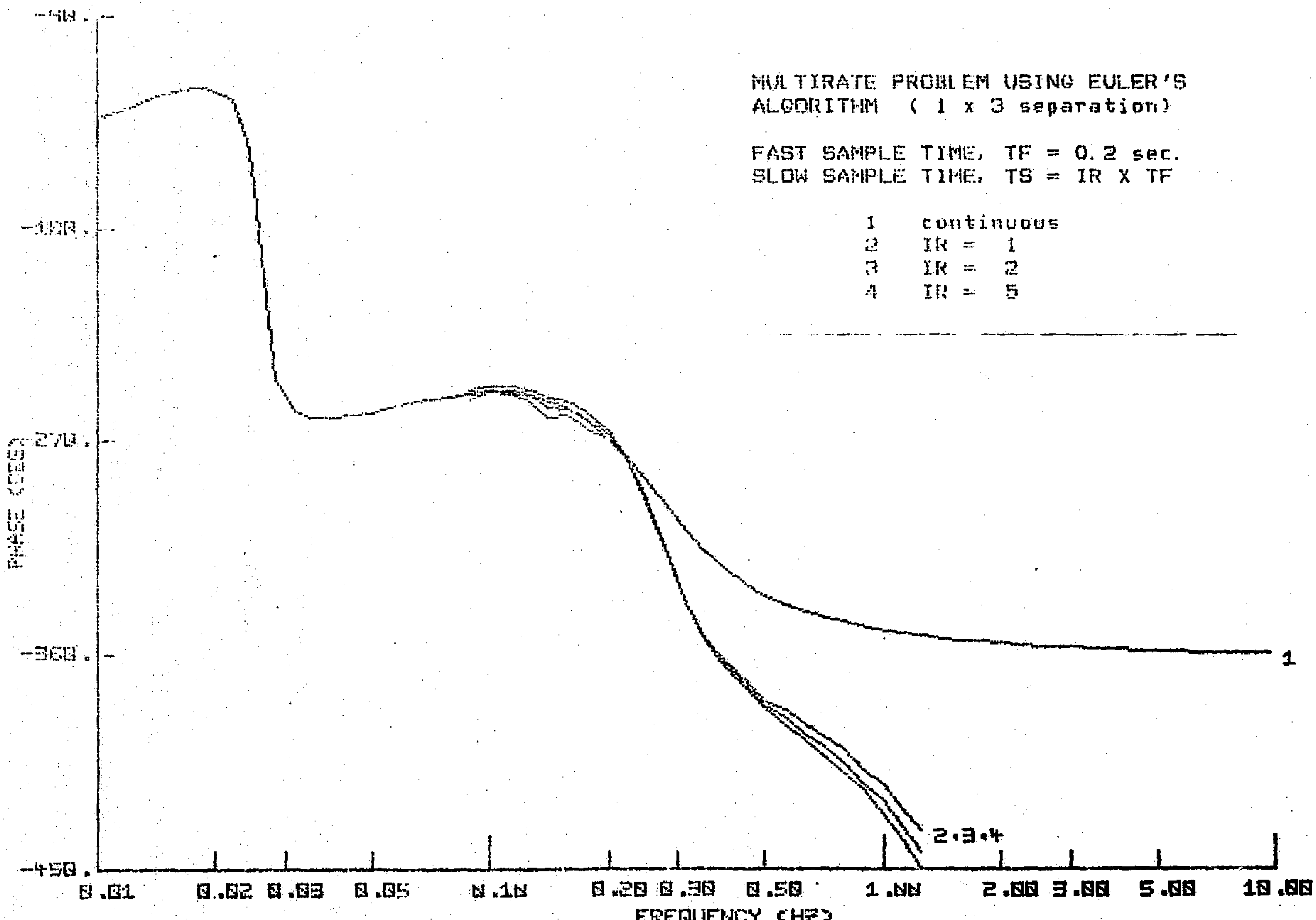

MULTIRATE PROBLEM USING EULER'S
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
2,3,4
1
0.01
0.02 0.03
0.05
0.10
0.20 0.30
0.50
1.00
2.00 3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 9a



MULTIRATE PROBLEM USING EULER'S
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
1
2,3,4
-270.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 9b

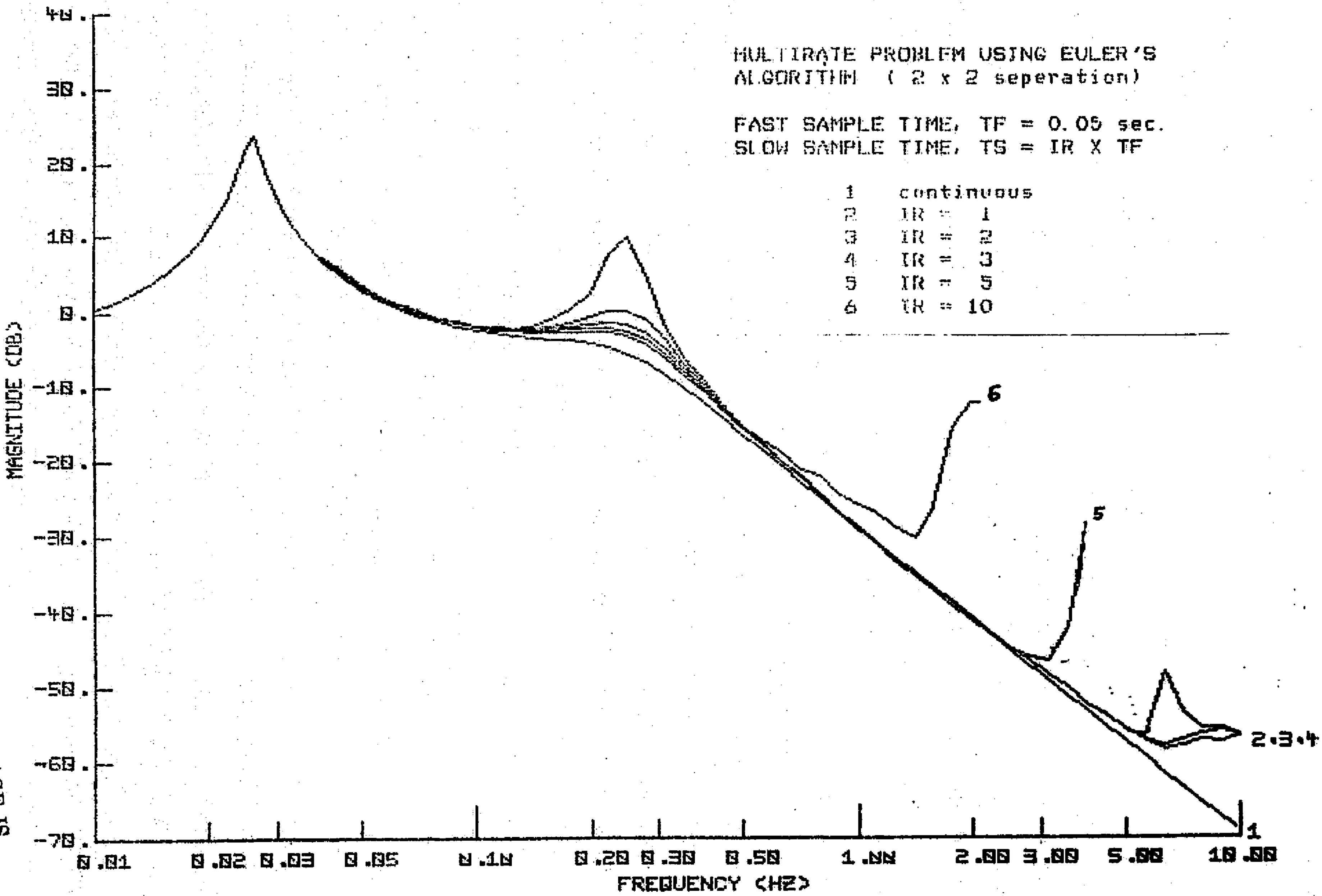

MULTIRATE PROBLEM USING EULER'S ALGORITHM (2 x 2 seperation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
FREQUENCY (HZ)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
6
5
2,3,4
1

FIGURE 10a

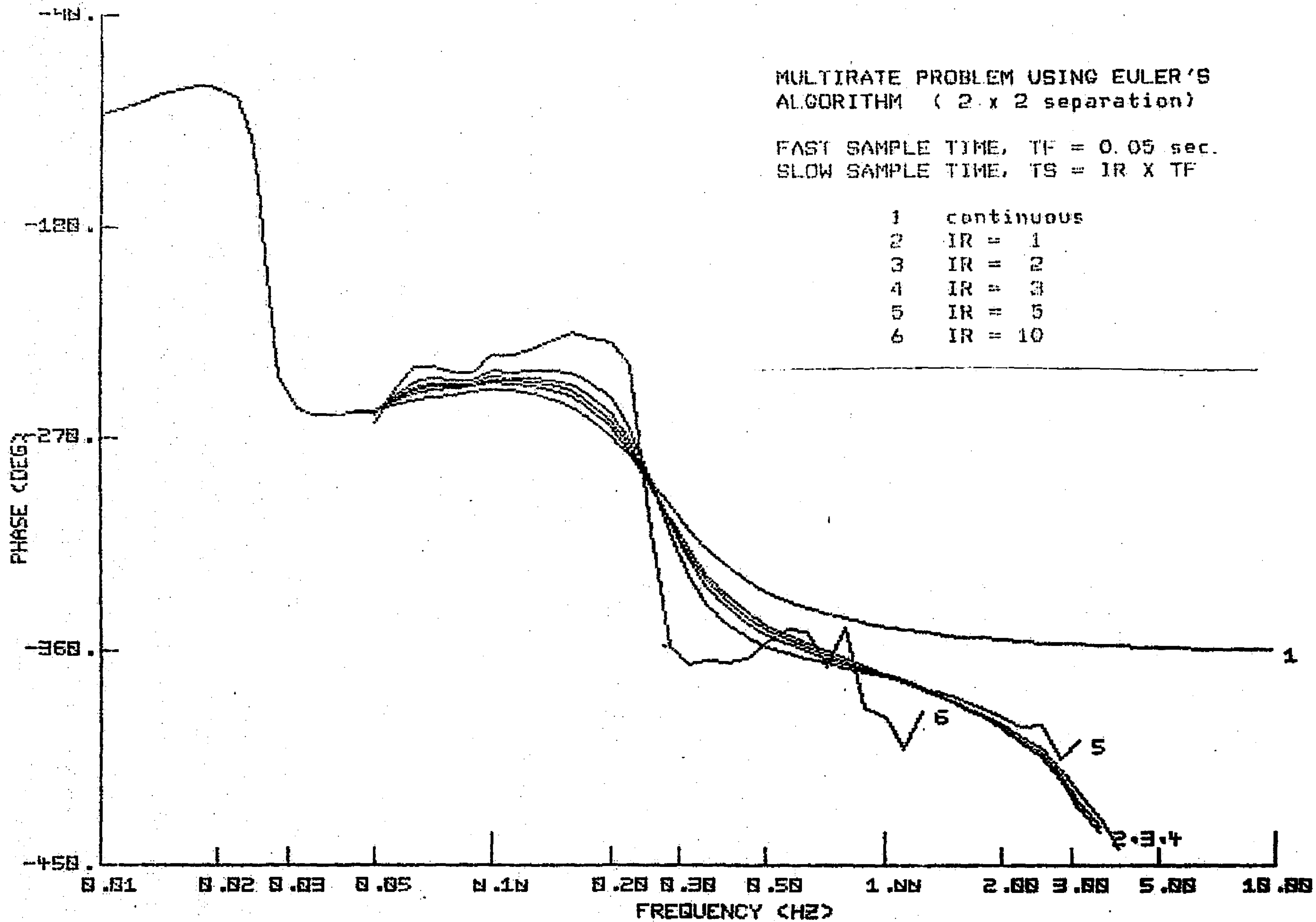

MULTIRATE PROBLEM USING EULER'S ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
PHASE (DEG)
-90.
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
6
5
2,3,4

FIGURE 10b

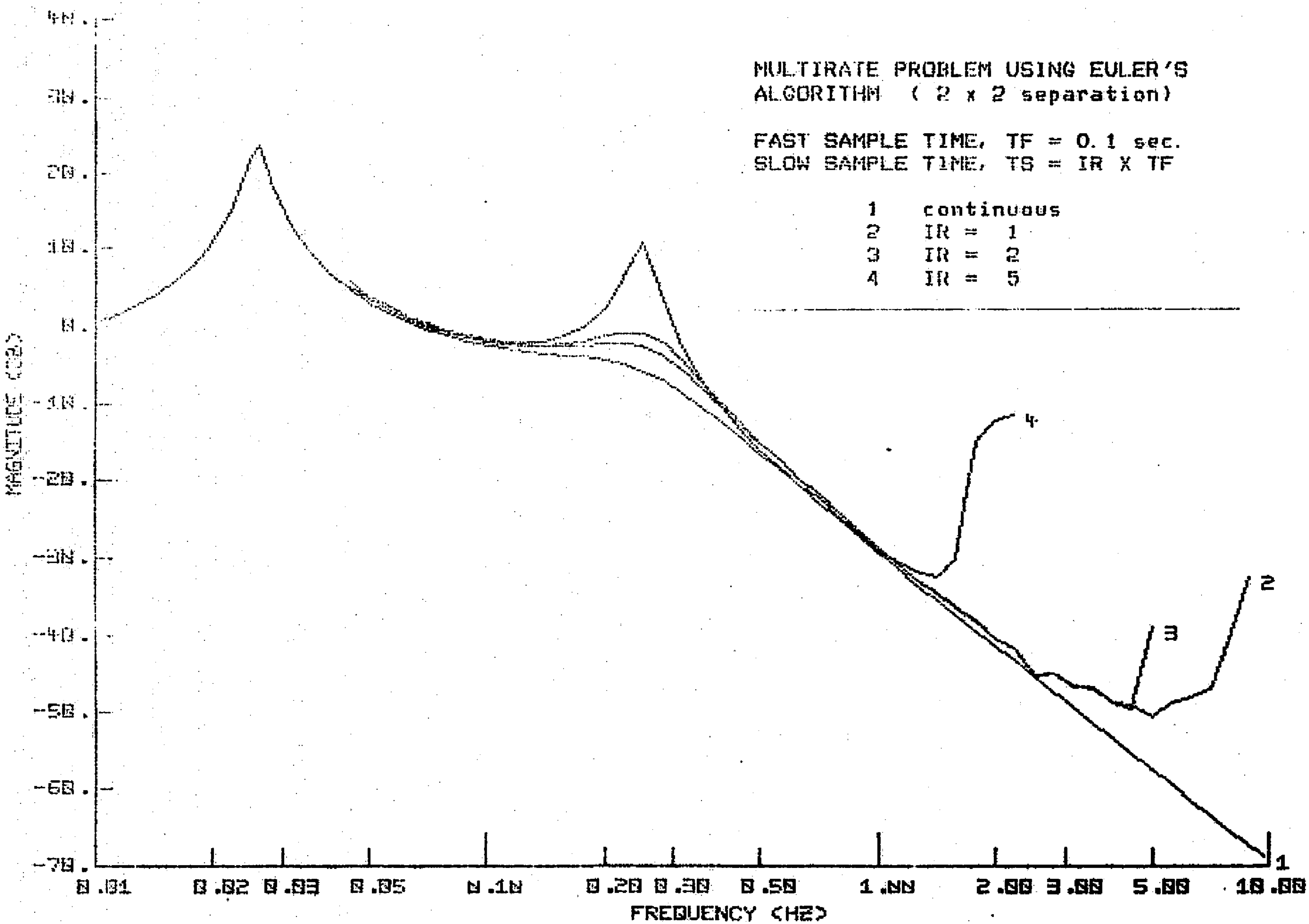

MULTIRATE PROBLEM USING EULER'S ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
MAGNITUDE (DB)
FREQUENCY (HZ)

FIGURE 11a

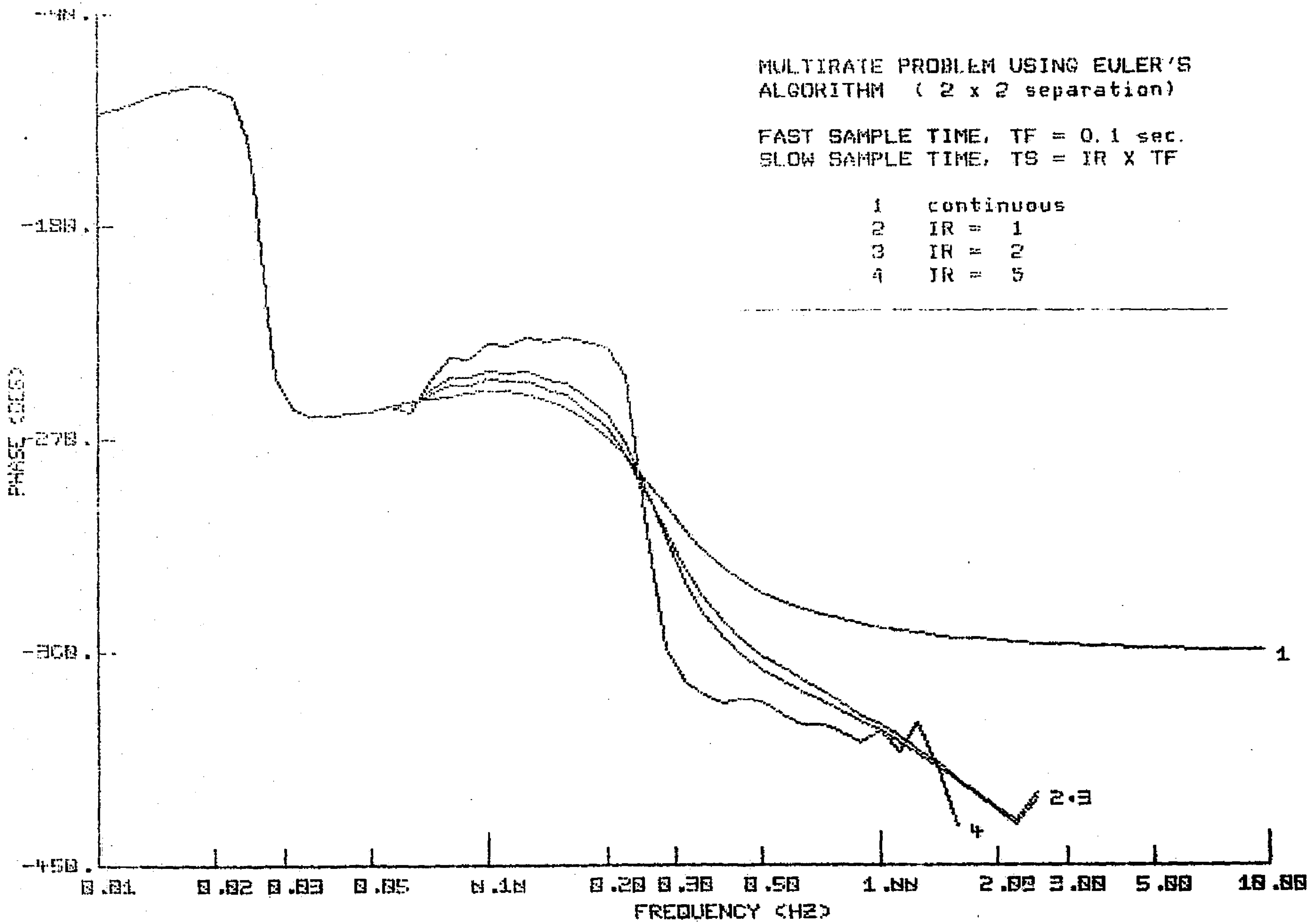

MULTIRATE PROBLEM USING EULER'S
ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
-180.
-270.
-360.
-450.
PHASE (DEG)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2,3
4

FIGURE 11b

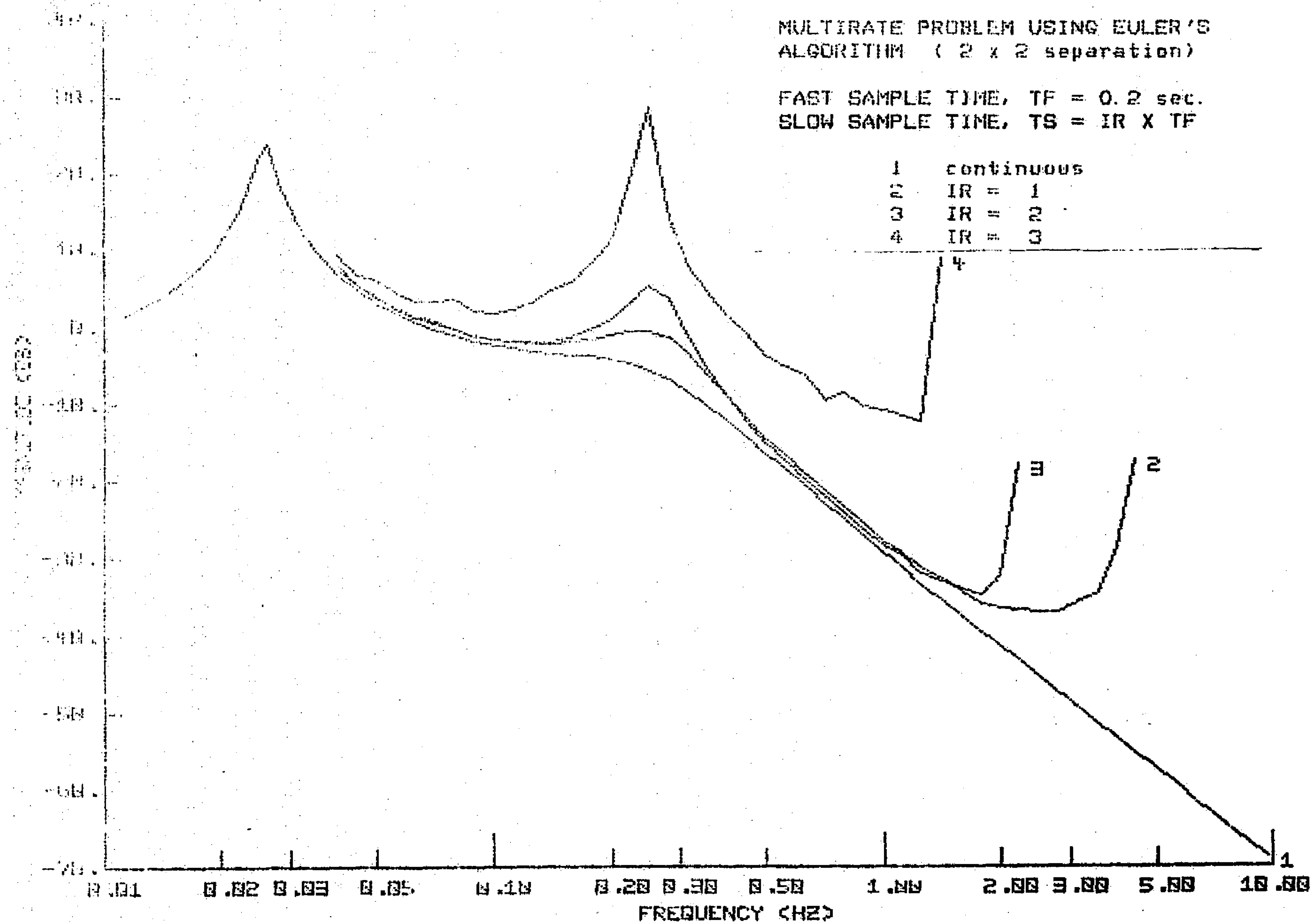

MULTIRATE PROBLEM USING EULER'S
ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 12a

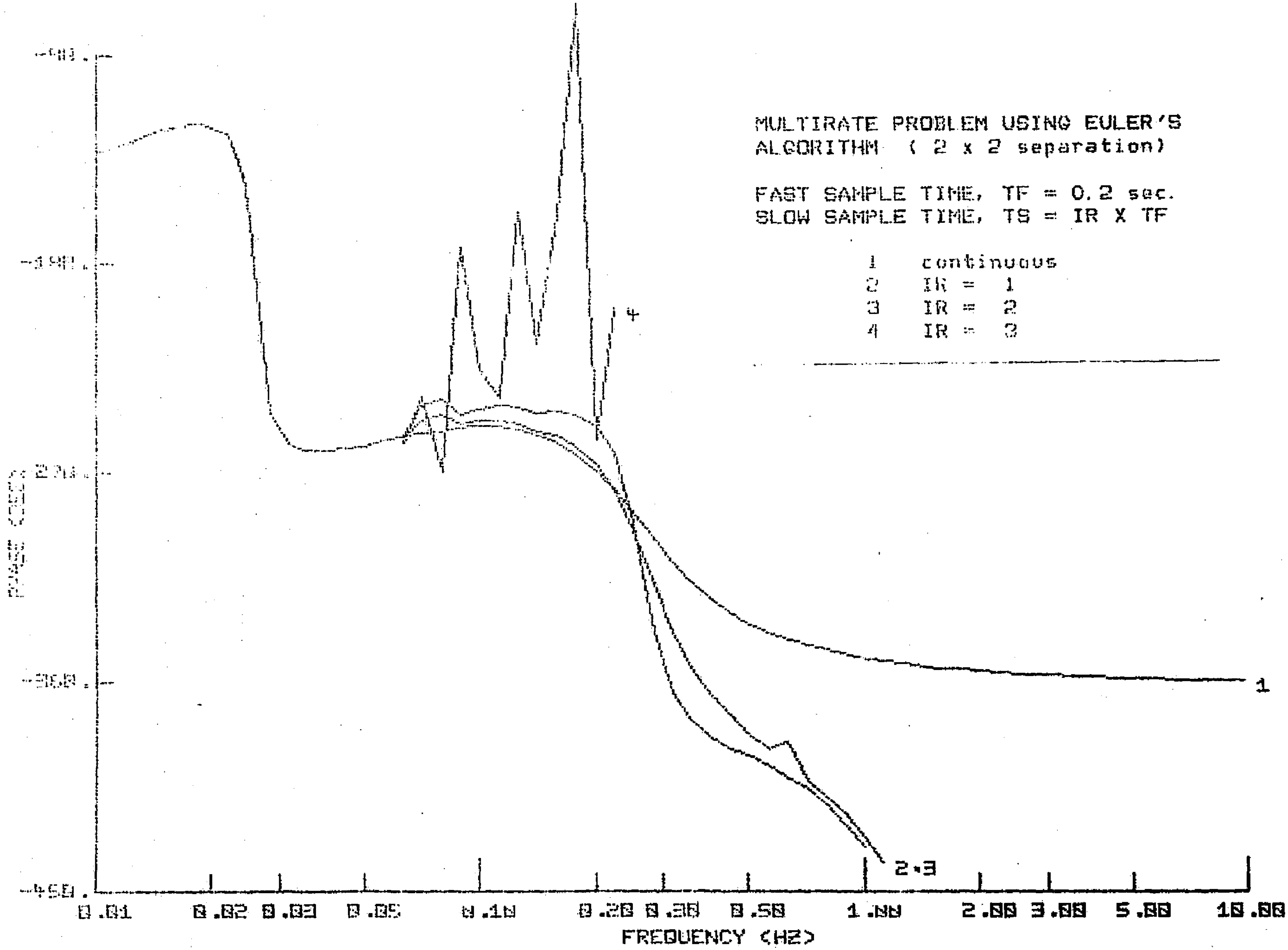

MULTIRATE PROBLEM USING EULER'S ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
4
1
2・3
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 12b

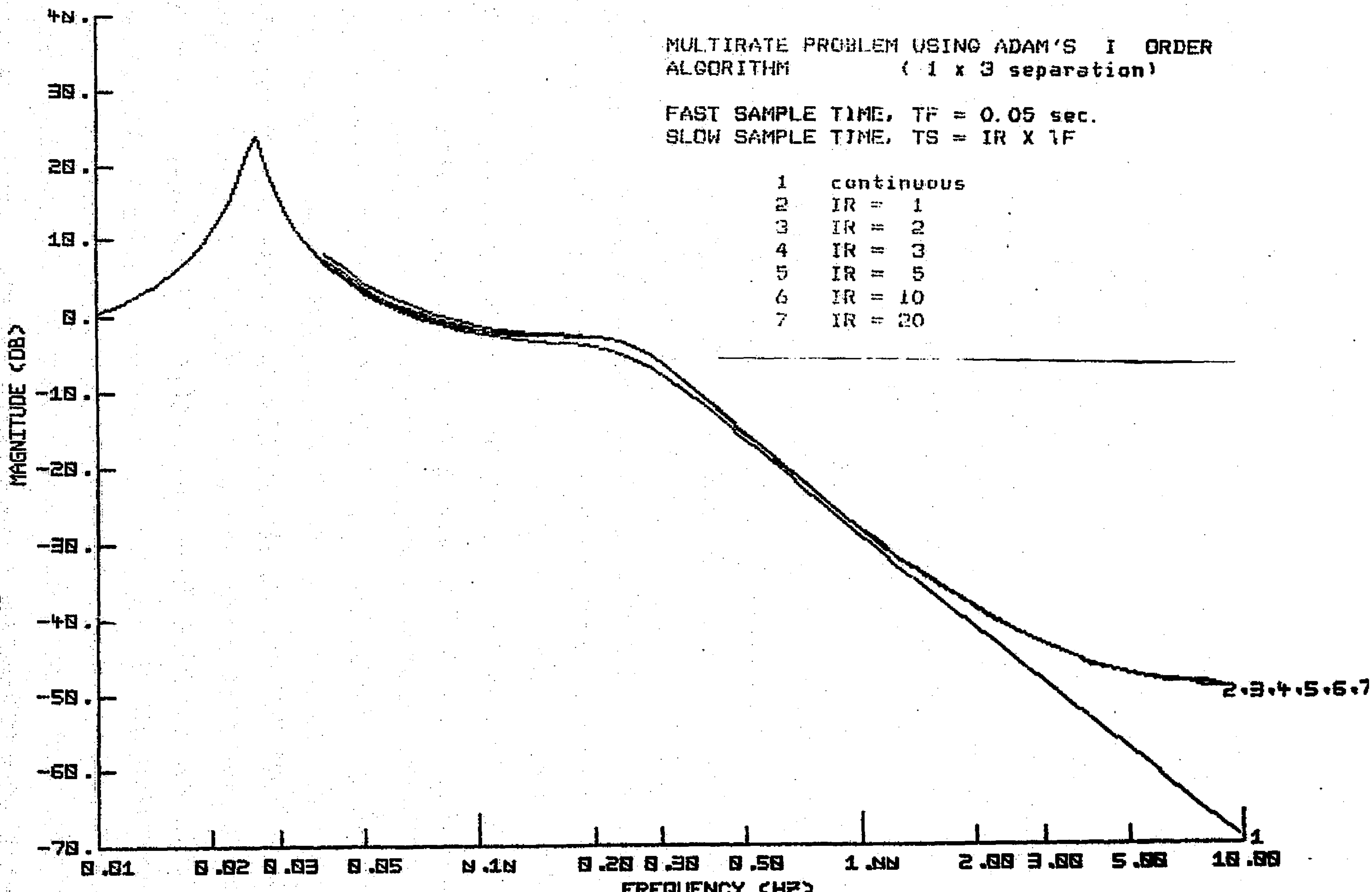

MULTIRATE PROBLEM USING ADAM'S I ORDER
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
7 IR = 20
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
FREQUENCY (HZ)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
2,3,4,5,6,7
1

FIGURE 13a

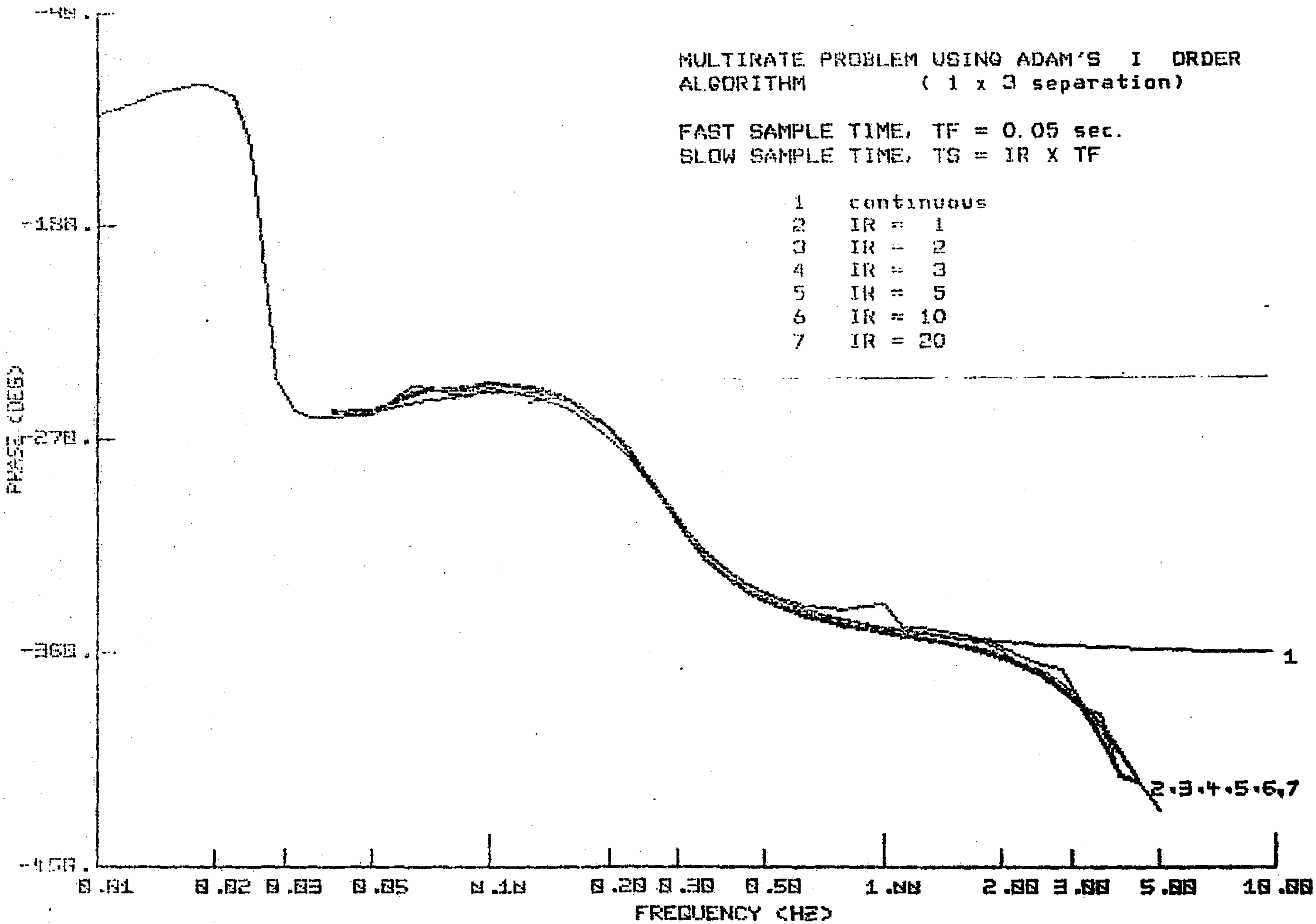

MULTIRATE PROBLEM USING ADAM'S I ORDER
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
7 IR = 20
PHASE (DEG)
-180.
-270.
-360.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2,3,4,5,6,7

FIGURE 13b

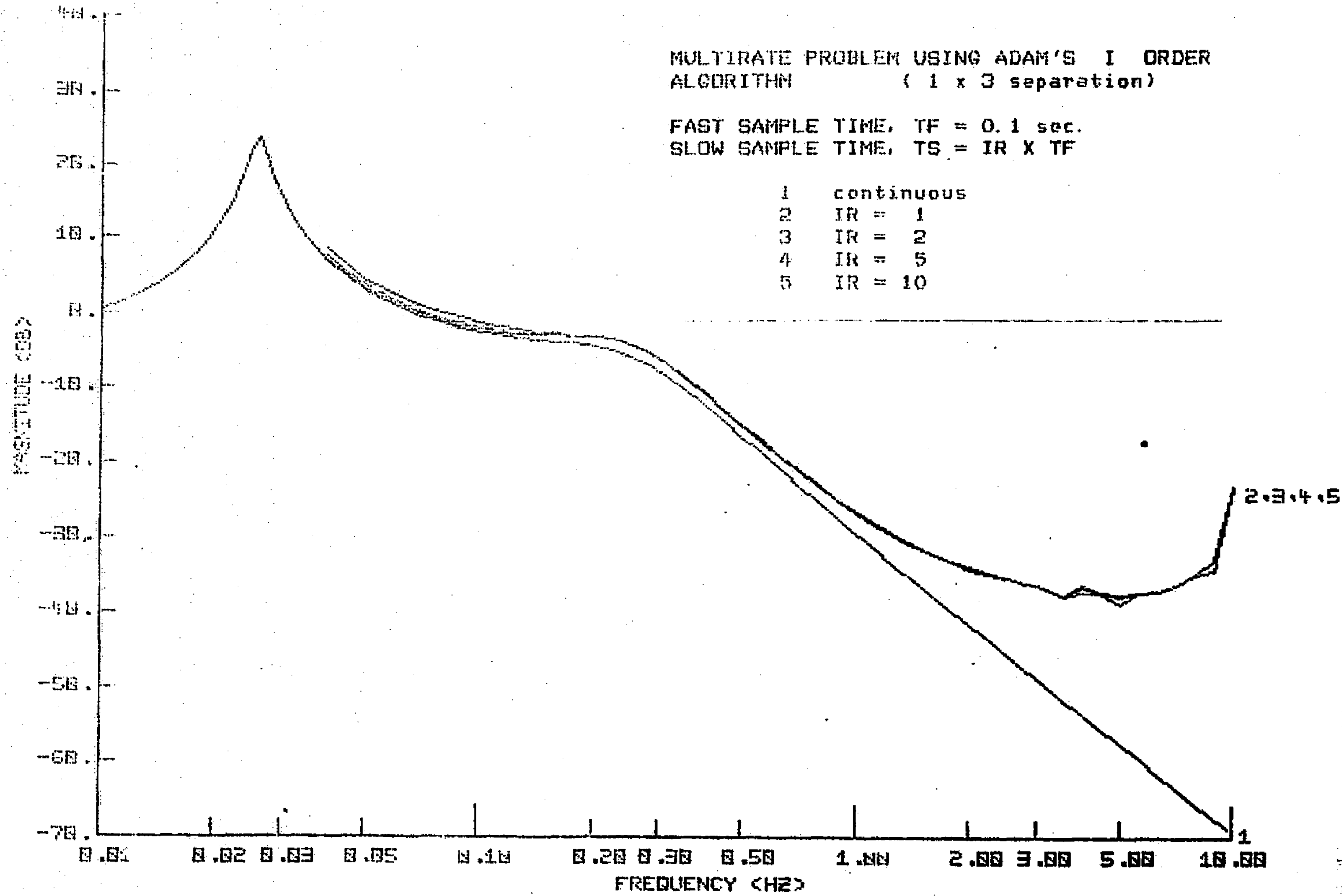

MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
5 IR = 10
2,3,4,5
1
MAGNITUDE (DB)
FREQUENCY (HZ)
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00

FIGURE 14a

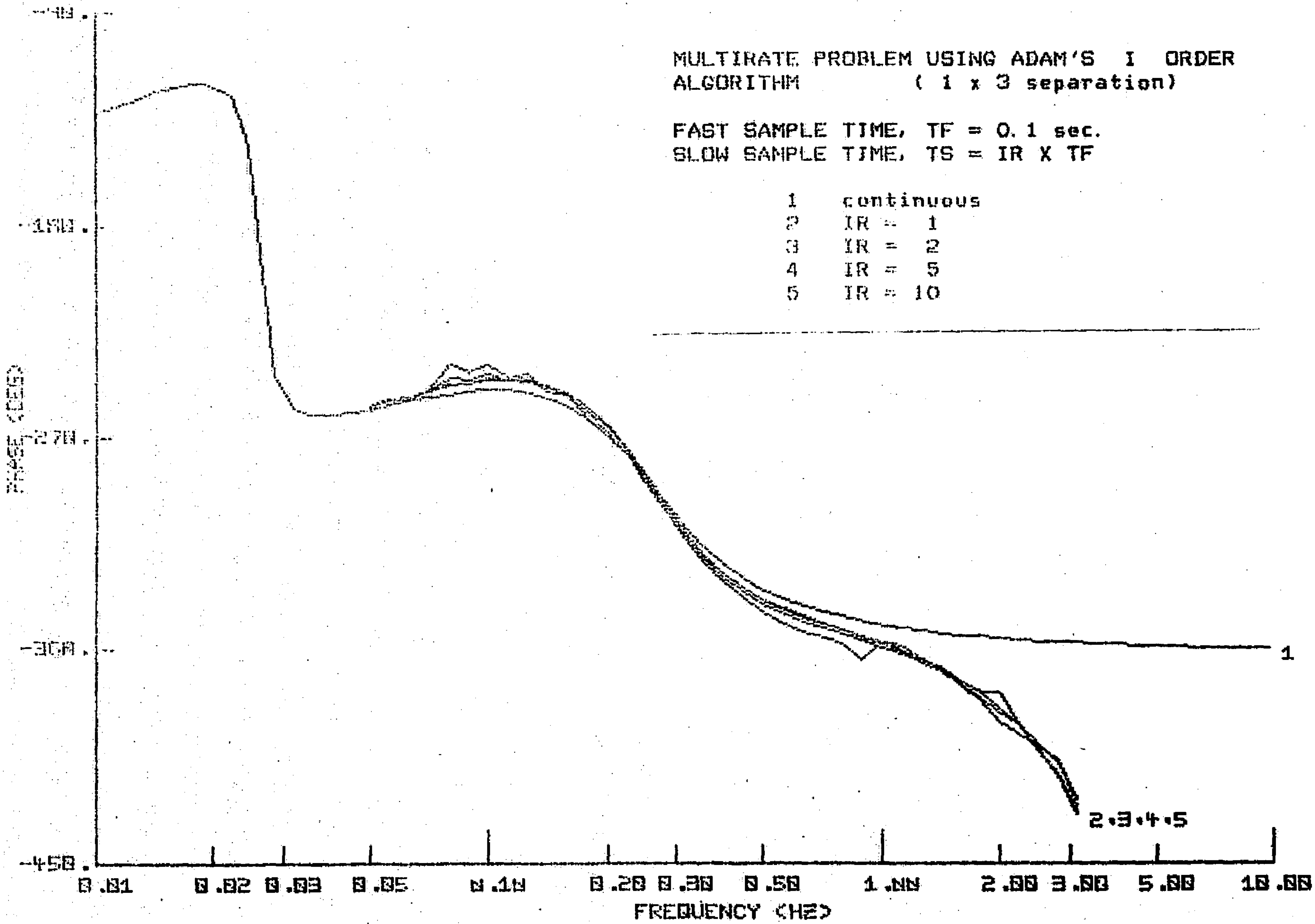

MULTIRATE PROBLEM USING ADAM'S I ORDER
ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
5 IR = 10
1
2,3,4,5
0.01 0.02 0.03 0.05 0.10 0.20 0.30 0.50 1.00 2.00 3.00 5.00 10.00
FREQUENCY (HZ)

FIGURE 14b

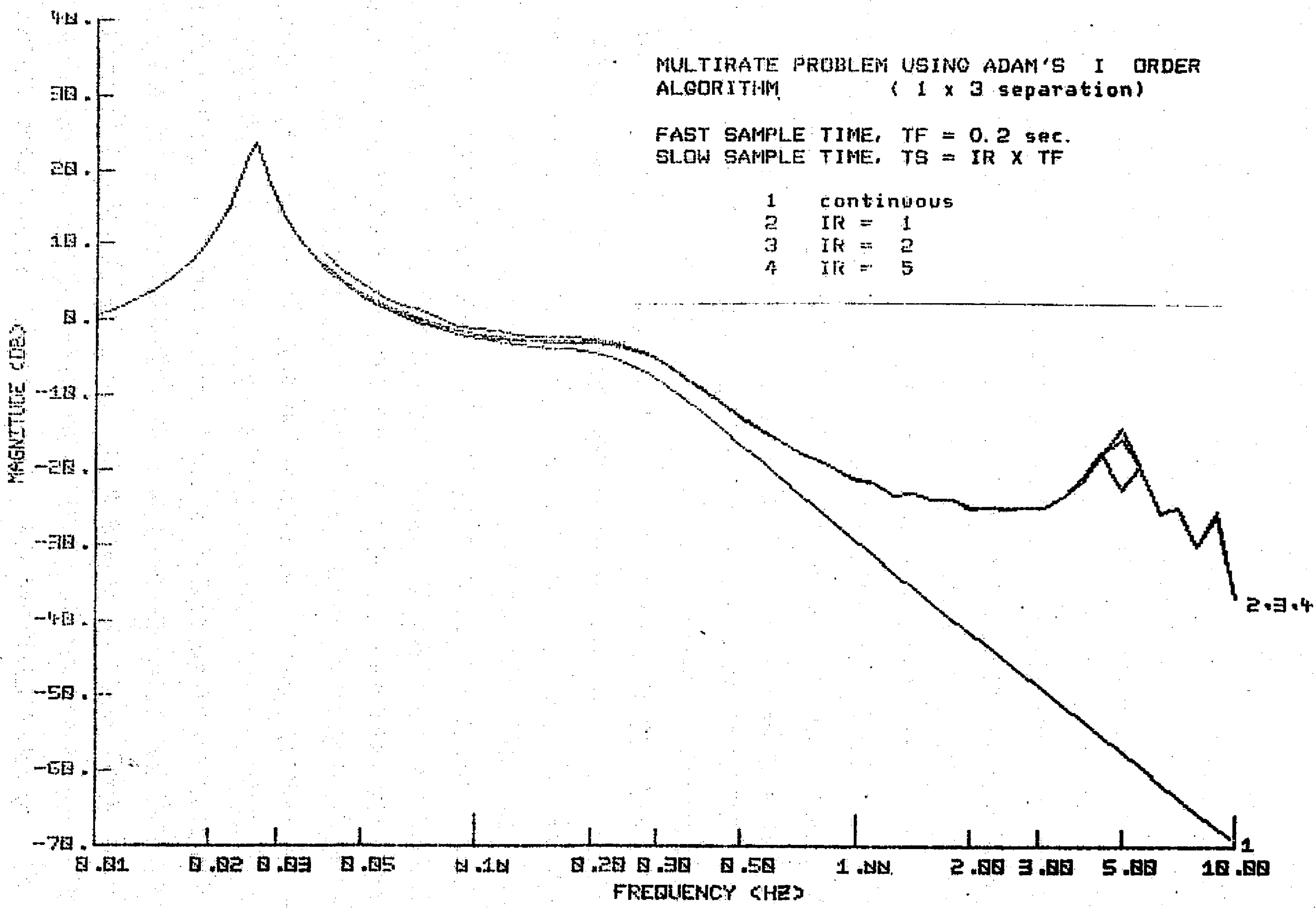

MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
2,3,4
1
0.01
0.02 0.03
0.05
0.10
0.20 0.30
0.50
1.00
2.00 3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 15a

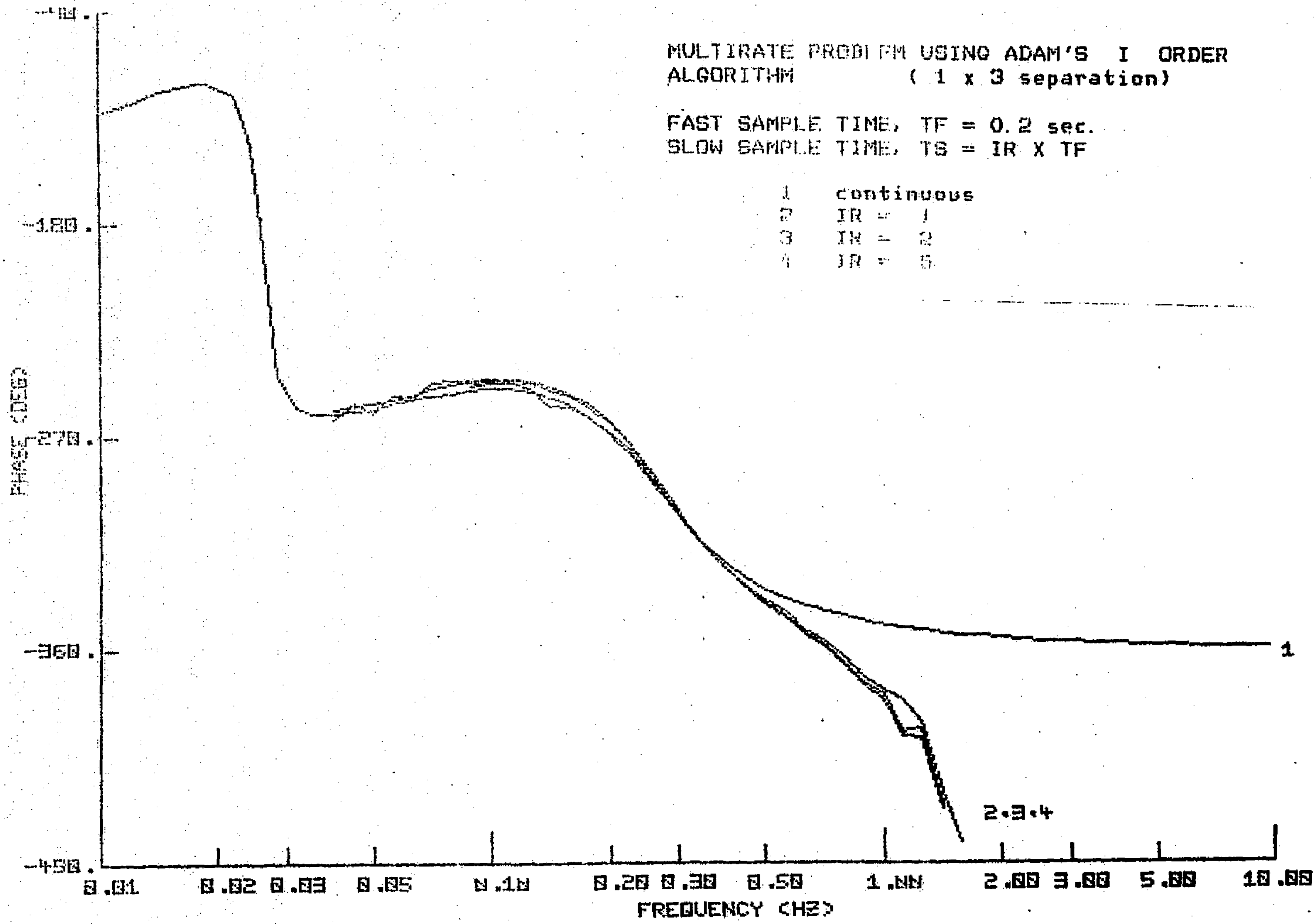

MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (1 x 3 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
PHASE (DEG)
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2.3.4

FIGURE 15b

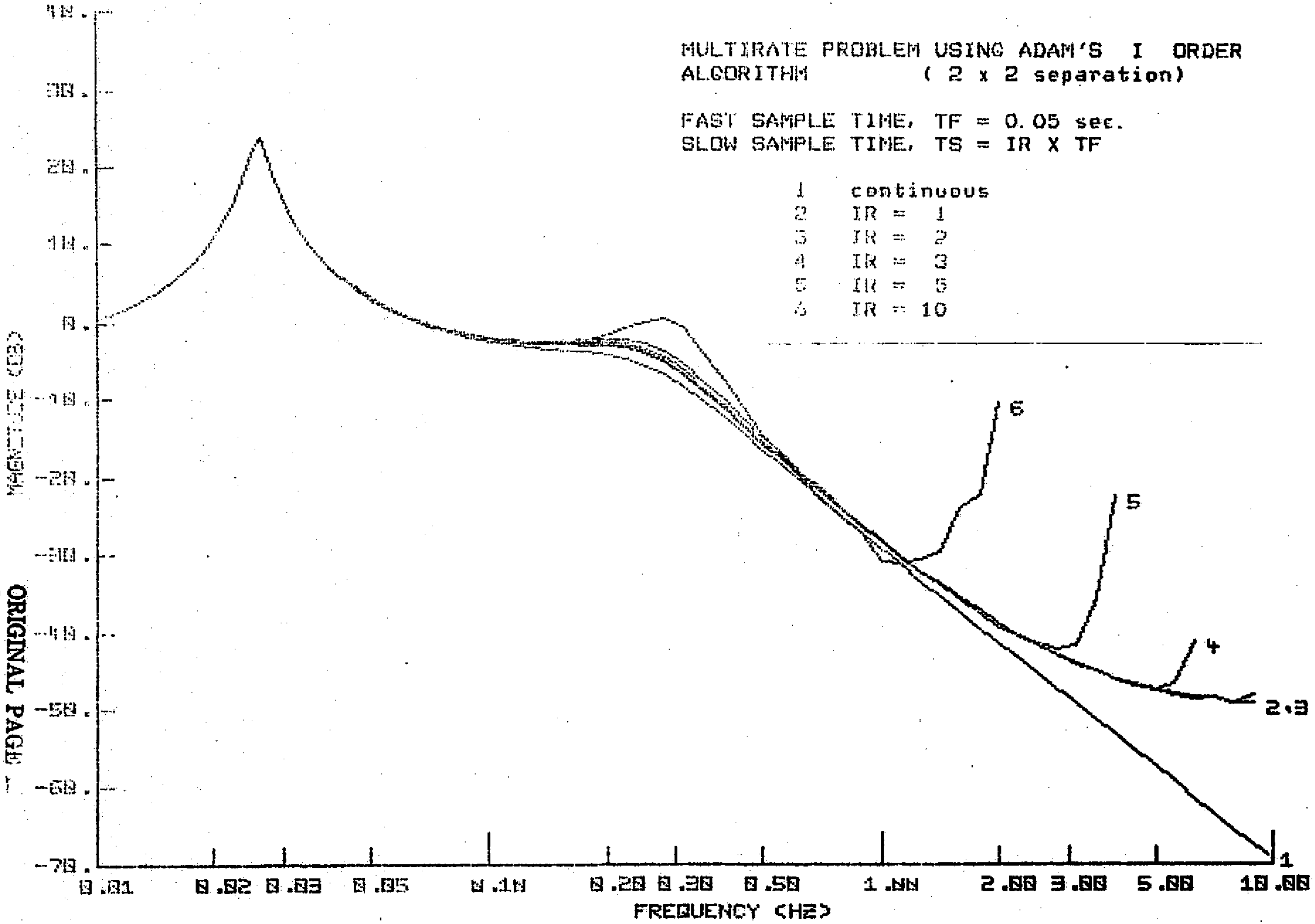

MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
MAGNITUDE (DB)
FREQUENCY (HZ)
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
1
2,3
4
5
6

FIGURE 16a

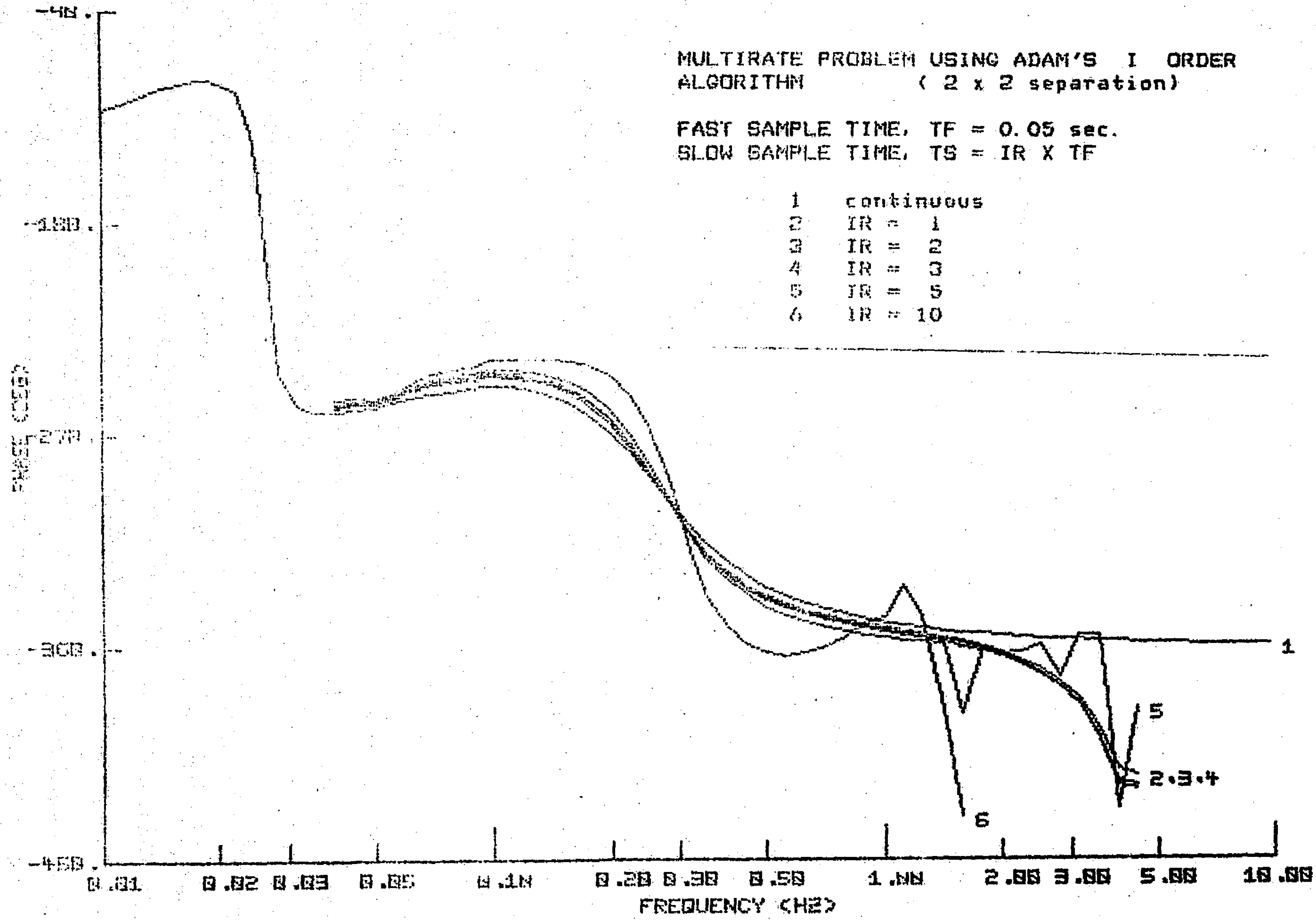

MULTIRATE PROBLEM USING ADAM'S I ORDER
ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.05 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
5 IR = 5
6 IR = 10
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
5
2,3,4
6

FIGURE 16b

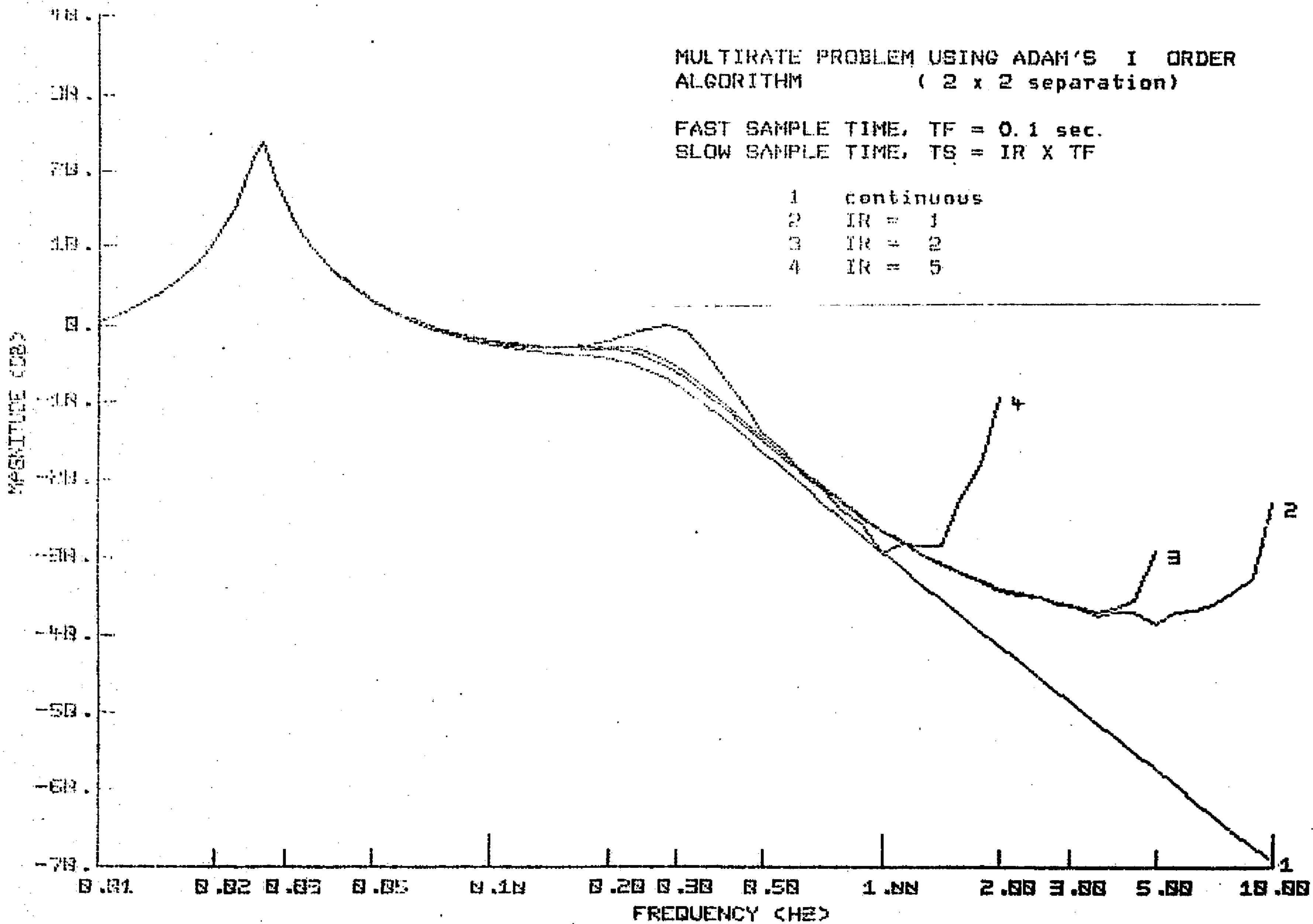


FIGURE 17a

ORIGINAL PAGE IS OF POOR QUALITY

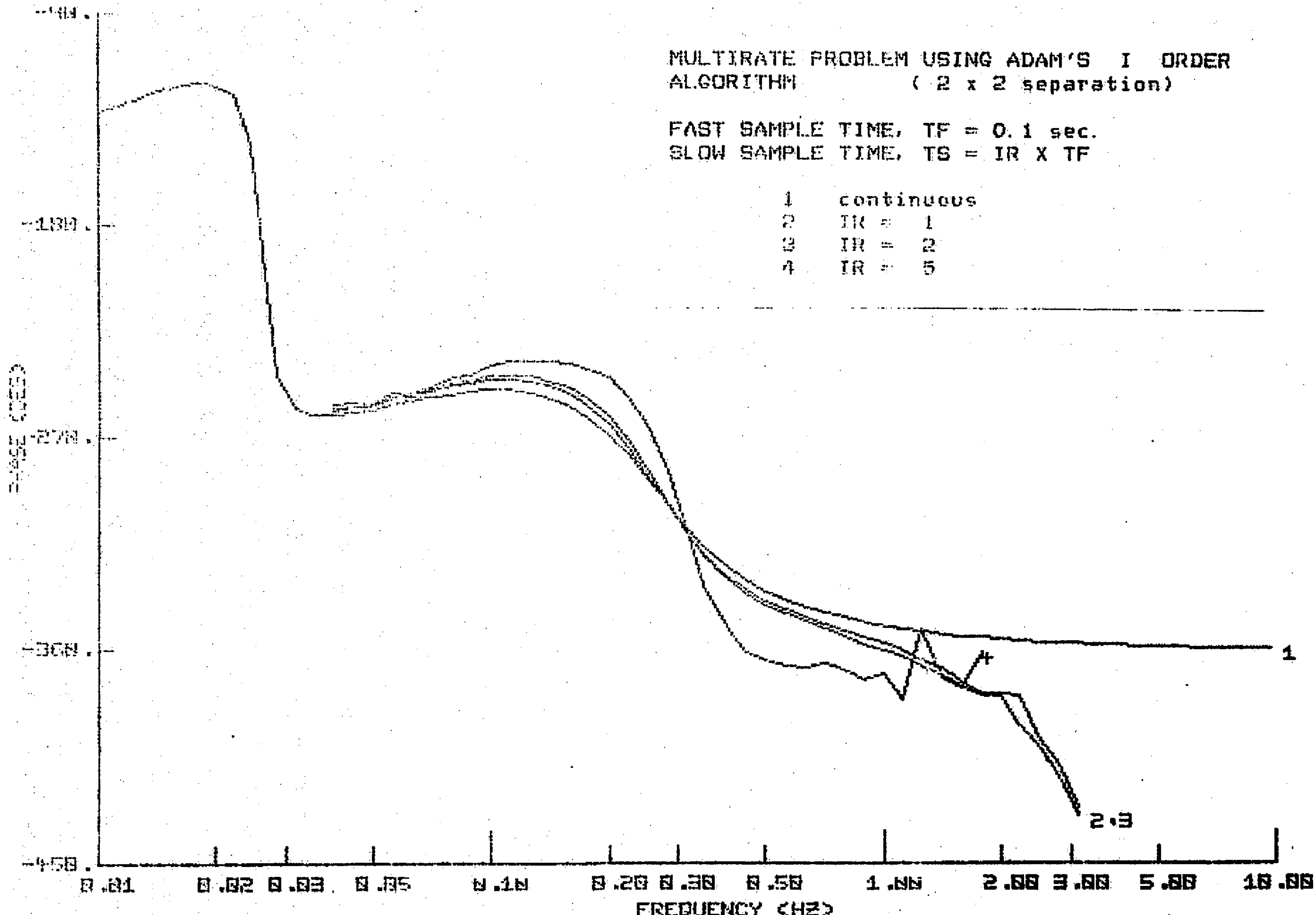
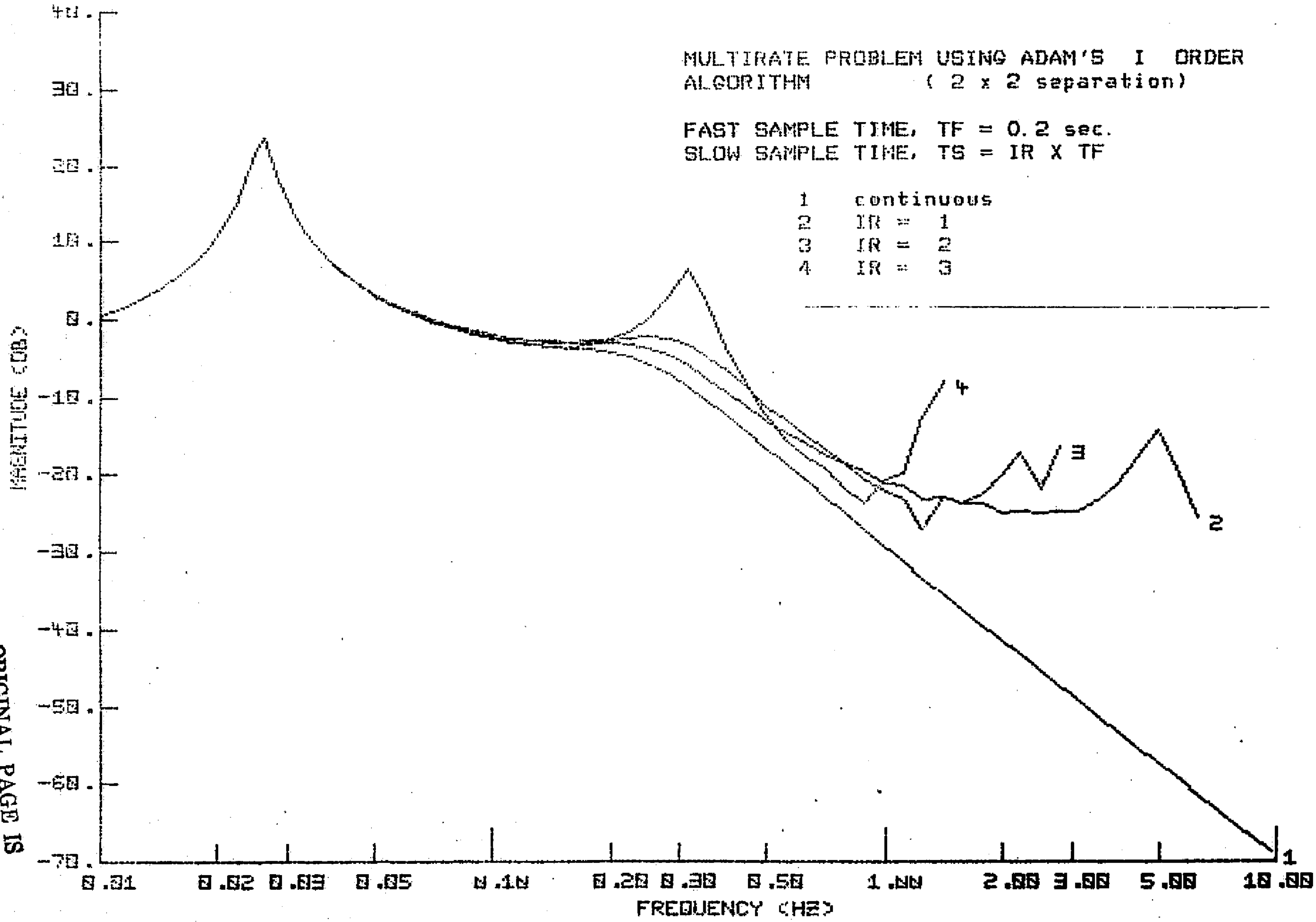

MULTIRATE PROBLEM USING ADAM'S I ORDER
ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.1 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 5
-90.
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)
1
2,3
4

FIGURE 17b



MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
MAGNITUDE (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
-70.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 18a

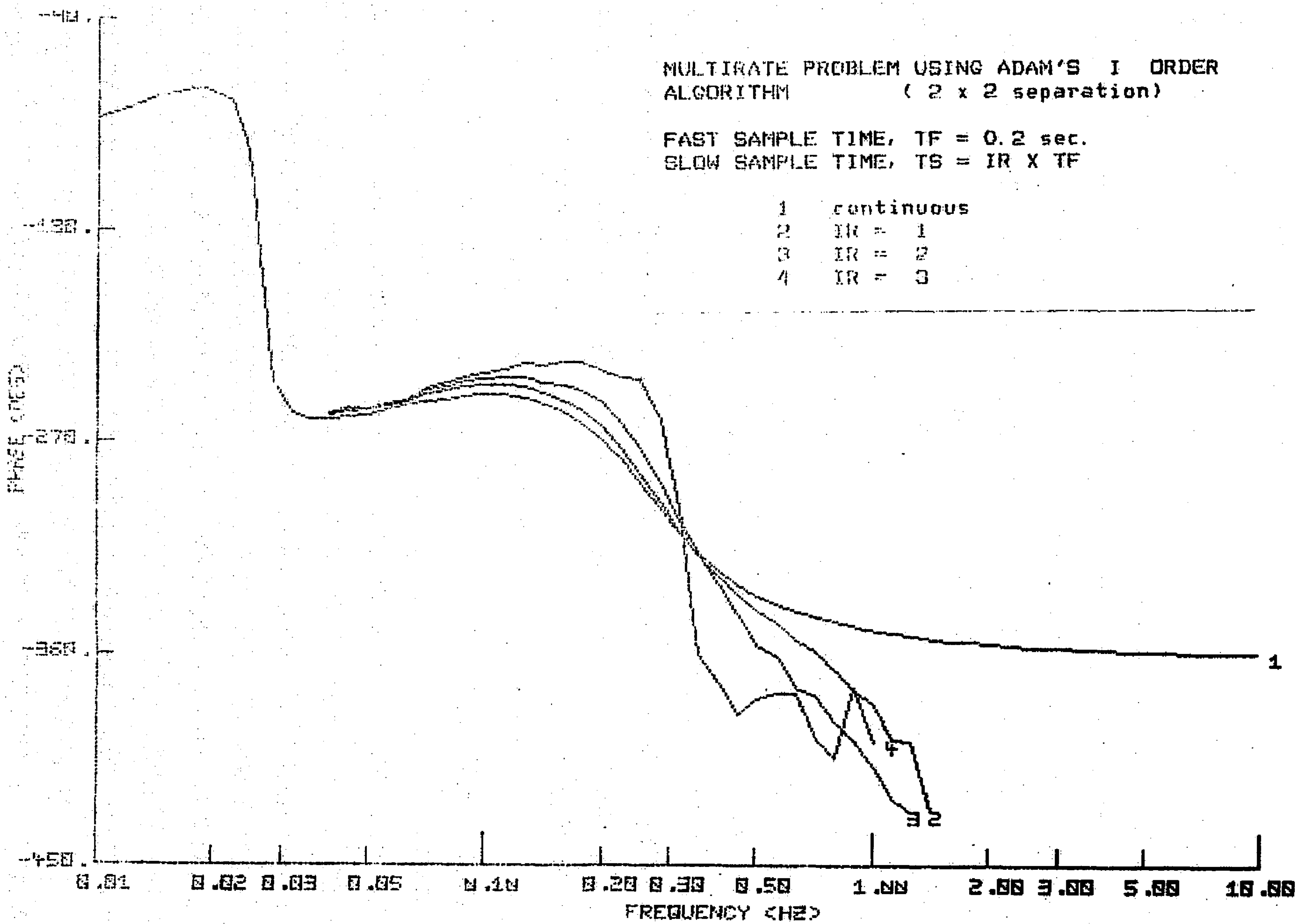

MULTIRATE PROBLEM USING ADAM'S I ORDER ALGORITHM (2 x 2 separation)
FAST SAMPLE TIME, TF = 0.2 sec.
SLOW SAMPLE TIME, TS = IR X TF
1 continuous
2 IR = 1
3 IR = 2
4 IR = 3
PHASE (DEG)
-180.
-270.
-360.
-450.
0.01
0.02
0.03
0.05
0.10
0.20
0.30
0.50
1.00
2.00
3.00
5.00
10.00
FREQUENCY (HZ)

FIGURE 18b

REFERENCES

1. Gear, C.W., "The Automatic Integration of Stiff Ordinary Differential Equations," Information Processing, 68, No. Holland Publ. Co., Amsterdam, 1969, pp. 187-193.

2. Chu, S.C. and M. Berman, "An Exponential Method for the Solution of Systems of Ordinary Differential Equations," Comm. of the ACM, Vol. 17, No. 12, Dec. 1974, pp. 699-702.

3. Nosrati, H., "A Modified Butcher Formula for Integration of Stiff Systems of Ordinary Differential Equations," Math. of Comp., Vol. 27, No. 122, April 1973, pp. 267-272.

4. Cash, J.R., "A Class of Implicit Runge-Kutta Methods for the Numerical Integration of Stiff Ordinary Differential Equations," J. of the Assoc. for Comp. Mach., Vol. 22, No. 4, Oct. 1975, pp. 504-511.

5. Enright, W.H., T.E. Hull and B. Lindberg, "Comparing Numerical Methods for Stiff Systems of O.D.E:s," BIT, 15, (1975), pp. 10-48.

6. Kalman, R.E. and J.E. Bertram, "A Unified Approach to the Theory of Sampling Systems," J. of the Franklin Inst., Vol. 267, No. 5, May 1959.

7. Friedland, B., "Sampled-Data Control Systems Containing Periodically Varying Members," Proc. of the 1st Internat'l. Congress of the Internat'l. Fed. of Auto. Contr., Auto. & Remote Contr., Moscow, 1960.

8. Boykin, W.H., and B.D. Frazier, "Multirate Sampled-Data Systems Analysis Via Vector Operators," IEEE Trans. Automatic Contr., Aug. 1975, pp. 548-551.

9. Franklin, G.F., and J.R. Ragazzini, Sampled-Data Control Systems, McGraw Hill, New York, 1958.

10. Whitbeck, R.F., "Analysis of Digitally Controlled Systems," Report on Contract F44-620-74-C-0061, Calspan Corp., 1974.

11. Teper, G.L., "Aircraft Stability and Control Data," STI Technical Report 176-1, NAS 2-4478, April 1969.

12. Gear, C.W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice Hall, Inc., Englewood Cliffs, N.J. 1971.

13. Cohen, A.M., Numerical Analysis, A Halsted Press Book, N.Y., 1973.

APPENDIX A

FREQUENCY RESPONSE EVALUATION OF EQUATION (4)

(EULER'S INTEGRATION)

The following computer code was used to analytically evaluate a G(z) by letting z take on values around the unit circle. The coefficients stored A1, A2,... and B1,B2 are for the continuous transfer function, equation (1) that represents the example described in this report. Lines 21 through 29 compute the coefficients required for an Euler integration of the continuous system.

```
         $WATFIV
         C GIVEN A TRANSFER FUNCTION IN THE Z-PLANE IN THE FOLLOWING FORM
         C
         C                   N            N-1             N-2
         C           B(N)  Z + B(N-1)  Z    + B(N-2)  Z     + ......, + B(1)  Z + B0
         C  H(Z) = K -----------------------------------------------------------------
         C                   N            N-1             N-2
         C           C(N)  Z + C(N-1)  Z    + C(N-2)  Z     + ......, + C(1)  Z + C0
         C
         C THIS PROGRAM CALCULATES, TABULATES AND PLOTS THE FREQUENCY RESPONSE
         C OF THE SYSTEM WHOSE TRANSFER FUNCTION IS GIVEN IN THE ABOVE FORM.
         C
  1              INTEGER ONE,ODP,TOPP,ZOOP
  2              REAL F(121),FAOD(20),GAIN(500),MAG(500),PHASE(500)
  3              REAL*8 SAMPT,T,B0,C0,B(30),C(30)
  4              REAL*8 A1,A2,A3,A4,B1,B2
  5              REAL*8 C1,C2,C3,C4,C5,C6,C7,C8,C9,D0,D1,D2,D3,D4,D5,D6,D7,D8,D9
  6              REAL*8 D10,D11,D12,G1,G2,G3,G4,G5,G6,G7,K1,K2,K3,K4,K5
  7              LOGICAL LOSOR(27), LORED(121)
  8              DATA A1,A2,A3,A4/1.711362200,2.661367200,0.097125260D,0.070369500/
  9              DATA B2,B1/0.032167500,0.555500/
 10              READ(5,11) ONE,ODP
 11       11     FORMAT (2I2)
 12              RUNS=0.
 13       30     READ (5,12) SAMPT,FMIN,FMAX
 14       12     FORMAT(F10.5,2(1X,F10.7))
 15              IF (SAMPT.EQ.0.) GO TO 900
 16              RUNS=RUNS+1.
 17              IF(RUNS.EQ.1) GO TO 50
 18              WRITE(6,55)
 19       55     FORMAT(1H1)
 20       50     T = SAMPT
         C CONS IS THE CONSTANT (IF ANY) IN THE TRANSFER FUNCTION.
         C THE COEFFICIENTS B0, B(I), C0, C(I) ARE THE COEFFICIENTS OF THE
         C POLYNOMIALS IN THE TRANSFER FUNCTION OF (THETA/DELTA(R) ) AS
         C CALCULATED BY THE BY USING THE EULER'S METHOD.   C'S ARE THE
         C DENOMINATOR COEFFS. AND B'S ARE THE NUMERATOR COEFFS.
 21              CONS = -1.338 * T ** 2
 22              B0 = B2*T**2-B1*T+1
 23              B(1) = B1*T-2
 24              B(2) = 1.
 25              C0 = A4*T**4-A3*T**3+A2*T**2-A1*T+1
 26              C(1) = A3*T**3-2*A2*T**2+3*A1*T-4
 27              C(2) = A2*T**2-3*A1*T+6
 28              C(3) = A1*T-4
 29              C(4) = 1.
 30              WRITE(6,14) SAMPT
 31       14     FORMAT(//,'   SAMPLE TIME = ',F12.5,//)
 32              WRITE(6,65) B0
 33       65     FORMAT('   B0    = ',E23.6//)
 34              DO 18 I=1,ONE,4
 35              WRITE(6,19) I,B(I),I+1,B(I+1),I+2,B(I+2),I+3,B(I+3)
 36       19     FORMAT (4(2X,'  B(',I2,') = ',E15.6)/)
 37       18     CONTINUE
 38              WRITE(6,75) C0
 39       75     FORMAT('   C0    = ',E23.6//)
 40              DO 15 I=1,ODP,4
 41              WRITE(6,16) I,C(I),I+1,C(I+1),I+2,C(I+2),I+3,C(I+3)
 42       16     FORMAT (4(2X,'  C(',I2,') = ',E15.6)/)
 43       15     CONTINUE
```


ORIGINAL PAGE IS OF POOR QUALITY

```
C
C
C     COMPUTE THE SET OF FREQUENCIES TO USE ( LOG SCALE ...)
  133 CONTINUE
      FMINLG = ALOG10(FMIN)
      LGFMIN = FMINLG
      IF ( FMIN .LT. 1. ) LGFMIN = LGFMIN - 1
      IDEC = LGFMIN
        DO 131 I = 1, 20
        LGNOR(I) = .FALSE.
  131   FNOR(I) = 10. ** ((I-1.)/20.)
      LGNOR( 1) = .TRUE.
      LGNOR( 7) = .TRUE.
      LGNOR(15) = .TRUE.
      NF = 1
      I = (FMINLG - LGFMIN)*20 + 1.5
  133 FACT = 10.**IDEC
  135 F(NF) = FACT*FNOR(I)
      LGRID(NF) = LGNOR(I)
      IF( F(NF) .GE. FMAX) GOTO 137
      I = I + 1
      NF = NF + 1
      IF ( I .LT. 21) GOTO 135
      I = 1
      IDEC = IDEC + 1
      GO TO 133
  137 NFREQ = NF
C
      F(69) = C.0241916
      F(70) = 0.0261010
      NFREQ = NFREQ + 2
C
      CALL POLWR (ONP,ODP,B,C,CONS,B0,C0)
      CALL UNICIR(ONP,ODP,B,C,CONS,SAMPT,NFREQ,F,B0,C0,GAIN,MAG,PHASE)
      I1=75
      RANGE = I1
      GO TO 30
  990 CONTINUE
      STOP
      END

      SUBROUTINE POLWRL(ONP,ODP,B,C,CONS,B0,C0)
C
      INTEGER ONP,ODP,TONP,TODP,TTONP,TTODP
      REAL*8 B0,C0,B(30),C(30)
      CHARACTER BUF1*2(20)/20*' '/,BUF2*15(20)/20*' '/,
     X          BUF3*2(20)/20*' '/,BUF4*15(20)/20*' '/
      LOGICAL*1 DUMMY(20)/20*'_'/,PLUS/'+'/
      WRITE (6,200)
  200 FORMAT (9X,'TRANSFER FUNCTION  : '//)
      IF (ONP.NE.0) GO TO 205
      WRITE (6,201) CONS
  201 FORMAT(30X,E12.5/' H(Z) = ',' ',30('_'))
      GO TO 215
  205 TONP = ONP
      TTONP = ONP + 1
      IF (ONP.EQ.1) GO TO 212
      DO 210 I = 1, ONP
      WRITE (BUF1(I),221) TONP
  221 FORMAT(I2)
      IF (B(TONP).LT.0.) GO TO 250
      WRITE(BUF2(I),231) PLUS,B(TONP)
  231 FORMAT (A1,E12.5,'Z',1X)
      GO TO 252
  250 WRITE (BUF2(I),230) B(TONP)
  230 FORMAT (1X,E12.5,'Z',1X)
  252 TONP = TONP - 1
  210 CONTINUE
  212 WRITE (BUF2(TTONP),271) B0
  271 FORMAT (F15.9)
C
      KN1 = 1
      KN2 = 6
      DO 225 J = 1, ONP, 6
      WRITE(6,255) DUMMY, (BUF1(KN), KN = KN1,KN2)
  255 FORMAT (20A1,6(14X,A2))
      WRITE(6,270)        (BUF2(KN), KN = KN1,KN2)
  270 FORMAT(20X,6A16/)
      KN1 = KN1 + 6
  225 KN2 = KN2 + 6
      WRITE (6,290) CONS
  290 FORMAT(' H(Z) =',E12.5,'*',130('_'))
  215 TODP = ODP
      TTODP = ODP + 1
      IF(ODP.EQ.1) GO TO 245
      DO 240 I = 1, ODP
      WRITE (BUF3(I),245) TODP
  245 FORMAT(I2)
      IF (C(TODP).LT.0.) GO TO 207
      WRITE(BUF4(I),232) PLUS,C(TODP)
  232 FORMAT (A1,E12.5,'*Z',1X)
      GO TO 208
  207 WRITE (BUF4(I),230) C(TODP)
  208 TODP = TODP - 1
  240 CONTINUE
  246 WRITE (BUF4(TTODP),271) C0
      WRITE(6,254)
  254 FORMAT (25X)
C
```

```
      KN1 = 1
      KN2 = 6
      DO 235 I = 1, JDP, 6
      WRITE(6,255) DUMMY, (DUP3(KN), KN = KN1,KN2)
      WRITE(6,270)        (DUP4(KN), KN = KN1,KN2)
      KN1 = KN1 + 6
  235 KN2 = KN2 + 6
      RETURN
      END

      SUBROUTINE UNICIR(JNP,JDP,B,C,CONS,SAMPT,NOFRQ,F,B0,C0,
     X                  GAIN,MAG,PHASE)
C
C
      INTEGER JNP,JDP,NOFRQ
      REAL GAIN(500),MAG(500),PHASE(500),F(121)
      REAL*8 B0,C0,SAMPT,CONS,B(30),C(30)
      REAL*8 XT,YT,DLOG10,DATAN2,DBLE
      COMPLEX ZCRES
      COMPLEX*16 DCMPLX,CNUM,CDEN,CRES,Z,CN,CD
      PI = 3.1415926

      T2PI = 2*PI*SAMPT
      DEGPRD = 180. / PI
      DO 180 J = 1, NOFRQ
  180 PHASE(J) = 0.
      IF(CONS.GE.0.) GO TO 185
      DO 190 J = 1, NOFRQ
  190 PHASE(J) = -180.
  185 WRITE(6,125)
  125 FORMAT(///,21X,'FREQUENCY(HZ)',14X,'GAIN',15X,'MAGNITUDE(DB)',
     X                                     11X,'PHASE(DEG)',//)
      DO 100 NF = 1, NOFRQ
      CNUM = DCMPLX (B0,0.0D0)
      CDEN = DCMPLX (C0,0.0D0)
      OMEGAT = T2PI * F(NF)
      XT = DBLE(COS(OMEGAT))
      YT = DBLE(SIN(OMEGAT))
      Z = DCMPLX(XT,YT)
      IF (JNP.EQ.0) GO TO 105
      CN = Z
         DO 110 I = 1,JNP
         CNUM = CNUM + B(I) * CN
         CN = CN * Z
  110    CONTINUE
  105 IF(JDP.EQ.0) GO TO 115
      CD = Z
         DO 120 I = 1,JDP
         CDEN = CDEN + C(I) * CD
         CD = CD * Z
  120    CONTINUE
  115 ZCRES = CNUM / CDEN
      CRES = CONS * (CNUM/CDEN)
      GAIN(NF) = CDABS(CRES)
      MAG(NF)=20.*DLOG10(CDABS(CRES))
      PHASE(NF)=PHASE(NF)+DEGPRD*ATAN2(AIMAG(ZCRES),REAL(ZCRES))
      WRITE(6,150) NF,F(NF),GAIN(NF),MAG(NF),PHASE(NF)
  150 FORMAT(3X,I2,9X,F12.5,3(12X,F12.7))
  100 CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

FREQUENCY RESPONSE EVALUATION OF EQUATION(6)

(1st ORDER ADAMS)

The computer code in Appendix A was modified by replacing lines 21 through 29 with the following coefficient evaluations.

```
21        CONS = -1.333
22        C1 = 81*T**4
23        C2 = -108*T**4
24        C3 = 54*T**4
25        C4 = -12*T**4
26        C5 = T**4
27        D1 = 16.
28        D2 = 24*A1*T-64
29        D3 = 36*A2*T**2-80*A1*T+96
30        D4 = 54*A3*T**3-96*A2*T**2+96*A1*T-64
31        D5 = 81*A4*T**4-108*A3*T**3+88*A2*T**2-48*A1*T+16
32        D6 = -108*A4*T**4+72*A3*T**3-32*A2*T**2+6*A1*T
33        D7 = 54*A4*T**4-20*A3*T**3+4*A2*T**2
34        D8 = -12*A4*T**4+2*A3*T**3
35        D9 = A4*T**4
36        G1 = 4.
37        G2 = 6*B1*T-8
38        G3 = 9*B2*T**2-3*B1*T+4
39        G4 = -6*B2*T**2+2*B1*T
40        G5 = B2*T**2
41        K1 = 9*T**2
42        K2 = -6*T**2
43        K3 = T**2
44        B0 = C5*G5
45        B(1) = C4*G5+C5*G4
46        B(2) = C3*G5+C4*G4+C5*G3
47        B(3) = C2*G5+C3*G4+C4*G3+C5*G2
48        B(4) = C1*G5+C2*G4+C3*G3+C4*G2+C5*G1
49        B(5) = C1*G4+C2*G3+C3*G2+C4*G1
50        B(6) = C1*G3+C2*G2+C3*G1
51        B(7) = C1*G2+C2*G1
52        B(8) = C1*G1
53        C0 = D9*K3
54        C(1) = D8*K3+D9*K2
55        C(2) = D7*K3+D8*K2+D9*K1
56        C(3) = D6*K3+D7*K2+D8*K1
57        C(4) = D5*K3+D5*K2+D7*K1
58        C(5) = D4*K3+D5*K2+D6*K1
59        C(6) = D3*K3+D4*K2+D5*K1
60        C(7) = D2*K3+D3*K2+D4*K1
61        C(8) = D1*K3+D2*K2+D3*K1
62        C(9) = D1*K2+D2*K1
63        C(10) = D1*K1
```

APPENDIX C

FREQUENCY RESPONSE EVALUATION OF EQUATION (8)

(2nd ORDER ADAMS)

The computer code of Appendix A was modified by replacing lines 21 through 29 with the following coefficient evaluations.

```
21        CONS = -1.338
22        C1 = 275941*T**4
23        C2 = -776688*T**4
24        C3 = 1155864*T**4
25        C4 = -884672*T**4
26        C5 = 498156*T**4
27        C6 = -152320*T**4
28        C7 = 49920*T**4
29        C8 = -8000*T**4
30        C9 = 625*T**4
31        D0 = 20736.
32        D1 = 32744*A1*T-82944
33        D2 = 76176*A2*T**2-146880*A1*T+124416
34        D3 = 146004*A3*T**3-258335*A2*T**2+210615*A1*T-82944
35        D4 = 273841*A4*T**4-450708*A3*T**3+353123*A2*T**2-116658*A1*T
         X                                            +20736
36        D5 = -778688*A4*T**4+511092*A3*T**3-266992*A2*T**2+53568*A1*T
37        D6 = 1055864*A4*T**4-120820*A3*T**3+115654*A2*T**2-5542*A1*T
38        D7 = -884672*A4*T**4+246412*A3*T**3-33240*A2*T**2
39        D8 = 498156*A4*T**4-81160*A3*T**3+3500*A2*T**2
40        D9 = -152320*A4*T**4+15300*A3*T**3
41        D10 = 49920*A4*T**4-1500*A3*T**3
42        D11 = -8000*A4*T**4
43        D12 = 625*A4*T**4
44        G1 = 144
45        G2 = 276*B1*T-288
46        G3 = 529*B2*T**2-468*B1*T+144
47        G4 = -736*B2*T**2+232*B1*T
48        G5 = 486*B2*T**2-60*B1*T
49        G6 = -160*B2*T**2
50        G7 = 25*B2*T**2
51        K1 = 579*T**2
52        K2 = -736*T**2
53        K3 = 486*T**2
54        K4 = -150*T**2
55        K5 = 25*T**2
56        B0 = C9*G7
57        B(1)=C8*G7+C9*G6
58        B(2)=C7*G7+C8*G6+C9*G5
59        B(3)=C6*G7+C7*G6+C8*G5+C9*G4
60        B(4)=C5*G7+C6*G6+C7*G5+C8*G4+C9*G3
61        B(5)=C4*G7+C5*G6+C6*G5+C7*G4+C8*G3+C9*G2
62        B(6)=C3*G7+C4*G6+C5*G5+C6*G4+C7*G3+C8*G2+C9*G1
63        B(7)=C2*G7+C3*G6+C4*G5+C5*G4+C6*G3+C7*G2+C8*G1
64        B(8)=C1*G7+C2*G6+C3*G5+C4*G4+C5*G3+C6*G2+C7*G1
65        B(9)=C1*G6+C2*G5+C3*G4+C4*G3+C5*G2+C6*G1
66        B(10)=C1*G5+C2*G4+C3*G3+C4*G2+C5*G1
67        B(11)=C1*G4+C2*G3+C3*G2+C4*G1
68        B(12)=C1*G3+C2*G2+C3*G1
69        B(13)=C1*G2+C2*G1
70        B(14)=C1*G1
71        C0 = D12*K5
72        C(1)=D11*K5+D12*K4
73        C(2)=D10*K5+D11*K4+D12*K3
74        C(3)=D9*K5+D10*K4+D11*K3+D12*K2
75        C(4)=D8*K5+D9*K4+D10*K3+D11*K2+D12*K1
76        C(5)=D7*K5+D8*K4+D9*K3+D10*K2+D11*K1
77        C(6)=D6*K5+D7*K4+D8*K3+D9*K2+D10*K1
78        C(7)=D5*K5+D6*K4+D7*K3+D8*K2+D9*K1
79        C(8)=D4*K5+D5*K4+D6*K3+D7*K2+D8*K1
80        C(9)=D3*K5+D4*K4+D5*K3+D6*K2+D7*K1
81        C(10)=D2*K5+D3*K4+D4*K3+D5*K2+D6*K1
82        C(11)=D1*K5+D2*K4+D3*K3+D4*K2+D5*K1
83        C(12)=D0*K5+D1*K4+D2*K3+D3*K2+D4*K1
84        C(13)=D0*K4+D1*K3+D2*K2+D3*K1
85        C(14)=D0*K3+D1*K2+D2*K1
86        C(15)=D0*K2+D1*K1
87        C(16)=D0*K1
```

APPENDIX D

SIMULATION FOR FREQUENCY RESPONSE EVALUATIONS IN THE MULTI-RATE CASES

The following code performs the calculations using Euler's integration. Note that lines 107 through 112 are shown twice, once for the 1 × 3 separation and once for the 2 × 2 separation.

```
         $WATFIV
         C
         C THIS IS THE DIGITAL SIMULATION OF THE LONGITUDINAL EQUATIONS OF
         C MOTION OF AN AIRCRAFT.
 1             REAL*8 F(121),FNOR(20),GAIN(150),MAG(150),PHASE(150)
 2             REAL*8 T,TF,TS
 3             REAL FMIN,FMAX,ALOG10
 4             LOGICAL LGNOR(20),LGRID(121)
         C
         C T IS THE FAST SAMPLING TIME AND IR IS THE INTEGER RATIO BETWEEN THE
         C FAST AND SLOW SAMPLING TIMES. FMIN AND FMAX ARE THE FREQUENCY LIMITS
         C FOR WHICH THE RESPONSE IS TO BE CALCULATED.
         C
 5        20   READ(5,50) T,IR,FMIN,FMAX
 6        50   FORMAT(F10.5,I2,2(2X,F10.0))
 7             IF (T.EQ.0.) GO TO 900
 8             WRITE (6,45)
 9        45   FORMAT (1H1)
10             WRITE (6,55) T,IR
11        55   FORMAT(' ','T=',F5.2,3X,'IR=',I2)
12             DATA AMAX,AMIN,TI /0.,0.,0./
13             TF = T
14             WF = 1 / TF
15             TS = IR * TF
16             WS = 1 / TS
17             WRITE (6,60) TF,WF,TS,WS
18        60   FORMAT(//,' ','FAST SAMPLING TIME = ',F7.2,3X,'SECS',5X,'( ',F7.2,
              $                                        'HZ )'//
              $        ,' ','SLOW SAMPLING TIME = ',F7.2,3X,'SECS',5X,'( ',F7.2,
              $                                        'HZ )'//)
         C
19             PI = 3.1415926
20             DO 100 I = 1, 200
21             DELTAE = SIN(2*PI*1.*TI)
22             TI = TI + TF
23             IF(DELTAE.LE.AMAX) GO TO 75
24             AMAX=DELTAE
25        75   IF(DELTAE.GT.AMIN) GO TO 100
26             AMIN=DELTAE
27       100   CONTINUE
28             AMPI=AMAX-AMIN
29             WRITE(6,19)AMPI
30        19   FORMAT(' INPUT AMPLITUDE = ',F10.4//)
         C
         C     COMPUTE THE SET OF FREQUENCIES TO USE ( LOG SCALE ...)
31       130 CONTINUE
32             FMINLG = ALOG10(FMIN)
33             LGFMIN = FMINLG
34             IF ( FMIN .LT. 1. ) LGFMIN = LGFMIN - 1
35             IDEC = LGFMIN
36                DO 131 I = 1, 20
37                LGNOR(I) = .FALSE.
38       131      FNOR(I) = 10. ** ((I-1.)/20.)
39             LGNOR( 1) = .TRUE.
40             LGNOR( 7) = .TRUE.
41             LGNOR(15) = .TRUE.
42             NF = 1
43             I = (FMINLG - LGFMIN)*20 + 1.5
44       133 FACT = 10.**IDEC
45       135 F(NF) = FACT*FNOR(I)
46             LGRID(NF) = LGNOR(I)
47             IF( F(NF) .GE. FMAX) GOTO 137
48             I = I + 1
49             NF = NF + 1
50             IF ( I .LT. 21) GOTO 135
51             I = 1
52             IDEC = IDEC + 1
53             GO TO 133
54       137 NFERQ = NF
         C
55             CALL SIMLTN (TF,TS,IR,NFERQ,F,AMPI,GAIN,MAG,PHASE)
56             GO TO 20
57       900   CONTINUE
58             STOP
59             END

60             SUBROUTINE SIMLTN (TF,TS,IR,NFERQ,F,AMPI,GAIN,MAG,PHASE)
         C
         C THIS SUBROUTINE IS THE SIMULATION OF THE LONGITUDINAL EQUATIONS OF
         C AIRCRAFT.  IT CALCULATES AND PRINTS THE RESPONSE (THETA) TO INPUT
         C SINE WAVE OF DIFFERENT FREQUENCIES.
         C
61             REAL*8 F(121),FNOR(20),GAIN(150),MAG(150),PHASE(150)
62             REAL*8 TS,TF,TI,TMAX,TMIN,THMIN
63             REAL*8 P7,PRC,PDT,MNU,W,Q,THETA,U,PRIDOT,PRWDOT
64             REAL*8 CONS,C,D,Z,PHMN,PHASE1,CF,PH3
65             REAL*8 DSQRT,DTX2,DATAN,DSIN,CLOG10,DARSIN
66             REAL SET1(7000),SET2(7000)
67             INTEGER IT,INO,NFERQ
68             PI = 3.1415926000
69             CONS = -1.33000
70             WRITE(6,103)
71       103   FORMAT(' ',2X,'I',13X,'T',17X,'FREQ',23X,'GAIN',19X,'MAGNITUDE',
              $                              14X,'PHASE'//)
         C
```

```
      DO 200 I = 1, NAFRQ
      W = 0.D0
      Q = 0.D0
      THETA = 0.D0
      U = 0.D0
      PRWDOT = 0.D0
      PRUDOT = 0.D0
      NI = 1
      N = 300 + (3.2/(F(I)*TF))
      TMAX = 0.D0
      TMIN = 0.D0
      TI = 0.D0
C
      C = (-0.6510761D-02+0.2421377D0*(2*PI*F(I))**2) / (0.8326793D-03
     $     -0.6003436D-01*(2*PI*F(I))**2  + 1.066553D0*(2*PI*F(I))**4)
      D = (0.3024675D-03-0.4443344D-02*(2*PI*F(I))**2) / (-0.8326793D-03
     $     +0.6003436D-01*(2*PI*F(I))**2  - 1.066653D0*(2*PI*F(I))**4)
C
      Z = D / C
      PHASE1  =   DATAN (0.1631995D0/(Z-0.9909810D0))
      CF = CONS*C*2*PI*F(I)*DSQRT(((Z-0.9990810D-02)**2+(0.1631995D0)**2)
     $                     / (0.1631995D0)**2)
           DO 210 J = 1, N
           PRW = W
           PRQ = Q
           PRT = THETA
           PRU = U
           PHUG = CF * DEXP (-0.9908 10D-02*TI) * DSIN (0.1631995D0 * TI
     $                                            + PHASE1)
           THMPH = THETA - PHUG
       SET1(J) = TI
       SET2(J) = THMPH
       IND = J
           IF (J.LT.300) GO TO 230
           IF (THMPH.LT.TMAX) GO TO 220
           TMAX = THMPH
 220       IF (THMPH.GT.TMIN) GO TO 230
           TMIN = THMPH
C
C
 230       Q = PRQ + TF * (-0.770-05*PRU-0.00870D0*PRW-0.7920D0*PRQ
     $                          -1.750D0*DSIN(2.0D0*PI*F(I)*TI))
           W = PRW + TF * (-0.2510D0*PRU-0.6290D0*PRW+243.500*PRQ
     $                          -10.200*DSIN(2.D0*PI*F(I)*TI))
           THETA = PRT + TF * PRQ
           IF (J.NE.IR*NI) GO TO 255
           U = PRU + TS * PRUDOT
           PRUDOT = -0.029100*U + 0.062900*W - 32.200*THETA
           NI = NI + 1
 255       TI = TI + TF
 210       CONTINUE
C
C
       GAIN(I) = ((TMAX-TMIN) / 2.0D0)
       MAG(I) = 20 * DLOG10(GAIN(I))
 301   IF (SET2(IND).LT.0.) GO TO 302
       IF (SET2(IND-1).LT.0.) GO TO 303
 302   IND = IND - 1
       GO TO 301
 303   M3 = F(I) * SET1(IND)
       PHI = 2 * PI * (F(I) * SET1(IND) - M3)
       PHASE(I) = (DARSIN(SET2(IND)/GAIN(I))-PHI) * 180.D0 / PI
       WRITE(6,280)  I,N,F(I),GAIN(I),MAG(I),PHASE(I)
 280   FORMAT (' ',I3,10X,I6,10X,F12.5,3(10X,F15.7))
 200   CONTINUE
       RETURN
       END
```

1X3

10ATA

```
           IF (J.NE.IR*NI) GO TO 255
           W = PRW + TS * PRWDOT
           U = PRU + TS * PRUDOT
           PRWDOT = -0.2510D0*U -0.6290D0*W + 243.5D0*Q
     $                 -10.2D0*DSIN(2.D0*PI*F(I)*(TI+TF))
           PRUDOT = -0.029100*U + 0.062900*W - 32.200*THETA
           NI = NI + 1
 255       TI = TI + TF
```

2X2

The modifications to the previous code so as to use first order Adams integration are shown below.

```
C
      DO 200 I = 1, NAFRQ
      CONS = -1.31469
      NI = 1
C
      C = (-0.6510761D-02+0.2421372D0*(2*PI*F(I))**2) / (0.4326793D-03
     $     -0.6004346D-01*(2*PI*F(I))**2  + 1.07665349D0*(2*PI*F(I))**4)
      D = (0.3024675D-03-0.4453340D-02*(2*PI*F(I))**2) / (-0.4326793D-03
     $     +0.6004346D-01*(2*PI*F(I))**2  - 1.07665340D0*(2*PI*F(I))**4)
C
      Z = D / C
      PHASE1  =   DATAN (0.1631995D0/(Z-0.0099081D0))
      CF =CONS*C*2*PI*F(I) * DSQRT(((Z-0.0099081D0)**2+(0.1631995D0)**2)
     $                          / (0.1631995D0)**2)
C
C
C     SINCE ADAMS FORMULA IS NOT SELF-STARTING, THE VALUE OF THETA AT
C     THE FIRST INTERVAL IS FOUND BY EULERS ALGORITHM USING THE FOLLOW-
C     ING SUBROUTINE.
C
      CALL START (F,IR,I,NI,TI,TF,TS,WDOT,QDOT,UDOT,W1,Q1,TH1,U1,CF,
     $                         PHASE1)
C
C     THE FOLLOWING SUBROUTINE FINDS THE DERIVATIVES AT EACH INTERVAL
C     AND FINDS THE VALUE OF THETA,PHUGOID.  THEN SUBTRACTS THE VALUE
C     OF THE PHUGOID FROM THETA TO ELIMINATE THE PHUGOID COMPONENT
C     AND PRINTS OUT THE VALUES.
C
      CALL DERIV (F,IR,NAFRQ,NI,TI,TF,TS,WDOT,QDOT,UDOT,W1,Q1,TH1,
     $                         U1,I,CF,PHASE1)
C
 200  CONTINUE
      GO TO 30
 900  CONTINUE
      STOP
      END
```

```
      SUBROUTINE START (F,IR,I,N1,T1,TF,TS,WODOT,QODOT,UODOT,W1,Q1,
     $                  TH1,U1,CF,PHASE1)
      REAL*8 F(121),PI,TF,TS,WODOT,QODOT,UODOT,CF,PHASE1
      REAL*8 W0,Q0,THETA0,U0,W1,Q1,TH1,U1,DSIN,DEXP
      REAL T1
      INTEGER IR,I,N1
      PI = 3.141592600
      Q0 = 0.D0
      W0 = 0.D0
      THETA0 = 0.D0
      U0 = 0.D0
      U1 = U0
      UODOT = 0.D0

      T1 = 0.D0
         PHUG = CF * DEXP (-0.990810-02*T1) * DSIN (0.16319950D
     $                   * T1 + PHASE1)
         THMPH = THETA0 - PHUG
C
         WODOT = -0.25100*U0 - 0.52800*W0 + 243.5D0*Q0 - 10.20D0*
     $                        DSIN(2.D0*PI*F(I)*T1)
         QODOT = -0.77D-05*U0 - 0.87D-02*W0 - 0.792D0*Q0 - 1.350D0*
     $                        DSIN(2.D0*PI*F(I)*T1)
C
      W1 = W0 + TF * WODOT
      Q1 = Q0 + TF * QODOT
      TH1 = THETA0 + TF * Q0
      IF (IR.NE.1) GO TO 205
         UODOT = -0.291D-01*U0 + 0.629D-01*W0 - 32.20D0*THETA0
      U1 = U0 + TS * UODOT
      N1 = N1 + 1
  205 T1 = T1 + TF
         PHUG = CF * DEXP (-0.990810-02*T1) * DSIN (0.16319950D
     $                   * T1 + PHASE1)
         THMPH = TH1 - PHUG
      RETURN
      END

      SUBROUTINE DERIV (F,IR,NBFRQ,N1,T1,TF,TS,WODOT,QODOT,UODOT,W1,
     $                  Q1,TH1,U1,I,CF,PHASE1)
      REAL*8 F(121),GAIN(150),MAG(150),PI,TF,TS,CF,PHASE1
      REAL*8 WODOT,QODOT,UODOT,Q0,W1,Q1,TH1,U1,W2,Q2,TH2,U2
      REAL*8 DSIN,DLOG10,DEXP
      REAL T1
      INTEGER I,N1,IR,NBFRQ
      PI = 3.141592600
      TMAX = 0.D0
      TMIN = 0.D0
      Q0 = 0.D0
      U2 = 0.D0
      N = 300 + (3.2/(F(I)*TF))
         DO 210 J = 2, N
C
         W1DOT = -0.2310D*U1 - 0.63800*W1 + 243.50D*Q1 - 10.20D*
     $                        DSIN(2.D0*PI*F(I)*T1)
         Q1DOT = -0.77D-05*U1 - 0.87D-02*W1 - 0.792D0*Q1 - 1.350D*
     $                        DSIN(2.D0*PI*F(I)*T1)
         U1DOT = -0.291D-01*U1 + 0.524D-01*W1 - 32.20D*TH1
C
         W2 = W1 + TF * (3.D0 * W1DOT - WODOT) / 2.D0
         Q2 = Q1 + TF * (3.D0 * Q1DOT - QODOT) / 2.D0
         TH2 = TH1 + TF * (3.D0 * Q1 - Q0) / 2.D0
C
*        IF (J.NE.IR+1) GO TO 225
         U2 = U1 + TS * (3.D0 * U1DOT - UODOT) / 2.D0            1x3
         N1 = N1 + 1
  225    T1 = T1 + TF
C
         PHUG = CF * DEXP (-0.990810-02*T1) * DSIN (0.16319950D
     $                   * T1 + PHASE1)
         THMPH = TH2 - PHUG
         IF (J.LT.300) GO TO 230
         IF (THMPH.LT.TMAX) GO TO 220

         TMAX = THMPH
  220    IF (THMPH.GT.TMIN) GO TO 230
         TMIN = THMPH
C
  230 CONTINUE
C
         WODOT = W1DOT
         QODOT = Q1DOT
         Q0 = Q1
         UODOT = U1DOT
         W1 = W2
         Q1 = Q2
         U1 = U2
         TH1 = TH2
  210    CONTINUE
      GAIN(I) = ((TMAX-TMIN) / 2.D00)
      MAG(I) = 20 * DLOG10(GAIN(I))
      WRITE(6,280)  I,N,F(I),GAIN(I),MAG(I)
  280 FORMAT (' ',I3,10X,I6,10X,F12.6,2(10X,F15.7))
      RETURN
      END

$DATA

*     IF (IR.NE.1) GO TO 235
         W1DOT = -0.2510D*U1 - 0.62800*W1 + 243.5D0*Q1 - 10.20D0*
     $                        DSIN(2.D0*PI*F(I)*T1)                  2x2
         U1DOT = -0.291D-01*U1 + 0.629D-01*W1 - 32.20D*TH1
  235 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY