(NASA-CR-161239) IDENTIFICATION OF LIMIT CYCLES IN MULTI-NONLINEARITY, MULTIPLE PATH SYSTEMS Final Report, 1 Jan. 1978 - 31 May 1979 (Mississippi State Univ., Mississippi State.) 108 p HC A06/MF A01 CSCL 09B G3/63

N79-25805

Unclas
23465

ENGINEERING & INDUSTRIAL RESEARCH STATION

Electrical Engineering/Mississippi State University

Identification of Limit Cycles in Multi-Nonlinearity, Multiple Path Systems

by

Jerrel R. Mitchell, Ph.D.
Ottis L. Barron

MSSU-EIRS-EE-79-4

COLLEGE OF ENGINEERING ADMINISTRATION

For additional copies or information
address correspondence to

ENGINEERING AND INDUSTRIAL RESEARCH STATION
DRAWER DE
MISSISSIPPI STATE UNIVERSITY
MISSISSIPPI STATE, MISSISSIPPI 39762

TELEPHONE (601) 325-2266

MISSISSIPPI STATE UNIVERSITY

DEPARTMENT OF ELECTRICAL ENGINEERING

Final Report
for Period Covering
January 1, 1978 to May 31, 1979

May 29, 1979

NAS8-32810

by
Jerrel R. Mitchell, Ph.D.
and
Ottis L. Barron

Prepared for:

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

ABSTRACT

Techniques for the identification of limit cycles in nonlinear control systems rely heavily upon describing function analysis. Most methods are restricted to single nonlinearity systems and consider only the fundamental component of the Fourier series representation of the nonlinearity output. No previously available technique considers multiple harmonic components in systems containing several nonlinearities and multiple forward paths.

This report presents a method of analysis which identifies limit cycles in autonomous systems with multiple nonlinearities in multiple forward paths. The nonlinearities may be any time- and frequency-invariant single input, single output type which can be adequately represented in piecewise-linear form. In addition, several harmonic components may be retained in the procedure.

Included in the report is the FORTRAN code for implementing the Harmonic Balance Algorithm (HBA) developed in the research. Using this code, limit cycles are identified in multi-path, multi-nonlinearity systems while retaining the effects of several harmonic components.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure | Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

The science of control system analysis has experienced rapid development in the years since WWII. Many techniques for determining the stability of linear feedback control systems* have evolved, some of which give a simple indication where others give additional information such as the margin of stability. These methods are described in detail in many texts, such as Truxal [1], Dorf [2], and Kuo [3], and are summarized here.

In a system such as that shown in Figure 1.1, a transfer function is the ratio of the output and input of a block. The transfer function of the upper block is thus $G(s) = C(s)/E(s)$ while that of lower block is $H(s) = V(s)/C(s)$. By algebraic manipulation the input-output relationship of the system is obtained as

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad . \tag{1-1}$$

The denominator of the above equation when set equal to zero is referred to as the characteristic equation; it determines the mathematical characteristic of the transient response of the system.

* A linear system is one for which the superposition principle holds. Superposition implies the properties of additivity and homogeneity with respect to both input and initial conditions. Additivity is illustrated in the following manner: if input r_1 produces output y_1 , and input r_2 produces output y_2 , then input $r_1 + r_2$ will produce output $y_1 + y_2$. Homogeneity is illustrated thusly: if input r_1 produces output y_1 , then input kr_1 will produce output ky_1 . A linear system is stable if its output tends to a finite value for any finite input.

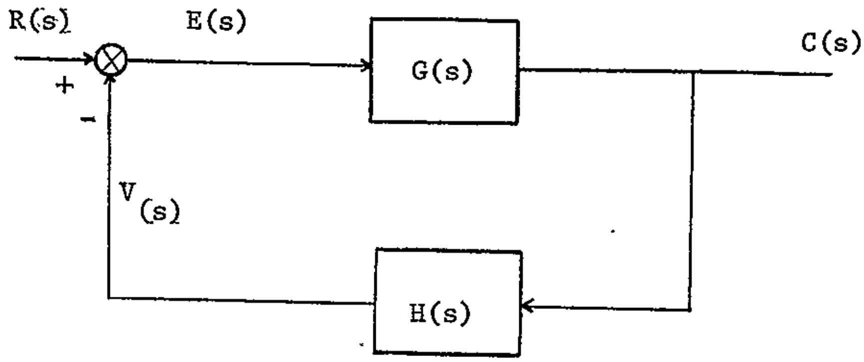


Figure 1.1. Feedback control system

A relatively simple mechanistic procedure for determining stability from the unfactored polynomial in s, which forms the characteristic equation, is the Routh test. This test, which originated about 1880, can be used to determine the number of the roots with positive real parts. The existence of roots with positive real parts indicates an unstable system.

In addition several graphical techniques are widely used in testing for stability. These, in general, provide information regarding the relative stability of the system. Included are the Nyquist Criterion, the Bode Diagram, and the Root Locus Plot.

There are cases where the transfer function G (or H) cannot be readily determined as a polynomial in s. When the polynomial is not available, the Routh test cannot be applied. The Nyquist stability criterion can be used in these cases if the frequency response characteristics of $G(s)H(s)$ can be determined.

Note that for a linear system to be stable, all zeroes of $1 + G(s)H(s)$ must lie in the left half of the s-plane. The Nyquist criterion determines if this is the case by examining a plot of

$G(s)H(s)$ as s is allowed to vary along a closed contour in the s-plane. This contour may be, for example, the imaginary axis of the s-plane together with a large semicircle enclosing the right half plane. An indication of stability can be obtained by examining the number of resulting clockwise encirclements of the point (-1,0) in the $G(j\omega)H(j\omega)$ plane. The number of encirclements is given by

$$N = Z - P \tag{1-2}$$

where N = the algebraic sum of encirclements of the (-1,0) point where clockwise encirclements are assumed to be positive.

Z = the number of zeroes of $1 + GH$ in the right half of the s-plane

P = the number of poles of GH in the right half of the s-plane

If there are any net clockwise encirclements of the (-1,0) point, there are zeroes of $1 + GH$ in the right half of the s-plane, and the system is unstable. The number of right half plane zeroes is given by the sum of N and P. In many cases the open loop system is known to be stable, i.e, there are no poles of GH in the right half s-plane. (Note that if the frequency response characteristics of GH are determined by measurement, GH must be stable.) Thus N must equal zero for the system to be stable.

Nyquist plots are normally made using polar notation. The magnitude and phase of $GH(s)$ is calculated and plotted for various values of ω when s is replaced by $j\omega$. Thus frequency information is not directly available from the plot; it must be obtained by estimation for points on the plot which are located between calculated points.

The Bode method, developed by Hendrik Bode of the Bell Telephone Laboratories in the 1930's, is a graphical technique which provides frequency information directly. This technique utilizes two graphs: the magnitude of $GH(j\omega)$ and the phase of $GH(j\omega)$, each plotted as a function of ω. For ease in construction and interpretation, logarithmic scales are commonly used for both the frequency and gain axis. For well-behaved systems an indication of stability is obtained by an examination of the gain at the frequency which results in 180° of phase, or the phase at the frequency which produces a gain of unity.* In the former case the gain should be less than unity while in the latter case the phase lag should be less than 180°.

The final method of linear system analysis to be discussed was developed by Evans [4] in 1948. This technique, referred to as the Root-Locus method, is a graphical representation of the s-plane loci of the variation of the poles of the closed-loop system function with changes in open-loop gain. [5] The poles and zeroes of the open loop system are located in the s-plane; then the loci of the closed-loop poles are plotted as some parameter is varied (usually open loop gain). If these loci are totally contained

---

* Well-behaved systems are defined as those which do not have any open loop poles in the right half plane and only have one 180° crossing and/or one unity gain crossing. For systems which are not well-behaved, stability is best studied with the Nyquist criterion.

within the left half of the s-plane, the system is stable. Should loci cross into the right half plane, the system will be unstable for the values of the parameter that result in that part of the locus in the right half plane.

Unfortunately the well developed techniques of linear systems analysis do not in general extend to nonlinear systems. Some authorities [6] hold that a general method of synthesis is impossible since a nonlinear system is one whose response is dependent upon its input. Nevertheless, techniques have been developed which apply to certain types or classes of nonlinear systems or which give limited information about system performance. Several of these techniques are discussed briefly here and are presented in detail in texts such as Hsu and Meyer [7] and those previously mentioned.

There are several unusual characteristics of nonlinear systems which are of interest. These are defined here and will be referred to throughout this discussion.

1. LIMIT CYCLE: Limit cycles are oscillations of fixed amplitude and period which can occur in nonlinear systems. There are two types of limit cycles, stable and unstable. When a system is operating in a stable limit cycle, any small disturbance results eventually in a convergence to the original limit cycle. On the other hand, any small disturbance to a system operating in an unstable limit cycle results in eventual operation at some other point. A system may exhibit behavioral modes that include both stable and unstable limit cycles.

2. HYSTERESIS: Hysteresis is a nonlinear phenomenon wherein the output depends not only upon the value of the input, but also upon its history. A familiar example is the conventional magnetization curve.

Many nonlinear systems, in which the deviation from linearity is not too great, may be examined using linear techniques. In these quasi-linear systems, the nonlinearity is replaced by a linear approximation of its characteristics. While the resulting analysis is approximate, this approach is useful under the concept that an approximate answer is better than no answer at all. Although practical systems are almost never purely linear they are often quasi-linear; this is precisely why techniques for analyzing linear systems produce such good results.

The describing function is an attempt to extend transfer function analysis techniques developed for linear systems to the analysis of quasi-linear systems. The normal form of this technique is a frequency-response method which is based on an analysis which neglects the effects of all harmonics except the fundamental*. It is therefore most successful when applied to systems which contain components that are low-pass.

A single-loop system with one nonlinearity is shown in Figure 1.2. Since the describing function technique applies directly only to autonomous systems, the system is more properly

---

* An extension of the normal describing function method is the dual input describing function, which retains two harmonics or the fundamental and dc term. The dual input describing function is not considered here.

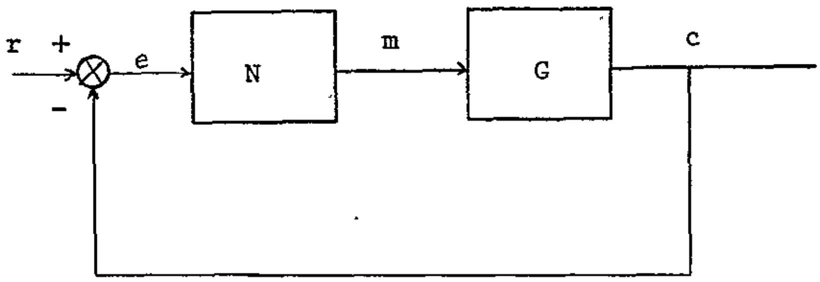


Figure 1.2. Single-loop system with nonlinearity N

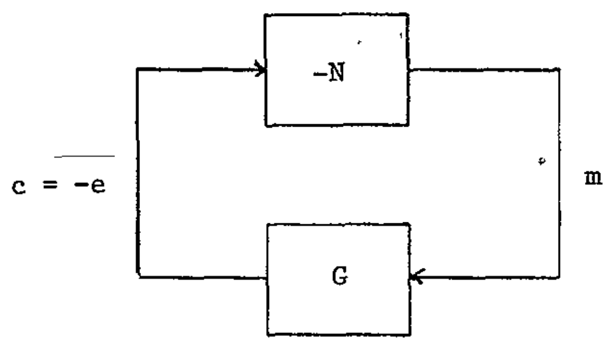


Figure 1.3. Autonomous system with nonlinearity N

considered in the forms of Figure 1.3. It is assumed that the linear element transfer function G is low-pass and allows only the fundamental component of m to pass. Therefore c , and thus e are sinusoidal and may be written as

$$e = E \sin \omega t \tag{1-3}$$

The output m of the nonlinearity may be expressed as

$$m = K(E) + f_d(e) \tag{1-4}$$

where K is a quasi-linear equivalent gain of the nonlinearity and f_d is a distortion term.

The distortion term f_d may be minimized in a mean-square sense by choosing K as the normalized coefficient of the fundamental of the output wave. Thus with an input e as given in equation (1-3) the Fourier series representation of the output will be

$$m = h_1 \sin \omega t + h_2 \cos \omega t + \ldots \tag{1-5}$$

The describing function is defined, in cartesian coordinates, as

$$K \triangleq \frac{h_1}{E} + j \frac{h_2}{E} \tag{1-6}$$

In general, for symmetric single-valued nonlinearities there will be no phase shift in output fundamental; the imaginary part will be zero. There will be phase associated with double-valued, or hysteresis, functions and with nonsymmetric nonlinearities.

Applying the describing function approximation to N yields

$$m = -Kc \tag{1-7}$$

If a limit cycle exists only the fundamental component is retained in this approximation. The steady-state form of the linear transfer function can thus be used; that is

$$G(j\omega) = \frac{c(j\omega)}{m(j\omega)} \tag{1-8}$$

If indeed a limit cycle does exist, (1-7) and (1-8) must be satisfied simultaneously, giving

$$G(j\omega) = -\frac{1}{K} \tag{1-9}$$

This relationship is the basis of a graphical analysis technique making use of the Nyquist criterion.

First, on a polar graph a Nyquist plot of the linear system is made as illustrated by $G(j\omega)$ in Figure 1-4. Next, a plot of $-\frac{1}{K}$ is made as a function of E on the same polar graph. Oscillations are indicated by intersections of the $G(j\omega)$ and the $-\frac{1}{K}$ curves as indicated by (1-9).

If the describing function contains no phase shift, i.e., the imaginary part is zero, the plot of $-\frac{1}{K}$ lies along the negative real axis, as illustrated by curve k_1 in Figure 1.4. If the nonlinearity can cause phase shift between its input and output a curve as illustrated by k_2 can be expected. If the $G(j\omega)$ curve and the k curve intersect, limit cycles will exist with approximate frequencies being those at the intersections. The values of

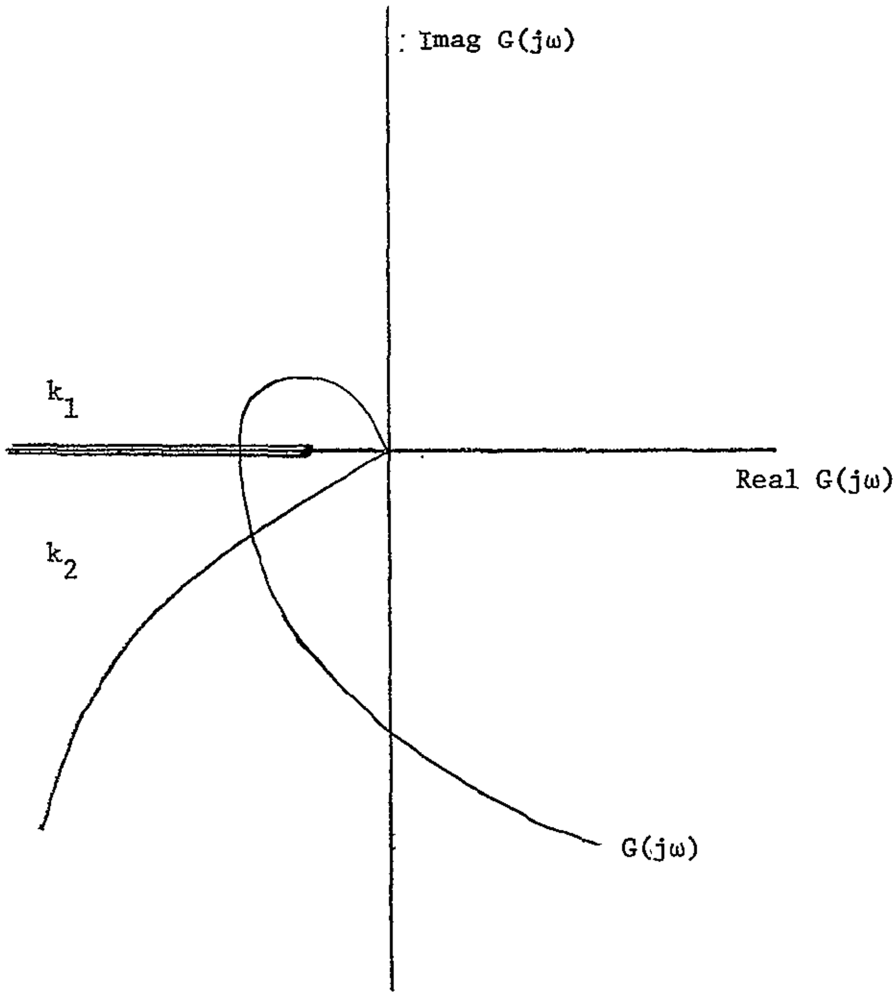


Figure 1.4. Nyquist-describing function plot of system of figure 1.3

E at which the intersections occur are the approximate amplitudes of the limit cycles.

Whereas the describing function technique is an adaptation of linear system analysis methods to the study of nonlinear systems, the phase plane method involves a radically different approach. This analysis is concerned with the characteristics of the solution of the differential equation

$$\ddot{x} + a(x,\dot{x})\dot{x} + b(x,\dot{x})x = 0 \tag{1-10}$$

The terms $a(x,\dot{x})$ and $b(x,\dot{x})$ are functions of the signal x and its derivative $\dot{x}$. The derivative $\dot{x}$ is plotted as a function of x producing a phase plane portrait. The initial conditions $\dot{x}(0)$ and $x(0)$, for time $t = 0$, define a point in the phase plane; the behavior of the system at all later times is identified by the curve whose path passes through this defined point. If as time increases the path tends to infinity, the system is unstable; if the path tends toward the origin, the system comes to rest and is stable. There is also the possibility that the path may close on itself; this indicates the existence of a limit cycle.

Around the turn of the century, Russian mathematician A. M. Lyapunov presented a useful stability criterion which has found extensive application in the past 20 years. This approach examines the response of a system at an equilibrium point when it is disturbed slightly.* An equilibrium point x_e is stable in the sense

* An equilibrium point is a state in which the system is in equilibrium with no tendency to move unless driven.

of Lyapunov if, for a given $\varepsilon > 0$, there exists a $\delta > 0$ such that when the initial disturbing condition x_o satisfies

$$||x_o - x_e|| < \delta \tag{1-11}$$

then

$$||x(t) - x_e|| < \varepsilon \quad \text{for all } t \geq 0 \quad . \tag{1-12}$$

A more restrictive sense of stability requires that the system return to the equilibrium point after disturbance. This condition (termed asymptotic stability) is met if, in addition to the previous requirements for stability (Equations 1-11 and 1-12),

$$\lim_{t\to\infty} x(t) = x_e \tag{1-13}$$

$$\text{for} \quad ||x_o - x_e|| < \delta$$

These conditions can be shown to be true if a function $V(x)$, called a Lyapunov function, can be found which is positive everywhere (except possibly zero at x_e), and whose derivative is negative everywhere except at the equilibrium point. The function $V(x)$ is analogous to (but in general not equal to) the energy stored in the system. If it is everywhere positive but decreasing, the system is continually settling back toward its equilibrium point and is asymptotically stable. Note that any function $V(x)$ which satisfies these conditions proves asympototic stability. It should be noted that $V(x)$ is not necessarily unique.

The techniques presented here represent a cross-section of methods useful in control system analysis; it is far from an exhaustive listing. Several other methods are discussed in the previously referenced texts and in the current literature. Generally these other methods either are restricted to a select class of systems, or contribute no information about stability which could not be obtained using the described methods. However, since each method has certain limitations and strengths, one finds it useful to apply several techniques when carefully examining a particular system. Insight to system characteristics is frequently gained from one analysis which is not evident using other techniques.

CHAPTER II

THE PROBLEM

The analysis of nonlinear feedback control systems is generally a complex task requiring a variety of tools. Sometimes a complete characterization of the system is not necessary; identification of limit cycles often yields sufficient information. The methods discussed previously are quite useful and valuable; however, they each suffer from certain limitations.

The technique of Lyapunov offers an important advantage in that stability of systems can be determined without having to solve the system equations. The method does require the identification of a Lyapunov function, however, which merely provides sufficient conditions for stability. If a function of the required type cannot be found for a given system it does not mean that the system is unstable. It means simply that the attempts to establish stability have failed. If, on the other hand, stability is established, the resulting region of stability may be more restrictive than necessary; stability may exist in a region larger than that specified by the method.

In addition, no systematic procedure exists for evaluating an arbitrary function of several variables. As a result, test functions are frequently restricted to the quadratic form, greatly limiting one's choice of test functions.

The graphical phase-plane method, in addition to providing stability information, also provides some insight into system characteristics. Limit cycles and stable equilibrium points are readily identified and some indication of system response is obtained. The technique is unfortunately limited to second-order systems since higher-order derivatives cannot be conveniently displayed on a two dimensional plot. (Knowledge of the variable and its first derivative would not completely define a higher-order system.) Also, as with all graphical techniques, the accuracy of the results is at least partially dependent upon the care used in the graphical construction methods. These methods are often tedious and somewhat prone to error.

Perhaps one of the most widely used techniques for determining limit cycles in practical control systems, particularly those of higher-order, is the classical describing function method. This method is based on an analysis which assumes that the signals in the system can be completely characterized by a truncated Fourier series which neglects the effect of most harmonics. With only a single nonlinearity, this assumption may yield incorrect or misleading conclusions. [8] In considering multiple nonlinearities, questionable results are even more likely.

The graphical approach normally associated with the classical describing function is not suitable when considering the effects of retaining more than the fundamental component of the truncated Fourier series. An extension of the describing function technique allows the consideration of two components. This is accomplished by use of the dual-input describing function, but at the expense of increased complexity of the analysis technique. In addition, these techniques are limited to the consideration of single path, single nonlinearity systems, although (as will be seen later) this restriction can sometimes be relaxed.

Gelb [9] developed a method of analysis of systems of multiple identical nonlinear-linear combinations. He observed that in a system such as that of Figure 2.1, for nonlinear operation on a limit cycle to exist, $x_1 = x_2$ except for a time phase difference. The method is applicable only if the linear-nonlinear combinations are identical.

Gronner [10] developed a method of analysis for systems having cascaded nonlinearities. The cascaded nonlinearities are first reduced to a composite; the describing function application of the Nyquist plot is used to identify any limit cycles. The limitations usually associated with the describing function technique apply to the resulting composite.

Gran and Rimer [11] developed a modified Root-Locus technique using the describing function representation of any nonlinearity to give closed-loop analysis. This technique may be used to determine the existence of any limit cycles and to obtain stability information. The method works for both amplitude and frequency sensitive nonlinearities, and for multiple nonlinearities separated by dynamics. Since system dynamics are displayed in pole-zero configuration, the technique is useful for system synthesis. Because the describing function is used, the previously enumerated limitations apply.

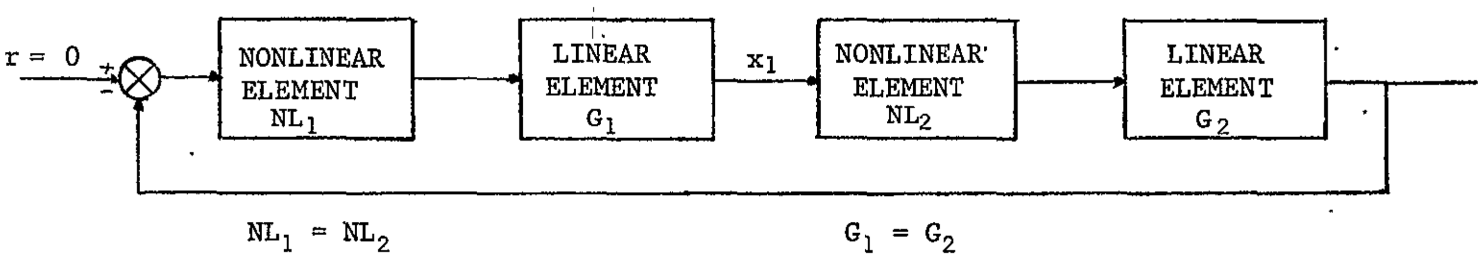


Figure 2.1. Multiple nonlinearity system with identical nonlinear-nonlinear pair

Jud [12] devised a method using the describing function technique to determine limit cycle conditions for any number of parallel nonlinear elements as long as signals reaching each nonlinear element have passed through a low pass filter. The implication is that each nonlinearity is preceeded by a low pass linear element. Thus, only a fundamental component of frequency reaches the nonlinearity, and the describing function produces only a fundamental component on the output of the nonlinearity.

Negoescu and Sebastian [13] introduced a method for determining self-oscillation in a nonlinear system which contains two nonlinearities separated by a linear device. The nonlinearities appear in parallel feedforward paths so constructed that in all paths each nonlinear element is separate from the other nonlinear element by a linear element, as shown in Figure 2.2. System equations are written, then examined for solutions. The existence of roots with positive real-parts indicates oscillations.

The most general method of analysis applicable to systems with linear and nonlinear elements was developed by Davison and Constantinescu [14]. While the technique is based on a single-loop system as shown in Figure 2.3, the authors have extended its use to include parallel systems of the type depicted in Figure 2.4. The normal describing function approximations are assumed, i.e., the linear elements are sufficiently low pass that, if a limit cycle exists, only the fundamental frequency component need be considered.

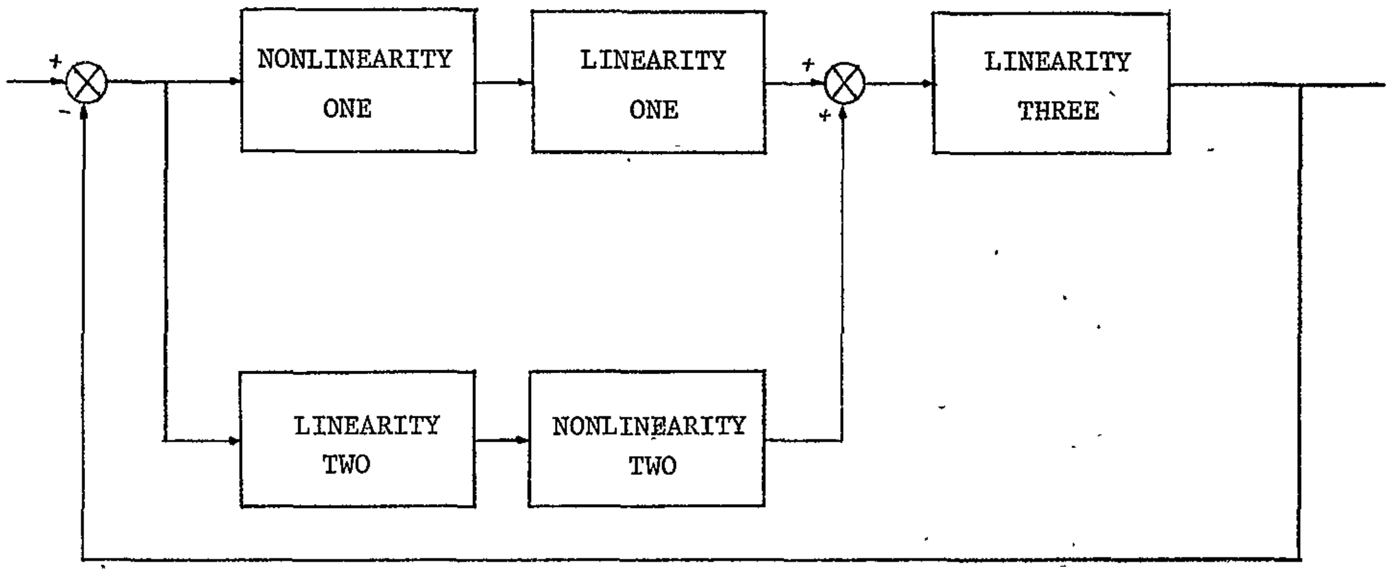
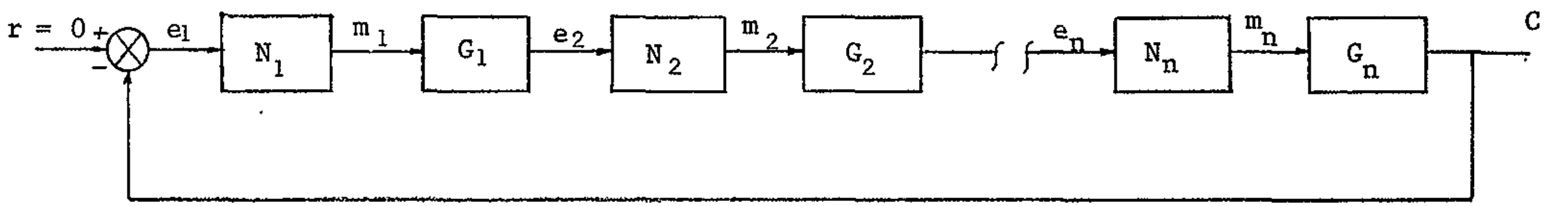
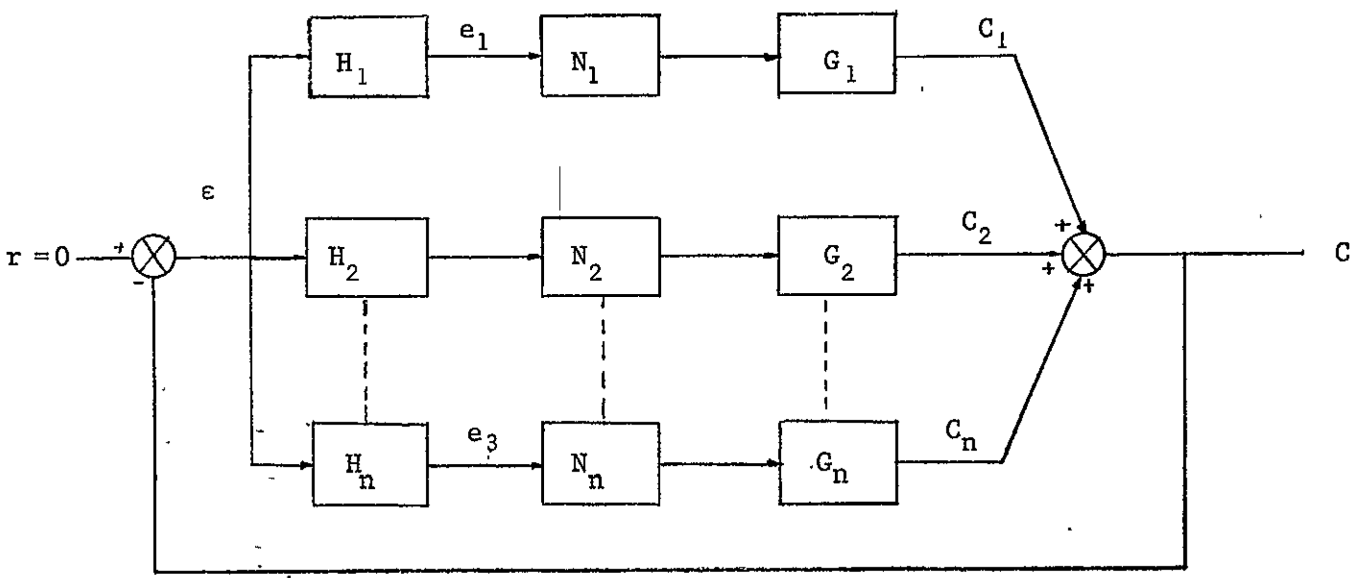


Figure 2.2. Parallel multiple nonlinearity system



N = NONLINEAR ELEMENT G = LINEAR ELEMENT

Figure 2.3. Single-loop system with n nonlinear pair



H = LINEAR ELEMENT N = NONLINEAR ELEMENT

G = LINEAR ELEMENT

Figure 2.4. System with n nonlinear elements in parallel

Under these conditions, the inputs to the nonlinearities take the form of

$$\begin{aligned} e_1(t) &= A_1 \sin \omega t \\ e_2(t) &= A_2 \sin (\omega t + \theta_2) \\ &\vdots \\ e_n(t) &= A \sin (\omega t + \theta_n) \quad . \end{aligned} \tag{2-1}$$

Since the describing function may be considered a linear gain whose magnitude M and phase ϕ are dependent upon the input magnitude, the nonlinearity outputs may be written as

$$\begin{aligned} m_1(t) &= M_1 A_1 \sin(\omega t+\phi_1) \\ m_2(t) &= M_2 A_2 \sin(\omega t+\theta_2+\phi_2) \\ &\vdots \\ m_n(t) &= M_n A_n \sin(\omega t+\theta_n+\phi_n) \end{aligned} \tag{2-2}$$

If the magnitude of the sinusoidal input to the nonlinearity is fixed, conventional linear analysis techniques may be applied to the system.

For the limit cycles to exist

$$\prod_{i=1}^{n} G_i(j\omega^*)\, G_{di}(A_i) = -1+j0 \tag{2-3}$$

where ω^* is the frequency of oscillation of the limit cycle, $G_{di}(A_i)$ is the magnitude and phase of the linear gain equivalent of the i^{th} nonlinear element and G_i is the gain of the i^{th} linear element.

In addition, each linear element gain may be expressed in terms of the nonlinear gains and inputs. Thus

$$G_i(j\omega^*) = \frac{e_{i+1}}{e_i G_{d_i}(A_i)} \qquad i = 1,2,\ \ldots,\ n-1 \tag{2-4}$$

$$G_n(j\omega^*) = \frac{e_1}{e_n G_{d_n}(A_n)}$$

When combined with (2-1), (2-4) may be written

$$\begin{aligned} A_2 &= A_1 |G_1(j\omega^*)| \, |G_{d1}(A_1)| \\ A_3 &= A_2 |G_2(j\omega^*)| \, |G_{d_2}(A_2)| \\ &\vdots \\ A_n &= A_{n-1} |G_{n-1}(j\omega^*)| \, |G_{d_{n-1}}(A_{n-1})| \end{aligned} \tag{2-5}$$

A limit cycle exists if there is an A_1, A_2, A_3, ... and ω^* which simultaneously satisfies both (2-3) and (2-5). A graphical method may be used to determine any such solutions. The following procedure is used.**

1. Plot $\prod_{i=1}^{n} G_i(j\omega)$ as a function of ω.
2. Choose an arbitrary value for A_1.
3. Plot $\prod_{i=1}^{n} G_i(j\omega) \prod_{i=1}^{n} G_{d_i}(A_i)$ as in Figure 2-5, using equation 2-5 and step (1).
4. Determine possible limit cycle frequencies by noting the frequencies at which the plot of step (3) intersects the negative real axis.

**If the number of memory type nonlinearities does not exceed one, a somewhat simplified procedure may be used. See Davison, et. al. [14] for details.

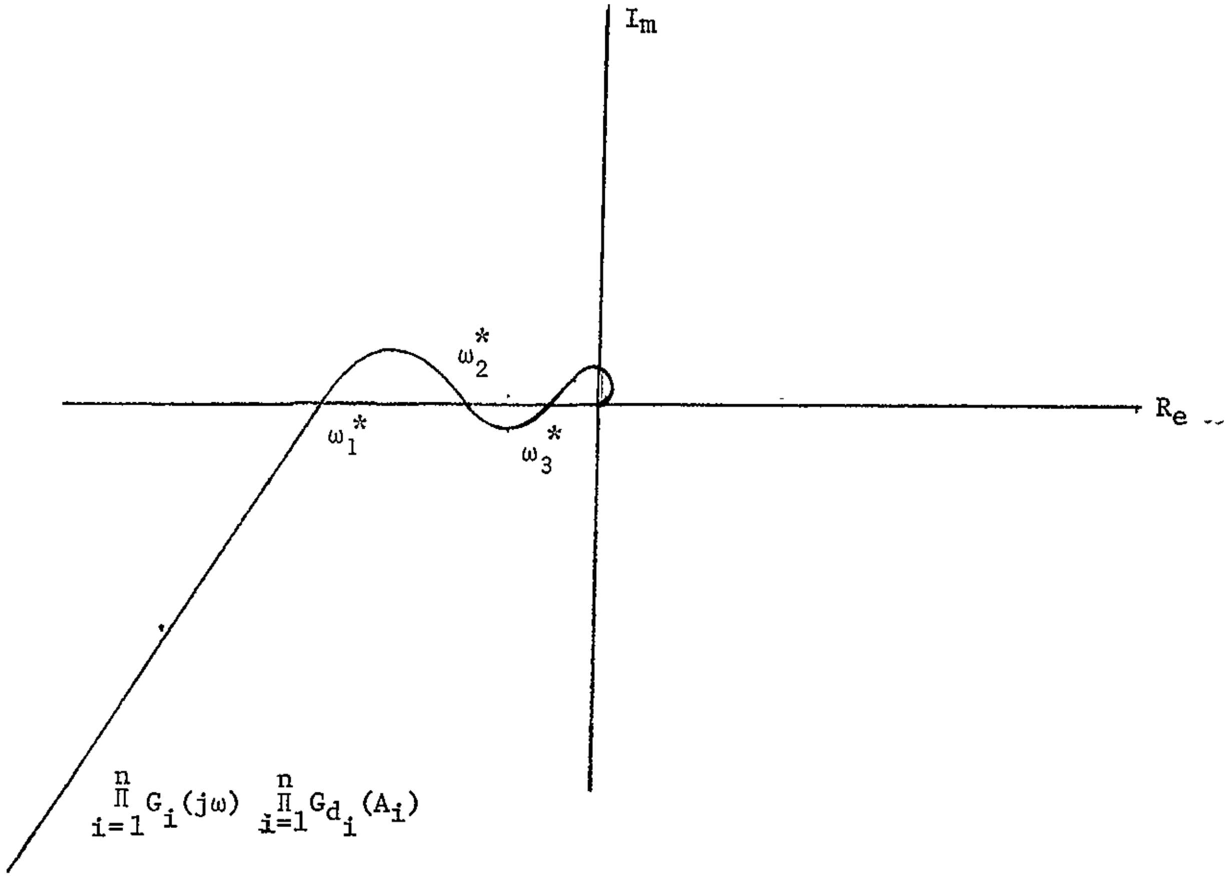


Figure 2.5. Graphical method for determining limit cycle frequencies, $\omega_1^*, \omega_2^*, \omega_3^*$

5. Determine the magnitude k of the plot of step (3) at frequencies determined in step (4).

6. Repeat steps (2)-(5) for different values of A_1. Plot the graph of k versus A as shown in Figure 2.6. The intersection with the k = 1 line identifies limit cycle frequency-amplitude values.

7. Using Equation 2-5, determine the remaining A_i values.

The system of Figure 2.4 (note that there is only one nonlinearity in each parallel path) may be analyzed in a similar fashion with the following modifications in notation:

The input ε is given by

$$\varepsilon = A \sin \omega t \qquad (2\text{-}6)$$

and the nonlinear element inputs are

$$e_i(t) = A_i \sin(\omega t + \phi_i) \qquad (2\text{-}7)$$

where

$$A_i = |H_i(j\omega)| A \qquad (2\text{-}8)$$

$$\phi_i = \angle (H_i(j\omega) \quad i = 1,2,\ldots n \qquad (2\text{-}9)$$

The conditions for the existence of limit cycles are Equation 2-.8 and

$$\sum_{i=1}^{n} H_i(j\omega)\ G_i(j\omega)\ G_{d_i}(A_i) = -1 + j0\ . \qquad (2\text{-}10)$$

The graphical procedure is as follows:

1) Choose A arbitrarily.

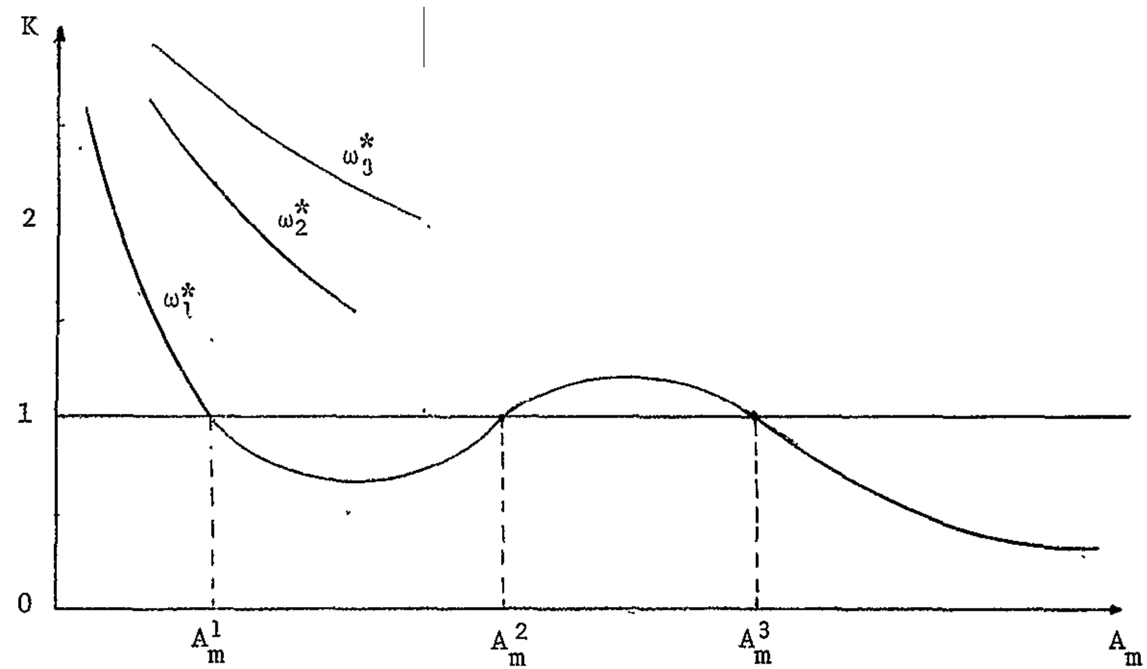


Figure 2.6. Plot of k versus A_m

2) Plot $\sum_{i=1}^{n} H_i(j\omega)\ G_i(j\omega)\ Gd_i(A_i)$ as a function of ω , making use of Equation 2-8.

3) Determine possible limit cycle frequencies from the intersection of the plot of step (2) and the negative real axis.

4) Determine $k = \left| \sum_{i=1}^{n} H_i(j\omega)\ G_i(j\omega)\ G_i(j\omega)\ Gd_i(A_i) \right|$ at the frequencies determined in step (3).

5) Repeat (1)-(4) for different values of A . Plot k versus A, noting the points of intersection with the $k = 1$ line.

6) Use (2-8) to determine corresponding frequencies of oscillation and input amplitude A_i for the limit cycles.

While the Davison-Constantinescu method provides a powerful technique for the analysis of systems with multiple nonlinearities, it suffers from the normal restrictions of all techniques using the describing function approximation. This approximation is based on the assumption that elements of the system are assumed to be low-pass. Thus the input to each nonlinearity is accurately represented by a sinusoid. If harmonic components are present at the nonlinearity inputs, results obtained by any method based on describing function analysis are suspect.

Thus there is a need for a method of analysis which relaxes the describing function restriction. The work presented in the following chapters removes the low-pass requirement of the system. The nonlinearity inputs may therefore contain a dc component plus several harmonics, as well as the fundamental frequency. In addition, parallel paths may contain multiple nonlinearities.

CHAPTER III

THE SOLUTION

In the last chapter the difficulties and problems associated with using describing function techniques to determine harmonic balance in feedback control systems were delineated. In this chapter an iterative approach for analyzing a general class of nonlinear feedback control systems for harmonic balance is developed, along with the necessary mathematical tools for implementation on the digital computer.

3.1 Harmonic Balance

A single loop time-invariant nonlinear feedback control system is shown in Figure 3.1. Under conditions of harmonic balance the signal at each point in the loop will be periodic and of the same period. Thus, each signal can be represented as a Fourier series, i.e.,

$$e\ (t) = a_o + \sum_{k=1}^{\infty} a_k \cos k\omega_o t + b_k \sin k\omega_o t \tag{3-1}$$

$$u\ (t) = \alpha_o + \sum_{k=1}^{\infty} \alpha_k \cos k\omega_o t + \beta_k \sin k\omega_o t$$

$$c\ (t) = A_o + \sum_{k=1}^{\infty} A_k \cos k\omega_o t + B_k \sin k\omega_o t$$

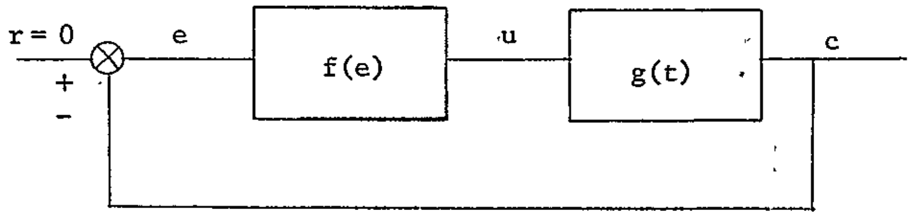


Figure 3.1: Single loop nonlinear, time-invariant control system.

The necessary and sufficient condition for harmonic balance is

$$e(t) = -c(t) \ . \tag{3-2}$$

Therefore, each term in the series for $e(t)$ must be equal to the corresponding term for $-c(t)$. This requires that

$$\begin{aligned} a_o &= -A_o \\ a_k &= -A_k \qquad k = 1,2,3,\ldots \\ b_k &= -B_k \end{aligned} \tag{3-3}$$

In processing the signal the linear element may alter the magnitude and phase of the various harmonics as a function of their corresponding frequencies, but no new harmonics will be generated. The nonlinear element, however, can significantly alter the harmonic content of the signal. The establishment of harmonic balance depends upon the selection of the magnitudes and frequencies of the signal such that the coefficient equalities in (3-3) are satisfied. This requires the solution of an infinite number of simultaneous, nonlinear, algebraic equations in an infinite number of unknowns (obviously an *impasse*).

Fortunately the problem can be somewhat simplified. In most practical systems the linear or nonlinear element can be assumed to be band limited, which results in a finite number of terms in the Fourier series representation of the output signals.*

The problem then becomes one involving only a finite number of equations in a finite number of unknowns, which can be solved using iterative procedures. Such procedures examine the output produced from an initial guess of the input, then refine the input to produce an output that more nearly satisfies (3-3). This process is repeated until the output matches the input within a desired tolerance.

3.2 General System Considered

The system of Figure 3.1 can be expanded to include several paths with multiple nonlinearities in each path, as shown in Figure 3.2. In order for harmonic balance to exist

$$\sum_{\ell=1}^{n} Y_\ell = c = -e \tag{3-4}$$

for all components of Y, c and e. That is, the dc component of e must be equal to the negative sum of the dc components of the Y's; a similar condition is true for each component of the Fourier series representation of the Y's and e.

* It should be noted that the describing function technique solves the harmonic balance problem in this way. In the describing function case only one or two terms are retained in the Fourier series.

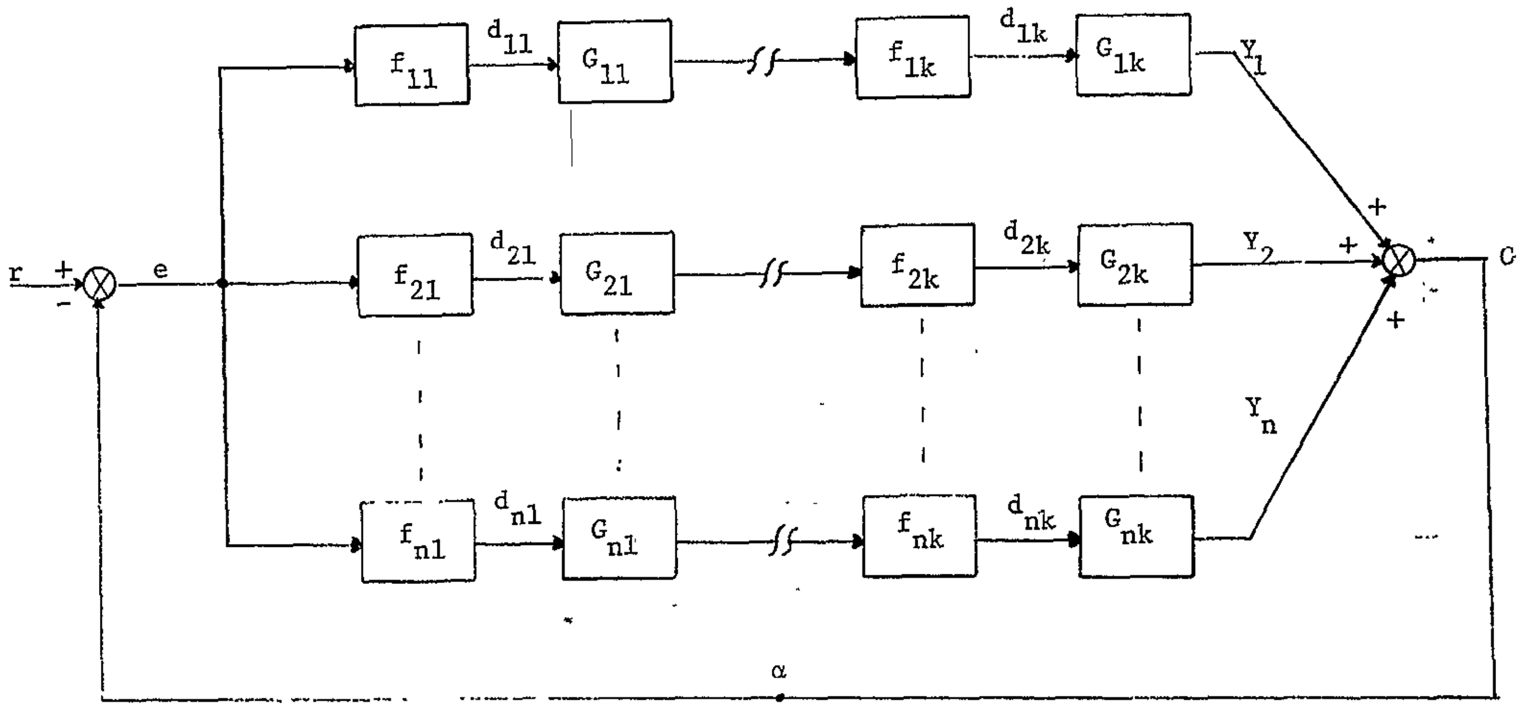


Figure 3.2. Multi-nonlinearity, multi-path system

The goal of the work presented here is to develop a computer code to determine harmonic balance in nonlinear feedback control systems with the general configuration shown in Figure 3.2. The technique that is developed on the ensuing pages is designed to contend with several Fourier series terms in the establishment of harmonic balance (thus making it superior to the describing function methods mentioned in Chapter 2).

In the development of any technique of this type some limiting assumption are necessary. In this case they are:

1) The system can be separated into linear and nonlinear components.
2) No input signal is applied.
3) The nonlinear element is time- and frequency-invariant.
4) The input/output characteristics of the nonlinear element can be approximated by piecewise-linear segments.
5) The nonlinear elements are single input/single output.
6) The linear elements are relatively low-pass.*

3.3 The Harmonic Balance Algorithm (HBA)

The flow chart in Figure 3.3 shows the major steps which must be undertaken to solve iteratively the harmonic balance problem. Using an initial guess of a limit cycle (including frequency and

---

*It will be assumed that no frequency component higher than the highest harmonic retained in the analysis will be passed by an element in the system.

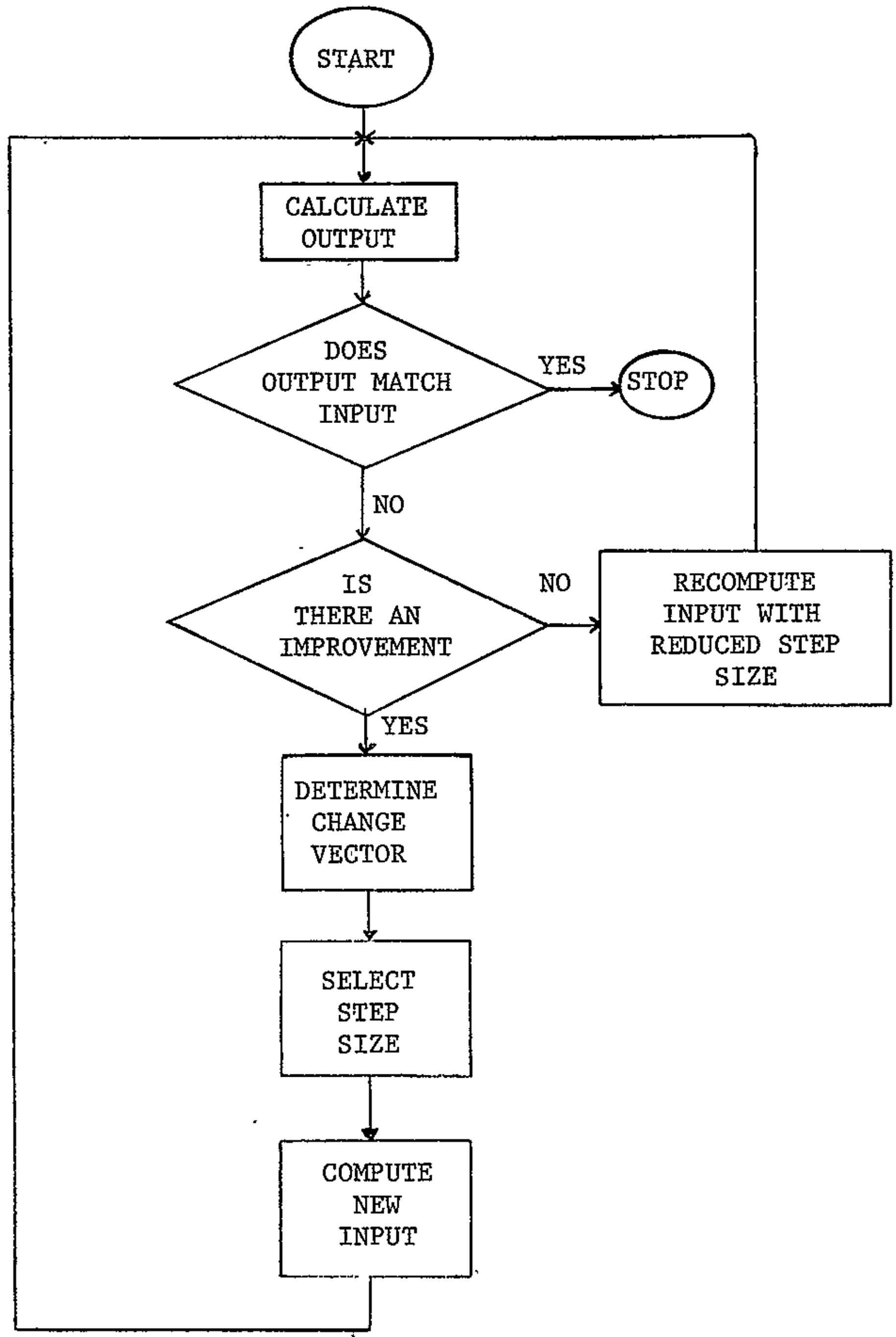


Figure 3.3. Flow chart for HBA

amplitude components) the system output is determined. For the present a "No" and "Yes" are, respectively, assumed for the two decision blocks. An input change vector is then determined so as to reduce the difference between the input and the output components. Using this change vector, a step size is selected, and the input values are incremented.

This new input is utilized in computing a new output. If the new output represents an improvement, the process continues through another cycle. If not, the selected step size is reduced and new input values are determined. This input is then used to recalculate the output, initiating a new cycle.

Each time new output values are computed they are compared to the input values to determine if another iteration is needed. The process continues until output and input agree within the desired tolerance, at which time the procedure terminates.

The details of the process together with the mathematical basis for the procedure are presented in the following sections. The FORTRAN code that implements the algorithm is presented in Appendix B. Directions for using the HBA and a sample output are presented, respectively, in the Appendices A and C.

3.4 Determination of System Output

Harmonic balance is established by the necessary and sufficient condition that

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = - \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} . \tag{3-5}$$

where the subscripts correspond to the Fourier components of the variable. The procedure described in the last section for accomplishing this equality involves several distinct operations (the efficient performance of which are central to the development of a pragmatic analysis tool).

First, the computation of the Fourier series of the output of nonlinearities when the input is a truncated Fourier series is necessary. At the onset of this research effort several methods were investigated: a Simpson's numerical integration technique, a Kalman filter technique, and a method devised by Braswell [15] in his Relaxation-Based Algorithm. As a result of testing, Braswell's technique was ultimately chosen on the basis of comparable accuracy with somewhat faster execution time.

The Braswell procedure uses the secant method (also known as Regula falsi, or method of false postion) to determine the times $\{t_i\}_{i=1}^{I-1}$ which identify the nonlinear breakpoint crossings by the input signal. These times are illustrated in Figure 3.4. Each breakpoint is identified by a coordinate pair consisting of the value of the input and the resulting output value at the junction of two of the linear-piecewise segments. The slope of each segment is also specified.

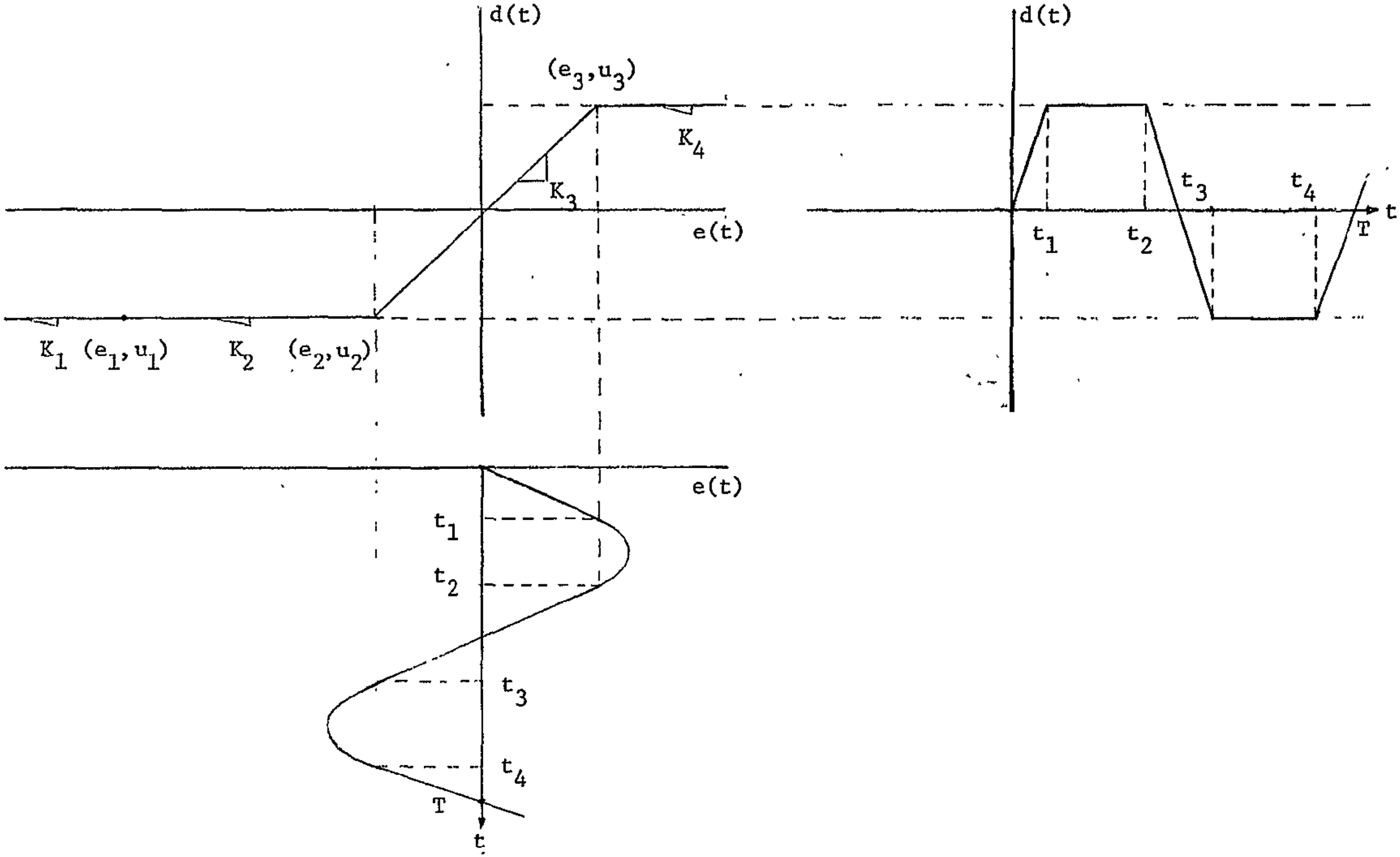


Figure 3.4. The response of a nonlinearity to a sine waveform

Once each breakpoint time is identified the proper breakpoint coordinate and slope are linked to it. With this information, then

$$d_m = \frac{1}{T} \sum_{i=1}^{I} B_i \int_{t_i-1}^{t_i} e^{-jm\omega_o t} dt \qquad (3\text{-}6)$$

$$+ \frac{1}{T} \sum_{i=1}^{I} K_i \sum_{n=-N}^{N} C_n \int_{t_i-1}^{t_i} e^{j(n-m)\omega_o t} dt$$

is used to calculate the resulting complex Fourier series coefficients of the nonlinearity output. In (3-6), d_m is the Fourier coefficient of the m^{th} harmonic term and K_i is the slope of the piecewise-linear segment used on the interval $[t_i, t_{i-1}]$. Also $B_i = u_{i-1} - K_i e_{i-1}$; this relationship identifies the ordinate intercept of the piecewise-linear segment.

In processing the signal, linear elements may alter the magnitude and phase of the various harmonics, but no new components will be created. Since the principle of superposition applies, the linear element output can be obtained by multiplying each harmonic component of the nonlinearity output by the gain and phase of the linear element at the harmonic frequency. Thus,

$$Y_m = G(jm\omega_o)\, d_m \qquad m = 0,1,2,\ldots,M \qquad (3\text{-}7)$$

where the m represents the zeroth, first, ---, and Mth harmonics.

The technique is applied to each path of Figure 3.2 and the results are summed to produce the system output. That is, the output of the first linear element in path one is taken as the input to nonlinear element two to obtain the output of linear element two. This, in turn, is used at the input of nonlinearity three to determine the output of linearity three. This process is continued until all linear-nonlinear pairs of path one are considered. Then the original input is assumed at nonlinear element one of path two and the path two output is determined. The process is repeated until all path outputs are found and their sum is obtained.

This procedure allows one to determine the output which results for a given input. Some means are necessary to determine the changes in input which would result in the solution of the relationship

$$c_n - e_n = 0 \,, \tag{3-8}$$

giving harmonic balance. This becomes a zero finding problem in which the variables are frequency, ω, and the Fourier coefficients of the input (and the output) signals.

In actuality each input consists of a Fourier series with an infinite number of terms. Thus, it is necessary to solve an infinite set of algebraic equations in an infinite number of unknowns. However, since the linear elements are assumed to be relatively low pass, only a finite number of terms will be significant. The resulting nonlinear simultaneous equations are of such complexity that they do not lend themselves to an exact analytical solution. A numerical solution technique base on the Newton-Raphson method is devised in the next section.

3.5 Determination of Trial Solutions

Another operation in Figure 3.3 is the computation of the input directional change vector. The Newton-Raphson method is an iterative technique that can be used for solving a system of simultaneous nonlinear equations [16] (such as those found in this problem). The method automatically selects the increment to be used in determining the next iteration input and as a rule exhibits a rapid rate of convergence (if it converges). The Newton-Raphson method is selected for the solution tool in this work. However, it is modified to improve convergence.*

The technique will first be illustrated for a system of two equations, i.e.

$$f_1(x_1, x_2) = 0 \tag{3-9}$$

$$f_2(x_1, x_2) = 0 .$$

An initial guess is made for the unknowns x_1 and x_2. Small increments Δx_1 and Δx_2 are assumed to give the correct solution to (3-9). That is,

$$f_1 (x_1^o + \Delta x_1 , x_2^o + \Delta x_2) = 0 \tag{3-10}$$

$$f_2 (x_1^o + \Delta x_1 , x_2^o + \Delta x_2) = 0$$

where x_1^o, x_2^o, Δx_1, and x_2 are respectively the initial guesses and increments.

*This results in essence in a gradient procedure.

Expanding Equation 3-10 in a Taylor series around the initial guess yields

$$f_1(x_1^o\ ,\ x_2^o) + \Delta x_1\ (\frac{\partial f_1}{\partial x_1}) + \Delta x_2\ (\frac{\partial f_1}{\partial x_2}) + \ldots\ = 0 \tag{3-11}$$

$$f_2(x_1^o\ ,\ x_2^o) + \Delta x_1\ (\frac{\partial f_2}{\partial x_1}) + \Delta x_2\ (\frac{\partial f_2}{\partial x_2}) + \ldots\ = 0$$

where the partial derivatives are evaluated at the initial guesses, x_1^o and x_2^o .

If the increments are assumed to be small, higher-order terms of Equation (3-11) are negligible. Equation (3-11) may thus be written in matrix-vector form as

$$F + J\ \Delta X = 0 \tag{3-12}$$

where

$$J \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \tag{3-13}$$

(the Jacobian matrix of the functions f_1 and f_2 with respect to x_1 and x_2),

$$F \triangleq \begin{bmatrix} f_1(x_1^o\ ,\ x_2^o\ ) \\ f_2(x_1^o\ ,\ x_2^o\ ) \end{bmatrix} \tag{3-14}$$

and

$$\Delta X \triangleq \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \tag{3-15}$$

Solving Equation (3-12) for ΔX yields

$$\Delta X \simeq -[J]^{-1} F \tag{3-16}$$

The components of ΔX are Δx_1 and Δx_2. When these values are added to the initial estimates x_1^o and x_2^o an approximate solution of (3-10) is determined. However, since they are approximate, the solution is not exact and additional refinement in the values of the x's are needed. This is obtained by using the values of $x_1^o + \Delta x_1$ and $x_2^o + \Delta x_2$ as new initial guesses of x_1 and x_2 and performing the operation again. The process is repeated, obtaining a better estimate for x_1 and x_2 each time until the exact solution (or one within acceptable error limits) is obtained.* This concept can be extended to the general case by allowing ΔX and F to be n-dimensional vectors and J to be an n-by-n matrix.

For the case at hand, the Newton-Raphson zero finding technique is applied to (3-8). Except at the point of solution the results are not zero but

$$p_n = c_n - e_n \tag{3-17}$$

---

* This assumes convergence will occur. In general, the procedure will converge to $f(\xi) = 0$ if $f(x)$, $f'(x)$, and $f''(x)$ are continuous on some interval containing ξ, if $f'(\xi) \neq 0$, and if the starting point is "close enough" to the root ξ. These conditions cannot be assured for a general nonlinear function or with randomly chosen starting values. See Conte and de Boor [17].

The desired result, of course, is for p_n to be equal to zero.

Note that the c_n values are produced by the system of Figure 3.1 operating on the assumed input values of e. Thus, for two input variables x_1 and x_2, the desired form of (3-17) is

$$p_1 = f_1(x_1,x_2) - x_1 = 0 \tag{3-18}$$

$$p_2 = f_2(x_1,x_2) - x_2 = 0$$

where x_1 and x_2 comprise the input signal e.

The Newton-Raphson technique is applied to the variable p, resulting in

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + [J] \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \simeq 0 \quad ; \tag{3-19}$$

thus,

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = [J]^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{3-20}$$

The Jacobian matrix J is given by

$$J = \begin{bmatrix} \dfrac{\partial p_1}{\partial x_1} & \dfrac{\partial p_1}{\partial x_2} \\ \\ \dfrac{\partial p_2}{\partial x_1} & \dfrac{\partial p_2}{\partial x_2} \end{bmatrix} \tag{3-21}$$

The partial derivatives are computed by

$$\frac{\partial p_1}{\partial x_1} = \frac{\partial}{\partial x_1} \left(f_1(x_1, x_2) - x_1\right) \tag{3-22a}$$

$$= \frac{\partial f_1}{\partial x_1} - 1$$

$$\frac{\partial p_1}{\partial x_2} = \frac{\partial f_1}{\partial x_2} - \frac{\partial x_1}{\partial x_2} \tag{3-22b}$$

$$= \frac{\partial f_1}{\partial x_2}$$

$$\frac{\partial p_2}{\partial x_1} = \frac{\partial}{\partial x_1} \left(f_2(x_1, x_2) - x_2\right) \tag{3-22c}$$

$$= \frac{\partial f_2}{\partial x_1} - \frac{\partial x_2}{\partial x_1}$$

$$= \frac{\partial f_2}{\partial x_1}$$

and

$$\frac{\partial p_2}{\partial x_2} = \frac{\partial f_2}{\partial x_2} - \frac{\partial x_2}{\partial x_2} \tag{3-22d}$$

$$= \frac{\partial f_2}{\partial x_2} - 1$$

The Jacobian matrix may thus be written

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} - 1 & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} - 1 \end{bmatrix} \tag{3-23}$$

The system of Figure 3.2 can be equivalently represented as a single block as shown in Figure 3.5. Let the input now be CN and the output YN. Let these correspond to the complex components of the Fourier series representation of the input and output signals. The component parts of CN are: a_o , the dc component; b_1 , the imaginary part of the fundamental component; a_2 and b_2 , the real and imaginary parts of the second harmonic, etc. The a_1 component is assumed to be zero so that the input signal corresponds to a sine wave at the fundamental frequency.* In addition, the fundamental frequency ω is considered an input.

The system operates on these inputs to produce the output YN. The components of the output are the same as the input except that a nonzero a_1 component is possible. The frequency ω is not changed by the system and thus is equal to the input ω .

The input CN component parts, through the second harmonic, are represented by a_0 , b_1 , a_2 , and b_2 (remember that the real part of the fundamental component is zero and there is no imaginary part of the dc component) and the output YN components are A_0 , A_1 , B_1 A_2 , and B_2 . Equations 3-18 take the form

*The actual reason for setting the real part of the fundamental component to zero is to establish a fixed time reference.

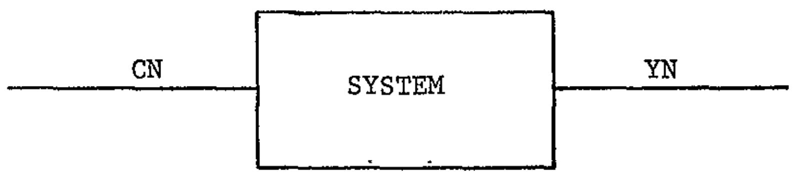


Figure 3.5. Single block representation of multi-element system

$$p_1 = A_0 - a_0 \tag{3-24}$$

$$p_2 = A_1 - 0 \text{ (since } a_1 = 0)$$

$$p_3 = B_1 - b_1$$

$$p_4 = A_2 - a_2$$

$$p_5 = B_2 - b_2$$

The Jacobian matrix then becomes

$$J = \begin{bmatrix} \frac{\partial A_o}{\partial a_o} - 1 & \frac{\partial A_o}{\partial b_1} & \frac{\partial A_o}{\partial a_2} & \frac{\partial A_o}{\partial b_2} & \frac{\partial A_o}{\partial \omega} \\ \frac{\partial A_1}{\partial a_o} & \frac{\partial A_1}{\partial b_1} & \frac{\partial A_1}{\partial a_2} & \frac{\partial A_1}{\partial b_2} & \frac{\partial A_1}{\partial \omega} \\ \frac{\partial B_1}{\partial a_o} & \frac{\partial B_1}{\partial b_1} - 1 & \frac{\partial B_1}{\partial a_2} & \frac{\partial B_1}{\partial b_2} & \frac{\partial B_1}{\partial \omega} \\ \frac{\partial A_2}{\partial a_o} & \frac{\partial A_2}{\partial b_1} & \frac{\partial A_2}{\partial a_2} - 1 & \frac{\partial A_2}{\partial b_2} & \frac{\partial A_2}{\partial \omega} \\ \frac{\partial B_2}{\partial a_o} & \frac{\partial B_2}{\partial b_1} & \frac{\partial B_2}{\partial a_2} & \frac{\partial B_2}{\partial b_2} - 1 & \frac{\partial B_2}{\partial \omega} \end{bmatrix} \tag{3.25}$$

Notice that because the a_1 component is always zero, there are no $\frac{\partial}{\partial a_1}$ terms. This eliminates one column of the Jacobian matrix. There is, however, a column corresponding to $\frac{\partial}{\partial \omega}$ since ω is an input. The matrix thus remains square.

The partial derivatives in the Jacobian matrix are computed by numerical means. This is accomplished by computing the ratio of the change in each output to a change in the input, e.g.

$$\frac{\partial A_o}{\partial a_o} \approx \frac{\Delta A_0}{\Delta a_0} \qquad (3\text{-}26)$$

As long as the deltas in the fraction are small, the approximation is sufficiently accurate. These values are computed, one column at a time, by obtaining the change in the output values as one of the input components is altered slightly. That component is returned to its original value and another is altered to calculate another column of values. The procedure is repeated until all values are derived.

Once all of the elements of the J matrix are obtained, the incremental changes in the input values are determined from

$$\begin{bmatrix} \Delta a_0 \\ \Delta b_1 \\ \Delta a_2 \\ \Delta b_2 \\ \Delta \omega \end{bmatrix} = [J]^{-1} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} \qquad (3\text{-}27)$$

where the p's are as shown in Equation 3-18. These delta values are then combined with the CN components to obtain the new values of CN to be used in the next iteration.

3.6 Iteration Limits

The numerical difference method of determining the partial derivative values of the Jacobian matrix is valid if the difference values are sufficiently small. In order to insure its validity, the change in each output value is monitored to assure that the change is not too large. The criterion chosen is based on limiting the maximum change to about 2.5% of the value of the output variable. If any of the outputs change more than this limiting value, the incremental size of the input producing the excessive output change is reduced by a factor of 2 and the outputs are recomputed. This process is continued until all output changes fall within the established limits.

Although the Newton-Raphson procedure rapidly converges to a solution in most cases, it is possible that the procedure will not converge at all. The automatic step size generation feature may result in step sizes which are too large to produce convergence. In order to circumvent this problem, the iteration step size is constrained to be no more than 1.5 times the last successful iteration step size. Also, for each iteration a cost function is computed by

$$COST = \sum_{i=1}^{I} p_i \tag{3-28}$$

This cost is compared to the cost computed for the previous iteration; if the cost has not decreased, the iteration step size is reduced by a factor of five and the cost is recomputed. If four attempts are made without a cost reduction, the last reduced step size is accepted.*

*It should be noted that by limiting the step size the Newton-Raphson procedure is really a gradient procedure.

Recognizing that numerical noise may become significant for computed values less than 10^{-8}, provision is made to terminate the process when iteration step sizes are consistently of this magnitude. Four consecutive iterations with step size less than 10^{-8} will terminate execution of the code.

The procedure is also terminated when an output limiting function is less than 10^{-8}. This function is found by

$$\mathrm{YNOR} = \sqrt{\sum_{i=1}^{I} [YN_i]^2} \tag{3-29}$$

When this limiting function is less than 10^{-8}, all output components are assumed to be zero.

The limiting function is also used to identify another stopping criterion. If each component of the output differs from the corresponding input component by an amount which is less than 0.1% of the limiting function of Equation 3-29 the procedure is terminated; a limit cycle has been successfully identified.

Several example systems were considered using the procedure outlined above. The results which were obtained are presented in the next chapter.

CHAPTER IV

RESULTS OF APPLYING THE HBA

The Harmonic Balance Algorithm (HBA) discussed in the previous chapter was applied to several example problems. These examples include single-path, multiple-nonlinearity systems and multi-path systems; they incorporate both memory and memoryless type nonlinearities. The results obtained are summarized in Tables 4.1 through 4.6.

The system of Figure 4.1 has two identical nonlinearities but dissimilar linear elements. With point A as the input, randomly chosen initial values were selected; the initial value of ω was set at 0.5 radians per second with a fundamental component amplitude of either 5 or 10 units. With an initial input of 5 units, the HBA terminated when the computed input increment consistently remained less than the perturbation for determining the J-matrix. However, where the input was initially set to 10 units, a limit cycle was identified. The frequency was determined as 1.015 radian/second with a fundamental amplitude of 18.16 units. As might be expected from the asymmetrical nature of the nonlinearities, a small dc component of -0.867 units was also determined. Second harmonic cosine and sine components were determined to be -0.124 and -0.110 respectively, while the third harmonic components were -0.819 and -0.374. (Only components through the 3rd harmonic were calculated.)

With the order of the linear elements reversed by choosing point B as the input, identification of a limit cycle proved to be

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 3 $e(t) = 10 \sin 0.5t$	1:29.9	$e(t) = -0.867 + 18.16 \sin 1.015t$ $-0.124 \cos 2.029t - 0.110 \sin 2.029t$ $-0.819 \cos 3.045t - 0.374 \sin 3.045t$
Number of Harmonics = 3 $e(t) = 5 \sin 0.5t$	2:04.4	Terminated On Step Size
Number of Harmonics = 1 $e(t) = 5 \sin 0.5t$	:14.7	Terminated on Step Size

Table 4.1 HBA Results For System of Figure 4.1, Input At A .

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 3 $e(t) = 5 \sin 0.5t$	1:52.7	Terminated on Step Size
Number of Harmonics = 1 $e(t) = 1 \sin 0.5t$	:13.3	Terminated on Step Size
Number of Harmonics = 1 $e(t) = 0.1 \sin 0.5t$	:22.4	Terminated on Step Size
Number of Harmonics = 1 $e(t) = -0.1 + 2 \sin t$	:24.7	$e(t) = -0.064 +$ $1.811 \sin 1.054t$

Table 4.2: HBA Results For System of Figure 4.1, Input At B .

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 1 e(t) = 0.1 sin 2t	:04.6	Amplitude = 0
Number of Harmonics = 1 e(t) = 4.0 sin 2t	:18.9	e(t) = 5.43 sin 4.00 t
Number of Harmonics = 1 e(t) = 0.5 sin 3t	:13.1	e(t) = 0.410 sin 4.00 t
Number of Harmonics = 1 e(t) = 10.0 sin 8t	:15.0	Amplitude = 0
Number of Harmonics = 3 e(t) = 4.0 sin 5t	:53.0	e(t) = 5.33 sin 4.09t − 0.026 cos 8.19 t + 0.024 sin 8.19 t + 0.488 cos 12.28 t + 0.239 sin 12.28 t

Table 4.3: HBA Results For System of Figure 4.2 With NL2a .

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 3	:52.1	$e(t) = 0.502 \sin 3.58\,t$
$e(t) = 1.0 \sin 2.5\,t$		$+ 0.12 \times 10^{-4} \cos 7.15\,t$
		$- 0.98 \times 10^{-5} \sin 7.15\,t$
		$+ 0.0336 \cos 10.72\,t$
		$- 0.0218 \sin 10.72\,t$

Table 4.4: HBA Results for System of Figure 4.2 with NL2b .

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 1 $e(t) = 0.4 \sin 2.5\,t$	:18.1	$e(t) = 0.467 \sin 3.91\,t$
Number of Harmonics = 3 $e(t) = 0.4 \sin 2.5\,t$	1:15.7	$e(t) = 0.454 \sin 4.04\,t$ $- 0.5\text{x}10^{-5} \cos 8.07\,t$ $+ 0.1\text{x}10^{-4} \sin 8.07\,t$ $+ 0.0228 \cos 12.11\,t$ $- 0.0230 \sin 12.11\,t$

Table 4.5: HBA Results For System of Figure 4.3.

INITIAL VALUES	COMPUTER TIME (MIN:SEC)	HBA RESULTS
Number of Harmonics = 1 $e(t) = 0.25 \sin 0.11\, t$	:28.0	$e(t) = 254.7 \sin 0.100\, t$
Number of Harmonics = 3 $e(t) = 0.25 \sin 0.1\, t$	2:08.1	$e(t) = 265.1 \sin 0.0932\, t$ $+ 0.00166 \cos 0.186\, t$ $- 0.00286 \sin 0.186\, t$ $+ 22.73 \cos 0.280\, t$ $- 13.84 \sin 0.280\, t$

Table 4.6: HBA Results For System of Figure 4.4.

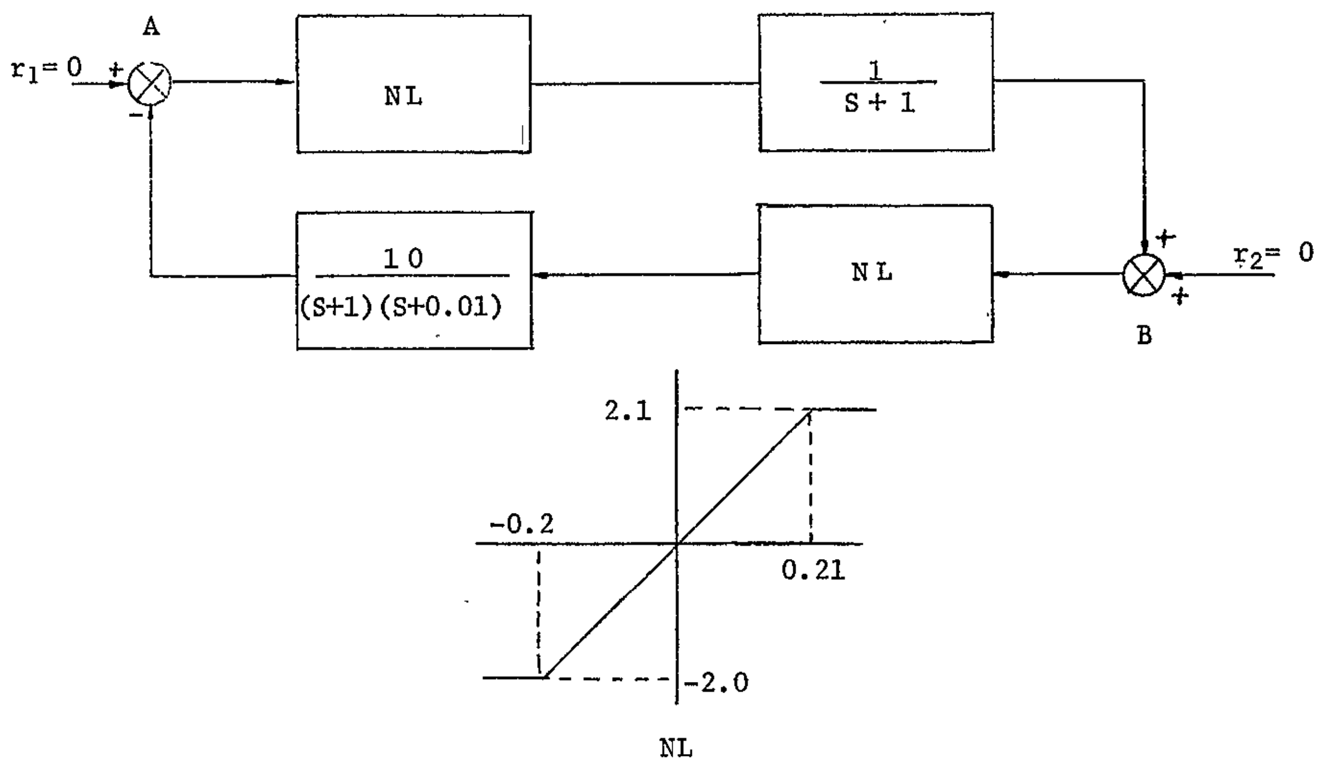


Figure 4.1. Single path system with two identical asymmetrical nonlinearities

more difficult with randomly selected values. A number of frequency and fundamental amplitude combinations resulted in termination based on the step size criterion. When a small dc component was added, however, a limit cycle was identified. Selecting the input as $\omega = 1.0$, DC = -0.1 , and fundamental = 2.0 resulted in convergence to $\omega = 1.054$, DC = -0.064, and fundamental = 1.811. (No harmonics were calculated in this example.) Since these two closed loop systems are identical except for the point chosen as the input, they should have identical limit cycle frequencies. Notice that the frequencies determined by the HBA agree within 5%.

Davison and Constantinescu [14] considered several examples in illustrating their method of limit cycle identification. These examples were also examined using the HBA. Limit cycles were identified in each case.

The systems of Figure 4.2 involves three nonlinearities in a single path; with NL2a in the system, several combinations of initial ω and fundamental amplitude were selected, ranging from ω = 2.0 to 8.0 and amplitude from 0.1 to 10.0 . When considering only the fundamental component (i.e., no harmonic components) limit cycles were identified at $\omega = 4.00$ rad/sec. Two limit cycle amplitudes were possible, 0.410 and 5.43. In general, for $\omega \geq 6.0$, or with amplitude less than or equal to 0.5 and $\omega = 2.0$, the HBA terminated with the amplitude equal to zero. This is, of course, a possible solution although it yields no useful information.

Limit cycle identification was also achieved when three harmonics were considered. An initial input of 5 rad/sec with a

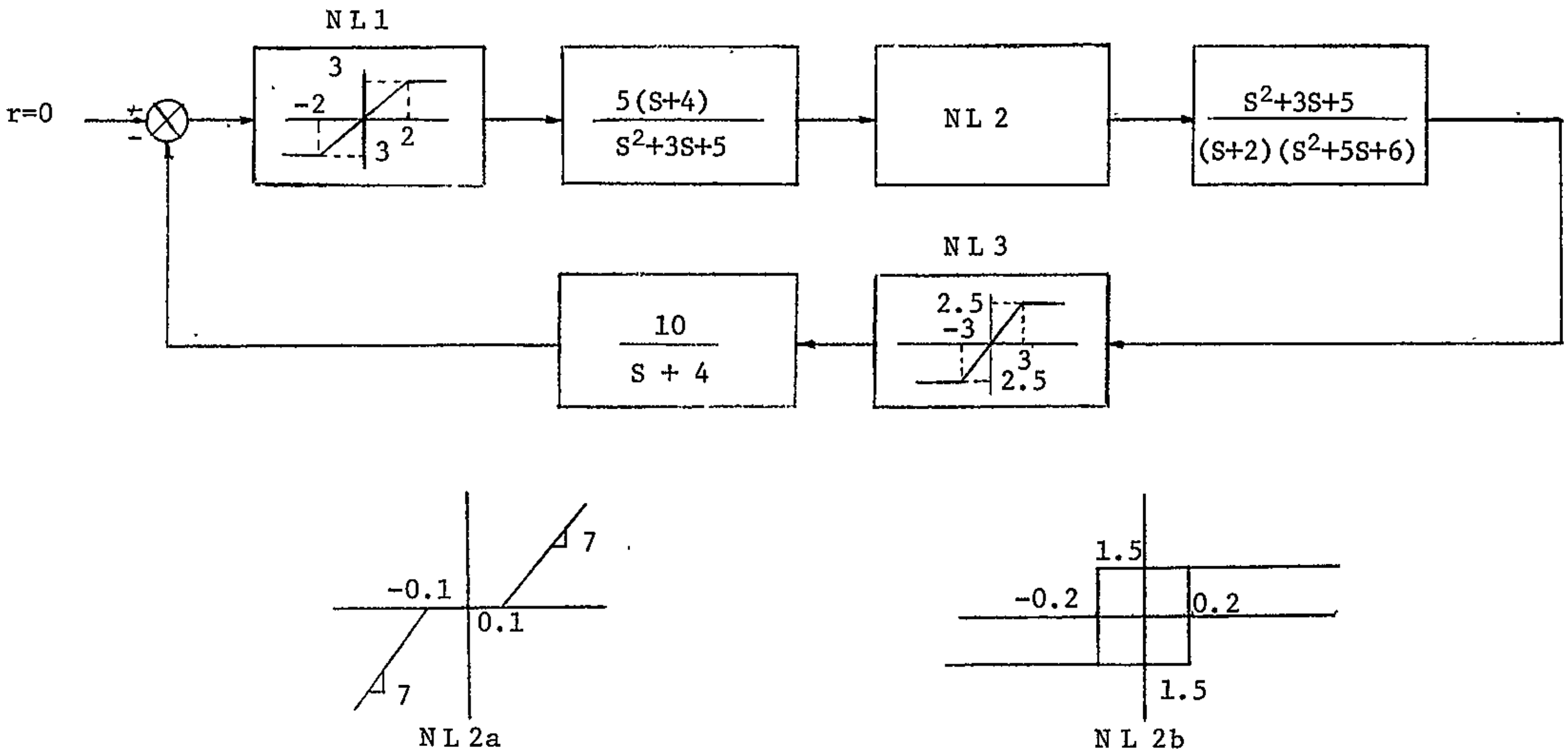


Figure 4.2. Single path, three nonlinearity system

fundamental amplitude of 4.0 produced a limit cycle frequency of 4.01 rad/sec. with a fundamental amplitude of 5.33 units. Second harmonic components were less than 1% of the fundamental component, but the third harmonic components were somewhat larger. A third harmonic cosine component of 0.488 and a sine component of 0.239 were identified.

The Davison and Constantinescu method identified a limit cycle at 4.00 rad/sec. with a fundamental amplitude of 0.58 or 5.32. While the frequency agrees with that determined by the HBA, the amplitudes are somewhat different. This may be due to inherent inaccuracies of graphical techniques as employed by Davison and Constantinescu.

The second example considered by Davison and Constantinescu replaces the second nonlinearity by a relay with hysteresis. (The hysteresis gap is $\pm$ 0.2 and the amplitude is $\pm$ 1.5.) A limit cycle with an amplitude of 0.50 units at 3.61 rad/sec. was identified. The HBA, considering three harmonic components, identified a limit cycle of 3.58 rad/sec. and a fundamental amplitude of 0.502 units. The second harmonic components were less than 0.01% of the fundamental, but the third harmonic components were larger; the third harmonic cosine amplitude was 0.0336 and the sine amplitude was -0.0218.

A modification of this example was also considered using the HBA. The first nonlinearity of Figure 4.2 was altered slightly in addition to the replacement of the second nonlinearity by the relay characteristics. The system of Figure 4.3 resulted. This produced a slightly different limit cycle, one with a frequency of 3.91 rad/sec and a fundamental amplitude of 0.467 units. When three harmonic components were considered, the limit cycle frequency shifted to 5.04 rad/sec. The resulting fundamental amplitude was 0.454 units. The second harmonic amplitudes were again negligible, but the third harmonic components were not, having amplitudes of 0.0228 and -0.0230 for the cosine and sine values respectively.

The last example considered was a parallel path system as shown in Figure 4.4. The HBA was applied for three harmonics, an initial frequency of 0.1 rad/sec., and a fundamental input component of 0.25 units. A limit cycle was identified at 0.0932 rad/sec. with a fundamental amplitude of 265.1 units. Second harmonic amplitudes were less than one unit, and third harmonic cosine and sine values were 22.73 and -13.84 respectively. Davison and Constantinescu identified the limit cycle at 0.1 rad/sec. with an amplitude of 254.8 units. When the HBA was limited to a single harmonic component, the limit cycle was determined at $\omega = 0.100$ with an amplitude of 254.7 units.

The agreement of the HBA results and the Davison-Constantinescu results support the validity of the HBA method, but the examples also show that harmonics can alter the limit cycle. In each case where multiple harmonics were considered, both the frequency and the fundamental amplitude were somewhat different from the single harmonic case.

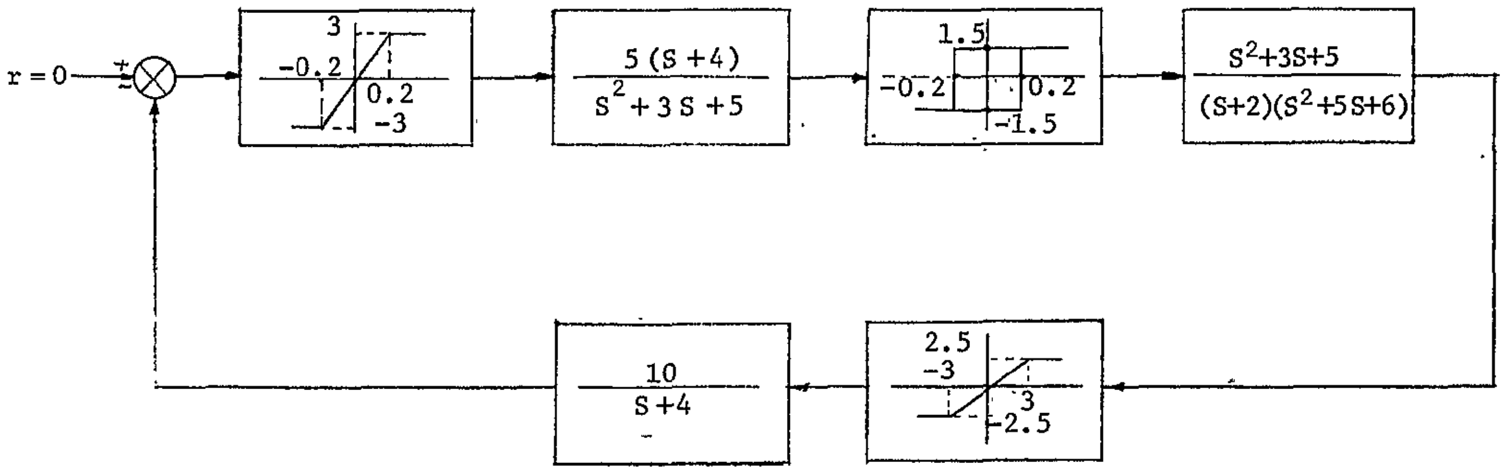


Figure 4.3. System of figure 4.2, modified

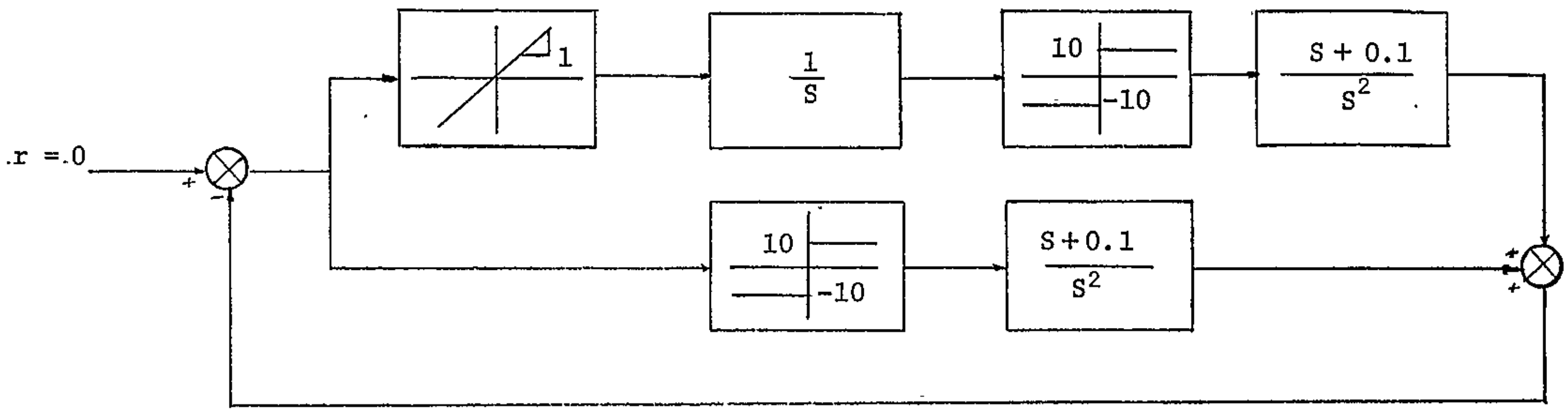


Figure 4.4. Multi-nonlinearity parallel path system

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Techniques for determining limit cycles in nonlinear systems rely heavily upon describing fuction analysis which considers only the fundamental component of the Fourier series representation of the nonlinear element output. Thus any contribution that may be made by other harmonic components is ignored. No practical method of analysis considered multiple harmonic components in systems containing several nonlinearities or forward paths. Since the effect of these components is not always negligible, a method of analysis was needed which included them. In an effort to meet this need, the Harmonic Balance Algorithm was developed.

The HBA can be used to identify limit cycles in multiple nonlinearity, multiple path systems. It cannot, however, be generally applied in a random manner. Random selection of initial input values may result in the identification of a zero amplitude limit cycle (a trivial case) or termination of the iterative process due to insufficient step sizes. Thus it appears that the initial input values should be intelligently selected, perhaps based upon the results of some other method (the Davison-Constantinescu techniques being the most general). In the event that a limit cycle is not identified by the describing function or other techniques, but indications are that one almost exists (i.e., the curves almost intersect), the HBA would be useful in determining if a limit cycle exists when harmonic components are present.

At any rate, the HBA cannot be profitably used by an individual unfamiliar with nonlinear system analysis. Its success depends in large measure upon the user's insight and ability to select the initial input values; it should not be considered a general purpose tool to be used to circumvent analysis using other available methods.

While the HBA sucessfully identifies limit cycles, its success is highly dependent upon the selection of the initial input values. While methods exist for determining possible limit cycles when only one or two harmonic components are considered, they are generally very tedious and somewhat error prone. Techniques for simplifying these methods, as well as new techniques, would be appropriate.

One specific area which could be examined is the Davison-Constantinescu method. A computer algorithm for generating the required curves or the data for plotting them would substantially enhance the attractiveness of the method. The limit cycle or near limit cycle values suggested by this method could then be used as starting values for the HBA in examining the effect of harmonics.

Further work may also prove fruitful in the area of automatic step size generation. In its present form the HBA may converge to a nonzero minimum of the input-output difference function. When this occurs, the algorithm terminates because further refinement of the input values will not produce the desired results. Some method of automatic input selection may be possible for these cases which would extricate the algorithm from these local minimum traps, allowing the zero finding process to continue.

Using the HBA as a base, additional coding may allow consideration of a more general system. In its present form the HBA does not allow cross-coupling between forward paths, i.e. feeding the output of an element in one path to the input of an element in another path. Also, no provision is made for internal feedback loops within a path. Both of these variations appear possible using the general concept upon which the HBA is based. Further efforts in this area should prove fruitful.

APPENDICES

APPENDIX A

HARMONIC BALANCE ALGORITHM USER'S GUIDE

The Harmonic Balance Algorithm (HBA) is intended to be a nonlinear system analysis tool. Its success depends upon the intelligent selection of input data; an individual untrained in nonlinear system theory is unlikely to be able to use it beneficially.

With one exception, data is entered in free format form; i.e., data entries are not sensitive to columnar location on the input data card. Multiple data entries on a single line or card are separated by a comma, denoting the end of one data field and the beginning of another.

The single exception is the first data card, which designates a number of iteration variables and limits. Blanks or zeroes in the fields defining these inputs will cause predetermined values to be selected. These values are defined in Table A.1.

The significance of input variable NFLAG requires some further explanation. Under normal use only a limited amount of output data is required. As shown in Appendix C, this includes the iteration number, the size of the largest pertubation used in calculating the partial derivatives in the J-matrix, the cost figures, the maximum input increment, and the new values of input to be used on the next iteration. The user may choose to output additional data by selecting larger values for NFLAG. Each larger number results in more intermediate data being printed, such as the J-matrix values, the outputs of the nonlinearities, the linear system gains, etc. When

VARIABLE	FORMAT	COLUMN	DEFINITION
DELTA	E	1 - 10	Perturbation used in calculating the J-Matrix (Default = 10^{-6})
NPMX	I	11-20	Number of points used to describe the inputs function (Default = 301)
ITERNO	I	21-30	Maximum number of iterations in calculating the output of nonlinear element (Default = 100)
EPS	E	31-40	Secant method stopping limit (Default = 10^{-5})
EE	E	41-50	Minimum step size limit for termination (Default = 10^{-8})
SIZE	E	51-60	Initial Maximum input increment (Default = 0.1)
NFLAG	I	61-65	Output Data selection (Default = 0)

Table A.1. Input Variables on First Data Card.

NFLAG is set to 3, all WRITE statements are used; intermediate calculations and subroutine entries and exits are indicated. The latter case may prove useful in helping one to understand the intricacies of the program.

The remaining input data can best be explained by use of an example. Each input variable is defined in Table A.2. and will be identified as the sample input data in Table A.3 is examined. Note that each input data card or line is numbered. These line numbers are not actually present on the input data card, but are included here for reference purposes. A flow chart depicting the data input process is shown in Figure A.1. An explanation of each line follows.

Card 1: DELTA, NPMX, ITERNO, EPS, EE, SIZE, NFLAG. This is the only input data not using free format. A blank or a zero for any value results in default selection.

Card 2: Number of harmonics (NOHAR) and number of parallel paths (NP).

Card 3: Initial guess at limit cycle frequency (WO).

Card 4: Initial guess at limit cycle amplitudes (HAR, entered in order of DC, fundamental, second harmonic cosine, second harmonic sine, third harmonic cosine, etc.).

Card 5: Number of nonlinearities (NNL) in each of the paths (path one, path two, etc.).

Card 6: Number of breakpoints (NBKPTS) for the nonlinearity and the nature of its symmetry (NLSYM). The number of breakpoints is one more than the actual number since an artificial point is inserted at the far left of the nonlinearity characteristic.

Card 7: JD, the nonlinearity curve number. (See Figure A.2.) This is always 1 for memoryless nonlinearities. If the nonlinearity is a memory type JD is 1 or 2 depending upon which curve is being examined.

Card 8: Abscissa coordinates for the nonlinearity breakpoints (EBKPT) beginning at the artificial left coordinate and

VARIABLE	DEFINITION
NOHAR	Number of harmonic terms to be considered in addition to DC term
NP	Number of parallel paths
WO	Initial guess at limit cycle radian frequency
HAR	Initial guess at input fourier series components
NNL	Number of nonlinearities in each path
NBKPTS	The number of breakpoints in the piecewise linear representation of the nonlinearity
NLSYM	Nonlinearity symmetry characteristic (0 = asymmetrical, 1 = symmetrical)
JD	Nonlinearity curve designation (used to identify curves on hysteresis type nonlinearities)
EBKPT	Nonlinearity breakpoint abscissa
UBKPT	Nonlinearity breakpoint ordinate
PWGAIN	Nonlinearity piecewise-linear slope
NH	Number of common breakpoints on hysteresis type nonlinearity
AB	Hysteresis nonlinearity common breakpoint abscissa
MORDER	Order of numerator of linear transfer function
NORDER	Order of denominator of linear transfer function
AA	Coefficients of linear transfer function numerator terms*
BB	Coefficients of linear transfer function denominator terms*
*of the form	$\dfrac{A_1S^n + A_2S^{n-1} + A_3S^{n-2} + \ldots + A_nS + A_{n+1}}{B_1S^m + B_2S^{m-1} + B_3S^{m-2} + \ldots + B_mS + B_{m+1}}$

Table A.2: Input Variables Using Free Format

Card or Line Number	Data					
1	(Card 1 was left blank in this example)					
2	3,	0				
3	2.5					
4	0.0,	1.0,	0.0,	0.0,	0.0,	0.0
5	3					
6	3,	1				
7	1					
8	-1.E+05,	-2.0,	2.0			
9	-3.0,	-3.0,	3.0			
10	0.0,	0.0,	1.5,	0.0		
11	0					
12	0					
13	1,	2				
14	5.0,	20.0				
15	1.0,	3.0,	5.0			
16	4,	1				
17	1					
18	-1.E+05,	-0.2,	0.2,	0.2		
19	-1.5,	-1.5,	-1.5,	1.5		
20	0.0,	0.0,	0.0,	1.E+20,	0.0	
21	2					
22	-1.E+05,	-0.2,	-0.2,	0.2		
23	-1.5,	-1.5,	1.5,	1.5		
24	0.0,	0.0,	1.E+20,	0.0,	0.0	
25	0					
26	2					
27	-0.2,	0.2				
28	2,	3				
29	1.0,	3.0,	5.0			
30	1.0,	7.0,	16.0,	12.0		
31	3,	1				
32	1					
33	-1.E+05,	-3.0,	3.0			
34	-2.5,	-2.5,	2.5			
35	0.0,	0.0,	0.83333,	0.0		
36	0					
37	0					
38	0,	1				
39	10.0					
40	1.0,	4.0				

Table A.3: Sample Input Data for Single-Path, Three Nonlinearity System With Hysteresis

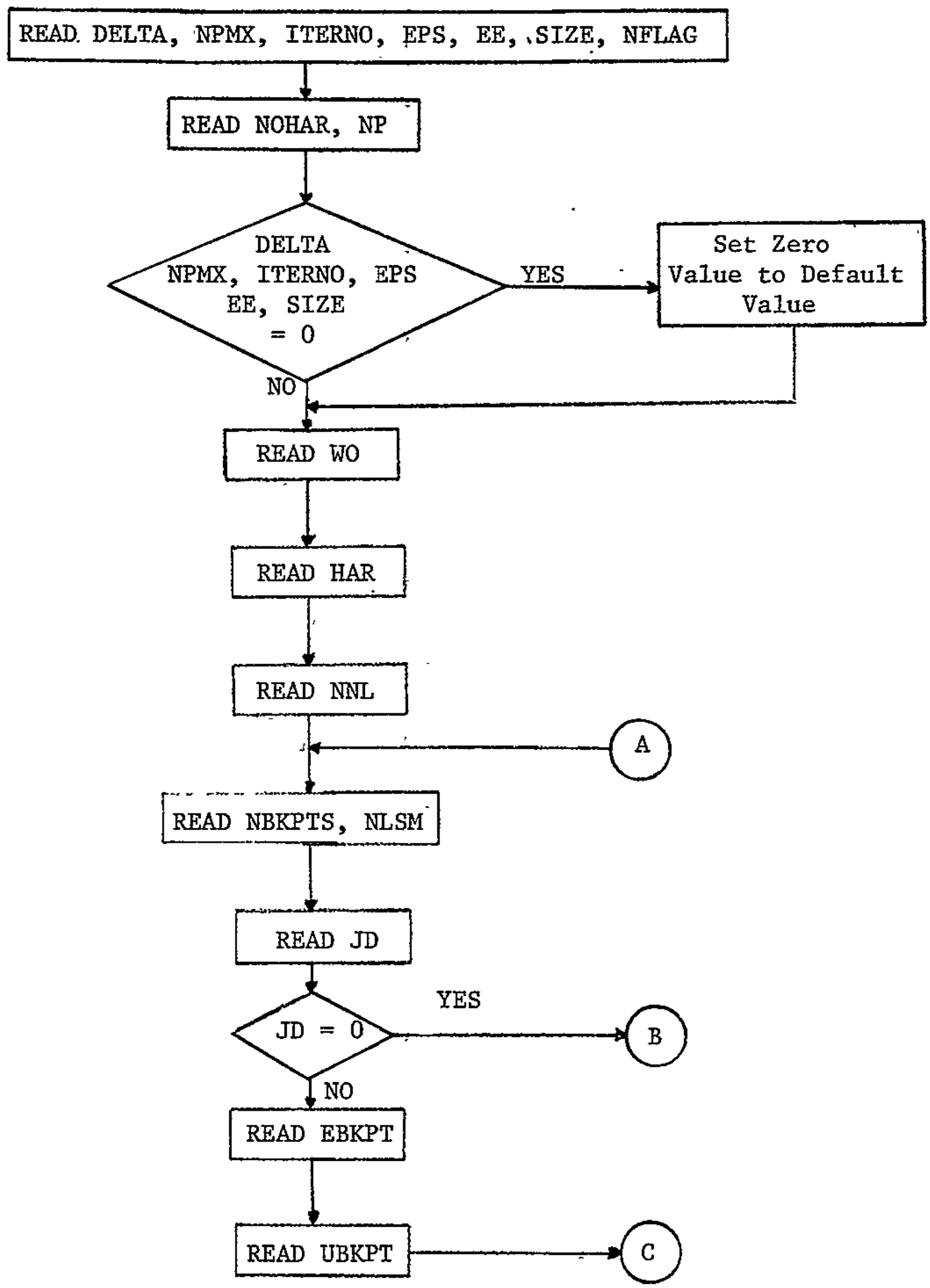


Figure A.1. Data input flow chart

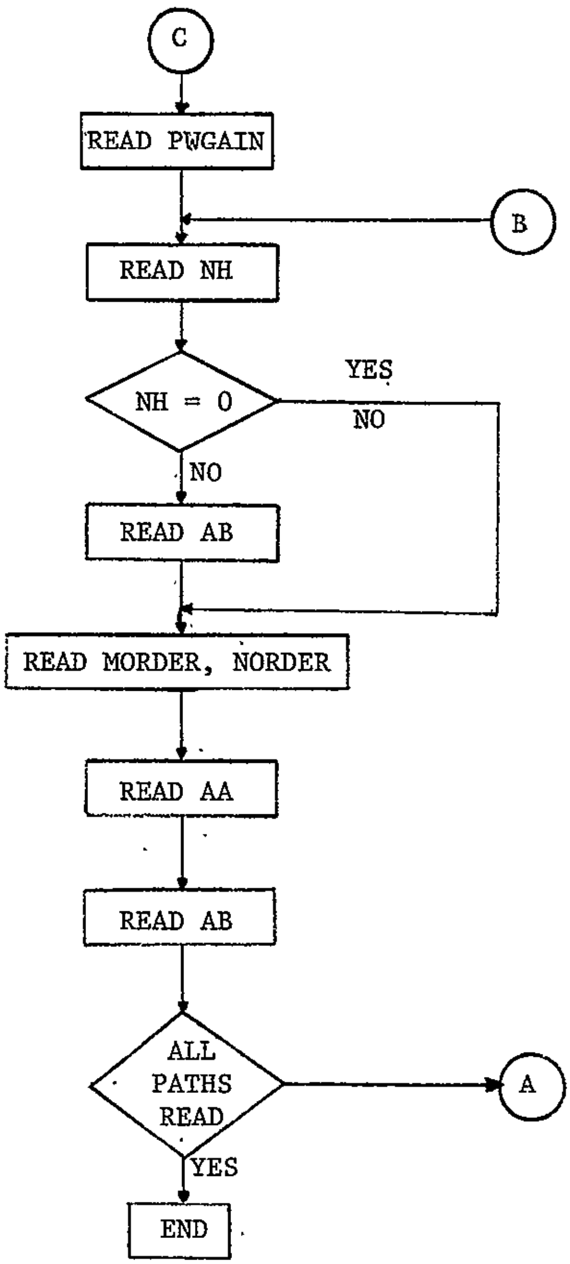


Figure A.1. (Continued) Data input flow chart progressing to the right. The artificial breakpoint is set far enough to the left so that it is never crossed.

Card 9: Ordinate coordinates for the nonlinearity (UBKPT), beginning at the artificial left breakpoint and proceeding to the right.

Card 10: Slope of the nonlinearity (PWGAIN) to the left of the associated breakpoint. There is always one more slope than breakpoint, the last being the slope to the right of the rightmost breakpoint. The slope to the left of the artificial breakpoint is set to zero. Vertical slopes are indicated by a very large number (i.e., 10^{20}).

Card 11: JD. If JD = 2, cards 8 - 10 are repeated for curve 2 of the hysteresis characteristic. JD = 0 indicates the nonlinearity has been fully described.

Card 12: NH, the number of hysteresis curve common point. NH = 0 for a memoryless type nonlinearity.

Card 13: Order of numerator (MORDER) and denominator (NORDER) of the linear transfer function.

Card 14: Coefficients of the numerator polynomial (AA) of the linear transfer function, with the coefficient of the highest order term first.

Card 15: Coefficients of the denominator polynomial (BB) of the linear transfer function, highest order term first.

This completes the entry of data for the first nonlinear-linear pair for path 1. If other pairs remain in path 1, the process (cards 6 through 15) is repeated until all path 1 data is entered. Data is then entered for each successive path until all data is entered.

Cards 16 - 20: Same as cards 6 - 10, but for nonlinearity 2, curve 1.

Card 21: JD = 2, thus breakpoint information for curve 2 of the hysteresis nonlinearity follows. (See Figure A.2)

Cards 22 - 24: Same as cards 8 - 10, but for nonlinearity 2, curve 2.

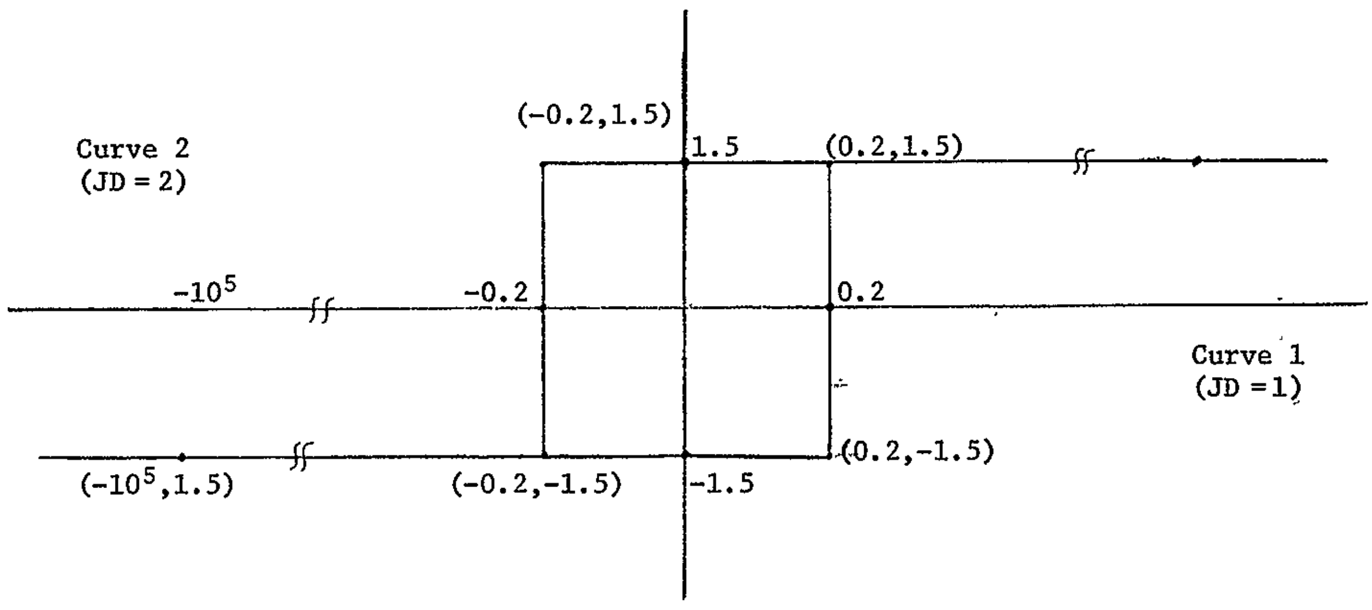


Figure A.2. Hysteresis type nonlinearity

Card 25:	JD = 0, so all breakpoint information for this nonlinearity has been entered.
Card 26:	Number of hysteresis curve common points (2).
Card 27:	Abscissa common breakpoints for the hysteresis curves (-0.2, 0.2).
Cards 28 - 30:	Same as cards 13 - 15, but for second linear element.
Cards 31 - 35:	Same as cards 6 - 10, but for third nonlinearity.
Card 36:	JD = 0, so all breakpoint information for third nonlinearity has been entered.
Card 37:	NH = 0, so no hysteresis common points exist.
Cards 38 - 40:	Same as cards 13 - 15, but for third linear element.

This completes the data entry for path 1, and since only one path is used in this example, all data has been entered. For multi-path examples, data entry for path 2 would follow card 40 and would follow the pattern begun at card 6.

Once the data entry phase is completed, no further user input is required. The HBA outputs identify possible limit cycles as determined from the initial inputs. Note that failure to identify a nontrivial limit cycle does not mean that no limit cycle exists; it simply means that, if one does exist, it has not yet been identified.

APPENDIX B

FORTRAN CODED LISTING OF HBA

```
C           INPUT DATA CONSIST OF INITIAL GUESS AT COEFFICIENTS OF
C           INPUTS (CN) AND FREQUENCY (WO) FOR HARMONIC BALANCE.
C           BREAKPOINTS AND SLOPES WHICH DEFINE THE NONLINEARITY,
C           AND COEFFICIENTS OF THE LINEAR ELEMENT POLYNOMIAL
C
      COMPLEX CNI(11),YNV(11),CNI(11)
      COMPLEX CNO(11),YNR(11)
      DIMENSION AJ(25,25), AJINV(25,25), DELT(25)
      DIMENSION JFLAG(25), DELTA(25), FY(25)
      COMMON NFLAG
      COMMON /MAINE/ CNI,DELTA,FF,SIZE,NOHAR /MN/ WO /TOMI/ JMAX
      COST1=1E10
      NSC=0
      DOLT=1E06
      CALL READ
      IF (NFLAG.GE.3) WRITE (6,530)
      K=0
      KCNI=1
      WRITE (6,950)
      WRITE (6,860) DELTA
      ILIM=(2*NOHAR)+1
      DO 10 J=1,JMAX
   10   IF (NFLAG.GE.3) WRITE (6,840) J,CNI(J)
      DO 20 J=1,ILIM
   20   DELT(J)=DELTA
C       OBTAIN OUTPUT VALUES FOR THE REF. INPUT
   30 CALL OUTPUT (YNR,CNI)
      IF (NFLAG.GE.3) WRITE (6,540)
      CALL FUNT ($520,FY,ILIM,YNR,JFLAG,CNI,JMAX,COST2,0)
      IF (NFLAG.GE.3) WRITE (6,550)
      WRITE (6,800) COST1,COST2
      IF (COST2.GT.COST1) GO TO 400
   40 DO 50 J=1,JMAX
   50   CNO(J)=CNI(J)
C        DETERMINE J MATRIX EXCEPT FOR LAST COLUMN BY
C        VARYING CN COMPONENTS ONE AT A TIME
      DO 270 K=1,JMAX
        IF (K.EQ.2) GO TO 120
   60   J=K
        IF (K.GT.2) J=(2*K)-3
        CNO(J)=CNI(K)+CMPLX(DELT(J),0.0)
        IF (NFLAG.GE.2) WRITE (6,540) K,CNI(K),K,CNO(K),J,DELT(J),K
C     OBTAIN OUTPUT VALUES FOR REAL PART OF INPUT INCREMENTED
        CALL OUTPUT (YNV,CNO)
        IF (NFLAG.GE.3) WRITE (6,570)
        IF (NFLAG.GE.2) WRITE (6,590) (YNR(NN),NN=1,JMAX)
        IF (NFLAG.GE.2) WRITE (6,590) (YNV(NN),NN=1,JMAX)
        DO 110 I=1,JMAX
          DIF=REAL(YNV(I))-REAL(YNR(I))
          DIV=ABS(REAL(YNV(I)))+ABS(REAL(YNR(I)))
          IF (DIV.EQ.0.0) GO TO 80
          IF (ABS(DIF/DIV).LT.0.005) GO TO 80
          DELT(J)=0.2*DELT(J)
          DO 70 J=1,JMAX
   70       CNO(J)=CNI(J)

          GO TO 60
   80     CONTINUE
          IF (I.EQ.1) M=1
          IF (I.GT.1) M=(2*I)-2
          AJ(M,J)=DIF/DELT(J)
          DIF=AIMAG(YNV(I))-AIMAG(YNR(I))
          DIV=ABS(AIMAG(YNV(I)))+ABS(AIMAG(YNR(I)))
          IF (DIV.EQ.0.0) GO TO 100
          IF (ABS(DIF/DIV).LT.0.005) GO TO 100
          DELT(J)=0.2*DELT(J)
          DO 90 J=1,JMAX
   90       CNO(J)=CNI(J)
          GO TO 60
  100     CONTINUE
          IF (I.EQ.1) GO TO [illegible]
```

ORIGINAL PAGE IS OF POOR QUALITY

```
            IF (I.GT.1) M=(2*I)-1
            AJ(M,J)=DIF/DELT(J)
  110       CONTINUE
  120     CONTINUE
          IF (K.EQ.1) GO TO 200
          DO 130 J=1,JMAX
  130       CND(J)=CNI(J)
  140     J=(2*K)-2
          CND(K)=CNI(K)+CMPLX(0.0,DELT(J))
          IF (NFLAG.GE.2) WRITE (6,560) (K,CNI(K),K,CND(K),J,DELT(J),WO)
C . OBTAIN OUTPUT VALUES FOR IMAGINARY PART OF INPUT INCR.
          CALL OUTPUT (YNV,CND)
          IF (NFLAG.GE.3) WRITE (6,590)
          IF (NFLAG.GE.2) WRITE (6,580) (YNR(NN),NN=1,JMAX)
          IF (NFLAG.GE.2) WRITE (6,580) (YNV(NN),NN=1,JMAX)
          DO 190 I=1,JMAX
            DIF=REAL(YNV(I))-REAL(YNR(I))
            DIV=ABS(REAL(YNV(I)))+ABS(REAL(YNR(I)))
            IF (DIV.EQ.0.0) GO TO 160
            IF (ABS(DIF/DIV).LT.0.025) GO TO 160
            DELT(J)=0.2*DELT(J)
            DO 150 J=1,JMAX
  150       CND(J)=CNI(J)
            GO TO 140
  160       CONTINUE
            IF (I.EQ.1) M=1
            IF (I.GT.1) M=(2*I)-2
            AJ(M,J)=DIF/DELT(J)
            DIF=AIMAG(YNV(I))-AIMAG(YNR(I))
            DIV=ABS(AIMAG(YNV(I)))+ABS(AIMAG(YNR(I)))
            IF (DIV.EQ.0.0) GO TO 180
            IF (ABS(DIF/DIV).LT.0.025) GO TO 180
            DELT(J)=0.2*DELT(J)
            DO 170 J=1,JMAX
  170         CND(J)=CNI(J)
            GO TO 140
  180       CONTINUE
            IF (I.EQ.1) GO TO 190
            IF (I.GT.1) M=(2*I)-1
            AJ(M,J)=DIF/DELT(J)
  190       CONTINUE
  200     CONTINUE
          DO 210 KK=1,JMAX
  210       CND(KK)=CNI(KK)
  220     CONTINUE
C          DETERMINE LAST COLUMN OF J MATRIX BY VARYING WO
  230 WO=WO+DELT(ILIM)
        DO 240 KK=1,JMAX
  240     CND(KK)=CNI(KK)
C     OBTAIN OUTPUT VALUES FOR INCR. IN WO
        CALL OUTPUT (YNV,CND)
        IF (NFLAG.GE.3) WRITE (6,600)
        IF (NFLAG.GE.2) WRITE (6,580) (YNR(NN),NN=1,JMAX)
        IF (NFLAG.GE.2) WRITE (6,580) (YNV(NN),NN=1,JMAX)
        WO=WO-DELT(ILIM)

        DO 270 MM=1,JMAX
          DIF=REAL(YNV(MM))-REAL(YNR(MM))
          IF (ABS(0.2*DELT(ILIM)/WO).LT.1.0E-07) GO TO 250
          DIV=ABS(REAL(YNV(MM)))+ABS(REAL(YNR(MM)))
          IF (DIV.EQ.0.0) GO TO 250
          IF (ABS(DIF/DIV).LT.0.025) GO TO 250
          DELT(ILIM)=0.2*DELT(ILIM)
          GO TO 230
  250     CONTINUE
          IF (MM.EQ.1) M=1
          IF (MM.GT.1) M=(2*MM)-2
          AJ(M,ILIM)=DIF/DELT(ILIM)
          DIF=AIMAG(YNV(MM))-AIMAG(YNR(MM))
          IF (ABS(0.2*DELT(ILIM)/WO).LT.1.0E-07) GO TO 260
          DIV=ABS(AIMAG(YNV(MM)))+ABS(AIMAG(YNR(MM)))
```

```
          IF (DIV.EQ.0.0) GO TO 250
          IF (ABS(DIF/DIV).LT.0.025) GO TO 260
          DELT(ILIM)=0.2*DELT(ILIM)
          GO TO 230
  260     CONTINUE
          IF (MW.EQ.1) GO TO 270
          IF (MW.GT.1) M=(2*MW)-1
          AJ(M,ILIM)=DIF/DELT(ILIM)
  270     CONTINUE
        DO 280 L=1,ILIM
  280     DELT(L)=1.5*DELT(L)
C         CHECK J MATRIX FOR ZERO ROWS
C            JFLAG IDENTIFIES WHICH ROWS ARE ZERO BY A 0
        AJ(1,1)=AJ(1,1)-1.
        DO 290 K=ILIM,3,-1
          AJ(K,K-1)=AJ(K,K-1)-1.
  290     CONTINUE
        DO 310 IM=1,ILIM
          BVAL=0.0
          DO 300 JM=1,ILIM
            BVAL=BVAL+ABS(AJ(IM,JM))
  300       CONTINUE
          JFLAG(IM)=1
          IF (IM.EQ.2) GO TO 310
          IF (BVAL.LT.1.E-10) JFLAG(IM)=0
  310     CONTINUE
        IF (NFLAG.GE.1) WRITE (6,770)
        DO 320 K=1,ILIM
          IF (NFLAG.GE.1) WRITE (6,660) K
  320     IF (NFLAG.GE.1) WRITE (6,650) (AJ(K,JV),JV=1,ILIM)
        IF (NFLAG.GE.1) WRITE (6,820) (DELT(I),I=1,ILIM)
C            ELIMINATE ROWS AND COLUMNS CORRESPONDING TO ZERO ROW SUBSCRI
        DO 350 K1=ILIM,1,-1
          IF (JFLAG(K1).EQ.1) GO TO 350
          DO 330 K2=1,ILIM
            DO 330 K3=K1,ILIM
              AJ(K2,K3)=AJ(K2,K3+1)
  330         CONTINUE
          DO 340 K3=1,ILIM
            DO 340 K2=K1,ILIM
              AJ(K2,K3)=AJ(K2+1,K3)
  340         CONTINUE
  350     CONTINUE
        KBIG=0
        DO 360 IBUMP=1,ILIM
C         KBIG IS THE NUMBER OF NON-ZERO ROWS
          KBIG=KBIG+JFLAG(IBUMP)
  360     CONTINUE
C         WRITE ADJUSTED J MATRIX
        IF (NFLAG.GE.1) WRITE (6,690)
        DO 370 IB=1,KBIG
          IF (NFLAG.GE.1) WRITE (6,660) IB
          IF (NFLAG.GE.1) WRITE (6,650) (AJ(IB,JVI),JVI=1,KBIG)
  370     CONTINUE
C        FIND INVERSE OF J MATRIX

         CALL MATINV (AJ,KBIG,AJINV,IER,25)
         IF (NFLAG.GE.3) WRITE (6,610)
         IF (IER.EQ.2) GO TO 460
         IF (NFLAG.GE.1) WRITE (6,760)
         DO 380 I=1,KBIG
           IF (NFLAG.GE.1) WRITE (6,670) I
           IF (NFLAG.GE.1) WRITE (6,700) (AJINV(I,JL),JL=1,KBIG)
  380      CONTINUE
C          CALCULATE FUNCTION F FOR USE IN FINDING INPUT DELTAS
         CALL FUNC (C496,FY,ILIM,YNR,JFLAG,CHT,DMR,UP,2,1)
         IF (NFLAG.GE.3) WRITE (6,520)
C        USING CALCULATED FUNCTION AND J INVERSE, DETERMINE DELTA
C          VALUES FOR THE INPUT
         CALL DELC (CALCH,AJINV,ILIM,FY,KBIG,JFLAG,D,SIMP)
         IF (NFLAG.GE.3) WRITE (6,630)
```

```
C       DETERMINE NEW VALUES OF CN FOR USE IN ITERATION
      ICOST=0
      COST1=COST2
      W1=WO
      DO 390 L=1,JMAX
  390   CN1(L)=CN(L)
      GO TO 420
  400 WRITE (6,780)
      IF (ICOST.GE.4) GO TO 40
      DO 410 K=1,ILIM
  410   DELCN(K)=DELCN(K)/5.
      D=D/5.
      KCNT=KCNT-1
      WRITE (6,790) KCNT
      ICOST=ICOST+1
  420 WO=W1-DELCN(ILIM)
      WRITE (6,720) WO
      CN(1)=CN1(1)-CMPLX(DELCN(1),0.0)
      DC=REAL(CN(1))
      WRITE (6,730) DC
      CN(2)=CN1(2)-CMPLX(0.0,DELCN(2))
      FUND=-2.*AIMAG(CN(2))
      WRITE (6,740) FUND
      IF (JMAX.LE.2) GO TO 440
      DO 430 KJ=3,JMAX
        KR=(2*KJ)-3
        KI=(2*KJ)-2
        CN(KJ)=CN1(KJ)-CMPLX(DELCN(KR),DELCN(KI))
        HARC=2.*REAL(CN(KJ))
        HARS=-2.*AIMAG(CN(KJ))
        JK=KJ-1
        WRITE (6,750) JK,HARC,HARS
  430   CONTINUE
  440 KCNT=KCNT+1
      IF (KCNT.GT.100) STOP
      YMAX=ABS(DELT(1))
      DO 450 K=1,ILIM
  450   YMAX=AMAX1(YMAX,ABS(DELT(J)))
      IF (D.LT.YMAX) ND=ND+1
      IF (D.GT.YMAX) ND=0
      IF (ND.GT.2) GO TO 480
      SIZE=2.*D
      IF (SIZE.LT.EE) NSC=NSC+1
      IF (SIZE.GT.EE) NSC=0
      IF (NSC.GT.4) GO TO 470
      WRITE (6,710) KCNT
      WRITE (6,810) SIZE
      GO TO 30
  460 WRITE (6,690)
      STOP
  470 WRITE (6,830)
      STOP
  480 WRITE (6,640)
      STOP
C
  490 WRITE (6,870) DC
      WRITE (6,880) FUND
      IF (JMAX.LE.2) GO TO 510
      DO 500 JJ=3,JMAX
        HARC=2.*REAL(CN(JJ))
        HARS=-2.*AIMAG(CN(JJ))
        JH=JJ-1
  500   WRITE (6,900) JH,HARC,HARS
  510 WRITE (6,890) WO
      STOP
  520 WRITE (6,910)
C
  530 FORMAT (20X,'REENTER MAIN FROM READ')
  540 FORMAT (20X,'REENTER MAIN FROM OUTPUT F')
  550 FORMAT (20X,'REENTER MAIN FROM FUNY')
```

```
560 FORMAT (3X,'CHI(',I3,')= ',2(G11.5,5X)/2X,'CNO(',I3,')= ',2(G11.5/
   1/2X,'DELT(',I3,')= ',G11.5,5X,'WO= ',G11.5)
570 FORMAT (20X,'REENTER MAIN FROM OUTPUT V, REAL VAL INCR')
580 FORMAT (5X,8(G11.5,3X))
590 FORMAT (20X,'REENTER MAIN FROM OUTPUT V, IMAG VAL INCR')
600 FORMAT (20X,'REENTER MAIN FROM OUTPUT V, W INCREMENTED')
610 FORMAT (20X,'REENTER MAIN FROM MATINV')
620 FORMAT (20X,'REENTER MAIN FROM FUN')
630 FORMAT (20X,'REENTER MAIN FROM DELC')
640 FORMAT (30X,'*** STEP SIZE SMALLER THAN PARTIAL INCREMENT ***')
650 FORMAT ('+',10X,8(E11.5,2X))
660 FORMAT (1X,'J(',I2,',-)= ')
670 FORMAT (2X,'JINV(',I2,',-)= ')
680 FORMAT (10X,'JINV DOES NOT EXIST')
690 FORMAT (//10X,'ADJUSTED J MATRIX',/)
700 FORMAT ('+',15X,8(G11.5,2X))
710 FORMAT (////,40X,'*** RUN NUMBER',I3,' FOLLOWS ***')
720 FORMAT (/40X,'NEW WO = ',G12.7)
730 FORMAT (/40X,'NEW DC COMPONENT = ',G12.7)
740 FORMAT (/40X,'NEW FUNDAMENTAL SINE COMPONENT = ',G12.7)
750 FORMAT (/40X,'NEW COMPONENTS FOR HARMONIC NO.',I2,': COSINE = ',G
   112.7,5X,'SINE = ',G12.7)
760 FORMAT (///)
770 FORMAT (30X,'J MATRIX')
780 FORMAT (/,5X,'***** FAILED COST TEST *****',/)
790 FORMAT (10X,'REPEAT CALCULATION FROM RUN ',I3)
800 FORMAT (20X,'*** COST 1 = ',G12.7,10X,' COST 2 = ',G12.7,'***')
810 FORMAT (40X,'NEW MAX. PARTIAL PERTURBATION = ',G12.7)
820 FORMAT (1X,' NEXT PARTIAL INCREMENT = ',8(E11.5,2X))
830 FORMAT (20X,' ** DELTA STEP SIZE CONSISTENTLY LESS THAN 10-8 **')
840 FORMAT (10X,'COMPLEX CHI(',I2,') = ',2(G12.7,5X))
850 FORMAT (////,40X,'RUN NUMBER 1 FOLLOWS')
860 FORMAT (40X,'MAX. PARTIAL PERTURBATION = ',G12.7)
870 FORMAT (//40X,' DC = ',G12.7)
880 FORMAT (//40X,'FUNDAMENTAL SINE COMPONENT = ',G12.7)
890 FORMAT (//40X,'LIMIT CYCLE FREQUENCY = ',G12.7,' RADIANS/SEC.')
900 FORMAT (//40X,'COMPONENTS FOR HARMONIC NO.',I2,': COSINE = ',G12.7
   1,5X,'SINE = ',G12.7)
910 FORMAT (/,30X,'RETURN ERROR FROM FUN')
C
      END
      FUNCTION E (T)
      COMPLEX CH(11),PCH
      COMMON /NOPT/ CH /OMN/ WO /POMN/ JMAX
      E=REAL(CH(1))
      DO 10 I=2,JMAX
        PCH=CH(I)*CEXP(CMPLX(0.,FLOAT(I-1)*WO*T))
   10   E=E+REAL(PCH*CONJG(PCH))
      RETURN
C
      END


C     SUBROUTINE OUTPUT DETERMINES THE OUTPUT OF A PARALLEL
C     SYSTEM WITH N PARALLEL PATHS AND M NONLINEARITIES.
C
      SUBROUTINE OUTPUT (YOUT,CHI)

      COMPLEX CHI(11),C(4,11),YOUT(11),CH(11)
      COMMON /OPT/ NP,NL(5) /NOPT/ CH /POMN/ JMAX
      COMMON NFLAG
      IF (NFLAG.GE.3) WRITE (6,100)
      DO 10 J=1,JMAX
        IF (NFLAG.GE.4) WRITE (6,110) CHI(J)
   10   YOUT(J)=CMPLX(0.0,0.0)
      NPP=1
   20 DO 30 J=1,JMAX
   30   CH(J)=CHI(J)
      NIT=1
   40 CONTINUE
      CALL OUT (C,NIT,NPP)
      IF (NFLAG.GE.3) WRITE (6,120)
      NIT=NIT+1
```

```
      IF (NIT.GT.NNL(NPP)) GO TO 60
      IF (NFLAG.GE.2) WRITE (6,140) NIT,NPP
      DO 50 J=1,JMAX
        IF (NFLAG.GE.2) WRITE (6,130) YN(J)
   50   CN(J)=YN(J)
      GO TO 40
   60 NPP=NPP+1
      DO 70 J=1,JMAX
   70   YOUT(J)=YOUT(J)+YN(J)
      IF (NPP.LE.NP) GO TO 20
      DO 80 J=1,JMAX
   80   YOUT(J)=(-1)*YOUT(J)
      DO 90 J=1,JMAX
   90   IF (NFLAG.GE.1) WRITE (6,150) J,YOUT(J)
      IF (NFLAG.GE.3) WRITE (6,110)
      RETURN
C
  100 FORMAT (20X,'ENTER OUTPUT')
  110 FORMAT (20X,'LEAVE OUTPUT')
  120 FORMAT (20X,'REENTER OUTPUT')
  130 FORMAT ()
  140 FORMAT (20X,'COMPLEX INPUT FOR NONLINEARITY',I3,' PATH',I3)
  150 FORMAT (10X,'COMPLEX YOUT(',I2,')=',2G15.7)
C
      END
```

```
      SUBROUTINE ORDER (TB,JINDXB,KINDXT,II)
      DIMENSION KCAT(500), KDOG(500), TB(500), JINDXB(500), KINDXT(500)
      DIMENSION TBX(500)
      COMMON NFLAG
      IF (NFLAG.GE.3) WRITE (6,90)
      DO 20 J=1,II
        DUMB=0.0
        DO 10 I=1,II
          IF (DUMB.GT.TB(I)) GO TO 10
          DUMB=TB(I)
          K=I
   10     CONTINUE
        KDOG(J)=K
   20   TB(K)=-TB(K)
      DO 30 I=1,II
        K=KDOG(I)
   30   TBX(II+1-I)=-TB(K)
      DO 40 I=1,II
   40   TB(I)=TBX(I)
      DO 50 I=1,II
        K=KDOG(I)
   50   KCAT(II+1-I)=JINDXB(K)
      DO 60 I=1,II
   60   JINDXB(I)=KCAT(I)
      DO 70 I=1,II
        K=KDOG(I)
   70   KCAT(II+1-I)=KINDXT(K)
      DO 80 I=1,II
   80   KINDXT(I)=KCAT(I)
      IF (NFLAG.GE.3) WRITE (6,100)
      RETURN
C
   90 FORMAT (20X,'ENTER ORDER')
  100 FORMAT (20X,'LEAVE ORDER')
C
      END
```

```
      SUBROUTINE OUT (YN,N,NPP)
C         SUBROUTINE OUT COMPUTES THE OUTPUT COEFFICIENTS YN
C         FOR A GIVEN SET OF INPUT COEFFICIENTS (FROM MAIN)
      COMPLEX DM(11),G,YN(11),A(11,5,5),B(11,5,5)
      COMMON /OT/ NBKPTS(5,5),EBKPT(25,2,5,5),UBKPT(25,2,5,5),PWGAIN(25,
     12,5,5),NLSYM(5,5),NH(5,5),AB(10,5,5),MORDER(5,5),NORDER(5,5),A,B
      COMMON NFLAG
      COMMON /OMN/ WO /POMN/ JMAX
C         NLOC FINDS THE OUTPUTS OF THE NONLINEAR ELEMENT
      IF (NFLAG.GE.3) WRITE (6,30)
      CALL NLOC (DM,EBKPT(1,1,N,NPP),UBKPT(1,1,N,NPP),PWGAIN(1,1,N,NPP),
     1NBKPTS(N,NPP),NH(N,NPP),AB(1,N,NPP),NLSYM(N,NPP))
      IF (NFLAG.GE.3) WRITE (6,40)
      YN(1)=0.0
      NK=NLSYM(N,NPP)+1
      DO 10 J=NK,JMAX
C       GJW CALCULATES THE VALUE OF THE LINEAR ELEMENT FOR THE
C       GIVEN VALUE OF W
        CALL GJW (G,J,WO,A(1,N,NPP),B(1,N,NPP),MORDER(N,NPP),NORDER(N,NP
     1P))
        IF (NFLAG.GE.3) WRITE (6,50)
C         THE OUTPUT YN VALUES ARE FOUND BY MULTIPLYING THE NONLINEAR ELE
C         OUTPUTS BY THE TRANSFER FUNCTION VALUE OF THE LINEAR ELEMENT.
        YN(J)=G*DM(J)
        IF (J.GT.6) GO TO 20
   10   CONTINUE
      IF (NFLAG.GE.2) WRITE (6,60) (DM(K),K=1,JMAX)
      IF (NFLAG.GE.3) WRITE (6,70)
      RETURN
   20 CONTINUE
      IF (NFLAG.GE.3) WRITE (6,80)
      STOP
C
C
   30 FORMAT (20X,'ENTER OUT')
   40 FORMAT (20X,'REENTER OUT')
   50 FORMAT (30X,'REENTER OUT')
   60 FORMAT (5X,'DM = ',11(G9.5,2X))
   70 FORMAT (20X,'LEAVE OUT')
   80 FORMAT (/,20X,'NUMBER OF HARMONIC TERMS EXCEEDS 5')
C
      END
```

```
C       INVERT THE X MATRIX
C
        SUBROUTINE MATINV (X,N,Y,IER,NMAX)
        DIMENSION X(NMAX,NMAX), Y(NMAX,NMAX), JC(25), V(2)
        COMMON NFLAG
        IF (NFLAG.GE.3) WRITE (6,70)
        IER=0
        DO 10 I=1,NMAX
          DO 10 J=1,NMAX
            Y(I,J)=X(I,J)
   10       CONTINUE
        V(1)=1.
        CALL GJR (Y,NMAX,NMAX,N,N,$20,JC,V)
        GO TO 30
   20   WRITE (6,40)
        WRITE (6,50) JC(1),V(1)
        IER=2
   30   IF (NFLAG.GE.3) WRITE (6,60)
        RETURN
C
   40   FORMAT (10X,'OVERFLOW')
   50   FORMAT (20X,2(G13.7,10X))
   60   FORMAT (40X,'LEAVE MATINV')
   70   FORMAT (40X,'ENTER MATINV')
C
        END
```

```
      SUBROUTINE HLOC (DM,EBKPT,UBKPT,PWGAIN,NBKPTS,NH,A,NSYM)
      DOUBLE PRECISION PI
      COMPLEX CN(11),ARGUE,ARG,DM(11),IM(11),JM(11),SUM1,SUM2,SUM3
      DIMENSION EIN(2001), JINDXB(500), KINDXT(500), ISW(500)
      DIMENSION TB(500), XK(500), BETA(500)
      DIMENSION EBKPT(25,2), UBKPT(25,2), PWGAIN(25,2), A(10)
      COMMON /HL/ NPMX,ITERNO,EPS /NOPT/ CN /OMN/ WO /POMN/ JMAX
      COMMON NFLAG
      ARG(NN,TZ)=CMPLX(0.0,FLOAT(NN)*WO*TZ)
      TX(J)=DELT*FLOAT(J-1)+0.001*DELT
      PI=3.141592653589793238
      PERIOD=2.*PI/ABS(WO)
      NSTP=NPMX-1
      NPWG=NBKPTS+1
      DELT=PERIOD/NSTP
      IF (NFLAG.GE.3) WRITE (6,440)
      IF (NFLAG.GE.3) WRITE (6,480) NBKPTS
      DO 10 I=1,2
        IF (NFLAG.GE.3) WRITE (6,480) (EBKPT(K,I),K=1,NBKPTS)
        IF (NFLAG.GE.3) WRITE (6,480) (UBKPT(K,I),K=1,NBKPTS)
        IF (NFLAG.GE.3) WRITE (6,480) (PWGAIN(K,I),K=1,NPWG)
   10   CONTINUE
      IF (NFLAG.GE.3) WRITE (6,480) (CN(I),I=1,JMAX)
      IF (NFLAG.GE.3) WRITE (6,480) NH,NSYM
      L=1
      XK(1)=0.
      BETA(1)=0.
      J=1
   20 IF (J-NPMX) 30,30,40
   30 T=TX(J)
      EIN(J)=E(T)
      J=J+1
      GO TO 20
   40 JFLAG=J
      TB(1)=0.0
      II=1
C       A(I) = ARRAY OF HYSTERESIS CURVE BREAKPOINTS
C       NH = NUMBER OF HYSTERESIS CURVE BREAKPOINTS
C       JSW=CONTROL FOR HYSTERESIS CURVE SELECTION
C
C      SELECT INITIAL NONLINEARITY CURVE
      JSW=1
      IF (EIN(2).LT.EIN(1)) JSW=2
      IF (WO.LE.0) JSW=1
C
C     CALCULATE ALL BREAKPOINT TIMES BY THE METHOD OF FALSE POSITION,
C     WHICH IS ALSO KNOWN AS THE SECANT METHOD.
C
      DO 90 JJ=2,NBKPTS
        JJ2=JJ
        IF (EIN(1).LT.EBKPT(NBKPTS,JSW)) GO TO 50
        JJ2=JJ+1
        JJ1=JJ+1
        GO TO 80
   50   CONTINUE
        IF (EIN(1).GT.EBKPT(1,JSW)) GO TO 60
        JJ1=JJ-1
        GO TO 80
   60   IF ((EIN(1)-EBKPT(JJ-1,JSW)).GT.0.0).AND.((EIN(1)-EBKPT(JJ,JSW)
     1).LT.0.)) GO TO 70
        GO TO 90
   70   JJ1=JJ
   80   XK(2)=PWGAIN(JJ1,JSW)
        BETA(1)=UBKPT(JJ2-1,JSW)-PWGAIN(JJ1,JSW)*EBKPT(JJ2-1,JSW)
   90   CONTINUE
      DO 250 KK=1,NSTP
C      DETERMINE BREAKPOINT TIMES
        KSW=0
        KSGN=+1
        DO 250 JJ=1,NBKPTS
          IF (KSW.EQ.1) GO TO 250
          IF ((EIN(KK)-EBKPT(JJ,JSW))*(EIN(KK+1)-EBKPT(JJ,JSW))) 100,100
     1,250
```

```
  100      IF (JJ.EQ.NBKPTS) GO TO 120
           IF ((EIN(KK)-EBKPT(JJ+1,JSW))*(EIN(KK+1)-EBKPT(JJ+1,JSW))) 110
     1,110,120
  110      KSGN=-1
           IF ((EIN(KK)-EIN(KK+1)).LT.0.) GO TO 250
  120      IF (EIN(KK).EQ.EBKPT(JJ,JSW)) GO TO 250
           ESW=1
           II=II+1
           JINDXB(II)=JJ+KSGN
           KINDXT(II)=KK
           ISW(II)=JSW
           KSGN=+1
           ITER=1
           L=L+1
           X1=TX(KK)
           X2=TX(KK+1)
           F1=EIN(KK)-EBKPT(JJ,JSW)
           F2=EIN(KK+1)-EBKPT(JJ,JSW)
           IF (NFLAG.GE.3) WRITE (6,430)
           IF (NFLAG.GE.3) WRITE (6,470) KK,JJ,II,X1,X2,F1,F2,EIN(KK),EIN
     1(KK+1),EBKPT(JJ,JSW)
           IF (NFLAG.GE.3) WRITE (6,480) JSW
           IF (F1.EQ.F2) STOP
  130      X3=X2-(X2-X1)*F2/(F2-F1)
           IF (F2.EQ.0.0) GO TO 190
           F3=E(X3)-EBKPT(JJ,JSW)
           IF (ABS(X3-X1)-EPS) 190,140,140
  140      IF (ABS(X3-X2)-EPS) 190,150,150
  150      IF (F1*F3) 170,160,180
  160      IF (ABS(F3).LT.EPS) GO TO 190
  170      F2=F3
           X2=X3
           ITER=ITER+1
           IF (ITER.EQ.ITERNO) GO TO 190
           GO TO 130
  180      F1=F3
           X1=X3
           ITER=ITER+1
           IF (ITER.EQ.ITERNO) GO TO 190
           GO TO 130
  190      TB(L)=X3
C
C      DETERMINE WHICH CURVE TO USE WHEN HYSTERESIS BREAKPOINTS
C      ARE CROSSED
C
           IF (KK.EQ.1) GO TO 240
           IF (NH.LE.0) GO TO 240
           IF (EIN(KK+1).GT.A(1)) GO TO 200
           JSW=1
           GO TO 240
  200      IF (EIN(KK+1).LT.A(NH)) GO TO 210
           JSW=2
           GO TO 240
  210      CONTINUE
           N1=(NH-1)
           DO 220 I=1,N1
  220        IF ((EIN(KK+1).GE.A(I)).AND.(EIN(KK+1).LE.A(I+1))) GO TO 230
           STOP
  230      IF (EIN(KK).LE.A(I)) JSW=1
           IF (EIN(KK).GE.A(I+1)) JSW=2
  240      CONTINUE
           IF (II-NSTP) 250,250,260
  250      CONTINUE
C
        II=II+1
        ISTOP=II-1
  260 TB(II)=PERIOD+0.001*DELT
C     ORDER PLACES THE BREAKPOINTS IN THE CORRECT TIME SEQUENCE
        CALL ORDER (TB,JINDXB,KINDXT,II)
C
C       THIS DO LOOP KEEPS TRACK OF WHICH PIECEWISE-LINEAR GAIN
C       GOES WITH EACH BREAKPOINT TIME.
```

```
C
      DO 310 MX=2,II
        MZ=MX-1
        IF (MX-II) 270,300,300
  270   JX=IABS(JINDXB(MX))
        KX=KINDXT(MX)
C   .  DETERMINE PROPER NONLINEARITY CURVE
        JSW=ISW(MX)
C      MATCH GAIN AND BREAKPOINT
        INC=0
        EDOT=EIN(KX+1)-EIN(KX)
        IF ((JSW.EQ.2).AND.(JINDXB(MX).LT.0)) INC=1
        IF ((JSW.EQ.1).AND.(JINDXB(MX).LT.0)) INC=-1
        IF ((EDOT.LT.0.).AND.(JINDXB(MX).LT.0.).AND.(NH.EQ.0)) INC=1
        IF ((EDOT.GT.0.).AND.(JINDXB(MX).LT.0.).AND.(NH.EQ.0)) INC=-1
        JX=JX+INC
        IF (EDOT) 290,420,280
  280   JX=JX-1
  290   BETA(MZ)=UBKPT(JX,JSW)-EBKPT(JX,JSW)*PWGAIN(JX+1,JSW)
        XK(MX)=PWGAIN(JX+1,JSW)
        GO TO 310
  300   BETA(MZ)=BETA(1)
        XK(MX)=XK(2)
  310   CONTINUE
      IF (MFLAG.GE.3) WRITE (6,460) (BETA(I),XK(I),I=1,II)
C
C     USE BREAKPOINT TIMES AND PIECEWISE-LINEAR GAIN INFORMATION TO
C     INTEGRATE TO GET ALL DM(I).
C
      DO 410 MM=1,JMAX
      M=MM-1
      IM(MM)=CMPLX(0.0,0.0)
      IF (M.EQ.0) GO TO 330
      DO 320 I=1,ISTOP
  320   IM(MM)=IM(MM)+BETA(I)*(CEXP(ARG(-M,TB(I+1)))-CEXP(ARG(-M,TB(I)
     1)))
      IM(MM)=IM(MM)+CMPLX(0.0,1./(2.*PI*FLOAT(M)))
      GO TO 350
  330 DO 340 I=1,ISTOP
  340   IM(MM)=IM(MM)+BETA(I)*(TB(I+1)-TB(I))
      IM(MM)=IM(MM)/PERIOD
  350 JM(MM)=CMPLX(0.0,0.0)
      DO 400 I=1,ISTOP
        SUM1=CMPLX(0.0,0.0)
        SUM2=CMPLX(0.0,0.0)
        SUM3=CMPLX(0.0,0.0)
        SUM1=CN(1)*(TB(I+1)-TB(I))
        DO 380 N=2,JMAX
          IF (M.EQ.N-1) GO TO 360
          ARGUE=CN(N)*(CEXP(ARG(N-M-1,TB(I+1)))-CEXP(ARG(N-M-1,TB(I))))
     1)/CMPLX(0.0,WO*FLOAT(N-M-1))
          GO TO 370
  360     ARGUE=CN(N)*(TB(I+1)-TB(I))
  370     SUM2=SUM2+ARGUE
  380     CONTINUE
        DO 390 N=2,JMAX
  390     SUM3=SUM3+CONJG(CN(N))*(CEXP(ARG(-N-M+1,TB(I+1)))-CEXP(ARG(-
     1N-M+1,TB(I))))/CMPLX(0.0,WO*FLOAT(-N-M+1))
  400   JM(MM)=JM(MM)+(SUM1+SUM2+SUM3)*XK(I+1)/PERIOD
      DM(MM)=IM(MM)+JM(MM)
  410 CONTINUE
     IF ((NSYM.EQ.1).AND.(ABS(REAL(CN(1))).LT.1E-10)) DM(1)=0.0
     IF (MFLAG.GE.3) WRITE (6,450)
     RETURN
  420 STOP
C
  430 FORMAT (T7,'KK',T18,'JJ',T30,'II',T42,'X1',T54,'X2',T64,'F1',T76,'
     1F2',T80,'EIN(KK)',T92,'EIN(KK+1)',T104,'EBKPT(JJ)')
  440 FORMAT (20X,'ENTER NLOC')
  450 FORMAT (20X,'LEAVE NLOC')
  460 FORMAT (/,(2('  BETA = ',1F12.7,10X,'XK = ',1F12.7,10X)))
  470 FORMAT (' ',1X,3(I3,2X),4(F12.7,3X),3(F12.7,3X))
  480 FORMAT (/)
C
      END
```

```
      SUBROUTINE FUNY (*,FY,JLIM,YN,JFLAG,CN,JMAX,COST,MF)
      COMPLEX CN(11),YN(11)
      COMMON NFLAG
      DIMENSION FY(25), JFLAG(25)
      IF (NFLAG.GE.3) WRITE (6,120)
      DO 10 I=1,JMAX
        IF (NFLAG.GT.3) WRITE (6,90) I,YN(I),I,CN(I)
   10   CONTINUE
C     FY IS THE DIFFERENCE BETWEEN THE OUTPUT AND THE ASSUMED INPUT
      FY(1)=(REAL(YN(1))-REAL(CN(1)))
      DO 20 K=2,JLIM
        L=(2*K)-2
        M=(2*K)-1
        FY(L)=(REAL(YN(K))-REAL(CN(K)))
        FY(M)=(AIMAG(YN(K))-AIMAG(CN(K)))
   20   CONTINUE
      COST=0.0
      DO 30 I=1,JLIM
   30   COST=COST+ABS(FY(I))
      IF (MF.EQ.0) RETURN
      INDEX=0
      DO 40 KF=1,JLIM
        IF (JFLAG(KF).EQ.0) INDEX=INDEX+1
        FY(KF)=FY(KF+INDEX)
   40   CONTINUE
      LIM=JLIM-INDEX
      KFLAG=0
C     COMPUTE THE NORMALIZING FACTOR FOR THE OUTPUT
      YNOR=0.0
      DO 50 K=1,JMAX
        YNOR=YNOR+(REAL(YN(K)))**2
        YNOR=YNOR+(AIMAG(YN(K)))**2
   50   CONTINUE
      YNOR=SQRT(YNOR)
      IF (YNOR.LE.1.0E-08) GO TO 80
C     CHECK FOR ERROR WITHIN SATISFACTORY LIMITS
      DO 60 KK=1,LIM
        IF ((ABS(FY(KK))/YNOR).GT.0.001) KFLAG=1
   60   CONTINUE
      IF (KFLAG.EQ.0) GO TO 70
      RETURN
   70 WRITE (6,100)
      WRITE (6,130)
      RETURN 1
   80 WRITE (6,110)
      STOP
C
   90 FORMAT (20X,'YN(',I2,')=',2(F12.6,5X),2X,'CN(',I2,')=',2(F12.6,5X)
     1)
  100 FORMAT (//,20X,'ERROR LIMIT SATISFIED')
  110 FORMAT (20X,'ALL OUTPUTS ARE ZERO')
  120 FORMAT (20X,'ENTER FUNY')
  130 FORMAT (20X,'ACCEPTABLE VALUES ARE')
C
      END
```

```
C     SUBROUTINE READ COLLECTS INPUT DATA.
C     INPUT DATA CONSISTS OF INITIAL GUESSES AT INPUT (CN)
C     COEFFICIENTS AND FREQUENCY (WO) FOR HARMONIC BALANCE,
C     BREAKPOINTS AND SLOPES WHICH DEFINE THE NONLINEARITY,
C     AND COEFFICIENTS OF THE LINEAR ELEMENT POLYNOMIALS.
C     ALSO INCLUDED ARE THE NUMBER OF PARALLEL PATHS AND THE
C     NUMBER AND FORM OF NONLINEAR ELEMENTS IN EACH PATH.
C
C
C         NOHAR = NUMBER OF HARMONICS ( EXCLUDING DC )
C         JMAX = NOHAR + 1 = NUMBER OF HARMONICS INCLUDING DC
C         DELTA IS THE PERTURBATION FOR FINDING J
C         JD IS THE CURVE SELECTOR
C         AB =  COMMON BREAKPOINTS ON HYSTERESIS CURVE
C         NLSYM=0 FOR ASYM NL, 1 FOR SYM NL
C         ITERNO= MAX. NO. OF ITERATIONS FOR NL OUTPUT
C         EPS= SECANT METHOD STOPPING LIMIT
C         NPMX= NO.OF TIME POINTS USED TO DESCRIBE INPUT FUNCTION
C         EE= MINIMUM STEP SIZE FOR TERMINATION
C         NORDER, MORDER= ORDER OF DENOM. AND NUMER. OF LIN. EL.
C       SIZE DETERMINES THE LARGEST CHANGE IN CN WHICH CAN OCCUR.
C
C
      SUBROUTINE READ
      COMPLEX CNI(11),A(11,5,5),B(11,5,5)
      DIMENSION HAR(22), HA(11), BB(11)
      COMMON NFLAG
      COMMON /MAINE/ CNI,DELTA,EE,SIZE,NOHAR
      COMMON /OMN/ WO /POMN/ JMAX /NL/ NPMX,ITERNO,EPS
      COMMON /OPT/ NP,NNL(5)
      COMMON /OT/ NBKPTS(5,5),EBKPT(25,2,5,5),UBKPT(25,2,5,5),PWGAIN(25,
     .12,5,5),NLSYM(5,5),NH(5,5),AB(10,5,5),MORDER(5,5),NORDER(5,5),A,B
      WRITE (6,110)
      READ (5,180) DELTA,NPMX,ITERNO,EPS,EE,SIZE,NFLAG
      READ (5,140) NOHAR,NP
      IF (DELTA.EQ.0.) DELTA=1.E-06
      IF (NPMX.EQ.0) NPMX=301
      IF (ITERNO.EQ.0) ITERNO=100
      IF (EPS.EQ.0.) EPS=1.E-05
      IF (SIZE.EQ.0.) SIZE=0.1
      IF (NP.EQ.0) NP=1
      IF (EE.EQ.0) EE=1.E-08
      IF (NFLAG.GE.3) WRITE (6,120)
      WRITE (6,160)
      WRITE (6,380) DELTA,NPMX,ITERNO,EPS
      WRITE (6,400)
      WRITE (6,390) NFLAG,EE,SIZE
      JMAX=NOHAR+1
      IND=2*NOHAR
      WRITE (6,150) JMAX
      READ (5,140) WO
      READ (5,140) (HAR(J), J=1,IND)
      READ (5,140) (NNL(I),I=1,NP)
      WRITE (6,170) NOHAR
      WRITE (6,190) WO
      WRITE (6,200) (NP,(NNL(I),I=1,NP))
      WRITE (6,210)
      WRITE (6,230) HAR(1)
      WRITE (6,220)
      WRITE (6,240) HAR(2)
      IF (NOHAR.EQ.1) GO TO 20
      DO 10 IT=2,NOHAR
10      WRITE (6,250) IT,HAR(2*IT-1),HAR(2*IT)
20    DO 80 KP=1,NP
        NL=NNL(KP)
```

```
      DO 80 K=1,NL
        READ (5,140) NBKPTS(K,KP),NLSYM(K,KP)
        WRITE (6,260) K,KP,NBKPTS(K,KP)
        IF (NLSYM(K,KP).EQ.0) WRITE (6,270) K,KP
        IF (NLSYM(K,KP).EQ.1) WRITE (6,280) K,KP
        IOP=NBKPTS(K,KP)+1
        KK=NBKPTS(K,KP)
 30     READ (5,140) JD
        IF (JD.EQ.0) GO TO 40
        WRITE (6,290) JD,K,KP
        READ (5,140) (EBKPT(I,JD,K,KP),I=1,KK)
        READ (5,140) (UBKPT(I,JD,K,KP),I=1,KK)
        READ (5,140) (PWGAIN(I,JD,K,KP),I=1,IOP)
        WRITE (6,300) (EBKPT(I,JD,K,KP),I=1,KK)
        WRITE (6,310) (UBKPT(I,JD,K,KP),I=1,KK)
        WRITE (6,320) (PWGAIN(I,JD,K,KP),I=1,IQP)
        GO TO 30
 40     READ (5,140) NH(K,KP)
        NX=NH(K,KP)
        IF (NH(K,KP).EQ.0) GO TO 50
        WRITE (6,330) NH(K,KP)
        READ (5,140) (AB(I,K,KP),I=1,NX)
        WRITE (6,340) (AB(I,K,KP),I=1,NX)
 50     CONTINUE
        READ (5,140) MORDER(K,KP),NORDER(K,KP)
        WRITE (6,350) MORDER(K,KP),NORDER(K,KP),K,KP
        MTERMS=MORDER(K,KP)+1
        NTERMS=NORDER(K,KP)+1
        READ (5,140) (AA(J),J=1,MTERMS)
        READ (5,140) (BB(J),J=1,NTERMS)
        WRITE (6,360) (AA(J),J=1,MTERMS)
        WRITE (6,370) (BB(J),J=1,NTERMS)
        DO 60 J=1,MTERMS
 60       A(J,K,KP)=CMPLX(AA(J),0.0)
        DO 70 J=1,NTERMS
 70       B(J,K,KP)=CMPLX(BB(J),0.0)
 80     CONTINUE
      CHI(1)=CMPLX(HAR(1),0.0)
      CHI(2)=CMPLX(0.0,(HAR(2))/-2.)
      IF (NOHAR.EQ.1) GO TO 100
      DO 90 N=3,JMAX
        M=2*N
 90     CHI(N)=CMPLX((HAR(M-3))/2.,(HAR(M-2)/(-2.)))
100   CONTINUE
      WRITE (6,410)
      IF (NFLAG.GE.3) WRITE (6,130)
      RETURN
C
110   FORMAT (1H1,////10X,10('* '),'I N P U T     D A T A ',10(' *'))
120   FORMAT (20X,'ENTER READ')
130   FORMAT (20X,'LEAVE READ')
140   FORMAT ()
150   FORMAT (/10X,'JMAX = ',I2)
160   FORMAT (/////,5X,'DELTA',T15,'NO. POINTS',T29,'NL ITER',T48,'SEC.
     1STOP LIM')
170   FORMAT (//10X,'NO. OF HARMONICS =',I1)
180   FORMAT (E10.5,2I10,3E10.5,I5)
190   FORMAT (//10X,'INITIAL W =',G12.7)
200   FORMAT (//10X,'NUMBER OF PATHS =',I1/,2X,'NUMBER OF',' NONLINEARIT
     1IES FOR PATH =',5I4)
210   FORMAT (//20X,'INITIAL INPUT VALUES')
220   FORMAT (//T25,'HARMONIC VALUES',T45,'HAR. NO.',T57,'COSINES',T72,'
     1SINES')
230   FORMAT (//20X,'DC =',G12.7)
240   FORMAT (/20X,'FUNDAMENTAL COMP.',T70,G12.7)
250   FORMAT (/T47,I2,T55,G12.7,T70,G12.7)
260   FORMAT (//1X,'NO. OF BREAKPOINTS IN NONLINEARITY',I2,' OF PATH',I2
     1,' IS ',I1)
270   FORMAT (//1X,'NONLINEARITY',I2,' OF PATH',I2,' IS ASSYMETRICAL')
280   FORMAT (//1X,'NONLINEARITY',I2,' OF PATH',I2,' IS SYMETRICAL')
```

```
290 FORMAT (//1X,'BREAKPOINT VALUES FOR CURVE',I2,' OF NONLINEARITY',I
   1 2,' IN PATH',I2)
300 FORMAT (/,5X,'HORIZONTAL COORDINATES',T30,8(E10.5,5X))
310 FORMAT (/5X,'VERTICAL COORDINATES',T30,8(E10.5,5X))
320 FORMAT (//3X,'SLOPE OF SEGMENT TO LEFT',T30,8(E10.5,5X))
330 FORMAT (//10X,'NUMBER OF COMMON POINTS ON HYS.CURVES =',I2)
340 FORMAT (//2X,'HORIZONTAL COORDINATES OF COMMON POINTS',3X,8(E10.5,
   14X))
350 FORMAT (//2X,I2,' ORDER NUMERATOR,',I2,' ORDER DENOM.',' IN NONLIN
   1EARITY',I2,' IN PATH',I2)
360 FORMAT (//2X,'COEF.OF NUM. TERMS(HIGHEST FIRST)',8(5X,E10.5))
370 FORMAT (//2X,'COEF.OF DEN. TERMS(HIGHEST FIRST)',8(5X,E10.5))
380 FORMAT (1X,E10.5,2I10,T50,E10.5)
390 FORMAT (1X,I10,T25,E10.5,T50,E10.5)
400 FORMAT (//5X,'PRINT FLAG',T25,'MIN. STEP SIZE',T45,'INIT. STEP')
```

```
C     FIND THE TRANSFER MAGNITUDE AND PHASE OF THE LINEAR SYSTEM DESCRIBE
C      BY A AND B AT THE FREQUENCY WO
C
      SUBROUTINE GJW (G,J,WO,A,B,MORDER,NORDER)
      COMPLEX G,Z,P,A(11),B(11)
      COMMON NFLAG
      IF (NFLAG.GE.3) WRITE (6,50)
      JL=J-1
      Z=A(1)
      IF (MORDER.EQ.0) GO TO 20
      DO 10 MZ=1,MORDER
 10    Z=Z+CMPLX(0.0,FLOAT(JL)*WO)*A(MZ+1)
 20 CONTINUE
    P=B(1)
    IF (NORDER.EQ.0) GO TO 40
    DO 30 NP=1,NORDER
 30    P=P+CMPLX(0.0,FLOAT(JL)*WO)*B(NP+1)
 40 CONTINUE
    IF (CABS(P).EQ.0.0) P=P+CMPLX(1.E-30,0.)
    G=Z/P
    IF (REAL(G).EQ.0.0) G=G+CMPLX(1.E-30,0.)
    IF (NFLAG.GE.2) WRITE (6,60) G
    IF (NFLAG.GE.3) WRITE (6,70)
    RETURN
C
C
 50 FORMAT (20X,'ENTER GJW')
 60 FORMAT (30X,'G(REAL) =',G15.7,10X,'G(IMAG) =',G15.7)
 70 FORMAT (20X,'LEAVE GJW')
C
    END
```

```
C       DETERMINE THE CHANGE TO BE MADE IN THE ASSUMED INPUTS
C
      SUBROUTINE DELC (DELCN,AJINV,JLIM,FY,KBIG,JFLAG,DUMMY,SIZE)
      DIMENSION DELCN(25), AJINV(25,25), FY(25), JFLAG(25)
      COMMON NFLAG
      IF (NFLAG.GE.3) WRITE (6,80)
      DO 10 I=1,KBIG
   10   CONTINUE
      DO 20 K=1,JLIM
        DELCN(K)=0.0
        DO 20 J=1,JLIM
          DELCN(K)=(AJINV(K,J)*FY(J))+DELCN(K)
   20     CONTINUE
C       ADJUST DELCN TO ACCOUNT FOR ZERO ROWS IN J MATRIX
      NSHIFT=JLIM-KBIG
      IF (NSHIFT.EQ.0) GO TO 40
      DO 30 JVAR=JLIM,1,-1
        IF (JFLAG(JVAR).EQ.1) DELCN(JVAR)=DELCN(JVAR-NSHIFT)
        IF (JFLAG(JVAR).EQ.0) DELCN(JVAR)=0.0
        IF (JFLAG(JVAR).EQ.0) NSHIFT=NSHIFT-1
   30   CONTINUE
   40 CONTINUE
C     DETERMINE THE LARGEST CHANGE
      DUMMY=0.0
      DO 50 K1=1,JLIM
        IF (ABS(DELCN(K1)).GT.DUMMY) DUMMY=ABS(DELCN(K1))
   50   CONTINUE
      WRITE (6,100) DUMMY
C     IF THE LARGEST CHANGE IS GREATER THAN 'SIZE', SCALE DOWN
      IF (DUMMY.LT.SIZE) GO TO 70
      DO 60 K2=1,JLIM
        DELCN(K2)=DELCN(K2)*SIZE/DUMMY
   60   CONTINUE
      DUMMY=SIZE
      WRITE (6,110) DUMMY
   70 CONTINUE
      IF (NFLAG.GE.3) WRITE (6,90)
      RETURN
C
   80 FORMAT (20X,'ENTER DELC')
   90 FORMAT (20X,'LEAVE DELC')
  100 FORMAT (20X,'MAX. DELTA COMPUTED =',G12.7)
  110 FORMAT (1H+,T60,'MAX. DELTA ALLOWED =',G12.7)
C
      END
```

APPENDIX C

SAMPLE OUTPUT OF HBA

* * * * * * * * * * * I N P U T   D A T A * * * * * * * * * * *

```
  DELTA       NO. POINTS    NL ITER          SEC. STOP LIM
.10000-05     301           150              .10000-04

   PRINT FLAG         MIN. STEP SIZE       INIT. STEP
      0                 .10000-07             .10000+00
       JMAX =  4

       NO. OF HARMONICS =3

       INITIAL W = 2.500000

       NUMBER OF PATHS =1
 NUMBER OF NONLINEARITIES FOR PATH =    3

                   INITIAL INPUT VALUES  .

                   DC = .0000000

                      HARMONIC VALUES     HAR. NO.    COSINES        SINES
                   FUNDAMENTAL COMP.                                1.000000
                                             2        .0000000      .0000000
                                             3        .0000000      .0000000

NO. OF BREAKPOINTS IN NONLINEARITY 1 OF PATH 1 IS 3

NONLINEARITY 1 OF PATH 1 IS SYMMETRICAL

BREAKPOINT VALUES FOR CURVE 1 OF NONLINEARITY 1 IN PATH 1

   HORIZONTAL COORDINATES   -.10000+06    -.20000+01      .20000+01

   VERTICAL COORDINATES     -.30000+01    -.30000+01      .30000+01

  SLOPE OF SEGMENT TO LEFT   .00000        .00000         .15000+01     .00000

  1 ORDER NUMERATOR, 2 ORDER DENOM. IN NONLINEARITY 1 IN PATH 1

 COEF.OF NUM. TERMS(HIGHEST FIRST)      .50000+01      .20000+02

 COEF.OF DEN. TERMS(HIGHEST FIRST)      .10000+01      .30000+01      .50000+01

NO. OF BREAKPOINTS IN NONLINEARITY 2 OF PATH 1 IS 4

NONLINEARITY 2 OF PATH 1 IS SYMMETRICAL

BREAKPOINT VALUES FOR CURVE 1 OF NONLINEARITY 2 IN PATH 1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   HORIZONTAL COORDINATES  -.10000+06   -.20000+00   .20000+00    .20000+00
   VERTICAL COORDINATES    -.15000+01   -.15000+01   -.15000+01   .15000+01
  SLOPE OF SEGMENT TO LEFT  .00000       .00000       .00000       .10000+21    .00000
BREAKPOINT VALUES FOR CURVE 2 OF NONLINEARITY 2 IN PATH 1
   HORIZONTAL COORDINATES  -.10000+06   -.20000+00   -.20000+00   .20000+00
   VERTICAL COORDINATES    -.15000+01   -.15000+01   .15000+01    .15000+01
  SLOPE OF SEGMENT TO LEFT  .00000       .00000       .10000+21    .00000       .00000
        NUMBER OF COMMON POINTS ON HYS.CURVES = 2
 HORIZONTAL COORDINATES OF COMMON POINTS   -.20000+00   .20000+00
 2 ORDER NUMERATOR, 3 ORDER DENOM. IN NONLINEARITY 2 IN PATH 1
 COEF.OF NUM. TERMS(HIGHEST FIRST)     .10000+01    .30000+01    .50000+01
 COEF.OF DEN. TERMS(HIGHEST FIRST)     .10000+01    .70000+01    .16000+02    .12000+02
NO. OF BREAKPOINTS IN NONLINEARITY 3 OF PATH 1 IS 3
NONLINEARITY 3 OF PATH 1 IS SYMMETRICAL
BREAKPOINT VALUES FOR CURVE 1 OF NONLINEARITY 3 IN PATH 1
   HORIZONTAL COORDINATES  -.10000+06   -.30000+01   .30000+01
   VERTICAL COORDINATES    -.25000+01   -.25000+01   .25000+01
  SLOPE OF SEGMENT TO LEFT  .00000       .00000       .83333+00    .00000
 0 ORDER NUMERATOR , 1 ORDER DENOM. IN NONLINEARITY 3 IN PATH 1
 COEF.OF NUM. TERMS(HIGHEST FIRST)     .10000+02
 COEF.OF DEN. TERMS(HIGHEST FIRST)     .10000+01    .40000+01
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* * * * * * * * * * * * * *     R E S U L T S     * * * * * * * * * * * * * * *

                    RUN NUMBER 1 FOLLOWS
                    MAX. PARTIAL PERTURBATION =  .1000000-05
*** COST 1 =  .1000000+11                 COST 2 =  .4697541     ***
MAX. DELTA COMPUTED = .9280023               MAX. DELTA ALLOWED = .1000000+00

                    NEW W0 =  2.000000

                    NEW DC COMPONENT =  .0000000

                    NEW FUNDAMENTAL SINE COMPONENT =  .9506131

                    NEW COMPONENTS FOR HARMONIC NO. 2; COSINE =  1.999915        SINE = -.1725285-03

                    NEW COMPONENTS FOR HARMONIC NO. 3; COSINE =  2.004194        SINE = -.9212824-02


                    *** RUN NUMBER  2 FOLLOWS ***
                    NEW MAX. PARTIAL PERTURBATION =  .2000000
*** COST 1 =  .4697541                    COST 2 =  .4128452     ***
MAX. DELTA COMPUTED = .3110745               MAX. DELTA ALLOWED = .2000000

                    NEW W0 =  2.800000

                    NEW DC COMPONENT =  .0000000

                    NEW FUNDAMENTAL SINE COMPONENT = . .8606877

                    NEW COMPONENTS FOR HARMONIC NO. 2; COSINE =  1.999630        SINE = -.1327442-02

                    NEW COMPONENTS FOR HARMONIC NO. 3; COSINE =  2.012131        SINE = -.2120595-01


                    *** RUN NUMBER  3 FOLLOWS ***
                    NEW MAX. PARTIAL PERTURBATION =  .4000000
*** COST 1 =  .4128452                    COST 2 =  .3136145     ***
MAX. DELTA COMPUTED = .6457211               MAX. DELTA ALLOWED = .4000000

                    NEW W0 =  3.200000

                    NEW DC COMPONENT =  .0000000

                    NEW FUNDAMENTAL SINE COMPONENT =  .6725611

                    NEW COMPONENTS FOR HARMONIC NO. 2; COSINE =  2.000208        SINE =  .1514935-02

                    NEW COMPONENTS FOR HARMONIC NO. 3; COSINE =  2.022547        SINE = -.2949087-01
```

```
                     *** RUN NUMBER  4 FOLLOWS ***
                     NEW MAX. PARTIAL PERTURBATION =  .8000000
*** COST 1 =  .3136143                COST 2 =  .1386140   ***
MAX. DELTA COMPUTED = .3559791

                     NEW WO =  3.555979

                     NEW DC COMPONENT =  .0000000

                     NEW FUNDAMENTAL SINE COMPONENT =  .5129287

                     NEW COMPONENTS FOR HARMONIC NO. 2; COSINE =  2.000385        SINE = -.5514878-03

                     NEW COMPONENTS FOR HARMONIC NO. 3; COSINE =  2.018822        SINE = -.2236249-01


                     *** RUN NUMBER  5 FOLLOWS ***
                     NEW MAX. PARTIAL PERTURBATION =  .7119583
*** COST 1 =  .1386140                COST 2 =  .9625562-02***
MAX. DELTA COMPUTED = .1932673-01

                     NEW WO =  3.575306

                     NEW DC COMPONENT =  .0000000

                     NEW FUNDAMENTAL SINE COMPONENT =  .5019447

                     NEW COMPONENTS FOR HARMONIC NO. 2; COSINE =  2.000006        SINE = -.9780168-05

                     NEW COMPONENTS FOR HARMONIC NO. 3; COSINE =  2.016803        SINE = -.2183889-01


                     *** RUN NUMBER  6 FOLLOWS ***
                     NEW MAX. PARTIAL PERTURBATION =  .3865346-01
*** COST 1 =  .9625562-02             COST 2 =  .7602896-04***


ERROR LIMIT SATISFIED
ACCEPTABLE VALUES ARE

                      DC =  .0000000

                     FUNDAMENTAL SINE COMPONENT =  .5019447

                     COMPONENTS FOR HARMONIC NO. 2; COSINE =  .1220431-04     SINE = -.9780168-05

                     COMPONENTS FOR HARMONIC NO. 3; COSINE =  .3360616-01     SINE = -.2183889-01

                     LIMIT CYCLE FREQUENCY =  3.575306    RADIANS/SEC.
```

REFERENCES

1. Truxal, John G., Introductory System Engineering, McGraw-Hill Book Company, Inc., New York, New York, 1972.

2. Dorf, Richard C., Modern Control Systems, Addison-Wesley Publishing Co., 2nd Ed., Reading, Massachusetts, 1974.

3. Kuo, Benjamin C., Automatic Control Systems, Prentice-Hall, Inc., 2nd Ed., Englewood Cliffs, N. J., 1975.

4. Evans, W. R., "Graphical Analysis of Control Systems," Trans. of AIEE, Vol. 67, 1948, pp. 547-551.

5. Truxal, John G., Control System Synthesis, McGraw-Hill Book Company, Inc., New York, New York, 1955, p.224.

6. Gibson, John E., Nonlinear Automatic Control, McGraw-Hill Book Company, Inc., New York, New York, 1973, pp. vii, 342-438.

7. Hsu, Jay C. and Andrew U. Meyer, Modern Control Principles and Applications, McGraw-Hill Book Company, Inc., New York, New York, 1968.

8. Fitts, R. E., "Two Counterexamples to Aizerman's Conjecture," IEEE Trans. on Automatic Control, Vol. AC-11, No.3, July, 1966, p. 553-556.

9. Gelb, Arthur, "Limit Cycles in Symmetric Multiple Nonlinearity Systems," IEEE Trans. on Automatic Controls, Vol. AC-8, April, 1963, pp. 177-178.

10. Gronner, A. D., "The Describing Function of Backlash Followed by a Deadzone," AIEE Trans.-Applications and Industry, Vol. 77, November, 1958, pp. 403-409.

11. Gran, R. and M. Rimer, "Stability Analysis of Systems with Multiple Nonlinearities," IEEE Trans. on Automatic Controls, Vol. AC-10, January 1965, pp. 94-97.

12. Jud, H. G., "Limit Cycle Determination for Parallel Linear and Nonlinear Elements," IEEE Trans. on Automatic Controls, Vol. AC-9, April, 1964, pp. 183-184.

13. Negoescu, N. and L. Sebastian, "Grapho-Analytical Method for the Determination of Self-Sustained Oscillations for a Class of Automatic Nonlinear Systems," IEEE Trans. on Automatic Controls, Vol. AC-16, No.3, June, 1971, pp. 272-274.

14. Davison, E. J. and D. Constantinescu, "A Describing Function Technique for Multiple Nonlinearities in a Single-loop Feedback System," IEEE Trans. on Automatic Control, Vol. AC-16, No. 1, February 1971, pp. 56-60.

15. Braswell, H. T., "Digital Computation of Multiple Harmonic Balance Conditions for a Class of Nonlinear Feedback Control Systems," M. S. Thesis, Mississippi State University, Mississippi State, MS., 1977.

16. Elgred, O. I., Electric Energy System Theory: An Introduction, McGraw-Hill Book Company, Inc., New York, New York, 1971, pp. 242-252.

17. Conte, S. D. and Carl de Boor, Elementary Numerical Analysis: An Algorithmic Approach, McGraw-Hill Book Company, Inc., 2nd Ed; New York, New York, 1972, pp. 58-59.