THREE-DIMENSIONAL DISPLAY OF PHASE SPACE DIAGRAMS

by

Donald L. English

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute

in partial fulfillment for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Chairman A. W. Bennett

L. L. Grigsby W. A. Blackwell

R. H. Miller

Date

May 1968

Blacksburg, Virginia

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author gratefully acknowledges the helpful comments and suggestions made by Dr. A. Wayne Bennett of the Electrical Engineering Department, Virginia Polytechnic Institute, in the initiation and conduction of this project. The author also wishes to express his gratitude to of the Virginia Polytechnic Institute Movie Service for his aid in making a movie of a perspective projection of the results.

The author would also like to express his sincere appreciation to his wife, , for her untiring patience during the writing of this thesis and to for her effort in typing this thesis.

I. INTRODUCTION

Phase plane analysis of linear and nonlinear systems has been widely used to good advantage in recent years. These graphical displays give considerable insight into system response. A serious short coming of this procedure is the limitation to two dimensions. A third-order system must be studied on two, two-dimensional graphs or represented by a second-order system. For nonlinear systems employing relays, saturation, etc., the approximations must be made with care.

In an effort to reduce discrepancies and improve the visualization of system response, this paper presents a method to obtain three-dimensional displays of system graphs. Third-order phase graphs of nonlinear and linear systems can be graphed directly without approximations. A discussion of the principles involved will be presented first. Then, analog and digital programs will be given. A discussion of problems encountered will then follow. Finally, several example problems will be discussed. These examples will be chosen to cover a wide variety of systems that may be encountered. The state-space approach will be used exclusively to obtain system responses.

II. DEVELOPMENT OF THE NECESSARY FUNDAMENTALS

2.1 Introduction

In order to project a three-dimensional view of an object, several problems must be solved. First, consider the problem of perspective. As an object moves closer to the eye it appears larger. The front edge of a box would look larger than its back edge. However, perspective control alone is not always sufficient to tell the relative distance of a set of objects from the observer. If a picture contains two objects and the viewer is familiar with their size, some indication of relative location can be determined. However, if the size and shape of the objects are not familiar, the mind has no way to determine their relative position. This is the problem confronted when viewing phase graphs. The mind has no "a priori" knowledge as to which portions of the curve are closest to the viewer. Also, any three-dimensional curve must be projected onto a plane for viewing.

This immediately leads to the second problem of three-dimensional graphics. The observer must be presented with two slightly different views of an object representing the view that each respective eye would perceive. The view for each eye is dependent upon the distance from the object to the observer. The combination of these two views along with perspective control is needed to construct the proper three-dimensional "illusion".

The third problem encountered is the construction of a viewer to present each of the graphs to its respective eye. This device should be designed so that the two graphs "fuse" into one three-dimensional picture.

Finally, it is often desirable to rotate the original input axes to allow them to appear in "normal perspective" (i.e. all three axes clearly visible instead of looking directly into the Z-axis). This can be accomplished by an orthogonal rotation. This rotation should be performed before perspective control. Continuous rotation about a given axis can also be used to achieve the three dimensional effect. This requires only one view and is very useful for movie projection and for use with graphical display units now available for digital computers.

2.2 Development

Consider first the problem of perspective. In order to draw an object it must be projected into a plane. Usually, it is a plane perpendicular to the viewing axis located some distance from the coordinate of the observer. This plane can also be defined by a set of three orthogonal coordinate axes, with the X and Y axis on the plane and the Z axis out of the plane towards the observer. Consider a point, r, (see Figure 2.2.1) on a continuous curve to be projected onto the plane for viewing. Assume the point, r, (lower case letters represent points and upper case letters represent vectors) to be the end of a vector from the Z axis parallel to the XY plane. R is projected onto the XY plane as R'.

The length of R' can be determined by simple geometry. Triangle z r d is similar to triangle o r' d. Therefore

$$\frac{R'}{D} = \frac{R}{E} \qquad (2.2.1)$$

or,

$$R' = \frac{D}{E} R. \qquad (2.2.2)$$

Since

$$E = D - Z, \qquad (2.2.3)$$

then,

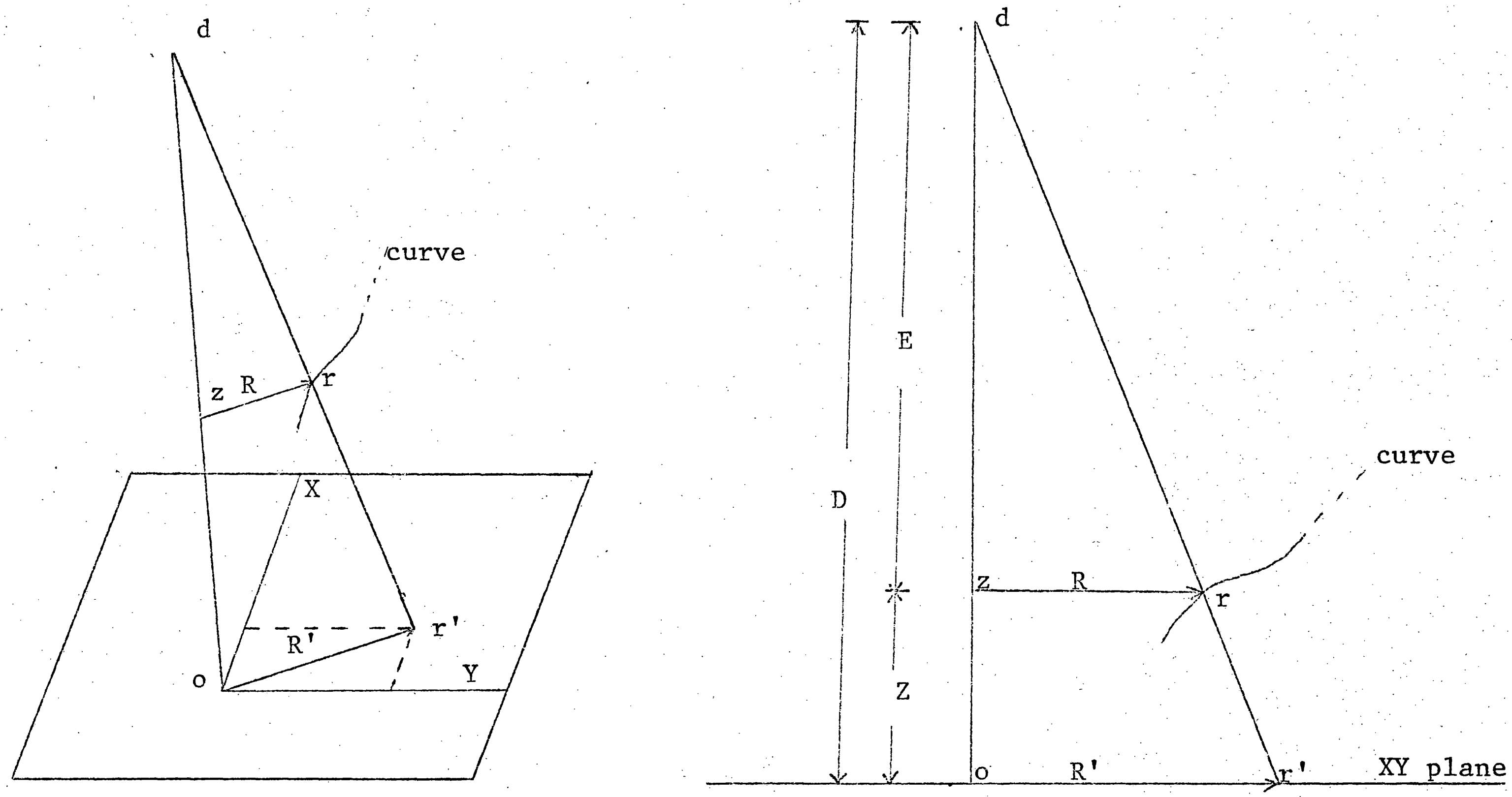
$$R' = \frac{D}{D - Z} R. \qquad (2.2.4)$$

The vector R' can be resolved into its X and Y components. This yields

$$X' = \frac{D}{D - Z} X \qquad (2.2.5)$$

and

$$Y' = \frac{D}{D - Z} Y. \qquad (2.2.6)$$



Perspective view Edge view

Figure 2.2.1 Determination of perspective projection onto a plane

Now consider the distance from the observation point, d, to the coordinate center of the system to be observed. This distance should be several times larger than the maximum value of Z expected for "normal perspective". If this distance is only twice the value of the maximum Z, the projection on the XY plane will become

$$R' = \frac{D}{D - \frac{D}{2}} R = 2R. \qquad (2.2.7)$$

If the coordinate axes were at the back of an object, then the front edge would appear twice as long as the back edge. This is more distortion than the eye is accustomed to seeing and, therefore, it would seem artificial. Also, the real image size would be limited to one half the dimension of the graph. Therefore, it was decided to select an observation distance ten times the maximum expected positive value of Z. Thus, the projected image will at most be

$$R' = \frac{10}{10 - 1} R. \qquad (2.2.8)$$

For special problems and "effects" it may be desirable to select a different ratio. The maximum values of X, Y, and Z can be controlled by scaling.

The determination of the different view for each eye is the second problem. As shown in Figure 2.2.2, each eye

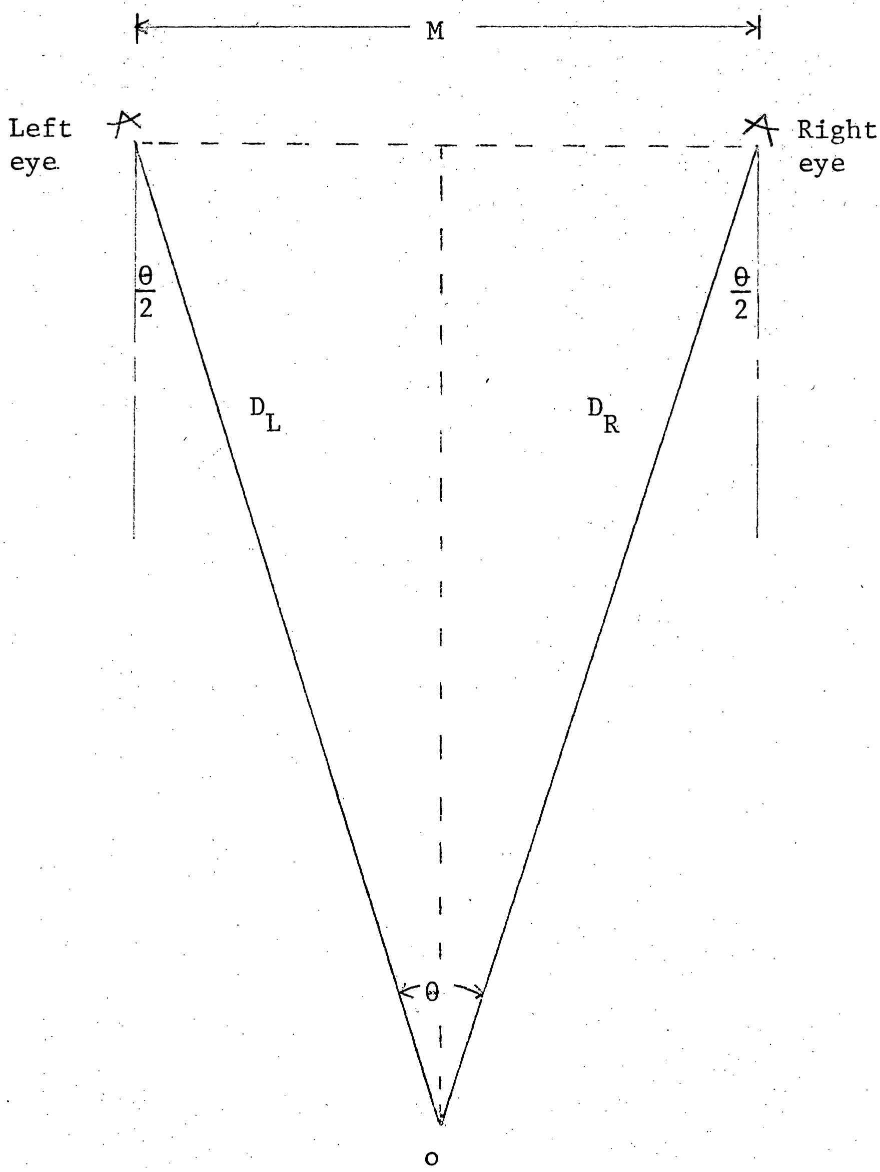


Figure 2.2.2 Different view as seen by each eye

looks at an object from a slightly different angle. Let a point on the object become the coordinate centers for a separate coordinate system for each eye. Select the axes so that the Y and Z axes for each eye lie in a plane perpendicular to the axis of vision. These axes are rotated from each other by an angle θ. This angle is dependent upon the optical distance from each eye and the separation of the two eyes. The distance D_R and D_L can be made equal by selecting the coordinate center at some point on a perpendicular located half way between the eyes. The distance of the object from the observation point is established in effect by the angle between these two planes and by the amount of perspective control utilized. Therefore, simulation of the object distance can be controlled by simply choosing the desired distance for perspective control and then computing the necessary rotation between the two optical planes. It should be noted that this is independent of the optical distance from the eye to the plane of the graph. The angle of rotation is given by

$$\theta = 2 \arcsin \left(\frac{M}{2D}\right). \qquad (2.2.9)$$

The next problem is the construction of an optical viewing device. In Figure 2.2.3, the graphs (one for each eye) are separated by a distance, L, and the optical system

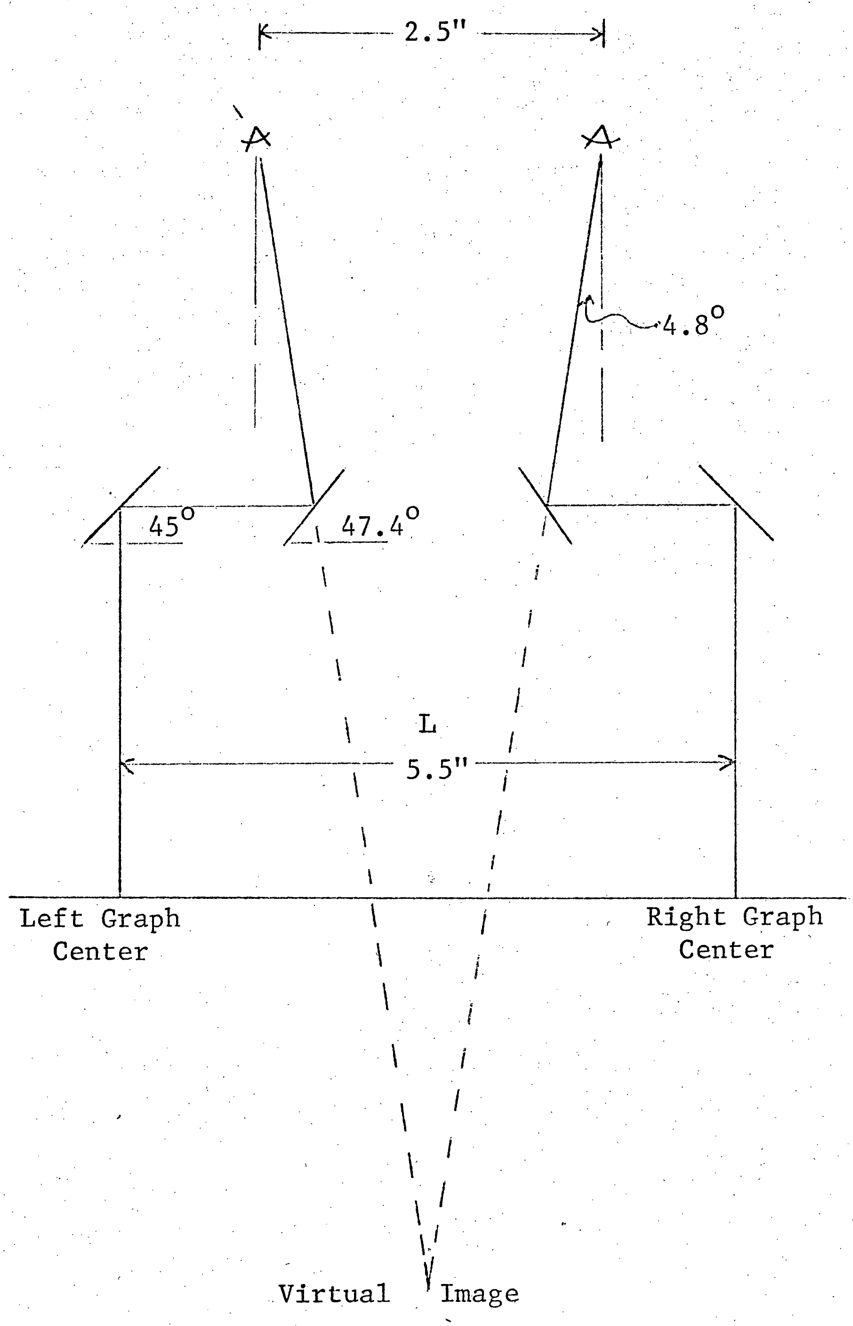


Figure 2.2.3 Three dimensional viewing device

must make the graph centers appear to coincide. The normal viewing distance of printed matter is between 15 and 20 inches. Therefore, the eyes will assume a slightly crossed position. Assuming an optical distance of 15 inches and a distance of 2.5 inches between the eyes, the angle each eye will be rotated from perpendicular line is

$$\theta = \arcsin \left(\frac{1.25}{15}\right) \approx 4.8° \qquad (2.2.10)$$

It was decided to select a distance of 5.5 inches between the centers of two graphs. This selection was somewhat arbitrary, the determining factor being convenient centering of the two graphs on standard 8 1/2 by 11 inch paper. Any reasonable distance could be selected and then the optical device designed for that distance.

The light reflecting objects may be mirrors or prisms. Prisms were selected because of their low light loss and availability. Consider a ray of light leaving the origin of one of the graphs and moving perpendicular toward the first prism (or mirror). If this device is set at an angle of 45 degrees with respect to a horizontal plane the light ray will be reflected at an angle of 90 degrees. Since the eyes are each assumed to be crossed by an angle of 4.8 degrees, the second reflecting device will have to be increased in angle from the horizontal to 47.4 degrees. Therefore, the light will be reflected at an angle of 85.2

degrees as shown. The eyes will see a virtual image from each graph fused into a single image. Since the graphed image is rotated for each eye, and is in perspective, the brain will reconstruct the original three-dimensional object.

It was pointed out previously that it is often desirable to rotate the coordinate axes to other positions to view the three original axes in "normal perspective". This can be accomplished by an orthogonal rotation. This rotation should be performed first and the new coordinates used to give eye rotation and perspective. An orthogonal rotation can be resolved into two axis rotations visualized by imagining a rotation about the Z axis and then a rotation about the new X_1 axis.

First rotate the coordinate system about the Z axis in a clockwise direction. See Figure 2.2.4. This results in a new set of orthogonal coordinates given by

$$X_1 = -X \sin\theta + Y \cos\theta, \qquad (2.2.11)$$

$$Y_1 = Z, \qquad (2.2.12)$$

and

$$Z_1 = X \cos\theta + Y \sin\theta. \qquad (2.2.13)$$

Now assume a side view of the system instead of a top view (X_1 axis out of the paper) and rotate the axis counter-

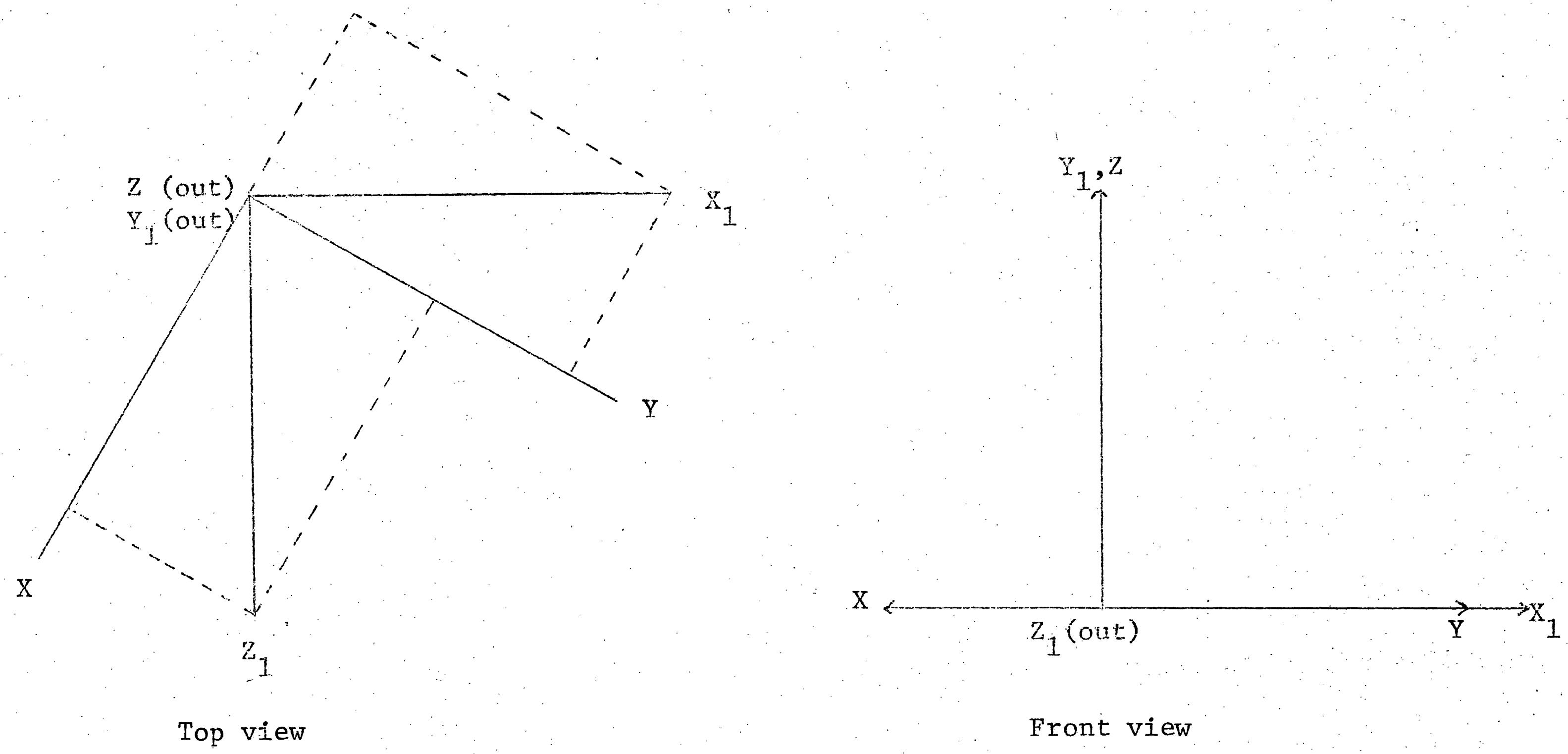


Figure 2.2.4 Clockwise coordinate system rotation about the Z-axis

clockwise about this X_1 axis as shown in Figure 2.2.5. Then we can define the new axis to be

$$X_2 = X_1 \qquad (2.2.14)$$

$$Y_2 = Y \cos \phi - Z \sin \phi, \qquad (2.2.15)$$

and

$$Z_2 = Y \sin \phi + Z \cos \phi. \qquad (2.2.16)$$

The rotated coordinates are expressed in terms of the original coordinates by

$$X_2 = -X \sin \theta + Y \cos \theta \qquad (2.2.17)$$

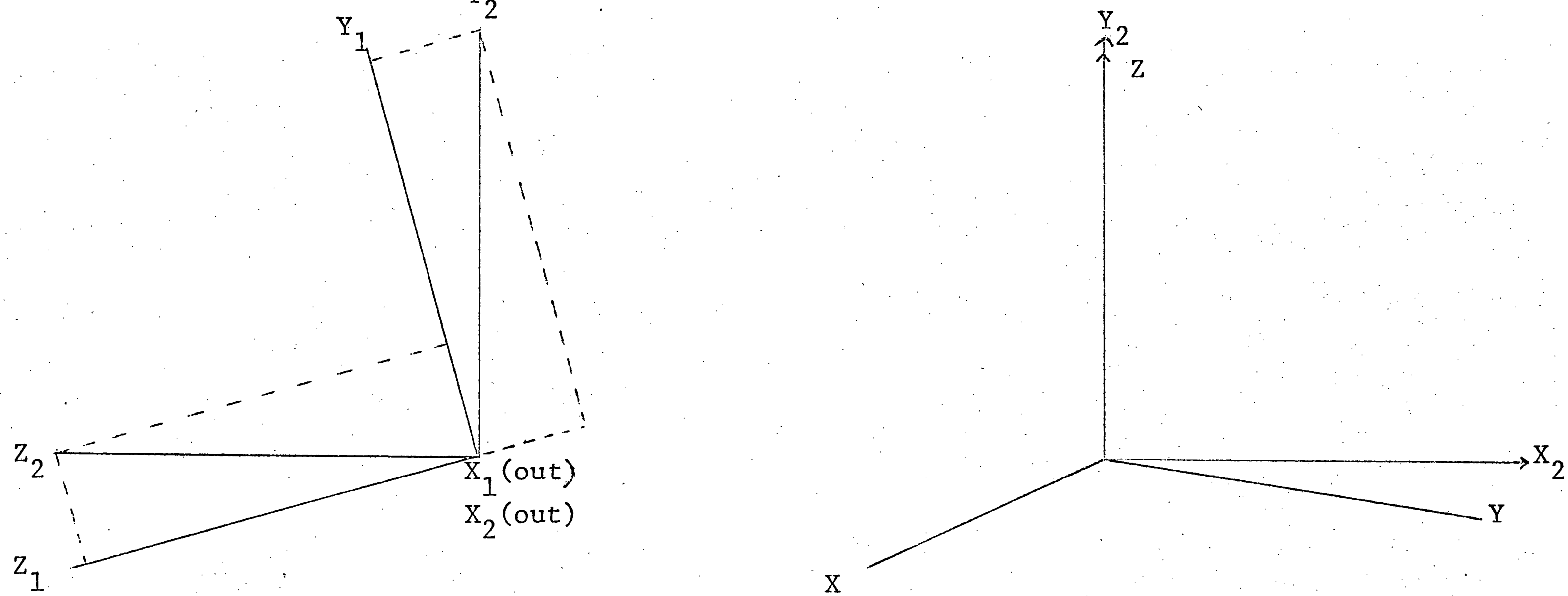
$$Y_2 = -X \cos \theta \sin \phi - Y \sin \theta \sin \phi + Z \cos \phi \qquad (2.2.18)$$

and

$$Z_2 = X \cos \theta \cos \phi + Y \sin \theta \cos \phi + Z \sin \phi \qquad (2.2.19)$$

This new coordinate system can now be used to give eye rotation and perspective. The distance to the projected object and the distance between the eyes must be known in order to determine the angle of rotation. Using 20 inches and 2.5 inches respectively, the angle between the two coordinate axes will be given by

$$\alpha = 2 \arcsin \left(\frac{1.25}{20}\right) \approx 7.1. \qquad (2.2.20)$$



Right side view Front view

Figure 2.2.5 Counterclockwise coordinate system rotation about the X_1 axis for left eye view

Therefore the right eye view will be rotated from the left eye view by 7.1 degrees as shown in Figure 2.2.6. This rotation is accomplished by

$$X_3 = X_2 \cos \alpha - Z_2 \sin \alpha, \qquad (2.2.21)$$

$$Y_3 = Y_2, \qquad (2.2.22)$$

and

$$Z_3 = X_2 \sin \alpha + Z_2 \cos \alpha, \qquad (2.2.23)$$

or in terms of the original coordinates by

$$\begin{aligned} X_3 = & X(-\sin \theta \cos \alpha - \cos \theta \cos \phi \sin \alpha) \\ & + Y(\cos \theta \cos \alpha - \sin \theta \cos \phi \sin \alpha) \\ & + Z(-\sin \phi \sin \alpha), \end{aligned} \qquad (2.2.24)$$

$$Y_3 = - X \cos \theta \sin \phi - Y \sin \theta \sin \phi + Z \cos \phi, \qquad (2.2.25)$$

and

$$\begin{aligned} Z_3 = & X(\cos \theta \cos \phi \cos \alpha - \sin \theta \sin \alpha) \\ & + Y(\cos \theta \sin \alpha + \sin \theta \cos \phi \cos \alpha) \\ & + Z(\sin \phi \cos \alpha). \end{aligned} \qquad (2.2.26)$$

For $\theta = 30°$, $\phi = 15°$, the equations for the left eye view will become

$$X_2 = -.500X + .866Y, \qquad (2.2.27)$$

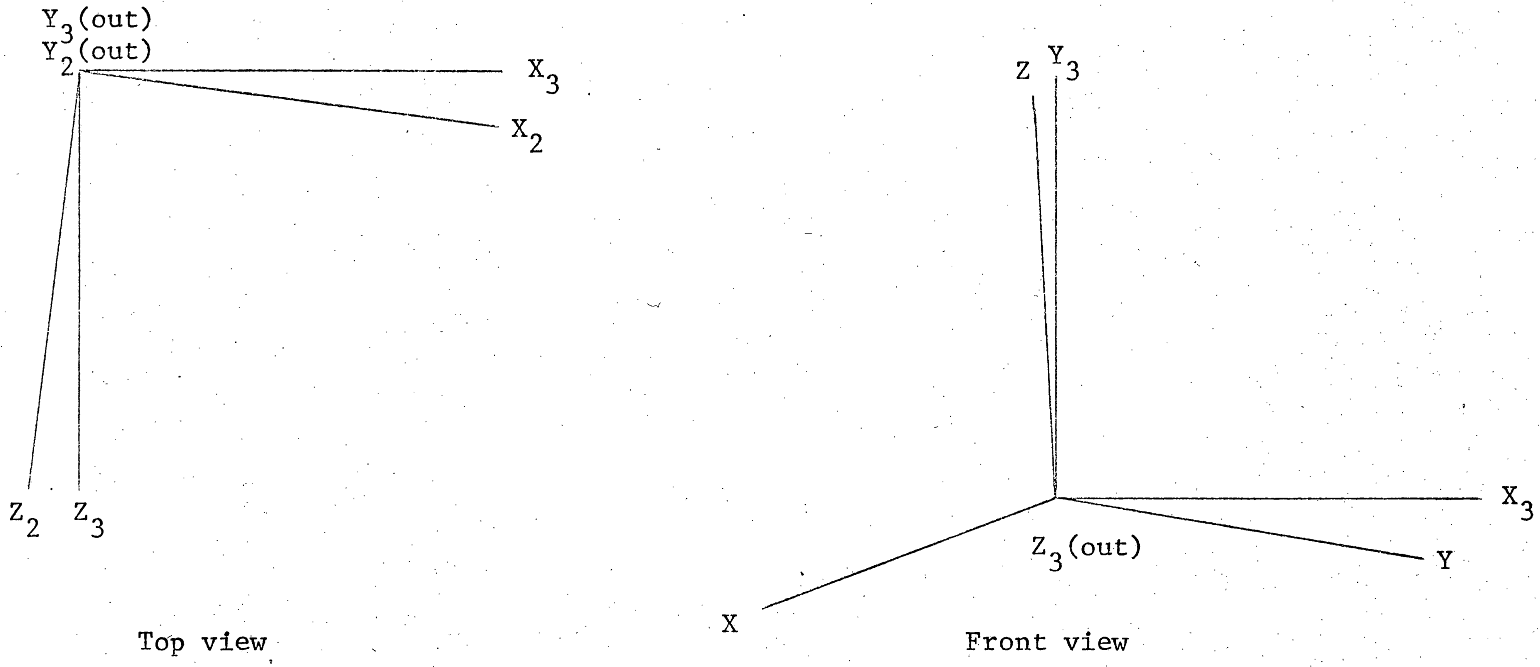


Figure 2.2.6 Clockwise coordinate system rotation about the Y_2 axis for right eye view

$$Y_2 = -.224X - .129Y + .966Z, \quad (2.2.28)$$

and

$$Z_2 = .837X + .483Y + .259Z. \quad (2.2.29)$$

The angle of rotation for the right eye view is $\alpha \approx 7.1°$. Therefore, the right eye view equations reduce to

$$X_3 = .595X + .799Y - .032Z, \quad (2.2.30)$$

$$Y_3 = -.224X - .129Y + .966Z, \quad (2.2.31)$$

and

$$Z_3 = .768X + .568Y + .258Z. \quad (2.2.32)$$

III. COMPUTER SOLUTION FOR THREE-DIMENSIONAL PROJECTION

3.1 Introduction

The equations developed in the last chapter can be solved on either a digital computer or an analog computer. It is essential to have a graphical display for the digital computer and an XY plotter for the analog computer. In presenting the computer solution, a general computer algorithm will be developed first. Then, the resulting programs will be discussed.

3.2 Computer Algorithm

Assume that the three state-space variables of some problem are available. These variables can be generated from presently available digital programs or simulated on the analog computer. These three variables can be considered as the X, Y, and Z components of a point on the state-space trajactory. As time increases, these variables will continually define its position in state-space.

As stated earlier, it is often desirable to rotate the coordinates to see "normal perspective". This orthogonal rotation should be performed before projection onto the graphical plane. Consider the left eye view. Figure 3.2.1 gives the rotated coordinates X_1, Y_1, and Z_1. X_1 and Y_1

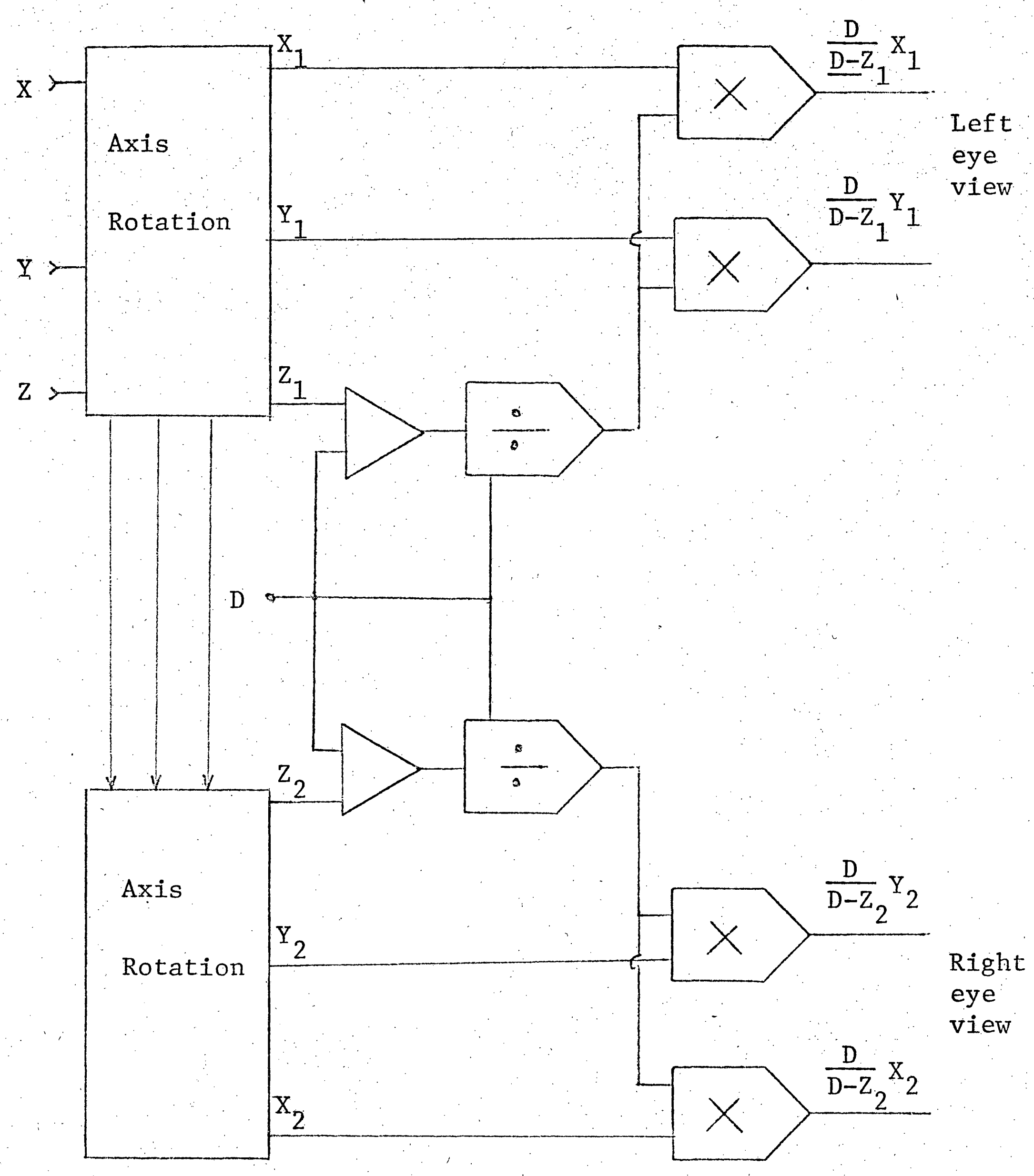


Figure 3.2.1 Computer algorithm

are the coordinates of the viewing plane and Z_1 is the coordinate toward the observation point.

Next, the distance of the observer from the plane of the graph must be generated. If the input signal is scaled to a maximum of two units, then "D" should be set to twenty units to give the arbitrary ten to one ratio selected earlier.

The Z_1 and D signals must next be combined to give $D - Z_1$. Then both this signal and the D variables can be combined to give

$$\frac{D}{D - Z_1} \quad . \qquad (3.2.1)$$

This is the necessary multiplier factor for perspective control. As the Z_1 portion of the system trajectory becomes positive, its planar projection must move further from the origin.

This perspective control function, along with the X_1 and Y_1 trajectory, will give the view for the left eye when plotted. The axes can be computed by setting two of the three input variables to zero and varying the third variable through the range desired. Since a maximum of two units was selected for the system trajectory, each of the three variables should be varied between plus two and minus two units individually.

The right eye view should be considered next. The rotated variables must now be rotated through an additional angle, α, as defined in equations 2.2.20, 2.2.21, 2.2.22, and 2.2.23. Perspective control must be applied to the new variables X_2, Y_2 and Z_2. It should be noted that the graph must be shifted to the left by five and one half inches before this right eye view is drawn. This is the distance selected earlier for the three-dimensional viewer constructed.

3.3 Digital Program

The digital program follows the algorithm just developed in the previous section. A simplified flow graph is shown in Figure 3.3.1. To simplify the program and decrease the amount of redundancy two major subroutines were used. One of these subroutines was a numerical integration routine and the other was an axis plotting routine.

The program begins by reading θ, ϕ, and the distance from the observation point to the coordinate centers. Next, the angle, α, necessary for eye rotation is computed. The numerical integration subroutine was then called.

The numerical integration routine selected was a Runge-Kutta-Gil numerical procedure. This program had previously been adapted to state-space problems. Therefore, it is only necessary to take the output of this routine and

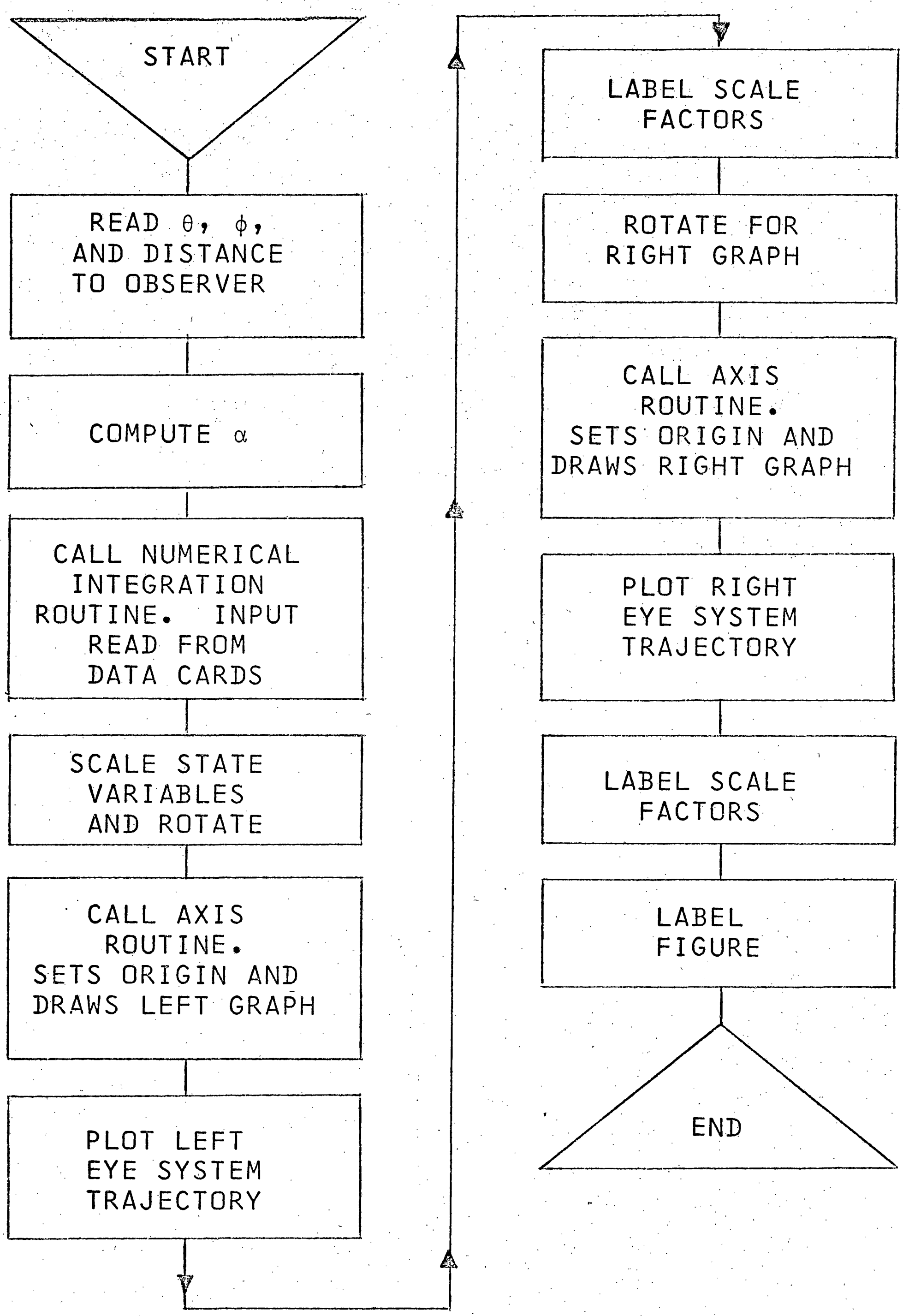


Figure 3.3.1 Simplified Digital Computer Flow Graph

return it to the main program where it is rotated and plotted. The numerical routine will be discussed in detail in Chapter IV when example problems are presented.

Next, the axis plotting routine selects the origin for the left graph. This routine contains the equations of rotation for the left eye view axis and the right eye view axis. The X, Y, and Z signals are varied from +2 to -2 units individually to plot the three axes. Then, the routine places eight tic marks on each of these axes. The spacing of these tic marks is a function of the distance from the plane of projection and the angle of the axis from the plane. At this point, control returns to the main program and the left eye system trajectory is plotted along with the scale factors.

The system trajectory is next rotated for the right eye view and the axis routine sets the new origin and draws the axes for the right eye graph. The trajectory for the right eye is then drawn and the scale factors labeled. Finally, the figure number and title are labeled by the plotter.

This basic program was also modified to graph the axes and system response and then rotate the coordinate system through an angle of 5 degrees. The program repeated this routine a total of seventy-two times for a total coordinate rotation of 360 degrees. With a graphics terminal and a movie camera, the system response can be photographed and the resulting movie will give the illusion of three-dimensions. On digital systems with an oscilloscope type display, this response can be obtained immediately.

3.4 Analog Program

The algorithm was also adapted for use on an Electronic Associates Incorporated model TR-20 analog computer. The available computers had only one quarter-square multiplier on each console and therefore, it was necessary to simulate the left eye view first and then adjust system parameters and simulate the right eye view. Since three quarter-square multipliers were required, three of the TR-20's had to be patched together.

At this point it will be assumed that a third order system has been selected. This system can easily be programmed on the analog computer and the three state variables used as input for the system rotation. Also it will be assumed that the rotations selected are $\theta = 30°$ and $\phi = 15°$, and a distance to the observation point from the coordinate center is 18 inches.

The equations of rotation for the left eye view for $\theta = 30°$ and $\phi = 15°$ are

$$X_2 = -.500X + .866Y, \qquad (3.4.1)$$

$$Y_2 = -.224X - .129Y + .966Z, \qquad (3.4.2)$$

and

$$Z_2 = .837X + .483Y + .259Z. \qquad (3.4.3)$$

This set of rotated coordinates can be realized on the TR-20 by using two summer-amplifiers. Two are required because only two inputs of the same gain can be applied to each amplifier. Three of these circuits are required to generate the three axis outputs.

After the axis rotation, perspective control must be generated. The function representing perspective control is

$$f(D\ ,\ Z) = \frac{D}{D - Z_2}\ . \qquad (3.4.4)$$

The actual analog circuit is shown in Figure 3.4.1. The analog input was scaled such that the perspective output was

$$\frac{5D}{D - Z_2}\ . \qquad (3.4.5)$$

This higher signal was selected to increase the voltage input to the multipliers. The overall accuracy of the multiplier is better for wider excursions in signal voltage. The maximum signal voltage must be limited to ± 10 volts. Since a distance of 18 units (1 volt = 1 unit) was selected for the observation distance, the input Z_2 and the input D must be scaled. A scale factor of one half was selected.

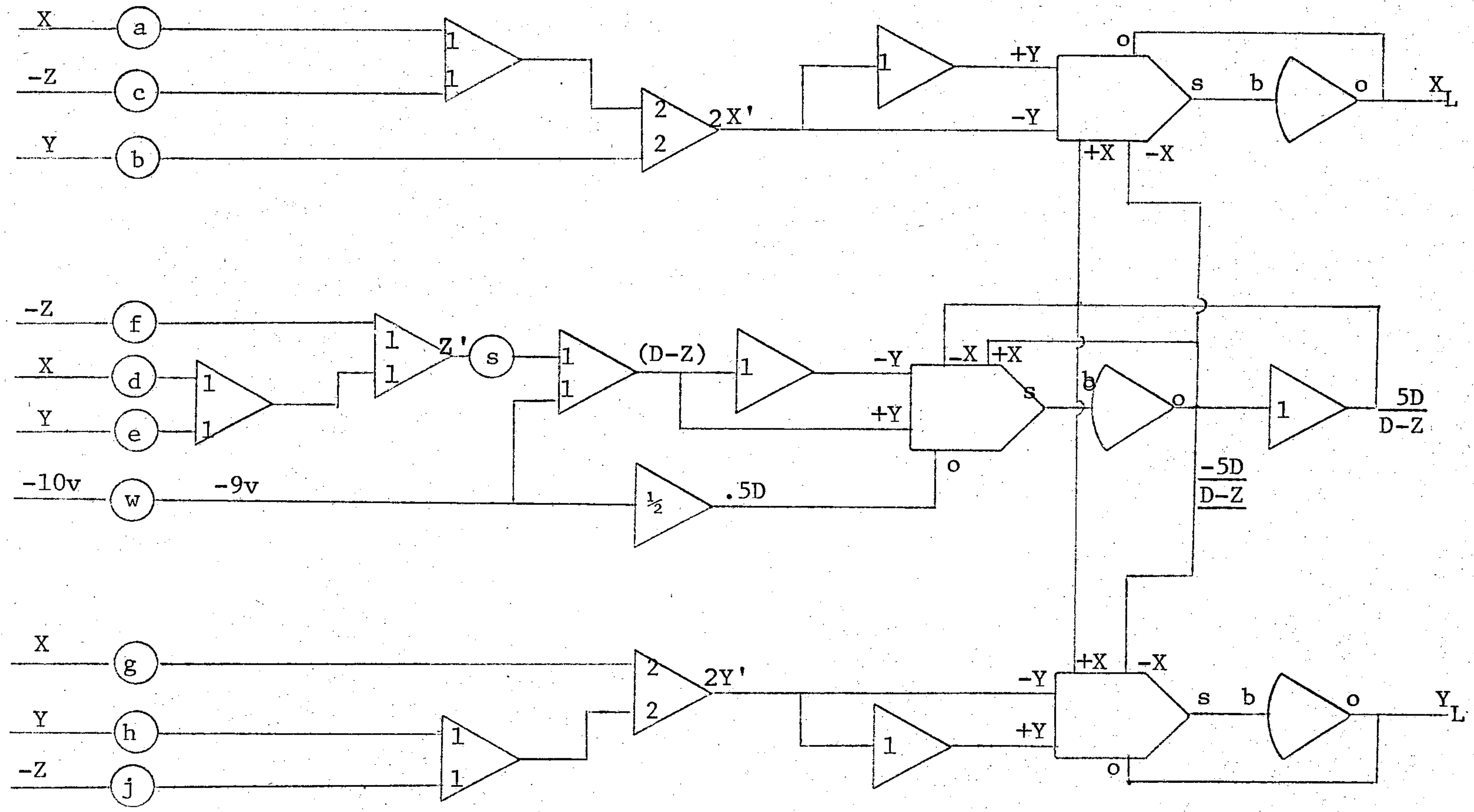


Figure 3.4.1 TR-20 analog computer program

This reduces the value of -D to 9 volts which is within the tolerance limits of the analog computer. Since the Z' signal is also scaled by this same factor of one half, the inputs to the divider will become

$$\text{Numerator} = \frac{1}{2}(.5D) \qquad (3.4.6)$$

and

$$\text{Denominator} = \frac{1}{2}(D - Z). \qquad (3.4.7)$$

The output of the divider on the TR-20 for the inputs "X" and "Y" is

$$\frac{10X}{Y}. \qquad (3.4.8)$$

Therefore, the output of the divider will become

$$\frac{5D}{D - Z'} . \qquad (3.4.9)$$

This scaling of both of the divider inputs by the same factor has not affected the output of this perspective control generator.

Next it will be necessary to multiply both the X' and Y' signal by a factor of two. The output of the multiplier on the TR-20 for inputs X and Y is

$$\frac{XY}{10} . \qquad (3.4.10)$$

Therefore the outputs will become

$$X_2 = \frac{D}{D - Z'} X' \qquad (3.4.11)$$

and

$$Y_2 = \frac{D}{D - Z'} Y'. \qquad (3.4.12)$$

These two outputs can then be applied to the XY plotter to give the response curve.

The coordinates for the right eye view can next be set with the coefficient potentiometers and the right eye coordinate center set with the plotter zero. Then the above procedure can be repeated to obtain the right eye view.

IV. APPLICATION

4.1 Introduction

In this chapter the techniques that have been developed will be used. Several examples exhibiting stability characteristics that may be encountered in a practical situation will be presented. The integration routine used in digital simulation will also be discussed. One basic system will be used to illustrate the linear system characteristics and a position control will be used for showing the nonlinear effects. The rational canonical system was chosen for the linear problems because of its relationship to the standard differential equation. Only the autonomous case will be considered in the examples presented (i.e. the input will be set to zero). The system diagram is shown in Figure 4.1.1.

4.2 System Simulation

The integration routine selected is a Runge-Kutta-Gil procedure. It has been previously adapted to the state-space matrix system

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \qquad (4.2.1)$$

and

$$\underline{y} = C\underline{x} + D\underline{u} \qquad (4.2.2)$$

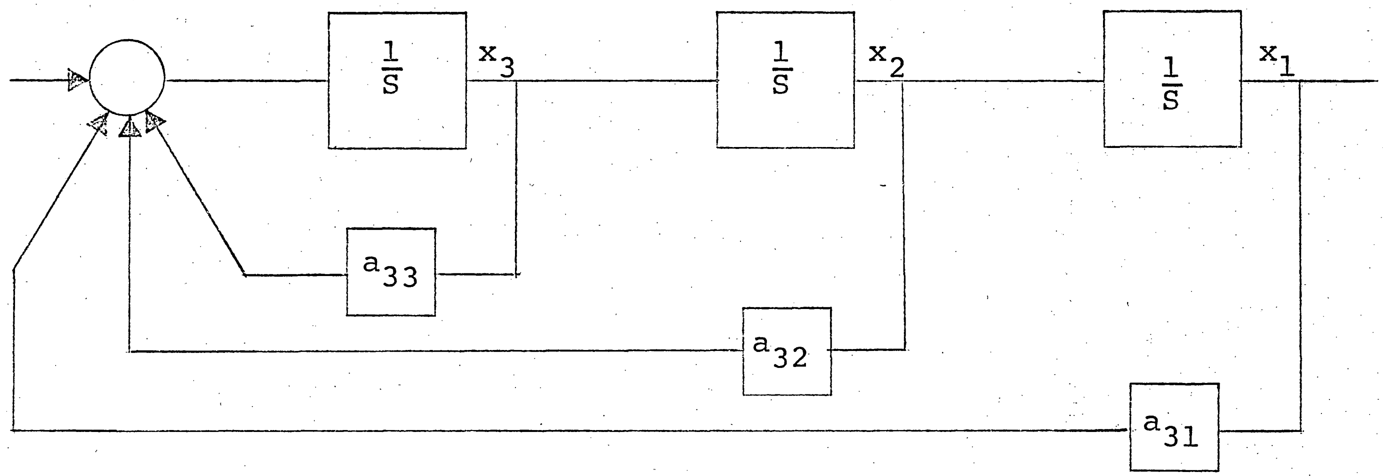


Figure 4.1.1 Rational canonical form system

where $\underline{\dot{x}}$ and $\underline{x}$ are n-vectors, A is an n x n matrix, B is an n x 4 matrix and $\underline{u}$ is a column vector with 4 components. The first two components of $\underline{u}$ are unit step functions and the last two are sine and cosine functions.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1\ (t - \tau_1) \\ 1\ (t - \tau_2) \\ 1\ \sin\ (2\pi\tau_3 t) \\ 1\ \cos\ (2\pi\tau_4 t) \end{bmatrix} \qquad (4.2.3)$$

The numerical routine chosen allows τ to be selected arbitrarily. The entries of the B matrix determine the elements of $\underline{u}$ to be applied to the system. The $\underline{y}$ vector has m-components. This is the output of the routine which will be graphed and therefore, will contain three elements. In the examples presented here the state variables $\underline{x}$ will be graphed. Therefore, the C matrix will be the identity matrix and D will be the null matrix.

The simulation program allows the $\underline{A}$, $\underline{B}$, $\underline{C}$, and D matrices to be read from data cards. Any initial conditions that exist in the system may also be read from data cards along with the driver parameter vector $\underline{\tau}$. The integration interval, total integration time, and output print (plot) interval must also be read on data cards. These equations are also printed out for handy reference. An example problem is printed in the appendix along with the computer program.

A state-space graph of the third order rational canonical form is shown in Figure 4.1.1. The three state variables are noted and the feedback paths are labeled with the appropriate elements in the $\underline{A}$ matrix. For the autonomous case the B matrix will always be the null matrix.

4.3 Example Problems

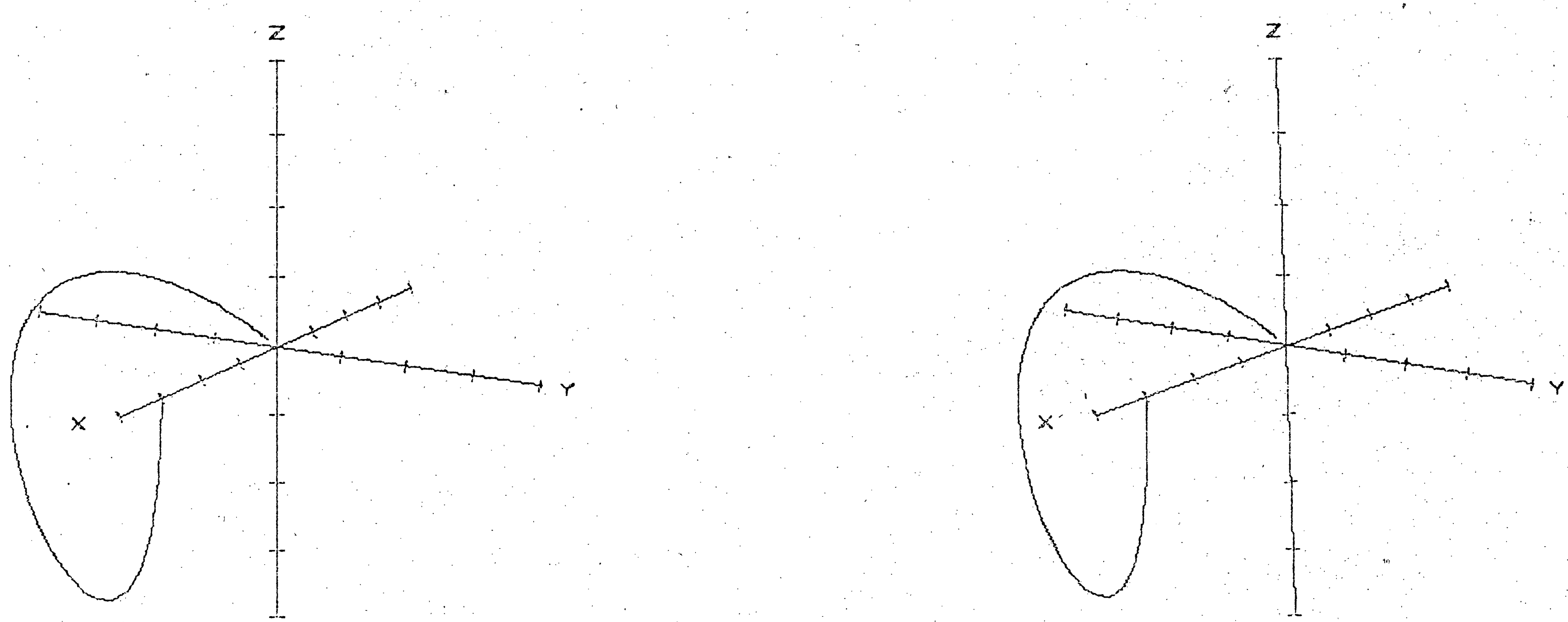
Consider first the system where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix} \qquad (4.3.1)$$

and

$$\underline{x}(t = 0) = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} . \qquad (4.3.2)$$

This system was simulated and the resulting state-space phase graph is shown in Figure 4.3.1. With the aid of the optical viewer constructed these two graphs become one three-dimensional graph. The scale factor for each state vector is noted on the graph. Usually these vectors are not scaled independently. However, the problem of keeping



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .360 UNITS
1 Z DIVISION = .307 UNITS

1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .360 UNITS
1 Z DIVISION = .307 UNITS

FIGURE 4.3.1 EQUAL AND REAL ROOTS IN LEFT-HALF PLANE the output within specified margins made this scaling necessary. An initial condition was applied only to the x_1 component and the trajectory moved such that all three vectors changed. Also, the system trajectory moves to the origin without any oscillations. This indicates that for this set of initial conditions the system is not dis-coupled.

The characteristic equation, $(A - \lambda I)$, was found to be

$$f(\lambda) = \lambda^3 + 3\lambda^2 + 3\lambda + 1 = 0 \qquad (4.3.3)$$

or

$$f(\lambda) = (\lambda + 1)(\lambda + 1)(\lambda + 1) = 0. \qquad (4.3.4)$$

There are three roots located at -1 and this system should exhibit stability without oscillations. The Jordan canonical form of A is

$$J = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \qquad (4.3.5)$$

This indicates that, regardless of the initial conditions, the system equations will be coupled and it is not possible to excite an individual mode.

The second problem is defined by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -.25 & -1.25 & -2 \end{bmatrix} \qquad (4.3.6)$$

and

$$\underline{x}(t = 0) = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} . \qquad (4.3.7)$$

This system was simulated and oscillations were noted as the trajectory spiraled into the origin (see Figure 4.3.2). This implies a pair of damped imaginary roots and a real root in the left-half plane. The roots of the system were found to be

$$\lambda_1 = -1,$$

$$\lambda_2 = -.5 - j1,$$

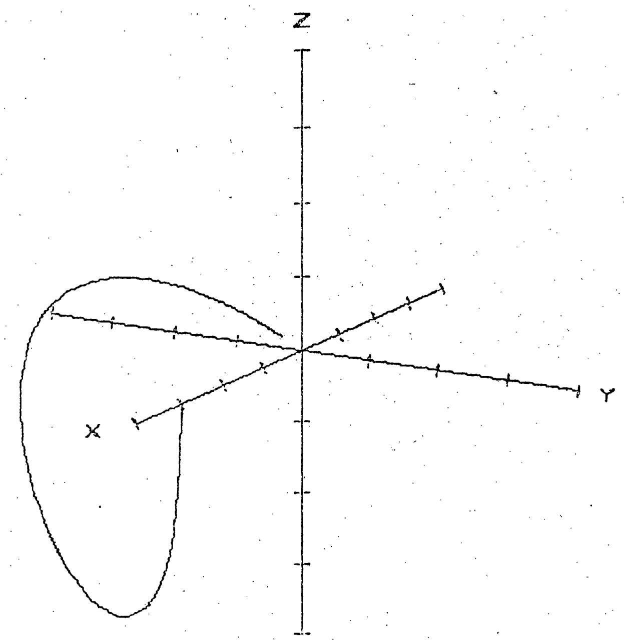
and

$$\lambda_3 = -.5 + j1.$$

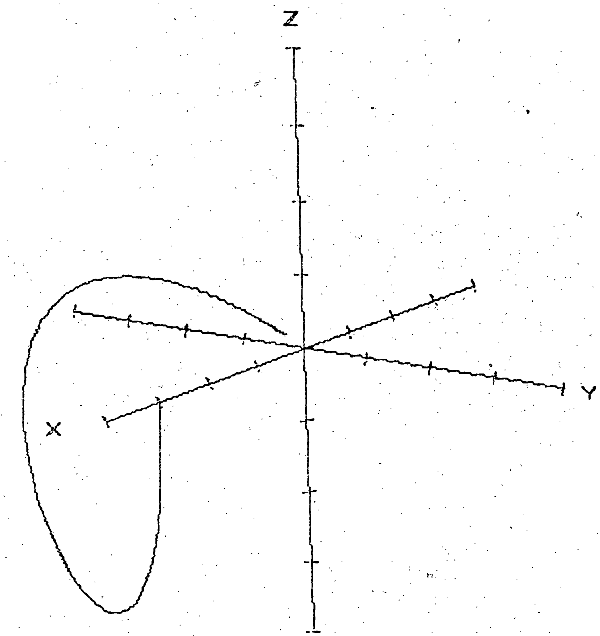
This is a stable system as indicated by the system trajectory.

The third system to be programmed was represented by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -.5 & -1. & -.5 \end{bmatrix} . \qquad (4.3.8)$$



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .215 UNITS
1 Z DIVISION = .116 UNITS



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .215 UNITS
1 Z DIVISION = .116 UNITS

FIGURE 4.3.2 ALL ROOTS IN LEFT-HALF PLANE COMPLEX ROOTS

Figure 4.3.3 shows that this system trajectory moves into a planar limit cycle whose plane of oscillation appears to pass through the origin. The roots of the characteristic equation of the system are

$$\lambda_1 = -1,$$

$$\lambda_2 = +j1,$$

and

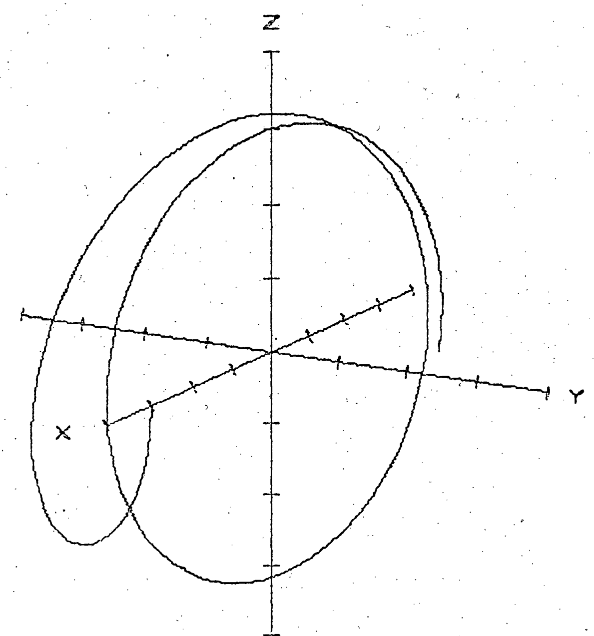
$$\lambda_3 = -j1.$$

The imaginary roots are responsible for the limit cycle. Since the real root is in the left-half plane the limit cycle should be in a plane passing through the origin. This problem was also programmed on the analog computer and the resulting graph is shown in Figure 4.3.4. The XY plotter available has an erratic drive system. Therefore, the output curve is not very smooth.

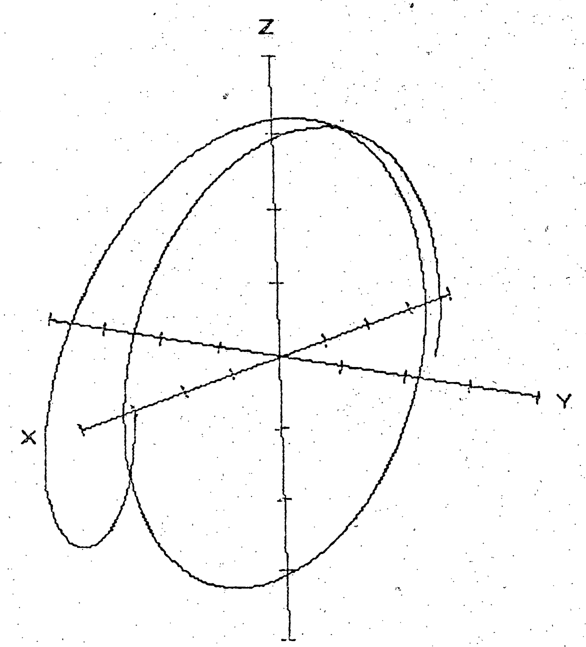
For a fourth problem consider the system defined by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1.25 & -.25 & 0 \end{bmatrix} \qquad (4.3.9)$$

and



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .739 UNITS
1 Z DIVISION = .628 UNITS



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = .739 UNITS
1 Z DIVISION = .628 UNITS

FIGURE 4.3.3 REAL ROOT IN LEFT-HALF PLANE COMPLEX ROOTS ON JW AXIX

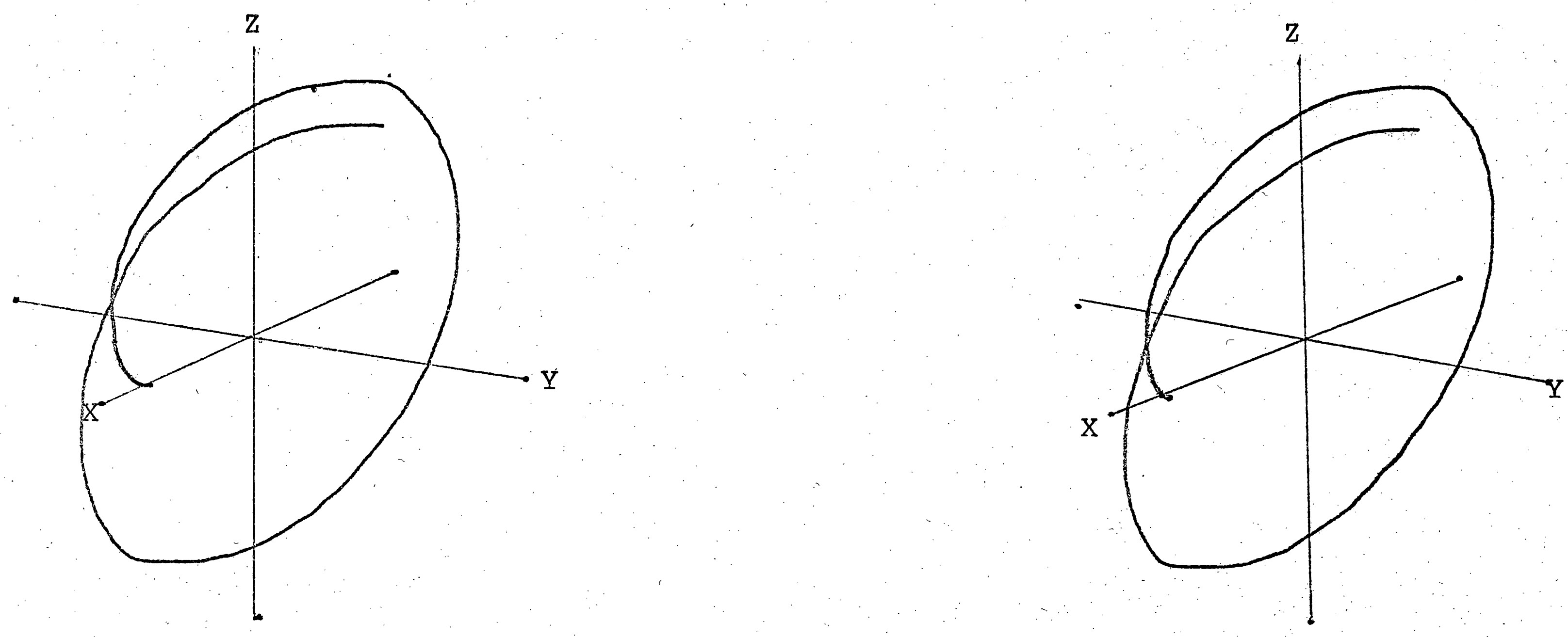


Figure 4.3.4 Analog graph of system with limit cycle

$$\underline{x}(t = 0) = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} . \qquad (4.3.10)$$

Investigation of the system equation reveals that the roots are

$$\lambda_1 = -1,$$

$$\lambda_2 = +.5 + j1,$$

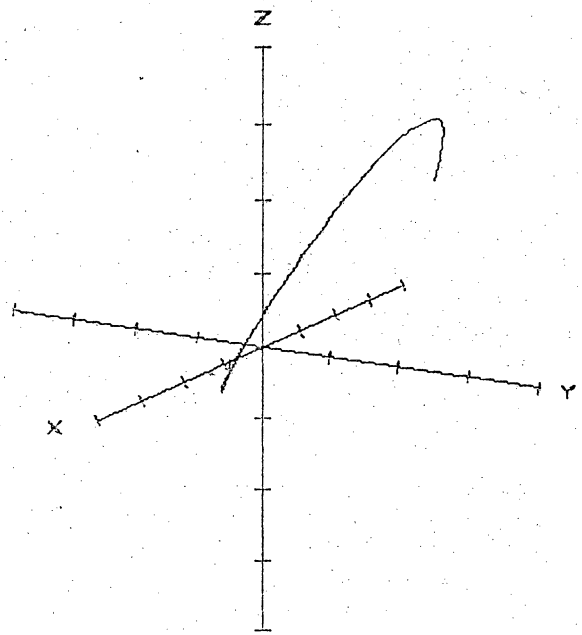
and

$$\lambda_3 = +.5 - j1.$$

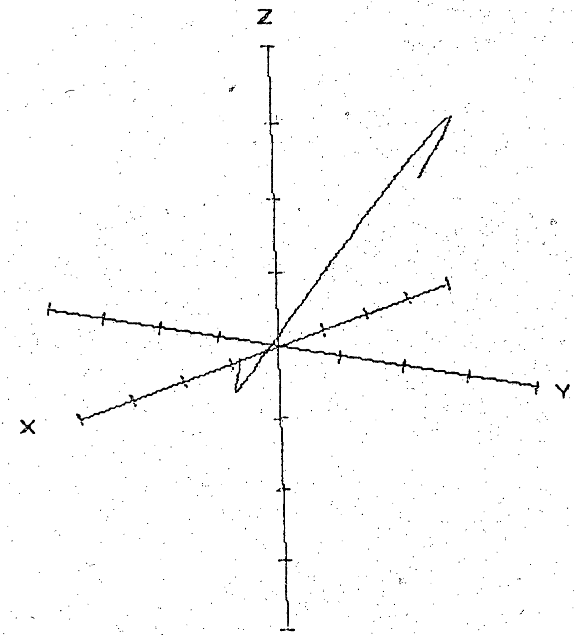
This system has one real root in the right-half plane and the imaginary roots are positively damped. Therefore, the system should oscillate about a plane passing through the origin and grow with each oscillation. This is shown in Figure 4.3.5.

Next consider the problem of a typical position control system with input relay control. This relay control has the characteristics shown in the relay block in Figure 4.3.6.

One approach to a solution is to treat the input state variable $\dot{x}_3$ as a variable gain. This gain can be readjusted at each integration interval such that the input to $\dot{x}_3$ remains exactly A for an output of more than



1 X DIVISION = 4.758 UNITS
1 Y DIVISION = 10.754 UNITS
1 Z DIVISION = 9.480 UNITS



1 X DIVISION = 4.758 UNITS
1 Y DIVISION = 10.754 UNITS
1 Z DIVISION = 9.480 UNITS

FIGURE 4.3.5 REAL ROOT IN LEFT-HALF PLANE COMPLEX ROOTS IN RIGHT-HALF PLANE

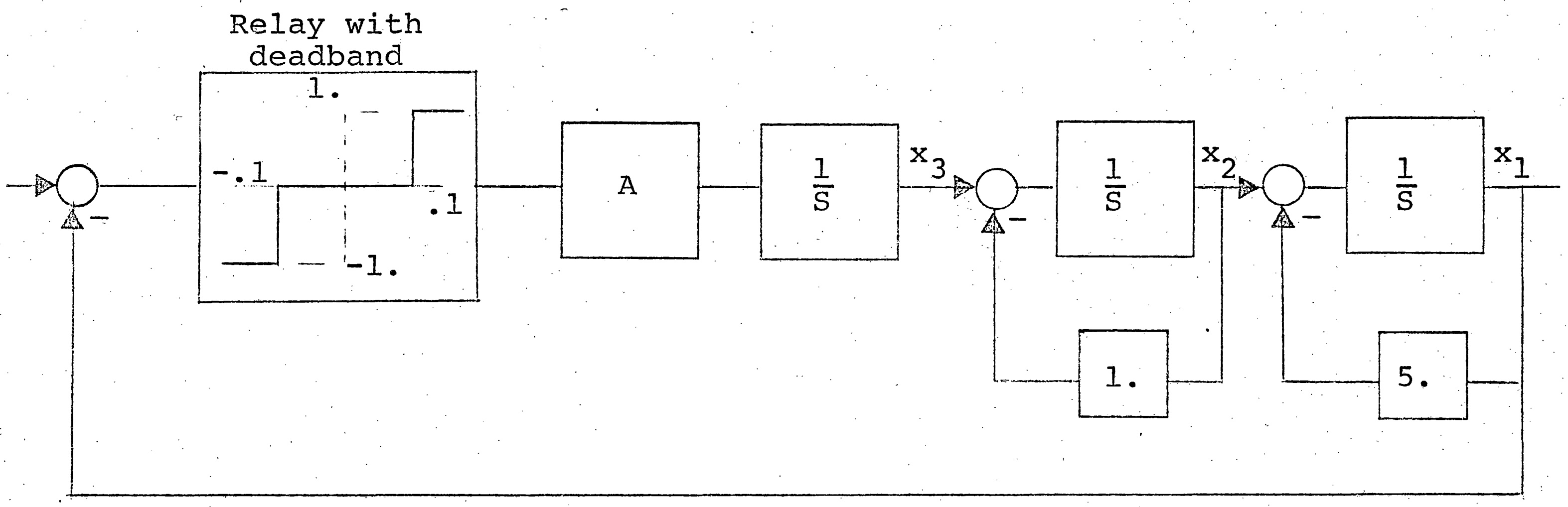


Figure 4.3.6 Position control system with relay control

one unit and zero for an output of less than one unit. The A matrix will then become

$$A = \begin{bmatrix} -5 & 1 & 0 \\ 0 & -1 & 1 \\ N & 0 & 0 \end{bmatrix} \qquad (4.3.11)$$

where

$$N = 0 \qquad \text{for } |x_1| < 1 \qquad (4.3.12)$$

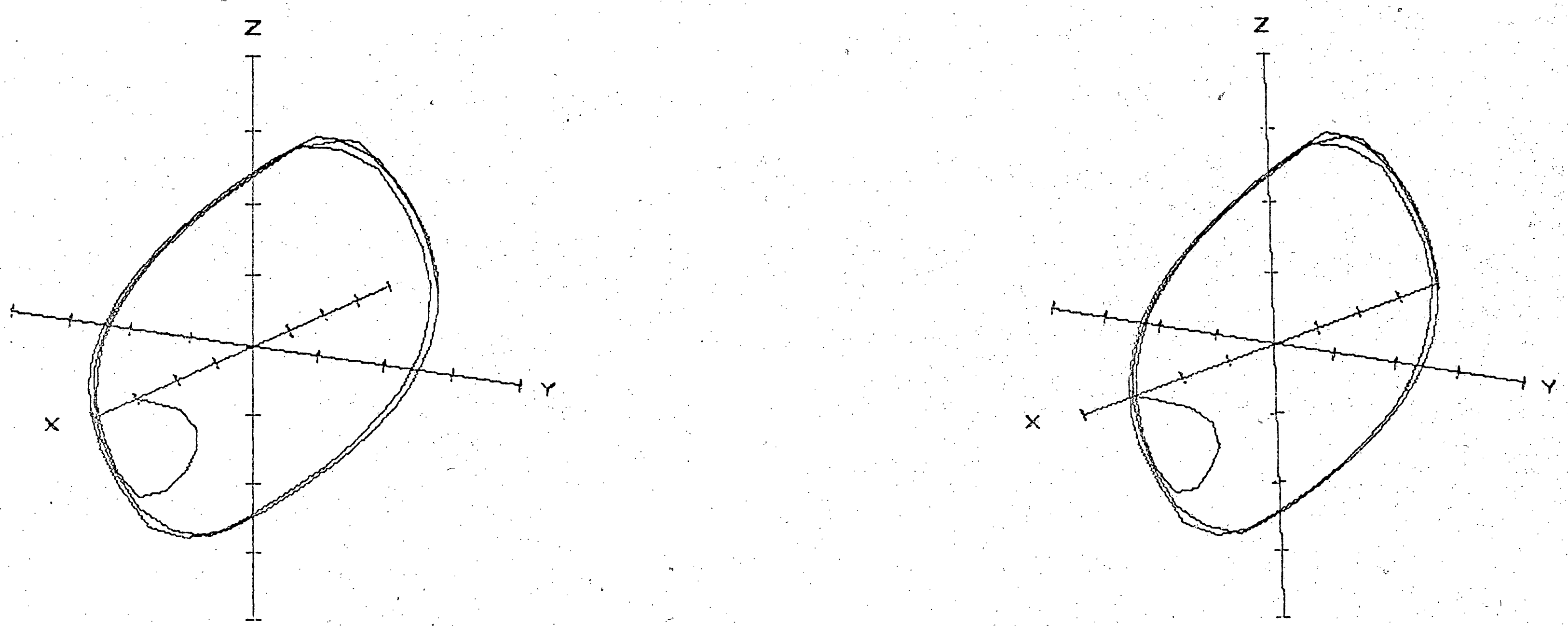
and

$$N = -\frac{A}{|x_1|} \qquad \text{for } |x_1| \geq 1. \qquad (4.3.13)$$

The initial condition vector was

$$\underline{x} = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} . \qquad (4.3.14)$$

The response of this system is shown in Figure 4.3.7.



1 X DIVISION = 1.333 UNITS
1 Y DIVISION = 1.170 UNITS
1 Z DIVISION = 3.267 UNITS

1 X DIVISION = 1.333 UNITS
1 Y DIVISION = 1.170 UNITS
1 Z DIVISION = 3.267 UNITS

FIGURE 4.3.7 SYMMETRIC RELAY CONTROL WITH DEADBAND

V. CONCLUSION

A method has been presented to construct three-dimensional views of third order phase-space systems. It is easily programmed on either an analog or digital computer. Both methods were used to solve several example problems. The quality of the analog output graph was not extremely good. This can be attributed to the poor response of the XY plotter available.

Several different types of viewing systems could be adapted. The system constructed for this work separated the graph centers by five and one half inches. This distance was selected because the response fitted into the margins required for thesis binding. The standard commercially available three-dimensional viewer used in topographical mapping is another example of a viewing device. The graph centers for this viewer are separated by only two and one half inches. This would require much smaller graphs and probably some type of photographic reduction process would be necessary. Another method of creating the three-dimensional illusion was also investigated. A single prospective graph was rotated by five degree increments and photographed with several frames of movie film. The resulting animated movie gave the "illusion" of depth by observing the system under rotation.

These techniques give valuable insight into the response of systems and can be most useful in analysis and design. The methods are not restricted to phase-space graphs and therefore would be quite useful in visualizing any three-dimensional object.

BIBLIOGRAPHY

Adarnowicz, Michael. "Graphpak I: A Three-Dimensional Manipulation Program." New York University, September 10, 1965.

Arens, Karen F. Plotter User's Guide. A publication of Virginia Polytechnic Institute Computing Center. March 1968.

Stoodley, G. R. "Digital Computer Generated Stereoscopic Pointille Surfaces." Publication of Grumman Research Department, April 1965.

Zajac, E. E. "Computer Animation - New Scientific and Educational Tool." Society of Motion Picture and Television Engineers. Volume 74, November 1965, pp. 1006-1008.

Zajac, E. E. "Computer Made Perspective Movies as a Scientific and Communication Tool." Communications of the Association for Computing Machinery. Volume 7, March 1964, pp. 169-170.

APPENDICES

THIS PROGRAM IS BROKEN DOWN INTO ONE MAIN ROUTINE AND TWO MAJOR SUBROUTINES. THE MAIN PROGRAM PLOTS THE SYSTEM TRAJECTORY FOR EACH EYE VIEW AND LABELS THE SCALE FACTOR AND FIGURE LEGEND. SUBROUTINE THRDM PLOTS THE COORDINATE AXES. SUBROUTINE BRUTF IS THE SYSTEM EQUATION SOLVER.

```
 IBFTC MAINT
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NCNL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
      DIMENSION XE(2),YE(2),ZE(2),WORD(13)
   31 FORMAT(I10)
   32 FORMAT(3F10.5)
  135 FORMAT(1H1,7X,6HANGLE1,9X,6HANGLE2,9X,6HANGLE3,9X,8HDI
     1STANCE/5X,  F10.5,5X,F10.5,5X,F10.5,5X,F10.5)
  137 FORMAT(10X,5HSCAX=,F10.6/10X,5HSCAY=,F10.6/10X,5HSCAZ=
     1,F10.6)
  139 FORMAT(13A6)
      CALL PLOT(3.,7.5,-3)
      CALL FACTOR(.78)
      READ(5,31) KSETS
      DO 39 M=1,KSETS
      K=1
      READ(5,32) ANG1,ANG2,DIST
      ANG3=360./3.14159*ARSIN(1.24/DIST)
      WRITE(6,135) ANG1,ANG2,ANG3, DIST
      CALL BRUTF(IT)
      READ(5,139) (WORD(I),I=1,13)
      WRITE(6,147)  (WORD(I),I=1,13)
  147 FORMAT(10X,13A6)
      XMX=0.
      YMX=0.
      ZMX=0.
      DO 33 J=1,IT
      IF(ABS(XX(J)).GT.XMX) XMX=ABS(XX(J))
      IF(ABS(YY(J)).GT.YMX) YMX=ABS(YY(J))
   33 IF(ABS(ZZ(J)).GT.ZMX) ZMX=ABS(ZZ(J))
      SCAX=1.5/XMX
      SCAY=1.5/YMX
      SCAZ=1.5/ZMX
      WRITE(6,137) SCAX,SCAY,SCAZ
      DO 34 J=1,IT
```

```
      XX(J)=XX(J)*SCAX
      YY(J)=YY(J)*SCAY
   34 ZZ(J)=ZZ(J)*SCAZ
      A1=SIN(ANG1*3.14159/180.)
      B1=COS(ANG1*3.14159/180.)
      A2=SIN(ANG2*3.14159/180.)
      B2=COS(ANG2*3.14159/180.)
      A3=SIN(ANG3*3.14159/180.)
      B3=COS(ANG3*3.14159/180.)
   37 CALL THRDM(K,ANG1,ANG2,ANG3,DIST)
      DO 36 I=1,IT
      XE(1)=YY(I)*B1-XX(I)*A1
      YE(1)=ZZ(I)*B2-YY(I)*A1*A2-XX(I)*B1*A2
      ZE(1)=ZZ(I)*A2+YY(I)*A1*B2+XX(I)*B1*B2
      XE(2)=XE(1)*B3-ZE(1)*A3
      YE(2)=YE(1)
      ZE(2)=XE(1)*A3+ZE(1)*B3
      XLL=DIST*XE(K)/(DIST-ZE(K))
      YLL=YE(K)*DIST/(DIST-ZE(K))
      IF(I.GT.1) GO TO 35
      CALL PLOT(XLL,YLL,3)
      GO TO 36
   35 CALL PLOT(XLL,YLL,2)
   36 CONTINUE
      SX=.5/SCAX
      SY=.5/SCAY
      SZ=.5/SCAZ
      CALL SYMBOL(-1.60,-2.60,.14,28H1 X DIVISION =
     1UNITS,0.,28)
      CALL NUMBER(0.3,-2.60,.14,SX,0.,3)
      CALL SYMBOL(-1.60,-2.85,.14,28H1 Y DIVISION =
     1UNITS,0.,28)
      CALL NUMBER(0.3,-2.85,.14,SY,0.,3)
      CALL SYMBOL(-1.60,-3.10,.14,28H1 Z DIVISION =
     1UNITS,0.,28)
      CALL NUMBER(0.3,-3.10,.14,SZ,0.,3)
   41 K=K+1
      IF(K.GT.2) GO TO 39
      GO TO 37
   39 CALL SYMBOL(-8.00,-4.90,.14,WORD(1),0.,78)
      CALL PLOT(0.,0.,-4)
   99 STOP
      END
```

```
IBFTC THRDM
      SUBROUTINE THRDM (K,ANG1,ANG2,ANG3,DIST)
      DIMENSION SHFT(2),X(3),Y(3),Z(3),N(3),XE(2),YE(2),ZE(2
```

```
   1)
    N(1)=04
    N(2)=09
    N(3)=08
    A1=SIN(ANG1*3.14159/180.)
    B1=COS(ANG1*3.14159/180.)
    A2=SIN(ANG2*3.14159/180.)
    B2=COS(ANG2*3.14159/180.)
    A3=SIN(ANG3*3.14159/180.)
    B3=COS(ANG3*3.14159/180.)
    DO 10 I=1,3
    X(I)=0.
    Y(I)=0.
 10 Z(I)=0.
    X(1)=2.
    Y(2)=2.
    Z(3)=2.
    SHFT(1)=9.0
    SHFT(2)=6.95
    CALL PLOT(SHFT(K),0.,-3)
    DO 13 I=1,3
    XEL= Y(I)*B1-X(I)*A1
    YEL= Z(I)*B2-Y(I)*A1*A2-X(I)*B1*A2
    ZEL= Z(I)*A2+Y(I)*A1*B2+X(I)*B1*B2
    XER= XEL*B3-ZEL*A3
    YER= YEL
    ZER= XEL*A3+ZEL*B3
    RJ=-1
    XE(1)= RJ*XEL
    YE(1)= RJ*YEL
    ZE(1)= RJ*ZEL
    XE(2)= RJ*XER
    YE(2) =RJ*YER
    ZE(2)= RJ*ZER
    XLL=DIST*XE(K)/(DIST-ZE(K))
    YLL=YE(K)*DIST/(DIST-ZE(K))
    CALL PLOT(XLL,YLL,3)
    RJ=1.
    XE(1)= RJ*XEL
    YE(1)= RJ*YEL
    ZE(1)= RJ*ZEL
    XE(2)= RJ*XER
    YE(2) =RJ*YER
    ZE(2)= RJ*ZER
    XLL=DIST*XE(K)/(DIST-ZE(K))
    YLL=YE(K)*DIST/(DIST-ZE(K))
    CALL PLOT(XLL,YLL,2)
    IF(ABS(XLL).LT.(.0001*ABS(YLL))) GO TO 11
    WELL=YLL/XLL
    ANGL=ATAN(WELL)
```

```
   11 IF(ABS(XLL).LT.(.0001*ABS(YLL))) ANGL=1.570796
      ANGLE= 180.*ANGL/3.141592
      XLL=1.1*XLL
      IF(XLL.LT.0.) XLL=1.15*XLL
      YLL=1.1*YLL
      CALL SYMBOL(XLL,YLL,.09,N(I),0.,-1)
      DO 12 J1=1,4
      SJ=5-J1
      RJ=SJ/4.
      XE(1)= RJ*XEL
      YE(1)= RJ*YEL
      ZE(1)= RJ*ZEL
      XE(2)= RJ*XER
      YE(2) =RJ*YER
      ZE(2)= RJ*ZER
      XLL=DIST*XE(K)/(DIST-ZE(K))
      YLL=YE(K)*DIST/(DIST-ZE(K))
   12 CALL SYMBOL(XLL,YLL,.09,13,ANGLE,-1)
      DO 13 J1=1,4
      SJ=-J1
      RJ=SJ/4.
      XE(1)= RJ*XEL
      YE(1)= RJ*YEL
      ZE(1)= RJ*ZEL
      XE(2)= RJ*XER
      YE(2) =RJ*YER
      ZE(2)= RJ*ZER
      XLL=DIST*XE(K)/(DIST-ZE(K))
      YLL=YE(K)*DIST/(DIST-ZE(K))
   13 CALL SYMBOL(XLL,YLL,.09,13,ANGLE,-1)
      RETURN
      END
```

```
IBFTC BRUTF
      SUBROUTINE BRUTF (IT)
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NONL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
 1000 FORMAT(I10,30X,I10)
 1001 FORMAT (5F10.5)
 1002 FORMAT(1H1,9X,4HTIME,15X,4HY  1,16X,4HY  2,16X,4HY  3,
     116X,4HY  4, 16X,4HY  5/)
 1003 FORMAT (1H0,F15.5,5F20.5)
 1004 FORMAT (F15.5)
 1005 FORMAT(/5X,25HNUMBER OF STATE VARIABLES,20X,12HNONLINE
     1ARITY/I10,  40X,I7)
 1006 FORMAT(/5X,8HA-MATRIX)
```

```
1007 FORMAT(/5X,8HB-MATRIX)
1008 FORMAT(/5X,26HINITIAL CONDITIONS ON X(I)/10F12.5)
1009 FORMAT(/5X,17HDRIVER PARAMETERS/4F12.5)
1010 FORMAT(/5X,20HINTEGRATION INTERVAL/F15.5)
1011 FORMAT(/5X,22HTOTAL INTEGRATION TIME/F15.5)
1012 FORMAT(/5X,21HOUTPUT PRINT INTERVAL/F15.5)
1013 FORMAT(/5X,26HNUMBER OF OUTPUT VARIABLES/I10)
1014 FORMAT(/5X,8HC-MATRIX)
1015 FORMAT(/5X, 8HD-MATRIX)
1016 FORMAT(10F12.5)
1017 FORMAT(9X,60HYOUR STATE VARIABLES ARE GETTING PRITTY C
    1OTTON PICKINLARGE       )
   5 READ(5,1000) NVAR, NONL
     DO 10 I=1,NVAR
     DO 10 J=1,NVAR,5
  10 READ(5,1001)   A(I,J),A(I,J+1),A(I,J+2),A(I,J+3),A(I,J
    1+4)
     DO 20 I=1,NVAR
  20 READ(5,1001) B(I,1),B(I,2), B(I,3), B(I,4)
     DO 30 I=1,NVAR,5
  30 READ(5,1001) X(I), X(I+1), X(I+2), X(I+3), X(I+4)
     READ(5,1001) PARAM(1),PARAM(2),PARAM(3),PARAM(4)
     READ(5,1004) DELTA
     READ(5,1004) TMAX
     READ(5,1004) TSAMP
     NTH=TSAMP/DELTA+.001
     READ(5,1000) NSPEC
     DO 40 I=1,NSPEC
     DO 40 J=1,NVAR,5
  40 READ(5,1001) C(I,J),C(I,J+1),C(I,J+2),C(I,J+3),C(I,J+4
    1)
     DO 42 I=1,NSPEC
  42 READ(5,1001) D(I,1), D(I,2), D(I,3), D(I,4)
  47 WRITE(6,1005) NVAR,NONL
     WRITE(6,1006)
     DO 43 I=1,NVAR
  43 WRITE(6,1016) (A(I,J), J=1,NVAR)
     WRITE(6,1007)
     DO 44 I=1,NVAR
  44 WRITE(6,1016)B(I,1),B(I,2), B(I,3), B(I,4)
     WRITE(6,1008) (X(I), I=1,NVAR)
     WRITE(6,1009)PARAM(1),PARAM(2),PARAM(3),PARAM(4)
     WRITE(6,1010)   DELTA
     WRITE(6,1011)  TMAX
     WRITE(6,1012) TSAMP
     WRITE(6,1013)  NSPEC
     WRITE(6,1014)
     DO 45 I=1,NSPEC
  45 WRITE(6,1016) (C(I,J), J=1,NVAR)
     WRITE(6,1015)
```

```
      DO 46 I=1,NSPEC
   46 WRITE(6,1016)D(I,1), D(I,2), D(I,3), D(I,4)
      DO 50 I=1,NVAR
   50 QU(I)=0.0
      WRITE(6,1002)
      T=0.0
      T1=0.0
      CALL SUBDE1
      CALL SUBDE3
      IT=1
      WRITE(6,1003) T,(Y(I), I=1,NSPEC)
      XX(IT)=Y(1)
      YY(IT)=Y(2)
      ZZ(IT)=Y(3)
      NCK=1
   60 T1=T
      CALL SUBDE1
      P=0.5
      Q=2.0
      R=0.5
      CALL SUBDE2
      T1=T+DELTA/2.0
      CALL SUBDE1
      P=0.2928932
      Q=1.0
      R=0.2928932
      CALL SUBDE2
      P=1.7071068
      R=1.7071068
      CALL SUBDE2
      T1=T+DELTA
      CALL SUBDE1
      P=0.16666667
      Q=2.0
      R=0.5
      CALL SUBDE2
      T=T+DELTA
      IF(T.GT.TMAX) GO TO 105
      IF(NCK.LT.NTH) GO TO 75
      TIT=IT
      T=TIT*TSAMP
   80 CALL SUBDE3
      WRITE(6,1003) T,(Y(I), I=1,NSPEC)
      DO 49 I=1,NVAR
      IF(ABS(X(I)).GT.99999.)GO TO 104
   49 CONTINUE
      IT=IT+1
      XX(IT)=Y(1)
      YY(IT)=Y(2)
      ZZ(IT)=Y(3)
```

```
      NCK=0
   75 NCK=NCK+1
      GO TO 60
  104 WRITE(6,1017)
  105 CONTINUE
  106 RETURN
      END
$IBFTC SUBDE1
      SUBROUTINE SUBDE1
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NONL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
      IF (T1-PARAM(1)) 400, 410, 410
  400 E(1)=0.0
      GO TO 420
  410 E(1)=1.0
  420 IF (T1-PARAM(2)) 430, 440, 440
  430 E(2)=0.0
      GO TO 450
  440 E(2)=1.0
  450 E(3)=SIN(6.2831852*PARAM(3)*T1)
      E(4)=COS(6.2831852*PARAM(4)*T1)
      RETURN
      END


$IBFTC SUBDE2
      SUBROUTINE SUBDE2
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NONL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
      DIMENSION DXDT(10),XPRIM(10)
      IF(NONL.LT.1) GO TO 505
      CALL SUBDE4
  505 DO 510 J=1,NVAR
      DXDT(J)=0.0
      DO 510 K=1,NVAR
  510 DXDT(J)=DXDT(J)+A(J,K)*X(K)
      DO 515 J=1,NVAR
      DO 515 K=1,4
  515 DXDT(J)=DXDT(J)+B(J,K)*E(K)
      DO 520 J=1,NVAR
  520 DXDT(J)=DELTA*DXDT(J)
      DO 525 J=1,NVAR
      XPRIM(J)=P*(DXDT(J)-Q*QU(J))
      X(J)=X(J)+XPRIM(J)
  525 QU(J)=QU(J)+3.0*XPRIM(J)-R*DXDT(J)
      RETURN
      END
```

```
 IBFTC SUBDE3
      SUBROUTINE SUBDE3
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NONL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
      DO 90 I=1,NSPEC
      Y(I)=0.0
      DO 90 J=1,NVAR
   90 Y(I)=Y(I)+C(I,J)*X(J)
      DO 95 I=1,NSPEC
      DO 95 J=1,4
   95 Y(I)=Y(I)+D(I,J)*E(J)
      RETURN
      END

 IBFTC SUBDE4
      SUBROUTINE SUBDE4
      COMMONX(10),Y(5),A(10,10),B(10,4),C(5,10),D(5,4),NONL,
     1PARAM(4),T1,QU(10),NVAR,NSPEC,TSAMP,P,Q,R,DELTA,E(4),X
     25,10),D(5,4),NONL,PARAM(4),T1,
C     ENTER NONLINEARITY CHANGE******************BEGIN
C     ENTER NONLINEARITY CHANGE******************END
      RETURN
      END
```

ANGLE1	ANGLE2	ANGLE3	DISTANCE
30.00000	15.00000	7.10924	20.00000

NUMBER OF STATE VARIABLES
3

NONLINEARITY
-0

A-MATRIX

0.00000	1.00000	0.00000
0.00000	0.00000	1.00000
-0.50000	-1.00000	-0.50000

B-MATRIX

0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000

INITIAL CONDITIONS ON X(I)

4.00000	0.00000	0.00000

DRIVER PARAMETERS

0.00000	0.00000	0.00000	0.00000

INTEGRATION INTERVAL
0.05000

TOTAL INTEGRATION TIME
7.00000

OUTPUT PRINT INTERVAL
0.10000

NUMBER OF OUTPUT VARIABLES
3

C-MATRIX

1.00000	0.00000	0.00000
0.00000	1.00000	0.00000
0.00000	0.00000	1.00000

D-MATRIX

0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000

TIME	Y 1	Y 2	Y 3
0.00000	4.00000	0.00000	0.00000
0.10000	3.99967	-0.00983	-0.19475
0.20000	3.99740	-0.03857	-0.37805
0.30000	3.99137	-0.08501	-0.54854
0.40000	3.97986	-0.14781	-0.70493
0.50000	3.96131	-0.22549	-0.84611
0.60000	3.93431	-0.31649	-0.97104
0.70000	3.89762	-0.41913	-1.07887
0.80000	3.85016	-0.53167	-1.16888
0.90000	3.79102	-0.65229	-1.24051
1.00000	3.71949	-0.77914	-1.29337
1.10000	3.63505	-0.91033	-1.32725
1.20000	3.53734	-1.04396	-1.34210
1.30000	3.42624	-1.17812	-1.33805
1.40000	3.30176	-1.31095	-1.31542
1.50000	3.16415	-1.44060	-1.27469
1.60000	3.01381	-1.56530	-1.21649
1.70000	2.85131	-1.68335	-1.14166
1.80000	2.67742	-1.79311	-1.05114
1.90000	2.49302	-1.89309	-0.94606
2.00000	2.29917	-1.98188	-0.82765
2.10000	2.09706	-2.05822	-0.69731
2.20000	1.88798	-2.12099	-0.55650
2.30000	1.67334	-2.16922	-0.40680

2.40000	1.45465	-2.20211	-0.24987
2.50000	1.23346	-2.21901	-0.08744
2.60000	1.01139	-2.21947	0.07873
2.70000	0.79012	-2.20320	0.24684
2.80000	0.57131	-2.17010	0.41507
2.90000	0.35666	-2.12024	0.58162
3.00000	0.14782	-2.05389	0.74470
3.10000	-0.05358	-1.97148	0.90258
3.20000	-0.24596	-1.87360	1.05355
3.30000	-0.42782	-1.76105	1.19601
3.40000	-0.59772	-1.63474	1.32845
3.50000	-0.75434	-1.49574	1.44944
3.60000	-0.89648	-1.34527	1.55768
3.70000	-1.02306	-1.18467	1.65200
3.80000	-1.13312	-1.01537	1.73140
3.90000	-1.22589	-0.83892	1.79499
4.00000	-1.30072	-0.65692	1.84206
4.10000	-1.35715	-0.47107	1.87209
4.20000	-1.39487	-0.28309	1.88469
4.30000	-1.41375	-0.09472	1.87969
4.40000	-1.41386	0.09226	1.85707
4.50000	-1.39540	0.27611	1.81700
4.60000	-1.35880	0.45509	1.75983
4.70000	-1.30460	0.62752	1.68608
4.80000	-1.23356	0.79178	1.59644

The vita has been removed from the scanned document

THREE-DIMENSIONAL DISPLAY OF PHASE SPACE DIAGRAMS

Donald L. English

ABSTRACT

Phase plane analysis of linear and nonlinear systems has been widely in recent years. These graphical displays give considerable insight into system response. A serious short coming of this procedure is the limitation to two dimensions. A third-order system must be studied on two, two-dimensional graphs or represented by a second-order mathematical approximation for the third-order system. For nonlinear systems employing relays, saturation, etc., the approximations must be made with care.

In an effort to reduce discrepancies and improve the visualization of system response, this paper presents a method to obtain three-dimensional displays of system graphs. Third-order phase graphs of nonlinear and linear systems can be graphed directly without approximations. A discussion of the principles involved will be presented first. Then, analog and digital programs will be given. Finally, example problems will be discussed.