Spring 5-1992

# Digital Model of a Generic Infrared Tracker

James Michael Brown

# DIGITAL MODEL OF A
# GENERIC INFRARED TRACKER

A Thesis

Submitted to the Faculty

of

Rose-Hulman Institute of Technology

by

James Michael Brown

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical Engineering

May 1992

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

**Graduate Council**

# APPOINTMENT OF FINAL EXAMINATION COMMITTEE AND FINAL EXAMINATION REPORT

Student JAMES BROWN Degree MSEE

Department ELECTRICAL AND COMPUTER ENG. Date MARCH 6, 1992

☒ Thesis Title DIGITAL MODEL OF A GENERIC INFRARED TRACKER

☐ Non-Thesis

## I. EXAMINATION COMMITTEE

**It is requested that the following committee be appointed to conduct the final examination of the student named above.**

| Professor | Department |
|---|---|
| FREDERICK C. BROCKHURST Chairman | ECE |
| PAUL D. SMITH | ECE |
| ROGER LAUTZENHEISER | MATH |
| DAVID J. PURDY | ME |

Requested by Frederick C. Brockhurst Approved by Buck F. Brown

Major Professor Dept. Chairman

## II. FINAL EXAMINATION REPORT

☒ Passed with recommendation for doctoral study

☐ Passed

☐ Failed

Date of Exam May 7, 1992 Committee Signatures Frederick C. Brockhurst Chm.

Roger Lautzenheiser

David J. Purdy

P. D. Smith

Note: When the report is complete, the Chairman of the Examining Committee should send this form to the Director of Graduate Studies. Copies will be returned to the Chairman of the Advisory Committee, Department Chairman, and student.

# ABSTRACT

Brown, James Michael. M.S., Rose-Hulman Institute of Technology, May 1992. Digital Model of a Generic Infrared Tracker. Major Professor: Frederick C. Brockhurst.

Personal computer based simulations are popular because of the wide availability of personal computers. In this thesis, a simple simulation model of a generic infrared tracker is developed for the personal computer.

This thesis first centers on the definition of a generic infrared tracker and the development of a tracker model. The remainder of this thesis is a description of the personal computer based computer simulation model, and a discussion of results of simulation runs to characterize the tracker model response.

The conclusion of this thesis is that the simple simulation model serves both as a prototype for developing a simulation model of a real tracker, and as a simple introduction to infrared trackers.

# DISCLAIMER

This thesis is being submitted as partial fulfillment of the requirements of Rose-Hulman Institute of Technology needed to obtain a Master of Science in Electrical Engineering Degree.

The conclusions and opinions expressed in this thesis are those of the author and do not necessarily represent the position of Rose-Hulman Institute of Technology or the United States Government, or any of its directors, officers, agents, or employees about the matters discussed.

## ACKNOWLEDGMENTS

This thesis was completed with the assistance of many people, several of whom I can acknowledge here. First I would like to thank my Rose-Hulman faculty advisor, Dr. Frederick C. Brockhurst, for his guidance throughout the thesis project. Thanks also to the other members of my advisory committee, Dr. Paul D. Smith, Dr. Roger G. Lautzenheiser, and Dr. David J. Purdy.

I would like to acknowledge the management of the Naval Surface Warfare Center, Crane Division, who provided the opportunity to participate in this educational experience. I would especially like to thank the following Crane personnel for their assistance: Mr. Carl Lohkamp, Mr. Jerry Kemp, Mr. Norman Papke, Mr. Mike Walls, Mr. Tim Bradley, Mr. Cabot Faultless, Mr. Jeff Campbell, Mr. Brent Waggoner, Mr. Gary Milhoan, Mr. Jeff Johann, and Mr. Gordon Miller. Thanks also to former Crane employee Mrs. Karen Moen.

Finally, I would like to thank my family for their support during my participation in this program.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AGC | Automatic Gain Control |
| AM | Amplitude Modulation |
| DC | Direct Current |
| FORTRAN | FORmula TRANslation |
| Gyro | Gyroscope |
| Hz | Hertz (cycles per second) |
| IMSL | International Mathematical and Statistical Library |
| IR | Infrared |
| kg | Kilogram |
| m | Meter |
| N | Newton |
| PC | Personal Computer |
| RETPHA | Model Reticle Phasing |
| TGTAZ | Model Target Azimuth |
| TGTEL | Model Target Elevation |
| XGAIN | Model Loop Gain |

# I. INTRODUCTION

One technique of evaluating the effectiveness of infrared trackers is digital simulation of the tracker system. Digital simulation is inexpensive when compared with the costs associated with testing real tracker hardware. Digital simulation modeling does not replace hardware testing, but rather complements the necessary hardware testing required to characterize existing tracker hardware.

The objective of this thesis is to construct a simple simulation model of a generic infrared tracker that runs on a personal computer. This simple model serves two purposes:

1. As a simple introduction to infrared trackers, and
2. As a prototype for developing a simulation model of a real tracker.

The simulation design process consists of four distinct procedures [1]. The first phase involves a description of the system to be simulated. The second procedure is to design a mathematical model of the system. The third procedure is to design the simulation model. The fourth procedure is test and evaluation of the simulation.

In this thesis, Section II contains the description of the generic infrared tracker. The generic infrared tracker is a composite of several different trackers described in the literature.

Section III describes the mathematical model of the generic infrared tracker.

Section IV contains the description of the simulation model. The simulation model is written in FORTRAN (FORmula TRANslation).

Section V deals with characterization of the simulation model response. Since this tracker is only a conceptual tracker, there does not exist any data upon which to base a test and evaluation of the simulation. Section V includes a discussion of the results of several simulation runs to characterize tracker response.

Section VI presents the conclusions and recommendations of this thesis.

## II. GENERIC TRACKER DESCRIPTION

The simulation design process consists of four distinct procedures [1]. The first procedure of the four phase simulation design process involves a description of the system to be simulated [1]. This section contains a description of the generic tracker.

The generic tracker used for this thesis is a composite of the trackers described by Wolfe and Zissis [2], Pignataro [3], Carey [4], Hudson [5], and Dow [6]. The diagram of a basic reticle tracker is shown in Figure 1 [2]. The major components of the generic tracker are the reticle, optics, infrared (IR) detector, tracker electronics, and the position drive.

The reticle for the generic tracker is the rising sun reticle shown in Figure 2 [2]. The purpose of the spinning reticle is to modulate the infrared energy that reaches the detector so the generic detector output contains the encoded target position information.

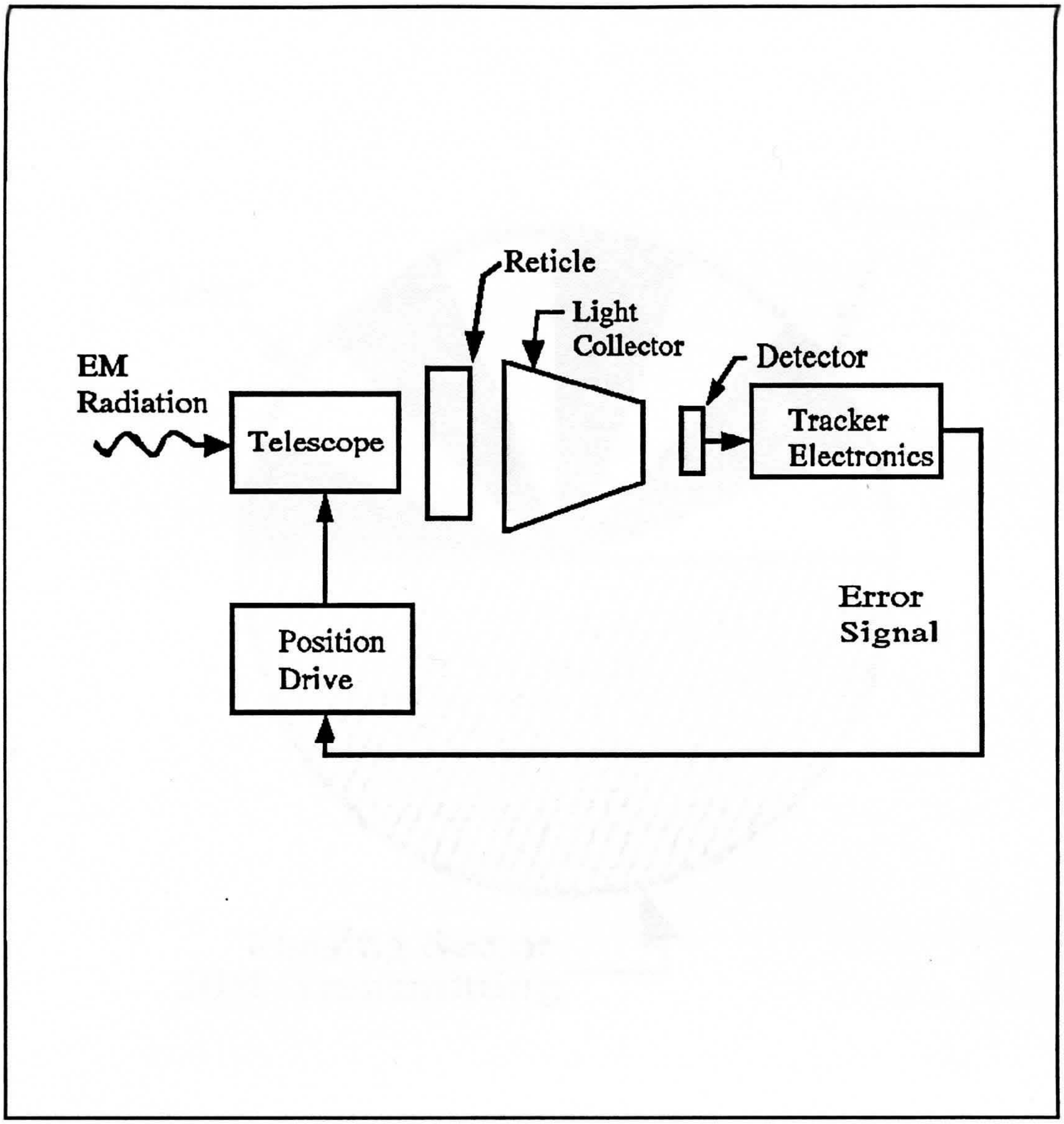



Figure 1. Basic Reticle Tracker. [2]

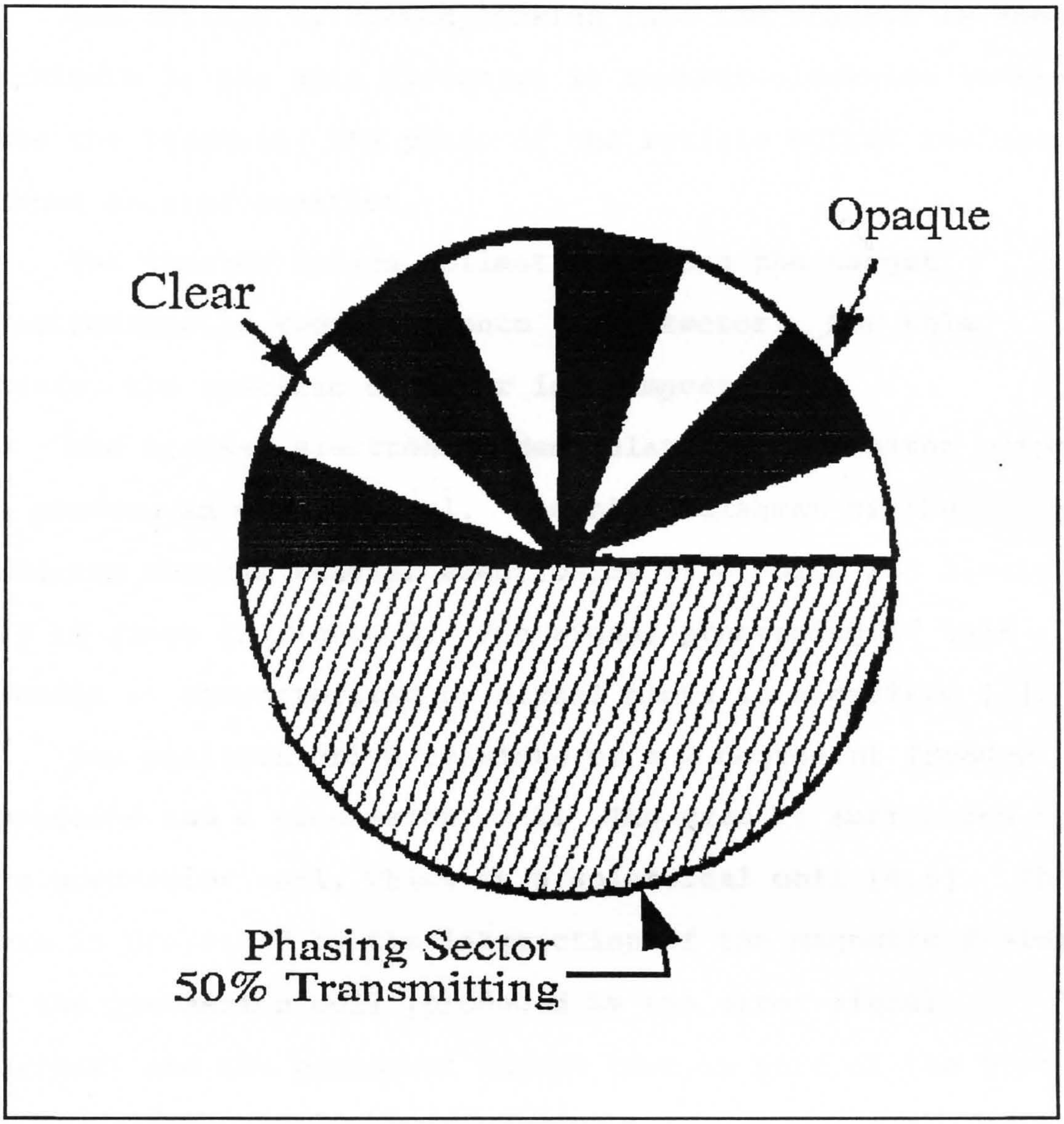



Figure 2. Rising Sun Reticle. [2]

The reticle as viewed looking into the tracker is shown in Figure 2; the spin direction is counter-clockwise looking into the tracker. The phase of the reticle output indicates target angular position.

The tracker optics collect and focus the target electromagnetic radiation onto the detector. For this thesis, the specific detector is unimportant.

The tracker electronics demodulates the detector output to produce an error signal. The block diagram of the rotating reticle tracker electronics from Wolfe and Zissis [2] is shown in Figure 3. The signal processing of this tracker is characterized by signal plots in Pignataro [3].

The position drive consists of a 2 degree of freedom gyroscope and a precession coil. The gyro is surrounded by the precession coil, which is a solenoidal coil [4,6]. The gyro is precessed by the interaction of the magnetic fields of the precession coil (produced by the error signal current) and the permanent magnet that is part of the gyro [4,6].

The precession coil is a solenoidal coil essentially in the plane perpendicular to the gyro spin axis [4,6]. The current in the precession signal produces a magnetic field

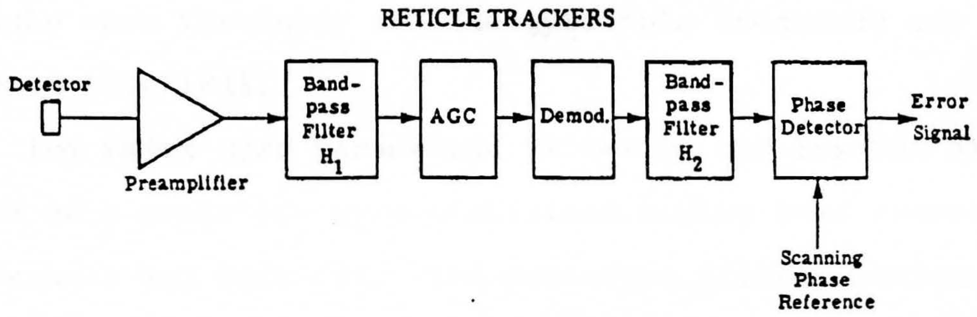



Figure 3. Rotating Reticle Tracker Electronics. [2]

that applies torque to the generic gyro, which contains a permanent magnet perpendicular to the gyro spin axis [4,6], to track a single target in the field-of-view.

The gyroscope is a composite of the gyroscopes described by Scarborough [7], Friedland [8], and Garnell and East [9]. For mechanical construction simplicity, the reticle spin frequency and the gyro spin frequency are assumed identical.

The major gyro parametric values of the tracker are based on a prototype gyro-stabilized homing head presented by Garnell and East [9]. The prototype tracker maximum open loop tracking rate is 10 degrees per second, and the prototype tracker gyro spin rate is 10,000 revolutions per minute (i.e. 166.7 Hz or 1047.2 radians per second).

The gyro parameters C (moment of inertia about the spin axis), and A (moment of inertia about the transverse axis) are based on the prototype gyro in Garnell and East [9]. Assuming the gyro shape is a right circular cylinder, the values of C and A are calculated as follows [10]:

$$C=\frac{m\,r^2}{2} \qquad (1)$$

$$A=m\frac{(3r^2+h^2)}{12} \qquad (2)$$

where m is the mass, r is the radius, and h is the height of the right circular cylinder. For m=.7 kilograms (kg), r=.054 meters (m), and h=.027 meters, then A=.05528 kg-$m^2$ and C=.1021 kg-$m^2$.

# III. GENERIC TRACKER MODEL

The second procedure of the four phase simulation design process is to design a mathematical model of the system [1]. This section contains a description of the tracker model. The generic tracker model is an approximation of the generic tracker described in Section II.

The block diagram of the generic tracker model is shown in Figure 4. The major components of the model are the following: scene model, reticle model, electronics model, and gyro model.

The scene model is described in Section III.A. The scene model contains the target position and intensity information necessary for the reticle model to calculate the detector output.

The reticle model is described in Section III.B. The reticle model accounts for the tracker optics, reticle, and detector described in Section II.

The electronics model is described in Section III.C. The electronics model processes the output of the reticle model to produce the error signal.

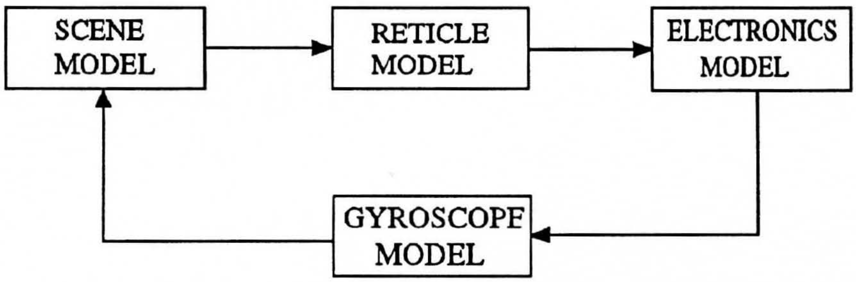



**Figure 4. Block Diagram of Generic Tracker Model.**

The gyroscope model is described in Section III.D. The gyroscope model includes both the tracker precession coil and the tracker gyroscope described in Section II. The gyroscope model uses the error signal output from the electronics model to precess the gyroscope.

## A. SCENE MODEL

The scene model is described in this section. The scene model contains the target position and intensity information necessary for the reticle model to calculate the detector output.

The target position is given by the target azimuth and elevation angles relative to a fixed coordinate system. The gyro spin axis direction is given by the gyro azimuth and elevation angles relative to a fixed coordinate system.

For this thesis, the background contribution to the scene is assumed zero. Also, the tracker gyroscope center of gravity is assumed to be stationary. More details on the gyroscope are in Section III.D.1. The target is assumed to be a point source. The scene model provides both open and closed loop capability.

## B. RETICLE MODEL

The reticle model is described in this section. The reticle model accounts for the tracker optics, reticle, and detector described in Section II.

The reticle model calculates the reticle output voltage based on the position information of the infrared source from the scene model. The reticle model is for a single target; however, superposition could be assumed for multiple targets. The reticle model allows the target blur circle produced by the optics to be modeled as either a blur circle or a point. For simplicity, the reticle model output is assumed independent of target intensity, and is assumed to have a minimum value of 0 and a maximum value of 1 volt.

The initial time zero reticle position determines the reticle modulation phasing. This phasing is one of the critical parameters of the tracker closed loop performance. More details on reticle phasing are contained in Sections V.C and V.D.

C. ELECTRONICS MODEL

This section contains a description of the electronics model. The electronics model processes the output of the reticle model to produce the error signal.

The bandpass filter H1 in Figure 3 was modeled by the transfer function

$$G(s) = \frac{T_1 s}{(T_3 s+1)(T_4 s+1)} \quad (3)$$

where the bandpass is from $1/T_3$ to $1/T_4$. The bandpass H1 filter poles were chosen based on the prototype gyro spin frequency of 10,000 revolutions per minute (i.e. 166.7 Hz or 1047.2 radians per second). Since the reticle has 4 transparent spokes in one half of the reticle, the carrier frequency is 1333.6 Hz (8*166.7). The bandpass H1 filter response is a peak at 1333.6 Hz.

The demodulator was modeled as shown in Figure 5 [11], and an ideal diode was assumed. When the diode is not conducting, the output O(t) decays exponentially with time constant $t=R_2*C$. When the diode is conducting, the transfer function is approximately the lowpass filter (assuming $R_2$ is much greater than $R_1$):

$$G(s) = \frac{O(s)}{I(s)} \approx \frac{1}{ts+1} \quad (4)$$

where the time constant is $t=R_1*C$. The demodulator

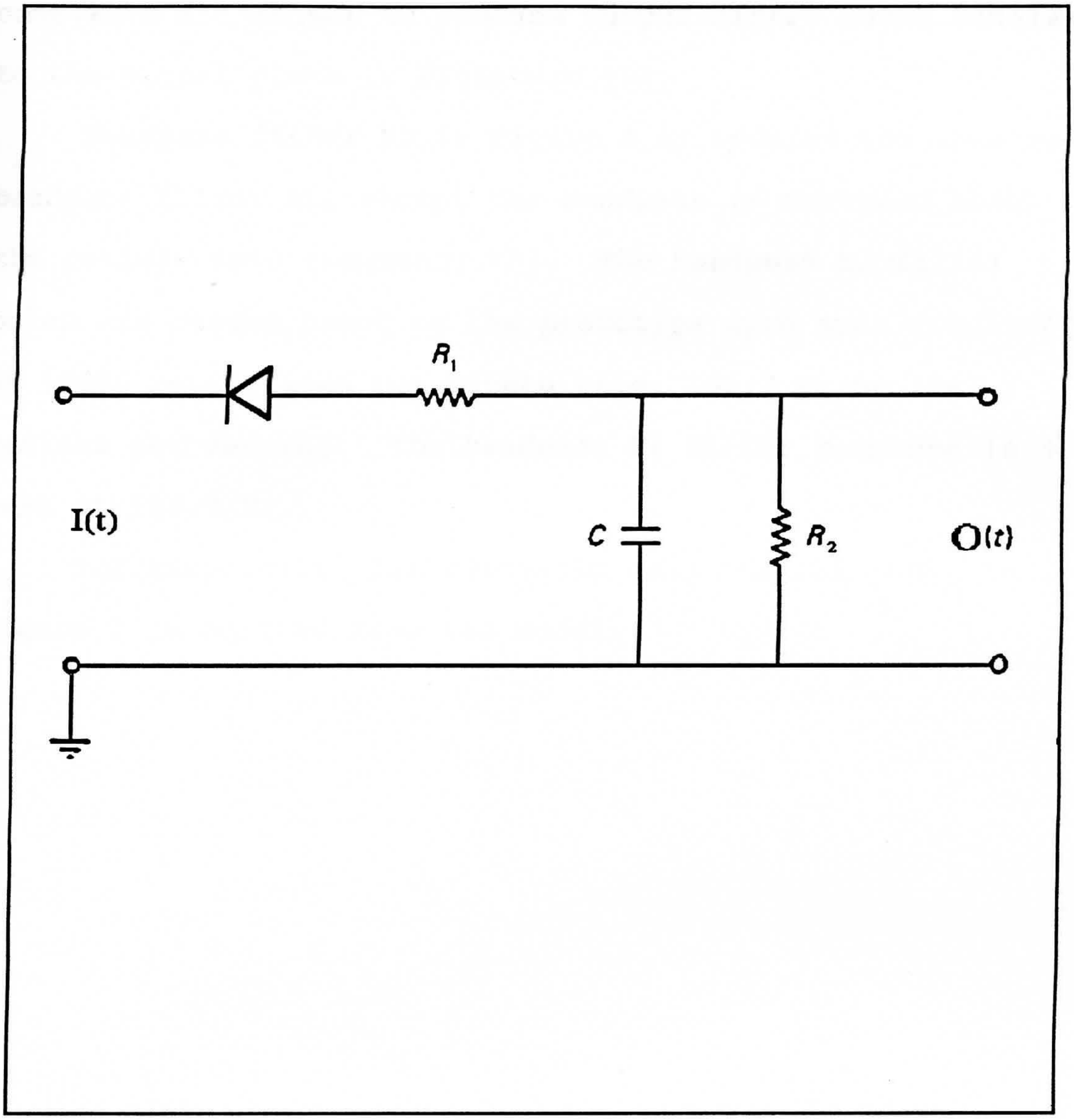



Figure 5. Demodulator Schematic. [11]

constants are chosen to produce output signal forms similar to the signal plots in Pignataro [3].

Bandpass filter H2 in Figure 3 is modeled the same as bandpass filter H1, except the bandpass is centered about the reticle spin frequency [2]. The bandpass H2 filter poles are chosen based on the prototype gyro spin frequency of 10000 revolutions per minute (i.e. 166.7 Hz or 1047.2 radians per second). The bandpass H2 filter response is a peak at 166.7 Hz.

*For simplicity, the automatic gain control (AGC) in Figure 3 is omitted from the model.*

## D. GYROSCOPE MODEL

This section contains a description of the gyroscope model. The gyroscope model includes both the tracker precession coil and the tracker gyroscope described in Section II. The gyroscope model uses the error signal output from the electronics model to precess the gyroscope.

The precession coil is modeled by assuming the torque on the gyro (due to the current in the precession coil) is proportional to the error signal output from the tracker electronics model. The gyro precession is similar to armature motion in a field controlled direct current motor. For a field controlled DC motor, if the armature current is constant then the torque (in the unsaturated region) is directly proportional to the field current [12]. The permanent magnet on the spinning gyro is equivalent to constant armature current in the field controlled dc motor [13].

The gyroscope model used for this project is a composite of the gyroscopes described by Scarborough [7], Friedland [8], and Garnell and East [9], and is described in the following section.

## 1. GYROSCOPE DIFFERENTIAL EQUATIONS OF MOTION

This section includes a discussion of the gyroscope differential equations of motion. The gyroscope model differential equations of motion used for this thesis are from Scarborough [7].

The gyroscope differential equations of motion are given by Scarborough [7] as the following (for a rotating body dynamically symmetrical about the spin axis Z):

$$A\dot{\omega}_x + (C-A)\omega_y\omega_z = M_x \quad (5)$$

$$A\dot{\omega}_y + (A-C)\omega_z\omega_y = M_y \quad (6)$$

$$C\dot{\omega}_z = M_z \quad (7)$$

These equations assume the following:

1.Ideal gyroscope (i.e. perfectly constructed and no friction).

2. Axis of spin is axis of cylinder and always passes through the gyro center of gravity (i.e. gyro center of gravity mounted at intersection of gimbal axes).

3. The rotating body is dynamically symmetrical about the spin axis (Z axis in Figure 6).

4. C is the moment of inertia about the spin axis, and A is the moment of inertia about any axis through the center of gravity and perpendicular to the axis of the cylinder.

5. $M_x$, $M_y$, and $M_z$ are the external moments about the center of gravity of all the external forces acting on the body, where

$$\vec{M} = M_x\hat{i} + M_y\hat{j} + M_z\hat{k} \tag{8}$$

and $\hat{i}$, $\hat{j}$, and $\hat{k}$ are the unit vectors along the X, Y, and Z axis (see Figure 6) of the spinning gyro coordinate system.

6. The angular velocities about the gyro X, Y, and Z axis are $\omega_x$, $\omega_y$, and $\omega_z$ respectively.

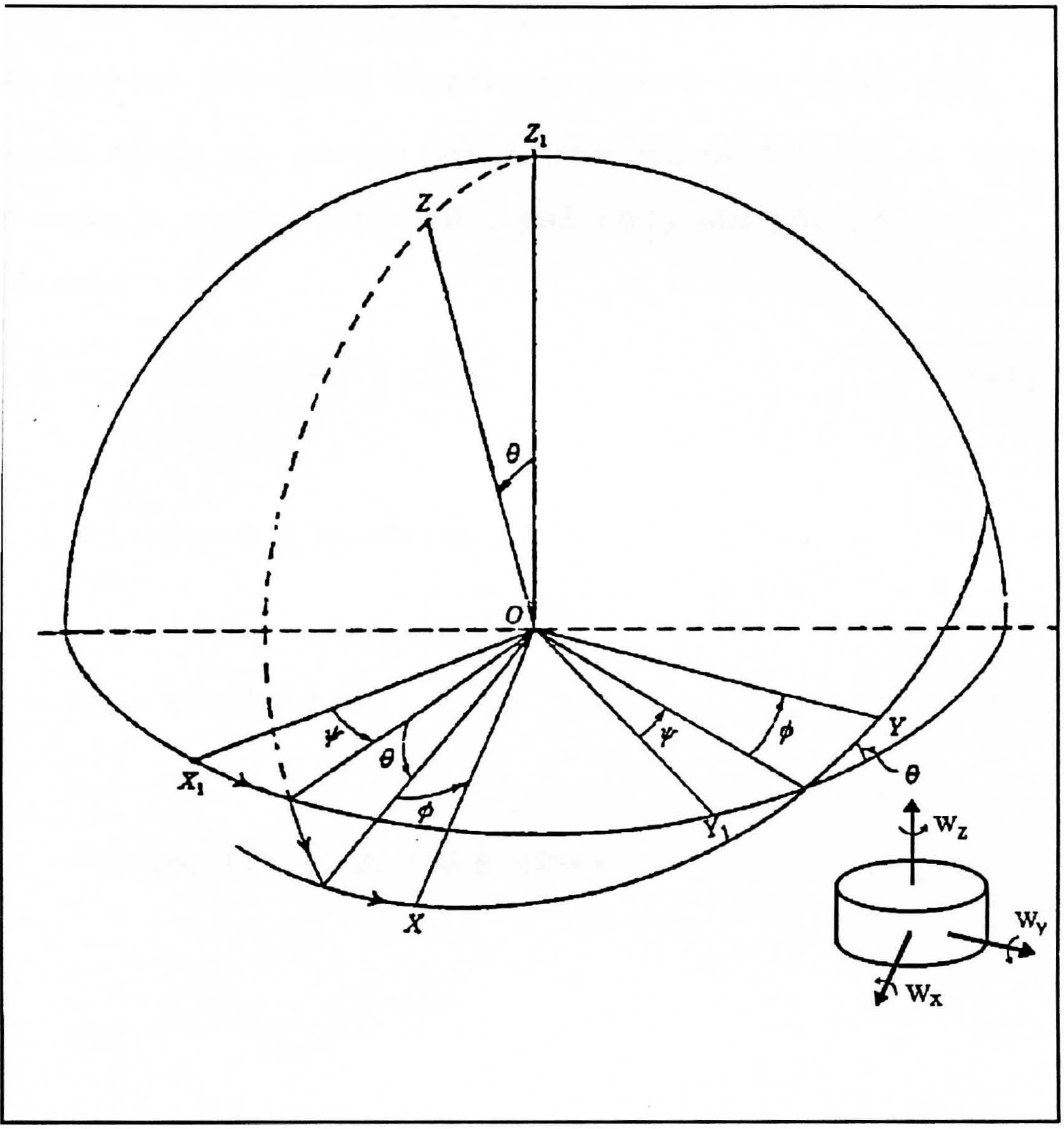
7. The right handed coordinate system is shown in Figure 6, with z the spin axis.

Three additional simplifying assumptions are made. First, Friedland [8] assumes the gyro is maintained at a constant spin velocity $\omega_z$ by external means (i.e. $M_z$ is assumed to be zero). Typically, the tracker performance is only of interest for a short time (say 5 seconds or less). For these short time periods, the constant spin velocity assumption appears reasonable. Thus the gyroscope differential equations of motion reduce to the following:

$$A\dot{\omega}_x + (C-A)\omega_y\omega_z = M_x \tag{9}$$

$$A\dot{\omega}_y + (A-C)\omega_z\omega_x = M_y \tag{10}$$

where $\omega_z$ is a constant.




.gure 6. Moving and Fixed Axis Coordinate Systems. [7]

The orientation of a rigid body moving about a fixed point is best described by Euler's angles [7]. Scarborough [7] relates the fixed coordinate system $O\text{-}X_1\text{-}Y_1\text{-}Z_1$ (see Figure 6) to the moving coordinate system O-X-Y-Z in terms of Euler's angles theta ($\theta$), psi ($\psi$), and phi ($\phi$) as follows:

$$\omega_x = -\dot{\psi}\sin\theta\cos\phi + \dot{\theta}\sin\phi \tag{11}$$

$$\omega_y = \dot{\theta}\cos\phi + \dot{\psi}\sin\theta\sin\phi \tag{12}$$

$$\omega_z = \dot{\psi}\cos\theta + \dot{\phi} \tag{13}$$

Solving for $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$ gives

$$\dot{\psi} = \frac{\omega_y\sin\phi - \omega_x\cos\phi}{\sin\theta} \tag{14}$$

$$\dot{\theta} = \omega_x\sin\phi + \omega_y\cos\phi \tag{15}$$

$$\dot{\phi} = \omega_z - \omega_y\sin\phi\frac{\cos\theta}{\sin\theta} + \omega_x\cos\phi\frac{\cos\theta}{\sin\theta} \tag{16}$$

Second, it is assumed that the external torques only produce small angle deviations from the nominal time zero horizontal spin axis direction with initial values of:

$$\theta_0 = \frac{\pi}{2} \quad (17)$$

$$\psi_0 = \phi_0 = 0 \quad (18)$$

Thus for small angular deviations, these equations are approximately the following:

$$\dot{\phi} \approx \omega_z \quad (19)$$

$$\dot{\psi} \approx \omega_y \sin\phi - \omega_x \cos\phi \quad (20)$$

$$\dot{\theta} = \omega_x \sin\phi + \omega_y \cos\phi \quad (21)$$

Since the gyroscope model assumes small angles, the tracker field-of-view is arbitrarily set at $\pm$ 2 degrees, and the gyro model should not be used outside this field-of-view.

Third, it is assumed the north pole of the spinning gyro permanent magnet is on the X axis. Then there exists a direction of precession coil winding such that the external torques on the gyro are the following:

$$M_y = K * \text{error signal} \quad (22)$$

$$M_x = 0 \quad (23)$$

Since the gyro is assumed to be maintained at a constant spin velocity, it is assumed that $M_z=0$.

Thus the gyro differential equations of motion are the following:

$$A\dot{\omega}_x + (C-A)\omega_y\omega_z = M_x \qquad (24)$$

$$A\dot{\omega}_y + (A-C)\omega_z\omega_x = M_y \qquad (25)$$

$$\omega_z = \text{constant} \qquad (26)$$

$$\dot{\phi} = \omega_z \qquad (27)$$

$$\dot{\psi} = \omega_y \sin\phi - \omega_x \cos\phi \qquad (28)$$

$$\dot{\theta} = \omega_x \sin\phi + \omega_y \cos\phi \qquad (29)$$

$$M_x = 0 \qquad (30)$$

$$M_y = K * \text{error signal} \qquad (31)$$

# IV. DIGITAL COMPUTER MODEL

The third procedure of the four phase simulation design process is to design the simulation model [1]. This section includes a discussion of the simulation model.

The digital model is written in FORTRAN. The model is not a real time model. The model source code listing is contained in Appendix A, and a brief user's manual is contained in Appendix B. A block diagram of the closed loop tracker model is shown in Figure 7.

The model runtime is not a primary consideration, but rather the model is as simple as possible so future modification (to simulate real tracker hardware) is a straightforward task. The runtime for a .1 second simulation is approximately seven minutes on a Unisys 25 megahertz PC-AT with an 80386 microprocessor and a coprocessor, and approximately 7.5 hours on a 10 megahertz PC-XT with a V20 microprocessor and no coprocessor.

The numerical solution of the differential equations is discussed in the next section.

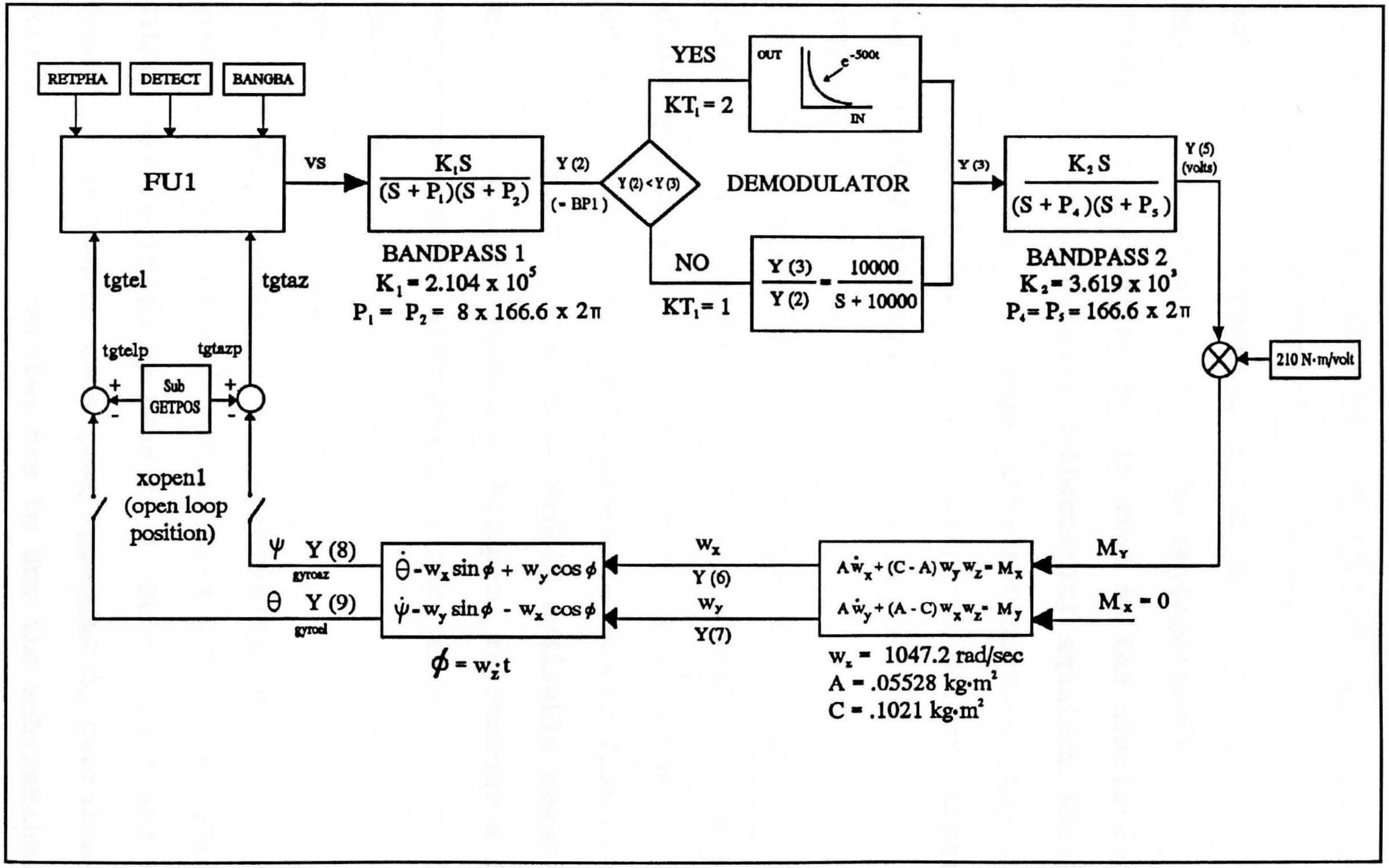



**Figure 7. Block Diagram of Closed Loop Tracker Model.**

## A. NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS

This section contains a discussion of the numerical solution of differential equations using the Runge-Kutta-Verner method. An implementation of the Runge-Kutta-Verner method is used in the simulation model.

A widely available differential equation solving subroutine uses the Runge-Kutta method with step size control due to Verner [14]. Verner's method compares the results of a fifth and sixth order Runge-Kutta to determine step size selection [14].

The International Mathematical and Statistical Library (IMSL) manual recommends the routine IVPRK for solving non stiff systems where moderate accuracy is needed [15]. Routine IVPRK uses the Runge-Kutta-Verner fifth and sixth order method [15]. Another widely available numerical analysis package, Numerical Recipes, recommends a fourth order Runge-Kutta with adaptive step size control for general applications not demanding high precision and where convenience is paramount [16].

An implementation of the Runge-Kutta-Verner method was used for this project. Subroutines DIFSOL, DISQQB, and DISQQA were written by Kenneth A. Becker [17] and are public domain. The source code (see Appendix A) contains many comment lines explaining how to use the subroutines.

In reality, the model described in section III is at best an approximation of the tracker system described in Section II. The use of the Runge-Kutta-Verner method is consistent with the level of modeling used for the electronics and gyroscope in this thesis.

## V. TRACKER MODEL RESPONSE CHARACTERIZATION

The fourth procedure of the four phase simulation design process is test and evaluation of the simulation [1]. Since this tracker is only a conceptual tracker, there does not exist any data upon which to base a test and evaluation of the simulation. This section includes a discussion of the results of several simulation runs to characterize the tracker model response (since the model cannot be verified).

For modeling real trackers, it is recommended that particular attention be paid to the test and evaluation phase of model development to insure that the tracker model adequately reflects the real hardware response for the scenarios of interest.

Section V.A. contains a discussion of the results of a simulation run to "validate" the simulation numerical integration method.

Section V.B. contains a discussion of the results of a simulation run where the open loop response of the tracker electronics model is characterized by comparing the steady state output of the tracker electronics model to the modeled amplitude modulated detector input.

Section V.C. contains a discussion of the results of simulation runs where the tracker model open loop response is characterized for various values of the reticle phasing angle and the loop gain.

Section V.D. contains a discussion of the results of simulation runs where the tracker model closed loop response is characterized for various values of the reticle phasing angle.

## A. CHARACTERIZATION OF SUBROUTINE DIFSOL

This section includes a discussion of the results of a simulation run to "validate" the simulation numerical integration method.

A desirable feature of a simulation is a self test. One approach to determining the accuracy of the numerical solution of differential equations is to compare the simulation results for a known input to the closed form solution for the known input.

The simulation model contains a self test feature for the step response of the following:

$$A\dot{\omega}_x + (C-A)\omega_y\omega_z = 0 \qquad (24)$$

$$A\dot{\omega}_y + (A-C)\omega_z\omega_x = M_y \qquad (25)$$

Since $w_z$ is constant (Equation 26), the closed form solution to a step input $5M_y$ is the following (for all initial conditions of zero):

$$\omega_x(t) = -\frac{M_y}{A \cdot b_2}[1-\cos(b_2 \cdot t)] \quad \frac{\text{radians}}{\text{second}} \qquad (32)$$

$$\omega_y(t) = \frac{M_y}{A \cdot b_2}[\sin(b_2 \cdot t)] \quad \frac{\text{radians}}{\text{second}} \qquad (33)$$

where $b_2 = ((C-A)/A) * w_z$ and t is time in seconds.

Using this self test feature, the simulation results and closed form results agree to 4 significant figures at the end of a .1 second simulation run.

In reality, $M_y$ is never a step function. The only purpose of the simulation run discussed in this section is to illustrate one approach to "validating" the numerical integration method (in this case subroutine DIFSOL and associated routines) by comparing the simulation results for a known input (in this case, a step function) to the closed form solution for a set of differential equations (Equations 24 and 25 in this case). The agreement to four (4) significant figures between the simulation results and the closed form solution indicates the numerical integration routines are performing satisfactorily.

## B. CHARACTERIZATION OF MODEL ELECTRONICS RESPONSE

This section includes a discussion of the results of a simulation run where the open loop response of the tracker electronics model is characterized by comparing the steady state output of the tracker electronics model to the modeled amplitude modulated detector input.

The open loop response of the tracker electronics model is characterized by comparing the steady state output of the tracker electronics model to an amplitude modulated detector input. The amplitude modulated detector input is modeled by the formula [18]:

$$y=a_0*[1+m*\sin(2*\pi*w*t)]*\sin(2*\pi*w_0*t) \qquad (34)$$

where
- $m$ is the degree of modulation
- $w$ is 166.7 Hz
- $w_0$ is 1333.6 (166.7*8) Hz
- $\pi$ is approximately 3.1415927
- $t$ is time in seconds
- $a_0$ is arbitrary gain constant

Figure 8 displays the envelope of the amplitude modulated detector input and the steady state output of the tracker electronics model for the case where m=1.0 and 1/2. The tracker electronics model has somewhat distorted the pure sinusoidal input and there is a fundamental frequency phase shift of about 39 degrees.

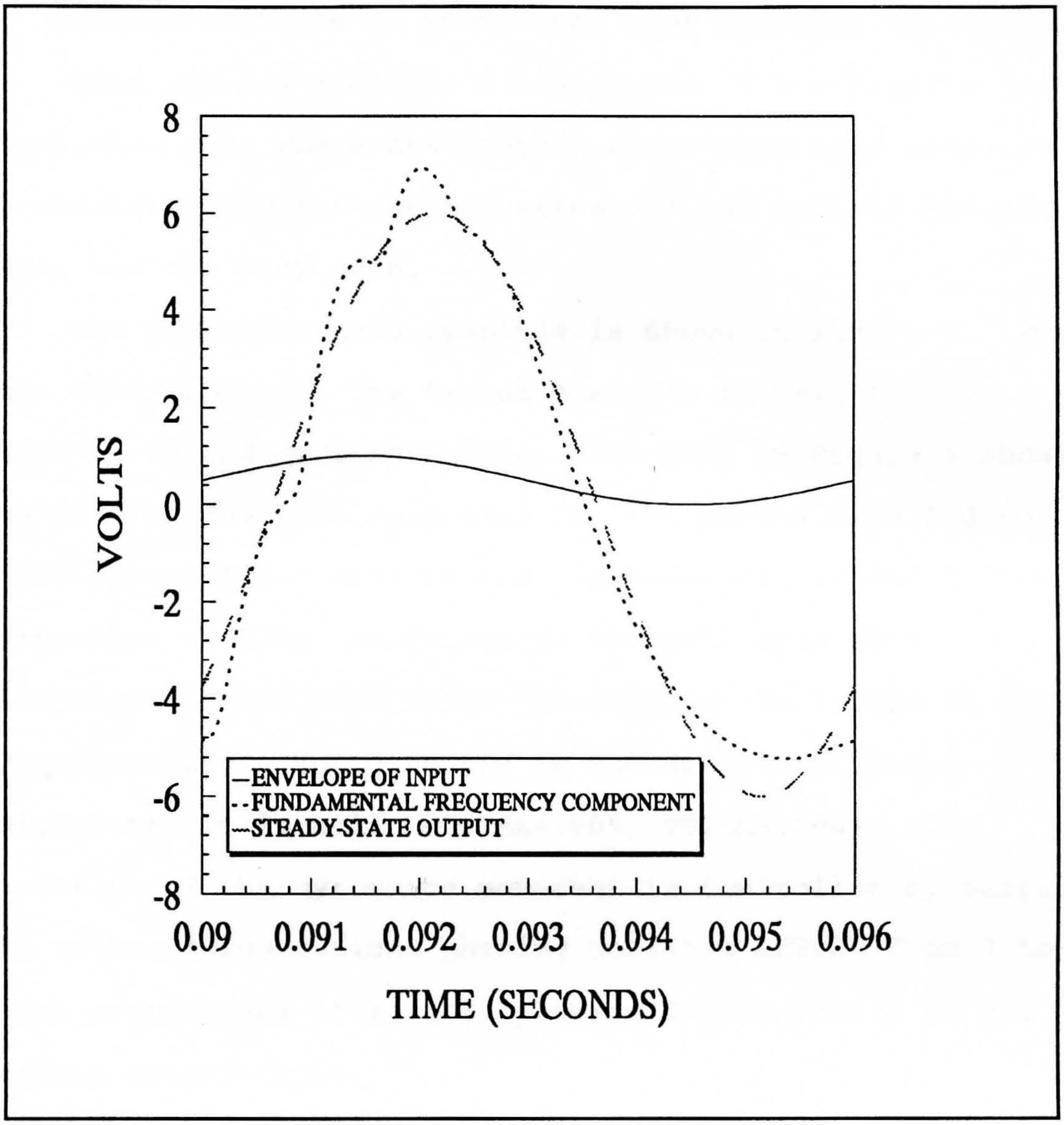



**Figure 8. Characterization of Model Electronics.**

## C. CHARACTERIZATION OF MODEL OPEN LOOP RESPONSE

This section includes a discussion of the results of simulation runs where the tracker model open loop response is characterized for various values of the reticle phasing angle and the loop gain.

The open loop gyro response is shown in Figure 9. For this open loop run, the target position is held fixed relative to the gyro spin axis. The plot in Figure 9 shows the path of the gyro spin axis for the period from 0.0 to .0999 second for a reticle phasing value of 0.0 and a loop gain value of 210. In Figure 9, the gyro spin axis direction at time zero is at the origin. In Figure 9, the gyro movement is not directly in the direction of the initial target location (TGTAZ=.005, TGTEL=.000). The direction of the gyroscope movement is controlled by varying the value of the reticle phasing variable RETPHA from 0 to 2 which corresponds to reticle phasing angles from 0 to 360 degrees respectively.

The rate of the gyroscope movement is controlled by varying the value of the loop gain variable XGAIN. The gyroscope open loop tracking rate is about 12 degrees per second for XGAIN equal to 210.

The electrical power required for steady state maximum precession rate was calculated assuming a lossless electromechanical energy converter. For a lossless electromechanical converter, the steady state electrical

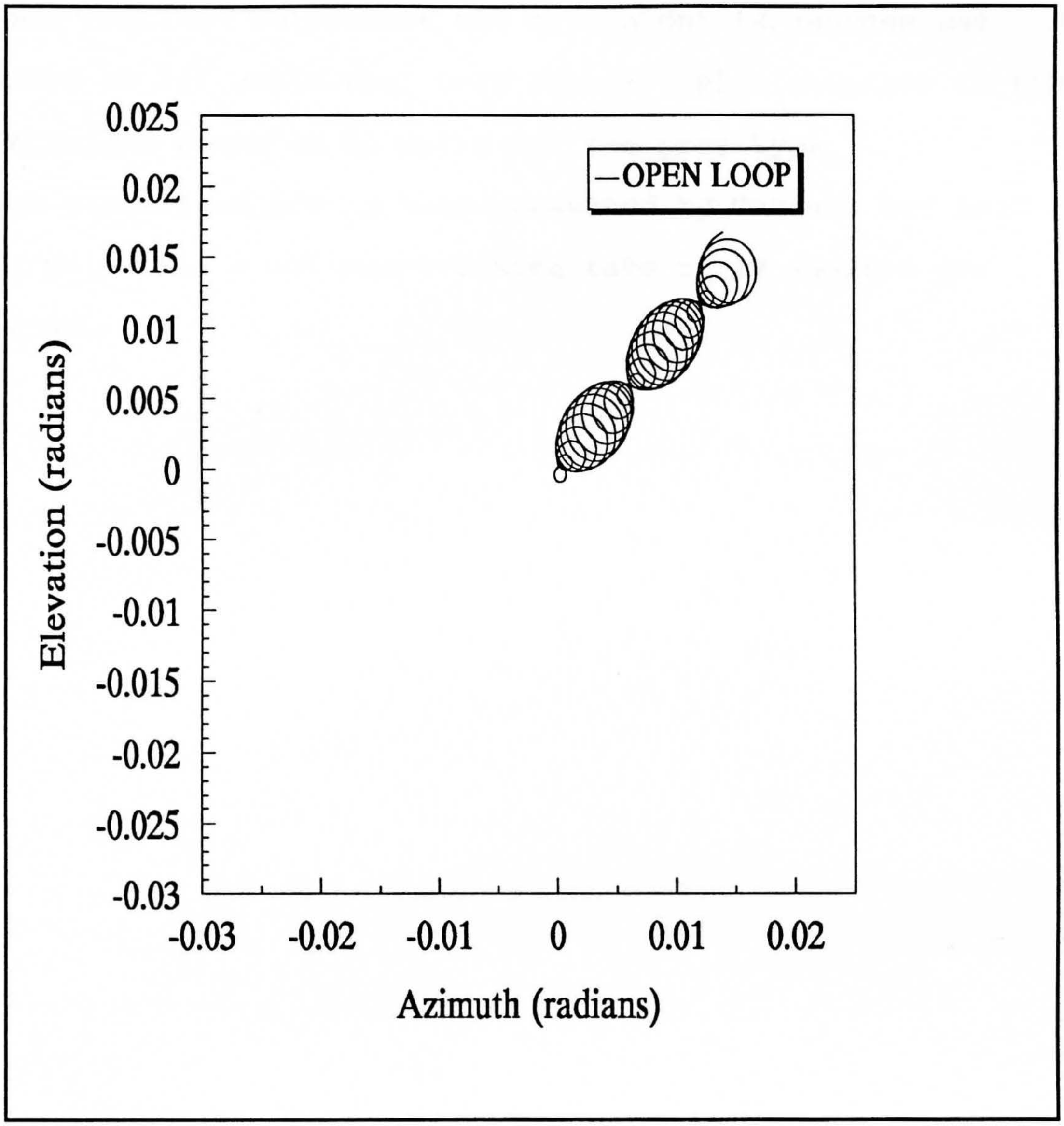



Figure 9. Open Loop Gyroscope Response.

power required to precess the gyro about 12 degrees per second is 116 watts root mean square, which compares to the electrical power of 80 watts for the prototype gyro-stabilized homing head presented by Garnell and East [9] which has a maximum tracking rate of 10 degrees per second.

CHARACTERIZATION OF MODEL CLOSED LOOP RESPONSE

This section includes a discussion of the results of mulation runs where the tracker model closed loop response characterized for various values of the reticle phasing gle.

The tracker model closed loop response shows that the ticle phasing is one of the critical parameters termining tracker closed loop performance. Appendix C ntains plots of the tracker model closed loop response for veral different values of reticle phasing. The results in pendix C show that the value chosen for the reticle asing variable RETPHA determines the tracker model closed op response. Based on the plots in Appendix C, a value ar 0.0 for the reticle phasing variable RETPHA produces a osed loop response such that the gyro spin axis is moved, is desirable, to a point near the target (with some teady state" tracking noise after the gyro is pointed near e stationary target).

# VI. CONCLUSIONS AND RECOMMENDATIONS

## A. CONCLUSIONS

The objective of this thesis is to construct a simple simulation model of a generic infrared tracker that runs on a personal computer. This objective is met.

This simple model serves two purposes:

1. As a simple introduction to infrared trackers, and
2. As a prototype for developing a simulation model of a real tracker.

## B. RECOMMENDATIONS

For modeling real trackers, particular attention must be paid to the validation phase of model development to insure that the tracker model adequately reflects the real hardware response for the scenarios of interest. For example, the reticle phasing value used in the tracker model must adequately reflect the real hardware response since reticle phasing is a critical factor determining tracker model closed loop performance.

APPENDICES

# APPENDIX A. DIGITAL COMPUTER PROGRAM LISTING

This appendix contains a listing of the tracker model source code. A brief user's manual is contained in Appendix B.

```
C     THIS PROGRAM WRITTEN BY JAMES M BROWN
C     MAY 1992
C      SUBROUTINES FROM OTHER AUTHORS HAVE AUTHOR NAME AT
C      BEGINNING OF SOURCE CODE
      PROGRAM TRACKR
      IMPLICIT DOUBLE PRECISION (A-Z)
      COMMON/YPR/ XGAIN,BIAS,BIASD,REGULA,
     C       ADIFEQ,BDIFEQ
      COMMON/FU/TGTAZ,TGTEL,RETPHA,BANGBA
      COMMON/DET/DETECT
      COMMON/XOPEN/XOPENL,TGSPRT
      COMMON/IOC/TIMPLO,H,XF,FILEOU,SCREEN,STOUT,ENDOUT
      OPEN(6,'SCROLL.OUT')
       H=.00001D0
C       H=.0002
       XF=.1D0
       XGAIN=210.0D0
       TGTAZ=.005D0
       TGTEL=.00D0
C       TGSPRT=0.873D0
       TGSPRT=0.0D0
       BIAS=.0D0
c       BIAS=5.0D0
       BIASD=0.D0
       REGULA=1.0D0
C       RETPHA=1.760000D0
       RETPHA=0.D0
       ADIFEQ=-885.9D0
       BDIFEQ=0.D0
       XOPENL=10.0D0
       BANGBA=10.D0
       DETECT=20.D0
C      OUTPUT CONTROL
       SCREEN=0D0
       STOUT=.000D0
       ENDOUT=.1D0
       TIMPLO=02D0
       FILEOU=0.085D0
C
       CALL BROCKH
C
      STOP
      END
C
```

```
      SUBROUTINE YPRIM(N,T,Y,YPRIME)
C     THIS SUBROUTINE CONTAINS THE STATE EQUATIONS
      INTEGER KT1
      DOUBLE PRECISION T,Y(N),YPRIME(N),VS
     C   ,BP1,T1,T3,T4,B1,B2,A1,A2
     C  ,XGAIN,BIAS,BIASD,REGULA,ADIFEQ,BDIFEQ,DETECT,PI
     C  ,ERROR,WS,EBIAS,XINPUT,XIPUT1,OUTTX,OUTTY
            COMMON VS,KT1,BP1,CF,A,OUTTX,OUTTY
      COMMON/YPR/ XGAIN,BIAS,BIASD,REGULA,
     C        ADIFEQ,BDIFEQ
      COMMON/DET/DETECT
C     BANDPASS1,OUTPUT IS BP1
      T1=.0030D0
C     GAIN IS LINEAR FUNCTION OF T1
      T3=.0001194D0
      T4=.0001194D0
      B1=T1/(T3*T4)
      B2=0.D0
      A1=(T3+T4)/(T3*T4)
      A2=1D0/(T3*T4)
      YPRIME(1)=Y(2)
      YPRIME(2)=-A2*Y(1)-A1*Y(2)+B1*VS
      BP1=(B2/B1)*Y(1)+Y(2)
C     DEMOD, OUTPUT IS Y(3)
      CF=10000.D0
      A=1.D0/.002D0
C     GAIN IS LINEAR COEF OF CONSTANT COEFF OF BP1
      IF(KT1.EQ.1)    YPRIME(3)=-CF*Y(3)+(CF/1.D0)*BP1
      IF(KT1.EQ.2)    YPRIME(3)=-A*Y(3)
C     BANDPASS2, OUTPUT IS Y(5)
      T1=.0033D0
      T3=.0009549D0
      T4=.0009549D0
      B1=T1/(T3*T4)
      B2=0.D0
      A1=(T3+T4)/(T3*T4)
      A2=1D0/(T3*T4)
      YPRIME(4)=Y(5)
      YPRIME(5)=-A2*Y(4)-A1*Y(5)+B1*Y(3)
C     GYRO, OUTPUT IS Y(8) AND Y(9)
      PI=ATAN(1.0D0)*4.0D0
      ERROR=Y(5)
      WS=2.0D0*PI*166.666666666666D0
      EBIAS=BIAS*SIN(WS*T+BIASD*PI)
      ERROR=REGULA*ERROR+EBIAS
C      ERROR=BIAS
938   XINPUT=0.D0
      XIPUT1=XGAIN*ERROR
C     NEXT STATEMENT USED FOR CLOSED FORM SOLUTION CHECK
C      XIPUT1=XGAIN
839   YPRIME(6)=XINPUT-BDIFEQ*Y(6)+ADIFEQ*Y(7)
      YPRIME(7)=XIPUT1-BDIFEQ*Y(7)-ADIFEQ*Y(6)
      YPRIME(8)=-Y(6)*COS(WS*T)+Y(7)*SIN(WS*T)
      YPRIME(9)= Y(6)*SIN(WS*T)+Y(7)*COS(WS*T)
      OUTTX=XINPUT*.05528D0
      OUTTY=XIPUT1*.05528D0
C      WRITE(*,*)T,Y(8),Y(9)
200   RETURN
      END
C
```

```
      SUBROUTINE GETPOS(IFIRST,Y1,Y1EL,TGTAZ,TGTEL
     X                               ,TIME,TGSPRT)
C     THIS SUBROUTINE CALCULATES TARGET POSITION RELATIVE TO
C     THE GYRO SPIN AXIS.
      IMPLICIT DOUBLE PRECISION (A-Y)
      IF (IFIRST.GT.1) GOTO1
      TGTAZP=TGTAZ
      TGTELP=TGTEL
1     CONTINUE
      TGTAZ=TGTAZP+TGSPRT*TIME-Y1
      TGTEL=TGTELP-Y1EL
      RETURN
      END
C
```

```
      FUNCTION FU1(TIME)
C     THIS SUBROUTINE CALCULATES RETICLE OUTPUT
      DOUBLE PRECISION TIME,DETOUT,FU1
     C    ,TGTAZ,TGTEL,RTPHAA,BANGBA,DETECT
      COMMON/FU/TGTAZ,TGTEL,RTPHAA,BANGBA
      COMMON/DET/DETECT
      PI=ATAN(1.0D0)*4.0D0
      WS=2.0D0*PI*166.6666666666D0
      IF (DETECT.GE.1.0)GOTO938
C     CALCULATE AM MODULATION FROM MATHEMATICAL FORMULA
      FU1=(.50D0+.5D0*SIN(WS*TIME))*SIN(8.0D0*WS*TIME)
      RETURN
 938  CONTINUE
      IF (DETECT.GE.50.0)GOTO1938
C     CALCULATE RETICLE CHOP
      ERROR=SQRT(TGTAZ*TGTAZ+TGTEL*TGTEL)
      FEPS=1*ERROR
      IF(FEPS.GT..035)DETOUT=0.
      IF(FEPS.GT..035)GOTO 7650
      IF(FEPS.EQ..0)DETOUT=0.
      IF(FEPS.EQ..0)GOTO 7650
      IF(FEPS.GT..010)FEPS=.01
      IF(BANGBA.LT.1.)GOTO974
      FEPS=.01
 974  CONTINUE
      RETPHA=PI*RTPHAA
C     CALCULATE RETICLE POSITION
        RETPOS=((TIME*WS)+RETPHA)/(2.*PI)
        RETPOS=RETPOS-INT(RETPOS)
        TGTPOS=ATAN2(-TGTEL,TGTAZ)/(2.*PI)
C        TGTPOS=ATAN2(TGTEL,TGTAZ)/(2.*PI)
        IF (TGTPOS.LT. 0) TGTPOS=TGTPOS+1.
        TGTRET=TGTPOS-RETPOS
        IF (TGTRET .LT.0.) TGTRET=TGTRET+1.
        IF (TGTRET.LT..0625 )GOTO 600
        IF (TGTRET.LT..125 )GOTO 650
        IF (TGTRET.LT..1875 )GOTO 600
        IF (TGTRET.LT..25 )GOTO 650
        IF (TGTRET.LT..3125 )GOTO 600
        IF (TGTRET.LT..375 )GOTO 650
        IF (TGTRET.LT..4375 )GOTO 600
        IF (TGTRET.LT..5    )GOTO 650
        IF (TGTRET.LT.1.) GOTO 620
600     DETOUT=(.01+FEPS)/.02
610     GOTO 660
620     DETOUT=.01/.02
630     GOTO 660
650     DETOUT=(.01-FEPS)/.02
660     DETOUT=DETOUT*1.0
7650    FU1=DETOUT
C        IF (TIME.GE..02D0)FU1=.5
C      WRITE(*,572)TIME,TGTAZ,TGTEL,RETPOS,TGTPOS,TGTRET,DETOUT
 572    FORMAT(1X,F8.5,2F8.4,6(F9.4,1X))
        RETURN
 1938   CONTINUE
C     CALCULATE RETICLE CHOP
      ERROR=SQRT(TGTAZ*TGTAZ+TGTEL*TGTEL)
      FEPS=1*ERROR
```

```
      IF(FEPS.GT..035)DETOUT=0.
      IF(FEPS.GT..035)GOTO 8650
      IF(FEPS.EQ..0)DETOUT=0.
      IF(FEPS.EQ..0)GOTO 8650
      IF(FEPS.GT..010)FEPS=.01
      IF(BANGBA.LT.1.)GOTO874
      FEPS=.01
 874  CONTINUE
      RETPHA=PI*RTPHAA
C     CALCULATE RETICLE POSITION
        RETPOS=((TIME*WS)+RETPHA)/(2.*PI)
        RETPOS=RETPOS-INT(RETPOS)
        TGTPOS=ATAN2(-TGTEL,TGTAZ)/(2.*PI)
C        TGTPOS=ATAN2(TGTEL,TGTAZ)/(2.*PI)
        IF (TGTPOS.LT. 0) TGTPOS=TGTPOS+1.
        TGTRET=TGTPOS-RETPOS
        IF (TGTRET .LT.0.) TGTRET=TGTRET+1.
        DETOUT=.5+.5*SIN(PI*16D0*TGTRET)
        IF (TGTRET.LT..5   )GOTO 8650
        DETOUT=.01/.02
8650    FU1=DETOUT
C        IF (TIME.GE..02D0)FU1=.5
C     WRITE(*,572)TIME,TGTAZ,TGTEL,RETPOS,TGTPOS,TGTRET,DETOUT
        RETURN
        END
C
```

```
      SUBROUTINE BROCKH
      EXTERNAL YPRIM
      EXTERNAL FU1
      DOUBLE PRECISION T,Y(9),TEND,TOL,HSTART,HMIN,HMAX,WKAREA(9,12)
     #,VS,BP1,FU1
     # ,XOPENL,TGTAZ,TGTEL,RETPHA,BANGBA,IFIRST,OUTX,OUTY
     # ,TIMPLO,TGSPRT,H,XF,FILEOU,SCREEN,STOUT,ENDOUT
      REAL TIME(1501),OUT5(1501),OUT1(1501),OUT2(1501),OUT3(1501),
     #  OUT8(1501),OUT9(1501),OUTTY(1501)
      INTEGER NDIF,NW,IER,IOUT,LINTYP(4),LINFRQ,IGRAPH,N
     #    ,M1,KT1
      COMMON VS,KT1,BP1,CF,A,OUTX,OUTY
      COMMON/XOPEN/XOPENL,TGSPRT
      COMMON/FU/TGTAZ,TGTEL,RETPHA,BANGBA
      COMMON/IOC/TIMPLO,H,XF,FILEOU,SCREEN,STOUT,ENDOUT
      CHARACTER*20 XLABLE,YLABLE,TEMP,ZLABLE,ULABLE,VLABLE,WLABLE,TLABLE
      CHARACTER*20 TYLABL
C SET THE DEFAULT VALUES FOR THE DIFSOL AND PLOTS SUBROUTINES
      NW=9
      IOUT=1501
      NDIF=9
C       TOL=1.D-6
      TOL=1.D-4
      HSTART=0.04D0/20.D0
C       HMIN=TOL*0.04D0*.0001D0
      HMIN=1.D-6*0.04D0*.0001D0
C       HMAX=0.7D-6
      HMAX=H
C SET THE VALUES FOR THE CONSTANTS
C SET THE INITIAL CONDITIONS
      T=0.0D0
      Y(1)=0.0D0
      Y(2)=0.0D0
      Y(3)=0.0D0
      Y(4)=0.0D0
      Y(5)=0.0D0
      Y(6)=0.0D0
      Y(7)=0.0D0
      Y(8)=0.0D0
      Y(9)=0.0D0
      KT1=1
      II=0
C     DEFINE THE OUTPUT ARRAYS
      TIME(1)=REAL(T)
      OUT1(1)=0.0D0
      OUT2(1)=0.0D0
      OUT3(1)=0.0D0
      OUT5(1)=0.0D0
      OUT8(1)=0.0D0
      OUT9(1)=0.0D0
      OUTTY(1)=0.0D0
C
C THIS BEGINS THE INTEGRATION LOOP
       IFIRST=1D0
 123  CONTINUE
       TEND=T + H
      IF (XOPENL.GE.1.)GOTO 978
       CALL GETPOS(IFIRST,Y(8),Y(9),TGTAZ,TGTEL,T,TGSPRT)
```

```
 978   CONTINUE
      VS=FU1(T)
       KT1=1
       IF(BP1.LT.Y(3))KT1=2
       IF (SCREEN.EQ.0.D0)GOTO1514
       IF (T.LT.STOUT) GOTO1514
       IF (T.GT.ENDOUT) GOTO1514
C         WRITE(6,543) T,VS,BP1,Y(3),Y(5),Y(8),Y(9)
        WRITE(*,543) T,VS,BP1,Y(3),Y(5),Y(8),Y(9)
 543    FORMAT(1X,7F10.6)
 1514 CONTINUE
C THIS CALLS THE DIFF. EQUATION SOLVER, WHICH CALLS SUBROUTINE YPRIM
       CALL DIFSOL(NDIF,YPRIM,T,Y,TEND,TOL,HSTART,HMIN,HMAX,
     #   WKAREA,NW,IER)
       IF(IER.NE.0)   WRITE(*,*) IER,T
C
C     DO RUNTIME PLOT TO SCREEN FOR LAHEY FTN IF DESIRED
      IF (TIMPLO.NE.2.)GOTO 7534
      ZT1=REAL(Y(8))+.04
      ZT2=REAL(Y(9))+.04
      ZFAC=100.0
      CALL TIM(ZT1,ZT2,ZFAC)
      write(6,2341)t,y(8),y(9)
 2341 format(1x,f7.5,2f6.4)
 7534  CONTINUE
C
C     PUTTING VALUES INTO OUTPUT ARRAYS FOR POST RUN PLOTS
C     MICROSOFT FORTRAN SPECIFIC
      IF (TIMPLO.NE.3.)GOTO 4883
      IF(FILEOU.GT.T)GOTO4883
       TIME(II+1)=REAL(T)
       OUT1(II+1)=REAL(VS)
       OUT2(II+1)=REAL(Y(2))
       OUT3(II+1)=REAL(Y(3))
       OUT5(II+1)=REAL(Y(5))
       OUT8(II+1)=REAL(Y(8))
       OUT9(II+1)=REAL(Y(9))
       OUTTY(II+1)=REAL(OUTY)
      II=II+1
      IF(II.GE.1501)FILEOU=XF+10D0
 4883 CONTINUE
       IFIRST=10D0
      IF(T.GE.XF)GOTO6738
C     NEXT 5 STATEMENTS USED FOR CLOSED FORM SOLUTION CHECK
C      WS=-885.9
C      CFS6=OUTY*(1-COS(WS*T))/(WS*.05528)
C      CFS7=OUTY*SIN(WS*T)/(WS*.05528)
C      WRITE(*,9581)T,Y(6),Y(7),CFS6,CFS7
C 9581 FORMAT (5F10.8)
      GOTO123
 6738 CONTINUE
C THE INTEGRATION LOOP HAS ENDED

C    POST RUN PLOTS (MICROSOFT FORTRAN SPECIFIC)
      IF (TIMPLO.NE.3.)GOTO 4738
C SET THE DEFAULT VALUES FOR PLOTTING
      N=1501
      M1=1
```

```
      LINTYP(1)=0
      LINTYP(2)=0
      LINTYP(3)=1
      LINTYP(4)=0
      LINFRQ=40
      IGRAPH=1
      XLABLE='TIME (SECONDS)'
      YLABLE='DETECTOR (VOLTS)'
      ZLABLE='BANDPASS 2 (VOLTS)'
      ULABLE='BANDPASS 1 (VOLTS)'
      VLABLE='DEMODULATOR (VOLTS)'
      WLABLE='GYRO AZ (RADIANS)'
      TLABLE='GYRO EL (RADIANS)'
      TYLABL='TORQUE Y (N*M)'
C     PLOT THE FIRST GRAPH
      CALL PLOTS(TIME,OUT1,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    XLABLE,YLABLE,IER)
C     KEEP THE PLOT ON THE SCREEN UNTIL A KEY IS PRESSED
      READ(*,'(A)')TEMP
      CALL PLOTS(TIME,OUT2,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    XLABLE,ULABLE,IER)
      READ(*,'(A)')TEMP
      CALL PLOTS(TIME,OUT3,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    XLABLE,VLABLE,IER)
      READ(*,'(A)')TEMP
      CALL PLOTS(TIME,OUT5,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    XLABLE,ZLABLE,IER)
      READ(*,'(A)')TEMP
      CALL PLOTS(OUT8,OUT9,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    WLABLE,TLABLE,IER)
      READ(*,'(A)')TEMP
      CALL PLOTS(TIME,OUTTY,IOUT,N,M1,LINTYP,LINFRQ,IGRAPH,
     #    XLABLE,TYLABL,IER)
      READ(*,'(A)')TEMP
C     TAKE THE SCREEN OUT OF TEXT MODE
      CALL SCRNQQ(3)
 4738       STOP
      END
```

```
C     SUBROUTINE DIFSOL WRITTEN BY KENNETH A BECKER
      SUBROUTINE DIFSOL(N, FCN, X, Y, XEND, TOL, HSTART, HMIN,
     *                  HMAX, W, NW, IER)
C
C FINDS THE SOLUTION OF N 1ST ORDER COUPLED DIFFERENTIAL EQUATIONS.
C ARGUEMENTS:
C
C     N     - NUMBER OF EQUATIONS. (INPUT)
C     FCN   - NAME OF SUBROUTINE FOR EVALUATING FUNCTIONS. (INPUT)
C               THE SUBROUTINE ITSELF MUST ALSO BE PROVIDED BY THE USER
C               AND IT SHOULD BE OF THE FOLLOWING FORM
C                 SUBROUTINE FCN(N,X,Y,YPRIME)
C                 DOUBLE PRECISION X,Y(N),YPRIME(N)
C                   .
C                   .
C                   .
C               FCN SHOULD EVALUATE YPRIME(1),...,YPRIME(N)
C                 GIVEN N, X, AND Y(1),...,Y(N). YPRIME(I) IS
C                 THE FIRST DERIVATIVE OF Y(I) WITH RESPECT TO X.
C               FCN MUST APPEAR IN AN EXTERNAL STATEMENT IN THE
C                 CALLING PROGRAM AND N,X,Y(1),...,Y(N) MUST NOT BE
C                 ALTERED BY FCN.
C     X     - INDEPENDENT VARIABLE. (INPUT AND OUTPUT)
C               ON INPUT, X SUPPLIES THE INITIAL VALUE.
C               ON OUTPUT, X IS REPLACED WITH XEND UNLESS ERROR
C               CONDITIONS ARISE. SEE THE DESCRIPTION OF THE PARAMETER
C               IER.  X IS DOUBLE PRECISION.
C     Y     - DEPENDENT VARIABLES, VECTOR OF LENGTH N. (INPUT AND
C               OUTPUT).
C               ON INPUT, Y(1),...Y(N) SUPPLY INITIAL VALUES.
C               ON OUTPUT, Y(1),...,Y(N) ARE REPLACED WITH AN
C               APPROXIMATE SOLUTION AT XEND UNLESS ERROR CONDITIONS
C               ARISE.  SEE THE DESCRIPTION OF PARAMETER IER.
C               Y IS A DOUBLE PRECISION ARRAY.
C     XEND  - VALUE OF X AT WHICH SOLUTION IS DESIRED. (INPUT)
C               XEND MAY BE LESS THAN THE INITIAL VALUE OF X, IN WHICH
C               HSTART MUST BE LESS THAN ZERO.
C               XEND IS DOUBLE PRECISION.
C     TOL   - TOLERANCE FOR ERROR CONTROL. (INPUT)
C               THE SUBROUTINE ATTEMPTS TO CONTROL A NORM OF THE LOCAL
C                 ERROR IN SUCH A WAY THAT THE GLOBAL ERROR IS PRO-
C                 PORTIONAL TO TOL.  MAKING TOL SMALLER IMPROVES
C                 ACCURACY, BUT INCREASES PROCESSING TIME.  THE GLOBAL
C                 ERROR IS DEFINED AS
C                   MAX(DABS(E(1)),...,DABS(E(N)))
C                 WHERE E(K) = (Y(K)-YT(K))/MAX(1,ABS(Y(K))), WHERE
C                 YT(K) IS THE TRUE SOLUTION AND Y(K) IS THE COMPUTED
C                 SOLUTION AT XEND, FOR K = 1,2,...,N.
C                 TOL IS DOUBLE PRECISION.
C     HSTART- INITIAL STEP SIZE USED TO INCREMENT X IN THE ALGORITHM.
C               (INPUT AND OUTPUT).
C               ON OUTPUT, HSTART IS REPLACED WITH THE STEP SIZE USED
C                 AT XEND.
C               WHEN MAKING MULTIPLE CALLS TO DIFSOL, HSTART SHOULD BE
C                 SET ONLY ON THE INITIAL CALL.
C               HSTART IS DOUBLE PRECISION.
C     HMIN  - MINIMUM STEP SIZE USED TO INCREMENT X IN THE ALGORITHM.
C               (INPUT)
```

```
C              THE STEP SIZE H IS VARIED DYNAMICALLY IN THE ALGORITHM
C                TO ACHIEVE SOME MINIMUM GLOBAL ERROR DEFINED BY TOL.
C                AN ERROR IS DEFINED WHEN H BECOMES LESS THAN HMIN.
C                SEE THE SECTION ON IER.
C                HMIN IS DOUBLE PRECISION.
C     HMAX   - MAXIMUM STEP SIZE USED TO INCREMENT X IN THE ALGORITHM.
C              (INPUT).  THE STEP SIZE H IS CONSTRAINED TO BE LESS
C              THAN HMAX.
C     W      - WORKSPACE MATRIX
C              THE FIRST DIMENSION OF W MUST BE NW AND THE SECOND
C              GREATER THAN OR EQUAL TO 12.  W IS A DOUBLE PRECISION
C              ARRAY.
C     NW     - ROW DIMENSION OF THE MATRIX W EXACTLY AS STATED IN THE
C              CALLING PROGRAM. (INPUT).
C            NW MUST BE GREATER THAN OR EQUAL TO N.
C     IER    - ERROR PARAMETER.   (OUTPUT)
C            TERMINAL ERROR.
C            IER = 0         NO ERROR.
C            IER = 129       NW IS LESS THAN N, TOL IS LESS THAN OR
C                            EQUAL TO ZERO, OR X GT XEND.
C            IER = 130       H HAS BECOME LESS THAN HMIN.  THIS USUALLY
C                            OCCURS WHEN ONE OF THE COMPONENTS OF Y IS
C                            CHANGING VERY RAPIDLY.  FOR EXAMPLE, THE
C                            ROUTINE MAY BE ATTEMPTING TO EVALUATE A
C                            SOLUTION OF THE FORM Y = 1/(1-X) AS X
C                            APPROACHES ONE.  THEREFORE, IF IER=130
C                            OCCURS, IT IS USUALLY A GOOD IDEA FOR THE
C                            PROGRAMMER TO EYEBALL THE INPUT
C                            PARAMETERS, FCN, AND POSSIBLE THE PROBLEM
C                            IN GENERAL.
C                             NOTE THAT IF TOL IS SET TOO SMALL, H WILL
C                            BE DRIVEN SMALLER THAN HMIN, CAUSING
C                            IER = 130.  IF HMIN IS TOO LARGE, NORMAL
C                            VALUES OF H MAY CALUSE IER=130.  SETTING
C                            HMIN TO AN EXCEEDINGLY SMALL VALUE MAY
C                            REMOVE EXTRANEOUS PROBLEMS DURING NORMAL
C                            OPERATION.  HOWEVER, IF A Y=1/(1-X) PROBLEM
C                            THEN OCCURS, IT MAY TAKE AN EXCEEDINGLY
C                            LONG TIME BEFORE THE ALGORITHEM NOTICES.
C                            THEREFORE, HMIN SHOULD BE SET TO A SMALL
C                            NUMBER - BUT NOT TOO SMALL!
C                       PROCESSING
C                         WHEN IER=130 OCCURS, X IS SET TO THE VALUE AT
C                            WHICH THE ALGORITHM FAILED AND THE Y ARRAY
C                            IS SET TO THE APPROXMIATE SOLUTION AT THAT
C                            VALUE OF X.
C
C DIMENSIONING
C
      EXTERNAL FCN
      DOUBLE PRECISION Y(*),W(NW,*),HOLD,HSTART,X,XEND,HMIN,HMAX,
     *          TOL,HNEW,GAMMA,GAMLA,XA,B21,B31,B32,B41,B42,B43,
     *               B51,B52,B53,B54,B61,B62,B63,B64,B65,B71,B72,B73,
     *               B74,B75,B81,B82,B83,B84,B85,B87,A1,A3,A4,A5,A6,R1,
     *               R3,R4,R5,R6,R7,R8
      INTEGER IER,I
      COMMON /DIFQQC/B21,B31,B32,B41,B42,B43,B51,B52,B53,B54,B61,B62,
     *               B63,B64,B65,B71,B72,B73,B74,B75,B81,B82,B83,B84,
```

```
     *                 B85,B87,A1,A3,A4,A5,A6,R1,R3,R4,R5,R6,R7,R8
C
C INITIALIZE CONSTANTS FOR RKF
C
      IER = 0
      IF (((N.GT.NW).OR.(TOL.LE.0.D0)).OR.(X.GT.XEND)) THEN
        IER = 129
        RETURN
      ENDIF
      B21 = 1.D0/18.D0
      B31 = -1.D0/12.D0
      B32 = 0.25D0
      B41 = -2.D0/81.D0
      B42 = 4.D0/27.D0
      B43 = 8.D0/81.D0
      B51 = 40.D0/33.D0
      B52 = -4.D0/11.D0
      B53 = -56.D0/11.D0
      B54 = 54.D0/11.D0
      B61 = -369.D0/73.D0
      B62 = 72.D0/73.D0
      B63 = 5380.D0/219.D0
      B64 = -12285.D0/584.D0
      B65 = 2695.D0/1752.D0
      B71 = -8716.D0/891.D0
      B72 = 656.D0/297.D0
      B73 = 39520.D0/891.D0
      B74 = -416.D0/11.D0
      B75 = 52.D0/27.D0
      B81 = 3015.D0/256.D0
      B82 = -9.D0/4.D0
      B83 = -4219.D0/78.D0
      B84 = 5985.D0/128.D0
      B85 = -539.D0/384.D0
      B87 = 693.D0/3328.D0
      A1 = .0375D0
      A3 = 0.16D0
      A4 = 486.D0/2240.D0
      A5 = 154.D0/320.D0
      A6 = 73.D0/700.D0
      R1 = 33.D0/640.D0
      R3 = -132.D0/325.D0
      R4 = 891.D0/2240.D0
      R5 = -.103125D0
      R6 = -73.D0/700.D0
      R7 = 891.D0/8320.D0
      R8 = 2.D0/35.D0
C
C INITIALIZATION PHASE. START SOLUTION PROCESS AT (A,Y(A)) WITH
C STEP SIZE HSTART.
C
      HOLD = HSTART
      XA = X
      DO 10 I = 1,N
        W(I,1) = Y(I)
   10 CONTINUE
C
C MAKE STEP OF LENGTH HOLD FROM (X,YOLD) FINDING YNEW, EST, AND
```

```
C GAMMA.  CALCULATE HNEW AND TEST TO SEE IF YNEW IS ACCEPTABLE.
C
    1 CALL DISQQA(FCN,N,XA,NW,W,HOLD)
      GAMMA = 1.0D38
      DO 20 I = 1,N
        CALL DISQQB(X,XEND,W(I,1),W(I,3),TOL,GAMLA)
        IF (GAMLA.LT.GAMMA) GAMMA = GAMLA
   20 CONTINUE
      HNEW = 0.8D0*GAMMA*HOLD
      IF (GAMMA.GE.1.D0) GO TO 2
C
C YNEW IS NOT ACCEPTABLE. RESTART FROM (X,YOLD) IF HNEW IS NOT
C LESS THAN HMIN
C
      IF (HNEW.LT.HOLD/10.D0) HNEW = HOLD/10.D0
      IF (HNEW.LT.HMIN) GO TO 3
      HOLD = HNEW
      GO TO 1
C
C YNEW IS ACCEPTABLE. TAKE NEXT STEP FROM (X+HOLD,YNEW) IF HNEW
C IS NOT LESS THAN HMIN.
C
    2 IF (HNEW.LT.HMIN) GO TO 3
      IF (HNEW.GT.5.D0*HOLD) HNEW = 5.D0*HOLD
      IF (HNEW.GT.HMAX) HNEW=HMAX
      IF (XA+HOLD.GE.XEND) GO TO 4
      XA = XA + HOLD
      HOLD = HNEW
      DO 30 I = 1,N
        W(I,1)= W(I,2)
   30 CONTINUE
      GO TO 1
C
C ALGORITHM FAILED.  LAST POINT REACHED WAS (X,YOLD)
C
    3 IER = 130
      X = XA
      HSTART = HNEW
      DO 40 I = 1,N
        Y(I) = W(I,1)
   40 CONTINUE
      RETURN
C
C ALGORITHM SUCCEEDED.  INITIALIZE HSTART FOR FURTHER INTEGRATIONS
C AND EXECUTE STEP THAT REACHES EXACTLY TO X=XEND.
C
    4 IER = 0
      HSTART = HNEW
      HOLD = XEND-XA
      CALL DISQQA(FCN,N,XA,NW,W,HOLD)
      DO 50 I = 1,N
        Y(I) = W(I,2)
   50 CONTINUE
      X = XEND
      RETURN
      END
C
```

```
C     SUBROUTINE DISQQB WRITTEN BY KENNETH A BECKER
      SUBROUTINE DISQQB(X,XEND,YOLD,EST,TOL,GAMMA)
      DOUBLE PRECISION X,XEND,YOLD,EST,TOL,GAMMA,ABSEST
      ABSEST = DABS(EST)
      IF (ABSEST.EQ.0.D0) GO TO 1
      IF (YOLD.EQ.0.D0) GO TO 2
      GAMMA = (TOL*DABS(YOLD)/(ABSEST*(XEND-X)))**.25
      RETURN
    1 GAMMA = 6.25D0
      RETURN
    2 GAMMA = (TOL*TOL/(ABSEST*(XEND-X)))**.25
      RETURN
      END
C
```

```
C     SUBROUTINE DISQQA WRITTEN BY KENNETH A BECKER
      SUBROUTINE DISQQA(FCN,N,X,NW,W,H)
      DOUBLE PRECISION X,W(NW,*),H,XT,
     *               B21,B31,B32,B41,B42,B43,B51,B52,B53,B54,B61,B62,
     *               B63,B64,B65,B71,B72,B73,B74,B75,B81,B82,B83,B84,
     *               B85,B87,A1,A3,A4,A5,A6,R1,R3,R4,R5,R6,R7,R8
      COMMON /DIFQQC/B21,B31,B32,B41,B42,B43,B51,B52,B53,B54,B61,B62,
     *               B63,B64,B65,B71,B72,B73,B74,B75,B81,B82,B83,B84,
     *               B85,B87,A1,A3,A4,A5,A6,R1,R3,R4,R5,R6,R7,R8
C
C COMMENCE
C
      DO 10 I = 1,N
        W(I,4) = W(I,1)
   10 CONTINUE
      XT = X
      CALL FCN(N,XT,W(1,4),W(1,5))
      DO 20 I = 1,N
        W(I,4) = W(I,1) + H*B21*W(I,5)
   20 CONTINUE
      XT = X + H/18.D0
      CALL FCN(N,XT,W(1,4),W(1,6))
      DO 30 I = 1,N
        W(I,4) = W(I,1) + H*(B31*W(I,5)+B32*W(I,6))
   30 CONTINUE
      XT = X + H/6.D0
      CALL FCN(N,XT,W(1,4),W(1,7))
      DO 40 I = 1,N
        W(I,4) = W(I,1)+H*(B41*W(I,5)+B42*W(I,6)+B43*W(I,7))
   40 CONTINUE
      XT = X + 2.D0*H/9.D0
      CALL FCN(N,XT,W(1,4),W(1,8))
      DO 50 I = 1,N
        W(I,4) = W(I,1)+H*(B51*W(I,5)+B52*W(I,6)+B53*W(I,7)+B54*W(I,8))
   50 CONTINUE
      XT = X + 2.D0*H/3.D0
      CALL FCN(N,XT,W(1,4),W(1,9))
      DO 60 I = 1,N
        W(I,4) = W(I,1)+H*(B61*W(I,5)+B62*W(I,6)+B63*W(I,7)+B64*W(I,8)+
     *           B65*W(I,9))
   60 CONTINUE
      XT = X + H
      CALL FCN(N,XT,W(1,4),W(1,10))
      DO 70 I = 1,N
        W(I,4) = W(I,1)+H*(B71*W(I,5)+B72*W(I,6)+B73*W(I,7)+B74*W(I,8)+
     *           B75*W(I,9))
   70 CONTINUE
      XT = X + 8.D0*H/9.D0
      CALL FCN(N,XT,W(1,4),W(1,11))
      DO 80 I = 1,N
        W(I,4) = W(I,1)+H*(B81*W(I,5)+B82*W(I,6)+B83*W(I,7)+B84*W(I,8)+
     *           B85*W(I,9)+B87*W(I,11))
   80 CONTINUE
      XT = X + H
      CALL FCN(N,XT,W(1,4),W(1,12))
      DO 90 I = 1,N
        W(I,2) = W(I,1)+H*(A1*W(I,5)+A3*W(I,7)+A4*W(I,8)+A5*W(I,9)+
     *           A6*W(I,10))
```

```
      W(I,3) = R1*W(I,5)+R3*W(I,7)+R4*W(I,8)+R5*W(I,9)+R6*W(I,10)+
     *         R7*W(I,11)+R8*W(I,12)
   90 CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE TIM(AZ,EL,PLOTFC)
C     THIS SUBROUTINE WAS WRITTEN BY TIM BRADLEY.
C     THIS SUBROUTINE IS LAHEY FORTRAN SPECIFIC.
      DATA IIRST / 1 /
      TAZ=AZ
      TAZ=TAZ*PLOTFC
      TEL=EL
      TEL=TEL*PLOTFC
      IF (IIRST.EQ.1) THEN
        CALL PLOTS(0,0,0)
        CALL NEWPEN(5)
        CALL PLOT(TAZ,TEL,3)
        IIRST=IIRST+1
      END IF
      CALL PLOT(TAZ,TEL,2)
      RETURN
      END
```

# APPENDIX B. USER'S MANUAL FOR DIGITAL COMPUTER MODEL

This appendix contains a brief user's manual. A listing of the simulation model source code is contained in Appendix A.

The basic structure of the program is shown in Figure 10. The input data is set up in the main program (program TRACKR). The input data is described in this appendix in section 2.

Subroutine BROCKH is called from program TRACKR. In subroutine BROCKH, the simulation runs the desired number of time steps. For each time step, the subroutine does the following:

1. If closed loop (XOPENL < 1.), then calls subroutine GETPOS to update target position relative to the gyro.

2. Calls function subprogram FU1 to calculate the detector output (variable VS).

3. Determines value of variable KT1 based on the values of variables BP1 and Y(3) at the end of the last time step. KT1 is used in subroutine YPRIM to determine which demodulator model to use for this time step.

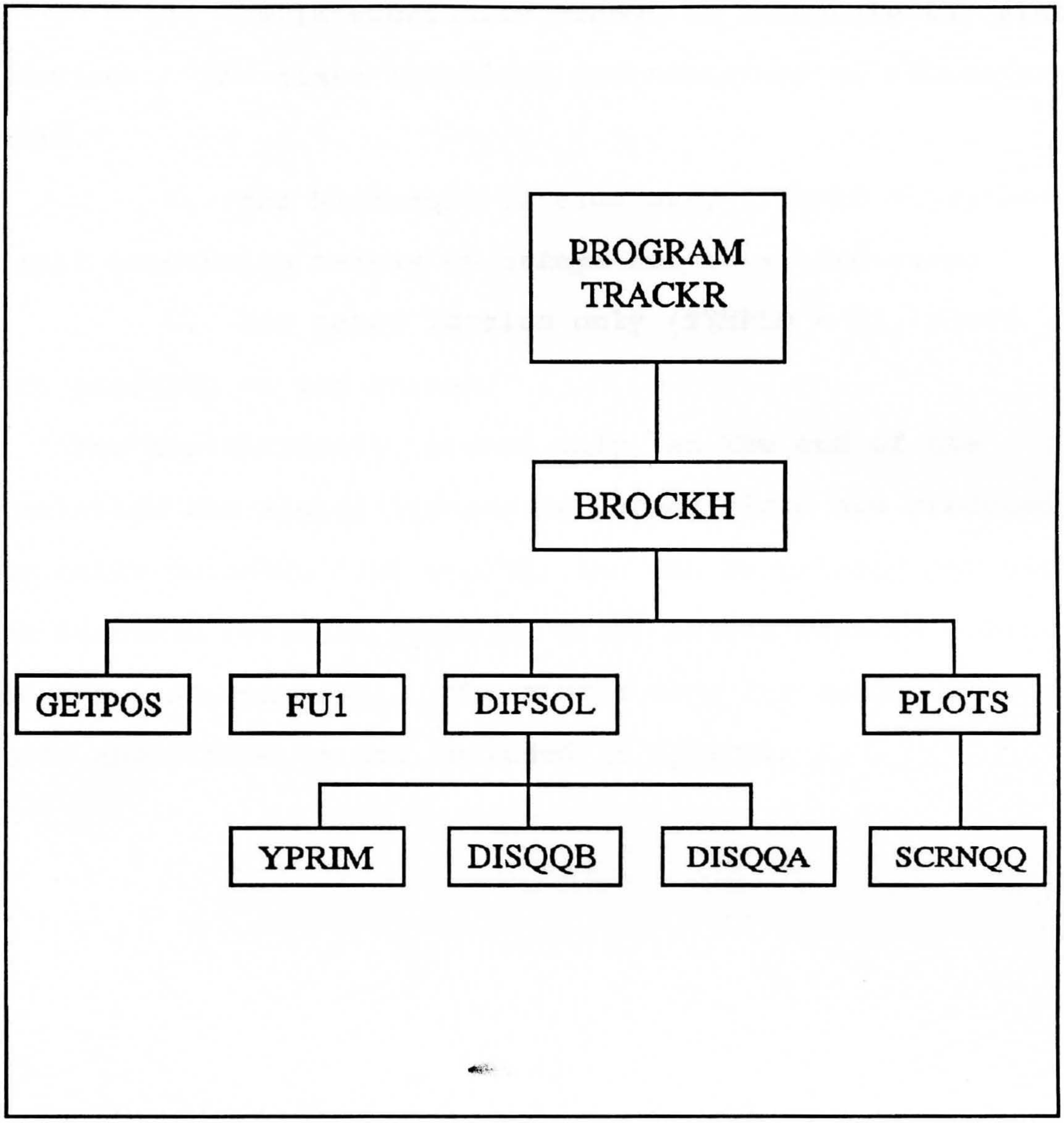



Figure 10. Basic Structure of Program.

4. Calls subroutine DIFSOL to integrate the state equations. The state equations are contained in subroutine YPRIM.

5. For Microsoft Version only (TIMPLO = 3), saves signal processing values to arrays for this time step.

6. For Lahey Version only (TIMPLO = 2), plots gyro position on the screen.

For the Microsoft version only, at the end of the simulation the signal processing screen plots are produced (by calls to subroutine PLOTS), and can be printed out with the standard personal computer print screen command (shift and print screen keys). The source code for subprograms PLOTS and SCRNQQ is not included in Appendix A.

1. COMMENTS ON INPUT AND OUTPUT

To compile the same source code on different FORTRAN compilers, several compromises are made including making the "input" part of the program.

The "input" was made part of the program for two reasons:

1. Since the program compiles quickly, little time is lost this way, and

2. The program compiles on various FORTRAN compilers with fewer input-output problems.

The output is designed to be independent of specific compilers except the following:

1. The runtime screen plot is Lahey FORTRAN specific (see Section 4 of this appendix).

2. The post run screen plots are Microsoft FORTRAN specific (see Section 3 of this appendix).

## 2. DEFINITION OF INPUT PARAMETERS

The input variables of main interest in the source code listing in Appendix A are defined as follows:

H is the simulation time step in seconds.

XF is the time (in seconds) to end simulation.

XGAIN is the loop gain and determines the maximum tracking rate; units of XGAIN are N*m/volt.

TGTAZ is the time zero target azimuth relative to the gyro (in radians); the time zero gyro azimuth is assumed zero.

TGTEL is the time zero target elevation relative to the gyro (in radians); the time zero gyro elevation is assumed zero.

TGSPRT is the target crossing rate (in radians per second)

RETPHA is the reticle phasing angle. RETPHA values from 0 to 2 correspond to 0 to 360 degrees respectively.

XOPENL is used for open or closed loop selection. Open loop if XOPENL is greater than or equal to 1; otherwise closed loop.

BANGBA is used to select the type tracking for small error angles. Bang-bang tracking is used if BANGBA greater than or equal to 1; otherwise the tracking rate is proportional to the tracking error.

DETECT is used to select how the detector output is generated. For DETECT less than 1, a math formula for AM modulation is used to calculate the detector output. For DETECT greater than or equal to 50, a blur circle is assumed in detector output calculation. Otherwise, a point is assumed in detector output calculation.

SCREEN selects runtime scroll of output to screen, one line per time step H for SCREEN not equal to 0. No scroll for SCREEN=0. Scroll controlled by STOUT and ENDOUT.

STOUT is time to start scroll to screen (in seconds).

ENDOUT is time to end scroll to screen (in seconds).

TIMPLO controls plot generation. For TIMPLO equal to 2, runtime plot for Lahey FORTRAN is generated, and no post run screen plots are generated. For TIMPLO equal 3, post runtime screen plots for Microsoft FORTRAN are generated (also see variable FILEOU). If TIMPLO is other than 2 or 3, then no plots are produced.

FILEOU is start time (in seconds) of 150 millisecond window for the postrun plots for Microsoft only (also see TIMPLO above).

## 3. MICROSOFT FORTRAN COMPILER COMMENTS

In order to compile the same source code on different FORTRAN compilers, several compromises are made including Microsoft specific source code.

The FORTRAN used is the PC version Microsoft FORTRAN77 V3.20. For use on other compilers, the subroutine PLOTS should be removed along with the calling statements. The only function of the PLOTS code is to produce post runtime screen plots of various signals of interest for the Microsoft version.

The input variable TIMPLO is set to 3 if post runtime plots are desired; otherwise any other value except 2 (which selects Lahey specific output).

The linker used is the Microsoft Overlay Linker Version 3.51. The following batch file is used to compile, link and run the program:

```
FOR1 %1,%1,%1,%1
PAS2
LINK %1,%1,NUL,D:NUMER+D:GRAPHS+D:8087+D:GRAPHICS
%1
```

When using the Microsoft FORTRAN77 V3.20 compiler, the source code for subroutine YPRIM must precede the source code for subroutine BROCKH.

4. LAHEY FORTRAN COMPILER COMMENTS

In order to compile the same source code on different FORTRAN compilers, several compromises are made including Lahey specific source code.

The FORTRAN used is the PC version Lahey F77L FORTRAN 77 version 4.00. For use on other compilers, the subroutine TIM should be removed along with the calling statements. The only function of the TIM code is to produce runtime screen plots of the tracker gyro position for the Lahey version.

The input variable TIMPLO is set to 2 if post runtime plots are desired; otherwise any other value except 3 (which selects Microsoft specific output).

The linker used is the Lahey OPTLINK Release 1.02. The following batch file is used to compile, link and run the program:

```
F77L %1
OPTLINK %1,,,GRAPH
%1
```

5. SAMPLE OUTPUT

This section contains the Microsoft specific post run screen plots for the source code listed in Appendix A. The plot of the gyroscope position for this open loop run is shown in Figure 9.

The detector output plot is shown in Figure 11. The bandpass 1 output plot is shown in Figure 12. The demodulator output plot is shown in Figure 13. The bandpass 2 output plot is shown in Figure 14. The torque $M_y$ plot is shown in Figure 15.

The open loop gyro response is shown in Figure 16. For this open loop run, the target position is held fixed relative to the gyro spin axis. The plot in Figure 16 shows the path of the gyro spin axis for the period from 0.085 to 0.1 second for a reticle phasing value of 0.0 and a loop gain value of 210. In Figure 16, the gyro spin axis direction at time zero is at the origin. The plot in Figure 16 is for the period from .085 to 0.1 second, and the plot in Figure 9 is for the period from 0.0 to .0999 second.

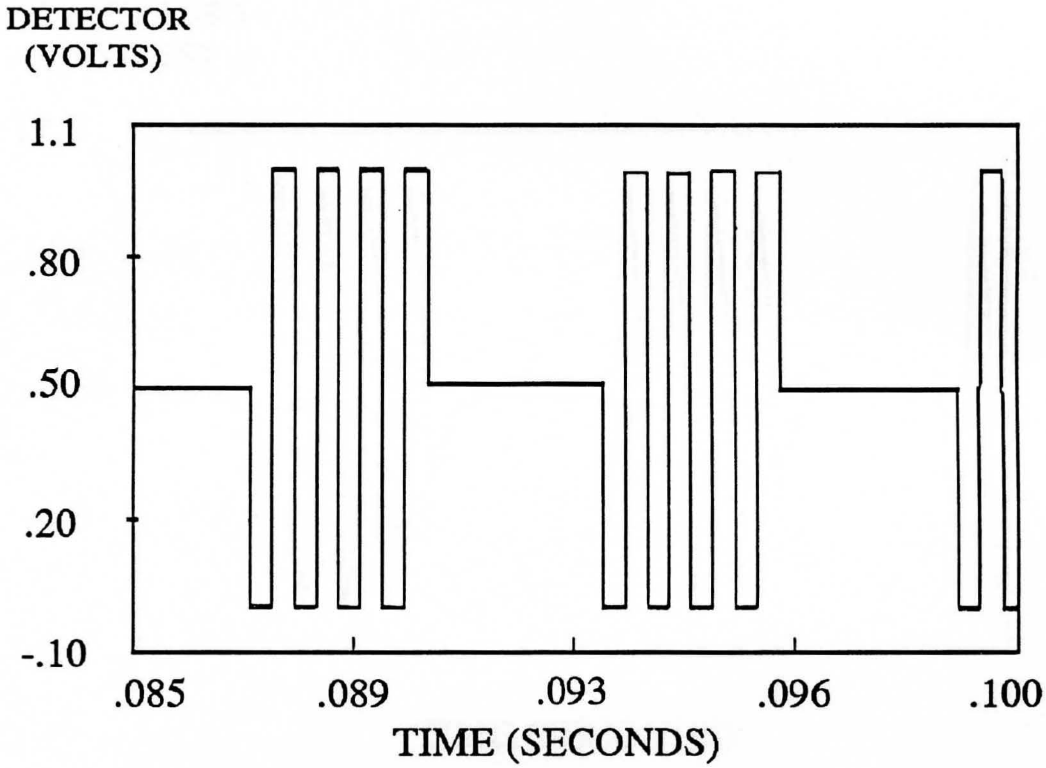



Figure 11. Detector Output.

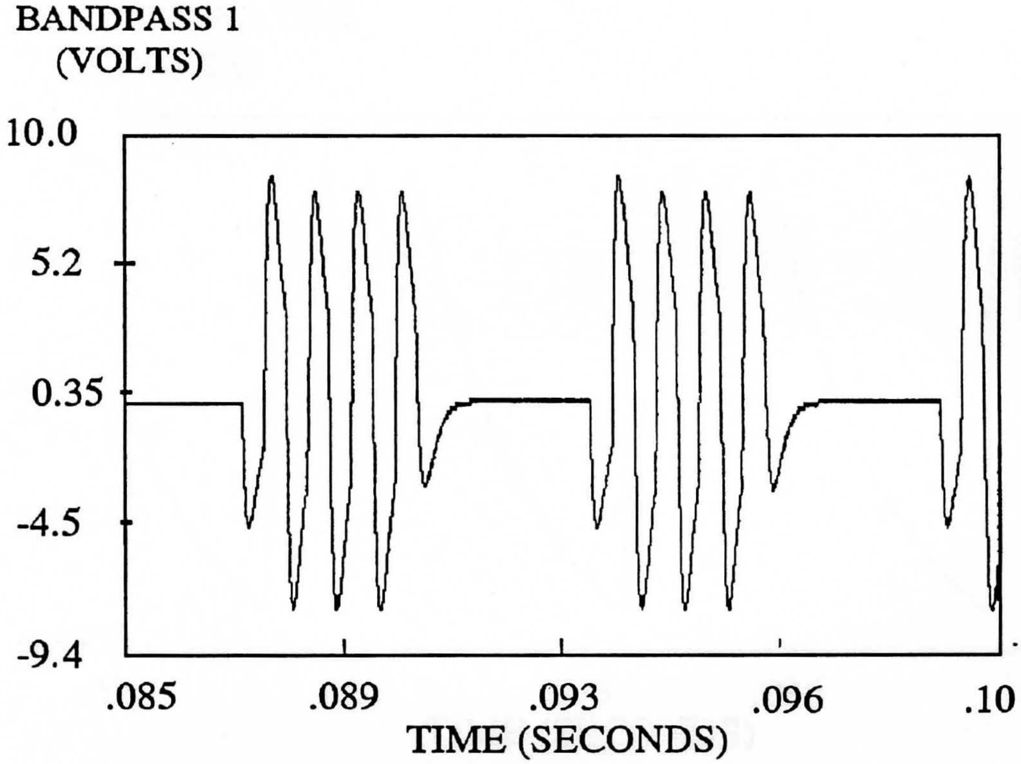



**Figure 12. Bandpass 1 Output.**

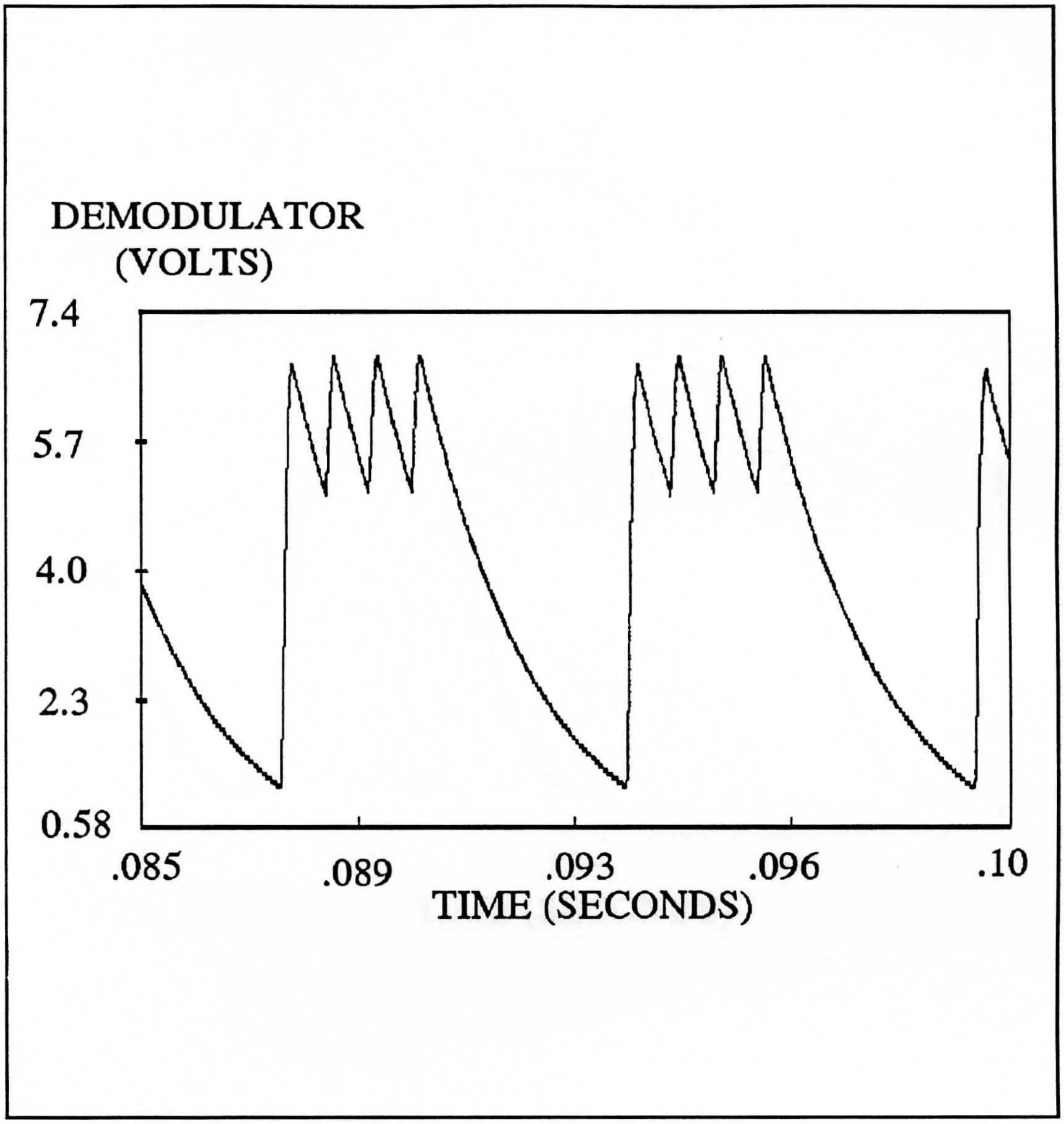



Figure 13. Demodulator Output.

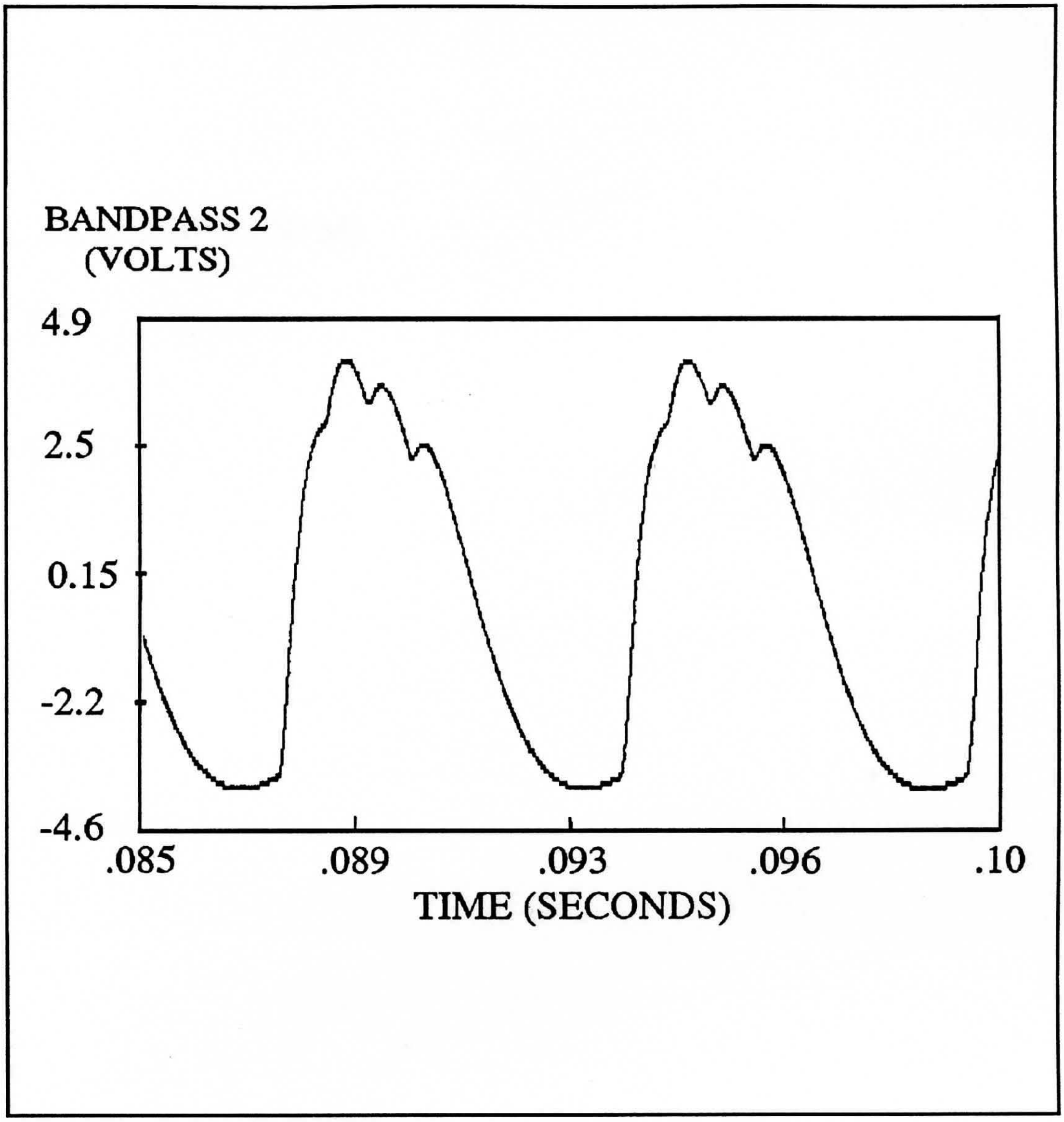



**Figure 14. Bandpass 2 Output.**

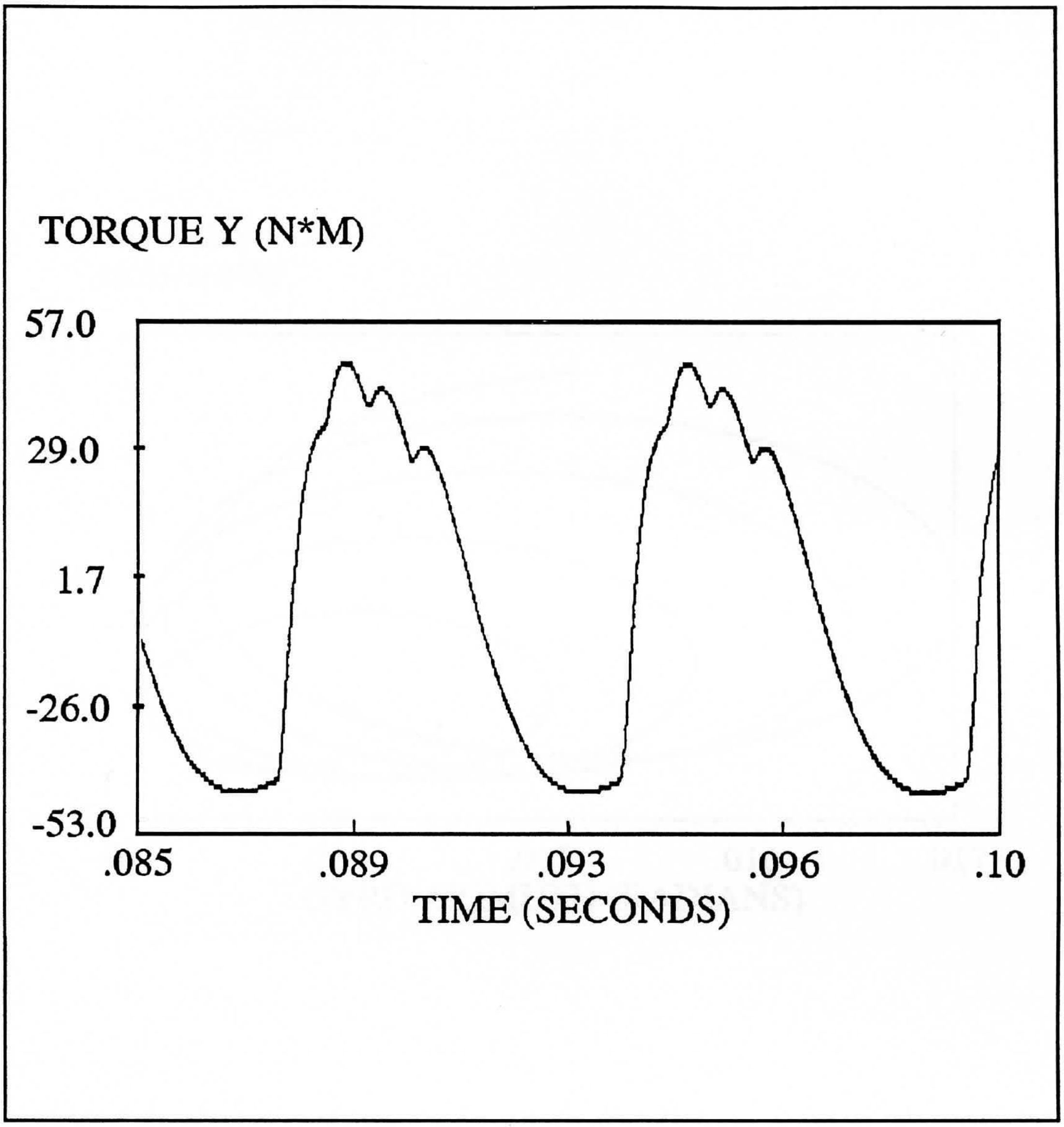



**Figure 15. Torque $M_y$ Plot.**

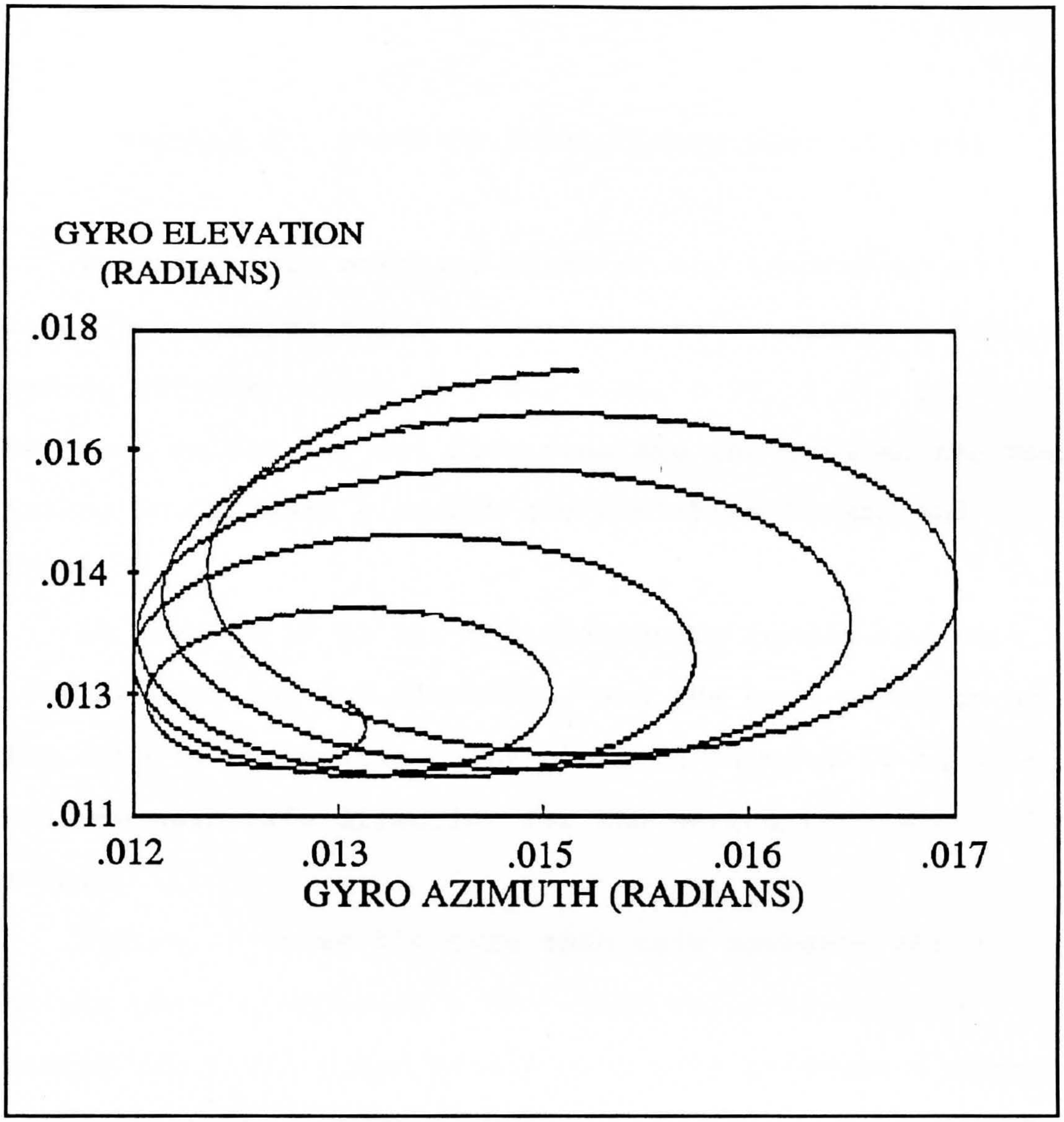



Figure 16. Gyro Position Plot.

# APPENDIX C. PLOTS OF MODEL CLOSED LOOP RESPONSE

This appendix contains plots of the tracker model closed loop response for 5 representative values of reticle phasing (RETPHA values of 0.00, 0.26, 0.71, 1.01, and 1.26). The input values for all five runs are the same as the model listing in Appendix A except the variables XOPENL and RETPHA.

In Figures 17 to 21, the stationary target location is 0.005 azimuth and 0.0 elevation, and the gyro position at time zero is the origin. The plots in Figures 17 to 21 show the gyro spin axis direction for the period from 0.0 to 0.1 seconds.

Figure 17 shows the gyro spin axis movement for a reticle phasing angle of 0.00. This value of reticle phasing (as would other values near 0.0) produces a closed loop response such that the gyro spin axis is moved, as is desirable, to a point near the target (with some "steady state" tracking noise after the gyro is pointed near the stationary target). The open loop response for the 0.00 value of reticle phasing is shown in Figure 9.

Figure 18 shows the gyro spin axis movement for a reticle phasing angle of 0.26. This value of reticle phasing produces a closed loop response that does not track the target.

Figure 19 shows the gyro spin axis movement for a reticle phasing value of 0.71. The closed loop gyro response is in the wrong direction since the target is located at an azimuth of .005 and an elevation of 0.0.

Figure 20 shows the gyro spin axis movement for a reticle phasing angle of 1.01. This value of reticle phasing produces a closed loop response that does not track the target.

Figure 21 shows the gyro spin axis movement for a reticle phasing angle of 1.26. This value of reticle phasing produces a closed loop response that is unstable.

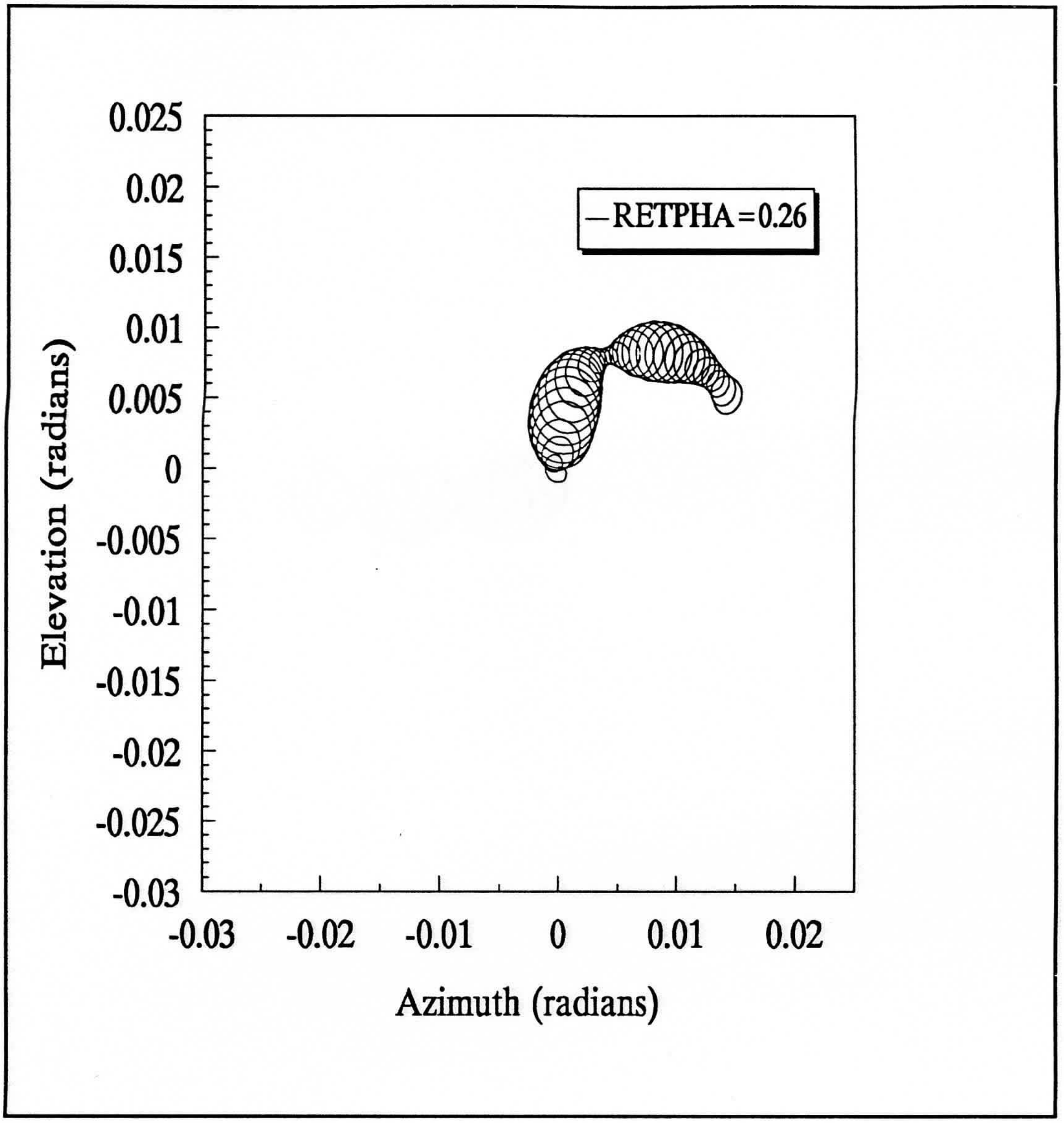



**Figure 18. Closed Loop Gyro Response (RETPHA = 0.26).**

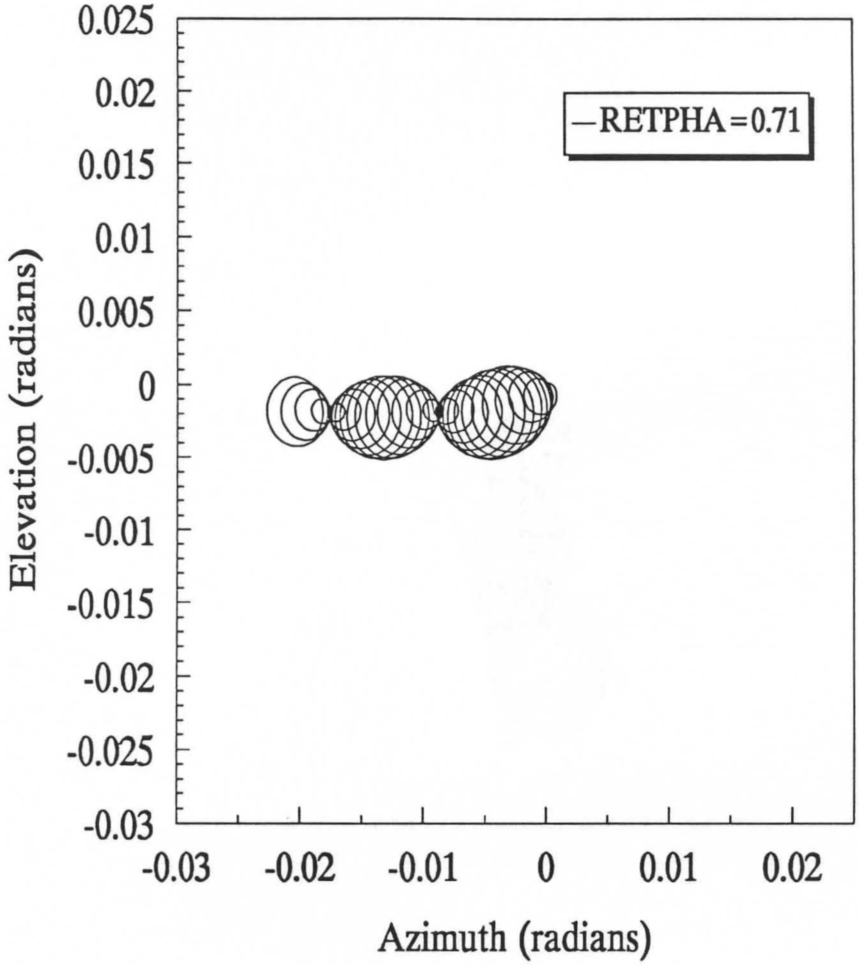



Figure 19. Close Loop Gyro Response (RETPHA = 0.71).

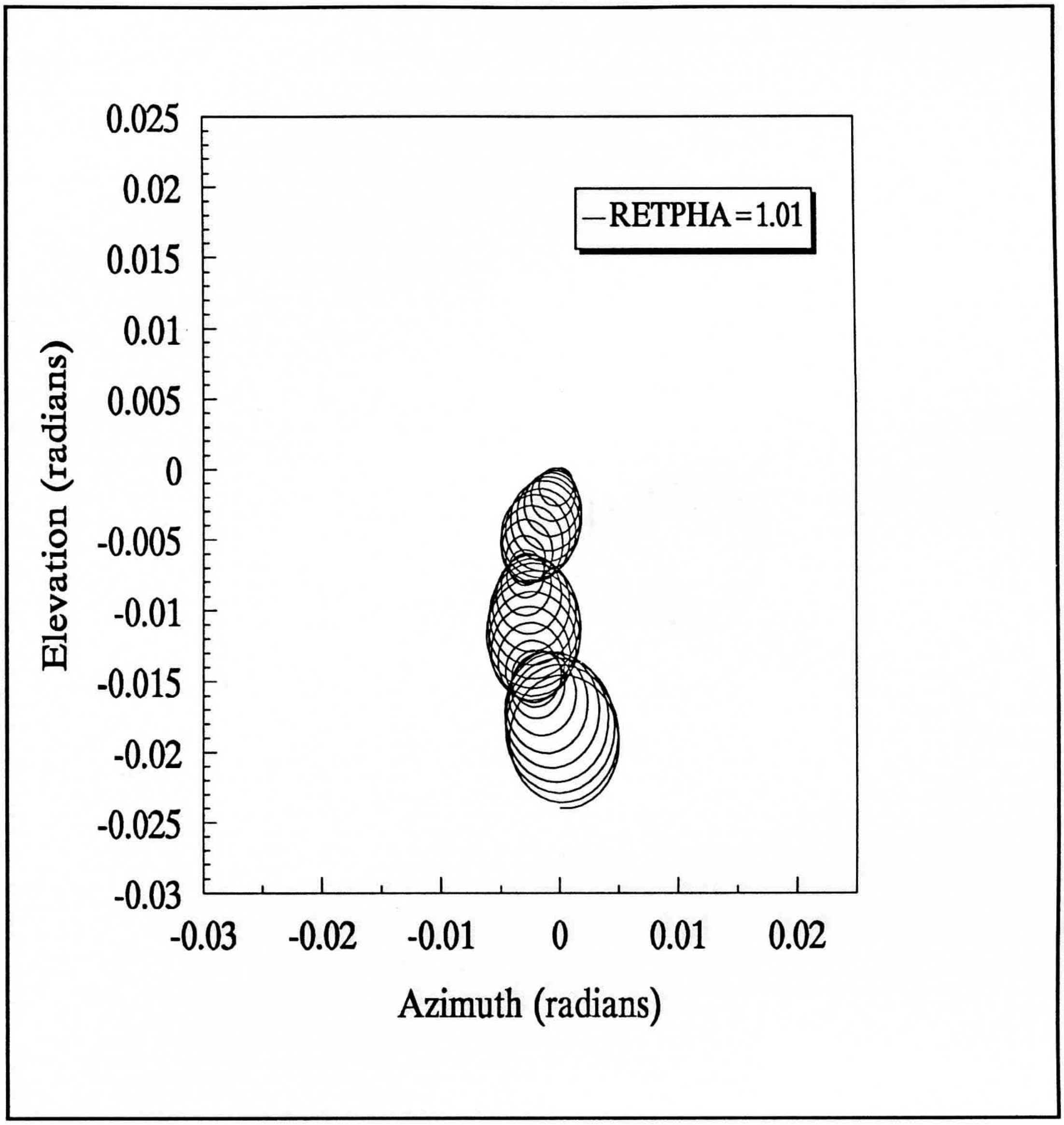



**Figure 20. Closed Loop Gyro Response (RETPHA = 1.01).**

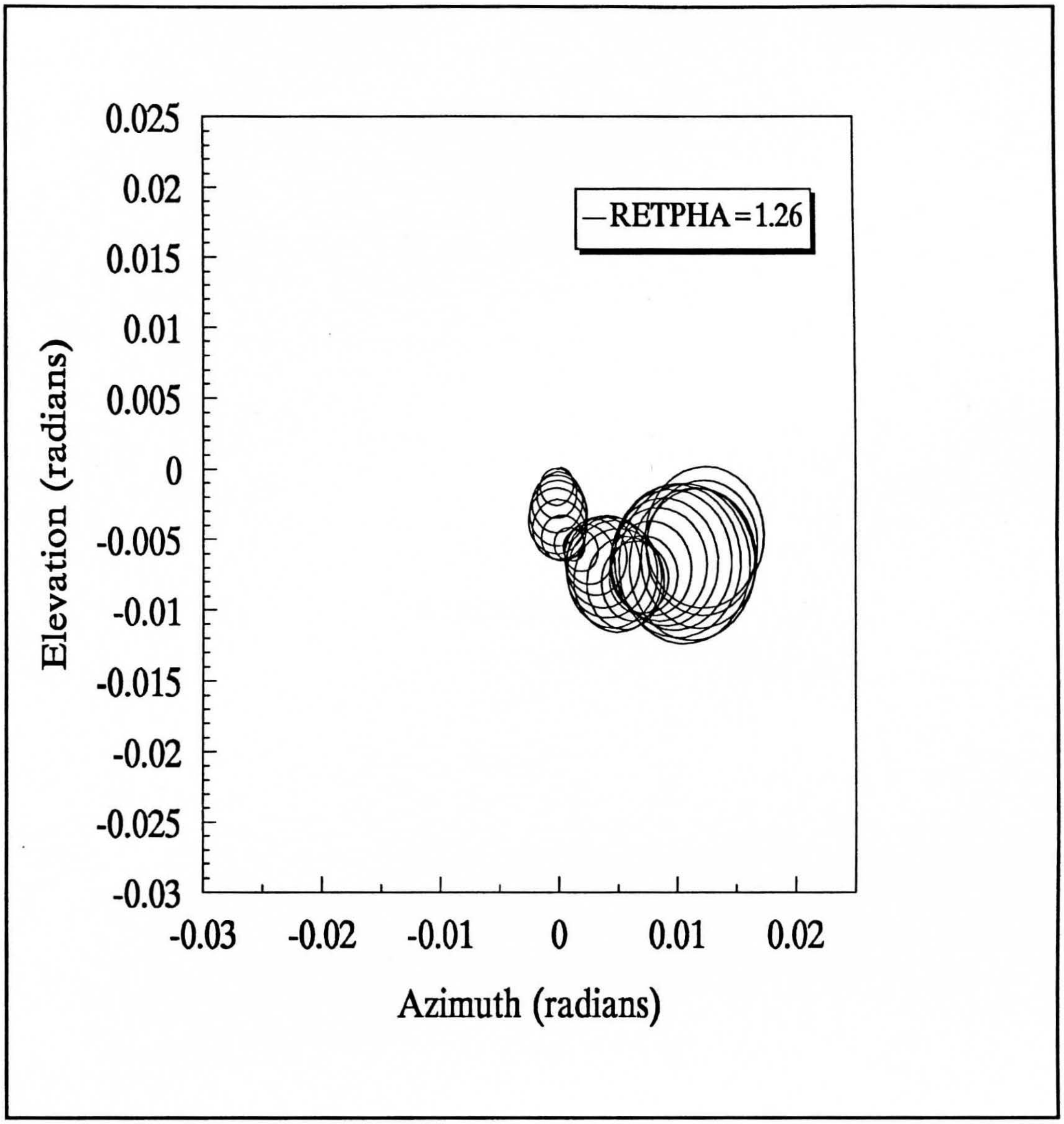



Figure 21. Closed Loop Gyro Response (RETPHA = 1.26).

# BIBLIOGRAPHY

# BIBLIOGRAPHY

1. Rosko, Joseph S. Digital Simulation of Physical Systems. Reading, Massachusetts, Addison-Wesley, 1972.

2. Wolfe, William L., and George J. Zissis (editors). "Tracking Systems" in The IR Handbook. Washington, D.C., Environmental Research Institute of Michigan for the Office of Naval Research, 1978.

3. Eichblatt, E.J. Jr., and A. Pignataro. Test and Evaluation of the Tactical Missile. Washington, D.C., American Institute of Aeronautics and Astronautics, 1989.

4. Carey, Jim. "Active Infrared Countermeasures Provide Continuous Protection for Today's Aircraft", The International Countermeasures Handbook. 8th Edition. Palo Alto, California, EW Communications, 1983.

5. Hudson, Richard D. Jr. Infrared System Engineering. New York, Wiley-Interscience, 1969.

6. Dow, Richard B. Fundamentals of Advanced Missiles. New York, John Wiley, 1958.

7. Scarborough, James B. The Gyroscope: Theory and Applications. New York, Interscience Publishers, 1958.

8. Friedland, Bernard. Control System Design: An Introduction to State Space Methods. New York, McGraw-Hill, 1986.

9. Garnell, P. and D.J. East. Guided Weapon Control Systems. Oxford, England, Pergamon Press, 1977.

10. Greenwood, Donald T. Principles of Dynamics. Second Edition. Englewood Cliffs, New Jersey, Prentice-Hall, 1988.

11. Ogata, Katsuhiko. Discrete-Time Control Systems. Englewood Cliffs, New Jersey, Prentice-Hall, 1987.

12. D'Azzo, J. and Houpis, C. Linear Control System Analysis and Design. New York, McGraw-Hill, 1975.

13. Kraus, John D. Electromagnetics. New York, McGraw-Hill, 1953.

14. Conte, S.D., and Carl deBoor. Elementary Numerical Analysis: An Algorithmic Approach. Third Edition. New York, McGraw-Hill, 1980.

15. IMSL. IMSL Math/Library User's Manual. Houston, Texas, IMSL, 1989.

16. Vetterling, William T., Saul A. Teukolsky, William H. Press, and Brian P. Flannery. Numerical Recipes Example Book (C). Cambridge, Cambridge University Press, 1988.

17. Becker, Kenneth A. Graphics and Numerics for Microsoft FORTRAN. 15 August 1985.

18. Allendoerfer, Carl B., and Cletus O. Oakley. Fundamentals of Freshman Mathematics. New York, McGraw-Hill, 1965.