NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMPUTER MODELING OF JAMMING EFFECTS ON ROLL STABILIZED MISSILES

by

Craig Alan Hill

September 2000

Thesis Advisor:	D. Curtis Schleher
Second Reader:	David C. Jenn

Approved for public release; distribution is unlimited.

20001027 114

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2000	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: COMPUTER MODELING OF JAMMING EFFECTS ON ROLL STABILIZED MISSILES	5. FUNDING NUMBERS
6. AUTHOR(S) Hill, Craig A.	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Air Weapons Center- Weapons Division, China Lake, CA	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Development of countermeasures against infrared missiles is enhanced by an ability to quantify the effects of the countermeasure. Analysts must be capable of accurately determining the attitude of the missile throughout its flight. This thesis describes the use of micro-miniature technologies to measure the rates experienced by a missile and the model required to effectively determine the missile's attitude.

The Applied Technology Associates ARS-04E and the Tokin America CG-16D sensors were evaluated for use as rate sensors and the Honeywell, SSEC, HMC1002 was evaluated for use as a roll sensor. Of these sensors, the CG-16D proved its ability to perform in this application. The ARS-04E was ineffective in this application.

A Simulink model is presented that performs the tasks of demodulating the sensors, performing coordinate transformation, and providing animation of the missile attitude for analysis. The model was evaluated for its ability to accurately determine the attitude of the missile based on input from the IMU packages. Sensor data was obtained from testing performed on a CARCO table flight motion simulator, and compared to the ground truth data provided by the CARCO table. Through testing, the model was capable of providing solutions within the 2 degrees RMS requirement.

14. SUBJECT TERMS Simulink, Missile Attitude, Inertial Measuring Unit, Euler Rotation, Animation	15. NUMBER OF PAGES 78
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

i

THIS PAGE INTENTIONALLY LEFT BLANK

COMPUTER MODELING OF JAMMING EFFECTS ON ROLL STABILIZED MISSILES

Craig Alan Hill
Lieutenant, United States Navy
B.S., Old Dominion University, 1993

Submitted in partial fulfillment of the
requirements for the degree of

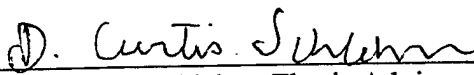## MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

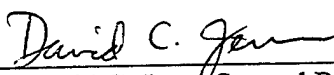## NAVAL POSTGRADUATE SCHOOL
September 2000

Author: _____
Craig Alan Hill

Approved by: _____
D. Curtis Schleher, Thesis Advisor

_____
David C. Jenn, Second Reader

_____
Dan C. Boger, Chairman
Computer and Information Sciences & Operations

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Development of countermeasures against infrared missiles is enhanced by an ability to quantify the effects of the countermeasure. Analysts must be capable of accurately determining the attitude of the missile throughout its flight. This thesis describes the use of micro-miniature technologies to measure the rates experienced by a missile and the model required to effectively determine the missile's attitude.

The Applied Technology Associates ARS-04E and the Tokin America CG-16D sensors were evaluated for use as rate sensors and the Honeywell, SSEC, HMC1002 was evaluated for use as a roll sensor. Of these sensors, the CG-16D proved its ability to perform in this application. The ARS-04E was ineffective in this application.

A Simulink model is presented that performs the tasks of demodulating the sensors, performing coordinate transformation, and providing animation of the missile attitude for analysis. The model was evaluated for its ability to accurately determine the attitude of the missile based on input from the IMU packages. Sensor data was obtained from testing performed on a CARCO table flight motion simulator, and compared to the ground truth data provided by the CARCO table. Through testing, the model was capable of providing solutions within the 2 degrees RMS requirement.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.   INTRODUCTION

A. BACKGROUND

Infrared (IR) guidance has increased the lethality of missile systems worldwide and has overcome the significant advances in electronic countermeasures that have served as a defense against radar guided systems. In the period 1979 to 1985, 90% of aircraft lost in combat were destroyed by IR guided missiles (Naval Air Systems Command [NAVAIRSYSCOM], 1998). Clearly, the threat embodied in the IR missile has exceeded that of the radar guided missile, yet the countermeasures developed against this threat have been limited, principally by an inability to effectively test and quantify the effects of countermeasures on IR guidance systems. Naval Air Weapons Center (NAWC), China Lake is one of the Department of Defense's principal researcher facilities in testing IR countermeasures.

Quantifying the effects of countermeasures requires very specific information about the response of the missile to the countermeasure. Most importantly, researchers must be capable of determining the exact position of the missile as well as its attitude throughout its flight.

Several methods are used to track missiles in flight and determine the Time, Space, and Position Information (TSPI) of the missile: 1) high-speed photography, 2) radar tracking, 3) laser tracking, 4) Global Positioning System (GPS) tracking. Each of these methods has its benefits. However, each system has significant deficiencies that prevent any one method from being superior.

High–speed photography provides accuracy to approximately 5 feet. However, generating the footage necessary perform this analysis is manpower intensive and the

time required to analyze the footage is extremely time intensive. This method requires the stationing of high-speed cameras throughout the range, each manned by an operator, who attempts to visually and manually track the missile with the camera. Once completed, the film is developed and each frame of the film is correlated to the angular position of the camera and then TSPI data is determined by the position of the missile in the frame. While high speed-photography provides good spatial accuracy, and access to attitude data, it is costly in manpower and time. (Naval Air Warfare Center, 2000)

Radar, laser, and GPS all reduce the manpower and time requirements of missile tracking. However, they do not provide as much information as high-speed photography, and they all vary in the accuracy achievable. Radar tracking provides relatively poor spatial accuracy, accuracy to hundreds of feet, but no attitude information. Laser tracking provides much better spatial accuracy, within inches, but again provides no attitude information. GPS provides spatial accuracy to within 30 feet, but again provides no attitude information. Like high-speed photography, these methods have strengths, but are limited by their individual deficiencies and none are capable of performing over water or in a multi-target environment. (Naval Air Warfare Center, 2000)

Recognizing the need for improved TSPI data, NAWC has developed the TM Tracker. TM Tracker was developed to operate in a wide variety of environments and weather conditions against multiple targets while achieving accuracy of 1 meter, using telemetry and the principles of time difference of arrival (TDOA). While TM Tracker can operate in the weather and environmental conditions specified, and provide the spatial accuracy required, it does not provide the information necessary to determine the attitude of the missile throughout its flight.

Analysts depend on missile attitude information to determine the reaction of the missile guidance systems to countermeasures. Without this information, detailed analysis is limited to the information that can be derived from the actual effects of the countermeasure as exhibited in the missiles flight path, limiting the value of the analysis.

Attitude data is best obtained through the use of gyroscope based inertial measuring units (IMUs). However, successful use of gyroscopes for rate measurement requires heading data from a stable platform for resolution of the rates. To overcome this obstacle, numerous other micro-miniature technologies have been used with success. In 1999, Dr. Curtis Schleher and Troy Johnson, at the Naval Postgraduate School, were able to achieve an accuracy of less than 2 degrees RMS with telemetry data from a quartz-rate sensor installed in a non-roll stabilized missile, a missile that does not spin. (Johnson, 1999)

Their research demonstrated that the output of micro-miniature components could be used to effectively recreate the attitude of the missile through the flight. However, their research did not extend to roll stabilized missile, which requires a different IMU package and model to demodulate rates based on roll angle. This thesis expands the work performed by Dr. Schleher and Troy Johnson to include roll stabilized missiles.

B. APPROACH

Reconstruction of missile attitude requires the collection of the rates experienced in the missile, and then introduction into a PC based model for interpretation and simulation. To collect the rate data, an IMU assembly is installed in the missile and the IMU sensor data is transmitted to a ground station where it is collected for analysis. The

3

IMU required for a roll stabilized missile consists of a sensor for sensing roll angle and two angular rate sensors for sensing rates relative to the pitch and yaw axes.

Once collected, the data is introduced into a conditioning model where bias is removed, and scale factors are applied prior use in the simulation model. Within the simulation model, the rate sensor data is demodulated using the roll angle, providing pitch and yaw rates in the missiles frame of reference, or strapdown. Cross coupling is then corrected. Once complete, then earth reference is applied through coordinate transformation. In coordinate transformation the attitude of the missile in pitch and yaw is determined based on the relation of the missile's rates, R, Q, and P, to the earth using the Euler rotation model. Once transformed, the pitch and yaw angles are used to provide a visual depiction of the missile throughout its flight.

The Euler rotation model operates on the conventions of the aerospace industry. As depicted in Figure 1, the attack angle (α) is the angle between the resultant velocity vector (v) and the x-axis of the missile. Angle of attack, or attitude, is defined by the pitch (θ), yaw (ψ), and roll (ϕ) angles of the missile. Yaw is defined as the angle between the central, x-axis, of the missile and the velocity component in the x-y plane. Pitch is defined as the angle between the central, x-axis, of the missile, and the velocity component in the x-z plane. Roll is defined by the relationship of the missile's central axis (x) to the velocity component in the y-z plane. The rates experienced relative to these axes are defined as R about the yaw axis, Q about the pitch axis, and P about the roll axis. These six components completely define the attitude of the missile in flight.

Demodulation of the rate sensor data requires highly accurate sensors. The roll sensors must be capable of faithfully recreating the roll position of the missile and the

4

rate must be capable of recreating the rates experienced without significant non-linear effects. The combination of rate and roll sensors composed the IMU package.
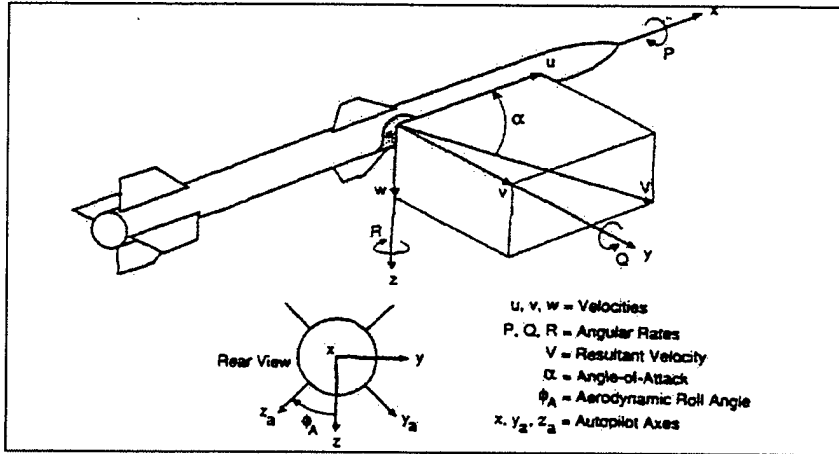

Figure 1. Strapdown Coordinate Axes and Conventions. From Eichblatt, 1989.

Two rate sensor technologies were explored for compatibility in the roll stabilized missiles and for processing in a PC based model. The first rate sensor, the Applied Technology Associates (ATA) ARS-04E, a magneto-hydrodynamic (MHD) rate sensor, was chosen due to its range, and its ability to fit in the roll stabilized missile. Additionally, previous testing by the Army Research Laboratory on munitions had lead to the purchase of a large quantity of these sensors. As testing began to indicate that the ATA sensors were inadequate, a second sensor was acquired for testing. The second rate sensor, the Tokin America CG-16D, a piezoelectric rate sensor, was chosen as an alternative because of its size, despite its specified range being less than desired.

Both IMU packages, ATA and Tokin sensors, were tested using a flight motion simulator. Additionally, the ATA sensors were tested in flight. The data relayed from the sensors during the tests were evaluated for effectiveness and accuracy, and compatibility with the Euler rotation model.

C. QUESTIONS ANSWERED

Several questions were answered in the conduct of this research. 1) Can the attitude of the rolling missile be effectively determined throughout a flight using the rate sensor and, roll sensor data, and a computer simulation model? 2) Can the ATA ARS-04E hydrodynamic rate sensor effectively sense and linearly portray the rates experienced by the missile in flight? 3) Can the Tokin CG-16D piezoelectric rate sensor effectively sense and linearly portray the rates experienced by the missile in flight? 4) Can the Tokin piezoelectric sensor operate over the range of rates expected?

The methods and quantitative results of this research are presented in the pages below. Briefly, the answers to these questions are presented here. 1) Yes, the attitude of a roll stabilized missile can be determined with a high degree of effectiveness and accuracy using rate and roll sensors, and a computer simulation model. 2) No, the ATA ARS-04E sensor cannot effectively sense and linearly portray the rates experienced by a missile in flight, apparently due to hysteresis and non-linear gain requirements. 3) Yes, the Tokin CG-16D sensor can effectively and accurately sense and portray the rates experienced by a missile in flight. 4) Yes, the Tokin CG-16D is capable of operating over the range of rates experienced by a missile in flight, exceeding the range specified.

II. SENSOR FUNCTIONAL ANALYSIS

The IMU for the rolling missile is composed of three principal components, one Honeywell HMC1002 dual axis magneto-resistive sensors to sense roll, and two rate sensors to sense pitch and yaw rates. The sensors chosen for these applications are described below. Within the IMU, the roll sensor is mounted along the roll axis of the missile, while the pitch and yaw sensors are mounted orthogonal to one another parallel to their respective axis.

The IMU provides the three components necessary to completely describe the attitude of the missile, pitch and yaw rate, and roll position. Pitch and yaw rates are necessary to solve for the pitch and yaw angles. Roll position is necessary to provide the phase information required for demodulation of the pitch and yaw rates.

A. ROLL SENSOR

The sensor chosen for the roll was the HMC1002 Dual-Axis Discrete Magnetic Sensor, a magneto-resistive (MR) sensor designed and built by Honeywell. The HMC1002 is a micro-miniature, low field, solid state sensor capable of measuring direction and magnitude of a magnetic field of $\pm$ 2 Gauss. Most importantly for this application, it is small and rugged enough to be installed within the IMU package, and is capable of sensing the earth's magnetic field. A more complete set of specifications is presented in Table 1. (Honeywell SSEC, 1996)

The HMC1002 operates on the principle of anisotropic magnetoresistance (AMR). AMR causes a change in resistance in a ferrous material when a magnetic field is applied across the material. The magnitude of the resistance changes with the angle of incidence of the magnetic field on the conductor. When parallel to the flow of current,

7

the added resistance is zero. When orthogonal to the flow of current, the added resistance is at its peak. These changes in resistance, depicted in the output voltage of the sensor, are highly predictable and can be used to determine the angle of incidence of the ambient magnetic field.

Sensitivity	3 mV/V/Gauss
Field Resolution	40 μGauss
Field Range	± 2 Gauss
Linearity	± 0.5-1% full scale
Bandwidth	Over 1 MHz

Table 1. HMC1002 Specifications. After Honeywell SSEC, 1996.

The HMC1002 is installed along the x or central axis of the missile. As the missile spins, the output voltage reaches its greatest positive magnitude as the magnetic flux on face of the MR strip reaches its maximum. As the missile rolls 90 degrees and the MR strip becomes parallel to the flux of the magnetic field, the added resistance is zero. When the missile rolls another 90 degrees, the added resistance is again at its maximum but inverted to give a negative voltage. As the missile spins through 360 degrees, the sensor output describes a sinusoidal wave that reflects the roll position of the missile (ϕ).

B. RATE SENSORS

The rate sensors employed in the IMU are absolutely vital to accurate modeling of the missile's attitude. The rate sensors are used to measure the rates, Q and R, experienced by the missile about its pitch and yaw axes. These sensors must be small enough for installation in the IMU package. They must have a high enough range and linearity over that range to accurately measure the range of rates experienced by the

8

missile in flight. And, they must have extremely small cross axis sensitivity. Size is the characteristic that limits the use of most rate sensors in this application.

The range of rates and linearity across that range was required to be high in both cases. Initial testing performed at White Sands Missile Range using the ATA ARS-04E sensors indicated that the missile experienced rates in pitch and yaw that approach 200 degrees per second. Based on this information, sensors were selected that could accommodate those rates.

While constant gain factors are preferred, linear changes in the required scale factors are acceptable assuming that expected rates and their corresponding scale factors are known. However, as these scale factors become non-linear, the conditioner will no longer be capable of correcting the deficiency. Therefore, if scale factors vary non-linearly, they must be corrected in the model.

Cross axis sensitivity is also a major concern when dealing with angular rate sensors. The yaw and pitch axes are by definition orthogonal to one another. Referring back to Figure 1, yaw is the angle between the central, x-axis, of the missile and the component of the velocity in the x-y plane, and pitch is the angle between the central, x-axis, of the missile and the component of the velocity in the x-z plane. In order for the sensors to accurately read the yaw or pitch rates, they must be perfectly aligned in their respective planes. Any deviation from perfect alignment will cause undue influence of rates along one axis on the rate sensor aligned to the other axis. Effectively, the sensor with high cross axis sensitivity will be affected by rates that are not along its sensitive axis and will therefore provide incorrect output. In order to avoid this situation, sensors must be orthogonal with resulting low cross axis sensitivity.

Two sensors met these requirements. They are described below.

1. Applied Technology Associates

Of the two rate sensors chosen for analysis, the Applied Technology Associates

(ATA) ARS-04E rate sensor appeared to be the most promising. Its operating

characteristics met or exceeded all requirements for use in this application. The ARS-

04E operates over an extremely high range of rates, up to ± 5,750 degrees per second. It

is small enough for installation in the IMU package of a 2.75 inch missile, .8 inches long

by .425 inches wide, and .461 inches high. It can operate under extreme conditions of

linear acceleration, up to 500 times the force of gravity on any axis. Additionally, the

ARS-04E has no moving parts, has low cross axis angular sensitivity, reducing the

likelihood of cross-coupling, and is low noise. A complete set of specifications is

presented in Table 2. (Applied Technology Associates [ATA], 1999)

Range	± 100 radian/sec (± 5,750 degree/sec)
Sensitivity	100 mV/radian/sec (1.7 mV/degree/sec)
Bandwidth	.5 to 1000 Hz
Cross-axis Angular Error	<2%
Linear Acceleration Sensitivity	<0.005 radian/sec/g (<0.03 gdegree/sec/g)
Non-linearity	<0.1%
Linear Acceleration, Max Operating	500g any axis
Linear Acceleration, Max Survivable	800g any axis

Table 2. ARS-04E Specifications. After ATA, 1999.

2. Tokin America

The second of the two rate sensors evaluated was the Tokin America Inc., CG-

16D. Designed for application in vehicle navigation systems, the Tokin sensor met the

size requirements at 8x20x8 millimeters, but had a specified maximum detectable angular

rate of only ± 90 degrees per second which was considered to be too low for the

application in missile testing. A more complete set of specifications is presented in Table

3.(Tokin America Inc., 2000)

Maximum Detectable Angular Rate	$\pm$ 90 degree/sec @25degrees C
Sensitivity	1.1 $\pm$20% @ 25degrees C
Frequency Response	100 Hz min

Table 3. CG-16D Specifications. After Tokin, 2000.

A ceramic gyro, the Tokin sensor has a very basic construction of a ceramic

column printed with electrodes and operates on the principle of the piezoelectric effect.

In short, the piezoelectric effect occurs when an ionic bonded crystal is placed under

stress. When stress is applied, the crystal deforms and a dipole moment is created. This

dipole moment creates an electric field that, in turn, generates a charge that is

proportional to the pressure applied. In the presence of reciprocating pressures or rates as

experienced by a rolling missile, an alternating current is produced. This current reflects

the rates that the sensor experiences about the pitch or yaw axis. (Texas Instruments,

1999)

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODEL DESCRIPTION

A. ENVIRONMENT

Modeling of the roll stabilized missile was performed using SIMULINK 3.0.1, an interactive tool packaged with and utilizing the MATLAB language environment produced by The MathWorks Inc., and MATLAB's 3-dimensional graphics functions. MATLAB is a language much like C that was specifically optimized for use in matrix and vector calculations. SIMULINK extends the functionality of MATLAB by providing an easy to use, point and click user interface for modeling, and simulation of dynamic systems. Using the ordinary differential equation solvers of the MATLAB language, SIMULINK provides near real time solutions to simulations and models and can access all the functionality of the MATLAB language.

Data collection is performed on site at the national missile ranges. Sensor data is transmitted from the missile and is processed to 12 bit accuracy at a sampling rate of 1389 Hz. Once collected, the data is converted to ASCII format in columnar tables. Because of SIMULINK data input requirements, several changes are made to the files. First, SIMULINK operates only on MATLAB binary files that are in the form of matrices in which time is stored in the first row. Conversion requires the removal of header data from the ASCII file. Then, the file must be opened in the MATLAB environment, transposed and saved as a MATLAB binary file. Additionally, time is provided in "IRIG," or range time, and must be changed to a zero origin time with step sizes of 1/1389. When these changes have been completed, the data is ready for introduction into the SIMULINK environment.

B. FUNCTIONAL DESCRIPTION

The model performs four principal operations, 1) conditioning, 2) demodulation, 3) coordinate transformation, and 4) animation. These operations are all performed in the SIMULINK environment using the SIMULINK and MATLAB functions. The functional diagram is presented as Figure 2.
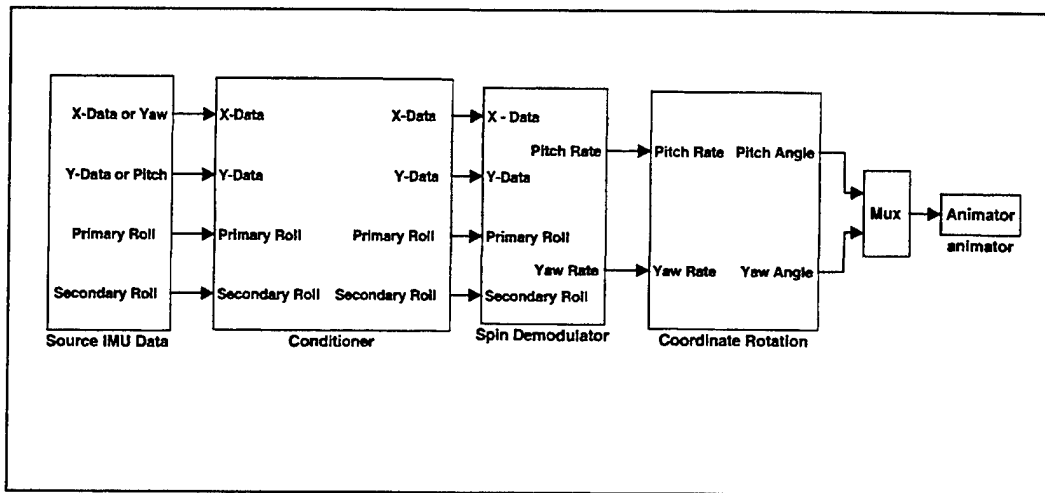

Figure 2. Model Functional Diagram.

1. Conditioning

Conditioning is performed on the data to account for inaccuracies in bias, and scale introduced by the sensors. Figure 3 represents the functional flow of the conditioner. The conditioners used for testing are presented in Appendix B. Conditioning is performed as the first step to reduce the mitigation of error throughout the model. Because integration is used for coordinate transformation, any error left uncorrected is compounded throughout the time period of the simulation, leading to significant errors in the final solution.

Bias is a product of each individual sensor. It is exhibited as a DC value output from the sensor though no rate is present on the sensor. Bias is typically constant through
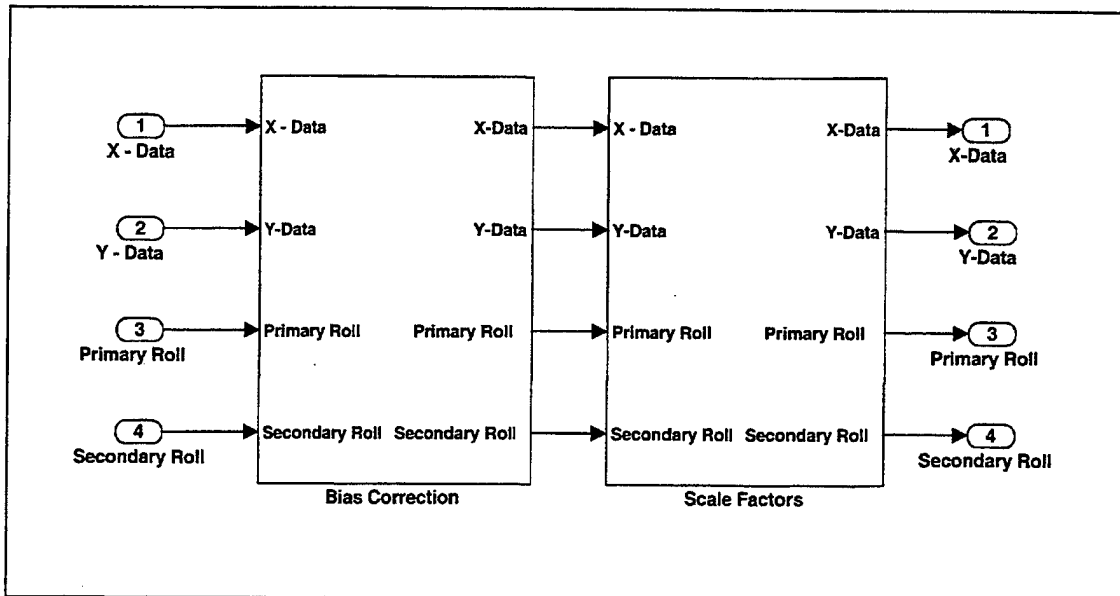
Figure 3. Conditioner Functional Diagram.

time. It is determined by averaging the value of the samples over time. The resulting

average or DC value is the bias, which is easily removed by either adding or subtracting a

constant to each sample to offset the average to zero. The effects of uncorrected bias are

presented in Figure 4. In this figure, the model outputs using bias corrected and non-bias

corrected data are depicted. In this case, the uncorrected bias resulted in a 23% increase

in RMS error in yaw and a 102% increase in RMS error in pitch.

Scaling errors are errors of magnitude indicated by the sensor. Both types of rate

sensors output a voltage proportional to the rate that the sensor is experiencing. The

magnitude of this voltage must be scaled to accurately indicate the rate being

experienced. This scale factor is typically provided by the manufacturer but can be

determined through hardware-in-the-loop tests. These tests must be performed in a

controlled environment such as a flight motion simulator where rates are known. This

approach is impossible in live fire applications due to the unknown rates experienced by

the missile. To correct for scaling errors, each sample of the data is multiplied by the inverse of the scale factor.
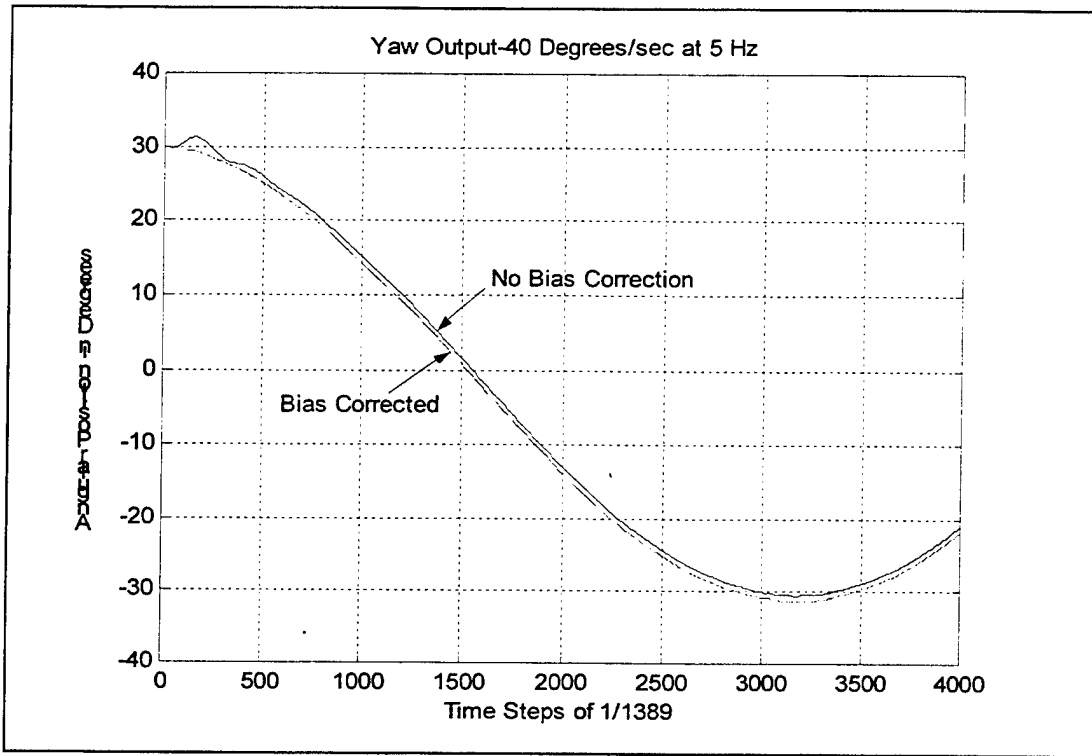

Figure 4. Effects of Non-Bias Corrected Data on Model Output.

Failure to account for and correct either of these errors will reduce the accuracy of the data input to the model. This in turn will prevent accurate solution of the missile's attitude.

2. Demodulation

The next operation performed by the model is the demodulation of the yaw, X-rate, sensor and pitch, Y-rate, sensor data. Because of the rolling action of the missile, demodulation requires a phase reference provided by the roll sensor installed in the missile. The functional diagram of the demodulator is presented in Figure 5.

Because the roll sensor may not necessarily provide a purely sinusoidal wave, the roll sensor output is conditioned by one of two methods, phase locked loop or arctangent

function. Both of these methods were used and compared for accuracy and effectiveness in the model.

The phase locked loop uses a voltage-controlled oscillator (VCO) to output a sinusoidal wave matched to the frequency of the roll sensor. The VCO must be tuned to the estimated frequency of the missile's roll and its gains must be adjusted to allow the VCO to accurately track and adjust to any changes in the missile's roll frequency.
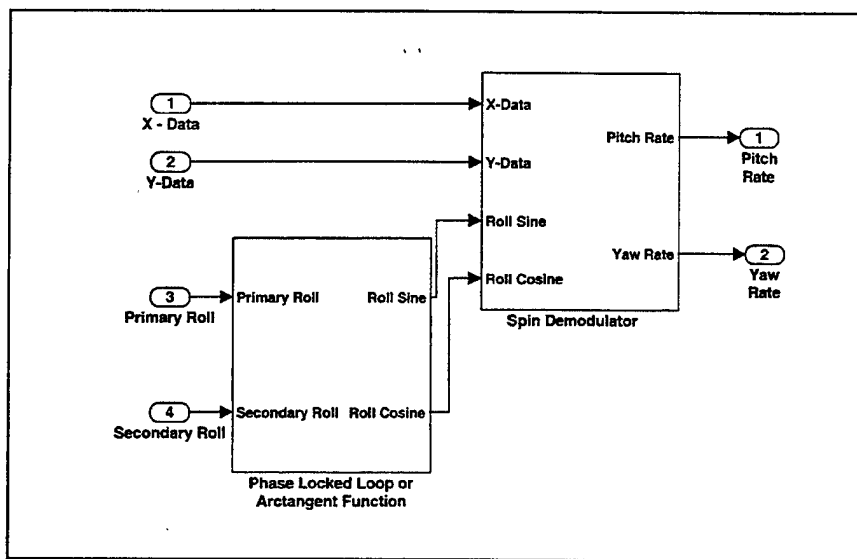

Figure 5. Demodulation Functional Diagram.

The arctangent function takes advantage of the dual axis output of the Honeywell HMC1002 sensor. In this method, a four quadrant inverse tangent is taken using the two roll sensor outputs. The output of the SIMULINK "arctan2" function is then split and run through the "sine" and "cosine" functions of MATLAB. The output of the sine function now matches the output of the roll sensor. However, because it is a function of the dual axis outputs, any common perturbations experienced by the roll sensor can be eliminated.

With a clean roll reference, demodulation is performed using the roll sine and roll cosine signals to separate the pitch and yaw components from the X and Y rate sensor

data. Because the missile is rolling, each sensor only outputs the rate along its designated

axis as it becomes aligned with that axis. At all other phases of roll, a sensor's output is

the resultant of the rates along the pitch and yaw axes. Visually, this is described in

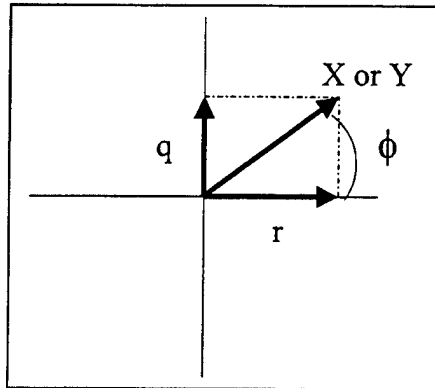Figure 6 and mathematically, in Equations 1 and 2.

Figure 6. Components Of
Sensor Output.

Demodulation of the pitch and yaw rates requires the decomposition of the

sensor output into its component parts. This is accomplished using Equations 3 and 4.

Once the rates along the pitch and yaw axes are known, they must then be corrected for

cross-coupling.

$$X = r \cos \phi + q \sin \phi$$

$$Y = q \cos \phi - r \sin \phi$$

Where:
X = X-Rate Sensor Output
Y = Y-Rate Sensor Output
r = Rate along the yaw axis
q = Rate along the pitch axis
ϕ = Roll Angle

Equations 1 and 2. Sensor Output.

$$Q = Y \cos \phi - X \sin \phi$$

$$R = Y \sin \phi + X \cos \phi$$

Where:
X = X-Rate Sensor Output
Y = Y-Rate Sensor Output
R = Yaw Rate
Q = Pitch Rate
ϕ = Roll Angle

Equations 3 and 4. Yaw And Pitch Rate.

Cross-coupling is the sensation of a rate by a sensor outside of its sensitive axis. Most commonly, cross-coupling is caused by a failure to ensure that the sensors are installed orthogonal to one another. This is best represented by placing an angular rate immediately along the axis of one of two "orthogonal" sensors. This rate should only be sensed in the sensor whose sensitive axis lies along the direction of that rate. If the second sensor is not truly orthogonal, then that sensor will sense a small portion of the rate. The result will be an addition to that sensor's output beyond what it is sensing along its own intended axis of sensitivity. Figures 7 and 8 demonstrate the effects of cross-coupling and effective decoupling. In this case, the output rate should be zero.

Cross-coupling is easily identified by observing excitations about a single axis. If this can be accomplished for single axis excitation about both axes, a clear determination can be made for the amount of cross coupling. In the absence of single axis excitation on both axes, the cross-coupling can be determined through trial and error. With the amount of cross-coupling identified, it can be easily removed by adding or subtracting an appropriate fraction of the opposing sensor's output.
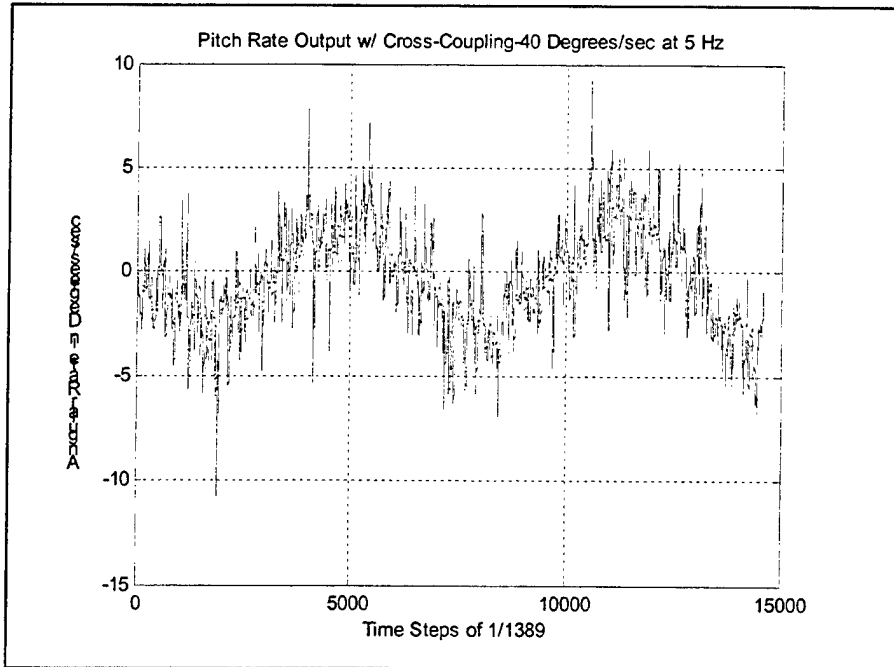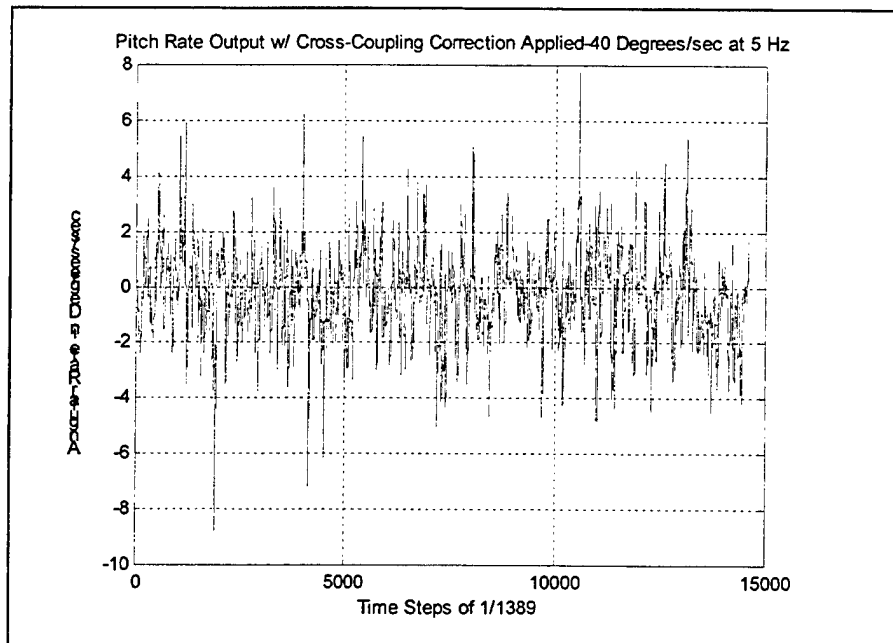
Figure 7. The Effects Of Sensor Cross-Coupling.

Figure 8. Corrected Rate After De-Coupling.

3. Strapdown to Earth Reference

Once demodulation is complete, the resulting rates are introduced into the Euler

rotation model for coordinate transformation from strapdown to earth reference. The

transform effectively takes the rates experienced by the sensors that are relative to the missile coordinate system, or strapdown, and applies them to an earth based reference system. In this manner, the missile's attitude with respect to the earth is determined.

Because Euler rotation uses three variables to represent three degrees of freedom and does not suffer from drifting, it is the preferred method for conducting the coordinate transformation (Bobick, 1998). In the Euler rotation, Euler angles are the variables by which the missile's attitude is described. Many systems of performing Euler rotation are used however, the model is built using the system defined by the aeronautical engineering discipline.

To visualize this system, a rigid body, in this case a missile, lies in a coordinate system, xyz, fixed to the missile in which the origin lies at the center of gravity of the missile. The positive x-axis extends through the nose of the missile and serves as the roll axis. The positive y-axis extends out of the tip of the right wing and serves as the pitch axis. The positive z-axis extends from the bottom of the missile and serves as the yaw axis. This situation is displayed in Figure 9.
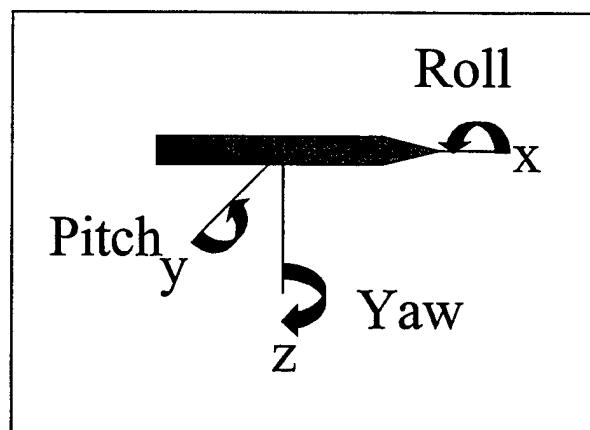

Figure 9. Missile Coordinate System.

The Euler angles are the yaw angle Ψ, the pitch angle θ, and the roll angle ϕ.

Initially this coordinate system coincides with that of the earth, XYZ. However, the

position of the missile relative to the earth coordinates, XYZ, can now be described using

the Euler angles or rotations as follows:

1. A positive rotation ψ about the Z-axis, resulting in the primed system.
2. A positive rotation θ about the y' axis, resulting in the double-primed system.
3. A positive rotation ϕ about the x" axis, resulting in the final unprimed system.(Greenwood, 1988)

The order of this rotation, yaw, pitch, then roll, is characteristic of this specific
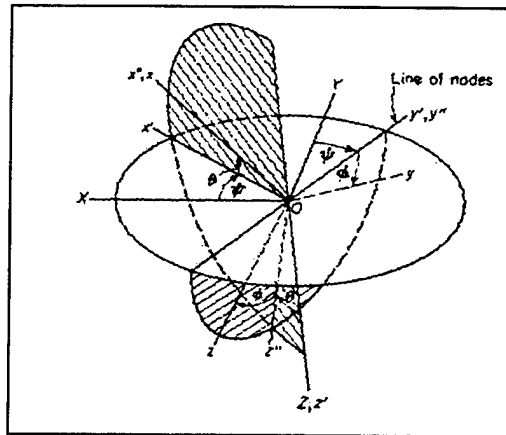
system. This rotation is demonstrated in Figure 10.

Figure 10. Euler Rotation. From Greenwood, 1988.

In going from the missile coordinate system, xyz, to the earth coordinate system,

XYZ, Equation 5 is prescribed. As is common, the dot notation indicates the first

derivative with respect to time. R, Q, and P are the rates in strapdown, the missile's

frame of reference. ψ, θ, and ϕ are the yaw, pitch, and roll angles, respectively, in the

earth coordinate system. For ease of implementation, in MATLAB, it was necessary to

restate this equation in the form defined in Equations 6, 7, and 8.

Once these equations have effectively solved the strapdown rates into earth coordinate rates, the rates are integrated to convert them from angular rates to angles. The initial earth angles are input by the user and are used to initialize the integration.

One limit to the Euler method of rotation is the problem of "Gimbal Lock." "Gimbal Lock" occurs when one of the rotations approaches 90 degrees. When this occurs, the angular velocity component along one of the axes cannot be represented using Euler angles and therefore, one degree of freedom is lost. This effect occurs in all of the Euler rotation systems and cannot be avoided while conducting an Euler rotation unless rotations approaching 90 degrees are restricted. In the case of this model, missile turns of 90 degrees are not expected and therefore "Gimbal Lock" is not expected to be a limiting concern. (Greenwood, 1988)

$$
\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi/\tan\theta & \cos\phi\tan\theta & 1 \\ \sin\phi/\cos\theta & \cos\phi/\cos\theta & 0 \end{bmatrix} \begin{bmatrix} Q \\ R \\ P \end{bmatrix}
$$

Equation 5. Euler Equations For Translation Of Rates From Missile To Earth Coordinates. After Blakelock, 1991.

$$
\frac{d\psi}{dt} = Q\frac{\sin\phi}{\cos\theta} + R\frac{\cos\phi}{\cos\theta}
$$

$$
\frac{d\theta}{dt} = P + \frac{d\psi}{dt}\sin\theta
$$

$$
\frac{d\phi}{dt} = Q\cos\phi - R\sin\phi
$$

Equations 6, 7, and 8. Alternate Euler Equations Used In The MATLAB Environment.

23

4. Animation

Animation is performed using a modified version of the SIMQUAT demonstration provided in the SIMULINK environment. The SIMQUAT demonstration allows a user to specify angles or angular rates either for Quaternion or Euler rotation visual display. The functionality for the Euler rotation is provided by the MATLAB s-file, EULERROTDISPLAY. This function uses the input Euler angles and sends them to the display where an aerodynamic structure created using MATLAB "patch" graphics is rotated accordingly. The EULERROTDISPLAY s-file was altered in order to accommodate a continuous input of Euler angles from the model rather than through a graphical user interface. Additionally, the coding for the display was altered to allow the user to pause and restart the animation at any time during the simulation. The resulting display is presented in Figure 11. The MATLAB code necessary to perform animation is presented in Appendix A.
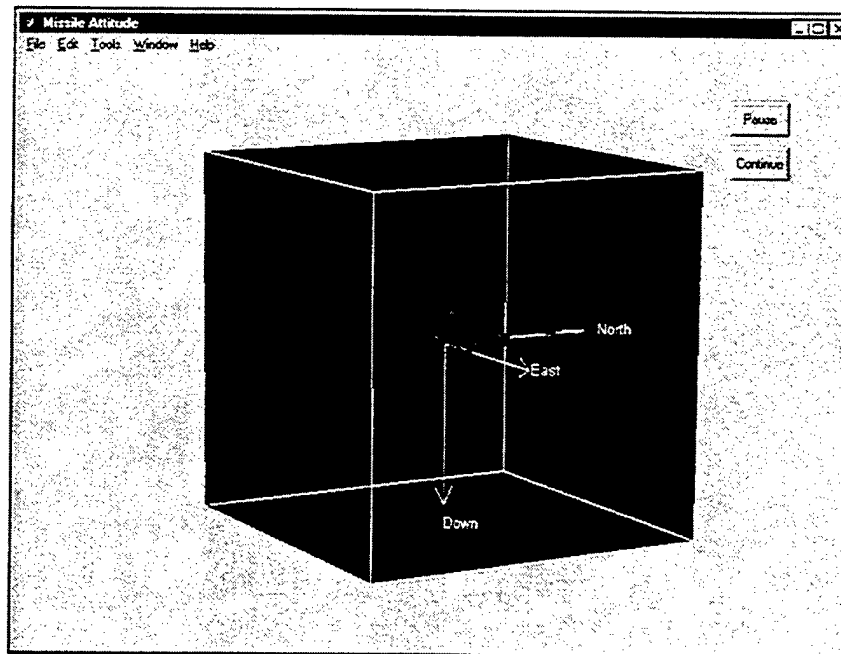

Figure 11. Missile Attitude Graphical Display.

C. MODEL VALIDATION

In order to perform model validation, the model was altered to incorporate

"ground truth" data from the CARCO table. This addition changed none of the existing

functionality of the original model with the exception that it allowed mathematical

comparison between the two sets of data. The resulting functional diagram is presented
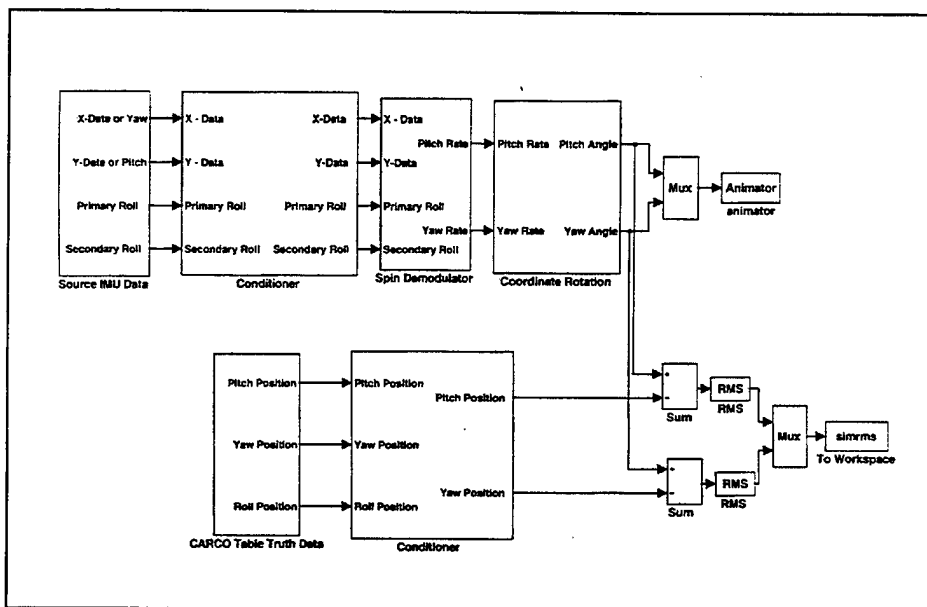
as Figure 12.

Figure 12. Validation Model.

When incorporated into the model, the CARCO table data is conditioned similarly

to the sensor data in order to remove bias, apply scale factors, match conventions

CARCO table conventions to industry conventions, and apply delays. Matching

conventions is necessary because the conventions of the CARCO table do not match the

industry convention. Therefore, what the sensors report as positive yaw may be referred

to as negative yaw by the table, despite the fact that both move in the same direction. In

this case, the CARCO table pitch position output was reversed in polarity. Additionally, a

transport delay of .025 seconds was applied to all CARCO table data to account for the delay imposed on the sensor data through processing and filtering in the model.

Comparison of the CARCO table truth data and model solution is performed by taking the arithmetic difference between the model solution and the "ground truth" data, and then, determining the RMS difference between the elements. This RMS difference is output to the workspace as a value indicating the difference in degrees which is used to quantify the accuracy of the model's representation of the missile's attitude.

IV. DATA COLLECTION

Data collection was performed using a flight motion simulator. The simulator, manufactured by CARCO Electronics Inc., is an electro-hydraulically operated table capable of exciting an installed object in five axes simultaneously. The CARCO table is also capable of outputting its exact alignment measured in angular position, throughout the test, providing ground truth against which model effectiveness can be measured.

The CARCO table is depicted in Figure 13. For testing purposes, the fourth and fifth axes were not used and have been removed from the figure.
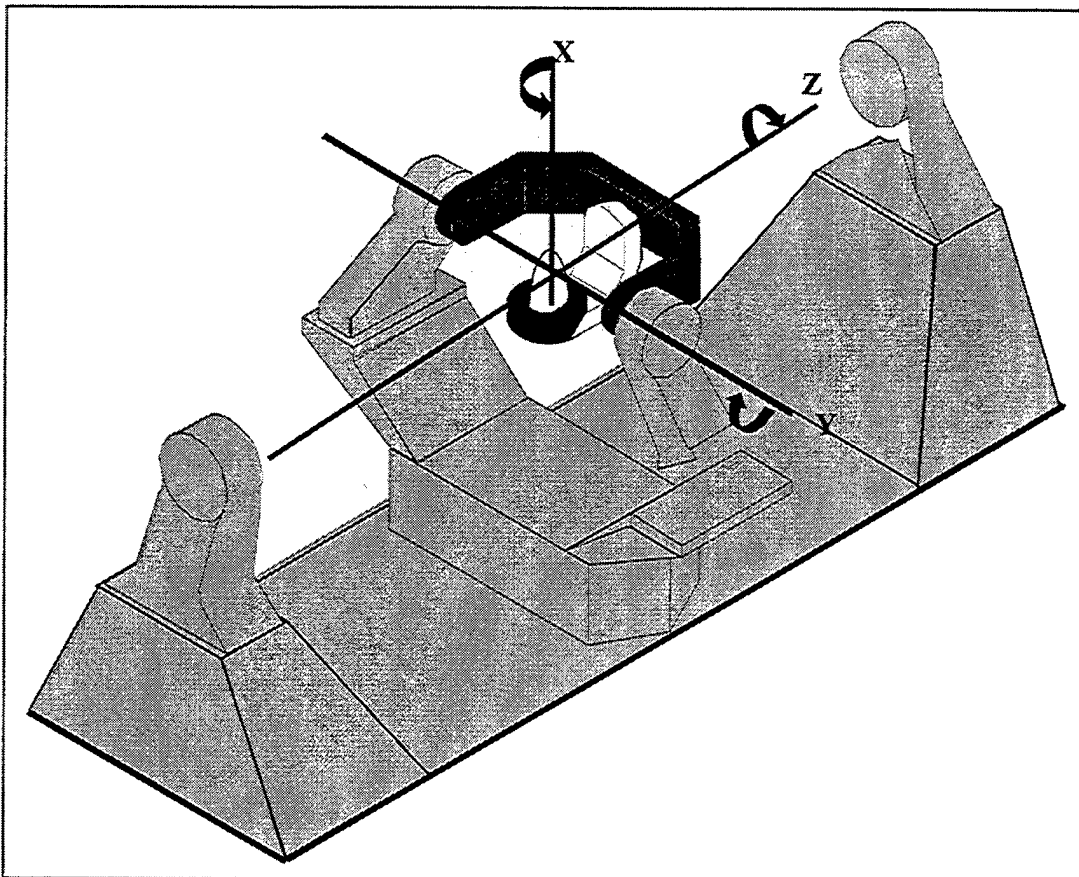

Figure 13. CARCO Table With Fourth And Fifth Degree Of Freedom Gimbals Removed.

Model validation required only a few runs of data however, evaluation of the sensors required runs of data for each sensor pair at varying roll rates and pitch and yaw rates. Because the ATA sensors had already been purchased in quantity, and because they were believed to be the superior sensor, the ATA sensors were tested much more extensively than the Tokin sensors.

The ATA sensor tests were designed to cover the full spectrum of rates under which a missile operates in flight. Previous testing with the ATA sensor in live fire operations indicated that the missile could experience roll rates from 8 to 18 Hz and rates in pitch and yaw from 0 to 200 degrees/second. Because of this, the ATA sensors were tested at spin rates of 5, 10, 15, and 18 Hz. At each of those roll rates, the sensors were tested at pitch and yaw rates of 20, 40, 60, 80, 100, 140, 180, and 220 degrees/second in single axis excitation, pitch or yaw, as well as in dual axis excitation, pitch and yaw. Additionally, the initial tests indicated that the scale factors for the ATA sensors may not be constant throughout the range of rates and therefore, a thorough understanding of the required scale factors was needed to evaluate the sensors.

Single axis excitation was necessary in order to determine cross coupling between the sensors. To do this, both axes were excited individually. Once a solution was determined for the individual axis, those scale factors could then applied in a dual axis excitation model.

Because of the Tokin sensors were not expected to be as effective as the ATA sensors, the Tokin sensor tests were limited. Like the ATA sensors, the Tokin sensors were tested at roll rates of 5, 10, 15 and 18 Hz. However, the Tokin sensors were only tested in pitch and yaw at the extremes, 40 and 220 degrees/second, and at the median

expected rate of 100 degrees/second. Like the ATA tests, the Tokin tests included single axis and dual axis excitation however, the single axis excitation tests were only conducted is the yaw axis. Additional tests on the Tokin sensor included rolling at 5, 10, 15, and 18 Hz with no excitation in order to observe the response of the sensors at these roll rates.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SENSOR EVALUATION

The ATA and Tokin rate sensors, were evaluated for constancy in scale factors, bias, hysteresis, and accuracy of response throughout the range of rates. These effects were easily observed and measured within the conditioner and the model.

A. APPLIED TECHNOLOGY ASSOCIATES RATE SENSOR

The Applied Technology Associates' (ATA) ARS-04E rate sensor is one of the two sensors used to provide the pitch and yaw rate data which is fundamental to determining the attitude of a missile in flight. The output of the ARS-04E must be extremely accurate over the range of operations of the missile and must have scale factors that are consistent throughout that range. The ARS-04E sensor was unable to demonstrate this ability.

The two principal defects detected in the ARS-04E sensor were both related to gain factors. First, the sensors required different scale factors when excited at different rates. These differences are exemplified in the 5Hz roll rate data. The scale factors required for varying pitch and yaw rates are presented in Table 4. This variation of the scale factors over the range of rates prevents accurate modeling of real live missile flight data that may experience numerous pitch and yaw rates during live fire operations.

Excitation Rate	Y-Rate Sensor Scale Factor	X-Rate Sensor Scale Factor
20 degrees/sec	8.6	8.5
100 degrees/sec	6.7	6.12
220 degrees/sec	6.63	5.63

Table 4. Scale Factors Required To Offset Variable Gain Requirements Exhibited In The ARS-04E Rate Sensor.

The second deficiency detected in the ARS-04E output was an effect called

hysteresis. Hysteresis is a variable gain effect in sinusoidal outputs in which the gain

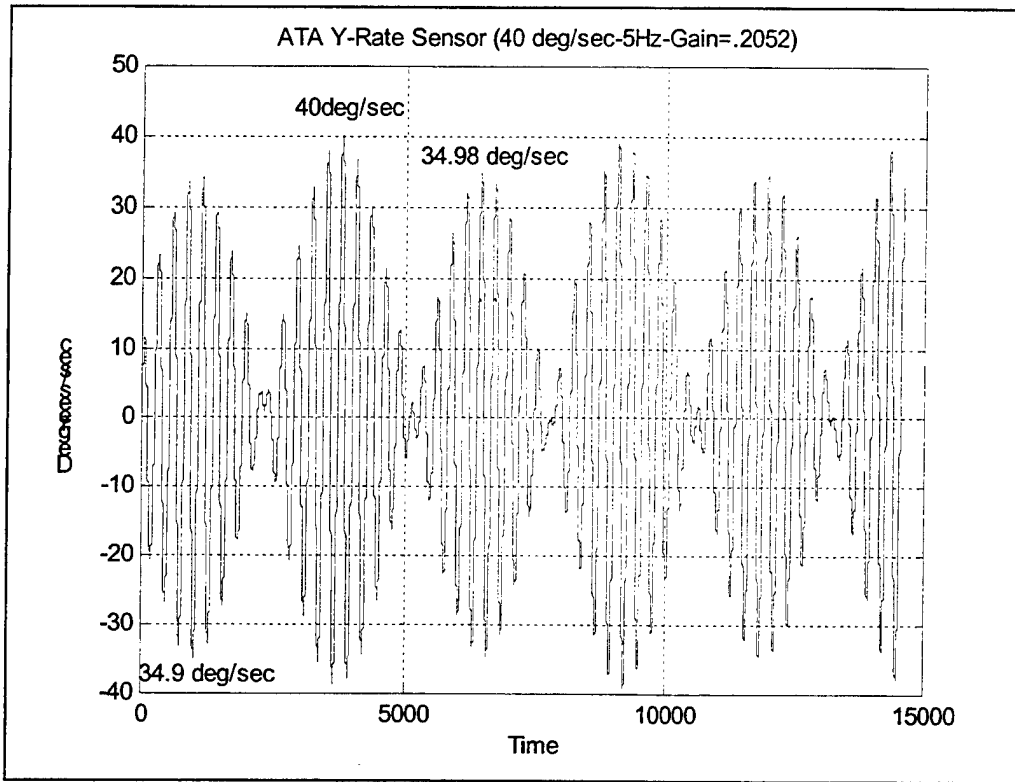factor changes based on the direction of motion. As demonstrated in Figures 14,

Figure 14. Effects Of Hysteresis In The ARS-04E Rate Sensor.

positive and negative rates of 40 degrees per second require different gains. As pictured,

the Y-Rate sensor, when excited at a rate of 40 degrees/second, exhibited a difference in

required gains of 13%.

Correcting for this deficiency requires time variable gains and *a priori* knowledge

of the rate being experienced. Because of this, correction in live fire operations is

impossible, making use of the ATA sensor impractical.

B. TOKIN AMERICA RATE SENSOR

The Tokin America CG-16D rate sensor performed extremely well and provided a
highly usable output. Despite the limitations implied by the CG-16D specification, the
sensor demonstrated the ability to accurately indicate pitch and yaw rates from 40 to 220
degrees/second with none of the deficiencies experienced by the ATA sensors.
Throughout the range of rates, no differences in scale factors were required.
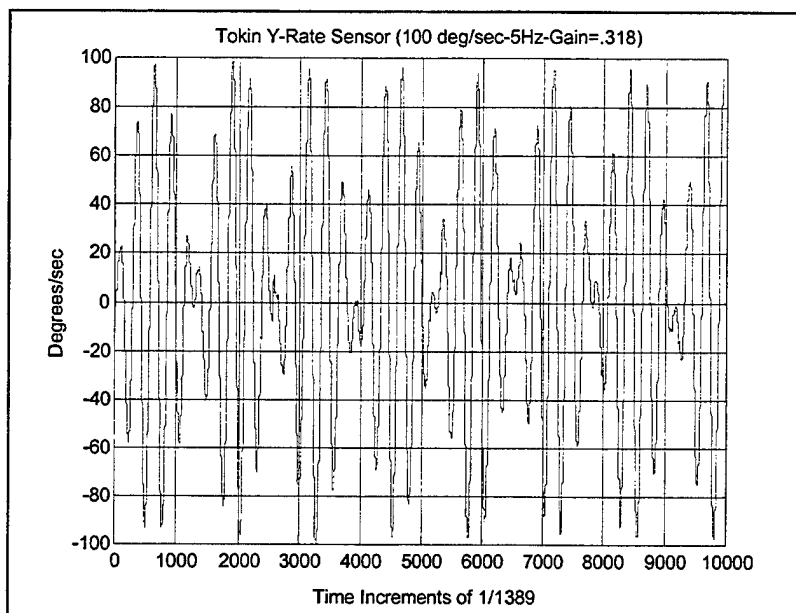Additionally, there is no indication of the hysteresis effect as demonstrated in Figure 15.

Figure 15. Output of CG-16D Sensor Demonstrating
Response With No Apparent Hysteresis.

33

THIS PAGE INTENTIONALLY LEFT BLANK

VI. MODEL EVALUATION

Model validation was conducted using an enhanced model that allowed the introduction of CARCO table "ground truth" data for comparison with the model solution. Once both sets of data were conditioned, a comparison was made to determine the ability of the model to effectively and accurately recreate the missile's attitude throughout the run. Effectiveness was determined by visually comparing the model output to ensure that the model solved position along the correct axis and with the correct polarity. Accuracy was determined by the root mean squared (RMS) method on the difference between the model output and the CARCO table output. Desired accuracy was prescribed as within 2 degrees RMS.

The model demonstrated that it was fully capable of recreating missile attitude well within the 2 degrees RMS requirement. This was confirmed using the 5 Hz data at 40, 100 and 220 degree/second single axis rates, and a dual axis sample at a rate of 100 degrees/second in the models presented in Appendix C. The results of the test are presented in Table 5.

	RMS Yaw Difference	RMS Pitch Difference
Single Axis 40 degrees/sec	.7643	.6115
Single Axis 100 degrees/sec	.7448	.6865
Single Axis 220 degrees/sec	.8360	.7017
Dual Axis 100 degrees/sec	1.6251	1.2917

Table 5. RMS Differences Between Model Solution And "Ground Truth."

Additionally, the phase locked loop and arctangent function methods for processing roll sensor output were compared for effectiveness and usability. Both methods performed well, faithfully reproducing a "clean" representation the phase of the roll sensor output.

The arctangent function performed the best, with no visible deviation when zero crossings were compared. The phase locked loop method did not perform as well. In our test, over 10 seconds at 5Hz, the phase locked loop failed to match 10 of the 100 zero crossings. Of these mismatches, the maximum deviation was .0032 seconds equating to 5.76 degrees of error. This error could not be fully corrected in the model. However, results for yaw only runs remained favorable. These results are presented in Table 6. The models for used in this application are presented in Appendix C.

	RMS Yaw Difference	RMS Pitch Difference
Single Axis 40 degrees/sec	.7075	1.225
Single Axis 100 degrees/sec	1.5656	1.0882
Single Axis 220 degrees/sec	1.0196	2.3958

Table 6. RMS Differences Between Model Solution And "Ground Truth" When Roll Sensor Used As Roll Reference.

In addition to its proven accuracy, the arctangent function method reduces the amount of user interface required. While the phase locked loop requires the user to specify the frequency and initial phase in order to lock to the sensor output, the

arctangent function method requires no input. These qualities make the arctangent

function method preferable to the phase locked loop.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS AND AREAS FOR FURTHER STUDY

The continuing threat of infrared guided munitions and the need to test and evaluate countermeasures against this threat has forced the Department of Defense to develop methods for accurately determining the response of a missile to countermeasures. TM Tracker provided an ability to determine time, space, and position information on a live fire missile, but does not provided attitude information. The use of micro-miniature sensor technologies has now made this possible. The size of these sensors permits their use in small diameter missiles while providing the data necessary to accurately reconstruct the missile's attitude throughout the flight. This capability is enhanced by the ability to provide near real-time analysis of that missile's attitude using a PC based model. Using the extensive and powerful algorithms of the MATLAB and SIMULINK environments, this model provides the user with the ability to analyze, process and display the effects of countermeasures on a missile.

A. SENSORS

The ability to model a roll-stabilized missile's attitude depends firmly on the ability of the sensors to provide accurate angular rate and roll data in a rugged environment in which rates of 200 degrees/second are feasible. Two angular rate sensors were evaluated for this application, the Applied Technology Associates ARS-04E and the Tokin America CG-16D. Of these sensors, the ATA sensor appeared to be the most promising, based on its specifications.

The ATA ARS-04E failed to perform as anticipated. The ATA sensor was apparently unable to provide the level of accuracy required for demodulation of the

angular rates experienced by the missile, due to non-linear gain requirements and hysteresis.

In contrast, the Tokin sensor performed much better than specified. Based on a limited maximum detectable angular rate, $\pm$ 90 degrees/second, the Tokin sensor was not specified to handle rates throughout a missile's range of operation, up to 200 degrees/second. In fact, the Tokin sensor performed well at rates of 220 degrees/second without any varying gain requirements or non-linear effects. Its ability to perform accurately and effectively throughout this range combined with its small size and solid state construction makes the Tokin sensor a viable option for use in testing roll-stabilized missiles.

Because of time limitations, extensive testing of the CG-16D was not performed. Further testing of the CG-16D should be conducted to further verify its effective and accurate operation in this environment and at the rates expected in live ire operations.

One roll sensor was evaluated for effectiveness and accuracy in this application, the Honeywell SSEC HMC1002. Like the ATA and Tokin sensors, the HMC1002 is small enough for inclusion in a missile's IMU package and its solid state construction suggested that it would be capable of effectively sensing and reporting roll angle. Further research and evaluation of micro-miniature roll sensors is necessary.

B. EULER ROTATION MODEL

The Euler rotation model exceeded project requirements. The MATLAB/SIMULINK environment provided the tools necessary to perform high level computations in order to determine the missile's attitude based on rate inputs. The ease of user interface and highly functional graphics capabilities provides the user with a tool

that can be easily reconfigured to provide the data required in near real-time with and accuracy and ease.

The Euler rotation model is limited only by the condition of Gimbal lock. The occurrence of Gimbal Lock is limited to operations in which the attitude of the missile is manipulated to an angle of 90 degrees and therefore should not reduce the effectiveness of this model in this application.

Proper conditioning of input data is vital to the correct operation of the Euler rotation model. Inaccuracies caused by bias, improper scale, and cross-coupling are accumulated throughout the model and result in inaccuracies in the Euler rotation model output. All three of these can be determined using hardware-in-the-loop testing and once determined can be effectively eliminated.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. MATLAB COMPUTER ANIMATION CODE

Listed below is the computer code written to animate the SIMULINK Euler

rotation model.

A. ANIMATOR.M

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by LT Troy Johnson, 1988.                               %
% This S-file is called by the Simulink Model                     %
% "Animate." Draw.m is called upon initialization and             %
% Redraw.m is called to redraw the aircraft each pass             %
% through the model.                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [sys,x0,str,ts] = Animator(t,x,u,flag)

BlockHandle=gcb;

% If working figure is present, get userdata from the object.
WorkingFig=get_param(BlockHandle,'UserData');
if
~ishandle(WorkingFig)|~strcmp(get(WorkingFig,'Tag'),'QuatWorkingFig'),
  WorkingFig=[];
  set_param(BlockHandle,'UserData',[]);
end  % if

switch flag,

  %%%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%%
  case 0,
      [sys,x0,str,ts]=mdlInitializeSizes(WorkingFig);

  %%%%%%%%%%%
  % Outputs %
  %%%%%%%%%%%
  case 3,
     sys=mdlOutputs(t,x,u,WorkingFig);

  %%%%%%%%%%%%%
  % Terminate %
  %%%%%%%%%%%%%
  case 9,
    sys=mdlTerminate(t,x,u,WorkingFig);

  case {1,2,4},
    % Don't do anything
```

```matlab
%%%%%%%%%%%%%%%%%%%%%
% Unexpected flags %
%%%%%%%%%%%%%%%%%%%%%%
otherwise
   error(['Unhandled flag = ',num2str(flag)]);

end


%=======================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for
% the S-function.
%=======================================================================

function [sys,x0,str,ts]=mdlInitializeSizes(WorkingFig)

draw(WorkingFig);                % This will draw the page

sizes = simsizes;

sizes.NumContStates  = 0 ;
sizes.NumDiscStates  = 0 ;
sizes.NumOutputs     = 0 ;
sizes.NumInputs      = 3 ;
sizes.DirFeedthrough = 0 ;
sizes.NumSampleTimes = 1 ;   % at least one sample time is needed

sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [20/1389 0];


%=======================================================================
=======
% mdlOutputs
% Return the block outputs.
%=======================================================================
=======

function sys=mdlOutputs(t,x,u,WorkingFig)


sys=[];

psi=u(1);theta=u(2);phi=u(3);

%perform this function only if the figure is present
if findall(0,'Type','figure','Tag','QuatWorkingFig')
   redraw(WorkingFig,psi,theta,phi);
end


%=======================================================================
% mdlTerminate
% Perform any end of simulation tasks.
%=======================================================================
```

```
function sys=mdlTerminate(t,x,u,WorkingFig)

sys=[];
```

B. DRAW.M

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by LT Troy Johnsion 1998                        %
% Modifed by LT Craig Hill to include pause               %
% and restart functions.                                  %
% Draw.m is called by Animator.m when the s-file          %
% block is invoked.                                       %
% Draws the figure, axes, and aircraft.                   %
% Both h1 lines must be altered to indicate the           %
% name of the model which is providing input.             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function draw(WorkingFig)

%check if figure is already on screen
[flag,fig] = figflag('Missile Attitude');

%if figure is already up, do nothing and return to Display
if flag
    return;
end


%%% General Info.
Black       =[0      0      0      ]/255;
White       =[255    255    255    ]/255;
UIBackColor=get(0,'DefaultUIControlBackgroundColor');
FigColor=UIBackColor;
UIForeColor=Black;

%%% Set Positions
ScreenUnits=get(0,'Units');
set(0,'Units','pixels');
ScreenSize=get(0,'ScreenSize');
set(0,'Units',ScreenUnits);

FigWidth=750;
FigHeight=530;
FigPos=[(ScreenSize(3:4)-[FigWidth FigHeight])/2 FigWidth FigHeight];


%%% Create InputFig
QuatFig=figure('BackingStore'   ,'off'                 , ...
               'Color'          , FigColor             , ...
               'Name'           ,'Missile Attitude'    , ...
               'NumberTitle'    ,'off'                 , ...
               'Pointer'        ,'arrow'               , ...
               'Position'       , FigPos               , ...
               'Renderer'       ,'zbuffer'             , ...
```

45

```
                'Tag'                ,'QuatWorkingFig'              , ...
                'IntegerHandle'  ,'off'                            , ...
                'Visible'            ,'off'                         );

%%% Create pushbuttons for Pause and Continue

h1 = uicontrol( 'Parent',QuatFig, ...
                'Units','points', ...
                'Callback','set_param(''CurrentModelroll'',
                    ''Simulationcommand'',''pause'')', ...
                'ListboxTop',0, ...
                'Position',[483 346.5 39.75 22.5], ...
                'String','Pause', ...
                'Tag','Pushbutton1');

h1 = uicontrol('Parent',QuatFig, ...
                'Units','points', ...
                'Callback','set_param(''Current model'' ,
                ''Simulationcommand'',''continue'')', ...
                'ListboxTop',0, ...
                'Position',[483 316.5 39.75 22.5], ...
                'String','Continue', ...
                'Tag','Pushbutton2');

%%% Create axes

QuatAxes=axes('Tag'                  ,'Quaternion Axes', ...
              'Units'                ,'pixels'          , ...
              'DataAspectRatio'      ,[1 1 1]           , ...
              'PlotboxAspectRatio',[1 1 1]              , ...
              'View'                 ,[60 10  ]         , ...
              'Box'                  ,'on'              , ...
              'Color'                ,Black             , ...
              'XColor'               ,White             , ...
              'YColor'               ,White             , ...
              'ZColor'               ,White             , ...
              'DrawMode'             ,'fast'            , ...
              'Projection'           ,'perspective'     , ...
              'XLim'                 ,[-100 100]        , ...
              'XTick'                ,[]                , ...
              'YLim'                 ,[-100 100]        , ...
              'YTick'                ,[]                , ...
              'ZLim'                 ,[-100 100]        , ...
              'ZTick'                ,[]                , ...
              'Visible'              ,'on'              );


set([QuatFig,QuatAxes],'HandleVisibility','on');
figure(QuatFig)
rotate3d on;


%Create arrows and plane


ArrowLineX=[ 0 90   80 90 80
             0  0    0  0  0
```

```matlab
                        0   0  -5   0   5
               ]';
ArrowLineY=[ 0   0  -5   0   5
               0  90  80  90  80
               0   0   0   0   0
            ]';
ArrowLineZ=[ 0   0   0   0   0
               0   0  -5   0   5
               0  90  80  90  80
            ]';


for lp=1:3,
   LineHandles(lp,1)=line('XData'    ,ArrowLineY(:,lp) , ...
                          'Ydata'    ,ArrowLineX(:,lp) , ...
                          'ZData'    ,-ArrowLineZ(:,lp), ...
                          'Color'    ,[1 1 1]          , ...
                          'Parent'   ,QuatAxes         , ...
                          'LineWidth',1                , ...
                          'Visible'  ,'on'               ...
                          );
end % for lp
LineText(1)=text(0,100,0,'North','Color',[1 1 1],'Parent',QuatAxes);
LineText(2)=text(92,0,0,'East','Color',[1 1 1],'Parent',QuatAxes);
LineText(3)=text(0,0,-100,'Down','Color',[1 1 1],'Parent',QuatAxes);

PointerHandle=line('XData'    ,0          , ...
                   'YData'    ,0          , ...
                   'ZData'    ,0          , ...
                   'Color'    ,[0 0 1]    , ...
                   'Parent'   ,QuatAxes,  ...
                   'LineWidth',2          , ...
                   'UserData' , ...
                   [ArrowLineX(:,3)';ArrowLineY(:,3)';ArrowLineZ(:,3)']
                   );


PlaneX=[75   40
         0    0
         0    0
       ];
PlaneY=[ 0    0
        30    0
       -30    0
       ];
PlaneZ=[ 0    0
         0    0
         0  -20
       ];


for lp=1:size(PlaneX,2),
  PlaneHandles(lp)=patch(PlaneY(:,lp),PlaneX(:,lp),-PlaneZ(:,lp),[1 0
0], ...
                         'LineWidth',1        , ...
                         'Parent'   ,QuatAxes, ...
                         'EdgeColor',[0 0 0] , ...
                         'EraseMode','normal'   ...
```

47

```
                              );
  set(PlaneHandles(lp), ...
     'UserData',[PlaneX(:,lp)';PlaneY(:,lp)';PlaneZ(:,lp)'] ...
     )
end % for lp

set(QuatFig,'UserData',[PlaneHandles QuatAxes]);


WorkingFig=QuatFig;

set_param(gcb,'UserData',QuatFig);
```

C. REDRAW.M

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by Troy Johnson 1988.                         %
% Redraw.m is called by Animator on each pass through   %
% the model.  It redraws the aircraft object by         %
% rotating each vertex of both of the triangles         %
% that compromise the aircraft.                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function redraw(WorkingFig,psi,theta,phi);

QuatFig=get_param(gcb,'UserData');
data=get(QuatFig,'UserData');
PlaneHandles=data(1:2);
QuatAxes=data(3);


%convert to radians
psi=psi*pi/180;
theta=theta*pi/180;
phi=phi*pi/180;

%transform factors

%points one and two
p1= cos(theta)*cos(psi);
p2= cos(phi)*sin(psi)*cos(theta)-sin(phi)*sin(theta);
p3= cos(theta)*sin(psi)*sin(phi)-sin(theta)*cos(phi);

%points three and four
p4=-cos(theta)*sin(psi)*cos(phi)+sin(theta)*sin(phi);
p5= cos(phi)*cos(psi);
p6=cos(theta)*sin(phi)+sin(theta)*sin(psi)*cos(phi);

%point five
p7= sin(theta)*cos(psi);
p8=-sin(phi)*cos(theta)+cos(phi)*sin(psi)*sin(theta);
p9= cos(theta)*cos(phi)+sin(theta)*sin(psi)*sin(phi);
```

```
%define vertex values
x01=75; x02=40; y0=30; z0=-20;

%rotate point one (plane nose)
xn1=p1*x01; yn1=p2*x01; zn1=p3*x01;

%rotate point two (forward vertex of tail)
ax=x02/x01;
xn2=ax*xn1; yn2=ax*yn1; zn2=ax*zn1;

%rotate point three (right wing tip)
xn3=p4*y0; yn3=p5*y0; zn3=p6*y0;

%rotate point four (left wing tip)
xn4=-xn3; yn4=-yn3; zn4=-zn3;

%rotate point five (top of tail)
xn5=p7*z0; yn5=p8*z0; zn5=p9*z0;

%rotate point six (plane exhaust point)
xn6=0; yn6=0; zn6=0;

%rearrange vertex cartesian coordinates in patch command format
PlaneX=[xn1 xn2
        xn3 xn5
        xn4 xn6];
PlaneY=[yn1 yn2
        yn3 yn5
        yn4 yn6];
PlaneZ=[zn1 zn2
        zn3 zn5
        zn4 zn6];


%draw the figure
for lp=1:size(PlaneX,2),
  set(PlaneHandles(lp),'vertices',[PlaneY(:,lp),PlaneX(:,lp),-
PlaneZ(:,lp)])
 end % for lp

set(QuatFig,'UserData',[PlaneHandles QuatAxes]);

WorkingFig=QuatFig;

set_param(gcb,'UserData',QuatFig);
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. CONDITIONERS

Presented below are the conditioners required to correct deficiencies in the data

for each data sample. The conditioner serves to remove bias from and apply scale factors

to the sensor output data as well as to ensure that the data conventions are correct for

solution in the Euler rotation model.
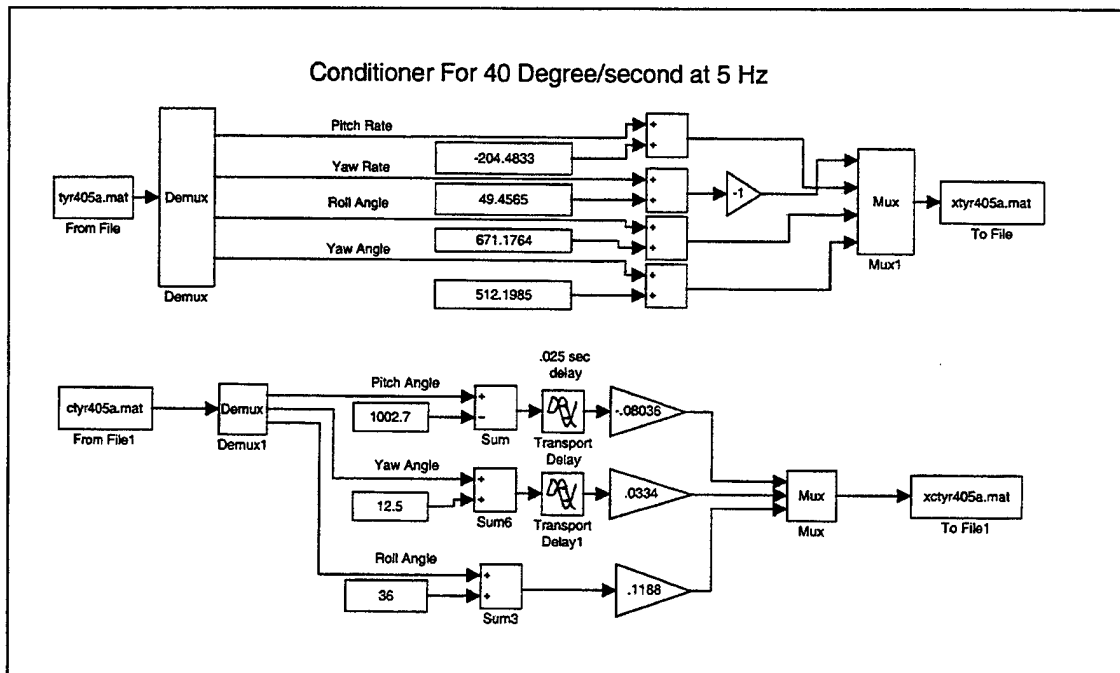
Figure 16. Conditioner for 40 Degrees/Second 5 Hz Data. Model TYR405B.MDL.
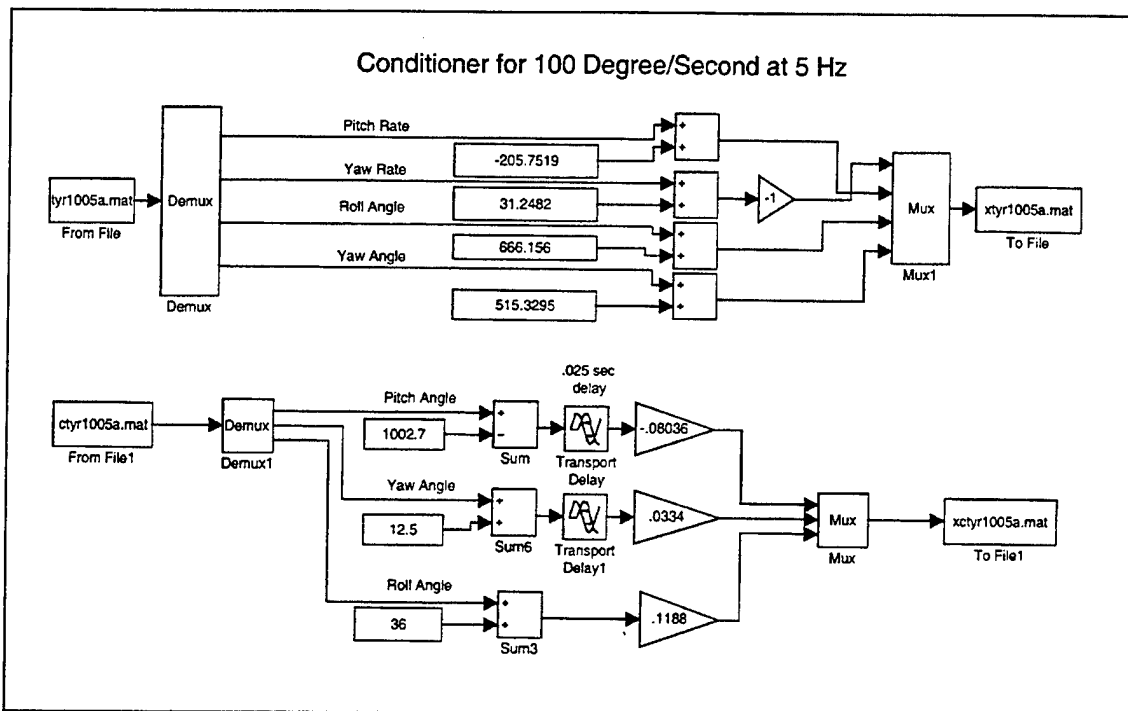
Figure 17. Conditioner For 100 Degrees/Second 5Hz Data. Model TYR1005B.MDL.
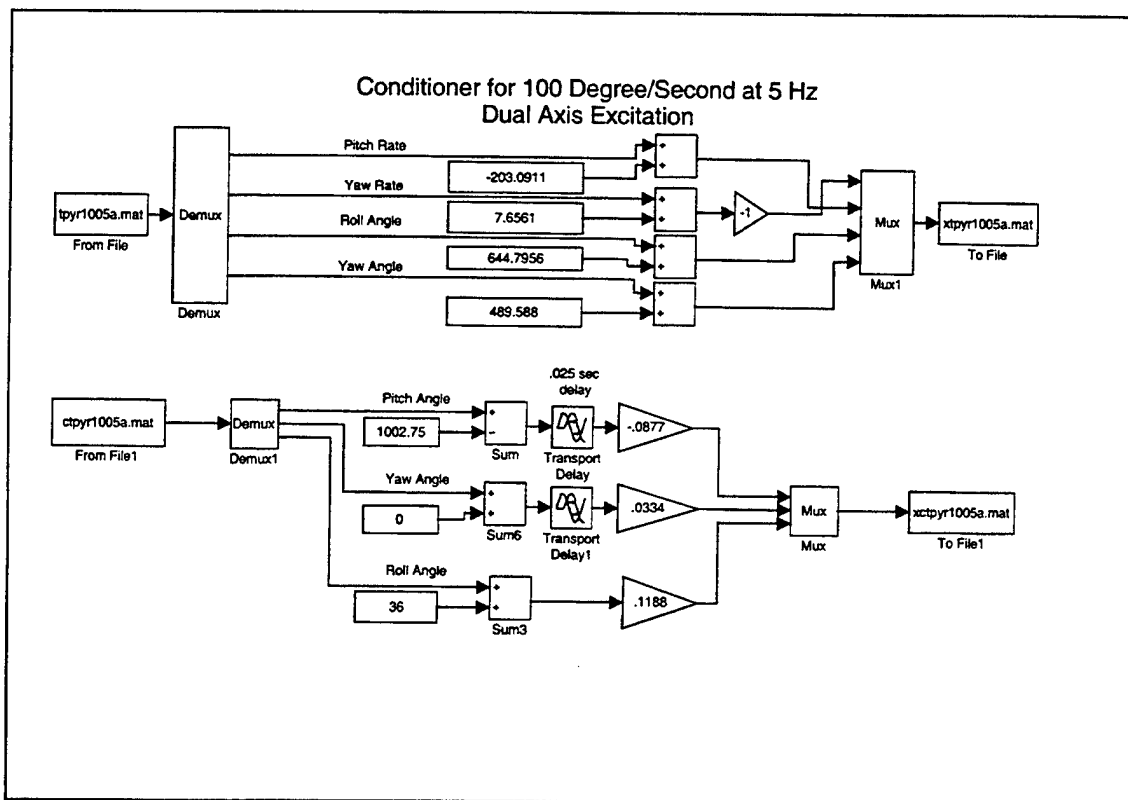
Figure 18. Conditioner For 100 Degrees/Second 5Hz, Dual Axes Excitation Data. Model TPYR1005B.MDL.

52
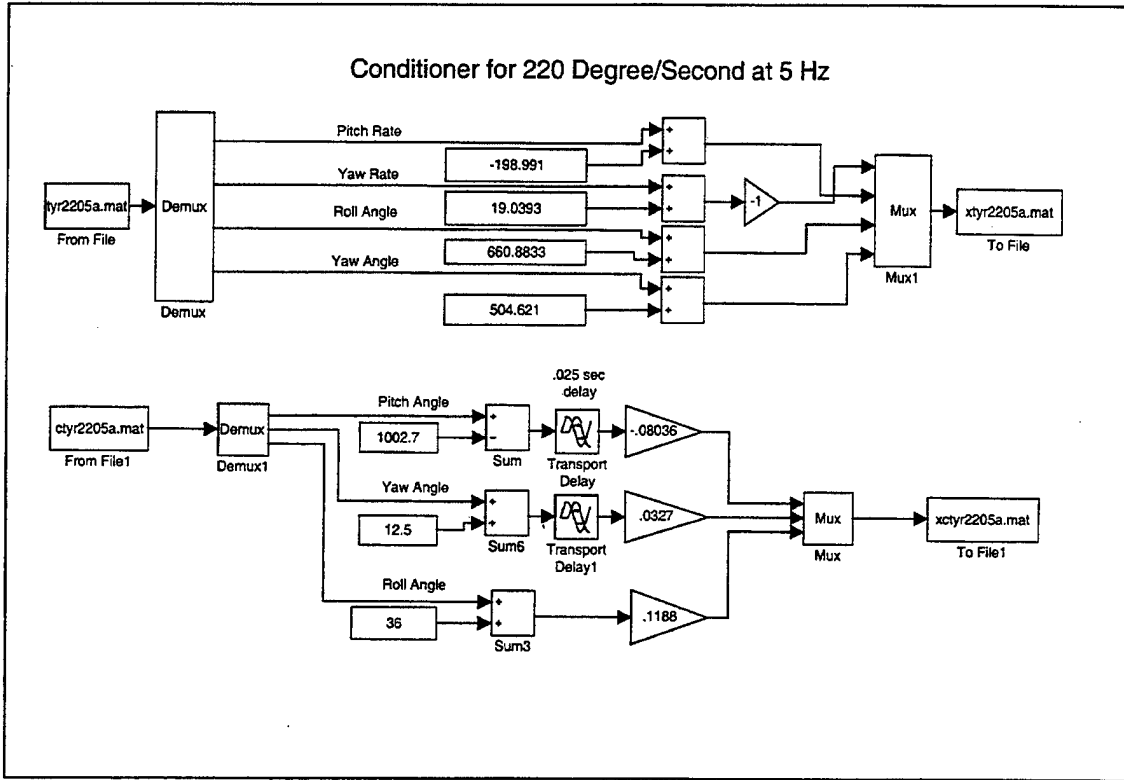
Figure 19. Conditioner For 220 Degrees/Second 5Hz Data. Model TYR2205B.MDL.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. EULER ROTATION MODELS

A. ROLL SENSOR ROLL REFERENCE

Presented below in Figures 20, 21, and 22 are the Euler rotation models for using

roll sensor output as roll reference. Figures 23 and 24 represent the demodulation and

coordinate rotation subsystems, respectively.
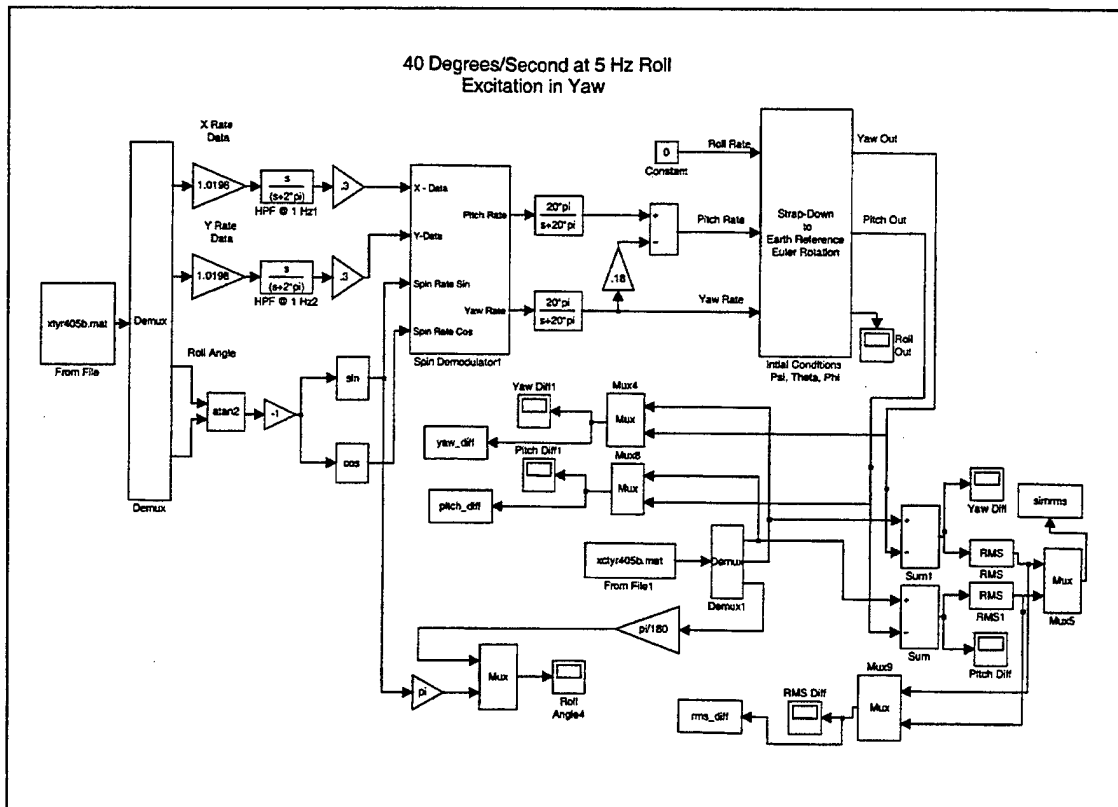
Figure 20. Euler Rotation Model For 40 Degrees/Second 5Hz Data With Roll
Reference Provided From The Roll Sensor. Model RMPTYR405B.MDL.
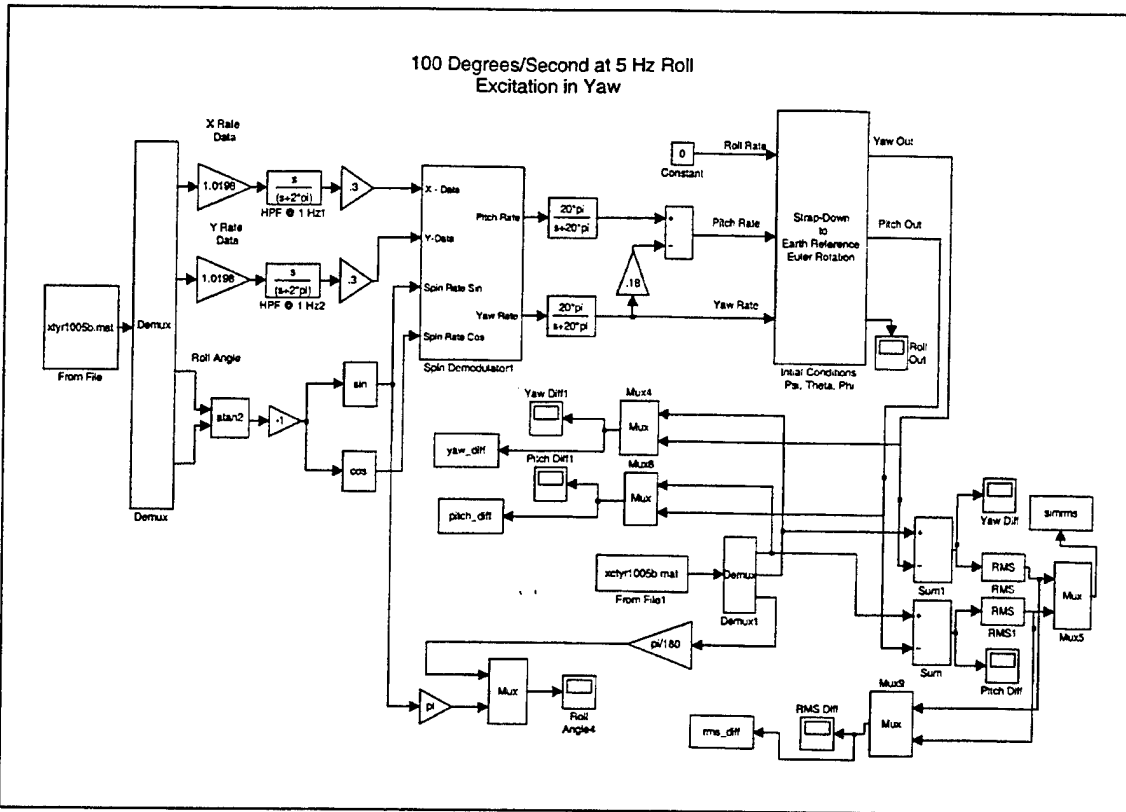
Figure 21. Euler Rotation Model For 100 Degrees/Second 5Hz Data With Roll
Reference Provided From The Roll Sensor. Model RMPTYR1005B.MDL.
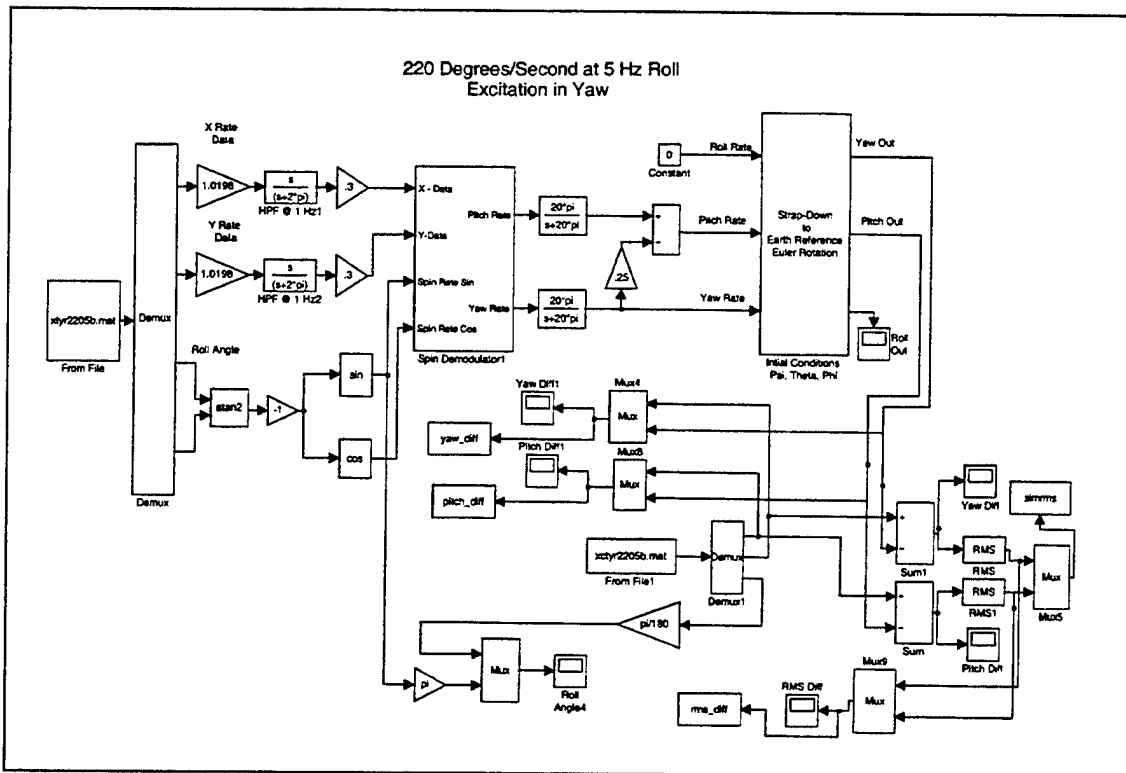
Figure 22. Euler Rotation Model For 220 Degrees/Second 5Hz Data With Roll
Reference Provided From The Roll Sensor. Model RMPTYR2205B.MDL.
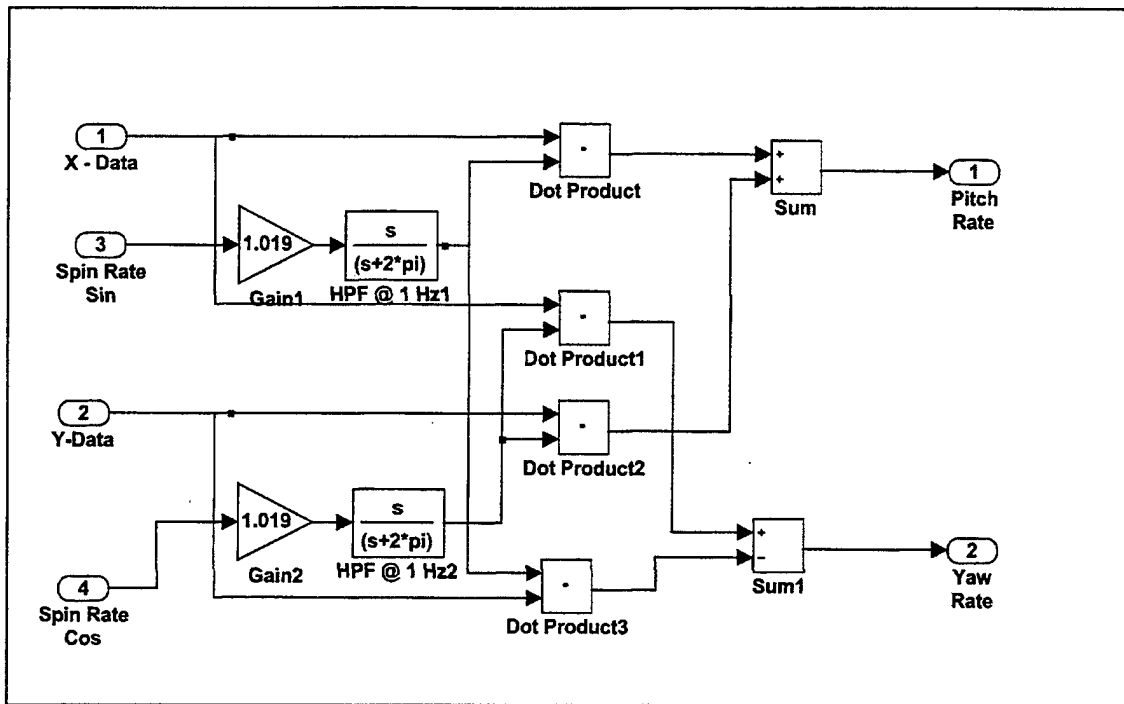
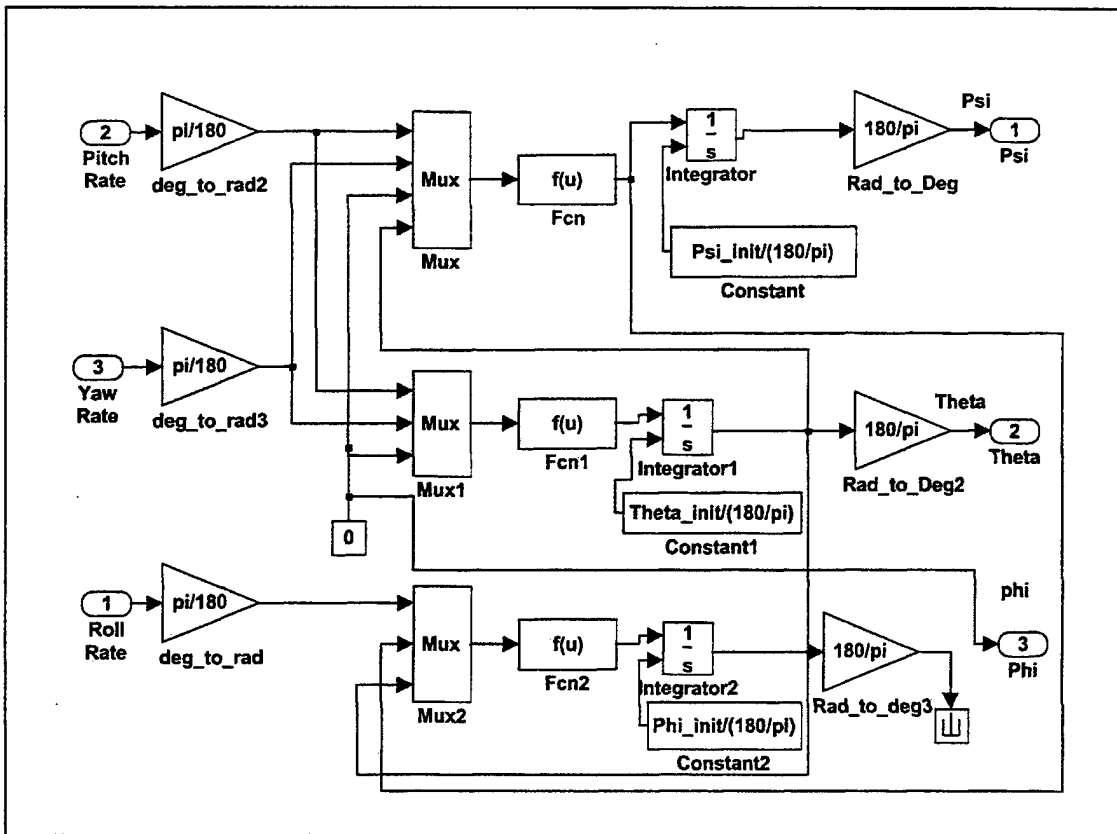Figure 23. Demodulator Subsystem.

Figure 24. Coordinate Rotation Subsystem.

B. CARCO TABLE ROLL REFERENCE

Presented below in Figures 25, 26, 27, and 28 are the Euler rotation models

implemented using the CARCO table roll reference. Figures 29 and 30 are the

demodulation subsystem and coordinate rotation subsystems respectively. Figures 31,

32, and 33 are the phase locked loop and its principal components, the charge-pump
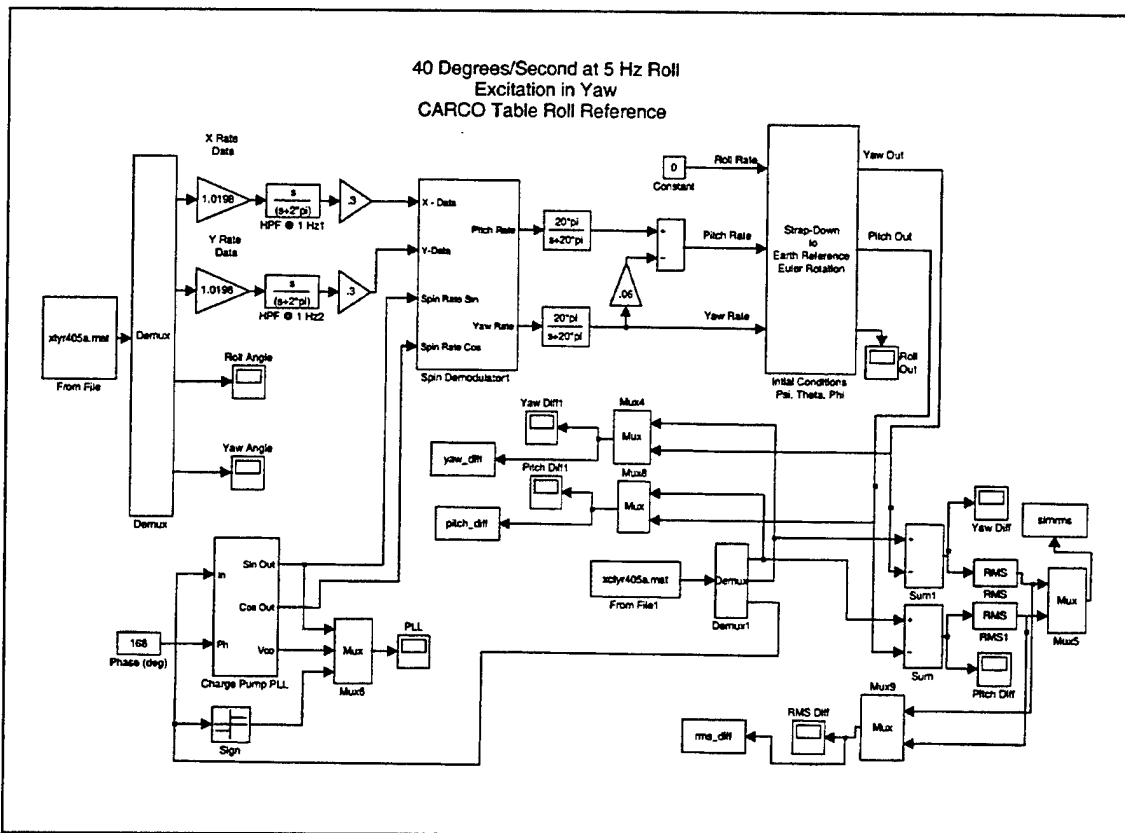
phase detector, and the voltage-controlled oscillator.

Figure 25. Euler Rotation Model For 40 Degrees/Second 5Hz Data With Roll
Reference Provided From The CARCO Table. Initial Phase Of The Voltage-Controlled
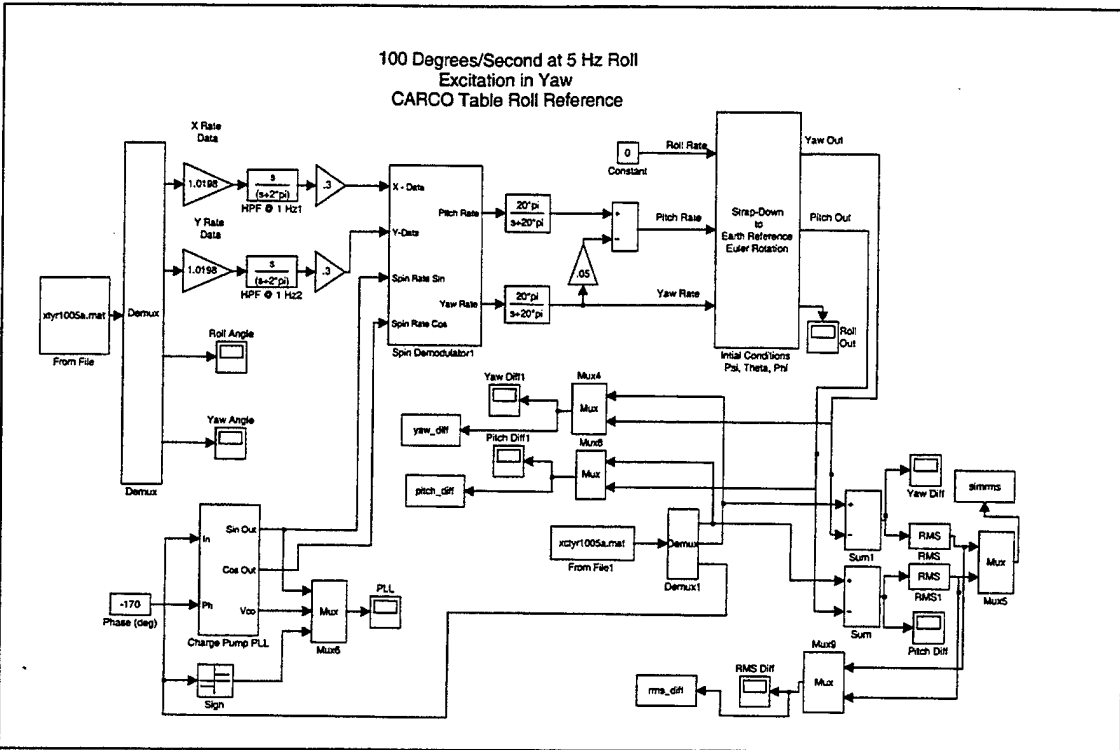Oscillator Is Set To −12 Degrees. Model RMPTYR405B.MDL.

Figure 26. Euler Rotation Model For 100 Degrees/Second 5Hz Data With Roll
Reference Provided From The CARCO Table. Initial Phase Of The Voltage-Controlled
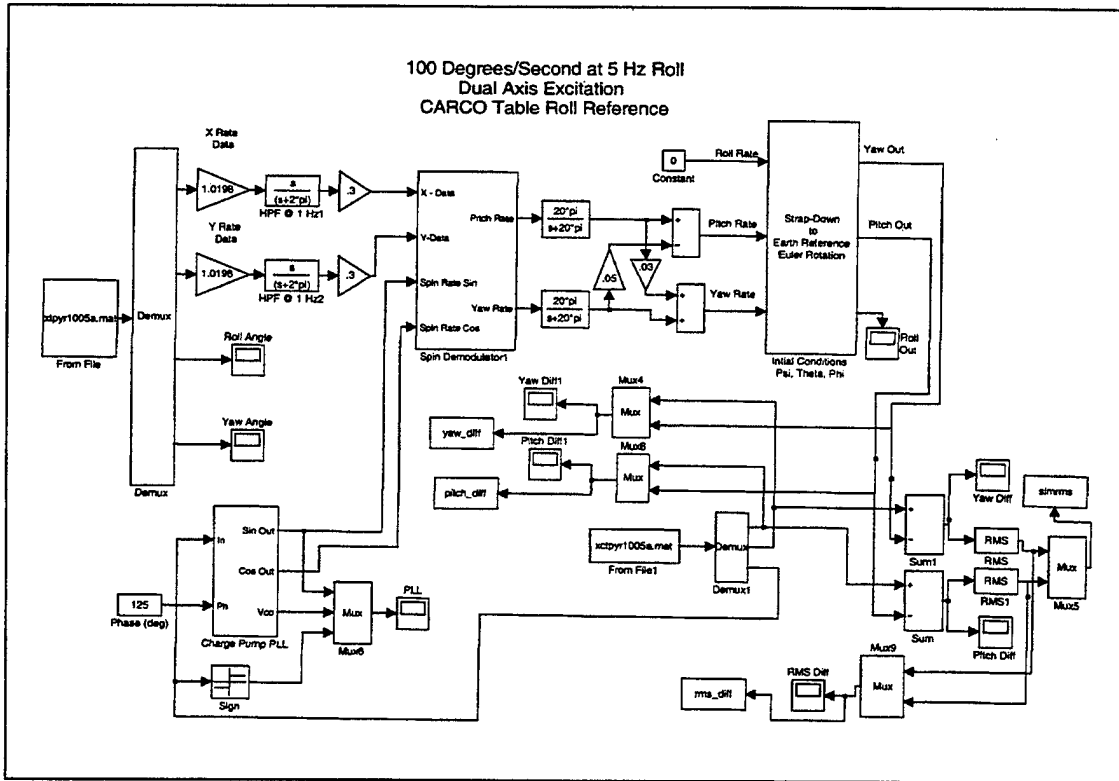Oscillator Is Set To 10 Degrees. Model RMPTYR1005B.MDL.

Figure 27. Euler Rotation Model For 100 Degrees/Second 5Hz Data With Dual Axis
Excitation And Roll Reference Provided From The CARCO Table. Initial Phase Of The
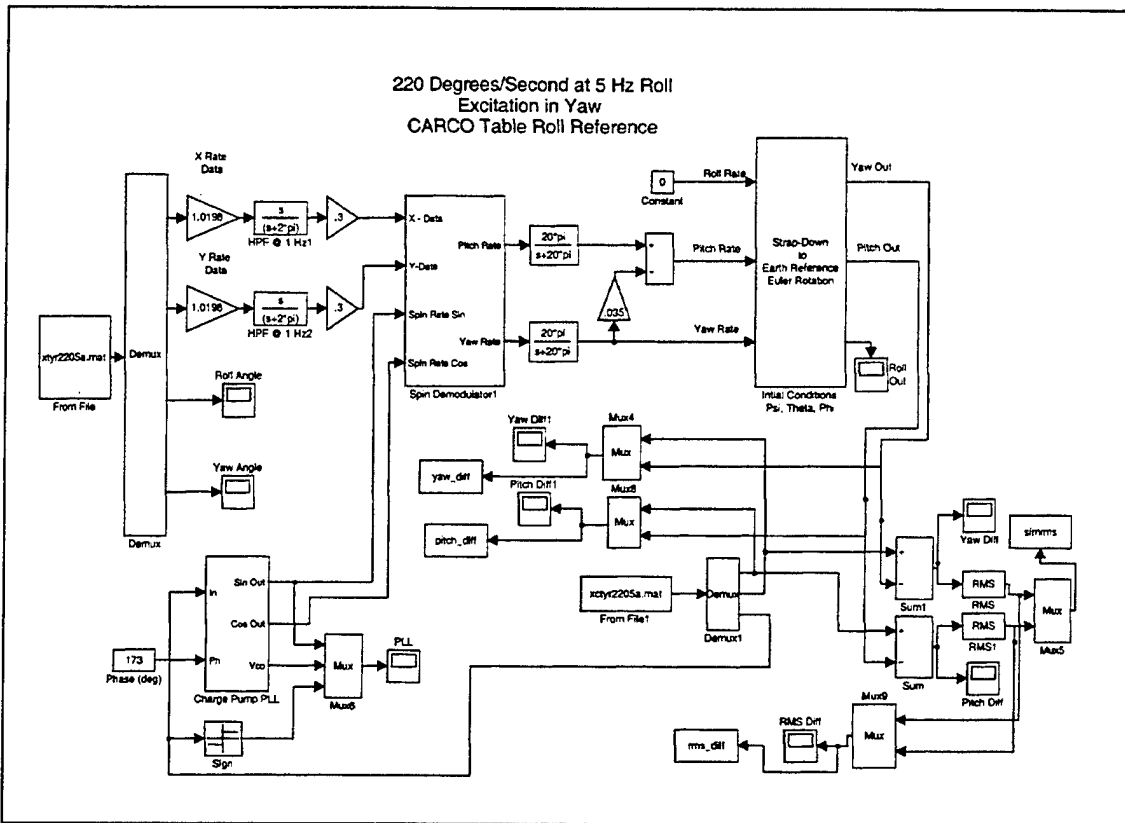Voltage-Controlled Oscillator Is Set To –55 Degrees. Model RMPTPYR1005B.MDL.

59

Figure 28. Euler Rotation Model For 220 Degrees/Second 5Hz Data With Roll Reference Provided From The CARCO Table. Initial Phase Of The Voltage-Controlled Oscillator Is Set To –7 Degrees. Model RMPTYR2205B.MDL.
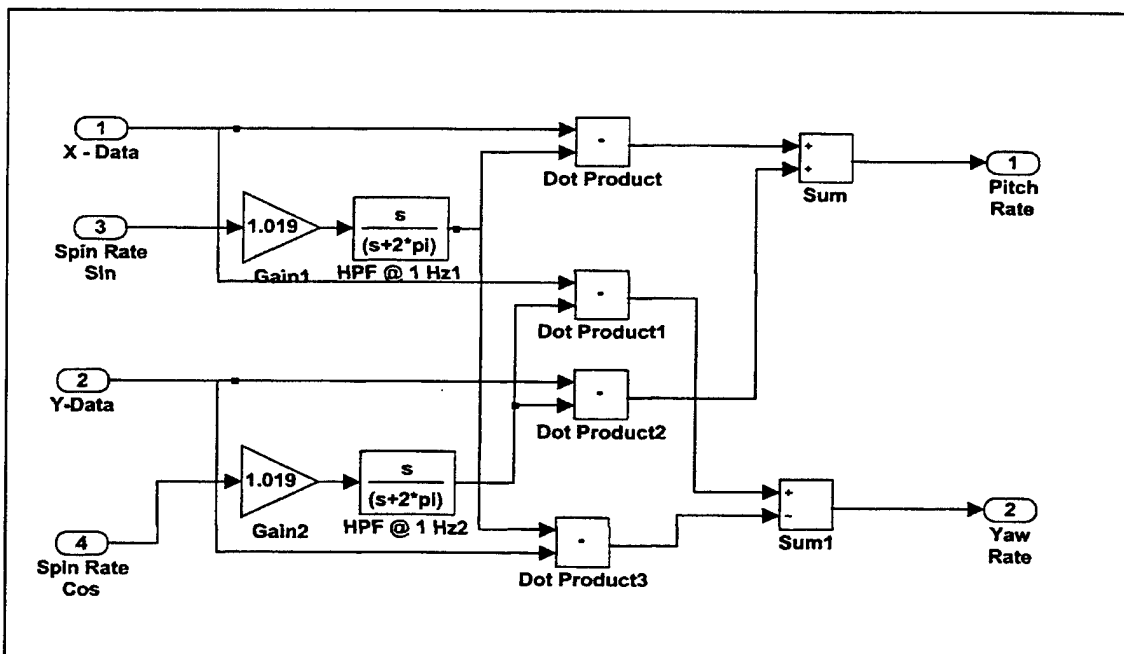

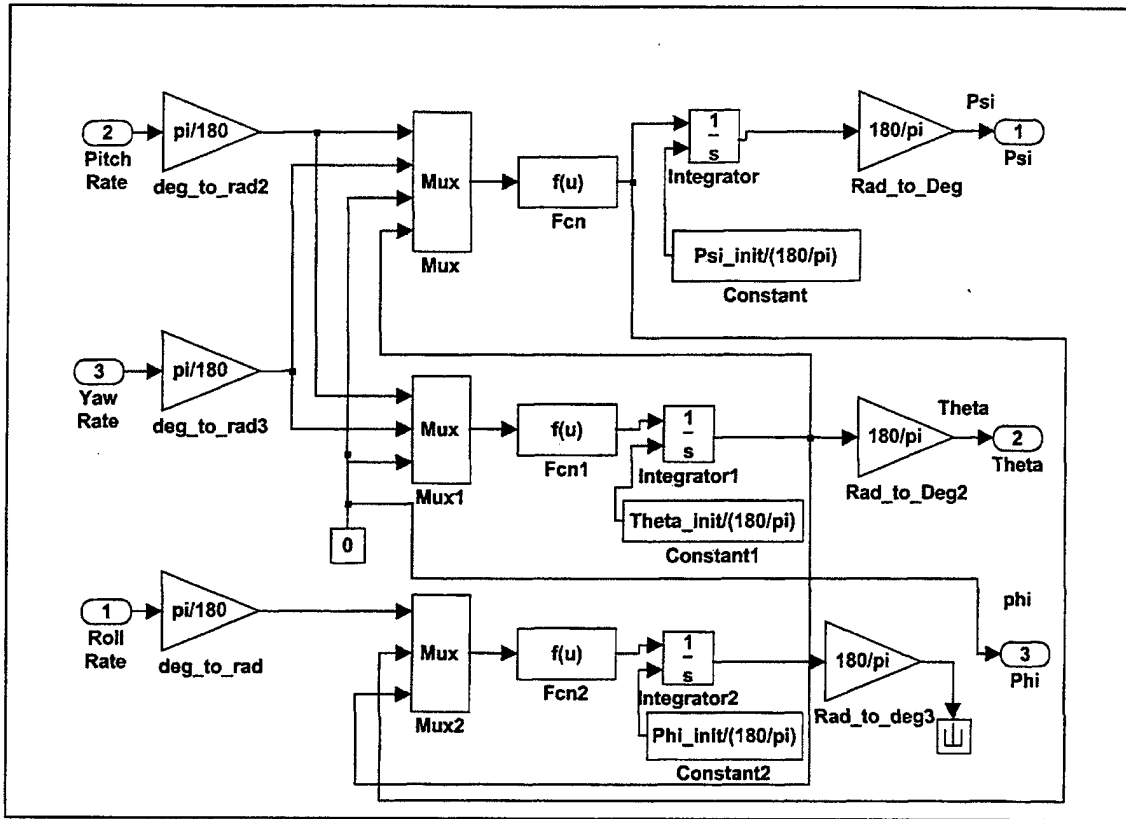Figure 29. Demodulation Subsystem.

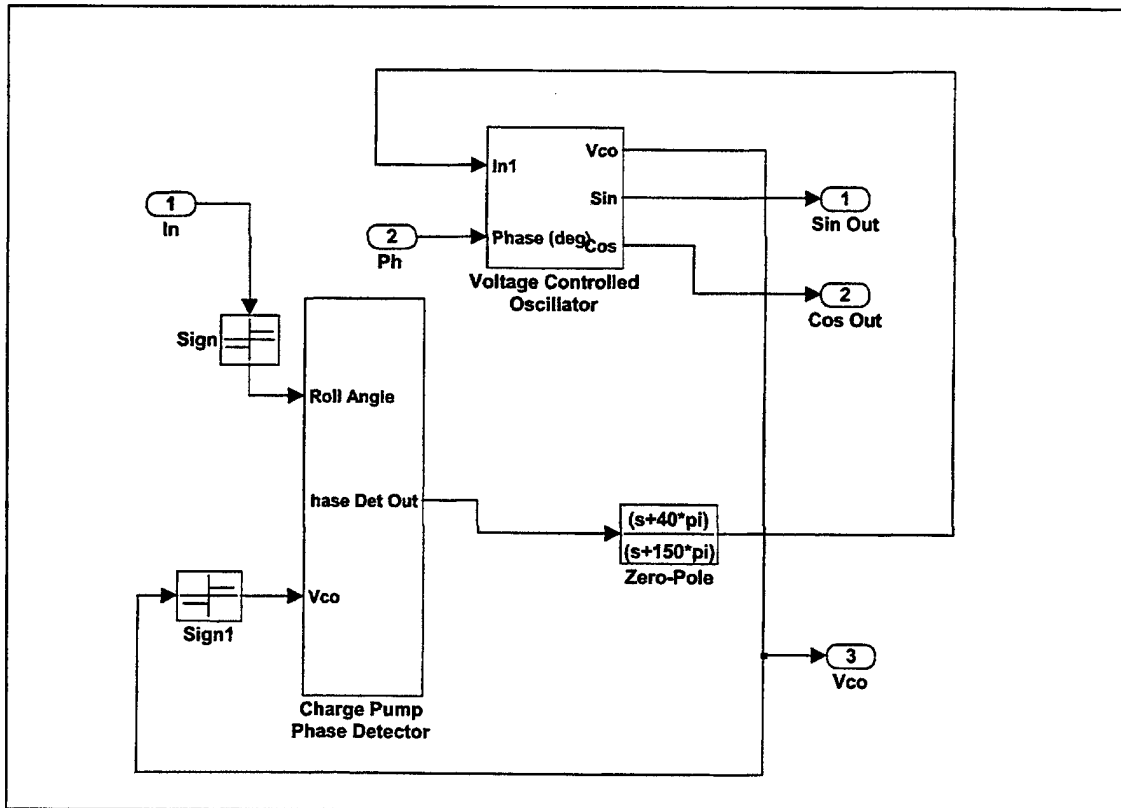Figure 30. Coordinate Rotation Subsystem.
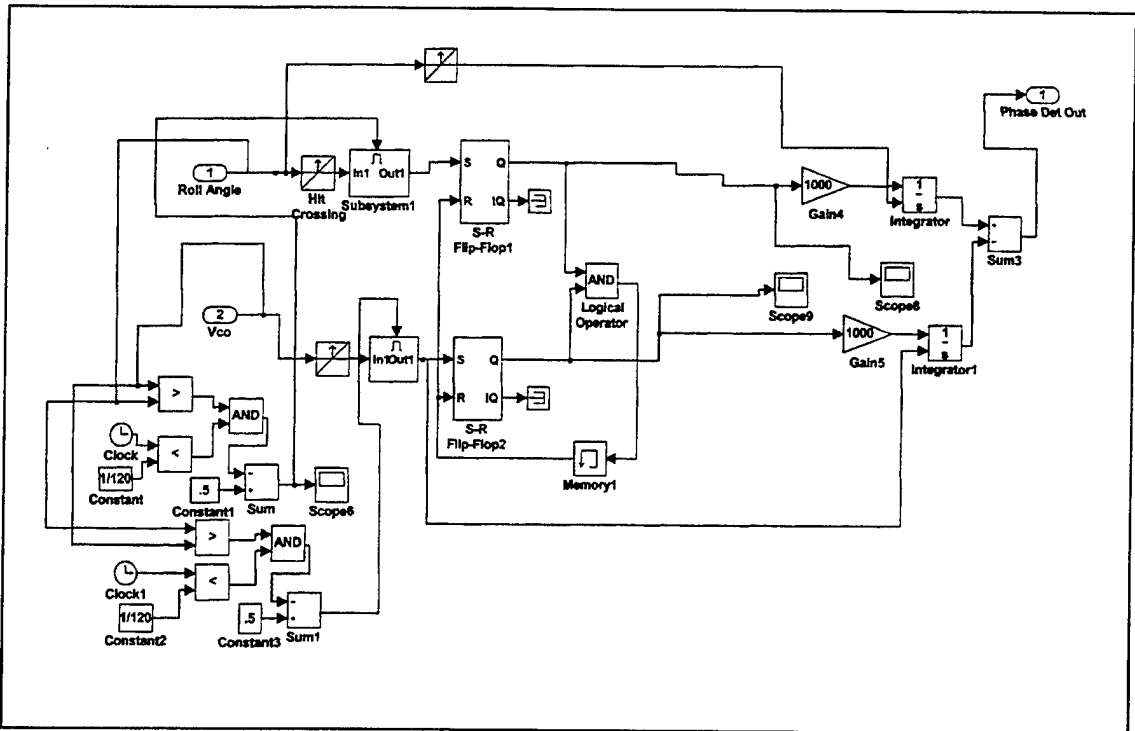

Figure 31. Phase Locked Loop.
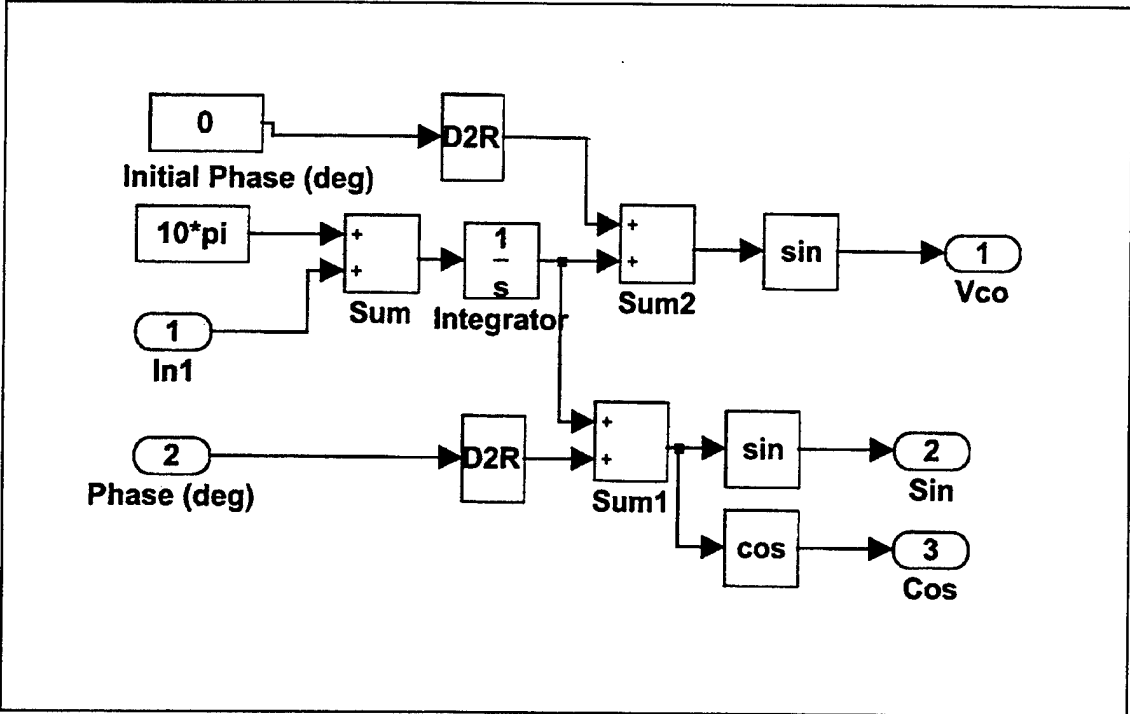
Figure 32. Charge Pump Phase Detector.

Figure 33. Voltage-Controlled Oscillator.

LIST OF REFERENCES

Applied Technology Associates, *ARS-04E & 04R MHD Angular Rate Sensor*, Data Sheet, 1999.

Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, 2d ed., Wiley-Interscience, 1991.

Bobick, N., "Rotating Objects Using Quaternions," *Game Developer*, Vol. II, July 1998.

Eichblatt, E. J., *Test and Evaluation of the Tactical Missile*, Vol. 119 of *Progress in Astronautics and Aeronautics*, American Institute of Aeronautics and Astronautics, 1989.

Greenwood, D. T., *Principles of Dynamics*, 2d ed., Prentice Hall, 1988.

Honeywell, Solid State Electronics Center, *Magnetic Sensor Products: HMC/HMR Series*, Data Sheet, 1996.

Johnson, T. M., *Computer Modeling of Jamming Effects on Infrared Missiles*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1999.

Naval Air Systems Command, *Electro-Optical/Infrared (EO/IR) Countermeasures (CM) Handbook*, Government Printing Office, Washington, D.C., July 1998.

Naval Air Warfare Center-Weapons Division, *TM Tracker Final Report*, China Lake, California, 2000.

Texas Instruments Inc., *Signal Conditioning Piezoelectric Sensors*, Application Report by J. Karki, 1999.

Tokin America Inc., *Ceramic Gyros*, Data Sheet, 2000.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Professor Curtis D. Schleher 1
   Code IW/Sc
   Naval Postgraduate School
   Monterey, California 93943-5101

4. Professor David C. Jenn 1
   Code EC/Jn
   Naval Postgraduate School
   Monterey, California 93943-5101

5. Professor Dan C. Boger 1
   Chairman, Code IW
   Naval Postgraduate School
   Monterey, California 93943-5101

6. Commanding Officer 2
   Code 473400D Bldg. 31440
   Attn: Mr. Larry Rollingson/Mr. Greg Velicer
   Naval Air Weapons Center
   China Lake, California
   93555

7. Craig A. Hill 2
   2155 Reynard Place
   Merritt Island, Florida 32952