# NAVAL POSTGRADUATE SCHOOL
## Monterey, California




# THESIS

DESIGN OF A SENSOR-BLENDING KALMAN FILTER FOR THE R2P2 FINE-TRACKING SYSTEM

by

Terrance J. Bauer

March 1990

Thesis Advisor: Harold A. Titus

## REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

| | |
|---|---|
| 1a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b RESTRICTIVE MARKINGS |
| 2a SECURITY CLASSIFICATION AUTHORITY | 3 DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited |
| 2b DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S) | 5 MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School<br>6b OFFICE SYMBOL (If applicable)<br>39 | 7a NAME OF MONITORING ORGANIZATION<br>Naval Postgraduate School |
| 6c ADDRESS (City, State, and ZIP Code)<br>Monterey, California 93943-5000 | 7b ADDRESS (City, State, and ZIP Code)<br>Monterey, California 93943-5000 |
| 8a NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b OFFICE SYMBOL (If applicable) | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c ADDRESS (City, State, and ZIP Code) | 10 SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO |

11 TITLE (Include Security Classification)
DESIGN OF A SENSOR-BLENDING KALMAN FILTER FOR R2P2 FINE-TRACKING SYSTEM

12 PERSONAL AUTHOR(S)
BAUER, Terrance J.

| 13a TYPE OF REPORT | 13b TIME COVERED | 14 DATE OF REPORT (Year, Month, Day) | 15 PAGE COUNT |
|---|---|---|---|
| Master's Thesis | FROM _____ TO _____ | 1990 March | 87 |

16 SUPPLEMENTARY NOTATION 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 US Government.

| 17 COSATI CODES | | | 18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | Kalman Filter; Adaptive Gating; Sensor Blending |
| | | | |
| | | | |

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The performance of a sensor-blending scheme for two different bandwidth sensors is significantly improved when a Kalman filter is used to blend the outputs vice classical control methods. This Kalman filter signal blender is designed and implemented in a computer program developed for this thesis. Several tracking scenarios are simulated and analyzed. These scenarios are representative of the input expected into the sensors on a Space Based Laser.

| | |
|---|---|
| 20 DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS | 21 ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a NAME OF RESPONSIBLE INDIVIDUAL<br>TITUS, Harold A. | 22b TELEPHONE (Include Area Code)<br>408-646-2650<br>22c OFFICE SYMBOL<br>ECTs |

DD Form 1473, JUN 86 *Previous editions are obsolete*

Approved for public release; distribution is unlimited

Design of a Sensor-Blending Kalman Filter for the R2P2 Fine-Tracking System

by

Terrance J. Bauer
Captain, U.S. Army
B.S., United States Military Academy, West Point, New York, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN
ELECTRICAL ENGINEERING

from the

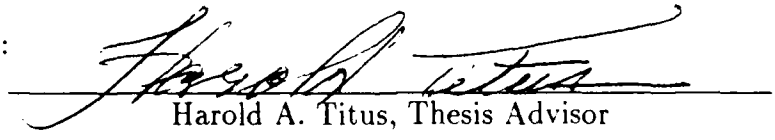
NAVAL POSTGRADUATE SCHOOL

March 1990

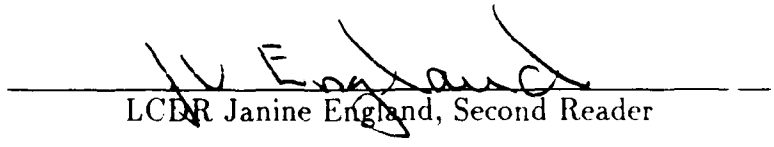
Author:

Terrance J. Bauer

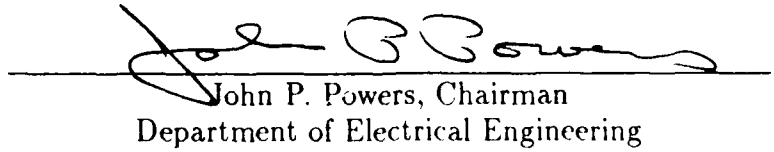
Approved by:



Harold A. Titus, Thesis Advisor



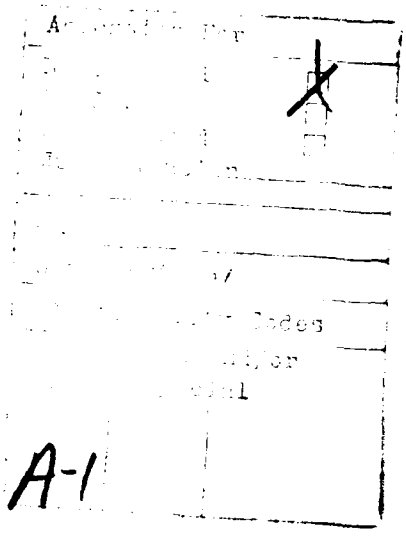
LCDR Janine England, Second Reader



John P. Powers, Chairman
Department of Electrical Engineering

# ABSTRACT

The performance of a sensor-blending scheme for two different bandwidth sensors is significantly improved when a Kalman filter is used to blend the outputs vice classical control methods. This Kalman filter signal blender is designed and implemented in a computer program developed for this thesis. Several tracking scenarios are simulated and analyzed. These scenarios are representative of the input expected into the sensors on a Space Based Laser.



# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ACKNOWLEDGMENTS

This thesis is the result of a lot of miles and a lot of pushing. The miles come from talking to Bruce Connors at Martin Marietta in Denver. He gave me the topic and assistance to complete it. The pushing was from my wife, Jane, who was not going to go through the last-minute finish-your-thesis routine.

I would also like to acknowledge the immense assistance I received from Captain Steve Spehn on the finer points of Kalman Theory and from Professor Hal Titus. Professor Titus kept me on track the whole way. It was a great pleasure working with him.

# I. INTRODUCTION

## A. GENERAL

The weapons being developed for the Strategic Defense Initiative require unprecedented pointing accuracies. For the case of the Space Based Laser (SBL) and Neutral Particle Beam, the pointing accuracy required is analogous to hitting a beach ball on the Empire State Building with a laser on Pike's Peak in Colorado. The problem does not end with being able to hit the beach ball; the laser has to illuminate the target for a specified period of time. The United States Army Strategic Defense Command has a precision pointing test bed located near Denver, Colorado. This facility is operated by the Martin Marietta Corporation. The test bed facility, known as the Rapid Retargeting/Precision Pointing (R2P2) facility is the vehicle through which the technologies required for the high pointing accuracies and rapid retargeting are being developed and tested. The R2P2 facility is currently configured to simulate the Space Based Laser, the inertial reference unit and the various other SBL components.

The heart of the R2P2 facility and the SBL is the fine pointing system. The fine pointing system's mission is to keep the line of sight of the weapon system pointed at the target. Steering mirrors are used to control the inertial line of sight angle. The error signal received by the steering mirrors can be treated as the difference of two signals, target position command angle ($0 < f < 0.5\ Hz$) minus the line of sight feedback angle ($0 < f < 40Hz$). The angles include disturbances such as command vehicle motion and beam expander structural vibration. The steering mirrors must

track the low frequency target and filter the high- and low-frequency disturbances from the line of sight.

Presently, the low frequency portion of the steering mirror error signal is provided by the Alignment Inertial Reference (AIR) platform, Figure 1.1. Due to the nature of the application and the type of sensor, the fine tracker operates at a low sampling rate and cannot provide high frequency information. The proposed concept is to use the AIR platform as a pseudo target, or cooperative target. It provides a mirrored surface pointed at the target and located on the weapons system. An alignment system marker beam is reflected by this surface and a sensor, other than the fine tracker, is used to obtain line of sight information. This alignment sensor does not have the low sample rate restriction and can be used to obtain high frequency information. The command signal for the AIR platform is formed by a sum of signals from the fine tracker, the alignment sensor and the AIR platform angle sensor.

The Strategic Defense Command and Martin Marietta desire an alternative approach for the fine pointing system on R2P2. The improvement, Figure 1.2, involves eliminating the AIR platform from the loop. Low frequency target data is obtained from the fine tracker, which samples at 50 Hz. A second signal is formed by blending the output from two sensors that measure the beam expander angle, a strap-down gyroscope and a magneto-hydrodynamic (MHD) angular vibration sensor. The strap-down gyro yields low frequency information while the MHD is designed to give high frequency observations. The difficulty with this scheme is the blending of the two signals to produce a broad-band measurement of the beam expander angle. The output of the alignment sensor is subtracted from the output of the signal mixer to yield a high frequency line of sight angle measurement. A second

signal mixing network combines this signal with the low frequency information from the fine tracker.

This research project focused on the mixing of the measurements from the two sensors, the gyroscope and the MHD, in an effort to fulfill the stated requirements. Those requirements, put forth by Martin Marietta, were:

1. Extremely accurate tracking of input signal.
2. Extremely fast lock on time, 20 ms or better.
3. Flatness in magnitude and phase for the combined low pass and high pass sensors as shown in Figure 1.3.
4. Steep cutoff rates for the outputs of the individual compensating filters, to minimize noise contributions from the individual sensors in their non-valid regions of measurement.
5. A selectable blending frequency, selectable at any point between $\omega_m$ and $\omega_g$ in order to blend the sensors for minimum noise.
6. Minimal sensitivity of the compensating network to parameter variation in the sensors.
7. Minimum number of poles and minimum DC gain in the compensating filters.

To meet these requirements, a Kalman filter was designed to mix the outputs from the two sensors. The Kalman predicts the states of the sensors, discarding the noise, based on previous measurements. The results should be the correct frequency response and an extremely accurate tracking of $\theta_{BX}$. The results of the Kalman filter signal blending will be compared with the signal blending filter scheme that

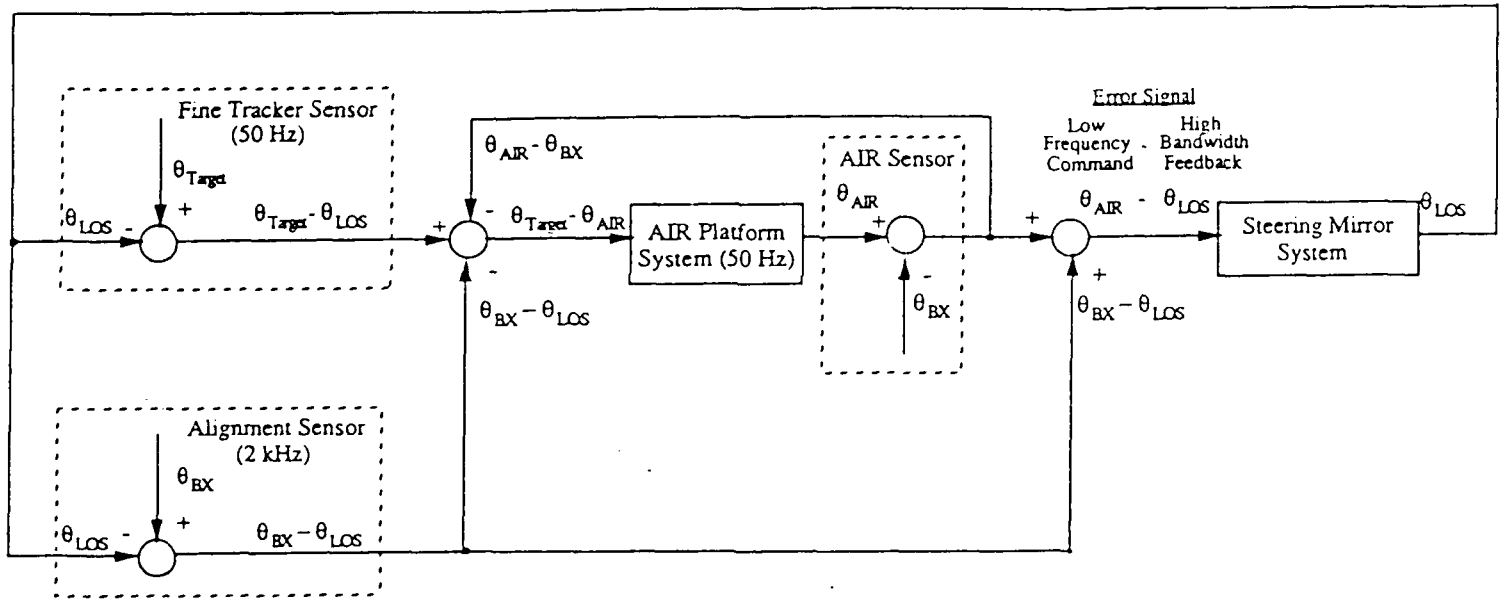



Figure 1.1: Currently Implemented Fine Pointing System

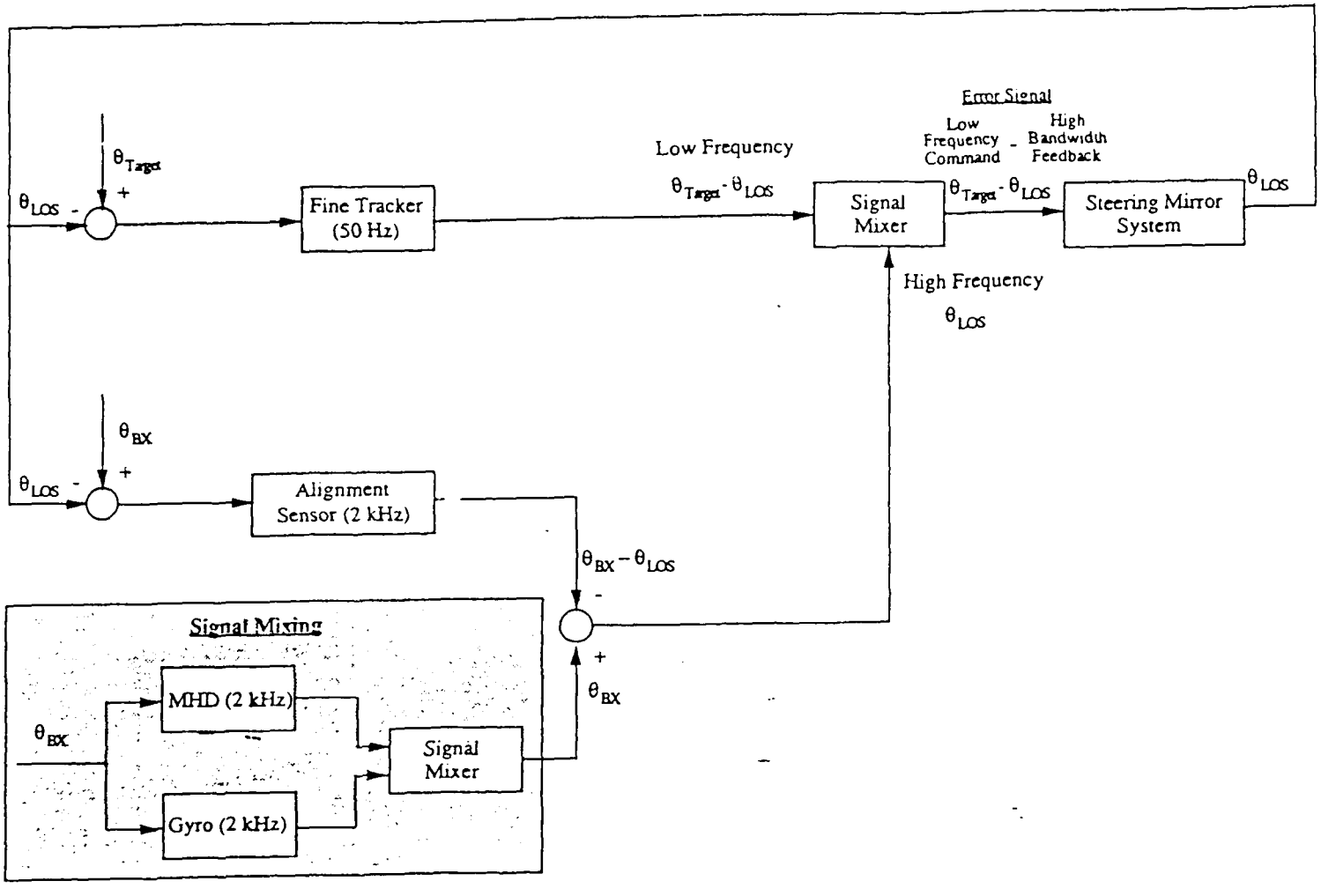



Figure 1.2: Alternative R2P2 Fine Pointing System

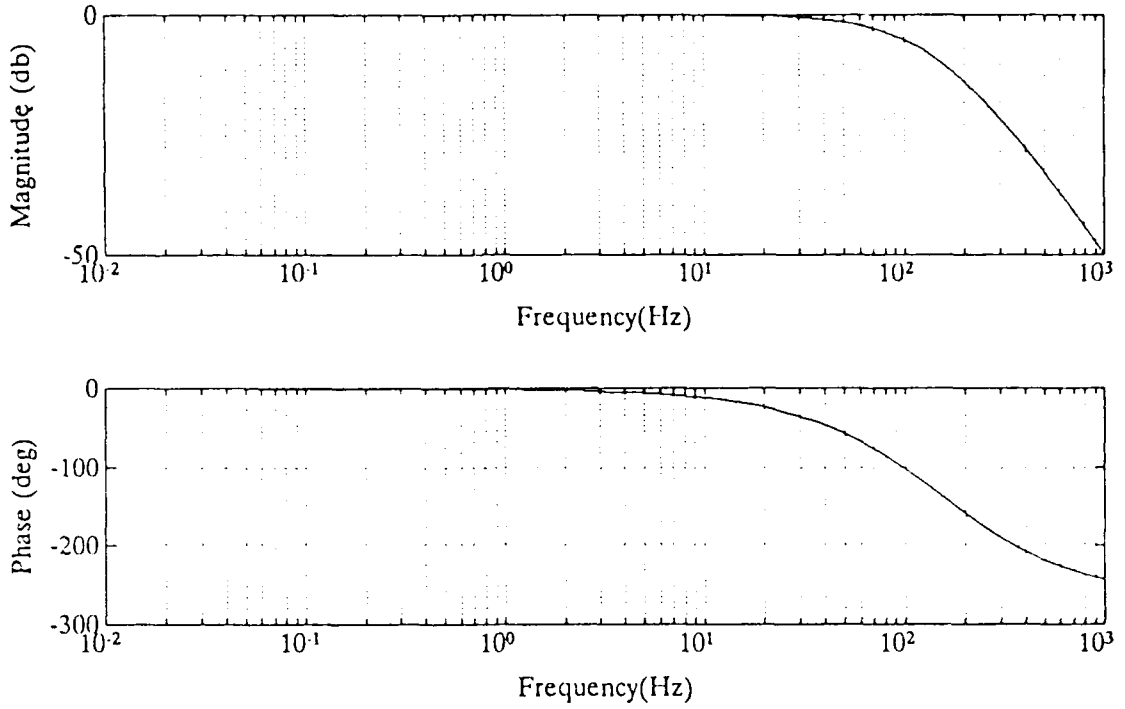



**Figure 1.3: Desired Frequency Response for System**

Martin Marietta has devised, utilizing classic filter design. Different sensor types will be compared with these results in an effort to further improve the design.

# II. PROBLEM STATEMENT

## A. GENERAL

The filter used involves two sensors with different bandwidths, measuring a common input. The filter then blends the two inputs using Kalman techniques.

The problem was developed using state space methods. Given the noise cluttered input angle, $\theta$, we are interested in the noise-free measurement of this angle over a broad band of frequencies. The state variables, ($x1$, $x2$, $x3$, $x4$, $x5$), for this plant are $\theta$, $\theta_G$, $\dot{\theta}_G$, $\theta_M$, and $\dot{\theta}_M$ as defined in Table 2.1.

**TABLE 2.1: STATE VARIABLE DEFINITIONS**

| | | |
|---|---|---|
| $x1$ | True state to be tracked | $\theta$ |
| $x2$ | Gyroscope angle | $\theta_G$ |
| $x3$ | Gyroscope angle rate | $\dot{\theta}_G$ |
| $x4$ | MHD angle | $\theta_M$ |
| $x5$ | MHD angle rate | $\dot{\theta}_M$ |

## B. SYSTEM MODEL

The system to be modeled in this problem is that of an inertial reference unit on the Space Based Laser. In the development of this work, the assumption was made that all noise encountered is white noise.

The simplified block diagram for the system is

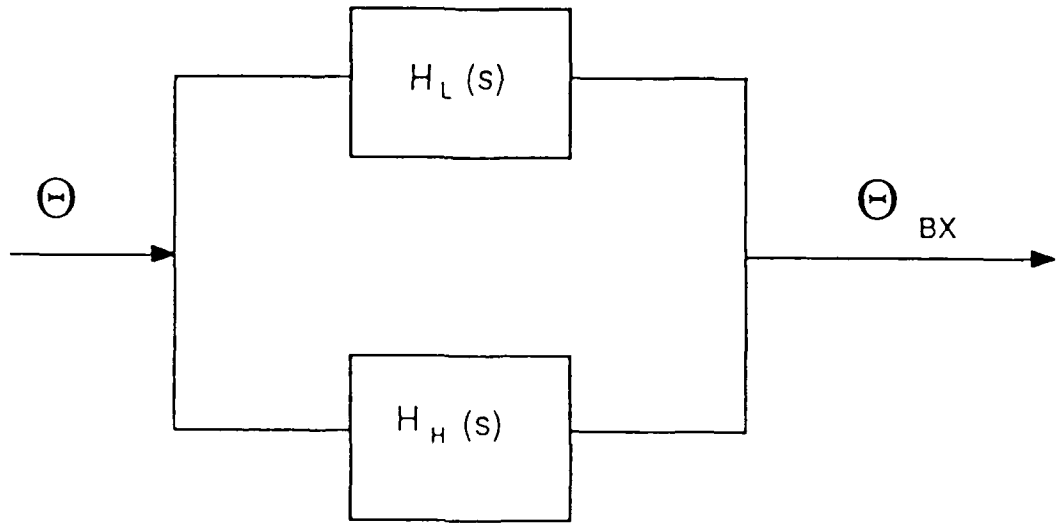



**Figure 2.1: Simplified Block Diagram for Sensors**

The transfer functions for the two sensors were given by Martin Marietta [Ref. 1] from manufacturer data and testing. The gyroscope's transfer function is

$$H_H(s) = \frac{3947.8}{s^2 + 88.844s + 3947.8} \tag{2.1}$$

The MHD transfer function is

$$H_L(s) = \frac{\frac{\omega^2}{2}(s+2)}{s^2 + 12.57s + 157.91} \tag{2.2}$$

Figure 2.2 shows the frequency response of both sensors. It can be seen that both sensors are second-order systems.

The continuous state space equations for the modeled system are

$$\dot{x} = Ax + Bw \tag{2.3}$$

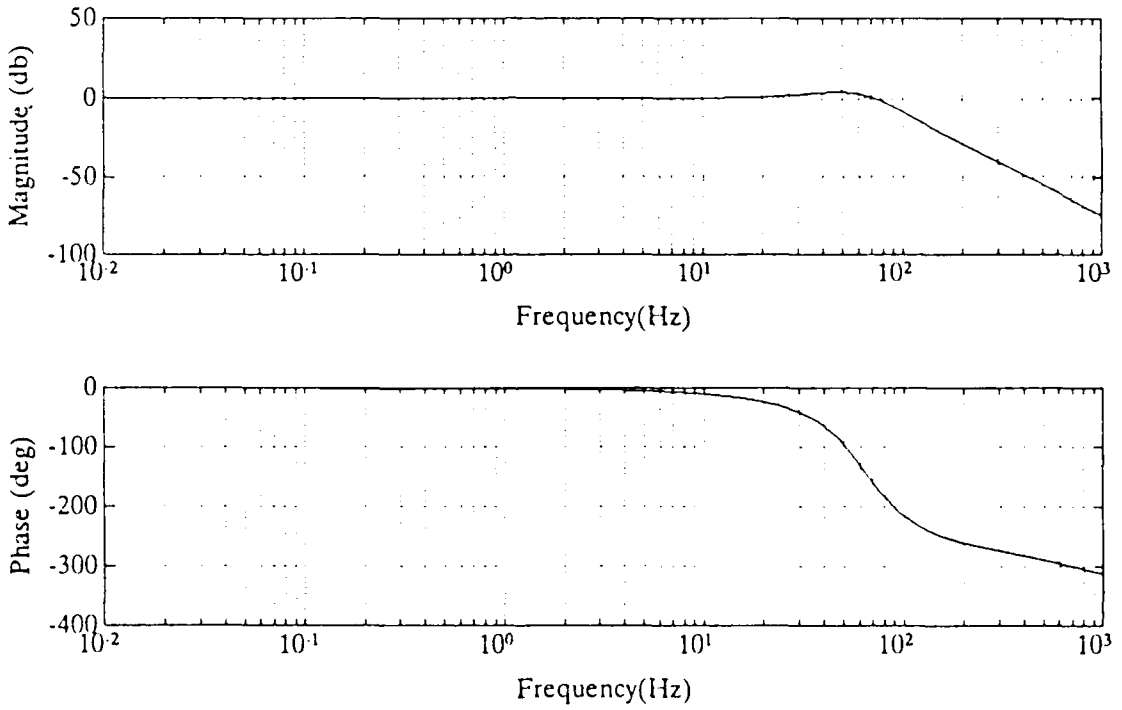



**Figure 2.2: Frequency Response of Both Sensors**

$$y = Cx + v \tag{2.4}$$

where

- $B$ = input driving function matrix
- $C$ = measurement matrix
- $w$ = system noise matrix
- $v$ = sensor noise.

and the state vector is

$$\underline{x} = \begin{bmatrix} \theta \\ \theta_G \\ \dot{\theta}_G \\ \theta_M \\ \dot{\theta}_M \end{bmatrix} \tag{2.5}$$

Using Equations 2.1 and 2.2 and the requirements for the phase and magnitude of the output, the A matrix can be formed as

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \omega_G^2 & -\omega_G^2 & -2\zeta\omega_G & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \omega_M^2 & 0 & 0 & -\omega_M^2 & -2\zeta\omega_M \end{bmatrix} \tag{2.6}$$

where

- $\omega_G$ = gyro cutoff frequency
- $\omega_M$ = MHD cutoff frequency
- $\zeta$ = damping coefficient for each sensor

For the model, it is desired that the fastest reaction time possible is achieved. To do this, the system is critically damped, $\zeta = 1$. The cutoff frequencies come from sensor specifications and testing. Equations 2.1 and 2.2 reflect the cutoff frequencies and damping coefficient values given. The classical $2^{nd}$ order damped system has the form

$$H_{(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega + \omega^2} \tag{2.7}$$

Discretizing the state equations yields the following discrete state space equations,

$$x_{k+1} = \phi x_k + \Delta w_k \tag{2.8}$$

where

- $x_k$ = parameter to be estimated (State Vector).
- $\phi$ = state transition matrix which describes how the states of the dynamic system are related.
- $\Delta$ = state transition matrix for input driving function.
- $w_k$ = system noise matrix.

From Equation 2.8 and the above assumptions, the $\phi$ matrix is

$$\phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 4.83 \times 10^{-4} & 0.999 & 4.854 \times 10^{-4} & 0 & 0 \\ 1.913 & -1.913 & 0.9886 & 0 & 0 \\ 1.966 \times 10^{-5} & 0 & 0 & 0.999 & 4.965 \times 10^{-4} \\ 7.846 \times 10^{-2} & 0 & 0 & -7.846 \times 10^{-2} & 0.9875 \end{bmatrix} x_k \tag{2.9}$$

The system noise for the model comes from the input that the sensors are measuring. This input will have noise from the vehicle, the mirrors and the beam expander. This noise was modelled in accordance with the R2P2 observations by Martin Marietta.

## C. MEASUREMENT MODEL

For a linear measurement process, the measurements are linearly related to the state variables and can be modeled using the discrete linear measurement equation from Equation 2.4,

$$z_k = Hx_k + v_k \tag{2.10}$$

where

- $z_k$ = set of measurements
- $H$ = observation matrix that gives the relationship between the measurements and the state vector
- $x_k$ = state vector
- $v_k$ = measurement noise from the sensors

With the appropriate values for $H$, Equation 2.10 becomes

$$z_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} \end{bmatrix} x_k + v_k \tag{2.11}$$

In this blending problem, the measurements are made of the beam expander by the sensors that make up the inertial reference unit. The measurements are made noisy by the noise inherent in the sensors. The sensors have been rigorously tested and the power spectral densities have been computed by Martin Marietta. Figure 2.3 shows the computed noise spectra for the two sensors.

The noise from the sensors is a function of many variables including temperature and bandwidth to be measured. Although this is generally a non-white, non-gaussian noise process, it can be adequately described as a white noise process over an extended period of time.

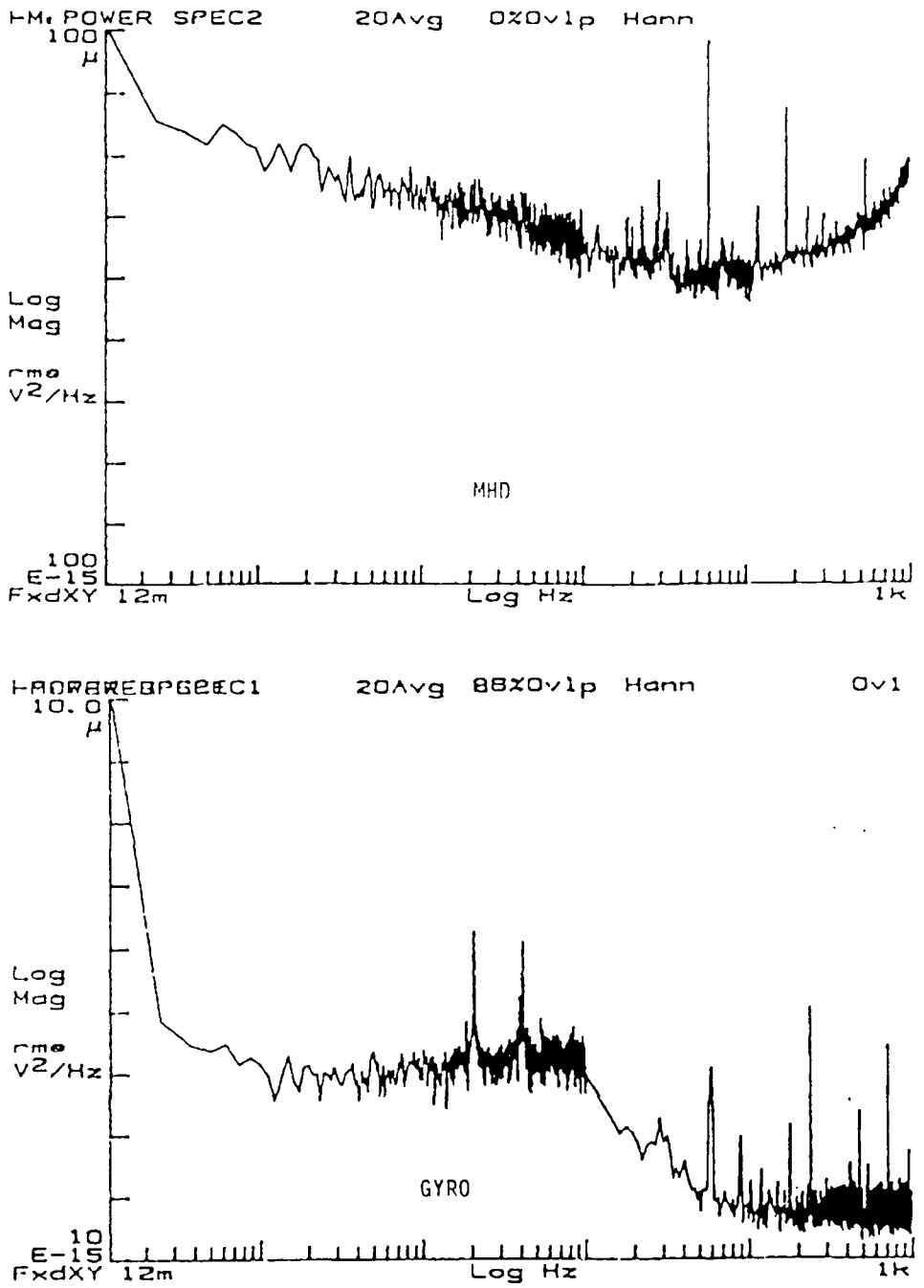



**Figure 2.3: Noise Spectral Densities for Both Sensors**

The state and measurement equations are now ready to be implemented in the simulation. What follows is the development of the Kalman filter equations that are the heart of this exercise.

# III. KALMAN FILTER THEORY

## A. GENERAL

Filtering refers to the process of estimating the state vector at the current time, based upon all past measurements. An optimal filter concentrates on optimizing a specific performance measure used to approximate the quality of the estimate. The Kalman filter is the optimal filter in a class of linear filters that minimize the mean square estimation error between actual and desired output. In other words, the Kalman filter attempts to minimize the elements along the main diagonal of the state error covariance matrix. The Kalman filter has been used extensively in the design of estimation models since it was first presented by Kalman and Bucy [Ref. 5] in 1960. The filter itself is actually a recursive algorithm for processing discrete measurements or observations in an optimal manner. [Ref. 6:p. 104] *A priori* knowledge of the state estimate and its error covariance, and the current observation is required. The Kalman filter is a useful algorithm when both the system model and the measurement model are linear functions of the state variables and these models can be described by the equations

$$x_{k+1} = \phi_k x_k + \Delta w_k \tag{3.1}$$

$$z_k = H x_k + v_k \tag{3.2}$$

## B. SYSTEM MODEL

The state space model of the system is given by Equation 3.1 and the measurements are described by Equation 3.2. This is a standard state space matrix

representation for a system of linear differential equations. In Equation 3.1, $x_k$ represents the physical state and $x_{k+1}$ represents the next state of the discrete system.

The values $\phi$ and $\Delta$ represent the discrete time state matrices. The value of $x_k$ is the true observed parameters of the state and $v_k$ and $w_k$ are observation noise and state expectation noise, respectively.

This system is time invariant since neither $\phi$ nor $H$ is dependent on time. The noise processes are considered to be stationary, independent, white gaussian noise with zero mean. This assumes that white noise is an idealization of nature's true state; however, it is an extremely good approximation for many systems. The statistical properties of the noise are given below.

$$E[w_k] = 0 \tag{3.3}$$

$$E\left[w_j w_k^T\right] = Q\delta_{jk} \tag{3.4}$$

$$E[v_k] = 0 \tag{3.5}$$

$$E\left[v_j v_k^T\right] = R\delta_{jk} \tag{3.6}$$

$$E\left[w_j v_k^T\right] = 0 \tag{3.7}$$

The matrices $Q$ and $R$ in Equations 3.4 and 3.6 are the covariance matrices for the noise processes. For this system, the noise covariance matrices are non-zero diagonal matrices, which denote the power present in the noise. This model will be further discussed in Chapter 4.

## C. LINEAR RECURSIVE FORM

Before deriving the filter equations, the form of the filter must first be determined. The form assumed for most Kalman filters is shown in Equations 3.8 and 3.9.

$$\hat{x}_{k+1|k} = k_1 \hat{x}_{k|k} \tag{3.8}$$

$$\hat{x}_{k+1|k+1} = k_2 \hat{x}_{k+1|k} + k_3 z_{k+1} \tag{3.9}$$

The current estimate, $\hat{x}_{k+1|k+1}$, is a linear combination of the previous estimate, $\hat{x}_{k+1|k}$, and the current observation, $z_{k+1}$. This form is chosen for its simplicity, but Reference 4 demonstrates it is optimal for a linear system.

## D. ERROR COVARIANCE

The error covariance matrices are described by Equations 3.10 and 3.11,

$$P_{k+1|k} = E\left[\hat{x}_{k+1|k}\hat{x}^T_{k+1|k}\right] \tag{3.10}$$

$$P_{k+1|k+1} = E\left[\hat{x}_{k+1|k+1}\hat{x}^T_{k+1|k+1}\right] \tag{3.11}$$

These matrices give a feeling for the expected magnitude of the estimation error. Their derivation can be found in Reference 6:p. 108. The Kalman equations begin to take shape when Equations 3.10 and 3.11 are combined,

$$P_{k+1|k} = \phi P_{k|k} \phi^T + Q \ . \tag{3.12}$$

Additionally, writing Equation 3.11 and incorporating the equations found in Reference 2 in the development of the covariance matrix, we get

$$P_{k+1|k+1} = (I - GH)\, P_{k+1|k}\, (I - GH)^T + GRG^T \tag{3.13}$$

where $G$ is the Kalman gain matrix. All that remains is to find the value of this Kalman Gain matrix.

## E. RESIDUAL AND VARIANCE

The definition of the residual will be helpful in simplifying the notation required for the remainder of the proof. The basis for the residual and its variance came from conversations with Steve Spehn [Ref. 7]. The residual is given by

$$r_{k+1} = z_{k+1} - E\left[z_{k+1}\right] \tag{3.14}$$

Since the estimate is unbiased, we see that

$$E[z_{k+1}] = E[Hx_{k+1}] + E[v_{k+1}] \tag{3.15}$$

$$E[z_{k+1}] = H\hat{x}_{k+1|k} \tag{3.16}$$

By substitution and algebra, we get the final form of the residual,

$$r_{k+1} = H\tilde{x}_{k+1|k} + v_{k+1} \tag{3.17}$$

(This derivation is from Reference 7.)

The covariance of the residual is found to be

$$var\ [r_{k+1}] = E\left[r_{k+1}r_{k+1}^T\right] \tag{3.18}$$

$$var\ [r_{k+1}] = HP_{k+1|k}H^T + R\ . \tag{3.19}$$

Using the definition of the residual, the observation update equation can be written as

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + Gr_{k+1} \tag{3.20}$$

The Kalman Gain equations can now be derived.

## F. KALMAN GAINS

Solving Equation 3.13 for $G$ gives,

$$G_{k+1} = P_{k+1|k}H^T\left(HP_{k+1|k}H^T + R\right)^{-1} \tag{3.21}$$

Recognizing the form of the equation in parenthesis to be that of Equation 3.19, we simplify to the final form.

$$G_{k+1} = P_{k+1|k}H^T var\,[r_{k+1}]^{-1} \tag{3.22}$$

Using techniques developed in Reference 3, we simplify Equation 3.13 to

$$P_{k+1|k+1} = [I - G_{k+1}H]\,P_{k+1|k} \tag{3.23}$$

## G. KALMAN FILTER EQUATIONS

This derivation has provided a set of recursive equations, which give a time-varying optimal gain matrix and a detected error analysis of the estimate. The Kalman filter equations are given below.

$$\hat{x}_{k+1|k} = \phi \hat{x}_{k|k} \tag{3.24}$$

$$P_{k+1|k} = \phi P_{k|k} \phi^T + Q \tag{3.25}$$

$$G_{k+1} = P_{k+1|k} H^T \left( H P_{k+1|k} H^T + R \right)^{-1} \tag{3.26}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + G_{k+1} \left( z_{k+1} - H \hat{x}_{k+1|k} \right) \tag{3.27}$$

$$P_{k+1|k+1} = \left( I - G_{k+1} H \right) P_{k+1|k} \tag{3.28}$$

These equations can be further simplified using the definitions of the residual and covariance of the residual. This simplication will be incorporated in the simulations. Since the Kalman equations are recursive, they are readily adaptable to computer simulation. All that is required are the initial conditions:

$\hat{x}_{0|0}$, Initial estimate

$P_{0|0}$, Error covariance.

This *a priori* knowledge is essential to the Kalman process.

The Kalman equations are now ready to be implemented in estimating a normal system. The next step is to make the Kalman adaptable thereby increasing its bandwidth. This will be accomplished by adding maneuver gating to the Kalman filter.

# IV. MANEUVER GATING

## A. MAHALANOBIS DISTANCE

The Mahalanobis distance (MD) is a measure of the derivation of the observation from the estimate. The derivation of the MD is found in Reference 8. The idea for this procedure was derived from Reference 7.

The Mahalanobis distance is found using the values for the residual and covariance of the residual, Equations 3.14 and 3.19,

$$MD = r_{k+1}^T var\left[r_{k+1}\right]^{-1} r_{k+1} \tag{4.1}$$

The resulting scalar is compared with a desired threshold in the program. This threshold was picked at MD = 4, which corresponds to the statistical $2\sigma$ point for the noise processes.

## B. RESIDUAL GATING

Residual gating is the process by which the Kalman adapts itself to large jumps in the observation. The system being tracked in this simulation can be expected to have large, nearly step-shaped, changes in the observations being tracked. (The following derivation comes from Reference 7.)

A normal Kalman filter would observe this jump and initially considers it as a noise perturbation. The Kalman will therefore ignore the jump, for several steps. If the large value persists, the filter will begin to react with speed dependent upon the value of the covariance matrix, $P$, at the time.

This reaction, although a great benefit for slow-moving tracking situations, is extremely restrictive for this system. The requirement for lock-on in 20 milliseconds

demands a more proactive Kalman filter. Residual gating provides this proactive behavior.

Residual gating uses the Mahalanobis Distance derived earlier as the "gate" for the incrementation of the covariance matrix. There are two ways for a Kalman filter to adapt, either by increasing the gain $G_{k+1}$ or the covariance matrix, $P_k$. The covariance matrix was selected as the means for adaptation. The gate is set up using the $2\sigma$ value discussed earlier. A value of

$$MD \geq 4 \tag{4.2}$$

results in the observation falling outside the gate and begins the adaptive incrementing of $P$,

$$P_{k|k} = FP_{k|k} \ . \tag{4.3}$$

The constant, $F$, was used to adaptively increase the last value of the covariance matrix, $P_{k|k}$. The value of $F$ was derived experimentally to obtain a value that results in optimal filter performance. $F$ was found to cause little variance over a wide range of values.

Through analysis, it was decided to use a gating reset of $P_{0|0}$. This results in some lag time in the filter, which is made up for by its faster lock-on time.

The next step in the design is to simulate the inputs and scenarios the system will see. What follows are the various simulations used to test the filter's ability to meet system requirements.

# V. SIMULATIONS

## A. SCENARIOS

Several scenarios were developed for this simulation to test its applicability to the sensor blending problem. In all scenarios, observation noise was present. State excitation noise was varied.

### 1. Scenario One

This scenario introduced a 1 Hz square wave with various noise levels into the system. Figure 5.1 shows the input wave.

### 2. Scenario Two

This scenario introduced a 10 Hz square wave into the system with various noise levels. See Figure 5.2.

### 3. Scenario Three

This scenario introduced a 50 Hz square wave into the system with various noise levels. See Figure 5.3.

### 4. Scenario Four

The input for this scenario is a 100 Hz square wave. This input is the high limit provided by Martin Marietta. [Ref. 1] See Figure 5.4.

## B. NOISE INPUTS

The noise inputs for the model were developed from input provided by Martin Marietta [Ref. 1]. Figure 2.3 shows the noise spectral power values for the two sensors, MHD and Gyro. The values used throughout the simulations for the sensor noises were taken as the median from the graphs. The values were entered as $v_G$ and $v_M$, after conversion.

The values entered for state excitation noise, $w_k$, were derived from the expected range of the fine tracking system. Varying the level of $w_k$ enables one to test the robustness of the model and filter. The mean noise level was selected as $10^{-5}$ rad.

## C. RESIDUAL GATING

A test case was run for Scenario One input without residual gating. Figure 5.5 shows the results of a normal Kalman filter without residual gating. As can be seen, the performance is unacceptable for the accuracy requirements stated. It will serve as a good reference for the Kalman filter used in the remainder of the simulations.

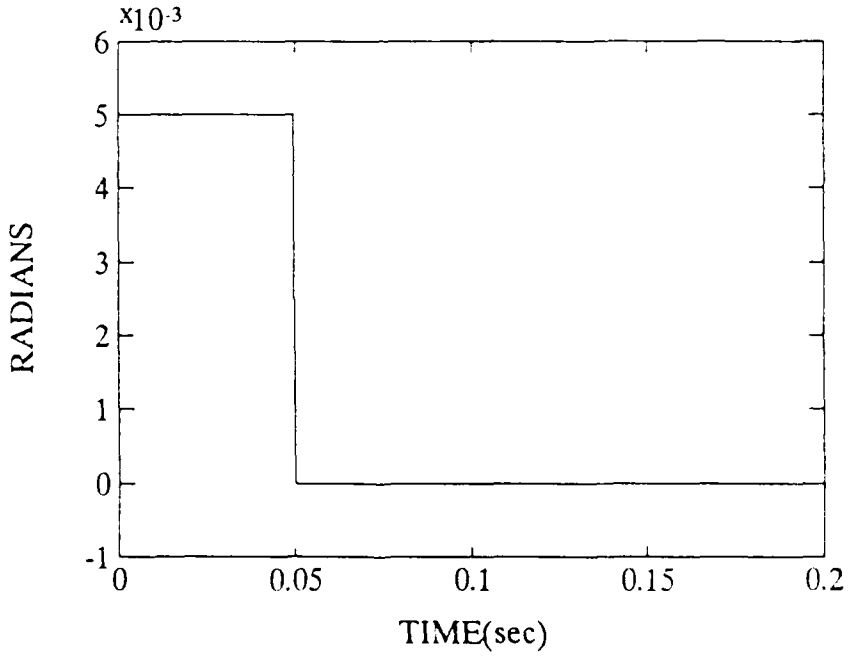



**Figure 5.1: Scenario One**

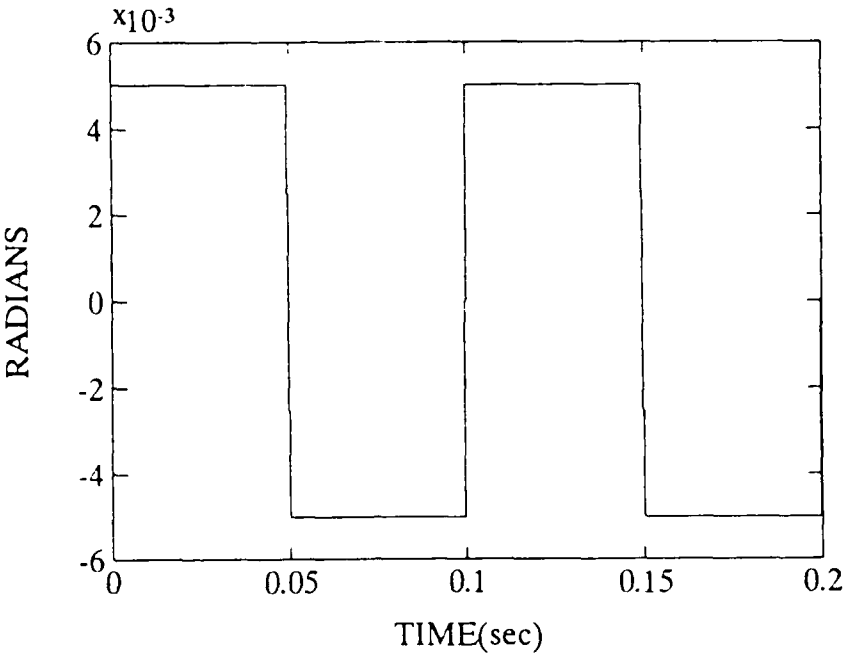



**Figure 5.2: Scenario Two**

## D. REQUIREMENT FOR FREQUENCY RESPONSE

The sponsor of this thesis, Martin Marietta, requested a frequency response of the filtered system as part of their specifications [Ref. 2]. The Bode plot developed from the model is a result of this requirement.

## E. BODE FORMULATION

A bode plot is a plot of a system transfer functions response over a range of frequencies. Martin Marietta desired a unity gain frequency response over the range of interested frequencies, 0.01-100 Hz. [Ref. 1] A transfer function was generated using steps put forth in Reference 6 for a Wiener steady state optimal filter. A Wiener filter is an optimal filter, identical to the Kalman, if the statistics are Gaussian. The results of the derivation give a filter transfer function of the form

$$H(z) = \left[(zI - \phi + GH)^{-1} G\right] \tag{5.1}$$

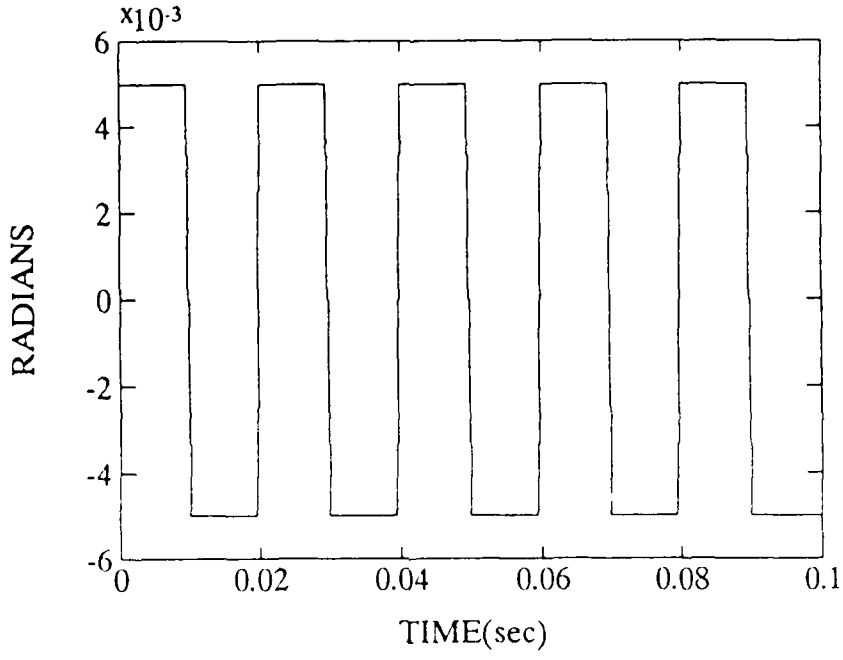



**Figure 5.3: Scenario Three**

The transfer function was derived using the program in Appendix B. This transfer function was combined with the sensor transfer functions, from Figure 6.45, and a Bode plot, Figure 6.46, was obtained.

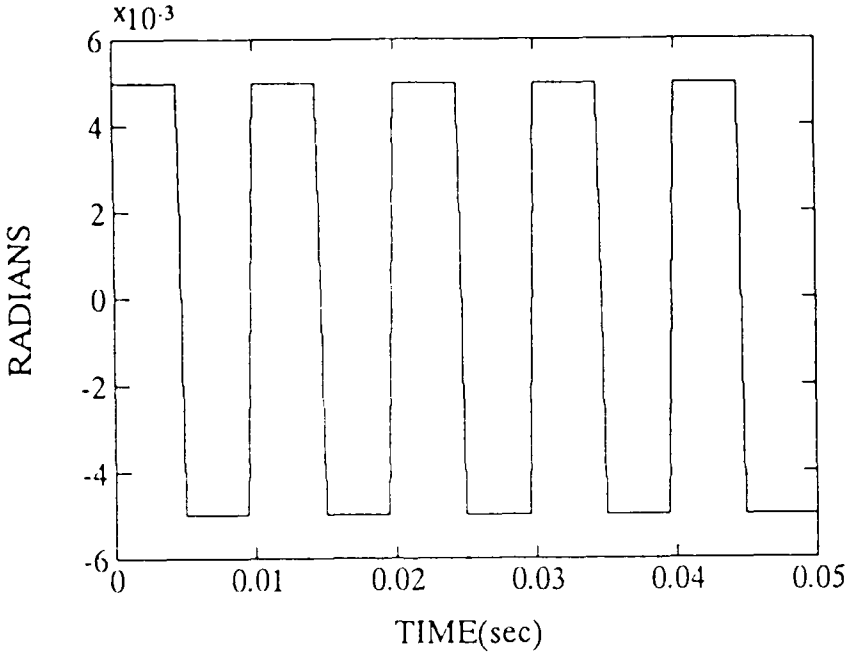



Figure 5.4: Scenario Four

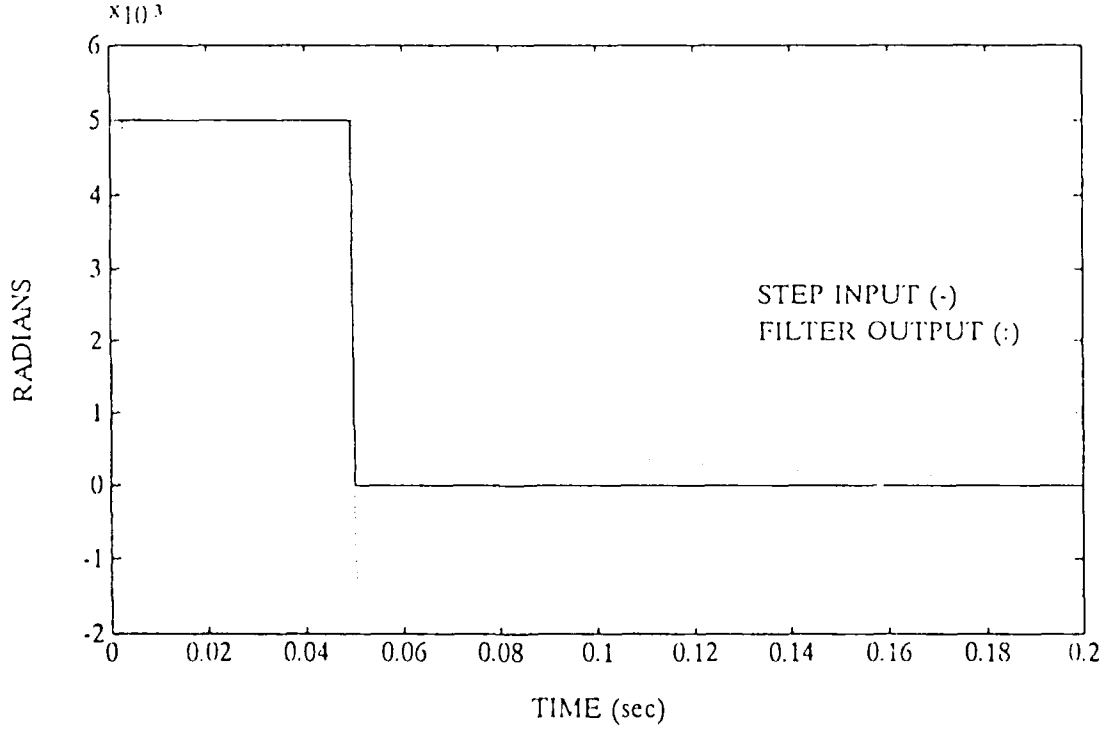



Figure 5.5: Scenario One, No Residual Gating

# VI. DISCUSSION OF RESULTS

## A. GENERAL

All of the simulations conducted were done using an IBM-PC and the software language PC-MATLAB. The program codes are contained in Appendices A and B. The results achieved could not be shown to completely satisfy the requirements put forth by Martin Marietta. Specificially, the frequency response of the steady state gain Kalman blended system did not meet the desired specifications. This inconsistency was resolved by the adaptive gating incorporated in the system designed. This will be discussed in detail in Section VI-C.

## B. KALMAN PERFORMANCE

The performance of the Kalman filter was evaluated through several steps of increasing noise and frequency of the input. The filter design was for step and square wave input, as per Martin Marietta's guidance [Ref. 1]. The Kalman is a Type 0 system, by design, so it will not be able to follow a ramp or sinusoid. It can be modified to follow those two inputs, but with the penalty of not being a real-time system any longer. The system this filter was built for, the R2P2, is extremely dependent on real-time results. Therefore, the Kalman was designed to be as fast as possible.

The first simulation conducted was for an input of 5 mrad that drops to 0 mrad at 0.05 seconds with $Q$=0 and $R \sim 0$. The $R$ matrix could not be made to equal zero due to MATLAB constraints. This simulation will act as a baseline for which the others will be compared. Figure 6.1 shows the input and filter output. Figure 6.2 illustrates the error between the actual input and the output of the Kalman. Figure

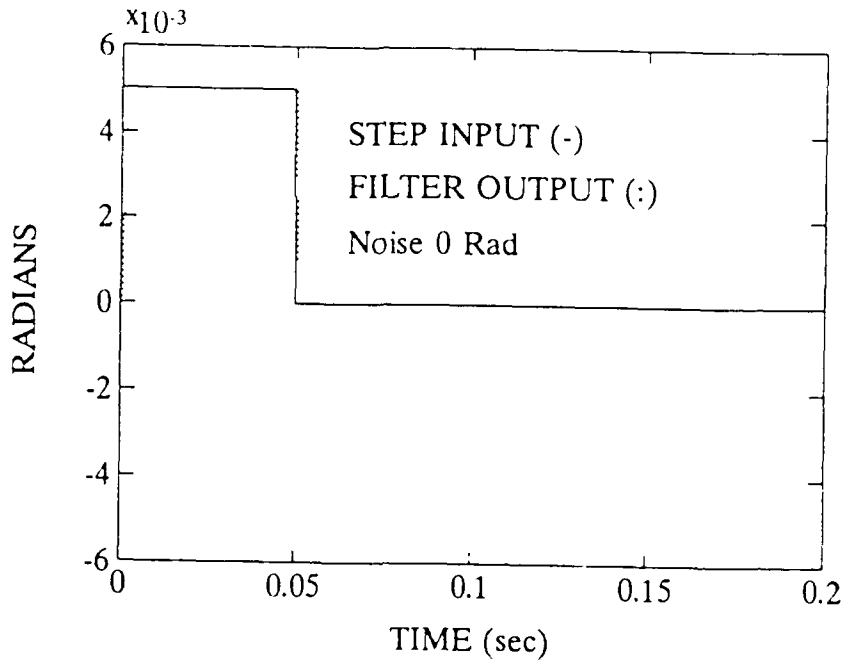



**Figure 6.1: Baseline - $X1$ Estimation for Model vs. Input**

6.3 shows the value of the mean of the residual over the period of the simulation and Figure 6.4 illustrates the ability of the filter to achieve rapid lock-on. The lock-on gate used in these simulations is $\pm$ 20 $\mu$rad. These graphs are of the first state ($x_1$) of the system, which is the state we are concerned with following. The no-noise input scenario is unrealistic, but is effective in giving a baseline for the rest of the analysis. With no noise, the Kalman is able to lock-on to the input in one time step. The mean of the residual and lock-on time are two ways of checking Kalman effectiveness. They will be used in analyzing the performance of the Kalman for the scenarios.

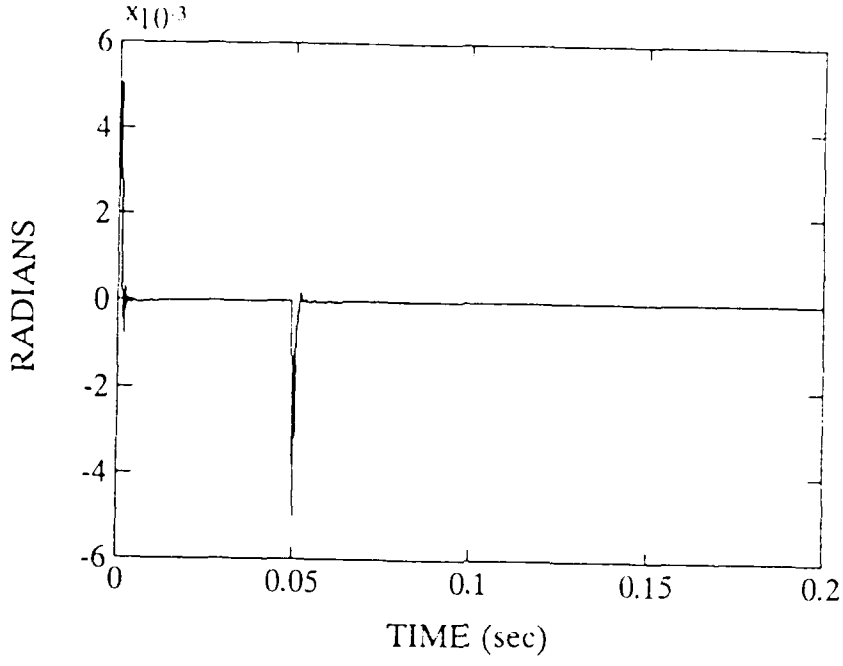



Figure 6.2: Baseline - Plot of Error Between Estimate and Input ($X1$)

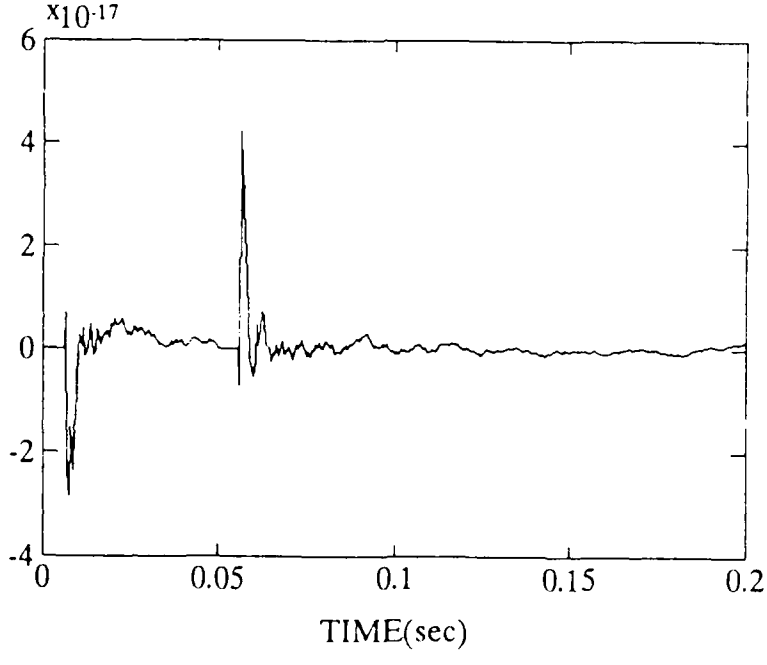



Figure 6.3: Baseline - Mean of Error

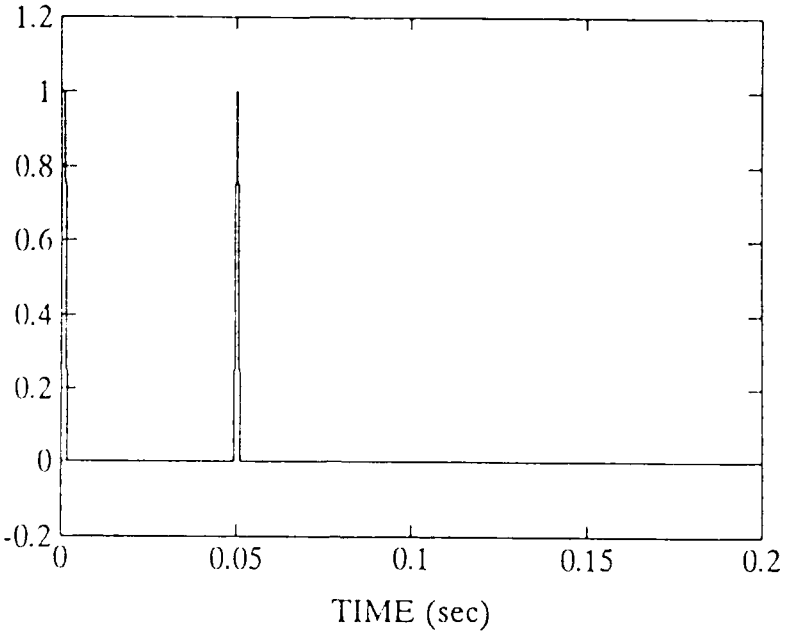



Figure 6.4: Baseline - System Lock-On Time

## C. SCENARIO RESULTS

### 1. Scenario One

Three different runs were made for Scenario One, in which the noise inputs were varied. The input signal was a 1 Hz square wave. The first run had the state noise covariance, $Q$, equal to 0 and the measurement noise covariance matrix equal to the values obtained from Figure 2.3. Figures 6.5 to 6.8 show the simulation results. Figures 6.9 to 6.12 show the results of the next run in which noise was introduced into the $Q$ matrix and $R \sim 0$. Figure 6.13 and 6.16 illustrate the results of entering representative noise into both the $Q$ and $R$ matrices.

As would be expected, the mean of the residual and lock-on times were progressively worse for each case. It is also obvious that state measurement noise, $R$, is the dominant noise input in the filter. The results of the final run of this

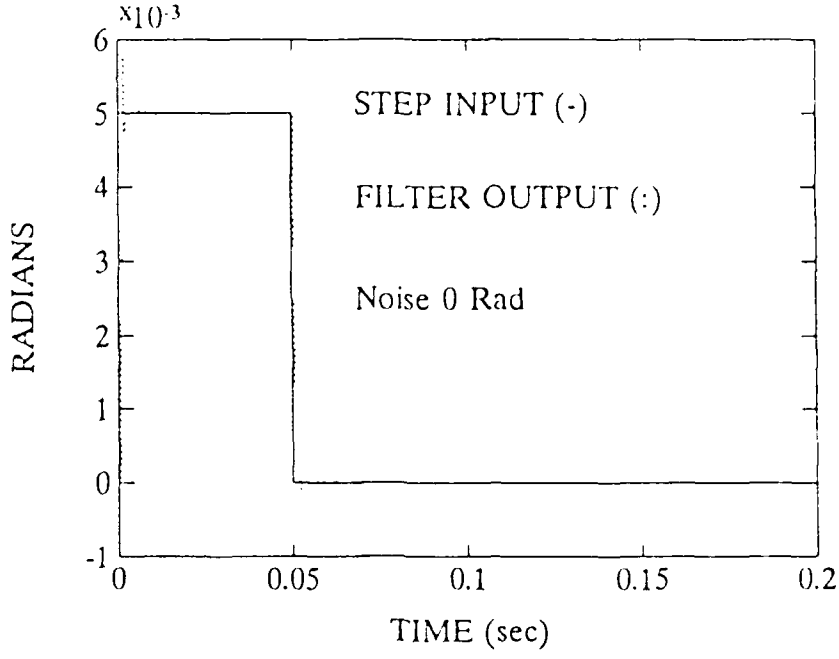



**Figure 6.5: $X1$ Estimation for Model vs. Input**

scenario are well within the desired specifications. The Kalman is locking on with little deviation in 10-15 time steps.

### 2. Scenario Two

This scenario takes the basic system and applies a 10 Hz square wave input with amplitude of $\pm$ 5 mrad. The values for $Q$ and $R$ will remain constant for the remainder of the scenarios. Figures 6.17 to 6.20 show the results of entering the 10 Hz wave into the Kalman.

For this input, the Kalman performs exceptionally well Lock-on, Figure 6.20, occurs in less than 20 time steps and the mean of the residual, Figure 6.19, and the error, Figure 6.18, are extremely low.

### 3. Scenario Three

Scenario Three applied a 50 Hz square wave into the Kalman. This frequency of input appears to tax the Kalman's ability to follow an input. Figure

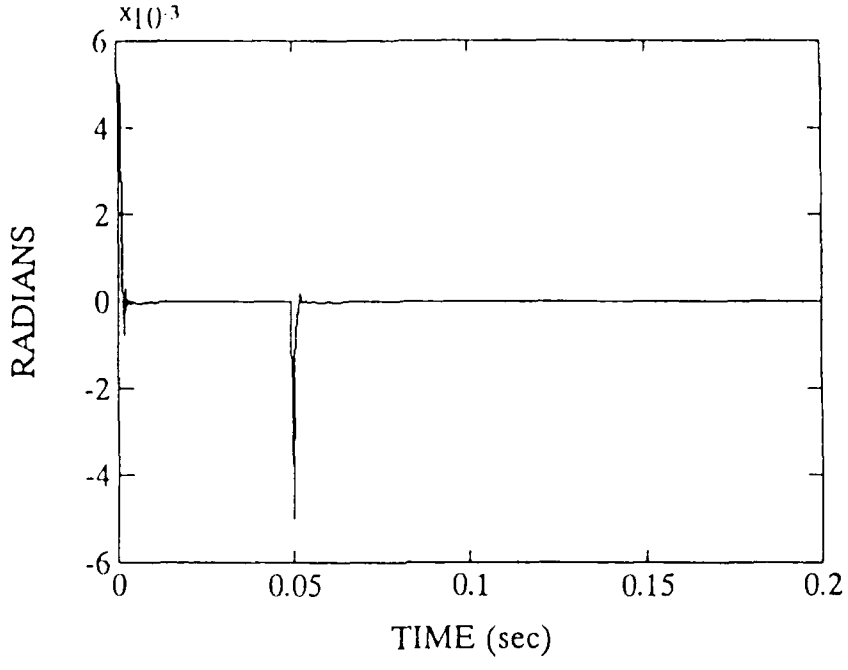



Figure 6.6: Plot of Error Between Estimate and Input (X1)

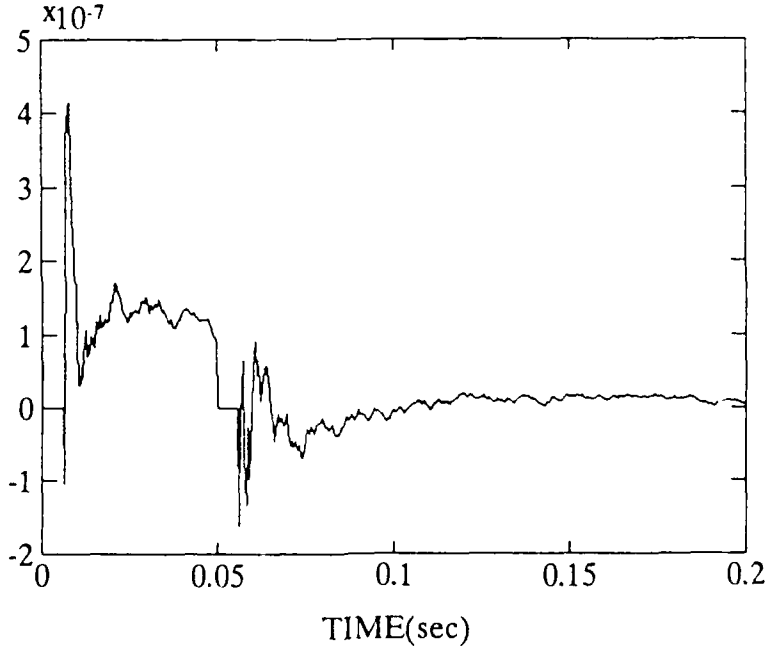



Figure 6.7: Mean of Error

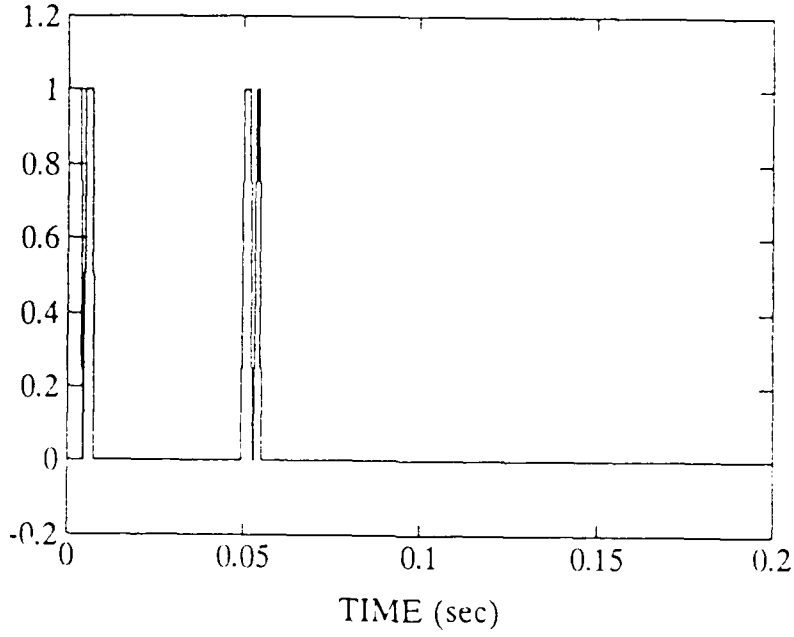



Figure 6.8: System Lock-On Time

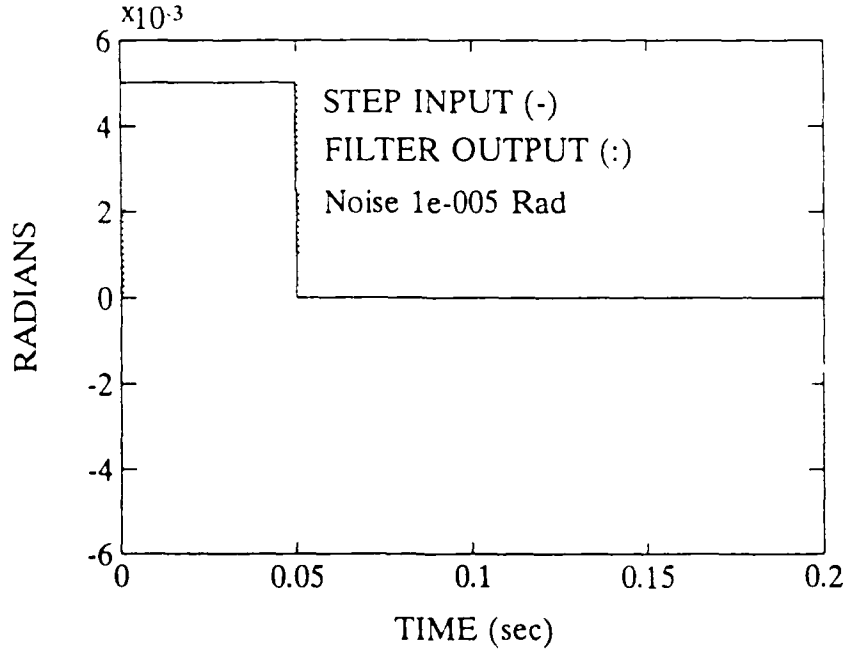



Figure 6.9: $X1$ Estimation for Model vs. Input

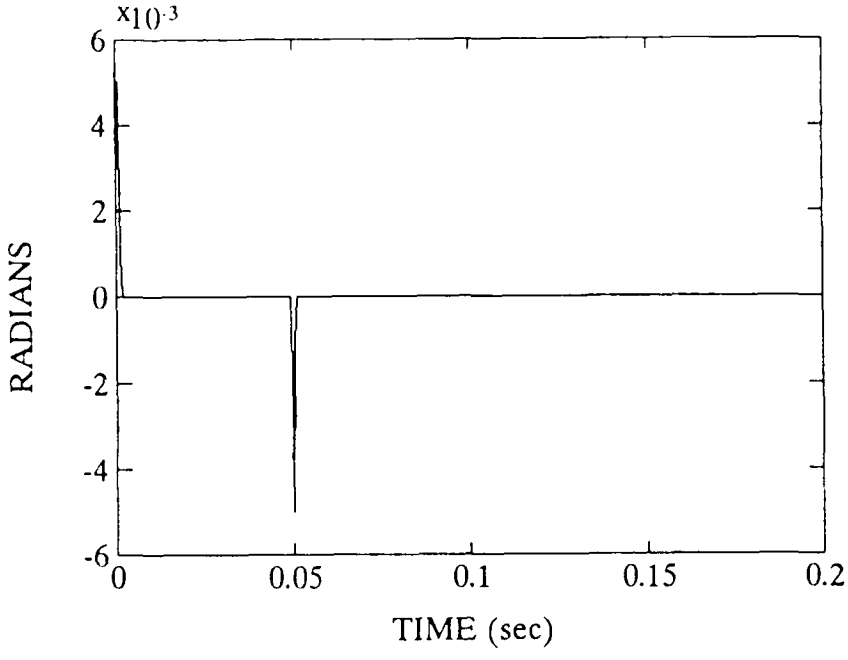



**Figure 6.10: Plot of Error Between Estimate and Input ($X1$)**

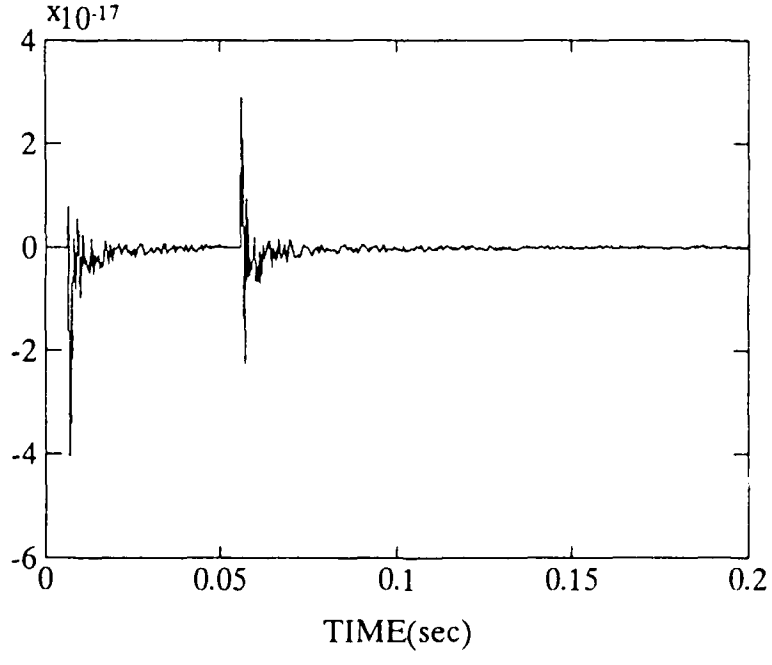



**Figure 6.11: Mean of Error**

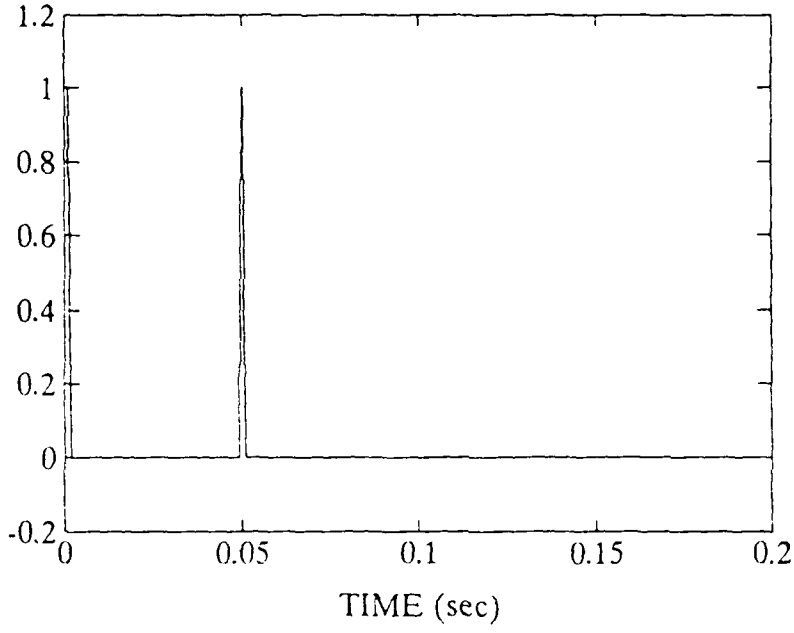



Figure 6.12: System Lock-On Time

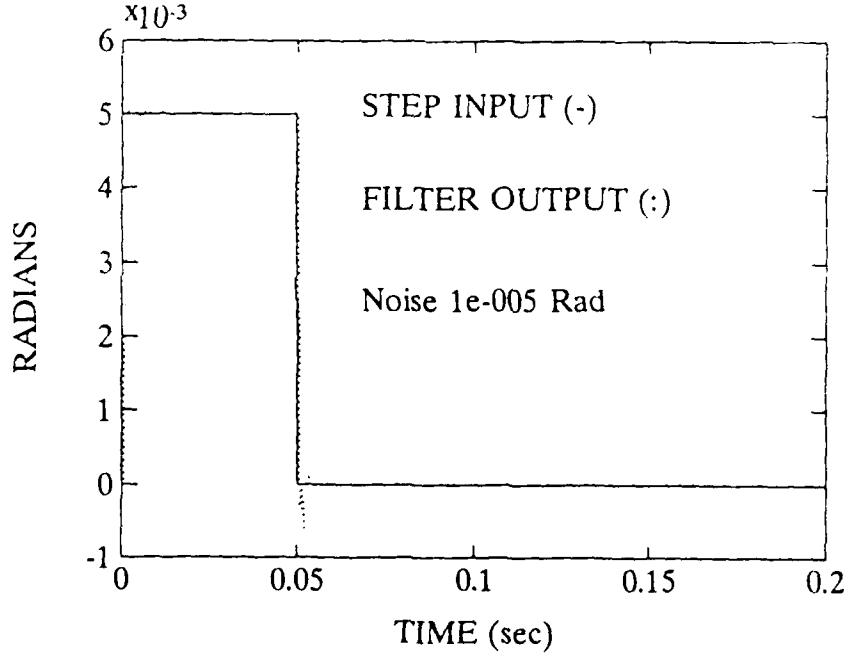



Figure 6.13: $X_1$ Estimation for Model vs. Input

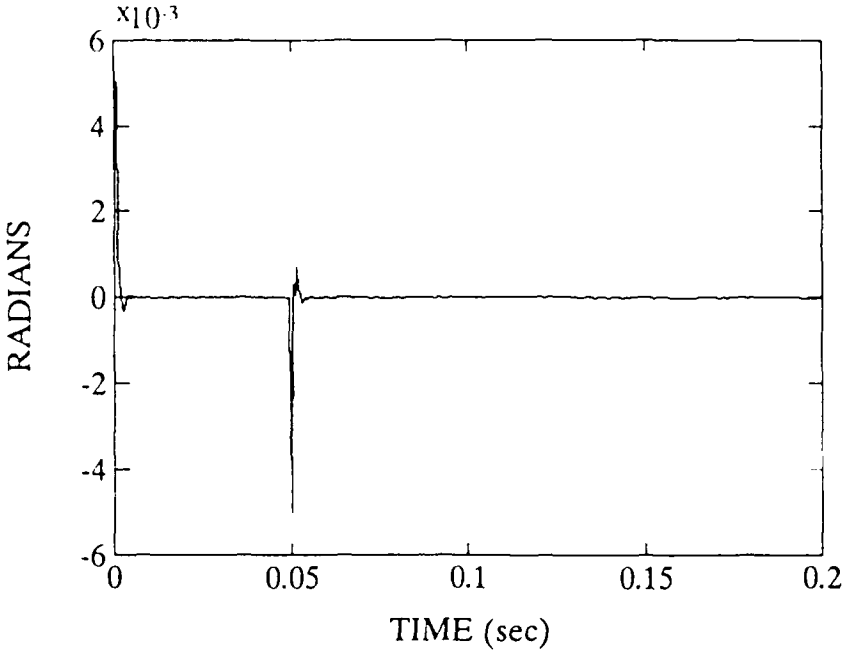



**Figure 6.14: Plot of Error Between Estimate and Input ($X1$)**

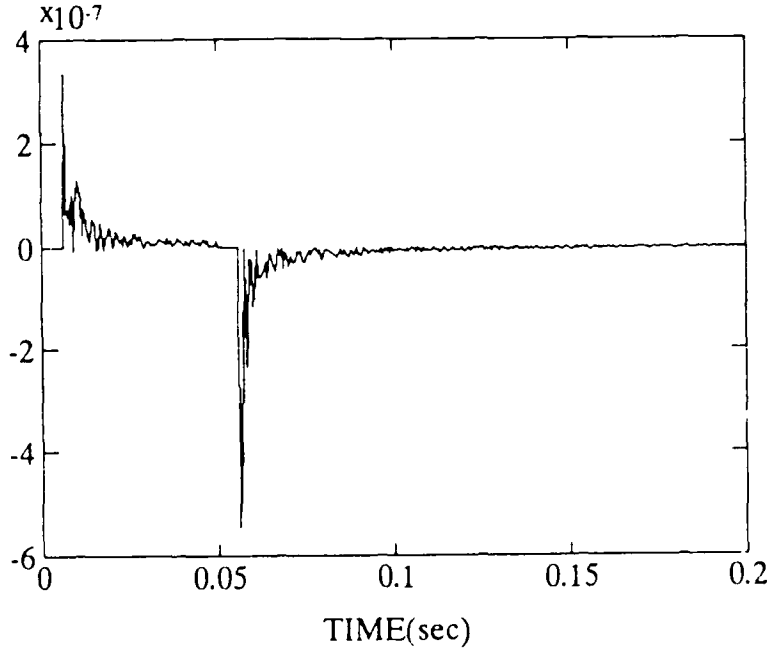



**Figure 6.15: Mean of Error**

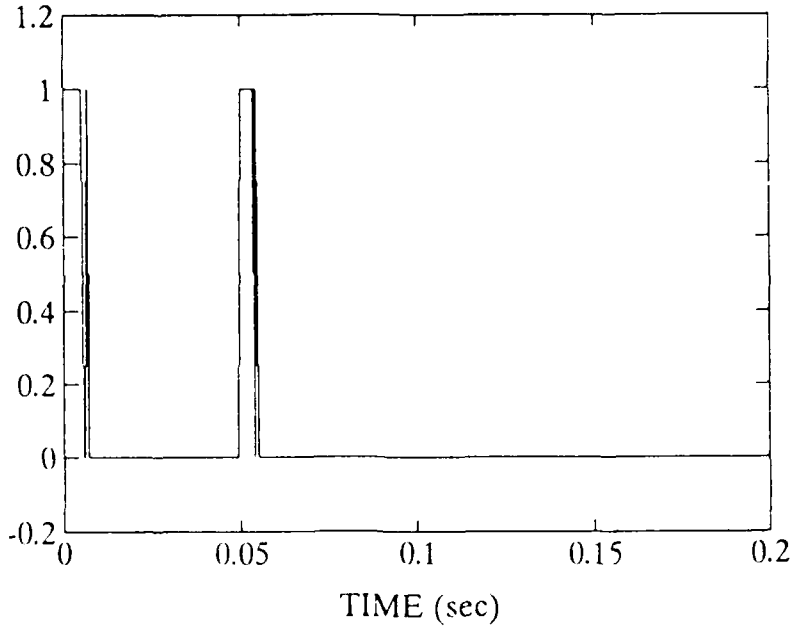



Figure 6.16: System Lock-On Time

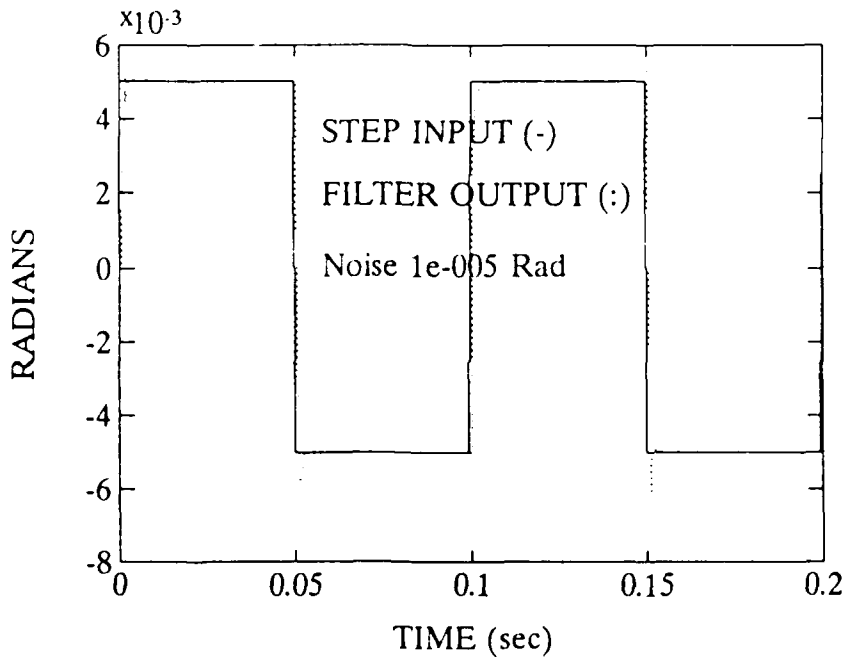



Figure 6.17: $X1$ Estimation for Model vs. Input

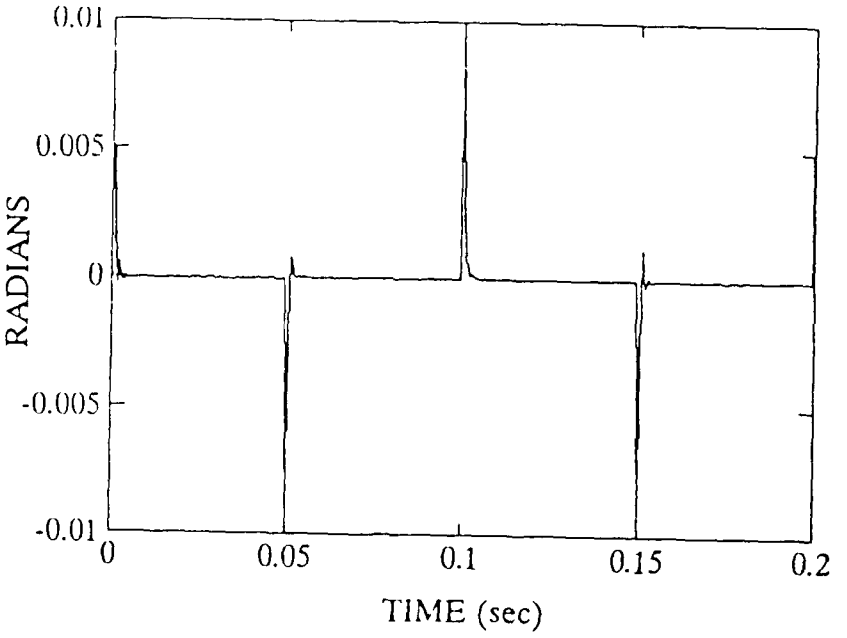



Figure 6.18: Plot of Error Between Estimate and Input ($X1$)

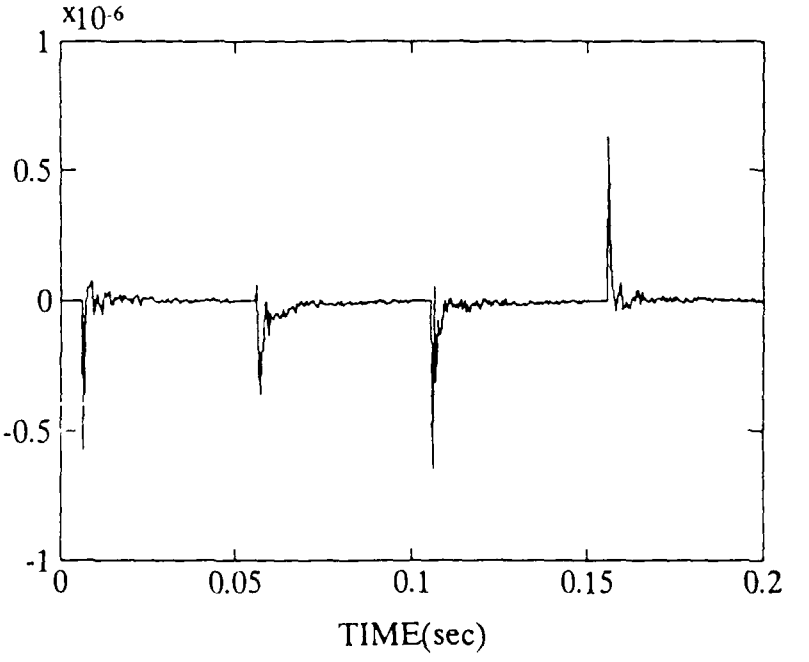



Figure 6.19: Mean of Error

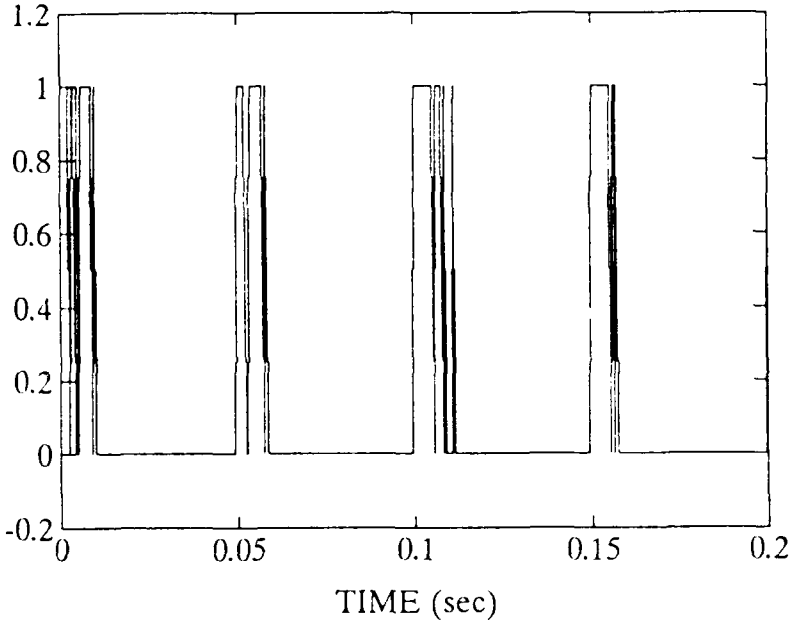



**Figure 6.20: System Lock-On Time**

6.21 shows the Kalman trying to track the input. The error graph, Figure 6.22, shows the output gets close to the input very rapidly, but does not lock-on, Figure 6.24, and stay there. Partial lock-on is achieved, but with the input stepping every 20 time steps, the Kalman has great difficulty getting the covariance matrix and gains down. There appears to be a credible performance by the Kalman at this point, but it is pushing its abilities with the present specified sample rate of 2 kHz.

#### 4. Scenario Four

A 100 Hz, 5 mrad square wave was input into the Kalman. This was the specified range for the blending filter given by Reference 1. Figures 6.25 and 6.28 show the Kalman's inability to follow an input of this high of a frequency. A 100 Hz wave calls for a step up or down every 10 time steps. In other words, the input goes through two complete periods in the required lock-on time of 20 msec. As with Scenario Three, a much higher sample time is needed for the Kalman to track this

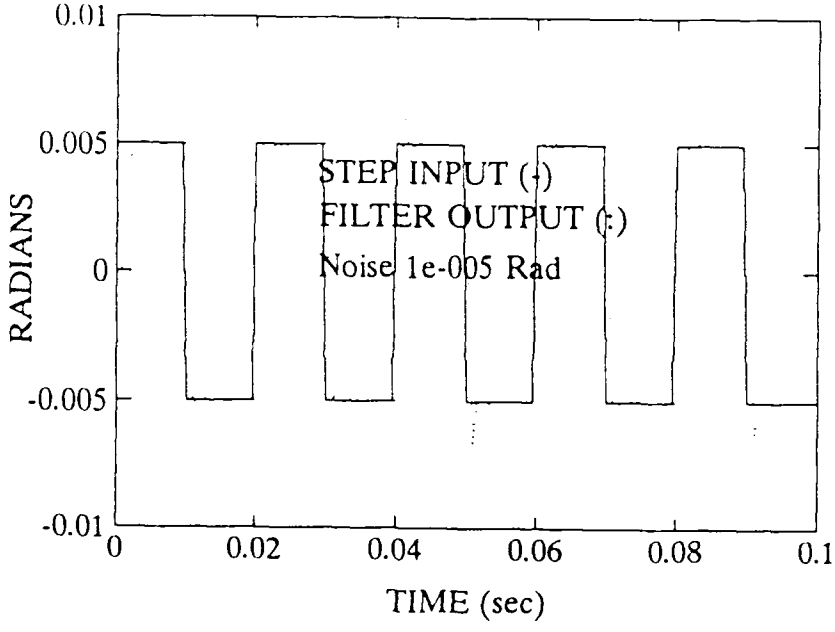



Figure 6.21: $X1$ Estimation for Model vs. Input

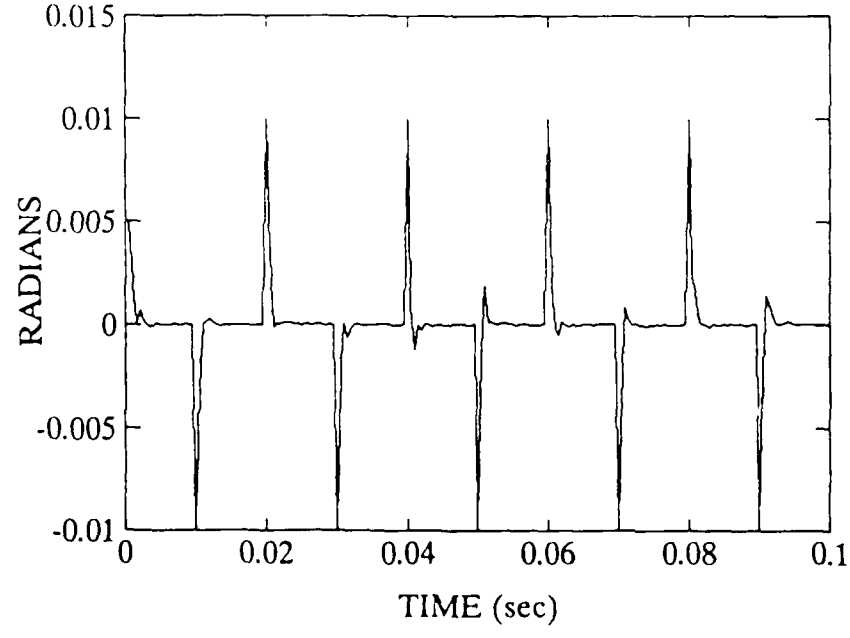



Figure 6.22: Plot of Error Between Estimate and Input ($X1$)

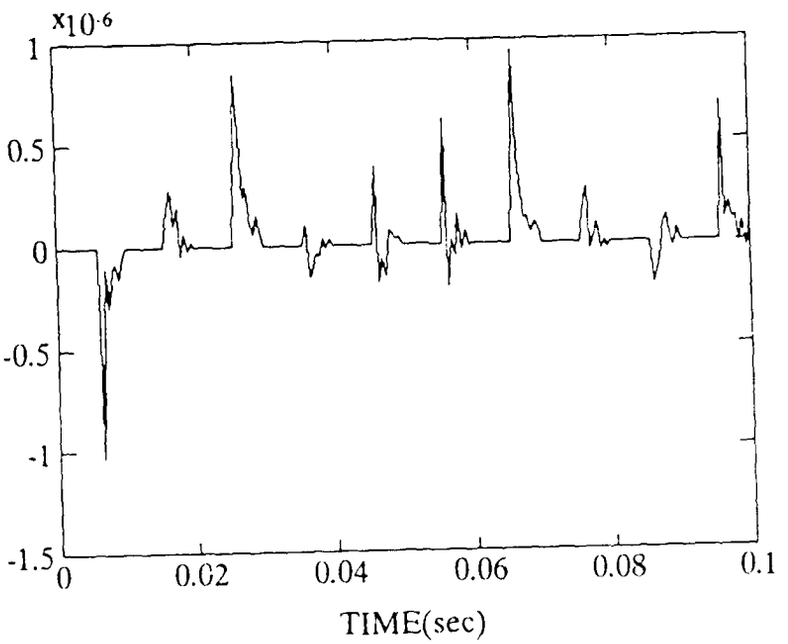



**Figure 6.23: Mean of Error**

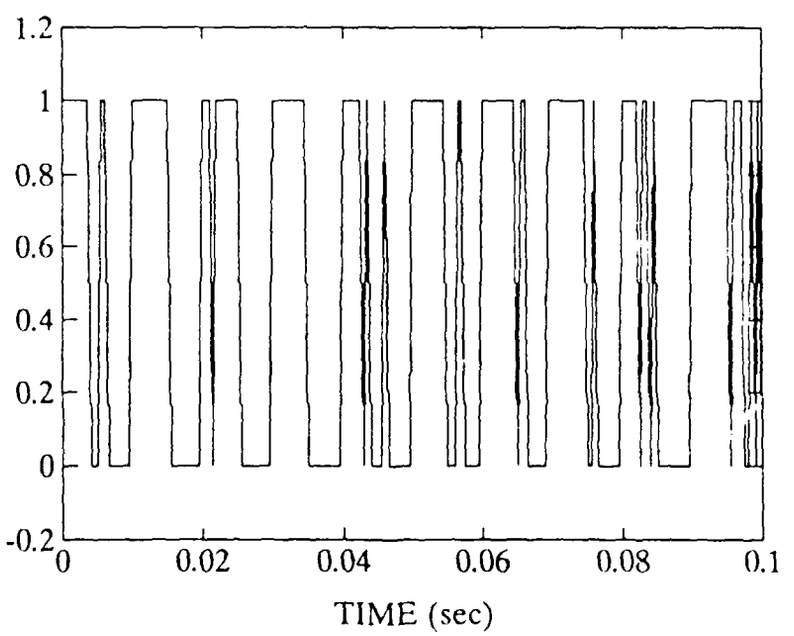



**Figure 6.24: System Lock-On Time**

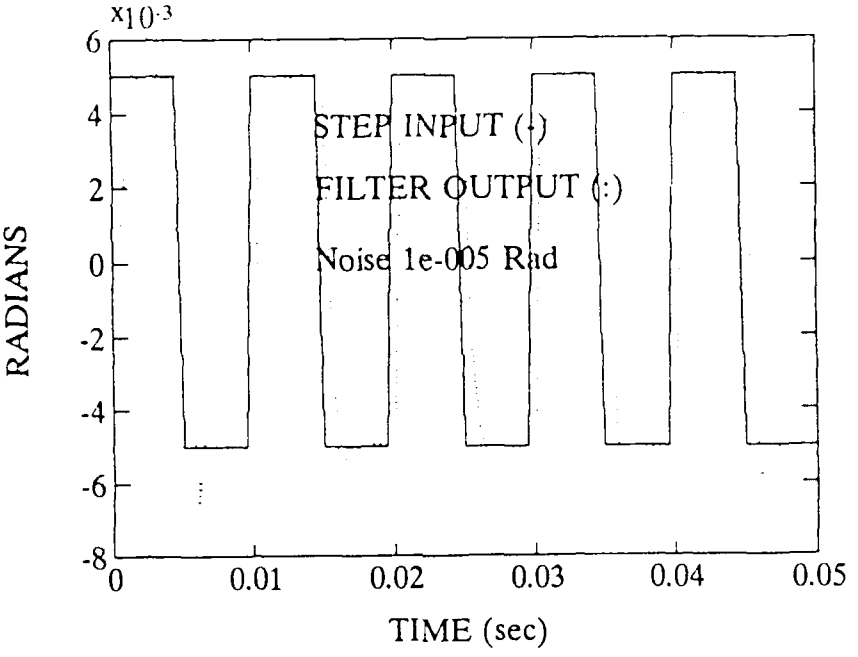



**Figure 6.25: $X1$ Estimation for Model vs. Input**

frequency of input. The zero mean error, Figure 6.27, results from how the mean is computed in the program. The program allows for a settling time of 13 time steps after gating. The 100 Hz wave causes the filter to gate every 10 time steps. The mean cannot be computed and remains zero.

## D. NOISE VARIATIONS

In order to verify the filter's insensitivity to noise, Scenario Two was modified with various levels of state noise and measurement noise.

The first simulation decreased the values in the $Q$ matrix by an order of magnitude. As shown in Figures 6.29 through 6.32, this had little or no effect on the outputs when compared to Figures 6.17 to 6.20.

The next three runs involved varying the $R$ matrix. The $R$ matrix was the noise input most easily influenced in the system. Variations in electrical current to the

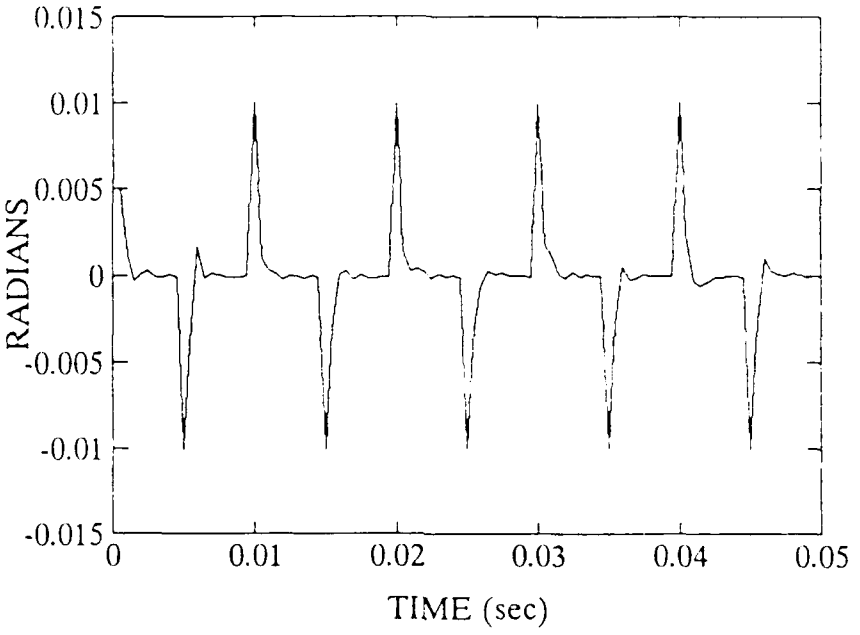



Figure 6.26: Plot of Error Between Estimate and Input ($X1$)

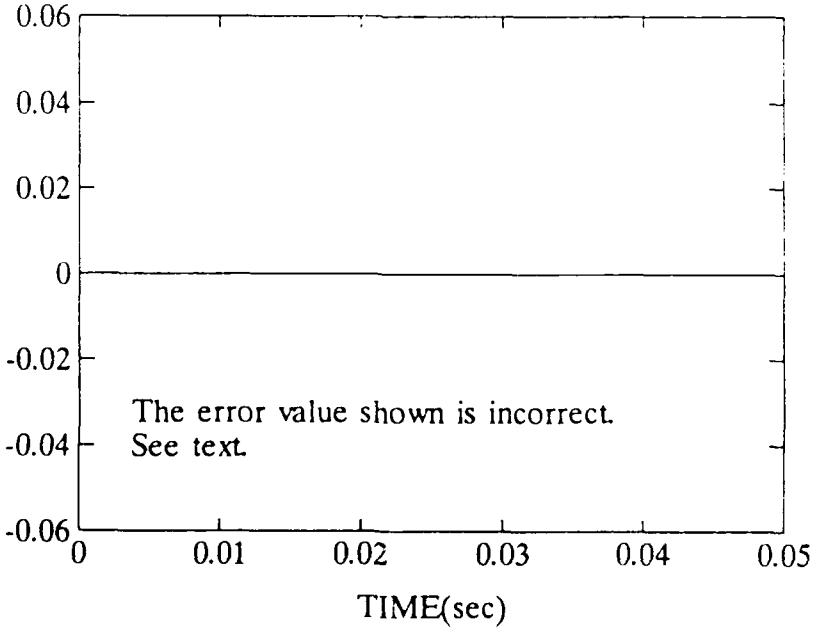



Figure 6.27: Mean of Error

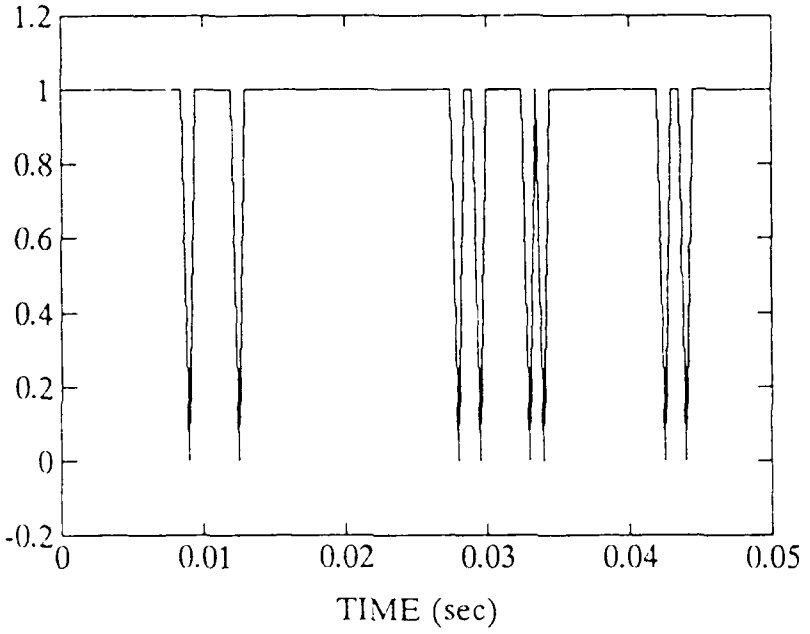



**Figure 6.28: System Lock-On Time**

sensors or misalignment of components are two ways that could increase the sensor noise. To increase the values in the $Q$ matrix would require a failure somewhere in the R2P2 damping mechanisms or, in real life, an impact on the structure in space.

Therefore, the next three simulations involved increasing the magnitude of the noise elements $V_G$ and $V_M$ of the $R$ matrix by factors of 2, 5, and 10. The resulting graphs are shown in Figures 6.33 through 6.44. The progression of the simulations show that the system can handle up to an order of magnitude increase in noise in both sensors and still function. Figures 6.41 and 6.44 show that the factor of 10 increase does push the system to the limits of its desired capabilities. Figures 6.33 through 6.40 show that the system performs quite well for 2 and 5 times the noise input.

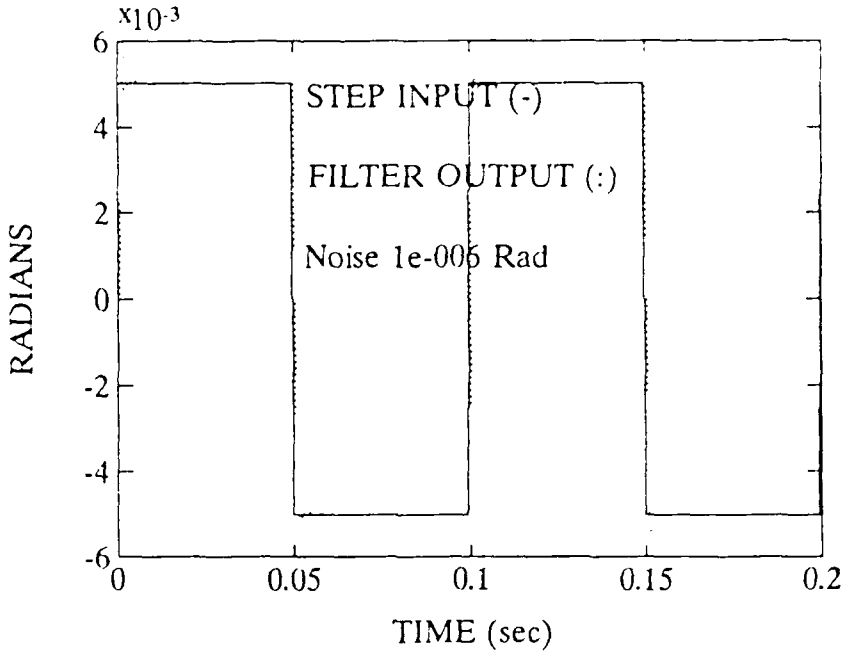



**Figure 6.29: $X1$ Estimation for Model vs. Input**

## E. FREQUENCY RESPONSE

The final portion of the analysis of the Kalman filtered system was the frequency response. As stated earlier, a flat response over the interval 0.01-100 Hz was desired. The frequency response of Martin Marietta's classical blending system, Figure 6.45, is shown in Figure 6.46. The frequency response for the steady state gain Kalman filter is shown in Figure 6.47. The Kalman's frequency response for the steady-state gains does not meet specifications. Due to the adaptive gating designed into the Kalman filter, it will not reach the steady-state gain values utilized in the Wiener development under normal conditions. With any kind of input, the gains will be adapting continually. The steady-state Kalman approximation meets the required error requirements. The adapting that occurs increases the bandwidth to the desired range, thereby fulfilling the requirements.

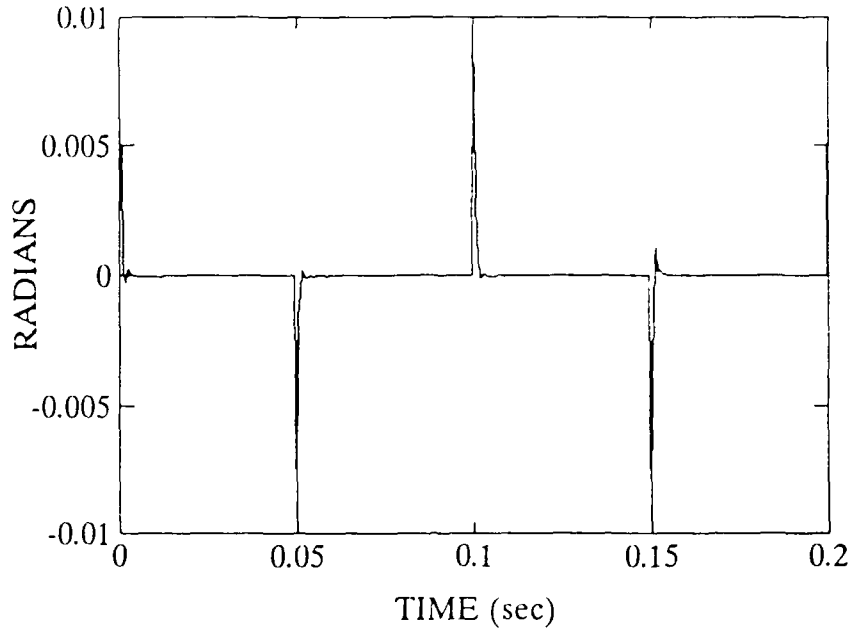



**Figure 6.30: Plot of Error Between Estimate and Input ($X1$)**

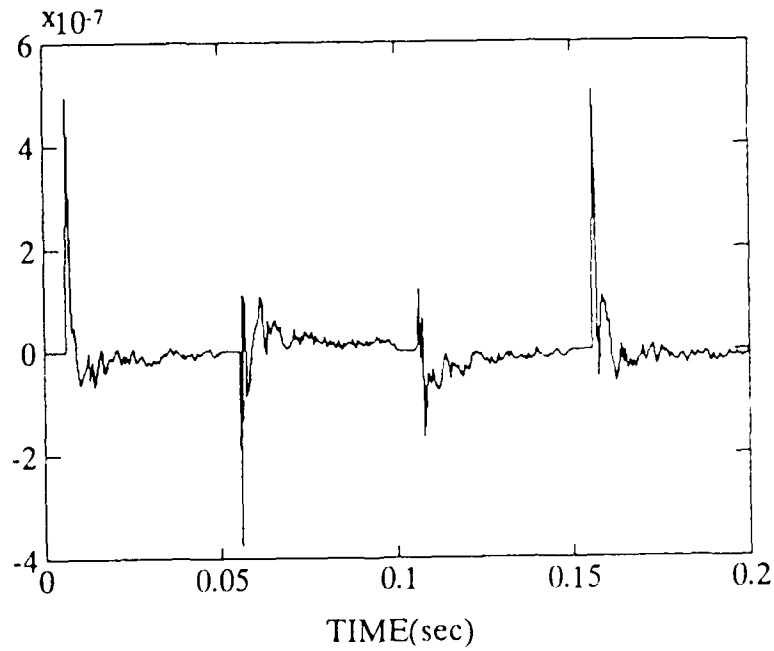



**Figure 6.31: Mean of Error**

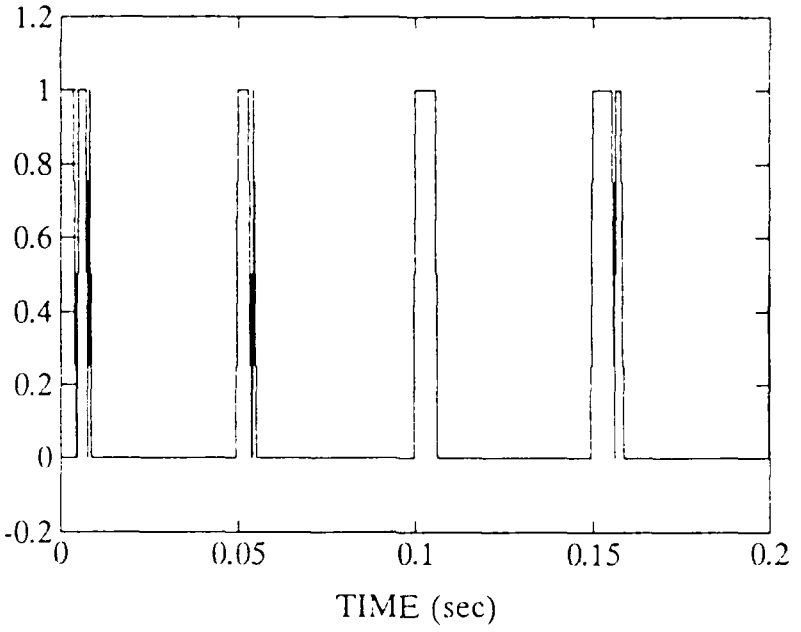



Figure 6.32: System Lock-On Time

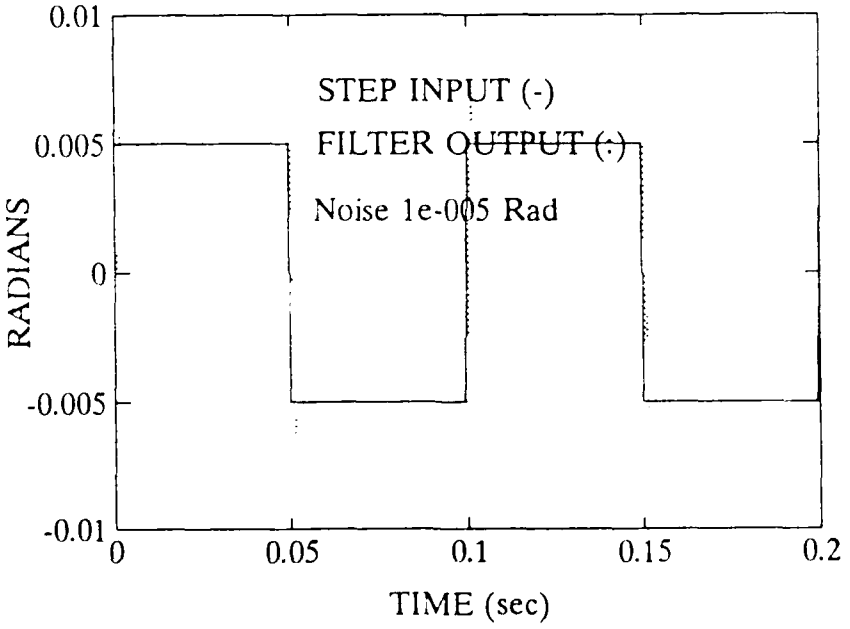



Figure 6.33: $X1$ Estimation for Model vs. Input

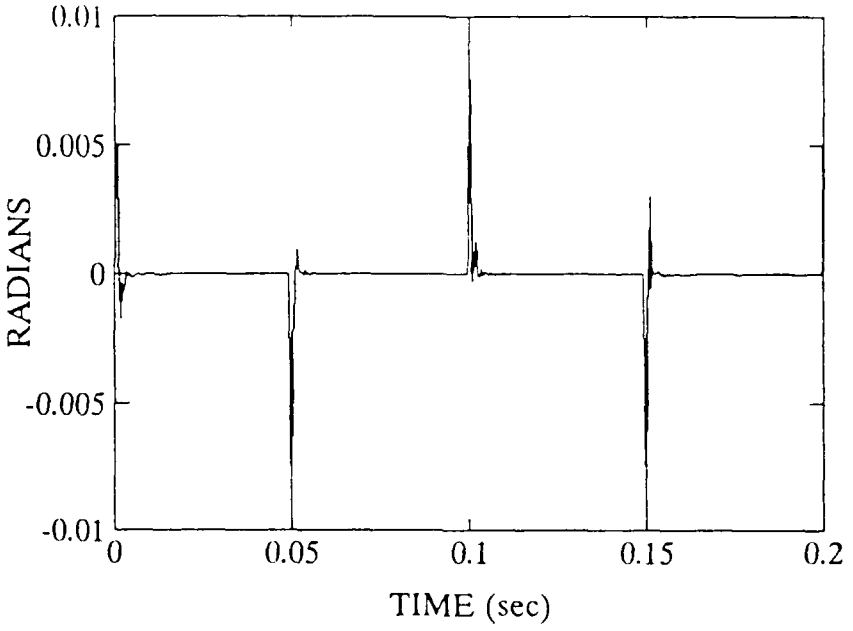



Figure 6.34: Plot of Error Between Estimate and Input ($X1$)

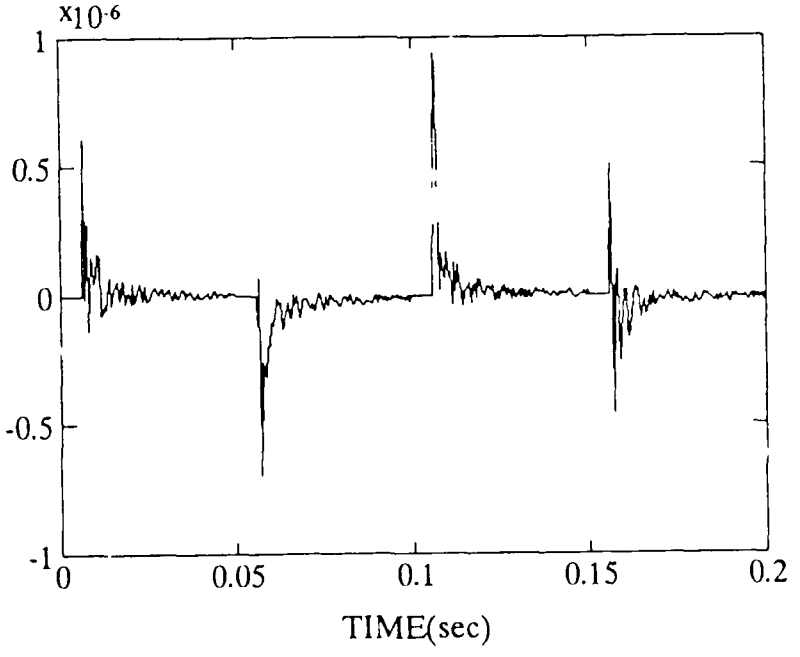



Figure 6.35: Mean Of Error

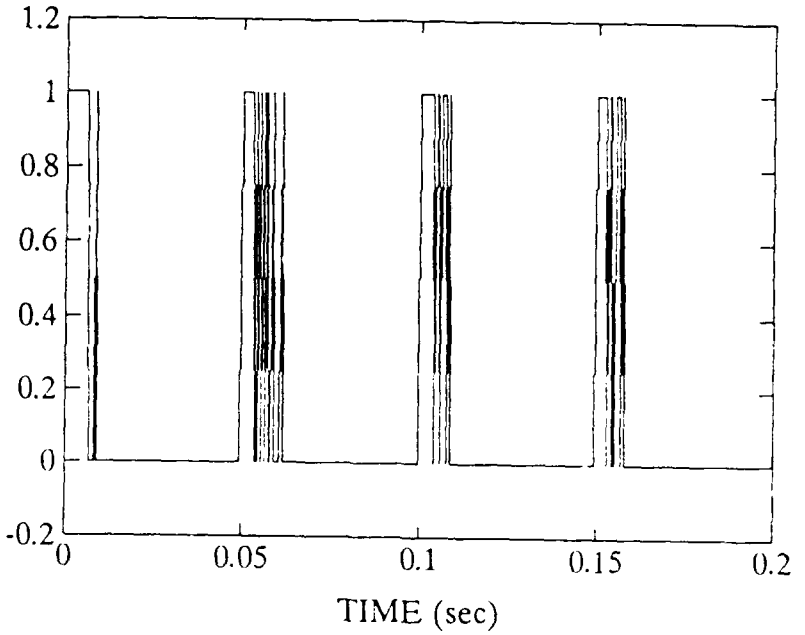



Figure 6.36: System Lock-On Time

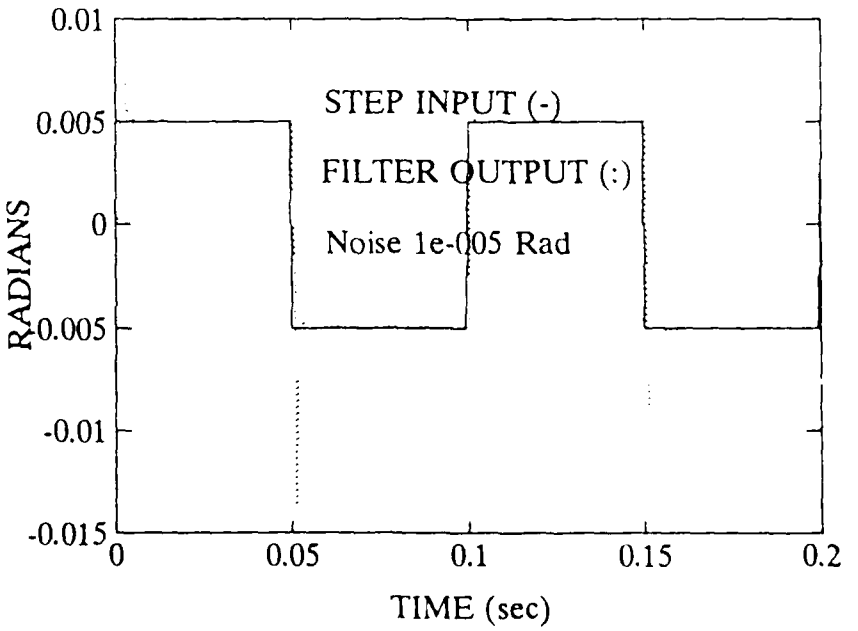



Figure 6.37: $X1$ Estimation for Model vs. Input

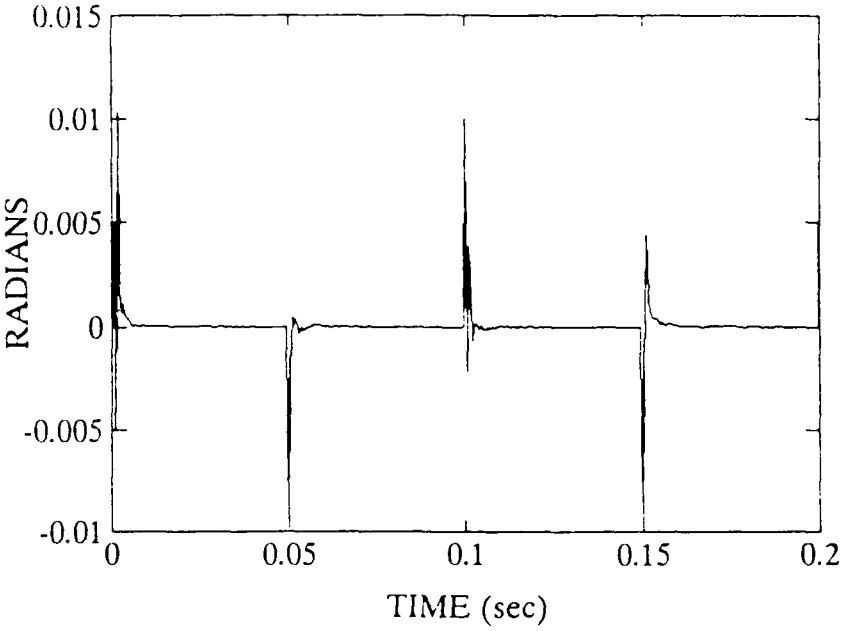



Figure 6.38: Plot of Error Between Estimate and Input ($X1$)

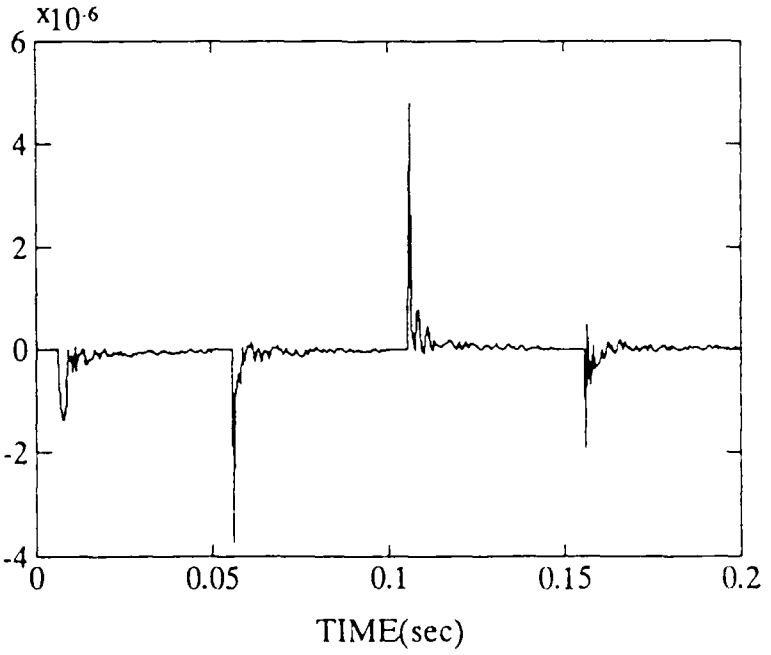



Figure 6.39: Mean of Error

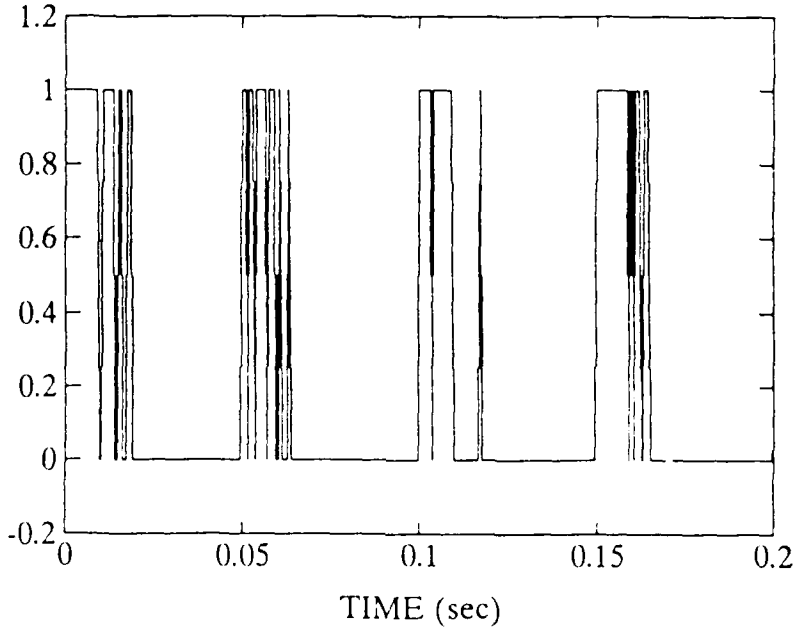



Figure 6.40: System Lock-On Time

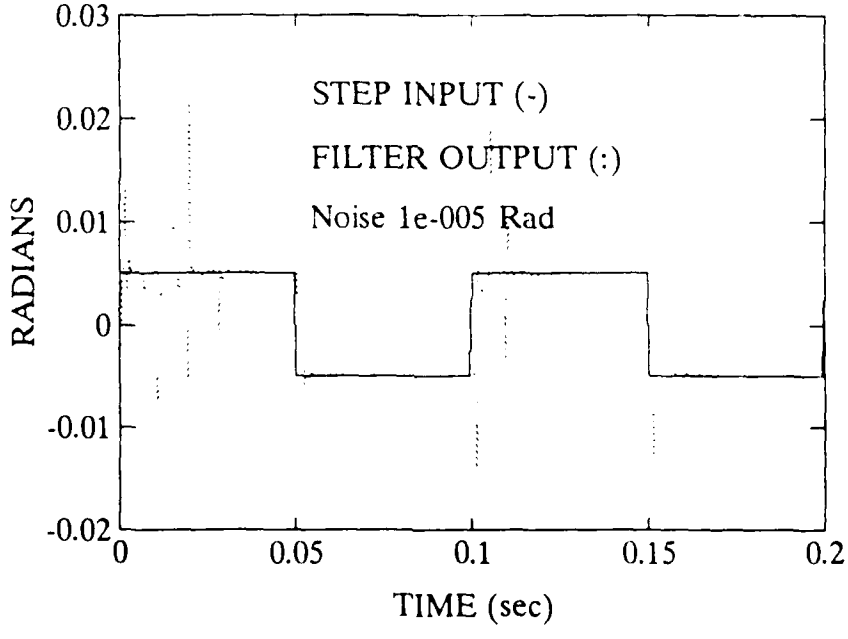



Figure 6.41: $X1$ Estimation for Model vs. Input

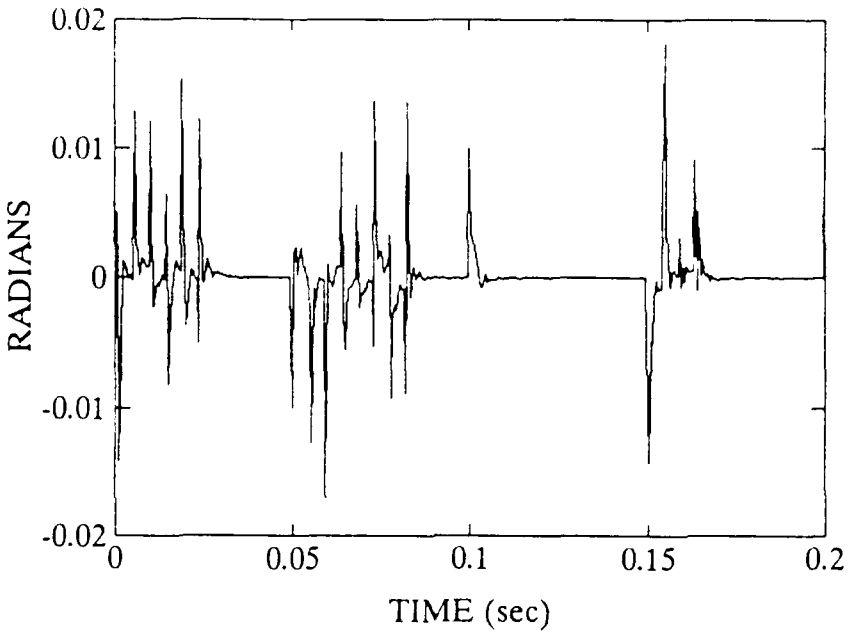



Figure 6.42: Plot of Error Between Estimate and Input ($X1$)

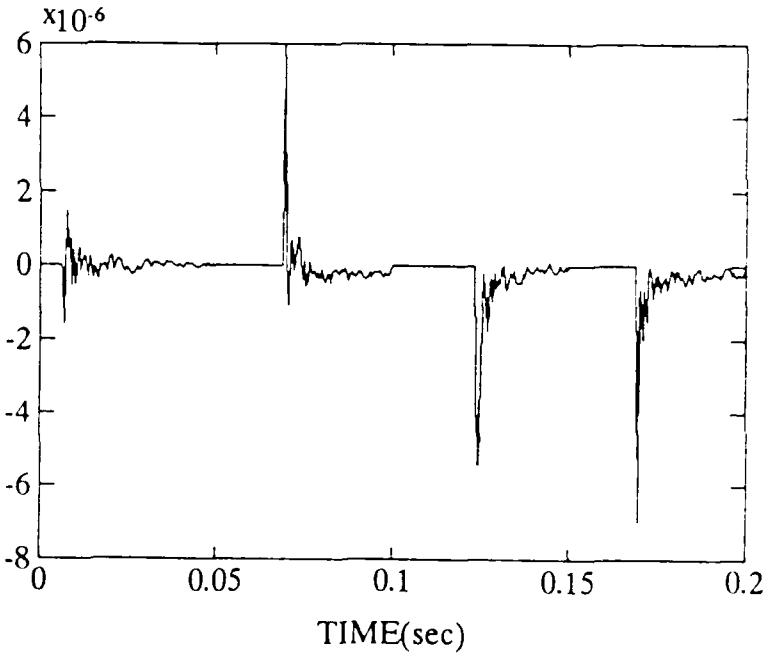



Figure 6.43: Mean of Error

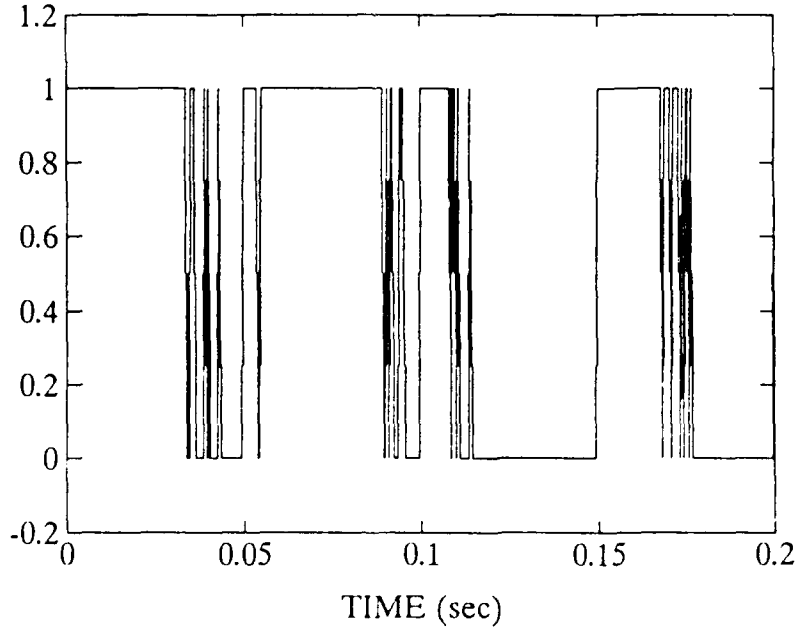



**Figure 6.44: System Lock-On Time**

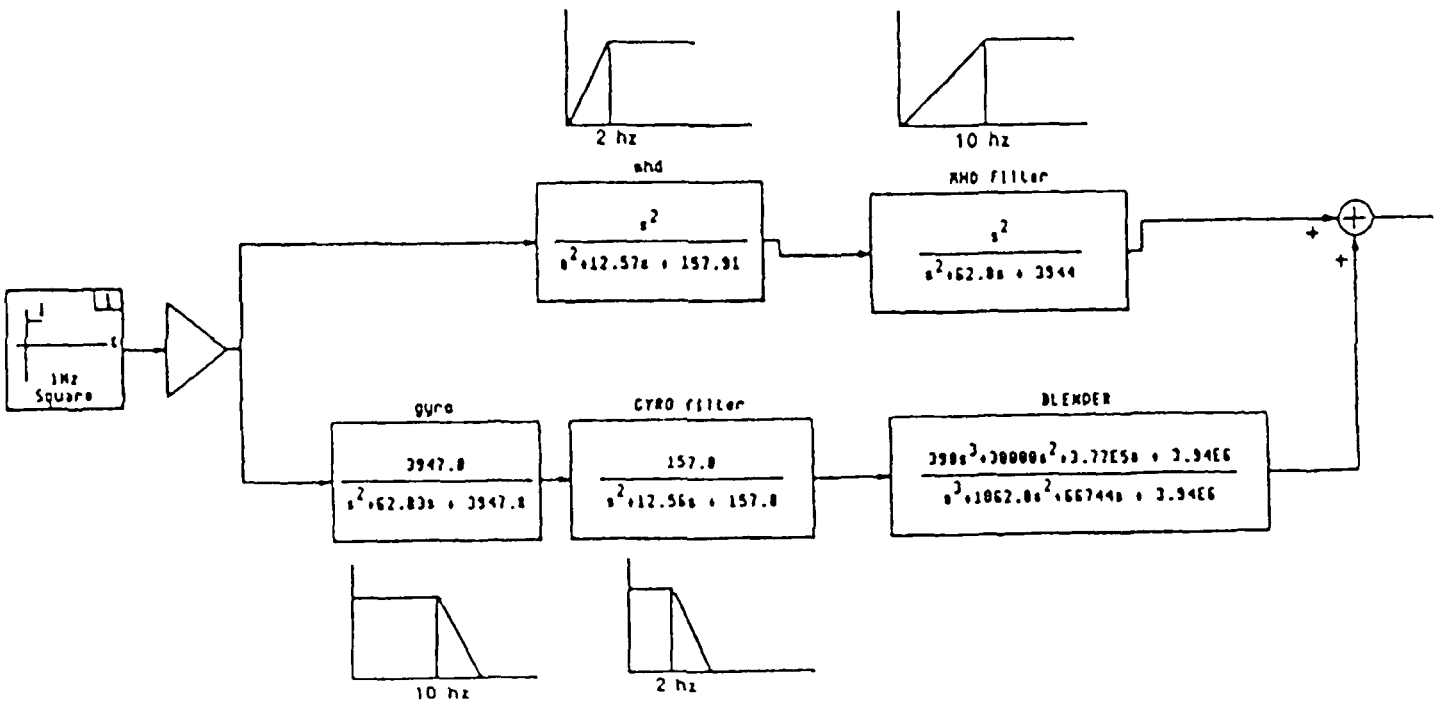



Figure 6.45: Classical Blending System

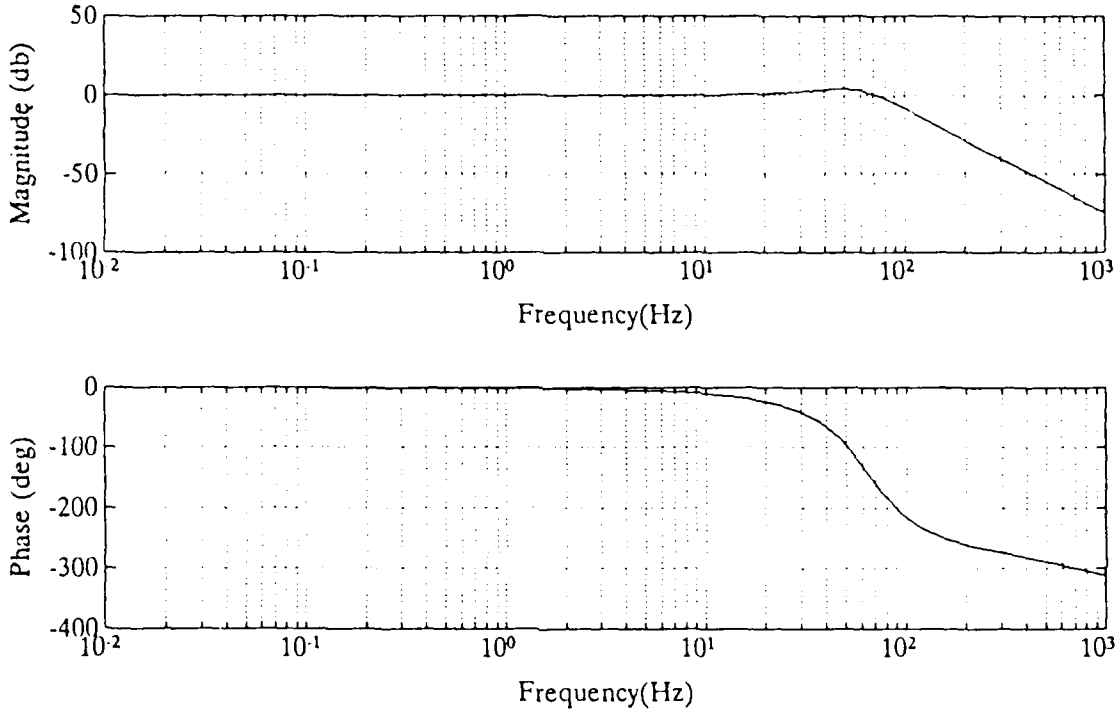



Figure 6.46: MMAG Blending Scheme

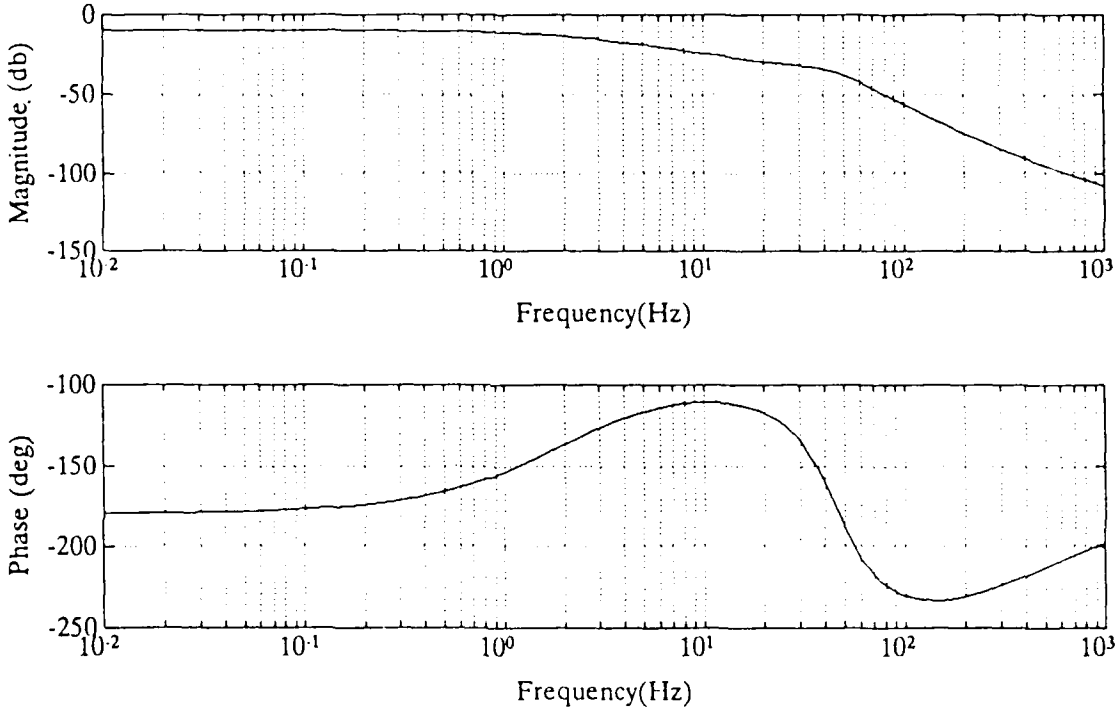



**Figure 6.47: KALMAN Blending Scheme (State $X1$)**

# VII. CONCLUSION

The Bode diagram for the steady-state Kalman filter clearly shows that a steady-state gain filter does not meet the bandwidth requirements for the blender. If a steady-state Kalman filter had been used, the blending scheme proposed would not have functioned properly. But with an adaptive gate Kalman filter, the signal blender achieves the desired bandwidth. This is shown in the various simulations conducted. The purpose of an adaptive Kalman is to adapt the bandwidth of the system it is estimating. The Bode shown is just an approximation of the Kalman filter developed. It is a snapshot at a point in time of the adaptive filter. Developing a frequency response for an adaptive Kalman filter is a possibility for further research.

For speed and accuracy, the Kalman is vastly superior to the classical blending scheme. Figure 7.1 shows the results of a 1 Hz square wave input into the Martin Marietta system. The results from Scenario One are orders of magnitude better.

The adaptive gating approach used in this design is very versatile in its application. Since time response was a high priority, this versatility was sacrificed to a degree. By adjusting the gate and factor, $F$, the Kalman filter can be adapted to follow any transient input. But, the faster the adaptation, the poorer the noise filtering the transient.

Overall, the Kalman filter is superior to the classical approach to blending two signals. For speed and accuracy, it is orders of magnitude better. With a few modifications, it can be made to follow any input.

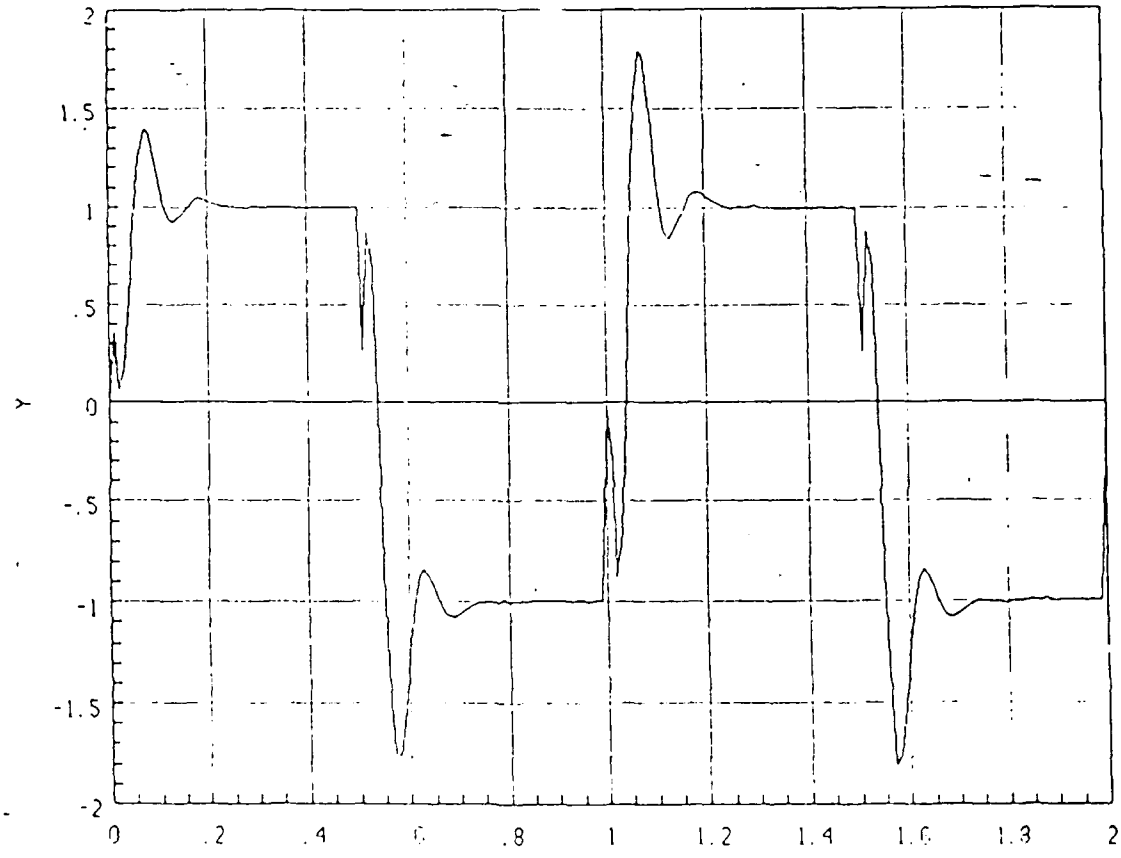



Figure 7.1: Transient Response - Classical Blending

# APPENDIX A: MAIN PROGRAM AND INPUT FILES

All of the simulations for this project were run on IBM-PC class computers using the matrix manipulation language MATLAB, version 3.5f. This appendix contains the source code for all of the functions written in support of this project.

Only minor programming experience is required to understand these files. While MATLAB is similar to Fortran, MATLAB's control structures are much less complex. Comments are started by the percent sign (%) and continue to the end of the line.

To aid the reader in scanning and retyping these functions, each file is started on a new page. Although an analysis of the workings of these files is not necessary to understand this report, the curious (or skeptical) reader is highly encouraged to examine them closely.

The author neither claims nor desires to hold any copyright privileges on the source code. Written requests for the source code on computer disk should be sent either to the author or to Professor Harold A. Titus. Address information can be found in the Initial Distribution List at the end of this report.

All of the files listed in the second section of this appendix provide general support for the main files listed in the first section. These support files are not specific to the simulations run for this report, but can be used for a variety of purposes.

```
%
%
%  ######################################################################
%  #                                                                    #
%  #         THESIST5.M                    30 JAN 90                    #
%  #                                                                    #
%  #   MATLAB Simulation of the Beam Expander Inertial Reference        #
%  #   Unit for the Rapid Retargeting/Precision Pointing (R2/P2)        #
%  #   System                                                           #
%  #     Before running this simulation the length for the              #
%  #   simulation in seconds must be defined as the variable            #
%  #   'kmax', and the sampling interval is defined as 'dt'.            #
%  #      The program uses a adaptive gate Kalman filter to             #
%  #   blend the output of two different sensors. The sensors           #
%  #   are a MagnetoHyrodynamic rate sensor and Singer Rate             #
%  #   Gyroscope.                                                       #
%  #     This  program tracks square  waves of  different               #
%  #   frequencies.                                                     #
%  #                                                                    #
%  ######################################################################
%
%
%
%
delete bx.met                              % delete previous meta file
delete output6.met
clear;
clg;

%  *******      INERTIAL REFERENCE UNIT     *****
%
%
% The IRU is made up of two sensors. A strap down gyro and a
% Magneto HydroDynamic rate sensor.
%
%       ****    Input Constants          ****

b = 1000*2*pi;                             %end break freq
wg = 10.0*2*pi;                            %gyro break freq down
wm = 2.0*2*pi;                             %MHD break freq up
zeta = 1.0;                                %damping ratio
dt = 0.0005;                               %sample rate for system
kmax = 0.20/dt+1;
I = eye(5);
date= 26;
%Wg = 1e-5;                                 % State noise for gyro
Wg = 0;
%Wm = 1e-5;
Wm = 0;

%   ******      Input State Matrices For Sensors        *****

AS = zeros(5,5);                           %initialize matrix at zero
BS = [0;0;Wg;0;Wm];                        %B matrix
ZS = zeros(2,5);                           %initialize observer matrix

%       ***     Enter Values Into A Matrix      ***
%
AS(1,1) = 0;
AS(2,3) = 1;
```

```
AS(3,1) = wg^2;
AS(3,2) = -wg^2;
AS(3,3) = -2*zeta*wg;
AS(4,5) = 1;
AS(5,1) = wm^2;
AS(5,4) = -wm^2;
AS(5,5) = -2*zeta*wm;

%
%       ***     Enter Values Into Observer Matrix       ***
%
ZS(1,2) = 1;
ZS(2,4) = 1;
ZS(2,5) = .5;
H = ZS ;

%
%       **      Build Observer Matrix   **
%
%
%       ***     Discretize State Equations              ***
%
[phi,del] = c2d(AS,BS,dt);          %Discretize states
%
%      ****     Construct Kalman Filter Equations       ****
%
%  **   Build State Noise Error Matrix

w1 = 0;
w2 = 1e-2;
w3 = 5e-5;
w4 = 1e-6;
w5 = 1e-5;

Q = w1^2*eye(5);
q = 0;%1e-5;
%
%
%   **  Build Measurement Noise Matrix  **

avg =.1.237e-6;
avm = 75.0e-6;
vg = 1e-10*avg;
vm = 1e-10*avm;
v = [vg vm]';
R = zeros(2);                           %R matrix values
R(1,1) = vg^2;
R(2,2) = vm^2;

%
%  **   Set Initial Error Covariance Matrix  **
%
P = 1e10*eye(5);
P0 = P;
%
%
%       ****  Build Kalman Equations    ****
%

y = zeros(2,kmax);
```

```
m = zeros(5,kmax);
tsql;
%
for i =1:kmax,
  y(:,i)= H*f(:,i)+v.*rand(v);
end
%
time = zeros(1,kmax);
time(1) = 0;
mr = zeros(2,kmax);
Xhat = zeros(5,kmax);
Xhat(:,1) = [0 0 0 0 0]';                    %Initial estimate of states
k_wait = 0;
mean_r = [0;0];                              % mean of the residual

for k=2:kmax;
   Xhat(:,k) = phi*Xhat(:,k-1)+del;          %X(k+1/k)
   while 1
      resid = y(:,k)-H*Xhat(:,k-1);
      vresid = H*P*H'+R;                     %variance of the residual
      md2 = resid'/vresid*resid;             %Mahalanobis distance
      if md2 < 4,                            %gating check
         break;
      else
         P = P0;                             %P(k/k)
         k_wait = 0;
         mean_r = 0*mean_r;
         mr(:,k) = mean_r;
      end
   end
   k_wait = k_wait + 1;
   G = P*(H)'/vresid;                        %Kalman Gains G(k+1)
   P = (I-G*H)*P;                            %P(k/k)
   P = phi*P*(phi)'+Q;                       %P(k+1/k)
   Xhat(:,k)=Xhat(:,k)+G*(y(:,k)-H*Xhat(:,k));    %X(k+1/k+1)
   if k_wait >= 13
      kw = k_wait - 13;
      mean_r = kw/(kw+1)*mean_r + 1/(kw+1)*resid;
      mr(:,k) = mean_r;
   end
   time(k)=time(k-1)+dt;
   home,k
end

%
%       ***  Calculate the Error  ***
%
error = m(1,:)-Xhat(1,:);
errg = m(1,:)-Xhat(2,:);
errm = m(1,:)-Xhat(4,:);

%
%                **  Lock on Check   **
%
for j=1:kmax,
  if abs(error(j))<= 2e-5,
     lo(j) = 0;
  else
     lo(j) = 1.0;
  end;
```

```
end;
subplot(211),plot(time,m(1,:),time,Xhat(1,:),':')
title('X1 ESTIMATION FOR MODEL VS. INPUT')
xlabel('TIME (sec)'),ylabel('RADIANS')
gtext('STEP INPUT (-) ')
gtext('FILTER OUTPUT (:) ')
gtext(['Noise ',num2str(q),' Rad'])
subplot(212),plot(time,error),title('PLOT OF ERROR BETWEEN ESTIMATE AND INPUT (X
xlabel('TIME (sec)'),ylabel('RADIANS')
meta output2
pause
%plot(time,m(1,:),time,Xhat(2,:),':')
%title('XHAT 2 INPUT')
%plot(time,errg),title('PLOT OF ERROR BETWEEN ESTIMATE AND INPUT X2')
%xlabel('TIME (sec)'),ylabel('RADIANS')
%meta
%pause
%plot(time,m(1,:),time,Xhat(4,:),':')
%title('XHAT 4 INPUT')
%plot(time,errm),title('PLOT OF ERROR BETWEEN ESTIMATE AND INPUT X4')
%xlabel('TIME (sec)'),ylabel('RADIANS')
%meta
plot(time,mr(1,:)),title('MEAN OF ERROR'),xlabel('TIME')
axis([0 .20 -0.2 1.2]);
plot(time,lo),title('SYSTEM LOCK ON TIME')
xlabel('TIME (sec)')
meta
pause
clg;
axis([0 .20 -5e-5 5e-5]);
plot(time,error),title('PLOT OF ERROR BETWEEN ESTIMATE AND INPUT (X1)')
xlabel  TIME (sec)'),ylabel('RADIANS')
meta
axis;
```

```
%
%
%  ##########################################################################
%  #                                                                        #
%  #          TSQ1.M                          25 JAN 90                     #
%  #                                                                        #
%  #    MATLAB Simulation of the Beam Expander Inertial Reference           #
%  #    Unit for the Rapid Retargeting/Precision Pointing (R2/P2)           #
%  #    System                                                              #
%  #      Before running this simulation the length for the                 #
%  #    simulation in seconds must be defined as the variable               #
%  #    'tmax', the sampling interval is defined as 'dt'.                   #
%  #      This program generates a step to be input into                    #
%  #    THESIST5.M.                                                         #
%  #                                                                        #
%  ##########################################################################
%
%
%
%
dt=0.0005;                                  %sample rate for system
kmax=0.20/dt+1;
%
%
%       ****  Build Square Wave Input  ****
y = zeros(2,kmax);
m = zeros(5,kmax);

for i= 1:kmax;
 if i<=100,
     state = [.005 .005 0 .005 0]';
   else
    state = [0 0 0 0 0]';
 end
  f(:,i)= state;
  m(:,i) = state;
end
```

```
%
%
%  ######################################################################
%  #                                                                    #
%  #          TSQ10.M                        25 JAN 90                  #
%  #                                                                    #
%  #    MATLAB Simulation of the Beam Expander Inertial Reference       #
%  #    Unit for the Rapid Retargeting/Precision Pointing (R2/P2)       #
%  #    System                                                          #
%  #      Before running this simulation the length for the             #
%  #    simulation in seconds must be defined as the variable           #
%  #    'tmax', the sampling interval is defined as 'dt'.               #
%  #      This program generates a  10 Hz square wave to be             #
%  #    input into THESIST.M.                                           #
%  #                                                                    #
%  ######################################################################
%
%
%
%
dt=0.0005;                                  %sample rate for system
kmax=0.50/dt+1;
%
%
%       ****  Build Square Wave Input  ****
y = zeros(2,kmax);
m = zeros(5,kmax);

for i= 1:kmax;
  if i<=100,
     state = [.005 .005 0 .005 0]';
   elseif i<=200
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=300
     state = [.005 .005 0 .005 0]';
   elseif i<=400
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=500
     state= [.005 .005 0 .005 0]';
  elseif i<=600
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=700
     state= [.005 .005 0 .005 0]';
  elseif i<=800
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=900
     state= [.005 .005 0 .005 0]';
  else
     state = -1*[.005 .005 0 .005 0]';
  end
  f(:,i)= state;
  m(:,i) = state;
end
```

```
%
%
%  ##########################################################################
%  #                                                                        #
%  #          TSQ50.M                        25 JAN 90                      #
%  #                                                                        #
%  #    MATLAB Simulation of the Beam Expander Inertial Reference           #
%  #    Unit for the Rapid Retargeting/Precision Pointing (R2/P2)           #
%  #    System                                                              #
%  #      Before running this simulation the length for the                 #
%  #    simulation in seconds must be defined as the variable               #
%  #    'tmax', the sampling interval is defined as 'dt'.                   #
%  #      This program generates a  50 Hz square wave to be                 #
%  #    input into THESIST.M.                                               #
%  #                                                                        #
%  ##########################################################################
%
%
%
%
dt=0.0005;                                %sample rate for system
kmax=0.10/dt+1;
%
%
%       ****  Build Square Wave Input  ****
y = zeros(2,kmax);
m = zeros(5,kmax);

for i= 1:kmax;
  if i<=20,
     state = [.005 .005 0 .005 0]';
   elseif i<=40
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=60
     state = [.005 .005 0 .005 0]';
   elseif i<=80
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=100
     state= [.005 .005 0 .005 0]';
  elseif i<=120
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=140
     state= [.005 .005 0 .005 0]';
  elseif i<=160
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=180
     state= [.005 .005 0 .005 0]';
  else
     state = -1*[.005 .005 0 .005 0]';
  end
  f(:,i)= state;
  m(:,i) = state;
end
```

```
%
%
%  ##########################################################################
%  #                                                                        #
%  #         TSQ100.M                    25 JAN 90                          #
%  #                                                                        #
%  #   MATLAB Simulation of the Beam Expander Inertial Reference            #
%  #   Unit for the Rapid Retargeting/Precision Pointing (R2/P2)            #
%  #   System                                                               #
%  #     Before running this simulation the length for the                  #
%  #   simulation in seconds must be defined as the variable                #
%  #   'tmax', the sampling interval is defined as 'dt'.                    #
%  #     This program generates a 100 Hz square wave to be                  #
%  #   input into THESIST.M.                                                #
%  #                                                                        #
%  ##########################################################################
%
%
%
%
dt=0.0005;                                  %sample rate for system
kmax=0.05/dt+1;
%
%
%       ****  Build Square Wave Input  ****
y = zeros(2,kmax);
m = zeros(5,kmax);

for i= 1:kmax;
  if i<=10,
     state = [.005 .005 0 .005 0]';
   elseif i<=20
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=30
     state = [.005 .005 0 .005 0]';
   elseif i<=40
     state = -1*[.005 .005 0 .005 0]';
   elseif i<=50
     state= [.005 .005 0 .005 0]';
  elseif i<=60
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=70
     state= [.005 .005 0 .005 0]';
  elseif i<=80
     state = -1*[.005 .005 0 .005 0]';
  elseif i<=90
     state= [.005 .005 0 .005 0]';
  else
     state = -1*[.005 .005 0 .005 0]';
  end
  f(:,i)= state;
  m(:,i) = state;
end
```

# APPENDIX B: BODE PROGRAMS

This appendix contains the programs used to compute the Bode diagrams contained in the main body of the thesis.

```
%
%    THESBDE.M                                  1 NOV 89
%
%    MATLAB simulation of Kalman Filter signal blending
%   scheme. Developed by the Terry J Bauer, CPT USA, for
%   the R2P2 fine tracking system.
%
%
!delete thbode.met
!delete tbode.met
%
%   Enter the Transfer Functions For The Network
%
numm=[78.95 157.91];                        %MHD sensor
denm=[1 12.57 157.91];
numg=[0 3947.8];                            %Gyro sensor
deng=[1 62.83 3947.8];
w= logspace(-2,3);                  %frequency range
%
%


%
%       ***     Enter Kalman Values     ***
%
I = eye(5);
wg = 10*2*pi;                               %Gyro break freq
wm = 2*2*pi;                                %MHD break freq
dt = .0005;                                 %Sample ratye
A = [0 0 0 0 0;
     0 0 1 0 0;
     wg^2 -wg^2 -2*wg 0 0;
     0 0 0 0 1;
     wm^2 0 0 -wm^2 -2*wm];
B = [0 0 1e-5 0 1e-5]';
Q = (1e-5)^2*I;
R = [(1.237e-6)^2 0;0 (75e-6)^2];
H = [0 1 0 0 0;0 0 0 1 .5];
[phi,del] = c2d(A,B,dt);
L = [0 0 1 0 1]';
D = [0 0;0 0;0 0;0 0;0 0];

%
%       ***     Bode for Kalman         ***
%
%
G = dlqe(phi,1,H,Q,R);
[numk1,denk1] = ss2tf(phi+G*H,G,I,D,1);
[numk2,denk2] = ss2tf(phi+G*H,G,I,D,2);

%
%       **      Combine Transfer Functions      **
%

nut1 = conv(numg,numk1(1,:)) + conv(numm,numk2(1,:));
nut2 = conv(numg,numk1(2,:)) + conv(numm,numk2(2,:));
nut3 = conv(numg,numk1(3,:)) + conv(numm,numk2(3,:));
nut4 = conv(numg,numk1(4,:)) + conv(numm,numk2(4,:));
nut5 = conv(numg,numk1(5,:)) + conv(numm,numk2(5,:));
det = conv(deng,denk1) + conv(denm,denk2);
%nut1 = numk1(1,:);
```

```
%nut2 = numk1(2,:);
%nut3 = numk1(3,:);
%nut4 = numk1(4,:);
%nut5 = numk1(5,:);
%
%  Calculate Bode Frequency  Response
%
[mag1,phase1]= bode(nut1,det,w);
[mag2,phase2]= bode(nut2,det,w);
[mag3,phase3]= bode(nut3,det,w);
[mag4,phase4]= bode(nut4,det,w);
[mag5,phase5]= bode(nut5,det,w);
clg
semilogx(w,20*log10(mag1)),title('KALMAN Blending Scheme (State X1)')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta thbode
pause
%semilogx(w,phase1),title('KALMAN Blending Scheme (X1)')
%xlabel('Frequency'),ylabel('Phase (deg)'),grid
%pause
%meta
semilogx(w,20*log10(mag2)),title('KALMAN Blending Scheme (X2)')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta
pause
%semilogx(w,phase2),title('KALMAN Blending Scheme(X2)')
%xlabel('Frequency'),ylabel('Phase (deg)'),grid
%meta
%pause
semilogx(w,20*log10(mag3)),title('KALMAN Blending Scheme(X3)')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta
pause
%semilogx(w,phase3),title('KALMAN Blending Scheme(X3)')
%xlabel('Frequency'),ylabel('Phase (deg)'),grid
%meta
%pause
semilogx(w,20*log10(mag4)),title('KALMAN Blending Scheme(X4)')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta
pause.
%semilogx(w,phase4),title('KALMAN Blending Scheme (X4)')
%xlabel('Frequency'),ylabel('Phase (deg)'),grid
%meta
%pause
semilogx(w,20*log10(mag5)),title('KALMAN Blending Scheme(X5)')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta
%pause
%semilogx(w,phase5),title('KALMAN Blending Scheme (X5)')
%xlabel('Frequency'),ylabel('Phase (deg)'),grid
%meta
```

```
%
%    THESBDE.M                                1 NOV 89
%
%    MATLAB simulation of Martin Marietta's signal blending
%   scheme. Developed by the R2/P2 Control Group in Denver,
%   Colorado.
%
%
%   Enter the Transfer Functions For The Network And Blender
%
numm=[1 0 0];                       %MHD sensor
denm=[1 12.57 157.91];
nummf=[1 0 0];                      %MHD filter
denmf=[1 62.8 3944];
numg=[3947.8];                      %Gyro sensor
deng=[1 62.83 3947.8];
numgf=[157.8];                      %Gyro filter
dengf=[1 12.56 157.8];
numbl=[398 30000 3.77e5 3.94e6];          %Blender
denbl=[1 1062.8 66744 3.94e6];
w= logspace(-2,3);                  %frequency range
%
%   Combine Transfer Functions Along Branches
%
nummt=conv(numm,nummf) ;
denmt=conv(denm,denmf);
c1=conv(numg,numgf)    ;
c2=conv(deng,dengf)    ;
numgt=conv(c1,numbl)   ;
dengt=conv(c2,denbl);

ldif1= length(numgt) - length(nummt);
if ldif1 >=0
   nummt = [zeros(1,ldif1) nummt];
  else
    numgt =[ zeros(1,abs(ldif1)) numgt];
end
ldif2= length(dengt) - length(denmt);
if ldif2 >=0
   denmt = [zeros(1,ldif2) denmt];
  else
    dengt =[ zeros(1,abs(ldif2)) dengt];
 end


numeq=nummt +numgt;                       %sum of the sensors
deneq=denmt+dengt;

%
%  Calculate Bode Frequency  Response
%
[mag,phase]= bode(numeq,deneq,w);
clg
semilogx(w,20*log10(mag)),title('MMAG Blending Scheme')
xlabel('Frequency'),ylabel('Magnitude (db)'),grid
meta tbode
pause
semilogx(w,phase),title('MMAG Blending Scheme')
xlabel('Frequency'),ylabel('Phase (deg)'),grid
meta
```

```
%
%
%       THESC2D.m                                   8 DEC 1989
%
%
%      This program converts the transfer functions to state space
%      reprensentations. It will then discretize the state space
%      representation.
%
%

%       Enter transfer functions
%
num1=[0 0 3947.8];
den1=[1 88.844 3947.8];
num2=[1 0 0];
den2=[1 12.57 157.91];
[a1,b1,c1,d1]=tf2ss(num1,den1);
[a2,b2,c2,d2]=tf2ss(num2,den2) ;
%
%    Convert to discrete time
%
dt=0.0005;
[phi1,del1]=c2d(a1,b1,dt)       ;
[phi2,del2]=c2d(a2,b2,dt)        ;
%
%       Find discrete 5x5 for whole system
%
a=[0 1 0 0 0
   -3947.8 -88.844 0 0 0
   0 0 0 1 0
   0 0 -157.91 -12.57 0
   0 0 0 0 -1000];
%
b=[0 -3947.8 0 -157.91 1000]';
%
[phi,del]=c2d(a,b,dt);


phit=phi1+phi2;
delt=del1+del2;
at=a1+a2;
bt=b1+b2;
ct=c1+c2;
dt=d1+d2;
```

# LIST OF REFERENCES


1. Interview and correspondence between Bruce Connor, Engineer, Martin Marietta Aerospace Group, Denver, Colorado, and the author, March 1989 - January 1990.
2. Martin Marietta Aerospace Group, Unclassified letter to author regarding subject, June 1989.
3. Kirk, Donald E., *Optimal Control Theory*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1970.
4. Galinas, William J., *Fixed Interval Smoothing Algorithm For An Extended Kalman Filter For Over-The-Horizon Ship Tracking*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1989.
5. Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Trans. ASME, J. Basic Eng.*, Vol. 83D, No. 1, pp. 95-108, March 1961.
6. Analytical Sciences Corporation, Technical Staff, *Applied Optimal Estimation*, Arthur Gelb, ed., The M.I.T. Press, Cambridge, Massachusetts, 1974.
7. Conversations between Steve Spehn, Captain, U.S. Marine Corps, and the author, November 1989 - January 1990.
8. Therrien, Charles W., *Decision Estimation and Classification*, John Wiley and Sons, New York, 1989.

# BIBLIOGRAPHY


1. Berkey, Dennis D., *Calculus*, Saunders College Publishing, Philadelphia, Pennsylvania, 1984.

2. Cooper, George R. and McGillem, Clare D., *Probabalistic Methods of Signal and Systems Analysis*, Holt, Reinhart and Winston, New York, New York, 1986.

3. Friedland, Bernard, *Control System Design*, McGraw Hill, Inc., New York, 1986.

4. Goodwin, Graham C. and Sin, Swai Sang, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

5. Grossman, Stanley I., *Multivariable Calculus, Linear Algebra, and Differential Equations*, Harcourt, Brace, Jovanovich, Inc., San Diego, California, 1986.

6. Hostetter Gene H., *Digital Control System Design*, Holt, Rinehart and Winston, Inc., New York, New York, 1988.

7. Papoulis, Athanasios, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Inc., New York, New York, 1984.

8. Strum, Robert D. and Kirk, Donald E., *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1988.

9. Thaler, George J., *Automatic Control Systems*, West Publishing Co., St. Paul, Minnesota, 1989.

10. Therrien, Charles W., *Decision Estimation and Classification*, John Wiley and Sons, New York, 1989.

# INITIAL DISTRIBUTION LIST

| | | No. of Copies |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22304-6145 | 2 |
| 2. | Library, Code 0142<br>Naval Postgraduate School<br>Monterey, California 93943-5002 | 2 |
| 3. | Superintendent<br>Chairman, Code EC<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5004 | 1 |
| 4. | Superintendent<br>Professor Harold A. Titus, Code EC/Ts<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5004 | 2 |
| 5. | Superintendent<br>LCDR Janine England, Code EC/Eg<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5004 | 1 |
| 6. | Superintendent<br>Space Systems/$C^3$ Curricular Office, Code CC<br>Naval Postgraduate School<br>Monterey, California 93943-5004 | 1 |
| 7. | Commander<br>Naval Space Command<br>Attn: Code N155<br>Dahlgren, Virginia 22448 | 1 |

8. United States Space Command — 1
   Attn: Technical Library
   Peterson AFB, Colorado 80914

9. Director — 1
   Navy Space Systems Division (OP-943)
   Washington, D.C. 20350-2000

10. Superintendent — 1
    Space Systems Academic Group, Code SP
    Naval Postgraduate School
    Monterey, California 93943-5000

11. Bruce Connors, R2P2 Simulator — 2
    Martin Marietta Astronautics Group
    P.O. Box 179
    Denver, Colorado 80201

12. CPT Terry Bauer — 2
    346 Dogwood Circle
    Radcliff, Kentucky 40160