NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMPUTER SIMULATED MISSILE-TARGET
ENGAGEMENT WITH A LUENBERGER OBSERVER
AND A GROUND OBSERVER

by

GERALD L. PAINTER

June, 1991

Thesis Advisor: H.A. Titus

Approved for public release; distribution is unlimited.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b OFFICE SYMBOL *(If applicable)*
EC

7a NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*
Computer Simulated Missile-Target Engagement with a
Luenberger Observer and a Ground Observer

12. PERSONAL AUTHOR(S)
Gerald L. Painter

13a. TYPE OF REPORT
Masters Thesis

13b TIME COVERED
FROM ________ TO ______

14 DATE OF REPORT *(Year, Month, Day)*
June 1991

15 PAGE COUNT
107

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 Defence or the U.S. Government.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
missile,proportional navigation,Luenberger Observer
Kalman Filter

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

The technique of proportional navigation is applied to a three dimensional missile-target engagement model. A Luenberger observer is developed to reduce the effects of noise or jamming to the missile's seeker head. A ground observer which computes the deviations in target position and velocity and uplinks the deviations to the missile is also evaluated using the Kalman Filter theory to determine the benefits of the additional information provided to the missile.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Harold A. Titus

22b. TELEPHONE *(Include Area Code)*
(408) 646-2560

22c OFFICE SYMBOL
EC/Ts

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

Approved for public release; distribution is unlimited.

Computer Simulated Missile-Target Engagement
with
a Luenberger Observer and a Ground Observer

by

Gerald L. Painter
Lieutenant, United States Navy
B.S., Temple University, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

ABSTRACT

The technique of proportional navigation is applied to a three dimensional missile-target engagement model. A Luenberger Observer is developed to reduce the effects of noise or jamming to the missile's seeker head. A ground observer which computes the deviations in target position and velocity and uplinks the deviations to the missile, is also evaluated using the Kalman Filter theory to determine the benefits of the additional information provided to the missile.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Many thanks to Professor Hal Titus for his patience and professional guidance when the author's inexperienced assumputions were erroneous. Special thanks to LT Frances Lukenbill, who earlier developed the initial condition section of the computer simulation, and Colin Cooper for his many hours of instruction using the MATLAB computer language. The author also wishes to acknowledge the total support of his wife, Theresa, and his sons, Jeremy and Andrew, for without their support and understanding, this thesis would not have been completed.

I. INTRODUCTION

Guided missiles can be classified into four categories depending on launch and target position characteristics. The categories are Air-to-Air, Air-to-Surface, Surface-to-Air, and Surface-to-Surface missiles.

Another classification among missiles is the guidance system of the missile. The missile can be command or homing guidance.

In the command guidance system the missile and target are continuously tracked and guided from one or more friendly vantage points, and the necessary path for intercept is computed and relayed to the missile.

In the homing guidance system, the missile has a homing device onboard which can detect the target and gives the necessary path directions for intercept to the missile. The homing missile is further subdivided into classes having active, semiactive, and passive guidance systems. Active detection is when the missile illuminates the target, i.e., with a radar, and receives the reflected signals. Semiactive detection is when the target is illuminated by a source other than the missile and the missile receives the reflected signals. Passive detection is used when the target is the source of energy, and the missile detects signals that propagate from the target.

Each of the missiles in the above categories will employ one or more of the three guidance laws. These laws are Pursuit Guidance, Line-of-Sight Guidance, and Proportional Guidance. The first portion of the missile flight path may use one of the guidance laws but the terminal phase of flight may be best suited for another.

The present work addresses the design and evaluation of a semiactive Surface-to-Air missile using Proportional Navigation as the guidance law. A ground based target tracker will also developed with the target deviations in position and velocity relayed to the missile.

Chapter II presents a description and comparison of the three different guidance laws. The Proportional Navigation guidance law will also be developed. In Chapter III the missile and target flight path models will be developed using the concepts of Chapter II and computer simulation studies will be performed. Chapter IV consists of a Luenberger observer design, and an evaluation of the estimator, and the guidance law over a range of conditions will be conducted. Chapter V consists of the development of the ground target tracker using the theory of the Kalman Filter and again, computer simulations will be included to determine the accuracy of the target tracker.

All computer simulations are developed and conducted using the Matrix Laboratory (MATLAB) language.

II. MISSILE GUIDANCE

A. GUIDANCE LAW SELECTION

The selection of a guidance law is a pre-requisite for determining the initial calculations for the model. The missile guidance system measures the error between the missile's actual and desired course, computes the corrections necessary to reduce the error based on the guidance law selected, and gives commands to the autopilot to activate the controls required to achieve acceptable intercept of the target. The miss distance and the acceleration required by the missile are functions of the guidance law.

1. Pursuit Guidance

The pursuit guidance law is illustrated in Figure 1 and is described as having the missile velocity vector directed toward the target at all times. The missile is always heading along the line-of-sight from the missile to the target. This guidance law is effective against slow moving targets, but the missile may lack sufficient maneuverability against fast moving maneuverable targets.

2. Line-of-Sight Guidance

Line-of-sight guidance is used in a beam-rider type missile and is illustrated in Figure 2. This guidance law requires that the missile remain on a line joining the target

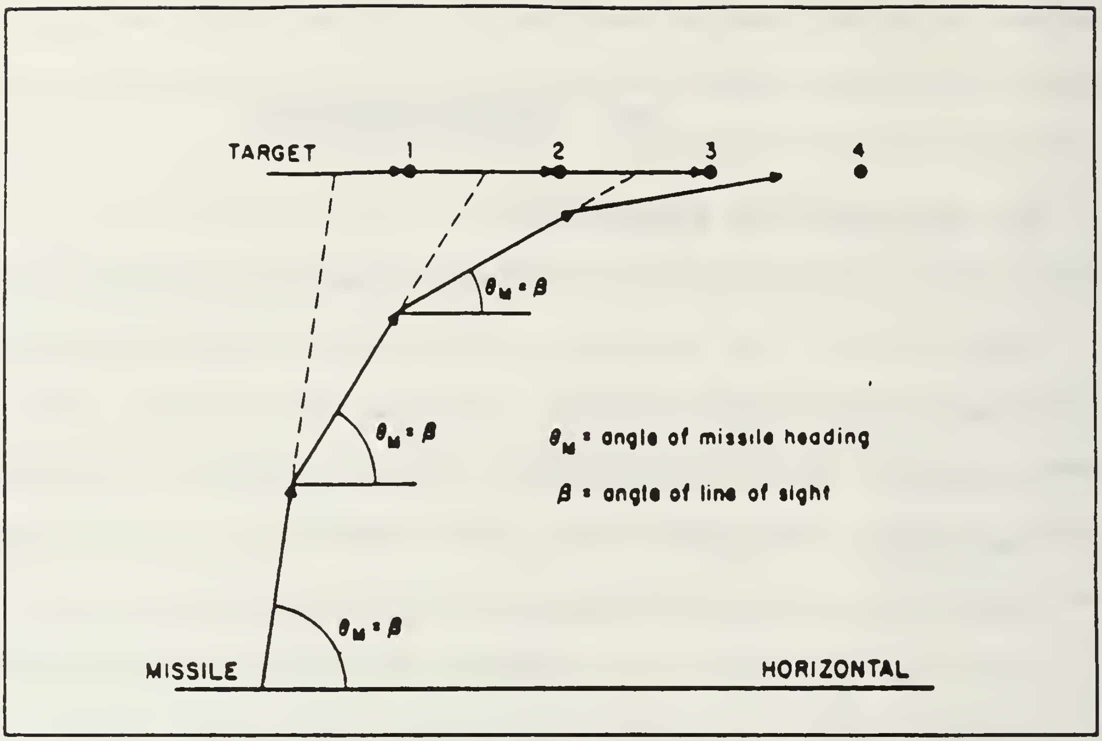

Figure 1. Pursuit Guidance Trajectory

and the target tracker. The purpose of the target tracker is to maintain the antenna boresight pointing at the center of the reflecting area of the target. This guidance scheme normally requires a dedicated fire control system from launch to intercept [Ref.1].

3. Proportional Navigation Guidance

This guidance law requires that the missile travel in such a way that its own rate of turn is proportional to the rate of turn of the line-of-sight from the missile to the target. Figure 3 illustrates the proportional guidance scheme in which the rate of change of the missile heading is made proportional to the rate of change of the line-of-sight between the missile and the target. The fixed or variable

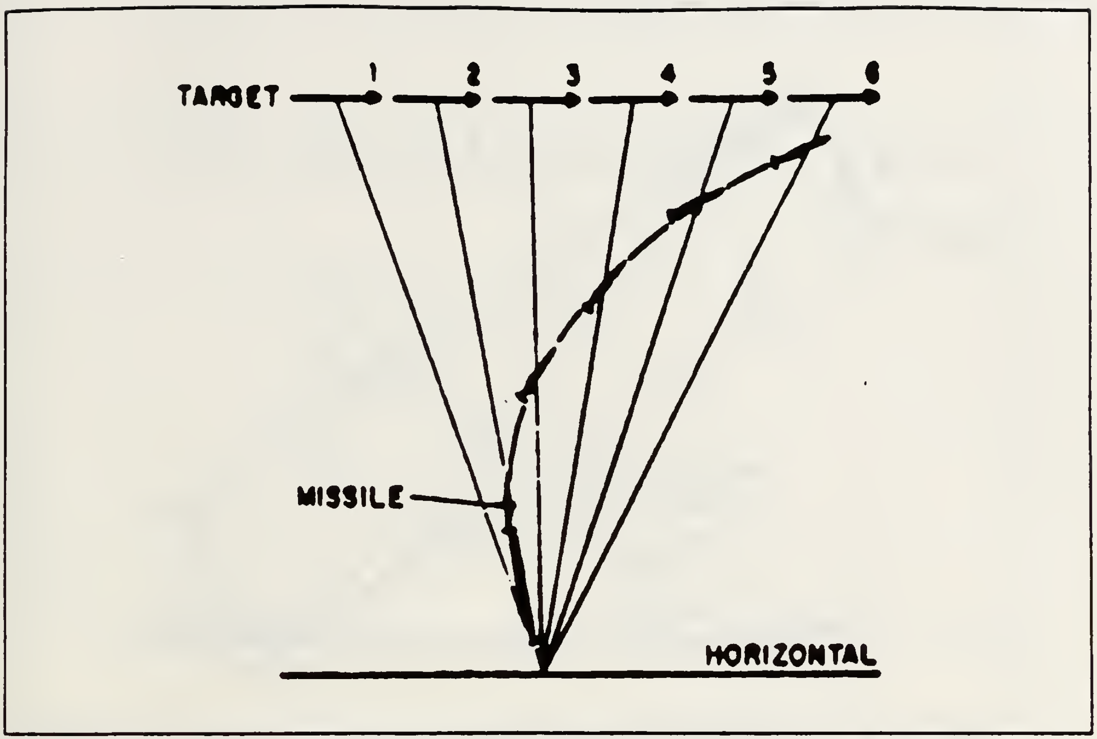

Figure 2. Line-of-Sight Trajectory

multiple between the missile rate of turn and the rate of turn of the line-of-sight is called the navigation ratio (NR). The proportional navigation guidance law attempts to generate an acceleration command perpendicular to the line-of-sight. One way to achieve this could be lateral acceleration coupled with angular or angular rate commands to place the acceleration perpendicular to the line-of-sight. The advantage of this guidance law is in its effectiveness against maneuvering targets. Since proportional navigation guidance anticipates the targets future position, it can attain a higher degree of responsiveness over the other two guidance laws.

Figure 4 illustrates the proportional navigation scheme [Ref.2]. If the seeker head of the missile follows the

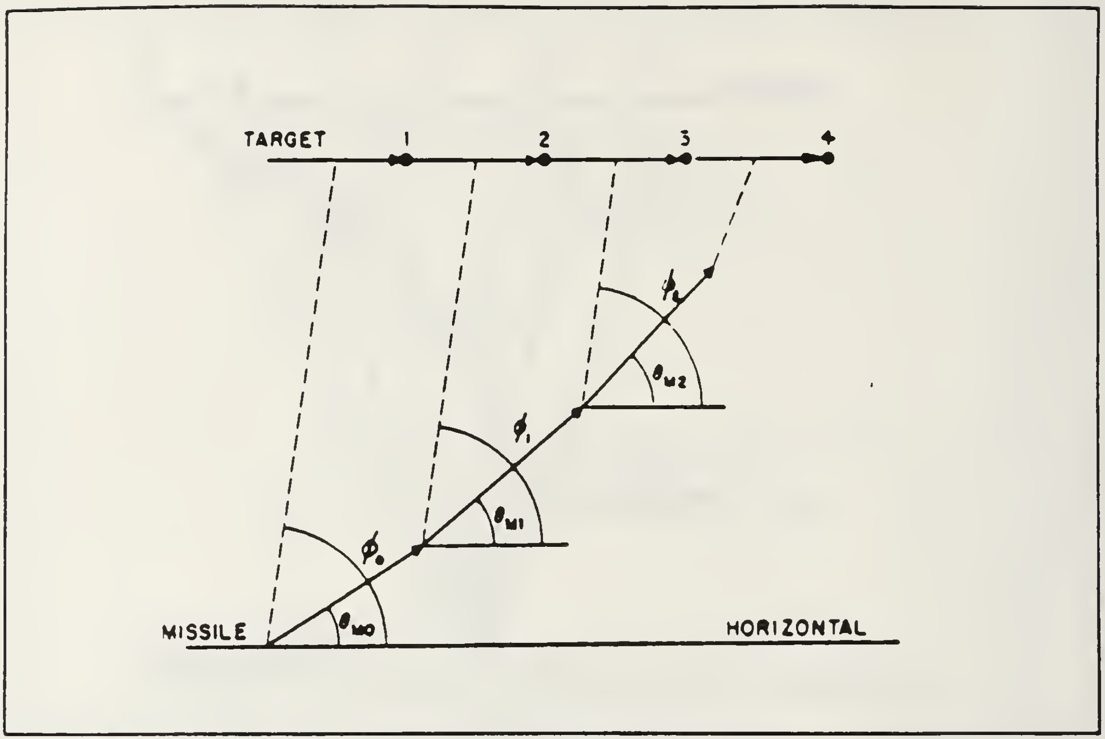

Figure 3. Proportional Guidance Trajectory

$$A_T = A_R = \frac{d}{dt} (\omega\ R) + \omega\ \dot{R} \qquad (2.1)$$

target, the target acceleration perpendicular to the line-of-sight will equal the acceleration of the R vector, where

R = missile-target line-of-sight vector

$\dot{R}$ = closing rate along R

ω = angular rate of change of R

A_T = target acceleration perpendicular to R

A_R = tangential acceleration of vector R

The term ($\omega\dot{R}$) represents the vectorial acceleration of R and the term $d/d\omega(\omega R)$ represents the rate of change of the tangential velocity. A missile acceleration, A_M, equal to the target acceleration A_T, at this point will make the line-of-

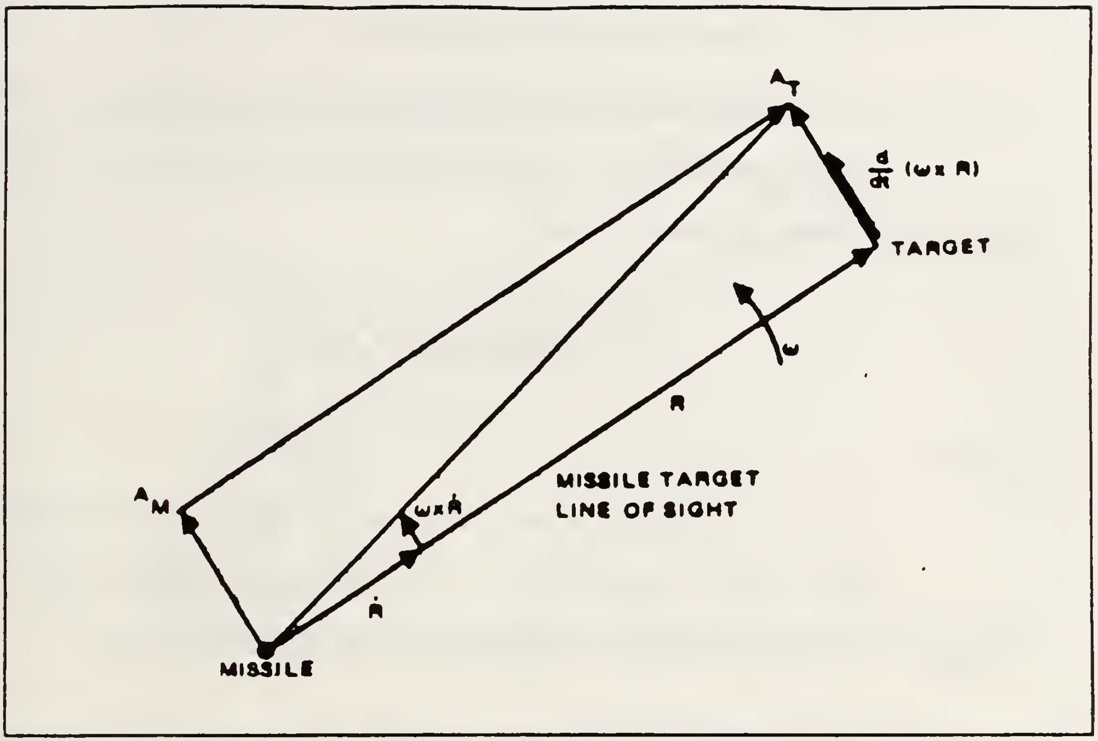

Figure 4. Proportional Navigation Scheme

sight parallel to its original direction. Since the velocity $\dot{R}$ is along the vector R, a missile/target intercept is assured. Therefore, from Equation (2.1), the executed missile acceleration commands should be

$$A_M = 2\ (\omega\dot{R})\ +(\dot{\omega}R) \qquad (2.2)$$

Since the direction of the velocity vector cannot be directly controlled, proportional navigation is achieved by controlling the commanded missile acceleration (acom).

$$acom = V_M\,\dot{\gamma} \qquad (2.3)$$

where

V_M = the missile velocity

$\dot{\gamma}$ = the rate of change of the velocity vector

Implementation of proportional navigation results in the following guidance law

$$acom = NR\ V_M\ \dot{\sigma} \tag{2.4}$$

where

$\dot{\sigma}$ = the rate of change of the line-of-sight

With this definition, the equation for the rate of change of the velocity vector can be written as

$$\dot{\gamma} = NR\ \dot{\sigma} \tag{2.5}$$

B. PROPORTIONAL NAVIGATION KINEMATICS

From Figure 5 the following equations of motion are obtained. The three dimensional linear model will be developed in Chapter III, but for simplicity and ease of understanding, the fundamental equations will be first developed in the x and y planes. The velocity vector V_M is at an angle γ_M from the established reference line. From the geometry of the problem, the missile flight path angle can be easily determined.

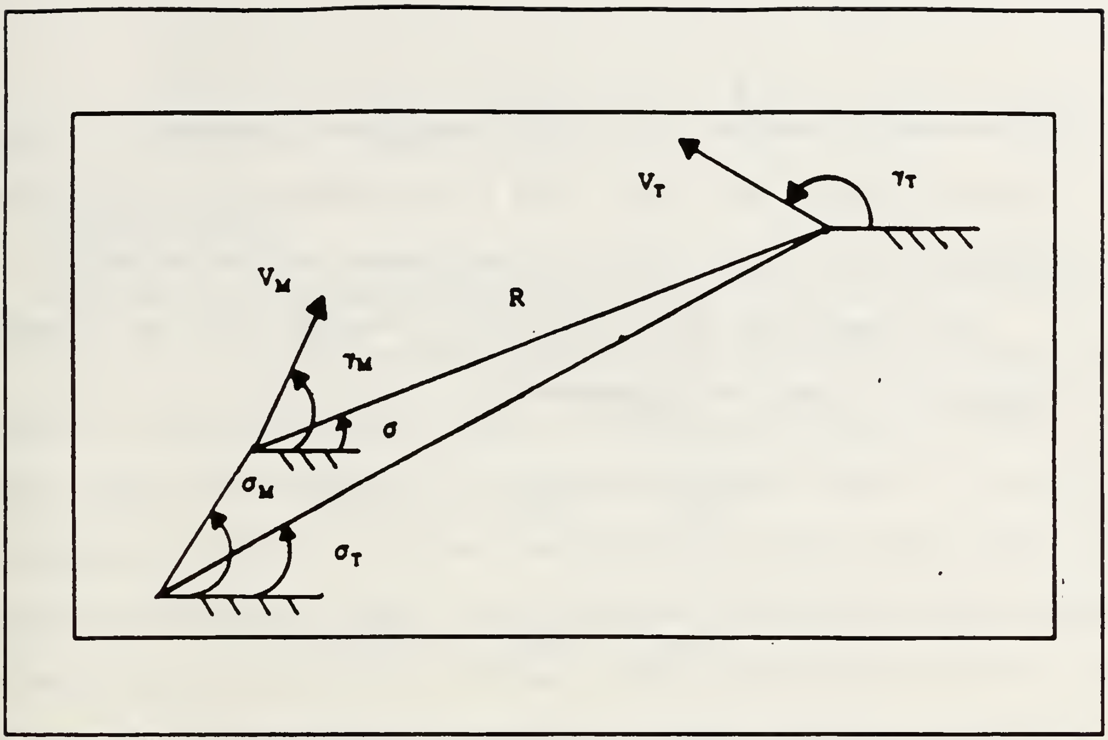

Figure 5. Intercept Geometry for Proportional Navigation

$$\gamma_M = \arctan\left(\frac{V_{MY}}{V_{MX}}\right) \tag{2.6}$$

where V_{MX} and V_{MY} are the components of the missile velocity vector in the x and y directions, respectively. The target flight path angle is found from the same geometry and is expressed as

$$\gamma_T = \arctan\left(\frac{V_{TY}}{V_{TX}}\right) \tag{2.7}$$

where V_{TX} and V_{TY} are the target velocities in the x and y directions, respectively.

The missile-target line-of-sight vector is R.

$$R = [(X_T - X_M)^2 + (Y_T - Y_M)^2]^{1/2} \tag{2.8}$$

where X_T and Y_T are the x and y coordinates of the target and X_M and Y_M are the x and y coordinates of the missile.

The magnitude of the velocity vector for the missile and target (V_T) can be expressed as

$$V_M = [(V_{MX})^2 + (V_{MY})^2]^{1/2} \tag{2.9}$$

$$V_T = [(V_{TX})^2 + (V_{TY})^2]^{1/2} \tag{2.10}$$

The line-of-sight angle (σ) is defined as

$$\sigma = \arctan\left(\frac{Y_T - Y_M}{X_T - X_M}\right) \tag{2.11}$$

and the rotation rate of the line-of-sight ($\dot{\sigma}$) can be expressed as

$$\dot{\sigma} = \frac{V_T \sin(\gamma_T - \sigma) - V_M \sin(\gamma_M - \sigma)}{R} \tag{2.12}$$

where $V_T \sin(\gamma_T - \sigma)$ and $V_M \sin(\gamma_M - \sigma)$ are the target and missile velocity components normal to the line-of-sight.

III. KINEMATIC REPRESENTATION OF A SKID-TO-TURN MISSILE

A. INTRODUCTION

Two basic methods of controlling the attitude of a missile to achieve the acceleration commanded by the guidance law are skid-to-turn and bank-to-turn. In the skid-to-turn method the roll angle is held to a small quantity and is usually considered zero in initial calculations. The magnitude and orientation of the body acceleration vector is achieved by permitting the missile to develop both an angle of attack and a sideslip angle. The presence of the sideslip imparts a skidding motion to the missile. The bank-to-turn missile generally will develop higher lift accelerations than the skid-to-turn method, so that the missile requires banking maneuvers to properly direct the control vector. To achieve the desired orientation, the missile is rolled or banked so that the plane of maximum normal force is oriented in the desired direction. The present work assumes a skid-to-turn missile that is roll stabilized.

B. ASSUMPTIONS

A simplified, point mass representation of the missile equations of motion will be developed under the following assumptions.

1. The missile thrust exactly cancels drag.
2. The orientation of the missile can be described by the Euler angles that represent the flight path angles in pitch and yaw.
3. The seeker head angle rate is a good estimate of the line-of-sight rate.
4. For small angles of attack and sideslip, the velocity vector is assumed to be aligned with the body center line and if the missile speed is maintained nearly constant, the missile acceleration in the x direction is fairly close to zero.

C. SIMPLIFIED EQUATIONS OF MOTION

Figure 6 illustrates the flight path geometry of the missile or target, with the velocity vector aligned with the body centerline. From the flight path geometry, the missile and target velocity magnitudes are expressed as

$$V_M = [\ (V_{MX})^2+(V_{MY})^2+(V_{MZ})^2\ ]^{1/2} \tag{3.1}$$

for the missile, and

$$V_T = [(V_{TX})^2+(V_{TY})^2+(V_{TZ})^2\ ]^{1/2} \tag{3.2}$$

for the target.

The angles of attack (γ_pitch) and sideslip (γ_yaw) are defined as kinematic quantities depending only on velocity ratios. From Figure 6, the missile and target flight path

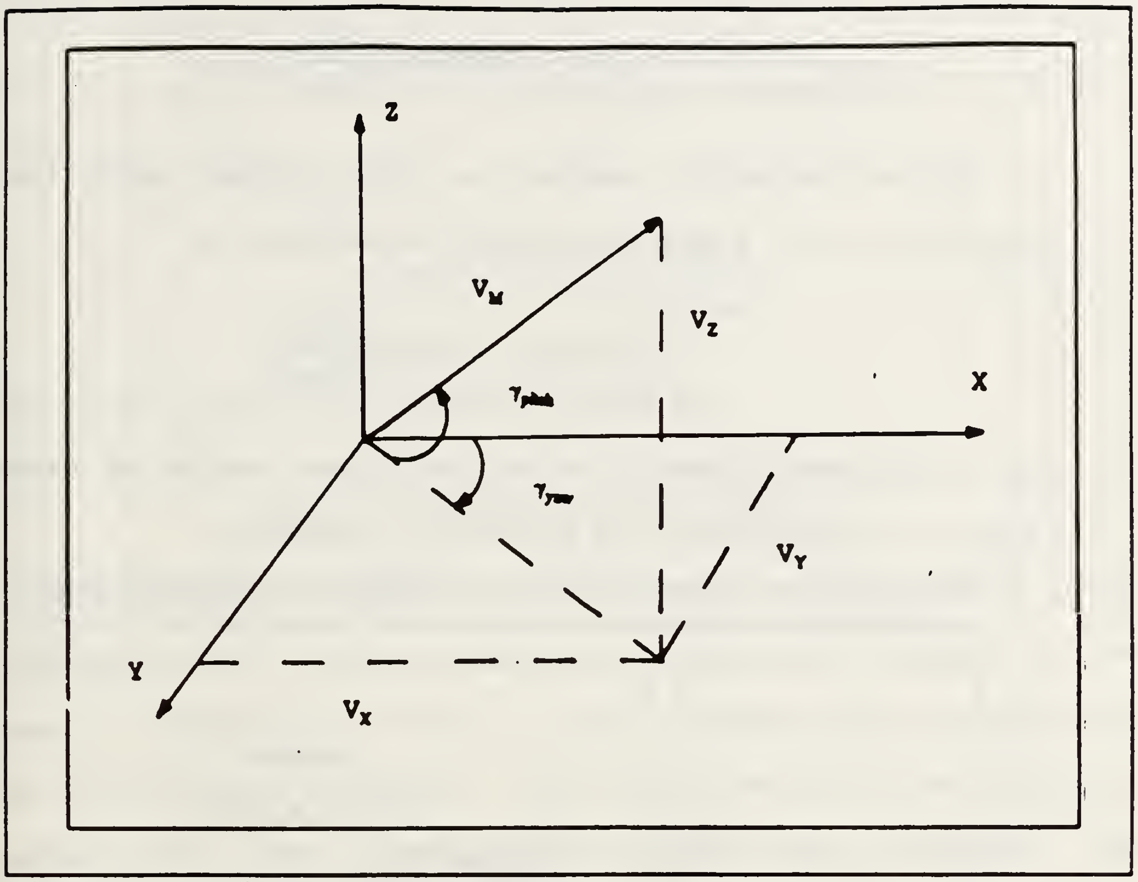

Figure 6. Flight Path Geometry

angles can be defined as

$$\gamma_M\, pitch = \arctan\left(\frac{V_{MZ}}{\sqrt{((V_{MX})^2+(V_{MY})^2))}}\right) \tag{3.3}$$

$$\gamma_M\, yaw = \arctan\left(\frac{V_{MY}}{V_{MX}}\right) \tag{3.4}$$

$$\gamma_T\, pitch = \arctan\left(\frac{V_{TZ}}{\sqrt{((V_{TX})^2+(V_{TY})^2)}}\right) \tag{3.5}$$

$$\gamma_T \, yaw = \arctan\left(\frac{V_{TY}}{V_{TX}}\right) \qquad (3.6)$$

If the velocity vector is not aligned with the body centerline, the angle of attack is defined as

$$\gamma_M \, pitch = \arctan\left(\frac{V_{MZ}}{V_{MX}}\right) \qquad (3.7)$$

with the sideslip angle in the yaw plane remaining unchanged. Figure 7 illustrates the sightline geometry.

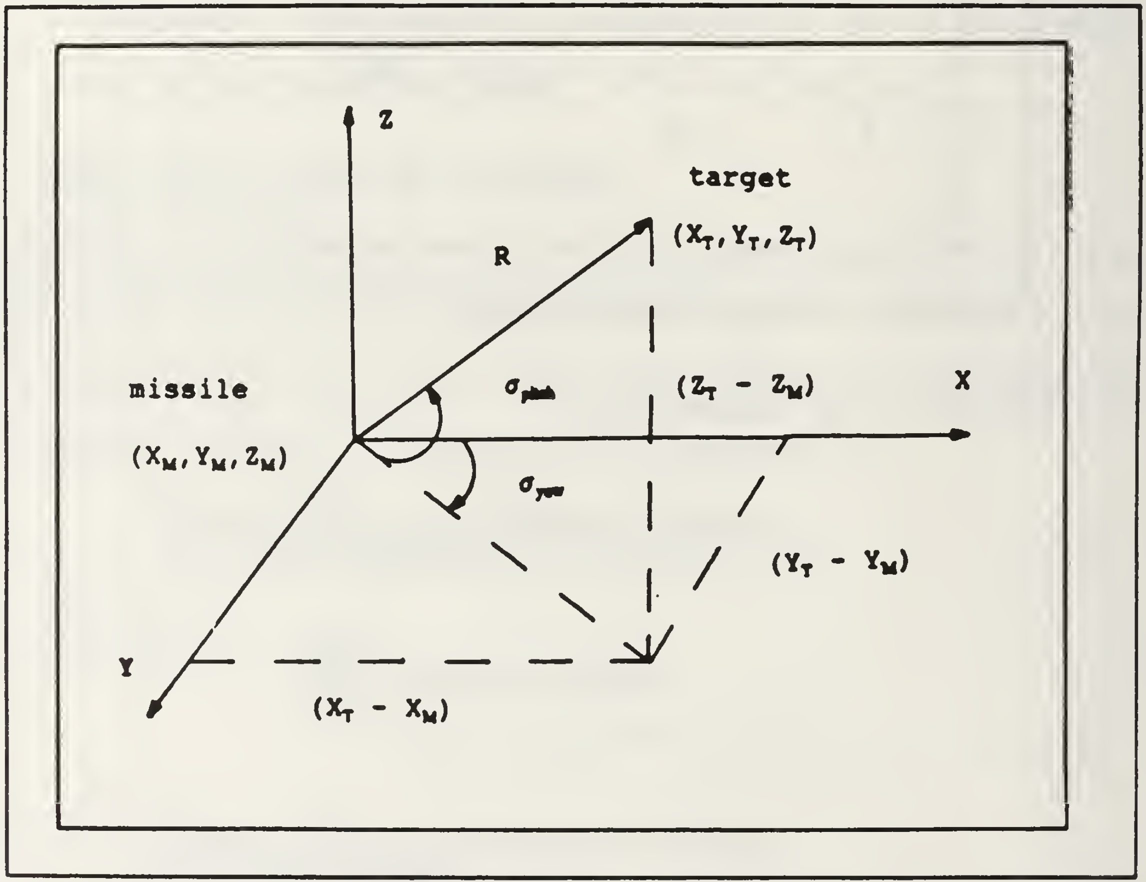

Figure 7. Sightline Geometry

From the sightline geometry, the line-of-sight angles can be determined.

$$\sigma\ pitch = \arctan\left(\frac{Z_T-Z_M}{\sqrt{((X_T-X_M)^2+(Y_T-Y_M)^2)}}\right) \qquad (3.8)$$

$$\sigma\ yaw = \arctan\left(\frac{Y_T-Y_M}{X_T-X_M}\right) \qquad (3.9)$$

The range vector (R) can be defined as

$$R = \sqrt{((X_T-X_M)^2+(Y_T-Y_M)^2+(Z_T-Z_M)^2)} \qquad (3.10)$$

The velocity components can be found from the Euler angle transformation. The derivation of the Euler angles can be found in Reference 4. The x, y, and z directions are taken as the longitudinal, lateral, and normal axis of the missile, respectively. The corresponding angles that represent the angular displacements are ϕ (roll), θ (pitch), and ψ (yaw). With the missile assumed to be roll stabilized, the roll angle will be taken as zero and not included. The Euler transformation matrix is given as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\sin(\psi) & \cos(\psi) & 0 \\ \sin(\theta)\cos(\psi) & \sin(\theta)\sin(\psi) & \cos(\theta) \end{bmatrix} \begin{bmatrix} I \\ J \\ K \end{bmatrix} \qquad (3.11)$$

where I points north, J east, and K down. The transformation matrix represents the total transformation from the inertial coordinate system to the missile body axes.

From the above matrix, the linear components of the missiles velocity can be found using the fight path angles in the yaw and pitch planes.

$$V_{MX} = V_M(\cos(\theta)\cos(\psi)) \tag{3.12}$$

$$V_{MY} = V_M(\cos(\theta)\sin(\psi)) \tag{3.13}$$

$$V_{MZ} = V_M(-\sin(\theta)) \tag{3.14}$$

A skid-to-turn missile is controlled by generating the required acceleration commands in the pitch and yaw plane. From Reference 2, the commanded acceleration in the pitch and yaw planes are defined as

$$Acom_{pitch} = V_M\, pitch\ \dot{\gamma}_M\, pitch \tag{3.15}$$

$$Acom_{yaw} = V_M\, yaw\ \dot{\gamma}_M\, yaw \tag{3.16}$$

The velocity components of the missile in the pitch and yaw planes are defined as

$$V_M\, pitch = V_M \cos(\gamma_M\, yaw - \sigma\, yaw) \tag{3.17}$$

$$V_M\, yaw = V_M \cos(\gamma_M\, pitch) \tag{3.18}$$

The autopilot of the missile will convert the commanded accelerations into angular rates $\dot{\gamma}_M$ pitch and $\dot{\gamma}_M$ yaw. These angular rates are passed to the control surface servos and are converted to the necessary fin deflections required to steer the missile to the desired course.

D. SYSTEM SIGNAL FLOW GRAPH

Figure 8 shows the total system signal flow graph in the y direction, where at launch the missile/target line-of-sight is taken as the x axis. The outputs from the seeker is the seeker head angle rate, which when multiplied by the navigation ratio, becomes the input to the autopilot. The autopilot provides the signals required for the missile dynamics.

E. LATERAL AUTOPILOT

As stated previously, the autopilot receives commands from a guidance computer and processes these commands to control the deflections of the control surface. There are three principle requirements when designing an autopilot. These are, quick response, stability, and robustness. The autopilot should be able to handle broad variations of the aerodynamic parameters. For example, the missile may be required to operate over a broad range of Mach numbers and at various altitudes which will effect the normal force coefficient, which is a function of the Mach number and the altitude.

In this simulation, a simple lateral autopilot will be modeled as a first order lag with a time constant (τ) of 0.33.

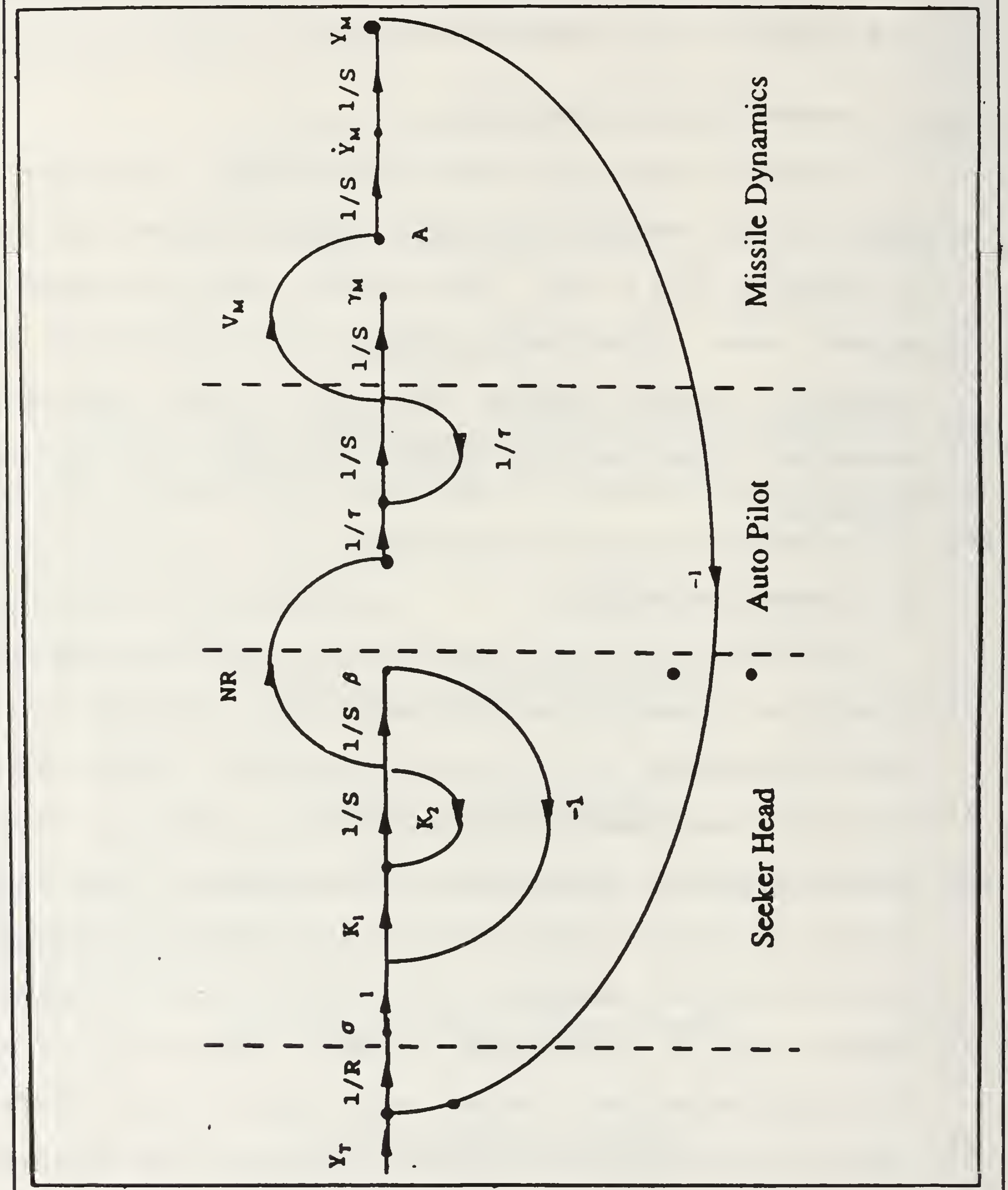

Figure 8. System Flow Graph

This has been demonstrated in past simulations to adequately approximate a more detailed autopilot.

The lateral autopilot controls the motion of the missile in the pitch and yaw planes. As in the case of this study, a symmetrical cruciform missile, the pitch and yaw autopilots are often identical so only one will need to be modeled. With the assumption that the angle of attack is very small, the velocity of the missile is aligned with the missile centerline. From Figure 8, we have the following expression

$$\ddot{\gamma}_M = -K\,\dot{\gamma}_M + K\,NR\,\beta \tag{3.19}$$

where $\dot{\beta}$ is the seeker head gimble angle rate and $\ddot{\gamma}_M$ is the angular acceleration of the missile flight path. K is equal to ($1/\tau$). The transfer function of the autopilot is given as

$$\frac{\dot{\gamma}_M\,(S)}{\beta\,\,(S)} = \frac{K\,NR}{S+K} \tag{3.20}$$

F. SEEKER HEAD DEVELOPMENT

A homing head, when mounted in an airborne missile, is called a seeker. The purpose of the seeker is to detect, acquire and track a target by sensing radiation or reflected energy from the target. In the present study, a narrow field-of-view seeker gimballed to the airframe will be designed. The seeker maintains the target within this narrow field-of-view by rotating the platform.

If the platform is inertially stabilized, rotation is achieved by applying torques which are proportional to the target's displacement. From the centerline of the field-of-view, the equation of motion is

$$T = I \beta \tag{3.21}$$

where T is the applied torque, I is the moment of inertia, and $\ddot{\beta}$ is the angular acceleration of the seeker head angle. From Figure 8 and equation (3.21) the resulting equation of motion is

$$\beta = \frac{T}{I} = -K_2 \beta - K_1 \beta + K_1 \sigma \tag{3.22}$$

where K_1 and K_2 are functions of the time constants of the seeker. The transfer function of the seeker head can then be expressed as

$$\frac{\beta (S)}{\sigma (S)} = \frac{K_1}{S^2 + K_2 S + K_1} \tag{3.23}$$

G. CONTINUOUS TIME STATE EQUATIONS

1. Missile Dynamics

Given the continuous time state equation

$$\dot{x} (t) = A x (t) + B u (t) \tag{3.24}$$

the missile or target state equation is defined.

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \qquad (3.25)$$

2. Autopilot State Equation

The pitch and yaw autopilots are identical, so the following state equation for the autopilot is defined as

$$\ddot{\gamma} = \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_N \, pitch \\ \dot{\gamma}_N \, yaw \end{bmatrix} + \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} \beta \; pitch \\ \beta \; yaw \end{bmatrix} \qquad (3.26)$$

3. Seeker Head State Equation

The seeker head continuous state equation with a time constant of 0.1 seconds in the pitch plane is defined as

$$\beta = \begin{bmatrix} 0 & 1 \\ -100 & -20 \end{bmatrix} \begin{bmatrix} \beta \; pitch \\ \beta \; pitch \end{bmatrix} + \begin{bmatrix} 0 \\ 100 \end{bmatrix} \sigma \qquad (3.27)$$

The yaw autopilot is identical to the pitch autopilot, so the above equation can be easily determined in the yaw plane.

4. Continuous to Discrete

Given the continuous time state equations, the discrete time equations are defined.

$$x\ (k+1) = \Phi\ x(k) + \Gamma\ u(k) \tag{3.28}$$

$$y\ (k+1) = C\ x(k) + D\ u(k) \tag{3.29}$$

The simulation study will use the discrete time representation of the given continuous time equations.

H. SIMULATION

This section presents the results of the missile/target engagement using the proportional navigation scheme. The following assumptions are made:

1. The missile is limited to 20 g's in either the yaw or pitch plane.
2. The target is capable of a 5 g maneuver.
3. The seeker head system is noise free. This will form a basis from which the following chapters can be referred to.
4. The minimum difference between the targets position and the missiles position, will be used as an estimate of the miss distance parameter.
5. The origin is taken as coordinates (0,0,0) in the x,y,z plane.

1. Constant Velocity Target

The first scenario will be a constant velocity target with the following initial conditions.

$V_{MX}(0) = 2000$ feet/sec

$X_T\ (0) = 20{,}000$ feet

$Z_T\ (0) = 50$ feet

$V_{TX}(0) = 1000$ feet/sec

All other initial conditions are zero. As shown in Figure 9, a successful intercept occurred for the constant velocity target. The minimum distance between the target and missile was 0.7665 feet. Figure 10 is a plot of the missile/target trajectories looking down on the X-Y plane. It shows the target starting at 20,000 feet on the X axis with the missile starting at the origin. Figure 11 is a plot of the missile/target trajectories in the Y-Z plane with the missile starting at the origin. Figure 12 is a plot of the missile commanded acceleration parameter. The acceleration increases rapidly during the initial phase of the engagement as force is applied to the missile and then drops off until just prior to intercept. Figure 13 shows the seeker head angle rate in the yaw plane and Figure 14 is a plot of the seeker head angle rate in the pitch plane. These parameters also increase rapidly during the first few seconds as the target is starting out on its flight profile and then decreases in time as the missile is on course due to the acceleration commands.

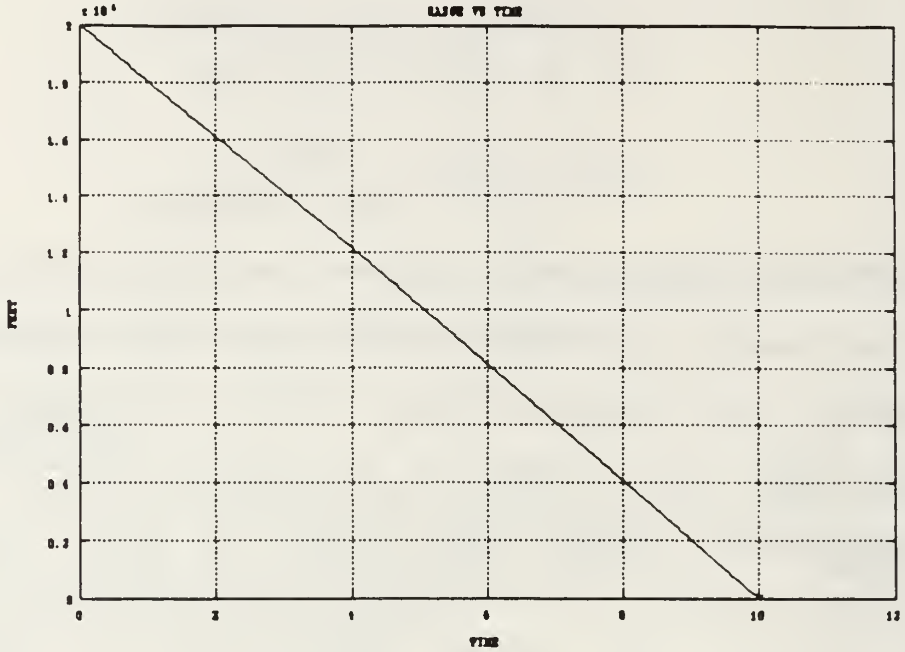

Figure 9. Range vs Time

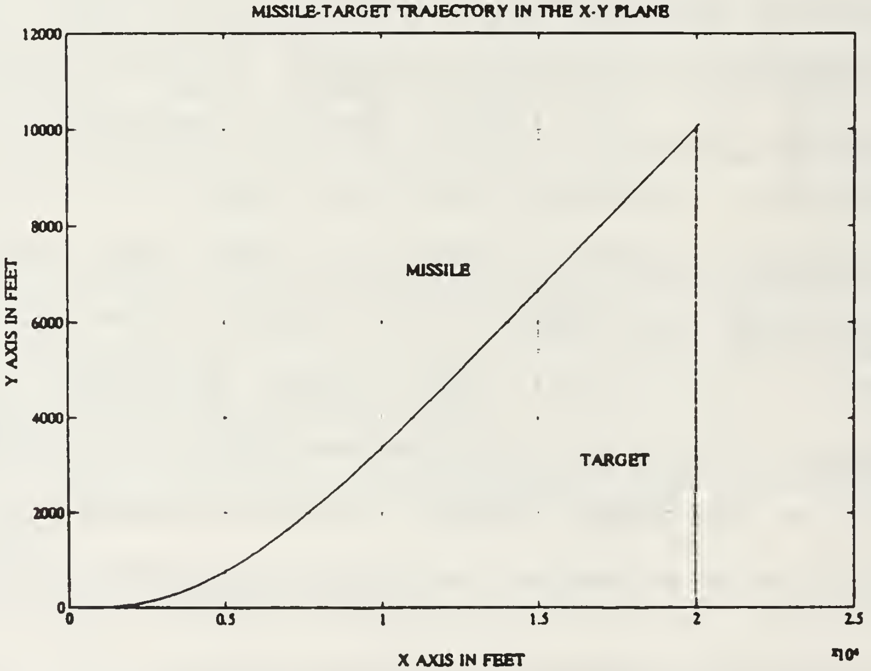

Figure 10. Trajectory in the X-Y Plane

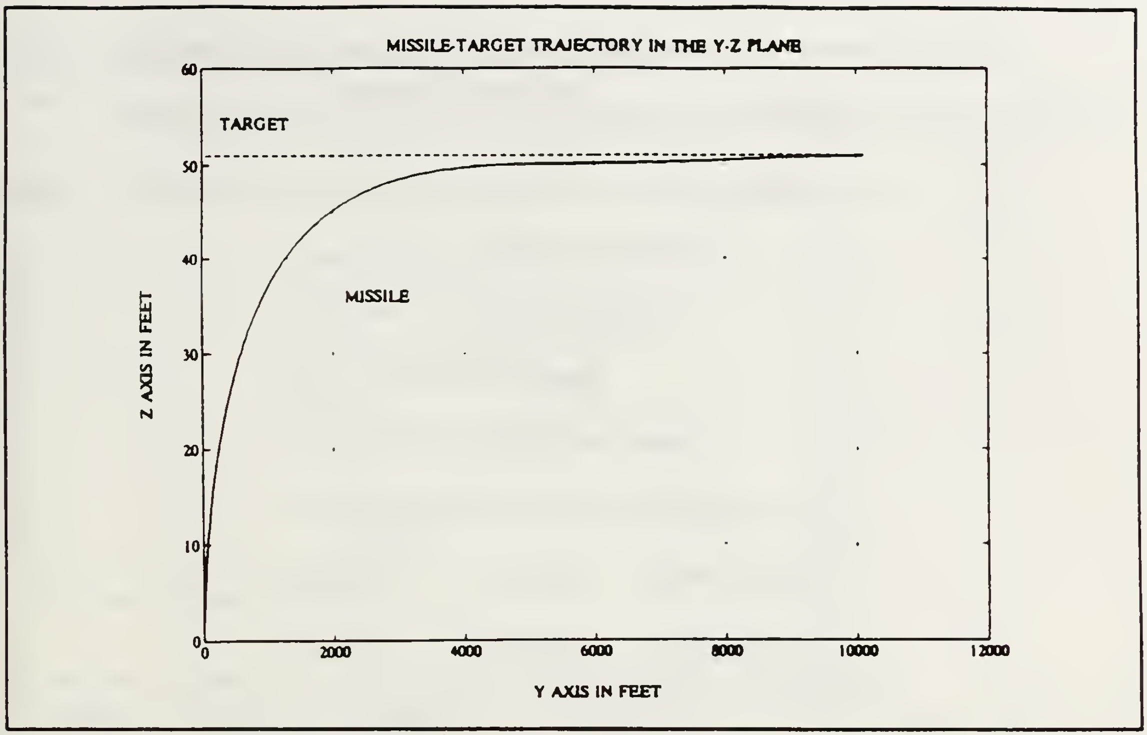

Figure 11. Trajectory in the Y-Z Plane

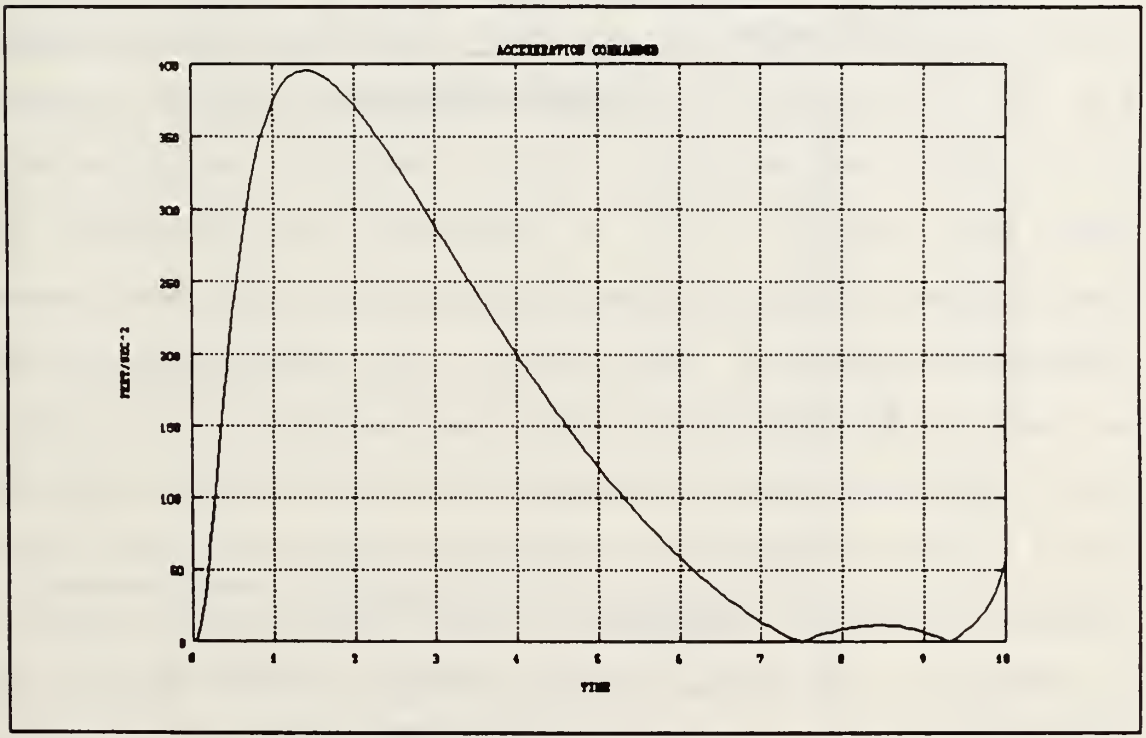

Figure 12. Commanded Acceleration

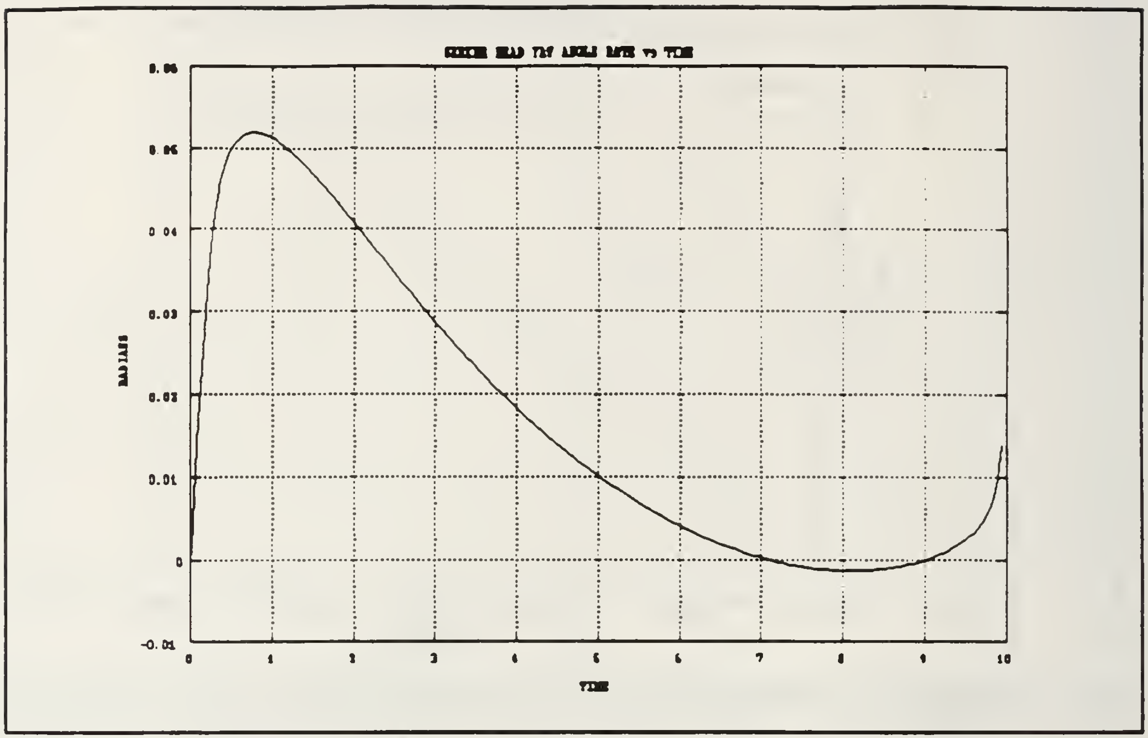

Figure 13. Seeker Head Angle Rate in the Yaw Plane

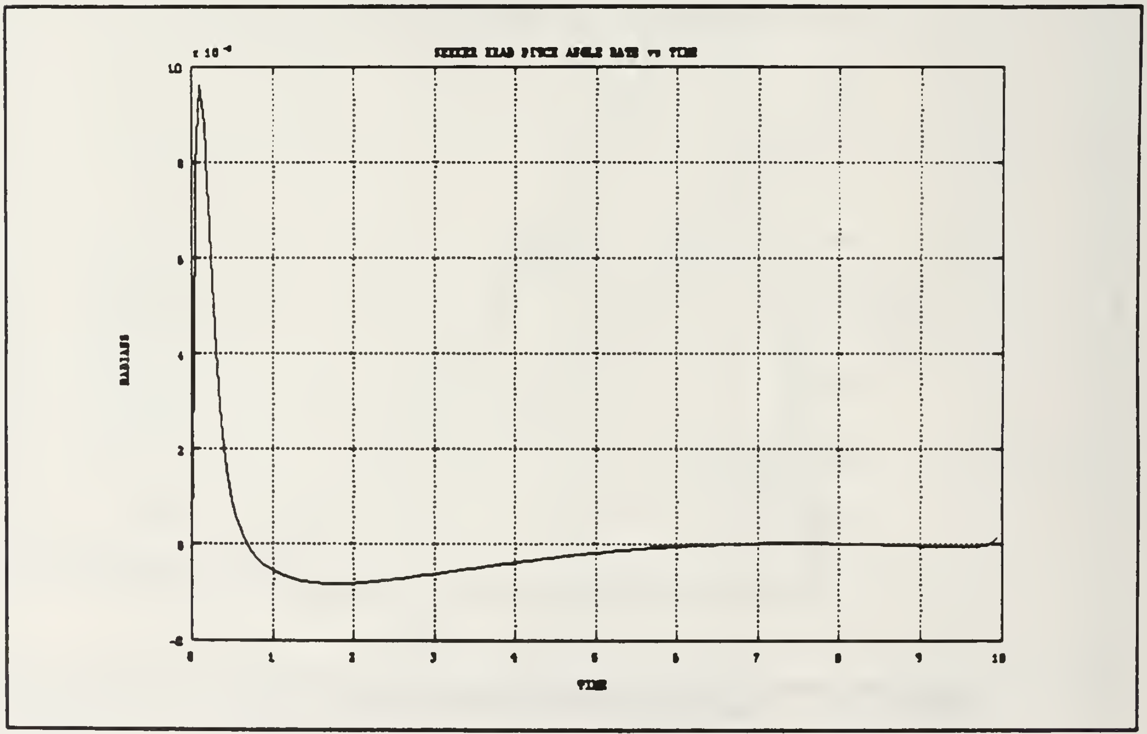

Figure 14. Seeker Head Angle Rate in the Pitch Plane

2. Constant Acceleration Target

The second scenario will be a constant acceleration target with the following initial conditions:

$V_{MX}(0) = 2000$ feet/sec

$X_T\ (0) = 20{,}000$ feet

$Z_T\ (0) = 50$ feet

$V_{TY}(0) = 500$ feet/sec

$A_{TX}(0) = -5.0*32.2$ feet/sec^2

$A_{TY}(0) = 3.5*32.2$ feet/sec^2

$A_{TZ}(0) = 3.5*32.2$ feet/sec^2

All other initial conditions are zero. Figure 15 is a plot of the minimum range between the missile and target. The minimum distance was 1.13 feet. This was as expected due to the changing target trajectory. Figure 16 is the plot of the missile/target trajectory in the X-Y plane and Figure 17 is the missile/target trajectory in the X-Z plane. Both plots demonstrate the ability of the missile using the proportional navigation guidance law, to effectively intercept the target. Figure 18 is a plot of the missile acceleration vs time. As expected, in contrast to the constant velocity simulation, the initial portion of the engagement is characterized by a low acceleration magnitude. As the engagement progress, however, due to the missile having to pull more g's in order to compensate for the more rapid trajectory of the target, the missile acceleration increased rapidly.

Figure 19 is a plot of the targets acceleration vs time. Figures 20 and 21 are the seeker head angle rates in the yaw and pitch planes, respectively. As with the acceleration of the missile, similar characteristics were realized for these parameters due to the fact that the missile is constantly changing its flight path to compensate for the acceleration of the target.

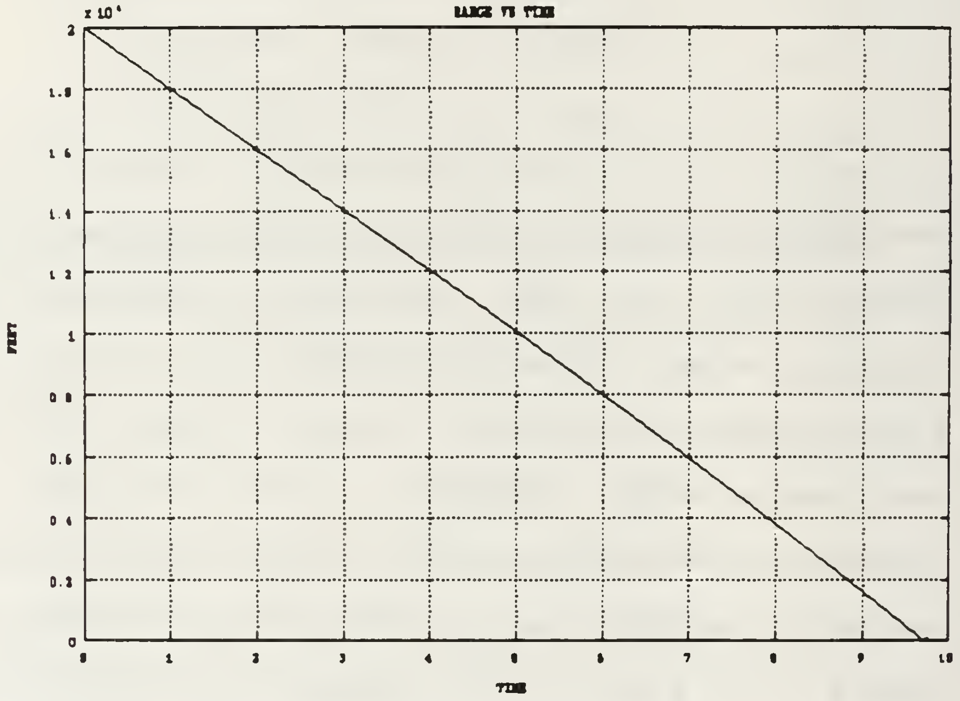

Figure 15. Range vs Time

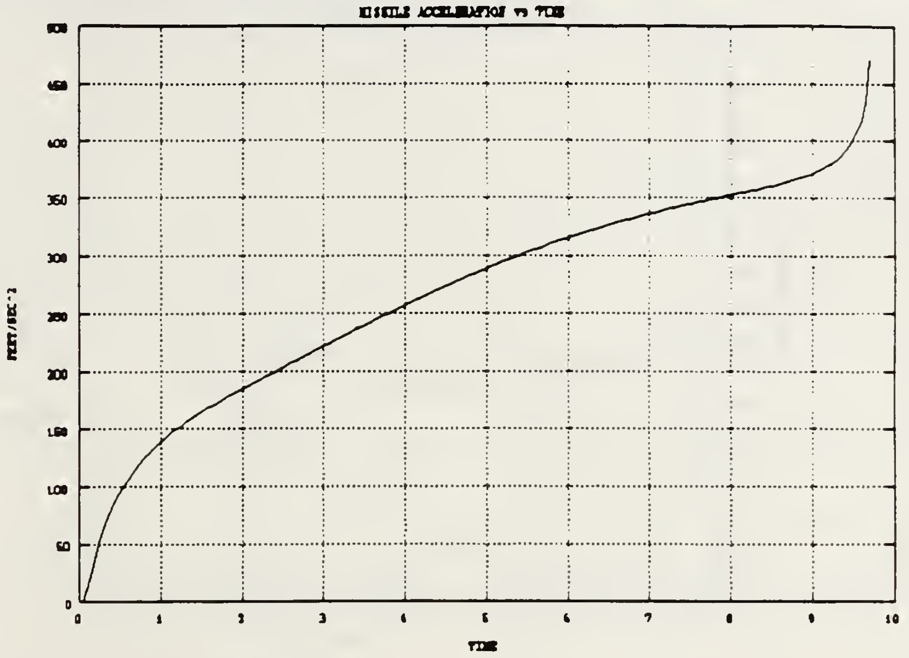

Figure 18. Missile Acceleration vs Time

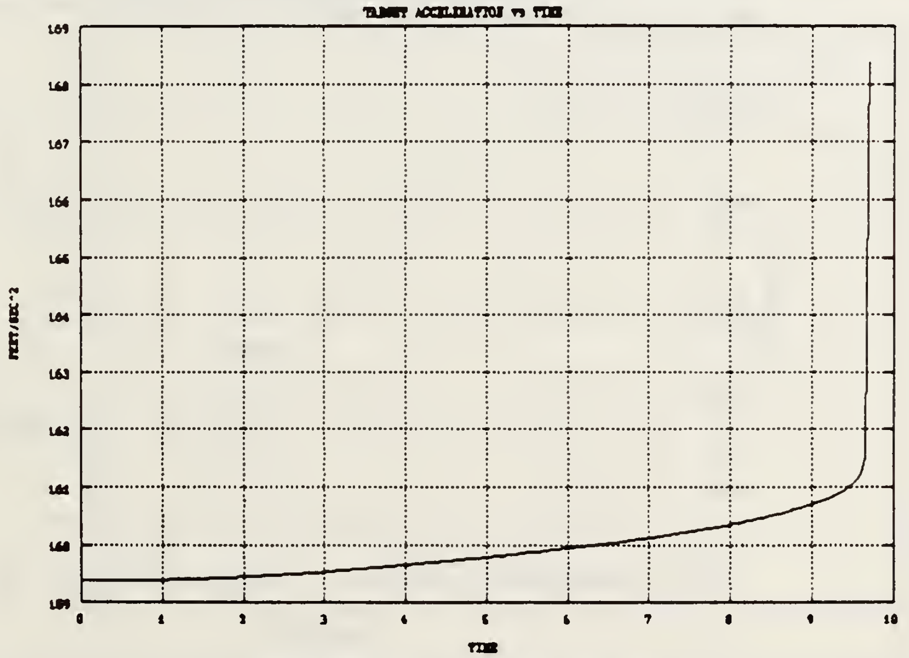

Figure 19. Target Acceleration vs Time

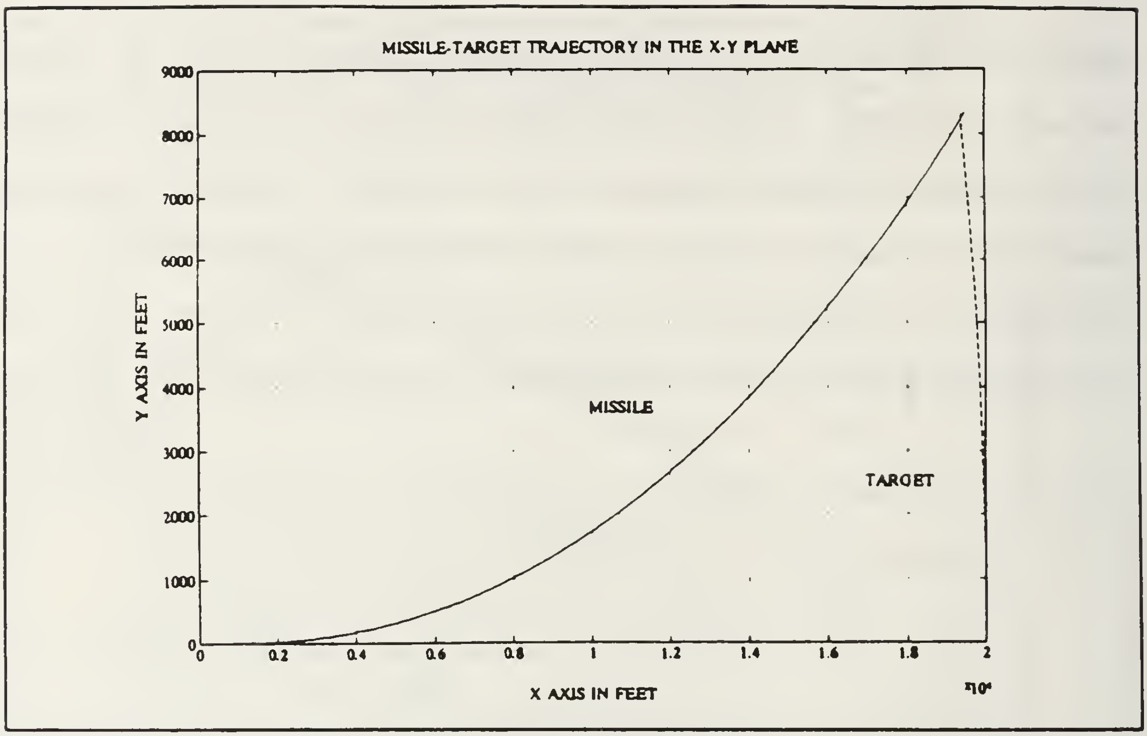

Figure 16. Trajectory in the X-Y Plane

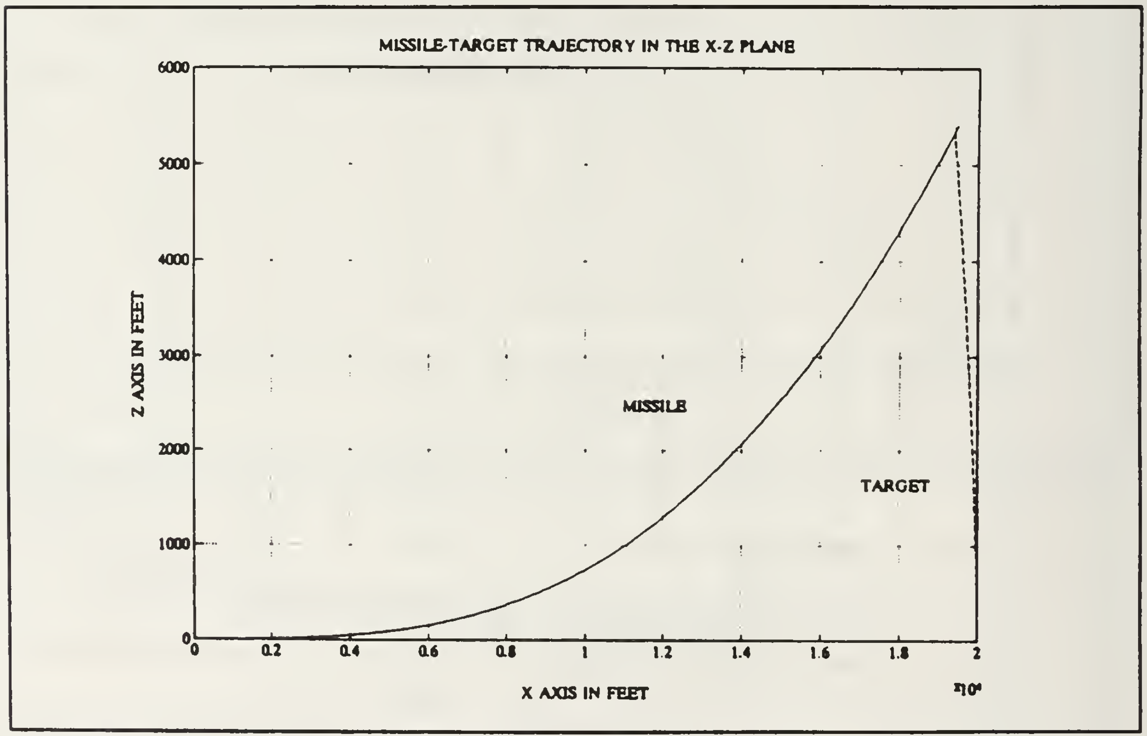

Figure 17. Trajectory in the X-Z Plane

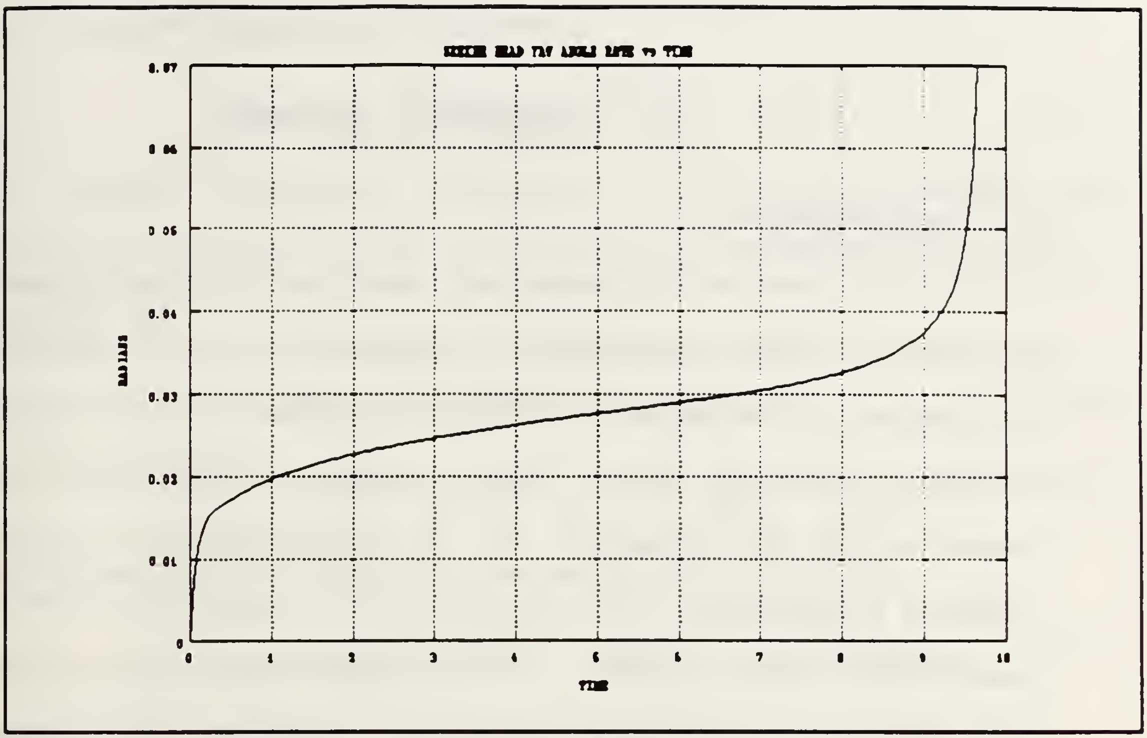

Figure 20. Seeker Head Angle Rate in the Yaw Plane

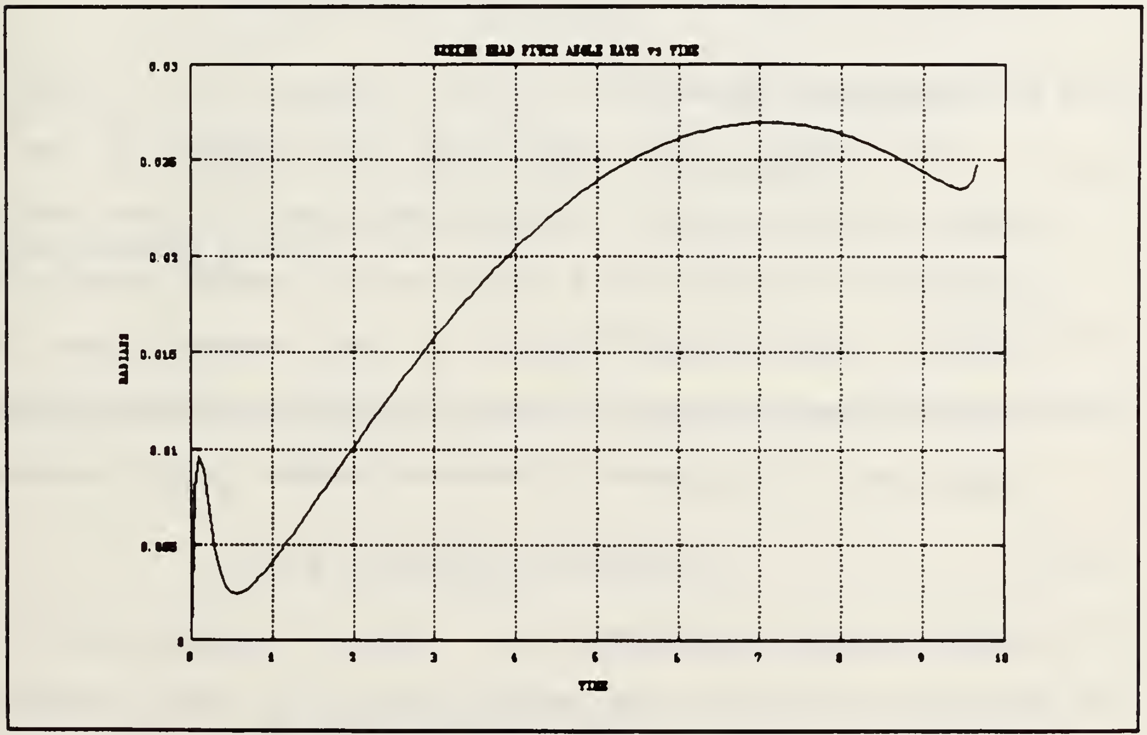

Figure 21. Seeker Head Angle Rate in the Pitch Plane

IV. LUENBERGER OBSERVER

A. INTRODUCTION

The purpose of a Luenberger observer for the proportional navigation model developed in Chapter III, is to estimate the variables in the missile/target engagement that might not be measured directly or to reduce noise on the states that are measured by the missile. It is not practical to design a missile to determine all inputs. Such inputs could be evasive maneuvers of the target, initial conditions, noise level, or the jamming techniques employed. The use of a Luenberger observer is beneficial in these situations.

B. OBSERVER DESIGN

The seeker head angle rate is assumed to be a good approximation of the line-of-sight rate. The observer will be designed to estimate the states of the seeker head, since the actual line-of-sight rate of the seeker head can be contaminated by noise or jamming. The following equations are described in Reference 4. Given a system

$$x(k+1) = \Phi x(k) + \Delta u(k) \tag{4.1}$$

and the output equation

$$y(k) = C\,x(k) + D\,u(k) \tag{4.2}$$

The observer will approximate x(k) with inputs y(k) and u(k).

The observer equation is given as

$$\hat{x}(k+1) = F\,\hat{x}(k) + G\,y(k) + H\,u(k) \tag{4.3}$$

with xhat(k) being the estimate of the original system. The error between the state x(k) and the estimated state can be defined as

$$e(k+1) = x(k+1) - \hat{x}(k+1) \tag{4.4}$$

$$e(k+1) = \Phi x(k) + \Delta u(k) - F\,\hat{x}(k) - G\,y(k) - H\,u(k) \tag{4.5}$$

which results in the error equation of the form

$$e(k+1) = (\Phi - GC)\,x(k) - F\,\hat{x}(k) + (\Delta - GD - H)\,u(k) \tag{4.6}$$

If we select F as

$$F = \Phi - GC \tag{4.7}$$

Equation (4.4) becomes a state equation of the following form

$$e(k+1) = (\Phi - GC)\,e(k) + (\Delta - GD - H)\,u(k) \tag{4.8}$$

By selecting H as

$$H = \Delta - GD \tag{4.9}$$

Equation (4.6), with input u(k), results in a state equation without input, of the following form

$$e(k+1) = (\Phi - GC)\,e(k) \tag{4.10}$$

If the original system is observable, by selecting the observer gain G, we can design the observer such that the error approaches zero at a rate that is desired based on the design characteristics.

The unforced system

$$x(k+1) = \Phi\ x \tag{4.11}$$

with the observation equation

$$y(k) = C\ x(k) \tag{4.12}$$

is observable if and only if the rank (N) of the observability test matrix

$$N = [C' \quad \Phi' C' \ldots\ldots . (\Phi)^{k-1} C'] \tag{4.13}$$

is equal to k, the order of the system [Ref.5].

The observer design begins by converting the state equation for the seeker head, Equation (3.27), from continuous time to discrete time representation. With a sampling time of 0.05 and an unknown disturbance w(k) the equation is given as

$$\begin{bmatrix} \beta(k+1) \\ \beta(k+1) \end{bmatrix} = \begin{bmatrix} 0.9098 & 0.0303 \\ -3.0327 & 0.3033 \end{bmatrix} \begin{bmatrix} \beta(k) \\ \beta(k) \end{bmatrix} + \begin{bmatrix} 0.0902 \\ 3.0327 \end{bmatrix} \sigma(k) + \begin{bmatrix} 0.0013 \\ 0.0513 \end{bmatrix} w(k) \tag{4.14}$$

with the output equation measuring the seeker head angle rate given as

$$y(k) = [0 \quad 1] \begin{bmatrix} \beta(k) \\ \beta(k) \end{bmatrix} \tag{4.15}$$

The system is observable, so the observer gain can be determined. It is usually desirable to select the eigenvalues of the observer so that the error between the state and estimated state will rapidly become small and will not be highly responsive to noise. Because the system is observable, the eigenvalues of the observer can be arbitrarily positioned.

If the observer gain is high, the observer will quickly drive the initial condition error to zero but will make the observer more susceptible to noise. By choosing a low observer gain, the observer response will be slower but the observer will be less affected by noise or jamming. Previous studies have shown that a good compromise between the two would be to choose continuous time poles of p_1 and p_2 to be between the values of 8.0 to 12.0. Continuous time poles of 10.5 were chosen for the observer which results in the discrete time poles given as

$$p_{01} = p_{02} = e^{(-10.5 \cdot 0.05)} = 0.5916 \tag{4.16}$$

The observer gain matrix is then given as

$$G = \begin{bmatrix} -0.0031 \\ 0.0299 \end{bmatrix} \tag{4.17}$$

The F matrix is equal to $(\Phi - GC)$ and is given as

$$F = \begin{bmatrix} 0.9098 & 0.0303 \\ -3.0327 & 0.3033 \end{bmatrix} - \begin{bmatrix} -0.0031 \\ 0.0299 \end{bmatrix} [0 \;\; 1] \tag{4.18}$$

The H matrix is equal to $(\Delta - GD)$ and with D equal to zero the H matrix reduces to Δ. The observer equation for the seeker head is given as

$$[\hat{\beta}(k+1)] = \begin{bmatrix} 0.9098 & 0.0297 \\ -3.0327 & 0.3094 \end{bmatrix} [\hat{\beta}(k)] + \begin{bmatrix} -0.0031 \\ 0.0299 \end{bmatrix} y(k) + \begin{bmatrix} 0.0902 \\ 3.0327 \end{bmatrix} \sigma(k) \tag{4.19}$$

C. SIMULATION RESULTS

The two previous scenarios given in Chapter III were used to test the observer design. Random noise was injected into the seeker head to determine if the Luenberger Observer would reduce the noise of the states that were directly measured by the seeker head and generate the required signals necessary for the autopilot.

1. Constant Velocity Target

The same initial conditions presented in Chapter III for the constant velocity engagement will be used for comparison in this section. Figures 22 through 27 are plots of the constant velocity engagement without the Luenberger Observer but with a random disturbance. Figures 22 and 23 are the plots of the trajectories in the X-Y and X-Z planes, respectively. The miss distance was 15.6 feet. The trajectory in the Y-Z plane depicts how the noise input to the seeker head adversely effects the missiles trajectory. Figures 24 and 25 are the seeker head angle rates in the yaw and pitch plane, respectively and Figure 26 is the plot of the commanded acceleration as a result of the seeker head angle rates being subjected to noise. Figure 27 is the plot of the random noise generated by the program.

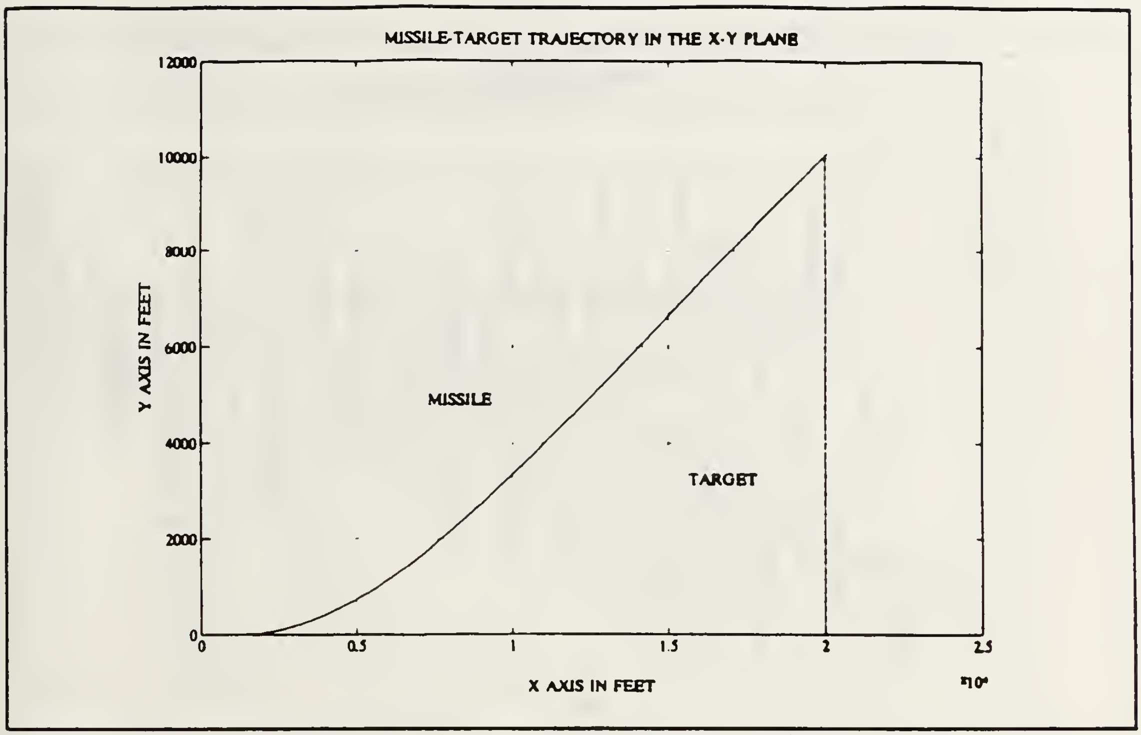

Figure 22. Trajectory in the X-Y Plane With Noise

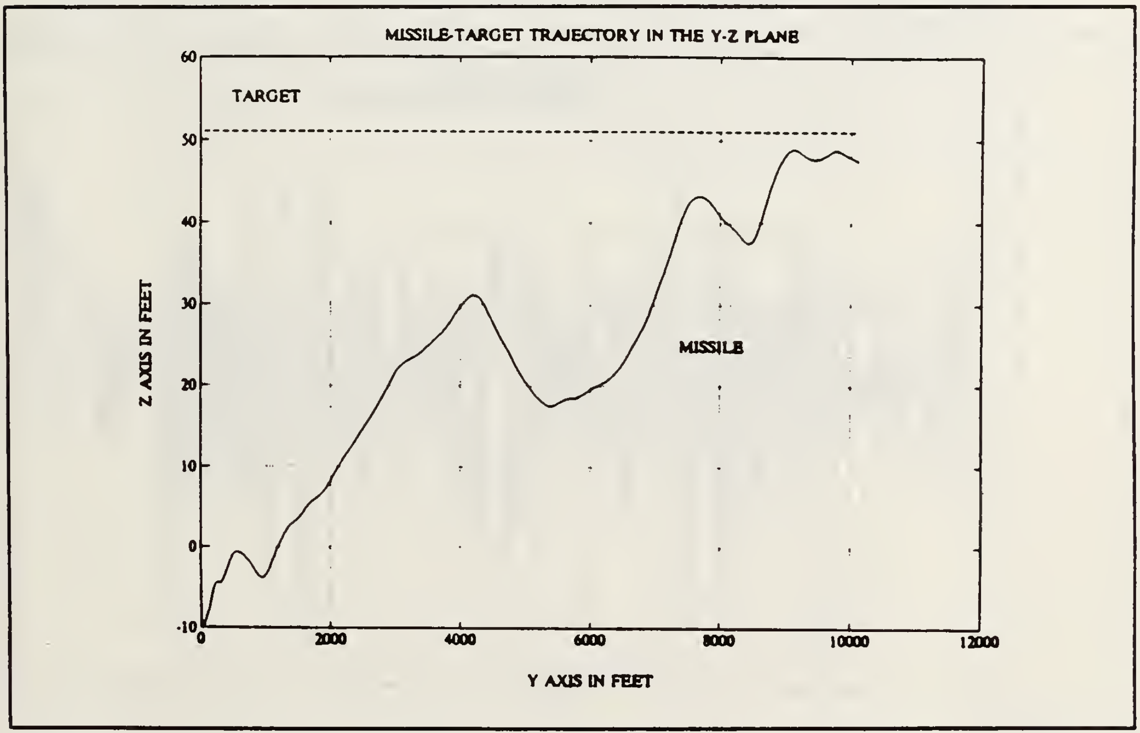

Figure 23. Trajectory in the Y-Z Plane with Noise

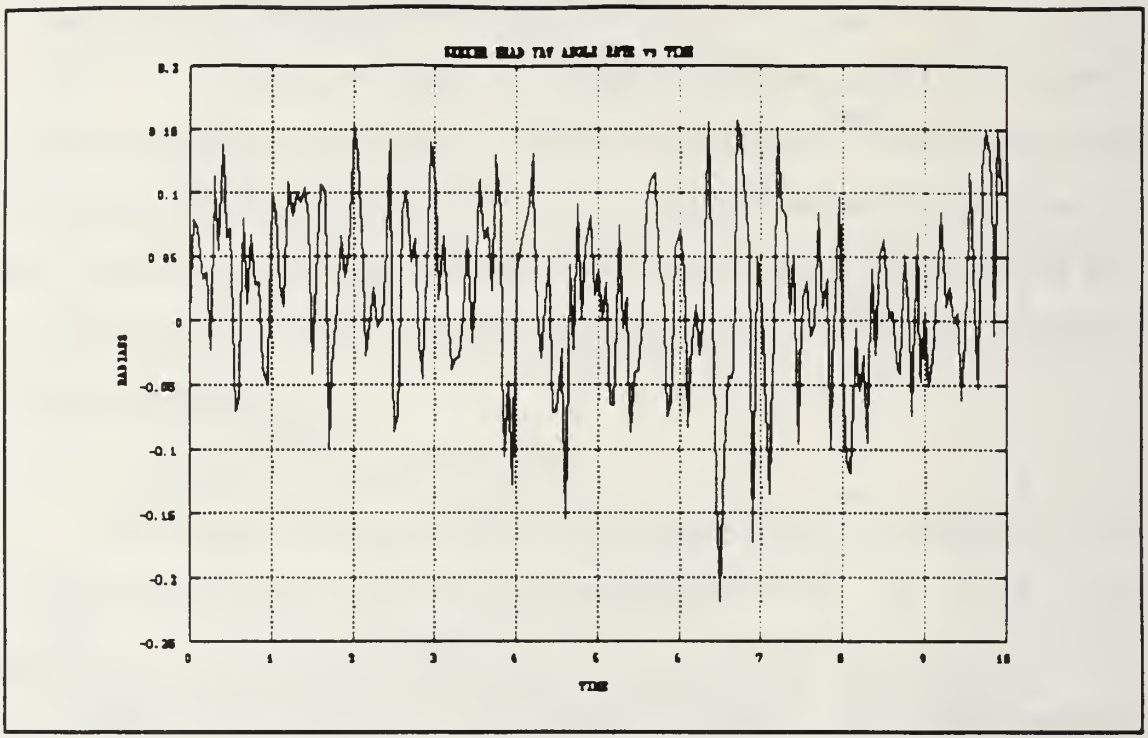

Figure 24. Seeker Head Angle Rate in the Yaw Plane

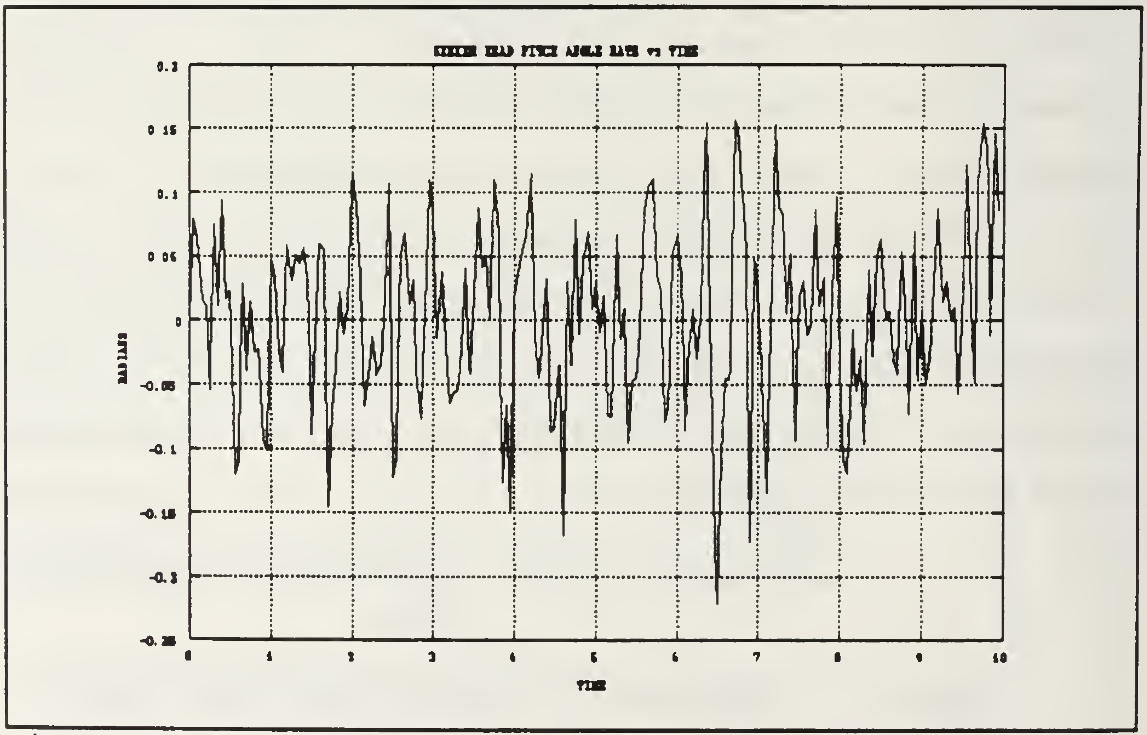

Figure 25. Seeker Head Angle Rate in the Pitch Plane

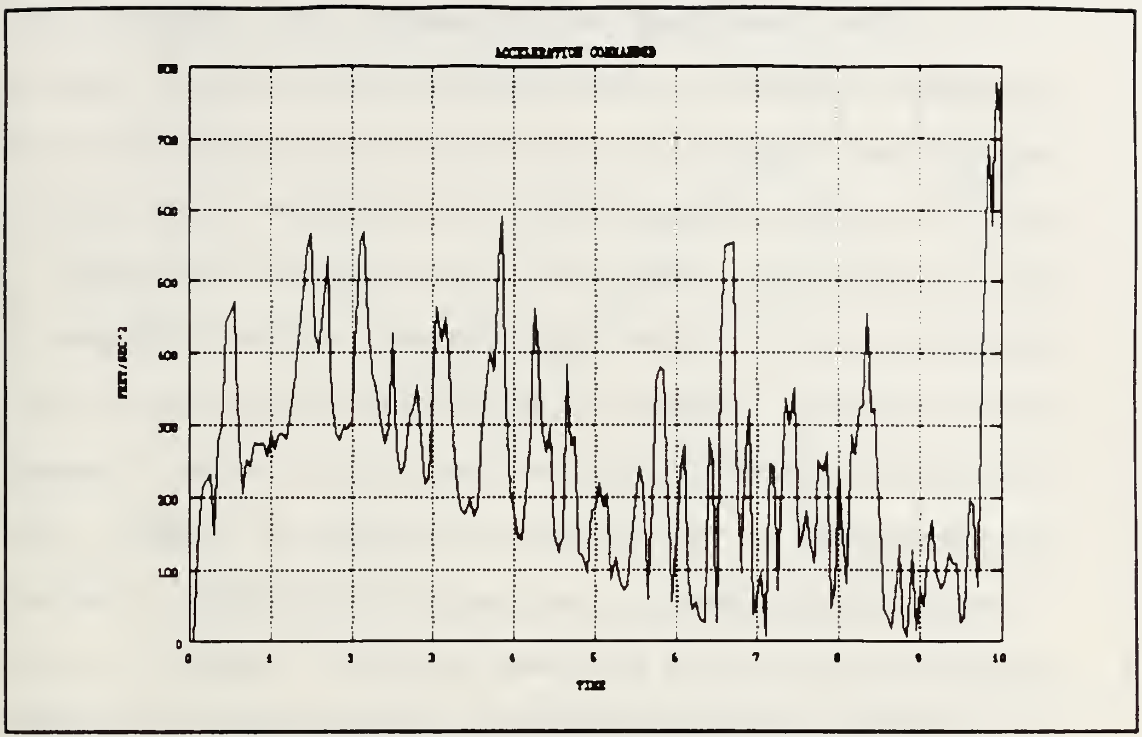

Figure 26. Commanded Acceleration with Noise

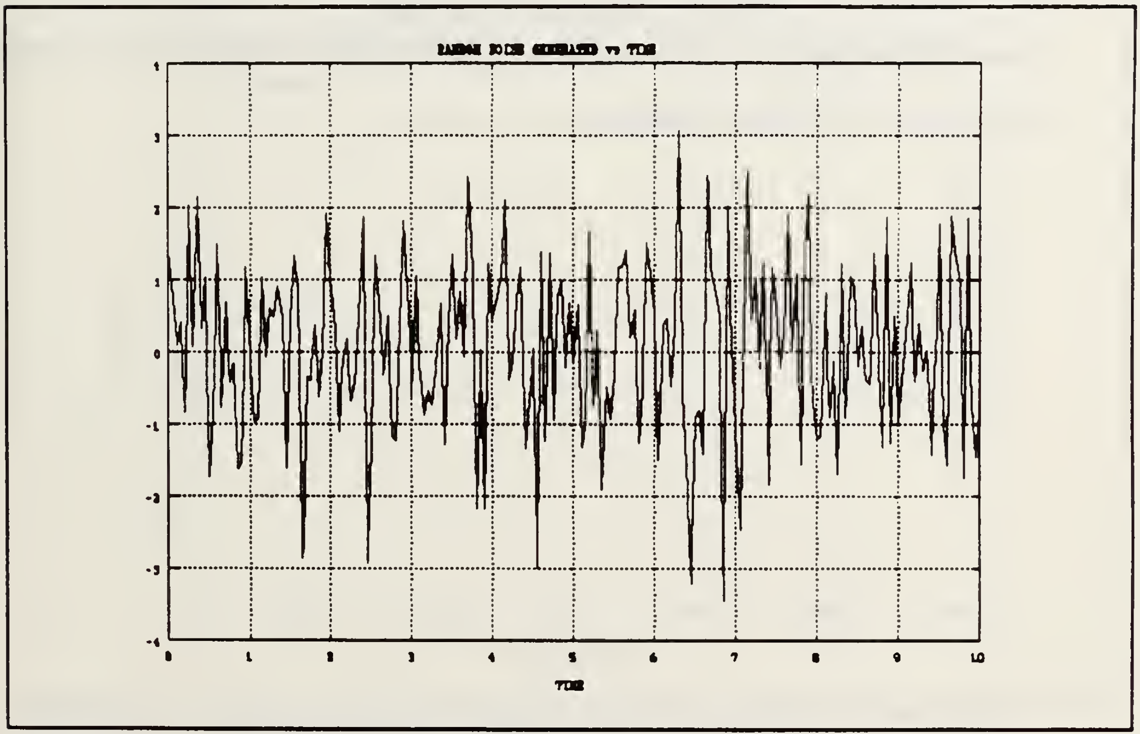

Figure 27. Random Noise Generated vs Time

The following plots depict the results with the observer included in the simulation with the same initial conditions. Figures 28 and 29 are the trajectories in the X-Y and Y-Z planes, respectively. In contrast with the Y-Z plane plot without the observer, the missiles trajectory is less effected by the random disturbance. A miss distance of 1.56 feet occurred. Figures 30 and 31 are the plots of the seeker head angle rates in the yaw and pitch plane, respectively. When compared to the angle rate plots of Chapter III, it is clear that the observer has reduced the effect of noise to the point that the plots are very similar. Figures 32 and 33 are the plots of the commanded acceleration and the random noise generated, respectfully. The commanded acceleration is also less effected by the random disturbance when the observer is made part of the missile.

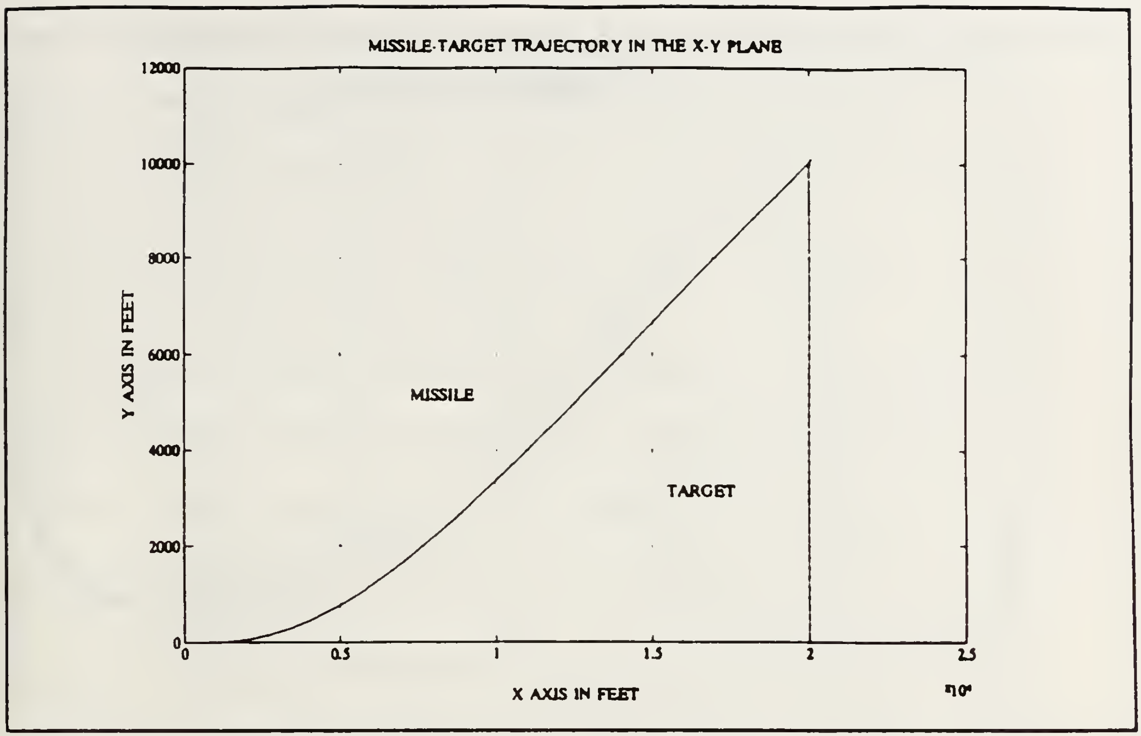

Figure 28. Trajectory in the X-Y Plane with Observer

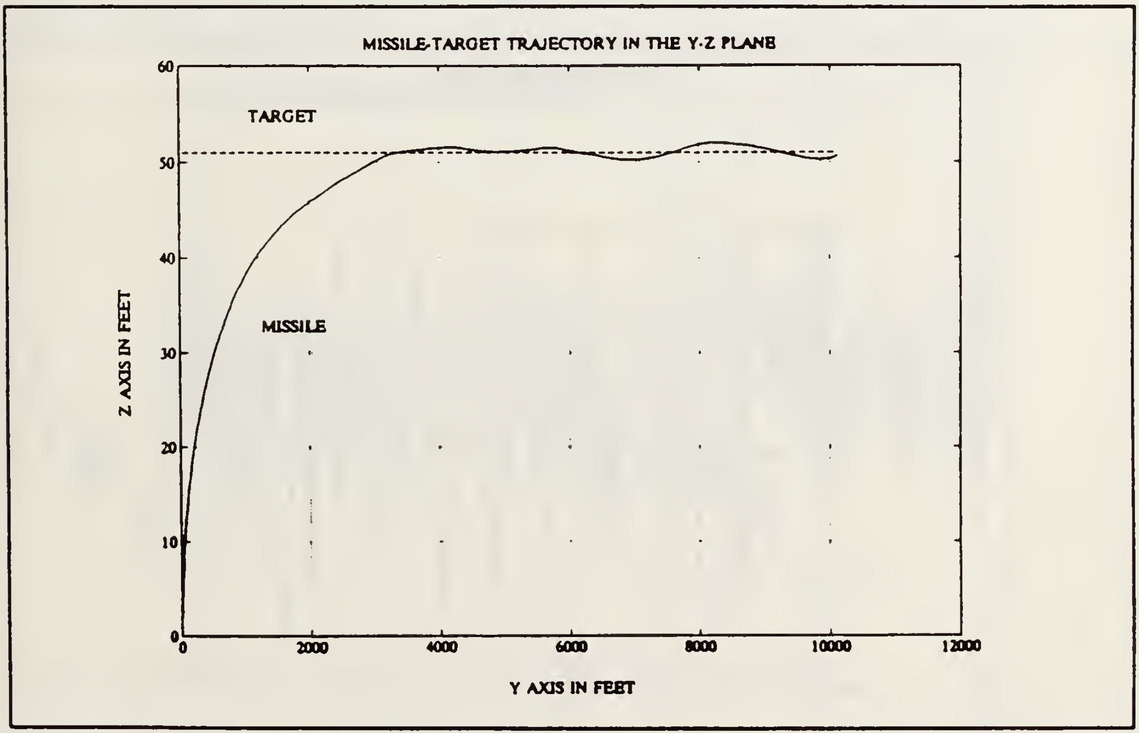

Figure 29. Trajectory in the Y-Z Plane with Observer

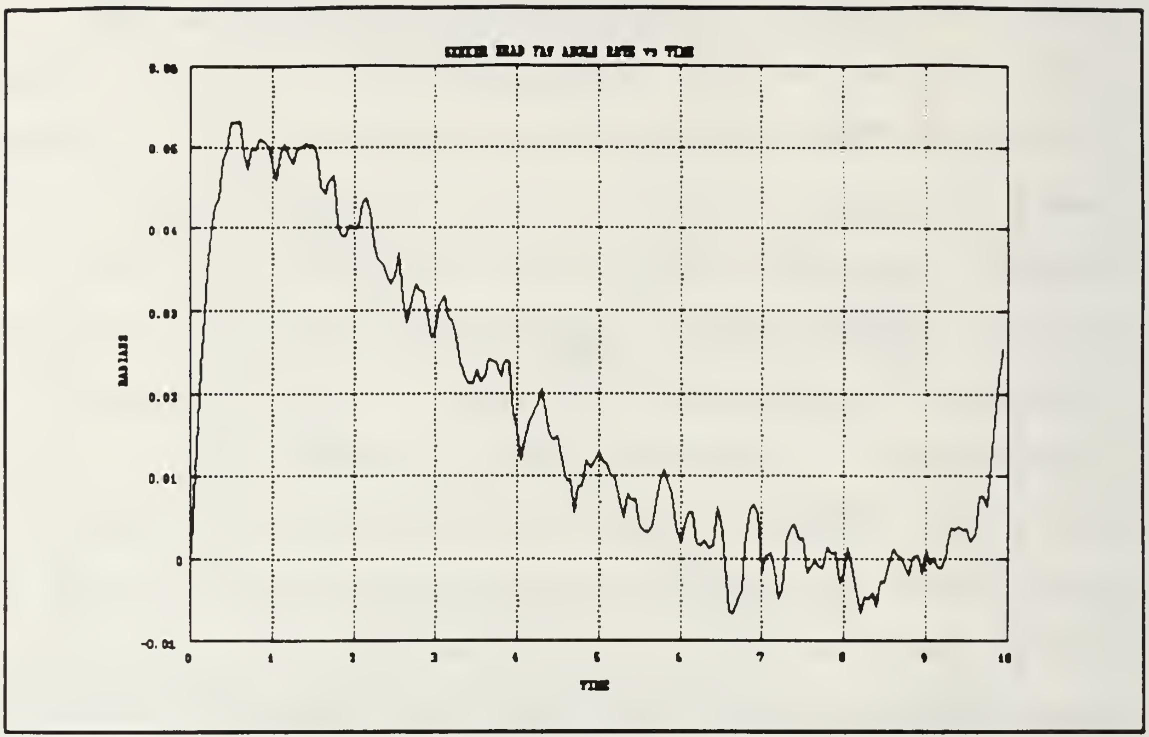

Figure 30. Seeker Head Angle Rate in the Yaw Plane

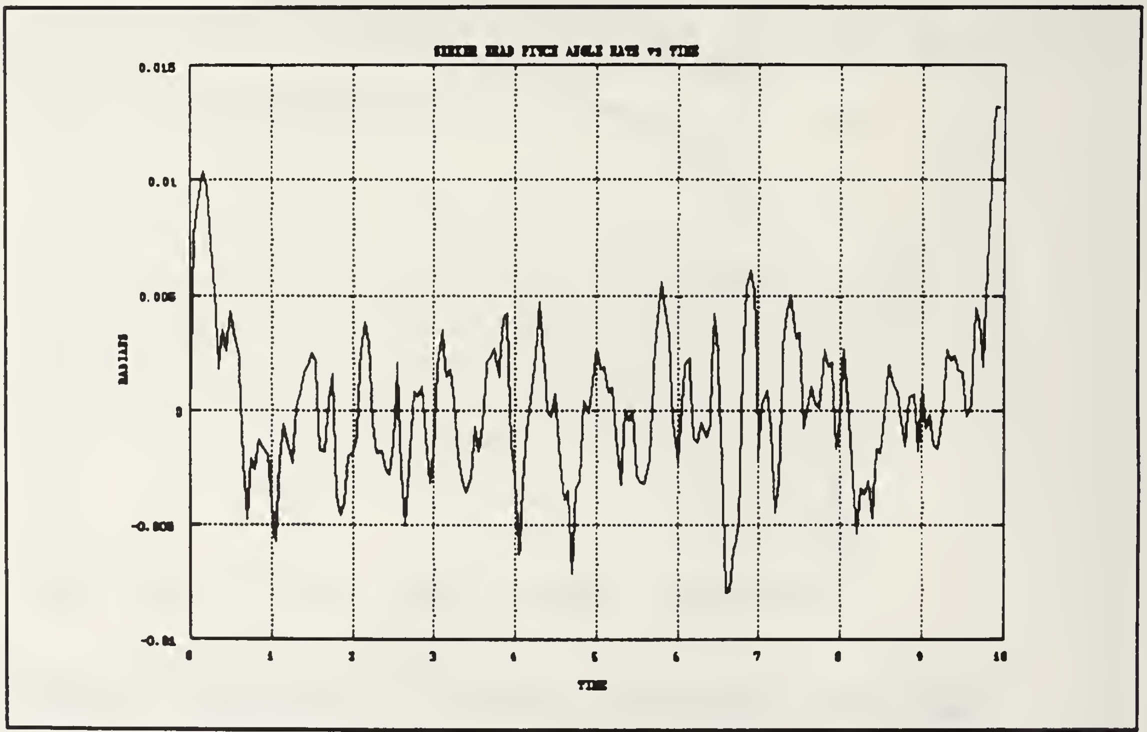

Figure 31. Seeker Head Angle Rate in the Pitch Plane

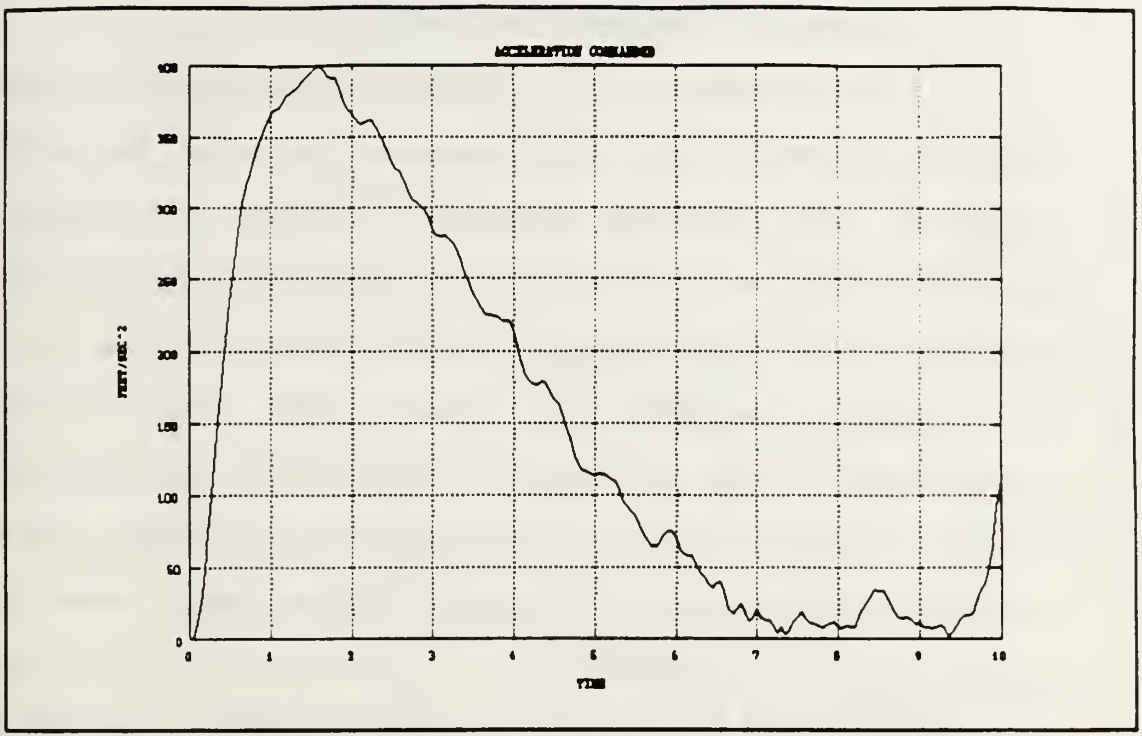

Figure 32. Commanded Acceleration with Observer

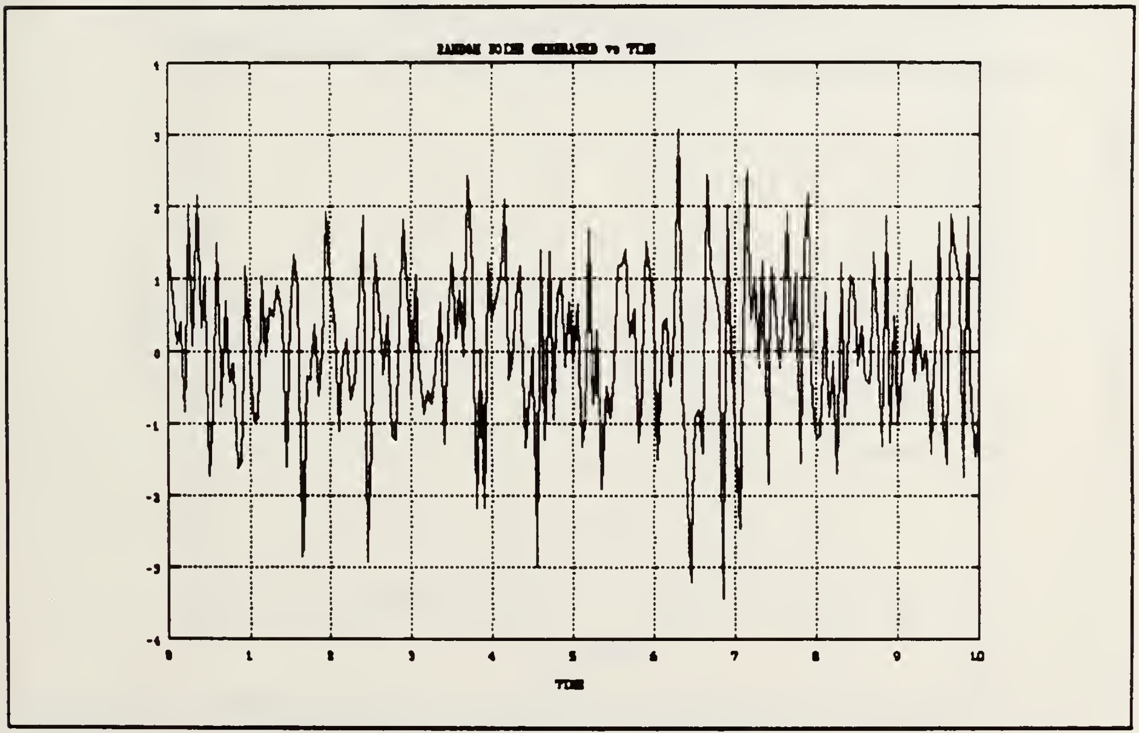

Figure 33. Random Noise Generated vs Time

2. Constant Acceleration Target

The results for the constant acceleration engagement were as expected, with the observer reducing the effects of noise so that intercept occurred. Figures 34 through 39 are plots of the constant acceleration model without the observer. Figures 34 and 35 are plots of the trajectories in the X-Y and X-Z planes, respectively. These plots do not show the variations in the trajectories due to the magnitudes of the scales. Figures 36 and 37 are plots of the seeker head angle rates in the yaw and pitch planes, respectively. When compared to the plots in Chapter III, it is obvious that the missiles performance is degraded. A miss distance of 26.5 feet occurred. Figures 38 and 39 are plots of the commanded acceleration and the random noise generated.

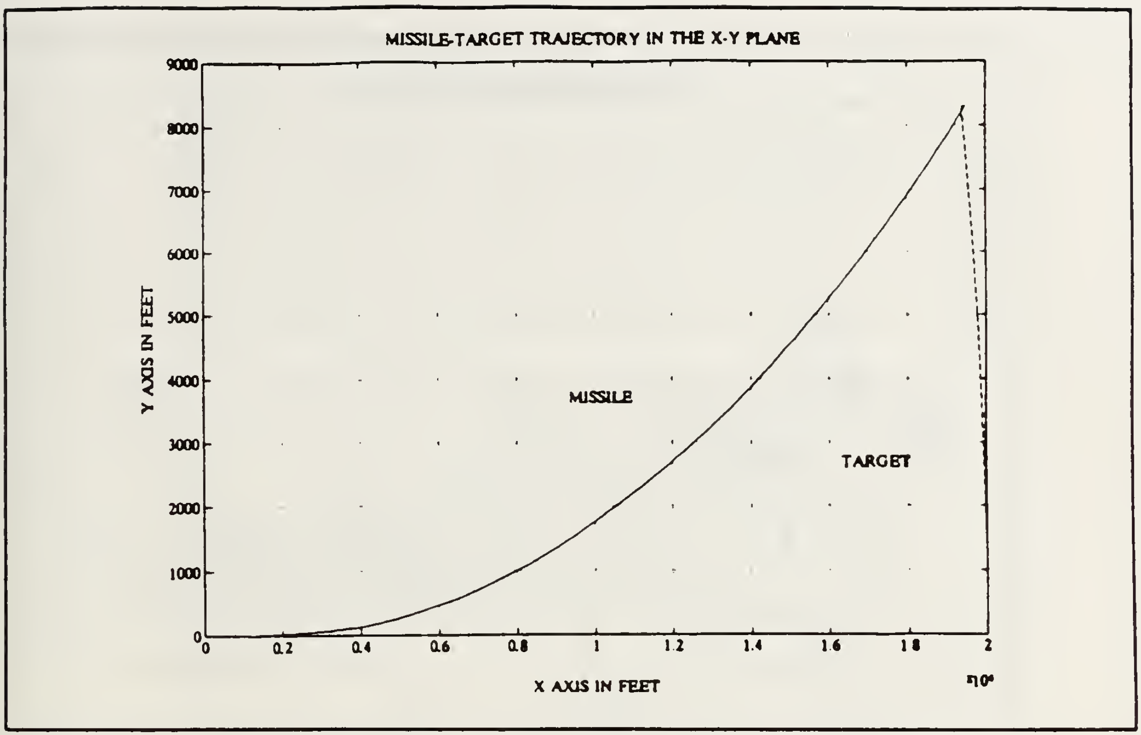

Figure 34. Trajectory in the X-Y Plane with Noise

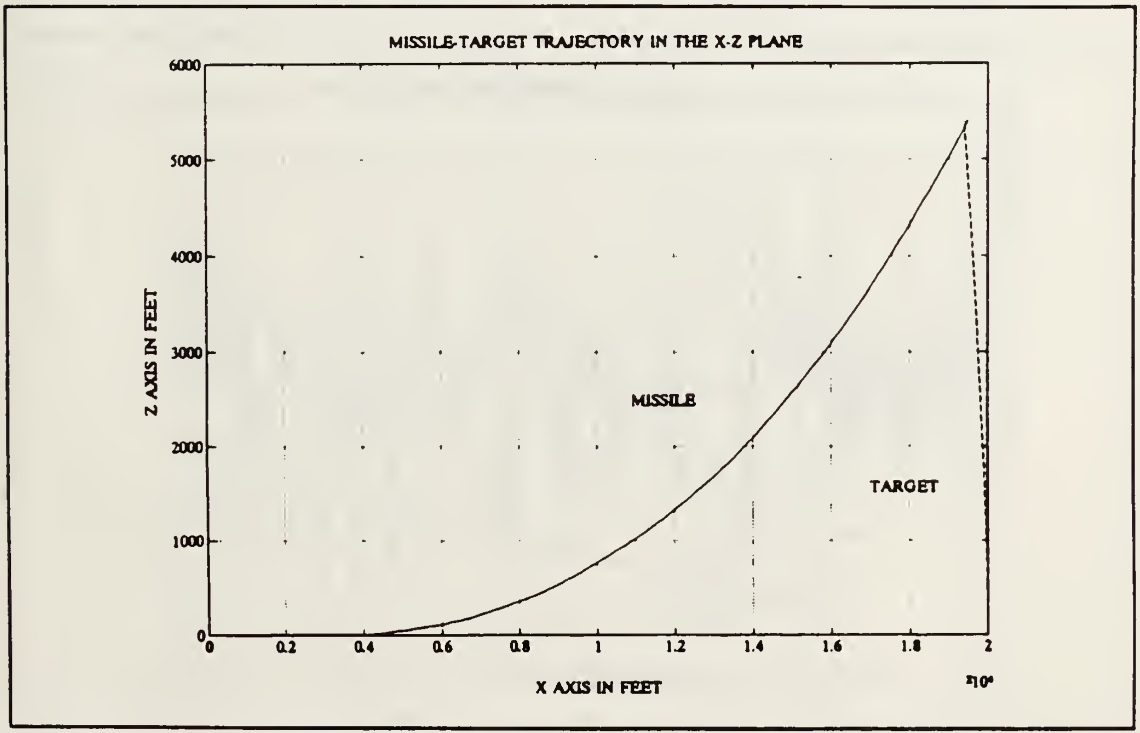

Figure 35. Trajectory in the X-Z Plane with Noise

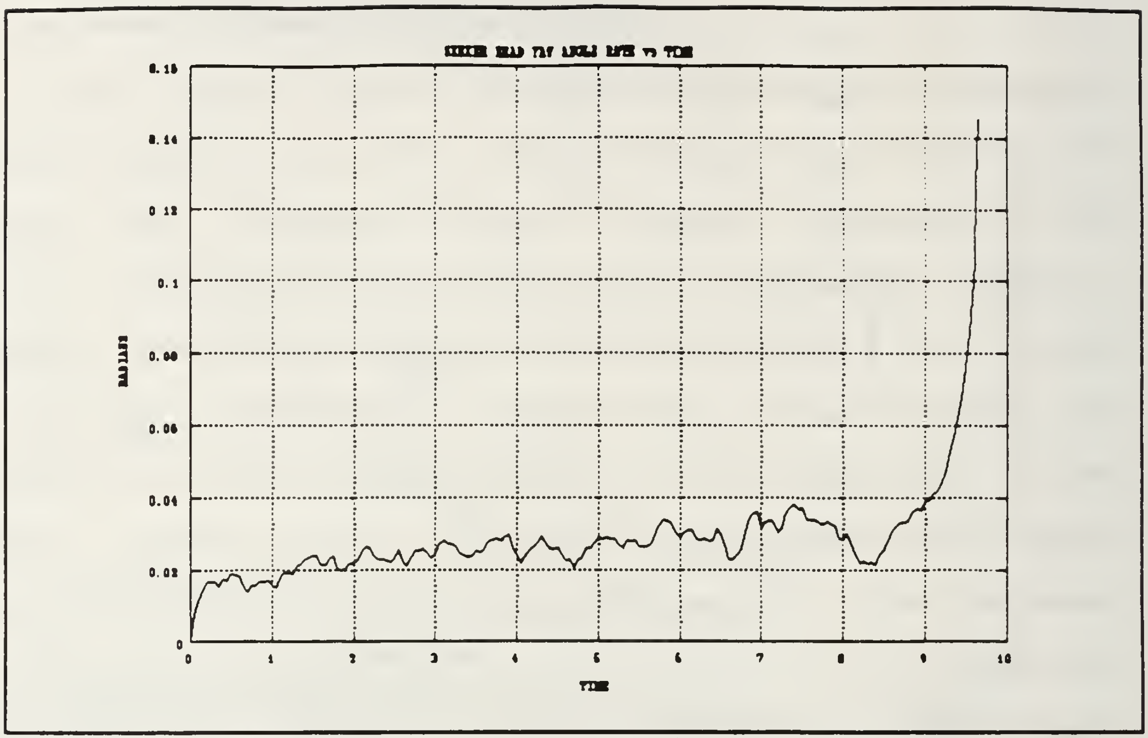

Figure 36. Seeker Head Angle Rate in the Yaw Plane

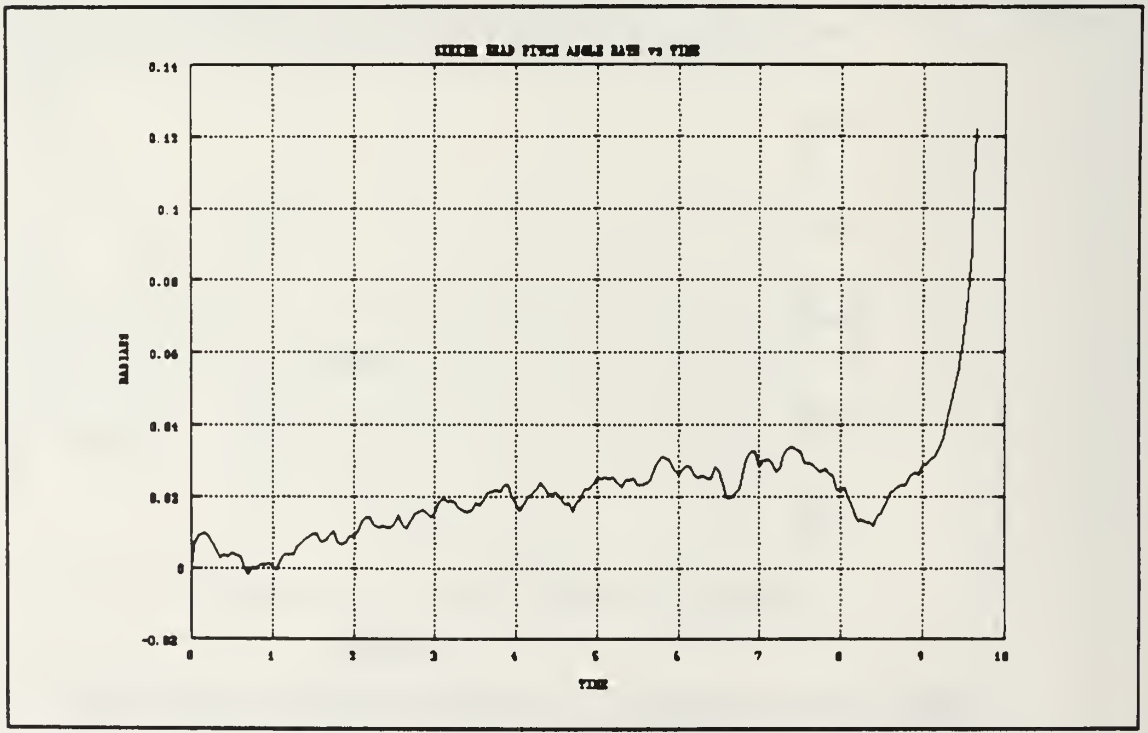

Figure 37. Seeker Head Angle Rate in the Pitch Plane

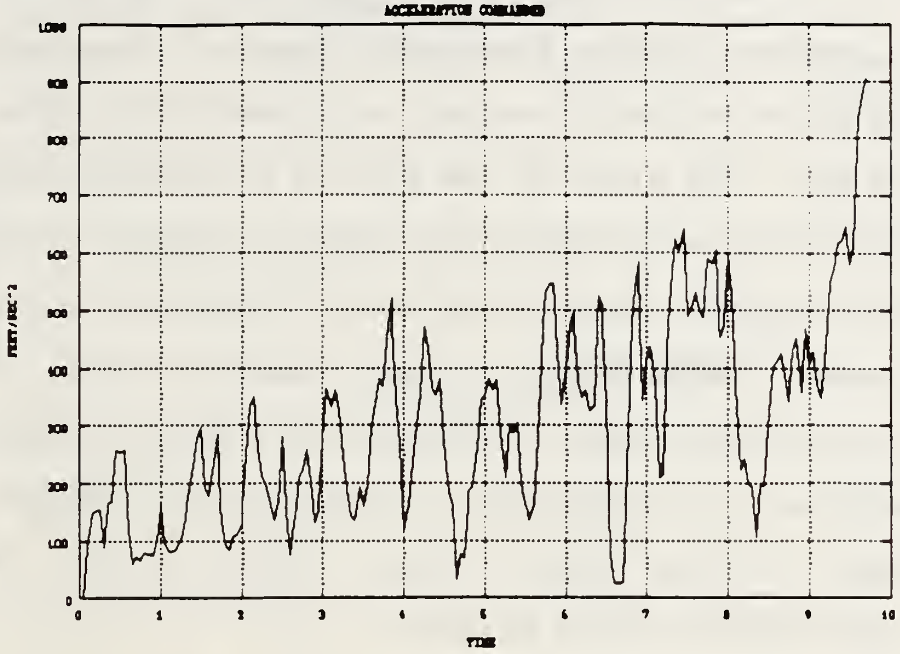

Figure 38. Commanded Acceleration with Noise

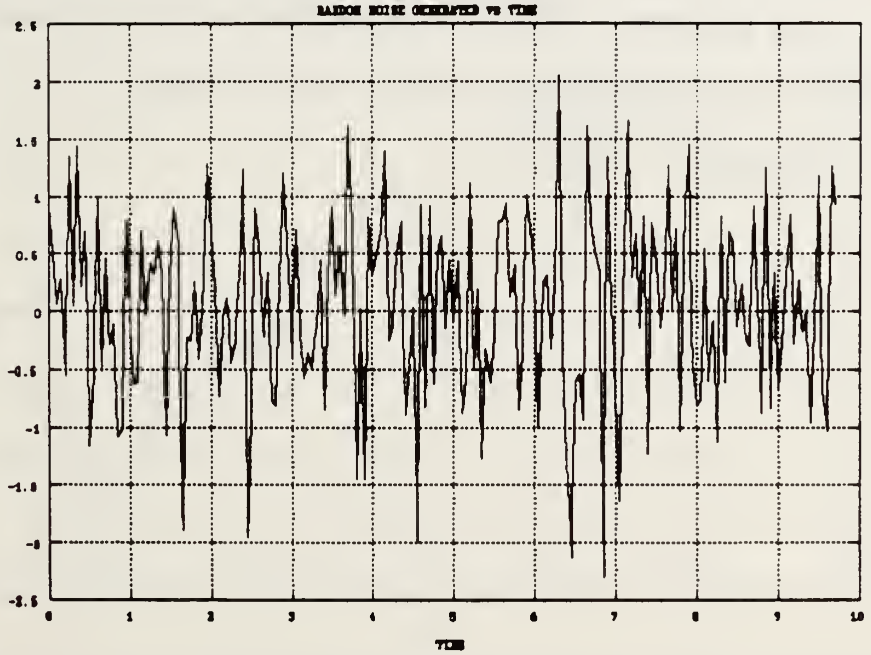

Figure 39. Random Noise Generated vs Time

The following plots depict the constant acceleration engagement with the Luenberger Observer. The observer reduced the noise so that intercept occurred with a miss distance of 4.5 feet. The plots in the X-Y and X-Z planes are not included due to the magnitude of the scales. Figures 40 and 41 are the plots of the seeker head angle rates in the yaw and pitch planes, respectfully. When compared with the constant acceleration model in Chapter III it is obvious that the observer is beneficial in reducing the random disturbance added to the seeker head. Figure 42 is the commanded acceleration of the missile.

In the present day environment, with fast maneuvering targets, the simulations in this chapter demonstrate the need for the observer concept to be included in the missile design and also in computer simulation studies.

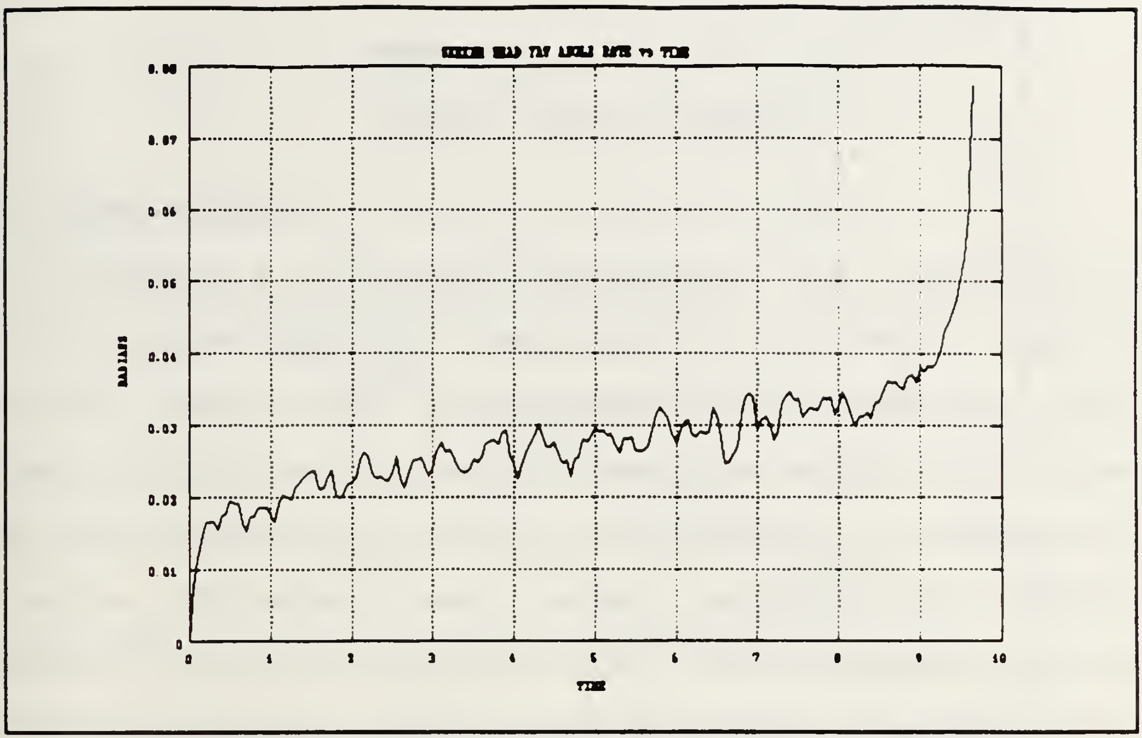

Figure 40. Seeker Head Angle Rate in the Yaw Plane

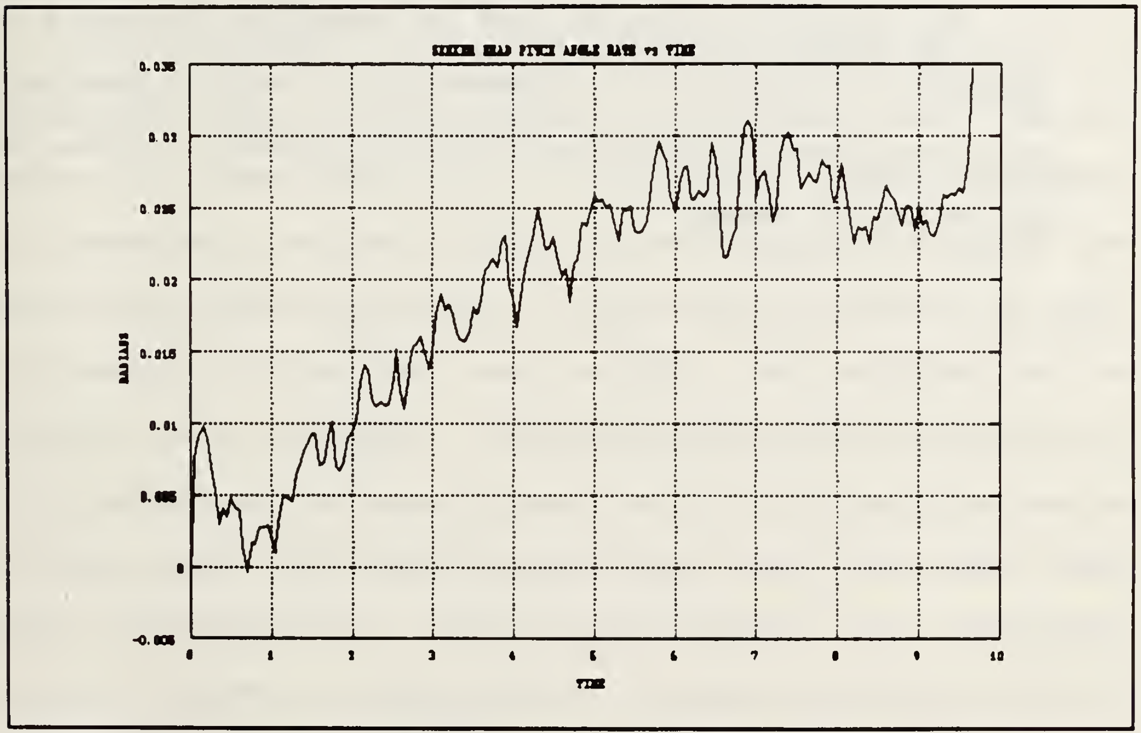

Figure 41. Seeker head Angle Rate in the Pitch Plane

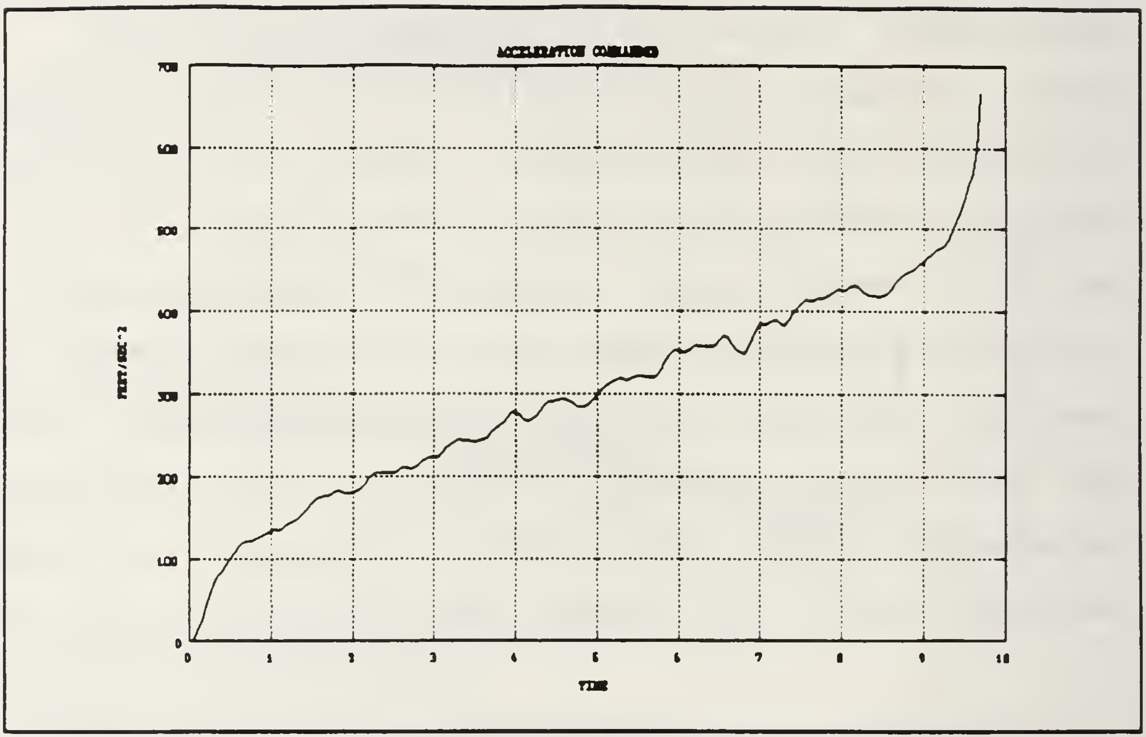

Figure 42. Commanded Acceleration with Observer

The computer program used to generate the plots in this chapter is included in the appendix, of which a portion is the work completed previously by LT F. Lukenbill in our missile and avionics class.

V. GROUND TARGET OBSERVER

A. INTRODUCTION

Air defense is becoming significantly more complex due to the increased speed and manueverability of hostile aircraft and the enemy's use of electronic countermeasures (ECM). Attacking aircraft will employ jamming as well as other forms of ECM to degrade the tracking performance of surface-to-air missiles. Therefore, many modern surface-to-air missile systems have target observers that track incoming aircraft from the ground and uplink the deviations in target position and velocity to the missile.

In this chapter, a simplified Kalman Tracking Filter is developed. The tracker consists of velocity and position estimation. Once the missile is launched, target deviations are computed by subtracting the target observer estimates from the current radar estimates. At specified intervals in time, the errors in position and velocity are uplinked to the missile. After reception, the missile converts the errors into the missile body reference frame via the attitude Euler angles and then into the line-of-sight frame using the seeker head angle transformation. The estimated angular rate deviations are then added to the missile's current estimate and the appropriate acceleration commands are generated.

B. KALMAN FILTER

The purpose of the Kalman Filter is to keep track of the states of a system given a sequence of noisy measurements. This is accomplished by recursively updating an estimate of the state by processing a sequence of noisy observations so that the effect of the measurement errors are reduced as much as possible.

First, previous estimates of the state x, and the covariance matrix P, are extrapolated one step ahead based on the assumed systems dynamics. These values are used to compute a set of optimum weights or Kalman Gains. The gains are applied to the prediction and to a new observation which provides an updated estimate of the state and the covariance matrix.

It is assumed that the target tracker has no a priori information on the target's position or velocity. Also, the error covariance matrix will be set sufficiently large in order that the filter gain will be able to make the estimate converge to the true value and make the simulation independent of any initial estimate states. The random forcing function W_k is zero mean with covariance Q_k and the measurement noise V_k is zero mean with covariance R_k. The Kalman Filter algorithm is given in Equations (5.1) through (5.5) in the order that the filter is implemented in the computer program. The algorithm is given as

$$G_k = P_{(k|k-1)}\ H^T\ (H\ P_{(k|k-1)}\ H^T + R)^{-1} \tag{5.1}$$

$$\hat{x}_{(k|k)} = \hat{x}_{(k|k-1)} + G_k\ (y_k - H\ \hat{x}_{(k|k-1)}) \tag{5.2}$$

$$P_{(k|k)} = (I - G_k\ H)\ P_{(k|k-1)} \tag{5.3}$$

$$\hat{x}_{(k+1|k)} = \Phi\ \hat{x}_{(k|k)} \tag{5.4}$$

$$P_{(k+1|k)} = \Phi\ P_{(k|k)}\ \Phi^T + Q \tag{5.5}$$

From Reference 2, the following covariance matrices are defined. The purpose of the random forcing function covariance matrix Q, is to account for model inaccuracies or for a target that has maneuvered. The covariance matrix Q is given as

$$Q = \begin{bmatrix} \sigma_{\dot{x}}^2 & 0 & 0 \\ 0 & \sigma_{\dot{y}}^2 & 0 \\ 0 & 0 & \sigma_{\dot{z}}^2 \end{bmatrix} \approx \begin{bmatrix} 160^2 & 0 & 0 \\ 0 & 160^2 & 0 \\ 0 & 0 & 160^2 \end{bmatrix} \tag{5.6}$$

Since the actual range of the target is unknown, an estimate of the range is used in determining R_k. The covariance of measurement error matrix will then be based on a three degree field of view radar, observing the target approaching an area 20,000 feet from the ground based tracker.

$$R_k \approx \begin{bmatrix} 500^2 & 0 & 0 \\ 0 & 125^2 & 0 \\ 0 & 0 & 12.5^2 \end{bmatrix} \tag{5.7}$$

The measurement equation y_k is given as

$$y(k) = H\,x(k) + \nu(k) \tag{5.8}$$

where H is the measurement matrix, a linear function of the state vector, and $\upsilon(k)$ the measurement noise.

C. UPLINK

The uplink from the ground target observer consists of deviations in target position and velocity. The deviations are calculated by determining the difference between the actual noisy measurement and the predicted state estimate.

$$y(k) - H\,\hat{x}_{(k|k-1)} \tag{5.9}$$

Upon reception, the missile transforms the deviations into the missile body frame.

$$\begin{bmatrix} V_{BX} \\ V_{BY} \\ V_{BZ} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\sin(\psi) & \cos(\psi) & 0 \\ \sin(\theta)\cos(\psi) & \sin(\theta)\sin(\psi) & \cos(\theta) \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} \tag{5.10}$$

where V_X, V_Y, and V_Z are the velocity deviations in the x, y, and z directions, respecttively, and V_{BX}, V_{BY}, and V_{BZ} are the transformed velocities in the missile reference frame. The range deviations are calculated the same and are defined:

$$\begin{bmatrix} R_{BX} \\ R_{BY} \\ R_{BZ} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\sin(\psi) & \cos(\psi) & 0 \\ \sin(\theta)\cos(\psi) & \sin(\theta)\sin(\psi) & \cos(\theta) \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \\ R_Z \end{bmatrix} \qquad (5.11)$$

where R_X, R_Y, and R_Z are the range deviations and R_{BX}, R_{BY} and R_{BZ} are the transformed range deviations in the x, y, and z directions, respectively. The transformation to the line-of-sight reference frame is accomplished by using the seeker head angle transformation and is given as

$$\begin{bmatrix} V_{SX} \\ V_{SY} \\ V_{SZ} \end{bmatrix} = \begin{bmatrix} \cos(\sigma_P)\cos(\sigma_Y) & \cos(\sigma_P)\sin(\sigma_Y) & -\sin(\sigma_P) \\ -\sin(\sigma_Y) & \cos(\sigma_Y) & 0 \\ \sin(\sigma_P)\cos(\sigma_Y) & \sin(\sigma_P)\sin(\sigma_Y) & \cos(\sigma_P) \end{bmatrix} \begin{bmatrix} V_{BX} \\ V_{BY} \\ V_{BZ} \end{bmatrix} \qquad (5.12)$$

where σ_P and σ_Y are the line-of-sight angles in the pitch and yaw plane, respectively. V_{SX}, V_{SY}, and V_{SZ} are the transformed velocities. The range is transformed using the same transformation matrix and is given as

$$\begin{bmatrix} R_{SX} \\ R_{SY} \\ R_{SZ} \end{bmatrix} = \begin{bmatrix} \cos(\sigma_P)\cos(\sigma_Y) & \cos(\sigma_P)\sin(\sigma_Y) & -\sin(\sigma_P) \\ -\sin(\sigma_Y) & \cos(\sigma_Y) & 0 \\ \sin(\sigma_P)\cos(\sigma_Y) & \sin(\sigma_P)\sin(\sigma_Y) & \cos(\sigma_P) \end{bmatrix} \begin{bmatrix} R_{BX} \\ R_{BY} \\ R_{BZ} \end{bmatrix} \qquad (5.13)$$

where R_{SX}, R_{SY}, and R_{SZ} are the transformed ranges in the x, y, and z directions, respectively. The final process in the uplink transformation is to generate the angular rate deviations. These are calculated by dividing the velocity deviations by the range and the magnitude of the range deviations. The line-of-sight rate in the yaw plane is given as Equation (5.14).

$$\dot{\sigma}_{yaw} = \frac{V_{SZ}}{R + R_D} \tag{5.14}$$

where R_D is given as

$$R_D = \sqrt{(}\, R_{SX}^2 + R_{SY}^2 + R_{SZ}^2 \,) \tag{5.15}$$

and the line-of-sight rate in the pitch plane defined as

$$\dot{\sigma}_{pitch} = \frac{V_{SZ}}{R + R_D} \tag{5.16}$$

D. SIMULATION OF THE TARGET OBSERVER

The constant velocity target in a noisy environment will be the first test case for the ground target observer. Figures 43 through 51 summarize the extensive estimator performance calculations for the constant velocity scenario. Figures 43 through 48 are the histories of the six elements of the state vector. An inspection of these figures indicates that the estimator tracks the system in position and velocity quite well. Figures 49 through 51 are examplese of the Kalman Filter Gains vs time. Figures 52 through 54 are the response of the missile to the uplink commands. The deviations in position and velocity were uplinked at one second intervals. The miss distance parameter was observed to be 0.44 feet.

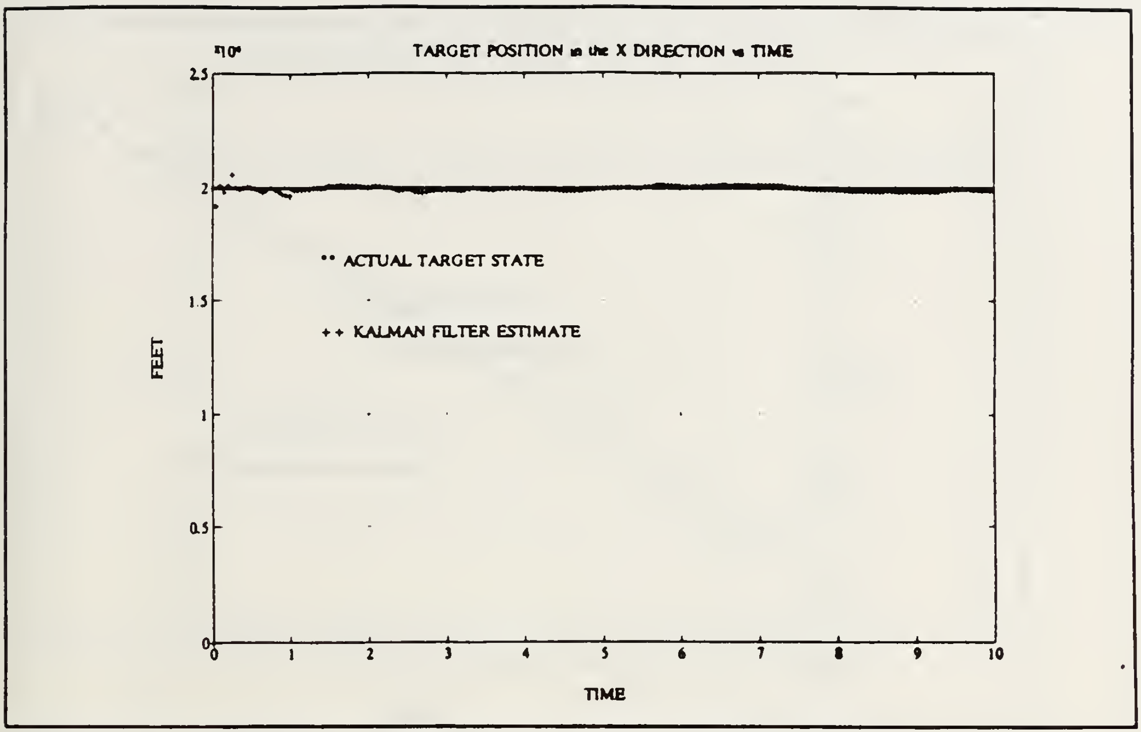

Figure 43. Target Position in the X Direction

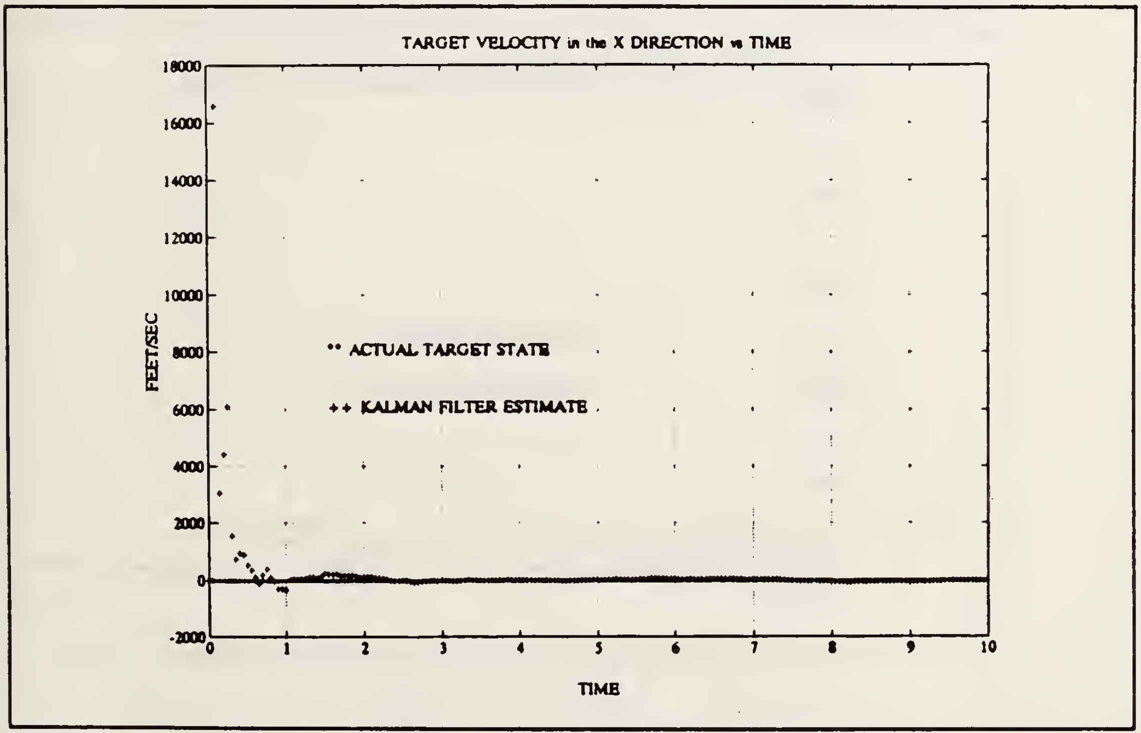

Figure 44. Target Velocity in the X Direction

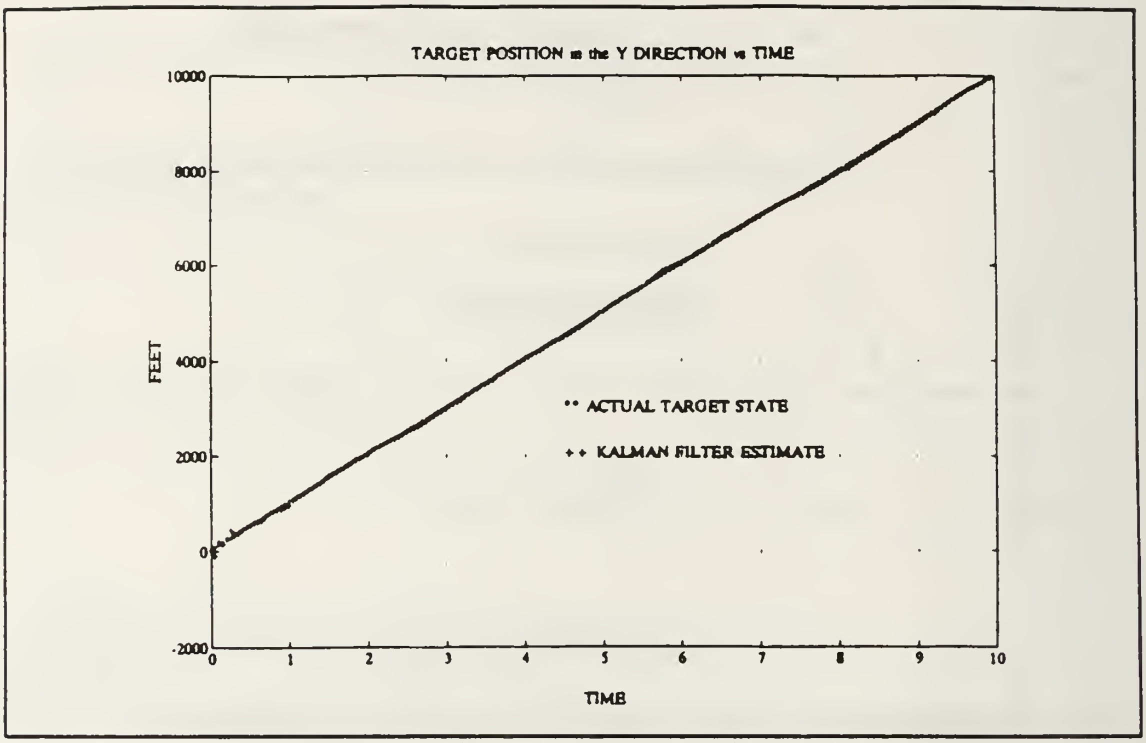

Figure 45. Target Position in the Y Direction

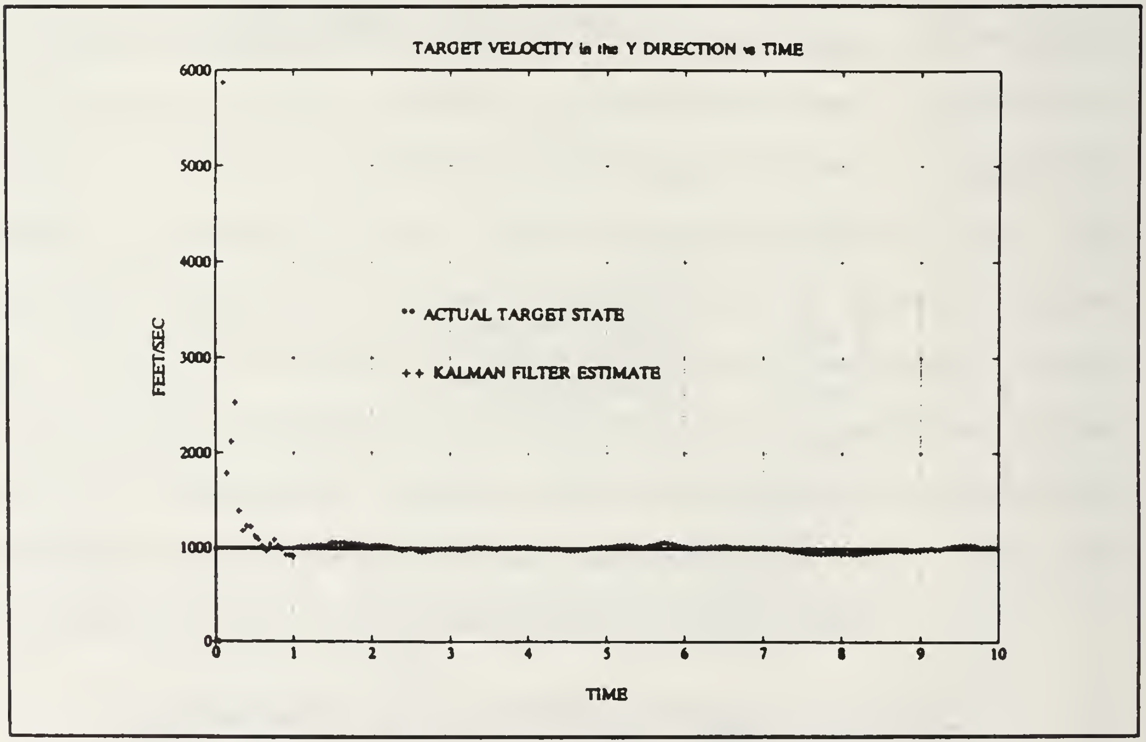

Figure 46. Target Velocity in the Y Direction

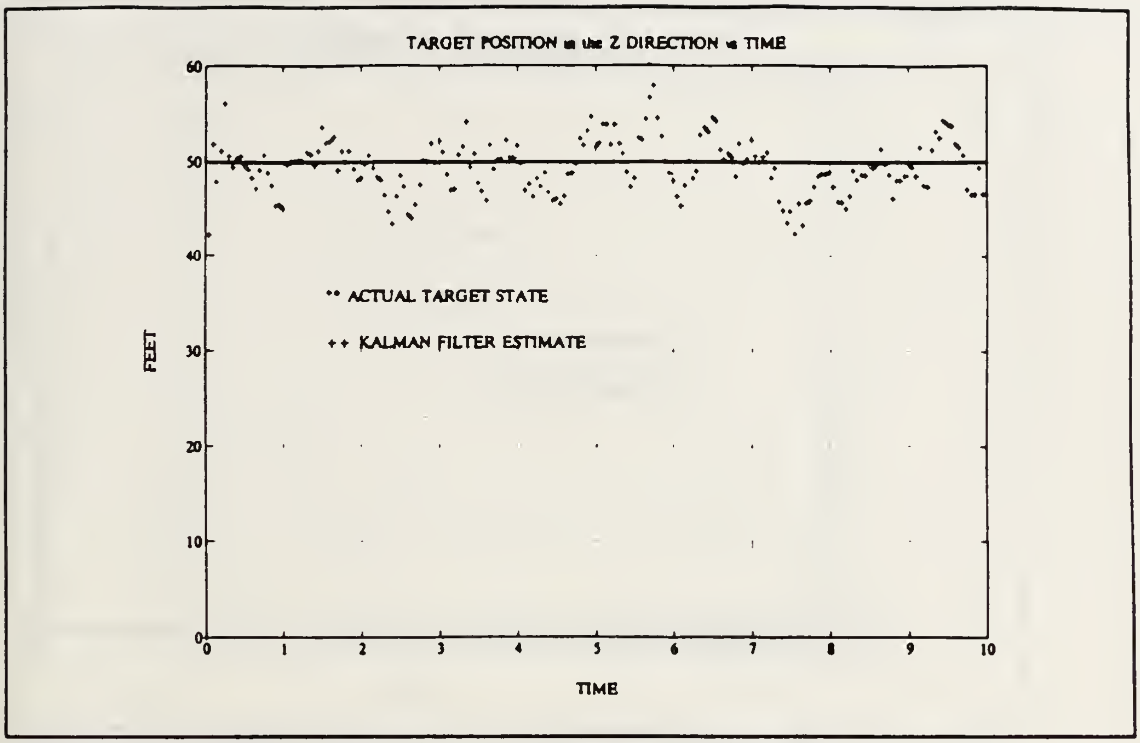

Figure 47. Target Position in the Z Direction

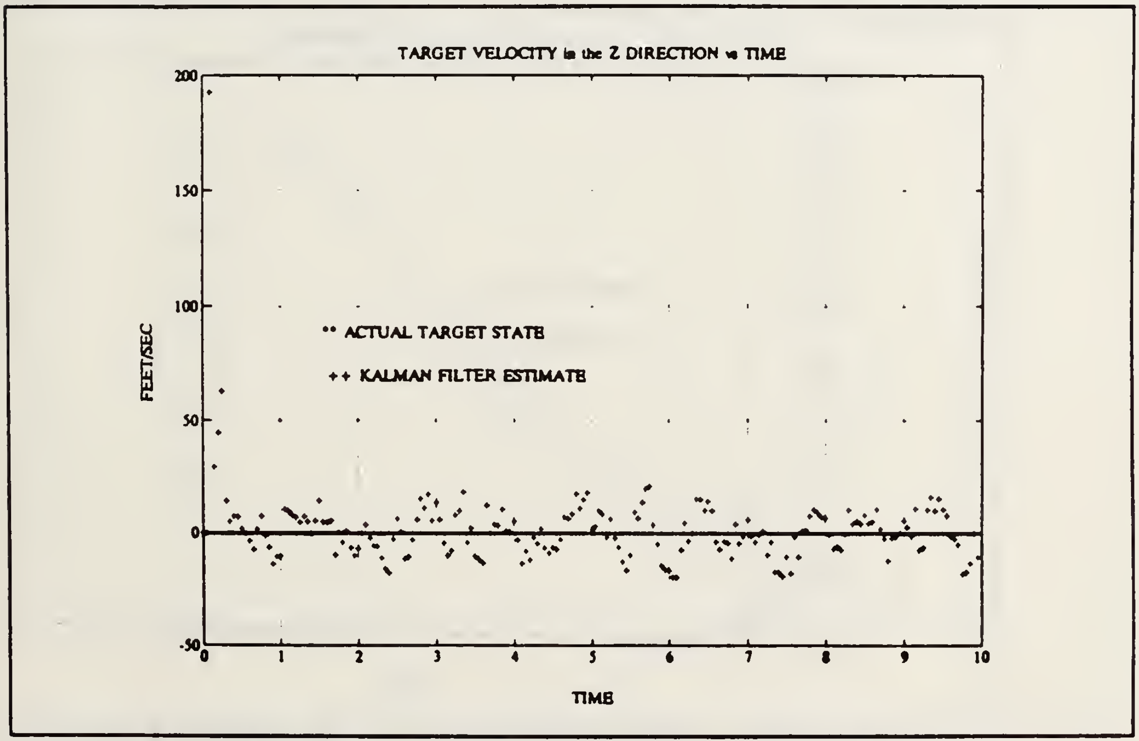

Figure 48. Target Velocity in the Z Direction

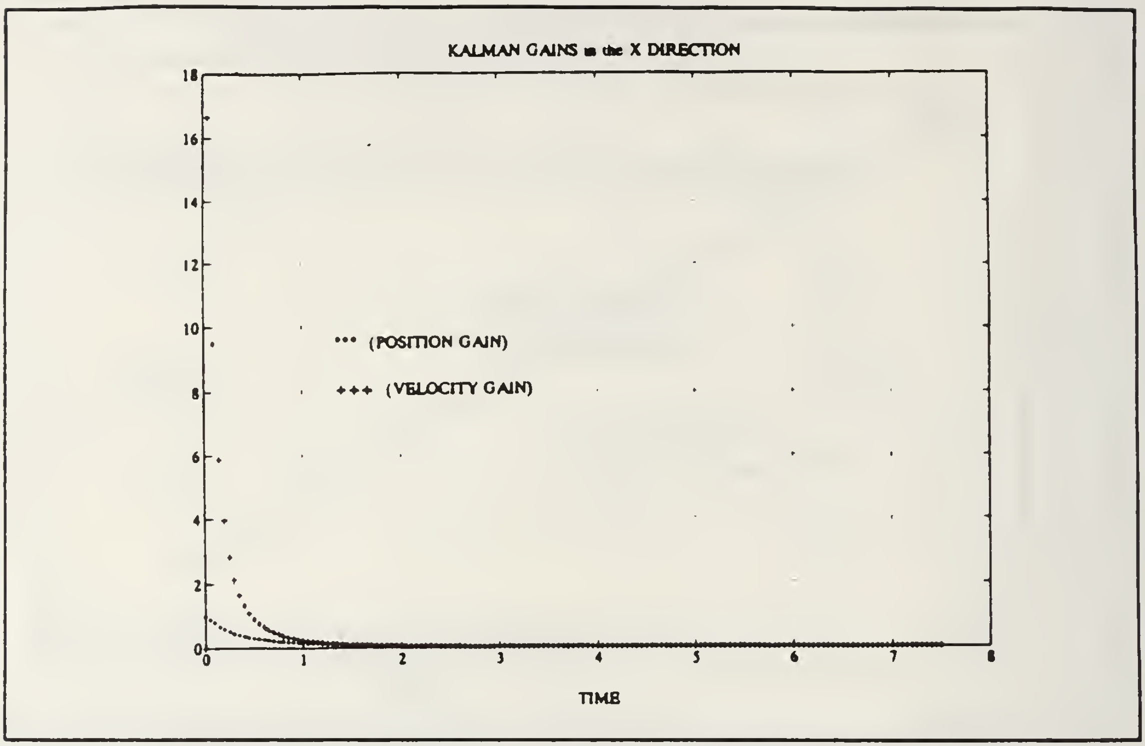

Figure 49. Kalman Gains in the X Direction

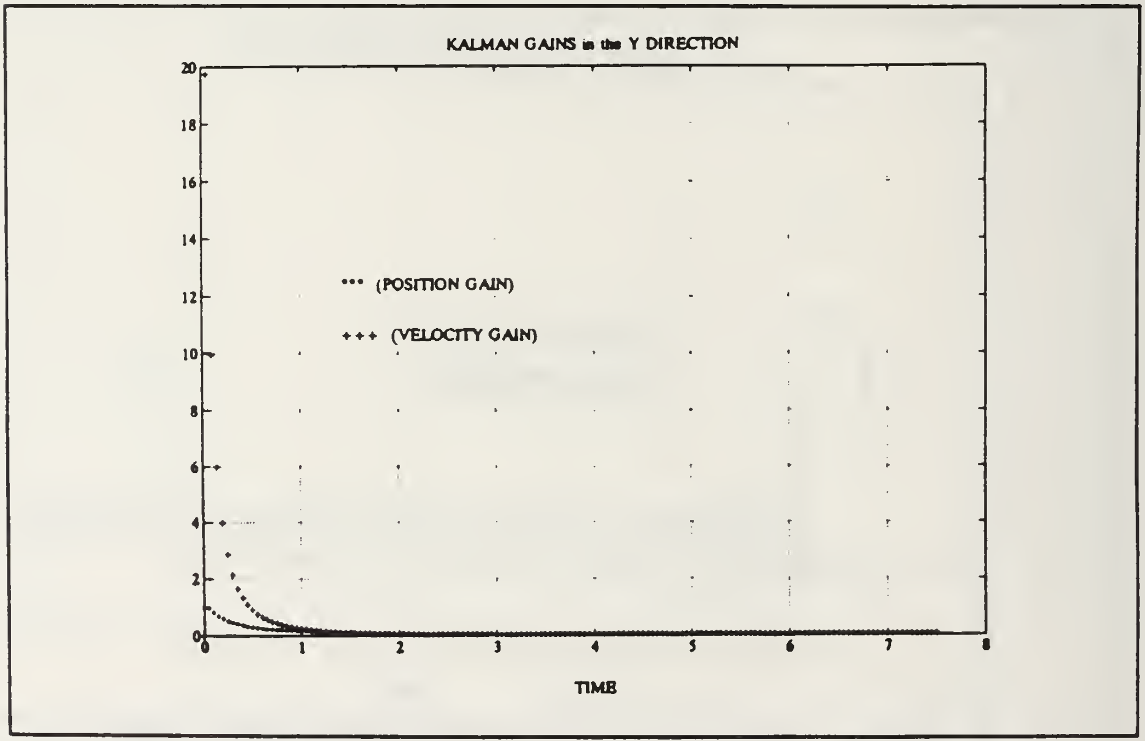

Figure 50. Kalman Gains in the Y Direction

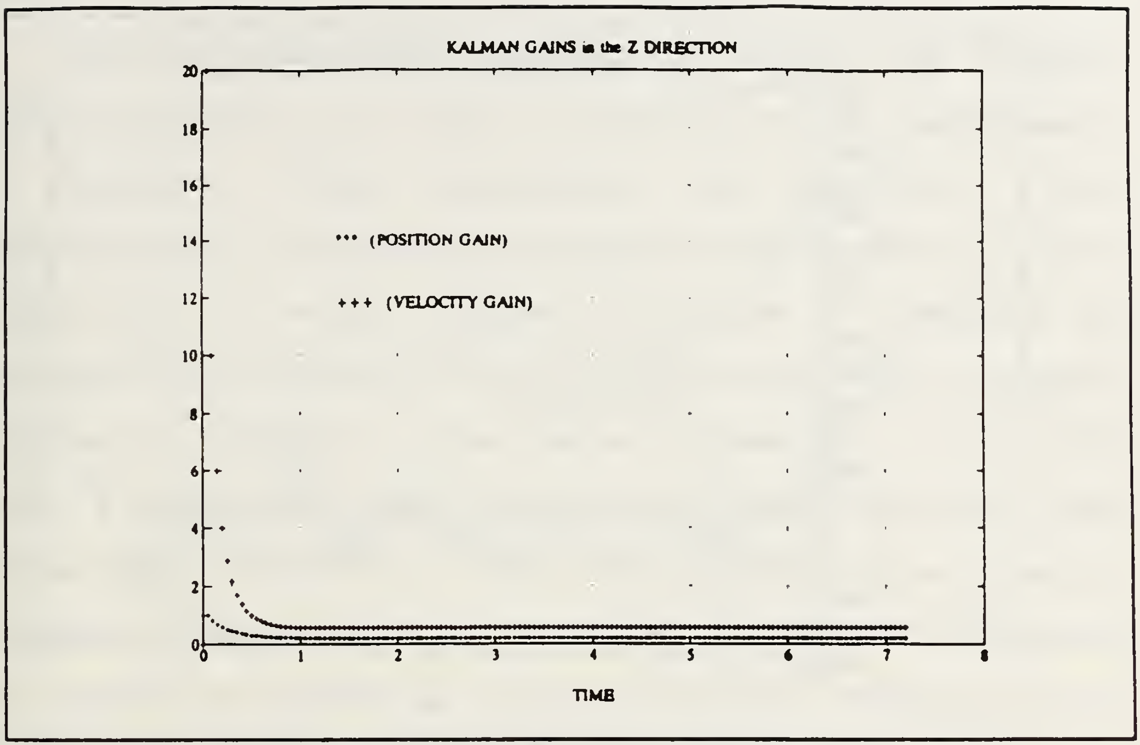

Figure 51. Kalman Gains in the Z Direction

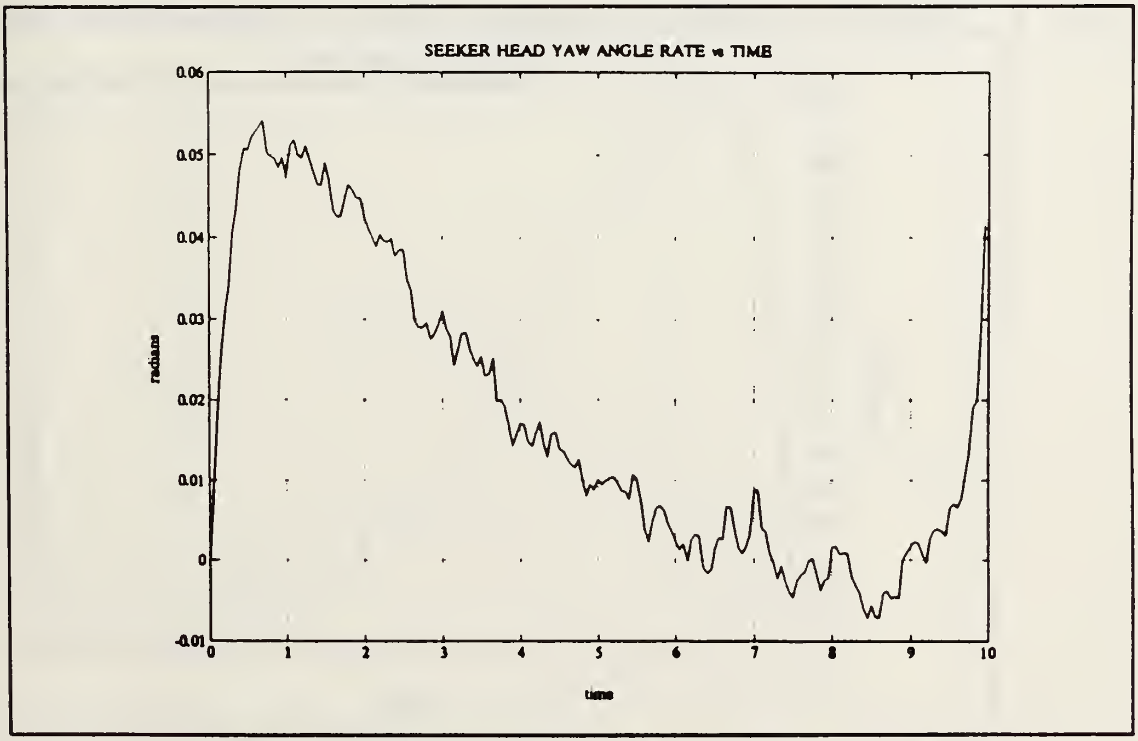

Figure 52. Seeker Head Angle Rate in the Yaw Plane

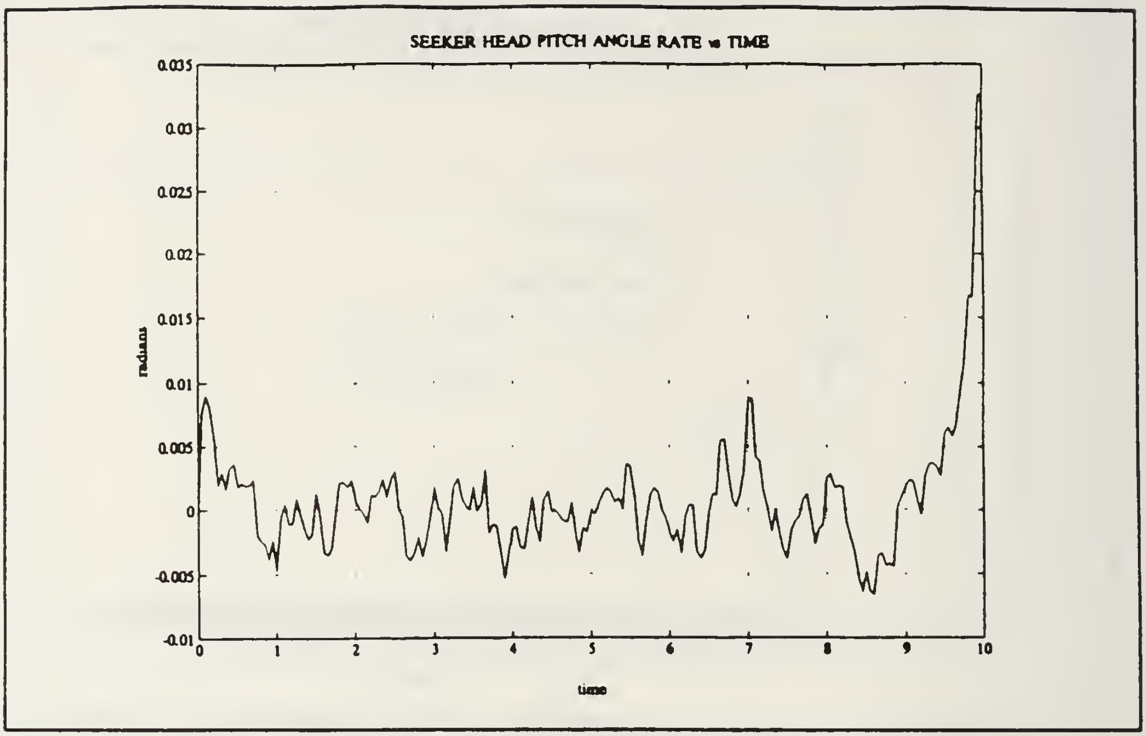

Figure 53. Seeker Head Angle Rate in the Pitch Plane

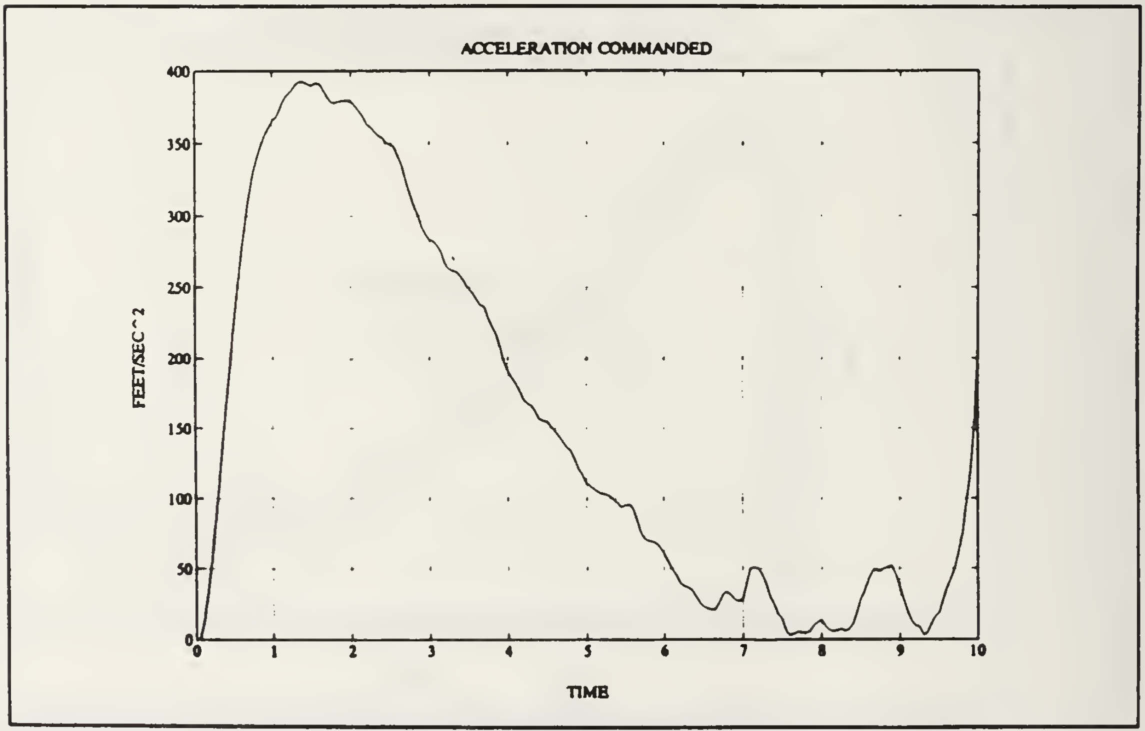

Figure 54. Acceleration Commanded

For the case of a maneuvering target, the constant acceleration model was used to determine filter performance. As expected, filter performance was marginal. Filter convergence to an accuracy attained in the constant velocity target did not occur. Figures 55 through 63 summarize the estimator performance for the constant acceleration scenario. Figures 55 through 60 are the six elements of the state vector. As shown in these plots, the estimator does not track the maneuvering target as well as the constant velocity model. Figures 61 through 63 are the Kalman Filter Gains generated and Figures 64 through 66 are the missile’s response to the uplink commands. The uplinks were transmitted using the same time interval as in the constant velocity scenario. A miss distance of 8.6 feet occurred.

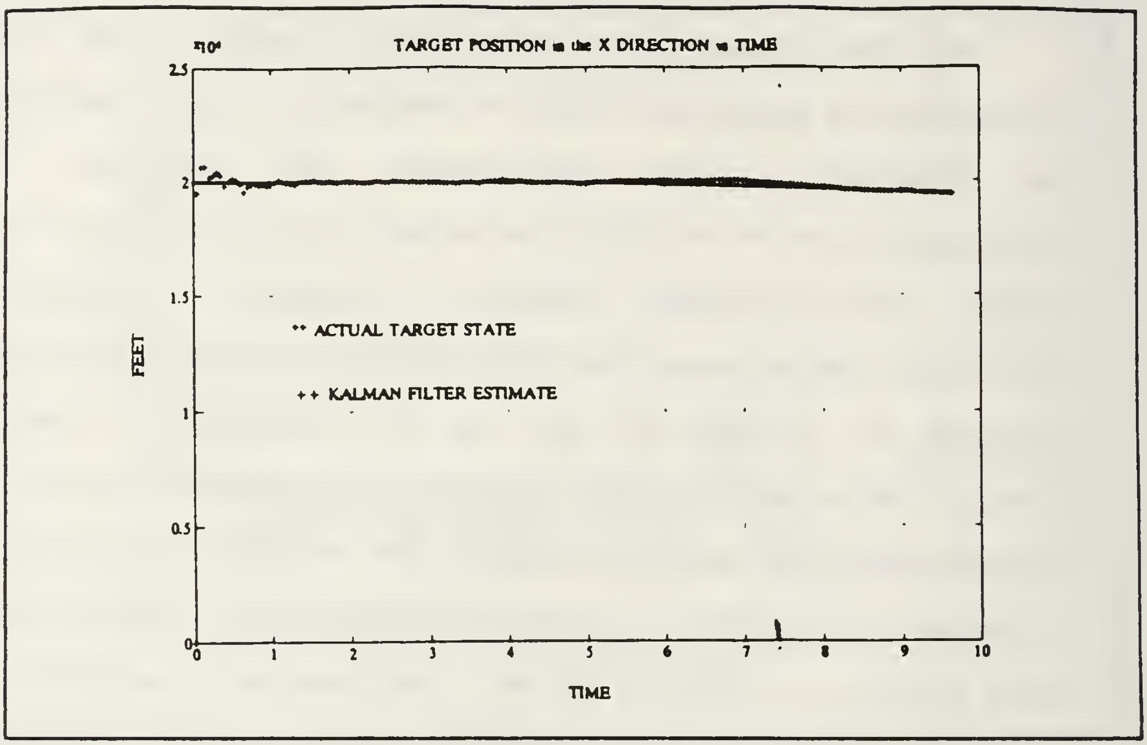

Figure 55. Target Position in the X Direction

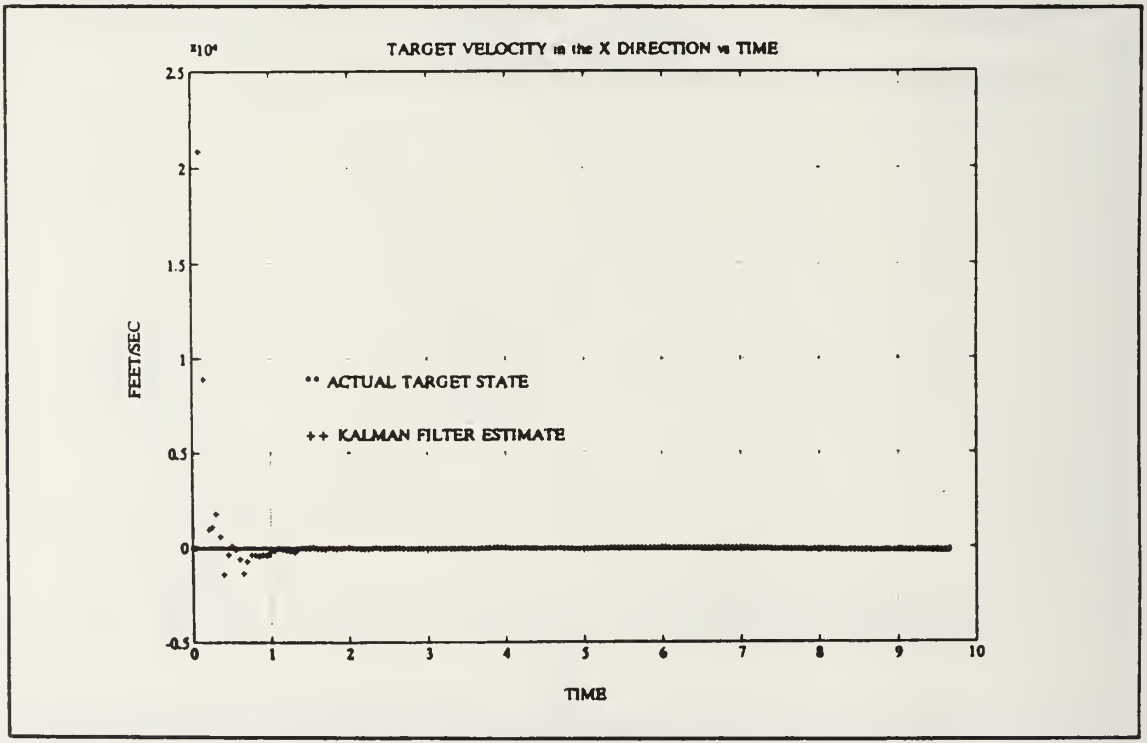

Figure 56. Target Velocity in the X Direction

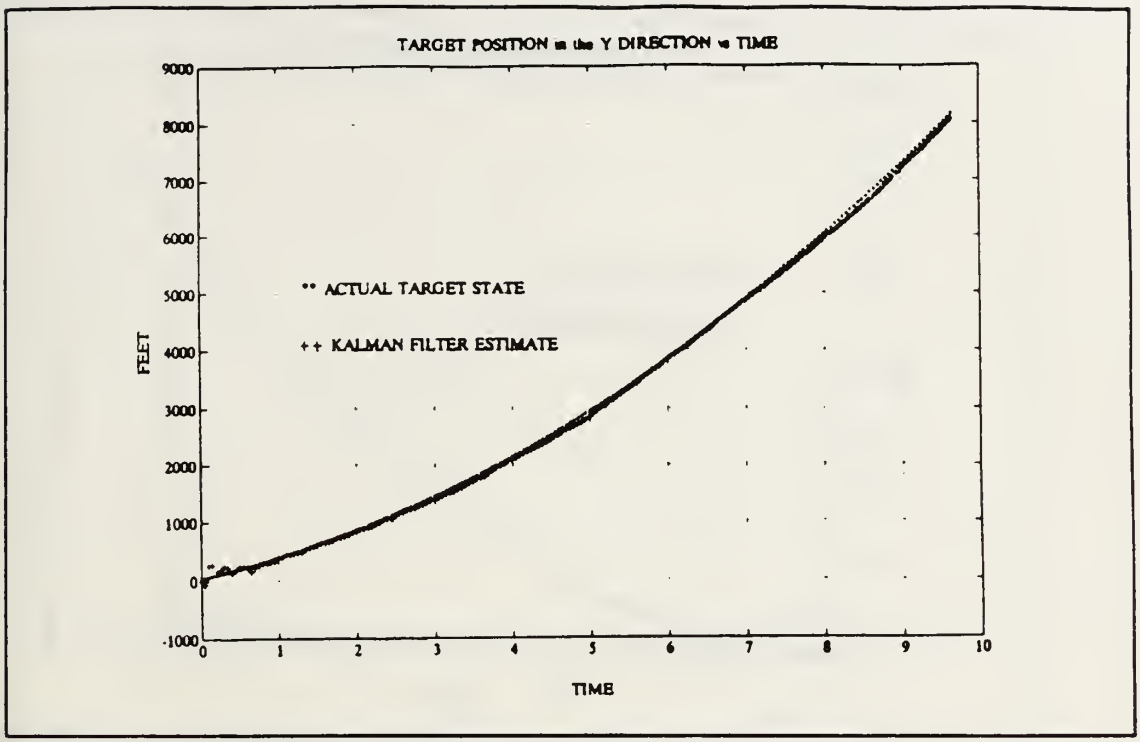

Figure 57. Target Position in the Y Direction

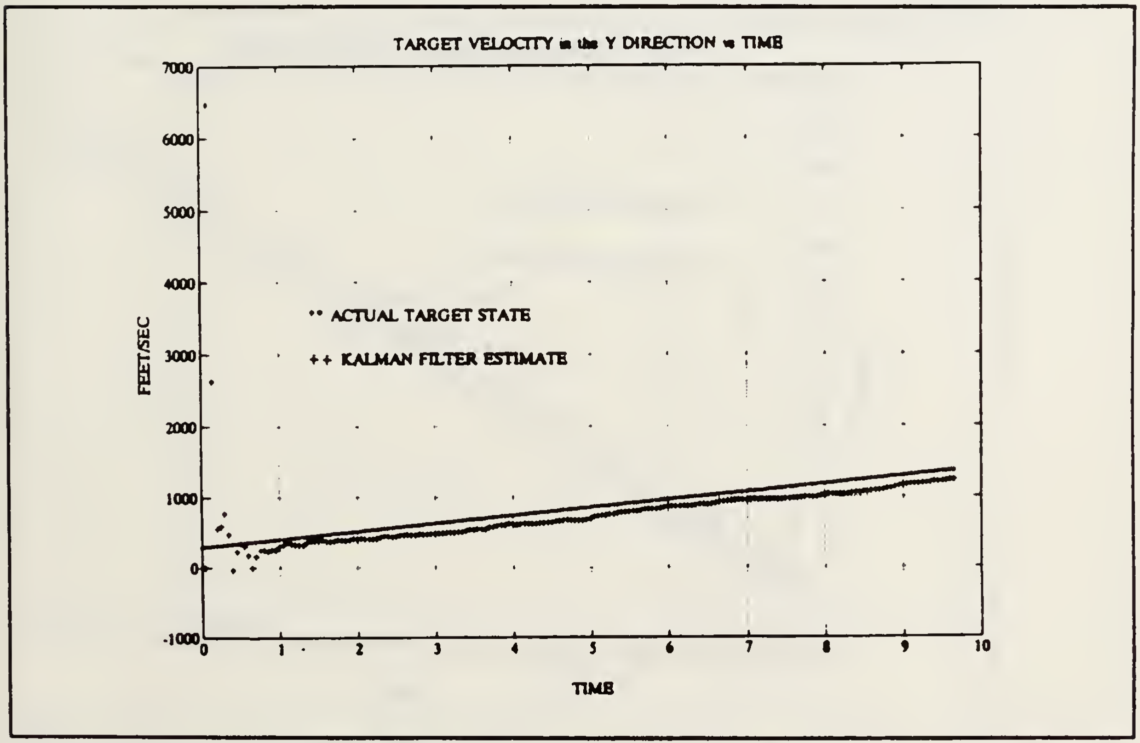

Figure 58. Target Velocity in the Y Direction

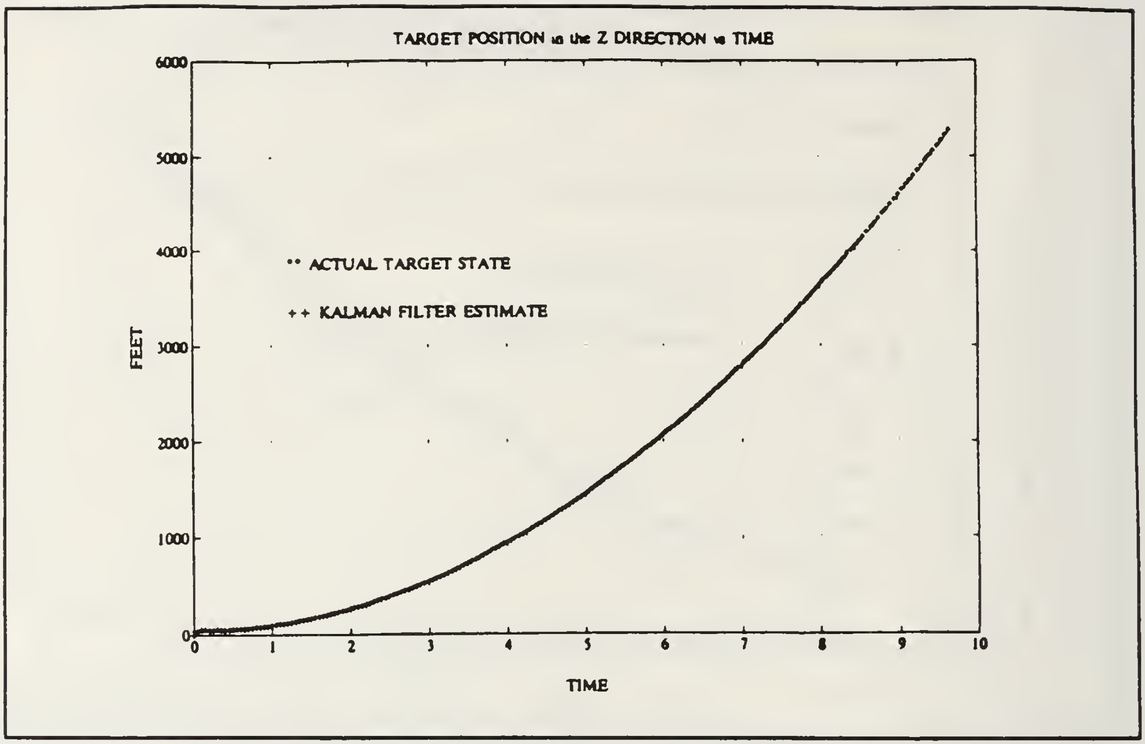

Figure 59. Target Position in the Z Direction

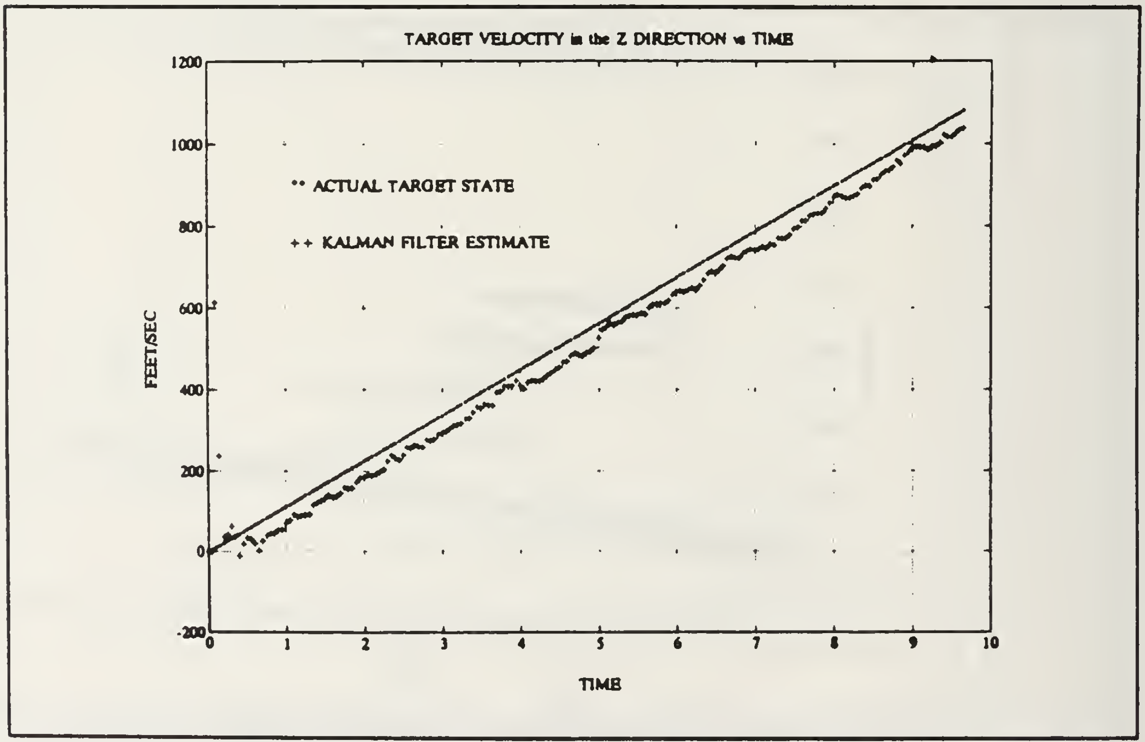

Figure 60. Target Velocity in the Z Direction

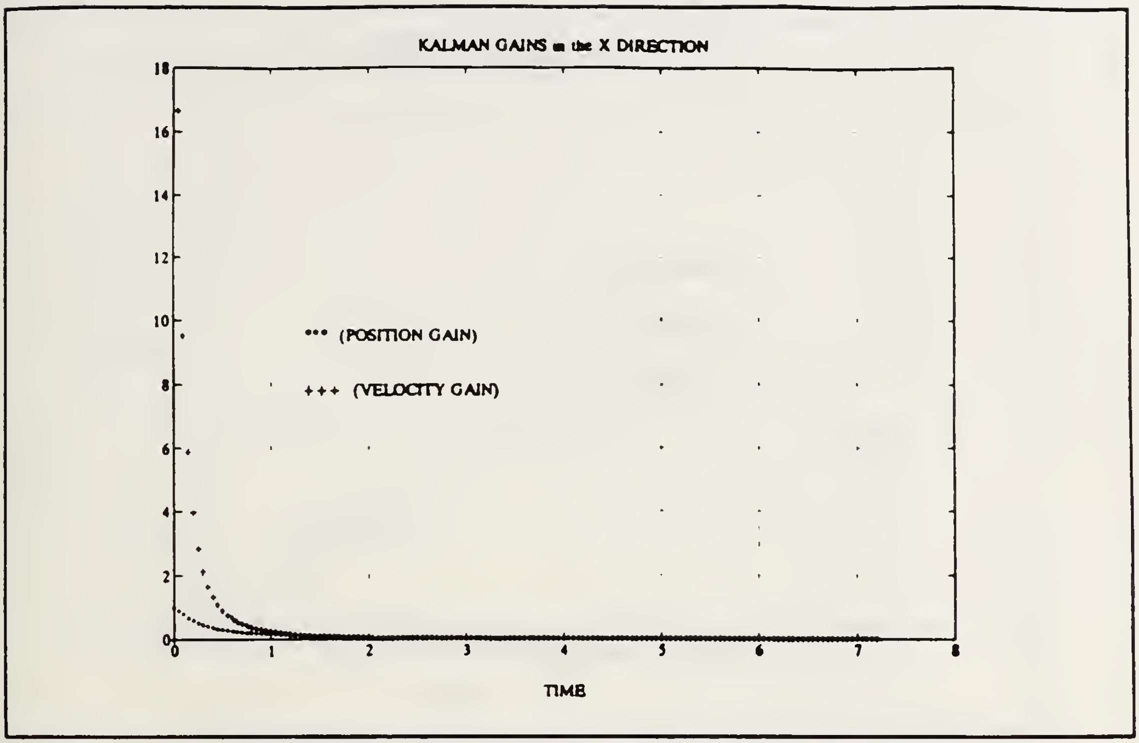

Figure 61. Kalman Gains in the X Direction

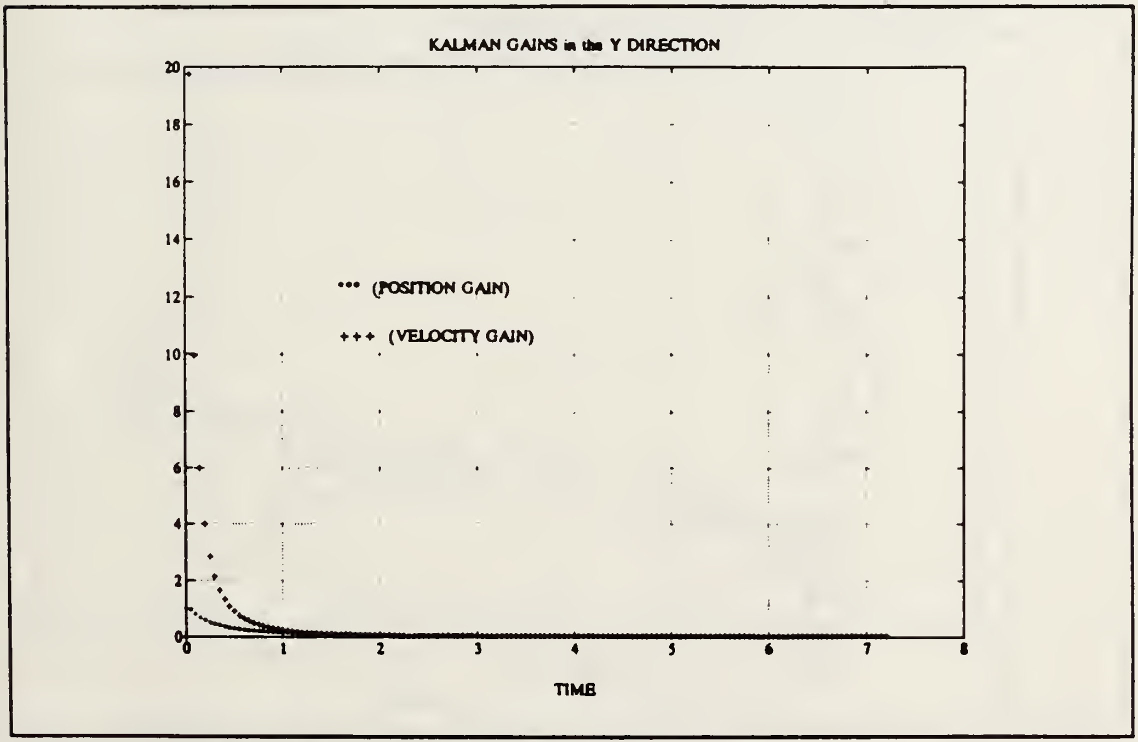

Figure 62. Kalman Gains in the Y Direction

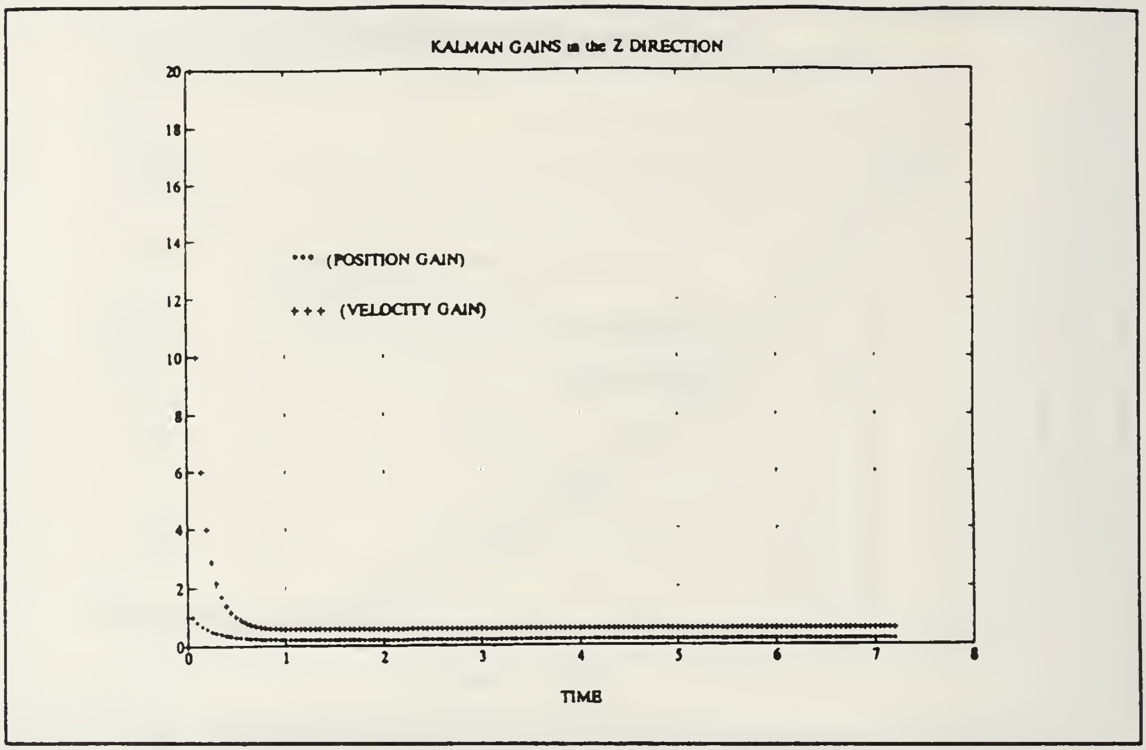

Figure 63. Kalman Gains in the Z Direction

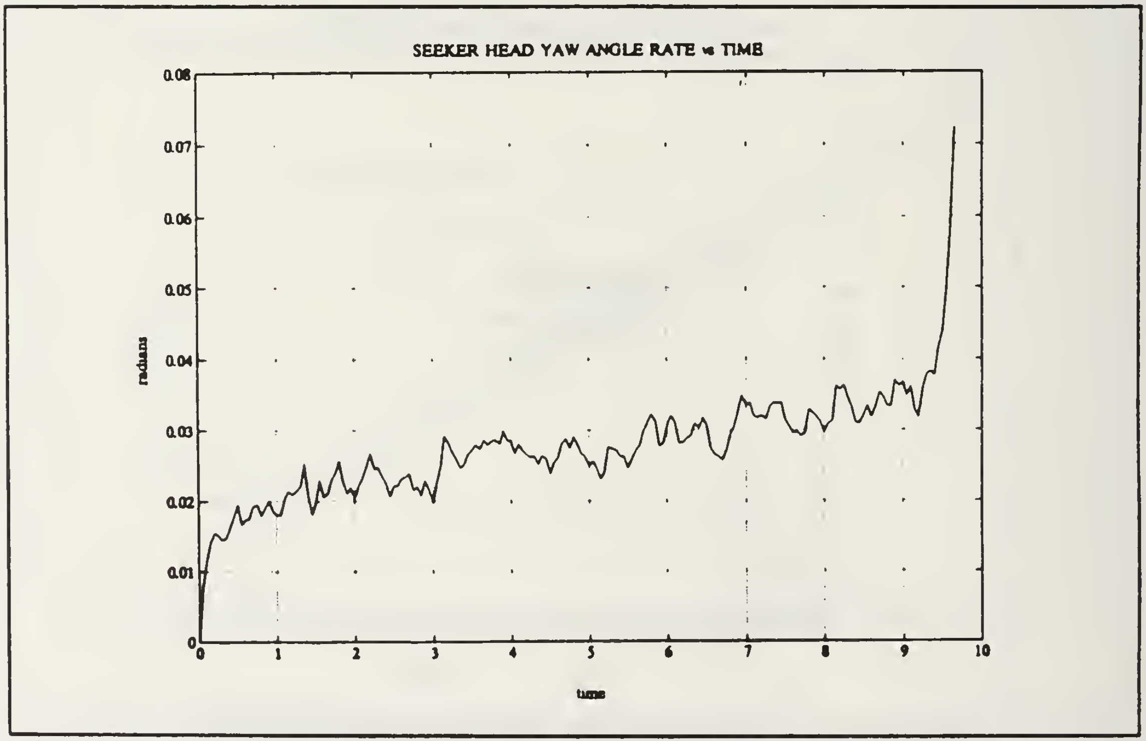

Figure 64. Seeker Head Angle Rate in the Yaw Plane

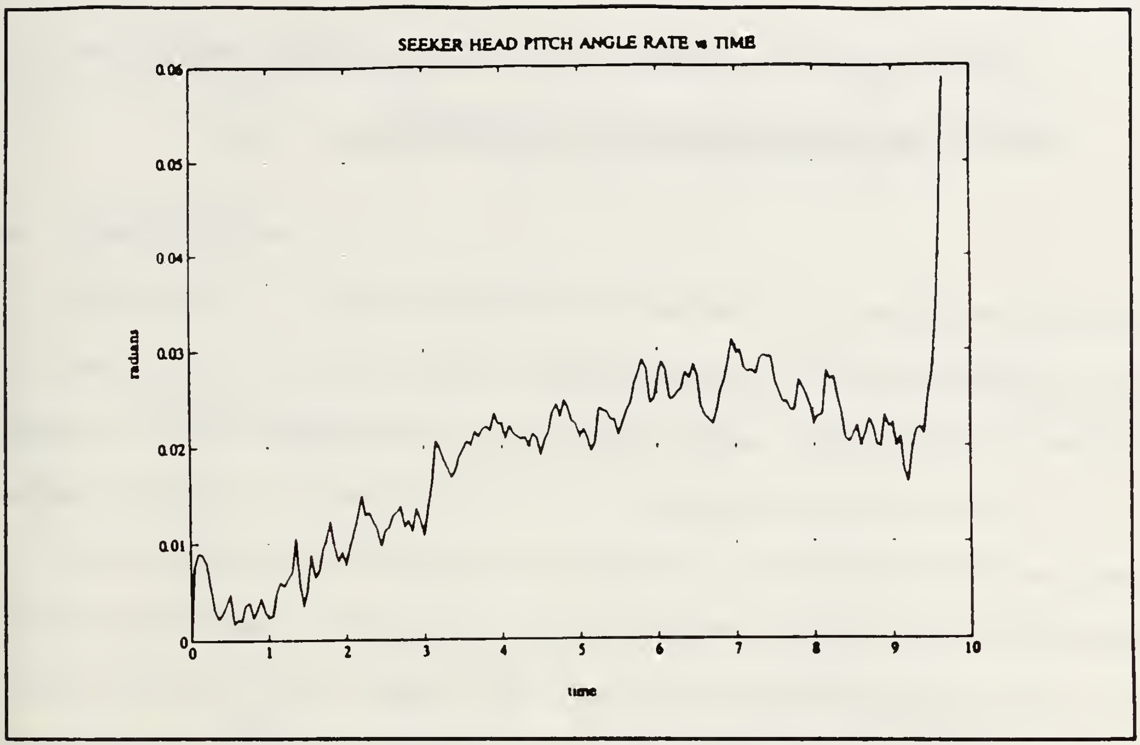

Figure 65. Seeker Head Angle Rate in the Pitch Plane

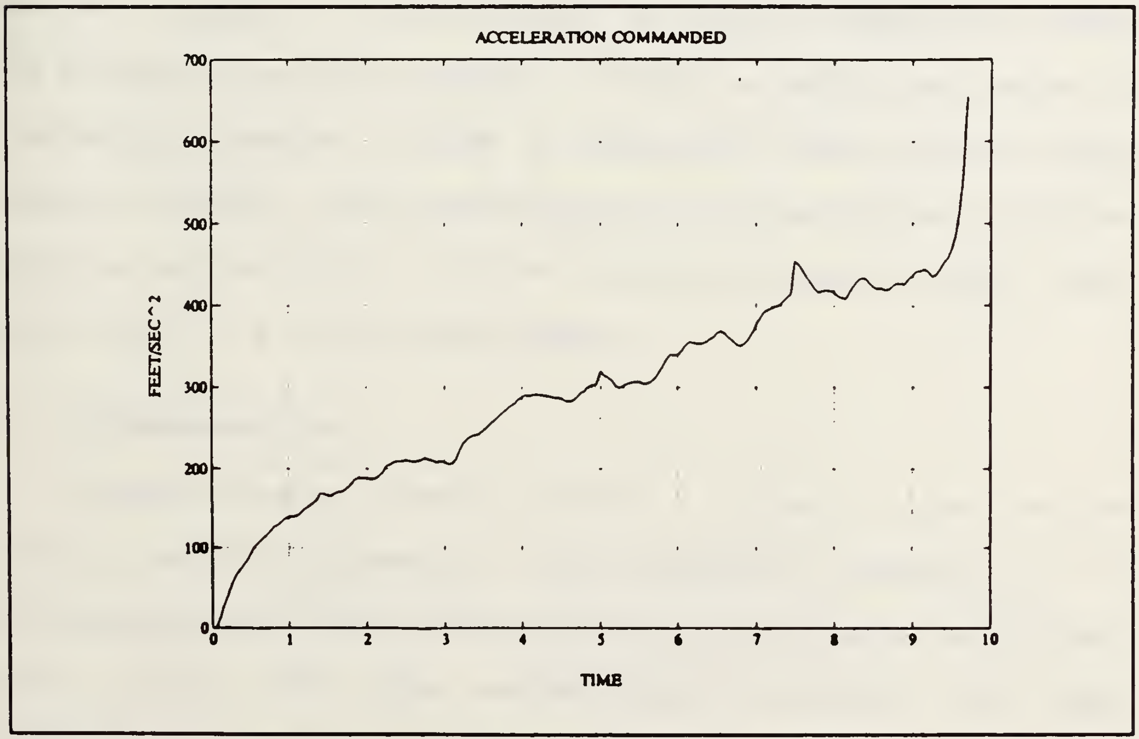

Figure 66. Commanded Acceleration

The computer program used to generate the plots in this chapter is included in the appendix.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The use of a three-dimensional model in simulation studies is beneficial in that the trajectories of the missile and target can be displayed in all three axes, therefore, a better understanding of the behavior of the systems involved.

The simulation studies in Chapter III demonstrated that the Proportional Navigation guidance law effectively achieves intercept of the target for both the maneuvering and non-manuvering target. Chapter IV demonstrated the need for the observer concept to be utilized in missile design to reduce the effects of noise. Chapter V showed an additional means of countering jamming or ECM by employing the ground target observer concept. This demonstrated that the uplinks in target position and velocity, for the constant velocity model, were beneficial in a noisy environment.

B. RECOMMENDATIONS

Further studies could include a Kalman Filter in the missile design instead of the Luenberger Observer. It is recommended that future studies include the use of a two part Kalman Filter with the ground target observer. One that estimates the states and the other being a maneuver detector.

The results could be compared with the present filter along with ECM effects to determine the effectiveness of the two.

APPENDIX

A. MISSILE-TARGET ENGAGEMENT PROGRAM

```
%
%      This program simulates a 3 dimensional target/missile
%       engagement using basic proportional navigation with
%       an asymptotic observer.

%       DEFINE STATES
%
%       Missile
%
%       ms=[xm    -  missile position in x direction
%           xdm   -  missile velocity in x direction
%           ym    -  missile position in y direction
%           ydm   -  missile velocity in y direction
%           zm    -  missile position in z direction
%           zdm   -  missile velocity in z direction]
%
Am=[0 1 0 0 0 0
    0 0 0 0 0 0
    0 0 0 1 0 0
    0 0 0 0 0 0
    0 0 0 0 0 1
    0 0 0 0 0 0];

Bm=[0 0 0
    1 0 0
    0 0 0
    0 1 0
    0 0 0
    0 0 1];
%
%       Seeker Head
%
%       betap=[beta pitch  - seeker head pitch angle
%              betad pitch - seeker head pitch angle rate
%       betay=[beta yaw    - seeker head yaw angle
%              betad yaw   - seeker head yaw angle rate]
%
%
Asp=[0 1;-100 -20.0];
Asy=[0 1;-100 -20.0];
Bsp=[0;100];
```

```
Bsy=[0;100];
%
%       Autopilot
%
%       gammamp=[gammam pitch - pitch body angle
%                gammadm pitch - pitch body angle rate   ]
%       gammamy=[gammam yaw - yaw body angle
%                gammadm yaw - yaw body angle rate]
%
 Ap=[-3 0;0 -3];

 Bp=[3 0;0 3];

%
%       Target
%
%       ts=[xt   -  target position in x direction
%           xdt  -  target velocity in x direction
%           yt   -  target position in y direction
%           ydt  -  target velocity in y direction
%           zt   -  target position in z direction
%           zdt  -  target velocity in z direction]
%
At=[0 1 0 0 0 0
    0 0 0 0 0 0
    0 0 0 1 0 0
    0 0 0 0 0 0
    0 0 0 0 0 1
    0 0 0 0 0 0];

Bt=[0 0 0
    1 0 0
    0 0 0
    0 1 0
    0 0 0
    0 0 1];
%
%       DISCRETE REPRESENTATION
%
dt=.05;

[phisp,delsp]= c2d(Asp,Bsp,dt);
[phisy,delsy]= c2d(Asy,Bsy,dt);
[phim,delm]= c2d(Am,Bm,dt);
[phia,dela]= c2d(Ap,Bp,dt);
[phit,delt]= c2d(At,Bt,dt);

tfinal= 20.0;
kmax=tfinal/dt +1;

%       INITIAL VALUES
```

```
%
%       navigation ratio
%
NR=[4.0 0;0 4.0];


%       initial seeker head angles and angle rates
%
beta_pitch0 =0.0;
betad_pitch0=0.0;
beta_yaw0   =0.0;
betad_yaw0  =0.0;

betap(:,1)=[beta_pitch0
           betad_pitch0 ];
betay(:,1)=[beta_yaw0
           betad_yaw0  ];

%       initial observer angles and angle rates

betae_pitch0=0.0;
betade_pitch0=0.0;
betae_yaw0=0.0;
betaed_yaw0=0.0;
%
%
betaep(:,1)=[betae_pitch0
             betade_pitch0];
betaey(:,1)=[betae_yaw0
             betaed_yaw0];
%
%
%       initial missile body angle rates
%
gammadm_pitch0=0.0;
gammadm_yaw0  =0.0;

gammadm(:,1)=[gammadm_pitch0
              gammadm_yaw0  ];
%
%       initial states  for missile flight profile
%
xm0 =000.0;
xdm0=2000.00;
ym0 =000.0;
ydm0=00;
zm0 =00;
zdm0=00;
ms(:,1)=[xm0
         xdm0
         ym0
         ydm0
         zm0
         zdm0];
%
%       initial  states for target flight profile
%
xt0 =20000;
xdt0=0000;
yt0 =050.0;
ydt0=0300.0;
zt0 =0051.0;
zdt0=000.0;

ts(:,1)=[xt0
         xdt0
         yt0
         ydt0
         zt0
         zdt0];
%
%       initial range information
%
rx0=(xt0-xm0);
rx(1)=rx0;
ry0=(yt0-ym0);
ry(1)=ry0;
rz0=(zt0-zm0);
rz(1)=rz0;
rm0=sqrt(xm0^2 + ym0^2 + zm0^2);
rm(1)=rm0;
rt0=sqrt(xt0^2 + yt0^2 + zt0^2);
rt(1)=rt0;
r0 =sqrt((xt0-xm0)^2 + (yt0-ym0)^2 + (zt0-zm0)^2);
r(1)=r0;


dist=[0.0013;0.0513];
rand('normal');
%
%       initial time
%
time(1)=0.0;
%
%       SIMULATION
%
for (i=1:kmax-1)

%      missile and target flight path angles

    gammam_pitch(i)=atan2(ms(6,i),sqrt(ms(2,i)^2+ms(4,i)^2));
    gammam_yaw(i)=atan2(ms(4,i),ms(2,i));
    gammat_pitch(i)=atan2(ts(6,i),sqrt(ts(2,i)^2+ts(4,i)^2));
    gammat_yaw(i)=atan2(ts(4,i),ts(2,i));


%      target velocity (magnitude)

     vt (i)=sqrt(ts(2,i)^2+ts(4,i)^2+ts(6,i)^2);
     vm (i)=sqrt(ms(2,i)^2+ms(4,i)^2+ms(6,i)^2);
%
%       line-of-sight angles
%

sigma_pitch(i)=atan2((ts(5,i)-ms(5,i)),sqrt((ts(1,i)-ms(1,i)
)^2...
                     +(ts(3,i)-ms(3,i))^2));


sigma_yaw(i)=atan2((ts(3,i)-ms(3,i)),((ts(1,i)-ms(1,i))));

      sigma(:,i)=[sigma_pitch(i)
                  sigma_yaw(i)  ];


%       update seeker head
%
wk(i)=rand*.9;


betap(:,i+1)=phisp*betap(:,i)   +
delsp*sigma_pitch(i)+dist*wk(i);

betay(:,i+1)=phisy*betay(:,i) + delsy*sigma_yaw(i)+dist*wk(i);

%
%       set up observer estimates
%
        y0(1)=0.0;
        y1(1)=0.0;
        %
        %
        gain=[-0.0031;0.0299];
        c=[0 1];
        f=phisp-gain*c;
        y0(i+1)=c*betap(:,i);
        betaep(:,i+1)=f*betaep(:,i)    +    gain*y0(i)    +
delsp*sigma_pitch(i);

        y1(i+1)=c*betay(:,i);
        f1=phisy-gain*c;
        betaey(:,i+1)=f1*betaey(:,i)    +    gain*y1(i)    +
delsy*sigma_yaw(i);
```

```
%
%
%        seeker head angle rate vector
%
        betad(:,i)=[betaep(2,i)
                    betaey(2,i)];

%
%       update autopilot

%       acceleration commanded = vm(i) * gammadm(i)

%       gammadm(i) = Nav constant * line of sight rate

        gammadm(:,i+1)=phia*gammadm(:,i) + dela*NR*betad(:,i);

        vm_pitch(i)=vm(i)*cos(gammam_yaw(i)-sigma_yaw(i));
        vm_yaw(i)=vm(i)*cos(gammam_pitch(i));


%       limit commanded acceleration to approximately 20 g's
%
              if (vm_yaw(i)*gammadm(2,i))<=640.0
                   amc_y(i)=(vm_yaw(i)*gammadm(2,i));
              else
                   amc_y(i)=640.0;
              end
%
%
              if(vm_pitch(i)*gammadm(1,i))<=640.0
                    amc_p(i)=(vm_pitch(i)*gammadm(1,i));
              else
                    amc_p(i)=640.0;
              end
%
%        magnitude of commanded acceleration

          acom(i)=sqrt(amc_y(i)^2 + amc_p(i)^2);

%        missile acceleration vector components


        xddm(i)=(gammadm(1,i)*ms(6,i)*cos(gammam_yaw(i)));
        yddm(i)=amc_y(i)*cos(gammam_yaw(i));
        zddm(i)=amc_p(i)*cos(gammam_pitch(i));

%        total missile acceleration magnitude
%
        am(i)=sqrt(xddm(i)^2 + zddm(i)^2 + yddm(i)^2 );
%
%        missile input vector
```

```
%
        um=[xddm(i)
            yddm(i)
            zddm(i)];
%
%       update missile
%
           ms(:,i+1)=phim*ms(:,i) + delm*um;

%
%       set target acceleration components
%
       if (r(i)<=21000.0)
           xddt(i)=-5.0*32.2*sin(sigma_yaw(i));
           yddt(i)= 3.5*32.2*cos(sigma_yaw(i));
           zddt(i)= 3.5*32.2*cos(sigma_pitch(i));
       else
           xddt(i)=0.0;
           yddt(i)=0.0;
           zddt(i)=0.0;
       end

%
%       target acceleration magnitude
%
        at(i)=sqrt(xddt(i)^2 + yddt(i)^2 + zddt(i)^2);
%
%        target input vector
%
        ut=[xddt(i)
            yddt(i)
            zddt(i)];
%
%       update target
%
            ts(:,i+1)=phit*ts(:,i) + delt*ut;

%
%
%        missile and target trajectory data
%
        missile(i,:)=[ms(1,i) ms(3,i) ms(5,i)];
        target(i,:) =[ts(1,i) ts(3,i) ts(5,i)];
%

%  update  range information
%

          r(i+1)=sqrt((ts(1,i+1)-ms(1,i+1))^2   +
(ts(3,i+1)-ms(3,i+1))^2...
                  + (ts(5,i+1)-ms(5,i+1))^2);
        rm(i+1)=sqrt(ms(1,i+1)^2 + ms(3,i+1)^2 + ms(5,i+1)^2);
        rt(i+1)=sqrt(ts(1,i+1)^2 + ts(3,i+1)^2 + ts(5,i+1)^2);
```

```
%       update time vector
%
        time(i+1)=time(i) + dt;
%
%        check to see if engagement is at closest point of
approach
%
        if(r(i)<r(i+1)),break,end
end;
%
%       PLOT RESULTS
%
%       range data
%
plot(time,r),grid,xlabel('TIME'),ylabel('FEET')
title('RANGE VS TIME')
text( 7,20000,['rng=',num2str(r(i)),' feet']);
meta msltgt,pause,clg

subplot(211),plot(time,rm),grid,xlabel('TIME'),ylabel('FEET')
title('MISSILE RANGE VS TIME');
subplot(212),plot(time,rt),grid,xlabel('TIME'),ylabel('FEET')
title('TARGET RANGE VS TIME');
meta jerry1,pause,clg
%
%
plot(ms(1,:),ms(3,:),ts(1,:),ts(3,:)),grid
title('MISSILE-TARGET TRAJECTORY IN THE X-Y PLANE')
xlabel('X AXIS IN FEET'),ylabel('Y AXIS IN FEET')
gtext('MISSILE')
gtext('TARGET')
meta msltgt,pause,clg

plot(ms(1,:),ms(5,:),ts(1,:),ts(5,:)),grid
title('MISSILE-TARGET TRAJECTORY IN THE X-Z PLANE')
xlabel('X AXIS IN FEET'),ylabel('Z AXIS IN FEET')
gtext('MISSILE')
gtext('TARGET')
meta msltgt,pause,clg

%       missile and target velocity
%
plot(time(1:i),vm),grid,xlabel('TIME'),ylabel('FEET/SEC')
title('MISSILE VELOCITY VS TIME');
meta jerry1,pause,clg

plot(time(1:i),vt),grid,xlabel('TIME'),ylabel('FEET/SEC')
title('TARGET VELOCITY VS TIME');
meta jerry1,pause,clg

%      missile and target acceleration
```

```
plot(time(1:i),am),grid,xlabel('time'),ylabel('feet/sec^2')
title('Missile acceleration vs Time')
meta jerry1,pause,clg
plot(time(1:i),at),grid,xlabel('time'),ylabel('feet/sec^2')
title('Target acceleration vs Time')
meta jerry1,pause,clg

%       seeker head angle rates

plot(time(1:i-1),betay(2,(1:i-1))),grid,xlabel('time'),ylabe
l('radians')
title('Seeker head Yaw angle rate vs Time')
meta jerry1,pause,clg
plot(time(1:i-1),betap(2,(1:i-1))),grid,xlabel('time'),ylabe
l('radians')
title('Seeker head Pitch angle rate vs Time')
meta jerry1,pause,clg

%       commanded acceleration

plot(time(1:i),acom),grid,title('acceleration commanded')
meta jerry1,pause,clg

END
```

B. MISSILE-TARGET PROGRAM WITH TARGET OBSERVER

```
clear
!erase msltgt.met
!erase jerry1.met
clg
%
%
%
%       This program simulates a 3 dimensional target/missile
%        engagement using basic proportional navigation with
%        an Luenberger observer and a ground target observer.

%        DEFINE STATES
%
%        Missile
%
%        ms=[xm   -  missile position in x direction
%            xdm  -  missile velocity in x direction
%            ym   -  missile position in y direction
%            ydm  -  missile velocity in y direction
%            zm   -  missile position in z direction
%            zdm  -  missile velocity in z direction]
%
Am=[0 1 0 0 0 0
    0 0 0 0 0 0
    0 0 0 1 0 0
    0 0 0 0 0 0
    0 0 0 0 0 1
    0 0 0 0 0 0];

Bm=[0 0 0
    1 0 0
    0 0 0
    0 1 0
    0 0 0
    0 0 1];
%
%        Seeker Head
%
%        betap=[beta pitch  - seeker head pitch angle
%              betad pitch - seeker head pitch angle rate
%        betay=[beta yaw    - seeker head yaw angle
%              betad yaw   - seeker head yaw angle rate]
%
%
Asp=[0 1;-100 -20.0];
Asy=[0 1;-100 -20.0];
Bsp=[0;100];
Bsy=[0;100];
```

```
%
%       Autopilot
%
%       gammamp=[gammam pitch - pitch body angle
%                gammadm pitch - pitch body angle rate  ]
%       gammamy=[gammam yaw - yaw body angle
%                gammadm yaw - yaw body angle rate]
%
 Ap=[-3 0;0 -3];


 Bp=[3 0;0 3];

%       Target
%
%       ts=[xt   -  target position in x direction
%           xdt  -  target velocity in x direction
%           yt   -  target position in y direction
%           ydt  -  target velocity in y direction
%           zt   -  target position in z direction
%           zdt  -  target velocity in z direction]
%
At=[0 1 0 0 0 0
    0 0 0 0 0 0
    0 0 0 1 0 0
    0 0 0 0 0 0
    0 0 0 0 0 1
    0 0 0 0 0 0];

Bt=[0 0 0
    1 0 0
    0 0 0
    0 1 0
    0 0 0
    0 0 1];
%
%       DISCRETE REPRESENTATION
%
dt=.05;

[phisp,delsp]= c2d(Asp,Bsp,dt);
[phisy,delsy]= c2d(Asy,Bsy,dt);
[phim,delm]= c2d(Am,Bm,dt);
[phia,dela]= c2d(Ap,Bp,dt);
[phit,delt]= c2d(At,Bt,dt);

tfinal= 20.0;
kmax=tfinal/dt +1;

%       INITIAL VALUES
%
```

```
%       navigation ratio
%
NR=[4.0 0;0 4.0];


%       initial seeker head angles and angle rates
%
beta_pitch0 =0.0;
betad_pitch0=0.0;
beta_yaw0   =0.0;
betad_yaw0  =0.0;

betap(:,1)=[beta_pitch0
           betad_pitch0 ];
betay(:,1)=[beta_yaw0
           betad_yaw0  ];

%       initial line-of-sight angular rates

sigmad_pitch0=0.0;
sigmad_yaw0  =0.0;

sigmad(:,1)=[sigmad_pitch0
             sigmad_yaw0 ];

%       initial observer angles and angle rates

betae_pitch0=0.0;
betade_pitch0=0.0;
betae_yaw0=0.0;
betaed_yaw0=0.0;
%
%
betaep(:,1)=[betae_pitch0
             betade_pitch0];
betaey(:,1)=[betae_yaw0
             betaed_yaw0];
%
%
%       initial missile body angle rates
%
gammadm_pitch0=0.0;
gammadm_yaw0  =0.0;

gammadm(:,1)=[gammadm_pitch0
              gammadm_yaw0  ];
%
%       initial missile states
%
xm0 =000.0;
xdm0=2000.00;
```

```
ym0 =000.0;
ydm0=00;
zm0 =00;
zdm0=00;
ms(:,1)=[xm0
         xdm0
         ym0
         ydm0
         zm0
         zdm0];
%
%       initial target states
%
xt0 =20000;
xdt0=0000;
yt0 =050.0;
ydt0=1000.0;
zt0 =050.0;
zdt0=000.0;

ts(:,1)=[xt0
         xdt0
         yt0
         ydt0
         zt0
         zdt0];
%
%       initial range information
%
rx0=(xt0-xm0);
rx(1)=rx0;
ry0=(yt0-ym0);
ry(1)=ry0;
rz0=(zt0-zm0);
rz(1)=rz0;
rm0=sqrt(xm0^2 + ym0^2 + zm0^2);
rm(1)=rm0;
rt0=sqrt(xt0^2 + yt0^2 + zt0^2);
rt(1)=rt0;
r0 =sqrt((xt0-xm0)^2 + (yt0-ym0)^2 + (zt0-zm0)^2);
r(1)=r0;
%       Initial Target Tracker values

   y(:,1)=[0;0;0];
   xhat1    =[0;0;0;0;0;0];

   pkkm1=[10^9 0 0 0 0 0
          0 10^9 0 0 0 0
          0 0 10^9 0 0 0
          0 0 0 10^9 0 0
          0 0 0 0 10^9 0
          0 0 0 0 0 10^9];

   rand('normal');

   h=[1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

   aT=[0 1 0 0 0 0
       0 0 0 0 0 0
       0 0 0 1 0 0
       0 0 0 0 0 0
       0 0 0 0 0 1
       0 0 0 0 0 0];

   bT=[0 0 0
       1 0 0
       0 0 0
       0 1 0
       0 0 0
       0 0 1];
%
   dist=[0.0013;0.0513];


% Continuous to Discrete
%
   [phi,del]=c2d(aT,bT,dt);
%
%       initial time
%
time(1)=0.0;
%
%       SIMULATION
j=0;
for (i=1:kmax-1)
j=j+1;
%      missile and target flight path angles

    gammam_pitch(i)=atan2(ms(6,i),sqrt(ms(2,i)^2+ms(4,i)^2));
    gammam_yaw(i)=atan2(ms(4,i),ms(2,i));
    gammat_pitch(i)=atan2(ts(6,i),sqrt(ts(2,i)^2+ts(4,i)^2));
    gammat_yaw(i)=atan2(ts(4,i),ts(2,i));


%      target velocity (magnitude)

     vt (i)=sqrt(ts(2,i)^2+ts(4,i)^2+ts(6,i)^2);
     vm (i)=sqrt(ms(2,i)^2+ms(4,i)^2+ms(6,i)^2);
%
%       line-of-sight angles
```

```
%

sigma_pitch(i)=atan2((ts(5,i)-ms(5,i)),sqrt((ts(1,i)-ms(1,i)
)^2...
                              +(ts(3,i)-ms(3,i))^2));


sigma_yaw(i)=atan2((ts(3,i)-ms(3,i)),((ts(1,i)-ms(1,i))));

        sigma(:,i)=[sigma_pitch(i)
                     sigma_yaw(i)  ];


%        update seeker head

 wk(i)=rand*.9;
%
    betap(:,i+1)=phisp*betap(:,i)    +
delsp*sigma_pitch(i)+dist*wk(i);

    betay(:,i+1)=phisy*betay(:,i)    +
delsy*sigma_yaw(i)+dist*wk(i);

%
%        set up observer estimates
%
         y0(1)=0.0;
         y1(1)=0.0;
         %
         %
         gain=[-0.0031;0.0299];
         c=[0 1];
         f=phisp-gain*c;
         y0(i+1)=c*betap(:,i);
         betaep(:,i+1)=f*betaep(:,i)    +    gain*y0(i)    +
delsp*sigma_pitch(i);

         y1(i+1)=c*betay(:,i);
         f1=phisy-gain*c;
         betaey(:,i+1)=f1*betaey(:,i)    +    gain*y1(i)    +
delsy*sigma_yaw(i);
%
%
%
%         seeker head angle rate vector
%         with uplinks at one second intervals


          if (j==40)
              betad(:,i)=[(betaep(2,i)+sigmad(1,i))
                           (betaey(2,i)+sigmad(2,i))];
```

```
               j=0;
         else
           betad(:,i)=[betaep(2,i)
                       betaey(2,i)];
        end

%       update autopilot

%       acceleration commanded = vm(i) * gammadm(i)

%       gammadm(i) = Nav constant * line of sight rate

        gammadm(:,i+1)=phia*gammadm(:,i) + dela*NR*betad(:,i);

        vm_pitch(i)=vm(i)*cos(gammam_yaw(i)-sigma_yaw(i));
        vm_yaw(i)=vm(i)*cos(gammam_pitch(i));


%       limit commanded acceleration to approximately 20 g's
%
              if (vm_yaw(i)*gammadm(2,i))<=640.0
                   amc_y(i)=(vm_yaw(i)*gammadm(2,i));
              else
                   amc_y(i)=640.0;
              end
%
%
              if(vm_pitch(i)*gammadm(1,i))<=640.0
                    amc_p(i)=(vm_pitch(i)*gammadm(1,i));
              else
                    amc_p(i)=640.0;
              end
%
%        magnitude of commanded acceleration

          acom(i)=sqrt(amc_y(i)^2 + amc_p(i)^2);

%        missile acceleration vector components


        xddm(i)=(gammadm(1,i)*ms(6,i)*cos(gammam_yaw(i)));
        yddm(i)=amc_y(i)*cos(gammam_yaw(i));
        zddm(i)=amc_p(i)*cos(gammam_pitch(i));

%        total missile acceleration magnitude
%
        am(i)=sqrt(xddm(i)^2 + zddm(i)^2 + yddm(i)^2 );
%
%        missile input vector
%
```

```
        um=[xddm(i)
            yddm(i)
            zddm(i)];
%
%       update missile
%
           ms(:,i+1)=phim*ms(:,i) + delm*um;
%
%       set target acceleration components
%
       if (r(i)<=00000.0)
           xddt(i)= -5.0*32.2*sin(sigma_yaw(i));
           yddt(i)=  3.5*32.2*cos(sigma_yaw(i));
           zddt(i)=  3.5*32.2*cos(sigma_pitch(i));
       else
           xddt(i)=0.0;
           yddt(i)=0.0;
           zddt(i)=0.0;
       end

%       target acceleration magnitude
%
        at(i)=sqrt(xddt(i)^2 + yddt(i)^2 + zddt(i)^2);
%
%        target input vector
%
        ut=[xddt(i)
            yddt(i)
            zddt(i)];
%
%       update target
%
            ts(:,i+1)=phit*ts(:,i) + delt*ut;
%
%
%        missile and target trajectory data
%
        missile(i,:)=[ms(1,i) ms(3,i) ms(5,i)];
        target(i,:) =[ts(1,i) ts(3,i) ts(5,i)];
%
%       Target traker

%       Input initial conditions
%

x(:,i)=[ts(1,i);ts(2,i);ts(3,i);ts(4,i);ts(5,i);ts(6,i)];
%
%   Plant noise covariance
%
         Q1=[160^2 0 0;0 160^2 0;0 0 160^2];
```

```
%   Measurement noise covariance

         R=[500^2 0 0
            0 125^2 0
            0 0 12^2];

%  Input equations

         Q=[160^2;160^2;160^2];
         nu=[sqrt(500)^2;sqrt(125^2);sqrt(12^2)]*rand;
         w(:,i) =sqrt(Q)*rand;



%  Update the plant states
%
         x(:,i+1)=phi*x(:,i)  +del*w(:,i);
         y(:,i+1)=h*x(:,i+1)  +nu;
%
%  Compute Kalman gains
%
         G = pkkm1*h'*inv(h*pkkm1*h'+R);
%
%  Define gain matrix for plotting

         g(:,i) =[G(1:2,1);G(3:4,2);G(5:6,3)];
%
%  Measurement step

         xhat (:,i+1) =xhat1+(G *(y(:,i+1)-h*xhat1));
         pkk          =(eye(6)-G *h)*pkkm1;
%
%  Movement step

         xhat1=phi*xhat (:,i+1);
         pkkm1=phi*pkk*phi'+ del*Q1*del';

%  compute the range deviation

         xhatr  =[xhat(1,i);xhat(3,i);xhat(5,i)];
         yr     =y(:,i);
         xr(:,i)=   (yr - xhatr);

%  compute the velocity deviation

         xhatv  =[xhat(2,i);xhat(4,i);xhat(6,i)];
         yv     =[ts(2,i);ts(4,i);ts(6,i)];
         xv(:,i)=   (yv - xhatv);

%
```

```
%   Transformation to body axes

          a11=cos(gammam_pitch(i))*cos(gammam_yaw(i));
          a12=cos(gammam_pitch(i))*sin(gammam_yaw(i));
          a13=(-sin(gammam_pitch(i)));
          a21=(-sin(gammam_yaw(i)));
          a22=cos(gammam_yaw(i));
          a23=0.0;
          a31=sin(gammam_pitch(i))*cos(gammam_yaw(i));
          a32=sin(gammam_pitch(i))*sin(gammam_yaw(i));
          a33=cos(gammam_pitch(i));

%   velocity transformation

          vbx(i)=xv(1,i)*a11+xv(2,i)*a12+xv(3,i)*a13;
          vby(i)=xv(1,i)*a21+xv(2,i)*a22+xv(3,i)*a23;
          vbz(i)=xv(1,i)*a31+xv(2,i)*a32+xv(3,i)*a33;

%   range transformation

          rbx(i)=xr(1,i)*a11+xr(2,i)*a12+xr(3,i)*a13;
          rby(i)=xr(1,i)*a21+xr(2,i)*a22+xr(3,i)*a23;
          rbz(i)=xr(1,i)*a31+xr(2,i)*a32+xr(3,i)*a33;

%   line-of-sight transformation

          s11=cos(sigma_pitch(i))*cos(sigma_yaw(i));
          s12=cos(sigma_pitch(i))*sin(sigma_yaw(i));
          s13=(-sin(sigma_pitch(i)));
          s21=(-sin(sigma_yaw(i)));
          s22=cos(sigma_yaw(i));
          s23=0.0;
          s31=sin(sigma_pitch(i))*cos(sigma_yaw(i));
          s32=sin(sigma_pitch(i))*sin(sigma_yaw(i));
          s33=cos(sigma_pitch(i));

%   velocity transformation

          vsx(i)=vbx(i)*s11+vby(i)*s12+vbz(i)*s13;
          vsy(i)=vbx(i)*s21+vby(i)*s22+vbz(i)*s23;
          vsz(i)=vbx(i)*s31+vby(i)*s32+vbz(i)*s33;

          rsx(i)=rbx(i)*s11+rby(i)*s12+rbz(i)*s13;
          rsy(i)=rbx(i)*s21+rby(i)*s22+rbz(i)*s23;
          rsz(i)=rbx(i)*s31+rby(i)*s32+rbz(i)*s33;

%   compute the magnitude of the range deviation

          rd(i)=sqrt(rsx(i)^2+rsy(i)^2+rsz(i)^2);

%   generate the angular rate deviations
```

```
         sigmad_pitch(i+1)= vsy(i)/(r(i)+rd(i));
         sigmad_yaw(i+1)= vsz(i)/(r(i)+rd(i));

         sigmad(:,i+1)=[sigmad_pitch(i)
                        sigmad_yaw(i)];


%  update  range information
%

          r(i+1)=sqrt((ts(1,i+1)-ms(1,i+1))^2   +
(ts(3,i+1)-ms(3,i+1))^2...
                  + (ts(5,i+1)-ms(5,i+1))^2);
        rm(i+1)=sqrt(ms(1,i+1)^2 + ms(3,i+1)^2 + ms(5,i+1)^2);
        rt(i+1)=sqrt(ts(1,i+1)^2 + ts(3,i+1)^2 + ts(5,i+1)^2);

%       update time vector
%
        time(i+1)=time(i) + dt;
%
%        check to see if engagement is at closest point of
approach
%
        if(r(i)<r(i+1)),break,end
end;
%
%       PLOT RESULTS
%
%       range data
%
plot(time,r),grid,xlabel('TIME'),ylabel('FEET')
title('RANGE VS TIME')
text( 7,20000,['rng=',num2str(r(i)),' feet']);
meta msltgt,pause,clg

subplot(211),plot(time,rm),grid,xlabel('TIME'),ylabel('FEET')
title('MISSILE RANGE VS TIME');
subplot(212),plot(time,rt),grid,xlabel('TIME'),ylabel('FEET')
title('TARGET RANGE VS TIME');
meta msltgt,pause,clg
%
%
%       missile and target velocity
%
%plot(time(1:i),vm),grid,xlabel('TIME'),ylabel('FEET/SEC')
%title('MISSILE VELOCITY VS TIME');
%meta msltgt,pause,clg

%plot(time(1:i),vt),grid,xlabel('TIME'),ylabel('FEET/SEC')
%title('TARGET VELOCITY VS TIME');
%meta msltgt,pause,clg
```

```
%      missile and target acceleration

plot(time(1:i),am),grid,xlabel(’TIME’),ylabel(’FEET/SEC^2’)
title(’MISSILE ACCELERATION vs TIME’)
meta msltgt,pause,clg
plot(time(1:i),at),grid,xlabel(’time’),ylabel(’feet/sec^2’)
title(’TARGET ACCELERATION vs TIME’)
meta msltgt,pause,clg

%      seeker head angle rates

plot(time(1:i-1),betaey(2,(1:i-1))),grid,xlabel(’time’),ylab
el(’radians’)
title(’SEEKER HEAD YAW ANGLE RATE vs TIME’)
meta msltgt,pause,clg
plot(time(1:i-1),betaep(2,(1:i-1))),grid,xlabel(’time’),ylab
el(’radians’)
title(’SEEKER HEAD PITCH ANGLE RATE vs TIME’)
meta msltgt,pause,clg

%      commanded acceleration

plot(time(1:i),acom),grid,title(’ACCELERATION COMMANDED’)
xlabel(’TIME’),ylabel(’FEET/SEC^2’)
meta msltgt,pause,clg


%      Kalman estimates

plot(time(1:i-1),x(1,(1:i-1)),’*’,time(1:i-1),xhat(1,(1:i-1)
),’+’)
xlabel(’TIME’),ylabel(’FEET’),grid
title(’TARGET POSITION in the X DIRECTION vs TIME’)
gtext(’** ACTUAL TARGET STATE  ’)
gtext(’++ KALMAN FILTER ESTIMATE’)
meta msltgt,pause,clg

plot(time(1:i-1),x(2,(1:i-1)),’*’,time(1:i-1),xhat(2,(1:i-1)
),’+’)
xlabel(’TIME’),ylabel(’FEET/SEC’),grid
title(’TARGET VELOCITY in the X DIRECTION vs TIME’)
gtext(’** ACTUAL TARGET STATE   ’)
gtext(’++ KALMAN FILTER ESTIMATE’)
meta msltgt,pause,clg

plot(time(1:i-50),g(1,(1:i-50)),’*’,time(1:i-50),g(2,(1:i-50
)),’+’)
title(’KALMAN GAINS in the X DIRECTION’),grid,xlabel(’TIME’),
gtext(’***  (POSITION GAIN)’)
gtext(’+++  (VELOCITY GAIN)’)
meta msltgt,pause,clg
```

```
%      Kalman estimates

plot(time(1:i-1),x(3,(1:i-1)),'*',time(1:i-1),xhat(3,(1:i-1)
),'+')
xlabel('TIME'),ylabel('FEET'),grid
title('TARGET POSITION in the Y DIRECTION vs TIME')
gtext('** ACTUAL TARGET STATE  ')
gtext('++ KALMAN FILTER ESTIMATE')
meta msltgt,pause,clg

plot(time(1:i-1),x(4,(1:i-1)),'*',time(1:i-1),xhat(4,(1:i-1)
),'+')
xlabel('TIME'),ylabel('FEET/SEC'),grid
title('TARGET VELOCITY in the Y DIRECTION vs TIME')
gtext('** ACTUAL TARGET STATE   ')
gtext('++ KALMAN FILTER ESTIMATE')
meta msltgt,pause,clg

plot(time(1:i-50),g(3,(1:i-50)),'*',time(1:i-50),g(4,(1:i-50
)),'+')
title('KALMAN GAINS in the Y DIRECTION'),grid,xlabel('TIME'),
gtext('***  (POSITION GAIN)')
gtext('+++  (VELOCITY GAIN)')
meta msltgt,pause,clg

plot(time(1:i-1),x(5,(1:i-1)),'*',time(1:i-1),xhat(5,(1:i-1)
),'+')
xlabel('TIME'),ylabel('FEET'),grid
title('TARGET POSITION in the Z DIRECTION vs TIME')
gtext('** ACTUAL TARGET STATE  ')
gtext('++ KALMAN FILTER ESTIMATE')
meta msltgt,pause,clg

plot(time(1:i-1),x(6,(1:i-1)),'*',time(1:i-1),xhat(6,(1:i-1)
),'+')
xlabel('TIME'),ylabel('FEET/SEC'),grid
title('TARGET VELOCITY in the Z DIRECTION vs TIME')
gtext('** ACTUAL TARGET STATE   ')
gtext('++ KALMAN FILTER ESTIMATE')
meta msltgt,pause,clg

plot(time(1:i-50),g(5,(1:i-50)),'*',time(1:i-50),g(6,(1:i-50
)),'+')
title('KALMAN GAINS in the Z DIRECTION'),grid,xlabel('TIME'),
gtext('***  (POSITION GAIN)')
gtext('+++  (VELOCITY GAIN)')
meta msltgt1,pause,clg
end
```

LIST OF REFERENCES

1. Garnell, P., *Guided Weapon Systems*, Pergamon Press, 1980.

2. Titus, H.A., *Missile Guidance*, Naval Postgraduate School, 1990, (unpublished notes).

3. Nielson, J.N., *Tactical Missile Dynamics*, American Institute of Aeronautics and Astronautics, vol 104, pp. 5-40, 1986.

4. Burl, J., *Luenberger Observer Design*, Naval Postgraduate School, 1990, (unpublished notes).

5. Friedland, B., *Control System Design*, McGraw-Hill, Inc., 1986.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 52 2
Naval Postgraduate School
Monterey, California 93943-5002

3. Chairman, Code EC 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

4. Professor H. A. Titus, Code EC/TS 2
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

5. Professor J. Burl, Code EC/BL 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

6. Dr. Jill Burt 2
Missile and Space Intelligence Center
Redstone Arsenal, Alabama 35898-5500

7. Lieutenant Gerald L. Painter 2
1821 Mars Drive
Bossier City, Louisiana 71112

Thesis
P119415 Painter
c.1
Computer simulated missile-target engagement with a Luenberger Observer and a ground observer.