2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A TARGET/MISSILE ENGAGEMENT SCENARIO USING CLASSICAL PROPORTIONAL NAVIGATION

by

Francis C. Lukenbill

December 1990

Thesis Advisor Harold A. Titus

Approved for public release; distribution is unlimited.

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report
2b Declassification/Downgrading Schedule	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 Number | Project No | Task No | Work Unit Accession No

11 Title (*Include Security Classification*) A Target/Missile Engagement Scenario Using Classical Proportional Navigation

12 Personal Author(s) Francis C. Lukenbill

13a Type of Report	13b Time Covered	14 Date of Report (*year, month, day*)	15 Page Count
Master's Thesis	From To	December 1990	121

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 U.S. Government.

17 Cosati Codes			18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Field	Group	Subgroup	proportional navigation, miss distance, adjoint

19 Abstract (*continue on reverse if necessary and identify by block number*

This thesis deals with a simulation of a missile versus target engagement scenario. After deriving simplified transfer functions for the missile seeker head, missile autopilot, missile dynamics, and target dynamics, a three dimensional simulation is developed using classical proportional navigation. The scenario is simulated using state variable design. A forward time solution of the two dimensional problem is developed which is converted to an adjoint model. The adjoint model is used to determine the optimal time to initiate simplified and tactical evasion maneuvers in order to maximize the final miss distance.

20 Distribution/Availability of Abstract ☒ unclassified/unlimited □ same as report □ 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

83 APR edition may be used until exhausted

All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

A Target/Missile Engagement Scenario Using Classical Proportional Navigation

by

Francis C. Lukenbill
Lieutenant, United States Navy
B.S., United States Naval Academy, 1983

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1990

Author: Francis C Lukenbill
Francis C. Lukenbill

Approved by: H A Titus
Harold A. Titus, Thesis Advisor

James R Powell
James R. Powell, Second Reader

Michael A. Morgan
Michael A. Morgan, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

This thesis deals with a simulation of a missile versus target engagement scenario. After deriving simplified transfer functions for the missile seeker head, missile autopilot, missile dynamics, and target dynamics, a three dimensional simulation is developed using classical proportional navigation. The scenario is simulated using state variable design. A forward time solution of the two dimensional problem is developed which is converted to an adjoint model. The adjoint model is used to determine the optimal time to initiate simplified and tactical evasion maneuvers in order to maximize the final miss distance.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logical errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I certainly would like to thank Professor Hal Titus for his support and good humor which were priceless in finishing this work. His dedication to the students of the Naval Postgraduate School is unsurpassed.

I would also like to thank Colin Cooper for his invaluable assistance throughout the completion of this thesis.

Of course, my wife Joy and daughter Michelle are to be especially thanked for their unyielding support .

I. INTRODUCTION

Since the early years of the Vietnam War the surface-to-air missile has become the largest threat to combat aviators. During the emergent age of low intensity conflicts, a goal of zero percent aircraft losses must remain a high priority to strike planners and decision makers. In order to minimize losses to enemy antiair weapon systems, in particular surface-to-air missiles, these systems must be generally understood and tactics necessary to defeat these missiles must be investigated.

Aircraft combat survivability is defined as "the capability of an aircraft to avoid and/or withstand a man-made hostile environment." [Ref. 1] Figure 1 displays the basic aircraft survivability relationships. The probability of an aircraft being hit by antiair weapons is termed P_H which is referred to as the susceptibility of the aircraft. Susceptibility is calculated by the product

$$P_H = P_A \cdot P_{DIT} \cdot P_{LGD} \qquad (1.1)$$

where

P_A = probability the threat is active and ready to engage the aircraft.

P_{DIT} = probability the aircraft is detected, identified, and tracked.

P_{LGD} = probability the threat is launched, guided and either hits the aircraft or detonates close enough to cause a hit.

The vulnerability of an aircraft is defined as the inability to withstand the damage caused by a hostile environment. It can be measured by $P_{K/H}$ which is the probability of an aircraft kill given it is hit by hostile fire. This leads to the relationship

$$\text{probability of kill} = \text{susceptibility} \cdot \text{vulnerability}, \qquad (1.2)$$

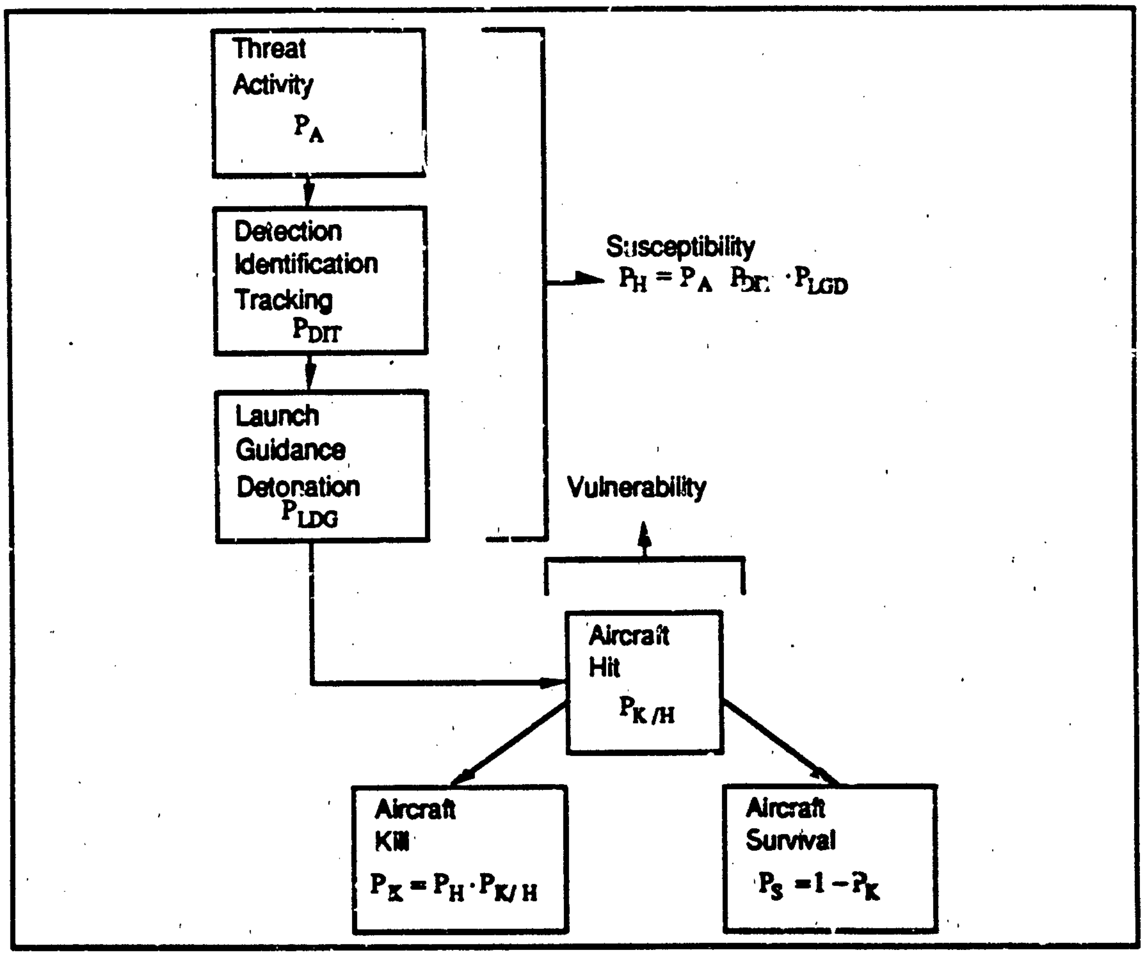


Figure 1. Aircraft Survivability Diagram

or in other words,

$$P_K = P_H \cdot P_{K/H}. \tag{1.3}$$

It is obvious then that the probability of survival is

$$P_S = 1 - P_K. \text{ [Ref. 1]} \tag{1.4}$$

Therefore, in order to increase the survivability of an aircraft, the probability of kill must be reduced. One way that this can be achieved is by decreasing P_{LGD}. The probability that a missile is successfully guided and either hits an aircraft or detonates close enough to hit it, can be reduced by degrading the guidance and control system of the missile. The employment of high acceleration turns and random maneuvers are examples of methods that can be used.

This research develops a missile/target simulation program using classical proportional navigation. A three dimensional model is produced assuming a dual gimbal axis seeker head. Chapter II introduces the idea of proportional navigation and the actual guidance law is developed. In Chapter III the transfer functions for the individual missile subsystems are determined. The specifics of the computer simulation geometry are discussed leading to the actual relationships used in the program. Additionally, a two dimensional forward time model is developed which is converted to an adjoint model. The adjoint model is a useful tool for analyzing time varying systems. It is used in this thesis to determine optimal miss distance parameters. Chapter IV consists of actual simulation results. Two typical scenarios are conducted to show the capabilities of the program against various target maneuvers. The adjoint model is then used to determine an optimal time to initiate various evasion maneuvers. Conclusions and recommendations follow in Chapter V.

All computer simulations are developed and conducted using the Matrix Laboratory (MATLAB) and the three dimensional plots are generated using the Display Integrated Software System and Plotting Language (DISSPLA).

II. BASIC PROPORTIONAL NAVIGATION

A. GENERAL

Proportional navigation is the basic guidance law used in the majority of the threat missiles in operational use today. This method of guidance generates missile acceleration commands proportional to the line of sight (LOS) rate. Figure 2 shows the basic parameters and geometry associated with the two dimensional missile/target engagement problem and the parameters are defined below:

V_t	= target velocity.
V_m	= missile velocity.
γ_t	= target flight path angle.
γ_m	= missile flight path angle.
R	= missile to target range.
R_t	= target range.
R_m	= missile range.
σ	= line of sight angle.
σ_t	= target line of sight angle.
σ_m	= missile line of sight angle.

B. CONSTANT BEARING COURSF

A constant bearing course is one where the line of sight between the target and the missile maintains a constant orientation in space. As a result, progressive lines of sight remain parallel to each other as the engagement procedes. Similarly, the line of sight angle, σ, remains constant. Figure 3

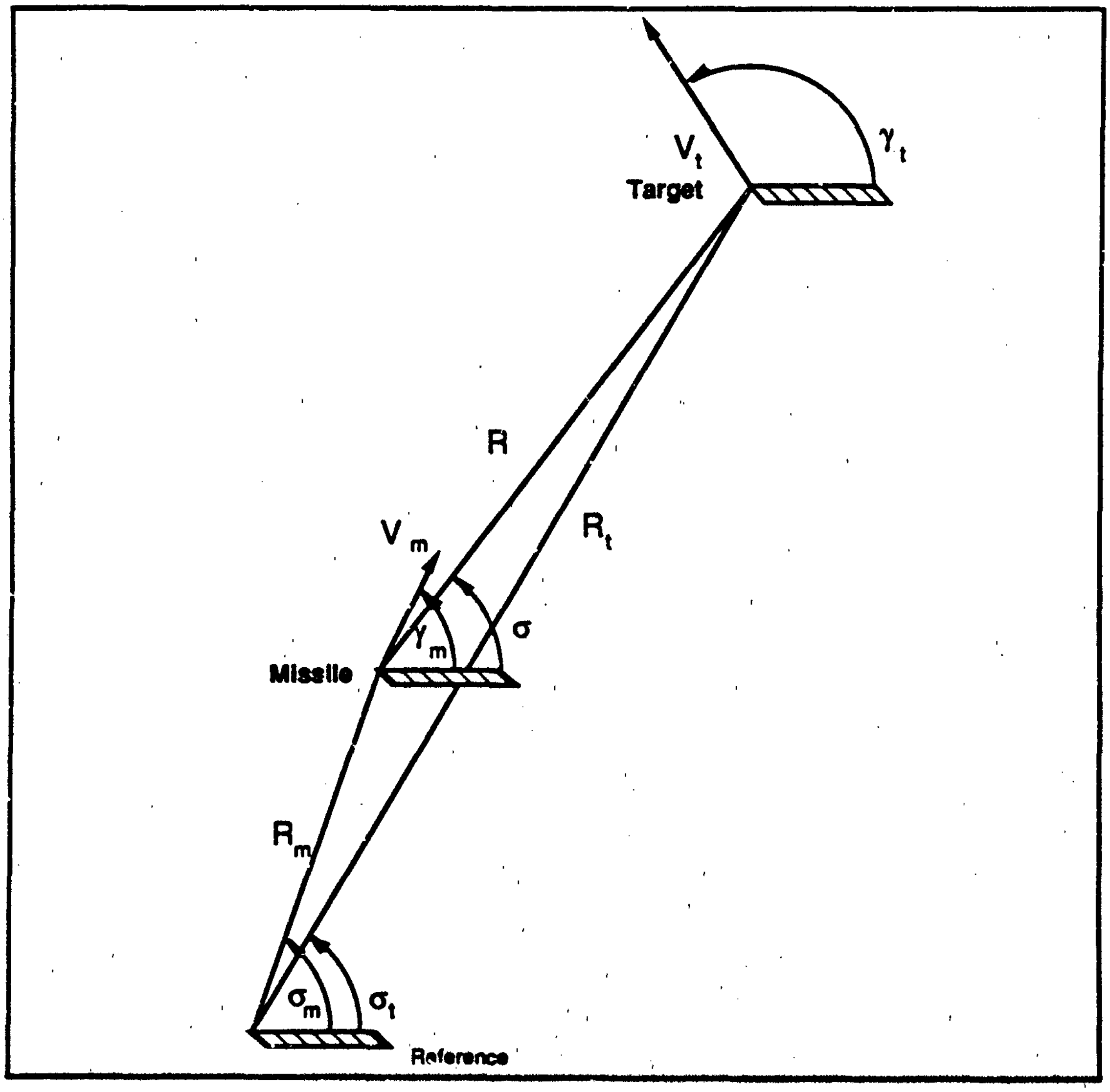


Figure 2. Missile/Target Geometry

depicts the constant bearing course idea. As long as there is a positive closing velocity between the missile and the target, the constant bearing course concept will ensure an intercept. Proportional navigation uses the constant bearing course idea by driving the line of sight rate, $\dot{\sigma}$, to zero. [Ref. 2]

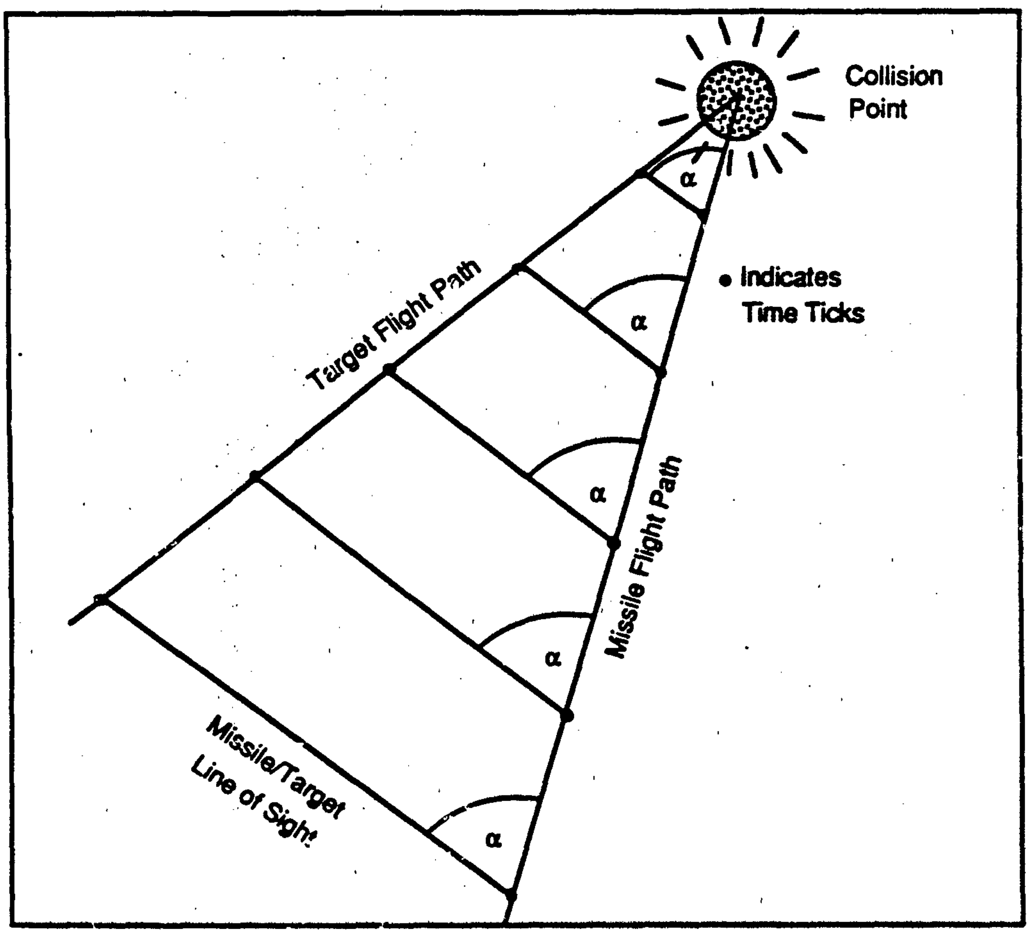


Figure 3. Constant Bearing Course

C. PROPORTIONAL NAVIGATION SCHEME

Figure 4 depicts the basic proportional navigation scheme. Assuming that the seeker head of the missile follows the target, the transverse acceleration perpendicular to the lin', of sight will equal the acceleration of the R vector in that direction. Mathematically, the accele;ation of R is

$$A_R = \left(\ddot{R} + \omega \times \omega \times R\right)\hat{i}_R + \left(2\omega \times \dot{R} \times R + \dot{\omega} \times R\right)\hat{i}_\sigma \qquad (2.1)$$

where

R = missile/target line of sight vector.

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

$\ddot{R}$ = acceleration along R.

ω = angular rate of change of R in inertial space.

A_m = missile acceleration perpendicular to R..

A_t = target acceleration perpendicular to R.

A_R = overall acceleration of R.

At this point, a missile acceleration, A_m, equal to the target acceleration, A_t, will make the line of sight parallel to its original direction. As long as $\dot{R}$ remains

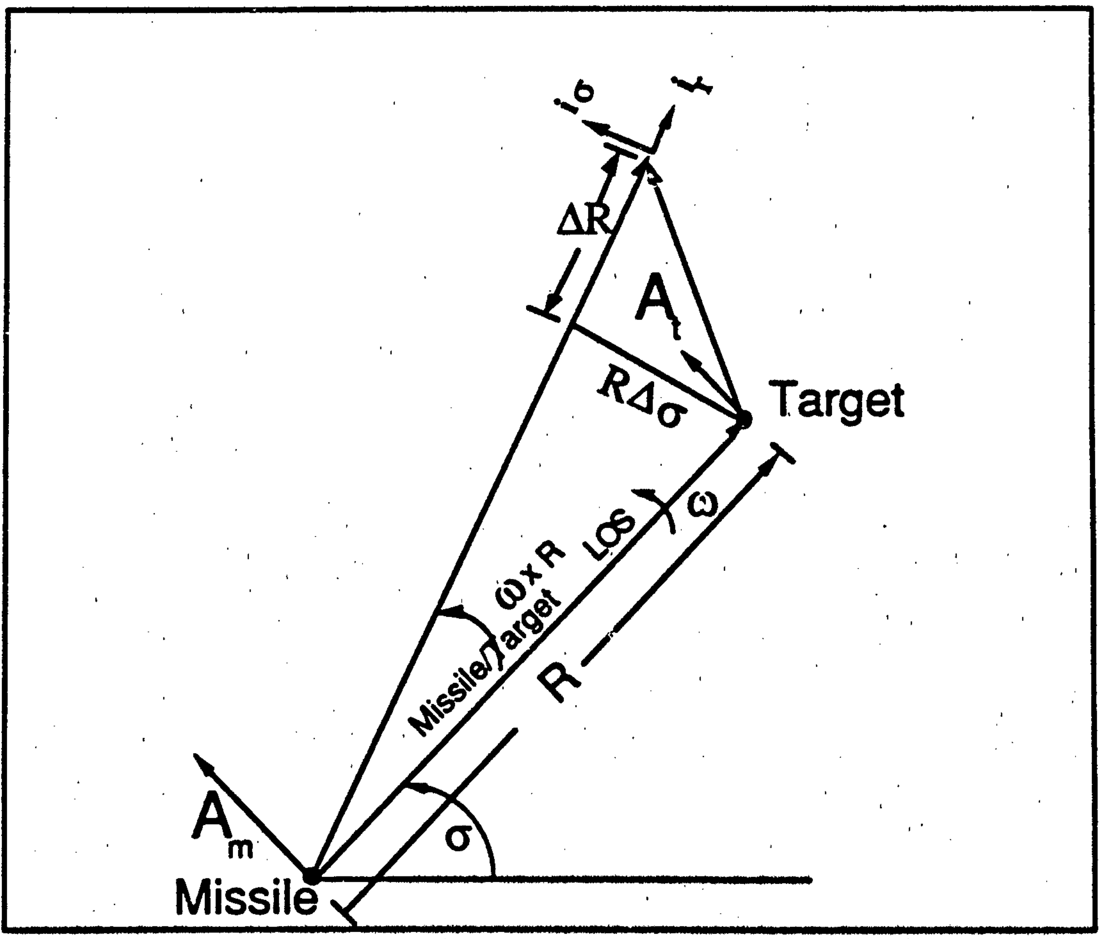


Figure 4. Vectorial Proportional Navigation Scheme

along R (ω=0) a missile/target impact is assured. So, the transverse acceleration command is

$$A_t - A_m = \dot{\omega} \times R + 2\left(\omega \times \dot{R}\right). \tag{2.2}$$

Assuming the line of sight rate is equal to the angular rate of change of R in inertial space, equation (2.2) now becomes

$$A_t - A_m = R\ddot{\sigma} + 2\dot{R}\dot{\sigma}. \tag{2.3}$$

D. PROPORTIONAL NAVIGATION GUIDANCE LAW

In the classical proportional navigation scheme, the missile course is one in which the rate of change of the missile heading is directly proportional to the rate of rotation of the line of sight vector from the missile to the target. As a result, this course change is intended to counteract the rotation of the line of sight, thus returning to a constant bearing course. The movement of the missile and target cause the line of sight to rotate resulting in a differential displacement between the missile and the targe. perpendicular to the range line. Figure 5 depicts this geometry. [Ref. 3] The proportional navigation guidance law attempts to generate an acceleration command, A_c, perpendicular to the line of sight.

Assume a gyro stabilized seeker head. If there is no torque applied to the gyro, the seeker will not rotate. Assuming the seeker tracks the target, the gyro angle will follow the line of sight. Applying the equation of motion for a gyro stabilized seeker

$$L = I\omega\Omega \tag{2.4}$$

where

L = applied torque.

ω = spin angular velocity.

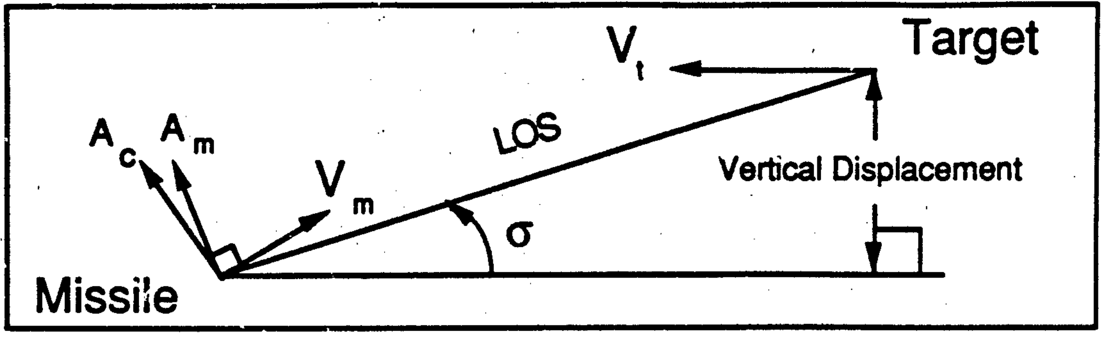


Figure 5. Missile Acceleration Orientation

I = moment of inertia of the gyroscope.

Ω = rate of precession of the gyroscope.

Applying this to the case when the seeker head tracks the target, Ω is then replaced by the rate at which the gyro is torqued in space. This is simply $\dot{\sigma}$ which is the line of sight rate. [Ref. 4] Thus, equation (2.4) becomes

$$L = I\omega\dot{\sigma}. \tag{2.5}$$

This torque is in turn applied to the control surface of the missile leading to the relationship

$$A_m = kL = kI\omega\dot{\sigma} \tag{2.6}$$

where k is a constant of proportionality. Referring to Figure 6, a relationship is determined for A_m in terms of the rate of change of the missile flight path angle, $\dot{\gamma}_m$. Given the missile velocity vector at some point in time, $V_m(t)$, and suppose the missile undergoes an acceleration, A_m, during an interval of time, dt. The velocity vector is then displaced and is represented by the vector $V_m(t+dt)$. The angle the vector is traversed is simply $d\gamma_m$, the differential missile flight

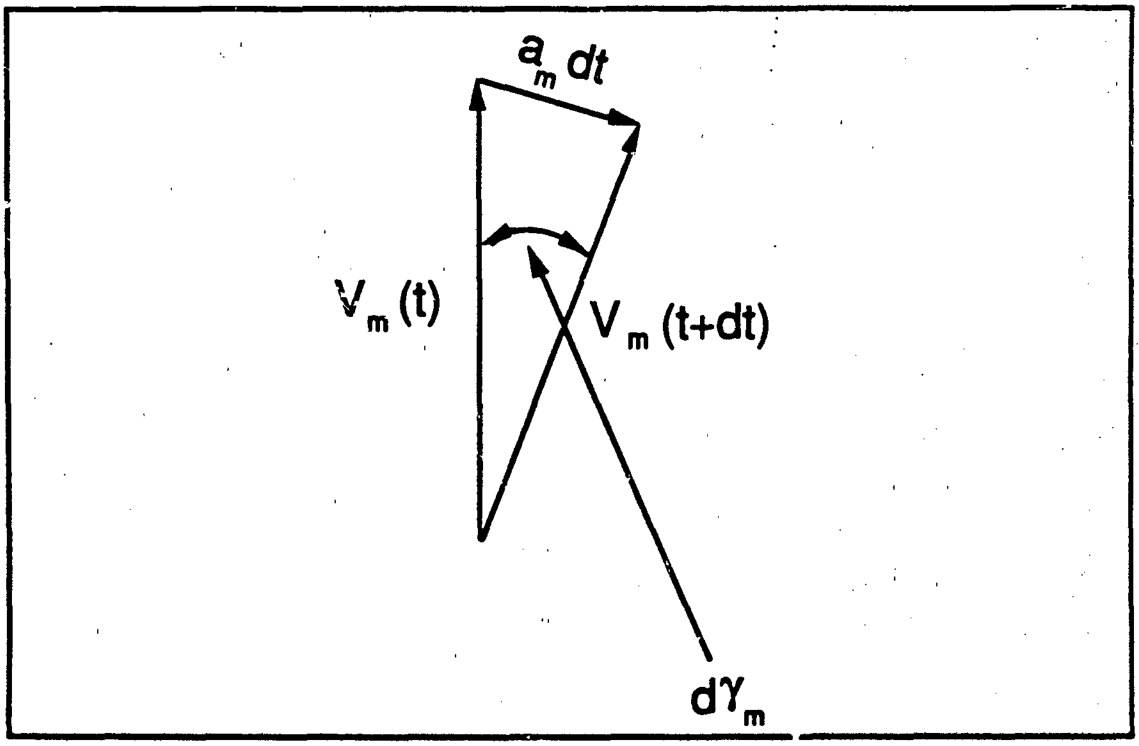


Figure 6. Missile Acceleration Relationship

path angle. [Ref. 4] For small angles (which are guaranteed by making dt small) the following relationship is obvious

$$A_m dt = V_m d\gamma_m. \tag{2.7}$$

Dividing equation (2.7) by the time interval, dt, the missile acceleration is defined as

$$A_m = V_m \frac{d\gamma_m}{dt} = V_m \dot{\gamma}_m. \tag{2.8}$$

Combining equations (2.6) and (2.8)

$$V_m \dot{\gamma}_m = kI\omega\dot{\sigma}. \tag{2.9}$$

Dividing through by V_m, the proportional navigation guidance law becomes

$$\dot{\gamma}_m = \left(\frac{kI\omega}{V_m}\right)\dot{\sigma}, \tag{2.10}$$

or

$$\dot{\gamma}_m = N\dot{\sigma}. \qquad (2.11)$$

Equation (2.11) represents the classical proportional navigation equation where

$\dot{\gamma}_m$	=	rate of change of the missile heading.
$\dot{\sigma}$	=	rate of change of the line of sight.
N	=	proportional navigation ratio.

The navigation ratio determines the sensitivity of the missile system. A high navigation ratio will lead to rather high gains resulting in large missile commands for small changes in the line of sight rate. On the other hand, small values for N will lead to small missile commands for a given $\dot{\sigma}$. Larger navigation ratios are preferred for head on engagements and smaller ones are preferred for tail chase cases. For this research the navigation ratio between three and five was chosen [Ref. 4].

III. SIMULATION DEVELOPMENT

A. INTRODUCTION

The development of the actual seeker head and autopilot transfer functions is discussed. Ideal missile and target trajectory equations are used and the state equations for these systems are generated. Basic geometric relationships are used to develop the missile and target proportional navigation equations. Once these equations are determined for the two dimensional problem, the state equations are then augmented to a three dimensional problem. The continuous state equations for the seeker head, autopilot, missile kinematics, and target kinematics are converted to equations appropriate for digital simulation. Finally, the two dimensional forward time model is developed and converted to an adjoint model.

B. MISSILE SUBSYSTEM DEVELOPMENT

1. Subsystem Layout

A basic functional block diagram of a generic tactical missile system is shown in Figure 7. Although the exact configuration and description of each component depends on many different factors, an attempt will be made to develop a basic system sufficient for simplified simulation. Figure 8 shows the typical physical location of the various missile subsystems.

2. Seeker Head Development

The seeker head can simply be thought of as the eye of the missile. It is able to detect, acquire, and track a target by sensing some unique

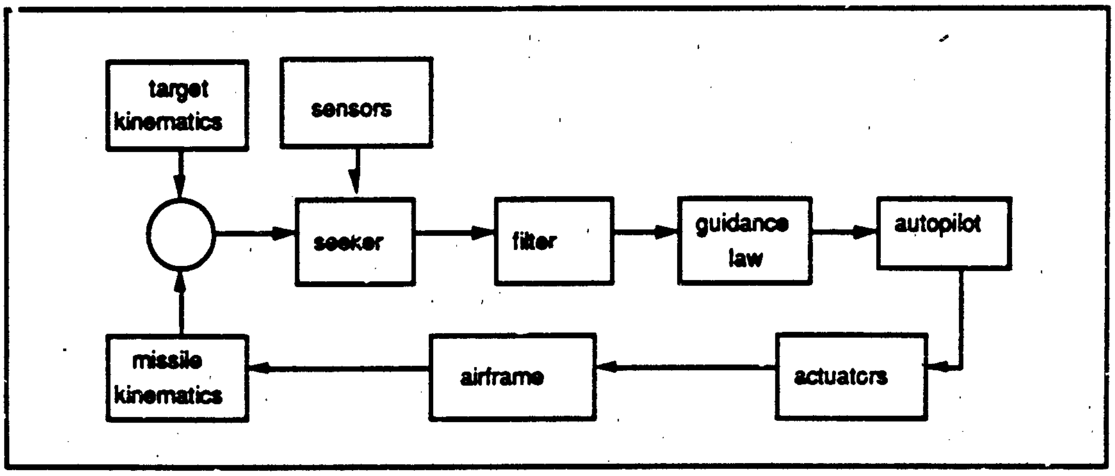


Figure 7. Missile Subsystem Block Diagram

characteristic of the target itself. This usually consists of the radiation or reflection of energy by the target.

A seeker with a narrow field of view will be used. Figure 9 shows a basic gimballed seeker head configuration. Here, the actual seeker is mounted on a gimballed platform and it maintains the target within the field of view by rotating the platform. The inertial rotation rate of the line of sight provides the missile with the required tracking information. [Ref. 5]

Figure 10 displays a diagram for an actual seeker head where

β = seeker head gimbal angle.

The control torque to the seeker results in the following equation of motion

$$T = I\ddot{\beta} \qquad (3.1)$$

where

T = applied torque.

I = moment of inertia.

$\ddot{\beta}$ = angular acceleration.

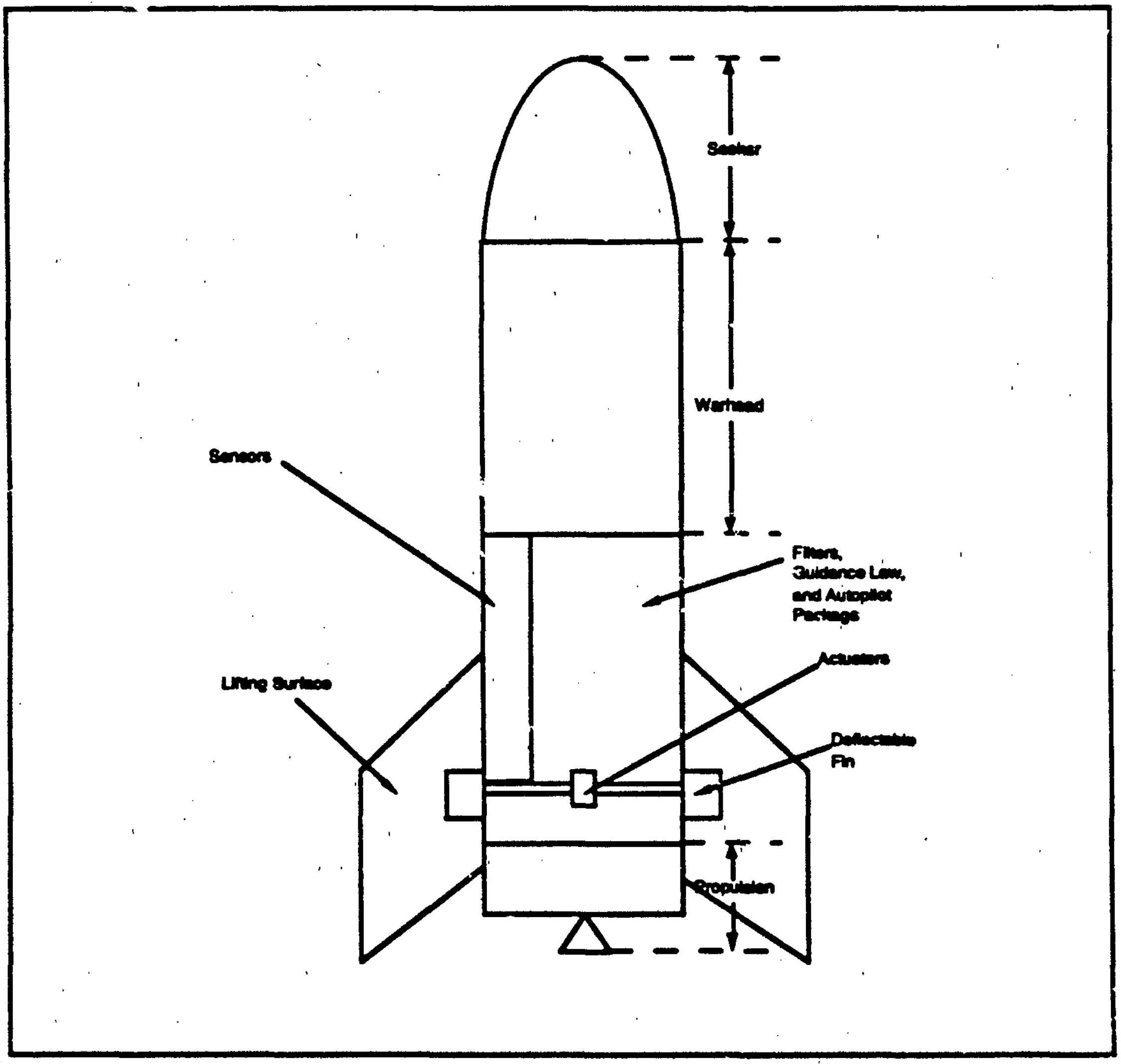


Figure 8. Basic Missile Layout

Solving for $\ddot{\beta}$

$$\ddot{\beta} = \frac{T}{I} = -k_1(\beta - \sigma) - k_2\dot{\beta} \tag{3.2}$$

$$= -k_2\dot{\beta} - k_1\beta + k_1\sigma. \tag{3.3}$$

The constants k_1 and k_2 are determined by the time constant of the seeker head system.

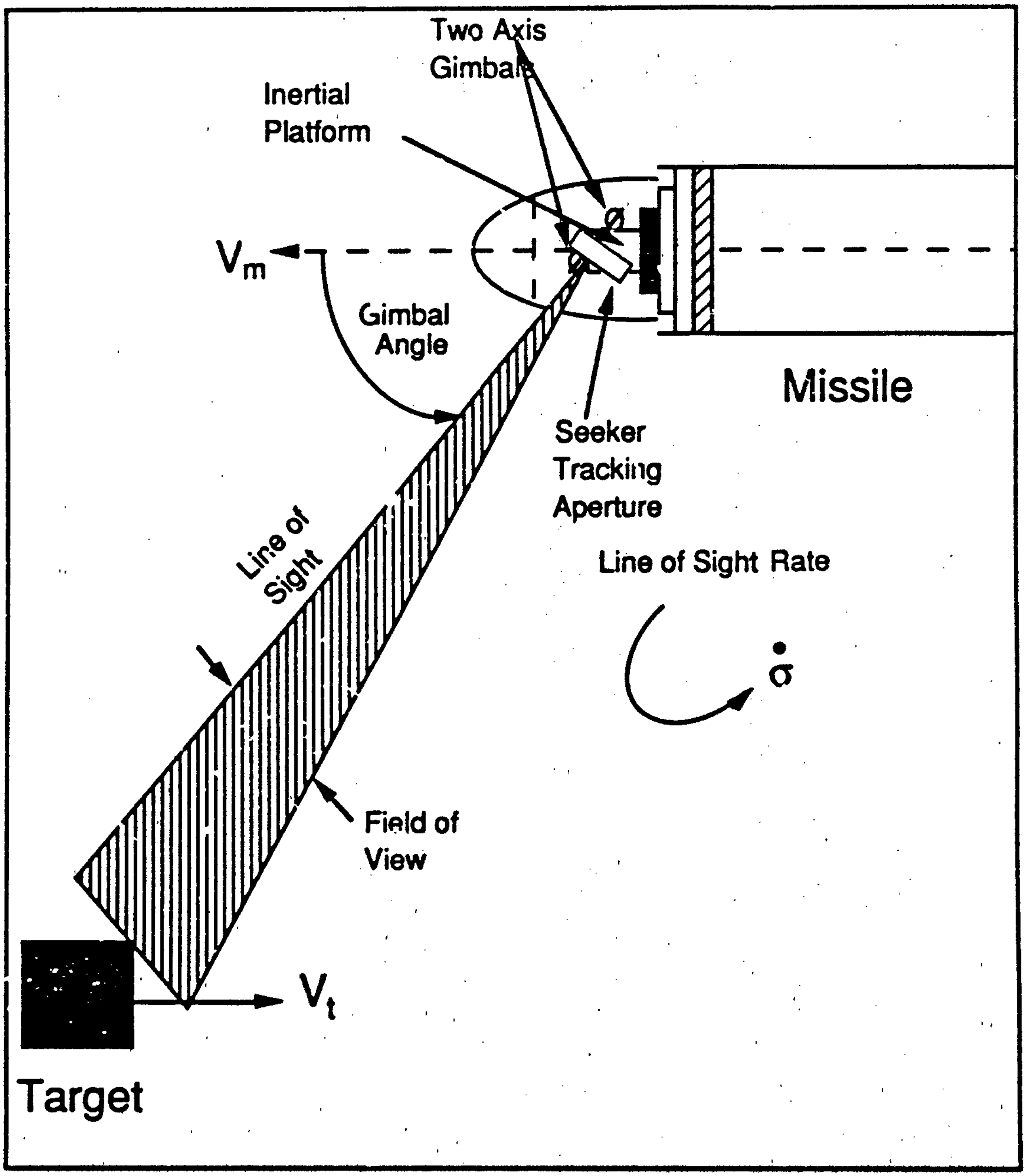


Figure 9. Gimballed Seeker Configuration

Now taking the Laplace Transform of equation (3.3) and assuming zero initial conditions

$$\mathcal{L}\left\{\ddot{\beta}\right\} = \mathcal{L}\left\{-k_2\dot{\beta} - k_1\beta + k_1\sigma\right\} \tag{3.4}$$

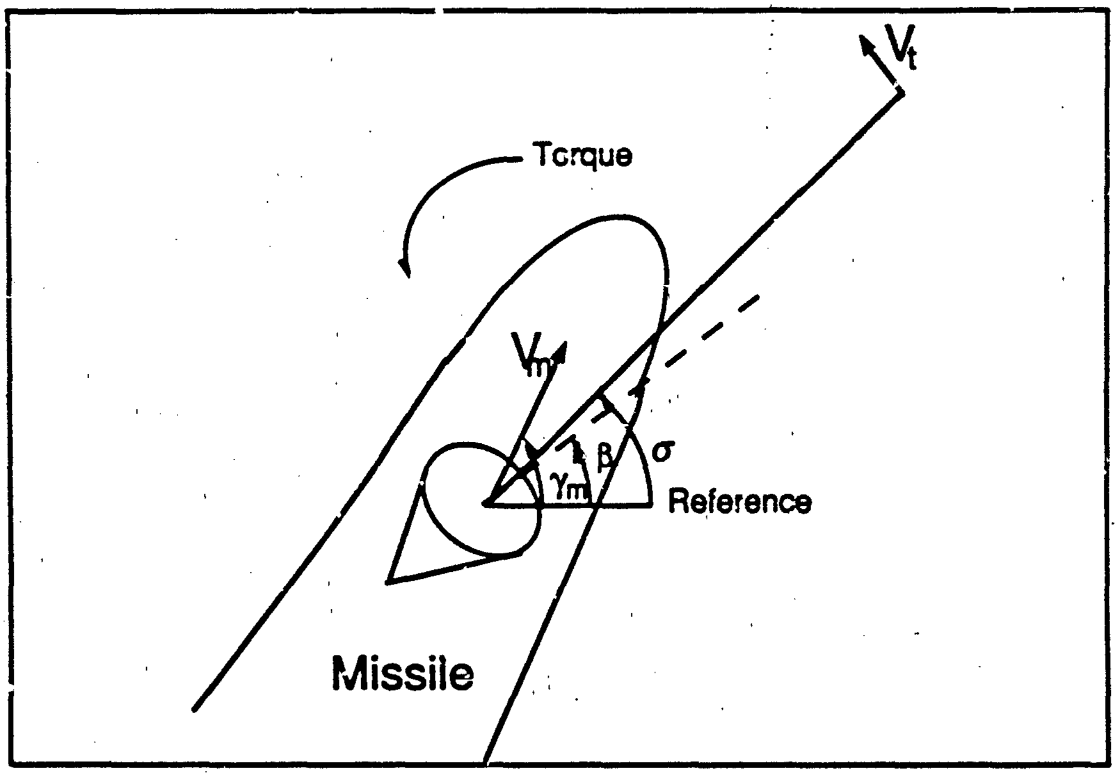


Figure 10. Seeker Head Angle Definitions

$$s^2\beta(s) = -k_2 s\beta(s) - k_1\beta(s) + k_1\sigma(s) \tag{3.5}$$

$$s^2\beta(s) + k_2 s\beta(s) + k_1\beta(s) = k_1\sigma(s). \tag{3.6}$$

Finally, the transfer function becomes

$$\frac{\beta(s)}{\sigma(s)} = \frac{k_1}{s^2 + k_2 s + k_1} = \frac{k_1}{(s + 1/\tau_{sh})^2}. \tag{3.7}$$

For this simulation the time constant of the seeker head, τ_{sh}, was chosen to be 0.1 second which is a good approximation of a real world system. From equation (3.7) the constants k_1 and k_2 are defined in terms of this time constant where

$$k_1 = (1/\tau_{sh})^2 = 100, \tag{3.8}$$

$$k_2 = 2(1/\tau_{sh}) = 20. \tag{3.9}$$

Figure 11 depicts the signal flow graph of the seeker head system. From this diagram the continuous time state equation of the form

$$\dot{x}_{sh} = Ax_{sh} + Bu_{sh} \tag{3.10}$$

is easily determined. Selecting the state vector to be

$$x_{sh} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \beta \\ \dot{\beta} \end{bmatrix} \tag{3.11}$$

and the input $u_{sh} = \sigma$, equation (3.10) becomes

$$\dot{x}_{sh} = \begin{bmatrix} 0 & 1 \\ -100 & -20 \end{bmatrix} x_{sh} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{sh}. \tag{3.12}$$

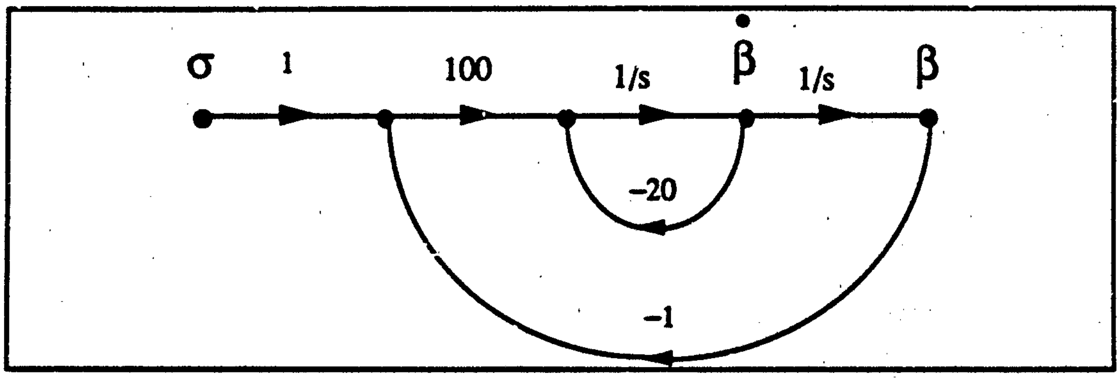


Figure 11. Seeker Head Signal Flow Graph

3. Guidance and Autopilot Development

The guidance law within the missile system determines the best trajectory for the missile based on the missile position, target position, missile capability and the desired objectives. A command is sent to the autopilot which determines the control (i.e., actuator position and thrust) necessary to perform such a command. In the proportional navigation guidance law, the missile commands are generated in order to change the missile flight path rate proportionally to the missile to target line of sight rate.

In this simulation a very simplified autopilot will be used. Referring to Figure 12, the transfer function is easily developed. The angle of attack of the missile is assumed to be zero, so the velocity of the missile will be aligned with the missile center line at an angle, γ_m. Applying a torque to the missile about the center of gravity results in tl following equation of motion

$$T_{com} = I_{cg}\ddot{\gamma} \tag{3.13}$$

where

T_{com}	=	commanded torque.
I_{cg}	=	moment of inertia about the center of gravity.
$\ddot{\gamma}$	=	angular acceleration of the missile flight path.

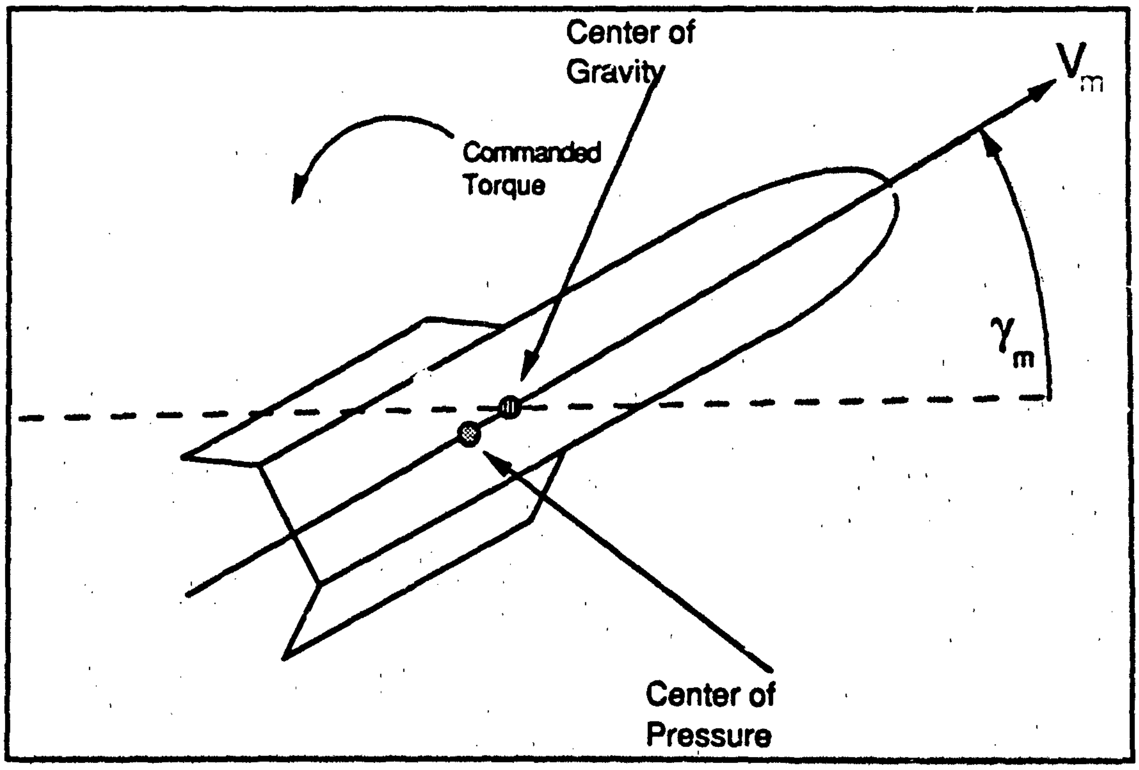


Figure 12. Basic Missile Layout

Solving for $\ddot{\gamma}$,

$$\ddot{\gamma} = \frac{T_{com}}{I_{cg}} = -k\dot{\gamma}_m + kN\dot{\beta} \tag{3.14}$$

where k is determined by the slowest time constant of the missile/autopilot.

Now taking the Laplace Transform of equation (3.14) and assuming zero initial conditions

$$\mathcal{L}\{\ddot{\gamma}_m\} = \mathcal{L}\left\{k\dot{\gamma}_m + kN\dot{\beta}\right\} \tag{3.15}$$

$$s^2\gamma_m(s) = -ks\gamma_m(s) + kNs\gamma_m(s) \tag{3.16}$$

$$\left(s^2 + ks\right)\gamma_m(s) = kNs\beta(s)\,. \tag{3.17}$$

Finally, the transfer function becomes

$$\frac{\dot{\gamma}_m(s)}{\dot{\beta}(s)} = \frac{kN}{s+k} \tag{3.18}$$

where k is defined above. The autopilot time constant ,τ_{ap},was chosen to be 1.0 second, so

$$k = 1/\tau_{ap} = 1.0. \tag{3.19}$$

Figure 13 shows the signal flow graph for the missile autopilot system. From this diagram the continuous state equation is easily determined. Selecting the state vector to be

$$x_{ap} = [x_1] = [\dot{\gamma}_m] \tag{3.20}$$

and the input u_{ap}=$N\dot{\beta}$, the state equation becomes

$$\dot{x}_{ap} = [-1]x_{ap} + [1]u_{ap}. \tag{3.21}$$

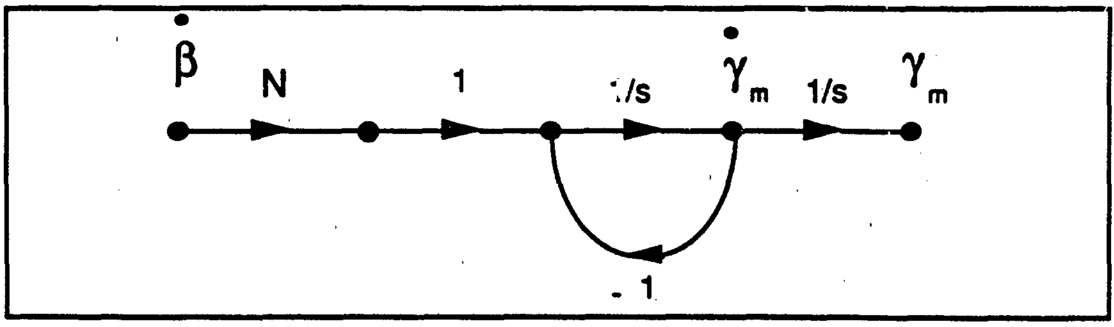


Figure 13. Guidance Autopilot Signal Flow Graph

4. Missile and Target Dynamics

The signal flow graphs for the target and missile dynamics are shown in Figure 14 where $A_m = V_m\dot{\gamma}_m$. A_t is selected independently. The state vectors are

$$x_m = \begin{bmatrix} x_m \\ \dot{x}_m \\ y_m \\ \dot{y}_m \end{bmatrix} \tag{3.22}$$

and

$$x_t = \begin{bmatrix} x_t \\ \dot{x}_t \\ y_t \\ \dot{y}_t \end{bmatrix}. \tag{3.23}$$

With $u_m = \begin{bmatrix} u_{mx} \\ u_{my} \end{bmatrix}$ and $u_t = \begin{bmatrix} u_{tx} \\ u_{ty} \end{bmatrix}$, the state equations become

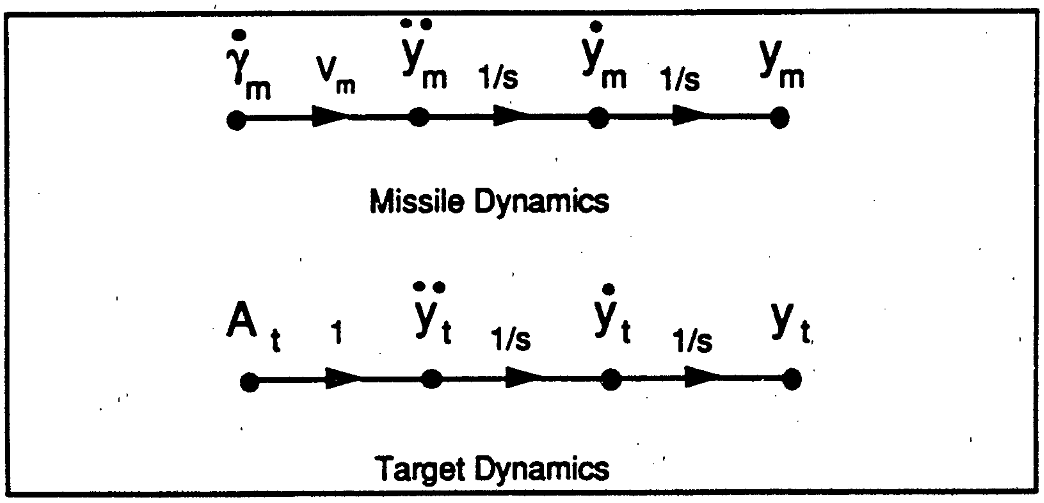


Figure 14. Missile and Target Dynamics

$$\dot{x}_m = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_m + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_m \tag{3.24}$$

and

$$\dot{x}_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_t. \tag{3.25}$$

5. Overall System Layout

Figure 15 shows the total system signal flow graph. The seeker head angle rate, $\dot{\beta}$, is the estimate of the line of sight rate, $\dot{\sigma}$.

6. Three Dimensional State Definitions

Assuming that the missile is roll stabilized and controllable in both pitch and yaw, the state equations for the missile subsystems are extended.

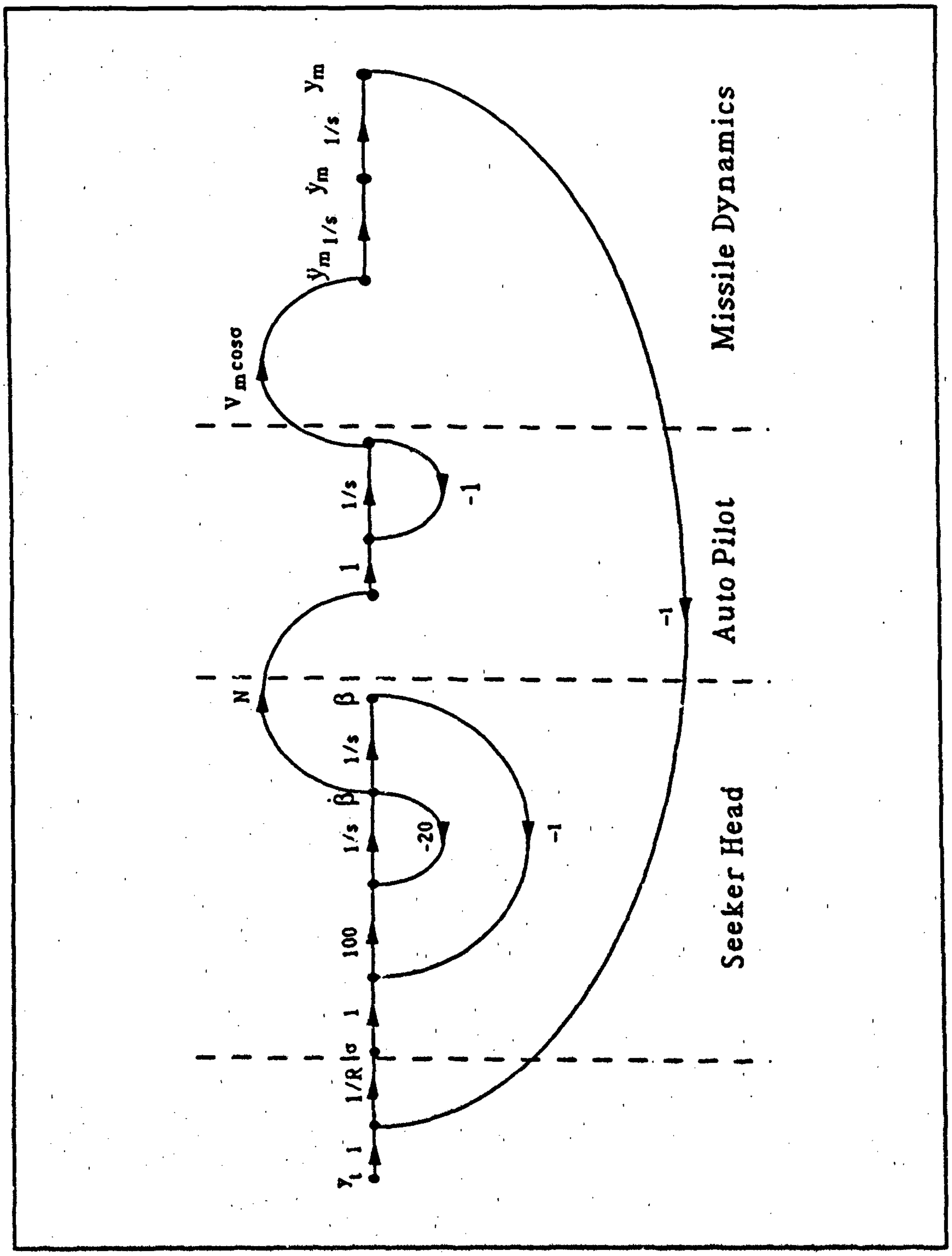


Figure 15. Total System Signal Flow Graph

The seeker head state matrix now becomes

$$x_{sh} = \begin{bmatrix} \beta_{pitch} \\ \dot{\beta}_{pitch} \\ \beta_{yaw} \\ \dot{\beta}_{yaw} \end{bmatrix} \tag{3.26}$$

and the continuous state equation is obtained by simply augmenting the two dimensional problem.

The resulting equation becomes

$$\dot{x}_{sh} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -100 & -20 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -100 & -20 \end{bmatrix} x_{sh} + \begin{bmatrix} 0 & 0 \\ 100 & 0 \\ 0 & 0 \\ 0 & 100 \end{bmatrix} u_{sh} \tag{3.27}$$

where $u_{sh} = \begin{bmatrix} \sigma_{pitch} \\ \sigma_{yaw} \end{bmatrix}$.

The autopilot state matrix now becomes

$$x_{ap} = \begin{bmatrix} \dot{\gamma}_{m\_pitch} \\ \dot{\gamma}_{m\_yaw} \end{bmatrix} \tag{3.28}$$

and the continuous state equation is again simply obtained from an augmentation of the two dimensional equation. The resulting equation for the autopilot is

$$\dot{x}_{ap} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x_{ap} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u_{ap} \tag{3.29}$$

where $u_{ap} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \cdot \begin{bmatrix} \dot{\beta}_{pitch} \\ \dot{\beta}_{yaw} \end{bmatrix}$.

For the missile and target dynamics equations the state equations are simply augmented with vertical position and vertical velocity states.

These equations are

$$x = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} \tag{3.30}$$

and the state variable dynamics become

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} u \tag{3.31}$$

where $u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$, the force per unit mass acting on the missile or target.

C. MISSILE AND TARGET GEOMETRY

1. Velocity Relationships

Figure 16 shows the vectorial relationship of the missile and target velocity. From this figure

$$V_m = \sqrt{{V_{mx}}^2 + {V_{my}}^2 + {V_{mz}}^2} \tag{3.32}$$

and

$$V_t = \sqrt{{V_{tx}}^2 + {V_{ty}}^2 + {V_{tz}}^2}. \tag{3.33}$$

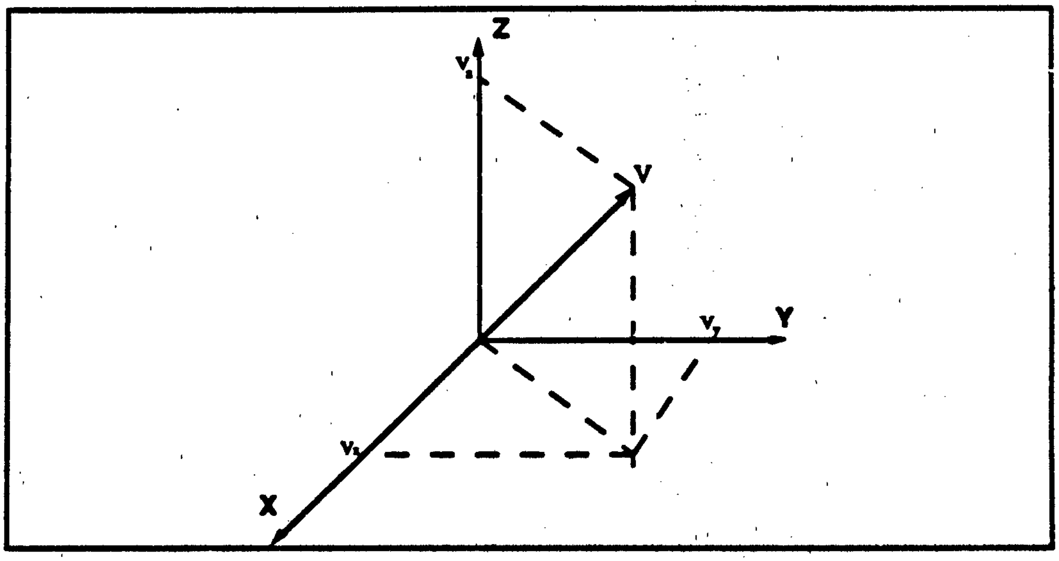


Figure 16. Velocity Relationship

2. Line of Sight Angles

In order to simulate the scenario in three dimensional space a line of sight angle in pitch and yaw must be defined. Figure 17 depicts these angular relationships. Given the target and missile position in Cartesian coordinates the line of sight angles are defined as

$$\sigma_{pitch} = \tan^{-1}\left[(z_t - z_m)/\sqrt{(x_t - x_m)^2 + (y_t - y_m)^2}\right] \tag{3.34}$$

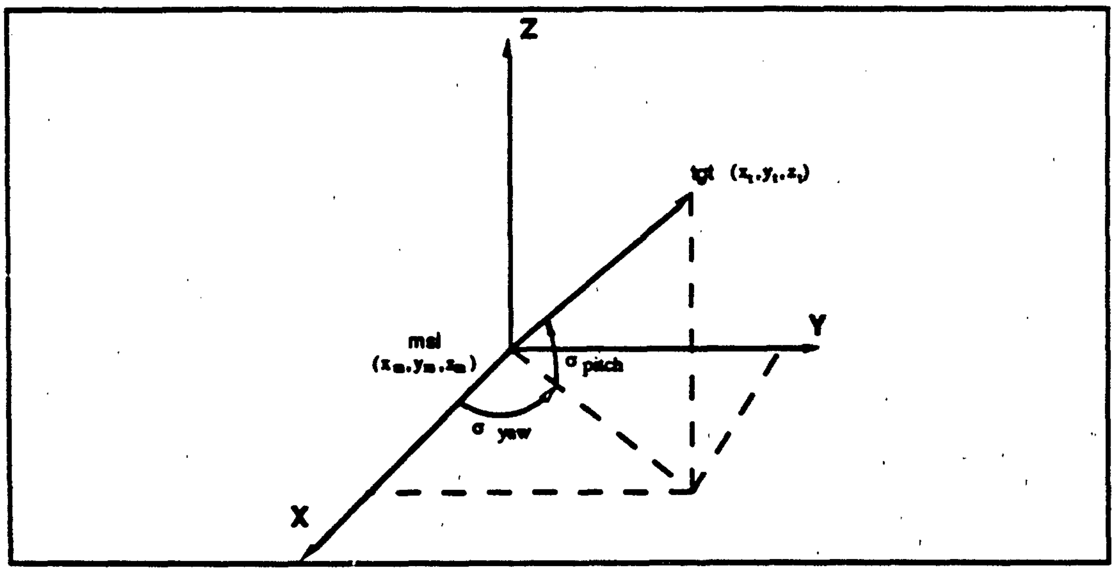


Figure 17. Pitch and Yaw Line of Sight Angle Definitions

and

$$\sigma_{yaw} = \tan^{-1}[(y_t - y_m)/(x_t - x_m)]. \qquad (3.35)$$

Figure 18 shows the target and missile line of sight angle definitions and these angles are expressed mathematically as

$$\sigma_{m\_pitch} = \tan^{-1}\left[z_m / \sqrt{x_m{}^2 + y_m{}^2}\right], \qquad (3.36)$$

$$\sigma_{m\_yaw} = \tan^{-1}[y_m / x_m], \qquad (3.37)$$

and

$$\sigma_{t\_pitch} = \tan^{-1}\left[z_t / \sqrt{x_t{}^2 + y_t{}^2}\right], \qquad (3.38)$$

$$\sigma_{t\_yaw} = \tan^{-1}[y_t / x_t]. \qquad (3.39)$$

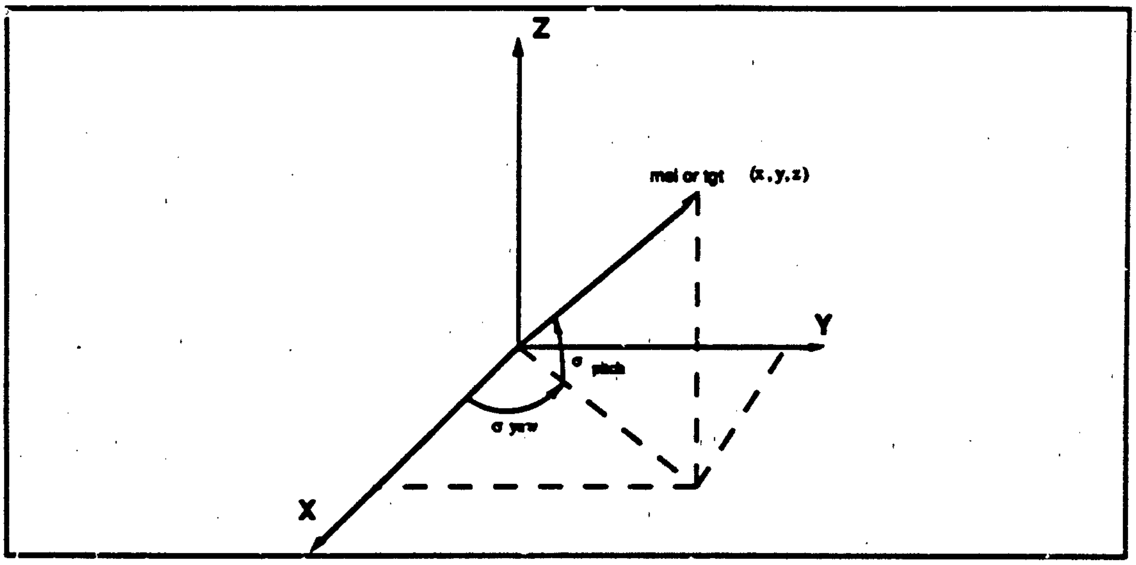


Figure 18. Target and Missile Line of Sight Angle Definitions

3. Flight Path Angles

The missile and target flight path angles are also defined in three dimensions. Figure 19 depicts the geometric definitions. Mathematically,

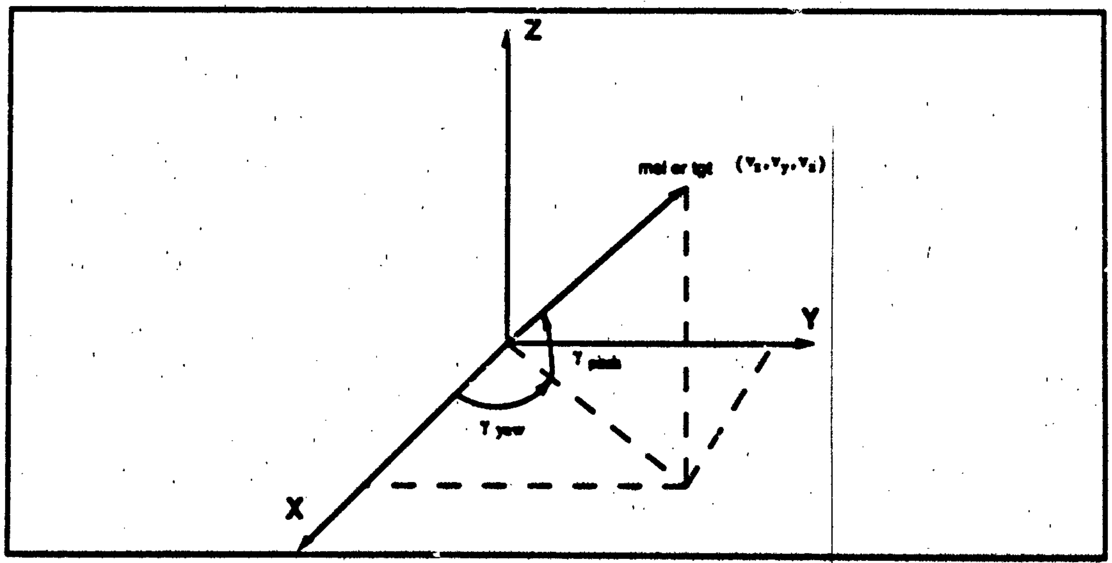


Figure 19. Target and Missile Flight Path Angle Definitions

$$\gamma_{m\_pitch} = \tan^{-1}\left[V_{mz} / \sqrt{V_{mx}{}^2 + V_{my}{}^2}\right], \tag{3.40}$$

$$\gamma_{m\_yaw} = \tan^{-1}\left[V_{my} / V_{mx}\right], \tag{3.41}$$

and

$$\gamma_{t\_pitch} = \tan^{-1}\left[V_{tz} / \sqrt{V_{tx}{}^2 + V_{ty}{}^2}\right], \tag{3.42}$$

$$\gamma_{t\_yaw} = \tan^{-1}\left[V_{ty} / V_{tx}\right]. \tag{3.43}$$

4. Velocity and Acceleration in the Pitch Plane

The missile is controlled in three dimensional space by generating acceleration commands in two orthogonal planes. These planes are defined as the pitch plane and the yaw plane. The yaw plane is taken as simply the horizontal XY plane and the pitch plane is the orthogonal plane rotated by the angle σ_{yaw}. Figure 20 depicts the geometric definition of the pitch plane. Given

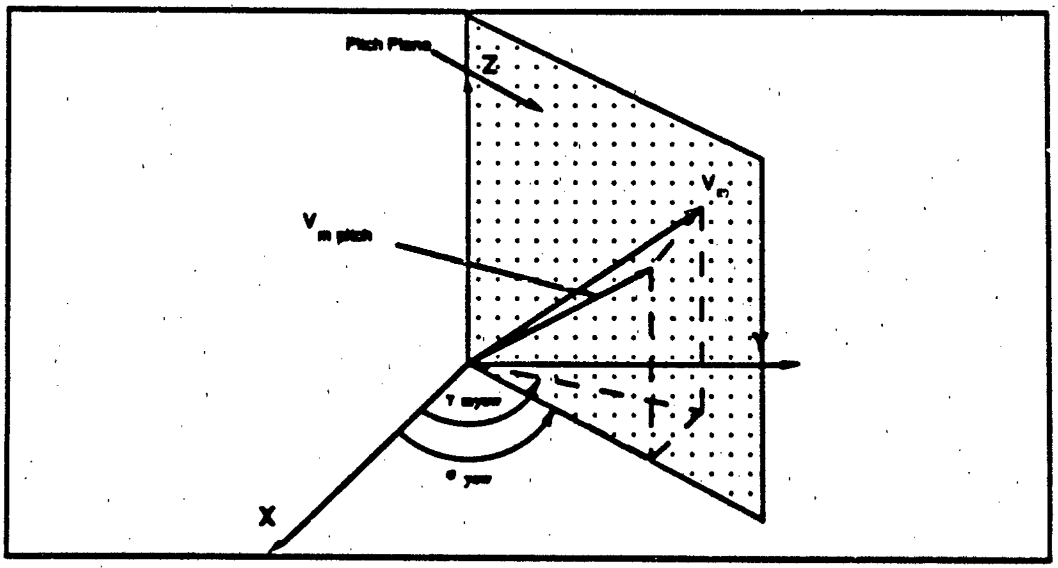


Figure 20. Pitch Plane Definition

the missile velocity, and the angles σ_{yaw} and γ_{m_yaw}, the velocity in the pitch plane is

$$V_{m\_pitch} = V_m \cos\left(\gamma_{m\_yaw} - \sigma_{yaw}\right). \tag{3.44}$$

From the basic relationship for the missile acceleration the missile pitch acceleration is defined as

$$A_{m\_pitch} = V_{m\_pitch} \cdot \dot{\gamma}_{m\_pitch}. \tag{3.45}$$

This acceleration vector is then broken down into Cartesian coordinate system components. Assuming that the pitch acceleration vector is perpendicular to the vertical line of sight vector between the missile and target, Figure 21 shows the orientation of the acceleration components. From this figure the following relationships are obtained

$$\ddot{x}_{m\_pitch} = -\left(A_{m\_pitch} \cdot \sin\sigma_{pitch}\right)\cos\sigma_{yaw}, \tag{3.46}$$

$$\ddot{y}_{m\_pitch} = -\left(A_{m\_pitch} \cdot \sin\sigma_{pitch}\right)\sin\sigma_{yaw}, \tag{3.47}$$

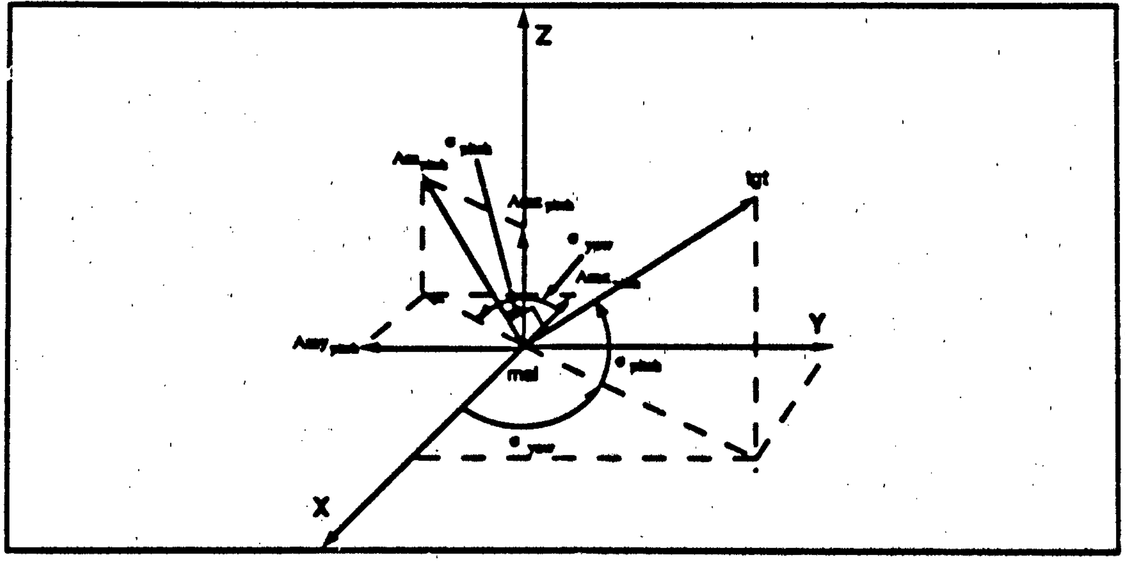


Figure 21. Pitch Plane Acceleration Components

and

$$\ddot{z}_{m\_pitch} = A_{m\_pitch} \cdot \cos\sigma_{pitch}. \tag{3.48}$$

5. Velocity and Acceleration in the Yaw Plane

The missile yaw plane is depicted in Figure 22. Given the missile velocity and the angle γ_{m_pitch}, the velocity in the yaw plane is simply

$$V_{m\_yaw} = V_m \cdot \cos\gamma_{m\_pitch}. \tag{3.49}$$

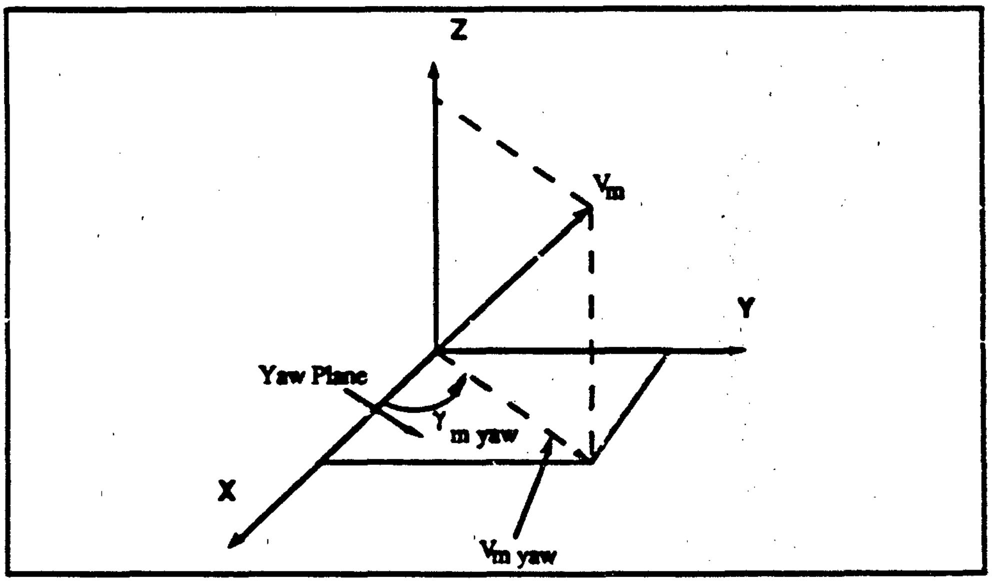


Figure 22. Yaw Plane Definition

Again, from the basic missile acceleration relationship, the missile yaw acceleration is defined as

$$A_{m\_yaw} = V_{m\_yaw} \cdot \dot{\gamma}_{m\_yaw}. \tag{3.50}$$

The missile yaw acceleration components are depicted in Figure 23 and are given mathematically as

$$\ddot{x}_{m\_yaw} = -A_{m\_yaw} \cdot \sin\sigma_{yaw}, \tag{3.51}$$

and

$$\ddot{y}_{m\_yaw} = A_{m\_yaw} \cdot \cos \sigma_{yaw}. \qquad (3.52)$$

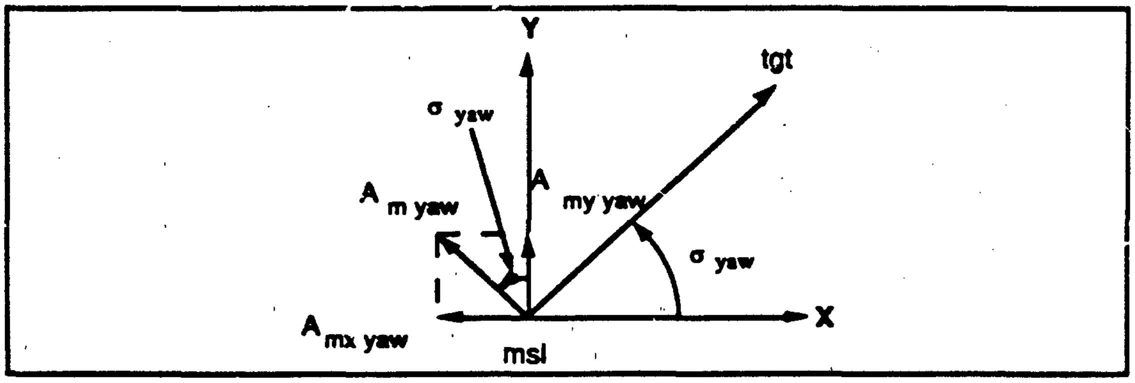


Figure 23. Yaw Plane Acceleration Components

6. Total Missile Acceleration Components

The input to the missile state equation is the vector consisting of the three Cartesian components of the overall missile acceleration vector. Given the missile acceleration components in both the pitch and yaw planes, the total missile acceleration components are

$$\ddot{x}_m = \ddot{x}_{m\_pitch} + \ddot{x}_{m\_yaw}, \qquad (3.53)$$

$$\ddot{y}_m = \ddot{y}_{m\_pitch} + \ddot{y}_{m\_yaw}, \qquad (3.54)$$

and

$$\ddot{z}_m = \ddot{z}_{m\_pitch}. \qquad (3.55)$$

Finally, the total missile acceleration is

$$A_m = \sqrt{\ddot{x}_m{}^2 + \ddot{y}_m{}^2 + \ddot{z}_m{}^2}. \qquad (3.56)$$

7. Target Acceleration Components

In this simulation the target acceleration components consist of the three Cartesian components of the overall target acceleration vector. These components are used as the input to the target kinematic equations. The input vector is determined by the type of target maneuver desired. The overall target acceleration is simply

$$A_t = \sqrt{\ddot{x}_t^{\,2} + \ddot{y}_t^{\,2} + \ddot{z}_t^{\,2}}. \tag{3.57}$$

8. Closing Velocity and Time To Go

From equation (3.48) the velocity of the target in the pitch plane is simply

$$V_{t\_pitch} = V_t \cos\left(\gamma_{t\_yaw} - \sigma_{yaw}\right). \tag{3.58}$$

Now, the range rate, $\dot{R}$, is found by simply projecting the missile and target pitch plane velocities along the vertical line of sight from the missile to the target. Mathematically

$$\dot{R} = V_{t\_pitch} \cdot \cos\left(\gamma_{t\_pitch} - \sigma_{pitch}\right) - V_{m\_pitch} \cdot \cos\left(\gamma_{m\_pitch} - \sigma_{pitch}\right). \tag{3.59}$$

Knowing that the range rate is the negative of the closing velocity, the time to go is simply

$$\text{Time} - \text{to} - \text{go} = \frac{R}{V_{closing}} = \frac{R}{-\dot{R}}, \tag{3.60}$$

where R is the range from the missile to the target.

D. DIGITAL SIMULATION USING STATE SPACE METHOD

1. Discrete State Equation Definition

Given the continuous state equations

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.61}$$

$$y(t) = Cx(t) + Du(t), \tag{3.62}$$

a more convenient form for digital simulation is

$$x(k+1) = \Phi x(k) + \Gamma u(k), \tag{3.63}$$

$$y(k+1) = Cx(k) + Du(k). \tag{3.64}$$

The discrete state matrices are defined as

$$\Phi = e^{A \cdot DT} = I + A \cdot DT + (1/2!)A^2 \cdot DT^2 + (1/3!)A^3 \cdot DT^3 + \ldots, \quad (3.65)$$

$$\Gamma = \left[I \cdot DT + (1/2!)A \cdot DT^2 + (1/3!)A^2 \cdot DT^3 + \ldots \right] B, \quad (3.66)$$

and for $u(t)$ constant over $[(k+1)-k] \cdot DT$.

2. Missile Subsystem Discrete State Equations

The missile/target scenario is simulated digitally using the MATLAB software package. A built in system function is used to convert the continuous state equations to discrete time state equations. The discrete state equations used in the simulation are defined as follows for the seeker head, autopilot, missile dynamics and target dynamics respectively;

$$x_{sh}(k+1) = \Phi_{sh} x_{sh}(k) + \Gamma_{sh} u_{sh}(k), \quad (3.67)$$

$$x_{ap}(k+1) = \Phi_{ap} x_{ap}(k) + \Gamma_{ap} u_{ap}(k), \quad (3.68)$$

$$x_m(k+1) = \Phi_m x_m(k) + \Gamma_m u_m(k), \quad (3.69)$$

and

$$x_t(k+1) = \Phi_t x_t(k) + \Gamma_t u_t(k). \quad (3.70)$$

E. FORWARD TIME AND ADJOINT MODEL

To accurately and simply assess miss distances for various target maneuvers, a two dimensional adjoint model is used. First, a forward time model is developed for the two dimensional missile/target engagement. Figure 24 shows the block diagram of the forward time model. All system inputs are converted to impulses for subsequent conversion to the adjoint model.

The forward time mode is used to show the miss distance due to heading error (where heading error is defined as the error in missile heading from a collision course) and the miss distance due to target maneuver. The missile to target range is defined as

$$R = V_c \cdot \text{time} - \text{to} - \text{go} \qquad (3.71)$$

where

$$\text{time} - \text{to} - \text{go} = T_{\text{final}} - t. \qquad (3.72)$$

This model shows the miss distance throughout the course of the engagement . The last value of miss distance is the miss distance at T_{final}.

An adjoint model is used to analyze linear time varying systems [Ref. 6]. The major advantage of the adjoint technique is that the miss distance for all final times is generated in one run rather than the multiple runs required when using a Monte Carlo simulation. Following the rules for adjoint construction [Ref. 6], Figure 24 is converted to the adjoint model depicted in Figure 25.

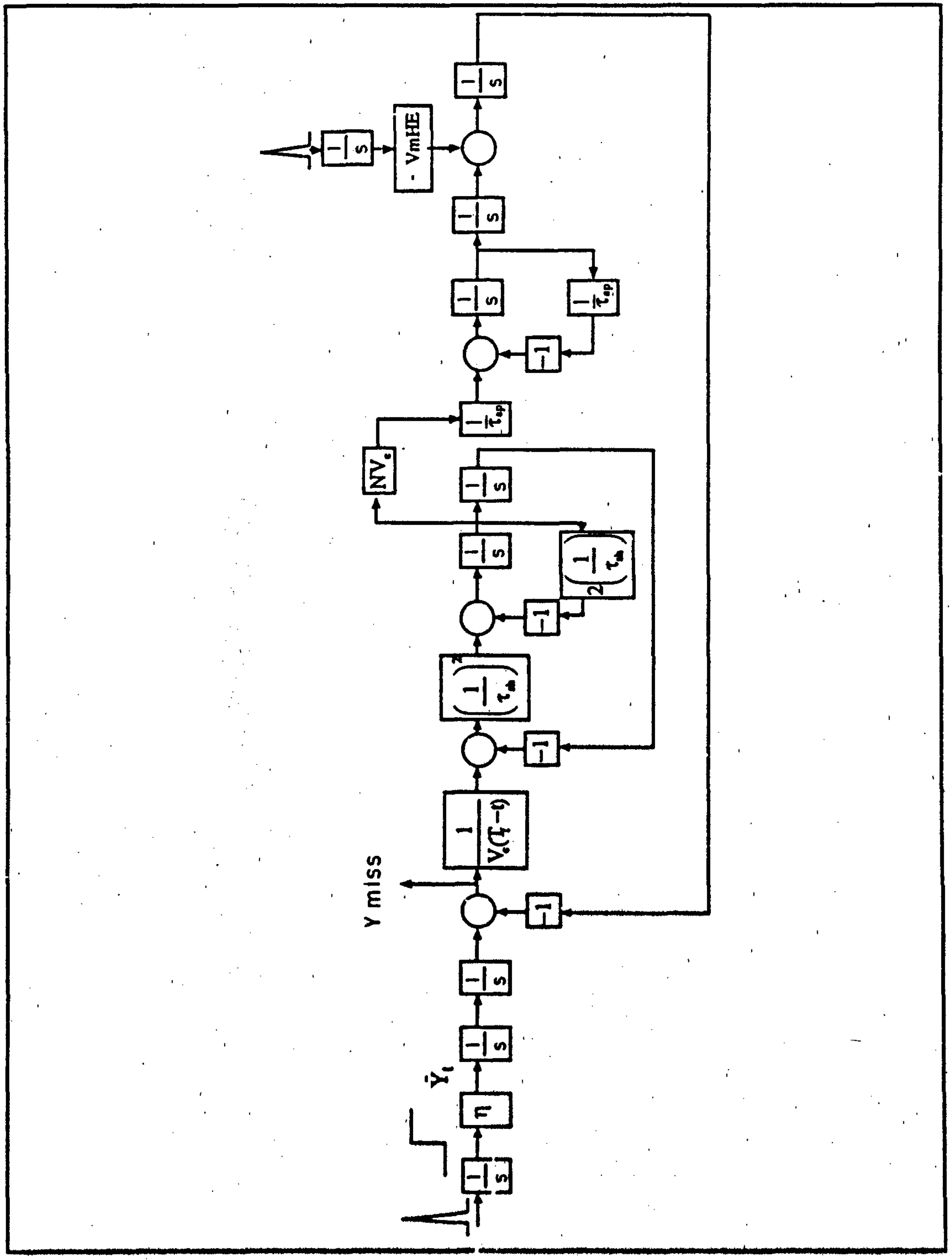
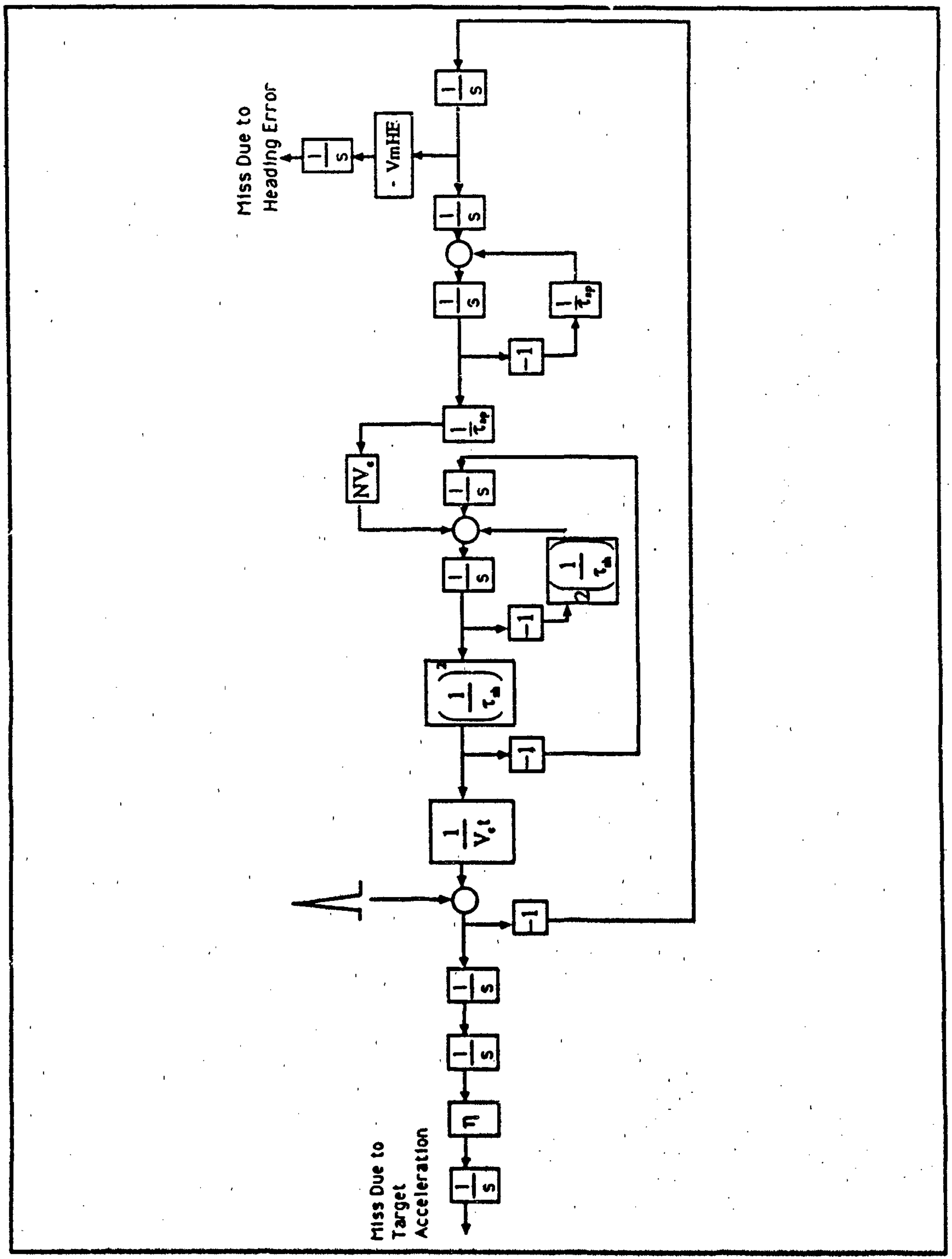


Figure 24. Forward Time Model



Miss Due to Heading Error
1/s
- VmHE
1/s
1/s
1/s
$1/\tau_{sp}$
-1
$1/\tau_{sp}$
NV_c
1/s
1/s
$2\left(\frac{1}{\tau_{ah}}\right)$
-1
$\left(\frac{1}{\tau_{ah}}\right)^2$
-1
$\frac{1}{V_c t}$
-1
1/s
1/s
η
1/s
Miss Due to Target Acceleration

Figure 25. Adjoint Model

IV. SIMULATION RESULTS

A. OVERVIEW

This section presents the results of the computer simulations. The scenario is conducted for several different initial conditions and with several different target maneuvers. The following assumptions are made throughout:

1. The missile is limited to 20 g's in either plane.
2. The acceleration due to gravity is ignored.
3. The target is capable of instantaneous acceleration.
4. The target is limited to a maximum of 8.0 g's.
5. The maximum missile speed is 3000 feet per second.
6. The maximum target speed is 1500 feet per second.
7. The proportional navigation constant is 4.
8. The missile is pointed toward the target at launch.

B. THREE DIMENSIONAL ENGAGEMENTS AGAINST VARIOUS TARGET MANEUVERS

1. Scenario 1: A Constant Velocity Target

The simulation is initially conducted with the missile engaging a constant velocity target. Figure 26 depicts the scenario geometry. The initial conditions are

$$\begin{aligned} xm(0) &= 0 \text{ feet} \\ ym(0) &= 0 \text{ feet} \\ zm(0) &= 0 \text{ feet} \\ vmx(0) &= 3000 \text{ fps} \end{aligned}$$

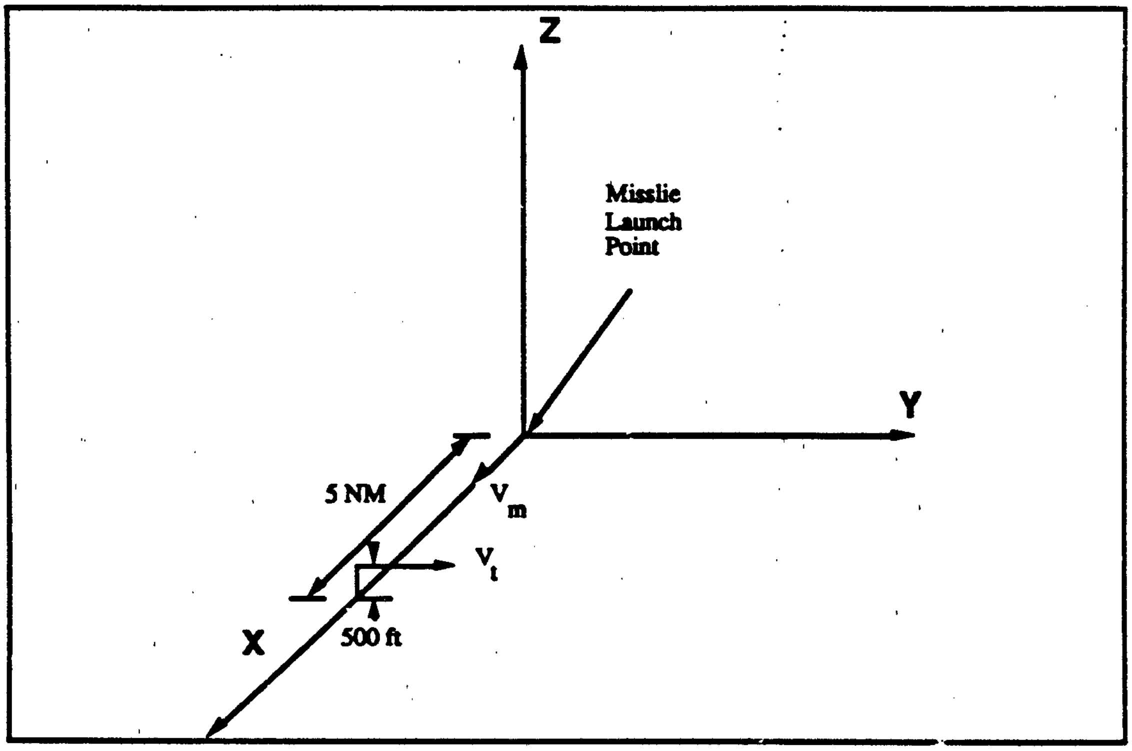


Figure 26. Constant Velocity Target Scenario Geometry

vmy(0) = 0 fps

vmz(0) = 0 fps

xt(0) = 30000 feet

yt(0) = 0 feet

zt(0) = 500 feet

vtx(0) = 0 fps

vty(0) = 1000 fps

vtz(0) = 0 fps.

Figure 27 depicts the scenario that reflects these initial conditions. In this case, the aircraft is attempting to stay on the outer edge of the missile envelope.

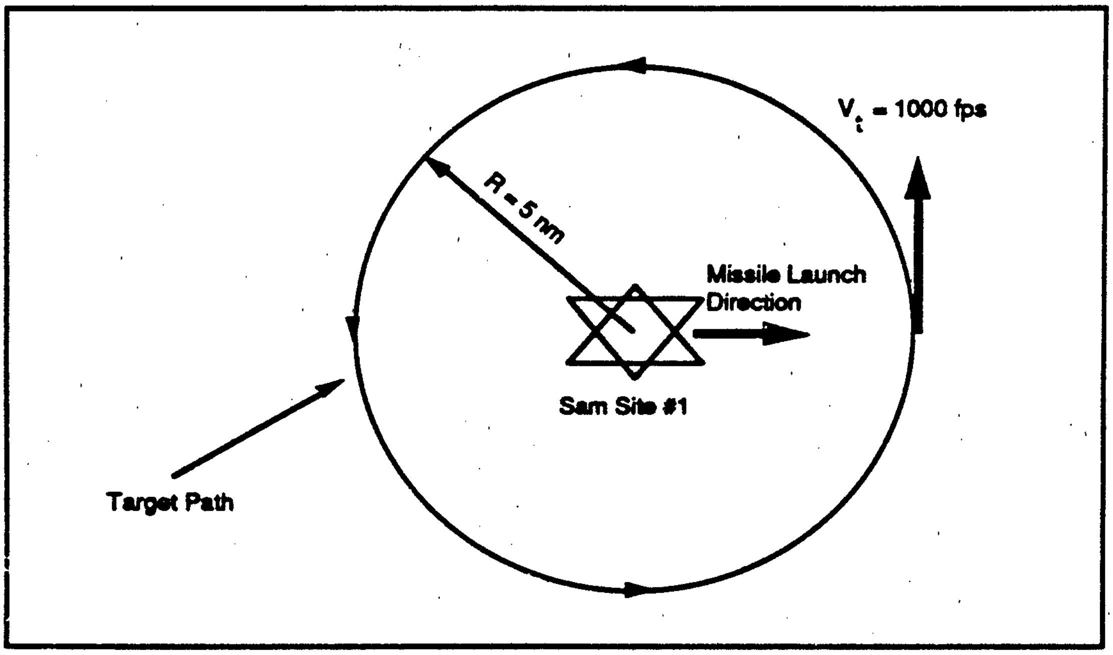


Figure 27. Tactical Scenario for Constant Velocity Target

2. Scenario 2: A Constant Acceleration Target

In this scenario, the target is accelerates at a constant rate of 15 feet/sec/sec in the y direction. Figure 28 shows the scenario geometry. The initial conditions are

xm(0) = 0 feet
ym(0) = 21000 feet
zm(0) = 0 feet
vmx(0) = 2100 fps
vmy(0) = -2100 fps
vmz(0) = 0 fps
xt(0) = 21000 feet
yt(0) = 0 feet
zt(0) = 500 feet

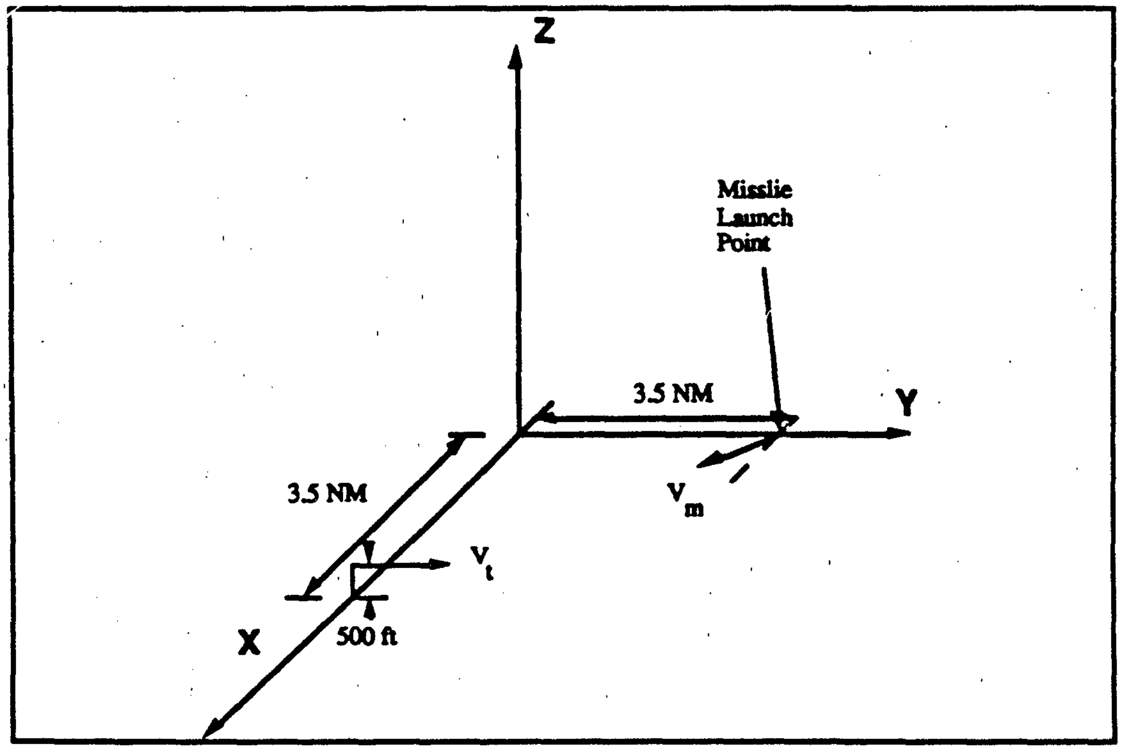


Figure 28. Constant Acceleration Target Scenario Geometry

vtx(0) = 0 fps

vty(0) = 1000 fps

vtz(0) = 0 fps.

xddt(0) = 0 fps^2

yddt(0) = 15 fps^2

zddt(0) = 0 fps^2

βd(0) = −45°.

Figure 29 depicts the scenario that reflects these initial conditions. In this case there is overlapping missile envelope coverage and the aircraft is attempting to strike a target located within this region. The missile is launched when the target enters the 5NM radius.

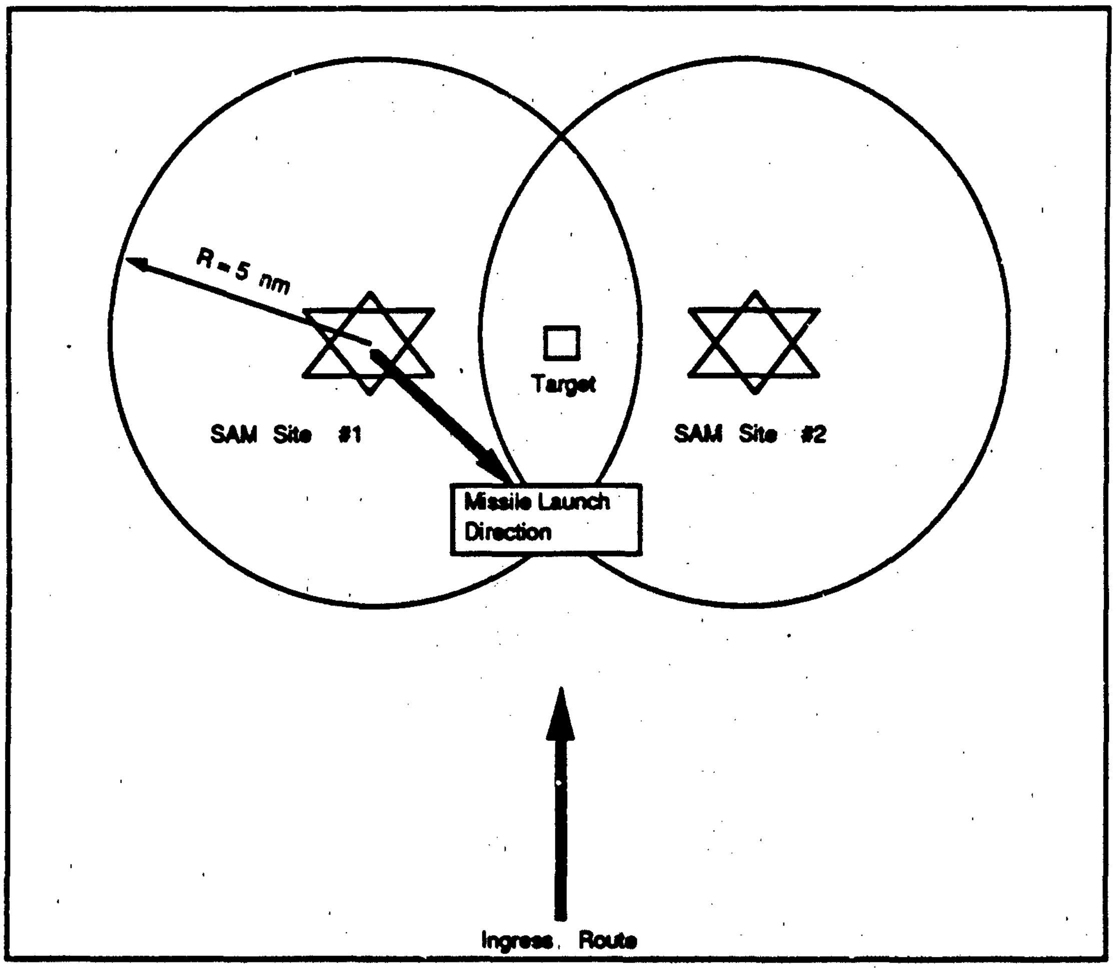


Figure 29. Tactical Scenario for Constant Acceleration Target

3. Scenario 3: A Two Dimensional Target Acceleration

Figures 26 and 27 depict the initial conditions for this scenario. The target performs a 6.5 g level turn toward the missile at a range of 12,000 feet. The target acceleration is

$$\ddot{x}_t = -6.5 \cdot 32.2 \cdot \sin\left(\gamma_{t\_yaw}\right) \tag{4.1}$$

$$\ddot{y}_t = 6.5 \cdot 32.2 \cdot \cos\left(\gamma_{t\_yaw}\right) \tag{4.2}$$

$$\ddot{z}_t = 0.0. \tag{4.3}$$

4. Scenario 4: A Three Dimensional Target Acceleration

Figures 28 and 29 depict the initial conditions for this scenario. The target performs a three dimensional maneuver toward the missile and into the vertical at a range of 12,000 feet. The target acceleration is

$$\ddot{x}_t = -6.5 \cdot 32.2 \cdot \sin\left(\gamma_{t\_yaw}\right) \tag{4.4}$$

$$\ddot{y}_t = 6.5 \cdot 32.2 \cdot \cos\left(\gamma_{t\_yaw}\right) \tag{4.5}$$

$$\ddot{z}_t = 6.5 \cdot 32.2 \cdot \cos\left(\gamma_{t\_pitch}\right). \tag{4.6}$$

5. Three Dimensional Simulation Results

Figures 30-75 display the results of the three dimensional simulations. Figures 30-40 relate to the constant velocity target scenario. Figures 41-52 relate to the constant acceleration scenario. Figures 53-64 relate to the two dimensional target acceleration scenario. Figures 65-75 relate to the three dimensional target acceleration scenario.

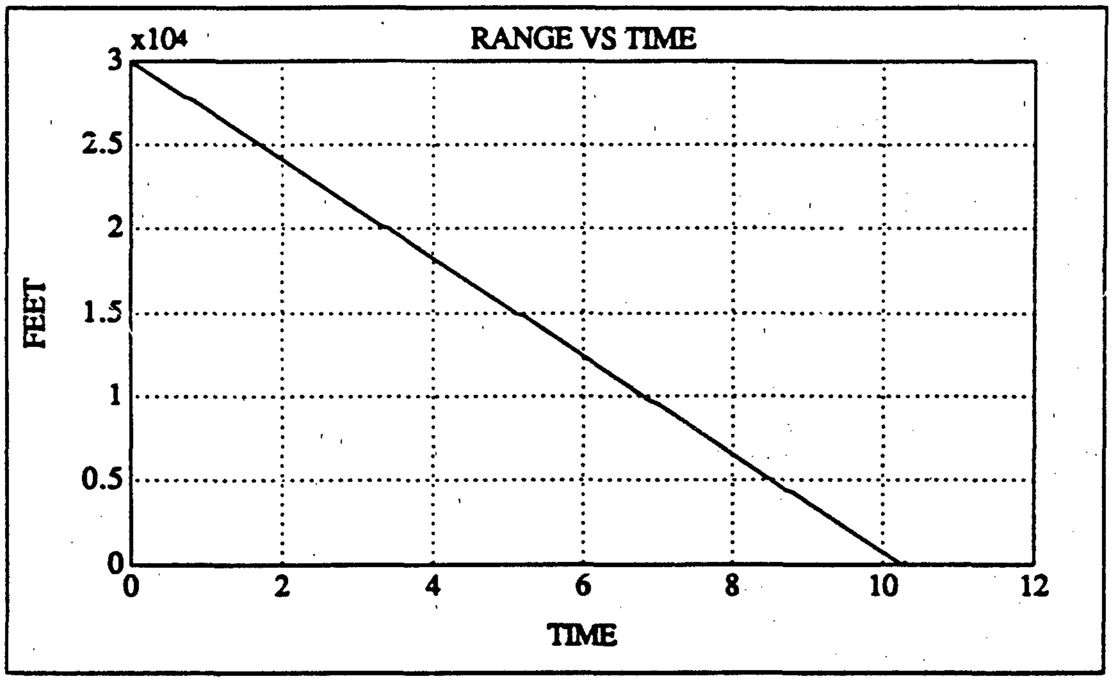


Figure 30. Scenario 1: Missile to Target Range

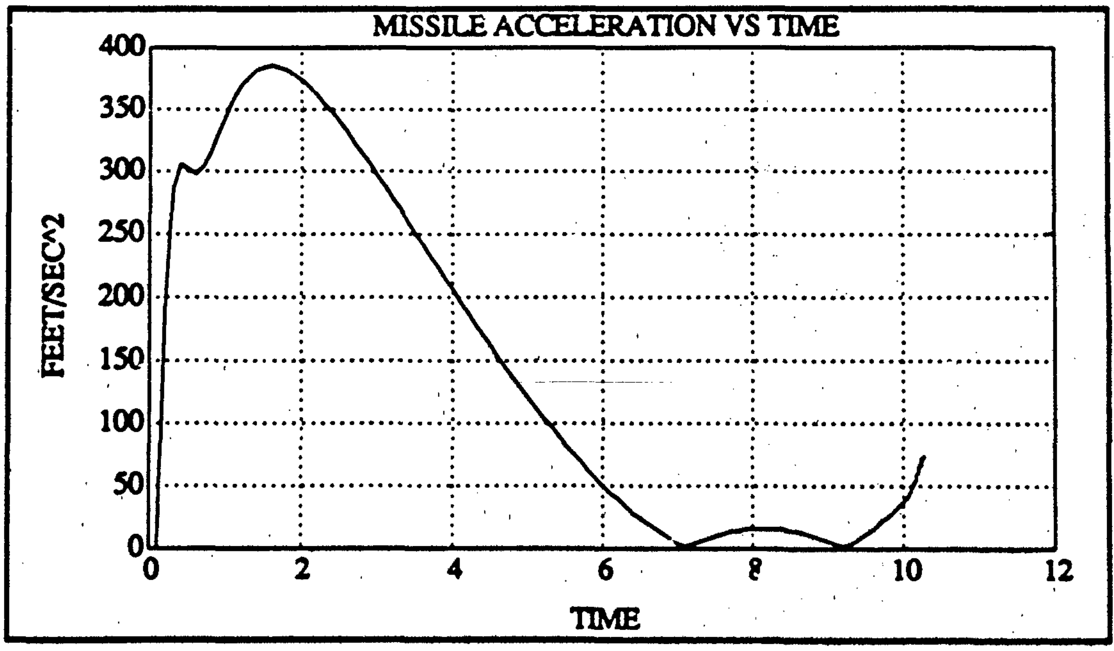


Figure 31. Scenario 1: Missile Acceleration $|A_m|$

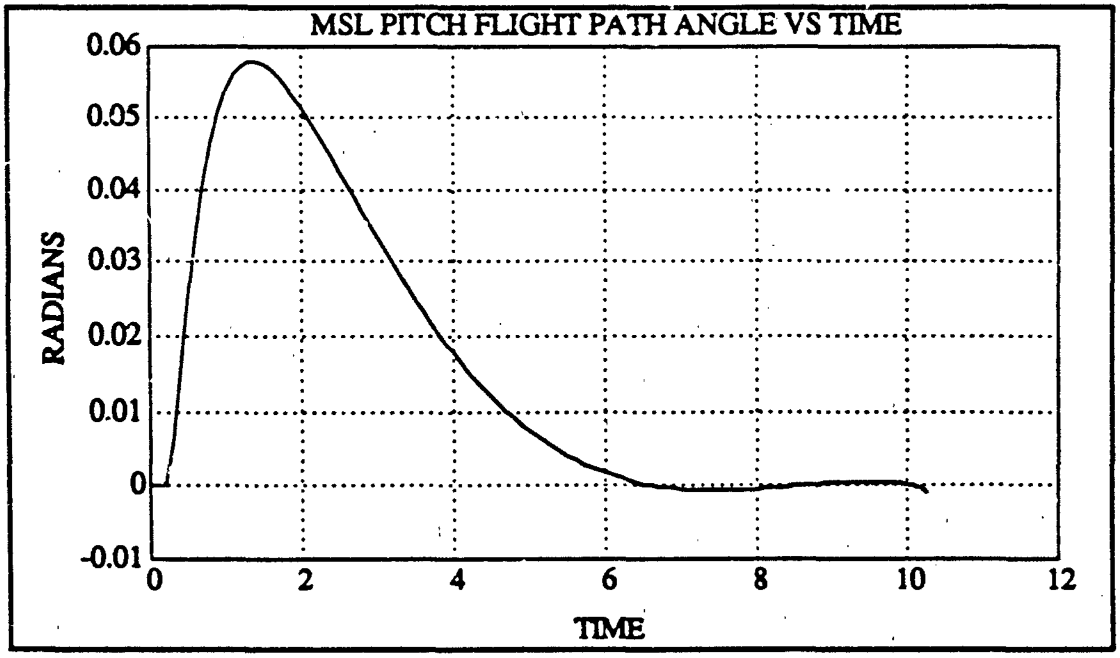


Figure 32. Scenario 1: Missile Pitch Flight Path Angle γ_{m_pitch}

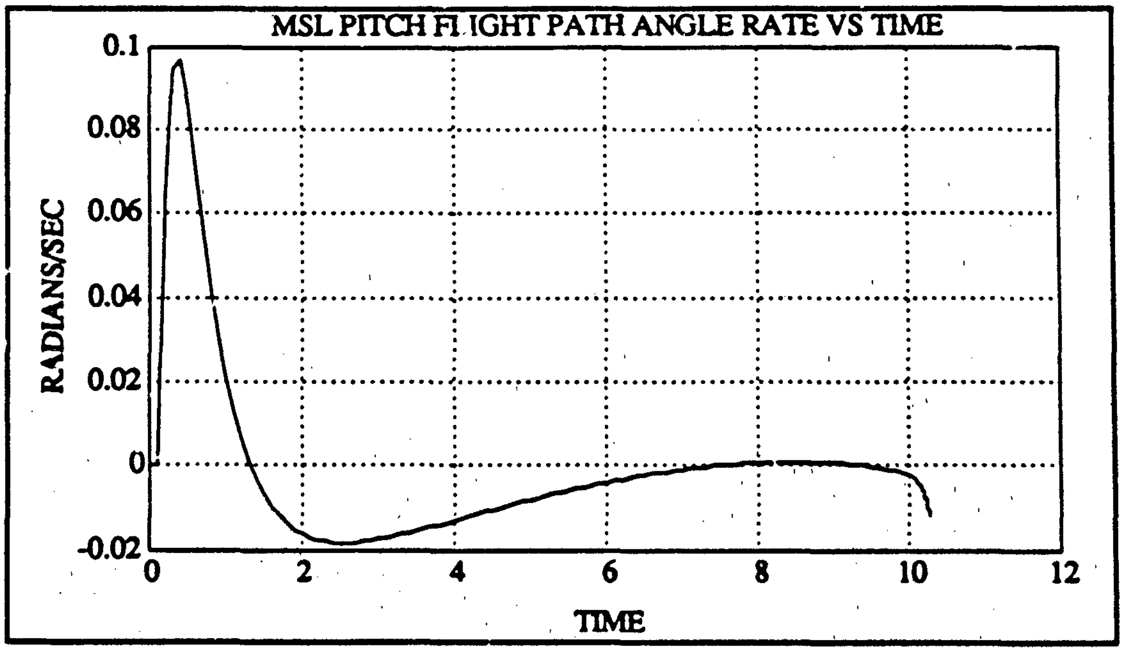


Figure 33. Scenario 1: Missile Pitch Flight Path Angle Rate $\dot{\gamma}_{m\_pitch}$

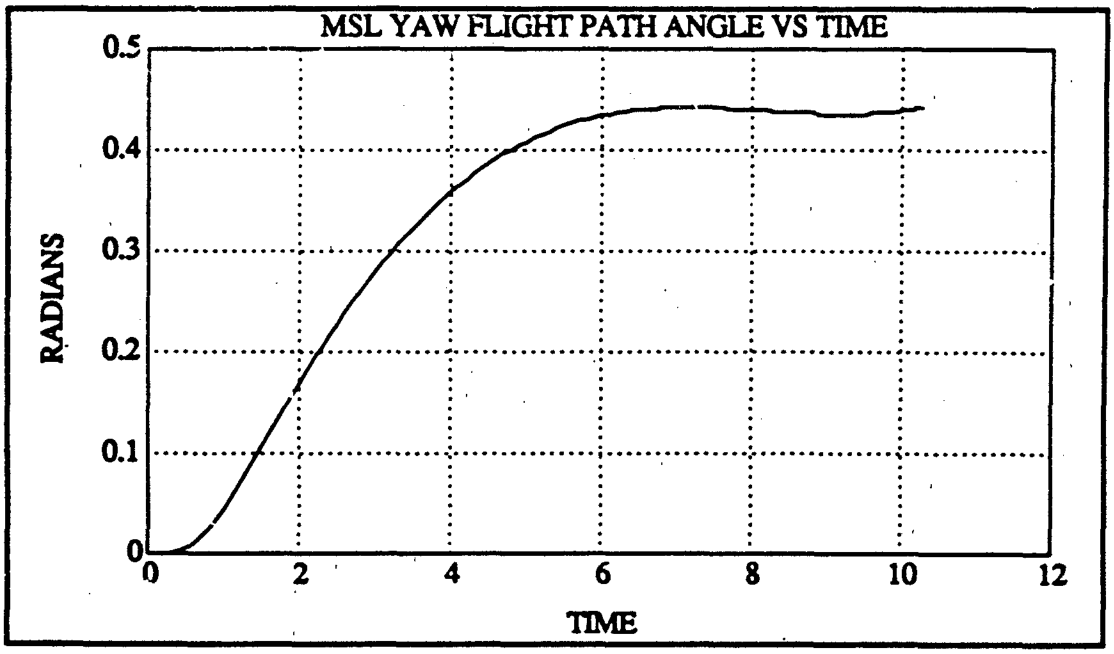


Figure 34. Scenario 1: Missile Yaw Flight Path Angle γ_{m_yaw}

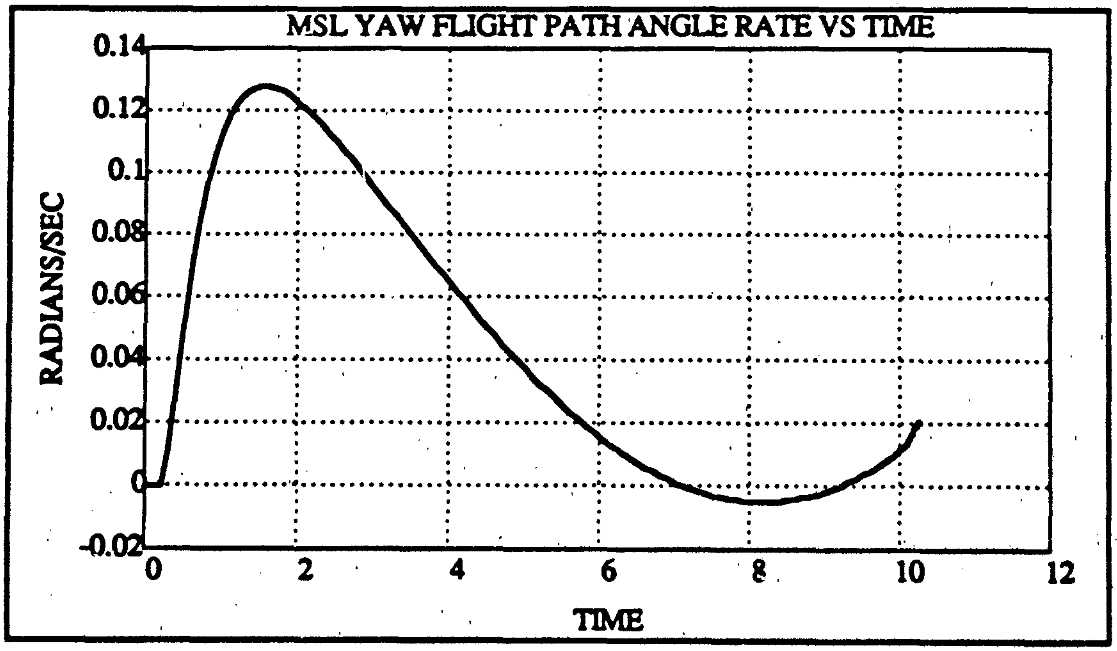


Figure 35. Scenario 1: Missile Yaw Flight Path Angle Rate $\dot{\gamma}_{m\_yaw}$

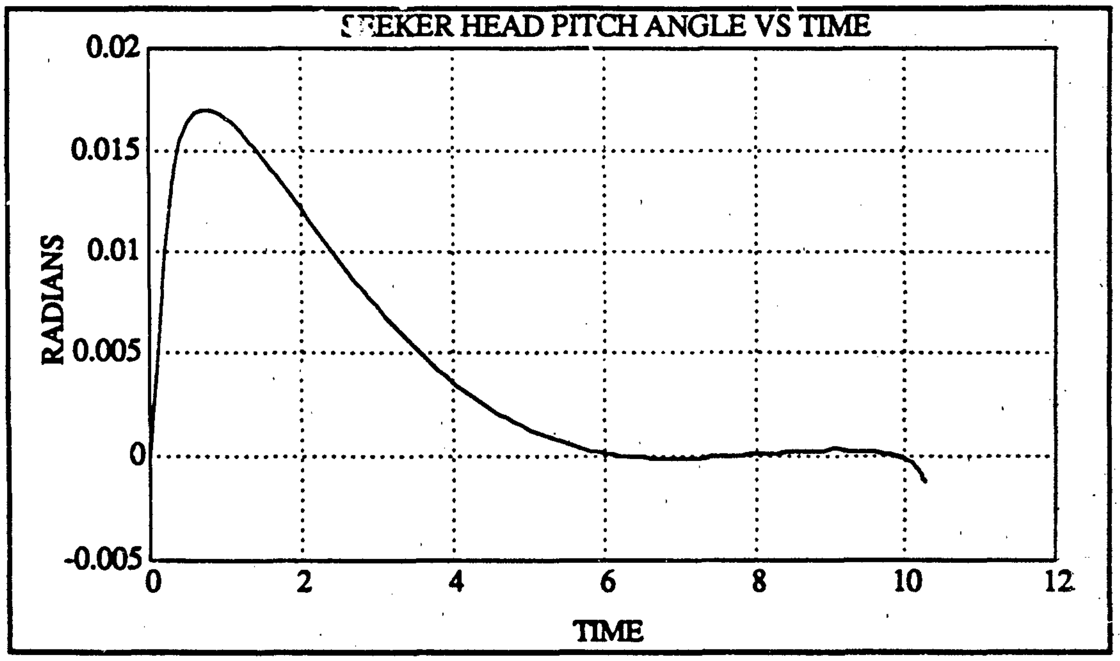


Figure 36. Scenario 1: Seeker Head Pitch Angle β_{pitch}

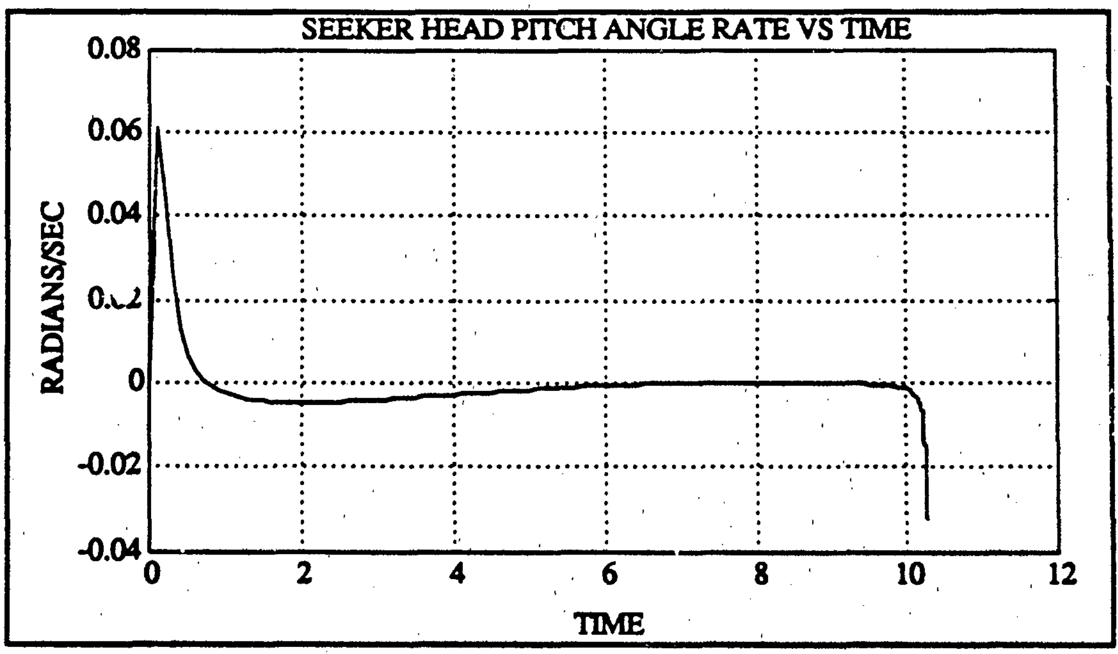


Figure 37. Scenario 1: Seeker Head Pitch Angle Rate $\dot{\beta}_{pitch}$

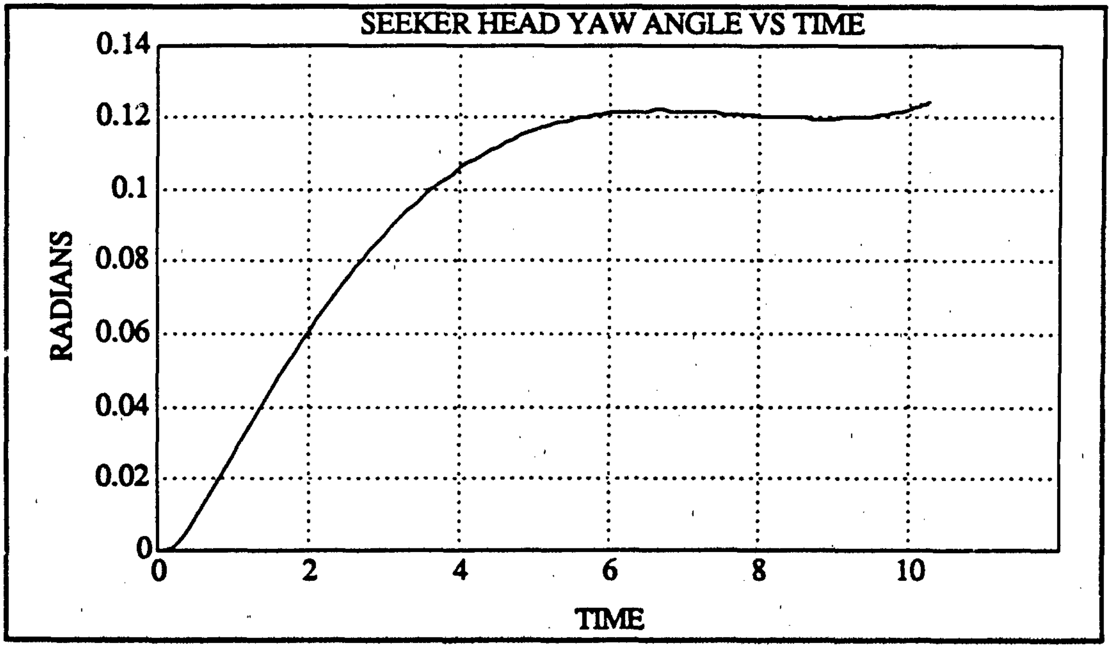


Figure 38. Scenario 1: Seeker Head Yaw Angle β_{yaw}

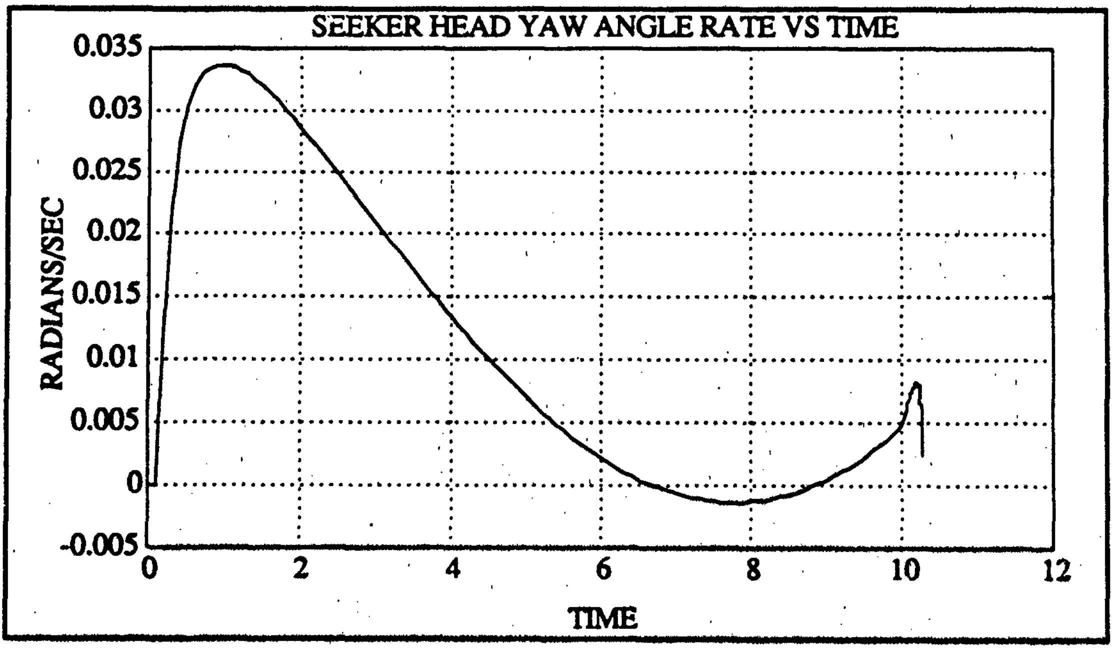


Figure 39. Scenario 1: Seeker Head Yaw Angle Rate $\dot{\beta}_{yaw}$

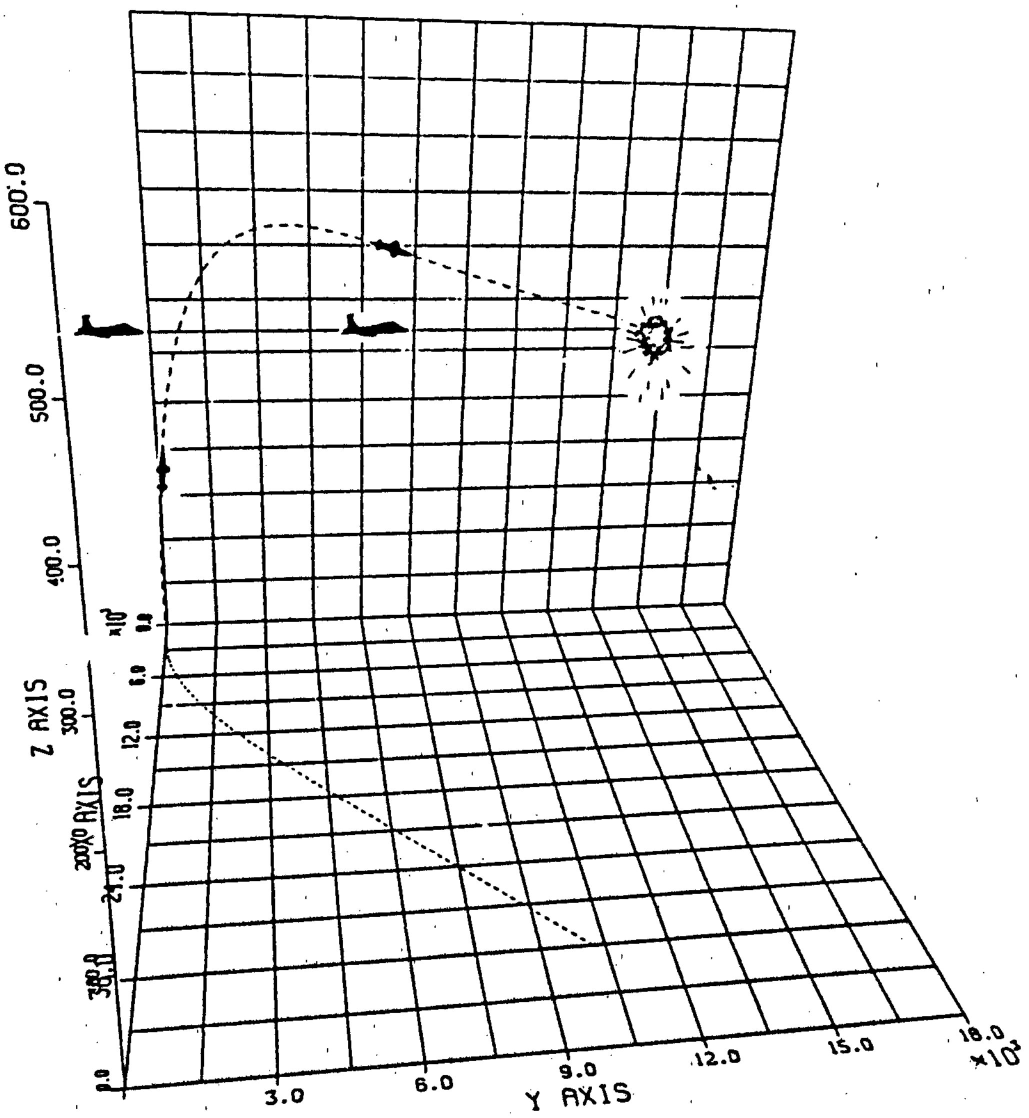


Figure 40 Scenario 1: Three Dimensional Plot

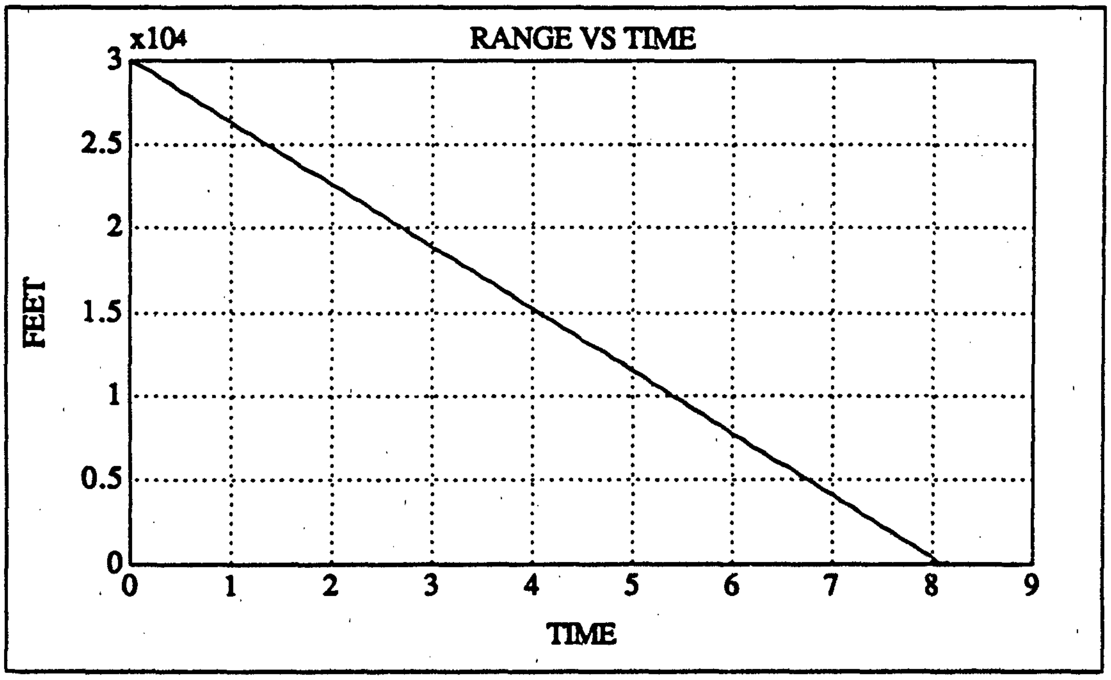


Figure 41. Scenario 2: Missile to Target Range

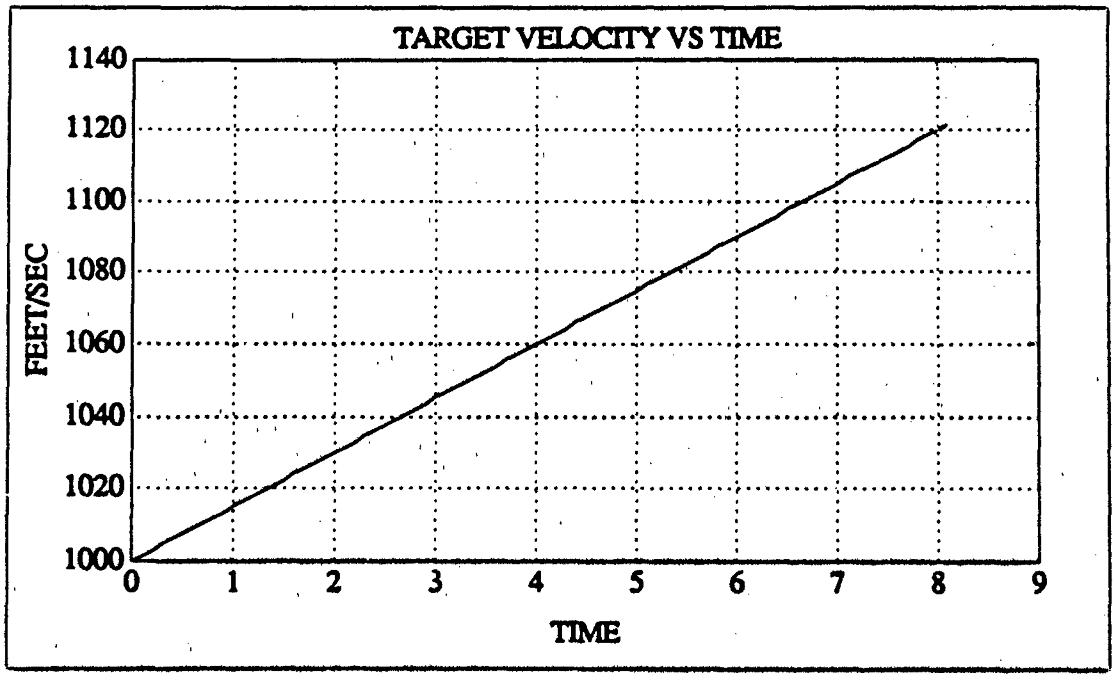


Figure 42. Scenario 2: Target Velocity V_t

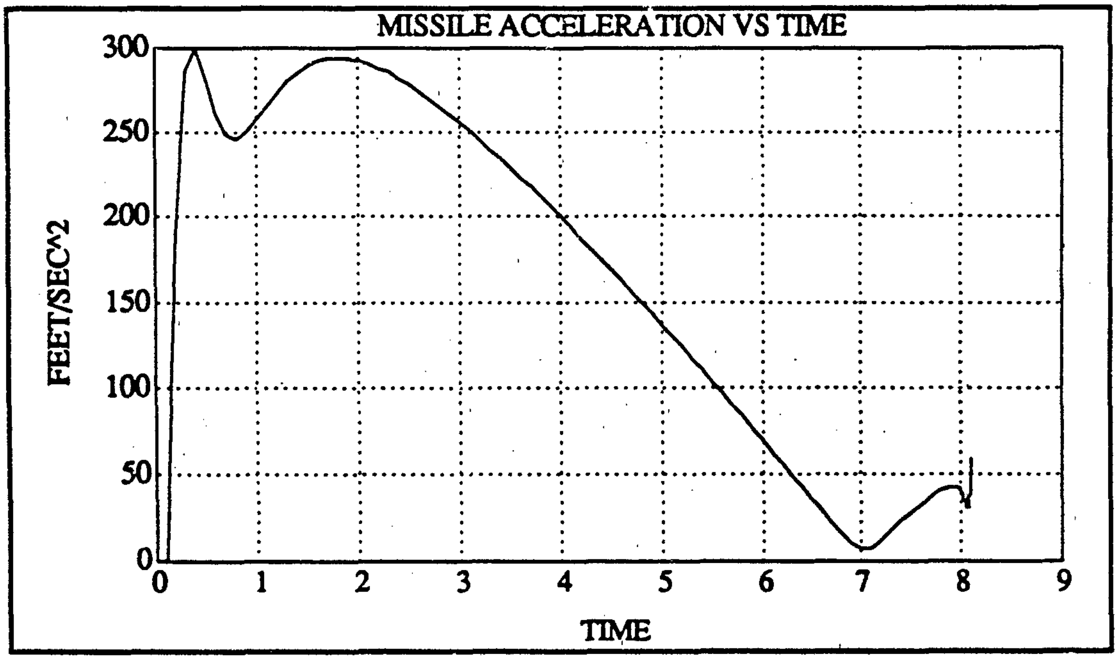


Figure 43. Scenario 2: Missile Acceleration $|A_m|$

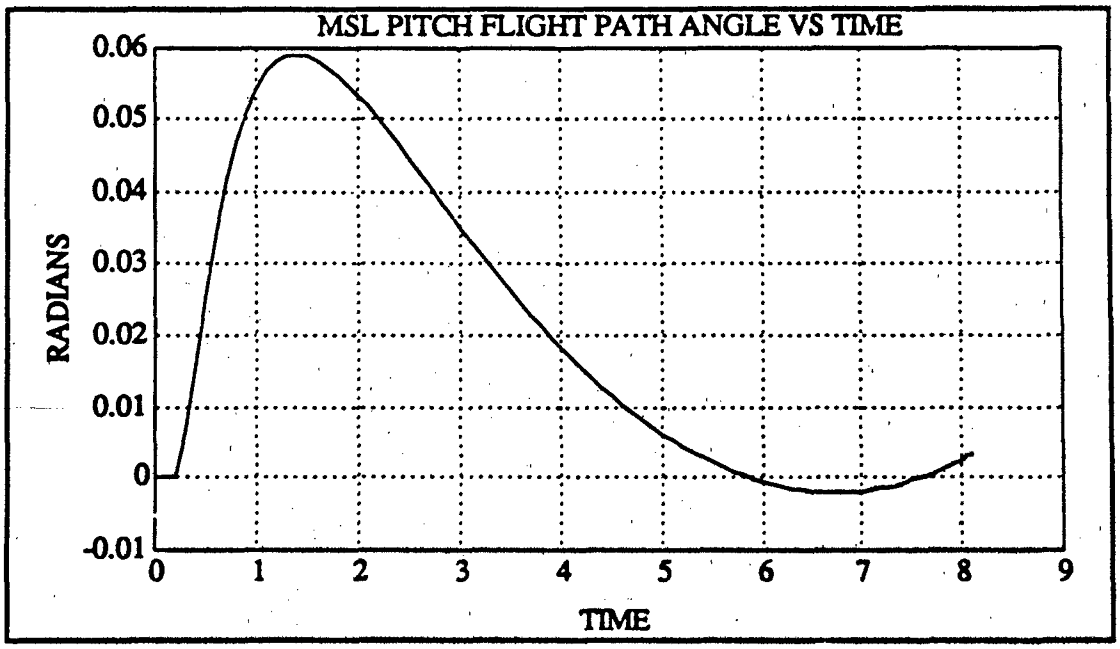


Figure 44. Scenario 2: Missile Pitch Flight Path Angle γ_{m_pitch}

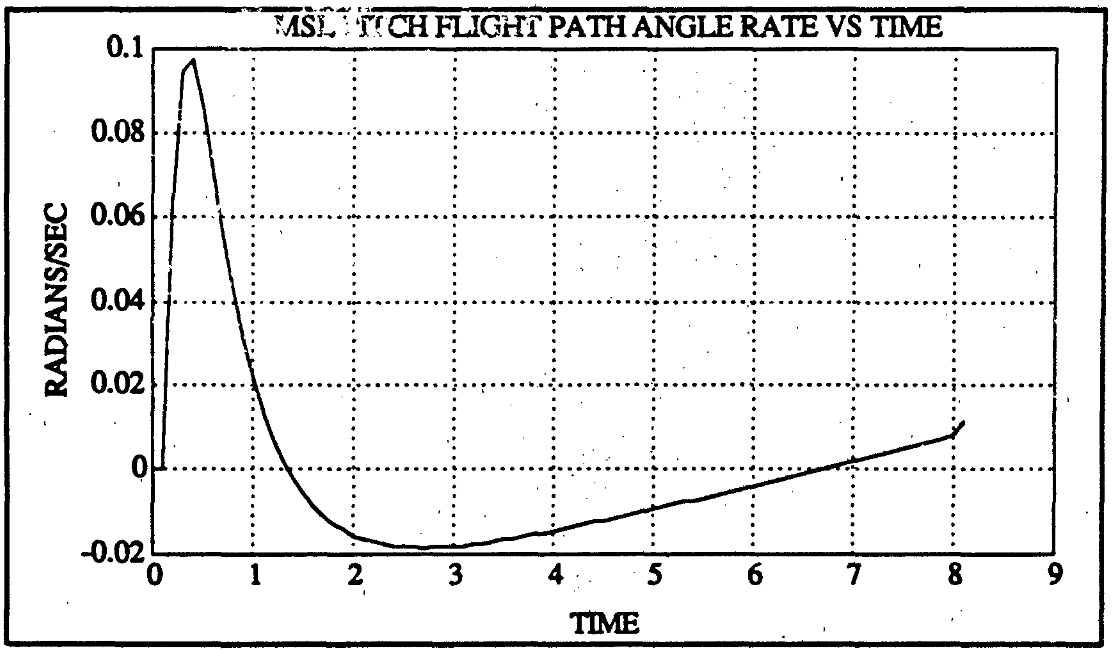


Figure 45. Scenario 2: Missile Pitch Flight Path Angle Rate $\dot{\gamma}_{m\_pitch}$

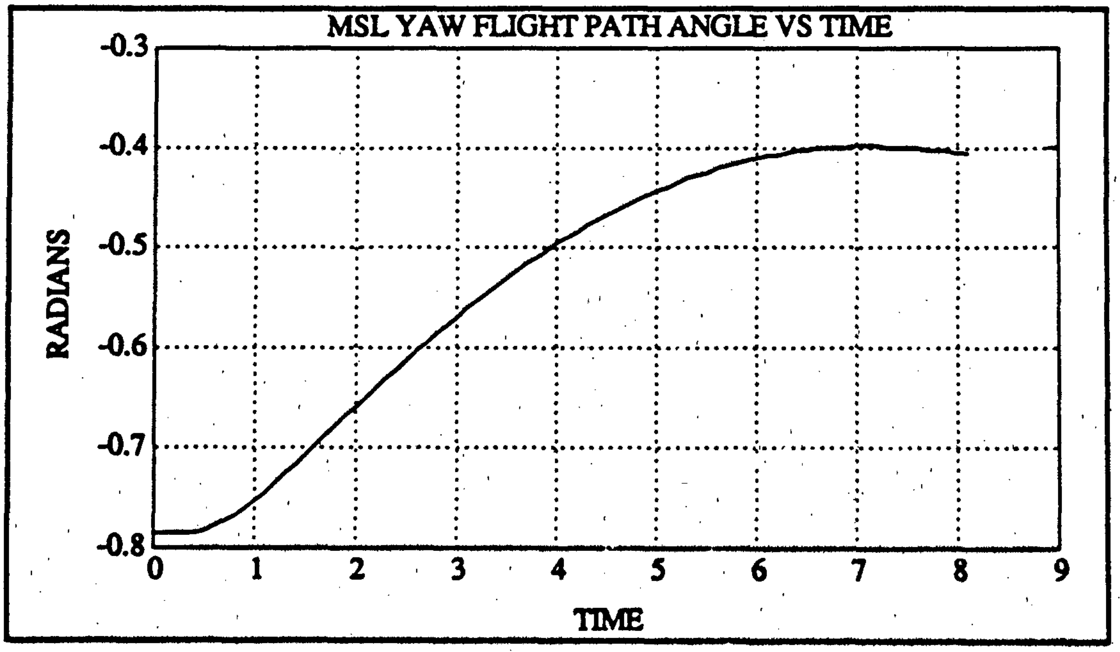


Figure 46. Scenario 2: Missile Yaw Flight Path Angle γ_{m_yaw}

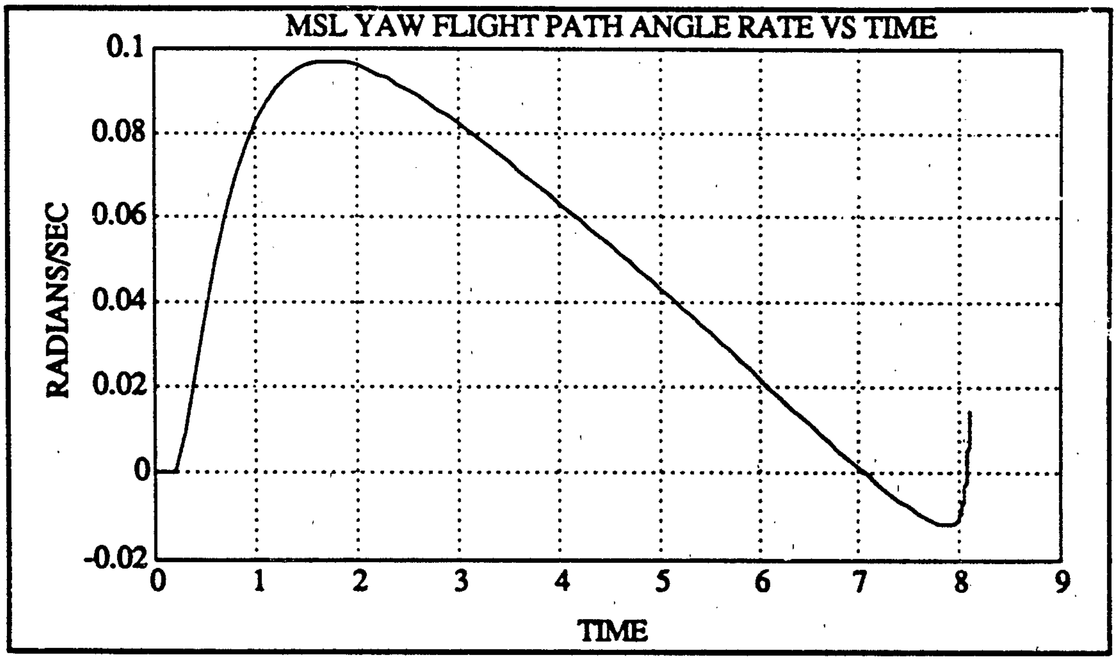


Figure 47. Scenario 2: Missile Yaw Flight Path Angle Rate $\dot{\gamma}_{m\_yaw}$

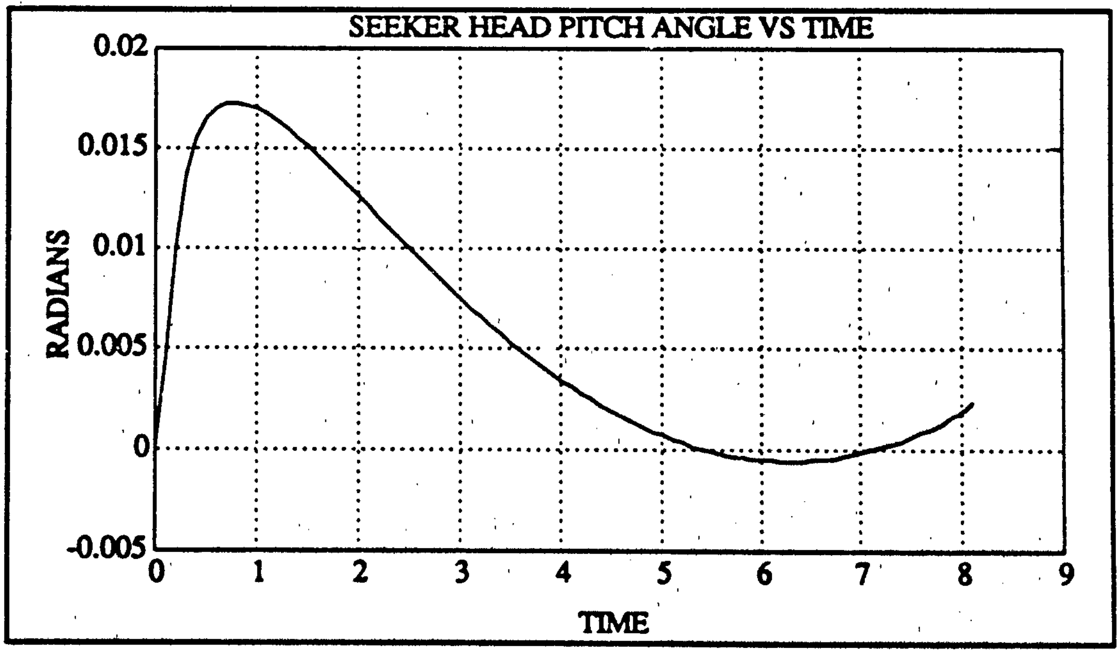


Figure 48. Scenario 2: Seeker Head Pitch Angle β_{pitch}

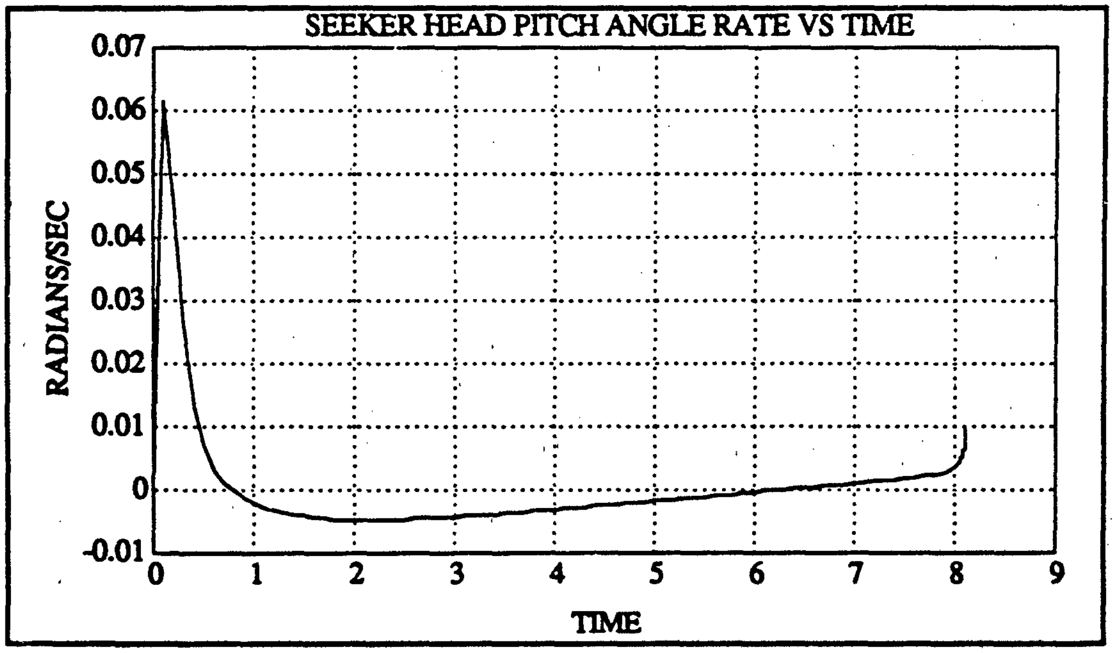


Figure 49. Scenario 2: Seeker Head Pitch Angle Rate $\dot{\beta}_{pitch}$

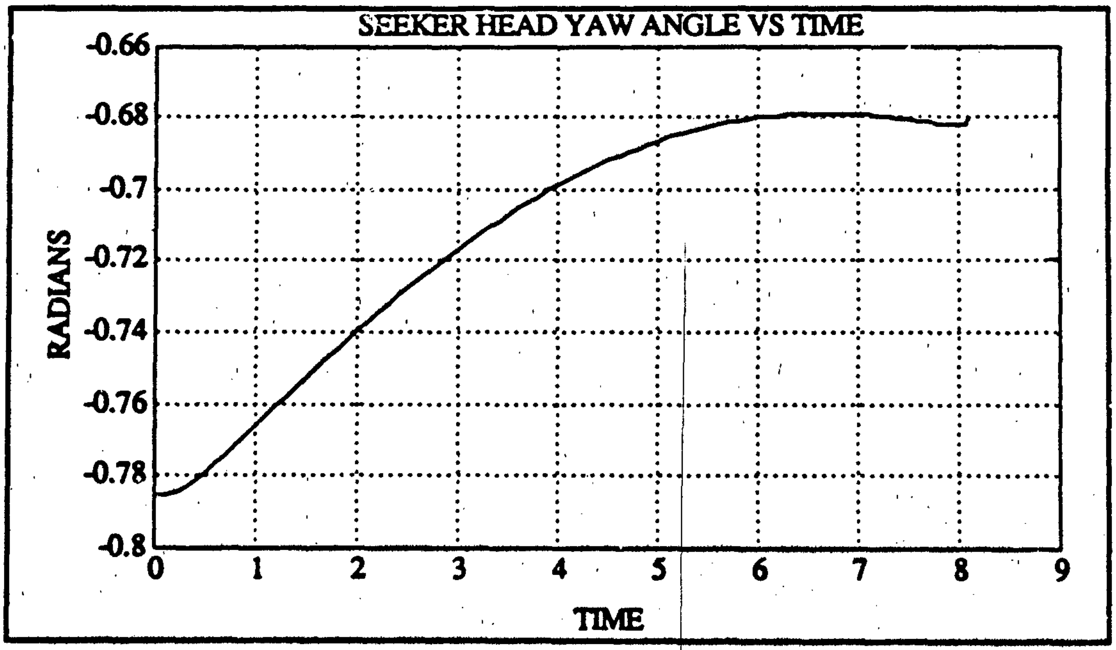


Figure 50. Scenario 2: Seeker Head Yaw Angle β_{yaw}

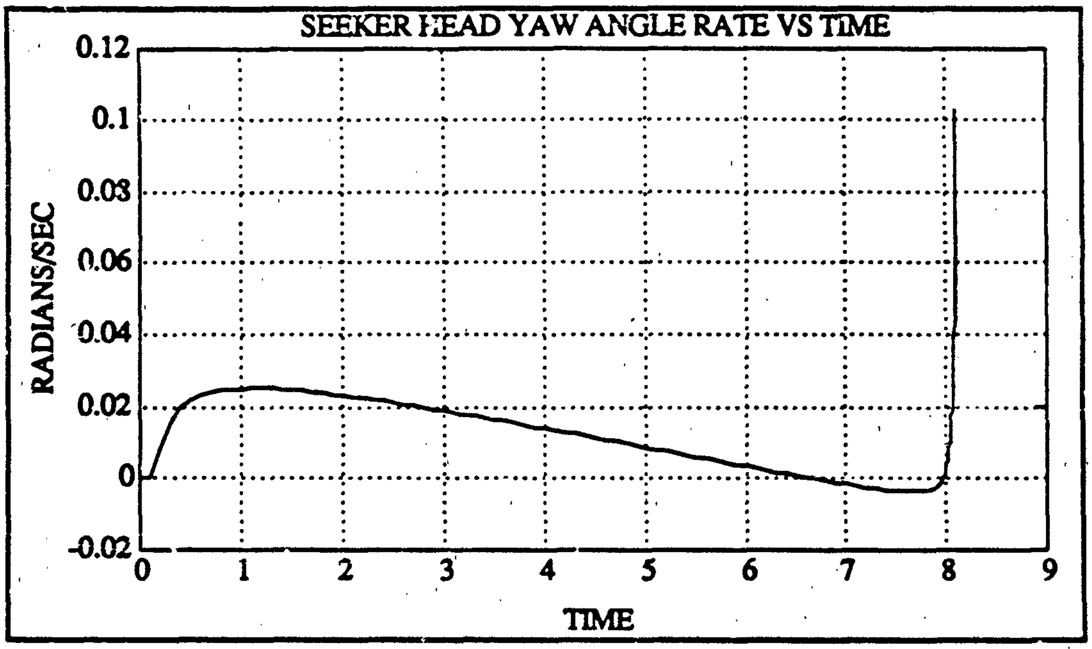


Figure 51. Scenario 2: Seeker Head Yaw Angle Rate $\dot{\beta}_{yaw}$

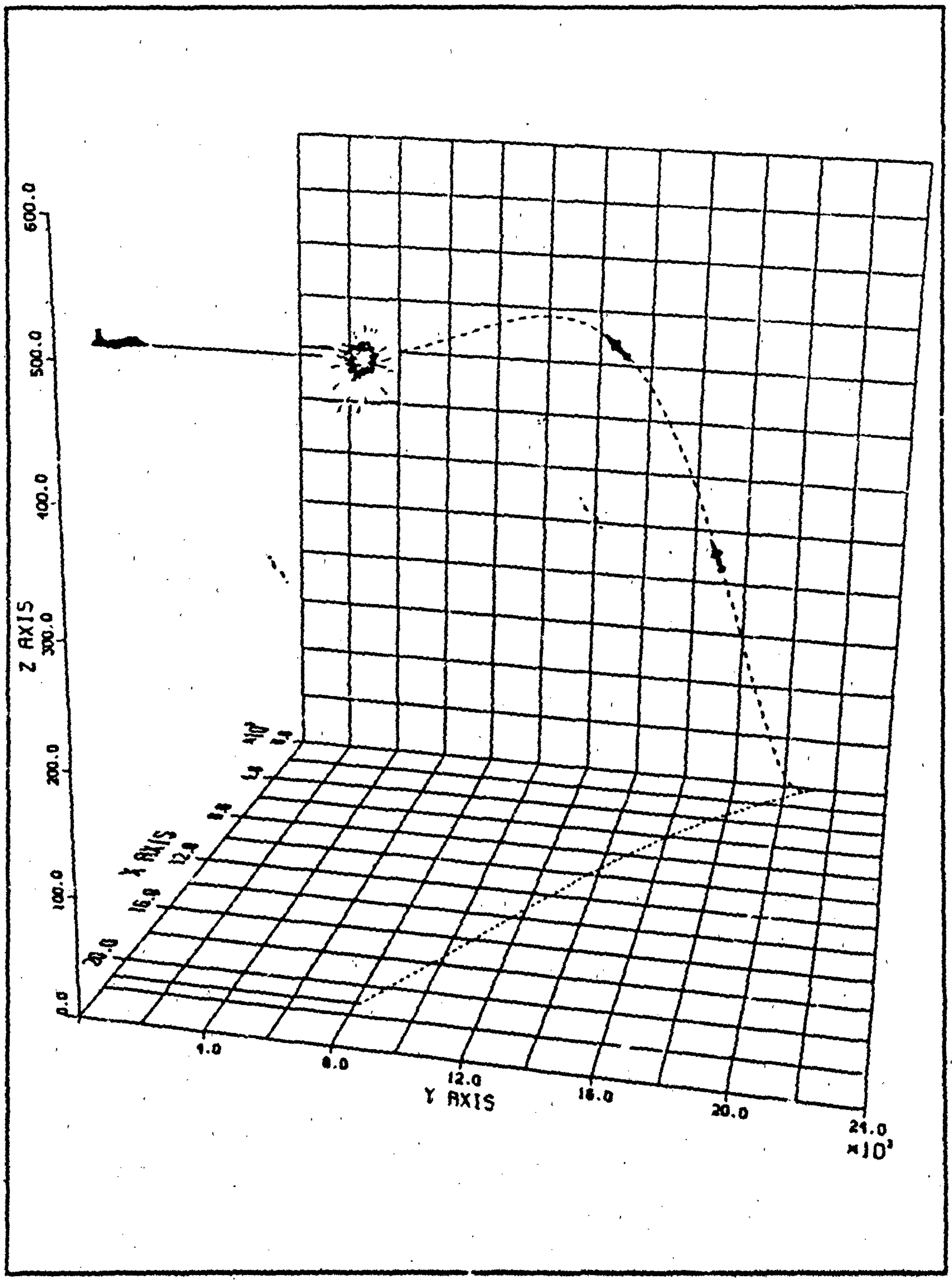


Figure 52. Scenario 2: Three Dimensional Plot

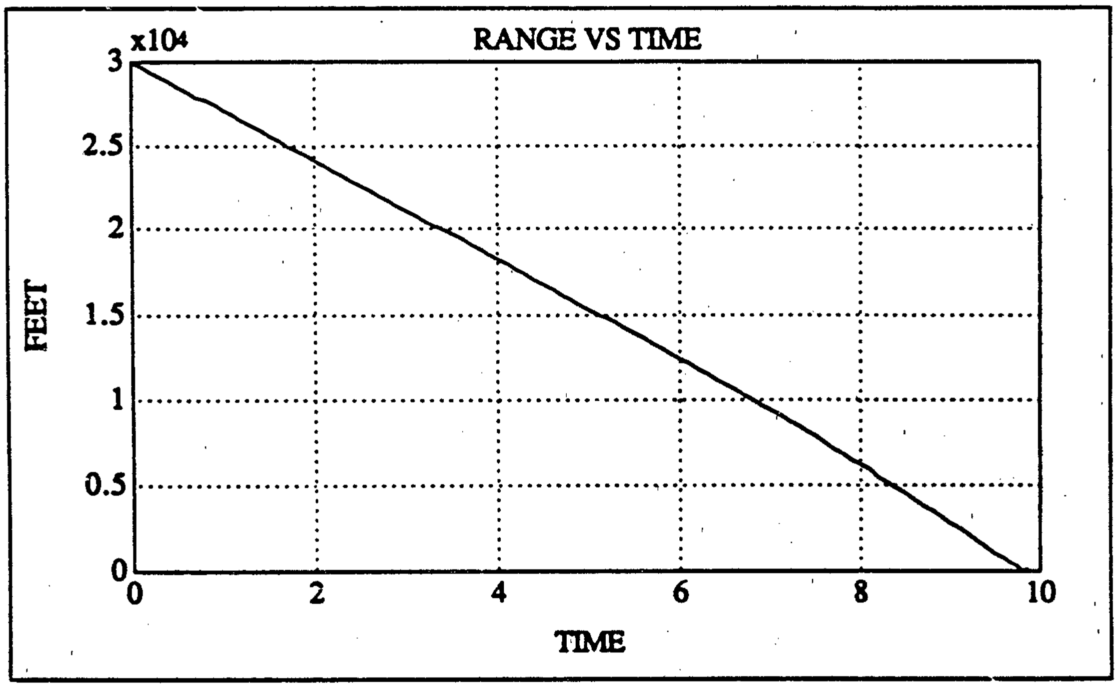


Figure 53. Scenario 3: Missile to Target Range

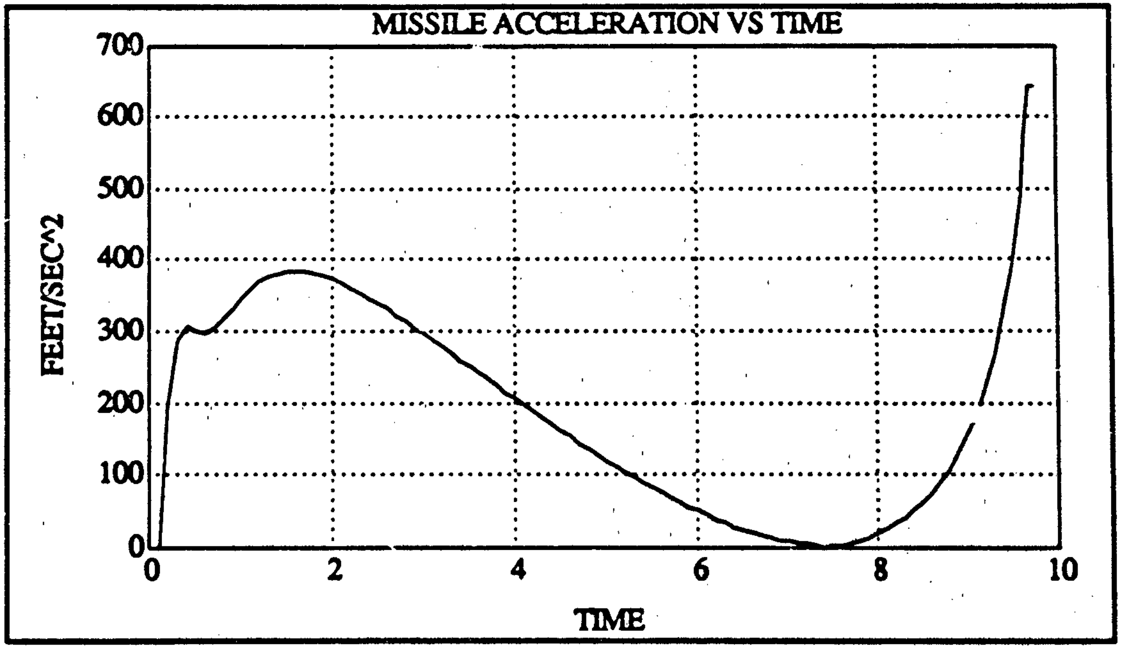


Figure 54. Scenario 3: Missile Acceleration $|A_m|$

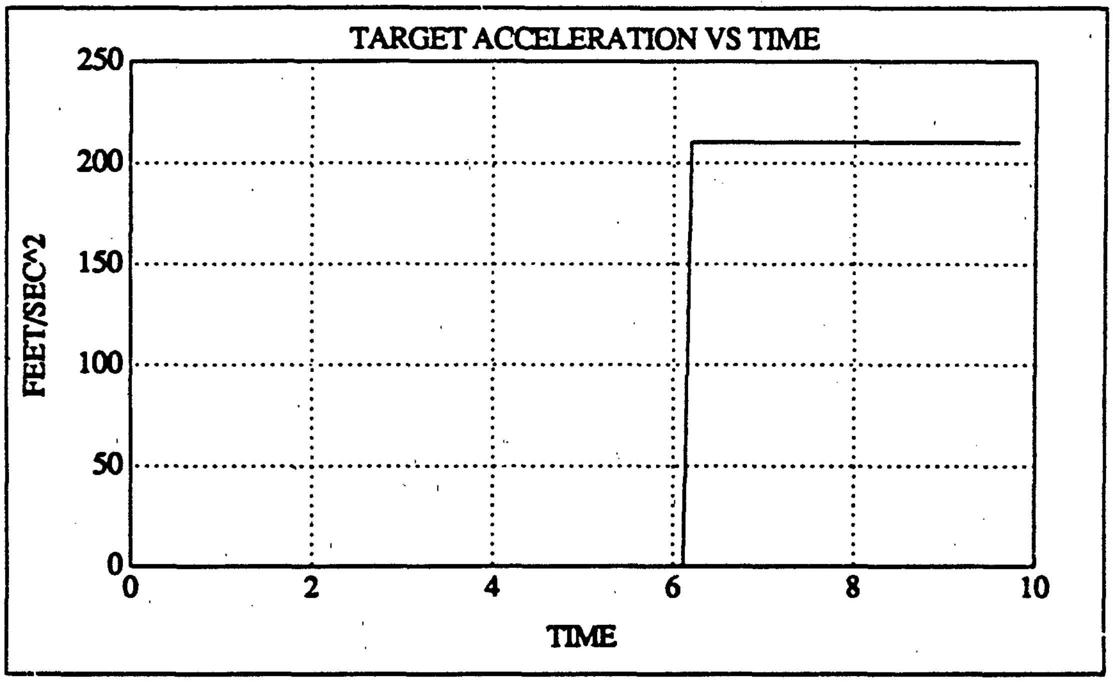


Figure 55. Scenario 3: Target Acceleration $|A_t|$

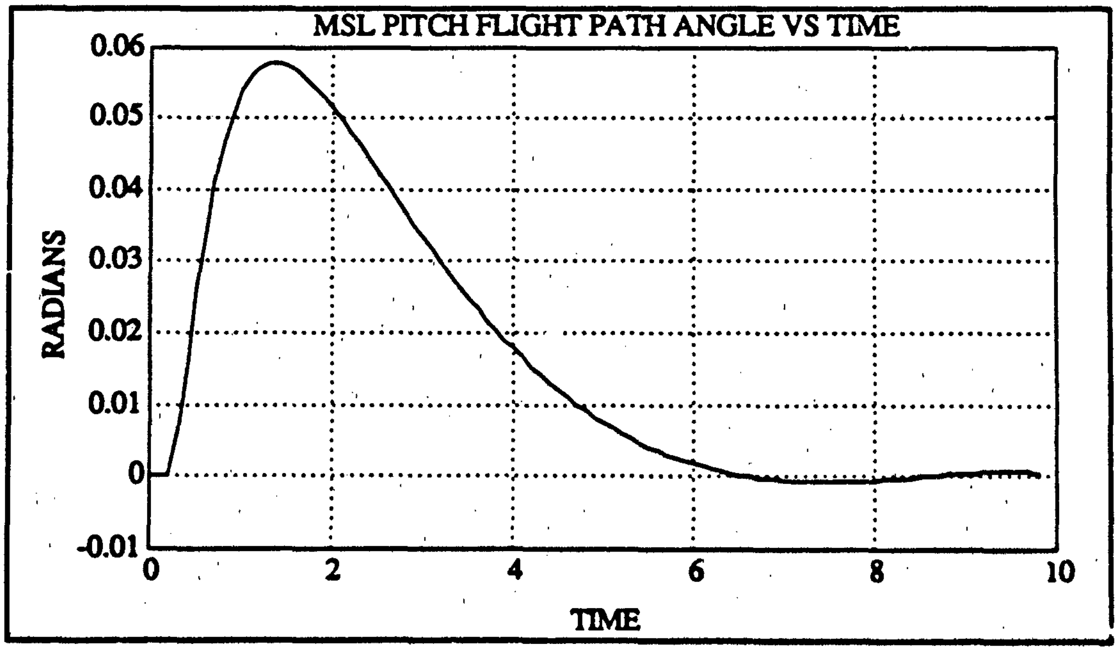


Figure 56. Scenario 3: Missile Pitch Flight Path Angle γ_{m_pitch}

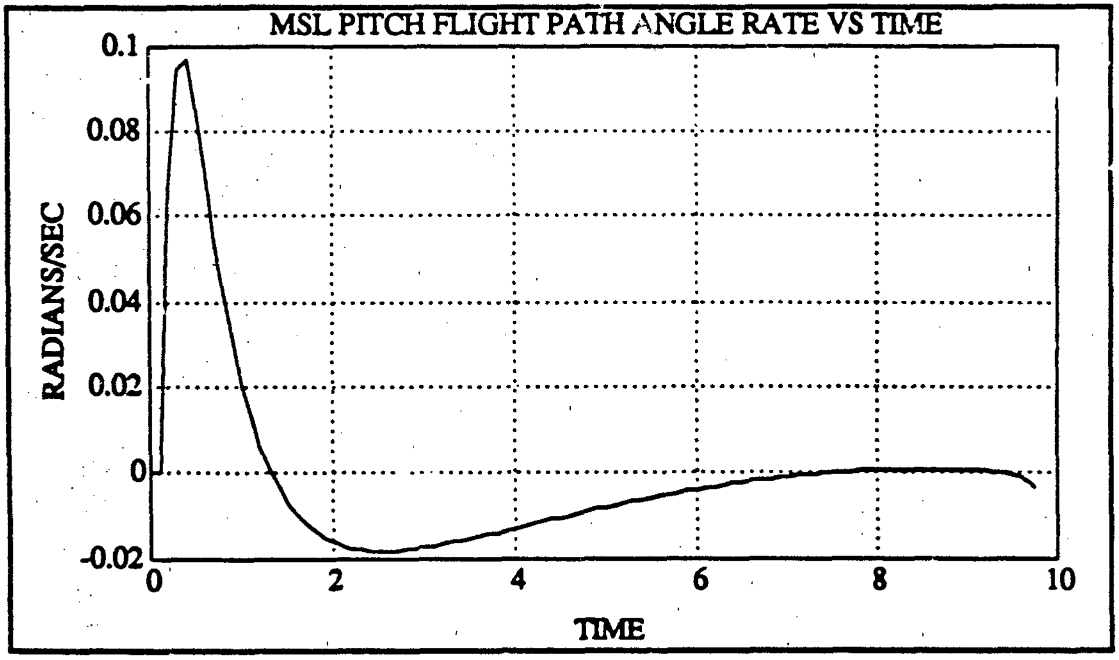


Figure 57. Scenario 3: Missile Pitch Flight Path Angle Rate $\dot{\gamma}_{m\_pitch}$

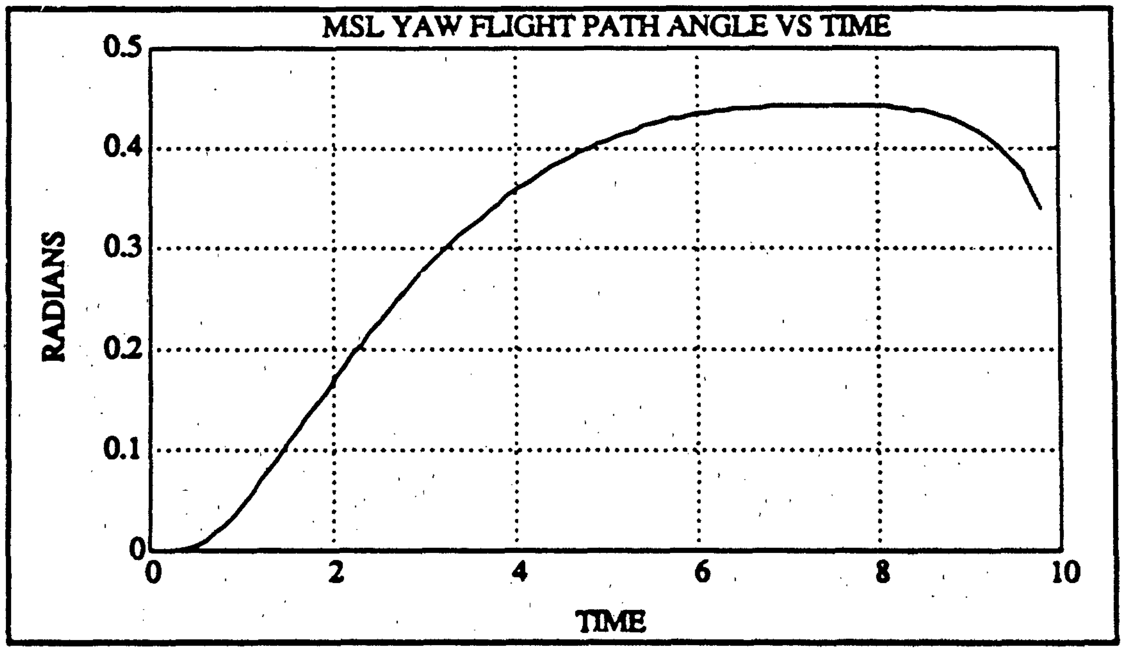


Figure 58. Scenario 3: Missile Yaw Flight Path Angle γ_{m_yaw}

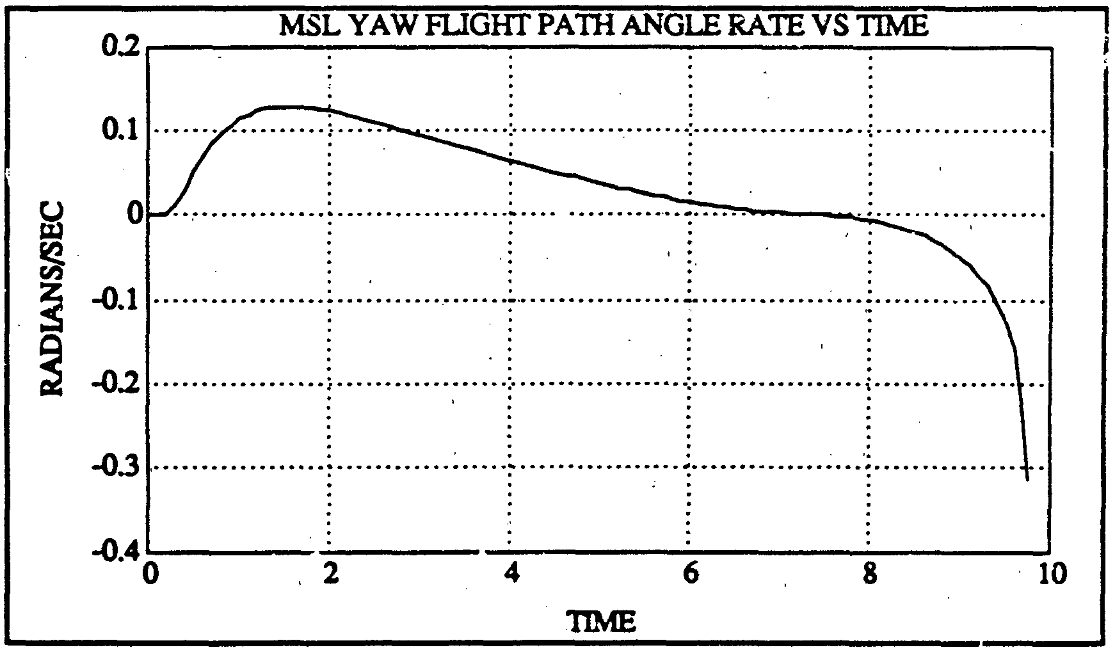


Figure 59. Scenario 3: Missile Yaw Flight Path Angle Rate $\dot{\gamma}_{m\_yaw}$

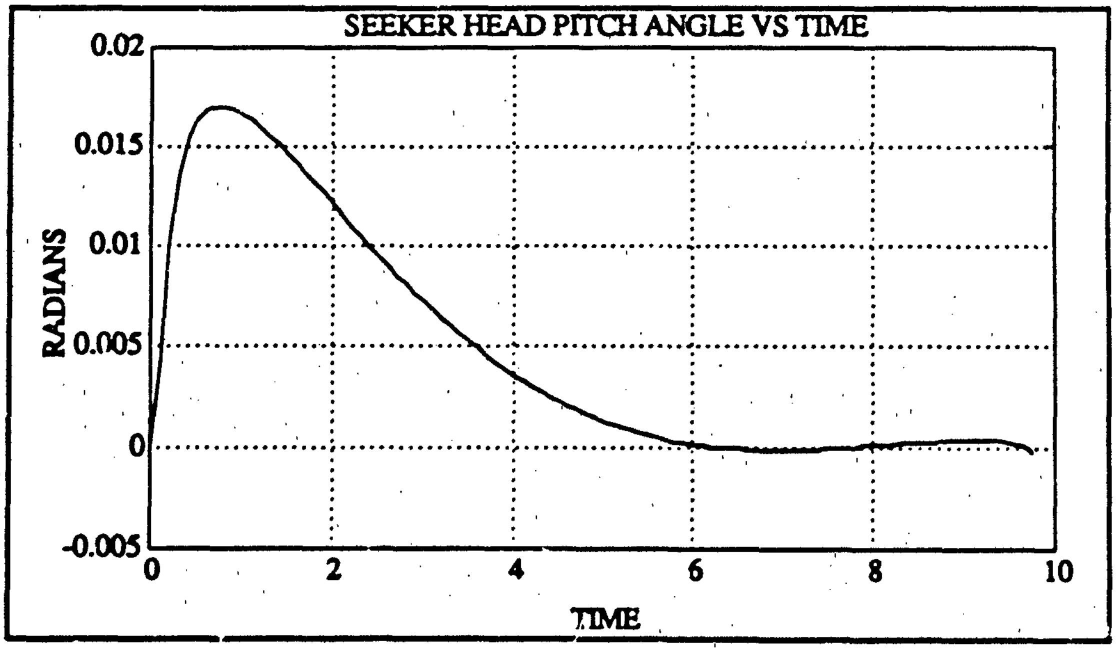


Figure 60. Scenario 3: S•ekerHead Pitch Angle β_{pitch}

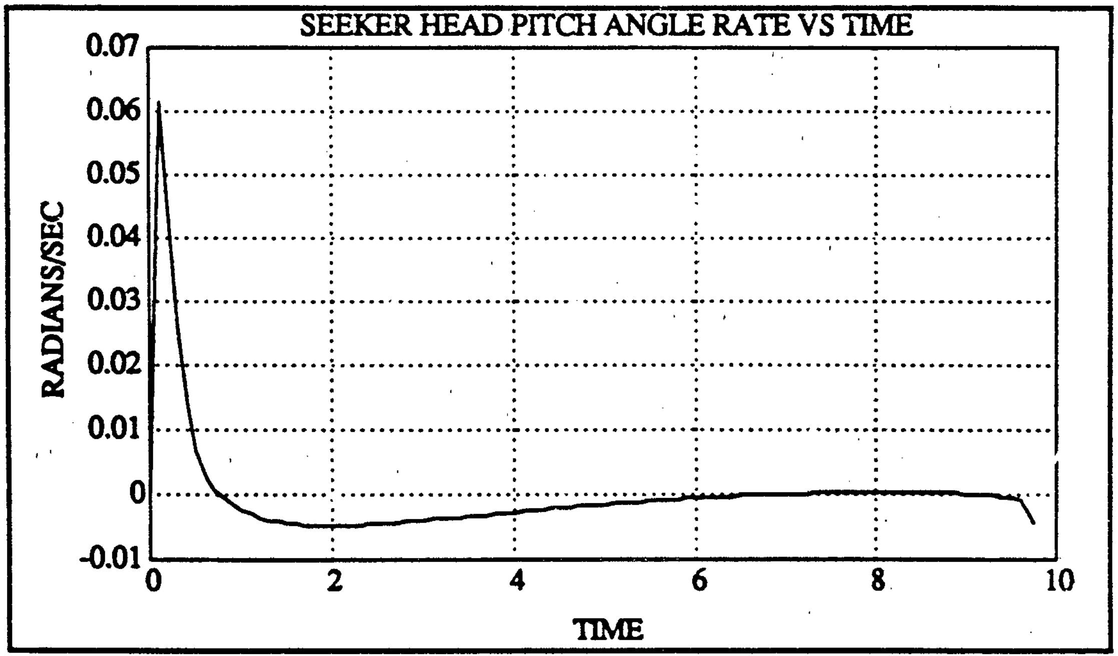


Figure 61. Scenario 3: Seeker Head Pitch Angle Rate $\dot{\beta}_{pitch}$

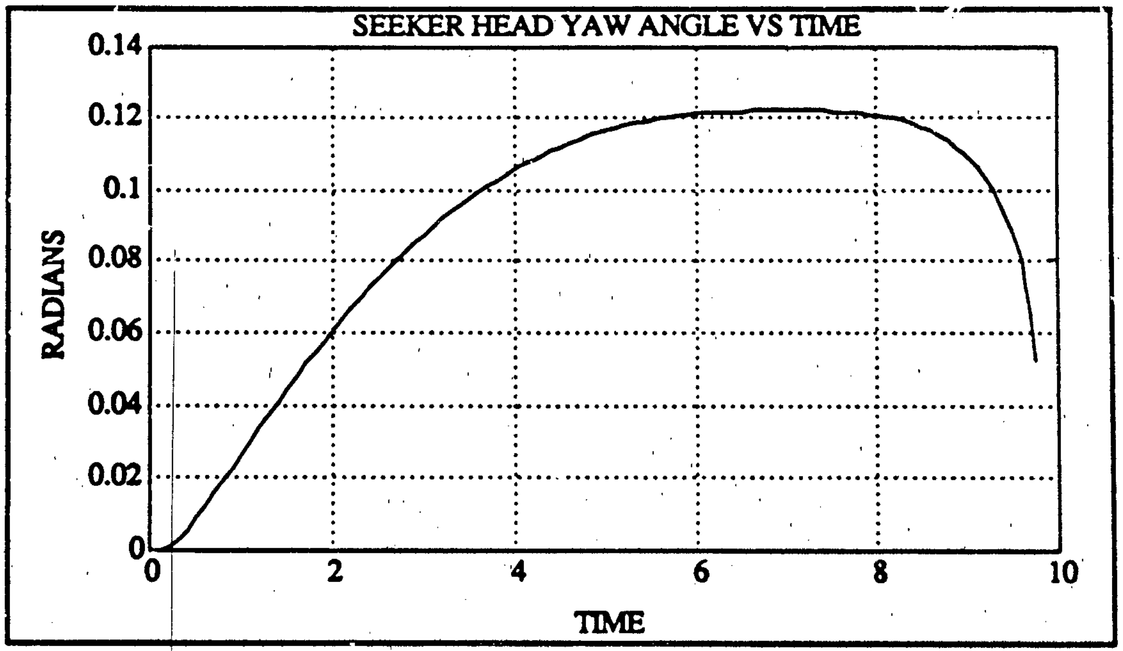


Figure 62. Scenario 3: Seeker Head Yaw Angle β_{yaw}

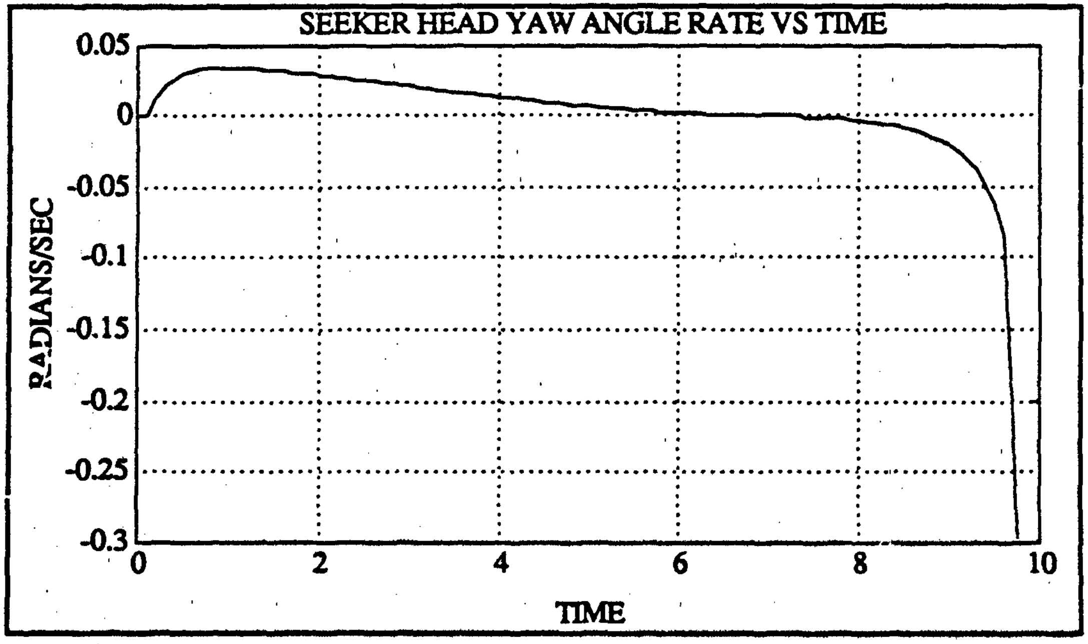


Figure 63. Scenario 3: Seeker Head Yaw Angle Rate $\dot{\beta}_{yaw}$

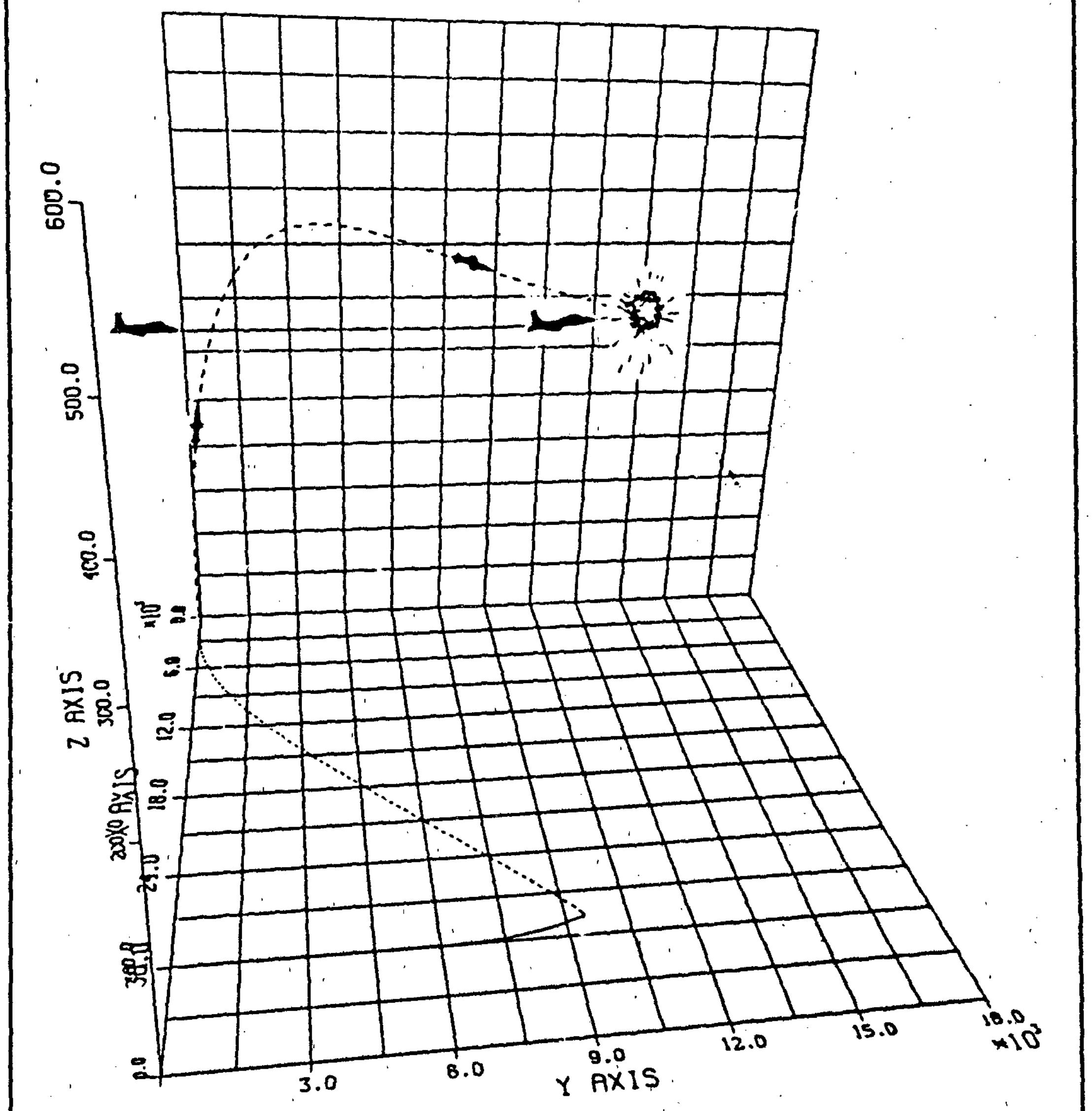


Figure 64. Scenario 3: Three Dimensional Plot

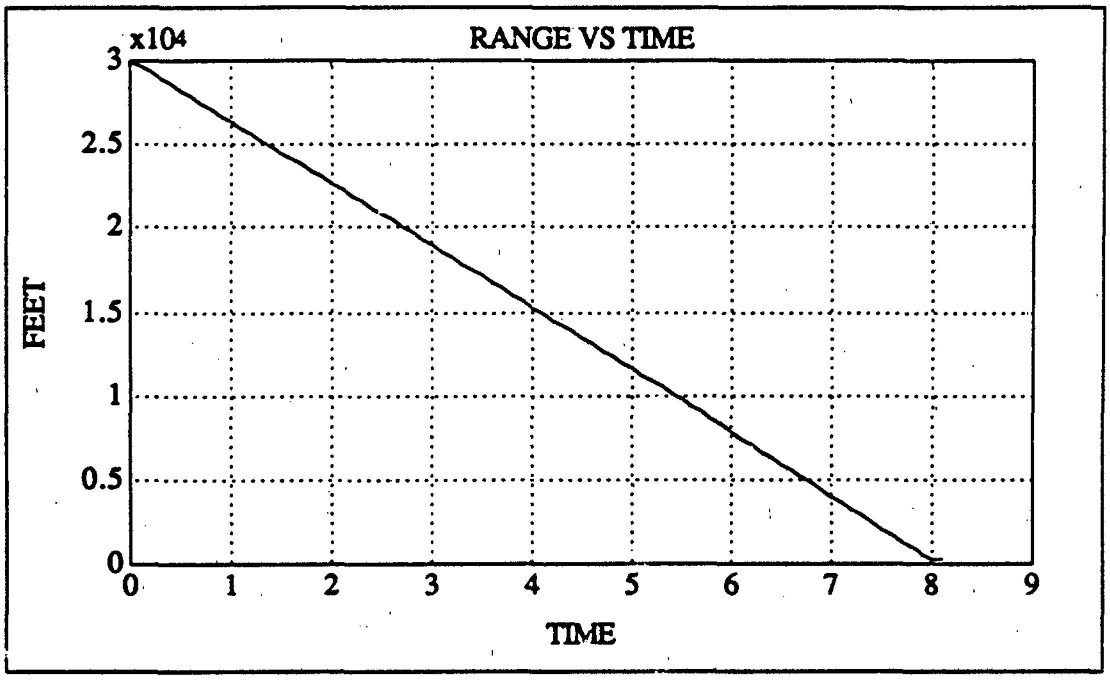


Figure 65. Scenario 4: Missile to Target Range

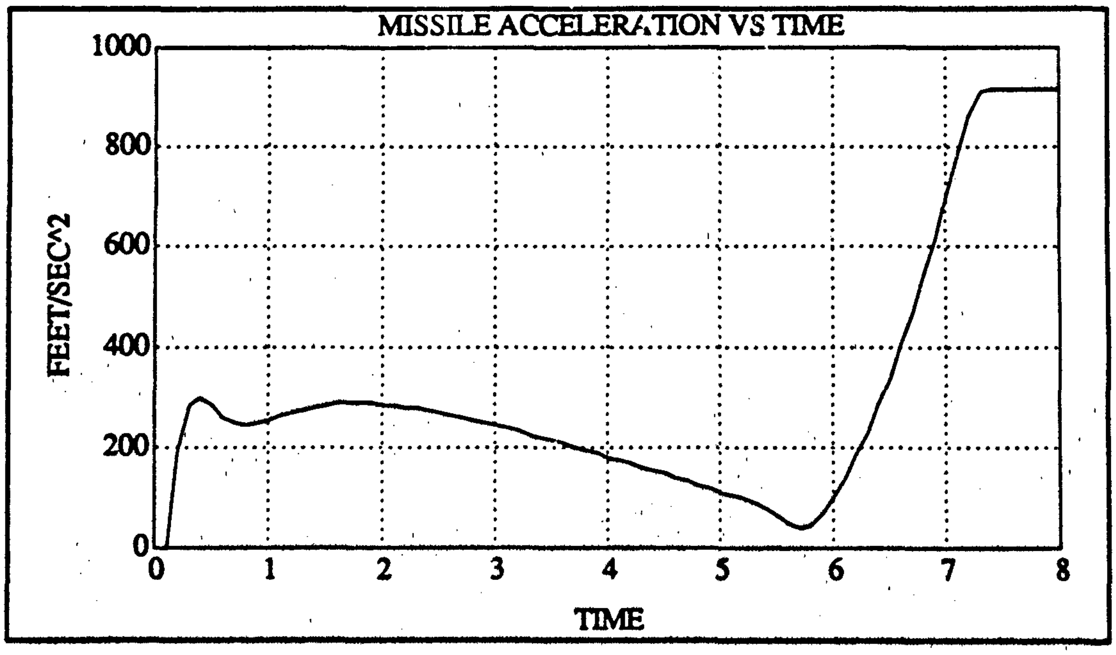


Figure 66. Scenario 4: Missile Acceleration $|A_m|$

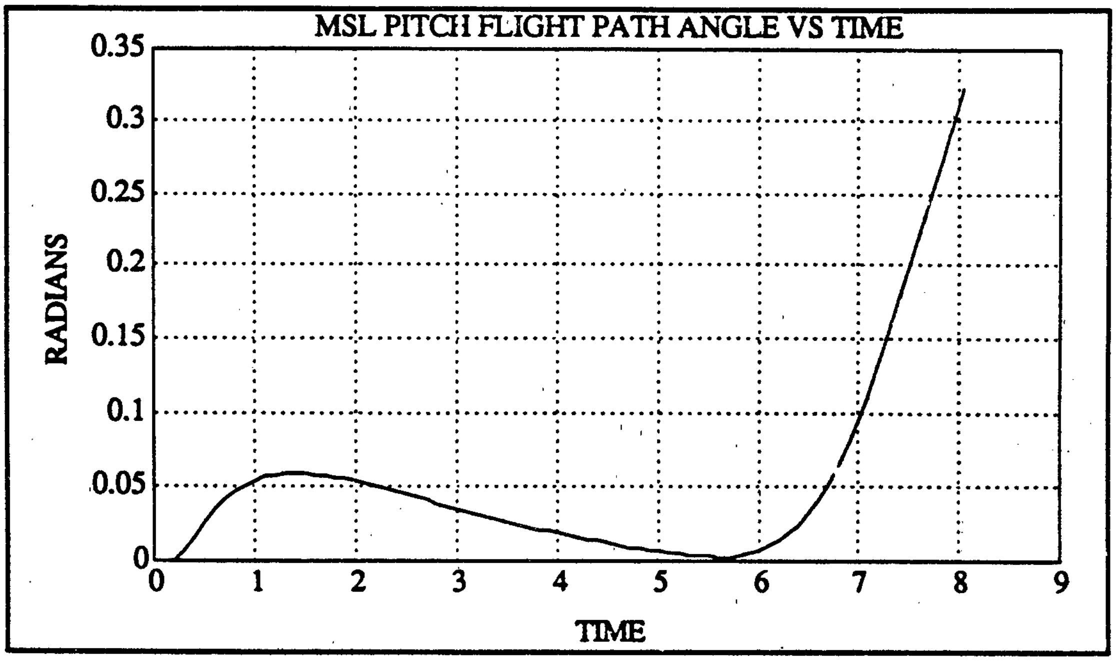


Figure 67. Scenario 4: Missile Pitch Flight Path Angle γ_{m_pitch}

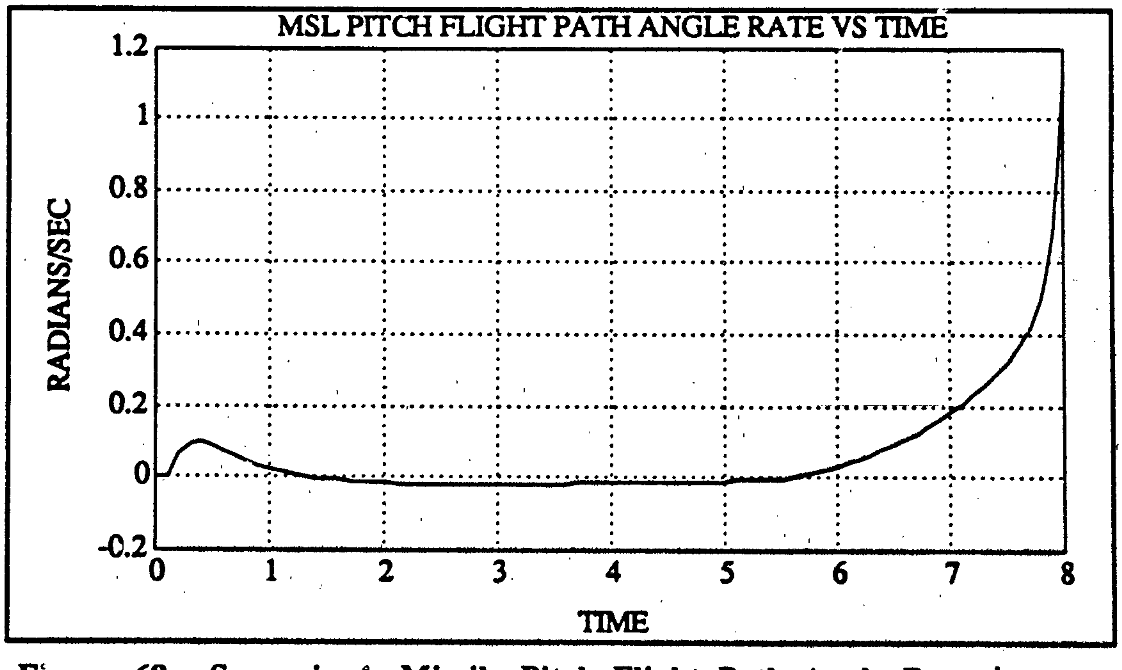


Figure 68. Scenario 4: Missile Pitch Flight Path Angle Rate $\dot{\gamma}_{m\_pitch}$

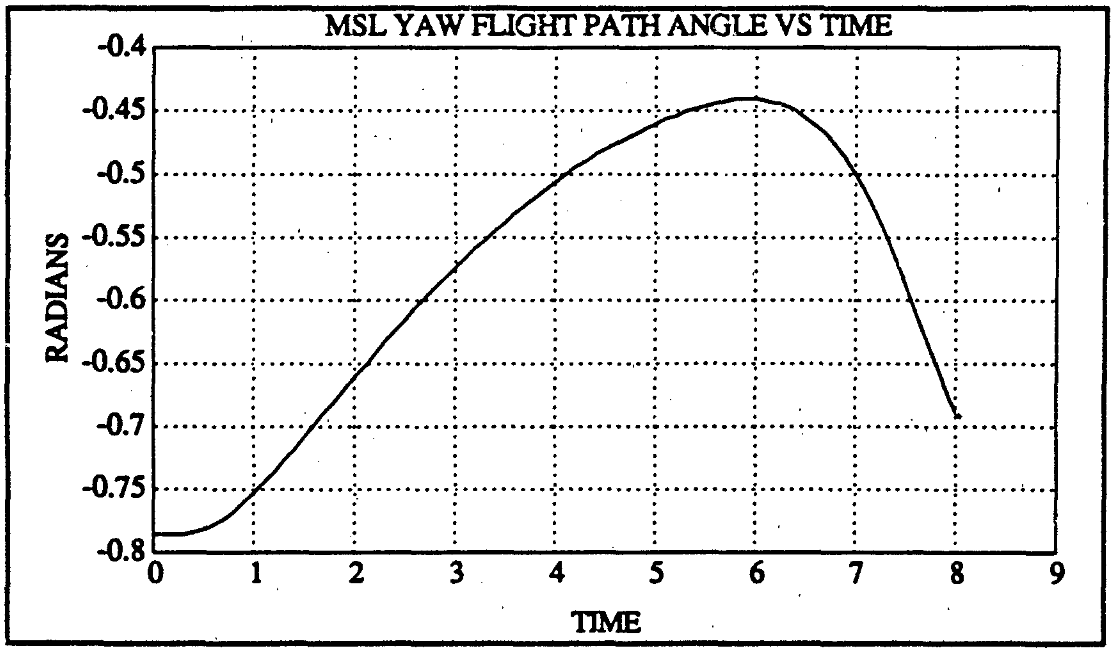


Figure 69. Scenario 4: Missile Yaw Flight Path Angle γ_{m_yaw}

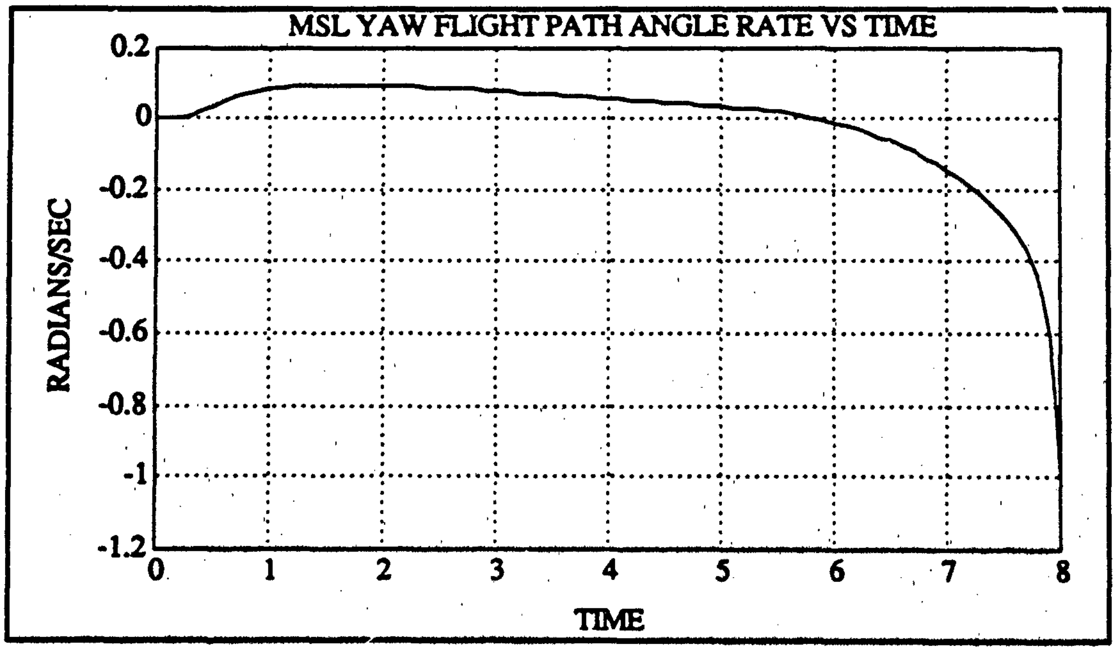


Figure 70. Scenario 4: Missile Yaw Flight Path Angle Rate $\dot{\gamma}_{m\_yaw}$

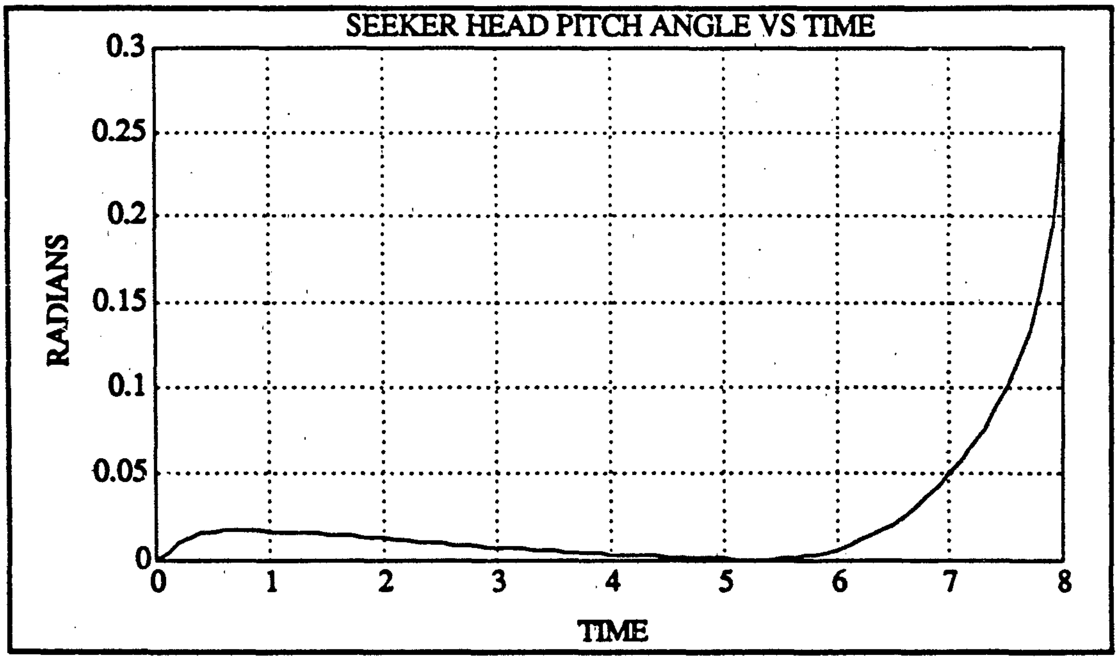


Figure 71. Scenario 4: Seeker Head Pitch Angle β_{pitch}

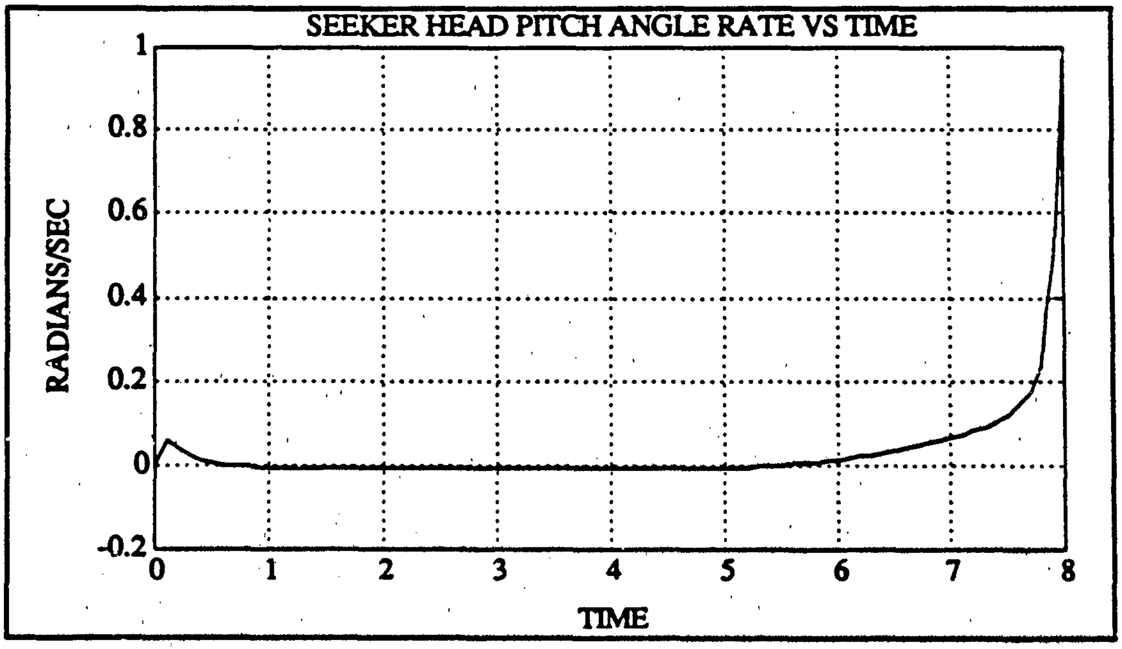


Figure 72. Scenario 4: Seeker Head Pitch Angle Rate $\dot{\beta}_{pitch}$

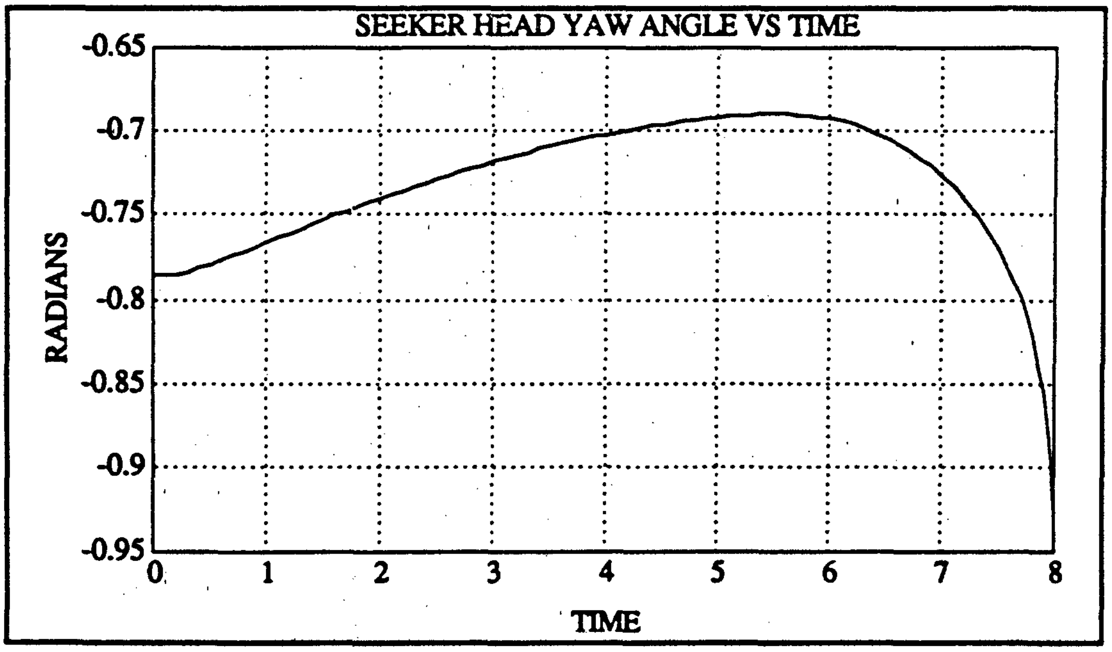


Figure 73. Sceuario 4: Seeker Head Yaw Angle β_{yaw}

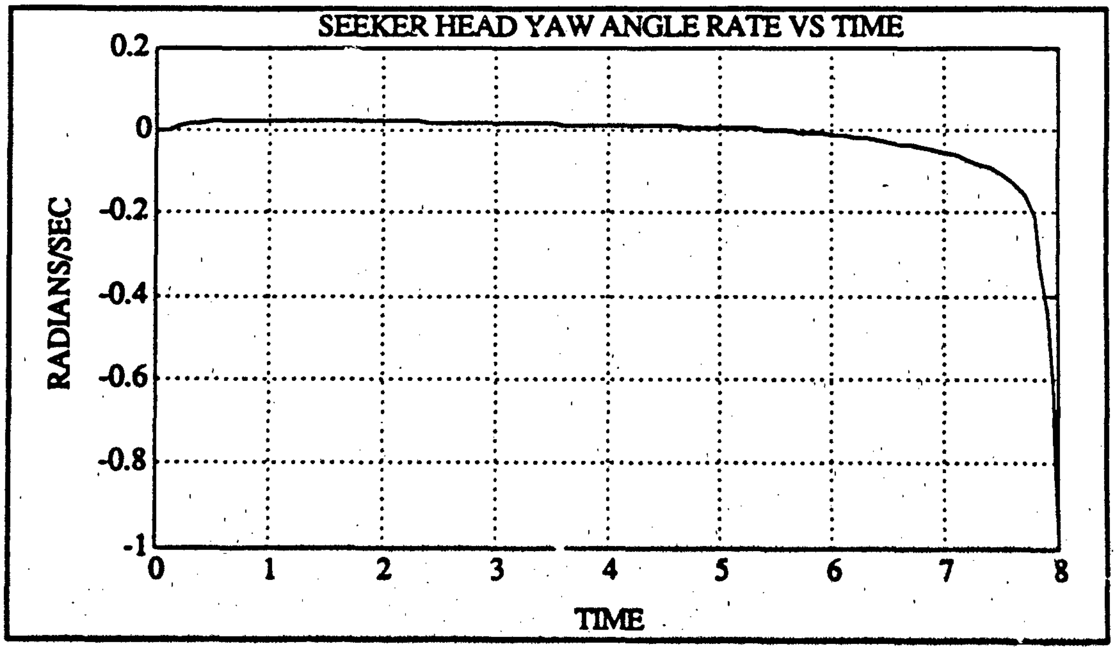


Figure 74. Scenario 4: Seeker Head Yaw Angle Rate $\dot{\beta}_{yaw}$

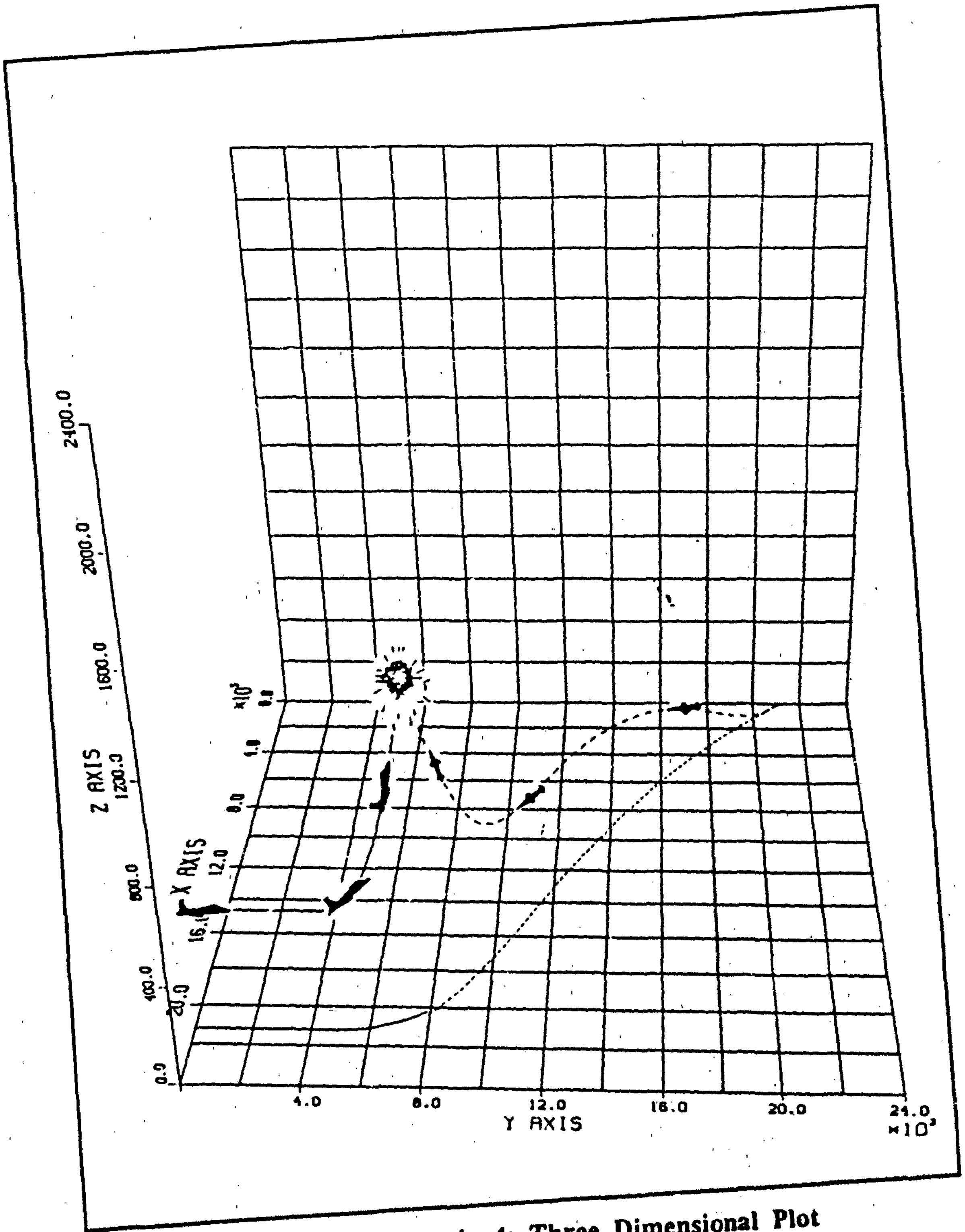


Figure 75. Scenario 4: Three Dimensional Plot

C. MISS DISTANCE ASSESSMENT

A two dimensional adjoint model is used to investigate the missile to target miss distance due to a variety of evasion maneuvers. The optimal time to initiate these maneuvers is determined.

Three maneuvers are used:

1. Step Target Acceleration.
2. Barrel Roll.
3. Split 'S'.

Additional variables are

V_c	=	closing velocity.
η_t	=	target acceleration magnitude.
V_mHE	=	missile velocity due to heading error.
ω_t	=	maneuver frequency.
L	=	half period of Split 'S'.

The following initial conditions are constant throughout all simulations:

N	=	4.0.
T_f	=	6.0 seconds.
V_c	=	2500 feet per second.
V_mHE	=	0.0 feet per second.
τ_{sp}	=	1.0 second.
τ_{sh}	=	0.1 second.

1. Step Target Acceleration

The step acceleration is simulated by multiplying a step input signal by the magnitude of the target acceleration, η_t. Target accelerations of 1.0, 6.0, and 8.0 g's are simulated.

2. Barrel Roll

A two dimensional Barrel Roll maneuver can be represented by a sinusoid of frequency ω_t. A shaping filter is used to simulate the maneuver. [Ref. 7] Figure 76 depicts a typical Barrel Roll maneuver and Figure 77 shows the shaping filter equivalent where ω_t equals 1.0 radian/second. The filter is converted to a block diagram which is simulated using state variable design. The block diagram is shown in Figure 78. A 4.0 and 6.0 g Barrel Roll maneuver are simulated.

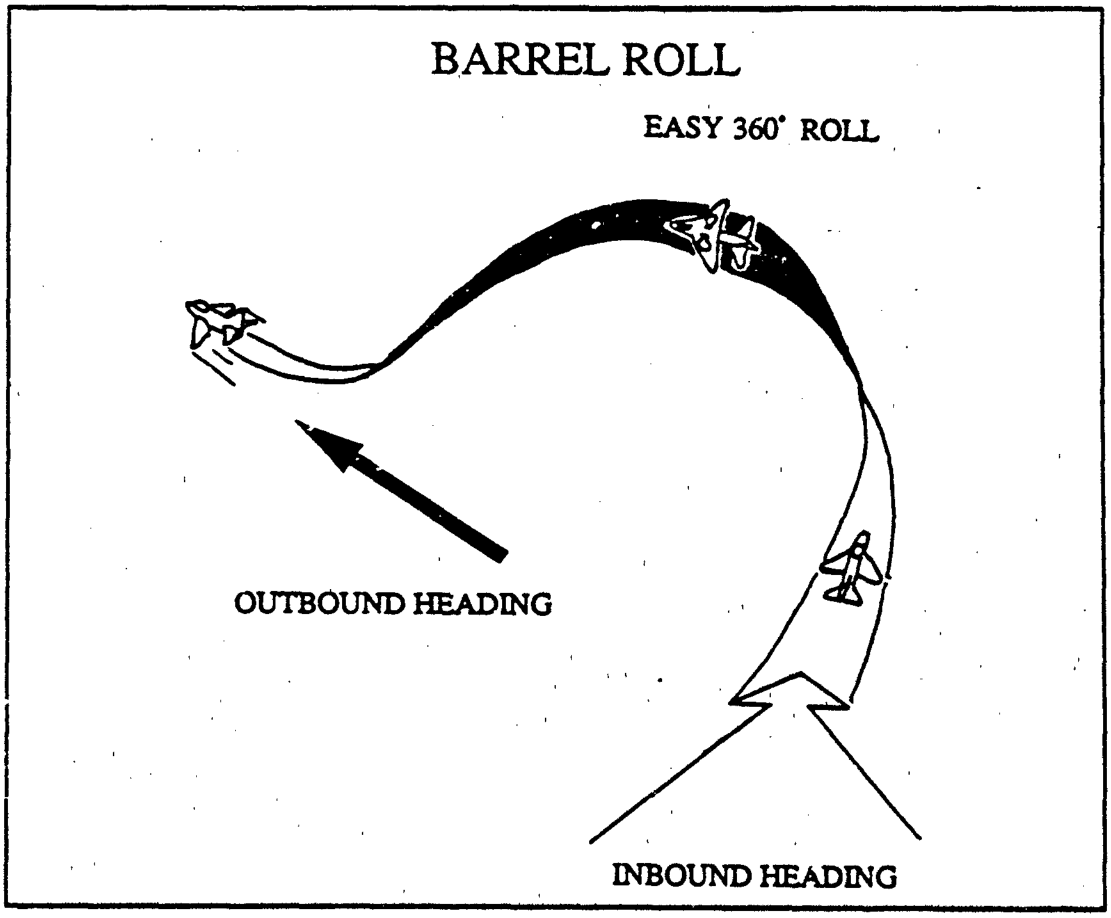
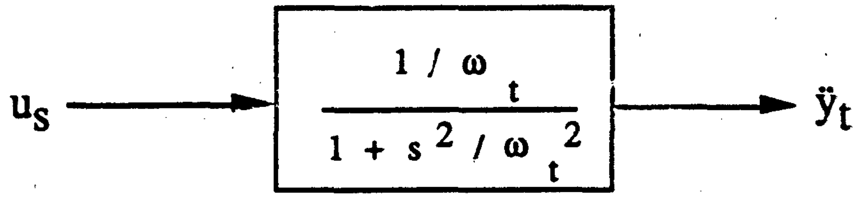


Figure 76. Typical Barrel Roll Maneuver



u_s = White Noise With Power Spectral Density $q_s(t)$

$q_s(t) = \eta_t^2 / T_f \quad 0 \le t \le T_f$

= 0 , otherwise

Figure 77. Shaping Filter Equivalent of a Barrel Roll

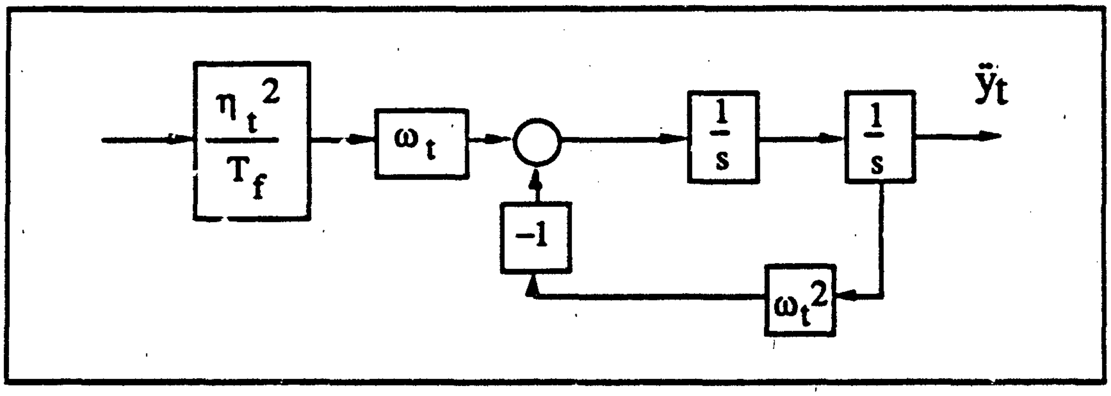


Figure 78. Barrel Roll Block Diagram

3. Split 'S'

A two dimensional Split 'S' maneuver is represented by a periodic square wave with period 2L. A shaping filter is also used to simulate the maneuver. [Ref. 7] Figure 79 depicts a typical Split 'S' maneuver and Figure 80 shows the shaping filter equivalent where L equals 1.0 second. The block diagram of this filter is shown in Figure 81. A 4.0 and 6.0 g Split 'S' maneuver are simulated.

4. Forward Time and Adjoint Model Simulation Results

Figures 82-95 show the results of the forward time and adjoint model simulations. Figures 82-87 show the results of the step acceleration maneuver. Figures 88-91 show the results of the Barrel Roll evasion maneuver. Figures 92-95 show the results of the Split 'S' evasion maneuver.

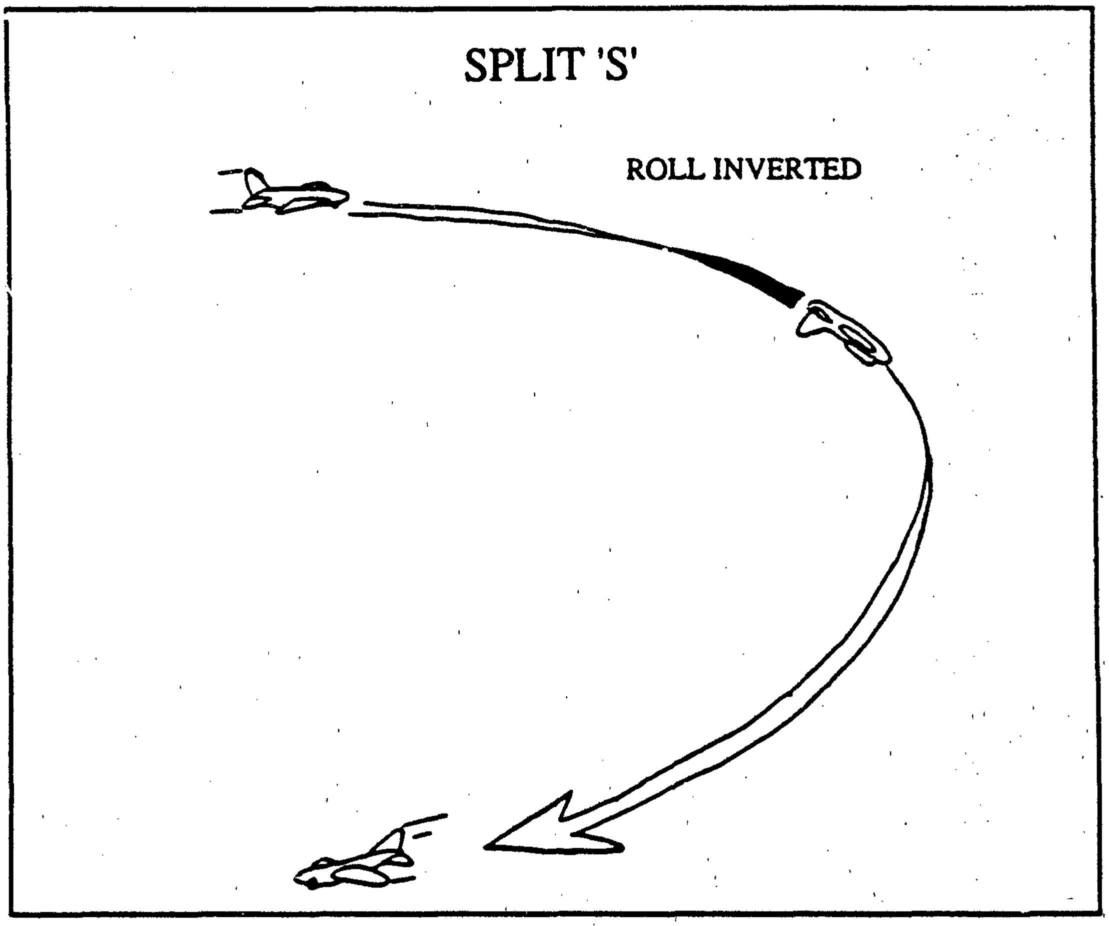
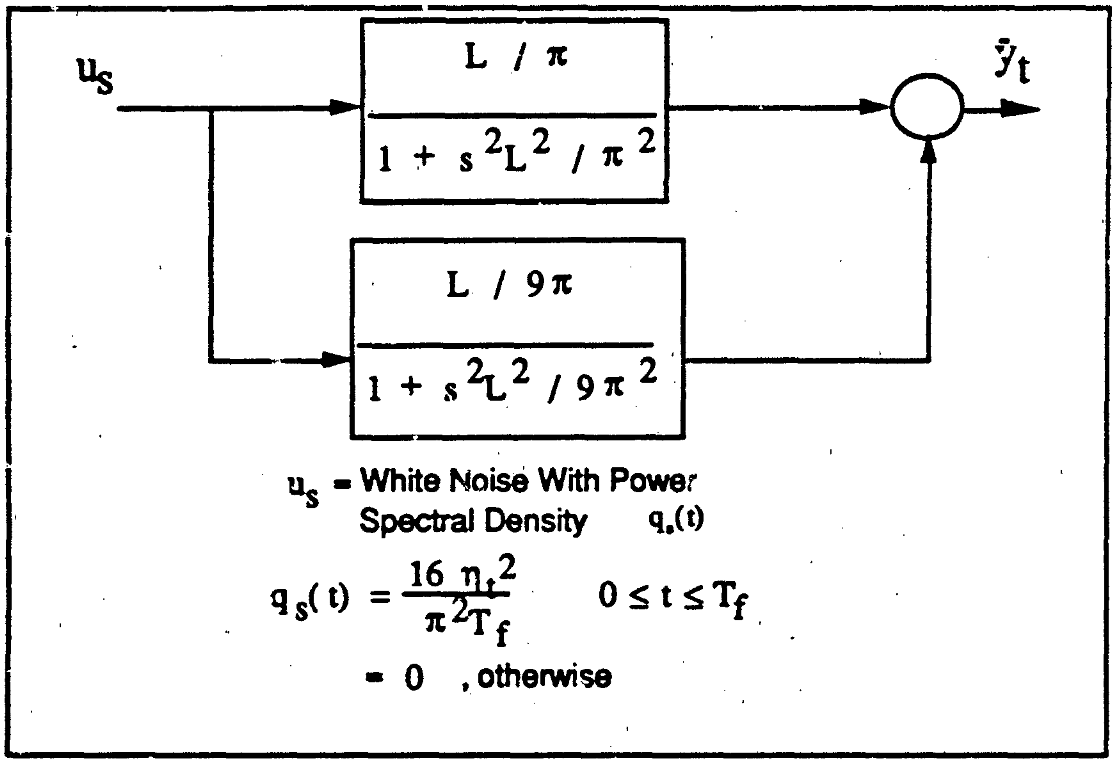


Figure 79. Typical Split 'S' Maneuver


u_S
$\frac{L/\pi}{1 + s^2L^2/\pi^2}$
$\frac{L/9\pi}{1 + s^2L^2/9\pi^2}$
$\ddot{y}_t$

u_s = White Noise With Power Spectral Density $q_s(t)$

$$q_s(t) = \frac{16\,\eta_t^2}{\pi^2 T_f} \qquad 0 \le t \le T_f$$
$$= 0 \quad \text{, otherwise}$$

Figure 80. Shaping Filter Equivalent of Split 'S' Maneuver

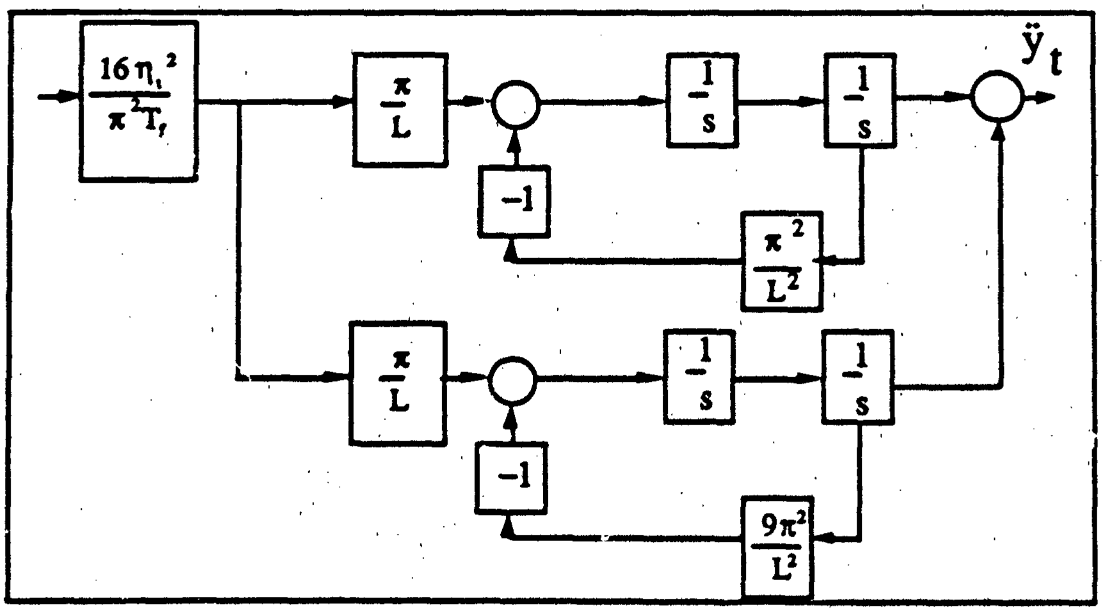


Figure 81. Split 'S' Block Diagram

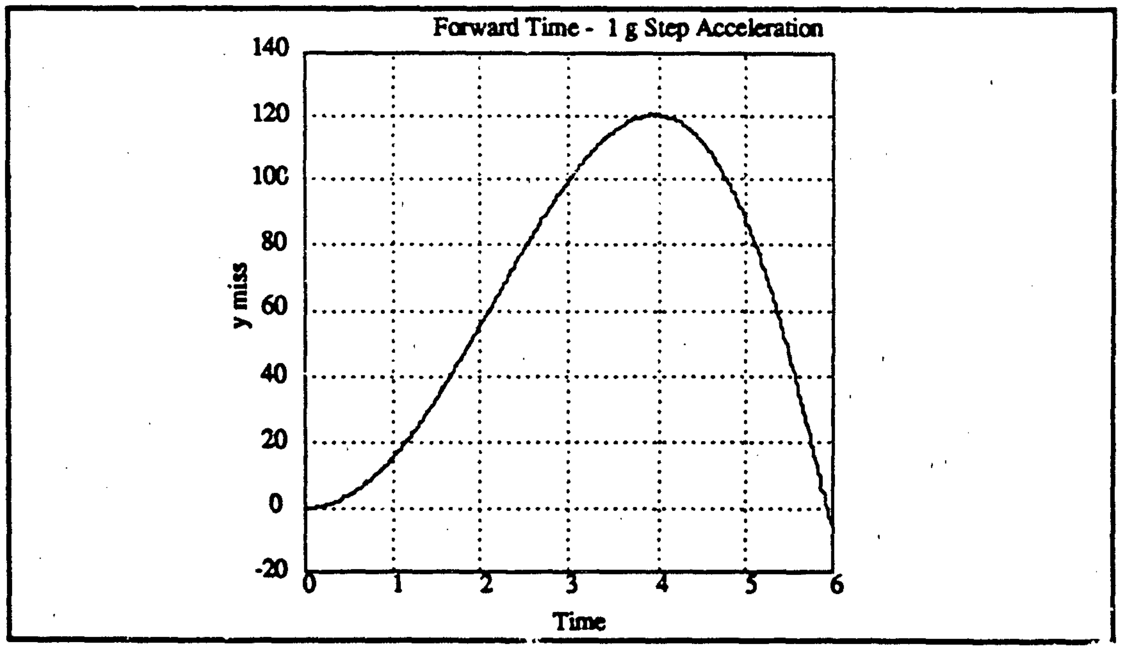


Figure 82. Forward Time, 1 g Step Acceleration

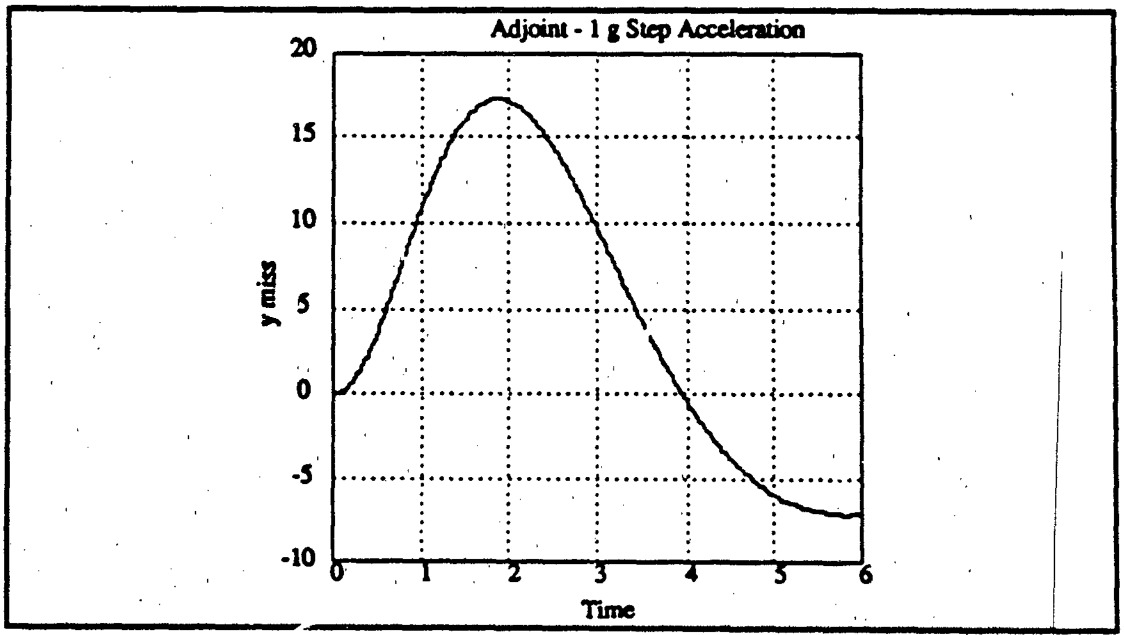


Figure 83. Adjoint, 1 g Step Acceleration

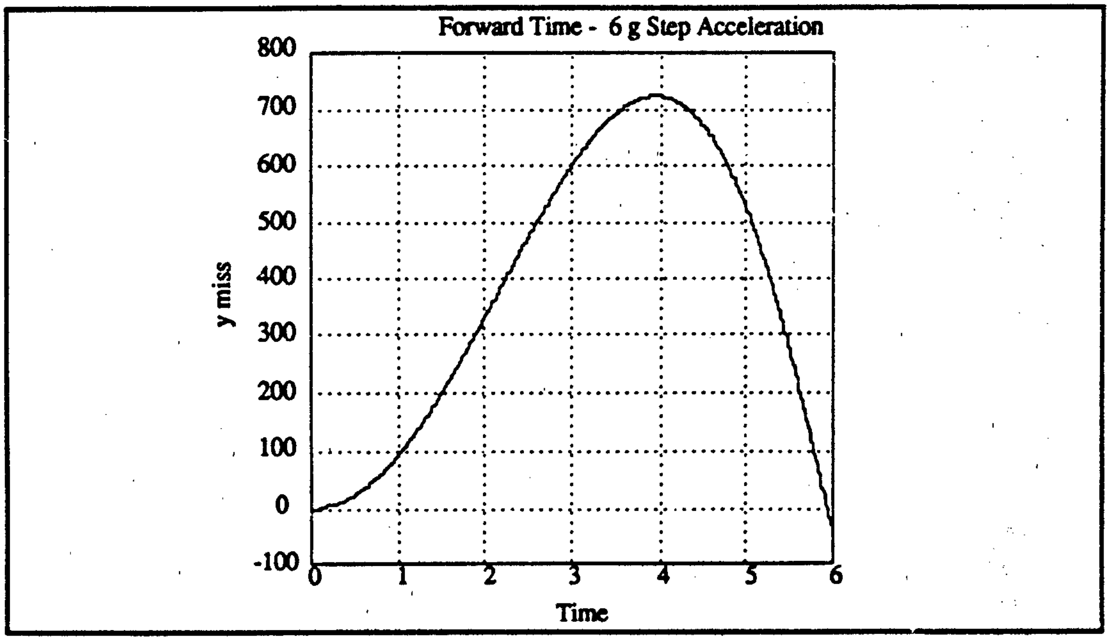


Figure 84. Fo:ward Time, 6 g Step Acceleration

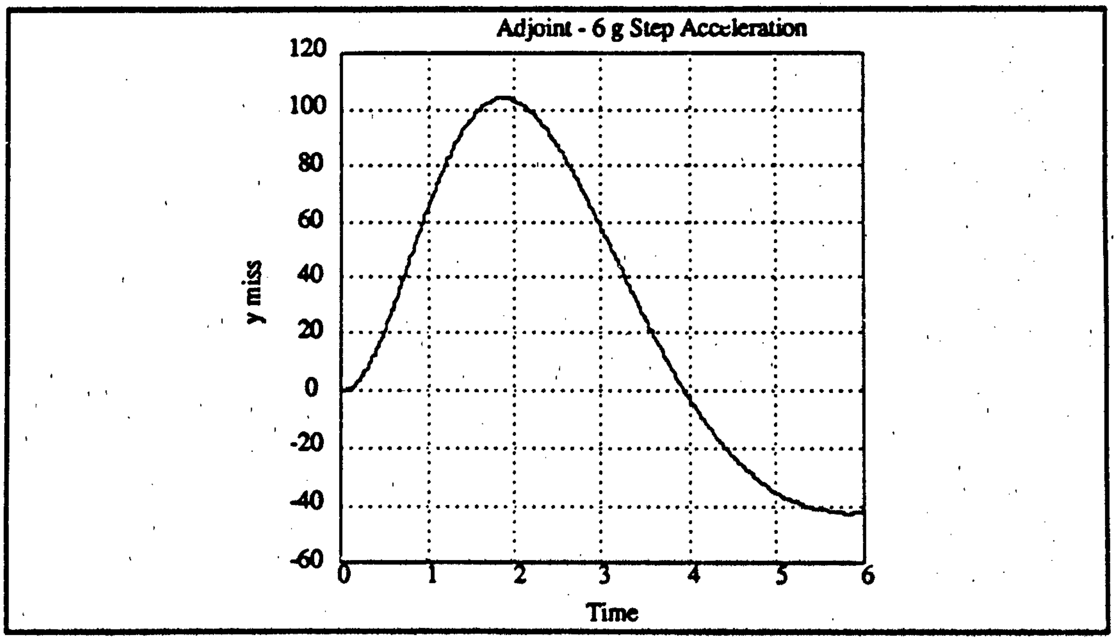


Figure 85. Adjoint, 6 g Step Acceleration

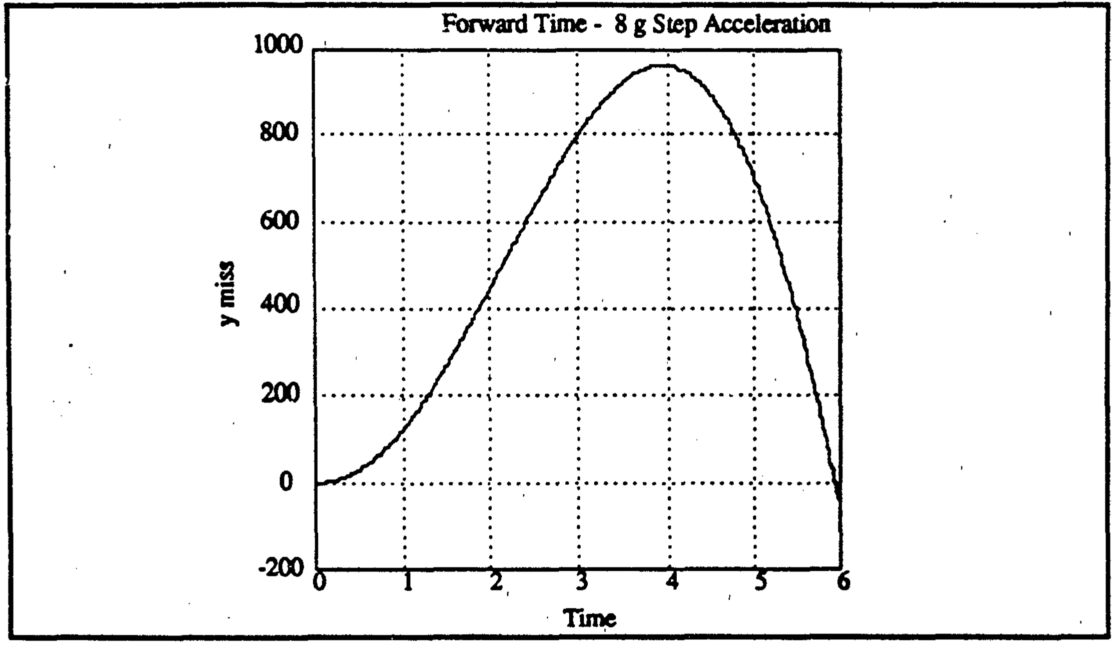


Figure 86. Forward Time, 8 g Step Acceleration

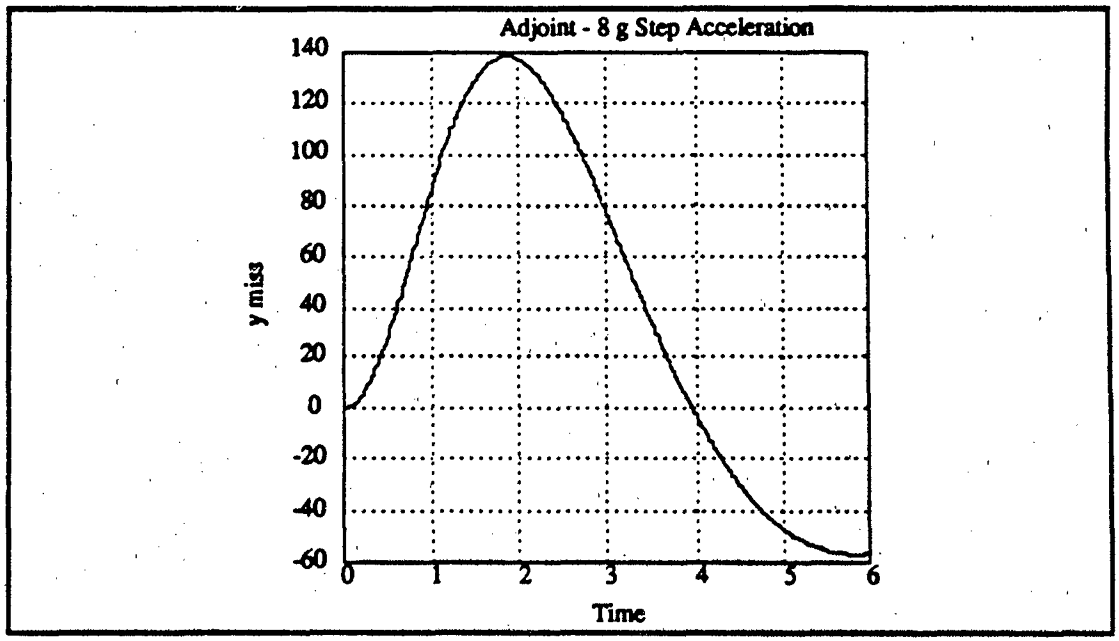


Figure 87. Adjoint, 8 g Step Acceleration

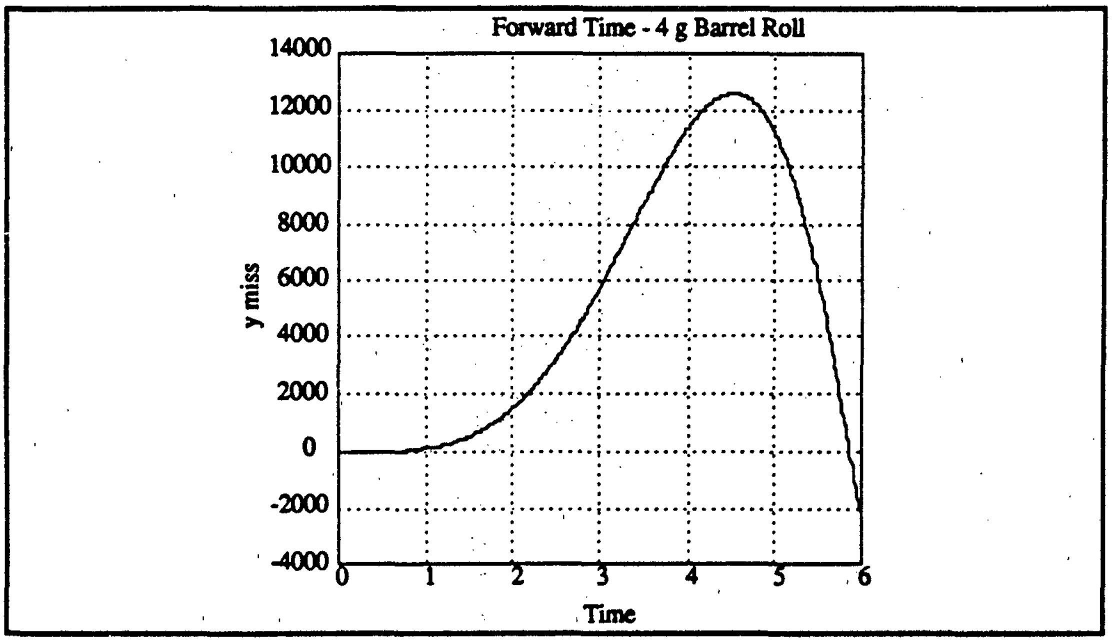


Figure 88. Forward Time, 4 g Barrel Roll

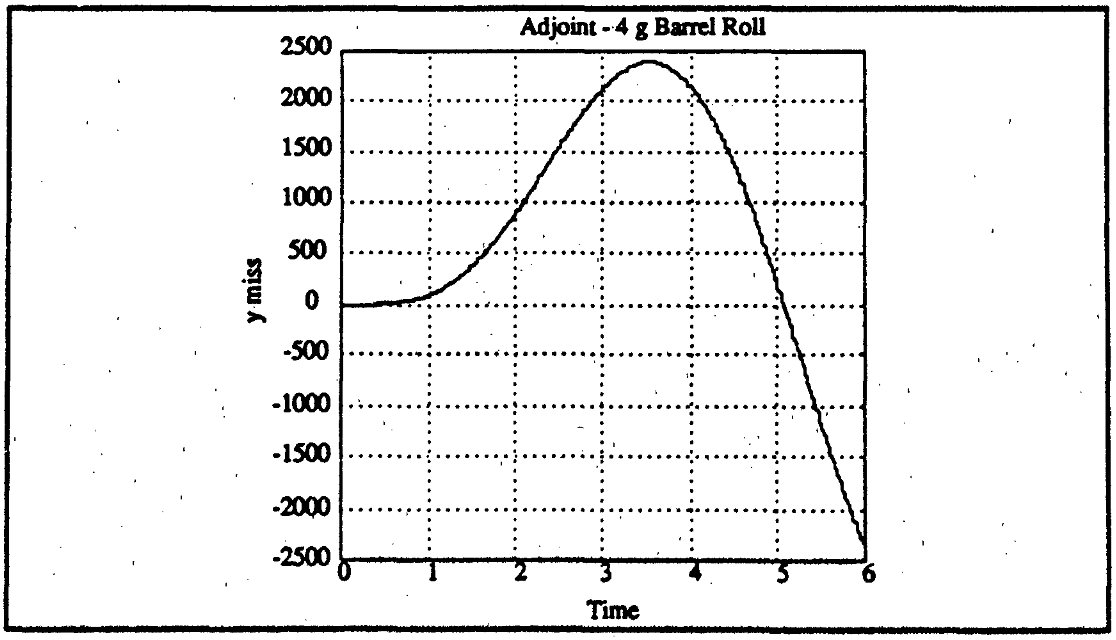


Figure 89. Adjoint, 4 g Barrel Roll

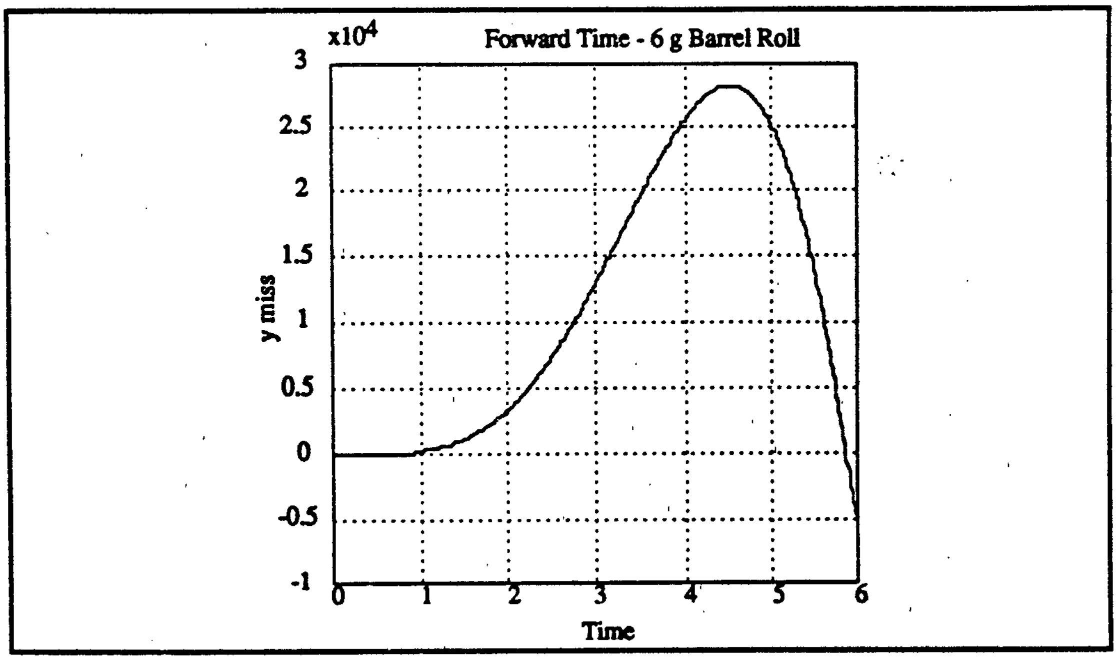


Figure 90. Forward Time, 6 g Barrel Roll

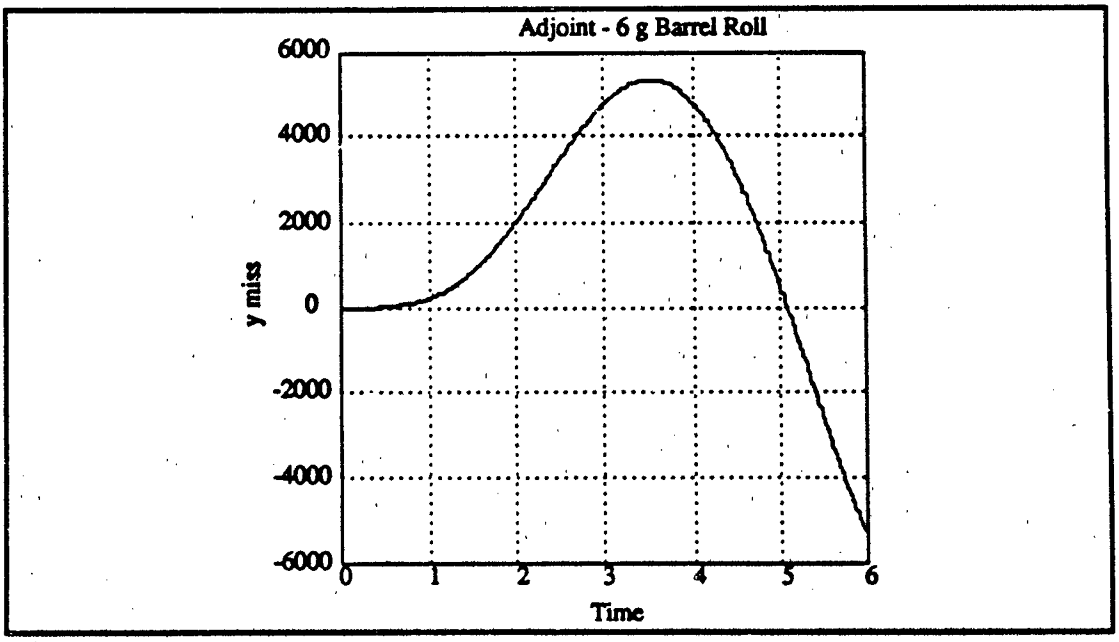


Figure 91. Adjoint, 6 g Barrel Roll

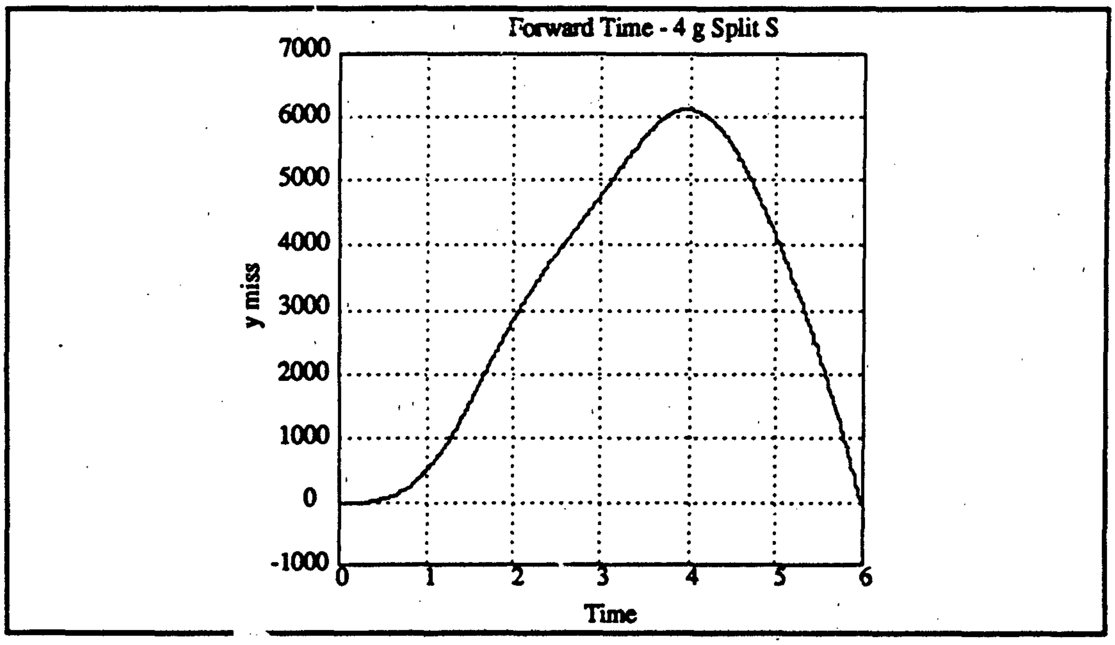


Figure 92. Forward Time, 4 g Split 'S'

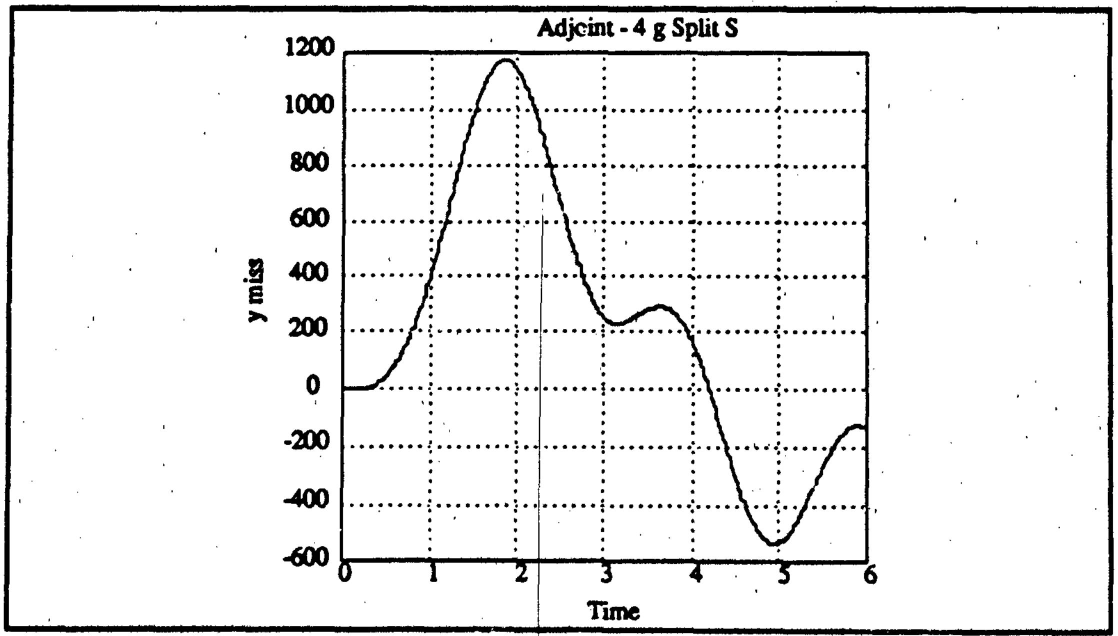


Figure 93. Adjoint, 4 g Split 'S'

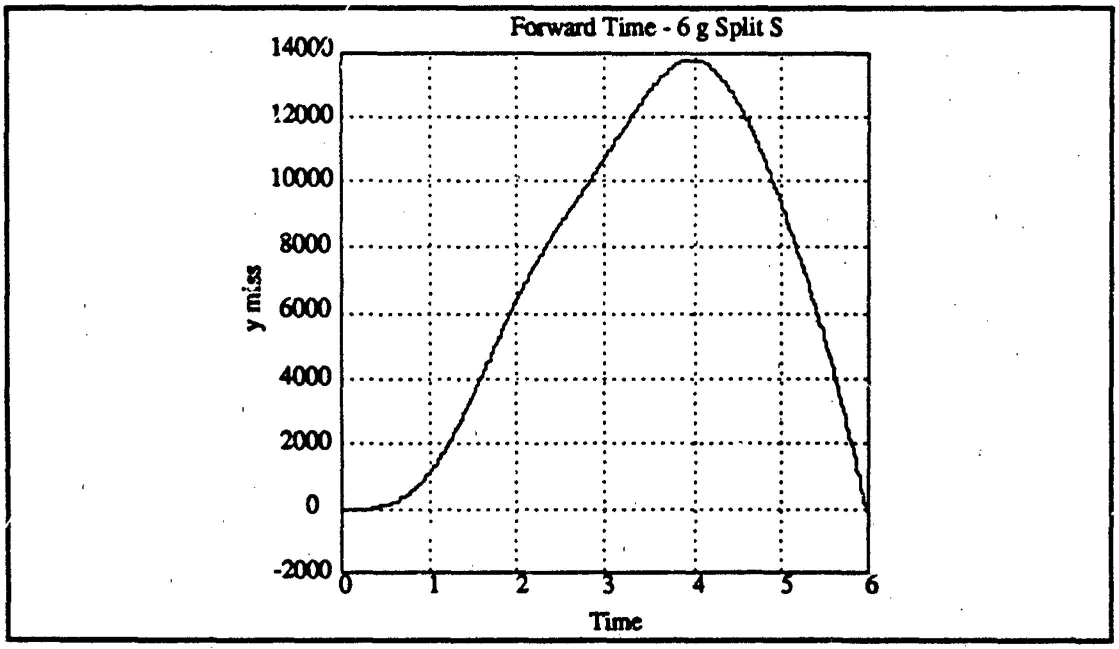


Figure 94. Forward Time, 6 g Split 'S'

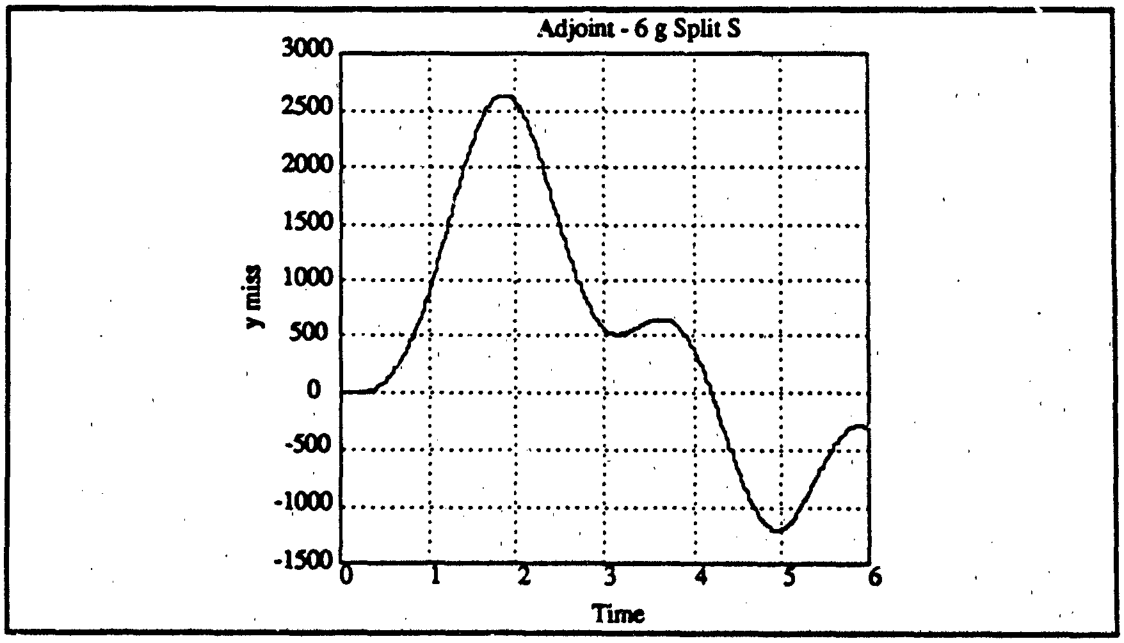


Figure 95. Adjoint, 6 g Split 'S'

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Clearly, the classical proportional navigation model of the missile guidance and control system is sufficient for simple trajectory study. The use of the seeker head angle rate, $\dot{\beta}$, as the estimate of the actual line of sight rate, $\dot{\sigma}$, gives accurate enough information for target tracking in a benign environment. The three dimensional model of the missile tracks a maneuvering target in one, two, or three dimensions with enough accuracy to cause an aircraft kill. A navigation ratio of 4.0 generates sufficient missile acceleration commands for the simplified scenarios studied.

The adjoint model proves to be a very useful indicator of miss distance properties. With the step acceleration evasion maneuver, the optimal time to initiate it is about 2.0 seconds, or 5000 feet (closing velocity is 2500 feet per second), prior to missile impact. The miss distance increases as the target's g loading increases. The maximum miss distance of 140 feet is achieved with an 8.0 g maneuver.

With the Barrel Roll maneuver, the optimal time to initiate it is about 3.5 seconds, or 8750 feet, prior to missile impact. The miss distance again increases with increased target g loading. The maximum miss distance of 5000 feet is achieved with a 6.0 g maneuver.

With the Split 'S' maneuver, the optimal time to initiate it is about 1.8 seconds, or 4500 feet, prior to missile impact. The miss distance increases with increased target g loading. The maximum miss distance of 2600 feet is achieved with a 6.0 g maneuver.

Clearly, the maximum miss distance and optimal time to maneuver depends on the evasion maneuver used. In general, it is recommended that a maximum g evasion maneuver be initiated when the missile is between 1.0 and 1.5 miles prior to impact.

B. RECOMMENDATIONS FOR FURTHER STUDY

The simulations developed in this thesis provide a very good generic model which can be easily modified for more complex missile systems. The programs are easily adjusted for systems with different system time constants and navigation ratios.

The problems associated with measurement noise and seeker head measurement errors might be investigated to see their role in miss distance evaluation.

It is also recommended that target Electronic Counter Measures (ECM) be added to the simulation of the evasion maneuvers to see their effects on miss distance.

APPENDIX A-THREE DIMENSIONAL PROGRAM

```
clear
clg
%
%       LT Francis C. Lukenbill, USN
%       08 August 1990
%       This program simulates a 3 dimensional target/missile
%       engagement scenario using classical proportional navigation.
%
%        DEFINE STATES
%
%       Missile
%
%       ms=[xm  - missile x coordinate
%          xdm  - missile velocity-x direction
%          ym   - missile y coordinate
%          ydm  - missile velocity-y direction
%          zm   - missile z coordinate
%          zdm  - missile velocity-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
```

```
 000
 0 0 1];
%
%     Seeker Head
%
%     beta=[beta pitch  - seeker head pitch angle
%           betad pitch - seeker head pitch angle rate
%           beta yaw    - seeker head yaw angle
%           betad yaw   - seeker head yaw angle rate]
%
AS=[ 0   1   0   0
   -100 -20  0   0
     0   0   0   1
     0   0 -100 -20];

BS=[ 0  0
   100  0
    0   0
    0  100];
%
%     Autopilot
%
%     gammadm=[gammadm pitch - pitch body angle rate
%              gammadm yaw   - yaw body angle rate  ]
%
AP=[-1  0
    0 -1];

BP=[1  0
   0  1];
%
%     Target
%
%     ts=[xt  -  target x coordinate
%         xdt  -  target velocity-x direction
%         yt   -  target y coordinate
%         ydt  -  target velocity-y direction
%         zt   -  target z coordinate
%         zdt  -  target velocity-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=.01;

[phis,dels]= c2d(AS,BS,dt);
[phim,delm]= c2d(AM,BM,dt);
[phia,dela]= c2d(AP,BP,dt);
[phit,delt]= c2d(AT,BT,dt);

tfinal= 15.0;
kmax=tfinal/dt +1;
%
%      INITIALIZE VARIABLES
%
```

```
%       navigation ratio
%
nr=4.0;
NR=[nr  0
   0   nr];
%
%       initial seeker head angles and angle rates
%
beta_pitch0 =0.0;
betad_pitch0=0.0;
beta_yaw0   =0.0;
betad_yaw0  =0.0;

beta(:,1)=[beta_pitch0
           betad_pitch0
           beta_yaw0
           betad_yaw0  ];
%
%       initial missile body angle rates
%
gammadm_pitch0=0.0;
gammadm_yaw0  =0.0;

gammadm(:,1)=[gammadm_pitch0
              gammadm_yaw0   ];
%
%       initial missile states
%
xm0 =0.0;
xdm0=3000.0;
ym0 =0.0;;
ydm0=0.0;
zm0 =0.0;
zdm0=0.0;
```

```
ms(:,1)=[xm0
        xdm0
        ym0
        ydm0
        zm0
        zdm0];
%
%       initial target states
%
xt0 = 30000.0;
xdt0=0.0;
yt0 =0.0;
ydt0= 1000.0;
zt0 =500.0;
zdt0=0.0;

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 time
%
time(1)=0.0;
%
%      SIMULATE THE SYSTEM
%
for (i=1:kmax-1)
%
%      calculate missile and target speeds
%
      vm(i)=sqrt(ms(2,i)^2 + ms(4,i)^2 + ms(6,i)^2);
      vt(i)=sqrt(ts(2,i)^2 + ts(4,i)^2 + ts(6,i)^2);
%
%      calculate 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)),(abs(ts(1,i)-ms(1,i))));

      sigma(:,i)=[sigma_pitch(i)
                sigma_yaw(i)  ];
%
%      calculate missile and target line-of-sight angles
%
      sigmam_pitch(i)=atan2(ms(5,i),sqrt(ms(1,i)^2 + ms(3,i)^2));
      sigmam_yaw(i)  =atan2(ms(3,i),ms(1,i));
      sigmat_pitch(i)=atan2(ts(5,i),sqrt(ts(1,i)^2 + ts(3,i)^2));
      sigmat_yaw(i)  =atan2(ts(3,i),ts(1,i));
```

```
%
%     update seeker head states
%
    beta(:,i+1)=phis*beta(:,i) + dels*sigma(:,i);
%
%     set up seeker head angle rate vector
%
    betad(:,i)=[beta(2,i)
               beta(4,i)];
%
%     calculate 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));
%
%     update autopilot states
%
    gammadm(:,i+1)=phis*gammadm(:,i) + dela*NR*betad(:,i);
%
%     compute missile velocity in the R'Z (pitch) plane (R'=Rxy)
%
   if (sigma_pitch(i)~=0.0)
       vm_pitch(i)=abs(vm(i)*cos(sigma_yaw(i)-gammam_yaw(i)));
%
%     limit pitch acceleration to approximately 20 g's
%
       if (abs(vm_pitch(i)*gammadm(1,i))<=644.0)
          am_pitch(i)=vm_pitch(i)*gammadm(1,i);
       else
          am_pitch(i)=644.0*sign(vm_pitch(i)*gammadm(1,i));
       end
   else
```

```
        vm_pitch(i)=0.0;
        am_pitch(i)=0.0;
    end
%
%    calculate missile pitch acceleration vector components
%
    am_pitch_xy(i)=abs(am_pitch(i)*sin(sigma_pitch(i)));
    xddm_pitch(i) =-am_pitch_xy(i)*cos(sigma_yaw(i));
    yddm_pitch(i) =-am_pitch_xy(i)*sin(sigma_yaw(i));
    zddm_pitch(i) =am_pitch(i)*cos(sigma_pitch(i));
%
%    compute missile velocity in the XY (yaw) plane
%
    if  ((sigmam_yaw(i) - sigmat_yaw(i))~=0.0)
         vm_yaw(i)=abs(vm(i)*cos(gammam_pitch(i)));
%
%    limit yaw acceleration to approximately 20 g's
%
        if(abs(vm_yaw(i)*gammadm(2,i))<=644.0)
             am_yaw(i)=vm_yaw(i)*gammadm(2,i);
        else
             am_yaw(i)=644.0*sign(vm_yaw(i)*gammadm(2,i));
        end
    else
        vm_yaw(i)=0.0;
        am_yaw(i)=0.0;
    end
%
%    calculate missile yaw acceleration vector components
%
    xddm_yaw(i)=-am_yaw(i)*sin(sigma_yaw(i));
    yddm_yaw(i)= am_yaw(i)*cos(sigma_yaw(i));
%
%    compute overall missile acceleration components
```

```
%
      xddm(i)=xddm_pitch(i) + xddm_yaw(i);
      yddm(i)=yddm_pitch(i) + yddm_yaw(i);
      zddm(i)=zddm_pitch(i);
%
%       compute total missile acceleration magnitude
%
      am(i)=sqrt(xddm(i)^2 + yddm(i)^2 + zddm(i)^2);
%
%       generate missile input vector
%
      um=[xddm(i)
         yddm(i)
         zddm(i)];
%
%       update missile states
%
      ms(:,i+1)=phim*ms(:,i) + delm*um;
%
%       set target acceleration components
%
       if (r(i)<=0.0)
           xddt(i)=-6.5*32.2*sin(gammat_yaw(i));
           yddt(i)= 6.5*32.2*cos(gammat_yaw(i));
          zddt(i)=0.0;

       else
          xddt(i)=0.0;
          yddt(i)=0.0;
          zddt(i)=0.0;
       end
%
%       compute target acceleration magnitude
%
```

```
        at(i)=sqrt(xddt(i)^2 + yddt(i)^2 + zddt(i)^2);
%
%        generate target input vector
%
        ut=[xddt(i)
           yddt(i)
           zddt(i)];
%
%        update target states
%
        ts(:,i+1)=phit*ts(:,i) + delt*ut;
%
%        calculate time to go
%
         vt_pitch(i)=abs(vt(i)*cos(sigma_yaw(i)-gammat_yaw(i)));
        rdot(i)= vt_pitch(i)*cos(gammat_pitch(i)-sigma_pitch(i))...
              - vm_pitch(i)*cos(gammam_pitch(i)-sigma_pitch(i));
        ttg(i)=r(i)/(-rdot(i));
%
%        calculate updated 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);
        rx(i+1)=(ts(1,i+1) - ms(1,i+1));
        ry(i+1)=(ts(3,i+1) - ms(3,i+1));
        rz(i+1)=(ts(5,i+1) - ms(5,i+1));
%
%        set up missile and target trajectory data for plotting in 3-D
%
        missile(i,:)=[ms(1,i) ms(3,i) ms(5,i)];
        target(i,:) =[ts(1,i) ts(3,i) ts(5,i)];
%
```

```
%     update time vector
%
    time(i+1)=time(i) + dt;
%
%     check to see it engagement is at closest point of approach
%
    if(r(i)<r(i+1)),break,end
end;
%
%     PLOT RESULTS
%
%     time to go
%
plot(r(1:i-1),ttg(1:i-1)),grid,xlabel('Range - FEET'),
ylabel('Time-to-go  - SECONDS');title('RANGE vs TIME TO GO');
pause,clg
%
%     range data
%
plot(time,r),grid,xlabel('TIME'),ylabel('FEET')
title('RANGE VS TIME')
pause,clg
%
%     missile and target velocity information
%
plot(time(1:i),vm),grid,xlabel('TIME'),ylabel('FEET/SEC')
title('MISSILE VELOCITY VS TIME');
pause,clg
plot(time(1:i),vt),grid,xlabel('TIME'),ylabel('FEET/SEC')
title('TARGET VELOCITY VS TIME');
pause,clg
%
%     missile and target acceleration information
%
```

```
plot(time(1:i-1),am(1:i-1)),grid,xlabel('TIME'),ylabel('FEET/SEC^2')
title('MISSILE ACCELERATION VS TIME');
pause,clg
plot(time(1:i),at),grid,xlabel('TIME'),ylabel('FEET/SEC^2')
title('TARGET ACCELERATION VS TIME');
pause,clg
%
%	filght path angles
%
plot(time(1:i),gammam_pitch),grid,xlabel('TIME'),ylabel('RADIANS')
title('MSL PITCH FLIGHT PATH ANGLE VS TIME');
pause,clg
plot(time(1:i),gammam_yaw),grid,xlabel('TIME'),ylabel('RADIANS')
title('MSL YAW FLIGHT PATH ANGLE VS TIME');
pause,clg
plot(time(1:i-1),gammadm(1,(1:i-
1))),grid,xlabel('TIME'),ylabel('RADIANS/SEC')
title('MSL PITCH FLIGHT PATH ANGLE RATE VS TIME');
pause,clg
plot(time(1:i-1),gammadm(2,(1:i-
1))),grid,xlabel('TIME'),ylabel('RADIANS/SEC')
title('MSL YAW FLIGHT PATH ANGLE RATE VS TIME');
pause,clg
%
%	seeker head angle information
%
plot(time(1:i-1),beta(1,(1:i-1))),grid,xlabel('TIME'),ylabel('RADIANS')
title('SEEKER HEAD PITCH ANGLE VS TIME');
pause,clg
plot(time(1:i-1),beta(2,(1:i-1))),grid,xlabel('TIME'),ylabel('RADIANS/SEC')
title('SEEKER HEAD PITCH ANGLE RATE VS TIME');
pause,clg
plot(time(1:i-1),beta(3,(1:i-1))),grid,xlabel('TIME'),ylabel('RADIANS')
title('SEEKER HEAD YAW ANGLE VS TIME');
```

```
pause,clg
plot(time(1:i-1),beta(4,(1:i-1))),grid,xlabel('TIME'),ylabel('RADIANS/SEC')
title('SEEKER HEAD YAW ANGLE RATE VS TIME');
pause,clg
end;
```

APPENDIX B-FORWARD TIME AND ADJOINT PROGRAMS

```
%
% Lukenbill, F. C. 20 November 1990
% This program simulates the forward time and adjoint models with a
%  step acceleration evasion maneuver.
%
%
% Initialize Variables
%
tm=1.0;
tsh=0.1;
a=1/tm;
b=2*(1/tsh);
c=(1/tsh)*(1/tsh);
N=4.0;
Vc=2500;
Tf=6.0;
ki=1/(Vc*Tf);
HE=0.0*pi/180;
Vm=2000.0;
VmHE=Vm*sin(HE);
Atgt=257.6;;
%
%  Input State Matricies
%
A=[0       1     0     0     0       0     0     0     -VmHE
   0       0     1     0     0       0     0     0     0
   0       0     -a    0   a*N*Vc    0     0     0     0
   0       0     0     0     1       0     0     0     0
   -c*ki   0     0     -c    -b    c*ki    0     0     0
   0       0     0     0     0       0     1     0     0
```

```
    0       0    0    0    0    0    0   Atgt   0
    0       0    0    0    0    0    0    0     0
    0       0    0    0    0    0    0    0     0];
%
B=[0
  0
  0
  0
  0
  0
  0
  1
  1];
%
C=[-1  0  0  0  0  1  0   0   0];
%
%  Input Timing Information
%
dt=0.01;
kmax=Tf/dt+1;
%
%  Set up impulse function
%
impulse=zeros(1,kmax);
impulse(1)=1/dt;
%
%  Initialize vectors
%
x=zeros([illegible],kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
%
%   Simulate the System (Forward Time Solution)
%
```

```
for (i=1:kmax-1)
                ki=1/(Vc*(Tf-time(i)));
  A=[0      1      0      0      0       0      0      0      -VmHE
  0         0      1      0      0       0      0      0      0
  0         0      -a     0      a*N*Vc  0      0      0      0
  0         0      0      0      1       0      0      0      0
  -c*ki     0      0      -c     -b      c*ki   0      0      0
  0         0      0      0      0       0      1      0      0
  0         0      0      0      0       0      0      Atgt   0
  0         0      0      0      0       0      0      0      0
  0         0      0      0      0       0      0      0      0];
  [Phifwd,Delfwd]=c2d(A,B,dt);
  x(:,i+1) = Phifwd*x(:,i) + Delfwd*impulse(i);
  y(:,i+1) = C*x(:,i+1);
  time(i+1)=time(i)+dt;
end;
y(i)
%
% Plot Results
%
plot(time(1:i),y(1,(1:i)));title('Forward Time -  8 g Step Acceleration');
xlabel('Time');,ylabel('y miss');grid;pause;
%
% Reinitialize variables and vectors
%
dt=0.01;
kmax=Tf/dt+1;
impulse=zeros(1,kmax);
impulse(1)=1/dt;
x=zeros(9,kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
time(1)=dt;
%
```

```
% Simulate the System (Adjoint Solution)
%
for (i=1:kmax-1)
            ki=1/(Vc*(time(i)));
   A=[0     1     0     0     0       0     0     0      -VmHE
   0        0     1     0     0       0     0     0      0
   0        0     -a    0     a*N*Vc  0     0     0      0
   0        0     0     0     1       0     0     0      0
   -c*ki    0     0     -c    -b      c*ki  0     0      0
   0        0     0     0     0       0     1     0      0
   0        0     0     0     0       0     0     Atgt   0
   0        0     0     0     0       0     0     0      0
   0        0     0     0     0       0     0     0      0];
   [Phiadj,Deladj]=c2d(A',C',dt);
   x(:,i+1) = Phiadj*x(:,i) + Deladj*impulse(i);
   y(:,i+1) = B'*x(:,i+1);
   time(i+1)=time(i)+dt;
end;
y(i)
%
% Plot Results
%
plot(time(1:i),y(1,(1:i)));title('Adjoint - 8 g Step Acceleration');
xlabel('Time');,ylabel('y miss');grid;pause;
end;
```

```
%
% Lukenbill, F. C. 20 November 1990
% This program simulates the forward time and adjoint models with a
% Barrel Roll maneuver.
%
%
% Initialize Variables
%
tm=1.0;
tsh=0.1;
a=1/tm;
b=2*(1/tsh);
c=(1/tsh)*(1/tsh);
N=4.0;
Vc=2500;
Tf=6;
ki=1/(Vc*Tf);
HE=0.0*pi/180;
Vm=2000.0;
VmHE=Vm*sin(HE);
Atgt=193.2;
wt=1.0;
W=wt*wt;
ATGT=wt*(Atgt*Atgt)/Tf;
%
% Input State Matricies
%
A=[ 0     1    0    0    0       0     0    0   -VmHE  0    0
    0     0    1    0    0       0     0    0    0     0    0
    0     0   -a    0  a*N*Vc    0     0    0    0     0    0
    0     0    0    0    1       0     0    0    0     0    0
  -c*ki   0    0   -c   -b      c*ki   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    0    0    0     0    0    0
   0      0    0    0    0    0    0    0     0    0    0
   0      0    0    0    0    0    0    0     0    0    1
   0      0    0    0    0    0    0   ATGT   0   -W    0];
%
B=[0
  0
  0
  0
  0
  0
  0
  1
  1
  0
  0];
%
C=[-1  0 0 0 0 1 0  0  0  0  0];
%
% Input Timing Information
%
dt=0.01;
kmax=Tf/dt+1;
%
% Set up impulse function
%
impulse=zeros(1,kmax);
impulse(1)=1/dt;
%
% Initialize vectors
%
x=zeros(11,kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
```

```
%
% Simulate the System (Forward Time Solution)
%
for (i=1:kmax-1)
                ki=1/(Vc*(Tf-time(i)));
A=[  0     1     0     0     0      0     0     0   -VmHE  0     0
     0     0     1     0     0      0     0     0     0    0     0
     0     0    -a     0  a*N*Vc   0     0     0     0    0     0
     0     0     0     0     1      0     0     0     0    0     0
   -c*ki   0     0    -c    -b    c*ki   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      0     0     0     0    0     0
     0     0     0     0     0      0     0     0     0    0     0
     0     0     0     0     0      0     0     0     0    0     1
     0     0     0     0     0      0     0   ATGT    0   -W     0];
    [Phifwd,Delfwd]=c2d(A,B,dt);
    x(:,i+1) = Phifwd*x(:,i) + Delfwd*impulse(i);
    y(:,i+1) = C*x(:,i+1);
    time(i+1)=time(i)+dt;
 end;
 y(i)
 %
 % Plot Results
 %
 plot(time(1:i),y(1,(1:i)));title('Forward Time - 6 g Barrel Roll');
 xlabel('Time');,ylabel('y miss');grid;pause;
 %
 % Reinitialize variables and vectors
 %
 dt=0.01;
 kmax=Tf/dt+1;
 impulse=zeros(1,kmax);
 impulse(1)=1/dt;
```

```
x=zeros(11,kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
time(1)=dt;
%
%  Simulate the System (Adjoint Solution)
%
for (i=1:kmax-1)
               ki=1/(Vc*(time(i)));
A=[  0     1     0     0     0       0     0     0    -VmHE 0     0
     0     0     1     0     0       0     0     0     0    0     0
     0     0    -a     0  a*N*Vc     0     0     0     0    0     0
     0     0     0     0     1       0     0     0     0    0     0
   -c*ki   0     0    -c    -b     c*ki    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       0     0     0     0    0     0
     0     0     0     0     0       0     0     0     0    0     0
     0     0     0     0     0       0     0     0     0    0     1
     0     0     0     0     0       0     0   ATGT    0   -W     0];
    [Phiadj,Deladj]=c2d(A',C',dt);
    x(:,i+1) = Phiadj*x(:,i) + Deladj*impulse(i);
    y(:,i+1) = B'*x(:,i+1);
    time(i+1)=time(i)+dt;
end;
y(i)
%
% Plot Results
%
plot(time(1:i),y(1,(1:i)));title('Adjoint - 6 g Barrel Roll');
xlabel('Time');,ylabel('y miss');grid;pause;
end;
```

```
%
% Lukenbill, F. C. 21 November 1990
% This program simulates the forward time and adjoint models with a
% Spilt 'S' maneuver
%
% Initialize Variables
%
tm=1.0;
tsh=0.1;
a=tm;
b=2*(1/tsh);
c=(1/tsh)*(1/tsh);
N=4.0;
Vc=2500;
Tf=6;
ki=1/(Vc*Tf);
HE=0.0*pi/180;
Vm=2000.0;
VmHE=Vm*sin(HE);
Atgt=193.2;
L=1.0;
a1=pi^2/L^2;
a2=pi*16*Atgt^2/(L*pi^2*Tf);
a3=9*a1;
a4=a2;
%
% Initialize State Matricies
%
A=[  0      1     0     0     0      0     0     0   -VmHE 0     000
     0      0     1     0     0      0     0     0     0   0     000
     0      0    -a     0  a*N*Vc    0     0     0     0   0     000
     0      0     0     0     1      0     0     0     0   0     000
   -c*ki    0     0    -c    -b    c*ki   0     0     0   0     000
     0      0     0     0     0      0     1     0     0   0     000
     0      0    0    0    0    0    0    0    0    1    0 1 0
     0      0    0    0    0    0    0    0    0    0    0 0 0
     0      0    0    0    0    0    0    0    0    0    0 0 0
     0      0    0    0    0    0    0    0    0    0    1 0 0
     0      0    0    0    0    0    0    a2   0   -a1   0 0 0
     0      0    0    0    0    0    0    0    0    0    0 0 1
     0      0    0    0    0    0    0    a4   0    0    0 -a3 0];
%
B=[0
   0
   0
   0
   0
   0
   0
   1
   1
   0
   0
   0
   0];
%
C=[-1  0 0 0 0 1 0  0  0  0  0 0 0];
%
% Input Timing Information
%
dt=0.01;
kmax=Tf/dt+1;
%
% Set up impulse function
%
impulse=zeros(1,kmax);
impulse(1)=1/dt;
%
```

```
% Initialize vectors
%
x=zeros(13,kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
%
% Simulate the System (Forward Time Solution)
%
for (i=1:kmax-1)
            ki=1/(Vc*(Tf-time(i)));
A=[  0      1     0     0     0       0     0     0   -VmHE  0    000
     0      0     1     0     0       0     0     0     0    0    000
     0      0    -a     0   a*N*Vc    0     0     0     0    0    000
     0      0     0     0     1       0     0     0     0    0    000
   -c*ki    0     0    -c    -b     c*ki    0     0     0    0    000
     0      0     0     0     0       0     1     0     0    0    000
     0      0     0     0     0       0     0     0     0    1    010
     0      0     0     0     0       0     0     0     0    0    000
     0      0     0     0     0       0     0     0     0    0    000
     0      0     0     0     0       0     0     0     0    0    100
     0      0     0     0     0       0     0    a2     0   -a1   000
     0      0     0     0     0       0     0     0     0    0    001
     0      0     0     0     0       0     0    a4     0    0   0 -a3 0];
  [Phifwd,Delfwd]=c2d(A,B,dt);
  x(:,i+1) = Phifwd*x(:,i) + Delfwd*impulse(i);
  y(:,i+1) = C*x(:,i+1);
  time(i+1)=time(i)+dt;
end;
y(i)
%
% Plot Results
%
plot(time(1:i),y(1,(1:i)));title('Forward Time - 6 g Split S');
xlabel('Time');,ylabel('y miss');grid;pause;
```

```
%
% Reinitialize variables and vectors
%
dt=0.01;
kmax=Tf/dt+1;
impulse=zeros(1,kmax);
impulse(1)=1/dt;
x=zeros(13,kmax);
y=zeros(1,kmax);
time=zeros(1,kmax);
time(1)=dt;
%
% Simulate the System (Adjoint Solution)
%
for (i=1:kmax-1)
                ki=1/(Vc*(time(i)));
A=[  0      1      0      0      0        0      0      0    -VmHE  0      0 0 0
     0      0      1      0      0        0      0      0      0    0      0 0 0
     0      0     -a      0   a*N*Vc      0      0      0      0    0      0 0 0
     0      0      0      0      1        0      0      0      0    0      0 0 0
  -c*ki     0      0     -c     -b      c*ki     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      0      0    1      0 1 0
     0      0      0      0      0        0      0      0      0    0      0 0 0
     0      0      0      0      0        0      0      0      0    0      0 0 0
     0      0      0      0      0        0      0      0      0    0      1 0 0
     0      0      0      0      0        0      0     a2      0   -a1     0 0 0
     0      0      0      0      0        0      0      0      0    0      0 0 1
     0      0      0      0      0        0      0     a4      0    0    0 -a3 0];
    [Phiadj,Deladj]=c2d(A',C',dt);
    x(:,i+1) = Phiadj*x(:,i) + Deladj*impulse(i);
    y(:,i+1) = B'*x(:,i+1);
    time(i+1)=time(i)+dt;
end;
```

```
y(i)
%
% Plot Results
%
plot(time(1:i),y(1,(1:i)));title('Adjoint - 6 g Split S');
xlabel('Time');,ylabel('y miss');grid;pause;
end;
```

APPENDIX C-DISSPLA PLOTTING PROGRAM

```
      CC Lukenbill, F. C. 6/15/90
      CC This code is used to generate the three dimensional plots
      CC
      DIMENSION MSLX(300), MSLY(300),MSLZ(300),
      TGTX(300),TGTY(300),TGTZ(300)
      NUM=100
      DO 10 I=1,NUM
      READ (10,1000) MSLX(I),MSLY(I),MSLZ(I)
      READ (11,1000) TGTX(I),TGTY(I),TGTZ(I)
10     CONTINUE
1000  FORMAT (3(1X,IE15.7))
      CALL COMPRS
      CALL AREA2D(8.,8.5)
      CALL X3NAME('X AXIS$',100)
      CALL Y3NAME('YAXIS$',100)
      CALL Z3NAME('Z  AXIS$',100)
      CALL VOLM3D(7.,7.,7.)
      CALL VIEW(80000.,24000.,2000.)
      CALL GRAF3D(0.,4000.,24000.,0.,4000.,24000.,0.,400.,2400.)
      CALL GRFITI(0.,0.,0.,0.,1.,0.,0.,1.,1.)
      CALL AREA2D(7.,7.)
      CALL GRAF(0.,1.,6.,0.,1.,6.)
      CALL GRID(2,2)
      CALL END3GR(0)
      CALL GRFITI(0.,0.,0.,0.,1.,0.,1.,1.,0.)
      CALL AREA2D(7.,7.)
      CALL GRAF(0.,4000.,24000,0.,4000.,24000.)
      CALL DASH
      CALL CURVE (MSLY,MSLX,NUM,0)
CALL RESET('DASH)
CALL CURVE (TGTY,TGTX,NUM,0)
CALL GRID(2,2)
CALL END3GR(0)
CALL DASH
CALL CURV3D(MSLX,MSLY,MSLX,NUM,0)
CALL RESET('DASH')
CALL CURV3D(TGTX,TGTY,TGTX,NUM,0)
CALL ENDPL(0)
CALL DONEPL
STOP
END
```

REFERENCES

1. Ball, R. E., *The Fundamentals of Aircraft Combat Survivability Analysis and Design*, pp. 1-4, American Institute of Aeronautics and Astronautics, Inc., 1985.

2. IIT Research Institute Report GACIAC HB-78-01, *Guidance Law Handbook for Classical Proportional Navigation*, by L. D. Paarmann, J. N. Faraone, and C. W. Smoots, pp. 5-9, 1 June 1978.

3. Nescine, F. W., and Zarchan, P., "Classical Versus Modern Homing Missile Guidance", *Advisory Group for Aerospace Research and Development (AGARD)* No. 292, pp. 22.1-22.2, 1980.

4. Wilcox, H. A., Video Taped Lecture Series, *Development Philosophy for Air-to-Air Guided Missiles*, Naval Weapons Center, China Lake, CA.

5. Gonzalez, J., "New Methods in the Terminal Guidance and Control of Tactical Missiles", *Advisory Group for Aerospace Research and Development (AGARD)* No. 101, pp. 3.1-3.4, 1979.

6. Zarchan, P., *Tactical and Strategic Missile Guidance*, pp. 37-44, American Institute of Aeronautics and Astronautics, Inc., 1990.

7. Zarchan, P., "Representation of Realistic Evasive Maneuvers by the Use of Shaping Filters", *Journal of Guidance and Control*, Vol. 2, July-August 1979, pp. 290-295.

INITIAL DISTRIBUTION LIST

No. Copies

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

2. Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943-5102

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

4. Professor H. A. Titus, Code EC/TS 7
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

5. Commander J. R. Powell 1
Executive Officer
VAQ-33
NAS Key West
Key West, FL 33040

6. Lieutenant F. C. Lukenbill 1
Operations Department
USS Forrestal (CV-59)
FPO Miami 34080

7. Lieutenant T. L. Mascolo 1
SATD/Flight Systems
Patuxent River, MD 20607