NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

OPTIMAL DIGITAL CONTROL
OF
A BANK-TO-TURN MISSILE

by

Carlos A. L. Velloso

March 1984

Thesis Advisor: Daniel J. Collins

Approved for public release; distribution unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Optimal Digital Control of a Bank-to-Turn Missile		5. TYPE OF REPORT & PERIOD COVERED Master Thesis March 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carlos Augusto Leal Velloso		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1984
		13. NUMBER OF PAGES 154
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Guidance and Control
Optimal Digital Control
Terminal Guidance Law
Bank-to-Turn Missile

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This work adresses the application of digital optimum control theory to a bank-to-turn missile.

A optimal guidance law has been developed and tested in several scenarios using a 2-D model. Effects of sample rate, pitch angle, gravity and approximations for small and large roll excursions are discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Optimal Digital Control of a Bank-to-Turn Missile

by

Carlos A.L. Velloso
Major, Brazilian Air Force
B.S., Instituto Tecnologico de Aeronautica, Brasil, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 1984

ABSTRACT

This work adresses the application of digital optimum control theory to a bank-to-turn missile.

A optimal guidance law has been developed and tested in several scenarios, using a 2-D model. Effects of sample rate, pitch angle, gravity and approximations for small and large roll excursions are discussed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

To my wife, Rosy, for her constant encouragment, support and love.

To my children, Ana, Antonio, Chica, Joao and Rita, for yours smiles and love.

Without you this work would be impossible.

I. INTRODUCTION

Because of threats from highly maneuverable high performance aircrafts and the need for increase standoff ranges, major improvements are needed in guidance and control capabilities of missiles.

The high maneuverability of targets, has led to defense missiles capable of develop higher lift accelerations and to more complex control laws, able to improve performance over well know laws as proportional navegation.

In order to accomplish these new requirements with large standoff ranges, propulsion systems using airbreathing engines has been studied and developed in recent years.

The advent of airbreathing engines leads natural to a consideration of bank-to-turn missiles in order to minimize the angle of attack of the inlets.

The necessity of more complex control laws, leads in a general way, to the application of modern control and estimation theory, since more complete informations of the states of missile and target are necessary than those states informed by sensors commonly in use in missiles today. This leads to the use of a airborne computer.

The present work adresses the design and evaluation of a optimal digital control for application to terminal guidance in a bank-to-turn missiles.

One continuous two dimensional model was adopted, in the following form:

$$\dot{X}(t) = A(t)\ x(t) + B(t)\ u(t) + E\ g \qquad (1.1)$$

where the effect of gravity appears explicitly in the third term on the righ hand side of expression 1.1.

After the development of a equivalent discrete model, the optimal control problem has been solved, using a modified Ricatti equation due to the existence of the third term representing to the gravity effect.

Next, several analysis has been made in order to check the effects of small and large roll excursions, the effect of the sample rate on the system, and the effect of the initial pitch angle, in order to check the validity of such two dimensional model, when applied in some scenarios of interst.

II. MODEL OF THE SYSTEM

A. INTRODUCTION

In the present work the problem of terminal guidance for long range, bank-to-turn missiles with ramjet engines,using a digitalized system has been investigated.

The model developed in reference 1 is used as the base for this work. After the digitalization of that model, an optimal control law was developed.

B. ASSUMPTIONS

Keeping the same assumptions as in ref.1,one has:

The missil is limited to -2g's and +15 g's of commanded acceleration in the pitch plane,with zero lag.Also its yaw auto pilot has zero lag, yaw regulator maintains zero sideslip.

Missile thrust exactly cancels drag.

The angle of attack is assumed to be very small,which leads to the commanded acceleration acting normal to the velocity vector.

The missile will not have to roll through a large angle.(Further considerations will give to this at the end of the derivation of the control law).

C. THE CONTINUOUS MODEL

Using the same reference frames as in ref.1,one assumes:

-Body frame with x_b axis parallel to the longitudinal missile axis, positive y_b axis out of the left wing ,and positive z_b axis upward.(see fig.2.1)

-Fligth path axis with x_F axis parallel to the velocity vector, positive z_F axis pointing upwards and y_F axis pointing to the left. (see fig. 2.2)

In fig 2.1 and 2.2, the angles ϕ and θ are the Eulerian roll and pitch angles.

The state vector is given as

$$\underline{x} = \left[y_F \quad \dot{y}_F \quad Aty \quad z_F \quad \dot{z}_F \quad Atz \quad \Delta\phi \right]^T \qquad (2.1)$$

where y_F and z_F are the components of the relative target position, $\dot{y}_F$ and $\dot{z}_F$ are the relative target velocity, Aty and Atz are the components of target acceleration, which is exponentially decaying with a time constant τ.

$$\Delta\phi = \phi - \phi_0 \qquad (2.2)$$

where ϕ_0 is the initial roll angle (at t=0).

The control vector is given as:

$$\underline{u} = \left[Ac \quad Pc \right]^T \qquad (2.3)$$

where Ac is the commanded acceleration and Pc is the commanded roll rate.

The nonlinear plant equation is

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) + \underline{G} \qquad (2.4)$$

or

$$\dot{\underline{X}} = \begin{bmatrix} \dot{y}_F \\ Aty + Ac\sin\phi \\ -Aty/\sigma \\ \dot{z}_F \\ Atz - Ac\cos\phi \\ -Atz/\sigma \\ P_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -g\cos\theta \\ 0 \\ 0 \end{bmatrix} \tag{2.5}$$

where g is gravity's acceleration and θ is the pitch angle as seen in fig.2.2

Linearizing and setting

$$\underline{G} = \underline{E}\ \underline{g} \tag{2.6}$$

one has

$$\dot{x} = \begin{bmatrix} \dot{y}_F \\ Aty + A'_c(\cos\phi_0)\Delta\phi \\ -Aty/\sigma \\ \dot{z}_F \\ Atz + A'_c(\cos\phi_0)\Delta\phi \\ -Atz/\sigma \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \sin\phi_0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\cos\phi_0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.7}$$

where

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & -\cos\theta & 0 & 0 \end{bmatrix}^T \qquad (2.8)$$

in eqn. 2.7, we have set

$$\cos\phi = \cos(\phi_o + \Delta\phi) = \cos\phi_o \cos\Delta\phi - \sin\phi_o \sin\Delta\phi$$

$$\sin\phi = \sin(\phi_o + \Delta\phi) = \sin\phi_o \cos\Delta\phi + \cos\phi_o \sin\Delta\phi$$

and expanded in $\Delta\phi$ which is considered small.

Now assuming that $A'c$, which is actually the desired control Ac, can be expressed in the form of:

$$A'c = Aco \left[1 - \frac{t}{Ti} \right] \qquad (2.9)$$

with Aco = Ac at t=0, one has

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u} + \underline{E}\,\underline{g} \qquad (2.10)$$

where

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & Aco\left[1-\frac{t}{Ti}\right]\cos\phi_o \\ 0 & 0 & -1/\tau & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & Aco\left[1-\frac{t}{Ti}\right]\sin\phi_o \\ 0 & 0 & 0 & 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2.11)$$

(2.12)

$$\underline{B} = \begin{bmatrix} 0 & 0 \\ \sin \phi_0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\cos \phi_0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\underline{E} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\cos\theta & 0 & 0 \end{bmatrix}^T \qquad (2.13)$$

$$\underline{g} = g \qquad (2.14)$$

D. THE DISCRETE MODEL

1. Introduction

With the introduction of a digital computer to control the continuous-time system, one has to have some kind of interface in order to take care of the communication between the discrete and continuous-time systems. In this case it will be considered, A-to-D and D-to-A converters as samplers and zero-order holders as in reference 2.

In such case, considering the system:

$$\dot{\underline{x}}(t) = \underline{A}(t)\ \underline{x}(t) + \underline{B}(t)\ \underline{u}(t) + \underline{E}(t)\ \underline{g}(t) \qquad (2.15)$$

one can write the state of the system at time t(k+1) as :

$$x(t_{k+1}) = (t_{k+1}, t_k)\ x(t_k) + \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \eta) B(\eta)\ d\eta\ u(t_k) \qquad (2.16)$$

$$+ \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \eta)\ E(\eta)\ d\eta\ g(t_k)$$

where $\phi(t, to)$ is the transiction matrix of the system represented by eqn. 2.13.

Furthermore, we will considered that the sampling instants are equally spaced, or:

$$t_{k+1} - t_k = T \qquad (2.17)$$

$$t_{k+1} = kT + T \qquad (2.18)$$

so one can replace

$t_k = kT$

thus,

$$x(kT+T) = \phi(kT+T)\ x(kT) \qquad (2.19)$$

$$+ \int_{kT}^{kT+T} \phi(kT+T, \eta)\ B(\eta)\ d\eta\ u(kT)$$

$$+ \int_{kT}^{kT+T} \phi(kT+T, \eta)\ E(\eta)\ d\eta\ g(kT)$$

or in a simplified notation:

$$x(k+1) = A_d(k)\ x(k) + B_d(k)\ u(k) + E_d(k)\ g(k) \tag{2.20}$$

where,

$$A_d(k) = \phi(kT+T, kT) \tag{2.21}$$

$$B_d(k) = \int_{kT}^{kT+T} \phi(kT+T, \eta)\ B(\eta)\ d\eta \tag{2.22}$$

$$E_d(k) = \int_{kT}^{kT+T} \phi(kT+T, \eta)\ B(\eta)\ d\eta \tag{2.23}$$

2. Calculation of the Matrices A(k), B(k) and E(k)

It is straightforward to show, using the sparseness of the matrix A(t), that the transiction matrix of equation 2-19 is :

$$\phi(kT+T) = \begin{bmatrix} 1 & T & Ad_{1,3} & 0 & 0 & 0 & Ad_{1,7} \\ 0 & 1 & Ad_{2,3} & 0 & 0 & 0 & Ad_{2,7} \\ 0 & 0 & e^{-T/\sigma} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & Ad_{4,6} & Ad_{4,7} \\ 0 & 0 & 0 & 0 & 1 & Ad_{5,6} & Ad_{5,7} \\ 0 & 0 & 0 & 0 & 0 & e^{-T/\sigma} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

where:

$$A_{d_{1,3}} = A_{d_{4,6}} = \tau T - \tau^2 (1-e^{-T/\tau})$$

$$A_{d_{2,3}} = A_{d_{5,6}} = (1-e^{-T/\tau})$$

For the calculation of the others terms one may make use of the property of the transiction matrix that:

$$\frac{d\phi(t_2,t_1)}{dt_2} = A(t_2)\, A_d(t_2,t_1)$$

so,

$$\frac{dA_d(kT+T,kT)}{d(kT+T)} = A(kT+T)\, A_d(kT+T,kT)$$

For A (2,7):

$$\frac{dA_{d_{2,7}}(kT+T,kT)}{d(kT+T)} = A_{2,7}(kT+T) = Aco\left[\left[1-\frac{t}{Ti}\right]\cos\phi_o\right]_{kT+T}$$

$$\frac{dA_{d_{2,7}}(kT+T,kT)}{d(kT+T)} = Aco\left[1-\frac{kT+T}{Ti}\right]\cos\phi_o$$

$$A_{d_{2,7}} = Aco\cos\phi_o\int_{kT}^{kT+T}\left[1-\frac{kT+T}{Ti}\right] d(kT+T)$$

$$A_{d_{2,7}} = Aco\cos\phi_o\left[T-\left[\frac{2k+1}{2Ti}\right]T^2\right]$$

For A (1,7) :

$$\frac{dA_{d_{1,7}}(kT+T,kT)}{d(kT+T)} = A_{d_{2,7}}(kT+T,kT)$$

$$A_{d_{1,7}}(kT+T,kT) = \int_{kT}^{kT+T} A_{d_{2,7}}\, d(kT+T)$$

$$Ad_{1,7} = \left[T^2 - \left[\frac{2k+1}{2Ti} \right] T^3 \right] Aco \cos\phi_0$$

Doing the same process for Ad(5,7) and Ad(4,7) one has:

$$Ad_{4,7} = Aco \sin\phi_0 \left[T^2 - \left[\frac{2k+1}{2Ti} \right] T^3 \right]$$

$$A = Aco \sin\phi_0 \left[T - \left[\frac{2k+1}{2Ti} \right] T^2 \right]$$

For the derivation of the matrix $B_d(k)$ one needs according to eqn. 2-22

$$\phi(kT+T)\, B(\eta) = A_d(kT+T,\eta)\, B(\eta)$$

where,

$$A_d(kT+T,\eta) = \begin{bmatrix} 1 & A\eta_{1,2} & A\eta_{1,3} & 0 & 0 & 0 & A\eta_{1,7} \\ 0 & 1 & A\eta_{2,3} & 0 & 0 & 0 & A\eta_{2,7} \\ 0 & 0 & A\eta_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & A\eta_{4,5} & A\eta_{4,6} & A\eta_{4,7} \\ 0 & 0 & 0 & 0 & 1 & A\eta_{5,6} & A\eta_{5,7} \\ 0 & 0 & 0 & 0 & 0 & A\eta_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $A\eta$ represents $A_d(kT+T,\eta)$, and

$$B(\eta) = \begin{bmatrix} 0 & 0 \\ \sin\phi_0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\cos\phi_0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

thus

$$A_d(kT+T,\eta)\,B(\eta) = \begin{bmatrix} A\eta_{1,2}\sin\phi_0 & A\eta_{1,7} \\ \sin\phi_0 & A\eta_{2,7} \\ 0 & 0 \\ -A\eta_{4,5}\cos\phi_0 & A\eta_{4,7} \\ -\cos\phi_0 & A\eta_{5,7} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where

$$A\eta_{1,2} = kT+T-\eta = A\eta_{4,5}$$

$$A\eta_{2,7} = \left[kT+T-\frac{(kT+T)^2}{2\,Ti} - \eta + \frac{\eta^2}{2\,Ti} \right] Aco\,\cos\phi_0$$

$$A\eta_{1,7} = \left[T(kT+T) - \left[\frac{2k+1}{2\,Ti}\right](kT+T)\,T^2 - T\eta + \left[\frac{2k+1}{2\,Ti}\right] T^2\eta \right] Aco\,\cos\phi_0$$

$$A\eta_{4,7} = \left[T(kT+T) - \left[\frac{2k+1}{2\,Ti}\right](kT+T)\,T^2 - T\eta + \left[\frac{2k+1}{2\,Ti}\right] T^2\eta \right] Aco\,\sin\phi_0$$

$$A\eta_{5,7} = \left[kT+T-\frac{(kT+T)^2}{2\,Ti} - \eta + \frac{\eta^2}{2\,Ti} \right] Aco\,\sin\phi_0$$

and for $B_d(k)$:

$$B_d = \begin{bmatrix} B_{d_{1,1}} & B_{d_{1,2}} \\ \sin\phi_0 & B_{d_{2,2}} \\ 0 & 0 \\ B_{d_{4,1}} & B_{d_{4,2}} \\ -\cos\phi_0 & B_{d_{5,2}} \\ 0 & 0 \\ 0 & T \end{bmatrix} \qquad (2.25)$$

where

$$B_{d_{1,1}} = \int_{kT}^{kT+T} (kT+T-\eta)\, d\eta \ \sin\phi_0 = \frac{T^2}{2} \sin\phi_0$$

$$B_{d_{4,1}} = -\frac{T^2}{2} \cos\phi_0$$

$$B_{d_{2,2}} = \int_{kT}^{kT+T} A\eta_{2,7}\, d\eta$$

which after some algebric work has been found

$$B\ (k) = \left[\frac{T^2}{2} - \frac{(kT+T)^3}{3Ti} - \frac{(kT+T)^2\, kT}{2\,Ti} - \frac{(kT)^2}{6Ti} \right] Acc \cos\phi_0$$

and

$$B_{d_{1,2}}(k) = \int_{kT}^{kT+T} A\eta_{1,7}\, d\eta$$

which can be found to be:

$$B_{d_{1,2}}(k) = \left[\frac{T^3}{2} - \frac{1}{2} \left[\frac{2k+1}{2T_i} \right] T^4 \right] A_{co} \cos \phi_o$$

thus

$$B_{d_{4,2}}(k) = \left[\frac{T^3}{2} - \frac{1}{2} \left[\frac{2k+1}{2T_i} \right] T^4 \right] A_{co} \sin \phi_o$$

$$B_{d_{5,2}}(k) = \left[\frac{T^2}{2} - \frac{(kT+T)^3}{3T_i} + \frac{(kT+T)^2}{2T_i} kT - \frac{(kT)^3}{6T_i} \right] A_{co} \sin \phi_o$$

In the same way:

$$\phi(kT+T, \eta)\, E(\eta) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -T\cos\theta \\ -\cos\theta \\ 0 \\ 0 \end{bmatrix}$$

thus $\underline{E}$ (k) is equal to

$$\underline{E} = \left[0 \quad 0 \quad 0 \quad 0 \quad -\frac{T^2}{2}\cos\theta \quad -T\cos\theta \quad 0 \quad 0 \right]$$

where we have considered θ as a constant angle. (Further comments on this after development of the control law).

Notice that througout this work ,the commanded acceleration has been considered an unknown and the assumption has been made that it will be in the form of eqn.2.9. One will need in further developments to consider the control Ac as a known Ac(k) ,which will be a constant between kT and kT+T. With such assumptions, the discrete representation of the system is easily found to be :

$$\underline{x}(k+1) = \underline{A}(k)\ \underline{x}(k) + \underline{B}(k)\ \underline{u}(k) + \underline{E}\ g \tag{2.26}$$

where

$$A_d = \begin{bmatrix} 1 & T & Ad_{1,3} & 0 & 0 & 0 & A_{d_{1,7}} \\ 0 & 1 & Ad_{2,3} & 0 & 0 & 0 & Ad_{2,7} \\ 0 & 0 & e^{-T/\tau} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & Ad_{4,6} & Ad_{4,7} \\ 0 & 0 & 0 & 0 & 1 & Ad_{5,6} & Ad_{5,7} \\ 0 & 0 & 0 & 0 & 0 & e^{-T/\tau} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

with

$$Ad_{1,3} = \tau T - \tau^2 \ (1 - e^{-T/\tau}) = Ad_{4,6}$$

$$Ad_{2,3} = \tau(1 - e^{-T/\tau}) = Ad_{5,6}$$

$$Ad_{1,7} = \frac{T^2}{2} \ Ac \cos \phi_o$$

$$Ad_{2,7} = T \ Ac \cos \phi_o$$

$$Ad_{4,7} = \frac{T^2}{2} \ Ac \sin \phi_o$$

$$Ad_{5,7} = T \ Ac \sin \phi_o$$

and

$$B_d(k) = \begin{bmatrix} \frac{T^2}{2}\sin\phi_o & \frac{T^3}{6} A_c \cos\phi_o \\ T \sin\phi_o & \frac{T^2}{2} A_c \cos\phi_o \\ 0 & 0 \\ -\frac{T^2}{2}\cos\phi_o & \frac{T^3}{6} A_c \sin\phi_o \\ -T\cos\phi_o & \frac{T^2}{2} A_c \sin\phi_o \\ 0 & 0 \\ 0 & T \end{bmatrix}$$

and

$$E_d(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{T^2}{2}\cos\theta \\ -T\cos\theta \\ 0 \\ 0 \end{bmatrix}$$

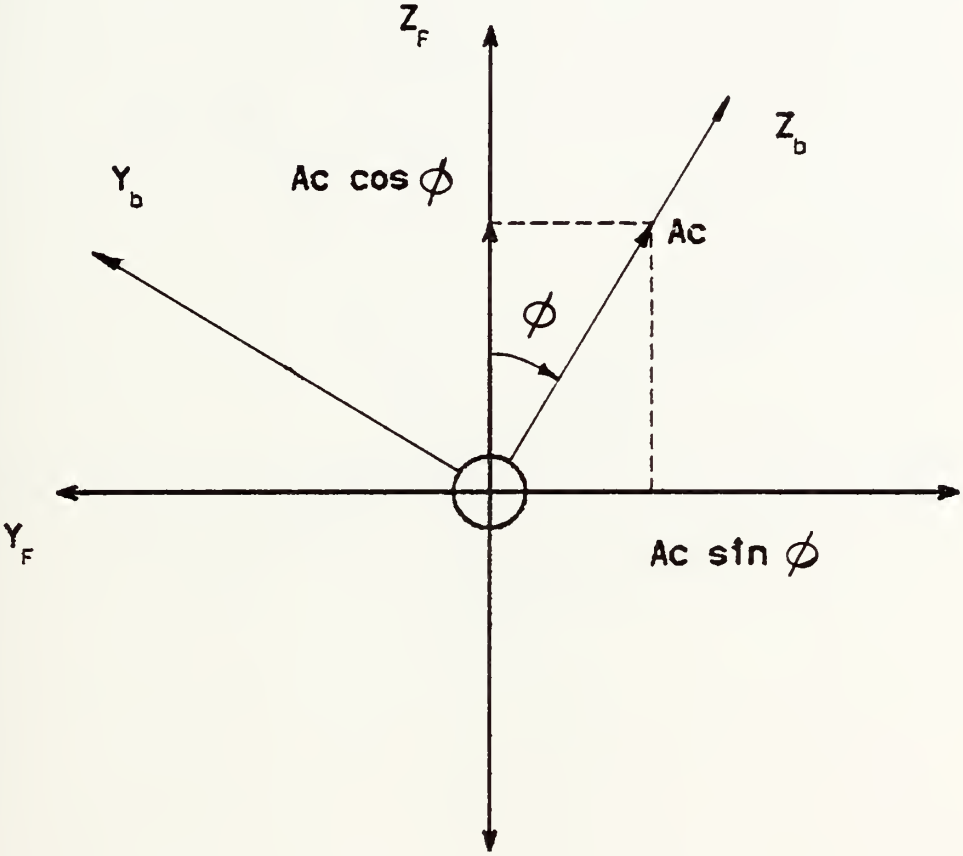

Figure 2.1 Reference Frames.

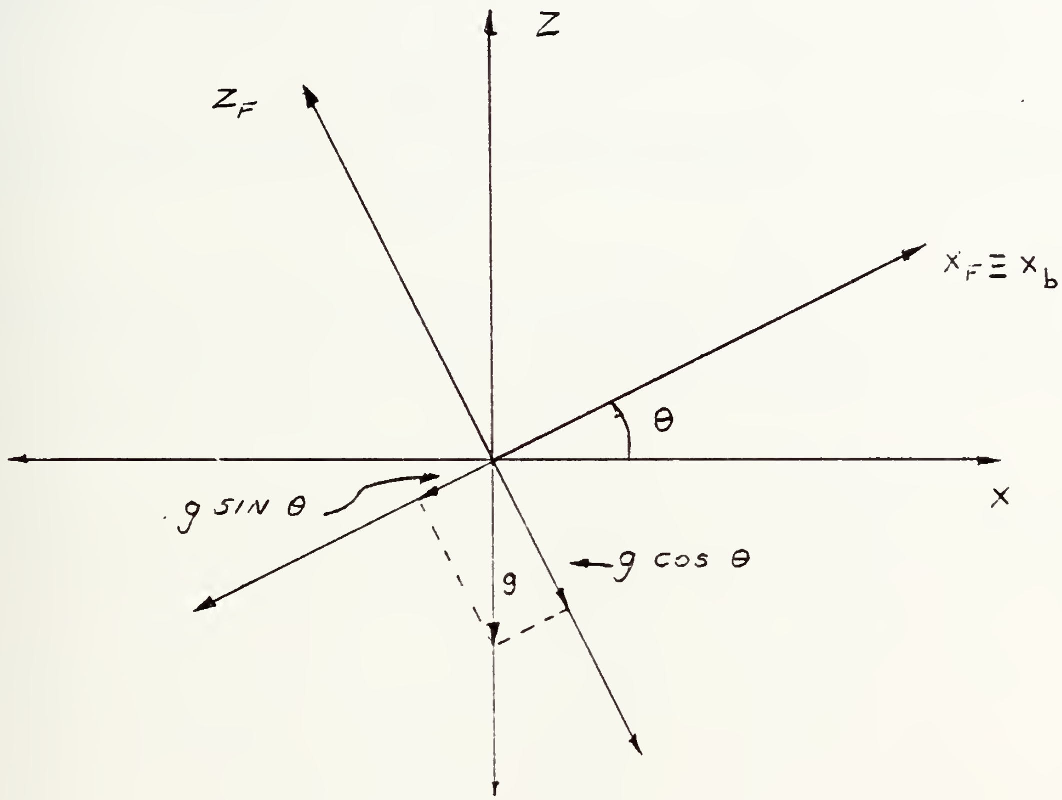

Figure 2.2 Reference Frames.

III. THE OPTIMAL CONTROLLER

A. DERIVATION OF THE OPTIMAL CONTROLLER

In order to have a suitable guidance law to implement the control commands, we will minimizing the following performance index:

$$J = \frac{1}{2} x^T(n) W(n) x(n) + \sum_{k=0}^{N-1} \frac{1}{2} u^T(k) Q(k) u(k) \qquad (3.1)$$

where x(N) is the final state at t=Tf

As we want to minimize the final miss distance, the weighting matrix W(N) is taken as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.2)$$

and Q(k) is a two by two positive definite symmetric weighting matrix to be chosen.

In the derivation of the solution, reference 3 has been followed keeping in mind that the state equation has the form:

$$\underline{x}(k+1) = \underline{A}(k)\ \underline{x}(k) + \underline{B}(k)\ \underline{u}(k) + \underline{E}\ g \tag{3.3}$$

or

$$\underline{x}(k+1) = f\ (\underline{x}(k), \underline{u}(k), g) \tag{3.4}$$

where the third term,which represents the effect of the gravity has been considered constant.

Considering that the performance index is in the form:

$$J = \phi[x(N)] + \sum_{k=0}^{N-1} L(k)\left[x(k), u(k), g(k)\right] \tag{3.5}$$

we need to find a sequence of u(k) that minimizes J.

Adjoin the system equation to J with a multiplier $\lambda(n)$

$$J = \phi\left[x(N)\right] + \sum_{k=0}^{N-1}\left\{ L(k)\left[x(k), u(k), g\right] + \right. \tag{3.6}$$

$$\left. + \lambda^T(k+1)\ \left\{ f_K\left[x(k), u(k), g\right] - x(k+1)\right\}\right\}$$

and defining a scalar sequence H(k)

$$H(k) = L(k)\left[x(k), u(k), g\right] + \tag{3.7}$$

$$+ \lambda^T(k+1)\ f_K\left[x(k), u(k), g\right]$$

$$k=0,1,2,\ldots. n-1$$

one has:

$$J = \phi\left[x(N)\right] - \lambda^T(N)\ x(N) + \tag{3.8}$$

$$+ \sum_{k=1}^{N-1}\left[H(k) - \lambda^T(k)\ x(k)\ \right] + H(0)$$

Considering differential changes in J:

$$dJ = \left[\ \frac{\partial \phi}{\partial x(N)} - \lambda^T(N)\right]\ dx(N) + \tag{3.9}$$

$$+ \sum_{k=1}^{N-1}\left\{\ \left[\frac{\partial H(k)}{\partial x(k)} - \lambda^T(k)\right] dx(k) + \right.$$

$$\left. + \frac{\partial H(k)}{\partial u(k)}\ du(k)\right\} + \frac{\partial H(0)}{\partial x(0)}\ dx(k) +$$

$$+ \frac{\partial H(0)}{\partial u(0)}\ du(0)$$

choosing the multiplier $\lambda(k)$ so that

$$\lambda^T(k) - \frac{\partial H(k)}{\partial x(k)} = 0 \tag{3.10}$$

thus

$$\frac{\partial L(k)}{\partial x(k)} + \lambda^T(k+1)\ \frac{\partial f_k}{\partial x(k)} = \lambda^T(k)$$

and

$$\frac{\partial H(k)}{\partial u(k)} = 0 \qquad (3.11)$$

with boundary condition

$$\lambda^T(N) = \frac{\partial \phi}{\partial x(N)} \qquad (3.12)$$

we obtain the minimization of the performance index.

In the present case we have:

$$J = \frac{1}{2} x^T(N)\, W(N)\, x(N) + \sum_{k=0}^{N-1} u^T(k)\, Q(k)\, u(k) \qquad (3.13)$$

$$+ \lambda^T(k+1) \left[A(k)\, x(k) + B(k)\, u(k) + E\, G - x(k+1) \right]$$

and H(k):

$$H(k) = \frac{1}{2} u^T(k)\, Q(k)\, u(k) + \qquad (3.14)$$

$$+ \lambda^T(k+1) \left[A(k)\, x(k) + B(k)\, u(k) + E\, g \right]$$

then in order to minimize J:

$$\frac{\partial H(k)}{\partial u(k)} = u^T(k)\, Q(k) + \lambda^T(k+1)\, B(k) = 0 \qquad (3.15)$$

or considering that $Q^T(k)=Q(k)$

$$Q(k)\ u(k) = -\ B^T(k)\ \lambda\ (k+1)$$

and

$$\lambda^T(k) = \frac{\partial H(k)}{\partial x(k)} \qquad (3.16)$$

$$\lambda^T(k) = \lambda^T(k+1)\ A(k)$$

and

$$\lambda^T(N) = x^T(N)\ W(N) \qquad (3.17)$$

Notice that we are not weighting the states in the performance index, except the last state.A more general form could be obtained, with all states being weigthing ,if we change the eqn 3.16 to:

$$\lambda^T(k) = \lambda^T(k+1)\ A(k) + x^T(k)\ W1(k) \qquad (3.18)$$

where W1(k) is the weighting matrix of the states.

With equations 3.15,3.16 and 3.17 one is able to find the sequence of u(k) that will give the minima controls.

Such set of equatics can be solved by the sweep method as in ref.2 .

We will look for a solution of the form:

$$u(k)=-F(k)x(k)-FG(k)g(k) \qquad (3.19)$$

what means that the commanded acceleration and roll rate, will have a correction due to the effect of gravity.

Placing:

$$\lambda(k) = S(k)\ x(k) + SG(k)\ g(k) \tag{3.20}$$

from eqn. 3.15

$$Q(k)u(k) = -B^T(k)\left[S(k+1)x(k+1) + SG(k+1)g(k+1)\right]$$

from eqn. 3.3

$$Q(k)\ u(k) = -B^T(k)\ S(k+1)\left[A(k)\ x(k) + \right. \tag{3.21}$$

$$\left. + B(k)\ u(k) + E\ g\right] - B^T(k)\ SG(k+1)\ g(k+1)$$

so

$$\left[Q(k) + B^T(k)\ S(k+1)\ B(k)\right]\ u(k) =$$

$$-B^T(k)\ S(k+1)\ A(k)\ x(k) - B^T(k)\ S(k+1)\ E\ g(k) -$$

$$-B^T(k)\ SG(k+1)\ g(k+1)$$

considering that g is a constant

$$u(k) = -\left[Q(k) + B^T(k)\ S(k+1)\ B\ (k)\right]^{-1} \tag{3.22}$$

$$\Big[B^T(k)\ S(k+1)\ A(k)\ x(K) +$$
$$+ \Big[B^T(k)\ S(k+1)\ E + B^T(k)\ Sg(k+1)\Big]\ g\Big]$$

so:

$$u(k) = -F(k)\,x(k) - FG(k)\,g$$

where

$$F = \Big[Q(k) + B^T(k)\ S(k+1)\ B(k)\Big]^{-1} \cdot \qquad (3.23)$$

$$\cdot\ B^T(k)\ S(k+1)\ A(k)$$

$$FG = \Big[Q(k) + B^T(k)\ S(k+1)\ B(k)\Big]^{-1} \qquad (3.24)$$

$$\Big[B^T(k)\ S(k+1)\ E + B^T Sg(k+1)\Big]$$

from eqn.3.16 and 3.19

$$\lambda(k) = A^T(k)\ \lambda(k+1) = A^T(k)\ \Big[S(k+1)\,x(k+1) + SG(k+1)\,g(k+1)\Big] =$$
$$= A^T(k)\ S(k+1)\ A(k)\ x(k) + A^T(k)\ S(k+1)\ B(k)\ u(k)$$
$$+ A^T(k)\ S(k+1)\ Eg + A^T(k)\ SG(k+1)\ g(k+1)$$

from eqn. 3.19

$$\lambda(k) = A^T(k)\ S(k+1)\ A(k)\ x(k) + A^T(k)\ S(k+1)\ B(k) .$$
$$\cdot \Big[-F(k)\ x(k) - FG(k)\ g\Big] + A^T(k)\ S(k+1)\ Eg +$$
$$+ A\ SG(k)\ g(k+1)$$

so, as g is a constant:

$$S(k)\,x(k) + SG(k)\,g = \Big[A^T(k)\,S(k+1)\,A(k) - A(k)\,S(k+1)\,B(k)\,F(k)\Big] +$$

$$+\left[A^T(k)\ S(k+1)\ E - A^T(k+1)\ B(k)\ FG(k) + A\ (k)\ SG(k+1)\right] g$$

thus

$$S(k)=A^T(k)S(k+1)A(k)-A\ (k)S(k+1)B(k)F(k) \qquad (3.25)$$

and

$$SG(k)=A^T(k)S(k+1)E-A^T(k)S(k+1)B(k)FG(k)+ \qquad (3.26)$$

$$+\ A^T(k)\ SG(k+1)$$

These equations, 3.25, 3.26, 3.23 and 3.24 can be solved backwards with the final condition:

$$S(N)=W(N)$$

$$SG(N)=0$$

Notice that this satisfies our previous boundary condition in eqn. 3.17 where:

$$(N)=W(N)x(N)$$

$$S(N)x(N)=W(N)x(N)$$

so

$$S(N)=W(N)$$

B. EXTENTION OF THE MODEL FOR LARGE ROLL ANGLES

Up to this point, one has to take into account that throughout the development of this work, the angle has been considered small, it is necessary to relax this restriction.

In order to do that, the system has been broken in two blocks as in figure 3.1.

The first block is a representation of the algorithm which will calculate the optimal commands. The algorithm has contain with itself an exact model of the system or missile. The model of the missile is initialized from the information on the initial states, the initial input command Aco and initial roll angle (at t=0); and computes the optimal gains and further the optimal commands which will be feed to the missile.

The method adopted in computing the optimal commands is more easily understood if one considers figure 3.2.

In figure 3.2, the lines numbered as 0 in the graph for Ac and for Pc are the optimal commands for a given initial roll angle(ϕ_0). Lines number 1 are the commands for a second initial roll angle(ϕ_1) larger than ϕ_0, and so on. Thus in figure 3.2 one has a family of optimal commands for any initial roll angle.

Notice that the upper line of the graph of Ac represents the accelerations of a missile which had at t=0 a correct initial roll angle in order to hit the target with no comands in roll rate.

In the present method the computer performs the calculation of the commands only for the first step of time and then feeds these commands to the missile. The missile is then driven to the next state (x(k+1)) and feed-backs to the computer the information on the roll angle at that step. The roll angle feed-backs from the missile is considered by the algoritm as the initial roll angle at t=0 and the next commands are calculated. Notice that at this second step the algorithm will feed to the system the second command (at $t=t_1$). This process is them repeated until t is equal to the intercepted time.

It is important to realize that with this method of calculation, since the algorithm was developed with the assumption of small roll excursions some error is expected due to the fact that in computing the gains by solving a Ricatti equation backwards, as has been done, it is necessary to update the system from t=0 to t=Ti at each step, and in this process the roll angle is not small. Notice however that we are applying the commands only in one step, and if one expects that the roll rate will decrease to zero, as we are increasing in time, the variation of the roll angle will tend to decrease, so, we can expected that the error will decrease as the time increases.

Another important point to be studied is how the missile itself (second block in fig.3.2) has to be implemented in order to be valid for small and large roll excursions.

It is considered that one has the perfect knowledge of the commands, thus the missile is modeled as a state variable system as in eqn 2.26 with the initial roll angle being update at each step. In this way the system will take the initial roll angle as the summation of all previous initial roll angles. (see fig 3.4)

From the original variables one has for the state $\Delta\phi$, the following:

$$\Delta\phi(k+1) = \Delta\phi(k) + Pc(k)\ T \qquad (3.27)$$

which is shcw in figg. 3.3, where the initial roll angle is kept constant and $\Delta\phi$ is update each step.

Notice however that considering large roll excursions (see fig. 3.4) and keeping in mind that the angle ϕ has been defined as

$$\phi = \phi_0 + \Delta\phi$$

the expression 3.27 is not valid, since in modeling the system it was assumed that $\Delta\phi$ would be samll.

This leads to a change in the expression for the variation of $\Delta\phi$ as in fig. 3.4. In fig. 3.4, the angle $\Delta\phi$ is update at each step, so one has:

$$\Delta\phi(k+1) = Pc(k)\ T \tag{3.28}$$

By consideration of the equation 2.27 it can be seen that by setting the element A(7,7) to zero, one can obtaim equation 3.28.

Thus the missile model for large roll excursions will be represented by the following equation.

$$x(k+1) = A(k)\ x(k) + B(k)\ u(k) + E\ g$$

where

$$A(k) = \begin{bmatrix} 1 & T & A_{1,3} & 0 & 0 & 0 & A_{1,7} \\ 0 & 1 & A_{2,3} & 0 & 0 & 0 & A_{2,7} \\ 0 & 0 & e^{-T/\tau} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & A_{4,6} & A_{4,7} \\ 0 & 0 & 0 & 0 & 1 & A_{5,6} & A_{5,7} \\ 0 & 0 & 0 & 0 & 0 & e^{-T/\tau} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{7,7} \end{bmatrix} \tag{3.29}$$

where A(7,7)=0 ,for large roll excursions
A(7,7)=1 ,for small roll excursions

$$B = \begin{bmatrix} \frac{T^2}{2}\sin\phi_0 & \frac{T^3}{6} A_c \cos\phi_0 \\ T\sin\phi_0 & \frac{T^2}{2} A_c \cos\phi_0 \\ 0 & 0 \\ -\frac{T^2}{2}\cos\phi_0 & \frac{T^3}{6} A_c \sin\phi_0 \\ -T\cos\phi_0 & \frac{T^2}{2} A_c \sin\phi_0 \\ 0 & 0 \\ 0 & T \end{bmatrix} \qquad (3.30)$$

$$E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{T^2}{2}\cos\theta \\ -T\cos\theta \\ 0 \\ 0 \end{bmatrix} \qquad (3.31)$$

We will redefine the states x_1, as the relative position, x_2 as the relative velocity in the Y direction and x_3 as the target acceleration in Y direction. The states x_4, x_5 and x_6 has the same meaning, but in the Z direction and

x_7 is $\Delta\phi$. In this representation of the system, $x_1(k)$ is equal to the component of the miss distance along the Y direction and:

$$x_1(k+1) = y_F(k+1) = y_F(k) + \dot{y}_F(k)\ T + \tag{3.32}$$

$$+ \left[\tau T - \tau^2 (1 - e^{-T/\tau})\right] Aty(k) + \frac{T^2}{2} Ac(k) \cos\phi_0\ \Delta\phi(k)$$

$$+ \frac{T^2}{2} \sin\phi_0\ Ac(k) + \frac{T^3}{6} Ac(k)\ \cos\phi_0\ \ Pc(k)$$

The first three terms in the RHS of eqn.3.32, are easily seen as the contribuition to the miss distance of respectively the previous miss distance, relative velocity and target acceleration. The following two terms represents the contribuition of the commanded acceleration and the last term represents the effect of coupled Ac and Pc, and tends to be small due to the cube of the sample period.

For the component of miss distance in the Z direction, one has:

$$x_4(k+1) = z_F(k+1) = z_F(k) + \dot{z}_F(k)\ T +$$

$$+ \left[\tau T - \tau^2 (1 - e^{-T/\tau}) \right. \ Atz(k) + \frac{T^2}{2} Ac(k)\ \sin\phi_0\ \Delta\phi(k)\ -$$

$$- \frac{T^2}{2} \cos\phi_0\ Ac(k) + \frac{T^3}{6} Ac(k)\ \sin\phi_0\ \ Pc(k) - \frac{T^2}{2} \cos\theta\ g$$

where its terms have the same physical meaning as in the expression for x (k+1), with the effect of the gravity added to the expression.

Since one can notice that in the representation of the miss distance in Y direction appear two terms as a function of $\cos\phi_0$, and in the representaction of the miss distance in Z direction appear two terms as function of $\sin\phi_0$, it is

interesting to verify that the fourth term in the RHS of both expressions acts like a correction for the fifty term. Refering to fig. 3.4b, one can see that at any step of time, the commanded acceleration is actually,

$$Ac \cos\phi_0 - \Delta Ac$$

and considering small angles:

$$\Delta Ac = Ac \cos\phi_0 - Ac \cos(\phi_0 + \Delta\phi) =$$

$$= Ac \cos\phi_0 - Ac \left[\cos\phi_0 \Delta\phi - \sin\phi_0 \sin\Delta\phi\right] =$$

$$= Ac \sin\phi_0 \Delta\phi$$

The same idea can be applied to the expression for x (k+1).

The terms x_2 and x_5, represent the relative velocity, and are:

$$x_2(k+1) = \dot{y}_F(k) + \tau\,(1 - e^{-T/\tau})\,Aty(k) + Ac(k) \cos\phi_0\, T$$
$$+ Ac(k) \sin\phi_0\, T + \frac{T^2}{2} Ac(k) \cos\phi_0\, Pc(k)$$

$$x_5(k+1) = \dot{z}_F(k) + \tau\,(1 - e^{-T/\tau})\,Atz(k) + Ac(k) \sin\phi_0\, T -$$
$$- Ac(k) \cos\phi_0\, T + \frac{T^2}{2} Ac(k) \sin\phi_0\, Pc(k) - T \cos\theta\, g$$

Where the two first terms in the RHS represents the effect of the velocity and acceleration at a previous step, and the other three terms has the same meaning as previously stated.

The terms x_3 and x_6 are the target accelerations, in this model being exponentially decaying.

1. <u>Effects on the Miss Distance of the Extention of the Model</u>

In previous subsection, a extention of the model for large roll excurssions has been performed. Notice that there are two models of the system being used. The first one, used

in the algorithm is valid only for samll roll excursions, and a second model, valid for small and large roll excursions used as a representation of the missile.

The algorithm with the first model, as explained before, is initialized at each step with the actual roll angle of the missil, and performs the calculation of the comands.

In order to check the effect of the extention of the model on the miss distance, one can define a ideal initial roll angle($\phi_{o_{IDEAL}}$), as the roll angle at t=0 in order to have the comanded acceleration vector pointing to the projected final target's position. This means that the missile would not have to roll to hit the target (see fig3.5), thus the commanded roll rate calculated by the algorithm will be equal to zero. This implies that from that point ahead, the roll angle is constant and equal to $\phi_{o_{ideal}}$, and that the time history of the control Ac will be a straight line.

The fact that the roll angle will tend to this limit deserves an investigation. Notice that, as show in fig. 3.2, at the moment the missile reachs its maximum roll angle, the control Ac will be the required acceleration to hit the target if the missile initial roll angle was $\phi_{o_{ideal}}$. This means that the control Ac computed would be correct only if the missile would have turned immediatly to this angle.

For very small angles, the previous comment would be acceptable, but in a normal situation, as showed in figure3.5, the missile will only reach the ideal initial roll angle after some time, and due to the vertical component of the acceleration, when this occurs the missile would be in a position above the ideal trajectory, which means that it would follow a course parallel to the ideal trajectory to intercept.

One can see that such problem will lead to a large miss distance, in the case that the target acceleration it is not small. Thus, some correction is necessary in order to improve the missile performance.

Figure 3.5b shows the missil at some point of its trajectory where it has reach its maximum roll angle, thus it is at a parallel course with its ideal trajectory to intercept. At this point, since all the states are known, it is possible to recompute a new ϕ_{ideal} . So, if at this point the computer is feed with with the states at this point it will compute the commands in order to drive the missil to the new ϕ_{ideal} , which will introduce the desired correction. Notice that such correction can be made during all the flight, from t=0, to t=Ti.

In the present work this method has been accomplished by feeding-back to the computer the roll angle and all the states of the missile. With this information the computer is able to perform the calculation of the corrected commands at each step of time. howeveras the states are being updated, it is also necessary to update the time to intercept, which has been done using the time to go to intercept, or:

$$Ti(k) = Ti - k\ T$$

where Ti is the nominal time to intercept.

C. SIMULATION

In order to keep the same assumptions as reference 1, the matrix Q (weighting the control) has been put as suggested in such reference or:

$$\begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \quad (3.33)$$

with

$b_1 = 5.78 \quad 10^3$

$b_2 = 5.0$ meters

Five different cases were run:

Case 0, tested with one simple model valid only for small roll angles begins with missil and target on parallel courses to the inertial x axis, the target 100 meters above the missile and with an evasive manoever exponentially decaying with time constant of 20 seconds. The initial acceleration of target was -.5 g's in y direction.(see fig. 3.6)

Case 1, with the same scenarios as case 0, but was run using the algorithm for large roll excursions.

Case 2,is the same scenario as in case 1, except that the initial acceleration of target was -1.0 g's in the y direction.(Same as case 1 in ref. 1).

Case 3, the same as case2,with target acceleration of -4 g's.(Same as case 2 in the reference 1)

Case 4,same as case 3 but with target at initial position 600 meters bellow the missile.(Same as case 3 in reference 1.)

D. COMMENTS AND CONCLUSIONS

1. Results

In case 0, the missile begins its trajectory commanding 26.5 m/sec^2 and the time history of the control Ac follows exactly a straight line in the form suggested in

ref. 1, as seen in fig.3.7The control Pc, begining at .35 rad per second, is decaying and reachs zero at k=80 (see fig.3.8). Figures 3.9 and 3.10 show the miss distance, where one can see that the missile is crossing the target with a CG-to-CG distance of 1.5 meters.

Figure3.11, shows the time history for the roll angle, which as expect, reachs a constant value, with the missile crossing target at t=Ti, with a bank of .44 radians.

In case 1,which was run with the model for large roll excursions, the missile kept the same Aco, but there is a very small increase in further commanded acceleration in order to correct the effect of the roll angle on the vertical component of the control Ac, as seen in fig.3.12, and table I.

Figure 3.13 shows that the roll rate decreases almost as before, and the final roll angle is .42 rds, as shown in fig.3.15 and table I. Refering to figure 3.14 there is a change in the final miss distance, which is better than case 0, due to a improvement in its Z component (see table I). This results in a final CG-to-CG miss distance of .65 meters.

In case 2, the missile has the same Ac at t=0, but with the correction for roll angle being increased due to the increase of target acceleration (see fig.3.16), the commanded Ac reachs a larger peak value.

The initial value of the commanded roll rate is .69 radians, which is larger than case 1, due to the increase in the target acceleration. In figure 3.18, there is no noticeable change in the shape of the curve for Z direction, and in the Y direction the final distance is about the same as in case 1. The CG-to-CG distance at t=Ti is .73 meters. The larger roll rate leads to larger roll angles as seen in fig. 3.19, where the final roll angle is .72 radians.

The effect of target acceleration can be easily seen in case 3, where one can see that with the same Aco, the accelerations are largelly increased from this point, and the missil begins its trajectory with very high roll rate(see figures 3.20 and3.21). There is a change in the Z component of the miss distance, that decreases its final value to .05 meters, but now the miss distance in Y direction is made worse as shown in figure 3.22, which leads to a final CG-to-CG distance of 1.5 meters. In fig.3.23 one can see that the missile crosses the target with a bank angle of 1.26 radians.

In case 4, due to the position of target 600 meters under the missile, the initial commanded acceleration is negative and reachs the limit of -2 g's. The initial roll rate begins at a smaller value than in case 3 but increases during the initial part of the flight reaching its peak value at 1.75 seconds when again as in the previous cases begins to decay. As the missile banks to roll angles larger than 90 degrees, the acceleration goes to positive values, as seen in figures 3.24 and 3.25 Figure 3.26, shows the worse case among these in respect to the miss distance, mainly in the Z direction, and in the final cg-to-cg distance, which is equal to 4.43 meters. Also the final roll angle of 3.0 radians is the largest among all these cases, as seen in fig. 3.27.

2. Comments

Defining the projected zero effort miss distance (ZEM) as the miss distance the missile cross the target with no commands. It can be calculated at t=0 as the initial distance between target and missile plus the miss distance due to the gravity or:

$$ZEM = z(0) + \frac{1}{2} g t^2 \qquad (3.34)$$

which is equal in all the three first cases.

In the first three cases, the initial missile's commanded acceleration is the same as seen in table 3.1. Considering that the control Ac necessary to correct the initial miss distance can be calculated as:

$$Ac_{ZEM_0} = \left\{ \frac{ZEM_z}{\frac{Ti^2}{2} - \frac{Ti^2}{6}} \right\} = 26.7 \qquad (3.35)$$

which is close to the initial control Ac.

This suggests that the initial Ac, would be that one necessary to correct the initial ZEM in Z direction, which agrees with reference 1. Notice however that in all cases the initial Ac is less than the calculated value of Aco, which agrees with the previous statement that some error was expected in the initial part of the computations.

Considering the ideal initial roll angle as defined before, one has:

$$\phi_{o_{ideal}} = \tan^{-1} \left[\frac{ZEM_y}{ZEM_z} \right] \qquad (3.36)$$

From table I, and figures showing roll angles, it can be verified that the missile is banking to reach angles larger than $\phi_{o_{ideal}}$, in order to correct its trajectory to hit the target.

Therefore, in all cases, the missile begins its trajectory with an Aco (discussed before), and a roll rate

which is proportional to the target acceleration in the Y direction. As the missile rolls at decreasing roll rates, the commanded acceleration is changed in order for to compensate the effect of the roll angle on its Z direction component. At some time when the control Pc is zero or near zero, the control Ac begins to follow a linear law, as suggested in eqn. 2.7:

$$Ac = Aco \left[1 - \frac{t}{Ti} \right]$$

Notice however that the term Aco in this equation is no more the actual initial commanded acceleration, but that one the missile would have if its initial roll angle was equal to the final .

Such behavior defines a boundary in the control Ac which is clearly seen in 3.29, where the commanded acceleration is bounded by the curve of the control Ac the missile would have if its initial ϕ_o was equal to $\phi_{o_{ideal}}$ (or, if no commanded roll was necessary to reach the target).

Althoug the missile commands roll angles larger than the ideal initial roll angle, it is possible to do a prediction- with no no computer work- of an aproximation for the maximum acceleration the missile would experience during its fligth, as following (see fig. 3.32):

$$A_c = \left[\frac{ZEM_z}{\frac{T_i^2}{2} - \frac{T_i^2}{6}} \right] \frac{1}{\cos \phi_{o_{ideal}}} \qquad (3.37)$$

where $\phi_{o_{ideal}}$ cames from eqn. 3.50, and Ac is a streight line as in fig 3.32.

Other interesting point is that the missile is commanding to reach roll angles about twice of $\phi_{o_{ideal}}$ when the target is at small accelerations, and when at large

accelerations, the missile is making a small correction on its roll angle, as shown in table I.

One can see from the figures showing miss distance, that the relative position of missile to target is about the same in all three initial cases. Thus, the new ϕ_{ideal} at each point is the same. As the $\phi_{o_{ideal}}$ computed at t=0 is smaller when the target is at small accelerations, the correction has to be larger in order to hit the target, as seen in figure 3.30.

In table I, one can see the final miss distances, the final miss distance cg to cg and the final roll angle. Such results show that with this digitalized model, good results has been obtained.

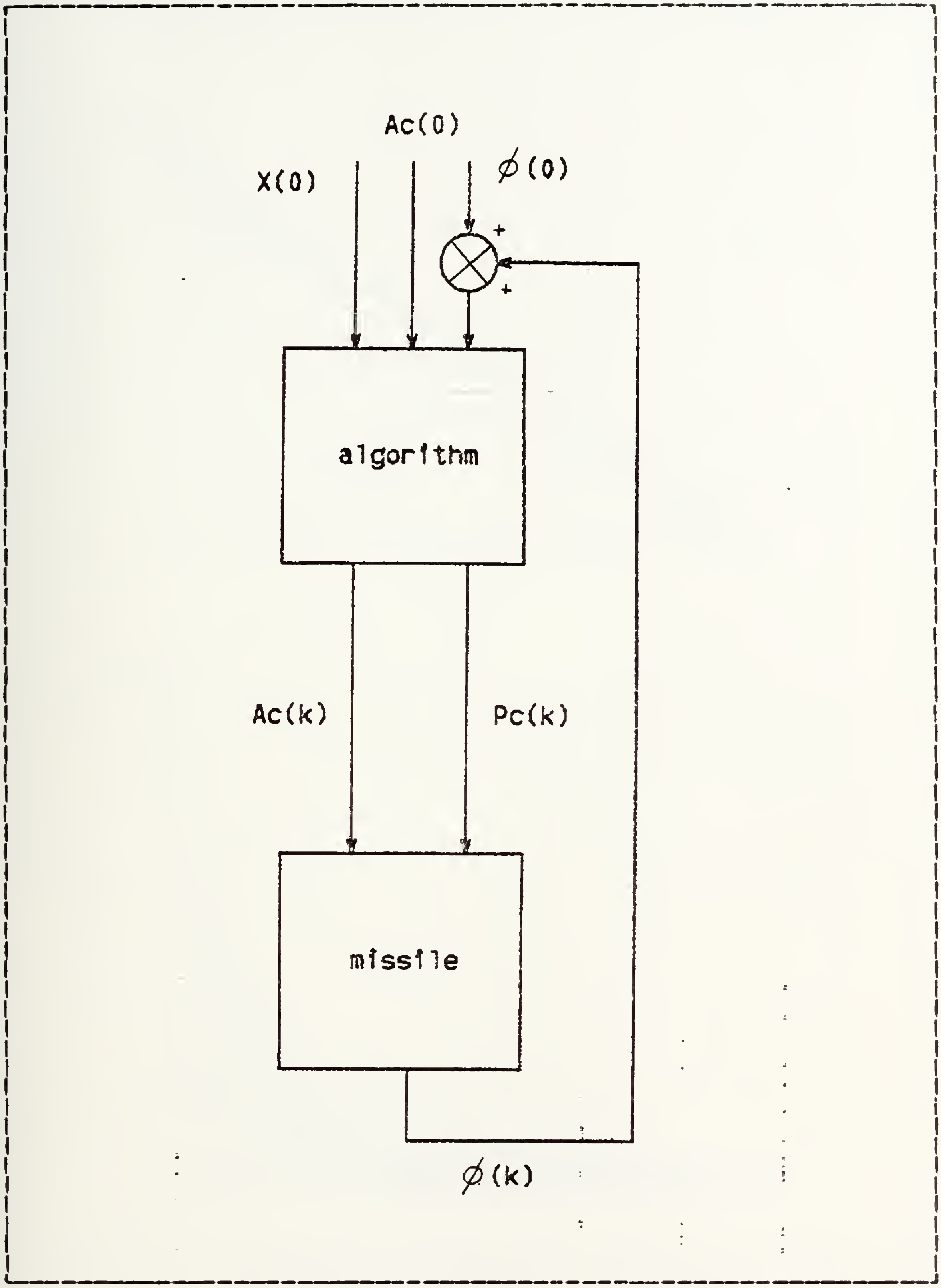

Figure 3.1 Representation of the System.

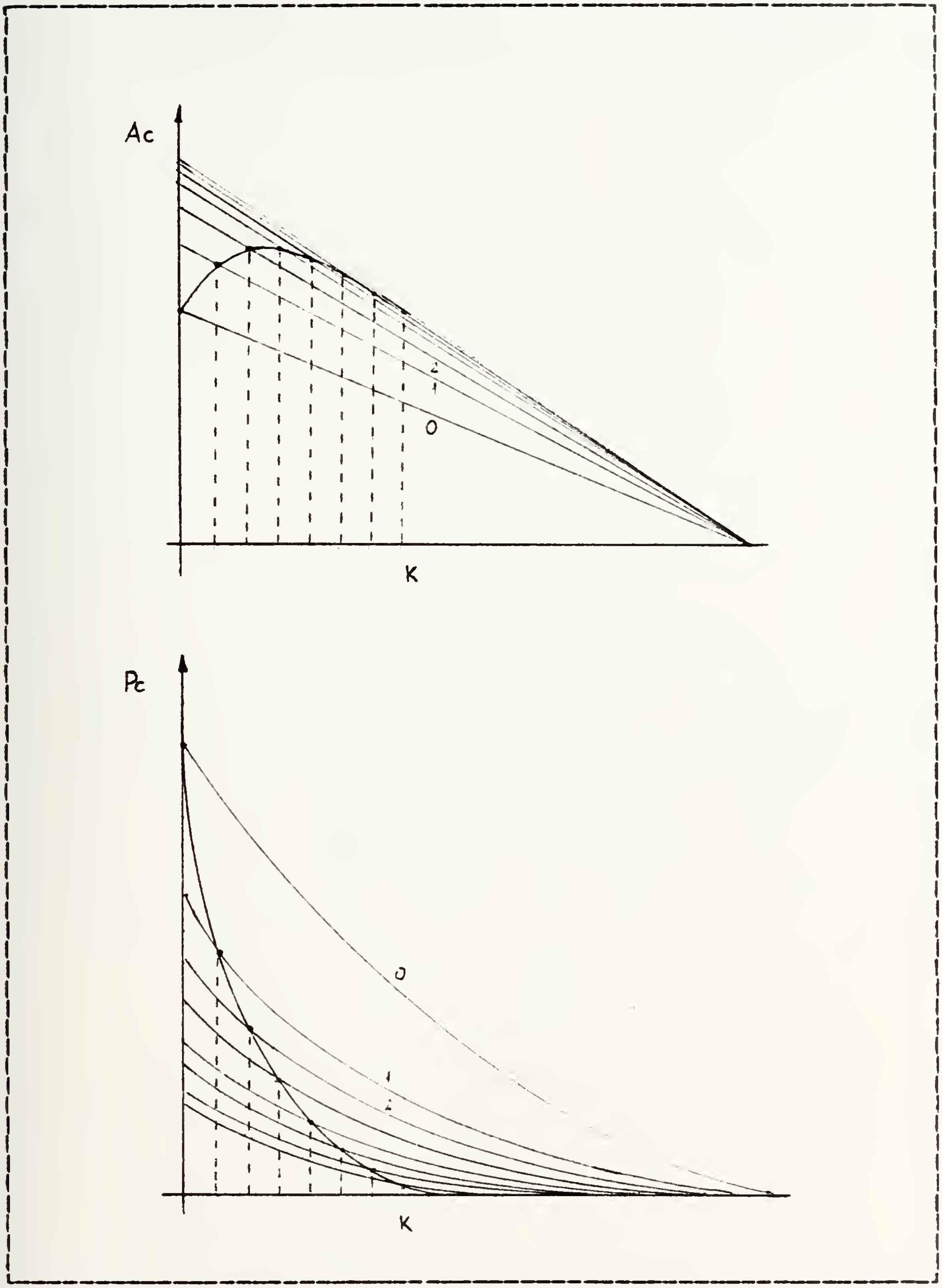

Figure 3.2 Variation of Commands with Initial Roll Angle.

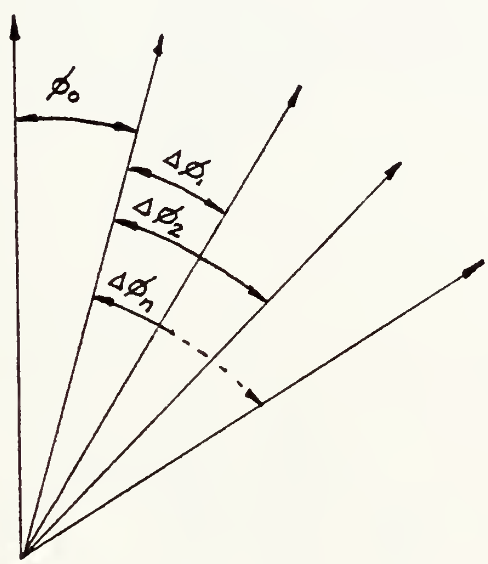

Figure 3.3 Variation of Roll Angle - Small Angles.

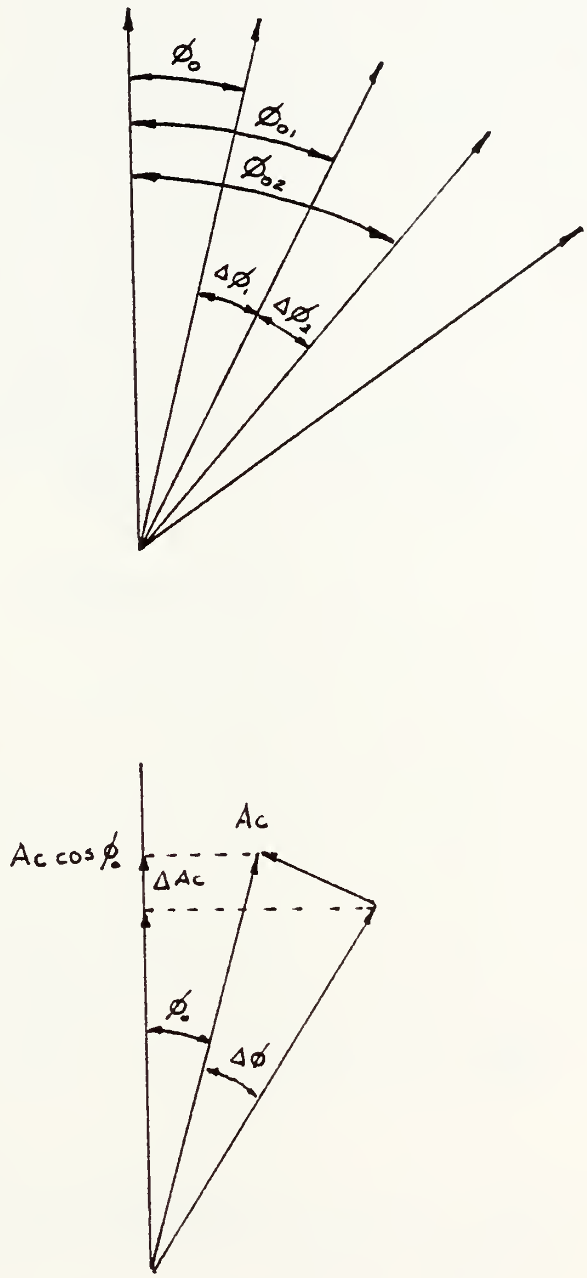

Figure 3.4 Variation of Roll Angle - Large Angles.

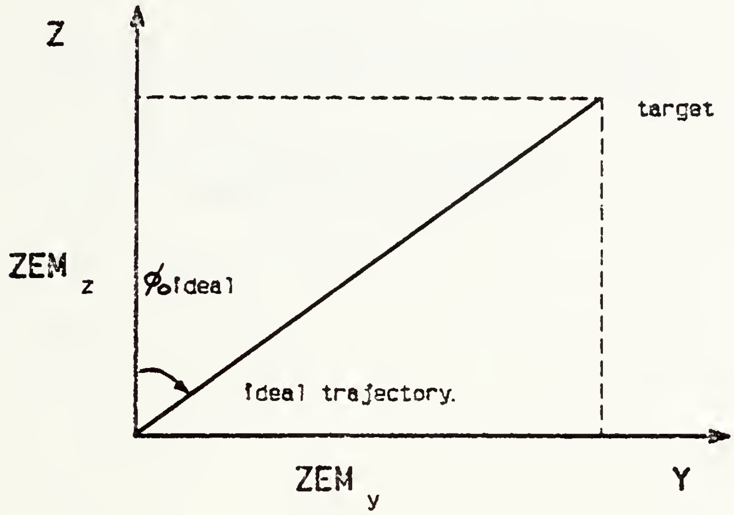

Figure 3.5a

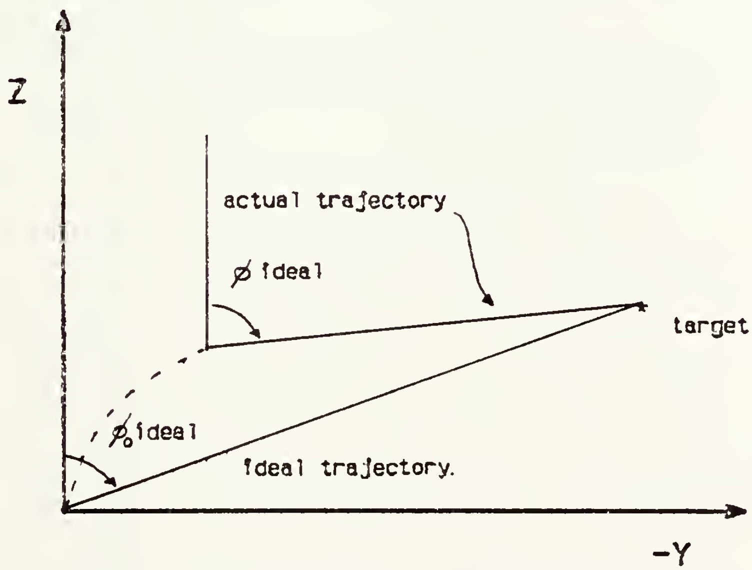

Figure 3.5b

Figure 3.5 Ideal Initial Roll Angles.

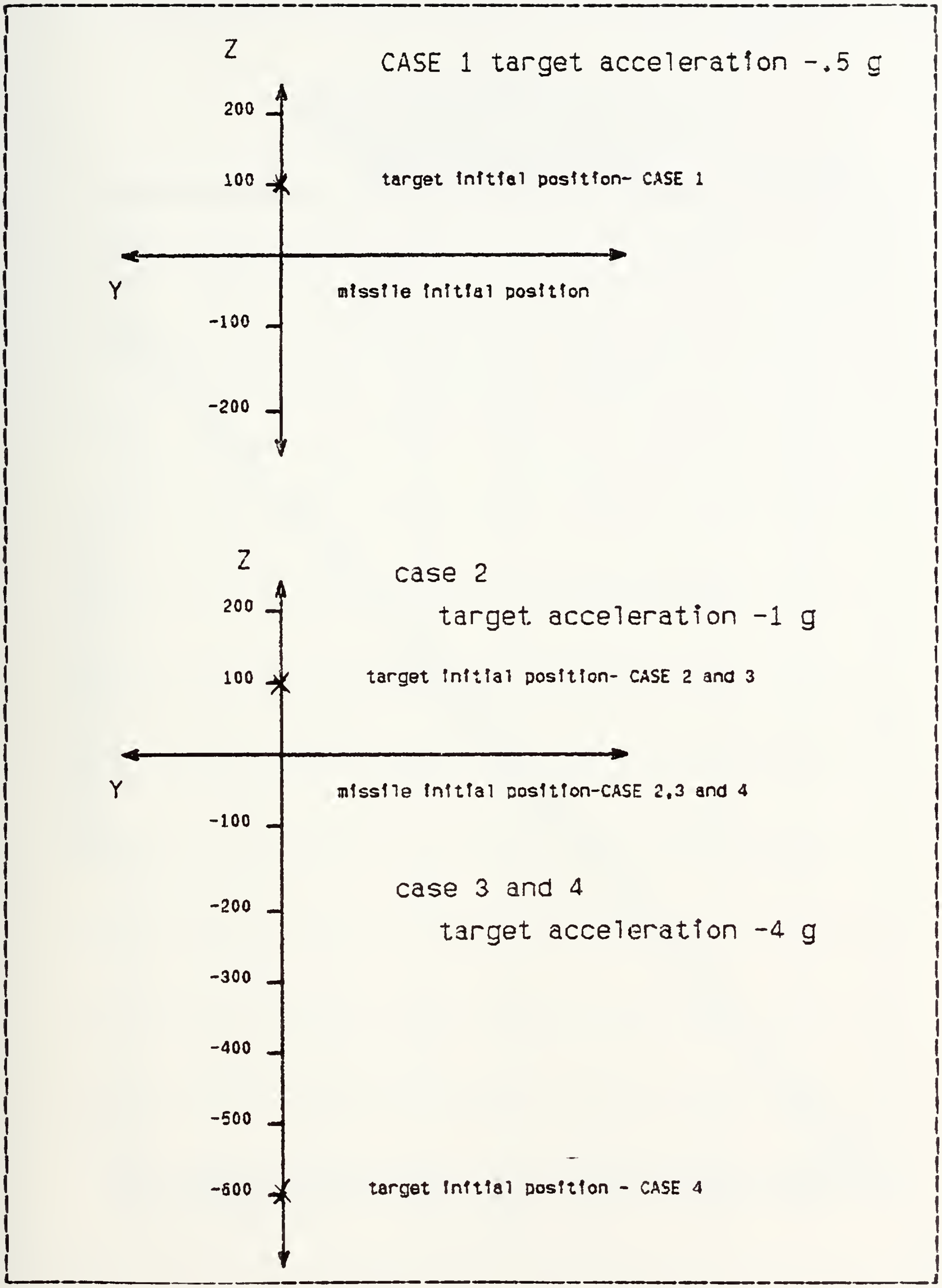

Figure 3.6 Scenarios for Simulation.

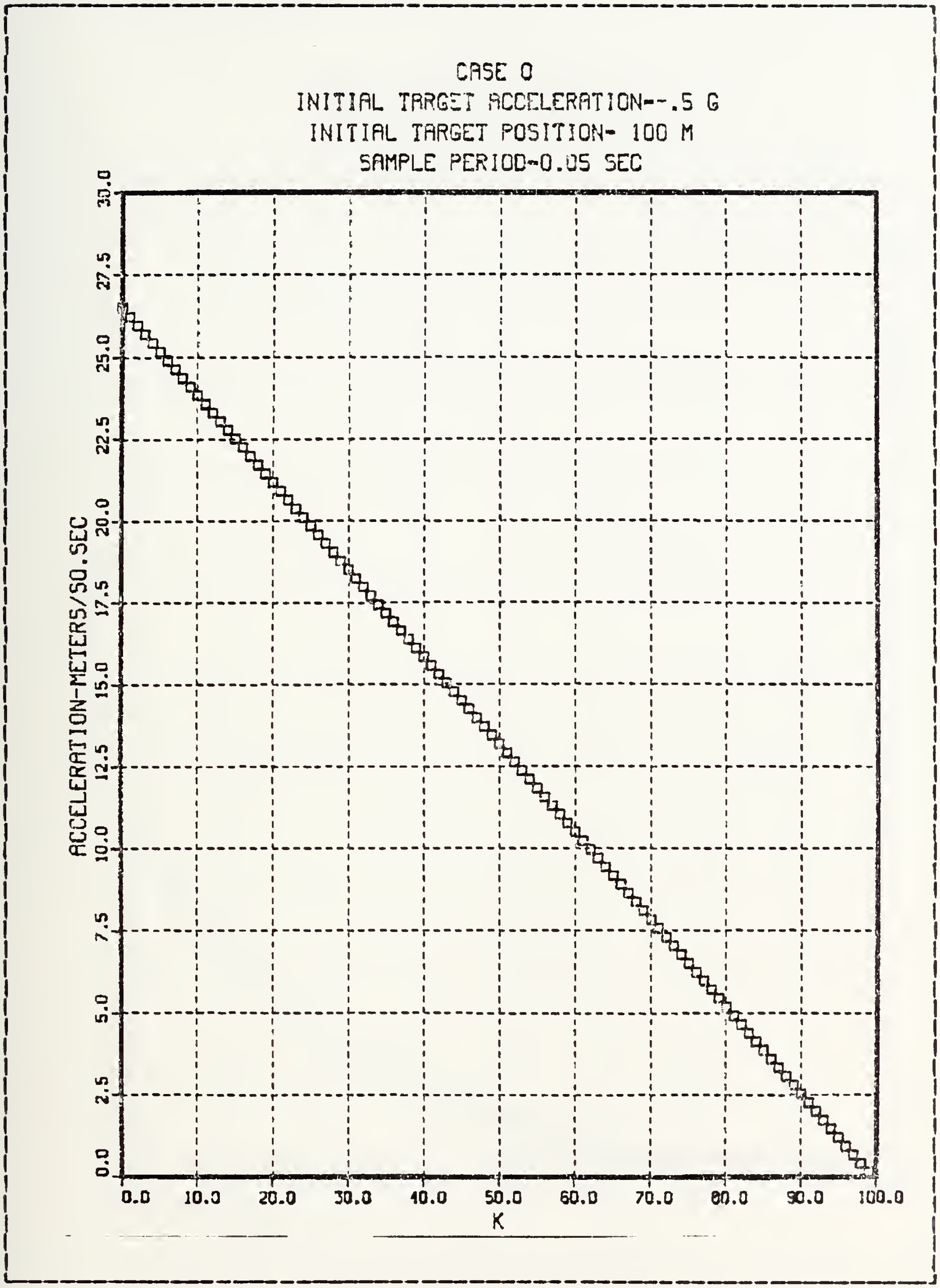

Figure 3.7 Commanded Acceleration- Case 0.

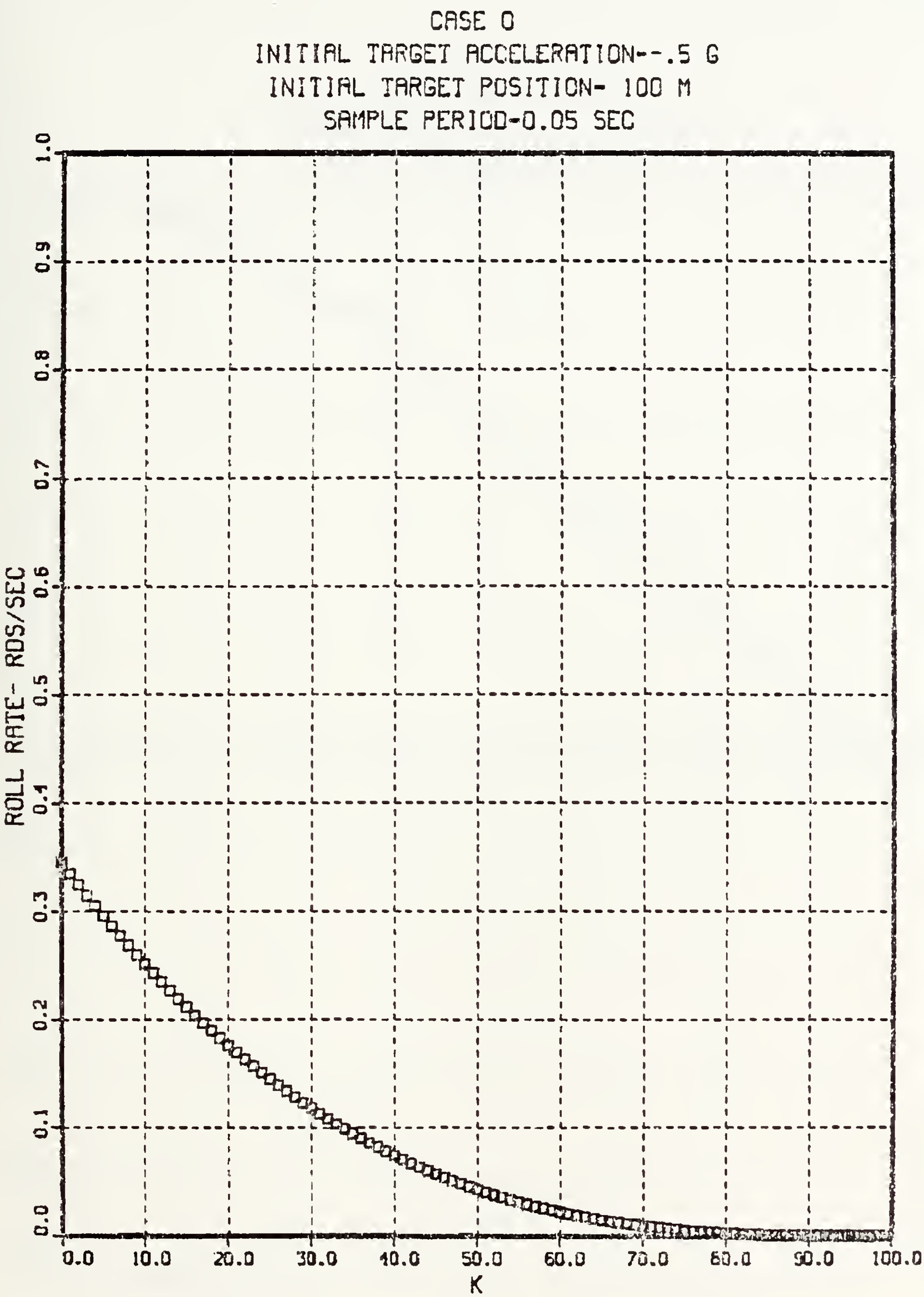

Figure 3.8 Commanded Roll Rate- Case 0.

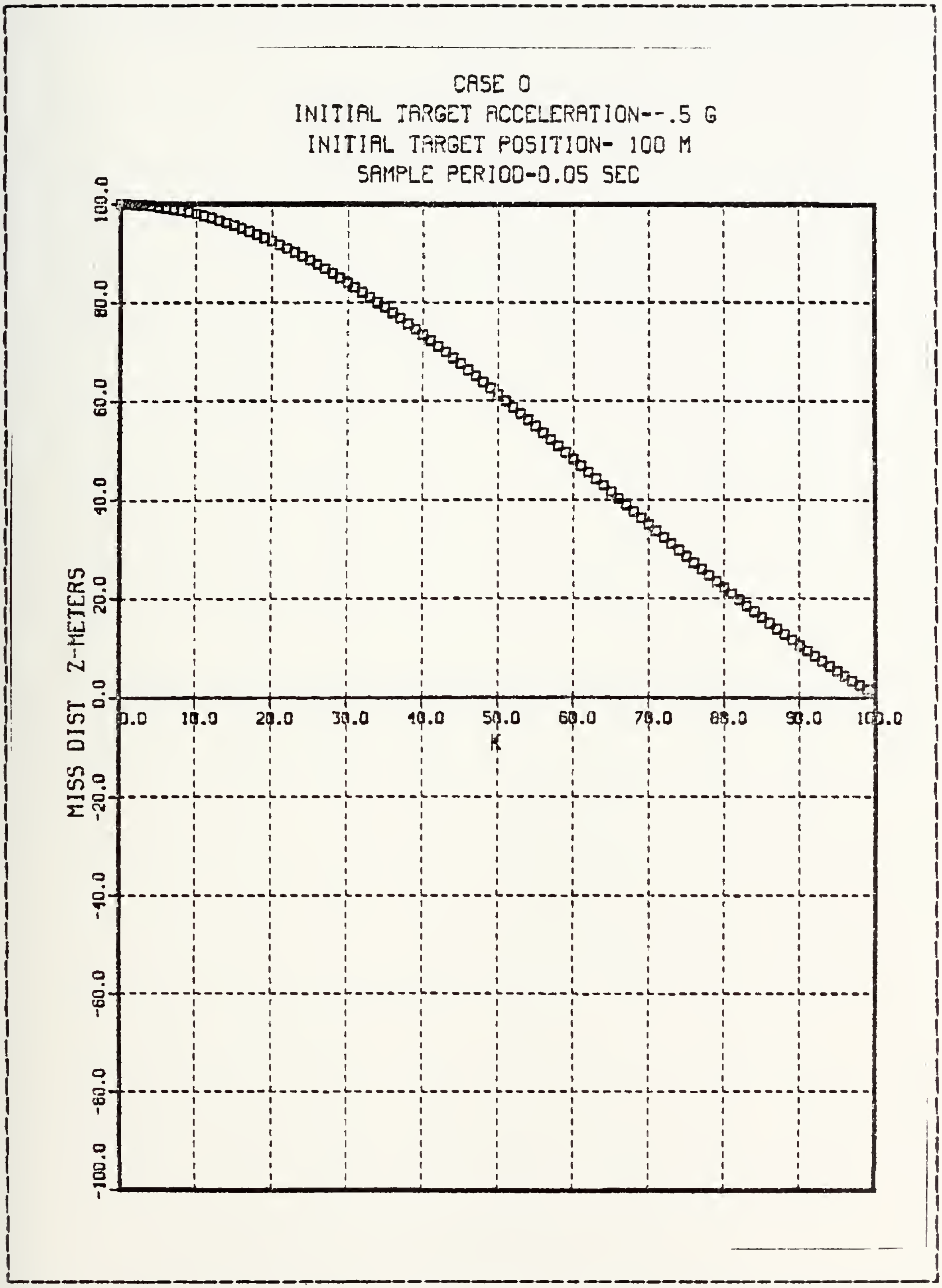

Figure 3.9 Miss Distance in Z Direction- Case 0.

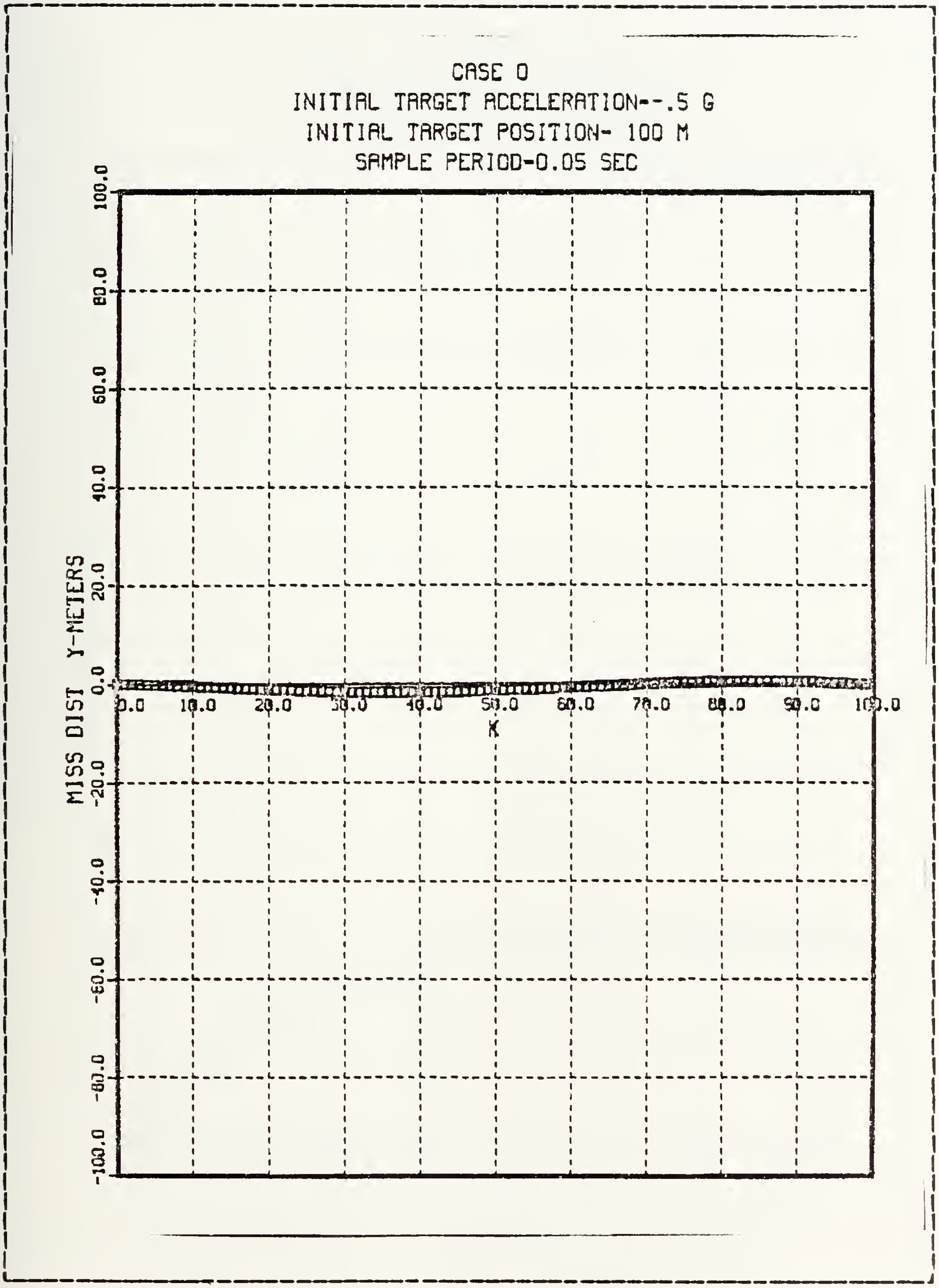

Figure 3.10 Miss Distance in Y Direction- Case 0.

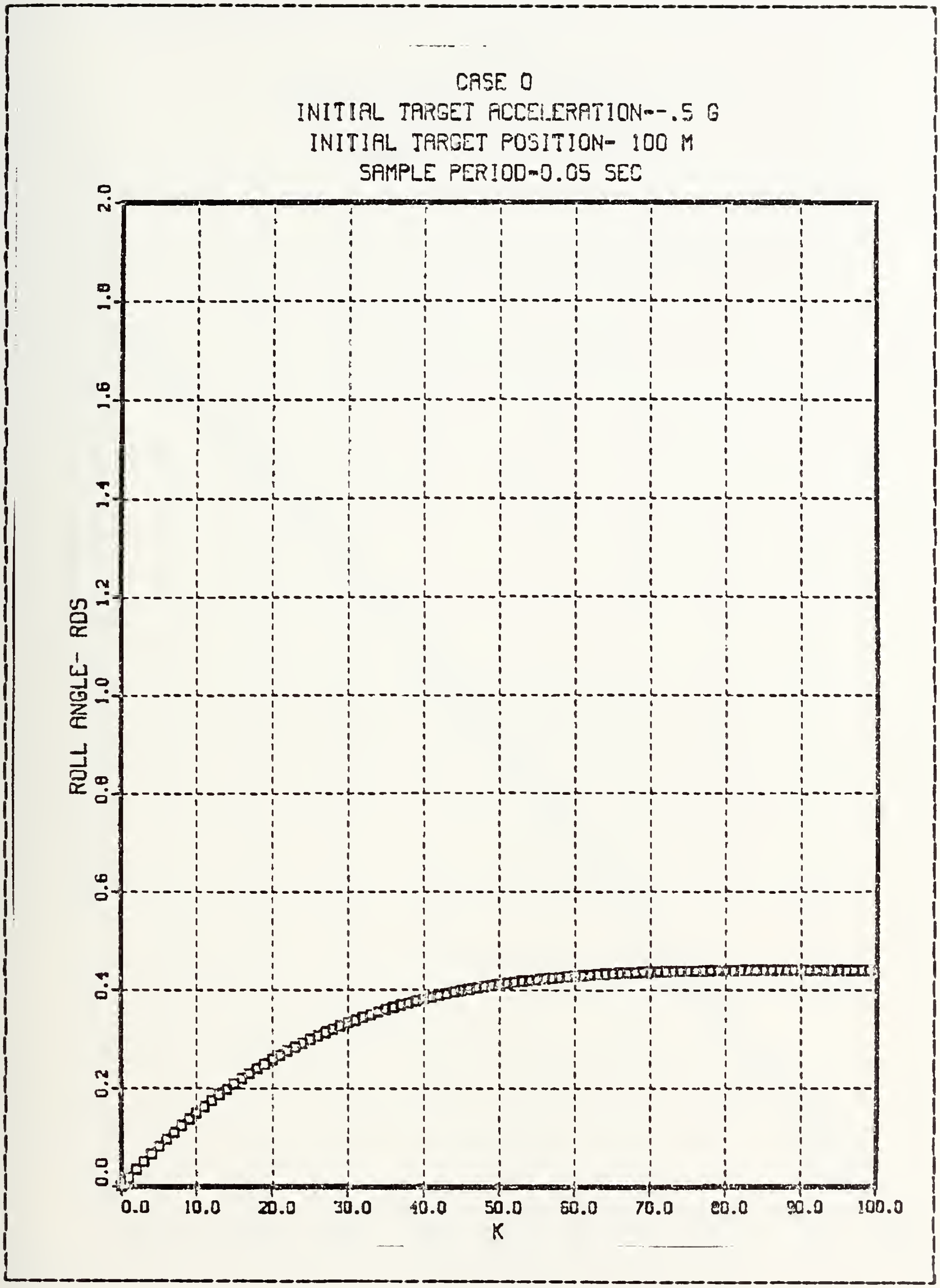

Figure 3.11 Roll Angle- Case 0.

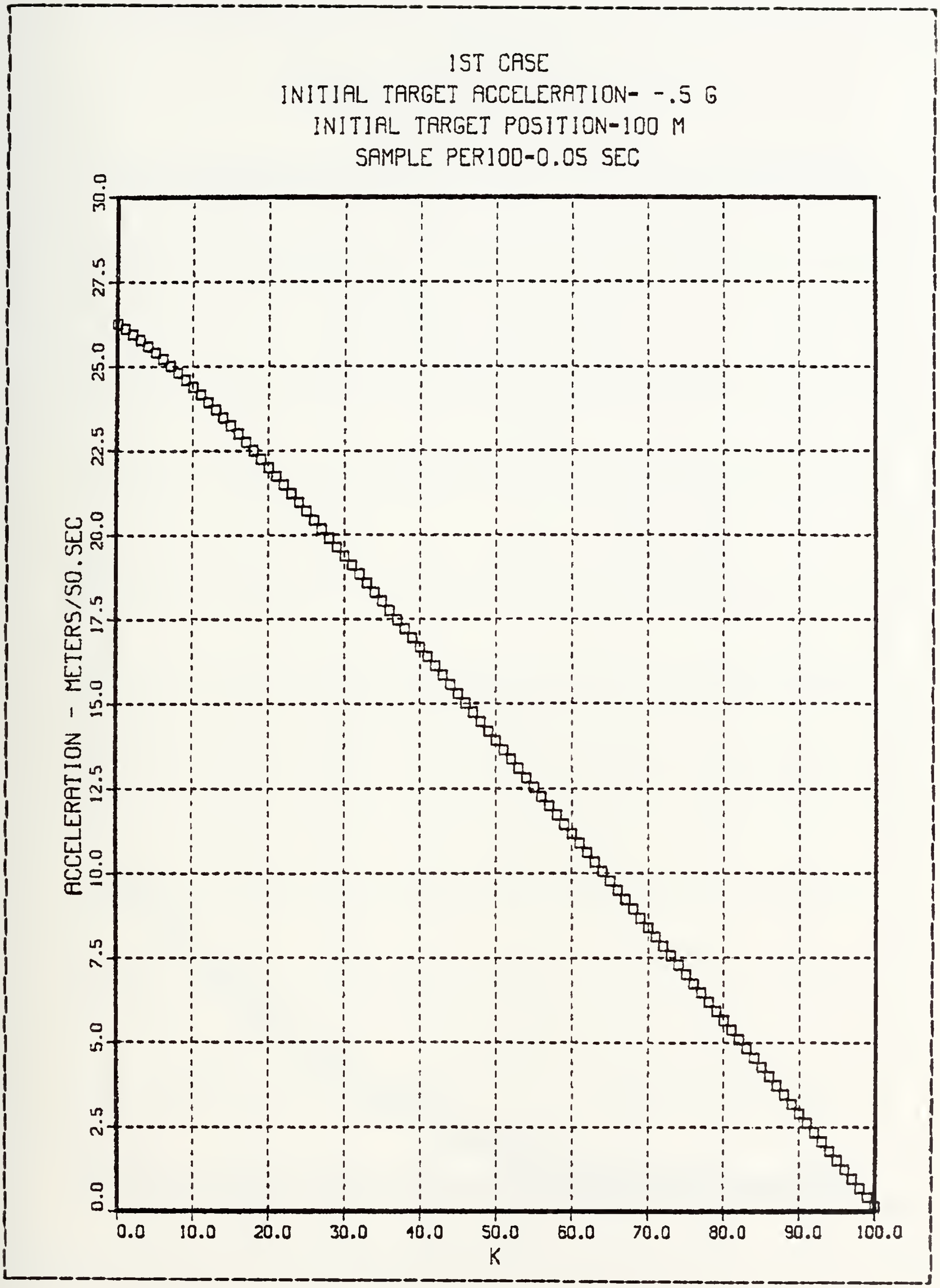

Figure 3.12 Commanded Acceleration- Case 1.

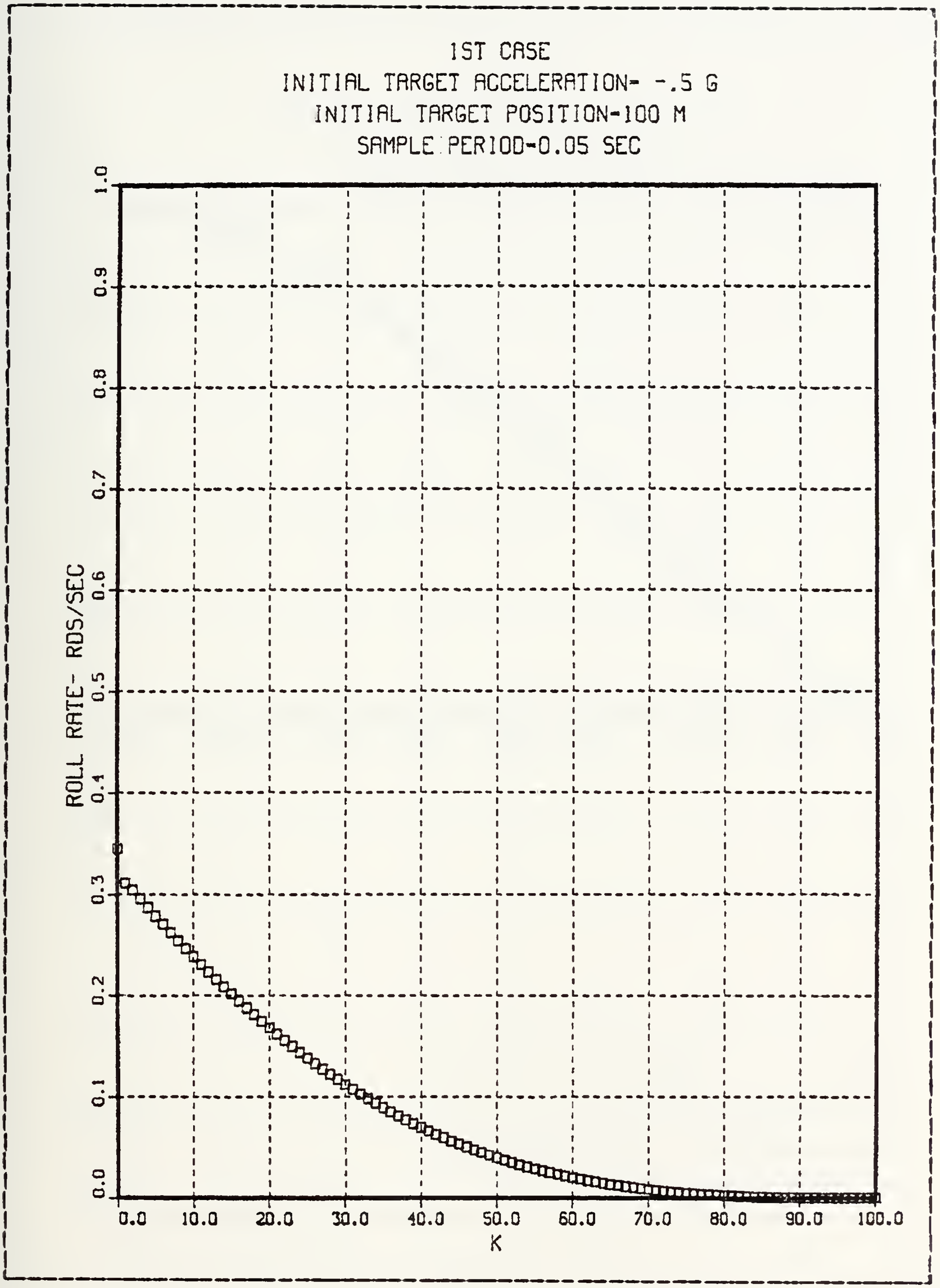

Figure 3.13 Commanded Roll Rate- Case 1.

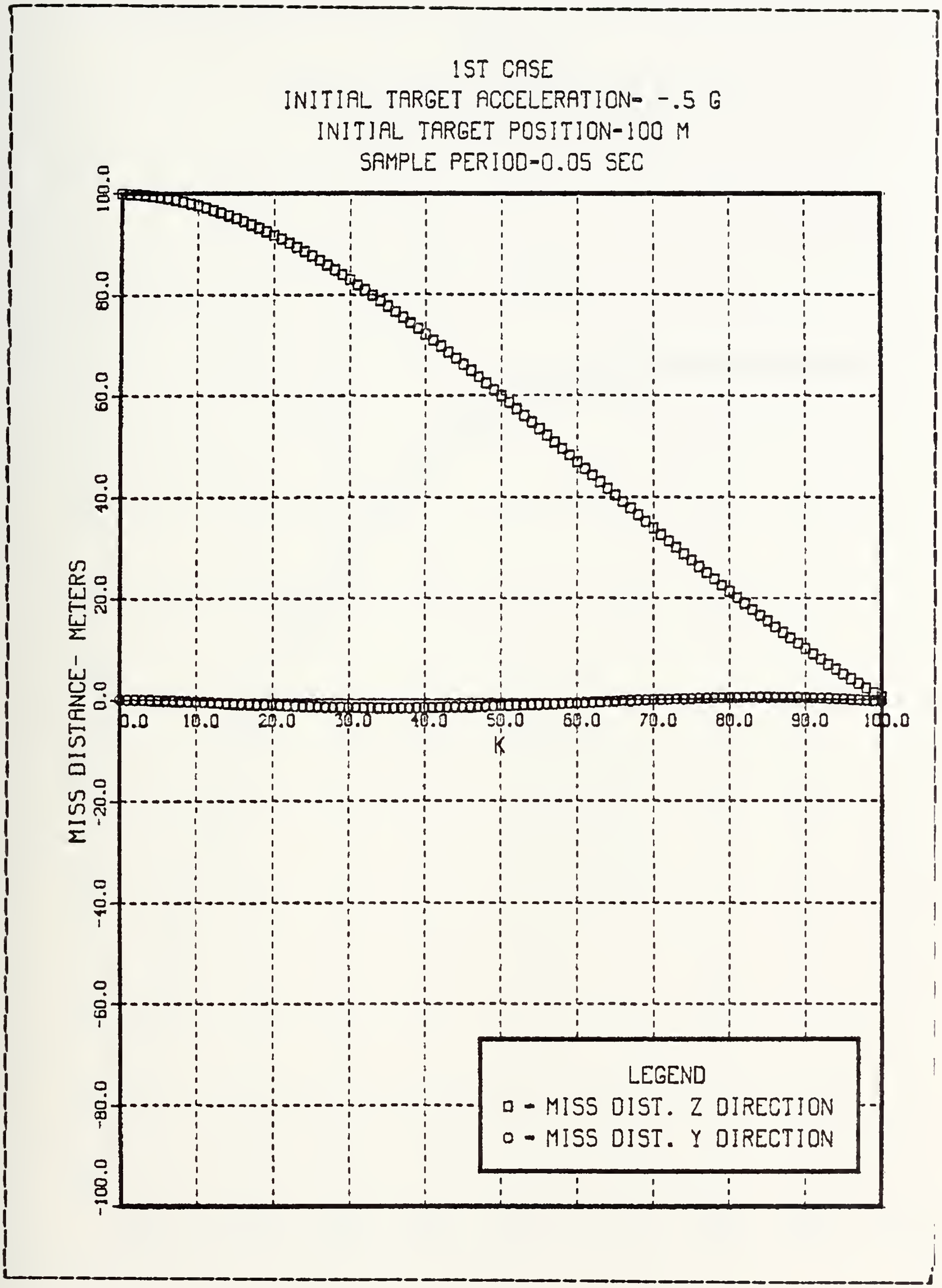

Figure 3.14 Miss Distance- Case1.

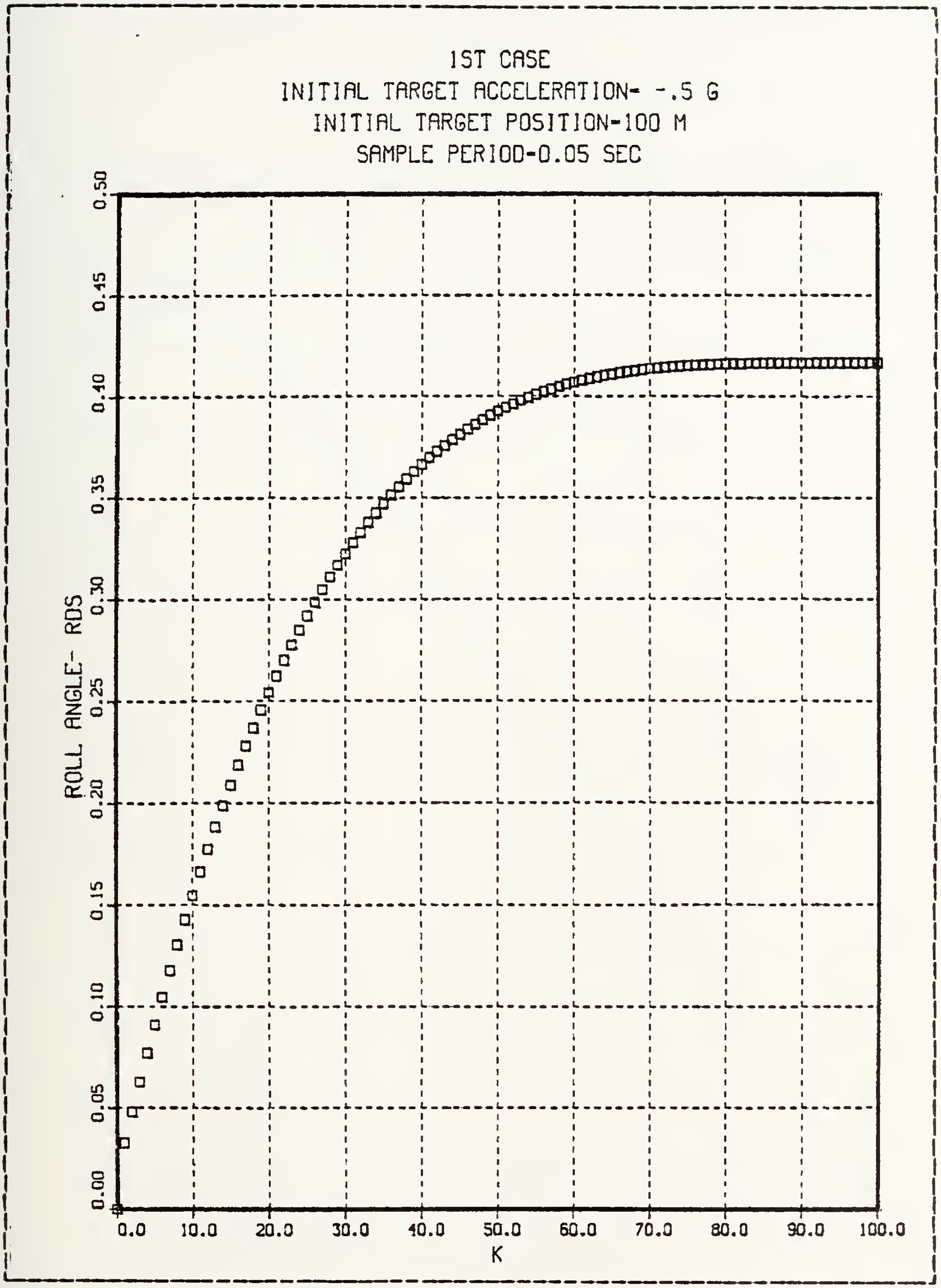

Figure 3.15 Roll Angle- Case 1.

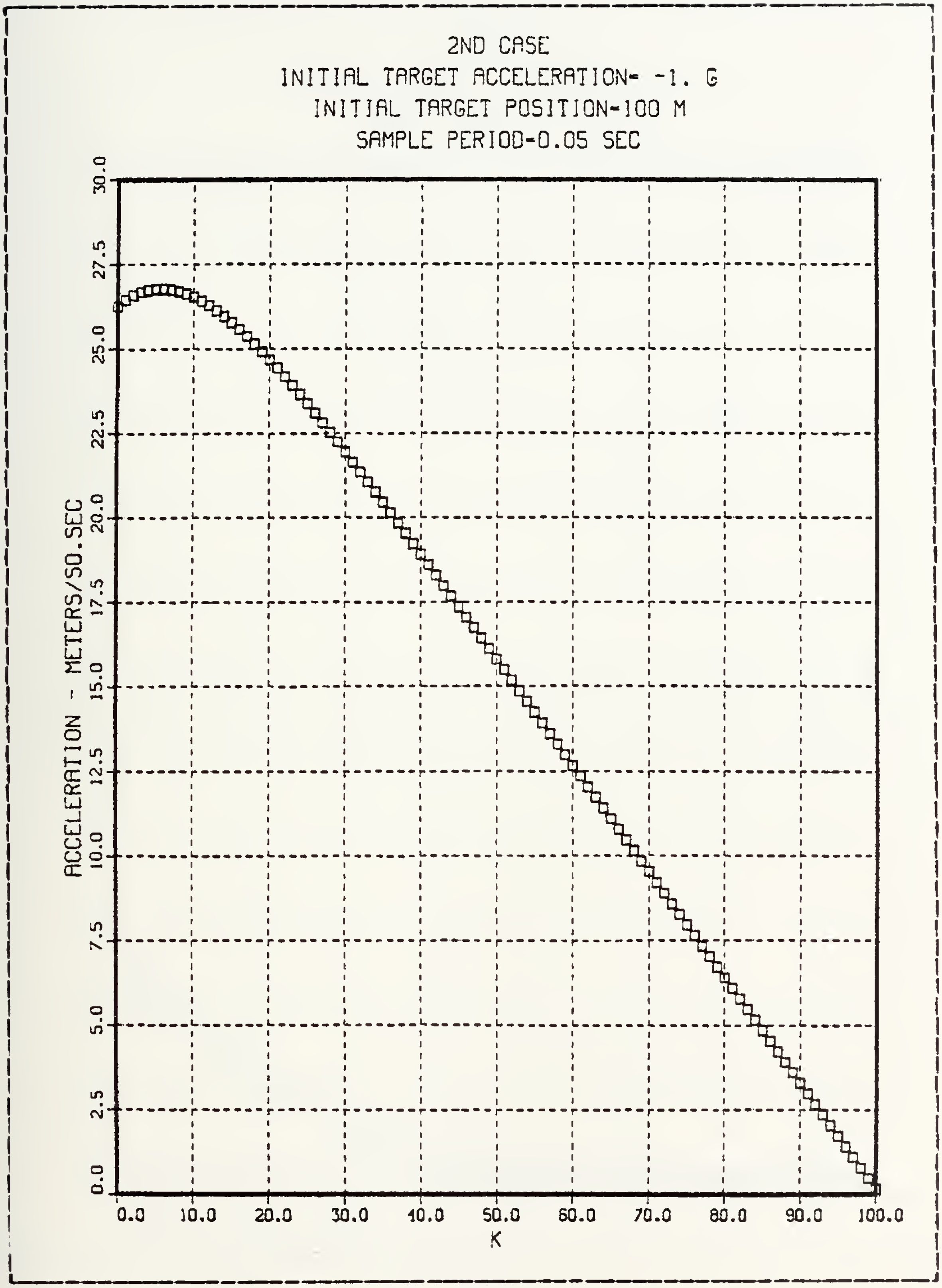

Figure 3.16 Commanded Acceleration- Case 2.

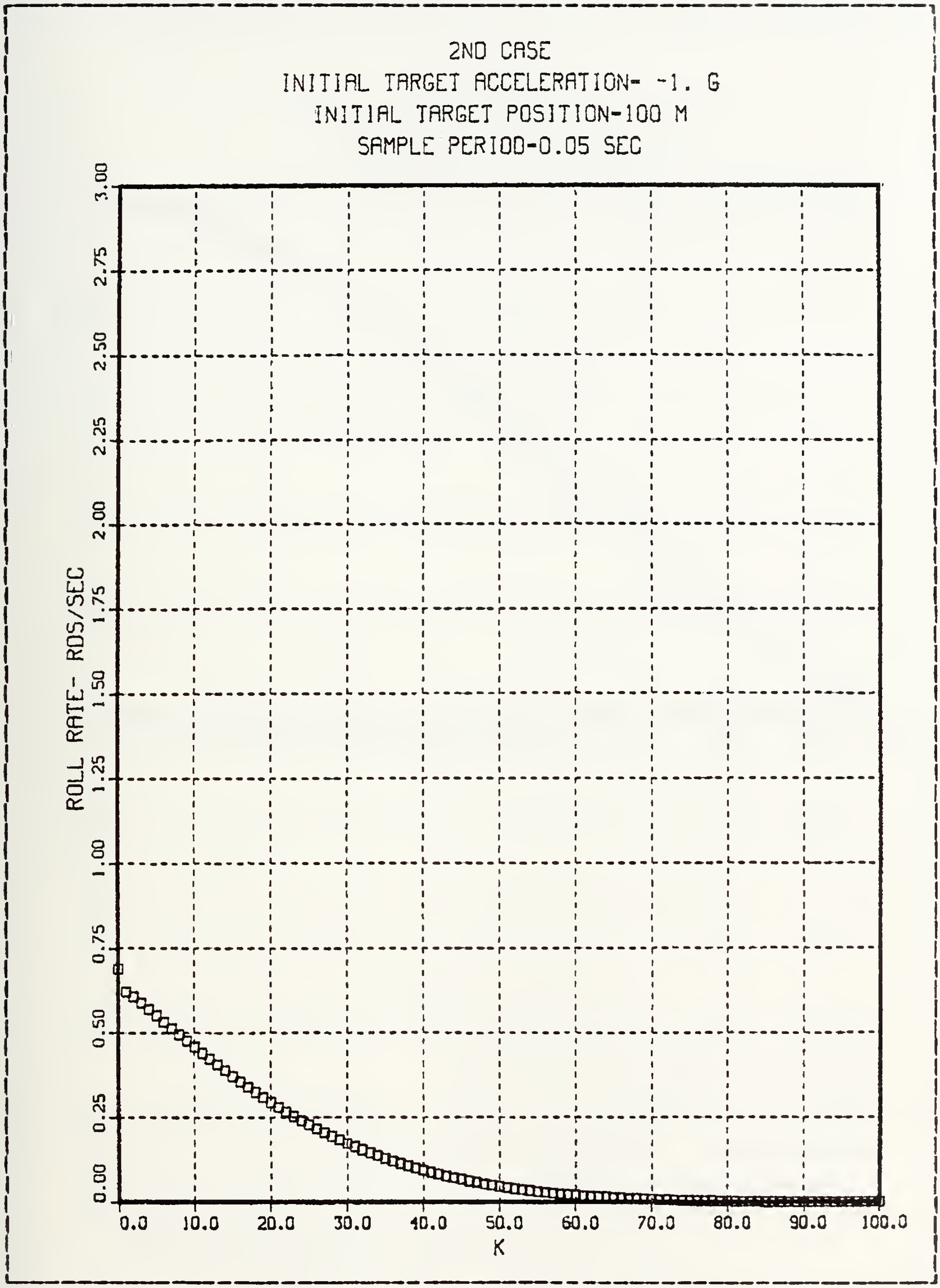

Figure 3.17 Commanded Roll Rate- Case 2.

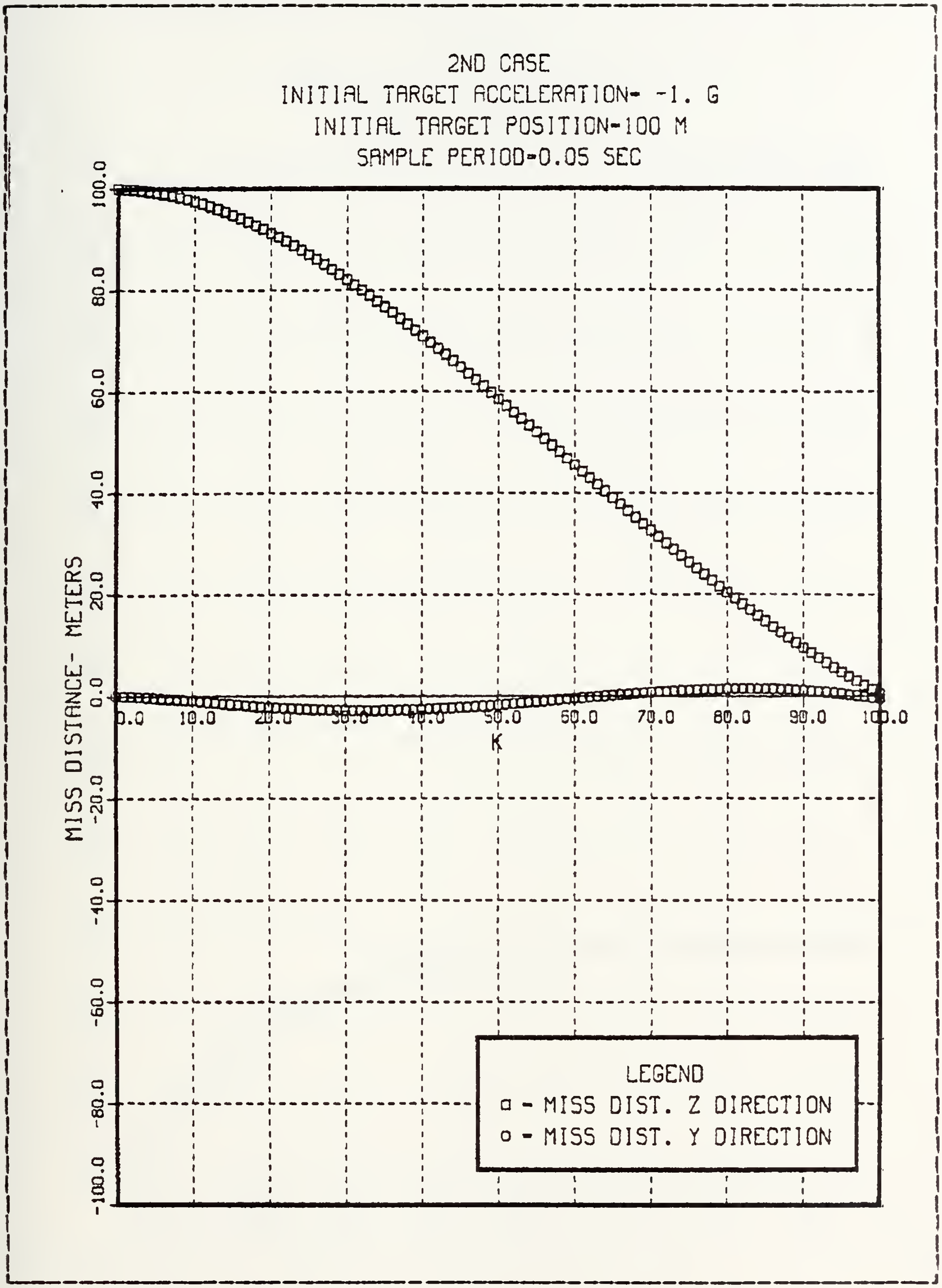

Figure 3.18 Miss Distance- Case 2.

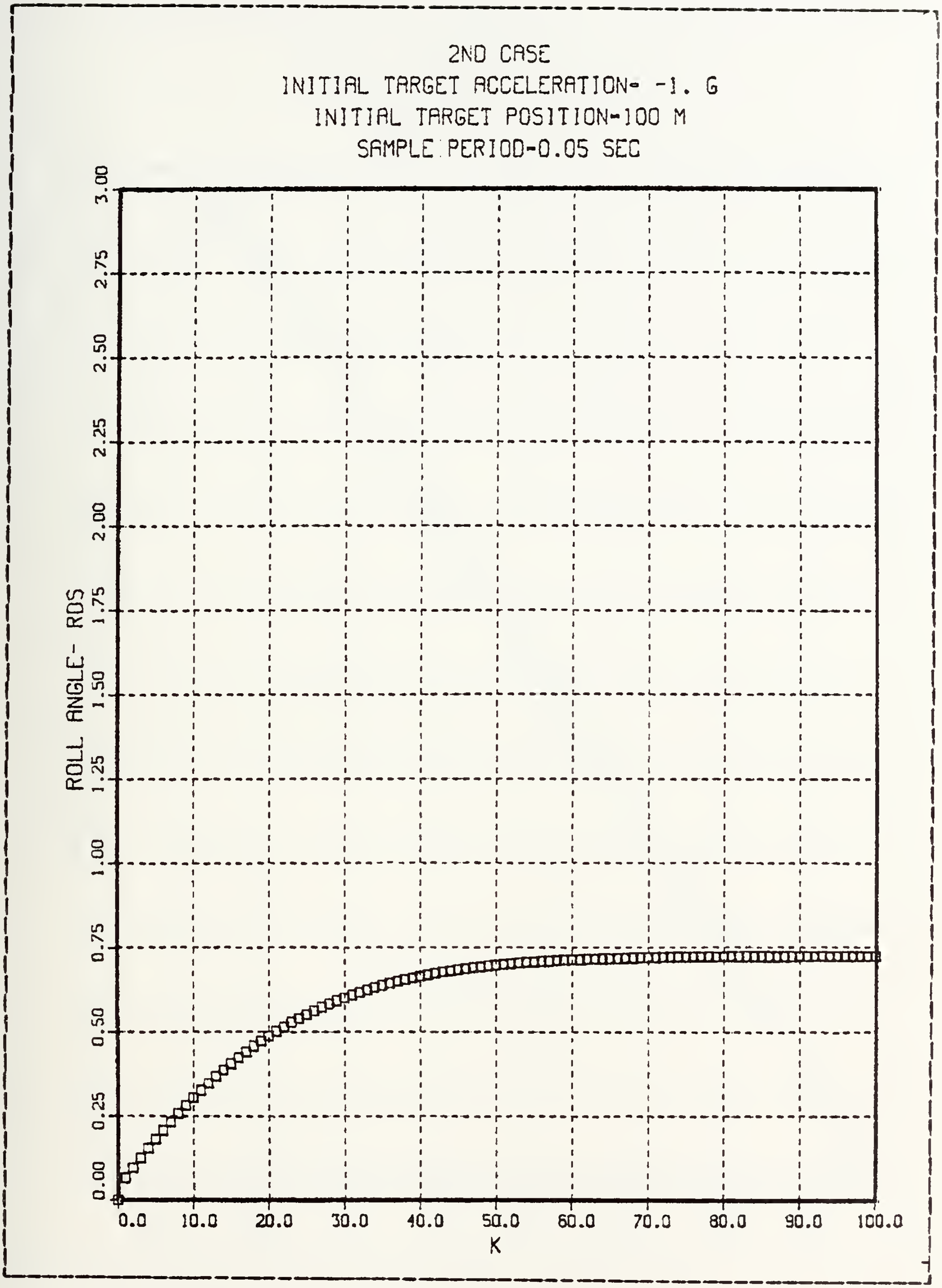

Figure 3.19 Roll Angle- Case 2.

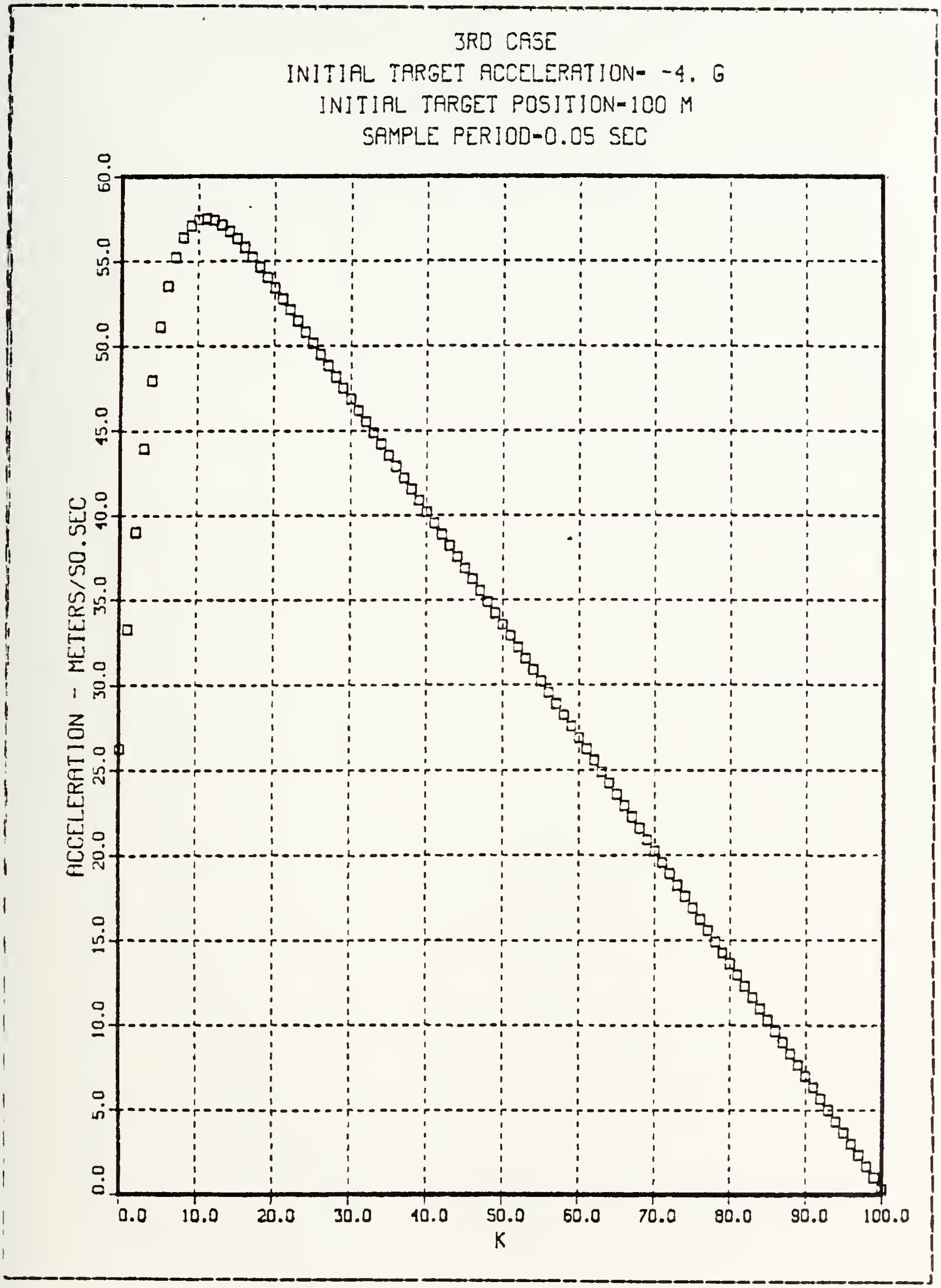

Figure 3.20 Commanded Acceleration- Case 3.

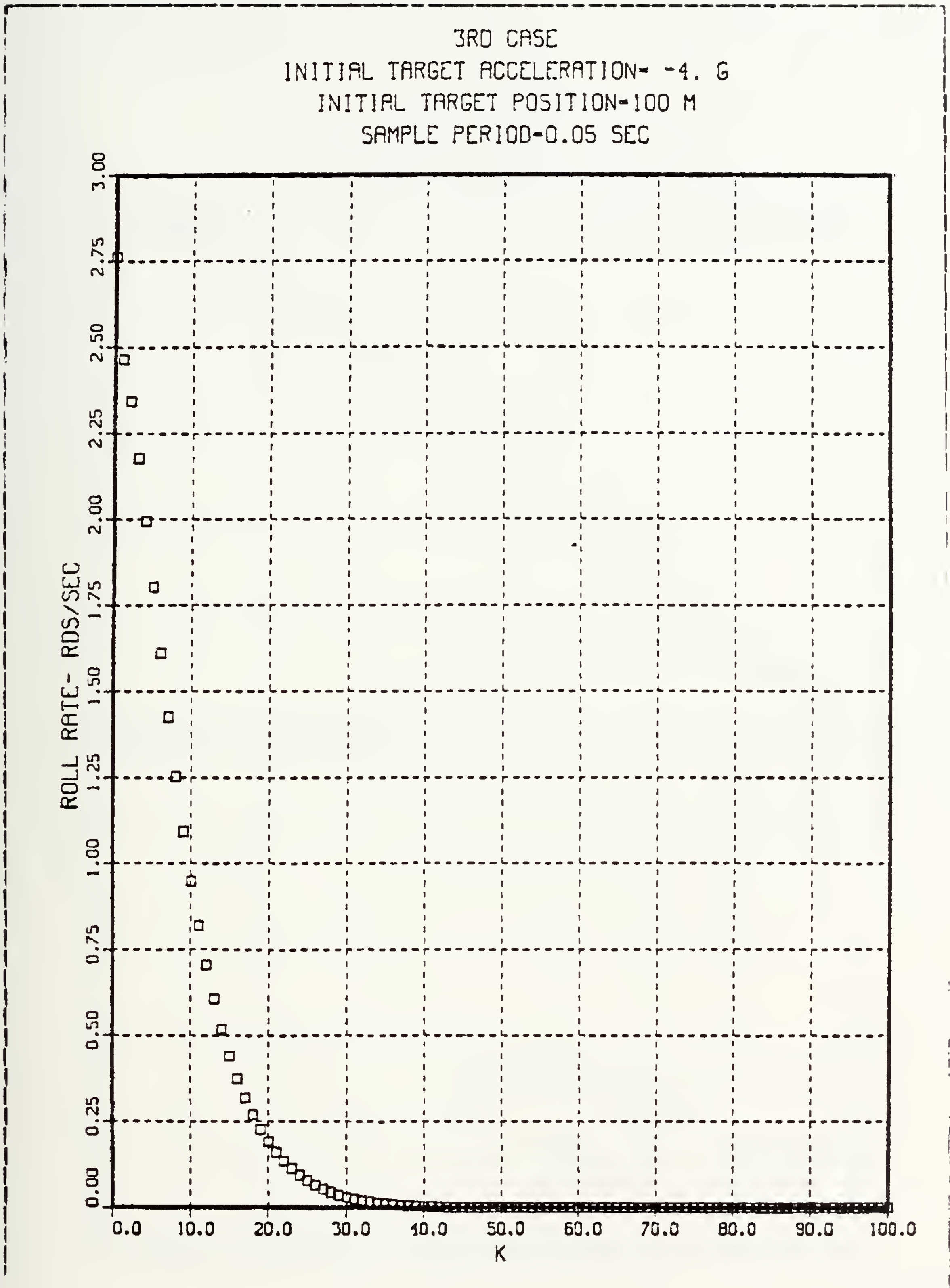

Figure 3.21 Commanded Roll Rate- Case 3.

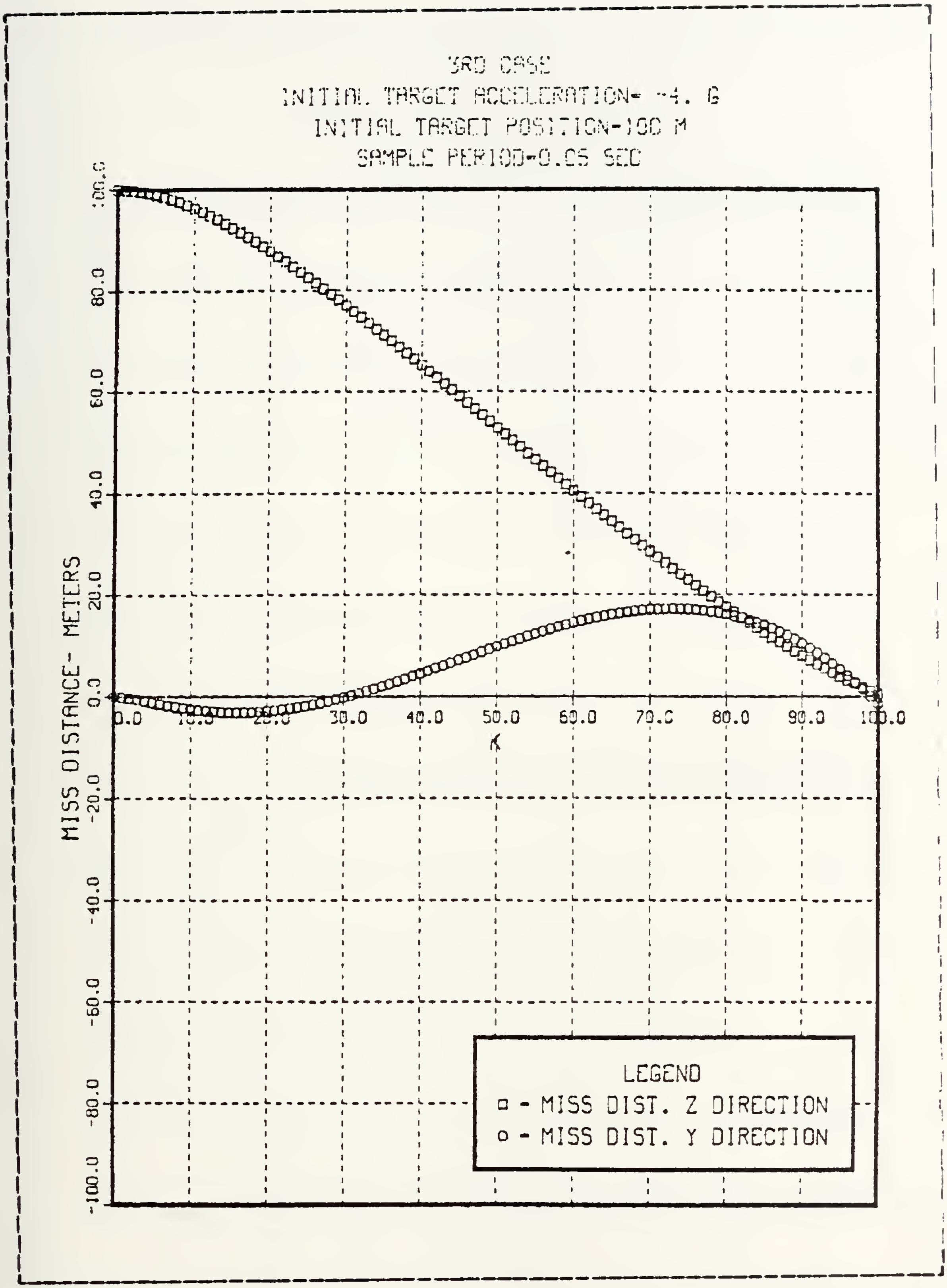

Figure 3.22 Miss Distance- Case3.

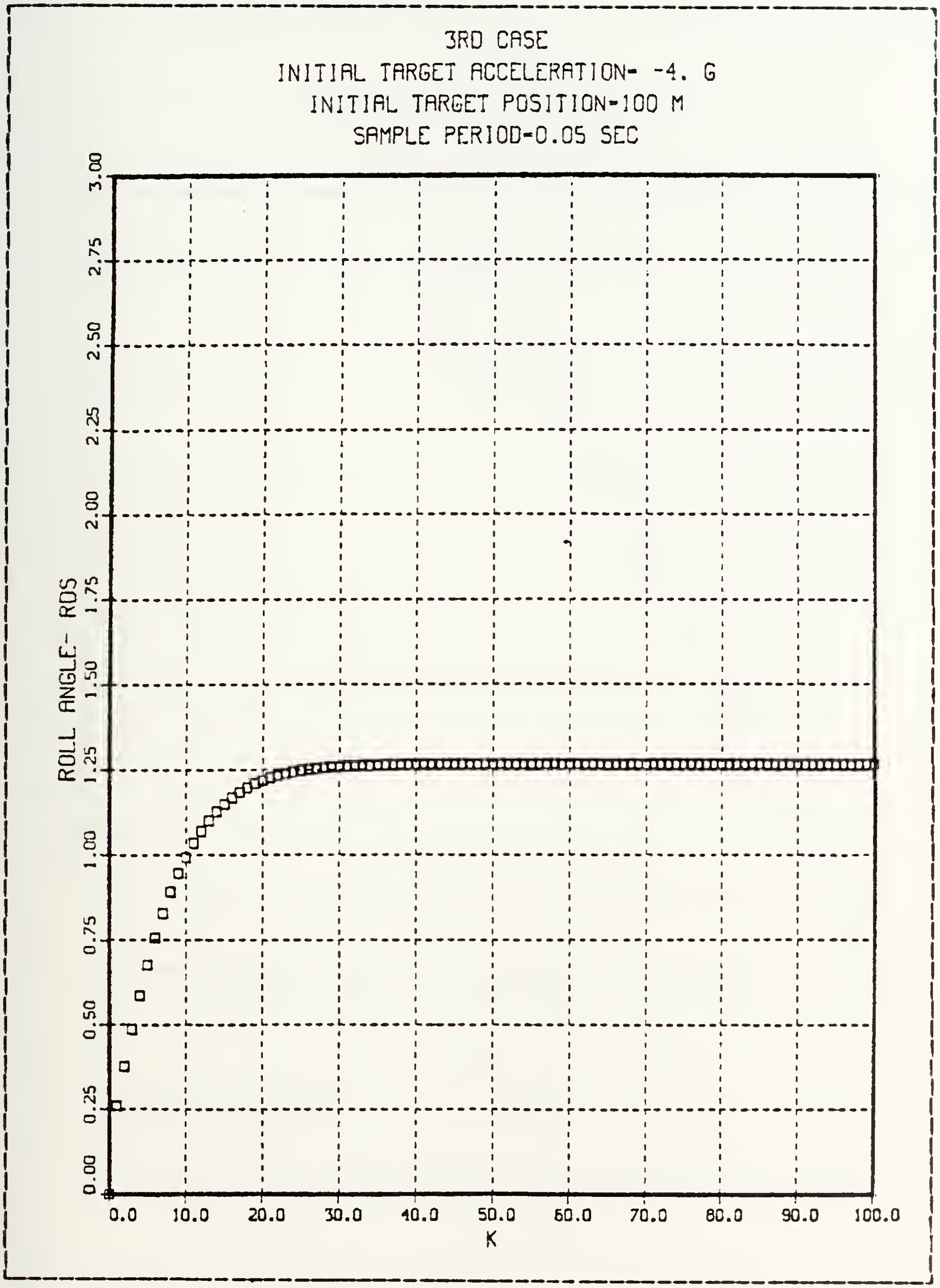

Figure 3.23 Roll Angle- Case 3.

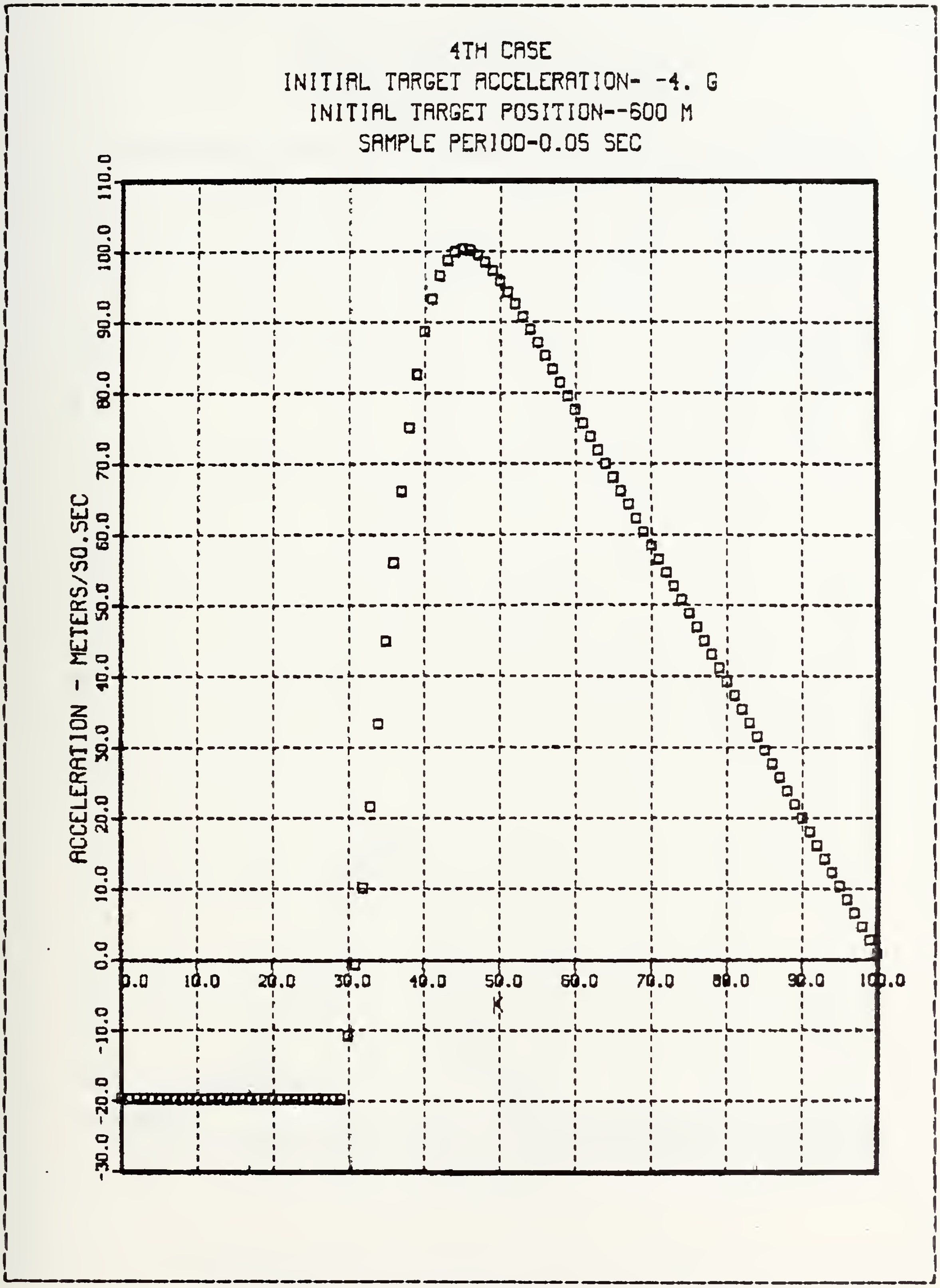

Figure 3.24 Commanded Acceleration- Case 4.

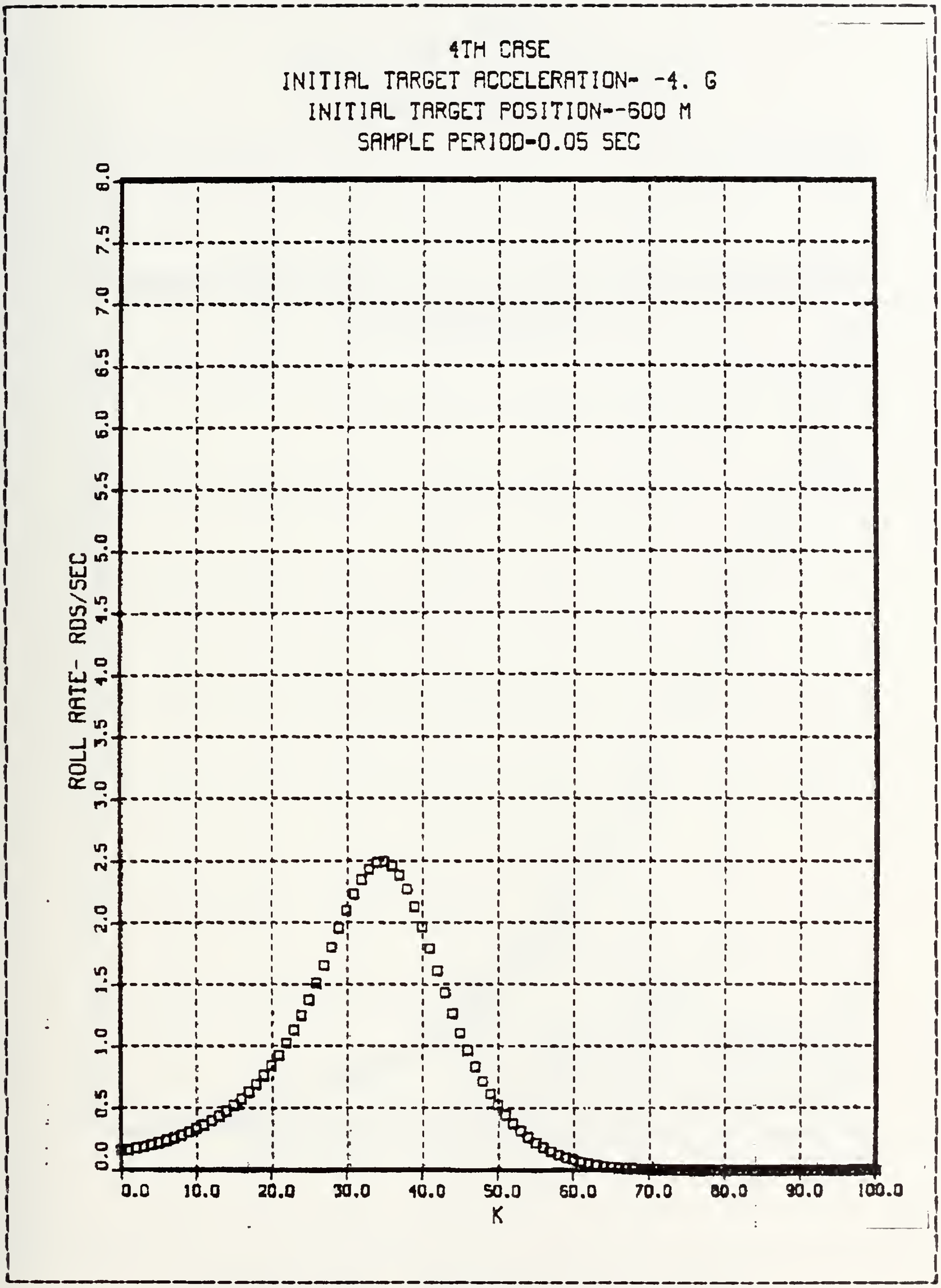

Figure 3.25 Commanded Roll Rate- Case 4.

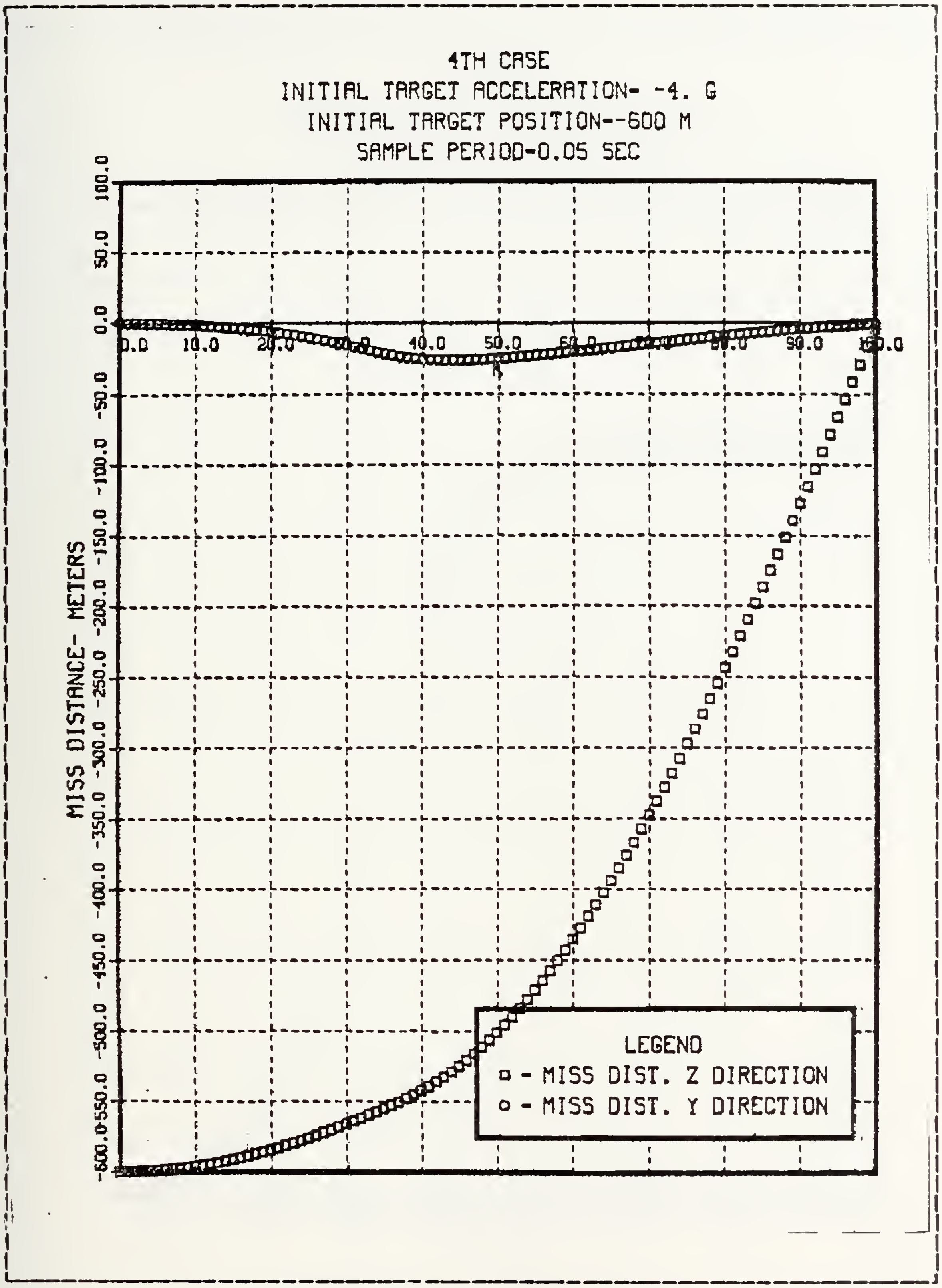

Figure 3.26 Miss Distance- Case 4.

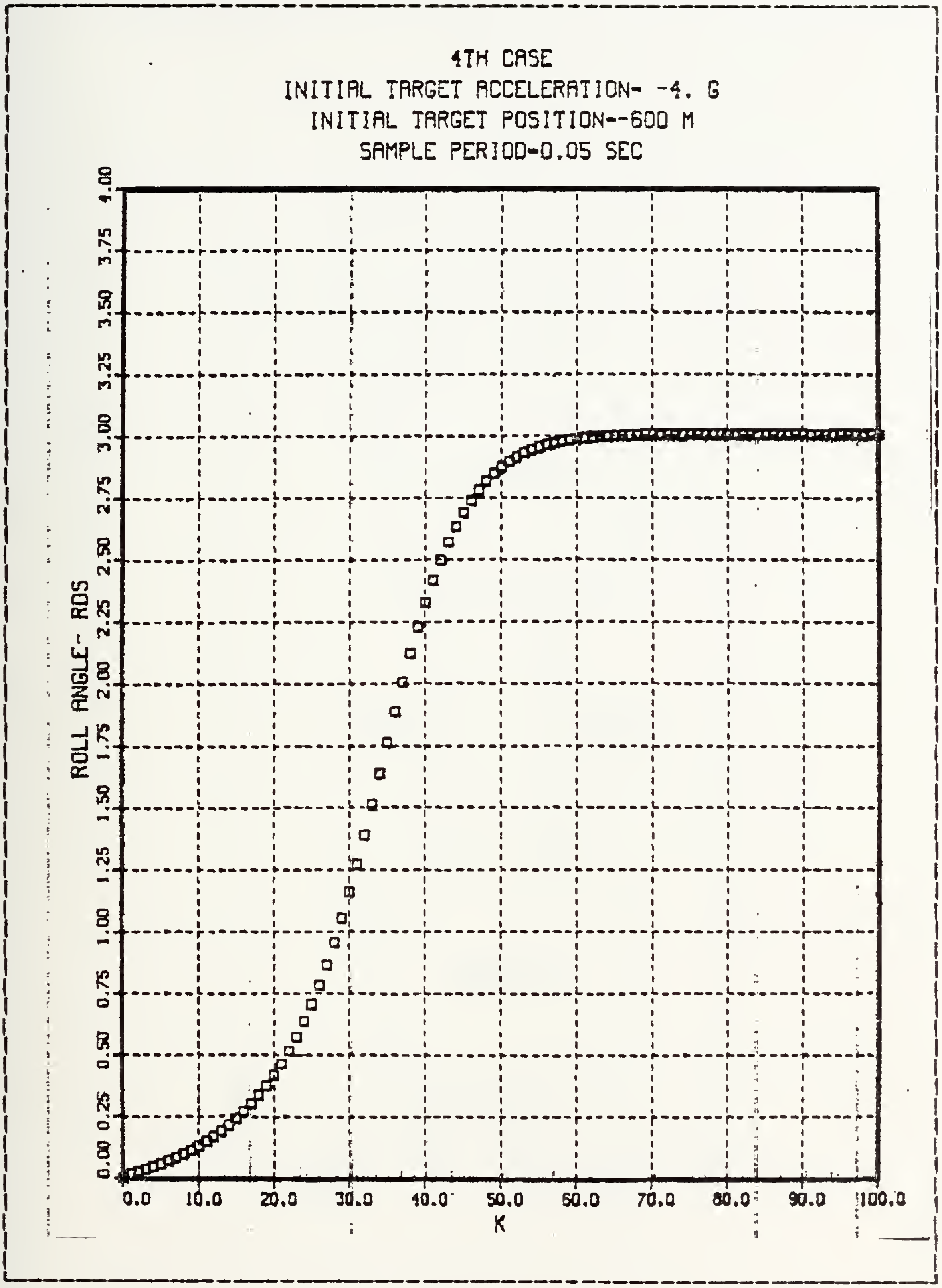

Figure 3.27 Roll Angle- Case 4.

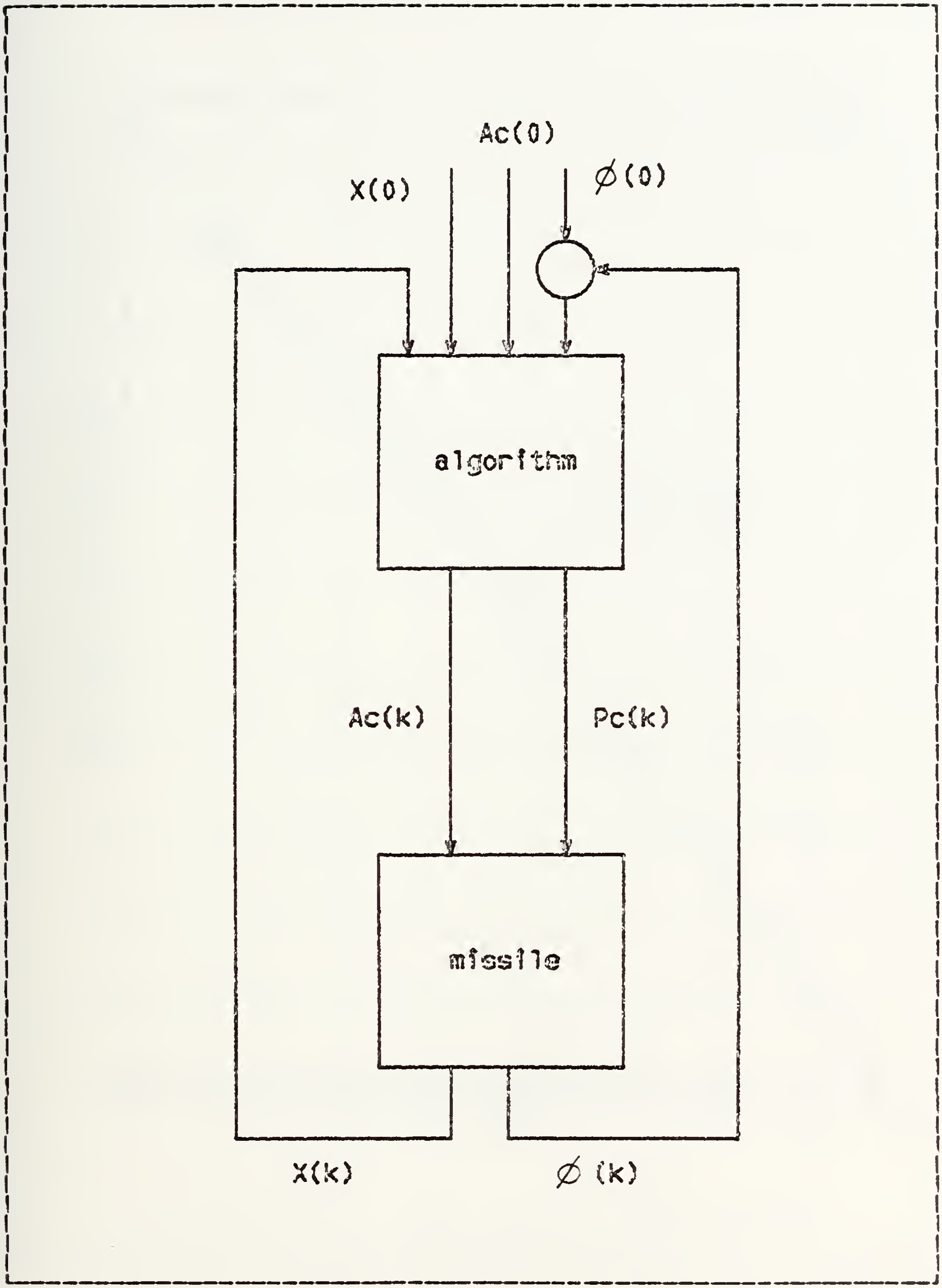

Figure 3.28 Corrected Model.

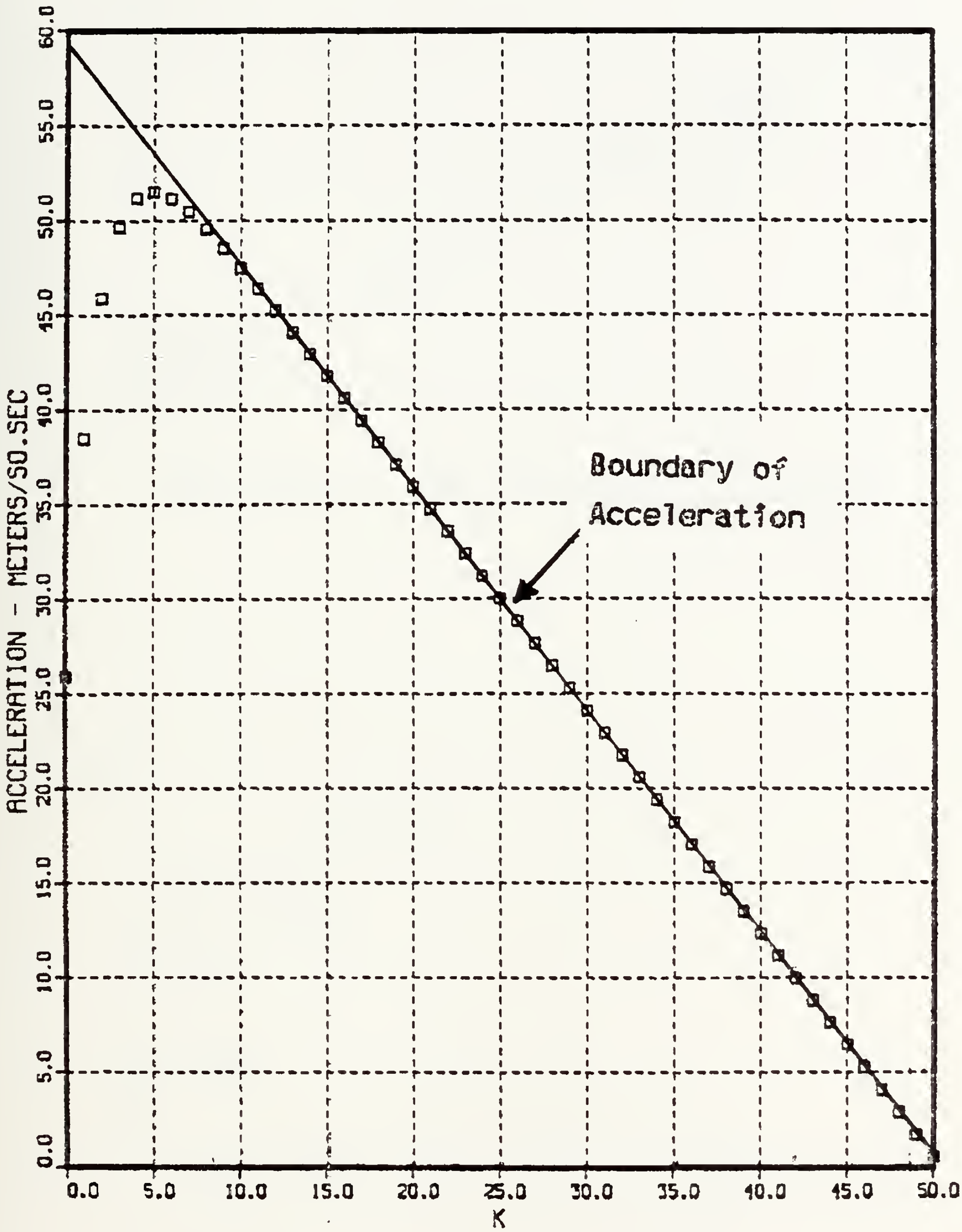

Figure 3.29 Boundary of the Commanded Acceleration.

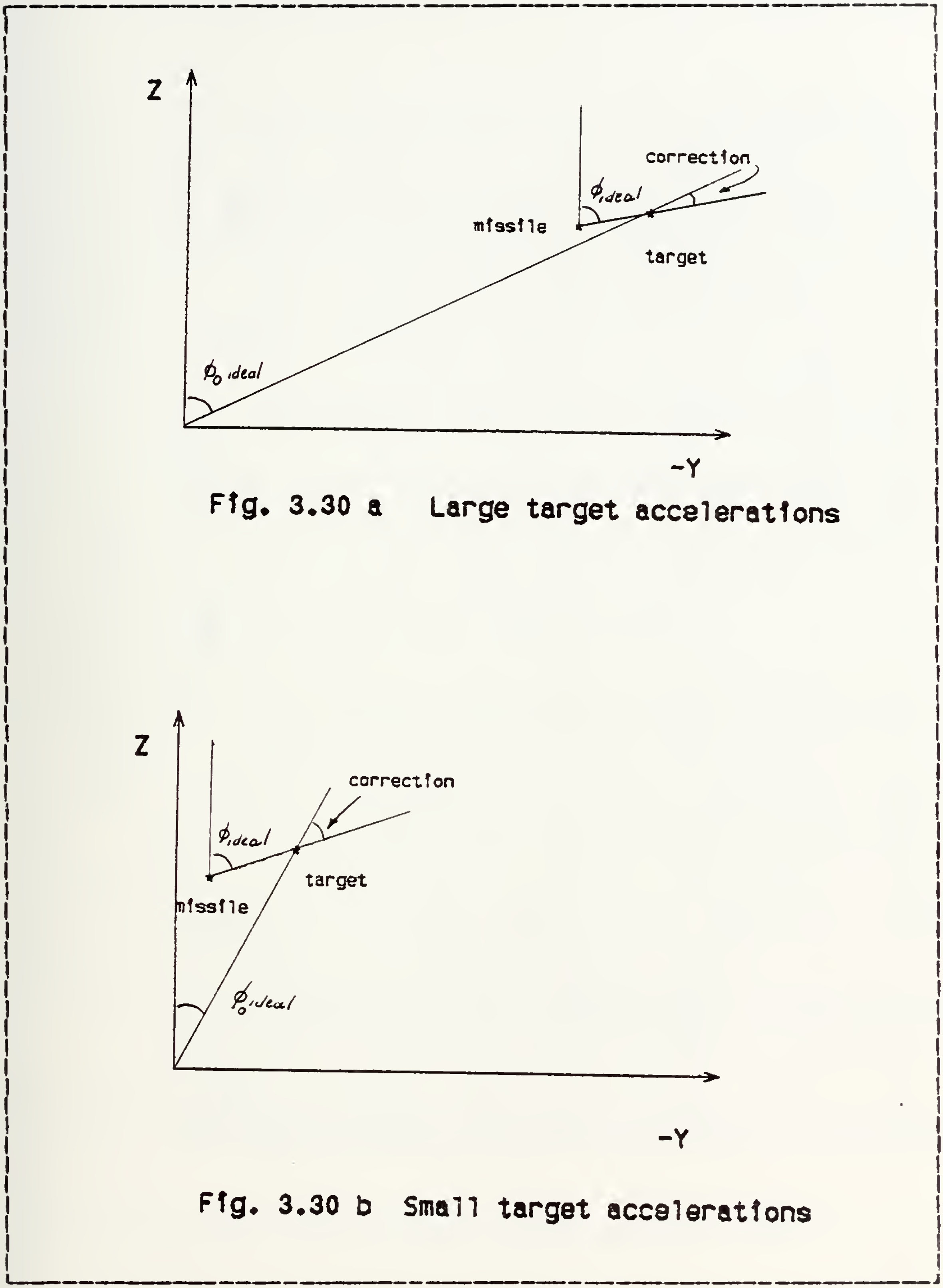

Figure 3.30   Corrections on the Roll Angle.

TABLE I

Results From Tests

case	t	Ac (m/sec)	Pc (rad/sec)	miss distance Y direction (m)	miss distance Z direction (m)	∅ (rad)	CG-to-CG miss distance (m)
0	0	26.47	.345	0.0	100.	0.0	100.
	T1	.133	0.0	-.542	1.50	.439	1.51
1		26.47	.345	0.0	100.	0.0	100.
		.138	0.0	-.383	.528	.416	.652
2	0	26.47	.691	0.0	100.	0.0	100.
	T1	.157	0.0	-.575	.458	.724	.735
3	0	26.47	2.76	0.0	100.	0.0	100.
	T1	.332	0.0	-1.47	.046	1.27	1.54
4	0	-19.6	.167	0.0	-600.	0.0	600.
	T1	.959	0.0	.592	-4.39	3.01	4.44

IV. ANALISYS OF GAINS, SAMPLE RATE AND PITCH ANGLE

A. ANALISYS OF THE GAINS

In chapter 3 has been developed a solution for the optimal control of a system as:

$$x(k+1) = A(k)\ x(k) + B(k)\ u(k) + E\ g$$

It would be interesting to check if the optimal gains reach steady state,but at the moment that the extention for large roll excursions has been introduced, and the system is being feed with optimal commands which are varying each step of time, such idea can not be applyed. However one can do such check in the model for small roll excursions, which is acctually used to compute the optimal commands.

Doing this, one has the time history of those gains as in 4.1 and 4.2

One can notice in fig. 4.1 that the gain FG(2,1) associated with the effect of gravity has no effect on the comanded roll rate, and that FG(1,1), as shown in fig. 4.2, has a large effect on the commanded acceleration. Furthermore, this gain reachs steady state very fast, thus it can be assumed that the gain FG will be equal to the steady state value during all time.

From eqn. 3.19, and assuming steady state:

$$u(k) = -F(k)\ x(k) - FG(k)\ g \qquad (4.1)$$

and substituting g:

$$u(k) = -F(k)\ x(k) - C \qquad (4.2)$$

where the second term in the right hand side is a constant, and its value is exactly equal to the value of the commanded acceleration necessary at t=0 to correct the gravity fall of the missil (or to correct the initial ZEM due to gravity).

It might be supposed that one could solve the optimal control problem for the system represented by eqn. 4.1, just considering one reduced system represented by:

$$x(k+1) = A(k)\ x(k) + B(k)\ u(k) \qquad (4.3)$$

with a bias in the control as:

$$u(k) = -F(k)\ x(k) + C \qquad (4.4)$$

But as showed in the following analysis, this is not possible.

The constant term in the right hand side of eqn. 4.4 is calculated as follows:

from eqn.:

initial ZEM due to gravity= $ZEM = \frac{1}{2}\ g\ t^2$

$$C = \frac{ZEM_g}{\left[\frac{t^2}{2} - \frac{t^3}{6Ti}\right] \cos \phi_0}$$

and the gains F(k) are calculated using a Ricatti equation as usual.

Case 5 was tested using the above considerations, and using the same scenario of case 3.

Figures4.3 and4.4, show the time history of the controls. One can see that the commanded acceleration has begun at same values as in case 3, but the commanded acceleration reachs a peak considerable higher, them dcreases does not following a linear law, with a final at 14.7 meters per second square, being this terminal value due to the constant term representing the effect of the gravity.

Refering to fig. 4.4, the commanded roll rate begins at a same value as in case 3, but as the control Ac is too high, it reachs negative values, going to zero almost at the end of the running time. This behaviour of the control leads to a large miss distance as seen in figure 4.5, and tableII.

Figure 4.6 shows the time history of the roll angle, which rises to values close to 1.5 radians. As the acceleration at this point is larger than the correct value, the the corrections are excessive and the roll angledecreases at the end of the running time to the value of .12 radians.

Thus, one can see that the gain due to the gravity's acceleration can not be replaced by its steady state value.This kind of simplification can thus not be done in the system being studied.

B. EFFECT OF THE SAMPLE RATE

Throughout all the simulatios a sample period of .05 seconds has been used.In this section a brief study of the effect of the change of this sample rate is given.

Two best cases have been selected to ilustrate the effect of the sample period.

The first case, case 6, has been run with a sample period of .1 seconds and consits of the same scenario as case 3.

As one can see in figure4.7 and tableIII, there is no noticable change in the commanded acceleration, but the commanded roll rate begins at smaller value than in case 3, as is seen in fig.4.8 This initial decrease in roll rate, leads to a large miss distance in Y direction as shown in fig.4.9, and to a small value of roll angle (see fig.4.10).

The second case, case 7,has a smple period of 0.025 seconds. There is no noticeable change in the time history of the control Ac as shown in fig.4.11. The commanded roll rate begins at a higher value than in case 3 as shows fig. 4.12, which leads to a final miss distance in Y direction of -2.22 meters and in Z direction smaller than case 6 (see fig4.13). Figure4.14 shows that the final roll angle is increased and the missile cross the target with 1.28 radians and with a CG-to-CG distance of 2.5 meters.

In both cases, the miss distance was increased over the nominal value obtained with a sampling rate of 0.05 seconds. Thus, it would appear that there is an optimal value for the sampling rate, which may be conected with the geometry of the scenario and with time to go.

C. EFFECT OF THE INITIAL PITCH ANGLE

It is important at this step to remenber that througout this work has we have been discussing a dimensional model, where there is no information on the X coordinate, so it is impossible to verify the behavior of the pitch angle.

In this work, since in all the previous scenarios the initial angle Θ was equal to zero, this value has been kept as a constant during all time, and considering that whithout any information of a third dimension it was not possible to correct the time to intercept, this time was also kept constant and equal to the nominal value of 5 seconds.

Notice that this assumption is likely to be correct if one has the horizontal initial distance from target to missile compared with the initial vertical distance between target and missile large enough in order to have small angles.

The question that rises is, how could this pitch angle affect the system if it was not small?

As seen in fig.4.15, the missile velocity in the X direction would be:

$$V_{mx} = V_m \cos\theta + g \cos\theta \sin\theta \ t \qquad (4.5)$$

which has an effect not only from the pitch angle, represented by the $\cos\theta$, but also from the gravity's acceleration, which will leads to a different time to intercept.

Considering the same physical scenario as in case 4, but changing the initial pitch angle, in order to have the missile pointing to the target (see fig.4.16), and keeping the missiles velocity of 1000 m/sec in the X direction, one has a completly different geometry of the problem as seen from the flight path reference frame.

With this new situation (see fig. 4.16), case 8 has been run. Figures 4.17 and 4.18 show that now the missile is comanding large positive accelerations, and the roll rate at the begining of the flight is too high, going to zero in a very small period of time. The miss distance as seen in fig.4.19 are increased in the initial part of the fligth and as the missile corrects its trejectory it is decreased to reach a final CG-to-CG miss distance of 2.17 m. The roll angle, due to the large control Pc is oscilatory in the begining and becames constant with a value of .57 radians (see fig.4.20 and table IV).

Notice that the high values of acceleration needed are in some part due to a vertical component of target's velocity, which is seen from the flight path frame as the target was manouvering in the Z direction with constant velocity. These large accelerations leads to roll rates too large for the physical integrity of the missile. This means that although althoug the good results obtained, if compared with case 4, they are not practical.

In order to get rid of the vertical manouever of the target, case 9 has been run. In this case, the scenario is the same as before with the target also pointing down, with the same pitch angle as the missile, and has a X velocity equal to the previous case (see fig.4.16).

From fig.4.21, one can see that the decrease in the control Ac is substantial if compared with case 8, but the commanded roll rate is still too large as shows fig.4.22 The time history of the miss distance is better, resulting in a final CG-to-CG distance of about 50% of case 8 (see fig.4.23). The roll angle is not oscilatory as seen in fig.4.24 and the missile cross the target with 1.3 radians in roll.(see tableIV).

The results obtained in the two previous cases, suggests that the algorithm developed in this work could be readly applied to air-to-surface missiles. In the latter, the scenaric would be favorable to the missile than in either previous cases, since one can consider that the target could be essencially stationary in comparison with the missile speed.

Case 10 has been run with this assumption, and the scenario as in fig. 4.16. The target is with zero velocity and acceleration, and the missile begins its flight 600 meters above it, with the same initial pitch angle as before.

Fig.4.25 shows the time history of the commanded acceleration, where one can see that as there is no roll rate to

be commanded, the acceleration is following a straight line with very resoanable values. The miss distance is show in figure4.27, which shows the final CG-to-CG distance of .31 meters.

Based in the results of these tests, one can see that there will be some effect of the pitch angle on the miss distance, not only due to its effect on the time to intercept, but also because at the moment that there is a pitch angle different from zero, even if the target is keeping its flight level, in the flight path frame a component of the target's velocity will show up leading the missile to command large accelerations and roll rates. Althoug this results harm the performance in an air-to-air missile, in the case of air-to-surface missiles, when the target has been considered with no motion, good results has been obtained.

D. EFFECT OF TIME TO INTERCEPT

In the simulation of case 1 and 2 in chapter 3,it has been observed that when the target was at small accelerations, the missile did larger corrections on its roll angle, with respect to the ideal initial roll angle, than when the target was with large accelerations. One can think that must be some kind of compromise between the velocity rate of target and missile (which will reflected on the time to intercept), and the relative position between them, which will affect the miss distance.

In order to do a brief analisys on this, case 11 and 12 has been run.

In case 11, the scenario of case 2 has been kept, with the exception that the missile's velocity was change to 2000 m/sec., which means that Ti was changed to 2.5 seconds.

Figure 4.29 shows that the acceleration is largely increased due to the small time required to correct the ZEM, and the commanded roll rate is almost twice of case 2 (see fig. 4.30). The final miss distance is more than four times the value obtained in case 2, as seen in fig. 4.31 and table V. The final roll angle is about half as in case 2, since the projected final position of the target in the Y direction is less than in case 2 (see fig. 4.32).

Case 12 was run with a scenario less favorable to the missile, where all the conditions of the previous case was kept, except the target position that has been increased to 200 meters above the missile.

Now, the commanded acceleration are much larger, with a initial Aco of 103 m/sec², being almost impossible to see the difference of the time history of the acceleration from one straight line, as shown in fig. 4.33 The commanded roll rate is small, about the same as in case 2 (see fig. 4.34). The change in the miss distance is noticeable, with a final CG-to-CG distance of 4.8 meters as in table V , and figure 4.35. The final roll angle explain the shape of the acceleration curve, since with the small roll angle as shown in fig. 4.36, the system is behaving as for small roll excursions.

Notice that from this analyzis, one has to realize that there is some kind of envelope where the 2-D system is valid. And in order to determine this envelope, one has to take in account not only the geometry of the scenario, but also the time to intercept, which is determined not only by the relation of velocities of missile and target, but the pitch angle too.

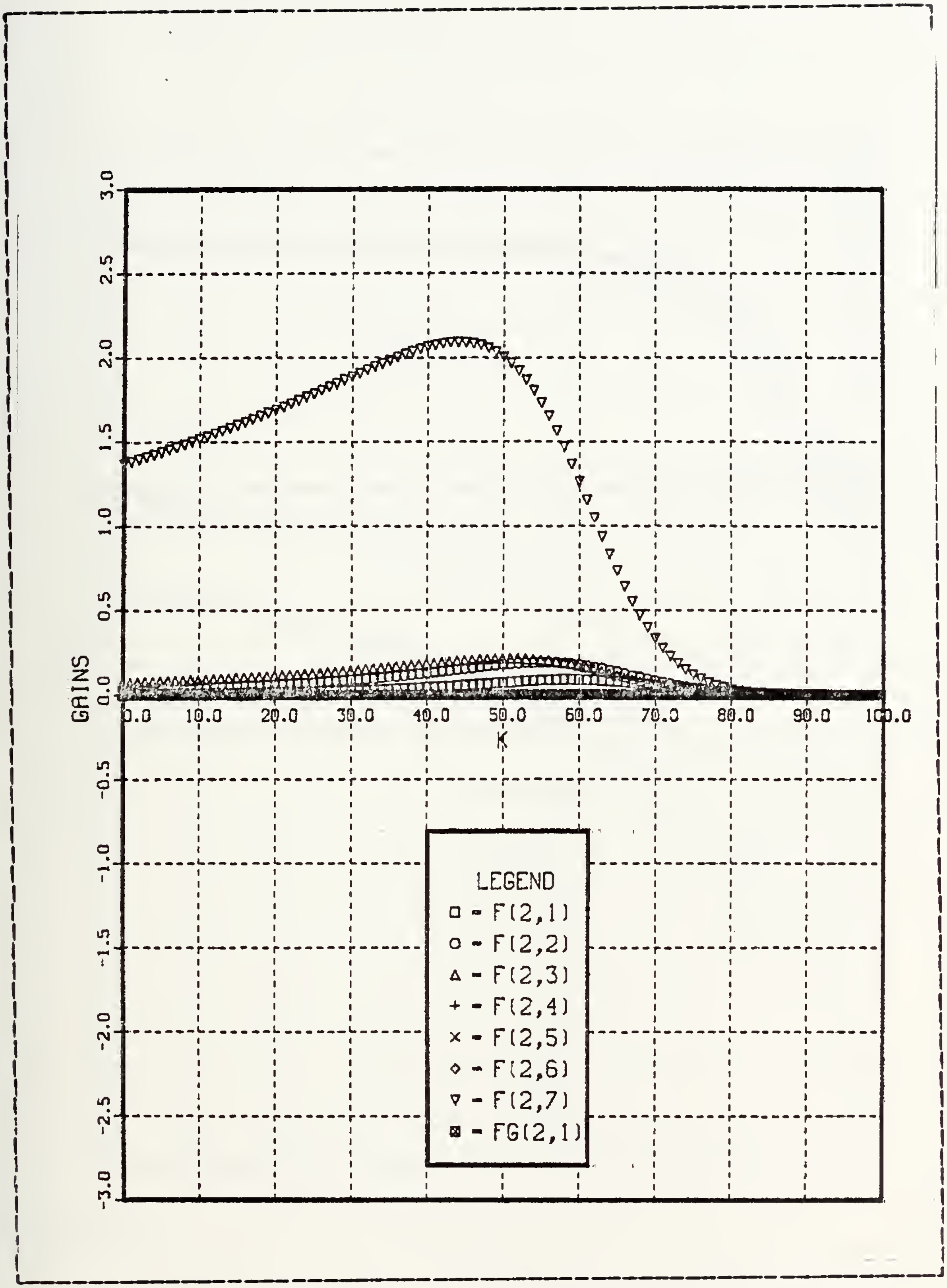

Figure 4.1 Gains affecting the commanded acceleration.

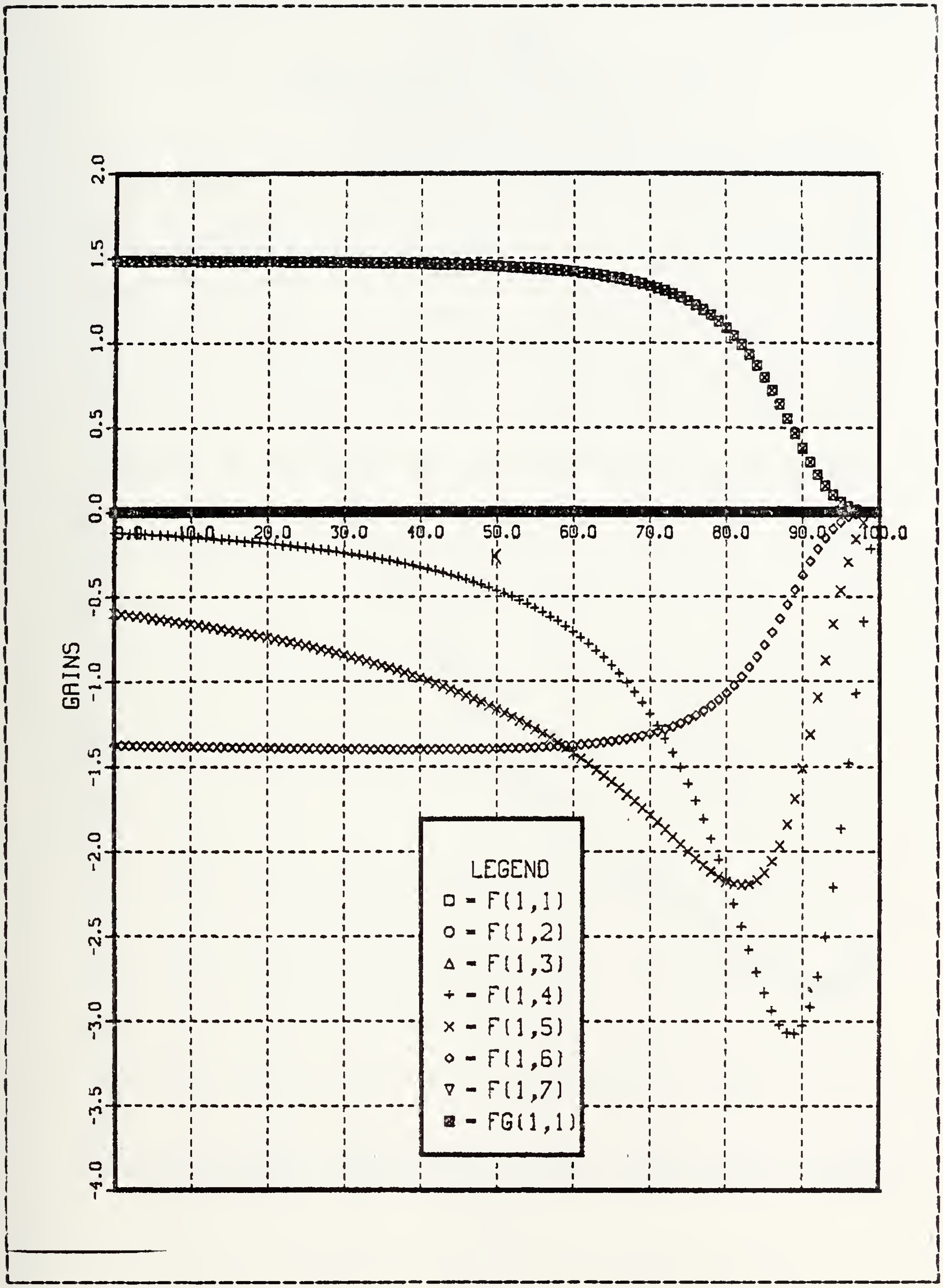

Figure 4.2 Gains Affecting the Commanded Roll Rate.

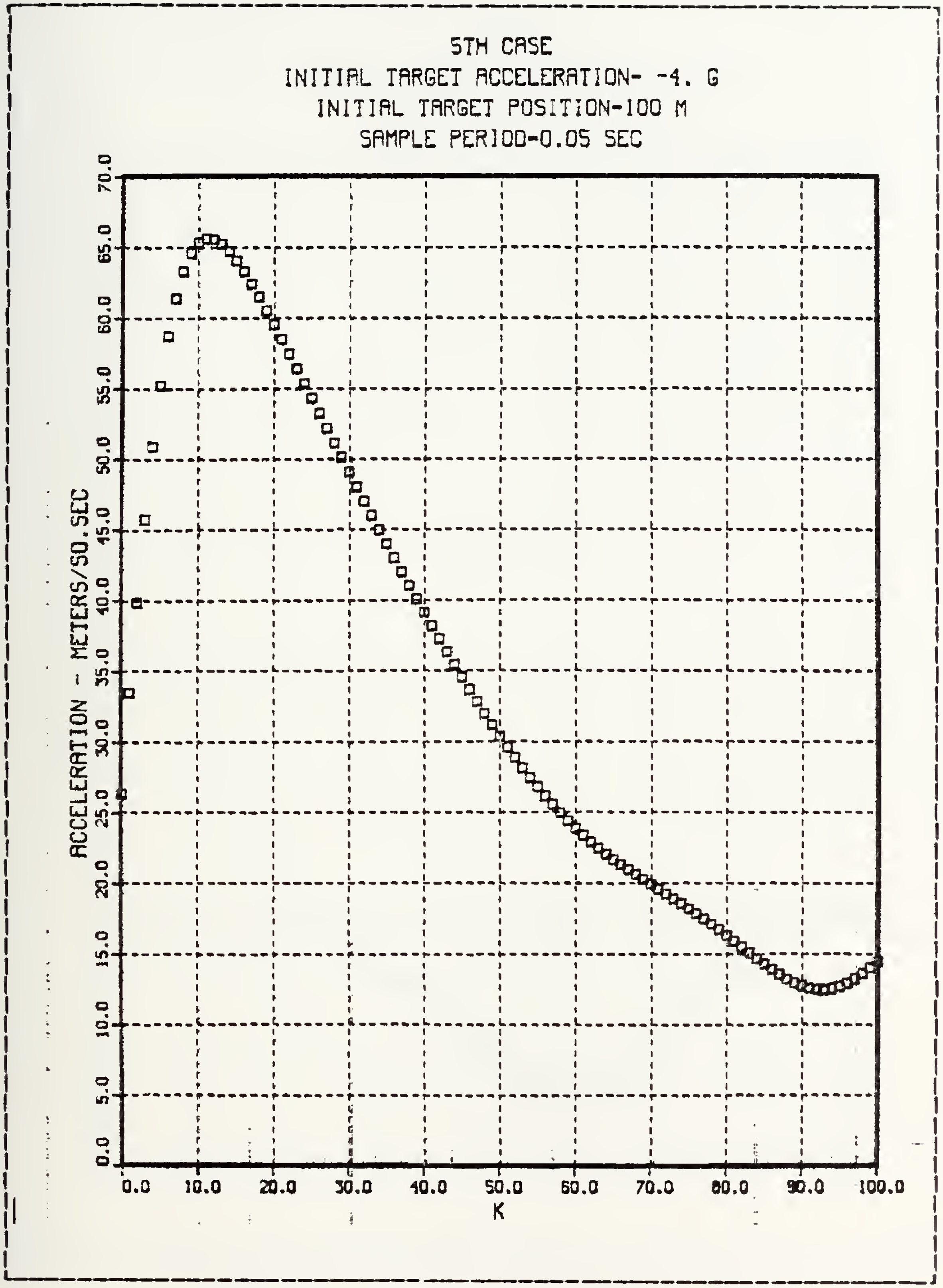

Figure 4.3 Commanded Acceleration-Case 5.

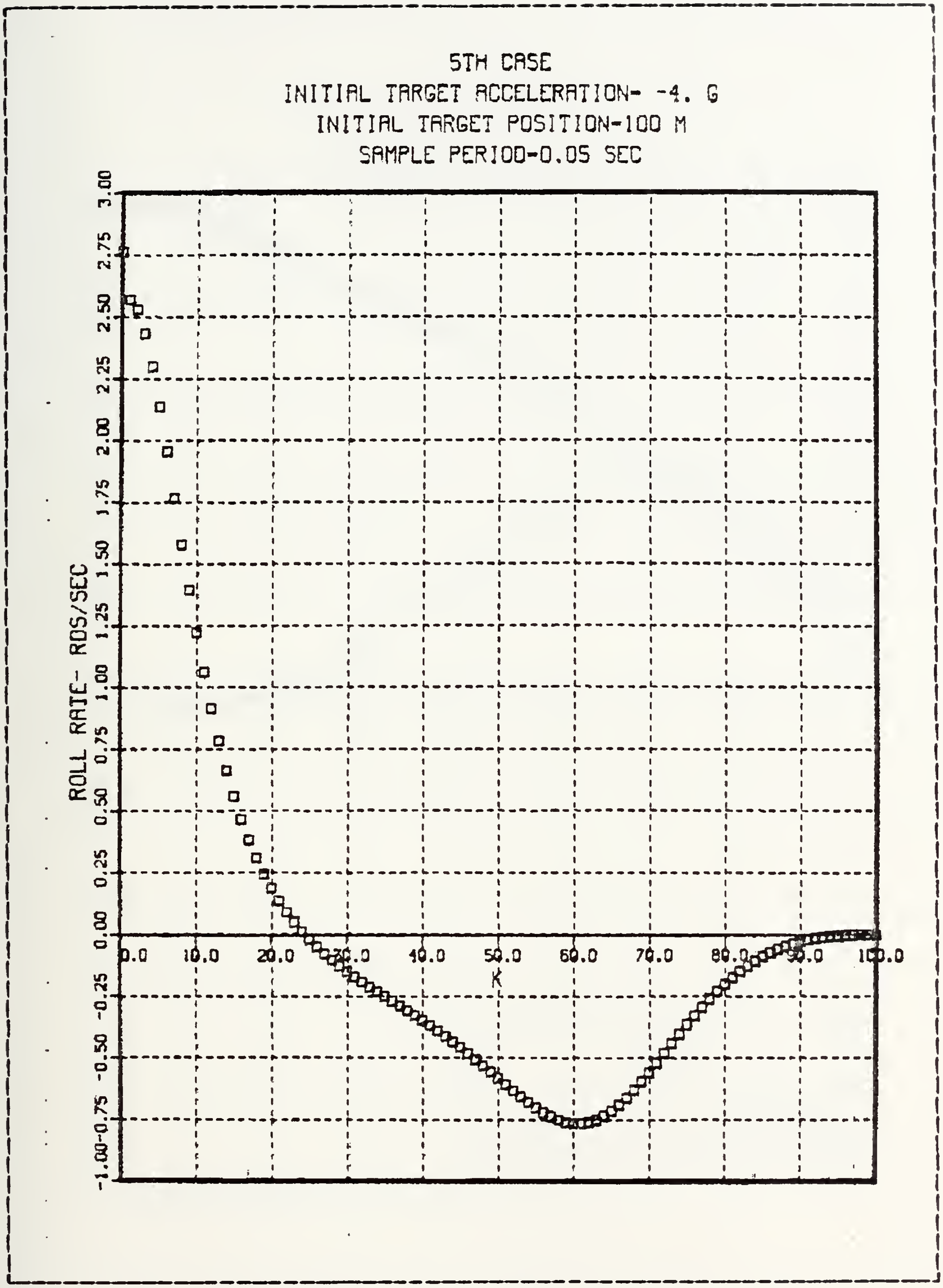

Figure 4.4 Commanded Roll Rate-Case 5.

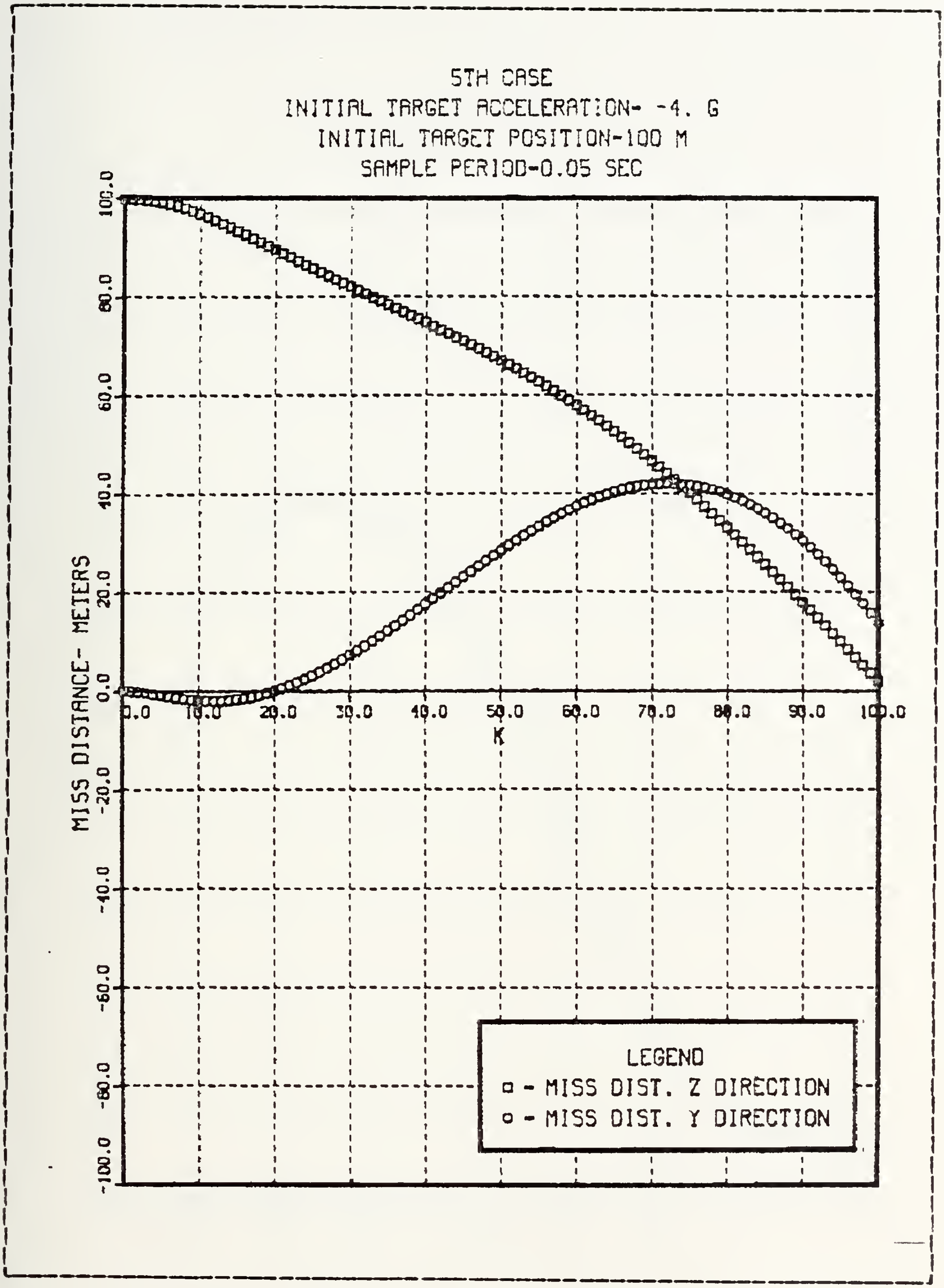

Figure 4.5 Miss Distance-Case 5.

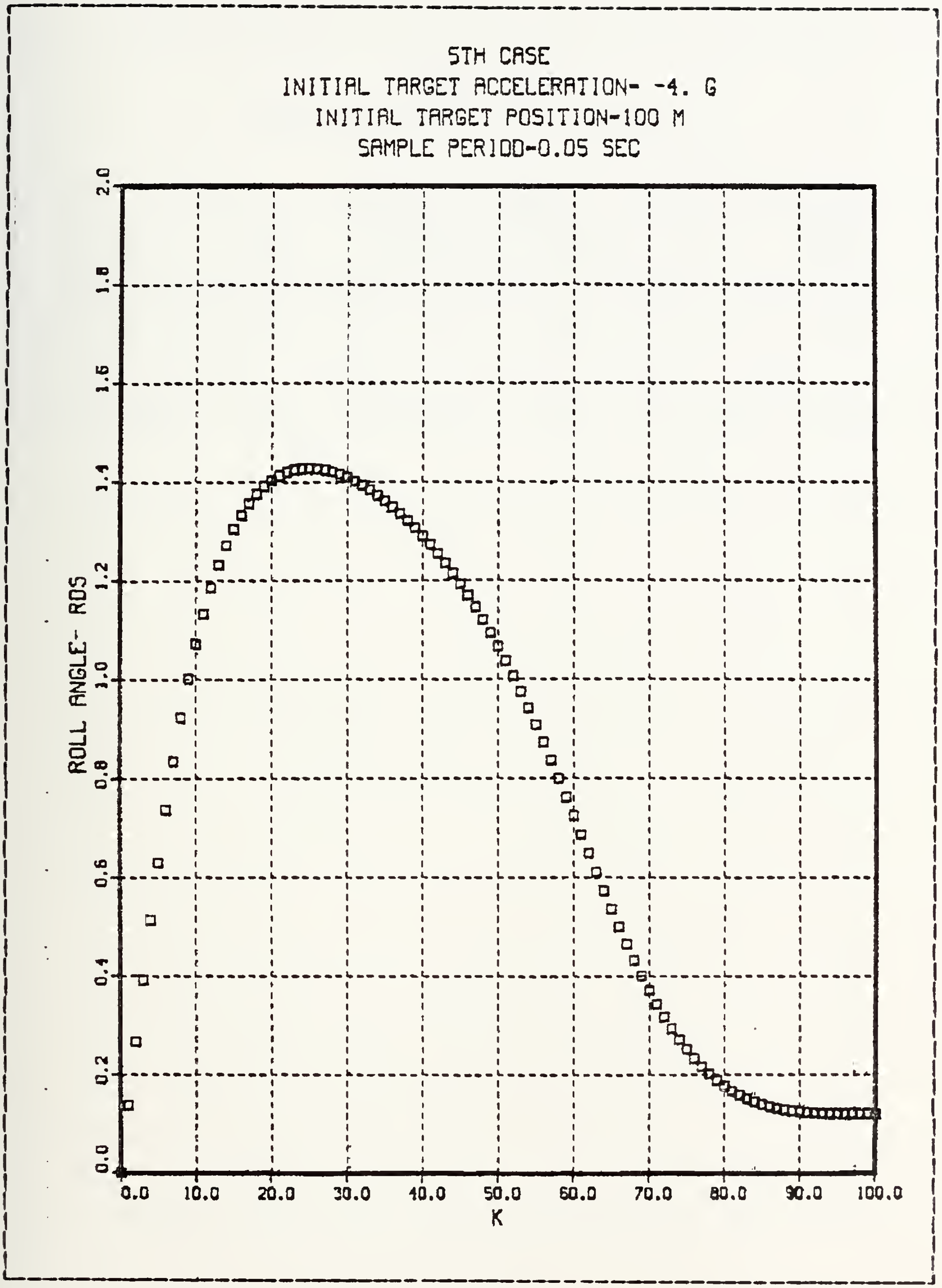

Figure 4.6 Roll Angle-Case 5.

TABLE II

Results from Biased Control

t	Ac (m/sec)	Pc (rad/sec)	miss distance Y direction (m)	miss distance Z direction (m)	Ø (rad)	CG-to-CG miss distance (m)
0	26.37	2.76	0.0	100.	0.0	100.
Tf	14.47	0.0	13.78	2.00	.119	13.93

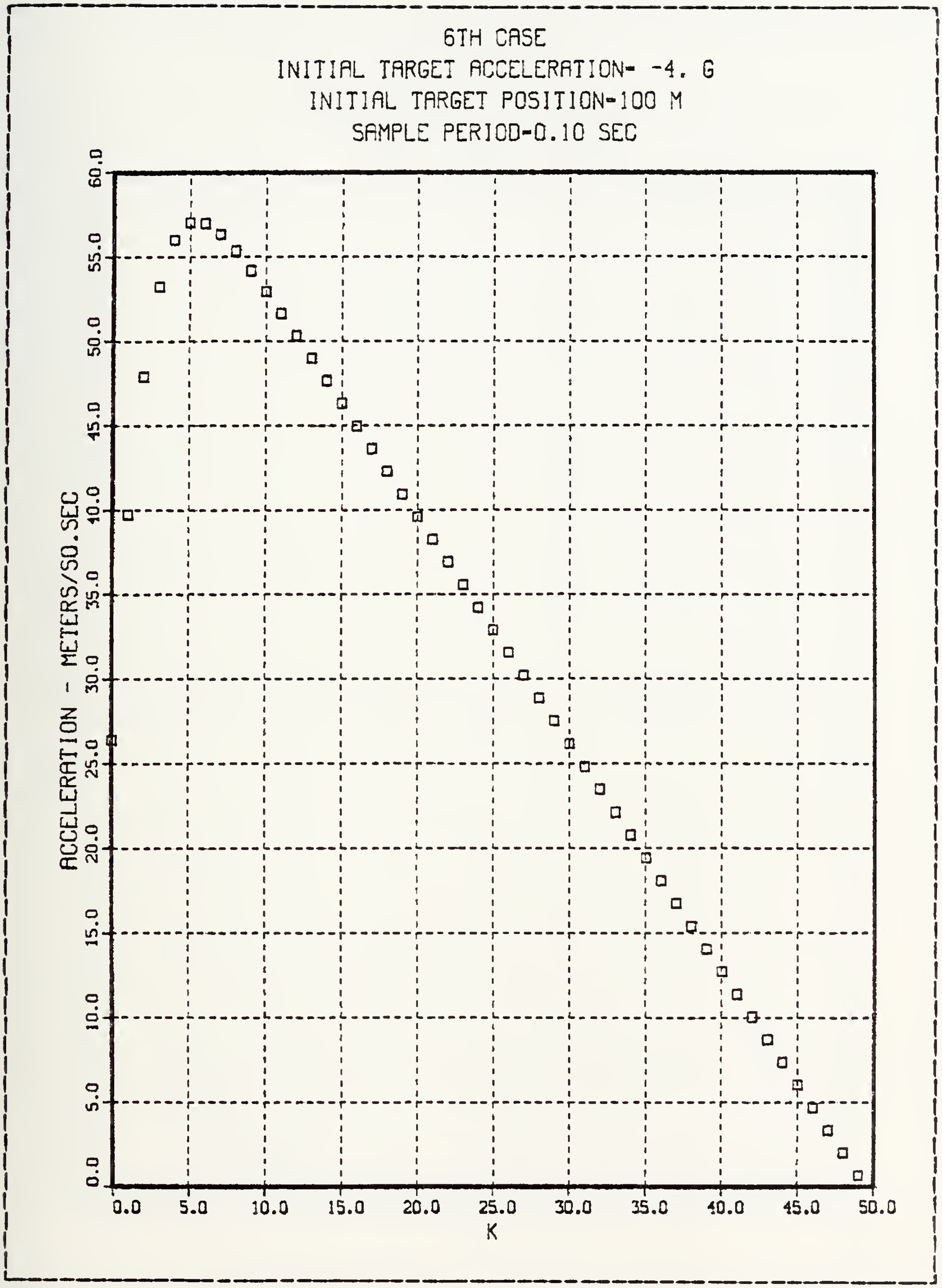

Figure 4.7 Commanded Acceleration-Case 6.

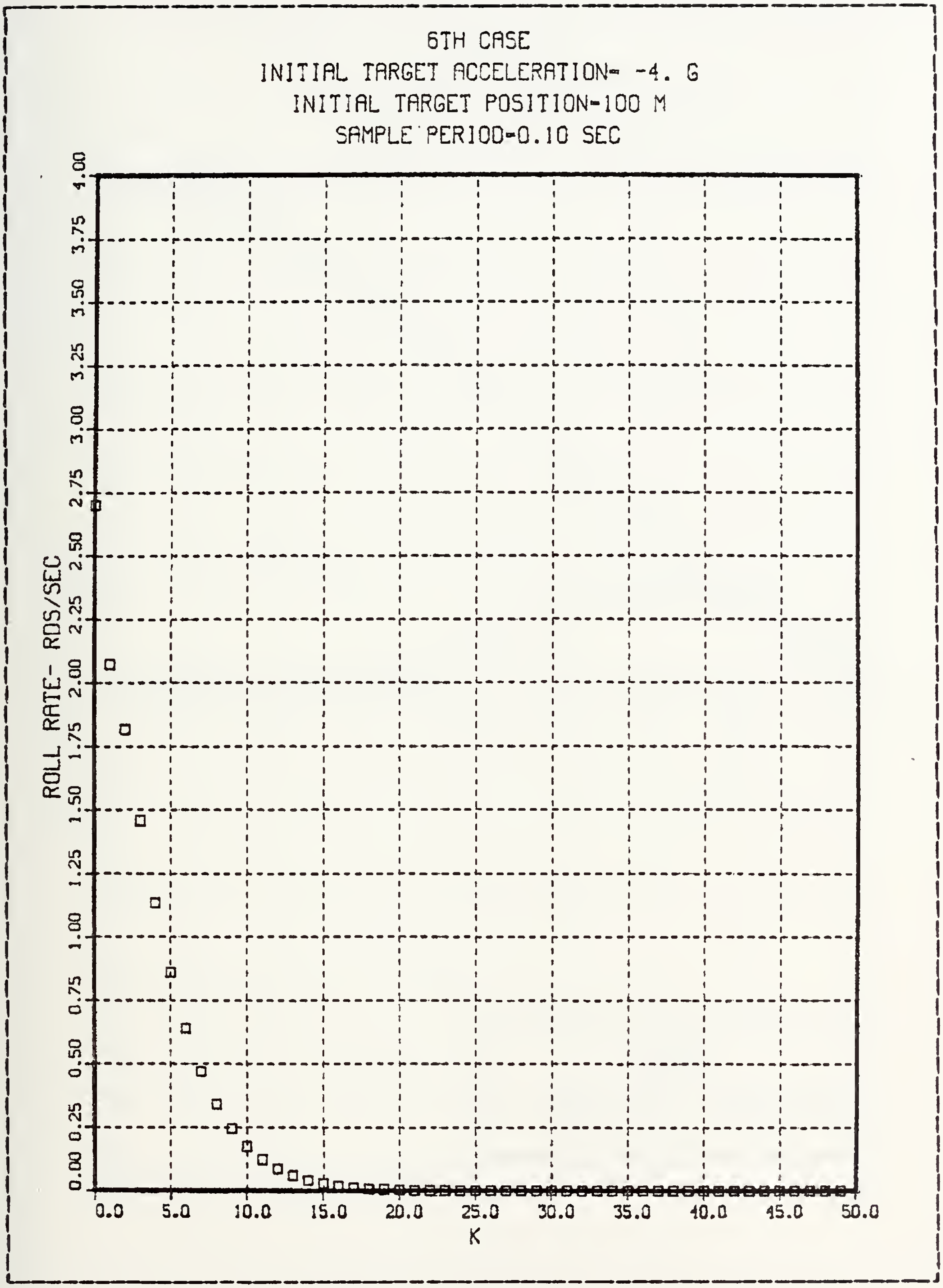

Figure 4.8 Commanded Roll Rate-Case 6.

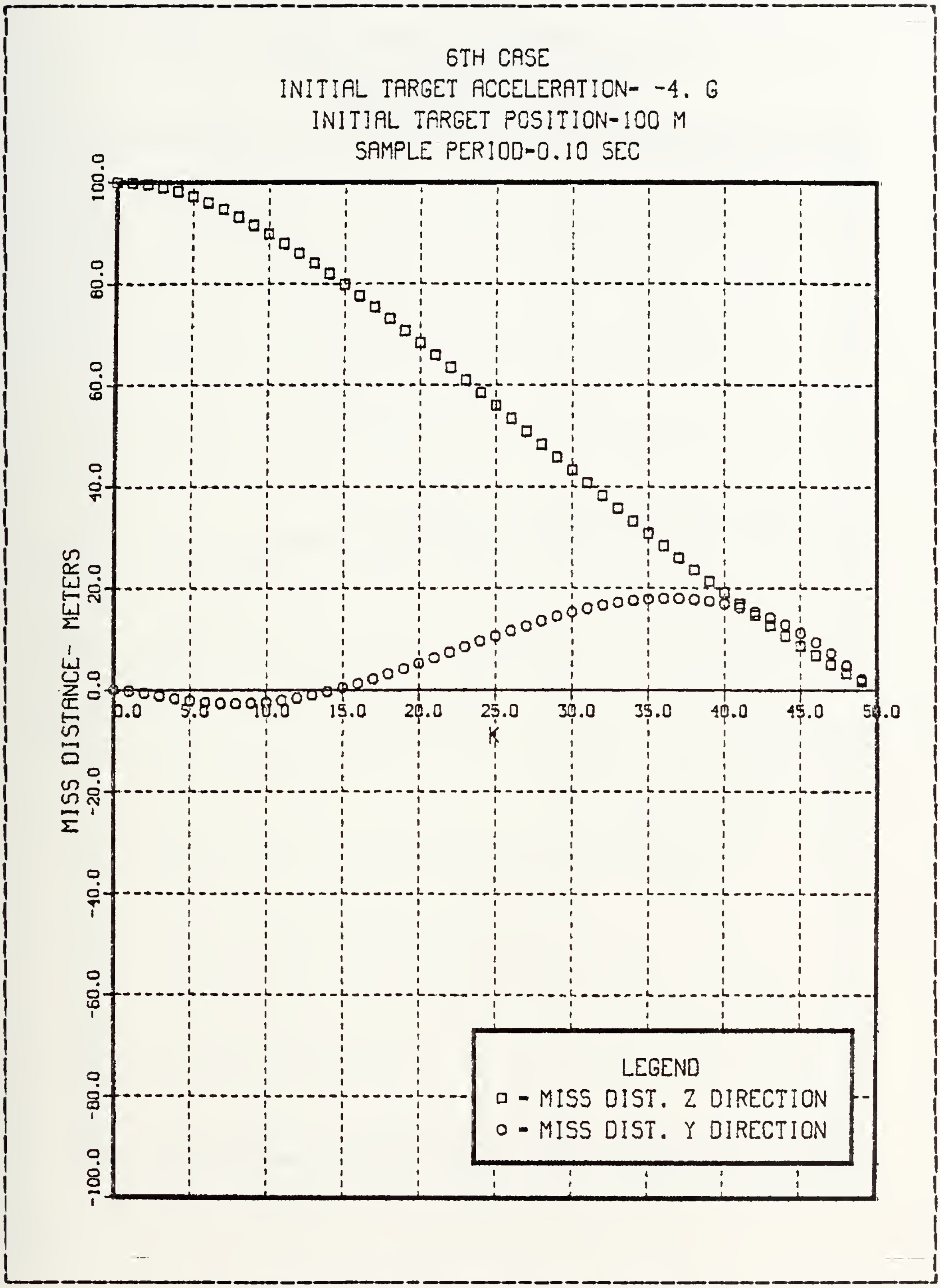

Figure 4.9 Miss Distance-Case 6.

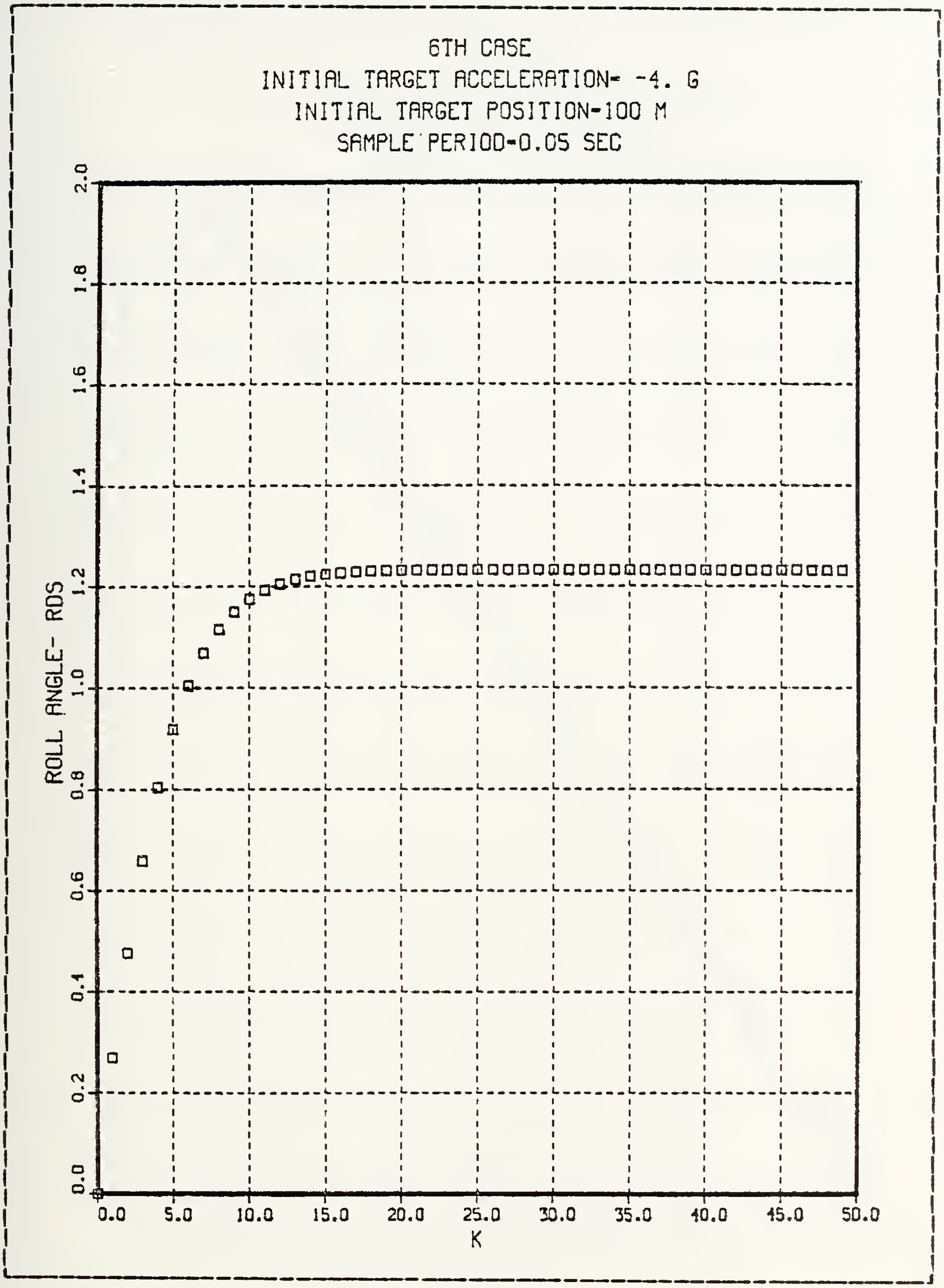

Figure 4.10 Roll Angle-Case 6.

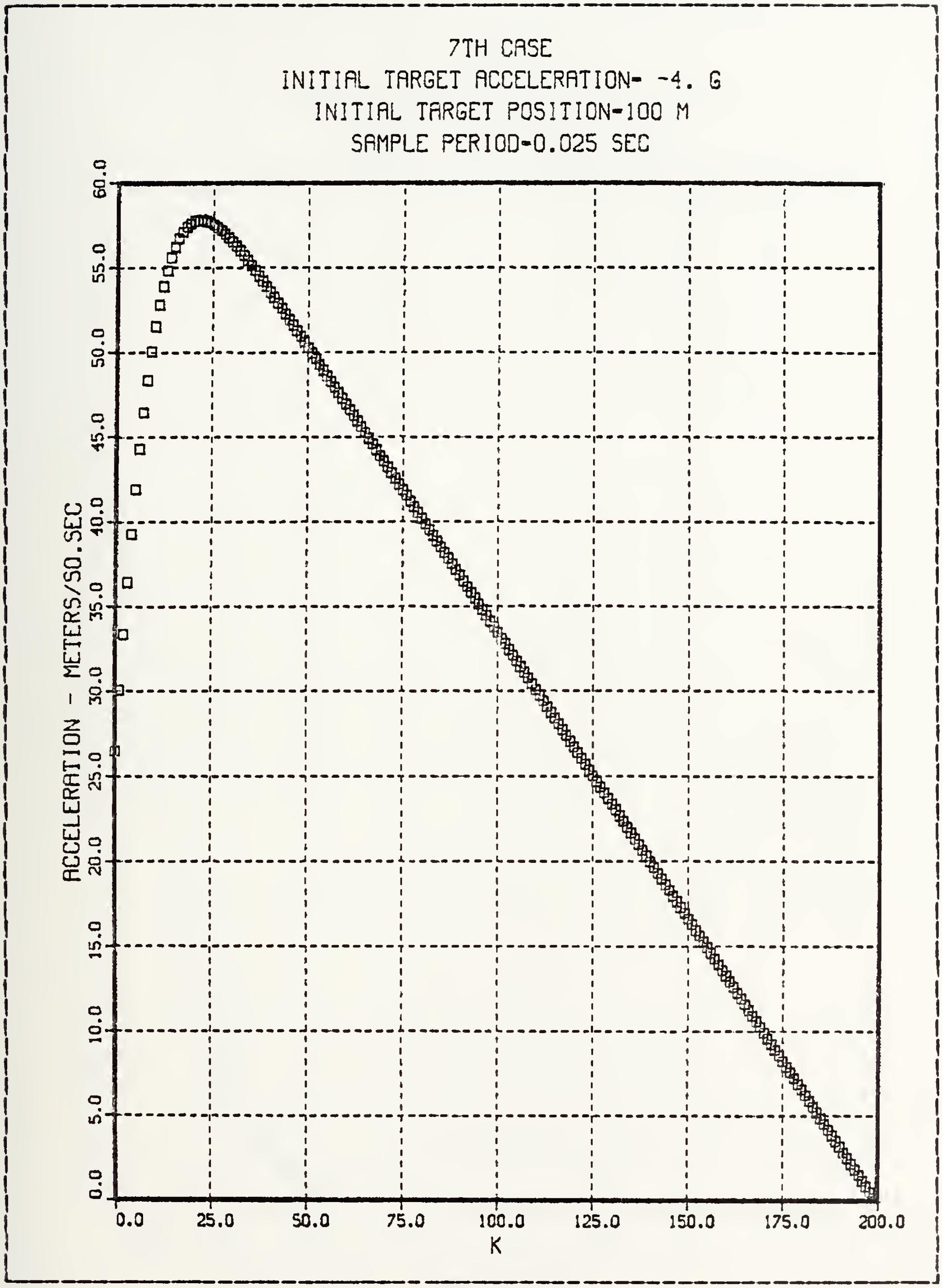

Figure 4.11 Commanded Acceleration-Case 7.

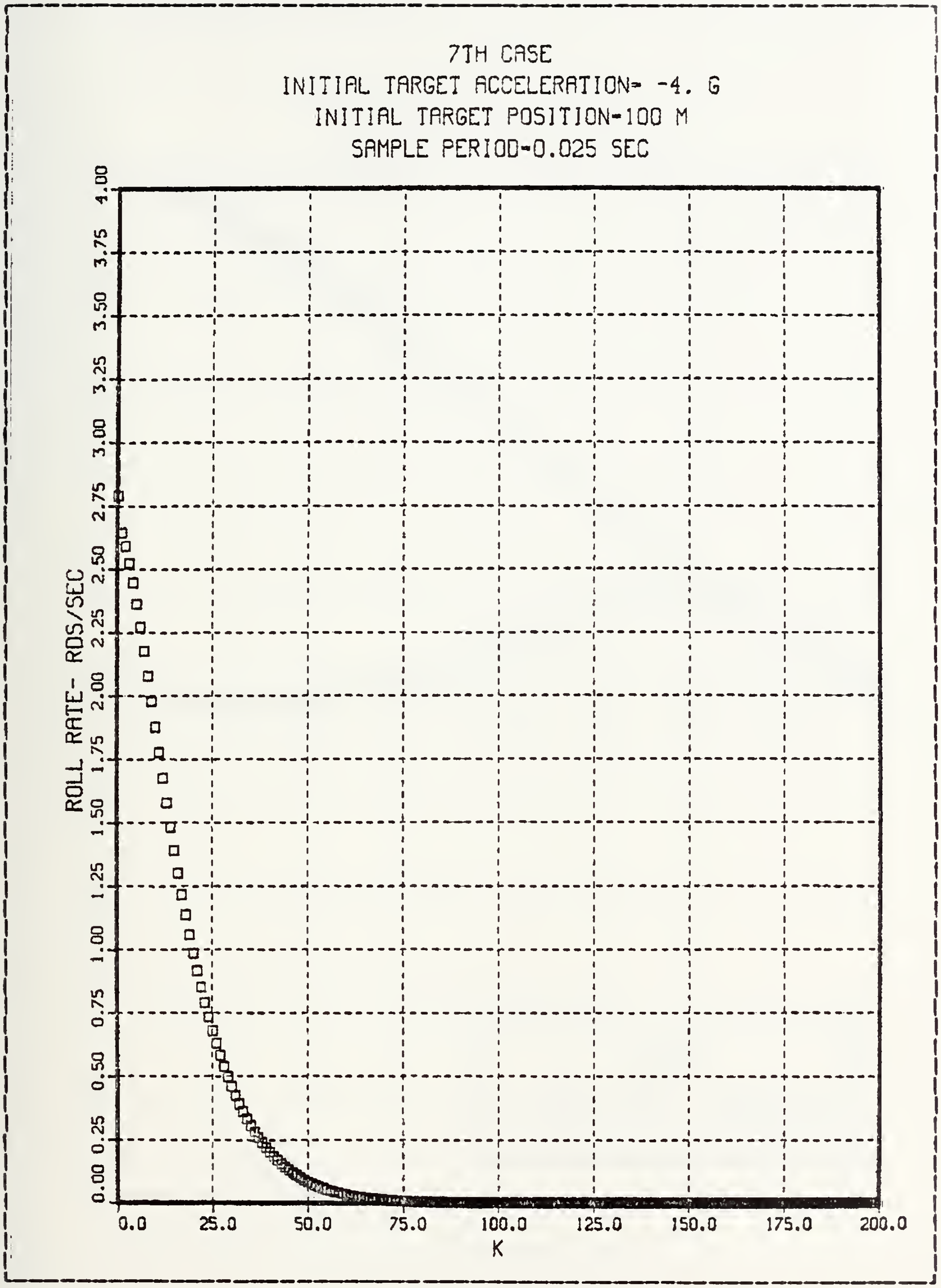

Figure 4.12 Commanded Roll Rate-Case 7.

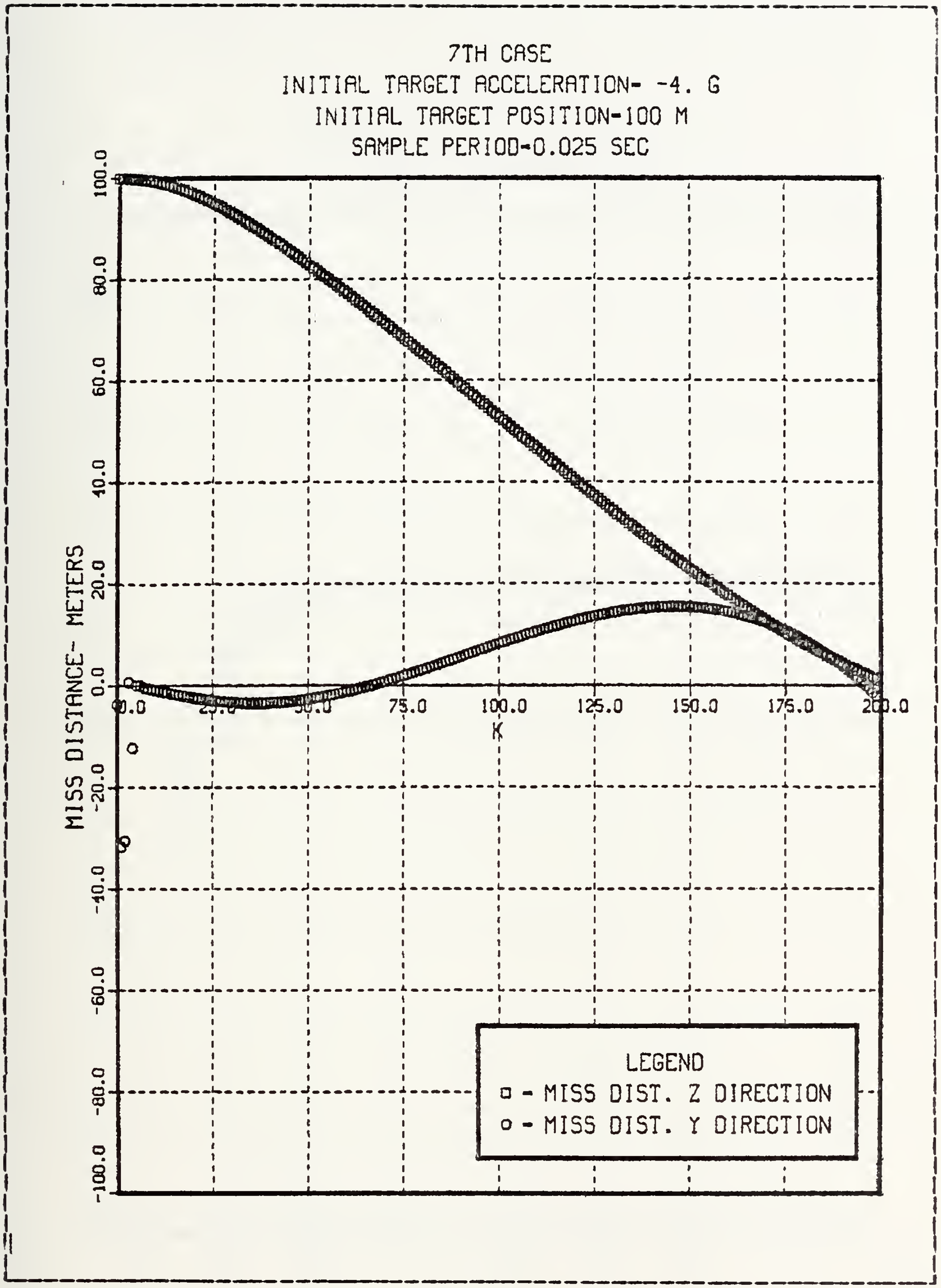

Figure 4.13 Miss distance-Case 7.

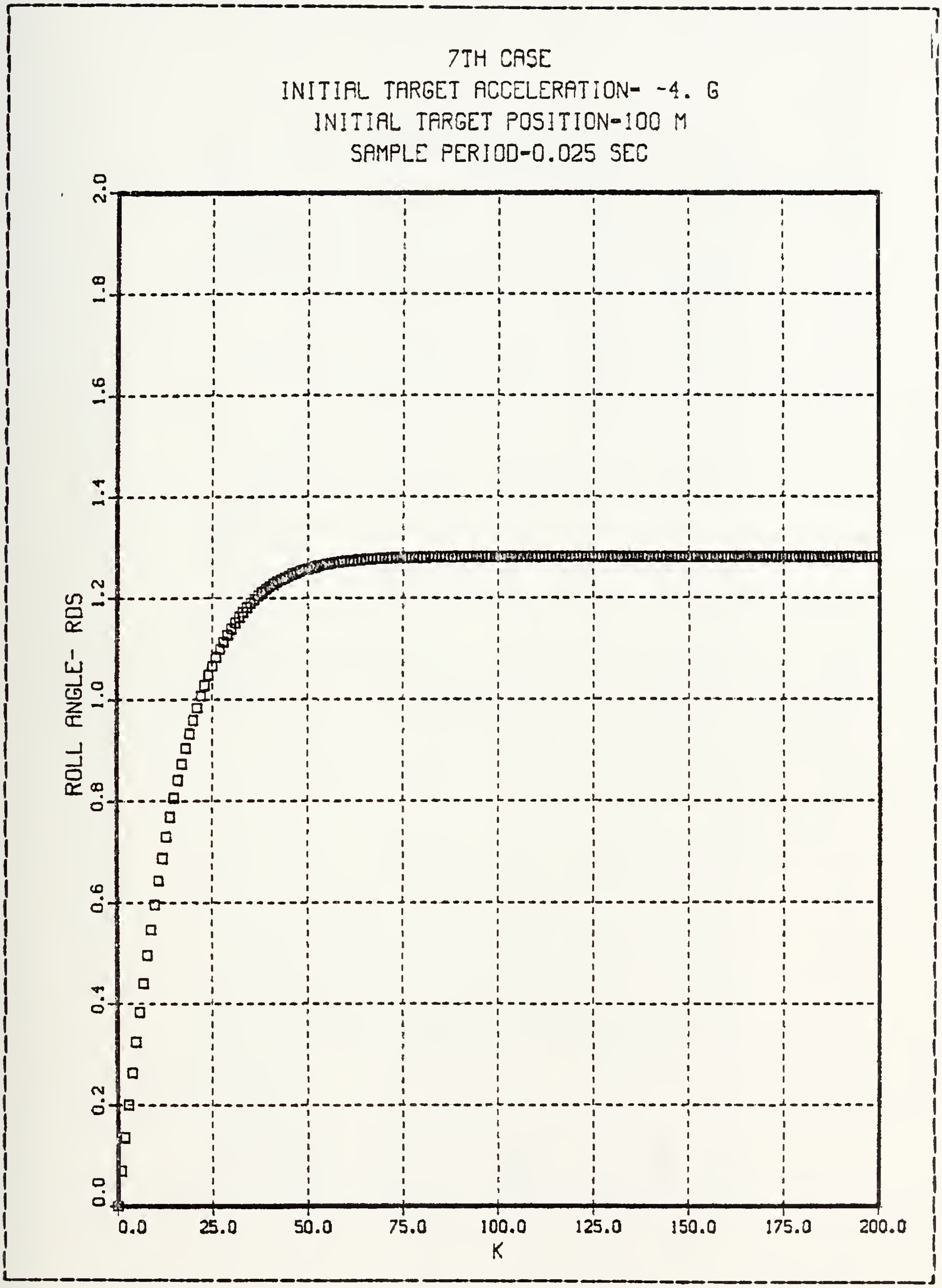

Figure 4.14 Roll Angle-Case 7.

TABLE III

Results for Differents Sample Periods

T (sec)	case	t	Ac (m/sec)	Pc (rad/sec)	miss distance Y direction (m)	miss distance Z direction (m)	Ø (rad)	CG-to-CG miss distance (m)
.1	6	0	26.40	2.70	0.0	100.	0.0	100.
		Tf	.67	0.0	2.23	1.77	1.23	2.85
.025	7	0	26.48	2.79	0.0	100.	0.0	100.
		Tf	.68	0.0	2.22	1.20	1.28	2.53
.05	3	0	26.47	2.76	0.0	100.	0.0	100.
		Tf	.33	0.0	-1.47	.046	1.27	1.54

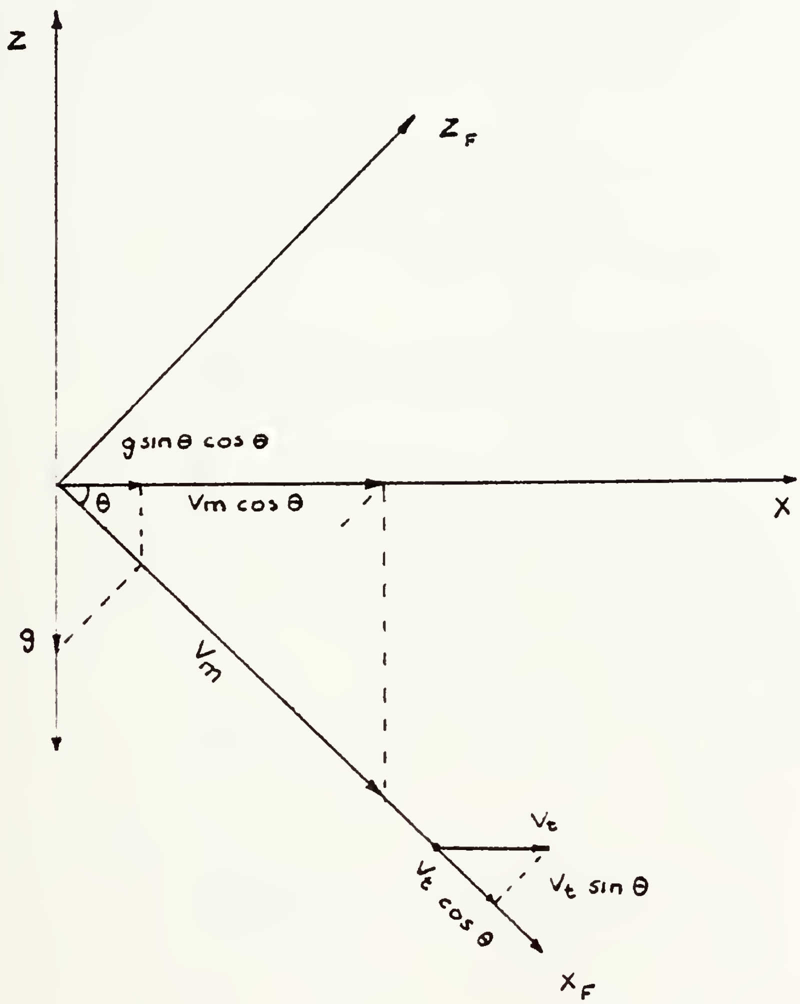

Figure 4.15 Effect of Initial Pitch Angle.

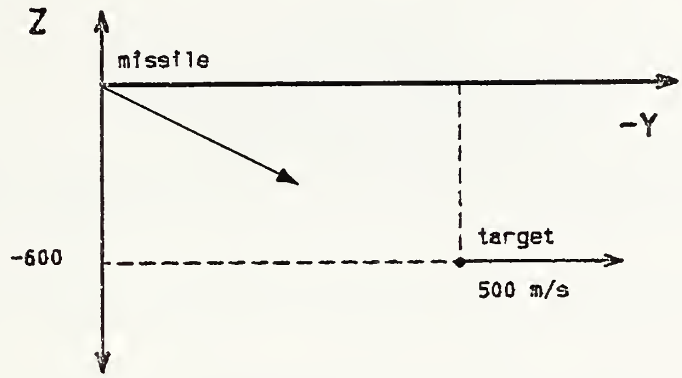

Figure 4.16a Case 8

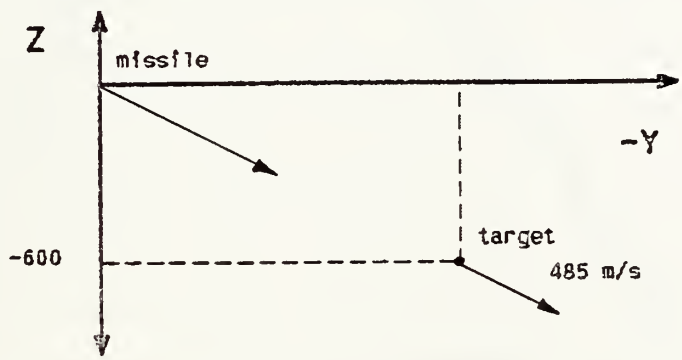

Figure 4.16b Case 9

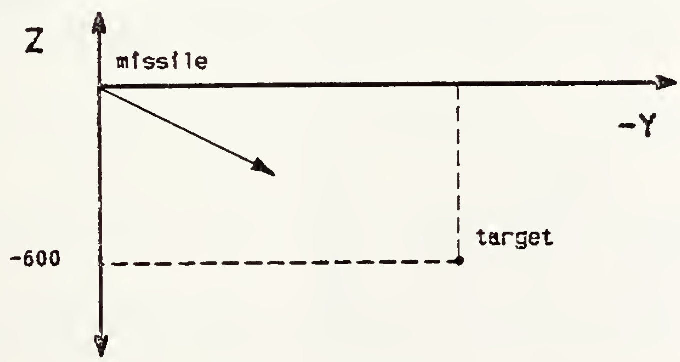

Figure 4.16c Case 10

Figure 4.16 Scenarios With Pitch Angle.

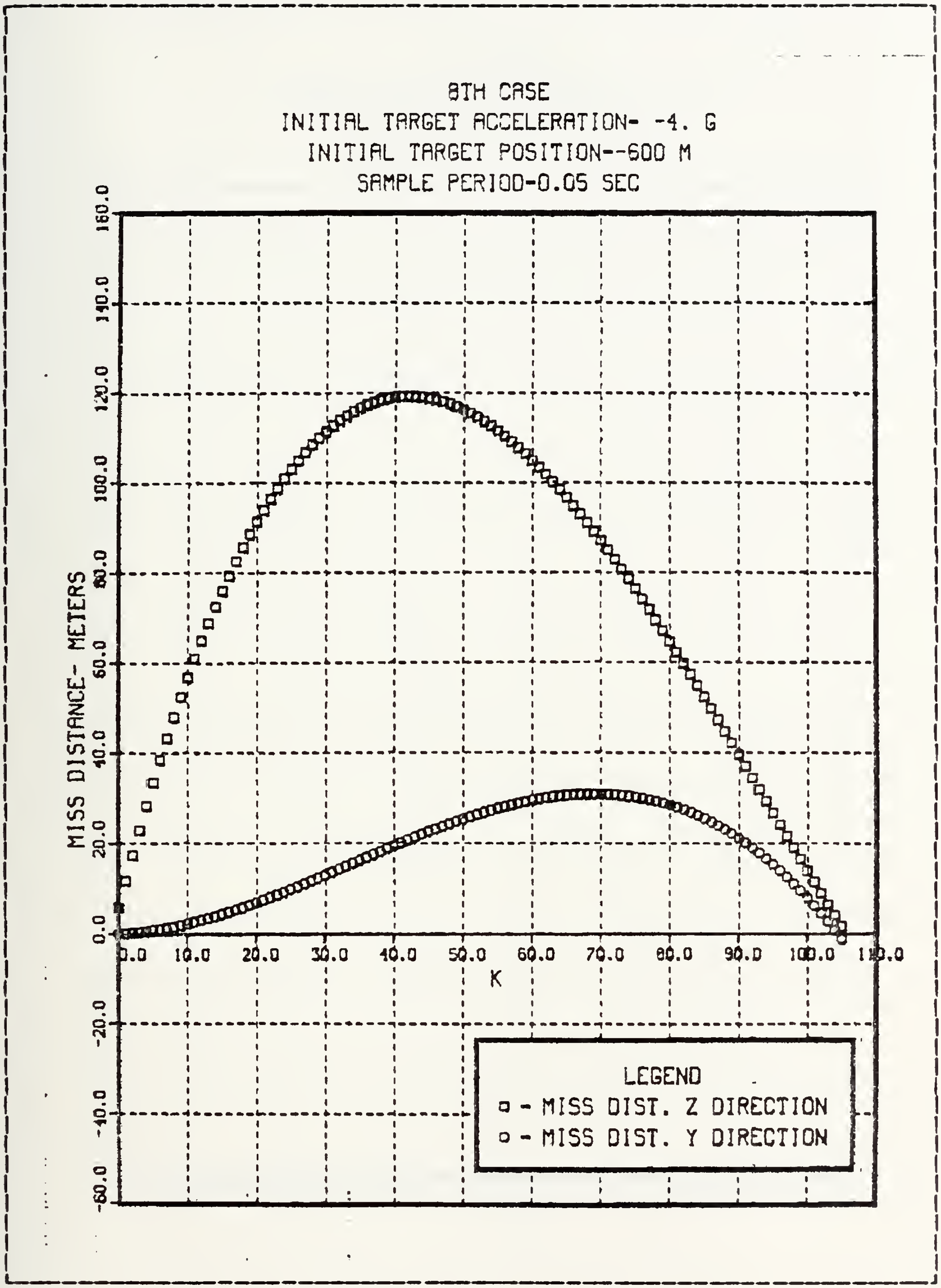

Figure 4.17 Commanded Acceleration-Case 8.

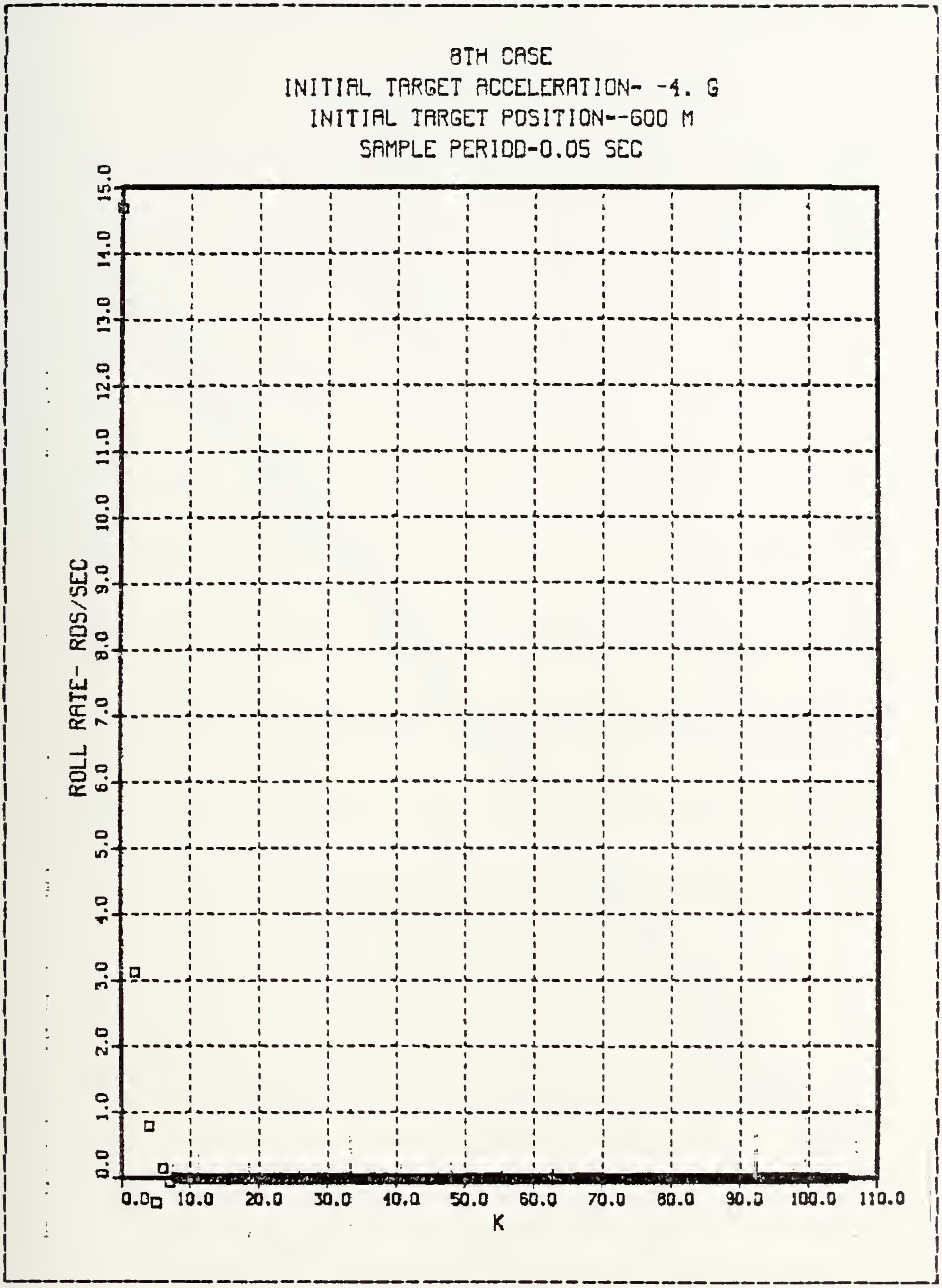

Figure 4.18 Commanded Roll Rate-Case 8.

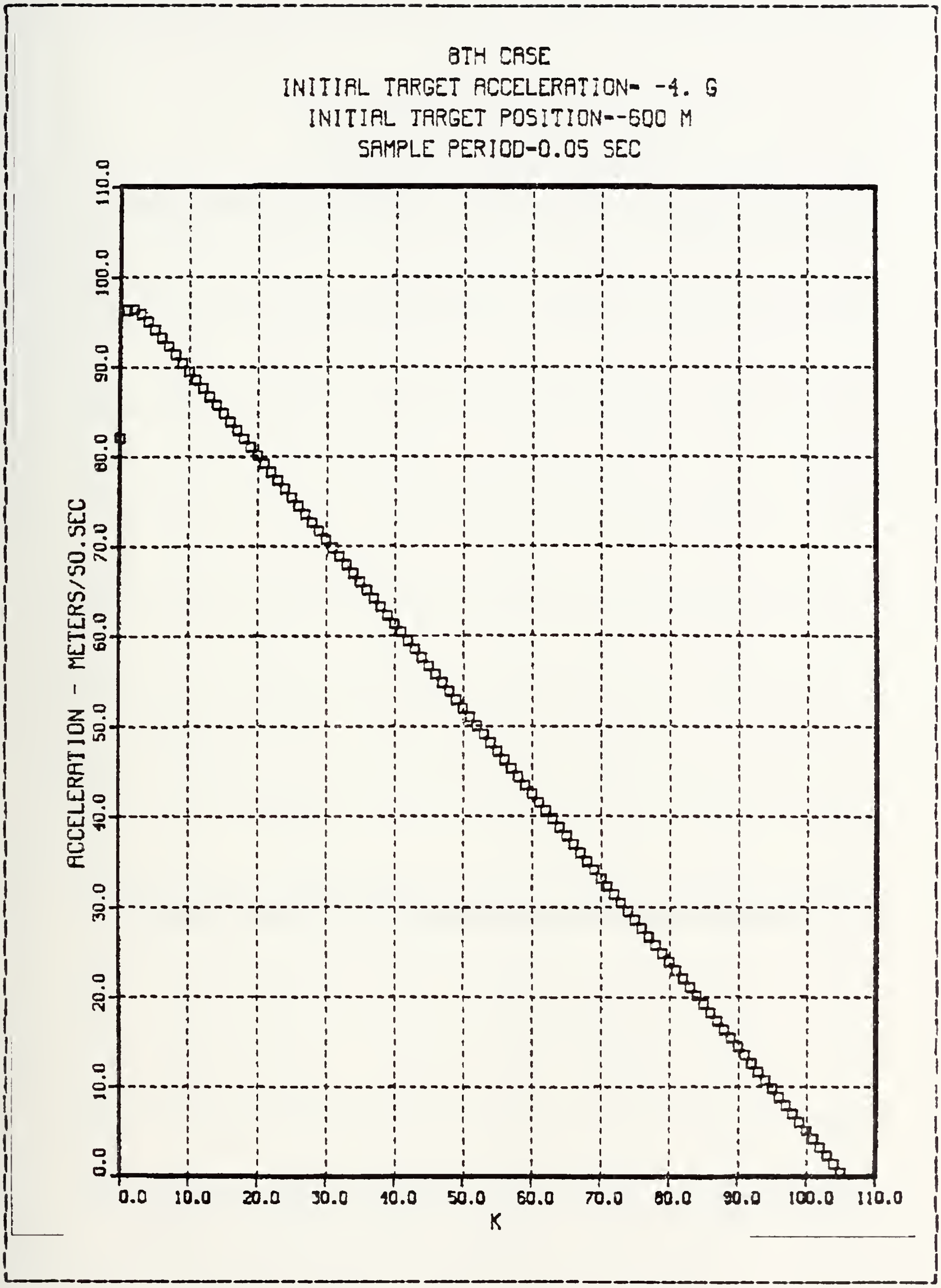

Figure 4.19 Miss Distance-Case 8.

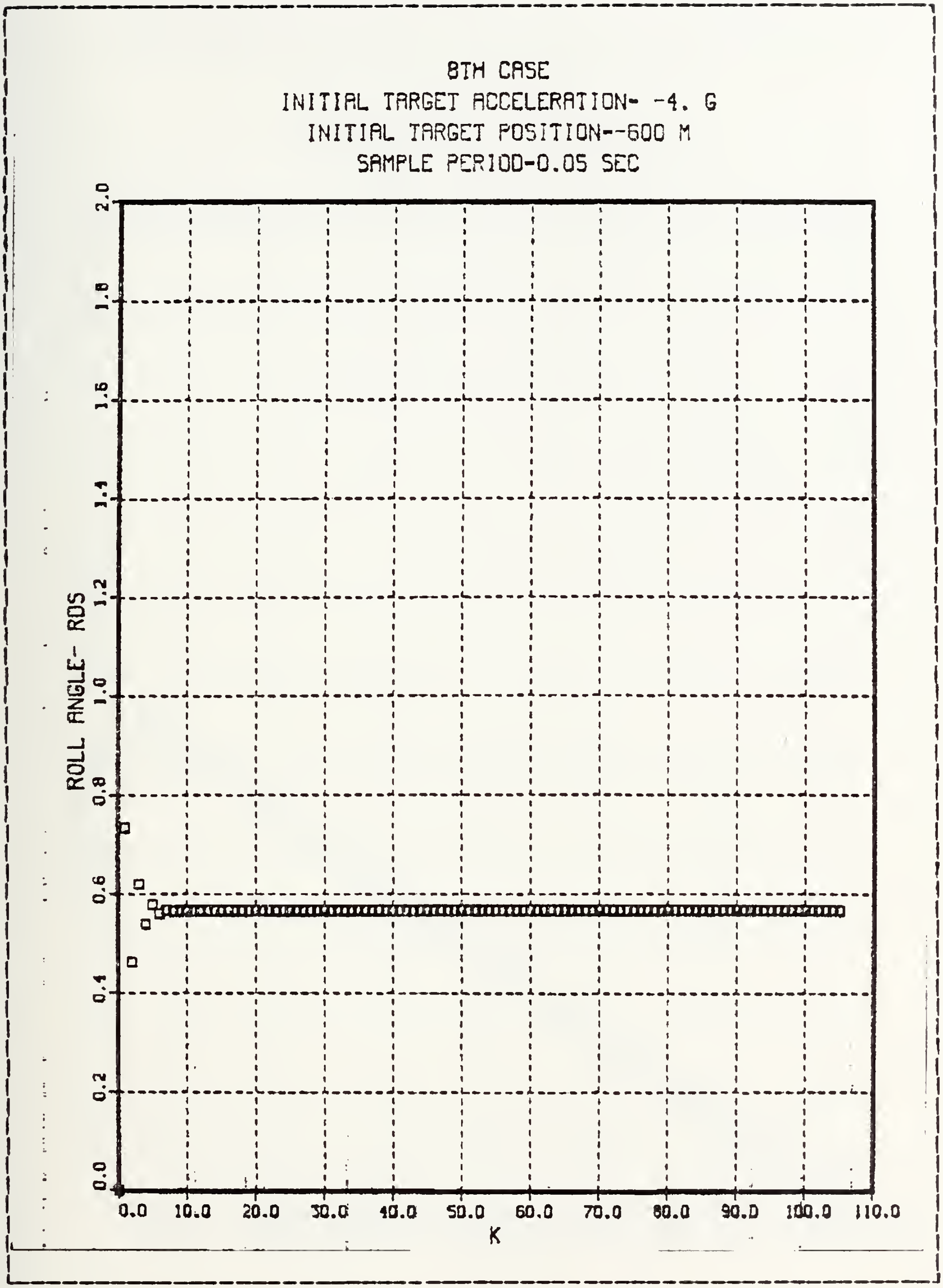

Figure 4.20 Roll Angle-Case 8.

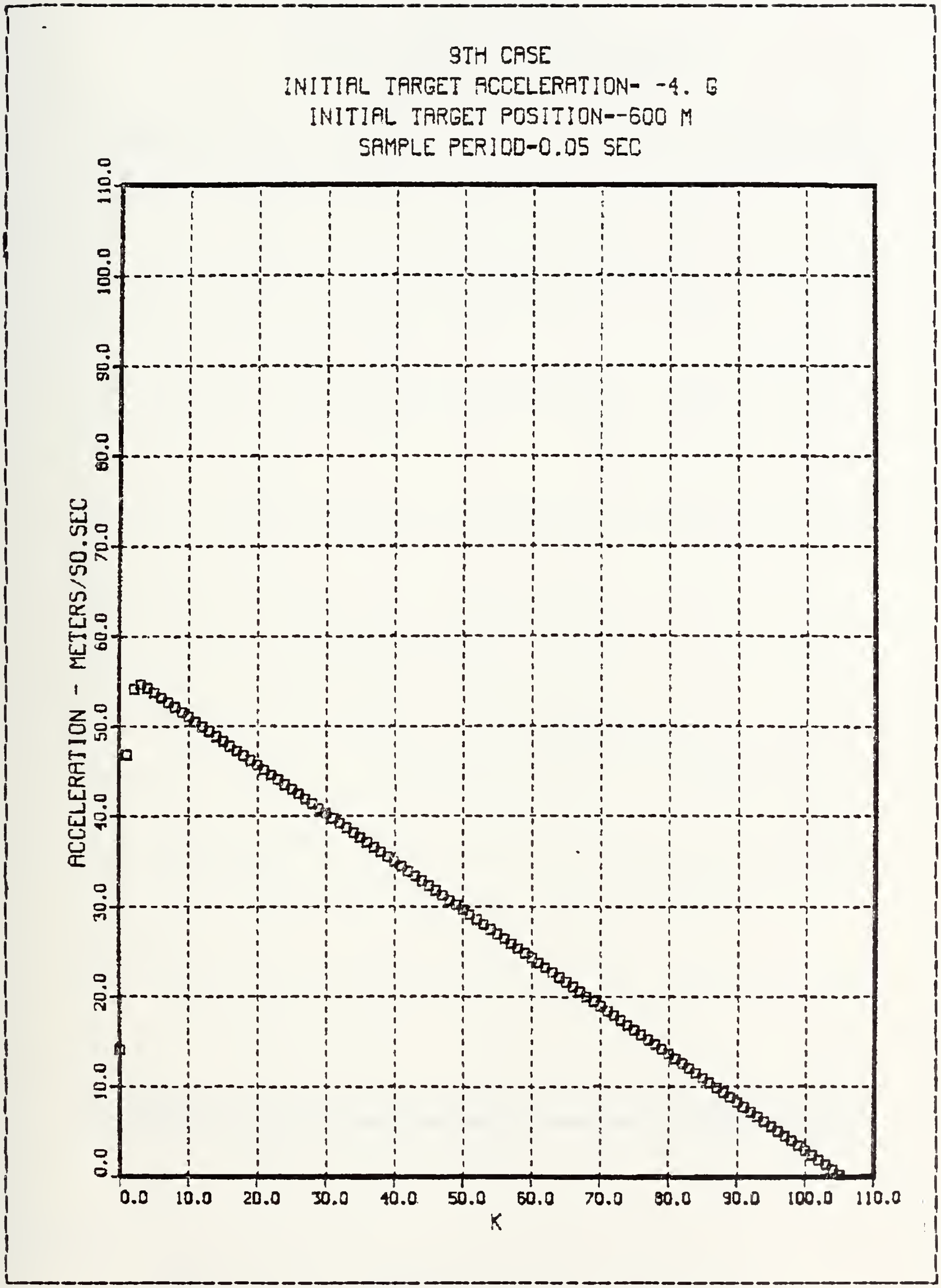

Figure 4.21 Commanded Acceleration-Case 9.

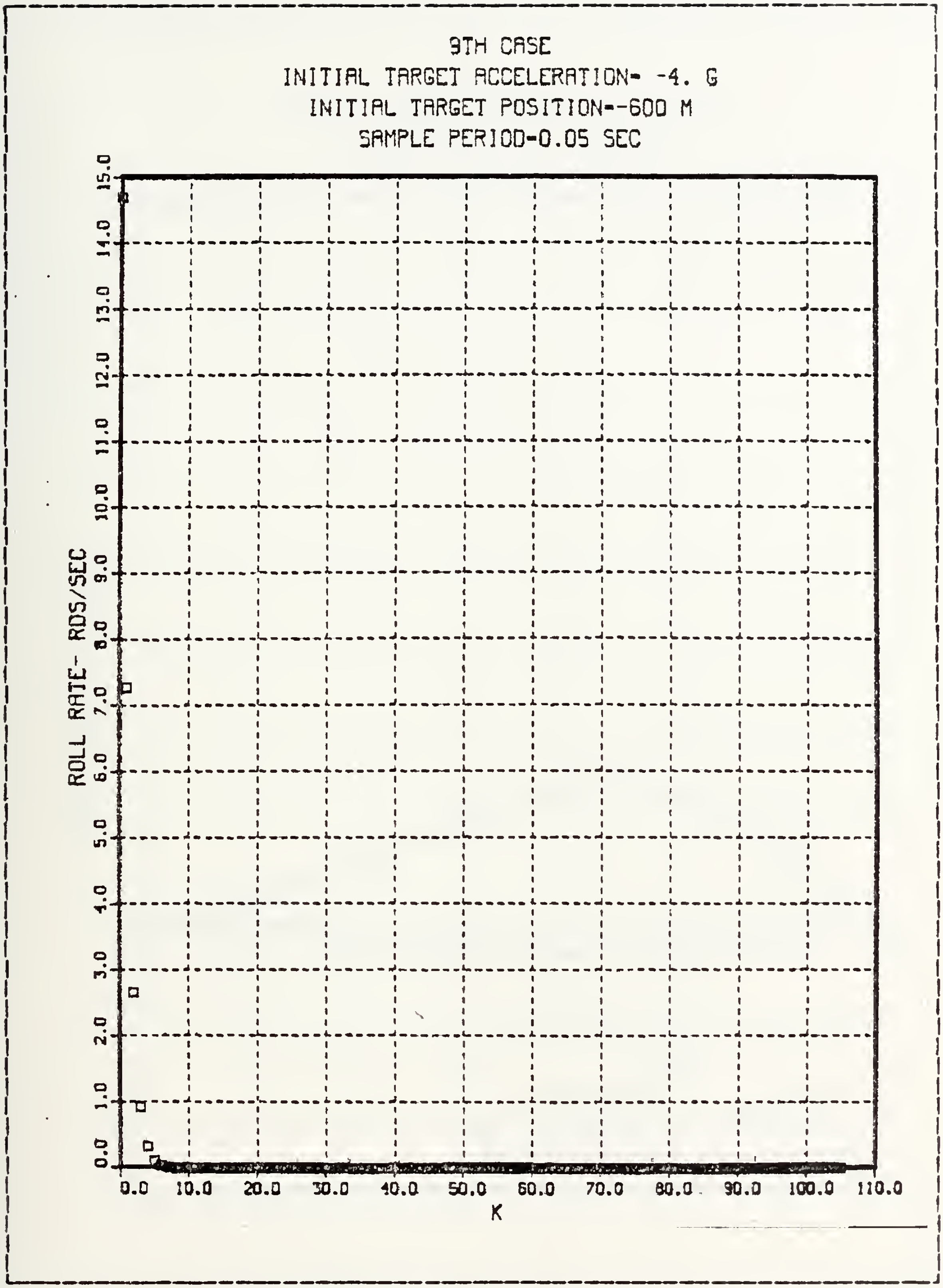

Figure 4.22 Commanded Roll Rate-Case 9.

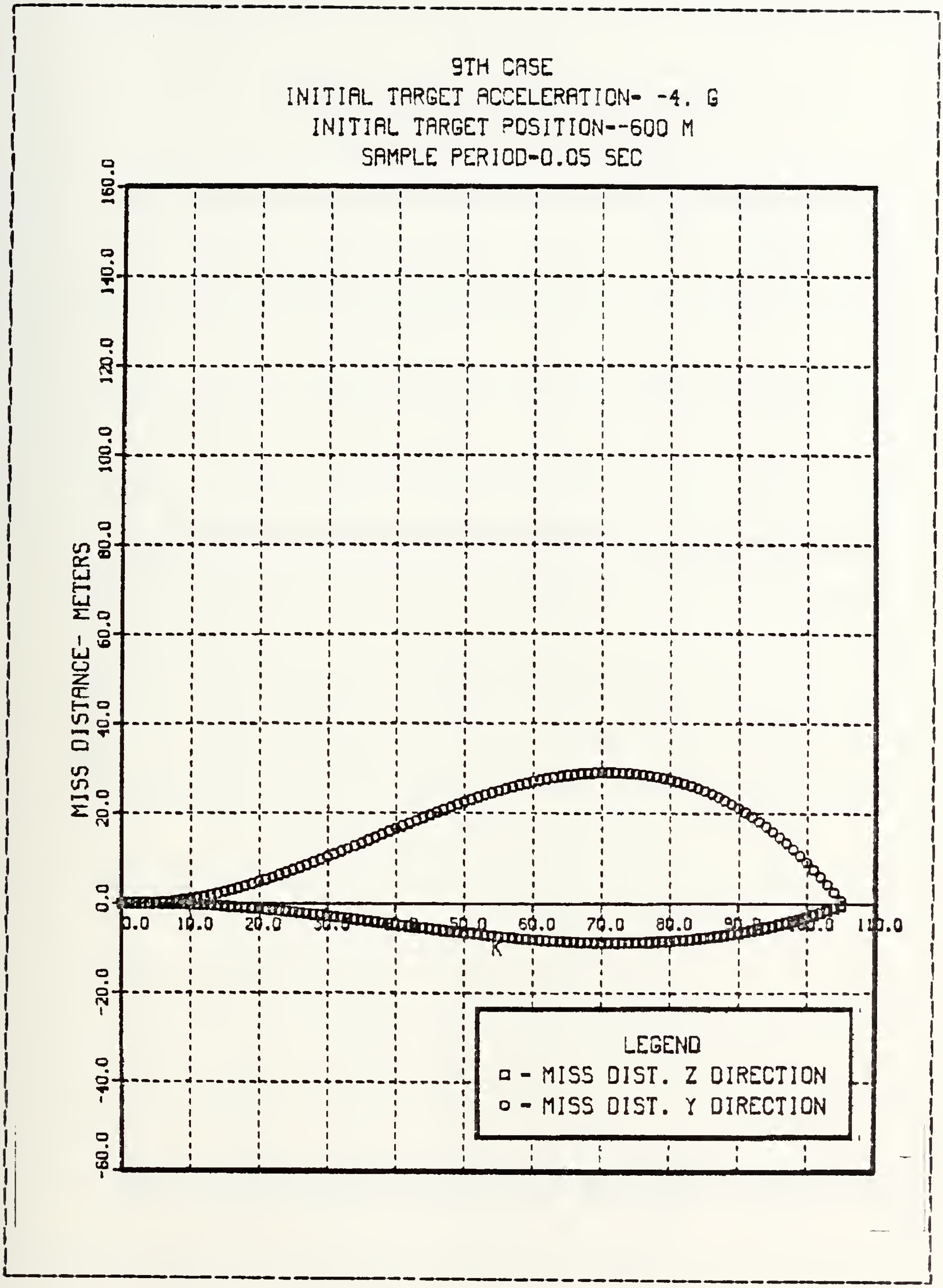

Figure 4.23 Miss Distance-Case 9.

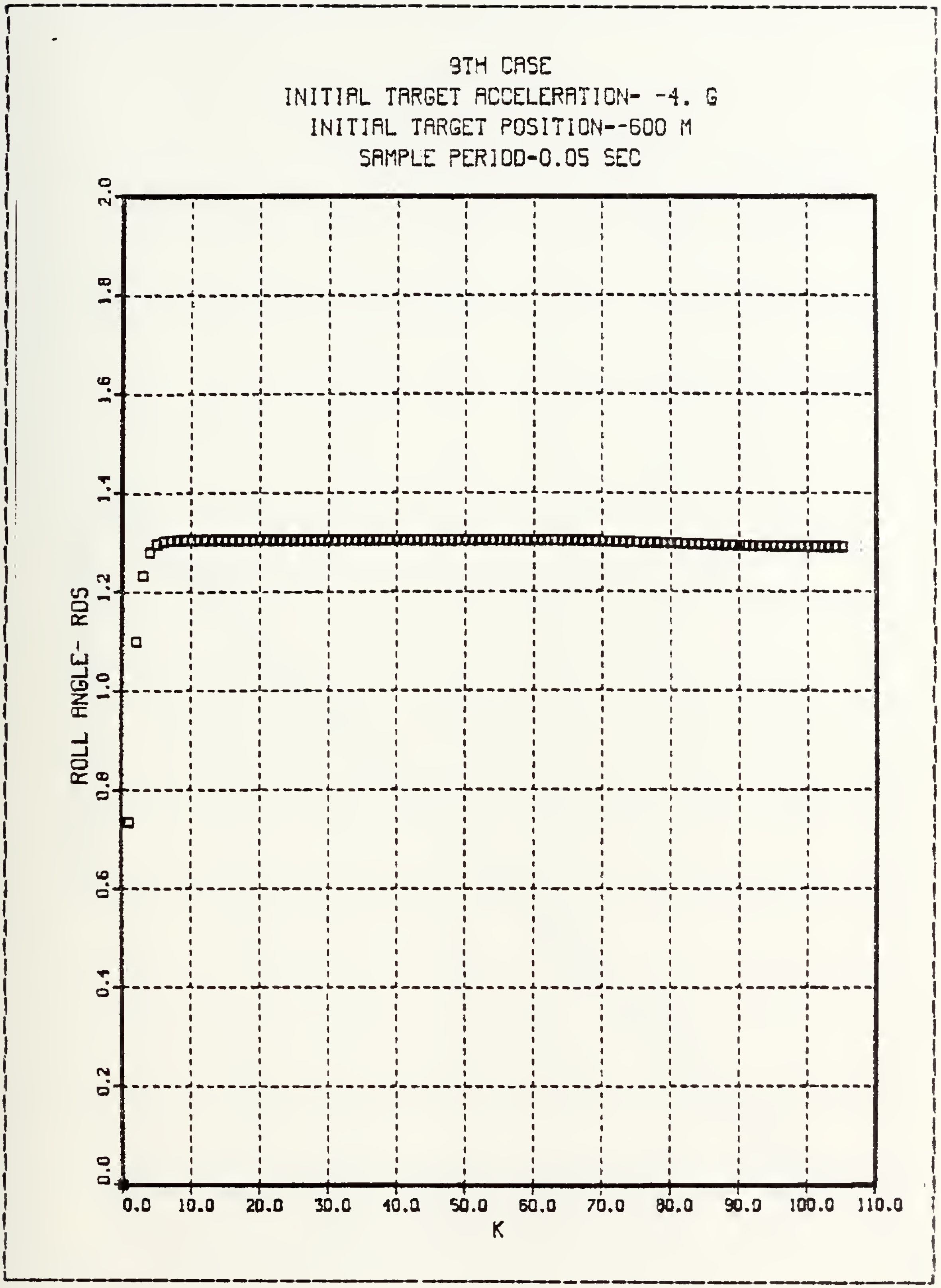

Figure 4.24   Roll Angle-Case 9.

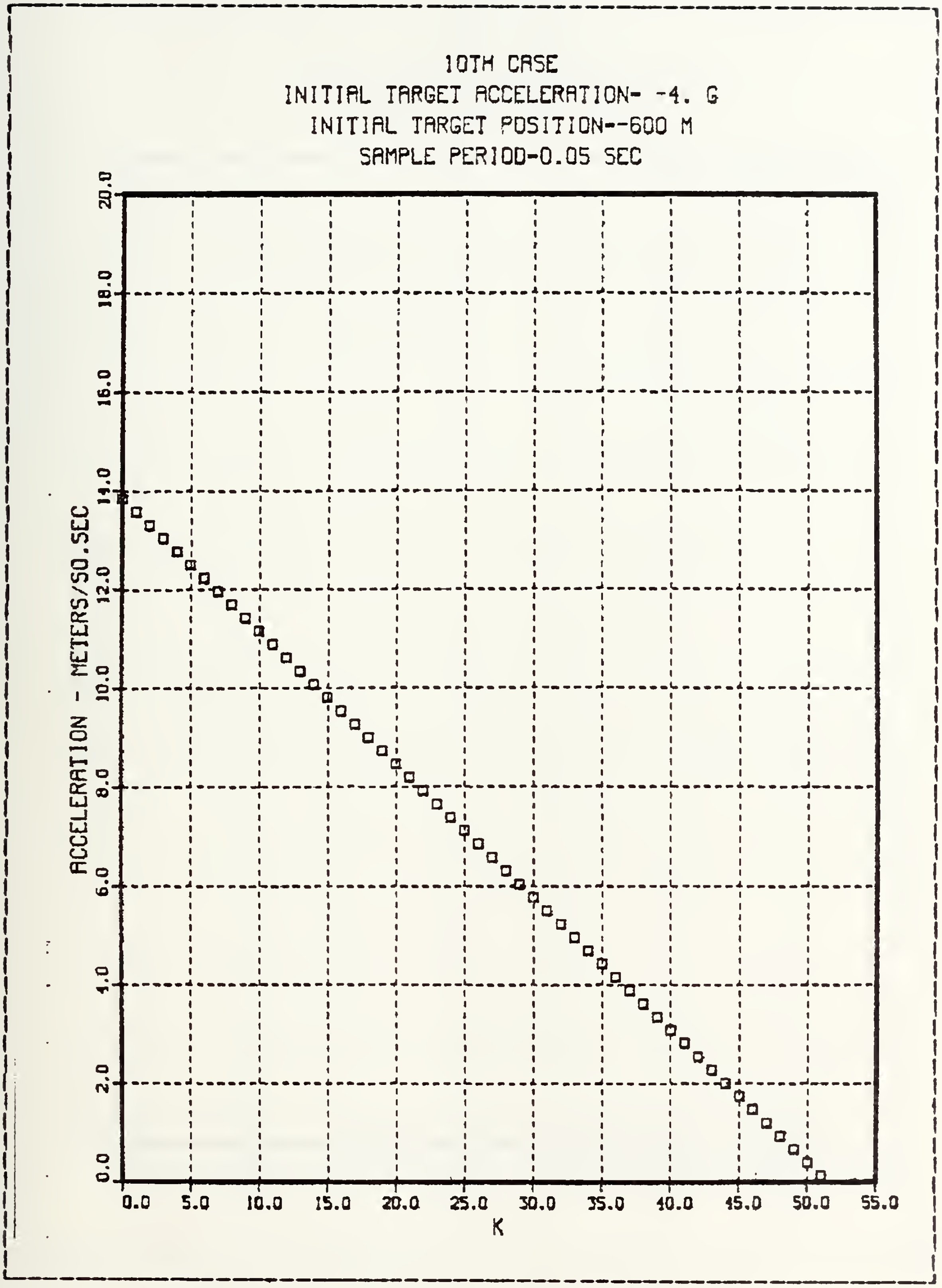

Figure 4.25 Commanded Acceleration-Case 10.

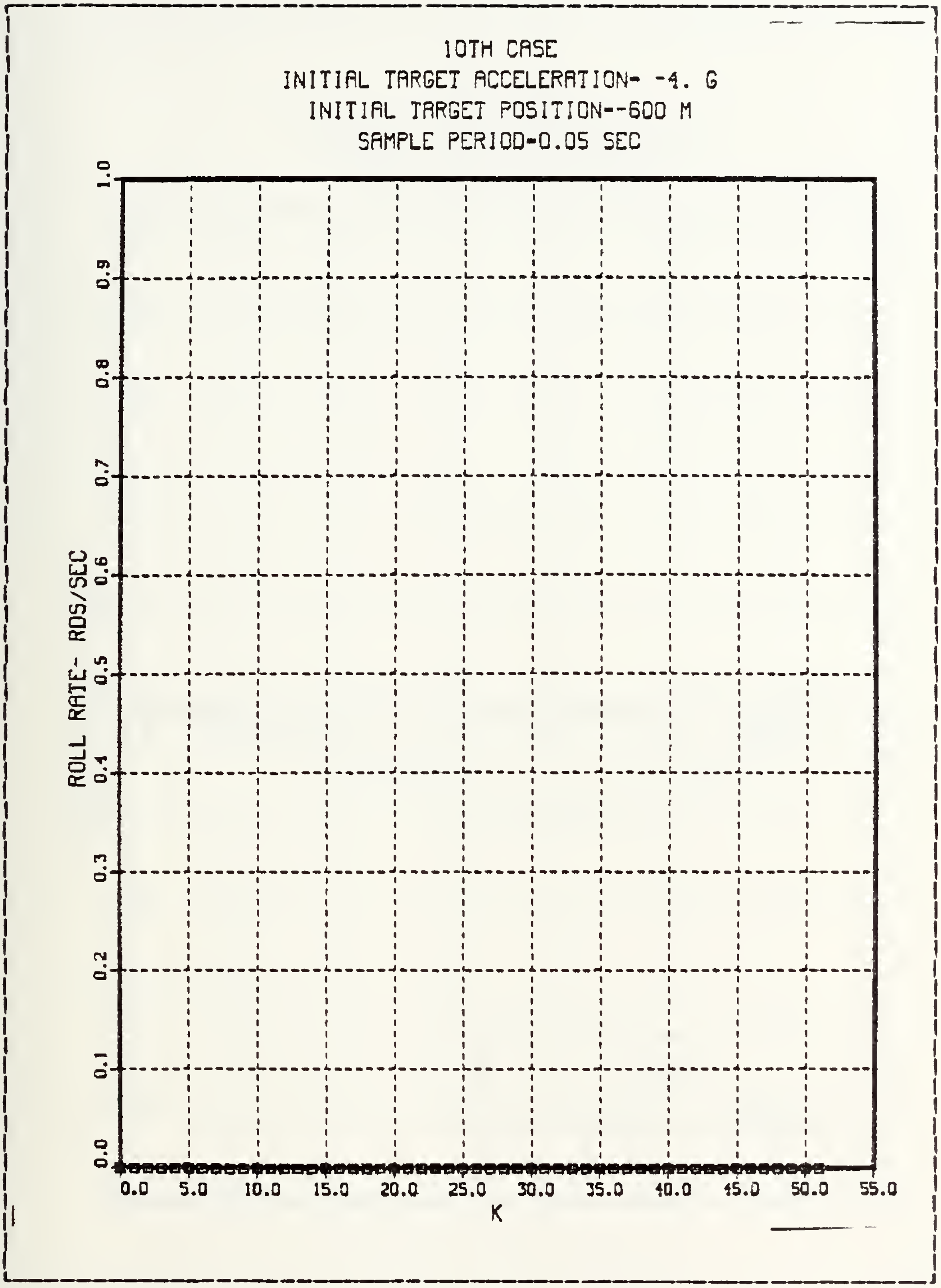

Figure 4.26 Commanded Roll Rate-Case 10.

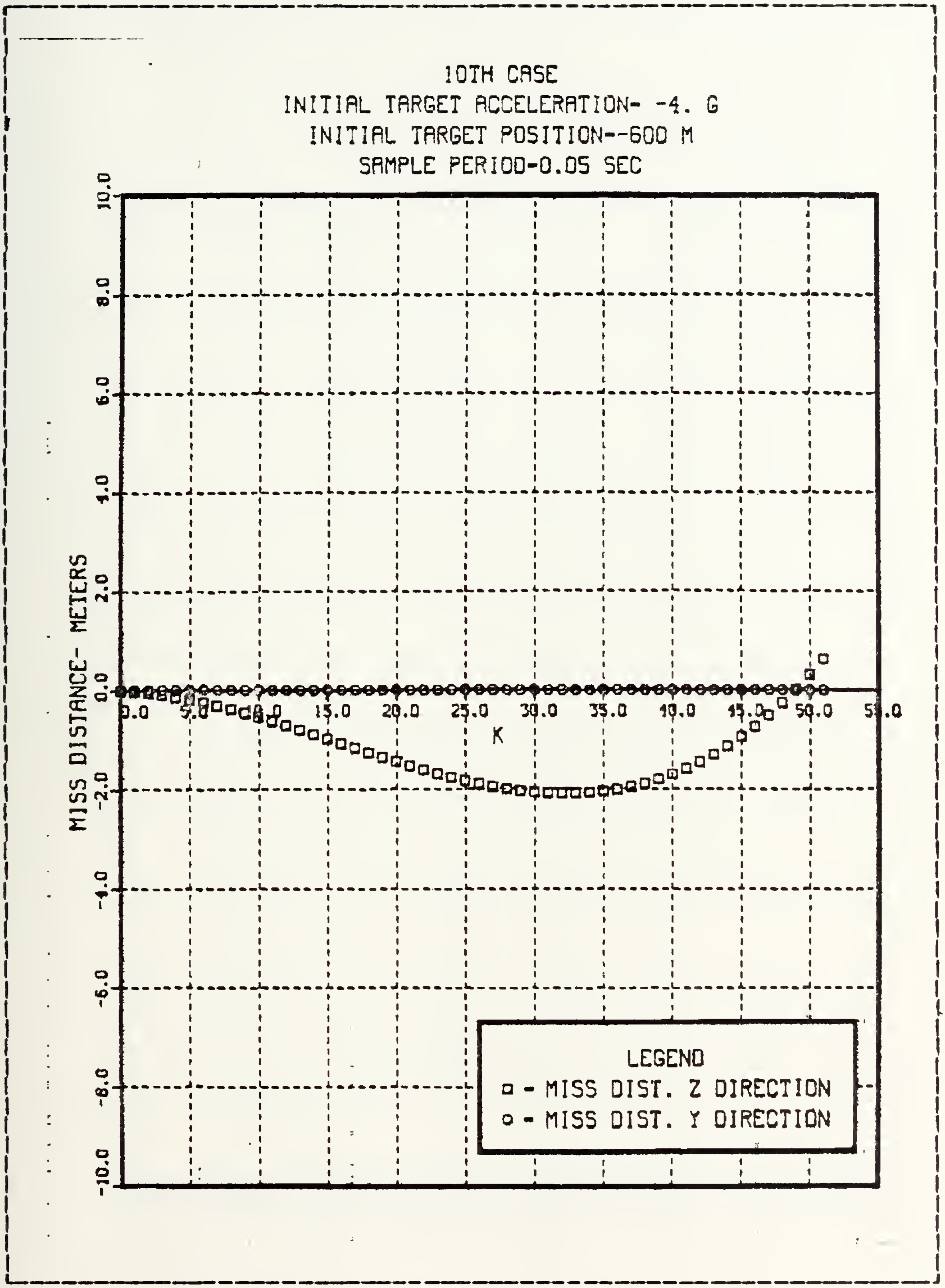

Figure 4.27 Miss Distance-Case 10.

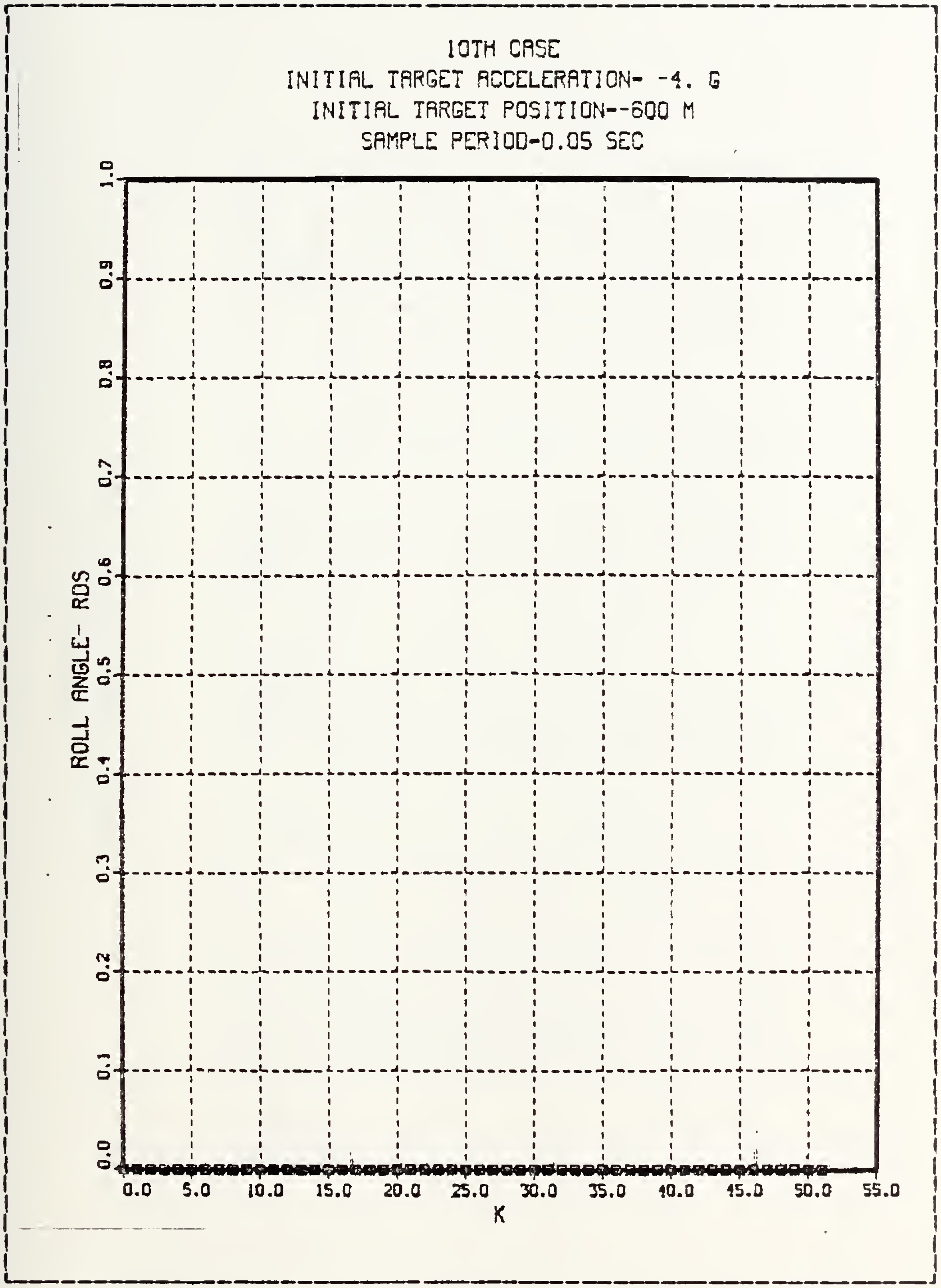

Figure 4.28 Roll Angle-Case 10.

TABLE IV

Results Using Pitch Angle

case	t	Ac (m/sec)	Pc (rad/sec)	miss distance Y direction (m)	miss distance Z direction (m)	(rad)	CG-to-CG miss distance (m)
8	0	82.13	14.71	0.0	-600	0.0	600
	Tf	.466	0.0	-1.17	.188	.562	2.17
9	0	14.16	14.71	0.0	-600	0.0	600
	Tf	.268	0.0	-1.12	.176	1.29	1.249
10	0	13.83	0.0	0.0	-600	0.0	600
	Tf	.135	0.0	0.0	.307	0.0	.307

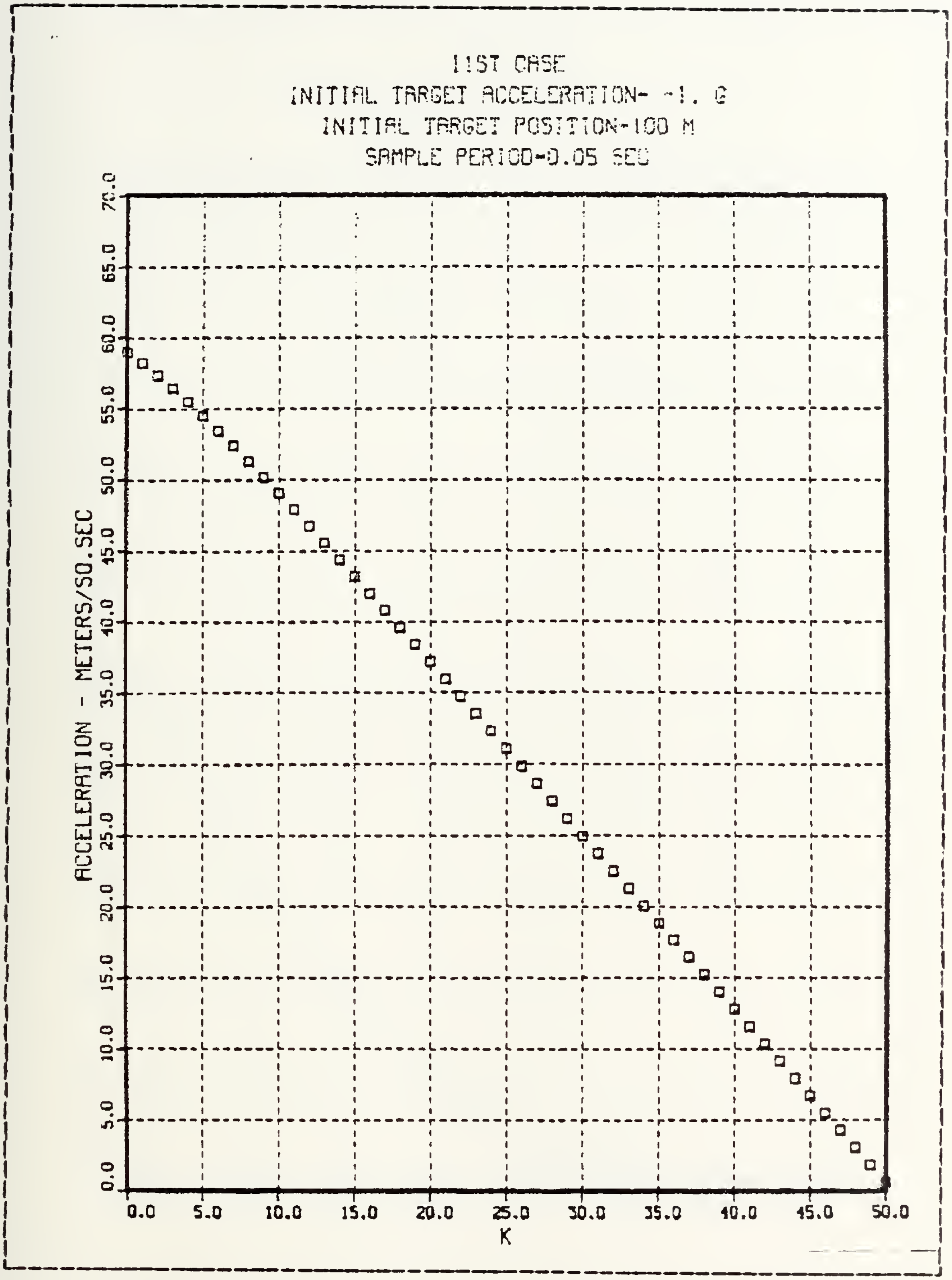

Figure 4.29 Commanded Acceleration-Case 11.

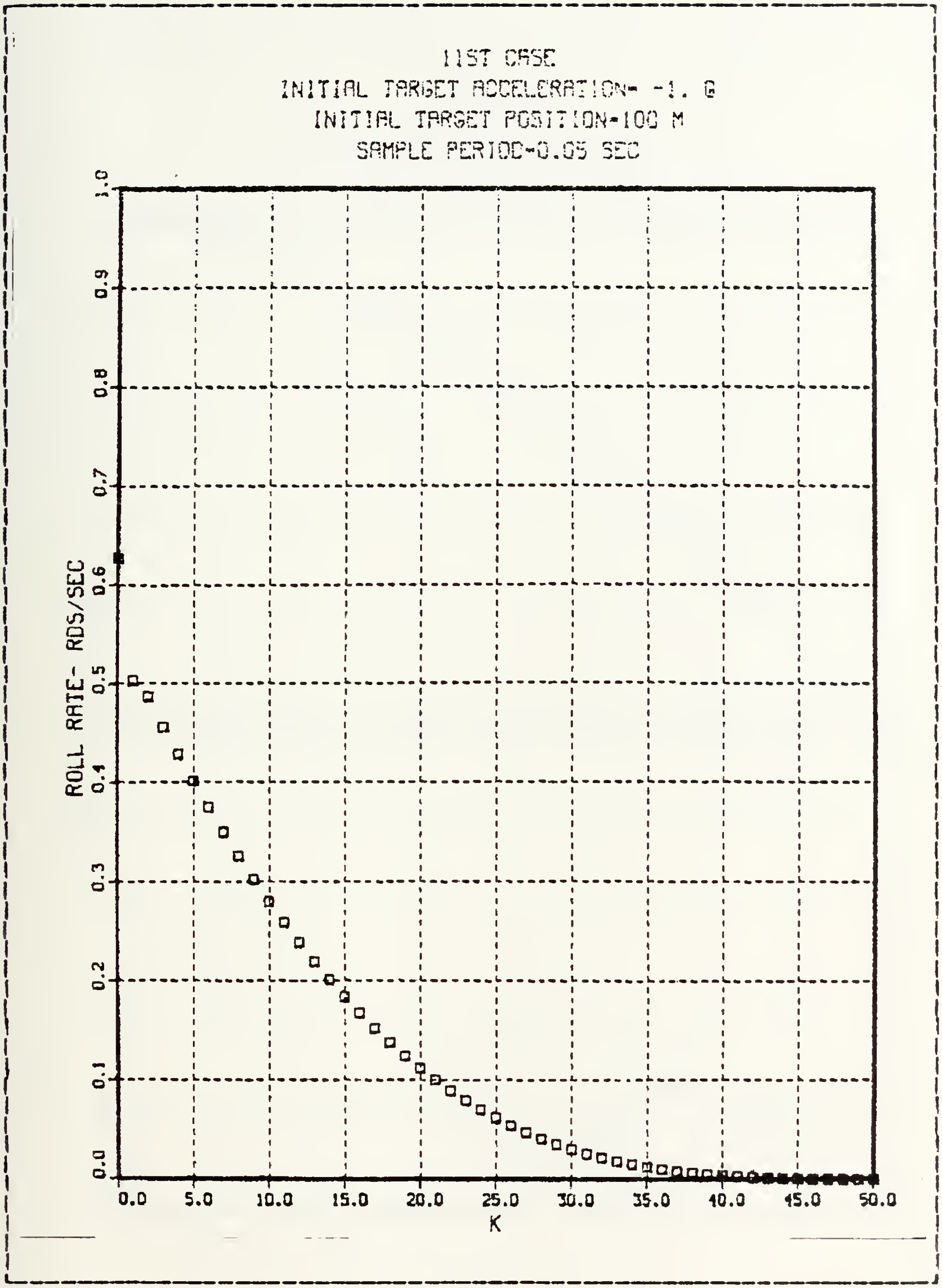

Figure 4.30 Commanded Roll Rate-Case 11.

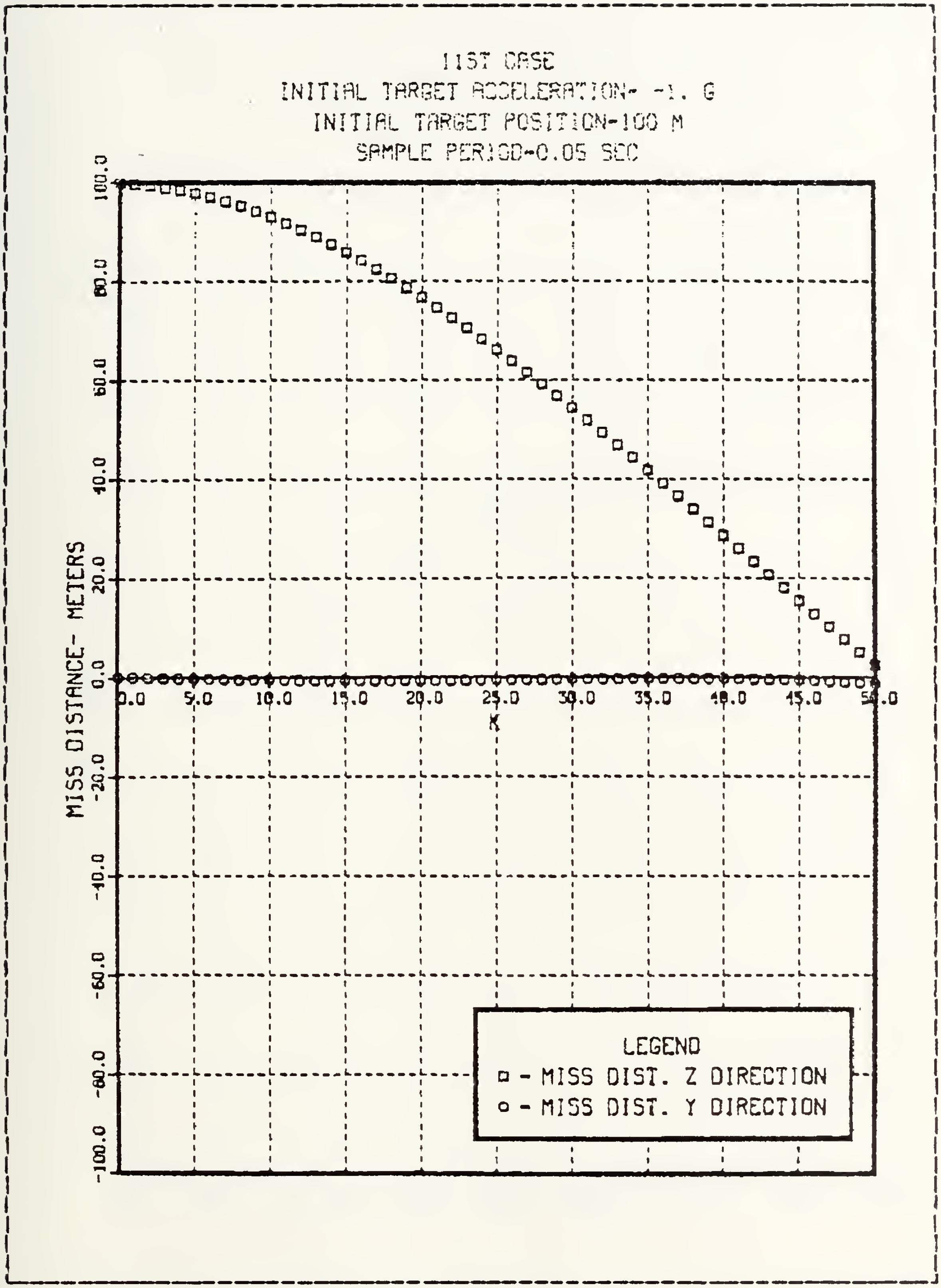

Figure 4.31 Miss Distance-Case 11.

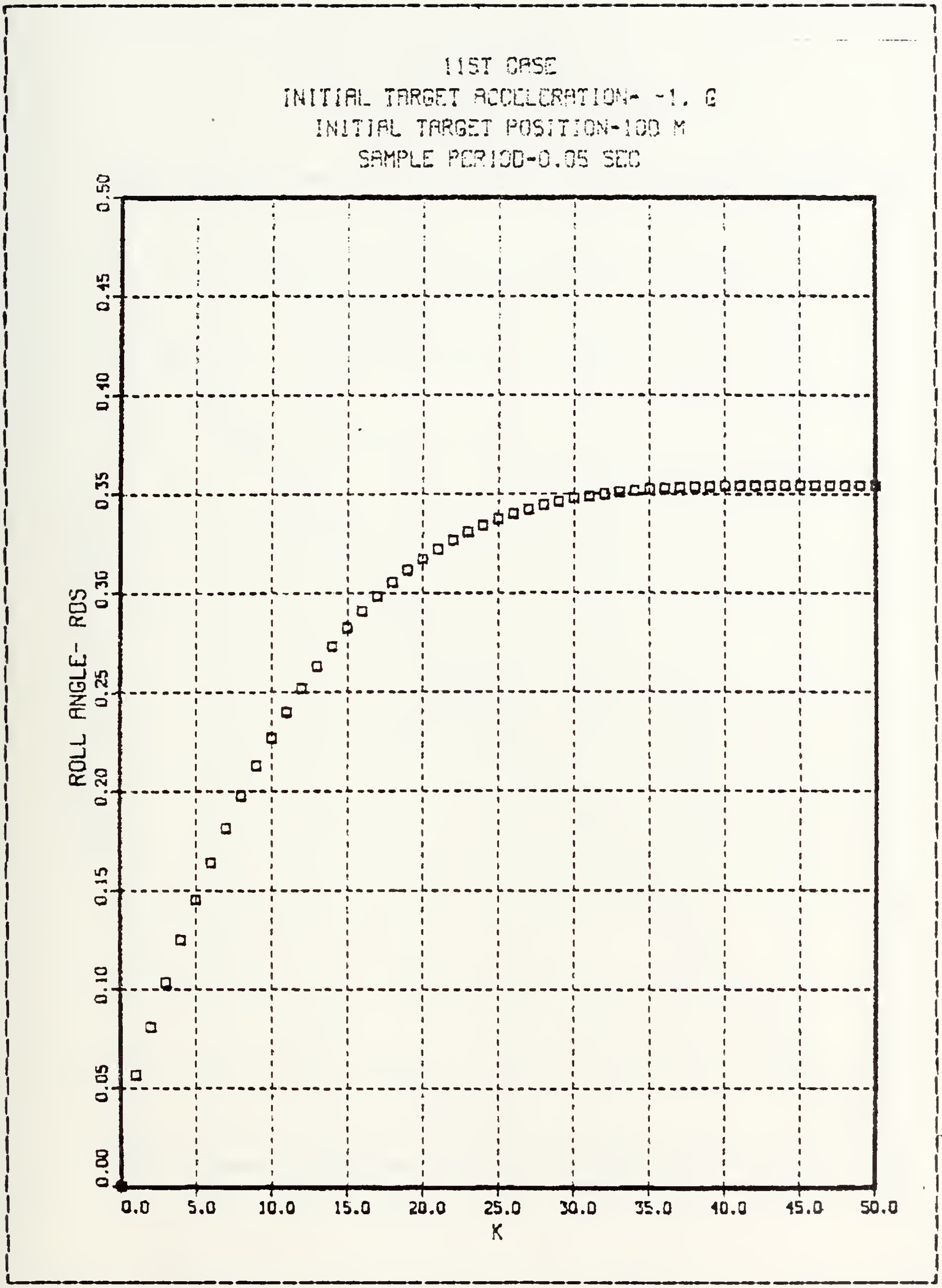

Figure 4.32 Roll Angle-Case 11.

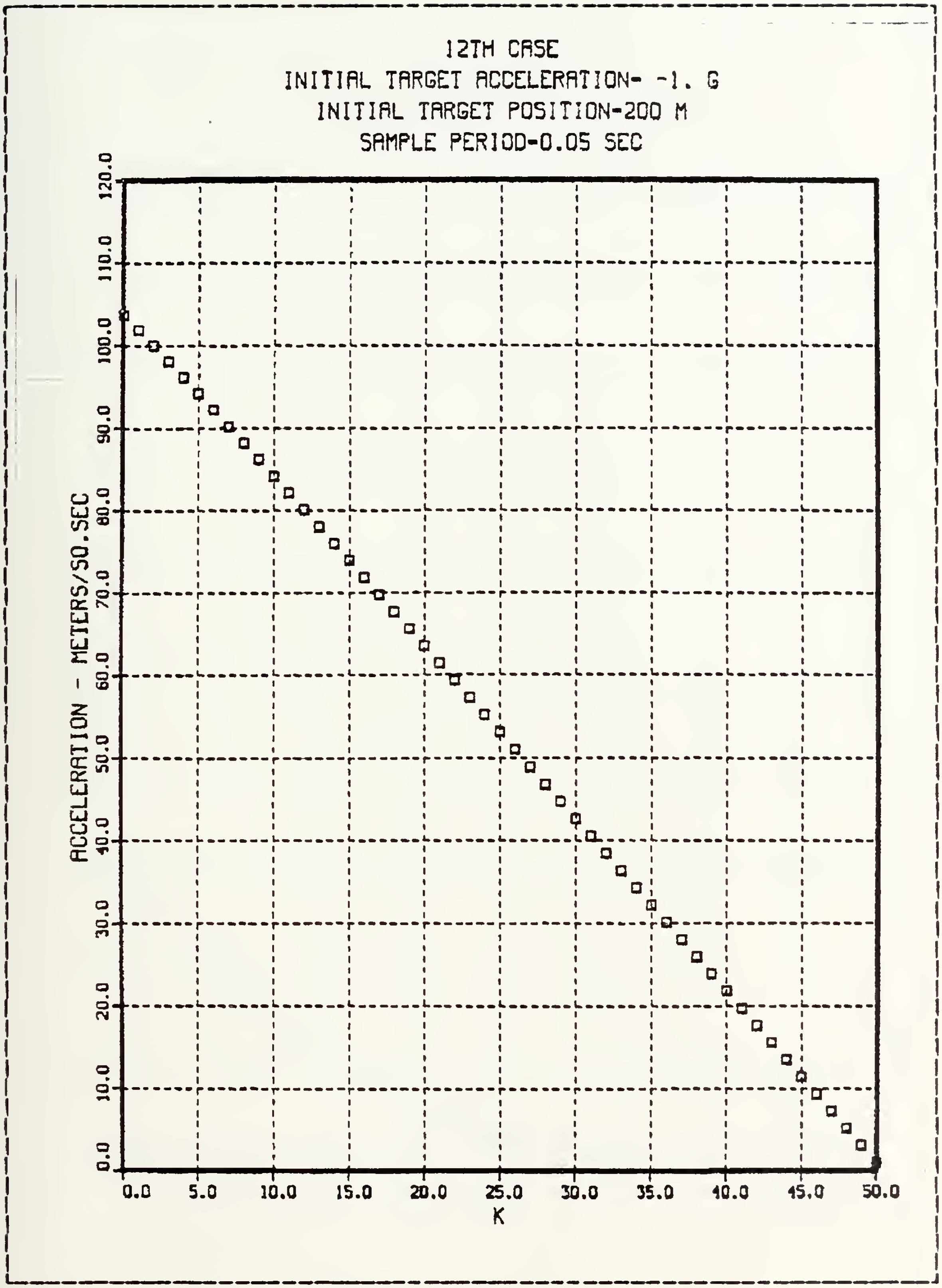

Figure 4.33 Commanded Acceleration-Case 12.

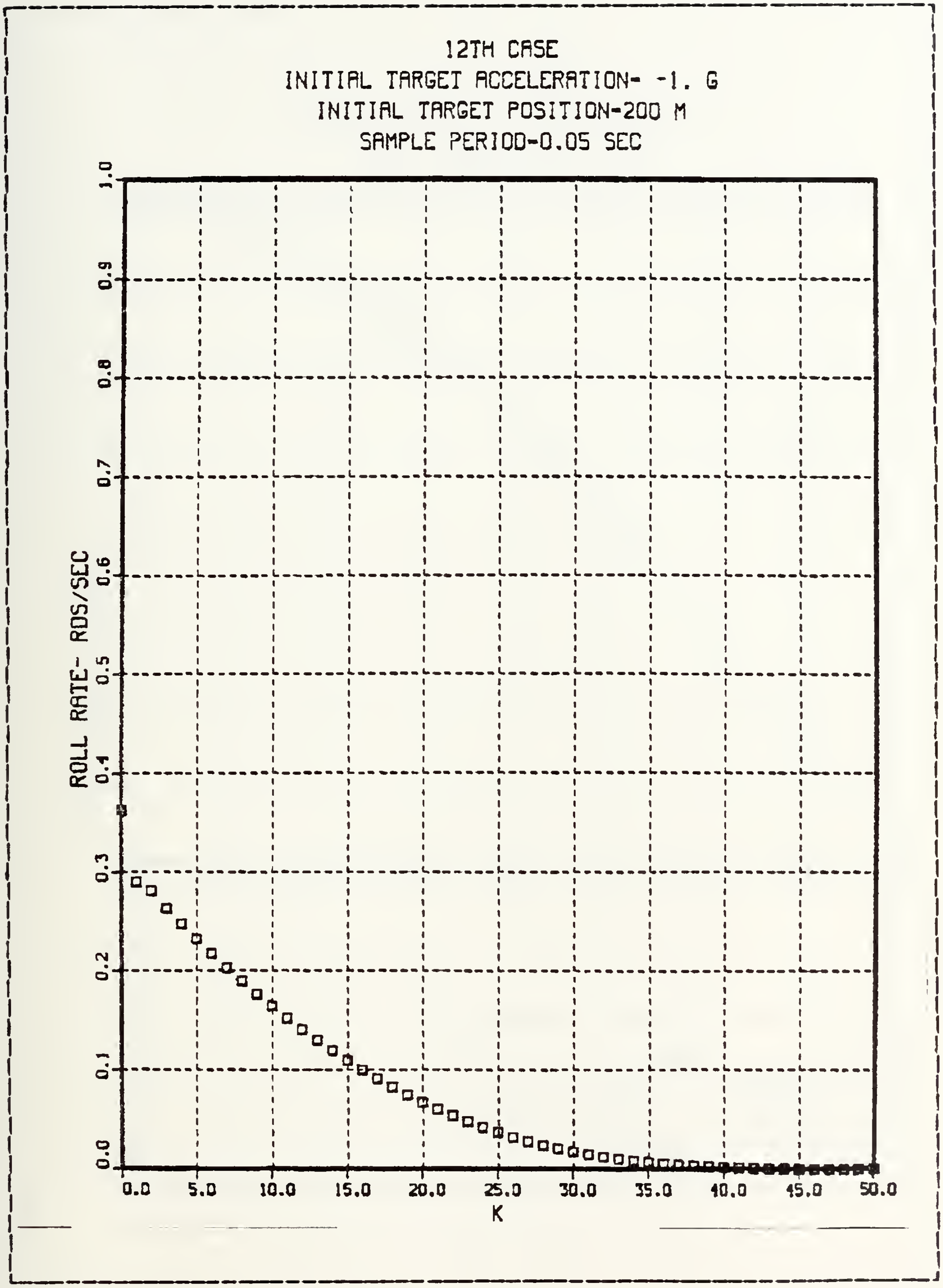

Figure 4.34 Commanded Roll Rate-Case 12.

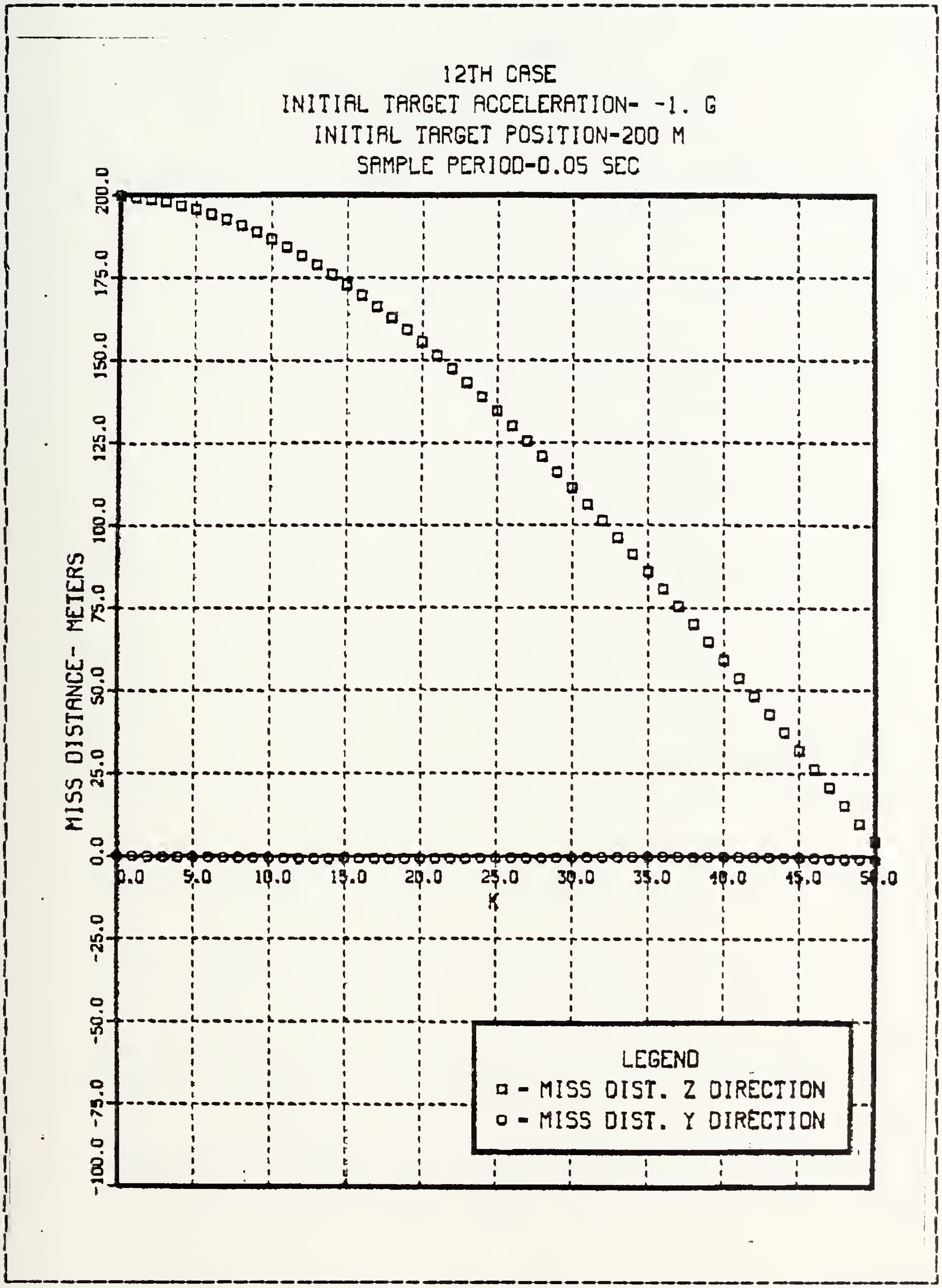

Figure 4.35 Miss Distance-Case 12.

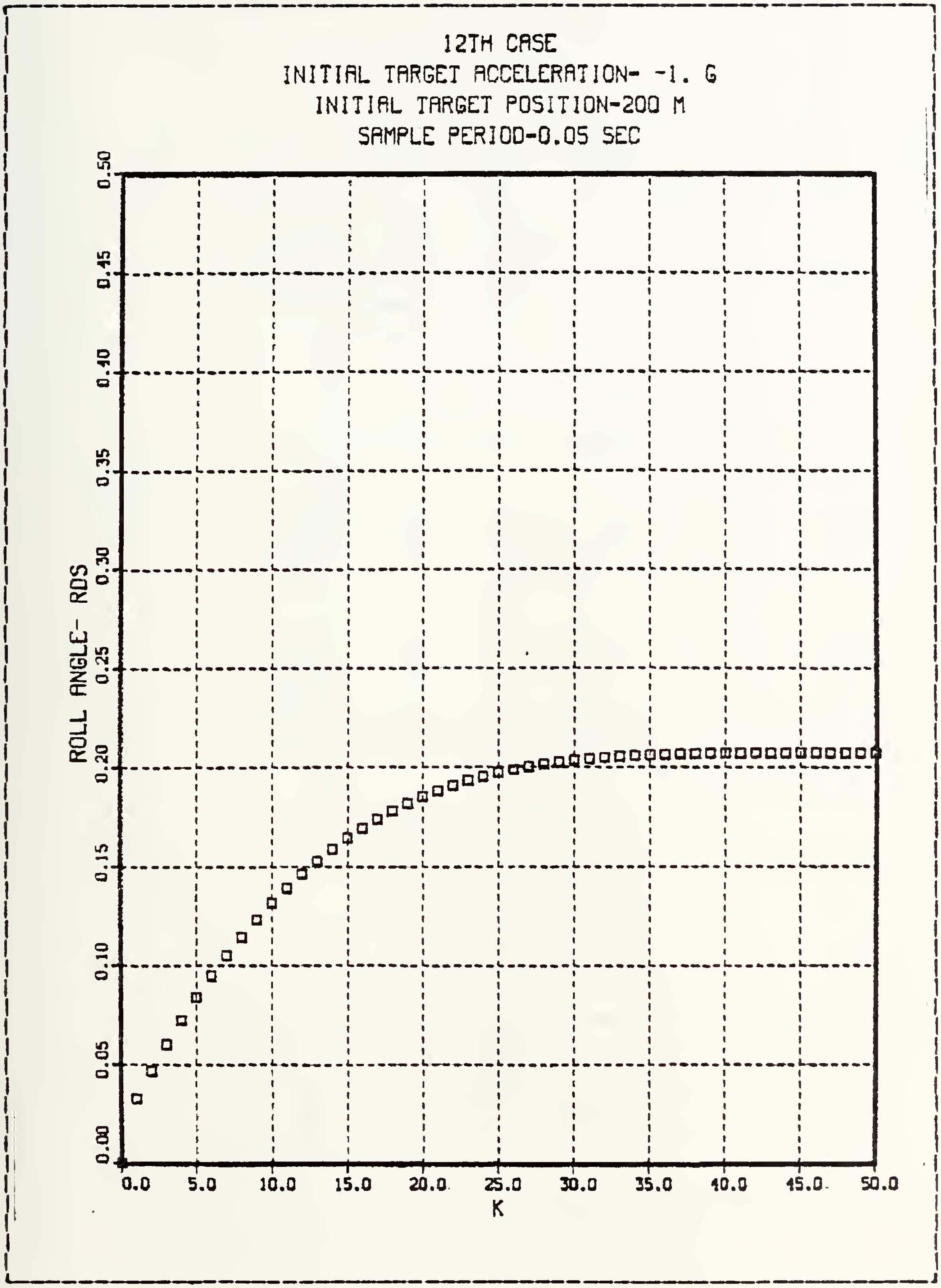

Figure 4.36 Roll Angle-Case 12.

TABLE V

Effect of Time to Intercept

case	t	Ac (m/sec)	Pc (rad/sec)	miss distance Y direction (m)	miss distance Z direction (m)	(rad)	CG-to-CG miss distance (m)
11	0	59.01	.627	0.0	100.	0.0	100.
	Γ1	.608	0.0	-1.31	2.53	.354	2.85
12	0	103.78	.363	0.0	200.	0.0	200.
	Γ1	1.04	0.0	-1.01	4.69	.207	4.82

V. FINAL CONCLUSIONS AND COMMENTS

The scope of the present work was the development of an optimal digital control to be applied on a bank-to turn missile.

A two dimensional model, as sugested in reference 1, was adopted. After the digitalization of the continuous model it was necessary to solve a modified Ricatti equation since in the state equation there was a third term representing the gravity's effect. The approach that has been adopted is new, and although good results were obtained for the scenarios considered in this work, is necessary that the algorithm be further tested and evaluated in similar problems due to its novelty.

The optimal was solved with an initial restriction to small angles. This condition was later relaxed so that large roll angles could be analyzed.

It is dificult to compare the present work with previous results since Stallard has indicated a mistake in his original paper, and further works in this area was not found.

However some comparison with Stallard work is possible. The commanded acceleration of the missile are such as to correct the ZEM at each point, this agrees with that reference. There is a proportional relationship between the commanded roll rate and the commanded acceleration , and the commanded roll rate is proportional to the defined ϕ_{ideal} at each point, which again agrees with reference 1.

The algorithm developed in this work requires extensive computation at each step, and is clear that some software optimization will be needed. The motivation for considering the constant steady-state gain due to gravity was to decrease this computational burden. This approach however resulted in unacceptable miss distance.

Another point of investigation that could reflect on the period available to the computer to perform its calculations was a change in the sample rate. Two different sample rates were investigated, both lead to larger errors than the nominal period of .05 seconds. A detailed study on this issue is 'left as sugestion for future works, since some optimal value of the sampling rate is clearly indicated.

It is important to keep in mind that the model adopted is two dimentinal, while the actual problem is three dimensional, thus some brief studies were conducted in order to check the region of validity of the 2-D.

In the analysis of the pitch angle, one can see that is necessary to have small variations in pitch in order to approximate it as a constant. However, at the moment that this angle is different from zero, as explained in chapter 4, it is possible to have in the flight path reference frame a target manouver in the Z direction that will lead to large acceleration commands, leading the missile to large miss distances, when considering a movable target. When the present system was tested tested against fixed targets, the results were quite good, this suggests the application of the model in air-to-surface missiles.

Further investigation were made on the effect of time-to-go. As expected, decreased time to go, results in increased miss distance. A detailed analysis of more complex scenarios is needed in order to properly define the effect of time to go.

Also, it would be interesting to extend the model to three dimensions and include the effects of lags on the system in future works. Finally, in appendix A, the computer model used in this work is enclosed. Some improvements in this program can be done, mainly in the data introduction, and in some optimization of the running time.

APPENDIX A

FORTRAN PROGRAM

These appendix provides a listing of the computer program used in the present study.

Since the routines used are non-IMSL, and a small change to double-precision was necessary, they are also being provided.

```
CJOB
C
C    TESE    1      VELLUSC,CAL
C
C      VARIABLES  DECLARATIONS
       REAL*8 A(7,7),B(7,2),AT(7,7),BT(2,7),TAU,DELT,E,DET,TI
       REAL*8 W11(7,7),BTSE(2,1),SG(7,1),BTSG(2,1),BTSEG(2,1)
       REAL*8 S(7,7),SB(7,7),AINV(7,7),B12V,B22V,F3(7,7,201)
       REAL*8 ATPS(7,7),BTPS(2,7),S11A(7,7),BTPSB(2,7),PHII
       REAL*8 BTPSA(2,7),BF(7,7),ABF(7,7)
       REAL*8 R(2,2),RINV(2,2),Z(7,2),XS(7,1,201)
       REAL*8 FG(2,1),BFG(7,1),EBFG(7,1),GS(7,1),F(2,7)
       REAL*8 Q(2,2),ATSG(7,1),AX(7,1),BU(7,1),E1G(7,1)
       REAL*8 P(7,7),PS(7,7),ATYO,UGN(2,1),UG(2,1),U1(2,1)
       REAL*8 YO,XC,ZO,VXO,VYO,VZO,ATXO,YS(201),ZS(201)
       REAL*8 X(7,1),U(2,1),E1(7,1),G(1,1),EG(7,1),UN(2,1)
       REAL*8 ABFX(7,1),KP(201),ACD(201),PCD(201),PHI
       REAL*8 FD11(201),FD12(201),FD13(201),FD14(201),FG3(7,1,201
       REAL*8 A3(7,7,201),B3(7,2,201),AXBU(7,1),G1(1,1)
       REAL*8 AT3(7,7,201),BT3(2,7,201)
       REAL*8 FD15(201),FD16(201),FD17(201),EBFGG(7,1),UC(2,1)
C
       REAL*8 FD21(201),FD22(201),FD23(201),FD24(201)
       REAL*8 FD25(201),FD26(201),FD27(201)
C
       INTEGER I,J,K,N,L,K1,COUNT,CCOUNT,M,KN,KL,NN
       REAL*8 PC1,PHIO,PCO,PH(201)
       REAL  AC(201),KPS(201),PC(201),YM(201),ZM(201),CGS(201)
       REAL  FS11(201),FS12(201),FS13(201),FS14(201)
       REAL  FS15(201),FS16(201),FS17(201),DPH(201)
C
       REAL  FS21(201),FS22(201),FS23(201),FS24(201)
       REAL  FS25(201),FS26(201),FS27(201)
C
       REAL*8 FGD1(201),FGD2(201),FGD3(201),FGD4(201)
       REAL*8 FGD5(201),FGD6(201),FGD7(201)
C
C
C
C
       AMIN=-19.6
          PHI=0.0
          PHIC=0.0
          COUNT=0.0
          TI=5.0
          ACO=26.7
          G1(1,1)=-9.8
C
```

```
C
C
C
    1 IF(COUNT.EQ.101)GC TC 50
C
      K=COUNT+1
C
C     INITIALIZE MATRIX W
C
      DO 3 I=1,7
         DO 2 J=1,7
            W11(I,J)=0.
    2    CONTINUE
    3 CCNTINUE
         W11(1,1)=1.
         W11(4,4)=1.
      DC 5 I=1,7
         SG(I,1)=C.
         DO 4 J=1,7
            S(I,J)=W11(I,J)
    4    CONTINUE
    5 CCNTINUE
C     WRITE(6,25)
C
         YTO=0.0
         YMO=0.
         XTO=250C.
         XMO=0.
         ZTC=100.
         ZMO=0.
         VYTO=0.
         VYMC=0.
         VXTC=500.
         VXMC=10CC.
         VZTC=0.C
         VZMO=0.
C
         YO=YTO-YMO
         XO=XTO-XMO
         ZO=ZTO-ZMO
         VXO=VXTC-VXMO
         VYO=VYTO-VYMO
         VZO=VZTC-VZMO
C
         ATXO=0
         ATYO=-4.*9.8
         ATZO=0.0
         DELPHO=C.
```

```
      COSTET=1.0
      DELPHO=0.0
C
C     INITIALIZATION OF STATE VECTOR X(K)
C
      X(1,1)=YC
      X(2,1)=VYO
      X(3,1)=ATYO
      X(4,1)=ZC
      X(5,1)=VZO
      X(6,1)=ATZO
      X(7,1)=DELPHO
C
C
C ******************************************************************
    DO 6 I=1,7
      XS(I,1,1)=X(I,1)
  6 CONTINUE
C
    DO 7 I=1,7
      X(I,1)=XS(I,1,K)
  7 CONTINUE
C
C     PHIO=PHI
    L=101
    CCOUNT=COUNT+1
C
C
    DO 19 N=K,L
      K1=N-1
C
C
    DO 9 I=1,7
      DO 8 J=1,7
        A(I,J)=0.
  8   CONTINUE
  9 CONTINUE
      TAU=20.
      DELT=.050
      E=2.718
      ACO=26.7
      TI=5.0-K1*T
      COSTET=1.0
      A(1,1)=1.
      A(1,2)=DELT
      A(2,2)=1.
      A(4,4)=1.
      A(5,5)=1.
```

```
      A(7,7)=1.
      A(3,3)=DEXP(-1.*DELT/TAU)
      A(6,6)=DEXP(-1.*DELT/TAU)
      A(4,5)=DELT
      A(1,3)=TAU*DELT-TAU**2*(1.-DEXP(-DELT/TAU))
      A(4,6)=A(1,3)
      A(2,3)=TAU*(1.-DEXP(-DELT/TAU))
      A(5,6)=A(2,3)
      A(1,7)=ACO*DCOS(PHIO)*(DELT**2-((2*K1+1.)/(2*TI))*DELT*
      A(2,7)=ACO*DCOS(PHIO)*(DELT-((2*K1+1.)/(2*TI))*DELT**2)
      A(4,7)=ACO*DSIN(PHIO)*(DELT**2-((2*K1+1.)/(2*TI))*DELT*
      A(5,7)=ACO*DSIN(PHIO)*(DELT-((2*K1+1.)/(2*TI))*DELT**2)
      DO 11 I=1,7
         DO 10 J=1,7
            A3(I,J,N)=A(I,J)
10       CONTINUE
11    CONTINUE
C
      CALL GMTRA(A,AT,7,7)
C
C     INITIALIZE MATRIX B
C
C
      B12V=DELT**3/2.-((2.*K1+1.)/(4.*TI))*DELT**4
C
C
      B22V=DELT**2/2.-((K1*DELT+DELT)**3/(3.*TI))
     1+((K1*DELT+DELT)**2)*K1*DELT/(2.*TI)-((K1*DELT)**3)/(6.*TI
C
      B(1,1)=DSIN(PHIO)*DELT**2/2.
      B(1,2)=ACO*DCOS(PHIO)*(B12V)
      B(2,1)=DELT*DSIN(PHIO)
      B(2,2)=ACO*DCOS(PHIO)*(B22V)
      B(3,1)=0.
      B(3,2)=0.
      B(4,1)=-DCOS(PHIO)*DELT**2/2.
      B(4,2)=ACO*DSIN(PHIO)*(B12V)
      B(5,1)=-DELT*DCOS(PHIO)
      B(5,2)=ACO*DSIN(PHIO)*(B22V)
      B(6,1)=0.
      B(6,2)=0.
      B(7,1)=0.
      B(7,2)=DELT
C
      DO 13 I=1,7
         DO 12 J=1,2
            B3(I,J,N)=B(I,J)
12       CONTINUE
```

```
 13     CONTINUE                                                         TESO1930
        CALL GMTRA(B,BT,7,2)                                             TESO1940
        DO 15 I=1,7                                                      TESO1950
              DO 14 J=1,7                                                TESO1960
                    AT3(I,J,N)=AT(I,J)                                   TESO1970
 14           CONTINUE                                                   TESO1980
 15     CONTINUE                                                         TESO1990
C                                                                        TESO2000
        DO 17 I=1,2                                                      TESO2010
              DO 16 J=1,7                                                TESO2020
                    BT3(I,J,N)=BT(I,J)                                   TESO2030
 16           CONTINUE                                                   TESO2040
 17     CONTINUE                                                         TESO2050
C       INITIALIZE MATRIX E                                              TESO2060
C                                                                        TESO2070
        DO 18 I=1,7                                                      TESO2080
              E1(I,1)=0.                                                 TESO2090
 18     CONTINUE                                                         TESO2100
        E1(4,1)=-DELT**2/2.*COSTET                                       TESO2110
        E1(5,1)=-DELT*COSTET                                             TESO2120
 19     CONTINUE                                                         TESO2130
C       INITIALIZE MATRIX Q                                              TESO2140
C                                                                        TESO2150
        Q(1,1)=.C0578                                                    TESO2160
        Q(1,2)=C.                                                        TESO2170
        Q(2,1)=C.                                                        TESO2180
        Q(2,2)=5.                                                        TESO2190
C                                                                        TESO2200
C     INITIALIZE MATRIX P                                                TESO2210
C                                                                        TESO2220
      DC 21 I=1,7                                                        TESO2230
        DO 20 J=1,7                                                      TESO2240
           P(I,J)=0.                                                     TESO2250
 20     CONTINUE                                                         TESO2260
 21   CCNTINUE                                                           TESO2270
C                                                                        TESO2280
C                                                                        TESO2290
C*************************************************                       TESO2300
C     SCLUTICN OF RICATTI EQUATION                *                      TESO2310
C*************************************************                       TESO2320
      M=101                                                              TESO2330
 22   IF(M.LT.1.)GO TO 31                                                TESO2340
      DC 24 I=1,7                                                        TESO2350
        DO 23 J=1,7                                                      TESO2360
           A(I,J)=A3(I,J,M)                                              TESO2370
           AT(I,J)=AT3(I,J,M)                                            TESO2380
 23     CONTINUE                                                         TESO2390
 24   CCNTINUE                                                           TESO2400
```

```
      CO 26  I=1,7
         DO 25  J=1,2
            B(I,J)=B3(I,J,M)
   25    CONTINUE
   26 CCNTINUE
      DO 28  I=1,2
         DO 27  J=1,7
            BT(I,J)=BT3(I,J,M)
   27    CONTINUE
   28 CCNTINUE
         CALL  GMADD(P,S,PS,7,7)
         CALL  GMPRD(BT,PS,BTPS,2,7,7)
         CALL  GMPRD(BTPS,B,BTPSB,2,7,2)
         CALL  GMFRD(BTPS,E1,BTSE,2,7,1)
         CALL  GMFRD(BT,SG,BTSG,2,7,1)
         CALL  GMADD(BTSE,BTSG,BTSEG,2,1)
         CALL  GMADD(Q,BTPSB,R,2,2)
         CALL  GAUSS3(2,EPS,R,RINV,KER,2)
C
         CALL  GMPRD(BTPS,A,BTPSA,2,7,7)
         CALL  GMFRD(RINV,BTPSA,F,2,2,7)
C
         CALL  GMPRD(RINV,BTSEG,FG,2,2,1)
         DO  30  I=1,2
               FG3(I,1,M)=FG(1,1)
         DO  29  J=1,7
               F3(I,J,M)=F(I,J)
   29    CONTINUE
   30    CONTINUE
CC*******************************************************
C
         CALL  GMFFC(B,F,BF,7,2,7)
         CALL  GMSLB(A,BF,ABF,7,7)
         CALL  GMPRD(AT,PS,ATPS,7,7,7)
C  ******************************************************
         CALL  GMPRD(B,FG,BFG,7,2,1)
         CALL  GMSUB(E1,BFG,EBFG,7,1)
         CALL  GMPRD(ATPS,EBFG,GS,7,7,1)
         CALL  GMPRD(AT,SG,ATSG,7,7,1)
         CALL  GMADD(GS,ATSG,SG,7,1)
C
         CALL  GMPRD(ATPS,ABF,S,7,7,7)
C        WRITE(6,81)COUNT
      M=M-1
      GC TO  22
   31 CCNTINUE
C
C        DEFINITICN OF THE MATRIX G
```

```
C
         G(1,1)=-9.8
C
C
C
C      X(K+1)=(A-EF)X(K)+EG(K)
C
C
               K=CCUNT+1
         KP(K)=K-1
        KN=K-1
   32 IF(KN.EQ.K)GO TO 42
C
      CC 34  I=1,2
         FG(I,1)=FG3(I,1,KN+1)
         DO  33  J=1,7
             F(I,J)=F3(I,J,KN+1)
   33    CONTINUE
   34 CONTINUE
C ***********************************************************************
C
              CALL  GMPRC(F,X,UN,2,7,1)
              CALL  GMPRC(FG,G,UGN,2,1,1)
              DO  35  I=1,2
                   U1(I,1)=-1.*UN(I,1)
                   UG(I,1)=-1.*UGN(I,1)
   35         CONTINUE
C                  UG(1,1)=14.70
C                  UG(2,1)=0.0
C  35         CONTINUE
              CALL  GMADD(U1,UG,U,2,1)
         DO  37  I=1,7
              DO  36  J=1,7
                   AT(I,J)=AT3(I,J,KN+1)
                   A(I,J)=A3(I,J,KN+1)
   36         CONTINUE
   37    CONTINUE
C
         DO  39  I=1,7
              DO  38  J=1,2
                   B(I,J)=B3(I,J,KN+1)
   38         CONTINUE
   39    CONTINUE
         DO  41  I=1,2
              FG(I,1)=FG3(I,1,KN+1)
              DO  40  J=1,7
                   BT(I,J)=BT3(I,J,KN+1)
                   F(I,J)=F3(I,J,KN+1)
```

```
   40         CONTINUE
   41    CONTINUE
C
         CALL GMPRD(B,F,BF,7,2,7)
         CALL GMSUB(A,BF,ABF,7,7)
         CALL GMPRD(ABF,X,ABFX,7,7,1)
         CALL GMPRD(B,FG,BFG,7,2,1)
         CALL GMSUB(E1,BFG,EBFG,7,1)
         CALL GMPRD(EBFG,G,EBFGG,7,1,1)
         CALL GMADD(ABFX,EBFGG,X,7,1)
        KN=KN+1
      GO TO 32
   42 CONTINUE
C
C *************************************************************
C        LIMIT IN COMANDEC ACCELERATION
                IF(U(1,1).LT.AMIN)U(1,1)=AMIN
C               CONTINUE
C *************************************************************
C       MISSILE  MISSILE  MISSILE  MISSILE
C
C        INITIALIZATION OF STATE VECTOR X(K)
C
C
C     INITIALIZE MATRIX A
      K1=COUNT
      DC 44 I=1,7
         DO 43 J=1,7
            A(I,J)=0.
   43    CONTINUE
   44 CONTINUE
         TAU=20.
         DELT=.050
         E=2.718
         ACO=U(1,1)
         COSTET=1.0
             DO 45 I=1,7
                X(I,1)=XS(I,1,K)
   45        CONTINUE
             A(7,7)=0.
         A(1,1)=1.
         A(1,2)=DELT
         A(2,2)=1.
         A(4,4)=1.
         A(5,5)=1.
C        A(7,7)=1.0
         A(3,3)=DEXP(-1.*DELT/TAU)
         A(6,6)=DEXP(-1.*DELT/TAU)
```

```
      A(4,5)=CELT
      A(1,3)=TAU*DELT-TAU**2*(1.-DEXP(-DELT/TAU))
      A(4,6)=A(1,3)
      A(2,3)=TAU*(1.-DEXP(-DELT/TAU))
      A(5,6)=A(2,3)
      A(1,7)=ACO*DCOS(PHIO)*DELT**2/2.
      A(2,7)=ACO*DCOS(PHIO)*DELT
      A(4,7)=ACO*DSIN(PHIO)*DELT**2/2.
      A(5,7)=ACO*DSIN(PHIO)*DELT
      CONTINUE
C     INITIALIZE MATRIX B
C
      B(1,1)=DSIN(PHIO)*DELT**2/2.
      B(1,2)=ACO*DCOS(PHIO)*DELT**3/6.
C     B(1,2)=C.0
      B(2,1)=CELT*DSIN(PHIO)
      B(2,2)=ACO*DCOS(PHIO)*DELT**2/2.
      B(3,1)=C.
      B(3,2)=C.
      B(4,1)=-CCOS(PHIO)*DELT**2/2.
      B(4,2)=ACO*DSIN(PHIO)*DELT**3/6.
C     B(4,2)=C.0
      B(5,1)=-CELT*DCOS(PHIO)
      B(5,2)=ACO*DSIN(PHIO)*DELT**2/2.
      B(6,1)=C.
      B(6,2)=C.
      B(7,1)=C.
      B(7,2)=CELT
C
      IF(K.EQ.1.)GO TO 46
          GO TO 47
   46     CONTINUE
         PH(K)=PHIO
         YS(K)=X(1,1)
         ZS(K)=X(4,1)
   47     CONTINUE
      CALL GMFRD(A,X,AX,7,7,1)
      CALL GMPRD(B,U,BU,7,2,1)
      CALL GMFRD(E1,G1,E1G,7,1,1)
      CALL GMADD(AX,BU,AXBU,7,1)
      CALL GMACD(AXBL,E1G,X,7,1)
      PHIO=PHIC+X(7,1)
C
C
     DO 48 I=1,7
        XS(I,1,K+1)=X(I,1)
   48CONTINUE
C
```

```
          IF(K.GT.1.)GO TO 49                                          TES04330
               GO TO 50                                                TES04340
   49          CONTINUE                                                TES04350
              PH(K)=PHIO                                               TES04360
              YS(K)=X(1,1)                                             TES04370
              ZS(K)=X(4,1)                                             TES04380
   50          CONTINUE                                                TES04390
               ACC(K)=U(1,1)                                           TES04400
              PCD(K)=U(2,1)                                            TES04410
C                                                                      TES04420
C                                                                      TES04430
C                                                                      TES04440
          COUNT=CCUNT+1                                                TES04450
        GO TO 1                                                        TES04460
   51   CONTINUE                                                       TES04470
        DO 3  K=1,101                                                  TES04480
            AC(K)=SNGL(ACD(K))                                         TES04490
            PC(K)=SNGL(PCD(K))                                         TES04500
            YM(K)=SNGL(YS(K))                                          TES04510
            ZM(K)=SNGL(ZS(K))                                          TES04520
            KPS(K)=SNGL(KP(K))                                         TES04530
            DPH(K)=SNGL(PH(K))                                         TES04540
            CGS(K)=SQRT(YM(K)**2+ZM(K)**2)                             TES04550
   52   CONTINUE                                                       TES04560
C                                                                      TES04570
          WRITE(6,25)                                                  TES04580
          WRITE(6,30)                                                  TES04590
          WRITE(6,31)ACD(1),PC(1),A(7,7),ZS(1),DPH(1)                  TES04600
          WRITE(6,31)AC(101),PC(101),YM(101),ZM(101),DPH(101),CGS      TES04610
          WRITE(6,31)AC(10),PC(10),ZM(10),XS(5,1,25),DPH(25),XS(6      TES04620
C         WRITE(6,25)                                                  TES04630
          WRITE(6,35)                                                  TES04640
C *****************************************************************    TES04650
C INVOKE DISSPLA PACKAGE                                               TES04660
C ***********************                                              TES04670
       CALL  TEK618                                                    TES04680
       CALL  HWROT('MOVIE')                                            TES04690
       CALL  AREA2C(7.0,9.0)                                           TES04700
       CALL  XNAME('K$',100)                                           TES04710
       CALL  YNAME('MISS DISTANCE- METERS$',100)                       TES04720
       CALL  HEADIN('4TH CASE$',100,1.,4)                              TES04730
       CALL  HEADIN('INITIAL TARGET ACCELERATION= -4. G$',100,1.,4     TES04740
       CALL  HEADIN('INITIAL TARGET POSITION=-600 M$',100,1.,4)        TES04750
       CALL  HEADIN('SAMPLE PERIOD=0.05 SEC$',100,1.,4)                TES04760
       CALL  CROSS                                                     TES04770
       CALL  GRAF(0.,'SCALE',100.,-600.,'SCALE',100.)                  TES04780
       CALL  CURVE(KPS,ZM,101,-1)                                      TES04790
       CALL  FRAME                                                     TES04800
```

```
      CALL  CURVE(KPS,YM,101,-1)                                           TES04810
      CALL  LINESP(2.0)                                                    TES04820
      CALL  LINES('MISS DIST. Z DIRECTION$',IPAK,1)                        TES04830
      CALL  LINES('MISS DIST. Y DIRECTION$',IPAK,2)                        TES04840
      CALL  LEGEND(IPAK,2,3.5,0.5)                                         TES04850
      CALL  BLREC(3.3,0.3,3.5,1.2,.03)                                     TES04860
      CALL  DASH                                                           TES04870
      CALL  GRID(1,1)                                                      TES04880
      CALL  RESET ('DASH')                                                 TES04890
      CALL  ENDPL(C)                                                       TES04900
      CALL  HWROT('MOVIE')                                                 TES04910
      CALL  AREA2D(7.0,9.0)                                                TES04920
      CALL  XNAME('K$',100)                                                TES04930
      CALL  YNAME('ACCELERATION - METERS/SQ.SEC$',100)                     TES04940
      CALL  HEADIN('4TH CASE$',100,1.,4)                                   TES04950
      CALL  HEADIN('INITIAL TARGET ACCELERATION= -4. G$',100,1.,4          TES04960
      CALL  HEADIN('INITIAL TARGET POSITION=-600 M$',100,1.,4)             TES04970
      CALL  HEADIN('SAMPLE PERIOD=0.05 SEC$',100,1.,4)                     TES04980
      CALL  CROSS                                                          TES04990
      CALL  GRAF(C.,'SCALE',100.,-25.,'SCALE',110.)                        TES05000
      CALL  CURVE(KPS,AC,101,-1)                                           TES05010
      CALL  FRAME                                                          TES05020
      CALL  DASH                                                           TES05030
      CALL  GRID(1,1)                                                      TES05040
      CALL  RESET ('DASH')                                                 TES05050
      CALL  ENDPL(C)                                                       TES05060
      CALL  HWROT('MOVIE')                                                 TES05070
      CALL  AREA2D(7.0,9.0)                                                TES05080
      CALL  XNAME('K$',100)                                                TES05090
      CALL  YNAME(' ROLL RATE- RDS/SEC$',100)                              TES05100
      CALL  HEADIN('4TH CASE$',100,1.,4)                                   TES05110
      CALL  HEADIN('INITIAL TARGET ACCELERATION= -4. G$',100,1.,4          TES05120
      CALL  HEADIN('INITIAL TARGET POSITION=-600 M$',100,1.,4)             TES05130
      CALL  HEADIN('SAMPLE PERIOD=0.05 SEC$',100,1.,4)                     TES05140
      CALL  CROSS                                                          TES05150
      CALL  GRAF(C.,'SCALE',100.,0.,'SCALE',8.)                            TES05160
      CALL  CURVE(KPS,PC,101,-1)                                           TES05170
      CALL  FRAME                                                          TES05180
      CALL  DASH                                                           TES05190
      CALL  GRID(1,1)                                                      TES05200
      CALL  RESET ('DASH')                                                 TES05210
      CALL  ENDPL(C)                                                       TES05220
      CALL  HWROT('MOVIE')                                                 TES05230
      CALL  AREA2D(7.0,9.0)                                                TES05240
      CALL  XNAME('K$',100)                                                TES05250
      CALL  YNAME(' ROLL ANGLE- RDS$',100)                                 TES05260
      CALL  HEADIN('4TH CASE$',100,1.,4)                                   TES05270
      CALL  HEADIN('INITIAL TARGET ACCELERATION= -4. G$',100,1.,4          TES05280
```

```
      CALL HEADIN('INITIAL TARGET POSITION=-600 M$',100,1.,4)          TES05290
      CALL HEADIN('SAMPLE PERIOD=0.05 SEC$',100,1.,4)                  TES05300
      CALL CROSS                                                        TES05310
      CALL FRAME                                                        TES05320
      CALL GRAF(C.,'SCALE',100.,0.,'SCALE',4.)                          TES05330
      CALL DASH                                                         TES05340
      CALL GRID(1,1)                                                    TES05350
      CALL RESET ('DASH')                                               TES05360
      CALL CURVE(KPS,DPH,101,-1)                                        TES05370
      CALL ENDPL(C)                                                     TES05380
      CALL DONEPL                                                       TES05390
C                                                                       TES05400
C *********************************************************************  TES05410
   25    FORMAT('1')                                                    TES05420
   30 FORMAT(1X/,5X,4X,'AC',9X,'PC',9X,'YM',9X,'ZM',9X,'DPH'             TES05430
     1,12X,'CGS')                                                       TES05440
   31 FORMAT(1X/,5X,6(F9.4,3X)//)                                       TES05450
   35    FORMAT(1X//////////////////////////////////////////////,'TYPE C TES05460
      STOP                                                              TES05470
      END                                                               TES05480
C        SUBROUTINE GMTRA                                               TES05490
      SUBROUTINE GMTRA(A,R,N,M)                                         TES05500
      DIMENSION A(1),R(1)                                               TES05510
      DOUBLE PRECISION A,R                                              TES05520
C                                                                       TES05530
      IR=0                                                              TES05540
      DO 10 I=1,N                                                       TES05550
      IJ=I-N                                                            TES05560
      DO 10 J=1,M                                                       TES05570
      IJ=IJ+N                                                           TES05580
      IR=IR+1                                                           TES05590
   10 R(IR)=A(IJ)                                                       TES05600
      RETURN                                                            TES05610
      END                                                               TES05620
C                                                                       TES05630
C     ..................................................................TES05640
C                                                                       TES05650
C        SUBROUTINE GMPRD                                               TES05660
C                                                                       TES05670
C        PURPOSE                                                        TES05680
C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GE        TES05690
C           MATRIX                                                      TES05700
C                                                                       TES05710
C        USAGE                                                          TES05720
C           CALL GMPRD(A,B,R,N,M,L)                                     TES05730
C                                                                       TES05740
C        DESCRIPTION OF PARAMETERS                                      TES05750
C           A - NAME OF FIRST INPUT MATRIX                              TES05760
```

```
C           B - NAME OF SECOND INPUT MATRIX                                     TESO5770
C           R - NAME OF OUTPUT MATRIX                                           TESO5780
C           N - NUMBER CF ROWS IN A                                             TESO5790
C           M - NUMBER CF COLUMNS IN A AND ROWS IN B                            TESO5800
C           L - NUMBER CF COLUMNS IN B                                          TESO5810
C                                                                               TESO5820
C        REMARKS                                                                TESO5830
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES                     TESO5840
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A                 TESO5850
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B                 TESO5860
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TC NUMBE                TESO5870
C           OF MATRIX B                                                         TESO5880
C                                                                               TESO5890
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIREC                          TESO5900
C           NONE                                                                TESO5910
C                                                                               TESO5920
C        METHOD                                                                 TESO5930
C           THE M BY L MATRIX B IS PREMULTIPLIED BY TFE N BY M M                TESO5940
C           AND THE RESLLT IS STORED IN THE N BY L MATRIX R.                    TESO5950
C                                                                               TESO5960
C     ..................................................................       TESO5970
C                                                                               TESO5980
      SUBROUTINE GMPRD(A,B,R,N,M,L)                                             TESO5990
      DIMENSION A(1),B(1),R(1)                                                  TESO6000
      DOUBLE PRECISION A,B,R                                                    TESO6010
C                                                                               TESO6020
      IR=0                                                                      TESO6030
      IK=-M                                                                     TESO6040
      DO 10 K=1,L                                                               TESO6050
      IK=IK+M                                                                   TESO6060
      DO 10 J=1,N                                                               TESO6070
      IR=IR+1                                                                   TESO6080
      JI=J-N                                                                    TESO6090
      IB=IK                                                                     TESO6100
      R(IR)=0                                                                   TESO6110
      DO 10 I=1,M                                                               TESO6120
      JI=JI+N                                                                   TESO6130
      IB=IB+1                                                                   TESO6140
   10 R(IR)=R(IR)+A(JI)*B(IB)                                                   TESO6150
      RETURN                                                                    TESO6160
      END                                                                       TESO6170
C                                                                               TESO6180
C     ..................................................................       TESO6190
C                                                                               TESO6200
C        SUBROUTINE GMSUB                                                       TESO6210
C                                                                               TESO6220
C        PURPOSE                                                                TESO6230
C           SUBTRACT ONE GENERAL MATRIX FROM ANOTHER TO FORM RES                TESO6240
```

```
C        MATRIX                                                          TES06250
C                                                                        TES06260
C     USAGE                                                              TES06270
C        CALL GMSUB(A,B,R,N,M)                                           TES06280
C                                                                        TES06290
C     DESCRIPTION OF PARAMETERS                                          TES06300
C        A - NAME OF FIRST INPUT MATRIX                                  TES06310
C        B - NAME OF SECOND INPUT MATRIX                                 TES06320
C        R - NAME OF OUTPUT MATRIX                                       TES06330
C        N - NUMBER OF ROWS IN A,B,R                                     TES06340
C        M - NUMBER OF COLUMNS IN A,B,R                                  TES06350
C                                                                        TES06360
C     REMARKS                                                            TES06370
C        ALL MATRICES MUST BE STORED AS GENERAL MATRICES                 TES06380
C                                                                        TES06390
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                      TES06400
C        NONE                                                            TES06410
C                                                                        TES06420
C     METHOD                                                             TES06430
C        MATRIX B ELEMENTS ARE SUBTRACTED FROM CORRESPONDING             TES06440
C        ELEMENTS                                                        TES06450
C                                                                        TES06460
C  ..................................................................    TES06470
C                                                                        TES06480
      SUBROUTINE GMSUB(A,B,R,N,M)                                        TES06490
      DIMENSION A(1),B(1),R(1)                                           TES06500
      DOUBLE PRECISION A,B,R                                             TES06510
C                                                                        TES06520
C     CALCULATE NUMBER OF ELEMENTS                                       TES06530
C                                                                        TES06540
      NM=N*M                                                             TES06550
C                                                                        TES06560
C     SUBTRACT MATRICES                                                  TES06570
C                                                                        TES06580
      DO 10 I=1,NM                                                       TES06590
   10 R(I)=A(I)-B(I)                                                     TES06600
      RETURN                                                             TES06610
      END                                                                TES06620
C                                                                        TES06630
C  ..................................................................    TES06640
C                                                                        TES06650
C     SUBROUTINE GMADD                                                   TES06660
C                                                                        TES06670
C     PURPOSE                                                            TES06680
C        ADD TWO GENERAL MATRICES TO FORM RESULTANT GENERAL M            TES06690
C                                                                        TES06700
C     USAGE                                                              TES06710
C        CALL GMADD(A,B,R,N,M)                                           TES06720
```

```
C                                                                       TESO6730
C     DESCRIPTICN OF PARAMETERS                                         TESO6740
C        A - NAME OF FIRST INPUT MATRIX                                 TESO6750
C        B - NAME OF SECOND INPUT MATRIX                                TESO6760
C        R - NAME OF CUTPUT MATRIX                                      TESO6770
C        N - NLMBER CF ROWS IN A,B,R                                    TESO6780
C        M - NLMBER CF COLUMNS IN A,B,R                                 TESO6790
C                                                                       TESO6800
C     REMARKS                                                           TESO6810
C        ALL MATRICES MLST BE STORED AS GENERAL MATRICES                TESO6820
C                                                                       TESO6830
C     SUBROUTINES AND FLNCTION SUBPROGRAMS REQUIREC                     TESO6840
C        NONE                                                           TESO6850
C                                                                       TESO6860
C     METHOD                                                            TESO6870
C        ADDITICN IS PERFORMED ELEMENT BY ELEMENT                       TESO6880
C                                                                       TESO6890
C     ..................................................................TESO6900
C                                                                       TESO6910
      SLBROUTINE GMADD(A,B,R,N,M)                                       TESO6920
      DIMENSION A(1),B(1),R(1)                                          TESO6930
      DCUBLE PRECISION A,B,R                                            TESO6940
C                                                                       TESO6950
C        CALCULATE NUMBER CF ELEMENTS                                   TESO6960
C                                                                       TESO6970
      NM=N*M                                                            TESO6980
C                                                                       TESO6990
C        ADD MATRICES                                                   TESO7000
C                                                                       TESO7010
      DC 10 I=1,NM                                                      TESO7020
   1C R(I)=A(I)+B(I)                                                    TESO7030
      RETURN                                                            TESO7040
      END                                                               TESO7050
C                                                                       TESO7060
C-----------------------------------------------------------------------TESO7070
C                                                                       TESO7080
C       SUBROUTINE GAUSS3                                               TESO7090
C                                                                       TESO7100
C       PURPOSE                                                         TESO7110
C          INVERT A DOUBLE PRECISION MATRIX BY THE GAUSS-JCRDAN M       TESO7120
C          THIS RCLTINE IS A DOUBLE PRECISION VERSION CF SSP ROUT       TESO7130
C          MINV USING F1-NPGS-GAUSS3 (F-63) CALLING SEQUENCE            TESO7140
C                                                                       TESO7150
C       LSAGE                                                           TESO7160
C          CALL GALSS3(N,EPS,A,X,KER,K)                                 TESO7170
C                                                                       TESO7180
C       DESCRIPTICN OF PARAMETERS                                       TESO7190
C          N:   ORCER OF MATRIX                                         TESO7200
```

```
      GO TO 140
  120 IJ=0
      NM=0
      DO 130 K=1,J
      DO 125 L=1,I
      IJ=IJ+1
      NM=NM+1
  125 S(IJ)=C(NM)
  130 NM=NM+NI
  140 RETURN
      END
C
C     ..................................................................
C
C        SUBROUTINE DMINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL DMINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A AND D MUST BE REAL*8*8
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLA
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERM
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES
C           THE MATRIX IS SINGULAR.
C
C     ..................................................................
C
      SUBROUTINE DMINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      DOUBLE PRECISION A,D,BIGA,HOLD
C
```

```
C          .......................................................  TES08170
C                                                                   TES08180
C          SEARCH FOR LARGEST ELEMENT                               TES08190
C                                                                   TES08200
      D=1.0D0                                                       TES08210
      NK=-N                                                         TES08220
      DO 80 K=1,N                                                   TES08230
      NK=NK+N                                                       TES08240
      L(K)=K                                                        TES08250
      M(K)=K                                                        TES08260
      KK=NK+K                                                       TES08270
      BIGA=A(KK)                                                    TES08280
      DO 20 J=K,N                                                   TES08290
      IZ=N*(J-1)                                                    TES08300
      DO 20 I=K,N                                                   TES08310
      IJ=IZ+I                                                       TES08320
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20                           TES08330
   15 BIGA=A(IJ)                                                    TES08340
      L(K)=I                                                        TES08350
      M(K)=J                                                        TES08360
   20 CONTINUE                                                      TES08370
C                                                                   TES08380
C          INTERCHANGE ROWS                                         TES08390
C                                                                   TES08400
      J=L(K)                                                        TES08410
      IF(J-K) 35,35,25                                              TES08420
   25 KI=K-N                                                        TES08430
      DO 30 I=1,N                                                   TES08440
      KI=KI+N                                                       TES08450
      HOLD=-A(KI)                                                   TES08460
      JI=KI-K+J                                                     TES08470
      A(KI)=A(JI)                                                   TES08480
   30 A(JI) =HOLD                                                   TES08490
C                                                                   TES08500
C          INTERCHANGE COLUMNS                                      TES08510
C                                                                   TES08520
   35 I=M(K)                                                        TES08530
      IF(I-K) 45,45,38                                              TES08540
   38 JP=N*(I-1)                                                    TES08550
      DO 40 J=1,N                                                   TES08560
      JK=NK+J                                                       TES08570
      JI=JP+J                                                       TES08580
      HOLD=-A(JK)                                                   TES08590
      A(JK)=A(JI)                                                   TES08600
   40 A(JI) =HOLD                                                   TES08610
C                                                                   TES08620
C          DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS  TES08630
C          CONTAINED IN BIGA)                                       TES08640
```

```
C                                                                       TES08650
   45 IF(BIGA) 48,46,48                                                 TES08660
   46 D=0.0D0                                                           TES08670
      RETURN                                                            TES08680
   48 DO 55 I=1,N                                                       TES08690
      IF(I-K) 50,55,50                                                  TES08700
   50 IK=NK+I                                                           TES08710
      A(IK)=A(IK)/(-BIGA)                                               TES08720
   55 CONTINUE                                                          TES08730
C                                                                       TES08740
C        REDUCE MATRIX                                                  TES08750
C                                                                       TES08760
      DO 65 I=1,N                                                       TES08770
      IK=NK+I                                                           TES08780
      HOLD=A(IK)                                                        TES08790
      IJ=I-N                                                            TES08800
      DO 65 J=1,N                                                       TES08810
      IJ=IJ+N                                                           TES08820
      IF(I-K) 60,65,60                                                  TES08830
   60 IF(J-K) 62,65,62                                                  TES08840
   62 KJ=IJ-I+K                                                         TES08850
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            TES08860
   65 CONTINUE                                                          TES08870
C                                                                       TES08880
C        DIVIDE ROW BY PIVOT                                            TES08890
C                                                                       TES08900
      KJ=K-N                                                            TES08910
      DO 75 J=1,N                                                       TES08920
      KJ=KJ+N                                                           TES08930
      IF(J-K) 70,75,70                                                  TES08940
   70 A(KJ)=A(KJ)/BIGA                                                  TES08950
   75 CONTINUE                                                          TES08960
C                                                                       TES08970
C        PRODUCT OF PIVOTS                                              TES08980
C                                                                       TES08990
      D=D*BIGA                                                          TES09000
C                                                                       TES09010
C        REPLACE PIVOT BY RECIPRUCAL                                    TES09020
C                                                                       TES09030
      A(KK)=1.0/BIGA                                                    TES09040
   80 CONTINUE                                                          TES09050
C                                                                       TES09060
C        FINAL ROW AND COLUMN INTERCHANGE                               TES09070
C                                                                       TES09080
      K=N                                                               TES09090
  100 K=(K-1)                                                           TES09100
      IF(K) 150,150,105                                                 TES09110
  105 I=L(K)                                                            TES09120
```

```
      IF(I-K) 12C,120,1C8                                               TES09130
  108 JC=N*(K-1)                                                        TES09140
      JR=N*(I-1)                                                        TES09150
      DO 110 J=1,N                                                      TES09160
      JK=JQ+J                                                           TES09170
      HCLD=A(JK)                                                        TES09180
      JI=JR+J                                                           TES09190
      A(JK)=-A(JI)                                                      TES09200
  11C A(JI) =HOLD                                                       TES09210
  12C J=M(K)                                                            TES09220
      IF(J-K) 10C,100,125                                               TES09230
  125 KI=K-N                                                            TES09240
      DC 130 I=1,N                                                      TES09250
      KI=KI+N                                                           TES09260
      HCLD=A(KI)                                                        TES09270
      JI=KI-K+J                                                         TES09280
      A(KI)=-A(JI)                                                      TES09290
  130 A(JI) =HOLC                                                       TES09300
      GC TO 100                                                         TES09310
  15C RETURN                                                            TES09320
      END                                                               TES09330
 CENTFY                                                                 TES09340
```

LIST OF REFERENCES

1. Stallard, D. V., "An Approach to Optimal Guidance for a Bank-to-Turn Missile," Proceedings of AIAA Guidance and Control Conference, Danvers, Mass., August 1980.

2. Kwakernaak, H. and Sivan, R. Linear Optimal Control Systems, Wiley-Interscience, 1972.

3. Bryson, A. E. and Ho, Y. C. Applied Optimal Control, Halsted Press, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor D.J. Collins, Code 67Co Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	3
5.	Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
6.	Office of The Air Attache Brazilian Embassy 3006 Massachussets Av NW Washington DC 20008	3
7.	Centro Tecnico Aeroespacial Rua Paraibuna S/N 12200 Sao Jose dos Campos - SP Sao Paulo, BRAZIL	1
8.	Centro Tecnico Aeroespacial Instituto de Atividades Espaciais Rua Paraibuna S/N 12200 Sao Jose dos Campos - SP Sao Paulo, BRAZIL	2
9.	Centro Tecnico Aeroespacial Instituto de Atividades Espaciais Divisao de Sistemas Belicos Rua Paraibuna S/N 12200 Sao Jose dos Campos - SP Sao Paulo, BRAZIL	2
10.	MAJ Carlos A. L. Velloso Centro Tecnico Aeroespacial Instituto de Atividades Espaciais Rua Paraibuna S/N 12200 Sao Jose dos Campos - SP Sao Paulo, BRAZIL	3

207793

Thesis
V36282 Velloso
c.1 Optimal digital control of a bank-to-turn missile.