







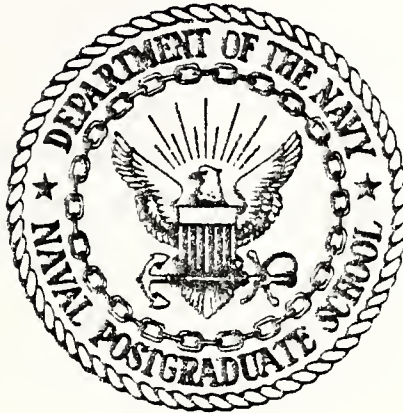






# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

COMPUTER SIMULATED DEVELOPMENT  
OF IMPROVED  
COMMAND TO LINE-OF-SIGHT  
MISSILE GUIDANCE TECHNIQUES

by

Frank F. Hewitt

March 1979

Thesis Advisor:

H. A. Titus

Approved for public release; distribution unlimited.

T187802





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) Computer Simulated Development of Improved Command to Line-of-Sight Missile Guidance Techniques		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1979
7. AUTHOR(s) Frank F. Hewitt		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE March 1979
		13. NUMBER OF PAGES 106
		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) Line-of-sight Guidance Digital missile simulation Missile guidance		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Three computer simulations of command to line-of-sight missile guidance systems were developed within an electronic warfare environment to test the feasibility of improved guidance techniques while maintaining simplicity for generic application among the various missiles of this general guidance classification. The first simulation modeled the basic guidance scheme; the second introduced a "lead-angle"		



concept; and the third simulation combined these techniques for use depending upon whether the target took evasive action or employed electronic countermeasures. It was found that consideration should be given for use of a "lead-angle" variant in conjunction with the basic guidance technique to enhance the effective engagement envelope of these missile systems against relatively slower-maneuvering targets employing only low-duty-cycle denial jamming.



COMPUTER SIMULATED DEVELOPMENT  
OF IMPROVED  
COMMAND TO LINE-OF-SIGHT MISSILE GUIDANCE TECHNIQUES

by

Frank F. Hewitt  
Lieutenant Commander, United States Navy  
B.S., United States Naval Academy, 1967


Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL

March 1979





## ABSTRACT

Three computer simulations of command to line-of-sight missile guidance systems were developed within an electronic warfare environment to test the feasibility of improved guidance techniques while maintaining simplicity for generic application among the various missiles of this general guidance classification. The first simulation modeled the basic guidance scheme; the second introduced a "lead-angle" concept; and the third simulation combined these techniques for use depending upon whether the target took evasive action or employed electronic countermeasures. It was found that consideration should be given for use of a "lead-angle" variant in conjunction with the basic guidance technique to enhance the effective engagement envelope of these missile systems against relatively slower-maneuvering targets employing only low-duty-cycle denial jamming.





TABLE OF CONTENTS

I.	INTRODUCTION-----	12
II.	OVER-VIEW OF GUIDANCE TYPES-----	13
III.	TYPICAL ENGAGEMENT SEQUENCE-----	15
IV.	COMMAND TO LINE-OF-SIGHT GUIDANCE GEOMETRY-----	19
V.	ON-OFF, THRUST-VECTOR, MISSILE CONTROL-----	21
VI.	'BASIC' COMMAND TO LINE-OF-SIGHT SIMULATION-----	32
	A. PROGRAM DESCRIPTION-----	32
	B. SYSTEM BLOCK DIAGRAM-----	35
	C. SIMULATION RESULTS-----	35
	D. EFFECTS OF DECEPTIVE JAMMING-----	39
VII.	'LEAD-ANGLE' COMMAND TO LINE-OF-SIGHT SIMULATION-----	42
	A. PROGRAM DESCRIPTION-----	42
	B. SYSTEM BLOCK DIAGRAM-----	43
	C. SIMULATION RESULTS-----	44
VIII.	'CONSOLIDATED' COMMAND TO LINE-OF-SIGHT SIMULATION----	47
	A. PROGRAM DESCRIPTION-----	47
	B. SYSTEM BLOCK DIAGRAM -----	52
	C. SIMULATION RESULTS-----	54
IX.	CONCLUSIONS-----	60
	APPENDIX A DESCRIPTION AND USE OF SUBROUTINE INTEG2-----	61
	COMPUTER PROGRAMS-----	68
	'BASIC' COMMAND TO LINE-OF-SIGHT SIMULATION-----	68
	'LEAD-ANGLE' COMMAND TO LINE-OF-SIGHT SIMULATION-----	74
	'CONSOLIDATED' COMMAND TO LINE-OF-SIGHT SIMULATION----	82
	SUBROUTINE INTEG2-----	95



BIBLIOGRAPHY----- 104

INITIAL DISTRIBUTION LIST----- 105



LIST OF FIGURES

FIGURE		PAGE
III-1	Roland Missile System Operational Schematic	18
IV-1	'Basic' Command to Line-of-Sight Geometry	19
IV-2	'Lead-Angle' Command to Line-of-Sight Geometry	20
V-1	Missile Cross-Range Error with Zero Initial Values for Cross-Range Error and Error Rate	21
V-2	Phase Plane When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate	22
V-3	Missile & Target Positional Coordinates When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate	23
V-4	Missile & Target Flight Paths When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate	24
V-5	Missile Cross-Range Error When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate	24
V-6	Optimum Parabolic Switching Boundary	25
V-7	Phase Plane When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate	26
V-8	Missile & Target Positional Coordinates When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate	27
V-9	Missile & Target Flight Paths When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate	27
V-10	Missile Cross-Range Error When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate	28



FIGURE		PAGE
V-11	Elementary Command to Line-of-Sight System Block Diagram	31
VI-1	'Basic' Computer Simulation Block Diagram	35
VI-2	'Basic' Simulation Phase Plane	36
VI-3	'Basic' Simulation Missile & Target Positional Coordinates	37
VI-4	'Basic' Simulation Missile & Target Flight Paths	37
VI-5	'Basic' Simulation Missile Cross-Range Error	38
VI-6	'Basic' Simulation Phase Plane When System Deceptively Jammed	40
VI-7	'Basic' Simulation Missile & Target Flight Paths When System Deceptively Jammed	41
VI-8	'Basic' Simulation Missile Cross-Range Error When System Deceptively Jammed	41
VII-1	'Lead-Angle' Computer Simulation Block Diagram	43
VII-2	'Lead-Angle' Simulation Phase Plane	44
VII-3	'Lead-Angle' Simulation Missile & Target Positional Coordinates	45
VII-4	'Lead-Angle' Simulation Missile & Target Flight Paths	45
VII-5	'Lead-Angle' Simulation Missile Cross-Range Error	46
VIII-1	'Consolidated' Simulation Missile Velocity Vectors	48
VIII-2	'Consolidated' Simulation Missile Acceleration Vectors	49
VIII-3	'Consolidated' Computer Simulation Block Diagram	53
VIII-4	'Consolidated' Simulation Missile & Target Flight Paths When Continuously Jammed	54
VIII-5	'Consolidated' Simulation Missile & Target Flight Paths When Not Jammed	55





FIGURE		PAGE
VIII-6	'Consolidated' Simulation Missile & Target Flight Paths When Intermittently Jammed	55
VIII-7	'Consolidated' Simulation Missile Cross-Range Error With Full Missile Control Surface Deflections	57
VIII-8	'Consolidated' Simulation Missile Cross-Range Error with Reductions in Missile Control Surface Deflections	57
VIII-9	'Consolidated' Simulation Phase Plane With Full Missile Control Surface Deflections	58
VIII-10	'Consolidated' Simulation Phase Plane With Reductions in Missile Control Surface Deflections	58



TABLE OF VARIABLES

DIAGRAM VARIABLES	COMPUTER VARIABLES	NOUN DESCRIPTION
CRE	X(1)	cross-range error
EX	EX	error function
G	G	In the 'consolidated' simulation this is the magnitude of missile's Y-direction acceleration. In the other simulations this is the missile's cross-range acceleration.
LOS	SIGT	line of sight to target
$R_m$	RM	missile range from tracker
SLOS	SLOS	synthetic line of sight
$\dot{SLOS}$	SLOSD	rate of synthetic line-of-sight's movement
t	T	time
U	U	In the 'consolidated' simulation this is the missile's Y-direction acceleration. In the other simulations this is the missile's cross-range acceleration.
$X_i$	X(18)	impact point's "X" coordinate
$\dot{X}_i$	VIX	X-direction velocity of impact point
$Y_i$	X(17)	impact point's "Y" coordinate
$\dot{Y}_i$	VIY	Y-direction velocity of impact point
$X_m$	X(8)	missile's "X" coordinate
$\dot{X}_m$	VMX	X-direction velocity of missile
$\ddot{X}_m$	VMXD	X-direction acceleration of missile



DIAGRAM VARIABLES	COMPUTER VARIABLES	NOUN DESCRIPTION
$X_t$	X(28)	target's "X" coordinate
$\dot{X}_t$	VTX	X-direction velocity of target
$\ddot{X}_t$	VTXD	X-direction acceleration of target
$Y_m$	X(7)	missile's "Y" coordinate
$\dot{Y}_m$	VMY	Y-direction velocity of missile
$\ddot{Y}_m$	VMYD	Y-direction acceleration of missile
$Y_t$	X(27)	target's "Y" coordinate
$\dot{Y}_t$	VTY	Y-direction velocity of target
$\ddot{Y}_t$	VTYD	Y-direction acceleration of target
$\epsilon$	E	angular error
$\dot{\epsilon}$	EDOT	angular error rate
$\sigma_m$	SIGM	angle of missile from reference
$\dot{\sigma}_m$	SIGMD	angular rate of missile movement
$\sigma_t$	SIGT	angle of target from reference
$\dot{\sigma}_t$	SIGTD	angular rate of target movement
1/s	(none)	integration



## I. INTRODUCTION

Command to line-of-sight guidance for several types of surface-to-air missiles was developed through computer simulation. Consideration was given to electronic countermeasures against the guidance methods of this type.

The simulations are of sufficient complexity to demonstrate feasibility while maintaining simplicity for generic application among the various missiles of this general guidance classification.





## II. OVER-VIEW OF GUIDANCE TYPES

Guidance systems can be considered to fall, generally, into one of three basic types: line of sight, pursuit, or proportional navigation.

The line-of-sight system maintains the missile on the vector from the ground-based point of missile control (the tracker) to the target aircraft. Variations to this system can provide for "lead-angle" intercept through missile command computer generation of a synthetic line of sight upon which the missile is commanded to fly for target intercept. Also, beam-riding missiles fit this category. However, in the beam-riding type of line-of-sight guidance the missile avionics are more complicated and expensive because the missile normally senses its position relative to the beam center and generates its own guidance commands internally. This extends the missile's conceivable range but usually necessitates some form of terminal homing. Basic command to line-of-sight systems can utilize relatively cheap missiles by performing guidance computations in the tracker unit and not in the missile itself. This limits the missile to short-range scenarios and requires that the missile possess a greater speed advantage over the target.

Pursuit guidance systems maintain the missile's velocity vector toward the target. Generally, sensors in the missile can be assumed to perform this function; therefore, the missile can be forgotten by the launcher after the missile has locked onto the target. This type of system is capable of medium range work against a slowly-maneuvering target. It has



relatively simple processing avionics.

Proportional navigation guidance systems hold a constant angle between the missile-to-target line of sight and the missile axis, thereby, generating a "constant-bearing-decreasing-range", collision situation. In other words, the change in missile heading is adjusted proportionally to the rate of change in the missile-to-target line of sight. This type of guidance is good for long ranges (and short) against highly-maneuvering targets. However, the missile requires either seeker and angle-rate sensing, or a command guidance signal based upon a collision course.



### III. TYPICAL ENGAGEMENT SEQUENCE

In order to provide a "vehicle" through which to better understand the basic aspects of command to line-of-sight guidance methodology, the engagement sequence of a short-range, air-defense, missile system is described. The Roland system was selected because the general operational aspects of the system are available at the unclassified level [1].

The entry of one or more aerial targets into the range of the search radar is indicated to the Roland vehicle commander by an audible tone. At the same time, a synthetic display of the targets appears on a screen to give the commander all the information needed to select the most threatening target. The screen images are different for friendly and enemy targets. Also, the entry of the target into the missile envelope, utilizing target advanced-range computations, is indicated by a change in the display. With the search antenna raised and the search radar activated, target acquisition is possible even when the vehicle is in motion.

There are three modes of identification, friend or foe (IFF) interrogation: automatic, manual, and automatic within a given range.

When the commander has recognized a target as hostile and decided to engage it, he places a cursor over the screen image. This automatically brings the turret to bear and tracking can commence in either the "radar" or "optical" modes.



In the "radar" mode, the tracking radar automatically accepts target designation from the search radar, searches for, locks onto, and tracks the target.

In the "optical" mode, the aimer searches for the target in elevation with an optical sight. To aid him an electronic instrument displays the maximum theoretical elevation for the search. When the aimer has acquired the target in his cross-hairs, he keeps the target in his sight by manipulating a control stick. This control keeps the target properly positioned by moving the turret in azimuth and swivelling a mirror in elevation.

As soon as the commander confirms that the target is within missile range, he initiates the firing sequence in the "radar" mode, or authorizes "optical" mode firing through a command displayed in the aimer's sight. The aimer, then, can initiate the firing sequence.

The missile is guided by a command to line-of-sight technique. This means that the target is tracked optically or by radar and the deviation of the missile from this line of sight is determined and corrected by a guidance command. The commander may switch from "radar" to "optical" and back again, as desired, even after the missile has been launched.

Target tracking and determination of the missile's deviation from the line of sight are different for each mode. In the "radar" mode, the guidance radar has two receiving channels. One is used for target tracking and the other is used to locate the missile in the radar lobe through reception of the





missile's radio frequency beacons. By comparing these angles, an error between the missile and the target line of sight can be determined. In the "optical" mode, a biaxially-stabilized mirror is manually controlled to keep the target vertically in the aimer's sight and the turret is rotated to the azimuth of the target line of sight. An infrared goniometer is mounted to provide missile angle from the tracker by following flares mounted on the rear of the missile. Then, a deviation of the missile angle from the target line of sight can be determined.

Two groups of signals are introduced into the command computer: the velocity of the line of sight in azimuth and elevation, and the deviation of the missile from the line of sight in azimuth and elevation. Based upon data from the line-of-sight movement and the angular deviations of the missile, the necessary guidance signals are calculated.

The guidance signals are relayed to the missile by a transmitter with highly directional characteristics. The command-transmitting antenna is slaved to the missile angle in both azimuth and elevation. It, therefore, is trained on the missile continuously.

The side forces required for missile course corrections are produced through deflection of the exhaust jet of the sustainer motor by spoilers at the rear of the missile (thrust-vector control).

When the missile reaches the point of impact with the target, the warhead is detonated by either percussion, contact fuse or the radio-frequency, proximity fuse. The warhead



consists of a radial-effect, multiple-fragmentation charge.

Figure III-1 presents an operational schematic of the basic Roland missile system operation.

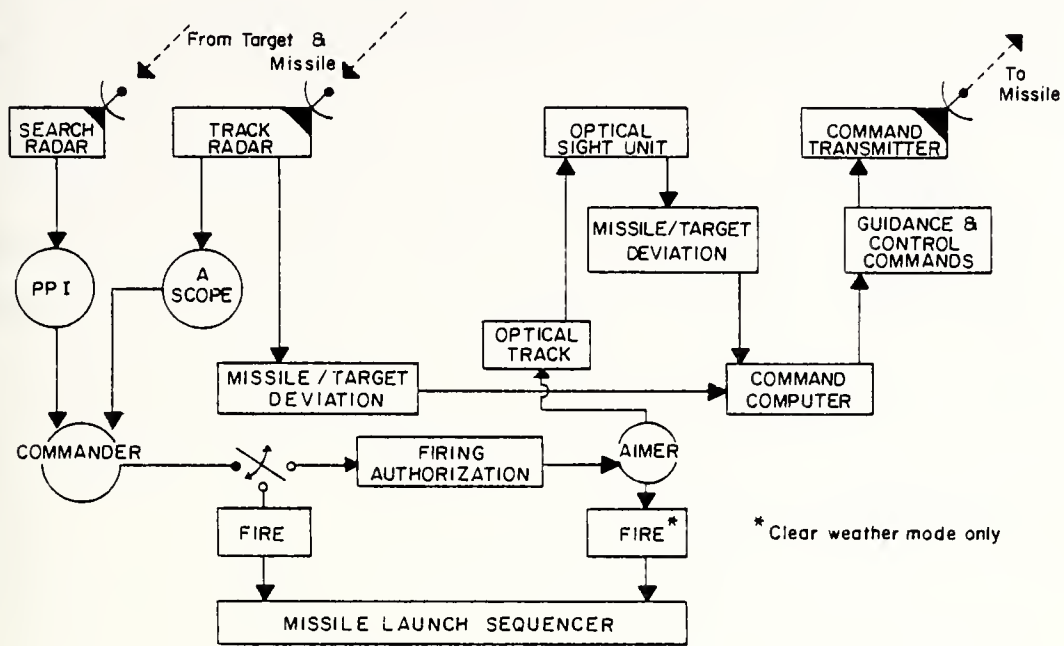


Figure III-1 Roland Missile System Operational Schematic

The computer simulations contained herein are generic in nature within the command to line-of-sight guided-missile type and have only reasonable estimates of missile capabilities introduced. This ensures unclassified results. At the same time, the simulations are of sufficient complexity to properly weigh the relative merits of the guidance variations discussed.



#### IV. COMMAND TO LINE-OF-SIGHT GUIDANCE GEOMETRY

There are many tactical, inherently short-ranged, surface-to-air missiles which use this guidance technique for their operation. Its objective, basically, is to keep the error term ( $\epsilon$ ) zero so that the missile flies directly up the line of sight.

Since only angles to the target and to the missile are required for guidance, attempts to preclude target range through denial jamming prove useless against this type of guidance system. The only necessary information is, in general, to know whether the target is within the missile envelope. Deceptive down-link jamming, however, appears to be feasible. One possible method is presented in Section VI.

The geometry depicted in Figure IV-1 summarizes the "basic" command to line-of-sight geometric situation.

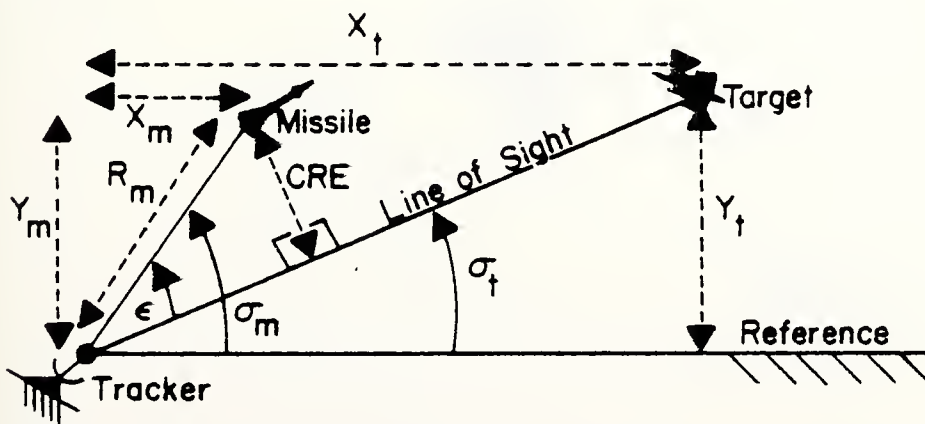


Figure IV-1 "Basic" Command to Line-of-Sight Geometry



A variant of this "basic" command to line-of-sight guidance approach, which shows promise for increasing system effectiveness, is to incorporate "lead-angle" into the basic guidance model. This technique would increase the effective range of the missile against a non-maneuvering, crossing target; however, it is degraded by a highly-maneuvering target. Since range information is required to calculate the impact point, additional electronic countermeasures could be employed by the target to deny this range information thus negating the "lead-angle" command to line-of-sight missile guidance scheme.

The previous "basic" geometry has been retained but it has been modified to utilize a synthetic line of sight upon which the missile flies. This synthetic line of sight is the direction of the impact point from the ground tracking unit. Therefore, the missile is commanded to fly toward the impact point rather than directly toward the target. This method, then, reduces the missile's flight time until impact, thus, improving the missile system's effective engagement range. Figure IV-2 shows this geometry.

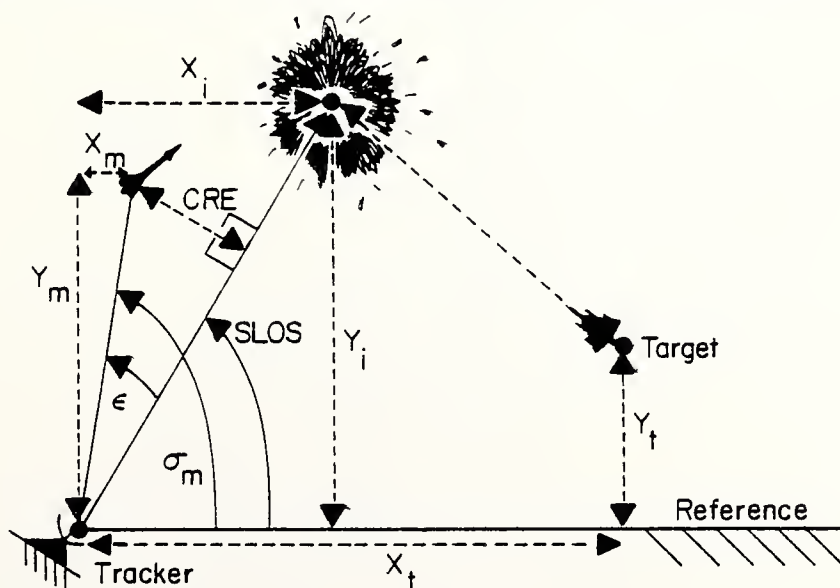


Figure IV-2 "Lead-Angle" Command to Line-of-Sight Geometry





## V. ON-OFF, THRUST-VECTOR, MISSILE CONTROL

The guidance models used in the computer simulations utilize on-off control. This type of control also is known as relay control or "bang-bang" control because the system's actuator is not moved in proportion to the error signal but, rather, is positioned either totally in the positive direction or totally in the negative direction. This causes the missile's thrust-vector control to apply missile-maneuvering thrust fully in one direction or the other. At first glance this type of system might appear to be very crude; however, it does have major advantages. First, the system does not require complex proportional amplification; the unit is smaller than its proportional control counterpart; it is, generally, the time-optimal control system; it weighs less than other systems; and, finally, it is relatively inexpensive [2].

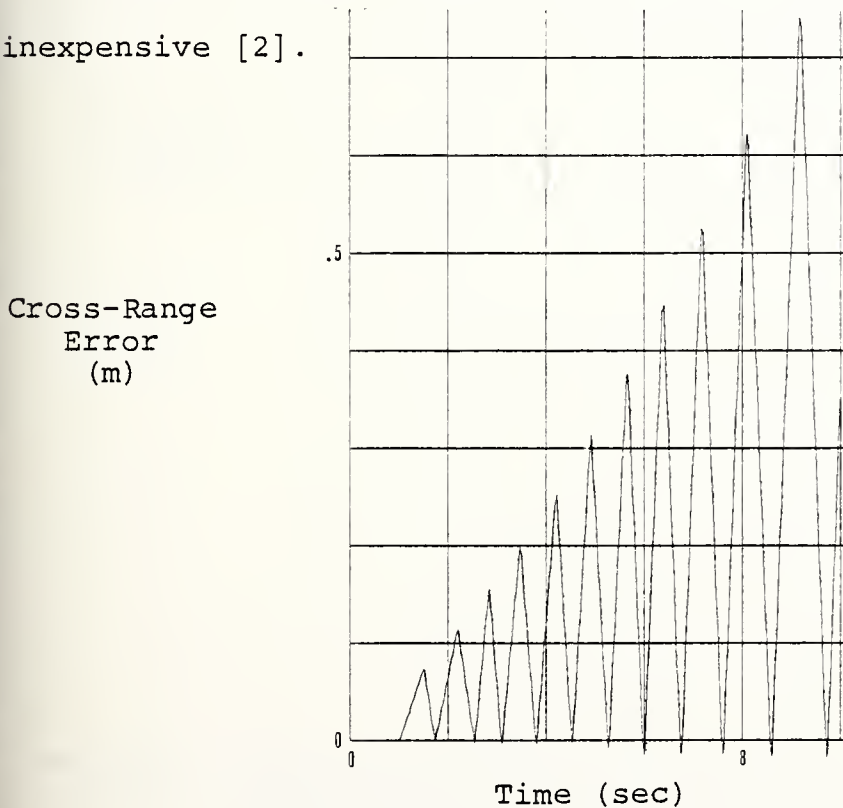


Figure V-1 Missile Cross-Range Error with Zero Initial Values for Cross-Range Error and Error Rate



Based upon this type of missile control system, initial computer simulations, not presented here in detail, were designed. They provided an elementary computer program structure which was modified incrementally to generate improved computer models. In these first programs, the error values were introduced directly into the missile's thrust-vector control without error rate. This produced increased missile cross-range error with flight time as shown in Figure V-1. This effect has been eliminated in the subsequent computer guidance models by proper acceleration-limiting techniques. Additionally, if initial erroneous values of cross-range error and cross-range error rate are introduced

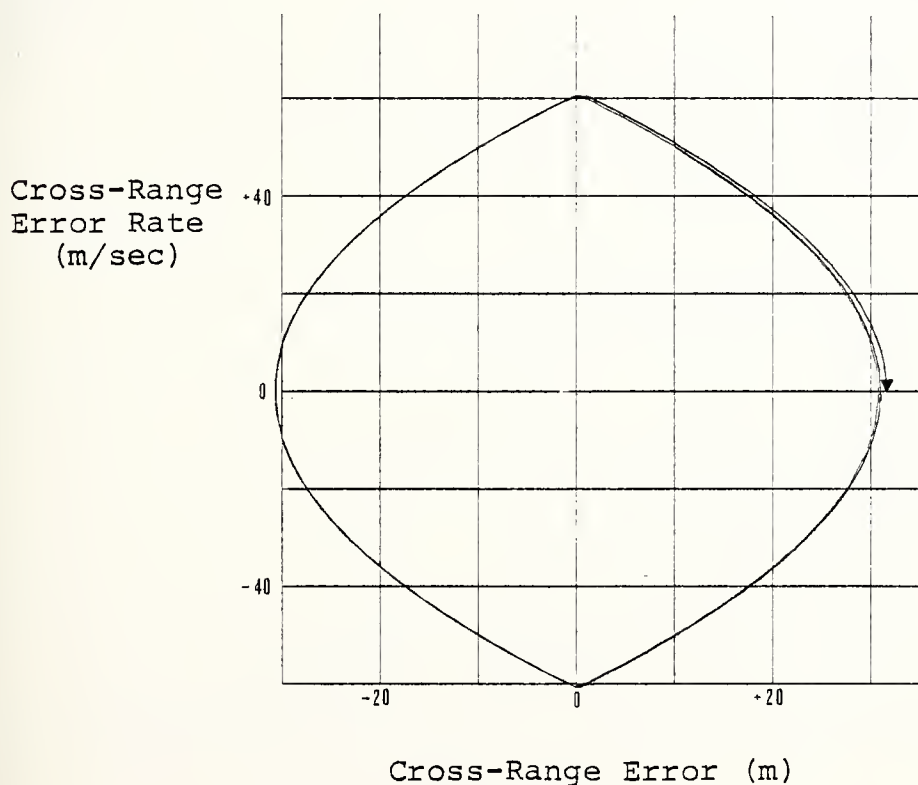


Figure V-2 Phase Plane When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate



just after missile launch, this "basic" guidance system does not dampen their effects. This increases the probability of having a greater miss distance. With an initial cross-range error of ten meters and with an initial cross-range error rate of fifty meters per second applied, the undamped effect, plus increased cross-range error with time, combine to cause the missile to miss the target. Results are shown in Figures V-2 through V-4. Although not readily apparent in Figure V-4, the

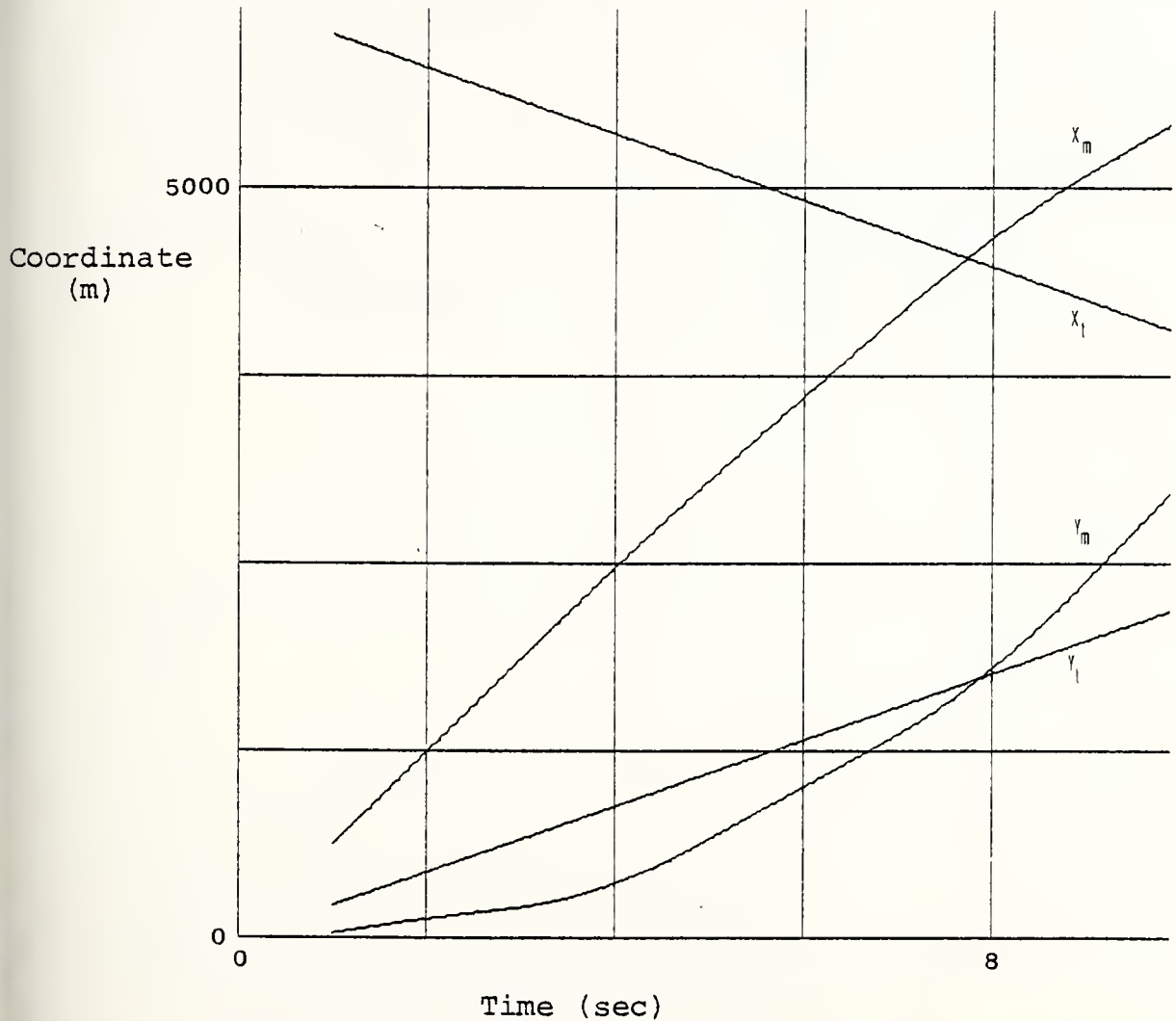


Figure V-3 Missile & Target Positional Coordinates When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate



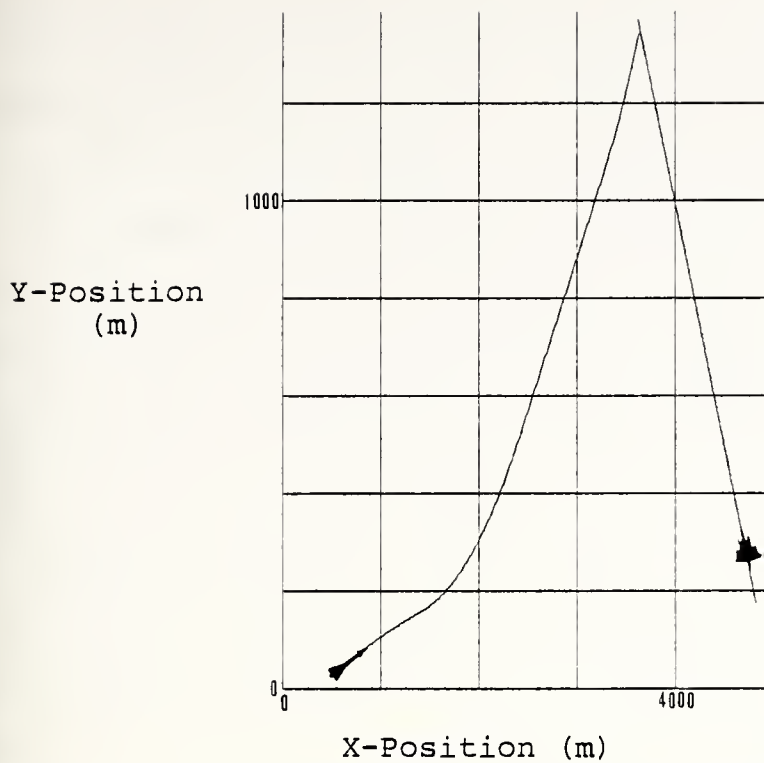


Figure V-4 Missile & Target Flight Paths When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate

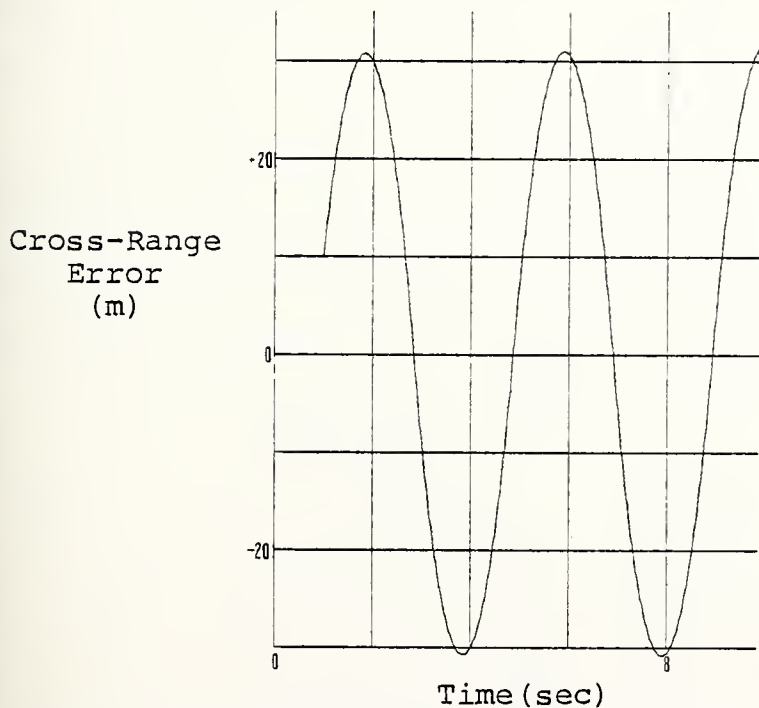


Figure V-5 Missile Cross-Range Error When Control Function By-Passed With Initial Values Introduced for Cross-Range Error and Error Rate





missile moves about the line of sight with a cross-range error varying as shown in Figure V-5. The cross-range error does increase by a small amount with time as shown in Figure V-1.

The models discussed later are somewhat unrealistic because they assume that the required force obtained from the missile's thrust-vector control can be applied instantaneously at the proper time and reversed in a similar fashion to decelerate the missile. This, however, does provide optimum missile performance for comparison with any real system.

In all the simulations the missile is acceleration-limited with instantaneous application of constant side-force assumed. One could allow for a time delay in switching the missile's thrust-vector control from the positive to the negative direction. However, this would raise the order of the analysis. The computer program, though, can be modified to handle systems to the thirtieth order.

Optimum acceleration switching requires that a switching-boundary be established that zeros the error in the minimum time. That boundary was determined to be parabolic as drawn in the phase-plane representation of Figure V-6 and as supported

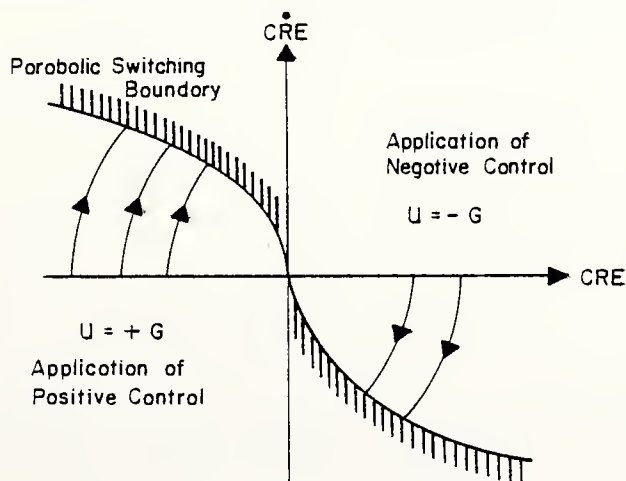


Figure V-6 Optimum Parabolic Switching Boundary



by the calculations which follow later. When the acceleration or deceleration trajectories attempt to cross this optimum curve (switching boundary) the system switches to that optimum path and the error is driven to zero along it.

In order to verify this result, a phase plane was produced by introducing, as initial conditions, large cross-range error and cross-range error rates into a computer-simulated guidance model containing acceleration-limiting methods within the tracker control function. The resulting phase plane, Figure V-7, substantiates that the tracker control function switches properly to the theoretically-optimum, parabolic curve for error reduction expected by calculation. This curve is represented in Figure V-6. Also, the target was intercepted by the missile as substantiated in Figures V-8 and V-9. Figure V-10 shows the

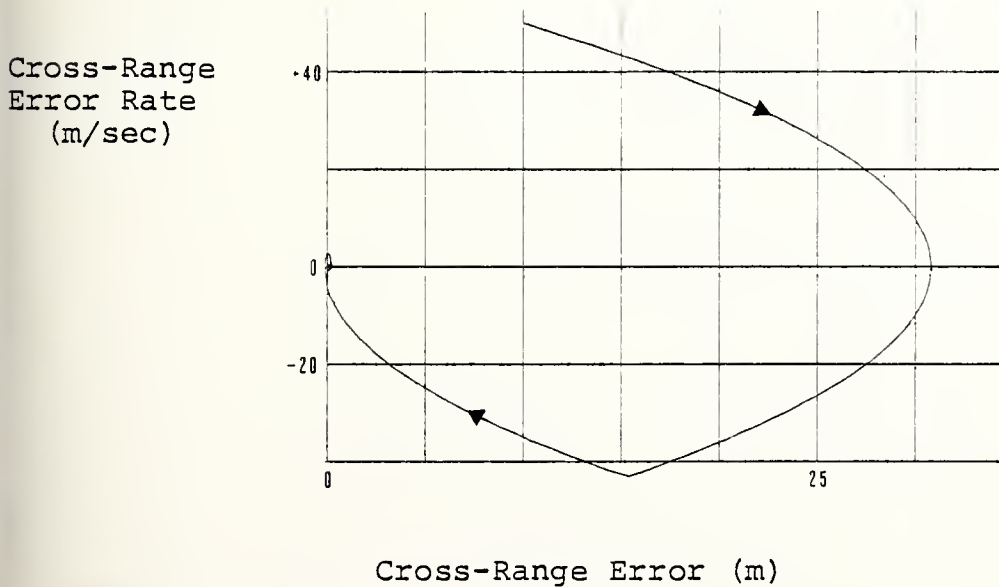


Figure V-7 Phase Plane When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate



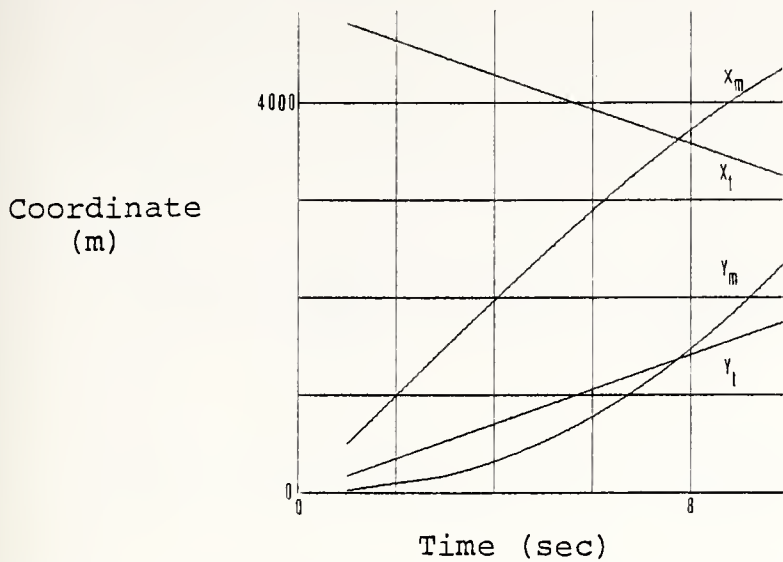


Figure V-8 Missile & Target Positional Coordinates When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate

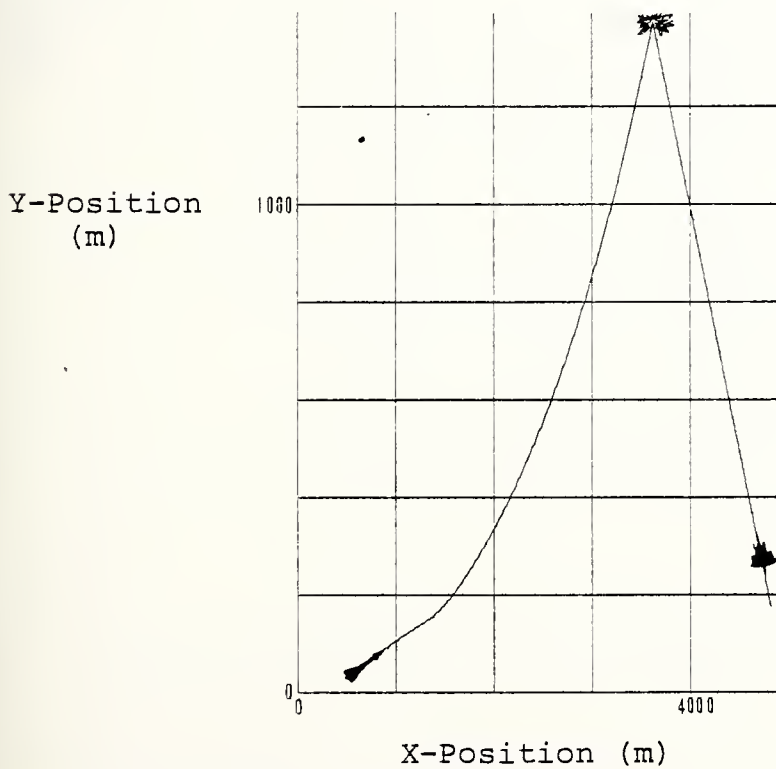


Figure V-9 Missile & Target Flight Paths When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate



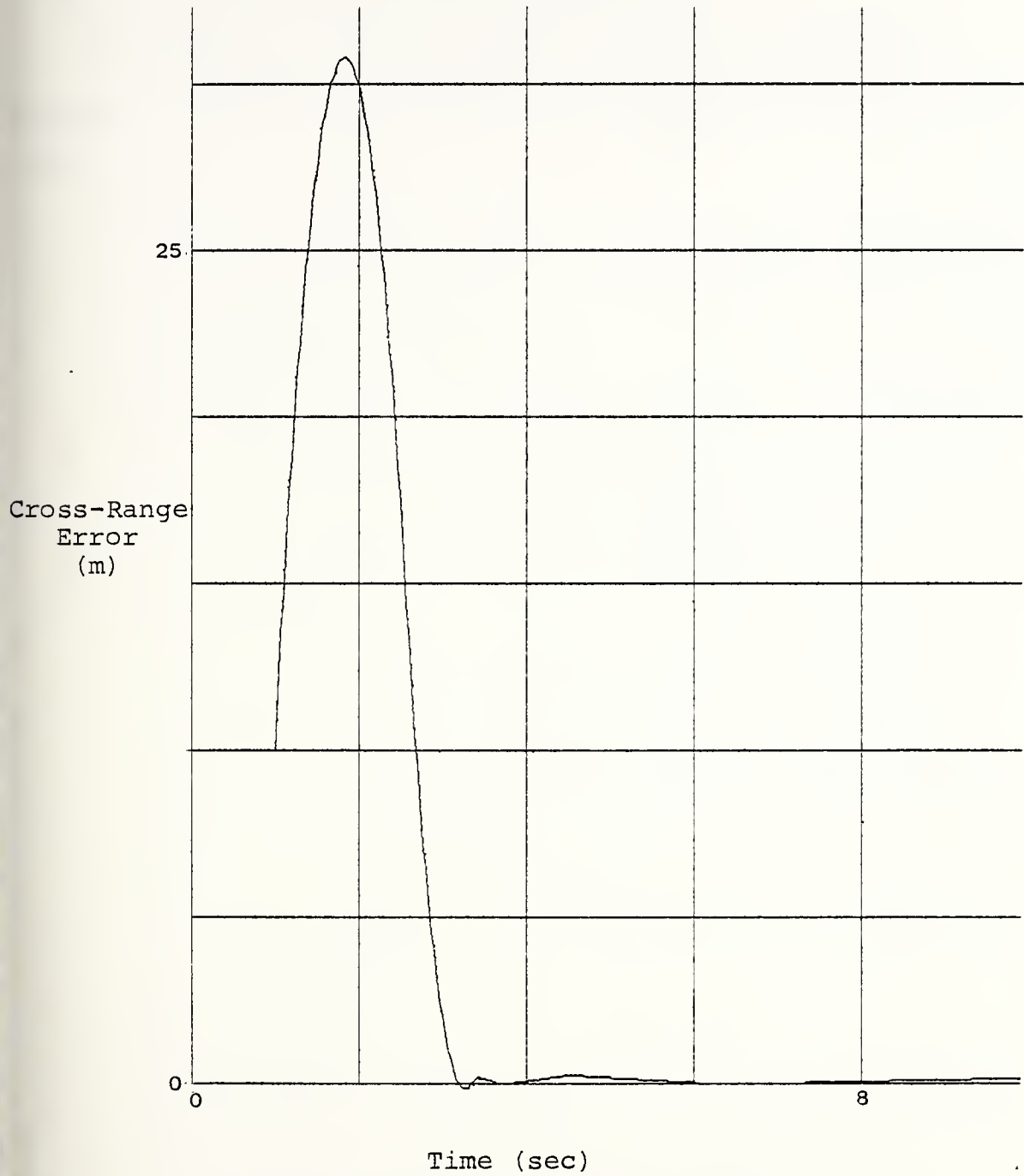


Figure V-10 Missile Cross-Range Error When Control Function Utilized With Initial Errors Introduced for Cross-Range Error and Error Rate





missile's cross-range error with time of flight. This error is not readily apparent in Figure V-9 because of the relatively large distance scale utilized in that figure. As can be seen this cross-range error is reduced quickly.

The theoretical calculations supporting the parabolic nature of the switching boundary represented by Figure V-6 follow:

$m$  = missile mass

$F$  = lateral force applied to missile

CRE = cross-range error

$\dot{CRE}$  = cross-range error rate

$\ddot{CRE}$  = cross-range error acceleration

$G$  = magnitude of cross-range missile acceleration

$U$  = cross-range missile acceleration

$C$  = constant

Since the mass of the missile and the magnitude of the output force from the missile thrust-vector control are assumed constant, the magnitude of the acceleration is constant, thus:

$$m \ddot{CRE} = |F|$$

$$|\ddot{CRE}| = |F|/m = G$$

$$\left| \frac{d}{dt} \dot{CRE} \right| = G$$

$$\int |d \dot{CRE}| = \int G dt \rightarrow |\dot{CRE}| = Gt + C$$

Let the initial conditions for time and cross-range error be zero. Therefore, the constant, "C",



is zero also.

$$\dot{CRE} = Gt, \quad \dot{CRE} \geq 0$$

$$t = \dot{CRE}/G \rightarrow t^2 = \dot{CRE}^2/G^2$$

$$\frac{d}{dt} CRE = Gt \quad \int d(CRE) = \int Gt dt$$

$$CRE = Gt^2/2 + |CRE(0)|, \quad t^2 = \dot{CRE}^2/G^2$$

$$CRE = \dot{CRE}^2/2G + |CRE(0)|, \quad CRE \geq 0$$

Since the direction of the missile's thrust-vector control force can be either in the positive or negative direction, the sign of the cross-range error (CRE) must be preserved. The initial missile cross-range error can be in either direction; consequently, the equation becomes:

$$CRE = \dot{CRE} |\dot{CRE}|/2G + CRE(0)$$

This equation is parabolic as depicted in Figure V-6.

Also, since the magnitude of the missile cross-range acceleration, "G", is constant, the corresponding acceleration vector, "U", is:

$$U = \pm G \quad \pm U = (-G) \text{SIGN}(\text{error function})$$

$$, \text{ error function} = EX = \dot{CRE} |\dot{CRE}|/2G + CRE$$

This drives the missile's control surfaces in the proper direction, a direction which is opposite to the sign of the error function, thus moving the missile back toward the line of sight with a constant magnitude of



acceleration, "G".

A simplified block diagram representing the command to line-of-sight guidance system is presented in Figure V-11.

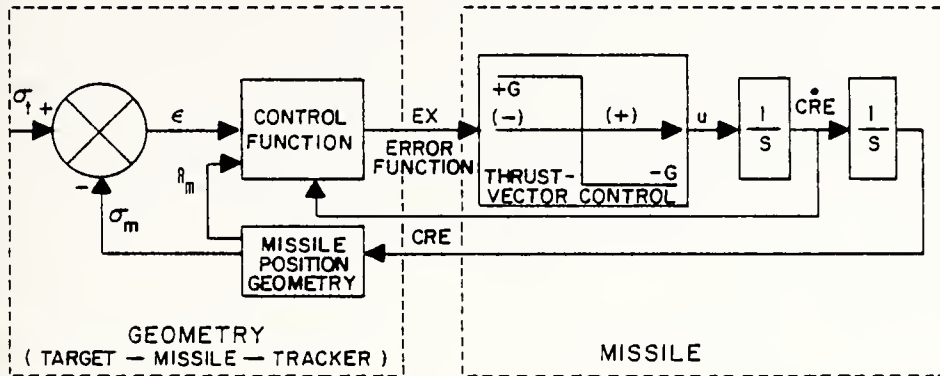


Figure V-11 Elementary Command to Line-of-Sight Block Diagram



## VI. 'BASIC' COMMAND TO LINE-OF-SIGHT SIMULATION

### A. PROGRAM DESCRIPTION

Utilizing the concepts previously discussed, this program simulates theoretically optimum surface-to-air missile maneuver against a constant-velocity target in two dimensions. The engagement was designed to occur within the first quadrant of a standard Cartesian-coordinate reference system with the ground tracker, missile launching unit, located at the origin. The missile was assumed to be "captured" by the guidance system approximately one second after missile launch or at a distance of about 500 meters from the ground tracker unit. At the time of missile "capture" an initial cross-range error of ten meters and a cross-range error rate of 50 meters per second were introduced to demonstrate the ability of the guidance system to neutralize initial errors. It should be noted that these values of error were selected entirely arbitrarily and that they could be any reasonable values assumed to exist at the time of missile "capture".

The target was "flown" across the first quadrant from a position 5000 meters from the tracker on the X-axis (5000,0) toward a position 5000 meters from the tracker on the Y-axis (0,5000).

In order to use a command to line-of-sight guidance scheme effectively, the missile must possess a speed advantage over the target. Therefore, in this model the missile velocity was chosen to be 500 meters per second versus a 250 meter-per-second target. At sea level these speeds equate to approximately





Mach 1.5 and Mach 0.75 respectively. The missile, also, after being launched directly toward the target, was maneuvered using on-off, thrust-vector control which applied a lateral acceleration of 60 meters per second squared to the missile. This is just greater than six times the earth's gravitational acceleration at sea level.

In general, keeping in mind the scenario above, the program operates as follows:

The equations for the target's movement were written and appropriately placed into the program so that the target and missile angular directions could be compared. The difference between these two angles creates an error angle. This error angle is geometrically converted by the tracker computer into the cross-range error, the distance of the missile from its intended flight path - the line of sight to the target in this case. That error distance is reduced to zero by applying a lateral, thrust-vector-controlled acceleration to the missile in the correct direction as determined and commanded by the ground tracker unit. The optimum parabolic switching boundary, discussed in Section V, was used in the tracker computations. In order to determine this cross-range error, the tracker estimates the missile's range by the elapsed time of flight and the missile's velocity profile. Also, to simplify the program somewhat, the



missile's velocity was assumed to be constant in a radial direction from the tracker. When the cross-range error is reduced to an acceptable distance, the amount of thrust-vector control is limited to prevent rapid changes between full thrust-vector deflections in both directions. If the error, again, becomes unacceptably large, the full deflection positions are utilized. Missile position is determined by state variable analysis using the INTEG2 subroutine for integration of the lateral thrust-vector-controlled missile acceleration. The use of this subroutine is explained in Appendix A. After two integrations, the missile's new cross-range error is determined and, then, this distance is used with estimated missile range to provide a revised missile angular position. This process is repeated in increments of five thousandth of a second from the time of missile "capture" by the tracker, a time of approximately one second, until maximum missile range is attained after ten seconds of flight. Thus, the missile is driven continually to a position upon the line of sight to the target. The program variables are "frozen" when the missile reaches the closest point of approach to the target if the value of constant C(1) is one. If this value is zero, the program computes until the time of missile flight reaches ten seconds.



The program was written in the FORTRAN IV language. A detailed, annotated copy of the program, including a table of program variable definitions, is provided following the narrative portion of this paper.

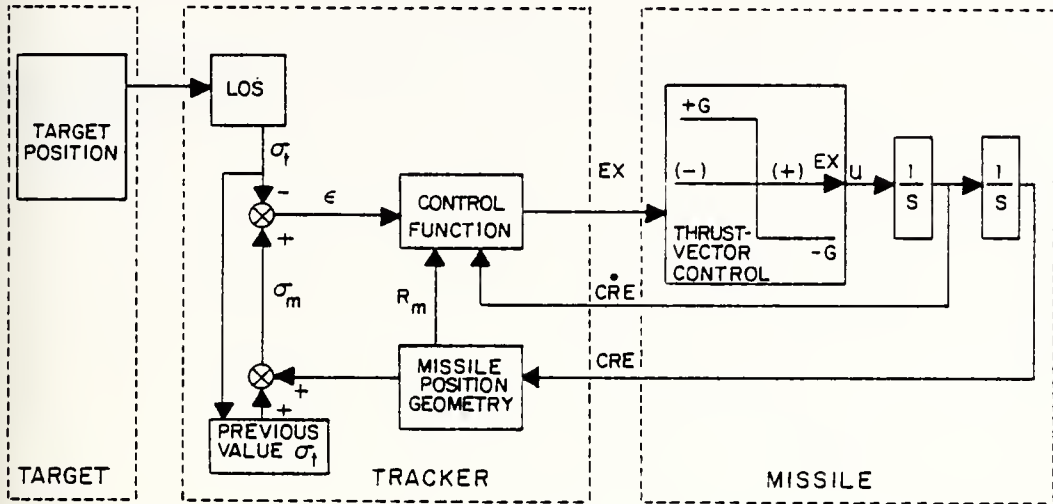


Figure VI-1 'Basic' Computer Simulation Block Diagram

## B. SYSTEM BLOCK DIAGRAM

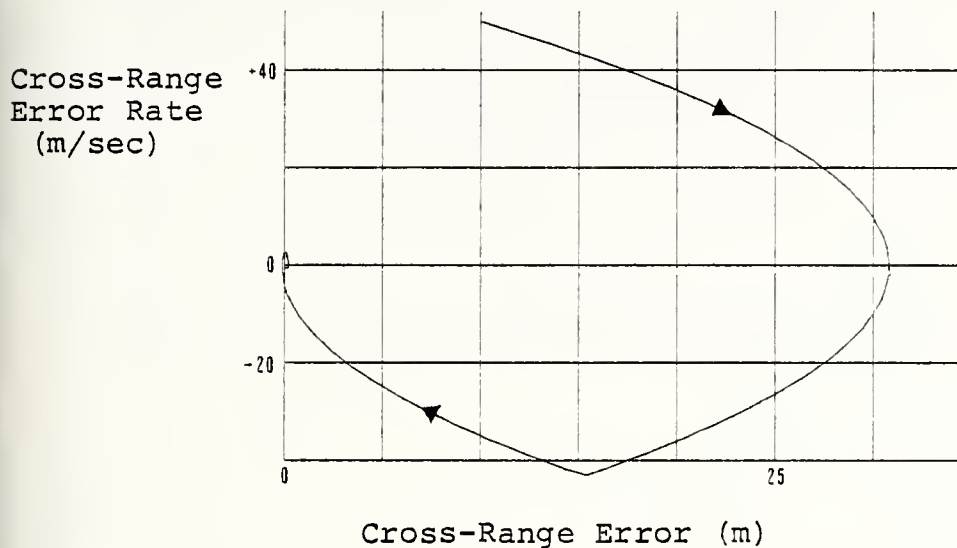
The command to line-of-sight computer simulation block diagram, Figure VI-1, shows how the elementary block diagram of this guidance technique, Figure V-11, was expanded upon in order to develop a computer model for the system. As will be seen, while appearing to be quite simple in concept, the simulation becomes increasingly more complicated as its sophistication is enhanced.

## C. SIMULATION RESULTS

The plotted output of the phase plane, Figure VI-2, shows how the cross-range error and cross-range error rate are driven to a near zero condition along the optimum, parabolic, switching



boundary. Once the cross-range error is less than one meter, the magnitude of the thrust vector control is reduced by a sixth to minimize the possibility of over-controlling the missile.



. Figure VI-2 'Basic' Simulation Phase Plane

Figure VI-3 presents the "X" and "Y" positions for the missile and the target versus time. As can be seen the intercept occurs in just under eight seconds after missile launch.

Figure VI-4 indicates the relative positions of the missile and target with the terminal points for each flight profile stopped at the closest point of approach. An intercept did occur.

In order to ensure that the missile's cross-range error from the line of sight was damped, Figure VI-5 was produced to show this cross-range error versus time of flight. The initial errors, applied at the time of missile "capture" by the





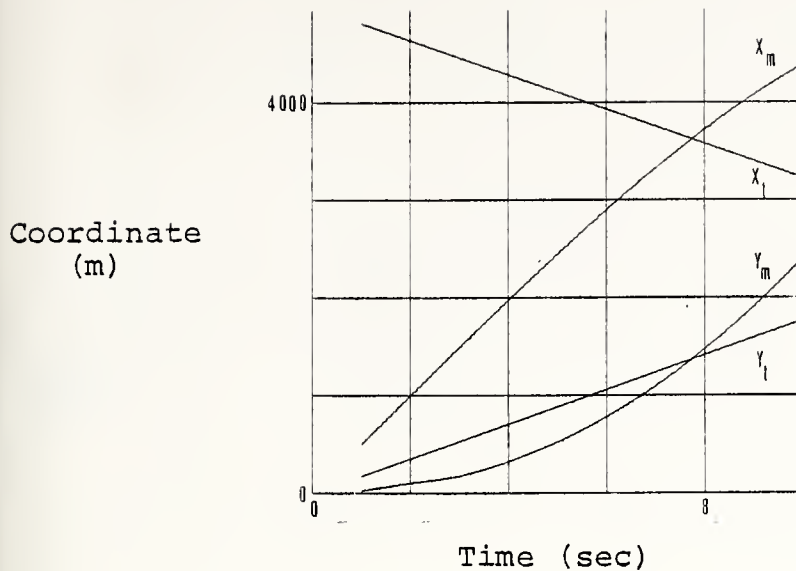


Figure VI-3 'Basic' Simulation Missile & Target Positional Coordinates

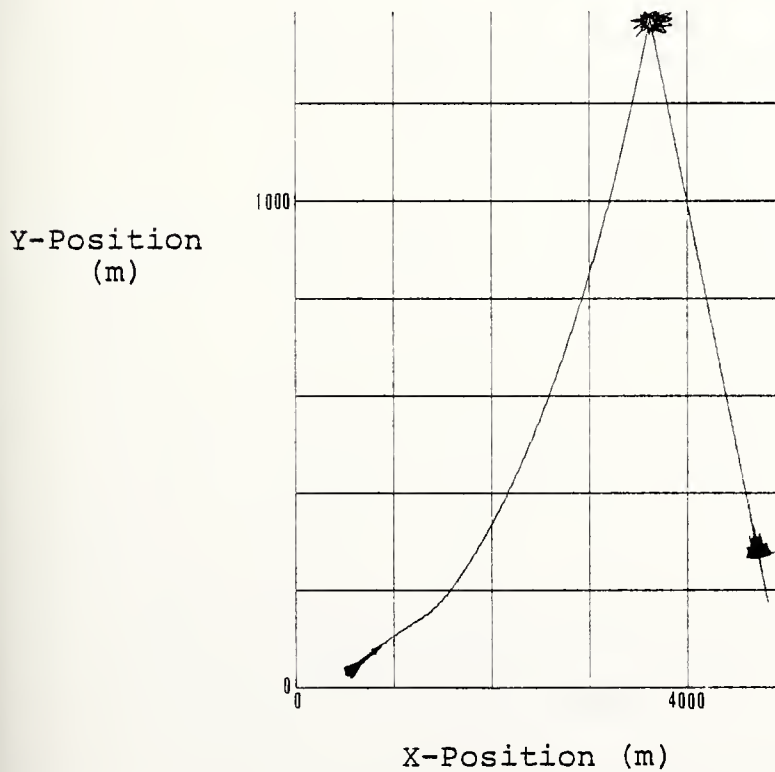


Figure VI-4 'Basic' Simulation Missile & Target Flight Paths



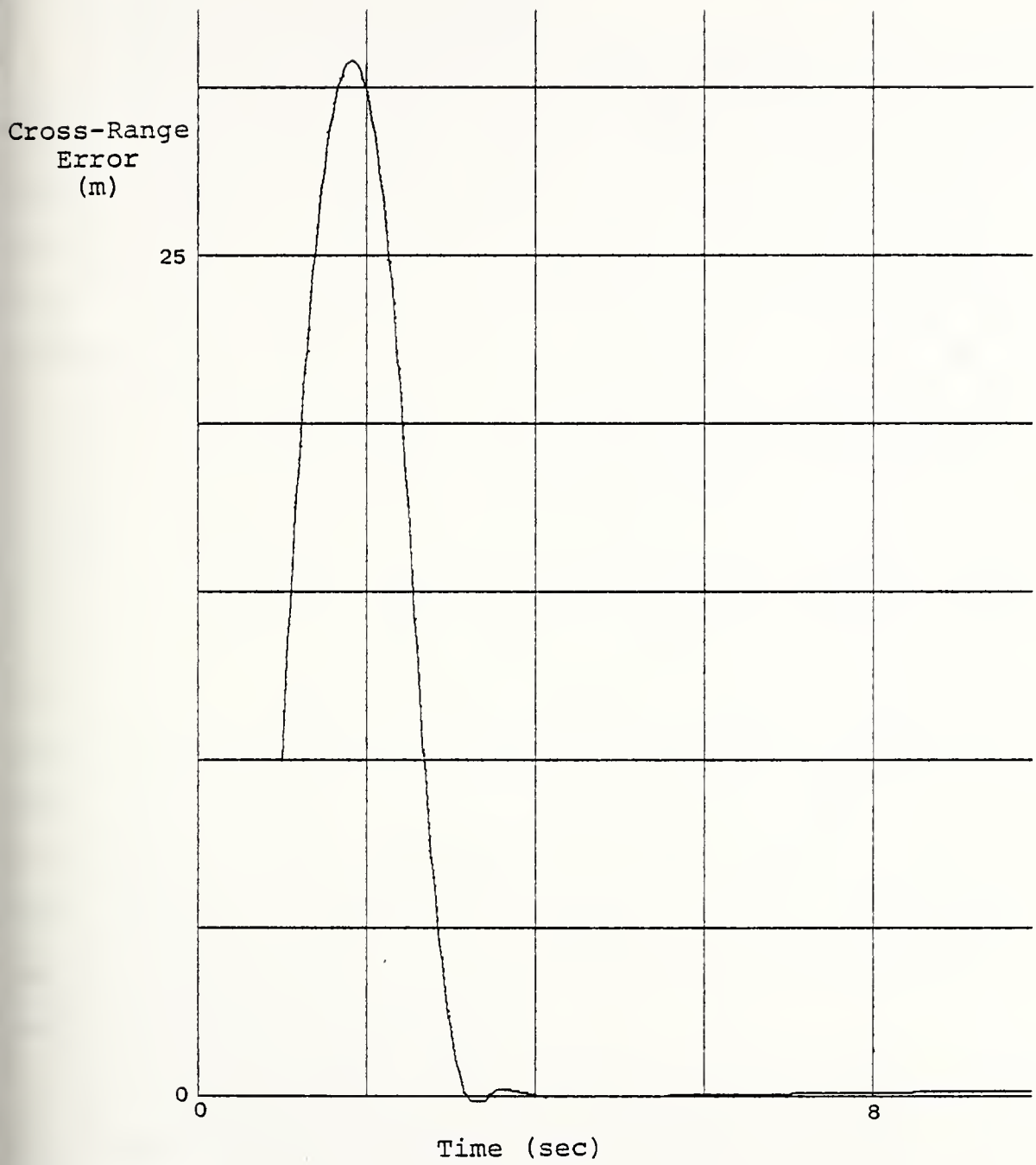


Figure VI-5 'Basic' Simulation Missile Cross-Range Error



guidance system (corresponding to a time of approximately one second in the figure), were reduced sufficiently in less than two seconds.

#### D. EFFECTS OF DECEPTIVE JAMMING

A readily apparent deceptive jamming technique, which might be effective against this class of guidance system, would be to employ, aboard the target aircraft, an emitter which imitates the signal that is used by the tracker to locate the missile's angle from the reference plane. This could be, for example, depending upon how the missile is tracked in angle, an imitation of the missile's transponder radio-frequency signal or a duplication of the missile's infrared emissions. In any case the deceptive jamming signal from the target aircraft should be of sufficient strength to ensure that the missile tracking system within the ground-located tracker considers the target to be the missile. There are, most assuredly, counter-countermeasures which could be applied to negate this deceptive technique; however, this, currently, would be one method of generating a zero error signal between the missile and target angles. Additionally, this deceptive countermeasure provides a method to demonstrate how the computer simulation drives the missile with inputs of erroneous guidance information.

In order to implement this deceptive jamming technique, it was assumed that the target aircraft jamming equipment became effective when the missile was approximately half the distance to the target. This would equate to about four seconds of missile flight time. Therefore, since this jamming



countermeasure system simply zeros the error angle, only one additional equation was necessary to modify the previously discussed "basic" command to line-of-sight guidance model. The equation,  $IF(T.GE.4.0)E=0.$ , was placed into the program following the computational step for the calculation of the angular error,  $E=SIGM-SIGT.$

The resulting missile trajectory and other parameters are readily apparent in Figure VI-6 through VI-8. As shown in Figure VI-6, the error was reduced to zero by the time that jamming commenced. After deceptive jamming was initiated, the cross-range error rapidly increased causing a miss of over 300 meters. Needless to say, because of the short times of flight involved, the countermeasures must be employed in a timely manner. Otherwise, the missile could approach, easily, within lethal range of the target aircraft.

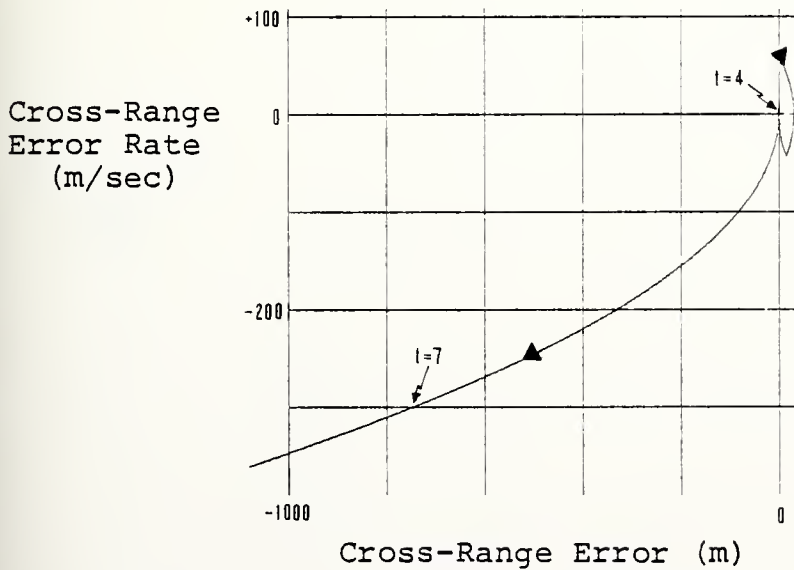


Figure VI-6 'Basic' Simulation Phase Plane When System Deceptively Jammed





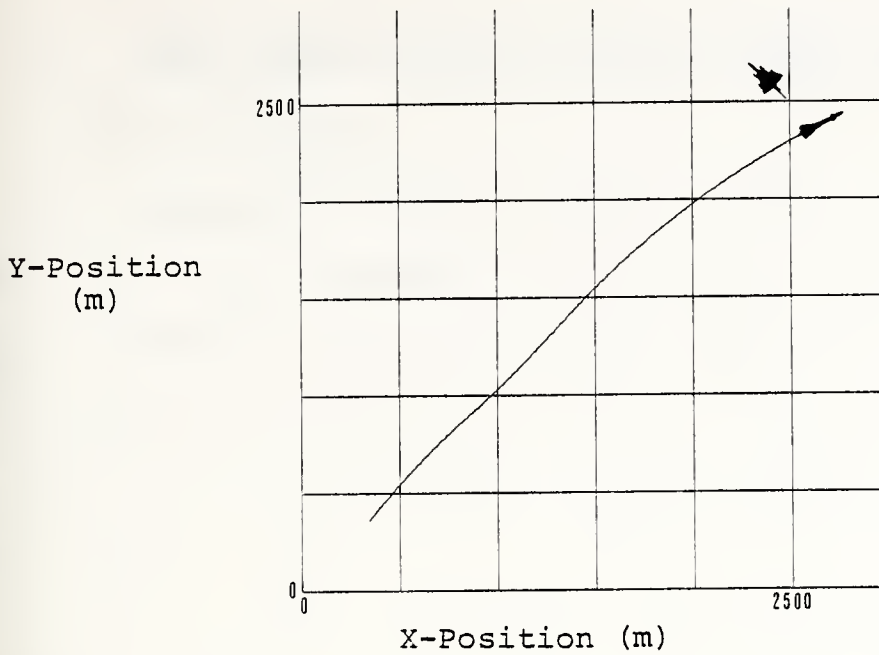


Figure VI-7 'Basic' Simulation Missile & Target Flight Paths When System Deceptively Jammed

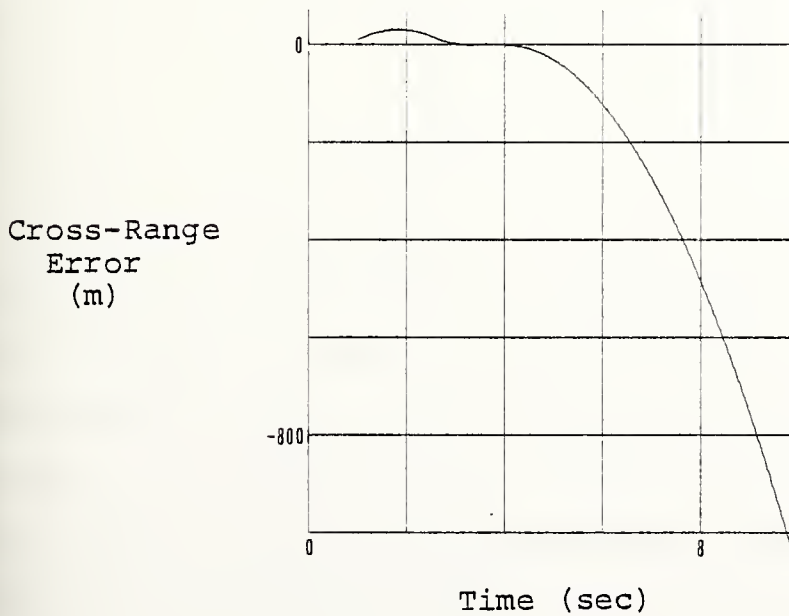


Figure VI-8 'Basic' Simulation Missile Cross-Range Error When System Deceptively Jammed



## VII. 'LEAD-ANGLE' COMMAND TO LINE-OF-SIGHT SIMULATION

### A. PROGRAM DESCRIPTION

This 'lead-angle' simulation uses the same target profile as that of the 'basic' command to line-of-sight simulation. This allows a comparison of the two simulations to a certain degree. One parameter, which can not be compared accurately, is the time to impact. This occurs because the radial missile velocity from the tracker was assumed to be a constant value - in this case 500 meters per second. Therefore, intercepts will take place at about the same time regardless of missile maneuvers. A general comparison can be made, however, by mentally comparing the plotted missile flight profiles, the longer profile taking the most time in an actual engagement. Also, in order to avoid redundancy, discussion will be limited to modifications of the previously described computer model. Only a few changes were required to produce a 'lead-angle' variant for the command to line-of-sight guidance system.

In this guidance scheme, the spacial positions of both the target and the missile must be accurately and continuously known in order to compute an advanced impact position. This is accomplished by calculating the distance from the missile to the target and their closing velocity. From this data, a time remaining until impact can be estimated. This remaining time of flight, then, is combined with the target's current velocity vector and position to ascertain the expected impact point. A synthetic line of sight toward this location is generated by the ground tracker unit computer and the missile is commanded



to "fly" up this flight path instead of along the line of sight directly toward the target. The guidance scheme used to maintain the missile on course, the synthetic line of sight, is, otherwise, identical to that of the "basic" simulation previously described. Once the missile passes the closest point of approach to the target, this simulation ceases to be valid. This is quite all right because, once the missile has passed the target, no maneuver would be sufficient to effectively engage the aircraft.

### B. SYSTEM BLOCK DIAGRAM

The "lead-angle" command to line of sight computer simulation block diagram is contained in Figure VII-1. It is essentially a modification of the "basic" simulation block diagram, Figure VI-1. As described previously, target and missile positions plus their velocities must be known for the guidance scheme to be viable.

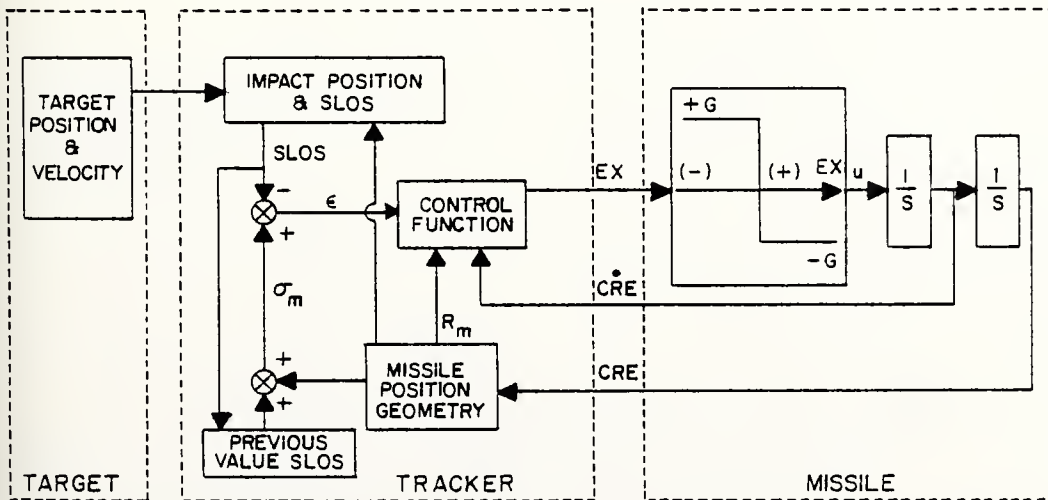


Figure VII-1 'Lead-Angle' Computer Simulation Block Diagram



### C. SIMULATION RESULTS

This simulation supports the idea that the use of a "lead-angle" variant to the "basic" command to line-of-sight guidance system enhances the effectiveness of the elementary guidance technique. As can be seen in Figures VII-2 through VII-5, the missile was "captured" by the guidance system at a time of approximately one second from launch. The initial cross-range and cross-range-rate errors were quickly reduced and the missile "flew" out the synthetic line-of-sight to successfully intercept the target aircraft. Of course, if the target employs electronic countermeasures such as denial jamming of the target tracking equipment, this "lead-angle" method would prove to be useless because accurate positioning information must be maintained to compute the estimated impact point. Also, if the target maneuvers sufficiently to cause the impact point to vary at high rates, the synthetic line of sight may shift at too high

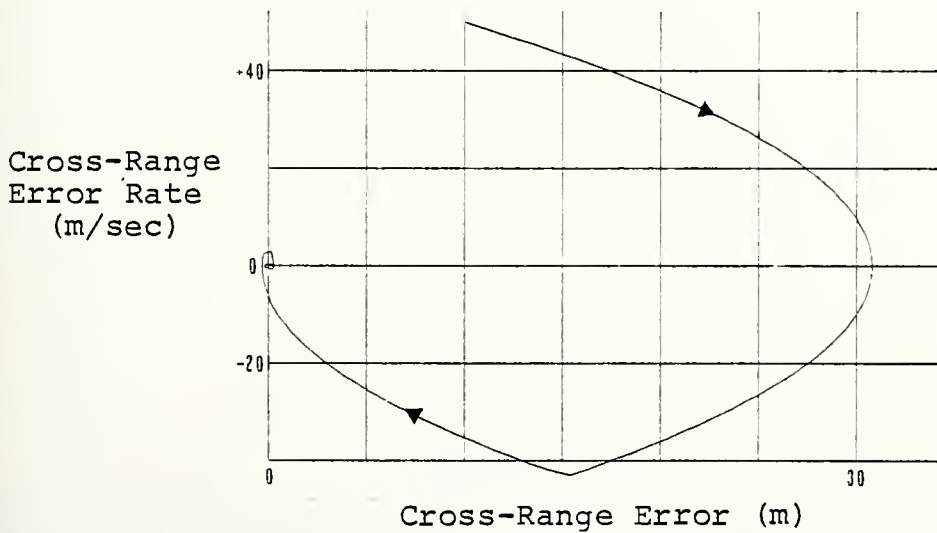


Figure VII-2 'Lead-Angle' Simulation Phase Plane





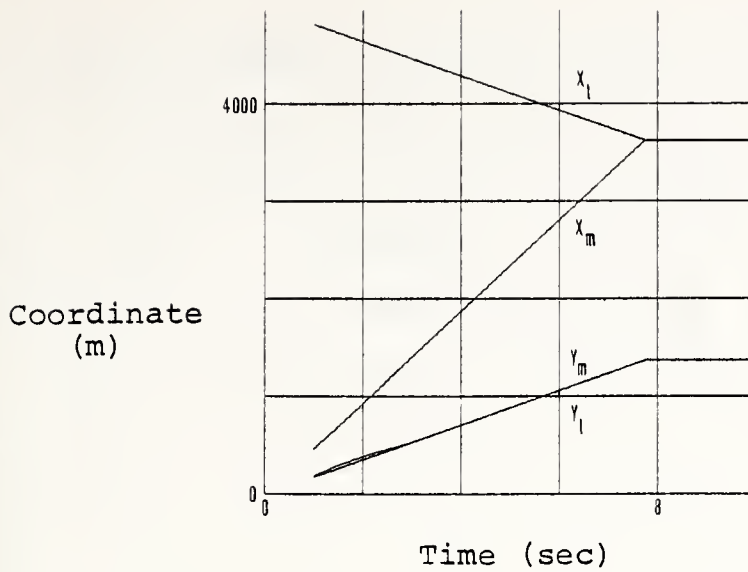


Figure VII-3 'Lead-Angle' Simulation Missile & Target Positional Coordinates

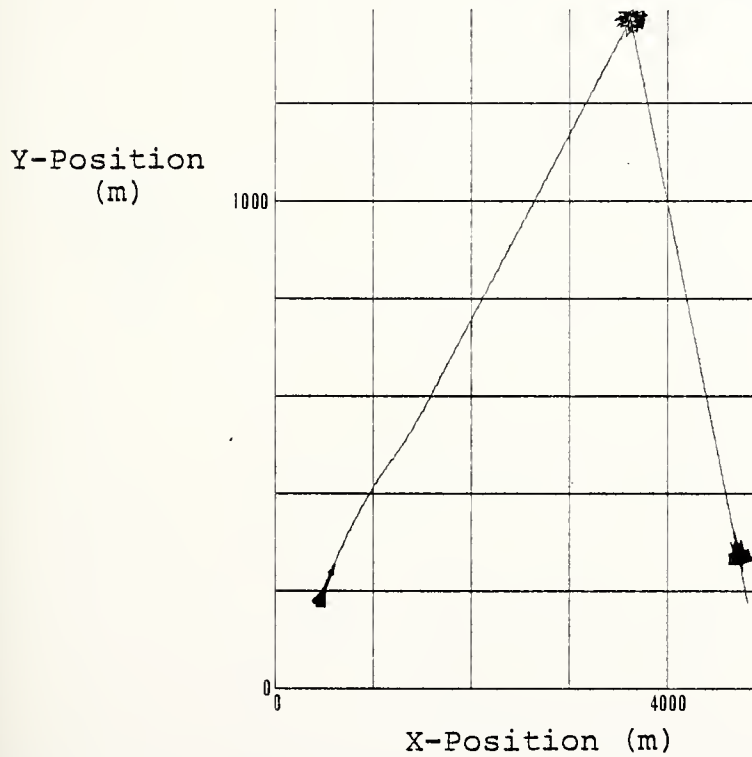


Figure VII-4 'Lead-Angle' Simulation Missile & Target Flight Paths



a rate for the missile to follow. Thus, the missile may not be able to physically engage the impact point but might be able to still effectively attack the target by the "basic" guidance scheme. Consequently, in order to capitalize upon the advantage of the "lead-angle" variant (an increased effective missile envelope against a relatively slow maneuvering target which is not employing electronic countermeasures), it seemed that a combined guidance technique should be considered. Therefore, the "consolidated" simulation discussed in the next section was contrived.

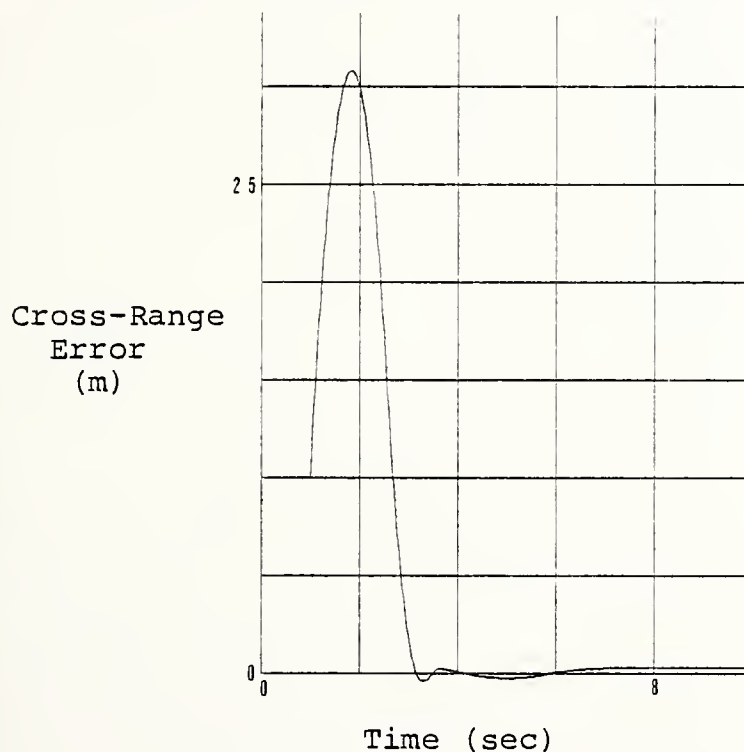


Figure VII-5 'Lead-Angle' Simulation Missile Cross-Range Error



## VIII. 'CONSOLIDATED' COMMAND TO LINE-OF-SIGHT SIMULATION

### A. PROGRAM DESCRIPTION

When the two previous simulations were combined, a major problem surfaced. That problem was basing the missile's new position upon two different references, the line of sight to the target in the case of the "basic" model and the synthetic line of sight to the impact point in the case of the "lead-angle" variant. This incompatibility caused the missile's position to jump radically as the "consolidated" simulation attempted to shift alternately between the two reference lines as the jamming situation varied. Therefore, a single reference had to be implemented for continuous use regardless of whether jamming was or was not encountered. The X-axis of the Cartesian-coordinate system was selected since various angular measurements were based upon it already. Also, this has proved to provide a more accurate model of the missile's optimum flight profile because the reference does not move during the time of missile flight. If a strictly "basic" simulation should be desired, jamming is considered to be effective during the entire engagement. Conversely, the absence of jamming during the missile's flight produces a total "lead-angle" intercept of the target aircraft. Of course, different duty cycles for jamming can be implemented by the target during a single computer run to test the ability of the missile system to transition between the two attack modes. A limited number of different jamming combinations were tested and are discussed later.



In order to implement jamming, the jamming time increments are read into the simulation through five time intervals specified on the first five data cards. These cards precede those required by the INTEG2 subroutine as explained in Appendix A. The time each jamming interval begins is punched into columns 1-10 of each card in F-format. The times that jamming ceases are entered in columns 11-20. Should a no-jamming condition be desired, then, all five cards are left blank but they must not be omitted. Total jamming can be obtained by making one card cover the entire time of flight. For example, the first card could have columns 1-10 blank and the value 11.0 entered into columns 11-20. This produces jamming from time zero until a time of eleven seconds, which is well after the end of the engagement. The remaining four cards are blank and must be inserted.

This simulation "drives" the missile in the "Y" direction with sufficient "Y" velocity to reduce the cross-range error to zero. The missile's X-direction velocity component is used to maintain a constant, outward, radial velocity, " $V_m$ ", toward the target. In this simulation, as in its predecessors, that radial velocity, " $V_m$ ", is assumed to be 500 meters per second. Therefore, the velocity of the missile in the "X" direction is related to that obtained from the forcing function in the "Y" direction. This is shown in Figure VIII-1.

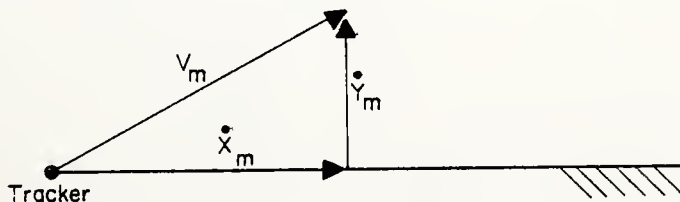


Figure VIII-1 'Consolidated' Simulation Missile Velocity Vectors





The missile's Y-direction velocity is obtained from the state variable analysis through a single integration of the forcing function, "U", which is the missile's Y-direction acceleration. This acceleration is the "Y" component of the cross-range error acceleration achieved by thrust-vector control of the missile as shown in Figure VIII-2.

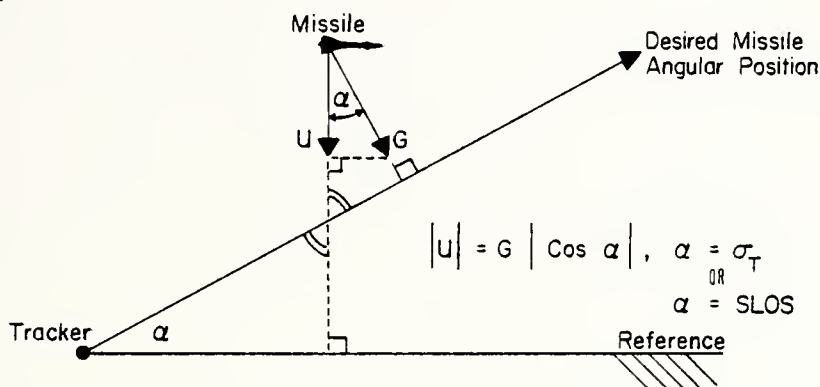


Figure VIII-2 'Consolidated' Simulation Missile Acceleration Vectors

The missile's Y-coordinate position, then, is the second integration of the forcing function. From these results, the X-direction components were determined since the velocities must yield a constant, radial velocity of 500 meters per second. In order to ensure a correct integration process, the missile's "Y" velocity was not limited to 500 meters per second; however, it never exceeded that value in any of the computer runs tested. It should be noted that the program will zero the acceleration and the velocity of the missile in the "X" direction if excessive missile "Y" velocity is experienced. Also, the results, then, will become invalid; consequently, the velocity of the missile,



" $v_m$ ", should be increased or the target's speed reduced to obtain a theoretically-possible engagement scenario.

The targets, in the previous simulations, "flew" a constant-velocity, linear course. In this simulation the target was maneuvered sinusoidally in the "Y" direction for the purpose of observing the effects of a moving impact point. The maximum velocity of the target in the "Y" direction is 64 meters per second with a half period of four seconds. The target's "X" velocity was a constant, negative 210 meters per second. This did produce some interesting results. First, this type of target maneuver caused the impact position to move at even higher rates than the target moved. Thus, in some cases, the missile may not be able to engage, adequately, the impact point which is moving at a very high velocity. It, however, may be able to attack the target itself by utilizing the "basic" command to line-of-sight approach. To determine which situation exists necessitated the use of the impact point's velocity to compute the synthetic line of sight, the angle upon which the missile "flies" in a non-jamming environment. If this impact-point velocity exceeded eight-tenths of the missile's velocity, then, the missile was commanded to "fly" directly toward the target's present position by the "basic" guidance method. This was in lieu of utilizing the "lead-angle" approach with only an unstable impact position and, thus, an erratic or too quickly moving synthetic line of sight.

In order to provide a longer time of flight, the target's initial position was moved to coordinates (6000,1000) and the



target was "flown" inbound at 210 meters per second while sinusoidally maneuvering about the 1000 meter Y-ordinate. Also, in regard to the short time of flight involved, excessively large initial errors in either distance or rate should be avoided because of the time required to neutralize these errors. For example, since the missile's maximum acceleration is only 150 meters per second squared, initial erroneous conditions of 250 meters and 250 meters per second for the missile's Y-direction components require many seconds to correct. However, with initial conditions of 150 meters and 150 meters per second, the simulation proved the missile to be effective against this target over a full scope of jamming duty cycle variations - from none to continuous denial jamming.

The initial angle to the missile at the time of "capture" by the guidance system, a time of one second or 500 meters from the launcher-tracking unit, is determined by the initial conditions, X(1) and X(2). As can be recalled, X(1) is the missile's "Y" position at the time that the weapon is 500 meters radially from the launcher. Therefore, the angle is the inverse sine of X(1) divided by 500 meters. In order to attain this "Y" position in one second, the missile's "Y" velocity, then, was approximately the "Y" distance per second. Consequently, in this simulation, initial conditions of 100 meters for X(1) and 100 meters per second for X(2) equate to an angle of 11.54 degrees above reference.

Also, the magnitude of the missile's maximum lateral acceleration, "G", was increased to 150 meters per second



squared. This provided a sufficient forcing function to allow engagement of the target. This was possible even with increased initial errors at the time of missile "capture" by the guidance system; with a maneuvering target accelerating at just over 50 meters per second squared, at maximum, in the "Y" direction; and with various jamming duty cycles from zero to one. As in the previous simulations, the thrust-vector control of the missile was reduced by one-sixth; however, this was done only when the sum of the absolute values of cross-range error and error rate was less than or equal to four vice the value of one, which was used previously. The error tolerance was increased to provide a larger "idling" area for the missile's thrust-vector control because of the increased forcing function utilized.

The actual computer program for the simulation, with comments included, follows the appendices.

## B. SYSTEM BLOCK DIAGRAM

The block diagram of the "consolidated" simulation is contained in Figure VIII-3. Although it appears formidable at first glance, the diagram merely combines the concepts of the two previous simulations with slight modifications to account for the use of a new reference from which to compute the missile's location. If the tracking unit ascertains that it is contending with either denial jamming or a highly-maneuvering target, then, only the upper portion of the tracker diagram is used for guidance. Conversely, if both a non-jamming condition and a relatively slowly-maneuvering target situation are experienced, the lower portion of the tracker diagram produces





guidance for the missile.

The simulation does produce necessary calculations in all areas of the tracker diagram continually throughout the engagement, but only generates commands to the missile from the appropriate section of the model.

After the program reads the data cards, the times of jamming are known and the appropriate control function within the tracker's computer is implemented to control the missile. The magnitude of the impact point's velocity is considered, also, to ascertain the stability of the synthetic line of sight and, thus, to determine whether target maneuvers make the "lead-angle" approach of guidance inappropriate.

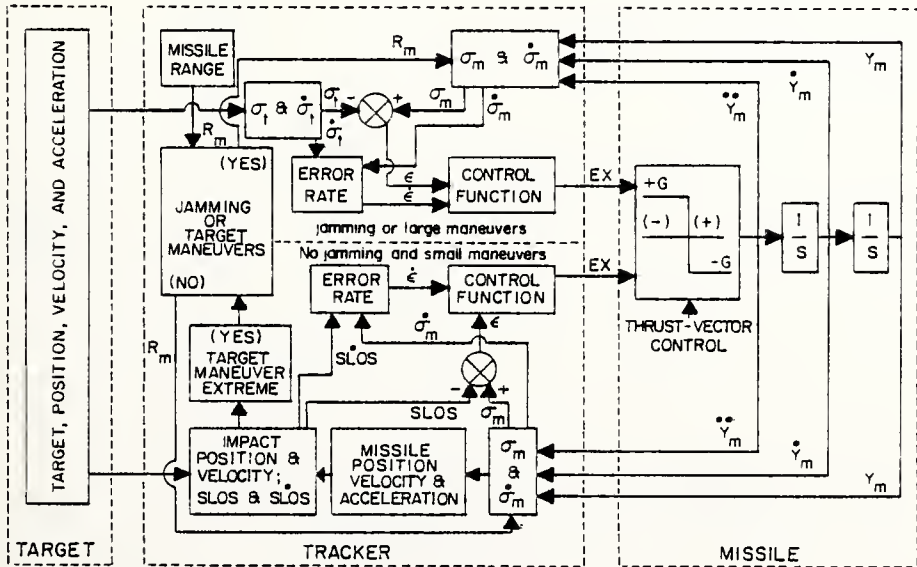


Figure VIII-3 'Consolidated' Computer Simulation Block Diagram



### C. SIMULATION RESULTS

Three engagements were computed under different jamming conditions. They were continuous jamming, intermittent jamming having an approximate duty cycle of one-half, and no jamming. During intermittent jamming, the ground-located tracker unit was denied range information within the following time increments after missile launch: 1.0-3.0, 4.0-5.0, 6.6-7.0, 7.4-7.8, and 8.2-8.6 seconds.

Utilizing the scenario which was previously discussed, all three attacks resulted in successful intercepts of the target. As can be observed in Figure VIII-4, the best flight profile of the three simulated runs occurred when total jamming was experienced. This jamming caused the missile to "fly" directly toward the target.

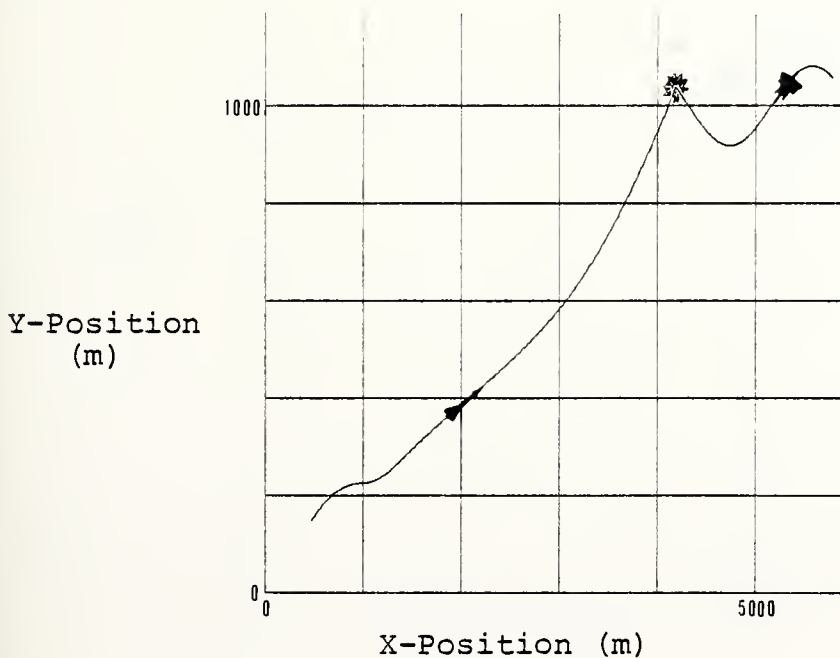


Figure VIII-4 'Consolidated' Simulation Missile & Target Flight Paths When Continuously Jammed



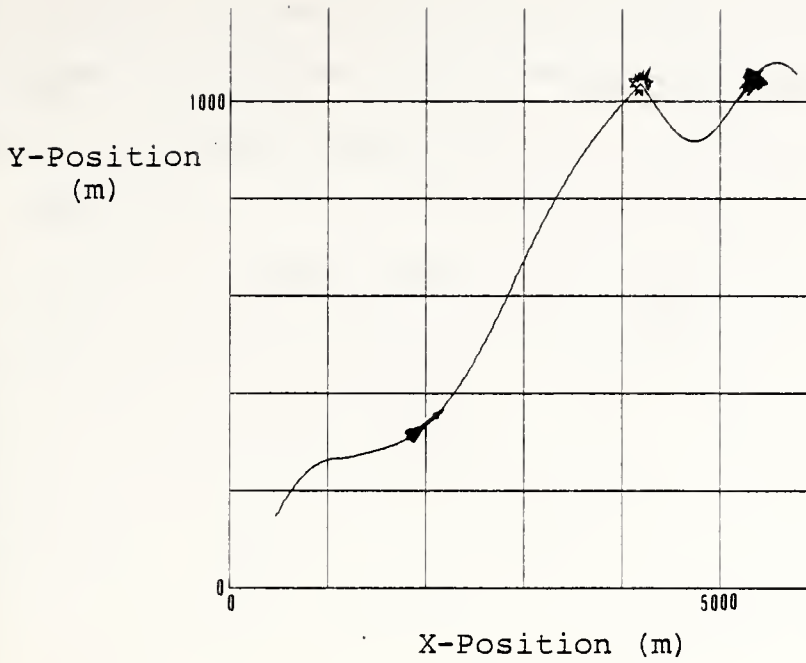


Figure VIII-5 'Consolidated' Simulation Missile & Target Flight Paths When Not Jammed

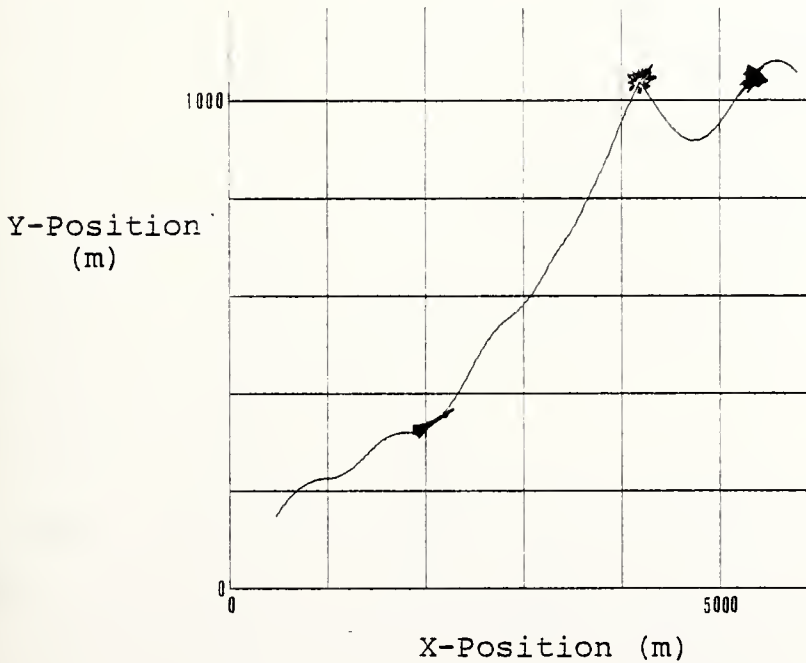


Figure VIII-6 'Consolidated' Simulation Missile & Target Flight Paths When Intermittently Jammed



When no jamming was encountered, the flight profile became more exaggerated because the target was maneuvering, thus causing the estimated impact point to move at an even higher rate. Recall that, in a non-jamming environment, the missile will "fly" for the impact point vice the target. Figure VIII-5 shows this profile. Finally, with intermittent jamming, the missile's flight profile, Figure VIII-6, became more erratic but did not reach the more extreme deviations experienced in the unjammed case. Therefore, it appears that the best method would be to use the "basic" command to line-of-sight guidance technique against a maneuvering target regardless of the jamming condition in effect. Also, as can be observed in Figure VIII-6, the flight profile during intermittent jamming, the amount of erratic variation in the missile's movement decreased considerably as it closed the target. This occurred because the difference between the synthetic line of sight and the line of sight becomes smaller with time until they are equal in value at the time of impact. Consequently, the "lead-angle" approach should be considered for utilization against a non-maneuvering target when adequate range information is available, i.e., when denial jamming is not encountered.

In order to reduce the amount of movement in the missile's thrust-vector controllers and, thus, to ensure that they do not "beat themselves to death" mechanically while trying to reduce a trivial cross-range error, a reduced amount of deflection was applied when the cross-range error was relatively small. This, then, was a trade-off of allowing some increase





in cross-range error for improved thrust-vector control mechanical reliability. Figure VIII-7 shows how the cross-range error was reduced without using a smaller amount of thrust-vector control deflection. Keep in mind that the control surfaces in the missile were shifting rapidly from a total positive to a total negative direction. In Figure VIII-8, a cross-range error of four meters was accepted as a reasonable maximum miss distance and the amount of control was decreased when the error was within that value. This assured improved mechanical reliability of the system. As can be seen in Figure VIII-8,

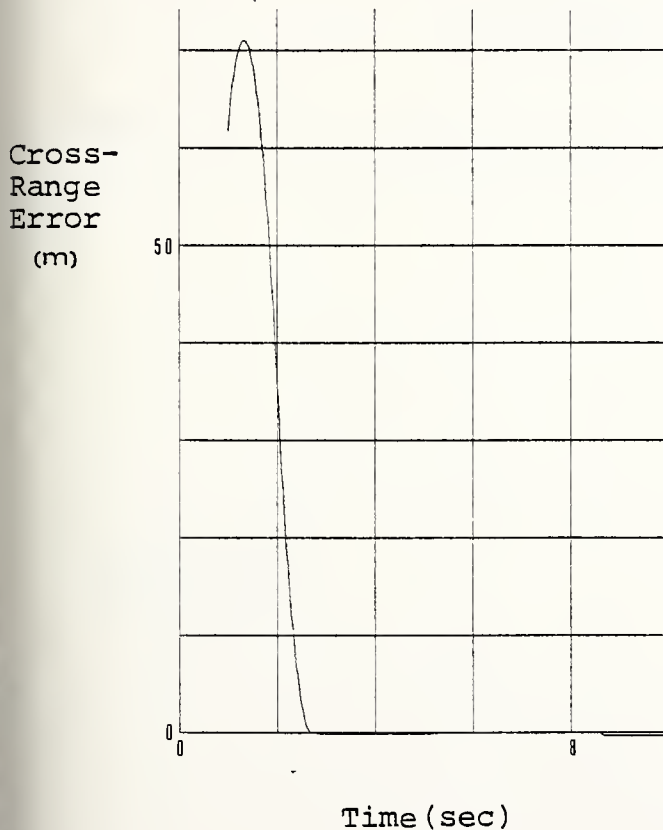


Figure VIII-7

'Consolidated' Simulation  
Missile Cross-Range Error  
With Full Missile Control  
Surface Deflections

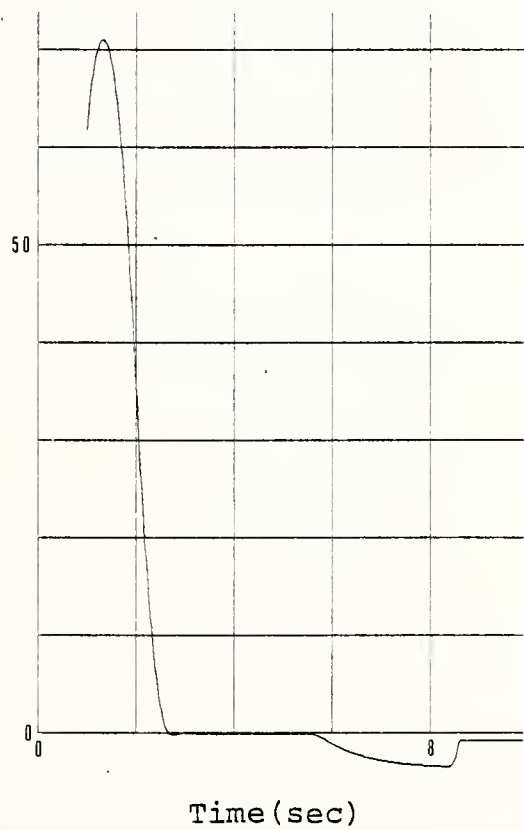


Figure VIII-8

'Consolidated' Simulation  
Missile Cross-Range Error  
With Reductions in Missile  
Control Surface Deflections



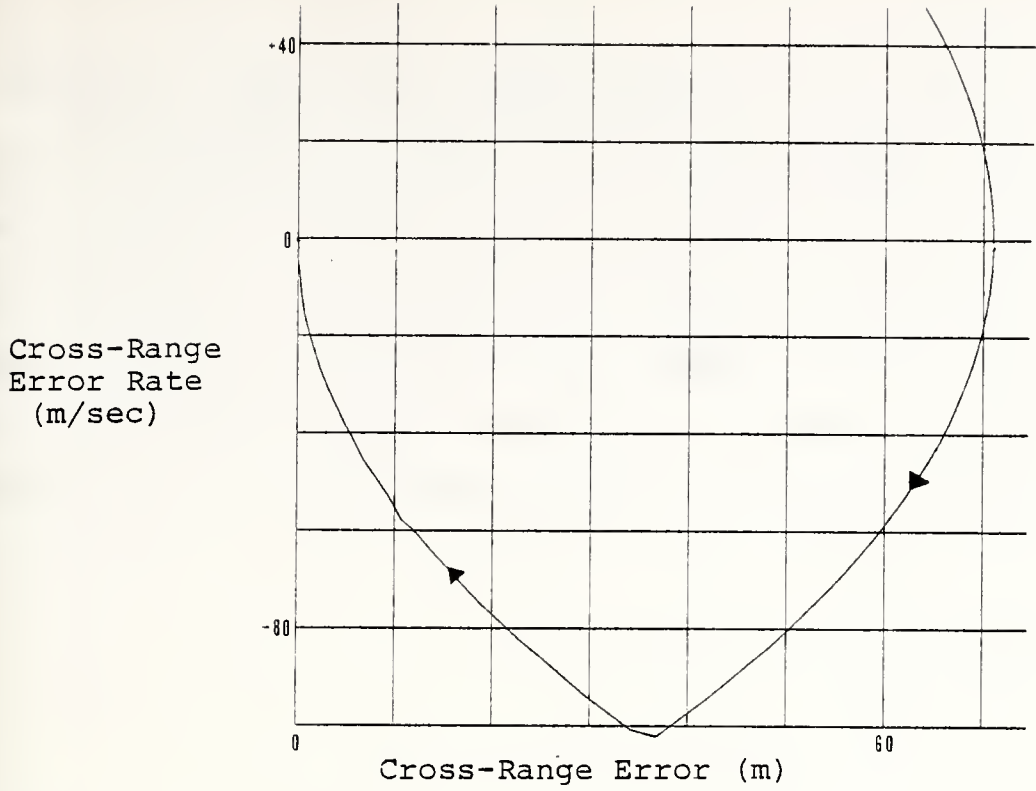


Figure VIII-9 'Consolidated' Simulation Phase Plane With Full Missile Control Surface Deflection

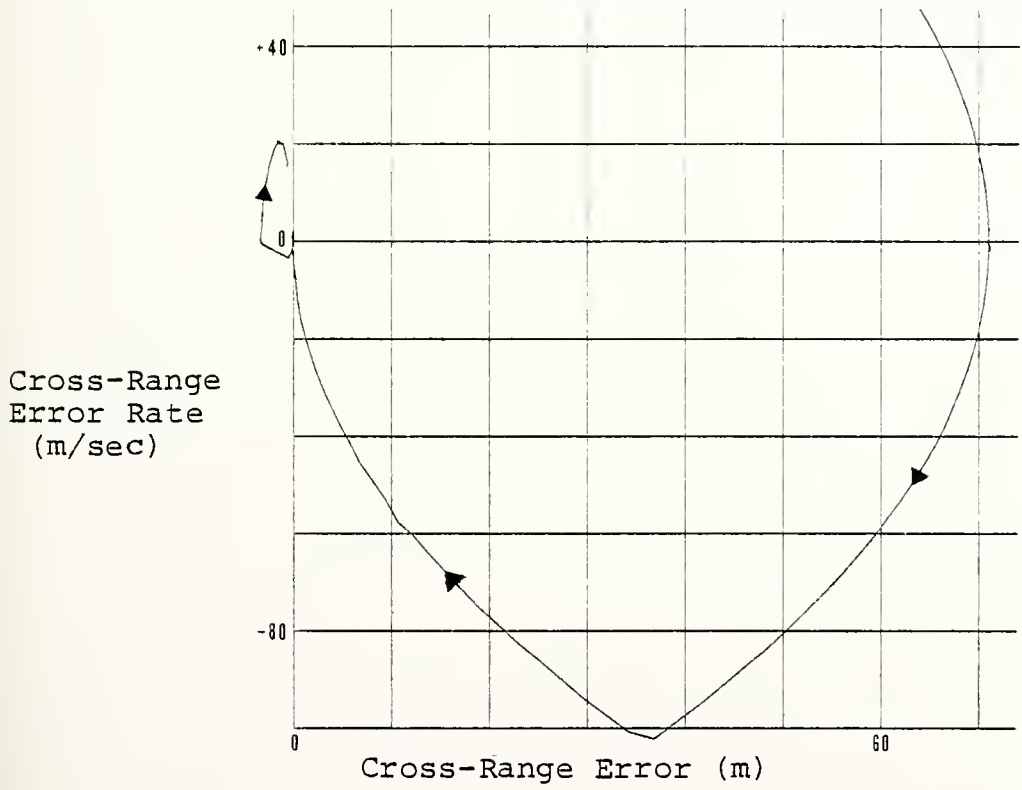


Figure VIII-10 'Consolidated' Simulation Phase Plane With Reductions in Missile Control Surface Deflections



at the time of impact the cross-range error was in the process of being reduced from a value of about four meters. The phase planes, Figures VIII-9 and VIII-10, also, show the effects of this trade-off. Figure VIII-9 was the result obtained when using no reduction in the amount of thrust-vector control; while Figure VIII-10 shows the effect of allowing a maximum four-meter error with a reduced amount of lateral thrust.



## IX. CONCLUSIONS

These simulations support the premise that consideration should be given to utilizing a "lead-angle" variant in conjunction with the "basic" command to line-of-sight guidance technique. This would increase the missile system's effective engagement envelope against a non-maneuvering target that employs low-duty-cycle denial jamming. If the target utilizes higher duty cycle jamming or begins to make evasive maneuvers, then, the system can revert, when necessary, to the "basic" guidance mode even after the missile has been launched.

This new approach appears to be generically applicable to missiles of the command to line-of-sight guidance class. If a range-tracking capability already is contained within the system, it should be possible to implement this improved guidance technique through modifications to the existing guidance computer.





## APPENDIX A

### DESCRIPTION AND USE OF SUBROUTINE INTEG2

INTEG2 provides a simple means for a Runge-Kutta solution of ordinary differential equations. Input and output routines (including graphical point plots on the versatec plotter) are built in.

The subroutine has been compiled and added to the FORTRAN library, SYS1.MPSLIB. The program is flexible and self contained. It will seldom be desirable to build the subroutine into a more complex program.

It is assumed that the equations to be solved have been reduced to 'N' simultaneous first order differential equations of the form:

$$DX(I)/DT = F(X(1), X(2), \dots, X(N), T), \quad I = 1, 2, \dots, N$$

where the function 'F' may be nonlinear. The user must translate these 'N' equations into FORTRAN as in the following second-order example:

$$XDOT(1) = X(2)$$

$$XDOT(2) = C(1)*X(1) - 0.2*X(2) + C(2)*SIN(T*T)$$

C(1) and C(2) are constants which are entered on data cards.

To solve these equations the OS/360 deck must be of the following form:

```
//----STANDARD JOB CARD (GREEN)----
```

```
// EXEC FORTCLGV,REGION.G0=150K
```

```
//FORT.SYSIN DD *
```

```
        DIMENSION X(30),XDOT(30),C(15)
```



```
C(10) = 1.
```

```
1 CALL INTEG2(T,X,XDOT,C)
```

```
. . .
```

```
USERS FORTRAN EQUATIONS
```

```
. . .
```

```
GO TO 1
```

```
END
```

```
/*
```

```
//GO.SYSIN DD *
```

```
DATA CARDS AS DESCRIBED BELOW
```

```
/* (ORANGE)
```

The number of XDOT equations is 'N', which must not exceed 30. These equations are supplied by the user, who defines each 'XDOT' in terms of the dependent variables X(1) through X(N) and the independent variable 'T'. In writing these equations the user may introduce at his convenience:

any unsubscripted variables;

the subscripted variables X(I), where the variable 'I' must be less than or equal to 30;

the constants C(I) to be entered as data, where the variable 'I' must be less than or equal to eight;

any normal FORTRAN technique or function; and

routines from any source library or user-supplied subroutines.

Note 1: The use of 'auxiliary' X(I), when the variable 'I' is greater than the number of equations to be integrated, does not alter the value of 'N',



the order of the equations.

Note 2: Loops, either with or without a 'DO' statement, are best avoided.

Note 3: 'IF' statements, provided that they do not create a loop, can be used to transfer control within the user's equations. For example, the statement,  $IF(T.GT.10.)C(3)=0.$ , would cause C(3) to take the value zero for all 'T' greater than ten.

The constants C(1) through C(8) may be used as described in the above examples, and are read in from a data card. C(10) must never be used, except as indicated in the standard OS/360 deck above. C(11) and C(12) control the output. For example, if the statements,  $C(11) = 10.0$  and  $C(12) = 2.0$ , are added to the users FORTRAN equations, every tenth integration step will be printed out, and every second step will be plotted. (If not set by the user, default values of 20 and five apply.) C(13) can be similarly used to modify the step size of the numerical integration which is more usually defined on a data card.

Only the independent variable 'T' and the variables X(1) through X(30) can be output. Therefore, to output a quantity which is not one of the 'N' dependent variables. The user must introduce an 'auxiliary' variable X(I), where the variable 'I' must be less than 31 but greater than 'N' - for example, by introducing the FORTRAN equation,  $X(27) = X(3)*X(3)$ , the square of X(3) can be output by the program as X(27). Normally, a line of printout is generated after every 20 integration



steps, and one graph point (per curve) after every five integration steps. These values can be modified as indicated in the previous section. A run is terminated if the run exceeds 450 lines or if more than 900 graph points are generated per curve.

The accuracy of the numerical integration is controlled by the choice of step size. Unfortunately there is no simple method for making this choice other than by trial and error. As a first guess, a step size of 1/1000 of the total range of interest of the independent variable 'T' is usually reasonable. If one or more of the variables change rapidly in part of the range of 'T', then a finer step size may be necessary for all or part of the run. The only way to be reasonably sure of the solution accuracy is to re-run the problem with a smaller step size and, then, to compare the results. A maximum of 4500 integration steps is permitted in any one run.

Data cards are assembled as follows:

- First:           The user's job identification label is punched in columns 1-42.
- Second:          The number of runs to be processed, which must be less than ten, is punched in column one. The run number together with the user's job identification label is placed on all output.
- Third:           The order of the differential equations, which must be less than 31, is punched with two digits in columns one and two - for example, '03'.





Fourth: Initial and final values of the independent variable (TI and TF) and the step size (DT) appear in the order TI - DT - TF and are punched, with decimal points, in columns 1-10, 11-20 and 21-30. Also, it is possible to process the integration in either two or three segments of differing step size. The data, in columns of ten as above, then takes the form TI - DT1 - TF1 - DT2 - TF or TI - DT1 - TF1 - DT2 - TF2 -DT3 - TF.

Fifth: The values of the constants C(1), C(2),..... C(8) are punched, with decimal points, in columns 1-10, 11-20,....,71-80. Ten blank columns may be used for a zero or unused constant. If no constants are used, this entire card will be blank --- but it must not be omitted.

Sixth: The initial values of X(1), X(2),....,X(N) are punched, with decimal points, in columns 1-10,11-20,... . Additional cards are required if 'N' is greater than eight. Ten blank columns may be used for a zero initial value.

Seventh: This card controls the choice of variables for printout. Each group of ten columns, 1-10,11-20,...up to 71-80, may be used to specify a column heading (eight characters)



and a two-digit, right-justified subscript identifying the corresponding variable - for example: TIME 00 and SPEED 03 in columns 1-10 and 11-20 would cause the independent variable 'T', represented on this card by the subscript 00, to be printed out under the column heading 'TIME' and the variable X(3) to be printed out under the heading 'SPEED'. No more than eight variables can be output during one run. If no printout is desired, this card must be blank.

Eighth: This card controls the choice of variables for graph output. Up to four curves can be plotted, either all on separate graphs or all on a single graph. Each group of 20 columns 1-20, 21-40, ... specifies a curve - for example: SPEED VS. TIME 0300 (in columns 1-20) would cause X(3) to be plotted vertically against the independent variable 'T' horizontally. (Again, the subscript 00 represents the independent variable 'T'.) The graph output would be labeled 'SPEED VS. TIME'. Note that the first 16 columns of each group are used for the label, the seventeenth and eighteenth for the 'Y' ordinates, and the last two for the 'X' ordinates.



If the curves are to be on separate graphs, each graph must have a label. If the curves are to be all on one graph, only the first label, columns 1-16, must be provided. The other labels must be blank. The entire card must be blank if no graphical output is required.

If several solutions are required of the same equations but with different constants or initial conditions, the number of runs, specified on the second data card, can be increased. For all runs after the first, only the fourth through the last data cards are to be supplied. Except for the data on the first three cards, and the equations themselves, no information is retained between runs.

Caution should be exercised by the user because too large an integration step size can result in unacceptable errors in the solution, and even in instability in the integration process.

This subroutine is maintained for use with the Naval Postgraduate School, Monterey, California, IBM 360 computer. It was programmed by J. R. Ward in October, 1963; was revised in November, 1971; and was redocumented in October, 1973.













TT=0.

```

** SUBROUTINE FOR RUNGA-KUTTA SOLUTION OF ORDINARY
** DIFFERENTIAL EQUATIONS. NOTE, NEW VALUES FOR
** RUN TIMES, TIME INCREMENTS, C(N) CONSTANTS, AND
** X(N) INITIAL CONDITIONS ARE READ.

```

1 CALL INTEG2(T,X,XDOT,C)

```

** VALUE OF C(1) CHANGED TO AN INTEGER STORED AS K

```

K=C(1)

```

** IF THE INTEGER VALUE OF C(1) IS 2, THE PROGRAM
** FREEZES TARGET AND MISSILE POSITIONS BY GOING
** INTO AN INFINITE LOOP UNTIL THE INTEGRATION
** REACHES FINAL TIME.

```

IF(K.EQ.2)GO TO 1

```

** TARGET X POSITION DETERMINED.

```

X(28)=A+VTX\*T

```

** TARGET Y POSITION DETERMINED.

```

X(27)=B+VTY\*T

```

** TARGET ANGLE FROM REFERENCE CALCULATED.

```

SIGT=A TAN(X(27)/X(28))

```

** ESTIMATED RANGE FROM TRACKER TO MISSILE
** CALCULATED FROM TIME OF FLIGHT

```

CCCCCCCC CCCCCC CCCCCCCCCC CCCCCC CCCCCC CCCCCC CCCCCC

















'LEAD-ANGLE' COMMAND TO LINE-OF-SIGHT SIMULATION

C\*\*\*\* INSERT STANDARD GREEN JOB CARD WITH TIME=3 AT THIS POSITION  
 // EXEC FORTCLGV  
 //FORT.SYSIN DD \*

LEAD-ANGLE CLOS MISSILE GUIDANCE SIMULATION

A INITIAL X POSITION OF TARGET (METERS)  
 B INITIAL Y POSITION OF TARGET (METERS)  
 C(1) INPUT VALUE ASSIGNED TO FLAG K. 0 PRODUCES FULL PROGRAM OUTPUT WHILE 1, FREEZES THE PROGRAM OPERATION AT THE CLOSEST POINT OF APPROACH. CPA. (UNITLESS)  
 C(2) INPUT VALUE ASSIGNED TO FLAG L. 0, WHICH IS ALWAYS READ AS INPUT, INDICATES THAT THE MISSILE HAS NOT REACHED CPA TO THE TARGET. THE VALUE IS CHANGED TO 1 WHEN THE MISSILE REACHES CPA. (UNITLESS)  
 C(10) CONSTANT FOR SUBROUTINE INTEG2 WHICH ALWAYS IS SET WITH VALUE 1.0 AS REQUIRED BY THE SUBROUTINE. (UNITLESS)  
 C(11) DETERMINES HOW OFTEN INTEGRATION STEPS ARE PRINTED OUT (IF EQUAL 5 THEN EVERY FIFTH PRINTED OUT) DEFAULT 20 (UNITLESS)  
 C(12) DETERMINES HOW OFTEN INTEGRATION STEPS ARE PLOTTED (IF EQUAL 2 THEN EVERY SECOND STEP IS PLOTTED) DEFAULT 5 (UNITLESS)  
 E ANGULAR ERROR. THE ANGULAR DIFFERENCE BETWEEN THE COURSE TO THE MISSILE SHOULD 'FLY' AND THE ANGLE TO THE MISSILE. (RADIAN)  
 E00T ANGULAR ERROR RATE (RADIAN PER SECOND)  
 EX VARIABLE WHOSE SIGN(+-) DETERMINES MISSILE MANEUVER DIRECTION. THE ERROR FUNCTION (METERS) GEOMETRIC DETERMINATION OF PERPENDICULAR DISTANCE OF MISSILE FROM THE LINE OF SIGHT. LOS. (METERS)  
 EX1 TIME RATE OF CHANGE OF EX1 (METERS PER SECOND)  
 EX2 MAGNITUDE OF THE MISSILE CROSS-RANGE ACCELERATION (METERS PER SECOND SQUARED)  
 K FLAG USED TO INDICATE ATTAINMENT OF CPA AND TO 'FREEZE' PROGRAM OPERATION (UNITLESS)  
 L FLAG USED TO INDICATE ATTAINMENT OF CPA AND TO SHIFT THE SIGN OF TDFR. (UNITLESS)  
 PI THE VALUE OF PI. (METERS)  
 RM RANGE TO MISSILE FROM THE REFERENCE PLANE (RADIAN)  
 SIGM ANGLE OF MISSILE FROM THE REFERENCE PLANE (RADIAN)



SIGT	ANGLE OF TARGET LOS FROM REFERENCE PLANE (RADIAN)
SLOS	ANGLE OF SYNTHETIC LOS FROM REFERENCE (RADIAN)
T	TIME OF FLIGHT REMAINING UNTIL IMPACT (SECONDS)
TOFR	PREVIOUS ITERATIVE VALUE OF TIME (SECONDS)
TT	MISSILE CROSS-RANGE ACCELERATION USED IN STATE
U	VARIABLE ANALYSIS, SVA, INCLUDING DIRECTION AS DETERMINED BY SIGN OF EX (METERS PER SECOND SQUARED)
VC	MISSILE-TARGET CLOSING VELOCITY (METERS PER SECOND)
VMX	MISSILE VELOCITY IN X DIRECTION (METERS PER SECOND)
VMY	MISSILE VELOCITY IN Y DIRECTION (METERS PER SECOND)
VIX	TARGET VELOCITY IN X DIRECTION (METERS PER SECOND)
VITY	TARGET VELOCITY IN Y DIRECTION (METERS PER SECOND)
X(1)	DETERMINATION OF PERPENDICULAR DISTANCE OF MISSILE FROM LOS, CROSS-RANGE ERROR (METERS)
X(2)	TIME RATE OF CHANGE OF X(1), CROSS-RANGE ERROR (METERS PER SECOND)
X(7)	MISSILE X POSITION (METERS)
X(8)	MISSILE Y POSITION (METERS)
X(9)	ANGLE OF MISSILE FROM THE REFERENCE PLANE (DEGREES)
X(10)	OUTPUT OF VARIABLE FOR U (METERS PER SECOND SQUARED)
X(11)	ANGLE OF VARIABLE FOR U FROM REFERENCE PLANE (DEGREES)
X(12)	SEPARATION OF TARGET MISSILE FROM TARGET (METERS)
X(13)	OUTPUT VARIABLE FOR E (DEGREES)
X(14)	OUTPUT VARIABLE FOR EDOT (DEGREES PER SECOND)
X(15)	OUTPUT VARIABLE FOR TOFR (SECONDS)
X(16)	OUTPUT VARIABLE FOR EX
X(17)	IMPACT POINT'S Y POSITION (METERS)
X(18)	IMPACT POINT'S X POSITION (METERS)
X(19)	OUTPUT VARIABLE FOR SLOS (DEGREES)
X(20)	OUTPUT VARIABLE FOR EX1 (METERS)
X(21)	OUTPUT VARIABLE FOR EX2 (METERS PER SECOND)
X(27)	TARGET Y POSITION (METERS)
X(28)	TARGET X POSITION (METERS)
XDOT(1)	SVA EQUATION
XDOT(2)	STORAGE FOR PREVIOUS VALUE OF X(1) (METERS PER SECOND)
WW	STORAGE FOR PREVIOUS VALUE OF X(2) (METERS PER SECOND)
Z	STORAGE FOR PREVIOUS VALUE OF X(12). THE INITIAL VALUE IS LARGE TO ENSURE PROGRAM IS NOT PREMATURELY FROZEN.

CC



```

** ** ** ** **
** INITIALIZATION OF CONDITIONS PRIOR TO FIRST RUN
** NOTE, IF ALTERED DURING A RUN, THESE CONDITIONS
** ARE NOT RESET FOR SUBSEQUENT RUNS.
** ** ** ** **

```

```

DIMENSION X(30), XDOT(30), C(15)

```

```

C(10)=1.
C(11)=5.
C(12)=2.159265
VN=500.
A=5000.
B=0.
SIGM=ATAN(B/A)
VTX=(-177.)
VTY=177.
Z=100000.
TT=0.
TCFR=0.

```

```

** ** ** ** **
** SUBROUTINE FOR RUNGA-KUTTA SOLUTION OF ORDINARY
** DIFFERENTIAL EQUATIONS. NOTE, NEW VALUES FOR
** RUN TIMES, TIME INCREMENTS, C(N) CONSTANTS, AND
** X(N) INITIAL CONDITIONS ARE READ.
** ** ** ** **

```

```

1 CALL INTEG2(T, X, XDOT, C)

```

```

** ** ** ** **
** VALUE OF C(1) CHANGED TO AN INTEGER STORED AS K
** ** ** ** **

```

```

K=C(1)

```

```

** ** ** ** **
** IF CPA ATTAINED, FREEZE VALUES OF CROSS-
** RANGE ERROR AND CRCS--RANGE ERROR RATE
** ** ** ** **

```

```

IF(K.EQ.2)X(1)=W
IF(K.EQ.2)X(2)=WW

```

```

C C C C C

```

```

C C C C C C C C C C C C C C C C

```









```

** ** ** ** **
** ESTIMATED RANGE FROM TRACKER TO MISSILE
** CALCULATED FROM TIME OF FLIGHT
** ** ** **

```

```
RM=VM*T
```

```

** ** ** **
** ANGULAR ERROR AND ANGULAR ERROR RATE DETERMINED
** NOTE, EDOT IS THE DERIVATIVE OF E.
** ** ** **

```

```

E=SIGM-SLOS
EDOT=(+1./SQRT(1.-X(1)*X(1)/(RM*RM)))*(X(2)/RM-X(1)/(RM*T))

```

```

** ** ** **
** GEOMETRIC DETERMINATION OF THE PERPENDICULAR
** DISTANCE OF THE MISSILE FROM THE LINE OF SIGHT.
** CALCULATION OF THE RATE OF CHANGE IN THAT
** DISTANCE.
** ** ** **

```

```

EX1=RM*SIN(E)
EX2=RM*EDOT*COS(E)+SIN(E)*VM

```

```

** ** ** **
** REDUCTION OF THE AMOUNT OF THRUST-VECTOR
** CONTROL UTILIZED WHEN PHASE PLANE ERROR IS
** CLOSE TO ZERO.
** ** ** **

```

```

G=60.
IF((ABS(EX1)+ABS(EX2)).LE.1.0)G=G/6.

```

```

** ** ** **
** PARABOLIC SWITCHING BOUNDARY ESTABLISHED WHICH
** ZEROS ERROR IN MINIMUM TIME.
** ** ** **

```

```

EX=EX1+(EX2*ABS(EX2)/(2.*G))
U=(-G)*SIGN(1.,EX)

```

```

** ** ** **
** STATE VARIABLE EQUATIONS
** ** ** **

```

```

XCOT(1)=X(2)
XDOT(2)=U

```

```

CCCCC CCCCCC CCCCCC CCCCCC CCCCCC CCCCCC CCCCCC

```













```

X(16)=EX
X(19)=SLOS*180./PI
X(20)=EX1
X(21)=EX2
GO TO 1
END

```

```

C*** INSERT A CARD WITH /* IN COLUMNS ONE AND TWO AT THIS POSITION
//GO.FTC6F001 DD SPACE=(CYL,(6,1))
//GO.SYSIN DD *
LEAD-ANGLE SIMULATION

```

```

3
2
1.005 .005 10.
10. 50.
TIME 00CRE 01CRERATE 02M-T SEP 12MX POSIT08TX POSIT28MY PCSIT07TY POSIT27
POSITION VS TIME0800 0700 2800 2700
1.005 .005 10.
10. 50.
TIME 00SIGM 09SIGT 11SLOS 15EX1 20EX2 21TOFR 15U 10
CRE VS TIME .005 10.
1. 10.
50.

```

```

RELATIVE POSITS 0708
C*** INSERT /* CRANGE CARD AT THIS POSITION IN THE PROGRAM

```



C\*\*\* INSERT STANDARD GREEN JOB CARD WITH TIME=3 AT THIS POSITION  
 // EXEC FOR TGLV, REGION. PLOTGEN=250K  
 // FORT.SYS IN DD \*

CCONSOLIDATED CLOS MISSILE GUIDANCE SIMULATION

A	INITIAL X POSITION OF TARGET (METERS)
AC	MISSILE-TARGET CLOSING ACCELERATION MAGNITUDE (METERS PER SECOND SQUARED)
B	INITIAL Y POSITION OF TARGET (METERS)
C(1)	INPUT VALUE ASSIGNED TO FLAG K. 0 PRODUCES FULL PROGRAM OUTPUT WHILE 2 'FREEZES' THE PROGRAM OPERATION AT THE CLOSEST POINT OF APPROACH, CPA. (UNITLESS)
C(10)	CONSTANT FOR SUBROUTINE INTEG2 WHICH ALWAYS IS SET WITH VALUE 1.0 AS REQUIRED BY THE SUBROUTINE. (UNITLESS)
C(11)	DETERMINES HOW OFTEN INTEGRATION STEPS ARE PRINTED OUT (IF EQUAL 5 THEN EVERY FIFTH PRINTED OUT) DEFAULT 20 (UNITLESS)
C(12)	DETERMINES HOW OFTEN INTEGRATION STEPS ARE PLOTTED (IF EQUAL 2 THEN EVERY SECOND STEP IS PLOTTED) DEFAULT 5 (UNITLESS)
D	HALF A PERIOD OF TARGET'S SINUSOIDAL MOTION (SECONDS)
E	ANGULAR ERROR OF THE MISSILE SHCULD 'FLY' AND THE COURSE TO THE MISSILE. (RADIAN)
EDOT	ANGULAR ERROR RATE (RADIAN PER SECOND)
EX	VARIABLE WHOSE SIGN(+-) DETERMINES MISSILE MANEUVER DIRECTION. THE ERROR FUNCTION (METERS)
EX1	CROSS-RANGE ERROR (METERS)
EX2	CROSS-RANGE ERROR RATE (METERS PER SECOND)
G	MAGNITUDE OF THE MISSILE CROSS-RANGE SQUARED
I	ACCELERATION (METERS PER SECOND)
J	COUNTER FOR 'OO' LOGS (UNITLESS)
K	FLAG USED TO INDICATE WHETHER TRACKER IS JAMMED, '0. (UNITLESS)
	FLAG 'FREEZE' PROGRAM OPERATION (UNITLESS)



M FLAG USED TO INDICATE WHETHER THE IMPACT POINT IS MOVING TOO FAST TO ATTACK, INDICATED BY 1, OR SUFFICIENTLY STABLE TO ATTACK, INDICATED BY 0. (UNITLESS)  
P STORAGE VARIABLE FOR ARITHMETIC EXPRESSION USED IN SUBSEQUENT CALCULATIONS IN THE PROGRAM. (UNITLESS)  
PI THE VALUE OF PI. (UNITLESS)  
RM ESTIMATED RANGE OF PI. TC MISSILE (METERS)  
SGMD DERIVATIVE OF SIGM. RATE OF CHANGE IN THE MISSILE'S ANGULAR POSITION. (RADIAN PER SECOND)  
SIGM ANGLE OF MISSILE FROM THE REFERENCE PLANE (RADIAN)  
SIGTD ANGLE OF TARGET LOS FROM REFERENCE PLANE (RADIAN)  
SIGTLD TIME RATE OF CHANGE OF SIGT (RADIAN PER SECOND)  
SLOS ANGLE OF SYNTHEIC LOS FROM REFERENCE PLANE (RADIAN)  
SLOSD TIME RATE OF CHANGE OF SLOS (RADIAN PER SECOND)  
T ARRAY USED TO STORE TIMES JAMMING BEGINS (SECONDS)  
TJBJ ARRAY USED TO STORE TIMES JAMMING CEASES (SECONDS)  
TJJC ARRAY USED TO STORE TIMES JAMMING CEASES (SECONDS)  
TOFR TIME OF FLIGHT REMAINING UNTIL IMPACT (SECONDS)  
TOFRD RATE OF CHANGE OF TIME OF FLIGHT REMAINING UNTIL IMPACT. (UNITLESS)  
TT PREVIOUS ITERATIVE VALUE OF TIME (SECONDS)  
U MISSILE Y-DIRECTION ACCELERATION USED IN STATE VARIABLE ANALYSIS, SVA, INCLUDING DIRECTION AS DETERMINED BY SIGN OF EX (METERS PER SECOND SQUARED)  
VC MISSILE-TARGET CLOSING VELOCITY (METERS PER SECOND)  
VI MAGNITUDE OF THE IMPACT POINT'S VELOCITY. (METERS PER SECOND)  
VIX VELOCITY OF IMPACT POINT IN X DIRECTION (METERS PER SECOND)  
VIY VELOCITY OF IMPACT POINT IN Y DIRECTION (METERS PER SECOND)  
VM MISSILE VELOCITY (METERS PER SECOND)  
VMX MISSILE VELOCITY IN X DIRECTION (METERS PER SECOND)  
VMXD MISSILE X-DIRECTION ACCELERATION (METERS PER SECOND SQUARED)  
VMY MISSILE VELOCITY IN Y DIRECTION (METERS PER SECOND)  
VMYD MISSILE Y-DIRECTION ACCELERATION (METERS PER SECOND SQUARED)  
VTX TARGET VELOCITY IN X DIRECTION (METERS PER SECOND)  
VTXD TARGET X-DIRECTION ACCELERATION (METERS PER SECOND SQUARED)  
VTXM MAXIMUM TARGET VELOCITY IN THE X DIRECTION (METERS PER SECOND)  
VTY TARGET VELOCITY IN Y DIRECTION (METERS PER SECOND)

CC













1 CALL INTEG2(T,X,XDOT,C)

CCCCC

\*\* VALUE OF C(1) CHANGED TO AN INTEGER STORED AS K \*\*

K=C(1)

CCCCC

\*\* IF CPA NOT ATTAINED, CONTINUE CALCULATIONS \*\*

IF(K.NE.2)GO TO 200

CCCCC

\*\* INITIALIZE CONDITIONS JUST PRIOR TO NEXT RUN \*\*

IF(T.LT.9.9)GO TO 100

J=0  
M=0  
SIGM=0.  
TOFR=0.  
SICSD=0.  
X7=0.  
X8=0.  
VMY=0.  
VMYD=0.  
VMX=0.  
VMXD=0.  
SGMD=0.

CCCCC

\*\* HOLD VALUES OF X(1) AND X(2) CCNSTANT AFTER PROGRAM FROZEN. \*\*

100 X(1)=W  
X(2)=WW

CCCCCCCC

\*\* THE INTEGER VALUE OF C(1) IS 2, THE PROGRAM FREEZES TARGET AND MISSILE POSITIONS BY GOING INTO AN INFINITE LOOP UNTIL THE INTEGRATION REACHES FINAL TIME. \*\*

GO TO 1







X(27)=B+VTY\*M/D/PI\*SIN(PI\*T/D)  
 \*\* TARGET ANGLE FROM REFERENCE AND TARGET ANGULAR  
 \*\* RATE CALCULATED.

SIGT=ATAN(X(27)/X(28))  
 SIGTD=(1./X(28)\*X(28)+X(27)\*X(27))\*X(28)\*VTY-X(27)\*VTX  
 \*\* X POSITION OF IMPACT REVISED.

X(18)=X(28)+VTX\*TOFR  
 \*\* Y POSITION OF IMPACT REVISED.

X(17)=X(27)+VTY\*TOFR  
 \*\* SYNTHETIC LOS FROM REFERENCE CALCULATED

SLOS=ATAN(X(17)/X(18))  
 \*\* MISSILE-TO-TARGET RANGE CALCULATED

X(12)=SQRT((X(27)-X(7))\*(X(27)-X(7))+X(28)-X(8))\*X(28)-X(8))  
 \*\* MISSILE-TO-TARGET CLCSING VELOCITY CALCULATED

VC=SQRT((VTY-VMY)\*(VTY-VMY)+(VTX-VMX)\*(VTX-VMX))  
 \*\* TIME OF FLIGHT REMAINING UNTIL IMPACT  
 \*\* COMPUTED.

TOFR=X(12)/VC

C C





\*\*\* PREVENT POSSIBLE DIVISION BY ZERO. \*\*\*

IF(X(12).LT.0.000001)X(12)=.000001  
IF(VC.LT.0.000001)VC=.000001

\*\*\* DETERMINE VALUE OF F AND CALCULATE RATE OF  
\*\*\* CHANGE OF X(12), XRATE. THIS IS USED LATER IN  
\*\*\* DETERMINING TOFRD. \*\*\*

P=SQRT(RM\*RM-X(11)\*X(11))  
XRATE=((X(27)-X(7))\*((VTY-X(2))+X(28)-X(8))\*(VTX-RM\*VM/P+X(1))\*X(2)  
1/P)/X(12)

\*\*\* CALCULATE MISSILE-TARGET CLOSING ACCELERATION  
\*\*\* AND RATE OF CHANGE OF TOFR \*\*\*

AC=((VTY-VMY)\*(VTYD-VMYD)+(VTX-VMX)\*(VTXD-VMXD))/VC  
TOFRD=XRATE/VC-X(12)\*AC/(VC\*VC)

\*\*\* DETERMINE RATE OF CHANGE IN THE IMPACT  
\*\*\* POSITION AND SYNTHETIC LINE OF SIGHT. \*\*\*

VIX=VIX\*(1.+TOFRD)+VTXD\*TOFR  
VIY=VIY\*(1.+TOFRD)+VTYD\*TOFR  
VI=SQRT(VIX\*VIX+VIY\*VIY)  
SLOSD=(X(18)\*VIX-X(17)\*VIX)/(X(18)\*X(18)+X(17)\*X(17))

\*\*\* DETERMINE IF TARGET MANEUVERS ARE CAUSING THE  
\*\*\* PREDICTED IMPACT POINT TO MOVE AT TOO HIGH A  
\*\*\* RATE FOR USE. IF SO, SET FLAG M TO 1 TO  
\*\*\* INDICATE A HIGHLY MANEUVERING TARGET AND, THUS,  
\*\*\* TO CAUSE THE MISSILE TO BE COMMANDED TO FLY  
\*\*\* DIRECTLY TOWARD THE TARGET VICE THE IMPACT  
\*\*\* POINT. \*\*\*

M=0  
IF(VI.GT.(.8\*VM))M=1

CCCCC CCCCCC CCCCCC CCCCCC CCCCCC CCCCCC C







```

U=(-G*COS(SIGT))*SIGN(1.,EX)
** STATE VARIABLE EQUATIONS **
** **
500 XDOT(1)=X(2)
    XCOT(2)=U
** **
** DETERMINE MISSILE VELOCITIES AND ACCELERATIONS. **
** THE BRANCHING LOGIC PREVENTS POSSIBLE DIVISION **
** BY ZERO. HOWEVER, THE VALUE OF X(2) SHOULD **
** NEVER EXCEED THE MISSILE'S VELOCITY. IF THIS **
** CONDITION OCCURRED, THE RESULTS WOULD BE **
** INVALID. **
IF(ABS(X(2)).LT.VM)GO TO 600
VMX=0.
VMXD=0.
GO TO 700
VMX=SQRT(VM*VM-X(2)*X(2))
VMXD=(-U)*X(2)/VMX
VMY=X(2)
VMYD=U
** **
** CALCULATION OF ANGLE OF MISSILE FROM REFERENCE **
** **
SIGM=ARSIN(X(1)/RM)
** **
SGMD=(+1./SQRT(1.-X(1)*X(1)/(RM*RM)))*(X(2)/RM-X(1)/(RM*T))
** **
** CALCULATION OF RATE OF CHANGE IN THE MISSILE'S **
** ANGULAR POSITION. **
** **
** MISSILE Y POSITION DETERMINED **
** **
X(7)=RM*SIN(SIGM)

```



C C C C  
C C C C C C C  
C C C C C  
C C C C C C C  
C C C C C C C  
C C C C C C C  
C

```

** MISSILE X POSITION DETERMINED
** ** ** ** **
X(8)=RM*COS(SIGM)
** ** ** ** **
** REVERSE THE VALUES OF X7 AND X8 SO THEY WILL NOT
** ** ** ** **
** ALTER THE VALUE OF X(7) AND X(8), THE MISSILE'S
** ** ** ** **
** POSITION, DURING THE NEXT ITERATION.
** ** ** ** **

```

```

X7=X(7)
X8=X(8)
** ** ** ** **
** STORE CURRENT VALUES OF X(1) AND X(2).
** ** ** ** **

```

```

W=X(1)
WW=X(2)
** ** ** ** **
** FREEZE PROGRAM AFTER 9.9 SECONDS. THE MISSILE
** ** ** ** **
** IS ASSUMED TO GO BALLISTIC AT THIS TIME.
** ** ** ** **

```

```

IF(T.GE.9.9)C(1)=2.
** ** ** ** **
** FREEZE PROGRAM, AFTER 8 SECONDS OF FLIGHT, ARE
** ** ** ** **
** IF BOTH TIME AND MISSILE-TO-TARGET RANGE ARE
** ** ** ** **
** GREATER THAN THE PREVIOUS ITERATION.
** ** ** ** **
** MISSILE HAS REACHED CPA.
** ** ** ** **

```

```

IF(X(12).GT.Z.AND.T.GT.TT.AND.T.GT.9.0)C(1)=2.
** ** ** ** **
** STORE CURRENT VALUES OF TIME AND MISSILE-TO-
** ** ** ** **
** TARGET RANGE FOR USE DURING THE NEXT ITERATION.
** ** ** ** **
** VALUES ARE USED TO DETERMINE IF CPA REACHED.
** ** ** ** **

```

```

TT=T
Z=X(12)

```









150. 150. 00AC 26TOFR 15TOFRD 29MX RATE 03MY RATE 02MX ACCEL04MY ACCEL10  
TIME 00AC 26TOFR 15TOFRD 29MX RATE 03MY RATE 02MX ACCEL04MY ACCEL10  
RELATIVE POSITS 0708 10.  
1.0 .005

150. 150. 13EDOT 14IX POSIT1E1Y POSIT17IX RATE 05IY RATE 06IV 16  
TIME 00E 13EDOT 14IX POSIT1E1Y POSIT17IX RATE 05IY RATE 06IV 16

C\*\*\*\* INSERT /\* ORANGE CARD AT THIS POSITION IN THE PROGRAM



SUBROUTINE INTEG2

NOTE, INTEG1/2 SUBROUTINES PRODUCE THE SAME RESULTS. HOWEVER, INTEG2 PRODUCES PLOTTED OUTPUT ON THE VERSATEC PLOTTER WHILE INTEG1 PRODUCES PLOTTED OUTPUT ON THE REGULAR OFF-LINE PRINTER.

```

SUBROUTINE INTEG1(/IC/,/XC/,/DX/,/C/)
REAL*8 ITITLE(I2),JTITLE(8),KTITLE(8),IBLANK/'
DIMENSION X(30),BX(30),XC(30),C(15),IP(10),IG(10),GR(10),
1TX(5),TY(5),X1(900),Y1(900),X2(900),Y2(900),X3(900),Y3(900),
2X4(900),Y4(900)
REAL LABEL, RUN, ' ,BT(9)/' 1 ' , 2 ' , 3 ' , 4
1 5 ' , 6 ' , 7 ' , 8 ' , 9 ' ,
EQUIVALENCE (ITITLE(7),RUN(1))
INDIC = C(10)+0.000001
GO TO (1, 2000, 50, 58, 88, 88), INDIC

C
C
C READ DATA AND PRINT RECORD.
1 READ(5,100) (ITITLE(I), I=1,6)
FCRMAT (10A8)
READ (5,101)NR
FORMAT (11)
101 READ (5,102)NN
FORMAT (12)
NRC = 0
IF (NN.LE.30) GO TO 1000
WRITE(6,200)
FORMAT (///, 48H ERROR IN ORDER OF EQUATION. MUST NOT EXCEED 30. )
STOP
1000 NRC = NRC + 1
WRITE (6,201) (ITITLE(I), I=1,6)
FORMAT (1H1,///,36X,6A8)
201 IF(NRC.EQ.1) AND.NR.EQ.1) GO TO 5
WRITE(6,202)NR
FORMAT (/,37X,11,20H RUNS ARE CALLED FOR )
GO TO 6
5 WRITE(6,203)
FORMAT (/,37X,21H ONE RUN IS CALLED FOR ,///,18H INPUT DATA RECORD)
203 GO TO 7
6 WRITE(6,204)NRC
FORMAT (///,34H INPUT DATA RECORD FOR RUN NUMBER ,11)
204 WRITE(6,205)NN

```

INT12560  
INT12570  
INT12580  
INT12590  
INT12600  
INT12610  
INT12620  
INT12630  
INT12640  
INT12650  
INT12660  
INT12670  
INT12680  
INT12690  
INT12700  
INT12710  
INT12720  
INT12730  
INT12740  
INT12750  
INT12760  
INT12770  
INT12780  
INT12790  
INT12800  
INT12810  
INT12820  
INT12830  
INT12840  
INT12850  
INT12860  
INT12870  
INT12880  
INT12890  
INT12900  
INT12910  
INT12920



```

205 FORMAT (/, 22H ORDER OF EQUATIONS = , I2)
103 READ (5, 103) TI, DT, TF1, DT2, TF2, DT3, TF3
    FFORMAT (8F10.4)
    TF = TF1
    IF (DT2.NE.0.) GO TO 9
    WRITE(6, 206) TI, TF
    FFORMAT (22H INITIAL TIME = , E10.4, /
    206 1 FFORMAT (22H FINAL TIME = , E10.4)
    WRITE(6, 207) DT
    FFORMAT (22H STEP SIZE = , E10.4)
    207 GO TO 12
    5 IF (DT3.NE.0.) GO TO 11
    TF = TF2
    WRITE(6, 206) TI, TF
    WRITE(6, 208) DT, TI, TF1, DT2, TF1, TF
    208 1 FFORMAT (22H STEP SIZE = , E10.4, 13H BETWEEN T = , E10.4,
    GO TO 12
    TF = TF3
    11 WRITE(6, 208) DT, TI, TF1, DT2, TF1, TF2, DT3, TF2, TF
    12 READ (5, 103) (C(I), I=1, 8)
    READ (5, 103) (X(I), I=1, NN)
    J = 0
    DC 14 I=1, 8
    IF (C(I).NE.0.) J=J+1
    14 K = 0
    CONTINUE
    DO 16 I=1, NN
    IF (X(I).NE.0.) K=K+1
    16 CONTINUE
    IF (J - 1) 17, 18, 19
    17 WRITE (6, 209)
    209 FFORMAT (/, 34H ALL THE CONSTANTS, C(I), ARE ZERO )
    GO TO 423
    18 WRITE (6, 210)
    210 FFORMAT (/, 30H THE ONLY NON-ZERO CCNSTANT IS )
    GO TO 420
    19 WRITE (6, 211)
    211 FFORMAT (/, 35H THE NON-ZERO CONSTANTS, C(I), ARE )
    420 DO 422 I=1, 8
    IF (C(I).NE.0.) WRITE(6, 212) I, C(I)
    212 FFORMAT (14X, 2HC(, I2, 4H) = , E10.4)
    422 CONTINUE
    423 IF (K - 1) 424, 425, 426
    424 WRITE (6, 1209)
    1209 FFORMAT (/, 36H ALL THE INITIAL CONDITIONS ARE ZERO )
    GO TO 20
    425 WRITE (6, 1210)

```

```

INT12930
INT12940
INT12950
INT12960
INT12970
INT12980
INT12990
INT13000
INT13010
INT13020
INT13030
INT13040
INT13050
INT13060
INT13070
INT13080
INT13090
INT13100
INT13110
INT13120
INT13130
INT13140
INT13150
INT13160
INT13170
INT13180
INT13190
INT13200
INT13210
INT13220
INT13230
INT13240
INT13250
INT13260
INT13270
INT13280
INT13290
INT13300
INT13310
INT13320
INT13330
INT13340
INT13350
INT13360
INT13370
INT13380
INT13390
INT13400

```





```

1210 FORMAT (/ ,3SH THE ONLY NON-ZERO INITIAL CONCNITION IS )
GO TO 427
426 WRITE (6,1211)
1211 FORMAT (/ ,36H THE NON-ZERO INITIAL CONDITIONS ARE )
427 DO 429 I=1,NN
IF(X(I).NE.0.) WRITE(6,1212) I,X(I)
1212 FORMAT(14X,2HX(,I2,4H) = ,E10.4)
429 CONTINUE
20 READ (5,104) (JTITLE(I),IP(I),I=1,8)
104 FORMAT(8(A8,I2))
C
C CHECK FOR THE NUMBER OF COLUMNS CALLED FOR BY LOCATING FIRST
C BLANK COLUMN HEADING
C
DO 21 J=1,8
IF(JTITLE(J).EQ.IBLANK) GO TO 22
21 CONTINUE
J = 9
22 JJ = J - 1
C
C JJ IS NOW THE NUMBER OF COLUMNS. REPEAT WITH THE GRAPHS.
C
C
105 READ (5,105)(KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,7,2)
FORMAT(4(2A8,2I2))
DC 24 K=1,7,2
IF(KTITLE(K).EQ.IBLANK.AND.KTITLE(K+1).EQ.IBLANK) GO TO 25
24 CCNTINUE
K = 8
25 KK = K/2
KKK = KK*2
MULTIP = 0
IF(KK.NE.1) GO TO 306
IF(IG(3) + IG(4).EQ.0) GO TO 306
IF(IG(5) + IG(6).NE.0) GO TO 303
MULTIP = 2
KKK = 4
GO TO 306
IF(IG(7) + IG(8).NE.0) GO TO 305
MULTIP = 3
KKK = 6
GO TO 306
MULTIP = 4
KKK = 8
303 IF MULTIP = 0, KK IS THE NUMBER OF SINGLE CURVE GRAPHS. OTHERWISE
MULTIP IS THE NUMBER OF CURVES ON A SINGLE GRAPH.
C
C
305 MULTIP = 4
KKK = 8
306 IF(JJ.EQ.0) GO TO 27

```

```

INT13410
INT13420
INT13430
INT13440
INT13450
INT13460
INT13470
INT13480
INT13490
INT13500
INT13510
INT13520
INT13530
INT13540
INT13550
INT13560
INT13570
INT13580
INT13590
INT13600
INT13610
INT13620
INT13630
INT13640
INT13650
INT13660
INT13670
INT13680
INT13690
INT13700
INT13710
INT13720
INT13730
INT13740
INT13750
INT13760
INT13770
INT13780
INT13790
INT13800
INT13810
INT13820
INT13830
INT13840
INT13850
INT13860
INT13870
INT13880

```



```

WRITE(6,214) (JTITLE(I),IP(I),I=1,JJ)
FORMAT (///,56H THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES
1S ARE //,(10X,A8,4X,2HX(,I2,1H)))
GO TO 28
27 WRITE(6,215)
FORMAT (///,25H NO PRINTOUT IS REQUIRED )
215 IF(KK.EQ.0) GO TO 308
IF(MULTIP.NE.0) GO TO 309
IF(KK.NE.1) GO TO 307
WRITE(6,216) KTITLE(1),KTITLE(2),IG(1),IG(2)
FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES AR
1E, //,10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H))
GO TO 31
216 WRITE(6,217) (KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,KK,2)
FORMAT (///,64H THE INDIVIDUAL GRAPH TITLES AND THE CORRESPONDING
1VARIABLES ARE //,(10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H)))
GO TO 31
308 WRITE(6,1217)
FORMAT (///,24H NO GRAPHS ARE REQUIRED )
GO TO 31
309 WRITE(6,1220)
FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES AR
1E, //)
WRITE(6,1221) KTITLE(1),KTITLE(2),(IG(I),IG(I+1),I=1,KK,2)
FORMAT (10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H),/, (30X,2HX(,I2,
8H) VS. X(,I2,1H)))
31 IPAGE = 0
T = TI
NOPTS = 0
NUMPTS = 0
ITITLE(8) = IBLANK
ITITLE(11) = IBLANK
ITITLE(12) = IBLANK
RUN(2) = BIT(NRC)
C(11) = 20.
C(12) = 5.
C(13) = DT
DC 42 I = 1, NN
XC(1) = X(I)
TC = T
C(10) = 2.
RETURN
200C IF(JJ.EQ.0) GO TO 54

```

INT1385C  
INT13900  
INT13910  
INT13920  
INT13930  
INT13940  
INT13950  
INT13960  
INT13970  
INT13980  
INT13990  
INT14000  
INT14010  
INT14020  
INT14030  
INT14040  
INT14050  
INT14060  
INT14070  
INT14080  
INT14090  
INT14100  
INT14110  
INT14120  
INT14130  
INT14140  
INT14150  
INT14160  
INT14170  
INT14180  
INT14190  
INT14200  
INT14210  
INT14220  
INT14230  
INT14240  
INT14250  
INT14260  
INT14270  
INT14280  
INT14290  
INT14300  
INT14310  
INT14320  
INT14330  
INT14340  
INT14350  
INT14360

C  
C  
C  
THIS ENDS THE BOOK-KEEPING. INITIALIZE BEFORE ENTERING MAIN LOOP.



```

INCPR = C(11)+0.000001
C(11) = 20.
IF( MOD (NOPTS,50*INCPR).EQ.0) GO TO 46
IF( MOD (NOPTS,10*INCPR).EQ.0) GC TO 47
IF( MOD (NOPTS,
46 IPAGE=EQ.1) GO TO 1047
IF(NR.EQ.1) GO TO 1047
WRITE(6,218) (JTITLE(I),I=1,6),IPAGE,ITITLE(7),(JTITLE(I),I=1,8)
WRITE(6,219)
GO TO 47
1047 WRITE(6,1218) (ITITLE(I),I=1,6),IPAGE,(JTITLE(I),I=1,8)
WRITE(6,219)
47 WRITE(6,219)
218 FCRMAT (1H,///,20X,6A8,10X,5HPAGE ,11,14H OF OUTPUT FOR,A8,////,
1218 FCRMAT (1H,///,20X,6A8,30X,5HPAGE ,11,////,11X,8(A8,5X))
219 FCRMAT (1H, )
48 DO 49 I=1,NN
49 XC(I) = X(I)
TC = T
C(10) = 3.
RETURN
C
C 50 DO 53 I=1,JJ
PR(I) = T
IF(IP(I).NE.0) PR(I)=XC(IP(I))
53 CONTINUE
WRITE(6,22C)(PR(I),I=1,JJ)
220 FCRMAT (7X,8E13.5)
54 IF(KK.EQ.0) GO TO 62
INCPR = C(12)+0.000001
C(12) = 5.
IF( MOD (NOPTS, INCPR).NE.0) GO TO 62
DO 57 I=1,NN
57 XC(I) = X(I)
TC = T
C(10) = 4.
RETURN
C
C 58 DO 61 I=1,KKK
GR(I) = T
IF(IG(I).NE.0) GR(I)=XC(IG(I))
61 CONTINUE
IF(KKK.GE.8) GO TO 1610
KPI = KKK + 1
DC 1612 I=KPI,8

```

```

INT14370
INT14380
INT14390
INT14400
INT14410
INT14420
INT14430
INT14440
INT1445C
INT14460
INT14470
INT14480
INT14490
INT14500
INT14510
INT14530
INT14540
INT14550
INT14560
INT14570
INT14580
INT14590
INT14600
INT14610
INT14620
INT14630
INT14640
INT14650
INT14660
INT14670
INT14680
INT14690
INT14700
INT14710
INT14720
INT14730
INT14740
INT14750
INT14760
INT14770
INT1478C
INT14790
INT14800
INT14810
INT14820
INT14830
INT1484C

```



```

1612 GR(I) = 0. NUMPTS + 1
1610 NUMPTS = NUMPTS
Y1(NUMPTS) = GR(1)
Y2(NUMPTS) = GR(2)
Y3(NUMPTS) = GR(3)
Y4(NUMPTS) = GR(4)
Y5(NUMPTS) = GR(5)
Y6(NUMPTS) = GR(6)
Y7(NUMPTS) = GR(7)
Y8(NUMPTS) = GR(8)
NOPTS = NOPTS + 1
62 IF(NUMPTS.LI.900) GO TO 64
WRITE(6,221)
FCRMT(////////,25H STOP AT 900 GRAPH POINTS )
221 GO TO 91
IF(NOPTS.LI.4500) GO TO 66
WRITE(6,222)
FCRMT(////////,31H STOP AT 4500 INTEGRATION STEPS )
222 GO TO 91
IF(IPAGE - 9)69,67,68
66 IF(MOD(NOPTS, 50*INCR).NE.0) GO TO 69
67 WRITE(6,223)
68 FCRMT(////////,27H STOP AT 9 PAGES OF OUTPUT )
223 GO TO 91
69 DC 70 I=1,NN
70 IF(ABS(X(I)).GT.1.E+12) GO TO 71
GO TO 72
71 WRITE(6,224)
72 FCRMT(////////,76H STOP WITH THE ABSOLUTE VALUE OF A DEPENDENT VAR
INTEGRATION PROBABLY UNSTABLE.
,75F
INTEGRATION PROBABLY UNSTABLE.
)
224 IF(ABLE GREATER THAN 1.0E+12.
,75F
INTEGRATION PROBABLY UNSTABLE.
)
2 TRY A SMALLER STEP SIZE.
,26HNO GRAPHS WILL BE PLCTED.
GO TO 330
72 DT = C(13)
73 IF(TI.GT.TF) GO TO 80
74 IF(T.LT.TF) GO TO 75
74 WRITE(6,225)
225 FCRMT(////////,26H NORMAL STOP AT FINAL TIME )
GO TO 91
75 IF(T.GE.TF1) GO TO 77
76 C(13) = DT
GO TO 87
77 IF(T.GE.TF2) GO TO 79
78 C(13) = DT2
GO TO 87
79 C(13) = DT3
GO TO 87
80 IF(TF.GE.T) GO TO 74

```

```

INT14850
INT14860
INT14870
INT14880
INT14890
INT14900
INT14910
INT14920
INT14930
INT14940
INT14950
INT14960
INT14970
INT14980
INT14990
INT15000
INT15010
INT15020
INT15030
INT15040
INT15050
INT15060
INT15070
INT15080
INT15090
INT15100
INT15110
INT15120
INT15130
INT15140
INT15150
INT15160
INT15170
INT15180
INT15190
INT15200
INT15210
INT15220
INT15230
INT15240
INT15250
INT15260
INT15270
INT15280
INT15290
INT15300
INT15310
INT15320

```





```

INT115330
INT115340
INT115350
INT115360
INT115370
INT115380
INT115390
INT115400
INT115410
INT115420
INT115430
INT115440
INT115450
INT115460
INT115470
INT115480
INT115490
INT115500
INT115510
INT115520
INT115530
INT115540
INT115550
INT115560
INT115570
INT115580
INT115590
INT115600
INT115610
INT115620
INT115630
INT115640
INT115650
INT115660
INT115670
INT115680
INT115690
INT115700
INT115710
INT115720
INT115730
INT115740
INT115750
INT115760
INT115770
INT115780
INT115790
INT115800

```

```

IF(TF1.LT.T) GO TO 76
IF(TF2 - T) 78,79,79
87 C(10) = 5.
C
88 CALL RKUTTA (NN,T,X,DT,C,TC,XC,DX)
C
90 IF(C(10).EQ.6.) RETURN
T = T + DT
GC TO 2000
91 IF(KK.EQ.0) GO TO 330
IF(MULTIP.NE.0) GO TO 97
C
C PRINT PLOT LP TO 4 INDIVIDUAL CURVES
C
NUMPTS=-NUMPTS
DO 310 I=1,KK
WRITE(6,9998)
FORMAT(1H1)
ITITLE(9)=KTTITLE(2*I-1)
ITITLE(10)=KTTITLE(2*I)
GO TO (211,212,313,314),I
311 CALL PLOTP(X1,Y1,NUMPTS,0)
GO TO 310
312 CALL PLOTP(X2,Y2,NUMPTS,0)
GO TO 310
313 CALL PLOTP(X3,Y3,NUMPTS,0)
GO TO 310
314 CALL PLOTP(X4,Y4,NUMPTS,0)
310 WRITE(6,9999) ITITLE
9999 FORMAT(1H0,8X,12A8)
GO TO 330
C
C PLOT DUMMY CURVE ALONG AXES TO SET SCALES FOR MULTIPLE PLOT
C
BIGX = 0.
BIGY = 0.
SMLX = 0.
SMLY = 0.
SMLY = 0.
DO 1970 I=1,NUMPTS
XMAX = A MAX I ( X1(I), X2(I), X3(I), X4(I) )
YMAX = A MAX I ( Y1(I), Y2(I), Y3(I), Y4(I) )
XMIN = A MIN I ( X1(I), X2(I), X3(I), X4(I) )
YMIN = A MIN I ( Y1(I), Y2(I), Y3(I), Y4(I) )
IF(BIGX.LT.XMAX) BIGX=XMAX
IF(BIGY.LT.YMAX) BIGY=YMAX
IF(SMLX.GT.XMIN) SMLX=XMIN
IF(SMLY.GT.YMIN) SMLY=YMIN
1970 C CONTINUE

```



INT15810  
 INT15820  
 INT15830  
 INT15840  
 INT15850  
 INT15860  
 INT15870  
 INT15880  
 INT15890  
 INT15900  
 INT15910  
 INT15920  
 INT15930  
 INT15940  
 INT15950  
 INT15960  
 INT15970  
 INT15580  
 INT15990  
 INT16000  
 INT16010  
 INT16020  
 INT16030  
 INT16040  
 INT16050  
 INT16060  
 INT16070  
 INT16080  
 INT16090  
 INT16100  
 INT16110  
 INT16120  
 INT16130  
 INT16140  
 INT16150  
 INT16160  
 INT16170  
 INT16180

```

TX(1) = 0.
TX(2) = 0.
TX(3) = 0.
TX(4) = 0.
TX(5) = 0.
TX(1) = SMLX
TX(2) = BIGX
TX(3) = BIGY
TX(4) = SMLY
TX(5) = 0.
TY(1) = 0.
TY(2) = 0.
TY(3) = 0.
TY(4) = 0.
TY(5) = 0.
WRITE(6,9998)
ITITLE(9) = KTITLE(1)
ITITLE(10) = KTITLE(2)
NT=-5
CALL PLOTP(TX, TY, NT, 1)
MODCUR = 2
DO 410 I=1, MULTIP
  IF(I.EQ.MULTIP) MODCUR=3
  GO TO (411,412,413,414), I
  CALL PLOTP(X1, Y1, NUMPTS, MODCUR)
411 GO TO 410
412 CALL PLOTP(X2, Y2, NUMPTS, MODCUR)
413 GO TO 410
414 CALL PLOTP(X3, Y3, NUMPTS, MODCUR)
415 GO TO 410
416 CALL PLOTP(X4, Y4, NUMPTS, MODCUR)
417 CONTINUE
418 WRITE(6,9999) ITITLE
C 330 IF(NRC.NE.NR) GO TO 1000
IF(NR.GT.1) GO TO 333
WRITE(6,226)
226 FORMAT(/,43H THE ONE RUN CALLED FOR HAS BEEN COMPLETED. ,/)
STOP
333 WRITE(6,227)NR
227 FORMAT(/,5F THE ,11,37H RUNS CALLED FOR HAVE BEEN COMPLETED.,/)
STOP
END

```

C  
 C  
 C  
 C  
 C



```

SUBROUTINE RKUTTA(/NN/,/T/,/X/,/DT/,/C/,/TC/,/XC/,/DX/,
DIMENSION X(30), C(15), XC(30), DX(30), CT(4), AK(4,30)
REAL*8 AK,CT
INDIC = C(10) - 4.0+0.0000001
IF(INDIC.GT.1) GO TO 3
CT(1) = 0.000
CT(2) = 0.500
CT(3) = 0.500
CT(4) = 1.000
II=0
7 II=II+1 + CT(II)*DT
DC 2 J=1,NN
2 XC(J) = X(J) + CT(II)*AK(II-1, J)
C(10) = 6.0
RETURN
DO 4 J=1,NN
4 AK(II, J) = DT*DX(J)
IF(II.LT.4) GO TO 7
DO 5 J=1,NN
5 X(J)=X(J)+(AK(1,J)+2.0*(AK(2,J)+AK(3,J))+AK(4,J))/6.0
C(10) = 7.0
RETURN
END

```

```

INT116190
INT116200
INT116210
INT116220
INT116230
INT116240
INT116250
INT116260
INT116270
INT116280
INT116290
INT116300
INT116310
INT116320
INT116330
INT116340
INT116350
INT116360
INT116370
INT116380
INT116390
INT116400
INT116410
INT116420

```



## BIBLIOGRAPHY

1. U.S. Army Foreign Science and Technology Center Report FSTC-1202-75, The Roland-1 Armored Self-Propelled AA Missile System, by Johannes and Weyand, p. 9-20, 10 June 1976.
2. Gibson, J. E., Nonlinear Automatic Control, p. 237, 342, 439-449, McGraw-Hill, 1963.
3. Goodstein, R., "Guidance Law Applicability for Missile Closing," Guidance and Control of Tactical Missiles, AGARD Lecture Series No. 52, 1972.
4. Heap, E., "Methodology of Research into Command-to-Line-of-Sight and Homing Guidance," Guidance and Control of Tactical Missiles, AGARD Lecture Series No. 52, 1972.
5. Ogata, K., Modern Control Engineering, p. 153-154, 537-538, 563-605, 663-704, Prentice-Hall, 1970.





INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California	2
3. Chairman, EW Academic Group, Code 73 Naval Postgraduate School Monterey, California 93940	1
4. Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5. LCDR Frank F. Hewitt 2315 Geranium Street San Diego, California 92109	1
6. Mr. Larry D. Parks and Mr. Ron Russell ADTC/ADA Eglin AFB, Florida 32542	1
7. Mr. Jack Willett Stanford Research Institute 333 Ravenswood Avenue Menlo Park, California 94025	1
8. Mr. David Powell DRDMI-YPE U.S. Army Missile R & D Command Redstone Arsenal, AL 35809	1
9. Mr. Arlo A. Herrick 3246th TW TEER Eglin AFB, Florida 32542	1
10. Mr. Stan Wagner, Code 3513 Naval Weapons Center China Lake, California 93555	1
11. Mrs. Sylvia L. Waller Headquarters USAF AF/PACC The Pentagon Washington, D. C. 20330	1



No. Copies

12. Mr. Vern Green  
Naval Air Systems Command  
AIR 53322E  
Washington, D. C. 20361

1







Thesis  
H5262445 Hewitt 180760  
c.1

Computer simulated  
development of im-  
proved command to  
line-of-sight missile  
guidance techniques.

30 APR 80  
6 MAR 86  
24 FEB 89

255831  
33111  
33624

Thesis 180760  
H5262445 Hewitt  
c.1

Computer simulated  
development of im-  
proved command to  
line-of-sight missile  
guidance techniques.



thesH5262445

Computer simulated development of improv



3 2768 000 99247 3

DUDLEY KNOX LIBRARY