

2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AD-A211 093

1b. RESTRICTIVE MARKINGS

3. DISTRIBUTION/AVAILABILITY OF REPORT

Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

AFOSR-TR-89-0980

6a. NAME OF PERFORMING ORGANIZATION

Moore School of EE, D2 Department
of SE6b. OFFICE SYMBOL
(if applicable)

7a. NAME OF MONITORING ORGANIZATION

AFOSR

6c. ADDRESS (City, State, and ZIP Code)

University of Pennsylvania
Philadelphia, Pennsylvania 19104

7b. ADDRESS (City, State, and ZIP Code)

BLDG 410
BAFB DC 20332-64488a. NAME OF FUNDING/SPONSORING
ORGANIZATION

AFOSR

8b. OFFICE SYMBOL
(if applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

AFOSR 77-3327

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

BLDG 410
BAFB DC 20332-6448

10. SOURCE OF FUNDING NUMBERS

PROGRAM
ELEMENT NO.
61102FPROJECT
NO.
2304TASK
NO.
A1WORK UNIT
ACCESSION NO.

11. TITLE (Include Security Classification)

A SIMULATION STUDY OF FOUR REAL-TIME HEURISTIC ALGORITHMS FOR MULTIPLE MISSILE MISSILE
EVASION: A GAME THEORETIC APPROACH

12. PERSONAL AUTHOR(S)

Max Mintz, Michael S. Sheketoff/ Stephan F. Huling

13a. TYPE OF REPORT

13b. TIME COVERED

FROM _____ TO _____

14. DATE OF REPORT (Year, Month, Day)

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD GROUP SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

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

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT

☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ OTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22c. OFFICE SYMBOL

AFOSR-TR- 89-0985

FINAL REPORT

Title: A SIMULATION STUDY OF FOUR REAL-TIME
HEURISTIC ALGORITHMS FOR MULTIPLE MISSILE
EVASION: A GAME THEORETIC APPROACH

Authors: Mr. Michael S. Sheketoff
Mr. Stephen F. Huling
Dr. Max Mintz

Date: June 1979

Submitted To: Air Force Office of Scientific Research

Prepared Under: Grant No. 77-3327

Submitted On Behalf Of: The University of Pennsylvania,
Philadelphia, PA

By: Dr. Max Mintz,
Principal Investigator

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution _____	
Availability Codes	
Avail and/or	
Dist	Special
A-1	



Navis 4
Abstract

Abstract

Four real-time heuristic algorithms for determining aircraft evasion strategies against a multiple missile threat are described. Algorithms 1 and 2 are based on a "myopic" saddle-point calculation which apportions the projection of the instantaneous aircraft acceleration among the normals to the individual maneuver or guidance planes defined by each missile and its target. Algorithms 3 and 4 are also based on "myopic" saddle-point calculations. These latter two algorithms apportion the projection of the instantaneous aircraft acceleration into the individual maneuver planes so as to maximize the minimum of a function which is related to the line of sight rate of each missile threat. These latter two algorithms are motivated by the concept of anti-proportional navigation.

Each algorithm has the following properties: i) each requires relatively minimal dynamic and parametric information; ii) each provides capability against an N missile threat; iii) each generates aerodynamically feasible aircraft maneuvers which meet both structural and pilot stress limitations; iv) each is computable using foreseeable hardware; v) each exhibits markovian behavior, i.e., each is restartable from present state information.

Simulation results using each algorithm with generic F-4 and AIM-9 truth models, characterized by nonlinear differential equations, including lift, drag, gravity, 3-dimensional point mass dynamics, aircraft load factor and roll rate limits, and missile autopilot dynamics and load factor limits are presented.

All four heuristic algorithms are motivated by a formal game theoretic model for multiple missile evasion. This formal game theoretic analysis is included as part of this study.

Acknowledgment

The authors wish to acknowledge the invaluable assistance and advice of the following individuals:

Dr. H. M. Dobbins, Capt. R. N. Lutter, USAF, & Mr. M. J. Noviskey, of the Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio.

Maj. J. G. Reid, USAF, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Cmdr. F. X. Shannon, USN, & Mr. W. Weddleton, of the University of Pennsylvania.

This research was supported in part by the United States Air Force through the following grants and contracts:

The Air Force Office of Scientific Research, under Grant 77-3327;

The Charles Stark Draper Laboratory, under Contract DL-H-139457.

The technical and financial support of all of these individuals and organizations is gratefully acknowledged.

Table of Contents

Abstract	ii
Acknowledgment	iii
Chapter 1	
1.0 Introduction and Summary	1-1
Chapter 2	
2.0 Overview, Alternative Problem Characterizations, and Background Information	2-1
2.1 Overview	2-1
2.2 Alternative Problem Characterizations and Background Information	2-2
Chapter 3	
3.0 Heuristic Algorithms for Determining Evasion Strategies Suitable for Real-Time Onboard Computation	3-1
3.1 Introduction	3-1
3.2 Aircraft and Missile Models	3-1
3.3 Remarks on Modeling Philosophy and Underlying Assumptions	3-10
3.3.1 Attributes of Aircraft-Missile Models in the Context of Evasive Aircraft Maneuvering	3-10
3.3.2 Potentially Important Factors Omitted in the Missile Model	3-12
3.4 Aircraft Evasion Strategies - Basic Considerations	3-12
3.4.1 Qualitative Aspects	3-12
3.4.2 Quantitative Methods	3-13
3.4.3 Geometric Aspects of Missile Guidance	3-14
3.5 Heuristic Algorithms	3-17
3.6 Information Requirements	3-20
Chapter 4	
4.0 Computer Simulation Models for Determining and Evaluating the Global and Local Characteristics of Evasive Maneuvers	4-1

4.1	Introduction	4-1
4.2	The Simulation Paradigm and Engagement Scenarios	4-1
4.2.1	The Simulation Paradigm	4-1
4.2.2	The Engagement Scenarios	4-3
4.3	Algorithm Performance	4-3
Chapter 5		
5.0	A Game Theoretic Model for Determining Aircraft Evasion Strategies Against a Multiple Missile Threat	5-1
5.1	Optimal Evasion Strategies Against Multiple Missiles: Part I - For Criteria with a Fixed Terminal Time	5-2
5.1.1	Introduction	5-2
5.2	An Optimal Evasion Problem with Linear Dynamics and Quadratic Cost on Control	5-4
5.2.1	Mathematical Problem Statement	5-4
5.2.2	Game Theoretic Formulation	5-5
5.3	An Optimal Evasion Problem with Linear Dynamics and a Total Energy Constraint	5-9
5.3.1	Problem Statement	5-9
5.3.2	Problem Solution	5-10
5.3.3	Solution Structure	5-11
5.4	A Second Optimal Evasion Problem with Linear Dynamics and a Total Energy Constraint	5-12
5.4.1	Problem Statement	5-12
5.4.2	Problem Solution	5-13
5.5	An Optimal Evasion Problem with Linear Dynamics and Hard Constraints on Control	5-17
5.5.1	Problem Statement	5-17
5.5.2	Problem Solution	5-17

5.6	An Optimal Evasion Problem with Nonlinear Dynamics and Hard Constraints on Control	5-20
5.6.1	Problem Statement	5-20
5.6.2	Problem Solution	5-21
5.7	An Optimal Evasion Problem with Nonlinear Dynamics, Hard Constraints on Control, and a General Terminal Cost Function	5-24
5.7.1	Problem Statement	5-24
5.7.2	Problem Solution	5-25
5.8	Optimal Evasion Strategies Against Multiple Missiles: Part II - For Criteria with a Free Terminal Time	5-27
5.8.1	Introduction	5-27
5.9	An Optimal Evasion Problem with Nonlinear Dynamics, Hard Constraints on Control, and a Generalized Cost Function	5-29
5.9.1	Problem Statement	5-29
5.9.2	Problem Solution	5-30
5.10	Historical Precis	5-31
Chapter 6		
6.0	Conclusions and Recommendations for Further Research	6-1
References		R-1
Appendix A		A-1

List of Tables and Figures

Table 2-1a	2-5
Table 2-1b	2-9
Table 3-1a	3-3
Figure 3-1a	3-4
Table 3-1b	3-5
Table 3-1c	3-7
Figure 3-1b	3-9
Figure 3-2	3-16
Figures 4-1(a - h)	4-5
Table 4-1	4-13
Figures 4-2(a - c)	4-14
Tables 4-2(a - b)	4-17
Table 4-3	4-19
Figures 4-3(a - e)	4-20
Table 4-4	4-25
Figures 4-4(a - d)	4-26
Table 4-5	4-30
Figures 4-5(a - e)	4-31
Table 4-6	4-36
Figures 4-6(a - d)	4-37
Table 4-7	4-41
Figures 4-7(a - d)	4-42
Table 4-8	4-46
Figures 4-8(a - e)	4-47
Table 4-9	4-52
Figures 4-9(a - d)	4-53

Table 4-10	4-57
Figures 4-10(a - e)	4-58
Table 4-11	4-63
Figures 4-11(a - d)	4-64

1.0 Introduction and Summary

This report presents the results of research on the application of a blend of game theory, control theory, and decision analysis to air-to-air fire control. The focus of this study embraces the determination of aircraft evasion strategies against both single and multiple missile threats. The results include:

i) The delineation of mathematical models for the determination of optimal evasion strategies against multiple missile threats.

ii) The design and implementation of a series of computer simulation models for studying the characteristics of evasive aircraft maneuvers against both single and multiple missile threats.

iii) The determination and validation of heuristic algorithms which could be implemented to achieve real-time onboard calculation and execution of evasive maneuvers against single and multiple missile threats.

This report is presented in six parts:

Part 1. Introduction and Summary

Part 2. Overview, Alternative Problem Characterizations, and Background Information

Part 3. Heuristic Algorithms for Determining Evasion Strategies Suitable for Real-Time Onboard Computation

Part 4. Computer Simulation Models for Determining and Evaluating the Global and Local Characteristics of Evasive Maneuvers

Part 5. A Game Theoretic Model for Determining Aircraft Evasion Strategies Against a Multiple Missile Threat

Part 6. Conclusions and Recommendations for Further Research.

The material is presented in this order to emphasize the goal of applicability, as opposed to the delineation of elegant mathematical models which provide base lines or figures of merit but do not lead directly to real-time algorithms.

However, the historical evolution of this research follows the ordering Part 5, Part 4, Part 3.

The work contained in Parts 3 & 4 represent research contributions by Mr. Michael Sheketoff and are the basis for his Masters Thesis in Systems Engineering (Sheketoff, 1979). Parts 3 & 4 also provide the basis for two papers (Sheketoff & Mintz, 1979a, & 1979b). The work in Part 5 represents research contributions by Mr. Stephen F. Huling and are a part of his Ph.D. Dissertation in Systems Engineering (Huling, 1979). Part 5 also provides the basis for a paper (Huling & Mintz, 1977).

2.0 Overview, Alternative Problem Characterizations, and Background Information

2.1 Overview

The generic problem of determining how to maneuver an aircraft to evade a missile threat is by no means new. The U.S. experience in Viet Nam and the Israeli experience in the 1967 and 1973 wars have clearly demonstrated the need to develop and perfect practical methods for blunting the effectiveness of various missile threats. We cite the following open literature as being representative of recent work in missile evasion and allied studies which have been carried out by other investigators: (FAAC, 1977); (Garnell & East, 1977); (Grumman, 1975); (Grumman, 1976); (Poulter, 1975); (Shinar & Steinberg, 1976); (TASC, 1977); & (Veda, 1977). The dual to the missile evasion problem, which has aroused at least equal interest, is the determination of Missile Launch Envelopes (MLE). The missile evasion problem and its dual, the MLE determination, are clearly linked, since knowledge of when or whether to launch a missile to counter an aircraft threat depends on the potential for evasive maneuvering on the part of the target aircraft. We cite the following recent technical proposals (submitted in response to RFP F33615-77-R-1014) by the First Ann Arbor Corporation (FAAC, 1977) and the Veda Corporation (Veda, 1977) as useful sources of relevant information for characterizing the problem of missile launch envelope sensitivity analysis for an air-to-air missile engagement.

Although the dual problems of evasive maneuver determination and launch envelope determination are of interest in the air-to-air and ground-to-air contexts, we shall focus our attention on the air-to-air case in the remainder of this report. The simulation models described in Part 4 of this report could however be relatively easily modified to represent the threat of a multiple Surface-to-Air Missile (SAM) attack. In principle, the real-time

heuristic algorithms which are described in Part 3 of this report could be applied to counter a multiple SAM threat. In order to evaluate the performance of these heuristic algorithms against a multiple SAM threat, the user would need to make some modifications in the simulation programs in Part 4 to account for the appropriate dynamics and guidance of the SAMs.

2.2 Alternative Problem Characterizations and Background Information

The art of problem solving in an engineering setting requires that we choose that minimal degree of complexity and fidelity for the various models which serve to define our underlying problem, in order to:

- i) capture a sufficient degree of accuracy and
- ii) provide a potentially tractable solution within the confines of our computational limitations.

The determination of evasive maneuvers against single and multiple missile threats provides no exception to this general rule. In this section we shall review the qualitative and quantitative aspects of a variety of alternatives which have been proposed as approaches to obtaining a "solution" to the evasion problem associated with a single missile threat. We begin the discussion of the determination of evasion strategies by considering a single missile threat, since any evasion strategy which protects against a multiple missile threat should have reasonably good properties when used against a single missile. In the process of delineating various approaches to the single missile evasion problem, we shall refer to the dual problem, which we have referred to previously as the MLE determination. This review of current thinking about the MLE problem will provide us with a useful baseline to compare such common facets as:

- i) Modeling Assumptions, &
- ii) Parametric Variability.

The taxonomy of missile evasion and MLE studies includes:

- i) The classification of the models of the underlying dynamical systems.
- ii) The classification of decision criteria which are used to measure the success of a given missile evasion scheme, or the effectiveness of a given missile engagement opportunity against a maneuvering target.
- iii) Evasive maneuver modeling and generation.

The characteristics of the underlying dynamical models can be separated into two basic classes:

- i) Purely Deterministic Models, &
- ii) Models which include Stochastic Effects.

These two classes can, in turn, be partitioned by taxonomic aspects of model complexity and applicability, which are determined by the design and level of detail incorporated in the models of missile and aircraft flight dynamics, control systems, and sensor behavior. These taxonomic aspects could include the following mutually dependent considerations:

- i) Nonlinear vs linear models of system dynamics.
- ii) Inclusion vs exclusion of various lift, drag, and gravity terms.
- iii) 3-dimensional kinematic behavior vs 2-dimensional kinematic behavior.
- iv) Incorporation of constraints on missile and aircraft maneuvers.
- v) Incorporation of deterministic and stochastic aspects of sensor and seeker dynamics, underlying constraints, and error sources.

We have classified the eight references cited in Section 2.1 based on these taxonomic considerations. The results of our classification are contained in Table 2-1. We emphasize that these eight references are merely

representative of recent work in missile evasion and allied studies, and that these references are, in their totality, neither complete with respect to the available literature, nor exhaustive with respect to its content. We also note that the design and rendering of a matrix or table for classifying the taxonomic aspects of a lengthy report, monograph, or proposal is subject to the interpretive analysis of the reviewer, and, hence, the reader should refer the original work to obtain the complete picture. It is also important to recognize that each report, monograph, or proposal was prepared with reference to some specific set of goals, and, hence, any blanket comparison between these works is to some degree inherently unfair.

Table 2-1a

Reference Identification and Miscellaneous NotesReports

Reference 1. (TASC, 1977):

Brown, C.M., and D.H. Johnson, "Fire Control Simulation (FICS) User's Manual," The Analytical Sciences Corp., TR-750-1, July 1977.

Reference 2. (Grumman, 1976):

Carpenter, G. and M. Falco, "Supercruise Vulnerability To Surface-To-Air Missile Threats," Grumman Aerospace Corp., 1976.

Reference 3. (Shinar & Steinberg, 1976):

Shinar, J. and D. Steinberg, "Analysis Of Optimal Evasive Maneuvers Based On A Linearized Two-Dimensional Kinematic Model," Technion - Israel Institute Of Technology, T.A.E. Report No. 230, 1976.

Reference 4. (Poulter, 1975):

Poulter, R.A., "Differential Game Guidance Versus Proportional Navigation For An Air-To-Air Missile," Air Force Institute Of Technology, GA/MC/75-5, December 1975.

Reference 5. (Grumman, 1975):

Carpenter, G. and M. Falco, "Analysis Of Aircraft Evasion Strategies In Air-To-Air Missile Effectiveness Models," Grumman Aerospace Corp., Report No. RE-506, August 1975.

Monographs

Reference 6. (Garnell & East):

Garnell, P. and D.J. East, Guided Weapon Control Systems, Pergamon Press, 1977.

Proposals - MLE Sensitivity Analysis

Reference 7. (FAAC, 1977):

FAAC Proposal FP402U/2762, "Part I - Technical Proposal For A Missile Launch Envelope Sensitivity Analysis," First Ann Arbor Corp., January 1977.

Reference 8. (Veda, 1977):

Veda Proposal 13005-77U/Q0105, "Technical Proposal Volume I - Missile Launch Envelope Sensitivity Analysis," Veda Corp., January 1977.

Continuation of Table 2-1a

Miscellaneous Notes

Reference 1, Abstract:

"This user's manual documents the FIre Control Simulation (FICS) computer program. FICS is a FORTRAN simulation designed for the evaluation of a wide range of defensive fire control processing techniques for a single missile-target in a three dimensional engagement. It contains realistic models for attacking missiles, aircraft maneuver dynamics, escorted aircraft, coarse tracking sensors, a fine tracker, and a weapon pointing and acquisition system. In addition, it contains algorithms for target tracking, threat and kill assessment, weapon handover, and flight control handover. ... It has full monte carlo statistical evaluation capabilities ... "

Reference 2, Abstract:

"This paper presents a new effectiveness methodology with which to quantitatively analyze vulnerability/survivability of aircraft to surface-to-air missile threats. Supercruiser survivability tradeoffs with maneuver performance and threat warning system parameters are presented for a known SAM threat. In addition the survivability sensitivity to a variety of penetration altitude and Mach number flight conditions has been examined.

The approach is a blend of optimal control theory, stochastic learning theory and dynamic simulation resulting in a learning algorithm which permits the evaluation of aircraft evasive maneuvering as an integral part of the survivability measure. The methodology develops optimal evasive strategies in the form of a feedback control policy based upon a discretized set of information states which are available as visual or threat warning system cues to the pilot. The algorithm develops evasive strategies using the optimization criteria of maximizing the survival probability for all possible missile launch conditions."

Reference 3, Abstract:

"Optimal evasion from proportionally guided missiles is analyzed assuming two-dimensional linearized kinematics. By this assumption a simple search technique can be used instead of the cumbersome solution of a two point boundary value problem. Due to the simplicity of this approach it is possible to include in the mathematical model factors which have been neglected in other analytic studies. It is demonstrated that these factors as: the exact dynamic structure of the guidance system, the location of the saturating element in the guidance loop, the limited roll rate of the evading aircraft, etc., have major effects on the optimal maneuver sequence and determine the order of magnitude of the resulting miss distance.

Comparison with studies, which used non-linear kinematic models, shows that the domains of validity of linearized kinematics and two-dimensional analysis coincide. In the case of optimal evasion assessment, both assumptions are limited in their validity to nearly head-on or tail chase engagements. To analyze engagements of other initial conditions a three-dimensional model is required. The method described in this paper can be extended for this type of three-dimensional study."

Continuation of Table 2-1a

Miscellaneous Notes

Reference 4, Abstract:

"Proportional navigation is a closed loop optimal control for the case of a linear dynamic model of the air-to-air missile intercept problem and a quadratic cost function (Ref ...).

This paper presents a differential game model of the intercept problem using nonlinear realistic dynamics, free final time and a terminal cost function related to probability of kill. With this model proportional navigation is no longer optimal and the extent of its nonoptimality is indicated for a range of saddle point solutions.

A guidance concept based on differential game theory is discussed and is compared to proportional navigation in an off line simulation. The considerable gains made by this scheme over proportional navigation provide the incentive to develop a real-time version."

Reference 5, Abstract:

"This report presents a new methodology with which to quantify missile effectiveness and aircraft vulnerability. The approach is a blend of applications of optimal control theory, stochastic learning theory, and simulation. This methodology permits an evaluation of aircraft evasive maneuvering and countermeasures deployment strategy as an integral part of the effectiveness/vulnerability measurements. The strategy determination is a form of feedback control policy based upon a discretized set of information thresholds in the relative coordinate space as would be available to an evading aircraft pilot. The optimization criteria of an evading aircraft is that of maximizing the survival probability for all relative coordinates. The representative model chosen for illustration is evasion from a close range air-to-air IR guided missile."

Reference 6, Notes:

The subject matter of this monograph is based on lecture notes given to the Guided Weapon Systems (M.Sc.) Course at the English Royal Military College Of Science in Shrivenham, Swindon. At the time of the completion of this monograph (August 1976), this course was the only one of its type in the U.K. and had been given continuously for twenty six years.

The chapter headings are: The Performance Of Target Trackers; Missile Servos; Missile Control Methods; Aerodynamic Derivatives And Aerodynamic Transfer Functions; Missile Instruments; Autopilot Design; Line Of Sight Loops; Homing Heads And Associated Stability Problems; Proportional Navigation And Homing Guidance Loops; Wiener Filter Theory Applied To Guidance Loop Design; Modern Control Theory Applied To Guidance Loop Design; Kalman Filters.

Chapter 9, Proportional Navigation And Homing Guidance Loops, contains the bulk of the material in this monograph relating to the effects of target maneuver.

Continuation of Table 2-1a

Miscellaneous Notes

References 7 & 8, Notes:

The FAAC Technical Proposal For A Missile Launch Envelope Sensitivity Analysis contains a very concise and lucid description of the essential factors effecting the performance of an air-to-air missile of the AIM-9L variety. This proposal calls for the use of an extensive simulation model of the AIM-9L, which had been developed by FAAC, to generate the required MLE sensitivity information. Similarly, the Veda Technical Proposal For A Missile Launch Envelope Sensitivity Analysis calls for the use of an extensive six degree of freedom cross-coupled flyout simulation model for the AIM-9L. This latter model was available through the Naval Weapons Center (NWC) at China Lake.

These simulation models represented, as of January 1977, the most widely accepted digital computer simulation models of the AIM-9L. In light of the extensive detail incorporated in these AIM-9L models, the fidelity of the missile models used in References 7 & 8 will dominate the fidelity of the generic air-to-air missile models used in the other representative references.

Table 2-1b

MSL Dynamics

Reference(s)	Type #	Description
3, 6	I	Linearized 2-dimensional point mass model with load factor constraint.
5	II	Nonlinear 2-dimensional point mass model with load factor constraint.
1, 2, 4	III	Nonlinear 3-dimensional point mass model with load factor constraint.
7, 8	IV	AIM-9L industry/government standard.

MSL Autopilot-Airframe Dynamics

Reference(s)	Type #	Description
4	0	Zero order dynamics in roll and AOA.
5	I	Linear first order roll (or yaw) dynamics and zero order AOA dynamics.
1	II	Linear first order AOA dynamics and zero order roll dynamics.
2	III	Linear first order pitch and yaw dynamics.
3, 6	IV	Linear nth order dynamics for lateral acceleration with/without saturating elements.
7, 8	V	AIM-9L industry/government standard.

Stochastic Attributes

Reference(s)	Type #	Description
2 - 5, 7, 8	0	None - a purely deterministic model or application.
6	I	Angular noise and glint modules in monte carlo missile simulation.
1	II	Large scale monte carlo simulation model including effects such as: missile seeker/sensor noise and wind gust disturbances.

Continuation of Table 2-1b

Missile Seeker/Sensor Dynamics and Error Sources

Reference(s)	Type #	Description
2, 3, 4	0	None - no dynamical constraints or error sources are imposed in these models.
5	I	FOV limits, gimbal limits, gimbal rate limit, and first order seeker dynamics.
6	II	FOV limits, gimbal limits, gimbal rate limit, second order seeker dynamics, angular noise, and glint.
1	III	Noise corrupted measurements of target: range, azimuth, elevation, range rate, azimuth rate, elevation rate, and glint (optional).
7, 8	IV	AIM-9L industry/government standard.

Aircraft Dynamics

Reference(s)	Type #	Description
5	I	2-D kinematics with piecewise constant longitudinal acceleration (3 values: +a, 0, -a) - motion along simple arcs.
6	II	2-D kinematics with constant aircraft-missile closure rate and piecewise constant lateral acceleration.
3	III	Same as Type II with a ramp function modification of the piecewise constant acceleration profile when roll rate limits are assumed.
7, 8	IV	3-D kinematics with constant airspeed and zero order lateral acceleration dynamics with load factor constraint.
2, 4	V	Same as Type IV without constant airspeed.
1	VI	3-D kinematics with constant airspeed and first order roll and AOA dynamics with load factor constraint.

3.0 Heuristic Algorithms for Determining Evasion Strategies Suitable for Real-Time Onboard Computation

3.1 Introduction

In this chapter we examine the real-time computation of aircraft evasion strategies against a multiple missile threat using a game theoretic approach. We describe herein several related heuristic algorithms for determining evasion strategies suitable for real-time onboard computation. These algorithms were developed subject to the following goals and constraints. Each algorithm should:

- i) require relatively minimal dynamic and parametric information,
- ii) provide capability against an N - missile threat ($N \geq 2$),
- iii) generate aerodynamically feasible aircraft maneuvers to meet aircraft design limitations and pilot stress limitations,
- iv) be computable using foreseeable hardware, and
- v) exhibit markovian behavior (be restartable from present state information).

3.2 Aircraft and Missile Models

For the purposes of this study we chose generic models of the F-4 aircraft (Phantom II - McDonnell Douglas) and AIM-9 missile (Sidewinder). Salient features of the aircraft and missile models employed in this study, include:

- i) nonlinear models of system dynamics,
- ii) the inclusion of various lift, drag, gravitational forces,
- iii) 3-dimensional kinematic behavior with point mass dynamics,
- iv) the incorporation of constraints on missile and aircraft maneuvers, including: missile autopilot dynamics, aircraft roll rate limits, and missile/aircraft load factor constraints, and
- v) the assumption of a deterministic environment and perfect tracking.

The choice of this type of model was made to achieve a balance between simulation fidelity and computational complexity, in the context of taking a first cut at the real-time computation of aircraft evasion strategies against a multiple missile threat. The attributes of the aircraft and missile models incorporated in this study are closely matched by the dynamical models used in (Poulter, 1975). The differences include the addition of missile autopilot dynamics and aircraft roll rate limitations in the present study.

The details of the F-4 aircraft and AIM-9 missile dynamics used in the present study are contained in Table 3-1.

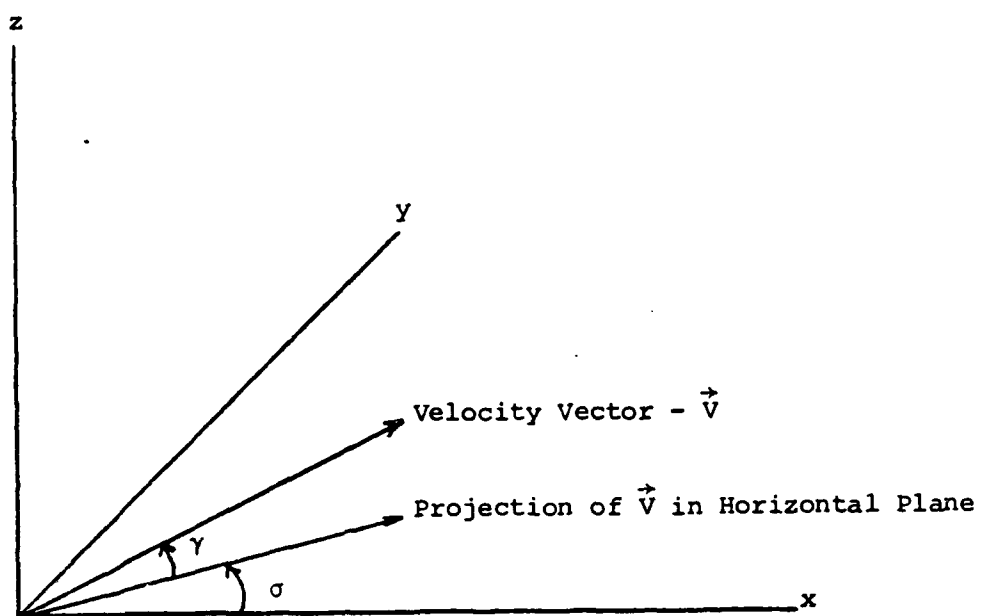
Table 3-1a

Definition of Dynamic Variables and Parametric Values which appear in the Aircraft and Missile Models

Generic Variables and Parameters Common to both Aircraft and Missile Models:

- x, y, z - earth based inertial coordinate system (Cartesian),
- v - airspeed (magnitude),
- γ - angle defined by velocity vector and horizontal (x-y) plane,
- σ - angle defined by the projection of the velocity vector in the horizontal plane and the x-axis,
- T - thrust along vehicle center line,
- α - angle of attack (AOA),
- μ - vehicle bank angle,
- L - lift force (normal to velocity vector and wing planform),
- D - drag force (colinear with velocity vector),
- m - vehicle mass,
- g - gravitational constant,
- n - vehicle load factor ($n = L/mg$),
- C_L - coefficient of lift,
- C_D - coefficient of drag,
- S - characteristic area,
- ρ - air density ($\rho = \rho_0 e^{-\beta z}$)

Figure 3-1a



Earth Based Coordinate System

Table 3-1b

F-4 Aircraft Dynamics

$$dx/dt = v \cos(\gamma) \cos(\sigma)$$

$$dy/dt = v \cos(\gamma) \sin(\sigma)$$

$$dz/dt = v \sin(\gamma)$$

$$dv/dt = (T \cos(\alpha) - D)/m - g \sin(\gamma)$$

$$d\gamma/dt = (L + T \sin(\alpha)) \cos(\mu) / mv - g \cos(\gamma) / v$$

$$d\sigma/dt = (L + T \sin(\alpha)) \sin(\mu) / mv \cos(\gamma)$$

$$T = k_1 + k_2 z + k_3 v$$

$$L = C_L \rho v^2 S / 2$$

$$C_L = C_{L_\alpha} \alpha$$

$$D = C_D \rho v^2 S / 2$$

$$C_D = C_{D_0} + \kappa C_L^2$$

Parametric Values

$$m = 1243.0 \text{ slugs}$$

$$k_1 = 22347.0 \text{ lbf (Afterburner Off); } 35347.0 \text{ lbf (Afterburner On)}$$

$$k_2 = -0.7018 \text{ lbf/ft}$$

$$k_3 = 18.141 \text{ lbf-sec/ft}$$

$$C_{L_\alpha} = 3.8986 \text{ rad}^{-1}$$

$$C_{D_0} = 0.01675$$

$$\kappa = 0.223$$

$$S = 530.0 \text{ ft}^2$$

Continuation of Table 3-1b

Constraints on Aircraft Dynamics

The maximum allowed load factor is 8.0 g. The angle of attack is limited to a maximum value of 25.0 degrees. This corresponds to approximately 29 AOA Units on the cockpit AOA Indicator, based on the relation,

$$\text{AOA Units} = 1.03(\alpha(\text{degrees}) + 3.3)^{\dagger}.$$

The maximum roll rate is limited to 600 deg/sec. This is three times the maximum suggested roll rate in the F-4 "dash - one manual" (USAF, 1977). The consequences of rolling at this higher rate could include damage to missile stores. However, we have learned through informal conversations with F-4 pilots, that F-4 roll rates in excess of 600 deg/sec could be employed in certain combat situations.

[†] McDonnell Douglas Co. - Private Communication

Table 3-1c

AIM-9 Missile Dynamics

$$dx/dt = v \cos(\gamma) \cos(\sigma)$$

$$dy/dt = v \cos(\gamma) \sin(\sigma)$$

$$dz/dt = v \sin(\gamma)$$

$$dv/dt = -(D/m + g \sin(\gamma))$$

$$d\gamma/dt = L \cos(\mu)/mv - g \cos(\gamma)/v$$

$$d\sigma/dt = L \sin(\mu)/mv \cos(\gamma)$$

$$d\alpha/dt = -(\alpha - \alpha_{com})/\tau; \alpha_{com} = \text{AOA commanded by missile guidance system}$$

$$L = C_L \rho v^2 S/2$$

$$C_L = C_{L_\alpha} \alpha$$

$$D = C_{D_0} \rho v^2 S/2$$

$$C_D = C_{D_0} + \kappa C_L^2$$

Parametric Values

$$m = 3.2 \text{ slugs}$$

$$C_{L_\alpha} = 22.918 \text{ rad}^{-1}$$

$$C_{D_0} = 0.7$$

$$\kappa = 0.042$$

$$S = 0.223 \text{ ft}^2$$

$$\tau = 0.15 \text{ sec}$$

Continuation of Table 3-1c

Missile Guidance Law

Proportional Navigation Relations:

Let θ and ψ denote respectively, the yaw and pitch line of sight (los) angles, for the los defined by the missile and aircraft depicted in Figure 3-1b. Let $d\theta/dt$ and $d\psi/dt$ denote their respective rates.

Classical proportional navigation (AFA, 1975) provides that the missile pitch and yaw rates ($d\gamma/dt$ & $d\sigma/dt$) are determined by: $d\gamma/dt = R_p d\psi/dt$, and $d\sigma/dt = R_y d\theta/dt$, where R_p and R_y are the Proportional Navigation Constants in pitch and yaw. Hence,

$$\tan(\mu) = (R_y d\theta/dt) / (R_p d\psi/dt + g \cos(\gamma)/v),$$

$$n = (R_y d\theta/dt) v \cos(\gamma) / \sin(\mu), \text{ if } d\theta/dt \neq 0$$

and

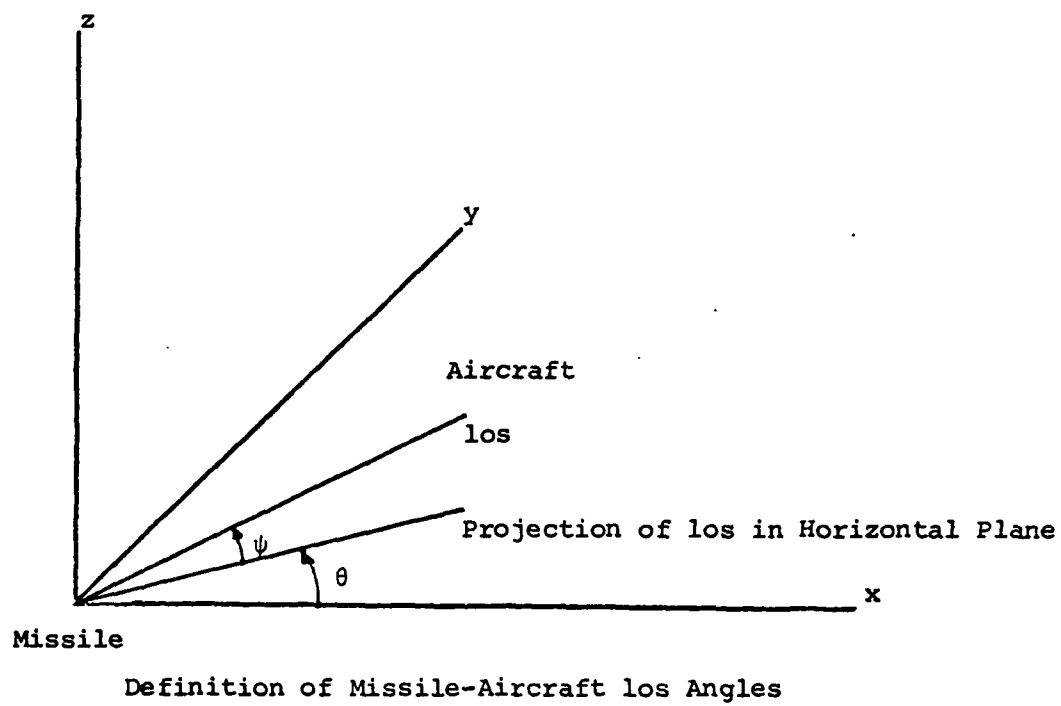
$$n = (v R_p d\psi/dt + g \cos(\gamma)) / g \cos(\mu), \text{ if } d\theta/dt = 0.$$

In this simulation we have specified that $R_p = R_y = 4.5$. We remark that the AIM-9L is roll rate stabilized, and, hence, the pitch and yaw commands in the present simulation are unrealistic to the extent that the lateral acceleration components will vary by as much as a factor of $(2)^{1/2}$, and the instantaneous value of this factor will depend on the roll history of the missile (FAAC, 1977).

Constraints on Missile Dynamics

The missile load factor is constrained to be less than or equal to 15.0 g in absolute value. No roll rate limit is specified. The AOA rate is limited through the first order dynamics specified previously in this Table ($\tau = 0.15$ sec).

Figure 3-1b



3.3 Remarks on Modeling Philosophy and Underlying Assumptions

The choices of aircraft and missile models incorporated in this study were made to capture those attributes of both dynamical systems which were felt to be important in determining significant aspects of both global and local characteristics of missile-aircraft interception behavior. In comparison with the missile models employed in the eight exemplary references, which were reviewed in Chapter 2, the missile model employed in the present study is basically similar in detail to the deterministic aspects of the generic missile model described in (TASC, 1977), the generic missile model described in (Grumman, 1976), and the generic missile model described in (Poulter, 1975) - augmented with autopilot dynamics. In making a similar comparison among the aircraft models, we note that the aircraft model employed in the present study is similar in detail to those described in (Grumman, 1976) and (Poulter, 1975) - when both are augmented to include aircraft roll rate limits.

3.3.1 Attributes of Aircraft-Missile Models in the Context of Evasive Aircraft Maneuvering

The choice of 3-dimensional nonlinear models including lift, drag, and gravitational forces, as well as the incorporation of constraints such as missile autopilot dynamics, aircraft roll rate limits, and missile-aircraft load factor limits, provides the means for accounting for significant aspects of the global characteristics of missile-aircraft interception behavior.

The assumption of "linearizability" has been invoked in prior studies, e.g., (Shinar & Steinberg, 1976) and (Garnell & East, 1977). This assumption, when coupled with a deterministic model, allows one to analyze the evasion problem as a fixed terminal time problem with vehicles that have constant air-speed. The validity of this type of analysis is limited basically to end game analyses on the order of ten missile autopilot-airframe time constants

in duration, with an engagement geometry which is within a small perturbation from tail-chase or head-chase conditions (Shinar & Steinberg, 1976), or within a small perturbation from a collision - constant velocity - flight path (Garnell & East, 1977). The analysis based on linearization is useful in the end game, when the conditions for its validity are met. However, the linearization approach does not allow for the effect of global maneuvers which are designed to evade missiles by initiating maneuvers outside the end game envelope, or to provide for entry into a "favorable set" of end game states. Simulation results, obtained in the present study, indicate that relative missile-aircraft load factor capability, missile autopilot-airframe time constants, aircraft roll rate limit, engagement geometry, relative airspeed, and maneuver timing play a significant role in the outcome of an evasive maneuver. The factors associated with engagement geometry and airspeed variability are global in extent, and therefore, these factors are not amenable to a linearized analysis outside the end game envelope.

It is important to note that the timing of a maneuver outside the end game envelope can significantly affect the actual time at which an intercept can occur. Hence, the goal of maximizing the missile-aircraft miss distance at an a priori specified terminal time can be quite misleading, because the miss distance at the a priori terminal time can be quite large, but, without any further maneuvering by the aircraft, the missile still may achieve an intercept at a later time.

The importance of these previous considerations have been noted in earlier studies, e.g., see the exemplary references cited in Chapter 2. The contribution of the present study focuses on the evasive maneuver strategy determination in a multiple missile environment, when the luxury of a single or one at a time end game analysis may not be available.

3.3.2 Potentially Important Factors Omitted in the Missile Model

The major limitation of the missile model in the present study is the unrealistic sensor-seeker characterization. Factors such as: seeker limits, seeker dynamics, stochastic effects, and blind range are ignored entirely. This lack of realism in seeker behavior makes the missile system appear to be much more effective than it actually is. This and other modeling considerations are the subject of a further study of evasive tactics in a multiple missile environment. Finally, the important issue of counter measures has also been ignored. From a systems point of view, it is clear that the potential use of counter measures should be considered in a missile evasion study. However, for practical reasons, it was felt to be outside the potential scope of this present study.

3.4 Aircraft Evasion Strategies - Basic Considerations

3.4.1 Qualitative Aspects

In the analysis and synthesis of evasive maneuver strategies, it is useful to partition the problem into two phases. Phase I will be referred to as the Extra-End-Game (EEG) phase, and will denote that portion of the missile-aircraft engagement outside the end game envelope. Phase II will be referred to as the Intra-End-Game (IEG) phase, and will denote that portion of the missile-aircraft engagement inside the end game envelope. It is important to recognize that the end game envelope is not a totally precise notion. Qualitatively, we shall mean by the end game envelope that portion of the missile-aircraft engagement in which the missile is within 10 - 15 missile time constants of an interception or point of closest approach (FAAC, 1977). It is evident that the state space description of the end game envelope will depend on the aircraft maneuvers in the EEG phase of the engagement. This coupling between the end game envelope and the EEG phase maneuvering of the aircraft

is conceptually quite similar to the variations that exist between MLE's in the cases referred to as "target aware" versus "target unaware" conditions.

During the EEG phase, aircraft maneuvers initiated against a missile launched inside the target unaware MLE would generally cause the missile to maneuver in response to the changing orientation of the missile-aircraft line of sight vector. With significant "time-to-go" available (relative to the missile autopilot-airframe time constants) the missile would be expected to "follow" the aircraft's maneuvers. Hence, the goal of the aircraft would be to seek to maneuver in such a way as to:

- i) cause the missile to give up enough energy so as to "place the missile beyond its maximum range" (FAAC, 1977), or
- ii) cause the missile and aircraft to enter the end game envelope in a region of state space favorable to the aircraft, i.e., a region from which the aircraft could outmaneuver the missile based on considerations such as: missile autopilot-airframe time constants, relative turning radii, and seeker behavior.

It is important to note that, depending on initial conditions, there may be "blends" of strategies based on i) and ii) which are reasonable for the aircraft to use. In addition to these considerations, the multiple missile threat adds to the complexity of the decision environment by constraining the aircraft to avoid situations where an escape from one missile leads to a "setup" for a second missile.

3.4.2 Quantitative Methods

Earlier studies, e.g., (Poulter, 1975), (Shinar & Steinberg, 1976), and (Garnell & East, 1977) consider the application of optimal control techniques to determine aircraft evasion strategies. The work by (Poulter, 1975) addresses the determination of optimal maneuvers in the EEG phase of the problem. The computational aspects of the problem are fraught with difficulties which are typical of free terminal time nonlinear programming problems. The compu-

tational problem may exhibit local maxima and, hence, the determination of a global solution may be difficult to obtain, and, in any case, very time-consuming to achieve. In the context of the overall goals of this present study, there seems to be no reasonable possibility of attaining a real-time global solution to the optimal maneuver problem (against even a single missile) starting from the EEG phase.

If one seeks to optimize the maneuver starting from the IEG phase, by assuming that the terminal time is known a priori and that linearization is valid, one is led to a set of computationally tractable approximations, see for example, (Shinar & Steinberg, 1976) and (Garnell & East, 1977).

The focus of the present study is on the real-time determination of evasive maneuvers in a multiple missile environment, where the luxury of a single or one at a time end game analysis may not be available, i.e., the analysis and implementation of evasive maneuvering must begin in the EEG phase. This brings us to the introduction of the heuristic algorithms which have been developed for this purpose.

3.4.3 Geometric Aspects of Missile Guidance

Before we describe the heuristic algorithms, it is useful to review two well-known concepts in missile guidance "lore",

- i) the geometric-kinematic concept known variously as the Intercept, Maneuver, or Guidance Plane (FAAC, 1977) and (Garnell & East, 1977), and
- ii) the geometric-kinematic concept known as Anti-Proportional Navigation (TASC, 1977).

The Maneuver Plane:

The maneuver plane is defined to be that plane determined by the missile-aircraft los vector and the aircraft velocity vector.

"For a non-maneuvering target, the optimum (missile) heading angle lies in the plane defined by the target (aircraft) velocity vector and the los at launch. This plane is sometimes called the intercept or maneuver plane. It

is an estimate of this optimum heading that the fire control system usually computes and provides to the pilot for steering cues. It is this same optimum heading from which heading errors are measured. Since launches from other than optimum heading require the missile to maneuver more than it would if launched from the optimum, more missile energy is consumed and performance is affected." (FAAC, 1977; p. 20).

When the missile velocity vector lies in the maneuver plane (as defined above), the maneuver plane can be defined equivalently as that plane determined by the missile-aircraft los vector and the relative velocity vector of the missile and aircraft. We will employ this definition in the sequel.

Anti-Proportional Navigation:

The concept of anti-proportional navigation can be simply stated as that aircraft maneuver which instantaneously maximizes the missile-aircraft line of sight rate (TASC, 1977).

Let a_ψ and a_θ denote respectively the aircraft acceleration in pitch and yaw, as measured in missile-aircraft los coordinates (see Figure 3-2). Then, a simple exercise in calculus demonstrates that, the missile-aircraft los rate is maximized instantaneously (myopic maximization) when the aircraft acceleration vector has maximum magnitude and an orientation defined by:

$$(a_\psi/a_\theta) = (d\psi/dt)/(d\theta/dt)\cos(\psi),$$

$$a_r = 0,$$

and the signs of the components are chosen to increase the instantaneous los rate.

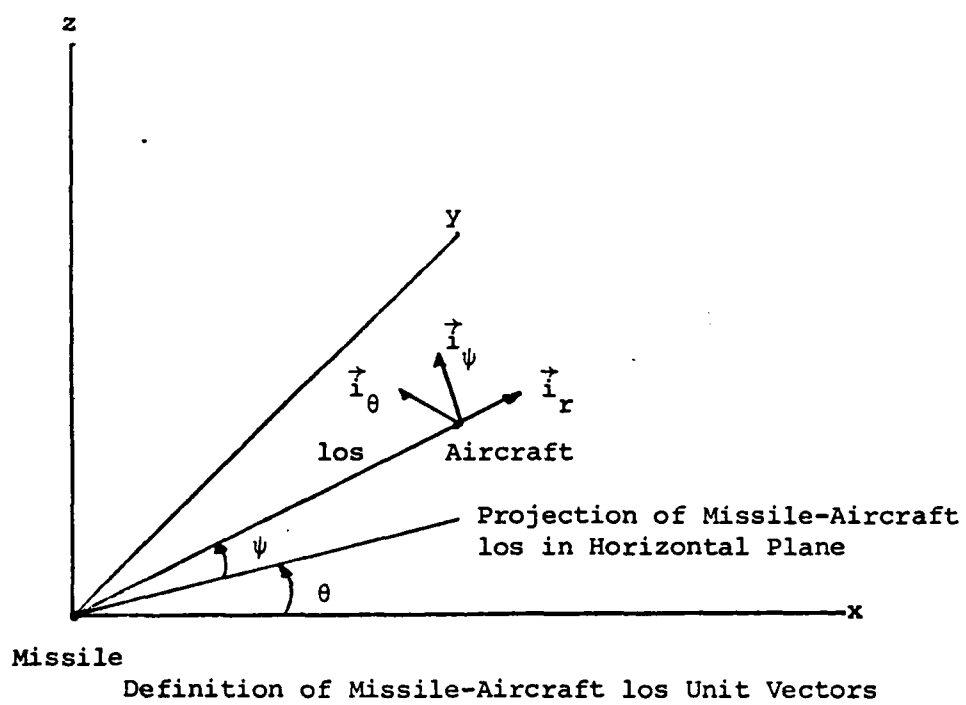
The Relationship Between Anti-Proportional Navigation and the Maneuver Plane:

The vector cross product $\vec{G} \triangleq \vec{l}_{os} \times \vec{v}_{rel}$ defines a normal vector to the maneuver plane, which is expressible in missile-aircraft los coordinates by:

$$\vec{G} = (r d\psi/dt) \vec{i}_\theta + (-r \cos(\psi) d\theta/dt) \vec{i}_\psi,$$

where $\vec{i}_\psi$ and $\vec{i}_\theta$ denote the unit vectors (in los coordinates) in the pitch and yaw directions respectively. We observe that the ratio of the components

Figure 3-2



(a_ψ/a_θ) of $\vec{G}$ is the negative reciprocal of the ratio (a_ψ/a_θ) formed by the components of the anti-proportional acceleration vector defined previously. Hence, the anti-proportional acceleration vector lies in the maneuver plane.

3.5 Heuristic Algorithms

Herein we describe four heuristic algorithms for countering a multiple missile threat. The algorithms are defined for the case of $N = 2$ missiles; however, the respective extensions to the general case ($N > 2$) is essentially trivial in each instance. To simplify the following discussion, we denote the algorithms by: Algorithm 1, ..., Algorithm 4.

Let $\vec{G}_i$ ($i=1,2$) denote the unit normal vector associated with the maneuver plane of missile i . Let $\vec{\Gamma}_i$ ($i=1,2$) denote the unit vector associated with the commanded (desired) direction defined by the anti-proportional navigation rule associated with missile i . We note that in los coordinates, the ratio of pitch to yaw components of $\vec{G}_i$ is the negative reciprocal of the ratio of pitch to yaw components of $\vec{\Gamma}_i$, $i = 1, 2$. Algorithms 1 & 2 are defined in terms of the maneuver plane unit normals $\vec{G}_1$ & $\vec{G}_2$; whereas, Algorithms 3 & 4 are defined in terms of the anti-proportional navigation unit vectors $\vec{\Gamma}_1$ & $\vec{\Gamma}_2$. Finally, let $\vec{A}(\mu)$ denote the instantaneous total acceleration vector of the aircraft. The orientation of the aircraft total acceleration vector is a function of the aircraft bank angle μ , as well as the angle of attack and the velocity vector. The magnitude of $\vec{A}(\mu)$ depends on the instantaneous aircraft drag, lift, and thrust forces, as well as the force of gravity.

Algorithm 1:

Define $J_i(\mu)$ by,

$$J_i(\mu) = \vec{G}_i \cdot \vec{A}(\mu), \quad i = 1, 2.$$

Define $P(\mu, \lambda)$ by,

$$P(\mu, \lambda) = |\lambda J_1(\mu) + (1-\lambda) J_2(\mu)|,$$

where $-\pi \leq \mu \leq \pi$ rad, and $0 \leq \lambda \leq 1$.

Algorithm 1 dictates that at each decision instant a pair (μ^*, λ^*) be determined such that:

$$P(\mu^*, \lambda^*) = \min_{\lambda} \max_{\mu} P(\mu, \lambda) = \max_{\mu} \min_{\lambda} P(\mu, \lambda),$$

until a closure rate negative (CRN) condition is achieved for one of the missiles. Then, μ^* is determined by:

$$\mu^* = \arg \max |J_i(\mu)|,$$

where the i^{th} missile is assumed to be in a closure rate positive (CRP) condition.

Here we assume that a saddle point pair (μ^*, λ^*) for $P(\mu, \lambda)$ exists and refer the reader to Chapter 5 of this report to review relevant related saddle point existence results.

The heuristic of Algorithm 1 requires, in effect, the myopic determination of an instantaneous saddle point pair. This saddle point has the following interpretation: The instantaneous bank angle μ^* is precisely that bank angle which maximizes the minimum value of the absolute value of the projection of the aircraft acceleration vector on a "synthetic" maneuver plane normal $\vec{G}(\lambda)$, obtained by forming a linear combination (convex combination) of the individual maneuver plane normals:

$$\vec{G}(\lambda) = \lambda \vec{G}_1 + (1-\lambda) \vec{G}_2,$$

i.e.,

$$P(\mu, \lambda) = |\vec{G}(\lambda) \cdot \vec{A}(\mu)|.$$

We note that

$$P(\mu, 1) = |\vec{G}_1 \cdot \vec{A}(\mu)|, \text{ and}$$

$$P(\mu, 0) = |\vec{G}_2 \cdot \vec{A}(\mu)|.$$

Thus, the value of μ which maximizes $P(\mu, 1)$, is that value of μ which instantaneously maximizes the absolute value of the projection of $\vec{A}(\mu)$ on $\vec{G}_1$, i.e., that value of μ which maximizes the aircraft acceleration normal to the

maneuver plane associated with missile 1.

The heuristic of Algorithm 1 provides a degree of weighting or importance λ^* and $(1-\lambda^*)$ to be associated with the maneuver plane of the respective missile threats. As the weighting factor λ^* varies from one to zero in value, the weight associated with the maneuver plane of missile 1, λ^* , is decreased, and the weight associated with the maneuver plane of missile 2, $(1-\lambda^*)$, is increased accordingly.

In summary, the heuristic of Algorithm 1 allows the amount of aircraft acceleration which is available at a given decision instant to be apportioned in such a way that a balance is achieved in "attempting" to simultaneously maximize the aircraft acceleration normal to each maneuver plane.

Algorithm 2:

Based on the notation introduced in the presentation of Algorithm 1, Algorithm 2 dictates that at each decision instant μ^* be determined by:

$$\mu^* = \arg \max (\min(|J_1(\mu)|, |J_2(\mu)|)),$$

until a CRN condition is achieved for one of the missiles. Then, μ^* is determined by:

$$\mu^* = \arg \max |J_i(\mu)|,$$

where the i^{th} missile is assumed to be in a CRP condition.

The heuristic of Algorithm 2 requires, in effect, the myopic determination of a bank angle μ^* which maximizes the minimum (over all missiles) of the absolute value of the projection of the instantaneous aircraft acceleration on the normal vector to the missile maneuver plane. This is equivalent to the following max min problem:

$$\max_{\mu} \min_{\lambda} (\lambda |J_1(\mu)| + (1-\lambda) |J_2(\mu)|),$$

where $-\pi \leq \mu \leq \pi$ rad, and $\lambda \in \{0,1\}$.

The heuristic of Algorithm 2 differs from that of Algorithm 1 in that

Algorithm 2 is "binary" and allows no smooth blending of objectives by the assignment of an intermediate degree of importance or weight to each missile threat in the objective function.

In summary, under Algorithm 2, the full attention of the aircraft is captured at each decision instant by just one of the two missiles when both missiles are closing on the aircraft.

Algorithm 3:

Algorithm 3 is the dual of Algorithm 1 with $\vec{G}_1$ replaced by $\vec{\Gamma}_1$ and $\vec{G}_2$ replaced by $\vec{\Gamma}_2$. Here, the objective is to seek a trade-off between the line of sight rate associated with each missile, as opposed to the aircraft acceleration normal to the individual maneuver planes.

Algorithm 4:

Algorithm 4 is the dual of Algorithm 2 with $\vec{G}_1$ replaced by $\vec{\Gamma}_1$ and $\vec{G}_2$ replaced by $\vec{\Gamma}_2$. The interpretation of the heuristic associated with Algorithm 4 is similar to that of Algorithm 2, with the objective now being line of sight rate maximization, as opposed to "maximizing" aircraft acceleration normal to the individual maneuver planes.

3.6 Information Requirements

One of the goals of this present study was the development of a real-time algorithm with relatively minimal requirements in terms of dynamic data and the description of missile system dynamics. It is important to note that all four heuristic algorithms described in this study require only information delineating: the relative velocity between each missile and the aircraft, and each missile-aircraft los. No a priori knowledge of the missiles' dynamics or control systems is required. This is not to say, however, that additional information would be useless, since a more complete description of each missile might permit an approximate faster than real-time flyout simulation to be run in parallel with the heuristic evasive maneuver decision model to enhance the decision making process.

4.0 Computer Simulation Models for Determining and Evaluating the Global and Local Characteristics of Evasive Maneuvers

4.1 Introduction

In this chapter we describe the details of the simulation models based on the four heuristic algorithms delineated in Chapter 3, and we present and analyze the simulation results pertaining to the exercising of these four algorithms against seven representative missile-aircraft engagement scenarios. The related FORTRAN IV source programs are contained in Appendix A of this report.

4.2 The Simulation Paradigm and Engagement Scenarios

4.2.1 The Simulation Paradigm

The four simulation models developed for use in this study are flyout simulations for evaluating and comparing the four heuristic evasion algorithms defined in Chapter 3. The individual programs are referred to as ACDYN.91, ..., ACDYN.94, and correspond respectively to Algorithms 1 - 4. The programs ACDYN.91 - ACDYN.94 have many features in common. Hence, program ACDYN.91 will be described in detail, and the relevant differences which characterize 92 - 94 will be provided to complete the program package. The source program for ACDYN.91 is complete with the exception of certain nonessential library graphics routines for producing plots. Flowcharts for programs 91 & 92 are included in this chapter. Programs 91 & 93 are virtually identical -- the sole exception pertains to the remarks in Section 3.5 (p. 3-20) in reference to the vectors $\vec{G}_i$ and $\vec{f}_i$, $i = 1, 2$. This same statement holds as well for programs 92 & 94.

The flyout simulations are based on the dynamical models of the F-4 aircraft and AIM-9 missile which are delineated in Chapter 3. All missile-aircraft engagement scenarios considered in this study exhibit the following salient characteristics:

- i) The missiles are assumed to be coasting (thrust equals zero) when the evasive tactics are initiated.
- ii) Without evasive maneuvering, all missile-aircraft engagements would result in a first missile kill within 4 - 8 seconds from the opening of the engagement scenarios.
- iii) The integration time step is defined by

$$\text{STEP} = \min(T, 0.5 \cdot \text{DSV} / \text{VREL}),$$
 where: DSV denotes the current separation distance between the closest missile and the aircraft; VREL denotes the magnitude of the relative velocity between the closest missile and the aircraft; and T equals 0.1 seconds.
- iv) The aircraft begins the maneuver process dictated by the relevant algorithm at the time instant $t = 0.1$ sec after the opening of the engagement scenario. Hence, a detailed study of maneuver initiation timing was not carried out as a part of this present study.
- v) During the maneuver process the aircraft afterburner is on, and the aircraft load factor is 8 g, or the aircraft is limited to a maximum AOA of 25 degrees.
- vi) The aircraft is roll rate limited to 600 deg/sec. Hence, if the commanded bank angle based on the decision algorithm is greater than $600 \cdot \text{STEP}$ degrees from the present roll position, the commanded roll increment is limited to $600 \cdot \text{STEP}$.

Flowcharts for ACDYN.91 & ACDYN.92

The flowcharts for programs ACDYN.91 and ACDYN.92 are included in Figures 4-1(a - h). Figure 4-1a depicts the overall organization of the generic ACDYN simulation program. The first two subroutines, INIT and VALUE, define the heart of the program (refer to Figures 4-1(a - f)). Subroutine INIT (Figure 4-1b) allows the user to either interactively initialize an engagement scenario, or read and update a file with such information. Subroutine VALUE (Figures 4-1(c - f)) provides the necessary flyout simulation logic and report generation. The variations between programs ACDYN.91 - 94 occur in subroutine VALUE, specifically, within the program module for determining the values of μ^* and λ^* (see Figures 4-1(c, e, & f)). Figures 4-1(e & f) flowchart the determination of the pair (μ^*, λ^*) within ACDYN.91 & 92 respectively. The generic functional notation $J(\mu, \lambda)$, which appears in the caption of Figures 4-1(e & f),

refers to the relevant game theoretic kernel functions defined explicitly in heuristic Algorithms 1 & 2 (Section 3.5, pp. 3-17 & 3-19). The remainder of the program, subroutines INTBOX & PLOUT (Figures 4-1(g & h)), provide the necessary integration capability and the calls to the library plotting routines.

4.2.2 The Engagement Scenarios

The performance of the heuristic algorithms was studied by means of comparative scenario analysis. The four heuristic algorithms were exercised against seven representative missile-aircraft engagement scenarios. The initial conditions at the opening of each scenario are summarized in Table 4-1. The x-y position and horizontal velocity of the missiles and aircraft in each scenario are depicted vectorially in Figures 4-2(a - c). The engagements were chosen to depict a combination of missile threats which ranged through mixtures of tail-chase, head-chase, and off-beam launch conditions. The engagements included scenarios (1 - 3) where one missile is above the aircraft initially and the second missile is launched from below, as well as coplanar initial conditions (5 & 7), and cases (4 & 6) where both missile are launched from below the aircraft. The engagements all lead to multiple hits if no aircraft maneuvering occurs.

4.3 Algorithm Performance

Summary and Ranking

The performance of Algorithms 1 - 4 parameterized by scenario and individual missile identification is summarized in Table 4-2a. The detailed results of each of nine scenario/algorithm pairs in which the aircraft achieved multiple misses are contained in Tables 4-3 through 4-11, and Figures 4-3 through 4-11. The definitions of program variables which appear in Tables 4-3 through 4-11 are contained in Table 4-2b.

In order to analyze and compare the performance (Table 4-2a) of the four

heuristic algorithms against the seven multiple missile threat scenarios defined in Table 4-1, it is necessary to adopt a suitable performance measure. A game theoretic approach suggests that the minimum of the miss distance, associated with missiles 1 & 2, be adopted as a measure of performance for each algorithm-scenario pair. Based on this approach, we observe that there is no single algorithm whose performance dominates the remaining three algorithms on a scenario by scenario basis. If, in addition, we calculate the number of multiple misses, we observe that the algorithms in decreasing order of performance are: Algorithms 1, 4, 2, & 3.

Detailed Simulation Output

Tables 4-3 through 4-11 contain the initial and final state information associated with the nine successful (multiple miss) algorithm-scenario pairs contained in Table 4-2a. In addition, these nine tables contain the minimum miss distance information collected as part of the overall simulation report generation. Figures 4-3 through 4-11 are multiple figures, e.g., Figure 4-3 contains Figures 4-3a, ..., 4-3e. Figures associated with Algorithm 1 contain five graphs (a - e) which depict: (a) Aircraft Bank Angle μ^* vs time; (b) $\lambda^* \times 100$ vs time; (c) the Projection of the Aircraft and Missile Trajectories in the (x-y) Plane; (d) the Projection of the Aircraft and Missile Trajectories in the (z-x) Plane; (e) the Projection of the Aircraft and Missile Trajectories in the (z-y) Plane. Figures associated with Algorithms 2 & 4 contains four graphs (a - d) which depict: (a) Aircraft Bank Angle vs time and λ^* vs time (the symbol "F" denotes λ^* , where an "F - value" of 50 denotes $\lambda^* = 1$, and an "F - value" of -50 denotes $\lambda^* = 0$); (b) - (d) depict respectively the projections of the aircraft and missile trajectories in the (x-y), (z-x), and (z-y) planes.

Figure 4-1a
ACDYN Flowchart

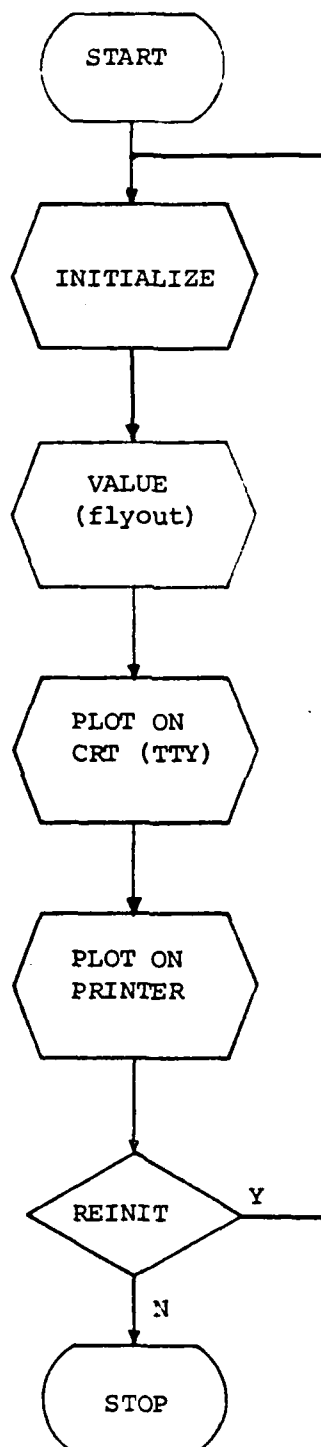


Figure 4-1b
Subroutine INIT
(Initialization)

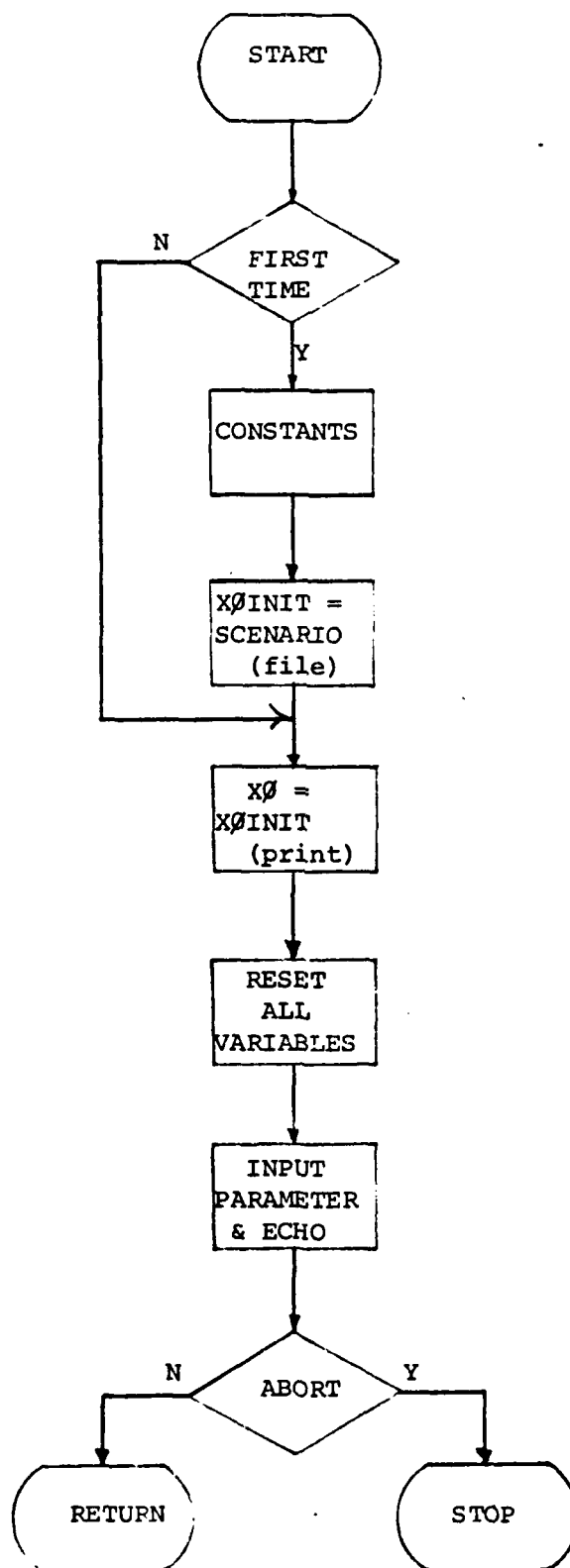


Figure 4-1c
 Subroutine VALUE
 (flyout simulation)

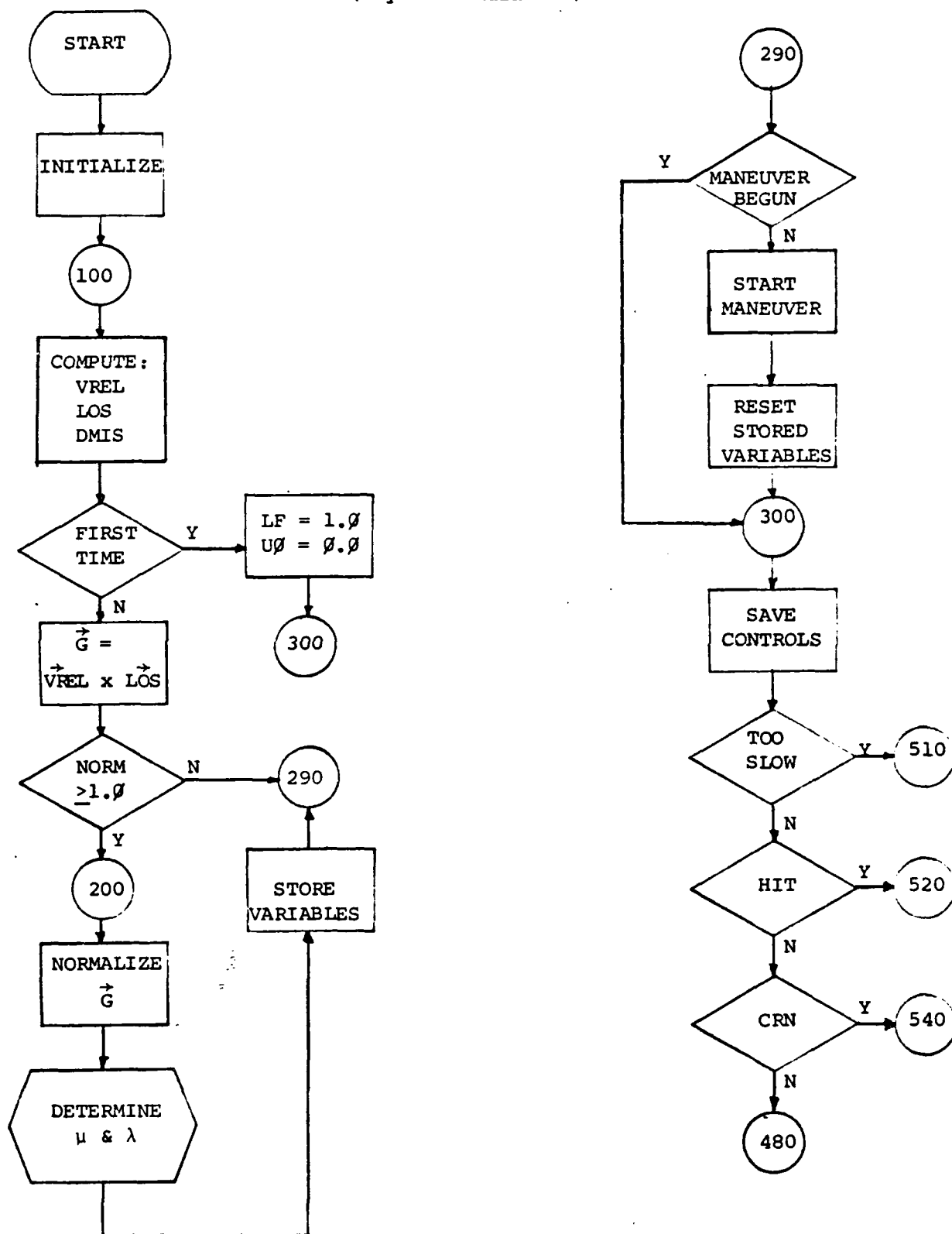
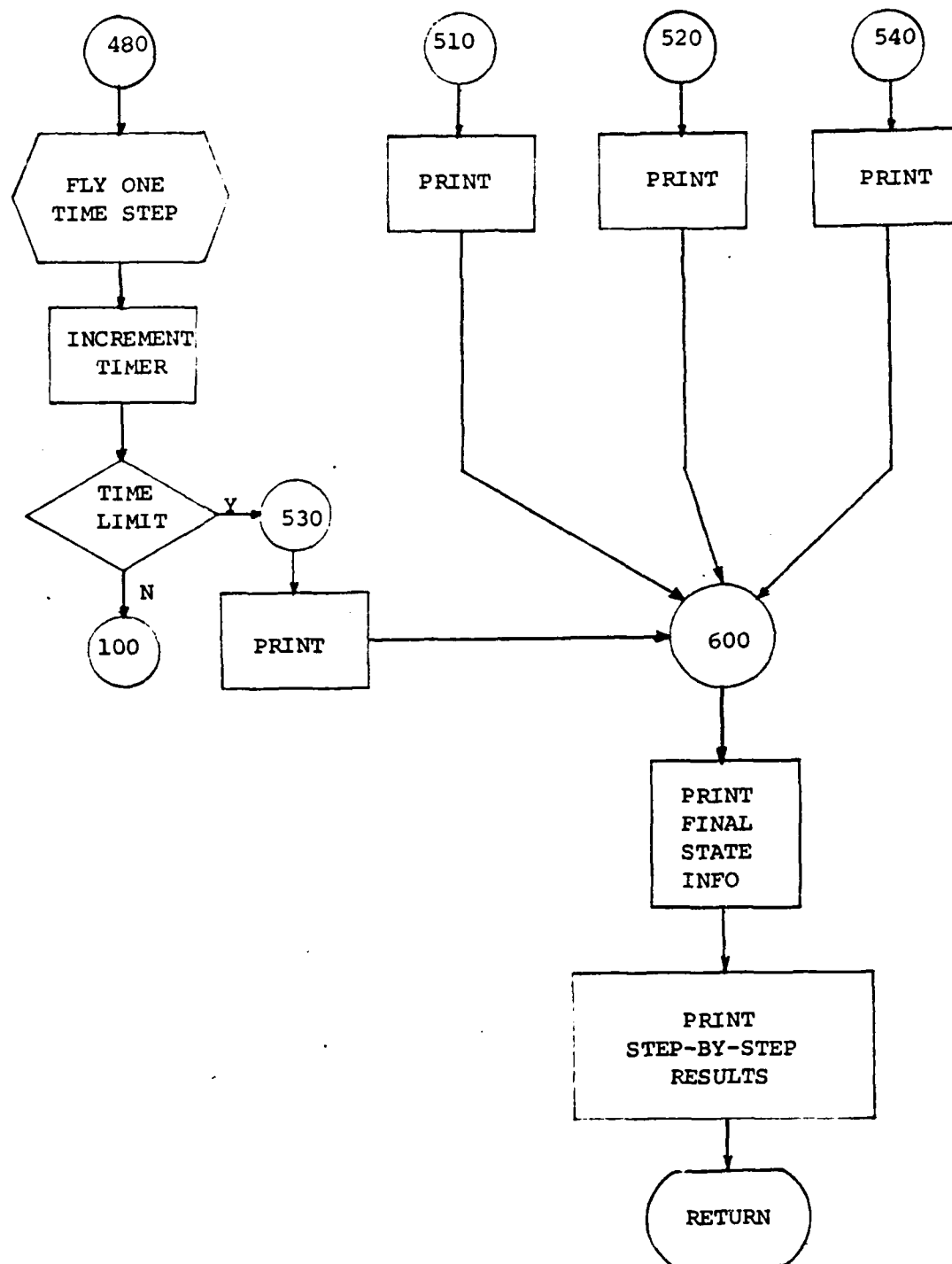
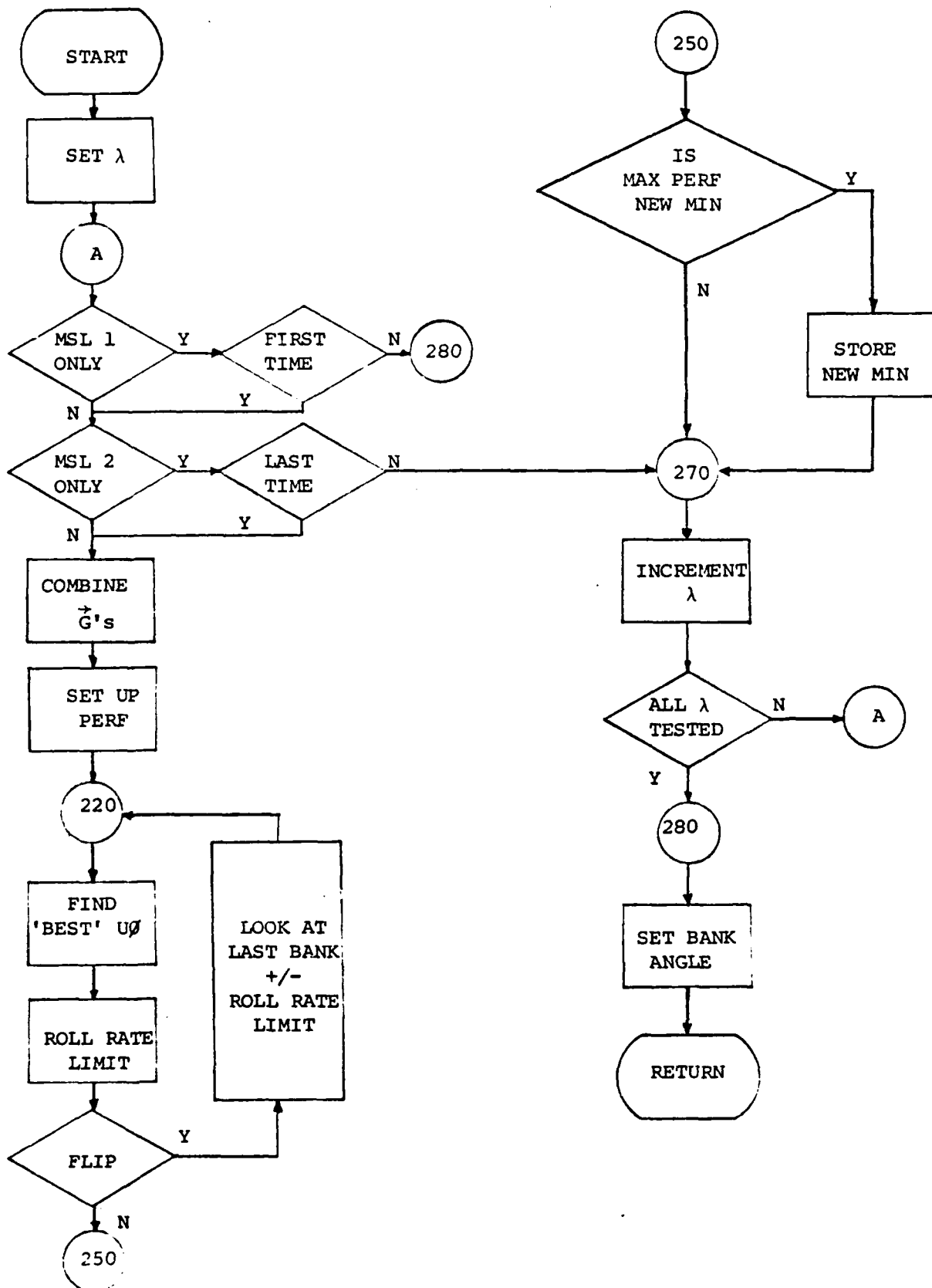


Figure 4-1d
Continuation of
Subroutine VALUE



Program Module within Subroutine VALUE to determine λ & μ via:

$$\min_{\lambda} \max_{\mu} J(\mu, \lambda) \text{ ACDYN.91 (Algorithm 1)}$$



Program Module within Subroutine VALUE to determine λ & μ via:

$\max_{\mu} \min_{\lambda} J(\mu, \lambda)$ ACDYN.92 (Algorithm 2)

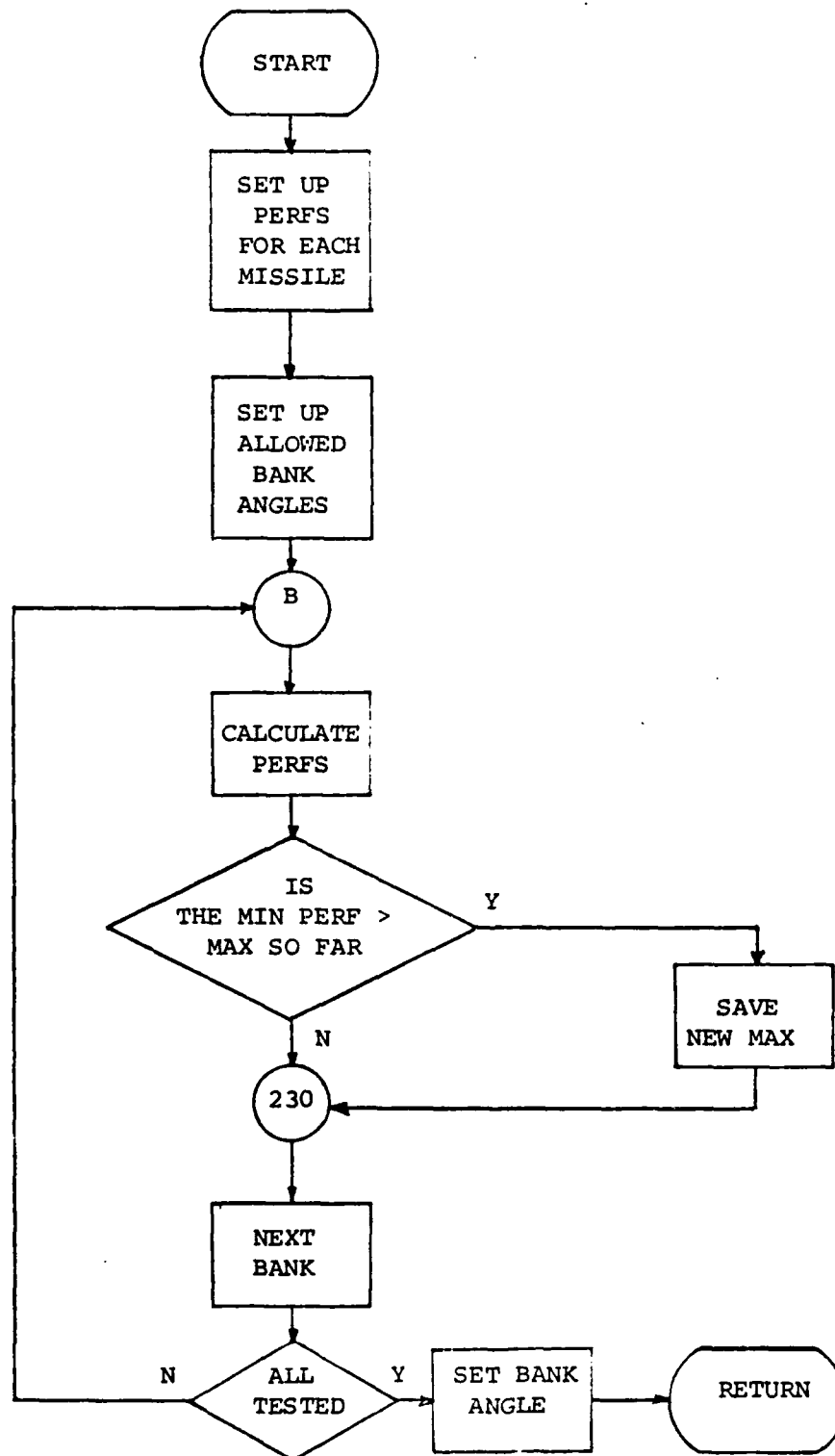
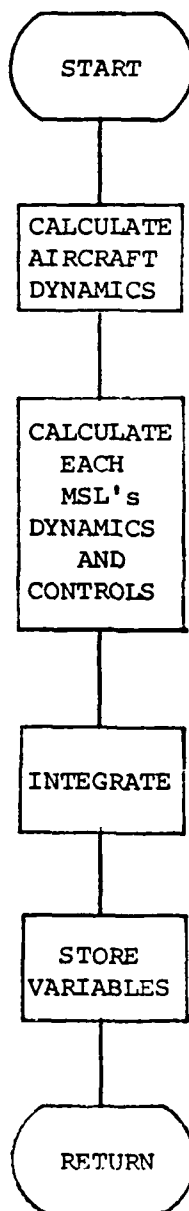


Figure 4-1g

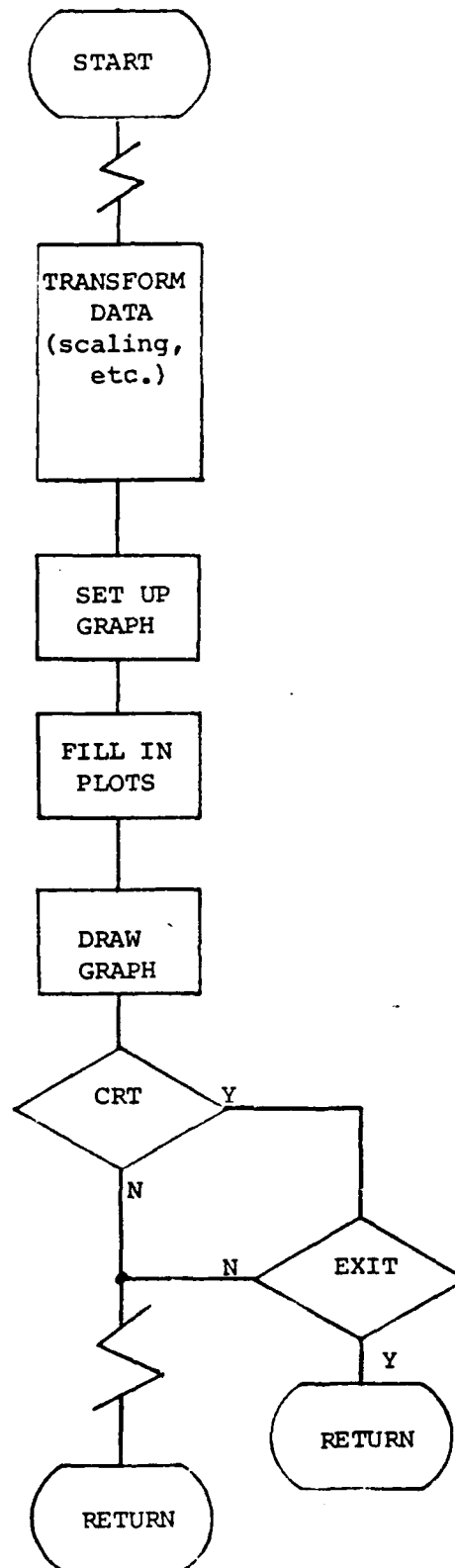
Subroutine INTBOX
(fly one time step)



Subroutine PLOUT

(CRT or PRINTER)

(typical use, one set per graph)



Summary of the Initial Conditions of Aircraft & Missiles
at the Beginning of Each Scenario

Scenario #	Vehicle Type:	x ft	y ft	z ft	v ft/sec	γ deg	σ deg
1	A/C	11258	0	30000	1100	0	0
1	MSL 1	11258	14000	33000	3150	-8	-60
1	MSL 2	28578	10000	27000	3450	8	-150
2	A/C	11258	0	30000	1100	0	0
2	MSL 1	11258	15000	33000	3150	-8	-60
2	MSL 2	28578	-10000	27000	3450	8	150
3	A/C	11258	0	30000	1100	0	0
3	MSL 1	0	6500	33000	3150	-8	-30
3	MSL 2	28578	10000	27000	3450	8	-150
4	A/C	11258	0	30000	1100	0	0
4	MSL 1	28258	0	33000	3300	-8	180
4	MSL 2	0	6500	33000	3300	-8	-30
5	A/C	15000	0	30000	1100	0	0
5	MSL 1	3000	0	30000	3300	0	0
5	MSL 2	15000	12000	30000	3300	0	-60
6	A/C	13000	0	30000	1100	0	0
6	MSL 1	30000	0	33000	3300	-8	180
6	MSL 2	0	0	33000	3300	-8	0
7	A/C	0	0	30000	1000	0	45
7	MSL 1	15000	0	30000	3300	0	180
7	MSL 2	0	15000	30000	3300	0	-90

Table 4-1

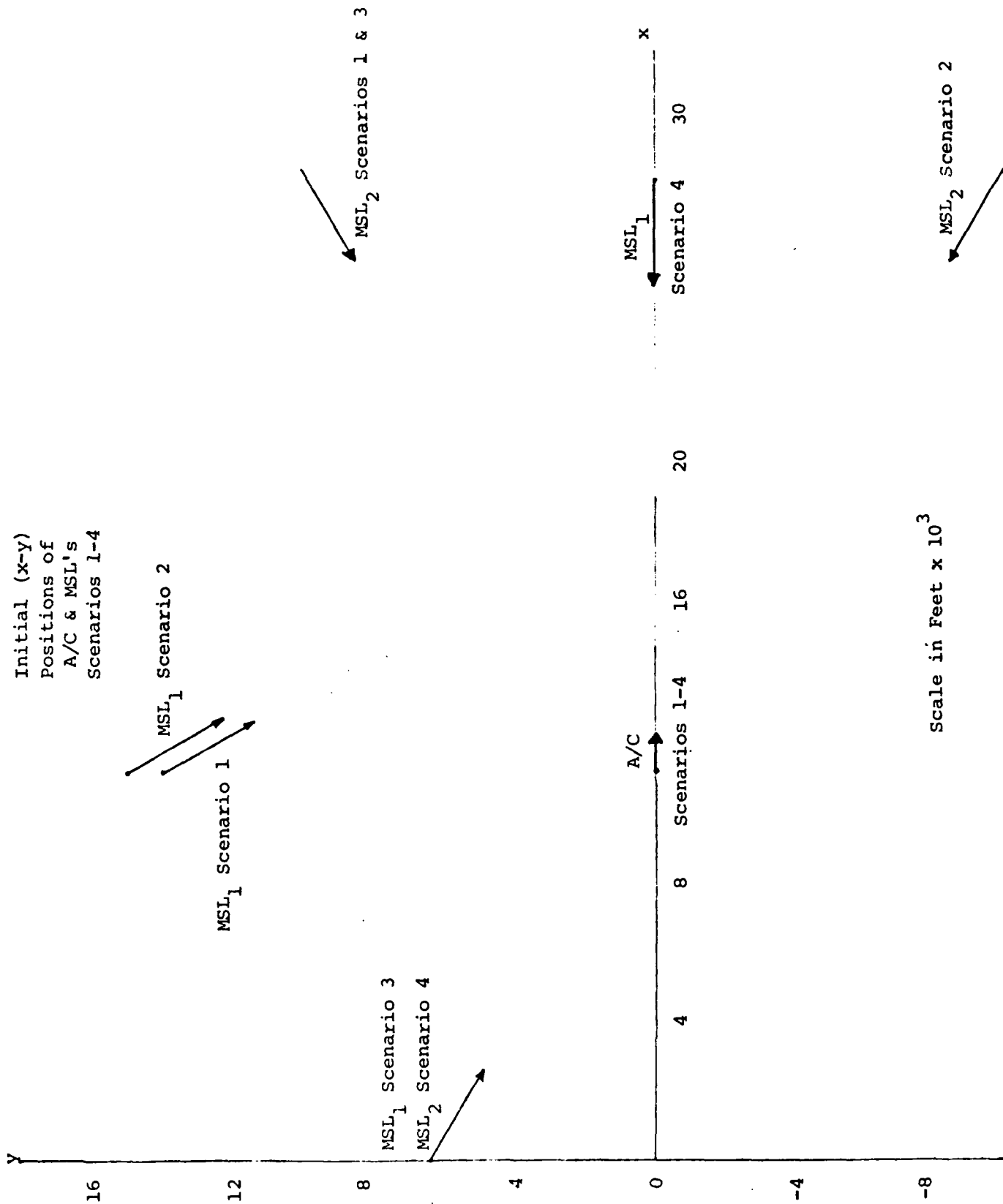


Figure 4-2a

Initial (x-y)
Positions of
A/C & MSL's
Scenarios 5-6

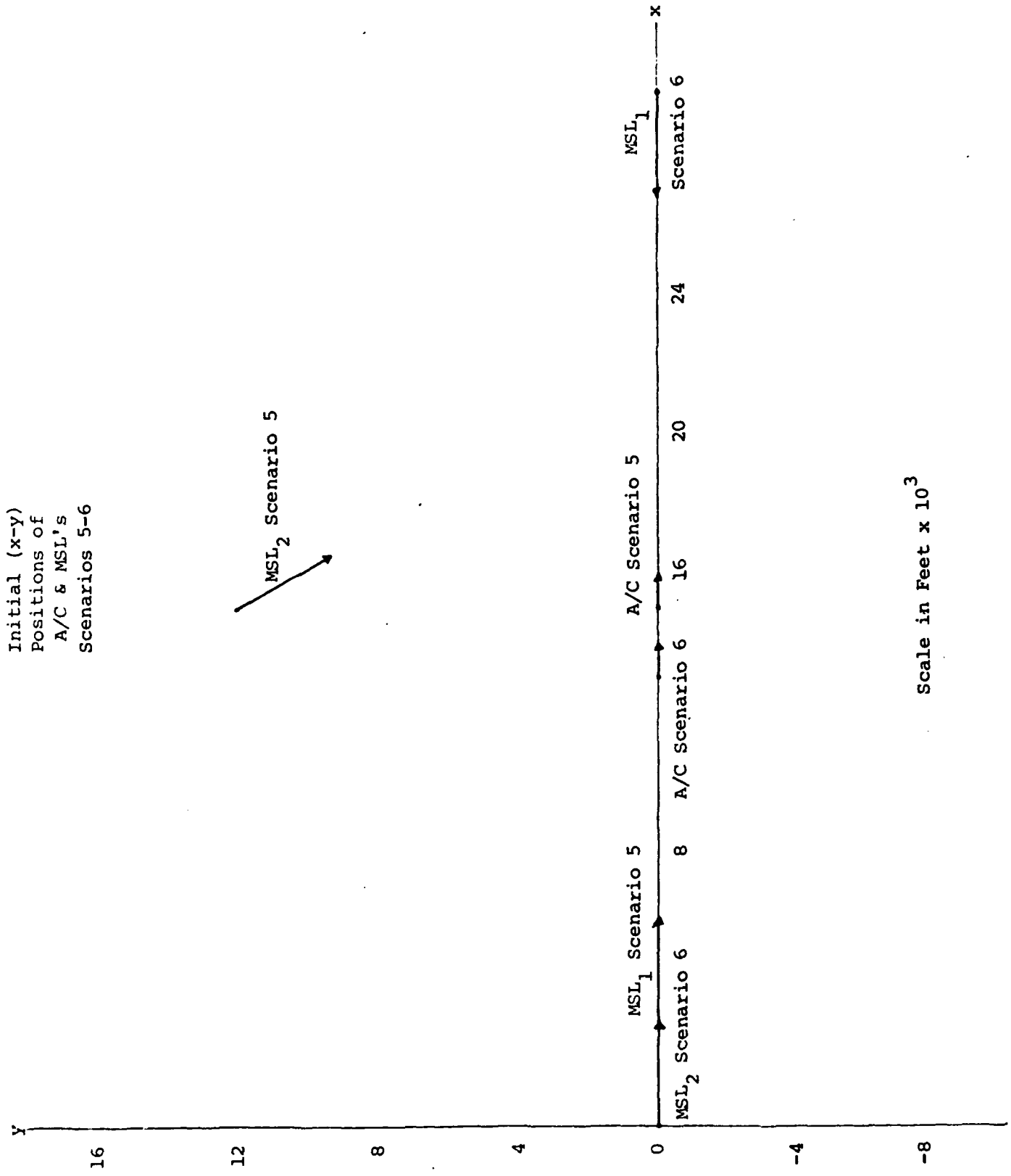
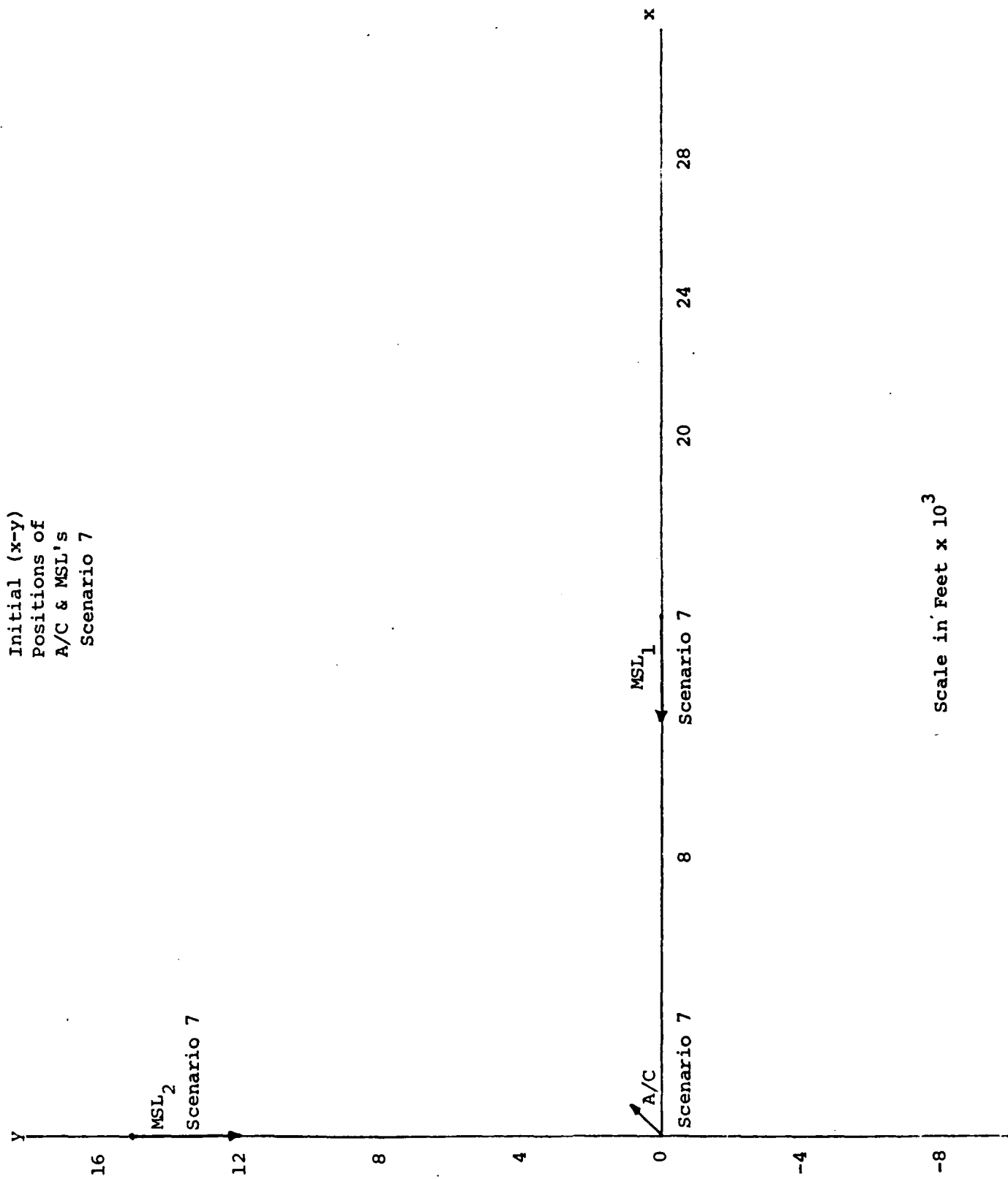


Figure 4-2b

Initial (x-y)
Positions of
A/C & MSL's
Scenario 7



Scale in Feet $\times 10^3$

Figure 4-2c

Table 4-2a

Summary of Algorithm Performance
(Miss Distance)

Scenario #	MSL #	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
1	1	32	18	-	-
1	2	37	61	H	H
2	1	40	H	H	-
2	2	73	-	-	H
3	1	T	-	H	17
3	2	25	H	30	69
4	1	H	H	19	21
4	2	-	-	H	23
5	1	29	17	H	-
5	2	53	52	39	H
6	1	56	60	H	H
6	2	22	H	-	-
7	1	64	H	H	158
7	2	H	25	-	268

Notes: a) Miss distance rounded to nearest foot.

b) H denotes hit, i.e., miss distance $\leq$ 15 feet.

c) - denotes untabulated miss or second hit.

d) T denotes occurrence of time limit in simulation.

Definition of Dynamical Variables & Parameters Appearing in Tables 4-3 Through 4-11

State Variables	A/C	MSL ₁	MSL ₂
x	X0(1)	X0(7)	X0(13)
y	X0(2)	X0(8)	X0(14)
z	X0(3)	X0(9)	X0(15)
v	X0(4)	X0(10)	X0(16)
γ	X0(5)	X0(11)	X0(17)
σ	X0(6)	X0(12)	X0(18)

Table 4-2b

Other Variables/Parameters

Name:	Definition:
DSEP1	Current Distance Between Aircraft and Missile 1
DSEP2	Current Distance Between Aircraft and Missile 2
DMIS	Distance of Closest Approach
TSTEP	Maximum Integration Time Step (see p. 4-2, TSTEP Δ T)
TAU	Autopilot-Airframe Time Constant for Missiles 1 & 2
RK11	Proportional Navigation Gain (Pitch) for Missiles 1 & 2
RK12	Proportional Navigation Gain (Yaw) for Missiles 1 & 2

Table 4-3
Algorithm 1, Scenario 1

ISTEP = 0.1000, TAU = 0.1500

INIT XU(1) = 11258.0	INIT XU(7) = 11258.0	INIT XU(13) = 28578.0
INIT XU(2) = 0.00000	INIT XU(8) = 14000.0	INIT XU(14) = 10000.0
INIT XU(3) = 30000.0	INIT XU(9) = 33000.0	INIT XU(15) = 27000.0
INIT XU(4) = 1100.00	INIT XU(10) = 3150.00	INIT XU(16) = 3450.00
INIT XU(5) = 0.00000	INIT XU(11) = -7.99999	INIT XU(17) = 7.99999
INIT XU(6) = 0.00000	INIT XU(12) = -60.0000	INIT XU(18) = -150.000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50, YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 0.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.143E 05	DSEP2 = 0.202E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.116E 05	DSEP2 = 0.160E 05
TIME = 2.000	DSEP1 = 0.924E 04	DSEP2 = 0.123E 05
TIME = 3.000	DSEP1 = 0.706E 04	DSEP2 = 0.908E 04
TIME = 4.000	DSEP1 = 0.473E 04	DSEP2 = 0.595E 04
TIME = 5.000	DSEP1 = 0.230E 04	DSEP2 = 0.286E 04
TIME = 5.924	DSEP1 = 113.	DSEP2 = 116.

*** CLOSURE RATE NEGATIVE AT TIME = 5.977***

TA1 : BEST DSEP = 32.3770	, NOW = 38.5233
TA2 : BEST DSEP = 37.3531	, NOW = 57.9274

XU(1): 0.1672E 05	XU(7): 0.1673E 05	XU(13): 0.1671E 05
XU(2): -505.0	XU(8): -536.0	XU(14): -550.5
XU(3): -0.3047E 05	XU(9): 0.3049E 05	XU(15): 0.3050E 05
XU(4): 752.5	XU(10): 2202.	XU(16): 2175.
XU(5): -1.645	XU(11): -14.79	XU(17): 14.34
XU(6): 24.45	XU(12): -76.61	XU(18): -146.6

DELX1: -3.84	DELY1: 26.0
DELT1: -24.9	DMIS1: 38.5
BEST DMIS WAS 32.4	

DELA2: 19.3	DELY2: 42.5
DELT2: -34.3	DMIS2: 57.9
BEST DMIS WAS 37.4	

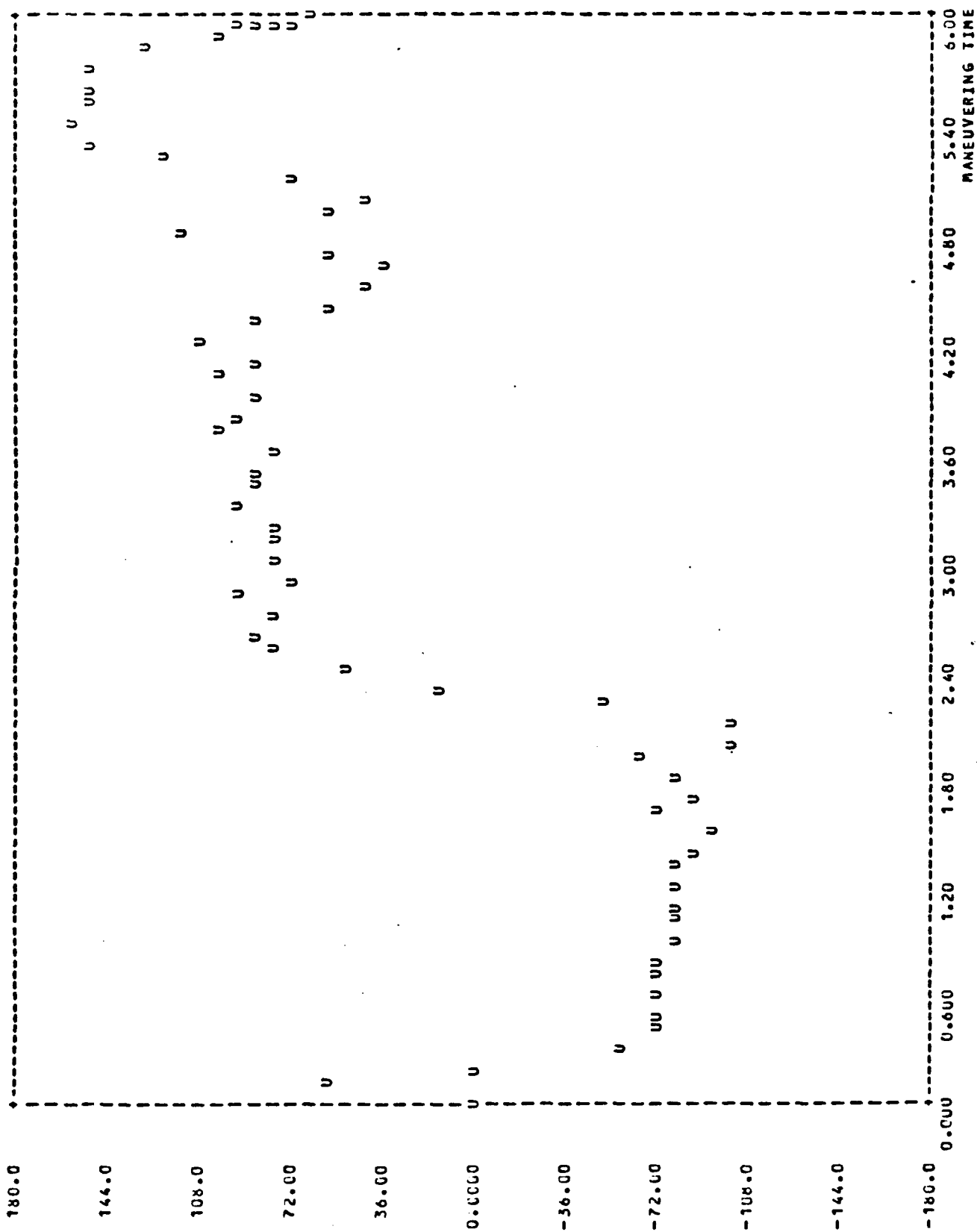


Figure 4-3a
Algorithm 1, Scenario 1

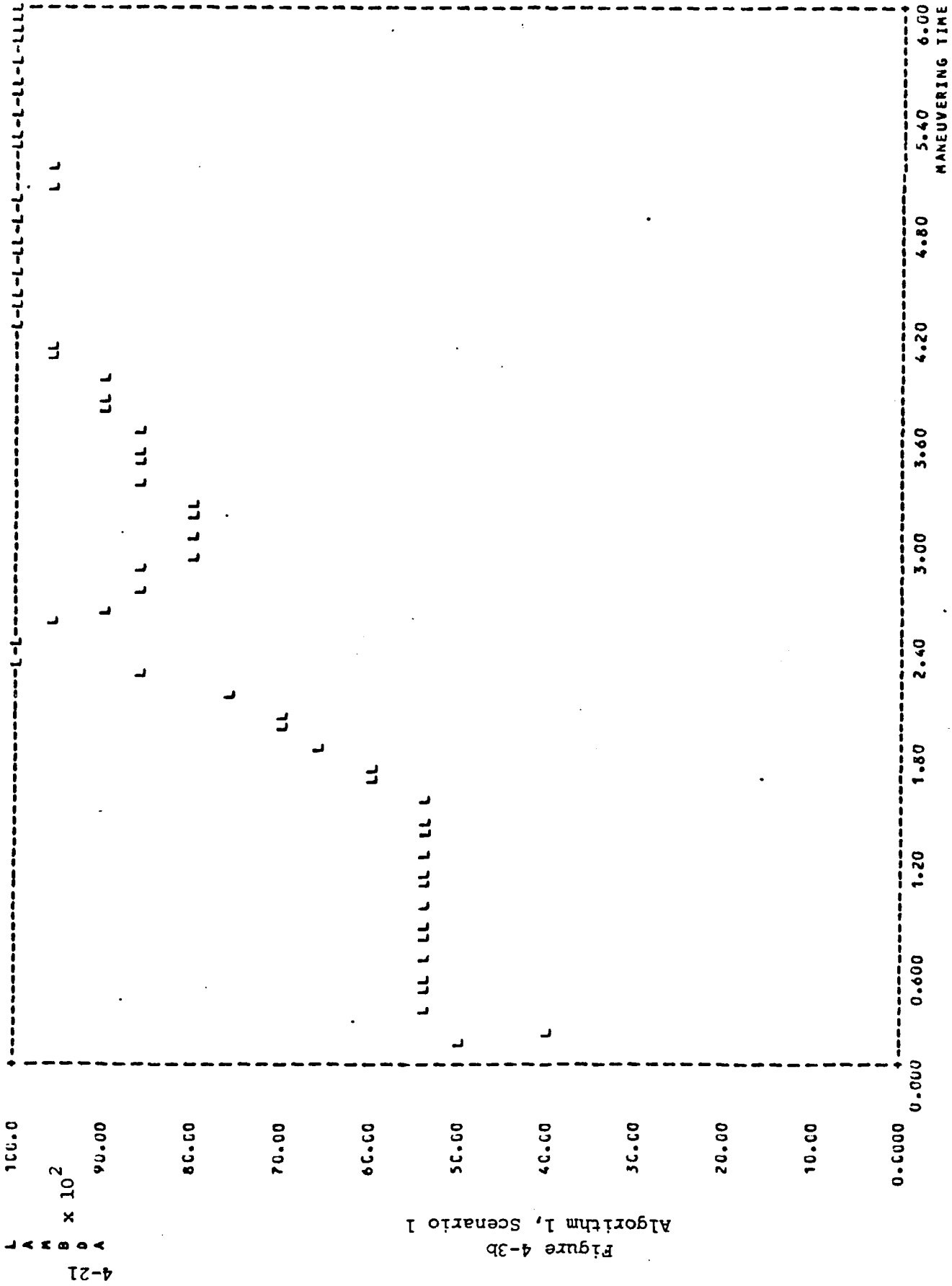
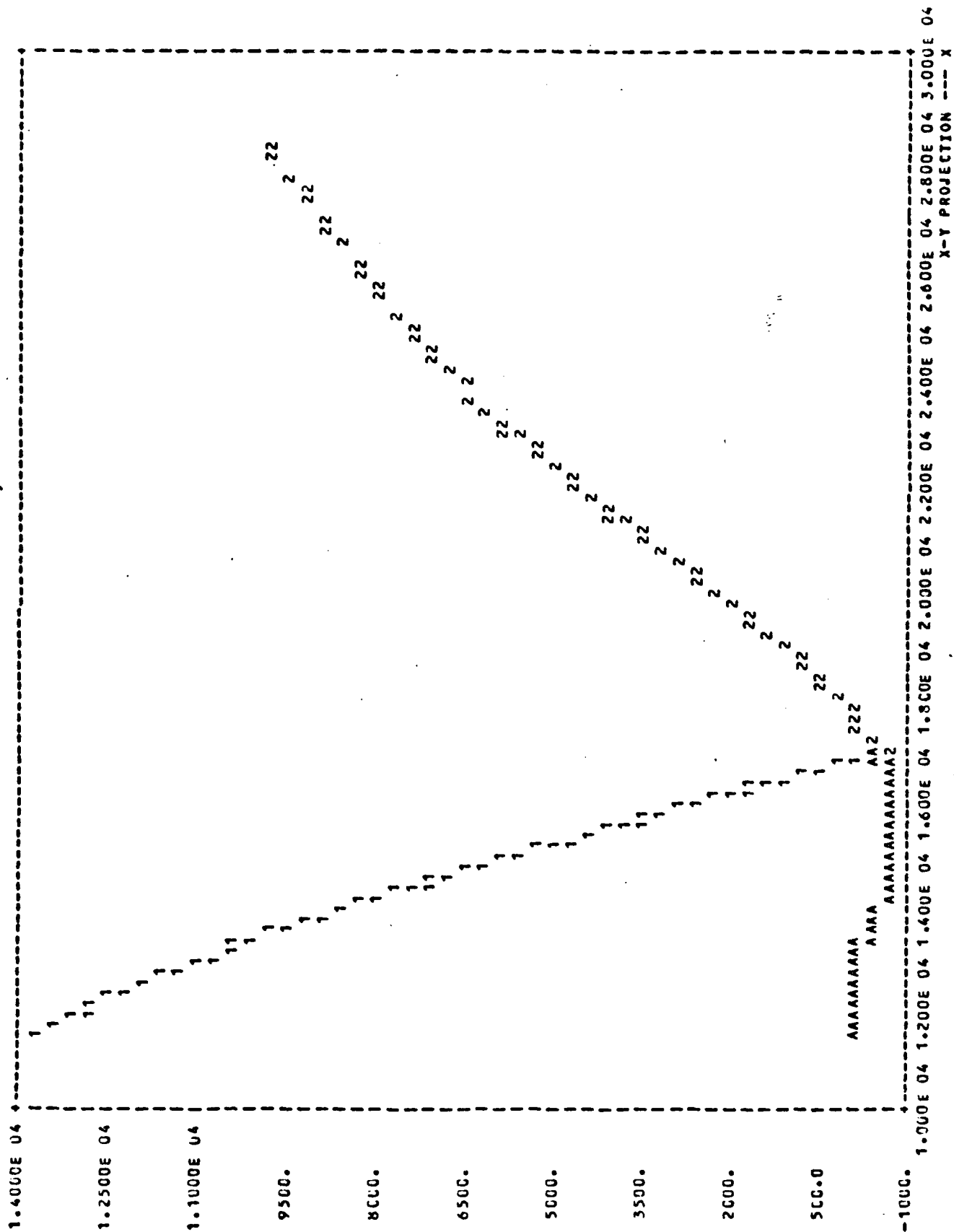
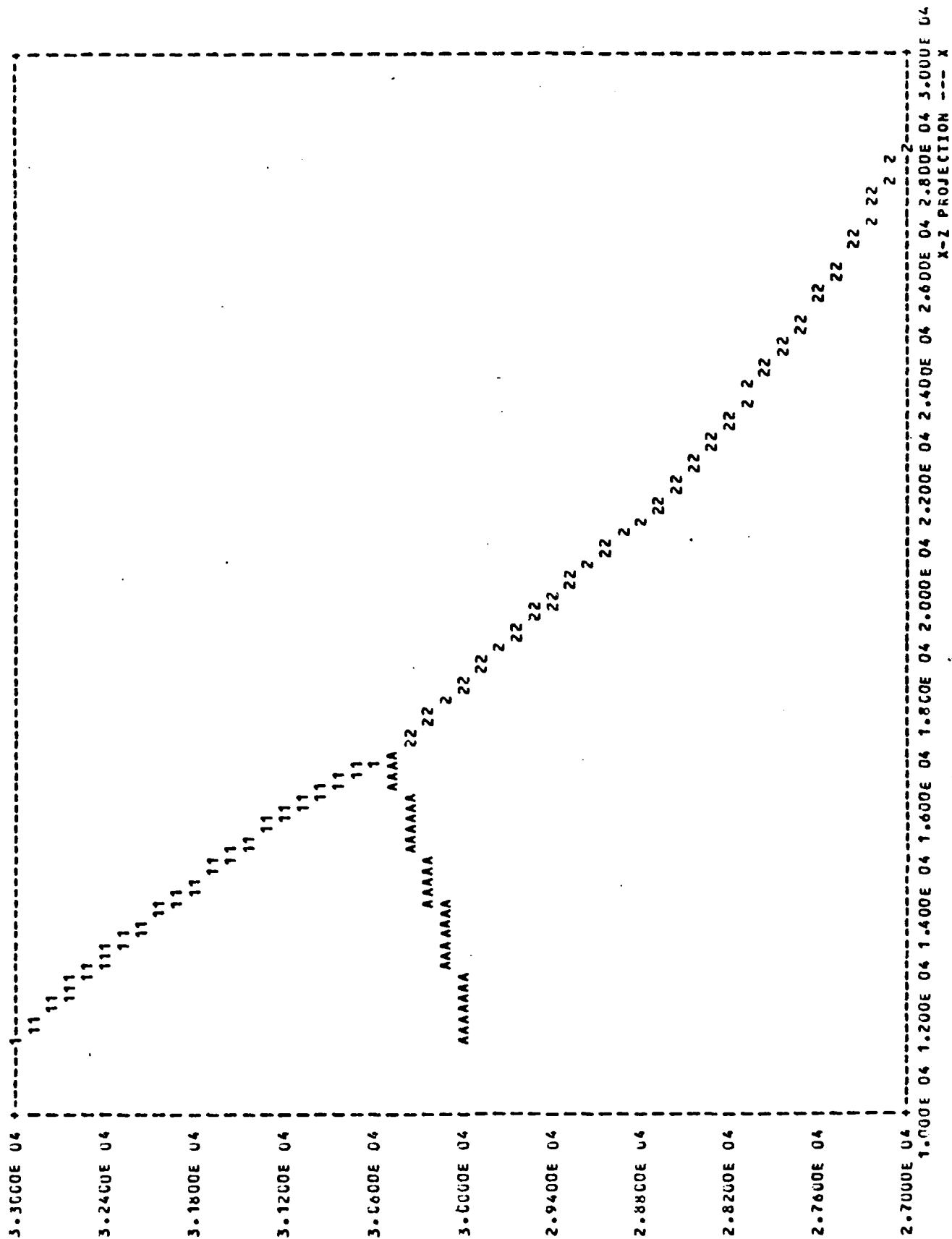


Figure 4-3c
Algorithm 1, Scenario 1





Algorithm 1, Scenario 1

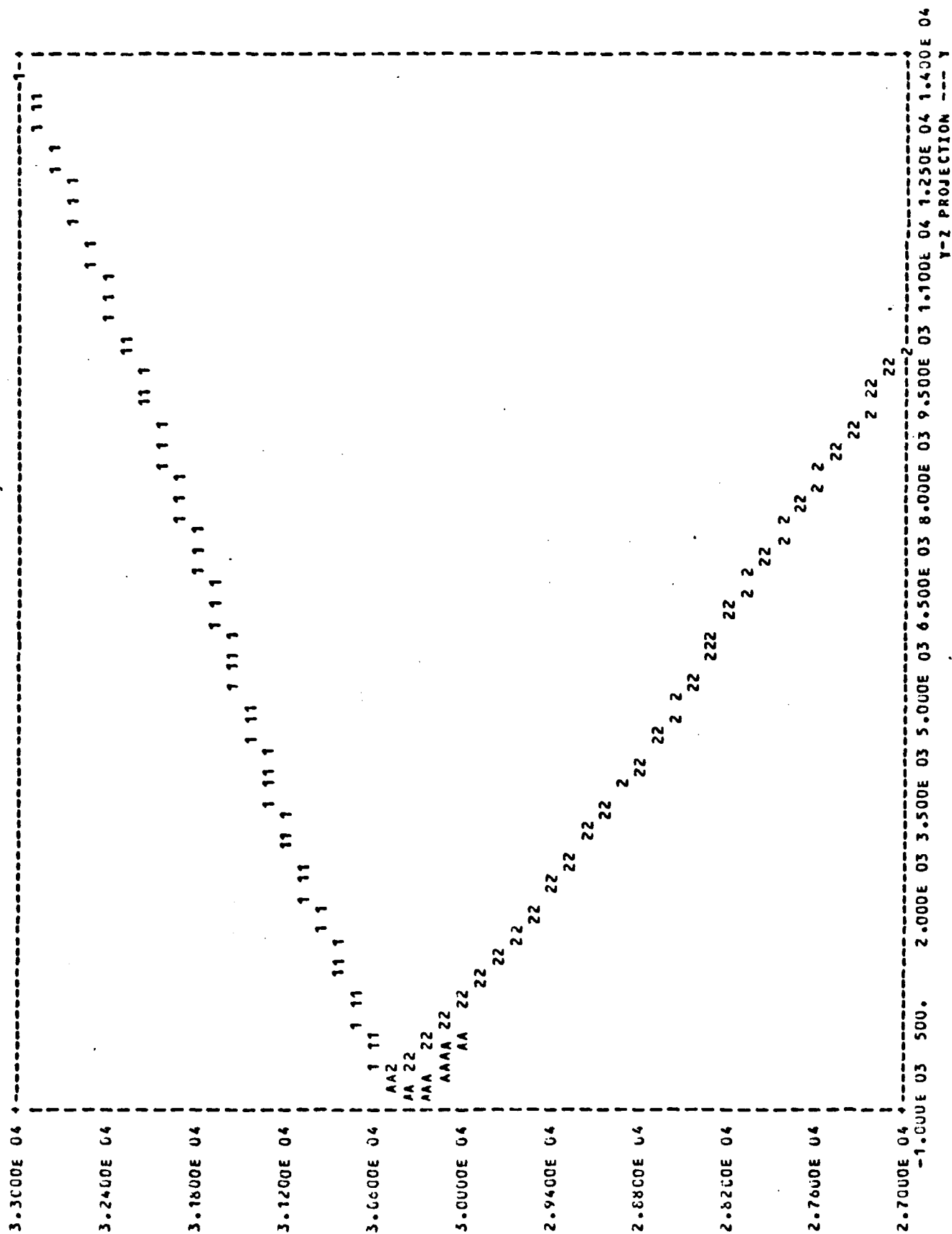


Table 4-4
Algorithm 2, Scenario 1

TSTEP = 0.1000, TAU = 0.1500

INIT XO(1) = 11258.0	INIT XO(7) = 11258.0	INIT XO(15) = 28578.0
INIT XO(2) = 0.000000	INIT XO(8) = 14000.0	INIT XO(14) = 10000.0
INIT XO(3) = 30000.0	INIT XO(9) = 33000.0	INIT XO(15) = 27000.0
INIT XO(4) = 1100.00	INIT XO(10) = 3150.00	INIT XO(16) = 3450.00
INIT XO(5) = 0.000000	INIT XO(11) = -7.99999	INIT XO(17) = 7.99999
INIT XO(6) = 0.000000	INIT XO(12) = -60.0000	INIT XO(18) = -150.000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.143E 05	DSEP2 = 0.202E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.116E 05	DSEP2 = 0.160E 05
TIME = 2.000	DSEP1 = 0.695E 04	DSEP2 = 0.122E 05
TIME = 3.000	DSEP1 = 0.634E 04	DSEP2 = 0.684E 04
TIME = 4.000	DSEP1 = 0.373E 04	DSEP2 = 0.557E 04
TIME = 5.000	DSEP1 = 0.121E 04	DSEP2 = 0.240E 04
TIME = 5.516	DSEP1 = 21.5	DSEP2 = 830.
TIME = 5.737	DSEP1 = 629.	DSEP2 = 66.9

CLOSURE RATE NEGATIVE AT TIME = 5.810***

TA1 : BEST DSEP = 17.8818	NOW = 600.464
TA2 : BEST DSEP = 60.9543	NOW = 69.5214

XO(1): 0.1635E 05	XO(7): 0.1634E 05	XO(13): 0.1637E 05
XO(2): 0.025.1	XO(8): -30.06	XO(14): 597.1
XO(3): 0.3084E 05	XO(9): 0.3066E 05	XO(15): 0.3090E 05
XO(4): 750.3	XO(10): 2150.	XO(16): 2182.
XO(5): -23.99	XO(11): -24.68	XO(17): 1.543
XO(6): 25.46	XO(12): -74.65	XO(18): -147.4

DELX1: 39.9	DELY1: 655.
DELZ1: 179.	DMIS1: 680.
BEST DMIS WAS 17.9	

DELX2: 13.3	DELY2: 28.0
DELZ2: -62.2	DMIS2: 69.5
BEST DMIS WAS 61.0	

Figure 4-4a
Algorithm 2, Scenario 1

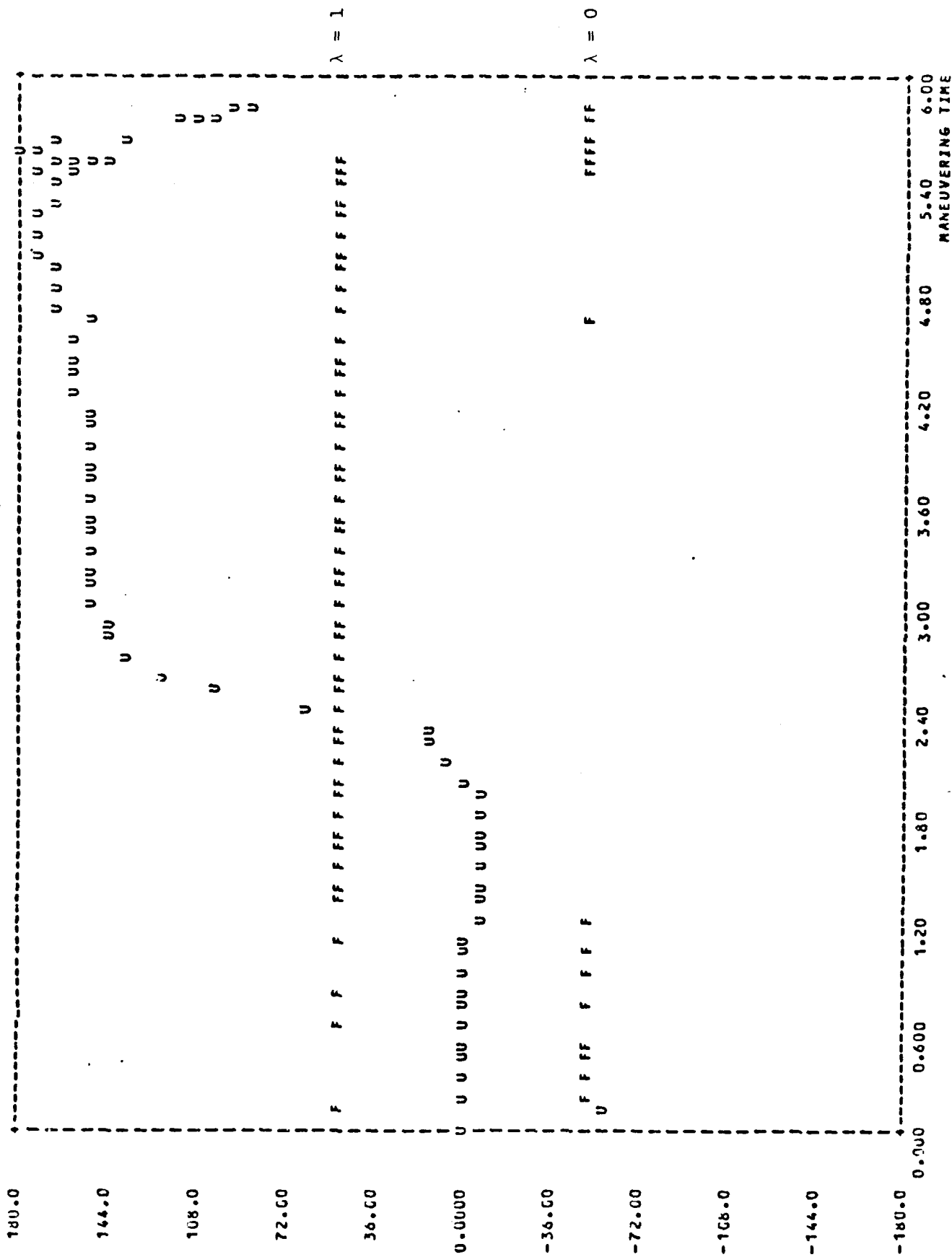


Figure 4-4b
Algorithm 2, Scenario 1

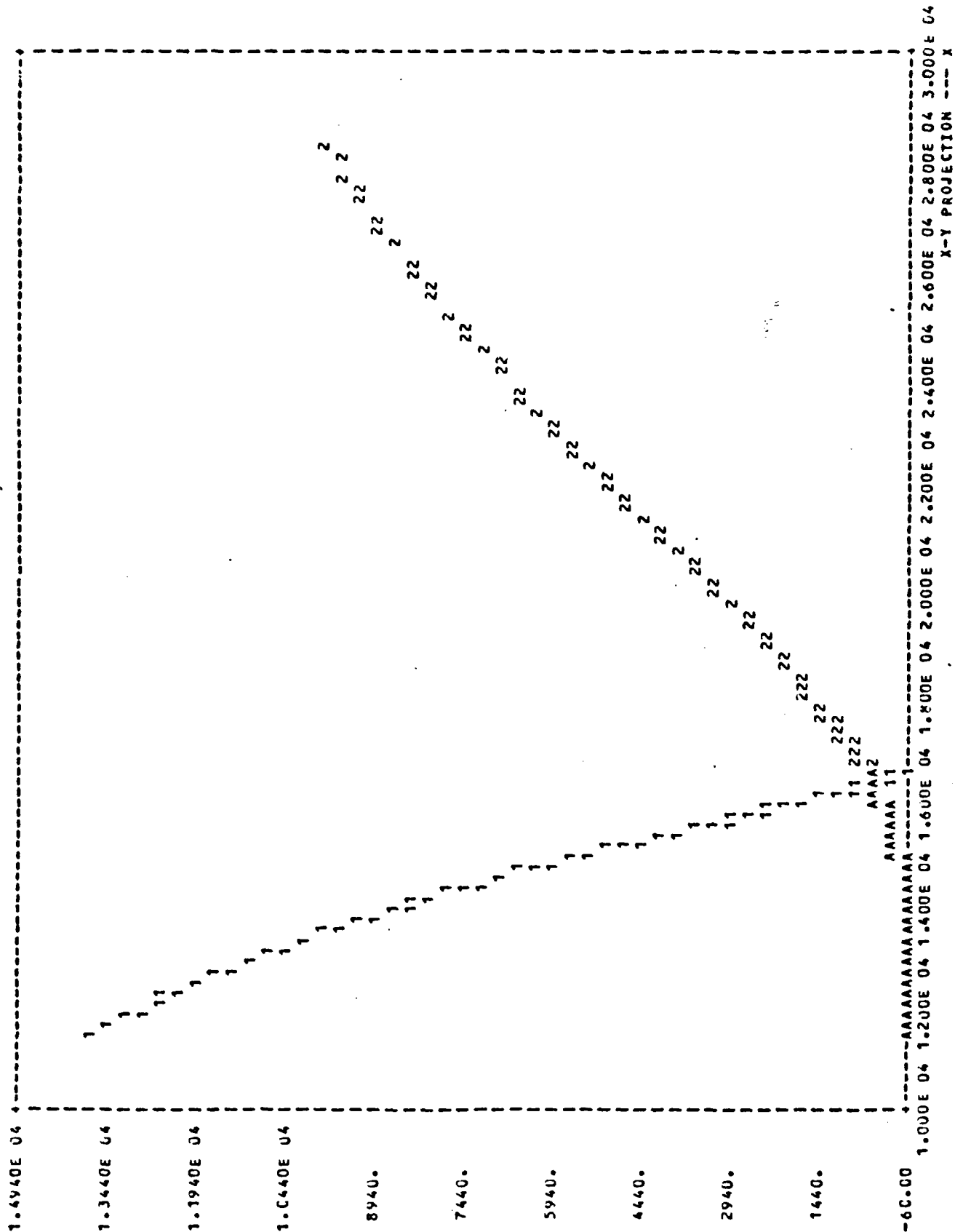


Figure 4-4c
Algorithm 2, Scenario 1

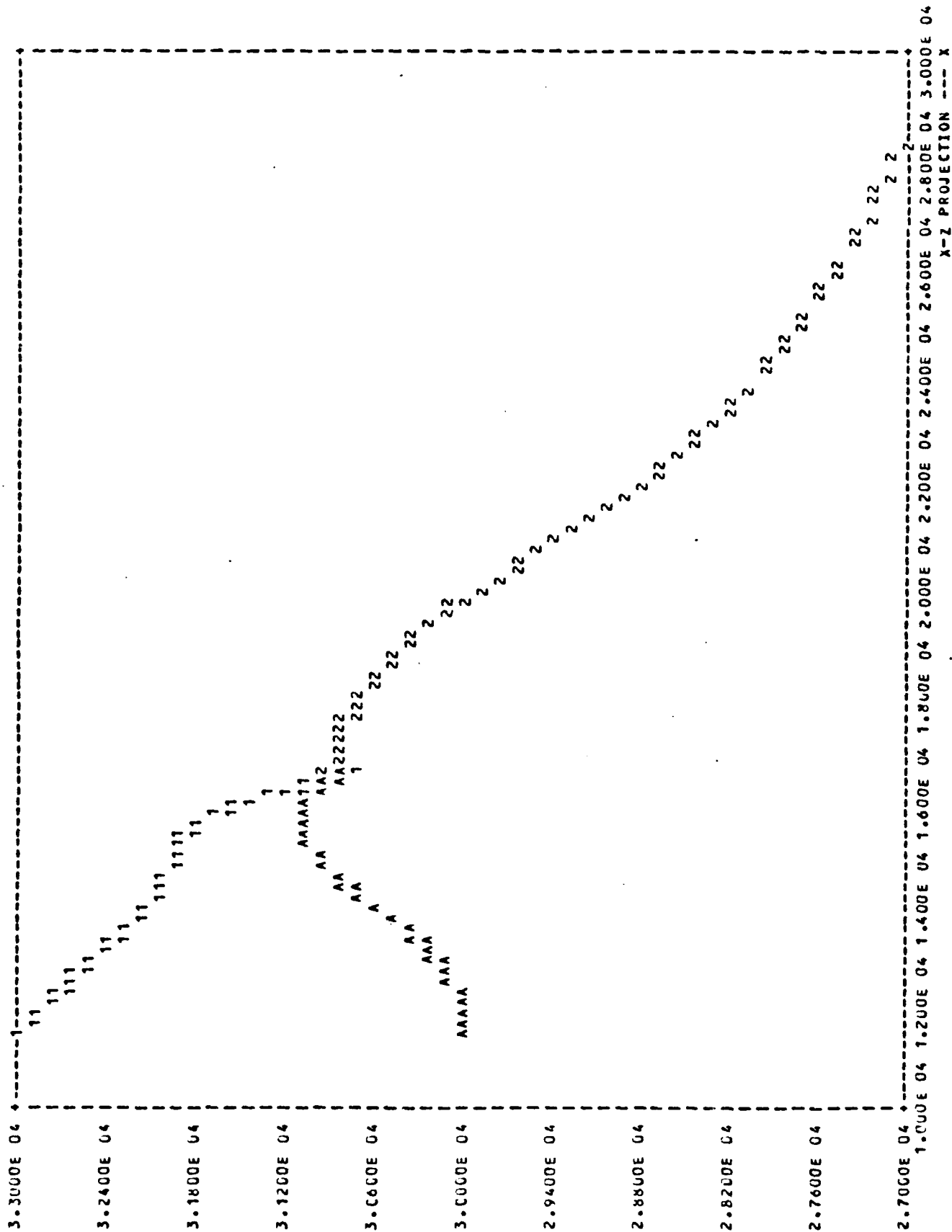


Figure 4-4d
Algorithm 2, Scenario 1

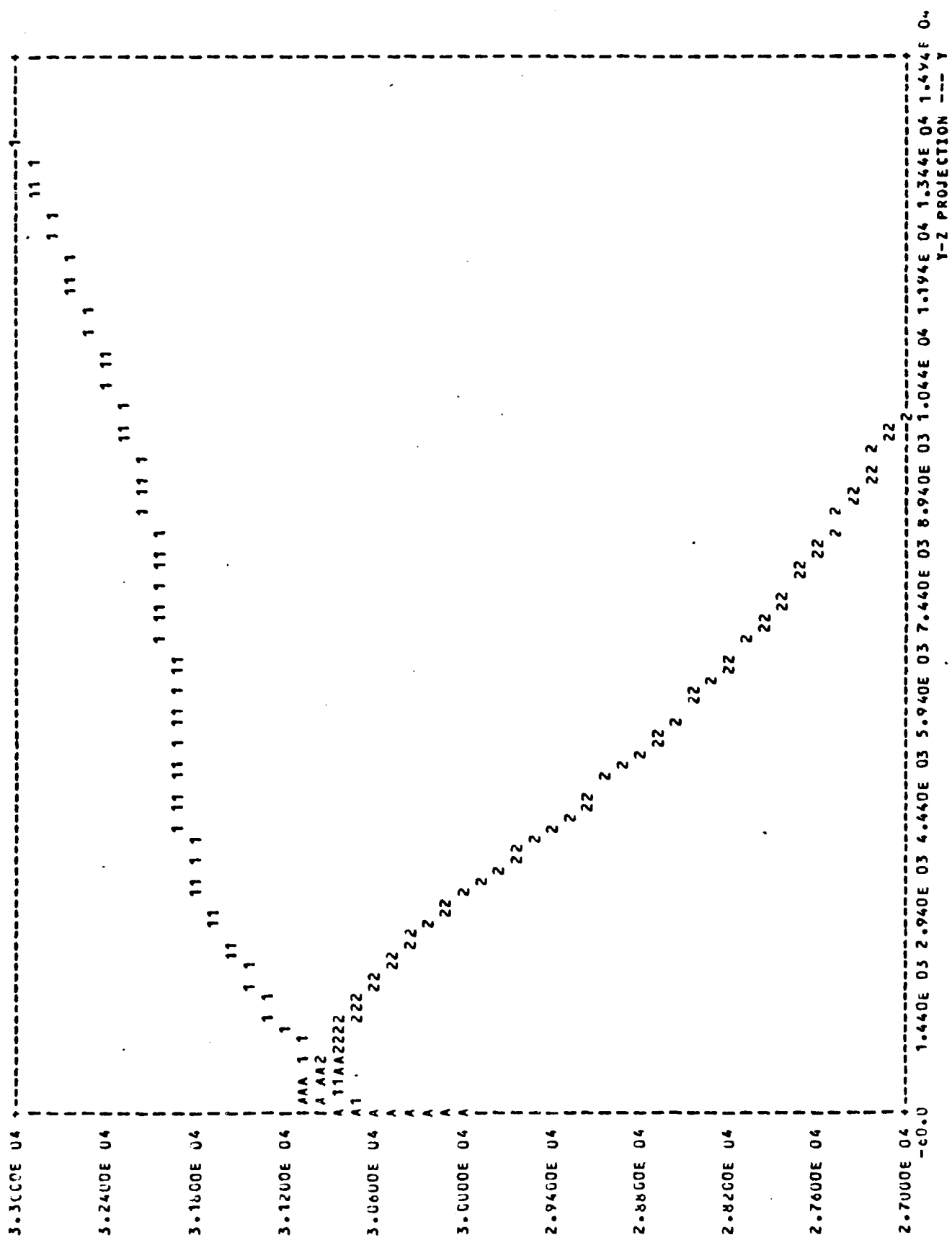


Table 4-5
Algorithm 1, Scenario 2

ISTEP = 0.1000, TAU = 0.1500

INIT X0(1) = 11258.0	INIT X0(7) = 11258.0	INIT X0(13) = 28578.0
INIT X0(2) = 0.000000	INIT X0(8) = 15000.0	INIT X0(14) = -9990.00
INIT X0(3) = 0.0000.0	INIT X0(9) = 35000.0	INIT X0(15) = 27000.0
INIT X0(4) = 1100.00	INIT X0(10) = 3150.00	INIT X0(16) = 3450.00
INIT X0(5) = 0.000000	INIT X0(11) = -7.99999	INIT X0(17) = 7.99999
INIT X0(6) = 0.000000	INIT X0(12) = -60.0000	INIT X0(18) = 150.000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50, YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.153E 05	DSEP2 = 0.202E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.126E 05	DSEP2 = 0.160E 05
TIME = 2.000	DSEP1 = 0.980E 04	DSEP2 = 0.122E 05
TIME = 3.000	DSEP1 = 0.720E 04	DSEP2 = 0.874E 04
TIME = 4.000	DSEP1 = 0.440E 04	DSEP2 = 0.573E 04
TIME = 5.000	DSEP1 = 0.170E 04	DSEP2 = 0.310E 04
TIME = 5.635	DSEP1 = 57.3	DSEP2 = 0.160E 04
TIME = 6.286	DSEP1 = 0.175E 04	DSEP2 = 188.

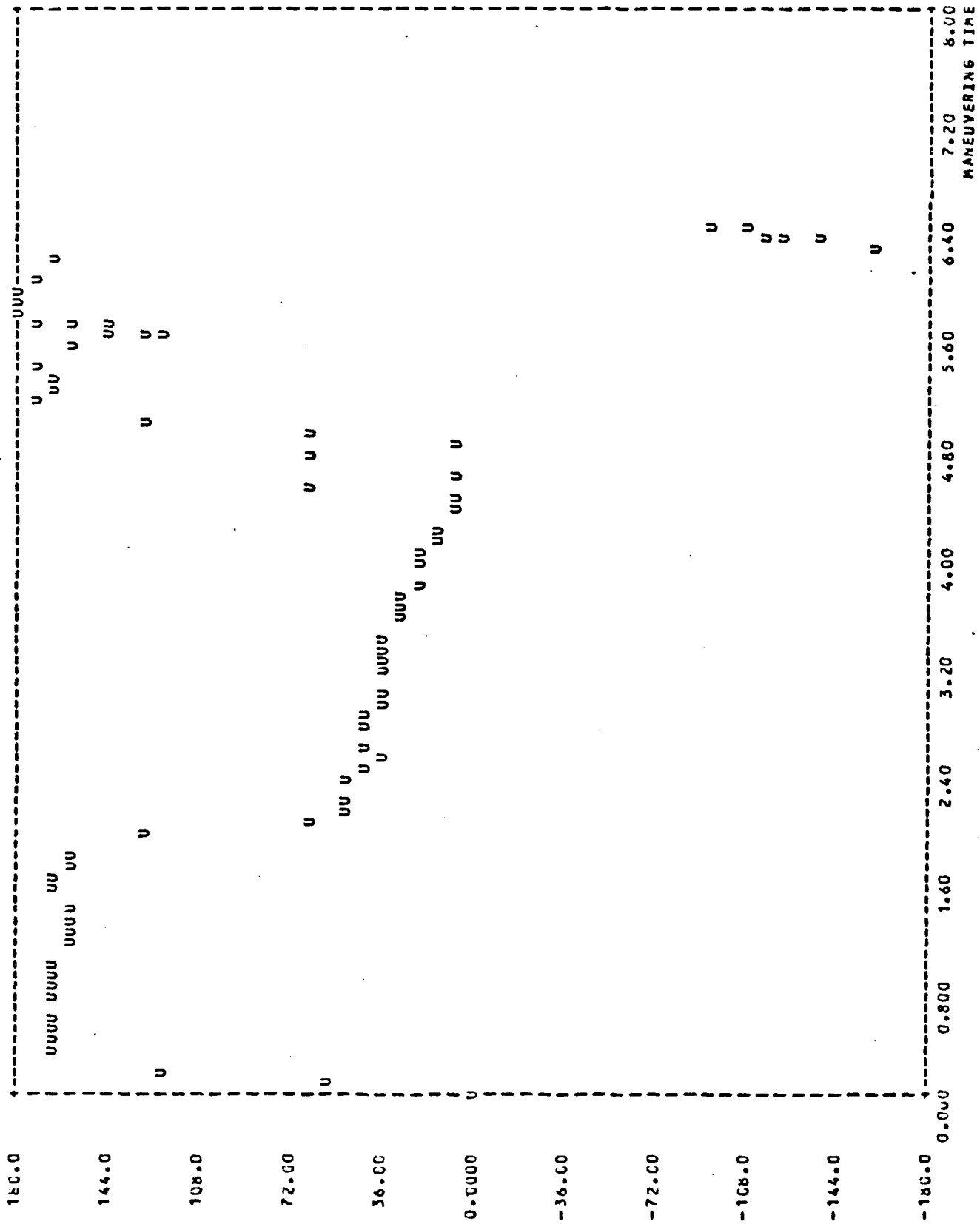
CLOSURE RATE NEGATIVE AT TIME = 6.376 ***

TA1 : BEST DSEP = 40.3800	, NOW = 1973.01
TA2 : BEST DSEP = 72.7325	, NOW = 75.4026

X0(1): 0.1676E 05	X0(7): 0.1664E 05	X0(13): 0.1676E 05
X0(2): 2052.	X0(8): 93.68	X0(14): 2064.
X0(3): 0.2088E 05	X0(9): 0.2867E 05	X0(15): 0.2096E 05
X0(4): 797.2	X0(10): 2114.	X0(16): 2110.
X0(5): -19.48	X0(11): -12.91	X0(17): 0.7152
X0(6): 40.36	X0(12): -79.90	X0(18): 121.1

DELTA1: 115.	DELY1: 0.196E 04
DELTA2: 211.	DELY2: 0.197E 04
BEST D-15 WAS 40.4	

DELTA2: -1.30	DELY2: -12.1
DELTA2: -72.4	DELY2: 73.4
BEST D-15 WAS 72.7	



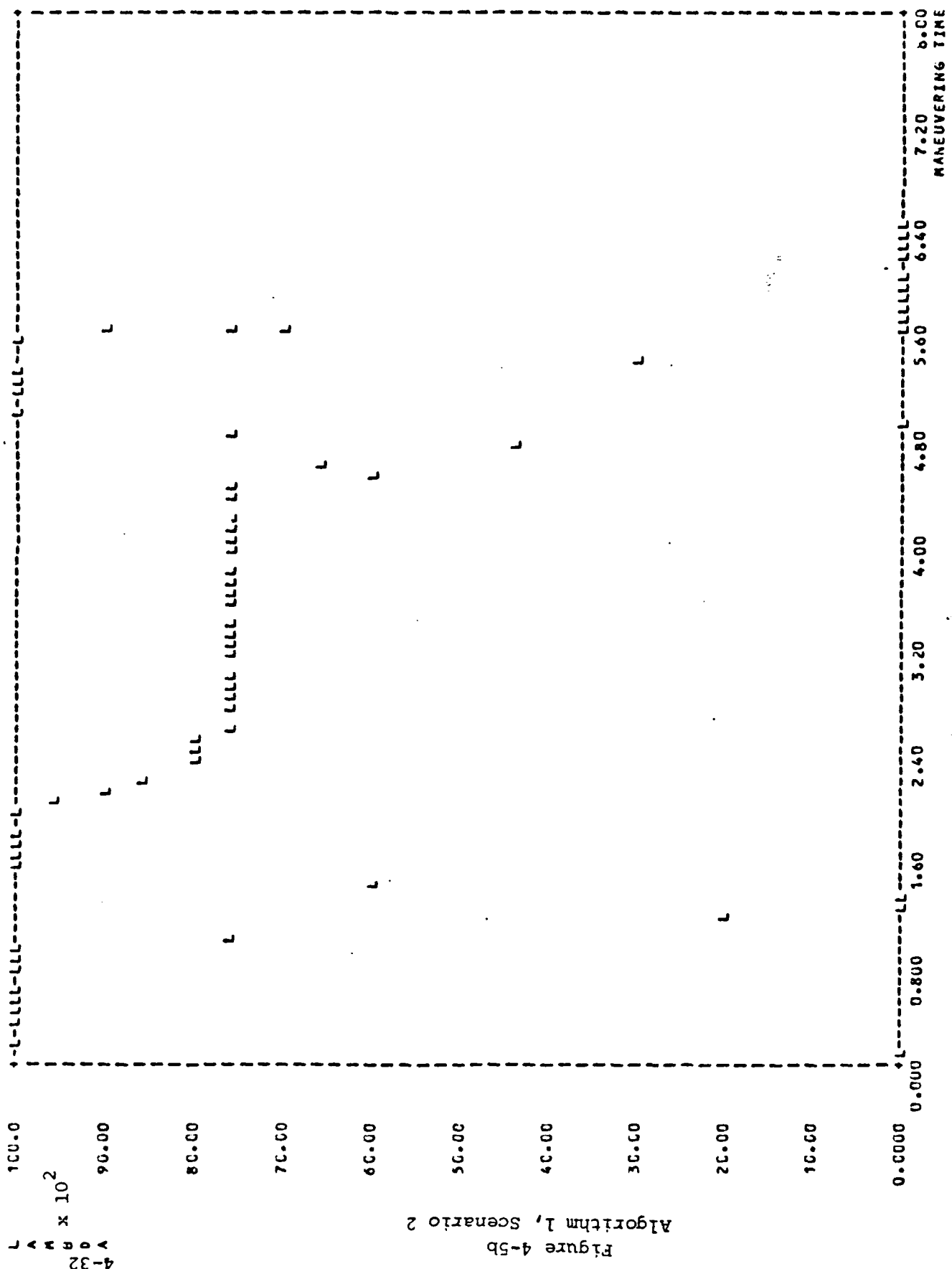


Figure 4-5c
Algorithm 1, Scenario 2

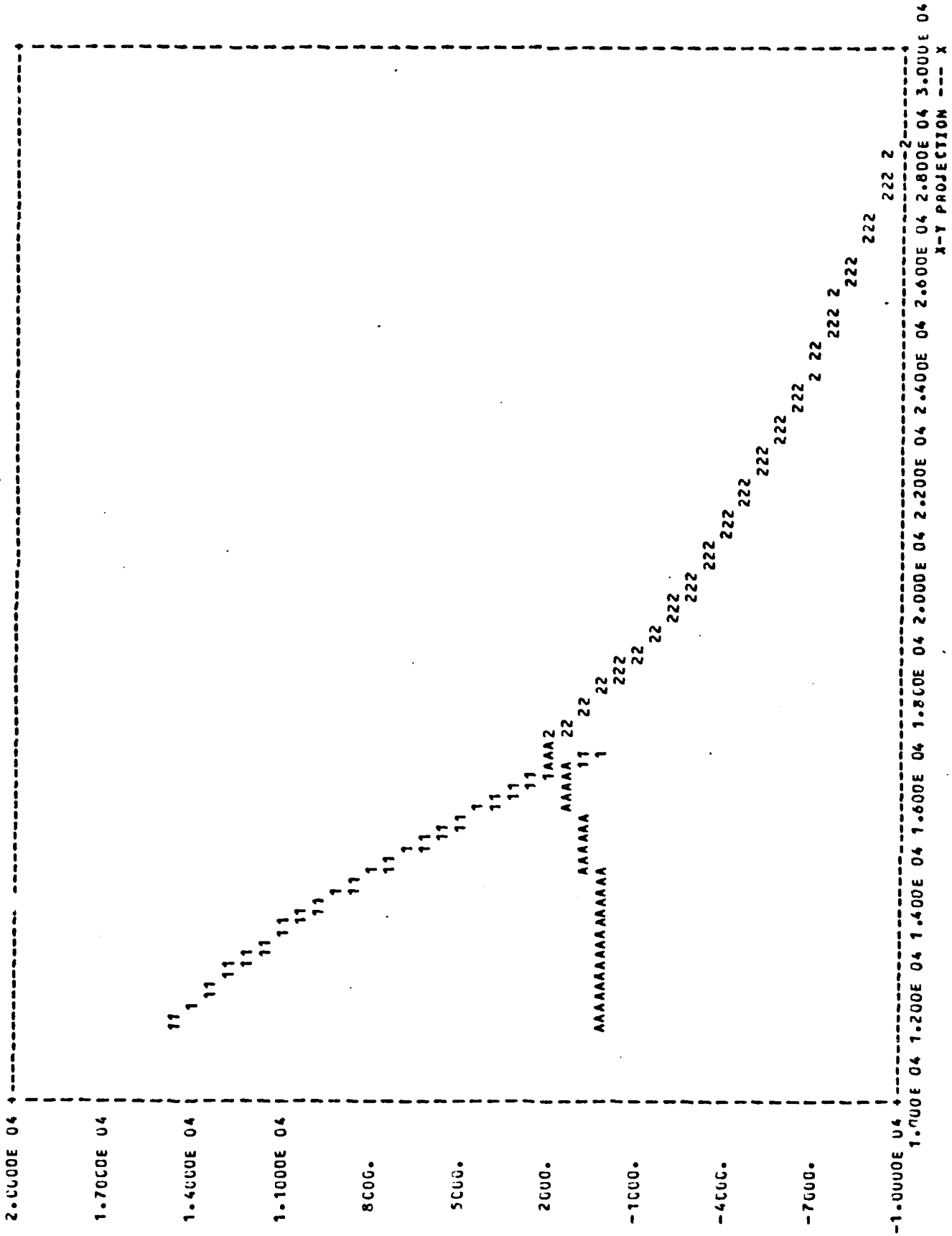
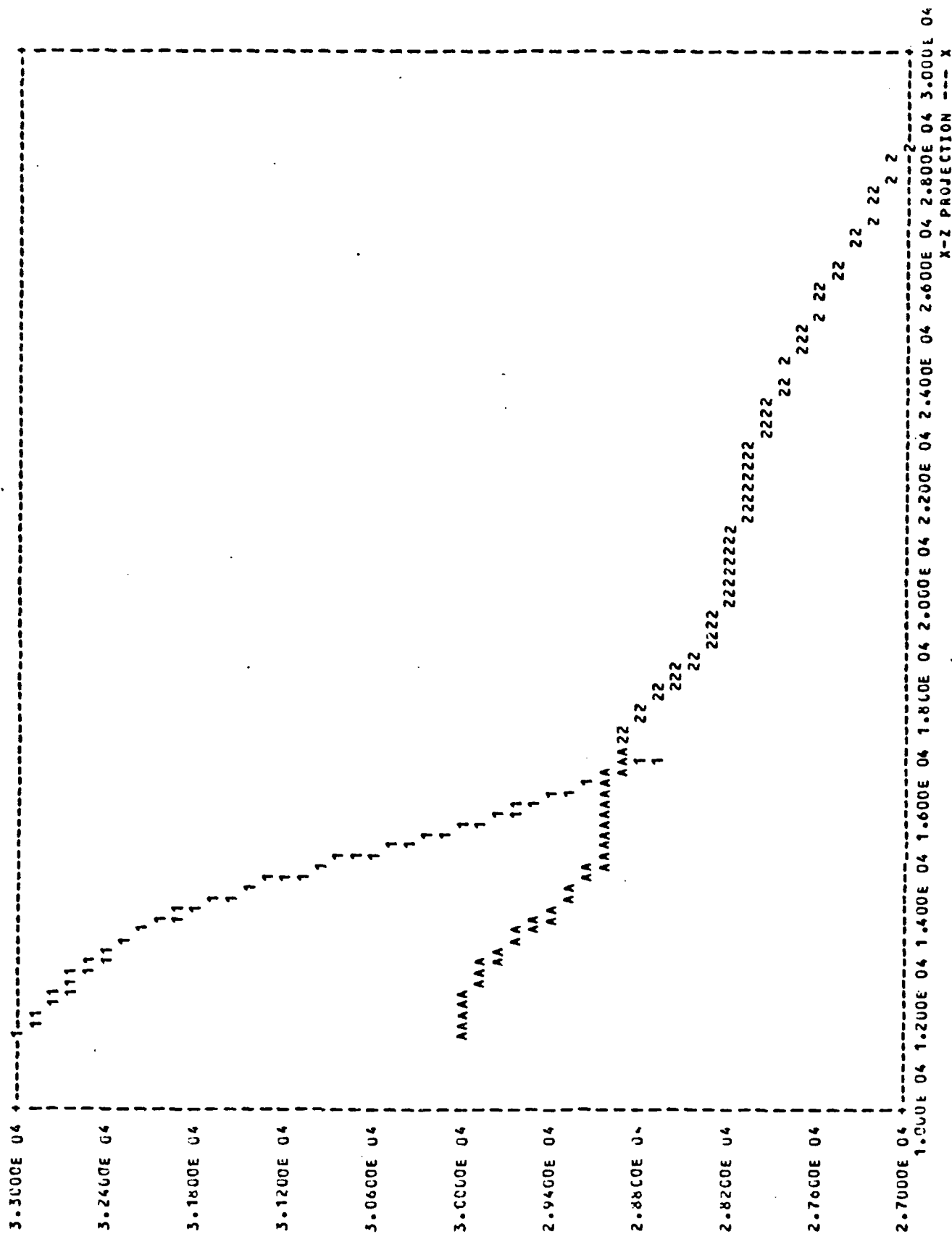


Figure 4-5d
Algorithm 1, Scenario 2



X-Z PROJECTION --- X

Figure 4-5e

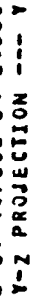


Table 4-6
Algorithm 4, Scenario 3

TSTEP = 0.1000, TAU = 0.1500

INIT XU(1) = 11258.0	INIT XU(7) = 0.000000	INIT XU(13) = 28578.0
INIT XU(2) = 0.000000	INIT XU(8) = 6500.00	INIT XU(14) = 9990.00
INIT XU(3) = 30000.0	INIT XU(9) = 33000.0	INIT XU(15) = 27000.0
INIT XU(4) = 1100.00	INIT XU(10) = 3150.00	INIT XU(16) = 3450.00
INIT XU(5) = 0.000000	INIT XU(11) = -7.94999	INIT XU(17) = 7.99999
INIT XU(6) = 0.000000	INIT XU(12) = -30.0000	INIT XU(18) = -150.000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.133E 05	DSEP2 = 0.202E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.112E 05	DSEP2 = 0.159E 05
TIME = 2.000	DSEP1 = 0.912E 04	DSEP2 = 0.119E 05
TIME = 3.000	DSEP1 = 0.711E 04	DSEP2 = 0.607E 04
TIME = 4.000	DSEP1 = 0.515E 04	DSEP2 = 0.457E 04
TIME = 5.000	DSEP1 = 0.326E 04	DSEP2 = 0.150E 04
TIME = 5.566	DSEP1 = 0.226E 04	DSEP2 = 76.9
TIME = 6.126	DSEP1 = 0.135E 04	DSEP2 = 0.144E 04
TIME = 7.006	DSEP1 = 49.6	DSEP2 = 0.356E 04

**** CLOSURE RATE NEGATIVE AT TIME = 7.045****

TA1 : BEST DSEP = 17.3176 , NOW = 18.4981

TA2 : BEST DSEP = 68.6587 , NOW = 3651.30

XU(1): 0.1586E 05	XU(7): 0.1587E 05	XU(13): 0.1249E 05
XU(2): 2601.	XU(8): 2612.	XU(14): 1272.
XU(3): 0.2728E 05	XU(9): 0.2727E 05	XU(15): 0.2774E 05
XU(4): 797.2	XU(10): 2016.	XU(16): 2035.
XU(5): -62.09	XU(11): -35.11	XU(17): -12.39
XU(6): -4.609	XU(12): -23.37	XU(18): -161.9

DELX1: -9.12 DELY1: -10.8

DELZ1: 11.9 DMIS1: 13.5

BEST DMIS WAS 17.3

DELX2: 0.337E 04 DELY2: 0.133E 04

DELZ2: -458. DMIS2: 0.365E 04

BEST DMIS WAS 68.7

180.0
144.0
108.0
72.0
36.0
0.0000
-36.00
-72.00
-108.0
-144.0
-180.0

Figure 4-6a
Algorithm 4, Scenario 3

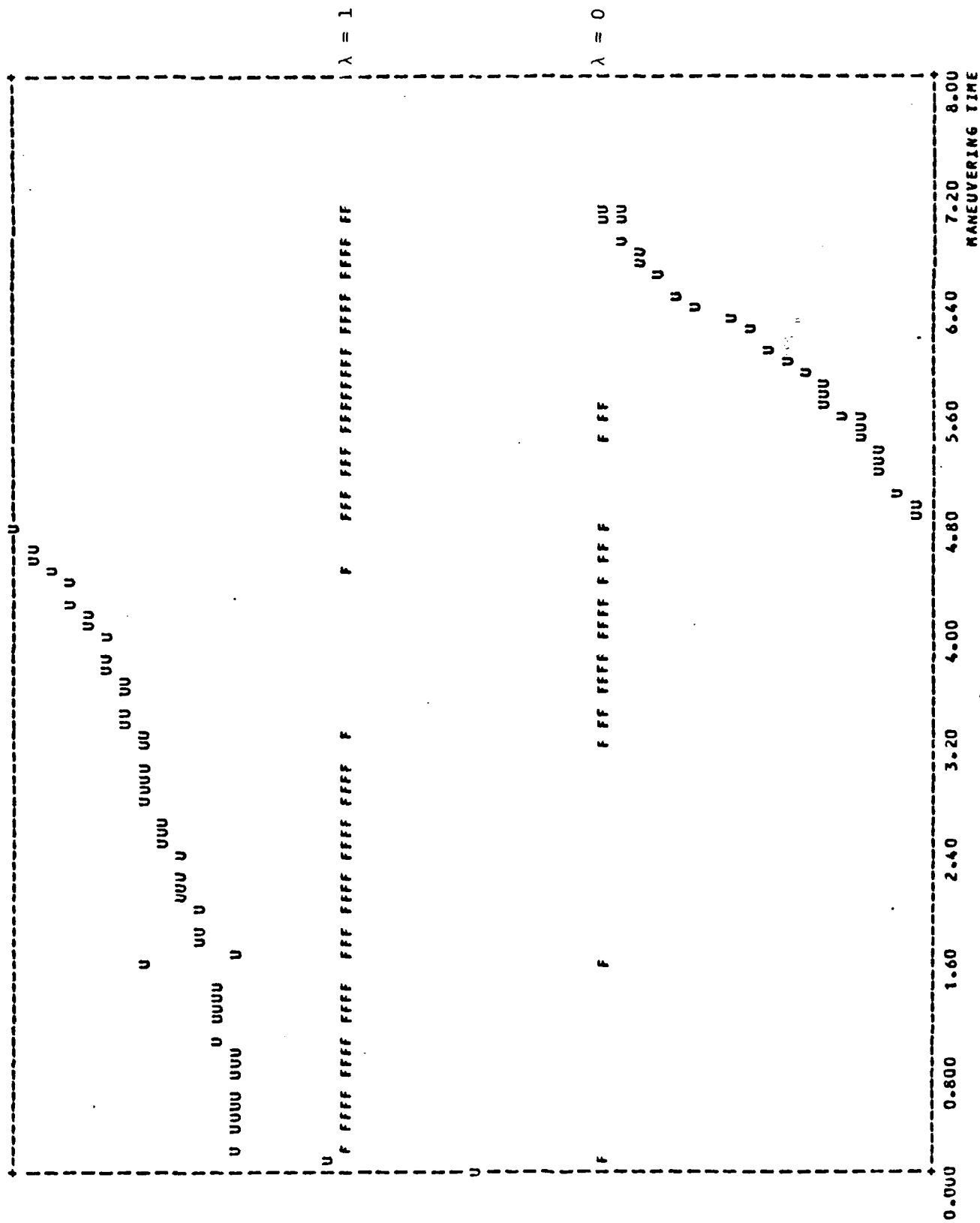


Figure 4-6b
Algorithm 4, Scenario 3

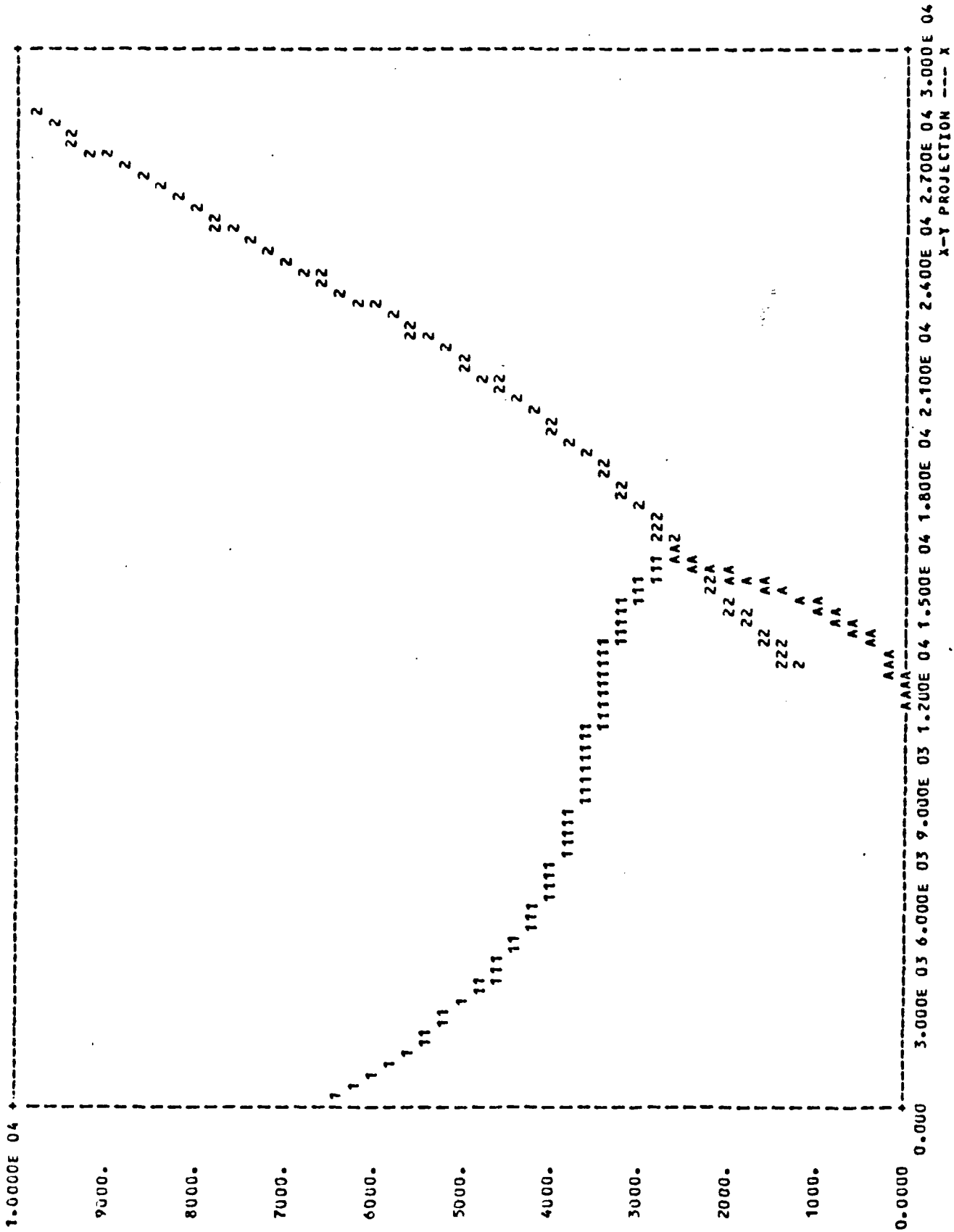


Figure 4-6c
Algorithm 4, Scenario 3

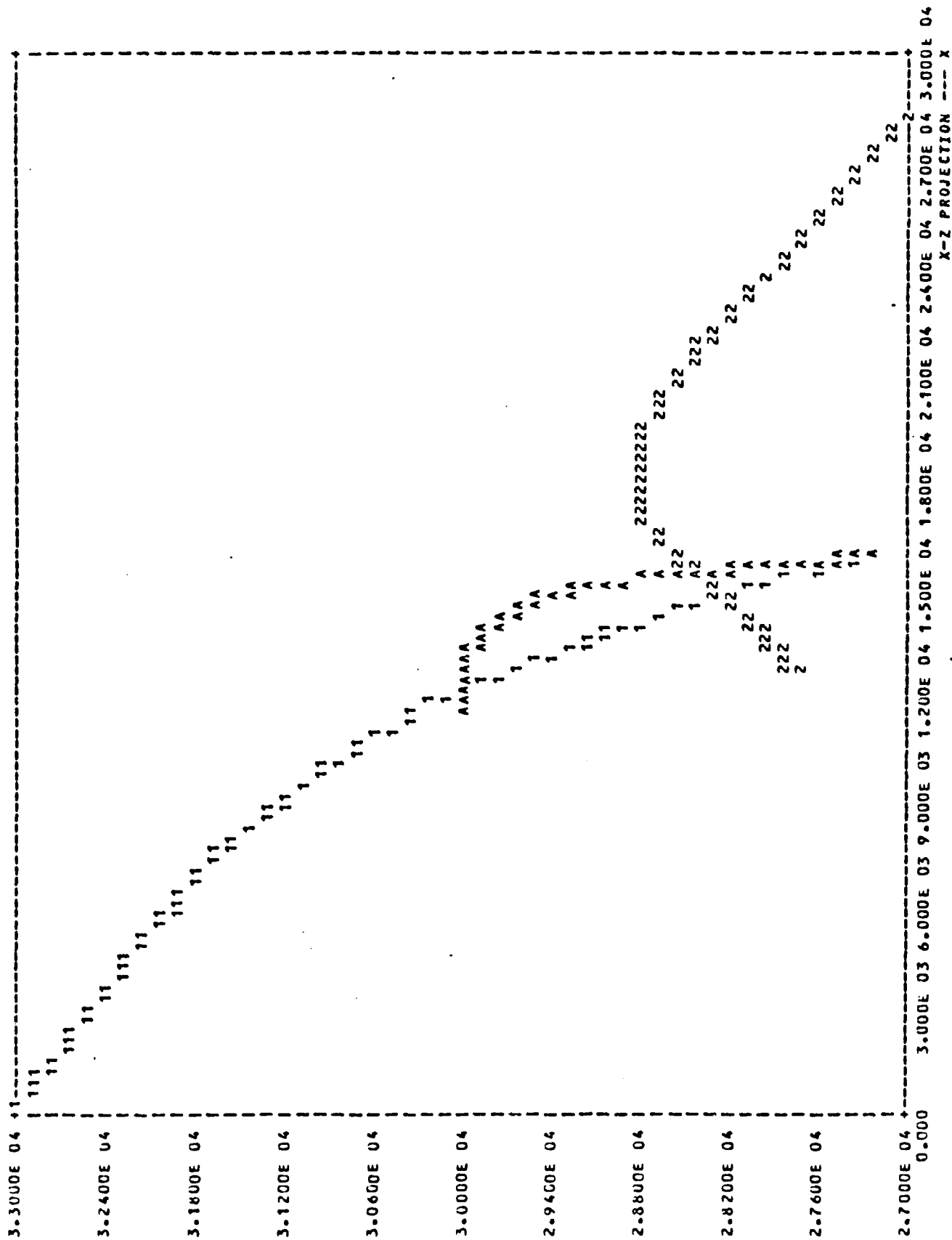


Figure 4-6d
Algorithm 4, Scenario 3

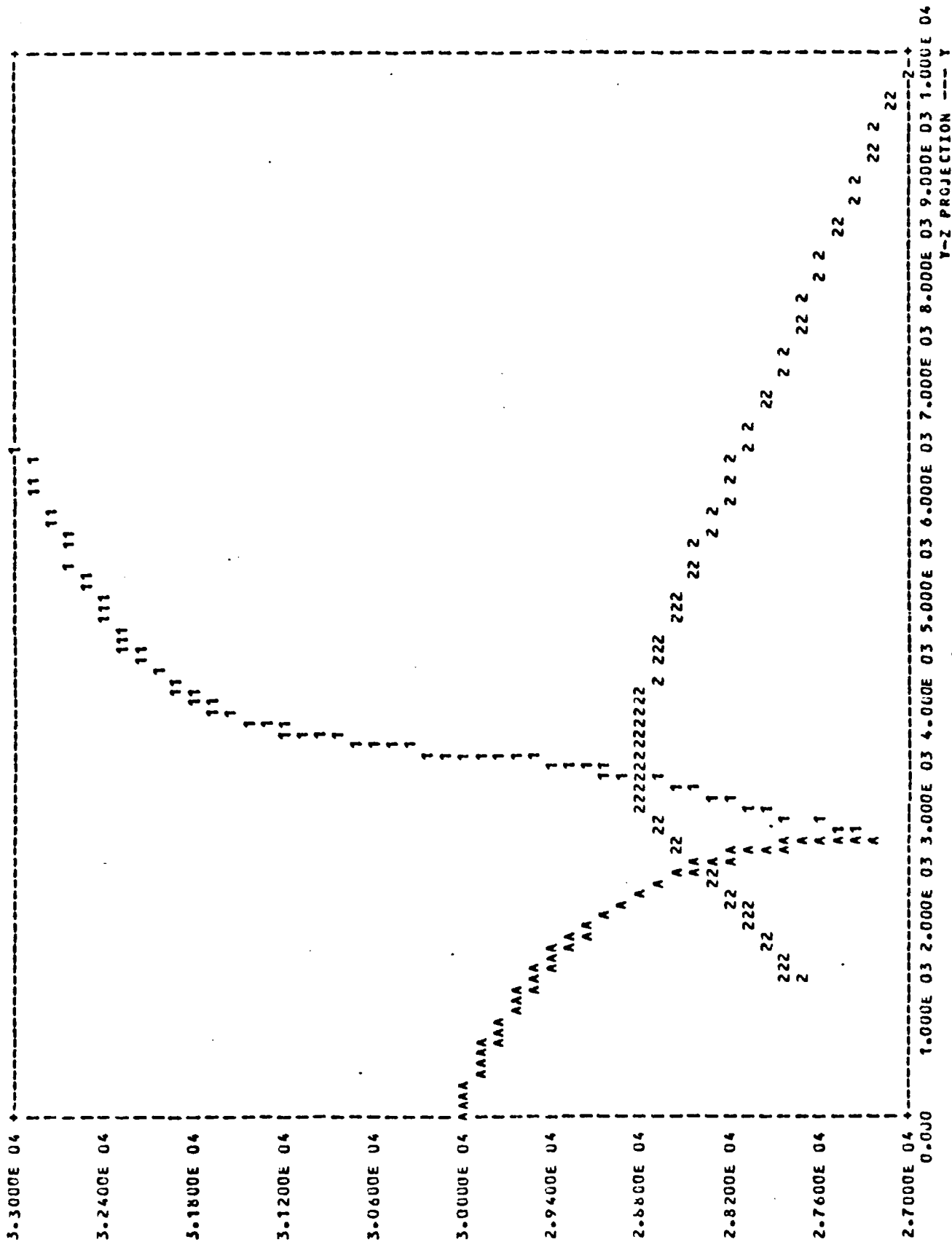


Table 4-7
Algorithm 4, Scenario 4

ISTEP = 0.1000, TAU = 0.1500

INIT X(1) = 11258.0	INIT X(7) = 28258.0	INIT X(13) = 0.000000
INIT X(2) = 0.000000	INIT X(8) = 0.000000	INIT X(14) = 6500.00
INIT X(3) = 30000.0	INIT X(9) = 33000.0	INIT X(15) = 33000.0
INIT X(4) = 1100.00	INIT X(10) = 3300.00	INIT X(16) = 3300.00
INIT X(5) = 0.000000	INIT X(11) = -7.99999	INIT X(17) = -7.99999
INIT X(6) = 0.000000	INIT X(12) = 180.000	INIT X(18) = -30.0000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50, YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.173E 05	DSEP2 = 0.133E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.130E 05	DSEP2 = 0.111E 05
TIME = 2.000	DSEP1 = 0.910E 04	DSEP2 = 0.689E 04
TIME = 3.000	DSEP1 = 0.563E 04	DSEP2 = 0.681E 04
TIME = 4.000	DSEP1 = 0.269E 04	DSEP2 = 0.479E 04
TIME = 4.970	DSEP1 = 274.	DSEP2 = 0.293E 04
TIME = 5.096	DSEP1 = 29.8	DSEP2 = 0.270E 04
TIME = 5.360	DSEP1 = 633.	DSEP2 = 0.225E 04
TIME = 5.360	DSEP1 = 0.287E 04	DSEP2 = 636.

*** CLOSURE RATE NEGATIVE AT TIME = 6.795***

TA1 : BEST DSEP = 20.5491	, NOW = 3791.88
TA2 : BEST DSEP = 23.0974	, NOW = 25.1589

X(1): 0.1586E 05	X(7): 0.1225E 05	X(13): 0.1590E 05
X(2): 2139.	X(8): 2707.	X(14): 2156.
X(3): 0.2693E 05	X(9): 0.2598E 05	X(15): 0.2692E 05
X(4): 346.0	X(10): 2124.	X(16): 2126.
X(5): -64.77	X(11): -37.31	X(17): -36.24
X(6): -19.29	X(12): 158.7	X(18): -24.84

DELX1: 0.363E 04	DELY1: -568.
DELZ1: 949.	DMIS1: 0.379E 04
BEST DMIS WAS 20.6	

DELX2: -17.2	DELY2: -16.8
DELZ2: 7.29	DMIS2: 25.2
BEST DMIS WAS 23.1	

130.0

144.0

108.0

72.00

36.00

0.0000

-36.00

-72.00

-108.0

-144.0

-180.0

4-42
S
M
T
R
A
V
C

Figure 4-7a
Algorithm 4, Scenario 4

$\lambda = 1$

$\lambda = 0$

FFF FFF FFF FFFF FFFF FFFF

FF FFFF FFFF FFFF FFFF F F FFFF FFFF FFFF

0.00 0.80 1.60 2.40 3.20 4.00 4.80 5.60 6.40 7.20 8.00
MANEUVERING TIME

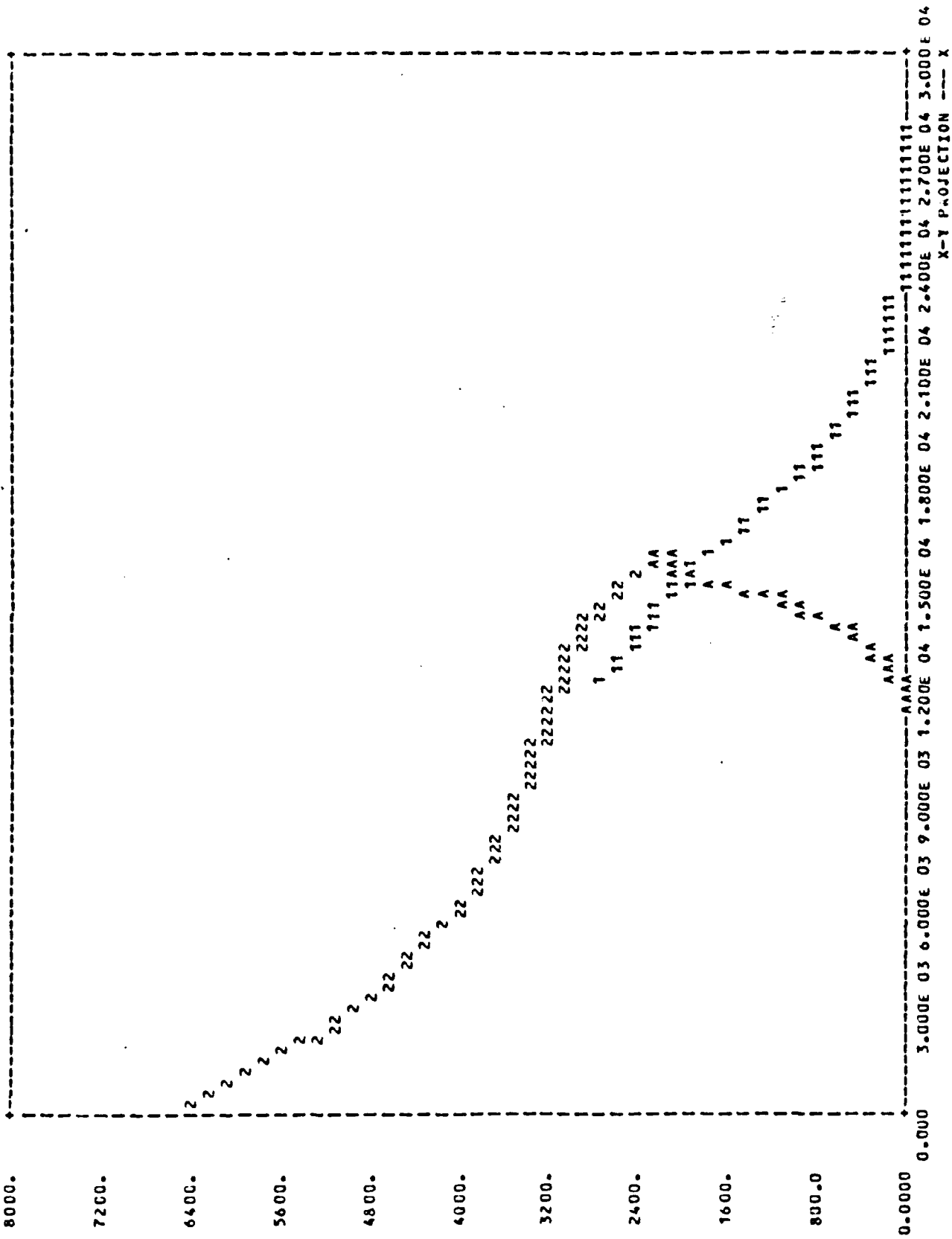


Figure 4-7c
Algorithm 4, Scenario 4

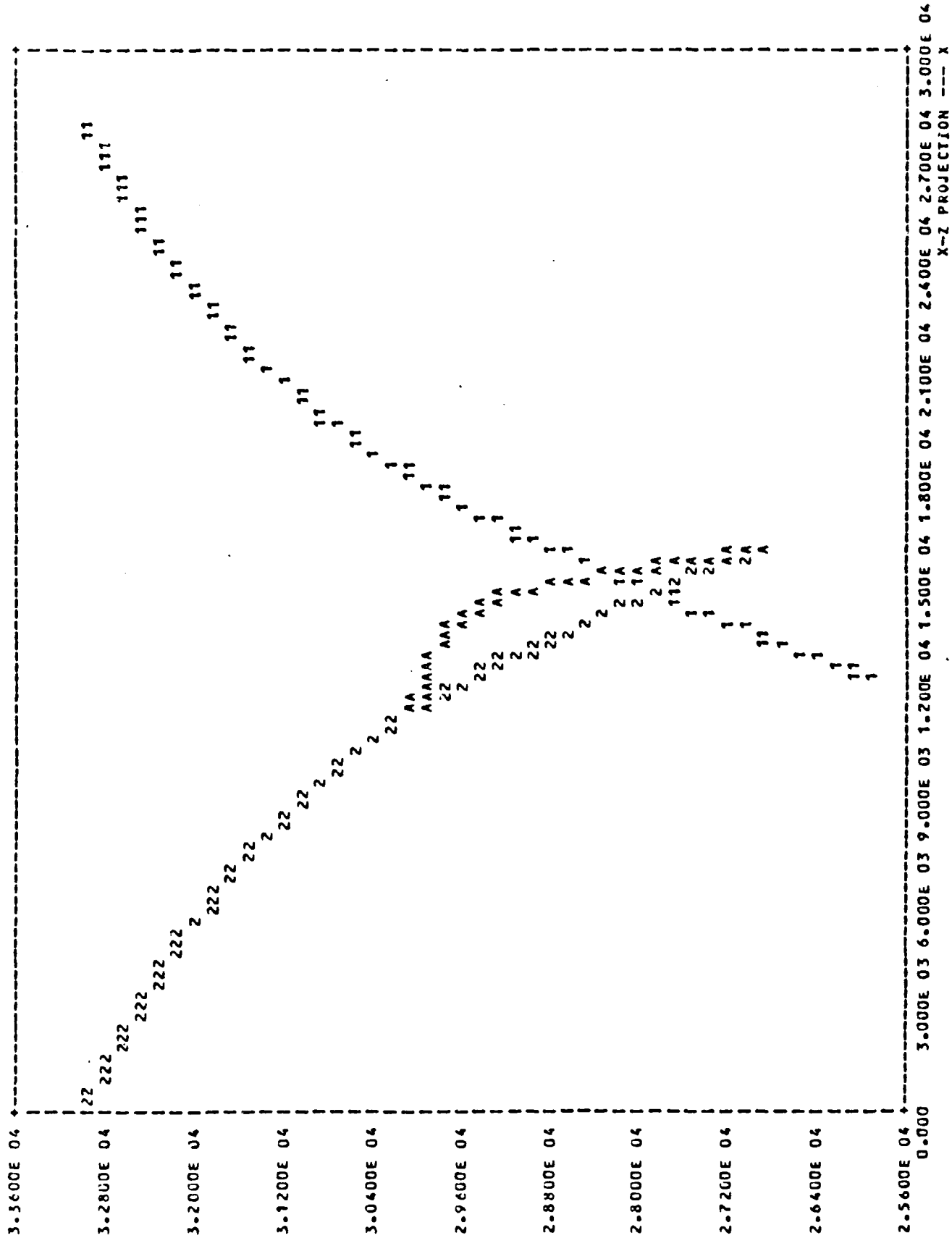


Figure 4-7d
Algorithm 4, Scenario 4

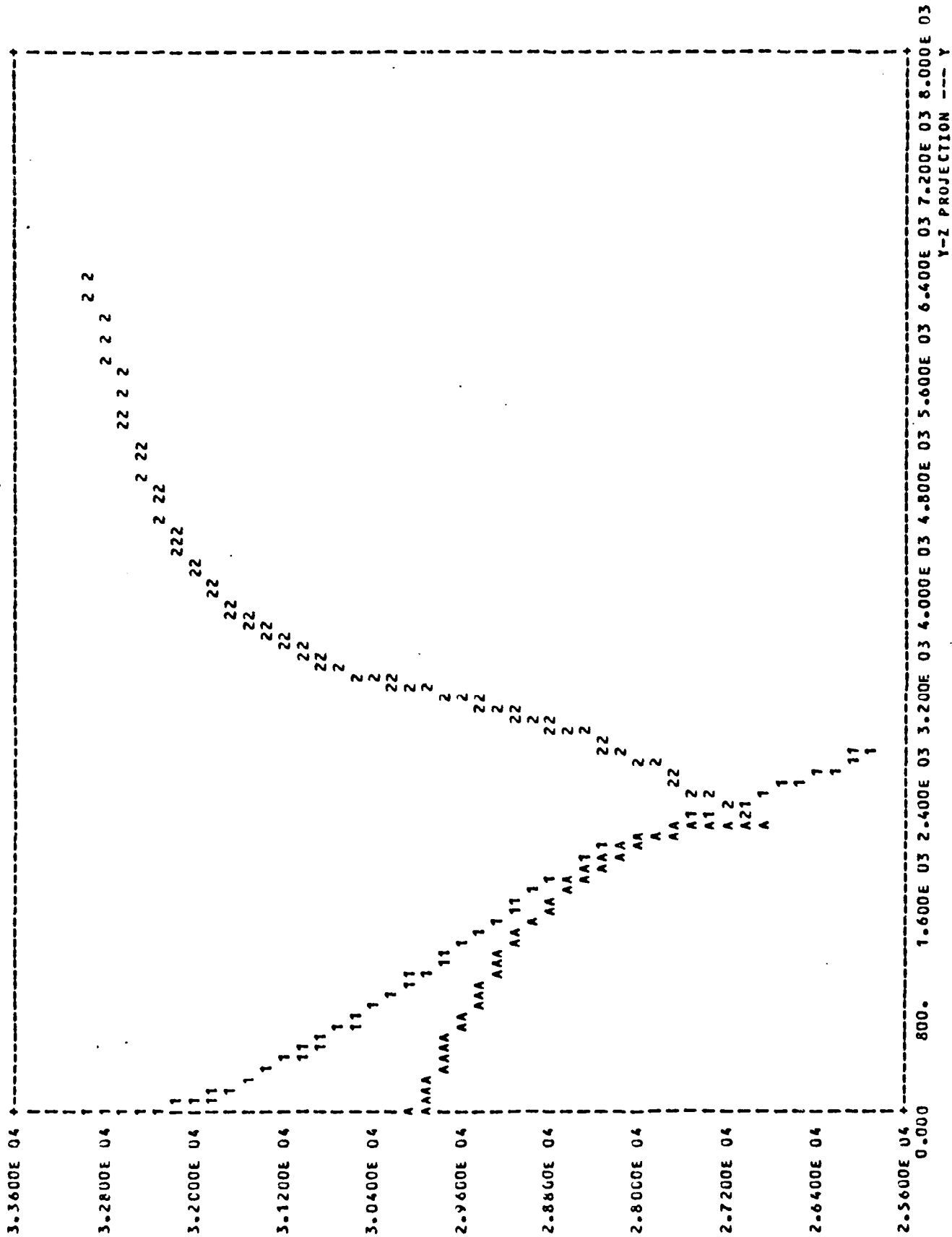


Table 4-8
Algorithm 1, Scenario 5

1STEP = 0.1000, TAU = 0.1500

INIT XU(1) = 15000.0	INIT XU(7) = 3000.00	INIT XU(13) = 15000.0
INIT XU(2) = 0.000000	INIT XU(8) = 0.000000	INIT XU(14) = 12000.0
INIT XU(3) = 30000.0	INIT XU(9) = 30000.0	INIT XU(15) = 30000.0
INIT XU(4) = 1100.00	INIT XU(10) = 3300.00	INIT XU(16) = 3300.00
INIT XU(5) = 0.000000	INIT XU(11) = 0.000000	INIT XU(17) = 0.000000
INIT XU(6) = 0.000000	INIT XU(12) = 0.000000	INIT XU(18) = -60.0000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.120E 05	DSEP2 = 0.120E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.989E 04	DSEP2 = 0.919E 04
TIME = 2.000	DSEP1 = 0.794E 04	DSEP2 = 0.660E 04
TIME = 3.000	DSEP1 = 0.613E 04	DSEP2 = 0.426E 04
TIME = 4.000	DSEP1 = 0.449E 04	DSEP2 = 0.206E 04
TIME = 4.927	DSEP1 = 0.306E 04	DSEP2 = 108.
TIME = 5.248	DSEP1 = 0.261E 04	DSEP2 = 577.
TIME = 6.248	DSEP1 = 0.116E 04	DSEP2 = 0.266E 04
TIME = 7.095	DSEP1 = 35.6	DSEP2 = 0.411E 04

*** CLOSURE RATE NEGATIVE AT TIME = 7.119***

TA1 : BEST DSEP = 28.7695 , NOW = 30.8579

TA2 : BEST DSEP = 52.8064 , NOW = 4150.34

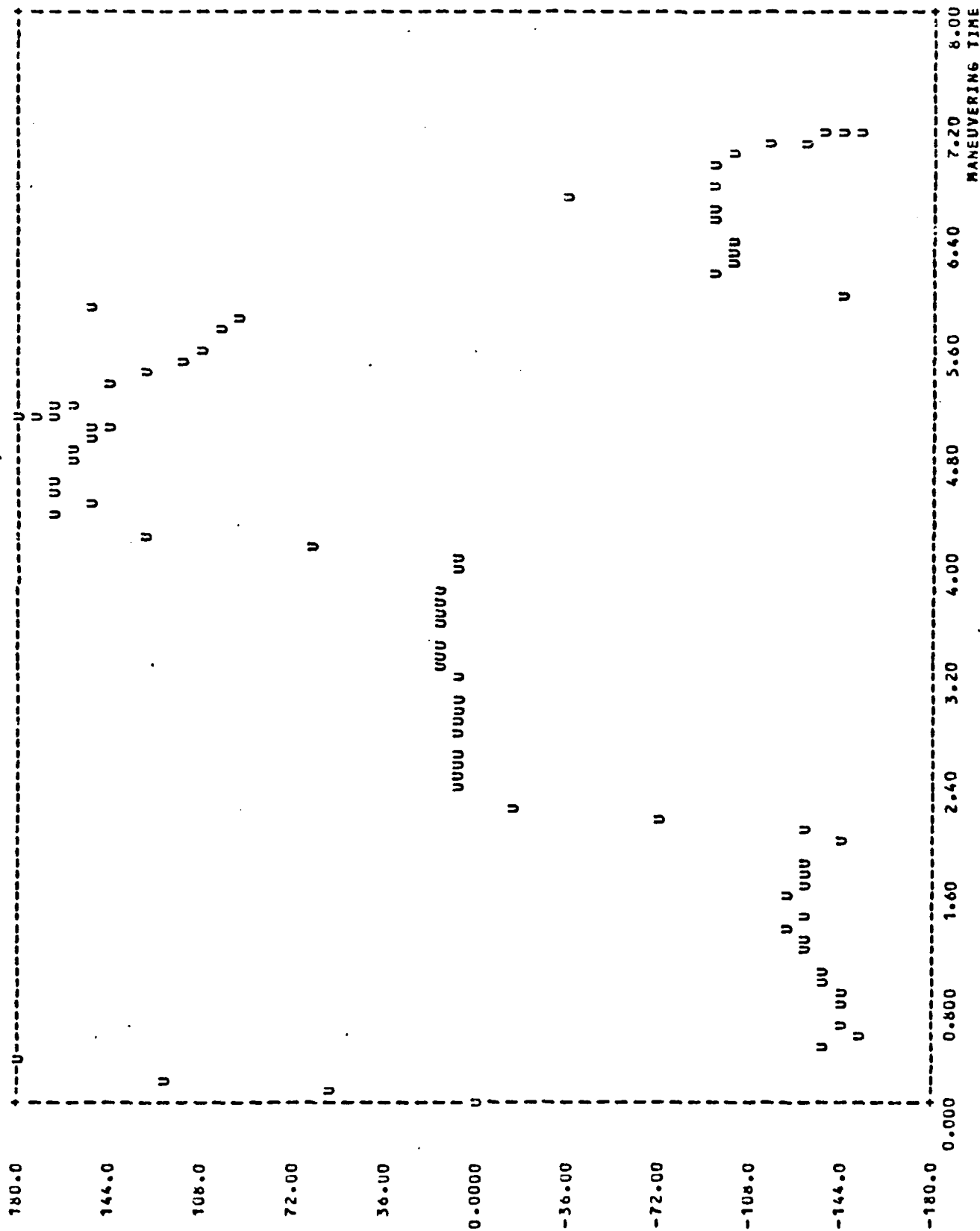
XU(1): 0.2145E 05	XU(7): 0.2146E 05	XU(13): 0.2173E 05
XU(2): -660.7	XU(8): -832.5	XU(14): -4955.
XU(3): 0.2870E 05	XU(9): 0.2869E 05	XU(15): 0.2932E 05
XU(4): 753.9	XU(10): 2041.	XU(16): 1965.
XU(5): -26.19	XU(11): -10.97	XU(17): -2.076
XU(6): -13.55	XU(12): -4.750	XU(18): -52.37

DELX1: -11.5	DELY1: -28.3
DELZ1: 4.48	DMIS1: 30.9
BEST DMIS WAS 28.8	

DELX2: -270.	DELY2: 0.409E 04
DELZ2: -621.	DMIS2: 0.415E 04
BEST DMIS WAS 52.8	

4-47
S E T S M V B

Figure 4-8a
Algorithm 1, Scenario 5



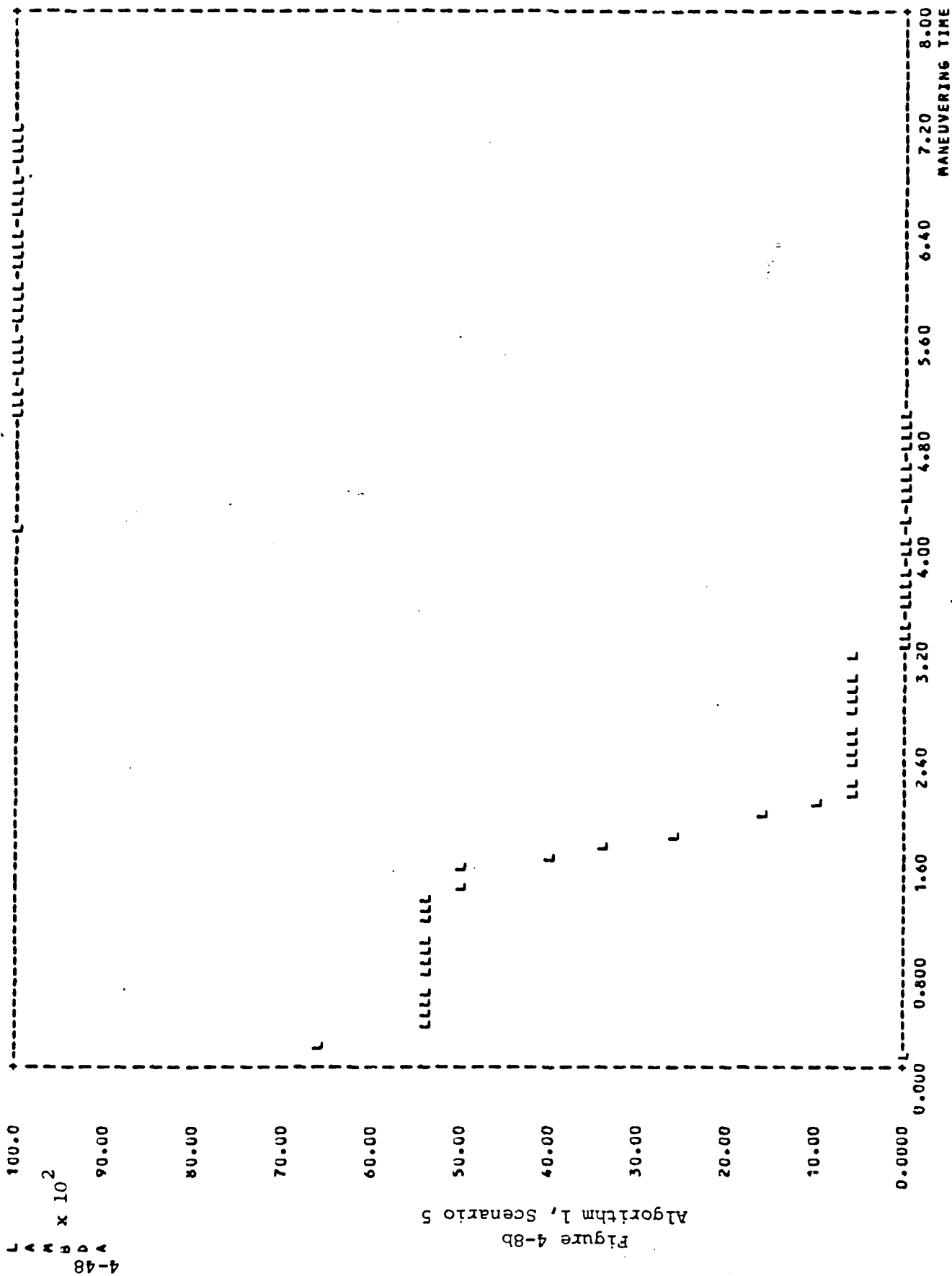


Figure 4-8c
Algorithm 1, Scenario 5

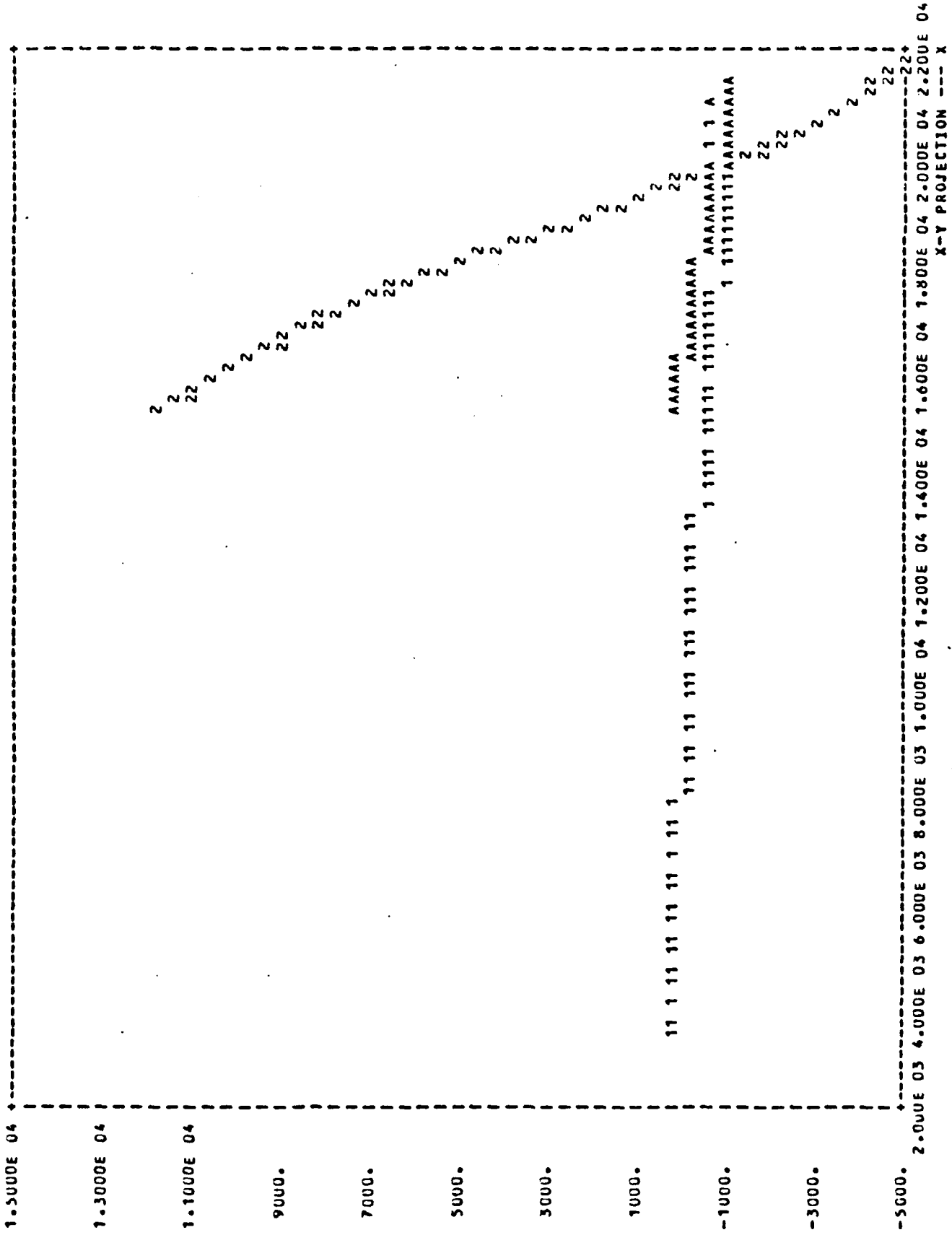


Figure 4-8d
Algorithm 1, Scenario 5

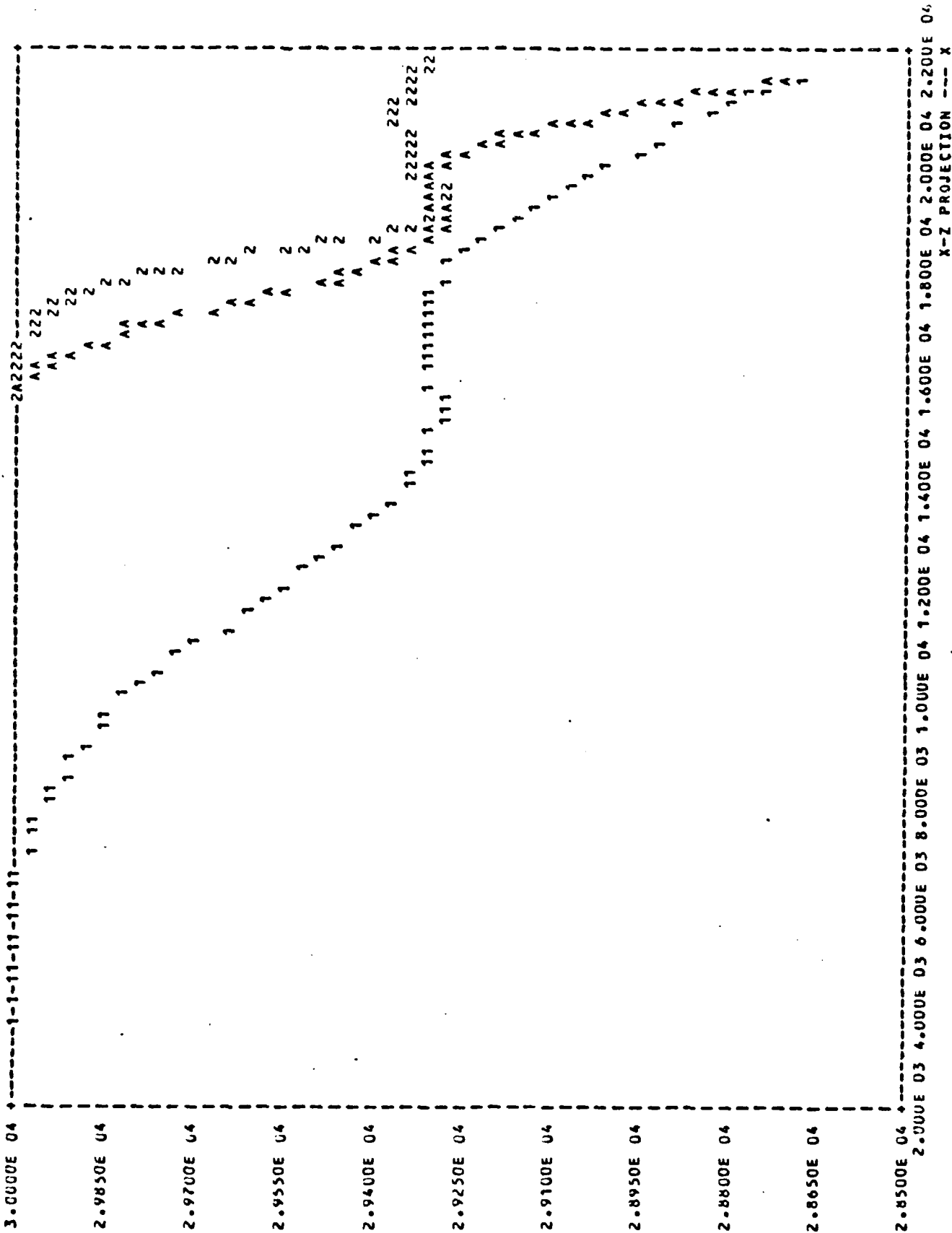


Figure 4-8e
Algorithm 1, Scenario 5

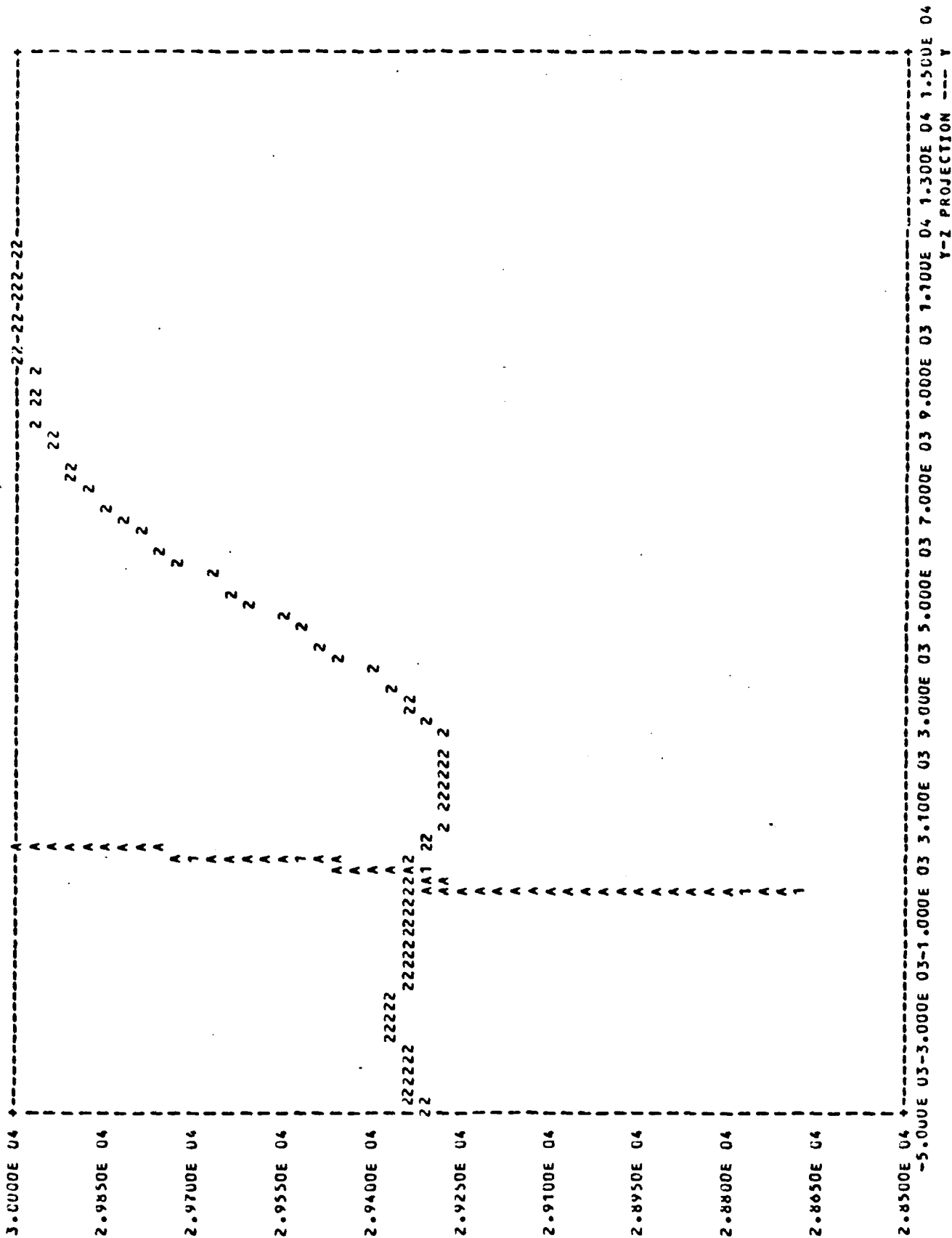


Table 4-9
Algorithm 2, Scenario 5

TSTEP = 0.1000, TAU = 0.1500

INIT XU(1) = 15000.0	INIT XU(7) = 3000.00	INIT XU(13) = 15000.0
INIT XU(2) = 0.000000	INIT XU(8) = 0.000000	INIT XU(14) = 12000.0
INIT XU(3) = 30000.0	INIT XU(9) = 30000.0	INIT XU(15) = 30000.0
INIT XU(4) = 1100.00	INIT XU(10) = 3300.00	INIT XU(16) = 3300.00
INIT XU(5) = 0.000000	INIT XU(11) = 0.000000	INIT XU(17) = 0.000000
INIT XU(6) = 0.000000	INIT XU(12) = 0.000000	INIT XU(18) = -60.0000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.120E 05	DSEP2 = 0.120E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.989E 04	DSEP2 = 0.914E 04
TIME = 2.000	DSEP1 = 0.794E 04	DSEP2 = 0.637E 04
TIME = 3.000	DSEP1 = 0.613E 04	DSEP2 = 0.368E 04
TIME = 4.000	DSEP1 = 0.445E 04	DSEP2 = 0.106E 04
TIME = 4.473	DSEP1 = 0.368E 04	DSEP2 = 130.
TIME = 5.511	DSEP1 = 0.234E 04	DSEP2 = 0.210E 04
TIME = 6.511	DSEP1 = 825.	DSEP2 = 0.418E 04
TIME = 6.683	DSEP1 = 19.5	DSEP2 = 0.525E 04

*** CLOSURE RATE NEGATIVE AT TIME = 6.683***

TA1 : BEST DSEP = 17.1382	, NOW = 19.4535
TA2 : BEST DSEP = 51.6636	, NOW = 5254.68

XU(1): 0.2091E 05	XU(7): 0.2092E 05	XU(13): 0.2037E 05
XU(2): 396.3	XU(8): 412.6	XU(14): -4824.
XU(3): 0.2959E 05	XU(9): 0.2959E 05	XU(15): 0.2984E 05
XU(4): 731.5	XU(10): 2071.	XU(16): 2043.
XU(5): -30.91	XU(11): -12.44	XU(17): -8.208
XU(6): -24.54	XU(12): -10.27	XU(18): -86.82

DELX1: -10.2	DELY1: -16.2
DELZ1: 3.35	DMIS1: 19.5
BEST DMIS WAS 17.1	

DELX2: 543.	DELY2: 0.522E 04
DELZ2: -252.	DMIS2: 0.525E 04
BEST DMIS WAS 51.7	

Figure 4-9a
Algorithm 2, Scenario 5

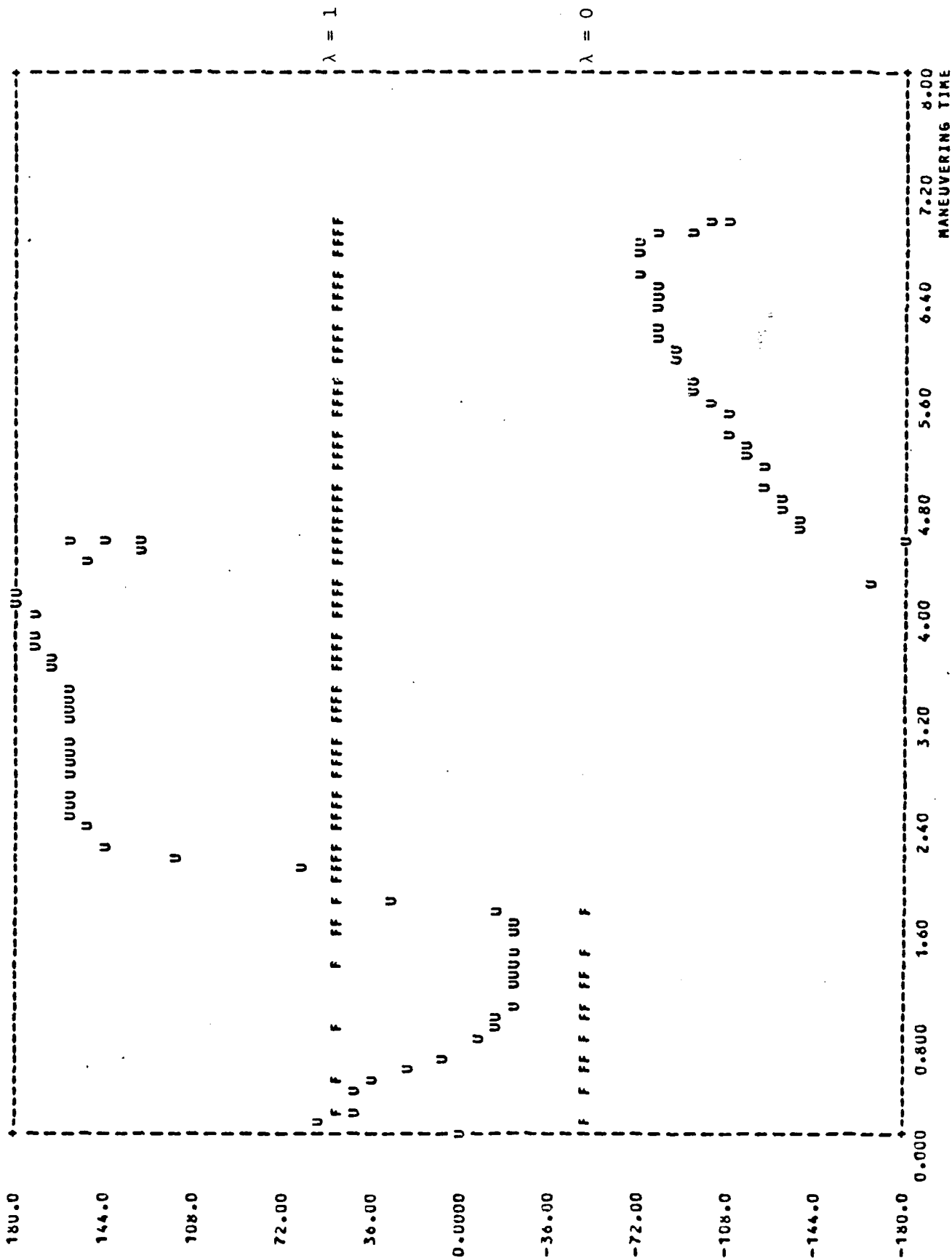


Figure 4-9b

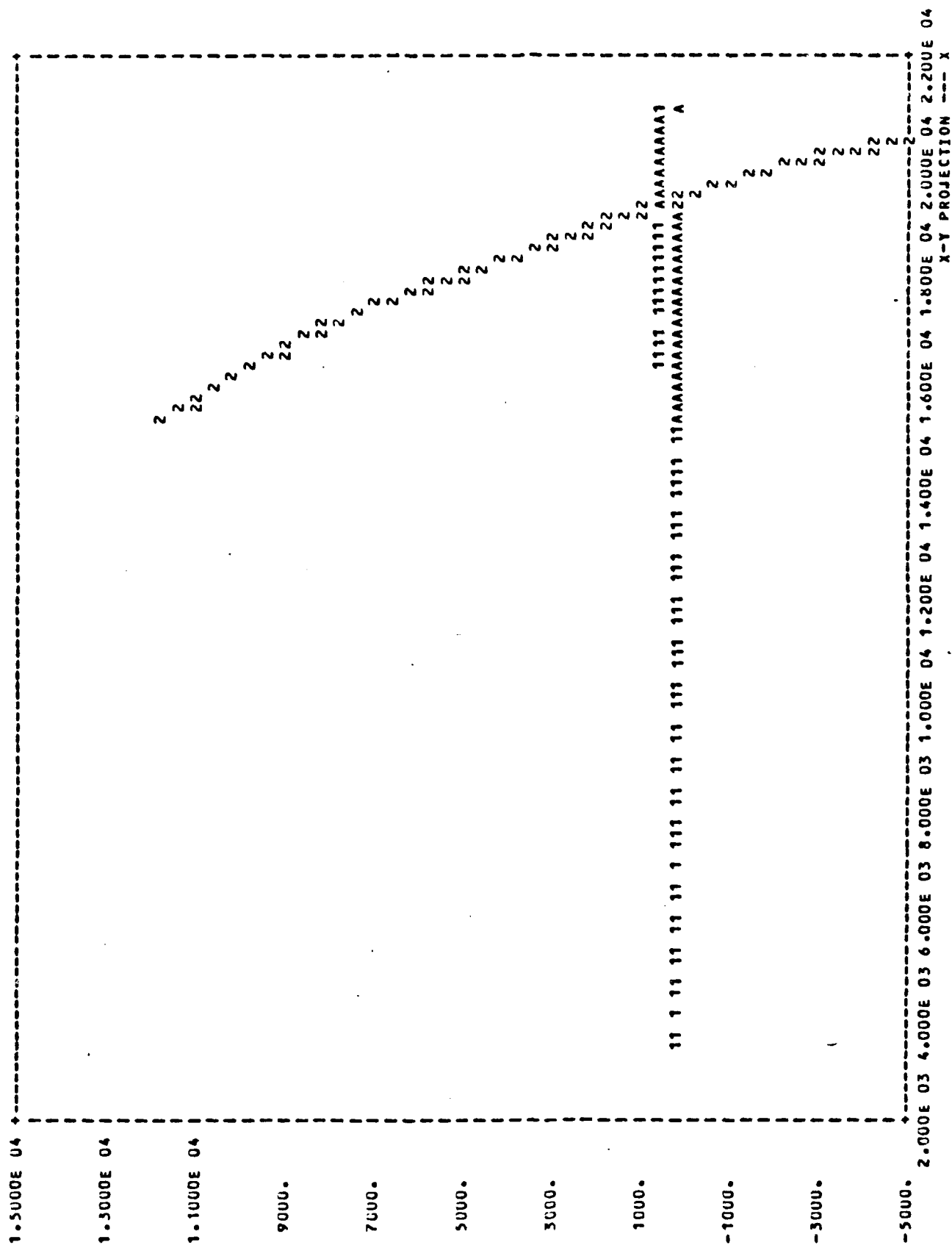


Figure 4-9c
Algorithm 2, Scenario 5

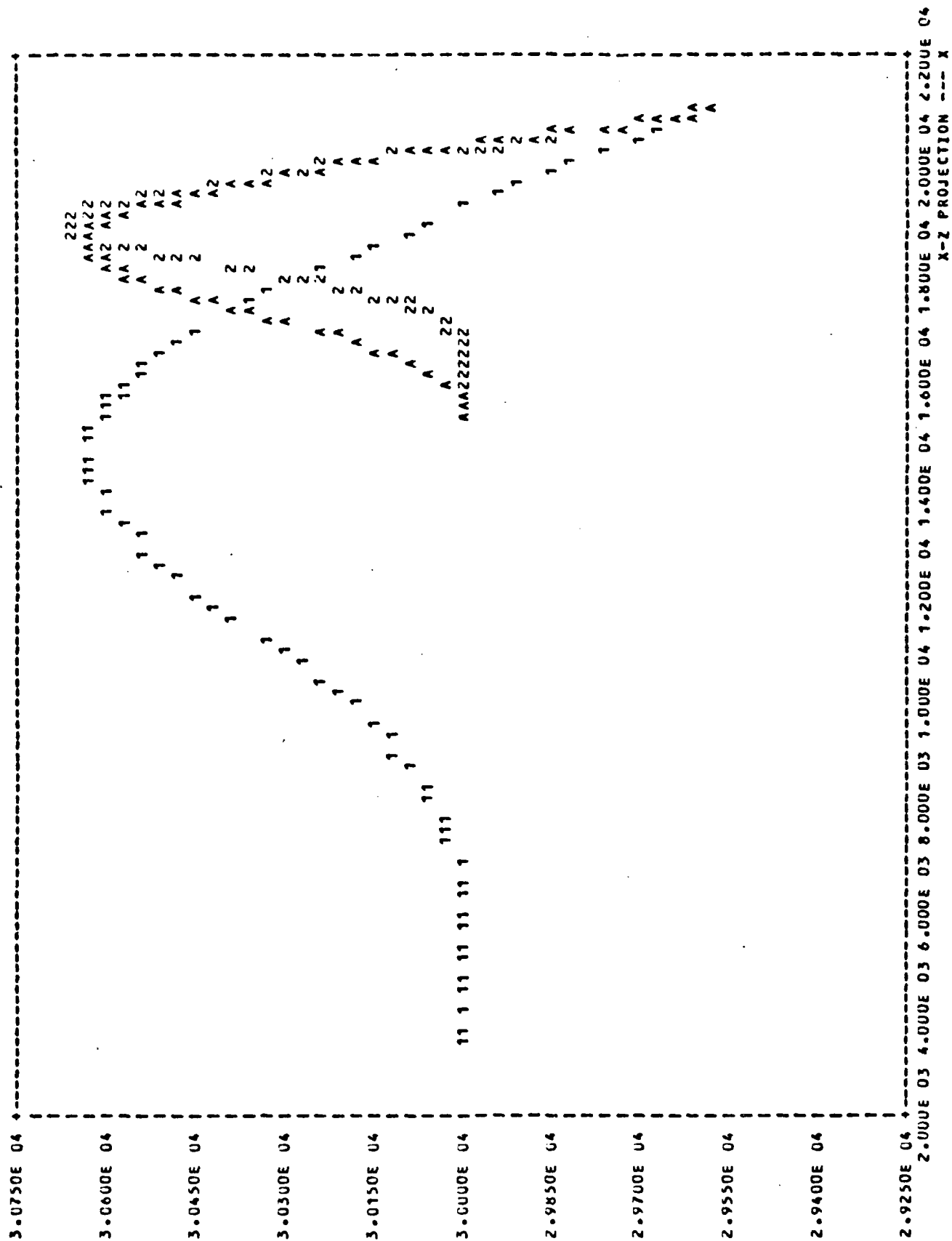


Figure 4-9d
Algorithm 2, Scenario 5

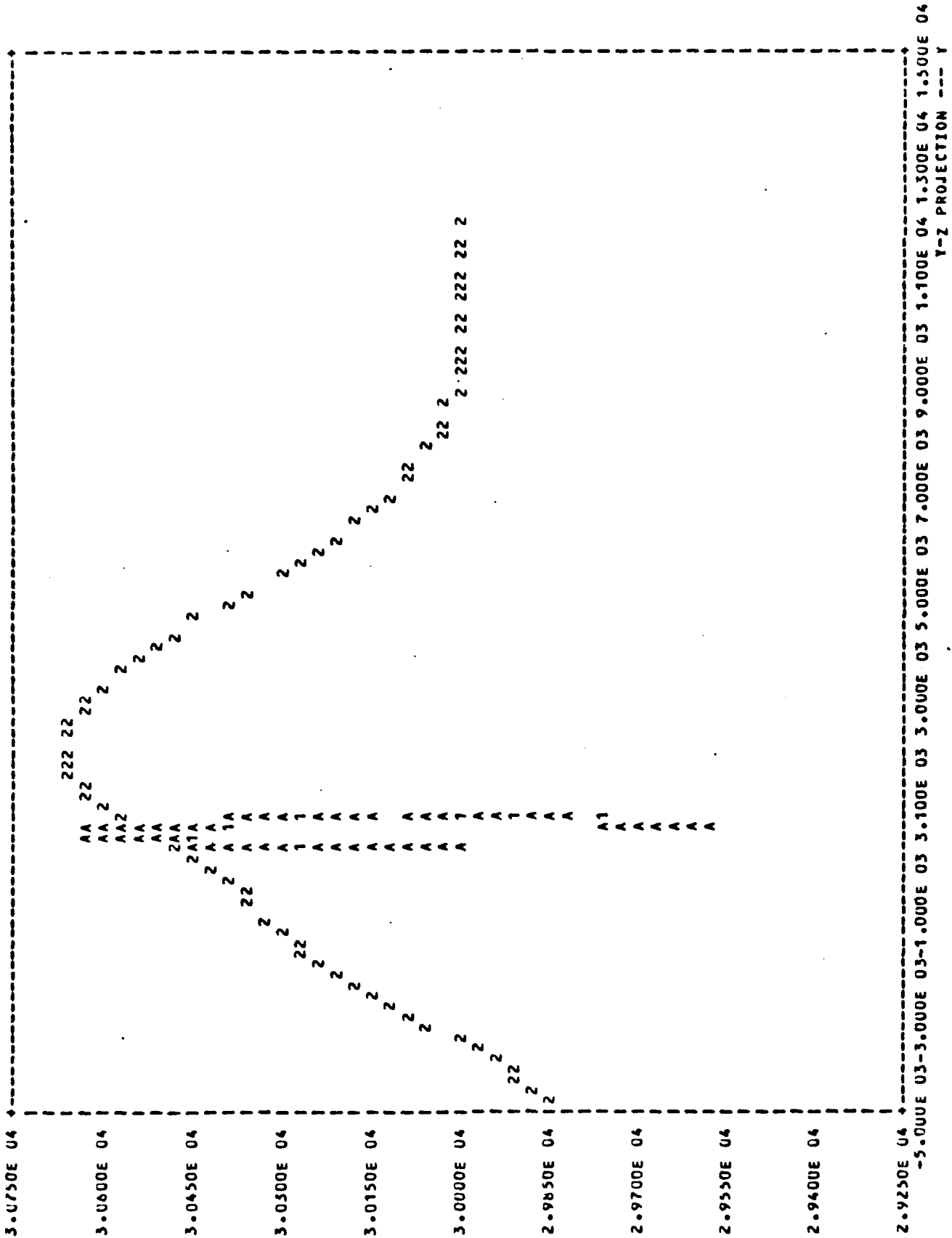


Table 4-10
Algorithm 1, Scenario 6

TSTEP = 0.1000, TAU = 0.1500

INIT XO(1) =	13000.0	INIT XO(7) =	30000.0	INIT XO(13) =	0.000000
INIT XO(2) =	0.000000	INIT XO(8) =	0.000000	INIT XO(14) =	0.000000
INIT XO(3) =	30000.0	INIT XO(9) =	33000.0	INIT XO(15) =	33000.0
INIT XO(4) =	1100.00	INIT XO(10) =	3300.00	INIT XO(16) =	3300.00
INIT XO(5) =	0.000000	INIT XO(11) =	-7.99999	INIT XO(17) =	-7.99999
INIT XO(6) =	0.000000	INIT XO(12) =	100.000	INIT XO(18) =	0.000000

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME =	0.000	DSEP1 =	0.173E 05	DSEP2 =	0.133E 05
START MANEUVER AT T =	0.10				
TIME =	1.000	DSEP1 =	0.130E 05	DSEP2 =	0.112E 05
TIME =	2.000	DSEP1 =	0.903E 04	DSEP2 =	0.926E 04
TIME =	3.000	DSEP1 =	0.531E 04	DSEP2 =	0.743E 04
TIME =	4.000	DSEP1 =	0.177E 04	DSEP2 =	0.572E 04
TIME =	4.539	DSEP1 =	72.9	DSEP2 =	0.483E 04
TIME =	5.130	DSEP1 =	0.197E 04	DSEP2 =	0.383E 04
TIME =	6.130	DSEP1 =	0.487E 04	DSEP2 =	0.236E 04
TIME =	7.130	DSEP1 =	0.749E 04	DSEP2 =	925.
TIME =	7.610	DSEP1 =	0.906E 04	DSEP2 =	22.1

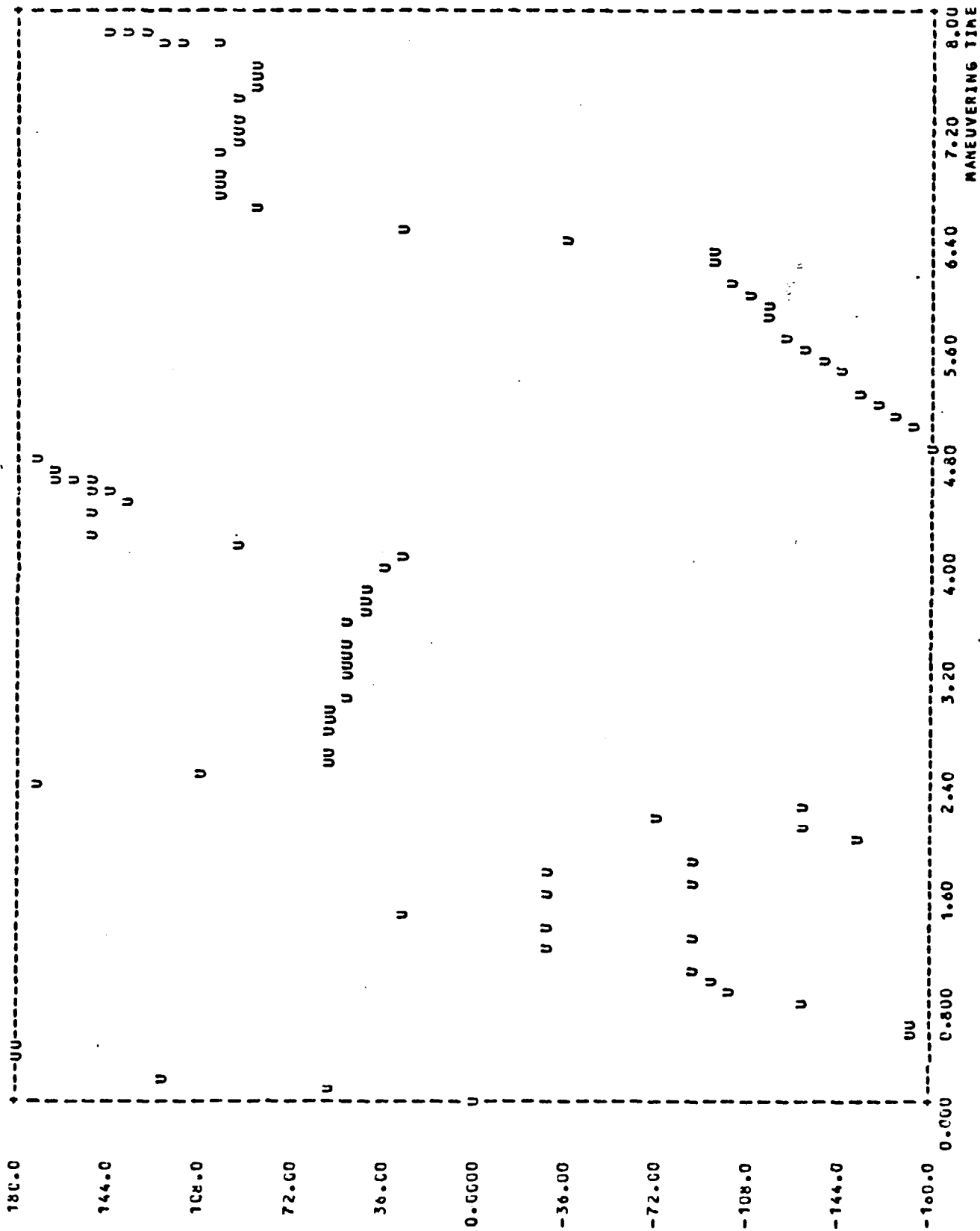
*** CLOSURE RATE NEGATIVE AT TIME = 7.618 ***

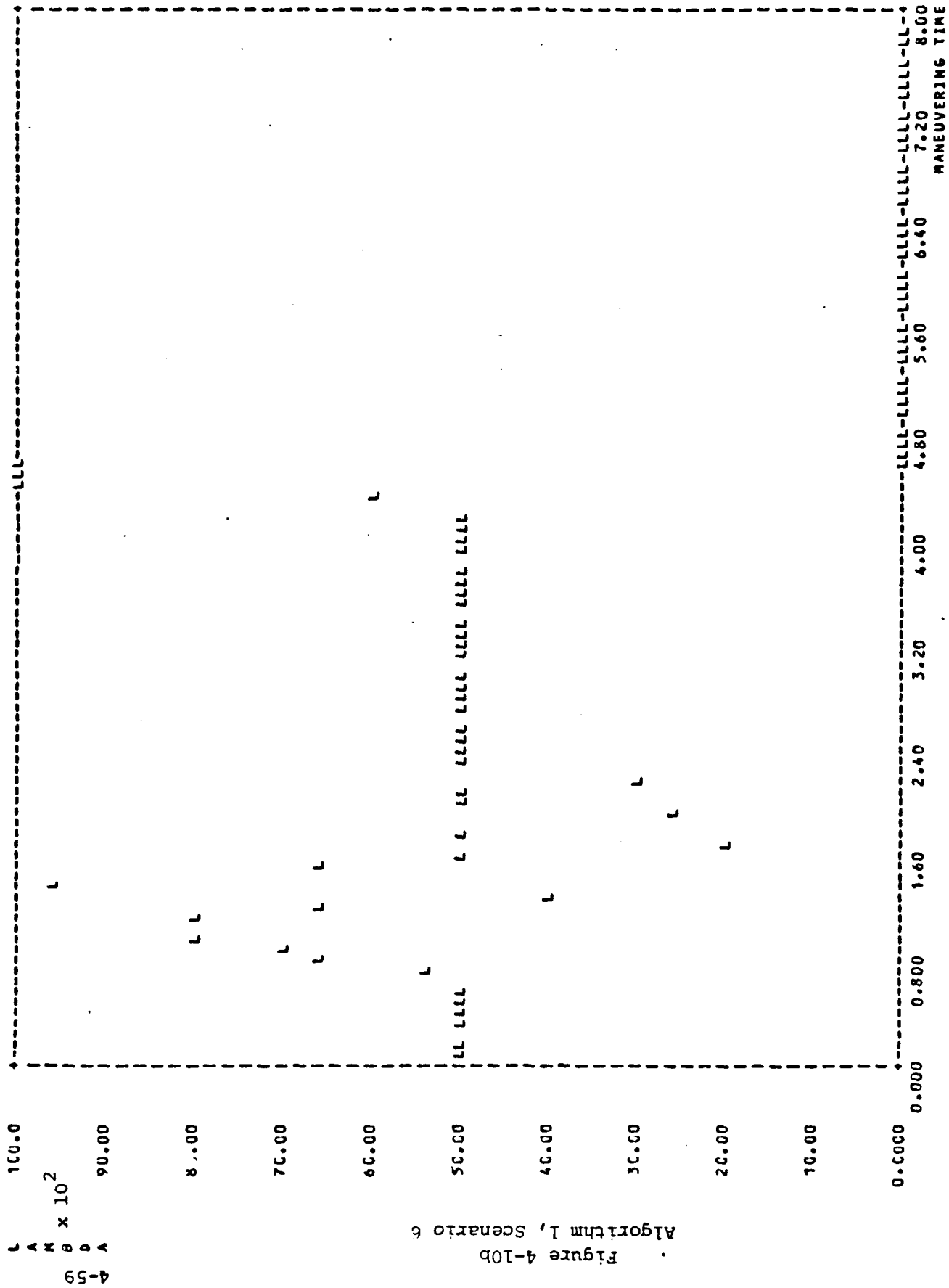
TA1 : BEST DSEP =	55.6756	, NOW =	9072.77
TA2 : BEST DSEP =	22.1066	, NOW =	24.5693

XO(1):	0.1993E 05	XO(7):	0.1120E 05	XO(13):	0.1994E 05
XO(2):	15.83	XO(8):	2086.	XO(14):	-5.526
XO(3):	0.2680E 05	XO(9):	0.2745E 05	XO(15):	0.2680E 05
XO(4):	713.0	XO(10):	1935.	XO(16):	2033.
XO(5):	-28.21	XO(11):	-19.04	XO(17):	-20.18
XO(6):	16.04	XO(12):	137.7	XO(18):	7.204

DELX1:	0.873E 04	DELY1:	-0.207E 04
DELZ1:	0.135E 04	DMIS1:	0.907E 04
BEST DMIS WAS	55.7		

DELA2:	-12.1	DELY2:	21.4
DELZ2:	0.590	DMIS2:	24.6
BEST DMIS WAS	22.1		





1

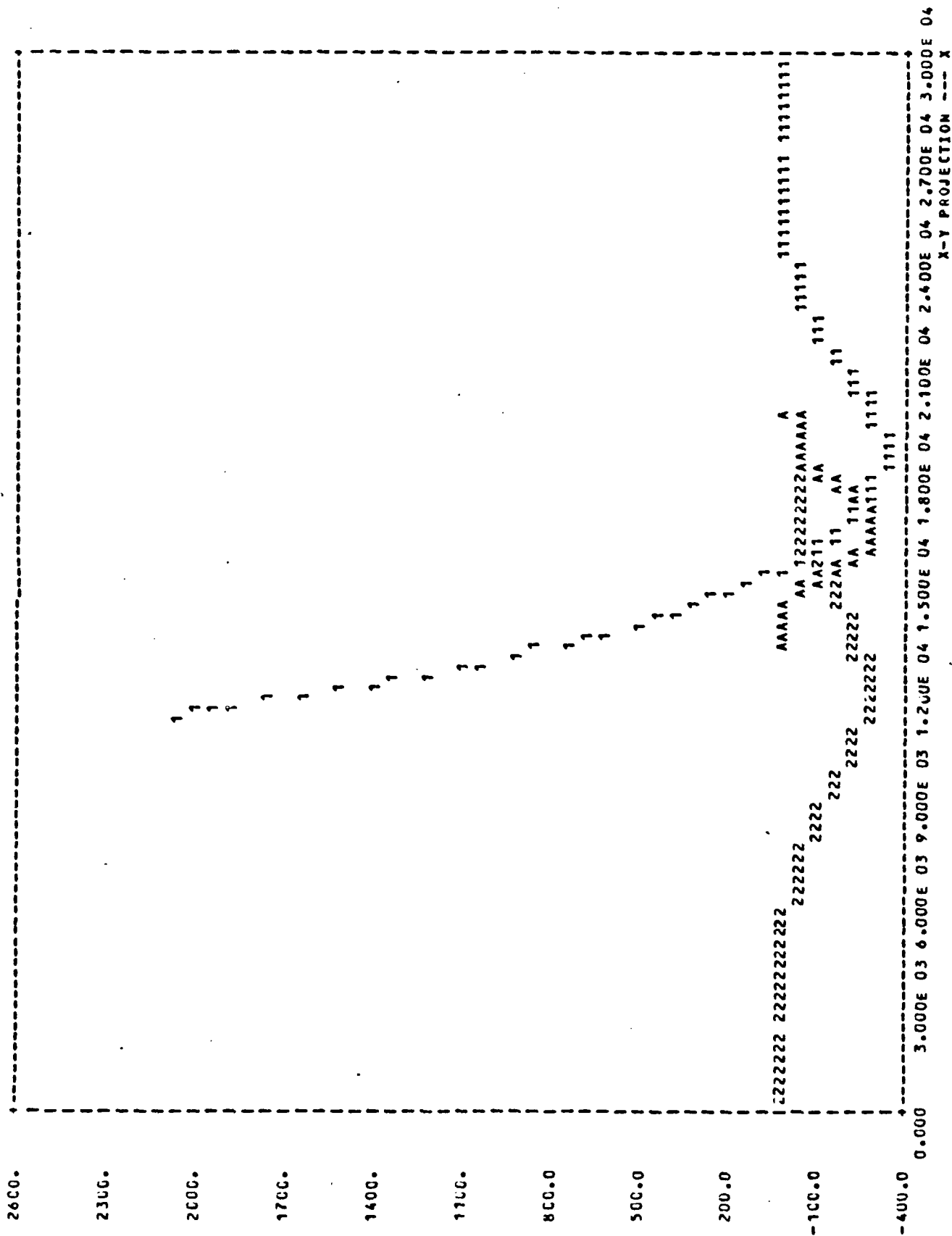
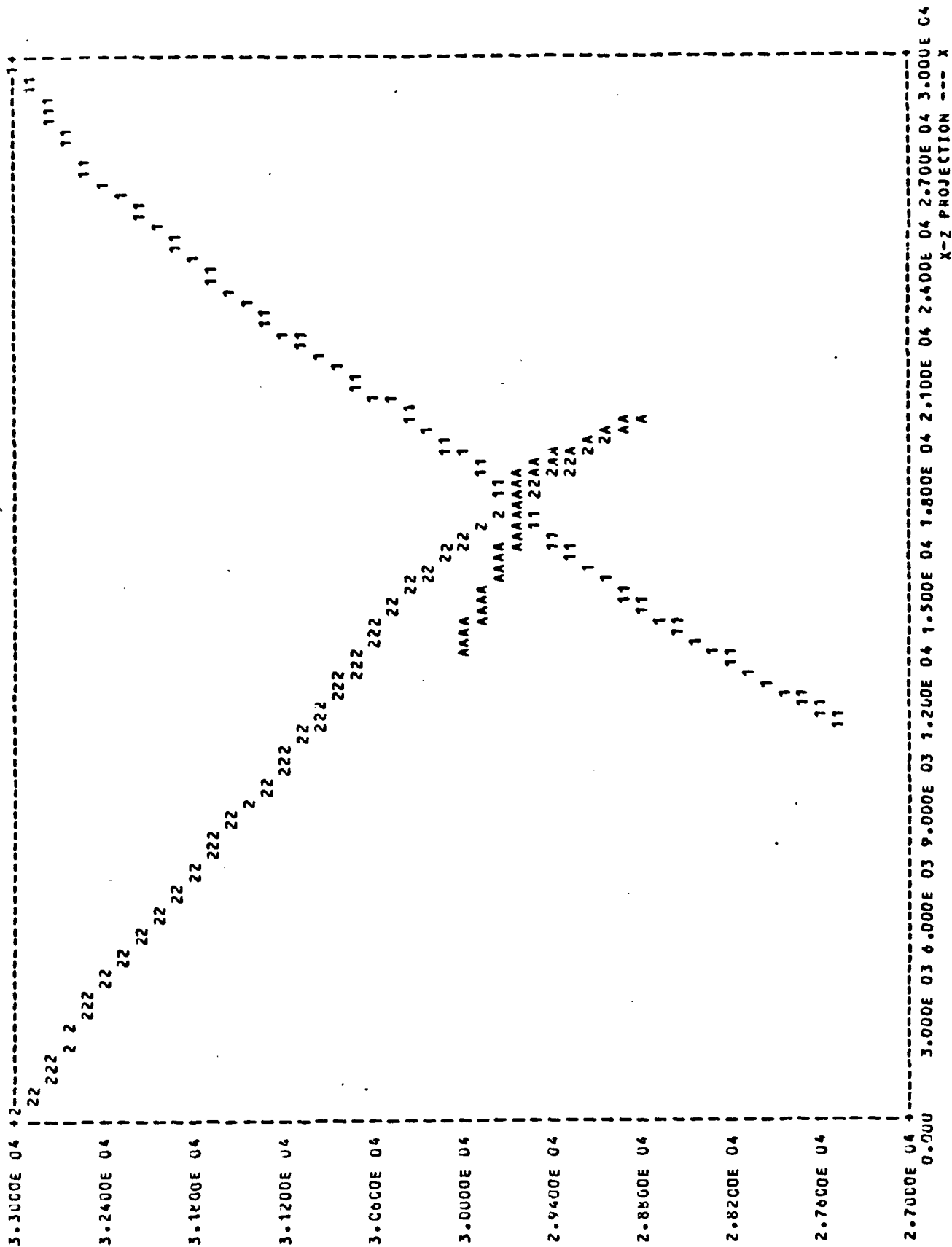


Figure 4-10d
Algorithm 1, Scenario 6



2

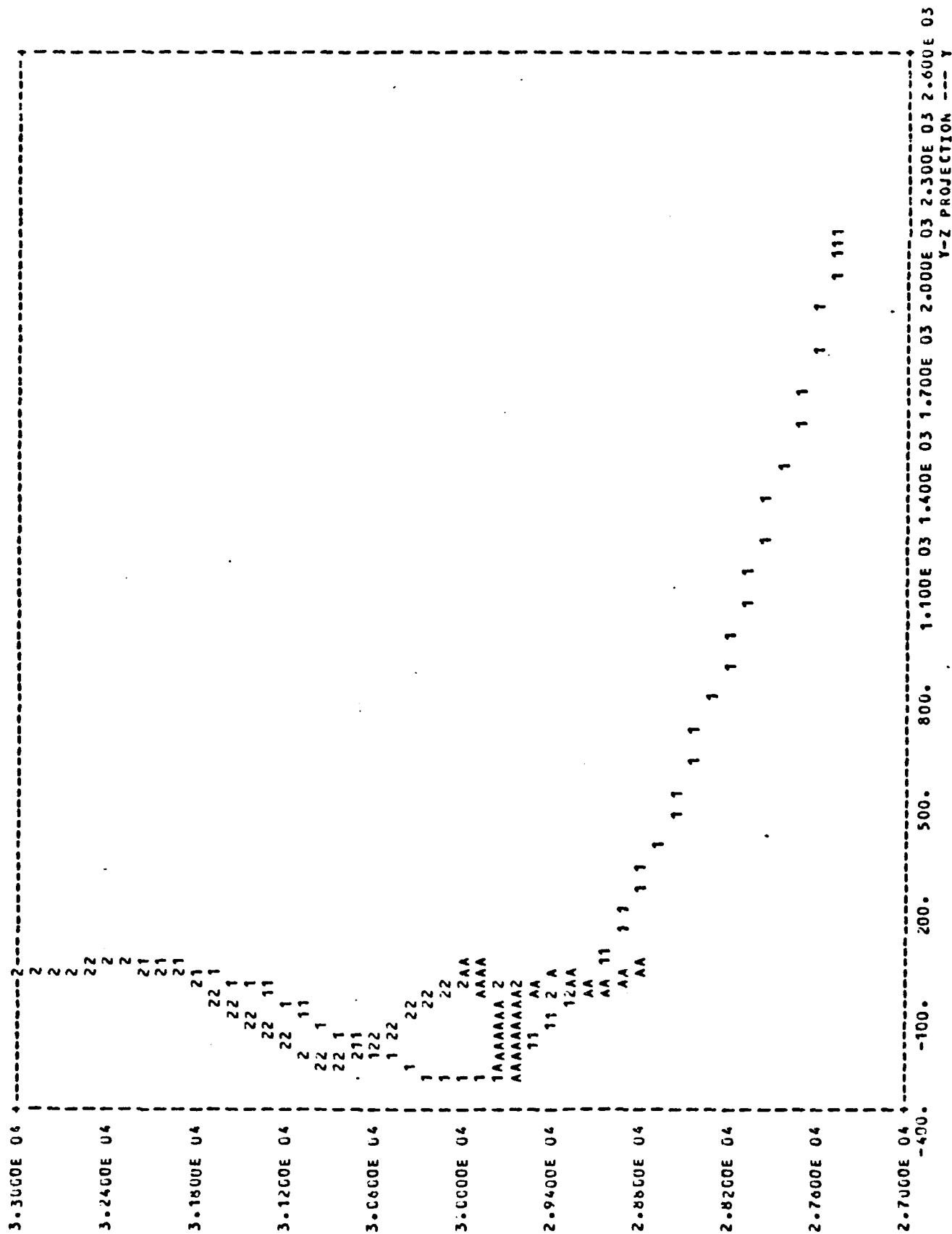


Table 4-11
Algorithm 4, Scenario 7

TSTEP = 0.1000, TAU = 0.1500

INIT XU(1) = 0.000000	INIT XO(7) = 15000.0	INIT XO(13) = 0.000000
INIT XU(2) = 0.000000	INIT XO(8) = 0.000000	INIT XO(14) = 15000.0
INIT XU(3) = 30000.0	INIT XO(9) = 30000.0	INIT XO(15) = 30000.0
INIT XU(4) = 1000.00	INIT XO(10) = 3300.00	INIT XO(16) = 3300.00
INIT XU(5) = 0.000000	INIT XO(11) = 0.000000	INIT XO(17) = 0.000000
INIT XU(6) = 45.0000	INIT XO(12) = 160.000	INIT XO(18) = -89.9998

PROPORTIONAL NAVIGATION GAINS:

PITCH (RK11) = 4.50 , YAW (RK12) = 4.50

A/C MAXIMUM LOAD FACTOR = 8.00

IN MANEUVER, AFTERBURNERS WILL BE ON

TIME = 0.000	DSEP1 = 0.150E 05	DSEP2 = 0.150E 05
START MANEUVER AT T = 0.10		
TIME = 1.000	DSEP1 = 0.112E 05	DSEP2 = 0.111E 05
TIME = 2.000	DSEP1 = 0.760E 04	DSEP2 = 0.759E 04
TIME = 3.000	DSEP1 = 0.434E 04	DSEP2 = 0.437E 04
TIME = 4.000	DSEP1 = 0.136E 04	DSEP2 = 0.146E 04

**** CLOSURE RATE NEGATIVE AT TIME = 4.588****

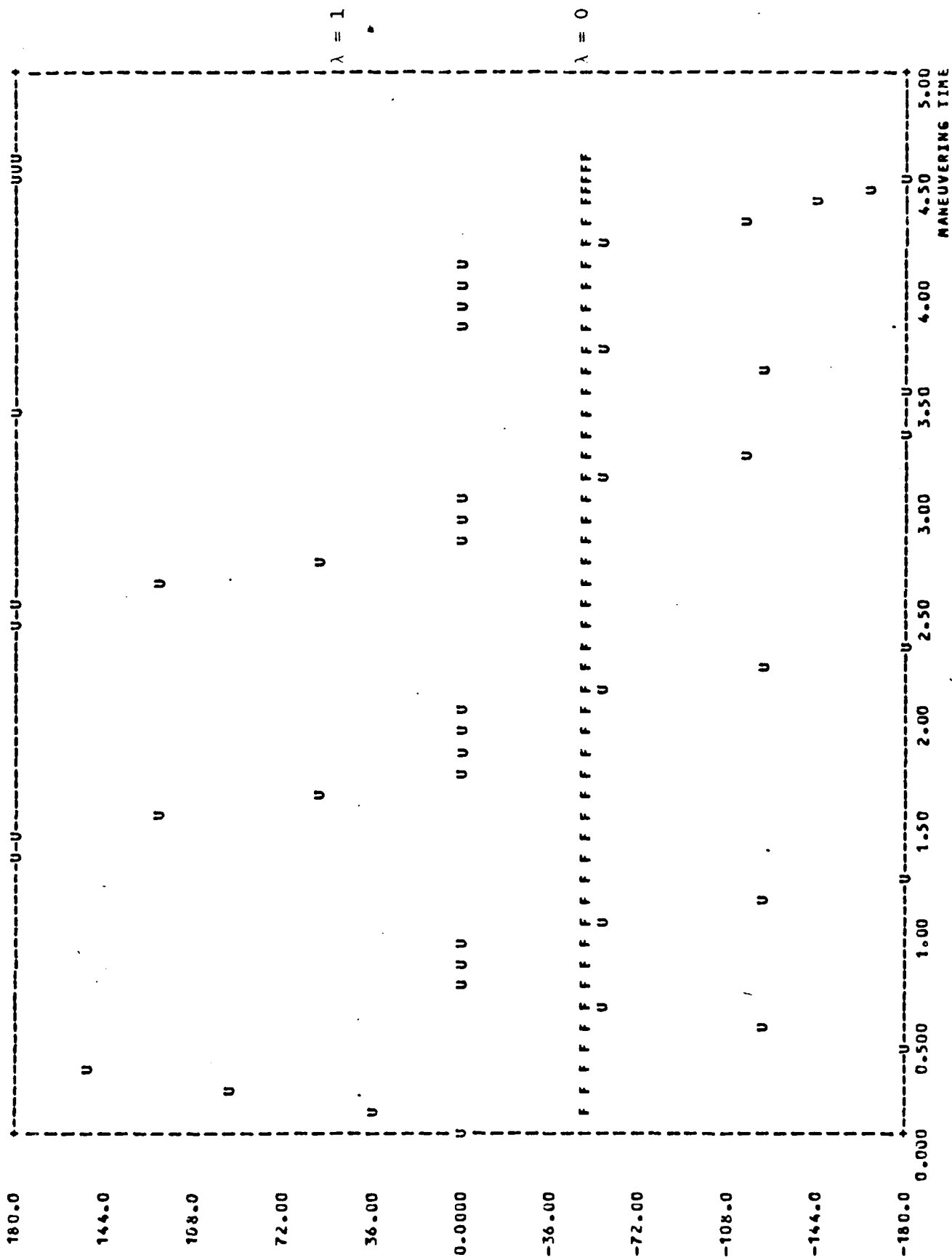
TA1 : BEST DSEP = 157.707	, NOW = 341.083
TA2 : BEST DSEP = 268.416	, NOW = 298.517

XU(1): 2833.	XO(7): 2552.	XO(13): 2648.
XU(2): 2731.	XO(8): 2652.	XO(14): 2548.
XU(3): 0.2982E 05	XO(9): 0.2981E 05	XO(15): 0.2981E 05
XO(4): 725.2	XO(10): 2371.	XO(16): 2370.
XU(5): -6.204	XO(11): -2.482	XO(17): -2.318
XO(6): 36.00	XO(12): 152.3	XO(18): -61.79

DELX1: 332.	DELY1: 79.2
DELZ1: 9.56	DMIS1: 341.
BEST DMIS WAS 158.	

DELX2: 236.	DELY2: 183.
DELZ2: 6.00	DMIS2: 299.
BEST DMIS WAS 268.	

Figure 4-11a
Algorithm 4, Scenario 7



Y

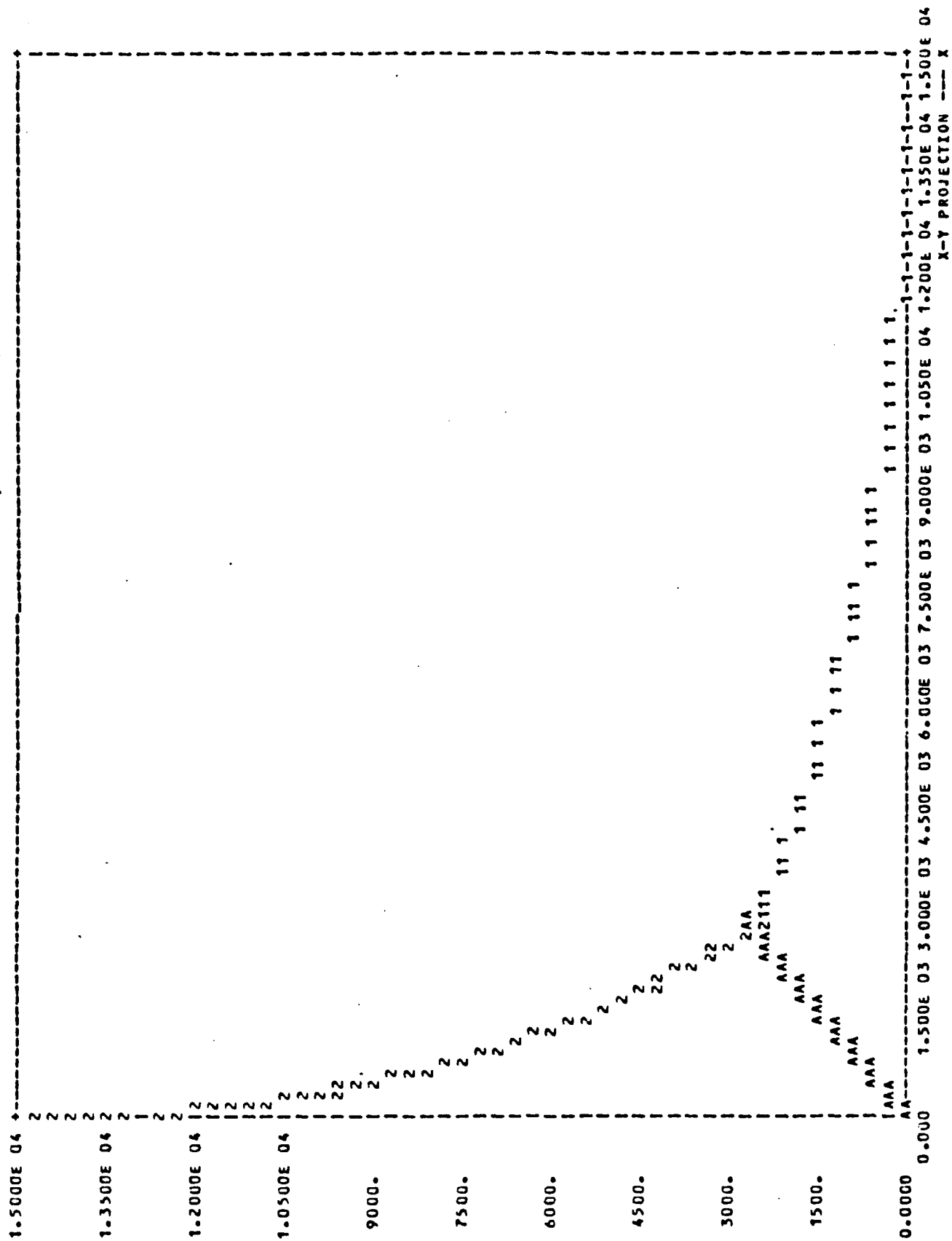


Figure 4-11c

Algorithm 4, Scenario 7

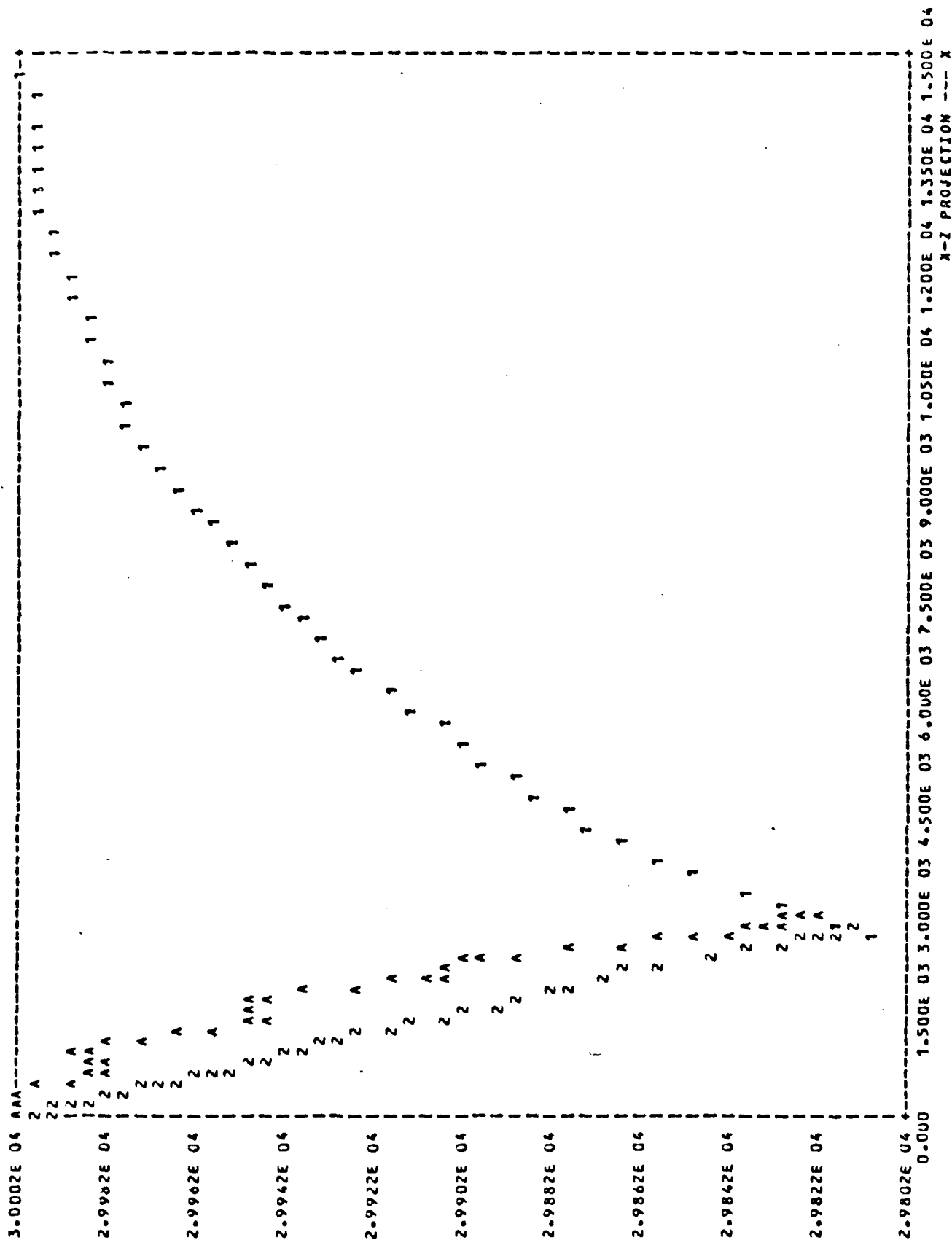
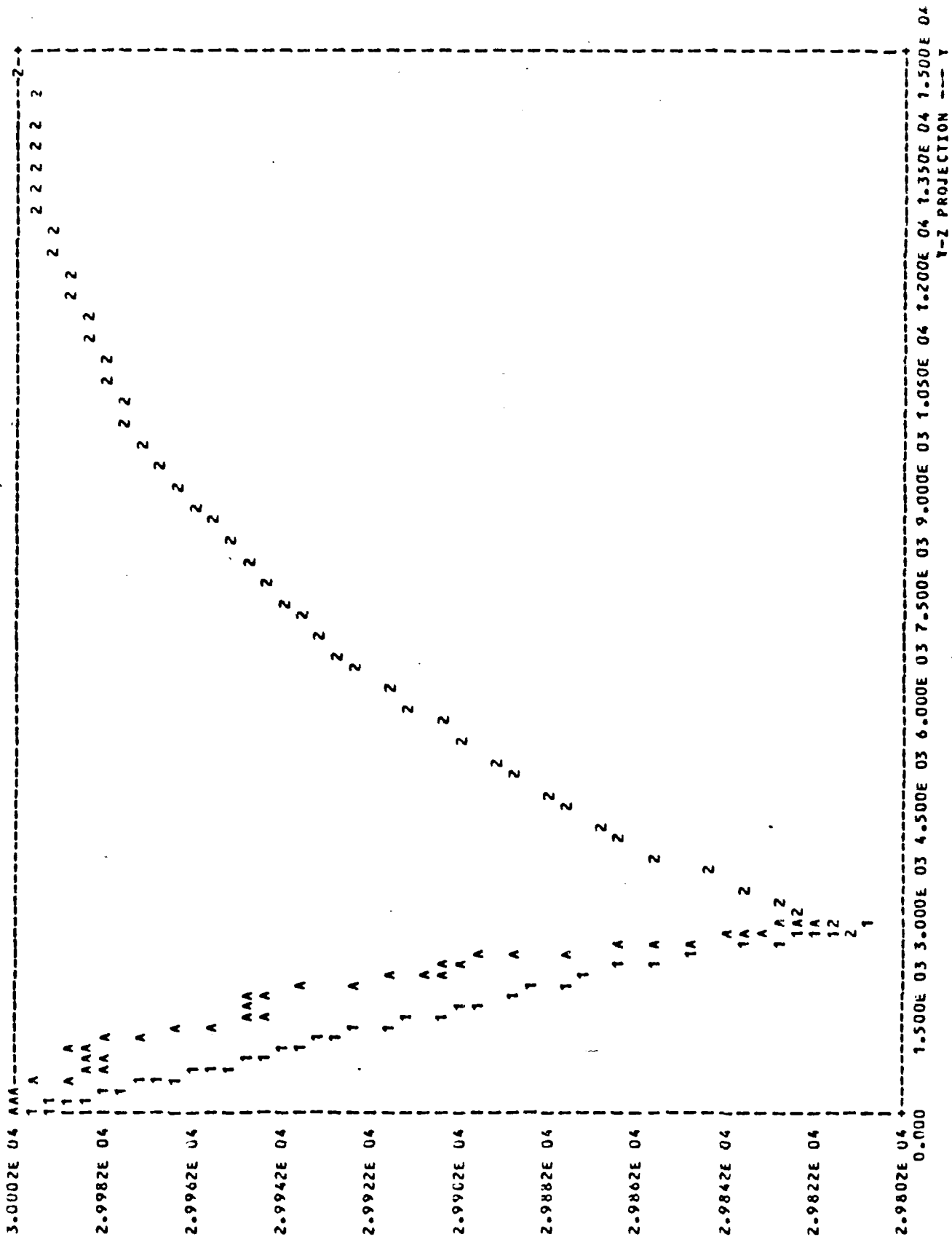


Figure 4-11d
Algorithm 4, Scenario 7



5.0 A Game Theoretic Model for Determining Aircraft Evasion Strategies Against a Multiple Missile Threat

Introduction

In this Chapter we consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. Based on various assumptions pertaining to dynamics, information patterns, and optimization criteria the existence, structure and behavior of a set of optimal evasion strategies are delineated. This study encompasses problem formulations with linear and nonlinear dynamics. In each case, the evader is assumed to know the guidance law for each pursuing missile. In this study two general classes of optimization criteria are considered: (i) fixed terminal time criteria and (ii) free terminal time criteria. In the former case, the evader seeks to maximize the terminal miss distance between himself and each pursuer. In the latter case, the evader seeks to maximize the distance of closest approach between himself and each pursuer. This decision problem is most naturally formulated as a multi-criterion or vector valued optimization problem. A game theoretic approach is taken in solving this class of problems.

The results in this chapter represent part of a Ph.D. Dissertation in Systems Engineering (Huling, 1979) and are the basis for a paper (Huling & Mintz, 1977). In the numbering convention for the following subsections in this chapter, we have omitted the leading (5.) designation.

1. OPTIMAL EVASION STRATEGIES AGAINST MULTIPLE MISSILES: PART I - FOR CRITERIA WITH A FIXED TERMINAL TIME

1.1 Introduction:

We consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. We delineate the existence, structure, and behavior of optimal evasion strategies for this problem based on the following assumptions pertaining to dynamics, information patterns, and optimization criteria.

Dynamics:

- (a) The evader's and pursuer's dynamics are each linear, i.e.,

$$\dot{x}_e = F_e x_e + G_e u_e$$

$$\dot{x}_{pi} = F_{pi} x_{pi} + G_{pi} u_i, \quad i = 1, 2.$$

- (b) The evader's and pursuer's dynamics are each nonlinear, i.e.,

$$\dot{x}_e = f_e(x_e, u_e, t)$$

$$\dot{x}_{pi} = f_{pi}(x_{pi}, u_i, t), \quad i = 1, 2.$$

Information:

Each pursuer ($i = 1, 2$) uses a given feedback control law for its guidance strategy. The evader's a priori information includes a complete description of each dynamical system including initial state information.

Optimization Criterion:

The evader seeks to maximize the "miss distance" between himself and each pursuer at a given terminal time T . Since there are two pursuers, the evader is faced with a multi-criterion or vector valued optimization problem. One approach to solving this multi-criterion problem is to seek an evasion strategy

which maximizes the minimum terminal miss distance. This maxmin or game theoretic approach requires that we solve a saddle-point problem to obtain the optimal evasion strategy.

In order to obtain a physically meaningful solution, we must either modify the original problem statement to include a set of constraints on the evader's permissible controls, or modify the pay-off functions to include a cost to the evader for using energy for evasive maneuvering. We shall take this later approach initially and assume that the evader's control function is weighted quadratically in each of the individual criteria. In section 3 and subsequent sections, we consider optimal evasion problems with a variety of constraints on the evader's permissible controls.

2. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND QUADRATIC COST ON CONTROL

2.1 Mathematical Problem Statement:

Let $x \triangleq (x_e, x_{p1}, x_{p2})'$,

then one obtains

$$\dot{x} = Fx + G_0 u_0 + G_1 u_1 + G_2 u_2, \quad x(0) \triangleq x_0$$

where F and G_i $i = 0, 1, 2$ denote partitioned matrices defined in terms of F_e, F_{p1}, G_e , and G_i $i = 1, 2$, and where $u_0 = u_e$.

We have assumed

$$u_i = S_i x; \quad i = 1, 2.$$

Hence

$$\dot{x} = \tilde{F}x + G_0 u_0; \quad x(0) = x_0$$

where $\tilde{F} = F + G_1 S_1 + G_2 S_2$.

Define $J_1(u_0, x_0)$ and $J_2(u_0, x_0)$ by:

$$J_1 = x'(T)C_1 x(T) - \int_0^T u_0' D_0 u_0 dt$$

$$J_2 = x'(T)C_2 x(T) - \int_0^T u_0' D_0 u_0 dt$$

where $C_i \geq 0$ and $D_0 > 0$.

In what follows, we shall consider x_0 a fixed vector and write $J_i(u_0, x_0)$ as

$$J(u_0, i) \quad i = 1, 2.$$

How can J_i be
made a
function of
 u_0 alone?

2.2 Game Theoretic Formulation:

Let the space of permissible open loop controls u_0 be $L^2[0,T]$ and let $I \triangleq \{1,2\}$. Let

$$J: L^2[0,T] \times I \rightarrow \mathbb{R}$$

denote the kernel of an abstract game.

Consider the following saddle-point problem (SPP):

$$\text{SPP1: Does } \max_{u_0 \in L^2[0,T]} \min_{i \in I} J(u_0, i) = \min_{i \in I} \max_{u_0 \in L^2[0,T]} J(u_0, i)?$$

Discussion (SPP1):

Due to the discrete nature of the set $I = \{1,2\}$, SPP1 will generally not have a saddle-point in pure strategies. This situation can often be resolved however by considering mixed strategies over I . This leads naturally to SPP2.

SPP2:

Let $p \in [0,1]$ and define $\tilde{J}(u_0, p) \triangleq pJ(u_0, 1) + (1-p)J(u_0, 2)$.

Does

$$\max_{u_0 \in L^2[0,T]} \min_{p \in [0,1]} \tilde{J}[u_0, p] = \min_{p \in [0,1]} \max_{u_0 \in L^2[0,T]} \tilde{J}[u_0, p]?$$

Discussion (SPP2):

For conceptual reasons, it is useful to consider p as the probability that strategy $i = 1$ is played. Then

*The missile is launched
or not
at random.*

$$\begin{aligned}
E[J(u_0, 1)] &= \tilde{J}(u_0, p), \quad \tilde{J}(u_0, p) = pJ_1 + (1-p)J_2 \\
&= x'(T)\tilde{C}x(T) - \int_0^T u_0' \tilde{D}_0 u_0 dt, \\
\text{where} \quad \tilde{C} &= pC_1 + (1-p)C_2.
\end{aligned}$$

In order to investigate the properties of the game denoted by $\{\tilde{J}(u_0, p), L^2[0, T], [0, 1]\}$ it is useful to rewrite $\tilde{J}(u_0, p)$ as follows.

First observe that

$$x(T) = Lx_0 + Lu_0$$

where $L \triangleq \phi(T, 0)$ and ϕ is the state transition matrix associated with $\tilde{F}$, and where $L: L^2_T[0, T] \rightarrow R^n$ is the linear operator defined by

$$Lu_0 = \int_0^T \phi(T, \tau) G_0 u_0 d\tau.$$

Hence we can write that

$$\tilde{J}(u_0, p) \triangleq \|Lx_0 + Lu_0\|_{\tilde{C}}^2 - \|u_0\|_{\tilde{D}_0}^2,$$

where the first norm is w.r.t. R^n and the second norm is w.r.t. $L^2[0, T]$.

2.3 Properties of $\tilde{J}: L^2[0, T] \times [0, 1] \rightarrow R$:

Observation 1:

If for all $p \in [0, 1]$, the matrix Riccati equation

$$\dot{R} = -\tilde{F}'R - R\tilde{F} - R\tilde{G}_0\tilde{D}_0^{-1}\tilde{G}_0'R; \quad R(T) = \tilde{C}$$

has a bounded solution on $[0, T]$, then $\tilde{J}[u_0, p]$ is concave in u_0 for every $p \in [0, 1]$.

We will denote the existence of bounded solutions to this Riccati equation for all $p \in [0, 1]$ as Condition 1.

An Aside: If Condition 1 does not hold, a relaxed version may hold on a subset of $[0,1]$, i.e., for only some values of p .

Observation 2:

If Condition 1 holds, then for fixed p , $\tilde{J}[u_0, p]$ is weakly upper semi-continuous in u_0 on $L^2[0, T]$.

Observation 3:

For fixed $u_0 \in L^2[0, T]$, $\tilde{J}[u_0, p]$ is continuous in p on $[0, 1]$.

Observation 4:

For fixed $u_0 \in L^2[0, T]$, $\tilde{J}(u_0, p)$ is convex (affine) in p on $[0, 1]$.

Observation 5:

$[0, 1]$ is a compact convex subset of R .

Observation 6:

$L^2[0, T]$ is convex.

Lemma 1:

If Condition 1 holds, then $\sup_{u_0} \inf_p \tilde{J}(u_0, p) = \inf_p \sup_{u_0} \tilde{J}(u_0, p)$.

Proof: It follows from Observations 1-6 that $\tilde{J}(u_0, p)$ is quasi-concave-quasi-convex in (u_0, p) and u.s.c. - l.s.c. Further, $L^2[0, T]$ and $[0, 1]$ are convex spaces, and $[0, 1]$ is compact in R . Therefore, Lemma 1 follows as a consequence of Sion's Theorem [Sion: 1958].

Lemma 2:

If Condition 1 holds, then $\min_p \sup_{u_0} \tilde{J}(u_0, p) = \sup_{u_0} \min_p \tilde{J}(u_0, p)$

i.e., there exists a value of p (say p^*) such that

$$\sup_{u_0} \tilde{J}(u_0, p^*) = \inf_p \sup_{u_0} \tilde{J}(u_0, p).$$

Proof: Lemma 2 is a consequence of Lemma 1 and the compactness of $[0, 1]$.

Observation 7:

If Condition 1 holds, there exists a value of u_0 (say u_0^*) such that

$$\sup_{u_0} \tilde{J}(u_0, p^*) = \tilde{J}(u_0^*, p^*).$$

An Aside: Observation 7 is a consequence of the assumption of Condition 1.

Theorem 1: The pair (u_0^*, p^*) defined through Lemma 2 and Observation 7 constitute a saddle-point for $\tilde{J}(u_0, p)$.

Proof: The verification that $\tilde{J}(u_0, p^*) \leq \tilde{J}(u_0^*, p^*) \leq \tilde{J}(u_0^*, p)$ follows directly from the definition of (u_0^*, p^*) in Lemma 2 and Observation 7.

3. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND A TOTAL ENERGY CONSTRAINT:

3.1 Problem Statement:

Let

$$\dot{x} = \tilde{F}x + G_0 u_0 ; x(0) \triangleq x_0 \dagger$$

where

$$u_0 \in \Omega_e ;$$

$$\Omega_e = \{u_0 \in L^2[0, T] : \|u_0\|_{D_e}^2 \leq e^2\}.$$

Define J_1 and J_2 by

$$J_1 = x'(T)C_1 x(T) - \int_0^T u_0' D_0 u_0 dt$$

$$J_2 = x'(T)C_2 x(T) - \int_0^T u_0' D_0 u_0 dt.$$

Consider the following Maxmin Problem:

$$\max_{u_0 \in \Omega_e} \min_{i \in I} J(u_0, i) .$$

The sole difference between this problem and that considered in the previous section is the restriction of u_0 to the weakly compact set $\Omega_e \subset L^2[0, T]$,

$$\Omega_e \triangleq \{u_0 \in L^2[0, T] : \|u_0\|_{D_e}^2 \leq e^2\}.$$

Reasoning as before, we are lead to consider

$$\max_{u_0 \in \Omega_e} \min_{p \in [0, 1]} \tilde{J}(u_0, p)$$

$\dagger \tilde{F}$ was defined in Section 2.

where $\tilde{J}(u_0, p) = pJ(u_0, 1) + (1-p) J(u_0, 2)$.

3.2 Problem Solution:

Consider the following saddle-point problem (SPP3):

Does

$$\max_{u_0 \in \Omega_e} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega_e} \tilde{J}(u_0, p)?$$

Observation 1:

Ω_e is a convex weakly compact subset of $L^2[0, T]$.

Condition 1:

The Riccati matrix differential equation

$$\dot{R} = -\tilde{F}'R - R\tilde{F} - R G_0 D_0^{-1} G_0' R$$

$$R(T) = \tilde{C} \underline{A} p C_1 + (1-p) C_2$$

has a bounded solution on $[0, T]$ for all $p \in [0, 1]$.

Now using the results associated with SPP2, we have:

Theorem 1:

If Condition 1 holds, then

$$\max_{u_0 \in \Omega_e} \min_{p \in [0,1]} \tilde{J}[u_0, p] = \min_{p \in [0,1]} \max_{u_0 \in \Omega_e} \tilde{J}[u_0, p].$$

Proof:

The proof again follows from Sion's Theorem [Sion: 1958] by noting that:

(a) Ω_e is a convex weakly compact subset of $L^2[0,T]$ and $[0,1]$ is a convex compact subset of $\mathbb{R}$.

(b) $\tilde{J}: \Omega_e \times [0,1] \rightarrow \mathbb{R}$

is quasi-concave-convex, weakly u.s.c. in u_0 for fixed p , and continuous in p for fixed u_0 .

3.3 Solution Structure:

Lemma 1:

Let (u_0^*, p^*) denote any saddle-point solution to SPP3, then

$$u_0^* = \arg \max_{u_0 \in \Omega_e} \tilde{J}(u_0, p^*)$$

where:

$$u_0^* = (D_0 + \alpha^* D_e)^{-1} G_0^T k x,$$

$$\dot{k} = -\tilde{F}^T k - k\tilde{F} - kG_0(D_0 + \alpha^* D_e)^{-1} G_0^T k,$$

$$k(T) = \tilde{C} = p^* C_1 + (1-p^*) C_2,$$

$$\text{and } \alpha^* \left(\int_0^T u_0^{*T} D_e u_0^* dt - e^2 \right) = 0,$$

$$\alpha^* \geq 0.$$

Proof: Lemma 1 follows as a consequence of the Linear-Quadratic (LQ) nature of the problem setting and the application of standard multiplier theory. (See for example [Luenberger: 1969, P. 217].)

4. A SECOND OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND A TOTAL ENERGY CONSTRAINT:

4.1 Problem Statement:

Let

$$\dot{x} = \tilde{F}x + G_0 u_0; \quad x(0) \triangleq x_0$$

where

$$u_0 \in \Omega_e;$$

$$\Omega_e = \{u_0 \in L^2[0, T]: \|u_0\|_{D_e}^2 \leq e^2\}.$$

Define J_1 and J_2 by

$$J_1 = x'(T)C_1 x(T)$$

$$J_2 = x'(T)C_2 x(T).$$

Consider the following Maxmin Problem:

$$\max_{u_0 \in \Omega_e} \min_{i \in I} J(u_0, i).$$

The sole difference between this problem and that considered in the previous section is the elimination of the quadratic terms in u_0 from the pay-off functions J_1 and J_2 .

Reasoning as before, we are lead to consider

$$\max_{u_0 \in \Omega_e} \min_{p \in [0, 1]} \tilde{J}(u_0, p)$$

where

$$\begin{aligned}\tilde{J}(u_0, p) &= pJ(u_0, 1) + (1-p)J(u_0, 2) \\ &= x'(T)[pC_1 + (1-p)C_2]x(T).\end{aligned}$$

4.2 Problem Solution:

Consider the following saddle-point problem (SPP4):

Does

$$\max_{u_0 \in \Omega_e} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega_e} \tilde{J}(u_0, p)?$$

Lemma 1: For fixed $p \in [0,1]$, $J(u_0, p)$ is w -continuous on Ω_e .

Proof: For notational simplicity, we consider the scalar case, i.e.,

$$\dim(x) = \dim(u_0) = 1.$$

We note that

$$x(T) = \phi(T, 0)x_0 + \int_0^T \phi(T, \tau)G_0 u_0 d\tau$$

where ϕ is the state transition function associated with $\tilde{F}$. Hence, $x(T)$ can be expressed as

$$x(T) = a + \langle b, u_0 \rangle$$

where $a \in \mathbb{R}$, $b \in L^2[0, T]$ and $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $L^2[0, T]$.

(Here we are of course assuming that

$$\phi(T, t)G_0 \in L^2[0, T].$$

Therefore, $x^2(T)$ is w -continuous for all $u_0 \in L^2[0, T]$ since $x(T)$ is w -continuous in u_0 . Finally,

$$\tilde{J}(u_0, p) = x^2(T) [pC_1 + (1-p)C_2],$$

and therefore, $\tilde{J}(u_0, p)$ is w -continuous on $L^2[0, T]$ for fixed $p \in [0, 1]$.

Observation 1:

$\tilde{J}(u_0, p) \triangleq x^2(T) [pC_1 + (1-p)C_2]$ is continuous in the pair (u_0, p) when

$\tilde{J}: \Omega_e \times [0, 1] \rightarrow \mathbb{R}$, where Ω_e is endowed with the weak topology. This follows from the product structure of $\tilde{J}$, where we observe that the product of the limits of two sequences is the limit of product of the individual sequences.

Observation 2:

$\tilde{J}: \Omega_e \times [0, 1] \rightarrow \mathbb{R}$ is continuous in the pair (u_0, p) in the general case - $\dim(x) = n$, $\dim(u) = r$; where Ω_e is endowed with the weak topology.

Observation 2 follows by noting that in the general case the components of $x(T)$ can be considered as a linear combination of underlying linear functionals.

Theorem 1:

Let M_u and M_p denote respectively the set of all probability measures μ_e (respectively μ_p) on Ω_e (respectively $[0, 1]$). Then,

$$\max_{\mu_e \in M_u} \min_{\mu_p \in M_p} E[\tilde{J}(u_0, p)] = \min_{\mu_p \in M_p} \max_{\mu_e \in M_u} E[\tilde{J}(u_0, p)],$$

where the expectation operation is with respect to the measures μ_e and μ_p .

Comment:

The interpretation of this theorem is that the saddle-point problem, (SPP4) has a solution in mixed strategies. In what follows, we will in fact show that (SPP4) has a solution in pure strategies. To do this, we will need Theorem 1 as a preliminary result.

Proof of Theorem 1:

Since $\tilde{J}(u_0, p)$ is continuous in the pair (u_0, p) and Ω_e and $[0, 1]$ are respectively w -compact and compact then Theorem 1 is a direct consequence of a general saddle-point theorem [See [Owen: 1968] (Theorem IV.6.1)].

Theorem 2:

Let (μ_e^*, μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by $\tilde{J}: \Omega_e \times [0, 1] \rightarrow \mathbb{R}$. Let $p^* \triangleq E_{\mu_p^*}[p]$. If $\arg \max_{\substack{u_0 \in \Omega_e \\ \mu_p^*}} \tilde{J}(u_0, p^*)$ is essentially unique, then, the pair (μ_e^*, μ_p^*) are one point, i.e., μ_e^* and μ_p^* correspond to pure strategies.

Proof:

Since $\tilde{J}[u_0, p]$ is affine in p for fixed u_0 , $p^* \triangleq E_{\mu_p^*}[p]$ can be viewed as an optimal pure strategy since

$$E_{\mu_p^*}[\tilde{J}(u_0, p)] = \tilde{J}(u_0, p^*).$$

Let $u_0^* = \arg \max_{u_0 \in \Omega_e} \tilde{J}(u_0, p^*)$. Since u_0^* is essentially unique by assumption, the pure strategy u_0^* is equivalent to the optimal mixed strategy μ_e^* .

Lemma 1:

$$u_0^* = \frac{1}{\alpha^*} D_e^{-1} G_0^T \tilde{R} x$$

where

$$\dot{\tilde{R}} = -\tilde{F}^T \tilde{R} - \tilde{R} \tilde{F} - \frac{1}{\alpha^*} \tilde{R} G_0 D_e^{-1} G_0^T \tilde{R}$$

$$\tilde{R}(T) \triangleq \tilde{C} = p^* C_1 + (1-p^*) C_2, \text{ and}$$

$$\int_0^T u_0^* D_e u_0^* dt = e^2.$$

Proof: Lemma 1 follows as a consequence of the Linear-Quadratic (LQ) nature of the problem setting and the application of standard multiplier theory. (See for example [Luenberger: 1969, P. 220].)

5. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND HARD CONSTRAINTS ON CONTROL:

5.1 Problem Statement:

Let

$$\dot{x} = Fx + G_0 u_0; \quad x(0) \triangleq x_0$$

where

$$u_0 \in \Omega_b; \quad \Omega_b \triangleq \{ u_0 \in L^2[0, T] : a_i \leq u_i(t) \leq b_i \}, \text{ and } u_i \text{ denotes the } i^{\text{th}} \text{ component of the vector } u_0.$$

Define J_1 and J_2 by

$$J_1 = x'(T) C_1 x(T)$$

$$J_2 = x'(T) C_2 x(T) \quad .$$

Consider the following Maxmin Problem:

$$\max_{u_0 \in \Omega_b} \min_{i \in I} J(u_0, i).$$

The sole difference between this problem and that considered in the previous section is the replacement of Ω_e with Ω_b .

Reasoning as before, we are lead to consider

$$\max_{u_0 \in \Omega_b} \min_{p \in [0, 1]} \tilde{J}(u_0, p)$$

where $\tilde{J}(u_0, p) = x'(T) [pC_1 + (1-p)C_2] x(T)$.

5.2 Problem Solution:

Consider the following saddle-point problem (SPP5):

Does

$$\max_{u_0 \in \Omega_b} \min_{p \in [0, 1]} \tilde{J}(u_0, p) = \min_{p \in [0, 1]} \max_{u_0 \in \Omega_b} \tilde{J}(u_0, p)?$$

In order to answer SPP5, we will first formulate and solve a related saddle-point problem SPP6:

We begin with the following definition:

Definition:

$$\bar{X}(T, x_0) \triangleq \{x(T) \in E^n : \dot{x} = \tilde{F}x + G_0 u_0; x(0) \triangleq x_0; u_0 \in \Omega_b\}$$

Observation 1:

$\bar{X}(T, x_0)$ is a compact convex subset of E^n .

Consider the following saddle-point problem (SPP6):

Does

$$\max_{x(T) \in \bar{X}(T, x_0)} \min_{p \in [0,1]} x'(T) \tilde{C}x(T) = \min_{p \in [0,1]} \max_{x(T) \in \bar{X}(T, x_0)} x'(T) \tilde{C}x(T) ?$$

Theorem 1:

Let M_x and M_p denote respectively the set of all probability measures μ_x

(respectively μ_p) on $\bar{X}(T, x_0)$ (respectively $[0,1]$). Then,

$$\max_{\mu_x \in M_x} \min_{\mu_p \in M_p} E[x'(T) \tilde{C}x(T)] = \min_{\mu_p \in M_p} \max_{\mu_x \in M_x} E[x'(T) \tilde{C}x(T)],$$

where the expectation operation is with respect to the measures μ_x and μ_p .

Proof:

The proof follows by noting that $\bar{X}(T, x_0)$ is compact in E^n , $[0,1]$ is compact in R , and $x'(T) \tilde{C}x(T)$ is continuous in the pair $(x(T), p)$. Hence the conclusion follows as a consequence of Theorem IV.6.1 [Owen: 1968].

Theorem 2:

Let (μ_x^*, μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by:

$$k: \bar{X}(T, x_0) \times [0,1] \rightarrow \mathbb{R}, \quad k(x(T), p) = x'(T) \tilde{C}x(T).$$

Let $p^* \triangleq E_{\mu_p^*}[p]$. If $x^*(T) = \arg \max_{x(T) \in \bar{X}(T, x_0)} k(x(T), p^*)$ is unique, then the pair (μ_x^*, μ_p^*) are one-point, i.e., μ_x^* and μ_p^* correspond to pure strategies.

Proof:

Since $k(x(T), p)$ is affine in p for fixed $x(T)$, $p^* \triangleq E_{\mu_p^*}[p]$ can be viewed as an optimal pure strategy since

$$E_{\mu_p^*}[k(x(T), p)] = k(x(T), p^*).$$

$$\text{Let } x^*(T) \triangleq \arg \max_{x(T) \in \bar{X}(T, x_0)} k(x(T), p^*).$$

Since $x^*(T)$ is unique by assumption, the pure strategy $x^*(T)$ is equivalent to the optimal mixed strategy μ_x^* .

Theorem 3:

$$\text{Let } u_0^* \text{ denote any control in } \Omega_b \text{ which achieves } x(T) = x^*(T), \text{ then}$$

$$\max_{u_0 \in \Omega_b} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega_b} \tilde{J}(u_0, p) = \tilde{J}(u_0^*, p^*),$$

i.e. (u_0^*, p^*) is a saddle-point pair for SPP5. [$x^*(T)$ and p^* are defined in Theorem 2].

Proof: The proof follows directly by noting that $(x^*(T), p^*)$ is a saddle-point pair for the game defined by:

$$k: \bar{X}(T, x_0) \times [0,1] \rightarrow \mathbb{R},$$

$$k(x(T), p) = x'(T) \tilde{C}x(T).$$

6. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS AND HARD CONSTRAINTS ON CONTROL:

6.1 Problem Statement:

Let $\dot{x} = f(x, u, t)$ $x(0) \triangleq x_0$ where $u \in \Omega$;

$\Omega = \{u_0: u_0(t) \in E^r; u_i \text{ Lebesgue measurable on } [0, T], a_i \leq u_i(t) \leq b_i, i = 1, \dots, r\}$.

Define J_1 and J_2 by

$$J_1 = x'(T) C_1 x(T)$$

$$J_2 = x'(T) C_2 x(T).$$

Consider the following maxmin problem:

$$\max_{u_0 \in \Omega} \min_{i \in I} J(u_0, i).$$

The sole difference between this problem and that considered in the previous section is that the dynamics are now allowed to be possibly nonlinear.

Reasoning as before we are lead to consider

$$\max_{u_0 \in \Omega} \min_{p \in [0, 1]} \tilde{J}(u_0, p)$$

where $\tilde{J}(u_0, p) = x'(T) [pC_1 + (1-p)C_2] x(T)$.

Consider the following saddle-point problem (SPP7):

Does

$$\max_{u_0 \in \Omega} \min_{p \in [0, 1]} \tilde{J}(u_0, p) = \min_{p \in [0, 1]} \max_{u_0 \in \Omega} \tilde{J}(u_0, p)?$$

In order to answer SPP7, we will first formulate and solve a related saddle-point problem SPP8. This approach parallels our analysis in section 5.

We begin with the following definition:

$$\bar{X}(T, x_0) \triangleq \{x(T) \in E^n : \dot{x} = f(x, u_0, t); x(0) = x_0; u_0 \in \Omega\}$$

Assumption 1:

Assume $\bar{X}(T, x_0)$ is compact in E^n .

Consider the following saddle-point problem (SPP8):

Does

$$\max_{u_0 \in \Omega} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega} \tilde{J}(u_0, p)?$$

Theorem 1:

Let M_x and M_p denote respectively the set of all probability measures μ_x (respectively μ_p) on $\bar{X}(T, x_0)$ (respectively $[0,1]$). Then,

$$\max_{\mu_x \in M_x} \min_{\mu_p \in M_p} E[x'(T) \tilde{C}x(T)] = \min_{\mu_p \in M_p} \max_{\mu_x \in M_x} E[x'(T) \tilde{C}x(T)]$$

where the expectation operation is with respect to the measures μ_x and μ_p .

Proof:

The proof of this result is precisely that of Theorem 1, section 5.

Theorem 2:

Let (μ_x^*, μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by:

$$k: \bar{X}(T, x_0) \times [0, 1] \rightarrow R,$$

$$k(x(T), p) = x'(T) C x(T).$$

Let $p^* = E_{\mu_p^*}[p]$. If $x^*(T) \triangleq \arg \max_{\bar{X}(T, x_0)} k(x(T), p^*)$ is unique then the pair

(μ_x^*, μ_p^*) are one-point i.e., μ_x^* and μ_p^* correspond to pure strategies.

Proof:

The proof of this result is precisely that of Theorem 2, section 5.

Theorem 3:

Let u_0^* denote any control in Ω which achieves $x(T) = x^*(T)$, then

$$\max_{u_0 \in \Omega} \min_{p \in [0, 1]} \tilde{J}(u_0, p) = \tilde{J}(u_0^*, p^*) = \min_{p \in [0, 1]} \max_{u_0 \in \Omega} \tilde{J}(u_0, p).$$

Hence, the pair (u_0^*, p^*) is a saddle-point for the game defined by $\tilde{J}(u_0, p)$

(SPP7), where $x^*(T)$ and p^* are defined in Theorem 3.

Proof:

The proof of this result is precisely that of Theorem 3, section 5.

6.3 Some Comments on Theorem 1-3, Sections 5 & 6:

In order to extend the saddle-point results for the case of linear dynamics (section 5) to include nonlinear dynamics (section 6), we have assumed that the set of attainment $\bar{X}(T, x_0)$ is compact in E^n (Assumption 1, section 6). In this section we will consider sufficient conditions to guarantee the compactness of $\bar{X}(T, x_0)$. The following results are due to Filippov (Filippov; 1962).

We begin with several definitions:

Let $\dot{x} = f(x, u, t)$, $x(0) = x_0$ where $\dim(x) = n$, $\dim(u) = r$. Suppose each component u_i of u is a measurable function of t satisfying $a_i \leq u_i(t) \leq b_i$, $0 \leq t \leq T$. Let U denote the permissible range of u , i.e. $U = \prod_{i=1}^r [a_i, b_i]$. Suppose that f is continuous in all arguments, and is continuously differentiable with respect to x , and that

$$x'f(x, u, t) \leq C[1 + \|x\|^2] \text{ for some } C \text{ and all } t, x, u \in U.$$

Let $R(t, x)$ denote the following set valued function:

$$R(t, x) = \{f(x, u, t) : u \in U\}$$

Theorem 4 (Filippov):

If f , and U are defined as above, and $R(t, x)$ is convex for all t , $0 \leq t \leq T$, and $x \in E^n$, then $\overline{X}(T, x_0)$ is compact.

Comment: Theorem 4 is also true when $u(t) \in U(t)$, where $U(t)$ is compact in E^r , and the set valued function $U(\cdot)$ is t -continuous in the Hausdorff Topology, for all t , $0 \leq t \leq T$.

Proof: See [Filippov; 1962], or [Hermes and LaSalle; 1969].

7. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS, HARD CONSTRAINTS ON CONTROL, AND A GENERAL TERMINAL COST FUNCTION:

7.1 Problem Statement:

Let $\dot{x} = f(x, u_0, t)$, $x(0) = x_0$, where $u_0 \in \Omega$;

$\Omega \triangleq \{u_0 : u_0(t) \in E^r; u_1 \text{ Lebesgue measurable on } [0, T], a_i \leq u_i(t) \leq b_i, i = 1, 2, \dots, r\}$, and $U \triangleq \prod_{i=1}^r [a_i, b_i] \subset E^r$.

Let $J_1(x(T))$ and $J_2(x(T))$ denote terminal cost functions.

Assumptions:

A_1 : J_1 and J_2 are continuous maps of E^n into R .

A_2 : $f: E^n \times U \times [0, T] \rightarrow E^n$ is continuous in all arguments, is continuously differentiable in its first argument (x);

$x'f(x, u, t) \leq C[1 + \|x\|^2]$ for some C and all $x \in E^n$, $u \in U$, and $t \in [0, T]$.

A_3 : If $R(t, x) \triangleq \{f(x, u, t) : u \in U\}$, then

the set $R(t, x)$ is convex for all pairs (t, x) where $0 \leq t \leq T$, $x \in E^n$.

A_4 : Let $\bar{X}(t, x_0)$ and $\bar{Y}$ denote

$\bar{X}(T, x_0) \triangleq \{x(T) \in E^n : \dot{x} = f(x, u_0, t), x(0) = x_0, u_0 \in \Omega\}$

$\bar{Y} \triangleq \{(J_1(x(T)), J_2(x(T))) \in E^2 : x(T) \in \bar{X}(T, x_0)\}$,

then $\bar{Y}$ is a convex set.

Consider the following maxmin problem:

$$\max_{u_0 \in \Omega} \min_{i \in I} J_i(x(T)).$$

The sole difference between this problem and that considered in the previous section is that the functions $J_1(x(T))$ are now allowed to be arbitrary continuous mappings.

Reasoning as before, we are lead to consider

$$\max_{u_0 \in \Omega} \min_{p \in [0,1]} \tilde{J}(u_0, p)$$

where $\tilde{J}(u_0, p) = pJ_1(x(T)) + (1-p)J_2(x(T))$.

7.2 Problem Solution:

Consider the following saddle-point problem (SPP9):

Does

$$\max_{u_0 \in \Omega} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega} \tilde{J}(u_0, p)?$$

Theorem 1:

Under Assumptions $A_1 - A_4$, SPP9 has a solution in pure strategies.

Proof: We begin with several observations.

Observation 1:

The set $\bar{Y}$ is compact in E^2 . This observation follows by noting that $\bar{Y}$ is the image of $\bar{X}(T, x_0)$ under $\hat{J}$, where $\hat{J}(x(T)) = (J_1(x(T)), J_2(x(T)))$, $\hat{J}$ is continuous (Assumption A_1), and $\bar{X}(T, x_0)$ is compact in E^n (Theorem 4, section 6).

Observation 2:

Let $k: \bar{Y} \times [0,1] \rightarrow R$, $k(y, p) = py_1 + (1-p)y_2$.

Since k is continuous in (y,p) , convex (affine) in p for fixed y , concave (affine) in y for fixed p , and $\bar{Y}$, $[0,1]$ are compact convex sets then by Sion's Theorem [Sion: 1958], the game defined by $\{k, \bar{Y}, [0,1]\}$ has a solution in pure strategies.

Hence, the proof of Theorem 1 is complete by noting that the games $\{\tilde{J}, \Omega, [0,1]\}$ and $\{k, \bar{Y}, [0,1]\}$ are equivalent. Therefore, if (y^*, p^*) denotes any solution to $\{k, \bar{Y}, [0,1]\}$, (u_0^*, p^*) denotes a saddle-point solution for $\{\tilde{J}, \Omega, [0,1]\}$ when u_0^* is any control which achieves $y^* = \hat{J}(x^*(T))$.

Comment:

Assumption A_4 of this section in a sense replaces the assumption of the uniqueness of $x^*(T) = \arg \max_{\bar{X}(T, x_0)} k(x(T), p^*)$ which appears in Theorem 2 of sections 5 and 6.

8. OPTIMAL EVASION STRATEGIES AGAINST MULTIPLE MISSILES: PART II - FOR CRITERIA WITH A FREE TERMINAL TIME

8.1 Introduction:

We consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. We delineate the existence, structure, and behavior of optimal evasion strategies for this problem based on the following assumptions pertaining to dynamics, information patterns, and optimization criteria.

Dynamics:

The evader's and pursuer's dynamics are each nonlinear, i.e.,

$$\dot{x}_e = f_e(x_e, u_e, t)$$

$$\dot{x}_{pi} = f_{pi}(x_{pi}, u_i, t), \quad i = 1, 2.$$

Information:

Each pursuer ($i = 1, 2$) uses a given feedback control law for its guidance strategy. The evader's a priori information includes a complete description of each dynamical system including initial state information.

Optimization Criterion:

The evader seeks to maximize the "distance of closest approach" between himself and each pursuer over a given time interval $[0, T]$. Since there are two pursuers, the evader is faced with a multi-criterion or vector valued optimization problem. One approach to solving this multi-criterion problem is to seek an evasion strategy, which maximizes the minimum. This maxmin or game theoretic distance of closest approach between the evader and each pursuer over a given time interval $[0, T]$.

This approach requires that we solve a saddle-point problem to obtain the optimal evasion strategy.

In order to obtain a physically meaningful solution, we shall modify the original problem statement to include a set of constraints on the evader's permissible controls. In addition, we shall assume that for this given problem setting there exists a value of T (suitably large) such that there is no need for the evader to consider strategies for $t > T$. This implies that the distance of closest approach for each missile occurs at an interior point of $[0, T]$. Hence, we have chosen the free terminal time nomenclature to describe this situation. This latter assumption is easily justified in the setting where each missile is non-thrusting after $t = 0$, and drag forces are included in the problem formulation.

In the following section, it is assumed that the reader is familiar with the material in sections 1-7 of this report.

9. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS, HARD CONSTRAINTS ON CONTROL, AND A GENERAL COST FUNCTION:

9.1 Problem Statement:

Let $\dot{x} = f(x, u_0, t)$, $x(0) = x_0$, where $u_0 \in \Omega$; and

$\Omega \triangleq \{u_0 : u_0(t) \in \mathbb{R}^r; u_i \text{ Lebesgue measurable on } [0, T], a_i \leq u_i(t) \leq b_i, \\ i = 1, 2, \dots, r\}$, and $U = \prod_{i=1}^r [a_i, b_i]$.

Let $J_1(x(t))$ and $J_2(x(t))$ denote general cost functions. For example $J_1(x(t))$ could denote the actual distance between missile i and the aircraft at time t . However, the function J_1 could also be chosen to depend on the relative orientation of the missile and aircraft at time t .

Assumptions:

A_1 : J_1 and J_2 are continuous maps of E^n into $\mathbb{R}$.

A_2 : $f: E^n \times U \times [0, T] \rightarrow E^n$ is continuous in all arguments, is continuously differentiable in its first argument (x);

$x'f(x, u, t) \leq C[1 + \|x\|^2]$ for some C , all $x \in E^n$, $u \in U$ and $t \in [0, T]$.

A_3 : If $R(t, x) \triangleq \{f(x, u_0, t) : a_i \leq u_i \leq b_i, i = 1, \dots, r\}$, then the set $R(t, x)$ is convex for all pairs (t, x) where $0 \leq t \leq T$, $x \in E^n$.

A_4 : Let

$B \triangleq \{(b_1(x_0, u_0), b_2(x_0, u_0)) \in E^2 : x(0) = x_0, u_0 \in \Omega\}$,

where $b_i(x_0, u_0) \triangleq \min_{0 \leq t \leq T} J_i(x(t))$, then B is a convex set.

$$0 \leq t \leq T$$

Consider the following Maxmin Problem:

$$\max_{u_0 \in \Omega} \min_{i \in I} b_i(x_0, u_0).$$

Reasoning as before, we are lead to consider

$$\max_{u_0 \in \Omega} \min_{p \in [0,1]} \tilde{J}(u_0, p)$$

$$u_0 \in \Omega \quad p \in [0,1]$$

$$\text{where } \tilde{J}(u_0, p) = pb_1(x_0, u_0) + (1-p)b_2(x_0, u_0).$$

9.2 Problem Solution:

Consider the following saddle-point problem (SPP10):

Does

$$\max_{u_0 \in \Omega} \min_{p \in [0,1]} \tilde{J}(u_0, p) = \min_{p \in [0,1]} \max_{u_0 \in \Omega} \tilde{J}(u_0, p)?$$

$$u_0 \in \Omega \quad p \in [0,1] \quad p \in [0,1] \quad u_0 \in \Omega$$

Theorem 1:

Under Assumptions $A_1 - A_4$, SPP10 has a solution in pure strategies.

Proof: We begin with several observations.

Observation 1:

The set B is compact in E^2 .

Observation 2:

Let $k: B \times [0,1] \rightarrow \mathbb{R}$, $k(b, p) = pb_1 + (1-p)b_2$.

Since k is continuous in (b, p) , convex (affine) in p for fixed b , concave (affine) in b for fixed p , and B , $[0,1]$ are compact convex sets then by Sion's Theorem [Sion: 1958], the game defined by $\{k, B, [0,1]\}$ has a solution in pure strategies.

Hence, the proof of Theorem 1 is complete by noting that the games $\{J, \Omega, [0,1]\}$ and $\{k, B, [0,1]\}$ are equivalent. Therefore, if (b^*, p^*) denotes any solution to $\{k, B, [0,1]\}$, (u_0^*, p^*) denotes a saddle-point solution for $\{J, \Omega, [0,1]\}$ when u_0^* is any control which achieves $b^* = (b_1(x_0, u_0^*), b_2(x_0, u_0^*))$.

10. HISTORICAL PRECIS:

10.1 Games of Evasion with More Than One Pursuer:

As far as we are aware, Games of Evasion against several pursuers have not appeared extensively in the literature. Isaacs considers a very simple example of such a Differential Game on p. 148 of his celebrated monograph [1965].

10.2 Games of Evasion with a Single Pursuer:

The literature on Games of Evasion against a single pursuer is relatively extensive. We cite the following sources as examples: The monographs by Isaacs [1965]; Bryson and Ho [1969]; and Friedman [1971]. The papers by Behn and Ho [1968]; Rhodes and Luenberger [1969]; Basar and Mintz [1973]; and Poulter and Anderson [1976]. The thesis by Poulter [1975].

10.3 Optimization Problems with Vector Valued Pay-off:

The literature concerning optimization problems with vector valued pay-off functions is considerable. We cite the seminal paper by Da Cunha and Polak [1967] and the contribution by Reid and Citron [1971] as examples.

10.4 Linear-Quadratic Optimization Problems with Vector Valued Pay-off:

The problem of solving an LO optimization problem with vector valued pay-off has been considered by Medanic and Andjelic in a sequence of papers and letters [Medanic and Andjelic: 1971, 1972a, and 1972b]. (See also the critical appraisal of this work by Ho [1971].)

The results of section 2 of this present report parallels the open-loop results obtained by Medanic and Anjelic [1971, 1972b]. The basic difference (besides the methodology) is that we consider a max min problem with weighting matrices with both positive and negative eigenvalues, whereas Medanic and Andjelic

consider a min max problem with positive and nonnegative weighting matrices. This difference leads to the possible existence of a conjugate point in the present problem setting, which in turn provides us with a natural interpretation for our Condition 1 (section 2). We note that no conjugate point phenomenon can arise in [Medanic and Andjelic: 1971, 1972b].

6.0 Conclusions and Recommendations for Further Research

Four real-time heuristic algorithms for determining aircraft evasion strategies against a multiple missile threat have been described. All four heuristic algorithms are motivated by a formal game theoretic model for multiple missile evasion. Algorithms 1 and 2 are based on "myopic" saddle point calculations which apportion the projection of the instantaneous aircraft acceleration among the normals to the individual maneuver or guidance planes defined by each missile and its target. Algorithms 3 and 4 are also based on "myopic" saddle-point calculations. These latter two algorithms apportion the projection of the instantaneous aircraft acceleration into the individual maneuver planes so as to maximize the minimum of a particular function which is related to the line of sight rate of each missile threat. These latter two algorithms are motivated by the concept of anti-proportional navigation.

Each algorithm has the following properties: i) each requires relatively minimal dynamic and parametric information; ii) each provides capability against an N missile threat; iii) each generates aerodynamically feasible aircraft maneuvers which meet both structural and pilot stress limitations; iv) each is computable using foreseeable hardware; v) each exhibits markovian behavior, i.e., each is restartable from present state information.

Simulation results using each algorithm with generic F-4 and AIM-9 truth models characterized by nonlinear differential equations, including lift, drag, gravity, 3-dimensional point mass dynamics, aircraft load factor and roll rate limits, and missile autopilot dynamics and load factor limits have been presented.

Based on the performance of the four heuristic algorithms against seven

representative multiple missile engagement scenarios (Table 4-2a), where algorithm performance in a given scenario is gauged by the minimum miss distance associated with missiles 1 & 2, one observes that, on a scenario by scenario basis, there is no single algorithm whose performance dominates the remaining three algorithms. However, if we calculate the total number of multiple misses, we observe that the algorithms ranked in decreasing order of performance are: Algorithms 1, 4, 2, & 3.

We believe that the importance of this present study lies more in the development and comparative analysis of real-time algorithms for multiple missile evasion, than in the absolute numerical miss distance results obtained herein. The missile model employed in this study exhibits performance capabilities which make it more effective than an actual missile would be. Improving the missile model's realism will therefore affect the numerical miss distance results in the simulation.

Further research work is currently being pursued in several areas to obtain further insight into the potential for actual implementation of a "future generation" of one or several of these algorithms in an operational setting: i) timing of maneuver initiation in the EEG phase; ii) transition strategies between the EEG and IEG phases; iii) the incorporation of load factor as well as bank angle in the aircraft maneuver strategy; iv) a cross comparison between algorithms as a function of initial scenario geometry and state description; v) a parametric sensitivity analysis for items i - iv.

References

Monographs:

- (1969) Bryson, A.E. and Y.C. Ho, Applied Optimal Control, Blaisdell, Waltham, Massachusetts.
- (1971) Friedman, A., Differential Games, Wiley-Interscience, New York.
- (1977) Garnell, P. and D.J. East, Guided Weapon Control Systems, Pergamon Press, Oxford, England.
- (1969) Hermes, H. and J.P. LaSalle, Functional Analysis and Time Optimal Control, Academic Press, New York.
- (1965) Isaacs, R., Differential Games, Wiley, New York.
- (1969) Luenberger, D.G., Optimization By Vector Space Methods, Wiley, New York.
- (1969) Owen, G., Game Theory, Saunders, Philadelphia, Pennsylvania.

Journal Articles and Conference Papers:

- (1973) Basar, T. and M. Mintz, "A Multistage Pursuit-Evasion Game That Admits a Gaussian Random Process as a Maximin Control Policy," Stochastics, Vol. 1, pp. 25-69.
- (1968) Behn, R. and Y.C. Ho, "On a Class of Linear Stochastic Differential Games," IEEE Transactions on Automatic Control, Vol. AC-13, No. 3.
- (1967) Da Cunha, N.O. and E. Polak, "Constrained Minimization Under Vector Valued Criteria in Finite Dimensional Spaces," Journal of Mathematical Analysis and Applications, Vol. 19, pp. 103-124.
- (1962) Filippov, A.F., "On Certain Questions in the Theory of Optimal Control," SIAM Journal on Control, Vol. 1, No. 1, pp. 76-84.
- (1972) Ho, Y.C., "Comment on the Paper By J. Medanic and M. Andjelic," Journal of Optimization Theory and Applications, Vol. 10, No. 3, pp. 187-189.]
- (1977) Huling, S.F. and M. Mintz, "Optimal Evasion Strategies Against Multiple Missiles -- Based on Differential Game Theory," Proceedings of the 1977 IEEE Conference on Decision and Control, pp. 480-486.
- (1971) Medanic, J. and M. Andjelic, "On a Class of Differential Games Without Saddle-Point Solutions," Journal of Optimization Theory and Applications, Vol. 8, No. 6, pp. 413-430.]
- (1972a) Medanic, J. and M. Andjelic, "Minimax Solution of the Multiple Target Problem," IEEE Transactions on Automatic Control, Vol. AC-17, No. 5.
- (1972b) Medanic, J. and M. Andjelic, "Author's Reply to Y.C. Ho," Journal of Optimization Theory and Applications, Vol. 10, No. 3, pp. 190-191. 7

References (continued)

Journal Articles and Conference Papers:

- (1976) Poulter, R. and G. Anderson, "A Guidance Concept for Air-to-Air Missiles Based on Nonlinear Differential Game Theory," Proceedings of the 1976 NAECON Conference, pp. 605-609.
- (1971) Reid, R.W. and S.J. Citron, "On Noninferior Performance Index Vectors," Journal of Optimization Theory and Applications, Vol. 7, No. 1, pp. 11-28.
- (1969) Rhodes, I.B. and D.G. Luenberger, "Differential Games With Imperfect State Information," IEEE Transactions on Automatic Control, Vol. AC-14, No. 1.
- (1979a) Sheketoff, M.S. and M. Mintz, "Real-Time Computation of Aircraft Evasion Strategies Against a Multiple Missile Threat -- A Game theoretic Approach," Proceedings of the Tenth Annual Pittsburgh Conference on Modeling and Simulation (to appear).
- (1979b) Sheketoff, M.S. and M. Mintz, "On the Application of Game Theory to the Design of a Real-Time Algorithm for Multiple Missile Evasion," Proceedings of the 1979 International Conference on Information Sciences and Systems (to appear).
- (1958) Sion, M., "On General Minimax Theorems," Pacific Journal of Mathematics, Vol. 8, pp. 171-176.

Theses:

- (1979) Huling, S.F., On the Application of Game Theory and Time Series Analysis to Fire Control Systems, Ph.D. Dissertation to be submitted to the University of Pennsylvania.
- (1975) Poulter, R., Differential Game Guidance Versus Proportional Navigation for an Air-to-Air Missile, M.S. Thesis submitted to the United States Air Force Air University Air Force Institute of Technology, Report No. GA/MC/75-5.
- (1979) Sheketoff, M.S., Real-Time Algorithms for Multiple Missile Evasion: A Game Theoretic Approach, M.S. Thesis to be submitted to the University of Pennsylvania.

Reports:

- (1977) Brown, C.M. and D.H. Johnson, "Fire Control Simulation (FICS) User's Manual," The Analytical Sciences Corp., Report No. TR-750-1.
- (1975) Carpenter, G. and M. Falco, "Analysis of Aircraft Evasion Strategies in Air-to-Air Missile Effectiveness Models," Grumman Aerospace Corp., Report No. RE-506.

References (continued)

Reports:

- (1976) Carpenter, G. and M. Falco, "Supercruise Vulnerability to Surface-to-Air Missile Threats," Grumman Aerospace Corp.
- (1976) Shinar, J. and D. Steinberg, "Analysis of Optimal Evasive Maneuvers Based on a Linearized Two-Dimensional Kinematic Model," Technion - Israel Institute of Technology, T.A.E. Report No. 230.

Proposals:

- (1977) FAAC Proposal FP402U/2762, "Part I - Technical Proposal for a Missile Launch Envelope Sensitivity Analysis," First Ann Arbor Corp.
- (1977) Veda Proposal 13005-77U/Q0105, "Technical Proposal Volume I - Missile Launch Envelope Sensitivity Analysis," Veda Corp.

Technical Manuals:

- (1977) USAF Flight Manual, TO 1F-4C-1, (for USAF Series F-4C & F-4D aircraft).

Course Notes:

- (1975) Hohwiesner, W., "Principles of Airborne Fire Control," Department of Astronautics and Computer Science, United States Air Force Academy.
(These course notes are referred to in Chapter 3 of this present study by (AFA, 1975).)

Appendix A

FORTRAN IV Source Programs

Program ACDYN.91: complete listing; pp. A-2 through A-25.

Program ACDYN.92: main program and subroutine VALUE; pp. A-26 through A-34.

Program ACDYN.93: main program and subroutine VALUE; pp. A-35 through A-43.

Program ACDYN.94: main program and subroutine VALUE; pp. A-44 through A-52.

```

1      PROGRAM ACDYN
2 C
3 C      ACDYN.91 --- MYOPIC, MULTIPLE (TWO) MISSILES
4 C
5 C      USES LF=1.0, BA=0.0 FOR CONTROLS FOR FIRST STEP,
6 C          THEN MANEUVERS TO MAXIMIZE ACCELERATION NORMAL
7 C          TO THE "GUIDANCE PLANE" FOR SINGLE MISSILE CRITERION
8 C
9 C      FOR COMBINATION, USE DO LOOP OF LAMBDA'S, I.E.
10 C          LAMBDA = 0.0 THRU 1.0 BY 0.05, AT EACH STEP,
11 C          TO FIND THE MIN(LAMBDA) OF THE MAX(U0) OF :
12 C
13 C          |(LAMBDA*ACCEL(DOT)G1)+(1-LAMBDA)*ACCEL(DOT)G2)/
14 C
15 C          WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
16 C          TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
17 C
18 C      DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
19 C          FUNCTION OF LAMBDA; I.E.
20 C
21 C          TA1(LAMBDA) = DMIS1(FINAL)
22 C          TA2(LAMBDA) = DMIS2(FINAL)
23 C
24 C      ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
25 C
26 C      WRITES TO THE TERMINAL, THEN PRINTC
27 C
28 C      RK11 = RK12 = 4.5
29 C
30 C      STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
31 C
32 C      PLOTTED VARIABLES ARE ---
33 C
34 C          PL(,1) = USTAR (DEG)
35 C          PL(,2) = PERFSTAR
36 C          PL(,3) = USTAR2 (DEG)
37 C          PL(,4) = PERF2
38 C          PL(,5) = GX )
39 C          PL(,6) = GY ) COMBINED USING LAMBDA
40 C          PL(,7) = GZ )
41 C          PL(,8) = LF1 (COMMANDED)
42 C          PL(,9) = SPECIFIC ENERGY (A/C)
43 C          PL(,10) = SP. EN. (MISSILE 1)
44 C          PL(,11) = ALPHA (A/C)
45 C          PL(,12) = GAMMA (A/C)
46 C          PL(,13) = SIGMA (A/C)
47 C          PL(,14) = NORM1(G) = (GX**2+GY**2+GZ**2)**0.5
48 C          PL(,15) = Z (A/C)
49 C          PL(,16) = AIRSPEED (A/C)
50 C          PL(,17) = SY1 (L.O.S. PITCH)
51 C          PL(,18) = THETA1 (L.O.S. YAW)
52 C          PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
53 C          PL(,20) = DRAG (A/C)
54 C          PL(,21) = Z (MISSILE 1)
55 C          PL(,22) = X (A/C)
56 C          PL(,23) = X (MISSILE 1)
57 C          PL(,24) = Y (A/C)
58 C          PL(,25) = Y (MISSILE 1)
59 C          PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
60 C          PL(,27) = SYDT1

```

```

61 C      PL(,28) = THEDT1
62 C      PL(,29) = NORM1(LOS DOT) = (SYDT1**2+THEDT1**2)**.5
63 C      PL(,30) = GX1 )
64 C      PL(,31) = GY1 ) MISSILE 1
65 C      PL(,32) = GZ1 )
66 C      PL(,33) = NORM2
67 C      PL(,34) = GX2 )
68 C      PL(,35) = GY2 ) MISSILE 2
69 C      PL(,36) = GZ2 )
70 C      PL(,37) = NORM2(LOS-DOT)
71 C      PL(,38) = LF2C
72 C      PL(,39) = SP. EN. (MISSILE 2)
73 C      PL(,40) = SY2 (LOS PITCH)
74 C      PL(,41) = THETA2 (LOS YAW)
75 C      PL(,42) = X (MISSILE 2)
76 C      PL(,43) = Y (MISSILE 2)
77 C      PL(,44) = Z (MISSILE 2)
78 C      PL(,45) = SYDT2
79 C      PL(,46) = THEDT2
80 C      PL(,47) = LAMDA1 * 100.
81 C      PL(,48) = GSY1
82 C      PL(,49) = GTH1
83 C      PL(,50) = GSY2
84 C      PL(,51) = GTH2
85 C      PL(,52) = GSY )
86 C      PL(,53) = GTH ) COMBINED USING LAMBDA
87 C
88 C      REAL *4 STRING,SDATE
89 C      INTEGER CUTIME
90 C      LOGICAL PRTEO,LONCE
91 C      COMMON /PARM7/ LONCE
92 C      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTEO
93 C
94 C      IDUM=CUTIME(U)
95 C      CALL TIMEOD(STRING)
96 C      CALL DATE(SDATE)
97 C      WRITE(6,66) SDATE, STRING
98 C      66 FORMAT(15X,40(' '),ACDYN.91 ',40(' '),/,
99 C      1 51X,A8,2X,A8,/,1H1)
100 C
101 C      LONCE=.FALSE.
102 C      999 CALL INIT
103 C      CALL VALUE
104 C      CALL PLOUT(2)
105 C      CALL PLOUT(6)
106 C
107 C      WRITE(2,100)
108 C      100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2
109 C      1 'FOR NO INIT PRINT')
110 C      CALL BELL(1)
111 C      READ(1,110) ISEC
112 C      110 FORMAT(I1)
113 C      IF(ISEC.NE.0) GO TO 999
114 C      STOP
115 C      END

```

```

1      SUBROUTINE INIT
2      DIMENSION IONOF(2)
3      REAL MU, LF, LO, M1, LF1, L1, LMIN, LMAX, MAXLF, NORMAX,
4      1  LOSMAX, LF2, L2
5      INTEGER CUTIME
6      LOGICAL PRTED, LONCE
7      COMMON /AMP/ TO,QOSO,ALFAO,CLO,DC,LO,G,MU,C01,C02,C03,
8      1  RHO,BETA,CLAFU,SO,DU1,DU2,M1,CLAF1,S1,D11,RLMAX,
9      2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALREAS
10     3  ,P1,TAU,TAFTB,TFLAG,G2S2,ALFA2,CL2,D2,L2
11     COMMON /PARM1/ JJ,KSTEP
12     COMMON /PARM2/ VTHU,VTH1,DTH,TSTEP,ICPUTM
13     COMMON /PARM3/ NO,IO
14     COMMON /PARM5/ MANUVR
15     COMMON /PARM7/ LONCE
16     COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
17     COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
18     COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
19     COMMON /ARRAY1/ XO(16),XUIN(18)
20     COMMON /FLVR/ PL(10,53),PMAX,PMIN,LMAX,LMIN,SPMAX,
21     1  SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
22     2  DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
23 C
24     DATA IYES/'Y'/, IONOF/'OFF','ON '/
25 C
26     1  FORMAT(I1)
27     3  FORMAT(I3)
28     6  FORMAT(F12.6)
29     11 FORMAT(A1)
30 C
31     IF(LONCE) GO TO 800
32     PI=3.14159
33     NO=18
34     N1=6
35     IO=100
36     ISEC=1
37     RLMAX=600.0
38     ALREAS=25.0*PI/180.
39     MU=1243.C
40     BETA=0.000030575
41     G=52.78
42     RHO=0.0023769
43     TAFTB=13000.
44     C02=-0.7018
45     C03=14.141
46     CLAFU=3.8986
47     SO=530.0
48     DU1=0.01675
49     DU2=0.223
50     M1=5.20
51     CLAF1=22.918
52     S1=0.223
53     D11=0.7
54     D12=0.042
55     RK11=4.5
56     RK12=4.5
57     THRESH=0.1
58     VTHO=200.0
59     VTH1=200.0
60     DTH=15.0

```

```

61 C
62 C      BEGIN INPUTTING DATA ---
63 C      USE FILE 10
64 C
65      READ (10,3) KSTEP
66      TSTEP=1./FLOAT(KSTEP)
67      READ (10,6) XU(1)
68      READ (10,6) XU(2)
69      READ (10,6) XU(3)
70      READ (10,6) XU(4)
71      READ (10,6) DA1
72      XU(5)=DA1*PI/180.
73      READ (10,6) DA2
74      XU(6)=DA2*PI/180.
75      READ (10,6) TAU
76      READ (10,6) XU(7)
77      READ (10,6) XU(8)
78      READ (10,6) XU(9)
79      READ (10,6) XU(10)
80      READ (10,6) DA3
81      XU(11)=DA3*PI/180.
82      READ (10,6) DA4
83      XU(12)=DA4*PI/180.
84      READ (10,6) XU(13)
85      READ (10,6) XU(14)
86      READ (10,6) XU(15)
87      READ (10,6) XU(16)
88      READ (10,6) DA5
89      XU(17)=DA5*PI/180.
90      READ (10,6) DA6
91      XU(18)=DA6*PI/180.
92 C
93      DO 750 I=1,N0
94      XUIN(I)=XU(I)
95      750 CONTINUE
96      GO TO 815
97 C
98      800 CONTINUE
99      WRITE(6,813)
100      813 FORMAT(1H1)
101      815 CONTINUE
102      DO 850 I=1,N0
103      XU(I)=XUIN(I)
104      850 CONTINUE
105      IF(ISEC.EQ.2) GO TO 870
106 C
107      LONCE=.TRUE.
108      WRITE(2,852) TSTEP, TAU
109      WRITE(6,852) TSTEP, TAU
110      852 FORMAT(3X, 'TSTEP = ',F6.4, ', TAU = ',F8.4,/)
111      WRITE(2,854)
112      854 FORMAT(/, '      XU(.)      AIRCRAFT      MISSILE #1',
113      1 '      MISSILE #2',/)
114 C
115      DO 860 I=1,N1
116      J=I+6
117      K=I+12
118      TEMP=XU(I)
119      TEMP1=XU(J)
120      TEMP2=XU(K)

```

```

121      IF(I.LT.5) GO TO 855
122      TEMP=TEMP*180./PI
123      TEMP1=TEMP1*180./PI
124      TEMP2=TEMP2*180./PI
125      855 CONTINUE
126      WRITE(2,856) I, TEMP, TEMP1, TEMP2
127      WRITE(6,856) I, TEMP, J, TEMP1, K, TEMP2
128      856 FORMAT(16,3(5X,615.6))
129      858 FORMAT(3X,"INIT X0(",I1,") = ",615.6,
130      1      1X,"INIT X0(",I2,") = ",615.6,1X,
131      2      "INIT X0(",I2,") = ",615.6)
132      860 CONTINUE
133 C
134      870 CONTINUE
135      ICPUT=CUTIME(0)
136      LF1=.25*DA3
137      LF2=.25*DA5
138      U1=0.0
139      U2=0.0
140      UO=0.0
141      ACT1(1)=0.0
142      ACLF(1)=1.0
143      ACBA(1)=0.0
144      MANUVR=0
145      PRTEQ=.FALSE.
146      C01=22345.7
147 C
148      ZD1=X0(3)-X0(9)
149      YD1=X0(2)-X0(8)
150      XD1=X0(1)-X0(7)
151      RXY=SQRT(XD1*XD1+YD1*YD1)
152      PSY1=ATAN2(ZD1,RXY)
153      THETA1=ATAN2(YD1,XD1)
154      DMIS1=SQRT(RXY**2+ZD1**2)
155 C
156      ZD2=X0(3)-X0(15)
157      YD2=X0(2)-X0(14)
158      XD2=X0(1)-X0(13)
159      RXY=SQRT(XD2*XD2+YD2*YD2)
160      PSY2=ATAN2(ZD2,RXY)
161      THETA2=ATAN2(YD2,XD2)
162      DMIS2=SQRT(RXY**2+ZD2**2)
163 C
164      DO 880 I=1,53
165      DO 860 J=1,10
166      PL(J,I)=0.0
167      880 CONTINUE
168      PPMAX=160000.0
169      PMIN=0.0
170      LMAX=0.0
171      LMIN=6.0
172      SPMAX=0.0
173      SPMIN=X0(3)+0.5=X0(4)*X0(4)/6
174      ALFMAX=0.0
175      ALFMIN=0.0
176      NORMAX=0.0
177      ANMAX=0.0
178      ZMAX=0.0
179      ZMIN=AMIN1(X0(3),X0(9),X0(15))
180      XMAX=0.0

```

```

181      XMIN=AMIN1(XU(1),XG(7),XC(13))
182      YMAX=0.0
183      YMIN=AMIN1(XU(2),XG(8),XC(14))
184      VMAX=XU(4)
185      VMIN=VU(4)
186      DMAX=0.0
187      DMIN=1.0E10
188      LOSMAX=0.0
189 C
190      WRITE(2,903)
191 903 FORMAT(/," BEGIN ALL MANEUVERS AFTER 1 STEP")
192 C
193      WRITE(2,913) RK11,RK12
194      WRITE(6,913) RK11,RK12
195 913 FORMAT(2LX," PROPORTIONAL NAVIGATION GAINS: ",/,
196 1 12X," PITCH (RK11) = ",G12.3," , YAW (RK12) = ",G12.3,/)
197 C
198      WRITE(2,916)
199 916 FORMAT(" ENTER MAXIMUM LOAD FACTOR FOR A/C (F12.6)")
200      CALL BELL(1)
201      READ(1,6) MAXLF
202      IF(MAXLF.EQ.0.0) MAXLF=6.0
203      WRITE(2,918) MAXLF
204      WRITE(6,918) MAXLF
205 918 FORMAT(1CX," A/C MAXIMUM LOAD FACTOR = ",F5.2,/)
206 C
207      WRITE(2,926)
208 926 FORMAT(" USE AFTERBURNERS IN MANEUVER (Y OR N)?")
209      CALL BELL(1)
210      READ(1,11) IDUM
211      TFLAG=0.0
212      IF(IDUM.EQ.1YES) TFLAG=1.0
213      I=TFLAG+1.0
214      WRITE(2,928) IONOF(I)
215      WRITE(6,928) IONOF(I)
216 928 FORMAT(1CX," IN MANEUVER, AFTERBURNERS WILL BE ",A3,/)
217 C
218      ISEC=1
219      WRITE(2,971)
220 971 FORMAT(" ENTER 1 TO ABORT")
221      CALL BELL(1)
222      READ(1,1) LOGIC
223      IF(LOGIC.EQ.1) STOP
224 C
225      RETURN
226      END

```

```

1  SUBROUTINE VALUE
2  REAL MU, LF, LO, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
3  1  NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
4  2  NORMG
5  DIMENSION ACDF(100,2)
6  LOGICAL PRTEO, FLAG1, FLAG2
7  INTEGER CUTIME
8  COMMON /AMP/ TO,QOSU,ALFAO,CLO,DO,LO,G,MU,C01,C02,C03,
9  1  RHO,BETA,CLAFU,SU,DU1,DOZ,M1,CLAF1,S1,D11,RLMAX,
10  2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALKEAS
11  3  ,P1,TAU,TAFTB,TFLAG,Z2S2,ALFA2,CL2,D2,L2
12  COMMON /PARM1/ JJ,XSTEP
13  COMMON /PARM2/ VTHO,VTH1,DTH,TSTEP,ICPUTM
14  COMMON /PARM3/ NU,IU
15  COMMON /PARM4/ STEP
16  COMMON /PARM5/ MANUVR
17  COMMON /PARM6/ ISEC,DMIS1,DMIS2,PRTEO
18  COMMON /ARRAY1/ XO(18),XCIN(18)
19  COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
20  COMMON /FLVR/ PL(100,53),PMAX,PMIN,LMAX,LMIN,SPMAX,
21  1  SP*IN,ALFMAX,ALFMIN,NORMMAX,ZMAX,ZMIN,VMAX,VMIN,
22  2  DMAX,CMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
23  COMMON /PSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
24 C
25  EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
26  EQUIVALENCE (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
27 C
28  DATA TESTN /1.0/, IEXIT/'X'/, IQUIK/'Q'/
29 C
30  PERF(U)=COEFA*COS(U)+COEFB*SIN(U)+COEFC
31 C
32  COSGAM=COS(GAMA)
33  SINGAM=SIN(GAMA)
34  COSSIG=COS(SIGMA)
35  SINSIG=SIN(SIGMA)
36 C
37  TA=0.0
38  JJ=0
39  DUM1=1.0E10
40  DUM2=1.0E10
41  FLAG1=.FALSE.
42  FLAG2=.FALSE.
43  LAMDA1=0.5
44 C
45 C  FLAG(I) INDICATES THAT MISSILE I ALREADY GOT AWAY
46 C
47  100 CONTINUE
48  IF(FLAG1) LAMDA1=0.0
49  IF(FLAG2) LAMDA1=1.0
50  LAMDA2=1.0-LAMDA1
51  IVAL=50
52  TPRINT=TA
53  VOX=VO-COSGAM*COSSIG
54  VOY=VO-COSGAM*SINSIG
55  VOZ=VO*SINGAM
56  V1X=V1-COS(XO(11))*COS(XO(12))
57  V1Y=V1-COS(XO(11))*SIN(XO(12))
58  V1Z=V1*SIN(XO(11))
59  V2X=V2-COS(XO(17))*COS(XO(18))
60  V2Y=V2-COS(XO(17))*SIN(XO(18))

```

```

61      VZ2=VZ*SIN(XO(17))
62      VRELX1=VUX-V1X
63      VRELY1=VUY-V1Y
64      VRELZ1=VUZ-V1Z
65      VREL11=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
66      VRELX2=VUX-V2X
67      VRELY2=VUY-V2Y
68      VRELZ2=VUZ-V2Z
69      VREL12=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
70      DELX1=XU(1)-XO(7)
71      DELY1=XU(2)-XO(8)
72      DELZ1=XU(3)-XO(9)
73      DELX2=XU(1)-XO(13)
74      DELY2=XU(2)-XO(14)
75      DELZ2=XU(3)-XO(15)
76 C
77      DUM1=AMIN1(DUM1,DMIS1)
78      DUM2=AMIN1(DUM2,DMIS2)
79      DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
80      DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
81      DSV1=0.5*DMIS1/VREL1
82      DSV2=0.5*DMIS2/VREL2
83      STEP=AMIN1(7STEP,DSV1,DSV2)
84      DMIS=AMIN1(DMIS1,DMIS2)
85      IF(JJ.GT.0) MANUVR=1
86      LF=1.0
87      ULAST=UO
88      UO=0.0
89      IF(MANUVR.NE.1) GO TO 300
90 C
91      UO=ULAST
92      LF=MAXLF
93      ISEC=ISEC+1
94      GX1 = VRELY1*DELZ1 - VRELZ1*DELY1
95      GY1 = VRELZ1*DELX1 - VRELX1*DELZ1
96      GZ1 = VRELX1*DELY1 - VRELY1*DELX1
97      NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
98      GX2 = VRELY2*DELZ2 - VRELZ2*DELY2
99      GY2 = VRELZ2*DELX2 - VRELX2*DELZ2
100     GZ2 = VRELX2*DELY2 - VRELY2*DELX2
101     NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
102     IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
103 C
104 C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
105 C
106     LF=ACLF(ISEC-1)
107     GO TO 290
108 C
109     200 CONTINUE
110     GX1=GX1/NORM1
111     GY1=GY1/NORM1
112     GZ1=GZ1/NORM1
113     GX2=GX2/NORM2
114     GY2=GY2/NORM2
115     GZ2=GZ2/NORM2
116     PERFM=1.0E20
117 C
118     DO 270 LAMB=1,21
119     LAMDA1=FLOAT(LAMB-1)/20.
120     LAMDA2=1.0-LAMDA1

```

```

121      IF(FLAG1.AND.(LAMB.GT.1)) GO TO 280
122      IF(FLAG2.AND.(LAMB.LT.21)) GO TO 270
123 C
124      GX=LAMDA1*GX1+LAMDA2*GX2
125      GY=LAMDA1*GY1+LAMDA2*GY2
126      GZ=LAMDA1*GZ1+LAMDA2*GZ2
127 C
128      COEFA=GZ*COSGAM-GY*SINGAM*SINSIG-GX*SINGAM*COSSIG
129      COEFA=COEFA*(LU+TU*SIN(ALFAU))
130      COEFB=GY*COSGAM-GX*SINGAM
131      COEFB=COEFB*(LU+TU*SIN(ALFAU))
132      COEFC=((TU*COS(ALFAU)-UU)*(GZ*SINGAM+GY*COSGAM*SINSIG
133      1 +GX*COSGAM*COSSIG)) - GZ*MO*6
134 C
135      USTAR=ATAN2(COEFB,COEFA)
136      USTAR2=USTAR+PI
137      IF(USTAR.GT.U.U) USTAR2=USTAR-PI
138      220 CONTINUE
139      PERFST=PERF(USTAR)
140      PERF2=PERF(USTAR2)
141      ABST=ABS(PERFST)
142      ABS2=ABS(PERF2)
143 C
144 C      USE "BEST" U AND RATE LIMIT TO RLMAX
145 C
146      IFLAG=IVAL
147      UMAX=USTAR
148      PERMIN=ABST
149      IF(ABST.GE.ABS2) GO TO 230
150      UMAX=USTAR2
151      IFLAG=-IVAL
152      PERMIN=ABS2
153      230 CONTINUE
154      ROLL=(UMAX-UU)*180./PI
155      IF(ROLL.GT.180.) ROLL=ROLL-360.
156      IF(ROLL.LT.(-180.)) ROLL=ROLL+360.
157      AROLL=ABS(ROLL)
158 C
159 C      WE NOW RATE-LIMIT WHICHEVER ANGLE WE GET TO RLMAX
160 C      DEG PER SEC...
161 C
162      IF(AROLL.LE.(RLMAX*STEP)) GO TO 250
163      ROLL=ROLL*RLMAX*STEP/AROLL
164      IF(AROLL.LE.175.) GO TO 250
165 C
166 C      ROLL IS ESSENTIALLY A COMPLETE FLIP --- USE PLUS/MINUS
167 C      RLMAX INTO PERF FUNCTION TO CHECK FOR BES' WAY
168 C      TO MAKE FLIPS
169 C
170      IVAL=75
171      USTAR=UU+(RLMAX*STEP*PI/180.)
172      IF(USTAR.GT.PI) USTAR=USTAR-(2.0*PI)
173      USTAR2=UU-(RLMAX*STEP*PI/180.)
174      IF(USTAR2.LE.(-PI)) USTAR2=USTAR2+(2.0*PI)
175      GO TO 220
176 C
177      250 CONTINUE
178      IF(PERMIN.GE.PERF2) GO TO 270
179      PERFM=PERMIN
180      JMIN=LAMB

```

```

181      RLMIN=ROLL
182      PL(ISEC,1)=USTAR*180./PI
183      PL(ISEC,2)=PERFST
184      PL(ISEC,3)=USTAR2*180./PI
185      PL(ISEC,4)=PERF2
186      PL(ISEC,5)=GX
187      PL(ISEC,6)=GY
188      PL(ISEC,7)=GZ
189      PL(ISEC,26)=IFLAG
190      PL(ISEC,47)=5-(JMIN-1)
191      PMAX=AMAX1(PMAX,PERFST,PERF2)
192      PMIN=AMIN1(PMIN,PERFST,PERF2)
193      270 CONTINUE
194      280 CONTINUE
195 C
196      ROLL=RLMIN
197      UU=UU+(ROLL*PI/180.)
198      IF(UU.GT.PI) UU=UU-(2.0*PI)
199      IF(UU.LE.(-PI)) UU=UU+(2.0*PI)
200 C
201 C      CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
202 C
203      ACCX=((TU+COS(ALFAU)-DU)*(COSGAM+COSSIG))-
204      1      ((LU+TU*SIN(ALFAU))*(SIN(UU)+SINGAM+COS(UU))+
205      2      SINGAM*COSSIG))
206      ACCY=((TU+COS(ALFAU)-DU)*(COSGAM+SINSIG))+
207      1      ((LU+TU*SIN(ALFAU))*(SIN(UU)+COSGAM-COS(UU))+
208      2      SINGAM*SINSIG))
209      ACCZ=((TU+COS(ALFAU)-DU)*SINGAM)-(MO*G)+
210      1      ((LU+TU*SIN(ALFAU))*COSGAM+COS(UU))
211      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
212 C
213 C      STORE PLOTTED VARIABLES
214 C
215      PL(ISEC,14)=NORM1
216      PL(ISEC,19)=ANORM/MO
217      PL(ISEC,30)=GX1
218      PL(ISEC,31)=GY1
219      PL(ISEC,32)=GZ1
220      PL(ISEC,33)=NORM2
221      PL(ISEC,34)=GX2
222      PL(ISEC,35)=GY2
223      PL(ISEC,36)=GZ2
224      NORM1=AMAX1(NORM1,NORM1,NORM2)
225      ANMAX=AMAX1(ANMAX,ANORM)
226 C
227      290 CONTINUE
228      IF(PRTED) GO TO 300
229      PRTED=.TRUE.
230      CU1=CU1+TAFTB*TFLAG
231      TCHG=TPRINT
232      WRITE(2,310) TPRINT
233      WRITE(6,310) TPRINT
234      310 FORMAT(2X,"START MANEUVER AT T = ",F6.2)
235      CALL BELL(3)
236 C
237      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
238      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
239      ALFMAY=ALFAU
240      ALFMIN=ALFAU

```

```

241 ZMAX=AMIN1(XC(3),XC(9),XC(15))
242 ZPIN=AMAX1(XC(5),XC(9),XC(15))
243 AMAX=AMAX1(XC(1),XC(7),XC(13))
244 XMIN=AMIN1(XC(1),XC(7),XC(13))
245 YMAX=AMAX1(XC(2),XC(6),XC(14))
246 YPIN=AMIN1(XC(2),XC(6),XC(14))
247 VMAX=VU
248 VPIN=VU
249 DMAX=DU
250 DMIN=DU
251 LOSMAX=0.0
252 C
253 300 CONTINUE
254 ACBA(ISEC)=UU*180./PI
255 ACLF(ISEC)=LF
256 ACTI(ISEC)=TPRINT
257 ACDM(ISEC,1)=DMIS1
258 ACDM(ISEC,2)=DMIS2
259 C
260 IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
261 WRITE (2,400) TPRINT, DMIS1, DMIS2
262 WRITE (6,400) TPRINT, DMIS1, DMIS2
263 400 FORMAT (5X,"TIME = ",F10.3,2X,"DSEP1 = ",G12.3,2X,
264 1 "DSEP2 = ",G12.3)
265 C
266 477 CONTINUE
267 IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
268 IF(DMIS.LT.DTH) GO TO 520
269 IF(TA.LT.3.0) GO TO 480
270 FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
271 FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
272 IF(FLAG1.AND.FLAG2) GO TO 540
273 480 CONTINUE
274 CALL INTBOX
275 TA=TA+STEP
276 JJ=JJ+1
277 IF(ISEC.GE.10) GO TO 530
278 GO TO 100
279 C
280 510 CONTINUE
281 WRITE (2,515) TPRINT
282 WRITE (6,515) TPRINT
283 515 FORMAT(1X,"* A/C OR MISSILE VEL. IS TOO LOW ",
284 1 "AT TIME: ",F10.3," *",/)
285 GO TO 600
286 C
287 C HIT OCCURRED, PRINT OUT
288 520 CONTINUE
289 WRITE(2,525) TPRINT,DUM1,DUM2
290 WRITE(6,525) TPRINT,DUM1,DUM2
291 525 FORMAT(1X,"***** HIT AT TIME = ",F10.3,
292 1 " *****",/,5X,"BEST DSEPS WERE 1:",
293 2 " G15.6," & 2:" G15.0,/)
294 GO TO 600
295 C
296 530 CONTINUE
297 WRITE(2,535) TPRINT
298 WRITE(6,535) TPRINT
299 535 FORMAT(5X,"TIME LIMIT AT T = ",F6.2)
300 GO TO 600

```

```

301 C
302 C CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
303 540 CONTINUE
304 WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
305 WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
306 544 FORMAT(3X,"--- CLOSURE RATE NEGATIVE AT TIME = ",F10.3,
307 1 " ",F10.3,"/,"X,"TA1 : BEST DSEP = ",G15.6," , NOW = ",G15.6,"/,"
308 2 "X,"TA2 : BEST DSEP = ",G15.6," , NOW = ",G15.6,"/,"
309 GO TO 606
310 C
311 606 CONTINUE
312 CALL BELL(1)
313 C
314 READ(1,635) LOGIC
315 IF(LOGIC.EQ.IEXIT) RETURN
316 N10=0
317 DO 620 J=1,N10
318 JJJ=J+6
319 KK=J+12
320 TEMP=XU(J)
321 TEMP1=XU(JJJ)
322 TEMP2=XU(KK)
323 IF(J.LT.5) GO TO 605
324 TEMP=TEMP*180./PI
325 TEMP1=TEMP1*180./PI
326 TEMP2=TEMP2*180./PI
327 605 CONTINUE
328 WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
329 WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
330 610 FORMAT(2X,"XU(",I1,")= ",G13.4,4X,
331 1 "XU(",I2,")= ",G13.4,4X,
332 2 "XU(",I2,")= ",G13.4)
333 620 CONTINUE
334 C
335 WRITE(2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
336 WRITE(6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
337 625 FORMAT(1,10X,"DELX1: ",G12.3,2X,"DELY1: ",G12.3,"/,"
338 1 10X,"DELZ1: ",G12.3,2X,"DMIS1: ",G12.3,"/,"
339 2 10X,"BEST DMIS WAS ",G12.3)
340 WRITE(2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
341 WRITE(6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
342 630 FORMAT(1,10X,"DELX2: ",G12.3,2X,"DELY2: ",G12.3,"/,"
343 1 10X,"DELZ2: ",G12.3,2X,"DMIS2: ",G12.3,"/,"
344 2 10X,"BEST DMIS WAS ",G12.3)
345 CALL BELL(1)
346 READ(1,635) LOGIC
347 IF(LOGIC.EQ.IEXIT) RETURN
348 IF(LOGIC.EQ.IQUIK) GO TO 660
349 635 FORMAT(A1)
350 C
351 WRITE(6,631)
352 631 FORMAT(1H1)
353 DO 650 I=1,ISEC
354 IF(MOD(I,20).EQ.0) CALL BELL(1)
355 IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
356 IF(LOGIC.EQ.IEXIT) RETURN
357 IF(LOGIC.EQ.IQUIK) GO TO 660
358 WRITE(2,636) ACT1(1),ACBA(1),PL(1,6),PL(1,38)
359 636 FORMAT(1X,"TIME ",F5.2,3X,"ACBA ",F5.2,3X,"LF1C ",F8.1,
360 1 "X",LF2C ",F8.1)

```

```

361 650 CONTINUE
362 660 CONTINUE
363     DO 670 I=1,ISEC
364     WRITE(6,637) ACT1(I),ACLF(I),ACBA(I),PL(1,8),PL(1,38),
365     1 PL(1,14),PL(1,14),PL(1,11),PL(1,20),
366     2 (ACUM(I,J),J=1,2),FL(1,47)
367 637 FORMAT(1X,"TIME = ",0PF6.2,3X,"ACLF = ",F5.1,3X,
368 1 "ACBA(DEG) = ",F8.2,3X,"LFIC = ",F8.1,3X,
369 2 "LFEC = ",F8.2,3X,"NORM(A) = ",G11.2,/,20X,
370 3 "NORM(G) = ",G11.2,3A,"ALPHA = ",F6.2,3X,
371 4 "DPAG = ",G11.2,/,20X,"DMIS1 = TA(1) = ",
372 5 F9.2,3X,"DMIS2 = TA(2) = ",F9.2,3X,
373 6 "LAMBDA1 = ",-2PF5.2,/)
374 670 CONTINUE
375 C
376     IDUM=CUTIME(U)-ICPUTM
377     WRITE(2,639) IDUM
378     WRITE(6,639) IDUM
379 639 FORMAT(4X,"CPU TIME (SECONDS) = ",15,0PF5.0)
380 C
381     RETURN
382     END

```

```

1  SUBROUTINE INTBOX
2  REAL MU, LF, LU, M1, LF1, L1, LMAX, LMIN, MAXLF, NORMAX,
3  1  LOSMAX, LF2, L2, MVU, MV1, MV2
4  LOGICAL PRTEO
5  DIMENSION XPU(15)
6  COMMON /AMP/ TU, QUSU, ALFAU, CLU, DU, LO, G, MU, CO1, CO2, COS,
7  1  RHO, BETA, CLAFU, SU, DU1, DU2, M1, CLAF1, S1, D11, RLMAX,
8  2  D12, THRESH, RK11, RK12, Q1S1, ALFA1, CL1, D1, L1, ALKEAS
9  3  , P1, TAU, TAFTB, TFLAG, Q2S2, ALFA2, CL2, D2, L2
10 COMMON /ARRAY1/ XU(16), XUIN(18)
11 COMMON /PARAM1/ JJ, KSTEP
12 COMMON /PARAM3/ NU, IU
13 COMMON /PARAM4/ STEP
14 COMMON /PARAM5/ MANUVR
15 COMMON /PARAM9/ ISEC, DMIS1, DMIS2, PRTEO
16 COMMON /CONTR/ ACLF(100), ACBA(100), ACTI(100), TCHG, LF, UO, MAXLF
17 COMMON /FLVR/ PL(100,55), PHAX, PMIN, LMAX, LMIN, SPMAX,
18 1  SPMIN, ALFMAX, ALFMIN, NORMAX, ZMAX, ZMIN, VMAX, VMIN,
19 2  DMAX, DMIN, XMAX, XMIN, YMAX, YMIN, ANMAX, LOSMAX
20 COMMON /PSL/ LF1, U1, PSY1, THETA1, LF2, U2, PSY2, THETA2
21 C
22 EQUIVALENCE (GAMA, XU(5)), (SIGMA, XU(6))
23 EQUIVALENCE (VU, XU(4)), (V1, XU(10)), (V2, XU(16))
24 C
25 C FIRST CALCULATE AIRCRAFT DYNAMICS
26 C
27 COSGA=CCS(GAMA)
28 SINGA=SIN(GAMA)
29 COSSIG=CCS(SIGMA)
30 SINSIG=SIN(SIGMA)
31 C
32 MVU=MU-VG
33 TU=CU1+CU2*XU(3)+CU3*VO
34 QUSO=0.5*RHO*EXP(-BETA*XU(3))+VO*VO*SD
35 C
36 C LIMIT ALFAU TO STALL ANGLE, AND STORE ALPHA ACHIEVED
37 C
38 ALFAU=AMIN1((MU*G+LF/(CLAFU*QUSO)), ALKEAS)
39 ACLF(ISEC)=ALFAU-CLAFU*QUSO/(MU*G)
40 SINL F=SIN(ALFAU)*TU
41 CLU=CLAFU*ALFAU
42 DU=QUSU*(DU1+DU2-CLU**2)
43 LU=CLU*QUSU
44 XPU(1)=VU*COSGA-COSSIG
45 XPU(2)=VU*COSGA*SINSIG
46 XPU(3)=VU*SINGA
47 XPU(4)=(TU*CCS(ALFAU)-DU)/MU-G*SINGA
48 XPU(5)=(LU+SINL F)*COS(UU)/(MVU)-G*COSGA/MU
49 XPU(6)=(LU+SINL F)*SIN(UU)/(MVU+COSGA)
50 C
51 C MISSILE # 1 --- CALCULATE DYNAMICS AND CONTROLS
52 C
53 COSGA=CCS(XU(11))
54 SINGA=SIN(XU(11))
55 COSSIG=CCS(XU(12))
56 SINSIG=SIN(XU(12))
57 C
58 MV1=M1*V1
59 Q1S1=0.5*RHO*EXP(-BETA*XU(9))+V1*V1*S1
60 ALFA1C=M1*G*LF1/(CLAF1*Q1S1)

```

```

61 IF(JJ.EQ.U) ALFA1=ALFA1C
62 IF(TAU.GE.U.CC001)ALFA1=EXP(-STEP/TAU)*(ALFA1-ALFA1C) + ALFA1C
63 IF(TAU.LT.U.CC001) ALFA1=ALFA1C
64 CL1=CLAF1*ALFA1
65 Q1=Q1S1*(D11+D12-CL1**2)
66 L1=CL1*Q1S1
67 XPU(7)=V1*COSGAM*COSSIG
68 XPU(8)=V1*COSGAM*SINSIG
69 XPU(9)=V1*SINGAM
70 XPU(10)=-D1/M1 - G*SINGAM
71 XPU(11)=L1*COS(U1)/(MV1) - G*COSGAM/V1
72 XPU(12)=L1*SIN(U1)/(MV1*COSGAM)
73 XD1=XU(1)-XU(7)
74 YD1=XU(2)-XU(8)
75 ZD1=XU(3)-XU(9)
76 XPD1=XPU(1)-XPU(7)
77 YPD1=XPU(2)-XPU(8)
78 ZPD1=XPU(3)-XPU(9)
79 RR1=AD1**2+YD1**2+ZD1**2
80 RXY1=AD1**2+YD1**2
81 RXY1=SQRT(RXY1)
82 PSY1=ATAN2(ZD1,RXY1)
83 THETA1=ATAN2(YD1,XD1)
84 THEDT1=(XD1*YPD1-YD1*XPDI)/RXY1
85 SYDT1=(RXY1*ZPD1-ZD1*(XD1*XPDI+YD1*YPD1)/RXY1)/RR1
86 DTHO1*=(SYDT1**2+THEDT1**2)**0.5
87 ARG1=RK12*THEDT1/(RK11*SYDT1+G*COSGAM/V1)
88 U1=ATAN(ARG1)
89 TEST=ABS(THEDT1)
90 IF(TEST.GE.THRESH) LF1=
91 1 RK12*THEDT1-V1*COSGAM/(G*SIN(U1))
92 IF(TEST.LT.THRESH) LF1=
93 1 (RK11*SYDT1+G*COSGAM/V1)*(V1/(G*COS(U1)))
94 C
95 PL(1SEC,C)=LF1
96 LMAX=AMAX1(LMAX,ABS(LF1))
97 LMIN=AMIN1(LMIN,(LF1))
98 LF1=AMIN1(15.0,AMAX1(-15.0,LF1))
99 C
100 C MISSILE # 2 --- CALCULATE DYNAMICS AND CONTROLS
101 C
102 COSGA*=COS(XU(17))
103 SINGA*=SIN(XU(17))
104 COSSIG=COS(XU(18))
105 SINSIG=SIN(XU(18))
106 C
107 MV2=M1*V2
108 W2S2=0.5*RHO*EXP(-BETA*XL(15))-V2*V2*S1
109 ALFA2C=M1*G*LF2/(CLAF1*Q2S2)
110 IF(JJ.EQ.U) ALFA2=ALFA2C
111 IF(TAU.GE.U.CC001)ALFA2=EXP(-STEP/TAU)*(ALFA2-ALFA2C) + ALFA2C
112 IF(TAU.LT.U.CC001) ALFA2=ALFA2C
113 CL2=CLAF1*ALFA2
114 Q2=Q2S2*(D11+D12-CL2**2)
115 L2=CL2*Q2S2
116 XPU(14)=V2*COSGAM*COSSIG
117 XPU(15)=V2*COSGAM*SINSIG
118 XPU(16)=V2*SINGAM
119 XPU(17)=-D2/M1 - G*SINGAM
120 XPU(18)=L2*COS(U2)/(MV2) - G*COSGAM/V2

```

```

121 XPO(14)=L2*SIN(U2)/(MV2*COSGAM)
122 XD2=XO(1)-XO(13)
123 YD2=XO(2)-XO(14)
124 ZD2=XO(3)-XO(15)
125 XPD2=XPO(1)-XPO(13)
126 YPD2=XPO(2)-XPO(14)
127 ZPD2=XPO(3)-XPO(15)
128 RR2=AD2**2+YD2**2+ZD2**2
129 RXY2=AD2**2+YD2**2
130 RXY2=SQRT(RXY2)
131 PSY2=ATAN2(ZD2,RXY2)
132 THETA2=ATAN2(YD2,XD2)
133 THEDT2=(XD2*YPD2-YD2*XPD2)/RXY2
134 SYDT2=(RXY2*ZPD2-ZD2*(XD2*XPD2+YD2*YPD2)/RXY2)/RR2
135 DTN2=(SYDT2**2+THEDT2**2)**0.5
136 ARG2=RK12*THEDT2/(RK11*SYDT2+G*COSGAM/V2)
137 U2=ATAN(ARG2)
138 TEST=ABS(THEDT2)
139 IF(TEST.GE.THRESH) LF2=
140 1 RK12*THEDT2*V2*COSGAM/(G*SIN(U2))
141 IF(TEST.LT.THRESH) LF2=
142 1 (RK11*SYDT2+G*COSGAM/V2)*(V2/(G+COS(U2)))
143 C
144 PL(ISEC,38)=LF2
145 LMAX=AMAX1(LMAX,ABS(LF2))
146 LMIN=AMIN1(LMIN,(LF2))
147 LF2=AMIN1(15.0,AMAX1(-15.0,LF2))
148 C
149 DO INTEGRATION
150 C
151 DO 100 I=1,NU
152 XO(I)=XO(1) + STEP*XPO(I)
153 100 CONTINUE
154 C
155 CALCULATE GUIDANCE PLANES IN SY-THETA COORDINATES
156 C
157 GSY1=-THEDT1/COS(PSY1)
158 GTH1=SYDT1
159 GSTN=SQRT(GSY1**2+GTH1**2)
160 IF(GSTN.GE.(1.0)) GSY1=GSY1/GSTN
161 IF(GSTN.GE.(1.0)) GTH1=GTH1/GSTN
162 C
163 GSY2=-THEDT2/COS(PSY2)
164 GTH2=SYDT2
165 GSTN=SQRT(GSY2**2+GTH2**2)
166 IF(GSTN.GE.(1.0)) GSY2=GSY2/GSTN
167 IF(GSTN.GE.(1.0)) GTH2=GTH2/GSTN
168 C
169 ALAM=PL(ISEC,47)/100.
170 GSY=ALAM*GSY1+(1-ALAM)*GSY2
171 GTH=ALAM*GTH1+(1-ALAM)*GTH2
172 C
173 STORE PLOTTED VARIABLES
174 C
175 SPEN0=XO(3) + 0.5*V0*V0/G
176 SPEN1=XO(9) + 0.5*V1*V1/G
177 SPEN2=XO(15) + 0.5*V2*V2/G
178 PL(ISEC,9)=SPEN0
179 PL(ISEC,10)=SPEN1
180 PL(ISEC,11)=ALFAU*180./PI

```

```

181 PL(ISEC,12)=GAMA*180./PI
182 PL(ISEC,13)=SIGMA*180./PI
183 PL(ISEC,15)=XU(3)
184 PL(ISEC,16)=VU
185 PL(ISEC,17)=PSY1*180./PI
186 PL(ISEC,18)=THETA1*180./PI
187 PL(ISEC,20)=DU
188 PL(ISEC,21)=XU(9)
189 PL(ISEC,22)=XU(1)
190 PL(ISEC,23)=XU(7)
191 PL(ISEC,24)=XU(2)
192 PL(ISEC,25)=XU(8)
193 PL(ISEC,27)=ASPEC(SYDT1)
194 PL(ISEC,28)=ASPEC(THEDT1)
195 PL(ISEC,29)=ASPEC(DTNORM)
196 PL(ISEC,37)=ASPEC(DTN2)
197 PL(ISEC,39)=SPEN2
198 PL(ISEC,40)=PSY2*180./PI
199 PL(ISEC,41)=THETA2*180./PI
200 PL(ISEC,42)=XU(11)
201 PL(ISEC,43)=XU(14)
202 PL(ISEC,44)=XU(15)
203 PL(ISEC,45)=ASPEC(SYDT2)
204 PL(ISEC,46)=ASPEC(THEDT2)
205 PL(ISEC,48)=GSY1
206 PL(ISEC,49)=GTH1
207 PL(ISEC,50)=GSY2
208 PL(ISEC,51)=GTH2
209 PL(ISEC,52)=GSY
210 PL(ISEC,53)=GTH
211 C
212 SPMAX=AMAX1(SPMAX,SPENG,SPEN1,SPEN2)
213 SPMIN=AMIN1(SPMIN,SPENG,SPEN1,SPEN2)
214 ALFMAX=AMAX1(ALFAU*180./PI,ALFMAX)
215 ALFMIN=AMIN1(ALFAU*180./PI,ALFMIN)
216 ZMAX=AMAX1(ZMAX,XU(3),XU(9),XU(15))
217 ZMIN=AMIN1(ZMIN,XU(3),XU(9),XU(15))
218 XMAX=AMAX1(XMAX,XU(1),XU(7),XU(13))
219 XMIN=AMIN1(XMIN,XU(1),XU(7),XU(13))
220 YMAX=AMAX1(YMAX,XU(2),XU(8),XU(14))
221 YMIN=AMIN1(YMIN,XU(2),XU(8),XU(14))
222 VMAX=AMAX1(VMAX,VU)
223 VMIN=AMIN1(VMIN,VU)
224 DMAX=AMAX1(DMAX,DU)
225 DMIN=AMIN1(DMIN,DU)
226 LOSMAX=AMAX1(LOSMAX,ASPEC(DTNORM),ASPEC(DTN2))
227 C
228 RETURN
229 END

```

```

1  SUBROUTINE PLOUT(10)
2  DIMENSION XT(99), YT(98), ARAY1(99), ARAY2(98)
3  REAL LMAX, LMIN, MAXLF, NORMAX, LOSMAX, LOSMIN, LF
4  INTEGER CUTIME
5  LOGICAL PRTEO
6  COMMON /FLVR/ PL(100,55),PMAX,PMIN,LMAX,LMIN,SPMAX,
7  1  SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
8  2  DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,AMAX,LOSMAX
9  COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
10 COMMON /PARM2/ VTHO,VTH1,DTH,TSTEP,ICPUTM
11 COMMON /PARM4/ ISEC,DMIS1,DMIS2,PRTEO
12 C
13 EQUIVALENCE (XT(1),ACTI(2)),(YT(1),ACTI(3))
14 C
15 DATA IEXIT/'X'/
16 C
17 JSEC=ISEC-1
18 KSEC=ISEC-2
19 IF(KSEC.LE.1) RETURN
20 C
21 IF(10.EQ.6) WRITE(2,11)
22 11 FORMAT(' TYPE "X" FOR NO PRINTED GRAPHS')
23 C
24 CALL BELL(1)
25 READ(1,1) LOGIC
26 IF(LOGIC.EQ.IEXIT) RETURN
27 1 FORMAT(A1)
28 C
29 PMAX=AMAX1(PMAX,ABS(PMIN))
30 PMIN=-PMAX
31 APMIN=PMAX
32 C
33 CALL XSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
34 IF(10.EQ.2) CALL VDT
35 CALL BOX
36 CALL PLARRY('F',XT,PL(2,26),JSEC)
37 CALL PLARRY('-',XT,PL(2,1),JSEC)
38 CALL PLARRY('Z',XT,PL(2,5),JSEC)
39 CALL PLARRY('L',XT,PL(2,47),JSEC)
40 CALL PLARRY('U',ACTI,ACBA,ISEC)
41 CALL GRAPH('MANEUVERING TIME',16,'BANK ANGLES & LAMBDA'
42 1  ',20)
43 IF(10.EQ.2) READ(1,1) LOGIC
44 IF(LOGIC.EQ.IEXIT) RETURN
45 C
46 IF(10.EQ.5) WRITE(2,12)
47 12 FORMAT(' PLEASE BE PATIENT --- I'M GOING AS FAST AS I CAN')
48 C
49 CALL XSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
50 IF(10.EQ.2) CALL VDT
51 CALL BOX
52 CALL PLARRY('U',ACTI,ACBA,ISEC)
53 CALL GRAPH('MANEUVERING TIME',16,'BANK ANGLES',11)
54 IF(10.EQ.2) READ(1,1) LOGIC
55 IF(LOGIC.EQ.IEXIT) RETURN
56 C
57 CALL XSPRED(ACTI(1),ACTI(ISEC),'LIN',100.,0.,'LIN')
58 IF(10.EQ.2) CALL VDT
59 CALL BOX
60 CALL PLARRY('L',XT,PL(2,47),JSEC)

```

```

61 CALL GRAPH("MANEUVERING TIME",16,"LAMBDA",6)
62 IF(10.EQ.2) READ(1,1) LOGIC
63 IF(LOGIC.EQ.1EXIT) RETURN
64 C
65 DO 60 I=3,ISEC
66 APAY1(I)=ABS(PL(I,2))
67 ARAY2(I)=ABS(PL(I,4))
68 APMIN=AMIN1(APMIN,ARAY1(I),ARAY2(I))
69 60 CONTINUE
70 C
71 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",APMIN,PMAX,"LIN")
72 IF(10.EQ.2) CALL VDT
73 CALL BOX
74 CALL PLARRY("A",YT,ARAY1,KSEC)
75 CALL PLARRY("Z",YT,ARAY2,KSEC)
76 CALL GRAPH("MANEUVERING TIME",16,"ABS PERFS",9)
77 IF(10.EQ.2) READ(1,1) LOGIC
78 IF(LOGIC.EQ.1EXIT) RETURN
79 C
80 DO 80 I=2,ISEC
81 PL(I,1)=PL(I,1)*1.0E3
82 PL(I,3)=PL(I,3)*1.0E3
83 ACBA(I)=ACBA(I)*1.0E3
84 80 CONTINUE
85 C
86 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",PMIN,PMAX,"LIN")
87 IF(10.EQ.2) CALL VDT
88 CALL BOX
89 CALL PLARRY("P",XT,PL(2,2),JSEC)
90 CALL PLARRY("P",XT,PL(2,4),JSEC)
91 CALL PLARRY("Y",XT,PL(2,1),JSEC)
92 CALL PLARRY("Z",XT,PL(2,3),JSEC)
93 CALL PLARRY("U",ACTI,ACBA,ISEC)
94 CALL GRAPH("U=BANK USED; Y,Z=BANK CALC" "D;P,R=PERFS",37,
95 1 "BANKS * 1000 AND PERFS",22)
96 IF(10.EQ.2) READ(1,1) LOGIC
97 IF(LOGIC.EQ.1EXIT) RETURN
98 C
99 DO 90 I=2,ISEC
100 PL(I,1)=PL(I,1)*1.0E-3
101 PL(I,3)=PL(I,3)*1.0E-3
102 ACBA(I)=ACBA(I)*1.0E-3
103 90 CONTINUE
104 C
105 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",VMIN,VMAX,"LIN")
106 IF(10.EQ.2) CALL VDT
107 CALL BOX
108 CALL PLARRY("V",ACTI,PL(1,16),ISEC)
109 CALL GRAPH("MANEUVERING TIME",16,"AIRSPEED",6)
110 IF(10.EQ.2) READ(1,1) LOGIC
111 IF(LOGIC.EQ.1EXIT) RETURN
112 C
113 CALL SPRED(XMIN,XYMAX,"LIN",YMIN,YMAX,"LIN")
114 IF(10.EQ.2) CALL VDT
115 CALL BOX
116 CALL PLARRY("1",PL(1,25),PL(1,25),JSEC)
117 CALL PLARRY("2",PL(1,42),PL(1,43),JSEC)
118 CALL PLARRY("A",PL(1,22),PL(1,24),JSEC)
119 CALL GRAPH("X-Y PROJECTION --- X",20,"Y",1)
120 IF(10.EQ.2) READ(1,1) LOGIC

```

```

121      IF(LOGIC.EQ.IEXIT) RETURN
122 C
123      CALL SPRED(XMIN,XMAX,"LIN",ZMIN,ZMAX,"LIN")
124      IF(IG.EQ.2) CALL VDT
125      CALL BOX
126      CALL PLARRY("1",PL(1,23),PL(1,21),JSEC)
127      CALL PLARRY("2",PL(1,42),PL(1,44),JSEC)
128      CALL PLARRY("A",PL(1,22),PL(1,15),JSEC)
129      CALL GRAPH("X-Z PROJECTION --- X",20,"Z",1)
130      IF(IG.EQ.2) READ(1,1) LOGIC
131      IF(LOGIC.EQ.IEXIT) RETURN
132 C
133      CALL SPRED(YMIN,YMAX,"LIN",ZMIN,ZMAX,"LIN")
134      IF(IG.EQ.2) CALL VDT
135      CALL BOX
136      CALL PLARRY("1",PL(1,23),PL(1,21),JSEC)
137      CALL PLARRY("2",PL(1,42),PL(1,44),JSEC)
138      CALL PLARRY("A",PL(1,24),PL(1,15),JSEC)
139      CALL GRAPH("Y-Z PROJECTION --- Y",20,"Z",1)
140      IF(IG.EQ.2) READ(1,1) LOGIC
141      IF(LOGIC.EQ.IEXIT) RETURN
142 C
143      IF(IG.EQ.6) WRITE(2,13)
144      13 FORMAT(' I JUST FINISHED PROJECTIONS -- HANG IN THERE')
145 C
146      LOSMIN=-LOSHAX
147      CALL SPRED(ACTI(1),ACTI(JSEC),"LIN",LOSMIN,LOSHAX,"LIN")
148      IF(IG.EQ.2) CALL VDT
149      CALL BOX
150      CALL PLARRY("N",ACTI,PL(1,29),JSEC)
151      CALL PLARRY("S",ACTI,PL(1,27),JSEC)
152      CALL PLARRY("T",ACTI,PL(1,28),JSEC)
153      CALL GRAPH("MANEUVERING TIME",16,"LOS RATES AND NORM1",19)
154      IF(IG.EQ.2) READ(1,1) LOGIC
155      IF(LOGIC.EQ.IEXIT) RETURN
156 C
157      CALL SPRED(ACTI(1),ACTI(JSEC),"LIN",LOSMIN,LOSHAX,"LIN")
158      IF(IG.EQ.2) CALL VDT
159      CALL BOX
160      CALL PLARRY("N",ACTI,PL(1,37),JSEC)
161      CALL PLARRY("S",ACTI,PL(1,45),JSEC)
162      CALL PLARRY("T",ACTI,PL(1,46),JSEC)
163      CALL GRAPH("MANEUVERING TIME",16,"LOS RATES AND NORM2",19)
164      IF(IG.EQ.2) READ(1,1) LOGIC
165      IF(LOGIC.EQ.IEXIT) RETURN
166 C
167      CALL YSPRED(ACTI(1),ACTI(JSEC),"LIN",-180.,180.,"LIN")
168      IF(IG.EQ.2) CALL VDT
169      CALL BOX
170      CALL PLARRY("S",ACTI,PL(1,17),JSEC)
171      CALL PLARRY("P",ACTI,PL(1,40),JSEC)
172      CALL PLARRY("T",ACTI,PL(1,18),JSEC)
173      CALL PLARRY("Z",ACTI,PL(1,41),JSEC)
174      CALL GRAPH("MANEUVERING TIME",16,"NAV ANGLES",10)
175      IF(IG.EQ.2) READ(1,1) LOGIC
176      IF(LOGIC.EQ.IEXIT) RETURN
177 C
178      CALL YSPRED(ACTI(1),ACTI(JSEC),"LIN",-180.,180.,"LIN")
179      IF(IG.EQ.2) CALL VDT
180      CALL BOX

```

```

181 CALL PLARRY("S",ACTI,PL(1,13),JSEC)
182 CALL PLARRY("C",ACTI,PL(1,12),JSEC)
183 CALL GRAPH("MANEUVERING TIME",16,"POSITN ANGLES",13)
184 IF(10.EQ.2) READ(1,1) LOGIC
185 IF(LOGIC.EQ.1EXIT) RETURN
186 C
187 IF(10.EQ.6) WRITE(2,14)
188 14 FORMAT(" NEXT ARE THE GUIDANCE PLANES -- NOT MUCH MORE")
189 C
190 CALL XSPRED(XT(1),XT(JSEC),"LIN",-1.0,1.0,"LIN")
191 IF(10.EQ.2) CALL VDT
192 CALL BOX
193 CALL PLARRY("S",XT,PL(2,52),JSEC)
194 CALL PLARRY("T",XT,PL(2,53),JSEC)
195 CALL GRAPH("MANEUVERING TIME",16,"GUIDANCE PLANE",14)
196 IF(10.EQ.2) READ(1,1) LOGIC
197 IF(LOGIC.EQ.1EXIT) RETURN
198 C
199 CALL XSPRED(XT(1),XT(JSEC),"LIN",-1.0,1.0,"LIN")
200 IF(10.EQ.2) CALL VDT
201 CALL BOX
202 CALL PLARRY("S",XT,PL(2,48),JSEC)
203 CALL PLARRY("T",XT,PL(2,49),JSEC)
204 CALL GRAPH("MANEUVERING TIME",16,"GUIDANCE PLANE-1",16)
205 IF(10.EQ.2) READ(1,1) LOGIC
206 IF(LOGIC.EQ.1EXIT) RETURN
207 C
208 CALL XSPRED(XT(1),XT(JSEC),"LIN",-1.0,1.0,"LIN")
209 IF(10.EQ.2) CALL VDT
210 CALL BOX
211 CALL PLARRY("S",XT,PL(2,50),JSEC)
212 CALL PLARRY("T",XT,PL(2,51),JSEC)
213 CALL GRAPH("MANEUVERING TIME",16,"GUIDANCE PLANE-2",16)
214 C
215 IF(10.EQ.2) READ(1,1) LOGIC
216 IF(LOGIC.EQ.1EXIT) RETURN
217 C
218 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",LMIN,LMAX,"LIN")
219 IF(10.EQ.2) CALL VDT
220 CALL BOX
221 CALL PLARRY("1",ACTI,PL(1,6),JSEC)
222 CALL PLARRY("2",ACTI,PL(1,38),JSEC)
223 CALL PLARRY("A",ACTI,ALF,ISEC)
224 CALL GRAPH("MANEUVERING TIME",16,"LOAD FACTORS",12)
225 IF(10.EQ.2) READ(1,1) LOGIC
226 IF(LOGIC.EQ.1EXIT) RETURN
227 C
228 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",ALFMIN,ALFMAX,"LIN")
229 IF(10.EQ.2) CALL VDT
230 CALL BOX
231 CALL PLARRY("A",ACTI,PL(1,11),ISEC)
232 CALL GRAPH("MANEUVERING TIME",16,"ANGLE OF ATTACK",15)
233 IF(10.EQ.2) READ(1,1) LOGIC
234 IF(LOGIC.EQ.1EXIT) RETURN
235 C
236 CALL SPRED(ACTI(1),ACTI(ISEC),"LIN",DMIN,DMAX,"LIN")
237 IF(10.EQ.2) CALL VDT
238 CALL BOX
239 CALL PLARRY("D",ACTI,PL(1,20),JSEC)
240 CALL GRAPH("MANEUVERING TIME",16,"DRAG",4)

```

```

241      IF(IG.EQ.2) READ(1,1) LOGIC
242      IF(LOGIC.EQ.IEXIT) RETURN
243 C
244      CALL SPRED(ACTI(1),ACTI(1,SEC),"LIN",SPMIN,SPMAX,"LIN")
245      IF(IG.EQ.2) CALL VDT
246      CALL RUX
247      CALL PLARRY("1",ACTI,PL(1,10),ISEC)
248      CALL PLARRY("2",ACTI,PL(1,39),ISEC)
249      CALL PLARRY("A",ACTI,PL(1,9),ISEC)
250      CALL GRAPH("MANEUVERING TIME",16,"SPECIFIC ENERGIES",17)
251      IF(IG.EQ.6) WRITE(2,15)
252      15 FORMAT("      DONE !!!!!")
253 C
254      IDUM=CUTIME(U)-ICPUTM
255      WRITE(10,239) IDUM
256      239 FORMAT(5X,"CFU TIME (SECONDS) = ",15)
257 C
258      RETURN
259      END

```

```
1  SUBROUTINE BELL(N)
2  INTEGER*2 A(10)
3  DATA A/10*ZE000/
4  100  FORMAT(10(5X,A1))
5  N=MIN0(N,10)
6  WRITE(2,100) (A(I),I=1,N)
7  RETURN
8  END
```

```
1  FUNCTION ASPEC(VAL)
2  TEMP=VAL
3  IF(VAL.GT.1.0) TEMP=ALOG10(VAL)
4  IF(VAL.LT.(-1.0)) TEMP=-ALOG10(-VAL)
5  ASPEC=TEMP
6  RETURN
7  END
```

```

1      PROGRAM ACOTN
2 C
3 C      ACOTN.92 --- MYOPIC, MULTIPLE (TWO) MISSILES
4 C
5 C      USES LF=1.0, BA=0.0 FOR CONTROLS FOR FIRST STEP,
6 C      THEN MANEUVERS TO MAXIMIZE ACCELERATION NORMAL
7 C      TO THE "GUIDANCE PLANE" FOR SINGLE MISSILE CRITERION
8 C
9 C      FOR COMBINATION, CHECK ALL POSSIBLE ROLLS (+/- 60 DEG)
10 C     FOR LAMBDA OF 0 AND 1 ONLY TO FIND THE MAX(UC) OF
11 C     THE MIN(LAMBDA) OF :
12 C
13 C      $\lambda \text{LAMBDA} \# \text{ACCEL}(\text{DOT}) \text{G1} \# (1 - \text{LAMBDA}) \# \text{ACCEL}(\text{DOT}) \text{G2}$ 
14 C
15 C     WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
16 C     TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
17 C
18 C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
19 C     FUNCTION OF LAMBDA; I.E.
20 C
21 C     TA1(LAMBDA) = DMIS1(FINAL)
22 C     TA2(LAMBDA) = DMIS2(FINAL)
23 C
24 C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
25 C
26 C     WRITES TO THE TERMINAL, THEN PRINTS
27 C
28 C     RK11 = RK12 = 4.5
29 C
30 C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
31 C
32 C     PLOTTED VARIABLES ARE ---
33 C
34 C     PL(,1) = USTAR (DEG)
35 C     PL(,2) = PERFSTAR
36 C     PL(,3) = USTAR2 (DEG)
37 C     PL(,4) = PERF2
38 C     PL(,5) = GX )
39 C     PL(,6) = GY ) COMBINED USING LAMBDA
40 C     PL(,7) = GZ )
41 C     PL(,8) = LF1 (COMMANDED)
42 C     PL(,9) = SPECIFIC ENERGY (A/C)
43 C     PL(,10) = SP. EN. (MISSILE 1)
44 C     PL(,11) = ALPHA (A/C)
45 C     PL(,12) = GAMMA (A/C)
46 C     PL(,13) = SIGMA (A/C)
47 C     PL(,14) = NORM1(G) =  $(GX^2 + GY^2 + GZ^2)^{.5}$ 
48 C     PL(,15) = Z (A/C)
49 C     PL(,16) = AIRSPEED (A/C)
50 C     PL(,17) = SY1 (L.O.S. PITCH)
51 C     PL(,18) = THETA1 (L.O.S. YAW)
52 C     PL(,19) = NORM(ACCEL) =  $(AX^2 + AY^2 + AZ^2)^{.5}$ 
53 C     PL(,20) = DRAG (A/C)
54 C     PL(,21) = Z (MISSILE 1)
55 C     PL(,22) = X (A/C)
56 C     PL(,23) = X (MISSILE 1)
57 C     PL(,24) = Y (A/C)
58 C     PL(,25) = Y (MISSILE 1)
59 C     PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
60 C     PL(,27) = SYDT1

```

```

61 C      PL(,29) = THEDT1
62 C      PL(,29) = NORM1(LCS-DOT) = (SYDT1**2+THEDT1**2)**0.5
63 C      PL(,30) = GX1 )
64 C      PL(,31) = GY1 ) MISSILE 1
65 C      PL(,32) = GZ1 )
66 C      PL(,33) = NORM2
67 C      PL(,34) = GX2 )
68 C      PL(,35) = GY2 ) MISSILE 2
69 C      PL(,36) = GZ2 )
70 C      PL(,37) = NORM2(LCS-DOT)
71 C      PL(,38) = LF2C
72 C      PL(,39) = SP. EN. (MISSILE 2)
73 C      PL(,40) = SY2 (LCS FITCH)
74 C      PL(,41) = THETA2 (LCS YAW)
75 C      PL(,42) = X (MISSILE 2)
76 C      PL(,43) = Y (MISSILE 2)
77 C      PL(,44) = Z (MISSILE 2)
78 C      PL(,45) = SYDT2
79 C      PL(,46) = THEDT2
80 C      PL(,47) = LAMBDA1 * 100.
81 C      PL(,48) = GSY1
82 C      PL(,49) = GTH1
83 C      PL(,50) = GSY2
84 C      PL(,51) = GTH2
85 C      PL(,52) = GSY )
86 C      PL(,53) = GTH ) COMBINED USING LAMBDA
87 C
88 C      REAL * STRING,SDATE
89 C      INTEGER CUTIME
90 C      LOGICAL PRTEO,LONCE
91 C      COMMON /FARM7/ LONCE
92 C      COMMON /FARM9/ ISEC,DHIS1,DHIS2,PRTEO
93 C
94 C      ICUM=CUTIME(0)
95 C      CALL TIMEO0(STRING)
96 C      CALL DATE(SDATE)
97 C      WRITE(6,66) SDATE, STRING
98 C      66 FORMAT(1X,40(' '), 'ACDY4.92 ',40(' '),/,
99 C      1 51Y,40,2X,40,/,1H1)
100 C
101 C      LONCE=.FALSE.
102 C      999 CALL INIT
103 C      CALL VALUE
104 C      CALL PLOUT(2)
105 C      CALL PLOUT(6)
106 C
107 C      WRITE(2,100)
108 C      100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
109 C      1 ' FOR NO INIT PRINT')
110 C      CALL BELL(1)
111 C      READ(1,110) ISEC
112 C      110 FORMAT(I1)
113 C      IF(ISEC.NE.0) GO TO 999
114 C      STOP
115 C      END

```

```

1  SUBROUTINE VALUE
2  INTEGER LOW(2),HIGH(2)
3  REAL X0, LF, L2, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
4  1  NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
5  2  NORMG
6  DIMENSION ACCM(100,2)
7  LOGICAL PRTEG, FLAG1, FLAG2
8  INTEGER CUTINE
9  COMMON /AMP/ T0,WCSC,ALFA0,CLO,DC,LO,G,MC,C01,C02,C03,
10 1  RHO,DETA,CLAF3,SC,D01,D02,M1,CLAF1,S1,D11,RLMAX,
11 2  D12,THRESH,FK11,RK12,G1S1,ALFA1,CL1,D1,L1,ALKEAS
12 3  P1,TAL,TAFT3,TFLAG,C2C2,ALFA2,CL2,D2,L2
13  COMMON /FARM1/ JJ,KSTEP
14  COMMON /FARM2/ VTH0,VTH1,DTH,TSTEP,ICPUTM
15  COMMON /FARM3/ NO,IO
16  COMMON /FARM4/ STEP
17  COMMON /FARM5/ MANUVR
18  COMMON /FARM6/ ISEC,DMIS1,DMIS2,PRTEG
19  COMMON /ARRAY1/ XC(18),XCIN(18)
20  COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,U0,MAXLF
21  COMMON /FLVR/ PL(100,50),PMAX,PMIN,LMAX,LMIN,SPMAX,
22 1  SPMIN,ALFPAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
23 2  DATA,EMIN,XMAX,XMIN,YMAX,YMIN,AMAX,LOSHAX
24  COMMON /SSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
25  COMMON /SINCOS/ TABSIN(360),TABCOS(360)
26 C
27 EQUIVALENCE (EAXA,X0(5)),(SIGMA,X0(6))
28 EQUIVALENCE (V0,X0(4)),(V1,X0(10)),(V2,X0(16))
29 C
30 DATA TESIN /1.0/, IEXIT/'X'/, IOLIK/'0'/
31 C
32 COSGAM=COS(GAMA)
33 SINGAM=SIN(GAMA)
34 COSSIG=COS(SIGMA)
35 SINSIG=SIN(SIGMA)
36 C
37 TA=0.0
38 JJ=0
39 UUM1=1.0E10
40 UUM2=1.0E10
41 FLAG1=.FALSE.
42 FLAG2=.FALSE.
43 LAMDA1=0.5
44 C
45 C FLAG(1) INDICATES THAT MISSILE I ALREADY GOT AWAY
46 C
47 100 CONTINUE
48 IF(FLAG1) LAMDA1=0.0
49 IF(FLAG2) LAMDA1=1.0
50 LAMDA1=1.0-LAMDA1
51 IVAL=50
52 TPRINT=TA
53 VCX=V0*COSSGAM*COSSIG
54 VCY=V0*COSSGAM*SINSIG
55 VCZ=V0*SINGAM
56 VTX=V1*COX(X0(11))*COS(X0(12))
57 VTY=V1*COX(X0(11))*SIN(X0(12))
58 VTZ=V1*SIN(X0(11))
59 VZX=V2*COX(X0(17))*COS(X0(18))
60 VZY=V2*COX(X0(17))*SIN(X0(18))

```

```

61 VZ2=VZ SIN(XC(17))
62 VRELX1=VEX-V1X
63 VRELY1=VEY-V1Y
64 VFELZ1=VCZ-V1Z
65 VREL11=SQRT(VFELX1**2+VRELY1**2+VRELZ1**2)
66 VRELX2=VEX-V2X
67 VRELY2=VEY-V2Y
68 VRELZ2=VCZ-V2Z
69 VREL12=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
70 DELX1=X(1)-XC(7)
71 DELY1=X(2)-XC(8)
72 DELZ1=X(3)-XC(9)
73 DELX2=X(1)-XC(13)
74 DELY2=X(2)-XC(14)
75 DELZ2=X(3)-XC(15)
76 C
77 DUM1=AMIN1(DUM1,DMIS1)
78 DUM2=AMIN1(DUM2,DMIS2)
79 DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
80 DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
81 DSV1=0.5*DMIS1/VREL11
82 DSV2=0.5*DMIS2/VREL12
83 STEP=AMIN1(TSTEP,DSV1,DSV2)
84 DMIS=AMIN1(DMIS1,DMIS2)
85 IF(JJ,GT,0) MANUVR=1
86 LF=1.0
87 ULAST=UD
88 UD=0.0
89 IF(MANUVR.NE.1) GO TO 30C
90 C
91 UD=ULAST
92 LF=MAXLF
93 ISEC=ISEC+1
94 GX1 = VRELY1*DELZ1 - VRELZ1*DELY1
95 GY1 = VRELZ1*DELX1 - VRELX1*DELZ1
96 GZ1 = VRELX1*DELY1 - VRELY1*DELX1
97 NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
98 CX2 = VRELY2*DELZ2 - VRELZ2*DELY2
99 CY2 = VRELZ2*DELX2 - VRELX2*DELZ2
100 CZ2 = VRELX2*DELY2 - VRELY2*DELX2
101 NORM2 = SQRT(CX2*CX2 + CY2*CY2 + CZ2*CZ2)
102 IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 20C
103 C
104 C NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
105 C
106 LF=ACLF(ISEC-1)
107 GO TO 29C
108 C
109 20C CONTINUE
110 GX1=GY1/NORM1
111 GY1=GY1/NORM1
112 GZ1=GZ1/NORM1
113 GX2=GY2/NORM2
114 GY2=GY2/NORM2
115 GZ2=GZ2/NORM2
116 C
117 COEFAT=GZ1-COSGAM*GY1+SINGAM*SINSIG-GX1*SINGAM-COSSIG
118 CCEFA1=CCEFA1-(LD*TD*SIN(ALFA0))
119 COEFB1=GY1-COSGAM*GX1+SINGAM
120 CCEFB1=CCEFB1-(LD*TD*SIN(ALFA0))

```

```

121      COEFC1=((ITU-COS(ALFAC)-DL)*(GZ1+SINGAM*GY1-COSGAM*SINSIG
122      1  *GX1*(COSGAM-COSSIG))-GZ1*MO*G
123 C
124      COEFA2=GZ2-COSGAM*GY2-SINGAM*SINSIG-GX2*SINGAM-COSSIG
125      COEFA2=COEFA2*(LU+TC-SIN(ALFAC))
126      COEFC2=GY2-COSGAM*GX2-SINGAM
127      COEFC2=COEFC2*(LU+TC-SIN(ALFAC))
128      COEFC2=((TC-COS(ALFAC)-DL)*(GZ2+SINGAM*GY2-COSGAM*SINSIG
129      2  *GX2*(COSGAM-COSSIG))-GZ2*MO*G
130 C
131 C      CHECK EVERY ANGLE (BY DEGREE) WITHIN +/- RLMAX
132 C
133      NBLK=1
134      PRFMAX=0.0
135      IUU=(UO*180./PI)+160.
136      IF(IUU.GT.360) IUU=IUU-360
137      IF(IUU.LT.1) IUU=IUU+360
138      IROLL=RLMAX*STEP
139      LOW(1)=IUU-IROLL
140      IF(LOW(1).GE.1) GO TO 21C
141      NBLK=2
142      LOW(1)=1
143      LOW(2)=360+IUU-IROLL
144      HIGH(2)=360
145 21C CONTINUE
146      HIGH(1)=IUU+IROLL
147      IF(HIGH(1).LE.360) GO TO 22C
148      NBLK=2
149      HIGH(1)=IUU+IROLL-360
150      HIGH(2)=360
151      LOW(2)=LOW(1)
152      LOW(1)=1
153 22C CONTINUE
154      DO 230 IJK=1,NBLK
155      ISTRT=LOW(IJK)
156      IEND=HIGH(IJK)
157      DO 230 ILO=ISTRT,IEND
158      PERF1=COEFA1*TABCOS(IUU)+COEFC1-TAB SIN(IUU)+COEFC1
159      PERF2=COEFA2*TABCOS(IUU)+COEFC2-TAB SIN(IUU)+COEFC2
160      ABS1=ABS(PERF1)
161      ABS2=ABS(PERF2)
162      IF(FLAG1) ABS1=ABS2*2.0
163      IF(FLAG2) ABS2=ABS1*2.0
164      PERF1=AMIN1(ABS1,ABS2)
165      IF(PRFMIN.LE.PRFMAX) GO TO 230
166      IFLAG=IVAL
167      IMAX=IUU
168      PRFMAX=PRFMIN
169      IF(ABS1.LE.ABS2) GO TO 230
170      IFLAG=-IVAL
171 23C CONTINUE
172      UO=FLOAT(IMAX-160)*PI/180.
173 C
174 C      NOTE --- IFLAG IS POSITIVE FOR LAMBDA = 1, AND NEGATIVE
175 C      FOR LAMBDA = 0
176 C
177      PL(1SEC,2)=PERF1
178      PL(1SEC,4)=PERF2
179      PL(1SEC,26)=IFLAG
180      PMAX=AMAX1(PMAX,PERF1,PERF2)

```

```

181      PMIN=AMIN(PMIN,PEAF1,PEAF2)
182 C
183 C      CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
184 C
185      ACCX=((TC-COS(ALFAC)-DU) (COSGAM-COSSIG))-
186      1 ((LC+TC-SIN(ALFAC)) (SIN(UG)+SINGAM+COS(UG))-
187      2 SINGAM-COSSIG))
188      ACCY=((TC-COS(ALFAC)-DU) (COSGAM-SINSIG))+
189      1 ((LC+TC-SIN(ALFAC)) (SIN(UG)-COSGAM-COS(UG))-
190      2 SINGAM-SINSIG))
191      ACCZ=((TC-COS(ALFAC)-DU) SINGAM)-(MO-G)+
192      1 ((LC+TC-SIN(ALFAC)) (COSGAM-COS(UG)))
193      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
194 C
195 C      STORE PLOTTED VARIABLES
196 C
197      PL(ISEC,14)=NORM1
198      PL(ISEC,19)=ANORM/MO
199      PL(ISEC,30)=GX1
200      PL(ISEC,31)=GY1
201      PL(ISEC,32)=GZ1
202      PL(ISEC,33)=NORM2
203      PL(ISEC,34)=GX2
204      PL(ISEC,35)=GY2
205      PL(ISEC,36)=GZ2
206      NORM=X=AMAX1(NORMAX,NORM1,NORM2)
207      ANMAX=AMAX1(ANMAX,ANORM)
208 C
209      290 CONTINUE
210      IF(PRTED) GO TO 300
211      PRTEU=.TRUE.
212      CC1=C*141AFT6*TF1AG
213      TCH6=TPRINT
214      WRITE(6,310) TPRINT
215      WRITE(6,310) TPRINT
216      310 FORMAT(2X,'START MANEUVER AT T = ',F6.2)
217      CALL PELL(3)
218 C
219      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
220      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
221      ALPMAX=ALFAC
222      ALPMIN=ALFAC
223      ZMAX=AMIN1(XC(5),XC(9),XC(15))
224      ZMIN=AMAX1(XC(5),XC(9),XC(15))
225      XMAX=AMAX1(XC(1),XC(7),XC(13))
226      XMIN=AMIN1(XC(1),XC(7),XC(13))
227      YMAX=AMAX1(XC(2),XC(6),XC(14))
228      YMIN=AMIN1(XC(2),XC(6),XC(14))
229      VMAX=VU
230      VMIN=VU
231      CMAX=DU
232      DMIN=DU
233      LOSMAX=0.0
234 C
235      300 CONTINUE
236      ACBA(ISEC)=UG*180./PI
237      ACLF(ISEC)=LF
238      ACTI(ISEC)=TPRINT
239      ACCN(ISEC,1)=NMIS1
240      ACCN(ISEC,2)=DMIS2

```

```

241 C
242 IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
243 WRITE (2,400) TPRINT, DMIS1, DMIS2
244 WRITE (6,400) TPRINT, DMIS1, DMIS2
245 400 FORMAT (5X, 'TIME = ', F10.3, 2X, 'DSEP1 = ', G12.3, 2X,
246 1 'DSEP2 = ', G12.3)
247 C
248 477 CONTINUE
249 IF((V1.LT.VTHC).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
250 IF(DMIS.LT.DTH) GO TO 520
251 IF(TA.LT.3.0) GO TO 480
252 FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
253 FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
254 IF(FLAG1.AND.FLAG2) GO TO 540
255 480 CONTINUE
256 CALL INTEOX
257 TA=TA+STEP
258 JJ=JJ+1
259 IF(ISEC.GE.10) GO TO 530
260 GO TO 100
261 C
262 510 CONTINUE
263 WRITE (2,515) TPRINT
264 WRITE (6,515) TPRINT
265 515 FORMAT(1X, 'A/C OR MISSILE VEL. IS TOO LOW ',
266 1 'AT TIME: ', F10.3, ' **', /)
267 GO TO 600
268 C
269 C HIT OCCURRED, PRINT OUT
270 520 CONTINUE
271 WRITE(2,525) TPRINT,DUM1,DUM2
272 WRITE(6,525) TPRINT,DUM1,DUM2
273 525 FORMAT(1X, '***** HIT AT TIME = ', F10.3,
274 1 ' ', ' ', /, 5X, 'BEST DSEPS WERE 1:',
275 2 ' G15.6, ', & 2: ' G15.6, /)
276 GO TO 600
277 C
278 530 CONTINUE
279 WRITE(2,535) TPRINT
280 WRITE(6,535) TPRINT
281 535 FORMAT(5X, 'TIME LIMIT AT T = ', F0.2)
282 GO TO 600
283 C
284 C CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
285 540 CONTINUE
286 WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
287 WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
288 544 FORMAT(3X, 'CLOSURE RATE NEGATIVE AT TIME = ', F10.3,
289 1 ' ', ' ', /, 5X, 'TA1 : BEST DSEP = ', G15.6, ' ', NOW = ', G15.6, /,
290 2 ' 5X, 'TA2 : BEST DSEP = ', G15.6, ' ', NOW = ', G15.6, /)
291 GO TO 600
292 C
293 600 CONTINUE
294 CALL DELL(1)
295 C
296 READ(1,635) LOGIC
297 IF(LOGIC.EQ.1EXIT) RETURN
298 N10=C
299 DO 620 J=1,N10
300 JJ=J+0

```

```

301      KK=J+12
302      TEMP=XG(JJ)
303      TEMP1=XO(JJJ)
304      TEMP2=AC(KK)
305      IF(J.LT.5) GO TO 605
306      TEMP=TEMP*180./PI
307      TEMP1=TEMP1*180./PI
308      TEMP2=TEMP2*180./PI
309      605 CONTINUE
310      WRITE(2,C10) J,TEMP,JJJ,TEMP1,KK,TEMP2
311      WRITE(6,C10) J,TEMP,JJJ,TEMP1,KK,TEMP2
312      610 FORMAT(2X,"XU(",I1,"): ",G13.4,4X,
313      1 "AC(",I2,"): ",G13.4,4X,
314      2 "AC(",I2,"): ",G13.4)
315      620 CONTINUE
316 C
317      WRITE(2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
318      WRITE(6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
319      625 FORMAT(1,10X,"DELX1: ",G12.3,2X,"DELY1: ",G12.3,/,
320      1 10X,"DELZ1: ",G12.3,4X,"DMIS1: ",G12.3,/,
321      2 10X,"BEST DMIS WAS ",G12.3)
322      WRITE(2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
323      WRITE(6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
324      630 FORMAT(1,10X,"DELX2: ",G12.3,2X,"DELY2: ",G12.3,/,
325      1 10X,"DELZ2: ",G12.3,2X,"DMIS2: ",G12.3,/,
326      2 10X,"BEST DMIS WAS ",G12.3)
327      CALL BELL(1)
328      READ(1,645) LOGIC
329      IF(LOGIC.EQ.1EXIT) RETURN
330      IF(LOGIC.EQ.1GUIK) GO TO 660
331      635 FORMAT(A1)
332 C
333      WRITE(6,631)
334      631 FORMAT(1H1)
335      DO 650 I=1,ISEC
336      IF(MOD(I,20).EQ.0) CALL BELL(1)
337      IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
338      IF(LOGIC.EQ.1EXIT) RETURN
339      IF(LOGIC.EQ.1GUIK) GO TO 660
340      WRITE(2,635) ACT1(1),ACBA(1),PL(1,6),PL(1,38)
341      635 FORMAT(1A,"TIME ",F5.2,3A,"ACBA ",F6.2,3X,"LF1C ",F8.1,
342      1 1A,"LF2C ",F8.1)
343      650 CONTINUE
344      660 CONTINUE
345      DO 670 I=1,ISEC
346      WRITE(6,637) ACT1(1),ACLF(1),ACPA(1),PL(1,6),PL(1,38),
347      1 PL(1,19),FL(1,14),FL(1,11),PL(1,20),
348      2 (ACU(I,J),J=1,2),FL(1,47)
349      637 FORMAT(1A,"TIME ",F5.2,2,3X,"ACLF = ",F5.1,3X,
350      1 "ACBA(DEG) = ",F5.2,3X,"LF1C = ",F6.1,3X,
351      2 "LF2C = ",F5.2,3X,"NORM(A) = ",G11.2,/,20X,
352      3 "NORM1(G) = ",G11.2,3X,"ALPHA = ",F6.2,3X,
353      4 "UPAG = ",G11.2,/,20X,"DMIS1 = TA(1) = ",
354      5 F5.2,3X,"DMIS2 = TA(2) = ",F6.2,5X,
355      6 "LAMBDA1 = ",-2F5.2,/)
356      670 CONTINUE
357 C
358      IDUM=CUTIME(U)-ICPUTM
359      WRITE(2,C39) IDUM
360      WRITE(6,C39) IDUM

```

```
361 639 FORMAT(4X,'CPU TIME (SECONDS) = ',15,CPF5.0)
362 C
363     RETURN
364     END
```

```

1  PROGRAM ACDDYN
2  C
3  C  ACDDYN.93 --- MYOPIC, MULTIPLE (TWO) MISSILES
4  C
5  C  USES LF=1.0, LA=0.0 FOR CONTROLS FOR FIRST STEP,
6  C  THEN MANEUVERS TO MAXIMIZE ACCELERATION IN THE
7  C  "GUIDANCE PLANE" ANTI-PROPORTIONALLY TO THE
8  C  MISSILE'S GUIDANCE
9  C
10 C  FOR COMBINATION, USE DO LOOP OF LAMBDA'S, I.E.
11 C  LAMBDA = 0.0 THRU 1.0 BY 0.05, AT EACH STEP,
12 C  TO FIND THE MIN(LAMBDA) OF THE MAX(UO) OF :
13 C
14 C      [(LAMBDA*ACCEL(DOT)G1)+((1-LAMBDA)*ACCEL(DOT)G2)]
15 C
16 C  WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
17 C  TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
18 C
19 C  DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
20 C  FUNCTION OF LAMBDA; I.E.
21 C
22 C      TA1(LAMBDA) = DMIS1(FINAL)
23 C      TA2(LAMBDA) = DMIS2(FINAL)
24 C
25 C  ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
26 C
27 C  WRITES TO THE TERMINAL, THEN PRINTS
28 C
29 C  RK11 = RK12 = 4.5
30 C
31 C  STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
32 C
33 C  PLOTTED VARIABLES ARE ---
34 C
35 C      PL(,1) = USTAR (DEG)
36 C      PL(,2) = PERFSTAR
37 C      PL(,3) = USTAR2 (DEG)
38 C      PL(,4) = PERF2
39 C      PL(,5) = GX )
40 C      PL(,6) = GY ) COMBINED USING LAMBDA
41 C      PL(,7) = GZ )
42 C      PL(,8) = LF1 (COMMANDED)
43 C      PL(,9) = SPECIFIC ENERGY (A/C)
44 C      PL(,10) = SP. EN. (MISSILE 1)
45 C      PL(,11) = ALPHA (A/C)
46 C      PL(,12) = GAMMA (A/C)
47 C      PL(,13) = SIGMA (A/C)
48 C      PL(,14) = NORM1(G) = (GX**2+GY**2+GZ**2)**0.5
49 C      PL(,15) = Z (A/C)
50 C      PL(,16) = AIRSPEED (A/C)
51 C      PL(,17) = SY1 (L.O.S. PITCH)
52 C      PL(,18) = THETA1 (L.O.S. YAW)
53 C      PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
54 C      PL(,20) = DRAG (A/C)
55 C      PL(,21) = Z (MISSILE 1)
56 C      PL(,22) = X (A/C)
57 C      PL(,23) = X (MISSILE 1)
58 C      PL(,24) = Y (A/C)
59 C      PL(,25) = Y (MISSILE 1)
60 C      PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2

```

```

61 C      PL(,27) = SYDT1
62 C      PL(,28) = THEDT1
63 C      PL(,29) = NORM1(LOS,DT) = (SYDT1**2+THEDT1**2)**0.5
64 C      PL(,30) = GX1 )
65 C      PL(,31) = GY1 ) MISSILE 1
66 C      PL(,32) = GZ1 )
67 C      PL(,33) = NORM2
68 C      PL(,34) = GX2 )
69 C      PL(,35) = GY2 ) MISSILE 2
70 C      PL(,36) = GZ2 )
71 C      PL(,37) = NORM2(LOS,DT)
72 C      PL(,38) = LF2C
73 C      PL(,39) = SP. EN. (MISSILE 2)
74 C      PL(,40) = SY2 (LOS PITCH)
75 C      PL(,41) = THETA2 (LOS YAW)
76 C      PL(,42) = X (MISSILE 2)
77 C      PL(,43) = Y (MISSILE 2)
78 C      PL(,44) = Z (MISSILE 2)
79 C      PL(,45) = SYDT2
80 C      PL(,46) = THEDT2
81 C      PL(,47) = LAMDA1 * 100.
82 C      PL(,48) = GSY1
83 C      PL(,49) = GTH1
84 C      PL(,50) = GSY2
85 C      PL(,51) = GTH2
86 C      PL(,52) = GSY )
87 C      PL(,53) = GTH ) COMBINED USING LAMBDA
88 C
89 C      REAL*8 STRING,SDATE
90 C      INTEGER CUTIME
91 C      LOGICAL PRTEO,LONCE
92 C      COMMON /PARM7/ LONCE
93 C      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTEO
94 C
95 C      IDUM=CUTIME(0)
96 C      CALL TIMEO0(STRING)
97 C      CALL DATE(SDATE)
98 C      WRITE(5,66) SDATE, STRING
99 C      66 FORMAT(15X,40(" "), " ACODYN.93 ",40(" "),/,
100 C      1 SIX,A0,2X,A0,/,1H1)
101 C
102 C      LONCE=.FALSE.
103 C      999 CALL I,IF
104 C      CALL VALUE
105 C      CALL PLOUT(2)
106 C      CALL PLOUT(6)
107 C
108 C      WRITE(2,100)
109 C      100 FORMAT(" ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 "
110 C      1 " FOR NO INIT PRINT")
111 C      CALL BELL(1)
112 C      READ(1,110) ISEC
113 C      110 FORMAT(I1)
114 C      IF(ISEC.NE.0) GO TO 999
115 C      STOP
116 C      END

```

```

1  SUBROUTINE VALUE
2  REAL MU, LF, LO, M1, LF1, L1, NDM1, LMAX, LMIN, MAXLF,
3  1  NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
4  2  NORMG
5  DIMENSION ACU(100,2)
6  LOGICAL PRTEU, FLAG1, FLAG2
7  INTEGER CUTIME
8  COMMON /AMP/ TO,MOJO,ALFAO,CLO,DO,LO,G,MO,C01,C02,C03,
9  1  KHU,BETA,CLAF0,SO,DO1,DO2,H1,CLAF1,S1,D11,RLMAX,
10  2  D1C,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALKEAS
11  3  ,PI,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
12  COMMON /PARM1/ JJ,KSTEP
13  COMMON /PARM2/ VTHU,VTH1,DTH,TSTEP,ICPUTM
14  COMMON /PARM3/ MU,LO
15  COMMON /PARM4/ STEP
16  COMMON /PARM5/ MANUVR
17  COMMON /PARM7/ ISEL,DMIS1,DMIS2,PRTEU
18  COMMON /ARRAY1/ XU(13),XUIN(13)
19  COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
20  COMMON /PLVR/ PL(100,55),PNAX,PMIN,LMAX,LMIN,SPMAX,
21  1  SPMIN,ALFMAX,ALFMIN,NORMAX,LMAX,ZMIN,VMAX,VMIN,
22  2  DRAA,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
23  COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2,
24  1  SYDT1,THEDT1,SYDT2,THEDT2
25 C
26  EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
27  EQUIVALENCE (VU,XO(4)),(V1,XO(10)),(V2,XO(16))
28 C
29  DATA TESTN /1.0E-06/, IEXIT/'X'/, IQUICK/'Q'/
30 C
31  PERF(U)=COEFA*COS(U)+COEFB*SIN(U)+COEFC
32 C
33  COSSGA=COS(GAMA)
34  SINGAM=SIN(GAMA)
35  COSSIG=COS(SIGMA)
36  SINSIG=SIN(SIGMA)
37  COSSY1=COS(PSY1)
38  COSSY2=COS(PSY2)
39  COSTH1=COS(THETA1)
40  COSTH2=COS(THETA2)
41  SINSY1=SIN(PSY1)
42  SINSY2=SIN(PSY2)
43  SINTH1=SIN(THETA1)
44  SINTH2=SIN(THETA2)
45 C
46  TA=0.0
47  JJ=0
48  DU1=1.0E10
49  DU2=1.0E10
50  FLAG1=.FALSE.
51  FLAG2=.FALSE.
52  LAMDA1=0.5
53 C
54 C  FLAG(1) INDICATES THAT MISSILE 1 ALREADY GOT AWAY
55 C
56  100 CONTINUE
57  IF(FLAG1) LAMDA1=0.0
58  IF(FLAG2) LAMDA1=1.0
59  LAMDA2=1.0-LAMDA1
60  IVAL=50

```

```

61      TPRINT=TA
62      VOX=V7*COSGAH*COSSIG
63      VOY=V7*COSGAH*SINSIG
64      VOZ=V7*SINGAH
65      V1X=V1*COS(XU(11))*COS(XU(12))
66      V1Y=V1*COS(XU(11))*SIN(XU(12))
67      V1Z=V1*SIN(XU(11))
68      V2X=V2*COS(XU(17))*COS(XU(13))
69      V2Y=V2*COS(XU(17))*SIN(XU(13))
70      V2Z=V2*SIN(XU(17))
71      VRELX1=VOX-V1X
72      VRELY1=VOY-V1Y
73      VRELZ1=VOZ-V1Z
74      VREL1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
75      VRELX2=VOX-V2X
76      VRELY2=VOY-V2Y
77      VRELZ2=VOZ-V2Z
78      VREL2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
79      DELX1=XO(1)-XO(7)
80      DELY1=XO(2)-XO(8)
81      DELZ1=XO(3)-XO(9)
82      DELX2=XO(1)-XO(13)
83      DELY2=XO(2)-XO(14)
84      DELZ2=XO(3)-XO(15)
85 C
86      DUM1=AMIN1(DUM1,DMS1)
87      DUM2=AMIN1(DUM2,DMS2)
88      DMS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
89      DMS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
90      DSV1=0.5*DMS1/VREL1
91      DSV2=0.5*DMS2/VREL2
92      STEP=AMIN1(TSTEP,DSV1,DSV2)
93      DMS=AMIN1(DMS1,DMS2)
94      IF(JJ.GT.0) MANUVR=1
95      LF=1.0
96      ULAST=UO
97      UO=0.0
98      IF(MANUVR.NE.1) GO TO 300
99 C
100     UO=ULAST
101     LF=MAXLF
102     ISEC=ISEC+1
103     GX1 = -(SYDT1*SINSY1*COSTH1 + THEDT1*COSSY1*SINTH1)
104     GY1 = THEDT1*COSSY1*COSTH1 - SYDT1*SINSY1*SINTH1
105     GZ1 = SYDT1*COSSY1
106     NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
107     GX2 = -(SYDT2*SINSY2*COSTH2 + THEDT2*COSSY2*SINTH2)
108     GY2 = THEDT2*COSSY2*COSTH2 - SYDT2*SINSY2*SINTH2
109     GZ2 = SYDT2*COSSY2
110     NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
111     IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
112 C
113 C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
114 C
115     LF=ACLF(ISEC-1)
116     GO TO 290
117 C
118 200 CONTINUE
119     GX1=GX1/NORM1
120     GY1=GY1/NORM1

```

```

121      GZ1=GZ1/NORM1
122      GX2=GX2/NORM2
123      GY2=GY2/NORM2
124      GZ2=GZ2/NORM2
125      PEKFA=1.0E20
126 C
127      DO 277 LAMB=1,21
128      LAMDA1=FLOAT(LAMB-1)/20.
129      LAMDA2=1.0-LAMDA1
130      IF(FLAG1.AND.(LAMB.GT.1)) GO TO 280
131      IF(FLAG2.AND.(LAMB.LT.21)) GO TO 270
132 C
133      GX=LAMDA1*GX1+LAMDA2*GX2
134      GY=LAMDA1*GY1+LAMDA2*GY2
135      GZ=LAMDA1*GZ1+LAMDA2*GZ2
136 C
137      COEFA=GZ*COSGAM-GY*SINGAM*SINSIG-GX*SINGAM*COSSIG
138      COEFA=COEFA*(LG+TD*SIN(ALFA))
139      COEFB=GY*COSGAM-GX*SINGAM
140      COEFB=COEFB*(LG+TD*SIN(ALFA))
141      COEFC=((TU*COS(ALFA)-DJ)*(GZ*SINGAM+GY*COSGAM*SINSIG
142      1  +GX*COSGAM*COSSIG)) - GZ*MD*G
143 C
144      USTAR=ATAN2(COEFB,COEFA)
145      USTAR2=USTAR+PI
146      IF(USTAR.GT.0.0) USTAR2=USTAR-PI
147 220 CONTINUE
148      PERF1=PERF(USTAR)
149      PERF2=PERF(USTAR2)
150      ABST=ABS(PERF1)
151      ABS2=ABS(PERF2)
152 C
153 C      USE "BEST" U AND RATE LIMIT TO RLMAX
154 C
155      IFLAG=IVAL
156      UMAX=USTAR
157      PERMIN=ABST
158      IF(ABST.GE.ABS2) GO TO 230
159      UMAX=USTAR2
160      IFLAG=-IVAL
161      PERMIN=ABS2
162 230 CONTINUE
163      KOLL=(UMAX-U0)*180./PI
164      IF(ROLL.GT.180.) ROLL=KOLL-360.
165      IF(ROLL.LT.(-180.)) ROLL=ROLL+360.
166      AROLL=ABS(ROLL)
167 C
168 C      NOW RATE-LIMIT WHICHEVER ANGLE WE GET TO RLMAX
169 C      DEG PER SEC...
170 C
171      IF(AROLL.LE.(RLMAX*STEP)) GO TO 250
172      ROLL=ROLL*RLMAX*STEP/AROLL
173      IF(AROLL.LT.180.) GO TO 250
174 C
175 C      ROLL IS ESSENTIALLY A COMPLETE FLIP --- USE PLUS/MINUS
176 C      RLMAX INTO PERF FUNCTION TO CHECK FOR BEST WAY
177 C      TO MAKE FLIPS
178 C
179      IVAL=75
180      USTAR=U0+(RLMAX*STEP*PI/180.)

```

```

181 IF(USTAR.GT.PI) USTAR=USTAR-(2.0*PI)
182 USTAR2=UU-(RLMAX*STEP*PI/180.)
183 IF(USTAR2.LE.(-PI)) USTAR2=USTAR2+(2.0*PI)
184 GO TO 220
185 C
186 250 CONTINUE
187 IF(PLMIN.LE.PERFM) GO TO 270
188 PERFM=PERMIN
189 JMIN=LAMU
190 RLMIN=ROLL
191 PL(ISEC,1)=USTAR*180./PI
192 PL(ISEC,2)=PERFST
193 PL(ISEC,3)=USTAR2*180./PI
194 PL(ISEC,4)=PERF2
195 PL(ISEC,5)=GX
196 PL(ISEC,6)=GY
197 PL(ISEC,7)=GZ
198 PL(ISEC,26)=IFLAG
199 PL(ISEC,47)=5*(JMIN-1)
200 PMAX=AMAX1(PMAX,PERFST,PERF2)
201 PMIN=AMIN1(PMIN,PERFST,PERF2)
202 270 CONTINUE
203 280 CONTINUE
204 C
205 ROLL=RLMIN
206 UU=UU+(ROLL*PI/180.)
207 IF(UU.GT.PI) UU=UU-(2.0*PI)
208 IF(UU.LE.(-PI)) UU=UU+(2.0*PI)
209 C
210 C CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
211 C
212 ACCX=((TU*COS(ALFAU)-DU)*(COSGAM*COSSIG))-
213 1 ((LU+U*SIN(ALFAU))*(SIN(UO)*SINGAM+COS(UO)*
214 2 SINGAM*COSSIG))
215 ACCY=((TU*COS(ALFAU)-DU)*(COSGAM*SINSIG))+
216 1 ((LU+U*SIN(ALFAU))*(SIN(UO)*COSGAM-COS(UO)*
217 2 SINGAM*SINSIG))
218 ACCZ=((TU*COS(ALFAU)-DU)*SINGAM)-(MU*G)+
219 1 ((LU+U*SIN(ALFAU))*COSGAM*COS(UO))
220 ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
221 C
222 C STORE PLOTTED VARIABLES
223 C
224 PL(ISEC,14)=NORM1
225 PL(ISEC,19)=ANORM/40
226 PL(ISEC,30)=GX1
227 PL(ISEC,31)=GY1
228 PL(ISEC,32)=GZ1
229 PL(ISEC,33)=NORM2
230 PL(ISEC,34)=GX2
231 PL(ISEC,35)=GY2
232 PL(ISEC,36)=GZ2
233 NORMAX=AMAX1(NORMAX,NORM1,NORM2)
234 ANMAX=AMAX1(ANMAX,ANORM)
235 C
236 290 CONTINUE
237 IF(PRTED) GO TO 300
238 PRTED=.TRUE.
239 C31=C31+TAFT6*TF16
240 ICH6=TPRINT

```

```

241      WRITE(2,310) TPRINT
242      WRITE(6,310) TPRINT
243 310  FORMAT(2X,"START MANEUVER AT T = ",F6.2)
244      CALL BELL(3)
245 C
246      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
247      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
248      ALFMAX=ALFAO
249      ALFMIN=ALFAO
250      ZMAX=AMAX1(XU(3),XU(9),XU(15))
251      ZMIN=AMIN1(XU(3),XU(9),XU(15))
252      XMAX=AMAX1(XU(1),XU(7),XU(13))
253      XMIN=AMIN1(XU(1),XU(7),XU(13))
254      YMAX=AMAX1(XU(2),XU(8),XU(14))
255      YMIN=AMIN1(XU(2),XU(8),XU(14))
256      VMAX=VU
257      VMIN=VU
258      OMAX=DO
259      OMIN=DO
260      LOSMAX=0.0
261 C
262 300  CONTINUE
263      ACBA(ISEC)=UU*180./PI
264      ACLF(ISEC)=LF
265      ACT1(ISEC)=TPRINT
266      ACDM(ISEC,1)=DMIS1
267      ACDM(ISEC,2)=DMIS2
268 C
269      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
270      WRITE (2,400) TPRINT, DMIS1, DMIS2
271      WRITE (6,400) TPRINT, DMIS1, DMIS2
272 400  FORMAT (1X,"TIME = ",F10.3,2X,"DSEP1 = ",G12.3,2X,
273 1      "DSEP2 = ",G12.3)
274 C
275 477  CONTINUE
276      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
277      IF(DMIS.LT.DTH) GO TO 520
278      IF(TA.LT.3.0) GO TO 480
279      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
280      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
281      IF(FLAG1.AND.FLAG2) GO TO 540
282 480  CONTINUE
283      CALL INTBOX
284      TA=TA+STEP
285      JJ=JJ+1
286      IF(ISEC.GE.IU) GO TO 530
287      GO TO 100
288 C
289 510  CONTINUE
290      WRITE (2,515) TPRINT
291      WRITE (6,515) TPRINT
292 515  FORMAT(1X,"** A/C OR MISSILE VEL. IS TOO LOW ",
293 1      "AT TIME: ",F10.3," **",/)
294      GO TO 600
295 C
296 C      HIT OCCURRED, PRINT OUT
297 520  CONTINUE
298      WRITE(2,525) TPRINT,DUM1,DUM2
299      WRITE(6,525) TPRINT,DUM1,DUM2
300 525  FORMAT(1X,"***** HIT AT TIME = ",F10.3,

```

```

301      1      '*****',/,5X,'BEST DSEPS WERE 1:',
302      2      G15.6,'', & 2:',G15.6,/)
303      GO TO 600
304 C
305      530 CONTINUE
306      WRITE(2,535) TPRINT
307      WRITE(3,535) TPRINT
308      535 FORMAT(5A,'TIME LIMIT AT T = ',F6.2)
309      GO TO 600
310 C
311 C      CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
312      540 CONTINUE
313      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
314      WRITE(3,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
315      544 FORMAT(5A,'***** CLOSURE RATE NEGATIVE AT TIME = ',F10.3,
316      1      '*****',/,5X,'TA1 : BEST DSEP = ',G15.6,'', NOW = ',G15.6,/,
317      2      5A,'TA2 : BEST DSEP = ',G15.6,'', NOW = ',G15.6,/)
318      GO TO 600
319 C
320      600 CONTINUE
321      CALL DELL(1)
322 C
323      READ(1,635) LOGIC
324      IF(LOGIC.EQ.IEXIT) RETURN
325      N10=0
326      DO 620 J=1,N10
327      JJJ=J+6
328      KK=J+12
329      TEMP=X0(J)
330      TEMP1=X0(JJJ)
331      TEMP2=X0(KK)
332      IF(J.LT.5) GO TO 605
333      TEMP=TEMP*120./PI
334      TEMP1=TEMP1*120./PI
335      TEMP2=TEMP2*120./PI
336      605 CONTINUE
337      WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
338      WRITE(3,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
339      610 FORMAT(2X,'XJ(',I1,') = ',G13.4,4X,
340      1      'XJ(',I1,') = ',G13.4,4X,
341      2      'XJ(',I2,') = ',G13.4)
342      620 CONTINUE
343 C
344      WRITE (2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
345      WRITE (3,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
346      625 FORMAT(/,10X,'DELX1: ',G12.3,2X,'DELY1: ',G12.3,/,
347      1      10X,'DELZ1: ',G12.3,2X,'DMIS1: ',G12.3,/,
348      2      10X,'BEST DMIS WAS ',G12.3)
349      WRITE (2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
350      WRITE (3,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
351      630 FORMAT(/,10X,'DELX2: ',G12.3,2X,'DELY2: ',G12.3,/,
352      1      10X,'DELZ2: ',G12.3,2X,'DMIS2: ',G12.3,/,
353      2      10X,'BEST DMIS WAS ',G12.3)
354      CALL DELL(1)
355      READ(1,635) LOGIC
356      IF(LOGIC.EQ.IEXIT) RETURN
357      IF(LOGIC.EQ.IGJIK) GO TO 660
358      635 FORMAT(A1)
359 C
360      WRITE(6,631)

```

```

361 631 FORMAT(1H1)
362 DO 650 I=1,ISEC
363 IF(MOD(I,20).EQ.0) CALL BELL(1)
364 IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
365 IF(LOGIC.EQ.IEXIT) RETURN
366 IF(LOGIC.EQ.IQUIK) GO TO 660
367 WRITE(2,636) ACTI(1),ACBA(1),PL(1,3),PL(1,38)
368 636 FORMAT(1X,'TIME ',F5.2,3X,'ACBA ',F8.2,3X,'LF1C ',F8.1,
369 1 3X,'LF2C ',F8.1)
370 650 CONTINUE
371 660 CONTINUE
372 DO 670 I=1,ISEC
373 WRITE(6,637) ACTI(1),ACLF(1),ACBA(1),PL(1,8),PL(1,38),
374 1 PL(1,19),PL(1,14),PL(1,11),PL(1,20),
375 2 (ACUM(I,J),J=1,4),PL(1,47)
376 637 FORMAT(1X,'TIME = ',JPF6.2,3X,'ACLF = ',F5.1,3X,
377 1 'ACBA(DEG) = ',F5.2,3X,'LF1C = ',F8.1,3X,
378 2 'LF2C = ',F8.2,3X,'NORM(A) = ',G11.2,/,20X,
379 3 'NORM1(G) = ',G11.2,3X,'ALPHA = ',F8.2,3X,
380 4 'DPAG = ',G11.2,/,20X,'DMIS1 = TA(1) = ',
381 5 F5.2,5X,'DMIS2 = TA(2) = ',F8.2,5X,
382 6 'LAMBDA1 = ',-2PF5.2,/)
383 670 CONTINUE
384 C
385 IDUM=CUTIME(0)-ICPUTM
386 WRITE(2,639) IDUM
387 WRITE(6,639) IDUM
388 639 FORMAT(4X,'CPU TIME (SECONDS) = ',I5,0PF5.0)
389 C
390 RETURN
391 END

```

```

1      PROGRAM ACDYN
2 C
3 C      ACDYN.94 --- MYOPIC, MULTIPLE (T.O) MISSILES
4 C
5 C      USES LF=1.0, EA=0.0 FOR CONTROLS FOR FIRST STEP,
6 C          THEN MANEUVERS TO MAXIMIZE ACCELERATION IN THE
7 C          "GUIDANCE PLANE" ANTI-PROPORTIONALLY TO THE
8 C          MISSILE'S GUIDANCE
9 C
10 C     FOR COMBINATION, CHECK ALL POSSIBLE ROLLS (+/- 60 DEG)
11 C     FOR LAMBDA OF 0 AND 1 ONLY TO FIND THE MAX(U0) OF
12 C     THE MIN(LAMBDA) OF :
13 C
14 C         (LAMBDA*ACCEL(DOT)G1)+(1-LAMBDA)*ACCEL(DOT)G2)/
15 C
16 C     WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
17 C     TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
18 C
19 C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
20 C     FUNCTION OF LAMBDA; I.E.
21 C
22 C         TA1(LAMBDA) = DMIS1(FINAL)
23 C         TA2(LAMBDA) = DMIS2(FINAL)
24 C
25 C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
26 C
27 C     WRITES TO THE TERMINAL, THEN PRINTS
28 C
29 C     RK11 = RK12 = 4.5
30 C
31 C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
32 C
33 C     PLOTTED VARIABLES ARE ---
34 C
35 C         PL(,1) = USTAR (DEG)
36 C         PL(,2) = PERFSTAR
37 C         PL(,3) = USTAR2 (DEG)
38 C         PL(,4) = PERF2
39 C         PL(,5) = GX )
40 C         PL(,6) = GY ) COMBINED USING LAMBDA
41 C         PL(,7) = GZ )
42 C         PL(,8) = LF1 (COMMANDED)
43 C         PL(,9) = SPECIFIC ENERGY (A/C)
44 C         PL(,10) = SP. EN. (MISSILE I)
45 C         PL(,11) = ALPHA (A/C)
46 C         PL(,12) = GAMMA (A/C)
47 C         PL(,13) = SIGMA (A/C)
48 C         PL(,14) = NORM1(G) = (GX**2+GY**2+GZ**2)**0.5
49 C         PL(,15) = Z (A/C)
50 C         PL(,16) = AIRSPEED (A/C)
51 C         PL(,17) = SY1 (L.O.S. PITCH)
52 C         PL(,18) = THETA1 (L.O.S. YAW)
53 C         PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
54 C         PL(,20) = DRAG (A/C)
55 C         PL(,21) = Z (MISSILE 1)
56 C         PL(,22) = X (A/C)
57 C         PL(,23) = X (MISSILE 1)
58 C         PL(,24) = Y (A/C)
59 C         PL(,25) = Y (MISSILE 1)
60 C         PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2

```

```

61 C      PL(,27) = SYDT1
62 C      PL(,28) = THEDT1
63 C      PL(,29) = NORM1(LOS-DOT) = (SYDT1**2+THEDT1**2)**0.5
64 C      PL(,30) = GX1   )
65 C      PL(,31) = GY1   ) MISSILE 1
66 C      PL(,32) = GZ1   )
67 C      PL(,33) = NORM2
68 C      PL(,34) = GX2   )
69 C      PL(,35) = GY2   ) MISSILE 2
70 C      PL(,36) = GZ2   )
71 C      PL(,37) = NORM2(LOS-DOT)
72 C      PL(,38) = LF2C
73 C      PL(,39) = SP. EN. (MISSILE 2)
74 C      PL(,40) = SY2 (LOS PITCH)
75 C      PL(,41) = THETA2 (LOS YAW)
76 C      PL(,42) = X (MISSILE 2)
77 C      PL(,43) = Y (MISSILE 2)
78 C      PL(,44) = Z (MISSILE 2)
79 C      PL(,45) = SYDT2
80 C      PL(,46) = THEDT2
81 C      PL(,47) = LAMDA1 * 100.
82 C      PL(,48) = GSY1
83 C      PL(,49) = GTH1
84 C      PL(,50) = GSY2
85 C      PL(,51) = GTH2
86 C      PL(,52) = GSY )
87 C      PL(,53) = GTH ) COMBINED USING LAMDA
88 C
89 C      REAL*8 STRING,SDATE
90 C      INTEGER CUTIME
91 C      LOGICAL PRTEO,LONCE
92 C      COMMON /PARAM7/ LONCE
93 C      COMMON /PARAM9/ ISEC,DMIS1,DMIS2,PRTEO
94 C
95 C      IDUM=CUTIME(0)
96 C      CALL TIMEOD(STRING)
97 C      CALL DATE(SDATE)
98 C      WRITE(3,66) SDATE, STRING
99 C      66 FORMAT(15X,40(' '), 'ACDYN.94 ',40(' '),/,
100 C      1 51X,15,2X,40,/,1H1)
101 C
102 C      LONCE=.FALSE.
103 C      999 CALL INIT
104 C      CALL VALUE
105 C      CALL PLOUT( )
106 C      CALL PLOUT( )
107 C
108 C      WRITE(2,100)
109 C      100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
110 C      1 ' FOR NO INIT PRINT')
111 C      CALL BELL(1)
112 C      READ(1,110) ISEC
113 C      110 FORMAT(I1)
114 C      IF(ISEC.NE.0) GO TO 999
115 C      STOP
116 C      END

```

```

1      SUBROUTINE VALUE
2      INTEGER LOW(2),HIGH(2)
3      REAL MU, LF, LC, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
4      1      NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
5      2      NUORG
6      DIMENSION ACDM(100,2)
7      LOGICAL PRTEB, FLAG1, FLAG2
8      INTEGER CUTIME
9      COMMON /AMP/ TO,WCSD,ALFAO,CLO,DO,LO,G,MO,CO1,CO2,CO3,
10     1      RHU,BETA,CLAFU,SO,DO1,DO2,M1,CLAF1,S1,D11,RLMAX,
11     2      DT2,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALREAS
12     3      ,P1,TAU,TAFTB,TFLAG,WS2,ALFA2,CL2,D2,L2
13     COMMON /PAKE1/ JJ,KSTEP
14     COMMON /PAKE2/ VTHU,VTH1,DTH,TSTEP,ICPUTM
15     COMMON /PAKE3/ NU,I0
16     COMMON /PAKE4/ STEP
17     COMMON /PAKE5/ MANOVR
18     COMMON /PAKE7/ ISEC,DMIS1,DMIS2,PRTEB
19     COMMON /ARRAY1/ XO(16),XUIN(19)
20     COMMON /CONTR/ ACLE(100),AC9A(100),ACTI(100),TCHG,LF,UO,MAXLF
21     COMMON /PLVA/ PL(100,55),PMAA,PMIN,LMAX,LMIN,SPMAX,
22     1      SP1V,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
23     2      DMAX,LMIN,XMAX,XMIN,YMAX,YMIN,AMAX,LOSMAX
24     COMMON /ASL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2,
25     1      SYDT1,THEDT1,SYDT2,THEDT2
26     COMMON /SINCOS/ TABSIN(360),TABCOS(360)
27 C
28     EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
29     EQUIVALENCE (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
30 C
31     DATA TESTN /1.0E-06/, IEXIT/'X'/, IQUIK/'Q'/
32 C
33     COSGAM=COS(GAMA)
34     SINGAM=SIN(GAMA)
35     COSSIG=COS(SIGMA)
36     SINSIG=SIN(SIGMA)
37     COSSY1=COS(PSY1)
38     COSSY2=COS(PSY2)
39     COSTH1=COS(THETA1)
40     COSTH2=COS(THETA2)
41     SINSY1=SIN(PSY1)
42     SINSY2=SIN(PSY2)
43     JINTH1=SIN(THETA1)
44     JINTH2=SIN(THETA2)
45 C
46     TA=0.0
47     JJ=0
48     DU1=1.0E10
49     DU2=1.0E10
50     FLAG1=.FALSE.
51     FLAG2=.FALSE.
52     LAMDA1=0.5
53 C
54 C     FLAG(1) INDICATES THAT MISSILE 1 ALREADY GOT AWAY
55 C
56     100 CONTINUE
57     IF(FLAG1) LAMDA1=0.0
58     IF(FLAG2) LAMDA1=1.0
59     LAMDA2=1.0-LAMDA1
60     IVAL=50

```

```

61      IPRINT=TA
62      VCX=V0*COSSGAM*COSSIG
63      VUY=V0*COSSGAM*SINSIG
64      VUZ=V0*SINGAM
65      V1X=V1*COS(XU(11))*COS(XU(12))
66      V1Y=V1*COS(XU(11))*SIN(XU(12))
67      V1Z=V1*SIN(XU(11))
68      V2X=V2*COS(XU(17))*COS(XU(18))
69      V2Y=V2*COS(XU(17))*SIN(XU(18))
70      V2Z=V2*SIN(XU(17))
71      VRELX1=VCX-V1X
72      VRELY1=VUY-V1Y
73      VRELZ1=VUZ-V1Z
74      VRELT1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
75      VRELX2=VCX-V2X
76      VRELY2=VUY-V2Y
77      VRELZ2=VUZ-V2Z
78      VRELT2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
79      DELX1=XU(1)-X0(7)
80      DELY1=XU(2)-X0(8)
81      DELZ1=XU(3)-X0(9)
82      DELX2=XU(1)-X0(13)
83      DELY2=XU(2)-X0(14)
84      DELZ2=XU(3)-X0(15)
85 C
86      DUM1=AMIN1(DUM1,DMS1)
87      DUM2=AMIN1(DUM2,DMS2)
88      DMS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
89      DMS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
90      DSV1=0.5*DMS1/VRELT1
91      DSV2=0.5*DMS2/VRELT2
92      STEP=AMIN1(TSTEP,DSV1,DSV2)
93      DMS1=AMIN1(DMS1,DMS2)
94      IF(JJ.GT.0) MANUVR=1
95      LF=1.0
96      ULAST=U0
97      U0=0.0
98      IF(MANUVR.EQ.1) GO TO 300
99 C
100     U0=ULAST
101     LF=MAXLF
102     ISEC=ISEC+1
103     GX1 = -(SYDT1*SINSY1*COSTH1 + THEDT1*COSSY1*SINTH1)
104     GY1 = THEDT1*COSSY1*COSTH1 - SYDT1*SINSY1*SINTH1
105     GZ1 = SYDT1*COSSY1
106     NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
107     GX2 = -(SYDT2*SINSY2*COSTH2 + THEDT2*COSSY2*SINTH2)
108     GY2 = THEDT2*COSSY2*COSTH2 - SYDT2*SINSY2*SINTH2
109     GZ2 = SYDT2*COSSY2
110     NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
111     IF(NORM1.GE.TESTH.AND.NORM2.GE.TLSTN) GO TO 200
112 C
113 C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
114 C
115     LF=ACLF(ISEC-1)
116     GO TO 290
117 C
118     200 CONTINUE
119     JX1=GX1/NORM1
120     JY1=GY1/NORM1

```

```

121      GZ1=GZ1/NORM1
122      GX2=GX2/NORM2
123      GY2=GY2/NORM2
124      BZ2=GZ2/NORM2
125 C
126      COEFA1=GZ1*COSGAM-GY1*SINGAM*SINSIG-GX1*SINGAM*COSSIG
127      COEFA1=COEFA1*(LU+IU*SIN(ALFA0))
128      COEFB1=GY1*COSGAM-GX1*SINGAM
129      COEFL1=COEFA1*(LU+IU*SIN(ALFA0))
130      COEFL1=((TU+COS(ALFA0)-DU)*(GZ1*SINGAM+GY1*COSGAM*SINSIG
131      1 +GX1*COSGAM*COSSIG)) - GZ1*ND*G
132 C
133      COEFA2=GZ2*COSGAM-GY2*SINGAM*SINSIG-GX2*SINGAM*COSSIG
134      COEFA2=COEFA2*(LU+IU*SIN(ALFA0))
135      COEFB2=GY2*COSGAM-GX2*SINGAM
136      COEFL2=COEFA2*(LU+IU*SIN(ALFA0))
137      COEFL2=((TU+COS(ALFA0)-DU)*(GZ2*SINGAM+GY2*COSGAM*SINSIG
138      2 +GX2*COSGAM*COSSIG)) - GZ2*ND*G
139 C
140 C      CHECK EVERY ANGLE (BY DEGREE) WITHIN +/- RLMAX
141 C
142      NBLK=1
143      PRFMAX=0.0
144      IU0=(IU*180./PI)+100.
145      IF(IU0.GT.360) IU0=IU0-360
146      IF(IU0.LT.1) IU0=IU0+360
147      IROLL=RLMAX*STEP
148      LOW(1)=IU0-IROLL
149      IF(LOW(1).GE.1) GO TO 210
150      NBLK=2
151      LOW(1)=1
152      LOW(2)=360+IU0-IROLL
153      HIGH(2)=360
154 210 CONTINUE
155      HIGH(1)=IU0+IROLL
156      IF(HIGH(1).LE.360) GO TO 220
157      NBLK=2
158      HIGH(1)=IU0+IROLL-360
159      HIGH(2)=360
160      LOW(2)=LOW(1)
161      LOW(1)=1
162 220 CONTINUE
163      DO 230 IJK=1,NBLK
164      ISTRT=LOW(IJK)
165      IEND=HIGH(IJK)
166      DO 230 IU0=ISTRT,IEND
167      PERF1=COEFA1*TABCOS(IU0)+COEFB1*TABSIN(IU0)+COEFL1
168      PERF2=COEFA2*TABCOS(IU0)+COEFB2*TABSIN(IU0)+COEFL2
169      ABS1=ABS(PERF1)
170      ABS2=ABS(PERF2)
171      IF(FLAG1) ABS1=ABS2+2.0
172      IF(FLAG2) ABS2=ABS1+2.0
173      PRFMIN=AMIN1(ABS1,ABS2)
174      IF(PRFMIN.LE.PRFMAX) GO TO 230
175      IFLAG=IVAL
176      I*AX=IU0
177      PRFMAX=PRFMIN
178      IF(ABS1.LE.ABS2) GO TO 230
179      IFLAG=-IVAL
180 230 CONTINUE

```

```

151      UD=FLOAT(IMAX-15J)*PI/180.
152 C
153 C      NOTE --- IFLAG IS POSITIVE FOR LAMBDA = 1, AND NEGATIVE
154 C            FOR LAMBDA = 0
155 C
156      PL(ISEC,2)=PERF1
157      PL(ISEC,4)=PERF2
158      PL(ISEC,26)=IFLAG
159      PMAX=AMAX1(PMAX,PERF1,PERF2)
160      PMIN=AMIN1(PMIN,PERF1,PERF2)
161 C
162 C      CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
163 C
164      ACCX=((TO+COS(ALFAU)-DO)*(COSGAM*COSSIG))-
165      1  ((LU+TO*SIN(ALFAU))*(SIN(UD)*SINGAM+COS(UD)*
166      2  SINGAM*COSSIG))
167      ACCY=((TO+COS(ALFAU)-DO)*(COSGAM*SINSIG))+
168      1  ((LU+TO*SIN(ALFAU))*(SIN(UD)*COSGAM-COS(UD)*
169      2  SINGAM*SINSIG))
170      ACCZ=((TO+COS(ALFAU)-DO)*SINGAM)-(MO*G)+
171      1  ((LU+TO*SIN(ALFAU))*COSGAM*COS(UD))
172      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
173 C
174 C      STORE PLOTTED VARIABLES
175 C
176      PL(ISEC,14)=NORM1
177      PL(ISEC,19)=ANORM/MO
178      PL(ISEC,30)=GX1
179      PL(ISEC,31)=GY1
180      PL(ISEC,32)=GZ1
181      PL(ISEC,33)=NORM2
182      PL(ISEC,34)=GX2
183      PL(ISEC,35)=GY2
184      PL(ISEC,36)=GZ2
185      NORMAX=AMAX1(NORMAX,NORM1,NORM2)
186      ANMAX=AMAX1(ANMAX,ANORM)
187 C
188 C      290 CONTINUE
189 C      IF(PRTED) GO TO 300
190 C      PRTEDE=TRUE.
191 C      CCI=CCI+TAFTB*IFLAG
192 C      ICHG=TPRINT
193 C      NPIT=(2,310) TPRINT
194 C      GPIT=(0,310) TPRINT
195 C      310 FORMAT(2X,"START MANEUVER AT T = ",F6.2)
196 C      CALL GELL(3)
197 C
198 C      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
199 C      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
200 C      ALFMAX=ALFAU
201 C      ALFMIN=ALFAU
202 C      ZMAX=AMIN1(XU(3),XU(9),XU(15))
203 C      ZMIN=AMAX1(XU(3),XU(9),XU(15))
204 C      XMAX=AMAX1(XU(1),XU(7),XU(13))
205 C      XMIN=AMIN1(XU(1),XU(7),XU(13))
206 C      YMAX=AMAX1(XU(2),XU(6),XU(14))
207 C      YMIN=AMIN1(XU(2),XU(6),XU(14))
208 C      VMAX=VU
209 C      VMIN=VU
210 C      UMAX=UO

```

```

241      DMIN=00
242      LOSMAX=0.0
243 C
244      300 CONTINUE
245      ACJ(ISEC)=UG*150./PI
246      ACLF(ISEC)=LF
247      ACII(ISEC)=TPRINT
248      ACDY(ISEC,1)=DMIS1
249      ACDM(ISEC,2)=DMIS2
250 C
251      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
252      WRITE (2,400) TPRINT, DMIS1, DMIS2
253      WRITE (3,400) TPRINT, DMIS1, DMIS2
254      400 FORMAT (JX,"TIME = ",F10.3,2X,"DSEP1 = ",G12.3,2X,
255      1 "DSEP2 = ",G12.3)
256 C
257      477 CONTINUE
258      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
259      IF(DMIS.LT.DFH) GO TO 520
260      IF(TA.LT.3.0) GO TO 480
261      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
262      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
263      IF(FLAG1.AND.FLAG2) GO TO 540
264      480 CONTINUE
265      CALL INTBOX
266      TA=TA+STEP
267      JJ=JJ+1
268      IF(ISEC.GE.10) GO TO 530
269      GO TO 100
270 C
271      510 CONTINUE
272      WRITE (2,515) TPRINT
273      WRITE (3,515) TPRINT
274      515 FORMAT(TA,"** A/C OR MISSILE VEL. IS TOO LOW ",
275      1 "AT TIME: ",F10.3," **",/)
276      GO TO 500
277 C
278 C      HIT OCCURRED, PRINT OUT
279      520 CONTINUE
280      WRITE(2,525) TPRINT,DUM1,DUM2
281      WRITE(3,525) TPRINT,DUM1,DUM2
282      525 FORMAT(1X,"***** HIT AT TIME = ",F10.3,
283      1 "*****",/,5X,"BEST DSEPS WERE 1:",
284      2 " G15.6,", " 2:",G15.6,/)
285      GO TO 600
286 C
287      530 CONTINUE
288      WRITE(2,535) TPRINT
289      WRITE(3,535) TPRINT
290      535 FORMAT(5X,"TIME LIMIT AT T = ",F6.2)
291      GO TO 600
292 C
293 C      CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
294      540 CONTINUE
295      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
296      WRITE(3,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
297      544 FORMAT(3X,"***** CLOSURE RATE NEGATIVE AT TIME = ",F10.3,
298      1 "*****",/,5X,"TA1 : BEST DSEP = ",G15.6," , NOW = ",G15.6,/,
299      2 " 5X,"TA2 : BEST DSEP = ",G15.6," , NOW = ",G15.6,/)
300      GO TO 600

```

```

301 C
302 600 CONTINUE
303 CALL BELL(1)
304 C
305 READ(1,635) LOGIC
306 IF(LOGIC.EQ.1EXIT) RETURN
307 N10=0
308 DO 620 J=1,N10
309 JJJ=J+6
310 KK=J+12
311 TEMP=YU(J)
312 TEMP1=XO(JJJ)
313 TEMP2=XO(KK)
314 IF(J.LT.5) GO TO 605
315 TEMP=TEMP*160./PI
316 TEMP1=TEMP1*160./PI
317 TEMP2=TEMP2*160./PI
318 605 CONTINUE
319 WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
320 WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
321 610 FORMAT(2X,"XJ(",I1,") = ",G13.4,4X,
322 1 "XO(",I2,") = ",G13.4,4X,
323 2 "XO(",I2,") = ",G13.4)
324 620 CONTINUE
325 C
326 WRITE(2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
327 WRITE(6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
328 625 FORMAT(/,10X,"DELX1: ",G12.3,2X,"DELY1: ",G12.3,/,
329 1 10X,"DELZ1: ",G12.3,2X,"DMIS1: ",G12.3,/,
330 2 10X,"BEST DMIS WAS ",G12.3)
331 WRITE(2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
332 WRITE(6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
333 630 FORMAT(/,10X,"DELX2: ",G12.3,2X,"DELY2: ",G12.3,/,
334 1 10X,"DELZ2: ",G12.3,2X,"DMIS2: ",G12.3,/,
335 2 10X,"BEST DMIS WAS ",G12.3)
336 CALL BELL(1)
337 READ(1,635) LOGIC
338 IF(LOGIC.EQ.1EXIT) RETURN
339 IF(LOGIC.EQ.1QUIK) GO TO 660
340 635 FORMAT(A1)
341 C
342 DO 650 I=1,ISEC
343 IF(MOD(I,20).EQ.0) CALL BELL(1)
344 IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
345 IF(LOGIC.EQ.1EXIT) RETURN
346 IF(LOGIC.EQ.1QUIK) GO TO 660
347 WRITE(2,636) ACTI(I),ACUA(I),PL(I,6),PL(I,38)
348 636 FORMAT(1X,"TIME ",F5.2,3X,"ACU = ",F3.2,3X,"LFIC ",F8.1,
349 1 3X,"LF2C ",F8.1)
350 650 CONTINUE
351 660 CONTINUE
352 WRITE(6,641)
353 641 FORMAT(1H1)
354 DO 670 I=1,ISEC
355 WRITE(6,637) ACTI(I),ACLF(I),ACRA(I),PL(I,6),PL(I,38),
356 1 PL(I,17),PL(I,14),PL(I,11),PL(I,20),
357 2 (ACU(I,J),J=1,4),PL(I,47)
358 637 FORMAT(1X,"TIME = ",F6.2,3X,"ACLF = ",F5.1,3X,
359 1 "ACRA(DEG) = ",F8.2,3X,"LFIC = ",F8.1,3X,
360 2 "LF2C = ",F5.2,3X,"NORM(A) = ",G11.2,/,20X,

```

```

361      3  NORM1(U) = ,G11.2,3X,ALPHA = ,F5.2,3X,
362      4  UPA5 = ,G11.2,/,20X,UMIS1 = TA(1) = ,
363      5  F9.2,3X,UMIS2 = TA(2) = ,F9.2,5X,
364      6  LAMBDA1 = ,ZPF5.2,/)
365  670 CONTINUE
366 C
367      IDUM=CUTIME(J)-ICPUTM
368      WRITE(2,639) IDUM
369      WRITE(6,639) IDUM
370  639 FORMAT(4X,CPU TIME (SECONDS) = ,I5,0PF5.0)
371 C
372      RETURN
373      END

```