

NASA Contractor Report 4160

Rule-Based Air Combat Simulation

G. H. Burgin and L. B. Sidor
Titan Systems, Inc.
La Jolla, California

Prepared for
Ames Research Center
Dryden Flight Research Facility
under Contract NAS2-11990



National Aeronautics
and Space Administration

Scientific and Technical
Information Division

1988

FOREWORD

While Titan was under Contract to the NASA Dryden Flight Research Center (Contract NAS2-11990), Titan was also under contract to the NASA Ames Research Center, Moffett Field (Contract NAS2-11824). Both contracts had as ultimate goals to improve the Adaptive Maneuvering Logic Air-to-Air Combat Computer program.

The emphasis in the Moffett Field program was to improve the guidance laws, regardless of required execution time on a computer. In contrast, the Dryden effort was to provide a robust decision logic, guaranteed to work in real time. The logic developed for Dryden should eventually drive an actual aircraft in real flight.

During the course of this work, it would have been unproductive to keep track of which of the AML improvements should be credited to the Moffett Field contract and which ones to the Dryden contract. This final report on contract NAS2-11990 is therefore essentially the same as the final report on contract NAS2-11824 (Simulation of Modern Air-to-Air Combat). The present report has some material added in section 3 and a substantially enlarged section 5. It also contains an Appendix with the Fortran listing of the subroutines implementing the "Basic Fighter Maneuvers".

CONTENTS

Section		Page
	SUMMARY	vii
1	OVERVIEW OF AIR COMBAT SIMULATIONS AND METHODOLOGY	1-1
	1.1 Deterministic and Stochastic Models	1-2
	1.2 Offline and Real-Time Simulations	1-5
	1.3 Weapons Models	1-7
	1.4 Aircraft Models	1-11
	1.5 Decision-Making Process	1-12
	1.6 Performance Measures	1-16
2	DEVELOPMENT HISTORY OF THE AML PROGRAMS	2-1
	2.1 NASA Langley DMS Program (AML 75)	2-1
	2.2 The DMS Control Model (AML 76)	2-1
	2.3 A One-Versus-Two Version of AML (AML 1V2)	2-3
	2.4 AML with Revised Equations of Motion (AML 84)	2-4
	2.5 Northrop Aerospace Laboratory AML (Interactive Target)	2-5
	2.6 General Electric's AML version on the Simulator for Advanced Airmanship (AML 86)	2-6
	2.7 AML 87/Expert Ivan	2-6
	2.8 Missile Evasion AML (AML/SAM)	2-7
	2.9 Pilot's Associate D1 AML Program	2-7
3	THE BASIC AML IF => THEN LOGIC	3-1
	3.1 Terminology	3-1
	3.2 Commonality with the Trial Maneuver Logic	3-1
	3.2.1 Timing Considerations	3-1

CONTENTS

Section		Page
	3.2.2 Maneuver Plane Concept	3-2
	3.3 Decision Hierarchy	3-4
	3.3.1 Ground Avoidance	3-4
	3.3.2 The Pointing Algorithm	3-4
	3.3.3 The Lead/Lag Maneuver Logic	3-5
4	AIRCRAFT AND CONTROL SYSTEMS DYNAMICS	4-1
	4.1 Selecting an Appropriate Model	4-1
	4.2 Synopsis of the Current Attitude Control Mechanism	4-3
	4.3 Refined Calculation of Commanded Pitch Rate	4-5
	4.4 The Pointing Control System	4-6
5	SAMPLE TRIAL MANEUVERS AML VERSUS IF => THEN AML RUNS	5-1
	5.1 Region 2	5-4
	5.2 Region 3	5-8
	5.3 Region 4	5-11
	5.4 Region 5	5-23
	5.5 Region 6	5-34
	5.6 Conclusion	5-34
6	ADDING BASIC FIGHTER MANEUVERS TO THE IF => THEN LOGIC	6-1
	6.1 Opposite Turn	6-5
	6.2 Diving Overshoot	6-11
	6.3 Pull-Up Overshoot	6-17
7	CONCLUSIONS AND RECOMMENDATIONS	7-1
8	REFERENCES	8-1

FIGURES

1-1	Level of Detail in Simulations	1-3
1-2	Gun Vulnerability Cone	1-8
1-3	Typical Missile Envelopes for Maneuvering and Non-Maneuvering Targets	1-10
1-4	Definition of Deviaton Angle and Angle-off	1-18
1-5	Ambiguities in the Line-of-Sight/Angle-off Representation	1-21
2-1	Development History of the AML Air-to-Air and Missile-Evasion Programs	2-2
3-1	Illustration of Maneuver-Plane Concept	3-3
3-2	Regions for Steering Laws	3-6
3-3	Summary of the AMLB Control Laws	3-7
4-1	Block Diagram of Pointing Control System	4-8
5-1	Definition of Regions	5-3
5-2	Region 2, X-Y Trace	5-5
5-3	Region 2, PI Plot	5-6
5-4	Region 3, X-Y Trace	5-9
5-5	Region 3, PI Plot	5-10
5-6	Forward Quarter Turn Options	5-12
5-7	Forward Quarter Pass, Case 1, X-Y Trace	5-13
5-8	Region 4, PI Plot	5-14
5-9	Forward Quarter Pass, Case 2, X-Y Trace	5-16
5-10	Case 2, Region 4, PI Plot	5-17
5-11	Forward Quarter Pass, Case 3, X-Y Trace	5-19
5-12	Case 3, PI Plot	5-20
5-13	Forward Quarter Pass, Case 4, X-Y Trace	5-21
5-14	Case 4, PI Plot	5-22

5-15	Region 5, Case 1, X-Y Trace	5-24
5-16	Region 5, Case 1, PI Plot	5-25
5-17	Region 5, Case 2, X-Y Trace	5-26
5-18	Region 5, Case 2, PI Plot	5-27
5-19	Region 5, Case 3, X-Y Trace	5-28
5-20	Region 5, Case 3, PI Plot	5-30
5-21	Region 5, Case 3, Line-of-Sight and Angle-off Plot	5-31
5-22	Region 5, Case 4, X-Y Trace	5-32
5-23	Region 5, Case 4, PI Plot	5-33
5-24	Region 6, X-Y Trace	5-35
5-25	Region 6, PI Plot	5-36
6-1	Outline of the Implementation of Basic Fighter Maneuvers	6-3
6-2	Opposite Turn Case 1	6-7
6-3	Opposite Turn, Case 1, PI Plot	6-8
6-4	Opposite Turn, Case 2, X-Y Trace	6-9
6-5	Opposite Turn, Case 2, PI Plot	6-10
6-6	Diving Overshoot, X-Y Trace	6-13
6-7	Diving Overshoot, PI Plot	6-14
6-8	Lead/Lag Overshoot, X-Y Trace	6-15
6-9	Lead/Lag Overshoot, PI Plot	6-16
6-10	Pull-Up Overshoot, Case 1, X-Y Trace	6-18
6-11	Pull-Up Overshoot, Case 1, PI Plot	6-19

TABLES

1-1	Complexity in Air Combat Simulations	1-6
1-2	Representative Values of Deviation Angle and Angle-Off	1-22
5-1	Initial Values for Sample Runs	5-1

SIMULATION OF MODERN AIR COMBAT

By George H. Burgin and Laurent B. Sidor

Titan Systems, Inc., La Jolla Ca.

SUMMARY

This final report on Contract NAS2-11824 is organized in seven sections plus a list of 25 references.

Section 1 provides an overview of current topics in the simulation of air-to-air combat, touching on such subjects as weapons simulation, aircraft modeling and performance measurement.

In section 2, the history of a set of computer programs, developed over the last 15 years is traced. These programs are generally known as "Adaptive Maneuvering Logic" (AML) programs. They exist in many versions: Air-to-air combat and missile evasion, real-time and non-real time versions.

The air-to-air combat simulation exists in two basically different versions: The older version, the "trial-maneuver" version, is described in other NASA reports. The newer version, the "IF => THEN " version, is the subject of section 3 of this report.

Section 4 summarizes some important aspects of aircraft dynamics modeling. The interrelationship between the tactical

decision process and the aircraft model is shown. For example, tactical performance can be significantly improved if the modelled aircraft can be controlled by a control system capable of orienting the aircraft's longitudinal axis into a desired direction (pointing control system).

Section 5 compares the performance of the trial maneuver logic with the IF => THEN logic and demonstrates how each logic may be improved by "playing" it against the other logic.

To make the performance of the IF => THEN logic less predictable, some basic fighter maneuvers were added to AML which are invoked, when appropriate, under the control of pseudo-random numbers. These maneuvers are described in section 6.

Finally, section 7 provides some suggestions for continued work in developing advanced guidance law for air-to-air combat.

1. OVERVIEW OF AIR COMBAT SIMULATIONS AND METHODOLOGIES

Simulation of air-to-air combat has become an indispensable tool for pilot training, for tactics development, for weapons systems evaluation and for a host of other applications. Air combat becomes more and more complex due to advances in electronic warfare. Air-to-air combat today begins a long time before the opponents have visual contact. Radar and other sensors provide critical input to the pilot at a range far beyond the visual range. It has therefore become common practice in the analysis and in the simulation of air-to-air combat to differentiate between a "Beyond Visual Range (BVR)" phase and a "Close In Combat (CIC)" phase.

The present report is concerned primarily with simulating the CIC environment. Specifically, we will discuss in detail a series of computer programs generally known as "Adaptive Maneuvering Logic Program (AML)". These models and simulations were developed under NASA sponsorship with the initial goal to have an intelligently interactive, real time opponenet on NASA's differential maneuvering simulator (DMS) at the Langley Research Center and with the long-range goal to provide assistance to a pilot during air combat engagements.

A measure of the complexity of modern air war may be obtained by reading the account of Israeli air operations over Lebanon in 1982 (Reference 2). These operations involved air superiority fighters in strike escort and combat air patrol roles operating in concert with many other elements such as

SAMs, AWACS, ground-based radars and communication centers, stand-off jammers, and RPVs. Similarly complex operations are involved in the air defense of U.S. carrier task forces (see for example Reference 3).

Due to the complexity of such air operations, individual air simulations must focus on a particular, limited area of air combat. We will briefly review the current state-of-the-art in air combat simulations in order to put a perspective on the area considered by AML. Some of the key issues addressed by this report will be:

- Number of aircraft involved in the simulation
- Types and properties of weapons employed by the combatants
- Degree of complexity of aircraft and weapons models
- How random effects are simulated
- Off-line simulation versus real-time simulation

1.1 DETERMINISTIC AND STOCHASTIC MODELS

The common point of departure for air combat simulations are various scenarios of Air Force and Navy Missions. In the final analysis, their common evaluation point relies on pilot opinion. In the design phase, a basic trade-off must be made between the accuracy in modelling individual elements and the size and execution time of the code. Figure 1.1 attempts to portray this trade-off. Engineering simulations which model in detail physical mechanisms (such as warhead fuzing) are limited to one or two units. At the other end of the spectrum are

LEVEL OF DETAIL IN SIMULATIONS

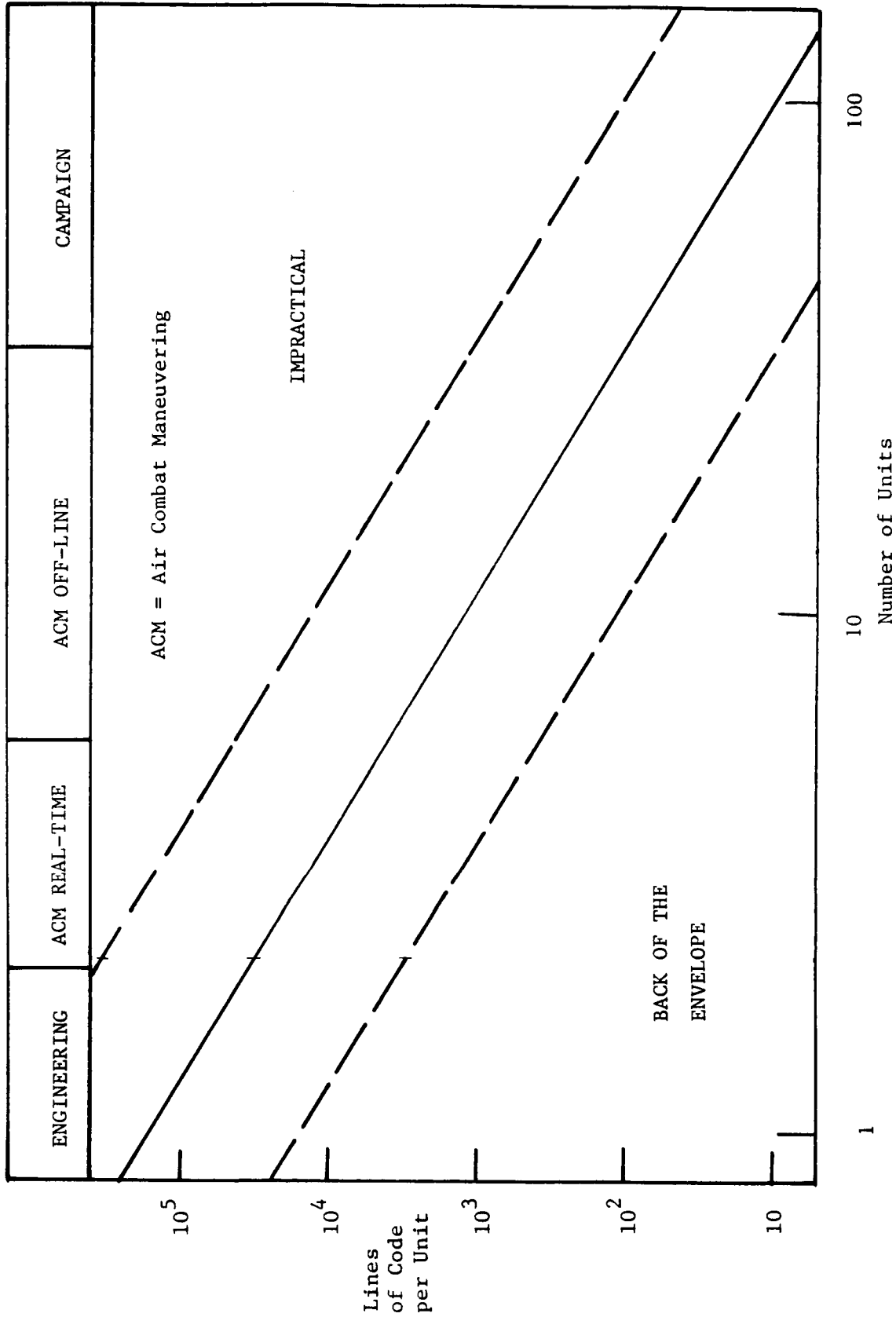


Figure 1-1. Level of Detail in Simulations
(Adapted from Reference 1)

campaign or force-on-force models with hundreds of simulated units. In these models, the representation of physical mechanisms in the simulation is done in terms of aggregated performance measures. The simulation of even a minimally representative number of opponents in the case of NATO vs Warsaw Pact scenarios (2 vs 4) leads to an explosion in the computational requirements.

The performance of many aspects of weapons systems is expressed in terms of probabilistic quantities, for example radar probability of detection or kill probability of a missile warhead against a target type. The combination of these probabilities can be performed in one of two ways: (1) Expected value method and (2) Monte-Carlo method. In the expected value method, the probabilities are combined using the law of probabilities for the particular probability law obeyed by the simulated process. For example, if there are N independent interceptors, each with a probability PD of detecting a single bomber over a period of time, then it may be shown that the expected fraction of bombers FDB detected at the end of the period of time will be :

$$FDB = 1. - \exp(N *PD/ M),$$
 where M is the total number of bombers. (Reference 15)

In contrast, in the Monte-Carlo method, the outcome of a probabilistic event is assessed based on the draw of a random number. For this reason, these are described as "discrete events". Repetitive trials must be performed to obtain averages,

a process which multiplies computational requirements. In addition, in Monte-Carlo simulations, the sheer volume of information makes it difficult to trace causative factors. For these practical reasons, Monte-Carlo models are popular up to the mid-range of Figure 1-1; for campaign models, only expected-value models are practical.

1.2 OFFLINE AND REAL-TIME SIMULATIONS

The simulations discussed above consist of "off-line" or "non-real time" simulations. Even these non-realtime simulations may require execution times which limit their economical use for studies and analyses.

In a real-time simulation, two tasks have to be performed. First, the equations of motion for each participant must be solved satisfying the condition that the CPU-time to perform the calculations for one integration step will not exceed the allocated frame time. The second task requires the simulation of the decision process for each platform. The required CPU time to perform this second task also must fit into the allocated frame time. Typical frame times for real-time close-in air-to-air combat simulations with a human pilot in the loop are between 10 and 50 milliseconds.

Table 1-1 illustrates the parameters involved at both ends of the spectrum in complexity in air combat simulations. The AML programs feature a high complexity simulation environment, moderate complexity in aircraft performance and tactics

<----- LOW COMPLEXITY ----->-----> HIGH COMPLEXITY ----->

AIRCRAFT

Point mass
Instantaneous response

AIRCRAFT

6 DOF model
Non-linear aerodynamics
Non-linear flight controls

WEAPONS

Guns: Pk, Vulnerability cone

Missiles: Pk, Launch envelopes
Time-of-flight

Point mass
Idealized guidance laws

WEAPONS

Guns, Missiles
6 DOF fly-out
Miss distance
Pk for A/C vulnerable areas

Detailed seeker, guidance and control
and propulsion models
Effects of jamming and/or IRCM

AVIONICS

Radar,IRST , visual detection ranges

AVIONICS

Radar,IRST detection probabilities
Target glint
Countermeasures

TACTICS

1-on-1
Pre-programmed

Generic production
rules and
guidance laws

TACTICS

Many-on-many; section tactics
Production rules and tactics
optimized for platform/weapon/target
Trial maneuvers
Command and control
Integration with SAMs

ENVIRONMENT

Offline simulation

ENVIRONMENT

Real time flight simulator
Terrain
Atmospheric visibility

020287:tablams.ptI

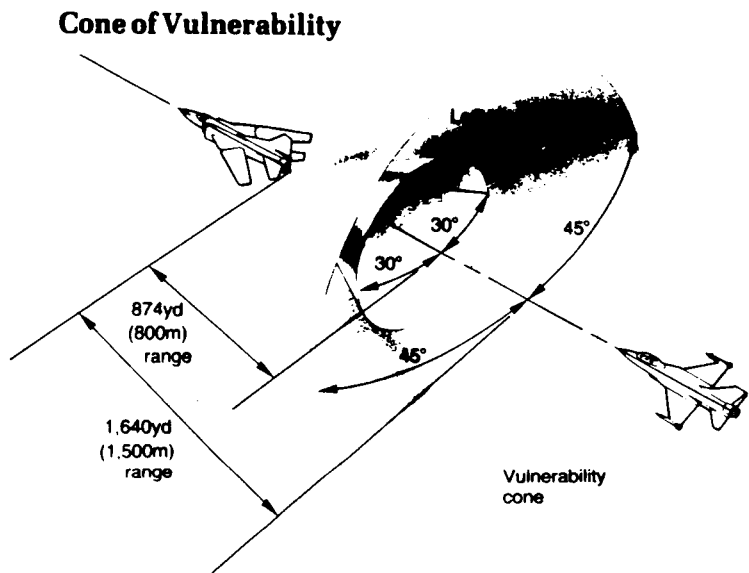
representation, and low complexity in weapons and avionics.

1.3 WEAPONS MODELS

The armament considered in air combat simulations consists of guns, guided missiles and, recently, lasers. Because of the research nature of these simulations, a significant amount of effort has been spent on simulating air-to-air lasers, while this weapon has yet to see operational use. They will not be discussed further here.

The basic requirement to achieve a gun firing position is to point the nose of the aircraft at the target. Steerable guns would alter this requirement, but presently there are none operational on fighters anywhere in the world. In general, to achieve a kill will require several hits on the target. For this reason, an off-tail position is preferable (Figure 1-2). In AML, the conditions to achieve a gun firing position are a line-of-sight angle less than 10 degrees and an angle-off tail less than 60 degrees and a range less than 3000 feet. (These quantities are defined in the paragraph "Performance measures" below.) Some models provide the option of integrating the trajectory of an individual bullet. The point of impact is calculated so that the effect of the hit can be accurately estimated using a vulnerable area approach.

The requirements that must be satisfied for a missile launch are customarily summarized in terms of a "firing envelope" or "launch-acceptable region (LAR)". A representative



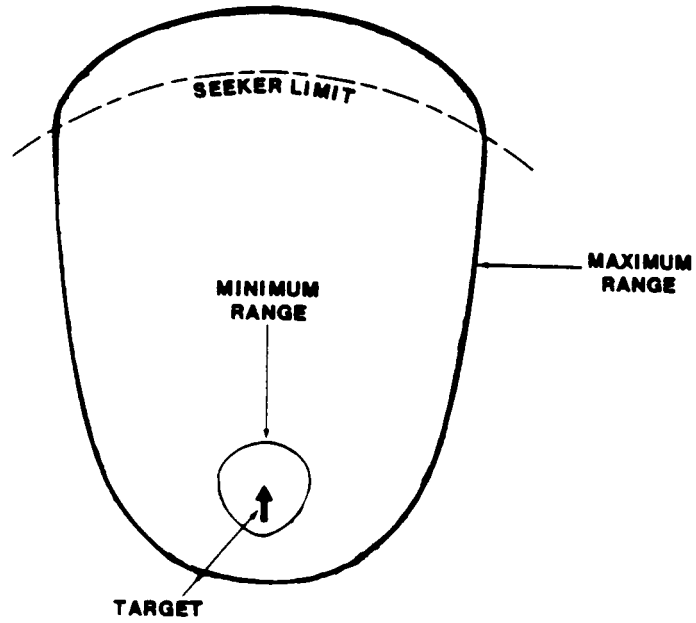
Above: Most successful gun attacks are made from astern at 30° angle-off or less, the lethal cone. Vulnerability cone is 45° angle-off and 1,640yd (1,500m) range.

Figure 1-2: Gun Vulnerability Cone (from Reference 24)

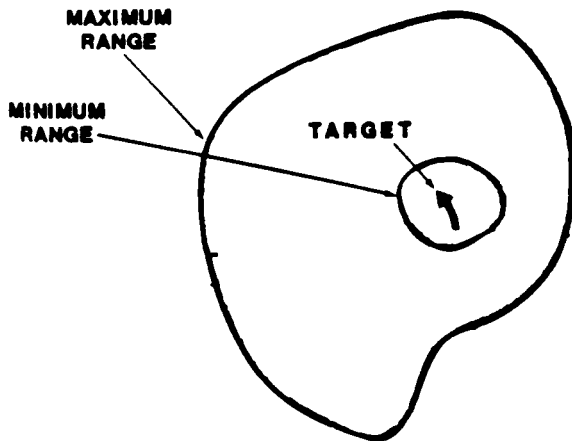
envelope, with the target at the center, is shown in Figure 1-3 for a typical radar doppler-homing missile (Reference 10). These envelopes are used by most modern air combat models. They have also been used in various studies using AML (Reference 13).

For non-maneuvering targets, an envelope has a maximum range with a roughly elliptical shape which reflects the aeropropulsive limit of a typical missile. The maximum range varies strongly as a function of altitude and target speed. Figure 1-3 b also illustrates the seeker limit, which in the illustrated case is smaller than the maximum range of the missile. The seeker limit is dependent on the target's radar reflectivity characteristics (a function of the target aspect as seen from the firing position.) Figure 1-3 also indicates an inner zone (minimum range or "dead zone"). It will be noted that the head-on maximum range is much greater than the off-tail range -- typically four to five times. The greater area means that there are more engagement opportunities in the forward target quarter. But it should also be remembered that the target's sensors are effective only in its forward quarter.

The "maneuvering envelope", as illustrated in figure 1-3b, represents the effect on the intercept capability of the missile when the target begins a level left-hand turn just as the missile is launched. The envelope typically assumes a shape that is distorted in the direction of the turn. The maximum range expands in the direction of the turn, as the target is flying towards the missile. It contracts in the direction opposite the



a) Non-maneuvering Target



b) Maneuvering Target

Figure 1-3: Typical Missile Envelopes for Maneuvering and Non-maneuvering Targets (from Reference 10)

turn as the target is flying away from the missile. These effects are proportional to the number of G's pulled by the target. In spite of the magnitude of the effect due to maneuvering on the shape of the envelope, none of the models listed in the references appears to modify the decision to fire based on this effect.

After the missile has been fired, the damage to the target must be assessed. Detailed simulations simulate the fly-out trajectory to the target, compute the miss distance and resulting survivability of the target. Less detailed simulations simplify this problem by computing a time-of-flight and survivability of the target based on a probability of kill and Monte Carlo draw.

1.4 AIRCRAFT MODELS

The simplest aircraft model used in air combat simulations consists of a point mass to which are applied the lift, thrust and drag forces. This provides a starting point, for instance to compute the endurance of an aircraft in the simulated engagement. This type of aircraft model is limited to "instantaneous turns", and cannot represent the attitude and turn capability of a fighter. Yet this limitation is often not recognized until realistic graphics are available, or the simulation is run in a flight simulator.

A full 6 degrees of freedom (6 DOF) model is required to simulate realistically the behavior of a fighter aircraft under

the high-Gs and very large angles of attack encountered in air combat. All the lift, drag and thrust characteristics as well as moments should be represented. In particular, roll performance is of primary importance in fighter aircraft tactics and would alone justify the use of a 6-DOF model (see for example Reference 18.)

The model currently used in AML is described in Reference 14. It is a "performance model", in which 6-DOF dynamics have been preserved, but in which the calculation of aerodynamic moments and control and stability derivatives has been omitted to meet execution time requirements on minicomputer-based flight simulators.

1.5 DECISION-MAKING PROCESS

The objective of the decision-making process is to derive maneuvers which will bring one's own weapons to bear on the target while at the same time minimizing exposure to the other side's weapons. It is essentially a representation of the action of the pilot during combat. In simulations involving multiple aircraft, the decision-making process also involves pairing groups of opponents.

The real-life approach to the solution of the problem of steering an individual aircraft relative to an opposing, dissimilar aircraft is known as the "Basic Fighter Maneuvers" (BFMs). Examples of training manuals describing BFMs for particular aircraft may be found in References 11 and 12.

Reference 10 is a more general treatment of this field. The objective of BFMs is twofold: (1) Gain and maintain a positional advantage with respect to the enemy allowing employment of armament, and (2) Gain and maintain sufficient energy to have maneuvering potential. BFMs are not exact maneuvers, but rather combinations of the three elementary actions that an aircraft is capable of -- roll, turn, and accelerate/decelerate -- used to gain advantage in a particular situation and against a particular opposing aircraft type. Well-known examples of BFMs are: the Immelman, the lead/lag turn, the Lufbery, the high-speed yo-yo.

In spite of the admittedly inexact nature of BFMs, they nevertheless constitute a sourcebook of possible maneuvers which has been used as the basis for the decision logic of models such as PACAM (Reference 8), AASPEM (Reference 7) and TACBRAWLER (Reference 9). As an example of this approach, a partial list of such maneuvers available in the AASPEM model includes: chandelle, split-S, high-speed yo-yo, barrel roll. The decision logic for selecting a maneuver is based for the most part on user-specified geometry rules. There is an amount of guesswork involved in specifying these maneuvers. For example, the user must insure that the energy state of each aircraft is sufficient to complete the specified maneuver. Otherwise, unrealistic and unacceptable maneuvers may result.

The specification of these maneuvers depends on the current phase of the engagement. For example, AASPEM considers seven

phases:

- Neutral: no threat detected
- Late set-up: setting-up phase near completion
- Early set-up: setting-up phase near completion
- Pre-attack: final set-up and preparation for attack
- Attack: attacking threat
- Post-attack: initial attack complete
- Disengage: engagement complete

For each of these phases, AASPEM requires specifying positional tactics, information-gathering tactics and information-denial tactics.

This approach suffers from the disadvantage which was noted in the original AML report (Reference 5) , and is echoed in some training manuals (Reference 11) that fighter pilots learn these basic fighter maneuvers in training, but they rarely complete them in a dogfight because of the continuous interaction and changes in the relative situation.

Another type of approach consists of programs which apply such disciplines as optimal control theory, and the theory of differential games to obtain control laws. Such approaches work best for idealized situations (e.g. co-altitude, analytic lift curves, etc...).

The trial-maneuver approach was introduced by AML to remedy the problems with these approaches. The AML technique determines the next tactical maneuver as it contributes to the goals of the pilot. It uses the concept of a situation matrix describing the tactical decision options in terms of various values assigned to each cell. The maneuver selected is the one which maximizes this value (References 5 and 6).

The trial-maneuver approach, originally published by Burgin et al (References 5 and 6) proved to be quite successful for a real-time decision logic. It does require, however, considerable computer resources. Pedotti and Hignard (Reference 22) plagiarized the above mentioned work. They used almost an identical set of trial maneuvers and had a real time version of their " Logique Adaptive de Manoeuvre Aerienne" running on an UNIVAC 1100/82 mainframe computer.

Austin et al (Reference 23) used a very similar trial maneuver technique in the simulation of air-to-air combat between two helicopters. This program is operational in real time on the NASA AMES Vertical Motion Simulator.

The trial-maneuver approach -- as the name implies -- involves searching over a series of flight paths. The computational requirements were found to exceed the capacity of VAX 11/780-class mini-computers for real-time applications. To remedy this situation, a different approach was devised: the rule-based AML (RB/AML). The rule-based AML uses a combination of production rules (i.e. IF ... THEN statements) and guidance laws as an alternative to the trial maneuvers. These rules will be discussed in greater detail in Sections 3.

In m-on-n simulations, the decision process must in addition pair various groups of opposing aircraft. The doctrines found in the tactics manuals are the welded-wing, free-engaged, and the double attack system. These tactics have been emulated in air combat models such as PACAM and AASPEM. In the welded-

wing doctrine, the wingman attempts to maintain a loose formation with his leader. He does not make independent maneuver decisions, but nevertheless he fires his weapons on his own initiative when such opportunities arise. In the doctrine of free-engaged tactics, the two fighters exchange the roles of leader and wingman as the tactical situation requires.

1.6 PERFORMANCE MEASURES

Three levels of performance measures can be found:

(1) Individual aircraft performance, e.g. turn rate or energy maneuverability as a function of Mach number and altitude. The ability to change state is a recently introduced performance measure in this category (Reference 18.)

(2) Differential aircraft performance measure, e.g. the difference in turn rate. These are commonly used in training manuals. The implicit assumption is that both opponents enter the combat arena under the same initial conditions.

(3) Tactical performance measures, which are made possible only through air combat simulations of the type analyzed in this report.

The relative position of two opposing aircraft, "A" and "B" is conventionally described in terms of the deviation angle λ and angle-off ϵ . These have been illustrated, from the point of view of "B", in Figure 1-4, where they are indicated as $\lambda(B)$ and $\epsilon(B)$. The deviation angle

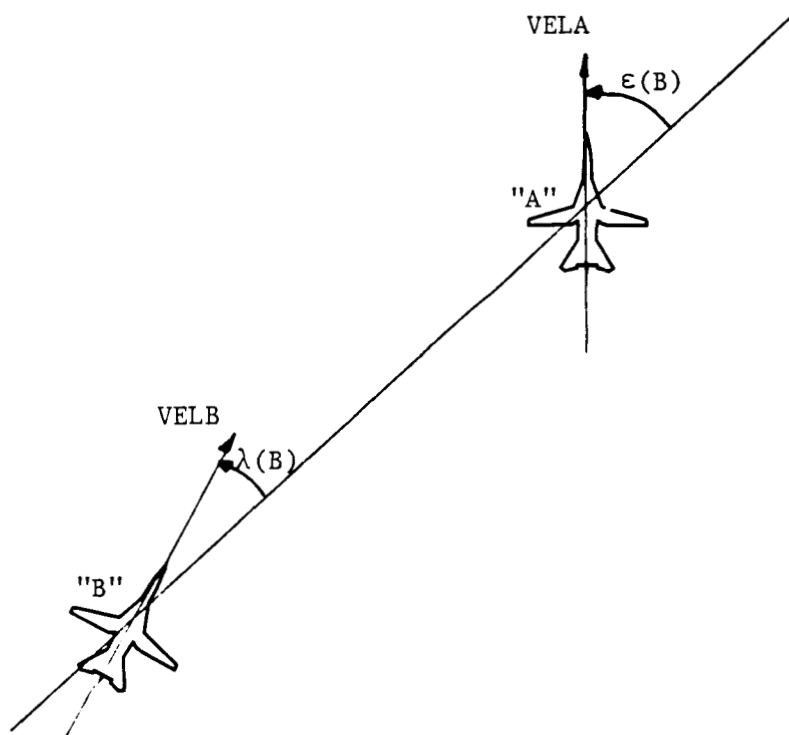
$\lambda(B)$ is the angle between "B"'s velocity vector and the line of sight from "B" to "A". For this reason, it is sometimes referred to as the "line-of-sight angle". This deviation angle is an indication for "B" of where "A" is: $\lambda(B) = 0$ degrees means "A" is directly in front of "B"; $\lambda(B) = 180$ degrees means "A" is directly behind "B".

The angle-off $\epsilon(B)$ is measured between the line-of-sight vector from "A" to "B" and "A"'s velocity vector. It tells "B" where "A" is going relative to "B": $\epsilon(B) = 180$ degrees means "A" is coming directly at "B"; $\epsilon(B) = 0$ degrees is going away from "B". Alternate names for angle-off are: angle-off-tail and aspect angle (Reference 11, page 2-2).

Similar angles can be defined for "A". Inspection of figure 1-4 shows that $\lambda(A) = 180 \text{ deg} - \epsilon(B)$ and $\lambda(B) = 180 \text{ deg} - \epsilon(A)$.

The line-of-sight angle and the angle-off are fundamentally important in air-combat; both for the tactical decision process as well as for the assessment of the current situation. A few clarifying remarks are therefore in order.

First note that the AML program carefully differentiates between line-of-sight angle and deviation angle. In the following discussion, we reference all the angles to aircraft "B", in other words, when we say, line-of-sight angle, we mean aircraft "B"'s line-of-sight angle. By AML's definition, the line-of-sight angle is the angle between the vector from "B"'s cg to "A"'s cg (the line-of-sight vector) and "B"'s body x-axis. The



"B"'s Perspective:

- o The deviation angle λ_B tells "B" where "A" is.
- o The angle-off ϵ_B tells "B" in which direction "A" is going with respect to "B".

Figure 1-4. Definition of Deviation Angle and Angle-off ("B"'s View)

deviation angle, on the other hand, is defined as the angle between the LOS vector and "b"'s velocity vector. Line-of-sight angle and deviation angle therefore are only identical if there exists no sideslip and no angle of attack.

From a tactical point of view, both the deviation angle and the angle-off are important. For gun-firing, the line-of-sight is of primary importance, because the guns are mounted such that they point in the direction of the aircraft's longitudinal axis. For missile firing, both the line-of sight angle and the deviation angle are important, the missile is mounted parallel to the aircraft's longitudinal axis, the initial missile velocity, however, is determined by the aircraft's velocity vector.

One last point: The line-of-sight vector can be changed by the pilot much more rapidly than the deviation angle, because modern fighter airplanes allow very rapid changes of angle of attack of the order of 10 to 20 degrees. This translates directly into a line-of-sight angle change of the same magnitude. The velocity vector however can not be changed that rapidly.

Although these definitions (or equivalent definitions) are widely in use air training manuals as well as air combat simulations, it should be noted that there are ambiguities arising from the fact that the values of the line-of-sight angle and the angle off are between 0 and 180 degrees and always

positive.

For example, if one expresses the situation in the $\lambda(B)$ - $\epsilon(B)$ plane, both situation 1-5 a and 1-5 b will be represented by the same point in that plane, namely $\lambda(B) = 90$ degrees, $\epsilon(B) = 90$ degrees.

Similarly, the two situations 1-5 c and 1-5 d fall into the same point in the λ - ϵ plane, $\lambda(B) = 90$ degrees, $\epsilon(B) = 45$ degrees.

Since 1-5 a represents a tactically different situation from 1-5 b, these ambiguities should be removed if one wants to base the tactical decision on the two angles λ and ϵ . One possibility would be to introduce, in addition to these two angles, also the line of sight angle rate. Assuming equal velocity for the two aircraft, the line of sight angle rate would remain zero for situation 1-5 a, but a large rate would result in situation 1-5 b. Similar observations can be made between situations 1-5 c and 1-5 d.

Notwithstanding these limitations, it has been found useful to introduce a performance index which combines these two angles into a single measure:

$$PI(B) = 50*(1 - \lambda(B)/180) + 50*(1 - \epsilon(B)/180)$$

$$PI(A) = 50*(1 - \lambda(A)/180) + 50*(1 - \epsilon(A)/180)$$

Values of PIB and PIA are illustrated in table 1-2.

For example, during its AML tests (Reference 4),

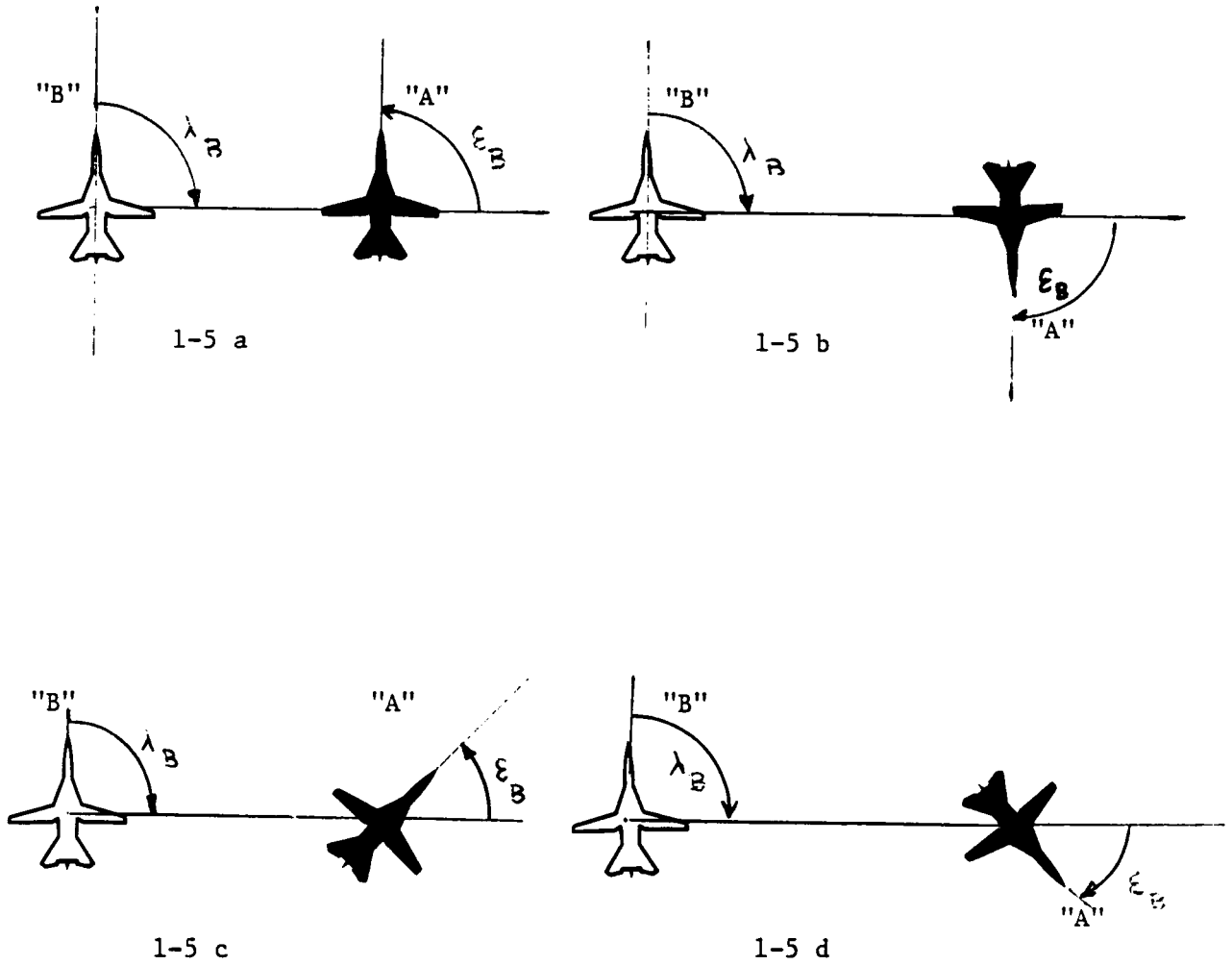


Figure 1-5 Ambiguities in the Line-of-Sight/Angle-Off Representation

	<u>Case</u>	<u>lambda (B)</u> <u>deg.</u>	<u>epsilon (B)</u> <u>deg.</u>	<u>PI (B)</u>	<u>lambda (A)</u> <u>deg.</u>	<u>epsilon (A)</u> <u>deg.</u>	<u>PI (A)</u>
	"B" on "A"'s tail	0	0	100	180	180	0
	"B" attacks "A" on the beam	0	90	75	90	180	25
	Head-on encounter	0	180	50	0	180	50
	"A" attacks "B" on the beam	90	180	25	0	90	75
	"A" on "B"'s tail	180	180	0	0	0	100

Table 1-2 Representative Values of Deviation Angle and Angle-Off

NASA/Langley considered that a pilot (say "B") enters a gun zone if $\lambda(B)$ does not exceed 10 degrees and if $\epsilon(B)$ does not exceed 60 degrees (and the range is less than 3000 feet). This condition corresponds to a performance index $PI(B)$ of 80 or better. (Correspondingly, $PI(A)$ would be 20 or less.) It should also be noted that this value of the PI is a necessary but not sufficient condition for a gun-firing position.

The integrated performance index is the time-averaged value of the instantaneous values of the performance index:

$$IPI = (PI_1*DT + PI_2*DT + \dots + PI_N*DT) / T$$

Offensive time

The offensive time is defined as the accumulated time during which the opponent was in front of the wing line of the reference aircraft. This was one of the figures-of-merit used during the original AML test runs at Langley (Reference 4).

The offensive time with advantage is the accumulated time during which the opponent was in front of the reference aircraft's wing line and the reference aircraft was behind the opponent's wing line. AML also used a more restrictive definition of the offensive time, consisting of the accumulated time during which the reference aircraft's deviation angle was less than 60 degrees and its angle off less than 60 degrees.

Time to first kill

Other performance measures account for the weapon

probability of kill. The time to achieve the first kill is an attractive measure of this kind. However, its drawback is that it does not properly reflect the future impact of the aircraft state at the time the first kill is achieved and its subsequent capability to engage more targets. An initial firing position may be achieved by turning at maximum instantaneous load factor in order to gain an angular advantage. However, this will result in the aircraft losing rapidly energy and thus position itself unfavorably for a subsequent engagement. In typical air-to-air scenarios, it is precisely the purpose of the leader/wingman team concept to take advantage of such situations.

Accumulated probability of kill

A commonly used measure of military effectiveness is the loss exchange ratio, defined as the ratio of the number of enemy killed divided by own losses. In a one-on-one duel in which multiple weapons are exchanged, this reduces to the ratio of the accumulated probabilities of kill. Neuman and Erzberger used this measure (Reference 13) as an alternative to the measures of effectiveness previously discussed.

The common procedure to calculate the exchange ratio is to use the Monte-Carlo method. An alternate method was used in Reference 13 in which the engagement continues independently of the outcome and these trajectories are recorded. A post-processing program uses these trajectories to identify firing opportunities and to compute the accumulated probability of kill. This method was used to avoid the problem often

encountered in air combat simulations that small changes in initial conditions or in the flight path somewhere in the engagement propagate into large differences in outcome. However, this method is limited to lvl, since in m on n there are cooperative effects which depend on the sizes of the forces.

2. DEVELOPMENT HISTORY OF THE AML PROGRAMS

2.1 NASA LANGLEY DMS PROGRAM (AML 75)

Development of the AML program started in 1969 under the sponsorship of the NASA Langley Research Center. The original AML program was developed to operate on NASA Langley Research Center's Differential Maneuvering Simulator (DMS). It is shown in figure 2-1 as the root of the AML program family. It was designed to be an interactive air-combat opponent operating in real time. This original version of the AML program is documented extensively (for example, References 5 and 6).

2.2 THE DMS CONTROL MODEL (AML 76)

In the original version operating on the DMS, AML would "drive" the displayed aircraft by providing body rotational commands p , q , and r to the DMS display program. AML calculated the values for p , q , and r such that the displayed aircraft would achieve an attitude compatible with the following conditions:

- resultant force vector (aerodynamic forces, propulsive forces and gravity force) must lie in the desired maneuver plane.

Air-to-Air Combat

Missile Evasion

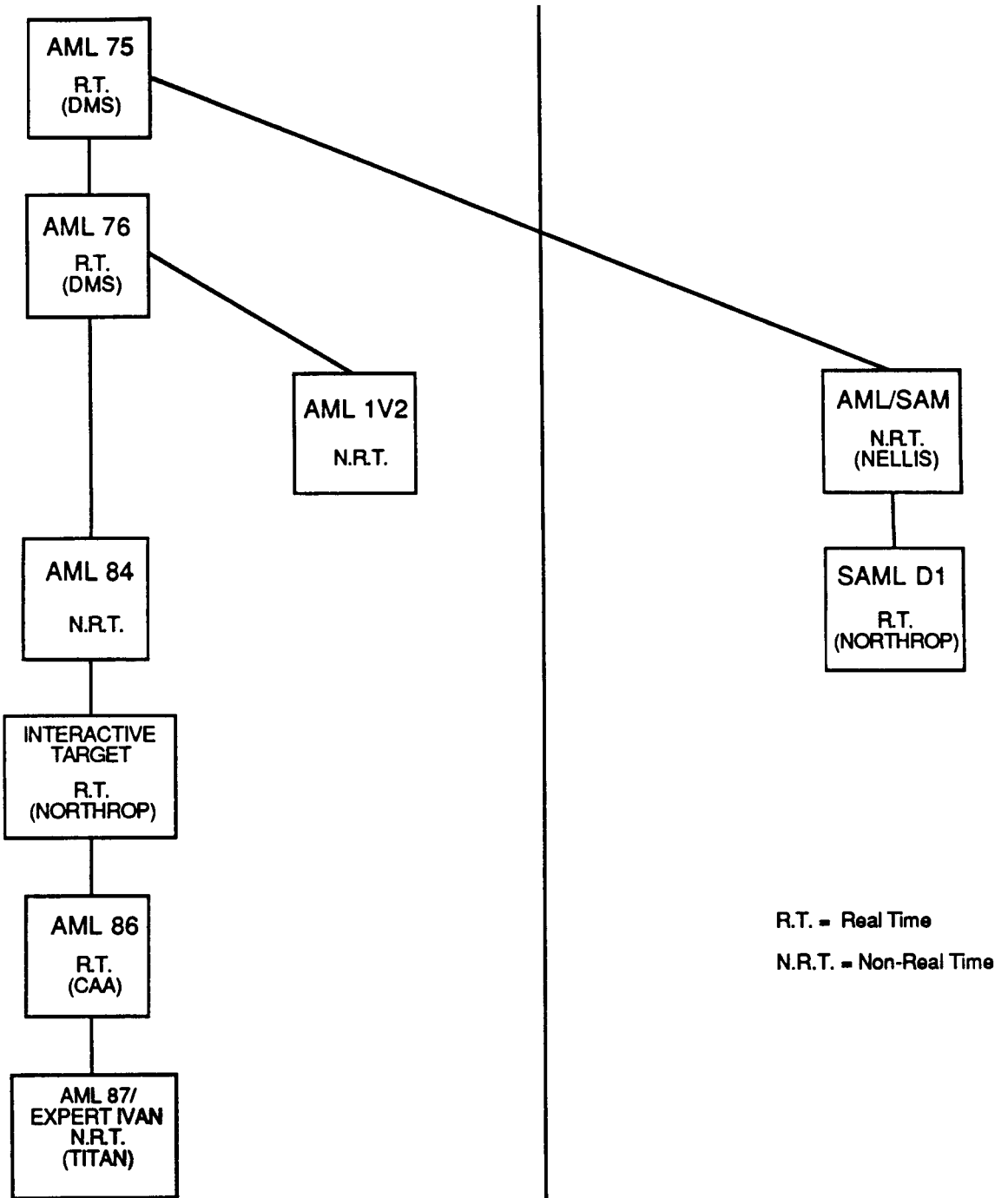


Figure 2-1. Development History of the AML Air-to-Air and Missile-Evasion Programs

- the angle of attack is such that the desired lift is produced
- sideslip is zero

The AML program then filtered these commanded body rotational rates in order to achieve a smooth motion of the displayed aircraft. However, when a pilot lost an engagement with the AML, he had a tendency to claim that the AML driven aircraft would perform flight maneuvers which were outside the performance envelope of the real aircraft. To counter this argument, a control system was developed which would actually move the simulated aircraft's control surfaces in exactly the same manner as the pilot did it with the stick and the rudder pedals. These commanded control surface deflections were then fed into the identical set of equations of motion as the were used to drive the human piloted aircraft. The development of this control system is described in reference 20. A thorough comparison between the performance of the original AML (called the performace-model AML) and the AML with a control-system is contained in reference 4.

2.3 A ONE-VERSUS-TWO VERSION OF AML (AML 1V2).

The Human Research Laboratories of the Air Force (AFHRL) sponsored subsequently an extension of the one-versus-one AML version to a one-versus-two version Here AML represents the single aircraft opposing two bogeys. This lead to a batch-version of AML which handles the one-versus-two situation based

on a set of value functions taking into account the relative situation between all three combatants. This version also replaced some of the binary value functions of the original AML by continuous functions, thus giving a better resolution between trial maneuvers and avoiding ambiguity in the scoring of different maneuvers.

2.4 AML WITH REVISED EQUATIONS OF MOTION (AML 84)

The original AML (AML 75) had a number of known deficiencies. The most serious one was an abnormal behavior of the AML aircraft when it approached 90 degrees in a vertical or in a near vertical turn. This anomaly was not due to the singularity of the Euler angles at $\theta = 90$ degrees (AML uses quaternions for the attitude integration and consequently there is no singularity at any attitude). The problem rather had to do with the decision logic and it may be explained in somewhat simplified form as follows: Most maneuvers in AML are executed in "maneuver-planes". A maneuver-plane always passes through the aircraft's velocity vector. Certain other parts of the decision logic are based on the line-of-sight angle, which is, in part, determined by the direction of the aircraft's body-x axis. In a vertical loop, under high angle-of-attack conditions, it will happen that the body axis has already exceeded the 90 degrees pitch angle, but the velocity vector's pitch angle is still below 90 degrees and still increasing. Under such a situation, it can occur that the AML reverses its maneuver command inappropriately. Specifically, it will command a maneuver plane

rotation angle of 180 degrees (or close to 180 degrees) when in reality, the maneuver-plane rotation angle for the intended maneuver should be zero (or close to zero). This reversal can take place during several subsequent decisions. The result in a flight simulator is that it looks as if AML wouldn't know what to do. The long range effect is a hammer-head stall of the AML aircraft. Section 4 of this report explains briefly how this problem was solved.

2.5 NORTHROP AEROSCIENCES LABORATORY AML (INTERACTIVE TARGET).

The Aerosciences Laboratory of the Aircraft Division of the Northrop Corporation, which had an early version of AML installed on their moving base simulator, was interested in an AML implementation with the new equations of motion, which eliminated completely the "over the top" problem explained in the previous paragraph. However, the computer hosting AML was a Harris Slash 4 minicomputer whose computational capability was inadequate to support AML in real time, not even with a frame time as large as 50 milliseconds. To reduce execution time, we abandoned the concept of trial maneuvers and of selecting the most promising of these trial maneuvers. Instead, we developed a logic which resembled closely the production rules of the then popular expert systems. This not only allowed us to perform a tactical decision well within the allocated frame time, but it also gave AML the flavor of an AI program.

2.6 GENERAL ELECTRIC'S AML VERSION ON THE SIMULATOR FOR ADVANCED AIRMANSHIP (AML 86)

At the time of this report, this is the most advanced real time version of AML. It uses basically the same decision logic as the Northrop version, with some added improvements for low speed, low energy avoidance. The host machine is an SEL 32/97 computer and the visual display is a General Electric Compuscene III computer generated image. AML86 has a number of additional features, such as minimum allowable altitude for the AML aircraft, a choice between three different aircraft (F-4, F-5 or F-15), a large number of selectable "canned" maneuvers for the AML aircraft and most interesting, a selectable skill-level for the AML aircraft. The skill level of the AML aircraft can be selected to be "ACE", "AVERAGE", or "GRAPE".

2.7 AML 87 / EXPERT IVAN

Presently, there is an in-house effort going on at Titan with the two objectives of:

1. - Expand AML's decision logic to BVR
2. - Expand AML's decision logic to handle multiple aircraft on both sides.

AML 87 is strictly a production rule based system, the rule-base being built by Navy fighter pilots with current experience in air-to-air combat in F-14's and F-18's.

2.8 MISSILE EVASION AML (AML/SAM)

The initial success of the AML program as an "iron pilot" in the DMS created confidence that the AML decision logic could be changed to "fly" AML such that it would avoid a surface-to-air missile. This work was initially sponsored by the Tactical Fighter Weapons Center at Nellis AFB, Nevada. The obvious required change to AML was to replace the value functions (which favored achieving a six-o'clock position with respect to the opponent) to functions which favored achieving a large distance between the missile and the AML aircraft. Obviously, the type of trial maneuvers also had to be changed. Less obvious is the fact that in case of missile evasion, a short term maneuver optimization (as it is performed in the air-to-air combat version) will not generate maneuvers with acceptable miss-distance. It is necessary to carry out the optimization from the decision time all the way to the impact (or the point of closest approach of the missile). The decision logic of the AML program was modified to implement these requirements and very successful evasive maneuvers against surface-to-air missiles, such as the SA6 were generated by AML. The program ran in non-real time on a CDC Cyber computer.

2.9 PILOT'S ASSOCIATE D1 AML PROGRAM (SAML D1)

The Aircraft Division of the Northrop Corporation participated in the demonstration phase of the Pilot's Associate

D1 program. The AML/SAM program was modified to work in real time on a flight simulator. AML determined suitable evasive maneuvers for the aircraft. These maneuvers were generated based on a set of production rules. The AML generated maneuvers were either used to provide cues displayed to the pilot on the heads'up display or they were fed directly into a flight control system. AML successfully avoided, at very low altitude, two SAMS simultaneously. For further details, see reference 21.

3. THE BASIC AML IF => THEN LOGIC

3.1 TERMINOLOGY

The purpose of this section is to give an overview of the tactics currently implemented in the IF => THEN version of AML which is in use at various flight simulation facilities. To avoid confusion, we will first clarify some terminology. In a real-time, one-on-one environment on a flight simulator, the AML driven aircraft is called A/C "B", or for short, AMLB. When the program operates in a batch environment, the opponent of AMLB is AMLA. Each of AMLA and AMLB can implement either the "trial-maneuver-logic" or the IF => THEN logic. The following discussion assumes that the "B" aircraft is driven by the IF => THEN logic. In the rest of the report, AMLA is a "trial-maneuver logic " AML.

3.2 COMMONALITY WITH THE TRIAL-MANEUVER LOGIC

3.2.1 Timing Considerations.

In the IF => THEN logic as well as in the trial maneuver logic, two time-intervals are used for maneuver decisions. The first one, which is the smaller of the two is equal to the integration stepsize (alternatively called frame-time or cycle-time). The AML maneuver logic subroutine (TACTICB, see figure 3-3) is invoked every integration step. At each invocation, the AML logic unconditionally checks for the necessity of either initiating a dive recovery or to continue a dive recovery currently in progress. If no dive recovery requirements exist,

then the logic tests whether there is time to perform a new tactical decision. This second time-interval between tactical decisions is called decision-interval. For close-in, one-on-one air-to-air combat, it is typically between 0.5 and 1.5 seconds.

(For missile evasion it is shorter and for a decision in a BVR situation it may be considerably longer).

3.2.2 Maneuver Plane Concept.

A significant contribution to the success of the early versions of the AML program came from the concept of the maneuver-plane. Strictly speaking, one should not call this plane a maneuver plane, but rather a maneuver half-plane. (As it is properly called in reference 22). It is the half plane in which, ideally, the next segment of the AML aircraft velocity vector will lie. It extends through the AML driven aircraft's velocity vector towards the side of the cockpit. The maneuver plane provides (1) a convenient mechanism to specify AML maneuvers (both in the trial maneuver and in the IF => THEN version) and (2) a computationally efficient way for prediction of the aircraft's future position and attitude. In the IF => Then logic, the maneuver-plane serves to specify the parameters for lead or lag pursuit maneuvers. The maneuver plane and its associated maneuver plane coordinate system are illustrated in figure 3-1.

The crucial problem in both AML versions is to control the aircraft's body rotational rates in such a manner that:

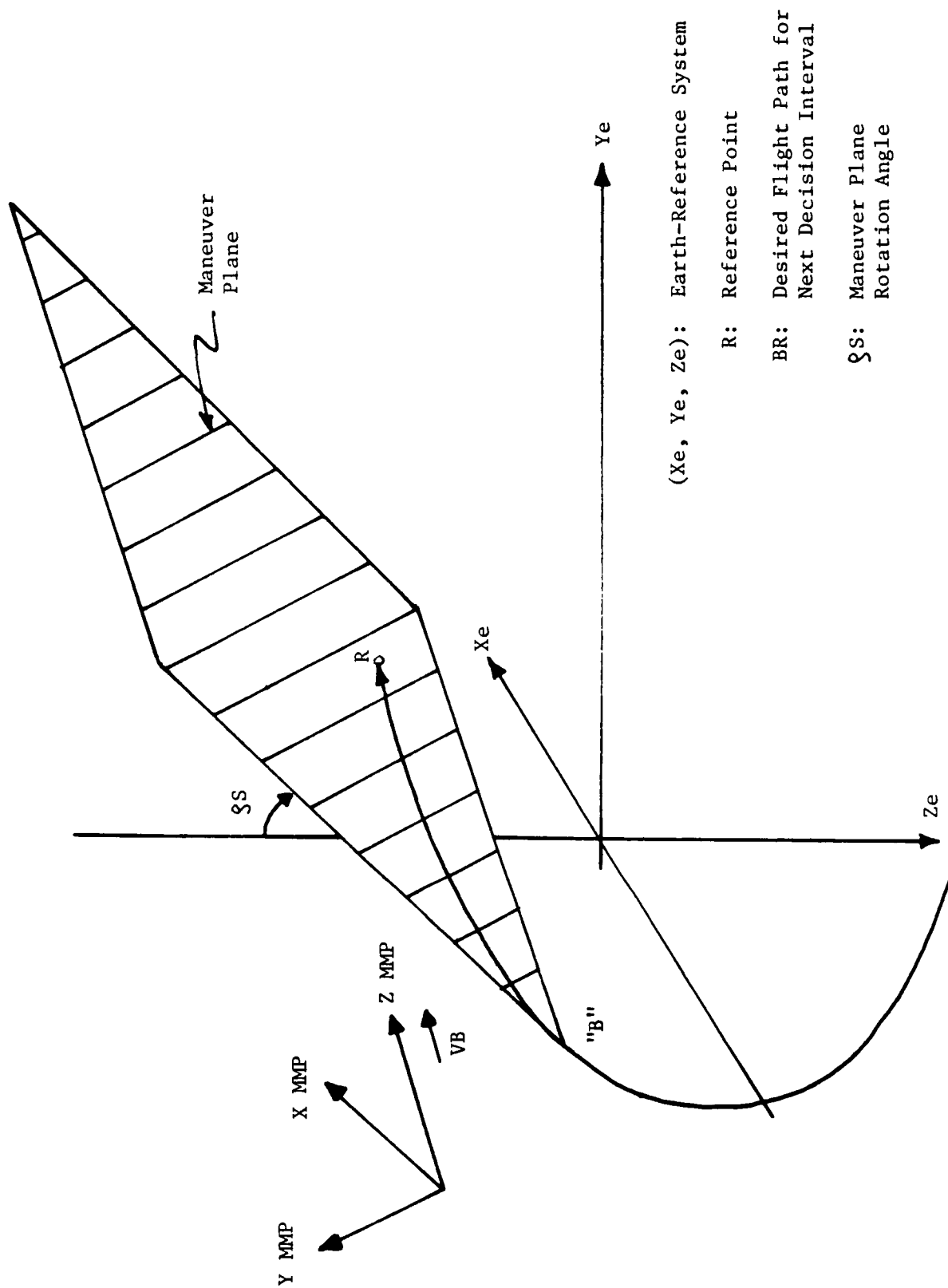


Figure 3-1. Illustration of Maneuver-Plane Concept

- a) they are physically executable under the prevailing flight conditions and
- b) the aircraft's velocity vector remains in the specified maneuverplane.

3.3 DECISION HIERARCHY

3.3.1 Ground Avoidance

The ground-avoidance logic is executed every integration step. This reflects the fact that ground-avoidance has higher priority than any other tactical decision. In both AML versions the decision on whether a ground-avoidance maneuver is required is based on a two dimensional table of the dive recovery angle. This angle is a function of airspeed and altitude. In the IF => THEN logic, a dive recovery maneuver leaves no choice, it is a roll to wings level followed by a maximum instantaneous g pullup. The throttle is controlled such that the aircraft is going to fly at corner-velocity. The dive-recovery maneuver may therefore succinctly be described as a maximum g turn in a maneuver-plane whose rotation angle ρ is zero.

3.3.2 The Pointing Algorithm

If dive recovery is not required, the program performs a test whether the aircraft should be controlled in such a way that its nose (i.e. its longitudinal axis) will point at the opponent or at a specified point in front of the opponent. This

is the only maneuver in the AML program (except of course the "canned" maneuvers in certain versions on flight simulators) where the maneuver is not based on a maneuver plane, but where directly body rotational rates which will bring the aircraft into the desired attitude, are calculated. The pointing algorithm is described in more detail in section 4.

3.3.3 The Lead/Lag Maneuver Logic.

These maneuvers form the heart of the basic AML maneuver decision logic. They implement one of the basic rules of air combat: Point your nose towards the opponent. The refinement consists in the determination of the exact point in reference with the opponent towards which we want to point the aircraft (behind = lag, in front = lead or exactly at the opponent = pure pursuit); the other refinements being the rate of turn by which we want to achieve this goal (in other words, the load factor) and finally how much thrust we will apply (throttle setting). The decision on whether to fly lead, lag or pure pursuit is based on the values of the line of sight angle and the angle off, as illustrated in figure 3-2.

Load Factor Selection The load factor is also selected as part of the LLG. This selection process, however, is primarily determined on the basis of airspeed considerations. A high load factor results in a high turn rate, which is desirable to achieve a firing position as quickly as possible. However, turns at the maximum load factor create a lot of drag which causes the airspeed to drop rapidly. This is actually desirable

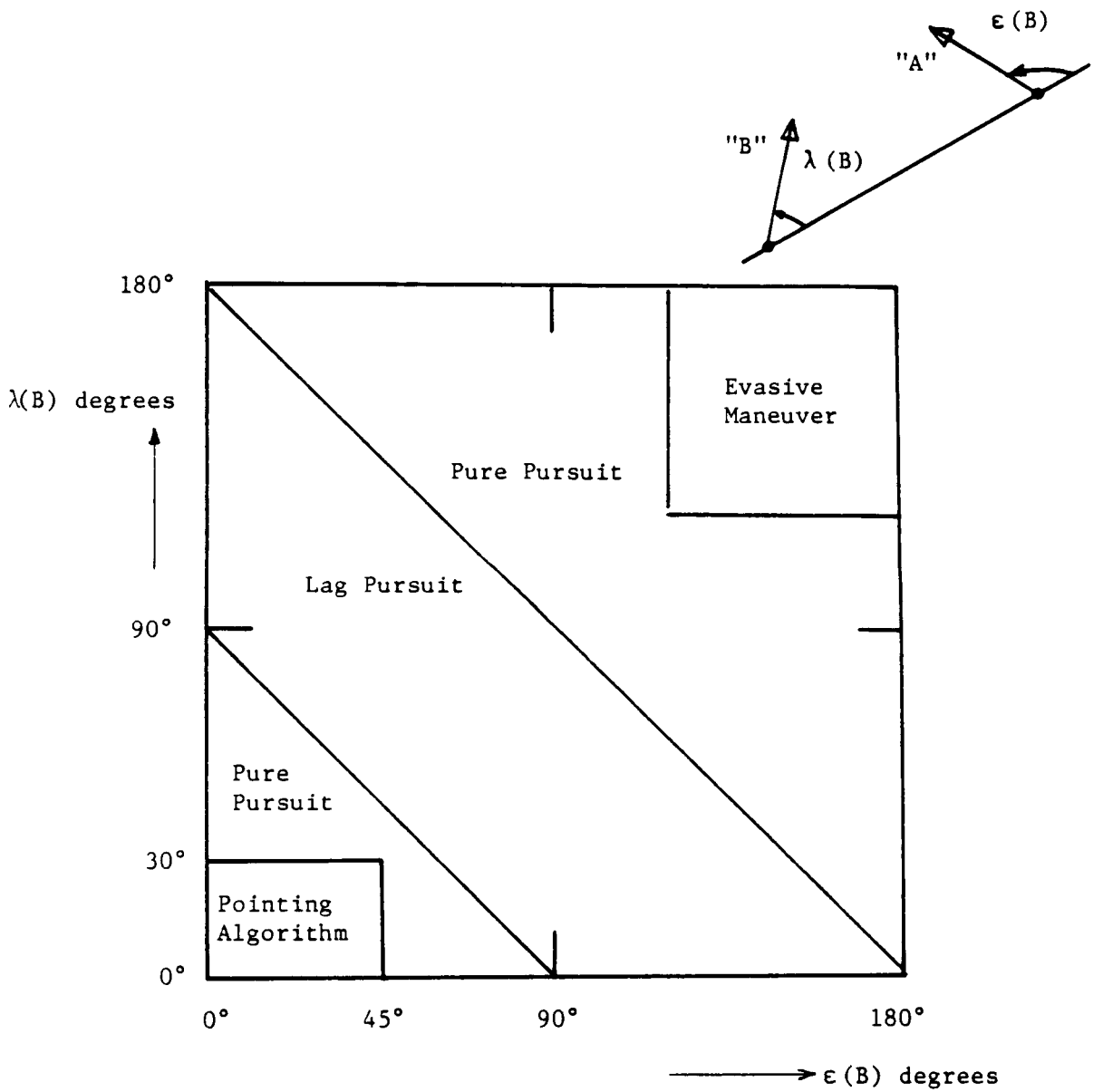


Figure 3-2. Regions for Steering Laws

when the current airspeed is above corner velocity, the velocity which yields the highest turn rate. For this reason, when the current airspeed is above corner velocity, the maximum load factor is commanded. When the current airspeed is near or below corner velocity, the sustained load factor is commanded to avoid losing further energy. In B's forward sector (LOS < 60 degrees), an additional test is performed which compares the load factor described previously, which is airspeed-oriented, to the load factor corresponding to the desired flight path, i.e. the flight path which intercepts the reference point. This "intercept trajectory" load factor is selected if it is lower than the airspeed-oriented load factor.

The pointing algorithm could generate negative load factors. An option to command negative load factors in the maneuver-plane method has been partially implemented. The load factors commanded in the original AML were always positive. The equation for the maneuver plane is given by (p 53 of Reference 6):

$$\text{Rhos} = \text{arc tan} \frac{-\dot{y}_{e,T} t_x + \dot{x}_{e,T} t_y}{(\dot{x}_{e,T} \dot{z}_{e,T} t_x + \dot{y}_{e,T} \dot{z}_{e,T} t_y - V_{h,T}^2 t_z) / V_T}$$

There are two solutions to this equation, Rhos and Rhos + 180 degrees. The second solution corresponds precisely to negative a load factor, and is calculated in this version of AML. A negative load factor will be chosen if all these conditions are satisfied:

(1) B's airspeed must be lower than A's
(2) A must be in B's forward quarter and low
(3) B's current roll angle must not exceed 30 degrees;
otherwise, it is preferable to roll inverted under a positive
load factor.

(4) the negative load factor yields the smallest
variation in maneuver plane rotation angle (and therefore in
roll angle.)

These conditions are restrictive and favor the well-known
pilot preference for positive load factors. They will however
make possible the use of a negative load factor for the purpose
of bringing B's nose onto A while avoiding a high positive load
factor and, hence, unnecessary loss of airspeed.

Throttle Control The throttle control laws are set
independently and can be summarized as follows:

(1) In dive recovery, set the throttle to bring the
airspeed near the corner velocity. Thus, the throttle is set to
idle if the airspeed is above corner velocity. The throttle is
otherwise set to afterburner.

(2) Under other conditions, the avoidance of an
overshoot takes precedence over the rule enunciated above. This
will occur if A is in front of B and B has a high overtake
velocity. In this case, the throttle is set to idle.

SUBROUTINE REACTB

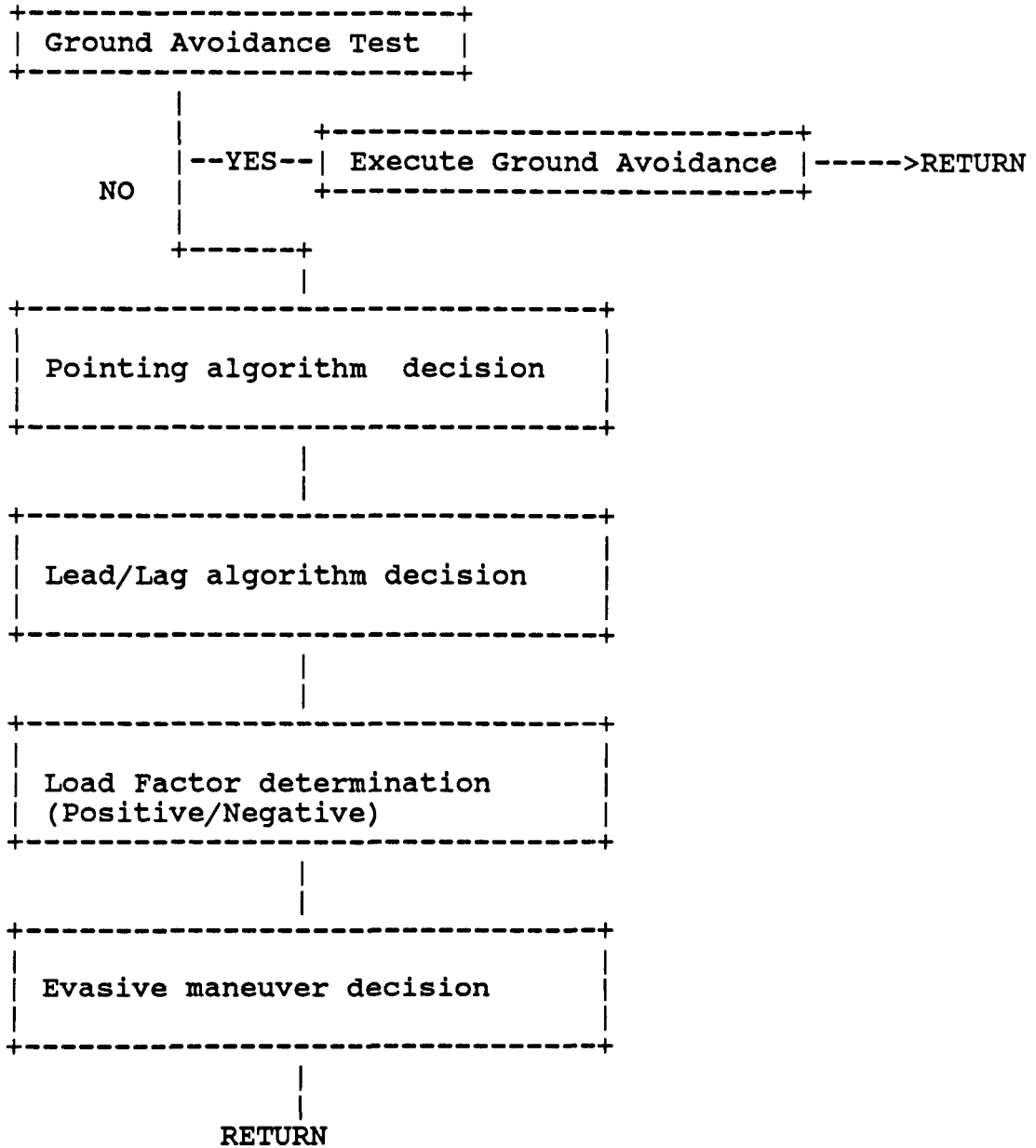


Figure 3-3. Summary of the AMLB Control Laws.

4. AIRCRAFT AND CONTROL SYSTEM DYNAMICS

4.1 SELECTING AN APPROPRIATE MODEL

What constitutes an appropriate model depends on the purpose of the simulation. As illustration, consider the two extreme cases: (1) Development of evasive maneuvers against an air-to air missile (2) Training of pilots in ECM tactics in a BVR environment. To capture the intricate dynamics between a highly agile missile and a fighter aircraft, it is necessary to simulate aircraft response to control surface deflections. This will rotate the aircraft in such a way that at any instant of time, the missile seeker head "sees" the aircraft under the proper aspect angle. In the BVR case, representing the aircraft as a point-mass may be adequate. Close-in visual air-to-air combat in a flight simulator lies somewhere between these two extremes. To achieve the necessary accuracy for the CIC simulation, two key performances of the aircraft must be modeled accurately:

- 1) The Normal Acceleration
- 2) The Roll Dynamics

Normal acceleration determines how tight the fighter can turn and whether or not he loses energy during the turn. Roll performance determines how quickly the fighter can change the direction of the lift. In AML, roll performance is the determining factor in how fast the flight path can be changed from one maneuver plane to another maneuver plane. The two important parameters for roll performance are maximum roll rate

and maximum roll acceleration. As Shaw (Reference 10 ,page 414) points out:

" In air combat, continuous rolls of more than 180 degrees are seldom required. Because a certain length of time is necessary to acclerate the roll rate from zero to its maximum value, maximum stabilized roll rate may not be reached during such short periods of roll. Therefore, roll acceleration is often the controlling factor in combat performance."

Shaw's quote is certainly true for air-to-air combat and even more so for evasive maneuvering against missiles.

The problem of properly simulating roll performance is complicated by the fact that a change in bank angle has often to be achieved under high angle of attack or that coupled with a change in bank angle is a large change in angle of attack. In AML, a maximum roll rate and a maximum pitch rate is specified. Both are a function of the particular aircraft type represented by AML. If a maneuver command requires both a large change in the pitch angle ($\hat{\theta}$) and the roll angle ($\hat{\phi}$) the details of how this maneuver is performed depend a great deal on the ratio between maximum pitch rate and maximum roll rate. A proposed method, which, due to lack of funding never has been implemented, is to calculate the maximum available pitch and roll acceleration every time one of these extreme maneuvers has to be performed:

As a first approximation, we suggest to calculate \dot{p}_{max} and \dot{q}_{max} as follows:

$$\dot{p}_{max} = \frac{g U^2 S b}{2} C_{lca} \frac{1}{I_{xx}} \delta a_{max} \quad \dot{q}_{max} = \frac{g U^2 S c}{2} C_{mce} \frac{1}{I_{yy}} \delta e_{max}$$

C_{lca} is the nondimensional control derivative for the

rolling moment due to aileron deflection and δa_{max} is the maximum available aileron deflection. To be accurate, $C_{l\delta a}$ would have to be known as a function of Mach number and of the angle of attack. Herin lies the problem: It is often difficult to obtain these control derivatives for the extreme flight conditions which occur so often in air-to-air combat. Analogous remarks apply for $C_{m\delta e}$ (control derivative for pitching moment due to elevator deflection)

If the AML maneuver command is fed into a simulated (or eventually, into a real) flight control system, the problem of properly simulating pitch and roll performance under high angles of attack is greatly simplified. The aircraft (F-X) in Northrop's Pilot Associate Program D1 was controlled by feeding AML provided load-factor and bank-angle commands into the flight control system. It can therefore be assumed that the dynamic response of the F-X to AML maneuver commands was very realistic.

4.2 SYNOPSIS OF THE CURRENT ATTITUDE CONTROL MECHANISM

A detailed account of the new equations of motion can be found in reference 14 where all the mathematical background underlying the treatment of the attitude control equations is presented. For the sake of completeness of this report, the significant changes between the AML-75 and AML-84 are summarized below.

As an introduction, a few words about "degrees of freedom" of an airplane model may be in order. If we consider the

aircraft to be a rigid body, then, by definition of classical mechanics, the number of degrees of freedom is equal to the number of independent coordinates required to uniquely define the position and the attitude of the body. A single rigid body can have at most six degrees of freedom (3 translational, 3 rotational). If we constrain the motion, the number of degrees of freedom is reduced, e.g. an aircraft whose cg could only move in a plane and whose longitudinal axis is constrained to lie in that plane, has 3 degrees of freedom (2 translational, one rotational). How many degrees of freedom does the AML model have? The answer is this: We try to make it a five degree of freedom motion, by postulating that the sideslip angle and the rate of the sideslip angle (not the yaw angle and the yaw rate!) be zero. But during a transition from flight in one maneuver plane into some other maneuver plane the calculated values of p and r do not necessarily exactly guarantee a zero sideslip angle. The model is therefore a true six degree of freedom model.

Most of the maneuver commands in AML are triplets defining

- a maneuver plane (by means of the maneuver plane rotation angle ρ)
- a load factor
- a throttle setting

Given the above three parameters, one can calculate what the aircraft's attitude, at the present time, (or one integration stepsize ahead) should be for the aircraft to fly in the commanded maneuver plane with the commanded load factor and

with zero sideslip.

Once we know the aircrafts desired attitude, we can calculate body rotational rates which will rotate the aircraft from its present attitude into its desired attitude. The important contribution of the "new equations of motion" is the way how these desired body rotational rates are calculated. To determine values of p , q and r Euler angles $\hat{\Psi}$, $\hat{\Theta}$ and $\hat{\Phi}$ are calculated. These angles are expressed in the aircrafts present body axis system and not, as in the "old equations of motion" in the inertial reference system. Therefore, only $\hat{\Phi}$ ever can become really large, $\hat{\Theta}$ and $\hat{\Psi}$ will always be relatively small ($\hat{\Theta}$ will never be greater than the difference between maximum and minimum allowable angle of attack) Consequently, there will never be a singularity in the set of Euler angles $\hat{\Psi}$, $\hat{\Theta}$ and $\hat{\Phi}$, and as a consequence, the previously encountered problem of "going over the top" will no longer occur.

4.3 REFINED CALCULATION OF COMMANDED PITCH RATE

The procedure to determine p , q and r as developed in reference 10 appeared to work reasonably well in the AML-84 program, but occasionally, the AML driven aircraft would fly into the ground even though dive recovery was initiated at the appropriate time. Careful analysis of trajectories during dive recovery revealed that the aircraft never achieved the commanded load factor but consistently flew with a load factor less than

the commanded load factor during the pull out maneuver. At first we thought that the problem lies in the first order transfer function between q command and q achieved. But even as the time constant in this transfer function was reduced to a very small value, the problem persisted. The real reason for the discrepancy between commanded angle of attack and achieved angle of attack lies in the fact that the calculation of the "desired" aircraft attitude is based on the present velocity vector. However, if the aircraft undergoes a large normal acceleration, the velocity vector will rotate during the next integration step and therefore, the commanded pitch rate must be increased by the rotational rate of the velocity vector which is:

$$\omega = \frac{a_n}{V}$$

In a hard turn, a better value for q commanded therefore is:

$$q_{\text{com}} = \min (q_{\text{max}} , \text{abs} (\hat{\theta}/dt)) \text{ sign} (\hat{\theta}) + \frac{\text{Lift}}{g^*V}$$

4.4 THE POINTING CONTROL SYSTEM

One of the most significant additions and improvements to the solution of the AML driven aircraft attitude control is the incorporation of a "pointing" control system. In several studies with AML, it was found that the AML controlled aircraft performed quite well to get behind the opponent, but once there, it lacked the capability to reduce the line of sight angle to the small value required for a gun solution. Controlling the aircraft by means of maneuver planes and loadfactors is indeed not a suitable way to point the aircraft's nose in a desired

direction. Therefore, a control system was implemented which would directly command roll and pitch rate to point the aircraft's longitudinal axis into a desired direction. Figure 4-1 illustrates in form of a block diagram the pointing control system. This control system is a modification of a control system suggested for use in surface-to-air missiles (Reference 25, page 37). It is highly effective in controlling the AML driven aircraft. The problem is to find appropriate values for the various gains in the control system if a new fighter aircraft is implemented in AML.

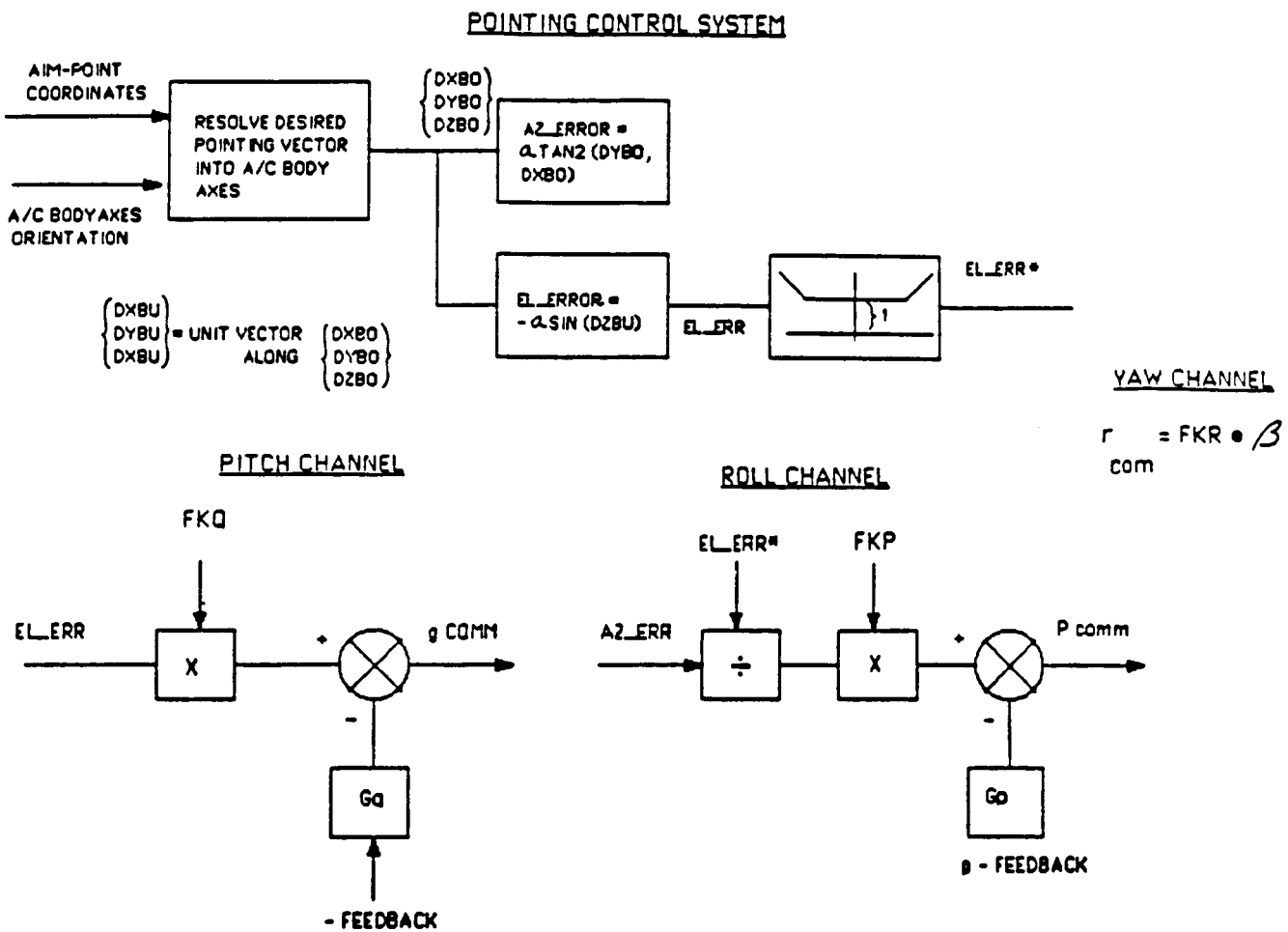


Figure 4-1. Block Diagram of Pointing Control System

5. SAMPLE TRIAL MANEUVERS AML VERSUS IF => THEN AML RUNS

A series of test cases was conducted to exercise the AMLB logic described in section 3. In this series of runs, the "A" aircraft was an F-15 controlled by the trial-maneuver logic AML (AMLA). The "B" aircraft was an F-4, controlled by the IF => THEN AML logic (AMLB). The initial conditions selected for these cases are shown in Table 5-1. A variety of initial velocities, altitudes and initial ranges were used. Initial velocities of M.77 at 20,000 feet were selected because they represent a typical entry conditions into the air combat arena. On the other hand, initial velocities of M.46 correspond to typical corner velocities at 10,000 feet. The initial angular conditions vary from neutral to very unfavorable to the F-4 : the initial PI range approximately from 50 to 90. Since in addition the F-4 is a considerably less performing aircraft than the F-15, which has a smaller turning radius, one would expect that the situation would develop in favor of the F-15.

Table 5-1
Initial values for sample runs

Region /Case	---- "A" ----			----- "B" -----			
	Mach No.	Altitude ft	Relative Range ft	Mach No.	Altitude ft	Eps(B) deg	Lmbd(B) deg
2/1	.46	10,000	1000	.46	10,100	90	90
3/1	.46	10,100	1000	.46	10,100	90	135
4/1	.77	20,000	2250	.77	20,000	154	30
4/2	.77	20,000	2000	.77	20,000	180	45
4/3	.46	10,000	1000	.46	10,100	135	0
4/4	.46	10,000	1000	.46	10,100	135	45
5/1	.46	10,000	1000	.46	10,000	135	90
5/2	.46	10,000	1000	.46	10,000	135	90
5/3	.76	16,000	3500	.77	20,000	135	90
5/4	.77	24,000	3500	.77	20,000	135	90
6/1	.46	10,000	1000	.46	10,100	135	135

A natural way of classifying these runs is to define regions in the epsilon(B)- lambda(B) plane which is used for AMLB maneuver selection (Section 3). This plane was divided in the regions shown below:

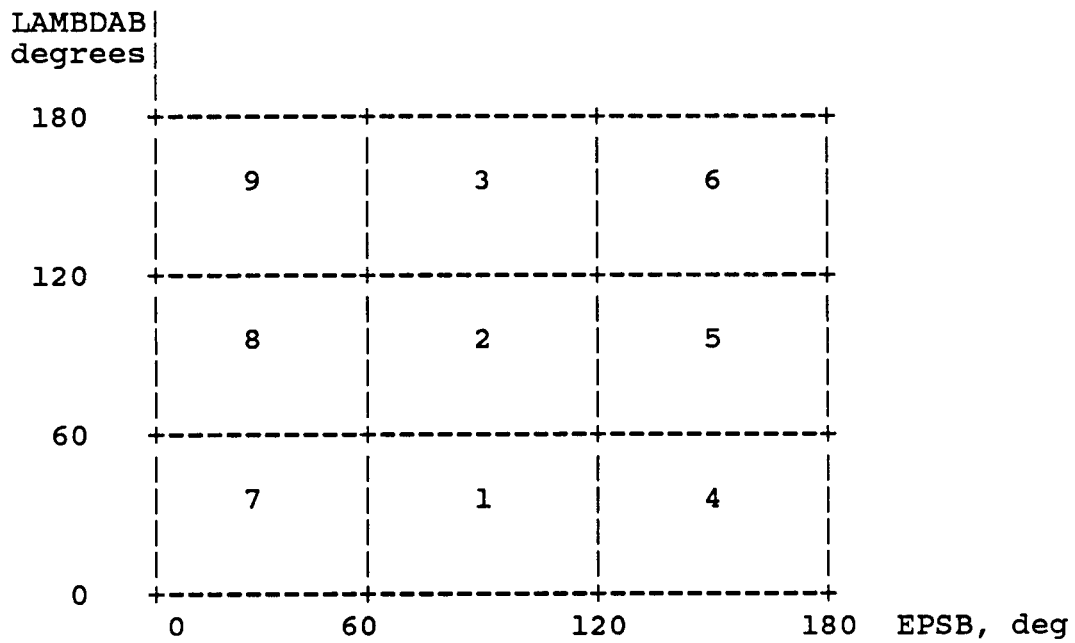


Figure 5-1. Definition of regions

These runs were made for a fixed period of time, typically 20 seconds, which only allows the observation of the development of the initial maneuvers. The X-Y traces of the trajectories were plotted for that period of time. Also, additional pertinent information such as altitude and airspeed have been indicated as labels on these plots. For all these runs, the performance index for aircraft "A", $PI(A)$, and the integrated performance index, $IPI(A)$, were plotted as a function of time. These were discussed in Section 1.6. Since $PI(B) = 100 - PI(A)$, only $PI(A)$ was plotted. The PI yields an indication of the relative angular attitude between the two aircraft and complements the information from the X-Y trace. It will be recalled that a PI of 80 is required (but not sufficient) to achieve a firing position.

5.1 REGION 2

A run was made in this region corresponding to neutral conditions in all initial variables (angles, velocities and altitude.) These results are shown in Figures 5-2 and 5-3.

It would be expected that the superior-turning F-15 would gain advantage. However, the performance index plot indicates that the situation remains essentially neutral throughout the simulated engagement. It will be noted that there is loss of airspeed on both sides (but is more severe for AMLA), and that the engagement remains approximately co-altitude.

An examination of the X-Y traces in Figure 5-2 contrasts

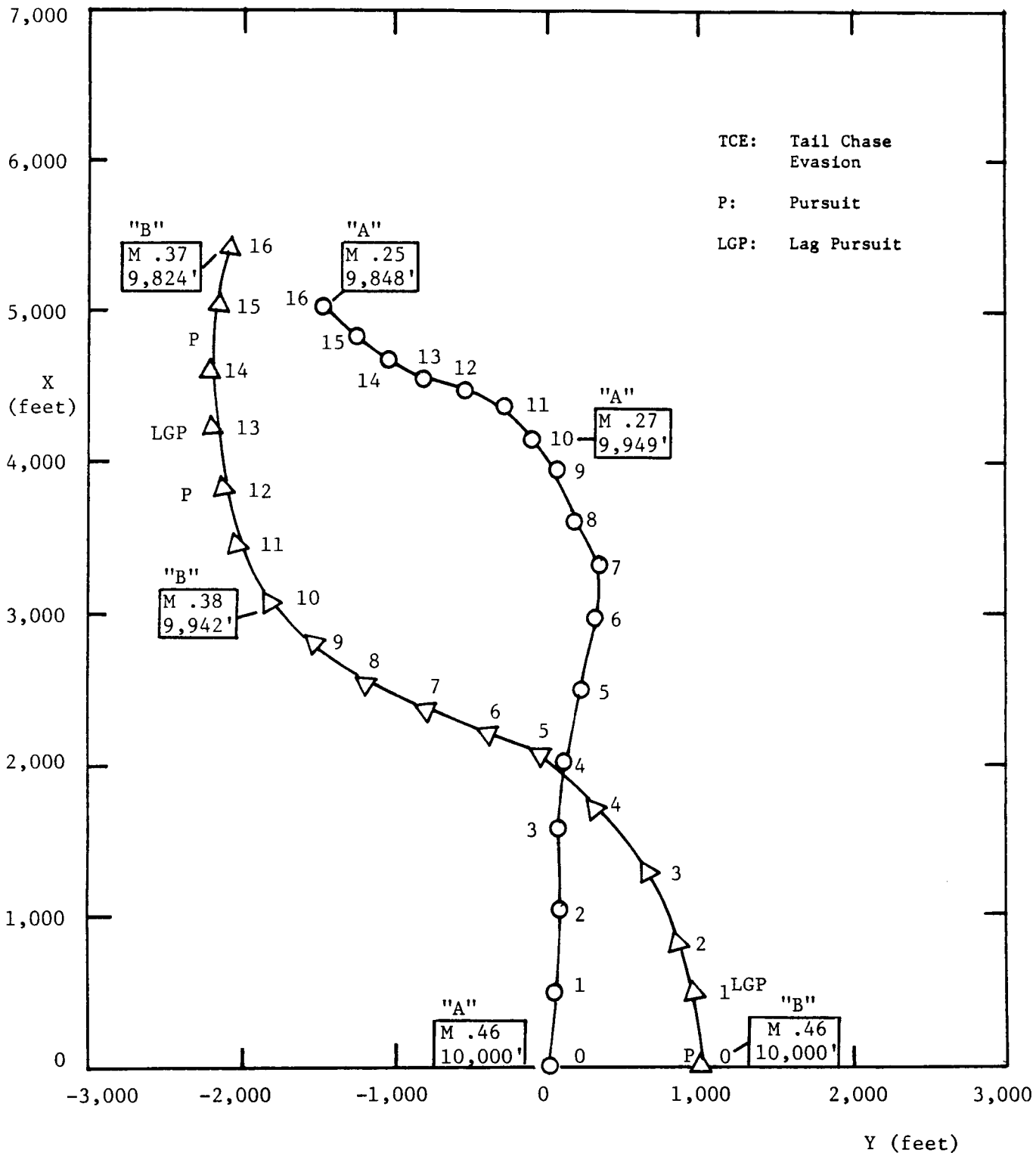


Figure 5-2
 Region 2 ($\lambda(B)=90$ deg.,
 $\epsilon(B)=90$ deg.) X-Y Trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

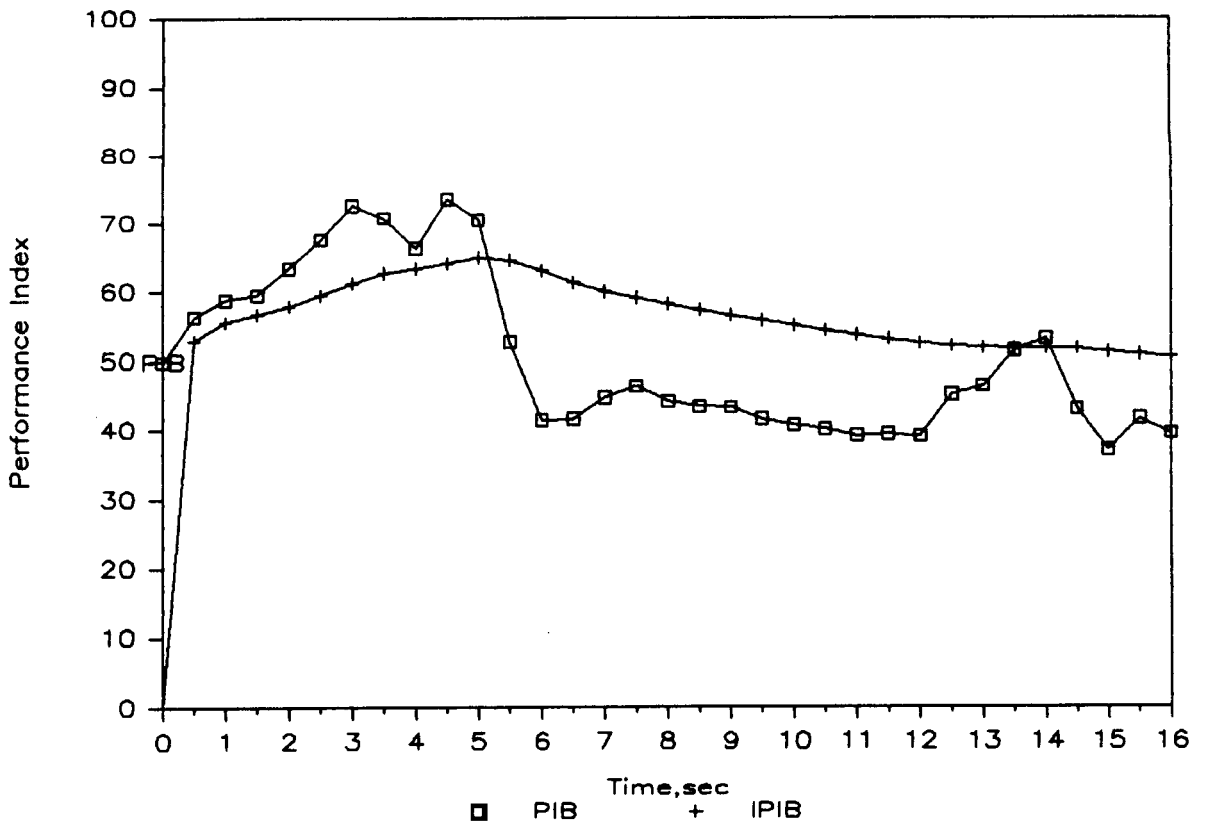


Figure 5-3 Region 2
PI Plot

the initial behavior of AMLA/F-15 and AMLB/F-4E. AMLB begins to turn immediately, while AMLA reacts very little until $t = 7$ seconds. Then AMLA begins a turn reversal during $t = 7$ seconds to 16 seconds. As indicated in Reference 10, this tactic should be expected from an aircraft with a smaller turn radius such as the F-15 compared to the F-4E. It will be seen again in more simulated runs. This tactic fails in the present case probably due to the small initial range (1000 ft).

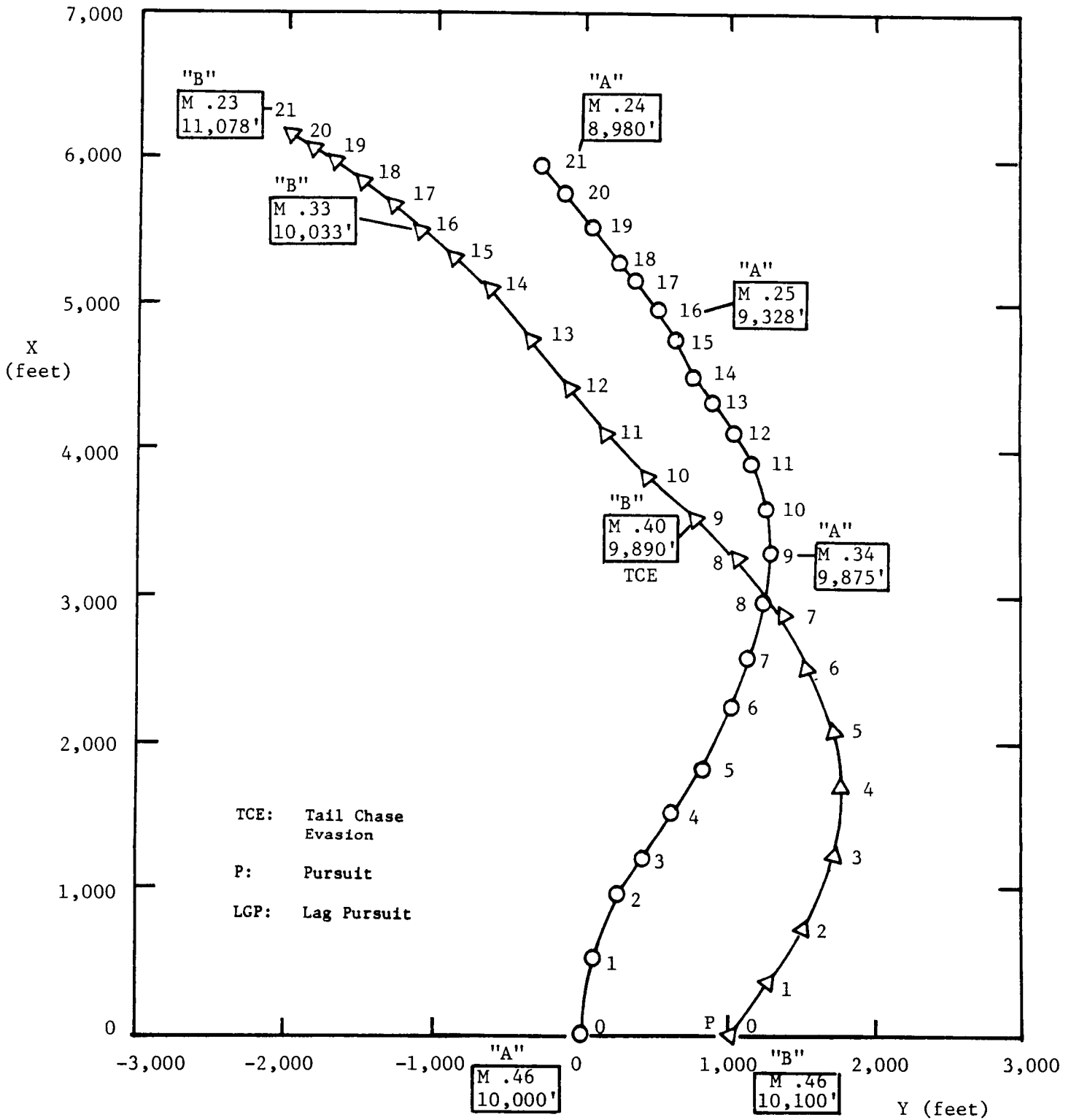
The steering laws used by "B" have also been indicated in Figure 5-2. The pursuit law is used intermittently ($t = 0$ sec to 1 sec; 12 sec to 13 sec; 14.5 sec to 16 sec) The lag pursuit law is used during the rest of the simulation.

Both the "A" aircraft and the "B" aircraft rapidly lose airspeed, but remain approximately co-altitude during the simulated engagement. This trend will be observed in other engagements starting co-altitude and not involving tail-chase evasion.

5.2.REGION 3

The initial conditions used ($\Lambda(B) = 135$ degrees, $\epsilon(B) = 90$ degrees) put AMLB at a significant disadvantage with an initial $PI(B) = 37$.

The trajectory, shown in Figure 5.4, exhibits flat scissors for the first 9 seconds of the engagement, during which the pursuit law is used. This part of the engagement is similar to the Region 2 case previously discussed. The F-15 has brief firing opportunities between $t = 8$ and 10 sec. Unlike the region 2 case, AMLB cannot initiate a second scissor and is forced instead into the tail-chase evasion mode after $t = 9$ seconds, which accounts for its fluctuations in altitude. AMLB has a slight speed advantage during the major part of the simulated engagement. In a real-life engagement, this might be exploited to disengage, a maneuver not included in the present AMLB disengagement maneuver.



TCE: Tail Chase Evasion
 P: Pursuit
 LGP: Lag Pursuit

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

Figure 5-4
 Region 3 ($\lambda(B)=135$ deg.,
 $\epsilon(B)=90$ deg.) X-Y Trace

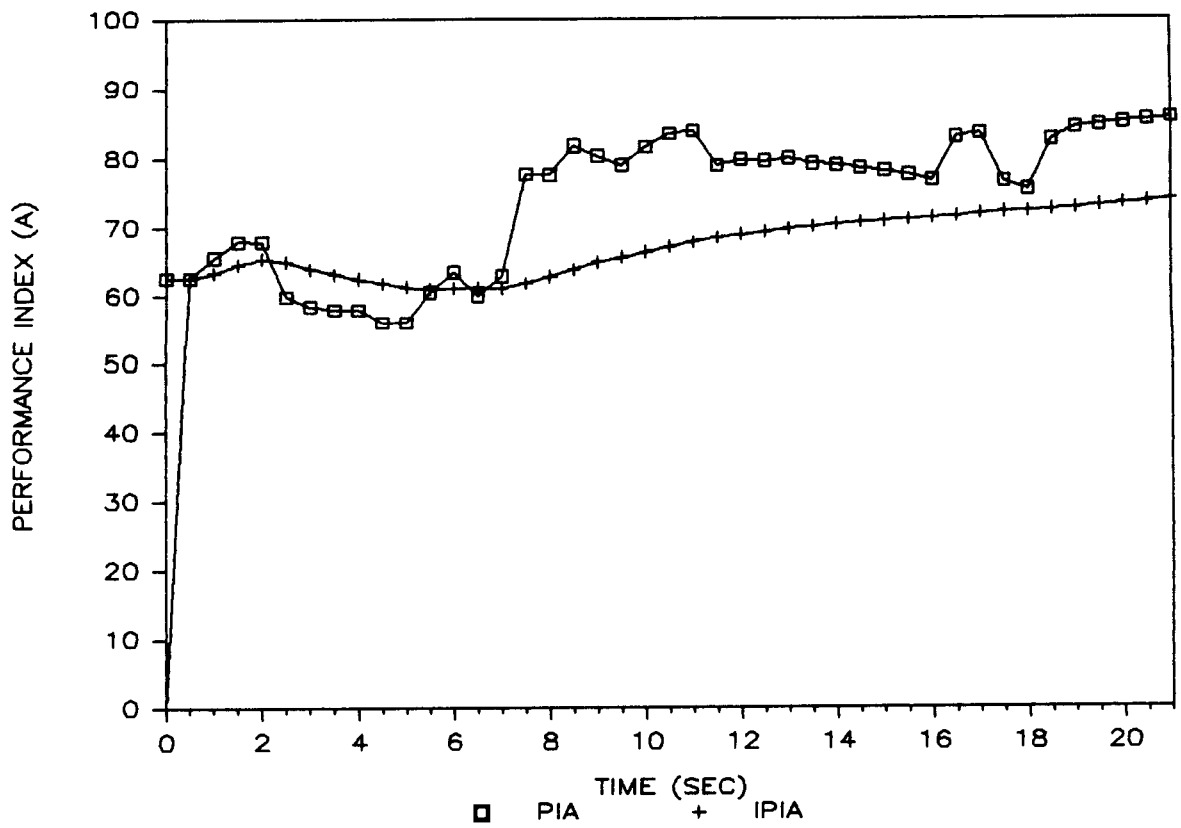
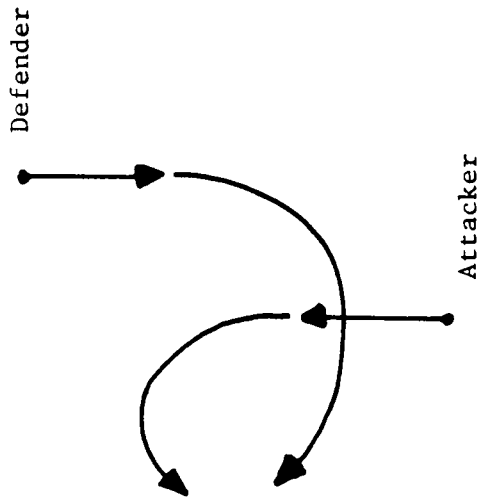


Figure 5-5 Region 3
PI Plot

5.3.REGION 4

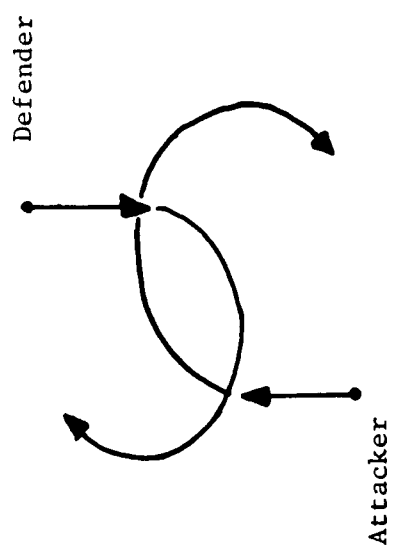
Referring to Table 5-1, the range of initial epsilon(B)-lambda(B) values considered in region 4 corresponds to forward-quarter passes. This means that the initial position of each aircraft is in the other's forward quarter. As indicated in Reference 10 (p77), there are two turn options available for fighters meeting in forward-quarter passes: the nose-to-nose turn option, and the nose-to-tail turn option. These are illustrated in Figure 5-6, adapted from Figure 2-11 in Reference 10. The terminology refers to the position of the fighters at the end of the maneuver. These options were compared to the results obtained with the AMLA/MLB logic in this series of four cases.

Case 1 is illustrated in Figure 5-7 and shows a nose-to-tail conversion generated by the present MLB. The instantaneous steering maneuver has also been indicated on Figure 5-7. "P" indicates that "B" follows a pure pursuit maneuver between $t = 0$ seconds and $t = 8$ seconds. "LGP" indicates that a lag pursuit maneuver is used between $t = 8$ seconds and $t = 21$ seconds (end of the simulated engagement.) In this particular case, the MLB steering law provides the F-4 with both a good defensive maneuver and a good maneuver for repositioning for attack. In contrast, the AMLA-controlled F-15 does not exhibit a repositioning tactic and seems instead to "wander off". The performance index plot (Figure 5-8) indicates that the situation evolves in favor of the F-4 from initially neutral conditions.



(b)

Nose-to-Nose



(a)

Nose-to-Tail

Adapted from Shaw, p. 78, Fig. 2-11

Figure 5-6
Forward quarter
turn options

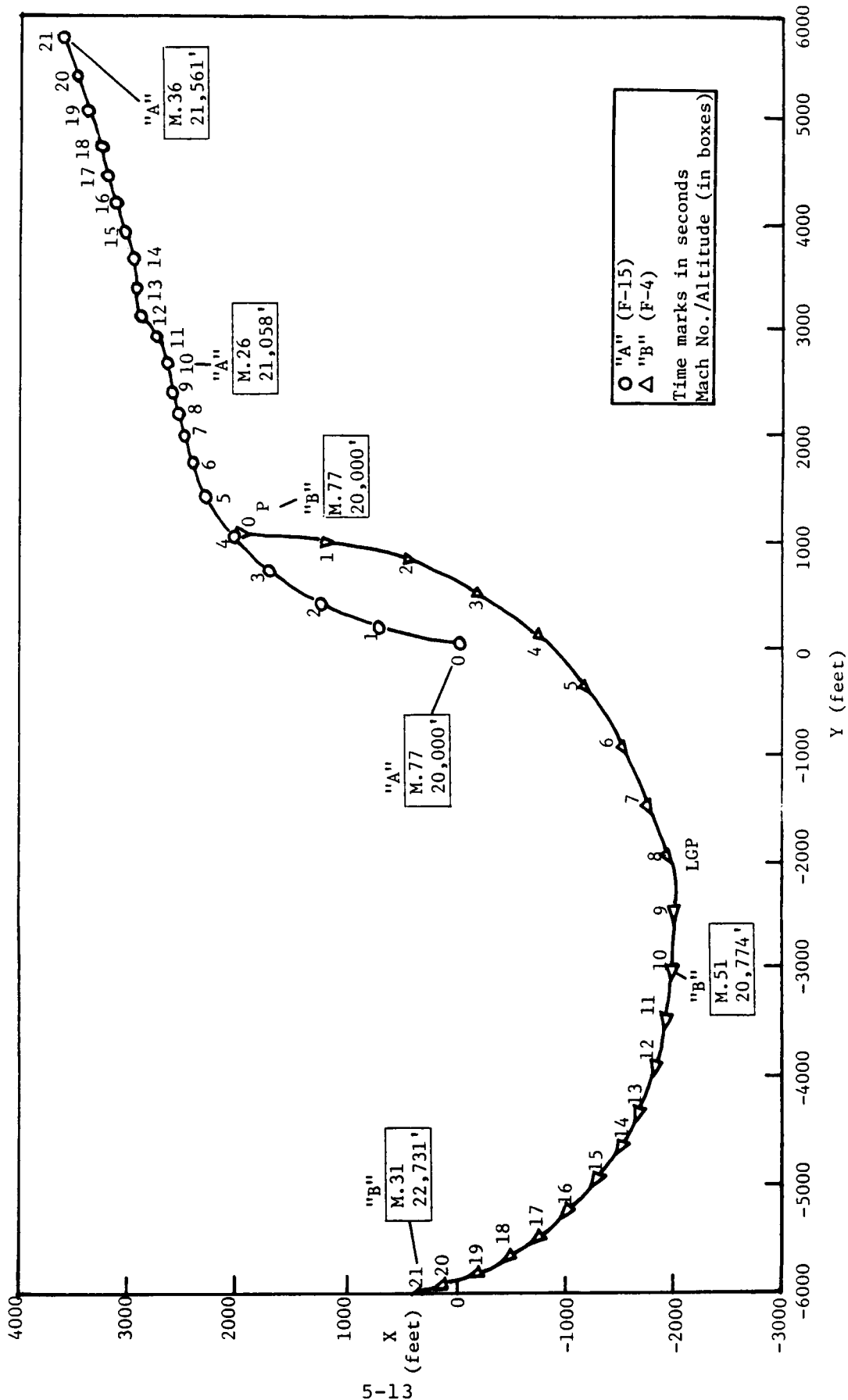


Figure 5-7 Forward Quarter Pass
 Case 1 ($\lambda(B)=26 \text{ deg.}$, $\epsilon(B)=135 \text{ deg.}$) X-Y trace

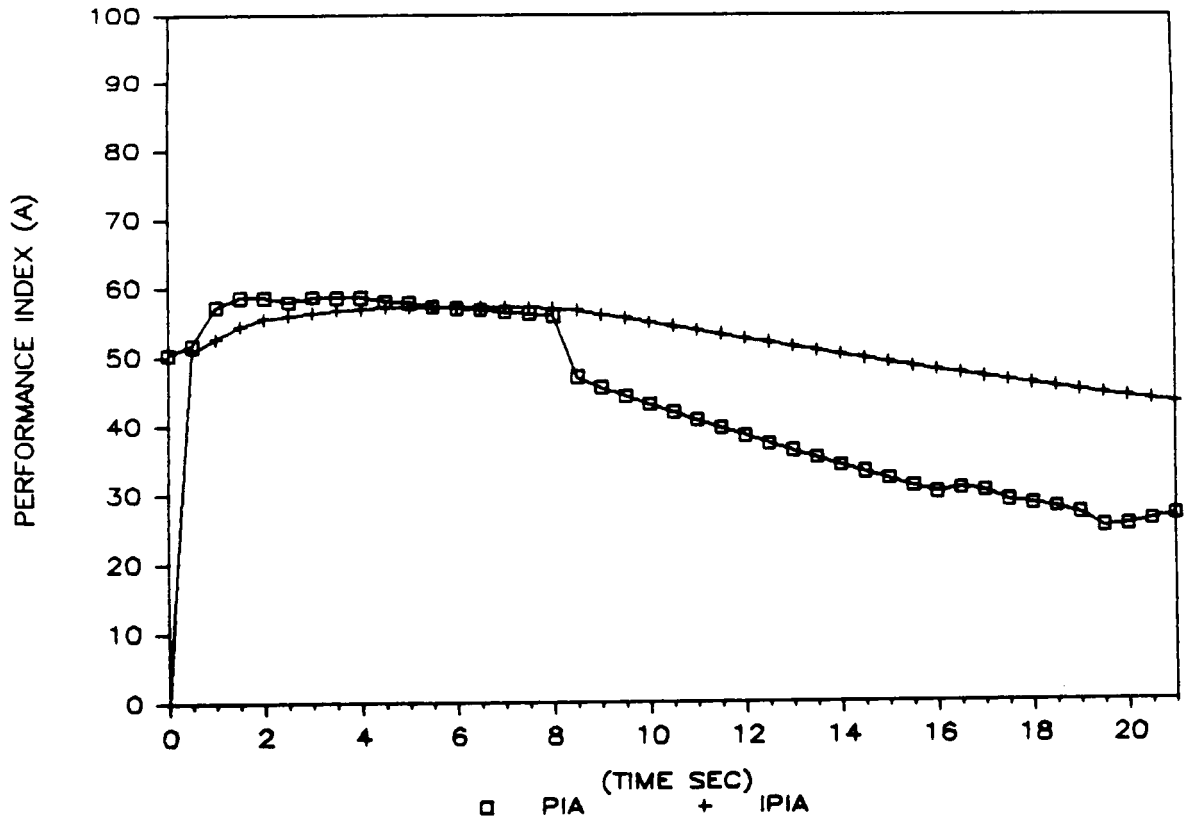


Figure 5-8 Case 1,
Region 4, PI Plot

Case 2 represents a slight variation in initial conditions compared to case 1: $\lambda(B) = 45 \text{ deg}$; $\epsilon(B) = 180 \text{ deg}$. It is illustrated in Figure 5-9. In contrast to the previous run, this results in a nose-to-nose conversion. This conversion mode offers the F-4 the potential for a subsequent head-on firing opportunity when the range closes to less than 3000 feet (but this is beyond the interval simulated).

The maneuvers used in Case 2 have also been indicated on Figure 5-9. Pursuit (P) is steered between $t = 0$ second and $t = 1$ second; between $t = 2$ seconds and $t = 12$ seconds; and between $t = 14$ seconds and 15 seconds. Lag pursuit (LGP) is steered between $t = 1$ second and $t = 2$ seconds; $t = 12$ seconds and 15 seconds; and between $t = 15$ seconds to the end of the simulated engagement.

The initial angular conditions for cases 3 and 4 also correspond to forward quarter passes, with slight variations in ϵ and λ compared to cases 1 and 2. However, the initial altitudes (10,000 feet) and speed (M.46) are very different. The initial speed was selected so that both aircraft start near corner velocity, the velocity at which both aircraft have their best turn performance. The X-Y traces for both cases (Figures 5-11 and 5-13) rapidly develop into well-defined "scissors". The effect of the F-15's smaller turn radius is apparent: "A" turns well within "B". However, in spite of this visible advantage, the PI plots for both cases indicate that "A" does not attain a gun-firing position. The situation remains

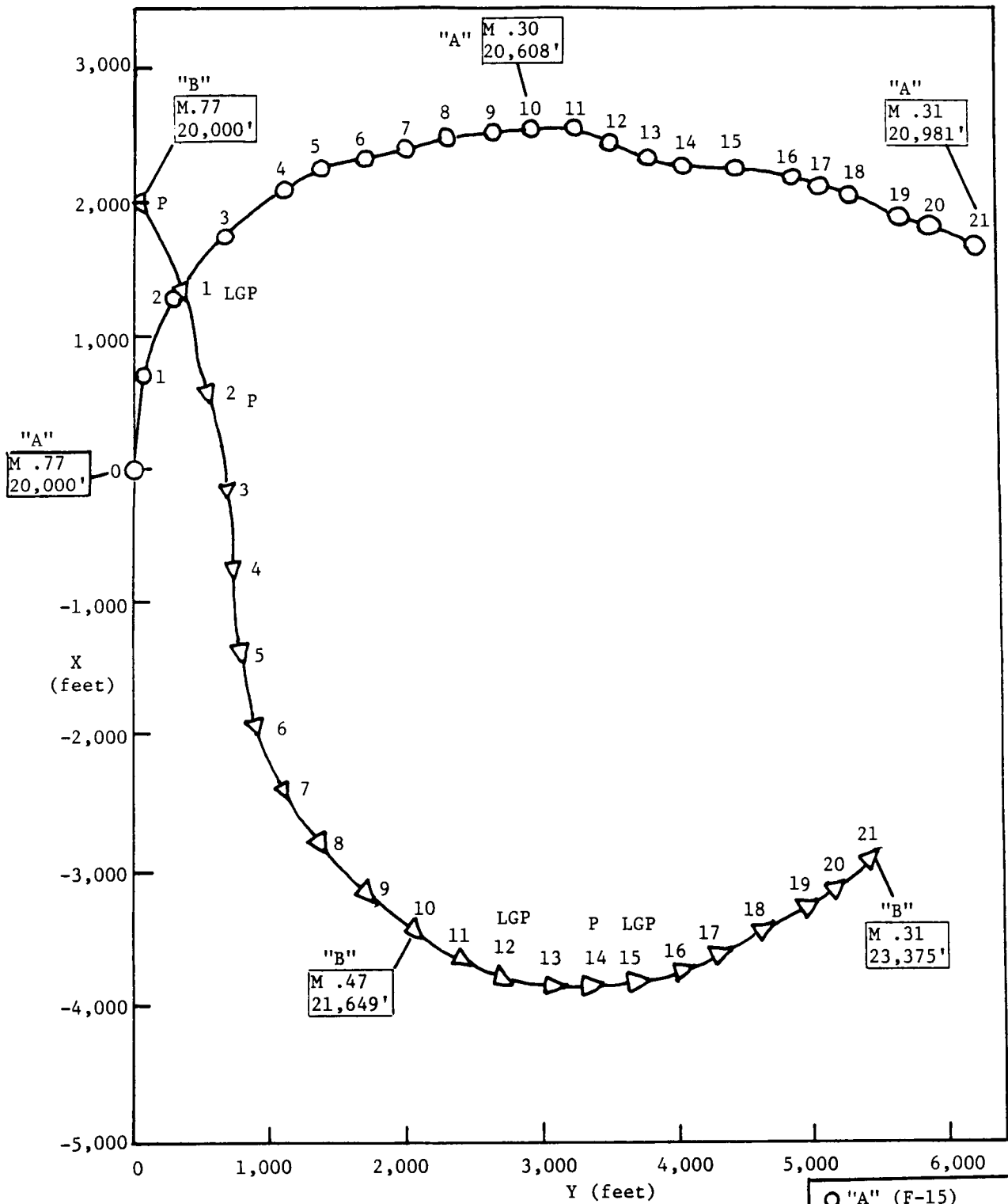


Figure 5-9 Forward Quarter Pass,
Case 2
 $\lambda(B)=45$ deg., $\epsilon(B)=180$ deg.)
X-Y trace

○ 'A' (F-15)
 △ 'B' (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

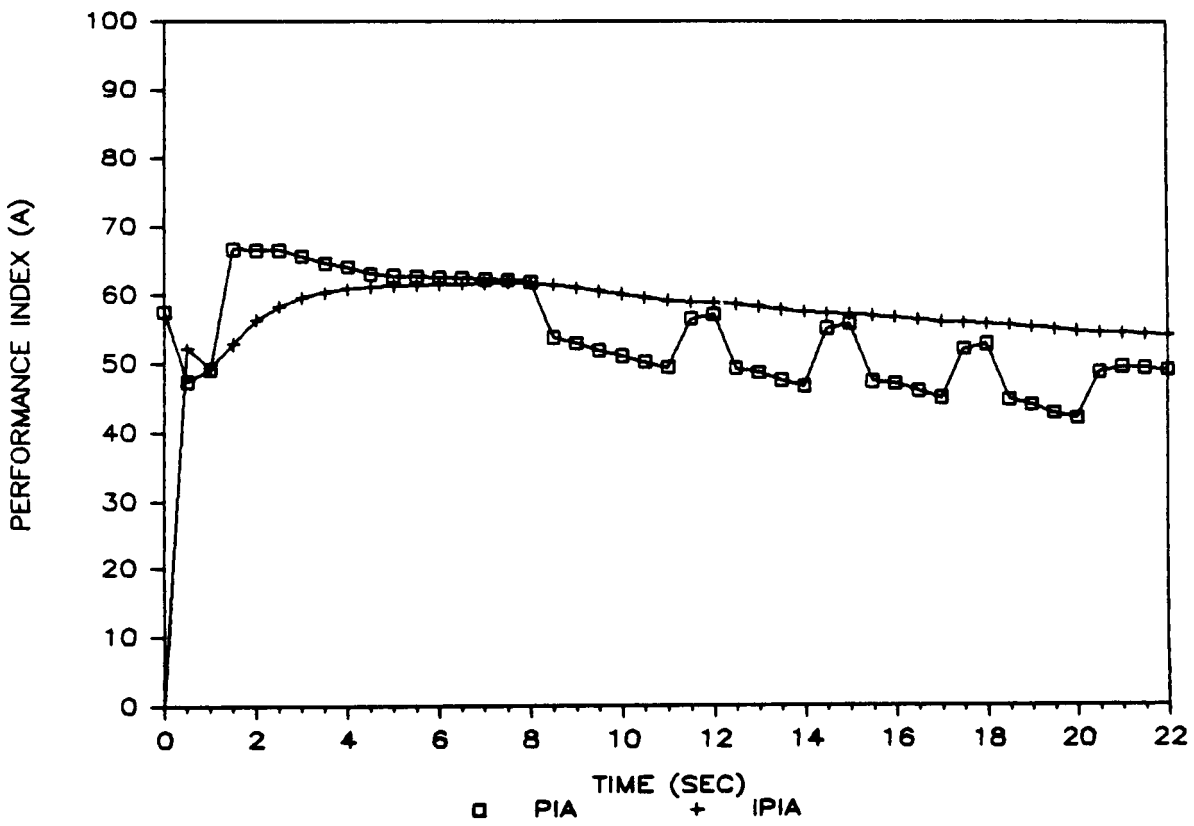


Figure 5-10
Case 2, Region 4,
PI Plot

essentially stalemated.

The steering laws have also been indicated on Figure 5-11 and 5-13. The pursuit law is used the most frequently, as would indeed be expected from the domains specified in Figure 3-2.

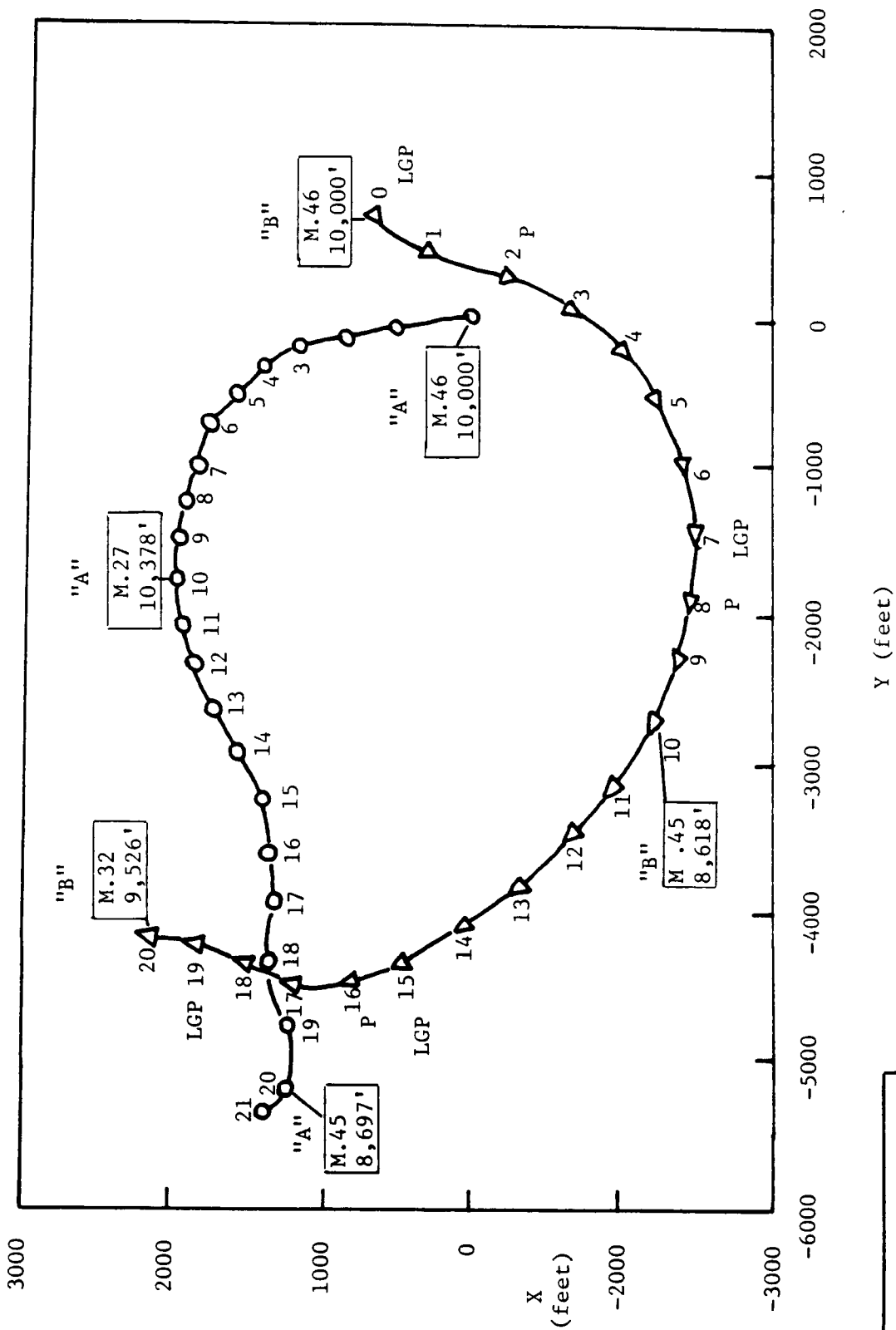


Figure 5-11 Forward Quarter Pass, Case 3
 ($\lambda(B)=0$ degrees, $\epsilon(B)=135$ degrees)
 X-Y trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude (in boxes)

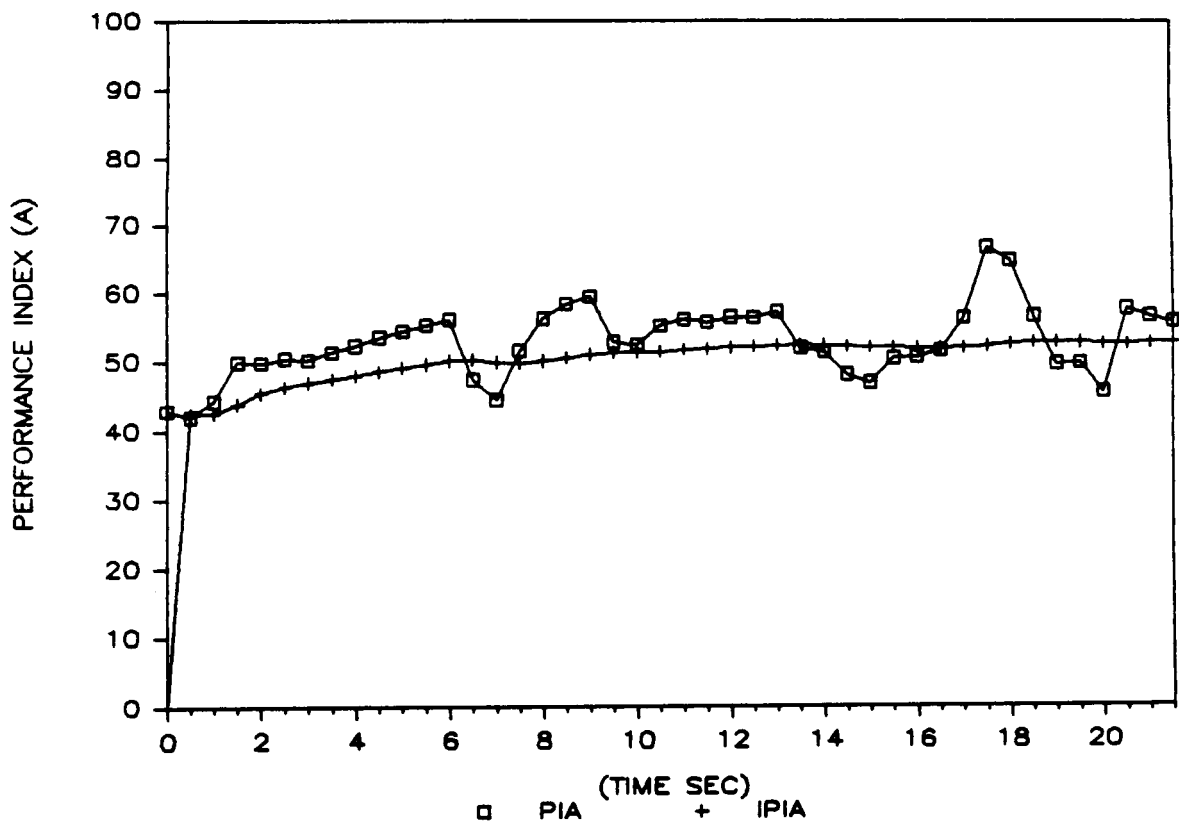
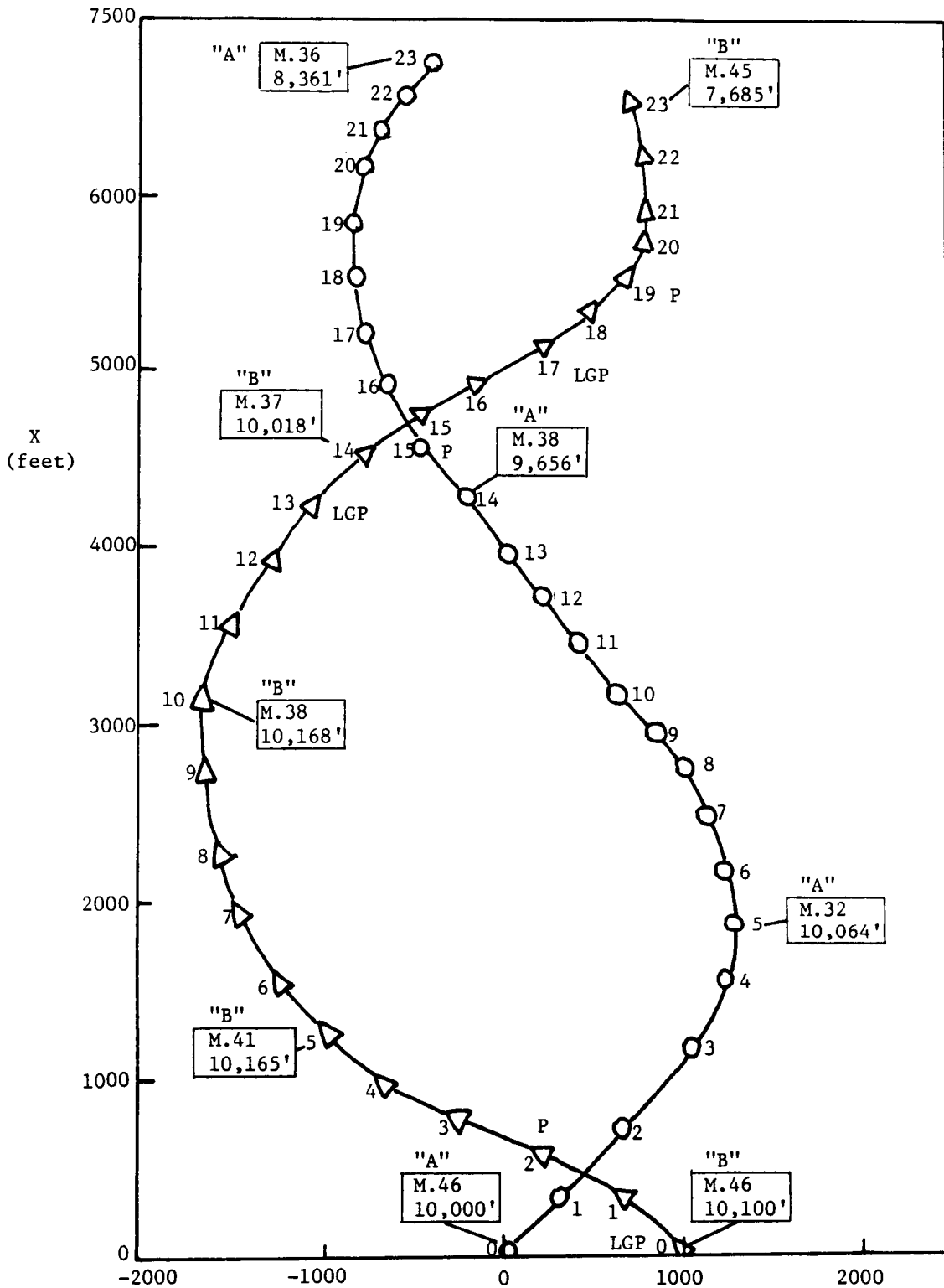


Figure 5-12
Case 3 PI Plot



○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

Figure 5-13 Forward Quarter Pass
 Case 4 ($\lambda(B)=45$ deg., $\epsilon(B)=135$ deg.)
 X-Y trace

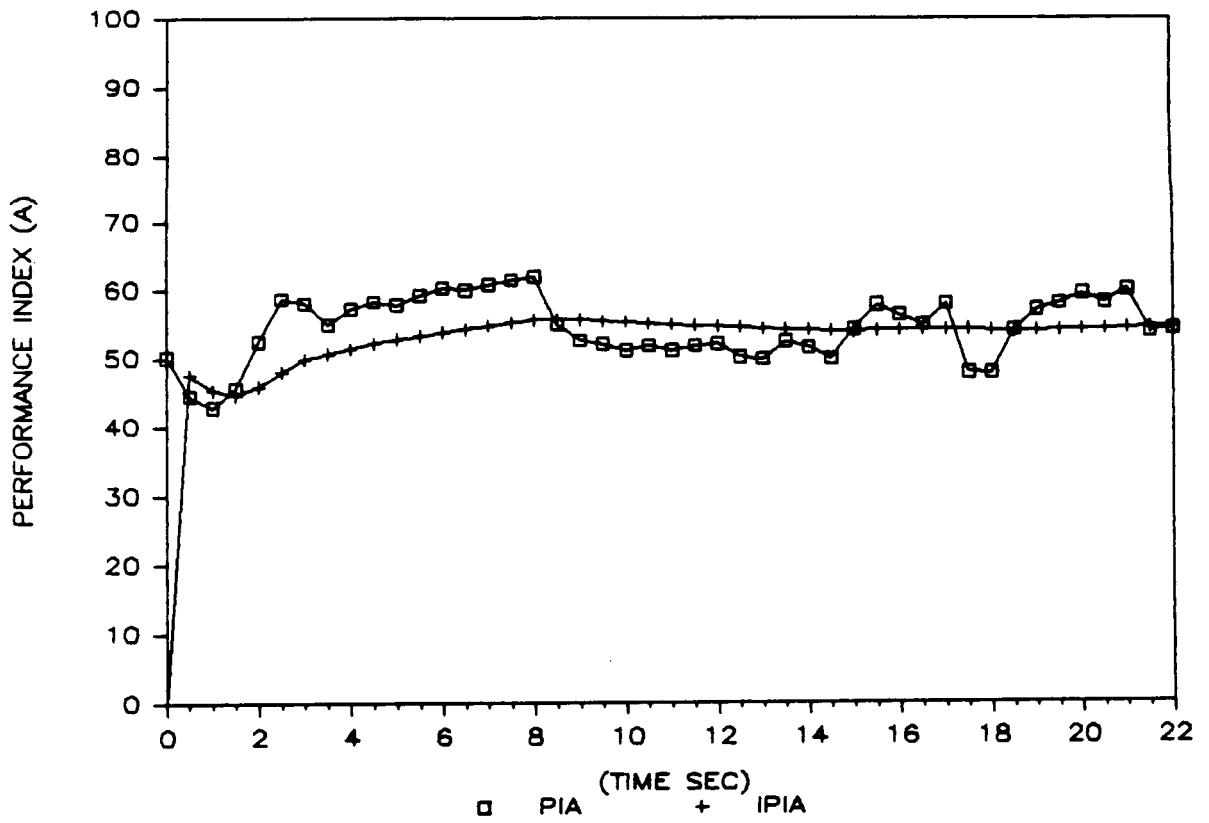


Figure 5-14
Case 4 PI Plot

5.4. REGION 5

A series of 4 runs were made in this region which corresponds to a slight angular advantage in favor of the F-15. In case 1 of region 5, shown in Figure 5-15, the F-15 eventually gains angular advantage after $t = 13$ seconds, but also loses more airspeed than the F-4E in the turn.

Case 2 of region 5 was run to highlight the influence of the turning ability of the F-15 on the result of an engagement with the same initial conditions as in Case 1. The thrust/weight ratio of the F-15 was reduced by increasing the weight from the nominal 40,000 lbs to an artificial 50,000 lbs, thus yielding a thrust-to-weight ratio of approximately 0.8 which is comparable to the F-4E. The results are illustrated in Figures 5-17 and -18. This run shows that the F-4E now has a firing opportunity between $t = 11$ and 12 sec.

Cases 3 and 4 illustrate the effect of an initial altitude difference on the same initial angular conditions as in Case 1. In case 3, "A" has an initial altitude advantage of 4,000 feet compared to "B". The results have been illustrated in Figures 5-19 and -20. The present AMLB logic commands a pursuit course with the aimpoint located at the altitude of "A". As a result, "B" rapidly loses altitude. The situation at the end of the simulated engagement shows that "B" ends up in a defensive position. Thus, the initial altitude advantage has not improved "B"'s situation.

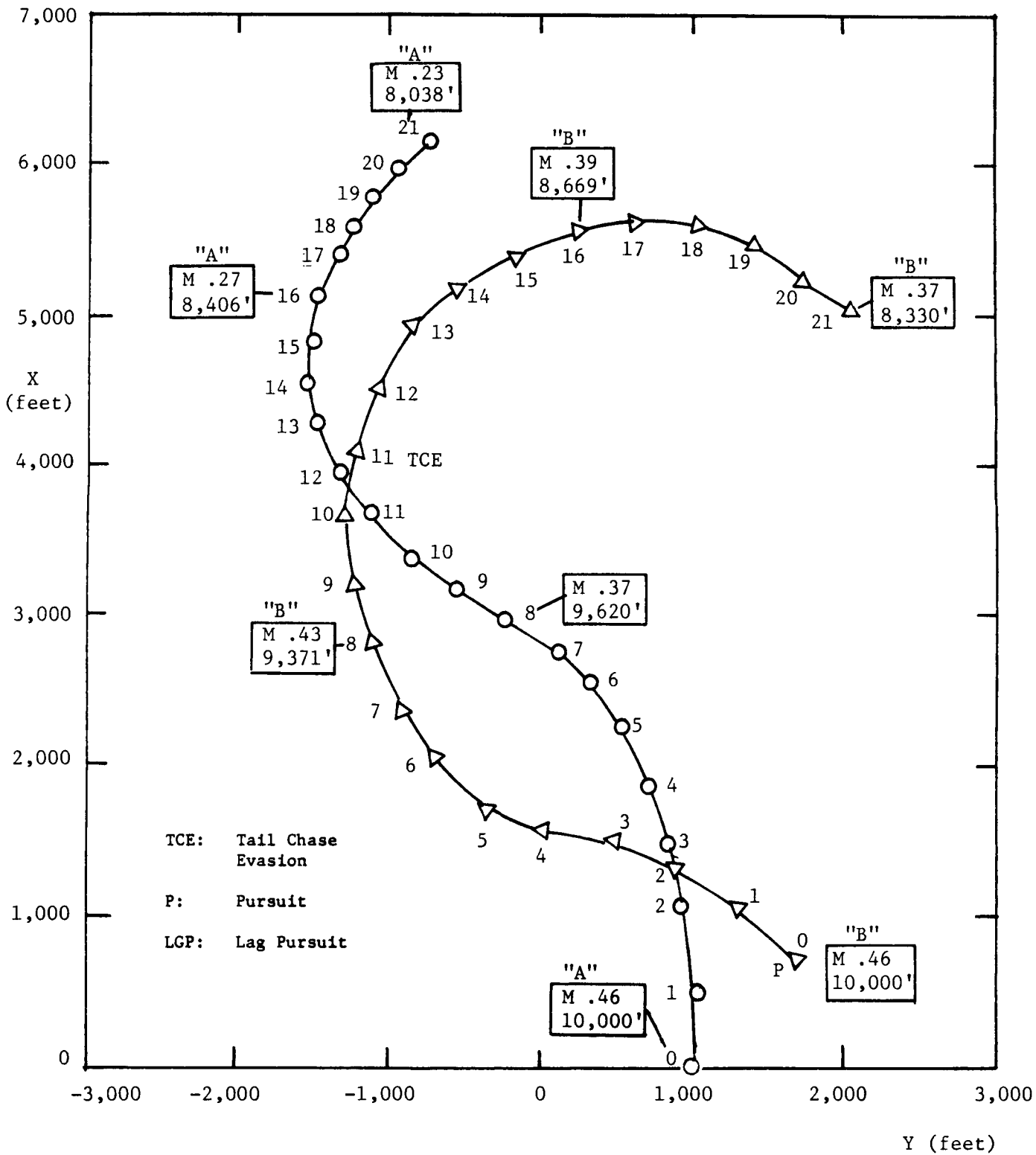


Figure 5-15 Region 5,
 Case 1 ($\lambda(B)=90$ deg.,
 $\epsilon(B)=135$ deg.)
 X-Y Trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

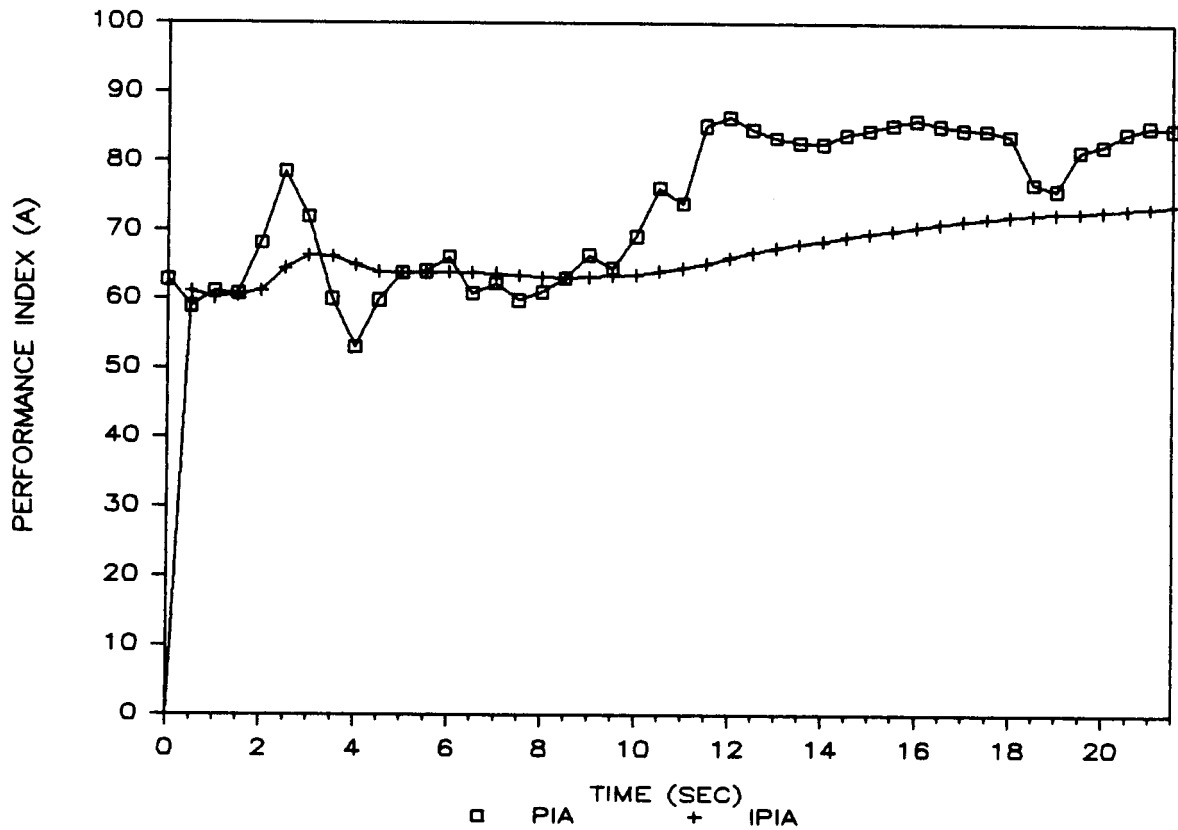


Figure 5-16 Region 5,
Case 1, PI Plot

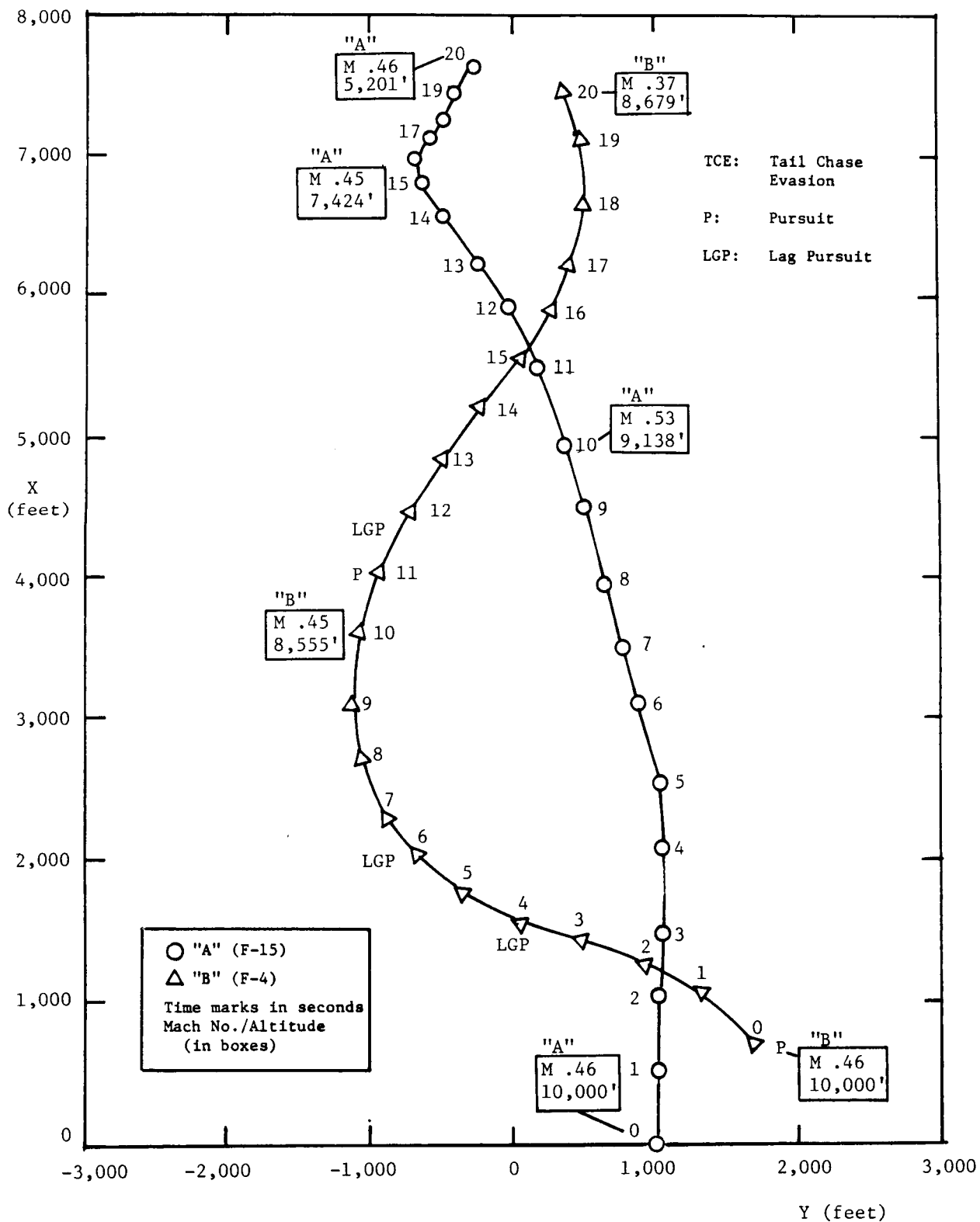


Figure 5-17 Region 5, Case 2
 $(\lambda(B)=90 \text{ deg.}, \epsilon(B)=135 \text{ deg.},$
 modified F-15) X-Y Trace

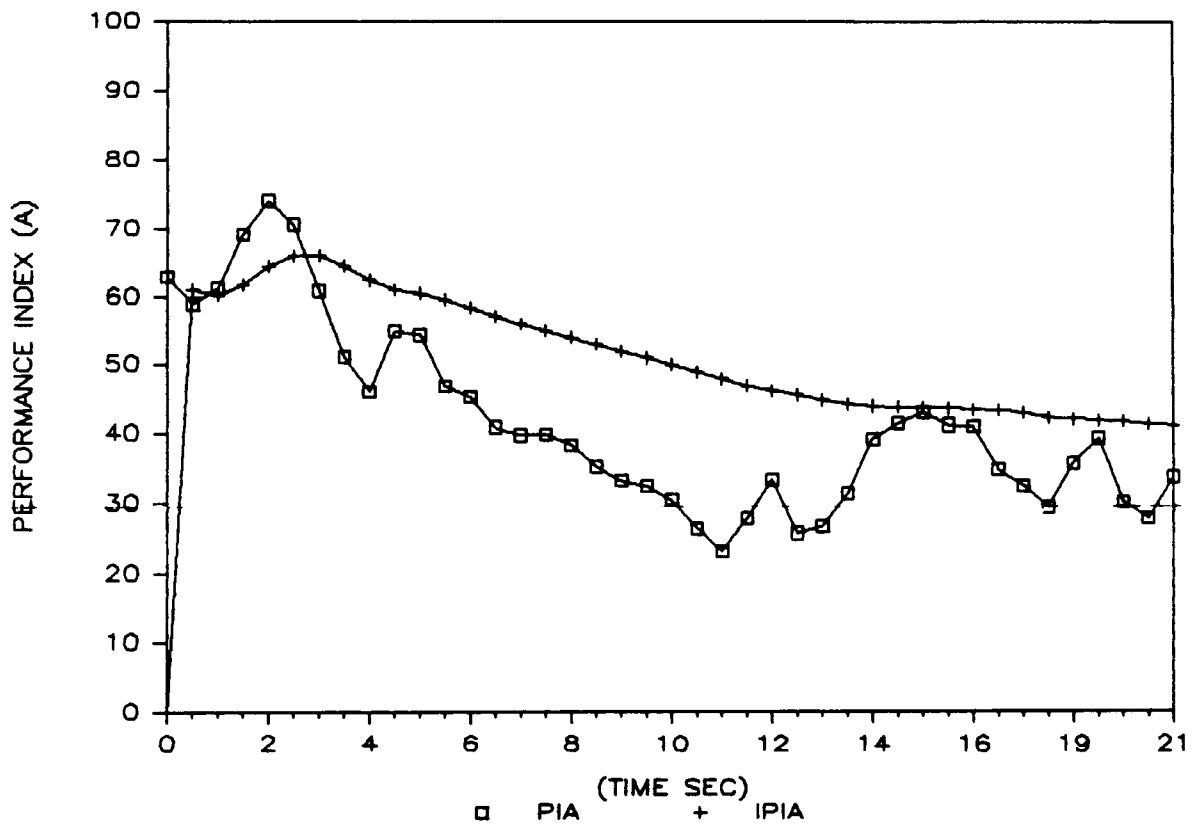


Figure 5-18 Region 5,
Case 2, PI Plot

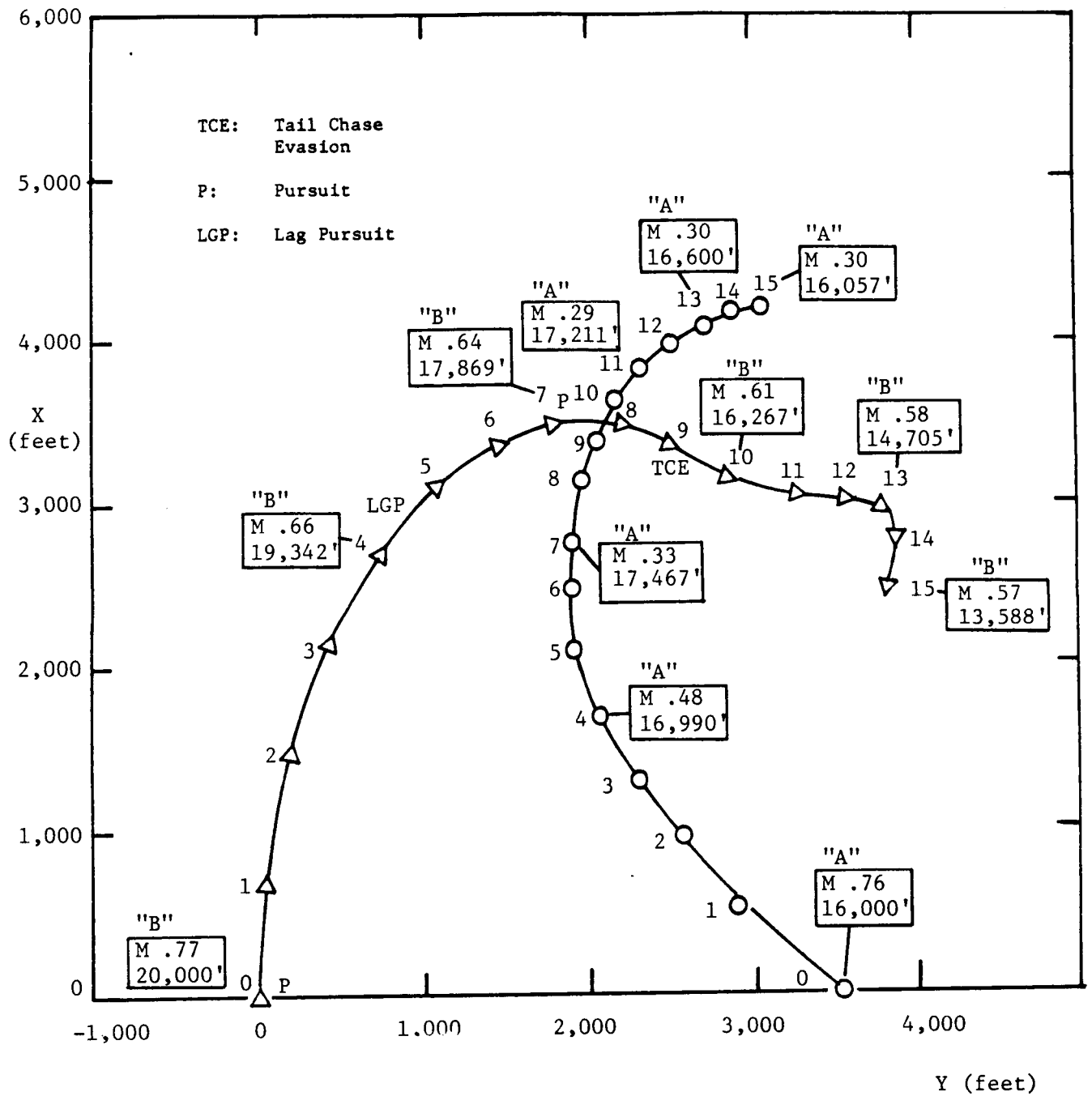


Figure 5-19 Region 5, Case 3
 ($\lambda(B)=90$ deg., $\epsilon(B)=135$ deg.,
 $Z(A)=16,000$ ft., $Z(B)=20,000$ ft.)
 X-Y Trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

The PI at the end of the simulated engagement (Figure 5-20) was much higher than one would expect at first from an examination of the X-Y trace. For this reason the line-of-sight and angle-off were plotted individually in Figure 5-21. A careful examination of the run shows that the "A" aircraft is strongly pitched down. This attitude explains the observed variation in these angles. This highlights the utility of the PI in summarizing the angular situation of the engagement.

Case 4 assumes an initial altitude disadvantage of 4,000 feet for "B". In this case, illustrated in Figure 5-22, the AMLB logic commands a climbing turn in "A"'s direction, resulting from the pursuit law which is used between $t = 0$ seconds to 7.5 seconds. This maneuver brings "B" in "A"'s forward quarter, but the PI plot in Figure 5-23 shows that "A" does not have a firing opportunity as a result of the altitude difference. The engagement ends up with "A" overshooting "B", without "B" having a gun-firing opportunity due to the altitude difference. Following the overshoot, the F-15 does not appear to be reacting.

In both cases 3 and 4, both "A"'s and "B"'s tactics could be improved by the inclusion of negative G's.

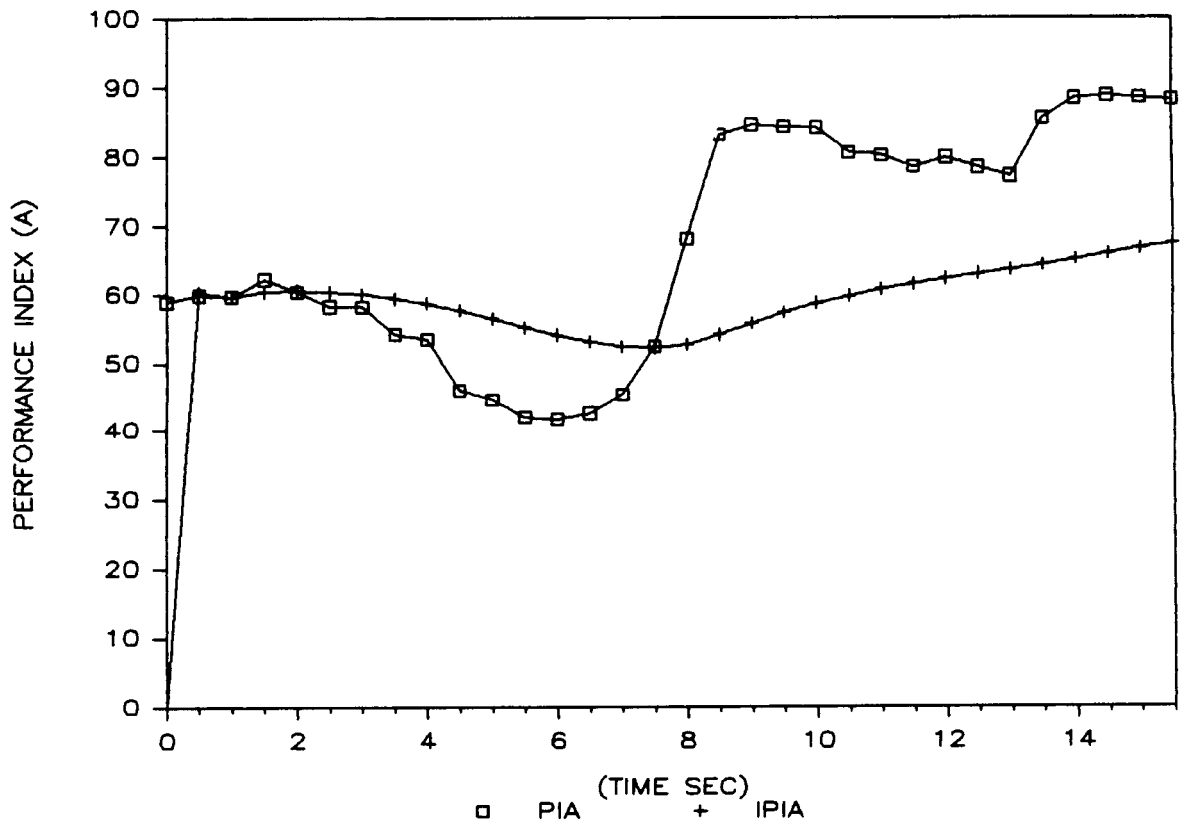


Figure 5-20 Region 5,
Case 3, PI Plot

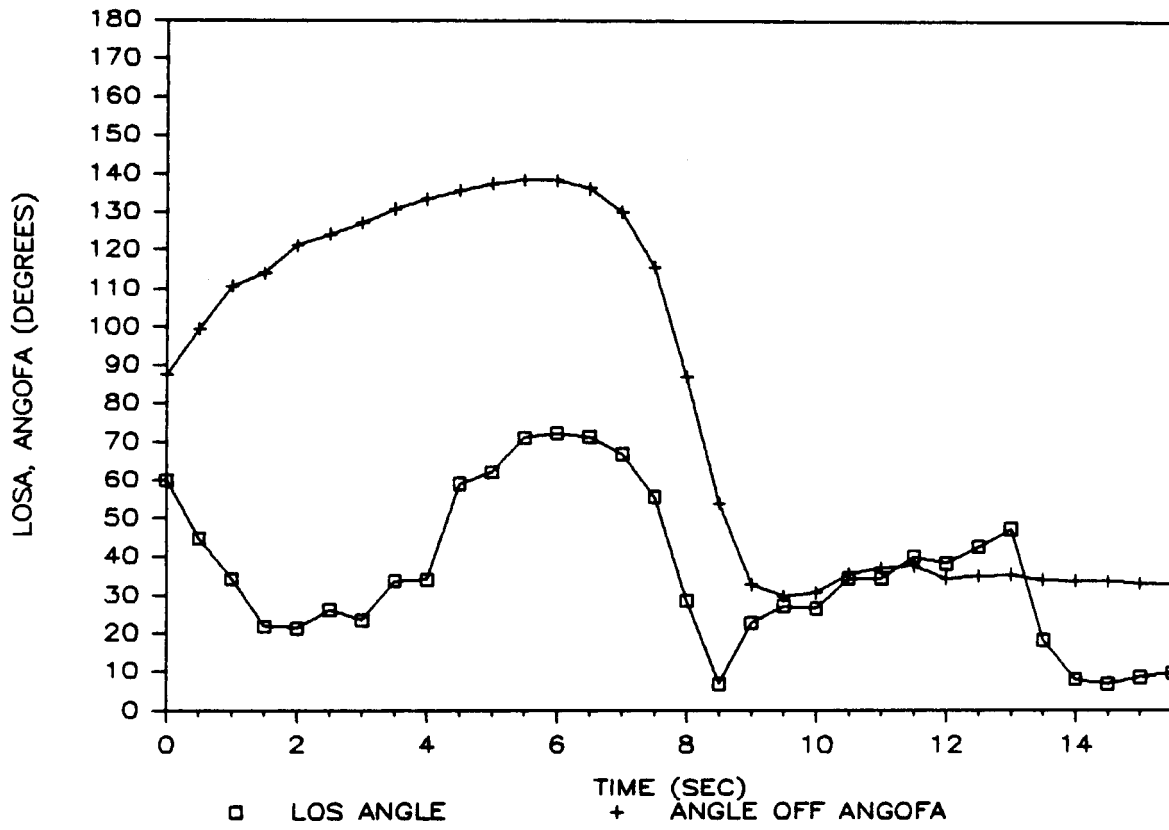


Figure 5-21 Region 5, Case 3
Line-of-Sight and Angle-off Plot

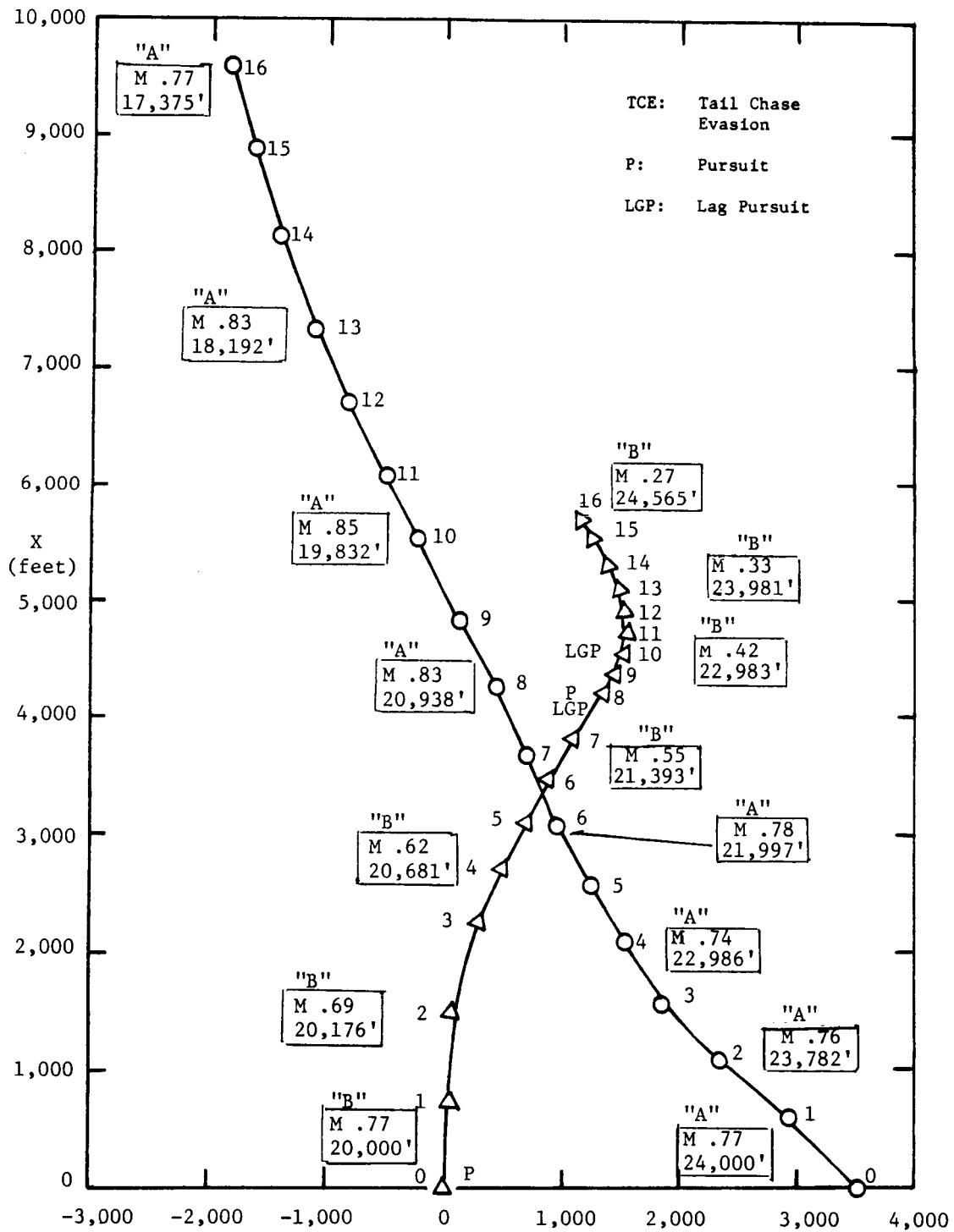


Figure 5-22 Region 5, Case 4
 $(\lambda(B)=90 \text{ deg.}; \epsilon(B)=135 \text{ deg.})$
 $Z(A)=24,000 \text{ ft.}; Z(B)=20,000 \text{ ft.})$
 X-Y Trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

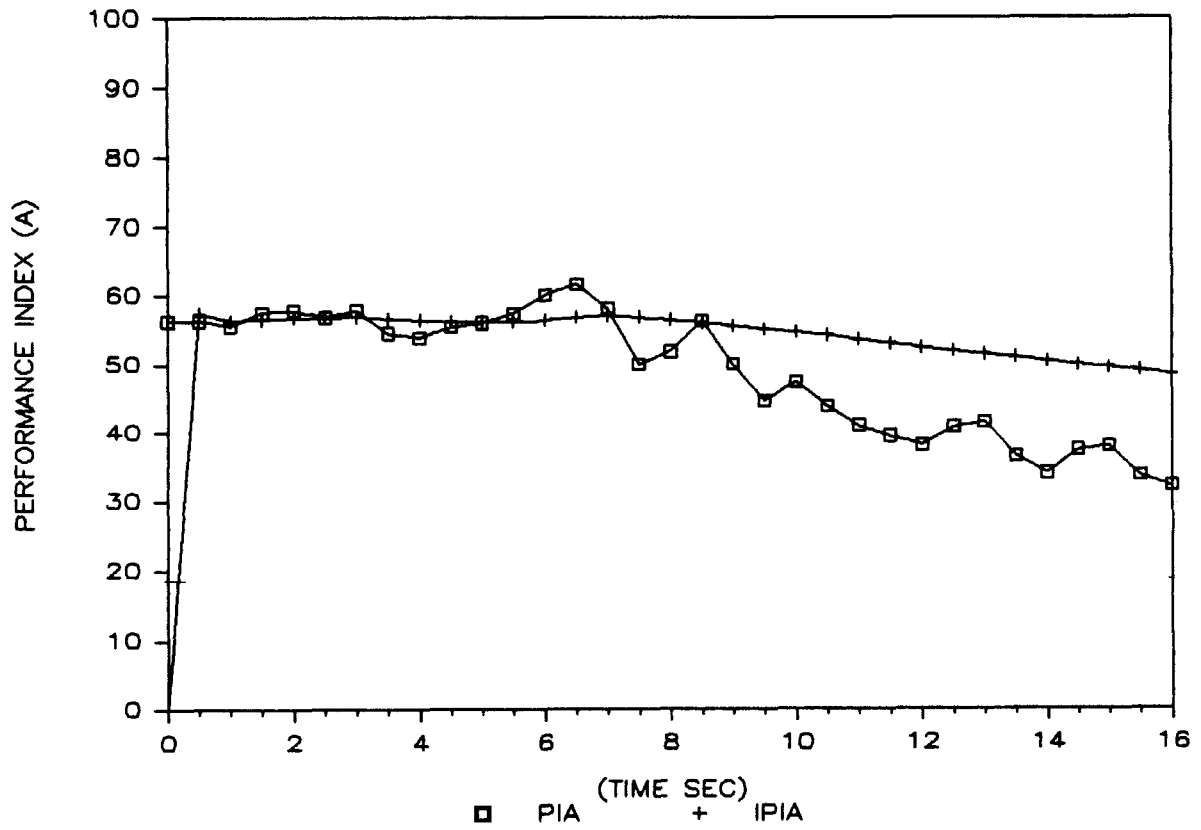


Figure 5-23 Region 5,
Case 4, PI Plot

5.5. REGION 6

This region corresponds to a severe initial angular disadvantage for "B". The case illustrated in Figures 5-24 and 5-25 exercises the AMLB evasive maneuver during the entire simulated engagement. The PI plot indicates that there is no improvement in "B"'s angular position. However, the X-Y trace indicates that the relative range increases from an initial 1000 feet to 3500 feet. This is due to "A"'s rapid loss of airspeed during the turn. This result shows suggests that "B" might have an opportunity to disengage.

5.6. CONCLUSIONS

In all cases, a wide difference in outcomes has been observed for small variations in the initial angular conditions. This result has often been observed in ACM simulations. Furthermore, this wide difference in outcomes occurred in spite of a small variation in initial PI. The use of the initial PI to classify and predict the entire engagement outcome does not appear promising.

In all cases, both AMLA and AMLB command initially high load factors which result in a rapid loss of airspeed. In all cases starting co-altitude, the fight remains roughly in the initial horizontal plane unless tail-chase evasion is initiated. The lag-pursuit and pure pursuit laws involve essentially a series of level turns, or an "angles fight" to use the terminology of Reference 10. In this fight, the F-4 cannot gain an advantage due to its lower turning capability compared to the

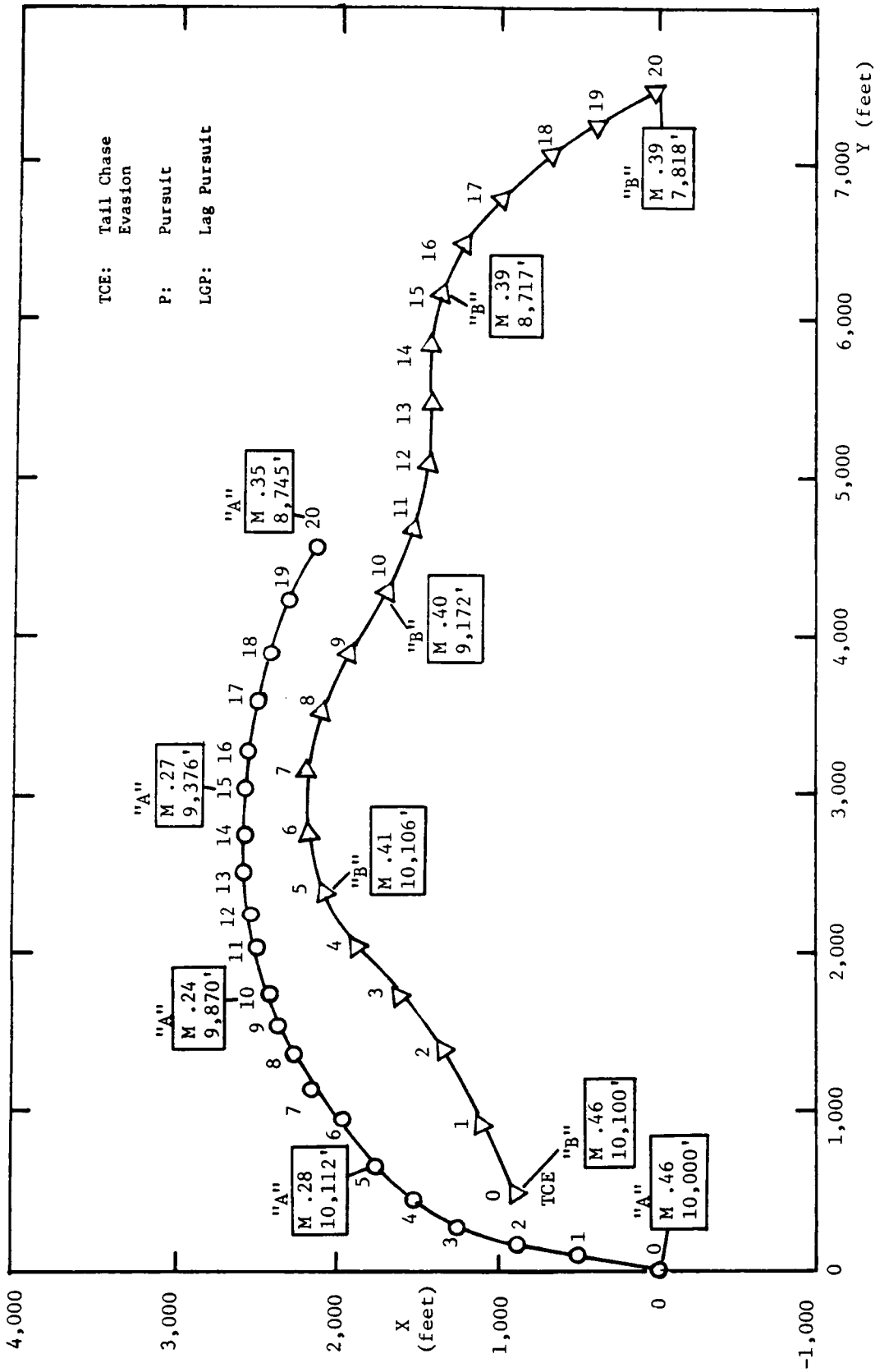


Figure 5-24 Region 6
 ($\lambda(B)=135$ deg., $\epsilon(B)=135$ deg.)
 X-Y Trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude
 (in boxes)

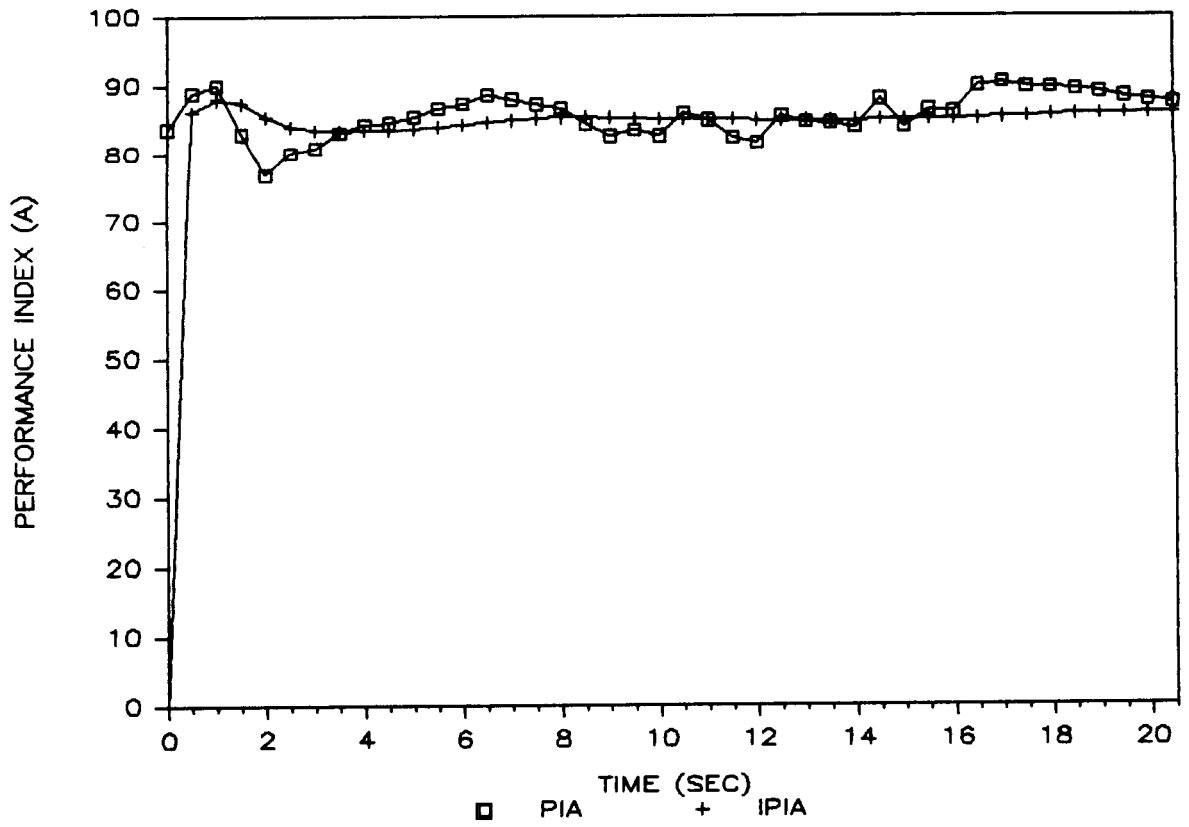


Figure 5-25 Region 6
PI Plot

F-15. In many cases, surprisingly, the fight does not rapidly evolve to the disadvantage of the F-4, as might be expected from the disparity in performance, and remains approximately neutral. Finally, it will be noted that the cases investigated did not present any opportunity to exercise the "pointing algorithm" dsicussed in Section 3.

Throughout the history of air combat, skilled pilots have been able to win engagements in spite of having the lower-performing aircraft. In this situation, they would avoid a turning fight as simulated above. In the next section, we describe an alternative approach to angular conversion which attempts to trade off altitude to gain an angular advantage.

6. ADDING BASIC FIGHTER MANEUVERS TO THE IF => THEN LOGIC

The maneuvers generated by the IF => THEN AMLB and discussed in Section 5 are realistic and have generally enjoyed good pilot acceptance in flight simulators. However, the predictability of the maneuvers it generates has been criticized because it makes it possible for a person to anticipate AMLB's future maneuvers after a few sessions in a simulator. To enrich the variety of maneuvers generated by AMLB, additional maneuvers based on the "basic fighter maneuvers" (BFMs) of the type found in ACM training manuals (for example, references 10, 11, 12) were added to the existing AMLB logic. It will be recalled from Section 1 that such an approach had been rejected at the time of the development of the original AML program. However, BFMs were used in the present effort because they improve the variety of maneuvers generated by AMLB, not only in flight simulators, but also against AMLA in offline programs, and has proven useful in these respects.

In examining samples of such BFMs, it was found that in general each of these maneuvers is appropriate under a narrow set of circumstances based primarily on relative geometry, and additionally on other situational parameters such as closing velocity, relative airspeed, relative altitude, to name just a few. While all these maneuvers have their individual, specific objective, the majority of them share with AMLB the underlying purpose of angular conversion on the opponent, except in the case of the disengagement maneuver, which does not exist in AMLB. This observation suggested keeping the underlying AMLB angular

conversion logic, and to replace it only when an opportunity for a BFM arises. To reduce predictability, when a choice between a BFM and AMLB is available, the selection of the maneuver is decided by the means of a random number.

Due to the limited scope of this work, only a set of three BFMs was investigated: a "diving overshoot", a "vertical overshoot", and "opposite turn". This terminology and the results obtained will be discussed in detail later in this section.

The features discussed above were implemented in a new subroutine called SELECTB. The particular requirements for the BFMs were (1) to identify when a particular BFM can be executed; (2) to execute that maneuver for a specified amount of time, or until conditions specific to that BFM are no longer met, and (3) to terminate the maneuver under the specified conditions and return control to the underlying AMLB. Function (1) is presently performed in a added subroutine called SELECTB, while functions (2) and (3) are performed in individual subroutines, as illustrated in Figure 6-1.

SUBROUTINE SELECTB

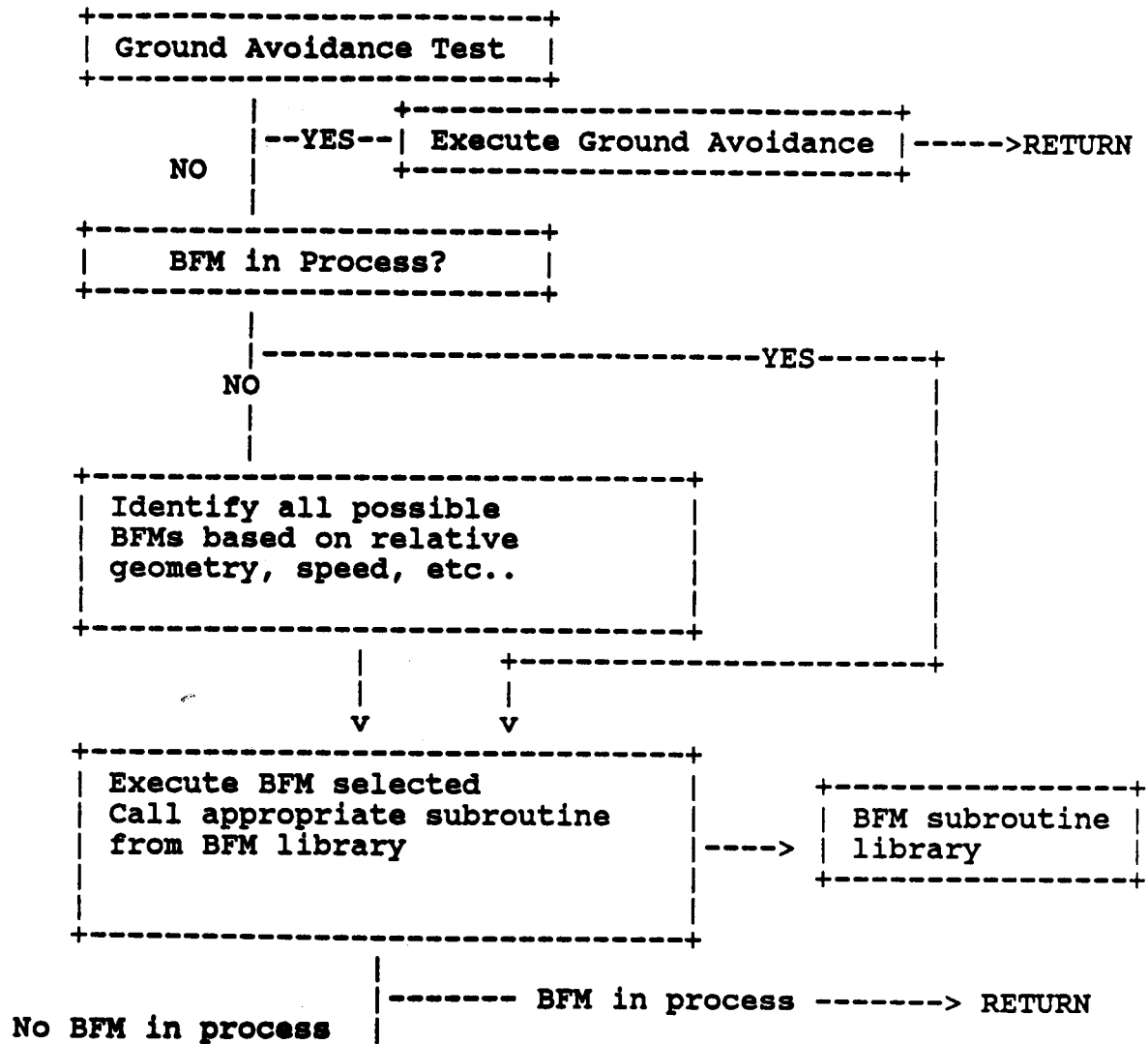


Figure 6-1 OUTLINE OF THE IMPLEMENTATION OF BASIC FIGHTER MANEUVERS

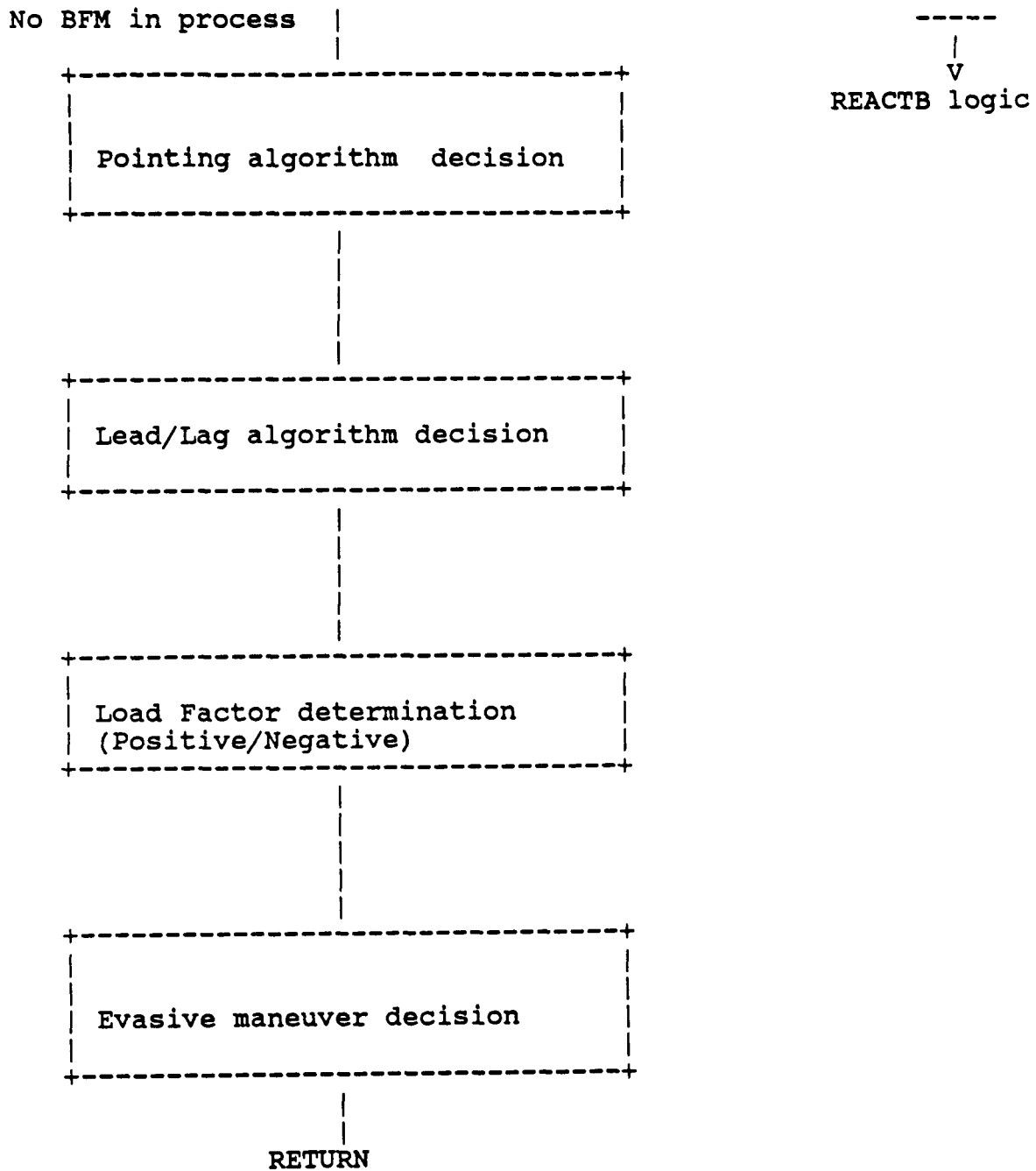


Figure 6-1 Concluded

6.1 OPPOSITE TURN

Several cases of "forward passes" were discussed in Section 5. It was found that the AMLB logic would generate either a nose-to-nose conversion, or a nose-to-tail conversion, depending on the initial conditions. Reference 10 (p 79) outlines the potential advantage of a nose-to-nose turn, which could result in achieving offensive advantage. To execute such a maneuver in Case 1 of section 5 would require "B" to turn away from his opponent, as was illustrated in Figure 5-1. This maneuver was implemented in SELECTB as the "opposite turn" BFM.

Sample results are shown in Figure 6-2 and 6-3. The effects of the superior turning ability of the F-15 over the F-4 is clearly demonstrated by the AMLA-controlled F-15 performing a "turn reversal" to gain a firing position on the F-4. This possibility of this situation is in fact predicted in Reference 8, Figure 2-12. In this case, AMLA derives the "textbook solution". The F-4 does not gain anything by performing an opposite turn with the initial conditions considered.

To underscore the role of the turn rate, another case was run with the same initial conditions as in case 1, but now with the F-E initially at corner velocity (M .41), and the F-15 remaining at the same initial velocity of M .77 . Case 2 has been illustrated in Figures 6-4 and 6-5. The F-E clearly achieves a nose-to-nose offensive position against the F-15. The F-15 also achieves an offensive position.

The results of these cases would encourage us to consider

the differential turn rate between the two aircraft at the time of the decision for the next maneuver. The implementation of this feature would be fairly simple from a computational standpoint. Furthermore, AML presently does not differentiate between a "defensive" aircraft and an "offensive" aircraft. The inclusion of the differential turn rate might be a good way to do so.

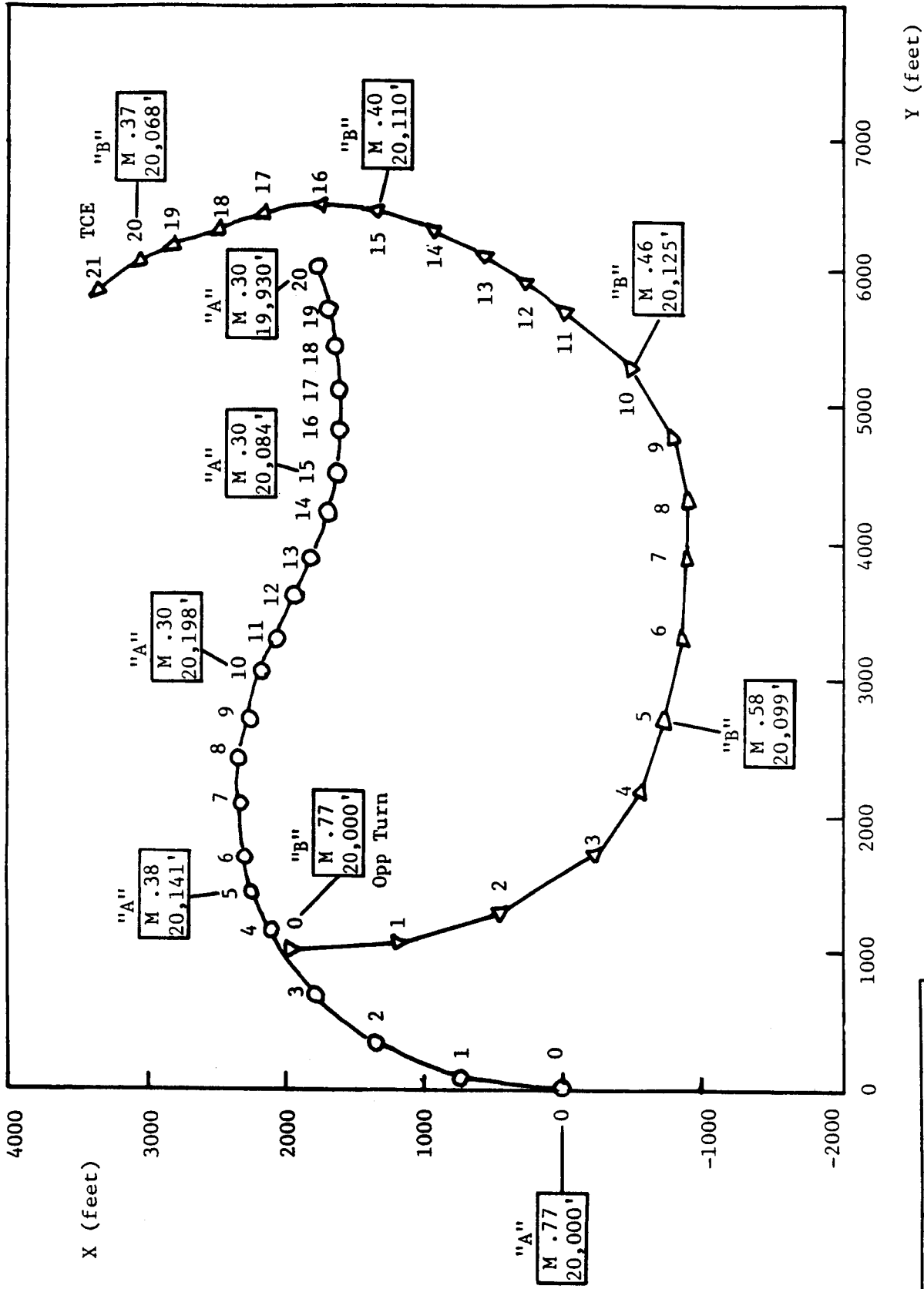


Figure 6.2
Opposite Turn Case 1
X-Y Trace

"A" (F-15)
"B" (F-4)
Time marks in seconds
Mach No./Altitude (in boxes)

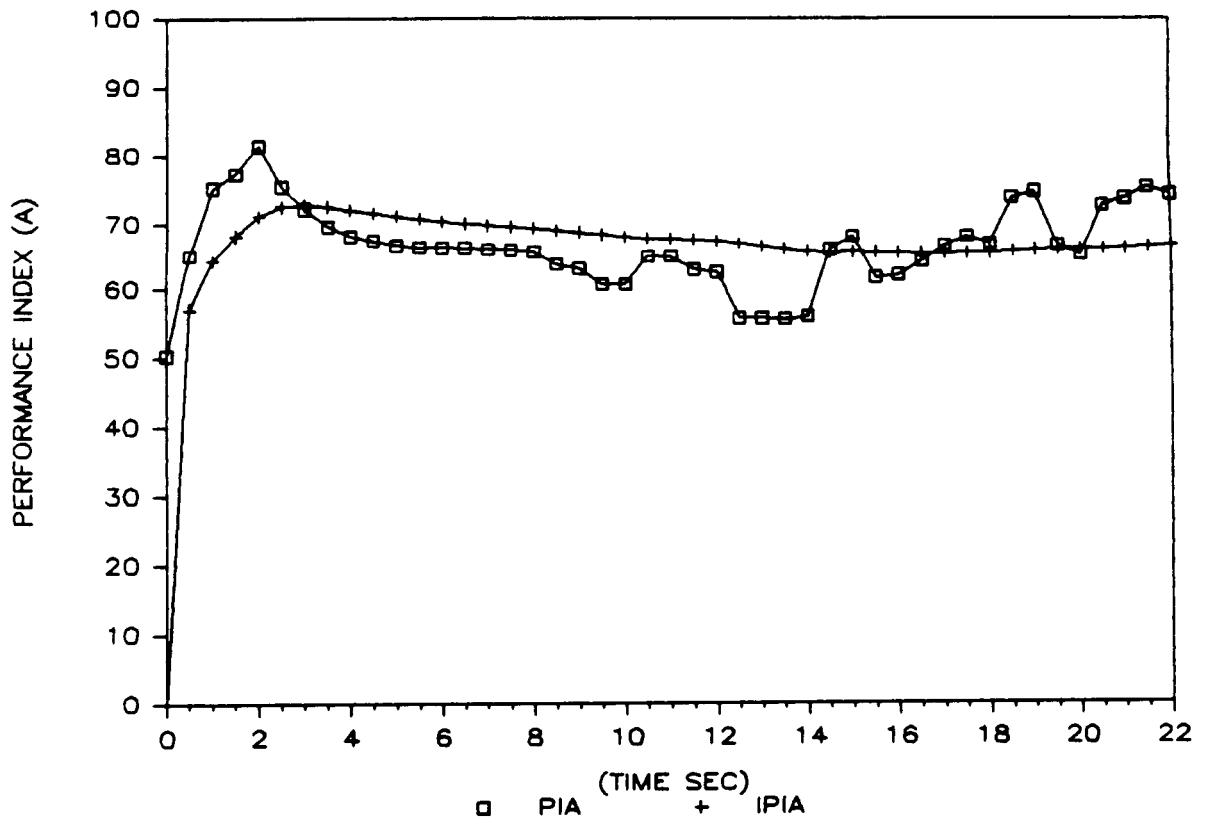


Figure 6-3 Opposite Turn,
Case 1 PI Plot

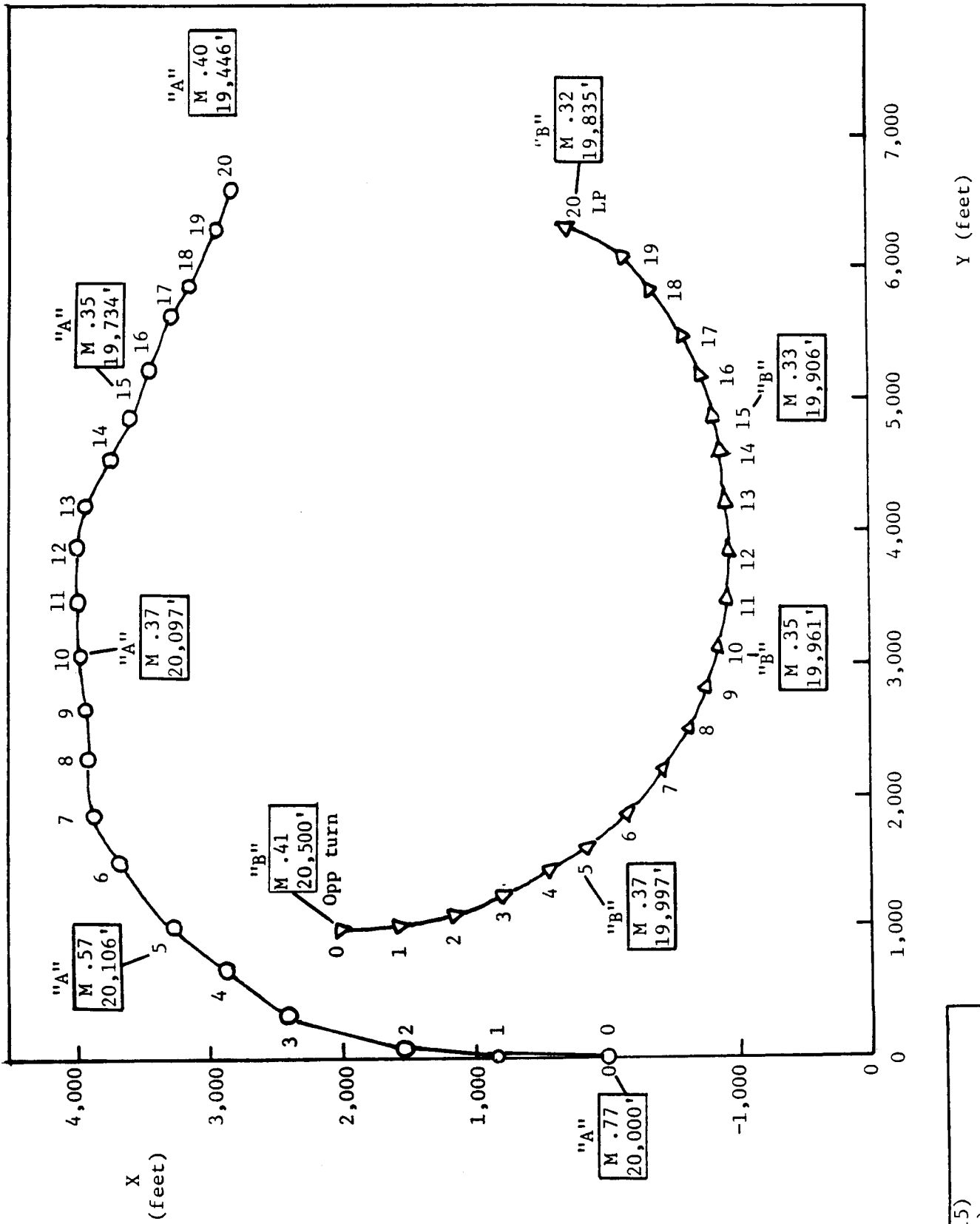


Figure 6-4. Opposite turn
Case 2, X-Y trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude (in boxes)

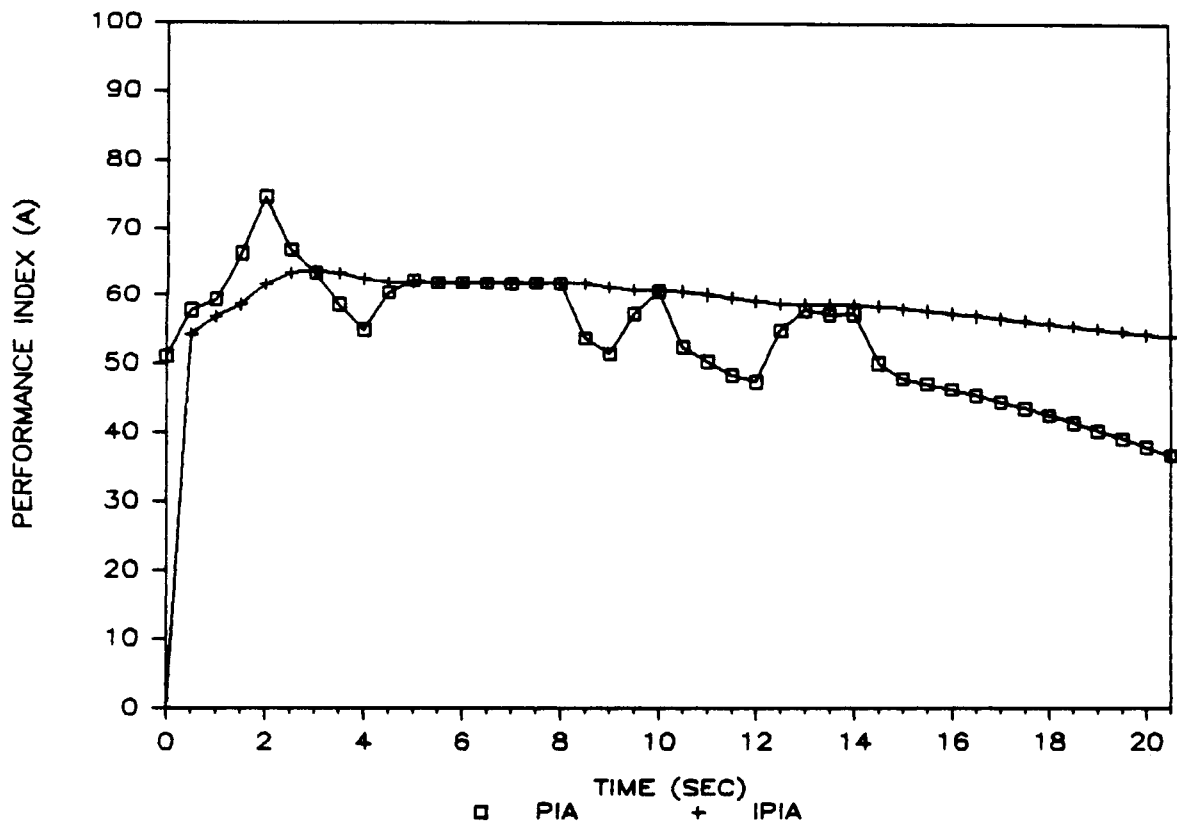


Figure 6-5 Opposite turn,
Case 2 PI Plot

6.2 DIVING OVERSHOOT

This BFM is a gun defense maneuver described in Reference 10 (pages 26-27). The defending aircraft dives in order to force the opponent to overshoot. For brevity, it is referred to in this document as a "diving overshoot". It is activated under the following conditions:

1. $120 \text{ deg} \leq \text{Epsilon}(B) \leq 180 \text{ deg}$ AND
2. $60 \text{ deg} \leq \text{Lambda}(B) \leq 120 \text{ deg}$ AND
3. $\text{ABS}(ZEA - ZEB) \leq 1000 \text{ feet}$ (i.e. approximately co-altitude), AND
4. $\text{Relative Range} \leq 6000 \text{ feet}$

It is implemented in subroutine OVRSH. The diving effect is accomplished by steering the B aircraft on a pursuit course which uses an aimpoint 10,000 feet below the A aircraft, i.e. with coordinates XEA, YEA, ZEA + 10000 (z positive downwards). The commanded load factor is 95% maximum. In spite of the dive, the commanded throttle setting is A/B because of the anticipated speed loss due to the high-G turn. Due to the anticipated loss of altitude, the maneuver can only be executed above a minimum altitude.

This maneuver was only tested against a non-interacting "A" aircraft, i.e. flying straight and level. The results are shown in Figures 6-6 and 6-7 (Case 1). A case with the original AMLB lead/lag logic is illustrated in Figures 6-8 and 6-9 (Case 2). The X-Y traces are dramatically different. However, there is less difference than anticipated in the PI plots. In Case 1, it will be observed that the "diving overshoot" results in a 5700 foot altitude drop for "B". "B"'s velocity decreases in spite of the

altitude drop and full afterburner setting. It can also be observed from Figure 6-6 that a large portion of the altitude drop occurs after the "diving overshoot command" (which is indicated on the figure as "DOVS") is replaced by the conventional pursuit (indicated as "P") after $t = 3$ seconds.

The maneuver is terminated when the range rate increases. However, an examination of case 1 suggests that the maneuver might instead be terminated earlier to avoid the altitude drop. For example, a criterion for maneuver termination might be when "B" has crossed "A"'s projected track.

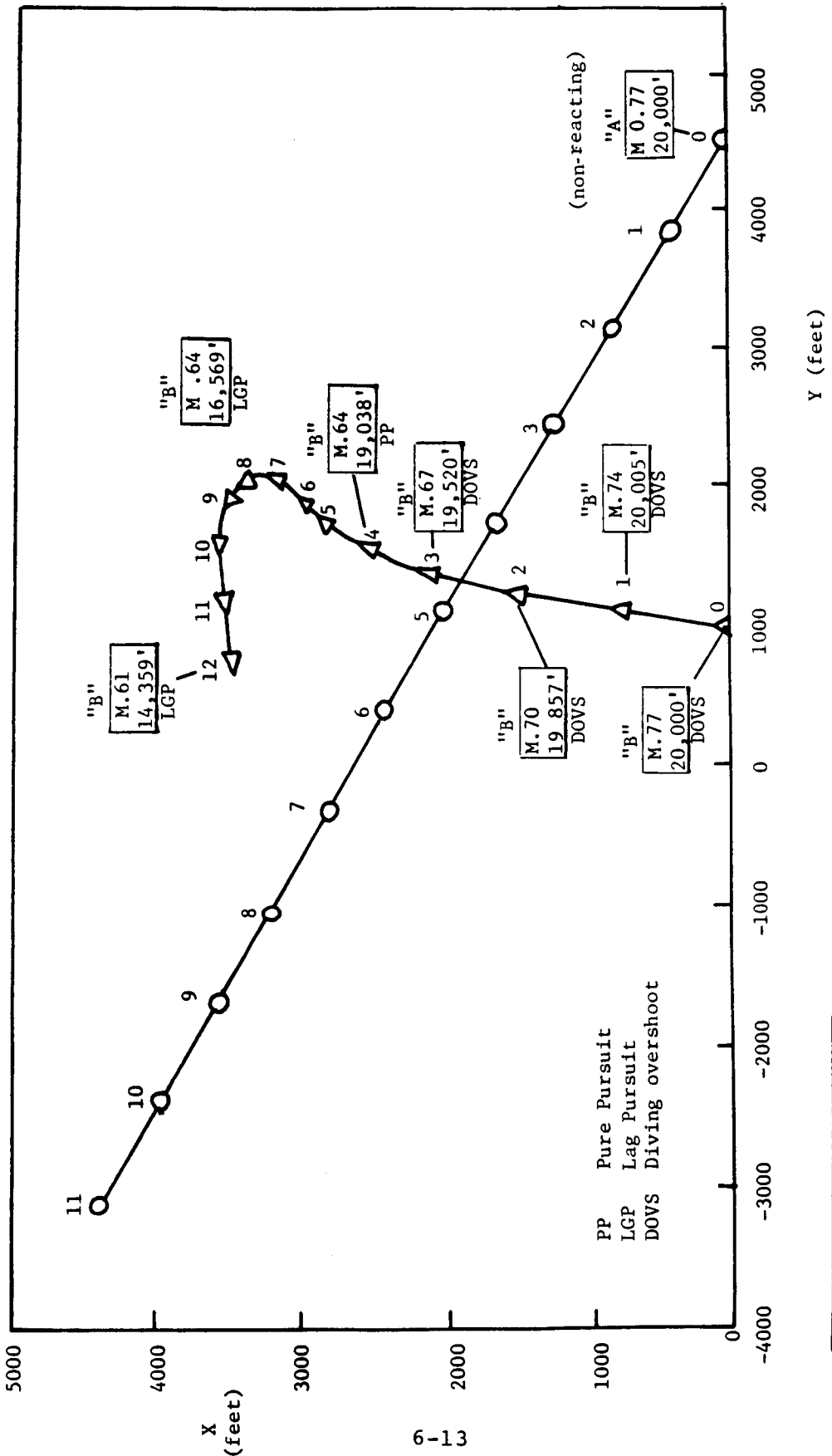


Figure 6-6. Diving overshoot X-Y trace

O "A" (F-15)
 Δ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude (in boxes)

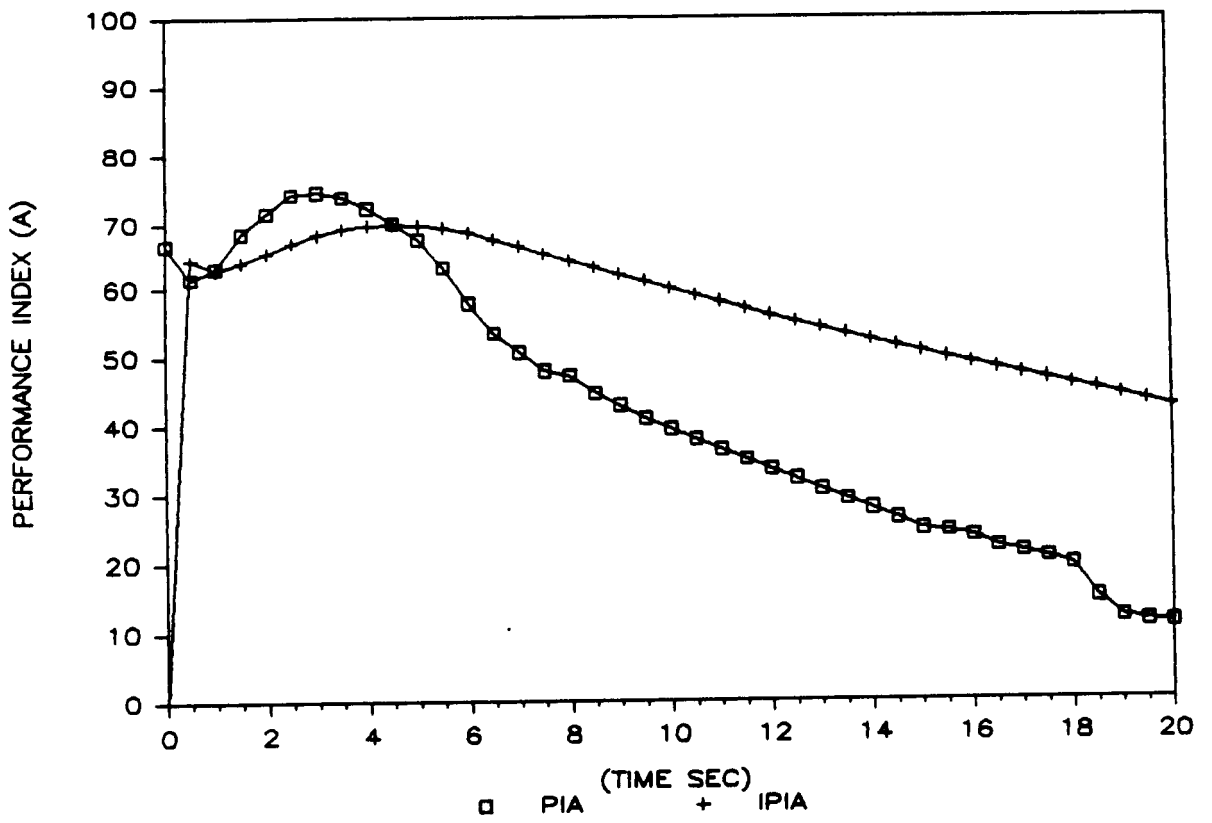


Figure 6-7 Diving Overshoot
PI Plot

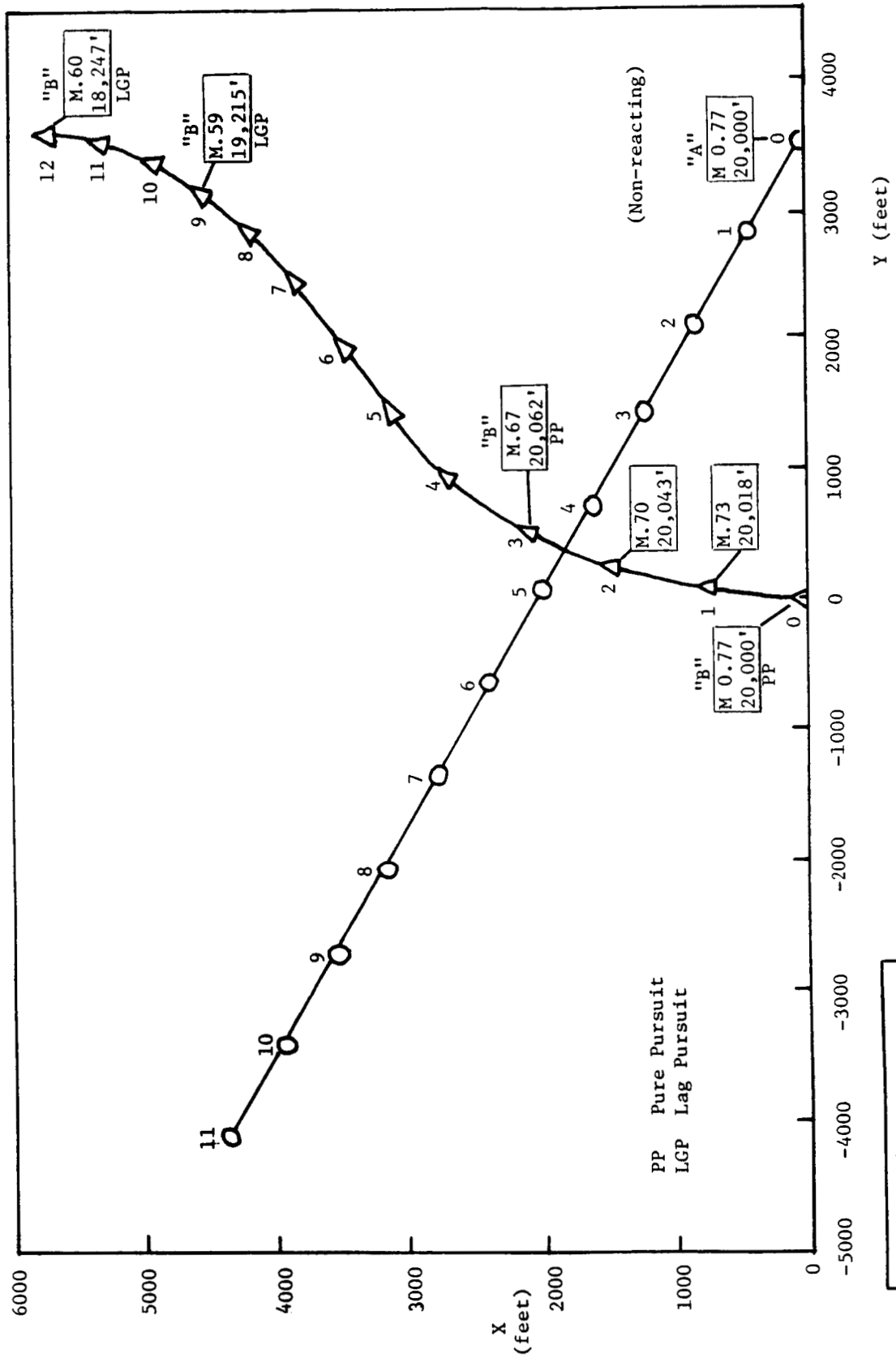


Figure 6-8. Lead/Lag overshoot X-Y trace

○ "A" (F-15)
 △ "B" (F-4)
 Time marks in seconds
 Mach No./Altitude (in boxes)

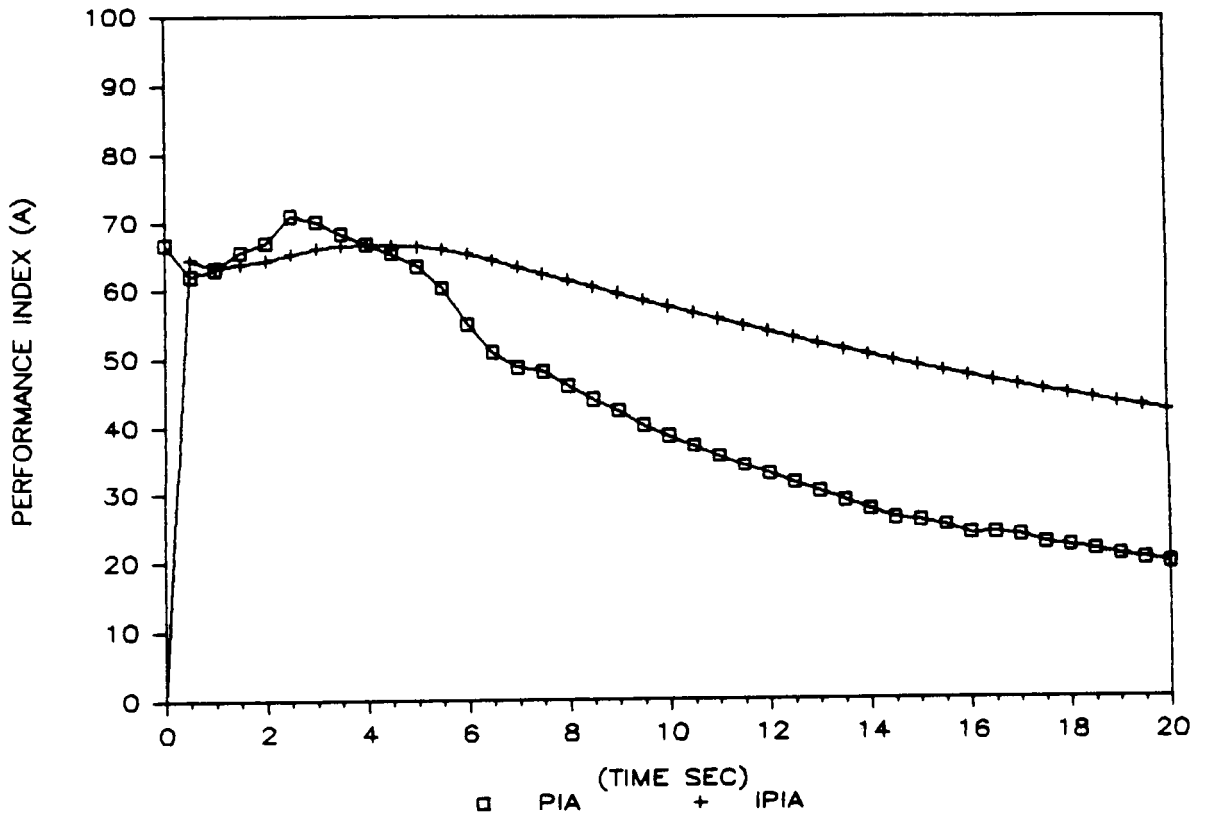


Figure 6-9 Lead/Lag
Overshoot PI Plot

6.3 PULL-UP OVERSHOOT

In section 5, some conditions under which scissors maneuvers are generated in forward passes were described. Reference 19 (page 6-28) describes a counter to that maneuver which consists initially of a pull-up with the intent of an overshoot. In order for this maneuver to work, the "B" aircraft needs a velocity excess which it can convert into an altitude advantage. This maneuver will be referred to here as a "pull-up overshoot".

A sample case is shown in Figure 6-10 and 6-11. These are neutral initial angular conditions, with the "B" aircraft having a speed advantage over the "A" aircraft (M.74 vs M.46). The initial conditions are similar to Case 4 of Section 5 shown in Figure 5-8.

The maneuver consists of a wings-level high-G (95% of maximum G) pull-up for a specified number of seconds (in this case 5 seconds.) Following the pull-up, control is reverted to the lead-lag logic to finalize the angular conversion with more favorable parameters. The wings-level pull-up causes the F-4 to separate angularly from the F-15. Because of the relative position at $t = 5$ sec (the time the F-4 stops climbing), the next maneuver was a tail-chase evasion (indicated in the figure as "TCE"), followed finally by a pursuit (indicated as "P"). As shown in Figure 6-11, the trend indicated by the performance index is that "B" is gaining an angular advantage. This would indicate that the maneuver has proven successful in breaking the stalemate of the scissors in "B"'s advantage.

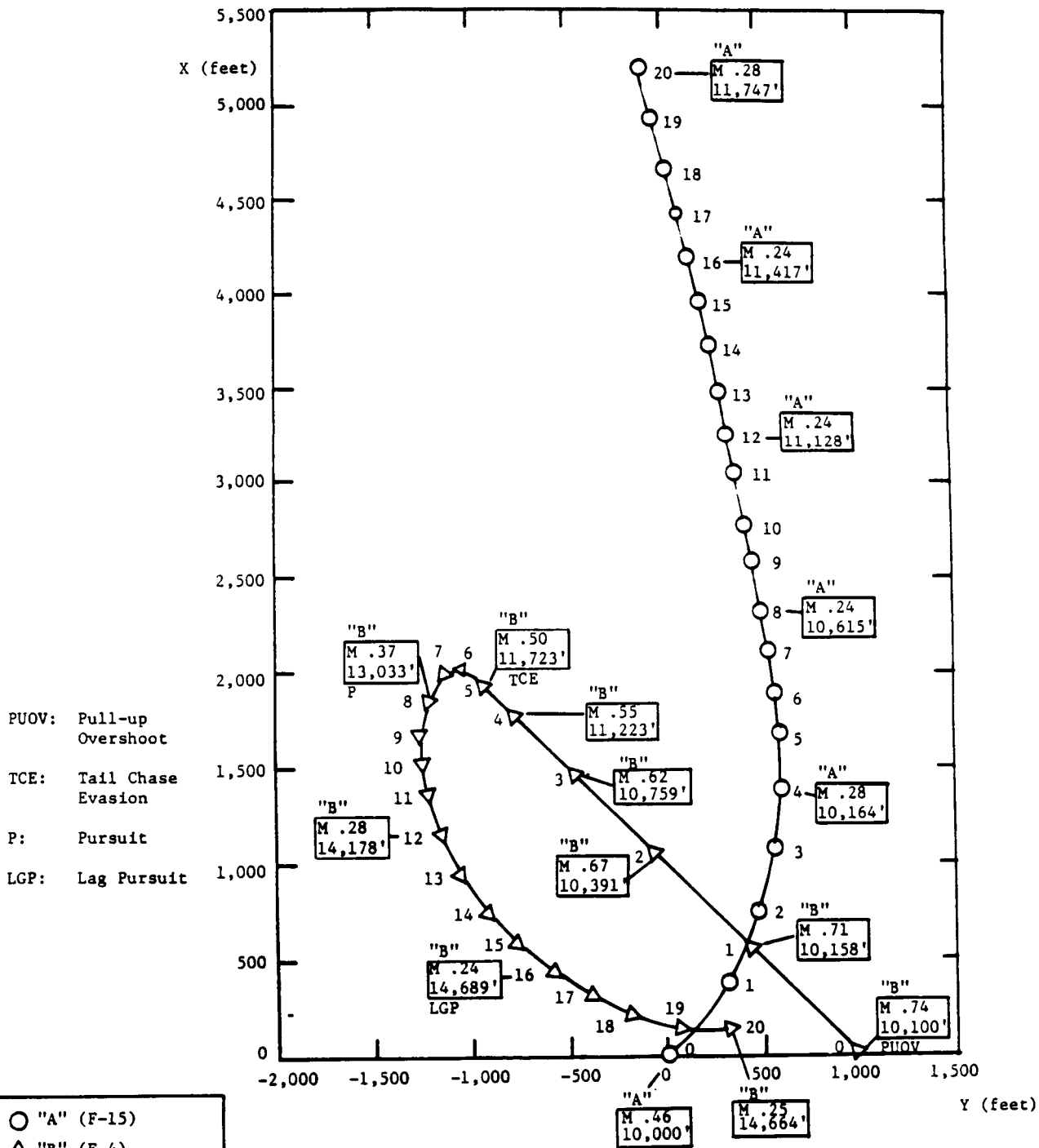


Figure 6-10 Pull-up Overshoot Case 1 X-Y Trace

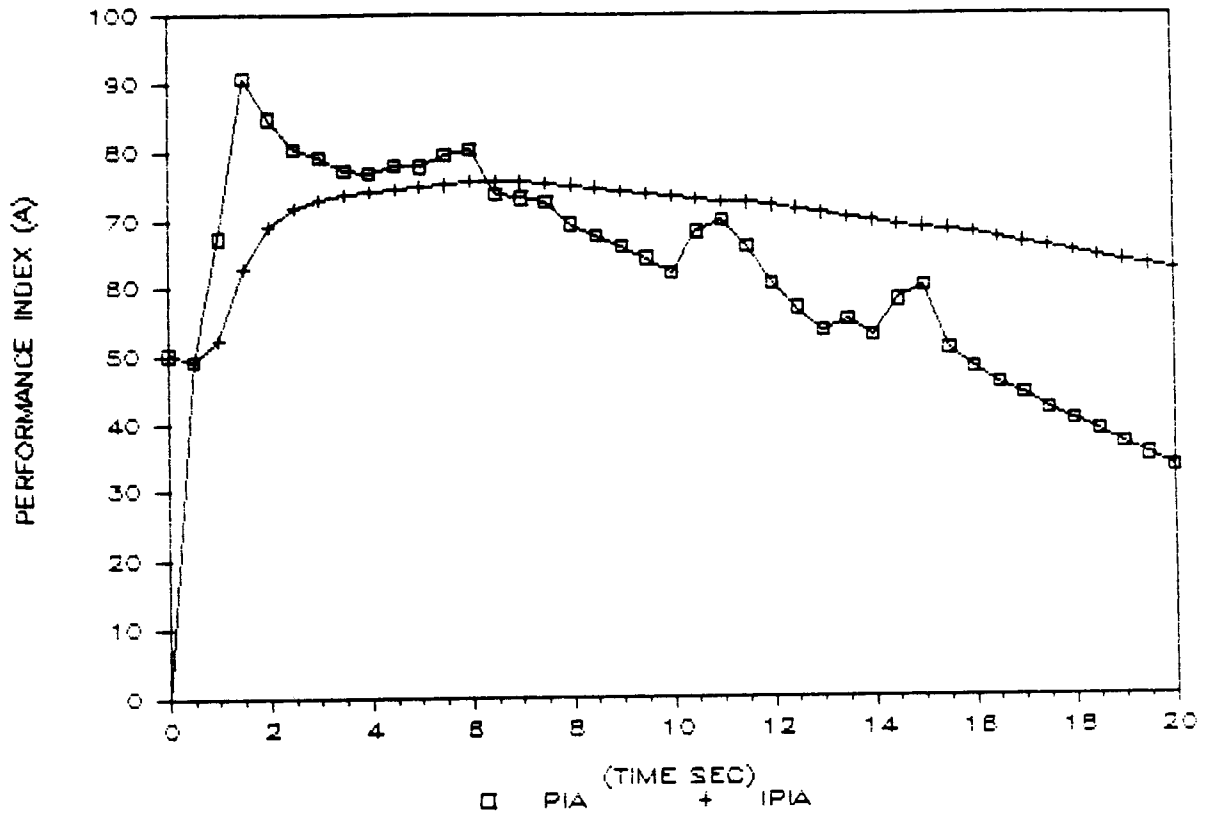


Figure 6-11 Pull-up Overshoot
Case 1 PI Plot

7. CONCLUSIONS AND RECOMMENDATIONS

In the almost twenty years since work on the adaptive maneuvering logic started, the AML programs have been continuously, little by little, improved. The AML version, which was distributed around 1978 by COSMIC had severe deficiencies, mostly in the attitude dynamics. It was unfortunately this versions (or derivatives thereof) which were installed in a number of US Navy and Air Force flight simulators.

Today's state of AML is that the motion of the AML driven aircraft is quite realistic and AML's tactical behavior is most of the time sound.

In the course of this long development period, we have learned a few basic and important lessons for the simulation of close-in air-to-air combat and for missile evasion:

- realistic aircraft motion, specifically the rotational dynamics, is of greatest importance for pilot acceptance.
- accurate roll and pitch dynamics are crucial when developing evasive maneuvers against surface-to-air or air-to-air missiles.
- improving the tactical behavior of AML is very time-consuming and tedious.
- the performance of any air-to-air combat program can only be evaluated statistically. Well over 100 different initial conditions must be exercised to

arrive at valid statistics.

- both methodologies, trial maneuver and IF => THEN, show promise.
- an analysis of the performance of an air-to-air combat program can not be made by analysis of non-real time, batch processing runs alone. An interaction with highly skilled human pilots is absolutely required.
- real-life air-to-air combat is extremely complex.

The original idea of this contract was to prepare a real-time base-line version of AML which could be used by the Flight Research Center to play the role as a "flight-director" controlling an actual airborne aircraft. By uplink telemetry, maneuver commands are issued to the aircraft and by down-link telemetry, aircraft status is received. Thus, the entire computational effort can be performed on ground. The complexity of such a project precluded implementation under this contract. We did, however, provide the Flight Research Center with an IF => THEN version of AML, running in real time in conjunction with an existing flight simulation. Due to lack of adequate real time display facilities, this air-combat simulation was not used much by the Flight Research Center.

We also recognize, at this point, that a number of problems in the simulation of one-versus-one combat still require additional studies and analyses. To name just a few:

- How can we prevent, early enough, the AML driven aircraft's energy to deteriorate to a very low value?
- If the AML driven aircraft "flies" against a dissimilar aircraft, how do we make best use in performance differences between the two aircraft?
- Is it possible to build an AML where all the IF => THEN production rules are formulated in plain English, so that a fighter pilot can change them at his will and investigate the effects of the change?
- How can distributed and parallel processing help to overcome some of the limitations imposed presently on real-time versions of AML?
- How can we put AML on-board a remotely piloted aircraft and then perform the ultimate "proof" for AML's tactics?
- How can we incorporate some of the results of the theory of differential games into AML?

If one admits that the decision logic of AML is not yet perfect (and the authors of this report certainly admit that), then a challenging problem is the following: How can we methodically improve AML? There are two aspects: (1) How to make changes to the decision logic and (2) How to evaluate the effects of these changes. It appears that a solution to this problem requires extensive use of a real-time full-dome flight simulator and the cooperation of experienced fighter pilots.

It appears, that after almost twenty years, the challenges in building an "iron-pilot" have not become smaller, but have grown.

REFERENCES

1. SLAATS (System Level Air-to-Air Tactical Simulation). Presented at the ORSA Conference, San Diego, October 25-27, 1982.
2. Moscow's Lessons from the 1982 Lebanon Air War, by Benjamin S. Lambeth, RAND Project Air Force, R-3000-AF.
3. Smart Weapons in Naval Warfare, by Paul F. Walker, Scientific American, vol 248, Number 5 (May 1983)
4. Computer-Automated Opponent for Manned Air-to-Air Combat Simulations, by Walter W. Hankins III, NASA Technical Paper 1518, September 1979.
5. An Adaptive Maneuvering Logic Computer Program for the Simulation of one-on-one air-to-air combat. Volume I: General Description, by George H. Burgin, Lawrence J. Fogel and J. Price Phelps. NASA Contractor Report CR-2582, September 1975.
6. An Adaptive Maneuvering Logic Computer Program for the Simulation of one-on-one air-to-air combat. Volume II: Program Description, by George H. Burgin and A. J. Phelps. NASA Contractor Report CR-2583, September 1975.
7. Advanced Air-to-air System Performance Evaluation Model (AASPEM) Analyst Manual, Vol I, Boeing Company Report D180-29122-1, November 1985.
8. PACAM V, Volume II, Analyst' Manual. AFWAL-TR81-3128, November 1981.
9. The TAC BRAWLER Air Combat Simulation, User's Manual (Rev 2.0), M.S. Carey et al. DSA Report No. 413, June 1982
10. Fighter Combat Tactics and Maneuvering, by Robert L. Shaw, Naval Institute Press, Annapolis, Maryland, 1985.
11. F-15 Basic Fighter Maneuvers, by Paul R. Stucky, USAF Fighter Weapons School, Nellis Air Force Base, Nevada, October 1982.
12. F-5E Combat Tactics Manual. Part 3 Air-to-air Combat Effectiveness, Northrop Corporation, September 1974.
13. Calculation of the Exchange Ratio for the Adaptive Maneuvering Logic Program, by Frank Neuman and Heinz Erberger. AIAA paper 85-0311. Presented at the 23rd Aerospace Sciences Meeting, January 14-17, 1985, Reno, Nevada.
14. Improvements to the Adaptive Maneuvering Logic Program, by George H. Burgin, NASA Contractor Report 3985, June 1986.
15. A Syllabus of Equations for Force Effectiveness Analysis, USAF Assistant chief of staff for Studies and Analysis, 1 December

1970.

16. AML/SAM User's Manual, Final Report, by G. H. Burgin, Contract F26600-78-C-0100, Tactical Fighter Weapons Center, Nellis Air Force Base, September 1979.

17. Multiple Tactical Aircraft Combat Performance Evaluation System, AIAA-80-0189, by D. S. Hague

18. Fighters get a quick draw, by Richard DeMeis, Aerospace America, September 1986.

19. Aerial Attack, Vol 1. Canadian Forces Air Combat Manual 2-311, 1980.

20. Design of an All-Attitude Flight Control System to Execute Commanded Bank angles and Angles of Attack, NASA-CR-145004, by George H. Burgin and David M. Eggleston.

21. The Adaptive Maneuvering Logic Program in support of the Pilot's Associate Program: A Heuristic Approach to Missile Evasion, AIAA Paper-86-0423, by G.H. Burgin, W.H. Williams and L.B. Sidor. Presented at the 24th AIAA Aerospace Sciences Meeting, January 6-9, 1986, Reno, Nevada.

22. Evaluation d'Avions en Combat Simulé, Calculateur contre Calculateur, ou Calculateur contre Pilote Humain, by J. Pedotti and Y. Hignard, in Combat Aircraft Manoeuvrability, AGARD-CP-319, October 1981.

23. Automated Maneuvering Decisions for Air-to-Air Combat, by Fred Austin, Giro Carbone and Michael Lewis, in the Proceedings of the Military Computing Conference, Anaheim, California, May 1987.

24. Modern Air Combat, by Bill Gunston and Mike Spick, Crescent Books, 1983.

25. Strapdown Seeker Guidance for Air-to-Surface Tactical Weapons Report AFATL-TR-78-60 by Missile Systems Division, Rockwell International Corporation, Columbus, Ohio May 1978.

A P P E N D I X

LISTING OF THE FORTRAN ROUTINES FOR
"BASIC FIGHTER MANEUVERS"

SUBROUTINE SELCTB	A-1
SUBROUTINE OVRSHT	A-8
SUBROUTINE VTOSH	A-10
SUBROUTINE OPSTRN	A-12
SUBROUTINE CLIMB	A-14

```

0001
0002 SUBROUTINE SELCTB(XEA,YEA,ZEA,XEDOTA,YEDOTA,ZEDOTA,DMTRXA)
0003 C
0004 C
0005 C SUBROUTINE BASED ON REACTB (MAR 1986) TO SELECT AND EXECUTE
0006 C APPROPRIATE MANEUVERS. WHEN SEVERAL MANEUVERS ARE FEASIBLE, ONE
0007 C SPECIFIC MANEUVER IS SELECTED USING A RANDOM NUMBER
0008 C WHEN NO SUCH MANEUVER IS FOUND, THIS SUBROUTINE REVERTS TO THE
0009 C LEAD/LAG STEERING LAW OF REACTB
0010 C
0011 C -----
0012 C
0013 C *** COMMON BLOCKS FOR REAL TIME SIMULATION AT NASA DRYDEN
0014 C
0015 C COMMON/CNSTNS/DT, TBEGN, TNOW, PI, PIDV2, PIDV4, TWOP1, DEGRD, RADDG, G,
0016 C 1 VAR(20), IVAR(20), TEND
0017 C
0018 C COMMON/CONTRL/MSTOP, IPRINT
0019 C
0020 C COMMON/COMNDB/ICMNB, GLEVLB, ROTB, MANVRB
0021 C
0022 C COMMON/RNDMAN/MLDECS, MANINI, ISLCTR
0023 C COMMON/DATA1B/XEB, YEB, ZEB, XEDOTB, YEDOTB, ZEDOTB, XEDDTB, YEDDTB,
0024 C 1 ZEDDTB, PS1B, THETAB, PH1B, UB, VB, WB, PB, QB, RB, A1B, A2B,
0025 C 2 A3B, A4B, VELB, VHORB
0026 C
0027 C COMMON/DATA2B/ALFAB, BETAB, CBARB(3,3), CDB, CLALFB, DMTRXB(3,3), DRAGB,
0028 C 1 LIFTB, LODMXB, LODSTB, MACHB, RHOB, SPECEB, SB, THRSTB,
0029 C 2 PSUBSB, TPOSB, INIZB, WEITB, CSB, CLB, PS1BRB, THETBB,
0030 C 3 AN1B, AN2B, AN3B, MASSB
0031 C
0032 C REAL LIFTB, LODMXB, LODSTB, MACHB, MASSB
0033 C
0034 C
0035 C
0036 C COMMON/TBF4EB/CLMAXB(14) ,XTAB1(14) ,NX1,
0037 C . THR1DB(7,14) ,XTAB2(7) ,YTAB2(14) ,NX2,NY2,
0038 C . THRMLB(7,14) ,XTAB3(7) ,YTAB3(14) ,NX3,NY3,
0039 C . THRABB(7,14) ,XTAB4(7) ,YTAB4(14) ,NX4,NY4,
0040 C . ALFCLB(16,10),XTAB5(16),YTAB5(10),NX5,NY5,
0041 C . CLFALB(10,11),XTAB6(10),YTAB6(11),NX6,NY6,
0042 C . CDFCLB(16,10),XTAB7(16),YTAB7(10),NX7,NY7,
0043 C . CLFCDB(18,10),XTAB8(18),YTAB8(10),NX8,NY8,
0044 C . RECAGB(10,12),XTAB9(10),YTAB9(12),NX9,NY9
0045 C
0046 C
0047 C
0048 C COMMON/POINTP/XEAIM, YEAIM, ZEAIM, PCOMB, QCOMB, RCOMB, IPOINT
0049 C
0050 C DIMENSION CMPL(3,3)
0051 C
0052 C COMMON/RELVAR/LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,
0053 C 1 LSDOTA, LSDOTB, DEVANA, DEVANB, DVDDOTA, DVDDOTB,
0054 C 2 RANGE, RRATE, XAINB, YAINB, ZAINB, XBINA, YBINA,
0055 C 3 ZBINA, ANGOFA, ANGOFB
0056 C
0057 C REAL LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,

```

SELCTH

17-Jul-1987 14:0
17-Jul-1987 14:0

```
0058          1          LSDOTA,LSDOTB
0059      C
0060      C
0061      C *** LET THROTTLE ROUTINE KNOW WHEN A/C IS IN DIVE RECOVERY
0062      C
0063          COMMON/DIVEB/IRECVB
0064      C
0065          COMMON/PAGECT/ICNT
0066      C
0067          DATA IXL1,JYL1,IXL2,JYL2/1,1,1,1/
0068          DATA IXL9,JYL9/1,1/
0069      C
0070      C THE DEFAULT DECISION INTERVAL IS MLDEF, ELSE IT IS SET IN
0071      C THE INDIVIDUAL MANEUVER ROUTINE
0072      C
0073          DATA MLDEF/20/,NEGTVG/1/
0074      C
0075          IF (INIZB.EQ.1) THEN
0076      C
0077              MLDECS=MLDEF
0078              ISLCTR=0
0079              IRECVB=0
0080              RETURN
0081
0082          ENDIF
0083      C
0084      C -----
0085      C
0086          GROUND AVOIDENCE LOGIC
0087      C
0088      C -----
0089      C
0090          HB=-ZEB
0091          DIVEAN=-THETBB
0092          IF(IRECVB.EQ.1 .AND. DIVEAN.GT.0.) THEN
0093              MANVRB=1
0094              ICMNWB=1
0095              ROTB=0.
0096              IPOINT=0
0097              GO TO 998
0098          ENDIF
0099          IF(IRECVB.EQ.1 .AND. DIVEAN.LE.0.) THEN
0100              IRECVB=0
0101          ELSE
0102              IF(HB.LT.20000.) THEN
0103                  FMACHX=MACHB
0104                  IF(FMACHX.LT..4)FMACHX=.4
0105                  HX=HB
0106                  IF(HX.LT.200.)HX=200.
0107                  CALL TLU2(HX,FMACHX,XTAB9,YTAB9,RECAGB,NX9,NY9,
0108                          IXL9,JYL9,RECAN,IC)
0109              ELSE
0110                  RECAN=PIDV2
0111              ENDIF
0112              IF(DIVEAN.GT.RECAN) THEN
0113                  ROTB=0.
0114                  GLEVLB=1.
```


SELCTB

17-Jul-1987 14:0:
17-Jul-1987 14:0:

```
0229             MANVRB=9
0230             ENDDIF
0231
0232             XEXA=XFA+DTPRED*XEDOTA
0233             YEXA=YEA+DTPRED*YEDOTA
0234             ZEXA=ZEA+DTPRED*ZEDOTA
0235
0236             TAXE=XEXA-XEB
0237             TAYE=YEXA-YEB
0238             TAZE=ZEXA-ZEB
0239             VHOR2=XEDOTB**2+YEDOTB**2
0240             VHORB=SQRT(VHOR2)
0241             VEL2=VHOR2+ZEDOTB**2
0242             VELB=SQRT(VEL2)
0243
0244             DZ=(XEDOTB*ZEDOTB*TAXE+YEDOTB*ZEDOTB*TAYE-VHORB**2*TAZE)/VELB
0245             DY=-YEDOTB*TAXE+XEDOTB*TAYE
0246             IF(DZ.EQ.0. .AND. DY.EQ.0.) THEN
0247                 ROTB=0.
0248             ELSE IF (DY.EQ.0.) THEN
0249                 IF(DZ.GT.0.) ROTB=0.
0250                 IF(DZ.LT.0.) ROTB=PI
0251             ELSE
0252                 ROTB=ATAN2(DY,DZ)
0253             ENDDIF
0254 C
0255 C *** SELECT THE POSITIVE G-LEVEL DEPENDING ON B'S VELOCITY
0256 C
0257             IF(VELB.GT.400.) THEN
0258                 GLVPOS=(LODSTB+LODMXB)/(2.*LODMXB)
0259             ELSE
0260                 GLVPOS=LODSTB/LODMXB
0261             ENDDIF
0262             TAS=VELB*0.5925
0263             CAS=TAS*SQRT(RHOB/0.0023768)
0264             IF(CAS.GT.1.2*330.) GLVPOS=(LODSTB+LODMXB)/(2.*LODMXB)
0265 C
0266             ICMNWB=1
0267 C
0268 C *** CALCULATE INTERCEPT TRAJECTORY G-LEVEL
0269 C
0270             IF(FLAMBD.LT.60.) THEN
0271                 CALI DIRCOS(PSIBRB,THETBB,ROTB,CMPL)
0272                 DIST2=TAXE**2+TAYE**2+TAZE**2
0273                 ZMT=TAXE*CMPL(3,1)+TAYE*CMPL(3,2)+TAZE*CMPL(3,3)
0274                 RADIS=DIST2/(2.*ZMT)
0275                 GL2=(ABS((VELB**2)/RADIS)/G)+CMPL(3,3)
0276                 GL3=ABS(CMPL(2,3))
0277                 GLEVRB=SQRT(GL2**2+GL3**2)/LODMXB
0278                 IF(GLEVRB.LT.GLVPOS) THEN
0279                     GLVPOS=GLEVRB
0280                 ENDDIF
0281 C
0282 C ----- 21 APRIL
0283 C CALCULATE INTERCEPT TRAJECTORY FOR NEGATIVE G'S
0284 C
0285             IF(ROTB.LE.0.) THEN
```

SELCTB

17-Jul-1987 14:01
17-Jul-1987 14:01

```
0286          ROTBNG=PI+ROTB
0287          ELSE
0288          ROTBNG=-PI+ROTB
0289          ENDIF
0290          CALL DIRCOS(PSIBRB,THETBB,ROTBNG,CMPL)
0291          ZMT=TAXE*CMPL(3,1)+TAYE*CMPL(3,2)+TAZE*CMPL(3,3)
0292          RADIS= DIST2/(2.*ZMT)
0293          GL2=(ABS((VELB**2)/RADIS)/G)+CMPL(3,3)
0294          GL3=ABS(CMPL(2,3))
0295          GLEVRB=SQRT(GL2**2+GL3**2)/LODMXB
0296          GLVNEG=2.0/LODMXB
0297          IF(GLEVRB .LT. 2.0/LODMXB)THEN
0298              GLVNEG=GLEVRB
0299          ENDIF
0300      C
0301      C   DETERMINE IF NEGATIVE G'S ARE ALLOWED, INCLUDING WHETHER GLVNEG EXCE
0302      C   ALLOWABLE LEVEL (PRESENTLY SET TO -2 G)
0303      C
0304      C   CONDITIONS FOR USING NEGATIVE G'S:
0305      C   1. B'S AIRSPEED MUST BE LOWER THAN A'S
0306      C   2. A MUST BE IN B'S FORWARD QUARTER AND LOW (FLAMBD <60 DEG,
0307      C       LOSELA < -5 DEG)
0308      C   3. B'S ROLL ANGLE MUST NOT EXCEED 30 DEGREES- ELSE HE IS BETTER OFF
0309      C       ROLLING INVERTED
0310      C   THE OBJECTIVE OF NEGATIVE G'S IS TO BRING B'S NOSE ON A WITHOUT
0311      C   PULLING HIGH G'S (HENCE LOSING AIRSPEED)
0312      C   (A FUNCTION HAVING SIMILARITIES WITH THE POINTING ALGORITHM)
0313      C   IN GENERAL, NEGATIVE G'S WOULD BE USED TO UNLOAD THE AIRPLANE IN
0314      C   ORDER TO GAIN/REGAIN AIRSPEED, E.G. TO GAIN SEPARATION
0315      C
0316          NEGGEF=1
0317          IF (LOSELB .GT. -5.0/DEGRD) NEGGEE=0
0318          IF (PHIB .LE. -30.0/DEGRD .OR. PHIB .GE. 30.0/DEGRD) NEGGEE=0
0319          SPEEDA=SQRT(XEDOTA**2+YEDOTA**2+ZEDOTA**2)
0320          IF (VELB .GT. .90*SPEEDA) NEGGEE=0
0321      C
0322      C
0323      C
0324      C   --SELECT THE MANEUVER WHICH YIELDS THE SMALLEST VARIATION IN ROTB
0325      C
0326          IF( ABS(ROTBNG-ROTPRV) .LT. ABS(ROTB-ROTPRV)
0327      1          .AND. NEGGEE .EQ. 1) THEN
0328              GLFVLB= -GLVNEG
0329              ROTB=ROTBNG
0330              MANVRB=11
0331              GOTO 998
0332          ELSE
0333              GLFVLB=GLVPOS
0334              MANVRB=6
0335              GOTO 998
0336          ENDIF
0337      C
0338          ENDIF
0339      C -----
0340      C
0341      C *** SEE IF WE ARE IN TROUBLE AND NEED AN EVASIVE MANEUVER
0342      C
```

21 APR 1987

SELCTB

17-Jul-1987 14:0

17-Jul-1987 14:0

```

0343         IF (EPSD.GT.120. .AND. FLAMBD.GT.120.) THEN
0344             GLEVLB=0.9
0345             IF (CAS.GT.330.) GLEVLB=1.
0346                 ROTB=ROTB-PIDV2
0347                 MANVRB=8
0348         ENDIF
0349 C
0350     998     CONTINUE
0351     WRITE(*,491)TNOW,MANVRB,ROTB*DEGRD,GLEVLB,RANGE
0352 C
0353     WRITE(77,491)TNOW,MANVRB,ROTB*DEGRD,GLEVLB,RANGE
0354 C
0355     491 FORMAT(' SELCTB S DECISION'F15.2,15,F10.2,F10.2,F12.1,/)
0356 C
0357     ICNT=ICNT+3
0358 C
0359 C *** END OF REGULAR DECISION MAKING PART
0360 C
0361     CALL THRTL B
0362     ENDIF
0363 C
0364 C     END OF LEAD/LAG IF 111111111111111111111111111111111111111111111111111
0365 C
0366     999 RETURN
0367 C
0368 C----- MANEUVER CODES -----
0369 C
0370 C     MANVRB=1     DIVE RECOVERY ACTIVE
0371 C     MANVRB=2     DIVE RECOVERY INITIATED
0372 C     MANVRB=3     LEAD PURSUIT
0373 C     MANVRB=4     PURSUIT (FLAMBD=    TO    ;EPSD=    TO    )
0374 C     MANVRB=5     LAG PURSUIT
0375 C     MANVRB=6     INTERCEPT TRAJECTORY
0376 C     MANVRB=7     POINTING ALGORITHM
0377 C     MANVRB=8     TAIL-CHASE EVASION
0378 C     MANVRB=9     PURSUIT (FLAMBD=    TO    ;EPSD=    TO    )
0379 C     MANVRB=11    NEGATIVE G'S
0380 END

```



```

0001      SUBROUTINE OVRSHT(XEA,YEA,ZEA,XEDOTA,YEDOTA,ZEDOTA,DMTRXA)
0002      C
0003      COMMON/RNDMAN/ MLDECS,MANINI,ISLCTR
0004      C
0005      COMMON/CNSTNS/DT,TBEGN,TNOW,PI,PIDV2,PIDV4,TWOP1,DEGRD,RADDG,G,
0006      1          VAR(20),IVAR(20),TEND
0007      C
0008      COMMON/CONTRL/MSTOP,IPRINT
0009      C
0010      COMMON/COMNDB/ICMNWB,GLEVLB,ROTB,MANVRB
0011      C
0012      COMMON/DATA1B/XEB,YEB,ZEB,XEDOTB,YEDOTB,ZEDOTB,XEDDTB,YEDDTB,
0013      1          ZEDDTB,PSIB,THETAB,PHIB,UB,VB,WB,PB,QB,RB,A1B,A2B,
0014      2          A3B,A4B,VELB,VHORB
0015      C
0016      COMMON/DATA2B/ALFAB,BETAB,CBARB(3,3),CDB,CLALFB,DMTRXB(3,3),DRAGB,
0017      1          LIFTB,LODMXB,LODSTB,MACHB,RHOB,SPECEB,SB,THRSTB,
0018      2          PSUBSB,TPOSB,INIZB,WEITB,CSB,CLB,PSIBRB,THETBB,
0019      3          AN1B,AN2B,AN3B,MASSB
0020      C
0021      REAL          LIFTB,LODMXB,LODSTB,MACHB,MASSB
0022      C
0023      C
0024      COMMON/POINTP/XEAIM,YEAIM,ZEAIM,PCOMB,QCOMB,RCOMB,IPOINT
0025      C
0026      DIMENSION CMPL(3,3)
0027      C
0028      COMMON/RELVAR/LOSELA,LOSELB,LOSAZA,LOSAZB,LOSANA,LOSANB,
0029      1          LSDOTA,LSDOTB,DEVANA,DEVANB,DVDDOTA,DVDDOTB,
0030      2          RANGE,RRATE,XAINB,YAINB,ZAINB,XBINA,YBINA,
0031      3          ZBINA,ANGOFA,ANGOFB
0032      C
0033      REAL          LOSELA,LOSELB,LOSAZA,LOSAZB,LOSANA,LOSANB,
0034      1          LSDOTA,LSDOTB
0035      C
0036      C      THIS SUBROUTINE GENERATES COMMANDS TO THE B AIRCRAFT TO
0037      C      FORCE AN OVERTHOOT OF THE A AIRCRAFT . THIS MANEUVER COMBINES ROLL
0038      C      IN THE DIRECTION OF A, COMBINED WITH A DIVE
0039      C
0040      C      THIS MANEUVER SHOULD BE INITIATED UNDER THE FOLLOWING CONDITIONS:
0041      C      RANGE <=3500 FT; EPSD
0042      C
0043      C      -----
0044      C      INITIALIZATION SECTION
0045      C
0046      IF(MANINI.EQ.-1)THEN
0047      TIMREQ= 15.
0048      MLDECS= INT(TIMREQ/DT)
0049      TINIT= TNOW
0050      TQUIT= TNOW+TIMREQ
0051      MANINI=0
0052      ENDIF
0053      C
0054      DTPRED=0.0
0055      XEXA= XEA + DTPRED*XEDOTA
0056      YEXA= YEA + DTPRED*YEDOTA
0057      ZEXA= ZEA + 10000.
           TAXE=XEXA-XEB
  
```

OVRSH T

17-Jul-1987 14:01
17-Jul-1987 14:01

```
0058      TAYE=YEXA-YEB
0059      TA7E=ZEXA-ZEB
0060      VHOR2=XEDOTB**2+YEDOTB**2
0061      VHORB=SQRT(VHOR2)
0062      VEI.2=VHOR2+ZEDOTB**2
0063      VEI.B=SQRT(VEL2)
0064
0065      DZ=(XEDOTB*ZEDOTB*TAXE+YEDOTB*ZEDOTB*TAYE-VHORB**2*TAZE)/VELB
0066      DY=-YEDOTB*TAXE+XEDOTB*TAYE
0067      IF(DZ.EQ.0. .AND. DY.EQ.0.) THEN
0068          ROTB=0.
0069      ELSE IF (DY.EQ.0.) THEN
0070          IF(DZ.GT.0.) ROTB=0.
0071          IF(DZ.LT.0.) ROTB=PI
0072      ELSE
0073          ROTB=ATAN2(DY,DZ)
0074      ENDIF
0075      C
0076      C SELECT MAXIMUM G TRUN
0077      C
0078          GLEVLH=0.95
0079          TPOSB= 2.0
0080          ICMNWH=1
0081      C
0082      C CHECK FOR MANEUVER TERMINATION CONDITIONS
0083      C
0084          IF(TNOW .GE. TQUIT)THEN
0085              ISLCTR=0
0086          ENDIF
0087          IF(RRATE .GT.0.) THEN
0088              ISLCTR=0
0089          ENDIF
0090
0091      RETURN
0092      END
```

```

0001      SUBROUTINE VTOVSH(XEA,YEA,ZEA,XEDOTA,YEDOTA,ZEDOTA,DMTRXA)
0002      C
0003      COMMON/RNDMAN/ MLDECS,MANINI,ISLCTR
0004      C
0005      COMMON/CNSTNS/DT,TBEGN,TNOW,PI,PIDV2,PIDV4,TWOPI,DEGRD,RADDG,G,
0006      1      VAR(20),IVAR(20),TEND
0007      C
0008      COMMON/CONTRL/MSTOP,IPRINT
0009      C
0010      COMMON/COMNDB/ICMNWB,GLEVLB,ROTB,MANVRB
0011      C
0012      COMMON/DATA1B/XEB,YEB,ZEB,XEDOTB,YEDOTB,ZEDOTB,XEDDTB,YEDDTB,
0013      1      ZEDDTB,PS1B,THETAB,PH1B,U8,VB,WB,PB,QB,FB,A1B,A2B,
0014      2      A3B,A4B,VELB,VHORB
0015      C
0016      COMMON/DATA2B/ALFAB,BETAB,CBARB(3,3),CDB,CLALFB,DMTRXB(3,3),DRAGB,
0017      1      LIFTB,LODMXB,LODSTB,MACHB,RHOB,SPECEB,SB,THRSTB,
0018      2      PSUBSB,TPOSB,INIZB,WEITB,CSB,CLB,PS1BRB,THETBB,
0019      3      AN1B,AN2B,AN3B,MASSB
0020      C
0021      REAL      LIFTB,LODMXB,LODSTB,MACHB,MASSB
0022      C
0023      C
0024      COMMON/POINTP/XEAIM,YEAIM,ZEAIM,PCOMB,QCOMB,RCOMB,IPOINT
0025      C
0026      DIMENSION CMPL(3,3)
0027      C
0028      COMMON/RELVAR/LOSELA,LOSELB,LOSAZA,LOSAZB,LOSANA,LOSANB,
0029      1      LSDOTA,LSDOTB,DEVANA,DEVANB,DVDOTA,DVDOTB,
0030      2      RANGE,RRATE,XAINB,YAINB,ZAINB,XBINA,YBINA,
0031      3      ZBINA,ANGOFA,ANGOFB
0032      C
0033      REAL      LOSELA,LOSELB,LOSAZA,LOSAZB,LOSANA,LOSANB,
0034      1      LSDOTA,LSDOTB
0035      C
0036      C      THIS SUBROUTINE GENERATES COMMANDS TO THE B AIRCRAFT TO
0037      C      CLIMB IN THE VERTICAL PLANE FOR A SPECIFIED TIME.
0038      C      THIS CAN BE USED TO FORCE AN OVERSHOOT OF THE A AIRCRAFT
0039      C      WHEN IT IS DIVING ONTO B
0040      C
0041      C
0042      C      THIS MANEUVER SHOULD BE INITIATED UNDER THE FOLLOWING CONDITIONS:
0043      C      RANGE <=3500 FT; EPSD
0044      C      -----
0045      C      INITIALIZATION SECTION
0046      C
0047      IF(MANINI.EQ.-1)THEN
0048      TIMREQ= 15.
0049      MLDECS= INT(TIMREQ/DT)
0050      TINIT= TNOW
0051      TQUIT= TNOW+TIMREQ
0052      MANINI=0
0053      ENDIF
0054      C
0055      ROTB=0.
0056      C
0057      C      SELECT LOW G IN ORDER NOT TO LOSE TOO MUCH ENERGY
  
```

VTOVSH

17-Jul-1987 14:0
17-Jul-1987 14:0

```
0058 C
0059 GLEVLB=1.5/LODMXB
0060 TPOSB= 2.0
0061 ICMNWB=1
0062 C
0063 C CHECK FOR MANEUVER TERMINATION CONDITIONS
0064 C
0065 IF(TNOW .GE. TQUIT)THEN
0066 ISLCTR=0
0067 ENDIF
0068 IF(RRATE .GT.0.) THEN
0069 ISLCTR=0
0070 ENDIF
0071
0072 RETURN
0073 END
```

```

0001
0002      SUBROUTINE OPSTRN(XEA, YEA, ZFA, XEDOTA, YEDOTA, ZEDOTA, DMTRXA)
0003      C
0004      COMMON/RNDMAN/ MLDECS, MANINI, ISLCTR
0005      C
0006      COMMON/CNSTNS/DT, TBEGN, TNOW, PI, PIDV2, PIDV4, TWOP1, DEGRD, RADDG, G,
0007      1          VAR(20), IVAR(20), TEND
0008      C
0009      COMMON/CONTHL/MSTOP, IPRINT
0010      C
0011      COMMON/COMNDB/ ICMNWB, GLEVLB, ROTB, MANVRB
0012      C
0013      COMMON/DATA1B/XEB, YEB, ZEB, XEDOTB, YEDOTB, ZEDOTB, XEDDTB, YEDDTB,
0014      1          ZEDDTB, PSIB, THETAB, PHIB, UR, VB, WB, PB, QB, RB, A1B, A2B,
0015      2          A3B, A4B, VELB, VHORB
0016      C
0017      COMMON/DATA2B/ALFAB, BETAB, CBARB(3,3), CDB, CLALFB, DMTRXB(3,3), DRAGB,
0018      1          LIFTB, LODMXB, LODSTB, MACHB, RHOB, SPECEB, SB, THRSTB,
0019      2          PSUBSB, TPOSB, INIZB, WEITB, CSB, CLB, PSIBRB, THETBB,
0020      3          AN1B, AN2B, AN3B, MASSB
0021      C
0022      REAL          LIFTB, LODMXB, LODSTB, MACHB, MASSB
0023      C
0024      C
0025      COMMON/POINTP/XEAIM, YEAIM, ZFAIM, PCOMB, QCOMB, RCOMB, IPOINT
0026      C
0027      DIMENSION CMPL(3,3)
0028      C
0029      COMMON/RELVAR/LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,
0030      1          LSDOTA, LSDOTB, DEVANA, DEVANB, DVDOTA, DVDOTB,
0031      2          RANGE, RRATE, XAINB, YAINB, ZAINB, XBINA, YBINA,
0032      3          ZBINA, ANGOFA, ANGOFB
0033      C
0034      REAL          LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,
0035      1          LSDOTA, LSDOTB
0036      C
0037      C      THIS SUBROUTINE GENERATES COMMANDS TO THE B AIRCRAFT TO
0038      C      TURN OPPOSITE TO A FOR A DURATION OF 15 SECONDS
0039      C      OR UNTIL A IS WITHIN A 60 DEG CONE ANGLE
0040      C      IT IS USED TO GENERATE A NOSE-TO-NOSE CONVERSION (B CONVERTS
0041      C      TO A'S NOSE)
0042      C
0043      C      THIS MANEUVER SHOULD BE INITIATED UNDER THE FOLLOWING CONDITIONS:
0044      C
0045      C      -----
0046      C          INITIALIZATION SECTION
0047      C
0048      C      IF(MANINI.EQ.-1)THEN
0049      C          TIMREQ= 20.
0050      C          MLDECS= INT(TIMREQ/DT)
0051      C          TINIT= TNOW
0052      C          TQUIT= TNOW+TIMREQ
0053      C          MANINI=0
0054      C
0055      C
0056      C      DTPRED=0.0
0057      C      XEXA= XEA + DTPRED*XEDOTA
  
```

OPSTRN

17-Jul-1987 14:0

17-Jul-1987 14:0

```
0058      YEXA= YEA + DTPRED*YEDOTA
0059      ZEXA= ZEA
0060      TAXE=XEXA-XEB
0061      C
0062      C      --- NOTE
0063      C
0064      TAYE=-(YEXA-YEB)
0065      C
0066      TAXE=ZEXA-ZEB
0067      VHOR2=XEDOTB**2+YEDOTB**2
0068      VHORB=SQRT(VHOR2)
0069      VEL2=VHOR2+ZEDOTB**2
0070      VE1.B=SQRT(VEL2)
0071
0072      DZ= (XEDOTB*ZEDOTB*TAXE+YEDOTB*ZEDOTB*TAYE-VHORB**2*TAZE)/VELB
0073      DY= -YEDOTB*TAXE+XEDOTB*TAYE
0074      IF(DZ.EQ.0. .AND. DY.EQ.0.) THEN
0075          ROTB=0.
0076      ELSE IF (DY.EQ.0.) THEN
0077          IF(DZ.GT.0.) ROTB=0.
0078          IF(DZ.LT.0.) ROTB=PI
0079      ELSE
0080          ROTB=ATAN2(DY,DZ)
0081      ENDIF
0082      C
0083      ENDIF
0084      C
0085      C      SELECT MAXIMUM G TURN
0086      C
0087      GLEVLB=0.95
0088      TPOSB= 2.0
0089      CAS=VFLB*0.5925
0090      CAS=TAS*SQRT(RHOB/0.0023768)
0091      VCORNH=330.
0092      IF (CAS .GT. VCORNH) GELVLB=LODSTB/LODMXB
0093      ICMNWH=1
0094      C
0095      C      CHECK FOR MANEUVER TERMINATION CONDITIONS
0096      C
0097      IF(TNOW .GE. TQUIT)THEN
0098          ISLCTR=0
0099      ENDIF
0100      C
0101      C      FLAMB=LOSANB*DEGRD
0102      C      IF(RRATE .GT.0.) THEN
0103      C          ISLCTR=0
0104      C      ENDIF
0105
0106      RETURN
0107      END
```

```

0001      SUBROUTINE CLIMB(XEA, YEA, ZEA, XEDOTA, YEDOTA, ZEDOTA, DMTRXA)
0002      C
0003      COMMON/RNDMAN/ MLDECS, MANINI, ISLCTR
0004      C
0005      COMMON/CNSTNS/DT, TBEGN, TNOW, PI, PIDV2, PIDV4, TWOP1, DEGRD, RADDG, G,
0006      1          VAR(20), IVAR(20), TEND
0007      C
0008      COMMON/CONTRL/MSTOP, IPRINT
0009      C
0010      COMMON/COMNDB/ ICMNWB, GLEVLB, ROTB, MANVRB
0011      C
0012      COMMON/DATA1B/XEB, YEB, ZEB, XEDOTB, YEDOTB, ZEDOTB, XEDDTB, YEDDTB,
0013      1          ZEDDTB, PS1B, THETAB, PH1B, UB, VB, WB, PB, QB, RB, A1B, A2B,
0014      2          A3B, A4B, VELB, VHORB
0015      C
0016      COMMON/DATA2B/ALFAB, BETAB, CBARB(3,3), CDB, CLALFB, DMTRXB(3,3), DRAGB,
0017      1          LIFTB, LODMXB, LODSTB, MACHB, RHOB, SPECEB, SB, THRSTB,
0018      2          PSUBSB, TPOSB, INIZB, WEITB, CSB, CLB, PS1BRB, THETBR,
0019      3          AN1B, AN2B, AN3B, MASSB
0020      C
0021      REAL          LIFTB, LODMXB, LODSTB, MACHB, MASSB
0022      C
0023      COMMON/POINTP/XEAIM, YEAIM, ZEAIM, PCOMB, QCOMB, RCOMB, IPOINT
0024      C
0025      DIMENSION CMPL(3,3)
0026      C
0027      COMMON/RELVAR/LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,
0028      1          LSDOTA, LSDOTB, DEVANA, DEVANB, DVDOTA, DVDOTB,
0029      2          RANGE, RRATE, XAINB, YAINB, ZAINB, XBINA, YBINA,
0030      3          ZBINA, ANGOFA, ANGOFB
0031      C
0032      REAL          LOSELA, LOSELB, LOSAZA, LOSAZB, LOSANA, LOSANB,
0033      1          LSDOTA, LSDOTB
0034      C
0035      C      THIS SUBROUTINE GENERATES COMMANDS TO THE B AIRCRAFT TO
0036      C      CLIMB IN THE VERTICAL PLANE FOR A SPECIFIED TIME.
0037      C      THIS CAN BE USED TO FORCE AN OVERSHOOT OF THE A AIRCRAFT
0038      C      WHEN IT IS DIVING ONTO B
0039      C
0040      C
0041      C      THIS MANEUVER SHOULD BE INITIATED UNDER THE FOLLOWING CONDITIONS:
0042      C      RANGE <=3500 FT; EPSD
0043      C      -----
0044      C          INITIALIZATION SECTION
0045      C
0046      C      IF(MANINI.EQ.-1)THEN
0047      C          TIMREQ=5.
0048      C          MLDECS= INT(TIMREQ/DT)
0049      C          TINIT= TNOW
0050      C          TQUIT= TNOW+TIMREQ
0051      C          MANINI=0
0052      C      ENDIF
0053      C
0054      C      ROTB=0.
0055      C
0056      C      SELECT LOW G IN ORDER NOT TO LOSE TOO MUCH ENERGY
0057      C      ELSE SELECT HIGH G IN ORDER TO GAIN ALTITUDE RAPIDLY
  
```

CLIMB

17-Jul-1987 14:01
17-Jul-1987 14:01

```
0058 C
0059     GLEVLH=0.95
0060     TPOSB= 2.0
0061     ICMNWH=1
0062 C
0063 C CHECK FOR MANEUVER TERMINATION CONDITIONS
0064 C
0065     IF(TNOW .GE. TQUIT)THEN
0066         ISLCTR=0
0067     ENDIF
0068 C     IF(RHATE .GT.0.) THEN
0069 C         ISI.CTR=0
0070 C     ENDIF
0071
0072     RETURN
0073     END
```

COMMAND QUALIFIERS

FORTRAN/LIST/SHOW= NOMAP APPENDIX

```
/CHECK=(NOBOUNDS, OVERFLOW, NOUNDERFLOW)
/DEBUG=(NOSYMBOLS, TRACEBACK)
/STANDARD=(NOSYNTAX, NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR, NOINCLUDE, NOMAP, NODICTIONARY, SINGLE)
/WARNINGS=(GENERAL, NODECLARATIONS, NOULTRIX)
/CONTINUATIONS=19 /NOCROSS_REFERENCE /NOD_LINES /NOEXTEND_SOURCE /F77
/NOG_FLOATING /14 /NOMACHINE_CODE /OPTIMIZE
```

COMPIIATION STATISTICS

```
Run Time:          12.01 seconds
Elapsed Time:      13.04 seconds
Page Faults:       1071
Dynamic Memory:    552 pages
```




Report Documentation Page

1. Report No. NASA CR-4160		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Rule-Based Air Combat Simulation				5. Report Date September 1988	
				6. Performing Organization Code	
7. Author(s) G.H. Burgin and L.B. Sidor				8. Performing Organization Report No. H-1501	
				10. Work Unit No. RTOP 505-66-11	
9. Performing Organization Name and Address Titan Systems, Inc. P.O. Box 12139 La Jolla, CA 92037				11. Contract or Grant No. NAS2-11990	
				13. Type of Report and Period Covered Contractor Report—Final	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				14. Sponsoring Agency Code	
15. Supplementary Notes NASA Technical Monitor: Eugene L. Duke, NASA Ames Research Center, Dryden Flight Research Facility, Edwards, California 93523-5000.					
16. Abstract This report documents an improved version of the Adaptive Maneuvering Logic (AML) program for air-combat maneuvering. The document details the modifications and improvements incorporated into the AML program.					
17. Key Words (Suggested by Author(s)) Adaptive maneuvering logic Air combat maneuvering Computer simulation			18. Distribution Statement Unclassified—Unlimited Subject category 61		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 136	22. Price A07

NASA FORM 1626 OCT 86

**For sale by the National Technical Information Service, Springfield, Virginia 22161.*

NASA-Langley, 1988