NASA Contractor Report 172445

Enhanced TCAS II/CDTI Traffic Sensor Digital Simulation Model And Program Description

Tsuyoshi Goka

CONTRACT NAS1-16135
October 1984

NASA

ENTER:
3 1 1 RN/NASA-CR-172445
DISPLAY 03/2/1
85N10927*# ISSUE 2 PAGE 136 CATEGORY 4 RPT#: NASA-CR-172445 NAS 1.26:172445 CNT#: NAS1-16135 84/10/00 175 PAGES UNCLASSIFIED DOCUMENT

UTTL: Enhanced TCAS 2/CDTI traffic Sensor digital simulation model and program description TLSP: Final Report

AUTH: A/GOKA, T.

CORP: Analytical Mechanics Associates, Inc., Mountain View, Calif. AVAIL.NTIS SAP: HC A08/MF A01

MAJS: /*AIR TRAFFIC CONTROL/*COLLISION AVOIDANCE/*COMPUTER PROGRAMS/* COMPUTERIZED SIMULATION/*MATHEMATICAL MODELS

MINS: / CONSTRAINTS/ DIGITAL DATA/ FORTRAN/ PROGRAM VERIFICATION (COMPUTERS)/ REAL TIME OPERATION

ABA: M.A.C.

ABS: Digital simulation models of enhanced TCAS 2/CDTI traffic sensors are developed, based on actual or projected operational and performance characteristics. Two enhanced Traffic (or Threat) Alert and Collision Avoidance Systems are considered. A digital simulation program is developed in FORTRAN. The program contains an executive with a semireal time batch processing capability. The simulation program can be interfaced with other modules with a minimum requirement. Both the traffic sensor and CAS logic modules are validated by means of extensive simulation runs. Selected validation cases are discussed in detail, and capabilities and

ENTER:

NASA Contractor Report 172445

Enhanced TCAS II/CDTI Traffic Sensor Digital Simulation Model And Program Description

Tsuyoshi Goka

Analytical Mechanics Associates, Inc.
Mountain View, California 94043

Prepared for
Langley Research Center
National Aeronautics and Space Administration
under Contract NAS1-16135

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

October 1984

N85-10927#

Foreword

This effort for developing enhanced TCAS II/CDTI traffic sensor models was supported under NASA Contract No. NAS1-16135 by Langley Research Center, Hampton, Virginia. The project technical monitor was Dr. Roland L. Bowles of Langley Research Center. Technical discussions with Dr. Bowles and David H. Williams of NASA Langley, and Joseph Fee of FAA Headquarters are gratefully acknowledged.

ENHANCED TCAS II/CDTI TRAFFIC SENSOR
DIGITAL SIMULATION MODEL
AND
PROGRAM DESCRIPTION

Tsuyoshi Goka

Analytical Mechanics Associates, Inc.
Mountain View, California 94043

SUMMARY

The objectives of this project were to develop digital simulation models of enhanced TCAS II/CDTI traffic sensors based on actual or projected operational and performance characteristics. Two enhanced traffic (or threat) alert and collision avoidance systems were considered. The main focus is a system based on the FAA/Bendix design which dates back to the so-called Beacon Collision Avoidance System (BCAS) concept. The other system is built by Dalmo Victor Division of Textron's Bell Aerospace Co. based on the FAA/MIT active BCAS concept.

Based on the engineering model, a digital simulation program was developed in FORTRAN. The program contains an executive with a semi-real time batch processing capability. Thus, the simulation program can be interfaced with other modules with a minimum requirement. The program is described in detail.

The TCAS systems are not specifically intended for Cockpit Display of Traffic Information (CDTI) applications. However, these systems are sufficiently general to allow implementation of CDTI functions within the real systems' constraints.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES - Continued

LIST OF ACRONYMS AND ABBREVIATIONS

AGL	Above Ground Level
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
BCAS	Beacon Collision Avoidance System
CAS	Collision Avoidance System
CDTI	Cockpit Display of Traffic Information
CPA	Closest Point of Approach
EFR	Electronic Flight Rules
EHSI	Electronic Horizontal Situation Indicator
FAA	Federal Aviation Administration
INS	Inertial Navigation System
LOT	Level Occupancy Time
LST	Level Switching Time
MFD	Multi-Function Display
MSL	Mean Sea Level
Mode A	An ATCRBS reply format with aircraft identity
Mode C	An ATCRBS reply format containing an altitude report
Mode S	New Selectable data formats used by a Mode S transponder
MOPS	Minimum Operational Performance Standards
NASA	National Aeronautics and Space Administration
NED	north-east-down (a standard coordinate system)
PWI	Proximity (or Pilot) Warning Indicator
RF	Radio Frequency
TCAS	Traffic (or Threat) Alert and Collision Avoidance System
VFR	Visual Flight Rules

I

INTRODUCTION

NASA Langley Research Center is pursuing a research effort concerning the Cockpit Display of Traffic Information (CDTI) concept. The CDTI is a device which presents information to the pilot and crew depicting the status of surrounding traffic including position and velocity states. The general CDTI avionics consists of a "traffic sensor", a pilot interface unit, a CDTI signal/data processor, and a display unit.

The pilot interface unit can be a simple on/off switch, a shared keyboard, or an elaborate touch sensitive screen. The CDTI signal/data processor can be a digital data link between the traffic sensor and signal operator or a bona fide computing module performing data editing, traffic selection or similar functions. The computational module may be imbedded, for example, in a navigation computer. A cathode ray tube (CRT) or a flat panel plasma display unit seems to be the most suitable display medium.

Again, this system could be dedicated, shared, or imbedded in with other functions. It would most logically be included in an Electronic Horizontal Situation Indicator (EHSI) or a Multi-Function Display's (MFD) symbology. It is also possible that the CDTI display could be included in a general "Air Traffic Control" display unit in a futuristic environment if one were to extrapolate the current digital avionics and up-and-down link communication technology and the centralized and automated ground control philosophy.

The CDTI functions include both passive (monitoring) and active roles. The latter is sometimes referred to as Electronics Flight Rules (EFR) as compared to the VFR or IFR control environment. Some of the envisioned CDTI functions are listed below:

Traffic Monitoring Roles:

General Traffic Monitoring

Longitudinal Separation of Arrivals

Independent Parallel Approaches

Oceanic Route Separation

Active Control Roles:

Arrival Merging

Arrival In-trail Spacing Control

Enroute Passing and Crossing

Severe Weather Avoidance

As can be seen from the above list of functions, the main utility of the CDTI is to provide the pilot with a strategic information base which allows him to make appropriate decisions. This is counter to other avionics which have well defined tactical functions. (For example, the MLS (Microwave Landing System) provides deviation signals from reference for the pilot to nullify.) This means that the CDTI utility can not be measured by quantitative attributes such as accuracy or sampling frequency alone. This utility also involves the question of how the pilot uses the information base; i.e., the human factors aspect.

Therefore, various CDTI issues must be resolved via active, pilot-in-the-loop simulation studies. This points out the importance of developing a realistic simulation facility to reflect a "real" full workload flight environment. This issue is not limited to the "traffic sensor" modeling, but it includes a host of other simulation elements.

Because there seems to be no official impetus to develop a CDTI traffic sensor per se at this time, an experimental sensor simulation model must be developed based on related systems which are currently being developed. The FAA developed Traffic Alert and Collision Avoidance System (TCAS) comes closest to fulfilling various CDTI research needs.

TCAS is strictly an airborne system which provides the aircraft separation protection information independent of the ground ATC system. The FAA plans call for developing two types of TCAS - TCAS I and TCAS II. Within each category, a certain latitude in capability is allowed to satisfy a wide spectrum of user requirements.

The enhanced TCAS II which is capable of obtaining relative bearing measurements between the protected (Own) and surrounding aircraft (Target) may be able to support CDTI applications. The enhanced TCAS II is capable of range and bearing (in addition to the encoded altitude) measurements with a medium degree of accuracy to the extent that a more sophisticated CDTI type display or horizontal collision avoidance logic may be supported. There are two designs in this enhanced TCAS II category. One design, developed by MIT/Dalmo Victor (DV TCAS), is based on the so-called active Beacon Collision Avoidance System (BCAS). The other, developed by Bendix (BX TCAS), is based on the so-called full BCAS concept.

In support of the NASA Langley CDTI research effort, a comprehensive digital simulation model of the enhanced TCAS II based on the Bendix design (BX TCAS) has been developed. The reasons behind this choice are twofold. First, because of its inherent accuracy, other TCAS based traffic sensors can be "emulated" by degrading the BX TCAS model performance. Second, the versatility of the TCAS processor provides a certain modification latitude suitable for CDTI applications.

This report is a companion to our report entitled "Analysis of Estimation Algorithms for CDTI and CAS Applications [1]", which studies the sensor accuracy analysis with respect to CAS logic and CDTI applications. This report is organized as follows. Chapter II provides an engineering description of the BX TCAS II system. Chapter III provides a detailed description of the supporting simulation programs. Chapter IV provides the Input/Output requirements for the BX TCAS module. Chapter V discusses a set of simulation validation cases. The validation cases were selected from the draft TCAS II Minimum Operating Performance Standards by the

RTCA SC-147 [2], Chapter V provides the conclusions. Appendix A lists all the major commons used in the simulation program. Appendix B discusses the major functional departure for the Dalmo Victor TCAS II (DV TCAS) design. It also includes a set of program modifications in order to incorporate system peculiarities.

II

ENHANCED TCAS II DESCRIPTION

Background

TCAS provides information for safe aircraft separation independent of the ground ATC sensors. Except for TCAS I (which operates on a passive listening mode in principle), TCAS contains an (active) transmitter as well as an ICAO approved Model S transponder. Target relative positional measurements are acquired by means of radio frequency (RF) transactions of 1030 MHz interrogation transmission and 1090 MHz reply reception. (Note that the RF formats are compatible with the current ATC surveillance system requirements.)

Four major submodules -- surveillance, state estimation, collision avoidance logic, and pilot warning and display -- further process the RF transactions to satisfy the aircraft separation requirements. Figure 1 depicts the functional break-down of these four submodules*. Each of these submodules will be discussed in subsequent sections.

A relative bearing measurement capability is not a requirement for minimum TCAS II; the bearing capability constitutes the "enhancement". Enhanced TCAS II is capable of range and bearing (in addition to the encoded altitude) measurements with a medium degree of accuracy to the extent that a more sophisticated CDTI type display or horizontal collision avoidance logic may be supported.

Two systems in this category are at various stages of development or testing. Flight evaluation of the DV TCAS will begin shortly in scheduled operational environment [3]. The TCAS generated advisories will be displayed to the flight crew. The BX TCAS engineering unit is currently undergoing an

---

* It is noted that the time sequence of events over one TCAS cycle is closely correlated with the top to bottom items, i.e., the collision avoidance logic requires the latest available state estimates which, in turn, depend on the latest available surveillance data.

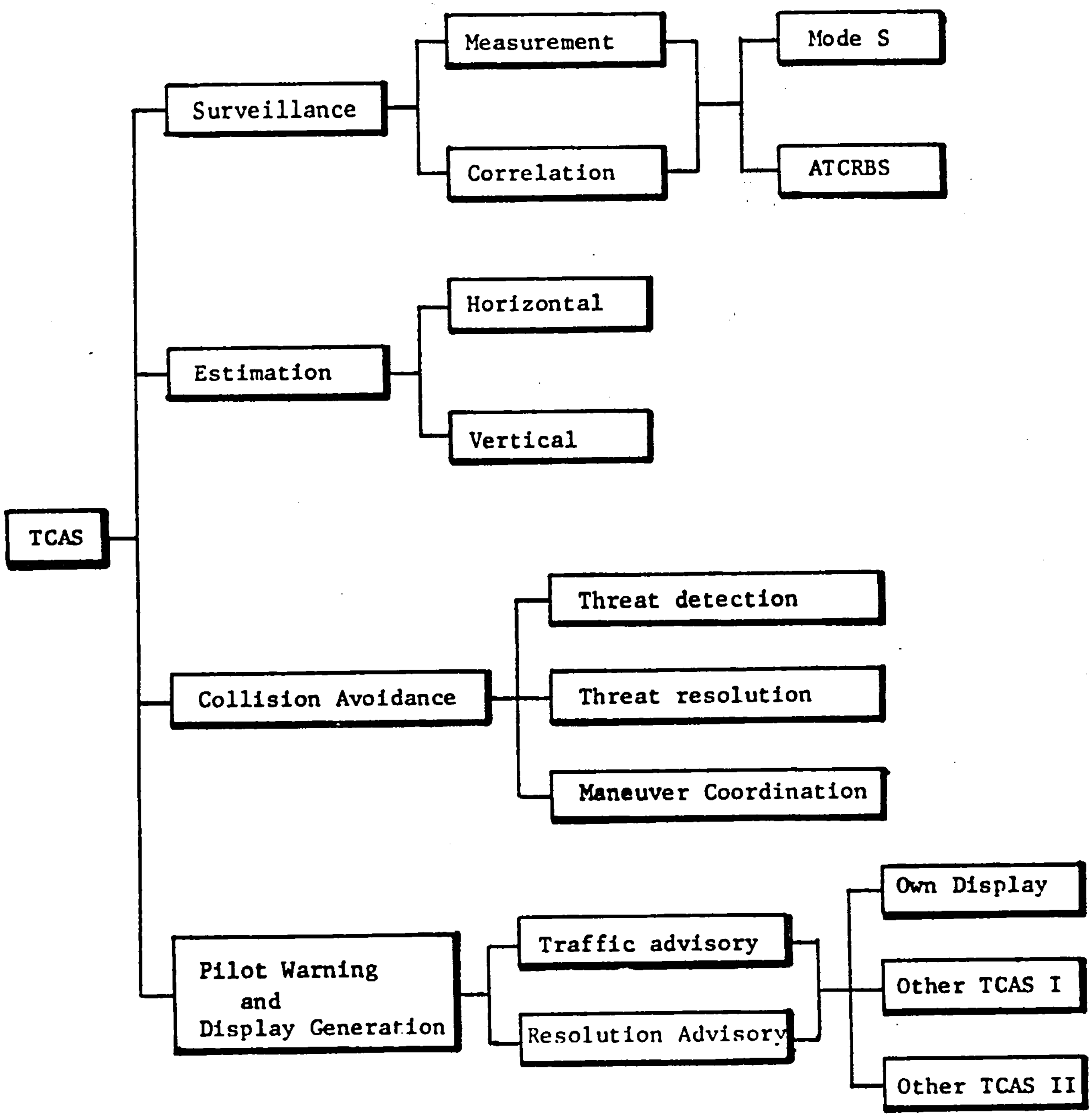


Figure 1. TCAS II Functional Elements.

extensive flight test evaluation [4]. Table 1 shows the over-all performance and operational characteristics of these two systems. Table 2 shows the consensus of engineering opinion indicating the TCAS functional breakdown and bearing accuracy.

In subsequent sections, functional descriptions of the four submodules are given based on the BX TCAS design. It seems to be more suitable than the DV TCAS for the CDTI applications in terms of coverage volume, accuracy, and versatility. The material presented here is condensed, mainly from the draft TCAS II MOPS and Bendix reports [1,5,6,7,8].

Table 1. Operational Characteristics of Candidate TCAS CDTI Sensors

	DV TCAS II	BX TCAS II
Mode Base	• Aircraft based • Independent of ground	• Aircraft based • Independent of ground
Coordinate System	• Aircraft fixed • Aircraft Body Axes	• Aircraft fixed • NED Local Level
Coverage	• Within 10 nmi (max) of Own	• Within 25 nmi (max) of Own
Accuracy	• range = 100 ft rms • bearing = 6-8 deg rms • altitude = 100 ft resolution	• range = 100 ft rms • bearing = 2 deg rms • altitude = 100 ft resolution
Sampling Rate	• 1 sec • reliability = f(range)	• 1-8 sec (variable) • reliability high
Comments	• $\alpha\beta$ tracker in r-β axes • LO tracker (altitude) • Estimate not stabilized with respect to Own attitude • Global protection . Medium traffic density	. $\alpha\beta$ tracker in x-y axis . LO tracker (altitude) . Estimate stabilized with respect to Own attitude, i.e., ϕ, θ, ψ are available within the TCAS processor . Global protection . High traffic density

Table 2. Functional Breakdown of TCAS II with Respect to Bearing Accuracy

	Bearing Accuracy (deg)	Function
Enhanced TCAS II	4 - 8	o Vertical Resolution (bearing modified) o PWI or CDTI
	0.6 - 2	o Horizontal and Vertical Resolution o CDTI

Surveillance

The surveillance process begins by the TCAS transmitting 1030 MHz interrogation signals and by receiving 1090 MHz replies from nearby transponders (Mode A, ATCRBS or Mode S) or by listening for Mode S squitter or air-to-ground transmission signals at 1090 MHz. The position measurements are then computed by the internal signal processor as follows:

(a) range - by the time duration between the interrogation and the corresponding reply reception, accounting for the transponder delay;

(b) bearing - by computing the angle-of-arrival from the phase distribution among several antennas; and,

(c) altitude - by decoding the Mode C altitude code contained in the reply. (For a Mode A only target, this will be nonexistent.)

The surveillance characteristics of the BX TCAS are somewhat similar to that of the ground based Mode S beacon sensor. Because a large number of transmitters in a small local will cause interference resulting in synchronous garble, fruit or false squitter detection, there are three techniques (in addition to the mono-pulse technique) to overcome the high density problem. One is the interrogation antenna directivity; the second is the so-called "whisper/shout" signal power level sequencing; and the third is the interrogation rescheduling if a reply is missed or garbled. The antenna beam width is 22 1/2 deg; however, by repeating the transmission four times and each time sliding the beam center by 5.625 deg, the effective beam width becomes 5.625 deg. The beam pointing and rescheduling as well as several levels of whisper/shout power sequencing are controlled by internal digital processors based on the internal track file, Own aircraft's orientation, and the ATCRBS/Mode S transponder mix. The task is facilitated by the fact that the beam is "stabilized" with respect to roll, pitch and yaw attitude angles.

Target Track Establishment The function of establishing the target track consists of two subfunctions: relative position measurement and the associated target correlation. The position measurement refers to the actual RF activities between Own's transmitter/receiver and Target's transponders and the subsequent signal processing to extract the position measurement. The correlation process (also referred to as the track acquisition or establishment) establishes the correspondence between a set of measurements and a particular tracked aircraft.

It is simple to track Mode S equipped aircraft because of the uniquely assigned discrete address in the reply format (which is also stored in the TCAS unit). The task of correlating between the measurements and the tracked aircraft is not as simple if the target is Mode C or Mode A equipped. Also, for the narrow beam system (BX TCAS), the correlation process is simpler than for the omni-directional or wide angle system, because the number of replies corresponding to an interrogation is generally much smaller. However, even the narrow beam width and the rescheduling capability present problems if two or more aircraft are clustered in close proximity.

A "gating" technique is used for the purpose of separating targets. If the current measurement falls within certain threshold values (which define the gate) of the predicted value of an aircraft in the track file, then the measurement is assigned to that aircraft, and the corresponding track file is updated. If the measurement does not correspond consistently (5 - 10 sec) within a gate to any existing aircraft in the track file, then a new track file is started for that set of measurements. Conversely, if none of the measurements consistently correspond to an aircraft in the file, then that aircraft is judged outside the beam reach, and hence, it is deleted from the track file.

The state estimates play an important role in computing the prediction since the last valid measurement time. The estimation algorithms used for this purpose are classical simple alpha-beta trackers. Accuracy is not too critical because the threshold values are sufficiently large to account for estimation errors.

There are still problems with the TCAS system proto-type which is currently being flight tested. Track establishment of an "image target" due to multi-path of the real target is one of the remaining major problems.

Coverage Volume The Own protected airspace provided by the system is physically limited because of transmitter power output and/or receiver reception sensitivity limitations. Also, the beam pattern due to the antenna configuration comes into play, especially for the vertical coverage. The maximum beam reach is estimated to be 35 nmi; this is at the highest sensitivity level. Within this distance, the 1030 MHz transmission signals can be distinguished from the ambient RF noise with a certain reliability.

The vertical limitation is due to the elevation beam shape. The top mounted antenna assembly is designed to provide coverage of approximately five (5) deg below and 23 deg above the antenna plane. The system may or may not include a similar antenna assembly located at the bottom of the fuselage.

To limit unnecessary RF activity, especially in a high traffic density area, the BX TCAS relies on an "artificial" boundary generated by the beam control microprocessor. The volume is dynamically computed and is defined by the relative range and relative altitude. In case of a Mode A transponder, the range defines the volume. Furthermore, the volume is subdivided into two regions - "acquisition" and "track".

The acquisition region is provided mathematically by

$$\Delta r \leq \Delta r_{acq}, \quad \text{and} \quad |\Delta h| \leq \Delta h_{acq} . \tag{1}$$

Here Δr_{acq} is the (acquisition) range threshold (nominally 17 nmi for Mode A and Mode C transponders and 25 nmi for Mode S transponders), and Δh_{acq} is the altitude threshold (nominally 6,000 ft but ignored for Mode A transponder).

The track region is provided mathematically by

$$\Delta r \leq \Delta r_{trk}, \quad \text{and} \quad |\Delta h| \leq \Delta h_{trk} . \tag{2}$$

Here Δr_{trk} is the (track) range threshold which is computed dynamically, and Δh_{trk} is the altitude threshold given by

$$\Delta h_{trk} = 3750 + |\dot{z}| \cdot 45 \text{ (ft)} . \tag{3}$$

The quantity Δr_{trk} is determined based on the relative bearing as well as Own ground speed and altitude. The equation defining for this term is given by

$$\Delta r_{trk} = \mathrm{Max}\{\Delta r_{lim}, (\hat{V}_0 \cos\Delta b + V_{Max}) T + \Delta r_s\} \tag{4}$$

Here,

$\hat{V}_0$ = Own ground speed, in kt,

T = "closure" time constant = 1/80 hr = 45 sec, and

Δb = relative bearing with respect to Own's body axes

= $\tan^{-1} (\Delta y_B/\Delta x_B)$

Δr_s = 1.65 nmi.

Δr_{lim} and V_{Max} are computed based on Own altitude $\hat{z}_o$ (ft), according to

$$\Delta r_{lim} = \begin{cases} 5 \text{ nmi}, & \text{if } \hat{z}_0 \leq 10{,}000 \text{ ft} \\ 10 \text{ nmi}, & \text{otherwise} \end{cases} \tag{5}$$

and

$$V_{Max} = \begin{cases} 250 \text{ kt}, & \hat{z}_0 \leq 3{,}000 \text{ ft.} \\ \hat{z}_0/20 + 100 \text{ kt}, & 3{,}000 < \hat{z}_0 \leq 10{,}000 \text{ ft.} \\ 600 \text{ kt}, & \hat{z}_0 \geq 10{,}000 \text{ ft.} \end{cases} \tag{6}$$

Figure 2 shows the track regions corresponding to three Own ground speeds at three Own altitude levels. It is noted that the Own forward direction is given a heavier protection weight.

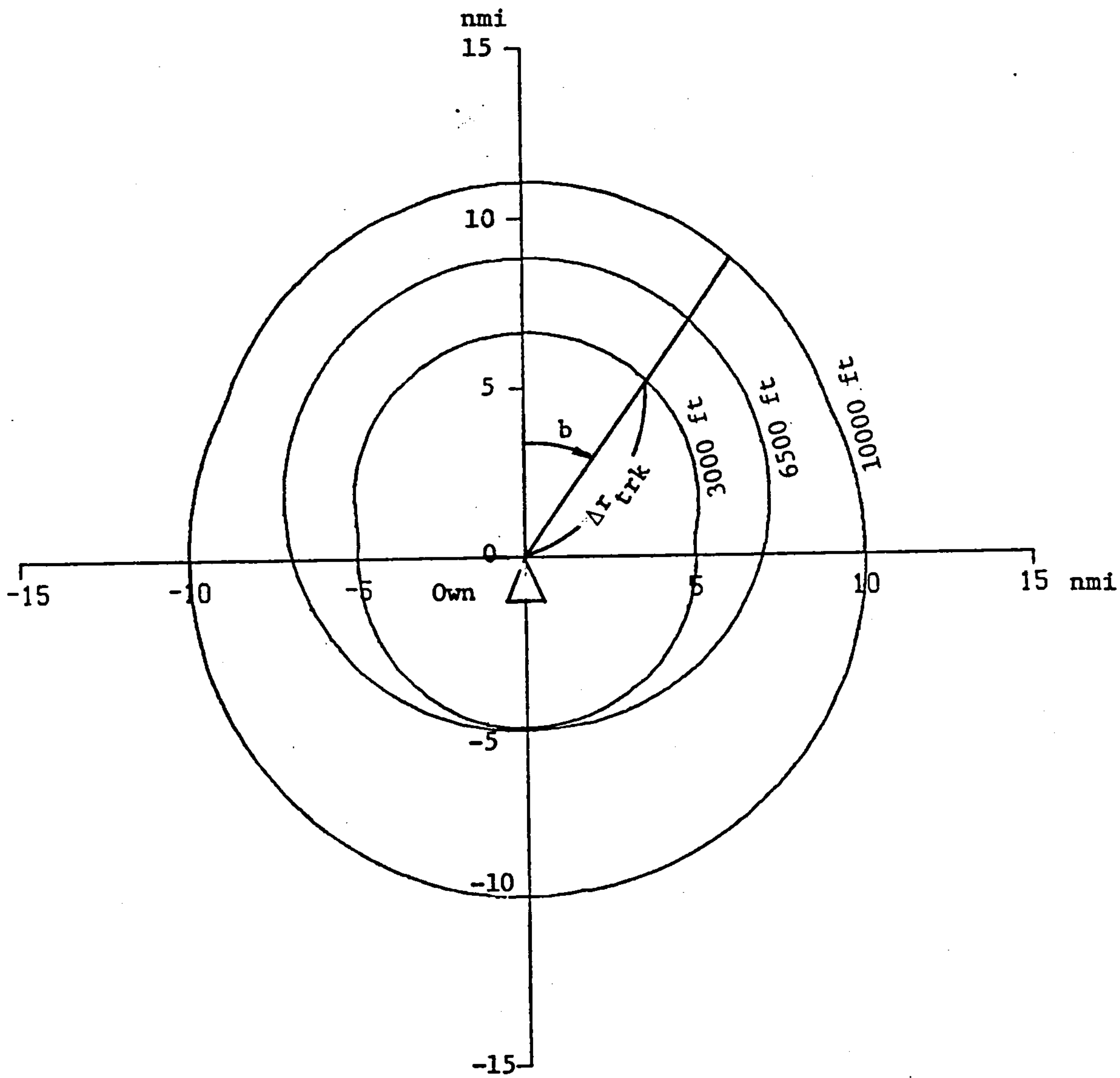
Interrogation Schedule Logic The antenna pointing controller module schedules the interrogation and reception timing (i.e., surveillance scheduling). The surveillance operation depends on two factors. One is the transponder type - Mode C or Mode S. The other is the operational mode - search/acquisition or track. The antenna dwell at a given azimuth angle is divided into one passive and three active processes. The active ones include: (a) ATCRBS transmissions to search for new targets (targets which are not in the internal track file); (b) ATCRBS transmissions for tracking existing targets (in the internal track file); and (c) Mode S transmissions for tracking Mode S equipped targets. The passive process consists of possibly listening for Mode S squitters or Mode S replies to ground interrogations.

The time interval between ATCRBS search interrogations, Δt_s, is computed according to the formula:

$$\Delta t_s = \min \left\{ 16 \text{ sec}, \frac{3600\, \Delta r_s}{V_{Max} + V_0 \cos\Delta b} \right\} \tag{7}$$

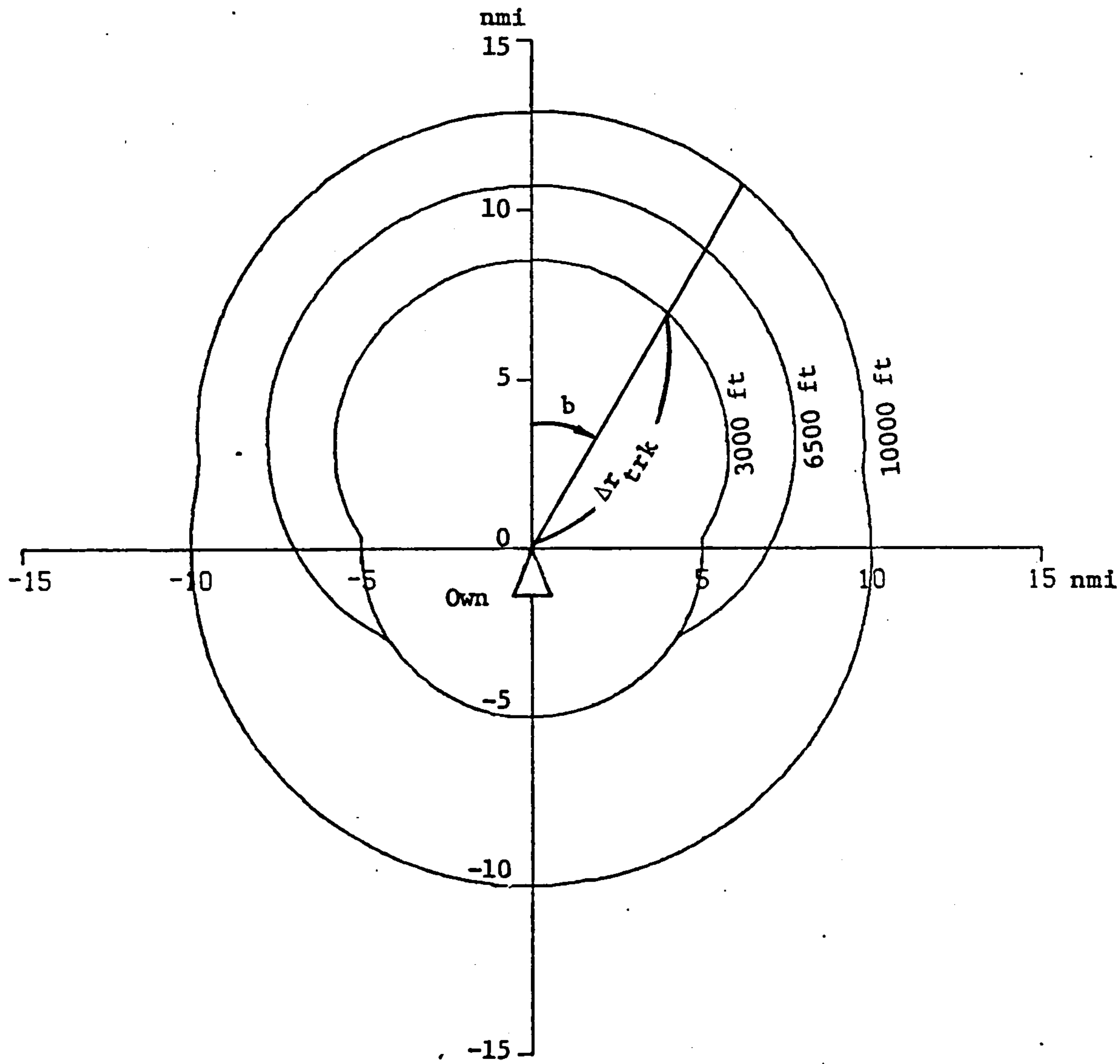
where the variables have been previously defined.

The ATCRBS track interrogations are made for those targets lying inside the track volume satisfying Eq. (2). The ATCRBS track interrogation time interval, Δt_T, is computed based on the predicted relative motion of



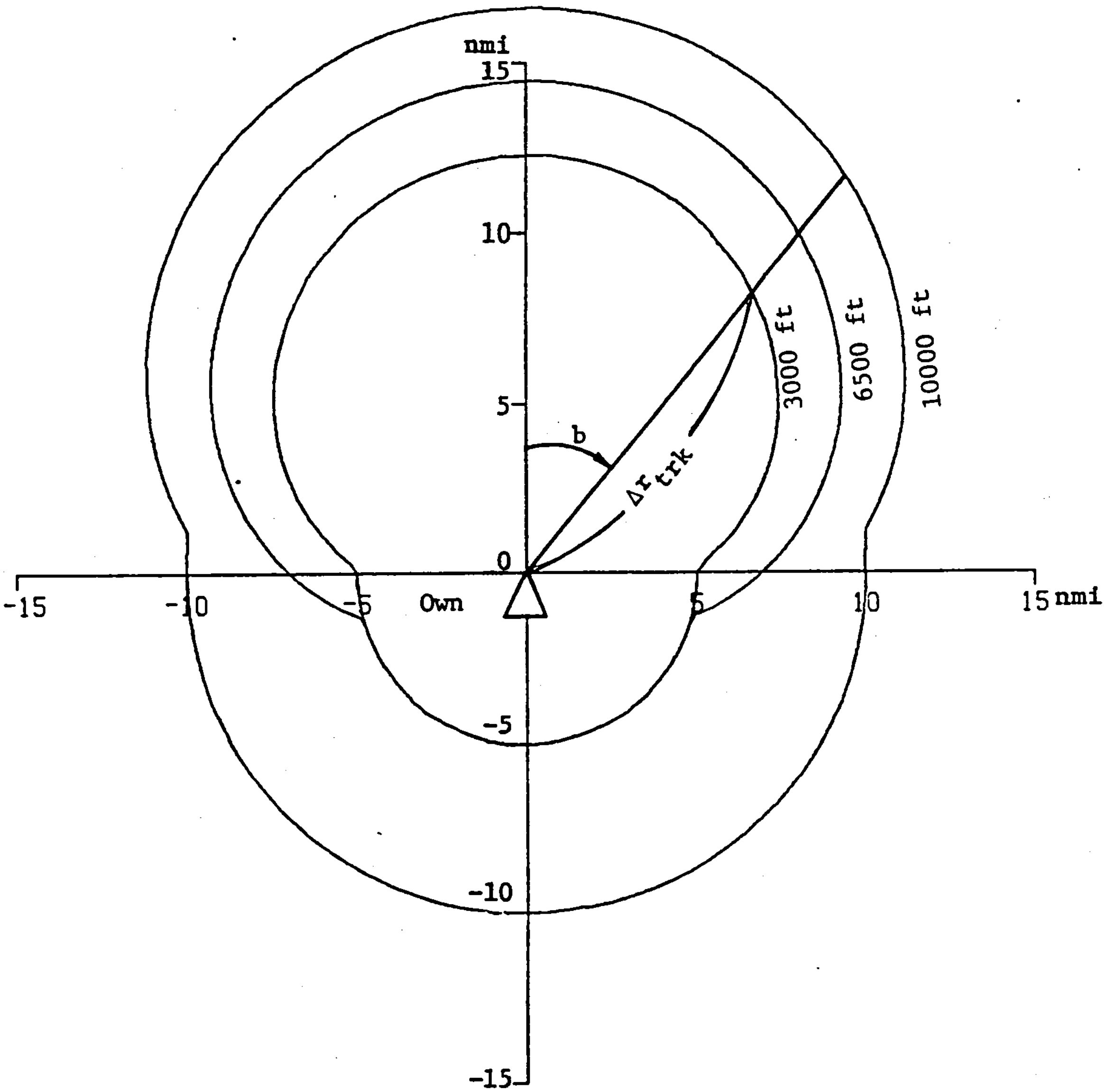
(a) V_0 = 150 kt

Figure 2. Range Track Regions at Various Own Altitudes.



(b) V_0 = 300 kt

Figure 2. - Continued



(c) V_0 = 600 kt

Figure 2. - Continued

the target. It is given by the formula

$$\Delta t_T = \text{Max}\ \{1 \text{ sec}, \min\ \{t_1, t_2, t_3, 8 \text{ sec}\}\}. \tag{8}$$

Here,

t_1 = the number of seconds it will take the target to move 3 deg in bearing,

t_2 = the number of seconds it will take the target to move 1000 ft in range, and

t_3 = the number of seconds it will take the target to move 250 ft in altitude.

Interrogation scheduling logic for targets with Mode A transponders is presumed to be similar to the ATCRBS interrogations. However, the target altitude logic would be ignored. Furthermore, if Own altitude is too high compared to the FAA's altitude encoding transponder requirements, then the Mode A targets can safely be ignored.

The interrogation scheduling logic for targets with Mode S transponders are similar to the ATCRBS transponder case. When a new Mode S target is detected by squitter listening, it is interrogated. If it is within the acquisition range, Δr_{acq}, a track is initiated. Mode S equipped targets inside the track volume are interrogated at the same rates as if they were ATCRBS targets. Those targets which are outside of the volume but within Δr_{acq} of Own are tracked at a regular interval of eight (8) sec.

If a target (either ATCRBS or Mode S equipeed) is closer than 6000 ft or if it has been declared a preliminary threat by the threat direction logic, the target track update rate is one (1) sec.

If replies are missing under repeated interrogation of a tracked target, the track is dropped. In addition, ATCRBS tracks will be dropped when their coasted position (position extrapolated by dead reckoning) lies outside of the track volume. A Mode S track will be dropped when the expected range is greater than Δr_{acq}. Table 3 summarized the surveillance function.

Table 3. Summary of BX TCAS Surveillance Function

	Target Search or Acquisition	Target Track
ATCRBS (Mode A or C) Transponder	IF ($\Delta r \le \Delta r_{acq}$ .AND. $\lvert\Delta h\rvert \le \Delta h_{acq}$) THEN $\Delta t_s = \min\{\Delta t, 16 \text{ sec}\}$ $\Delta t = \dfrac{3600 \cdot \Delta r_s}{V_{Max} + V_0 \cos\Delta_b}$	(i) IF (threat .OR. $\Delta r \le$ 1nmi), THEN Δt_T = 1 sec, or (ii) IF ($\Delta r \le \Delta r_{trk}$ .AND. $\lvert\Delta h\rvert \le \Delta h_{trk}$), THEN Δt_T = Max{1 sec, min $\{t_1, t_2, t_3$, 8 sec}}
Mode S Transponder	(i) Passive listening for Mode S squitters or replies to the ground Mode S interrogations, or (ii) IF ($\Delta r \le \Delta r_{acq}$ .AND. $\lvert\Delta h\rvert \le \Delta h_{acq}$), THEN Δt_s = 8 sec	(i) IF (threat .OR. $\Delta r \le$ 1nmi), THEN Δt_T = 1 sec, or (ii) IF ($\Delta r \le \Delta r_{trk}$ .AND. $\lvert\Delta h\rvert \le \Delta h_{trk}$), THEN Δt_T = Max{1 sec, min $\{t_1, t_2, t_3$, 8 sec}}

$\Delta r_{acq} = \begin{cases} 17 \text{ nmi (Mode A and C)} \\ 25 \text{ nmi (Mode S)} \end{cases}$; Δh_{acq} = 6000 ft (Mode C and S)

$\Delta r_{trk} = \text{Max}\left\{ \Delta r_s + (\hat{V}_0 \cos \Delta b + V_{Max})\, T,\ \Delta r_{lim} \right\}$; $\Delta h_{trk} = 3750 + |\hat{\dot{z}}_0| \cdot 45$

$\Delta r_{lim} = \begin{cases} 5 \text{ nmi}, & \hat{z}_0 < 10{,}000 \text{ ft} \\ 10 \text{ nmi}, & \text{otherwise} \end{cases}$

$V_{Max} = \hat{z}_0/20 + 100$; with $250 \le V_{Max} \le 600$

Δr_s = 1.65 nmi

$\hat{V}_0, \hat{z}_0, \hat{\dot{z}}$ = Own ground speed, altitude and altitude rate estimates

t_1 = number of seconds for 3 deg bearing change

t_2 = number of seconds for 1000 ft range change

t_3 = number of seconds for 250 ft altitude change

Surveillance Reliability Surveillance reliability is defined as the probability of obtaining a correlated set of measurements for an aircraft within the track file. The exact numbers are not known. The probabilities of 0.99 for Mode S, 0.95 for Mode C, and 0.9 for Mode A transponders seem to represent state-of-the-art.

Estimation

The estimation submodule generates estimates of suitable state variables for the surveillance and CAS submodules based on the relative range and bearing measurements, target altitude report, and Own altitude input. In actual implementation, parts of the estimation function may be distributed in other submodules depending on their needs. For example, a simple low gain alpha-beta tracker may be used in conjunction with the Mode C altitudes reports within the surveillance module. The tracker outputs are used for track establishment and correlation purposes via the previously mentioned altitude gating technique.

On the other hand, the collision avoidance logic requires more accurate estimates. Thus, it contains a separate, more complex non-linear vertical tracker algorithm as an integral part. One question is: Why is the nonlinear tracker not used for the surveillance applications as well? One reason is that the computational time requirements are high. At any one time, the number of threats within the CAS traffic file is limited to 5 or 6, whereas the number of tracks (that may include multi-path images) within the surveillance track file could be ten times as large. With the more time consuming algorithm, all the entries in the traffic file may not be completed within the time allocation of a fraction of one second.

Because the CAS application needs are more acute compared to the surveillance needs in terms of estimation accuracy, the subsequent discussion will emphasize the CAS aspects. (CDTI accuracy requirements are thought to be similar.)

The BX TCAS design relies on north referenced local level x and y components* as the basis for the horizontal position and velocity estimates. This is accomplished by utilizing Own roll, pitch and yaw angles. This approach is discussed in the following sections.

Measurement Accuracy The error characteristics for the BX TCAS in an operational environment are virtually unknown. The following characteristics represent a consensus of the immediate engineering community and are also inferred from a limited number of flight tests. A proto-type model has been in flight tests since January 1984.

Because the interrogation/reply process of this unit is similar to the Mode S ground sensor, it is reasonable to assume that the range error could be as accurate as $\pm$ 50 ft ($+1\sigma$). A standard deviation of $\pm$ 75 ft ($\pm 1\sigma$) is assumed for the simulation.

The bearing error depends on the sharpness of the directional beam and the internal clocking device. It also depends on the reflection (multi-path) characteristics from various points on the target and the Own aircraft fuselage. The consensus value for this error is between ± 0.6 and ± 2 deg ($\pm 1\sigma$). A standard deviation of ± 1.0 deg ($\pm 1\sigma$) is assumed. See Table 2.

The 100 ft quantization due to the encoding process dominates the altitude error. Twenty-five (25) foot seems to be a reasonable standard deviation number for the high frequency error; however, low frequency drift, bias or scale factor errors could be substantially larger with up to a ± 4 % scale factor error not being uncommon.

Coordinate Systems Two coordinate systems are important in BX TCAS sensor geometry. One is a north referenced local level coordinate system attached to the fuselage at the antenna. The other is the orthogonal

* The draft TCAS II MOPS recommends that orthogonal linear dimensions be used rather than range and bearing dimensions, even though ϕ, θ and ψ are not available.

coordinate system attached to the antenna plane, i.e., the aircraft body reference system. Figure 3 depicts the transformation geometry.

The relative bearing is measured with respect to the latter reference (the relative range is coordinate free); whereas the relative position (say, north-east-down) is measured in the local level system.

Using the conventional definition of Euler body angles ϕ, θ and ψ, the transformation (T_{BL}) from the local-level to Own body axes is given by

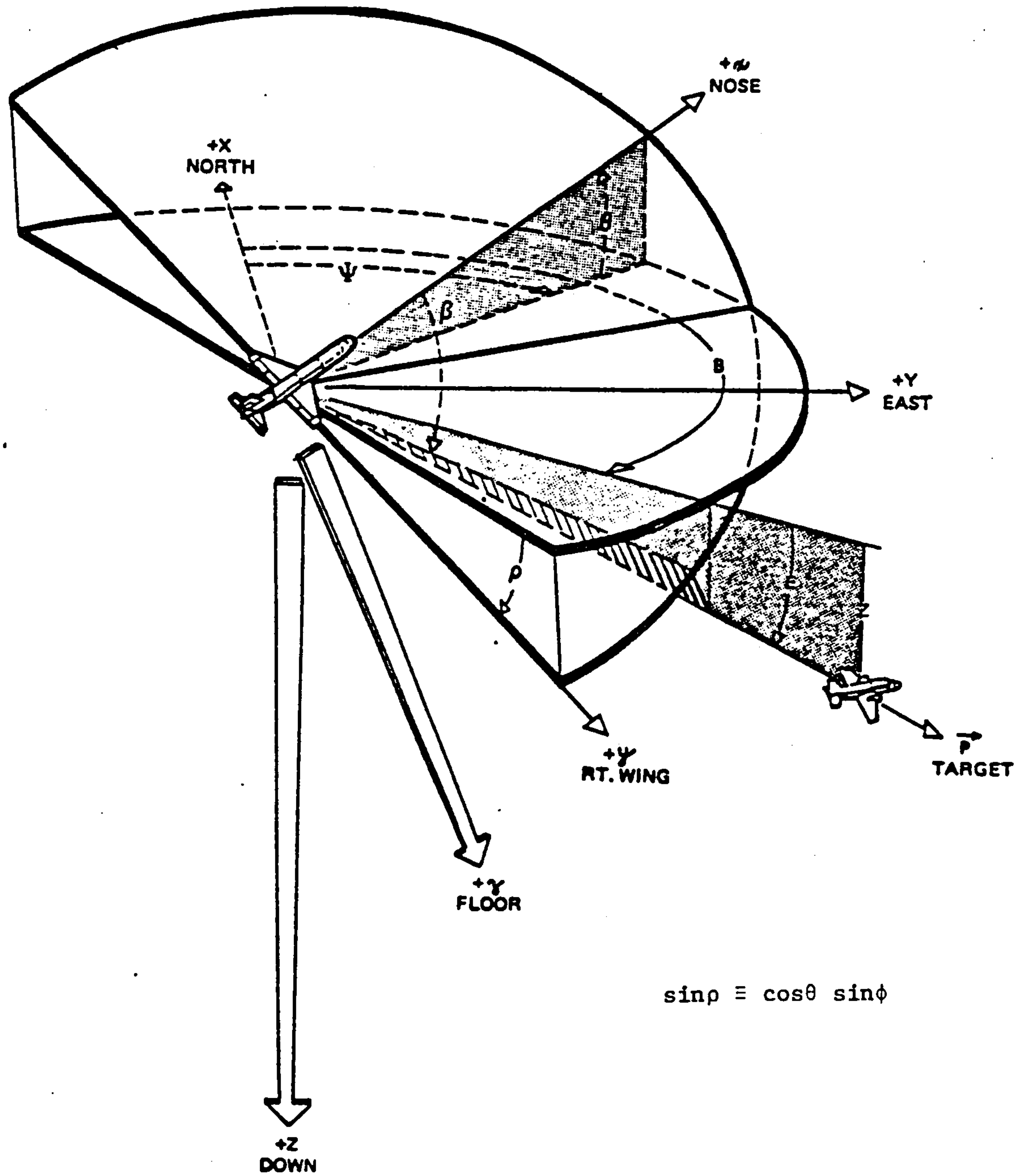
$$T_{BL} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \tag{9}$$

Using this transformation, a relative north-east-down position vector to a target aircraft transforms to one in the body axis as

$$\begin{bmatrix} \Delta x_B \\ \Delta y_B \\ \Delta z_B \end{bmatrix} = T_{BL} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} . \tag{10}$$

The relative bearing and elevation angles to the target are given by

$$\begin{aligned} \Delta b &= \tan^{-1} (\Delta y_B/\Delta x_B) \quad , \\ \Delta e &= \tan^{-1} (\Delta z_B/\Delta r) \quad . \end{aligned} \tag{11}$$



$$\sin\rho \equiv \cos\theta \, \sin\phi$$

Figure 3. TCAS Geometry

Inverse Transformation One requirement is to transform the TCAS measurements to north referenced x and y components. These measurements include the relative range and bearing (Δr and Δb), the target and Own estimated altitudes ($\hat{z}_T$ and $\hat{z}_0$) and the measured Own attitude angles (ϕ, θ and ψ). It is a reverse problem of finding Δb given x, y and z.

The following equation is obtained by using Eq (10):

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = T_{LB} \begin{bmatrix} \Delta x_B \\ \Delta y_B \\ \Delta z_B \end{bmatrix} . \tag{12}$$

Here, T_{LB} is the body-to-local-level direction cosine matrix given by $T_{BL}^{-1} = T_{BL}^{T}$. The body referenced position vector (Δx_B, Δy_B and Δz_B) can be written as

$$\begin{bmatrix} \Delta x_B \\ \Delta y_B \\ \Delta z_B \end{bmatrix} = \Delta r \begin{bmatrix} \cos(\Delta e)\ \cos(\Delta b) \\ \cos(\Delta e)\ \sin(\Delta b) \\ \sin(\Delta e) \end{bmatrix} \tag{13}$$

where Δe is the unknown elevation angle.

Substituting Eq (13) into Eq (12) and writing out the third row equation, it follows that

$$\frac{\Delta \hat{z}}{\Delta r} = A_1 \cos(\Delta e) + A_2 \sin(\Delta e), \tag{14}$$

$$\Delta \hat{z} = \hat{z}_T - \hat{z}_o ,$$

$$A_1 = T_{LB}(3,1) \cos(\Delta b) + T_{LB}(3,2) \sin(\Delta b), \tag{15}$$

$$A_2 = T_{LB}(3,3) ,$$

and $T_{LB}(i,j)$ is the ji^{th} element of the matrix of Eq (9).

Now define

$$
\begin{aligned}
|A| &= [A_1^2 + A_2^2]^{1/2} \quad , \\
C &= \frac{\hat{\Delta z}}{\Delta r} \, |A|^{-1} \quad , \\
A_1' &= A_1 \, |A|^{-1} \quad , \text{ and} \\
A_2' &= A_2 \, |A|^{-1} \quad .
\end{aligned}
\tag{16}
$$

Then, sin(Δe) and cos(Δe) are given by

$$
\begin{aligned}
\sin(\Delta e) &= CA_1' - (1-C^2)^{1/2} \, A_2' \quad , \\
\cos(\Delta e) &= CA_2' + (1-C^2)^{1/2} \, A_1' \quad .
\end{aligned}
\tag{17}
$$

The body referenced positions (Δx_B, Δy_B and Δz_B) can be computed by using Eq (13). From these the north referenced relative positions, (Δx and Δy) can be directly computed by using Eq (12).

In the BX TCAS, the roll and pitch stabilized bearing angle, B, is computed from the predicted Δx and Δy by

$$
B = \tan^{-1} (\Delta y / \Delta x) - \psi \; . \tag{18}
$$

It is used for the beam pointing mechanization for the next interrogation period.

When Own ϕ and θ are zero, it can be shown that

$$
\begin{aligned}
\Delta x &= \Delta r \cos(\psi + \Delta b) \quad , \\
\Delta y &= \Delta r \sin(\psi + \Delta b) \quad .
\end{aligned}
\tag{19}
$$

In general, when ϕ and θ are non-zero, the above relationships do not hold. However, they may be used as an approximation if ϕ and θ are small.

Horizontal x-y Tracker Algorithm The horizontal position and velocity estimates are obtained by using standard fixed gains in a two-state filter called an alpha-beta tracker. The north referenced position computed in the last section provides the input to the algorithm.

Equations for the standard $\alpha\beta$ tracker algorithm are given below for the x axis. Equations for the y-axis are entirely analogous.

$$
\begin{aligned}
&\text{Prediction:} && \Delta x^{+} = \Delta\hat{x}_{old} + T \cdot \Delta\hat{\dot{x}}_{old} \\
&\text{Innovation:} && \Delta\tilde{x} = \Delta x^{m}_{new} - \Delta x^{+} \\
&\text{Position Update:} && \Delta\hat{x}_{new} = \Delta x^{+} + \alpha \cdot \Delta\tilde{x} \\
&\text{Velocity Update:} && \Delta\hat{\dot{x}}_{new} = \Delta\hat{\dot{x}}_{old} + (\beta/T) \cdot \Delta\tilde{x}
\end{aligned} \qquad (20)
$$

where T is the time elapsed since the last valid measurements. The α and β filter gains are optimized with respect to the measurement noise and target maneuver levels as well as the sampling period, T. The optimal values are listed in Table 4.

Table 4. BX TCAS Horizontal Tracker α and β Gain Values

T sec	1	2	3	4	5	6	7	8
α	0.25	0.37	0.465	0.53	0.58	0.62	0.645	0.665
β	.066	.175	.3	.431	.565	.685	.886	.91

The first two consecutive valid measurements can be used to initialize the estimates. This is effective when the sampling period is long. If the noise levels are too high, more measurements can be used to initialize by using a least squares fit.

When the measurement is invalid, the predicted value, given by Eq (20.a), is used as the measurement, i.e., the position estimate is coasted until the next valid measurement.

Own Altitude Tracker Algorithm   The CAS logic requires both Target and Own altitude and altitude rate estimates to assess the vertical threat situation. The Target altitude is provided by the Mode C (or Mode S) reply with the standard 100 ft quantization. Two methods are available for Own altitude estimation. One is to use the Own Mode C signal with the 100 ft quantization. The other is to use pressure altitude at a finer quantization. Referring to Fig. 4, the finer quantization signal, for example, can be obtained by tapping the transducer output channel just prior to the Mode C encoder. The finer quantization signal is more suitable for estimation purposes. In this case a simple alpha beta tracker is used. The algorithm is given by Eq (2) (substitute z_0 for x_0). The recommended α and β gains are 0.5 and 0.15. If only the 100 ft quantized altitude is available for Own, then a nonlinear tracker algorithm (discussed in the next section) needs to be used.

Target Altitude Tracker Algorithm   A simple low gain alpha beta tracker algorithm given by Eq (20) is used within the surveillance module. The recommended gains are 0.28 and 0.06 for α and β, respectively. These values are presumably tuned at the nominal TCAS surveillance cycle period of one (1) sec.

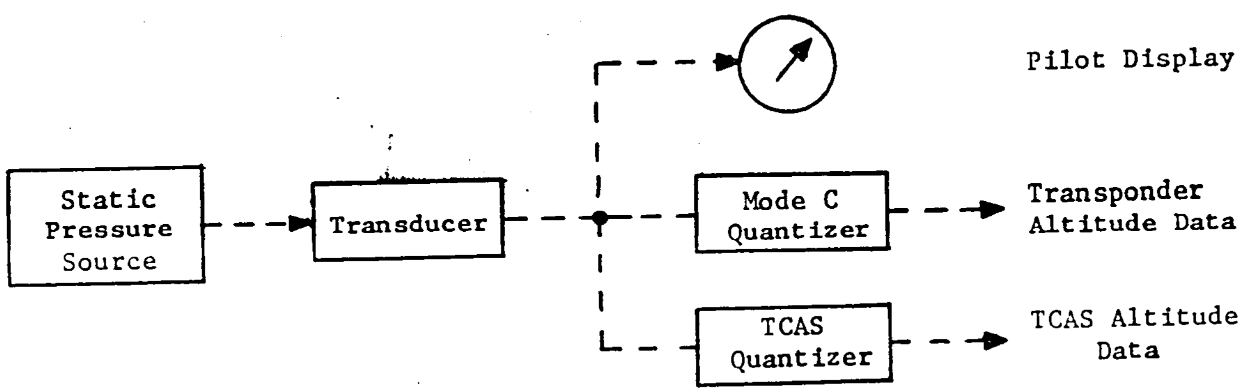


Figure 4. Schematic Diagram of Altitude Measurement Process.

A more complex nonlinear tracker is used by the CAS logic to track Mode C altitude reports. The algorithm is based on the so-called Level Occupancy Time (LOT) tracker designed for active BCAS application [9,10].

The basic idea of the LOT tracker is to estimate the altitude rate indirectly by estimating the time duration (called the level occupancy time) in a particular quantization level. If the altitude rate is constant, then so is the time duration. The estimate of level occupancy time, $\hat{T}$, is given by

$$\hat{T} = \hat{T} + k\,(T_{meas} - \hat{T}), \quad 0 < k < 1, \tag{21}$$

where

$$T_{meas} = t_{jump} - t_{last\ jump}. \tag{22}$$

Then, the altitude rate estimate is given by

$$\hat{\dot{z}} = 100/\hat{T}\ . \tag{23}$$

Some of the ramifications of the LOT tracker algorithm are:

(1) it requires at least two level changes to obtain rate information; and

(2) it requires at least three level changes to ascertain a change in rate (acceleration).

Therefore, for example, in the case of level-to-climb flight, it must wait approximately $3*(100/\hat{\dot{z}})$ sec before a reliable rate estimate is obtained. The above time duration is approximately 36 sec for a 500 ft/min vertical rate. The time delay of 36 sec is very significant when 30-45 sec is the CAS time constant.

The altitude sampling period is assumed to be one (1) sec. It is not known what the effect of the variable sampling period employed in the BX TCAS would be on the tracker performance.

Collision Avoidance Logic

Two types of CAS logic, vertical or horizontal, are planned for enhanced TCAS implementation. The vertical logic monitors the relative range and altitude states and performs the separation in altitude via altitude rate command, if necessary. The horizontal logic monitors the horizontal x and y and altitude states and performs the separation either horizontally or vertically. The added dimension presents two advantages: (a) more accurate situation assessment leading to a less frequent false alarm rate; and (b) a longer time period before the actual escape maneuver must take place. However, it requires use of a more accurate bearing sensor resulting in a higher system cost.

The vertical CAS logic has matured to the point that a version implemented in the DV TCAS is being flight tested in regular commercial airliner operation. The horizontal CAS logic is under development. A vertical only BX TCAS may soon be in a controlled flight test stage. For this reason, only the vertical logic is discussed here.

Referring to Fig. 1, the CAS logic consists of three major components - threat detection, threat resolution and maneuver coordination logic. The threat detection logic identifies the threat status of surrounding traffic. Furthermore, if a threat is dangerous, then the resolution logic determines a proper (vertical) escape maneuver. These are the basic ingredients. The maneuver coordination logic which is influenced by the implementation aspects consists of multi-threat logic and coordination logic with other TCAS systems.

The multi-threat logic is needed to resolve the maneuver commands caused by more than one threat. This consideration affects the command generation.

The coordination logic consists of two functions. One is to notify TCAS I equipped aircraft what Own intends to do and to communicate appropriate data. (Currently TCAS I is envisioned to be a totally passive system or a Mode S transponder with a pilot display/interface device.) The other function is to coordinate, resolve and communicate between Own and other TCAS II equipped aircraft. This is necessary to prevent both TCAS systems from generating maneuver commands independently which would aggreviate the threat situation further. Briefly, this is accomplished by one TCAS locking out the other.

Detection Logic

Threat status is classified into three threat levels called Advisories. These are Proximity, Traffic or Resolution Advisories. In the following sections, each classification logic as well as the maneuver command generation are discussed.

TCAS Sensitivity Level Selection The TCAS operating envelope is indexed by a parameter called the Sensitivity Level which ranges from 1 through 7. Level 1 is a standby condition in which TCAS does not interrogate and therefore does not perform any surveillance or resolution. This level is used only when TCAS has no collision avoidance responsibilities. In sensitivity level 2, TCAS continues surveillance, but is inhibited from declaring threats (and thus from issuing a resolution advisory). TCAS may generate traffic advisories in sensitivity level 2. Sensitivity levels 3 and 7 are reserved for future usage. Levels 4, 5 and 6 define a progressively "larger" protection volume.

The sensitivity level depends on three factors:

(a) pilot manual selection;
(b) uplink from the Mode S ground sensors; and
(c) radio and/or baro altitude.

The selection logic is as follows:

(1) Select the lowest level among the uplinked values, if any;
(2) Otherwise, select according to the altitude schedule - Table 5;
(3) Finally, use the pilot selected value, if it is lower.

Table 5. Sensitivity Selection Altitude Schedule

Sensitivity Level	Condition
2	Radio altitude less than 500 ft
4	Radio altitude between 500 and 2500 ft
5	Baro altitude less than 10,000 ft
6	Baro altitude greater than 10,000 ft

<u>Proximity Advisory Detection Logic</u> An intruder reporting Mode C altitude qualifies for a proximity advisory if range and altitude are small. These thresholds are 2 miles and 1200 feet, respectively. An intruder not reporting altitude, (i.e., Mode A only transponder) qualifies for a proximity advisory if its range is small and if Own is in the airspace in which altitude reporting is not required. This threshold is 15,500 ft.

<u>Traffic Advisory Detection Logic</u> The logic consists of two tests - range and altitude. These must be satisfied simultaneously in order to be declared a Traffic Advisory (TA) threat.

TA Range Test: Three cases are examined - range magnitude, range convergence, and range divergence cases. Table 6 lists the logic.

Table 6. TA Range Test

	Conditions for Passing
Magnitude	$\Delta\hat{r} \leq \theta_{\Delta r_{TA}}$
Convergence ($\Delta\hat{\dot{r}} \leq \Delta\dot{r}_{min}$)	$-(\Delta\hat{r} - \Delta r_{TA})/\hat{\dot{r}}' \leq \theta_{RTA}$, $\hat{\dot{r}}' = \min\{\hat{\dot{r}}, -\Delta\dot{r}_{min}\}$
Divergence ($\Delta\hat{\dot{r}} > \Delta\dot{r}_{min}$)	$\Delta\hat{r} \cdot \Delta\hat{\dot{r}} < \theta_{HTA}$ .AND. $\Delta\hat{r} < \Delta r_{TA}$

The parameter, $\Delta\dot{r}_{min}$ (nominally + 10fps), represents the extent to which the range rate can be estimated accurately. Other parameters $\theta_{\Delta rTA}$, Δr_{TA}, θ_{RTA}, and θ_{HTA} are mostly logic thresholds and depend on the TCAS sensitivity level. Specific values are given in Table 7.

TA Altitude Test: If the intruder is not capable of altitude reporting, then the test is passed automatically if Own aircraft is in the airspace in which altitude reporting is required. This threshold is again 15,500 ft or below.

For altitude reporting intruders, the test is slightly more involved. It is convenient to define the following relative altitude variables:

$$\begin{aligned} \Delta z &= \hat{z}_0 - \hat{z}_T \quad , \quad \Delta\dot{z} = \hat{\dot{z}}_0 - \hat{\dot{z}}_T \ ; \\ \Delta h &= |\Delta z| \quad , \quad \Delta\dot{h} = \text{sign}\ (\Delta z)\Delta\dot{z} \end{aligned} \tag{24}$$

The altitude test consists of two subtests - magnitude and convergence tests. Table 7 lists these tests.

The parameter, $\Delta\dot{h}_{min}$ (nominally - 1 fps), represents the minimin convergence threshold. $\theta_{\Delta h_{TA}}$ is the magnitude threshold, nominally 1200 ft. The parameter θ_{VTA} is the TA altitude closure time (tau) threshold, and its value depends on the sensitivity level. See Table 8.

It is noted that the threshold values corresponding to levels 3 and 7 do not have any significance at the current time. It is also noted that the main purpose of PA and TA is to help the pilot visually acquire the intruder.

Table 7. TA Altitude Test

	Conditions for Passing
Mode A Only	$z_o \leq 15{,}500$ ft
Magnitude	$\Delta h \leq \theta_{\Delta h_{TA}}$
Convergence	$\Delta\dot{h} < \Delta\dot{h}_{min}$ .AND. $-\Delta h/\Delta\dot{h} < \theta_{VTA}$

Table 8. TA Threshold Values

Parameter	Unit	Sensitivity Level					
		2	3	4	5	6	7
Δr_{TA}	nmi	.10	.13	.20	.40	1.20	1.60
θ_{HTA}	nmi^2/s	.00160	.002	.00278	.00278	.00278	.004
$\theta_{\Delta r_{TA}}$	nmi	.25	.35	.50	.75	1.50	2.00
θ_{RTA}	s	20	30	35	40	45	48
θ_{HTA}	s	20	30	35	40	45	48

Resolution Advisory Detection Logic This logic determines which intruders are predicted to be sufficiently close in range and altitude to require a resolution advisory. Similar to the Traffic Advisory logic both range and altitude are tested based on the estimated relative kinematic variables - $\hat{\Delta r}$, $\hat{\dot{\Delta r}}$, Δz and $\dot{\Delta z}$ (see Eq. (24) for definitions). If either test is not satisfied, an RA is not generated against the intruder. If both tests are satisfied, then the reliability of estimates are tested. This logic is necessary because, for example, the intruder altitude estimate contains a long dynamic delay.

Once an intruder is declared a RA threat, it remains in this status until either range or altitude test fails. It is also forced to remain a threat for at least five seconds to avoid overly brief displays of advisories.

No RA is issued for intruders not reporting altitude. In the interest of keeping the simulation program relatively simple, it is assumed that the simulated threat encounters are limited to single intruder cases. Actual TCAS logic contains a multi-threat encounter test by examining the internal threat intruder file.

Several of the thresholds used in the logic vary with the sensitivity level index. Higher sensitivity levels imply higher altitude flight (in non-terminal areas); this implies larger altimetry errors and sparser traffic. The detection thresholds are therefore made larger to help overcome the altimetry errors and to minimize the number of unwanted alerts in the sparser traffic environment. Lower sensitivity levels, on the other hand, imply lower altitude flight (in the vicinity of a terminal), which implies smaller altimetry errors and denser traffic. The detection thresholds are therefore made smaller in order to reduce unwanted alerts.

Range Test: The range test is divided into two cases - range convergence (negative range rate) and range divergence (positive range rate). Figure 5 depicts the criteria in the range - range rate plane. Each of these cases are discussed in more detail in the following paragraphs.

In the case of a converging intruder, the criterion is simply that the range closure time (called tau) to the Closest Point of Approach (CPA) be small. The standard tau, τ_r, is computed as range divided by range rate. A modified tau, τ_r , is computed in the same way except that the minimum range guard, Δr_{RA} is subtracted from the range. In equation form

$$\begin{aligned} \tau_r &= \hat{\Delta r}/\hat{\dot{\Delta r}} \ , \\ \tau_r' &= (\hat{\Delta r} - \Delta r_{RA})/\hat{\dot{\Delta r}} \ . \end{aligned} \tag{25}$$

Mainly, τ_r' is used for the converging intruder. The rationale is that when the range rate is large, then τ_r and τ_r' are similar in magnitude; but when range and range rate are both small, then τ_r may be large whereas τ_r' is small. The range test is given by

$$\tau_r \leq \theta_{RRA} \ . \tag{26}$$

The parameters, Δr_{RA} and θ_{RRA}, are dependent on the sensitivity level index.

For the non-converging case, the test is similar to the corresponding TA range test, i.e.,

$$\hat{\Delta r} \leq \Delta r_{RA} \ \text{.AND.} \ \hat{\Delta r} * \hat{\dot{\Delta r}} \leq \theta_{HRA} \ . \tag{27}$$

The RA range tests are summarized in Table 9.

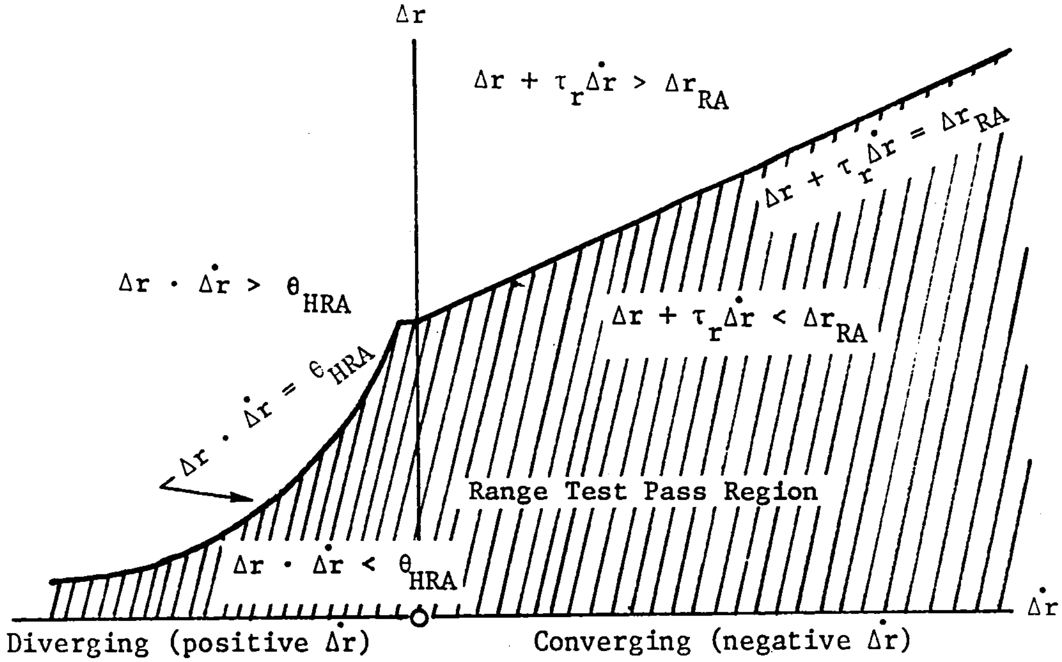


Figure 5. Region Defining RA Range Test

Table 9. RA Range Test

	Conditions for Passing
Convergent ($\Delta\dot{r} \le \Delta\dot{r}_{min}$)	$-(\hat{\Delta r} - \Delta r_{RA})/\hat{\Delta\dot{r}} \le \theta_{RRA}$, $\hat{\Delta\dot{r}} = \min\{\hat{\Delta\dot{r}}, -\Delta\dot{r}_{min}\}$
Divergent $\Delta\dot{r} > \Delta\dot{r}_{min}$	$\hat{\Delta r} < \Delta r_{RA}$ .AND. $\hat{\Delta r} \cdot \hat{\Delta\dot{r}} < \theta_{HRA}$

Altitude Test: The RA altitude test is more complex compared to the Traffic Advisory test. The shaded region of the $(\Delta z, \dot{\Delta z})$ plane in Fig. 6 represents geometries that pass the altitude test. Linear lines are related to vertical closure times and the horizontal lines $(\Delta z = \pm\, \theta_z)$ represent minimum vertical separation. There are two cases to consider. The altitude tests depend on the vertical miss-distance, Δz^+, at the CPA (minimum range). It is given by Table 10.

The following tests constitute the RA altitude logic.

(i) When the current vertical separation is small, then the vertical miss-distance, Δz^+, is tested. Thus, the criteria is given by

$$|\hat{\Delta z}| \leq \theta_z \text{ .AND. } |\Delta z^+| \leq \theta_z \ , \tag{28}$$

where the threshold parameter θ_z depends on Own altitude.

Table 10. Vertical Miss-Distance Computation

Vertical Miss-Distance	Condition	Comment
$\hat{\Delta z}$	$\dot{\Delta r} > 0$	Range diverging
Δz_1^+	$\Delta z_1^+ = \Delta z_2^+$	Range Converging
0	$\Delta z_1^+ \cdot \Delta z_2^+ \leq 0$	
$\min\{\Delta z_1^+, \Delta z_2^+ + 100\}$	$\Delta z_1^+ > 0$	
$\mathrm{Max}\{\Delta z_1^+, \Delta z_2^+ + 100\}$	$\Delta z_1^+ < 0$	

$$\tau_1 = \min\{-\hat{\Delta r}/\dot{\Delta r}', \tau_{vc}\}$$

$$\tau_2 = \min\{-(\hat{\Delta r} - \delta r)/\dot{\Delta r}', \tau_{vc}\}, \quad \dot{\Delta r}' = \min\{\hat{\dot{\Delta r}}, -\dot{\Delta r}_{min}\}$$

$$\Delta z_1^+ = \hat{\Delta z} + \tau_1 \hat{\dot{\Delta z}} \ ; \quad \hat{\Delta z} = \hat{z}_o - \hat{z}_I \ , \quad \hat{\dot{\Delta z}} = \hat{\dot{z}}_o - \hat{\dot{z}}_I$$

$$\Delta z_2^+ = \hat{\Delta z} + \tau_2 \hat{\dot{\Delta z}} \ ;$$

τ_{vc} = minimum allowable time interval depending on the sensitivity

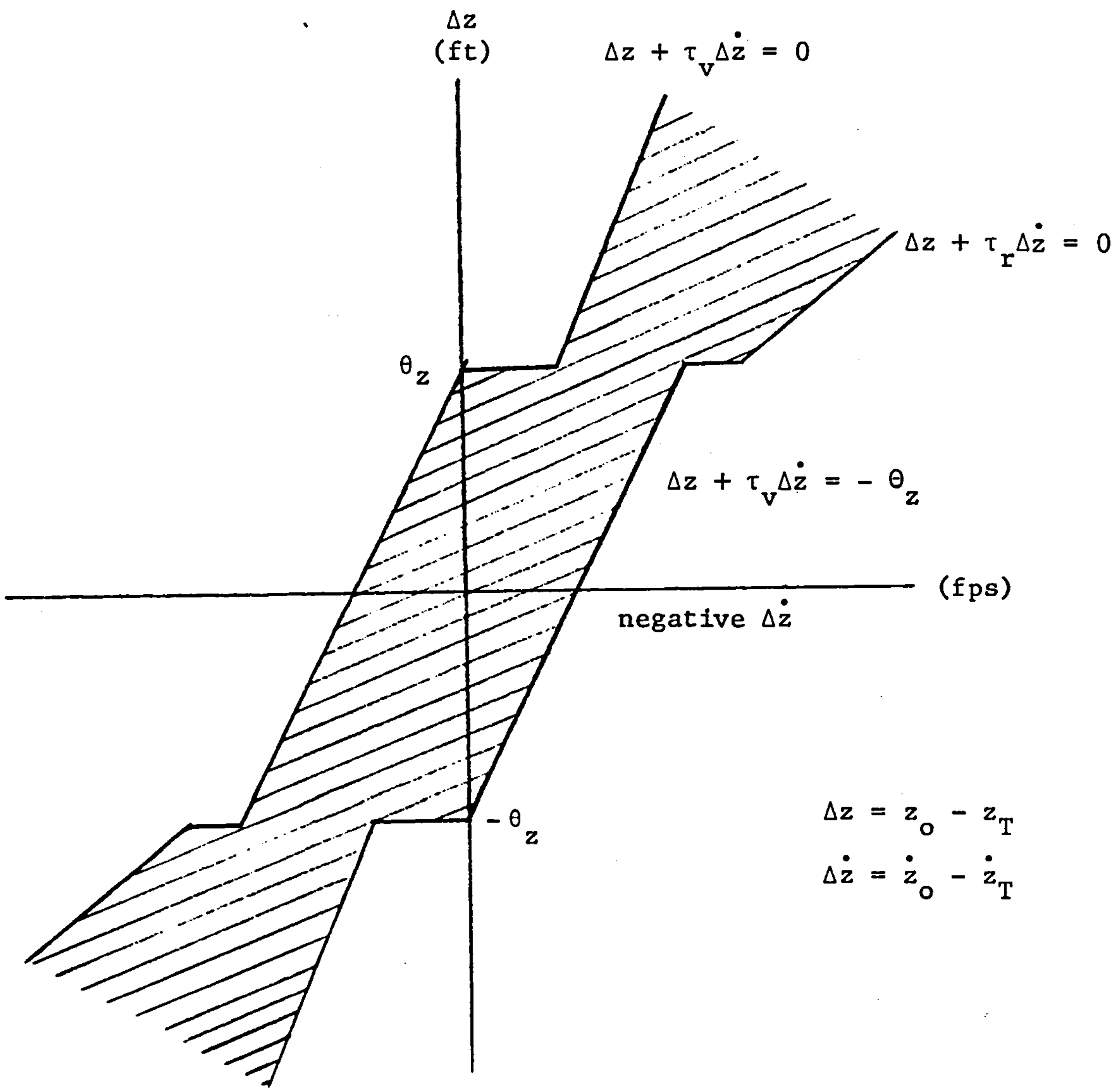


Figure 6. Region Defining RA Altitude Test

(ii) When the current vertical separation is large and the relative vertical rate is closing, then the vertical closure time and the co-altitude range are computed and used for assessment. Thus, the criteria is given by:

$$\tau_v = -\hat{\Delta z}/\hat{\Delta \dot{z}} < \theta_{VRA} \quad .\text{AND}. \qquad (29)$$
$$(\Delta r_{CA} \equiv \hat{\Delta r} + \tau_V \cdot \hat{\Delta \dot{r}} < \theta_{RCA} \quad .\text{OR}. \quad |\Delta z^+| < \theta_z)$$

The threshold parameters are sensitivity level dependent and are given by Table 11.

The RA altitude tests for an intruder which is already a threat are less stringent in order to retain an advisory until safe separation is assured. The altitude test is considered passed as long as the threat is coverging in range.

Table 12 summarizes the threat detection logic.

Table 11. RA Threshold Values

Parameter	Unit	Sensitivity Level				
		3	4	5	6	7
Δr_{TA}	nmi	.075	.1	.3	1.0	1.3
θ_{HRA}	nmi^2/s	.002	.00278	.00278	.00278	.004
θ_{RCA}	nmi	.3	.3	.4	.6	.9
θ_{RRA}	s	18	20	25	30	35
τ_{VC}	s	35	40	40	45	48
θ_{VRA}	s	25	30	30	35	40
θ_z	ft	750	$18000 > z_o$			
		850	$29000 > z_o > 18,000$			
		950	$z_o \geq 29,000$			
θ_a	ft	340	$z_o < 10,000$ ft			
		440	$18,000 > z_o \geq 10,000$			
		640	$30,000 > z_o \geq 18,000$			
		770	$z_o \geq 30,000$			

Table 12. Summary of Threat Detection Tests

Threat	Range Test	Altitude Test	Comments
Proximity	$\Delta r \leq \theta_{\Delta r_{TA}}$	$\hat{z}_0 \leq 15{,}500$ ft	. Mode A only transponder
		$\lvert\Delta z\rvert \leq 1{,}200$ ft	. Mode C or S transponder
Special TA	$\Delta r \leq 2 \cdot \theta_{\Delta r_{TA}}$	$\lvert\Delta z\rvert \leq 2 \cdot \theta_z$	. These tests are applicable to pop-ups because of unreliable velocity estimates.
Special RA	$\Delta r \leq 2 \cdot \theta_{\Delta r_{RA}}$		. Roughly 4 times the protection volumns.
Traffic Advisory	$\Delta r \leq \theta_{\Delta r_{TA}}$ .AND. $\Delta r \cdot \dot{\Delta r} \leq \theta_{HTA}$	$\hat{z}_0 \leq 15{,}500$ ft (Mode A) .OR. $\lvert\Delta z\rvert \leq \theta_{\Delta h_{TA}}$ .OR.	. Range Divergent Case
	$\min\{\tau_r, \tau_r'\}^{(1)} \leq \theta_{RTA}$	$\tau_V \leq \theta_{VTA}$	. Range Convergent Case
Resolution Advisory	$\Delta r \leq \Delta r_{RA}$ .AND. $\Delta r \cdot \dot{\Delta r} \leq \theta_H$	$\{\lvert\Delta z\rvert \leq \theta_z$ .AND. $\lvert\Delta z^+\rvert^{(2)} \leq \theta_z\}$.OR.	. Range Divergent Case
	$\min\{\tau_r, \tau_r'\} \leq \theta_{RRA}$	$\{\tau_V \leq \theta_{VRA}$ .AND. $\{\Delta r_{CA}^{(3)} \leq \theta_{RCA}$ .OR. $\lvert\Delta z^+\rvert \leq \theta_z\}$	. Range Convergent Case

(1) τ_r and τ_r' are the standard and modified range taus.

(2) Δz^+ is the projected vertical separation at the range critical times.

(3) Δr_{CA} is the relative range at co-altitude.

Resolution Advisory Selection

An intruder which is declared a threat by the detection logic of the previous section is processed further by the Resolution Advisory Selection logic which is discussed here. The modules which are simulated in the digital simulation program are described; however, two major modules are deleted from the simulation program - coordination and multi-aircraft advisory logic.

The coordination logic refers to the situation when both Own and the intruder are TCAS II equipped. In this situation, it is necessary that both TCAS systems work in a coordinated fashion, so that one does not generate a conflicting advisory with respect to the other. This is accomplished by means of one TCAS locking out the other; then, the locked out TCAS would behave similar to a TCAS I system. The coordination locking is performed through Mode S crosslink protocols.

The multi-aircraft advisory logic coordinate the advisories due to two or more simultaneous threats. For example, "Climb" (caused by threat A) and "Descend" (caused by B) cannot be given simultaneously. Thus one of these must be changed. For example, if Own is flying level, then one of these may be such that descend or climb produce similar results in the vertical separation. Thus, "Descend" may be changed to "Climb" (it depends on the "don't-care" flag).

In the following sections, the rest of the Resolution Advisory Selection logic elements are discussed. These include the sense selection, Own vertical modeling aspect, and advisory selection.

Sense Selection. The maneuver directional sense (climb or descend - for the resolution advisory) selection is performed only once for each threat. The sense is selected based on the projected vertical separation over the critical time period. The threat vertical profile is computed based on the constant vertical speed assumption. Own vertical is projected on the basis of 0.25g of acceleration (deceleration) to the nominal $\pm$ 1500 fpm of climb or descent. Thus, the climb or descend sense selection is performed by actually predicting the altitude separation and evaluating against a separation threshold.

The threat and Own predicted altitudes are computed by the following expressions (see Fig 7 for Own computation):

$$\text{Threat:} \qquad z_T^+ = \hat{z}_T + \tau \, \hat{\dot{z}}_T$$

$$\text{Own:} \qquad z_0^+ = \begin{cases} \hat{z}_0 + \hat{\dot{z}}_0 & T_D \geq \tau, \\ \hat{z}_0 + \tau \, \hat{\dot{z}}_0 + 0.5(\tau - T_D)^2 a, & T_A + T_D \geq \tau > T_D, \\ \hat{z}_0 + \tau \hat{\dot{z}}_0 + 0.5 \, T_D^2 a & \\ \quad + (\tau - T_A - T_D)\dot{z}_G \quad , & \tau > T_A + T_D. \end{cases} \tag{30}$$

Here,

T_D = Own pilot reaction delay time (nominally 5 sec),

$T_A = |\hat{\dot{z}}_0 - \dot{z}_G| \ddot{z}_{Max}$ = acceleration time,

$a = \ddot{z}_{Max}$ Sign $(\dot{z}_G - \hat{\dot{z}})$ = maximum acceleration (¼ g),

$\dot{z}_G$ = desired vertical speed (+ 1500 fpm for climb and - 1500 fpm for descend) .

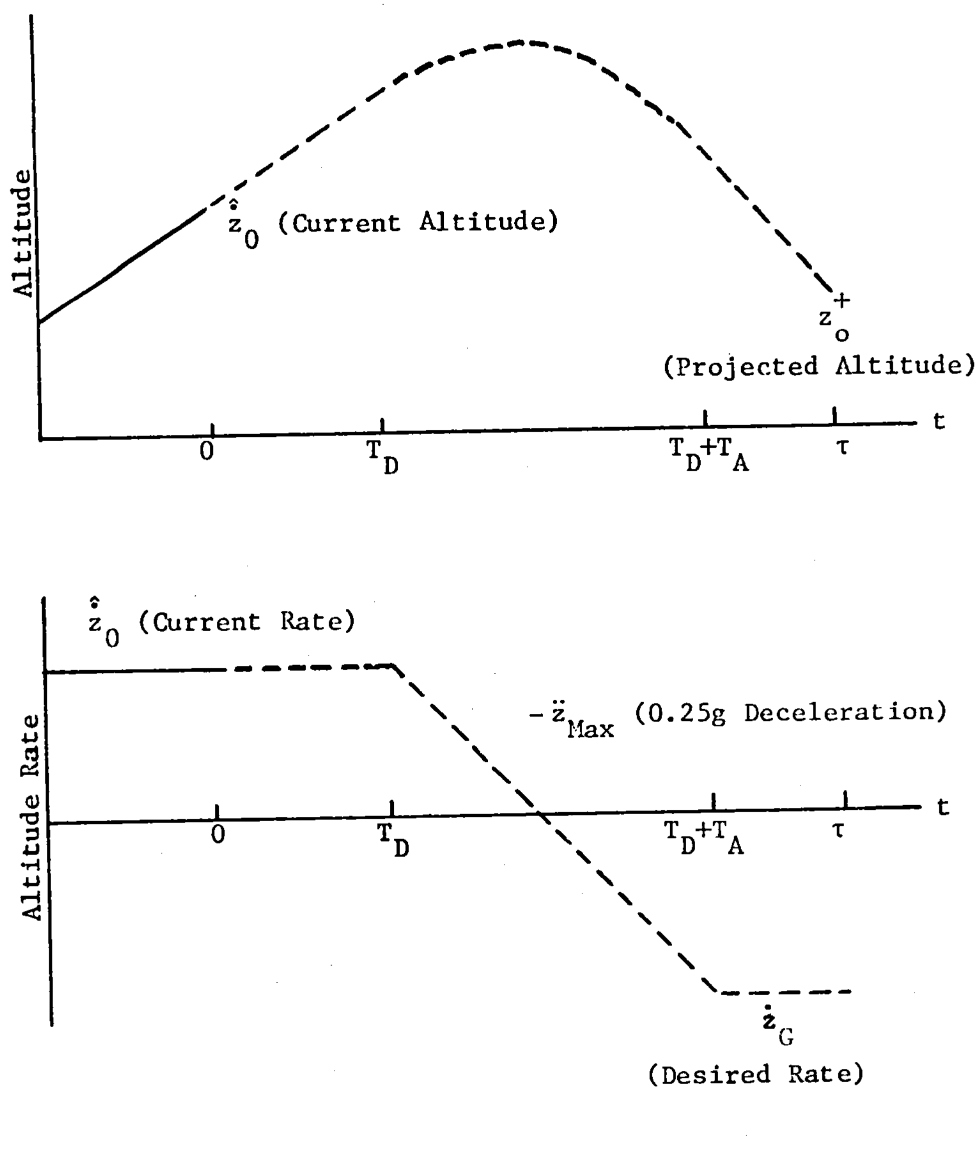


Figure 7. Own Altitude Projection Model

Using the above vertical model, the climb/descend sense is chosen according to Table 13. Basically, the sense is selected on the basis of the worst predicted vertical separation over the critical time interval defined by τ_r' (modified range tau) and τ_r (range tau).

Table 13. Climb or Descend Sense Selection

(1)	Compute target predicted altitudes at τ_r and τ_r': $z_T^+(\tau_r)$ and $z_T^+(\tau_r')$
(2)	Compute Own predicted altitudes at τ_r and τ_r', for $\dot{z}_G = 1500$ fpm; $z_{OC}^+(\tau_r)$ and $z_{OC}^+(\tau_r')$
(3)	Compute Own predicted altitudes at τ_r and τ_r', for $\dot{z}_G = -1500$ fpm, $z_{OD}^+(\tau_r)$ and $z_{OD}^+(\tau_r')$
(4)	Compute the separation for climb sense; $\Delta z_C^+ = \min\{[(z_{OC}^+(\tau_r) - z_T^+(\tau_r)], [z_{OC}^+(\tau_r') - z_T^+(\tau_r')]\}$
(5)	Compute the separation for descend sense; $\Delta z_D^+ = \min\{[z_T^+(\tau_r) - z_{OD}^+(\tau_r)], [z_T^+(\tau_r') - z_{OD}^+(\tau_r')]\}$
(6)	Select vertical sense by comparing the separations; climb (LSENSE = 1) if $\Delta z_C^+ > \Delta z_D^+$ descend (LSENSE = -1) if $\Delta z_D^+ > \Delta z_C^+$
	$\tau_r = \Delta\hat{r}/\Delta\dot{r}'$; $\tau_r' = -(\Delta r - \delta r)/\Delta\dot{r}'$, $\Delta\dot{r}' = \min\{\Delta\hat{\dot{r}}, -\Delta\dot{r}_{min},\}$. $\delta r = 0.247$ nmi, $\Delta\dot{r}_{min} = 0.00167$ nmi/sec (10 fps)

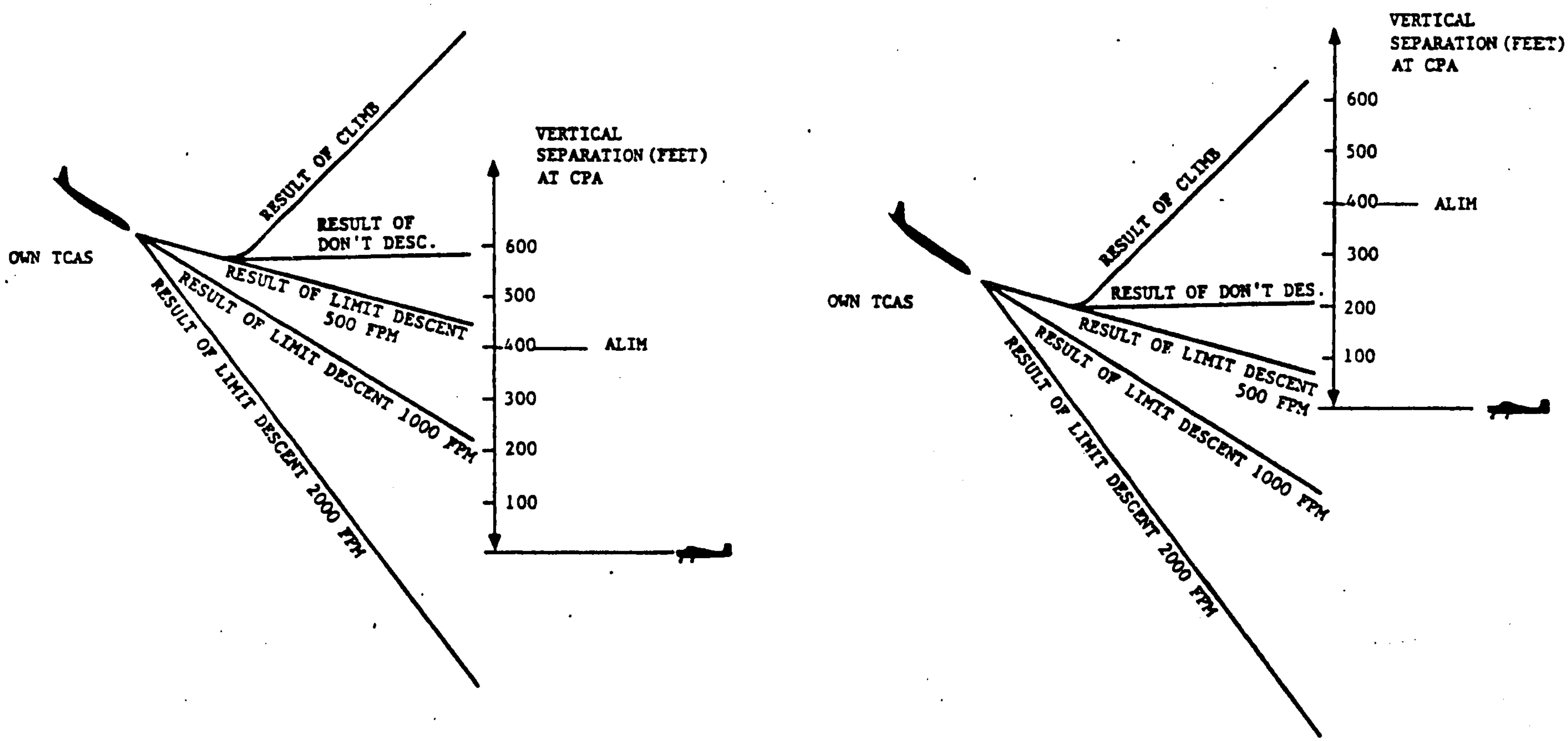
If both senses provide equally acceptable vertical separation over the critical time interval, then the "don't-care" flag is set. This flag is used if other threats exist. It is noted that this selected sense is chosen once and only once for the duration of this threat encounter.

Advisory Selection. A resolution advisory is selected against the threat using the sense selected by the previous section. In general, all the possible advisories are considered from weakest to strongest. The weakest advisory that satisfies the threshold, θ_a, (ALIM (ft)), against the threat at the closest approach is selected. The value of the RA threshold, ALIM, is given by Table 11. Table 14 shows weakest to strongest advisories.

Table 14. TCAS Resolution Advisories

	Climb (Descend) Sense
Vertical Speed Limit	Don't Descend (Climb) faster than 2000 fpm Don't Descend (Climb) faster than 1000 fpm Don't Descend (Climb) faster than 500 fpm
Negative	Don't Descend (Climb)
Positive	Climb (Descend)

For example, if the selected sense is positive (climb), then the projected separation is tested with the assumed vertical speed of - 2000 fpm. If the separation will not be achieved with this choice of vertical speed, then the next stronger (vertical speed of - 1000 fpm) speed is tried. This process is continued until the safe separation is achieved. Figure 8 shows this process for the vertical speed limit and positive advisory cases. It is noted that with this search sequence, either an advisory is found which is the weakest, or no advisory is found. If it is the latter case, then it is due essentially to the



(a) Don't descend faster than 500 fpm (vertical speed limit)

(b) Positive Climb

Figure 8. Resolution Advisory Selection Logic

detection logic delay or a sudden maneuver by the Target. (The intruder may have been a pop-up.) It is also noted that the above search procedure may have been time consuming.

There are two exceptions to the above logic- (a) when the relative vertical rate is primarily due to the intruder and the vertical miss-distance is less than the resolution threshold, and (b) when Own is nearly level and both the current and projected vertical separations are within the threshold. In both cases an immediate positive advisory is issued.

The descend sense is converted to negative if Own is proximate to the ground. If Own is near its climb limit (near the Own maximum altitude envelope), then the climb sense is converted to "Don't climb".

The above advisory selection procedure applies to intruders which are converging. If an intruder is not converging, then a negative (Don't climb or descend) advisory is issued. The reasoning is that an immediate positive advisory is not required.

Advisory Evaluation Logic. The advisory selection logic examines the projected separations of possible non-positive advisories systematically. If none are found which provide safe separation (ALIM), then the positive advisory corresponding to the selected sense is chosen without explicitly examining the vertical separation. Thus, the advisory evaluation logic is invoked to examine the projected separation corresponding to the positive advisory. If it is within the safety limit, then a flag is set. This situation can happen by a late maneuver by the intruder, by Own's failure to respond to an existing advisory, or by a late track acquisition.

The flag (indicating that a safe separation is not achievable) is used to warn the pilot. He must resort to other means (such as visual acquisition or the ground ATC) of achieving a sufficient separation. This indicates a "panic" situation.

In order to reduce the occurrence of such situations, the positive advisory is considered adequate if 100 ft of separation is achieved by the immediate escape maneuver of a nominal 1500 fpm vertical rate. If an altitude crossing is inevitable, then the positive maneuver is assumed acceptable if it occurs before τ_r' (modified range tau).

Resolution Advisory Packed Discretes. Selected resolution advisory is packed into a twelve bit word. This is to facilitate the communication of the advisory to external devices such as a CRT display or audi-alarm in the cockpit. Table 15 gives the bit pattern definitions. The lower 9 bits are given; the upper 3 bits are reserved for horizontal advisories and are zero for the vertical advisories. It is noted that bit patterns corresponding to RA Map No. 6 - 10 are identical with bit patterns corresponding to 1 - 5, except Bit 6. Bit 6 signifies the climb/descend sense.

Table 15. Vertical Resolution Advisory Bit Patterns

Packed Word	RA Map No.	Advisory
100000000	1	Climb
110000000	2	Don't Descend (DDES)
111000001	3	DDES/500 (1)
111000010	4	DDES/1000
111000011	5	DDES/2000
100100000 (2)	6	Descend
110100000	7	Don't Climb (DCL)
111100001	8	DCL/500 (3)
111100010	9	DCL/1000
111100011	10	DEL/2000

(1) DDES/500 = Don't Descend faster than 500 fpm.

(2) Bit 6 = sense sign bit (1 → descend; 0 → climb).

(3) DCL/500 = Don't Climb faster than 500 fpm.

III

BENDIX TCAS II SIMULATION PROGRAM DESCRIPTION

Introduction

Real time TCAS operation is very complex in terms of physical principals and computational functions. Most of the important aspects of the BX TCAS were covered in the previous chapter. A digital simulation model compatible with the CDTI applications is discussed in this chapter.

Simulation models can be built at many levels of sophistication and fidelity depending on their applications. The BX TCAS model was developed with two basic requirements: (1) it be applicable for use in an active pilot-in-the-loop simulator; and (2) it be operationally accurate. The first requirement implies that the simulation must be able to run in real time along with other simulation elements such as aircraft aerodynamics, engine, actuators and cockpit instrumentation. It also implies that the TCAS simulation model must be able to take traffic data, process them, and output the results in real time repetitively. Thus, it dictates a certain executive structure.

The second requirement implies that important kinematic and dynamic characteristics are preserved. This means that the model must contain enough detail of the actual system without being overly complex. At the onset, it was decided to not be concerned with the details of radio transmissions. Thus, the TCAS simulation is based on the aircraft relative kinematics, since this aspect is the most important in the CDTI applications.

The model was developed as an analysis tool in an off-line mode. However, the semi-real time executive, highly modular construction and the fact that it is written in Fortran (very portable) made it a very simple task to convert it into a real-time module.

The simulation program is designed according to the over-all signal flow presented in Figure 9. The Traffic Generator is an external module which provides

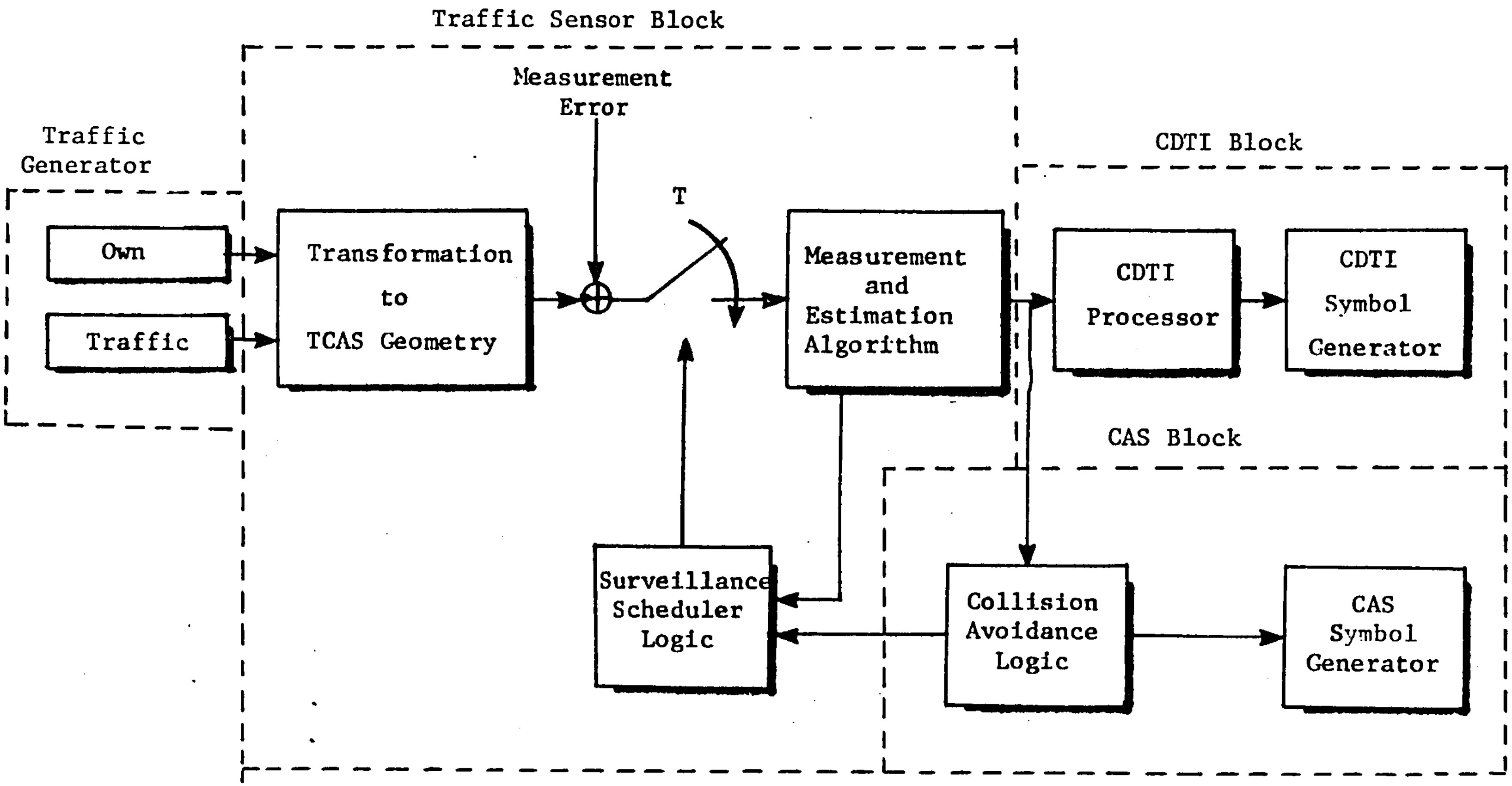


Figure 9. Enhanced BX TCAS II/CDTI Sensor Model Block Diagram

kinematic information of Own and other traffic in Own's vicinity. The Traffic Sensor block processes the input through geometry, measurement errors, estimation and surveillance scheduling modules to create (or update) the internal Own and traffic files. The CDTI block processes the Own and traffic files to generate display information. The CAS block makes threat assessments based on the same information to generate various advisories. In the following sections, macro aspects of the Traffic Sensor and CAS blocks are discussed in detail at module levels. The traffic generator, CDTI and CAS symbol generator modules are assumed to be external to the current simulation program.

Executive Logic

Figure 10 shows the BX TCAS executive logic flow. During each cycle time (1 sec interval), Own and Traffic information is passed to the simulation by means of an input common. Table 16 lists the necessary inputs. First, if the TCAS is operational (not on ground), then file entries and parameters are initialized. This is the power-up mode. Afterwords, Own-dependent parameters and variables are computed and updated. Within the proximate traffic Do-loop, traffic data associated with each aircraft are processed to initiate, update or delete the internal traffic file entry. The basic sequential steps are:

(i) Check the surveillance schedule for this time period;

(ii) If not scheduled, skip this cycle. Otherwise, compute TCAS measurements;

(iii) If the report is invalid (probabilistic model), skip this cycle. Otherwise, add measurement errors;

(iv) Perform inverse transformation;

(v) Check the acquisition or track status. If acquisition, update the track file entry and skip the rest; and

(vi) If in the track region, update or initialize the position and velocity estimates and update the traffic file entry.

The above steps are performed until each traffic element is exhausted. By this time, the traffic file is initialized, updated or deleted. This part constitutes the sensor logic.

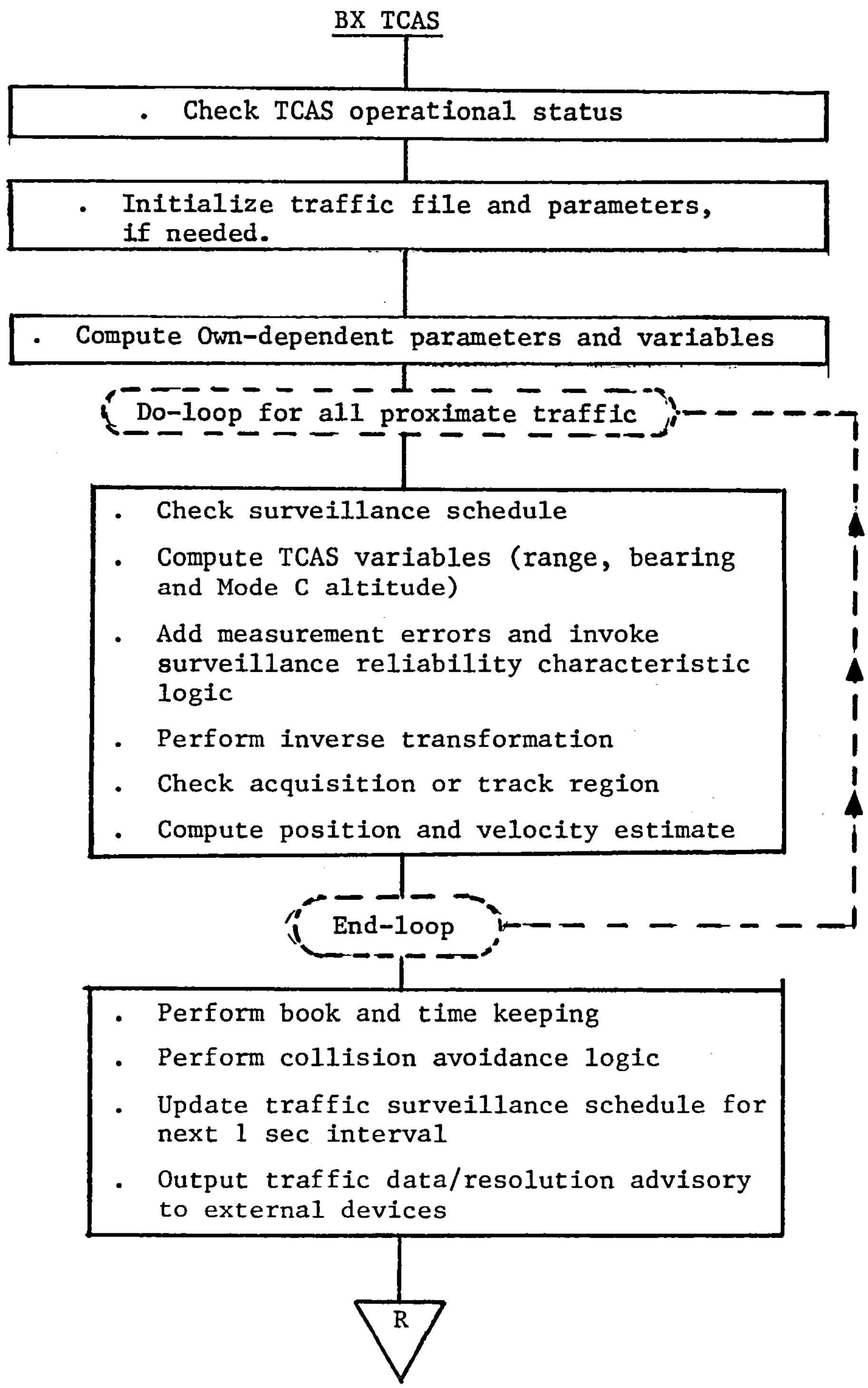


Figure 10. BX TCAS Macro Flow Chart

Table 16. List of External Input

	Own Aircraft
t	Simulation time (sec)
x, y, z	North-east (nmi) and altitude above MSL (ft)
v_G	Ground speed (kt)
ϕ,θ,ψ	Altitude angles (rad)
ITRGT	CDTZ target identification number
IWOW	Weight-on-wheel discrete (=1= on ground)
MANSEN	TCAS sensitivity level (0 = off; 1-7 = manual; 8 = automatic)
	Traffic Data (up to 40)
IDAC	Uniquely assigned traffic identification number
IXNSP	Transponder type indicator (1 = Mode A only; 2 = ATCRBS; 3 = Mode S
x, y, z	North-east (nmi) and altitude above MSL (ft)
NAC	Number of Traffic in the Data Set

After the sensor part, certain housekeeping functions are performed. This includes timer count-down and file entry elimination of traffic which is no longer relevant. Then the collision avoidance logic is invoked. This module processes the traffic file corresponding to aircraft within the track region. The CAS logic results are stored in its own CAS file.

The last major function is to schedule which aircraft (within the traffic file) needs to be interrogated in the next cycle. This function depends on the relative kinematics, acquisition or track status, or the threat status. Lastly, various output variables are extracted from the traffic file and stored in an output common.

The above cycle is repeated again with a new set of external traffic data. The new set may contain new traffic. Traffic which was included in the previous cycle may not be present this cycle. The program is flexible to handle this changing traffic pattern.

Figure 11 shows a more detailed flow chart. It has a very compact top-down structure, and all the computations are performed by dedicated modules which include housekeeping functions. Even though it may be inefficient compared to the in-line coding, this structure enhances the readability and maintainability of the software.

Figure 12 shows the subroutine structure consisting of forty relatively small modules. These are grouped about the functional characteristics rather than the order of the calling sequence. Table 17 lists the major subroutines with short functional descriptions. There is direct correspondence between these descriptions and the TCAS operational descriptions given in Chapter 2.

Major program commons are listed in Appendix A. Tables A.1 through A.7 are the dictionary references between the Fortran names and engineering variables and parameters. (Unfortunately, the Fortran compiler does not have cross-reference capability across the subroutines.) Transfer of variables among the subroutines is accomplished through these commons except for general usage subroutines and functions. The common variables are grouped according to functional definitions. Table 18 contains many common explanations which are useful.

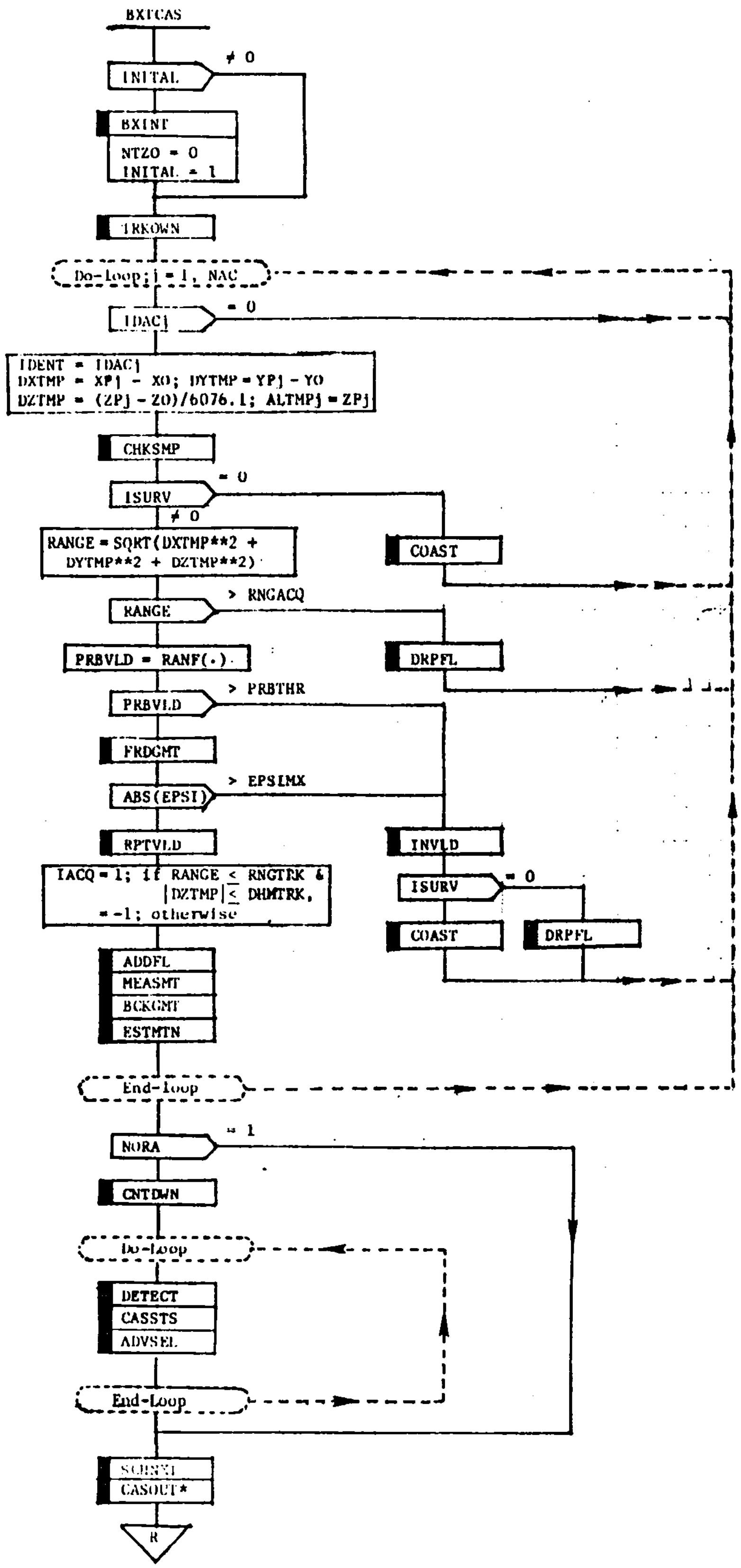


Figure 11. BX TCAS Executive Logic

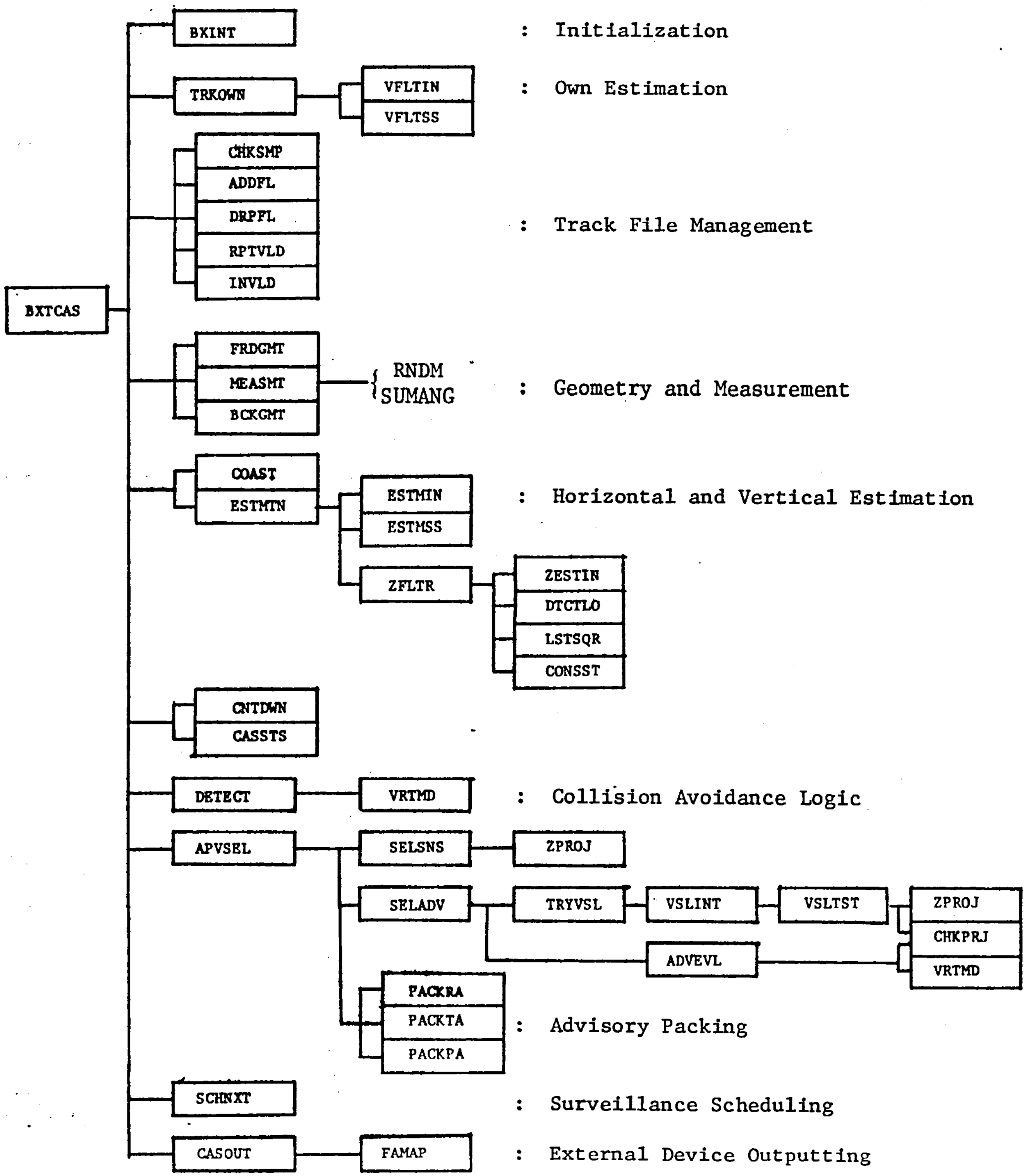


Figure 12. BX TCAS Subroutine Structure

Table 17. Short Functional Description of Major Subroutines

(1)	BXINT	: Set system parameters and initialize variables. Zero out track file. Equivalent to "power-up" initialization.
(2)	TRKOWN	: Perform Own vertical estimation based on the fine quantized altitude. Computes the transformation matrix, TBL, from ϕ, θ and $\dot{\phi}$. Computes dynamic parameters which depend only on Own variables.
(3)	CHKSMP	: Looks up the file index corresponding to the AC ID No. If the aircraft is scheduled for surveillance, then ISURV flag is set. This routine is companion to SCHNXT (4).
(4)	ADDFL	: Adds aircraft to the track file if not yet filed.
	DRPFL	: Deletes aircraft from the track file and the pertinent file elements are reset to zero.
	RPTVLD	: Resets invalid report counter, IRPT.
	INVLD	: Increments the invalid report counter, IRPT; it is compared with IRPTMX (nominally 4) and set deletion flag; if not, the aircraft is scheduled for the next surveillance period, i.e., one second later. These four routines are sensor track file management routines.
(5)	FRDGMT	: Computes the forward geometry; i.e., the relative bearing b, the track region range limit $r_{Max}(b)$, and the look-up angle ε.
(6)	MEASMT	: Computes measurement errors and the TCAS measured variables.
(7)	BCKGMT	: Computes the backward geometry; i.e., local level north and east position.
(8)	COAST	: Prediction part of the estimation algorithm.
	ESTMTN	: Performs the estimation function including the filter initialization. It also computes next scheduled surveillance time (rounded up to second). This routine calls two major subroutines ESTMSS (horizontal x-y tracker) and EFLTR (vertical tracker based on Level Switching Time).
(9)	CNTDWN	: This routine counts down CAS timers - 5 sec advisory counter and a 10 sec timer. Inactive threat is dropped in 5 sec and inactive traffic in 10 sec. When timers run out, various flags are reset.

Table 17 - Continued

(10)	CASSTS :	This routine checks the Advisory status of the candidate threat. It will find an empty CAS file slot and assigns the slot to the candidate if appropriate. It also cleans up CAS file if it is no longer a threat.
(11)	DETECT :	Examines all the actively tracked aircraft against the proximity, traffic or resolution advisory conditions. Then creates a tentative collision avoidance file.
(12)	ADVSEL :	This routine generates Proximate, Traffic and Resolution Advisories. Two major subroutines are called. These are SELSNS (select climb/descend sense) and SELADV (generates escape maneuver). It also calls packing routine to pack output variables.
(13)	SCHNXT :	Processes the next scheduled surveillance time for each aircraft in the traffic file. The flag, NXTS, is set if the aircraft is scheduled for the next one second interval.
(14)	CASOUT :	Stores output variables and also packs the BA discrete word.

Table 18. Explanation of Program Common Contents

1. /OWNTFL/ includes Own track variables including the Own inputs from external Own generation modules. The variables include position, altitude angles, body to local-level transformation matrix, altitudes and vertical estimate and initialization flag.

   /ITRGT/ includes CDTI target designation identifier (provided by the pilot interface, weight-on-wheel status and TCAS operational status indicator).

2. /BNDXFL/ includes targets identification, various time data, true and measured range and bearing, true, measured and estimated x-y position and velocity. This and /ZTRKFL/ forms the sensor traffic file. The variables are allocated 20 elements so that the simulation program can track up to 20 targets in Own's vicinity.

3. /ZTRKFL/ includes initialization flag, stored mode C altituducts and times, stored level switching variables and internal and external altitude and altitude rate estimates. These are used to implement the Level Switching Time vertical tracker algorithm.

   /ZWRKUR/ includes level switching time flag (temporary) and other variables which are used as temporary work memories.

4. /WOKKVR/ includes indicators and flags for the executive logic, temporary position variables and sensor constants, parameters and dynamic thresholds.

5. /CASFIL/ is the collision avoidance logic threat file. Up to 5 threats are accomodated each cycle (no multi-threat logic, however). The common includes number of declared threats, Traffic file index (identification), RA, TA and PA counter and status, 10 sec timer, and threat related variables such as time to CPA, relative x-y-z position at CPA, reference vertical speed for escape maneuver, RA packed-word and RA text. This common needs to be accessed for CAS output construction depending on the external device requirements.

6. /CWRKVR/ includes number of candidate threats, temporary /CASTIL/ index, RA capability flag, Own and Target (Mode S only) sensitivity index, PA, TA and RA temporary flags and temporary CAS timer, vertical current and projected separations.

   /CWRKV2/ is supplemental to /CWRKVR/ and includes temporary RA packed-word, various flags and projected vertical separations.

7. /CASPAR/ contains all the necessary collision avoidance threshold values. Some of these are constant. These are dynamically determined each cycle time.

In the following sections Sensor and Collision Avoidance Logic modules are discussed in detail.

It is advisable for readers to obtain the program listing - ETCAS II.VLD - from Dr. R. L. Bowles or Mr. D. Williams of NASA/Langley. The program is very easy to read and contains detailed engineering comments.

The TCAS sensor module contains twenty subroutines and includes the following major components:

(1) Computation related to Own variables;

(2) Traffic file management routines;

(3) TCAS geometry and measurements;

(4) Traffic aircraft position and velocity estimation; and

(5) Surveillance (interrogation) scheduling.

These are discussed in detail using tables, flow charts, and equations in the following sections.

Computation Related to Own Variables The major routine is TRKOWN. In addition to the Own related computation, it is used to initialize or reset variables for the 1 sec computation cycle. This is the first subroutine to be called by the BXTCAS executive. Table 19 summarizes the computation in proper sequence. The following remarks apply to the item numbers in Table 19.

Item (3) - Initialization is performed by the VFLTIN subroutine as follows:

$$\begin{aligned} \dot{z}_0 &= z_{0m} \\ \hat{\dot{z}}_0 &= 0 \end{aligned} \tag{31}$$

The estimation update is performed by the VFLTSS subroutine using the standard alpha-beta tracker algorithm as follows:

$$\begin{aligned} z_0^+ &= \hat{z}_0 + \hat{\dot{z}}_0, \\ \tilde{z}_0 &= z_0^m - z_0^+, \\ \hat{z}_0 &= z_0^+ + \alpha\tilde{z}_0, \\ \hat{\dot{z}}_0 &= \hat{\dot{z}}_0 + \beta\tilde{z}_0, \end{aligned} \tag{32}$$

Table 19. Own Related Computation (TRKOWN)

(1) Transfer Own input (t, x, y, z, V_G, ϕ, θ, ψ) to a separate TCAS common called /OWNTFL/. Compute radio altitude.

(2) Using ϕ, θ, and ψ compute T_{BL}. See Eq (19).

(3) Initialize (Subroutine VFLTIN) or update (Subroutine VLFTSS) Own altitude and altitude rate estimates.

(4) Compute the acquisition or tract region related variables (DHMTRK, RLIM, STMAX). See Eqs (3,5,6)

(5) Determine Own sensitivity level index (LVLOWN) depending on radio altitude (RALT), baro altitude estimate (ZOM(1)), and pilot manual select (MANSEN). See Table 4.

(6) Estimate altitude above ground using radio and baro altitudes.

(7) Compute collision avoidance thresholds, ZTHR and ALMT. See Table 9.

(8) Reset variables NORA (No Resolution indicator), NCAND(10), INTENT, INTHR, LVSLOK, LDCFLG, and LADNOK.

where α and β are equal to 0.4 and 0.15, respectively.

Item (6). Because most radio-altimeters are valid for readings below 2500 ft, the height above ground level is estimated only where radio altitude is less than 2500 ft. The estimate is given by the baro-altimeter based altitude estimate minus the radio-altitude.

Traffic File Management Routines There are four routines in this category - ADDFL, DRPFL, RPTVLD, and INVLD (CHKSMP is discussed later). Each is discussed below.

The ADDFL routine goes through the traffic file list, IDF (see common/BNDXFL/). If the input aircraft identification corresponds to one of the IDF's, then the corresponding index is returned. If the input identification is not included, then an empty file is searched, and the index is returned. If an empty file is not found, then INDEX is set to -1 to indicate the status. When the index is found, then the acquisition or track status and the transponder type are stored.

The DRPFL routine is essentially the reverse of the above process. If the input aircraft is no longer within the serveillance volume, then the aircraft identification is deleted from the list, IDF, and the corresponding file elements are reset.

The RPTVLD routine is called when a valid surveillance report is received. In this case, the IRPT counter corresponding to this aircraft's file index is set to 0.

The INVLD routine is called when an expected surveillance report is not received. The routine performs the following computations:

(1) Increment the invalid report counter, IRPT, by 1;

(2) If IRPT is greater than 3, then drop further surveillance effort for this aircraft; or

(3) If IRPT is less than or equal to 3, then schedule the surveillance interrogation for the next operating cycle by setting TNXT(INDEX) to 1 sec.

The corresponding program listings should by consulted. Because of the file management nature of these routines, the logic is complex but otherwise straight forward.

TCAS Geometry and Measurement There are three subroutines in this category - FRDGMT, MEASMT, and BCKGMT. These are discussed below.

The FRDGMT routine performs forward geometry computation. The north-east-down (NED) target relative position is transformed to body referenced position using the T_{LB} transformation. See Eq (10). Then the bearing and look-up angles are computed using Eq (11). The look-up angle is used to check for antenna vertical coverage. Additionally, the bearing dependant maximum track range (RNGTRK) is computed using variables computed in TRKOWN and the bearing angle. Equation (4) is used for this purpose. It is noted that the range is computed in the executive (BXTCAS), because it is independent of the orthogonal transformation.

The MEASMT routine computes range and bearing errors (assumed to be zero mean white noise with 75 ft and 1 deg standard deviations, respectively). These errors are added to true range and bearing to form the TCAS measurements. It is assumed that the input altitude contains the baro-altimeter error characteristics.* Therefore, this routine simply quantizes the input altitude to generate the Mode C altitude report according to the following formula:

$$Ih = \mathrm{Int}[z_T + 0.5\ q)/q]; \quad q = 100 \text{ ft.} \tag{33}$$

For a Mode A target, Ih is assigned a large number (1270) (an altitude of 127,000 ft).

The BCKGMT routine performs the inverse transformation, i.e., the NED relative position is obtained from measured range, bearing and altitude. This is the reverse of the FRDGMT computations.

*It is noted that that the current program contains reference to the altitude scale factor error (SFT) and the high frequency noise (SIGZ); therefore, SFMX and SIGZ must be assigned 0 to affect the above requirements.

When the target transponder has only Mode A capability, the altitude measurement is not available. Thus, in this case, the exact inverse transformation cannot be obtained. (The Mode A target can only generate TA and PA.) The problem is side-stepped by assuming that (1) the target is at co-altitude and (2) Own pitch and roll angles are zero. With these assumptions, the Mode A target's horizontal position is given by

$$\begin{aligned} \Delta x_m &= \Delta r \ \cos (\psi + \Delta b), \\ \Delta y_m &= \Delta r \ \sin (\psi + \Delta b), \end{aligned} \tag{34}$$

Where ψ is the Own heading angle. This procedure is recommended by the draft TCAS MOPS [2].

When the target transponder is Mode C (ATCRBS) or Mode S, then the more exact transformation is obtained by following Eqs (12) through (17). A small modification is added to prevent a singularity problem caused by measurement errors. When the relative range and altitude are similar, then the additive noise may make range smaller than the relative altitude. This causes the singularity problem in the above inverse transformation. This is avoided by redefining the range as

$$\Delta r_m \equiv \Delta z + 300 \text{ ft} . \tag{35}$$

It is noted that this routine uses the same transformation matrix used in the FRDGMT routine. In reality, the computation performed by the FRDGMT and MEASMT routines are not part of TCAS system; this part is done via various RF activities and associated processes. Therefore, in order to increase the simulation fidelity, the transformation matrix used in BCKGMT must be computed from the measured ϕ, θ, and ψ rather than the true ϕ, θ and ψ. In many simulation environments, the measured values as well as the "measured" transformation are available in the navigation module; therefore, these may also be input and used rather than the true values.

At this stage, the measured NED positions are obtained and the corresponding track file elements are stored. These are used by the next task to generate estimates.

Position and Velocity Estimation The estimation function based on the TCAS sensor measurements is performed by two subroutines - ESTMTN and COAST. The ESTMTN subroutine is the estimation executive and takes care of all estimation function except the prediction part which is performed by COAST.

The COAST routine performs the 1 sec prediction for targets which are within the track region according to the following formula:

$$\hat{\Delta x} = \hat{\Delta x} + \hat{\Delta \dot{x}}; \quad \hat{\Delta y} = \hat{\Delta y} + \hat{\Delta \dot{y}} \tag{36}$$

If the target is Mode C or Mode S, then the external altitude is also coasted by a similar equation:

$$\hat{z}_{EX} = \hat{z}_{EX} + \hat{\dot{z}}_{EX} \; . \tag{37}$$

The COAST routine is called by the BXTCAS executive if:

(1) The individual target was not scheduled; or

(2) There is a failure in surveillance caused by either the reliability test or antenna shadowing.

For the BXTCAS, the former is usually the reason, because targets are usually scheduled with longer than a 1 sec surveillance interval.

The ESTMTN executive performs the following three tasks:

(a) Process altitude input;

(b) Process horizontal input; and

(c) Compute next surveillance time based on the latest estimates.

These are performed according to the computer flow chart depicted in Fig. 13. First, the target transponder flag is checked. If the target is Mode A, then the estimate is set to the input value (which is set to 127,000 ft in MEASMT); otherwise VFLTR is called to process further. The VFLTR routine is explained later.

Next, the target's track status is checked. There are four possibilities based on the previous and current status:

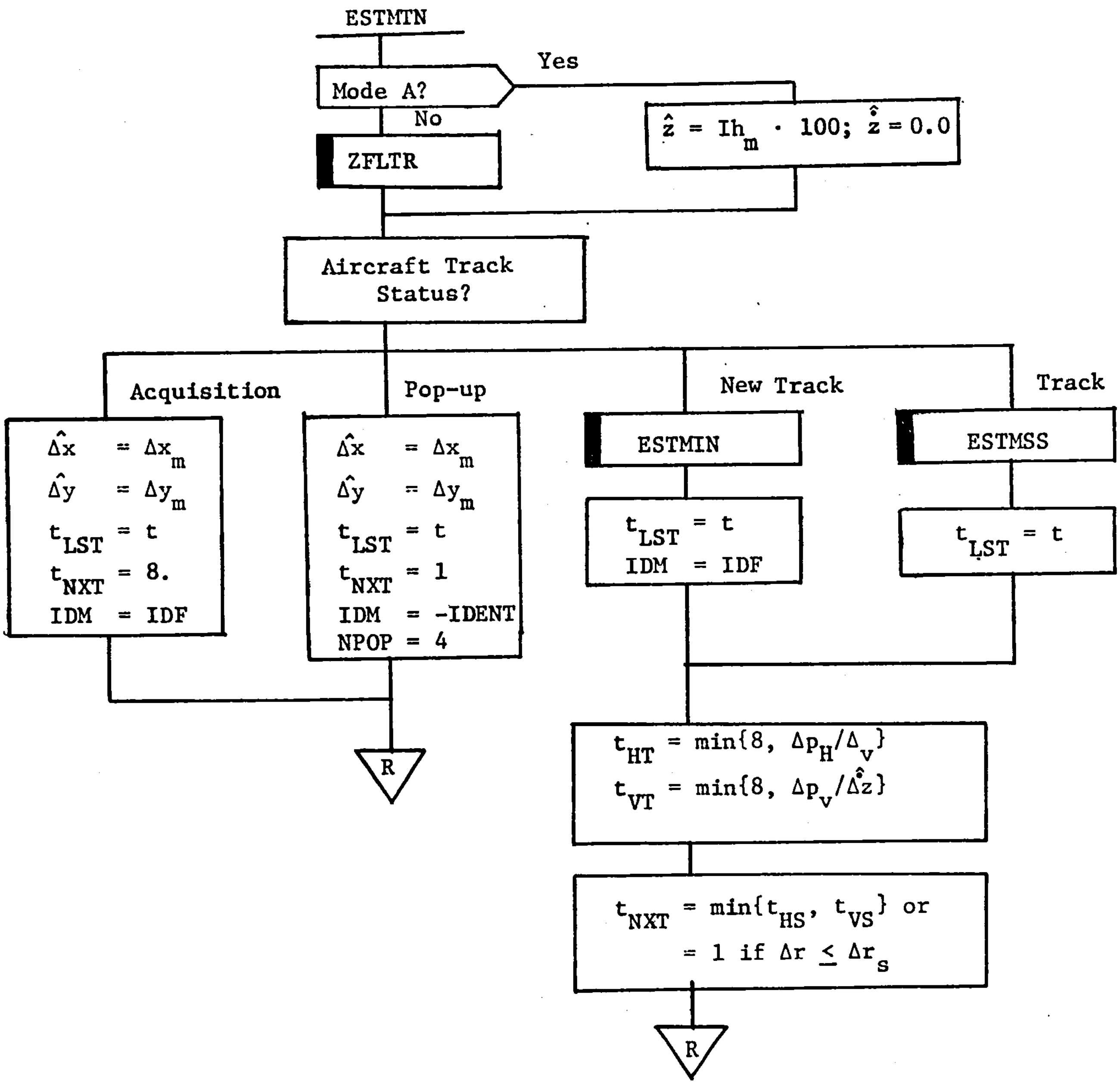


Figure 13. Estimation Executive (ESTMTN) Flow Chart

(1) in the acquisition region;

(2) sudden track region (pop-up);

(3) transitioned from acquisition to track status; and

(4) in the track region.

In Case (1) the estimates are set to the measurements (this is necessary for initialization purposes). The last valid report time, TLST, is set to the current time; the next surveillance time, TNXT, is set to 8 sec; and the track status (IDM) is noted.

Case (2) represents a peculiar dynamic situation. Because the track region is contained in the acquisition region, the targets in the traffic file are expected to "transition" from the acquisition to track (and back to acquisition and then disappear). Because encounters for simulation scenarios may begin with given traffic patterns, and/or because targets may have been in the antenna shadow, targets may appear in the track region without being identified in the acquisition region. Therefore, this situation requires a special logic. The target is assigned to be in the acquisition traffic. Thus, the same actions are taken as in Case (1) except TNXT is set to 1 (this prompts the surveillance during the upcoming 1 sec cycle time). Additionally, the pop-up counter, NPOP, is set to 4 so that for the next four cycle periods the target is interrogated.

In Case (3), the target has transitioned from the acquisition to track regions in a conventional way. The estimates are initialized in the ESTMIN routine using the last two measurements as follows (sequential order of computation must be kept):

$$\Delta\hat{\dot{x}} = (\Delta x_m - \Delta\hat{x})/(t - t_{LST}); \text{ and} \tag{38}$$

$$\Delta\hat{x} = \Delta x_m,$$

Here, $\Delta\hat{x}$ in the velocity equation contains the last valid report time (TLST) measurement. The Δy equations are analogous. The valid report time and the track status are updated.

In Case (4), the target has been in track region; thus, the "steady state" algorithm (Eq (20) and Table 3) is used in ESTMSS. There is one exception. The prediction is performed for a 1 sec interval only, since the COAST routine has been performing the prediction for other sampling periods.

Targets which are applicable to Cases (3) and (4) are further processed to compute the next surveillance time (t_{NXT}) based on the updated estimates. For this purpose, targets are assumed to be Mode S equipped and a modified Eq (8) is used, i.e.,

$$\Delta t_T \ (t_{NXT}) = \text{Max}\ \{1 \text{ sec}, \min \{t_{HT}, t_{VT}, 8\}\}$$

Here,

t_{HT} = the number of seconds it will take the target to move 1,000 ft horizontally; and

t_{VT} = the number of seconds it will take the target to move 200 ft vertically.

Furthermore, if the target range is less than Δr_s (1.65 nmi), then t_{NXT} is set to 1 sec. Also, if the target is identified as a preliminary threat, then t_{NXT} is set to 1 sec by the CAS logic.

The vertical estimation is performed by the ZFLTR routine which implements the Level Switching Time (LST) tracker algorithm. Figure 14 depicts a macro flow chart. Inputs are the traffic file index (INDEX), system time and Mode C altitude. Outputs are the external altitude and altitude rate estimates ($\hat{z}_{EX}$, $\hat{\dot{z}}_{EX}$) contained in the /ZTRKFL/ common. The logic is complex due to the 100 ft quantization of the Mode C altitude report. The situation is depicted by Fig. 15.

First, the altitude estimates (and host of other internal variables) are initialized, if not yet done. The DTCTLO routine is next invoked to see whether a new altitude level was attained. If new level switching was detected,

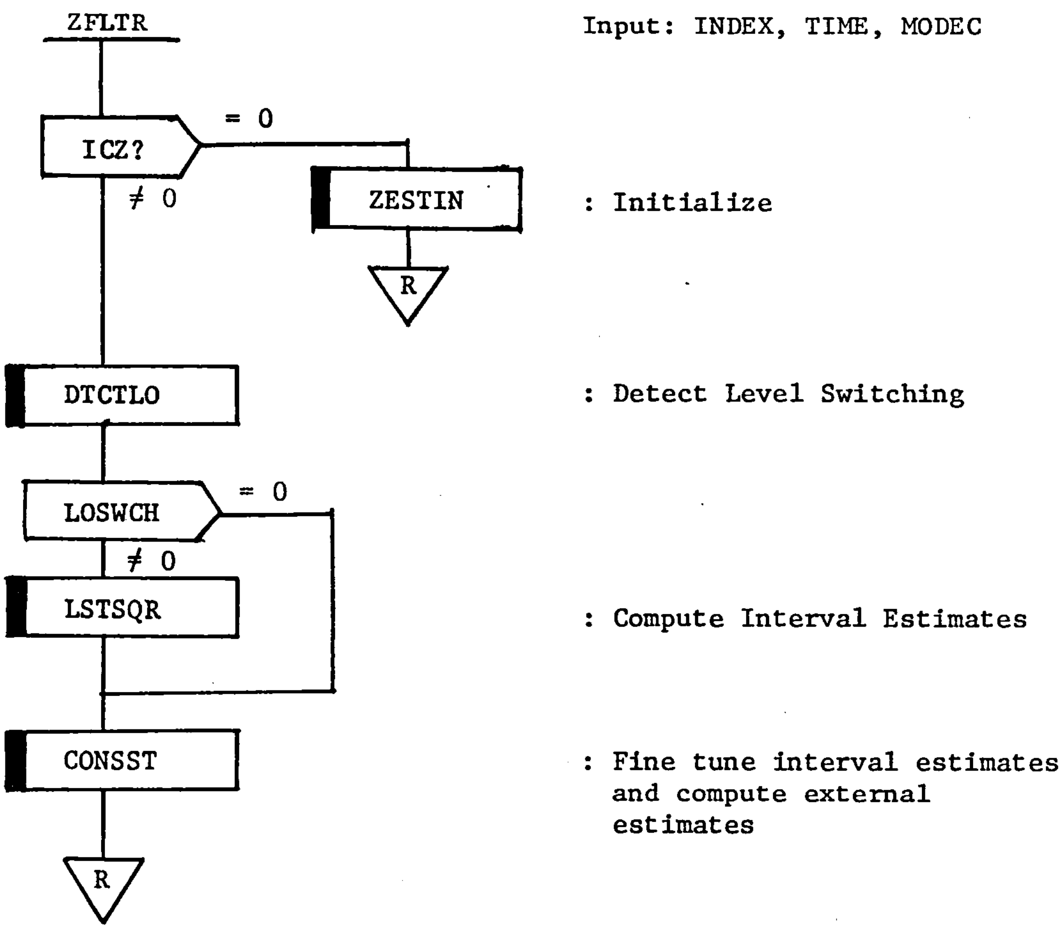


Figure 14. Vertical Tracker Algorithm Macro Flow Chart

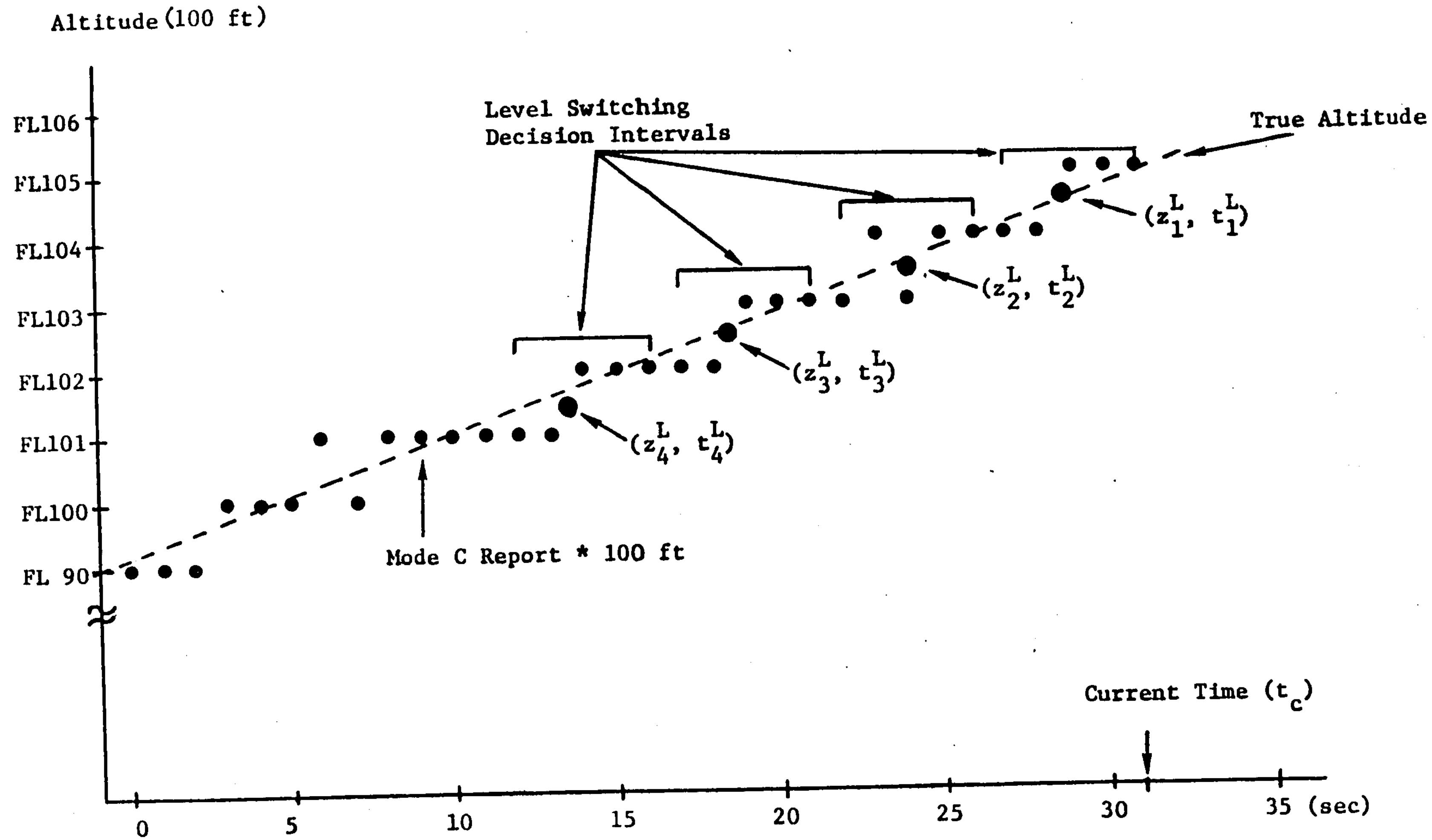


Figure 15. Illustration of Level Switching Times and Altitude Measurements

then LSTSQR is called to calculate the altitude and altitude rate rough estimates. Finally, the rough estimates are tested for consistency with respect to the measurements. Essentially the following computational steps take place. Reference [1] should be consulted for further detail.

The DTCTLO routine performs the level switching time detection in two ways. One is for cases where the average sampling period is slower than 4.5 sec or the last valid report time was more than 4 sec ago. The other case is for the more frequent sampling case. For the first case, average altitude and time are computed by

$$z_L = \frac{100}{2}\left[Ih_m(t_{LST,1}) + Ih_m(t_{LST,2})\right] \text{ ; and}$$
$$t_L = \frac{1}{2}\left[t_{LST,1} + t_{LST,2}\right] \tag{38}$$

For the second case, they are computed by

$$z_L = \frac{100}{5}\sum_{i=1}^{5} Ih_m\,(t_{LST,i}) \text{ ; and} \tag{39}$$

$$t_L = \left[\sum_{i=1}^{5} Ih_m(t_{LST,i}) * t_{LST,i}\right] / (z_L/20) \text{ .}$$

Here, $Ih_m\,(t_{LST,i})$ and $t_{LST,i}$ are the latest five valid stored Mode C reports and corresponding times. If the corresponding altitude level, L, given by

$$L = \text{Int}\ [z_L + 50)/100] \text{ ,} \tag{40}$$

is different from the previous level (stored L), then the level switching is declared, and z_L, t_L and L are saved in a 3-tier shift registers (i.e., current and previous two). It is noted that Eqs (39) represent the three-out-of-five rule.

If the level switching is detected by the above logic, a flag (LOSWCH) is set to 1, and the filter status indicator, ICZ, is incremented by 1. If not, LOSWCH is set to 0. Additionally, if the current and the one previous to the last levels are the same, the ICZ is set to 1 indicating a level flight.

Rough or internal altitude and altitude rate estimates are obtained by the LSTSQR routine when LOSWCH is 1. This is accomplished in three ways according to the ICZ indicator as follows:

(1) When ICZ = 1, then

$$z_{IN} = (z_{L,1} + z_{L,2})/2 \text{ and } \hat{\dot{z}}_{IN} = 0 \; ; \tag{41}$$

(2) When ICZ = 2, then a linear equation is used

$$\hat{z}_{IN} = z_{L,1} \text{ and } \hat{\dot{z}}_{IN} = (z_{L,1} - z_{L,2})/(t_{L,1} - t_{L,2}) \; ; \text{ or} \tag{42}$$

(3) When ICZ $\geq$ 3, then a three-point least squares formula is used

$$\begin{bmatrix} \hat{z}_{IN} \\ \hat{\dot{z}}_{IN} \end{bmatrix} = \frac{1}{D} \begin{bmatrix} \Delta_2^2 + \Delta_3^2 & -(\Delta_2 + \Delta_3) \\ -(\Delta_2 + \Delta_3) & 3 \end{bmatrix} \begin{bmatrix} z_{L,1} + z_{L,2} + z_{L,3} \\ \Delta_2 z_{L,2} + \Delta_3 z_{L,3} \end{bmatrix} \tag{43}$$

where

$$\Delta_2 = t_{L,2} - t_{L,1} ; \; \Delta_3 = t_{L,3} - t_{L,1} ,$$

and (44)

$$D = 2 \; [\Delta_2^2 - \Delta_2 \Delta_3 + \Delta_3^2] \; .$$

In the above, $(z_{L,i}, t_{L,i})$ are saved "average" altitudes and times with 1 being the latest and 3 being the oldest.

The internal estimates, $\hat{z}_{IN}$ and $\hat{\dot{z}}_{IN}$, are fine tuned by feedback laws in the CONSST subroutine, as follows. First, the time interval spent at the new altitude level is tested for consistency. The current time (t) minus the time the latest level switching was detected should not exceed either the expected time duration ($100/|\hat{\dot{z}}_{IN}|$) or the average duration $(t_{L,1}-t_{L,3})/2$); thus

$$t - t_{L,1} \quad : \quad 100/|\hat{\dot{z}}_{IN}| \qquad \text{(expected level occupancy time)}$$

$$t - t_{L,1} \quad ; \quad (t_{L,1}) - (t_{L,3})/2 \text{ (average level occupancy time).}$$

The actual logic is modified by tolerance factors for noise protection purposes. If $t - t_{L,1}$ exceeds either of two quantities, then a level flight is declared with proper actions. The idea is that if the altitude reports have not been changed for a long time duration, then the aircraft must have leveled off.

If the above test is passed (aircraft is assumed to maintain a steady altitude rate), then a feedback law is used to fine tune the internal estimates. The average prediction error is computed by

$$\begin{aligned} \tilde{z}_i &= 100\ Ih_m(t_{LST,i}) - [\hat{z}_{IN} + (t_{LST,i} - t_{L,1})\hat{\dot{z}}_{IN}] , \\ \tilde{z}_{av} &= \frac{1}{5}\sum_{i=1}^{5} z_i . \end{aligned} \tag{44}$$

Because the above error includes $\pm$ 50 ft of basic uncertainty due to the 100 ft quantization, it is modified by

$$\tilde{z}_{av} = \begin{cases} \tilde{z}_{av} - 50 & \text{if} \quad \tilde{z}_{av} \geq 52 , \\ 0 & \text{if} \quad |\tilde{z}_{av}| \leq 52 , \\ \tilde{z}_{av} + 50 & \text{if} \quad \tilde{z}_{av} \leq -52 . \end{cases} \tag{45}$$

This error is used to modify the rough estimates according to

$$
\begin{aligned}
\hat{z}_{IN} &= \hat{z}_{IN} + \alpha \cdot \tilde{z}_{av} \quad . \\
\hat{z}_{IN} &= z_{L,1} + [[\hat{z}_{IN} - z_{L,1}]]^{\tilde{z}_M} \quad , \\
\hat{\dot{z}}_{IN} &= \hat{\dot{z}}_{IN} + [[\beta z_{av}]]^{\tilde{\dot{z}}_M} \quad .
\end{aligned}
\tag{46}
$$

where

$[[\ x\ ]]^M \equiv x$ is limited to the maximum value M in magnitude.

Terms in these equations are

$\tilde{z}_M$ = 25 ft = maximum allowable altitude tuning,

$\hat{\dot{z}}_M$ = 8 fps = maximum allowable altitude rate tuning,

α = 0.632 = altitude error gain, and

β = 0.25 = altitude rate error gain.

After the rough estimates are fine-tuned, the external estimates are computed as follows:

$$
\begin{aligned}
\tilde{z}_{EX} &= \hat{z}_{IN} + (t - t_{L,1})\, \hat{\dot{z}}_{IN} \quad , \\
\hat{\dot{z}}_{EX} &= \dot{z}_{IN} \quad .
\end{aligned}
\tag{47}
$$

Equation (47a) signifies that the internal altitude estimate, $\hat{z}_{IN}$, is computed and referenced at time $t_{L,1}$ (the last time level switching was detected). Therefore, the second term, $(t - t_{L,1})\, \hat{\dot{z}}_{IN}$ accounts for the altitude rate up to the current time, t.

It is noted that the ZFLTR routine is invoked even though the target is still in the acquisition region (horizontal components are not updated in this region). There are two reasons: (1) altitude error is independent of range; and (2) long time delays in the altitude rate estimate need to be minimized.

Surveillance (or Interrogation) Scheduling The surveillance scheduling function is performed by two subroutines - SCHNXT and CHKSMP. The SCHNXT subroutine identifies the target which requires the surveillance for the upcoming cycle period, and CHKSMP checks the scheduled list. These are the last and the first traffic sensor routines in any given cycle time.

The SCHNXT routine examines if the pilot selected any CDTI target.* If so, the selected targets t_{NXT} (TNXT) is set to 0. All the pop-up targets are treated the same way (NPOP counter is decremented by 1) until NPOP becomes 0. Then it is no longer a pop-up.

The surveillance time counters (t_{NXT}) computed by the ESTMTN routine for all other targets are decremented by 1. If they are greater than 0, then the corresponding target is not scheduled. (This is indicated by NXTS = $-|IDF|$.) If the counter is between 0 and -4, then the corresponding target is scheduled. (This is indicated by NXTS = $+|IDF|$.) If the counter is less than or equal to -5, then the corresponding target is deleted from the traffic file, because it is no longer in Own's vicinity.

The CHKSMP routine determines which target is not scheduled for the surveillance transaction. This is accomplished by examining the input aircraft identification number (IDENT) with the ones in the NXTS list. There are two cases to be considered: (1) IDENT is not in the traffic file or is already scheduled; and (2) IDENT is contained in the NXTS list. In Case (1), the rest of the sensor simulation computation needs to be performed. In Case (2), further computation is not necessary for this target except for the COAST (position prediction) routine.

* This option is not a TCAS system requirement; it is added for CDTI research purpose.

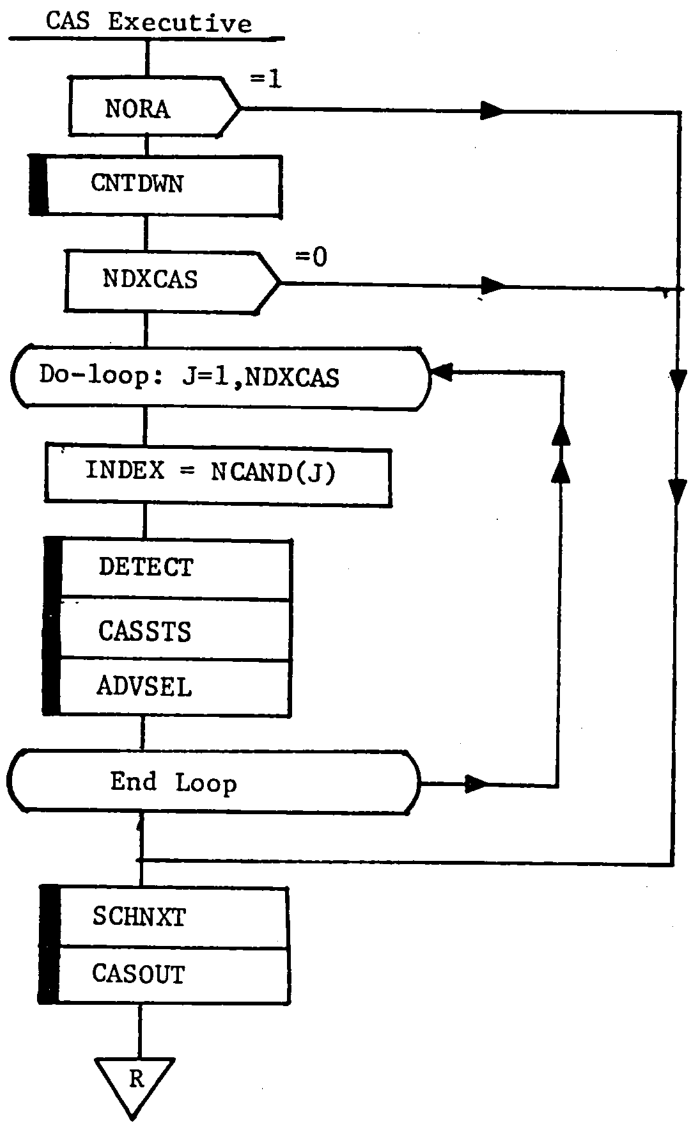


Figure 16. Vertical CAS Logic Executive Macro Flow Chart

Collision Avoidance Logic Module

The collosion avoidance logic module contains nearly 20 subroutines which can be grouped into the following major components:

(1) Threat detection;

(2) Counter and housekeeping routines;

(3) Climb/descent sense selection;

(4) Advisory selection algorithm; and

(5) Advisory packing and outputing.

Before each component is discussed in detail, it is advantageous to examine the CAS executive logic to gain the over-all logic structure. Fig. 16 shows the vertical CAS logic executive macro flow chart and is a lower portion of the BXTCAS macro flow chart shown in Fig. 11.

First, the CAS operational status flag (NORA) is checked. If it is 1 (or up) due to pilot selection or the Own altitude being too low, then the entire CAS logic is skipped. The CNTDWN subroutine is then called to update counters and timers for each threat in the CAS file. If there is no candidate traffic to examine, then the rest of the logic is skipped. The candidate traffic is defined to be aircraft which are in the track region during the current sensor cycle. The traffic file indecies are saved in the NCAND list.

For each candidate in the NCAND list, DETECT, CASSTS, and ADVSEL routines are invoked. The functions of these routines are threat detection, threat detection status update and advisory (or escape maneuver) selection. After all the candidate traffic are cycled through, the CASOUT routine is invoked to pack advisories for output purposes.

The internal CAS file elements are dimensioned five so that up to five threats can be accommodated.

Threat Detection Logic The DETECT subroutine performs the detection logic computation. This routine contains all three threat situations - proximity, traffic and resolution Advisory (PA, TA and RA) tests. The logic and computational steps are as described before in chapter III. Table 20 summarizes the top-down computations performed by DETECT. References are made to tables and equations so that the details can be supplied. The following notes correspond directly to the item numbers appearing in Table 20.

Note (1): Temporary flags LPA, LTA and LRA are used to track the status. They are defined as follows:

Reset	LTA (LRA) = 0
Range test pass	LTA (LRA) = 1
Altitude test pass	LTA (LRA) = LTA + 10 (LRA+10)

Therefore, if both range and altitude tests are passed, then the flags would have a value of 11; otherwise they would be 0, 1 or 10.

Note (2): The target sensitivity level is computed only for Mode S. The combined sensitivity (LVLINT) is given by

LVLINT = min {LVLOWN, LVLINT},

where LVLOWN and LVLINT are computed according to Table 4. The level index, LVLNDX, is given by

LVLNDX = LVLINT-2

LVLNDX is used to index the altitude sensitive CAS parameters.

Table 20. DETECT Subroutine Computational Steps

(1) Reset CAS temporary flags (LPA, LTA, LRA = 0)

(2) Compute TCAS sensitivity level and index (LVLINT, LVLNDX) (Table 5)

(3) IF Mode A target, perform Mode A PA tests. GO TO Step 11.

(4) Perform PA test for Modes C and S.

(5) Perform special TA and RA tests, IF it is a pop-up. GO TO Step 11.

(6) Compute range and range-rate (Tables 6 and 7)

(i) perform divergent TA and RA range tests, or
(ii) perform convergent TA and RA range tests

IF both TA and RA range tests fail, GO TO Step 11.

(7) Compute relative altitude and altitude rate.

(8) Perform TA altitude test. (Table 7)

(i) current altitude separation small, or
(ii) current altitude separation large

(9) Compute vertical miss distance (VMD) (Table 10)

(10) Perform RA altitude test

(i) current vertical separation small [Eq (28)], or
(ii) current vertical separation large [Eq (29)]

(11) Check if any of the PA, TA or RA tests are passed by testing LPA, LTA and LRA flags. If so, set the next surveillance flags (NPOP and TNXT).

(12) EXIT

Note (3): As mentioned previously, the PA test for Mode A target is given by $\Delta r \leq 2$ nmi . AND . $\hat{z}_0 \leq 15{,}500$ ft

The traffic test is not performed for a Mode A target, since PA and TA for Mode A are essentially for pilot warning purposes, and the PA range test contains the TA range test.

Note (4): The PA test for Modes C and S targets is given by

$$\Delta r \leq 2 \text{ nmi . AND . } |\hat{z}_0 - \hat{z}_T| \leq 1200 \text{ ft}$$

Note (5): These are special tests unique to this simulation. When the target is known for less than 24 sec, then it is treated as a pop-up, and the following test applies.

$$|\Delta z| < 2^* \theta_Z \text{ . AND . } \Delta r < 2^* \Delta r_{TA} \Rightarrow LTA = -11$$

$$|\Delta z| < 2^* \theta Z \text{ . AND . } \Delta r \quad 2^* \Delta r_{RA} \Rightarrow LRA = -11$$

The rationale is that during this initial period it could be that the TCAS has possibly two surveillance data points about this target; this is not enough data to ascertain important estimates. The thresholds are taken to be twice the nominal, and the negative signed flags warn the pilot to look out for the traffic (VFR acquisition).

Note (6): The vertical miss-distance, VMD (Δz^+), is computed by the VRTMD subroutine.

Additional Note: The TA and RA test thresholds are given in Tables 7 and 9 respectively. Within the program, values corresponding to the sensitivity levels of 3, 4, 5, 6, and 7 are stored, and these correspond to the level indecies of 1 through 5.

Counter and Housekeeping Routines Two subroutines - CASSTS and CNTDWN - are involved in this category. Each is explained below.

The CNTDWN routine counts down each of the 5 sec advisory counters (IPACNT, ITACNT and IRACNT), the RA counter (NSTS), and the 10 sec timer (TMCNT). The 5 sec and RA counters are divided by 2 (i.e., right-shift by 1 operation) and the timer is decremented by 1. The results are that the "inactive threat" is dropped from the display in 5 sec and becomes dormant for 5 more seconds until the 10 sec timer runs out. When this happens, the CAS file for that traffic is cleaned and readied for other threats. The RA counter is used to prevent an on-again off-again Resolution Advisory.

The CASSTS routine monitors the CAS status by performing the following functions:

(1) Assign a file slot if it is a threat;
(2) Test against the previous threat status; and
(3) Clean up if no longer a threat.

Table 21 shows the top-down computational flow for this subroutine. The following notes correspond to the item numbers in the table.

Note (1): If the aircraft is an RA threat, then it is automatically a TA threat which implies that it is a PA threat.

Note (2): The IDCAS list contains the index number of the aircraft in the traffic file, /BNDXFL/, so that the accessing of the traffic file for a threat is direct.

Note (4): If all the entries in the IDCAS list are full, then the further processing of aircraft is held until the next cycle time. A better way is to bump a lower ranking threat, i.e., if the aircraft is an RA threat, then the PA or TA threat in the CAS file should be deleted and the slot given to the RA threat.

Table 21. CASSTS Subroutine Computational Steps

(1) Set LTA = 11 if LRA = ± 11
LPA = LRA if LTA = ∓ 11

(2) Look through IDCAS list for this aircraft.

(3) IF not in the list and non-threat, RETURN.

(4) IF new threat, look for a CAS file slot.
IF found, assign to this aircraft; OTHERWISE RETURN.

(5) Set timer and counters
TMCNT = 10
NSTS, IPACNT, ITACNT, IRACNT = 16

(6) Check against previous status and set flags, IPAFLG, ITAFLG and IRAFLG.

(7) IF the counters are all zero, clean-up the file slot.

(8) RETURN.

Note (6): This is done by the subfunction LADSTS. For example, $IRAFLG_{(New)}$ = LADSTS ($IRACNT_{(New)}$, $IRAFLG_{(Old)}$, 16). Functional evaluations are defined by the following table with explanations.

Table 22. Functional Evaluation of LADSTS

$IRACNT_{(New)}$	$IRAFLG_{(Old)}$	$IRAFLG_{(New)}$	EXPLANATION
non-zero	-1	1	continuing threat
16	0	-1	new threat
0	1	0	old threat

$IRAFLG_{(New)} = -1$ signifies that this aircraft is a threat, therefore an RA must be generated (but only once). If it is 1, then this is a continuing threat, i.e., an RA already exists. If it is 0, then the aircraft was an old RA threat. Thus, the old RA must be deleted.

It is noted that the old logic contained the so-called 2-out-of-3 counting rule, i.e., a threat is not declared (or nullified) if not done so in two out of the latest three cycles. This rule made the old versions of CASSTS and CNTDWN routines very complex.

Advisory Selection Mini-Executive (ADVSEL) The climb/descend sense selection, advisory selection algorithm, and advisory packing modules are controlled by a mini-executive called ADVSEL. It examines the CAS status flags (computed by the DETECT and CASSTS routines) and calls appropriate subroutines depending on their status. Figure 17 shows a detailed computational flow chart of ADVSEL. An explanation follows.

First, the RA flag is tested. (JDX is the CAS file index for this threat). If it is positive, then it is a continuing RA threat, and presumably an RA has been generated; therefore, exit. If the flag is zero, then it was an RA threat previously; therefore, downgrade it to a TA threat and test further. If the RA flag is negative, then it is a new RA threat; therefore, an RA must be generated. Now, test LRA (temporary RA flag generated by the DETECT routine). If it is -11, this signifies a special pop-up; therefore, pack a special RA by calling SPCLRA subroutine and exit. If LRA is 11, then this is a "regular" RA threat; therefore, determine the climb/descend sense by calling SELSNS and generate an escape vertical maneuver by calling SELADV. If a satisfactory maneuver was not found (NOMAN = -1), then pack a special RA by calling SPCLRA. Otherwise, pack a "regular" RA and exit.

If the aircraft is not an RA threat, then the TA flag (ITAFLG) is tested. If it is positive, then it is a continuing TA threat; therefore, exit. If it is 0, then it was a TA threat in the previous cycle. Therefore, downgrade it to a PA threat, and perform further tests. If ITAFLG is negative signifying a new TA threat, then pack a TA and exit.

The proximity advisory portion is entirely analogous to the TA logic. Major subroutines called by the ADVSEL mini-executive are SELSNS, SELADV and various packing routines. These are explained below.

Climb/Descend Sense Selection (SELSNS) As mentioned previously, the CAS logic chooses the climb/descend sense only once per RA encounter, and the chosen sense will remain in force until the threat is no longer. Basi-

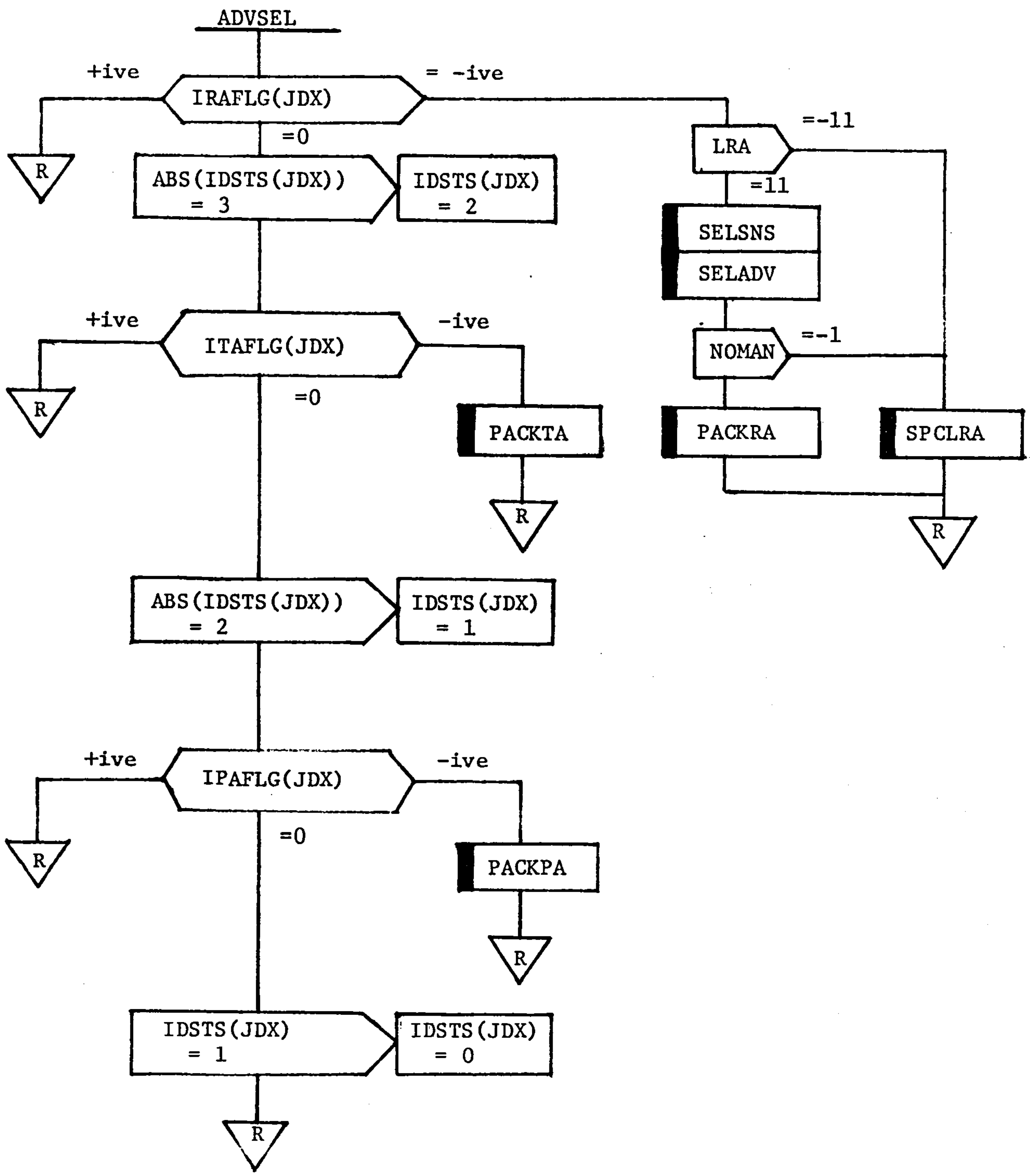


Figure 17. ADVSEL Detailed Flow Chart

cally, the following computational sequence takes place. Threat altitude is projected to critical times (defined by the range closure tau's) by assuming the current altitude rate is maintained. Own altitude is projected to the same critical times by assuming $\pm$ 1500 fpm of nominal vertical rate, 8 fpss acceleration to attain the nominal rate, and a 5 sec maneuver delay time, respectively. Using these projected altitudes, the projected altitude separations are computed. These are compared with each other and with the separation threshold to determine the sense.

Table 23 shows the top-down computational flow for the SELSNS subroutine. The following notes correspond to the item numbers in the table.

Note (2): The range and modified range tau's are given by Eq (25) with the range rate limited to RDTHR (10 kt) for the 0 divide check. The tau's are limited by t_{vc} (TVPCMP) from above. The modified tau's are limited to 10 sec from below.

Note (4): Projected altitudes are computed using Eq (30) which is implemented as the ZPROJ subfunction. The projected altitudes are limited to the field altitude. For computation and determination of the sense flag (LSENSE), see Table 10.

Note (5): The "Don't-care" flag (LDCFLG) is set to 1 if both climb and descend satisfies the safe separation criteria ALMT (θ_a). This flag would be used in the multi-threat logic.

Resolution Advisory Selection (SELADV) The software module for selecting a proper escape maneuver (Resolution Advisory) is very complex due to its logical rather than computational nature. The main idea of how this is done was explained in Chapter II. The procedure is explained further from the simulation program view point.

Figure 18 shows the SELADV subroutine tree. It shows two major branches. One is for generating a negative (or vertical speed limit) advisory, and the other is for a positive advisory. Each of the subroutines are discussed in the following paragraphs.

Table 23. SELSNS Subroutine Computation Steps

(1) Reset CAS logic flags (INTENT, INTHR, LSENSE, LDCFLG, LVSLDK, LADNOK)

(2) Limit the range and modified range tau's and set the lower altitude bound and the delay time.

(3) IF the tau's are shorter than the delay time, simply project the threat and Own altitude and compute the separation; GO TO Step (5)

(4) OTHERWISE
Compute Own projected altitudes assuming +1500fpm
Compute Own projected altitudes assuming -1500fpm
Determine worst case separations for climb and descend

(5) Determine the sense (LSENSE) comparing the two separations
Assign second choice separation
Set "Don't-care" flag (LDCFLG) if second choice is satisfactory.

(6) RETURN

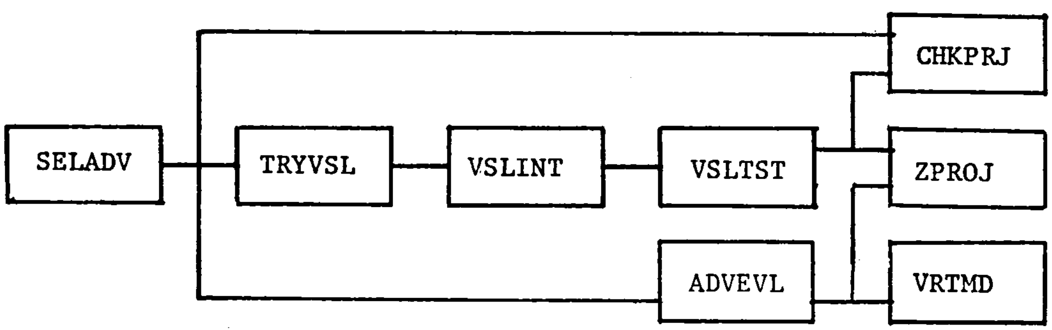


Figure 18. Resolution Advisory Selection Subroutine Tree

Figure 19 shows a detailed computer flow chart for the SELADV subroutine. First, the threat altitude rate is tested. If its magnitude is less than 16.67 fps (1000 fpm), then the current altitude separation is tested against the safety limit (ALMT) also considering the selected sense (this is done by the CHKPRJ subroutine shown in Fig. 20). If the current separation is safe (indicated by the in-threshold flag, INTHR, being 0), then negative advisories are examined by invoking the TRYVSL subroutine.

If the threat altitude rate exceeds 1000 fpm in magnitude or the current separation is not safe (from the previous test), then the vertical miss distance (VMD) is tested against the safety limit by considering the selected sense. If it is within the threshold, or the Own altitude rate exceeds 10 fps (600 fpm), then negative advisories are examined. Otherwise, the positive advisory corresponding to the selected sense is assumed (this is achieved by setting the VSL packed word, INTENT, to 0), and its effect is examined by the ADVEVL subroutine.

Figure 21 shows a detailed computer flow chart for the TRYVSL subroutine. If the threat is diverging in range, then the negative advisory is chosen by setting the first "bit" of INTENT. (Interpretation of INTENT bits will be explained later.) If the threat is converging, then the weakest advisory (i.e., limit vertical speed to 2000 fpm) is tested by invoking the VSLINT subroutine. If this is sufficient (the vertical-speed-limit-OK flag, LVSLOK, = 1), then the corresponding INTENT bits are set. If it is not satisfactory, then the next stronger advisory is tested. This process is continued (see Figure 8) until the negatives (don't climb/descend) are tested. If it is safe, then the first INTENT bit is set. Otherwise, the positive advisory is tested by setting INTENT to zero and invoking the ADVEVL subroutine.

Figure 22 shows a detailed computer flow chart for the VSLINT subroutine. Its function is to test the vertical miss distance with the assumed resolution vertical speed. This is done by testing the projected altitude separations at two critical times - the range tau and the modified range tau. Actual testing is performed by the VSLTST subroutines. It is noted that the safety margin is modified by the ALIMOD amount.

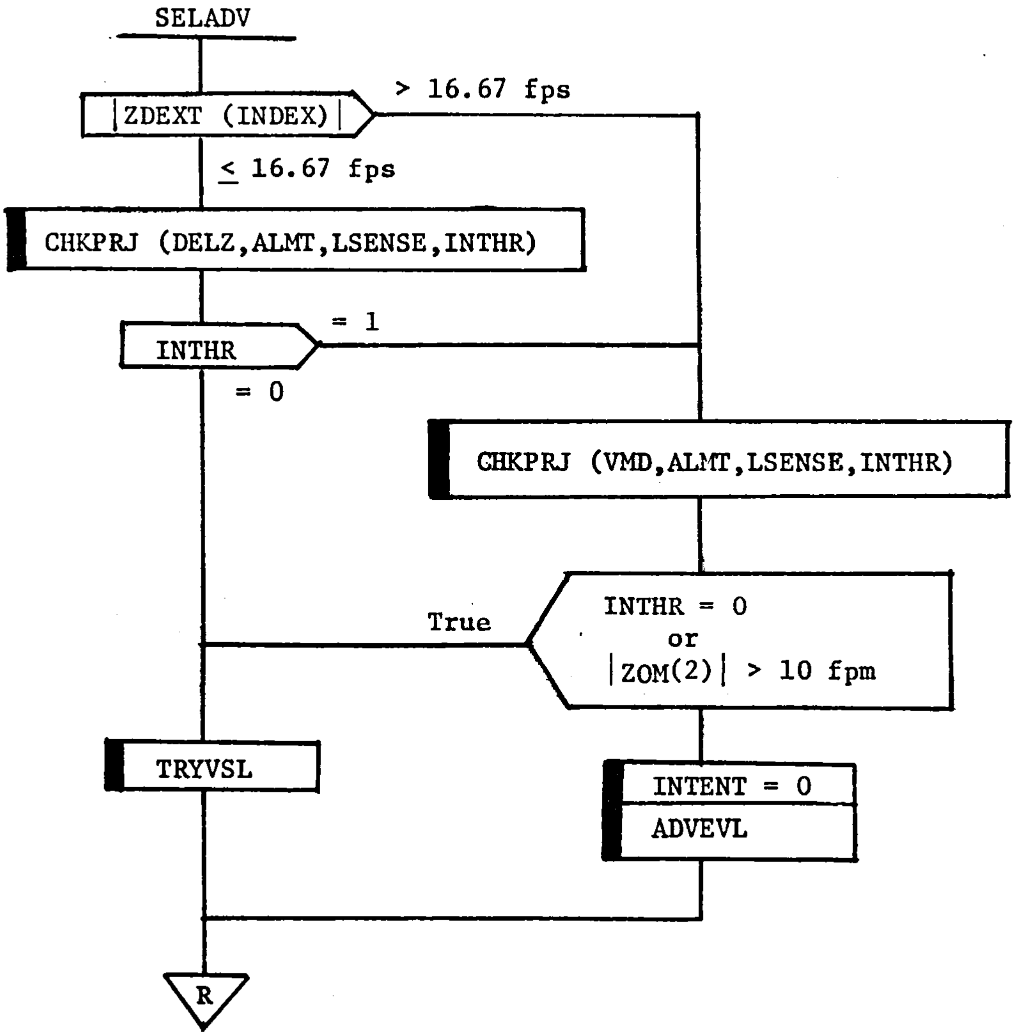


Figure 19. SELADV Detailed Flow Chart

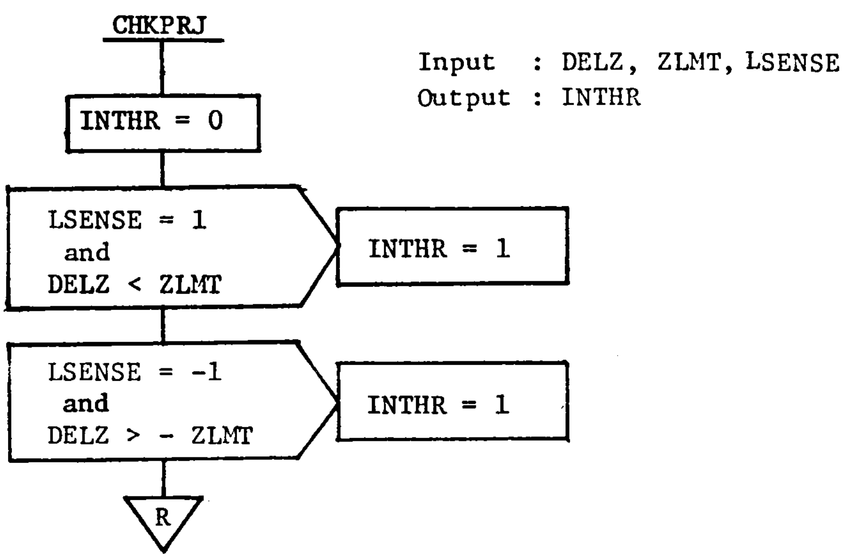


Figure 20. CHKPRJ Detailed Flow Chart

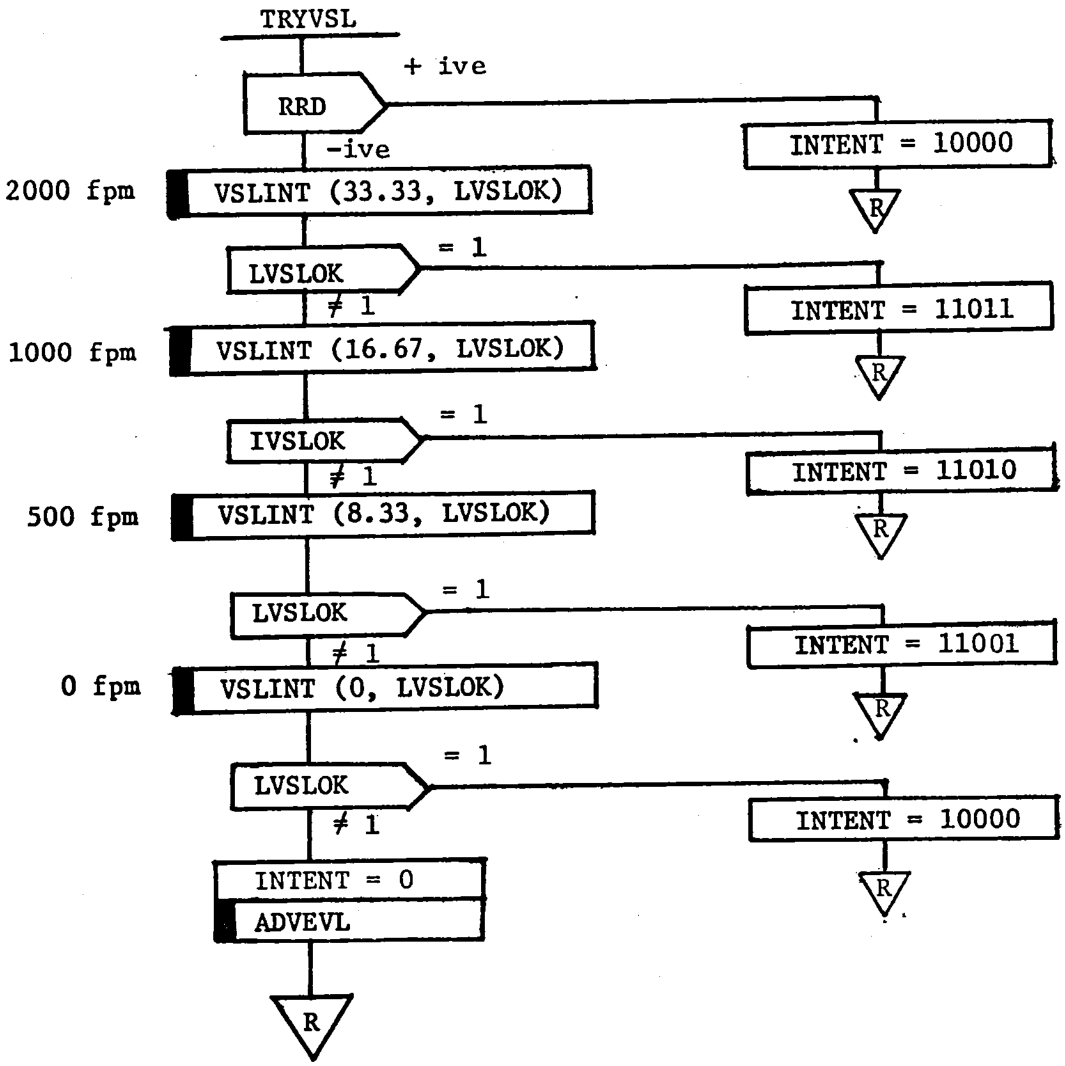
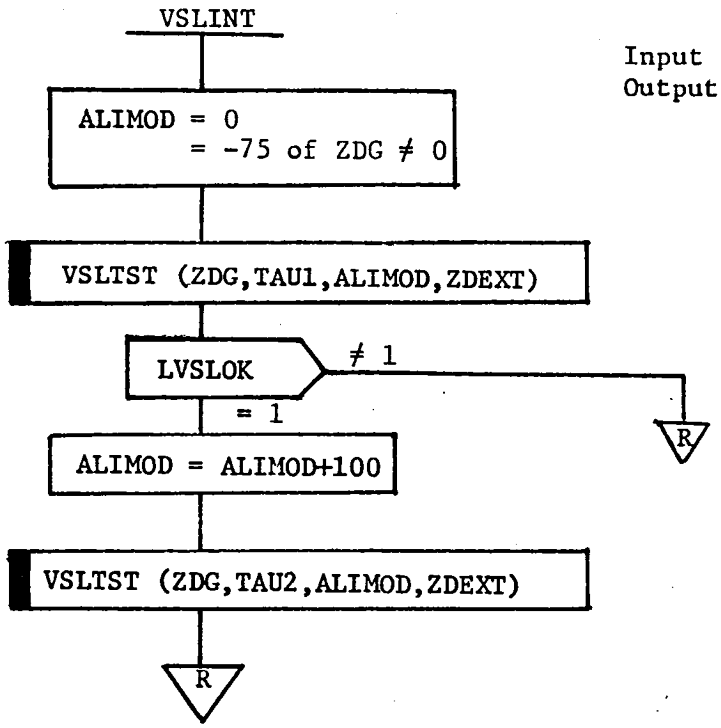
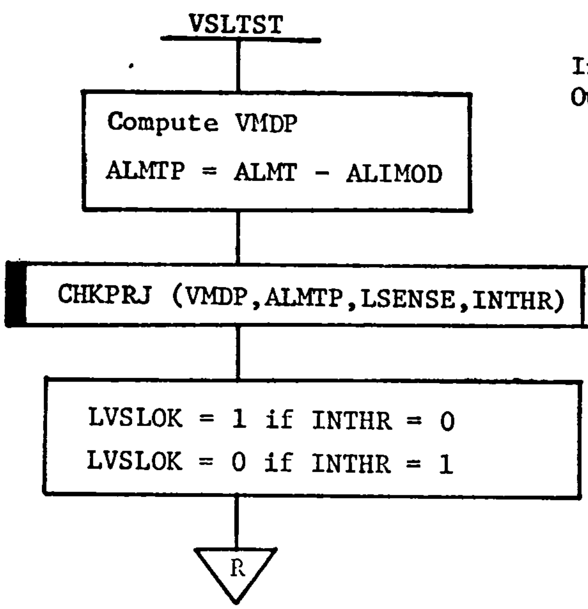


Figure 21. TRYVSL Detailed Flow Chart



Input : ZDG
Output : LVSLOK

Figure 22. VSLINT Detailed Flow Chart



Input : ZDG, TAUP, ALIMOD, ZDEXT
Output : LVSLOK

Figure 23. VSLTST Detailed Flow Chart

Figure 23 shows the detailed flow chart for the VSLTST subroutine which does the testing at each of the two critical tau's. First, the projected altitude separation is computed based on the altitude rate defined by this VSL advisory ($\dot{z}_G$), the critical tau (τ), delay time (5 or 0 sec) and the sense indicator (LSENSE). Table 24 defines the conditions and definitions of the separation, VMDP. Then, the projected vertical separation is compared with the modified safety limit. Depending on the in-threshold flag, INTHR, the vertical-speed-limit-OK flag (LVSLOK) is set. LVSLOK = 1 signifies that this speed limit is satisfactory.

Figure 24 shows a computational flow chart for the ADVEVL subroutine. It is remembered that the ADVEVL routine is called if a VSL or a negative advisory was not found, and a positive advisory must be issued. Also, the separation due to a positive advisory was never computed or tested. This routine does just that. First, the desired Own speed is computed (0,$\pm$1500 fpm, or Own current speed). Then based on the desired altitude rates, the expected vertical miss distance is computed. See Table 10 for this computation. If the expected separation exceeds the 100 ft safety limit, then set the advisory-not-OK flag (LADNOK) to 0. If the separation is within the 100 ft limit, then LADNOK is set to 1.

Table 24. VMDP Computation Table

LSENSE	$\dot{z}_0$(fps)	VMDP (ft)
-1 (descend)	> ZDG	$z_0^+ = \text{ZPROJ}\ (\hat{z}_0, \hat{\dot{z}}_0, \tau', 5.0, \text{ZDG}, \text{LSENSE})$* $\text{VMDP} = z_0^+ -(\hat{z}_T + \hat{\dot{z}}_T - \tau')$
	$\leq$ ZDG	$\text{VMDP} = (\hat{z}_0 - \hat{z}_T) + (\text{ZDG} - \hat{\dot{z}}_T)\ \tau'$
+1 (climb)	< - ZDG	$z_0^+ = \text{ZPROJ}\ (\hat{z}_0, \hat{\dot{z}}_0, \tau', 5, -\text{ZDG}, \text{LSENSE})$ $\text{VMDP} = z_0^+ - (\hat{z}_T + \hat{\dot{z}}_T\ \tau')$
	$\geq$ - ZDG	$\text{VMDP} = (\hat{z}_0 - \hat{z}_T) - (\text{ZDG} + \hat{\dot{z}}_T)\ \tau'$

*z_0^+ is the projected altitude at τ' if ZDG is the desired rates. See Eq (30).

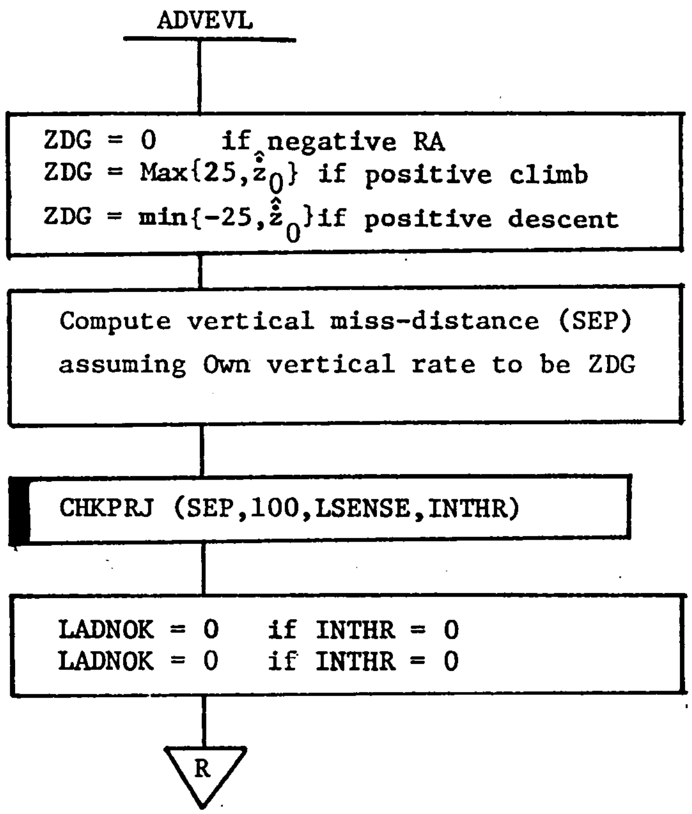


Figure 24. ADVEVL Detailed Flow Chart

The fact that this flag is set to 1 indicates that the CAS logic <u>cannot</u> find a satisfactory resolution (positive or otherwise); therefore, the pilot must be warned immediately. This concludes the Resolution Advisory Selection module description.

<u>Advisory Packing and Outputting</u> There are four routines which pack the CAS file - PACKRA, SPCLRA, PACTA, and PACKPA. The CASOUT routine goes through each entry in the CAS file and "summarizes" the output. These routines are dependent on the external device which utilizes the output data. Therefore, they should be modified according to the external device requirements.

Table 25 summarizes the four packing routine operations. These are not necessarily the requirements; however, these are thought to enhance the graphic display on the cockpit CRT.

The CASOUT routine scans through the CAS file and obtains a single set of advisory outputs including text. First, the RA threats are scanned, and the latest RA is picked, and the outputs are packed. Depending on the LTENT flag, the output variables - IDCMD, ZDGOUT and RATEXT - are set through a table-look-up procedure performed by the RAMAP routine. The relationships among LTENT, IDCMD, ZDGOAL and RA TEXT are given in Table 26.

Table 26. CAS Output Variable Map

LTENT	IDCMD	ZDGOAL	RA TEXT
01000	-1	0.	WARNING
00000	1	1500.	CLM
10000	2	0.	DDES
11001	3	-500.	DDES 500
11010	4	-1000.	DDES1000
11011	5	-2000.	DDES2000
00100	6	-1500.	DES
10100	7	0.	DCLM
11101	8	500.	DCLM 500
11110	9	1000.	DCLM1000
11111	10	2000.	DCLM2000

Table 25. CAS File Variables Stored by the Packing Routines

CAS File Variables	PACKRA	SPCLRA	PACKTA	PACKPA
IDSTS	3	-3	2	1
LTENT	INTENT(+100)(1)	1000	0	0
TMCAS	TIME	TIME	TIME	TIME
TMCPA	TAU1	0	TAU1	0
XCPA	$\hat{\Delta x} + \tau_1 \cdot \hat{\dot{\Delta x}}$	0	$\hat{\Delta x}(+ \tau_1 \cdot \hat{\dot{\Delta x}})$ (2)	0
YCPA	$\hat{\Delta y} + \tau_1 \cdot \hat{\dot{\Delta y}}$	0	$\hat{\Delta y}(+ \tau_1 \cdot \hat{\dot{\Delta y}})$	0
DZCPA	$\hat{\Delta z} + \tau_1 \cdot \hat{\dot{\Delta z}}$	Δz	$\hat{\Delta z}(+ \tau_1 \cdot \hat{\dot{\Delta z}})$	0
ZDGOAL	ZDG	0	0	0
IRAFLG	1(3)	-	-	-
ITAFLG	1	1	-	-
IPAFLG	1	1	1	-

(1) LTENT = INTENT + 100 implies that the third bit is set if LSENSE is negative (descend).

(2) If $\tau_1 \geq 0$.

(3) These are set elsewhere in CASSTS.

This table corresponds directly to Table 15 with IDCMD matching the RA Map Number. (IDCMD = -1 was added to account for the special RA case applicable to the pop-up threat.) The difference between INTENT and LTENT is that LTENT contains the climb/descend (LSENSE) bit. The third bit of LTENT is the sign bit, (i.e., if it is 1, then the selected sense is descent. LTENT is part of the Packed Word (OWNTENT in the MITRE logic). These are (8, 7, 6, 2, 1) bits from the right.

This concludes the description of the BXTCAS simulation program. The sensor simulation part is believed to be operationally accurate. At least no significant parts are knowingly omitted. An approximation was made for computing the next schedule time. The CAS logic simulation part follows the MITRE logic(as represented in the draft TCAS MOPS [2] by the pseudo E-code). There are a few exceptions. These are:

(1) Only the vertical logic is simulated;
(2) The coordination logic between TCAS systems is not simulated;
(3) The multi-threat logic is not included; and
(4) The altitude tracker algorithm is entirely different.

Some comments and remarks are in order.

(i) At this time only the vertical logic is sufficiently mature to construct a simulation model. The horizontal logic is considered to be at a design stage. Our previous model was updated due to specification changes especially in the detection logic.

(ii) The coordination and multi-threat logic are not included in the simulation because of no immediate CDTI needs. These two functions add a substantial complexity to the program without apparent or direct CDTI research pay-off.

(iii) The CAS logic contains dedicated estimation functions which are performed in range and altitude axes. In the present simulation this function is imbedded into the sensor module. Furthermore, the estimation function is carried out in an Own attitude stabilized north-east-down coordinate system.

(iv) The vertical tracker algorithm implemented in the simulation program is based on the work reported in Ref. 1. Its performance is expected to be similar to the MITRE design; however, the coding requirement is less in terms of memory. See Ref. 1 for the detailed description and performance analysis.

IV

SIMULATION VALIDATION

The BXTCAS simulation program is written in standard Fortran IV and is highly transportable among general purpose main-frame computers (CDC-7600, CRAY X-MP) or mini-computers (VAX 11-780). The program contains approximately 3,000 lines of source code including a simple traffic generator.

During the course of the development of the BXTCAS, the TCAS simulation program was imbedded into a semi-real time operating environment to help debug, exercize and validate the TCAS module. These efforts were carried out essentially in two steps - the sensor part and the CAS part.

An earlier version of the TCAS simulation module was implemented in a real time active simulator at NASA/Langley Research Center. It was used to drive a CDTI display with a realistic traffic pattern and full pilot work-load experiments in April, 1984. This proved the utility of the program.

In this chapter, the TCAS simulation validation efforts are discussed, and results are presented.

Simulation Set-Up

Figure 25 shows a <u>detailed</u> flow chart of the simulation set-up to exercize and validate the TCAS module. It consists of the following elements:

ACDY: Air traffic (40 aircraft including Own) generation routine integrated at 0.5 sec interval;

ACIC: Initialization routine for ACDY; initial values are selected randomly;

XNSFR: Creates the TCAS input file from the traffic generation routine;

BXTCAS: Traffic sensor and vertical collision avoidance logic model based on the Bendix enhanced TCAS II design; and

OUTPUT: Output routine.

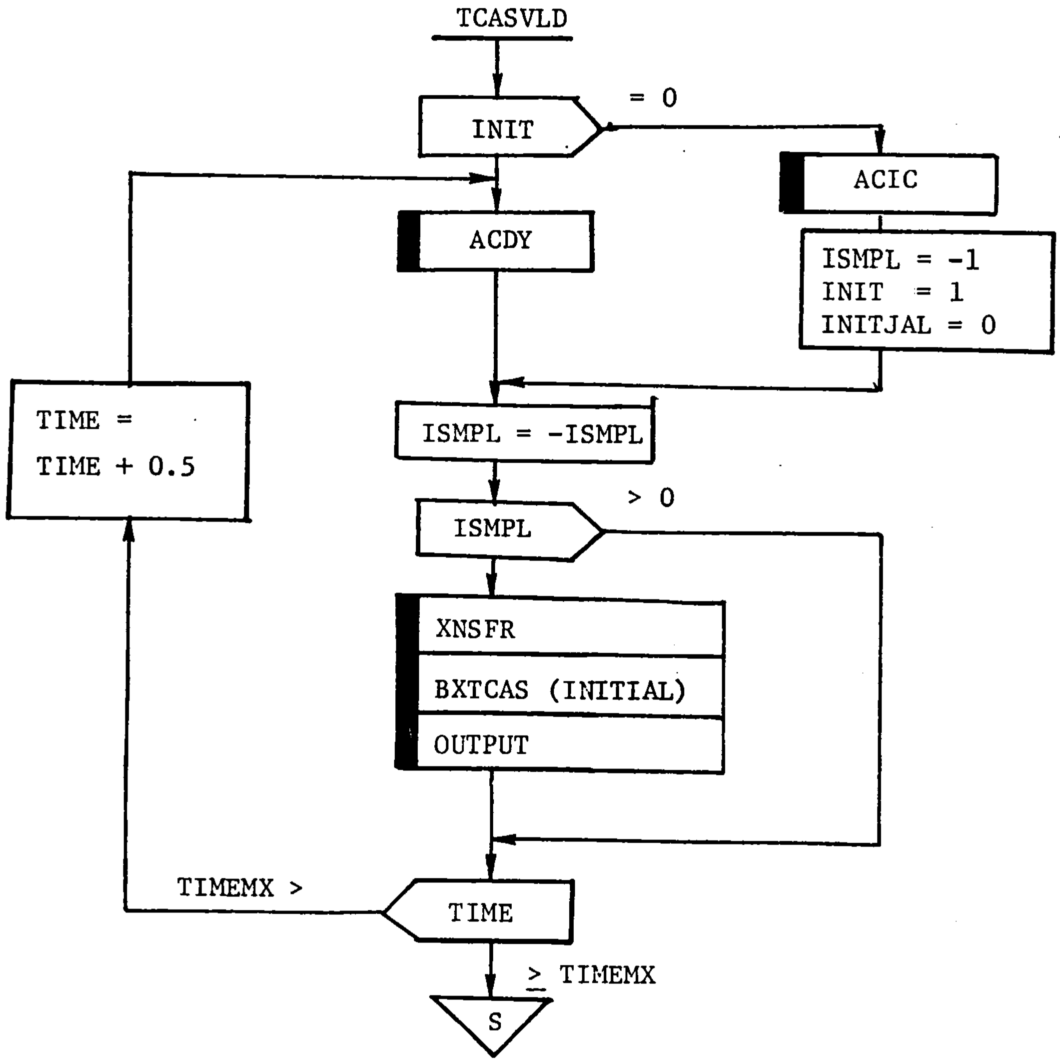


Figure 25. TCAS Simulation Validation Program

The aircraft dynamic states are integrated every 0.5 sec to minimize the numerical integration error. By closing the time loop around the traffic generator, semi-real time operation is achieved. The kinematic variables are input to the BXTCAS module through a data transfer routine (XNSFR) every second (because of the ISMPL flip-flop). After the call to BXTCAS results are printed (or stored in arrays) through the OUTPUT routine, then the time is advanced by 0.5 sec and this cycle is repeated. It is noted, however, that up to 40 aircraft are generated during the same time frame and the kinematic data concerning all aircraft are transferred simultaneously. In a real system, at least the surveillance operation would be performed one aircraft at a time distributed over one second. The processing of the surveillance data may be timed at one second intervals, if the surveillance data are stored in a buffer.

As mentioned previously, 40 aircraft are generated in ACDY by using simple point-mass 7-state dynamic equations. The details are given in Figure 26 and Table 27. The feedback loop in each control axis results in a first order lag characteristics between the command and response. This gives altitude or airspeed hold capability. The lateral axis is assumed to be coordinated. The pitch angle is assumed to be the flight path angle. The cross-feed of the pitch error to the airspeed loop is introduced to simulate the difference in the respective loop closure time constants.

The various rate and authority limits are thought to represent commercial airliner operations. The states are integrated using the second order Euler (trapezoidal) method.

The state and command variables are initialized in the ACIC subroutine. Nominally, Own aircraft is placed at the origin flying due north with level flight at 6000 ft altitude and 200 kt airspeed. Other aircraft are initialized with respect to Own. The initial pitch, heading, airspeed, range, bearing and relative altitude are drawn from uniform density functions. The north-east-down coordinate positions are then computed from the range, bearing and relative altitude values. The roll, pitch and airspeed commands are initialized to the corresponding state variables so that there will not be initial transients. Table 28 lists the specific numerical values chosen for the validation cases. The randomized initialization (Monte Carlo) method was

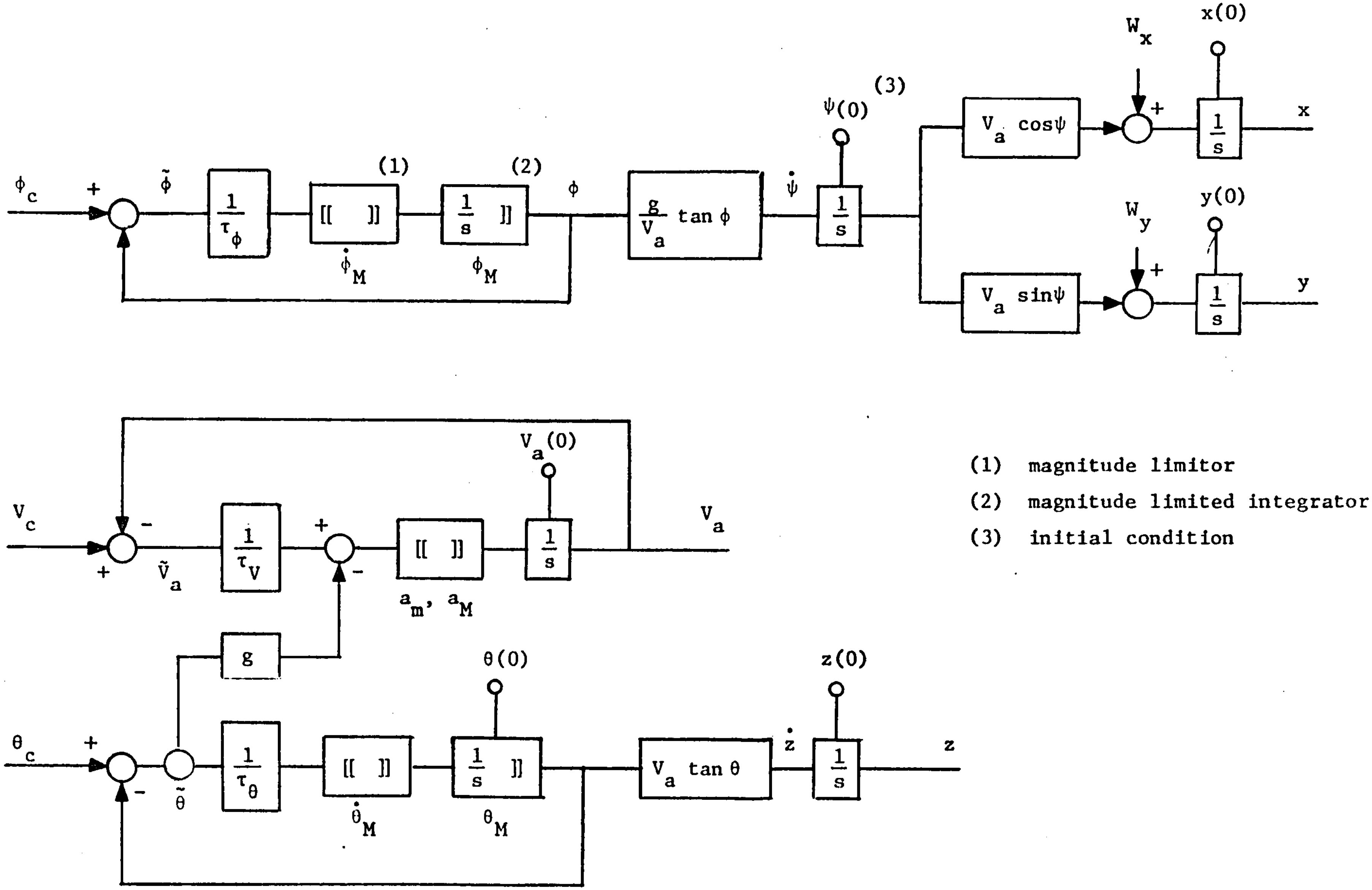


Figure 26. Simplified Command-Response Based Point Mass Aircraft Dynamic Model

Table 27. Dynamic Equations for Point Mass Aircraft Dynamic Model (7-State)

$s\phi = [[\frac{1}{\tau_\phi} (\phi_c - [[\phi]]_{\phi_M})]]_{\dot\phi_M}$		(roll dynamics)
$s\psi = \frac{g}{V_a} \tan\phi$		(heading)
$sx = V_a \cos\psi + W_x$		(kinematics)
$sy = V_a \sin\psi + W_y$		
$s\theta = [[\frac{1}{\tau_\theta} (\theta_c - [[\theta]]_{\dot\theta_M})]]_{\dot\theta_M}$		(pitch or flight path angle)
$sz = V_a \tan\theta$		(altitude)
$s\,V_a = [[\frac{1}{\tau_V} (V_{a,c} - V_a) + g(\theta - \theta_c)]]_{\dot V_{a,m}}^{\dot V_{a,M}}$		(airspeed)
$\begin{bmatrix} W_x \\ W_y \end{bmatrix} = W \begin{bmatrix} \cos\sigma \\ \sin\sigma \end{bmatrix}$		

g = gravitational constant = 32.17 ft/sec^2

$\tau_\phi, \tau_\theta, \tau_V$ = "equivalent" first order time constants

$[[\alpha]]_a^b$ means α is limited between a and b.

$s(\ )$ = Laplace operator = $\frac{d}{dt}(\ )$

Table 28. Traffic Initialization

Own Aircraft:	
$x = y = 0$	(origin)
$z_0 = -6000$ ft	
$\psi = 0.0$	(due north)
$V_a = 200$ kt	
$\theta = 0$	(level flight)
Target Aircraft: (39 aircraft within 60 nmi)	
$\phi = 0$	
$\theta \; \varepsilon \; U\,[-20, 20$ deg$]$	
$\psi \; \varepsilon \; U\,[-180, 180$ deg$]$	
$V_a \; \varepsilon \; U\,[100, 200$ kt$]$	
rng $\varepsilon \; U\,[2, 60$ nmi$]$	
brg $\varepsilon \; U\,[-180, 180$ deg$]$	
$\Delta z \; \varepsilon \; U\,[-3000, 3000$ ft$]$	
$x = \text{rng} \cdot \cos(\text{brg})$	
$y = \text{rng} \cdot \sin(\text{brg})$	
$z = z_0 + \Delta z$	; U [a,b] = Uniformly distributed random number between a and b
Commands initialization	
$\phi_c = 0$	
$\theta_c = \Theta$	
$V_{ac} = V_a$	

adopted so that all the targets are targets of opportunity. Therefore, no specific traffic can be catered for a specific purpose to "control" the experimental outcome. This method may not encompass all the important cases; however, omission is not by intention, and the somewhat large volume of traffic (39 aircraft) should provide a certain protection.

Commands are generated in an open loop manner by a timed table look-up procedure. Thus, they are piecewise constant time functions. The command tables are chosen arbitrarily within reasonable operational limits. One shortcoming of this method is that the chosen traffic patterns do not simulate specific air traffic control patterns or scenarios. Therefore, it would not be efficient if the desired outcome depends on the scenario such as in the CAS applications. For testing and validating the traffic sensor module, the above method should be adequate.

Own state variables and target kinematic variables are transferred to the BXTCAS subroutine by means of two input commons, /OWNINP/ and /TGTINP/ by the XNSFR subroutine. These are given by Table 16.

The validation of the BXTCAS module was performed in two stages - the sensor and CAS logic portions. The sensor validation results are discussed next followed by the CAS logic section.

BX TCAS Sensor Validation

During the course of the simulation development, numerous cases were run. Several cases are chosen and the simulation results are discussed. For the sensor validation purposes, the initialization of the aircraft state variables are as discussed in the previous section. Different cases are obtained by choosing different command tables. The four validation cases are listed below.

Case (1) - 120 sec run

Own: straight and level flight due north at 200 kt and 6000 ft (no maneuvers).

Others: constant horizontal and vertical flight path angle flight; position and velocity are initialized from uniform density functions.

Case (2) - 120 sec run

Own: straight and level flight due north at 200 kt and 6000 ft altitude.

Others: Two aircraft (648 and 840) are commanded the following roll and pitch maneuvers:

A/C 648 - at t = 15, 60, and 105 sec, ϕ_c = 15, -15, and 0 deg;
at t = 15, 75, and 90 sec, θ_c = 7, 0, and -10 deg.

A/C 840 - at t = 15, 60, and 105 sec, ϕ_c = -15, 15, and 0 deg;
at t = 15, 75, and 90 sec, θ_c = 7, 0, and -10 deg.

Case (3) - 180 sec run

Own: straight and level flight due north at 200 kt and 6000 ft altitude; at times 15, 45 and 150 sec, Own makes roll maneuvers of 20, 0, and - 20 deg, respectively.

Others: constant horizontal and vertical flight path angle flight; position and velocity are initialized from uniform density functions.

Case (4) - 120 sec run

The high frequency measurement errors are introduced. The error magnitudes are:

σ_r = 75 ft (range); and
σ_β = 1.5 deg (bearing).

<u>Case</u> (1): Figures 27 (a) through 27 (f) show the simulation results. These are explained and discussed.

Figure 27 (a) shows the horizontal projections of all the tracks within 25 nmi of Own. True, measured and estimated positions are shown respectively from left to right. The target tracks are relative, i.e., Own aircraft is at the origin throughout the simulation period. There are 17 tracks within 25 nmi of Own. The true positions are shown every second. The measured positions are shown according to the sampling periods computed by the scheduling algorithm. The estimated positions are shown corresponding to the measurement times. The latter two figures show the effects of the variable sampling periods and of the different (track or acquision) regions. These will become clear when the individual time plots are discussed.

Figure 27 (b) shows time lines of events associated with individual aircraft. For example, Aircraft No. 840 begins in the acquisition region; at time t = 40.5 sec it transitions into the track region and stays in the region for 60 sec; at t = 104.5 sec it exits back to the aquisition region. The time line shows the dynamic nature of the internal traffic file updates. Several aircraft are deleted from the file as they fly through the acquisition region; and a few individual aircraft are added to the traffic file as they fly into the acquisition region.

Figures 27 (c) through (f) show the time, plots of true, measured and estimate position and velocity components, as well as estimation errors for aircraft in the track region. Figure 27 (c) shows the aircraft No. 144. The true, measured and estimated x and y positions are shown in the upper left plots. The true and estimated x and y velocities are shown in the lower left plots. The position estimation errors are shown in the upper right plots, and the velocity errors are shown in the lower right plots. The velocity errors show a small initial transient behavior, but remains small. The position estimation error plots show a cyclic behavior; zero, small and larger; zero, small, and larger; and so forth. This indicates that the surveillance schedule for this aircraft is every 3 sec.

Figure 27 (d) shows similar results for aircraft No. 288. This aircraft stayed in the track region approximately 20 sec and was apparently sampled every 1 sec. At approximately t = 20 sec, there was a surveillance failure. The aircraft exited to the acquisition region because of the altitude threshold rather than the range threshold.

Figure 27 (f) shows a case where aircraft No. 840 begins in the acquisition region, flies through the track region, and exists to the acquisition region. During the track region, the aircraft was sampled at 2 sec intervals. The transient behaviors are similar to other cases.

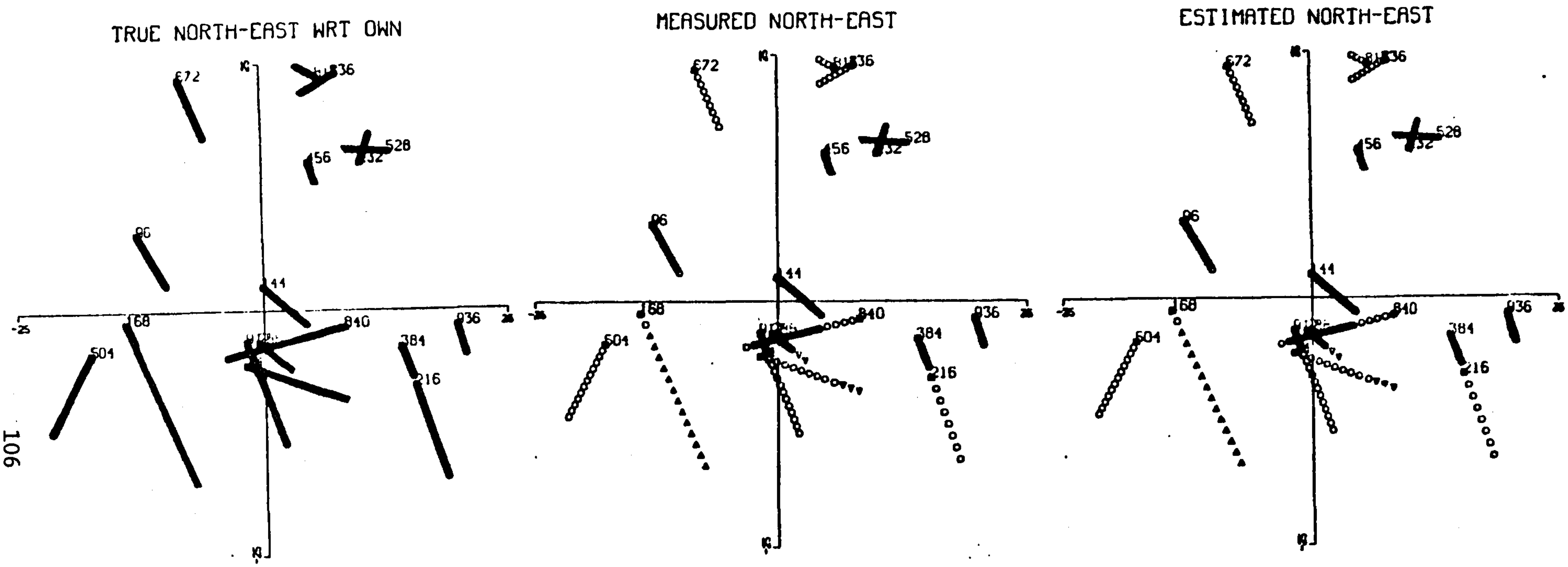


Figure 27(a). Horizontal Projection for Case (1)

A/C ID No. \ Time Epoch	0.5	1.5	20.5	22.5	32.5	40.5	53.5	64.5	70.5	90.5	100.5	104.5	105.5	120.5
24	A										O			
96	A													
144	T													
168	A													
216	A							O						
288	T			A		O								
384	A													
432	|													
456	|													
528	|									O				End Sim. Run
816	|				O									
840	A					T						A		
912	T		O											
936	A													
504		A											O	
672							A							
336									A					

T → Track Region

A → Acquisition Region

O → Outside of Coverage

Figure 27(b). Time Line for Case (1)

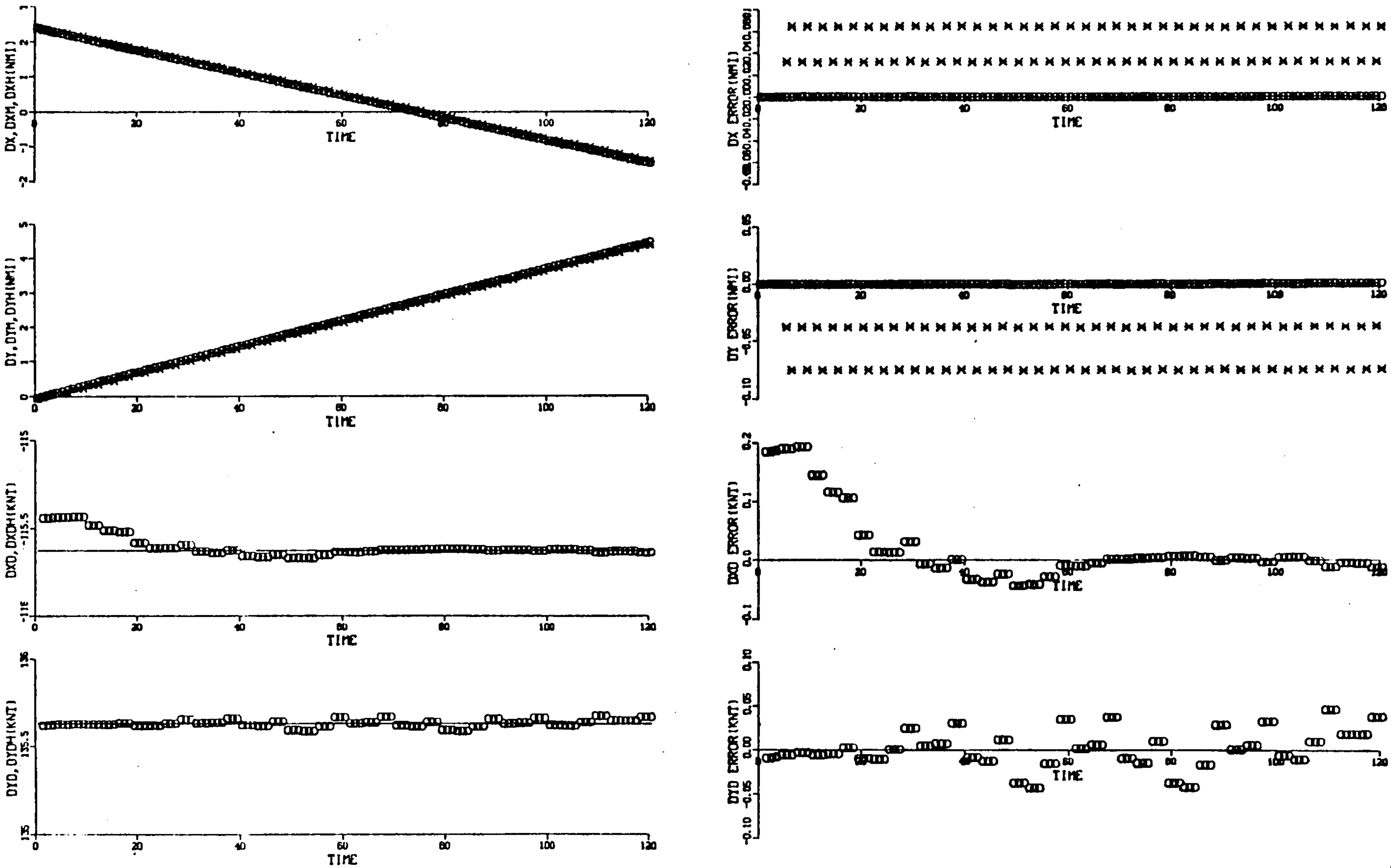


Figure 27(c). Relative Position, Velocity and Errors for A/C 144 - Case (1)

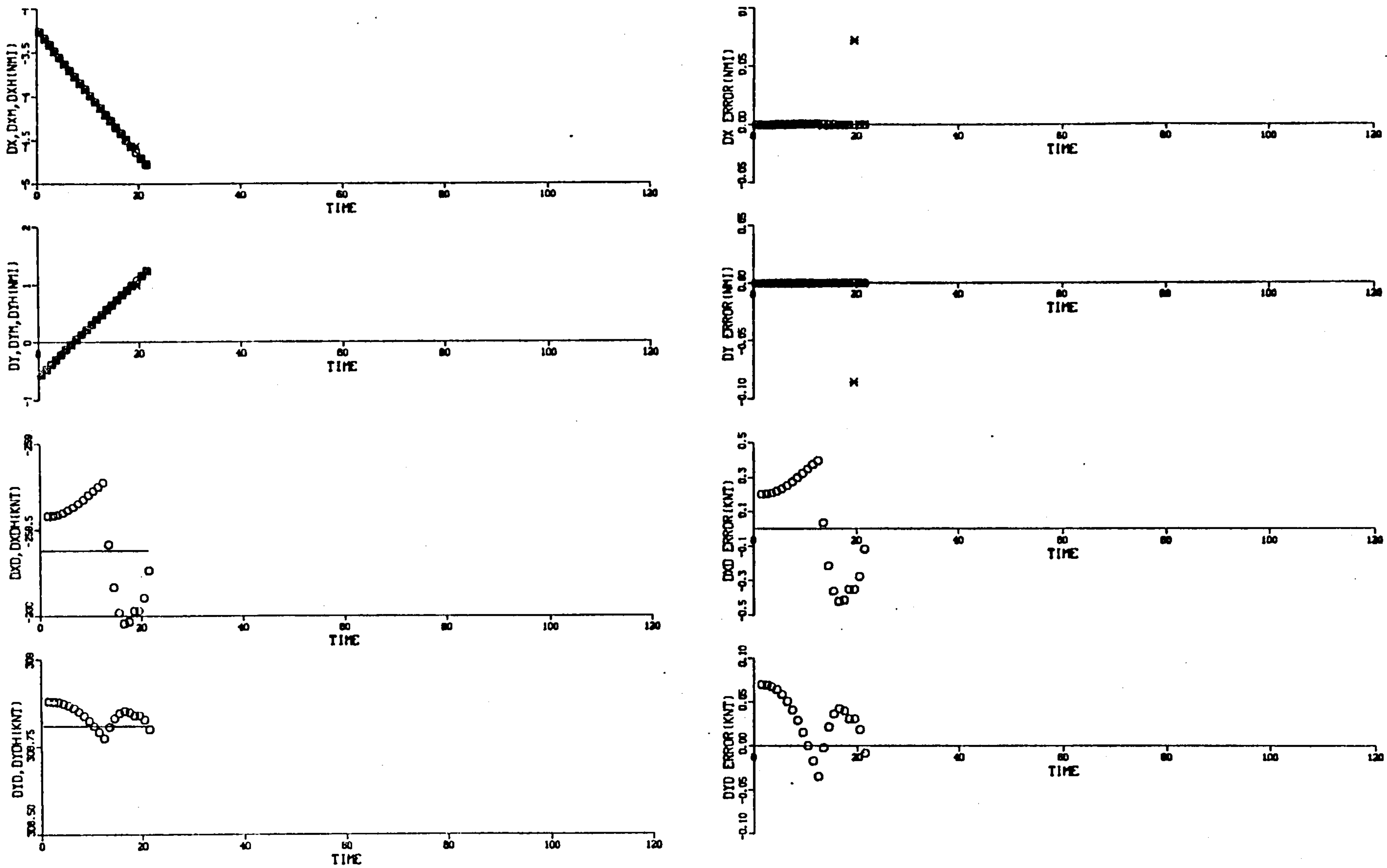


Figure 27(d). Relative Position, Velocity and Errors for A/C 288 - Case (1)

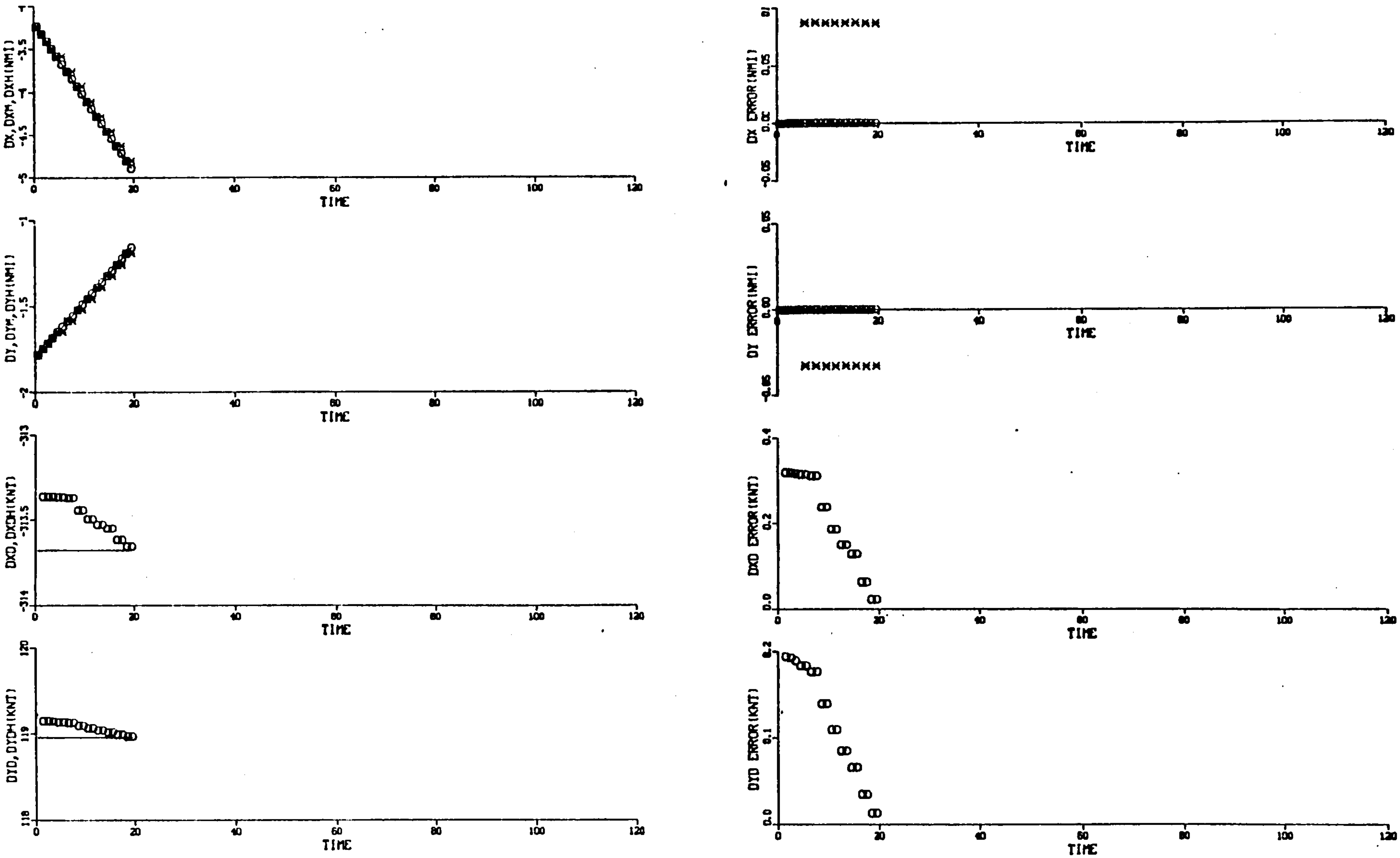


Figure 27(e). Relative Position, Velocity and Errors for A/C 912 - Case (1)

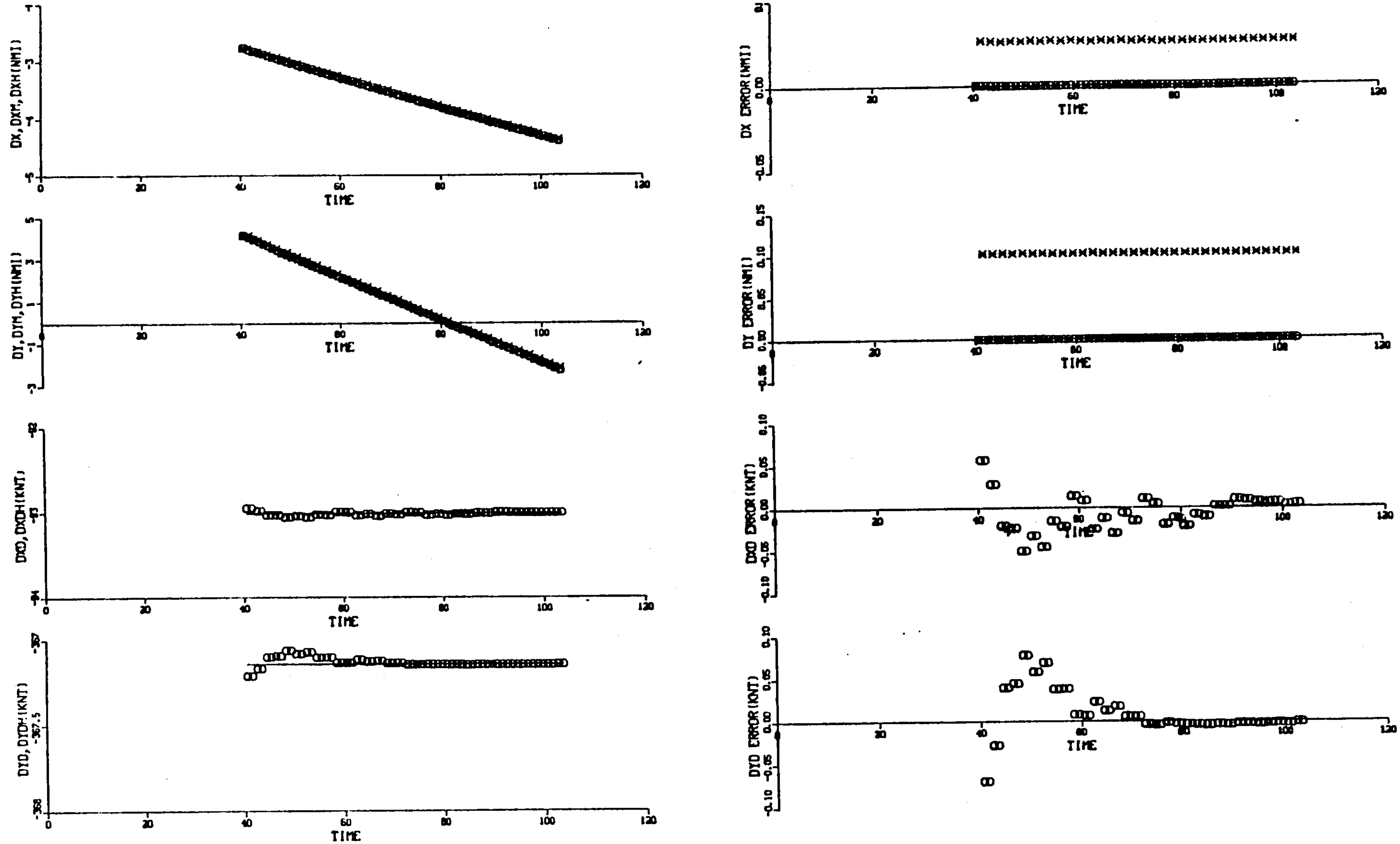


Figure 27(f). Relative Position, Velocity and Errors for A/C 840 - Case (1)

Case (2): Figures 28 (a) through 28(g) show the results for Case (2). Here, two aircraft (Nos. 840 and 648) underwent roll maneuvers. The horizontal projections show these two maneuvering aircraft very clearly.

Figure 28 (d) shows the time plots for aircraft No. 840. They show interesting points: (1) surveillance failed for four consecutive seconds at apporximately t = 20 sec; (2) the velocity errors during the maneuver are large ($\hat{\dot{\Delta}}_x$ of 20 kt and $\hat{\dot{\Delta}}_y$ of -18 kt) and the position errors are on the order of 0.01 nmi, and (3) toward the end of track period (t = 60 sec), the sample period becomes 2 sec from 1 sec due to altered geometry.

Figure 28(e) shows the time plots for aircraft No. 648 with similar characteristics. Here, the maximum velocity errors are 45 kt for $\dot{x}$ and 13 kt for $\dot{y}$. Because of the changing geometry, the surveillance rate drops to a 2 sec interval at approximately t = 25 sec. Because of this lower sampling frequency, the velocity errors become larger which induce larger position errors.

Figure 28 (g) shows the time plots for aircraft No. 696. This aircraft flies into the track region at approximately t = 95 sec. The sampling rate is 6 sec. This example shows clearly that the position prediction (COAST) routine is working, since the measurement errors increase (because the oldest measurement is used to compute error along the current position) while the predicted estimates have negligible errors.

Case (3): This is the case when Own aircraft makes two roll maneuvers - + 20 deg at t = 15 sec and -20 deg at t = 150 sec. Figure 29 shows the roll, yaw-rate and yaw angle time plots. It is seen that the essential dynamic characteristics are captured by the simple aircraft model.

Figure 30 (a) through (f) shows the results for this case. Figure 30(a) shows the horizontal projections. The plots show the other aircraft tracks in the NED coordinate system moving with Own as it maneuvers. (It is not a track-up display.) Therefore, as Own maneuvers it looks as though all other traffic is maneuvering.

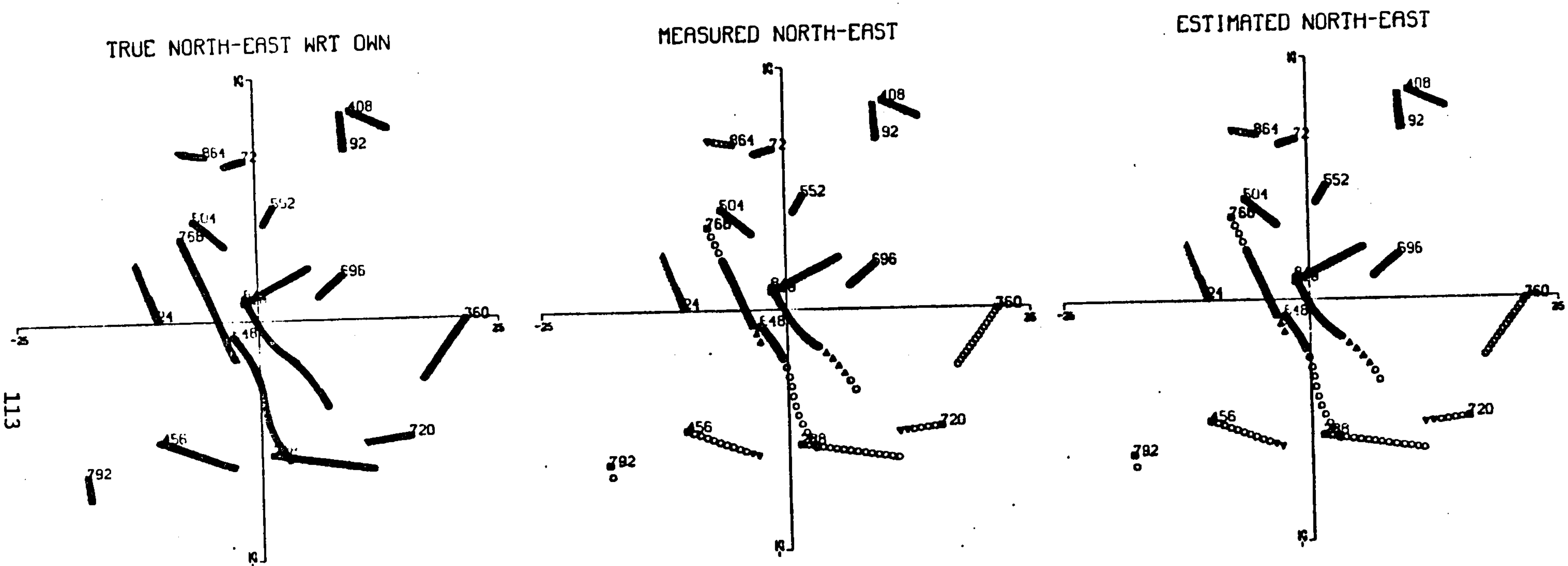


Figure 28(a). Horizontal Projection for Case (2)

Time Epoch / A/C ID No.	0.5	1.5	17.5	32.5	35.5	41.5	52.5	64.5	92.5	96.5	104.5	118.5	120.5
24	A												
72													
192												O	
288													
360													
408													
456									O				
504											T		
552													
696										T			End Sim. Time
720							O						
768	A			T						A			
816	T												
840	T							A					
864	A					O							
648		T			A								
792		A	O										

T → Track Region
A → Acquisition Region
O → Outside of Coverage

Figure 28(b). Time Line for Case (2)

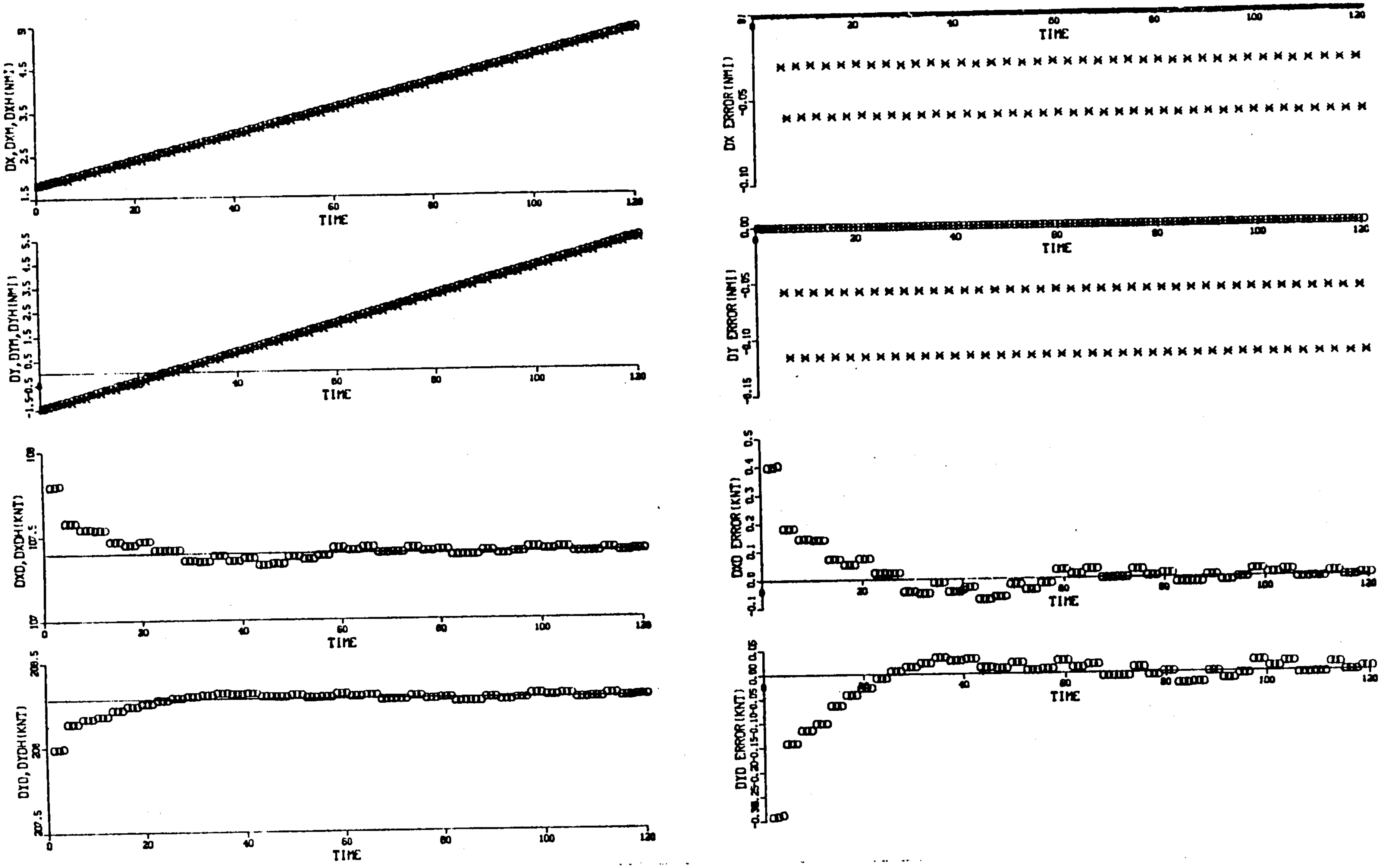


Figure 28(c). Relative Position, Velocity and Errors for A/C 816 - Case (2)

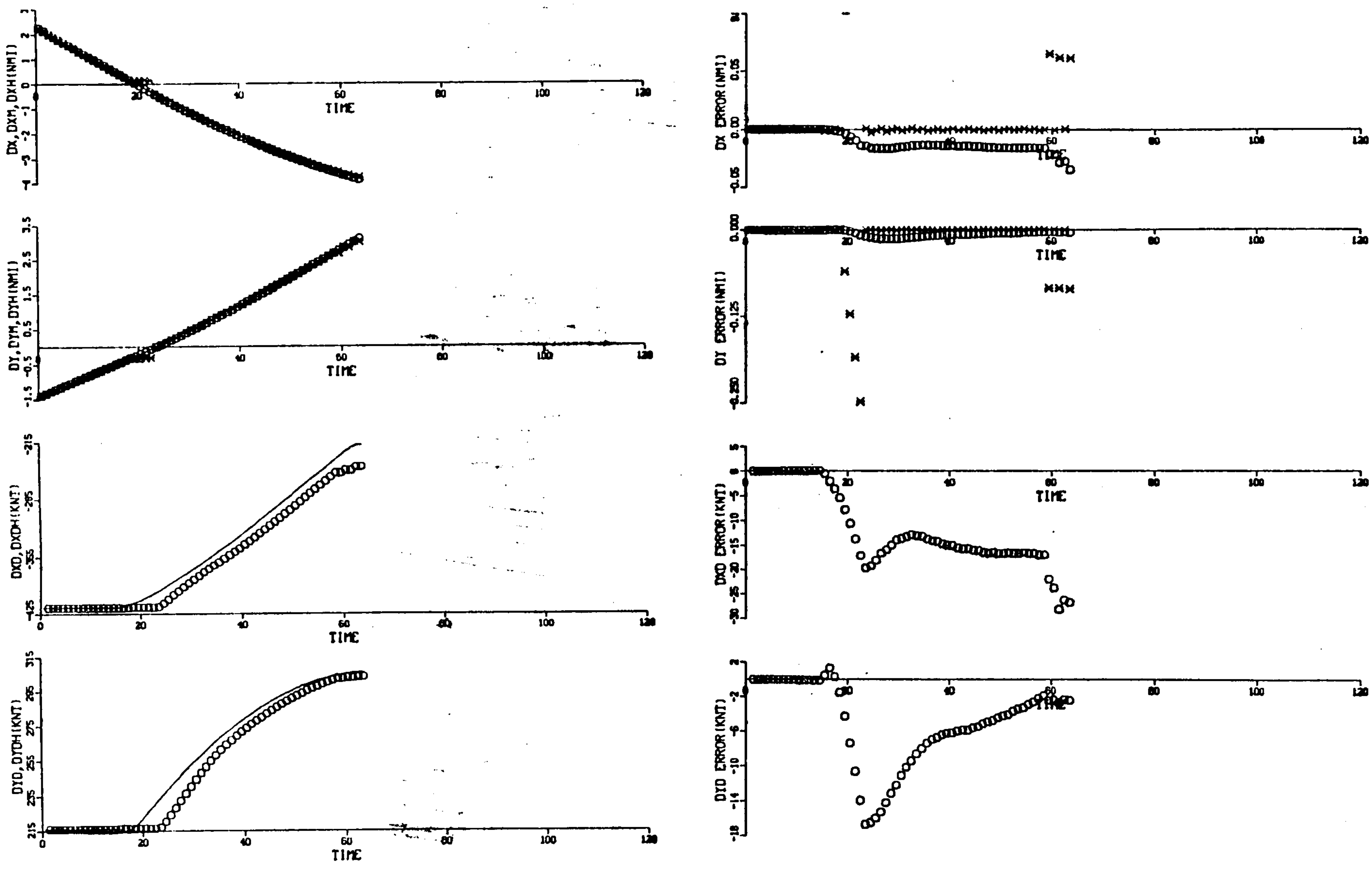


Figure 28(d). Relative Position, Velocity and Errors for A/C 840 - Case (2)

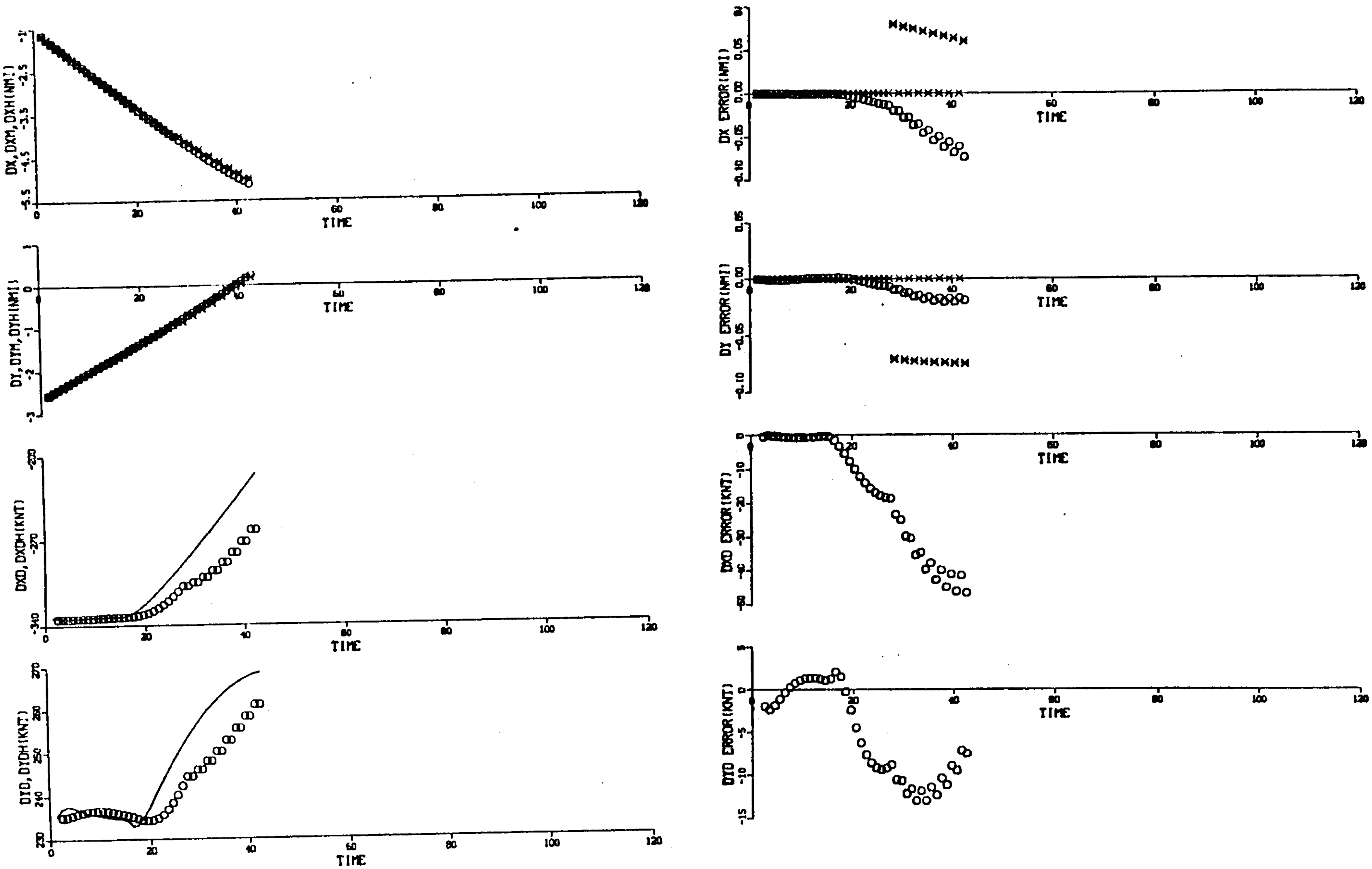


Figure 28(e). Relative Position, Velocity and Errors for A/C 648 - Case (2)

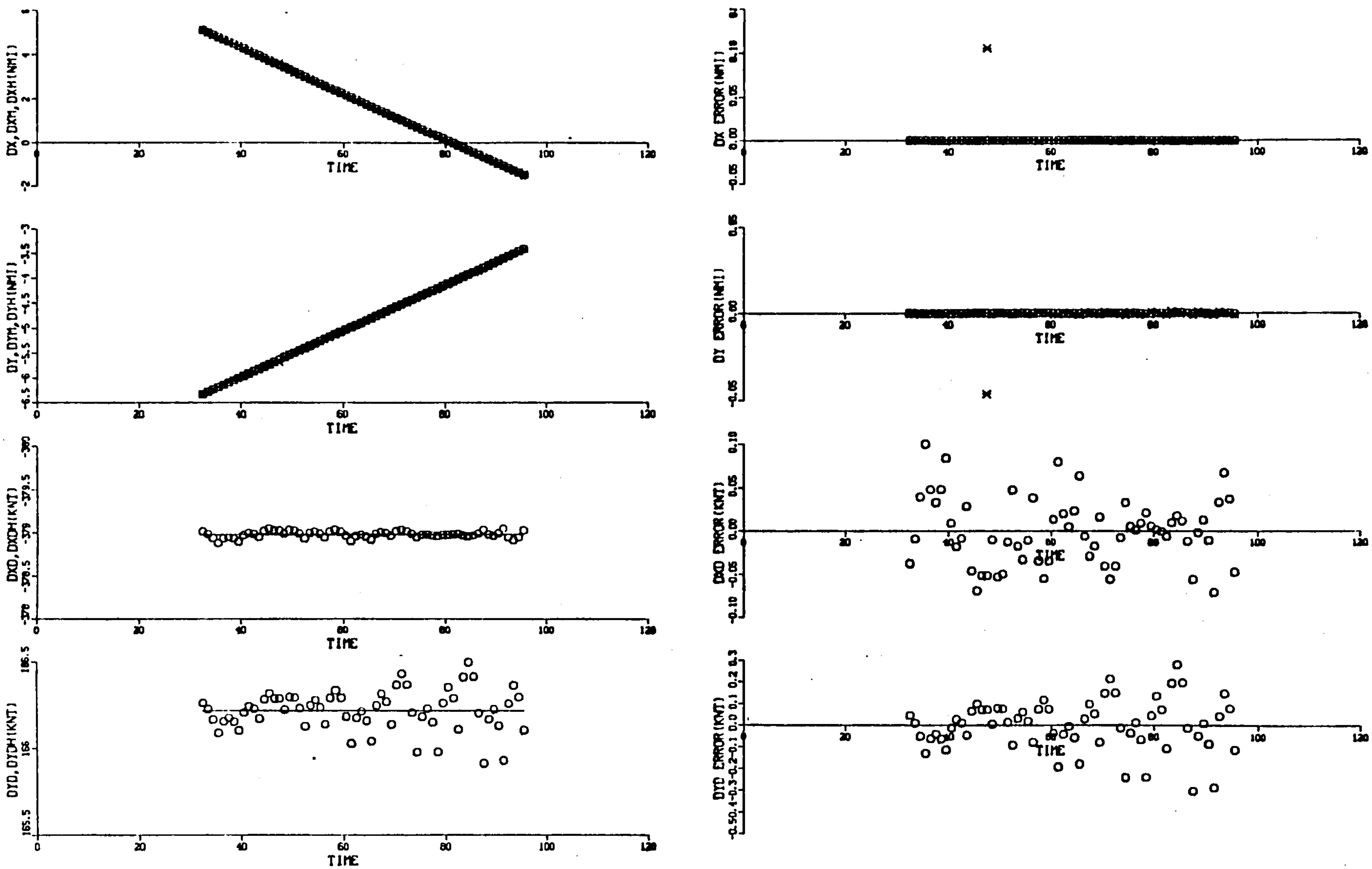


Figure 28(f). Relative Position, Velocity and Errors for A/C 768 - Case (2)

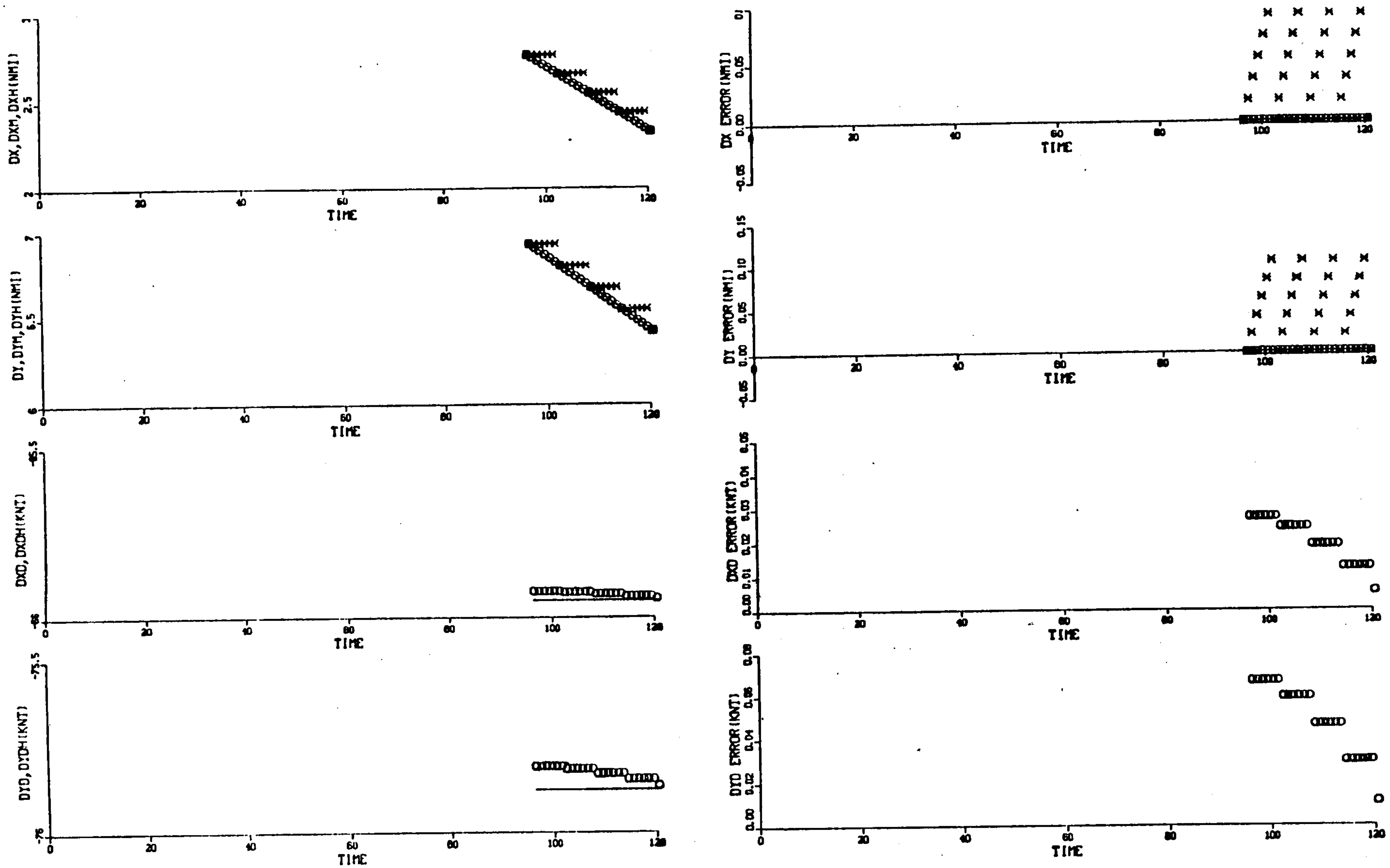


Figure 28(g). Relative Position, Velocity and Errors for A/C 696 - Case (2)

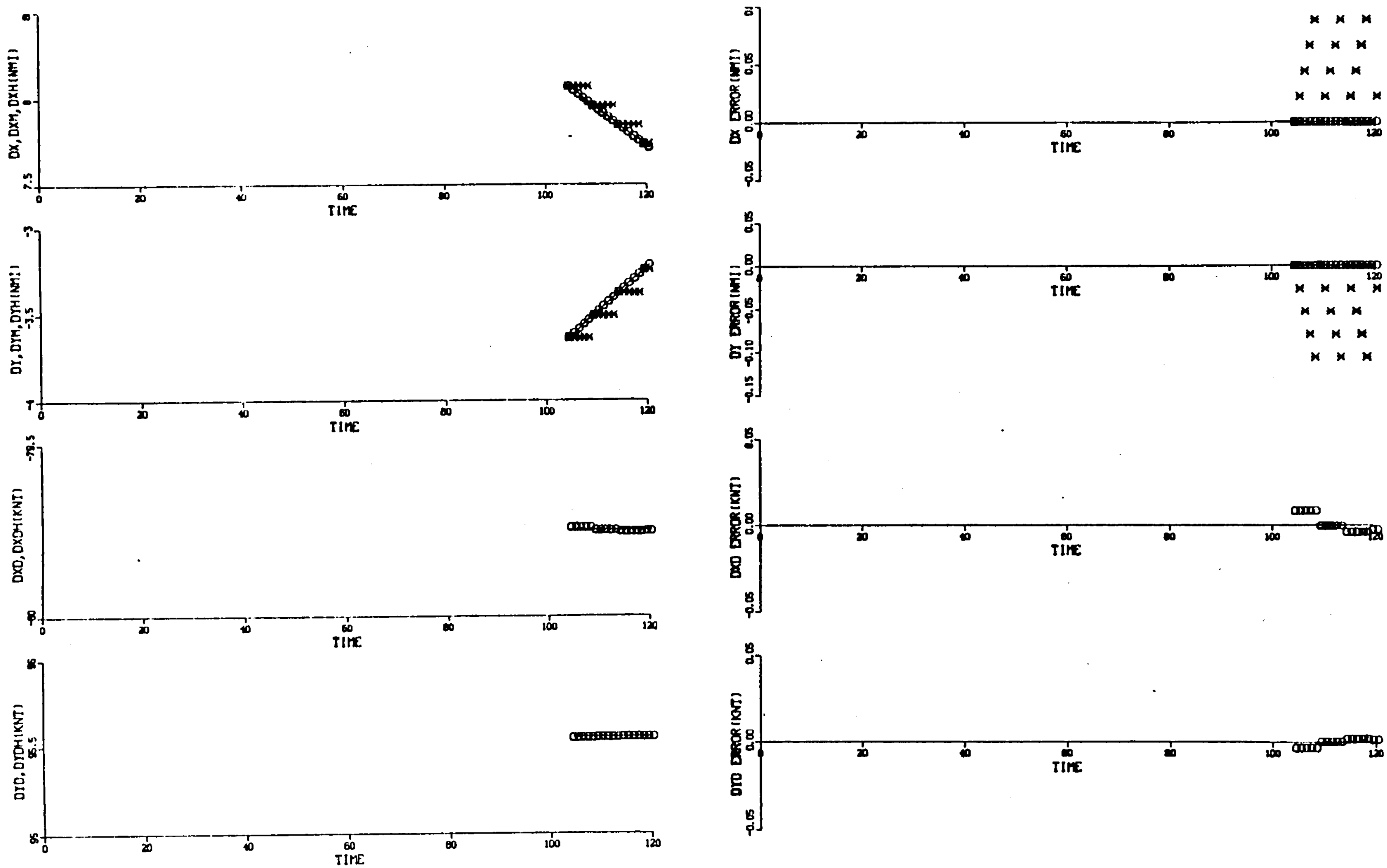


Figure 28(h). Relative Position, Velocity and Errors for A/C 504 - Case (2)

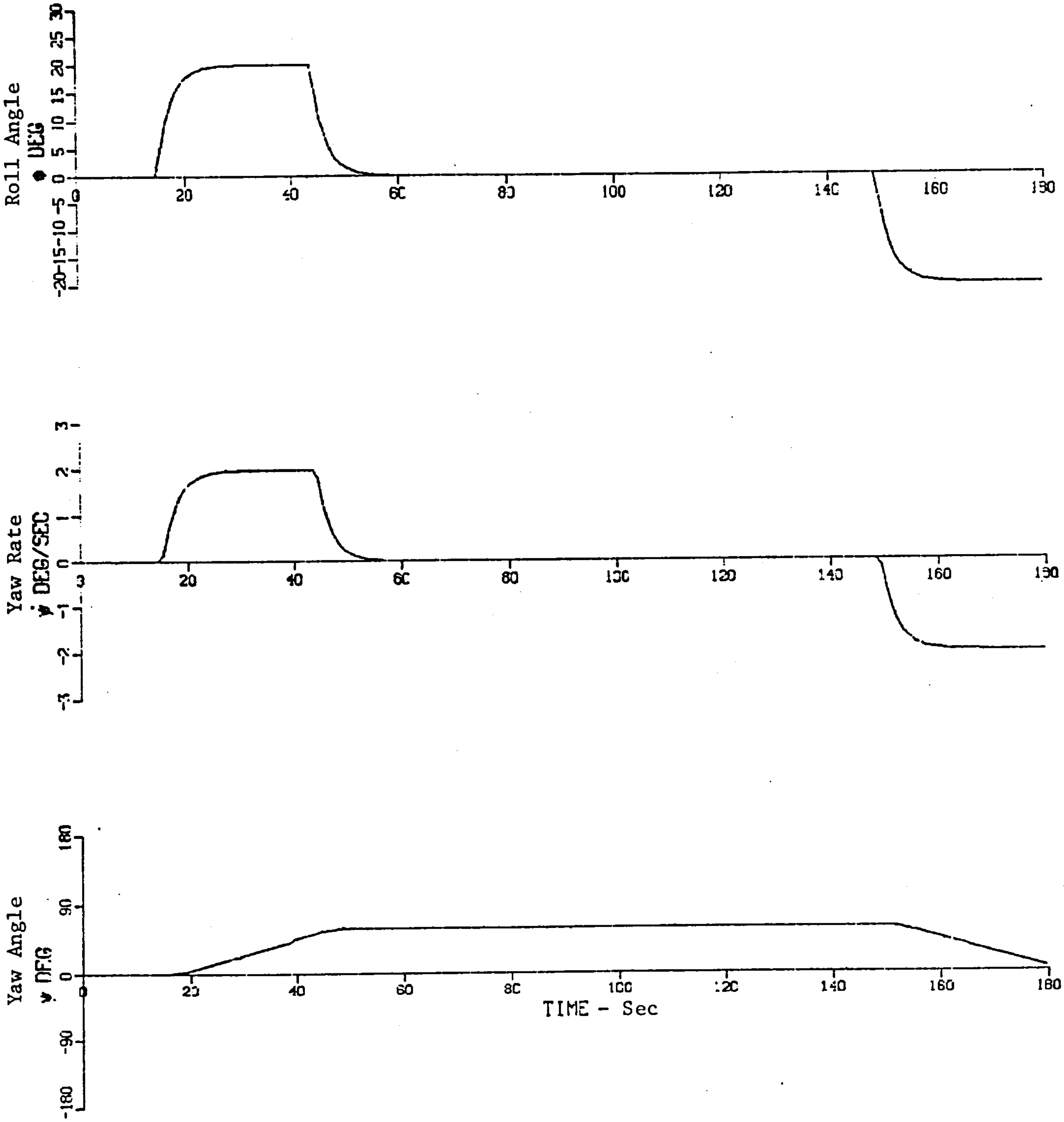


Figure 29. - Own Lateral Dynamic Variables for Case (3)

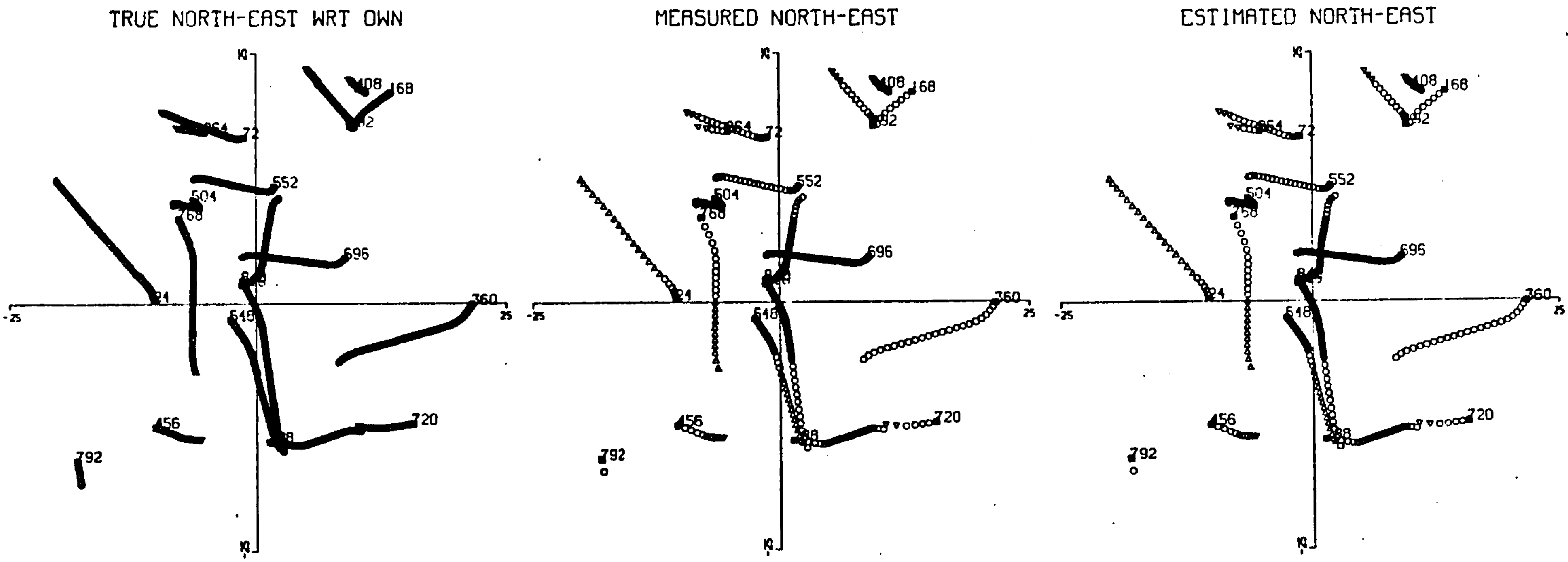


Figure 30 (a) Horizontal Projections for Case (3)

A/C ID No. \ Time Epoch	0.5	1.5	17.5	38.5	40.5	41.5	52.5	79.5	92.5	118.5	130.5	156.5	160.5	180.5
24	A													
72													O	
192										O				
288														
360														
408												O		
456									O					
504														
552														
696					T									
720							O							
768	A													End Sim. Run
816	T										O			
840	T							A						
864	A					O								
648		T		A										
792		A	O											
168										A				

T ⟹ Track Region
A ⟹ Acquisition Region
O ⟹ Outside of Coverage

Figure 30(b) Time Line for Case (3)

Figures 30(c) through 30(f) show the time, measured and estimated position and velocity time plots. Referring to Figure 30(c) (aircraft No. 816), the following comments apply:

(i) This aircraft started out as a pop-up. After the initial 5 sec of continuous surveillance, the sampling period became a regular 3 sec;

(ii) The peak velocity errors were -60 and 90 kt for x and y coordinates respectively. The transient period was x long - approximately 30-40 sec; and

(iii) Because the measurement errors (when they are sampled) are zero, the Own attitude effects through the forward and backward transformations are properly accounted for.

Figure 30(f) shows the time plots for aircraft No. 696. Two interesting observations are: (1) the sampling period changed from 3 to 2 to 3 sec; and (2) the position and velocity errors are symmetric with respect to Own's maneuver. The peak velocity errors were -50 and -80 kt for x and y coordinates respectively. This case clearly points out an ill effect of the dynamic sampling period selection procedure. The sampling period should have been reversed, i.e., it should have been two (or even one) sec during maneuvers and could have been slower during the straight flight period.

Case (4): Figures 31(a) through 31(i) show the simulation results. All the tracks are straight for this case; however, the range and bearing measurements contained additive high frequency errors with the standard deviation magnitudes of 75 ft and 1.5 deg, respectively. Injection of high frequency noise sources does not prove or validate the simulation software per se, once the input-output relationships are proven. However, it will provide a certain operational latitude in terms of the system robustness; that is, the simulation module does not require perfect information.

Figure 31(a) shows the true, measured and estimated horizontal tracks for this scenario. The measurement and estimation plots show quite perceivable noise effect. Considering the somewhat small plot scale of

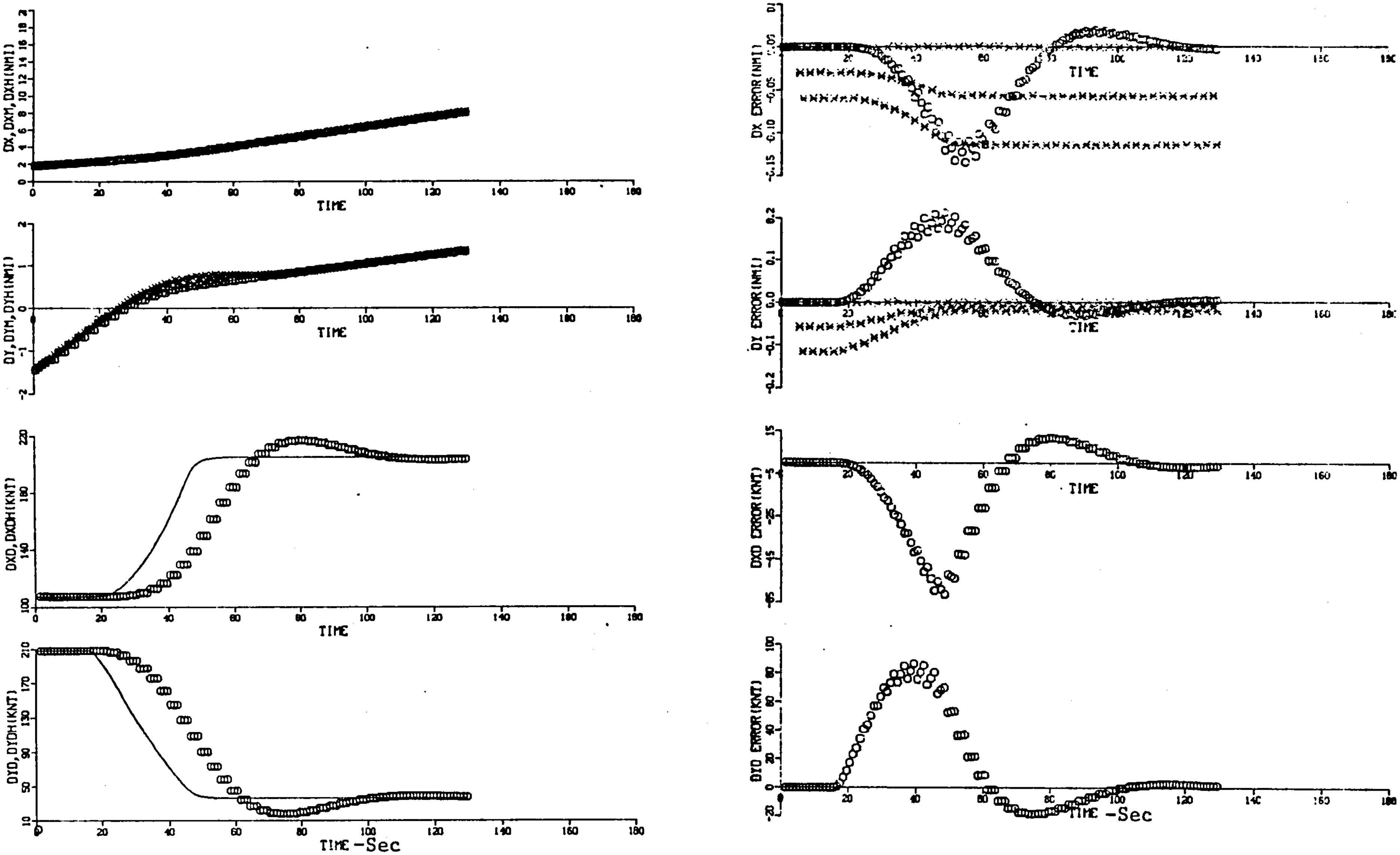


Figure 30(c) Relative Position, Velocity and Errors for A/C 816 - Case (3)

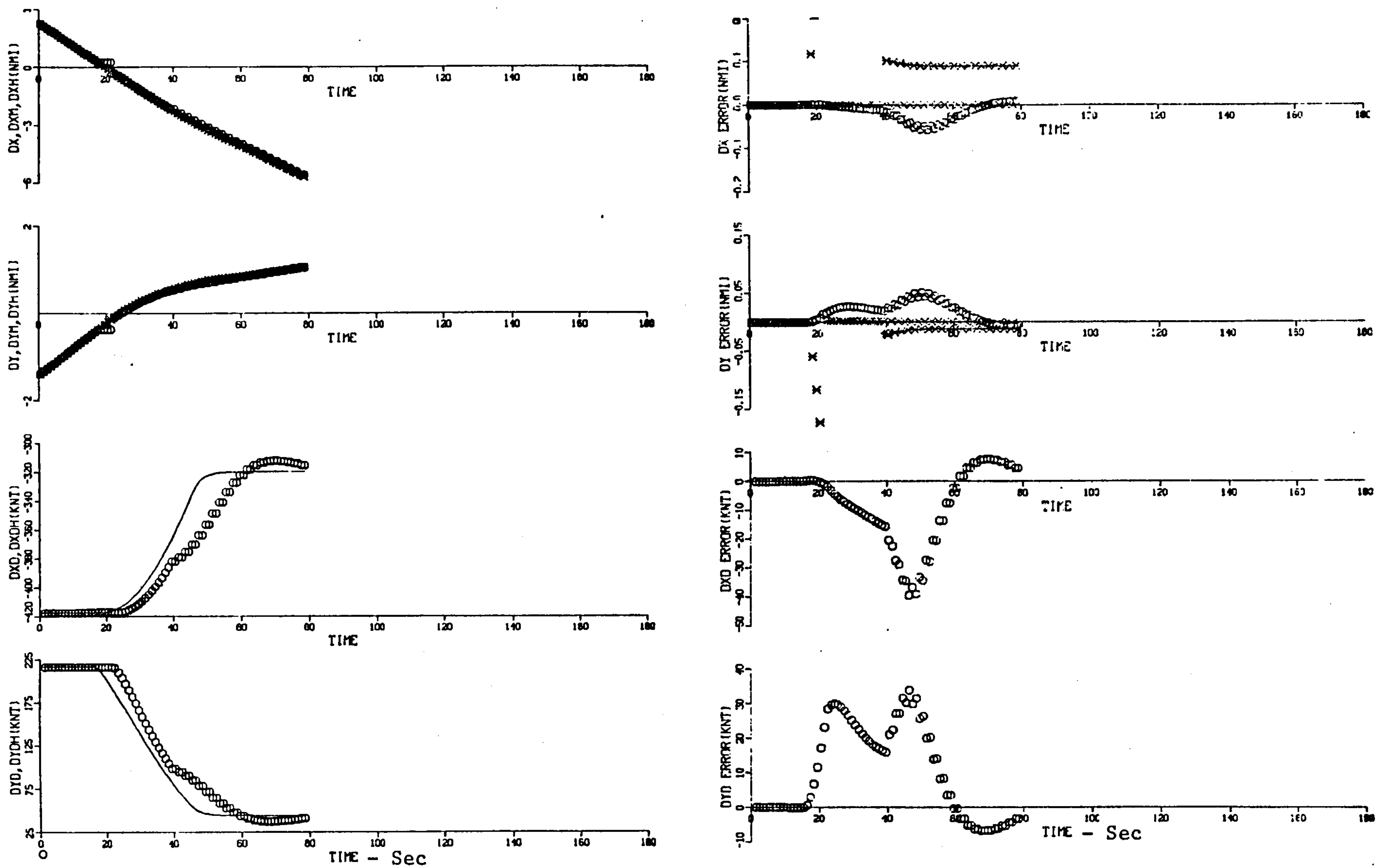


Figure 30 (d) Relative Position, Velocity and Errors for A/C 840 - Case (3)

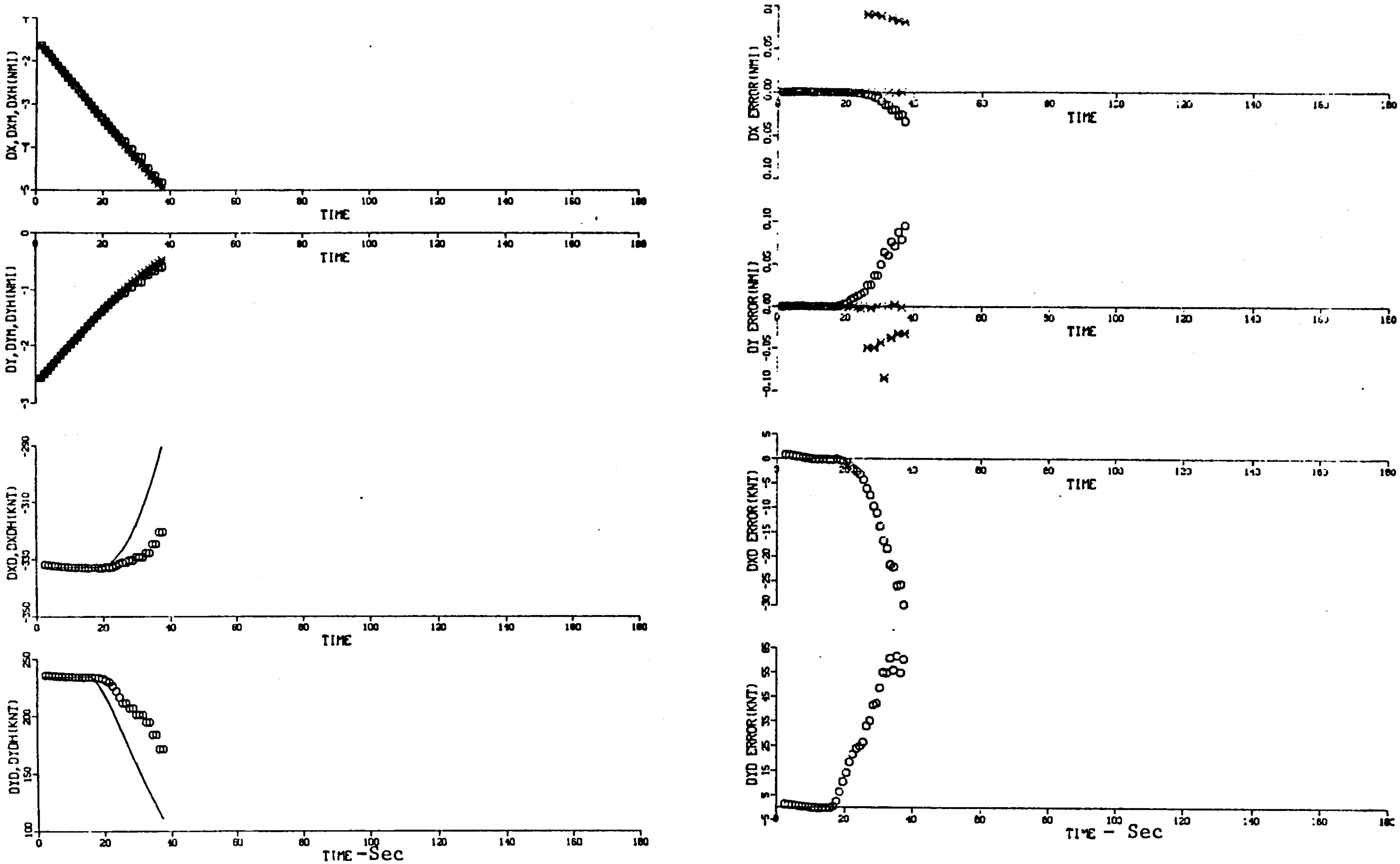


Figure 30(e) Relative Position, Velocity and Errors for A/C 648 - Case (3)

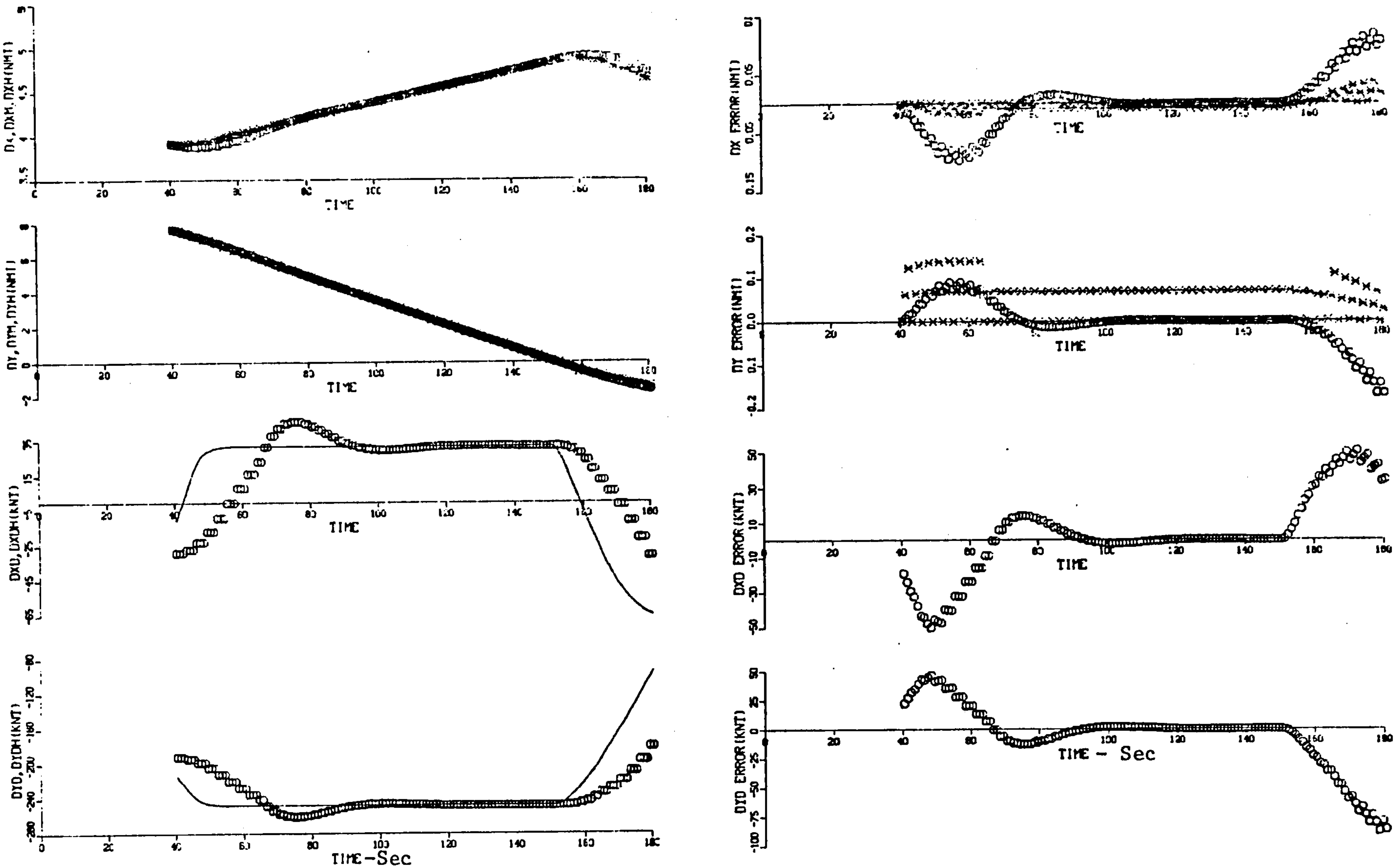


Figure 30 (f) Relative Position, Velocity and Errors for A/C 696 - Case (3)

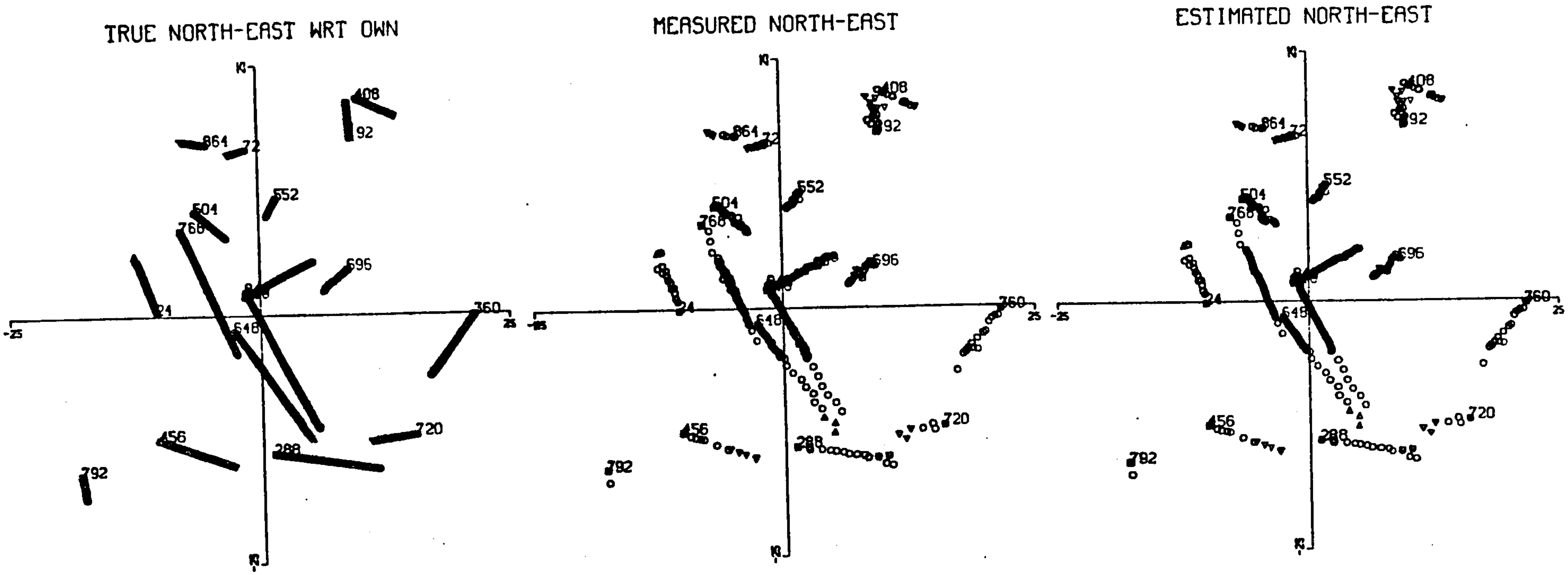


Figure 31(a) Horizontal Projection for Case (4)

A/C ID No. \ Time Epoch	0.5	1.5	17.5	32.5	38.5	41.5	52.5	56.5	63.5	73.5	80.5	84.5	92.5	95.5	96.5	98.5	106.5	114.5	118.5	120.5
24	A																			
72																				
192																			O	
288																				
360																				
408																				
456													O							
504										T						A	T			
552											T			A						
696								T				A								
720							O													End Sim. Time
768	A			T											A					
816	T																	A		
840	T								A											
864	A					O														
648		T			A															
792		A	O																	

T → Track Region
A → Acquisition Region
O → Outside of Coverage

Figure 31(b) Time Line for Case (4)

approximately 12 nmi/inch and the "static display" (the entire track history dots are seen at once), the measurement noise may have substantially worse effects on pilots in the CDTI applications. For example, the display scale may be 1 nmi/inch and each history dot may be displayed dynamically as they become available. This means that the linear dimension would be perceived one order of magnitude larger.

The time line of events for this case - Figure 31(b) - shows one interesting aircraft (No. 504). It begins as an acquisition aircraft until T = 74 sec; then it transitions to the track region. It then transitions to the acquisition regions at T = 98.5 sec; and then it transitions back to the track region at T = 108.5 sec. (Even though not shown, the latter behavior is due to Own's altitude maneuver.)

Figures 31(c) through 31(i) show the time plots for all the aircraft in the track region. All of these show the noise effect. The following comments summarize the result:

(i) Position measurement errors are between 0.2 ~ 0.4 nmi;

(ii) Position estimation errors are generally smaller due to the smoothing effect on the α-β tracker;

(iii) Initial velocity errors are large up to 200 kt; and

(iv) Sampling time variations of 1 or 2 sec were observed.

This "performance" is thoughtto be realistic. The actual system is expected to behave worse than the simulation, since many factors are not incorporated in the simulation program. Two of the important factors are (a) the correlation process (of establishing between the targets and measurements) is 100% reliable with the simulation programs; and (b) the measurement errors are assumed to be white noise sequences, whereas for the actual system, the errors would be more correlated due to multi-path and reflection effects.

During the course of debugging and development effort, each module's inputs and outputs are checked and rechecked to validate the software comparing with "design" requirements. To that extent the simulation software is validated.

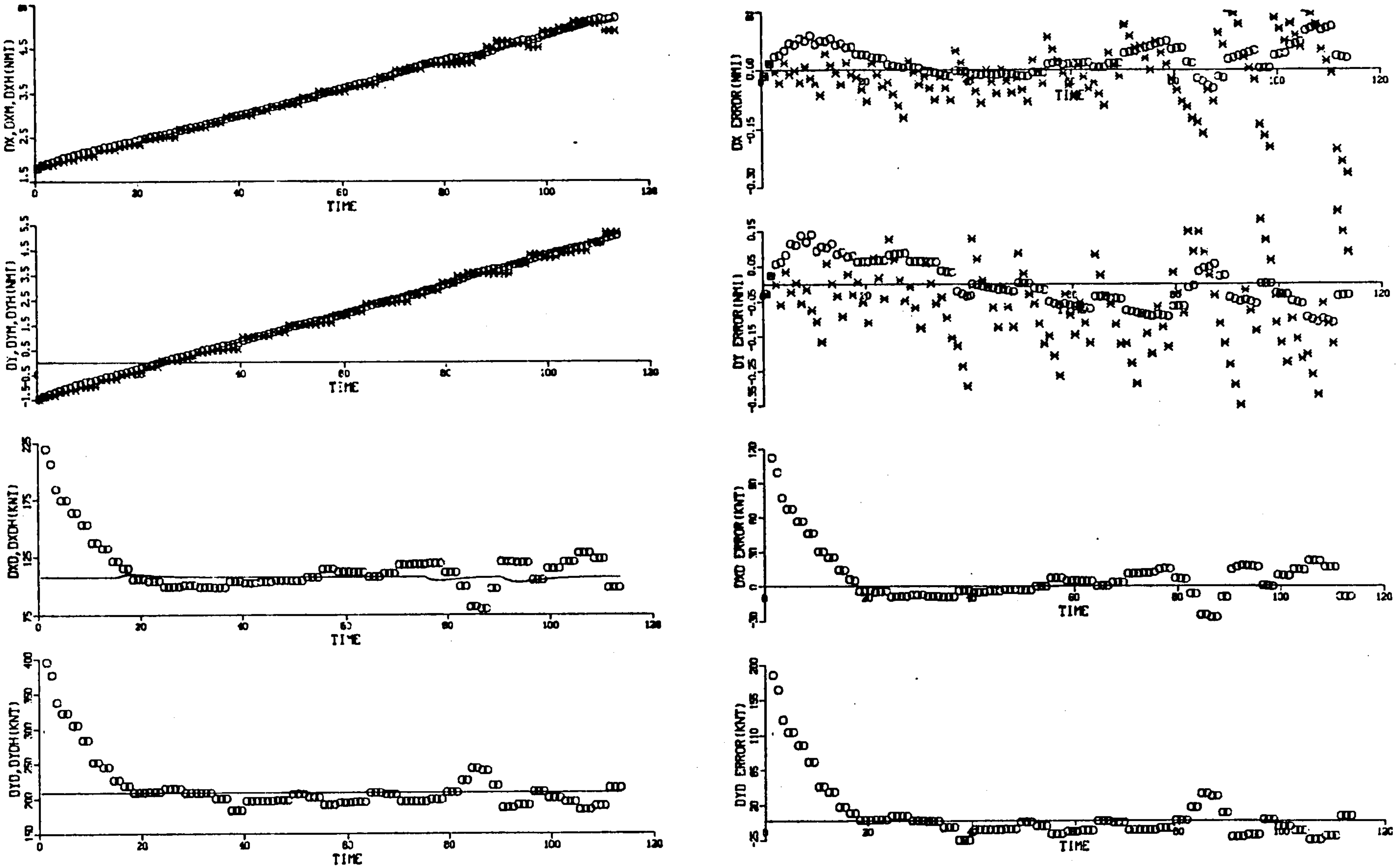


Figure 31(c) Relative Position, Velocity and Errors for A/C 816 - Case (4)

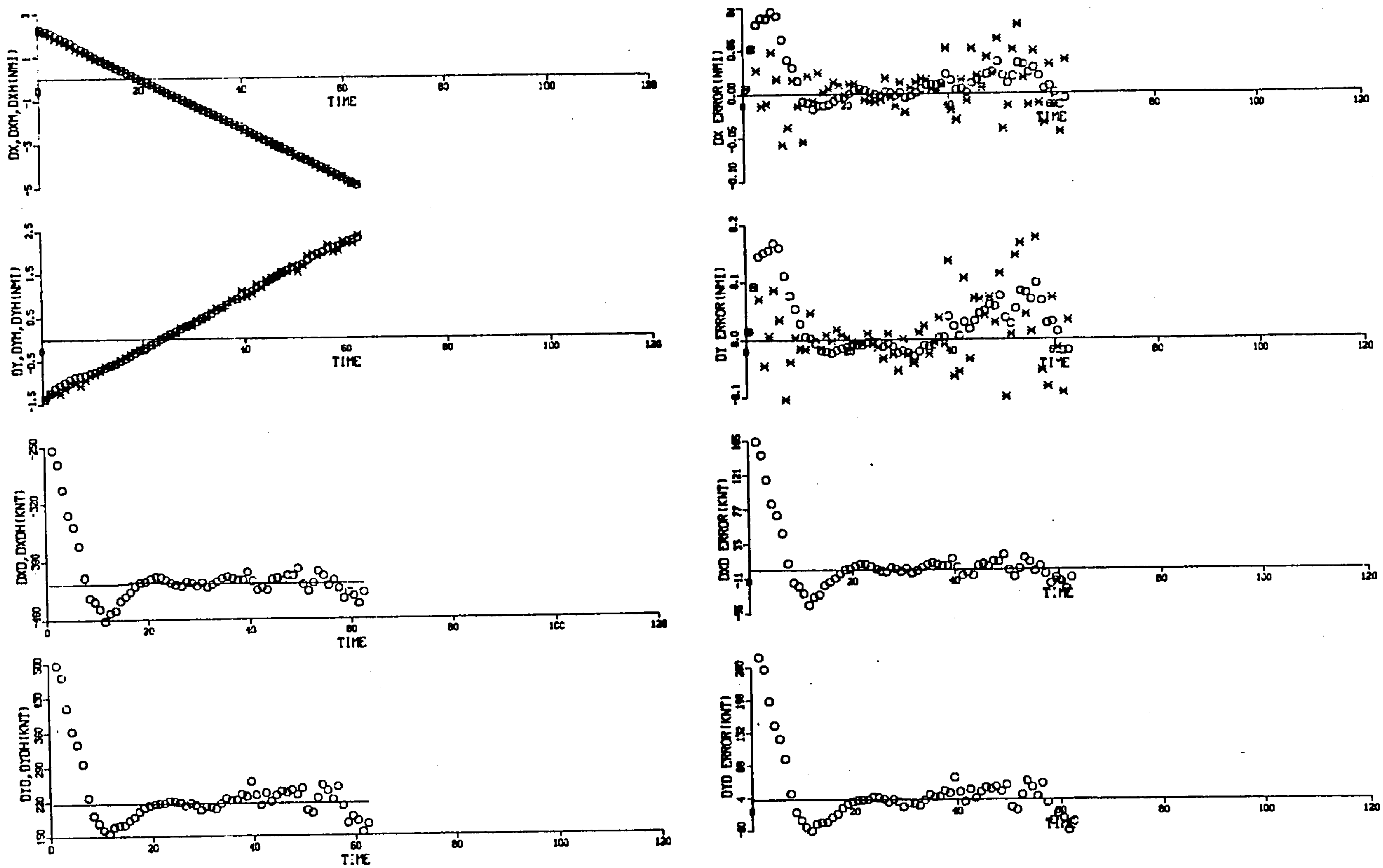


Figure 31(d) Relative Position, Velocity and Errors for A/C 840 - Case (4)

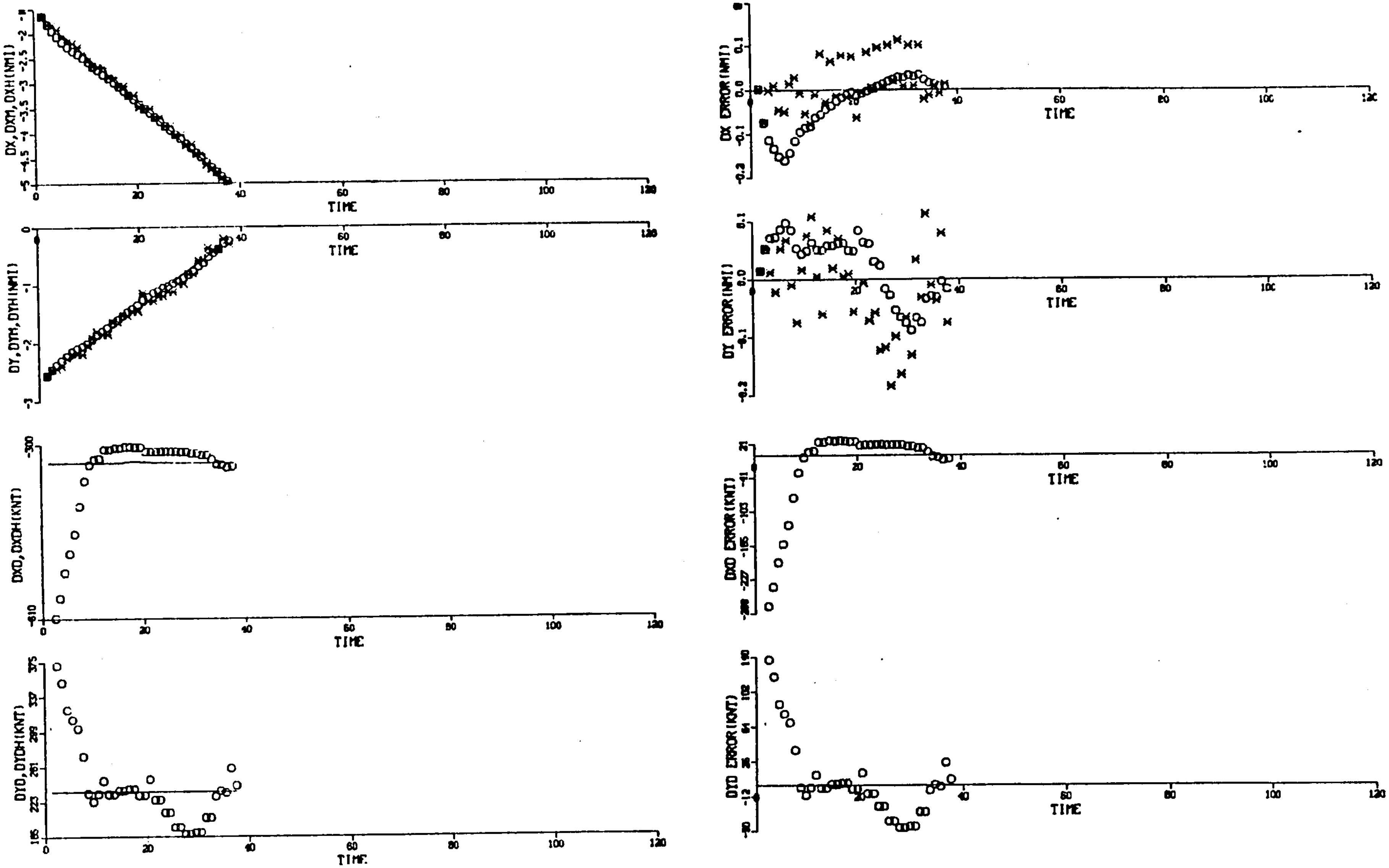


Figure 31 (e) Relative Position, Velocity and Errors for A/C 648 - Case (4)

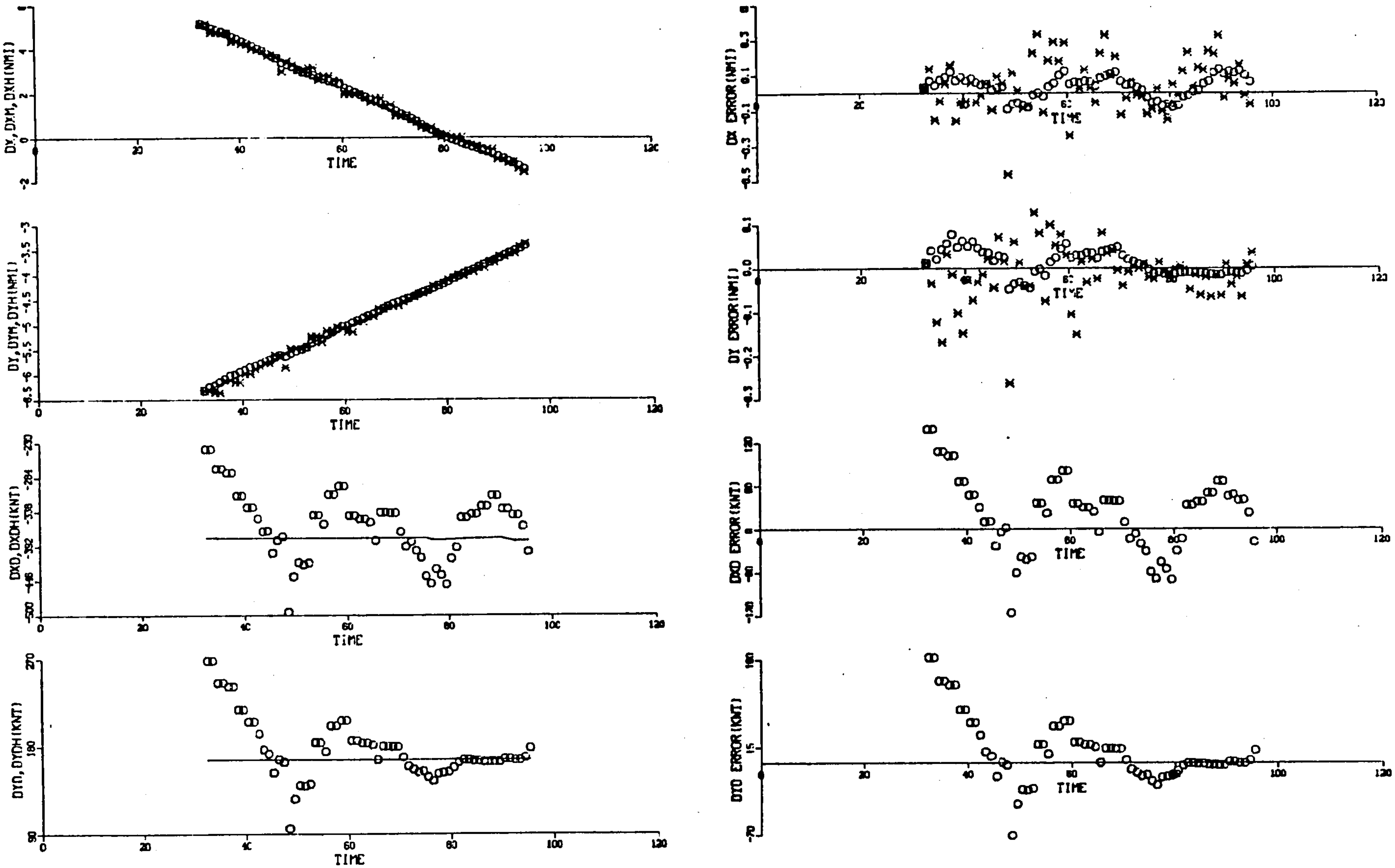


Figure 31(f) Relative Position, Velocity and Errors for A/C 768 - Case (4)

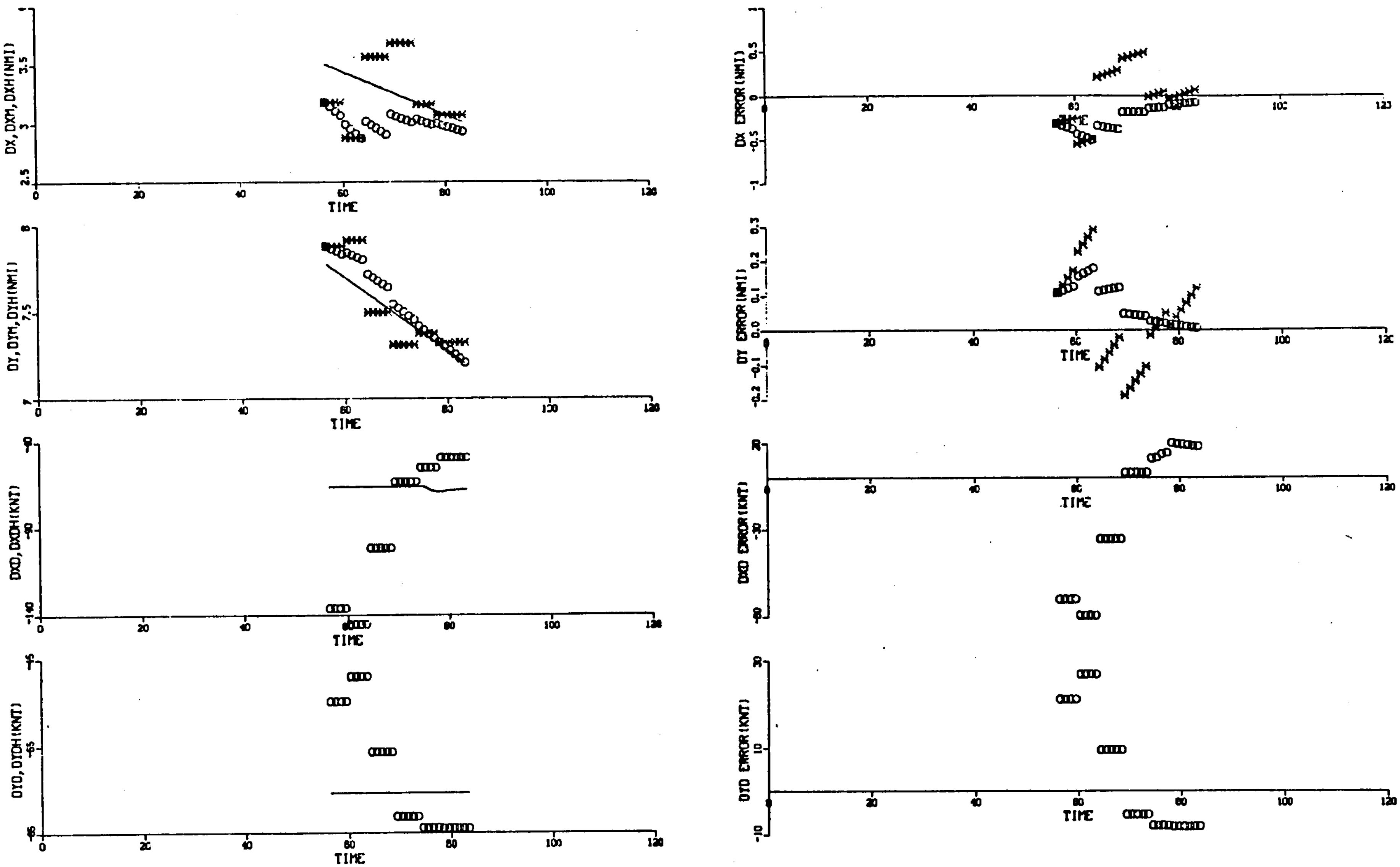


Figure 31(g) Relative Position, Velocity and Errors for A/C 696 - Case (4)

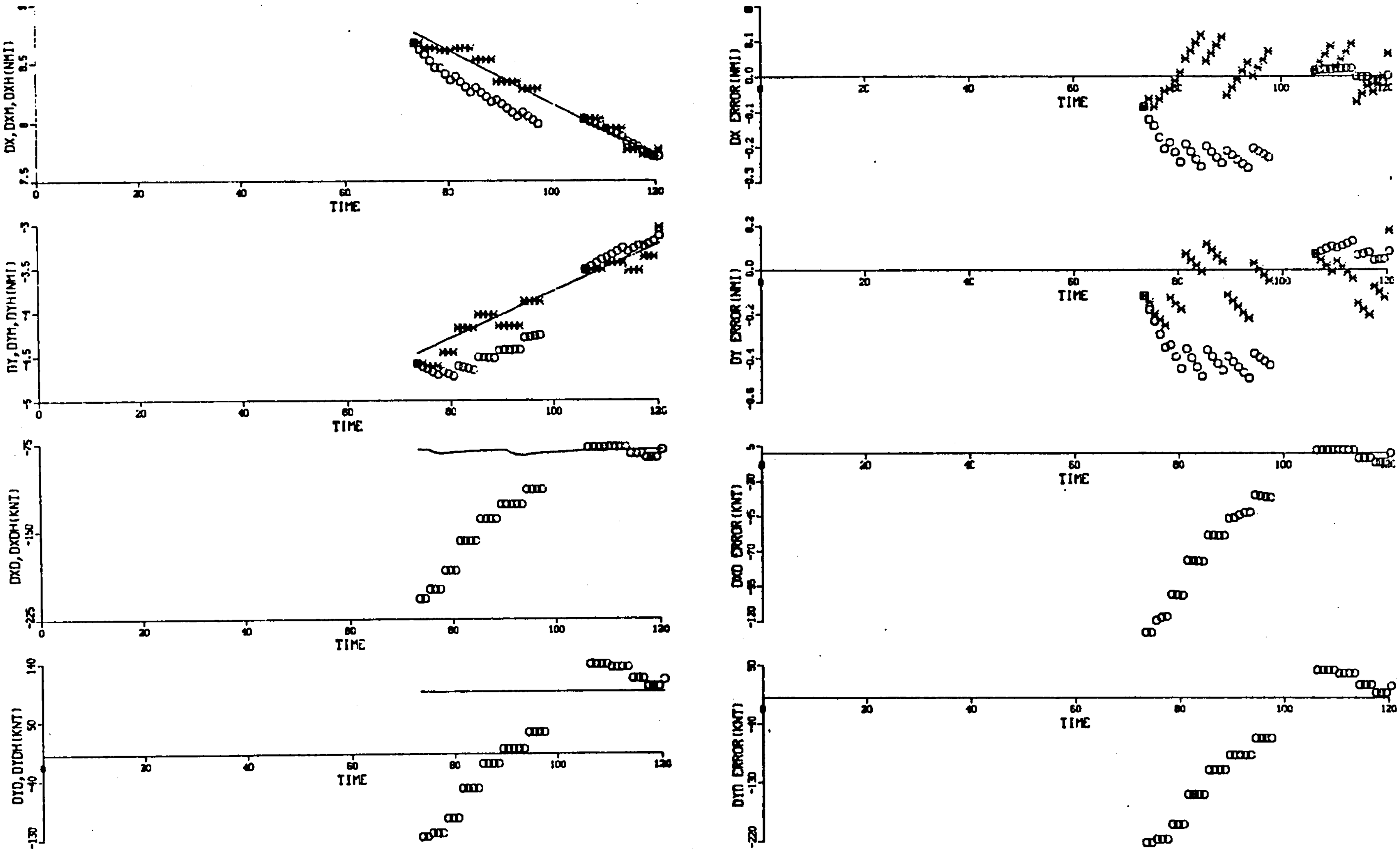


Figure 31(h) Relative Position, Velocity and Errors for A/C 504 - Case (4)

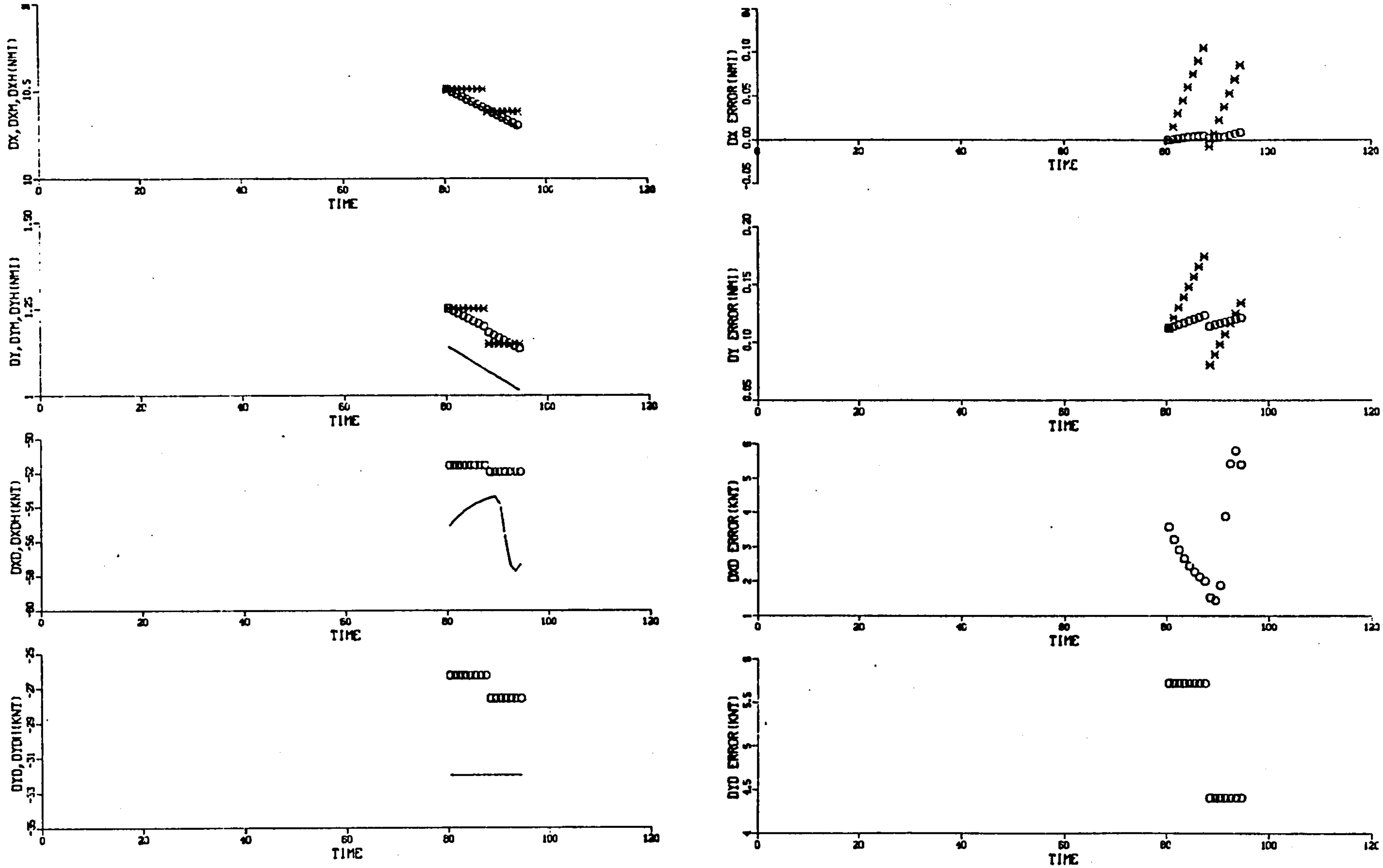


Figure 31(i) Relative Position, Velocity and Errors for A/C 552 - Case (4)

CAS Logic Validation

Only certain aspects of the collision avoidance logic were tested, since it is highly dependent on the chosen encounter scenerios. Validation efforts are pursued by running selected scenerios, and the simulation results are manually checked to explain discrepancies, if any.

Two methods were used to set up encounter scenerios. Both required modifications in the aircraft initialization routine, ACIC. One was to simply fix the initial conditions for the desired geometries. The other was to choose desired miss-distances (in three dimension) at future times and, by using the randomly selected velocities, the initial positions of the targets were calculated back to time 0. Therefore, the target velocities were constant but randomized. There are two significant points:

(i) The vertical tracker with the Mode C altitude reports with 100 ft quantization did not have much effect on the CAS logic output, once the tracker algorithm was established.

(ii) The randomized (Monte Carlo) selection of target velocities minimizes any subjective intent in the selected encounter scenarios. This feature reduces the number of experiements needed to validate the CAS logic module.

Own aircraft was allowed to maneuver in pitch in an open-loop manner (not in response to CAS RA advisories). Thus, Own's dynamic effect was a factor in the outcome. Figure 32 explains the terminology used in defining the encounter scenerio in the horizontal dimension. The important variables are the range, range rate, Own and target altitudes and altitude rates for defining the encounter in the range and altitude dimension.

Validation of the Proximity and Traffic Advisories were established by manually checking the computational steps in the DETECT subroutine and comparing the computer results. This aspect was tested to a certain

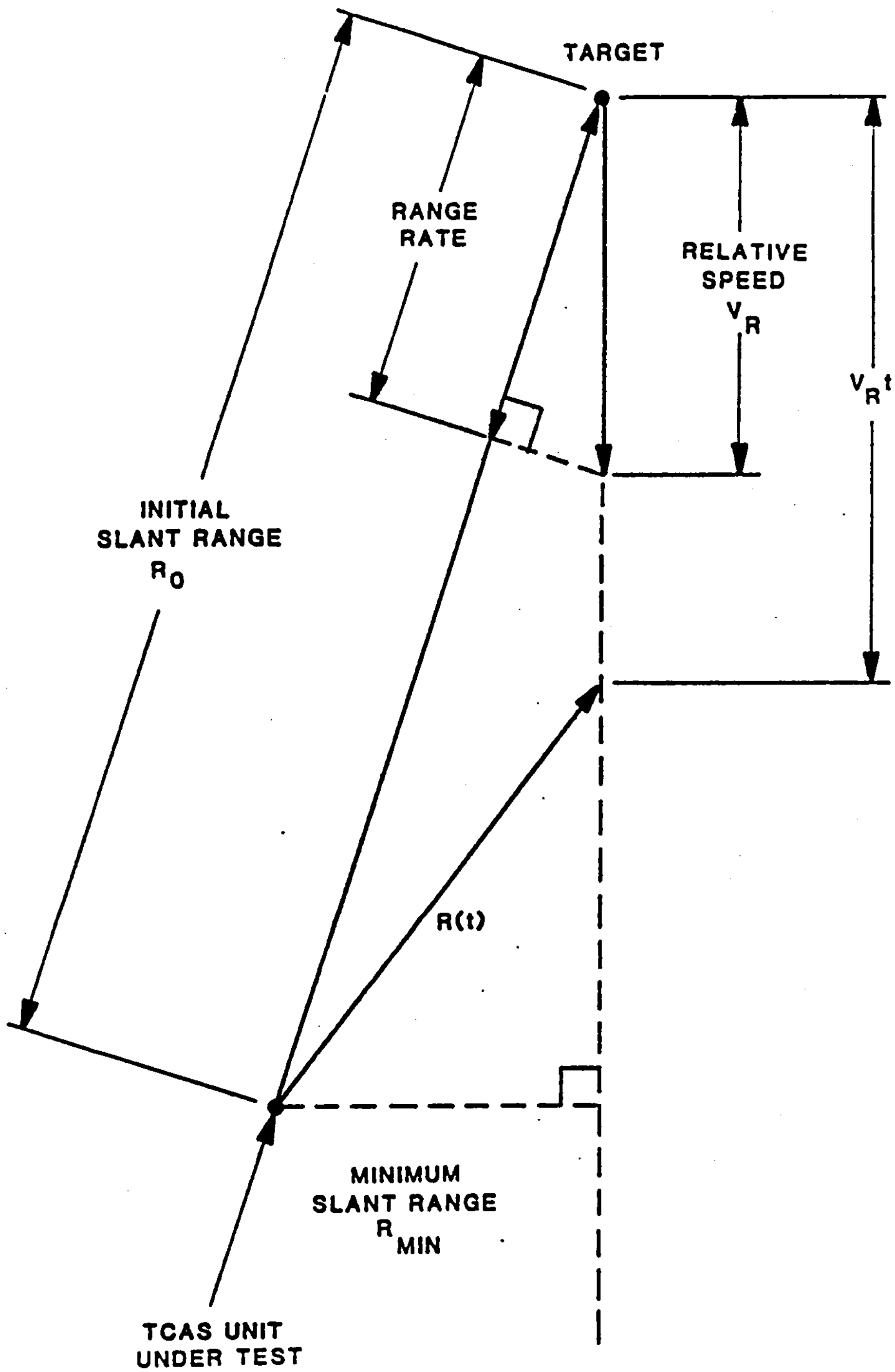


Figure 32. Definitions of Variables Associates
with the Linear Movements of Targets
in the Range Dimension

extent by the NASA/Langley piloted simulation since PA and TA advisories were used to color code the concerned traffic within the CDTI symbology. It is noted, however, that the current and active simulator versions are different. The most notable difference is in the counting procedures in the threat detection logic. The older version relied on the so-called two-out-of-three rule; whereas the current version does not have this cushion.

The RA modules (detection, sense and escape maneuver selections) are validated in similar ways by mostly manual check of the computation steps. Table 29 contains a partial list of CAS encounters tested during this effort. The table lists the initial conditions in terms of Own altitude and altitude rate, target transponder equippage (2 → ATCRBS and 3 → Mode S), relative range and range rate, and altitude and altitude rate. These initial conditions are defined 20 sec before the advisory issuance time. The CAS logic output is shown in the right most column. For example, Scenario No. 1 shows that Own and target are both flying level at 11,000 and 11,600 feet respectively. The target is equipped with a Mode C transponder, and the relative range was 6 nmi at the closing rate of 360 kt (a negative sign indicates that this is a converging target). Twenty seconds into the encounter, an RA is issued. At that time, the range and modified range closure times (taus) are 39.5 and 37 sec, respectively. The specific resolution advisory is a negative (LTENT = 10100 → don't climb). See Table 26 for the LTENT definitions.

Scenarios (1) through (7) are adopted from the suggested test scenerios contained in the TCASI II MOPS [2]. These are rather simple encounters with Own flying level. Scenarios (8) through (19) are generated using the Monte Carlo initialization method. Brief discussions are given below.

Scenario (1): The minimum τ (range closure time) is 27 sec which is smaller than the threshold value of 30 sec at this Own altitude; the RA threat is declared. The current and future vertical separation is 600 ft, which is less than the detection threshold of 750 ft, and larger than the RA threshold of 440 ft. Since the threat is above, the negative "don't climb" advisory is issued.

Table 29 CAS Logic Validation Cases

No.	z_0 (ft)	$\dot{z}_0$ (fps)	EQP	r (nmi)	$\dot{r}$ (kt)	z_I (ft)	$\dot{z}_I$ (fps)	Results and Comments
1	11,000	0	2	6.0	-360	11,600	0	τ_1 = 29.5, τ_2 = 27.0 LTENT = 10100 (DCLM)
2	11,000	0	2	6.0	-360	10,760	0	τ_1 = 29.5, τ_2 = 27.0 LTENT = 00000 (CLM)
3	11,000	0	2	6.0	-360	11,200	0	τ_1 = 29.5, τ_2 = 26.3 LTENT = 00100 (DES)
4	11,000	0	2	8.0	-500	12,516	-33.3	τ_1 = 33.7, τ_2 = 29.3 LTENT = 10000 (DDES)
5	11,000	0	2	8.0	-500	9,517	-33.3	τ_1 = 33.7, τ_2 = 28.3 LTENT = 00100 (DES)
6	12,500	0	2	6.0	-360	11,800	0	τ_1 = 29.5, τ_2 = 22.1 LTENT = 00000 (CLM)
7	12,500	0	2	6.0	-360	12,100	0	τ_1 = 29.5, τ_2 = 22.2 LTENT = 00100 (DES)
8	21,508	0	3	1.15	-100	22,513	-13.5	τ_1 = 19.7, τ_2 = 11.0 LTENT = 00100 (DES) *Own filter was lagging with $\tilde{z}$ = -25 ft, $\dot{\tilde{z}}$ = 17.7
9	6,033	41.5	2	2	-166	7,190	17.2	τ_1 = 22.1, τ_2 = 16.8 LTENT = 11110 (DCLM 1000)
10	24,448	41.5	3	2.1	-101	27,212	-13.5	τ_1 = 32.5, τ_2 = 16.9 LTENT = 11110 (DCLM 1000)
11	20,903	34.0	2	3.11	-157.2	21,877	22.2	τ_1 = 27.3, τ_2 = 24.4 LTENT = 11110 (DCLM 1000) *Own filter was lagging with $\tilde{z}$ = 30 ft, $\dot{\tilde{z}}$ = 13.4

Table 29. CAS Logic Validation Cases Continued

No.	z_0 (ft)	$\dot{z}_0$ (fps)	EQP	r (nmi)	$\dot{r}$ (kt)	z_I (ft)	$\dot{z}_I$ (fps)	Results and Comments
12	8,079	19.4	3	1.27	-105	8,810	-13.5	$\tau_1 = 22.8$, $\tau_2 = 14.2$ LTENT = 00000 (CLM)
13	14,079	19.4	3	1.27	-105	14,820	-13.5	$\tau_1 = 22.8$, $\tau_2 = 14.2$ LTENT = 00000 (CLM)
14	14,337	-35.5	2	2.16	-152	11,437	22.2	$\tau_1 = 28.6$, $\tau_2 = 22.9$ LTENT = 11001 (DDES 500)
15	26,237	-35.5	2	2.16	-152	23,437	22.2	$\tau_1 = 28.6$ $\tau_2 = 22.9$ LTENT = 10000 (DDES)
16	14,872	-35.5	3	0.03	65.2	14,500	-13.5	$\tau_1 = 0$, $\tau_2 = 10$ LTENT = 10100 (DCLM) *Range Divergent
17	26,872	-35.5	3	0.03	68.3	26,500	-13.5	$\tau_1 = 0$ $\tau_2 = 10$ LTENT = 10100 (DCLM) *Range Divergent
18	11,626	-41.5	2	1.28	-52.1	8,198	24.5	$\tau_1 = 32.3$, $\tau_2 = 26.6$ LTENT = 10000 (DDES)
19	23,584	-41.5	2	1.26	-52.1	20,222	24.5	$\tau_1 = 31.4$, $\tau_2 = 25.7$ LTENT = 00000 (CLM)

Scenario (2): This situation is similar to (1) except that now the vertical separation is 340 ft. This is less than the RA threshold. Because the target is below Own, a positive climb is issued.

Scenario (3): This is similar to (2) except that the threat is 200 ft above, resulting in a positive "descend" advisory.

Scenario (4): In this scenario, the target is descending at 33.3 fps (2000 rpm). It would be within 540 ft of Own altitude at the critical time. Since the separation is not within the RA threshold a negative "don't descend" is issued.

Scenario (5): This is a symmetric case of (4). However, the worst case separation (vertical miss-distance) is within the RA threshold, and the target will be above during the critical interval; thus, a positive "Descend" is issued.

Scenarios (6) and (7): These cases do not need explanation. See (2) and (3).

Scenario (8): This was a case when Own was in the middle of a "pitch-down-to-level" maneuver. Consequently the Own altitude and altitude-rate are in error by -25 ft and 17.7 ft/sec, respectively. Figure 33 depicts the situation. Because Own has leveled-off and the miss-distance would be 460 ft, the correct RA would have been "Don't climb". Because the Own estimator was lagging the miss-distance computed from the estimates was in error by 357 ft resulting in the positive "Descend" advisory. Fortunately, the wrong advisory did not induce a worse situation.

Scenario (9): This case shows an example of a vertical speed limit. The miss-distance is approximately 100 ft over the critical interval. The target is above throughout the encounter. Thus, the descend sense is selected. The vertical rate of +2000 fpm is tested, and it fails. But the vertical rate of 1000 fpm satisfies the RA threshold. Therefore, this is selected as the speed limit, resulting in the "Don't climb faster than 1000 fpm".

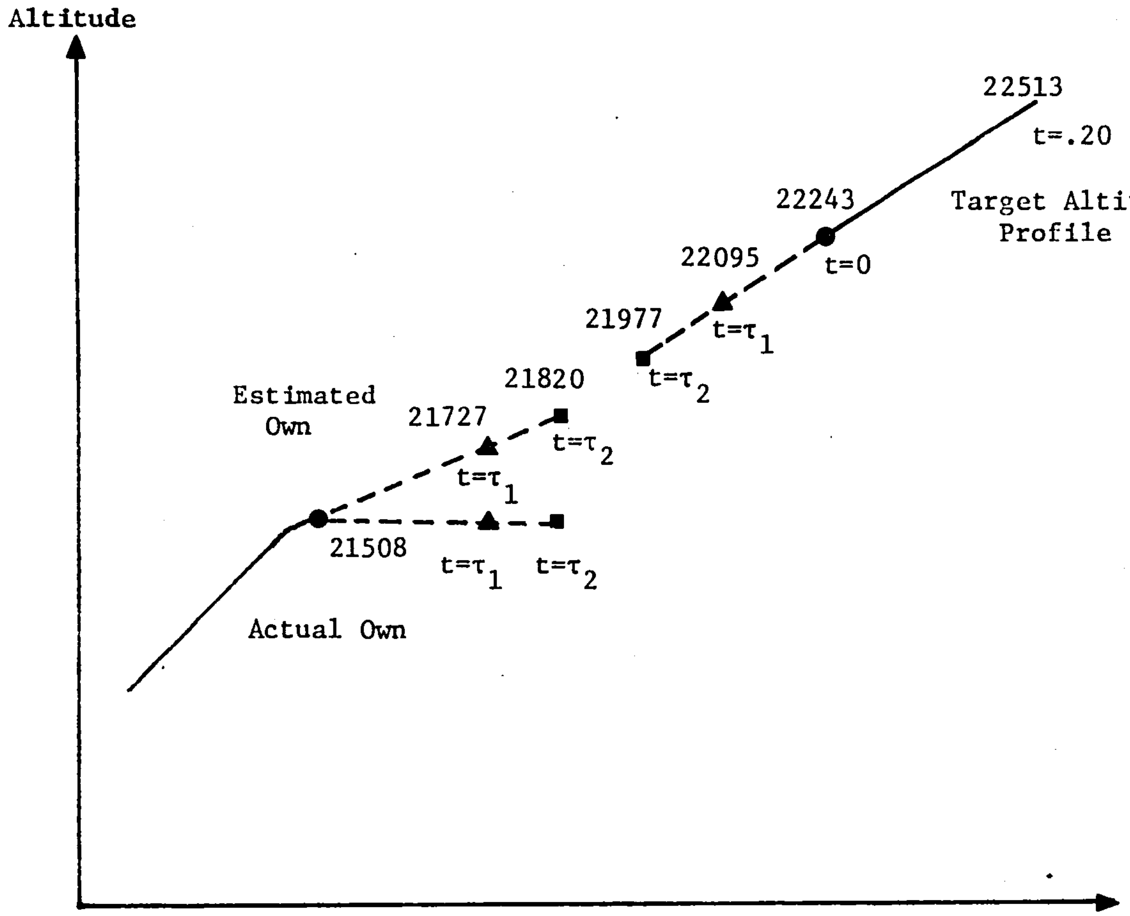


Figure 33. Schematic Diagram for Scenario (8)

Scenario (10): The vertical miss-distance is approximately 130 feet. During the critical time period, Own and target altitudes cross each other. The negative sense is chosen for obvious reasons. However, if the vertical speed was no faster than 1000 fpm, the initial separation will remain safe throughout the critical time period; thus, a VSL "Don't climb faster than 1000 fpm" is selected.

Scenario (11): This is a wrong advisory case due to Own estimation errors. Actual RA should be a negative "Don't descend" instead of "Don't climb faster than 1000 fpm".

Scenarios (12) and (13): These two are identical except for altitude. Because the miss-distances were smaller than the threshold value below 10,000 ft; therefore, the RA is identical for both cases.

Scenarios (14) and (15): These two are identical except for the altitude. The RA vertical threshold is 440 ft for (14), and 640 ft for (15). The vertical miss-distance is approximately 100 ft below Own. A positive (climb) sense was chosen. The vertical speed limits were tested sequentially. The smaller limit was satisfied first for (14). It took an extra "altitude rate margin" to satisfy the larger limit for (15). See Figure 34. However, DDES is one "rank" stronger than DDES 500.

Scenarios (16) and (17): Again, these are identical except for the altitude. These two cases are continuations of cases (12) and (10), respectively. Because of the antenna shadowing due to the pitch maneuver, this particular target came out on the "other side". Because of the target proximity, it passed the non-convergent range test. This is shown as τ_1 being 0 and τ_2 being limited to 10 sec. Because the current altitude separation is small for both cases, the selection logic selects a negative "Do not climb".

Scenarios (18) and (19): These are two identical encounters except for altitude. Because of the difference in threshold values at different altitudes, the chosen advisories are different. One is negative (Don't descend), and the other is positive (climb).

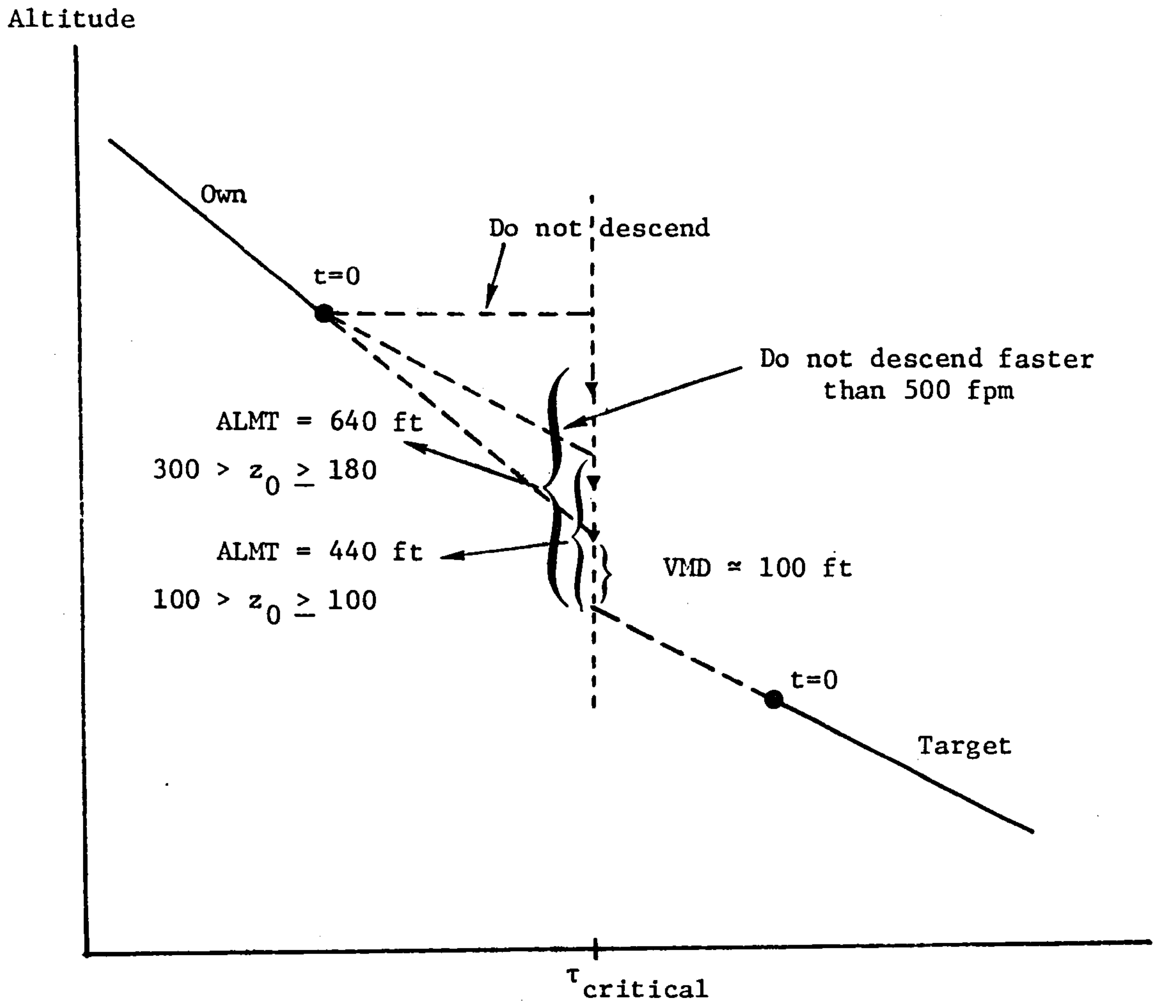


Figure 34. Comparison of VSL's at Different Altitudes

Many more encounter scenarios were examined than discussed in the text. We have reasonable cause to believe that the CAS logic is validated with respect to the operational and functional characteristics as discussed in the text. An examination of the test scenario indicates that the major portion of the CAS logic is captured faithfully in this simulation. The logic was tested with somewhat benign traffic patterns - no horizontal or vertical maneuvers by the target. Furthermore, no measurement errors were introduced to test the CAS logic robustness.

In conclusion, the simulation is validated from the input/output relationship view-point in the following sense:

(a) it obtains operationally correct surveillance/measurement data given a traffic pattern;

(b) it provides correct position and velocity estimates given the surveillance data;

(c) it makes correct threat assessment given the estimates; and

(d) it generates correct CAS advisory given the estimates and the threat assessment.

However, this does not imply that the resultant advisory is "correct" in avoiding a near-miss because the "correct" position and velocity estimates do not imply "error-free". It is felt that the altitude measurement and estimation errors would be the most critical elements.

V

CONCLUSIONS

A comprehensive simulation program was developed of the Enhanced TCAS II system based on the Bendix design. The program is written in standard Fortran IV, and it is highly transportable among main frame or mini-computers (CDC-7600, CRAY X-MP, VAX 11/780). The size of the program is approximately 3,000 lines of source code including a simple traffic generation module. It is designed so that it can operate in semi-real time. There is no reason why the program cannot be installed into a real-time simulation environment (without any modification), if the host executive has a one second background priority loop.

Both the traffic sensors and the collision avoidance logic modules are validated with respect to the known operational characteristics. Some limitations are imposed, mostly because of convenience and computational complexity. These are defined in the text.

The program is believed to be a valuable tool, not only for simulation but also for analysis purposes.

APPENDIX A

Major Program Commons

This appendix provides the list of major commons used by the program. The commons are listed in Table 18.

TABLE A.1/OWNTFL/and/ITLGT/COMMON DEFINITIONS

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
TIME	t	sec	System time
XO	x_0	nmi	Own north-east coordinate
YO	y_0	nmi	
ZO	z_0	ft	Altitude above MSL
VO	v_0	kt	Ground speed
ATT(1)	ϕ	rad	Attitude angles
(2)	θ	rad	
(3)	ψ	rad	
TBL(1,1)	$T_{BL}(1,1)$		T_{BL} (body to local-level) transformation
(1,2)			
.	.		
.	.		
.	.		
(3,2)			
(3,3)	$T_{BL}(3,3)$		
SFO	-	-	NOT ACTIVE
IALO	Ih_0	100 ft	Mode C altitude
RALT	ho_R AGL	ft	Radio altitude
ZGRND	z_{AGL}	ft	Altitude above ground level
ZOM(1)	$\hat{z}_0$	ft	Altitude and altitude rate estimates
(2)	$\hat{\dot{z}}_0$	fps	Altitude and altitude rate estimates
(3)	z_{0m}	ft	Own altitude measurements
NTZO	-	-	Altitude estimate initialization flag
ITRGT	-	-	IDF of pilot selected CDTI target
IWOW	-	-	Weight-on-wheel switch (1 → on ground
MANSEN	-	-	TOAS operating mode (0, 1 + 2 → stand-by; 3-7 → manual sensitivity; 8 → automatic)

TABLE A.2 /BNDXFL/COMMON DEFINITIONS

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
IDF*			Target Identification Number
IDM			Previous IDF (sign is used as tracker stations
MXN			Transponder type (1 → Mode A, 2 → Mode C, 3 → Mode S)
NXTS			= −|IDF| → no need for surveillance
IRPT			= 0 → invalid report
NPOP			= 1 → pop-up target
NSTS			Advisory status counter
TNXT	t_{NXT}	sec	Next scheduled surveillance time
TLST	t_{LST}	sec	Last valid surveillance time
RNGT	r	nmi	True range, bearing altitude
BRGT	b	rad	
ALTT	z_T	ft	
SFT	-	-	NOT ACTIVE
RNGE	$\tilde{r}$	nmi	Range and bearing errors
BRGE	$\tilde{b}$	rad	
RNGM	r_m	nmi	Range and bearing measurements
BRGM	b_m	rad	
EPSM	e_m	rad	Look-up angle
IALT	Ih_T	100 ft	Mode C altitude report
DXM	Δx_m	nmi	Measured north-east components
DYM	Δy_m	nmi	
DXH	$\Delta\hat{x}$	nmi	Horizontal position and velocity estimates
DYH	$\Delta\hat{y}$	nmi	
DXDH	$\Delta\hat{\dot{x}}$	nmi/sec	
DYDH	$\Delta\hat{\dot{y}}$	nmi/sec	
DXP	Δx^+	nmi	Coasted horizontal position
DYP	Δy^+	nmi	

*These are all arrays of twenty.

TABLE A.3 /ZTRKCFL/ and /ZWRKVR/Common Definitions

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
ICZ *	-	-	Initialization flag; also used as vertical profile status index
MALT(1) . . . (5)	$Ih(t_i)$	100 ft	Latest (1 = current) five valid Mode C reports corresponding to TMALS.
TMAL(1) . . . (5)	t_i	sec	Times of latest five valid Mode C reports
LSV(1) (2) (3)	$L(t_{LS})$	100 ft	Latest three Mode C level values corresponding to times TSV
TSV(1) (2) (3)	t_{LS}	sec	Latest three level switching times
TZIC	τ_0	sec	Initialization time
ZSV(1) (2) (3)	$z_m(t_{LS})$	ft	Altitude inferred at the time of devel switching times
ZH	$\hat{z}_I$	ft	Internal attitude and altitude rate estimates
ZDH	$\hat{\dot{z}}_z$	fps	
ZEXT	$\hat{\dot{z}}_T$	ft	External altitude and altitude rate estimates
ZDEXT	$\hat{\dot{z}}_T$		
*These are all arrays of twenty elements			
LOSWCH	-		Level switching flag (1 → level switched
ITST	-		
TDUMMY			Used as temporary memory; arrays of five elements.
ZDUMMY			

TABLE A.4 /WORKVR/Common Definitions

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
IDENT			Temporary identification number
INDEX			Temporary file index
ISURV			Surveillance flag
IACQ			Acquisition/track flag
MDX			Transponder type
DXTMP	Δx	nmi	
DYTMP	Δy	nmi	Temporary position variables
DZTMP	Δz	ft	
ALTMP	h	ft	
DUMM(1)			
(2)			
(3)			
(4)			
(5)			Temporary working memory
(6)			
(7)			
(8)			
(9)			
(10)			
RANGE	Δr	nmi	Range, bearing and look-up angle
BEARG	Δb	rad	
EPSI	Δe	rad	
EPSIMX	Δe_M	rad	Maximum look-up and down angle = $\pm 23^o$
RNGACQ(1)	Δr_{ACQ}	nmi	Maximum search range for Mode A = 17nmi
(2)		nmi	Maximum search range for Mode C = 17nmi
(3)		nmi	Maximum search range for Mode S = 25nmi
STMAX	V_{MAX}	kt	Max. target speed (250-600 kt)
RLIM	Δr_{rim}	nmi	Min. track range (5 or 10 nmi)
RMIN	Δr_s	nmi	Track range modification = 1.65 nmi
TAU	T	hr	Track range time period 1/80 hr (= 45 sec)
RNGTRK	Δr_{trk}	nmi	Track region range and altitude boundary
DHMTRIG	Δh_{trk}	nmi	Computed dynamically

TABLE A.4 /WORKVR/Common Definitions Continued

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
TMPWRK(1)			∿ RNGSQR*
(2)	r	nmi	∿ TSTRNG = range computed from x and y estimates
(3)			∿ TSTALT
(4)			∿ VDOTP
(5)	$\dot{r}$	nmi/sec	∿ RNGDOT = true range rate
(6)			∿ TSTRRD
(7)	τ_r	sec	∿ TAU1 = range closure time limited by TVPCMD
(8)			∿ SPDSQR
(9)	τ_r'	sec	∿ TAU2 = modified range closure time limited by TVPCMD
(10)			∿ TSTDST
(11)	$\dot{r}'$	nmi/sec	∿ RDTMP = range rates limited by RDTHR for 0 divide-check
(12)			∿ DELZD
(13)			∿ DZMOD
ALFAH(8)	α		Horizontal tracker gains. See Table 3.
BETAH(8)	β		
Q	q		Mode C altitude quantization = 100 ft.
SFMX	-	-	THESE SHOULD BE 0.
SIGZ	-	-	
SIGR	σ_r	nmi	Range error standard deviation = 0.0123 nmi
SIGB	σ_b	rad	Bearing error standard deviation = 0.01745 rad
DELTH	ΔP_v	ft	Surveillance altitude threshold = 250 ft
DELTP	ΔP_H	nmi	Surveillance horizontal position threshold = 0.1645 nmi
IRPTMX	-	-	Number of maximum allowable surveillance failures -1 = 4
PRBTHR(1)			Surveillance reliability for Mode A = 0.9
(2)	Prob		Surveillance reliability for Mode C = 0.95
(3)			Surveillance reliability for Mode S = 0.99

*Equivalenced in DETECT subroutine. These are mostly temporary in nature except ones with comments.

TABLE A.5 /CASFIL/Common Definitions

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
NTADV*		-	Number of threat advisories this cycle
NCAND+			Identification of threat candidates
IDCAS			Contains threat IDF of /BNDXFL/
IRACNT			RA, TA and PA counters and status flags
ITACNT			
ITAFLOT			
IPACNT			
IPAFLG			
TMCNT		sec	Absolute 10 sec timer
IDSTS			Advisory status flag (1 → PA; 2 → TA: 3 → RA and negative → pilot discretion)
IDCMD*			NOT ACTIVE
TMCAS	t_{CAS}	sec	Time of CAS advisory generation
ZDGOAL	$\dot{z}_{DG}$	fps	Individual vertical speed reference
TMCPA	t_{CPA}	sec	Time to CPA
XCPA	Δx_{CPA}	nmi	x, y, z position at CPA
YCPA	Δy_{CPA}	nmi	
DZCPA	Δz_{CPA}	ft	
ZDGOUT*	$\dot{z}_{OUT}$	fps	Over-all vertical speed reference
NTRA*			Number of RA threats
LTENT			Packed word
RATEXT*			RA test output

*Single variable

+ Array of 10 elements

All others are five element arrays.

TABLE A.6 /CWRKVR/ and /CWRKV2/Common Definitions

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
NDXCAS			Number of candidate threats
JDX			Temporary CAS file index
NORA			No RA capability flag
LVLOWN			Own sensitivity level
LVLINT			Target (and total) sensitivity level
LPA			PA, TA, RA Detection flags
LTA			
LRA			
TSTTAU	τ_r	sec	Range closure time
TAUVRT	τ_v	sec	Altitude closure time
ZDG	$\dot{z}_G$	fps	RA vertical speed reference
DELZ	Δz	ft	Current altitude separation
VMD	Δz^+	ft	(Projected) vertical miss-distance
INTENT			RA packed word
IHTHR			Within threshold (ALMT) flag
LSENSE			Climb/descend sense
LDCFLG			Don't care flag
LVSLOK			Vertical speed limit RA OK flag
LADNOK			Advisory Not OK flag
ZMPCLM		ft	Projected vertical separation for climb
ZMPDES		ft	Projected vertical separation for descend
ZPSECH		ft	Projected vertical separation for second choise

TABLE A.7 /CASPAR/Common Definitions

PROGRAM NAME	ENGINEERING SYMBOL	UNIT	COMMENTS
DMODT	Δr_{TA}	nmi	Minimum range guard for TA modified tau
DMODR	Δr_{RA}	nmi	Minimum range guard for RA modified tau
TAUTA	θ_{RTA}	sec	TA tau threshold
TAURA	θ_{RRA}	sec	RA tau threshold
TVPCMD	τ_{vc}	sec	Vertical tau limit value
H1	$\theta_{H_{TA}}$ (θHRA)	nmi^2/sec	$r \cdot \dot{r}$ threshold for diverging range
RDTHR*	$\dot{r}_{min}$	nmi/sec	Minimum range rate for 0 divide check = 10 kt
ZTHR*	θ_Z	ft	Detection altitude separation threshold
DZTHR*	$\delta\theta_Z$	ft	Vertical threshold modification = 60 ft
ALMT*	θ_a	ft	Resolution altitude separation threshold
ZDTHR*		fps	Minimum vertical speed for 0 divide check = -1 fps
ZDLVL*			NOT ACTIVE
DZDSML*			NOT ACTIVE
ZDCMD(7)			NOT ACTIVE
KSEQ(6,7)			NOT ACTIVE
RCOAL	θ_{RCA}	nmi	Minimum range at co-altitude
RTHRTA	$\theta_{\Delta rTA}$	nmi	Minimum range threshold
TAILDS*		nmi	Minimum range guard for modified tau = 0.247
RTSTMX*		nmi	Maximum effective RA range = 12 nmi

*Single variable

All others are five element arrays unless specified.

APPENDIX B

DV TCAS II MODEL CONSIDERATIONS

Introduction

In this appendix, DV TCAS II is discussed from the simulation viewpoint. The operational characteristics are summarized in Table 1. The DV TCAS II possesses a bearing accuracy of 6-8 deg rms which should be sufficient for most, if not all, of the perceived CDTI applications.

There are three major differences between the DV TCAS and BX TCAS, mainly in surveillance functions. These are:

(1) wide angel (90 deg) directive transmit antenna and quadrant monopulse receiver processing;

(2) regular interrogation interval (1 sec); and

(3) aircraft body fixed coordinate system.

The system is essentially a range-altitude system with the angle-of-arrival (bearing) information used to satisfy the cockpit horizontal display needs. In order to operate in a high traffic density (0.3 aircraft/nmi^2) area, extensive use is made of multiple level whisper/shout interrogations. This is the technique used to minimize "fruit" and "garbles".

The impacts of items (1) and (2) are several:

(a) smaller effective beam reach;

(b) different internal acquisition/track processing;

(c) somewhat less surveillance reliability which may be range dependent; and

(d) larger bearing error.

The impact of the body fixed corrdinate system results in a distortion of horizontal projection of the surrounding traffic because the Own body attitude is unaccounted for.

Implementation of the above aspects into the BX TCAS simulation program are considered further in the following sections.

Surveillance Function

Coverage volume, reliability and sampler logic. As stated above DV TCAS does not schedule interrogation. It transmits interrogation pulses over a 90 deg angular sector every 1 sec and waits for return signals. Therefore, the effective coverage is limited by the transmit and the reception power levels. These, in turn, "define" the surveillance reliability as a function of range. This means that the effective coverage volume is provided by a suitable surveillance reliability model. Table B.1 lists the probabilities of data drop-out (failed surveillance). It is noted that the probabilities depends on the range as well as the past drop-out characteristics. The coverage is given implicitly because the probability distribution depends on the actual range. That is, the probability of obtaining surveillance data beyond a certain range becomes extremely low; this, effectively limits the beam reach.

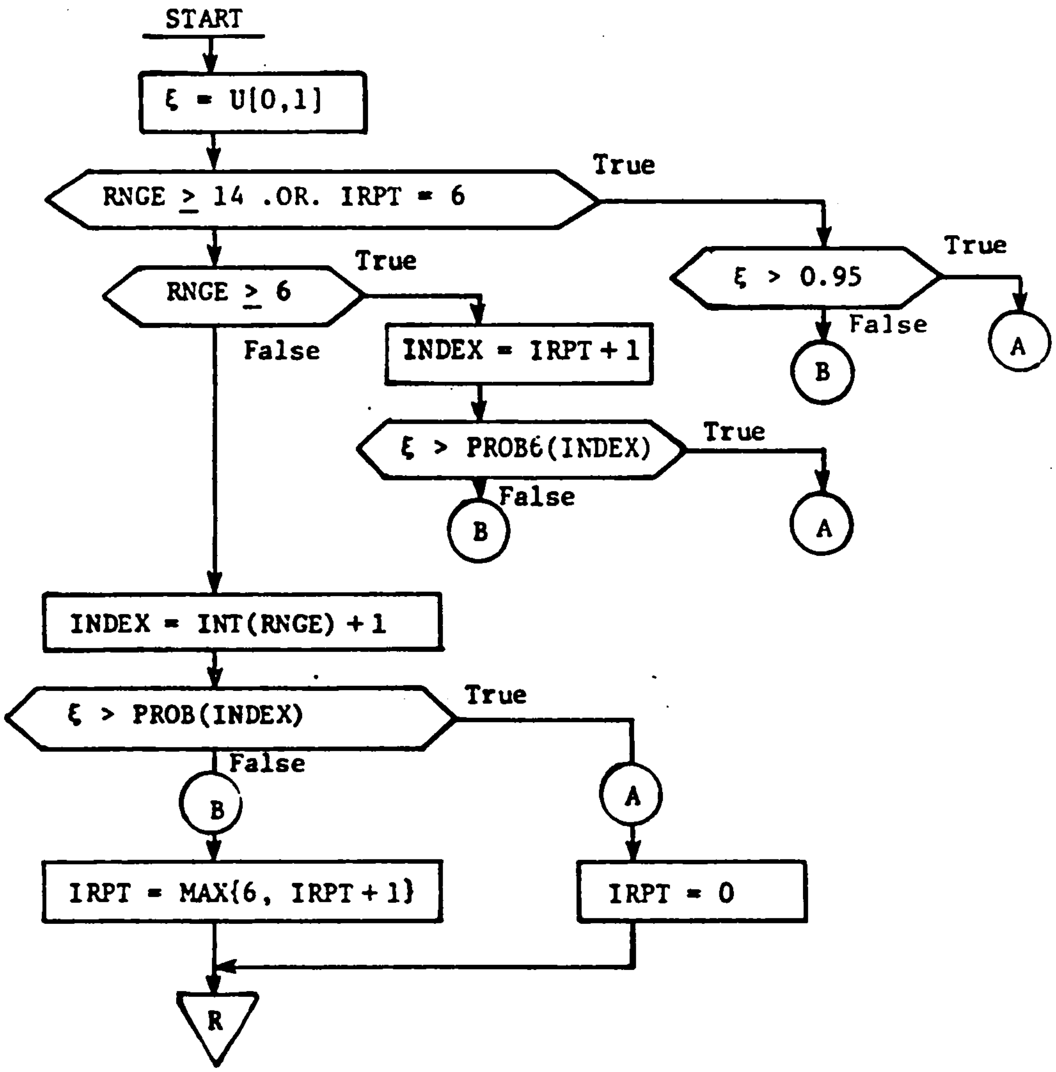
The model is accurate only for the aircraft whose track has been established. The "aquisition" phase is approximated by giving a small probability of valid signal reception at an extreme range (say 25 nmi).

Figure B.1 Shows a program flow chart which models the sampler logic. This subroutine must be invoked every second with the actual range and previous drop-out status. If the surveillance logic is successful, the routine returns with the flag IRPT set to 0; otherwise, the flag is incremented by 1. If the counter becomes 6, then the target is dropped from the internal track file.

Table B.1 Probability of DV TCAS Surveillance Data Drop-out

Condition	Probability of Data Drop-out
$0 \leq$ range < 1 nmi	0.1
$1 \leq \quad < 2$	0.13
$2 \leq \quad < 3$	0.15
$3 \leq \quad < 4$	0.15
$4 \leq \quad < 5$	0.18
$5 \leq \quad < 6$	0.21
($6 \leq$ range < 14).AND.	
IRPT [(a)] = 0	0.27
= 1	0.39
= 2	0.59
= 3	0.70
= 4	0.79
= 5	0.85
($14 \leq$ range < 20).OR. IRPT = 6	0.95 [(b)]

(a) IRPT = Number of consecutive drop-out
(b) Flight test results indicate 1.0



Note: Input = RNGE(nmi), IRPT
Output = IRPT
Data : PROB/0.1, 0.13, 0.15, 0.15, 0.18, 0.21/
PROB6/0.27, 0.39, 0.59, 0.7, 0.79, 0.85/

Figure B.1 Flow Chart for the DV TCAS Sampler Logic.

It is noted that the drop-out model was obtained for the so-called omni-directional active BCAS (which is a direct predecessor of DV TCAS II) based on flight test results. The reliability performance may have improved substantially since BCAS because of two design improvements: 90 deg antenna directivity and multi-level whisper/shout power sequencing.

Estimation

There are two requirements for the estimation functions within DV TCAS. One is the need for internal traffic file management including the target and measurement correlation by means of a gating technique. The other is the CAS logic requirement. As mentioned above, these are performed along the range and altitude axes. Therefore, the entire estimation functions are developed along these axes.

An alternative is to use the current algorithm and approximate the inverse transformation, as given in Eq (19). That is, the measured Δx and Δy are given by

$$\begin{aligned} \Delta x &= \Delta r \cos (\psi + \Delta b) \ , \\ \Delta y &= \Delta r \sin (\psi + \Delta b) \ . \end{aligned} \qquad (B.1)$$

here,

Δr = range ,

Δb = bearing ,

and

ψ = Own heading.

The resulting Δx and Δy measurements are then input to the filter algorithm module.

It is noted that the approximation given by Eq. (B.1) is fairly accurate if the roll and pitch angles are small. Also, it can be used to support a horizontal CRT display unit in the cockpit.

With the above two adjustments, the remainder of the simulation program should stay intact. By increasing the bearing noise level to 6-8 deg, the resultant program can be used to support the CDTI research effort reflecting a less capable system than BX TCAS.

REFEFENCES

1. Goka, T., "Analysis of Estimation Algorithms for CDTI and CAS Applications, " NASA CR-172444, 1984.

2. Anon., "Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System (TCAS) - Final Draft", RTCA Paper No. 106-83/SC K7-RO, Radio Technical Commission for Aeronautics, Washington, D.C., March 1983.

3. Sobocinski, R., "Flight Test Results with Collision Avoidance Systems," Paper No. 20-2. Proceed. of the Digital Avionics Systems Conf., Seattle, WA, October 1983.

4. Reed, J.E., "Enhanced TCAS II Test Results", Paper No. 20-3, Proceed. of 5th Digital Avionics Systems Conf., Seattle, WA, October 1983.

5. Anon., "Full BCAS Critical Design Review for the Engineerng Model--Task 1", Bendix Corp., Communications Division, Baltimore, MD, September 1981.

6. Read, J.E., "TCAS-II Engineering Model Design Specification (Model A and Model B)--Revision II, "Report No. BCD-TR-0063, Bendix Corp., Baltimore, MD, April 1982.

7. Sinsky, A.I., and B. J. Tier, "Coordinate Transformation Algorithms", Report No. BCD-TN-82-020, Bendix Corp., Baltimore, MD, May 1982.

8. Sinsky, A. I., "Alpha-Beta Tracking Errors for Orbiting Targets," Report No. BCD-TN-81-033, Bendix Corp., Baltimore, MD, June 1981.

9. Andrews, J. W., "An Improved Technique for Altitude Tracking of Aircraft," FAA Report No. ATC-105, MIT/Lincoln Laboratory, Lexington, MA, 1980.

10. Billman, B. R., "Analysis of a Nonlinear Altitude Tracking Method," FAA Report No. FAA RD-81-60A, FAA Technical Center, Atlantic City, NJ, February 1982.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-172445		

4. Title and Subtitle	5. Report Date
ENHANCED TCAS II/CDTI TRAFFIC SENSOR DIGITAL SIMULATION MODEL AND PROGRAM DESCRIPTION	OCTOBER 1984
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Tsuyoshi Goka	
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
Analytical Mechanics Associates, Inc. 2483 Old Middlefield Way Mountain View, California 94043	NAS1-16135
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitors: Roland L. Bowles and David H. Williams
Final Report

16. Abstract

The objectives of this project were to develop digital simulation models of enhanced TCAS II/CDTI traffic sensors based on actual or projected operational and performance characteristics. Two enhanced Traffic (or Threat) Alert and Collision Avoidance Systems were considered. The main focus is a system based on the FAA/Bendix design which dates back to the so-called full Beacon Collision Avoidance System (BCAS) concept. The other system is built by Dalmo Victor Division of Textron's Bell Aerospace Co. based on the FAA/MIT active BCAS concept.

Based on the engineering model, a digital simulation program was developed in FORTRAN. The program contains an executive with a semi-real time batch processing capability. Thus, the simulation program can be interfaced with other modules with a minimum requirement. The program is described in detail.

Both the traffic sensor and CAS logic modules are validated by means of extensive simulation runs. Selected validation cases are discussed in detail, and capabilities and limitations of the actual and simulated systems are noted.

The TCAS systems are not specifically intended for Cockpit Display of Traffic Information (CDTI) applications. However, these systems are sufficiently general to allow implementation of CDTI functions within the real systems' constraints.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
CDTI Traffic Sensor TCAS SIMULATION	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	167	